

Old Dominion University

ODU Digital Commons

Electrical & Computer Engineering Theses & Dissertations

Electrical & Computer Engineering

Spring 2007

Long-Range Target Classification in a Cluttered Environment Using Multi-Sensor Image Sequences

Cenk Yaman
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/ece_etds



Part of the [Electrical and Computer Engineering Commons](#), and the [Theory and Algorithms Commons](#)

Recommended Citation

Yaman, Cenk. "Long-Range Target Classification in a Cluttered Environment Using Multi-Sensor Image Sequences" (2007). Master of Science (MS), Thesis, Electrical & Computer Engineering, Old Dominion University, DOI: 10.25777/fqd4-sa17
https://digitalcommons.odu.edu/ece_etds/576

This Thesis is brought to you for free and open access by the Electrical & Computer Engineering at ODU Digital Commons. It has been accepted for inclusion in Electrical & Computer Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

**LONG-RANGE TARGET CLASSIFICATION IN A CLUTTERED
ENVIRONMENT USING MULTI-SENSOR IMAGE SEQUENCES**

by

**Cenk YAMAN
B.S. August 1995, YILDIZ TECHNIC UNIVERSITY
ISTANBUL/TURKEY**

**A Thesis submitted to the Faculty of
Old Dominion University in Partial Fulfillment
of the Requirement for the Degree of**

MASTER OF SCIENCE

ELECTRICAL ENGINEERING

**OLD DOMINION UNIVERSITY
May 2007**

Approved by:

Vijayan K. Asari (Director)

Sacharia Albin (Member)

Zia-ur Rahman (Member)

ABSTRACT

LONG-RANGE TARGET CLASSIFICATION IN A CLUTTERED ENVIRONMENT USING MULTI-SENSOR IMAGE SEQUENCES

Cenk YAMAN
Old Dominion University, 2007
Director: Dr. Vijayan Asari

Accurate identification of unknown contacts is crucial in military intelligence. Automated systems which quickly and accurately determine the identity of a contact could be a benefit in backing up electronic signal identification methods such as Identification Friend and Foe (IFF) systems. Radio Detection and Ranging (RADAR) images are often undesirable in military applications since they reveal the location of the imaging system. So we explore the use of visible and infrared images of which are generally more consistent than RADAR images and for which it is easy to compensate for environmental effects. Recent advances in visible and IR imaging technology improve the ability to observe objects at very long distances, but it is still militarily desirable to stay away as far as possible from potential enemy ships that may be observed at different viewing angles.

A Principal Component Analysis (PCA) based pattern recognition technique is presented in this thesis for small boat classification from visible and IR images. Extracting features from naval ship images is a difficult task to solve since the background may contain rough-textured regions. Conventional edge and corner operators tend to fail due to high contrast in these regions. Appropriate preprocessing steps are necessary to separate the object from the background. Subsequent feature extraction techniques can be applied to the whole object region. Two efficient segmentation

algorithms are used for the extraction of small boat regions from visible and IR images. A popular and convenient approach named Adaptive Progressive Thresholding (APT) technique is used for segmentation of boat region from visible images. Infrared images in general are not sharp in displaying objects compared to the visible images. In fact, the noise components present in these images are always much greater than typical visible images. Therefore, pure edge-based image segmentation will not be sufficient in this case. We applied SUSAN-APT and Graph cut image segmentation algorithms for small boat region extraction from IR images. We obtained 86% accuracy in classification rate in visible images and 77% in IR images. The classification accuracy can be improved by using more training images captured in every 15 degrees viewing angle for all classes as opposed to only 5 training images used in this work for every class. This research can also be easily applied to air and land based target classification systems.

To my wife and my daughter,

I have to thank my wife for your endless love and support. Your support has been more than anyone can expect. Therefore I dedicated my thesis to my wife Serap, my best friend, partner and mother of my daughter Serra. Without your contribution and understanding, I would have never been able to do this work.

ACKNOWLEDGEMENTS

First of all, I am very thankful to my advisor, Dr. Vijayan Asari, for his sincere guidance, continuous support and patience to listen. I heartily acknowledge his constant encouragements during my research. I benefited a lot from his knowledge and personality.

My sincere acknowledgements to Dr. Sacharia Albin and Dr. Zia-ur Rahman for serving on my thesis committee.

I would also like to thank my friends Numan Unaldi, Okay Isik, Ismail Kosum, Sertan Erkanli for their cooperation, support and precious friendship right from the beginning of the master's period.

I extend my thanks to Dr. Oktay Baysal and Dr. Sefer Kurnaz for initiating the collaborative MS. and PhD. programs between Old Dominion University and Aerospace and Space Technologies Institute.

Finally, I want to thank the Turkish Air Force (TUAF) who supported this work greatly. I will always be proud to be a TUAF member.

TABLE OF CONTENTS

	Page
LIST OF FIGURES.....	viii
LIST OF TABLES.....	ix
CHAPTER	
1. INTRODUCTION.....	1
1.1. Thesis Contribution.....	8
1.2. Thesis Outline.....	8
2. LITERATURE SURVEY.....	10
2.1. Statistical Methods.....	12
2.1.1. Motivation in Dimension Reduction Problem.....	12
2.1.2. Principal Component Analysis.....	13
2.1.3. Target Classification with PCA.....	15
2.1.3.1. Training.....	15
2.1.3.2. Testing.....	17
2.2. Image Segmentation.....	18
2.2.1. Image Thresholding Technique.....	21
2.2.2. Otsu's Method	22
3. VISIBLE AN IR SMALL BOAT IMAGE SEGMENTATION.....	27
3.1. Visible Small Boat Image Segmentation.....	28
3.1.1. Background Subtraction	29
3.1.2. Adaptive Progressive Thresholding(APT).....	31
3.1.2.1. APT Algorithm	32
3.2. IR Small Boat Image Segmentation.....	37
3.2.1. SUSAN with APT for IR Small Boat Image Segmentation.....	39
3.2.2. Entropy Based Thresholding.....	45
3.2.2.1. Minimum Cross Entropy Thresholding.....	47
3.2.3. Graph-cut Segmentation Techniques.....	49
3.2.3.1. Graph-cut for Image Segmentation.....	50
3.2.3.2. Graph-cut Algorithm.....	51
3.2.3.3. Grab-cut for Image Segmentation.....	54
3.2.3.4. Region Modeling Using Grayscale Histograms.	57

3.2.3.5. The Graph Weights.....	57
3.2.3.6. Region Modeling Using Gaussian Mixture Model.....	60
3.2.3.6.1. Gaussians	61
3.2.3.6.2. Gaussian Mixture Model.....	61
3.2.3.7. Clue Marking Method.....	63
3.2.3.8. The Graph Weights	65
3.2.3.9. Region Modeling Using Color Histograms.....	66
3.2.3.10. The Graph Weights.....	68
3.2.3.11. The “Grab-cut” Image Segmentation Implementation for IR Grayscale Image.....	69
4. CLASSIFICATION RESULTS.....	76
5. CONCLUSION AND FUTURE WORK.....	83
5.1. Future Work and Recommendation.....	85
REFERENCES.....	87
VITA.....	92

LIST OF FIGURES

Figure	Page
1.1. Illustration of enemy RADAR detecting a RADAR system location.....	1
1.2. AGM-88 HARM missile.....	2
1.3. Example of RADAR base system is not detecting target.....	3
1.4. Block diagram of typical ATR system.....	5
2.1. Illustration of mean image.....	16
2.2. Reconstruction of an image using eigenvectors	18
2.3. Illustration of Otsu segmentation.....	25
2.4. Original image histogram.....	26
3.1. Illustration of unsatisfactory segmentation result with Otsu	28
3.2. Illustration of unsatisfactory segmentation results with edge detection.	29
3.3. Illustration of background subtraction segmentation method.....	31
3.4. Visible image segmentation step-by-step with APT.....	36
3.5. Some of the segmented image with APT.....	37
3.6. Comparison of the hard and soft cut-off functions for pixel assimilation in the SUSAN algorithm.....	41
3.7. Segmented image step by step with SUSAN-APT.....	43
3.8. Example of segmented image with SUSAN-APT	44
3.9. Example of unsatisfactory segmentation result with SUSAN-APT.....	44
3.10. Example of bad or no segmentation result with SUSAN-APT.....	45
3.11. Some of the segmentation result with entropy based thresholding.....	49
3.12. Explanation of Grab-cut segmentation process.....	55
3.13. Histogram models in a grayscale image.....	58
3.14. A Gaussian probability function.....	60
3.15. A 1D Gaussian mixture model.....	63
3.16. Pixel labelling when using gaussian mixture models.....	65
3.17. Color image segmentation.....	68
3.18. The neighborhood labeling system.....	70
3.19. IR grayscale image segmentation.....	73
3.20. Some of the example segmented image.....	74
3.21. Unsatisfactory segmentation was improved with user touch-up.....	74

3.22.	Wrong or even no segmentation result after touch-up.....	75
4.1.	The magnitude of the eigenvalues represent covariance matrix.....	76
4.2.	Classification program user interface.....	82

LIST OF TABLES

Table	Page
3.1. Edge weights of the graph.....	57
3.2. Edge weights between nodes in the graph.....	65
3.3. Color image segmentation times.....	67
3.4. Grayscale image segmentation times	72
4.1. Unsegmented visible image classification result.....	78
4.2. Unsegmented IR image classification result.....	79
4.3. Segmented visible image classification result.....	80
4.4. Segmented IR image classification result.....	81

CHAPTER 1

INTRODUCTION

Accurate identification of unknown contacts is crucial in military intelligence. Automated systems which quickly and accurately determine the identity of a contact could be a benefit in backing up electronic-signals identification methods like IFF (Identification Friend and Foe) systems.

RADAR (Radio Detection and Ranging) images are often undesirable in military applications because they reveal the location of the imaging system. RADAR based target detection and classification systems have the following three vulnerable points.

a. RADAR systems propagate electromagnetic (EM) energy, and the enemy could be able to detect the RADAR system location easily. (Figure 1.1.)

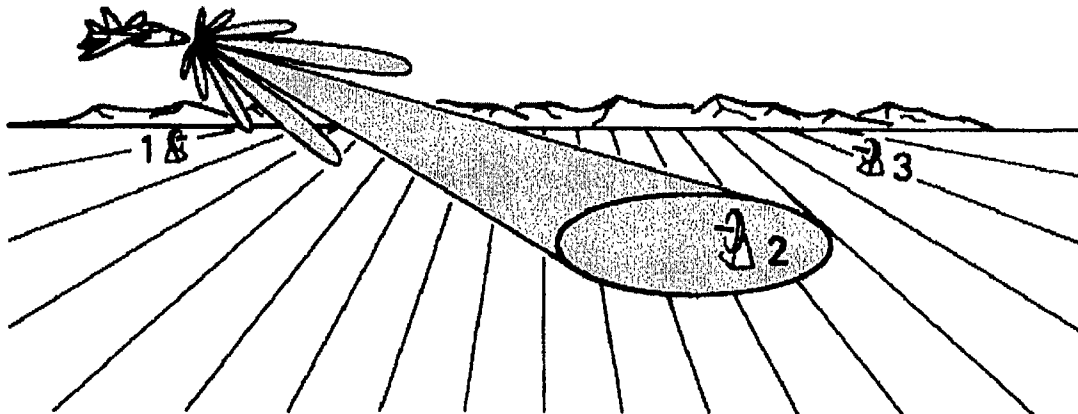


Figure 1.1. Illustration of enemy RADAR detecting a RADAR system location.

The enemy could use very effective and deadly HARM (High Anti-Radiation Missile) systems against the RADAR based detection system. The AGM-88 (Air to

Ground Missile) HARM, which is shown in Figure 1.2, is a supersonic air-to-surface tactical missile designed to seek and destroy enemy radar-equipped air defense systems. Guidance is provided through reception of signals emitted from a ground-based threat radar. Antiradiation missiles have an unparalleled ability to home in on enemy emitters and disrupt or destroy the elements of an integrated air defense system (IADS). It is called the primary lethal Suppression of Enemy Air Defense (SEAD) platform. It has 50 miles maximum effective range [1].



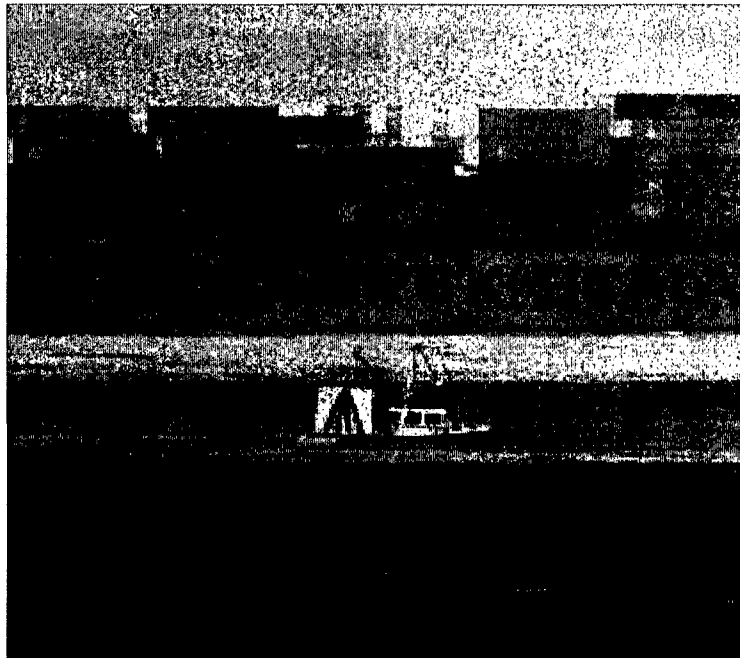
Figure1.2. AGM-88 HARM missile.

b. RADAR based systems are deceived very easily with decoys, ECM (Electronic Counter-Measure) System and IR (Infrared) countermeasures such as flares and chaffs.

c. RADAR based systems do not detect small boats and yachts, as evidenced by the attack on the USS Cole. For example, as shown in Figure 1.3, a small fishing boat could not be detected by RADAR based systems, since the big transport ship's RCS (RADAR Cross Section) hides or camouflages the RCS of the small boat.



Figure 1.3.a. The big transport ship's RCS hides the RCS of the small boat.



b. Small boat image.

Figure 1.3. Example of RADAR base system is not detecting target.

So we explore visible and infrared images of ships which are generally more consistent than RADAR images and for which it is easier to compensate for

environmental effects. Recent advances in visible and IR imaging technology improves its ability to observe objects at very long distances, but it is still militarily desirable to stay as far away as possible from potential enemy ships that may be observed at different viewing angles.

Human visual system performance greatly exceeds computer capabilities, probably because of superior high-level image understanding, contextual knowledge, and massive parallel processing capabilities. Human capabilities deteriorate drastically in a low-visibility environment. After an extended period of surveillance, certain working environments are either inaccessible or too hazardous for human beings [2].

For these reasons, automatic recognition systems are developed for various military and civilian applications. Driven by advances in computing capability and image processing technology, computer mimicry of human vision has recently gained ground in a number of practical applications. Specialized recognition systems are becoming more likely to satisfy stringent constraints in accuracy and speed, as well as the cost of development and maintenance.

The task of automatically recognizing targets and classification in visible and IR imagery has a history of approximately 25 years of research and development. Automatic target recognition (ATR) is an application of pattern recognition and scene analysis in the field of defense industry and, it is still one of the challenging problems. ATR task, which is also referred to as weapon vision [3], is one of the challenging problems of the defense industry. The aim of an ATR system is to remove the role of human beings from the process of target detection and recognition and hence for implementing a real-time and reliable system of high performance.

The automatic target recognition term originated with the Low Altitude Navigation

and Targeting Infrared for Night (LANTIRN) program in the early 1980's from the US Air Force's (USAF) F-16 Fighting Falcon modernization project. Prior to the LANTIRN program, little had been done in the area that became known as ATR [4]. In the most general sense, an ATR system is composed of a target detector and recognizer. Such an inclusive system of ATR covers the problems of preprocessing, detection, segmentation, classification, tracking and prioritization. Figure 1.4 represents the block diagram of such a typical system and the gray box represents the scope of this thesis.

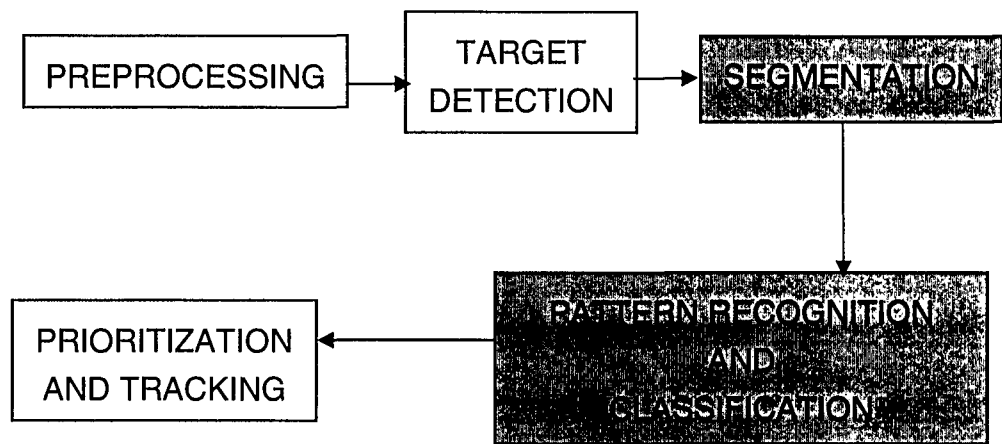


Figure 1.4. Block diagram of typical ATR system.

Briefly, the preprocessing is the step that improves target contrast and reduces noise and clutter present in the image. Examples of preprocessing functions are noise suppression, focus control, adaptive contrast enhancement, histogram equalization, etc [5].

Target detection is the process of localizing those areas in the image where a potential target is likely to be present. Conventionally, the detection techniques are based on the contrast between the target and its immediate background. Most of the techniques

can be adapted to detect either light or targets. Once a potential target is localized, it is extracted from the background as accurately as possible in the segmentation step.

Pattern recognition and classification is the process of associating detected targets with target classes. Classification of target images classifies the still images or video sequences into one of the predefined target classes using the information extracted through the examination of the precollected data, called the training set.

Prioritization is the process of assigning priorities to the targets in the field of view. This information, which is pre-stored, is normally based on the type of the target and the probability of its correct classification. The prioritized target is then tracked.

The development of robust ATR systems must still overcome a number of well-known challenges, such as the large number of target classes and aspects, long viewing range, obscured targets, high-clutter background, different geographic and weather conditions, sensor noise, and variations caused by translation, rotation, and scaling of the targets. Inconsistencies in the signature of targets, similarities between the signatures of different targets, limited training and testing data, camouflaged targets, nonrepeatability of target signatures, and difficulty using available contextual information make the recognition problem even more challenging.

An ATR system may be used as guidance to armed forces operating under inferior weather conditions or at night. Moreover, with ATR, remotely controlled vehicles such as Unmanned Air Vehicles (UAV) or cruise missiles are also designed. UAV are another application area of the ATR systems. The performance of such systems is dictated by the large volumes of data requiring analysis and by the short timelines required by the target acquisition scenarios. ATR is a high leverage technology, and the challenge is to go beyond human-aided capabilities to build autonomous systems [6].

A Principal Component Analysis (PCA) based pattern recognition system is presented in this thesis for small boat classification from visible and IR images.

Feature extraction is more complex than detection, since extraction implies that we have a description of a shape, such as its position and size, whereas detection of a shape merely implies knowledge of its existence within an image.

Extracting features from naval ship images is a difficult task to solve because the background may contain rough-textured regions. Conventional edge and corner operators tend to fail due to high clutter in these regions. Appropriate preprocessing steps are necessary to separate the object from the background. Subsequent feature extraction techniques can be applied to the whole object region. Two efficient segmentation algorithms are used for the extraction of small boat regions from visible and IR images. A popular and convenient approach called Adaptive Progressive Thresholding (APT) technique is used for segmentation of visible images.

