AUTOMATIC SMALL TARGET DETECTION IN INFRARED IMAGES OF VARIOUS BACKGROUNDS FROM VARIOUS DISTANCES

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

BY

OZAN YARDIMCI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JULY 2015

Approval of the thesis:

AUTOMATIC SMALL TARGET DETECTION IN INFRARED IMAGES OF VARIOUS BACKGROUNDS FROM VARIOUS DISTANCES

Submitted by **OZAN YARDIMCI** in partial fulfilment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by**,**

Prof. Dr. Gülbin Dural Ünver _________________
Dean, Graduate School **of Natural and Applied Sciences**

Prof. Dr. Gönül Turhan Sayan _________________
Head of Department, **Electrical and Electronics Engineering**

Assoc. Prof. Dr. İlkay Ulusoy Parnas _________________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Uğur Halıcı _________________
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. İlkay Ulusoy Parnas _________________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. A. Aydın Alatan _________________
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Cağatay Candan _________________
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Murat Efe _________________
Electronics Engineering Dept., Ankara University

Date: 14.07.2015

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, Surname: Ozan Yardımcı

Signature:

ABSTRACT

AUTOMATIC SMALL TARGET DETECTION IN INFRARED IMAGES OF VARIOUS BACKGROUNDS FROM VARIOUS DISTANCES

Yardımcı, Ozan
M.S., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. İlkay Ulusoy Parnas

July 2015, 160 Pages

Automatic detection of small targets in various backgrounds from far distances is a very challenging problem. Background clutter and small target size are the main difficulties which should be solved while reaching a very high detection performance as well as a very low computational load. In this thesis, various methods such as Top-Hat and Wavelet Transform, edge, filtering, saliency and feature based algorithms are investigated. All of the methods are compared using some realistic test scenarios, which are created synthetically. Precision, recall, processing time and number of user dependent parameters are used to evaluate the approaches. The comparative results indicate that no algorithm can detect the target with having high precision and recall at the same time in none of the scenarios. Besides, we have realized that the methods that are used in pre-processing, detection, thresholding and post-processing stages of the algorithms are very effective on the final results. Thus, the methods used in these stages are evaluated separately and the best approach for each stage is verified. Finally, an algorithm is constructed which constitutes of the best approach for each stage. However, although we end up with a very high precision rate such as 100%, the recall values are low. In this context, a post-processing method is proposed which increases the recall value while keeping the precision at 100% in prepared test scenarios. It is analyzed and indicated that the proposed post-processing method increases the recall value averagely 130% in all prepared test scenarios.

Keywords: automatic target detection, small target detection, infrared image, background estimation, morphological operations, wavelet transform, saliency detection.

ÖZ

DEĞİŞİK ARKA PLANLARDAN VE FARKLI UZAKLIKLARDAN ELDE EDİLEN KIZILÖTESİ GÖRÜNTÜLERDE OTOMATİK KÜÇÜK HEDEF TESPİTİ

Yardımcı, Ozan
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. İlkay Ulusoy Parnas

Temmuz 2015, 160 Sayfa

Küçük hedeflerin, uzak mesafelerden ve farklı arka plan koşullarında otomatik olarak tespit edilmesi, oldukça arzu edilen bir özelliktir. Arka planın karmaşık olması, hedefe olan mesafenin fazla olması, yüksek hedef tespit başarımına düşük işlem yükü ile ulaşma mecburiyeti çözülmesi gereken başlıca sorunlardır. Bu tezde, bu sorunlara çözüm olabilecekleri değerlendirilerek morfolojik işlemler, dalgacık dönüşümü, kenar tespiti, süzgeçleme, çıkıntı ve öznitelik tabanlı algoritmalar incelenmiştir. Bu algoritmalar, gerçeğe uygun olarak hazırlanan ve üç senaryolar oluşan sanal veri tabanında karşılaştırılmıştır. Kesinlik, geri getirme, işlem süresi ve kullanıcı tanımlı değişken sayısı değerlendirme ölçütü olarak kullanılmıştır. Elde edilen sonuçlar, hiçbir algoritmanın hazırlanan tüm senaryolarda yüksek kesinlik ve geri çağırım değeri oluşturmadığını göstermiştir. Bunun yanısıra, algoritmaların önişleme, tespit, eşik değerleme ve ard işleme yöntemlerinin toplam tespit başarımında oldukça etkili oldukları görülmüştür. Bu sebeple, bu aşamalarda kullanılan yöntemler ayrı ayrı değerlendirilmiş ve her aşama için en iyi yöntem belirlenmiştir. Sonuç olarak, her aşama için en iyi yöntemin kullanıldığı bir algoritma oluşturulmuştur. Fakat oluşturulan algoritmalar %100 kesinlikle hedef tespitini gerçekleştirirken, geri çağırım değerinin düşük olduğu görülmüştür. Bu kapsamda, hazırlanan test senaryolarında, hem kesinlik değerini %100 tutan hem de geri çağırım değerini yükselten bir ard işleme yöntemi önerilmiştir. Önerilen ard işleme yönteminin geri çağırım değerini hazırlanan senaryolarda ortalama %130 arttırdığı analizler ile ortaya konulmuştur.

Anahtar kelimeler: Otomatik hedef tespiti, kızılötesi görüntü, arka plan kestirimi, morfolojik operasyonlar, dalgacık dönüşümü, çıkıntı tespiti.

To My Dear Wife

ACKNOWLEDGEMENTS

I would like to express my thanks and deep appreciation to my supervisor Dr. İlkay Ulusoy Parnas for her guidance, positive suggestions, criticism and encouragement throughout this study.

I would like to thank my colleagues Seyit Tunç, Nuri Altürk and Ulaş Kürüm for their precious support and fellowship.

I am thankful to my “brother” Arda Özertem. His comments about my thesis were so helpful. He helped and encouraged me during this study. He will always be more than a friend to me.

I am also grateful to my parents (Yusuf Yardımcı, Bedriye Yardımcı). They supported me in all cases. They provided me comfortable and trouble less environment.

I would like to thank Roketsan for great environment. This research is funded by Roketsan. It had also provided datasets used in this study.

Finally, I am grateful to my wife Ceren Zeynep Yardımcı for her love. She always supported my career. This thesis is dedicated to her.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

LIST OF ABBREVIATIONS

APC	: Armored Personnel Carrier
ATD	: Automatic Target Detection
BTHT	: Black Top-Hat Transform
CTHB	: Cold Target On Hot Background
DCT	: Discrete Cosine Transform
FAR	: False Alarm Rate
FT	: Fourier Transform
HTCB	: Hot Target On Cold Background
IRSGS	: Infrared Scene Generation System
LAS	: Log Amplitude Spectrum
LCM	: Local Contrast Measurement
MBTHT	: Modified Black Top-Hat Transform
MWEC	: Mutual Wavelet Energy Combination
MWTHT	: Modified White Top-Hat Transform
RGF	: Robinson Guard Filtering
SE	: Structural Element
SNR	: Signal to Noise Ratio
SODD	: Second Order Directional Derivative
TEO	: Target Enhancement Operation
THT	: Top-Hat Transform
WT	: Wavelet Transform
WTHT	: White Top-Hat Transform

CHAPTER 1

INTRODUCTION

1.1 Small Target Detection

With the rapid development of digital image acquisition and processing technology, target detection has become applicable to civilian and military fields such as surveillance in public area, guidance in weapon system, automatic control in platforms and border security. Considering the current applications target detection in images is a very popular technology. The importance of this area has led to many studies being conducted over the last few decades. Target detection algorithms are specialized according to the applications and generally the system requirements determine the limit of the algorithms. For example, some systems have powerful computers and cover large areas. The advent of more powerful computers allows the use of more flexible, reliable and robust algorithms. Furthermore, the size of the area directly affects the requirements for the quality of the hardware used in target detection such as optics, detectors and electronic cards. There are also other systems which have limited capabilities due to its environmental conditions.

Target detection is usually used as a part of a system. After detecting target, it is expected that the procedure will continue with target recognition or target tracking [1]. Target detection can be performed automatically and/or manually depending on the system requirements. In some systems, the operator can choose the target to be tracked or recognized manually. For example, some tracking systems are used to calculate movement of a player in football matches. In this kind of system, manual target detection algorithms are used. Furthermore, identification of criminals is carried out using manual detection algorithms. However, in some systems, the operator cannot interfere with the system. For example, automatic target detection (ATD) algorithms are used in speed control systems. Another important use of ATD is in missile technology. In particular, in long range or high speed missile systems, ATD is required.

There are challenges in ATD which create difficulties in the process. The first problem is the highly cluttered backgrounds which increases the false alarm rate (FAR). If clutter is regarded as noise, a cluttered background decreases the signal-to-noise ratio (SNR). There are several components in the ground, sea and air that constitute background clutter. For the ground clutter, these are forests and buildings etc. For the sea clutter, they are sun glint, sun, clouds and land etc. For the air environment, sun and clouds are types of background clutter. Distinguishing the target from this background clutter is a huge problem and much work has been undertaken to manage this problem. A further problem is the size of the target. For example, in military applications, to gain more time to eliminate the possibility of threat, targets should be detected from long distances. This means detecting the targets when they are very small, even when they have similar characteristics as noise. This is a very challenging problem and the absence of a specific shape or structure of the target makes the job more difficult.

Generally, the characteristics of ATD algorithm are highly dependent on the platform that supports the imaging system. The type of platform varies depending on the application. In surveillance systems, cameras are fixed and the images are not affected by the motions of the platform. Inversely, images obtained from a moving platform camera will be affected by the motion and the target detection will be much more difficult.

In relation to moving platform ATD, two types of algorithms are presented in the literature. These differ according to the number of frames used to detect the target. ATD can be achieved by processing a single frame or successive frames in a moving camera system. When compared, these two approaches have advantages and disadvantages. For example, in single frame target detection, since there is no temporal information, FAR can be very high, on the other hand, need less processing time (PT). In multiple frame target detection, the detection results of consecutive frames can be fused and more accurate results can be obtained. However, conditions are especially difficult in moving camera applications since consecutive frames should be registered.

1.2 Literature Survey

This thesis considers the single frame ATD concept to detect small targets from infrared images. For this purpose, a literature survey is conducted mainly focusing on this concept.

Generally, single frame target detection algorithms follow two main processes. They either estimate the background beforehand then subtract it from the image to obtain target areas or target is detected directly from the whole image content. A simple ATD algorithm usually has three main stages. First is pre-processing which prepares the image for the second step and also fixes image problems. The second step is the detection step in which target information is extracted and the target is segmented from background. Usually this is achieved by thresholding. The last step is post-processing which aims to erase some false alarms.

In this thesis the algorithms are differentiated by the method used in detection step. Many different detection methods were found in literature. In most of the studies, generally morphological operations are used for background suppression [4]-[9]. Wavelet Transform (WT) is another commonly used method for both background suppression and target detection where high frequency components are considered as the target regions [10]-[14]. Edge detection based methods are usually used directly for target detection [15] [16]. In addition to these methods, target detection can be performed by filtering methods examples of which are mean and median filters [17] [18]. Saliency detection is another method used to detect targets directly, where salient regions (regions different from their surrounding) are extracted automatically [19]-[22]. In feature based target detection, features such as mean, standard deviation, maximum value and gradient difference are used to evaluate a region as background or target [23]-[25]. Markov Random Field (MRF) can also be used for target detection. However, it is usually used for bigger targets and it needs training stage [26]. Due to these properties, MRF is not suitable for the purpose of this thesis.

1.3 Overview of the Thesis

The aim of this thesis is to detect a small target from farthest possible distance in prepared infrared test scenarios. In order to construct reliable and robust ATD algorithm, test scenarios are highly critical. They directly measure the algorithm performance. For this purpose, they should be created in accordance with real circumstances. In this thesis, first, test scenarios were prepared by using Infrared Scene Generation System (IRSGS). The test scenarios characteristics were determined with literature survey. Then, the algorithms given in the literature were investigated in detail. The existence of an algorithm that produces best results in all scenarios has been investigated. Each stage of these algorithms were analyzed carefully and then applied to the test scenarios. To compare these algorithms, the following three metrics were used; detection performance, processing time and the number of user-dependent parameter (UDP). The detection performance is defined with two approaches known as precision and recall. Precision is related to false alarm and calculated by counting how many detected pixels actually belong to the target. Recall value is calculated by counting how many target pixels are detected after the detection process ends. From the results, the best detection performance is obtained using filtering based algorithm.

After comparing the algorithms obtained from the literature, each step of the algorithms were analyzed separately. Since the algorithms proposed in the literature did not accomplish target detection with high precision and recall in all prepared scenarios. This motivated us to analyze algorithm steps. All the pre-processing, detection, thresholding and post-processing methods of the algorithms were extracted. The best pre-processing, detection, thresholding and post-processing methods were determined by comparison. The best pre-processing method is a block based enhancement, in addition, the mean filter produced the best detection performance. All the thresholding methods had nearly the same performance so a thresholding method which calculates threshold value by using maximum and minimum value of the image is chosen because it does not require any manually set parameter. Finally, the Robinson Guard Filtering (RGF) based post-processing method is selected as the best because it increases the precision value. Thus, using the best methods for each step, the algorithm is constructed to increase the performance however, the mean recall value of this algorithm is still very low.

Finally, a new post-processing method is proposed to increase the recall value while keeping the precision value high in the prepared test scenarios. This proposed method directly focused on the pixels detected as target region. For each candidate target region, first the contrast level is calculated to determine detected region as target or background. If the region has low contrast then this region is determined to be background. If the region has high contrast then it is determined as target. After eliminating false alarms, the remaining target regions are analyzed in detail. For each target region the pixel values for the original image are considered. Each target region is expanded by two and a thresholding algorithm is applied to each expanded region to segment the pixels more accurately. These two approaches considerably increase detection performance. The effectiveness of this proposed post-processing method in prepared test scenarios is presented by applying it to the test images. The detection performance of the proposed method is given also with the precision and recall values.

1.4 Outline of the Thesis

The structure of the thesis is summarized below.

Chapter 2 is devoted to the test scenarios. This chapter includes both literature review and prepared test scenarios. First, the characteristics of the test scenarios are explained in the literature review section. Then, the prepared test scenarios and their specifications are given in the second section.

Chapter 3 focuses on the state of the art ATD algorithms, with the experimental results on the test scenarios and a comparison of the results. Firstly, the detection algorithms are investigated in detail. All the stages of the algorithms are clearly presented. Secondly, all outputs of the algorithms stages are presented in the second section. Finally, the results from the comparison of the algorithms; detection performance (precision and recall), processing time and the number of user dependent parameter are given.

Chapter 4 is devoted to the steps of the ATD algorithm. Firstly, the methods used in the pre-processing, detection and post-processing steps of the algorithms are investigated. The importance of the each step is emphasized. Secondly, the results of the individual comparison of the methods used in each step are shared.

Chapter 5 begins with a detailed explanation of the proposed post-processing method. It continues with the comparison results of the proposed method to be tested on the prepared test scenarios. The effectiveness of the proposed method is presented in terms of its detection performance; in particular, the high R and P values obtained from the proposed method are clearly described.

Finally, the work undertaken in this thesis is summarized and concluded in Chapter 6.

CHAPTER 2

TEST SCENARIOS

The calculation of algorithm performance is an extremely critical process. It is usually the case for test scenarios to be used to calculate algorithm performance in image processing applications. For this reason, test scenarios should be created realistically. Since ATD contains challenging problems such as background clutter and target size therefore these elements should be included in the test scenarios to correctly calculate performance.

A literature survey is conducted and the characteristics of the test scenarios are determined. Then, the property of the test scenarios were prepared and also generated using the IRSGS.

The test scenarios cover all properties of images as used in the literature for testing the algorithm. In addition, using a real tank model as a target and including dynamic images taken under controlled range with a certain trajectory are significant improvements with compared to those used in the literature as test images [4]-[25].

Before the test scenarios are created, research is conducted regarding the data characteristics used for the testing of the ATD algorithms in literature. The aim is to use more accurate and realistic approaches while creating the test scenarios using the data in the literature research.

Generally, algorithms are tested on images taken from ground, sea and air environments in literature [4]-[25]. But more importantly, the images are selected to include specific properties. These features are really critical and they directly affect the calculation of algorithm performance. These features can be assumed as variables. These variables can be grouped into three categories based on background, target and distance.

Background information is a very important variable in test scenarios. This information includes some other sub-information such as background type and contrast level. The test images used in the literature can be divided into two categories according to the background type; a hot target on cold background (HTCB) and a cold target on hot background (CTHB). Other sub-information related to the background is the contrast level between target and background. Decreasing contrast level makes target detection more difficult. Better algorithms should have high performance when there is a low contrast level between target and background. The researches in the literature used images which had both low and high contrast levels.

The target characteristics in a test scenario are other important variables. In the literature different types of the targets are used. In particular, ground, sea and air targets such as tanks, armored personnel Carriers (APC), ships and aircraft. The number of targets in an image is also critical however, the test images used in the literature are mostly with single target.

The size of the target is significant parameter for ATD since detection becomes more difficult when the size of the target becomes smaller. The distance of the target to the imaging system precisely determines the target size but this information is not available from the literature review.

2.1 Prepared Test Scenarios

After obtaining the characteristics of test scenarios from literature, scenarios were created using this information to produce realistic approaches. Test scenarios were synthetically obtained by using IRSGS in ROKETSAN Missiles Incorporation. IRSGS creates infrared images by adhering to the real circumstances. Some target and background models exist in IRSGS. Ground type background model infrared simulation can be created through IRSGS. A real tank model, truck and thermal surface can be used as a target in the system. The temperature difference between the target and the background (contrast level) can be adjusted by changing the system date and time information. Additionally, the simulation of a missile flight can also be undertaken by adding waypoint information to the system. Dynamic images can be obtained using this feature. Since, the target size and background change in consecutive frames in dynamic images. Figure 2-1 presents an example scenario of frames generated that correspond to approximate range between target and imaging system starting from the 3250 m range.

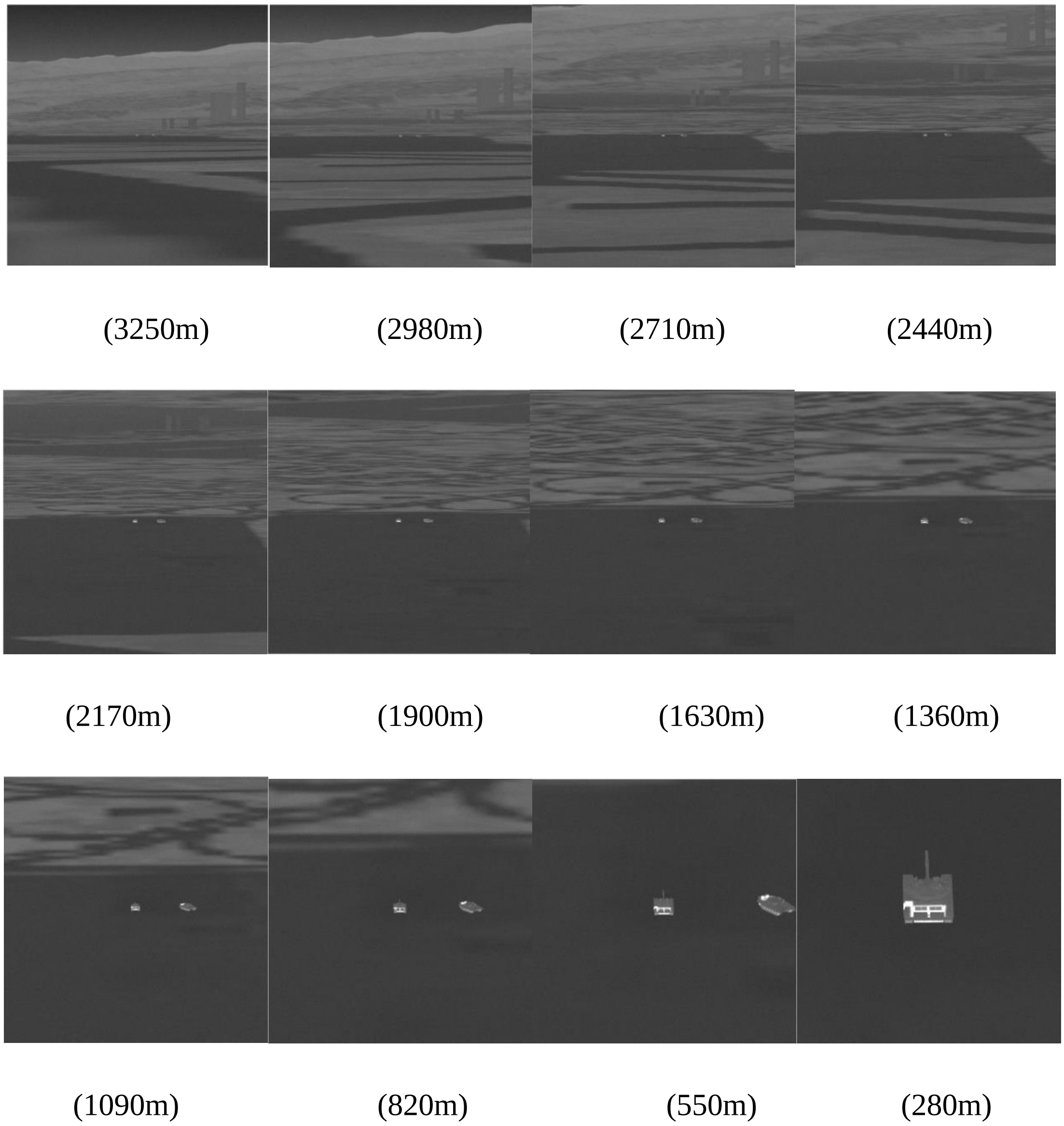


Figure 2-1: The frames taken from the IRSGS example scenario

A background model is selected for the test scenarios then the background details were determined. In total, three background details were prepared including the contrast level between the target and background. The best condition is defined as a condition where there are too many gray level differences between the target and background. This background detail is created as the first scenario. Target is highly visible and can be seen with the naked eye. Then, for the second scenario, the gray level difference between target and background is reduced providing a more difficult situation for detection. Finally, the contrast level between target and background is reduced to a level in which the target and background contrast is nearly zero. This is the third scenario and is assumed to be the most difficult one. All background details were created as HTCB target background characteristics.

In all but the third scenario, the target is a thermal surface. However, in the first and second scenarios, since, the real tank model does not allow the manual adjustment of the temperature, the thermal surface is used as a target to create the highest contrast level between the target and background. The orientation of target is adjusted so that the hot part of the tank engine is aligned with the point of view of the camera. The target number is set to three. All scenarios include three targets in each frame. All test scenarios were obtained from 3500 meter range flights towards the target. Since the frame rate and flight speed is known, the whole range between 3500m - 0m can be tested on a single scenario. The schematic representation of the prepared scenarios is given in Figure 2-2.

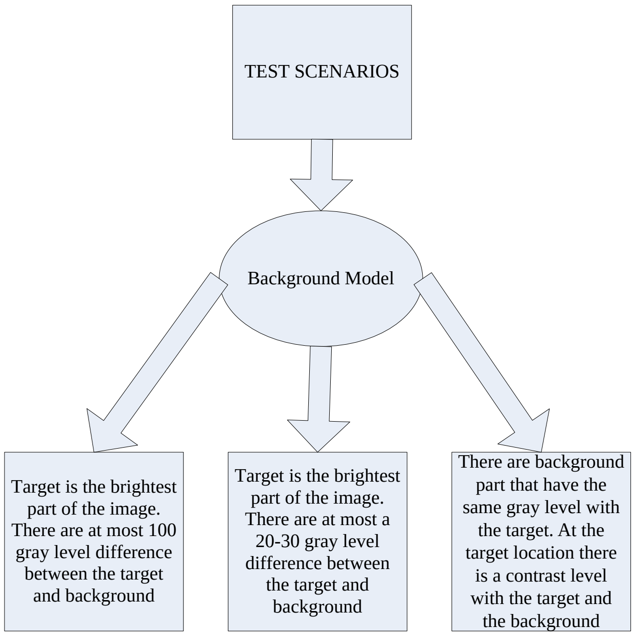


Figure 2-2: Schematic representation of the prepared scenario

The images taken from the first scenario are given in Figure 2-3. These frames represent the different distances between the target and imaging system. In the first scenario, the highest gray level value is the target which is used as a thermal surface and the contrast level is also quite high in the target region. This scenario can be defined as the simplest scenario due to the high contrast level between the target and background.

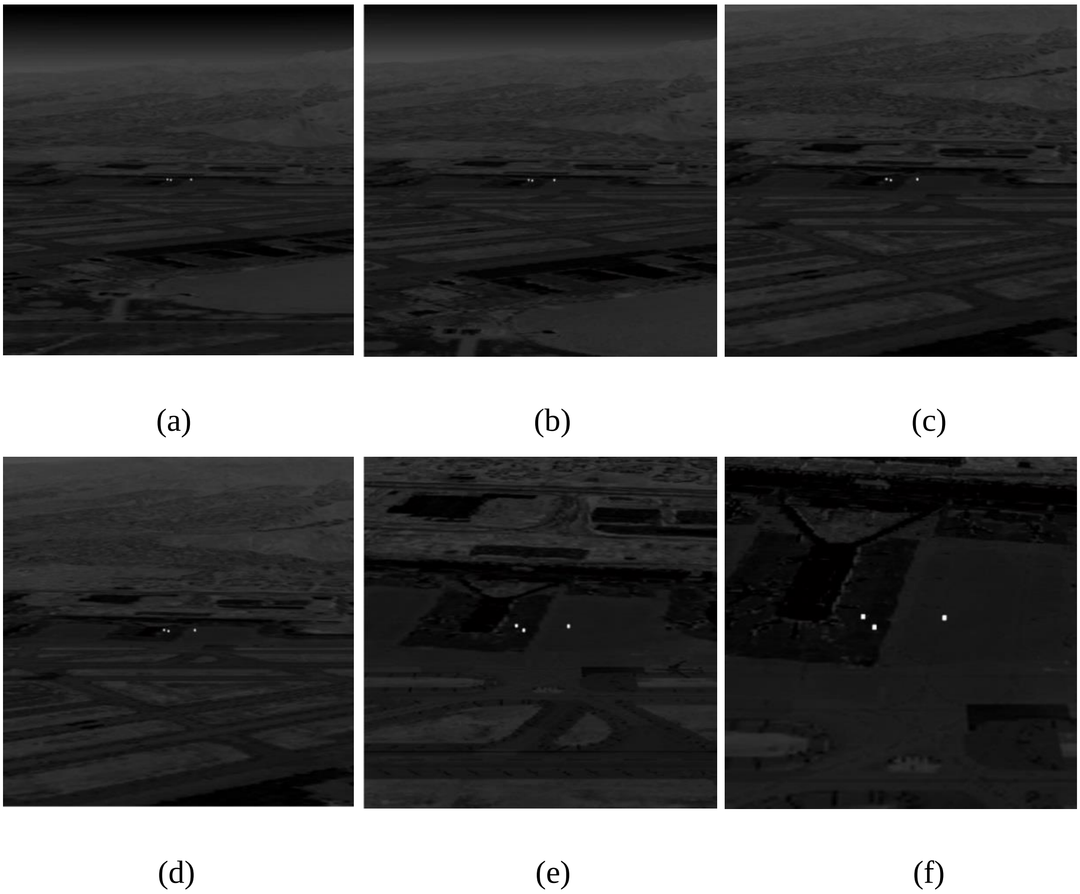
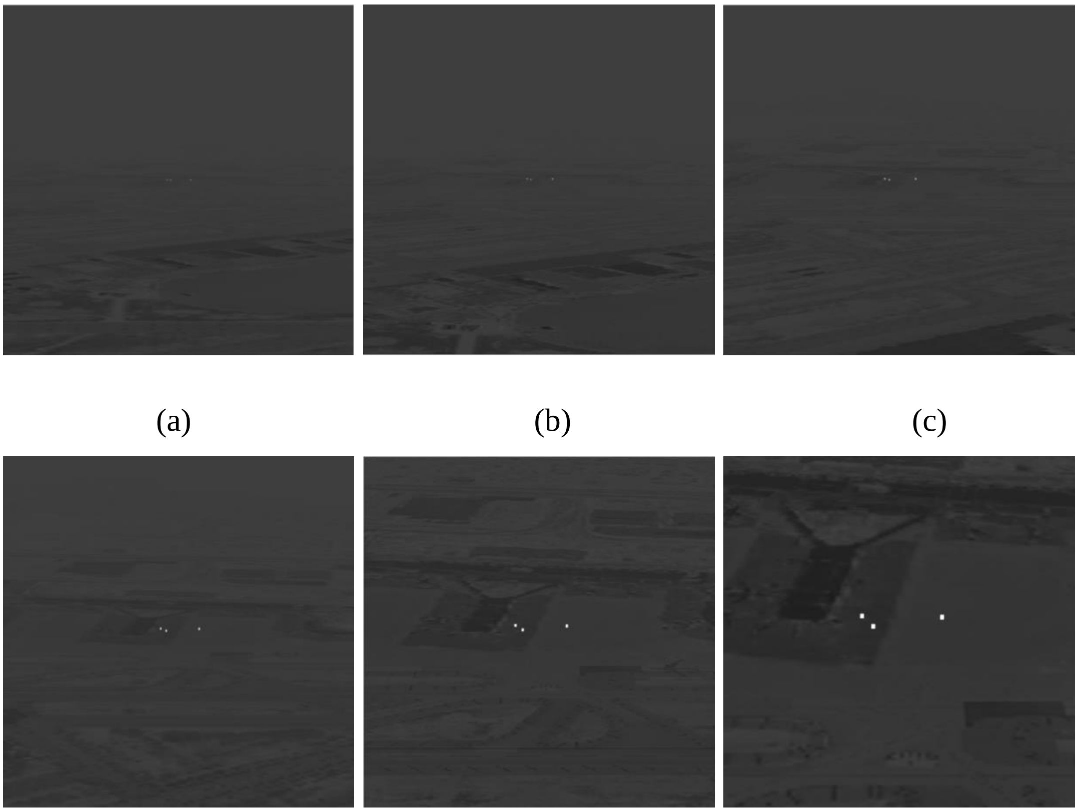


Figure 2-3: Frames taken from the first scenario

In comparison with the first scenario the second scenario given in Figure 2-4 is challenging in terms of target detection. In this scenario, although the targets have the highest gray level, the gray level difference between the targets and the background is less than the first scenario. A thermal surface model is used as the target in this scenario. To reduce contrast level between target and background, fog is added to scenario in the system.



(a) (b) (c)

(d) (e) (f)

Figure 2-4: Frames taken from the second scenario

The third scenario is very challenging. In this scenario, there is a background region that has nearly the same gray level as the target in the image. However, there is a gray level difference between the target and background part of the target. Frames taken from the third scenario are given in Figure 2-5. A real tank model is used in the third scenario. The system time is changed in order to adjust target background contrast level.

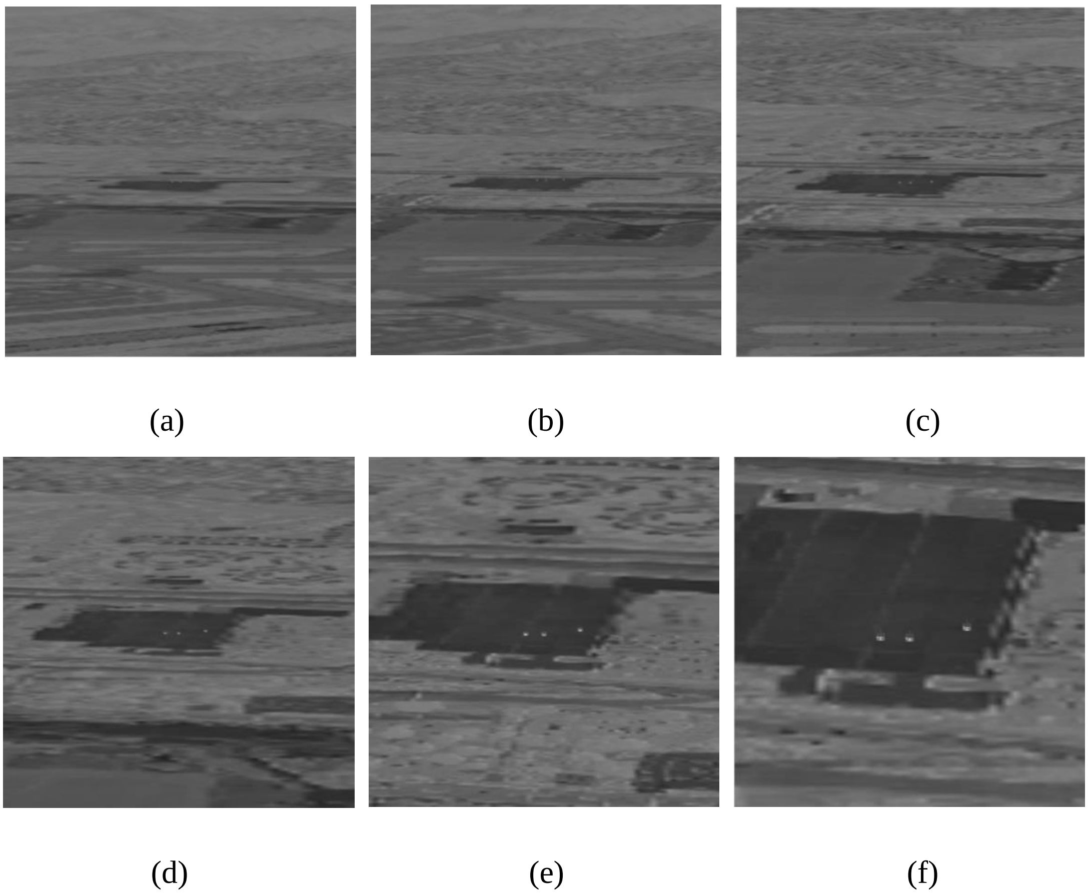


Figure 2-5: Frames taken from the third scenario

CHAPTER 3

AUTOMATIC TARGET DETECTION ALGORITHMS

3.1 Introduction

In this chapter, automatic target detection (ATD) algorithms obtained from the literature are described, applied to the test image and compared. First, the literature survey is presented and all algorithms are explained in detail. Each step of the algorithms is also given. Then, the experimental results of the algorithms are given respectively for each step. The effects of the parameters on the results are tested by adjusting their values where necessary. Finally, the algorithms are compared by increasing the number of test images selected from the prepared scenarios. The comparison of the algorithms is presented using detection performance, processing time and the number of user dependent parameter. Selected frames (20th, 100th, 200th, 300th, 400th and 500th) are used to calculate the detection performance defined with the precision and recall values obtained from the analysis.

3.2 Literature Review

Many studies in the literature are related to ATD. First, studies in the literature were separated into civilian and military applications on the basis of the test images used in these studies. Visible and infrared wavelength images are two types of test images generally referred to in the literature, and they have characteristic differences. In visible wavelength images light is used as an energy source on the other hand, temperature is used as energy in infrared images. Since visible test images are mainly used in civilian applications studies involving visible test images are mostly not present in the section concerning the literature research. In military applications infrared images are generally used therefore, studies in the literature using infrared images were evaluated.

The main goal in ATD is to detect targets as fast as possible from the farthest possible distance and with high precision. With these characteristics the overall system gains more response time. However, this situation requires the detection of very small targets and the size of the target is affected by the distance between the target and the imaging system. The detection of a target within complex background clutter is another desired property for an ATD system. These two properties are defined as the challenging situations and thus, have been the focus of much research [4]-[25].

The survey of the literature indicates that algorithms contain a maximum of three stages; pre-processing, detection and post-processing (Figure 3-1). For each stage, different methods have been proposed in the literature.

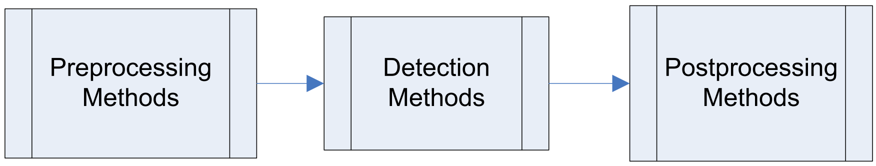


Figure 3-1: A flowchart of the stages in the detection process.