In general, infrared images are not sharp while displaying objects compared to visible images. In fact, the noise components present in these images are always much greater than typical visible images. Therefore, pure edge-based image segmentation will not be sufficient. We applied two different algorithms, named SUSAN-APT and Graph cut image segmentation for IR small boat images.

We obtained 86% accuracy in classification success rate in visible images and 77% accuracy in IR images. The classification accuracy can be improved by using more training images captured in every 15 degree viewing angle for all classes as opposed to only 5 training images used in this work. This work can also be easily applied to air and land based target classification system.

1.1. Thesis Contribution

Several methods have been described in literature to perform the automatic classification of ships from images, but none are based on small boat classification. Note that most papers are 5 to 20 years old, which is a long time for an image processing and pattern recognition application.

Some of the recent proposed classification algorithms are based on the classification of big ships (e.g. aircraft carriers, combat ships, transportation ships, etc). But the proposed research work is based on small boat classification for which it is obviously harder to get good results.

This thesis contributes two robust segmentation algorithm for grayscale IR small boat images and one for visible small boat images. These algorithms are tested for images captured in a highly cluttered background.

1.2. Thesis Outline

Chapter 2 presents a detailed literature review on the thesis topic of this research. Several segmentation techniques have been studied and explained in this section. The PCA algorithm and the necessity of the dimensionality reduction for pattern recognition is also explained.

Chapter 3 discusses the visible and IR small boat image segmentation methods. An APT algorithm which is suitable for visible images and SUSAN-APT, Grab cut techniques which are suitable for IR images are explained in detail in this chapter.

Chapter 4 discusses PCA based classification results. The first part of this chapter deals with classification techniques and classification of unsegmented and segmented images are also explained in detailed.

The result of our experiment with the proposed algorithms is summarized in

Chapter 5. This part concludes this thesis work and presents suggestions for future research work in this area.

CHAPTER 2

LITERATURE SURVEY

Visual surveillance has been an active research area in computer vision and image processing due to its crucial role in helping military intelligence and law enforcement agencies to fight against crime and terrorist activities. The goal of a visual surveillance system is to detect abnormal object behaviors and to raise alarms when such behaviors are detected. After moving objects are detected, it is essential to classify them into predefined categories, so that their motion behaviors can be appropriately interpreted in the context of their identities and their interactions with the environment. Consequently, object classification is a vital component in a complete visual surveillance system. Pattern Recognition Algorithms make Object classification task.

The image or video based surveillance system is decomposed in two main parts, whose main goals are:

1. Detect and track objects of interest (e.g., ships, personnel, and persons on the coastline) from an incoming video or image data,
2. Recognize and classify their behaviors as normal or abnormal and take necessary actions (e.g., raise alarms) in the case of abnormal behaviors.

A visual surveillance system typically works according to the following essential stages:

1. Moving objects are detected from the surveillance video. Background subtraction is a common and effective technique when the camera is fixed. It is also applicable for mobile cameras if the motion of the camera can be computed to compensate for the motion in the video. This stage is crucial since undetected objects cannot be tracked and

further analyzed.

2. Detected moving objects are tracked across the video frames to generate trajectories that enable the characterization of the motion behaviors of the moving objects.

3. Tracked moving objects are classified into a set of predefined objects. This stage is vital since the interpretation of the motion behaviors and assessment of potential risk are usually based on the type and identity of the moving objects.

4. The motion behaviors of the classified objects are analyzed. When abnormal behaviors are detected, the system should raise alarms to inform the security personnel.

The objective of pattern recognition is to classify objects into different categories and classes. It is a fundamental component of artificial intelligence and computer vision [9].

As humans, it is easy (even for a child) to recognize letters, objects, numbers, voices of friends, etc. However, making a computer solve these types of problems is a very difficult task. Pattern recognition methods are used in various areas such as science, engineering, business, medicine, etc. Interest in pattern recognition is fast growing in order to deal with the enormous amount of information we encounter in our daily life.

There are plenty of algorithms developed and being developed in the area of pattern recognition. These algorithms fall into one of the three general categories of statistical, neural and model based approaches [7]. In the first two categories, the information to be used in the classification task is implicitly extracted, while in the latter a model database is constructed either with CAD tools or from the real data, and then comparison is achieved among these templates.

2.1 Statistical Methods

Statistical Pattern Recognition is one of the major approaches in the pattern recognition discipline. In statistical methods, a target image is represented with a set of characteristic measurements called feature vectors [8].

Features constitute points in a d -dimensional space, and the feature space is formed in a decision theoretic basis with the a priori knowledge of underlying distribution and/or the statistical properties of a set of known samples, namely the training set. The goal is to form the feature space such that the features of the targets belonging to different classes are clustered at different regions of the feature space.

Statistical methods include projection based methods, such as Principal Component Analysis, Linear Discriminant Analysis and Independent Component Analysis. In classification, patterns are images of targets and this constitutes a very high dimensional space of vectors. The aim for using statistical methods in classification is reducing the dimension and overcoming the problems of the limited data sets with which the feature extractors are trained. Statistical methods are especially satisfactory for patterns with well-behaved distributions.

2.1.1 Motivation in Dimension Reduction Problem

Due to advances in data collection capabilities, researchers in engineering, astronomy, economics and statistics encounter an increasing number of variables associated with each observation. These high dimensional datasets present many mathematical challenges, as well as some opportunities. As an important problem in high dimensional datasets for many cases, not all the measured variables are ‘important’ for understanding the underlying phenomena of interest [7].

Having large amounts of high dimensional sensory data to process, analyze or

store, dimension reduction is needed for:

- a. Visualization
- b. Data compression for transmission or storage
- c. Decreasing computation time and memory usage
- d. Change of representation for statistical pattern recognition and modeling

Dimension reduction is the problem of finding a k -dimensional representation of a d -dimensional random variable, with $k < d$, that captures the content in the original data with respect to some criterion. There exist different criteria for each dimension reduction problem, such as minimizing the reconstruction error, preserving distances or maximizing likelihood with respect to some model [10].

Within the context of pattern recognition, high dimensionality introduces the well-known limitation, which is denoted by [11] as “the curse of dimensionality”. For linear or quadratic classifiers, the required number of training samples depends linearly or quadratically on the data dimensionality. Furthermore, the training sample set size needs to increase exponentially, in order to effectively estimate the multivariate densities needed to perform nonparametric classification [12].

In the following subsections, the prominent PCA (Principal Component Analysis) will be summarized.

2.1.2 Principal Component Analysis

Principal Component Analysis (PCA) method is a conventional method, as it has been used in numerous disciplines and applications. Examples include color representation, texture segmentation, multispectral image classification, face recognition, source separation, visualization, and image database queries and appearance-based recognition [13].

Principal component analysis (PCA) is a popular linear transformation method used for dimensionality reduction. PCA involves the eigen analysis of the co-variance matrix obtained from training data. PCA is an optimum data compression method when training data is available to construct the transformation matrix [14]. Synonyms of PCA include Karhunen–Loève (KL) transform, Hotelling transform, eigenvalue analysis, eigenvector decomposition, and spectral decomposition [15]. In image analysis it is used to reduce dimensions, and to find subspaces in which recognition works better than taking the full space.

This method is applicable to the target classification problem provided in which a region of interest is detected within the input image [9]. PCA is based on the correlation between image pixels, and its utility to image analysis is in part due to the correlation between nearby pixels in a real-world image.

The idea behind PCA is to express the data in a lower dimensional space with the minimum error in the least square sense. This means that the criterion of the feature extraction with PCA is the minimization of the reconstruction error. It should be noted that PCA is an unsupervised method. In the mean squared error sense, Principal Component Analysis (PCA) is the best linear dimension reduction technique. Since it is based on the covariance matrix of the variables, it is a second order method; therefore PCA considers the pair-wise relationships between variables of an observation set (e.g. pixels in the image database).

PCA looks for orthogonal basis functions for which the components of the signal are uncorrelated. The main aim of the PCA is to reduce the dimensionality by finding orthogonal linear combinations of the original variables with the largest variance. An N -dimensional random variable has N principal components. However, for many datasets,

the first several principal components retain most of the variance so that the rest can be discarded with minimal loss of information. In order to rephrase the PCA method in mathematical terms, consider M observations of N -dimensional random variable.

2.1.3. Target Classification with PCA

PCA aims to encode the relevant information of the target image as efficiently as possible [16]. One encoded target is compared to targets similarly encoded. The variation information in a collection of targets is compared to individual target images. The principal components or the eigenvectors of a covariance matrix of the set of target images are found. Each image can be represented as a linear combination of the eigenvectors, those corresponding to the largest eigenvalues.

2.1.3.1. Training

Consider the target images to be size N by N . These images are represented by a vector size N^2 . Since the target images have a similar structure, the vectors representing them will be correlated. These vectors define subspace of the target images which is called “space”. Due to correlation, the images can be represented by a lower dimensional space.

Let I_1, I_2, \dots, I_M be the training set of target images. The average is found by:

$$A = \frac{1}{M} \sum_{i=1}^M I \quad (2.1)$$

It is called the mean of the image database. Figure 2.1 shows that mean image.

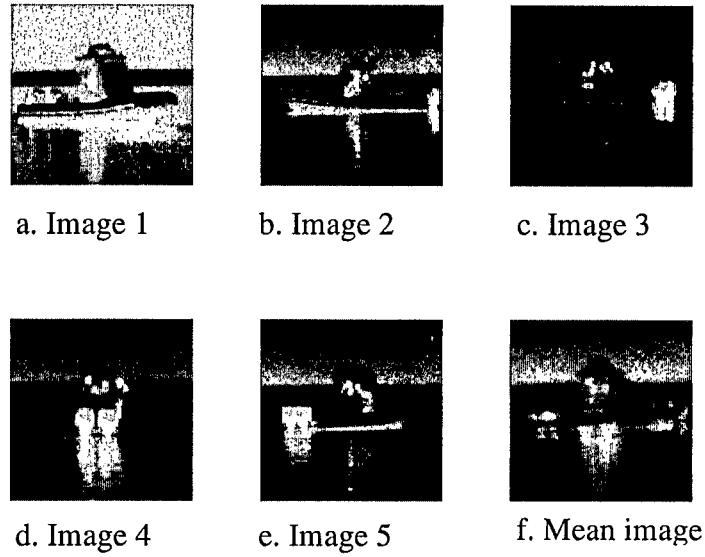


Figure 2.1. Illustration of mean image.

The vector $Y_i = I_i - A$ is the difference image of each target. The covariance matrix is obtained from the difference vectors as:

$$C = \frac{1}{M} \sum_{i=1}^M Y_i Y_i^T \quad (2.2)$$

The eigen vectors of the covariance matrix are computed. N' eigen vectors corresponding to the largest eigen values are selected. The weight vector for each image is computed using these eigen vectors as:

$$W_{ik} = E_k^T \cdot (I_i - A) \quad (2.3)$$

where, E_k are the eigenvectors corresponding to the N' largest eigenvalues of C , and k varies from 1 to N' . A mean weight vector is found for each individual in the testing database.

2.1.3.2 Testing

The test image is transformed to weight vector form as:

$$W_{test} = E_k^T \cdot (I_{test} - A) \quad (2.4)$$

The mean weight vectors are used to fit the test image to a predefined test class. A simple method to do this would be finding the minimum distance from test weight vector W_{test} to the mean weight vector T_p . The test image can be classified to be in class p when

$$D_p = \|W_{test} - T_p\|, \min(D_p) < \theta_i \quad (2.5)$$

Where θ_i is threshold.

The eigenimage technique is also used to detect a target in an image. The distance between the local sub-image and target space is calculated. This distance from the target space is used as a measure. Figure 2.2 shows the reconstructed images with the first 1, 2, 5, 50 eigenvectors.

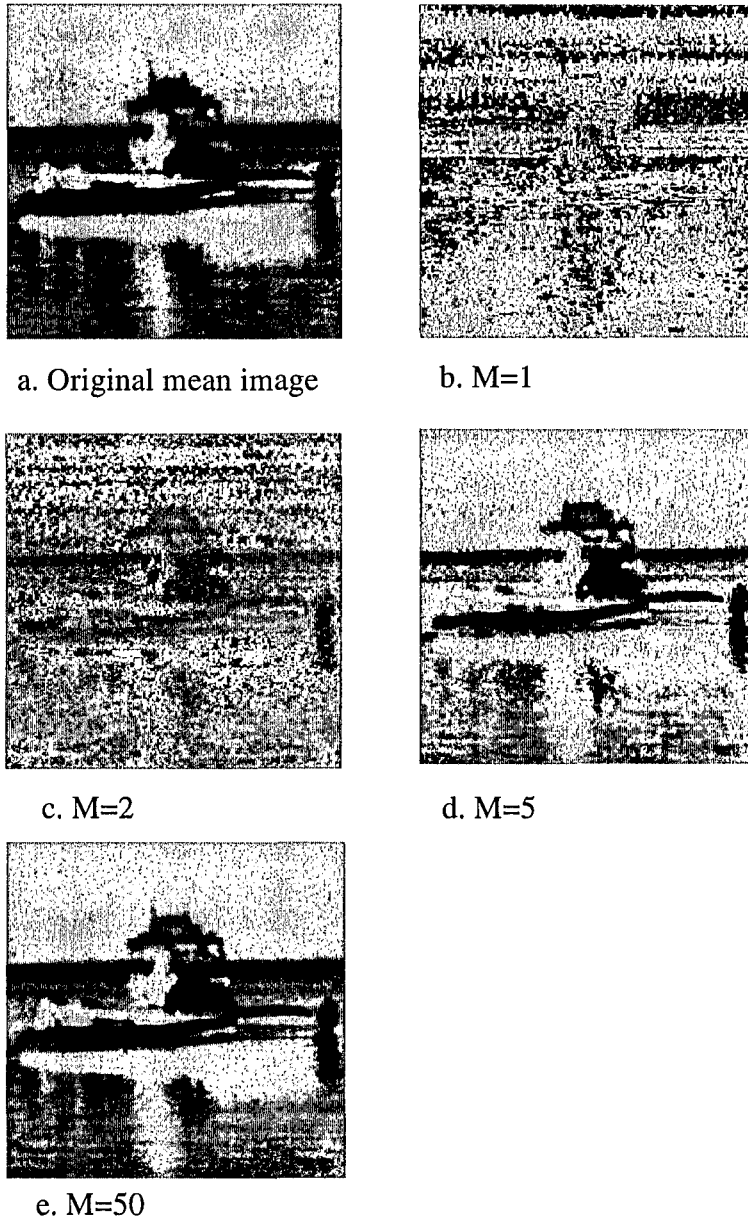


Figure 2.2. Reconstruction of an image using eigenvectors. a. Original mean image b. $M=1$ reconstructed from first eigenvector c. $M=2$ reconstructed from first and second eigenvector. D. $M=5$ reconstructed eigenimage. e. $M=50$ reconstructed eigenimage.

2.2 Image Segmentation

Computer pattern recognition, as its name suggests, is an artificial intelligence system that recognizes certain patterns such as characters, human faces, lines and their

structures.

But these simple tasks which are easy for our human recognition system are not so easy for a computer although the latter has a great capacity for calculation. Documents, photos or drawings have to be scanned into the computer as images before they can be processed further. These images which contain the interesting objects, such as characters, faces or lines, appear to the computer just as lots of dots with certain color intensity value and do not have any obvious meanings. Also there may be some noise introduced into the images during the scanning process.

Before we can ask the computer to recognize any pattern, the features of these patterns have to be extracted from the possibly distorted, blurry or faint images, such as extracting characters from a dark background or the outline of a face from a photograph.

Image and video segmentation is one of the most critical tasks of analysis which has the objective of extracting information (represented by data) from an image or a sequence of images (video). In the last 40 years, this field has experienced significant growth and progress. The history of segmentation of digital images using computers could be traced back 40 years. Since then, this field has evolved very quickly and has undergone great change [17].

Image segmentation requires the ability to quickly and easily extract foreground objects from digital images. A foreground object refers to any object of interest in an image. The background of the image refers to all pixels in the image that are not part of the foreground object. The process of separating an image into foreground and background parts is known as image segmentation.

Humans have the ability to recognize objects and can perform image segmentation by just identifying an object in an image. Children going to preschool learn how to

segment images by cutting an object out of a picture along a dotted line. We find image segmentation to be an extremely simple task, but it is definitely not a trivial task for computers. In order for computers to perform image segmentation the algorithm performing the segmentation needs to use information encapsulated in the digital image to calculate the best segmentation [21].

Computers have no means of intelligently recognizing objects, so many different methods have been developed in order to segment images. The segmentation process is based on various features found in the image. This might be color information that is used to create histograms, or information about the pixels that indicate edges or boundaries or texture information [18].

Image, from its general sense, could embrace all media that can be visualized by human beings, such as still image, video, animation, graphics, charts, drawings and even text. From images, human beings obtain the majority of information from the real world. To better perceive images and to gain more information from these perceptions, various techniques have been developed and many applications have been discovered.

Image segmentation is the first step and also one of the most critical tasks of image analysis. It has the objective of extracting information (represented by data) from an image via image segmentation, object representation and feature measurement. It is evident that the results of segmentation will have considerable influence over the accuracy of feature measurement [19].

Image segmentation is often described as the process that subdivides an image into its constituent parts and extracts those parts of interest (objects). It is one of the most critical tasks in automatic image analysis because the segmentation results will affect all the subsequent processes of image analysis, such as object representation and description,

feature measurement and even the following higher level tasks such as object classification and scene interpretation.

The first development of techniques for image segmentation can be traced back 40 years. In 1965, an operator for detecting edges between different parts of an image, the Roberts operator (also called the Roberts edge detector), was introduced [20]. This detector was the first step toward decomposing an image into its constitutional components. Since then, a large number of techniques and algorithms for image segmentation have been proposed, the result of much effort devoted to the research and application of image segmentation processes and development. In the meantime, concept and scope of images have been extended greatly.

The extension of 2-D images to 3-D, still images to moving images or sequences of images (video), gray level images to color or multi-band images, etc. have also helped the concepts and techniques of image segmentation expand widely.

Despite several decades of investigation, image segmentation remains a challenging research topic.

2.2.1. Image Thresholding Technique

Thresholding is a very simple form of segmentation and a commonly used technique in image segmentation because of its fast and easy application. For this reason threshold selection is an important issue. There are two general approaches to threshold selection. One approach is based on the histogram of the image while the other is based on the grayscale information located in local small areas. The histogram of an image contains some statistical data of the grayscale or color ingredients.

Segmentation is performed by defining a threshold, which will most often be an intensity value in the image. Every pixel in the image is compared with this threshold, and

if the pixel's intensity is greater than the threshold value it will be marked as foreground, and if it is less than the threshold as background. Other forms of thresholding exist where the threshold is allowed to vary across the image.

Thresholding is a primitive technique and will only work for very simple segmentation tasks. This is because complex foreground objects will often contain pixels which have intensity values that also lie in the background. Under these circumstances pixels which are part of the background will be incorrectly segmented as part of the foreground, and pixels which should be segmented as foreground will be segmented as background.

In this thesis, an adaptive progressive thresholding method is proposed for the binarization of visible small boat images first.

2.2.2. Otsu's Method

Nobuyuki Otsu has developed a nonparametric and unsupervised method of automatic threshold selection for image segmentation [22]. This method tries to evaluate the “goodness” of the threshold and automatically selects an optimal threshold according to a certain criterion.

It is based on the assumption that well thresholded classes would be separated in grey-levels and conversely, that a threshold giving the best separation of several classes (assumed to correspond to uniform grey-levels) should be the best classification threshold. In statistical terms this is translated into minimizing the within-class variance and maximizing the between-class variance with respect to grey-level. In this assumption “grey-level” could of course also be replaced by some other attribute.

Let the pixels of a given image be represented by L grey-levels $[1, 2, \dots, L]$. The number of

pixels at level i is denoted by n_i and the total number of pixels by $N = n_1 + n_2 + \dots + n_L$. In order to simplify the discussion, the grey-level histogram is normalized and regarded as a probability distribution:

$$p_i = n_i / N, \quad p_i \geq 0, \quad \sum_{i=1}^L p_i = 1 \quad (2.6)$$

With this normalization, we have the global mean level of the original image:

$$\mu_T = \mu(L) = \sum_{i=1}^L i p_i \quad (2.7)$$

Dividing the pixels into two classes C_0 and C_1 , with a threshold k gives us C_0 denoting pixels with levels $[1, \dots, k]$, and C_1 denoting pixels with levels $[k+1, \dots, L]$. Then, the probabilities of class occurrence are given by:

$$\omega_0 = P_r(C_0) = \sum_{i=1}^k p_i = \omega(k) \quad (2.8)$$

$$\omega_1 = P_r(C_1) = \sum_{i=k+1}^L p_i = 1 - \omega(k) \quad (2.9)$$

Where of course $\omega_0 + \omega_1 = 1$

The class mean levels are given by

$$\mu_0 = \sum_{i=1}^k i \Pr(i|C_0) = \sum_{i=1}^k i p_i / \omega_0 = \mu(k) / \omega(k) \quad (2.10)$$

$$\mu_1 = \sum_{i=k+1}^L i \Pr(i|C_1) = \sum_{i=k+1}^L i p_i / \omega_1 = \frac{\mu_T - \mu(k)}{1 - \omega(k)} \mu(k) / \omega(k) \quad (2.11)$$

Where

$$\omega(k) = \sum_{i=1}^k p_i \quad \text{and} \quad \mu(k) = \sum_{i=1}^k i p_i \quad (2.11-12)$$

For any choice of k we have

$$\omega_0\mu_0 + \omega_1\mu_1 = \mu_T \quad (2.13)$$

According to the definition of the variance for a discrete random variable [18] and using the notation $\Pr(i|C_1)$ for the probability of grey-level i given that the pixel is classified as C_0 , the class variances are given by

$$\sigma_0^2 = \sum_{i=1}^k (i - \mu_0)^2 \Pr(i|C_0) = \sum_{i=1}^k (i - \mu_0)^2 p_i / \omega_0 \quad (2.14)$$

$$\sigma_1^2 = \sum_{i=1}^L (i - \mu_1)^2 \Pr(i|C_1) = \sum_{i=1}^L (i - \mu_1)^2 p_i / \omega_1 \quad (2.15)$$

In order to evaluate the “goodness” of the threshold Otsu, use the within-class variance

$$\sigma_w^2(k) = \omega_0\sigma_0^2 + \omega_1\sigma_1^2 \quad (2.16)$$

the between class variance

$$\sigma_B^2(k) = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \quad (2.17)$$

and the total variance of levels

$$\sigma_T^2(k) = \sum_{i=1}^L (i - \mu_1)^2 p_i \quad (2.18)$$

where

$$\sigma_w^2 + \sigma_B^2 = \sigma_T^2 \quad (2.19)$$

Otsu defined the following measures of class separability, which are to be maximized by the optimal threshold k :

$$\lambda_1 = \frac{\sigma_B^2(k)}{\sigma_w^2(k)}, \lambda_2 = \frac{\sigma_T^2}{\sigma_w^2(k)}, \lambda_3 = \frac{\sigma_B^2(k)}{\sigma_T^2} \quad (2.20)$$

Since they are related according to equation 2.19 we are free to choose one of

them. Maximizing $\sigma_B^2(k)$ which becomes

$$\sigma_B^2(k) = \frac{[\mu_T \omega(k) - \mu(k)]^2}{\omega(k)[1 - \omega(k)]} \quad (2.21)$$

Finally the optimal threshold k^* that maximizes the between-class variance and minimizes the intra-class variance is:

$$\sigma_B^2(k^*) = \max_{1 \leq k \leq L} (\sigma_B^2(k)) \quad (2.21)$$

The range of k for which the maximum is sought can be restricted to

$$S^* = \{k; \omega_0 \omega_1 = \omega(k)[1 - \omega(k)] > 0 \text{ or } 0 < \omega(k) < 1\} \quad (2.22)$$

$\sigma_B^2(k)$ takes the minimum value of zero if we make all pixels C_0 or C_1 , which is not in our interest. For k it takes positive and bounded values and therefore there is a maximum. In Figure 2.3 and 2.4, we show the result of applying Otsu's method to an example image.



a. Original image



b. Segmented image with Otsu

Figure 2.3. Illustration of Otsu segmentation.

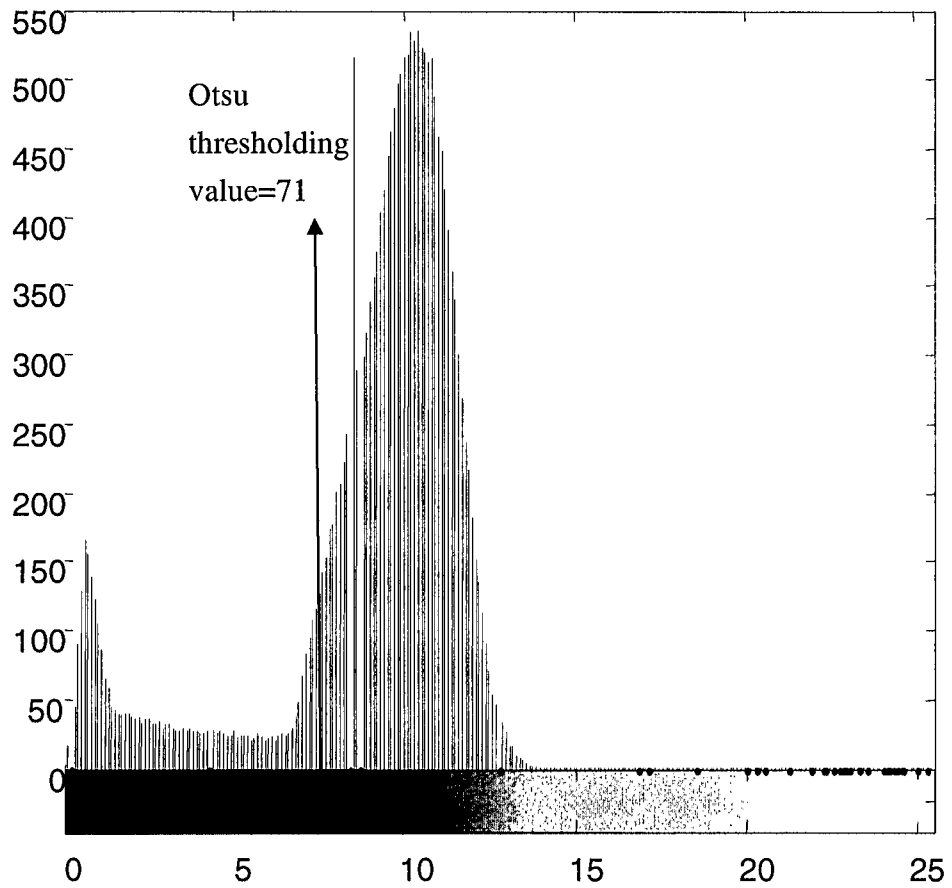


Figure 2.4. Original image histogram.

CHAPTER 3

VISIBLE AND IR SMALL BOAT IMAGE SEGMENTATION

Object segmentation in the domain of marine surveillance is faced with the task of distinguishing between an object of interest and a complex moving background. Extracting features from naval ship images is a difficult task to solve, because the background contains rough-textured regions, such as water. Conventional edge and corner operators tend to fail due to a high contrast in these parts. First appropriate preprocessing steps are necessary to separate the object from the background. Then subsequent feature extraction techniques can be applied to the object itself.

Feature extraction is more complex than detection, since extraction implies that we have a description of a shape, such as its position and size, whereas detection of a shape merely implies knowledge of its existence within an image.

Two efficient segmentation algorithms are used for the extraction of small boat regions from visible and IR images. The popular and convenient approach Adaptive Progressive Thresholding (APT) technique is used for segmentation of visible images.

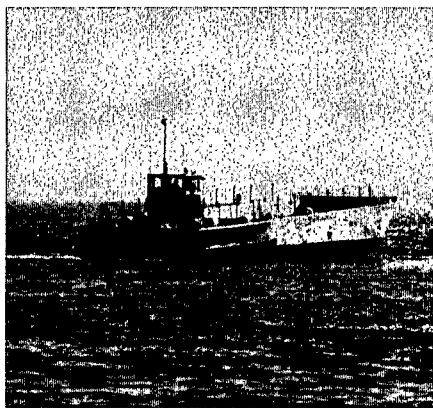
Infrared images in general are not that sharp in displaying objects compared to daylight images. In fact the noise floor present in the images is always much greater than in typical man-made pictures. Therefore, pure edge-based image segmentation will not be sufficient at all. For the IR naval small boat images, we applied the Grab-cut image segmentation algorithm explained in 3.2.3.3. The idea of Graph-cut is an energy minimization technique for image segmentation and solving it with graph-based algorithms. Graph-cut combines the two already known approaches for image segmentation: algorithms based on colors (or more precisely gray-levels) and

segmentation based on the contrast in different regions of an image.

Grab-cut extends this (Graph-cut) useful scheme to color images by using the iterative optimization structure. Instead of gray-level histograms, it makes use of Gaussian mixture models (GMM) and the Grab-cut algorithm uses only a three channel (RGB) based color image. Background and foreground are each described with five full-covariance Gaussian components.

3.1. Visible Small Boat Image Segmentation

Thresholding is a simple shape extraction technique, as illustrated in chapter 2. Our target image database has high background clutter and rough-textured regions, such as water. Classical edge and thresholding methods (e.g. Otsu's) fail due to high contrast in these parts. Otsu methodology results are given in Figure 3.1. Classical Sobel and Canny operator results are shown in Figure 3.2. Since both segmentation methods failed, we investigate other segmentation methodologies. First, background subtraction segmentation method which is popular and well-known for military applications will be explained.



a. Original visible image.



b. Segmentation result with Otsu.

Figure 3.1 Illustration of unsatisfactory segmentation result with Otsu.

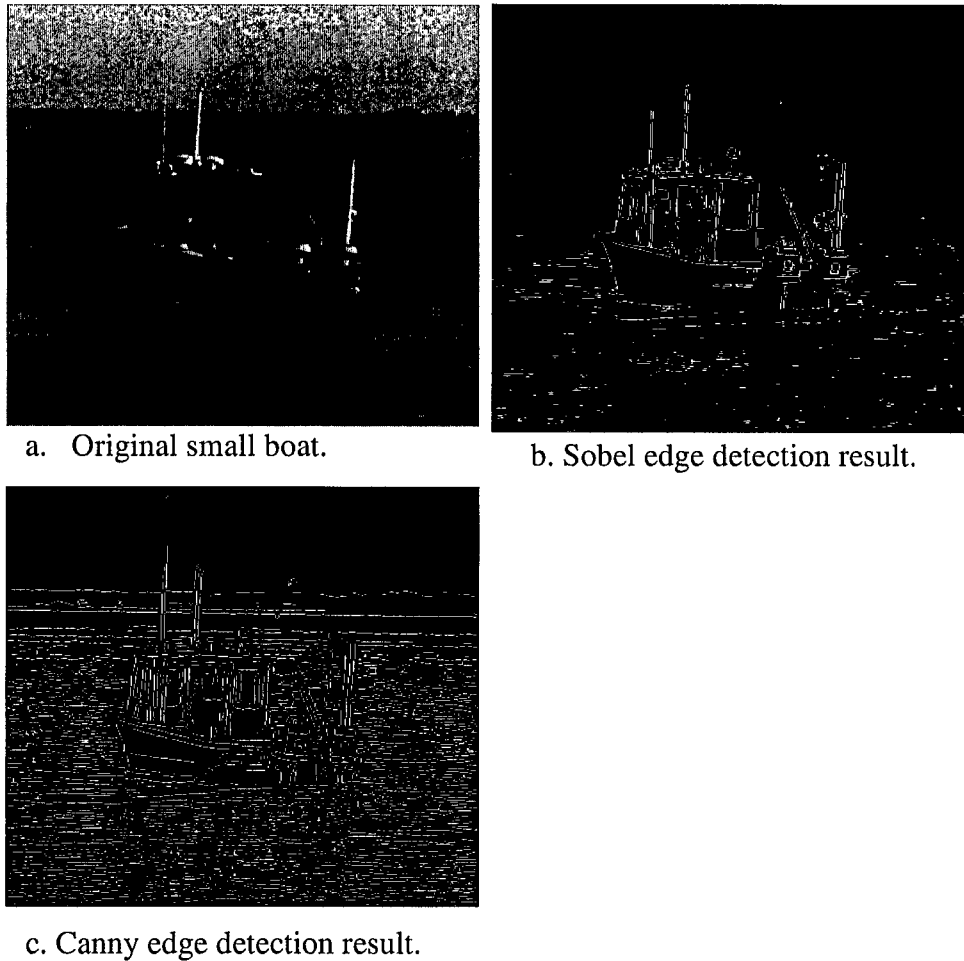


Figure 3.2. Illustration of unsatisfactory segmentation results with edge detection.

3.1.1. Background Subtraction

An alternative approach is to subtract an image used for target detection if the background image is a known background before thresholding. Most of the image based military applications get their images from visible or FLIR video cameras. All of the military application video cameras have a video capture card to obtain frames from video.

Common solutions to foreground object detection from a digital video are based on some form of background subtraction or background suppression [27]. These approaches work well when the camera is in a fixed position, and when there is no background

movement. This assumes that the background is known, and then this can be subtracted to obtain the pixels that define the shape of an object superimposed on the background to determine potentially threatening objects within a scene containing a complex, moving background.

The marine surveillance scene is likely to contain a moving background such as flickering water surfaces, moving clouds, wavering trees, and similar scenarios. Thus, it is essential that the algorithm can deal with such backgrounds and still detect the potential objects of interest. Even though thresholding and subtraction are attractive (because of simplicity and hence their speed), the performance of both techniques is sensitive to partial shape data, noise, variation in illumination and to occlusion of the target shape by other objects.

The subtraction approach is illustrated in Figure 3.3. Here, we seek to separate or extract the yacht subject from the background. When we subtract the background of Figure 3.3.b. from the image of Figure 3.3.a., we obtain most of the subject with some extra background just behind the subject in Figure 3.3.c. This is due to the effect of the moving subject on lighting. Also, removing the background removes some of the subject: the horizontal bars in the background have been removed from the subject by the subtraction process. These aspects are highlighted in the thresholded image, Figure 3.3.d. This video sequence is taken from the U.S. Coast Guard web page. First, the image is taken from a video sequence and subtraction is applied. Finally, thresholding is applied. This application is made in MATLAB 2006a.

Although this approach is easy, very effective, easy to use for military applications, we couldn't apply it for our image database, since we don't know the image background. If we get our image from a video camera, this approach is easily applicable.

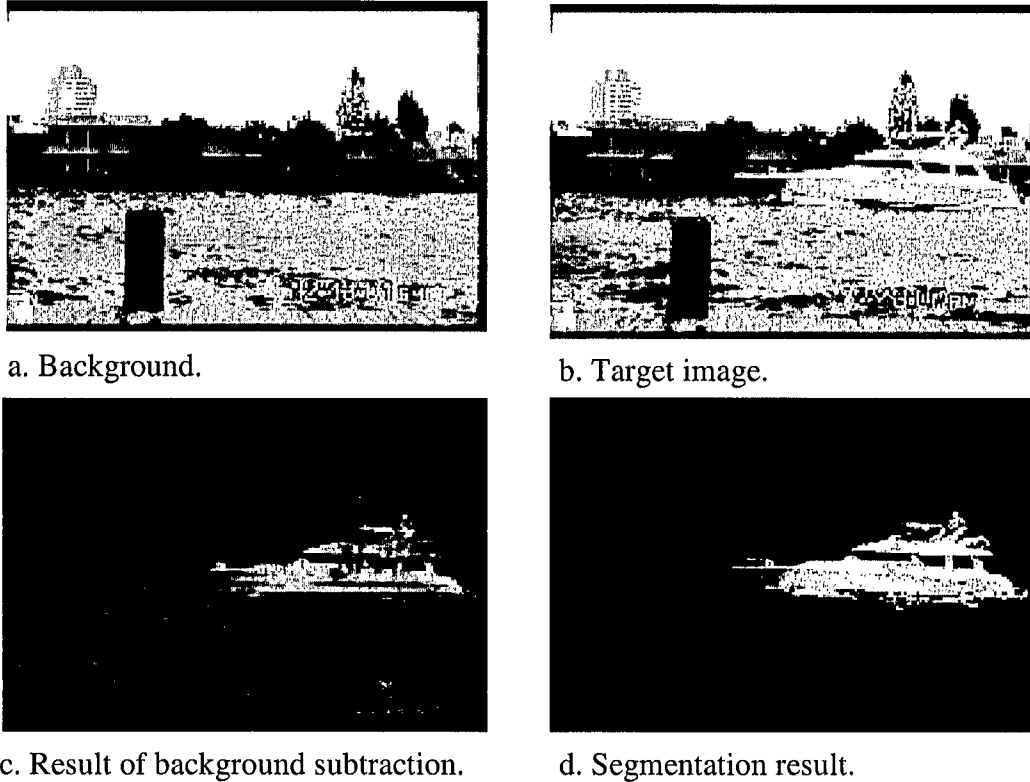


Figure 3.3. Illustration of background subtraction segmentation method.

3.1.2. Adaptive Progressive Thresholding(APT)

After some of the segmentation algorithms failed in visible small boat images as explained above, we decided to apply APT for target segmentation.

An adaptive technique is developed for computation of the threshold that depends only on the spatial characteristic of the image [25]. The segmentation procedure begins with the computation of an optimum threshold to distinguish the darker regions in the image. It is an automatic thresholding algorithm that works under all lighting conditions where pre-fixing of thresholding value is considered ineffective [27].

3.1.2.1 APT Algorithm

Given an image of size $N \times N$ pixels consisting of R gray levels, the spatial location of a pixel (x,y) would correspond to gray level $g(x,y)$. The image function as a mapping can be defined as:

$$G: N \times N \rightarrow P \quad (3.1)$$

A smoothed image can be generated by passing the image through a smoothing filter S of window size $s \times s$. The smoothed image is represented as:

$$g'(x, y) = \frac{1}{s^2} \sum_{(k,l) \in S} g(x+k, y+l) \quad \forall (x, y) \in N \times N \quad (3.2)$$

The thresholding method based on Otsu's method, which follows discriminant analysis. This partitions the gray levels in the image into two classes $P_0 = \{0, 1, 2, \dots, t\}$ and $P_1 = \{t+1, t+2, \dots, R-1\}$ at gray level t . The total variance (σ_T^2) and the between class variance (σ_B^2) are computed. The normalized between-class variance (σ_B^2 / σ_T^2) is maximized and the optimum threshold t^* is the gray level corresponding to the maximum normalized between-class variance

$$t^* = \text{Arg Max}_{0 \leq t \leq R} \left\{ \frac{\sigma_B^2}{\sigma_T^2} \right\} \quad (3.3)$$

$$\sigma_B^2 = \left(\sum_{i=0}^t \frac{n_i}{N^2} \right) \left(1 - \sum_{i=0}^t \frac{n_i}{N^2} \right) (\mu_1 - \mu_0)^2 \quad (3.4)$$

$$\sigma_T^2 = \sum_{i=0}^{R-1} (i - \mu_T)^2 \frac{n_i}{N^2} \quad (3.5)$$

$$\mu_T = \sum_{i=0}^{R-1} \frac{i \cdot n_i}{N^2} \quad (3.6)$$

μ_T is the total mean of the original image, N^2 is the total number of pixels present in the

image and n_i is the number of pixels at the i^{th} gray level.

$$\mu_0 = \frac{\sum_{i=0}^l \frac{i \cdot n_i}{N^2}}{\sum_{i=0}^l \frac{n_i}{N^2}}, \quad \mu_T = \frac{\mu_T - \sum_{i=0}^l \frac{i \cdot n_i}{N^2}}{1 - \sum_{i=0}^l \frac{n_i}{N^2}} \quad (3.7)$$

μ_0 is the class mean for Po and μ_1 is the class mean for P_1 and μ_T is the total mean for the initial image and Δ is the iteration number. The cumulative limiting factor is calculated for each progressive image as follows:

$$CLF(\Delta) = \frac{\sigma_B^2(\Delta)}{\sigma_T^2} \text{ for } \Delta \geq 1 \quad (3.8)$$

The image has a spatial distribution that is random in nature. The measure of randomness provides a number of ways for analysis of spatial data [28, 29]. It is assumed in spatial data analysis that the observations follow Poisson distribution, whose characteristic feature is that its mean is equal to its variance. A natural test for the Poisson distribution is the ratio of the sample mean, which is called relative variance [30, 31].

$$V_r = \frac{\sigma_T^2}{\mu_T} \quad (3.9)$$

It is observed that in all the experiments that were conducted the relative variance was found to be greater than 1. The test for the frequency distribution corresponds to a negative binomial distribution in the test image. Hence the separability factor for obtaining the optimum threshold is related to the relative variance and is defined as the square root of the inverse of that. That is,

$$SF = \sqrt{\frac{\mu_T^3}{\sigma_T^2}} \quad (3.10)$$

The cumulative limiting factor is compared with the separability factor and the process of

recursive thresholding is stopped when the condition

$$CLF(\Delta) < SF \quad (3.11)$$

is satisfied. The separability factor thus helps in stopping the algorithm at a point where the pixels are present as a dense cluster, which is a characteristic of the negative binomial distribution. This clustered nature of pixels causes the object region to be prominent with the background completely thresholded.