Generally, pre-processing methods prepare raw images for detection stage then, in the detection stage, the potential targets are identified using the features obtained from the image. Usually, targets are detected by utilizing feature differences between the target and background signal characteristics. The separation of target and background is undertaken in thresholding which is the final part of the detection stage. Post-processing methods are used to erase the false alarms produced by thresholding method. This stage is not must for ATD algorithms.

Pre-processing techniques are very useful in the detection of small targets [2] and in the literature there are many spatial and temporal filters used as pre-processing methods. The general purpose of using one of the pre-processing methods is to apply detection method more precisely and reduce the FAR [2][3]. If the targets are quite small in the image, they usually do not have shape or texture features. In these types of images, the SNR is normally very low and this makes it difficult to detect targets. In this context, pre-processing methods can be considered as having a role in increasing the value of the SNR.

Infrared imaging systems usually produce noisy images therefore, it is necessary to eliminate as much noise as possible before target detection. The Gaussian low-pass filter is a spatial filter that has a property of eliminating most of the noise in the image. In addition, spatial mean and median spatial filters are also used to remove noise. The median filter maintains edges in an image better than a mean filter [2]. A comparison of algorithms using temporal filtering compared with algorithms using spatial filtering shows that they perform the same function using multiple frames instead of a single frame. The Gaussian, mean and median filters also have applications in the temporal domain.

In literature, there are many algorithms that detect target from a single frame [4]-[25]. These algorithms commonly detect a target with two different approaches. The first group of algorithms estimates the background beforehand. Then the estimated background is subtracted from the image to obtain the target. The second group algorithms aim to directly detect the target information.

Morphological operations are used in many detection algorithms [4]-[9]. These algorithms are considered to be in the first type detection group. Morphological operations are generally used to suppress background in the image. There are different morphological operations given in literature, including; opening, closing, white Top-Hat Transform (WTHT) and black Top-Hat Transform (BTHT). Another commonly used detection method is wavelet transform [10]-[14] which represents image in both the spatial and frequency domain. With this method, both background suppression and target high frequency components detection can be achieved. Wavelet based algorithms can be placed in both groups. Edge detection is another method given in the literature [15] [16]. Edge information is used to detect the target in these algorithms that are considered to be the second type detection group. In addition to these methods, target detection can be performed by filtering methods [17] [18]. These spatial filters are the mean and median filters. These filters are used in pre-processing and also for background estimation [3]. Filtering based algorithms are grouped in the first type detection group. Saliency detection is another method used to detect a target in images [19]-[22]. These algorithms are concerned with the salient region of the image. Saliency is the feature that attracts the attention of the human eye when viewing an image. The main assumption of these algorithms is that the salient regions are highly related to the target. Saliency based algorithms are considered to belong to the second type detection group. Target detection can also be achieved through the detection of features in image [23]-[25]. These features are the mean, standard deviation, maximum value and gradient difference. Feature based algorithms also belong to the second type detection group. Another important part of the detection stage is thresholding which completes the target detection. After thresholding, in the image, target pixels are assigned as one and background pixels are assigned as zero. There are many thresholding methods used in the literature and they are described in the following sections.

Post-processing methods are usually used to reduce false alarms which are produced by the detection methods. One result of the literature research is that only Robinson Guard Filter [8] and a method that improves the background estimation performance by iterative approach [17] are used as post-processing methods.

3.3 Detailed Description of Algorithms

In this section, the algorithms found in the literature that are suitable to apply in the context of target detection are described. These algorithms have been grouped and named on the basis of the method used in the detection stage. Since some algorithms can be considered as both first and second detection type group to avoid confusion the algorithms were not grouped according to type. However, the group type of algorithms is given in the relevant sections and all stages of the algorithms are explained in detail.

3.3.1 Morphological Operations Based Algorithms

There are many mathematical morphological operations used in target detection in the literature [4]-[9]. These algorithms are considered to be of the first type detection group because they aim to estimate the background. Morphological operations are defined through two basic mathematical morphologies of erosion and dilation that are applied to the image with a structural element (SE) as shown in Equation (3.1) and Equation (3.2) respectively. The SE has three features; shape, size and value. In equations (3.1) and (3.2), the gray level image is represented by *f(x, y)* and the SE is represented by *s(x, y)*.

$$f \ominus s(x,y) = min\{f(x-x',y-y') - s(x-x',y-y')\} \qquad (3.1)$$

$$f \oplus s(x,y) = max\{f(x-x',y-y') + s(x-x',y-y')\} \qquad (3.2)$$

Opening and closing morphological operations are defined by using erosion and dilation. These operations are constructed by changing the order of erosion and dilation. Opening aims to estimate the dark background regions. Inversely, closing is used to estimate the bright background regions. The opening and closing operations are given Equation (3.3) and Equation (3.4) respectively.

$$f(x,y) \circ s(x,y) = (f(x,y) \ominus s(x,y)) \oplus s(x,y) \tag{3.3}$$

$$f(x,y) \cdot s(x,y) = (f(x,y) \oplus s(x,y)) \ominus s(x,y) \tag{3.4}$$

Finally, the THT is defined as opening and closing. The WTHT uses an opening operation as shown in Equation (3.5) and BTHT uses a closing operation Equation (3.6). Target regions that are brighter than background regions can be enhanced with a WTHT. Inversely, target regions that are darker than background regions can be enhanced with a BTHT.

$$f(x,y) - [f(x,y) \circ s(x,y)] \tag{3.5}$$

$$[f(x,y) \cdot s(x,y)] - f(x,y) \tag{3.6}$$

The algorithms presented in this section are all considered as the second type detection group because they primarily aim to estimate the background. Different morphological operations are used in detection stage such as the opening, closing and different versions of THT. They are explained in detailed respectively below.

The algorithm presented in [4] is called MO1 in this thesis. In this algorithm, two different filters are used in pre-processing stage. First, the median filter is used and its purpose is to reduce the noise in image. False alarms caused by noise are also reduced indirectly by the median filter. Then, the gray level intensity of the target region is increased using another filter (H). The H filter kernel is given in Equation (3.7). This kind of a filter kernel aims to enhance the square size targets.

$$H = \begin{bmatrix} 11111 \\ 11411 \\ 14841 \\ 11411 \\ 11111 \end{bmatrix} \tag{3.7}$$

Morphological operations are used in the detection stage of MO1. Square shape structural element is used in morphological operations. Two openings are performed using two different size SE one structural is selected to be smaller than the target and the other one is selected to be bigger than the target. Thus, the obtained estimated backgrounds differ from each other and can be subtracted from each other. The background estimated with the larger structural element is subtracted from the background estimated with smaller structural element. Opening performed with a large structural element eliminates noise in the image. A small part of the target still remains after opening however, target and noise pixels are mostly filtered with opening performed using small SE. Small values are obtained after subtraction in the background regions. Inversely, large values are obtained after subtraction in the target regions. The opening performed with different sizes of structural element produces this kind of a difference. The subtraction results in values that are smaller than zero are assigned to zero and other values are unchanged which are larger than zero. This readjusting is called an offset process and is given in Equation (3.8).

$$f(x) = \begin{cases} I_3(x,y), & I_3(x,y) \geq 0 \\ 0 \quad , & x < 0 \end{cases} \tag{3.8}$$

Thresholding is included in the detection stage. The adaptive thresholding method is used in MO1. The mean value of the image is calculated with non-zero pixels in this first cycle of thresholding method. Then, the zero valued pixels are left unchanged, the pixels between the zero and mean values are assigned to the mean value and the pixels higher than the mean value are unchanged. After the first cycle is completed, a new mean value is calculated in the second cycle. The pixels, lower than the new mean value, are assigned to zero and others higher than the mean value are assigned to one. The first cycle of the thresholding step is given in Equation (3.9).

$$f(x) = \begin{cases} 0, & f(x,y) = 0 \\ E, & 0 < f(x,y) \leq E \\ f(x,y), & f(x,y) > E \end{cases} \tag{3.9}$$

MO1 does not include post-processing stage.

The algorithm given in [5] is called MO2 in this thesis. There is no pre-processing stage in MO2.

A THT is used in the detection stage of MO2. The purpose of this algorithm is to reduce the structural element size dependency of the THT. For this purpose, circular shape different sized structural elements are used in the THT. The THT is performed with structural element of a size from 1 to 16. Then, the target and background pixels are separated from each other using the Otsu thresholding method which is a well-known method in the literature [5][8][12][24]. This method assumes that the image histogram is bimodal. The purpose of this method is to find the separation point of these two peaks in the image bimodal histogram. The Otsu thresholding method is concerned with the probability of the image pixels.

After separating the target and background pixels, the entropy of the target pixels is calculated. The structural element size that produces the smallest entropy value is assigned as the selected structural element size. The smallest entropy value is calculated when the image histogram has one or more peak [5]. A smaller entropy value means that the target pixels have high probability of occurrence in the image.

MO2 does not have any post-processing stage.

The algorithm given in [6] is called MO3 in this thesis. MO3 does not have a pre-processing stage.

The THT is used to detect the target in MO3. In the detection stage, WTHT and BTHT are used together. The purpose of this process is to detect both bright and dark targets in image. The results are fused by obtaining the maximum value of each THTs pixel by pixel.

An iterative thresholding method is used in MO3. The thresholding method is applied to the fused image in which there are many user-defined parameters. These are T_0, t_1, t_4, c, $\mathcal{E}$ and d. T_0 represents the number of target pixels, t_1 and t_4 represent the limits of the large threshold region, t_2 and t_3 represent the limit of the narrow threshold region, c is between 0-1 t is used to calculate the threshold value t_1. $\mathcal{E}$ is an increasing quantity in each step, d is defined as c+ $\mathcal{E}$ and is used to calculate t_2. This thresholding method matches the number of detected target pixels with T_0. If the number of detected pixels is higher than T_0, the threshold value t_2 is increased. It is assumed that the target pixels are more accurately detect using this iterative characteristic.

The algorithm given in [7] is called MO4 in this thesis. The most important part of this algorithm is the pre-processing method that is referred to as a mask image in the paper [7]. The reason for using this pre-processing method is to reduce the false alarm produced by the THT.

In the mask image method, the image is divided into blocks of sizes determined by the user. Then the mean values are calculated and subtracted from the pixels values in each block. The pixels that are more than the mean values are assumed to be the target pixels. The mask image calculation is given in Equation (3.10). In this formula, $f(i,j)$ represent the gray level image and N is the size of the masks. Thresholding is carried out for each block with α value.

$$Mask(i,j) = \begin{cases} 1, & \left| f(i,j) - \frac{1}{NxN} \sum_{(i,j)\in A} f(i,j) \right| > \alpha \\ 0, & otherwise \end{cases} \tag{3.10}$$

The threshold value α used in pre-processing stage is calculated automatically using a specific process. The calculation process of α includes three interconnected equations. This equation group is given in Equation (3.11). In this equation, μ_b and σ_b represent the mean and standard deviation of background respectively. P_f is the FAR. The value of α is determined using 10^{-3} as the FAR. The α value is increased iteratively to match the 10^{-3} FAR.

$$\begin{gathered} P_f = \frac{1}{2}\left[1 - erf\left(\frac{SCR}{\sqrt{2}}\right)\right] \\ erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x exp(-t^2)\, dt \\ SCR = \frac{|\alpha - \mu_b|}{\sigma_b} \end{gathered} \tag{3.11}$$

In detection stage of MO4, opening is performed on the pixels, which are assigned to one, and the pixels that remain the same are assigned to zero after the pre-processing stage. In this way, opening is performed on the pixels, which may belong to the target. The purpose of leaving the non-target pixels left unchanged is to reduce the noise produced by the THT. The thresholding step of the detection stage is applied to the Top-Hat transformed image. A simple thresholding method is used in MO4. The threshold value is calculated using Equation (3.12). In this equation, *minPixel* and *maxPixel* represent the minimum and maximum pixel values of the Top-Hat transformed image.

$$T = minPixel + \frac{1}{3}(maxPixel - minPixel) \tag{3.12}$$

The algorithm given in [8] is called MO5 in this thesis. There is no pre-processing stage in MO5.

In the detection stage, a modified THT is used. This method assumes that pixels higher than a specific threshold t are target pixels. False alarms produced by the THT are reduced using this modification. After obtaining the results for both modified white Top-Hat Transform (MWTHT) and modified black Top-Hat Transform (MBTHT), the target enhancement operation (TEO) is performed. This operation is defined in Equation (3.13). In this equation, $f(x, y)$ represent the gray level image, $MWTHT$ and $MBTHT$ represent the modified WTHT and the modified BTHT, respectively. There are three user defined parameters (α, β, γ) in Equation (3.13). Following the TEO, it is assumed that target pixel values are increased and the background pixel values are decreased.

$$TEO(x, y) = \alpha \times f(x, y) + \beta \times MWTHT - \gamma \times MBTHT \tag{3.13}$$

There is a post-processing stage in MO5 which is performed with a RGF filter with a nonlinear characteristic. The filter is applied to the thresholded image. This filter is used to reduce the number of false alarms. RGF is a 7x7 square filter as shown in Equation (3.14). The edge and center values of the filter consist of the corresponding image pixel values. The other values are assigned as zero. This nonlinear filtering process is given in Equation (3.15). The noise that cannot be eliminated by the THT is reduced using RGF.

$$Z = \begin{bmatrix} f1 & f2 & f3 & f4 & f5 & f6 & f7 \\ f24 & 0 & 0 & 0 & 0 & 0 & f8 \\ f23 & 0 & 0 & 0 & 0 & 0 & f9 \\ f22 & 0 & 0 & X & 0 & 0 & f10 \\ f21 & 0 & 0 & 0 & 0 & 0 & f11 \\ f20 & 0 & 0 & 0 & 0 & 0 & f12 \\ f19 & f18 & f17 & f16 & f15 & f14 & f13 \end{bmatrix} \tag{3.14}$$

According to Equation (3.15), if the X value is greater than the edge values (*fi*) then the maximum value of the *fi*'s is subtracted from X and assigned to a position where X is located. If the X value is smaller than the edge values (*fi*) then X is subtracted from the smallest fi's and assigned to the position where X is located. If X is between the smallest and biggest *fi*'s then zero is assigned to the position where X is located.

$$f(x) = \begin{cases} X - \max(fi), & if(X \geq \max(fi) \\ \min(fi) - X, & elseif\ (X \leq \min(fi)) \\ 0, & else(\min(fi) \leq X \leq \max(fi)) \end{cases} \tag{3.15}$$

This RGF procedure is applied to each candidate target region, which is determined using the Otsu thresholding method. After the RGF operation, the obtained image is thresholded. The threshold value is calculated using Equation (3.16). In this equation, m is the mean operation and σ is the SD operation. A user defined c constant is used to calculate the threshold value.

$$T = m(RGF) + c \times \sigma(RGF) \tag{3.16}$$

The algorithm given in [9] is called MO6. There is no preprocessing stage in MO6.

In the detection stage, a different type of structural element is used in the THT. Commonly, a flat structural element is selected in the traditional THT however, for MO6 a grid structural element is used. Equation (3.17) shows the grid structural element schematic that is applied to detect targets that have a transition region. 16 neighboring points are used in the morphological operations and other 8 center points are preserved in a flat structural element [9].

$$Z = \begin{bmatrix} Y1 & Y2 & Y3 & Y4 & Y5 \\ Y16 & 0 & 0 & 0 & Y6 \\ Y15 & 0 & X0 & 0 & Y7 \\ Y14 & 0 & 0 & 0 & Y8 \\ Y13 & Y12 & Y11 & Y10 & Y9 \end{bmatrix} \tag{3.17}$$

First, erosion is carried out using a grid structural element and then dilation is performed using flat structural element. This equation is shown in Equation (3.18). The gray level transition of the target is preserved using a flat structural element in dilation operation. There is no thresholding step in the detection stage and furthermore, MO6 does not have post-processing stage.

$$f(x,y) \circ s(x,y) = (f(x,y) \ominus s_{girdle}(x,y)) \oplus s_{flat}(x,y) \tag{3.18}$$

3.3.2 Wavelet Transform Based Algorithms

WT is another highly popular detection method reported in the literature [10]-[14]. The image is decomposed to low and high frequency components using WT. Figure 3-2 presents a test image decomposed by WT together with detail of the decomposed is also given.

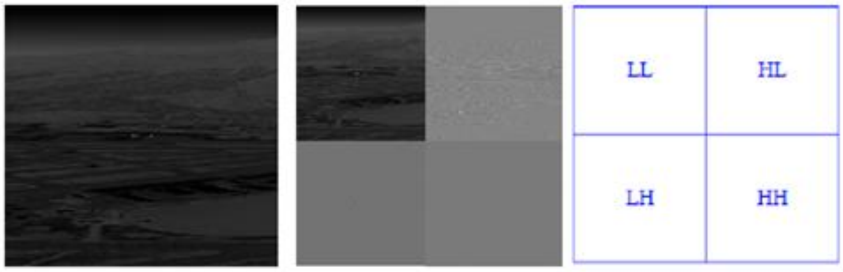


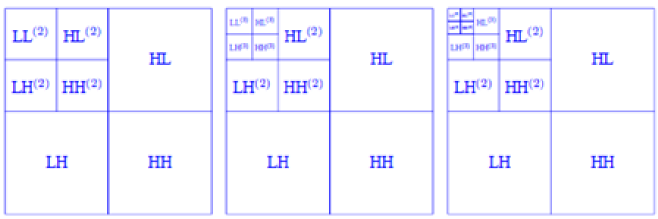
Figure 3-2: Decomposing of the example image with WT

WT is applied to the image in two dimensions. When WT is applied to the test image as given in Figure 3-2, four different types information are obtained. The first information is the approximated image of the original image (shown as LL in Figure 3-2). This information is obtained by filtering rows and columns of the image with low-pass filter. The second and third information are the edges of the image in horizontal and vertical directions (shown as HL and LH in Figure 3-2). These edges are obtained by filtering image with low-pass and high-pass filters. The last information is the diagonal edges of the image. Diagonal edges are obtained by filtering approximated image with high-pass filter (shown as HH in Figure 3-2).

The WT equation is given in Equation (3.19) [13]. In Equation (3.19), $\emptyset_{j,n}$ is the scale function, $\varphi_{k,n}$ is the wavelet function and J is the scale of WT.

$$F_i(x,y) = \sum_n c_{j,n}\emptyset_{j,n}(x,y) + \sum_{k=1}^{J}\sum_n w_{k,n}\varphi_{k,n}(x,y) \tag{3.19}$$

A multi resolution WT schematic diagram is given in Figure 3-3 [13]. All the scales are generated from the previous scales approximated image as can be seen in Figure 3-3. The approximated image obtained from each scale has different resolution and a different level of detail [13].



(a) (b) (c)

Figure 3-3: Schematic illustration of a multi resolution WT.

a) Second scale, b) Third scale, c) Fourth scale

WT can be used as both detection type groups. Background and target edges can be detected with WT. The algorithms obtained from literature are explained below.

The algorithm given in [10] is called WT1 in this thesis. This algorithm has a pre-processing stage in which a THT is used and the background is suppressed by using this transform.

In detection stage, WT is used to extract the horizontal and vertical high frequency components of the image. This process place WT1 into the second type of detection group. A single-scale sym4 wavelet family is used. The thresholding step of the detection stage is performed using the statistical information of the image. The threshold value calculation is given in Equation (3.20). In this formula, μ represents the mean value of the image and σ represents the standard deviation of the image. After calculating the threshold value, the pixels higher than the threshold value are assigned to one and the others are assigned to zero.

$$T = \mu + K \times \sigma \tag{3.20}$$

There is no post-processing stage in WT1.

The algorithm given in [11] is called WT2 in this thesis. There is no pre-processing stage in WT2.

In detection stage, firstly WT2 is applied to the image in order to extract the horizontal and vertical information of the image thus this algorithm is considered to be in the second type detection group. However, two scales of WT are used because the target has different characteristics in the first and second scale of WT. To benefit from this behavior difference in WT scales, a fusion process is performed. After obtaining horizontal and vertical component of the image in both scale, these information are fused. The fusion process is given in Equation in which, H_1 and H_2 are the horizontal detail of the image obtained with first and second scale WT respectively. In the same way, V_1 and V_2 are the vertical detail of the image produced by the first and second scale of WT respectively. H_{12} and V_{12} are the horizontal and vertical fusion results.

$$\begin{aligned} H_{12} &= abs(H_1) .* abs(H_2) \\ V_{12} &= abs(V_1) .* abs(V_2) \end{aligned} \tag{3.21}$$

After this fusion process, the coefficient preservation operation (CPO) is undertaken. The purpose of CPO is to eliminate the noise and reduce the false alarms in the detection stage. CPO is applied to the image in both horizontally and vertically. The horizontal CPO is given in Equation (3.22) and vertical CPO is given in Equation (3.23). Each value in a row is compared with threshold T_i. If the value is higher than T_i then the value is left as the same. If the value is lower than T_i, zero is assigned to this location. This process is repeated for each row in the image.

$$H'_{12}(i,j) = \begin{cases} H_{12}(i,j), & if H_{12}(i,j) > T_i \\ 0, & else \end{cases} \qquad T_i = m_i + k.\delta_i \tag{3.22}$$

In the vertical CPO, each value in the column is compared with the threshold T_j. The comparison process is same as the horizontal CPO. This process is also repeated for each column. Threshold values T_i and T_j are calculated using the mean (m) and standard deviation (δ) of each row and column.

$$V'_{12}(i,j) = \begin{cases} V_{12}(i,j), & if\ V_{12}(i,j) > T_j \\ 0, & else \end{cases} \qquad T_j = m_j + k.\delta_j \tag{3.23}$$

After performing the CPO, H'_{12} and V'_{12} are fused. Common parts of H'_{12} and V'_{12} are assumed to belong target and others are assumed to relate to the background clutter in the fusion process. This fusion process is given in Equation (3.24) and it is also carried out to reduce false alarms and increase the detail level of target information in detection stage.

$$HV = H'_{12}.* V'_{12} \tag{3.24}$$

There is no thresholding step in the detection stage of WT2. Target detection is achieved by finding the maximum value of HV. The maximum value is assumed to be the target in WT2.

There is no post-processing stage in WT2.

The algorithm given in [12] is called WT3 in this thesis. There is no pre-processing stage in WT3.

In the detection stage, a second scale WT is applied to the image. Then, the approximated image obtained with the second scale WT is reconstructed. The reconstructed image is assumed to represent the estimated background of the image therefore, due to this property WT3 is in the first type detection group. The estimated background is subtracted from the original image following subtraction, the target and noise pixels are left in the image. The detection stage is finalized using Otsu thresholding method.

There is no post-processing method in WT3.

The algorithm given in [13] is called WT4 in this thesis. WT4 is in the first type (background suppression) detection group. There is no pre-processing stage in WT4.

In detection stage, the WT and improved frame difference are used together**.** The aim is to detect both still and moving targets by fusing these two method's results and this frame difference method is very popular in moving target detection studies. The first step is to improve the frame difference in the detection stage however, the frame difference method is parameterized in WT4 and it is not performed with the consecutive frame as in the traditional methods. The improved frame difference equation is given in Equation (3.25). The difference number (k) of frames is a user-defined parameter.

$$d_F(x,y) = F_i(x,y) - F_{i-k}(x,y) \tag{3.25}$$

In the second step of detection, WT is used to suppress the background of the image. Third scale WT is performed and an approximated image is obtained. Then, it is reconstructed to estimate the image background. Finally, the estimated background is subtracted from the original image. After suppressing the background, the improved frame difference result and suppression result are fused. Two results are multiplied pixel by pixel in the fusion process and the detection stage is completed. There is no thresholding method in WT4.

The post-processing stage is also not included in WT4.

The algorithm given in [14] is called WT5 in this thesis. There is a pre-processing stage in WT5. However, this pre-processing method is optimized for the sea-sky background since it aims to find the horizon in the image. Therefore, a pre-processing stage for WT5 is not considered in this thesis.

In detection stage, Mutual Wavelet Energy Combination (MWEC) is applied to the image. MWEC is shown in Equation (3.26) in which the sgn function aims to find the signature of each element. D_j^h and $D_{j+1}^h$are horizontal details of the consecutive scales of WT. In this process the different scale WT horizontal details are fused.

$$D_e = sgn\{D_j\}x\left|D_j^h\right|x\left|D_{j+1}^h\right| \tag{3.26}$$

There is no traditional thresholding method in the detection stage. WT5 finds the target by calculating the contrast of the image. The contrast calculation is achieved with sliding window structures. The inner and outer windows are distributed to the image. In all window structures, the mean value of the outer window is subtracted from the inner window mean value. The maximum mean difference (contrast) is assumed to be in the target location. There is no information related to the window size given in WT5.

There is no post-processing stage in WT5.

3.3.3 Edge Detection Based Algorithms

Targets in infrared image have pixels that constitute the contrast level between the target and background. These contrasts produce the edges in the image. In literature edge detection is a common approach to detect targets [15][16]. Edges represent the high frequency component of the image. For this reason, two algorithms are considered as being members of the second type detection group (directly extract target information). These high frequency components can be detected using a gradient such as the Fourier transform (FT), WT, gradient and cosine transform the latter two-edge detection algorithms are considered in this section [15][16].

Edge detection is performed by taking gradient of the image and the calculation is achieved with filter kernels. For this purpose, filter kernel is convolved with the image.

The algorithm given in [15] is called ED1 in this thesis. Edge detection is used to detect target in ED1 and contains a preprocessing stage. The pre-processing method aims to suppress the background and reduce the number of false alarms. This method assumes that the pixels in the same row are exposed to the same atmospheric condition. In this context, the mean value of the pixels is calculated in the same row and this process is repeated for all rows in the image. This calculation is given in

Equation (3.27) in which, m_{ij} is the pixel value, j is the column number, i is the row number and n is the number of pixels in the row.

$$\overline{m_i} = \sum_{j=1}^{n} m_{ij}/n \qquad (3.27)$$

After calculating the mean image, it is subtracted from the original image. This kind of a calculation gives more reliable results. The background is suppressed by subtracting the mean image from the original image. In the detection stage, a multi-degree and multi-orientation gradient is calculated. This multi-orientation gradient calculation is shown in Equation (3.28). The gradient is calculated by taking difference of k distance pixels. This process is applied to the image both horizontally and vertically.

$$\begin{aligned} \Delta x_{+} &= |X(m,n) - X(m,n+k)| \\ \Delta x_{-} &= |X(m,n) - X(m,n-k)| \\ \Delta y_{+} &= |X(m,n) - X(m+k,n)| \\ \Delta y_{-} &= |X(m,n) - X(m-k,n)| \end{aligned} \qquad (3.28)$$

A multi-degree gradient calculation is applied to the image by selecting k as 1, 3 and 5 as given in Equation (3.28). After this calculation, thresholding is applied to the results which are given in Equation (3.29). In each degree, if the three directions are higher than the threshold then it is assumed to be the target and the value of one is assigned to these pixels. This process is repeated for three degrees. After this process, if the location is calculated as target two or three times in three degrees, then it is realized as real target. All these multi-degree and multi-orientation gradient calculations are aimed to reduce the number of false alarms and are intended to detect a real target.

$$\begin{aligned} Th_x &= m_x + \alpha.\sigma_x \\ Th_y &= m_y + \alpha.\sigma_y \end{aligned} \qquad (3.29)$$

There is no post processing stage in ED1.

The algorithm given in [16] is called ED2 in this thesis and edge detection used as a target detection method in ED2. There is a pre-processing stage in ED2 in which, firstly mean value of the image is calculated and subtracted from the image pixels. In this method the user defined pre-processing parameter is defined and it is this parameter that adjusts the enhancement level of target region. However, it is not always selected as the highest value because noise is also enhanced with the target. There is a trade of problem in the selection of this parameter. The pre-processing method is given in Equation (3.30).

$$I(x,y)_{enhanced} = (I(x,y) - \bar{I})^n \quad (3.30)$$

After completing the noise reduction process, discrete cosine transform (DCT) is applied to the image in both horizontal and vertical directions. After applying DCT, results are high pass filtered then, inverse DCT is applied to the filtered results whereby, the edges are detected by high-pass filtering. The high frequency spectrum is considered as the half the whole spectrum [16]. DCT and inverse DCT formulas are given in Equation (3.31) and Equation (3.32) respectively. In these equations, $f(i)$ is original image, $F(u)$ are frequency coefficients, M is the pixel number in rows and columns, and $\widetilde{f_i}$ is the reconstructed image.

$$F(u) = \frac{2C(u)}{\sqrt{M}} \sum_{i=0}^{M-1} \cos\frac{(2i+1).u\pi}{2M} f(i) \quad (3.31)$$

$$\widetilde{f_i} = \sum_{i=0}^{M-1} \frac{C(u)}{2} \cos\frac{(2i+1).u\pi}{2M} F(u) \quad (3.32)$$

After obtaining the horizontal and vertical gradients using the above procedure, Gaussian low-pass filtering is applied for smoothing. After the smoothing process, horizontal and vertical results are multiplied together (Equation (3.33)). Function G represents the Gaussian low-pass filtering.

$$CE = G(\tilde{f}_{vertical}) * G(\tilde{f}_{horizontal}) \quad (3.33)$$

Thresholding is applied to the multiplied image with a threshold calculated with the mean and standard deviation of the image. The threshold calculation is given in Equation (3.34). In the thresholding step, pixels higher than the threshold are assigned as one and others are assigned as zero.

$$T = mean(CE) + \alpha.std(CE) \quad (3.34)$$

There is a post-processing stage in ED2 in which erosion is performed on the detection result to reduce FAR.

3.3.4 Filtering Based Algorithms

Spatial filtering is also used as target detection technique [17][18]. Generally, the background is estimated using different filters and subtracted from the original image. These two algorithms are considered to belong to the first detection type group since they estimate the background in an image. The purpose of these algorithms is to estimate the low frequency component of the image such as background regions.

The algorithm given in [17] is called Fi1 in this thesis. There is no pre-processing stage in Fi1.

In the detection stage of Fi1, image is filtered with 7x7 size mean or median filters. Filtering is achieved by convolving image with a filter kernel. There is no thresholding step in the detection stage in Fi1.

The basic proposal is made for the post-processing stage in Fi1 in that it aims to reduce the FAR. The main assumption is that the still background can be estimated with filtering methods but it may not be possible to estimate the noise due to its random characteristics. For this reason, the post-processing method is used to reduce the estimation error as given in Equation (3.35) in which $Y_0(x,y)$ represents the original image and $Y'(x,y)$ represents the estimated background of the original image. The estimation error $\varepsilon(x,y)$ is defined as the difference between the original image and estimated background.

$$\varepsilon(x,y) = |Y_0(x,y) - Y'(x,y)| \tag{3.35}$$

In the post-processing method, $\varepsilon(x,y)$ is thresholded with η value to determine the pixels which belong to H_1 and H_0 in the image (Equation (3.36)). H_1 is used to represent target and noise signals, H_0 is used to represent only the noise signal (Equation (3.36)). The H_1 and H_0 signals are defined to separate the target and noise pixels.

$$\begin{gathered} X(x,y) \in \begin{cases} H_1, if\ \varepsilon(x,y) \geq \eta \\ H_0, if\ \varepsilon(x,y) < \eta \end{cases} \\ H_1: X(x,y) = S(x,y) + v(x,y) \\ H_0: X(x,y) = v(x,y) \end{gathered} \tag{3.36}$$

The background estimation is corrected with Equation (3.37). Here, the k parameter is used differently with the H_1 and H_0 pixels. If the corresponding pixel belongs to H_0 then the estimation error ε is assumed to be low. Using a high k value makes the corresponding H_0 pixel more similar to the value of the original image. Inversely, if the corresponding pixel belongs to H_1, then the estimation error ε is expected to be high. For this H_1 estimated pixel to be similar to the original image pixel then the k value should be selected as low. In the above procedure, a high k value is set very close to one and a low k value is set very close to zero.

$$Y(x,y) = k.Y_0(x,y) + (k-1).Y'(x,y) \tag{3.37}$$

The algorithm given in [18] is called Fi2 in this thesis. There is no pre-processing stage in Fi2.

In the detection stage, max-mean and max-median filtering methods are used instead of traditional mean and median filtering. Max-median filtering is shown in Figure 3-4 [18] and max-mean filtering has same characteristic with max-median filtering. In max-median filtering, the median of pixels in Z_1, Z_2, Z_3 and Z_4 directions are calculated respectively. Then the maximum value of these median values are selected and assigned to the central point of the window. The max-median filter reduces the noise more precisely than the traditional median filter, and it is better at preserving edges.

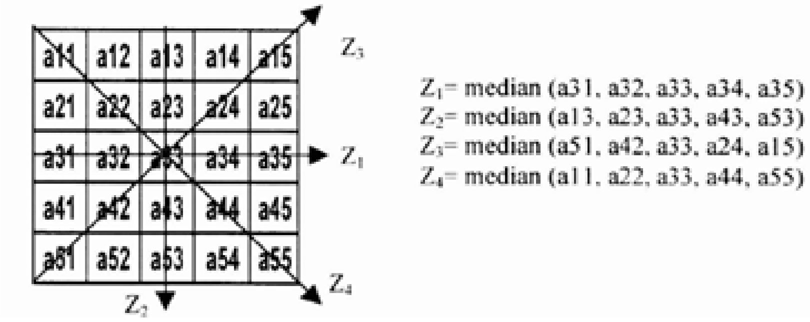


Figure 3-4: Max-median filtering

After estimating the background using the methods given above, it is subtracted from original image. Then, the threshold value is calculated with Equation (3.38). Threshold calculation is undertaken in each window with pixels of a size determined by user. After the threshold calculation, the pixels higher than the threshold are set to one and others are set to zero.

$$T_i = \mu(m,n) + k.\sigma(m,n) \tag{3.38}$$

3.3.5 Saliency Based Algorithms

Targets generally have features that distinguish them from the background. These differences are called saliency and it is directly related with the target information

[19]. For this reason, all saliency-based algorithms included in this thesis are considered as belonging to the second type detection group. The literature features some studies that used saliency to detect targets [19]-[22].

The algorithm given in [19] is called S1 in this thesis. There is no pre-processing stage in S1.

In the detection stage, a second order directional derivative (SODD) and FT are used together to detect the saliency in an image. For this purpose, first the SODD maps are obtained from the image. SODD basically calculates gradient of the image and this method is more robust to noise compared to the traditional Laplacian gradient method [19]. Three filter kernels are used in SODD. After calculating gradient of the image with given filter kernels (Equation (3.39)), directional derivatives are calculated with Equation (3.40) by changing the angle values.

$$W_4 = \frac{1}{70}\begin{pmatrix} 2 & 2 & 2 & 2 & 2 \\ -1 & -1 & -1 & -1 & -1 \\ -2 & -2 & -2 & -2 & -2 \\ -1 & -1 & -1 & -1 & -1 \\ 2 & 2 & 2 & 2 & 2 \end{pmatrix}$$

$$W_5 = \frac{1}{70}\begin{pmatrix} 4 & 2 & 0 & -2 & -4 \\ 2 & 1 & 0 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 1 & 2 \\ -4 & -2 & 0 & 2 & 4 \end{pmatrix} \qquad (3.39)$$

$$W_6 = W_4'$$

Filter kernel W_4 is used to calculate the vertical gradient of the image, filter kernel W_5 is intended to calculate the diagonal gradient and W_6 is used to calculate the horizontal gradient of the image. Gradients are calculated by convolving the image with the corresponding filter kernels. After calculating the vertical, diagonal and horizontal gradients of the image, SODD is completed with Equation (3.40). In this equation, K_4 is the vertical, K_5 is the diagonal and K_6 is the horizontal gradient of the image.

$$\frac{\delta^2 f(X,Y)}{\delta l^2} = 2K_4(x_0y_0)cos^2\alpha + 2K_5(x_0y_0)\cos\alpha\sin\beta + 2K_6(x_0y_0)cos^2\beta \qquad (3.40)$$

Four different SODD maps are calculated with Equation (3.40). These maps are generated by selecting α as 0 and β as 90 (channel1), α as 90 and β as 0 (channel2),

α as 45 and β as 45 (channel3), α as 135 and β as 45 (channel4). After calculating the SODD maps, they should be amended. This procedure is given below;

1) Set map values larger than zero to zero,
2) Inverse the whole map after all values are normalized to a fixed range [0, 1],
3) Filter the map using a 3 × 3 smoothing mask to smooth edges of targets and strips.

After amending SODD maps, the salient regions are extracted using FT. the saliency map is obtained by reconstructing the phase information of the FT result as given in Equation (3.41). In this equation, $p(u,v)$ represents the phase information of the FT result and $g(x,y)$ represents the Gaussian low-pass filter. After reconstructing the phase information, result is filtered with a Gaussian low-pass filter which has a standard deviation of 2.5.

$$S(x,y) = g(x,y) * \left\| F^{-1}\left[e^{i.p(u,v)}\right] \right\|^2 \tag{3.41}$$

After these processes, the channel1 and channel2 results are multiplied; channel3 and channel4 are also multiplied similarly. Then the results of the multiplication are added. These fusion processes aim to detect the target region in image. Before the fusion process, the maps of corresponding channels are normalized. No thresholding method is used in detection stage.

There is no post-processing stage in S1.

The algorithm given in [20] is called S2 in this thesis. There is no pre-processing stage in S2.

In the detection stage, information regarding intensity, orientation and shape are used as the salient features in an image.

The orientation information is obtained using a Gabor filter used in four different directions (0, 45, 90 and 135). The Gabor filtering formulas are given in Equation (3.42) in which θ_f represents the angle value and the orientation information is adjusted according to this parameter. In the last formula given in Equation (3.42), σ_x and σ_y are the scale parameters.

$$h(x,y) = g(x',y')\cos(2\pi\omega_f x')$$
$$(x',y') = (x\cos\theta_f + y\sin\theta_f - x sin\theta_f + y cos\theta_f) \tag{3.42}$$
$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2})$$

Shape information is obtained by using canny edge detection. This method can also detect weak edges and in addition, it gives a satisfactory performance under noisy images. The Gaussian low-pass filter variance used in Canny is selected as one. After obtaining the shape information, a total of five feature maps are obtained which have four orientations (0, 45, 90 and 135) and shape information. Saliency maps are extracted from this information. Nine scale Gaussian pyramids are applied to each information map to calculate the saliency maps. Each scale of the Gaussian pyramid is constructed by filtering using Gaussian low-pass filter by changing the variance of the filter in each scale. Then, (2-5), (2-6), (3-6), (3-7), (4-7), (4-8) difference of the pyramid is calculated for each piece of information. Finally, the difference of the information provided by six pyramids yield a total of 30 feature maps. For each feature map, the values are set to 0 and 1. Then, the normalization process is completed by multiplying all elements by a value which is calculated by taking the square of the difference between the maximum and mean value of the maps. In the final stage, the pixels higher than 20% of the maximum value on a map are assigned as one and others are assigned as zero. Then, all the spatial saliency maps are added and one final saliency map (S_m) is obtained.

The last piece of information is intensity and obtained with FT in the frequency domain. The reason for this is that the higher intensity produces a high frequency component in image. After applying FT to the image, then phase and amplitude information are extracted followed by the calculation of the log amplitude spectrum (LAS) then the LAS is filtered with a series of Gaussian low-pass kernels (different variance). After this operation, the filtered LAS (with a series of Gaussian kernels) and phase information are reconstructed to obtain series of saliency maps as given in Equation (3.43). In this equation, Λ represents Gaussian filtered LAS, P represent phase information and S_k represents series of saliency maps.

$$S_k(x,y) = F^{-1}[\exp(\Lambda(u,v;k) + jP(u,v))] \tag{3.43}$$

To obtain the final saliency map, entropy is calculated for each S_k and the minimum is selected as the final saliency map (S_p). Finally, the spatial saliency map and frequency saliency maps are fused. This process is given in Equation (3.44). The constant t is used for weighting the saliency maps.

$$S = S_m \times (t) + S_p \times (1 - t) \tag{3.44}$$

There is no post-processing stage in S2.

The algorithm given in [21] is called S3. There is no pre-processing stage in S3.

In detection stage of S3, FT is used to obtain saliency region in image. The basic approach is that the salient region produces high frequency components in image. For this purpose, the amplitude and phase information are extracted with FT. Then the LAS is calculated. The LAS is smoothed with a Gaussian filter (3x3). In the final stage, the difference between smoothed LAS and original LAS is obtained. Finally, the saliency map is obtained by reconstructing difference between the LAS and phase information with an inverse FT. No thresholding method is used to separate target and background pixels in S3.

There is also no post-processing stage in S3.

The algorithm given in [22] is called S4. There is no pre-processing stage in S4.

In the detection stage of S4, DCT is used to detect salient regions. This thesis also assumes that salient regions produce high frequency components. First, DCT is applied to the image then the positive coefficients are selected and an inverse DCT is applied to the selected positive coefficients. Here, the positive coefficients are considered to represent high frequency components. Then, the square of the inverse DCT is taken. In the last step, the result is filtered with a Gaussian filter and normalized. These operations aim to reduce the noise. No thresholding method is used in detection stage.

There is no post-processing stage in S4.

3.3.6 Feature Based Algorithms

Feature based algorithms detect targets directly using target information. Generally features that provide the difference between target and background are selected. This approach places feature-based algorithms into the second detection group (directly extract target information). Some feature based target detection algorithms are given in the literature [23]-[25].

The algorithm given in [23] is called Fe1 in this thesis. This algorithm uses a contrast level feature in the image to detect target. There is no pre-processing stage in Fe1.

In the detection stage, local contrast measurement (LCM) is used to calculate the contrast. LCM consists of two window structures. The first one is a large window and the second is a small window. Each large window includes 9 small windows (Figure 3-5). The large window size is selected as three times greater than the small window.

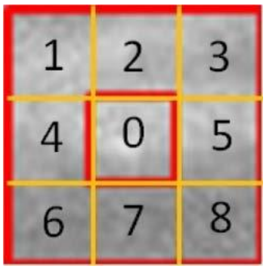


Figure 3-5: The sliding window structure

LCM is given in Equation (3.45). In the LCM process, first, the highest value (L_n) within the zero-tagged small window is calculated. The basic assumption in these operations is that the target is mostly located in the zero-tagged small window. Then, the mean values (m_i) of the neighboring eight small windows are calculated. These neighboring windows are assumed to belong to the background. C_n is calculated by dividing L_n by m_i and C_n is assigned to all elements of the zero-tagged small window. This value is considered as a contrast value of the corresponding large window.

$$C_n = \min(\frac{L_n^2}{m_i}) \tag{3.45}$$

The above contrast measurement process is applied to the whole image by sliding the window structure. After completing this process, the feature map is obtained. However, the use of a single window size may cause the loss of the target [23]. To solve this problem, LCM operation is applied to the image in different scales by changing the window sizes. The scales are defined with the large size window. The scale limit is dependent on the target size [23]. The maximum target pixels number is assumed to be 81 [23]. The scale upper limit should be set using this information.

After the LCM calculation is performed on the different scales, the results are fused. In the fusion process, the maximum contrast value is taken from each scale result to construct the final contrast map. The purpose of calculating LCM in different scales is to increase the performance of this method for different target sizes. Thresholding is applied to the final contrast map. The threshold value is calculated using Equation (3.46). In this equation, $\overline{I_c}$ represents the mean value of the final contrast map and σ_I^c represents the standard deviation of the final contrast map.

$$T = \overline{I_c} + k.\sigma_I^c \tag{3.46}$$

There is no post-processing stage in F1.

The algorithm given in [24] is called Fe2 in this thesis. In this algorithm, mean, standard deviation and gradient are used as the features to detect the target. There is no pre-processing stage in Fe2.

In the detection stage, three features are calculated. First is the mean value that is assumed to be high in the target region. The mean value calculation is undertaken with a sliding window. The sliding window size is determined by the user. The mean values are calculated within the window elements (Equation (3.47)). In this equation, $P_{i,j}$ is the image pixel value, m and n are the window sizes.

$$F_{x,y}^0 = \frac{1}{mn} \sum_{i=y-n/2}^{y+n/2} \sum_{j=x-m/2}^{x+m/2} P_{i,j} \tag{3.47}$$

The second feature is the standard deviation and it is also calculated using the sliding window (Equation (3.48)). The standard deviation value is assumed to be high in the target region. The window size is user defined. Calculated standard deviation values in each window are assigned to the corresponding window elements to construct $F_{x,y}^1$.

$$F_{x,y}^1 = \frac{1}{mn} \sum_{i=y-n/2}^{y+n/2} \sum_{j=x-m/2}^{x+m/2} P_{i,j}^2 - F_{x,y}^0 * F_{x,y}^0 \tag{3.48}$$

The third feature is the gradient and it is applied to the smoothed image (Equation (3.49). This process aims to reduce the sensitivity of the sensor noise. The gradient is assumed to be high in the target region.

$$F_{x,y}^2 = \left|F_{x-1,y}^0 - F_{x+1,y}^0\right| + \left|F_{x,y-1}^0 - F_{x,y+1}^0\right| \tag{3.49}$$

All the feature maps in different scales thus, in order to place them in the same scales, normalization is performed using the mean and standard deviation of the feature maps as shown in Equation (3.50).

$$\widehat{F_{xy}} = \left\{ \frac{F_{x,y}^0 - \mu_{x,y}^0}{\sigma_{x,y}^0}, \frac{F_{x,y}^1 - \mu_{x,y}^1}{\sigma_{x,y}^1}, \frac{F_{x,y}^2 - \mu_{x,y}^2}{\sigma_{x,y}^2} \right\} \tag{3.50}$$

After the normalization process, all the feature maps are summed to obtain one feature map. Thresholding is applied to the final feature map and a user defined threshold value is used. Values higher than the threshold are assigned to one and the others are assigned to zero. However, it sometimes produces false alarms. To handle this problem, feature maps are weighted. The weighting process is performed using likelihood ratio. The likelihood ratio is the ratio of the probability of target and background which is calculated for each feature map and given in Equation (3.51).

$$LR_{x,y}^{\widehat{F_{xy}^0}} = \frac{P(\widehat{F_{xy}^0}|T)p(T)}{P(\widehat{F_{xy}^0}|B)p(B)} \tag{3.51}$$

The target and background should be separated to calculate the likelihood ratio and it is achieved using the Otsu thresholding method. After calculating the threshold value in this way, all pixels higher than the threshold are assumed to belong to the target and others are assumed to belong the background. The probabilities are calculated using Equation (3.52).

$$P\left(\widehat{F_{xy}^0}\middle|B\right) = \frac{1}{\sqrt{2\pi}\sigma_{\widehat{F_{xy}^0}}} e^{-\left(\frac{\widehat{F_{xy}^0} - \mu_{x,y}^0}{\sigma_{x,y}^0}\right)^2} \tag{3.52}$$

The likelihood ratios are calculated for each feature map and the weighting process is completed by multiplying these likelihood ratios with feature maps as shown in Equation (3.53). The likelihood ratio is assumed to be high in the target pixels and inversely calculated as low in the background pixels.

$$LR_{xy} = \frac{\widehat{F_{xy}^0} * LR_{x,y}^{\widehat{F_{xy}^0}} + \widehat{F_{xy}^1} * LR_{x,y}^{\widehat{F_{xy}^1}} + \widehat{F_{xy}^2} * LR_{x,y}^{\widehat{F_{xy}^2}}}{LR_{x,y}^{\widehat{F_{xy}^0}} + LR_{x,y}^{\widehat{F_{xy}^1}} + LR_{x,y}^{\widehat{F_{xy}^2}}} \tag{3.53}$$

After the weighting process, thresholding is applied to the LR_{xy} with a user-defined threshold value. The pixels higher than the threshold are assigned to one and others are assigned to zero.

There is no post-processing stage in Fe2.

The algorithm given in [25] is called Fe3 in this thesis. There is no pre-processing stage in Fe3. Target detection is done by using four different features in Fe3. These are maximum gray level, mean difference, gradient difference and standard deviation difference.

In the detection stage, four feature maps are obtained. The first feature assumes that the target has the maximum gray level in the image. The maximum gray level is calculated with a sliding window which is given in Equation (3.54) and the window size is determined by the user. After finding the maximum value inside the window it is assigned to each corresponding window (f) location to construct the first feature map ($F^0_{x,y}$).

$$F^0_{x,y} = maximum(f(m,n)) \tag{3.54}$$

For the contrast level difference, this feature assumes that the target has the maximum contrast value in image. Contrast is calculated using two (inner, outer) windows. The window structure is given in Figure 3-6.

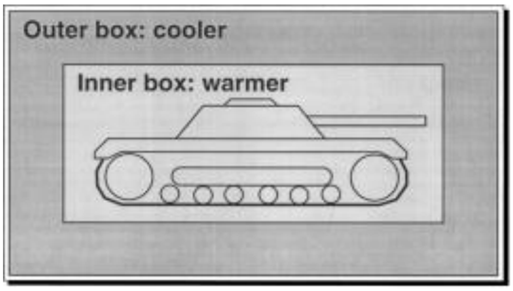


Figure 3-6: Window structure

The contrast is calculated by subtracting the outer window mean value from the inner window mean value (3.55).

$$F^1_{x,y} = \frac{1}{n_{in}} \sum_{m,n}^{Nin(x,y)} f(\mathrm{m,n}) - \frac{1}{n_{out}} \sum_{m,n}^{Nout(x,y)} f(\mathrm{m,n}) \tag{3.55}$$

The third feature is defined as the gradient difference of inner ($Nin(x,y)$) and outer window ($Nout(x,y)$) (Equation (3.57)). This feature assumes that the background has a low gradient and target has a high gradient. The gradient is defined as the first derivative of the image (Equation (3.56)).

$$G(m,n) = |f(m,n) - f(m+1,n)| + |f(m,n) - f(m,n+1)| \quad (3.56)$$

$$F_{x,y}^2 = \frac{1}{n_{in}} \sum_{m,n}^{Nin(x,y)} G_{in}(\mathrm{m,n}) - \frac{1}{n_{out}} \sum_{m,n}^{Nout(x,y)} G_{out}(\mathrm{m,n}) \quad (3.57)$$

The fourth feature which calculation is given in Equation (3.58) is defined as the standard deviation difference of the inner and outer window. This feature also assumes that the standard deviation difference will be high in the target region. Inversely, it should be low in background region.

$$F_{x,y}^3 = \frac{\sum_{m,n}^{Nin(x,y)} |f(\mathrm{m,n}) - \mu_{in}(m,n)|}{n_{in}} - \frac{\sum_{m,n}^{Nout(x,y)} |f(\mathrm{m,n}) - \mu_{out}(m,n)|}{n_{out}} \quad (3.58)$$

After calculating the four feature maps, normalization is applied to these feature maps using the mean (μ_k) and standard deviation (σ_k) of the corresponding feature map (Equation (3.59)).

$$\overline{F_{xy}^k} = \frac{F_{x,y}^k - \mu_k}{\sigma_k} \quad (3.59)$$

After the normalization operation, all feature maps are added and one final feature map is obtained. There is no thresholding step in detection stage. The maximum value is assigned as the target.

There is a post processing method which is called clutter rejection. However, this method is based on non-linear learning. Learning based methods not the concern of this thesis.

3.4 Experimental Results of the Algorithms

The algorithm’s results that have user dependent parameters change from scenario to scenario. For this reason, the parameters that produce the best detection result are selected. The results obtained with various parameter values are also given in this section. In this way, the selection of the most appropriate values for user dependent parameters allows for a more objective comparison of the algorithms given in the following section.

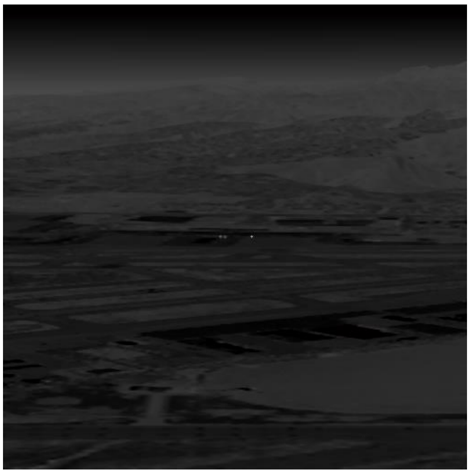
Algorithms are applied to the first scenario that is assumed to be the easiest scenario. The best values of user-dependent parameters are determined in this section. All scenarios are used to compare the algorithms in section 3.5.

3.4.1 Morphological Operation Based Algorithms

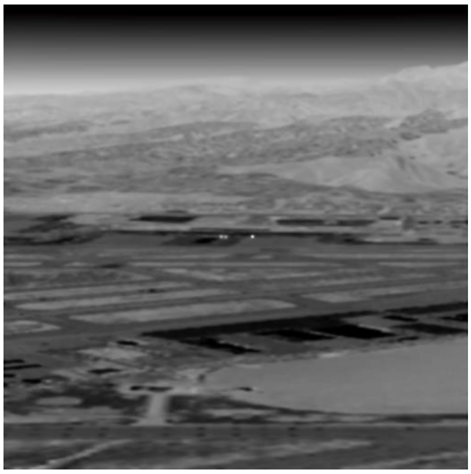
Six morphological operation based algorithms are detailed in Chapter 2. These algorithms are applied to the 20th frame of the first scenario as shown in Figure 3-7.

Figure 3-7: The test image

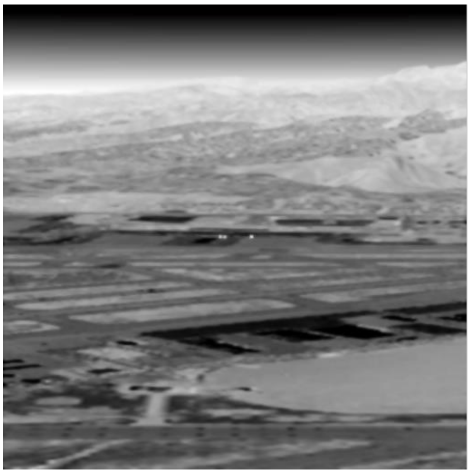
All the outputs obtained from the each step of MO1 are given in Figure 3-8.The minimum and maximum values are represented with white-black instead of black-white for Figure 3-8 (f), (g) and (h). In this application, the median filter size is selected as 3x3 because its exact value is not given in the reference [4]. In addition, the size of the small and big structuring element (SE) is not included in the paper. Only the square shape of the structuring element is given [4]. In addition, the sizes of the small and large structuring element are defined to be smaller than the target and greater than the target respectively. Using this information, the small structuring element is selected as a 2x2 square and the large structuring element is selected as a 5x5 square.



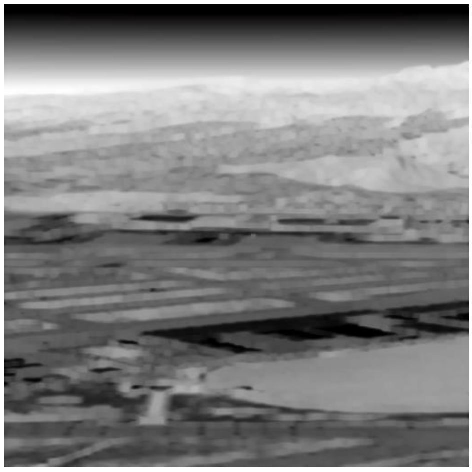
(a)



(b)



(c)



(d)

(e)

(f)

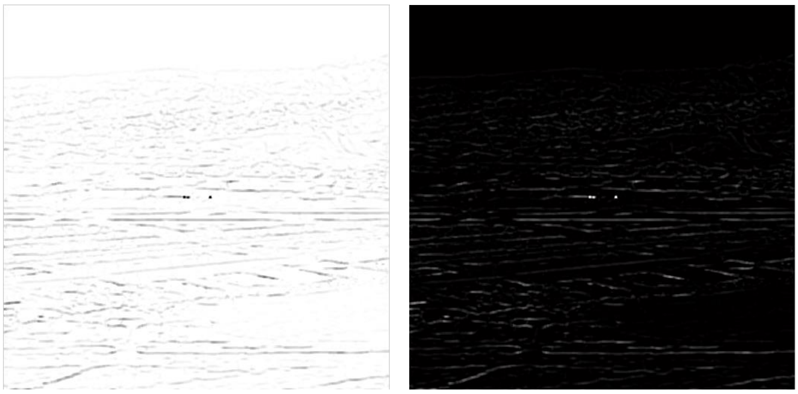
Figure 3-8: The outputs of the MO1.

(g) (h)

Figure 3-8 (Continued)

a) Median filtered, b) H filtered, c) Opening with small SE, d) Opening with big SE, e) Difference between (c) and (d), f) Offset processing of (e), g) Thresholding result of (f), h) Thresholding result of (g).

When the detection results given in Figure 3-8 are analyzed, it is seen that the target cannot be detected (Figure 3-8 (h)). This problem results from the subtraction order of the opening processes. The result of the opening obtained with small SE is subtracted from the result of the open process obtained with large SE as is indicated in [4]. The results given in Figure 3-9 refer to this procedure being undertaken vice versa. The results obtained from the application and the paper results are matched after changing the subtraction order.

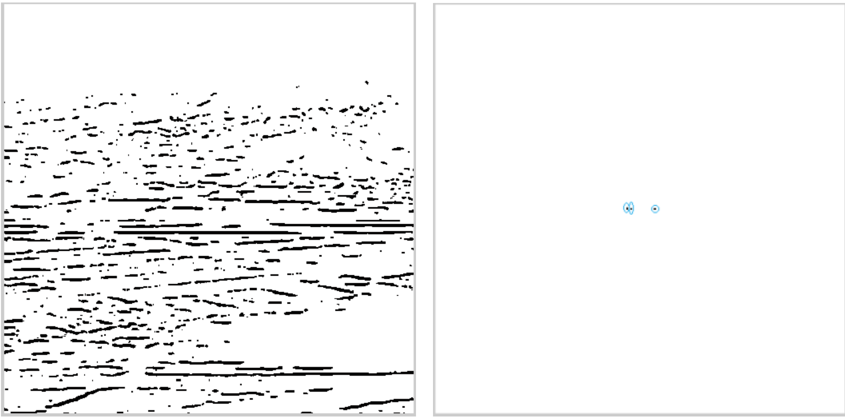


(a) (b)

Figure 3-9: The subtraction results of MO1.

a) Result of the given subtraction order, b) Result of the changed subtraction order

The detection result is obtained by changing the subtraction order of opening processes (Figure 3-10 (a)). However, false alarm rate is quite high. This is assumed to be produced by the offset value. To resolve this situation the offset value is increased. The new detection result obtained with a high offset value is given in Figure 3-10 (b). These detection results color scale is converted from black-white to white-black for minimum-maximum values to increase the visibility. As a result, method is highly dependent on the selected threshold parameter.



(a) (b)

Figure 3-10: Detection results obtained with a different offset value.

a) The offset value is 0. b) The offset value is 660.

In the application of the MO2 algorithm to the test image, first, Top-Hat transform is applied using a different size structural element (Figure 3-11). Minimum and maximum values of the output results are converted from black-white type to white-black type to increase the visibility. A circular structural element shape used in THT is selected [5]. The size of the structural element is changed from 1 to 16 [5].

Figure 3-11: The THT result of MO2 with different SE's.

SE Size 10 SE Size 11 SE Size 12

SE Size 13 SE Size 14 SE Size 15

SE Size 16

Figure 3-11 (Continued)

Considering the THT results obtained from different sizes structural elements, it is observed that the amount of suppressed background decreases with the increasing structural element size. In the next step, the entropy of the THT results is calculated (Figure 3-12). As shown from the graph, the smallest entropy value is obtained with the SE of size 1.

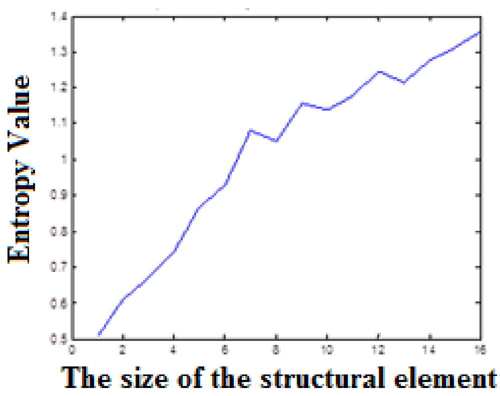


Figure 3-12: The entropy results of the THT results

Finally, the Otsu thresholding method is applied to the THT result which is obtained with 1 size SE (Figure 3-13). The target detection result includes too much noise. This problem is assumed to be a result of the Otsu thresholding method since it only works very well with a bimodal histogram image. If the image does not have bimodal histogram than the method cannot correctly separate target and background pixels.

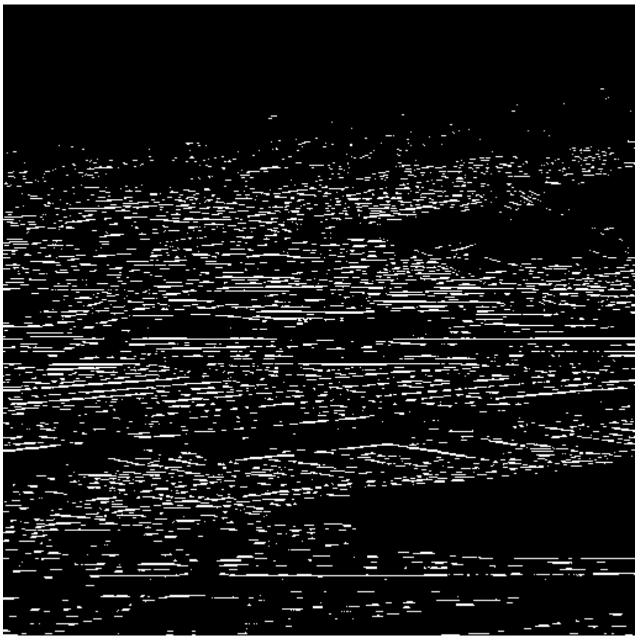


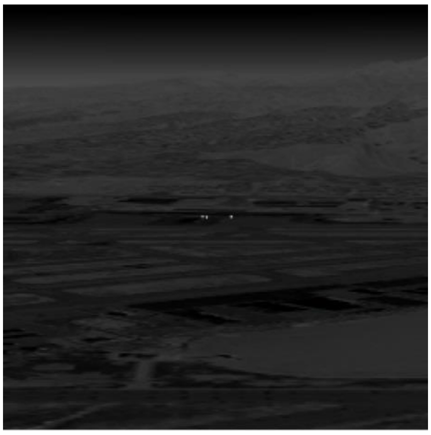
Figure 3-13: The target detection result of MO2

The outputs of each step of MO3 are given in Figure 3-14. Minimum and maximum values of the output results are converted from black-white type to white-black type to increase the visibility. Firstly, white and black THT's were applied to the image. However, no information is included related to the SE in the paper [6]. To apply the algorithms objectively, the SE is selected as a 3x3 square. The MO3 thresholding method has user-defined parameters that are T_0 as 19, c as 0.1 and ε as 0.01.

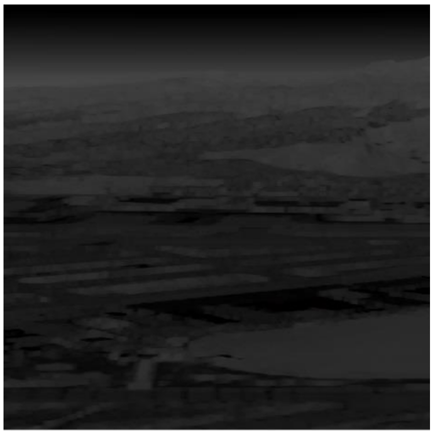
Figure 3-14: The outputs of MO3 in each step.

a) White THT result, b) Black THT result, c) Fusion of (a) and (b), d) Detection result

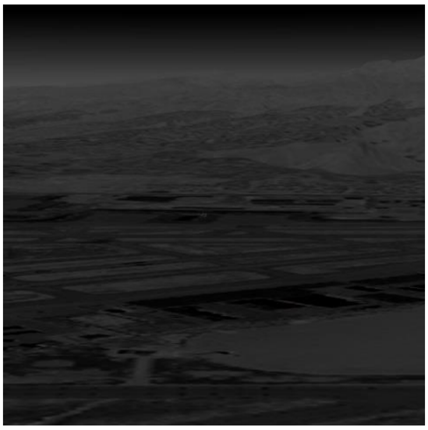
The outputs obtained in each step of MO4 are given in Figure 3-15. Minimum and maximum values of Figure 3-15 (b), (c), (f) and (g) are converted from black-white type to white-black type to increase the visibility. The opening is performed using 1x5 horizontal and 5x1 vertical SE's [7]. The size of the mask is 8x8.




(a) (b)



(c) (d)



(e) (f)

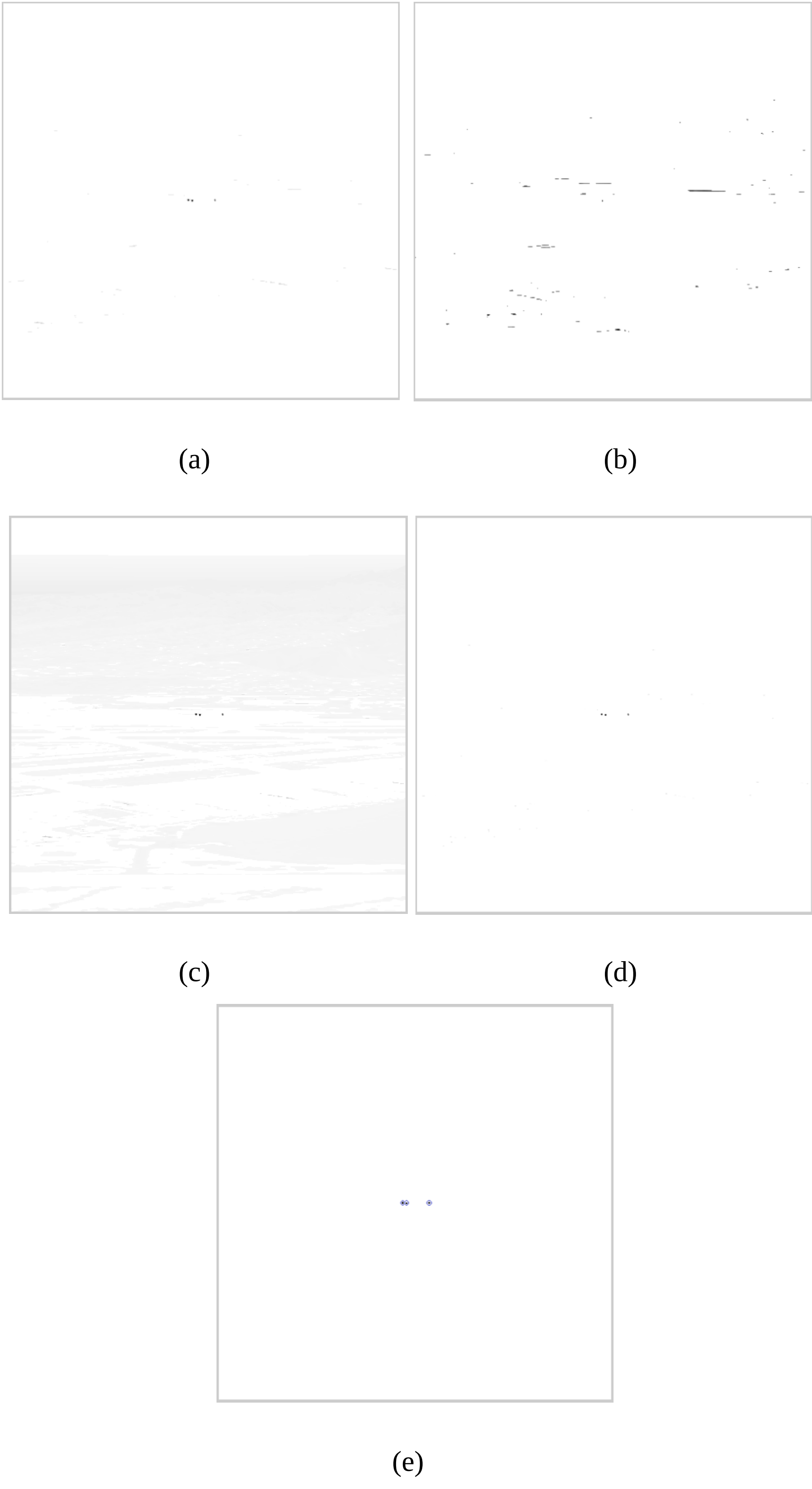
Figure 3-15: The outputs obtained in each step of MO4.

(g)

Figure 3-15 (Continued)

a) Background obtained with masking, b) Subtraction of (a) from original image c) Thresholding (b) with α, d) Opening process result, e) Fusion of (c) and (d), f) Subtraction of (e) from original image, g) Last thresholding result (targets are encircled).

The outputs of MO5 steps are given in Figure 3-16. Minimum and maximum values of output results are converted from black-white type to white-black type to increase the visibility. No information related with the SE shape and size is given in [8]. For this reason, the SE is selected as a 3x3 square. This kind of structural element selection allows a comparison to be made more fairly. There are some user-dependent parameters in target enhancement operation and they are not defined in [8]. For this thesis the parameters are α as 0.2, β as 0.8 and γ as 0.6. The parameter t used in the improved WTHT and the BTHT are taken as 20. Threshold constant c the last user-dependent parameter of MO5 is taken as 1. These user-dependent parameters are selected as to produce the best detection results.



(a) (b)

(c) (d)

(e)

Figure 3-16: The outputs obtained in each step of MO5.

a) White THT result, b) Black THT result, c) TEO result, d) RGF filtering result, e)Thresholding result.

The application results of MO6 are given in Figure 3-17. Minimum and maximum values of output results are converted from black-white type to white-black type to increase the visibility. For this application, the grid structrural element is combined with small and large SE since there is no information about this order in [9]. The results are investigated and it is found that the grid and large flat SE combination suppresses the background better than the grid and small flat SE combination.


(a) (b)

(c)

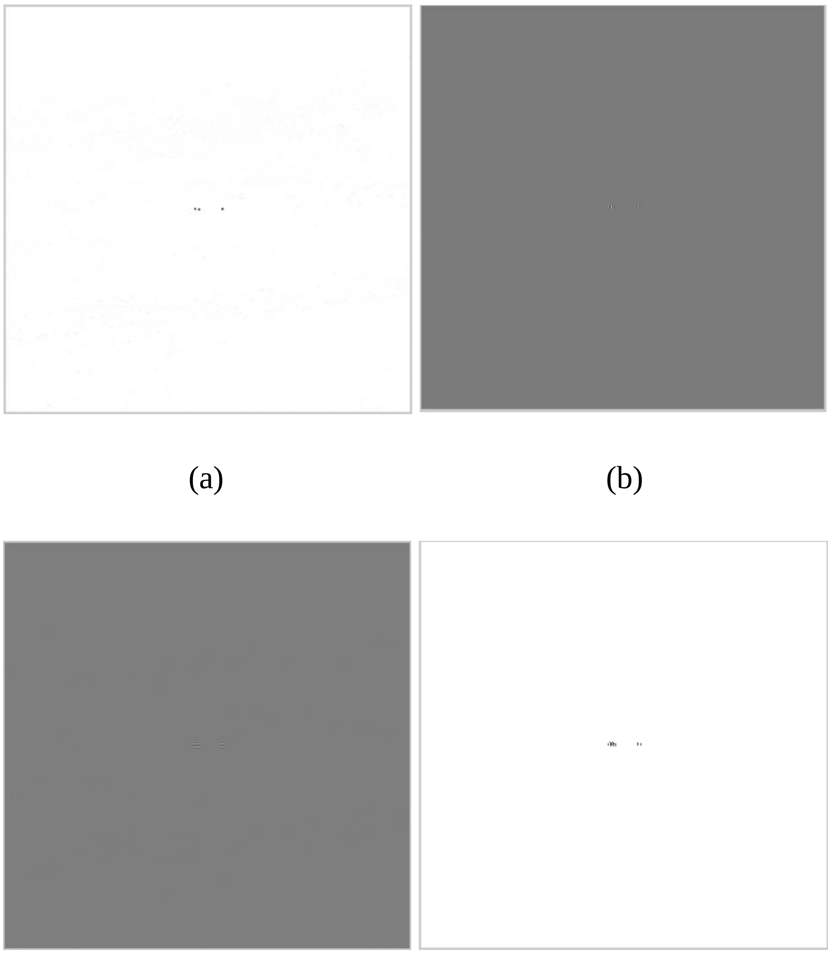
Figure 3-17: The application results of MO6.

a) Grid and small flat SE THT result, b) Grid and big flat SE THT result, c) Thresholding result of MO6.

3.4.2 Wavelet Transform Based Algorithms

Five target detection algorithms are presented within the scope of WT in Chapter 2 [10]-[14]. These algorithms are applied to the test image (Figure 3-7) to obtain the outputs of each step and determine the user-dependent parameters.