The concept of APT is shown in Figure 3.4. The original histogram of the image is shown in Figure 3.4.b. The Otsu thresholding image is shown in Figure 3.4.c. A particular threshold value is considered and all the gray levels above this value are discarded as illustrated in the “histogram after first thresholding”. This process is iterated many times till the separability condition is reached resulting in a histogram as illustrated in Figure 3.4.d. and the segmented image in Figure 3.4.e. with APT.

After the APT, there are still holes in the interior of the some of the targets. We filled these holes for better segmentation. During the preprocessing stage, we smoothed the images using standard median filtering on 3-by-3 pixel neighborhoods. Filtering reduced the graininess of the images so that fewer false edges were detected in the next step of segmentation [32]. The segmentation process itself eliminated many small regions of the image that resulted from noise, as we only dealt with line segments of significant length.

To remove some false segmentation part, morphological operations were applied to the segmented image for better segmentation result. Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size

[33]. The most basic morphological operations are dilation and erosion. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors.

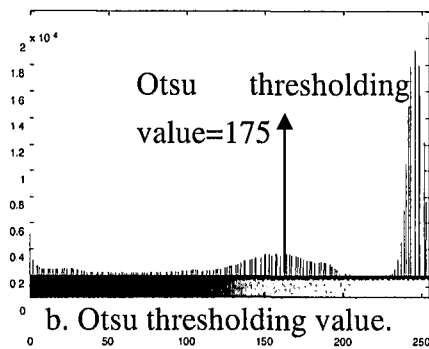
Dilation and erosion are often used in combination to implement image processing operations. For example, the definition of a morphological opening of an image is an erosion followed by a dilation, using the same structuring element for both operations. Morphological opening removes small objects from an image while preserving the shape and size of larger objects in the image. We applied a 7 pixel wide structuring element for erosion, and an 11 pixel wide structuring element for dilation. We chose a bigger structuring element for dilation so as not to lose any information from the target. Figure 3.4.f. illustrates the morphological opening operation.

Finally, outlined and segmented images are shown in Figure 3.4.g. and 3.4.h. Some of the segmented images are given in Figure 3.5.

Because of the morphological operator, some of the images' small superstructures parts (antennas, masts, ...) are lost, but these losses will not affect our classification result a lot, since we have to resize our visible image from 450x450 to 100x100 for PCA dimension reduction. The same types of morphological operators are used for all types of segmentation in this thesis.



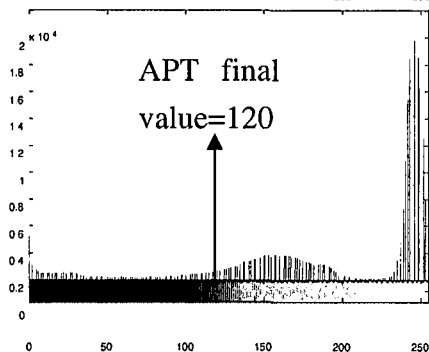
a. Visible image.



b. Otsu thresholding value.



c. Otsu segmentation result.



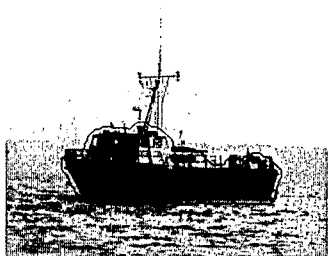
d. APT final value.



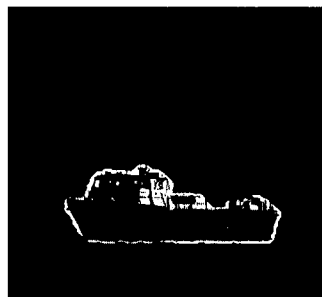
e. APT segmentation result.



f. Morphological opening result.



g. Outlined image.



h. Final segmentation result.

Figure 3.4. Visible image segmentation step-by-step with APT.

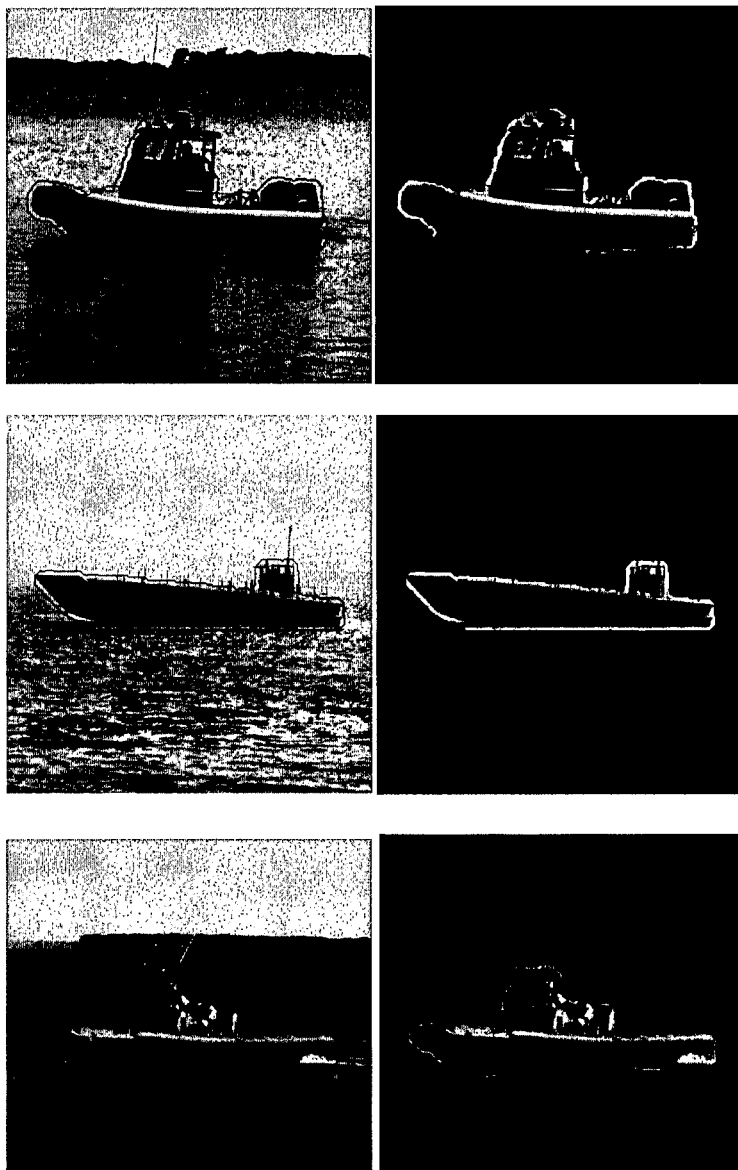


Figure 3.5. Some of the segmented image with APT.

3.2 IR Small Boat Image Segmentation

IR images are gray-level representations of infrared emissions that cannot be seen directly by a human observer. The colorization is therefore purely virtual, almost like in x-ray images, where bright gray-values represent dense materials [34]. Colorizations are

normally chosen such that warm (hot) regions are displayed in a bright gray-value, whereas cold regions are shown in darker gray-values, but this mapping can also be inverted. Furthermore, the relative infrared emissions depend on weather and climate conditions.

The radiometric data captured by the IR passive sensor consist of 1) energy emitted by thermal radiation from the object bodies, 2) atmospheric emission reflected from object surfaces. Therefore, the grayscale level of the IR images depends on the differences in temperature, emissivity and reflectivity of the objects in the scene. All of the scene temperature, emissivity and reflectivity contributions taken together can be represented at any point in the scene by an effective temperature at that point. The effective temperature is that at which a blackbody radiator would produce the measured irradiance near the point.

There are significant gray-scale level disparities between the IR and visible image. A good thermal emitter is not necessarily a good visual reflector. A surface of high visual reflectivity (white surface) in a visible band usually has low emissivity so that the visually bright objects may be dark in the thermal scene and vice versa. This is the reversal of contrast polarity between visible and IR images. However, a simple reversal of contrast polarity cannot remove all the grayscale level disparities in outdoor images. For instance, the sky is usually the brightest region in the visible image.

It is a dark region in the IR image because of the low temperature and the lack of reflectance, but clouds in the sky can have the same polarity of contrast in both IR and visible images: they are brighter than the sky in the visible image because of their higher reflectivity, and they are also brighter than the sky in the IR image because of its higher reflectivity and emissivity. In addition, shadows in the visible images are absent in the IR

images, which cannot be corrected by a reversal of contrast polarity.

Object extraction from an IR image background is of great interest both to the military and the commercial sector. IR image segmentation is a crucial task for image analysis and has been studied widely in the past. IR images in general are not that sharp in displaying objects than daylight images. In fact the noise floor present in the images is always much greater than in typical man-made pictures. Therefore, pure edge-based image segmentation will not be sufficient at all.

Most segmentation algorithms rely on changes in contrast or clustering the same colors only. Yet there seem to be no real one-and-for-all solution to the problem. A convenient and popular approach to object extraction is image thresholding for air and land base targets. Impressive and successful image segmentation algorithms have been proposed with a variety of techniques based on the image gray level histogram and edge based techniques [35][36].

But, these successful IR segmentation algorithms for land or air target have failed for our application in naval small boat images, since the background contains rough-textured regions like water and high clutter backgrounds. Generally, IR images have 3 other problems which are poor SNR-ratio, varied gray levels and naval ships or small boats have a homogenous bottom side.

The objects themselves (the ships) might be warmer (or brighter) than the surrounding water, or even colder (i.e. darker) than the rest of the image, so using predefined thresholding for image segmentation will not work.

3.2.1. SUSAN with APT for IR Small Boat Image Segmentation

SUSAN (Smallest Univalued Segment Assimilating Nucleus) evolved to be a general approach to low level image processing. The SUSAN approach is used for low

level image processing, specifically, edge detection (one dimensional feature detection), “corner detection” (two dimensional feature detection including corners, junctions, etc) and structure preserving noise reduction [37].

SUSAN was originally designed for edge and corner detection, but was also adapted for structure preserving denoising. SUSAN processes grayscale images in a local manner using an approximately circular window containing 37 pixels. The centre pixel in the window is called the nucleus. Neighboring pixels similar in intensity to the nucleus are grouped into an USAN, or Univalve Segment Assimilating Nucleus. The USAN creation process may appear to be a form of region growing using the nucleus as a seed, except that pixels in the USAN are not necessarily connected. The formation of an USAN is more related to clustering than segmentation.

The pixel similarity function is controlled by a brightness threshold t SUSAN first describes the USAN as accepting only pixels with an intensity difference of t units from the nucleus [37].

This can be considered crisp clustering, where similar pixels receive full weight, and dissimilar pixels receive no weight. The size of the USAN (Smallest Univalve Segment Assimilating Nucleus) is equal to the sum of weights given to pixels in the window, which for crisp clustering, equals the number of pixels assimilated. It was found that a fuzzy membership function gives better feature detection. This fuzzy function was “optimally” derived to have the form $e^{-\frac{\Delta}{t^2}}$, where Δ is the intensity difference from the nucleus. Figure 3.6 compares the crisp and fuzzy membership functions as a function of Δ . When the fuzzy form is used, all pixels have partial contributions to the size of the USAN. This is no longer an explicit segmentation of the local neighborhood.

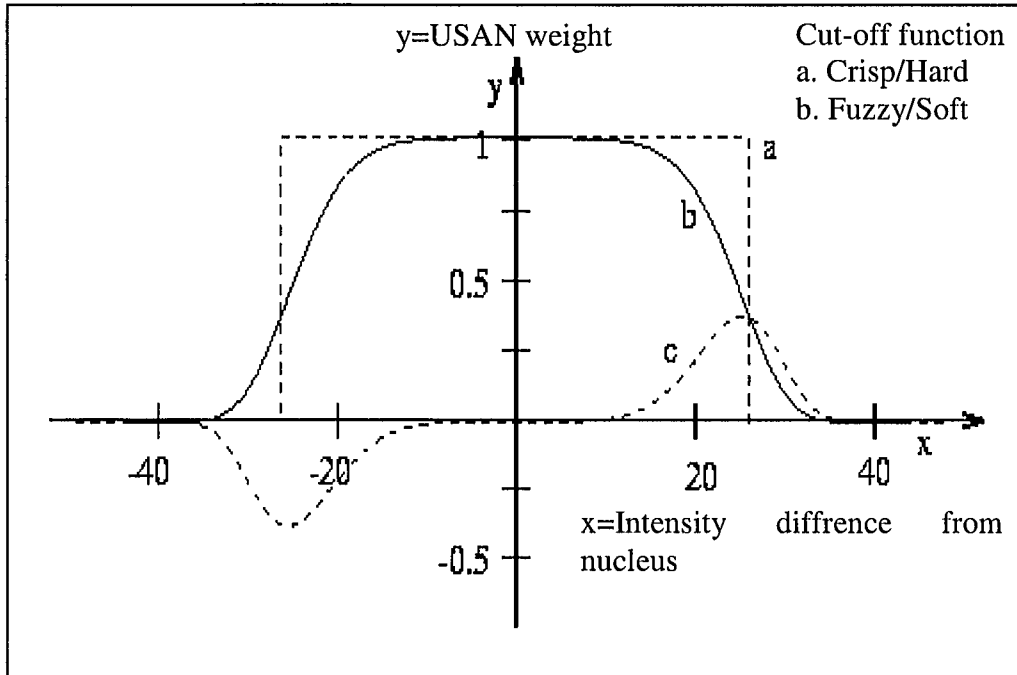


Figure 3.6.: Comparison of the hard and soft cut-off functions for pixel assimilation in the SUSAN algorithm.[37]

To perform feature detection, the USAN size for each pixel is plotted. On this surface homogeneous regions correspond to plateaus, edges to valleys, and corners to deeper valleys. SUSAN processes this USAN surface to determine the directions and strengths of edges and the positions of corners. Smith proposed the SUSAN feature detector to give results on a par with Canny's famous edge detection algorithm [38].

The main difficulty with SUSAN is that its performance is dependent on a user supplied brightness threshold, t . SUSAN is relatively insensitive to the choice of t , and advocates a default of 20 and t may be used to vary SUSAN's sensitivity to features. The value of t really controls the minimum feature contrast that can be detected.

My experiments show that it should be proportional to the natural level of intensity variation expected within image segments. This is the only way for the USAN to

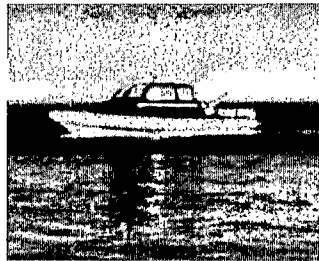
assimilate pixels in homogeneous regions and not produce false edge output.

SUSAN was applied to an IR ship image, after that most of the image clutter was suppressed. The segmentation algorithm is illustrated step by step in Figure 3.7. First, the image was filtered by low-pass median filtering and the image intensity value was enhanced with histogram stretch in Figure 3.7.b. The SUSAN-based functions to detect image edges and their orientation angle object boundaries are shown in Figure 3.7.c., but SUSAN applied result image has still 0-255 gray value. After that, APT was applied to the SUSAN result image in Figure 3.7.d. Binarization was made in this part and some of the holes were filled in the segmented image at the same part. After this, the segmentation process itself eliminated many small regions of the image that resulted from noise, as it dealt with line segments of significant length. To remove some false segmentation part, morphological operations were applied to the segmented image for better segmentation results in Figure 3.7.e. and f. Finally, the original image was outlined with the segmented image in Figure 3.7.g. and the segmented image outside was filled with 0's (zeros) in Figure 3.7.h.

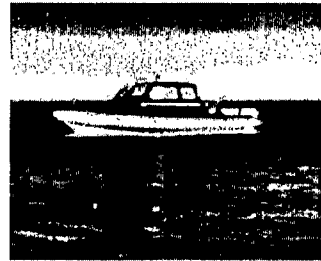
A set of 100 infrared images showing different levels of quality were used. The SUSAN-APT algorithm gave 65% satisfactory segmentation result and 20% of segmentation results are unsatisfactory and 15% of the images gave no segmentation results at all. Figure 3.8 shows some examples of the segmentation results. Unsatisfactory segmentation results are given in Figure 3.9.

The proposed SUSAN and APT algorithm are able to give 65 out of 100 suitable results. An additional 20 unsatisfactory segmentation results can be improved with Wavelet base enhancement techniques. The last 15 images result in wrong or even no segmentation borders at all. This is mainly due to the bad signal-to-noise ratio of the

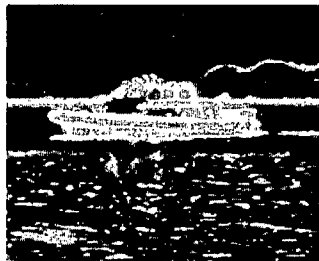
infrared images. It is presumed that even a human user might have difficulties in drawing a suitable segmentation border in images of the last type seen in Figure 3.10.



a. Original Boat image



b. Enhanced image



c. SUSAN applied image



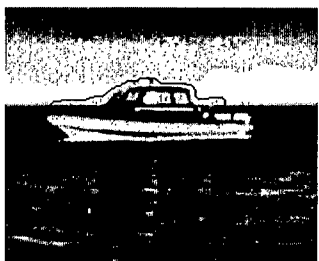
d. Binary image with filled holes



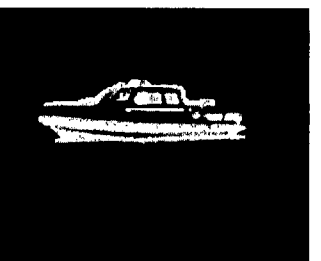
e. Erosion is applied



f. Dilation is applied after erosion



g. Outlined image



h. Segmented image

Figure 3.7. Segmented image step by step with SUSAN-APT.

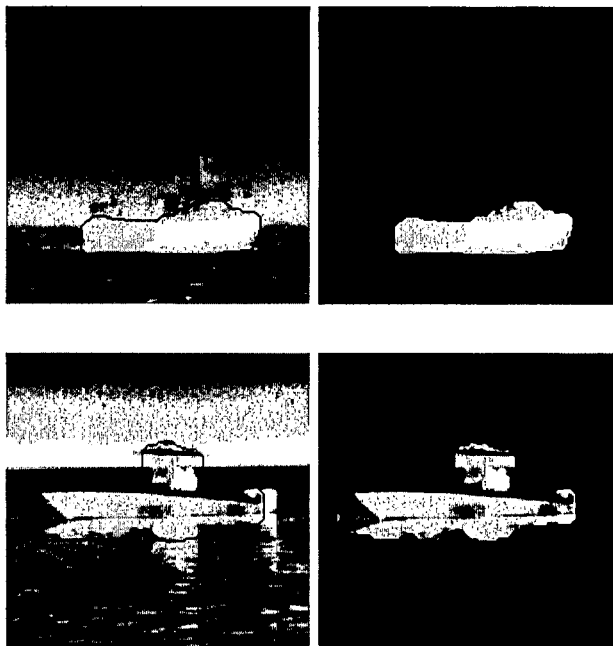


Figure 3.8. Example of segmented image with SUSAN-APT.

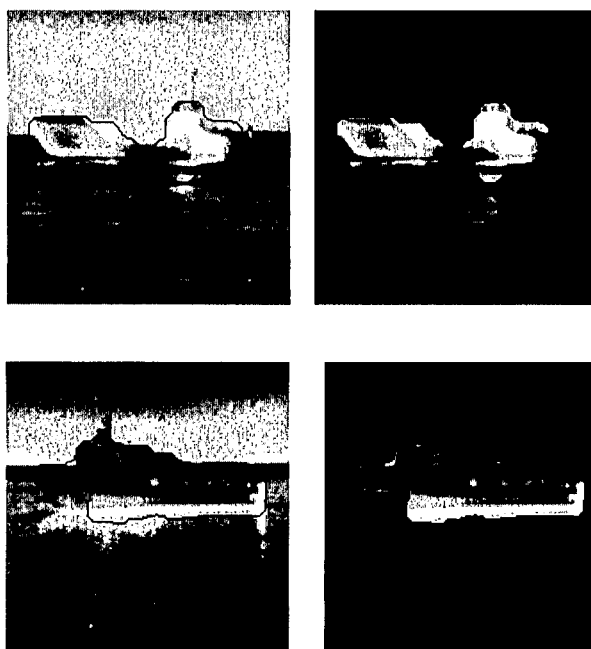


Figure 3.9. Example of unsatisfactory segmentation result with SUSAN-APT.

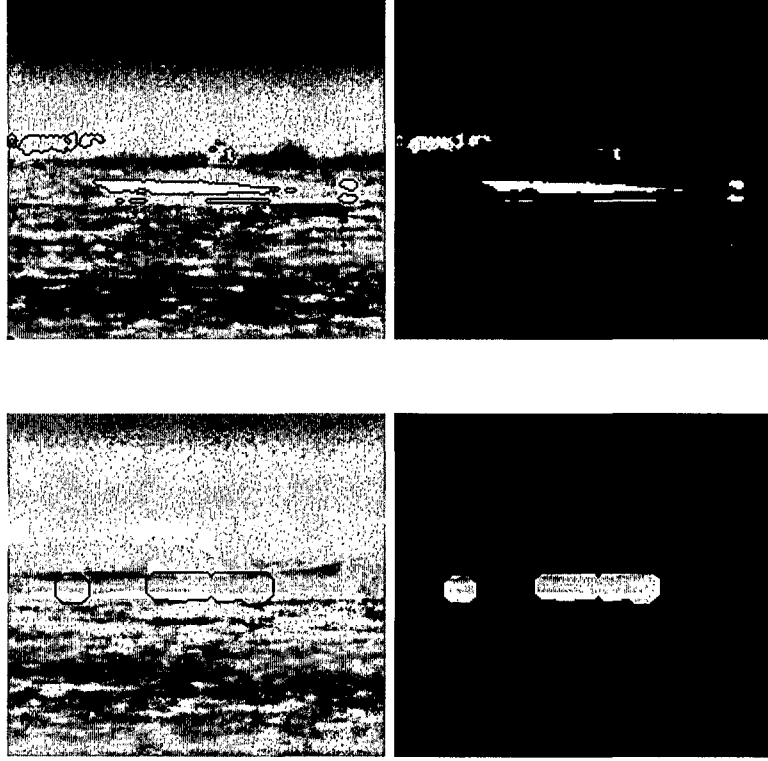


Figure 3.10. Example of bad or no segmentation result with SUSAN-APT.

3.2.2. Entropy Based Thresholding

Shannon [39] defined the entropy of a system as a function of the probability of occurrence of different states of the system. If a system has n different states with probability of occurrence p_i , $i = 1, 2, \dots, n$, $\sum_{i=1}^n p_i = 1$ then the gain in information from the occurrence of the event i is defined as $\Delta I = -\log_2 p_i$.

The expected value of such a gain in information is defined as the entropy of the system. Thus the entropy H of the system is: $\sum_{i=1}^n p_i \cdot \log_2 p_i$. Let $F = [f(x, y)]_{P \times Q}$ where $f(x, y)$ is the gray value at (x, y) ; $f(x, y) \in G_l = \{0, 1, \dots, L-1\}$, the set of gray levels. Let N_i be the frequency of the gray level i . Then $\sum_{i=1}^{L-1} N_i = P \times Q = N$. Following Shannon's

definition of entropy, entropy of the image can be defined as:

$$H = -\sum_{i=0}^{L-1} p_i \log_2 p_i, \quad p_i = N_i/N \quad (3.12)$$

for the image segmentation problem. Thus different images with identical histograms will result in same entropic value in spite of their different spatial distributions of gray levels. N.R. Pal et al. [40] have given a different formulation of entropy for images. We know that in an image pixel intensities are not independent on each other. This dependency of pixel intensities can be incorporated by considering sequences of pixels to estimate the entropy. In order to arrive at the expression of entropy of an image the following theorem due to Shannon can be stated:

Theorem: Let $p(s_i)$ be the probability of a sequence s_i of gray levels of length q , where a sequence s_i of length q is defined as a permutation of q gray levels. Let us define

$$H^{(q)} = -\frac{1}{q} \sum_i p(s_i) \log_2 p(s_i), \quad (3.13)$$

where the summation is taken over all gray level sequences of length q . Then $H^{(q)}$ is a monotonic decreasing function of (q) and $\lim_{q \rightarrow \infty} H^{(q)} = H$, the entropy of the image.

For different values of q we get different orders of entropy.

Case 1: $q = 1$, i.e. sequence of length one. If $q = 1$ we get $H^{(1)} = -\sum_{i=0}^{L-1} p_i \log_2 p_i$,

where p_i is the probability of occurrence of the gray level i . Such an entropy is a function of the histogram only and it may be called the “global entropy” of the image. Therefore different images with identical histograms would have same $H^{(1)}$ value irrespective of their contents.

Case 2: $q = 2$, i.e. sequences of length two. Hence, $H^{(2)} = -\frac{1}{2} \sum_{i=0}^{L-1} p_i \log_2 p_i$, where s_i is a sequence of gray level of length two, $= \sum_i \sum_j p_{ij} \log_2 p_{ij}$, where p_{ij} is the probability of co-occurrence of the gray levels i and j . Therefore $H^{(2)}$ can be obtained from the co-occurrence matrix. $H^{(2)}$ takes into account the spatial distribution of gray levels. Therefore, two images with identical histograms but different spatial distributions will result in different entropy, $H^{(2)}$ values. Expressions for higher order entropies ($q > 2$) can also be deduced in a similar manner. $H(i)$, $i \geq 2$, may be called the “local entropy of the image”. The power of the gray level co-occurrence approach is that it characterizes the spatial interrelationships of the gray levels in a textural pattern and can be done so in a way that it is invariant under monotonic gray level transformations.

Its weakness is that it does not capture the shape aspects of the gray level primitives. Hence, it is not supposed to work well for textures composed of large area primitives. Also, it cannot capture the spatial relationships between primitives that are regions larger than a pixel.

3.2.2.1. Minimum Cross Entropy Thresholding

This algorithm has been described by Li et al [41]. For a histogram h defined on the gray level range $[0, L-1]$, the zeros and the first moments of the foreground and background portions of the thresholded histogram are respectively,

$$m_{0a}(t) = \sum_{i=0}^{t-1} h(i) \quad , \quad m_{0b} = \sum_{i=t}^{L-1} h(i) \quad (3.14)$$

$$m_{1a}(t) = \sum_{i=0}^{t-1} ih(i) \quad , \quad m_{1b} = \sum_{i=t}^{L-1} ih(i) \quad (3.15)$$

The portions means are defined as

$$\mu_a(t) = \frac{m_{1a}(t)}{m_{2a}(t)}, \quad \mu_b(t) = \frac{m_{1b}(t)}{m_0(t)} \quad (3.16)$$

The minimum cross entropy method selects the threshold which minimizes the cross entropy of the image and its segmented version. The criterion function is found to be:

$$\eta(t) = -m_{1a}(t) \log(\mu_a(t)) - m_{1b}(t) \log(\mu_b(t)) \quad (3.16)$$

Thus the optimal threshold is given by the minimizer of equation 3.16,
 $t_{op} = \arg \min_t \eta(t)$

Minimum Cross entropy, Local entropy, Joint entropy and Global entropy thresholding methods were applied to the IR small boat image database. The Minimum cross entropy method gave the highest segmentation result from these four methods. After segmentation, morphological dilation operation is applied to segmented images so as not to lose any target information and 35% of image database segmentation results are satisfactory. Some of the segmented image is given Figure 3.11.

As a result, Entropy based segmentation is not applicable to our image database. Although the entropy based thresholding algorithms were proposed for IR image and claimed to give good results in the literature [52, 53], they did not give satisfactory segmentation results for our application.

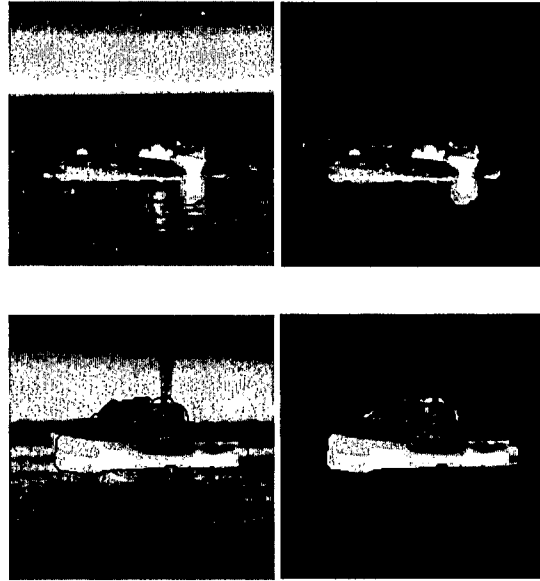


Figure 3.11. Some of the segmentation result with entropy based thresholding.

3.2.3. Graph-cut Segmentation Techniques

A wide variety of computer vision tasks can be expressed in terms of a labeling problem to pixel regions on noisy image data. Stereo reconstruction, image restoration and segmentation are among such tasks which can be regarded as an optimization problem in the presence of uncertainties. For such labeling problems a specialized graph can be defined corresponding to the energy function to be minimized which fits into an elegant minimum cut optimization framework. It can be shown that we can solve the maximum cut problem without exhaustive computation using maximum flow algorithms

The use of graphs to solve energy minimization problems has become more and more popular in the context of low level computer vision [42]. Many problems in computer vision can be reformulated as an energy minimization problem. Energy minimization has in the past been computed by using dynamic programming (which only works in very simple cases) and simulated annealing (which is very slow). Recently there

are many approaches to energy minimization by using graph-cut techniques, and in most cases each graph has to be specially created for solving a specific energy minimization problem [43]. Kolmogorov and Zabih look at the types of functions that can be minimized by graph-cut and they give a general construction of the graphs that can solve different classes of problems [43].

All segmentation techniques using graphs use some form of a graph-cut algorithm to segment the graph into two regions, and in doing so minimize the energy. Boykov and Kolmogorov provide a comparison between current Min-Cut/Max-Flow algorithms with regards to their efficiency [42]. This paper includes Goldberg-Tarjan style “Push-Relabel” methods and Ford-Fulkerson style “Augmenting Path” methods. It also introduces a new algorithm which they claim works several times faster than all other known methods in most cases. This technique uses energy minimization in the context of image restoration, stereo and image segmentation in order to compare the speeds of the different methods.

3.2.3.1 Graph-cut for Image Segmentation

Boykov and Jolly introduce a segmentation technique that uses a graph to represent the image, and a Min-Cut/Max-Flow algorithm to segment the graph [44]. Pixels in the image are represented by nodes on the graph. The edge weights on the graph are defined by a cost function, which is defined by region and boundary information found in the image. A Min-Cut/Max-Flow algorithm is used to segment the image by minimizing the cost function. This technique uses an intensity histogram to store region information found in the image, and only works on grayscale images.

Lazy Snapping was introduced by Li [46] as an interactive means of segmentation based on graph minimization techniques very similar to the method of Boykov and Jolly

[44]. It involves two major steps. The first step is the object marking stage in which certain pixels are marked as either background or foreground. These are hard constraints which have to be met during the segmentation process. The Graph-cut is used together with an over-segmentation technique which greatly increases the speed of segmentation and can provide segmentation results quickly to the users. The second step is a boundary editing step in which individual vertexes on the segmentation can be moved around until the user is satisfied.

Interactive segmentation is becoming more and more popular, and it is far preferred over fully automatic segmentation methods that are hard to perfect [44]. There are a few methods that try to perform automatic segmentation. Blake describes a method of image segmentation that operates with very few hints from the user [45].

Most image segmentation techniques use parameters set by the user. In this case an algorithm is used to learn parameters from the image data and then perform the segmentation based on these parameters. The segmentation technique proposed by Boykov and Jolly is used in this experiment, but the parameters needed for segmentation are learned by a pseudo-likelihood algorithm [44]. A database of images with the correctly segmented results is used to test the approach. The percentage error using interactive approaches was considerably lower than this approach [45].

3.2.3.2. Graph-cut Algorithm

The Graph-cut algorithm is generally based on energy minimization. The energy function consists of two parts: a probability term P and a contrast term C . The influence of each part can be adjusted with a weighting factor γ .

$$E = P + \gamma.C \quad (3.17)$$

Gray-valued histograms are used to compute two probabilities for each pixel from the image: one for the background and one for the object part, to define how well the pixel color fits into the background/object histogram. A cost measure $P_{z,\alpha}$ for assigning a pixel z to the wrong class α background or object has to be defined, for example the log-likelihood function of their probability to fit into each histogram.

Graph-cut techniques are closely related to the theory of Markov Random Fields. The optimization of the Graph-cut energy function of Equation 3.17 is commonly done using the Markov Random Field (MRF) model.

The total sum over all costs is the first objective to be minimized:

$$P = \sum_{z \in I} P_{z,\alpha} \quad (3.18)$$

The second (optional) smoothing term describes an arbitrary measure of contrast in the image. Usually this is calculated from the image gradient between all neighboring pixels N , but other edge detection techniques are also possible. At least a kind of an edge map, where edges have higher values than non-edges, has to be provided to the algorithm. It is used to make the segmentation boundary fall on edges more accurately.

$$C = \sum_{(p,q) \in N} C_{p,q} \cdot \delta(\alpha_p, \alpha_q) \quad (3.19)$$

where

$$\delta(\alpha_p, \alpha_q) = \begin{cases} 1, & \alpha_p = \alpha_q \\ 0, & \text{otherwise} \end{cases} \quad (3.20)$$

All contrast terms $C_{p,q}$ of pixels lay on the segmentation border, where the pixel p and q belong to different classes α , expressed by the indicator function $\delta(\cdot)$ are summed

up. The minimization criterion is to find the shortest possible segmentation border that gives the smallest sum over its contrast terms. The contrast between neighboring pixels p and q can be expressed as

$$C_{p,q} = \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2 \cdot \|p - q\|}\right) \quad [44] \quad (3.21)$$

where the norm $\|p - q\|$ indicates the (spatial) Euclidean distance between two neighboring pixels p and q . In a 4-way connectivity this is, of course, always one. The variance σ^2 over all differences in intensity can be seen as the noise floor present in the picture.

From these two properties of every pixel, one belonging to the object or the background and the other being an edge or not, a graph is built. More precisely, a so-called S/T-graph is built, where the two terminals S and T represent the object and the background. Edges from and to these terminals are weighted with the corresponding object/background probabilities respectively. Neighboring pixels are connected with edges in 4-way neighborhood, weighted with the corresponding contrast.

For the initial background/object distribution, two regions containing background parts and object parts of the image have to be first specified by the user. The remaining unknown parts of the image will be segmented based on these histograms with a standard minimum-cut/maximum-flow algorithm. This resulting segmentation is proved to be the global minimum that can be found with the two given energy terms. Therefore an optimum solution for the segmentation task has been found and the algorithm terminates. The pixels that are connected to the S terminal belong to the object and the pixels connected to the T terminal represent the background.

The results shown in Figure 3.12 are binary hard segmentations between object

and background. In case of failures, the initial specifications for object and background have to be refined and the Graph-cut algorithm has to be applied again. There is no user interaction after the algorithm has been started.

3.2.3.3. Grab-cut for Image Segmentation

Grab-cut is an iterative image segmentation technique based on the Graph-cut algorithm [48]. Grab-cut extends Graph-cut to color images and to incomplete trimaps. These developments greatly increase the usefulness of Graph-cut [48]. User interaction is simplified to drawing a rectangle around the desired foreground, followed by a small amount of corrective editing [47]. The inclusion of color information in the Graph-cut algorithm and the iterative learning approach increases its robustness [48]. Thus, Grab-cut is a very promising image editing tool for foreground extraction.

Like most segmentation techniques, Grab-cut uses information encapsulated in the image. Most segmentation techniques make use of either edge information or region information in the image [49]. Grab-cut makes use of both edge and region information. This information is used to create an energy function which, when minimized, produces the best segmentation results.

The “Grab-cut” segmentation process can be broken down into four stages. Figure 3.12 outlines this process.

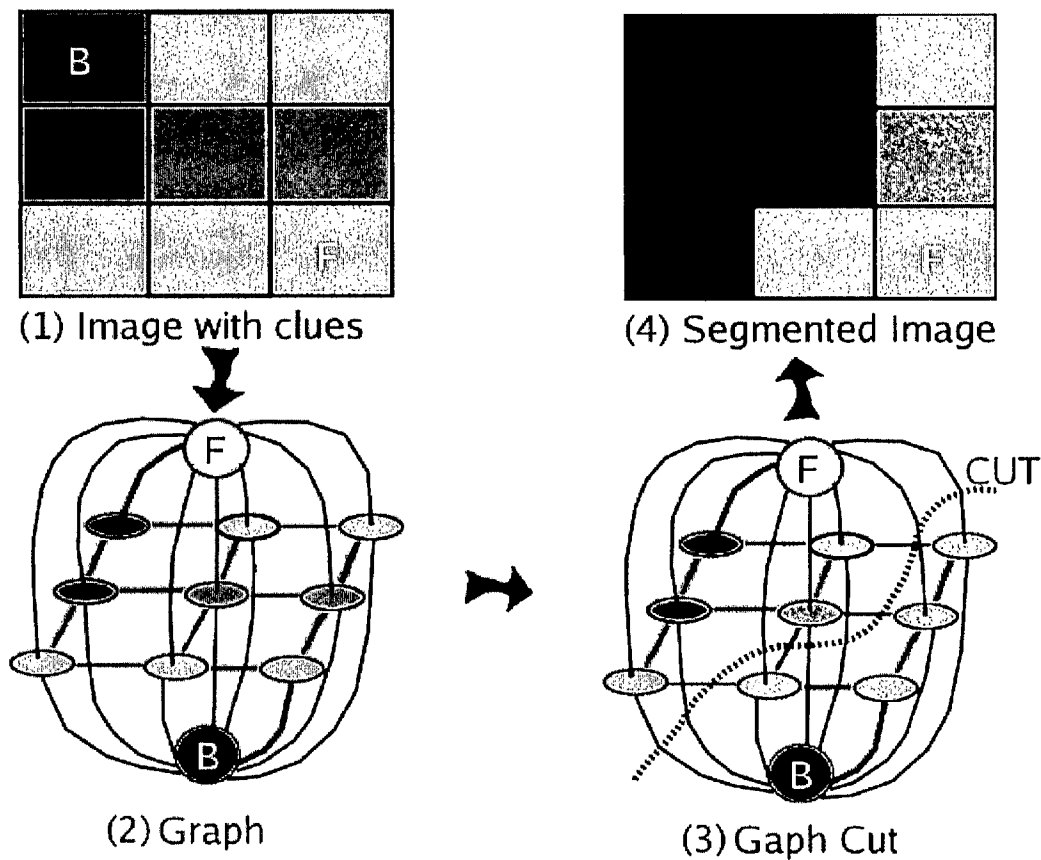


Figure 3.12. Explanation of Grab-cut segmentation process.