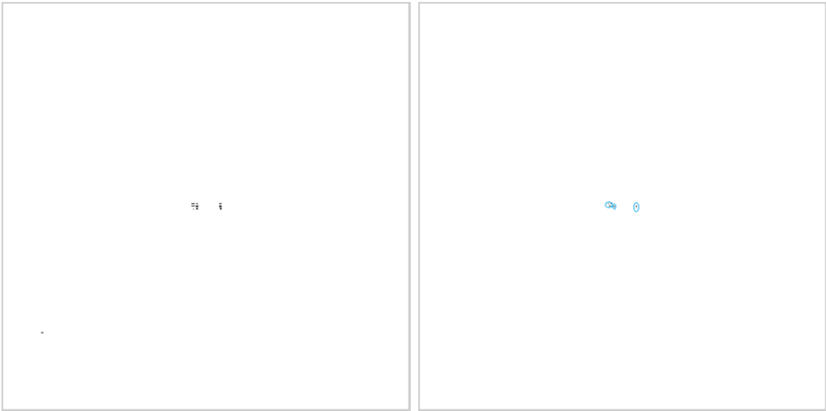
All outputs obtained from the application of WT1 are given in Figure 3-18. Minimum and maximum values of output results are converted from black-white type to white-black type to increase the visibility. Firstly, THT is applied to the image to suppress the background. The SE is 1x6 horizontal lines in the THT process [10]. After THT, the first scale WT is applied to the image using a "sym4" filter [10]. Then horizontal and vertical components are extracted from the WT. These components are thresholded using the *k* constant as 15. This parameter value is not given in [10].



(a) (b)

(c) (d)

Figure 3-18: The application results of the WT1 algorithm.

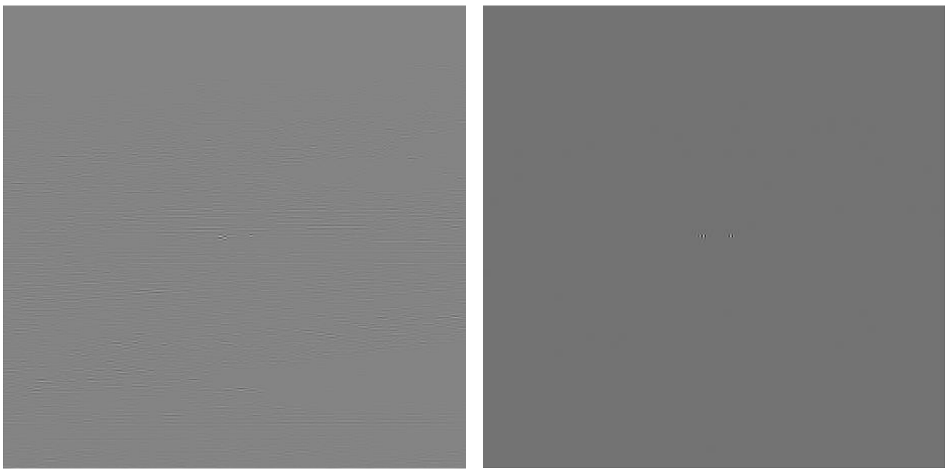


(e) (f)

Figure 3-18 (Continued)

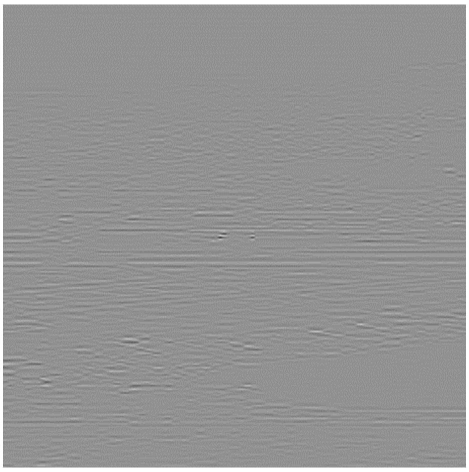
a) THT result of the original image, b) Vertical component WT (a), c) Horizontal component of (a), d) Thresholding of (b), e) Thresholding of (c), f) The fusion result of (d) and (e).

All the outputs of WT2 are given in Figure 3-19. Minimum and maximum values of Figure 3-19 (e), (f), (g), (h) and (i) results are converted from black-white type to white-black type to increase the visibility. A two scale WT is applied to the image [11]. For the WT, the 'db3' wavelet family is selected as a filter [11]. The threshold constant *k* of WT2 is not included in [11]. In the current application this parameter is selected as 2. The threshold constant *k* is selected as between 1 and 3.

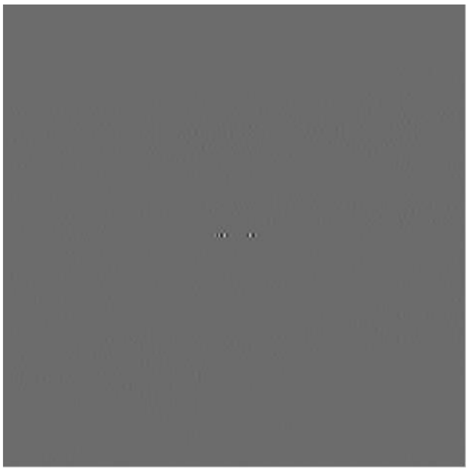


(a) (b)

Figure 3-19: The outputs of WT2 in each step.



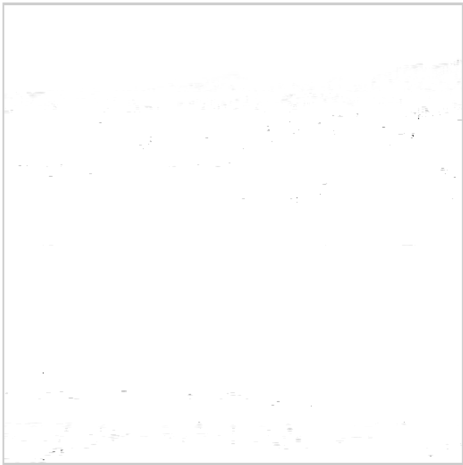
(c)



(d)

(e)

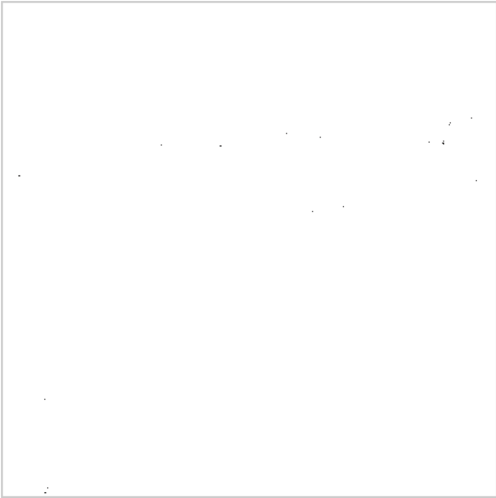
(f)



(g)

(h)

Figure 3-19 (Continued)



(i)

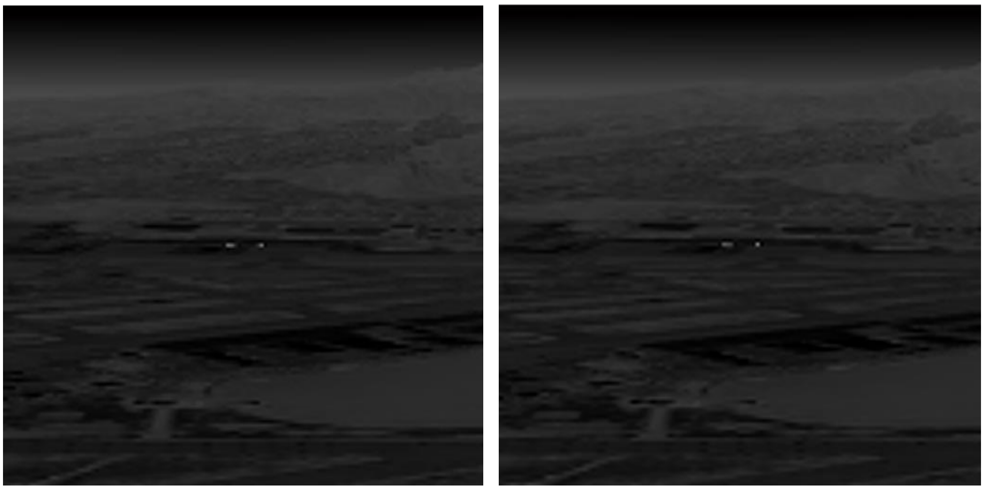
Figure 3-19 (Continued)

a) Horizontal component of the first scale WT, b) Vertical component of the first scale WT, c) Horizontal component of second scale WT, d) Vertical component of the second scale WT, e) Fusion of (a) and (b), f) Fusion of (c) and (d), g) Coefficient protection process of (e), h) Coefficient protection process of (f), i) Fusion of (g) and (h).

The analysis of the detection result of WT2 indicates that no target could be detected. This problem is assumed to result from the thresholding method. In the description of thresholding method of WT2, it is stated that the mean and variance values are used. However, in the current application σ^2 is used in the formulation instead of σ. This symbol (σ) is generally used to define the standard deviation. One this basis, it is assumed that the variance expression is misused in [11]. The thresholding step of WT2 is repeated by calculating threshold value with the standard deviation (Figure 3-20). Minimum and maximum values of output results are converted from black-white type to white-black type to increase the visibility. This result is assumed to be better than the previous detection result.

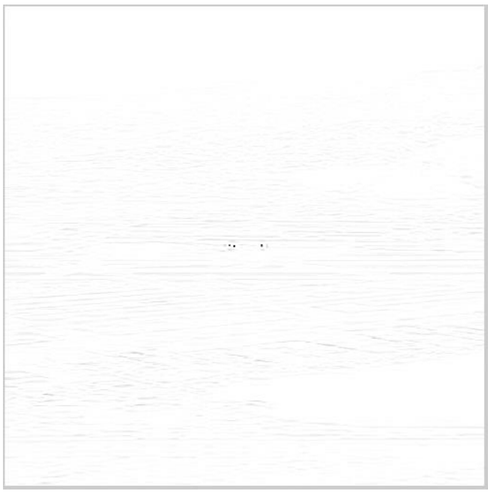
Figure 3-20: Detection result of the corrected thresholding method.

The results of the application of WT3 are given in Figure 3-21. Minimum and maximum values of Figure 3-21 (c), (d) and (e) results are converted from black-white type to white-black type to increase the visibility. In the detection stage of WT3, it is given that the two scales WT is applied to the image [12]. However, the wavelet family is not included. To determine the correct type of WT filter, two frequently used WT filters, db3 and sym4, are applied to the image. However, there is no great difference between results of using these two filters so db3 is selected as WT filter in this application. Otsu thresholding method is used to detect targets in the image [12]. There is no user-dependent parameter in the thresholding method.

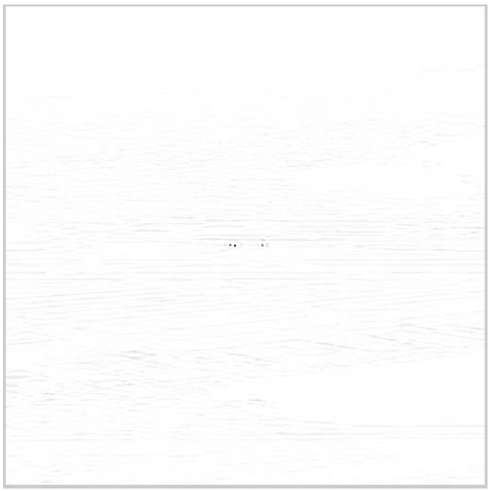


(a) (b)

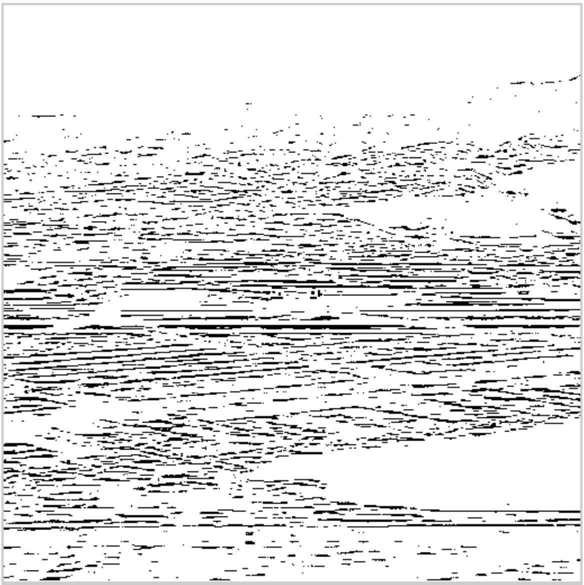
Figure 3-21: The results of the application of WT3.



(c)



(d)

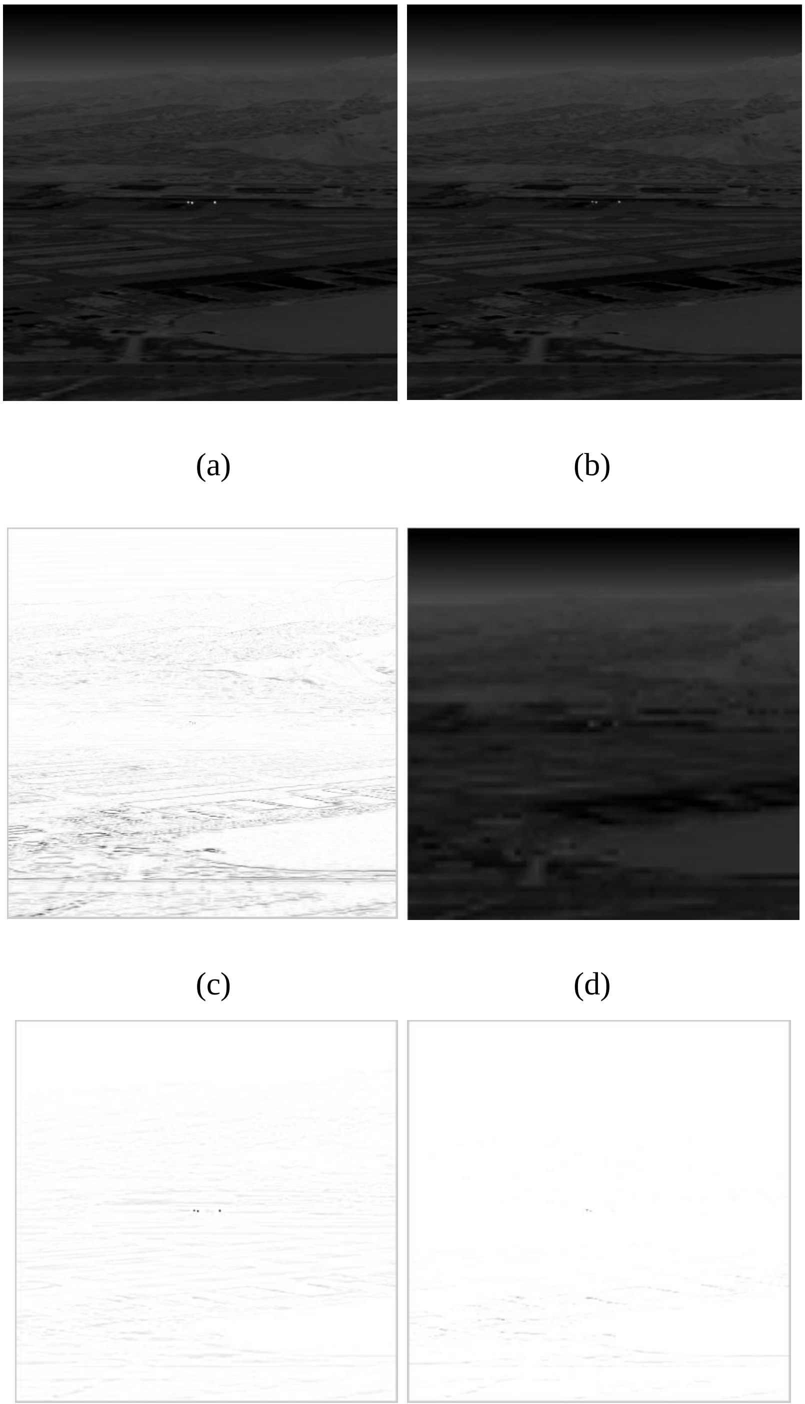


(e)

Figure 3-21 (Continued)

a) Approximation component of WT obtained with the db3 filter, b) Approximation component of WT obtained with "sym4" filter, c) Subtraction result of (a) from original image, d) Subtraction result of (b) from the original image, e) Thresholding result

The outputs of WT4 obtained in each step are given in Figure 3-22. Minimum and maximum values of Figure 3-22 (c), (e), (f) and (g) results are converted from black-white type to white-black type to increase the visibility. The step number used in the frame difference is defined as 14 in [13]. To obtain the estimated background a third scale WT is used [13]. The db2 WT filter is selected [13]. This algorithm does not have thresholding step. To compare all algorithms equally, the thresholding method is added to WT4. The threshold value is calculated by multiplying the maximum value with a constant. Then the pixels higher than the threshold are assigned as one and others were assigned as zero. The threshold constant is 0.35 in this application.

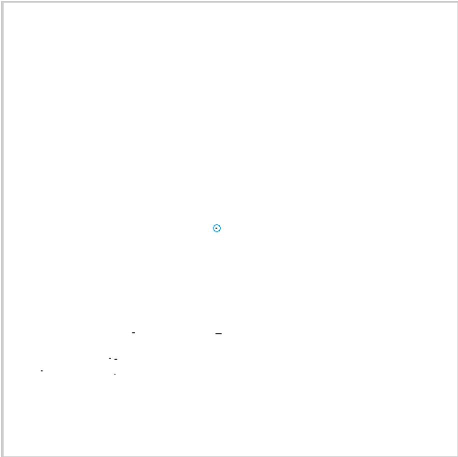


(a) (b)

(c) (d)

(e) (f)

Figure 3-22: The results of the application of WT4.

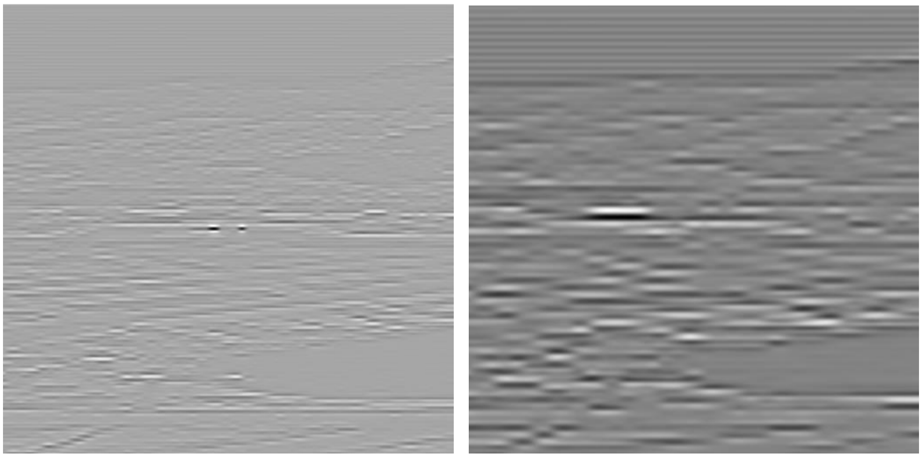


(g)

Figure 3-22 (Continued)

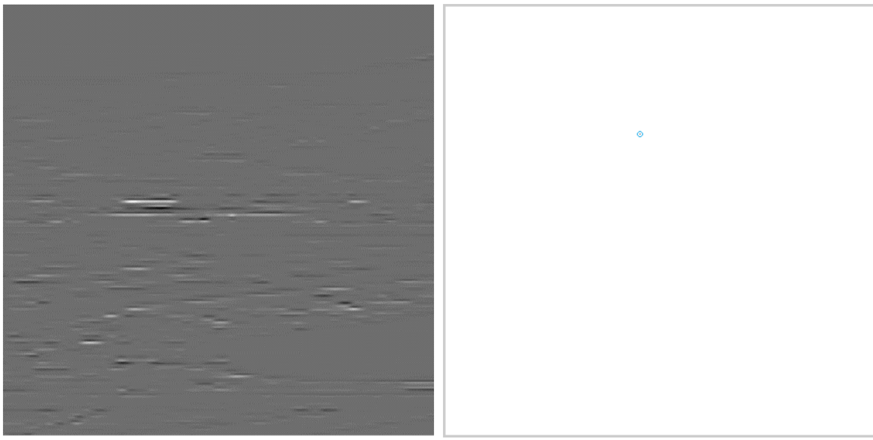
a) 20th frame of the first scenario, b) 34th frame of the first scenario, c) Difference between (a) and (b), d) Approximation component of WT, e) Difference between (d) from (b), f) Fusion of (c) and (e), g) Thresholding result.

The results of the application of WT5 are given in Figure 3-23. Minimum and maximum values of Figure 3-23 (d) result are converted from black-white type to white-black type to increase the visibility. Second and third scale WT's are applied to the image using specified filters [14]. The fusion process used in the algorithm is also given in [14] (Equation (3.26)). In the detection stage, the large window size is 16 and small window size is 3.



(a) (b)

Figure 3-23: The application results of WT5.



(c) (d)

Figure 3-23 (Continued)

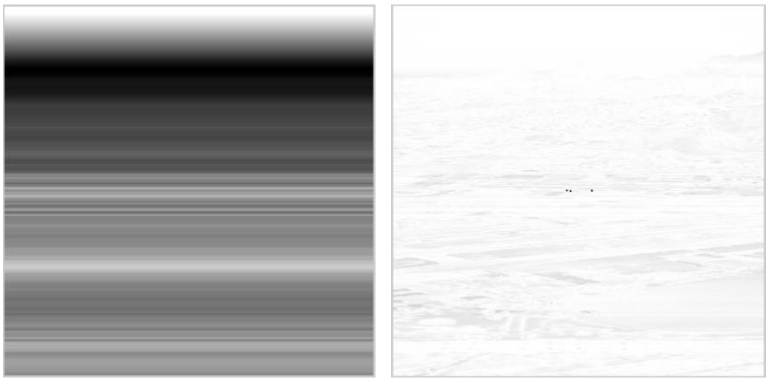
a) Horizontal components of second scale WT, b) Horizontal components of third scale WT, c) Fusion of (a) and (b), d) Detection result.

The analysis of the detection result showed that the target cannot be detected. The single user-dependent parameter in WT5 is the window size of the thresholding method. However, the input of the thresholding method does not enhance the target location. It is assumed that this problem is related to the WT process because WT cannot provide a sufficient enhanced image for the thresholding method.

3.4.3 Edge Detection Based Algorithms

The two ED based algorithms detailed in Chapter 2 [15][16] are applied to the test image (Figure 3-7). User-dependent parameters are determined. The outputs of each step are given in this section. The output results of ED based algorithms color scales are converted from black-white to white-black for minimum-maximum values to improve the visibility.

The results of the application of ED1 pre-processing stage are given in Figure 3-24. In the pre-processing stage of ED1, first the background is estimated and then subtracted from the original image. The user-dependent threshold constant α is selected as 10 in this application.

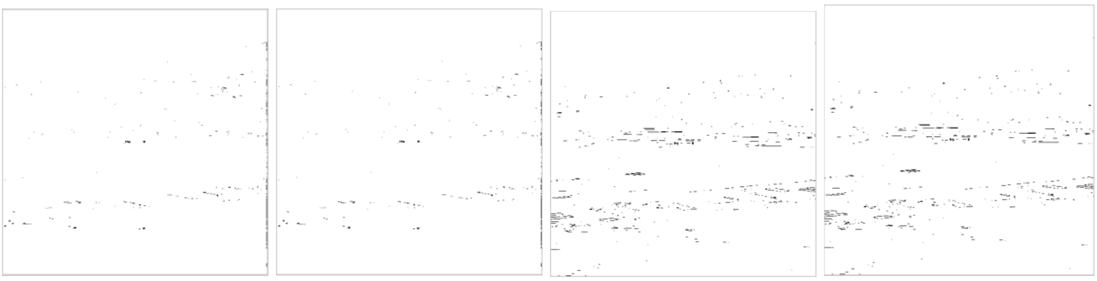


(a) (b)

Figure 3-24: Preprocessing stage results of ED1.

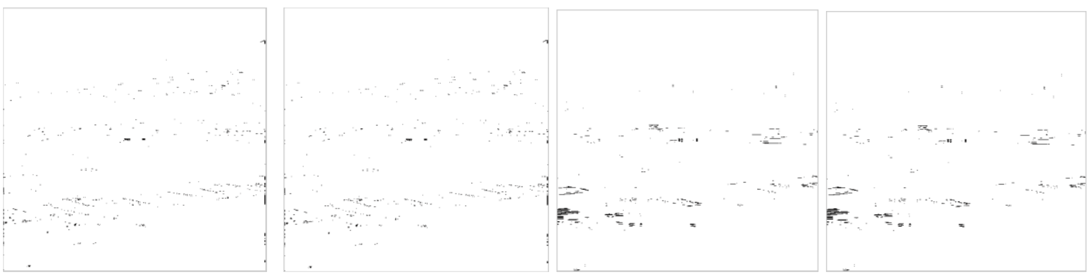
a) Estimated background, b) Subtraction (a) from the original image

Then, in each scale horizontal and vertical positive and negative gradients are obtained using the target image (Figure 3-24 (b)). These gradient results are given in Figure 3-25.



(Hor. Positive) (Hor. Negative) (Ver. Positive) (Ver. Negative)

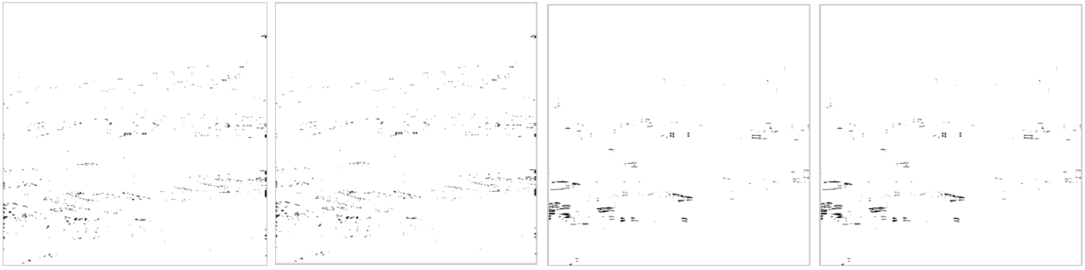
(a)



(Hor. Positive) (Hor. Negative) (Ver. Positive) (Ver. Negative)

(b)

Figure 3-25: The gradient results of ED1.



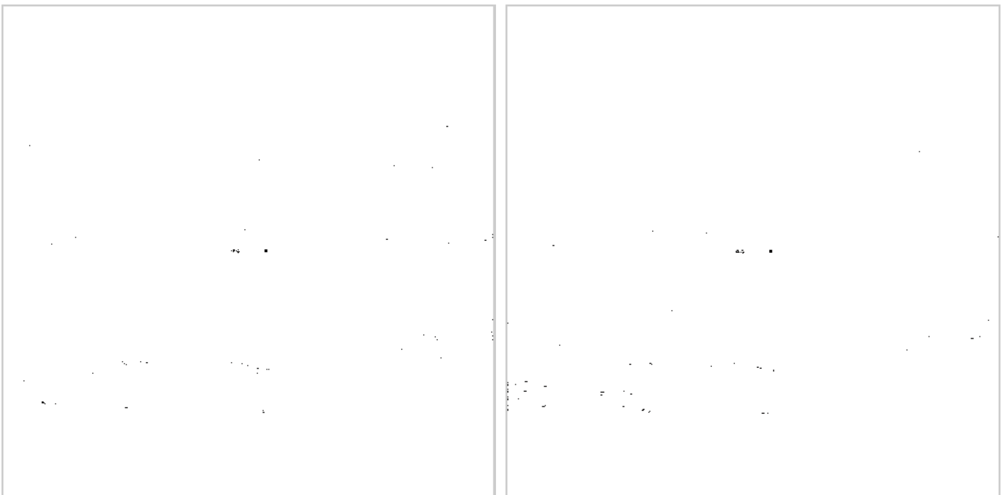
(Hor. Positive) (Hor. Negative) (Ver. Positive) (Ver. Negative)

(c)

Figure 3-25 (Continued)

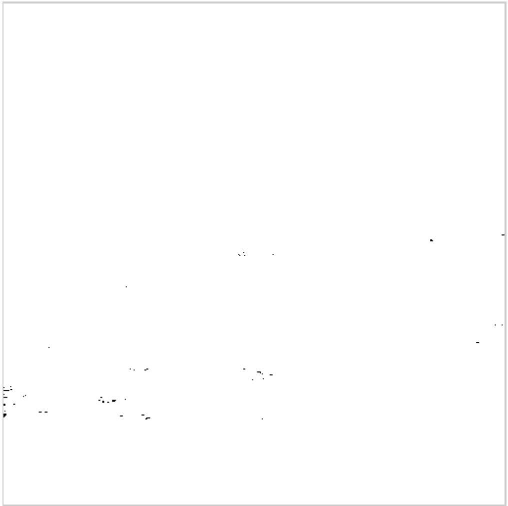
a) First scale gradient results, a) Second scale gradient results, a) Third scale gradient results.

Finally, the gradient results are added to each other pixel by pixel. If the corresponding pixel value is higher than three, then that pixel is assigned to one. The obtained results represent the gradient of each scale. Then, all the scale gradient results are added together pixel by pixel. If the final gradient value is higher than two then it is assumed to be a target.

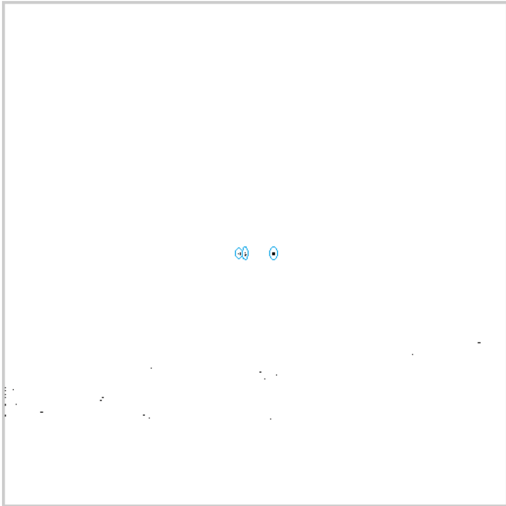


(a) (b)

Figure 3-26: The gradient fusion results of ED1.



(c)



(d)

Figure 3-26 (Continued)

a) Result of the first scale gradient fusion, b) Result of the second scale gradient fusion, c) Result of third scale gradient fusion, d) Detection result (targets are encircled).

The results of the application s of ED2 are given in Figure 3-27. First, the target is enhanced by removing the noise in image. Then, the horizontal and vertical gradients are obtained with discrete cosine transform. These gradient results are smoothed with a gaussian filter. However, no detail is given in the literature related to the smoothing filter [16]. For this reason, a 2x2 size gaussian filter (σ^2=2) is used to smooth the gradient results. Another user-dependent parameter is the threshold constant α however, no information is given related to this parameter [16] so the value is selected as 20 which gives the best detection result. In the post-processing stage, a 2x2 square SE is used to eliminate false targets. This parameter is also not included in [16]. However, the minimum target size is 3x3 in the scenario and this parameter is determined using the minimum target size information.

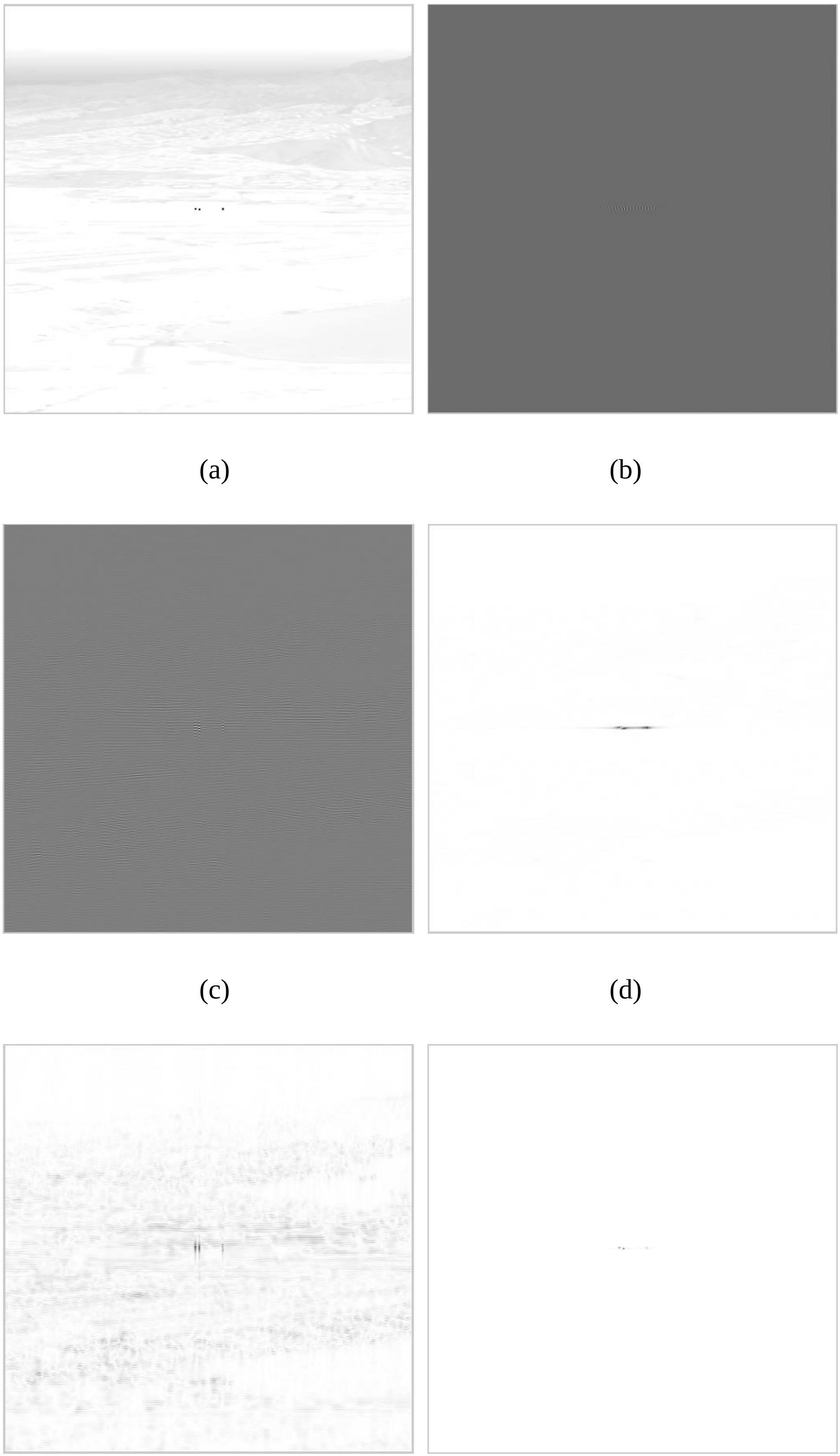


Figure 3-27: The application results of ED2.

(g) (h)

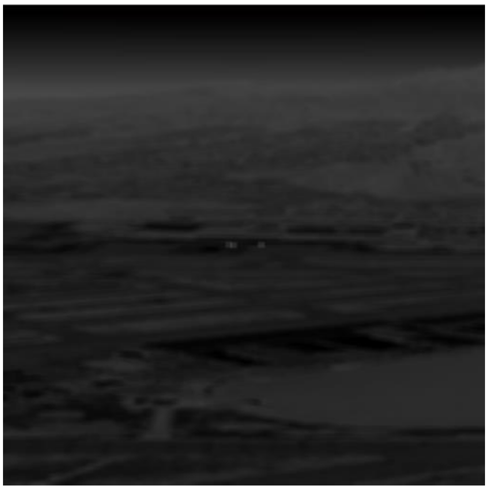
Figure 3-27 (Continued)

a) Results of target enhancement result, b) Horizontal gradient result, c) Vertical gradient result, d) Smoothed (b), e) Smoothed (c), f) The fusion of (d) and (e), g) Thresholding result, h) Detection result (targets are encircled).

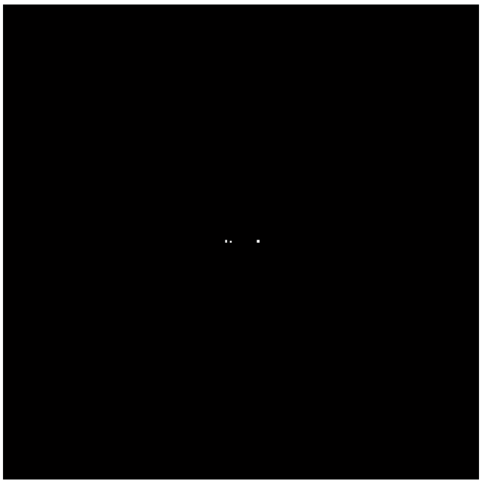
3.4.4 Filtering Based Algorithms

Two algorithms are given under filtering based group in Chapter 2 [17][18]. These algorithms are applied to the test image (Figure 3-7). All the outputs from the steps are obtained to analyze the effect of the user-dependent parameter on the detection performance.

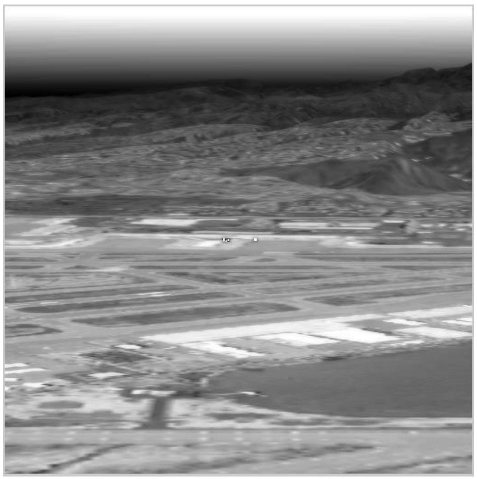
Figure 3-28 presents the application results of Fi1. The color scale of Figure 3-28 (b), (e), (f), (g), (h) and (i) results is converted from black-white to white-black for minimum and maximum values to increase the visibility. First, the background is estimated using 7x7 mean filter. The threshold value η that separate noise and the target pixel is selected as 50 in this application. This value is the most suitable value for the separation of noise and the target pixel. To estimate the new background the weighting constants k_1 and k_2 are 0.7 and 0.3 respectively.

(a) (b)

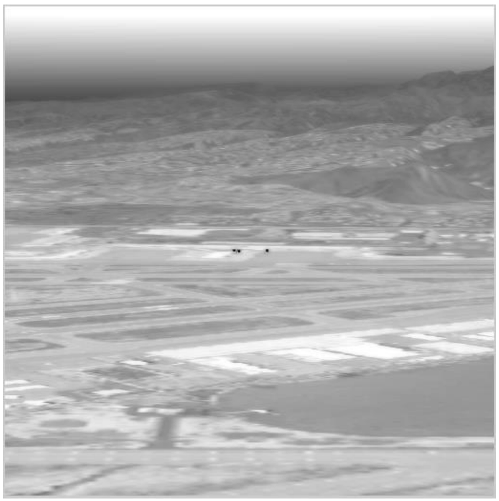
 

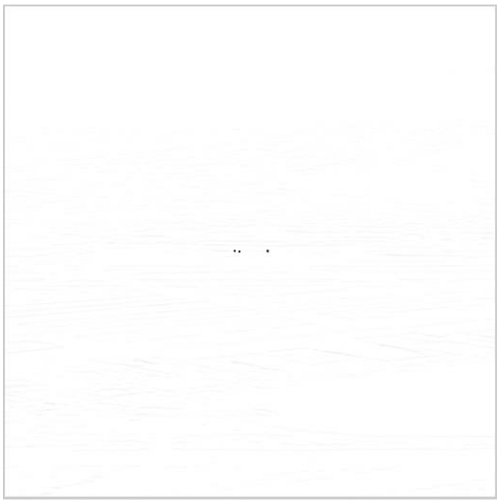
(c) (d)



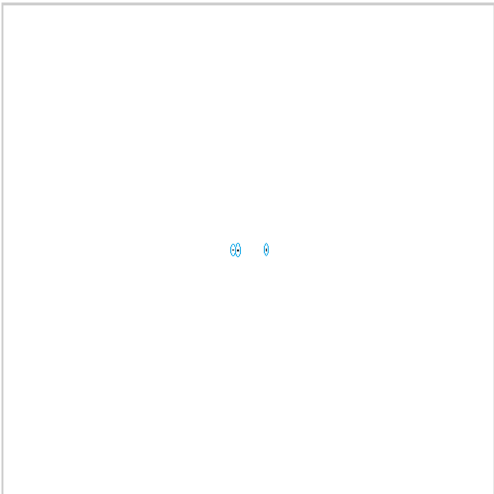
(e) (f)

Figure 3-28: The application results of Fi1.





(g) (h)

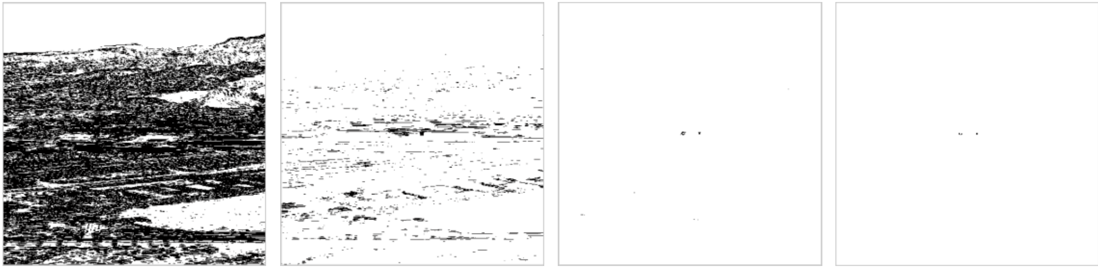


(i)

Figure 3-28 (Continued)

a) Background estimation result, b) The subtraction of (a) from original image, c) Separated noise pixels indicated with white, d) Separated target pixels indicated with white pixels, e) Noise image, f) Signal image, g) The estimated background using (e) and (f), h) The subtraction of (g) from original image, i) Thresholding result.

To analyze the different separation threshold (η) performance, different values are selected to obtain the target pixels (Figure 3-29). Maximum and minimum values are represented with black and white. It can be seen in Figure 3-29, that it is very important to select η correctly since it directly affects performance by separating noise and the target pixels.



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

Figure 3-29: The target pixels obtained with different η values.

a) η=1, b) η=10, c) η=30, d) η=50

Another evaluation is undertaken to analyze the effect of weighting coefficients (Figure 3-30). This output results are represented with the same color scale with Figure 3-29. Figure 3-30 shows that k_1 should be selected close to one and k_2 should be selected as close to zero.

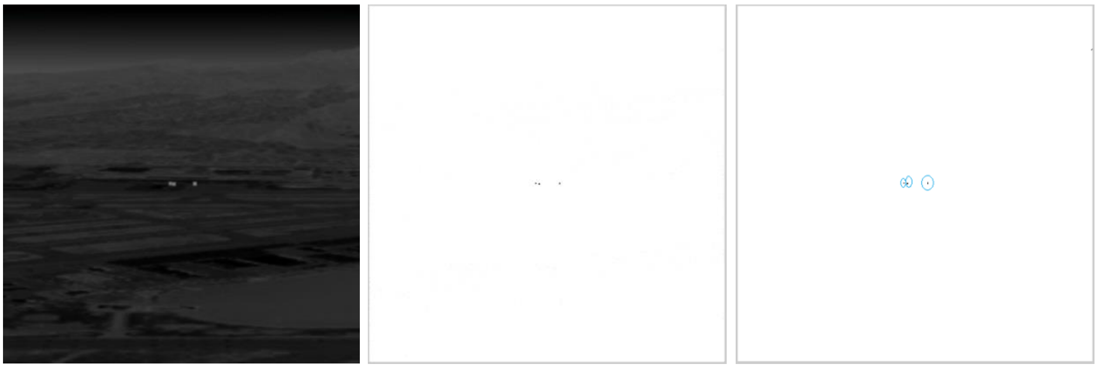
(a) (b) (c) (d)

Figure 3-30: The target images obtained with different weighting coefficients.

a) k_1=0.1, k_2 = 0.9 b) k_1=0.3, k_2 = 0.7, c) k_1=0.7, k_2 = 0.3, d) k_1=0.9, k_2 = 0.1

The Fi1 algorithm does not have a thresholding step. To compare all algorithms equally, a thresholding method is added to Fi1. This method calculates the threshold value by multiplying the maximum value with a constant.

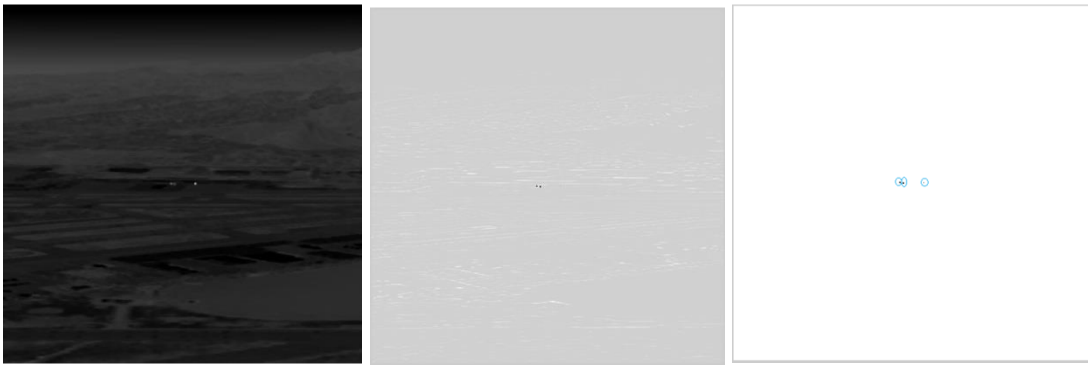
Figure 3-31 and Figure 3-32 give the application results of Fi2 using the max-mean and max-median filter respectively. Color scale of Figure 3-31 and Figure 3-32 (b) and (c) are inverted to black-white for maximum and minimum values. In Fi2, the threshold constant is user-dependent. This parameter is 25 in the max-mean filter version of Fi2 and 6 in the max-median filter version of Fi2. Another UDP is used in the thresholding method. This parameter defines the window size that is selected as 64 in this application.



(a) (b) (c)

Figure 3-31: The application results of Fi2_1.

a) Background estimation result of max-mean filter, b) Subtraction of (a) from original image, c) Thresholding result (targets are encircled).



(a) (b) (c)

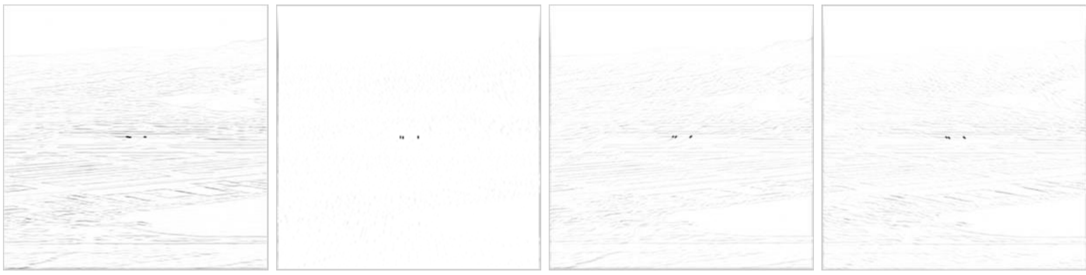
Figure 3-32: The application results of Fi2_2.

a) Background estimation result of max-median filter, b) Subtraction of (a) from the original image, c) Thresholding result (targets are encircled).

3.4.5 Saliency Based Algorithms

In Chapter 2 four saliency based algorithms are investigated [19]-[22]. These algorithms are applied to the test image (Figure 3-7) and the outputs were obtained from each step. Minimum-maximum value of output results color scale is converted from black-white to white black to increase the visibility. To present the effectiveness of user-dependent parameters, results are obtained using different parameter values.

The application results of S1 are given in Figure 3-33. Color scale of the output results are converted from black-white to white-black for minimum-maximum values to increase the visibility. S1 does not have any user-dependent parameters [19]. In the thresholding stage, the threshold value is calculated by multiplying the maximum value with a 0.35 constant [19]. However, the constant value differs in each scenario [19]. Derived from this information the threshold constant is 0.5 in this application. The use of this constant increases the detection result in this application.



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

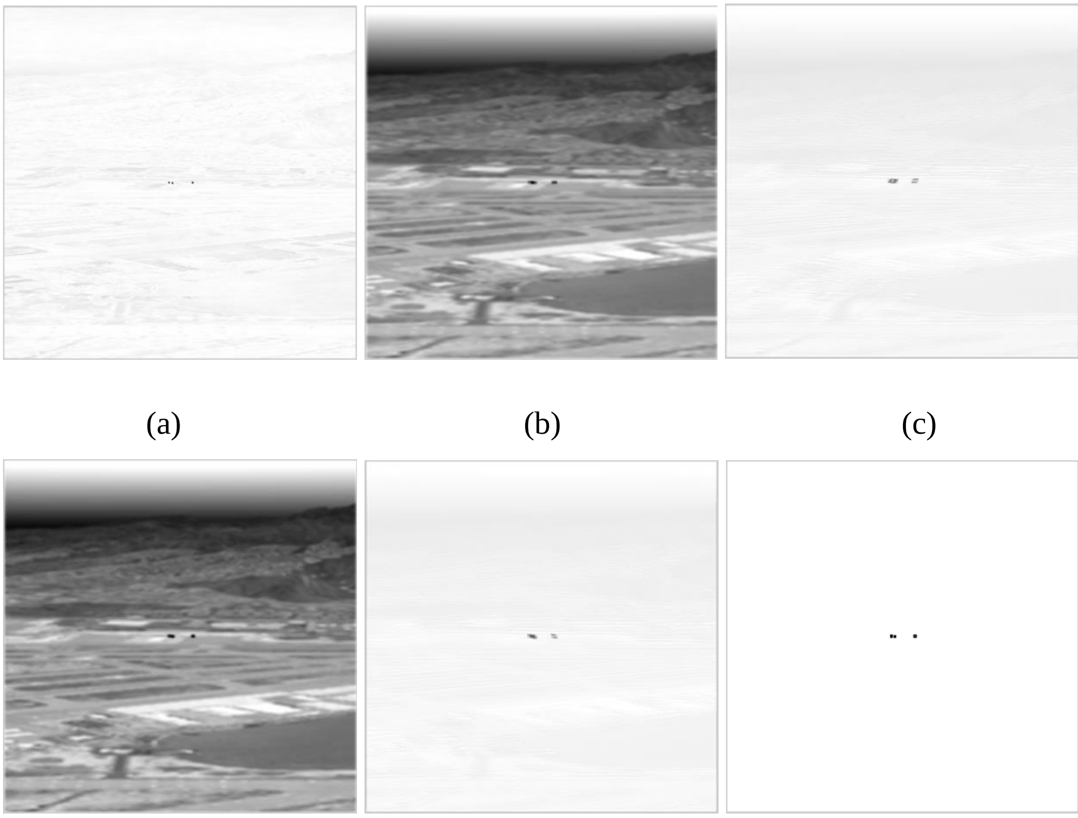
(e) (f) (g) (h)

(i) (j) (k)

Figure 3-33: The application results of S1.

a) First channel SODD, b) Second channel SODD , c) Third channel SODD, d) Fourth channel SODD, e) First channel saliency regions, f) Second channel saliency regions, g) Third channel saliency regions, h) Fourth channel saliency regions, i) Fusion result of (e), (f), (g) and (h), j) Detection result with a 0.35 threshold constant (targets are encircled), k) Detection result with a 0.5 threshold constant (targets are encircled).

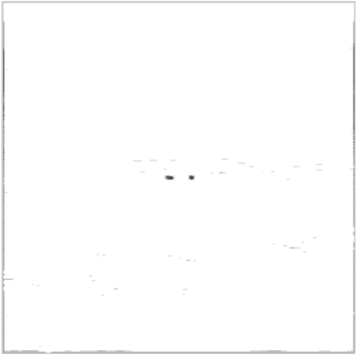
The application results of S2 are given in Figure 3-34. In this algorithm, saliency maps are obtained in both the spatial and frequency domain [20]. S2 also does not have any user-dependent parameters [20]. However, there is no thresholding step in S2. To compare all algorithms equally, the same thresholding method as S1 is added to S2. Threshold constant is selected as 0.4 for this application.



(a) (b) (c)

(d) (e) (f)

Figure 3-34: The application results of S2.



(g)

(h)

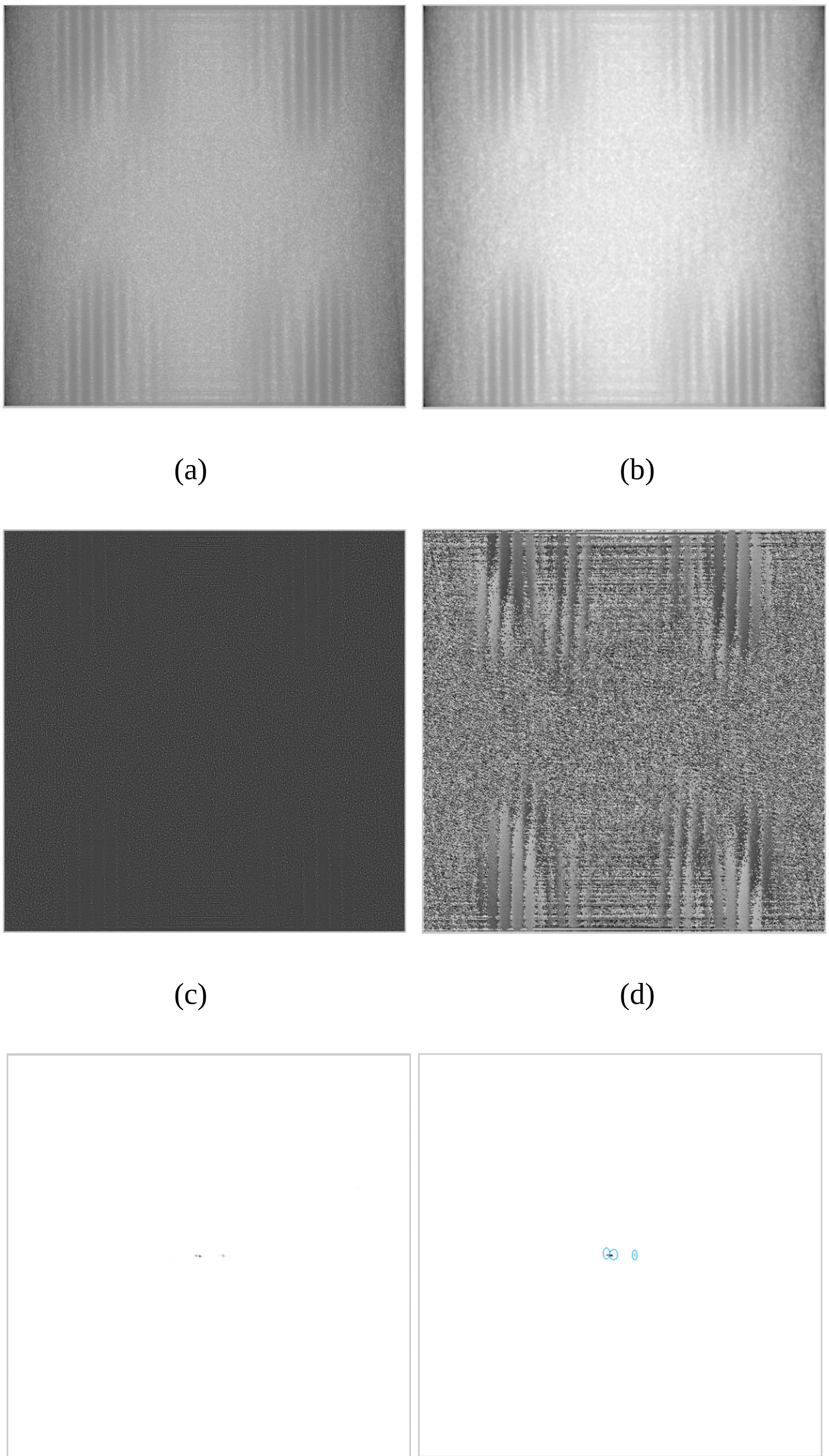
(i)

(j)

Figure 3-34 (Continued)

a) Saliency regions obtained in frequency domain, b) Gabor filter (0^o), c) Gabor filter (45^o), d) Gabor filter (90^o), e) Gabor filter (135^o), f) Canny edge detection, g) Fusion of spatial saliency maps, h) Normalized frequency saliency map, i) Fusion of (g) and (h), j) Detection result (targets are encircled).

The application results of S3 are given in Figure 3-35. There are no user-dependent parameters in S3. The smoothing filter is conducted with as a 3x3 kernel. S3 does not employ a thresholding method. For the same purpose as S2, the same thresholding method used with S2 is added to S3. Threshold constant of this method is selected as 0.3.



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

Figure 3-35: The application results of S3.

a) Log-amplitude spectrum of FT or the original image, b) Smoothed (a), c) difference of (a) and (b), d) Phase spectrum of FT of original image, e) Inverse FT result by using (c) and (d) (saliency map), f) Thresholding result (targets are encircled).

The application results of S4 are given in Figure 3-36. S4 does not have any UDPs. The smoothing process is performed using 3x3 gaussian kernel. S4 also does not have a thresholding method. To compare all algorithms equally, the same thresholding method used in S2 is added to the S3 algorithm. A threshold constant of this method is selected as 0.4.

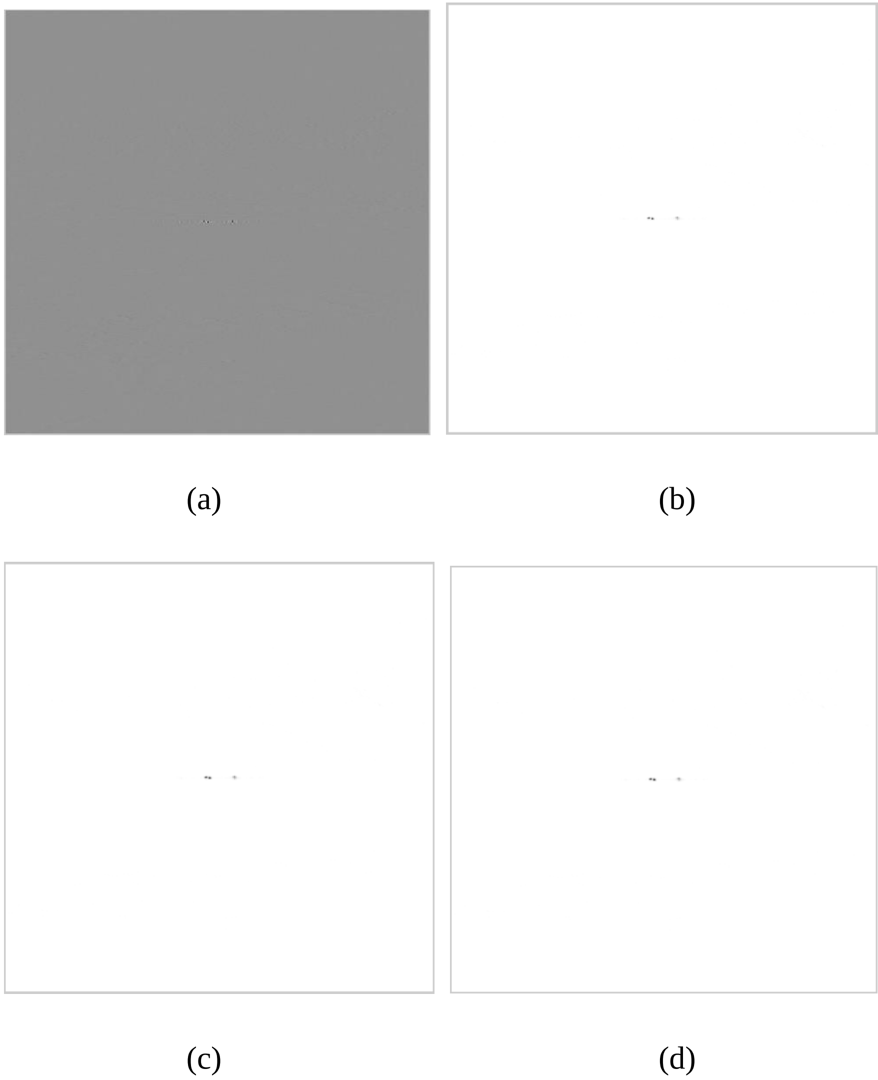


Figure 3-36: The application results of S4.

(e)

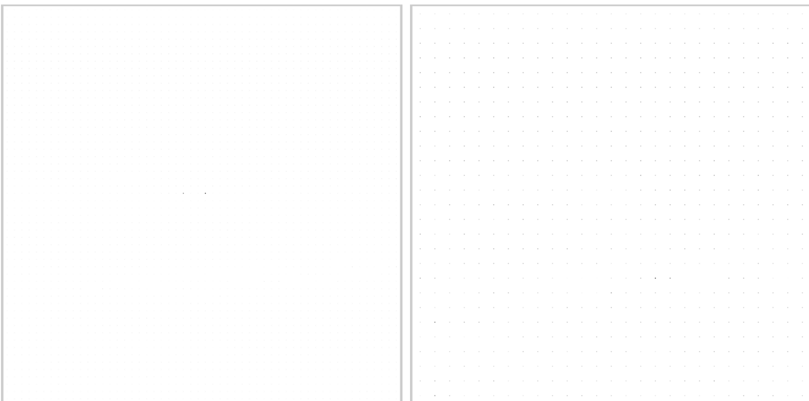
Figure 3-36 (Continued)

a) Inverse DCT of positive coefficients, b) Square of (a), c) Smoothed (b), d) Normalized (c), e) Thresholding result (targets are encircled).

3.4.6 Feature Based Algorithms

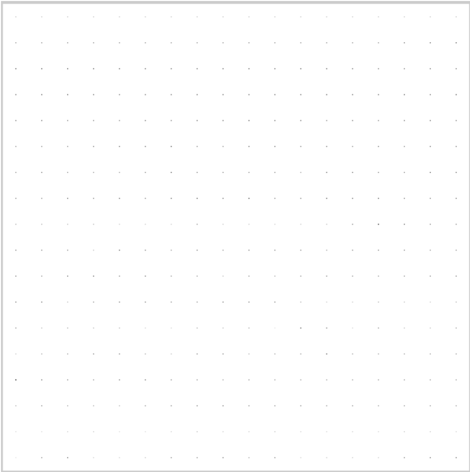
Three algorithms are explained under the feature based group in Chapter 2 [23]-[25]. These algorithms were applied to the test image (Figure 3-7) to analyze the importance of the user-dependent parameters. All the outputs obtained with the applications are given in this section. The minimum and maximum values colors are inverted. Minimum and maximum values are represented white and black colors respectively.

The application results of Fe1 are given in Figure 3-37. Fe1 has just one user-dependent parameter that is related to the thresholding method and sets the threshold value. The threshold constant is used as 5 in this application.



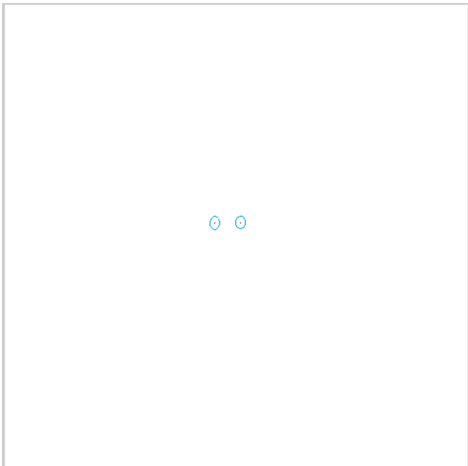
(a) (b)

Figure 3-37: The application results of Fe1.



(c)

(d)

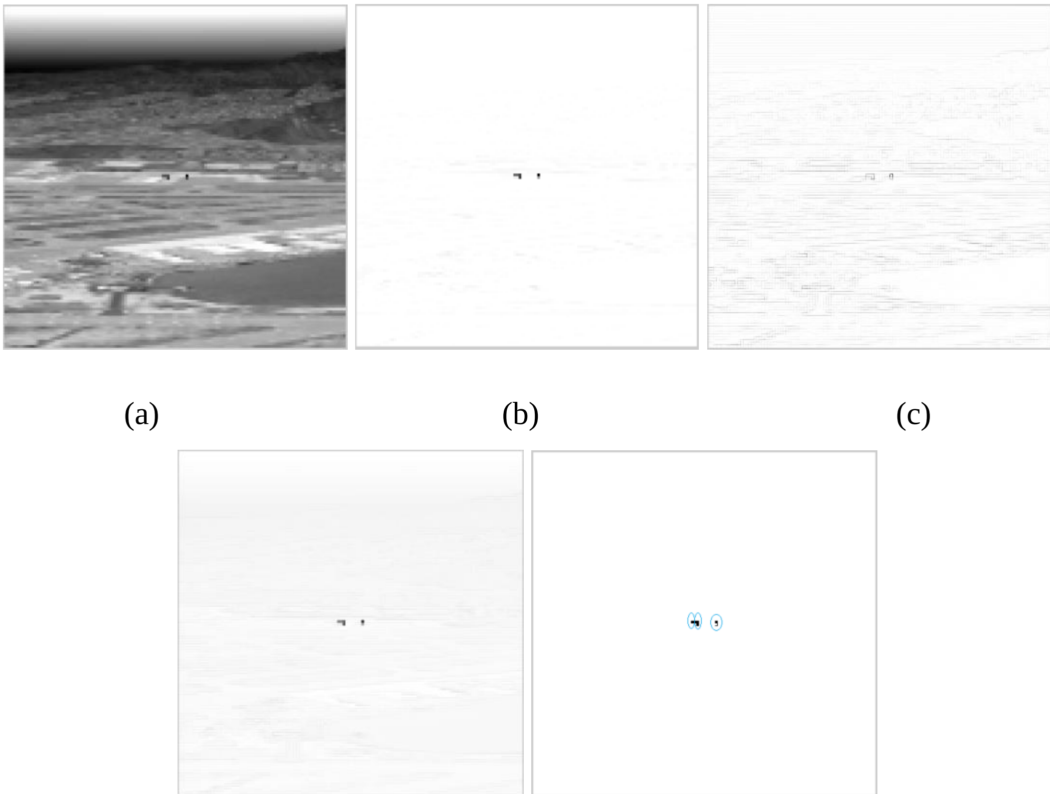


(e)

Figure 3-37 (Continued)

a) First scale LCM, b) Second scale LCM, c) Third scale LCM, d) The fusion of (a), (b) and (c), e) Thresholding result (targets are encircled).

The first application results of Fe2 are given in Figure 3-38. There are two user-dependent parameters in Fe2. First one is the window size that is used to calculate features. This parameter is used as 4x4 because it is highly related to the target size. Another user-dependent parameter is the threshold constant, which is used to calculate threshold value. This parameter is used as 30 in the first application of Fe2.



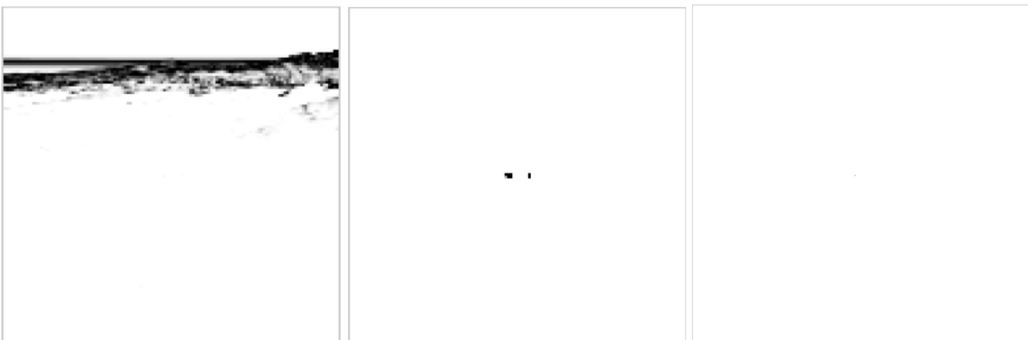
(a) (b) (c)

(d) (e)

Figure 3-38: The application results of Fe2.

a) First feature map, b) Second feature map, c) Third feature map, d) Fusion results of (a), (b) and (c), e) Thresholding result of (d) (targets are encircled).

The second application results of Fe2 are given in Figure 3-39. In this application, likelihood ratios are used to obtain the detection result. In this application, the window size is 4x4 which is same as the first application. However, threshold constant is selected as 15, which differs from the first application.



(a) (b) (c)

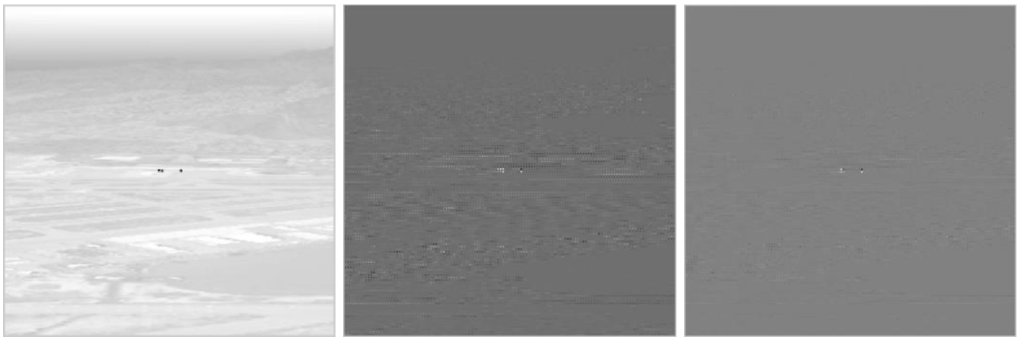
Figure 3-39: The application results of second version Fe2.

(d) (e)

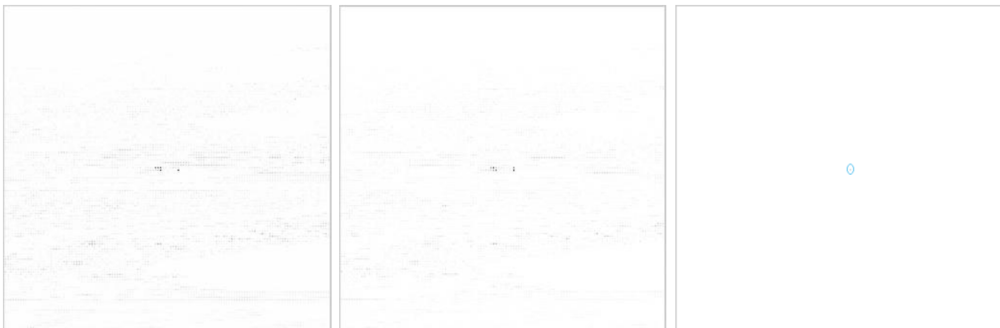
Figure 3-39 (Continued)

a) First feature map, b) Second feature map, c) Third feature map, d) Fusion results of (a), (b) and (c), e) Thresholding result of (d) (targets are encircled).

The application results of Fe3 are given in Figure 3-40. In Fe3, two user-dependent parameters that are related to the feature extraction process are the small and large window size. The small window size is 2x2 and the large window size is selected as 4x4.



(a) (b) (c)



(d) (e) (f)

Figure 3-40: The application results of Fe3.

a) First feature map, b) Second feature map, c) Third feature map, d) Fourth feature map, e) Fusion result of (a), (b), (c) and (d), f) Thresholding result of (e) (targets are encircled).

3.5 Comparison Results of the Algorithms

The algorithms detailed in section 3.3 are compared in this section. These algorithms are applied to the prepared scenarios which are given in Chapter 2. The algorithms are applied to 20th frame, 100th frame, 200th frame, 300th frame, 400th frame and 500th frame of each scenario. These frames are selected randomly and represent the different distances. For example, the 20th frame represents the farthest distance and 500th frame represents the nearest distance between imaging system and target. The performance of the algorithms are analyzed first within the groups and then between the groups in this section.