1. In order for the segmentation to take place the “Grab-cut” algorithm needs to have some idea as to which parts of the image are foreground and which parts are background. This can be done by labeling pixels as either foreground or background. Image 1 in Figure 3.12 shows how the top left pixel has been labeled as background and the bottom right pixel as foreground. This first stage of labeling pixels is known as the clue marking stage. All pixels labeled in this stage may not change their labeling later on during segmentation, and so these labels are hard constraints.

2. A graph is created, where nodes in the graph represent pixels in the image. In

addition to all the pixel nodes, two special nodes are also created. These are the Sink and Source nodes. Every pixel node in the graph is connected to the Source and Sink nodes. The Source node represents the foreground of the image, and the Sink node the background. In image 2 in Figure 3.12 the F node is the foreground or Source node and the B node is the background or Sink node. The weights between pixel nodes in the graph are determined by edge information in the image. Thus, a strong indication of an edge between two pixels (a large difference in pixel color) results in a very small weight between two pixel nodes.

The pixels labeled as either foreground or background in the clue marking stage is used to create a foreground and background region model. These models determine the weights between pixel nodes and the Source and Sink nodes. These weights are calculated by determining the probability of the pixel node being part of the background or foreground model.

3. The Source and Sink nodes must be separated in order to perform segmentation. This is done by cutting the links between pixel nodes and the Source and Sink nodes. An energy function E is created which is defined by the sum of all the edge weights that are cut. The Min-Cut/Max-Flow algorithm created by Boykov and Kolmogorov is used to minimize this energy function and segment the graph. [47] This algorithm determines the minimum cost cut that will separate the Source and Sink nodes. The cost of the cut is determined by the sum of all the weights of the links that are cut.

4. After stage 3 is complete all pixel nodes will only be connected to either the Source or Sink node. The purpose of this stage is to transfer the graph information back to the image. All pixel nodes in the graph which are connected to the Source node cause their representative pixels in the image to become part of the foreground. All pixel nodes which

are connected to the Sink node cause their representative pixels in the image to become part of the background.

3.2.3.4. Region Modeling Using Grayscale Histograms

Pixels marked as either foreground or background during the clue marking stage are used to model region information by building up foreground and background histograms. Histograms are built up by noting the frequency with which a pixel occurs within a group of pixels. For grayscale images the intensity of each pixel can range between 0 and 255. A histogram with 256 bins is created for grayscale images so that each bin holds the frequency at which pixels with the intensity of the bin number occur in an image. Figure 3.13 shows how foreground and background information is used to create histograms in a grayscale image.

The weights between a pixel node and the Source node are determined by the probability that the pixel node lies inside the foreground histogram. Similarly the weight between a pixel node and the Sink node is determined by the probability that it lies inside the background histogram.

3.2.3.5 The Graph Weights

The edge weights of the graph are defined in Table 3.1.

Table 3.1. Edge weights of the graph

Edge	Weight	Applied to
Between nodes	γV	All neighboring pixels
	U(Background)	Pixels with no hard constraints
	K	Foreground constraints
	0	Background constraints
Between node and Sink terminal	U(Foreground)	Pixels with no hard constraints
	0	Background constraints
	K	Foreground constraints

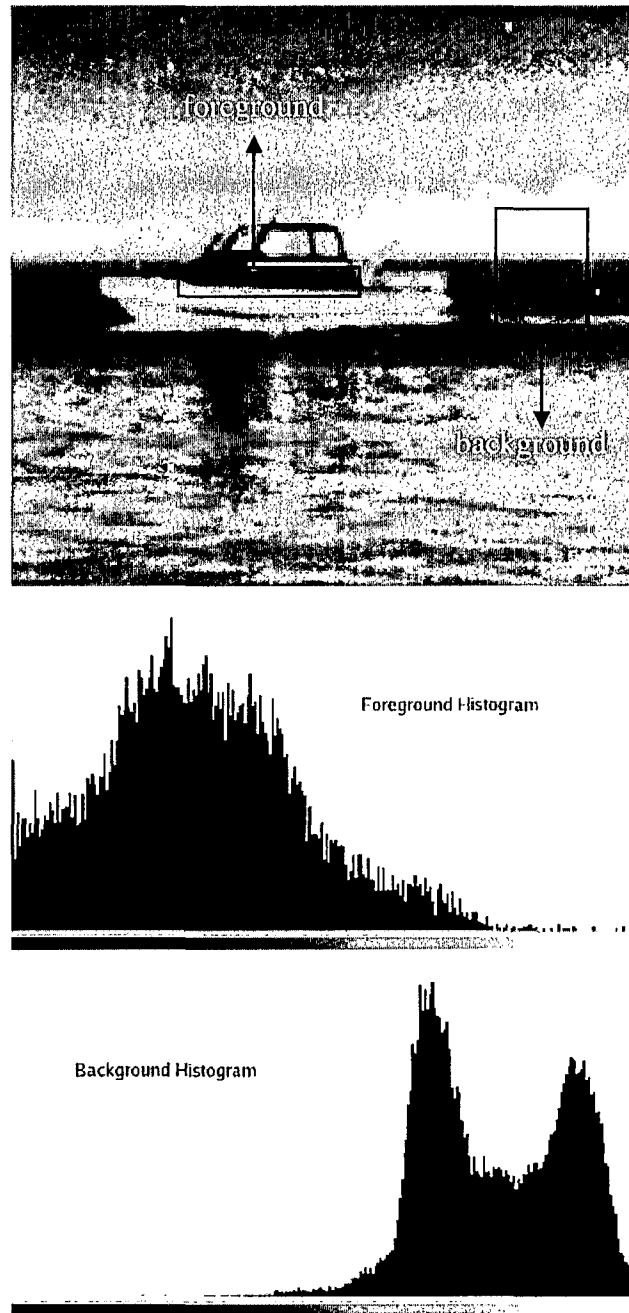


Figure 3.13. Histogram models in a grayscale image.

In Table 3.1, K is a constant that is set to a large value so that edges between nodes that are labeled as foreground or background during the clue marking stage are never cut. The constant γ is defined in Equation 3.22.

U returns the probability that pixel intensity lies in either the foreground or background, and is defined as:

$$U(\text{Background}) = \gamma * \log(\text{HistogramBack}(\text{Pixel})) \quad (3.22)$$

$$U(\text{Foreground}) = - \gamma * \log(\text{HistogramFore}(\text{Pixel})) \quad (3.23)$$

In Equation 3.22 HistogramBack, returns the probability that a pixel lies inside the background histogram. In Equation 3.23 HistogramFore, returns the probability that a pixel lies inside the foreground histogram.

V measures the difference in intensity between neighboring pixels and is defined as:

$$V = - \exp(-B(I_{\text{pixel1}} - I_{\text{pixel2}})^2) \frac{1}{\text{distance}} \quad (3.24)$$

In Equation 3.24, B is a constant, I pixel refers to the intensity of a pixel, and distance is the distance between the two neighboring pixels.

Setting the weights in this way causes the following:

- The larger the difference in intensity between neighboring pixels, the smaller the cost, and the greater the chance of the edge being cut. Thus edges are more easily cut than regions of intensity coherence.

- Pixels which are labeled during the clue marking stage are never relabeled.

This is due to the large constant K which ensures large edge weights.

- The region costs of unlabelled pixels between the Source and Sink nodes are defined by the probability that their intensity lies in the foreground and background histogram. A high probability value results in large cost and thus region coherence is established.

In Figure 3.13, pixels with a high intensity value will have a higher probability of occurrence in the background histogram than in the foreground histogram, and so the edge

weight between the pixel node and the Sink node will be stronger than the edge weight between the pixel node and the Source node. Pixels with a low intensity value will have a higher probability of occurrence in the foreground histogram than in the background histogram, and so the edge weight between the pixel node and the Source node will be stronger than the edge weight between the pixel node and the Sink node.

3.2.3.6. Region Modeling Using Gaussian Mixture Model

In this design pixels marked as either foreground or background during the clue marking stage are used to model region information by building up Gaussian Mixture Models. A Gaussian Mixture Model is the weighted sum of several Gaussians. To understand Gaussian Mixture Models the concept of a Gaussian has to be understood.

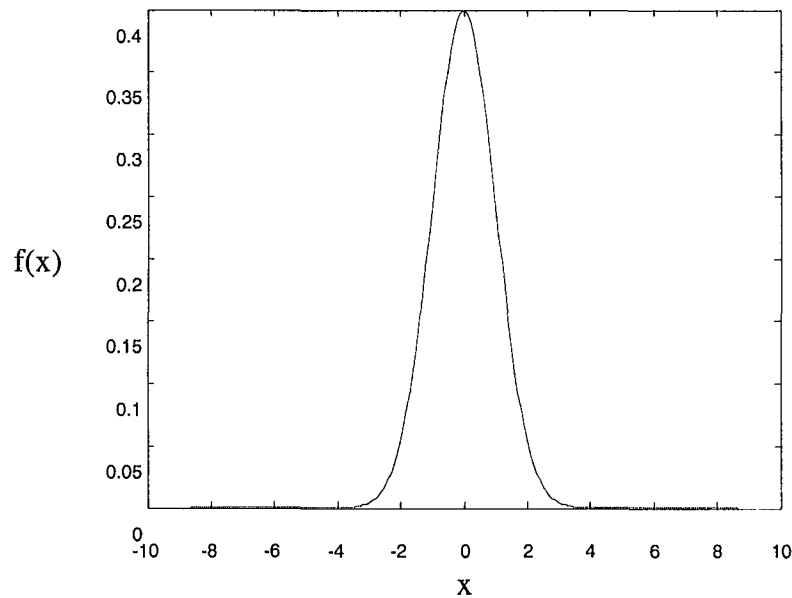


Figure 3.14. A Gaussian probability function.

3.2.3.6.1. Gaussians

A Gaussian is similar to a histogram in that it also returns a probability. Given a value x , the Gaussian probability density function returns the probability that x occurs within the distribution. The formula for the basic 1D Gaussian probability density function is as follows:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-(x - \mu)^2 / 2\sigma^2) \quad (3.25)$$

In equation 3.25, σ is the standard deviation and μ the average or mean. This formula produces Figure 3.14. In this case $\sigma = 1$ and $\mu = 0$. The 1D Gaussian probability density function can be extended to 3D, and is as follows:

$$f(x) = \frac{1}{2\pi\sqrt{\sum_k}} \exp(-\frac{1}{2}(I_z - \mu_k)^T \sum_k^{-1} (I_z - \mu_k)) \quad (3.26)$$

where the term I_z now reflects a three-valued RGB color of the pixel z . The μ_k are the mean color of each component and \sum_k are full-covariance matrices reflecting color dependencies between the three color layers. For the purpose of modeling color, these vectors store red, green and blue color values. This formula will accept a color, which is made up of red, green and blue values, and returns the probability that color occurs in the density distribution. This is very useful as a 3D Gaussian probability function is able to encapsulate 3 values into the single concept of a color. Histograms are only able to encapsulate 1 value.

3.2.3.6.2. Gaussian Mixture Model

A Gaussian Mixture Model is the weighted sum of several Gaussians.

$$\text{GMM} = \sum_{i=0}^n \pi_i G_i \quad (3.27)$$

Equation 3.27 defines how the Gaussian Mixture Model, GMM, is built up by summing i Gaussians, where each Gaussian, G_i , is weighted by a factor of π_i . Rother used 5 Gaussians each to model the foreground region and the background region [45].

The five Gaussians which make up the foreground region are each centered around a different color found in the foreground region. These colors are the most common colors found in the foreground. The same occurs for the five Gaussians making up the background. The weighting of each Gaussian depends on how common the color is within the foreground or background region.

The process of selecting the most common color from a foreground or background region in an image is performed by an EM clustering algorithm. The EM clustering algorithm clusters the Gaussians around the most common colors and in doing so determines the means, covariance matrices and mixture parameters of the Gaussians. Figure 3.15 shows an example of 1D Gaussian Mixture Model which is made up of 5 Gaussians.

A 1D Gaussian Mixture Model has been used as it employs the same concept as 3D Gaussian Mixture Models, but it is easy to visualize and illustrate. Each peak in this diagram is centered around a common color in the region being modeled.

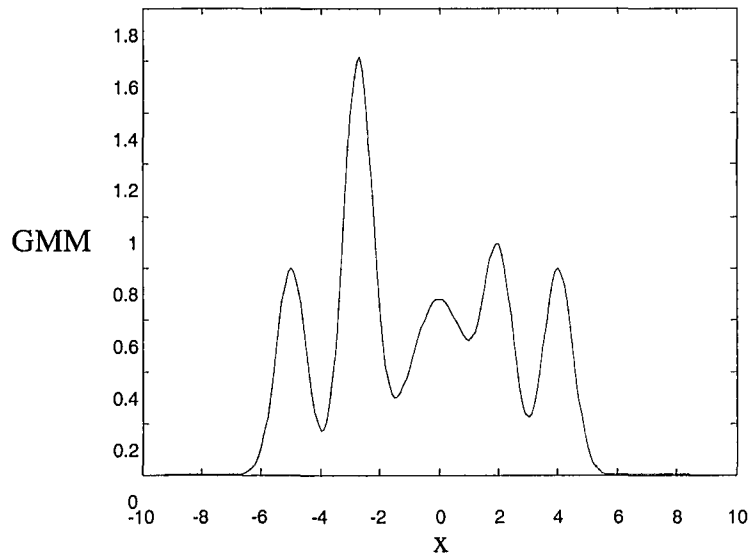


Figure 3.15. A 1D Gaussian mixture model.

3.2.3.7. Clue Marking Method

The foreground and background clue marking method differs from the grayscale case. In the grayscale mode, pixels could be marked as either foreground or background. In this case, the user drags a rectangle around the foreground object, and the inverse of the selection is set as hard background. Unlike the grayscale implementation, this implementation iteratively solves for the best segmentation, and the pixels inside the rectangle are labeled as unknown and allowed to change between foreground and background on each iteration. Figure 3.16 shows how a rectangle is drawn around the object of interest, in this case an eagle. All pixels inside this rectangle are set as unknown, and all pixels outside the rectangle are set as hard background.

Segmentation of an image is performed as follows:

- A background Gaussian Mixture Model is created from the hard labeled pixels in the clue marking stage.

- A foreground Gaussian Mixture Model is created from pixels inside the user drawn rectangle.

- The following steps are then repeated until the energy function converges:

1. Estimate the means, covariance matrices and mixing parameters of the background and foreground Gaussian Mixture Models based on the current segmentation.

The EM algorithm is used for this purpose

2. Assign to every pixel a Gaussian component from the foreground and background model.

Each pixel is assigned a background and foreground Gaussian. This is done by determining which Gaussian in the foreground Gaussian Mixture Model produces the highest probability given the pixels color, and likewise for the background Gaussian Mixture Model.

3. Create the graph and segment it, marking pixels as either foreground or background. Region weights in the graph are determined by the Gaussians assigned to each pixel. The weights between a pixel node and the Source node are determined by the probability that the pixel node lies inside the foreground Gaussian. Similarly the weight between a pixel node and the Sink node is determined by the probability that it lies inside the background Gaussian.

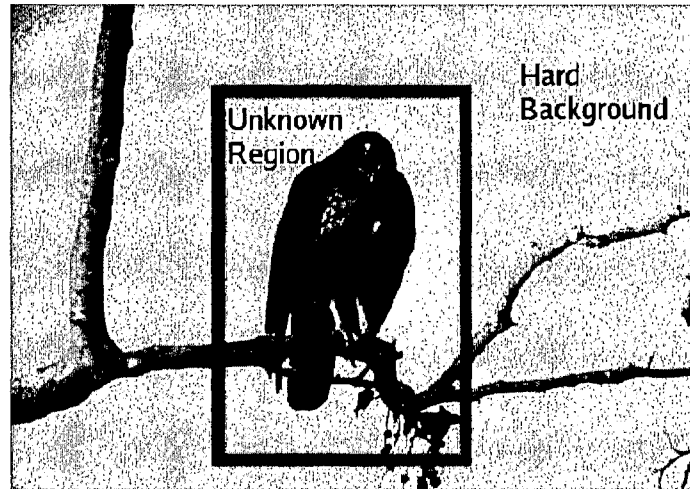


Figure 3.16. Pixel labeling when using gaussian mixture models.

3.2.3.8. The Graph Weights

The edge weights between nodes in the graph are defined in Table 3.2.

Table 3.2. Edge weights between nodes in the graph.

Edge	Weight	Applied to
Between nodes	γV	All neighboring pixels
Between node and Sink terminal	0	Pixels with hard Background constraints
	U(Foreground)	Temporary Foreground constraints
	U(Foreground)	Temporary Background constraints
Between node and Sink terminal	K	Pixels with no hard constraints
	U(Background)	Temporary Background constraints
	U(Background)	Temporary Foreground constraints

The only hard constraints while working with Gaussians are the initial background labeling. These constraints are enforced by the large constant K . During each iteration, the remaining pixels may change their labeling. The weights between pixel or the Source and Sink nodes are determined by the Gaussian Mixture Models. Weights between pixels and the Source terminal are determined by the foreground Gaussian Mixture Model and weights between pixels and the Sink terminal by the background Gaussian Mixture Model. The probability that a pixel lies in a Gaussian Mixture Model is taken as the maximum probability of the pixel which lay in any of the Gaussians making up the Gaussian Mixture Model. Thus even though there are 5 Gaussians making up each mixture model, only one of the Gaussians in each model will be used to set the region weights for a pixel node in the graph. The weight terms are defined as follows:

$$U(Foreground) = -\log(\pi_F) + \log\left(\sum \left| \right|_F\right) / 2 + f(pixel) / 2 \quad (3.28)$$

$$U(Background) = -\log(\pi_b) + \log\left(\sum \left| \right|_B\right) / 2 + f(pixel) / 2 \quad (3.29)$$

$$V = \gamma * \exp(-B(pixel1 - pixel2)^2) * \frac{1}{dis\ tan\ ce} \quad (3.30)$$

$$(pixel1 - pixel2)^2 = (pixel1_{RED} - pixel2_{RED})^2 + (pixel1_{GREEN} - pixel2_{GREEN})^2 + (pixel1_{BLUE} - pixel2_{BLUE})^2 \quad (3.31)$$

Equation 3.31 determines the difference in color between two pixels, and is taken as the Euclidean distance in RGB color space. In equations 3.28 and 3.29 $f(pixel)$ refers to the probability that the pixel lies in the foreground or the background Gaussian Mixture Model.

3.2.3.9. Region Modeling Using Color Histograms

This design is the same as the grayscale version, except here histograms are built

up for each color channel. The red, green and blue channels in color images have a possible range of 0 to 255, so histograms are built up in exactly the same way as in grayscale images. The probability that a pixel lies in the foreground region is calculated by the sum of the probabilities that its red value lies in the red foreground histogram, its blue value in the blue foreground histogram and its green value in the green foreground histogram. The weights between a pixel node and the Source node are determined by the probability that the pixel node lies inside the foreground region.

Similarly the weight between a pixel node and the Sink node is determined by the probability that it lies inside the background region. Color histograms do not fully represent color regions in an image. This is because the histograms are separate, and have no concept that a color is made up of 3 RGB values. Gaussian Mixture Models are more appropriate for representing color regions as they cluster actual colors, not individual RGB values. Table 3.3 shows color image segmentation times. Figure 3.17 shows some color image segmentation results with the proposed techniques.

Table 3.3. Color image segmentation times.

Image Size	200x200	540x340	800x600
Segmentation time	25.34 sec	36.76 sec	44.54 sec



Figure 3.17. Color Image Segmentation.

3.2.3.10. The Graph Weights

This design is an extension of the grayscale case, and all the edge weights are defined as in the grayscale case. In this design three histograms are used to model region data: a histogram for each of the red, green and blue color channels. The U and V terms of the energy function are defined as follows:

$$\begin{aligned}
 U(\text{Foreground}) = & -\gamma * U(\text{RedHistogramFore}(\text{Pixel}_{\text{RED}})) \\
 & -\gamma * U(\text{GreenHistogramFore}(\text{Pixel}_{\text{GREEN}})) \\
 & -\gamma * U(\text{BLUEHistogramFore}(\text{Pixel}_{\text{BLUE}}))
 \end{aligned} \tag{3.32}$$

$$U(\text{Background}) = -\gamma * U(\text{RedHistogramBack}(\text{Pixel}_{\text{RED}}))$$

$$\begin{aligned}
& -\gamma * U(\text{GreenHistogramBack}(\text{Pixel}_{\text{GREEN}})) \\
& -\gamma * U(\text{BLUEHistogramBack}(\text{Pixel}_{\text{BLUE}}))
\end{aligned} \tag{3.33}$$

$$V = \gamma * \exp(-B(\text{pixel1} - \text{pixel2})^2) * \frac{1}{\text{distance}} \tag{3.34}$$

$$\begin{aligned}
(\text{pixel1} - \text{pixel2})^2 = & (\text{pixel1}_{\text{RED}} - \text{pixel2}_{\text{RED}})^2 + \\
& (\text{pixel1}_{\text{GREEN}} - \text{pixel2}_{\text{GREEN}})^2 + (\text{pixel1}_{\text{BLUE}} - \text{pixel2}_{\text{BLUE}})^2
\end{aligned} \tag{3.35}$$

3.2.3.11. The “Grab-cut” Image Segmentation Implementation for IR Grayscale

Image

The graph class is used to create the graph. It has methods to set up the edge weights between nodes in the graph, and uses the Min-Cut/Max-Flow method to segment the graph. The neighbor's class is used to store the neighboring relationship between pixels in the image. During the evaluation of the weights between pixel nodes in the graph, information about neighboring pixels nodes is needed. This includes the distance between the pixels, and their x and y offsets relative to their neighbors.

The neighbor's class stores all the relationships between pixels, and simplifies the process of evaluating edge weights. A neighboring labeling system is adopted, in which neighbors are labeled in a clockwise fashion. Thus, the neighbor directly above a pixel will be neighbor 0. Figure 3.18 shows this relationship between pixel x and its neighbors. The neighbor's class also ensures that pixels along the border of images are given the correct neighbors, and are not allowed to have neighbors which extend beyond the boundaries of the image.

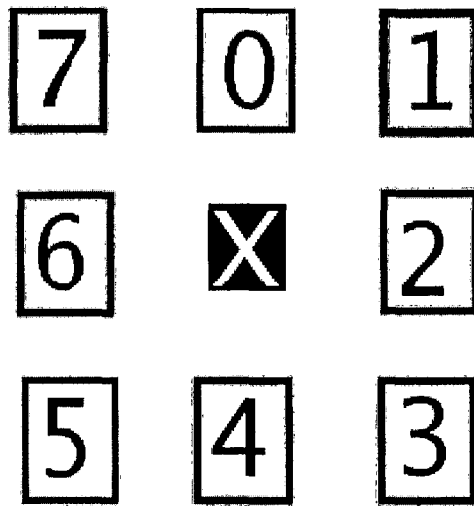


Figure 3.18: The neighborhood labeling system.

The histogram class is used to model region data in the grayscale and color histogram implementation. This class builds up histograms from an array of values and has methods to normalize the histogram and evaluate the probability of an intensity value lay in the histogram. The histogram has 256 levels, and each level stores the percentage where the intensity occurs in the image. For the grayscale implementation one histogram is used for each region model. For the color implementation three histograms are used for each region model.

Expectation Maximization data-clustering algorithm is suggested, and is only used in the Gaussian Mixture Model implementation [50]. This algorithm is used to iteratively determine a specified number of Gaussian clusters from an array of values. For the purpose of this implementation, 5 Gaussians are chosen for each Gaussian Mixture Model. The purpose of the EM clustering algorithm is to determine the means, covariance matrices and the weights of each Gaussian in the Gaussian Mixture Model.

The idea of Graph-cut is an energy minimization technique for image segmentation and solving it with graph-based algorithms. Graph-cut combines the two already known approaches for image segmentation: algorithms based on colors (or more precisely gray-levels) and segmentation based on the contrast in different regions of an image. Grab-cut extends this (Graph-cut) useful scheme to color images by using the iterative optimization structure. Instead of gray-level histograms, it makes use of Gaussian mixture models (GMM) and the Grab-cut algorithm uses only a three channel (RGB) based color image. Background and foreground are each described with five full-covariance Gaussian components, since Grab-cut is only applicable for color image. The Graph-cut method is modified with the iterative structure taken from the Grab-cut algorithm. Table 3.4 shows grayscale image segmentation times.

Figure 3.19 shows the example of the segmentation step by step. First, the user marks the background of the image, then the algorithm will give a segmented binary image as shown in Figure 3.19.b. To remove some false segmentation parts, a 5 pixel wide disk shape morphological dilation are applied to the segmented image to get a better segmentation result, and the resulting binary image is shown in Figure 3.19.c. Figure 3.19.d-e. show the segmented image. In Figure. 3.20, some examples of the segmented images are shown.

Some of the segmentation results improved by user touchup are given in Figure 3.21. The user marks the background and foreground with touchup if the segmentation results are unsatisfactory. This is shown in Figure 3.21.b. The segmentation process will start again automatically with user touchup. The final segmentation results are given in Figure 3.21.c.-d.

The last 10 images result in wrong or even no segmentation borders at all. This is mainly due to the poor signal-to-noise ratio of the infrared images. We presume even a human user might have difficulties in drawing a suitable segmentation border in images of the last type seen in Figure 3.22. Figure 3.22.a. shows unsatisfactory segmentation results which even after user touchup does not give satisfactory segmentation as seen in Figure 3.22.b.

Table 3.4 Grayscale image segmentation times.

Image Size	200x200	450x450	800x600
Segmentation time	12.34 sec	20.17 sec	27.21 sec

The Grab-cut algorithm gave 75% of image database segmentation results as satisfactory, 15% of the segmentation result as unsatisfactory but these results were improved with user touchup and 10% of the image gave no segmentation results at all.

The Graph-cut is a complicated algorithm. Implementing it can be quite time consuming. Instead, the Graph-cut C++ program is taken from the publicly available implementation which is made by Vladimir Kolmogorov. It is available at <http://www.cs.cornell.edu/people/vnk/software.html>. [51] This software is very easy to use and compiles without problems in Visual C++. This compiled file is converted as mex format to be able to use in MATLAB.

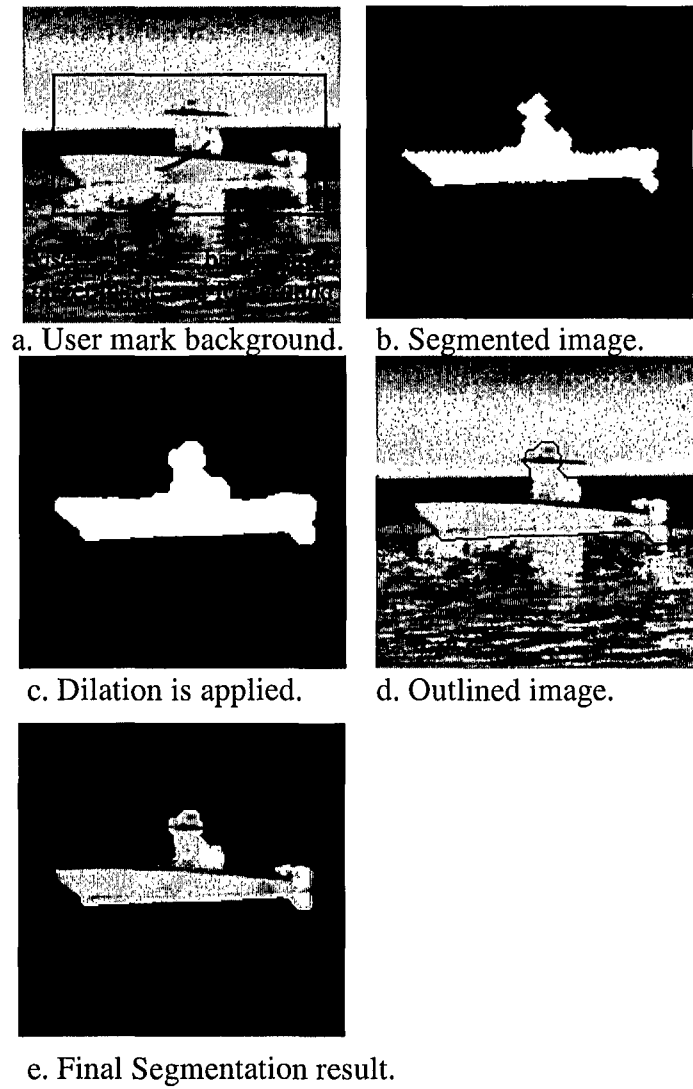


Figure 3.19. IR grayscale image segmentation.

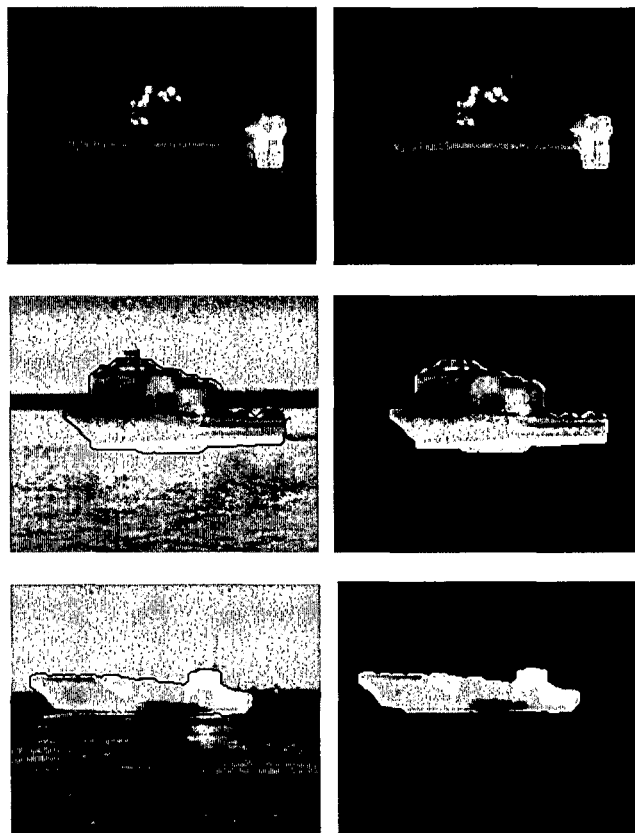


Figure 3.20. Some of the example segmented image.

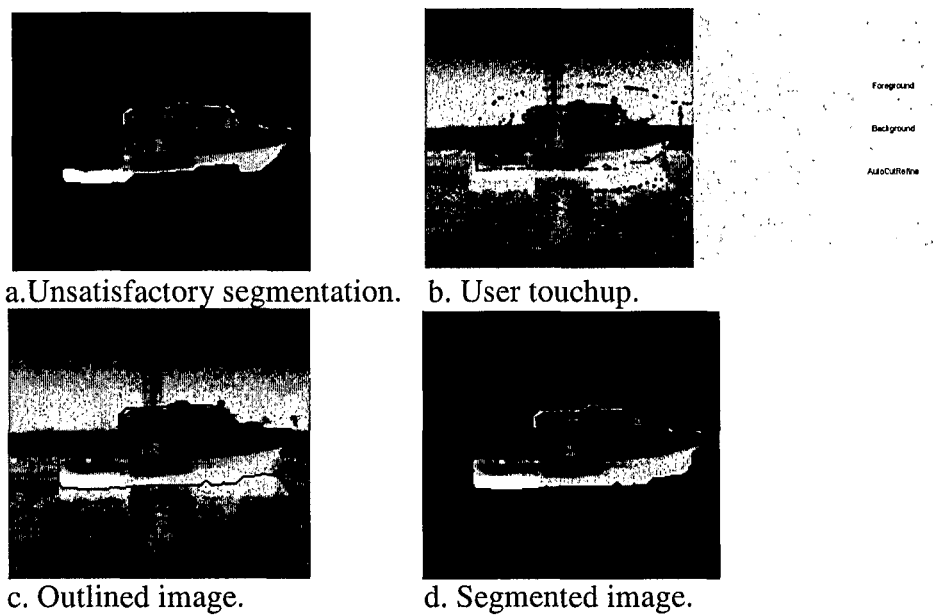
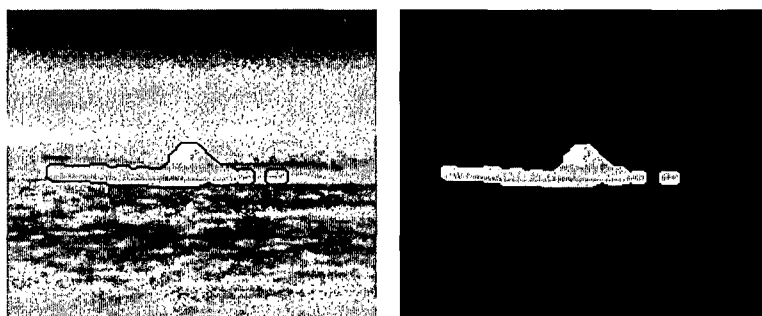
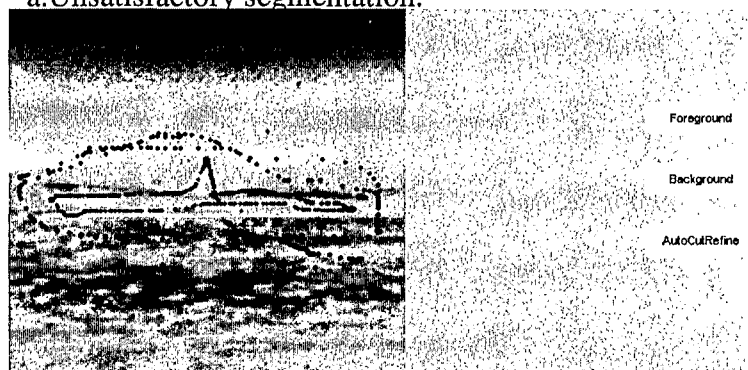


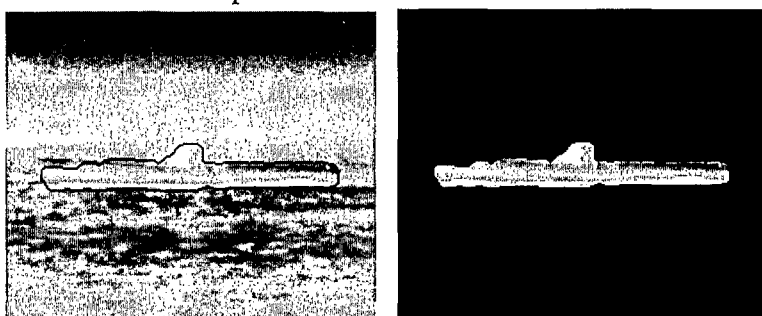
Figure 3.21. Unsatisfactory segmentation was improved with user touch-up.



a. Unsatisfactory segmentation.



b. User touchup.



c. Segmented image.

Figure 3.22. Wrong or even no segmentation result after touch-up.

CHAPTER 4

CLASSIFICATION RESULTS

The Principal Component Analysis (PCA) based statistical pattern recognition system is presented in this thesis for small boat classification from visible and IR images.

The percentage of the total variability explained by each principal component (eigenvalue) is calculated. The 10 biggest eigenvalues represent 95% of the PCA. Eigenvalue(percentage) graphic is shown in Figure 4.1. In order to get a better classification rate, we choose the eigen vectors corresponding to the 20 largest eigenvalues for the PCA testing part.

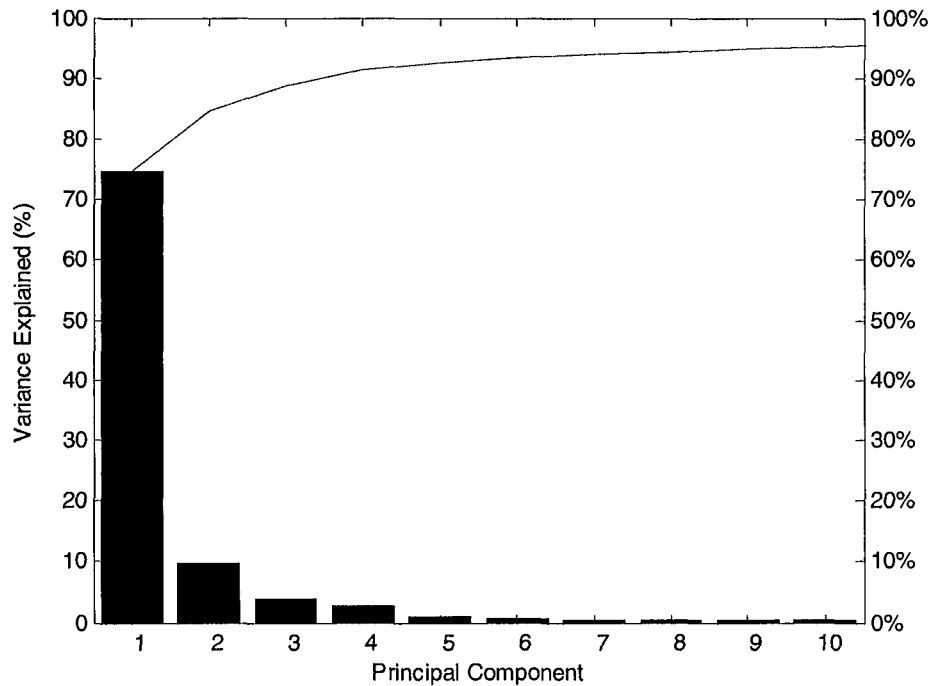


Figure 4.1. The magnitude of the eigenvalues represent covariance matrix.

First, the unsegmented visible target images were classified with PCA and a 68% successful classification rate was obtained for visible image. Although the image database has 10 different objects and 50 images (5 for each object), The PCA was trained with 9 different objects and 45 images (5 for each object). PCA was not trained for one other object and this object was marked down as an unknown-untrained object.

Every class is represented with 5 images and all of the 5 training images are classified as an independent class, and later, these five independent classes are joined together as a whole class. This independent classification is improved recognition rate.

Since MATLAB has a 10000x10000 maximum matrix computation capabilities, we have to resize it from 450x450 (pixel) visible images to 100x100 (10000x1 vector) image sizes. Experiments were conducted to choose optimal minimum Euclidian distance for untrained images. If the PCA is not trained with an image or an unknown target image, the program will tell user “This image is not in the classification database or an Unknown image”. Unsegmented visible image classification results are given in Table 4.1. When we inspect Table 4.1, some of the target images are lower classification rate. Since these images have a high clutter background (sea wave clutter, background, etc...), the image classification rate for these is worse than any other image rate. Highlighted cells show true classification.

Table 4.1. Unsegmented visible image classification result.

Pattern Recognition Technique				PCA, trained with 9 different objects, 45 images(5 for each object)						
Dimension Reduced To				450x450 visible target image resized to 100x100(10000x1 vector), reduced to 20 weight values for each object obtained from 20 biggest value eigen vectors corresponding to 20 largest eigenvalues.						
Classifier				Nearest Neighbor						
Classifier Training numbers				45						
	Result									
Test	Big Ship	Patrol Boat	Small Yacht	Zodiac	Attack Boat	Range Ship	Fisher Ship	Rhib	Manta	Untr.
Big Ship	5									
Patrol Boat		4			1					
Small Yacht			3			1			1	
Zodiac				5						
Attack Boat		1			3		1			
Range Ship			1			2			2	
Fisher Ship	1						4			
Rhib		1		1	1			2		
Manta			1			1			3	
Untrained							1		1	3
Correct Classification Rate: 68%										

The unsegmented IR target image was classified with PCA and we got a 62% successful classification rate for IR images. Although the image database has 12 different objects and 60 images (5 for each object), the PCA was trained with 11 different objects and 55 images (5 for each object). PCA was not trained for one other object and this object was marked down unknown-untrained object. Since MATLAB has a 10000x10000 maximum matrix computation capabilities, we have to resize from 200x200 (pixel) IR images to 100x100 (10000x1 vector) image sizes. The unsegmented IR image

Table 4.4. Segmented IR image classification result.

Pattern Recognition Technique						PCA, trained with 11 different objects, 55 images(5 for each object)						
Dimension Reduced To						200x200 IR target image resized to 100x100(10000x1 vector), reduced to 20 weight values for each object obtained from 20 biggest value eigen vectors corresponding to 20 largest eigenvalues.						
Classifier						Nearest Neighbor						
Classifier Training numbers						55						
	Result											
Test	Patrol	Yacht	Gat.	Zod.	Foun.	Hm	Lcm	Mant.	Mon.	Rcb	Rhib	Untr.
Patrol	5											
Yacht		4				1						
Gat.			5							1		
Zod.				5								
Foun.					3				1		1	
Hm						5						
Lcm			1				4					
Mant.								3		1		1
Mon.					1				3		1	
Rcb	1	1								3		
Rhib			1		1				1		2	
Untr.					1					1		3
Correct Classification Rate: 77%												

A very user friendly classification program was developed. Even if the user does not know MATLAB or PCA, he or she can easily add new training images and target images in the image target database. The classification program user interface is given Figure 4.1. Classification program has 7 different menus; this menu is listed below:

1. Select Image

2. New Record
3. Number of ID(s)
4. Target Recognition
5. Delete Database
6. ID Information
7. Mean of Target and Eigenimage

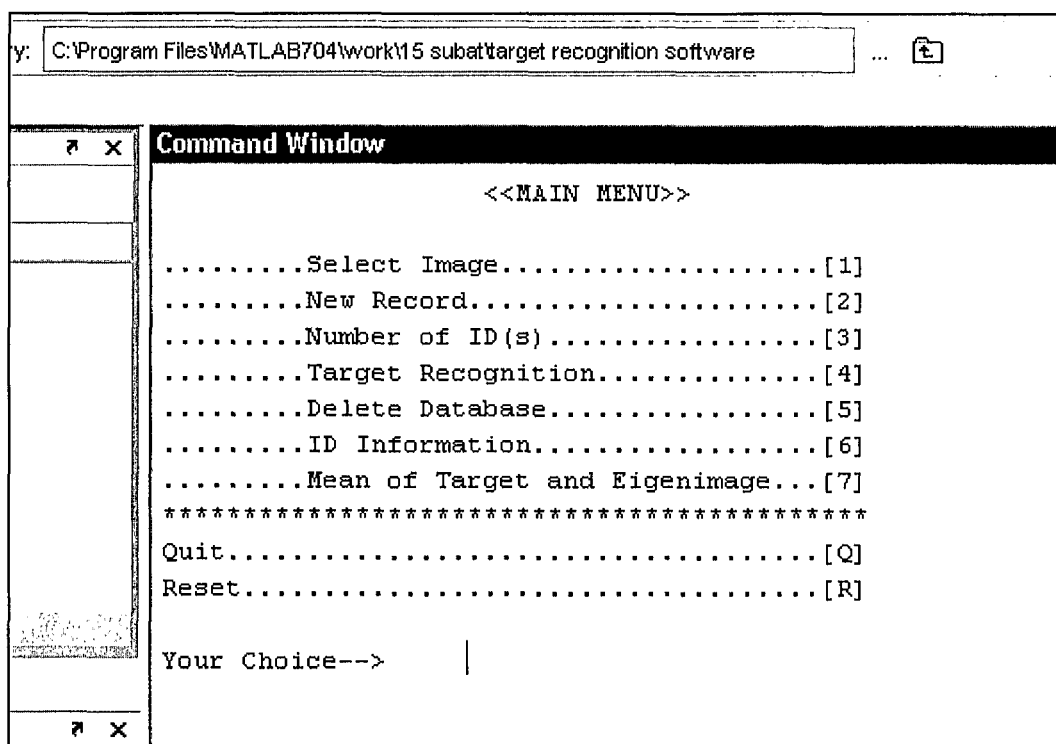


Figure 4.2. Classification program user interface.

CHAPTER 5

CONCLUSION AND FUTURE WORK

New techniques for the segmentation of small boat region from visible and IR images and the classification of features extracted from segmented regions for small boat identification have been presented in this thesis. The visible small boat images were segmented with the adaptive progressive thresholding algorithm.

On the other hand, the conventional segmentation algorithm did not work on highly cluttered IR ship images, since IR images in general are not that sharp while displaying objects compared to visible images. Basic thresholding, entropy-based, pure edge-based and APT image segmentation techniques were tested for IR small boat images. None of these methods gave satisfactory segmentation results. A set of 100 infrared images showing different levels of quality were used. Two new segmentation approaches for IR ship image segmentation were also employed.

The first method named the SUSAN-APT algorithm gave 65% of the absolute segmentation results and 20% of the results with inaccurate region can be improved with image pre-processing operation and 15% of the images gave no segmentation result at all.

The second method named the Grab-cut algorithm gave 75% of absolute segmentation results and 15% of segmentation results were unsatisfactory and 10% of images gave no segmentation result at all. However, unsatisfactory segmentation results were improved with user interaction at the clue marking stage. In total 90% of the images were segmented with Grab-cut.

SUSAN-APT algorithm does not need user interaction for segmentation. Because of the non-user interaction, SUSAN-APT algorithm looks attractive. On the other hand,

Graph-cut based Grab-cut algorithm sometimes needs user intervention for unsatisfactory segmentation, but Grab-cut algorithm got a better segmentation ratio and quality for IR images.

Both algorithms have unsatisfactory segmentation results and this can be improved by wavelet based image pre-processing procedures like enhancement techniques. The last unsegmented (failed-segmentation) images result in wrong or even no segmentation borders at all. This is mainly due to the bad signal-to-noise ratio of the infrared images. It is presumed that even a human user might have difficulties in drawing a suitable segmentation border for this type of images.

Because of the morphological operations, some of the objects small superstructures parts (antennas, masts,) are lost in segmentation, but these will not affect our classification result a lot since we have to resize our image to 100x100 for PCA based dimensionality reduction.

The Principal Component Analysis (PCA) based statistical pattern recognition system is presented in this thesis for small boat classification from visible and IR images. First, unsegmented target images are classified with PCA, and we obtained 62% successful classification rate for an IR image and 68% rate for a visible image. On inspecting the IR and visible image classification rates, some of the target images had a lower classification rate. Since these images have highly cluttered backgrounds (sea wave clutter, background, etc.), their classification rate is worse than that of the images with low clutter.

After the segmentation, classification rate has been improved and obtained 77 % successful classification rate for IR images, and 84% for visible images. If the PCA is not trained with an image or an unknown target image, the program will tell user “This image

is not in the classification database or an Unknown image”. Several experiments have been conducted to obtain the most appropriate Euclidian distance to distinguish trained and untrained images.

5.1. Future Work and Recommendation

If the training image database size increases to 24 (one image for every 15 degree viewing angle) for every class, the recognition rate may reach 100%. A neural network based classification technique will give better classification rates.

For IR ship images, pre-processing steps would be useful to improve the image quality and it would provide better object segmentation. The wavelet based noise reduction algorithm might be a very suitable step for IR small boat images as a pre-processing stage. An integration of these image enhancement algorithms has to be investigated as future work.

Matting is a technique which smoothes the boundaries of segmented objects, and makes their appearance more natural. The boundaries of objects in images are sometimes well defined and can easily be traced with some sort of edge detection mechanism. However, natural or organic objects often have very intricate boundaries. Hair is very hard to segment and trace with standard edge detection methods. Water spray, smoke and leaves are also very difficult to segment using any of the standard techniques. Grab-cut algorithm has its own matting technique for color images. On the other hand, a border matting method is not proposed in the literature for grayscale images. IR grayscale images need border matting for better segmentation.

Image frames from video stream can be used for fast and basic segmentation for this kind of application. A sequence of images contains more information than a single image. For this reason, image sequence analysis has been used in computer vision for

quite some time. In particular, a sequence of images is useful for object detection when the camera moves relative to the object. Due to the relative motion, objects at different distances from the camera have different image motions. Using this property, one can obtain information on the structure of the 3-D scene as well as the relative motion between the camera and the scene. Furthermore, the individual images are corrupted by camera noise. Use of a sequence of images enables suppression of this noise for reliable detection of low contrast objects.

Effective methodologies like image fusion and Dual-band IR imaging could be added to this work. Dual-band imaging has the advantage that it can better detect targets in clutter. It can distinguish between targets and decoys and has the ability to defeat IR countermeasures such as smoke, camouflage, and flares.

Sensor integration and registration is a prerequisite for exploiting the inherent advantages of multi-sensor systems over single sensor systems. Using a single sensor, we can monitor objects with a precision and accuracy that depend on the sensor characteristics. By using multiple sensors to observe a target, we can obtain multiple viewpoints, extend coverage both spatially and temporally, reduce the ambiguity and obtain a more precise estimate of feature inconsistency.

Multiple sensors have different electromagnetic spectral responses to capture distinguished features in different spectral bands. In the design of a multiple sensor imaging system, one selects wavelengths in an electromagnetic spectral band as wide as possible. Multiple sensors act in a synergistic manner, providing complementary information: the IR camera provides data to locate the objects of warmer temperatures within the scene, while the visible camera provides background in the scene. The fusion of two images allows locating the warm objects with respect to the background.

REFERENCES

- [1] www.defensenews.com, 19-26 December 2006 weekly news issue.
- [2] B. Bhanu, T. Jones, "Image Understanding Research for Automatic Target Recognition", IEEE Transactions on Aerospace and Electronic Systems, Vol. 8, No. 10, pp. 15-23, 1993.
- [3] W. M. Brown, C. W. Swonger, "A Prospectus for Automatic Target Recognition", IEEE Transactions on Aerospace and Electronic Systems, Vol. 25, No. 3, pp. 58-69, 1989.
- [4] M. Roth, "Survey of Neural Network Technology for Automatic Target Recognition", IEEE Transactions on Neural Networks, Vol. 1, No. 1, pp. 98-110, 1990.
- [5] B. Bhanu, "Automatic Target Recognition: State of the Art Survey", IEEE Transactions on Aerospace and Electronic Systems, Vol. 22, No. 4, pp. 364-379, 1986.
- [6] D. Nair, J. Aggarwal, "Bayesian Recognition of Targets by Parts in Second Generation Forward Looking Infrared Images.", Elsevier Image and Vision Computing, Vol. 18, No. 3, pp 849-864, 2000.
- [7] B. Li, R. Chellappa, Q. Zheng, "Experimental Evaluation of FLIR ATR Approaches, a Comparative Study", Elsevier Computer Vision and Image Understanding, Vol. 84, No. 1, pp 5-24, 2001.
- [8] J. R. Schalkoff, "Pattern Recognition: Statistical, Structural and Neural Approaches", John Wiley & Sons, 1992.
- [9] N. Sebe, "Machine Learning in Computer Vision", Springer, 2006.
- [10] B. Krose, "Introduction to Dimension Reduction and Feature Extraction", MIT Press Neural Computation, Vol.14, No.1, pp. 191-215, 2002.
- [11] K. Imola, "A survey of dimension reduction techniques", Technical Report UCRL-ID-148494, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 2002.
- [12] W. M. Brown, C. W. Swonger, "A Prospectus for Automatic Target Recognition",

- IEEE Transactions on Aerospace and Electronic Systems, Vol. 25, No. 3, pp. 143-154, 1989.
- [13] M. Turk, A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, Vol. 3 No. 1, pp. 71–86, 1991.
 - [14] M. Kirby, L. Sirovich, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces", IEEE Transactions on Pattern analysis and Machine Intelligence Vol. 12, No. 1, pp. 103–108, 1990.
 - [15] E.Oja, H.Ogawa, and J.Wangviwattana, "Principal Component Analysis by Homogeneous Neural Networks, Part I: Weighted Subspace Criterion", IEEE Transactions on Information and Systems, E75-D, Vol. 14, No. 3, pp. 366-375, 1992.
 - [16] R. Gonzalez, R. Woods, "Digital Image Processing", 2nd Ed., Prentice Hall, 2002.
 - [17] Y. Zhang, "Advances in Image and Video Segmentation", IRM press, 2006.
 - [18] H. Ballard, C. Brown., "Computer Vision", Prentice-Hall Inc., New Jersey, 1982.
 - [19] Y. Zhang (1995). "Influence of Segmentation over Feature Measurement", Elsevier Pattern Recognition Letters, Vol. 16, No. 2, pp. 201-206, 1995.
 - [20] L. Roberts, "Machine Perception of Three-dimensional Solids", MIT Press Optical and Electro-optical Information Processing, pp. 159-197, 1965.
 - [21] R. Haralick, L. Shapiro, "Image Segmentation Techniques", Computer Vision Graphics and Image Processing, Vol. 29, pp. 100-132, 1985.
 - [22] N. Otsu, "Thresholding Selection Method from Grey Level Histogram", IEEE Transaction on Systems Man and Cybernetics, Vol.9, No. 1, pp.62-66, 1978.
 - [23] P. Felzenszwalb, F. Huttenlocher, "Efficient Graph-based Image Segmentation", International Journal of Computer Vision, Vol. 59, No. 2, pp. 167–181, 2004.
 - [24] I. Haritaoglu, D. Harwood, "W4: Real-time Surveillance of People and Their Activities", IEEE Transaction Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 809–830, 2000.
 - [25] H. Tian, T. Srikanthan and K. V. Asari, "An Automatic Segmentation Algorithm for the Extraction of Lumen Region and Boundary from Endoscopic Images," IEEE Journal of Medical & Biological Engineering & Computing, Vol. 39, No. 1,

pp. 8-14, 2001.

- [26] K. V. Asari, "A Fast and Accurate Segmentation Technique for the Extraction of Gastrointestinal Lumen from Endoscopic Images," *Journal of Medical Engineering & Physics*, Vol. 22, No. 2, pp. 89-96, 2000.
- [27] W. Kim, Y. Kim., "A New Region-based Shape Descriptor", *Elsevier Signal Processing*, Vol. 16, No. 1, pp. 95-102, 2000.
- [28] M.A. Bhuiyan, V. Ampornaramveth, S. Muto, H. Uena, "Face Detection and Facial Feature Localization for Human-machine Interface", *Japan National Institute of informatics Journal*, Vol. 5, No.3, pp. 76-85, 2003.
- [29] N.R. Pal, S.K. Pal, "A Review on Image Segmentation Techniques", *Elsevier Pattern Recognition*, Vol. 26, No.3, pp.1277-1294, 1993.
- [30] S. Kawato, N. Tetsutani, "Detection and Tracking of Eyes for Gaze-camera Control", *Image Vision Computing*, Vol. 22, No. 12, pp. 1031-1038, 2004.
- [31] J. Edwards,"A User's Guide To Principal Components", John Wiley & Sons, 1991.
- [32] P. Withagen, K. Schutte, A. Vossepoel, and M. Breuers, "Automatic Classification of Ships from Infrared (FLIR) Images", *Proceedings of SPIE, Conference on Signal Processing, Sensor Fusion, and Target Recognition VIII*, Vol. 3720, pp 180-187, Orlando, FL, 1999.
- [33] *Image Processing Toolbox User's Guide*, Mathworks Inc., URL: www.mathworks.com/ip
- [34] B. Kamgar, "Object Extraction in IR Images", US patent #5923776, 1999.
- [35] J. Lee and M. Yang, "Threshold Selection Using Estimates from Truncated Normal Distribution", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, No. 3, pp. 422-429, 1989.
- [36] B. Li, Z. Shen, T. Lan, "A Real-time System for Target Detection and Tracking in IR Image Sequence", *Proc. IEEE National Aerospace and Electronics Conf.*, pp 156-170, New York, NY, 1996.
- [37] S. Smith, "SUSAN- a New Approach to Low Level Image Processing", *Technical Report TR95SMS1*, Defense Research Ag. ,UK, <http://www.fmrib.ox.ac.uk/steve/>, 1997.

- [38] J. Canny., "A computational Approach to Edge Detection", IEEE Transaction Pattern Analysis and Machine Intelligence, Vol. 8, No. 4, pp 679-698, 1986.
- [39] C. E. Shannon, "A Mathematical Theory of Communication", The Bell System Technical Journal, Vol. 7, pp. 379-423, 1948.
- [40] N. Pal, S. Pal, "Entropic Thresholding", Signal Processing, Vol.16, No 7, pp. 97-108, 1989.
- [41] C. Li, P. Tam, "An iterative algorithm for minimum cross entropy thresholding", www.microserf.org.uk/academic/archives/MIUA2003_CJR_.pdf, 2003.
- [42] Y. Boykov, V. Kolmogorov, "An Experimental Comparison of Min-cut/max-flow Algorithms for Energy Minimization in Vision", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 9, pp. 1124-1137, 2004.
- [43] V. Kolmogorov, R. Zabih, "What Energy Functions Can Be Minimized Via Graph Cuts?", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 2, pp. 207-230, 2004.
- [44] Y.Boykov, M. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images", In Proceedings of the International Conference on Computer Vision, Vol. 1, No. 3, pp. 105-112, 2001.
- [45] A. Blake, C. Rother, M. Brown, P. Perez, P. Torr, "Interactive Image Segmentation Using an Adaptive GMMRF Model", In Proceedings of the International Conference on Computer Vision, 8th European Conference on Computer Vision, Prague, Czech Republic, Vol. 3, No. 2, pp. 428-441, 2004.
- [46] Y. Li, J. Sun, C. Tang, H. Shum, "Lazy snapping", ACM Transactions on Graphics, Vol. 23, No. 3, pp. 303-308, 2004.
- [47] T.Xiaoqian, "Implementing GrabCut", In Proceeding IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Vol. 2, No 4, pp. 122-129, 2006.
- [48] U. Zolzer, "Gray Cut Object Segmentation in IR-Images", 2nd International Symposium on Visual Computing (ISVC), Lake Tahoe, Nevada, USA, November 6-8 2006.
- [49] O. Rusch, C. Ruwwe, U. Zoelzer, "Image Segmentation in Naval Ship Images", 11. Workshop Farbbild verarbeitung, pp. 63-70, 2005.

- [50] S. Kung, M. Mak, S. Lin., "Biometric Authentication: A Machine Learning Approach", Prentice Hall, 2004.
- [51] V. Kolmogorov, <http://www.cs.cornell.edu/people/vnk/software.html>
- [52] H. D. Cheng, Y. H. Chen, X. H. Jiang, "Thresholding Using Two Dimensional Histogram and Fuzzy-entropy Principle", IEEE Transaction Image Processesing, Vol. 9, No 8, pp. 732-735, 2000.
- [53] W. Jim , J. Li, "Infrared Image Segmentation Based on 2-D Maximum Fuzzy Entropy and Intelligent Genetic Algorithm", Proceeding of the 2004 international Conference on Intelligent Mechatronics and Automation Chengdu, China, Vol. 3, No. 8, pp. 166-180, 2004.

VITA

NAME :Cenk Yaman

DATE OF BIRTH

DEGREES :Bachelor of Science (Electronics and Communication Engineering), YILDIZ TECHNIC UNIVERSITY
Istanbul, Turkey, August 1995.
Master of Science (Electrical Engineering), Old Dominion University, Norfolk, Virginia, May 2007.

EMPLOYEMENT :Officer (Cap.) in Turkish Air Force, August 1996 to present.

CONFERENCE PAPER : 1. Cenk Yaman and K. Vijayan Asari, "Long-range Target Classification in a Cluttered Environment Using Multi-sensor Image Sequences", 3rd International Conference on Recent Advances in Space Technologies - RAST 2007, Istanbul, Turkey, June 14 - 16, 2007 (accepted for presentation).