Four criteria are used to present the overall performance of these algorithms. Two of these are related with the detection performance that is defined using the recall value and precision value. Recall indicates the number of target pixels detected. Precision determines how many detected pixels belong to the target. The recall value is calculated by dividing the number of detected target pixels by the number of actual target pixels. The precision value is calculated by dividing the number of detected target pixels by the number of all the detected pixels. A further two performance criteria are the number of user-dependent parameter and the processing time calculated in one frame. User-dependent parameter is defined as the parameter that has a value that is not given in references.

The ground truth table that includes the number of actual target pixels is given in Table 3-1. T1, T2 and T3 represent the three targets in the image and the numbering is presented from left to right. The number of target pixels is counted manually by analyzing all the detection results.

Table 3-1: The ground truth of the corresponding frames of each scenario

Ground Truth Table				
Scenario	Frame	T1	T2	T3
Scenario 1	20	6	4	9
	100	6	4	6
	200	7	9	10
	300	11	11	14
	400	20	14	21
	500	42	50	52
Scenario 2	20	4	4	5
	100	4	4	6
	200	7	9	12
	300	11	12	16
	400	23	20	23
	500	46	53	54
Scenario 3	20	2	2	4
	100	2	2	3
	200	5	5	4
	300	5	8	6
	400	20	24	21
	500	38	50	48

3.5.1 Morphological Operation Based Algorithms

Six MO based algorithms are compared in this section. The first five of them have a thresholding method thus, in order to compare all the algorithms under the same performance criteria a thresholding method is added to MO6. First, the comparison of these algorithms is carried out in terms of detection capability.

The detection results of the six MO based algorithms obtained from the first scenario are given in Table 3-2.

Table 3-2: The detection results of MO based algorithms in the first scenario.

Scenario 1	Algorithms	Number of detected pixels	Number of detected target pixels			R	P
			T1	T2	T3		
20th Frame	MO1	16	6	4	6	88.8	100
	MO2	21956	6	4	7	89.2	0.1
	MO3	18	6	4	5	78	84
	MO4	18	5	4	9	94	100
	MO5	15	6	4	5	78	100
	MO6	19	6	4	9	100	100
100th Frame	MO1	16	6	4	6	100	100
	MO2	18358	6	4	6	100	0.1
	MO3	18	6	4	6	100	88.9
	MO4	17	5	4	6	93.75	94.2
	MO5	18	6	4	6	100	88.9
	MO6	17	6	5	6	100	95
200th Frame	MO1	18	3	7	8	69.2	100
	MO2	19819	6	7	8	84	0.1
	MO3	19	6	5	8	73	100
	MO4	26	7	9	10	100	100
	MO5	18	6	7	4	65	94.5
	MO6	27	7	8	10	96.16	92.6
300th Frame	MO1	26	8	9	7	72.2	100
	MO2	17238	8	8	11	75	0.2
	MO3	36	10	11	10	86.1	86.2
	MO4	34	10	11	13	94.4	100
	MO5	25	6	5	5	44.4	64
	MO6	39	11	11	14	100	92.4

Table 3-2 (Continued)							
400th Frame	MO1	52	20	14	18	92.8	100
	MO2	11503	4	4	5	23.6	0.2
	MO3	54	4	0	0	7.2	7.5
	MO4	51	18	18	15	92.7	100
	MO5	122	4	0	0	7.2	3.3
	MO6	54	20	14	16	90.1	92.6
500th Frame	MO1	55	23	18	14	38.19	100
	MO2	134	35	44	48	93	100
	MO3	116	1	0	1	1.3	1.8
	MO4	22	6	5	11	15.2	100
	MO5	37	2	0	1	2	8.2
	MO6	2310	10	9	12	56.3	1.35

When the detection results obtained from the first scenario are analyzed, it can be seen that MO2 produced too many false alarms. The reason for producing too many false alarms is the method used as thresholding. Otsu thresholding method works well under bimodal histogram. However, the test images do not have a bimodal histogram. The precision values of MO2 are really low despite detecting most of the target. The detection performance of MO3, MO5 and MO6 decreases with the enlargement of the target size. This problem is addressed with the use of a fixed sized structural element in these algorithms. MO1 and MO4 maintain the detection performance until the last point. MO4 has a very effective pre-processing method that has important role in detection performance. However, MO1 has better detection performance than MO4 in this scenario when precision values are considered. MO1 has a functional filtering process in the pre-processing stage. An adaptive thresholding method is also used in the detection stage and they substantially improve the detection performance of MO1. In the second scenario, the MO1 and MO4 algorithms are determined as the best.

The detection results of all the MO based algorithms obtained from the second scenario are given in Table 3-3.

Table 3-3: The detection results of MO based algorithms obtained from the second scenario.

Scenario 2	**Algorithms**	**Number of detected pixels**	**Number of detected target pixels**			**R**	**P**
			T1	**T2**	**T3**		
20th Frame	MO1	195	4	4	5	100	6
	MO2	9053	4	4	3	78.5	0.02
	MO3	14	4	4	2	71.4	71.5
	MO4	1	0	0	1	7.1	100
	MO5	3	2	0	1	21.4	100
	MO6	13	4	4	5	100	100
100th Frame	MO1	172	4	4	6	100	8.13
	MO2	12650	4	4	6	100	0.1
	MO3	14	4	4	4	85.7	85.8
	MO4	9	2	3	4	64.2	100
	MO5	6	2	1	3	42.8	100
	MO6	13	4	4	5	92.87	100
200th Frame	MO1	50	7	9	12	100	56
	MO2	36252	6	8	8	78.5	0.1
	MO3	28	7	6	8	75	75
	MO4	12	4	4	4	42.8	100
	MO5	7	2	1	4	25	100
	MO6	16	4	5	7	57.14	100
300th Frame	MO1	48	11	12	16	100	81.25
	MO2	18	6	6	6	46	100
	MO3	30	9	10	10	74.3	96.7
	MO4	20	6	6	8	51.2	100
	MO5	8	4	2	2	20	100
	MO6	19	6	6	7	48.71	100

Table 3-3 (Continued)

400th Frame	MO1	94	23	20	23	100	70.2
	MO2	46	12	17	17	69.6	100
	MO3	48	14	10	13	53.6	77.1
	MO4	47	12	26	18	68.1	100
	MO5	6	4	2	0	8.6	100
	MO6	42	12	16	14	63.6	100
500th Frame	MO1	141	44	50	49	93.4	100
	MO2	17	7	4	6	11.1	100
	MO3	130	14	7	6	17.6	20.8
	MO4	66	30	34	2	43.13	100
	MO5	0	0	0	0	0	0
	MO6	68	33	35	0	47,5	100

When the detection results obtained from the second scenario are analyzed, MO1 and MO2 have too many false alarms until the 200th frame and 300th frame respectively. These problems are related to low contrast between target and background. It is observed that the MO1 pre-processing stage does not handle this scenario and the MO2 thresholding method problem continues in the second scenario. Median filtering and H filtering used in MO1 are not suitable for the second scenario. Therefore, MO3 has good detection performance while the target size is small. However, with the target size enlargement, the detection performance of MO3 is decreased. MO3 does not have any pre-processing stage and it probably reduces the detection performance. MO4 and MO6 detect target with 100% precision. MO5 cannot detect any target pixel in the last frame and it decreases overall detection performance. MO6 has the best detection performance while the target size is small but MO4 detect target pixels better than MO6 in the large target sizes. Block based pre-processing method of MO4 is still very effective in the second scenario. Under these analyzes, MO4 and MO6 are determined to be the best algorithms in the second scenario.

The detection results of all the MO based algorithms in the third scenario are given in Table 3-4.

Table 3-4: The detection results of MO based algorithms obtained from the third scenario

Scenario 3	Algorithms	Number of detected pixels	Number of detected target pixels			R	P
			T1	T2	T3		
20th Frame	MO1	13020	2	2	2	75	0.04
	MO2	28888	2	2	4	100	0.1
	MO3	7	2	2	0	50	57.2
	MO4	5	2	2	1	62.5	100
	MO5	131	2	2	4	100	6.2
	MO6	4	2	2	0	66.7	100
100th Frame	MO1	14229	2	2	3	100	0.04
	MO2	23526	2	2	3	100	0.1
	MO3	7	2	2	2	85	85.72
	MO4	5	2	2	1	71.4	100
	MO5	174	2	2	3	100	4.1
	MO6	6	2	2	2	85.7	100
200th Frame	MO1	15436	5	5	4	100	0.09
	MO2	24201	5	5	4	100	0.1
	MO3	14	2	4	3	64.28	64.29
	MO4	14	4	4	4	85.71	85.72
	MO5	177	5	4	2	78.57	6.22
	MO6	2	0	0	2	14.28	100
300th Frame	MO1	14432	5	8	6	100	0.13
	MO2	20353	5	8	6	100	0.1
	MO3	19	3	5	4	63.15	63.16
	MO4	70	4	6	5	78.94	79
	MO5	16	3	4	3	52.63	52.64
	MO6	7	3	1	2	36.8	100

Table 3-4 (Continued)

400th Frame	MO1	12049	20	24	21	100	0.53
	MO2	11246	20	24	21	100	0.6
	MO3	62	8	12	7	41.53	46.16
	MO4	32	12	8	9	44.61	90.7
	MO5	8	3	2	3	12.3	100
	MO6	11	4	2	5	16.9	100
500th Frame	MO1	5344	38	50	48	100	2.54
	MO2	62	22	15	24	44.85	98.39
	MO3	133	37	49	47	97.79	100
	MO4	84	29	28	27	61.7	100
	MO5	6	1	2	0	2.2	50
	MO6	28	11	6	11	20.5	100

When the detection results obtained in the third scenario are analyzed, the false alarm of the MO1 and MO2 are shown to be too high as in the previous scenarios. The reason for this is same as the other scenarios. The MO1 algorithm cannot overcome low contrast between target and background. The detection performance of MO3 is at the average level but in the last frame it produces the best values. MO4 has a good detection performance for the small target size but its performance decreased with the enlargement of the target size. The MO5 algorithm has too many false alarms when the target size is small. MO6 has a 100% precision value in each frame. The recall values obtained with MO6 are unstable but the 100% precision value made MO6 the best in this scenario.

The number of user-dependent parameter and processing time of each algorithm are given in Table 3-5. When the processing times of the algorithms are analyzed, it can be seen that the order from the fast algorithm to slowest algorithm is MO6, MO3, MO2, MO1, MO4 and MO5.

The RGF non-linear filtering method used in the MO5 post-processing takes too much time. The Windowing structure used in the pre-processing stage of MO4 is also time-consuming process. Median filtering used with MO1 and the use of a different structural element in MO2 created the processing load. MO5 is the slowest algorithm and also has the most parameters. Not using pre-processing or post-processing in MO3 and MO6 substantially decreases the PT.

MO5 consist six parameters that are related to the target enhancement operation, modified THT and the threshold value selection. MO3 uses four parameters and three are related to the threshold selection. MO1 has the following three parameters; structural element size and offset value. MO6 uses the two parameters of structural element size and threshold constant. MO4 has one parameter and MO2 has zero parameters. All the algorithms except MO2 require a SE size.

Table 3-5: The number of UDP and the PT of each of the MO based algorithms

Algorithms	**UDP**	**PT (s)**
MO1	3	0.1053
MO2	0	0.0732
MO3	4	0.0202
MO4	1	0.1302
MO5	6	2.1626
MO6	2	0.0159

The detection results obtained in each scenario containing the MO based algorithms are given in Table 3-6. In this table, the mean and standard deviations of the precision and recall values are calculated for each scenario detection result.

Table 3-6: The overall performance results of MO based algorithms in each scenario.

		20th frame		100th frame		200th frame		300th frame		400th frame		500th frame							
Scenario	**Algorithm**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R Mean**	**R Std**	**P Mean**	**P Std**	**UDP**	**PT (s)**
Scenario 1	MO1	100	100	100	100	100	100	100	100	100	100	100	100	100	0	100	0	3	0,1053
	MO2	89,2	0,1	100	0,1	84	0,1	75	0,2	23,6	0,2	93	100	77,47	27,7	16,78	40,8	0	0,0732
	MO3	78	84	100	88,9	73	100	86,1	86,2	7,2	7,5	1,3	1,8	57,6	42,4	61,4	44,3	4	0,0202
	MO4	94	100	93,75	94,2	100	100	94,4	100	92,7	100	15,2	100	81,68	32,7	99,03	2,37	1	0,1302
	MO5	78	100	100	88,9	65	94,5	44,4	64	7,2	3,3	2	8,2	49,43	39,2	59,82	43,7	6	2,1626
	MO6	100	100	100	95	96,16	92,6	100	92,4	90,1	92,6	56,3	1,35	90,43	17,2	78,99	38,1	2	0.0159
Scenario 2	MO1	100	6	100	8,13	100	56	100	81,25	100	100	93,4	93,4	98,9	2,69	57,46	41,8	3	0,1053
	MO2	78,5	0,2	100	0,1	78,5	0,1	46	100	69,6	100	11,1	100	63,95	31,2	50,07	54,7	0	0,0732
	MO3	71,4	71,5	85,7	85,8	75	75	74,3	96,7	53,6	77,1	17,6	20,8	62,93	24,5	71,15	26,3	4	0,0202
	MO4	7,1	100	64,2	100	42,8	100	51,2	100	68,1	100	43,13	100	46,09	21,8	100	0	1	0,1302
	MO5	21,4	100	42,8	100	25	100	20	100	8,6	100	0	0	19,63	14,7	83,33	40,8	6	2,1626
	MO6	100	100	92,87	100	57,14	100	48,71	100	63,6	100	47,5	100	68,3	22,7	100	0	2	0.0159
Scenario 3	MO1	75	0,04	100	0,04	100	2,54	100	0,13	100	0,53	100	0,09	95,83	10,2	0,562	0,99	3	0,1053
	MO2	100	0,1	100	0,1	100	0,1	100	0,1	100	0,6	44,85	98,39	90,81	22,5	16,57	40,1	0	0,0732
	MO3	50	57,2	85	85,72	64,28	64,29	63,15	63,16	41,53	46,16	97,79	100	66,96	21,1	69,42	19,8	4	0,0202
	MO4	62,5	100	71,4	100	85,71	85,72	78,94	79	44,61	90,7	61,7	100	67,48	14,6	92,57	8,95	1	0,1302
	MO5	100	6,2	100	4,1	78,57	6,22	52,63	52,64	12,3	100	2,2	50	57,62	42,9	36,53	38,4	6	2,1626
	MO6	66,7	100	85,7	100	14,28	100	36,8	100	16,9	100	20,5	100	40,15	29,6	100	0	2	0.0159

When the performances of all scenarios are analyzed, it is observed that MO1 cannot handle the low contrast level between the target and background. It also needs three parameters which directly affect the detection performance. MO2 algorithm has no pre-processing stage. In addition, the use of the Otsu thresholding method decreases the precision of the detection. Having no parameter does not have a positive effect. The MO3 algorithm has a good precision value. However, it cannot reach 100%. If some pre-processing methods are added to MO3, the precision values can be increased. However, this algorithm requires four parameters and three are used to calculate the threshold value and greatly affect the detection performance. The MO4 algorithm detects the target with high accuracy and it only needs one parameter that is structural element. The pre-processing method used in MO4 is very effective however, the duration of the processing time is high. The MO5 detection performance is not high. In addition, it needs six parameters that are directly related to detection performance and it also takes too much processing time. MO6 produces a good detection performance in the second and third scenarios. It does not have pre-processing stage and it is considered that if a pre-processing method is used, the performance of MO6 could be increased.

3.5.2 Wavelet Transform Based Algorithms

Five WT based algorithms are compared in this section. The WT2 and WT5 algorithms do not have a thresholding method. To compare all algorithms equally, a thresholding method is added to WT2 and WT4. The thresholding method calculates the threshold value by multiplying the maximum value with a fixed constant.

The detection results of the Wavelet transform based algorithms obtained from the first scenario are given in Table 3-7.

Table 3-7: The detection results of the WT based algorithms obtained from the first scenario

Scenario1	Algorithms	Number of detected pixels	Number of detected target pixels			R	P
			T1	T2	T3		
20th Frame	WT1	8	2	2	2	31	75
	WT2	6	1	1	2	21.05	66.7
	WT3	30682	4	4	9	89.47	0.1
	WT4	847	1	3	1	26.31	0.6
	WT5	1	0	0	0	0	0
100th Frame	WT1	8	1	3	2	37.5	99.75
	WT2	10	3	3	1	43.75	70
	WT3	32366	6	4	5	93.75	0.1
	WT4	21	3	3	4	62.5	47.7
	WT5	1	0	0	0	0	0
200th Frame	WT1	8	2	4	2	2.82	87.5
	WT2	5	2	1	2	19.2	100
	WT3	35474	7	9	10	100	0.1
	WT4	21	2	4	6	46.15	57.2
	WT5	1	0	0	0	0	0
300th Frame	WT1	10	4	4	2	27.7	100
	WT2	5	3	1	0	11.1	80
	WT3	32371	11	11	14	100	0.1
	WT4	34	7	7	8	61.1	64.71
	WT5	1	0	0	0	0	0
400th Frame	WT1	11	4	3	4	20	100
	WT2	7	3	3	1	12.72	100
	WT3	24023	17	14	16	85.4	0.1
	WT4	56	5	7	6	32.77	32.2
	WT5	1	0	0	0	0	0

Table 3-7 (Continued)

500th Frame	WT1	10	4	2	4	6.9	100
	WT2	28	12	11	1	23.61	85.72
	WT3	9693	34	35	35	72.2	1.1
	WT4	130	2	8	7	11.8	13.08
	WT5	1	0	0	0	0	0

The analysis of the detection results of the WT based algorithms obtained from the first scenario indicates that WT3, WT4 and WT5 algorithms produce too many false alarms. These false alarms reduce the P values. There are no pre-processing or post-processing stages in WT3, WT4 and WT5; this may be the reason for the production of so many false alarms. In addition, the WT3 and WT4 algorithms are included in the first type detection group. It is considered that WT does not make good background estimation. The WT1 and WT2 algorithms produce better detection results than the other algorithms. They aim to directly obtain target information instead of estimating the background. In particular, using THT as a pre-processing method in WT1 is a good approach. The WT2 algorithm fuses different scale horizontal and vertical information extracted with WT. This approach decreases the false alarm rate. In this scenario, WT1 is determined as the best algorithm considering the detection results presented above.

The detection results of the Wavelet transform based algorithms obtained from the second scenario are shown in Table 3-8.

Table 3-8: The detection results of the WT based algorithms obtained from the second scenario

Scenario2	Algorithms	Number of detected pixels	Number of detected target pixels			R	P
			T1	T2	T3		
20th Frame	WT1	4	1	1	2	28.5	100
	WT2	7	1	2	2	38.46	71.43
	WT3	12864	4	4	5	100	0.1
	WT4	1469	4	1	2	53.84	0.5
	WT5	1	0	0	0	0	0
100th Frame	WT1	4	1	1	2	28.5	100
	WT2	10	3	4	1	57.14	80
	WT3	15326	4	4	4	85.71	0.1
	WT4	1665	4	4	5	92.87	0.8
	WT5	1	0	0	0	0	0
200th Frame	WT1	4	1	1	1	10.7	75
	WT2	9	3	0	6	32.14	100
	WT3	52011	7	49	11	100	0.1
	WT4	26	6	7	8	75	80.77
	WT5	1	0	0	0	0	0
300th Frame	WT1	5	2	1	2	12.8	100
	WT2	9	1	4	2	17.9	77.8
	WT3	18	6	6	6	46.15	100
	WT4	31	5	6	7	46.15	58.07
	WT5	1	0	0	0	0	0
400th Frame	WT1	11	3	6	2	15.9	100
	WT2	3	3	0	0	4.5	100
	WT3	22	12	5	5	33.3	100
	WT4	54	11	12	6	43.96	53.71
	WT5	1	0	0	0	0	0

Table 3-8 (Continued)

500th Frame	WT1	14	4	4	6	9.1	100
	WT2	21	2	14	2	13.72	100
	WT3	92	21	23	23	39.6	77.2
	WT4	121	4	6	4	9.15	11.58
	WT5	1	0	0	0	0	0

Table 3-8 shows that the detection results pertaining to the WT3, WT4 and WT5 algorithms have a very low precision values. These results are very similar to those of the first scenario. The lack of the pre-processing and post-processing stages affects the detection performance of these algorithms in negative way. The WT2 algorithm also indicates good performance however, the mean precision value does not reach 100%. The WT1 algorithm can detect the target in the fourth stage of the second scenario with 100% precision. The WT1 algorithm is a step ahead since it uses Top-Hat transform as pre-processing method. The best detection results are produced by the WT1 algorithm in the second scenario. These results indicate the importance of pre-processing stage of algorithms on detection performance.

The detection results of Wavelet transform based algorithms obtained from the third scenario are presented in Table 3-9.

Table 3-9: The detection results of WT based algorithms obtained from the third scenario

Scenario 3	**Algorithms**	**Number of detected pixels**	**Number of detected target pixels**			**R**	**P**
			T1	**T2**	**T3**		
20th Frame	WT1	3	1	1	1	37.5	100
	WT2	15	2	0	1	37.5	20
	WT3	36903	2	2	4	100	0.1
	WT4	28943	1	1	1	37.5	0.1
	WT5	1	0	0	0	0	0

Table 3-9 (Continued)

100th Frame	WT1	3	1	1	1	42.8	100
	WT2	11	2	2	2	85.71	54.55
	WT3	36793	2	2	3	100	0.1
	WT4	45896	2	2	3	100	0.1
	WT5	1	0	0	0	0	0
200th Frame	WT1	2	0	1	1	14.28	100
	WT2	4	0	0	4	28.57	100
	WT3	40553	5	5	4	100	0.1
	WT4	23664	5	5	4	100	0.1
	WT5	1	0	0	0	0	0
300th Frame	WT1	7	2	1	1	21	57.15
	WT2	13	4	0	5	47.36	69.24
	WT3	31365	5	8	6	100	0.1
	WT4	10050	5	8	6	100	0.1
	WT5	1	0	0	0	0	0
400th Frame	WT1	13	4	4	4	18.46	92.31
	WT2	12	4	2	6	18.46	100
	WT3	33621	13	17	18	73.84	0.1
	WT4	7346	20	24	21	100	0.9
	WT5	1	0	0	0	0	0
500th Frame	WT1	17	5	2	5	8.8	70.6
	WT2	19	9	2	8	13.97	100
	WT3	13791	35	40	41	85.2	0.9
	WT4	1008	35	40	35	80.8	10.92
	WT5	1	0	0	0	0	0

In the third scenario, again WT3, WT4 and WT5 have a high false alarm rate. The WT1 algorithm can detect the target with a good precision value when the size of the target is small. However, the WT1 detection performance decreases with the enlargement of the target. This problem is related to the fixed size structural element use in the pre-processing stage. The WT2 detection performance has inverse characteristics, its performance is low when the target size is small however, enlarging the target size improves the performance of WT2. The WT1 algorithm is determined as the best in the third scenario.

The number of user-dependent parameter and the processing time of each of the WT based algorithms are given in Table 3-10. The WT5 algorithm has two user-dependent parameters that are the window sizes used in the thresholding step. WT1, WT2 and WT4 have one user-dependent parameter and they are all threshold constant. WT3 also has one user-dependent parameter and it is related to the wavelet family. The order of processing time from fastest to slowest is WT3, WT4, WT1, WT2 and WT5. PT's are really similar to each other except obtained from WT5. Thresholding method used in WT5 takes too much time which includes sliding window structure.

Table 3-10: The number of UDP and PT of each WT based algorithms

Algorithms	**UDP**	**PT (s)**
WT1	1	0.0780
WT2	1	0.0855
WT3	1	0.0424
WT4	1	0.0519
WT5	2	0.2995

The whole performance evaluations are given in Table 3-11. In this table, mean and standard deviation of precision and recall values are included.

Table 3-11: The performance evaluation results of each WT based algorithm in three scenarios.

Scenario	Methods	20th Frame R	20th Frame P	100th Frame R	100th Frame P	200th Frame R	200th Frame P	300th Frame R	300th Frame P	400th Frame R	400th Frame P	500th Frame R	500th Frame P	R Mean	R Std	P Mean	P Std	UDP	PT (s)
Scenario 1	WT1	31	75	37,5	99,75	2,82	87,5	27,7	100	20	100	6,9	100	20,99	13,77	93,71	10,43	1	0,078
	WT2	21,05	66,7	43,75	70	19,2	100	11,1	80	12,72	100	23,61	85,72	21,91	11,74	83,74	14,32	1	0.0855
	WT3	89,47	0,1	93,75	0,1	100	0,1	100	0,1	85,4	0,1	72,2	1,1	90,14	10,51	0,267	0,408	1	0.0424
	WT4	26,31	0,6	62,5	47,7	46,15	57,2	61,1	64,71	32,77	32,2	11,8	13,08	40,11	20,12	35,92	25,31	1	0.0519
	WT5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.2995
Scenario 2	WT1	28,5	100	28,5	100	10,7	75	12,8	100	15,9	100	9,1	100	17,58	8,757	95,83	10,21	1	0,078
	WT2	38,46	71,43	57,14	80	32,14	100	17,9	77,8	4,5	100	13,72	100	27,31	19,14	88,21	13,22	1	0.0855
	WT3	100	0,1	85,71	0,1	100	0,1	46,15	100	33,3	100	39,6	77,2	67,46	31,14	46,25	51,24	1	0.0424
	WT4	53,84	0,5	92,87	0,8	75	80,77	46,15	58,07	43,96	53,71	9,15	11,58	53,5	28,72	34,24	34,3	1	0.0519
	WT5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.2995
Scenario 3	WT1	37,5	100	42,8	100	14,28	100	21	57,15	18,46	92,31	8,8	70,6	23,81	13,42	86,68	18,41	1	0,078
	WT2	37,5	20	85,71	54,55	28,57	100	47,36	69,24	18,46	100	13,97	100	38,6	26,12	73,97	32,7	1	0.0855
	WT3	100	0,1	100	0,1	100	0,1	100	0,1	73,84	0,1	85,2	0,9	93,17	11,17	0,233	0,327	1	0.0424
	WT4	37,5	0,1	100	0,1	100	0,1	100	0,1	100	0,9	80,8	10,92	86,38	25,15	2,037	4,364	1	0.0519
	WT5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.2995

When the total performance of WT based algorithms are analyzed, it can be seen that WT1 has the best detection performance. WT1, WT2, WT3 and WT4 have one UDP and the fastest algorithm is WT3. These explanations make it hard to determine the best between WT based algorithms. However, WT1 is considered to be better than other algorithms.

These detection results also show that the usage of MO as pre-processing method increases the performance of WT. Furthermore, frame subtraction used in WT3 does not affect the detection performance in a good way. On the other hand, the fusion of different scale WT's is shown to be a good choice by analyzing WT2's detection results. However, WT5 algorithm uses just the horizontal details obtained with WT in different scales. This assumption is not sufficient to increase the detection performance. Otsu thresholding method is not suitable for all images. It can misdirect the algorithm because the determination of correct threshold value with Otsu method used to separate target and non-target pixels usually fails. In addition to these results, detection performance of WT3, WT4 and WT5 indicates that wavelet transform can not estimate background very well.

3.5.3 Edge Detection Based Algorithms

The scenarios of the two ED based algorithms given in Chapter 3 are compared in this section. Both two algorithms have a thresholding method. The comparison is made first on the detection performance. The grand truth of the test scenarios is also given in Table 3-1.

The detection results of two edge detection based algorithms obtained from the first scenario are given in Table 3-12.

Table 3-12: The detection results of ED based algorithms obtained from the first scenario

Scenario 1	**Algorithms**	**Number of detected pixels**	**Number of detected target pixels**			**R**	**P**
			T1	**T2**	**T3**		
20th Frame	ED1	12	3	2	7	57.8	91.7
	ED2	24	4	4	4	63.15	50
100th Frame	ED1	11	3	1	2	37.5	54.6
	ED2	26	5	2	6	81.25	50
200th Frame	ED1	19	3	7	4	53.8	73.69
	ED2	28	4	8	3	57.6	53.6
300th Frame	ED1	25	5	6	9	55.5	80
	ED2	4	3	1	1	11.1	100
400th Frame	ED1	42	8	7	10	45.4	59.6
	ED2	44	17	8	15	72.7	91
500th Frame	ED1	54	11	15	17	29.8	83.4
	ED2	49	15	12	12	27	79.6

When the detection results of the ED based algorithms in the first scenario are analyzed, it can be seen that the two algorithms have precision values about 70%. This creates a problem. The reason for this problem is related to the pre-processing methods that are very similar with each other and they are not able to adequately prepare the raw image for the detection stage. They use global approximations for pre-processing. This global approach does not have a high peformance in the pre-processing stage. However, ED1 has higher precision value than ED2 in the first scenario.

The detection results of edge detection based algorithms obtained from the second scenario are given in Table 3-13.

Table 3-13: The detection results of ED based algorithms obtained from the second scenario

Scenario 2	Algorithms	Number of detected pixels	Number of detected target pixels			R	P
			T1	T2	T3		
20th Frame	ED1	8	1	1	3	38.4	62.5
	ED2	30	4	4	6	100	46.7
100th Frame	ED1	6	2	1	2	33.3	83.4
	ED2	16	4	2	2	57.14	50
200th Frame	ED1	21	5	8	3	57.14	76.2
	ED2	30	5	9	0	50	46.7
300th Frame	ED1	29	4	4	4	30.7	41.38
	ED2	42	6	9	6	53.8	50
400th Frame	ED1	45	10	15	16	62.12	91.2
	ED2	67	4	15	18	56	55.3
500th Frame	ED1	56	20	21	15	36.6	100
	ED2	82	10	35	32	50.3	94

The analysis of the detection results of the ED based algorithms obtained from the second scenario indicates that both algorithms have too many false alarms. These results are related with the low performance of pre-processing methods. Furthermore, these detection results indicate that global pre-processing approaches are not suitable to obtain high detection precision. However, the ED1 algorithm produces higher precision values than the ED2 algorithm.

The detection results of the edge detection based algorithms obtained from the third scenario are given in Table 3-14.

Table 3-14: The detection results of ED based algorithms obtained from the third scenario

Scenario 3	Algorithms	Number of detected pixels	Number of detected target pixels			R	P
			T1	T2	T3		
20th Frame	ED1	3814	1	2	1	50	0.2
	ED2	20	0	2	4	75	30
100th Frame	ED1	4509	1	1	3	71.4	0.1
	ED2	10	0	2	0	28.5	20
200th Frame	ED1	5305	1	1	3	35.71	0.1
	ED2	14	5	0	2	50	46.7
300th Frame	ED1	4861	1	3	4	42.1	0.1
	ED2	40	5	4	5	77.8	35
400th Frame	ED1	4707	17	18	17	80	1.2
	ED2	42	10	13	10	50.7	78.6
500th Frame	ED1	4173	20	31	38	65.4	2.4
	ED2	30	14	8	8	22	100

The analysis of the detection results from the third scenario shows that ED2 detects target better than ED1. However, both algorithms produce high false alarms. The ED2 precision values decrease with the enlargement of the size of the targets. Inversely, the ED1 algorithm precision values do not slightly change in the scenario. The row-based clutter cancelling pre-processing method used in ED1 does not work well in the third scenario due to lack of contrast level difference of the scenario because the gray level of target and background pixels are very close in the same row.

The number of UDP and processing time of the algorithms are given in Table 3-15. The ED1 algorithm has one user-dependent parameter related to the threshold calculation. The ED2 algorithm has the following three user-dependent parameters: Gaussian filter size, structural element size and threshold constant. These user-dependent parameters are used to calculate the threshold values. Furthermore, the ED1 is faster than the ED2.

Table 3-15: the UDP and PT values of the ED based algorithms

Algorithms	**UDP**	**PT (s)**
ED1	1	0.11
ED2	3	0.30

The overall performance results of the ED based algorithms are given in the Table 3-16 showing, the mean and standard deviation of the recall and precision values obtained in each scenario.

Table 3-16: Overall performance results of the ED based algorithms

		20th Frame		100th Frame		200th Frame		300th Frame		400th Frame		500th Frame							
Scenario	**Methods**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R Mean**	**R Std**	**P Mean**	**P Std**	**UDP**	**PT (s)**
Scenario 1	**ED1**	57,8	91,7	37,5	54,6	53,8	73,69	55,5	80	45,4	59,6	29,8	83,4	46,63	11,16	73,83	14,29	1	0.11
	ED2	63,15	50	81,25	50	57,6	53,6	11,1	100	72,7	91	27	79,6	52,13	27,34	70,7	22,36	3	0.30
Scenario 2	**ED1**	38,4	62,5	33,3	83,4	57,14	76,2	30,7	41,38	62,12	91,2	36,6	100	43,04	13,21	75,78	21,18	1	0.11
	ED2	100	46,7	57,14	50	50	46,7	53,8	50	56	55,3	50,3	94	61,21	19,22	57,12	18,34	3	0.30
Scenario 3	**ED1**	50	0,2	71,4	0,1	35,71	0,1	42,1	0,1	80	1,2	65,4	2,4	57,44	17,49	0,683	0,945	1	0.11
	ED2	75	30	28,5	20	50	46,7	77,8	35	50,7	78,6	22	100	50,67	22,99	51,72	31,1	3	0.30

Both algorithms have a pre-processing stage. Background suppression is used as pre-processing. However, different background suppression techniques are used in each algorithm. The ED1 suppresses the background in each row. Inversely, the ED2 suppresses the background in the whole image. It can be seen from the detection results that the suppression of the background in each row works better in the first and second scenarios. However, row based background suppression technique decreases the performance in the third scenario. This is not surprising since the gray level difference of the target and background is nearly same in the third scenario. This algorithm suppresses both the background and target.

Both ED based algorithms use the same thresholding methods. When the mean precision values are analyzed, it can be seen that the algorithms always produce false alarms. In addition, the processing time of the algorithms is quite high when compared with the other algorithms. No algorithm is determined as the best because the detection performance, number of user-dependent parameter and processing time do not intersect in one algorithm. To resolve this problem, changing the thresholding method would be a good choice.

3.5.4 Filtering Based Algorithms

Section 3.3 contains the detail of two filtering based algorithms and the application results. The Fi1 algorithm does not contain a thresholding method. To compare the two algorithms equally, a thresholding method is added to Fi1. The threshold value used in the thresholding method is calculated by multiplying the maximum value with a constant. In the first step, Fi based algorithms are compared in terms of the detection performance.

The detection results of the filtering based algorithms obtained from the first scenario are given in Table 3-17.

Table 3-17: The detection results of the Fi based algorithms in the first scenario

Scenario 1	Algorithms	Number of detected pixels	Number of detected target pixels			R	P
			T1	T2	T3		
20th Frame	Fi1	19	6	4	9	100	100
	Fi2_1	11	2	4	3	57.89	100
	Fi2_2	9	4	4	1	47.36	100
100th Frame	Fi1	17	6	4	6	100	94.2
	Fi2_1	12	2	4	4	62.5	83.33
	Fi2_2	13	4	4	4	75	92.31
200th Frame	Fi1	27	7	9	10	100	96.3
	Fi2_1	6	0	1	2	11.53	50
	Fi2_2	11	4	3	4	42.3	100
300th Frame	Fi1	34	10	11	13	94.4	100
	Fi2_1	2	0	0	0	0	0
	Fi2_2	0	0	0	0	0	100
400th Frame	Fi1	61	18	14	15	85.4	78.5
	Fi2_1	4	0	0	2	3.63	50
	Fi2_2	0	0	0	0	0	100
500th Frame	Fi1	145	42	42	47	90.9	90.4
	Fi2_1	8	0	0	4	27.78	50
	Fi2_2	5	0	0	0	0	0

The analysis of the Fi based algorithms indicates that in the first scenario this algorithm produces the highest precision values. The recall values obtained with the Fi based algorithms shows that Fi1 detects target pixels better than the other algorithms. Fi1 includes a post-processing stage which allows it to produce better detection performance. This post-processing method is related with background estimation enhancement and it really reduces the false alarm. Fi2 does not have any pre-processing or post-processing stages and this decreases its detection performance.

The detection results of the filtering based algorithms obtained from the second scenario are given in Table 3-18.

Table 3-18: The detection results of the Fi based algorithms obtained from the second scenario

Scenario 2	**Algorithms**	**Number of detected pixels**	**Number of detected target pixels**			**R**	**P**
			T1	**T2**	**T3**		
20th Frame	Fi1	7	2	4	1	53.84	100
	Fi2_1	7	2	3	2	53.84	100
	Fi2_2	9	4	4	1	69.23	100
100th Frame	Fi1	12	4	4	4	85.71	100
	Fi2_1	15	2	3	4	64.28	60
	Fi2_2	12	4	4	4	85.71	100
200th Frame	Fi1	16	5	6	5	57.14	100
	Fi2_1	7	1	2	4	25	100
	Fi2_2	12	4	4	4	42.85	100
300th Frame	Fi1	20	6	6	8	51.28	100
	Fi2_1	7	2	2	3	17.94	100
	Fi2_2	11	4	3	4	28.2	100
400th Frame	Fi1	46	12	17	17	69.6	100
	Fi2_1	2	0	0	2	43.48	100
	Fi2_2	6	4	0	2	13.04	100
500th Frame	Fi1	107	38	34	35	69.9	100
	Fi2_1	4	0	0	4	2.61	100
	Fi2_2	0	0	0	0	0	100

When the detection results obtained in the second scenario are analyzed, all the algorithms produce nearly 100% precision values. This result makes the recall values more important and these values obtained with Fi1 are higher than with other algorithms. The post-processing method of Fi1 increases the recall value with its background estimation enhancement characteristic.

The detection results of the filtering based algorithms obtained from the third scenario are given in Table 3-19.

Table 3-19: The detection results of the Fi based algorithms obtained from the third scenario

Scenario 3	**Algorithms**	**Number of detected pixels**	**Number of detected target pixels**			**R**	**P**
			T1	**T2**	**T3**		
20th Frame	Fi1	5	2	2	1	62.5	100
	Fi2_1	9	2	2	2	75	66.67
	Fi2_2	6	2	2	2	75	100
100th Frame	Fi1	6	2	2	2	85.71	100
	Fi2_1	6	2	2	2	85.71	100
	Fi2_2	6	2	2	2	85.71	100
200th Frame	Fi1	3	1	0	2	21.4	100
	Fi2_1	8	1	3	2	42.85	75
	Fi2_2	9	3	4	2	64.28	100
300th Frame	Fi1	6	3	1	2	31.57	100
	Fi2_1	17	2	2	2	31.57	35.3
	Fi2_2	12	3	4	4	57.89	91.67
400th Frame	Fi1	12	5	2	5	18.46	100
	Fi2_1	14	1	1	2	6.15	28.58
	Fi2_2	15	2	5	2	13.84	60
500th Frame	Fi1	24	10	6	8	17.6	100
	Fi2_1	10	0	1	1	1.47	20
	Fi2_2	23	2	5	4	8.08	47.83

When the detection results obtained in the third scenario are analyzed, the Fi1 produces 100% precision values in all stages of the third scenario. The Fi2_2 is better than the Fi2_1 in analyzing the precision values in this scenario. The median filter is more effective than the mean filter that is verified by the detection results.

The number of user-dependent parameter and PT are given in Table 3-20. F2 has two user-dependent parameters and Fi1 has three user-dependent parameters. The user-dependent parameters that affect the detection performance of Fi1 are k_1, k_2 and η. The user-dependent parameters of Fi2 are the threshold constant and window size. In addition, F1 is the fastest algorithm. The max-median or max-mean methods used in Fi2 take too much time.

Table 3-20: UDP and PT values of Fi based algorithms

Algorithms	**UDP**	**PT (s)**
Fi1	3	0.003
Fi2_1	2	23.92
Fi2_2	2	19.53

The overall performance results of the Fi based algorithms are given in Table 3-21. The mean and standard deviation of the recall and precision values are also given in this table.

Table 3-21: The overall performance comparison of the Fi based algorithms

		20th Frame		100th Frame		200th Frame		300th Frame		400th Frame		500th Frame							
Scenario	**Methods**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R Mean**	**R Std**	**P Mean**	**P Std**	**UDP**	**PT (s)**
Scenario 1	Fi1	31	100	37,5	94,2	2,82	96,3	27,7	100	20	78,5	6,9	90,4	20,99	13,77	93,23	8,087	3	0.003
	Fi2_1	21,05	100	43,75	83,33	19,2	50	11,1	0	12,72	50	23,61	50	21,91	11,74	55,56	34,43	2	23,92
	Fi2_2	89,47	100	93,75	92,31	100	100	100	100	85,4	100	72,2	0	90,14	10,51	82,05	40,31	2	19,53
Scenario 2	Fi1	28,5	100	28,5	100	10,7	100	12,8	100	15,9	100	9,1	100	17,58	8,757	100	0	3	0.003
	Fi2_1	38,46	100	57,14	60	32,14	100	17,9	100	4,5	100	13,72	100	27,31	19,14	93,33	16,33	2	23,92
	Fi2_2	100	100	85,71	100	100	100	46,15	100	33,3	100	39,6	100	67,46	31,14	100	0	2	19,53
Scenario 3	Fi1	37,5	100	42,8	100	14,28	100	21	100	18,46	100	8,8	100	23,81	13,42	100	0	3	0.003
	Fi2_1	37,5	66,67	85,71	100	28,57	75	47,36	35,3	18,46	28,58	13,97	20	38,6	26,12	54,26	31,21	2	23,92
	Fi2_2	100	100	100	100	100	100	100	91,67	73,84	60	85,2	47,83	93,17	11,17	83,25	23,27	2	19,53

Fi based algorithms detect targets by estimating the background of the image. However, the Fi1 has a post-processing stage, which increases the performance of background estimation. It can be seen from the precision values that the most important difference between Fi1 and Fi2 is the use of a post-processing method in Fi1. In addition, the Fi2 algorithm processing time is too high however, the number of user-dependent parameter used in Fi1 are also higher than in Fi1.

3.5.5 Saliency Based Algorithms

The comparisons of four saliency-based algorithms detailed in the Section 3.3 are given in this section. Except for S1 these saliency based algorithms do not have a thresholding stage therefore to compare all the saliency based algorithms in the same way; a thresholding method is added to the other three algorithms. The threshold value is calculated by multiplying the maximum value of the last result with a fixed constant.

The detection results of the saliency based algorithms obtained from the first scenario are given in Table 3-22.

Table 3-22: The detection results of S based algorithms obtained from the first scenario

Scenario 1	**Algorithms**	**Number of detected pixels**	**Number of detected target pixels**			**R**	**P**
			T1	**T2**	**T3**		
20th Frame	S1	24	2	4	7	68.42	54.17
	S2	17	4	4	9	89.47	100
	S3	13	2	4	2	42.1	61.54
	S4	23	4	4	5	68.42	56,53
100th Frame	S1	26	3	4	5	75	46.16
	S2	15	5	4	6	93.75	100
	S3	17	4	4	4	75	70.59
	S4	24	4	4	4	75	50

Table 3-22 (Continued)

200th Frame	S1	30	5	6	4	57.69	50
	S2	27	7	9	10	100	96.3
	S3	12	4	4	2	38.46	83.4
	S4	21	5	5	5	57.69	71.43
300th Frame	S1	24	4	5	3	33.3	50
	S2	36	10	11	13	94.4	94.5
	S3	24	6	4	9	52.77	79.17
	S4	38	8	7	11	72.22	68.5
400th Frame	S1	24	4	6	2	21.81	50
	S2	53	18	14	17	89.09	92.5
	S3	27	9	4	4	30.9	62.97
	S4	36	9	9	5	41.81	63.89
500th Frame	S1	76	23	13	2	26.38	50
	S2	120	40	40	40	83.33	100
	S3	21	5	2	5	8.33	57.15
	S4	37	8	6	7	14.58	56.76

The analysis of the detection results obtained from the first scenario indicates that all algorithms produce false alarm. The reason for this problem may be the lack of the pre-processing and post-processing stages in all of them. However, the S2 algorithm has highest precision value. S2 works well in the first scenario because of the intensity value of the target is very high and this information is used as saliency in S2. If the algorithms are sorted in terms of the detection performance, the S3 is better than the S4 and the S4 is better than the S1. These evaluations are undertaken considering the precision values. In conclusion the S2 is the best in the second scenario.

The detection results of the saliency based algorithms obtained from the second scenario are given in Table 3-23.

Table 3-23: The detection results of the S based algorithms obtained from the second scenario

Scenario 2	Algorithms	Number of detected pixels	Number of detected target pixels			R	P
			T1	T2	T3		
20th Frame	S1	308	0	4	5	69.23	2.93
	S2	7	2	3	1	46.15	85.72
	S3	8	2	3	3	61.53	100
	S4	12	2	4	3	69.23	75
100th Frame	S1	124	4	4	6	100	11.3
	S2	10	2	4	4	71.42	100
	S3	11	2	4	4	71.42	90.91
	S4	22	4	4	4	85.71	54.6
200th Frame	S1	89	6	9	12	96.42	30.34
	S2	9	3	2	4	32.14	100
	S3	7	2	1	4	25	100
	S4	10	3	3	4	35.71	100
300th Frame	S1	84	11	11	14	92.3	42.86
	S2	16	4	6	6	41.02	100
	S3	15	4	5	6	38.46	100
	S4	19	5	8	6	48.71	100
400th Frame	S1	92	14	16	16	69.6	50
	S2	28	9	8	10	42.42	100
	S3	13	5	4	4	19.69	100
	S4	33	13	12	8	50	100
500th Frame	S1	242	40	41	37	77.12	48.77
	S2	94	30	30	34	61.4	100
	S3	12	1	4	7	7.8	100
	S4	33	9	9	15	21.56	100

The detection performance analysis of the saliency based algorithms in the second scenario indicates that the S1 algorithm has too many false alarms in all steps of the second scenario. Conversely, S2 generates 100% precision value except the first step. After the 100th frame of the second scenario, S2, S3 and S4 detect the targets with high precision. S2 and S3 produce low precision values only one time in the second scenario. The S3 algorithm also depends on the intensity value because it uses FT as saliency. The low intensity value of the target in the 20th frame caused the production of the low precision value of S1. These evaluations make the S2 and S3 the best in the second scenario.

The detection results of the saliency based algorithms obtained from the third scenario are given in Table 3-24.

Table 3-24: The detection results of the S based algorithms obtained from the second scenario

Scenario 3	**Algorithms**	**Number of detected pixels**	**Number of detected target pixels**			**R**	**P**
			T1	**T2**	**T3**		
20th Frame	S1	6645	1	0	1	25	0.1
	S2	456	2	2	1	62.5	1.1
	S3	7	2	2	1	62.5	71.43
	S4	5	2	2	1	62.5	100
100th Frame	S1	6884	2	2	3	100	0.2
	S2	189	2	2	2	85.71	3.2
	S3	6	2	2	2	85.71	100
	S4	6	2	2	2	85.71	100
200th Frame	S1	8461	5	5	5	100	0.17
	S2	38	2	1	1	42.85	15.79
	S3	2	0	0	0	14.28	100
	S4	2	0	0	0	14.28	100

Table 3-24 (Continued)

300th Frame	S1	8231	5	8	8	100	0.1
	S2	17	3	2	2	36.84	41.18
	S3	3	1	1	1	15.78	100
	S4	5	2	1	1	26.31	100
400th Frame	S1	5172	20	24	24	100	1.28
	S2	19	7	4	4	29.23	100
	S3	5	2	3	3	7.69	100
	S4	12	4	5	5	18.46	100
500th Frame	S1	6304	38	50	50	100	2.2
	S2	43	16	12	12	31.61	100
	S3	2	1	0	0	1.4	100
	S4	3	2	0	0	2.2	100

When the detection results of saliency based algorithms obtained from the third scenario are analyzed, it can be seen that S1 algorithm detects too many background pixels as a target. The S2 detection performance increases with the enlargement of the targets. S3 detects targets with a very high precision but in the first step it produces some false alarms. On the other hand, the S4 detects targets with a 100% precision in all steps of the third scenario. In accordance with these assessments, the S4 is considered as the best in the third scenario.

The number of user-dependent parameter and processing time values of the saliency based algorithms are given in Table 3-25 showing that the S1 and S2 have one user-dependent parameter which is the threshold constant. S3 and S4 have two user-dependent parameters which are the smoothing filter size and threshold constant. In addition, the S1 algorithm is the fastest. The S2 works very slowly since it calculates too much saliency information. The S3 and S4 processing times are very close to each other because they use DCT and FT that are nearly same function.

Table 3-25: The UDP number and PT values of S based algorithms

Algorithms	UDP	PT (s)
S1	1	0.08
S2	1	2.06
S3	2	0.12
S4	2	0.13

All performance results obtained from the three scenarios are given in Table 3-26.

Table 3-26: The overall performance comparison of S based algorithms

Scenario	Algorithms	20th Frame R	20th Frame P	100th Frame R	100th Frame P	200th Frame R	200th Frame P	300th Frame R	300th Frame P	400th Frame R	400th Frame P	500th Frame R	500th Frame P	R Mean	R Std	P Mean	P Std	UDP	PT (s)
Scenario 1	S1	68,42	54,17	75	46,16	57,69	50	33,3	50	21,81	50	26,38	50	47,1	22,82	50,06	2,534	1	0,08
	S2	89,47	100	93,75	100	100	96,3	94,4	94,5	89,09	92,5	83,33	100	91,67	5,696	97,22	3,277	1	2,06
	S3	42,1	61,54	75	70,59	38,46	83,4	52,77	79,17	30,9	62,97	8,33	57,15	41,26	22,23	69,14	10,45	2	0,12
	S4	68,42	56,53	75	50	57,69	71,43	72,22	68,5	41,81	63,89	14,58	56,76	54,95	23,2	61,19	8,15	2	0,13
Scenario 2	S1	69,23	2,93	100	11,3	96,42	30,34	92,3	42,86	69,6	50	77,12	48,77	84,11	13,8	31,03	19,97	1	0,08
	S2	46,15	85,72	71,42	100	32,14	100	41,02	100	42,42	100	61,4	100	49,09	14,53	97,62	5,83	1	2,06
	S3	61,53	100	71,42	90,91	25	100	38,46	100	19,69	100	7,8	100	37,32	24,84	98,49	3,711	2	0,12
	S4	69,23	75	85,71	54,6	35,71	100	48,71	100	50	100	21,56	100	51,82	22,98	88,27	19,29	2	0,13
Scenario 3	S1	25	0,1	100	0,2	100	0,17	100	0,1	100	1,28	100	2,2	87,5	30,62	0,675	0,876	1	0,08
	S2	62,5	1,1	85,71	3,2	42,85	15,79	36,84	41,18	29,23	100	31,61	100	48,12	21,93	43,55	46	1	2,06
	S3	62,5	71,43	85,71	100	14,28	100	15,78	100	7,69	100	1,4	100	31,23	34,4	95,24	11,66	2	0,12
	S4	62,5	100	85,71	100	14,28	100	26,31	100	18,46	100	2,2	100	34,91	32,19	100	0	2	0,13

When the total detection performances obtained in three scenarios are analyzed, all algorithms are shown to detect background pixels as a target and this decreases the precision values. This problem is due to the lack of pre-processing stage in all the algorithms. If some pre-processing methods are used to suppress the background or remove the noise, this can help to increase the precision values. Three of the algorithms use Fourier transform (FT) in the saliency detection. This common use shows us that FT is very important in detecting the salient region in image. The S1 uses second order directional derivative in addition to FT however, it does not increase the detection performance. The S2 uses Gabor filter and canny edge detection methods in addition to FT. These extra processes did not increase the detection performance. Finally, it is determined that S3 has the best detection performance due to the average of the precision mean values. In addition, its processing time is acceptable. However, it needs the user-dependent parameters of threshold constant and smoothing filter size. The threshold constant is extremely critical and affects detection performance.

3.5.6 Feature Based Algorithms

The three feature based algorithms detailed in Section 3.3 are compared in this section. Each feature based algorithm has a thresholding step so there is no need to add any thresholding step to algorithms.

The detection results of the feature based algorithms obtained from the first scenario are given in Table 3-27.

Table 3-27: The detection results of the Fe based algorithms obtained from the first scenario

Scenario 1	**Algorithms**	**Number of detected pixels**	**Number of detected target pixels**			**R**	**P**
			T1	**T2**	**T3**		
20th Frame	Fe1	2	0	1	1	10.52	100
	Fe2	97	6	4	9	100	19.59
	Fe3	4	0	0	0	0	0

Table 3-27 (Continued)

100th Frame	Fe1	1	1	0	0	6.25	100
	Fe2	117	6	4	6	100	13.68
	Fe3	4	0	0	0	0	0
200th Frame	Fe1	2	0	0	0	0	0
	Fe2	96	7	9	10	100	27.09
	Fe3	4	4	0	0	15.38	100
300th Frame	Fe1	5	0	1	1	5.55	40
	Fe2	104	11	11	14	100	34.62
	Fe3	8	8	0	0	22.22	100
400th Frame	Fe1	3	1	1	0	3.63	66.67
	Fe2	163	20	14	21	100	33.75
	Fe3	4	0	0	0	0	0
500th Frame	Fe1	0	0	0	0	0	100
	Fe2	267	42	50	52	100	54
	Fe3	8	0	0	8	5.55	100

From the analysis of the first scenario detection results Fe2 is shown to have too much noise in all the selected frames of the first scenario. It is observed that usage of a likelihood ratio does not work because the probabilities of target and background pixels are calculated using the Otsu thresholding method which does not provide good separation results. In the first two frames of the first scenario, Fe1 detects the target with a high precision. In addition, the Fe3 has an unstable detection characteristic thus, the Fe1 detects target with a higher precision than the other methods. These unstable characteristics mainly depend on localization of the window used to calculate features. In some frames, targets are located in the window but in other frames, they are not. This statistics makes Fe1 best in the first scenario.

The detection results of the feature based algorithms obtained from the second scenario are given in Table 3-28.

Table 3-28: The detection results of the Fe based algorithms obtained from the second scenario

Scenario 2	Algorithms	Number of detected pixels	Number of detected target pixels			R	P
			T1	T2	T3		
20th Frame	Fe1	2	0	1	1	15.38	100
	Fe2	99	4	4	5	100	13.14
	Fe3	8	0	0	0	0	0
100th Frame	Fe1	1	1	0	0	7.14	100
	Fe2	114	4	4	6	100	12.29
	Fe3	4	0	0	0	0	0
200th Frame	Fe1	2	0	0	0	0	0
	Fe2	102	7	9	12	100	27.46
	Fe3	4	4	0	0	14.28	100
300th Frame	Fe1	5	0	1	0	2.56	20
	Fe2	108	11	12	16	100	36.12
	Fe3	4	0	0	4	10.25	100
400th Frame	Fe1	3	1	0	0	1.51	33.34
	Fe2	171	23	20	23	100	38.6
	Fe3	4	0	0	0	0	0
500th Frame	Fe1	6	1	1	1	1.96	50
	Fe2	275	46	53	54	100	55.64
	Fe3	8	0	0	0	0	0

An examination of Table 3-28 indicates that the Fe2 detects too many background pixels as a target in all steps of the first scenario that results from the Otsu thresholding method, which separates target and background pixels. Both Fe1 and Fe3 detect the target with 100% precision twice in the second scenario. Probable targets are located inside the windows in these frames however, the mean precision values show that the Fe1 detects the target with highest precision compared with the other algorithms.

The detection results of feature based algorithms from the third scenario are given in Table 3-29.

Table 3-29: The detection results of Fe based algorithms obtained from the third scenario

Scenario 3	Algorithms	Number of detected pixels	Number of detected target pixels			R	P
			T1	T2	T3		
20th Frame	Fe1	1	0	1	0	12.5	100
	Fe2	10928	2	2	0	50	0.04
	Fe3	4	0	0	0	0	0
100th Frame	Fe1	2	0	1	1	28.57	100
	Fe2	12646	2	2	0	57.14	0.04
	Fe3	4	0	0	0	0	0
200th Frame	Fe1	2	0	1	0	7.142	50
	Fe2	13687	5	5	4	100	0.11
	Fe3	4	0	0	0	0	0
300th Frame	Fe1	1	0	1	0	5.26	100
	Fe2	13759	5	8	6	100	0.14
	Fe3	4	0	0	0	0	0
400th Frame	Fe1	4	0	0	0	0	0
	Fe2	11353	20	24	21	100	0.58
	Fe3	4	4	0	0	6.15	100
500th Frame	Fe1	4	0	0	1	0.73	25
	Fe2	10875	38	50	48	100	1.26
	Fe3	8	0	0	8	5.88	100

When the detection results obtained from the third scenario are analyzed, Fe2 and Fe3 have very high false alarms. The reason for the production of too many false alarms in Fe2 is related to the thresholding method. In addition, Fe3 cannot handle the difficulty in the third scenario because the features used in Fe3 cause the production of false alarms. On the other hand, the Fe1 detected targets with a 100%

precision four times in the third scenario thus it is the best algorithm in the third scenario.

The number of user-dependent parameter and processing times of feature based algorithms are given in Table 3-30 and this shows that Fe2 and Fe3 have two user-dependent parameters which have small and large window sizes which are really important for detection performance. The Fe1 has one user-dependent parameter that is a threshold constant. However, the fastest algorithm is Fe3. Fe1 and Fe2 are slower than Fe3 algorithm and works nearly same speed. The use of a different scale in Fe1 and the likelihood ratio calculation in Fe2 takes some time.

Table 3-30: The UDP and PT values of Fe based algorithms

Algorithms	**UDP**	**PT(s)**
Fe1	1	1.4
Fe2	2	1.5
Fe3	2	0.48

The overall performance results of the Fe based algorithms are given in Table 3-31.

Table 3-31: The overall performance comparison of the Fe based algorithms

		20th Frame		100th Frame		200th Frame		300th Frame		400th Frame		500th Frame							
Scenario	**Algorithms**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R Mean**	**R Std**	**P Mean**	**P Std**	**UDP**	**PT (s)**
Scenario 1	Fe1	10,52	100	6,25	100	0	0	5,55	40	3,63	66,67	0	100	4,325	4,037	67,78	41,19	1	1,4
	Fe2	100	19,59	100	13,68	100	27,09	100	34,62	100	33,75	100	54	100	0	30,46	14,09	2	1,5
	Fe3	0	0	0	0	15,38	100	22,22	100	0	0	5,55	100	7,192	9,495	50	54,77	2	0,48
Scenario 2	Fe1	15,38	100	7,14	100	0	0	2,56	20	1,51	33,34	1,96	50	4,758	5,733	50,56	41,66	1	1,4
	Fe2	100	13,14	100	12,29	100	27,46	100	36,12	100	38,6	100	55,64	100	0	30,54	16,56	2	1,5
	Fe3	0	0	0	0	14,28	100	10,25	100	0	0	0	0	4,088	6,461	33,33	51,64	2	0,48
Scenario 3	Fe1	12,5	100	28,57	100	7,14	50	5,26	100	0	0	0,735	25	9,034	10,6	62,5	44,02	1	1,4
	Fe2	50	0,04	57,14	0,04	100	0,11	100	0,14	100	0,58	100	1,26	84,52	24,08	0,362	0,485	2	1,5
	Fe3	0	0	0	0	0	0	0	0	6,15	100	5,88	100	2,005	3,107	33,33	51,64	2	0,48

The lack of a pre-processing stage in the algorithms resulted in low precision values in all scenarios. However, Fe1 has a better detection performance than others. This is related to the extraction of features in different scales. Fe1 extracts features in three scales and then fuses them with the results of different scales. This approximation manages the target size enlargement. However, the lack of a pre-processing stage creates difficulties in obtaining high precision values. In addition, it needs just one user-dependent parameter that is a threshold constant. However, processing time is a little high due to the calculation process being on different scales. The Fe2 and Fe3 algorithms use fixed window sizes to calculate features, which results in the target not being located inside the windows. These need two user-dependent parameters which directly affect the detection performance.

3.5.7 Overall Performance Analysis

Algorithms belonging to different approaches are compared in previous section separately and some algorithms from each approach are determined as the best within their group. The algorithms that produce best detection performances are considered in this section and are given in Table 3-32.

Table 3-32: The comparison of the best algorithms from each group

		20th frame		100th frame		200th frame		300th frame		400th frame		500th frame							
Scenario	Algorithms	R	P	R	P	R	P	R	P	R	P	R	P	R Mean	R Std	P Mean	P Std	UDP	PT (s)
Scenario 1	MO4	94	100	93,75	94,2	100	100	94,4	100	92,7	100	15,2	100	81,675	32,668	99,03	2,368	1	0,1302
	MO6	100	100	100	95	96,16	92,6	100	92,4	90,1	92,6	56,3	1,35	90,427	17,161	78,99	38,15	2	0.0159
	WT1	31	75	37,5	99,75	2,82	87,5	27,7	100	20	100	6,9	100	20,987	13,766	93,71	10,43	1	0,078
	Fi1	31	100	37,5	94,2	2,82	96,3	27,7	100	20	78,5	6,9	90,4	20,987	13,766	93,23	8,087	3	0.003
	S3	42,1	61,54	75	70,59	38,46	83,4	52,77	79,17	30,9	62,97	8,33	57,15	41,26	22,231	69,14	10,45	2	0,12
	Fe1	10,52	100	6,25	100	0	0	5,55	40	3,63	66,67	0	100	4,325	4,0368	67,78	41,19	1	1,4
Scenario 2	MO4	7,1	100	64,2	100	42,8	100	51,2	100	68,1	100	43,13	100	46,088	21,81	100	0	1	0,1302
	MO6	100	100	92,87	100	57,14	100	48,71	100	63,6	100	47,5	100	68,303	22,679	100	0	2	0.0159
	WT1	28,5	100	28,5	100	10,7	75	12,8	100	15,9	100	9,1	100	17,583	8,7568	95,83	10,21	1	0,078
	Fi1	28,5	100	28,5	100	10,7	100	12,8	100	15,9	100	9,1	100	17,583	8,7568	100	0	3	0.003
	S3	61,53	100	71,42	90,91	25	100	38,46	100	19,69	100	7,8	100	37,317	24,837	98,49	3,711	2	0,12
	Fe1	15,38	100	7,14	100	0	0	2,56	20	1,51	33,34	1,96	50	4,7583	5,7332	50,56	41,66	1	1,4
Scenario 3	MO4	62,5	100	71,4	100	85,71	85,72	78,94	79	44,61	90,7	61,7	100	67,477	14,562	92,57	8,946	1	0,1302
	MO6	66,7	100	85,7	100	14,28	100	36,8	100	16,9	100	20,5	100	40,147	29,62	100	0	2	0.0159
	WT1	37,5	100	42,8	100	14,28	100	21	57,15	18,46	92,31	8,8	70,6	23,807	13,421	86,68	18,41	1	0,078
	Fi1	37,5	100	42,8	100	14,28	100	21	100	18,46	100	8,8	100	23,807	13,421	100	0	3	0.003
	S3	62,5	71,43	85,71	100	14,28	100	15,78	100	7,69	100	1,4	100	31,227	34,397	95,24	11,66	2	0,12
	Fe1	12,5	100	28,57	100	7,14	50	5,26	100	0	0	0,735	25	9,0342	10,602	62,5	44,02	1	1,4

In the first scenario, the MO4, WT1 and Fi1 algorithms produce the best detection results. These three algorithms have a common property. They use the first type detection approach and background estimation is primarily performed. It is observed that background estimation is a very effective approach in this scenario. MO4 is the best of the three algorithms however, its processing time is higher than the others.

In the second scenario, all the algorithms except Fe1 produce good detection results. It is observed that both two type detection groups are useful in this scenario. However, only the S3 algorithm uses the second type detection approach (directly extract target information). MO4, MO6 and Fi1 algorithms detect targets with 100% precision and this situation makes them the best choice for the second scenario. However, the processing time of MO4 is high and MO6 and Fi1 user-dependent parameter numbers are high for these reasons, no algorithm is determined as the best when all performances are considered in the second scenario.

In the third scenario, the MO4, MO6, Fi1 and S3 algorithms are observed to be the best. The MO6 and Fi1 algorithms detect the target with 100% precision however, their user-dependent parameter need affects the overall performance in negative way. Thus, there is no best algorithm which is located in literature in the third scenario

CHAPTER 4

THE ANALYSIS OF PRE-PROCESSING, THRESHOLDING AND POST-PROCESSING METHODS

In Section 3.3, algorithms obtained from the literature review are compared. A reliable automatic target detection (ATD) algorithm should detect the target with 100% precision. In addition, the processing time of an algorithm should be short and the algorithm should not depend on user-dependent parameter. However, none of the algorithms in this study displayed high performance in all scenarios. The comparison of algorithms shows that pre-processing, thresholding and post-processing methods have a significant effect on the detection performance. Algorithms that use effective pre-processing methods have higher detection performance. Thresholding method also has an important role on the performance. For example, Otsu thresholding method does not produce good detection results in non-bimodal histograms. Furthermore, the number of user-dependent parameters used in thresholding method also affects the performance of the thresholding method since it generally varies from one scenario to another. Post-processing methods usually reduce the false alarm. However, they are only feasible if their processing time is shorter.

To construct an algorithm, which meets the desired specifications, each step of the process is investigated in detail by performing tests on all pre-processing, thresholding and post-processing methods used in algorithms. These methods are detailed in Chapter 3. In the current section, first pre-processing, thresholding and post-processing methods are briefly summarized. Then, different combinations of these methods are applied to test images to select the best method for each step. To determine the best detection method, different detection methods are added to the best combination. In this research, methods are compared based on their detection performance since it shows the effectiveness of the steps on the overall detection performance.

4.1 Pre-Processing Methods

Pre-processing methods are considerably important in ATD since they reduce false alarms. Algorithms using the pre-processing method achieve good detection performance. In this section, pre-processing methods are grouped into local and global enhancement approximations, each with different performance on the detection of target.

The first pre-processing method, which will be called Prep1 for comparative purposes, uses global enhancement [16]. In this method, the mean value of the image is first calculated, and then subtracted from the pixels of the whole image (Equation (3.30)). Further image enhancement is applied by computing the n^{th} power of the original intensity value. This way, if n is greater than 1, dark intensity values are suppressed and light intensity values are emphasized. The exact value of the power is left as a parameter, which is set manually. However, setting this parameter is very problematic and enhancement of noise cannot be avoided due to global processing.

The second pre-processing method called Prep2 uses both local and global enhancement approximations [4]. Assuming that the target has a higher intensity value than the background, the kernel in Equation ((3.7)) is used. The filter size is manually determined. Before filtering, median filtering is used to suppress the noise in the image.

The third pre-processing method, Prep3, uses local enhancement [7]. It is block-based processing in which the image is divided into blocks whose size is manually determined. Then, enhancement is locally performed subtracting the block mean value from the block pixels. If the target pixels have a higher gray level intensity than other pixels in the block, they are emphasized. Following local enhancement, thresholding is applied where the threshold value, α, is automatically calculated based on the image content. The calculation process of α involves the use of three interconnected equations (Equation (3.11)), where μ_b and σ_b represent the mean and standard deviation of the background, respectively. P_f gives the false alarm rate. The value of α is determined using 10^{-3} as the false alarm rate. The α value is iteratively changed to obtain the P_f value.

The fourth pre-processing method uses local enhancement [15]. This method, called Prep4, assumes that the pixels in the same row are exposed to the same atmospheric condition. Thus, the mean of the pixels in the same row is computed and subtracted from the row pixels (Equation (3.27)).

The fifth pre-processing method, Prep5, is related with Top-Hat transform (THT). Background is suppressed and the target is enhanced using the following mathematical morphological processes; erosion (Equation (3.1)), dilation (Equation (3.2)), opening (Equation (3.3)) and finally the THT (Equation (3.5)) where the structuring element *s* is used. Using THT, a smooth version of the image, namely the background estimation, is obtained. When this estimation is subtracted from the original image, the remaining emphasized regions are the possible target regions.

4.2 Thresholding Methods

Thresholding is the most important part of target detection. The threshold value should be chosen without knowing the target properties. There are some simple approaches to decide on a threshold value.

The first thresholding method is called Thresh1 [7], in which threshold value is calculated using minimum and maximum gray level pixel values in the image (Equation (3.12)). This approach has no manually set parameter but assumes an intensity range for possible target pixels.

The second thresholding method, Thresh2, is a well-known Otsu method that automatically finds a threshold to segment the image into two regions by considering all image intensities [5][8][12][24]. This method treats the image histogram as bimodal and automatically finds the separation point of these two peaks in the histogram.

The third thresholding method, called Thresh3 [4], has an iterative characteristic. In the first iteration, the mean value (E) is calculated and pixels with values below this value are assigned the mean value (Equation (3.8)). In the second cycle, the mean is computed again and the pixels lower than the new mean value are assigned to zero and others (those higher than the mean value) are assigned to one, namely the target (Equation (3.9)).

The fourth thresholding method, Thresh4 [6], is another iterative method, in which the threshold value is adjusted to match the number of detected pixels and the number of the actual target pixels (T0), which needs to be manually set as a parameter. This method requires several other parameters.

The final thresholding method, Thresh5 [11] is based on image statistics (Equation (3.22)) namely the mean (m) and standard deviation (σ) of the image (I). Here, a user defined constant, c, is needed.

4.3 Post-Processing Methods

After detecting possible target pixels, post-processing can be applied to eliminate noise, i.e. false alarms, and to smooth the resulting target regions. Therefore, the post-processing step is very important for the overall detection performance. There are two post-processing methods obtained from the literature research [8][17].

The first post-processing method is called PostP1 [8], which uses a Robinson Guard Filter (RGF) to reduce false alarms. A response of a 7x7 RGF kernel is given in Equation (3.14). RGF has nonlinear filtering characteristics. Equation (3.15) shows how to apply RG.

Another approach to post-processing is presented in [17] and called PostP2. The main assumption in PostP2 is that the still background can be estimated using the filtering methods but noise may not be estimated due to its random characteristics. Therefore, post-processing method intends to reduce the estimation error given in Equation (3.35), where $Y_0\ (x, y)$ represents the original infrared image, and $Y'\ (x, y)$ represents the estimated background. Estimation error $\varepsilon\ (x, y)$ is defined as the difference between the original image and the estimated background.

4.4 Evaluation of Pre-Processing, Thresholding and Post-Processing Methods

To compare the pre-processing, thresholding and post-processing methods, a generic method, i.e. Top-Hat transform (THT), is chosen as the detection method, and a 3x3 square is used as the structural element. Figure 4-1 presents this structure.

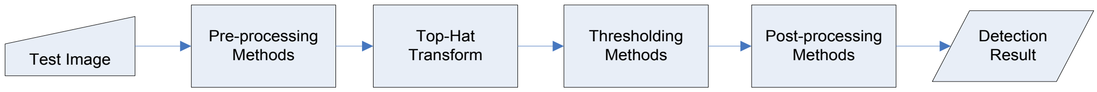


Figure 4-1: The order of the constructed system

First, detection results are obtained using the pre-processing method, THT and thresholding method. All combinations of pre-processing and thresholding methods are applied to the 20th frame of all scenarios in which targets are very small. Detection performance is presented using the recall and precision values. The calculation procedures of recall and precision are given in Section 3.5.

The detection results obtained from the first scenario are given in Table 4-1. Each row shows the result of a pre-processing method. The total detected number of pixels, recall and precision values are presented in separate columns for each thresholding method. Mean values of precision and recall are also given in the table for each pre-processing and thresholding method.

Table 4-1: The detection results obtained from the 20th frame of the first scenario

Scenario 1 20. Frame		Thresh1			Thresh2			Thresh3			Thresh4			Thresh5				
		Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Mean R	Mean P
	Prep1	11	57,89	100	20457	100	0,09	29826	100	0,06	20	100	95	15	78,94	100	87,366	59,03
	Prep2	9	47,36	100	3	15,7	100	58876	100	0,03	18	89,47	94,4	11330	100	0,16	70,506	58,926
	Prep3	11	57,89	100	11	57,89	100	17	89,47	100	15	78,97	100	16	84,21	100	73,686	100
	Prep4	11	57,89	100	36153	100	0,05	54707	100	0,03	19	94,73	94,7	13	68,42	100	84,208	58,962
	Prep5	11	57,89	100	24038	100	0,07	47655	100	0,03	18	78,97	83,3	14	73,68	100	82,108	56,686
	Mean Val		55,78	100		74,72	40,04		97,89	20		88,43	93,5		81,05	80,03		

Table 4-1 shows that Prep3 method can detect the target with 100% precision using any of the thresholding methods. In addition to that the mean recall value is around 74%. These results clearly indicate that the best pre-processing method in the first scenario is Prep3 in which the image is enhanced using the block-based approximation. Another important result presented in Table 4-1 is the high performance of Thresh1, which can detect the target with 100% precision in all pre-processing methods using the maximum and minimum value of image to calculate the threshold value. Furthermore, Thresh1 does not require any user-dependent parameters.

The detection results obtained from the second scenario are given in Table 4-2.

Table 4-2: The detection results obtained from the 20th frame of the second scenario

Scenario 2 20. Frame		Thresh1			Thresh2			Thresh3			Thresh4			Thresh5			Mean R	Mean P
		Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P		
	Prep1	11	84,61	84,6	11846	100	0,1	1747	100	0,74	12	84,61	91,7	11	84,61	100	90,766	55,422
	Prep2	44	15,9	53,8	2	7,69	50	31273	100	0,04	13	30,76	30,8	12	30,76	33,3	37,022	33,588
	Prep3	1	7,69	100	1	7,69	100	1	7,69	100	1	7,69	100	1	7,69	100	7,69	100
	Prep4	13	84,61	84,6	45367	100	0,02	11202	100	0,11	13	84,61	84,6	11	84,61	100	90,766	53,87
	Prep5	17	84,61	64,7	35068	100	0,03	10129	100	0,12	13	84,61	84,6	7	53,84	100	84,612	49,892
	Mean Val		55,48	77,6		63,08	30,03		81,54	20,2		58,46	78,3		52,3	86,66		

In the second scenario, Prep3 method also detects the target with 100% precision using any of the thresholding methods. However, the mean recall value is reduced to approximately 7%, which can be explained by the difficulty of the scenario. Furthermore, the calculated mean recall value is significantly smaller compared to the other methods. However, a 100% mean precision value is more important than the low recall value since other methods produce false alarms. On the other hand, the best detection performance is achieved using Thresh5 regardless of the pre-processing method. Thresh5 uses image statistics to calculate the threshold value. However, the mean P value obtained from Thresh5 does not reach 100% and still produces false alarms. The results from the second scenario indicate that block-based enhancement is a very effective method for the pre-processing stage.

The detection results obtained from the third scenario are given in Table 4-3.

Table 4-3: The detection results obtained from the 20th frame of the third scenario

Scenario 3 20. Frame	Thresh1			Thresh2			Thresh3			Thresh4			Thresh5				
	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Mean R	Mean P
Prep1	1026	75	0,68	62140	100	0,02	43810	100	0,02	12	50	33,3	4	50	100	75	26,804
Prep2	2296	25	0	458	0	0	66712	100	0,01	20	0	0	12	61,53	66,6	37,306	13,3222
Prep3	5	62,5	100	5	62,5	100	5	62,5	100	5	62,5	100	5	62,5	100	62,5	100
Prep4	1338	100	0,97	74195	100	0,01	74195	100	0,01	10	23,07	30	7	25	28,57	69,614	11,912
Prep5	746	100	1,74	64736	100	0,02	64736	100	0,02	11	46,13	54,5	4	50	100	79,226	31,256
Mean Val		72,5	20,7		72,5	20,01		92,5	20		36,34	43,6		49,81	79,03		

In the third scenario, Prep3 method still achieves 100% mean precision using any of the thresholding methods. The mean recall value obtained with Prep3 is approximately 62%, which indicates that block-based enhancement can estimate large background regions. In the third scenario, Prep3 produces better detection performance than in the second scenario. In terms of the thresholding methods, the best performance is achieved with Thresh5, which calculates threshold value using the mean and standard deviation of the image. However, Thresh5 does not have 100% precision.

The analysis given above shows that Prep3 method produces the best detection performance. This can be attributed to the use of local enhancement, i.e. the windowing structure. In particular, the calculation of the mean and performing enhancement in each window results in an increase in the accuracy. Other methods attempt to remove the clutter using global approximations. In addition, the Thresh1 method produces the highest precision values using the Prep2 method without the need to use any user-dependent parameters.

Based on these results, Prep3 and Thresh1 methods are chosen for the comparison of the post-processing methods in this study. These methods are used in the MO4 algorithm, which produces the highest detection performance and identified as the best algorithm in section 3.5.7. Prep3 and Thresh1 are applied to different frames, which are recorded at different distances from the targets. Table 4-4 presents all the detection results obtained with Prep3 (local windowing enhancement) and Thresh1 (maximum and minimum value) methods for different frames (20th, 100th, 200th, 300th, 400, and 500th) in all three scenarios. This time, the performance of these methods is investigated with respect to the apparent sizes of the targets. The 20th frame is the farthest view and the 500th frame is the closest view of the targets.

Table 4-4: The detection results of Prep3 and Thresh1 methods for different frames of all scenarios.

		20th Frame			100th Frame			200th Frame			300th Frame			400th Frame			500th Frame				
		Total Detected Pixels	R	P	Total Detected Pixels	R	P	Total Detected Pixels	R	P	Total Detected Pixels	R	P	Total Detected Pixels	R	P	Total Detected Pixels	R	P	Mean R	Mean P
Scenario 1	Prep3 + Thresh1	11	57,9	100	15	93,75	100	20	76,9	100	30	83,33	100	2	3,63	100	11	7,63	100	53,8583	100
Scenario 2		1	7,69	100	9	64,28	100	11	39,3	100	18	46,15	100	30	45,4	100	13	8,49	100	35,215	100
Scenario 3		5	62,5	100	4	57,14	100	11	78,6	100	15	63,15	80	19	29,23	100	75	55,14	100	57,6217	96,6667

Table 4-4 shows that in all distances of the first two scenarios, targets are detected with 100% precision. However, in the last scenario, very few false alarms are produced and the precision is reduced to 96%. Producing 100% precision value in the first and second scenarios is very important step for automatic target detection (ATD). However, some false alarms are produced in the third scenario. In addition, mean recall values are around 50%. To construct a robust ATD algorithm, this value should be increased to higher level. The reduction in the performance in the third scenario indicates the need for a post-processing stage.

To decide on post-processing approach, various post-processing methods are compared using Prep3 and Thresh1 as pre-processing and thresholding methods, respectively. Prep3 method enhances image with block based approach and Thresh1 calculates threshold value by using maximum and minimum intensity values of the image. The detection results obtained with Prep3+Thresh1+PospP1 and Prep3+Thresh1+PospP2 methods in the first scenario are given in Table 4-5.

Table 4-5: The detection results of different post-processing methods obtained from different frames in the first scenario

Scenario1																			
	20th Frame			100th Frame			200th Frame			300th Frame			400th Frame			500th Frame				
	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Mean R	Mean P
Prep3 Thresh1 PostP1	11	57,89	100	15	93,75	100	17	65,38	100	21	58,33	100	2	3,63	100	11	7,63	100	47,768	100
Prep3 Thresh1 PostP2	14	73,68	100	15	93,75	100	24	92,3	100	32	88,8	100	2	3,63	100	1	0,69	100	58,808	100

The results obtained with post-processing methods show that the calculated mean precision values are still 100%. This means that no background pixels are detected as the target. However, the first post-processing method decreases the mean recall value due to its clutter-rejection characteristics whereas the second method increases the mean recall value. This result is meaningful since it indicates that the second post-processing method estimates the background more accurately and results in the suppression of the background and an increase in the intensity value of the target pixels.

The same pre-processing, thresholding and post-processing combinations are applied to the second scenario. The results of the calculated detection are given in Table 4-6.

Table 4-6: The detection results of different post-processing methods obtained from different frames in the second scenario

	Scenario2																
	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Mean R	Mean P
Prep3 Thresh1 PostP1	1	7,69	100	9	64,28	100	11	39,28	100	18	46,15	100	18	27,77	100	9	5,88	100	31,842	100
Prep3 Thresh1 PostP2	1	7,69	100	9	64,28	100	11	39,28	100	18	46,15	100	26	39,39	100	132	86,27	100	60,383	100

The results are similar for the second scenario. The mean recall value is decreased by the first post-processing method and increased by the second method. Mean precision values obtained with both post-processing methods are still 100%. The characteristics of these two methods have the same effect on the images in the second scenario.

Finally, all the combinations are applied to the third scenario. The results of the calculated detection are given in Table 4-7.

Table 4-7: The detection results of different post-processing methods obtained from different frames in the third scenario

Scenario3																			
	20th Frame			100th Frame			200th Frame			300th Frame			400th Frame			500th Frame				
	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Tot. Detected Pixels	R	P	Mean R	Mean P
Prep3 Thresh1 PostP1	5	62,5	100	4	57,14	100	10	71,42	100	14	57,89	78,57	12	18,46	100	10	7,35	100	45,79333	96,42833
Prep3 Thresh1 PostP2	5	62,5	100	4	57,14	100	15	85,71	80	24	73,68	58,33	22	33,84	100	112	82,35	100	69,4025	89,72167

The results of the third scenario are different from those of the first and second scenarios. Even though the effect of the first and second post-processing methods on the mean recall value is the same as before, the mean precision values obtained from both methods decrease in the third scenario contrary to the previous scenarios. Furthermore, the calculated mean precision value of the first post-processing method is higher than that of the second method. These results indicate that the second post-processing method cannot estimate the background and the first post-processing method cannot reject all the clutter.

According to the analysis given above, despite the low recall value in the third scenario, the precision value is higher, which shows that the first post-processing method is better. Mean precision values are very similar to those obtained with PostP1. However, in this thesis, the results of the detection are first evaluated using the precision values since producing no false alarms is much more valuable for an automatic target detection system. Furthermore, since the PostP2 method decreases the precision value more than the PostP1 method, the latter is identified as the better method. On the other hand, PostP1 does not fulfill the system requirements because it still produces some false alarms in the third scenario. A more effective post-processing method should possess the characteristics of both post-processing methods in terms of increasing the recall value and producing no false alarms.

4.5 Evaluation of the Overall Detection

The results obtained from the previous section indicate that Prep3, which enhances the image using a block-based approach, is the best pre-processing method while Thresh1 is considered the best thresholding method since it calculates the threshold value using the maximum and minimum intensity values of the image. Furthermore, Thresh1 does not need any user-dependent parameters. Finally, PostP1 is determined as the better post-processing method for increasing and maintaining the precision value in the first two scenarios. However, due to its clutter rejection characteristic, PostP1 decreases the recall values in the third scenario.

In this section, the best detection method is identified by combining the best pre-processing, thresholding and post-processing methods with the six detection methods identified as having significant detection performance. These methods involve the use of Top-Hat transform (THT), wavelet transform (WT), filtering and saliency. The details of these detection methods are given in Chapter 3.

The first detection method is Det1, in which THT is applied to the image [7]. Erosion and dilation are applied to the image using a user defined structural element to estimate the background. Then, the estimated background is subtracted from the original image to enhance the target.

The second detection method, Det2, also uses THT [9]. In Det2, THT is applied to the image using a grid SE.

The third detection method, called Det3 uses WT [11]. The first and second order WTs are extracted from the image to obtain horizontal and vertical components. Then, the extracted components are fused together to detect the target in Det3.

The fourth detection method is called Det4 [17]. In this method, a filtering approach is used and the image is filtered using a mean filter to obtain the estimated background, which is then subtracted from the original image.

The fifth detection method is Det5, in which saliency detection is used [21]. Fourier transform is used to detect the salient regions in the image.

The last method is called Det6. In this method, discrete cosine transform is used to detect the salient region [21] [22].

The detection results of the best combination with selected detection methods in the first scenario are given in Table 4-8.

Table 4-8: The detection performance of all detection methods combined with the best in the first scenario

	Scenario1													
	20th Frame		100th Frame		200th Frame		300th Frame		400th Frame		500th Frame			
	R	P	R	P	R	P	R	P	R	P	R	P	R Mean	P Mean
Det1 (THT based)	57,8	100	93,8	100	65,4	100	58,3	100	3,63	100	7,63	100	47,753	100
Det2 (THT based)	57,8	100	93,8	100	65,4	100	58,3	100	3,63	100	7,63	100	47,753	100
Det3 (WT based)	36,8	100	62,5	100	19,2	100	5,55	100	9,09	100	2,77	100	22,663	100
Det4 (Fi based)	57,8	100	75	100	61,5	100	52,8	100	3,63	100	6,25	100	42,83	100
Det5 (S based)	26,3	100	56,3	100	30,8	100	19,4	100	3,63	100	2,08	100	23,078	100
Det6 (S based)	26,3	100	56,3	100	26,9	100	27,8	100	3,63	100	2,08	100	23,827	100

The detection results show that all combinations produce 100% precision. On the other hand, Top-Hat transform (THT) based (Det1 and Det2) and filtering based (Det4) methods have recall values of up to 40%. Wavelet transform (WT) and saliency based methods did not produce high recall values. THT and filtering based methods are included in the first type detection group, which first estimates the background, and then subtract it to enhance the target regions. The second type detection group consists of WT and saliency based methods, which directly detect target features. The results obtained from the first scenario shows that the THT and Fi-based methods are more successful. This means that if the targets have a higher intensity than the background, then the first type detection methods will perform better. Table 4-9 presents the detection results of the best combinations of the selected detection methods.

Table 4-9: The detection performance of the best combinations of all detection methods in the second scenario

	Scenario2													
	20th Frame		100th Frame		200th Frame		300th Frame		400th Frame		500th Frame			
	R	P	R	P	R	P	R	P	R	P	R	P	R Mean	P Mean
Det1 (THT based)	7,69	100	64,3	100	39,3	100	46,2	100	27,3	100	5,88	100	31,758	100
Det2 (THT based)	7,69	100	64,3	100	39,3	100	46,2	100	27,3	100	5,88	100	31,758	100
Det3 (WT based)	15,4	100	14,3	100	14,3	100	7,69	100	9,09	100	3,26	100	10,663	100
Det4 (Fi based)	15,4	100	14,3	100	39,3	100	46,2	100	25,8	100	6,53	100	24,562	100
Det5 (S based)	15,4	100	14,3	100	25	100	30,8	100	25,8	100	6,53	100	19,617	100
Det6 (S based)	15,4	100	78,6	100	32,1	100	30,8	100	18,1	100	3,92	100	29,812	100

When Table 4-9 is examined, it can be seen that none of the combinations produce false alarms. In addition, the recall values of THT based methods are as close as

30%. The saliency based method has a very similar recall value to that of the THT-based methods. The filtering based method is also included in the best group. These results show that both type detection approaches work well in the second scenario but the THT based methods of the first detection type are the best ones. This indicates that THT can still estimate the background under low contrast. On the other hand, WT and Fourier transform cannot detect the target due to the low contrast.

The detection results of the best combinations of the selected detection methods in the third scenario are given in Table 4-10.

Table 4-10: The detection performance of the best combinations of all detection methods in the third scenario

	Scenario3													
	20th Frame		100th Frame		200th Frame		300th Frame		400th Frame		500th Frame			
	R	P	R	P	R	P	R	P	R	P	R	P	R Mean	P Mean
Det1 (THT based)	62,5	100	57,1	100	71,4	100	57,9	78,57	18,5	100	7,35	100	45,793	96,428
Det2 (THT based)	62,5	100	57,1	100	71,4	100	57,9	78,57	18,5	100	7,35	100	45,793	96,428
Det3 (WT based)	100	57,1	57,1	50	7,14	100	26,3	100	9,23	100	5,14	100	34,16	84,523
Det4 (Fi based)	62,5	100	57,1	100	71,4	100	52,6	100	16,9	100	7,35	100	44,66	100
Det5 (S based)	62,5	100	0	0	35,7	83,3	21,1	80	9,23	100	1,47	100	21,66	77,222
Det6 (S based)	50	100	57,1	100	42,9	85,7	36,8	87,5	12,3	100	1,47	100	33,433	95,535

The results of the third scenario show that none of the combinations except the filtering based method can detect the targets without producing false alarms. This is probably due to the difficulty of this scenario. Following the filtering based method, the highest precision and recall values are obtained by the THT based methods. The mean precision values of THT and filtering based methods are very close to each other. In addition, the performance of the saliency-based Det6 method is similar to that of the Fi methods.

According to the results from all scenarios, the best detection method is Det4 (filtering based), which produces a 100% precision value and significant mean recall values in all scenarios. The filtering based Det4 method estimates the background of the image using a mean filter and subtracts it from the original image to enhance the target regions. This kind of detection is grouped under the first type (background estimation). In this case, a simple method such as mean filtering may be enough to obtain a good detection result when using the best pre-processing, thresholding and post-processing methods.

CHAPTER 5

THE PROPOSED POST-PROCESSING METHOD

5.1 Introduction

The analysis made in the previous sections indicates the need for developing a method to increase the recall values. A robust and reliable automatic target detection algorithm should produce 100% precision and very high recall values. After obtaining the best results with a combination of the selected pre-processing, detection, thresholding and post-processing methods, detection precision is calculated as 100%. However, recall values are not as high as the precision values. This reduces the performance of the subsystem that is going to be used following the target detection. For example, a target recognition algorithm needs the whole target region to recognize the target. In addition, a target-tracking algorithm also requires the whole target region; otherwise it can miss the target in the subsequent frames.

In this chapter, a new post-processing method is proposed to increase the recall values. This method maintains the precision values at 100% while increasing the recall value. In the proposed method, all the detected target regions are first analyzed to eliminate the false alarms, then, individually thresholded to increase the recall value. The effectiveness of the proposed method is presented based on its performance on the test scenarios.

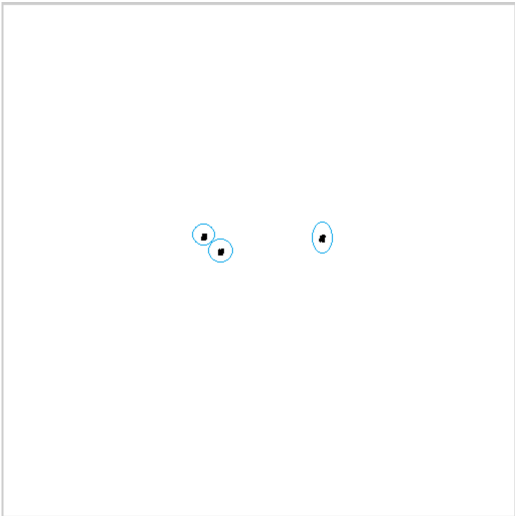
5.2 Proposed Method

The proposed post-processing method is two-fold: the first one concerns precision and the second one is related to recall. In the first part, to maintain a high detection precision value, detected pixel regions are analyzed by identifying the detected pixels as the target or the background based on their characteristics. These characteristics depend on the contrast level of the target and background. The contrast can be calculated using standard deviation. Standard deviation of the target regions are usually calculated as high due to the inclusion of the background and target pixels. On the other hand, standard deviation is low in background regions since background pixels are usually similar to each other. After calculating the standard deviation of each candidate target region, some regions are eliminated in relation to the contrast level of the whole image. A target region with a lower standard deviation than this threshold is eliminated whereas a target region with a higher standard deviation is retained.

The second part of the proposed method aims to increase the recall values. To this end, each detected region is analyzed by doubling their size. After enlarging the target regions, the best thresholding algorithm identified in the previous section is applied to these regions individually. The use of the best thresholding method in each target region specifically increases the accuracy of the thresholding method. In addition, enlarging the target region helps detect most of the target pixels.

Figure 5-1 presents an example of the detection performance of the proposed post-processing method. Minimum and maximum values of Figure 5-1 (b) and (c) are represented with white-black to improve the visibility. The detection result of the combination of the best pre-processing, detection and thresholding methods are given in Figure 5-1 (b). As indicated by the detected pixels, this combination is not sufficient to detect the whole target region. However, the use of the proposed post-processing method increases the recall value (detected target pixels).

(a) (b)

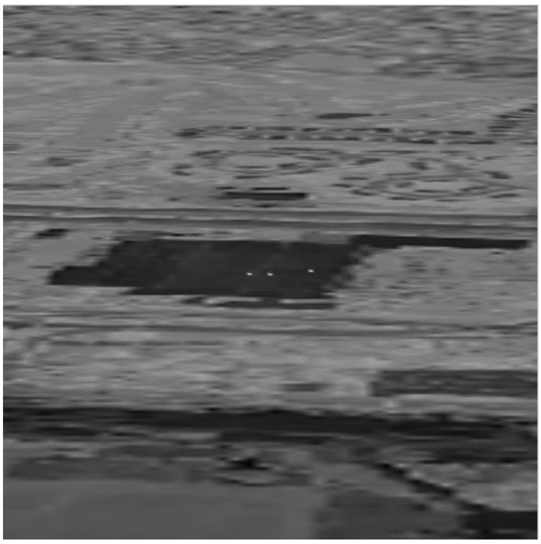


(c)

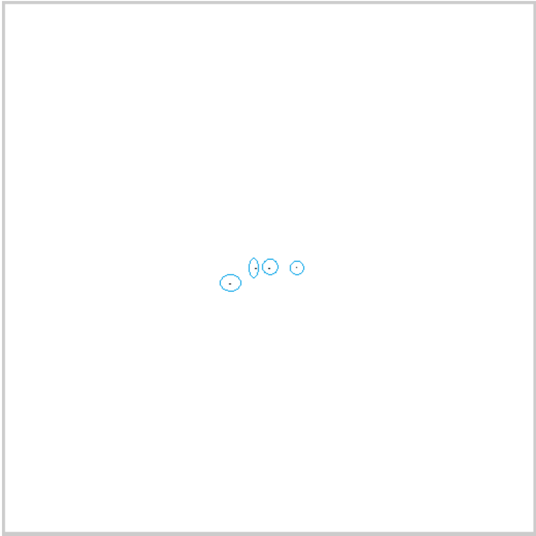
Figure 5-1: The detection performance of the proposed post-processing method.

a) Original infrared image, b) Detection result of combined pre-processing, detection and thresholding methods (targets are encircled), c) Detection result of the proposed method (targets are encircled).

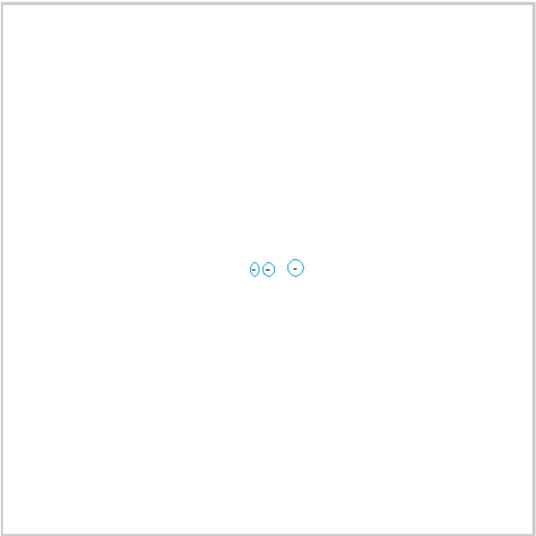
Figure 5-2 presents another example, in which there are some false alarms, which decreases the detection performance [Figure 5-2 (b)]. Color scale is converted from black-white to white-black in Figure 5-2 (b) and (c) for minimum-maximum values to increase the visibility. Using the proposed post-processing method eliminates the false alarms and increases the precision value.



(a)



(b)



(c)

Figure 5-2: The detection performance of the proposed post-processing method.

a) Original infrared image, b) Detection result of the combination of pre-processing, detection and thresholding methods (targets are encircled), c) Detection result of the proposed method (targets are encircled).

5.3 Detection Performance of the Proposed Method

This section presents in detail the detection performance of the proposed method based on the results of the test scenarios. The proposed post-processing method is added to the best combination of the pre-processing, detection and thresholding methods identified in the previous section. The resulting algorithm is applied to the test images. Table 5-1 presents the detection results.

Table 5-1: The detection performance of the proposed post-processing method in all test scenarios

	20th Frame		100th Frame		200th Frame		300th Frame		400th Frame		500th Frame		Proposed Method		Best Algorithm		Enhancement Percentage (%)	
	R	P	R	P	R	P	R	P	R	P	R	P	R Mean	P Mean	R Mean	P Mean	R	P
Scenario1	100	100	100	100	100	100	100	100	83,63	100	81,94	100	94,26	100	42,83	100	120	0
Scenario2	23,07	100	100	100	71,42	100	51,28	100	77,27	100	70,32	100	65,56	100	24,56	100	166	0
Scenario3	100	100	100	100	100	100	94,73	100	76,92	100	77,2	100	91,48	100	44,66	100	104	0

When Table 5-1 is analyzed, it can be seen that the proposed post-processing method increases the mean recall values. In addition, mean precision values are still 100%, which means that there are no false alarms. The mean recall value is calculated to be low as in the second scenario. In addition, there is an inconsistency in the recall values of the second scenario compared to other scenarios. This may be related to the pre-processing method, which sometimes eliminates the targets while at other times enhances the target regions. This may reduce the recall value.

The results show the effectiveness of the proposed post-processing method. Using the post-proposed method, an automatic target detection algorithm can detect target with 100% precision and a very high recall value.

CHAPTER 6

CONCLUSION

6.1 Summary

In this thesis work, the problem of automatic target detection of very small targets is considered. The main purpose is to construct a robust, reliable and fast automatic target detection algorithm, which can detect targets from different ranges and various cluttered backgrounds.

Many different algorithms have been used in the literature for automatic target detection. These algorithms are separated from each other mainly by the methods used in their detection stages. A typical automatic target detection algorithm consists of three stages, which are pre-processing, detection and post-processing, all of which have very important roles on the overall target detection performance. Images are prepared to detection stage by pre-processing methods. These methods usually intend to reduce noise in image and to enhance target regions. Detection stage is another important stage where target region in the image is extracted. Two different detection approaches are used in the literature. In one approach, background is estimated initially. Then by subtracting the estimated background from the original image, target regions are obtained. In the second approach, methods directly deal with target and they intend to obtain target information without using any background estimation procedure. Detection stage is generally finalized with thresholding where targets are separated from the background. Determination of the most appropriate threshold value is a highly critical decision, which directly affects the overall algorithm performance. Final stage of automatic target detection is post-processing. Generally, false alarms, which are produced in the detection stage, are eliminated in this stage.

Several pre-processing methods are used in automatic target detection algorithms. They are grouped into two as local and global approaches. Their performances are different from each other and local approaches are better than global approaches. Generally, filtering such as mean, median filter is applied to reduce noise in the image. This step is very critical for overall system because detection method performance highly depends on the capability of the pre-processing method.

In order to detect target with high performance, detection method should be chosen carefully. There are a lot of detection methods in the literature. Especially, morphological operation (MO) based algorithms which are detailed in Section 3.3.1 have the key position. Different versions of MO based algorithms are presented in [4]-[9]. These methods are grouped into first detection approach. They detect target by estimating background of the image and a structural element is used for this purpose. Background estimation performance is highly dependent on the selection of a suitable structural element. Another important detection method is Wavelet transform (WT). The methods are given in Section 3.3.2 [10]-[14]. WT is used in both of the two detection approaches. Background as well as the horizontal, vertical and diagonal information about the target can be extracted by using WT. Background of image is estimated by reconstructing approximation information of WT. On the other hand, different scale WT information is brought together to extract target information. In addition to these methods, edge detection is also used to detect target in the literature [15][16]. For this purpose, gradient based and discrete cosine transform based methods are presented for edge detection (Section 3.3.3). Furthermore, filtering based methods which are given in Section 3.3.4 are presented in literature [17][18]. These methods use first detection approach. They intend to estimate background of image by using filters such as mean, median, max-mean and max-median. Saliency based methods are also very popular to detect target [19]-[22]. They are grouped into second detection approach. They aim to directly detect saliency information in image by using Fourier transform, second order directional derivative and discrete cosine transform (Section 3.3.5). Another group of detection methods used in the literature is feature based methods [23]-[25]. They are given in Section 3.3.6. Feature based methods also use second detection approach. They use intensity, mean, standard deviation and gradient as features in image to detect target directly.

For post-processing, two methods, which have very different characteristics, are mentioned in the literature. One of them uses Robinson Guard non-linear filter to eliminate false alarms. It has clutter rejection characteristic and generally aims to increase the precision value. Second method aims to improve background estimation quality and directly affects the recall value. Improved background estimation directly increases the recall value.

Automatic target detection algorithms are tested using three different scenarios. The characteristics of these test scenarios are defined based on a literature survey. Test scenarios are constructed synthetically by using Infrared Scene Generation System, which generates infrared scenes of real environments. Three different background scenes are prepared which are different from each other in contrast level between targets and background. Furthermore, thermal surface and real tank models are used as targets in the scenarios. In addition to that, scenarios are recorded as consecutive frame series. Thus, target size and background clutter continuously change between consecutive frames.

All automatic target detection algorithms that are explained in Chapter 3 are applied to the test scenarios step by step and compared. Comparison of the algorithms' detection performances is done based on precision and recall values. Morphological operation based, wavelet transforms based, filtering based and saliency based algorithms have significant detection performances on the test scenarios. However, no algorithm can detect target with 100% precision in all scenarios, which means that some false alarms are always produced. In addition to that, the number of user-dependent parameters and processing time needs of the algorithms do not allow us to determine the best algorithm for automatic target detection. For this purpose, the methods used in each stage of automatic target detection algorithms are examined in detail. The best performing method of preprocessing, thresholding and post processing stages are determined using a generic detection method (Top-Hat transform). Block based image enhancement [7] is selected as the best pre-processing method. The thresholding method which requires no user-dependent parameter [7] is determined as the best thresholding method. This method calculates the threshold value using maximum and minimum intensity values of the image. Finally, the method, which uses Robinson Guard non-linear filtering process [8] to eliminate false alarms, is selected as the best post-processing approach.

After determining best pre-processing, thresholding and post-processing methods, morphological operation and wavelet transform, filtering and saliency based detection methods are added to this best combination respectively. Such an application helps to select the best overall system of target detection. Finally, filtering based method [17] is selected as the best approach for the detection stage. A combination of the best pre-processing, detection, thresholding and post-processing methods can detect target with 100% precision. However, it produces low recall values around 36%.

To increase recall values, a new post-processing method is proposed. This method increases recall values while keeping precision values at 100% in prepared test scenarios. It has both clutter rejection and recall increasing characteristics. Results obtained by the proposed post-processing method increases the mean recall values up to 90% in the first and third scenarios. The mean recall value is increased to 60% in the second scenario. This value is lower than the first and third scenarios. This fact is related to the applied pre-processing method, which has really strong background suppression characteristic. As a result, very limited target information is sent to the post-processing stage, which can increase the mean recall value with this limited information.

The final automatic target detection system can detect very small ground targets in infrared images acquired from various ranges and background clutters (prepared test scenarios) with 100% precision and close to 90% recall, which are the highest values in the previous literature.

Some future work may be suggested for further improving the whole system. The use of detection results of consecutive frames can improve detection performance. In addition to that, this algorithm does not use any environmental information. The use of some information such as distance to the target, the number of target and background type may also increase the detection performance.

REFERENCES

[1] Jiawei Wu, Shiyi Mao, Xiaoping Wang, Tianxu Zhang, “Ship Target Detection and Tracking in Cluttered Infrared Imagery”, SPIE Optical Engineering 50(5), 057207 (May 2011)

[2] Hari Babu Srivastava, Vinod Kumar, H.K. Verma, and S.S. Sundaram, “Image Pre-processing Algorithms for Detection of Small/Point Airborne Targets”, Defence Science Journal, Vol. 59, No. 2, March 2009, pp. 166-174

[3] Alessandro Rossi, Nicola Acito, Marco Diani, Giovanni Corsini, “Experimental Analysis of Adaptive Clutter Removal Techniques in IR Target Detection Systems”, SPIE 2011 Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications VIII

[4] JIANG Feng, JIN Min-hui, SONG Li-Quan, “Small Infrared Target Detection Algorithm Based on Mathematical Morphology”, International Symposium on Photoelectronic Detection and Imaging 2011: Advances in Infrared Imaging and Applications SPIE 2011

[5] Rong Dang, Jingzhou Zhang, Yuan Zhang, Rui Yang, “Infrared Target Detection Based on Image Entropy”, Artificial Intelligence Management Science amd Electronic Commerce (AIMSEC), IEEE 2011

[6] Kai Wang, Yan Liu, Xiaowei Sun, “Small Moving Infrared Target Detection Algorithm under Low SNR Background”, 2009 Fifth International Conference on Information Assurance and Security, IEEE 2009

[7] Yao Yunping Zhang Wei Duan Chang, “An Infrared Small and Dim Target Detection Algorithm Based on the Mask Image”, The Tenth International

Conference on Electronic Measurement & Instruments (ICEMI), IEEE 2011

[8] Mina Farajzadeh, Ali Mahmoodi, Mohammad Reza Arvan, "Detection of Small Target Based on Morphological Filters", 20th Iranian Conference on Electrical Engineering, (ICEE), IEEE 2012

[9] Chen Yong, Yan Gao-shi, Huo Yu, "Morphological detection method with girdle structure", International Symposium on Photoelectronic Detection and Imaging 2013: Infrared Imaging and Applications, SPIE 2013

[10] Yuan Zhang, Jingzhou Zhang, Dong Wang, Chao Chen, "Infrared Small Target Detection Based On Morphology And Wavelet Transform", Artificial Intelligence Management Science amd Electronic Commerce (AIMSEC), IEEE 2011

[11] Wang Ting, Shenyuan Yang, "Weak and Small Infrared Target Automatic Detection Based on Wavelet Transform", Second International Symposium on Intelligent Information Technology Application, IEEE 2008

[12] Yu-Qiu Sun, Jin-Wen Tian, Jian Liu, "Background Suppression Based-On Wavelet Transformation To Detect Infrared Target", Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, IEEE 2005

[13] Xiao-Yan Fan, Bao-Chang Pan, Jian Liang,Yong-Hui Huang, "A New Preprocessing Algorithm For Detection Of A Small Dim Target In An Ir Image Sequence", Proceedings of the 2012 International Conference on Wavelet Analysis and Pattern Recognition, IEEE 2012

[14] WEI Ying, SHI Zelin, YU Haibin, "An Automatic Target Detection Algorithm Based on Wavelet Analysis for Infrared Image Small Target in Background of Sea and Sky", Acquisition, Tracking, and Pointing XVII, SPIE 2003

[15] Chen Jinling, Zhang Jiming, Deng Linming, Niu Xiaoming, Wei Xueming, Xu Jing Wu Jian, "Detection of Weak and Small Infrared Targets Under

Complicated Background", 5th International Symposium on Advanced Optical Manufacturing and Testing Technologies, SPIE 2010

[16] Ki Tae Park, Jongmyeon Jeong, "Object Detection in Infrared Image With Sea Clutter", IEEE SCIS-ISIS 2012

[17] Yin Xu, JianQi Zhang, XinHuai Wang, and WeiKe Dong, "A Novel Method for Detecting Small Target in Infrared Clutter", Second International Workshop on Computer Science and Engineering, IEEE 2009

[18] Suyog D. Deshpande, M.H.Er, V. Ronda, Phillip Chan, "Max-Mean and Max-Median Filters for Detection of Small-Targets", Conference on Signal and Data Processing of Small Targets, SPIE 1999

[19] Shengxiang Qi, Jie Ma, Chao Tao, Changcai Yang, and Jinwen Tian, "A Robust Directional Saliency-Based Method for Infrared Small-Target Detection Under Various Complex Backgrounds", IEEE Geoscience And Remote Sensing Letters, VOL. 10, NO. 3, MAY 2013

[20] Yanfei Chen, Nong Sang, Zhiping Dan, "A Saliency-based Approach to Detection of Infrared Target", Automatic Target Recognition and Navigation, SPIE 2013

[21] G. Xu, H. Huo, T. Fang, D. Li, "Extracting salient object from remote sensing image based on guidance of visual attention", Proc. SPIE. Vol. 6790. 2007

[22] X. Hou, J. Harel, and C. Koch, "Image Signature: Highlighting Sparse Salient Regions", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 34, no. 1, pp. 194-201, Jan. 2012

[23] C. L. Philip Chen, Hong Li, Yantao Wei, Tian Xia, and Yuan Yan Tang, "A Local Contrast Method for Small Infrared Target Detection", Transactions On Geoscience And Remote Sensing, IEEE 2013

[24] Jonah C. McBride, Mark R. Stevens, Ross S. Eaton and Magnús Snorrason,

"Adaptive Infrared Target Detection", Automatic Target Recognition XIV, SPIE 2004

[25] Alex Lipchen Chan, Sandor Z. Der, Nasser M. Nasrabadi, "Multistage Infrared Target Detection", Optical Engineering, SPIE 2003

[26] Toufiq Parag, "Coupled Label and Intensity MRF Models for IR Target Detection", Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE