A FULLY AUTOMATIC SHAPE BASED GEO-SPATIAL OBJECT RECOGNITION

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

BY

MUSTAFA ERGÜL

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

SEPTEMBER 2012

Approval of the Thesis

A FULLY AUTOMATIC SHAPE BASED GEO-SPATIAL OBJECT RECOGNITION

Submitted by **MUSTAFA ERGÜL** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering, Middle East Technical University,** by,

Prof. Dr. Canan Özgen

Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. İsmet Erkmen

Head of Department, **Electrical and Electronics Engineering** _____________

Prof. Dr. A. Aydın Alatan

Supervisor, **Electrical and Electronics Eng Dept., METU** ______________

Examining Committee Members

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

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

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Engineering Dept., METU ______________

Dr. Kubilay Pakin
ASELSAN ______________

Dr. Emre Başeski
HAVELSAN ______________

Date: 04.09.2012

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

Name, Lastname : Mustafa Ergül

Signature :

ABSTRACT

A FULLY AUTOMATIC SHAPE BASED GEO-SPATIAL OBJECT RECOGNITION

Ergül, Mustafa
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. A. Aydın Alatan

September 2012, 152 pages

A great number of methods based on local features or global appearances have been proposed in the literature for geospatial object detection and recognition from satellite images. However, since these approaches do not have enough discriminative capabilities between object and non-object classes, they produce results with innumerable false positives during their detection process. Moreover, due to the sliding window mechanisms, these algorithms cannot yield exact location information for the detected objects. Therefore, a geospatial object recognition algorithm based on the object shape mask is proposed to minimize the aforementioned imperfections. In order to develop such a robust recognition system, foreground extraction performance of some of popular fully and semi-automatic image segmentation algorithms, such as normalized cut, k-means clustering, mean-shift for fully automatic, and interactive Graph-cut, GrowCut, GrabCut for semi-automatic, are evaluated in terms of their subjective and objective qualities. After this evaluation, the retrieval performance of some shape description techniques, such as ART, Hu moments and Fourier descriptors, are investigated quantitatively. In the proposed system, first of all, some hypothesis points are generated for a given test image. Then, the foreground extraction operation is achieved via GrabCut algorithm after utilizing these hypothesis points as if these are user inputs. Next, the extracted binary object masks are described by means of the integrated versions of shape description techniques. Afterwards, SVM classifier is used to identify the target objects. Finally, elimination of the multiple detections coming from the generation of hypothesis points is performed by some simple post-processing on the resultant masks. Experimental results reveal that the proposed algorithm has promising results in terms of accuracy in recognizing many geospatial objects, such as airplane and ship, from high resolution satellite imagery.

Keywords: object recognition, image segmentation, foreground extraction, shape description

ÖZ

TAM OTOMATİK ŞEKİL TABANLI YER UZAMSAL NESNE TANIMA

Ergül, Mustafa
Yüksek Lisans, Elektrik-Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. A. Aydın Alatan

Eylül 2012, 152 sayfa

Son yıllarda, uydu görüntülerinden yer uzamsal nesne tespiti ve teşhisi hakkında literatürde yerel öznitelikler veya bütünsel görünüş temelli çok sayıda metot önerilmiştir. Ama bu metotlar nesne sınıfı ile arkaplan sınıfı arasında yeterince ayırt edici kabiliyete sahip olmadığı için tespit işlemi sırasında çok sayıda yanlış alarmlı sonuç üretirler. Ayrıca tespit işleminde kullanılan kayan pencere mekanizmasından dolayı genellikle bu algoritmalar bulunan nesnelerin konum bilgisini tam olarak veremezler. Bundan dolayı yukarıda bahsedilen eksiklikleri gidermek için nesnenin şekline dayalı yer uzamsal nesne tanıma algoritması önerilmiştir. Böyle bir gürbüz nesne tanıma sistemini geliştirmek için öncelikle bazı popüler tam ve yarı otomatik görüntü bülütleme algoritmalarının, örneğin tam otomatik için düzgülü kesik, k-orta kümeleme, ortalama kayma ve yarı otomatik için interaktif grafik kesik, GrabCut, GrowCut, önplan nesne çıkarımı performansları nesnel ve öznel olarak değerlendirilmiştir. Ondan sonra ARD, Hu momentler ve Fourier tanımlayıcıları gibi bazı şekil tanımlayıcı tekniklerinin erişim performansları incelenmiştir. Önerilen sistemde ilk önce verilen test görüntüsündeki hipotez noktaları üretilmektedir. Daha sonra, bu hipotez noktaları kullanıcı girdisi gibi kullanıldıktan sonra interaktif GrabCut algoritması vasıtasıyla önplan nesnesini elde etme işlemi gerçekleştirilir. Ardından, çıkarılan ikili nesne maskeleri şekil tanımlayıcı tekniklerinin entegre edilmiş sürümü aracılığıyla tanımlanır. Daha sonra, hedef nesneleri tespit etmek için DVM sınıflandırıcısı kullanılır. Son olarak, hipotez noktalarının üretimi sırasında meydana gelen birden fazla tespitlerin elenmesi sonuç maskeleri üzerinde birkaç basit ileri işlemeyle gerçekleştirilir. Deney sonuçları önerilen algoritmanın yüksek çözünürlüklü uydu görüntülerinden uçak ve gemi gibi çok sayıda sınıftan yer uzamsal nesnelerin tanıma performansı açısından umut verici sonuçlara sahip olduğunu göstermiştir.

Anahtar Kelimeler: nesne tanıma, görüntü bölütleme, önplan çıkarımı, şekil tanımlayıcı

To My Family,

ACKNOWLEDGEMENTS

I would like to express my gratitude and deep appreciation to my supervisor Prof. Dr. A. Aydın Alatan for his guidance, positive suggestions and also for the excellent research environment he had provided.

I would like to thank to all the people in Multimedia Research Group for their assistance and delightful friendship. We have shared, more than an office, two years with bittersweet memories.

I would like to also acknowledge The Scientific and Technological Research Council of Turkey (TUBITAK) for their funds.

Finally, I would like to thank and express my sincere love to my family. Their endless love is one of the few things that make me truly happy.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

CHAPTER 1

INTRODUCTION

Computer vision is a discipline with wide application areas while it attracts many researchers due to its variety of innovative consequences. However, despite a considerable amount of efforts over the last few decades, many problems are still unsolved and require further investigation. The robot navigation systems, human motion modeling and its adaptation to animation, traffic surveillance, 3D data extraction, environment modeling, organizing information, target detection, object segmentation and recognition are still some of the active research areas under computer vision discipline. Therefore, this thesis is devoted to geospatial target detection techniques.

By virtue of the recent advances in satellite technology, acquisition and storage of satellite images of any region in the world is becoming easier and easier. This advance in the technology causes the creation of the huge amount of data in electronic formats. Unfortunately, this much amount of data cannot be interpreted easily by humans; hence, there is a requirement for automated systems that could replace a human operator for image understanding. For this purpose, a remarkable amount of object detection and recognition algorithms from satellite or aerial imagery has been proposed in the literature. The local feature based object detection, such as Bag-of-Visual Words methods (BOVW) [1], and the global appearance based object detection, such as matched filters based methods [2], are some examples of these algorithms. Nevertheless, they have some imperfections affecting the system performance. The first drawback is that these algorithms usually give outputs with high false alarm rates, since the extracted features cannot sufficiently represent the target objects in the image, and so this deficiency causes the discrimination problem between the target and non-target regions. The second disadvantage is that these algorithms cannot give the exact location of the target objects due to the sliding window mechanism used for detection process. On the other hand, unlike the local feature based and global appearance based methods, the shape based object recognition methods might yield results with high precision rates and precisely known position, although it is extremely difficult to extract the object shape from the image. This result is due to the fact that the shape of an object includes more precise and complete knowledge about the object identity and function.

In this thesis, the main goal is to be able to perform automatic detection, recognition and localization of certain kinds of objects, such as aircrafts and ships, in a satellite image by using a shape based method.

1.1 Literature Review for Object Detection and Recognition

Object detection and recognition is a crucial, yet a challenging vision task. It is a critical part in many applications, such as image search, image auto-annotation and scene understanding; however, it is still an open problem due to the complexity of object classes and images. Therefore, there exists an extensive literature for the object detection and recognition. Among various methods proposed for different situations over the years, those of the state-of-the-art approaches that could be utilized for object detection and recognition from satellite and aerial images could be categorized as follows:

- Local Feature Based Approaches
- Global Appearance Based Approaches
- Shape Based approaches

In the following three parts, a brief survey is presented for the efforts that construct the basis for each one of the aforementioned three approaches as well as the studies that exploit those methods for geospatial object detection and recognition problem.

1.1.1 Local Feature Based Approaches

The central idea of local feature-based object detection and recognition algorithms lies in finding interest points, often occurred at intensity discontinuity, and then describing regions around those interest points. Local features are extracted from highly repetitive salient regions that are invariant to changes due to scale, illumination and affine transformation. These regions are characterized by corners, edges, blobs, ridges etc. Some of the well-known and successful interest point detection techniques in the sense of runtime, repeatability, invariance to scale, illumination and affine changes can be recorded as: Harris and Hessian based feature detectors, Difference of Gaussian (DoG) blob detector, Maximally Stable Extremal Regions (MSER), Entropy Based Salient Region detector (EBSR), Intensity Based Regions and Edge Based Regions (IBR, EBR) [3]. Regions around the interest points already identified by one of the detector algorithms are described in terms of certain invariance properties by feature descriptors. Invariance means that the description should be robust against various image variations, such as affine distortions, scale changes, illumination changes or compression artifacts (e.g. JPEG). The most popular descriptors proposed up to now can be listed as follows: Scale Invariant Feature Transform (SIFT), PCA-SIFT, gradient location-orientation histograms (GLOH), spin images, shape context, Locally Binary Patterns (LBP), complex and steerable filters [3]. A performance evaluation among various local descriptors can be examined in [3] [4], and an extensive study on region detectors is presented in [3] [5]. After extracting and describing local features, the object type is learned by using discriminative classifier, such as Support Vector Machines (SVM), Linear Discrepant Analysis (LDA) or Boosting.

There are specific techniques that aim geospatial object detection from satellite or aerial imagery. As a prominent example, Sun et al. [6] have proposed a novel method to solve the problem of detecting geospatial objects in high resolution remote sensing images automatically. Each image is represented as a segmentation tree by performing a multi- scale segmentation algorithm at first, and all of the tree nodes are described as coherent groups instead of binary classified values. Afterwards, the trees are matched to choose the maximally matched subtrees, denoted as common subcategories, and these are organized to learn the embedded taxonomic semantics of object classes. This approach allows categories to be defined recursively, and express both explicit and implicit spatial arrangement of categories. Hence, this procedure provides a meaningful explanation for image understanding by realizing detection, recognition and segmentation of the geospatial objects in an image simultaneously.

In another local feature based approach, Tao et al. [7] present a new technique in order to detect airports in large high-spatial-resolution IKONOS satellite images. For this purpose, an airport is described by a set of SIFT keypoints and detected by using an improved SIFT matching strategy. After extracting matched SIFT points, a novel region–location method is suggested to discard the redundant matched keypoints and locate the possible regions of candidates that contain the target. This method utilizes the clustering knowledge from matched SIFT keypoints and the region information extracted through the image segmentation. Finally, airport detection is achieved by employing the prior information to the candidate regions.

Similar to the previous method proposed by Tao et al. [7], Sahli et al. [8] offer a feature based vehicle detection algorithm from high resolution aerial imagery. SIFT algorithm is employed to obtain keypoints in the image and the local structure in the neighboring of those keypoints are described by 128 gradient orientation based features. Then, a support vector machine is utilized to generate a model which can predict whether the keypoints belong to car structure in the image or not. Lastly, the collection of SIFT keypoints with car label are clustered in the geometric space into subsets such that each subset is associated to one car.

In another study, Rainey et al. [9] improve ship recognition and classification algorithm in electro-optical satellite imagery. The authors propose a modification for the sparse representation based classification (SRC) algorithms typically used in face detection. This modification focuses on the appropriate feature selection and description. To this end, feature vectors are obtained from SIFT keypoints by using SIFT descriptors coupled with the visual Bag-of-Words method. This approach improves the performance of the SRC algorithm on data set with nonuniform size, small amounts of training data, and large in-class variations significantly.

As a general notice, the local feature based methods are not robust to the challenges encountered in a typical satellite image, such as shadows and background clutters, since they dramatically change local appearances, and thus the corresponding local features. Furthermore, all of these methods experience the difficulty of lack of keypoints repeatability caused by intra-class variation of object categorizes.

1.1.2 Global Appearance Based Approaches

Global appearance based methods attempt to exploit the visual outlook of an object as a whole in order to identify this entity. They formulate the object detection and recognition as a classification problem, and hence, they could also be named as classifier-based methods. In these approaches, the image is partitioned into a set of overlapping windows and a decision is taken at each window about whether it contains a target object or not. The algorithms carry out the detection\recognition process as a binary classification which aims at learning an object class and discriminating it from the background. The classification function is learned through a set of labeled positive and negative examples together with some features extracted from them. Haar wavelets, Haar filters (rectangle features), Histogram Distance on Haar Regions (HDRD), edges together with chamfer distance, edge fragments and Histogram of Oriented Gradients (HOG) are some examples of features used in the algorithms. A discriminative classifier is used to exploit the class specific information and some of them are k-nearest neighbor, neural network, dynamic link architecture, Fisher linear discriminant, SVM and boosting algorithms.

Template matching of the different appearances of the detected object can be given as a simple example for this type of algorithms. The proposal of more complex and elaborate algorithms has started with a face detection\recognition problem. Most notably, the eigenface method suggested by Turk and Pentland [10] is one of the first face recognition systems that are computationally efficient and relatively accurate. The underlying idea of this approach is to compute eigenvectors from a set of vectors where each one represents one face image as a raster scan vector of grayscale pixel values. Each eigenvector, called as an eigenface, captures certain variation among all vectors, and a small set of eigenvectors captures almost all the appearance variation of facial images in the training set. Given a test image represented as a vector of grayscale pixel values, its identity is specified by determining the nearest neighbor of this vector after being projected onto a subspace spanned by a set of eigenvectors. The eigenface approach has been also adopted in recognizing generic objects. Afterwards, state-of-art global appearance based algorithms are proposed by Viola-Jones [11], Papageorgiu-Poggio [12], and Heisele et al. [13] in order to detect and recognize the face or human classes in the image.

Perrotton et al. [14] offer a specific solution to geospatial object detection and localization problem, such as aircrafts in clutter backgrounds, from aerial or satellite imagery by exploiting the appearance characteristics of objects. A boosting algorithm is employed to select discriminating features. A new descriptor, Histogram Distance on Haar Regions (HDHR), which is robust to the background and target texture variations, is introduced. This descriptor is based on the assumption that targets and background have different textures. Unfortunately, collecting the large amount of training data required for the AdaBoost learning algorithm is quite difficult to achieve considering the unavailability of such large annotated database. Therefore, image synthesis is utilized to generate a large amount of learning data such that the AdaBoost has access to sufficient representative data to take into account the variability of real operational scenes.

On the other hand, Cai et al. [15] present a technique to detect airplanes in panchromatic remote-sensing images. A circle-frequency filter (CF-filter) is introduced to locate airplane centers from the background. The filter first extracts candidate points of airplane centers and then airplane centers can be located through a basic clustering method. Since the CF-filter takes only the nine specific intensity changes around a circle fitted on an airplane center into account, this method cannot exactly represent the global outlook of the object. Thus, the system would naturally yield higher false alarm rates on regions with man-made objects, and structures with cross shape arrangements etc.

Global appearance based approaches have several drawbacks preventing to achieve high detection performances. The first of these drawbacks is that they are sensitive to illumination changes and occlusions. Secondly, the contrast between the object and the background is a critical factor in learning and detection procedure; for example, a bright-colored plane and a dark-colored plane cannot be learned and detected as the same object and needs utilization of different features. The next one is that these types of methods also tend to memorize the object categorize and might not provide a good generalization in detection and recognition process. Finally, the most notable limitation is that the detection is not invariant to rotation of the object, and hence, one should train a new classifier for each of the different pose or viewpoint of an object class. The other solution is that the detection operation should be performed by rotating the test image.

1.1.3 Shape Based Approaches

Early attempts on object detection and recognition were focused on using geometric models or shape of objects to achieve invariance across their appearance variation due to viewpoint and illumination change. The main idea is that the target structure has characteristic and distinguishable information and so this knowledge facilitates the recognition process using the edge or boundary of the object which is invariant to certain illumination change. However, an initial segmentation is required in order to acquire shape information of the object. After extracting the object shape, segmented region is described by one of the shape representation methods. Finally, the classification of segmented region is implemented via a classifier trained with a training set. There are numerous number of the shape representation algorithms in the literature and detail explanation of them can be found in Chapter 3.

In a particular shape based approach, Hsieh et al. [16] present hierarchical classification algorithm to accurately recognize aircrafts in satellite images. With the purpose of the achievement of rotation invariance, a new algorithm based on symmetry property is proposed to estimate the orientation of an aircraft. In addition, several image preprocessing techniques such as noise removal, binarization, and geometrical adjustments are also applied to remove distortions on shape before recognition. After these steps, distinguishable shape features are derived from each aircraft for aircraft recognition. Different features have different discrimination abilities to recognize the types of aircrafts. Therefore, a novel boosting algorithm is proposed to learn a set of proper weights from training samples for feature integration. Finally, a hierarchical recognition scheme is proposed to recognize the types of aircrafts by using the area feature at first for a rough categorization on which detailed classifications are then achieved using several suggested features.

Similar to the method introduced in Hsieh et al. [16], Eikvil et al. [17] propose a solution to the problem of vehicle detection from high resolution satellite images. Firstly, the segmentation process is implemented to extract the target objects in the image. Then, the resulting regions are described by gray-level and shape-based features, such as area, compactness, Hu moments, height and width. Next, a rule-based classifier is utilized to eliminate non-vehicle objects. Lastly, the potential vehicles are obtained by two different statistical classifiers, Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA).

Similar to the aforementioned algorithms, Bo and Jing [18] propose a method for airplane target detection in remotely sensed imagery. The techniques consist of two steps. First, segmentation of the original image is implemented to create discrete image regions used to determine which region belongs to candidates in the target area. Second, each candidate area is reprocessed into a binary image and a simple statistical feature is calculated from the binary mask for target detection.

In addition, Iisaka et al. [19] introduce a robust method to describe any shape in the remote sensing images. The shape of the object is represented by a series of pattern primitives or structural elements with varying size. The initial few coefficients in a small number of primitives set can reasonably approximate the object shape in satellite images.

In another shape-based algorithm for geospatial object detection, a contour-based spatial model is introduced by Li et al. [20] to detect targets accurately in high resolution remote sensing images. To this end, each image is firstly partitioned into parts as target candidate regions by using multiple scale segmentation. Next, the automatic identification of target seed regions is conducted by computing the similarity of contour information with the target template via dynamic programming. In the final part, the contour-based similarity is updated and combined to manage the missing parts with spatial relationship.

The shape of an object carries valuable information about the object identity and function and so the features extracted from this information is more characteristic and discriminative than ones extracted from local and global appearance of objects for target detection and recognition purpose. Nevertheless, the segmentation procedure in the beginning is the weakness of shape based methods. Sometimes, it is not robust to low foreground-background contrast, shadows and occlusions by other object conditions which are encountered in real scenarios. Thus, a robust segmentation algorithm is needed in shape based approaches.

In the literature, there are also additional algorithms, which combine two or more different methods existing in the abovementioned approaches. In recent years, these approaches become prevalent, since they take advantage of more aspects and to produce a higher performance. One example of these approaches for object detection and recognition is proposed by Murphy et al. [21] [22]. They realize the object detection and localization process by combining local and global appearance features. Thus, the ambiguity caused from local features due to the small object of interest or imaging condition is reduced by using global features, which are called as “gist” of the scene, as additional evidence. In some other methods [23] [24] [25], the shape based approaches are combined with the other type of algorithms. There are two main steps in these approaches: a hypothesis generation step and a verification step. In the hypothesis generation step, a set of hypothesis of possible object locations is generated by tuning them for low-missed detections and high false-positives. Local feature based algorithms are commonly used for this purpose. In the second step, the hypothesis points are validated by using one of the image segmentation and shape representation methods for each.

1.2 Scope of the Thesis

This thesis is devoted to the problem of geospatial object recognition from satellite images, and the proposed solution exploits shape features extracted from visual data. The motivation behind this study can be presented in three main ways. The first one is to eliminate the false alarms generated by other geospatial object detection algorithms. The second one is to solve the localization problem. The last one is to find the aspect ratio and/or size of the object in order to determine the target type by getting the object silhouette in the image.

The dissertation can be analyzed in three main steps performed sequentially for the main purpose of the thesis. For each step, various possible solutions are tested, and their performances are examined qualitatively and quantitatively. The first step of the thesis is to extract the object mask from the image. For this objective, the several image segmentation algorithms are analyzed in detail. This step provides a comparison between fully automatic and semi-automatic methods that utilize the color or intensity distribution among the images. The next step is to describe the extracted object mask via shape representation and description method(s) in the feature domain. In order to determine the method used for this purpose, various different algorithms are analyzed and compared in this block. Finally, image segmentation algorithms and shape description algorithms are integrated in order to propose a solution to the object recognition problem in vision. The general flowchart of the proposed system and parts of the thesis is shown in Figure 1.

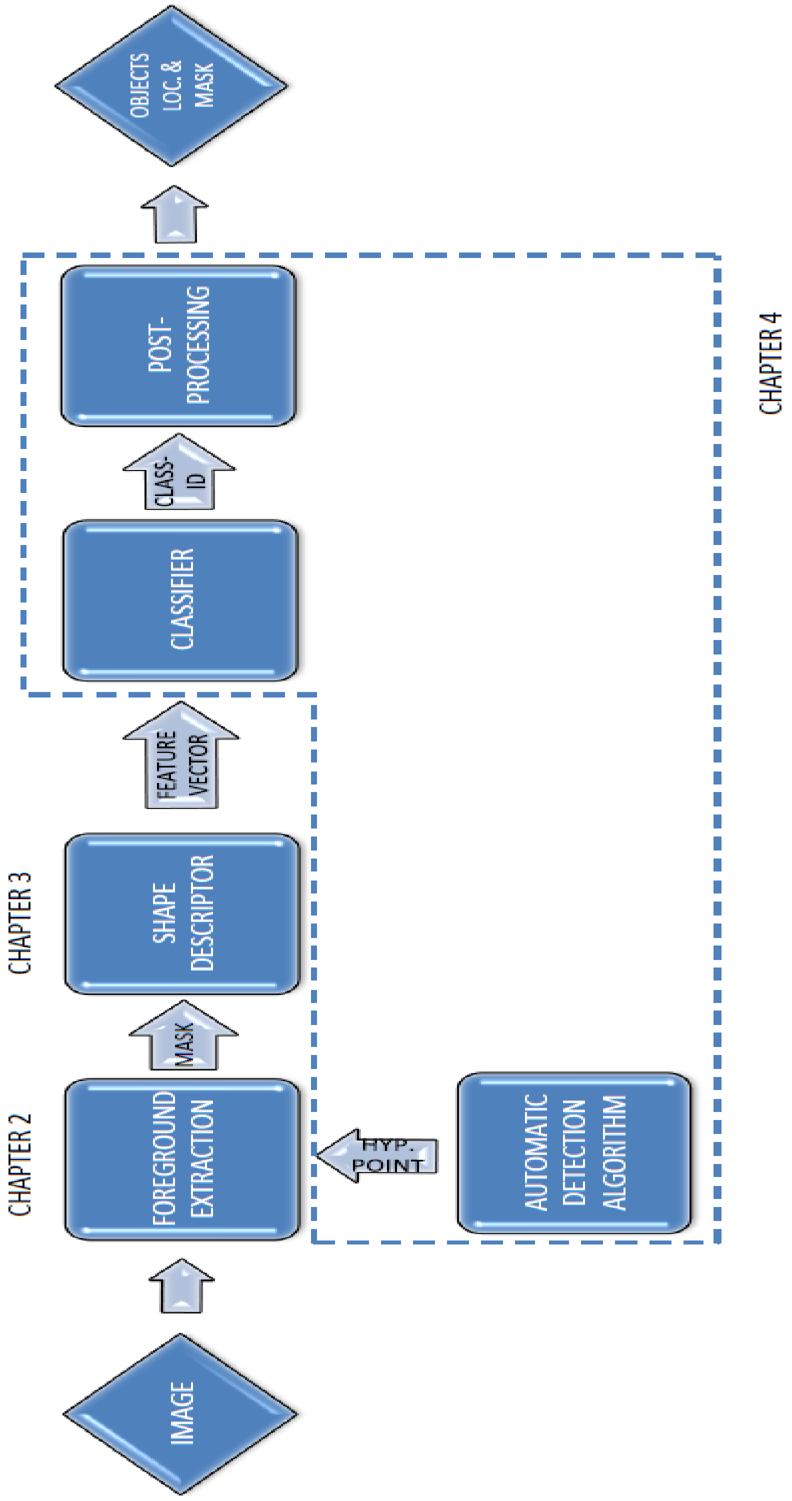


Figure 1: Overall flowchart of the proposed object recognition algorithm and sections of the thesis

1.3 Outline of the Thesis

The thesis is composed of three main parts involving the steps mentioned previously.

Chapter 2 focuses on image segmentation methods and their comparison. In the first subsection of Chapter 2, fully automatic image segmentation algorithms are analyzed. First, a literature survey on color image segmentation algorithms is presented, and methods are classified. Then, k-means clustering, mean-shift and normalized cut image segmentation algorithms are explained in detail, and comparison of these methods is performed on a variety of images with two types of targets; i.e aircrafts and ships, in an objective and subjective manner. The second subsection elaborately analyzes semi-automatic or interactive image segmentation algorithms. In this context, the interactive graph-cuts, GrowCut and GrabCut foreground extraction algorithms are represented and examined. Finally, the performances of all image segmentation methods are tested with the resulting images for subjective comparison and precision-recall values for objective comparison.

In Chapter 3, the shape representation algorithms are discussed. Initially, a literature review about shape description methods is presented, and a classification of these methods is given. Next, the performances of various shape description techniques which are angular radial transform (ART), geometric (Hu) moment invariants for region-based and Fourier descriptors for contour-based, are compared based on the Nearest Neighbor (NN) and k-NN classifiers. Finally, the experimental results are given with confusion matrices.

In Chapter 4, a novel geospatial object recognition algorithm is proposed and explained in detail. After clarifying the proposed method, the bag of visual words (BoVW) object detection algorithm is introduced for hypothesis generation in the following subsection. Next, as a classification method, support vector machine (SVM) is explained. In the final part, the proposed method is evaluated by experiments using images containing different object types subjectively and objectively.

Finally, the summary, as well as conclusions of the thesis is given in Chapter 5, and the future directions are suggested.

CHAPTER 2

IMAGE SEGMENTATION

In areas, such as computer vision and image processing, one of the most challenging problems is image segmentation, which plays an important role in many applications. Up to the present moment, innumerable segmentation methods have been proposed to solve this problem in the literature. Therefore, the problem of segmentation has been, and still is, relevant research field due to its wide range usage and application [26]-[39]. The image segmentation is usually utilized as an initial step in many high-level operations and so its accuracy is a critical and extremely significant issue for these high-level systems. Some of the practical applications of image segmentation are the medical imaging, locating objects in satellite images (road, forest, building etc.), vision-guided autonomous robotics, traffic control systems, image retrieval and segment-based image or video compression systems which may contribute to make better the human life [26].

The image segmentation is verbally defined as the process of domain independent partitioning an image into a set of disjoint (non-overlapping) regions that are visually distinct, homogenous and meaningful with respect to some characteristics or computed properties, such as gray level intensity, texture or color that makes image analysis (e.g. object identification, classification and reconstruction) easy [27] [28] [29] [30]. The formal or mathematical definition of segmentation is can be stated as follows [31]:

Let the image domain be I and P ( ) be a uniformity predicate defined on groups of the connected pixels. A segmentation result of I is a partitioning set of connected subset or image region $\{R_1, R_2, \dots, R_n\}$ such that

$$\bigcup_{i=1}^{n} R_i = I\,, \quad where\ R_l \cap R_m = \emptyset\,, \quad \forall\, l \neq m \tag{2.1}$$

and the uniformity predicate satisfies

$$P(R_i) = True,\ \forall\, i \tag{2.2}$$

$$P(R_m \cup R_l) = False, \qquad \forall\, R_l\ adjacent\ to\ R_m \tag{2.3}$$

$$(R_l\ \supset R_m) \wedge (R_m \neq \emptyset) \wedge (P(R_l) = True) \Rightarrow P(R_m) = True \tag{2.4}$$

To be able to exploit any segmentation results for high-level operations (e.g. object identification, classification, representation), it is desirable to succeed semantically meaningful segmentation output. Despite of the fact that many diverse algorithms for this issue have been proposed in the literature in the last couple of decades, exactly accurate and complete solution has not been found yet (still far from being) due to its complicated nature. In general, the difficulty of this problem originates from the ambiguity in the description of "*semantically meaningful*" segmentation. In other words, even two people might segment a presented image differently due to the individual human perception [32]. Since there is a lack of uniqueness criteria for the solution of segmentation, the general and complete solution cannot be acquired. Nevertheless, the unique and meaningful solution might only be obtained by considering some rules or constraints on the segmentation problem.

Within the scope of this thesis, image segmentation is utilized as an initial step in the object recognition process. To be able to identify the targets in the image precisely, the shape of the desired objects should be holistically extracted via any segmentation algorithm. In the light of this information, the "*semantically meaningful*" segmentation can be described a foreground/background separation such that the desired object should be thoroughly extracted as the foreground and the remaining parts are considered as the background. Therefore, this study does not attach to importance to the segmentation accuracy within the background parts, and this is also called as foreground extraction or foreground/background segmentation in the literature.

In this sense, two main types of the segmentation classes in the literature are examined for this purpose; fully automatic image segmentation algorithms and semi-automatic or interactive image segmentation algorithms. In the semi-automatic or interactive image segmentation algorithms, the user interaction, which identifies some part of the foreground and/or the background, is desired as input, whereas the fully automatic image segmentation algorithms does not need. In the first part of this chapter, three well-known fully automatic segmentation algorithms are briefly introduced, and foreground extraction performances are compared. In the following section, three popular interactive segmentation methods are described, and the performance comparison of fully and semi-automatic image segmentation methods are realized.

2.1 Fully Automatic Image Segmentation

Discontinuity and similarity/homogeneity are generally two fundamental properties of the pixels in relation to their local neighborhood used for segmentation purpose [30], [33], [34]. The segmentation methods based on discontinuity of pixels, which are called as boundary-based or contour-based methods, intend to identify some basic discontinuities (e.g. points, edges and lines) which constitute boundaries existing among the desired regions. On the other hand, the segmentation methods based on similarity or homogeneity of pixels, which are called region-based methods, utilize some similarity (homogeneity) property between the neighboring pixels to categorize image pixels into meaningful regions [30], [33], [34]. There are also many suggested segmentation methods based on both discontinuity and similarity property, which are called hybrid-based methods, to improve the segmentation performance in terms of accuracy and completeness [33], [35] [36] [37]

The fully automatic image segmentation techniques in the literature can be classified into four main categories in the following manner [38]:

- Thresholding based techniques
- Clustering based techniques
- Edge/boundary - based techniques
- Region - based techniques

In the thresholding techniques [38], [39], the thresholding process converts a multi-level image into binary image i.e., it assigns the value of 0 (background) or 1 (objects or foreground) to each pixel of an image based on comparison with some threshold value T, which is determined by using intensity or color distribution of the image by means of its histogram. In this approach, the neighborhood relations between image pixels are not considered and so the segmentation results may have disjoint pixels, which is usually an undesired case.

In the clustering approach [38] [39], a data set (image pixels) is combined into clusters with respect to pixel color, intensity or other computed features and then center values (means) of the clusters are assigned into the pixel values in the same cluster to constitute homogenous regions. The general principle in the clustering approach is that pixels within each cluster should display a high degree of similarity, while they should present a very low resemblance across different clusters. One of the most commonly used methods is the k-means algorithm.

The other group of techniques is the edge/boundary-based methods [38] [39] and the segmentation operation is realized by detecting edges or discontinuities among regions in the image. Classical edge detection methods, such as Laplacian of Gaussian, the Canny edge detector, can be used to identify the edges or boundaries in the image.

The final group of methods is the region-based segmentation [38] [39] whose goal is to exploit image features to map individual pixels in an input image to a set of pixels called regions that might correspond to an object or a meaningful part of it. The region-based segmentation algorithms can be classified into two sub-groups, the global and local methods. In the local methods, the algorithms usually begin with a single pixel and continue by combining the pixels until getting totally connected regions. These methods achieve the segmentation operations with a bottom-to-top approach. The global methods [40] [41] , on the contrary, are in a top-to bottom approach and realized such that an image is initially considered as a single region and the segmentation process is continued by sequentially splitting this region into smaller, homogenous regions.

Moreover, there are many hybrid-based methods [33] [35] [36] [37] which combine different segmentation algorithms in one algorithm to achieve holistically correct segmentation outputs.

This section is composed of five main sub-sections. In the first two subsections, the algorithmic steps of the k-means clustering [42], mean-shift [43] image segmentation methods are given in detail, respectively. Next, the definition and mathematical expression of the graph theory are introduced. After that, the detailed information about normalized cuts [40] image segmentation algorithm is given. Finally, the comparison among three different algorithms is performed in experimental results part with the segmentation outputs of different set of images qualitatively and quantitatively.

2.1.1 K-means Clustering Image Segmentation

K-Means algorithm [42], [44] is an unsupervised clustering technique that follows the simple procedure to classify a given input data set into multiple classes based on their inherent distance from each other [45]. The algorithm updates the partition of the given feature space iteratively, where the points in the feature space are exchanged between classes based on a predefined metric (e.g. the Euclidian distance between cluster centers) in order to satisfy the criteria of minimizing the variation within each cluster and maximizing the difference between the resulting clusters. The algorithm is iterated until no exchanged between clusters. As a feature, the RGB color value of the each pixel is utilized for the segmentation purpose. The traditional k-means algorithm steps are summarized as follows [45]:

1. Initially, the number of clusters K is preselected. Next, for every region random initial feature points are selected for the cluster centers μ_i.

2. Cluster the points based on Euclidian distance of their color values from centroid values according to (2.5).

$$c^{(i)} := arg \min_j \left\| x^{(i)} - \mu_j \right\|^2 \qquad (2.5\ )$$

3. Recalculate the centers of clusters receiving new data points and clusters losing data points according to (2.6).

$$\mu_i := \frac{\sum_{i=1}^{m} 1\{c_{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c_{(i)} = j\}} \tag{2.6}$$

4. Repeat the steps 2 and 3 until the cluster labels of the image do not change anymore.

where i iterate over the all the feature points, j iterates over all of the centroids and μ_i is the centroid points.

The main drawback of k-means clustering algorithm is that the spatial coherence of the regions is not taken into account and; thus, it generally yields spatially unconnected and incoherent regions. The other limitation is the uncertainty in selection starting points chosen for cluster centers. This deficiency causes the different algorithmic result for each run.

2.1.2 Mean-Shift Segmentation

Mean-shift (MS) [43] is a general non-parametric iterative technique proposed for analyzing a complex multimodal feature space. The mean shift paradigm can provide reliable solutions for many vision tasks due to its adaptable and excellent qualities. Therefore, it is widely used in many computer vision applications, such as discontinuity preserving smoothing, image segmentation, object tracking and identification. In the mean shift framework, the local peaks or modes of the multivariate probability density function (pdf) are tried to be estimated from the observed data sampled from this density function. Dense regions in the input feature space correspond to local peaks of the pdf or to the modes of the unknown density function. As soon as locations of the local peaks are obtained, the clusters associated with individual modes are generated around these modes based on the local structure of the feature space [43]. The most noticeable advantages of the mean shift algorithm are the automatic detection of the number of clusters and the lack of a fixed shaped cluster density. On the other hand, the most significant drawback is the high computational cost. However, various optimization techniques are proposed to minimize the computational cost recently [46], [47].

Before progressing to the algorithmic details, it should be mentioned about the feature space used in the mean shift image segmentation algorithm. An image is typically represented as a two-dimensional lattice of p - dimensional vectors (i.e. pixels), where $p = 1$ for gray-level images and $p = 3$ for color images. The space of lattice is evaluated as the *spatial domain*, whereas the color or gray-level intensity information is considered in the *range domain* [48]. The features space used in the segmentation algorithm consists of the color (range) and spatial domain information of the image pixels. To be able to get a meaningful segmentation output, the image in RGB color space is initially converted to another color space so that the perceived color differences correspond to Euclidian distances in the color space chosen to represent the pixels. Since an Euclidian metric is not guaranteed for the RGB color space, *L*u*v and L*a*b* color spaces were designed to best approximate perceptually uniform color spaces. The new color spaces are created by the nonlinear dependency on RGB [43]. *L* coordinate defines the “lightness” or “luminance” and the other coordinates define “chrominance” for both color spaces. While generating the feature space, the location and color vectors are concatenated in the joint domain whose dimension is $d = p + 2$, and the normalization operation is performed to be able to compensate their different nature.

After establishing the feature space utilized in the segmentation, algorithm details can be explained. The mean shift segmentation procedure is composed of two basic steps; a mean shift filtering of the original image data in feature space (*filtering step*) and then clustering of the filtered data points or the modes (*fusion step*) [48]. The detailed explanation for each of these steps will be given in the below.

2.1.2.1 Filtering step

The main purpose of this step is the determination of the modes of the underlying pdf and associating them any points in the basin of attraction. This process can be performed as follows [43]:

> Let $\mathbf{x}_i$ and $\mathbf{z}_i$, $i = 1, \ldots, n$ be the $\boldsymbol{d}$–dimensional input and filtered image pixels in the joint spatial-range domain, respectively. For a color image, $\boldsymbol{d}$ is equal to five (two of them from the spatial domain and other ones from the range domain). Let $\boldsymbol{L_i}$ be the label of the $\boldsymbol{i}^{\text{th}}$ pixel in the segmented image.

1. Initialize $\boldsymbol{j}$=1 and $\mathbf{y}_{i,1} = \mathbf{x}_i$ for all $\boldsymbol{i} = 1, \ldots, n$. (Denoting $\{\mathbf{y}_j\}_{j=1,2,\ldots}$ the sequence of consecutive locations of the kernel G)

2. Compute the next iterant $\mathbf{y}_{i,j+1}$ according to (2.7) until convergence, $\mathbf{y} = \mathbf{y}_{i,c}$ for all $\boldsymbol{i}$

$$\boldsymbol{y_{j+1}} = \frac{\sum_{i=1}^{n} \boldsymbol{x_i} g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} \qquad j = 1,2,\ldots \tag{2.7}$$

where h denotes the bandwidth parameter and g denotes the profile of symmetric kernel G. The bandwidth parameter h reveals as h_s for spatial radius and h_r for range (color) radius in this case, and $\boldsymbol{x_i}$ represents the data points (pixels in feature space) within the window defined by the bandwidth parameters, h_s and h_r.

3. Assign $\boldsymbol{z_i} = (\mathbf{x}_i^s, \mathbf{y}_{i,c}^r)$ for all $\boldsymbol{i} = 1,2\dots$ n, where the superscripts s and r denote the spatial and range components of a vector, respectively. The assignment states that the filtered data at the spatial location $\mathbf{x_i}^{\mathbf{s}}$ will have the range component of the point of convergence $\mathbf{y}_{i,c}^r$.

The kernel window in the mean shift procedure moves in the direction of the maximum increase of the joint spatial-range density gradient and the filtering process is guaranteed to converge. In addition, the determination of the mode associated with each data point smoothes the image, while preserving discontinuity. For example, if two points' $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ are far from each other in the feature space, $\boldsymbol{x_j}$ does not contribute to the mean shift vector gradient estimates of $\boldsymbol{x_i}$ and thus the trajectory of it will move away from $\boldsymbol{x_j}$. As a result, pixels on either side of strong discontinuity will not affect each other [48].

2.1.2.2 Fusion step

After running the mean shift filtering process for image and determining all the information about the convergence points or modes in $\boldsymbol{z_i}$, the fusion operation is performed in order to eliminate the noisy modes. This clustering operation can be carried out as follows [43]:

1. Delineate in the joint domain the clusters $\{C_p\}_{p=1,\ldots m}$ by grouping all $\mathbf{z_i}$ which are closer than h_s in the spatial domain and h_r in the range domain, i.e., concatenate the basin of attraction of the corresponding convergence points.

2. For each $i = 1,2, \ldots.,n,$ assign $\boldsymbol{L_i} = \{p| \ \boldsymbol{z_i} \in C_p\}$

3. Optionally, eliminate spatial regions containing less than M pixels and assign them to the closest neighboring region in the range domain (in terms of the norm of average region intensity or color differences).

The fusion step is considered as basic post-processing and so it is only implemented by simple single linkage clustering. However, this step can be refined by using some other methods for clustering purpose. One of such methods is that a region adjacency graph (RAG) is created in order to cluster the modes hierarchically. Moreover, edge knowledge from an edge detector and the color information is brought together to achieve better clustering performance [46].

To make better visualization, the algorithm steps are illustrated with an example in Figure 2. In Figure 2.b., the mean shift trajectories associated with every pixel and their modes are shown. The modes of hills that are available at the original image in the Figure 2.a. are localized in the filtering stage result, as shown in Figure 2.c, and this causes quasi-homogenous regions at the output. The effect of fusion operation is displayed in Figure 2.d.

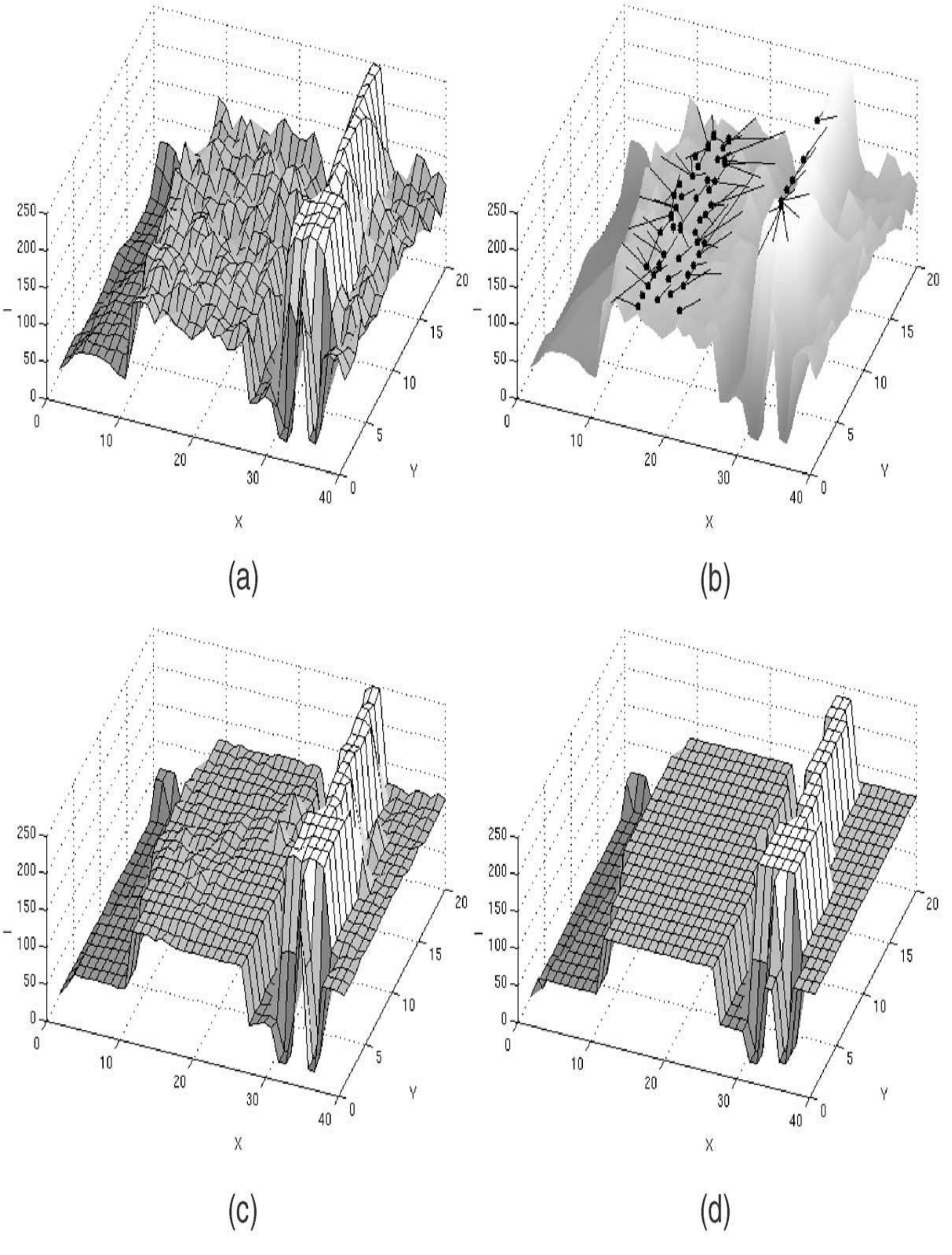


Figure 2: Visualization of mean shift segmentation steps. (a) The original data. (b) Mean shift trajectories for the pixels. The black dots show the modes. (c) Filtering stage result. (d) Fusion stage result [43].

2.1.3 Fundamentals of Graph Theory

A graph is an abstract representation of a set of nodes where some pairs of the nodes are connected by links. The interconnected nodes are represented by mathematical abstractions called as *vertices,* and links that connect some pairs of vertices are called as *edges*. The mathematical representation of a graph is given as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ composed of a set $\mathcal{V}$ of vertices or nodes together with a set $\mathcal{E}$ of edges or links. The example of a graph with six vertices and seven edges is shown in the Figure 3.

A graph is called as a weighted graph if a value (weight) is assigned to each edge. Such weights represent the connection strengths, costs, lengths or capacities etc. (depending on the problem at hand) between two vertices. If each pair of vertices is connected by an edge in the graph, then the graph is called as a *complete graph*. However, each vertex is not necessarily connected to every other vertex in many applications and such a graph is called as *partial graph.*

For some application, the direction of an edge can be significant for the weighting of the graph and the weights can vary according to the direction, this kind of graphs is called as *directed weighted graphs*. On the other hand, if the edge direction is not decisive and the weight values are equivalent in both directions, that graph is called as *undirected weighted graphs*.

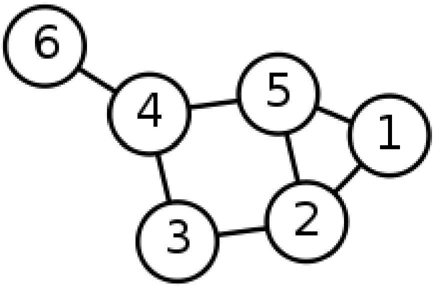


Figure 3: A simple graph with six vertices [49]

2.1.4 Normalized Cut Segmentation

Normalized cut image segmentation [40] is a global approach that solves the coherent perceptual grouping problem in vision. It is a global method due to its focus on the global impression of an image instead of its local features. The normalized cut algorithm considers the image segmentation problem as a graph partitioning problem that is solved by utilizing a splitting process beginning from the entire image into the bottom parts. The splitting process can be implemented by using the decomposition of eigenvectors of special matrices (e.g. affinity matrices) of constructed graph, which examines the spectral properties of affinity matrices. There exist different kinds of special (affinity) matrices to be decomposed in the literature. A review and comparison of these methods are represented in [50] and it is examined that the normalized cut method shows better performance than the other methods for overall case.

The normalized cut image segmentation algorithm is typically divided into two main steps, as *the formation of the graph* and *iterative graph partitioning.* The first step is the formation of the graph which initially maps the image to a graph that holds the relations between pixels. A set of points in an arbitrary feature space is represented as an undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which the vertices of the graph correspond to the points in the feature space and an edge is formed between every pair of vertices. The weight on each link, $\omega(i, j)$, is a function of the similarity between nodes i and j and the function is defined as the product of a feature similarity term and spatial proximity term as follows:

$$\omega_{i,j} = e^{\frac{-\|F(i)-F(j)\|_2^2}{\sigma_I^2}} * \begin{cases} e^{\frac{-\|X(i)-X(j)\|_2^2}{\sigma_X^2}} & if \quad \|X(i) - X(j)\|_2 < R \\ 0 & otherwise \end{cases} \tag{2.8}$$

The weight values resulted from this function reflect the likelihood that two pixels belong to the same object. In the cost function, $F(i)$ is the feature vector based on intensity, color or texture info at pixel i and $X(i)$ is the spatial term and indicates the spatial location of that pixel. According to this weight function, the constructed graph is partial (not complete), since the pixels that are located within a circle of radius *R* are only connected to each other. σ_I and σ_X are the parameters of feature similarity and spatial proximity terms, respectively.

The second step of the segmentation algorithm is the iterative graph partitioning and it is accomplished by iteratively partitioning the set of vertices $\mathcal{V}$ into disjoint sets $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_m$ where by some measure, the similarity among the vertices $\mathcal{V}_i$ is high and, across different sets, $\mathcal{V}_i$ an*d* $\mathcal{V}_j$ is low. At each iteration, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is divided into two disjoint sub-graph, *A* and *B*, such that $A \cap B = 0$, $A \cup B = V$, by simply removing edges connecting the two parts. A cut on a link separates the two nodes connected to each other with the corresponding link. The cuts on a graph generally involve more than one link and so the separation of two groups of nodes can be achieved as in the Figure 4. Every cut in the graph has a cost value, and it is calculated by adding the weight value of the removed links (cut). For example, the cost of the cut is the summation of the link weights of 2 and 5 in the Figure 4.

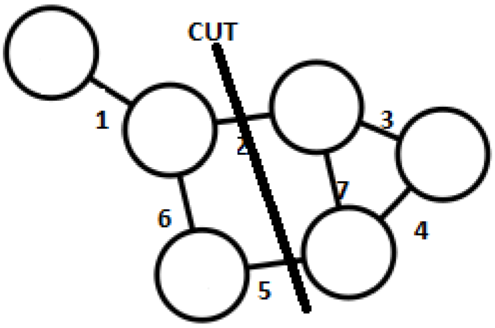


Figure 4: A simple cut on a graph

The spectral-based segmentation methods [40], [41], [51], [52] try to minimize a cost function that is totally related to the cost of the cuts and the association of the partitions during the segmentation process. The difference between those algorithms originates from the difference in the cost function to be minimized. The well-known three methods are reviewed and compared in [53] and the normalized cut method outperforms the other methods due to its normalization during the formulation of the segmentation algorithm. The degree of dissimilarity between the two sub-graphs can be computed as the summation of the weights of the links that have been removed (cut) and it is formally defined as:

$$cut(A,B) = \sum_{i\in A, j\in B} \omega(i,j) \tag{2.9}$$

The minimum cut algorithm [51] [52] tries to find the cut combination which minimizes this cut cost. However, as noticed in [40] [51], the minimum cut criteria favors cutting the graph into small pieces, especially if there are nodes located at distant locations (i.e. isolated nodes), resulting in a desperate segmentation. Figure 5 illustrates one such case. Assuming the edge weights are inversely proportional to the distance between the two nodes, it can be seen that the cut separating the node n_1 or n_2 will have quite a small cut cost, whereas bad partition.

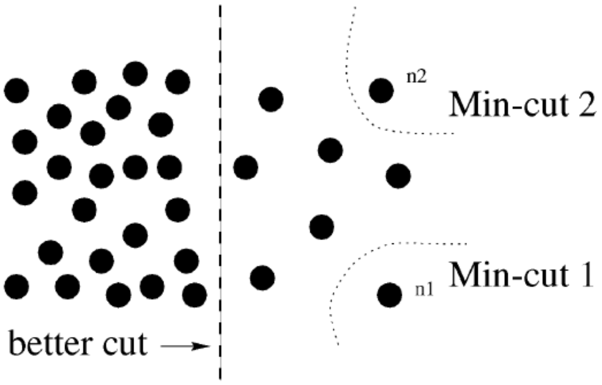


Figure 5: A case where minimum cut gives a bad partition [40]

To prevent the unnatural bias of the minimum cut algorithm for partitioning out small sets of isolated points, the new global criterion is defined in the normalized cut segmentation algorithm. In this criterion, the cut cost in the minimum cut algorithm is normalized with the total edge weights obtained by adding the total edge weights of the nodes in the separated groups and it is called as *Normalized cut* (Ncut):

$$Ncut\ (A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} \qquad (\,2.10)$$

where

$$assoc(A,V) = \sum_{i\in A, j\in V} \omega(i,j) \qquad (2.11\,)$$

In this formula, *cut (A*, *B)* indicates the cut cost computed between partition *A* and partition *B*. *assoc(A,V)* is the summation of total weight from nodes in *A* to all nodes in the graph *V*. The normalized measure reflects how tightly the nodes within the different separated groups are connected to each other.

The exact minimization of the normalized cut is a *np-complete* problem. Nevertheless, when the normalized cut problem is reformulated in real-valued domain by using the change of variables, an approximate solution can be obtained efficiently [40]. The theoretical analysis and the formation of the new formulation [40] are clarified in detail in Appendix A. This new formulation is rewritten as,

$$Ncut(A,B) = \frac{y^T(D-W)y}{y^TDy} \qquad (2.12)$$

where D is an $N\ x\ N$ diagonal matrix of a graph with N nodes and it states the total edge weights belonging to each node individually. W is an $N\ x\ N$ symmetrical matrix, the affinity matrix, of the graph indicating the similarities between the nodes and the entries of this matrix are $\omega(i, j)$ such that the diagonal entries are ***1*** because each node is completely similar to itself. Finally, y is an *N x 1* vector whose elements is real and corresponds to the similarities of the nodes satisfying the following constraints:

$$y(i) \in \{1, -b\} \tag{2.13}$$

and

$$y^T D \bar{1} = 0 \tag{2.14}$$

The open form of b is represented in Appendix A, and $\bar{1}$ is an *N x 1* vector whose rows are all “1”. The separation of the values in y identifies the partitioning such that the same signed nodes belong to sthe same group

The expression in (2.12) is the Rayleigh quotient [54] and if y is relaxed to take on real values, it can be minimized by solving the generalized eigenvalue system;

$$(D - W)y = \lambda D y \tag{2.15}$$

$$\Longrightarrow\ D^{-1/2}(D - W)D^{-1/2}y = \lambda y \tag{2.16}$$

The second smallest eigenvector of the generalized eigenvalue system in (2.15) and (2.16) is the real valued solution of the normalized cut problem [40]. However, there are two constraints on y which come from the condition on the corresponding indicator vector x stated in (2.13) and (2.14). The constraint in (2.14) is automatically satisfied by the solution of the generalized eigensystem.

On the other hand, the constraint in (2.13) is violated, since y has different real valued elements after the eigenvalue decomposition. Therefore, an approximate solution is obtained by means of discretization instead of exact one.

In summary, the eigenvalue decomposition and determination of the second smallest eigenvector provides a partitioning on the corresponding graph. Moreover, this algorithm recursively performs the segmentation as follows:

1. Construct a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ by taking each pixel as a node and linking each pair of pixels by a weighted edge. The weight of that edge should reflect the likelihood that two pixels belong to one object.

2. Create the D and W matrices through the edge weights.

3. Solve the generalized eigenvalue system in (2.15) and (2.16) for the eigenvectors with smallest eigenvalues.

4. Use the eigenvector with the second smallest eigenvalue to bipartition the graph

5. Decide whether the current partition should be subdivided and recursively bipartition the segmented parts if necessary.

The partitioning is implemented by thresholding the second smallest eigenvector and labeling it into two parts such that the nodes above a threshold and the ones below the threshold. This threshold is specified by a one-dimensional search between the minimum and the maximum values within the elements of the eigenvector, and the partition that minimizes the normalized cut value given in (2.12). The minimum normalized cut value evaluated during the one dimensional search determines the decision of the repartitioning the segmented parts. If the value is small enough, then the repartitioning is not performed for the corresponding segment. If it has high value, then the recursive partitioning continues.

2.1.5 Experimental Results

In this sub-section, the segmentation performances of the described fully automatic algorithms are evaluated on several test images via subjective and objective comparisons. Ten test images, while half of them are for airplanes and the other half are for ships, are utilized during the experiments with different scene complexities. The airplane images are acquired from different civilian and military airports, and the ship images are obtained from various different harbors via Google Earth. Those images are represented in Figure 6 and Figure 7 with their corresponding ground truth. The corresponding ground truth segmentations are generated manually.

During the experiments, each algorithm is executed three times with different parameters for each test image and produces three segmented images with different number of segments. Figure 8 and Figure 9 display the results of k-means clustering segmentation algorithm with 3, 5 and 10 segments for airplane and ship images. Similarly, normalized cut segmentation results with different number of segments are given for same images in Figure 10 and Figure 11. Finally, the mean-shift segmentation algorithm results are represented in Figure 12 and Figure 13. As it can be observed from the results, all of the examined fully automatic segmentation algorithms do not produce smooth region boundaries overlapping with the ground truth segment boundaries.

As it can be observed, k-means segmentation algorithm produces the segmented regions with arbitrary topological properties; in other words, the segmented regions might consist of several isolated connected blobs in the image. This result is due to the fact that k-means segmentation algorithm does not utilize the spatial connectivity constraint which takes into account the spatial coherence of the segmented regions. Moreover, it depends on the initially selected cluster centers, the number of clusters and other predefined parameters. Due to the aforementioned reasons, although some solid colored targets (e.g. white colored civilian planes) can precisely be extracted from the image by using this algorithm, most of the multicolored targets (e.g. multicolored airplanes and ships) cannot be segmented as a whole to be able to apply for the target recognition.

The implementation of Cour et al. [55] for normalized cut image segmentation algorithm is used during the experiments. For this algorithm, the initial number of segments must be specified as in k-means segmentation algorithm. The experimental results show that the boundaries of segmented regions do not coincide with ground truth segment boundaries for especially small number of segments. The reason of this result is that normalized cut algorithm takes into account the global characteristic of the target image instead of its local sense. Nevertheless, the overlapping between the boundaries of segmented regions and ground truth can be occurred partially by over-segmenting the target image via the increment of the number of segments. However, these results cannot be used for target recognition purposes due to incomplete extraction of the target.

The third segmentation algorithm, the mean-shift method, outperforms other methods and relatively produces segmented regions whose boundaries overlap with the regions of the ground truth segmentation owing to the discontinuity preserving property of mean-shift algorithm. However, the performance of the algorithm highly depends on the predefined parameter set such as minimum segment area, spatial and range bandwidth. As it can be observed from the results, the parameter set (minimum segment area, spatial and range bandwidth) should be precisely defined in order to get correct segmentation regions for each target image. Thus, target extraction with mean-shift segmentation algorithm cannot be possible for many images due to the parameter dependency and inadequacy of the algorithm.

Up to this point, the performance of the segmentation algorithms has been evaluated in terms of subjective metrics. For an objective evaluation, a performance metric, precision-recall values of the segmentation results with respect to the ground truth segmentation, is used in this work. Before giving the definition of precision and recall values, one should define the following concepts.

- True Positives, *tp* : The number of items correctly labeled as belonging to the positive class (i.e. correct results)
- False Positive, *fp*: The number of items incorrectly labeled as belonging to the negative class (i.e. unexpected results)
- True Negative, *tn* : The number of items correctly labeled as belonging to the negative class (i.e. correct absence of results)
- False Negative, *fn* : The number of items incorrectly labeled as belonging to the positive class (i.e. missing results)

Hence, precision and recall values are calculated by the following expressions,

$$Precision = \frac{tp}{tp + fp} \tag{2.17}$$

$$Recall = \frac{tp}{tp + fn} \tag{2.18}$$

In the light of the abovementioned definitions for the segmentation, *recall* is defined as the ratio of the number of correctly labeled target pixels (true positive) over the total number of target pixels in the image (true positive + false negative). *Precision* is described as the ratio of the number of correctly labeled target pixels (true positive) over the total number of pixels which are labeled as target (true positive + false positive). While calculating the precision-recall values during the evaluation of the fully automatic segmentation algorithms, the segment, which maximally coincides with the ground truth of the corresponding target is accepted as the target mask (true positive + false negative) obtained from the segmentation.

The precision-recall values of k-means clustering, normalized cut and means-shift segmentation algorithms are represented in the Tables 1-6, respectively. As it can be seen from these tables, the subjective evaluations are supported by the precision – recall values. As expected, mean-shift algorithm outperforms other methods and yields the highest precision-recall measures. Nevertheless, the recall values of that are far from the desired ones even though the precision is satisfactory.

Consequently, it is not possible (until now) to extract the foreground or target completely and accurately with fully automatic segmentation algorithms.

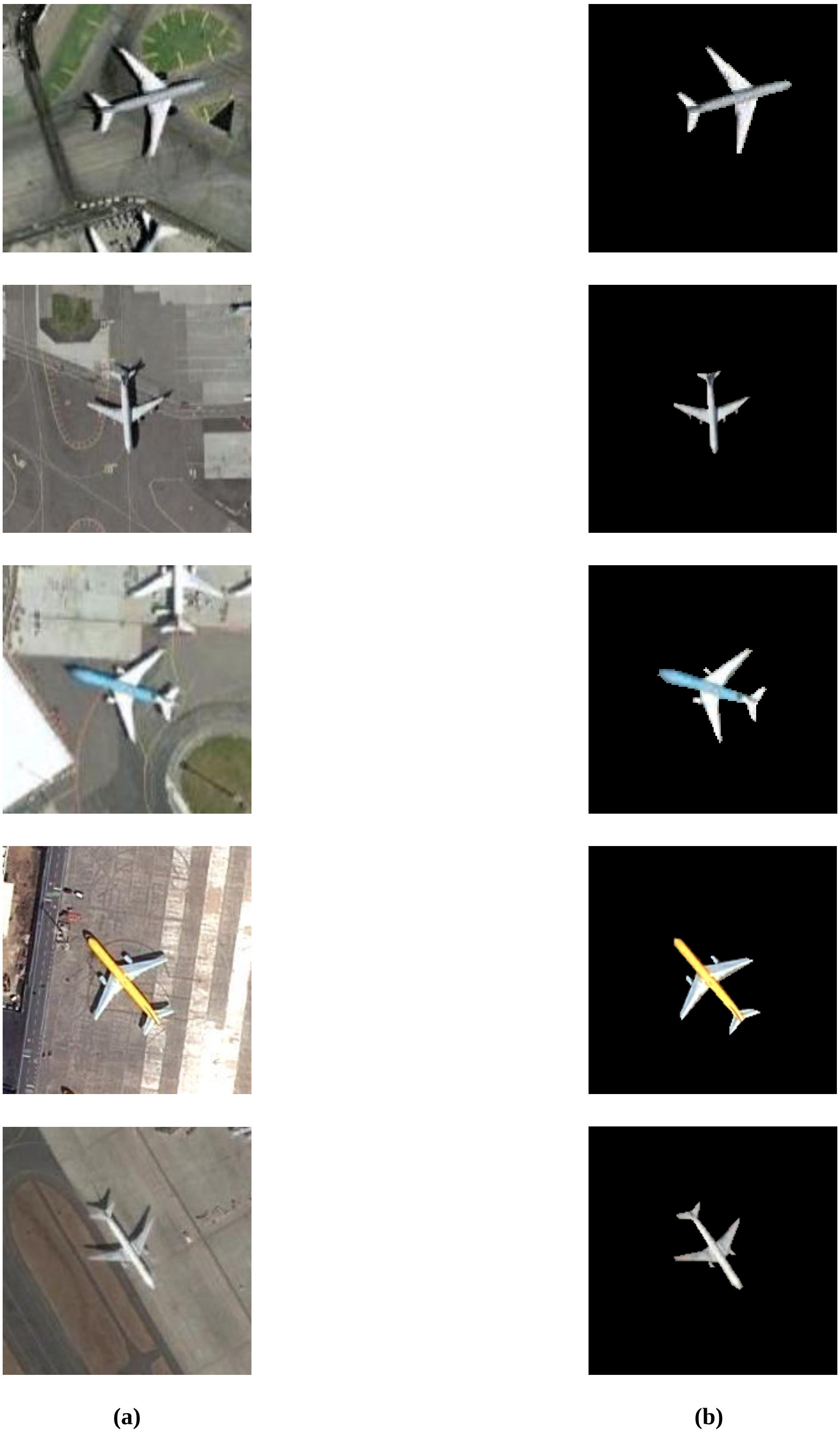


Figure 6: (a) Original plane images (b) The ground truth segmentation

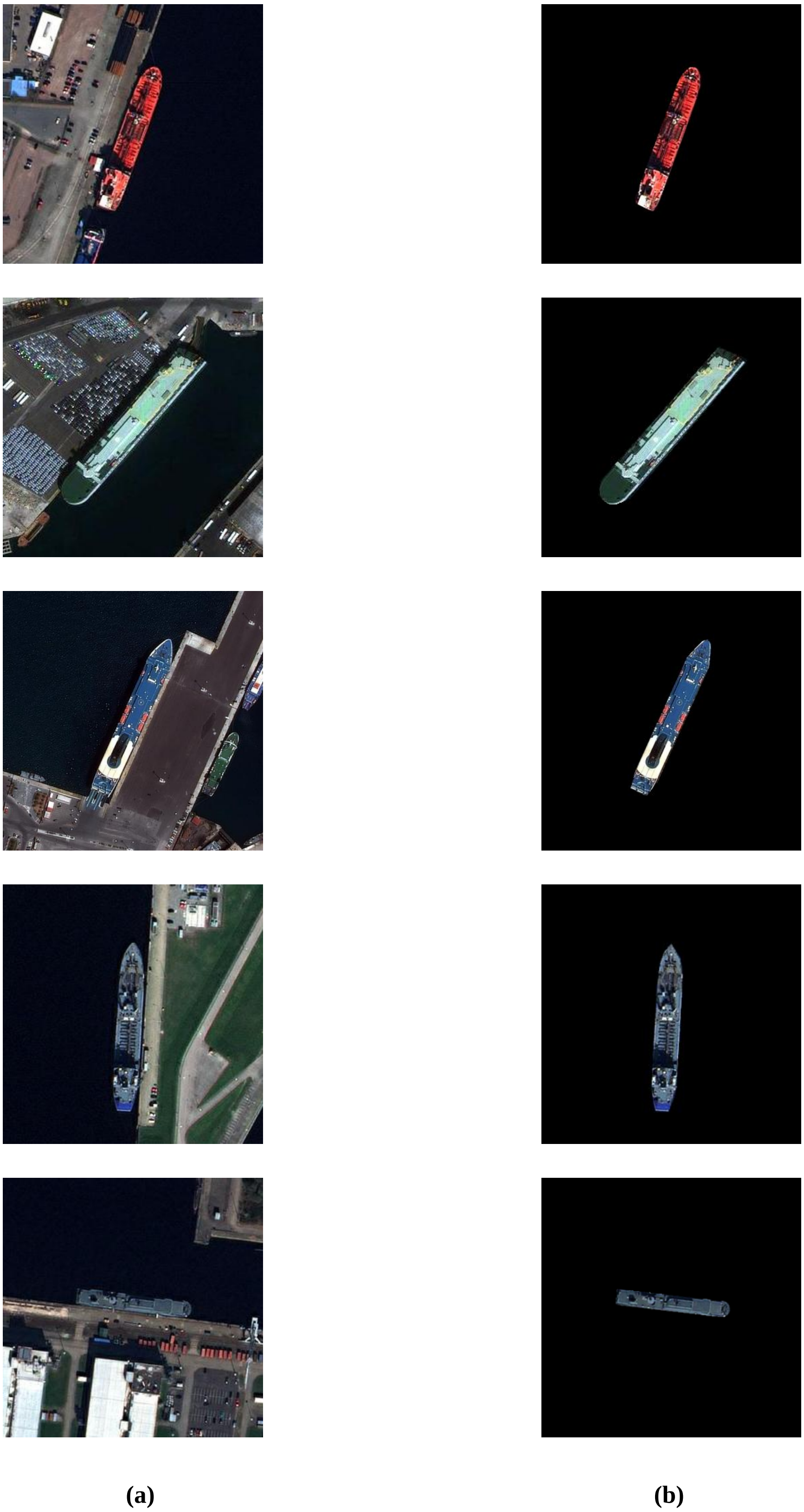


Figure 7 : (a) Original ship images (b) The ground truth segmentation

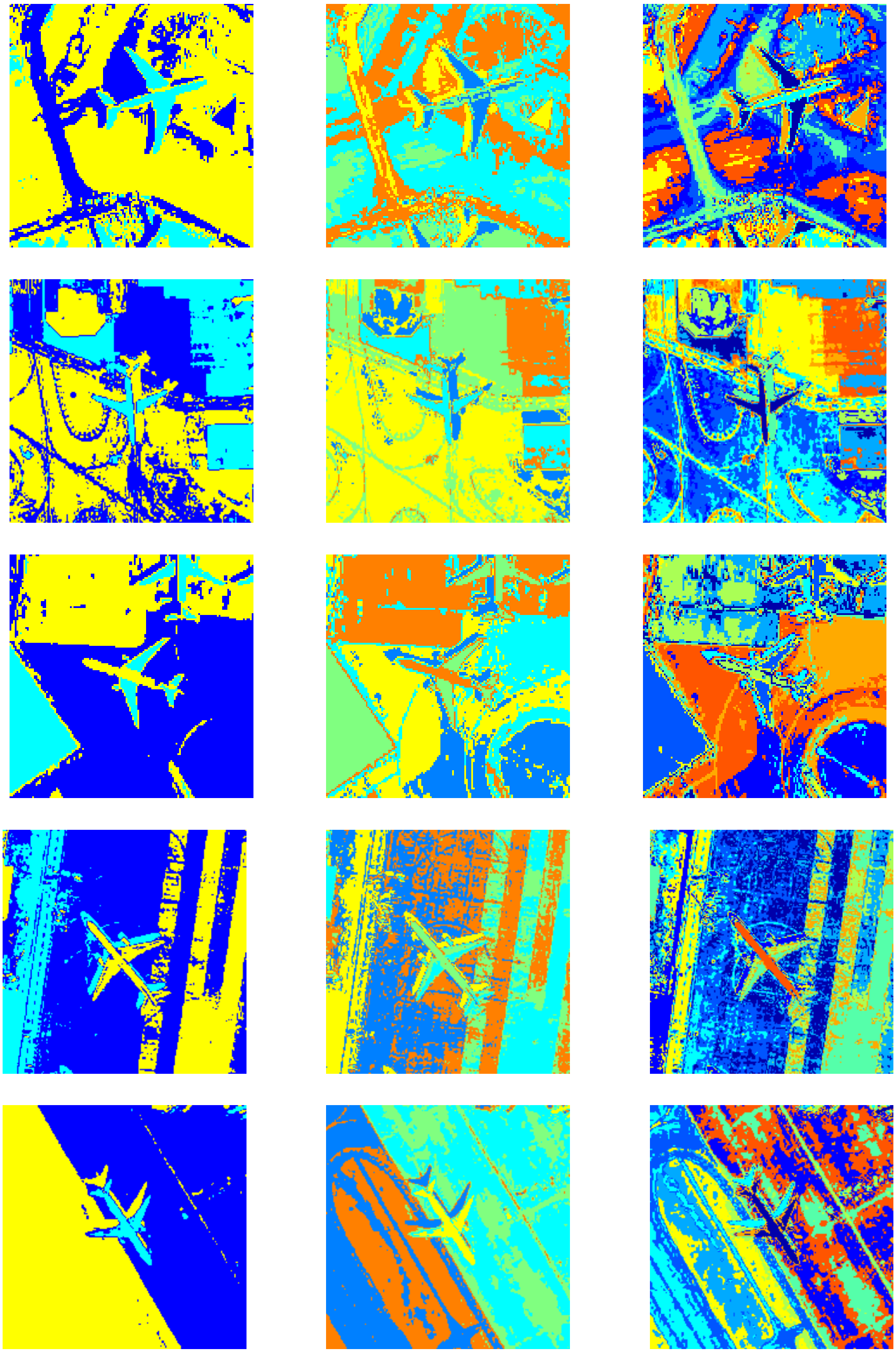


Figure 8: K-means segmentation results for plane images (a) k = 3 (b) k = 5   (c) k = 10

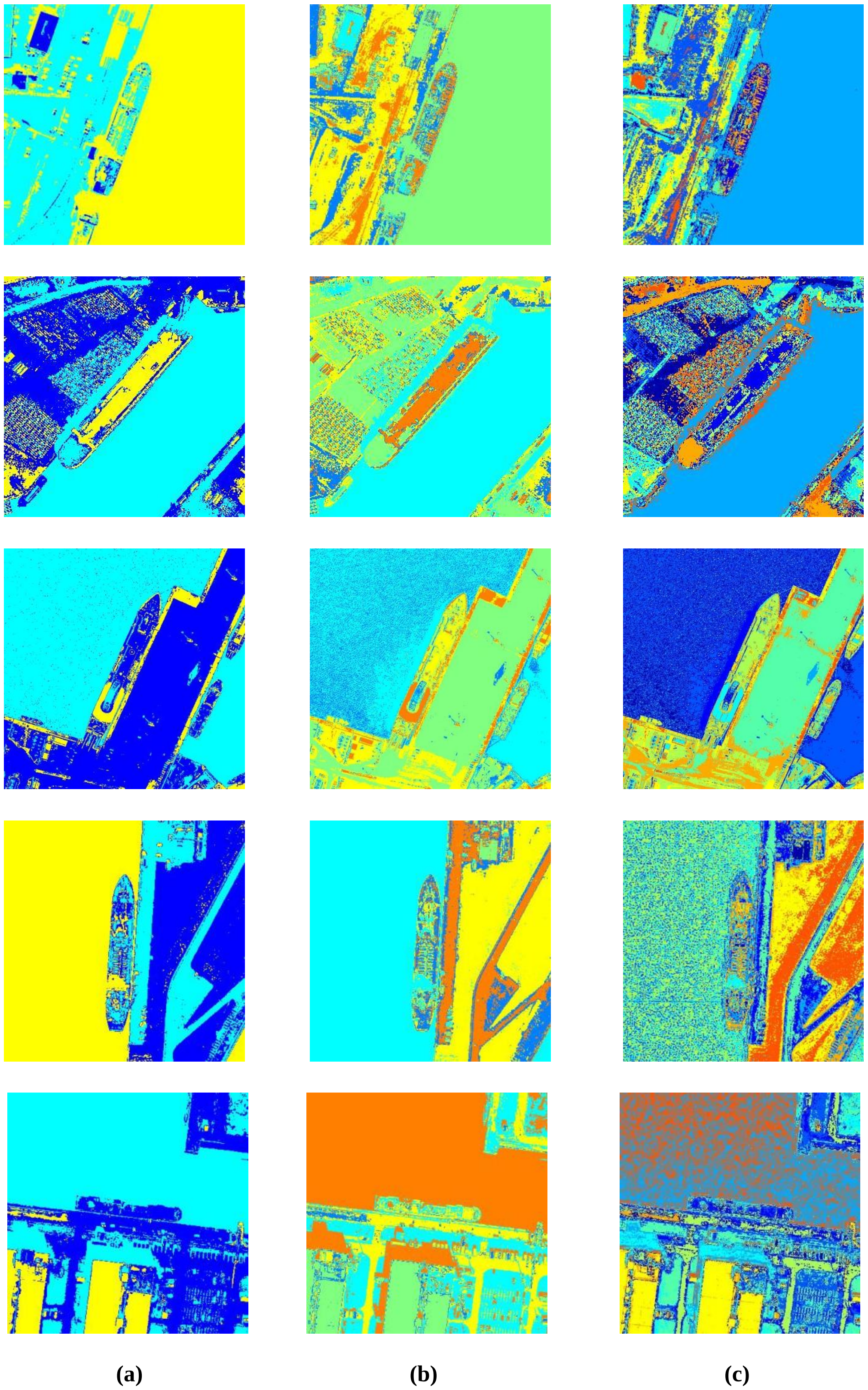


Figure 9: K-means segmentation results for ship images (a) k = 3 (b) k = 5 (c) k = 10

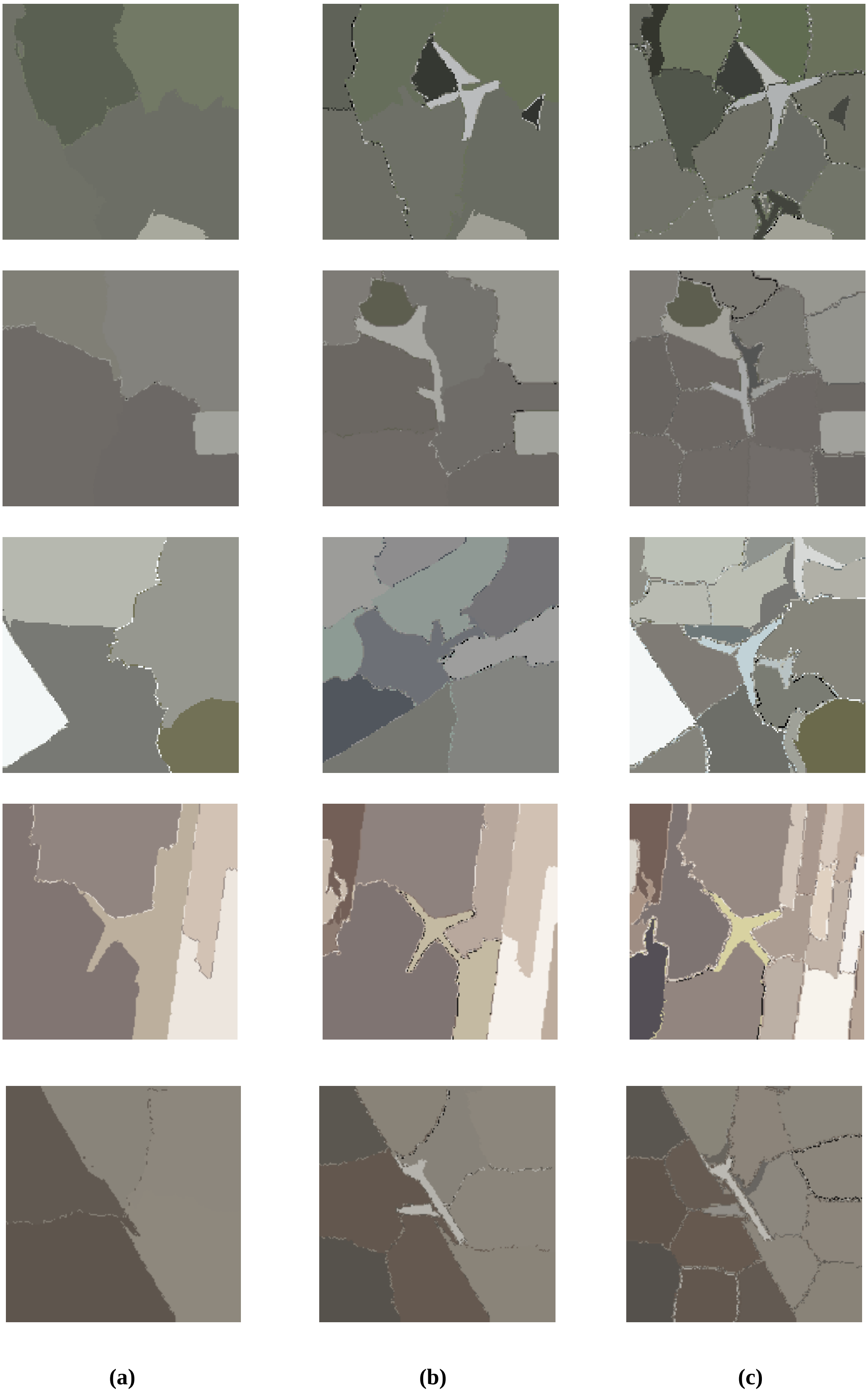


Figure 10 : Ncut segmentation results for plane images (a) N = 5 (b) N = 10 (c) N = 20

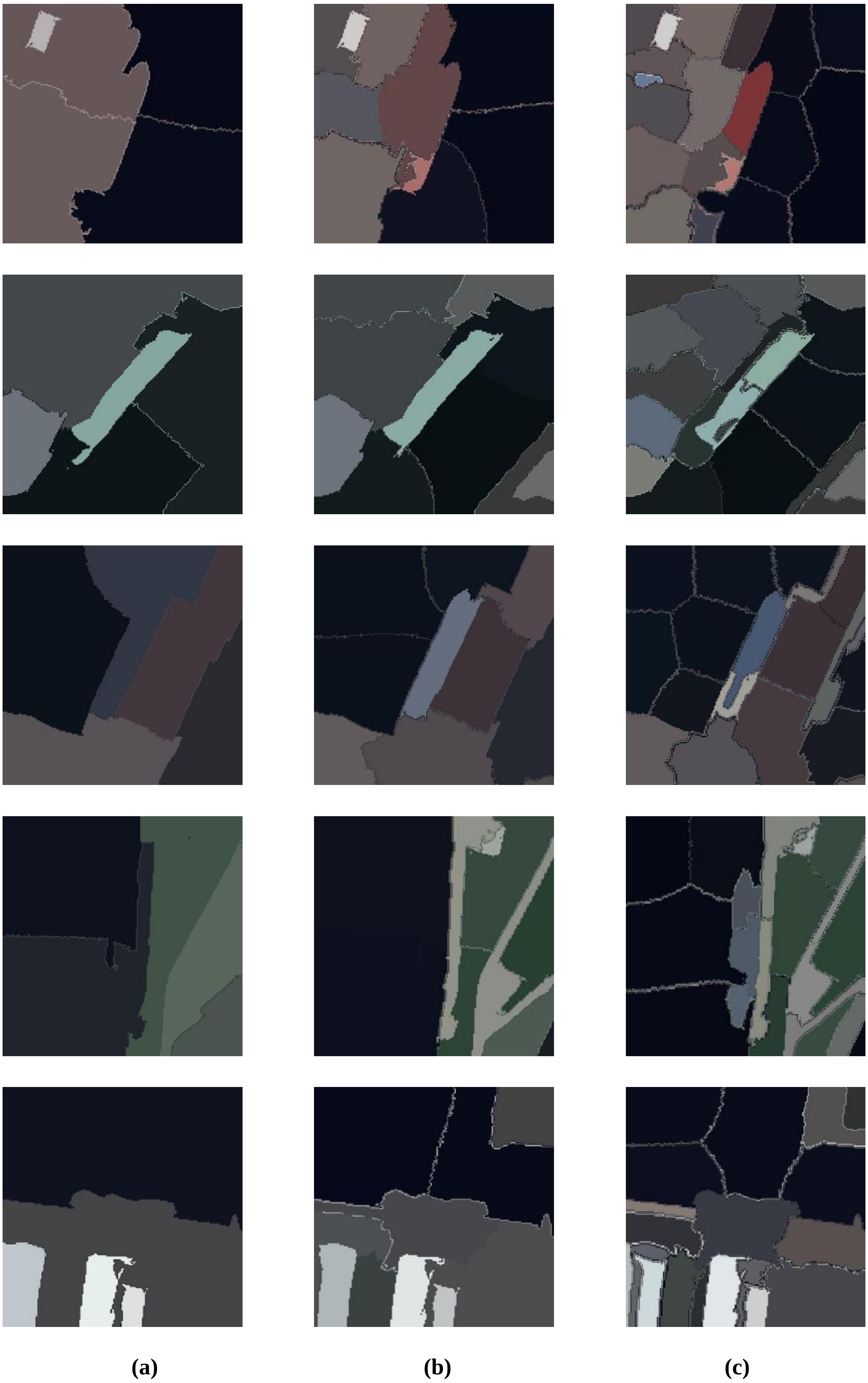
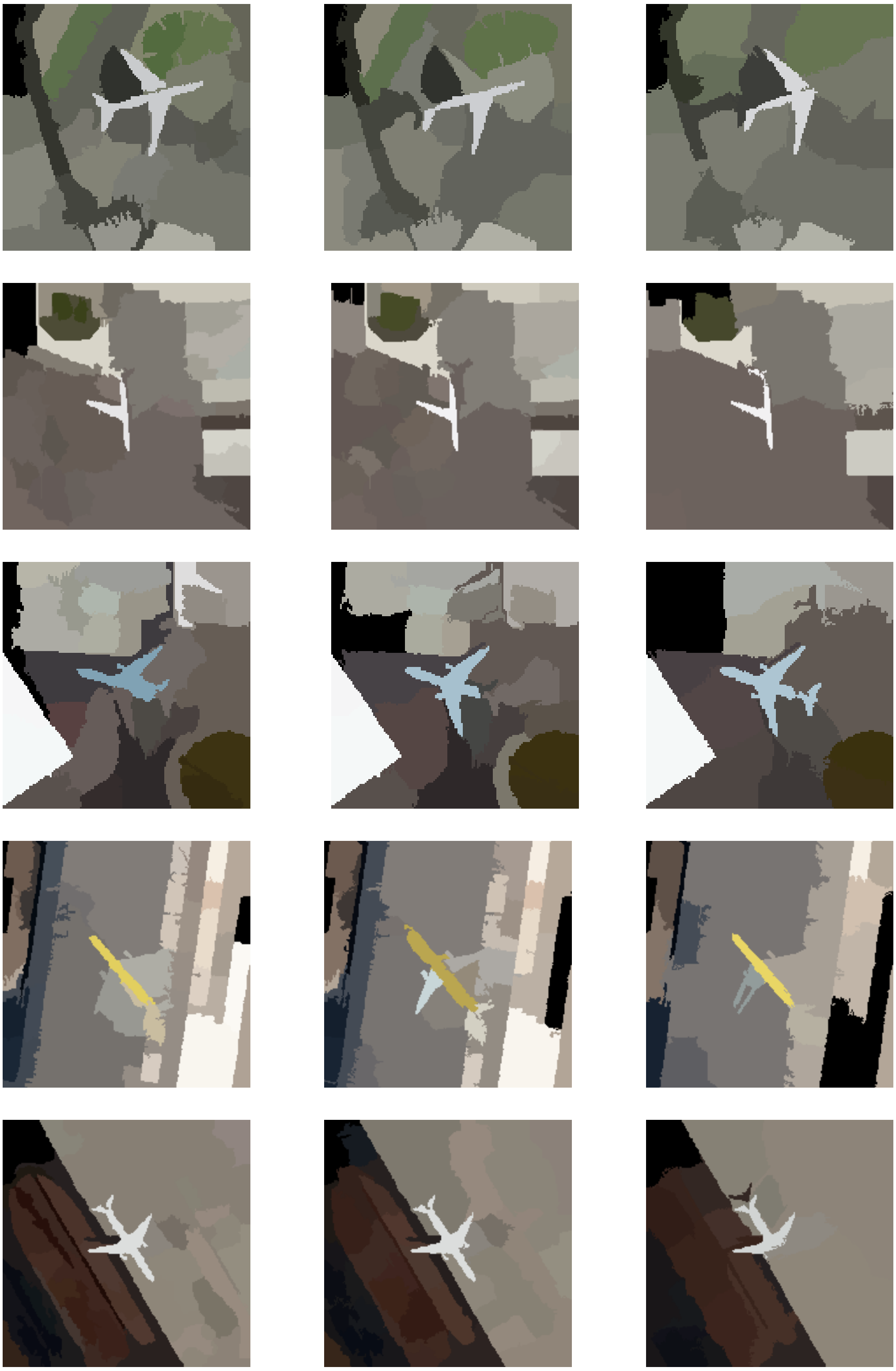


Figure 11: Ncut segmentation results for ship images (a) N = 5 (b) N = 10 (c) N = 20



(a) (b) (c)

Figure 12 : Mean-shift segmentation results for plane images (minimum segment area = 320)
(a) $h_s = 7$ and $h_r = 6.5$ (b) $h_s = 10$ and $h_r = 9.5$ (c) $h_s =17$ and $h_r = 16.5$

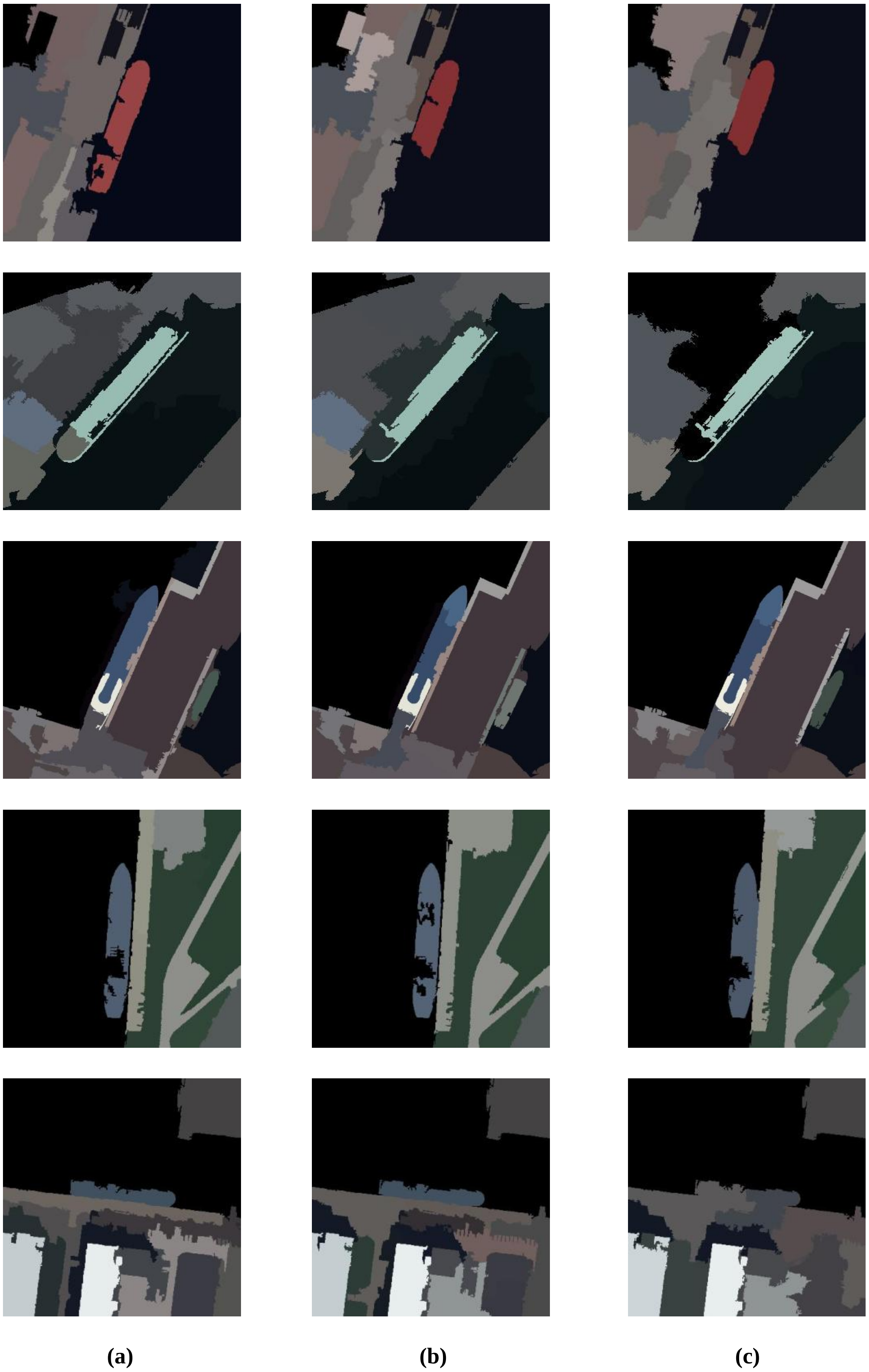


Figure 13: Mean-shift segmentation results for ship images (minimum segment area = 3020)
(a) h_s = 7 and h_r = 6.5 (b) h_s = 10 and h_r = 9.5 (c) h_s =17 and h_r = 16.5

Table 1: The precision-recall values of k-means segmentation results for plane images

Precision	k = 3	k = 5	k = 10
Plane 1	0,41	0,65	0,68
Plane 2	0,07	0,13	0,34
Plane 3	0,06	0,06	0,11
Plane 4	0,11	0,15	0,20
Plane 5	0,83	0,85	0,88
Average	0,30	0,37	0,44

Recall	k = 3	k = 5	k = 10
Plane 1	0,75	0,49	0,40
Plane 2	0,70	0,63	0,56
Plane 3	0,46	0,410	0,27
Plane 4	0,70	0,53	0,34
Plane 5	0,75	0,73	0,56
Average	0,67	0,56	0,43

Table 2: The precision-recall values of k-means segmentation results for ship images

Precision	k = 3	k = 5	k = 10
Ship 1	0,12	0,19	0,23
Ship 2	0,06	0,07	0,11
Ship 3	0,06	0,12	0,71
Ship 4	0,43	0,61	0,75
Ship 5	0,08	0,20	0,31
Average	0,15	0,24	0,42

Recall	k = 3	k = 5	k = 10
Ship 1	0,39	0,40	0,22
Ship 2	0,77	0,45	0,33
Ship 3	0,51	0,29	0,24
Ship 4	0,54	0,46	0,37
Ship 5	0,60	0,42	0,38
Average	0,56	0,40	0,31

Table 3: The precision-recall values of Ncut segmentation results for plane images

Precision	N = 5	N = 10	N = 20
Plane 1	0,06	0,10	0,87
Plane 2	0,02	0,19	0,64
Plane 3	0,09	0,13	0,87
Plane 4	0,18	0,33	0,82
Plane 5	0,11	0,73	0,72
Average	0,09	0,30	0,79

Recall	N = 5	N = 10	N = 20
Plane 1	0,39	0,40	0,47
Plane 2	0,47	0,38	0,42
Plane 3	0,62	0,64	0,52
Plane 4	0,86	0,73	0,63
Plane 5	0,69	0,78	0,74
Average	0,61	0,59	0,56

Table 4: The precision-recall values of Ncut segmentation results for ship images

Precision	N = 5	N = 10	N = 20
Ship 1	0,10	0,10	0,63
Ship 2	0,05	0,18	0,23
Ship 3	0,10	0,26	0,73
Ship 4	0,96	0,96	0,91
Ship 5	0,28	0,80	0,73
Average	0,30	0,46	0,64

Recall	N = 5	N = 10	N = 20
Ship 1	0,55	0,49	0,30
Ship 2	0,84	0,84	0,82
Ship 3	0,48	0,70	0,59
Ship 4	0,70	0,67	0,33
Ship 5	0,88	0,88	0,48
Average	0,69	0,72	0,50

Table 5: The precision-recall values of mean-shift segmentation results for plane images

Precision	hs = 7 hr = 6.5	hs = 10 hr = 9.5	hs = 17 hr = 16.5
Plane 1	0,87	0,99	0,99
Plane 2	0,92	0,98	0,93
Plane 3	0,83	0,94	0,96
Plane 4	0,94	0,71	0,96
Plane 5	0,94	0,94	0,91
Average	0,90	0,91	0,95

Recall	hs = 7 hr = 6.5	hs = 10 hr = 9.5	hs = 17 hr = 16.5
Plane 1	0,56	0,55	0,45
Plane 2	0,48	0,44	0,46
Plane 3	0,50	0,70	0,81
Plane 4	0,32	0,42	0,34
Plane 5	0,84	0,84	0,61
Average	0,54	0,59	0,53

Table 6: The precision-recall values of mean-shift segmentation results for ship images

Precision	hs = 7 hr = 6.5	hs = 10 hr = 9.5	hs = 17 hr = 16.5
Ship 1	0,97	0,96	0,91
Ship 2	0,94	0,93	0,34
Ship 3	0,91	0,86	0,85
Ship 4	0,99	0,99	0,99
Ship 5	0,97	0,99	0,99
Average	0,96	0,95	0,82

Recall	hs = 7 hr = 6.5	hs = 10 hr = 9.5	hs = 17 hr = 16.5
Ship 1	0,81	0,78	0,84
Ship 2	0,89	0,88	0,46
Ship 3	0,86	0,65	0,66
Ship 4	0,63	0,63	0,59
Ship 5	0,69	0,48	0,46
Average	0,78	0,69	0,60

2.2 Semi-automatic or Interactive Image Segmentations

Even though fully automated segmentation techniques are being continuously improved, there is still no automated image analysis technique applied in fully autonomous manner with guaranteed performance for the general case in the literature. Therefore, semi-automatic segmentation techniques or interactive segmentation techniques that allow solving moderate and hard segmentation tasks by modest effort on the part of the user are becoming more and more popular for last few decades [56], [57], [58], [59], [60], [61], [62], [63] [64]. The main goal of such segmentation techniques is to partition an image into two segments as *object* and *background* by employing the user defined inputs.

Graph cuts and deformable models are two of the most notable frameworks in semi-automatic image segmentation. Both of these approaches depend on energy minimization of certain objective function. For deformable models, the first algorithm, called as active contour models, is proposed by Kass et. al. [64], which delineates an object's outline from a 2D image. This framework tries to minimize the contour energy E defined as the sum of external energy and internal energy. The external energy pulls the contours towards desired image features such as edges whereas the internal energy helps achieve smooth boundaries. The active contours, also called as snakes, are based on deforming an initial contour at a number of control points selected along a given initial contour. This approach has several drawbacks. First of all, the minimized energy function in this approach depends on only boundary properties of the given image and it does not regard about region properties or coherence. Therefore, the segmentation capabilities dramatically decreases when there are no strong edges on the desired object boundary. Second, the snakes move toward the nearest local minimum of the initial contour, and hence, it has a tendency to find a local minimum which in general does not coincide with the object contour. This leads to sensitivity to initialization of the user defined contour, when there exist a large number of local minima near the user defined contour due to image noise or background clutter typically encountered in satellite imagery. Third, the algorithm highly depends on its parameters and so automatic selection of various parameters such as the weights in the energy function is still an open problem. Finally, the discretization of the contours into a number of control points may cause problems with uneven pacing and self crossing while the contours are deforming, and make it difficult to extract the complex shaped objects. Therefore, these limitations of the active contour models prevent to achieve robust and accurate segmentation results in the general case.

On the other hand, graph cuts based approaches have the ability to jump over local minima, and hence provide a more global and robust solution. Furthermore, the optimization process in graph cuts runs faster than the deformable models, and the exact solution can be found in polynomial time. As a result, graph cuts based interactive foreground extraction algorithms are investigated in this thesis.

In this work, three popular interactive image segmentation algorithms are examined. These are interactive graph cut [60], interactive GrabCut [63] and interactive GrowCut [62] image segmentation algorithms. After introducing the algorithmic detail in the following three sub-sections, the experimental results are given in the final part in order to compare the algorithms qualitatively and quantitatively.

2.2.1 Interactive Graph Cuts Image Segmentation

The interactive graph cut image segmentation algorithm is first described by Boykov and Jolly in 2001 [60]. The main issue with this algorithm is that a user imposes certain hard constraints which are relevant to the user’s segmentation purpose, and then the whole image is automatically divided into segments based on these constraints. This aim can be achieved by firstly marking certain pixels (seeds), which undoubtedly have to be part of the object and certain pixels which have to be part of the background. In other words, these hard constraints should provide clues on the objective of the segmentation of the user. The remaining pixels in the image are clustered automatically by obtaining a global optimum among all segmentations satisfying the hard constraints imposed by a user. For this purpose, a cost function, which is described in terms of the boundary and region properties of the resulted segments, is introduced. These properties can also be considered as soft constraints for the segmentation.

The cost function used as soft constraints for the segmentation should be comprehensive enough to incorporate both region and boundary information of segments. In the light of the abovementioned requirement, the cost function obtained in the context of MAP-MRF estimation can be defined as follows:

Consider an arbitrary set of data elements (pixels in an image) $\mathcal{P}$ and some neighborhood system represented by a set $\mathcal{N}$ all unordered pairs $\{p,\ q\}$ of neighboring elements in $\mathcal{P}$. Let $A = (A_{1},\ \ldots,\ A_{p},\ \ldots,\ A_{|P|})$ be a binary vector whose elements A_p identifies assignments to pixels p in $\mathcal{P}$ such that each A_p can be either “obj” or “bkg” which are the abbreviation of “object” and “background”, respectively. Hence, the binary vector A defines a kind of segmentation. Then, the cost function $E(A)$ which defines the soft constraint imposed on the boundary and region properties of A is:

$$E(A) = \lambda \,.\, R(A) + \ B(A) \tag{2.19}$$

where

$$R(A) = \sum_{p \epsilon P} R_p \left(A_p\right) \tag{2.20}$$

$$B(A) = \sum_{\{p,q\} \epsilon N} B_{\{p,q\}} \,.\, \delta(A_p, A_q) \tag{2.21}$$

and

$$\delta(A_p, A_q) = \begin{cases} 1, & if\ A_p \neq A_q \\ 0, & otherwise \end{cases} \tag{2.22}$$

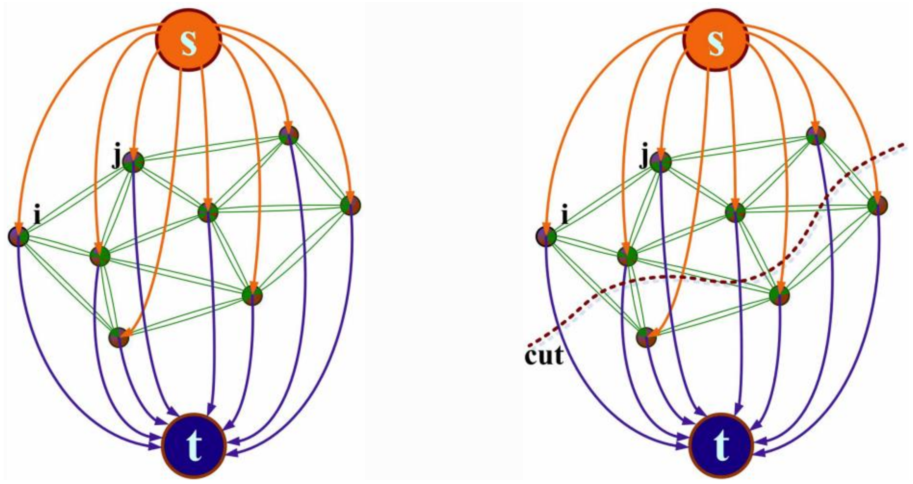
This cost function is the energy function utilized for the segmentation. The coefficient $\lambda \geq 0$ in (2.19) identifies a compromise between the relative significance for the region properties term $R(A)$ and the boundary properties term $B(A)$. The regional term $R(A)$ is the penalties for assigning the pixels *p* into *the object* or *background* and the individual penalties are described with the term $R_p("obj")$ and $R_p("bkg")$ for object and background, respectively. The term $B(A)$ expresses the boundary or discontinuity information of the segmentation *A*. The coefficient $B_{\{p,q\}} \geq 0$ indicates the penalty for a discontinuity between *p* and *q*. Generally, $B_{\{p,q\}}$ is large when pixels p and q are similar in terms of related feature(s) and it is close to zero as the two pixels are becoming more and more different. These features that can be used for this purpose might be any function of distance between two pixels, color and texture information, local intensity gradient, Laplacian zero-crossing, gradient direction, etc.

There are many different types of hard constraints proposed in the literature for interactive segmentation algorithms. Some of them are based on labeling of certain pixels in the desired segmentation regions and the other ones are based on indicating certain pixels on the segmentation boundary pixels. The hard constraints used in the interactive graph cut algorithm are based on labeling the segmented regions rather than boundary pixels such that some pixels are marked as internal (*object* seeds) and some as external (*background* seeds) for a given object of interest. The marked pixels can be anywhere in the related regions, since the algorithm yields the similar outputs irrespective of particular seeds positioning within the same image object.

The main idea of the algorithm is to incorporate the soft constraints encoded by (2.19) with user defined hard constraints in order to achieve perfect segmentation results. Before the detailed explanation of how it can be achieved, the basic terminology that is related to graph cuts in the context of this segmentation should be clarified. An undirected graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is defined as a set of nodes or vertices $\mathcal{V}$ and a set of undirected edges $\mathcal{E}$ that join these nodes. An example of a graph used in this context is illustrated in the Figure 14.a. The nodes of the graph can represent pixels of an image and the edges can represent any neighborhood relationship between the pixels.

There are also two additional special nodes which are called as terminals: an object terminal (a source *S*) and background terminal (a sink *T*). These nodes are connected like a grid by edges. These sets of edges consist of two types of undirected edges (or links): *n-links* which are also called as neighborhood links and *t-links* which are also called as terminal links. Terminal links (*t-links*), which are denoted as $\{p, S\}$ and $\{p, T\}$, connect each pixel *p* into the terminal nodes *S* and *T*. Similarly, each pair of neighboring pixels is connected by a neighborhood link (*n-links*) and denoted as $\{p, q\}$. Each edge $e \epsilon \mathcal{E}$ in the graph is assigned a nonnegative weight (cost) ω_e. A *s/t* cut on a graph with two terminals is a partition of vertices in the graph into two disjoint subsets such that the terminals become separated on the induced graph. Therefore, any cut will correspond to a binary segmentation. Figure 14.b illustrates one example of a cut. The severed n-links and t-links constitute a *s/t* cut-set $C \subset \mathcal{E}$ and so the cost of cut in the combinatorial optimization is defined as the sum of the weight of the edges in this cut-set C.

$$|C| = \sum_{e \in C} \omega_e \tag{2.23}$$



(a) (b)

Figure 14: (a) A graph with two terminals (b) A cut on the graph [65]

After defining the basic terminology regarding of the graph cuts, the algorithmic details about interactive graph cuts segmentation can be explained as follows:

Assume that $\mathcal{O}$ and $\mathcal{B}$ denote the pixels marked as object and background seeds, respectively. Moreover, they satisfy the conditions that $\mathcal{O} \subset \mathcal{P},\ \mathcal{B} \subset \mathcal{P}\ and\ \mathcal{O} \cap \mathcal{B} = \emptyset$. Then, the hard constraints are defined mathematically as follows:

$$\forall p\ \in \mathcal{O}\,, \qquad A_p = \text{"obj"} \tag{2.24}$$

$$\forall p\ \in \mathcal{B}\,, \qquad A_p = \text{"bkg"} \tag{2.25}$$

After imposing the hard constraints, the graph with two terminals $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is constructed from a given digital image such that the nodes of the graph except the source and sink terminals correspond to pixels $\mathcal{P}$ of the image and these nodes are connected by weighted edges via *n*-links and *t*-links. Therefore, the total vertices in the graph become $\mathcal{V} = \mathcal{P} \cup \{S, T\}$ and all edges become $\mathcal{E} = \mathcal{N}\ \cup_{p \in P} \big\{\{p, S\}, \{p, T\}\big\}$. The weight (cost) of each edge in $\mathcal{E}$ is introduced in Table 7.

Table 7: The weights of links

Edge	Weight (cost)	For
$\{p, q\}$	$B_{\{p,q\}}$	$\{p, q\} \in \mathcal{N}$
$\{p, S\}$	$\lambda \,.\, R_p(\text{"}bkg\text{"})$	$p \ \in \mathcal{P}, \ p \notin \mathcal{O} \cup \mathcal{B}$
	K	$p \ \in \mathcal{O}$
	0	$p \ \in \mathcal{B}$
$\{p, T\}$	$\lambda \,.\, R_p(\text{"}obj\text{"})$	$p \ \in \mathcal{P}, \ p \notin \mathcal{O} \cup \mathcal{B}$
	0	$p \ \in \mathcal{O}$
	K	$p \ \in \mathcal{B}$

where

$$K = 1 + \max_{p \in \mathcal{P}} \sum_{q:\{p,q\} \in \mathcal{N}} B_{\{p,q\}} \tag{2.26}$$

The weight of t-links that connect pixels labeled into the object or background should theoretically be infinite. However, parameter K which has very high value is used instead of the infinite in the calculation.

The computation of the weights of the edges plays a crucial role in the segmentation performance, since the value of weight can integrate all aspects of feature information, such as gray-level, color, and texture of images, into the segmentation. Consequently, the values of the edges are calculated based on the similarity, continuity and proximity of the two nodes, and reflect the likelihood that two pixels belong to the same segment.

The weight of *n*-link expresses the discontinuity of corresponding pixels p and q, and $B_{\{p,q\}}$ can be described as follows:

$$B_{\{p,q\}} \propto \exp\left(-\frac{\left(I_p - I_q\right)^2}{2\sigma^2}\right) . \frac{1}{dist(p,q)} \tag{2.27}$$

This function increasingly penalizes the discontinuities when two neighboring pixels are becoming more and more similar (i.e. $\left|I_p - I_q\right| < \sigma$). However, when pixels are remarkably different, $\left|I_p - I_q\right| > \sigma$, the penalty decreases.

The weight of the t-links is denoted by $R_p(\cdot)$ which reflects on how the intensity of pixels p fits into a known intensity model (or histogram) of the object and background. The intensity models (or histograms) of the object and background can be estimated by using the intensity value of the pixels marked as seeds. Then, these intensity models are employed for computing the regional penalties $R_p(\cdot)$ as negative log-likelihoods:

$$R_p("obj") = -\ln \Pr\left(I_p|\mathcal{O}\right) \tag{2.28}$$

$$R_p("bkg") = -\ln \Pr\left(I_p|\mathcal{B}\right) \tag{2.29}$$

After the graph is completely defined, the segmentation boundary can be constructed between the object and the background by finding the global minimum cost cut $\hat{C}$ on the graph $\mathcal{G}$. The globally minimum cost cut $\hat{C}$ on graph $\mathcal{G}$ can be calculated exactly in low-order polynomial time with the help of the optimization algorithms used for cutting the graph with two terminals assuming that the edge weights of the graph are non-negative. A version of "min-cut/max-flow" algorithm reported in [66] is utilized for the implementation of combinatorial optimization algorithm. Consequently, a cut with minimum cost gives the best segmentation of the image with the region and boundary properties satisfying the user defined hard constraints. The proof of minimum cut can be reached in Appendix B.

As a summary, implementation of the algorithm can be expressed in the following statements:

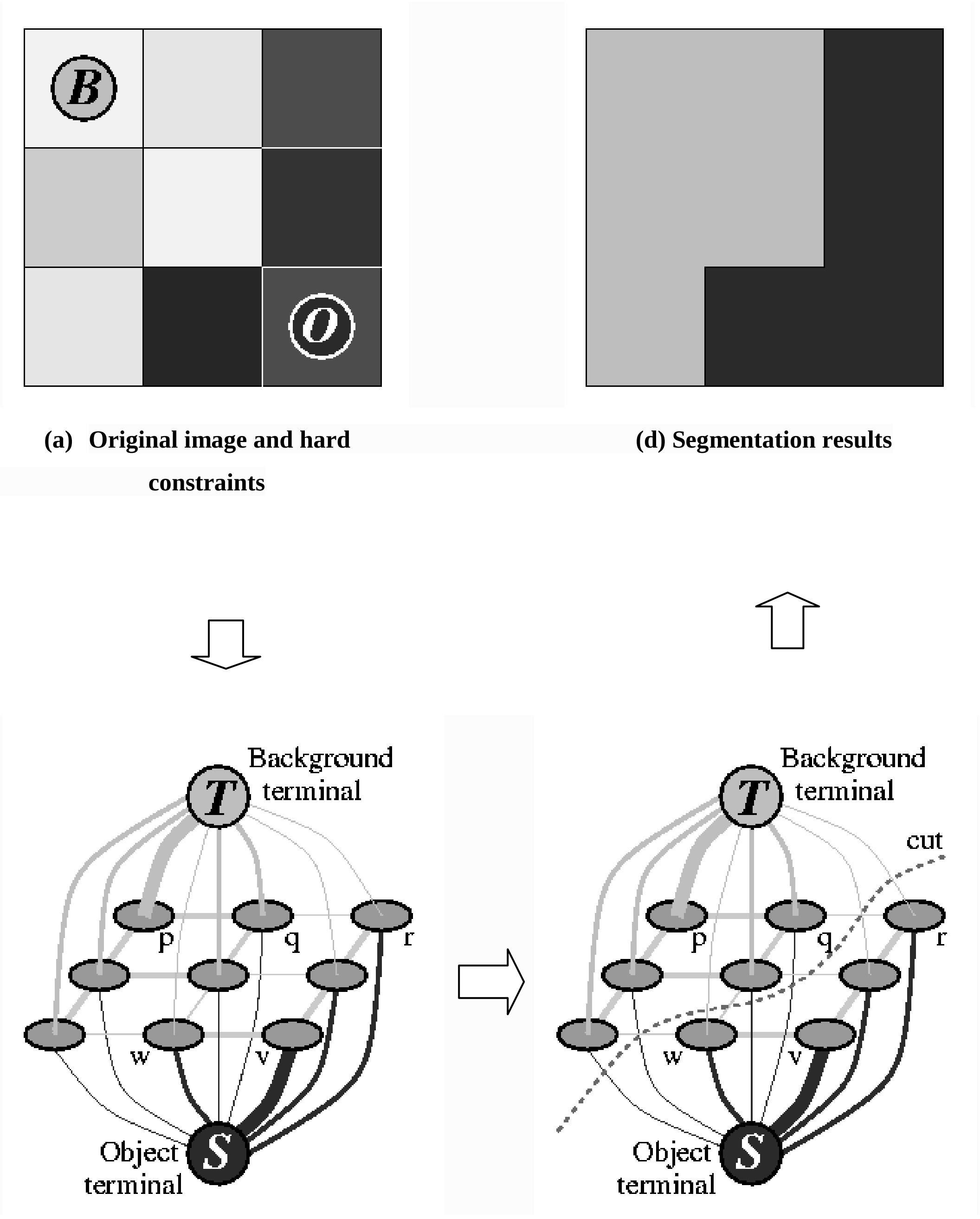
1. Impose the hard constraints by labeling certain pixels that have to be part of the object as object seeds and certain pixels which have to be part of background as background seeds.

2. Establish the cost function (energy function) that describes the region and boundary properties of the resulted segments.

3. Construct the graph with two terminals from the target image and assign a particular weight into each edge in the graph. The value of the weights should reflect the similarity, proximity and continuity of two nodes in the graph.

4. Finally, compute the minimum cost cut in the graph via "min-cut/max-flow" combinatorial optimization algorithm in order to get optimal segmentation satisfying soft and hard constraints.

In order to make the process more comprehensive, the general workflow of the algorithm is illustrated by an example in Figure 15. A graph with two terminals in Figure 15.b is built from a 3 x 3 synthetic image in Figure 15.a such that the cost of edges is defined in terms of the parameters in the region and the boundary properties (soft constraints) and hard constraints. The next step is to determine the globally optimum minimum cut separating two terminals, source and sink as shown in Figure 15.c. Finally, this *s/t* cut gives the best segmentation result of the original image in Figure 15.d.



(a) Original image and hard constraints

(d) Segmentation results

(a) Graph with two terminals

(c) A s/t cut

Figure 15 : A 3 x 3 synthetic image segmentation example. Seeds are $\mathcal{O}$ and $\mathcal{B}$. The weight value of each edge is illustrated by the edge's thickness. Low-priced edges are selected for the minimum cost cut [60].

2.2.2 *Interactive GrabCut Image Segmentation*

Interactive GrabCut is a powerful and innovative 2D image segmentation algorithm which is described by Rother et al. in 2004 [63]. This method aims to solve the problem of efficient, interactive extraction of a foreground object in a complex environment whose background cannot be trivially subtracted with high performance (i.e. accurate segmentation of object from background) at the cost of only modest interactive effort for the user.

GrabCut algorithm [63] relies on interactive graph cut algorithm which successfully combines both texture (color) and contrast (edge) information and it extends the graph cut algorithm in three regards. First of all, a Gaussian Mixture Model (GMM) is used for color data modeling instead of the monochrome image model such as histograms. Secondly, a more powerful, iterative procedure for the optimization process is developed rather than one-shot optimization. Thirdly, the requirements on the interactive user inputs are relaxed by allowing incomplete labeling as a result of the iterative cut estimation.

Since it is impractical to build satisfactory color space histograms, the Gaussian mixture model (GMM) is utilized for data modeling. Therefore, two GMMs, one for the background and one for the foreground, are created to be a full-covariance Gaussian mixture with K components (typically $K = 5$). For the construction of GMMs, both regions (the background and the foreground) are firstly divided into K pixel clusters via one of the clustering algorithms. Then, the Gaussian components are initialized from the colors in each cluster by calculating the parameters of the each component. Mathematically, GMM can be expressed by the equation,

$$p(z|\bar{\theta}) = \sum_{i=1}^{K} \omega_i \quad g(z|\mu_i, \Sigma_i) \tag{2.30}$$

where z is a d-dimensional continuous-valued data vector (RGB color space in this case), $\omega_i,\ i = 1, \dots, K$ are the mixture weights, and $g(z|\mu_i, \Sigma_i),\ i = 1, \dots, K$ are the component Gaussian densities. Each component density is a d-variate Gaussian function of the form,

$$g(z|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(z-\mu_i)^t {\Sigma_i}^{-1}(z-\mu_i)\right\} \qquad (2.31)$$

with mean vector μ_i and covariance matrix Σ_i. The mixture weights should satisfy the constraint that $\sum_{i=1}^{K} \omega_i = 1$. Thus, the complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation,

$$\bar{\theta} = \{\omega_i,\ \mu_i,\ \Sigma_i\}, \qquad i = 1, \dots, K \qquad (2.32)$$

Now, the cost function or Gibbs energy function for segmentation becomes

$$E(\bar{\alpha}, \boldsymbol{k}, \bar{\theta}, \boldsymbol{z}) = U(\bar{\alpha}, \boldsymbol{k}, \bar{\theta}, \boldsymbol{z}) + V(\bar{\alpha}, \boldsymbol{z}) \qquad (2.32)$$

where $\boldsymbol{z}$ = $(z_1, \dots, z_{n,} \dots, z_N)$ is an array of RGB color values of image pixels, indexed by index n and $\bar{\alpha} = (\alpha_1, \dots, \alpha_{n,} \dots, \alpha_N)$ is an array which expresses the segmentation of the image at each pixel by assigning $\alpha_n = 0$ for background and $\alpha_n = 1$ for foreground. The parameter $\bar{\theta}$ describes the GMM parameters defined by the expression (2.32) and $\boldsymbol{k}$ = $(k_1, \dots, k_{n,} \dots, k_N)$ is the GMM component variables. It is defined in order to deal with the GMM easily, in optimization viewpoint, by assigning each pixel into unique GMM component with $k_n \in \{1, \dots, K\}$ according to $\alpha_n = 0$ or 1. The data term U is now described as,

$$U(\bar{\alpha}, \boldsymbol{k}, \bar{\theta}, \boldsymbol{z}) = \sum_{n} D(\alpha_n, k_n, \bar{\theta}, z_n) \tag{2.33}$$

where

$$\begin{aligned} D(\alpha_n, k_n, \bar{\theta}, z_n) \; &= -\ln \omega(\alpha_n, k_n) + \frac{1}{2}\ln|\Sigma(\alpha_n, k_n)| \\ &+ \tfrac{1}{2}[z_n - \mu(\alpha_n, k_n)]^T \Sigma(\alpha_n, k_n)^{-1}[z_n - \mu(\alpha_n, k_n)] \end{aligned} \tag{2.34}$$

The smoothness term V is basically same for both the graph cut and GrabCut algorithms [63] except that the contrast term is calculated in color space by using the Euclidian distance as follows:

$$V(\bar{\alpha}, \boldsymbol{z}) = \gamma \sum_{(m,n)\in\mathcal{N}} dis(m,n)^{-1}[a_m \neq a_n]\text{exp } {-\beta}\|z_m - z_n\|^2 \tag{2.35}$$

The energy minimization procedure in GrabCut algorithm iteratively works by alternating between cut estimation and GMM parameter learning. While the algorithm is working iteratively, the segmentation result, $\overline{\alpha}$, changes at each iteration by displacing some pixels from the background to the foreground, and vice versa. Therefore, the GMMs should be updated in order to reflect the new foreground and background color distributions after the cut estimation. This can be achieved by running the clustering algorithm used for the initialization again, but most of the clustering methods are quite slow. Therefore, GrabCut algorithm employs an incremental clustering update to speed up the algorithm, and this is implemented in two steps. First, each pixel in the image is assigned into the unique GMM component which has the highest likelihood of producing the color of pixels. This is realized by simply evaluating the Gaussian equation with the color of pixel as input. Hence, the segmentation result, $\bar{a}$, and the component index, $\boldsymbol{k}$, uniquely identifies all of the $2K$ components. Next, once the pixels have been clustered, the current GMMs are thrown away and the new Gaussian components are generated by computing their parameters. The mean $\mu(\alpha, k)$ and the covariance $\Sigma(\alpha, k)$ are estimated as the sample mean and the sample covariance of pixels in each Gaussian component. The mixture weights $\omega(\alpha, k)$ are computed as the number of pixels in the Gaussian component divided by the number of all pixels in the GMM. After the parameter learning is carried out, the global optimization algorithm, the graph cut, is run to estimate the new segmentation result.

A graph should be constructed in order to obtain a global solution by means of the optimization algorithm. The constructed graph in this case is identical to the graph described by the graph cut algorithm except the weights of T-links. The probabilities acquired from GMMs are utilized for unknown pixels T-link weight. The weights of T-links for pixel m are stated in Table 8.

Table 8: The weights of links for GrabCut algorithm

Pixel type	Background T-link	Foreground T-link
$\boldsymbol{m \in Foregound}$	*0*	$L(m)$
$\boldsymbol{m \in Background}$	$L(m)$	*0*
$\boldsymbol{m \in Unknown}$	$D_{fore}(m)$	$D_{back}(m)$

In order to enforce pixel $\boldsymbol{m}$ be a member of either the background or foreground the following constraint must be satisfied:

$$L(m) > \sum_{(n,m) \in \mathcal{N}} N(n, m) \tag{2.36}$$

$D_{fore}(m)$ and $D_{back}(m)$ are the function of the likelihood that the pixel m belongs to the foreground and background GMMs, respectively. They are calculated for pixel m with the background and foreground GMMs as follows:

$$D(m) = -\log \sum_{i=1}^{K} \omega_i \frac{1}{\sqrt{|\Sigma_i|}} exp\left\{-\frac{1}{2}[z_m - \mu_i]^T \Sigma_i^{-1} [z_m - \mu_i]\right\} \tag{2.37}$$

Moreover, the values of T-link weights must be updated during the iteration due to the variation of the GMMs. On the other hand, the N-link weights are constant throughout the execution of GrabCut algorithm. Therefore, they can be calculated once and reused later.

The degrees of interactive effort generally range from editing individual pixels, at intensive workload, to only touching the foreground and/or background in a few locations. In GrabCut algorithm, the requests on interactive user for a given quality of the result are considerably reduced by allowing incomplete labeling as a result of iterative estimation. Incomplete labeling means that the user should merely specify the background region without any hard foreground labeling. This can be done simply by placing a rectangle or a lasso around the desired object, and so the strip of the pixels around the outside of the marked rectangle is selected as the background. Iterative cut estimation overcomes this incompleteness by allowing temporary labels (as the foreground) on the pixels in the marked rectangle which can subsequently be retrieved.

Finally, the summary of the algorithm can be stated as follows [67]:

1- The user specifies the hard constraints by dragging a rectangle around the desired object. Pixels inside the rectangle are selected as unknown. Pixels outside the rectangle are marked as known background.

2- Initially, all unknown pixels are provisionally placed in foreground class, and all known background pixels are placed in the background class.

3- Gaussian Mixture Models (GMMs) are constituted for initial foreground and background classes

4- Each pixel in the foreground class is assigned to the most likely Gaussian component in the foreground GMM. Likewise, each pixel in the background is assigned to most likely background Gaussian component.

5- The initial GMMs are ruled out, and new GMMs are constructed by estimating the Gaussian parameters from the pixel sets created previous set.

6- A graph is built with respect to the created GMMs, and the graph cut algorithm is run to find a new candidate foreground and background classification results.

7- Steps 4-6 are repeated until the classification converges to the desired segmentation output.

2.2.3 Interactive GrowCut Image Segmentation

GrowCut is an algorithm for interactive multi-label segmentation of N-dimensional images which is proposed by Vezhnevets and Konouchine in 2005 [62]. The main idea in the algorithm is that once a small number of user-labeled pixels are given, the rest of the image is segmented automatically by a cellular automaton in an iterative manner. The task statement and user input data of the method are similar to the graph cut algorithm, but the segmentation equipment differs. This method uses the cellular automata for solving the pixel labeling task rather than graph cut and is also iterative process, which can give feedback to the user while the process is computed.

The cellular automata were described to model a wide variety of dynamical systems in different applications by Ulam and von Neumann in 1966 [68]. It is usually discrete in both space and time and so operates on a lattice of spots $p \in P \subseteq Z^n$ (pixels in image processing). A bi-directional and deterministic cellular automaton is a triplet such that $A = (S, N, \delta)$, where S is an non-empty state set, N is the neighborhood system and $\delta : S^N \longrightarrow S$ is the local transition function which specifies the rule of calculating the state of cell at t+1 time step, given the states of neighboring cells at previous time step t. 4-neighborhood or 8-neighborhood relationship is the commonly used neighboring systems in this algorithm. The cell state S_p in this case is defined as a triplet (l_p, θ_p, C_p) in which identify the label, the strength and feature vector of the cell p, respectively. It is assumed that $\theta_p \in [0, 1]$.

An image is a two-dimensional array of $k\ x\ m$ pixels. Then, the unlabeled image may be regarded as a specific configuration state of a cellular automaton P and initial states for $\forall p \in P$ are set to:

$$l_p = 0,\ \ \theta_p = 0,\ \ C_p = RGB_p \qquad (2.38)$$

where RGB_p is the three dimensional vector of pixel p in RGB space. When the inputs are specified by a user, the seeded cells are labeled accordingly, and their strengths are set to the seed strength value defined by the user. After setting the initial state of the cellular automaton, the cell labels ${l_p}^{t+1}$ and strengths ${\theta_p}^{t+1}$ are updated at iteration $t + 1$ in the following automata evolution rule.

1. Initially, copy the state values (label and its strength) at time *t* into ones at time *t+1* for each pixel.

2. Then, the attack force is defined by the attacker cell's strength θ_q and the distance between the feature vectors of the attacker C_q and defender C_p and computed with mathematical expression in (2.39).

$$F_q = g\left(\left\|C_p - C_q\right\|_2\right).\theta_q^t \tag{2.39}$$

   where $g(.)$ is a monotonous decreasing function bounded to [0,1] and is defined,

$$g(x) = 1 - \frac{x}{\max\|C\|_2} \tag{2.40}$$

3. Next, the current cell comes under attack by all of its neighbors according to neighborhood system. If attack force at time *t* is greater than the strength of the defender (current cell) at time *t*, the defending cell is conquered, and its label and strength should be updated at time *t*+1.

4. The update rules of labels and strengths are

$$l_p{}^{t+1} = l_p{}^{t} \tag{2.41}$$

$$\theta_p{}^{t+1} = g\left(\left\|C_p - C_q\right\|_2\right).\theta_q^t \tag{2.42}$$

5. The process continues until automaton converges to stable image configuration, which means that the cell states hold to change.

In this algorithm, the seeds defined by user interaction do not necessarily specify hard constraints like the methods based on the graph cut. In other words, the user interaction does not need to identify the regions of firm foreground and/or firm background. Hence, this gives more versatile control of the segmentation from the user and makes the process tolerable to incorrect user inputs. Such seeds can be described with the help of the seed strength notion. If it is desired to characterize the seed as the hard constraint, which does not allow changing its label during the evolution, it is easily achieved by setting seed cell's strength to one. However, for soft constraint, initial strength values of the seed is set to smaller than one which allows increase or decrease the current cell strength by some value.

2.2.4 Experimental Results

In this sub-section, the performances of the interactive or semi-automatic segmentation algorithms introduced in this thesis are examined on test images selected for evaluating the fully automatic segmentation algorithms. These images can be observed from Figure 6 and Figure 7 with their corresponding ground truth.

Qualitative experimental results for aircraft and ship images are represented in Figure 16-21. Graph cut segmentation algorithm results with their corresponding user interaction points are displayed in Figure 16 and Figure 17. Similarly, Figure 18 and Figure 19 exhibit the result of GrowCut segmentation algorithm with their user inputs. In order to conduct a fair comparison between these two algorithms, the number and position of the user interaction points in the foreground and the background are tried to be selected similarly. The red dots represent the foreground pixels and the blue ones display the background ones in these cases. Finally, the results of GrabCut segmentation method and their related interactions are presented in Figure 20 and Figure 21. Unlike the previous two methods, the user interaction in this algorithm is only the dragging a rectangle around the desired object without specifying any foreground pixels.

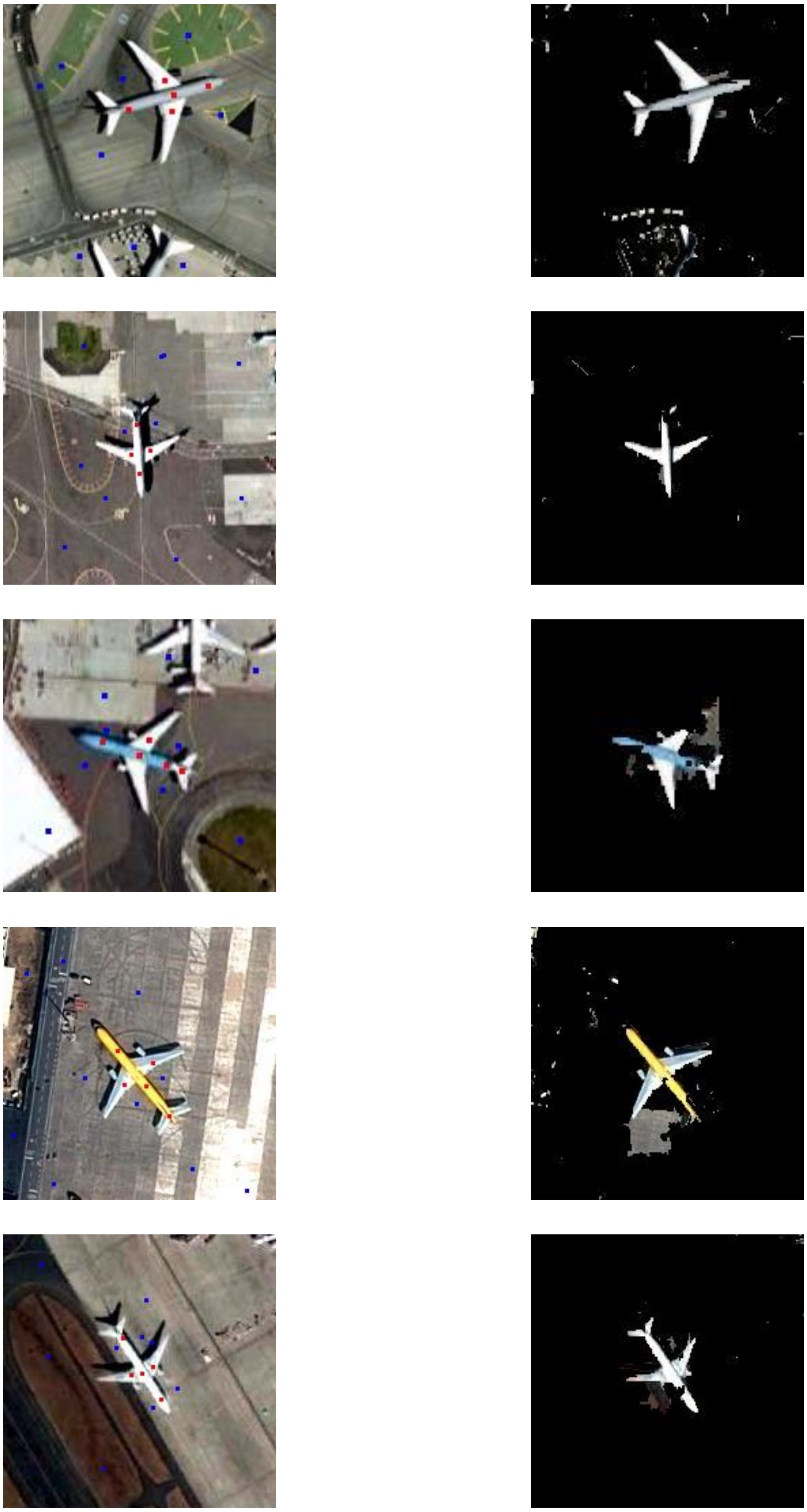
As it can be observed from the results, the performance of the graph cut method is the poorest approach among the interactive segmentation algorithms and even compared to some fully automatic ones such as the mean shift method. This is due to the used color data modeling in the algorithm. To calculate the regional penalty values in the method, the histograms of the seeds entered by the user are utilized in the background and the foreground modeling instead of GMM. Owing to the insufficient modeling of the histograms, the necessary and adequate information cannot be obtained to get the desired segmentation outputs. Furthermore, the effort on the user to select the seed regions for the object and the background is quite intense, and this makes difficult to utilize the graph cut algorithm for automatic high level applications. Lastly, the implementation of Boykov and Kolmogorov [69] is used for solving the graph cut problem in this study.

GrowCut method outperforms the graph cut algorithm and segments the test images similar to the ground truth segmentation. However, the performance of the algorithm in terms of the user interaction effort is quite inadequate as in the graph cut. Therefore, a large number of seeds should be selected by the user in order to obtain the expected results.

The final algorithm, GrabCut, outperforms the graph cut in terms of both the quality of results and the required user efforts. It also gives comparable results with respect to GrowCut algorithm in point of the quality of results whereas the required user effort is at the minimal level.

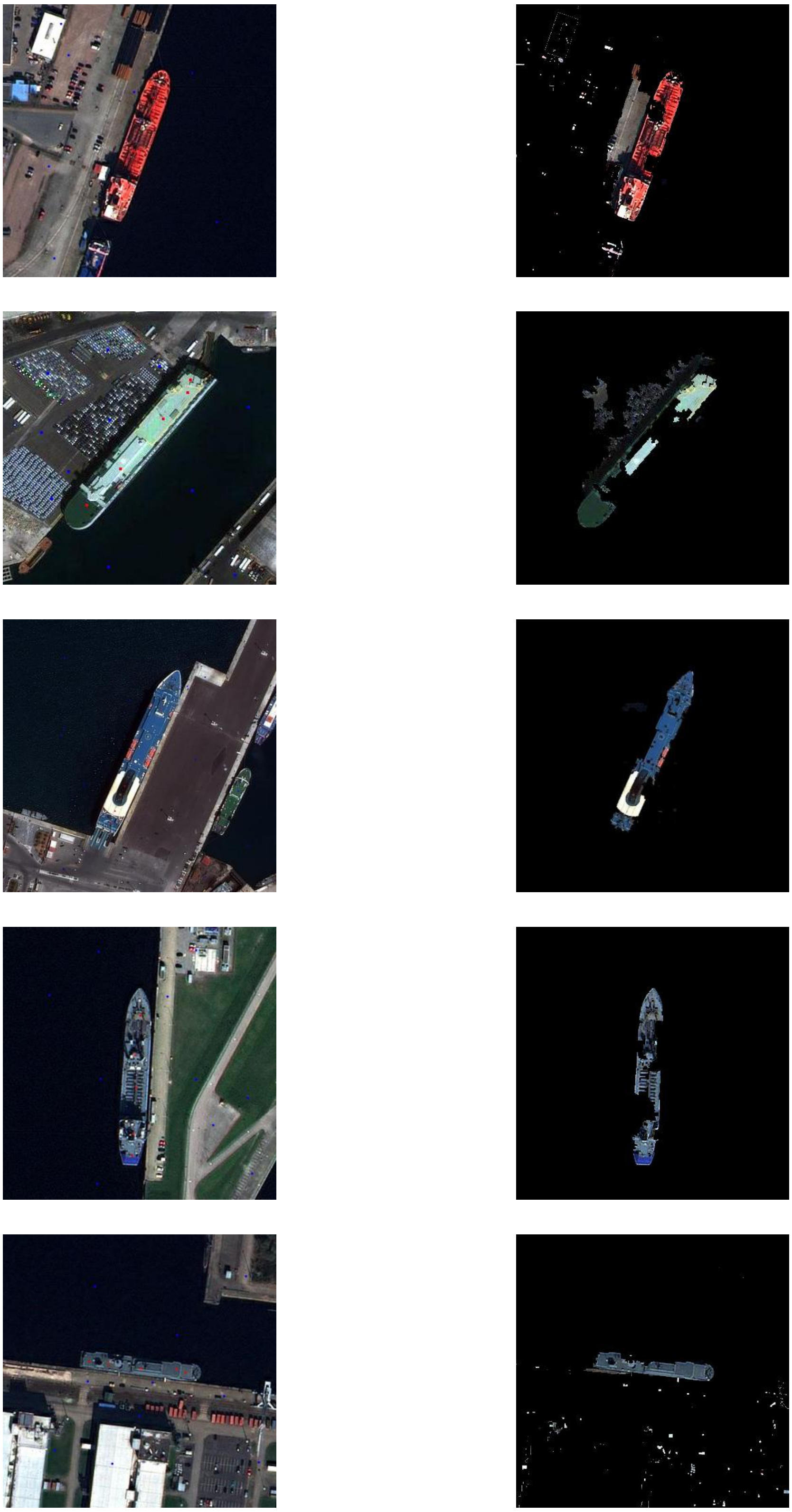
In order to evaluate the performance quantitatively, the precision-recall values of the algorithm results are expressed in Tables 9-14, respectively. As it can be observed, the objective evaluation confirms the qualitative results. As expected, GrowCut and GrabCut methods have convinced the precision-recall values, unlike the graph cut.

Ultimately, GrabCut image segmentation algorithm gives the best results among all the segmentation algorithms (fully and semi-automatic) introduced by this thesis with regards to the quality of results. At the same time, the user inputs needed for achievement of the expected results is at the level which can be easily obtained by some algorithms. As a result of these observations, it can be suggested that Grabcut method should be utilized as a fully automatic segmentation algorithm by taking the user inputs from the starting algorithms for high level applications.



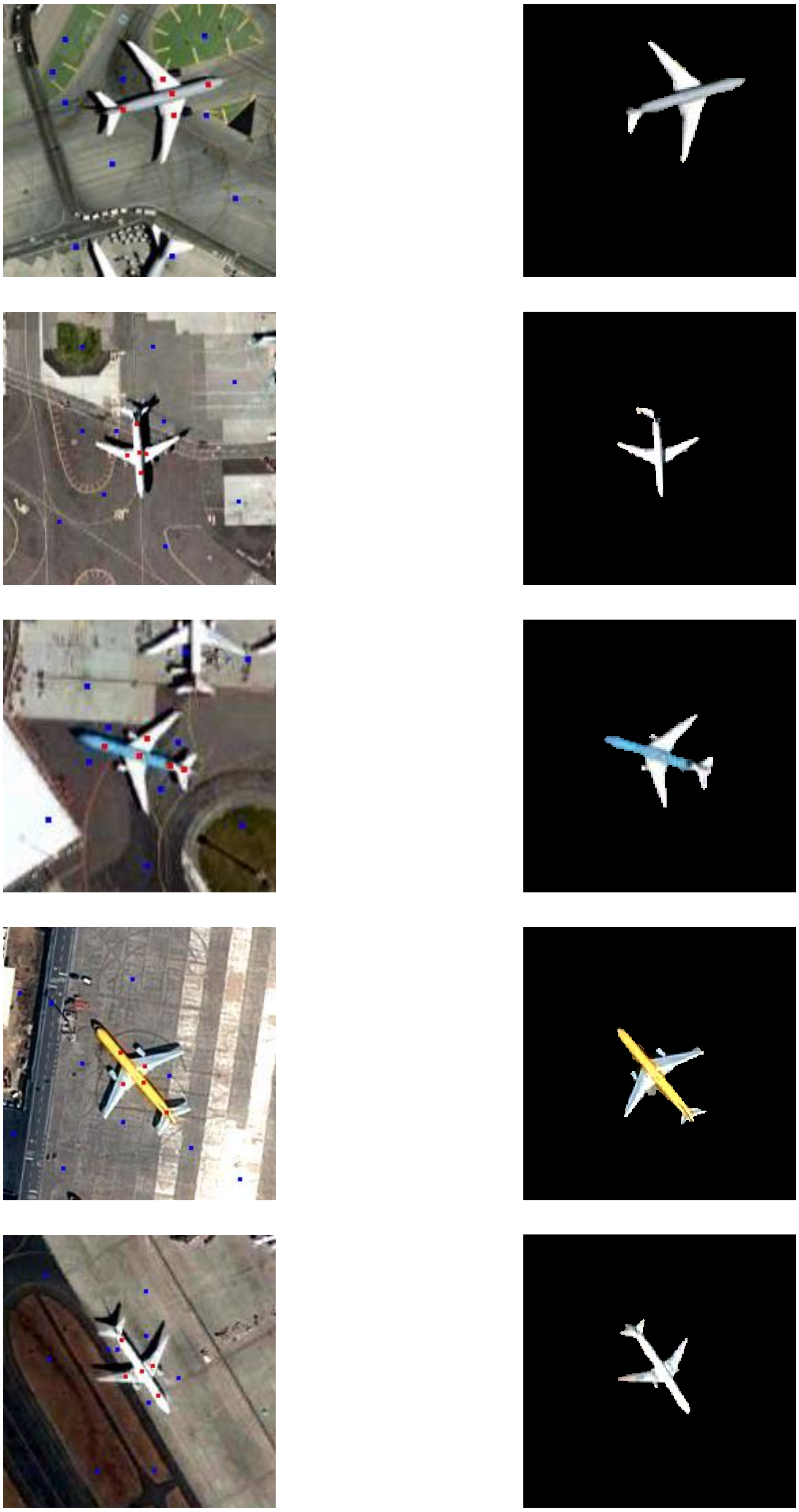
(a) (b)

Figure 16 : Interactive graph-cuts segmentation results for plane images (a) The user input (b) The segmentation results



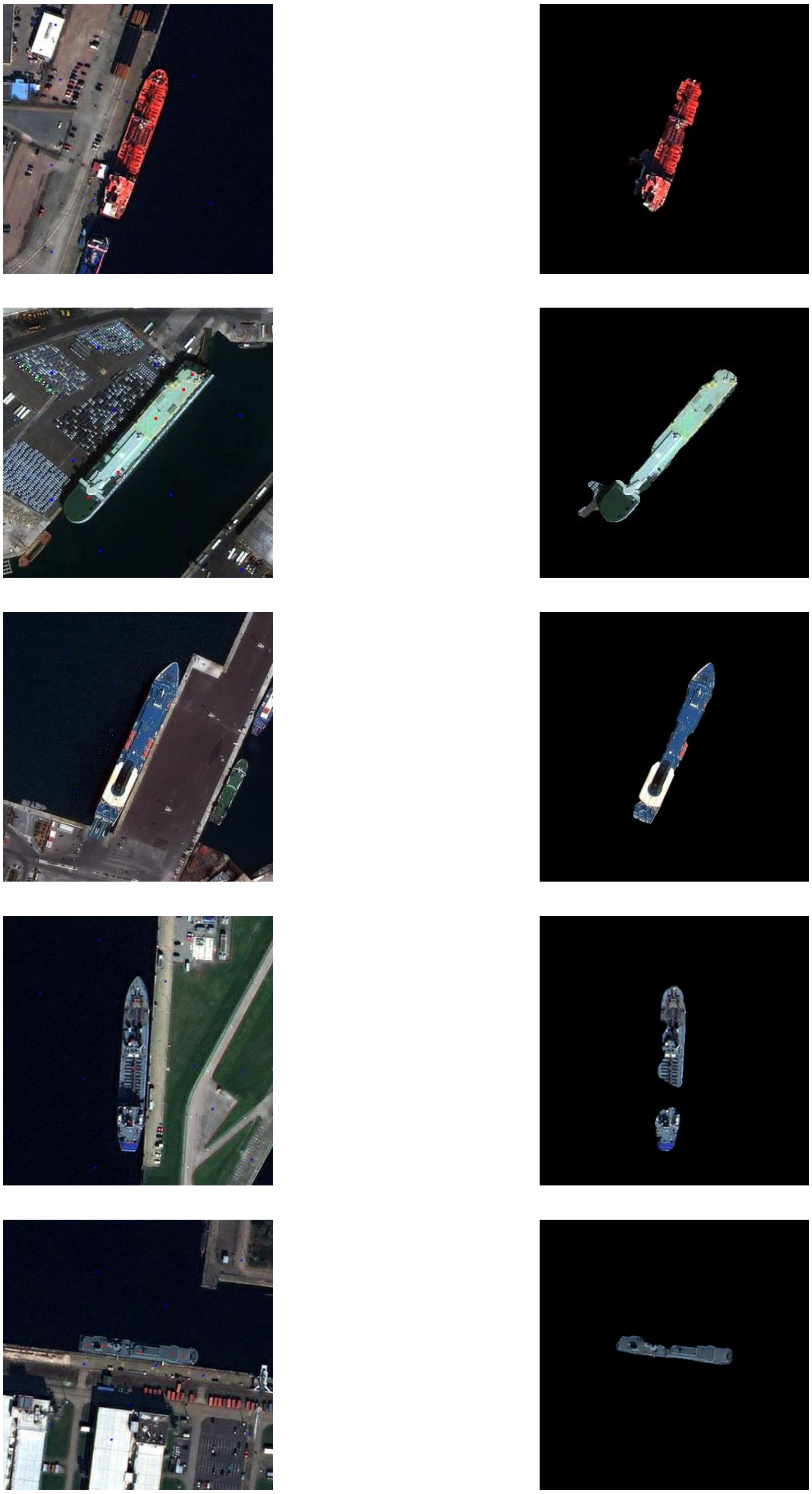
(a) (b)

Figure 17: Interactive graph-cuts segmentation results for ship images (a) The user input (b) The segmentation result



(a) (b)

Figure 18: Interactive GrowCut segmentation results for plane images (a) The user input (b) The segmentation results



(a) (b)

Figure 19: Interactive GrowCut segmentation results for ship images (a) The user input (b) The segmentation result

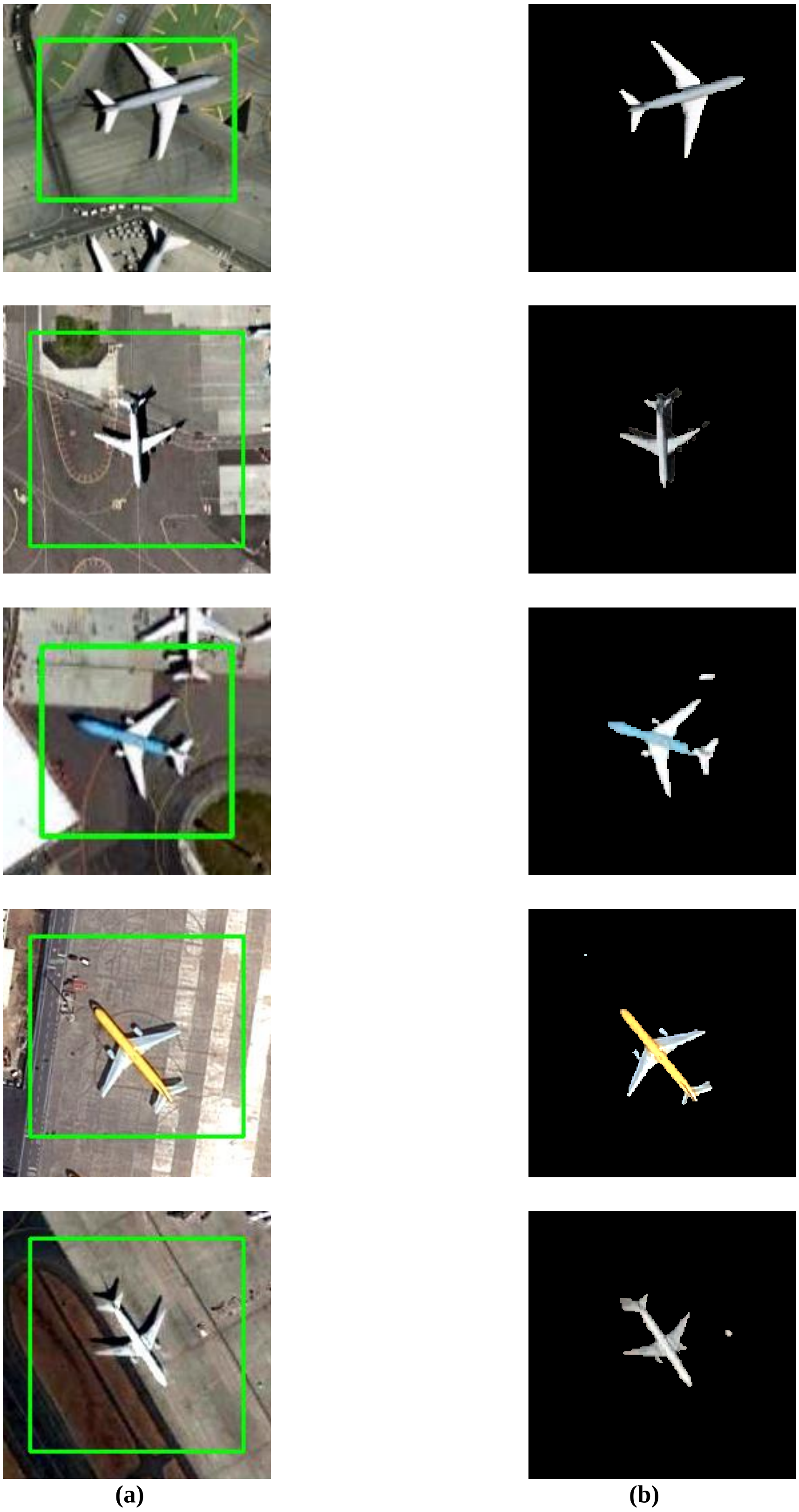
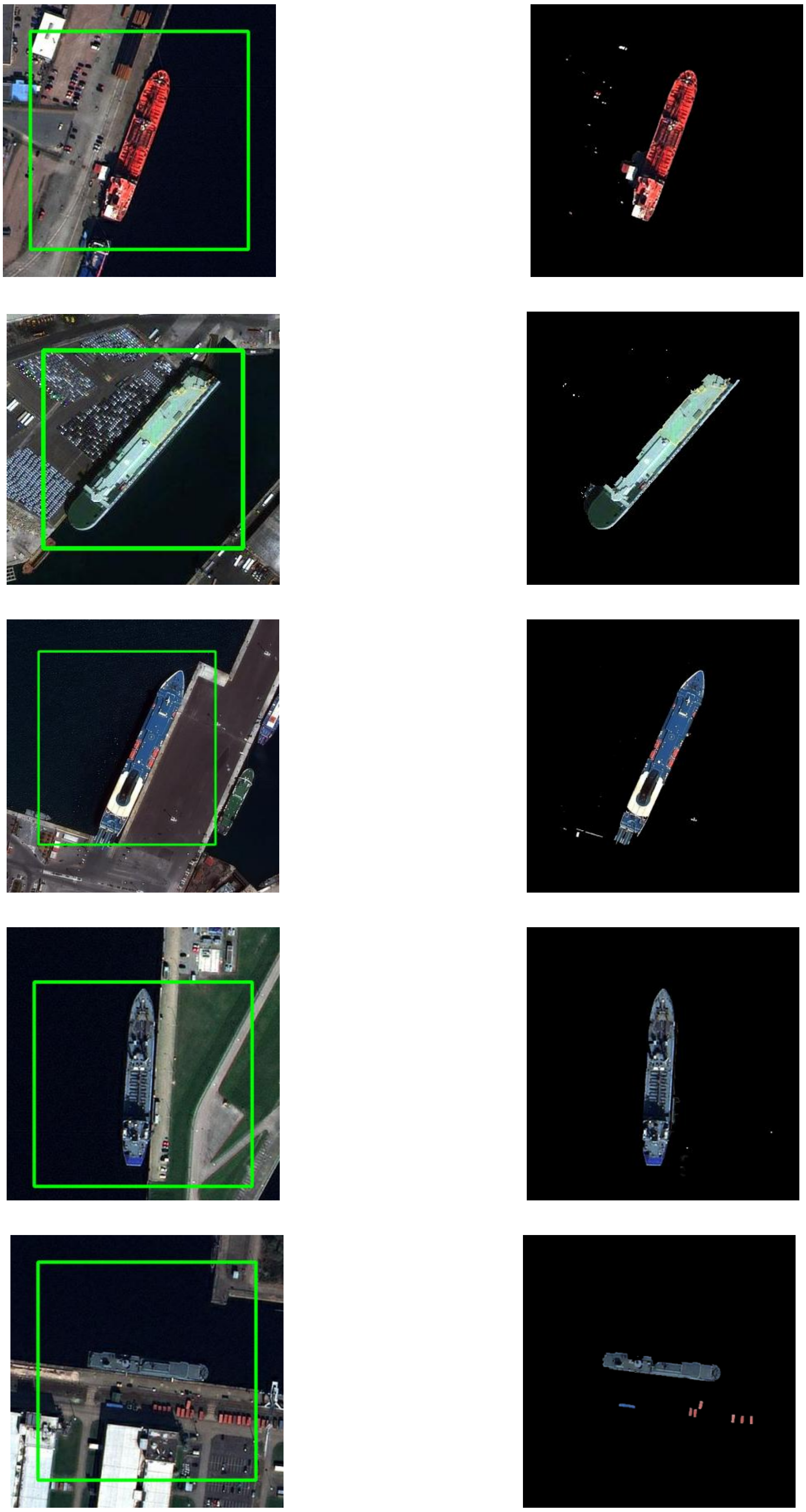


Figure 20: Interactive GrabCut segmentation results for plane images (a) The user input (b) The segmentation results



(a) (b)

Figure 21: Interactive GrabCut segmentation results for ship images (a) The user input (b) The segmentation results

Table 9 : The precision-recall values of graph cut segmentation results for plane images

Graph Cut	Precision	Recall
Plane 1	0,26	0,96
Plane 2	0,37	0,88
Plane 3	0,38	0,90
Plane 4	0,25	0,95
Plane 5	0,32	0,95
Average	0,32	0,93

Table 10 : The precision-recall values of graph cut segmentation results for ship images

Graph Cut	Precision	Recall
Ship 1	0,363	0,97
Ship 2	0,47	0,71
Ship 3	0,79	0,92
Ship 4	0,76	0,81
Ship 5	0,33	0,95
Average	0,54	0,87

Table 11 : The precision-recall values of GrowCut segmentation results for plane images

GrowCut	Precision	Recall
Plane 1	0,96	0,86
Plane 2	0,91	0,78
Plane 3	0,97	0,79
Plane 4	0,88	0,85
Plane 5	0,92	0,88
Average	0,93	0,83

Table 12: The precision-recall values of GrowCut segmentation results for ship images

GrowCut	Precision	Recall
Ship 1	0,84	0,79
Ship 2	0,91	0,70
Ship 3	0,99	0,87
Ship 4	0,99	0,70
Ship 5	0,97	0,83
Average	0,94	0,78

Table 13: The precision-recall values of GrabCut segmentation results for plane images

GrabCut	Precision	Recall
Plane 1	0,99	0,85
Plane 2	0,60	0,91
Plane 3	0,94	0,82
Plane 4	0,96	0,83
Plane 5	0,86	0,89
Average	0,87	0,86

Table 14: The precision-recall values of GrabCut segmentation results for ship images

GrabCut	Precision	Recall
Ship 1	0,82	0,95
Ship 2	0,95	0,80
Ship 3	0,92	0,95
Ship 4	0,89	0,96
Ship 5	0,88	0,84
Average	0,89	0,90

CHAPTER 3

SHAPE REPRESENTATION

Shape information compared to other primary low level image features, like texture and color, is much more effective in semantically describing the content of an image [70]. This is due to the fact that the shape of objects is strongly correlated to object functionality and identity. Human beings can recognize characteristic of objects merely from their shapes due to the human visual perception system. Therefore, this property discriminates shape information of an object from other elementary visual features, such as color and texture [71].

Shape descriptors are computational tools required for analyzing image shape information. They consist of mathematical functions, which are applied to image to produce numerical values that are representative of a specific characteristic of the given shape in the image. The nature and meaning of such numerical values depend on the definition of the shape descriptor [72]. After extracting the shape features, they can be used as input features for many image processing applications in many areas, such as meteorology, medicine, space exploration, manufacturing, entertainment, education, law enforcement and defense [73].

Although shape information is quite powerful feature for object representation, the accurate extraction and representation is a challenging process. Therefore, various numbers of studies have been published for the shape representation and description in the literature [70]–[94] until now. Shape descriptors are generally classified into two main sections with respect to the information that they take into account to calculate their measures. The whole shape pixels and boundary (perimeter) are two principal source of information and the methods used this information are denoted as *region-based* shape descriptors [89]-[94] and *contour-based* shape descriptor [74]-[88], respectively. Under each class, the different techniques are further divided into *structural* approaches and *global* approaches. This sub-class is based on whether the shape is represented as a whole or by segments/sections (primitives). Furthermore, these methods can be distinguished into *space domain* and *transform/spectral domain*, based on whether the shape features are derived from the spatial domain or spectral domain [74]. The complete hierarchy of the classification is displayed in the Figure 22.

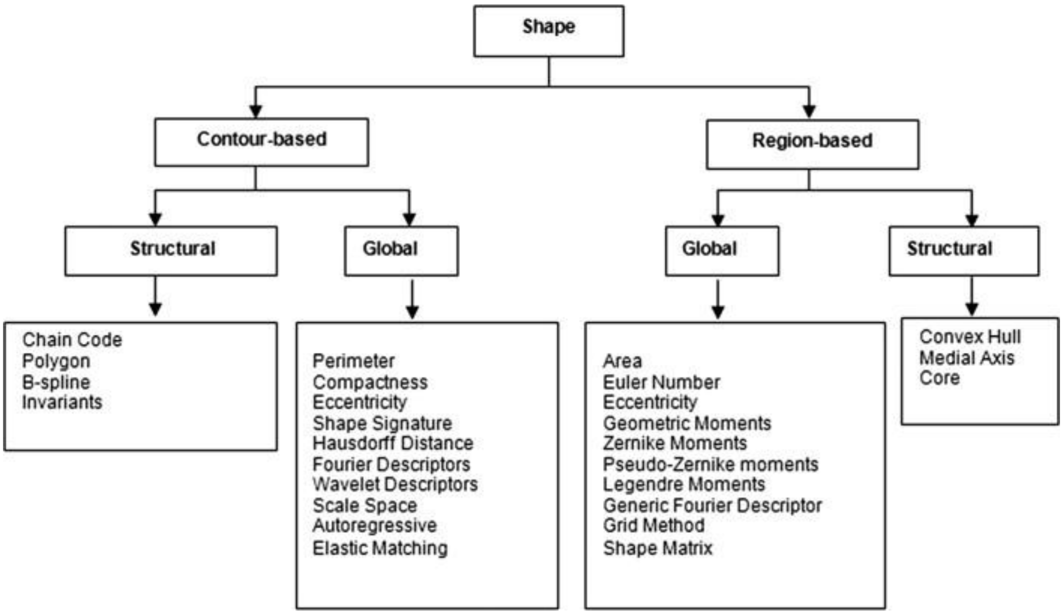


Figure 22: Taxonomy of the shape representation and description techniques [74]

The contour-based techniques only exploit the shape boundary information. They are usually clear to acquire and sufficiently descriptive for many applications. In the global approaches, a feature vector derived from the entire perimeter is used to describe the shape. The metric of the shape similarity is usually the Euclidian distance between the feature vectors. Many global contour-based shape descriptors exist in the literature. Simple shape descriptors [75] such as area, circularity (perimeter^2/area), eccentricity (length of major axis/length of minor axis), major axis orientation and bending energy are used to discriminate objects with large dissimilarities. The shape signature represents a shape by a one dimensional function derived from shape boundary points. Centroidal profile, complex coordinates, centroid distance, tangent angle, cumulative angle, curvature, area and chord-length are some of the shape signature in the literature [76] [77]. The Autoregressive (AR) method [78] is based on the stochastic modeling of a 1D function *f* obtained from the shape. The curvature scale space (CSS) method is firstly proposed for shape representation by Mokhtarian and Mackworth [79] and many improved versions of it are then published in order to eliminate its weakness [80] [81]. The spectral descriptions include Fourier descriptor [82] [83] [84] and Wavelet descriptor [85] [86] which are derived from spectral transforms on 1-D shape signatures. They overcome the problem of noise sensitivity and boundary variations existing of other approaches by analyzing the shape in the spectral domain.

Another member of the contour shape analysis approaches is a structural shape representation. The structural shape methods divide a boundary into segments, known as *primitives*, according to a particular criterion, and then an invariant is derived from each segment to represent the curve. The final representation is generally a string or a tree, and the measure of shape similarity is string matching or graph matching. Chain code representation [87], polygon decomposition [88] and smooth curve decomposition [89] are common algorithms in this category.

In region-based methods, all the pixels within a shape region are taken into account to obtain the shape representation, and so they have the ability to capture the interior content of a shape. Therefore, region-based techniques can be utilized to describe non-connected and disjoint shapes. Similar to contour-based approaches, region-based methods can also be separated into global and structural sub-sections, depending on whether they separate shapes into parts/segments or not. Moment-based descriptors [90] [91] [92] [93] [94], generic Fourier descriptor [95], grid method [96] and shape matrix [97] are commonly used to describe shapes in the global techniques. The moment-based descriptors include geometric moments (Hu moments) [90], orthogonal moments [93] such as Legendre moments, Zernike moments, pseudo-Zernike moments and angular radial transform [94]. Convex hull, medial axis [74] can be given as examples in the region-based structural approaches.

Contour-based techniques are usually sensitive to small changes or noises on the shape, which means that these methods produce different and undesirable results when the shape boundary changes slightly. Nevertheless, unlike other contour-based methods, spectral-based methods, such as Fourier and Wavelet descriptors, can handle these changes and noises on the shape boundary up to some extent [74]. On the other hand, region-based approaches are robust to such small changes on the shape; even large changes on the boundary can have a small change on the region descriptor. This means that the region-based descriptors are not sensitive to noise which can be originated from segmentation, occlusion, and distortion. However, this also means that they are unable to perceive small variations on the shape and ignore the shape details [72].

Shape descriptors generally require some essential properties in order to describe shapes efficiently and effectively. These can be explained as follows [98]:

- Identifiability: shapes perceived similar by human have the same feature representations and different from the others.
- Translation, rotation and scale invariance: the position, orientation and scaling variations of the shape must not be affected the extracted features.
- Affine invariance: the extracted features must be as invariant as possible with affine transformations.

- Noise resistance: the features must be as robust as possible against noise.
- Occlusion invariance: when some parts of a shape are occulted some other objects, the features extracted from the remaining part must be same compared to the original shape as possible.
- Statistically independent: two features extracted from same shape must be statistically independent because of compactness of the representation.
- Reliability: the derived features must remain the same as long as one deals with the same pattern.

In order to examine and compare the algorithms, which fully or partially satisfy the abovementioned properties, tremendous efforts have been devoted by researchers in computer vision and image processing communities during the last decades [99] [100] [70] [73] [101] [102] [103] [74].

In this thesis, the shape representation problem is analyzed by comparing three different methods. The performances of angular radial transform [94] [71] and geometric moment invariants (also called as Hu moment invariants) [90] for region-based and Fourier descriptor [83] for contour-based shape representation are investigated. Since the results of the segmentation algorithms which are noisy and occluded are utilized as test images during the experiments, Fourier descriptor is only the used method for contour-based representation due to its noise robustness.

This chapter is composed of four main parts; in the first section, the geometric moment invariants or Hu moment invariants are presented. Next, the angular radial transform description is explained, and the following section is devoted to Fourier descriptor method. Finally, the comparison among three different algorithms and their combinations is performed in the experimental results section.

3.1 Geometric Moments Invariants

Geometric moment invariants are firstly introduced by Hu in 1962 [90], and at that time Hu derived six absolute orthogonal invariants and one skew orthogonal invariants based on algebraic calculation, which are not only invariance of rotation, scaling and translation but also independent of general affine projection [104]. They have been extensively applied to image pattern recognition in a variety of applications due to its invariant property on image translation, scaling and rotation. Moreover, simple properties of the image such as area (or total intensity), its centroid, and information about its orientation can be obtained via image moments.

For a two-dimensional continuous image function $f(x, y)$, the moments (also called as raw moments) of order $(p + q)$ are defined as:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \tag{3.1}$$

for $p, q = 0,1,2, \ldots$. For the digital image, the raw moments are calculated as follows,

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \tag{3.2}$$

However, these moments are not invariant to rotation, translation and scaling of the object. The translation invariant feature can be acquired by using central moments and they are defined in a digital image as follows:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \tag{3.3}$$

where $p, q = 0, 1, 2, \ldots$ and $\bar{x} = \frac{m_{10}}{m_{00}}$ and $\bar{y} = \frac{m_{01}}{m_{00}}$ are the components of the centroid of the image $f(x, y)$. The centroid moments μ_{pq} calculated via equation (3.3) is equivalent to m_{pq} whose center has been shifted to the centroid of the image. Therefore, the central moments are invariant to image translation.

In order to acquire scale invariance moments, the central moments are normalized by dividing the corresponding central moment by properly scaled (00)th moment by using the following formula.

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{(1+\frac{p+q}{2})}} \ , \quad p + q \geq 2 \tag{3.4}$$

Based on the normalized central moments, Hu [90] established seven moments which are invariant under translation, changes in scale, and also rotation. These are listed as follows:

$$I_1 = \eta_{20} + \eta_{02}$$

$$I_2 = (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2$$

$$I_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$I_5 = (\eta_{30} - 3\eta_{12})\ (\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} -$$

$$\eta 03\ \eta 21{+}\eta 03\ 3\eta 30{+}\eta 122{-}(\eta 21{+}\eta 03)2$$

$$I_6 = (\eta_{20} - \eta_{02})\ [(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{21} + \eta_{03})\ (\eta_{12} + \eta_{30})$$

$$I_7 = (3\eta_{21} - \eta_{03})\ (\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} -$$

$$3\eta 12\ \eta 21{+}\eta 033\eta 30{+}\eta 122{-}(\eta 21{+}\eta 03)2$$

The first six coefficients of these moments represent the independence of translation, rotation and scaling of the object and the last one provides the skew invariance.

3.2 Angular Radial Transform

Angular Radial Transform (ART) is a moment-based image description method adopted in MPEG-7 as a region-based shape descriptor [71] [94]. It gives compact and efficient way to capture pixel distribution within a 2-D object region. This descriptor can describe complex objects composed of multiple disconnected regions as well as simple objects with or without holes. For example, during the segmentation process the target object might be split into disconnected sub-regions. Such an object can still be retrieved, if the information on the isolated sub-regions is provided and used during the descriptor extraction. Hence, the descriptor is robust to segmentation noise, such as the disconnected parts or salt and pepper noise. Furthermore, it has the property of invariance in rotational, translational and scaling of the object.

ART is a complex orthogonal unitary transform defined on a unit disk that consists of the complete orthonormal sinusoidal basis functions in polar coordinates [71]. Due to the orthonormal basis, it renders no redundant information among the coefficients and makes the representation compact and effective. The ART coefficients, F_{nm} of order n and m, are defined by:

$$F_{nm} = \langle V_{nm}(\rho,\theta), f(\rho,\theta)\rangle = \int_0^{2\pi}\int_0^1 V_{nm}^*(\rho,\theta) f(\rho,\theta)\rho d\rho d\theta \qquad (3.5)$$

where $f(\rho,\theta)$ is an image function in polar coordinates and $V_{nm}(\rho,\theta)$ is the ART basis functions that are separable along the angular and radial directions, that is,

$$V_{nm}(\rho, \theta) = A_m(\theta)\, R_n(\rho) \tag{3.6}$$

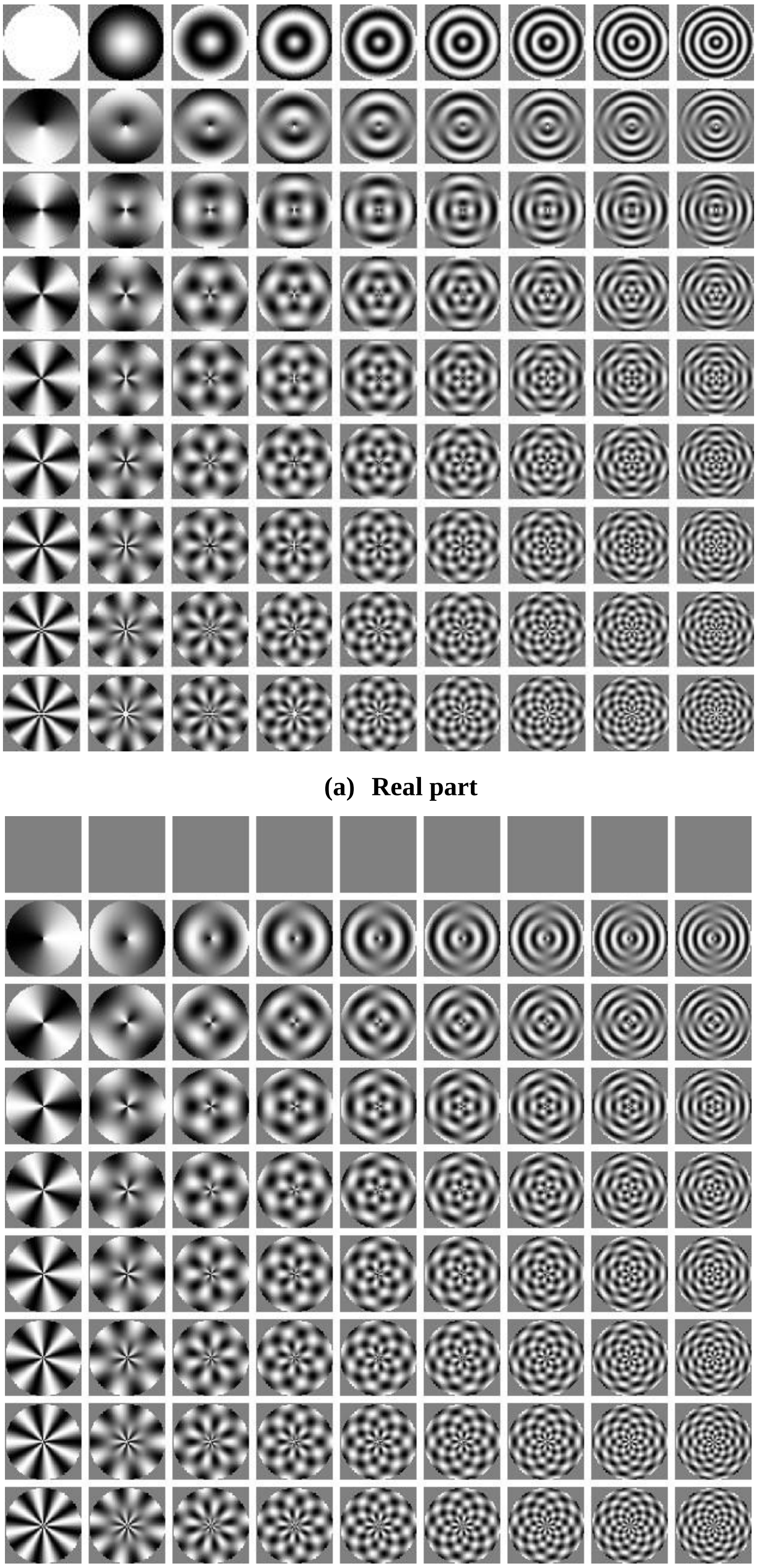
In order to achieve rotation invariance, an exponential function is used for the angular basis functions and given by,

$$A_m(\theta) = \frac{1}{2\pi} \exp(jm\theta) \tag{3.7}$$

The radial basis function is described by a cosine function,

$$R_n(\rho) = \begin{cases} 1, & n = 0 \\ 2\cos\,(\pi n \rho), & n > 0 \end{cases} \tag{3.8}$$

The real part and imaginary parts of ART basis functions are shown in Figure 23.



(a) Real part

(a) Imaginary part

Figure 23: Real and imaginary parts of ART basis functions

The magnitudes of the ART coefficients are inherently rotation invariant. This can be examined in the following simple calculations [71].

Let the image $f^{\alpha}(\rho,\theta)$ be the rotated version of $f(\rho,\theta)$ by an angle α about its origin,

$$f^{\alpha}(\rho,\theta) = f(\rho,\theta+\alpha) \tag{3.9}$$

the ART coefficients of rotated images are then given as:

$$F_{nm}^{\alpha} = \langle V_{nm}(\rho,\theta), f^{\alpha}(\rho,\theta)\rangle = \int_0^{2\pi}\int_0^1 V_{nm}^{*}(\rho,\theta) f(\rho,\theta+\alpha)\rho d\rho d\theta \tag{3.10}$$

and this expression can be expressed in,

$$F_{nm}^{\alpha} = F_{nm}\exp(-jm\alpha) \tag{3.11}$$

Hence, the magnitude of the ART of the rotated image and that of the reference is the same, that is,

$$\|F_{nm}^{\alpha}\| = \|F_{nm}\| \tag{3.12}$$

The ART descriptor is defined as a set of normalized magnitudes of ART coefficients. The rotational invariance is obtained by using the magnitude of the coefficients. For scale normalization, the ART coefficients are divided by the magnitude of the ART coefficient, F_{00}, of order *n* = 0, *m* = 0 which is proportional to the area of the object. In order to achieve the translational invariance, the center of the polar coordinate system is defined as the center of mass of the object, which can be easily acquired by geometric moments [105]. According to the MPEG-7 Visual Shape Descriptors [71], twelve angular and three radial functions

(n < 3, m < 12) are used for the calculation of the basis. Therefore, the ART descriptor is expressed by 35 coefficients which are normalized by $|F_{00}|$.

3.3 Fourier Descriptors

Fourier descriptor (FD) is a well-known spectral domain contour-based shape representation method which has nice characteristics, such as simple derivation, easy normalization, good representation of shape, noise robustness. These desirable properties have made the methods based on FD very popular in a wide range of applications, and many of them have been reported in the literature including for shape analysis [77] [82], shape classification [78], character recognition [83], shape retrieval [70] [95, 84, 106] and shape coding [87]. In these methods, different shape signatures have been exploited to obtain FD.

In general, FDs are obtained by applying Fourier transforms on a shape signature and the resulting transformed coefficients are normalized in order to get the Fourier descriptor of the shape. These descriptions represent the shape of the object in the frequency domain. By Fourier descriptors, global shape features are captured by the first few low frequency terms, while higher frequency terms capture the finer details of the shape. Thus, the noise sensitivity in the shape signature representation is surmounted by taking first few low frequency terms of FDs with powerful discrimination capability. Moreover, the compact representation of a shape is obtained by receiving a subset of FDs, and this offers low computational complexity which is an important characteristic of a desirable shape descriptor for indexing and online retrieval.

The shape signature is a one-dimensional function which is derived from boundary coordinates to represent any shape. Many shape signatures, such as centroidal profile, complex coordinates, centroid distance, tangent angle, cumulative angular function, curvature function, have been commonly exploited to obtain FD in the literature. However, FD derived from different signatures has significant different performance on shape retrieval. As it has been shown in [106], FD derived from centroid distance function outperforms the others in overall performance, and hence, this type of shape signature is utilized in the proposed system. In addition, shape signatures can be employed for shape representation without Fourier transform. However, noise sensitivity and rotation invariance issues create substantial problems in this case.

The first stage of calculating FD is to obtain the boundary coordinates with parametric representation $\big(x(t),\ y(t)\big),\ \ t = 0,1,2,\ldots N-1$ and N is the number of boundary points. The extraction of the shape boundary points is implemented in pre-processing stage, which consists of some image processing methods. After extracting the boundary points, the shape signatures are calculated. The centroid distance function $r(t)$ is expressed by the distance of the boundary points from the centroid (x_c, y_c) of the shape

$$r(t) = ([x(t) - x_c]^2 + [y(t) - y_c]^2)^{1/2}, \quad t = 0,1,\ldots,N-1 \qquad (3.13)$$

where

$$x_c = \frac{1}{N}\sum_{t=0}^{N-1} x(t) \qquad and \qquad y_c = \frac{1}{N}\sum_{t=0}^{N-1} y(t) \qquad (3.14)$$

Afterwards, the discrete Fourier transform of the centroid distance shape signature $r(t)$ is calculated by,

$$a_n = \frac{1}{N}\sum_{t=0}^{N-1} r(t)\exp\,(\frac{-j2\pi nt}{N}), \quad n = 0,1,2,\ldots,N-1 \qquad (3.15)$$

$a_n,\ \ n = 0,1,\ldots,N-1$ are the Fourier transformed coefficients of $r(t)$.

Since shapes generated through rotation, translation and scaling of any object are equivalent, the shape descriptors should be invariant to these operations. In the following, it has examined the effects of the change of starting point, translation, rotation and scale on the Fourier coefficients [84]. The original boundary is expressed as $r^{(o)}(t)$ for shape signature and as $a_n^{(o)}$ for its Fourier coefficients in this analysis.

- Change of starting point

The change of starting point can be stated as $r(t) = r^{(o)}(t + \delta)$. Then, the resulting Fourier coefficients become $a_n = \exp\,(jn\delta)a_n^{(o)}$.

- Translation

The translation of a shape can be expressed as $r(t) = r^{(o)}(t) + c$ and the translated contour then has the Fourier coefficients

$$a_n = \begin{cases} a_n^{(o)}, & n \neq 0 \\ a_n^{(o)} + c, & n = 0 \end{cases}$$

This implies that the Fourier coefficients except the first one have translational invariance property. The first coefficient (DC component) only represents information about the position or average scale of the shape with respect to the used shape signature and so it is not useful in describing the shape. Thus, it is discarded.

- Rotation

Assuming the center of mass is positioned at the origin, rotation of $r^{(o)}(t)$ around the origin with angle θ gives the curve expression $r(t) = r^{(o)}(t)\exp\,(j\theta)$. This results in the change of Fourier coefficients by $a_n = \exp(j\theta)\,.\,a_n^{(o)}$.

- Scaling

The scale change in a shape can be specified as $r(t) = s.r^{(o)}(t)$ and the change in the Fourier coefficients are expressed as $a_n = \mathrm{s}.a_n^{(o)}$

The outline features of similar shapes are only interested property in shape retrieval and description. Therefore, the shape representations must be invariant to translation, rotation and scale in order to make the model shape and data shapes comparable. Since the shape signature is invariant under translation, the corresponding FDs are also translation invariant. Rotation invariance and independence of starting point are achieved by ignoring the phase information and by taking only the magnitude values of the FDs. For the centroid distance shape signature, the first component or DC component of the FDs reflects the average scale of the corresponding shape. Therefore, scale invariance is then obtained by dividing the magnitude values of the first half of FDs by the DC component.

For the centroidal distance shape signature, unlike the other ones, only half of the FDs are needed to index the shape because the function in (3.13) is real-valued and so there are only N / 2 different frequencies in the Fourier transform. As a result, the FDs, which have invariance properties, can be expressed mathematically,

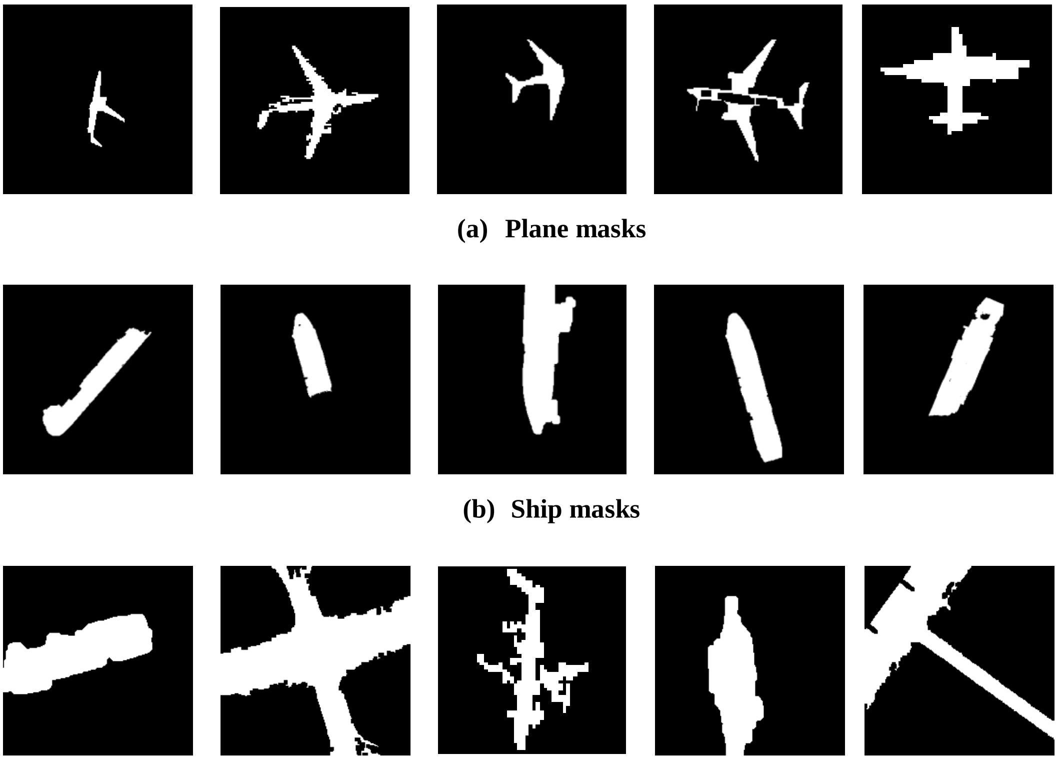
$$F = \left[\frac{|FD_1|}{|FD_0|}, \frac{|FD_2|}{|FD_0|}, \dots, \frac{|FD_{N/2}|}{|FD_0|}\right] \tag{3.16}$$

Finally, the similarity between a query shape Q and a target shapes T is determined by the Euclidean distance d between their FDs:

$$d = \left(\sum_{i=1}^{N/2} |FD_i^Q - FD_i^T|^2\right)^{1/2} \tag{3.17}$$

3.4 Experimental Results

In this part of the chapter, the shape description performances of the introduced algorithms are quantitatively evaluated on binary test images which are obtained from various segmentation algorithm outputs. Three types of binary masks are used during the experiments, and these are called as plane masks, ship masks and other masks which are procured from the segmentation results of the image parts which do not include the targets, such as plane and ship, but they can have confusion with them. Some of the examples of these masks are illustrated in the Figure 24. As it can be observed, the masks used as test images have the missing or adding parts resulted from the occlusion and/or segmentation deficiencies and are also noisy.



(a) Plane masks

(b) Ship masks

(a) Other masks

Figure 24 : Example of test silhouettes used during the tests

The performances of the methods are analyzed by means of the nearest neighborhood (NN) and k-nearest neighborhood (k-NN) classification algorithms. The nearest neighborhood (NN) algorithm is a method for classifying objects based on closest training examples in feature space. In this algorithm, the Euclidian distance is employed to determine the closest training sample from the query sample and the method is simply defined mathematically;

$$l = label \left\{ \min_{T} \sqrt{\sum_{i=1}^{d} (fv_i^Q - fv_i^T)^2} \right\} \tag{3.18}$$

where fv denotes d-dimensional feature vector, Q and T represent the query image and training set, respectively. The k-nearest neighbor method is an obvious extension of the NN method. This rule classifies the query sample by assigning it the label most frequently represented among the k nearest training samples. In other words, a decision is made by examining the labels on the k nearest neighbors and taking a vote [42].

The algorithm performances are displayed via confusion matrix in this study. The concept of the confusion matrix is proposed by Kohavi and Provost in 1998 [107] and it is defined as a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. One benefit of a confusion matrix is that it is easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

The experimental results are displayed for angular radial transform (ART), geometric moment invariants (Hu moment invariants), Fourier descriptors and their combinations as the shape descriptor methods in the Table 15-28, respectively. The training sets used during the experiments composed of 770 different masks in plane class, 380 masks in ship class and 2409 masks in the other class. The masks on the plane and ship class are acquired from ground truth of the target objects and so they have precise shapes representing the objects correctly and thoroughly. The number of masks in the test set is 82 for plane class, 83 for ship and 307 for the other.

Each shape representation algorithm gives feature vectors with value of elements in a different range of the feature space, and this situation brings about a different Euclidian distance result when compared same shapes. Therefore, this can result in erroneous classification and retrieval performances when various different shape descriptor methods are utilized at the same time. Thus, the feature vectors should be normalized while testing the performance of the combination algorithms.

As it can be observed from the confusion matrices, angular radial transform (ART) gives the best result among three analyzed shape algorithms for retrieving of the plane masks and ship masks. However, ART descriptors erroneously confuse the target with masks in the other class (especially between ship and other classes). Despite of high recall values, the utilization of the angular radial transform causes the performance with low precision in the retrieval and classification operations.

The geometric moment invariants submit quite inferior results for the plane mask retrieval, although the ship and the other mask result are comparable with the corresponding best results. The performance analysis about Hu moments invariants shows that they successfully retrieve the masks with uncomplicated characteristic like a ship, while the complex shape cannot be represented and described fully and accurately.

The results in the confusion matrices show that the Fourier descriptor method can highly discriminate masks in the other class from the target masks. Nevertheless, the target retrieval performance is worse than the other two algorithms.

Finally, the experiments are carried out for all of the combinations of the algorithms. The experimental results show that the performance characteristic of each method determines the performance of the integrated methods. As a result, the combination of all of the methods which are ART, Hu moments invariants, Fourier descriptors outperforms in the overall case.

Table 15: The confusion matrix for ART with NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	75 (92 %)	0 (0 %)	7 (8 %)
	Ship	0 (0 %)	78 (94 %)	5 (6 %)
	Other	38 (12 %)	77 (25 %)	192 (63 %)

Table 16: The confusion matrix for ART with k-NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	75 (92 %)	0 (0 %)	7 (8 %)
	Ship	0 (0 %)	81 (98 %)	2 (2 %)
	Other	40 (13 %)	88 (29 %)	179 (58 %)

Table 17: The confusion matrix for geometric moment invariants with NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	33 (40 %)	0 (0 %)	49 (60 %)
	Ship	0 (0 %)	74 (89 %)	9 (11 %)
	Other	38 (12 %)	41 (13 %)	228 (75 %)

Table 18: The confusion matrix for geometric moment invariants with k-NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	30 (37 %)	0 (0 %)	52 (63 %)
	Ship	0 (0 %)	80 (96 %)	3 (4 %)
	Other	30 (10 %)	58 (19 %)	219 (71 %)

Table 19: The confusion matrix for Fourier descriptors with NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	73 (89 %)	0 (0 %)	9 (11 %)
	Ship	0 (0 %)	69 (83 %)	14 (17 %)
	Other	23 (7 %)	45 (15 %)	239 (78 %)

Table 20: The confusion matrix for Fourier descriptors with k-NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	75 (91 %)	0 (0 %)	7 (9 %)
	Ship	0 (0 %)	76 (92 %)	7 (8 %)
	Other	21 (7 %)	44 (14 %)	242 (79 %)

Table 21: The confusion matrix for ART plus geometric moments with NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	74 (90 %)	0 (0 %)	8 (10 %)
	Ship	0 (0 %)	80 (96 %)	3 (4 %)
	Other	32 (10 %)	80 (26 %)	195 (64 %)

Table 22: The confusion matrix for ART plus geometric moments with k-NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	76 (93 %)	0 (0 %)	6 (7 %)
	Ship	0 (0 %)	81 (98 %)	2 (2 %)
	Other	32 (10 %)	91 (30 %)	184 (60 %)

Table 23: The confusion matrix for ART Fourier descriptors with NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	80 (98 %)	0 (0 %)	2 (2 %)
	Ship	0 (0 %)	76 (92 %)	7 (8 %)
	Other	29 (9 %)	59 (19 %)	219 (71 %)

Table 24: The confusion matrix for ART plus Fourier descriptors with k-NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	78 (95 %)	0 (0 %)	4 (5 %)
	Ship	0 (0 %)	81 (98 %)	2 (2 %)
	Other	27 (9 %)	66 (21 %)	214 (70 %)

Table 25: The confusion matrix for geometric moments plus Fourier descriptors with NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	76 (93 %)	0 (0 %)	6 (7 %)
	Ship	0 (9 %)	73 (88 %)	10 (12 %)
	Other	27 (9 %)	44 (14 %)	236 (77 %)

Table 26: The confusion matrix for geometric moments plus Fourier descriptors with k-NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	76 (93 %)	0 (0 %)	6 (7 %)
	Ship	0 (0 %)	77 (93 %)	6 (7 %)
	Other	21 (7 %)	54 (18 %)	232 (75 %)

Table 27: The confusion matrix for all of the algorithms with NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	80 (98 %)	0 (0 %)	2 (2 %)
	Ship	0 (0 %)	77 (93 %)	6 (7 %)
	Other	27 (9 %)	62 (20 %)	218 (71 %)

Table 28: The confusion matrix for all of the algorithms with k- NN

		PREDİCTED CLASS		
		Plane	Ship	Other
ACTUAL CLASS	Plane	79 (97 %)	0 (0 %)	3 (3 %)
	Ship	0 (0 %)	81 (98 %)	2 (2 %)
	Other	29 (9 %)	69 (22 %)	209 (69 %)

CHAPTER 4

PROPOSED ALGORITHM

As already mentioned in Chapter 1, a significant number of geospatial object detection algorithms in the literature have several deficiencies which can create a reduction in the system performance. The first drawback is that they produce detection outputs with a considerable amount of false positives (i.e. false alarms), which results in a low precision rate. The second one is that many detection algorithms cannot give the exact and accurate information about the location of the objects. Finally, numerous applications which use the object detection and recognition algorithm also request the object mask for various different purposes, such as measuring the object size or aspect ratio, assigning the type of an object etc. Note that the resolution of the image must be also known to be able to determine the object size. As a result, a shape based object recognition system is proposed to be able to overcome these drawbacks in this thesis. Since shape features offer characteristic information that provides powerful discrimination ability for many object classes, such as aircrafts and helicopters, the false positives created by any algorithm which does not use the shape features can dramatically be reduced by using the shape representation techniques. Moreover, after the object mask is extracted from the image, the target position can be procured with together the object mask itself.

The overall structure of the remaining of this chapter is organized as follows. In Section 4.1, the proposed method for geospatial objects is represented elaborately. Then, the subsequent section explains the algorithm used for hypothesis point generation purpose in detail. The next part clarifies the SVM classifier, which is utilized in the proposed method. Finally, the performance tests and results are presented in Section 4.4.

4.1 Proposed geospatial object recognition algorithm

The developed object recognition procedure in this thesis combines a top-down object detection algorithm with the bottom-up image segmentation approach. In general, top-down methods often include a training stage to obtain class-specific model features or to define object configurations. Hypotheses are then found by matching models to the image features. On the contrary, bottom-up approaches begin from low-level or mid-level image features. They build up hypotheses from such features, extend them by construction rules and then evaluate by certain cost functions. Thus, there are two main steps in the improved technique: a hypotheses generation step and a verification step. In the top-down hypotheses generation step, a typical top-down object detection method is used to generate a set of hypothesis of object locations, which have high recall and low precision rates. In the verification step, the feasible foreground object that is consistent with the top-down object hypothesis is first extracted via a bottom-up image segmentation algorithm, and then the shape descriptors and classifier are utilized to prune out the false positives. It exploits the fact that false positive regions typically have a different shape mask from the target object. As a result, the proposed algorithm can achieve both high precision and high recall rates by taking advantage of two types of object detection approaches.

The general flowchart of the proposed system is represented in Figure 25. The hypotheses generation step is implemented in the second block of flowchart called as the generation of hypothesis points. The remaining parts of the flowchart correspond to the verification step. The study in this thesis predominantly focuses and analyzes to these parts in the verification step.

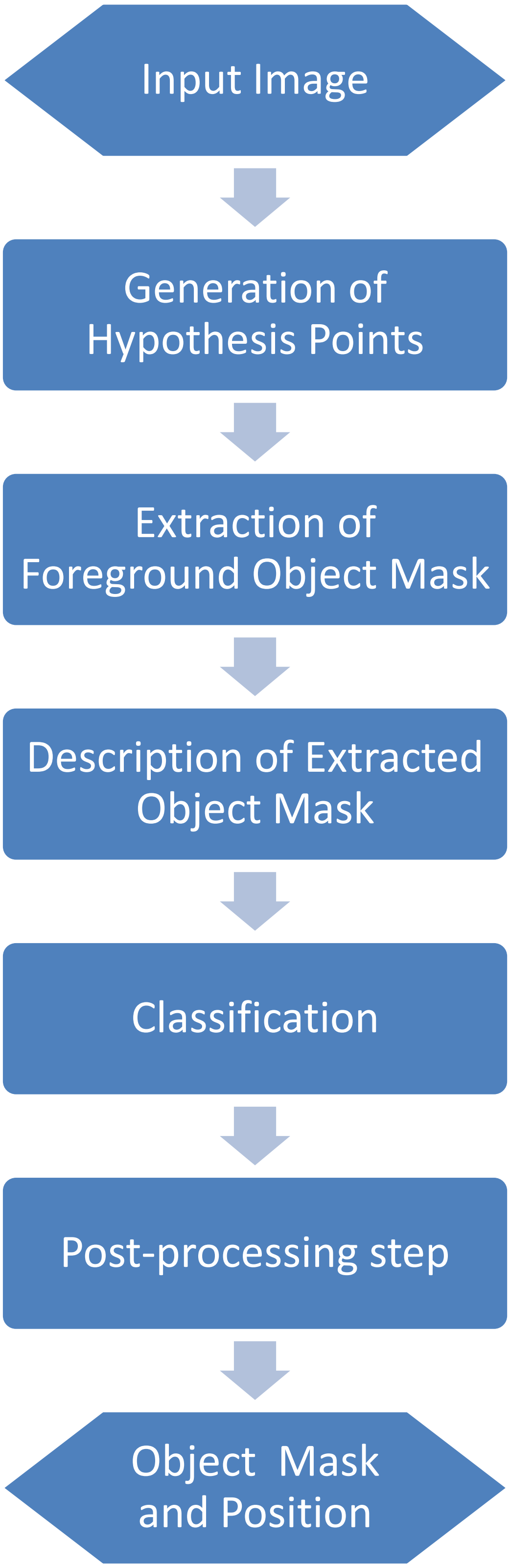

Input Image
Generation of Hypothesis Points
Extraction of Foreground Object Mask
Description of Extracted Object Mask
Classification
Post-processing step
Object Mask and Position

Figure 25: Overall flowchart of the proposed object recognition algorithm

During the generation of hypothesis points, potential target objects in the input image are detected by using one of the traditional top-down object detection algorithms in the literature. Afterwards, the hypothesis points which are on the object or near the object are extracted to represent the found objects in the image. Since many object detection algorithms utilize the sliding window mechanism during the detection process, the generation of hypothesis points and determination of their locations can be tricky. Therefore, the implementation of the generation process can vary for each object detection algorithm. For example, in the bag of visual words (BoVW) methods; these points can be obtained by using the position of the most important words in the sliding image window. In the appearance based methods, center point of the sliding window can be used as the hypothesis point.

After detecting the hypothesis points, the silhouette of the foreground object should be extracted from the image window which built around the hypothesis point by involving the target. The extraction process is realized by means of an interactive image segmentation algorithm as explained in Chapter 2. First of all, different fully automated image segmentation techniques are examined and tested for this purpose. However, for the general case none of them can separate foreground object from background correctly and thoroughly, since they only employ the low level features, such as color and intensity. Therefore, in order to regard high level information about the foreground object and background during the extraction, several interactive algorithms are analyzed in terms of both the quality performance and the amount of user effort which can get the same quality result.

As it can be examined in Chapter 2, interactive GrabCut image segmentation algorithm outperforms the other fully or semi-automatic methods in terms of the segmentation quality and the required user effort. To be able to give extra knowledge about the object and background during the extraction process, the hypothesis points obtained from the other object detection algorithm are exploited as user inputs in this study. Thereby, the interactive foreground extraction process becomes fully automated in the overall system. After identifying the hypothesis point, a rectangle including the target object is taken around this point within the built image window. The rectangle size is selected as twice of the average target size in order to capture the entire object in the image. This input rectangle states that outside of it belong to the background region and inside of it is possible foreground one. Thus, color information for the object and background, which is specified by other algorithm in this case, is utilized to establish the regional constraints in the system. Furthermore, the position of hypothesis point is employed to select the connected component belonging to the desired object after the segmentation algorithm runs. Hence, regions which appear in the segmentation output but do not belong to the desired object can be eliminated with this knowledge.

The next step is to describe the extracted mask via shape representation techniques. In Chapter 3, the performance of various shape description methods and their combinations are investigated. Orthogonal moment based Angular Radial Transform (ART) and geometric or algebraic moment based Hu moments invariants are used as the region based shape descriptors. Fourier descriptor is utilized for contour based representation due to its robustness against segmentation noise. The experimental results show that the combination of these three techniques gives the best retrieval performance and so this integrated method is employed in the proposed method.

After producing the shape based feature vectors, a classifier is trained by the shape descriptor vectors of the training set used in the Chapter 3. Afterwards, a test image is classified as object or non-object by the trained classifier. In this work, support vector machine, SVM, is employed as a classifier. It is exhaustively explained later.

Object detection algorithms, such as BoVW, can detect same object in multiple times and so the multiple detection causes big delusion while evaluating the performance of algorithms. Therefore, these multiple detections must be eliminated to make accurate performance analysis. As a final step, these multiple detections are pruned by means of a simple post-processing. In this step, object masks extracted from the same object as the result of multiple hypothesis points is united with respect to following uncomplicated thresholding technique:

$$result_{mask} = union(mask_1,\ mask_2), \quad if\ \frac{intersect(mask_1, mask_2)}{\min(mask_1, mask_2)} > T \quad (4.1)$$

This fusion operation reduces the number of false alarms resulting from multiple detections, and so the precision rate increases significantly.

4.2 Bag of Visual Words based Object Detection Algorithm

Among many different automatic object detection and recognition methods, Bag of Visual Words (BoVW) is widely used algorithm exploiting local features for object representation and description. In many scientific contests, such as TRECVID [108] or PASCAL [109], BoVW and its derivative methods are studied and they outperform the other algorithms in the literature. Therefore, a version of BoVW algorithm suggested in [1] is utilized for hypothesis generation in this thesis.

This method stems from text analysis wherein a document is represented by word frequencies (i.e. a sparse histogram over the vocabulary) without regard to their order. Similarly, bag of words algorithm in computer vision basically considers an image as a document combined with different number of visual words without regarding the position of the words in the document. The visual words are defined in terms of local image features. The flowchart of the algorithm is shown in the Figure 26.

As it can be seen, the algorithm typically involves five steps. These are listed as follows [1]:

- ✓ Feature detection
- ✓ Feature description
- ✓ Codebook generation
- ✓ Mapping images into histograms of visual words
- ✓ Classification

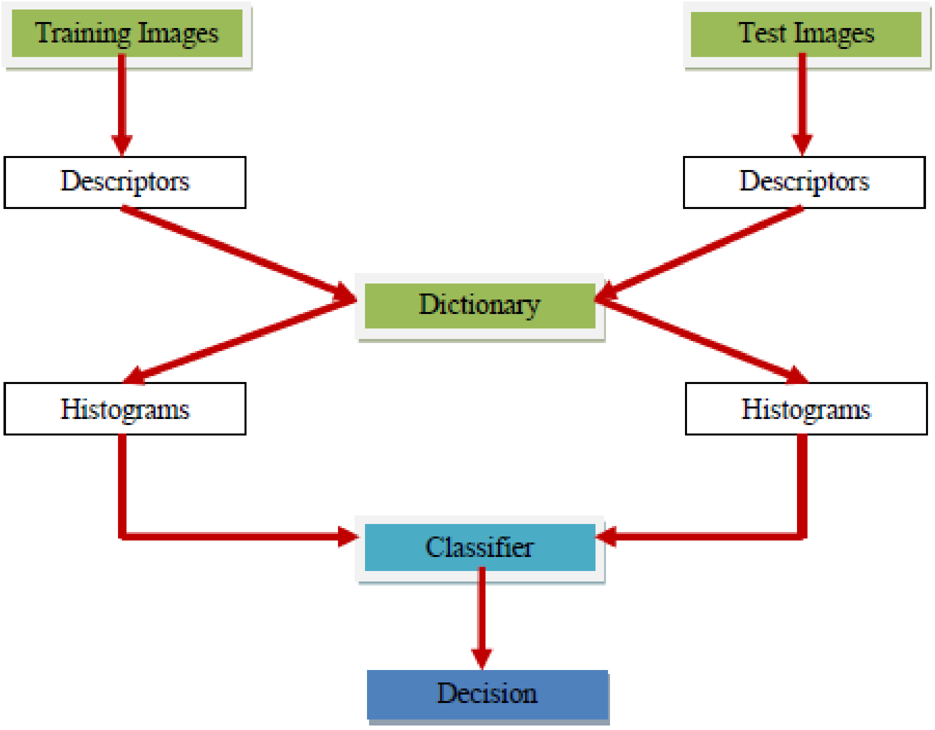


Figure 26: General block diagram of BoVW method [1]

The first step in the method is to detect interest points or regions (i.e. keypoints) from the image. For this purpose, the corner detectors, like Harris corner detector [110], or blob detectors, such as scale invariant feature transform (SIFT) [111], can be used. After detecting the interest points or regions, each image is abstracted by several local patches. Feature description methods deal with how to represent the patches as numerical vectors and these vectors are called feature descriptors. A feature descriptor should handle intensity, rotation, scale and affine variations to some extent. One of the most famous descriptors satisfying this ability is the scale-invariant feature transform (SIFT) [111]. SIFT descriptor converts each patch to 128-dimensional vector. After this step, each image is a collection of vectors of the same dimension (128 for SIFT). The next step after feature detection and description is to convert the feature vector represented patches to *codewords*, i.e. visual words, in order to produce a visual word dictionary, i.e. *codebook*. A codeword can be considered as a representative of several similar patches. One simple method to achieve this goal is performing k-means clustering [112] over all the descriptor vectors of visual data. Codewords are then defined as the centroids of the learned clusters. The number of the clusters is the visual word dictionary size, and it is manually defined. Thus, each patch in an image is mapped to a certain *codeword* through the clustering process and the image can be represented by the histogram of the codewords from a fixed dictionary of K words. The shape of the histogram is assumed to be the most informative clue about the existence of an object in an image. Finally, category assignment is then achieved by means of any classification algorithm through utilization of the shape of the histogram. Support vector machine (SVM) is used in this work.

4.3 Support Vector Machines (SVM)

The support vector machine (SVM) is a supervised learning method that analyzes data and recognizes patterns, used for classification and regression analysis. It was firstly proposed by Vladimir Vapnik in 1963 [113] as a linear classifier. Later on, a novel algorithm was developed for creating nonlinear classifier by applying the kernel trick which implicitly maps its inputs into high dimensional feature spaces and the soft margin concepts [114] [115].

The main problem encountered in a typical linear classifier is that the data cannot be generally separated linearly. In order to overcome this challenge, SVMs as a classifier rely on preprocessing of the data to represent patterns in a high dimension feature space. The aim of this process is to make the data separable in the new high dimensional feature space. With an appropriate nonlinear mapping function $\varphi(.)$ to a sufficiently high dimension, data from two categories can always be separable by a hyperplane or hyperplanes [42]. Here, it assumed that each sample $\boldsymbol{x_k}$ from the data has been transformed to $\boldsymbol{y_k} = \varphi(\boldsymbol{x_k})$. For each of the n samples, $k = 1,2,\dots,n$, we let $z_k = \pm 1$ with respect to the class that the sample $\boldsymbol{x_k}$ belongs to. Then, a linear discriminant in augmented $\boldsymbol{y}$ space becomes

$$g(\boldsymbol{y}) = \boldsymbol{a^t y} \tag{4.2}$$

where both the weight vector and the transformed sample vector are augmented (by $a_0 = \omega_0$ and $y_0 = 1$, respectively). Hence, a separating hyperplane guarantees

$$z_k.\, g(\boldsymbol{y_k}) \geq 1, \;\; k = 1,2,\dots,n \tag{4.3}$$

The margin is any positive distance from the decision hyperplane. The aim in the training of SVM classifier is to determine the separating hyperplane(s) which make the margin larger as much as possible because larger margin provides a better generalization of the classifier. The distance from any hyperplane to a transformed pattern $\boldsymbol{y}$ is measured as $|g(\boldsymbol{y})|/\|\boldsymbol{a}\|$ and with an assumption that a positive margin b exists, equation (4.3) expresses

$$\frac{z_k g(\boldsymbol{y_k})}{||\boldsymbol{a}||} \geq b, \qquad k = 1,2,\dots,n \tag{4.4}$$

The goal is to compute the weight vector $\boldsymbol{a}$ which maximizes the margin b. The solution vector can become one of many arbitrarily scaled versions of it that still preserve the hyperplane. Therefore, the constraint in (4.5) is imposed in order to assure the uniqueness of the solution. Thus, $||\boldsymbol{a}||$ is tried to be minimized with its constraint.

$$b||\boldsymbol{a}|| = 1 \tag{4.5}$$

The *support vectors* are the transformed training samples which satisfies the condition $z_k . g(\boldsymbol{y_k}) = 1$. This means that the support vectors are equally close to the hyperplane and are the closest training samples. It can be observed from the illustration in Figure 27. Moreover, the support vectors are the training patterns that describe the optimal separating hyperplane and are the hardest patterns to categorize. Informally, they are the most informative samples for the classification task.

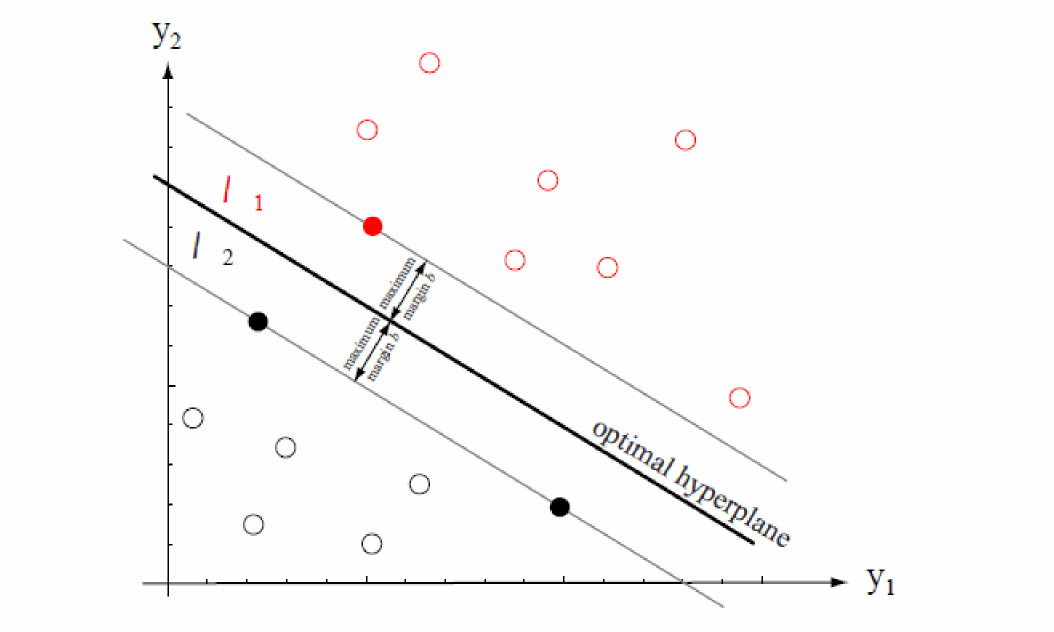


Figure 27: An illustration of optimal hyperplane and support vectors [42]

It is the most critical issue to assign a transformation function that separates the data so that the minimum number of support vectors satisfying maximum margin could be obtained which reduces the probability of misclassification. For this reason, the first step in training SVM is to select the nonlinear transformation functions, $\varphi(.)$, that map the input data into higher–dimensional space. The selection of this transform function usually depends on the characteristic of the problem domain. If there is no such information about the input data, one can use polynomials, Gaussian or other basis functions [42]. The dimensionality of the transformed space can be arbitrarily high, but in practice, it may be limited by computational resources. Some common kernel functions are listed as follows:

- Polynomial kernel (homogenous) : $k(\boldsymbol{y_j}, \boldsymbol{y_k}) = (\boldsymbol{y_j} \, . \, \boldsymbol{y_k})^d$
- Polynomial kernel (inhomogeneous) : $k(\boldsymbol{y_j}, \boldsymbol{y_k}) = (\boldsymbol{y_j} \, . \, \boldsymbol{y_k} + 1)^d$

- Gaussian Radial Basis kernel : $k(\boldsymbol{y_j}, \boldsymbol{y_k}) = \exp\left(-\gamma\|\boldsymbol{y_j} - \boldsymbol{y_k}\|^2\right)$

- Hyperbolic Tangent kernel : $k(\boldsymbol{y_j}, \boldsymbol{y_k}) = \tanh\,(\kappa \boldsymbol{y_j}\,.\,\boldsymbol{y_k} + c$

Based on experimental performance, Gaussian radial basis function is used in this study as a non-linear kernel, although the other selections could also be utilized.

The process of the computation of the optimal weight vector is started by recasting the problem of minimizing the magnitude of the weight vector constrained by the separation into an unconstrained problem via the method of Lagrange undetermined multipliers. Thus, the following function is constructed,

$$L(\mathbf{a}, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{a}\|^2 - \sum_{k=1}^{n} \alpha_k [z_k \mathbf{a}^{\boldsymbol{t}} \boldsymbol{y_k} - 1] \tag{4.6}$$

and seek to minimize $L(.)$ with respect to the weight vector $\mathbf{a}$ and maximize it with respect to the undetermined multipliers $\alpha_k \geq 0$. The last term in (4.6) specifies the aim of classifying the points correctly. This formulation can be reformulated as maximizing,

$$L(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{n} \alpha_i - \sum_{k,j}^{n} \alpha_k \alpha_j z_k z_j \boldsymbol{y_j}.\boldsymbol{y_k} \tag{4.7}$$

subject to the constraint,

$$\sum_{k=1}^{n} z_k\, \alpha_k = 0, \quad \alpha_k > 0, \qquad k = 1,2,\ldots,3 \tag{4.8}$$

These equations can be solved via quadratic programming optimization and so a vast number of schemes have been derived for this purpose. As a result, the solution can be expressed as a linear combination of the training vectors such that

$$\mathbf{a} = \sum_{i=1}^{\boldsymbol{n}} a_i z_i \boldsymbol{y}_i \qquad for \quad a_i > 0 \tag{4.9}$$

4.4 Performance Tests and Results

Experiments are conducted for two types of object classes (airplane and ship) in order to evaluate the performance of the proposed algorithm. The experiment for plane class is performed on a large set of airport images containing various kinds of airplanes. The data set for plane class consist of 23 Google Earth images of various sizes containing a total of 223 airplanes. For this experiment, Google Earth images are approximately adjusted to be 0.3 m - 0.5 m resolution by setting the eye altitude to the sum of the elevation terrain and 685 m of distance providing the desired resolution. Similarly, the test implemented for ship class is executed on a large set of Google Earth images which belong to different seaports in the world. These images contain 156 various ships different in terms of size, shape, function, type etc., and they are nearly adjusted to be 1.0 m resolution by means of the same procedure adopted plane class.

In order to evaluate the performance, precision-recall curves are exploited as a performance measure. A precision-recall curve is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied. These curves are highly informative about algorithm performance and provide a tool to select possibly optimal models and to reject suboptimal ones. The terminology used in the precision and recall rate calculation is defined in Chapter 2.1.5. Then, by using the related concept and the general description of precision and recall rates; in this context, recall can be considered as the proportion of correctly found the object number (true positives) over the total object number in the image (true positive +false negative). Similarly, the precision is the ratio of the number of correctly found objects (true positive) over the total number of found objects (true positive + false positive). In the light of this explanation, one can say that an ideal result should have a value equal to 1 for both recall and precision rates.

Derivation of the precision-recall curve by using an SVM classifier is a critical issue. The SVM classifier is defined as,

$$\boldsymbol{y_k} = \begin{cases} 1, & w\boldsymbol{x_k} - b > T \\ -1, & \boldsymbol{wx_k} - b < T \end{cases}, \quad k = 1,2,\ldots n \tag{4.10}$$

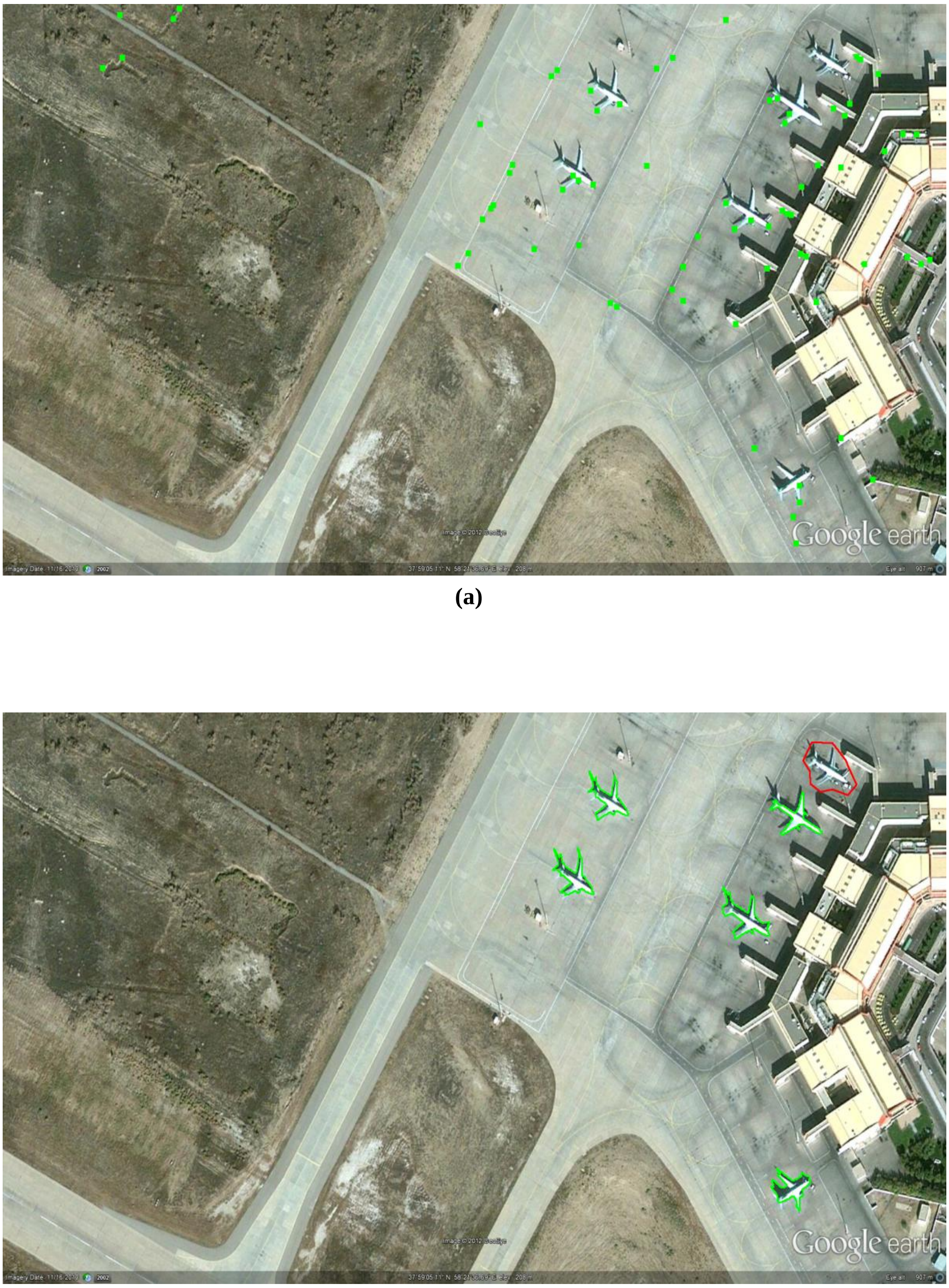
In classical SVM, T is equal to one as the classification condition. If T is chosen to be larger than the maximum of $(\boldsymbol{wx_k} - b)$ values in the data set, the classifier tends to classify all the data points in the feature space as negative class. Similarly, if it is chosen smaller than the minimum of $(\boldsymbol{wx_k} - b)$ values in the dataset, the all data points are classified as positive class. Therefore, by slowly varying the threshold T between these two extremes a precision vs. recall curve can be extracted by computing these rates for each value of threshold in the interval.

For each test image, its corresponding binary ground truth mask is prepared in order to count the number of true positives, false positives and false negatives. If the center of found object mask stays in its corresponding ground truth mask, that object is considered as true positive. Otherwise, it is a false positive. Furthermore, in performance evaluation, only one of the multiple detections falling inside the same masks are counted as true positive whereas the others are counted as false positives.

BoVW based object detection algorithm is utilized for the generation of hypothesis points during the experiments. The recall rate of the proposed algorithm highly depends on the performance of the algorithm used for hypothesis generation. Therefore, the recall value of the BoVW algorithm tries to increase by sacrificing of precision in order to obtain higher recall rate at the output. For the plane class test, 0.87 and 0.06 are the recall and precision rates, respectively. The ship class experiment uses the rates of 0.94 for the recall and 0.03 for the precision.

Figure 28-32 illustrate the proposed algorithm results for airplane class. The parts (a) in the figures show the hypothesis points (i.e. green dots) obtained from BoVW based object detection algorithm. The results of the proposed algorithm are represented by the boundary of the determined object mask. True positives, false alarms, and misses are symbolized with green, blue and red color, respectively. The results for ship class are shown with the similar representation manner in Figure 35-39. Moreover, Figure 34 and Figure 41 illustrate the recall vs. precision curves for airplane and ship classes, respectively. The best performance of the algorithm for airplane is obtained with the precision and recall values of 0.81 and 0.53, respectively. Likewise, the best performance for the ship class is measured as 0.73 and 0.6 for precision and recall rates, respectively.

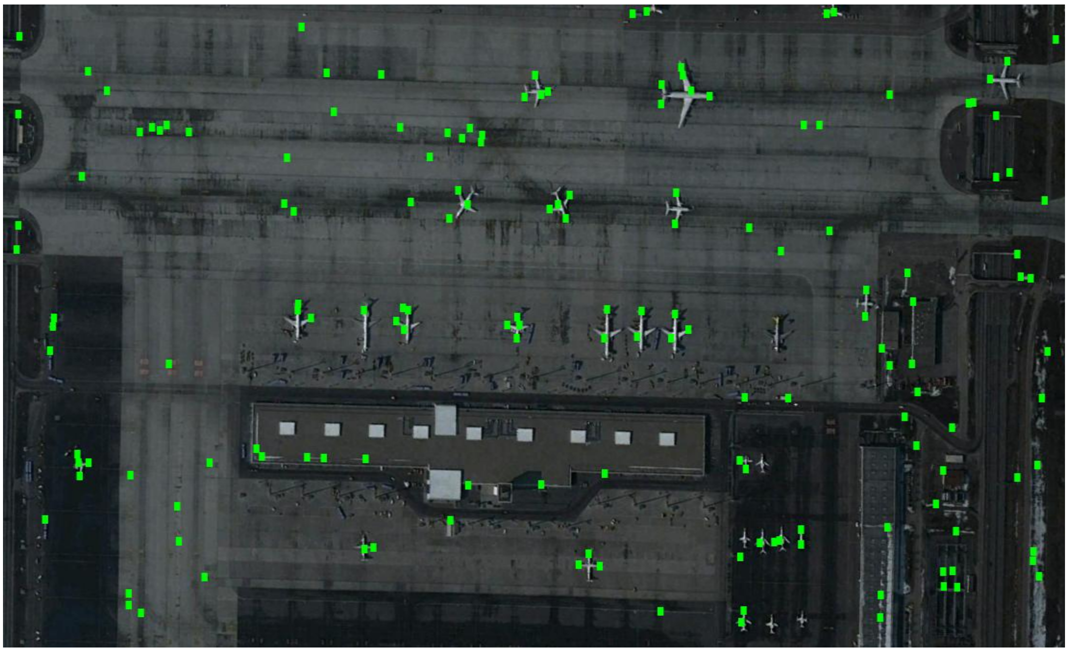
In order to analyze the effect of the last step, the elimination of multiple detections, on the overall system performance, the algorithm is executed for both classes without the last thresholding step. The resulting precision vs. recall curves are represented in Figure 33 for airplane and Figure 40 for ship categories. As it can be examined from the graphs, the precision values dramatically improves in both cases.



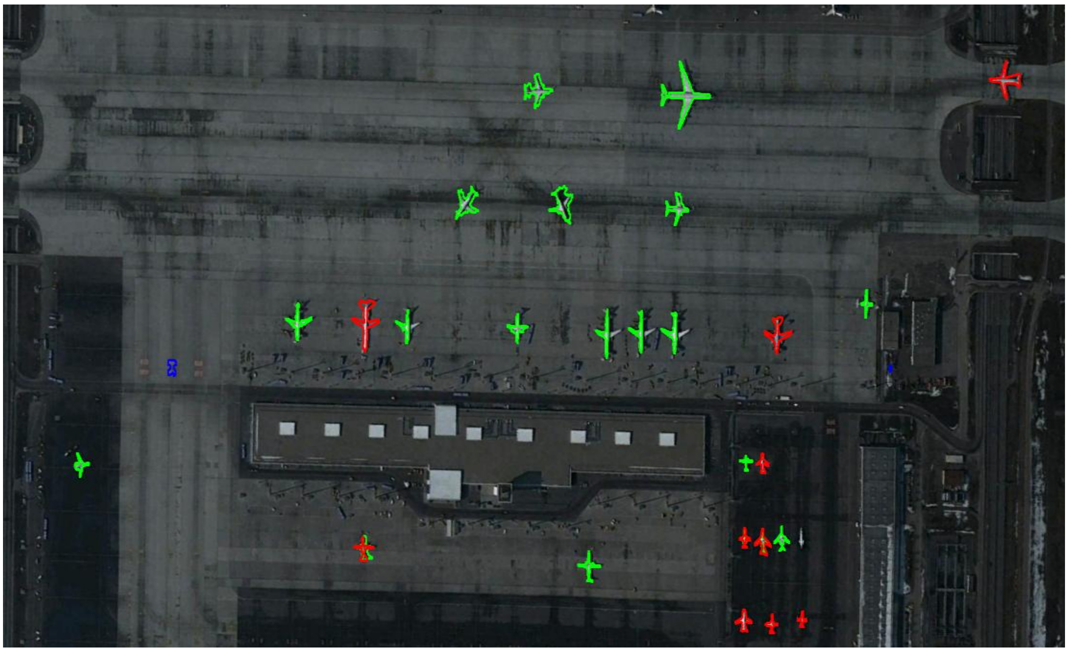
(a)

(b)

Figure 28: (a) Result of BoVW based detection algorithm (b) Results of the proposed algorithm for airplane

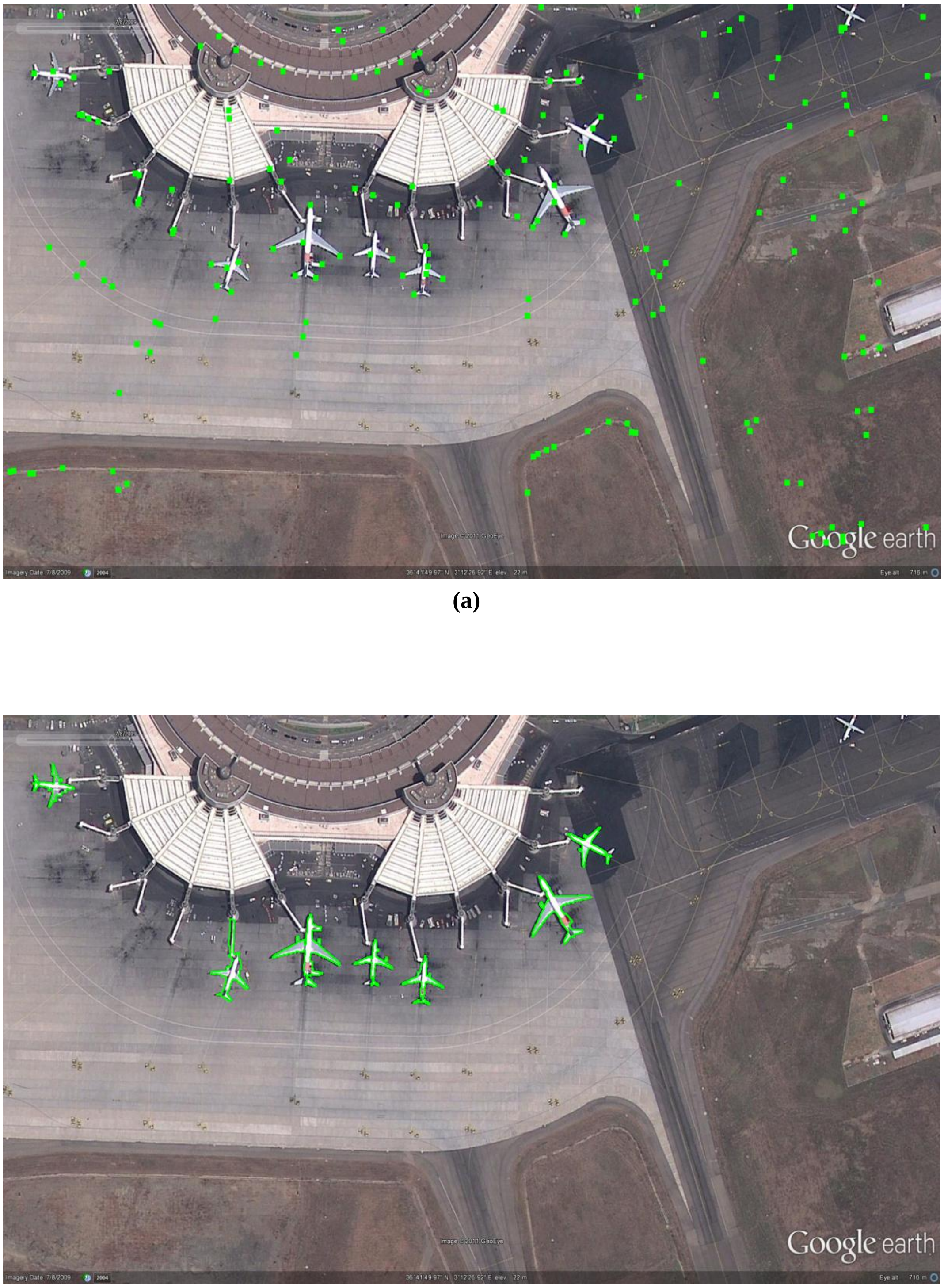


(a)



(b)

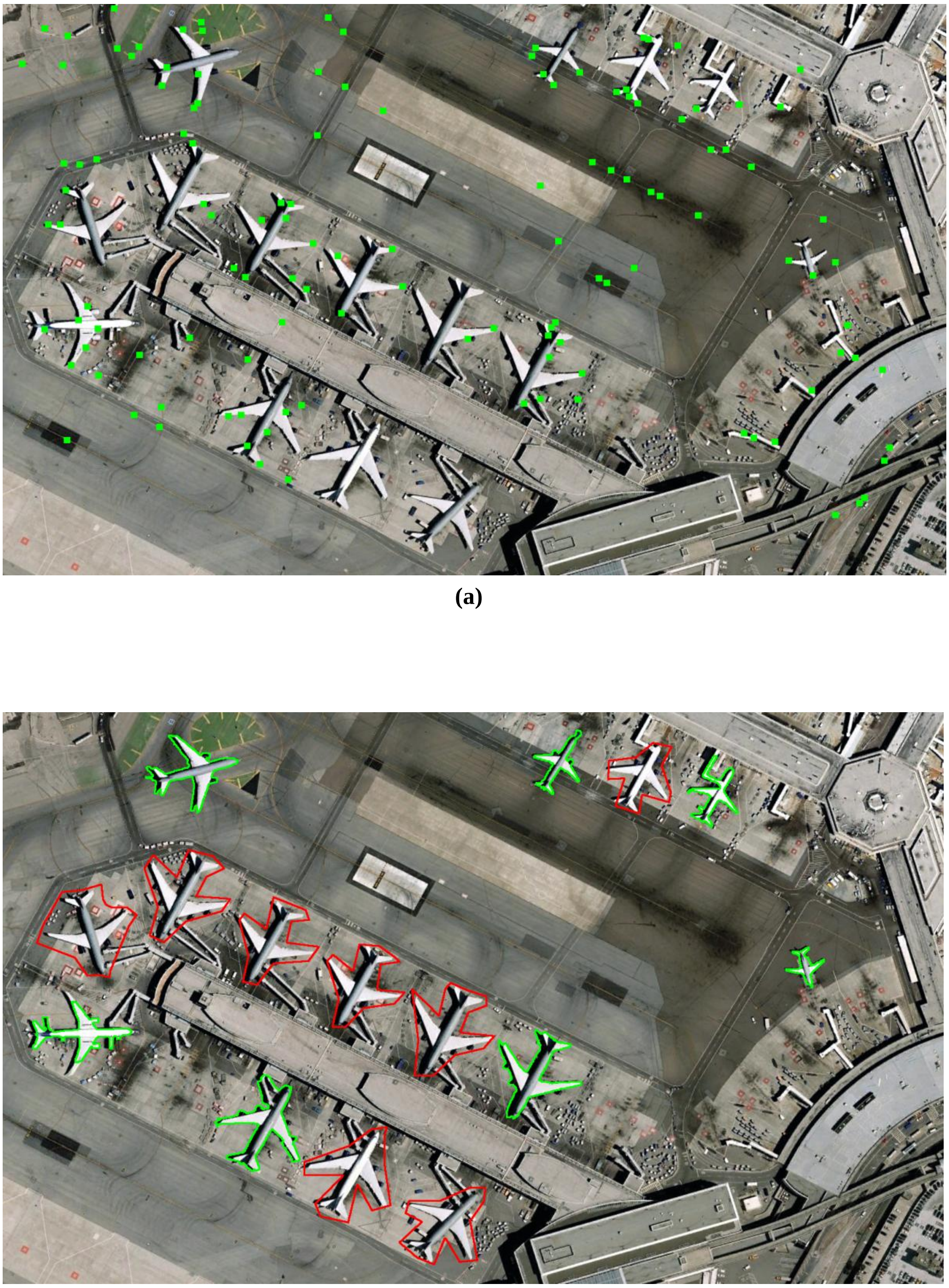
Figure 29: (a) Result of BoVW based detection algorithm (b) Results of the proposed algorithm for airplane



(a)

(b)

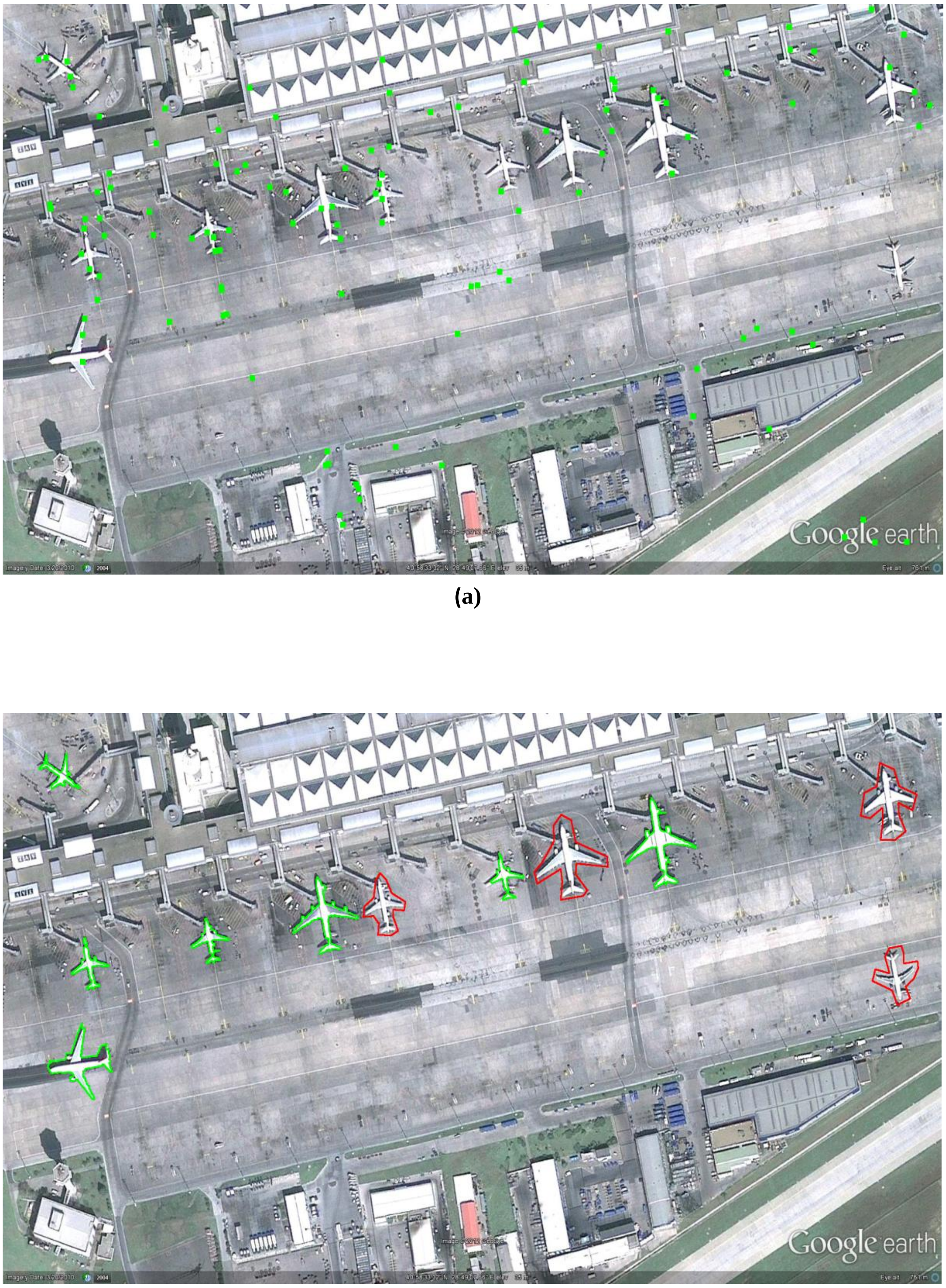
Figure 30: (a) Result of BoVW based detection algorithm (b) Results of the proposed algorithm for airplane



(a)

(b)

Figure 31: (a) Result of BoVW based detection algorithm (b) Results of the proposed algorithm for airplane



(a)

(b)

Figure 32: (a) Result of BoVW based detection algorithm (b) Results of the proposed algorithm for airplane

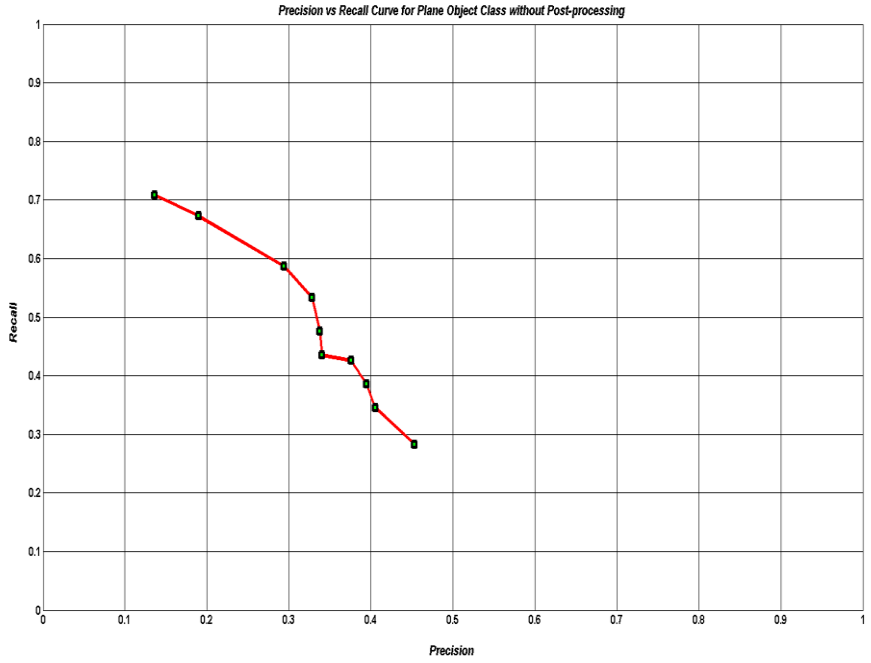


Figure 33: Precision vs. Recall curve without post-processing for airplane class

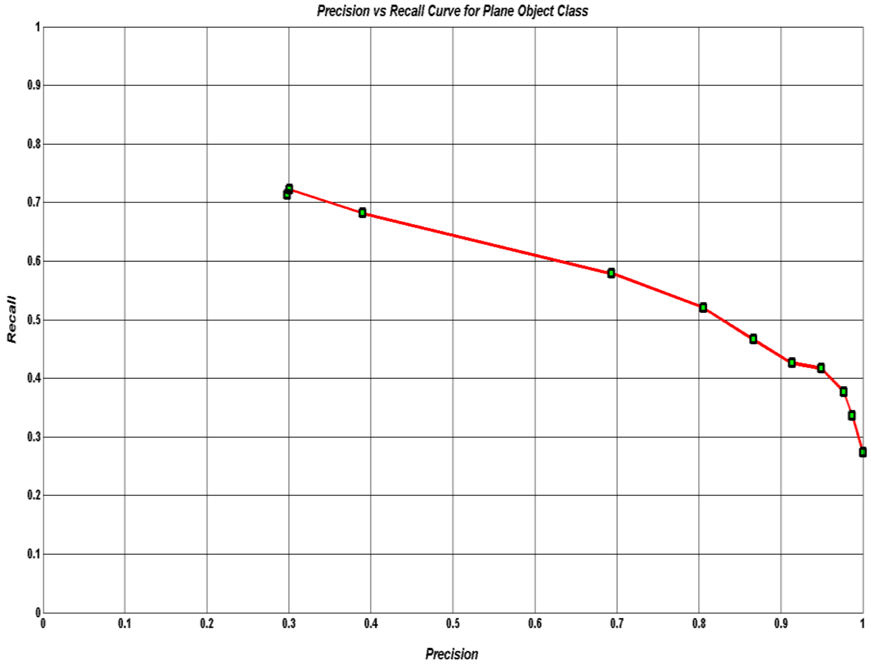
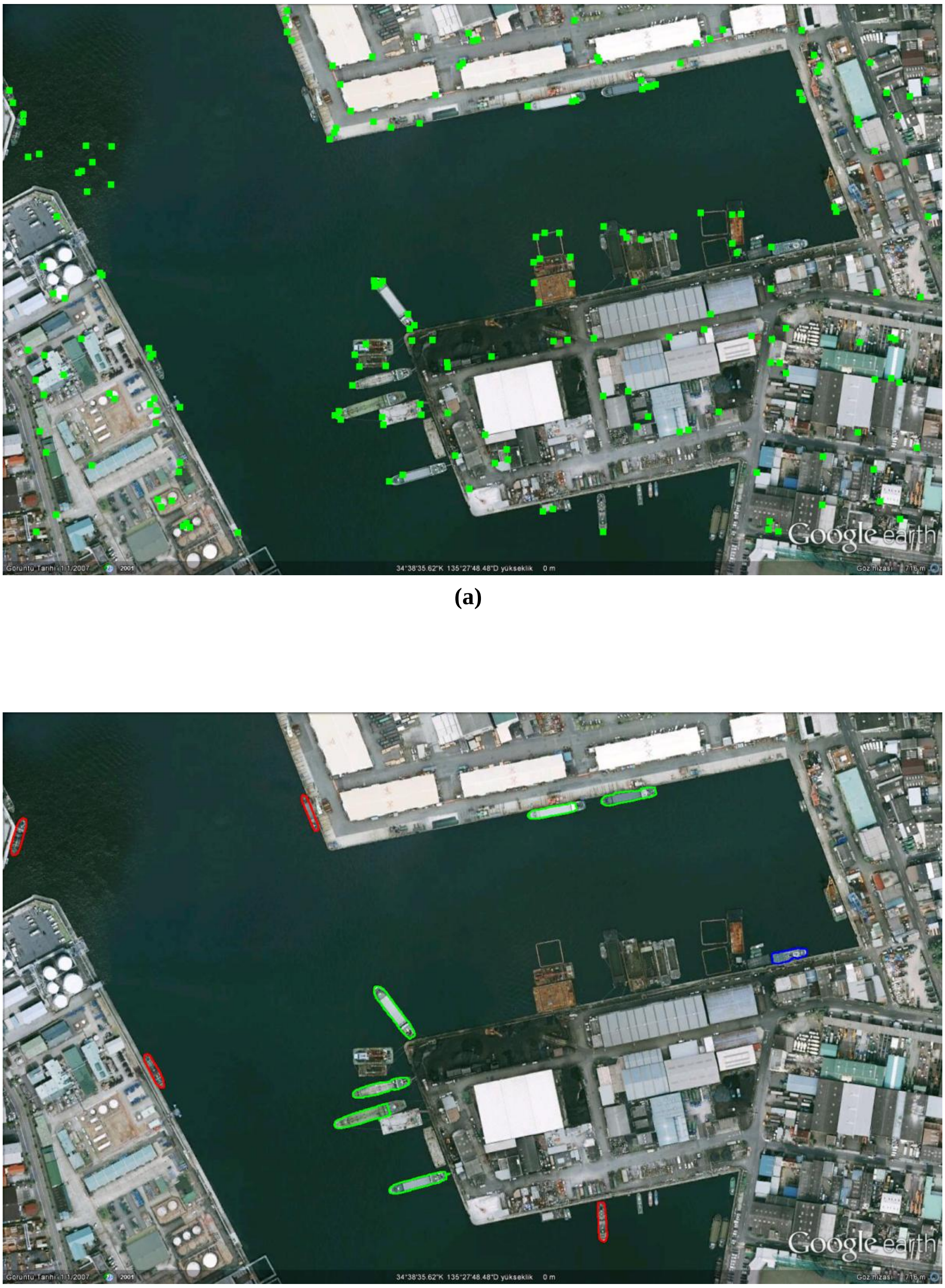


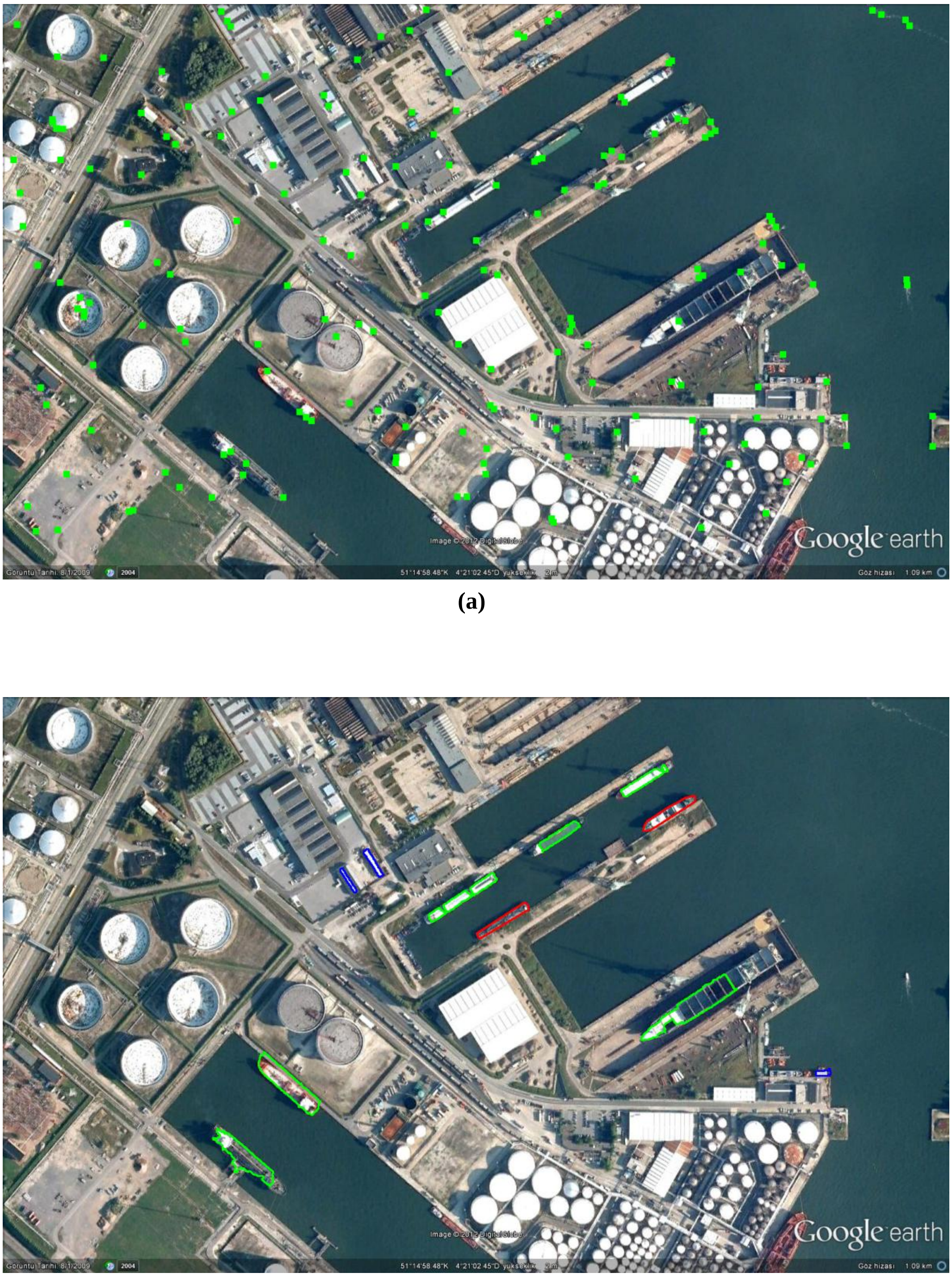
Figure 34: Precision vs. Recall curve for airplane class



(a)

(b)

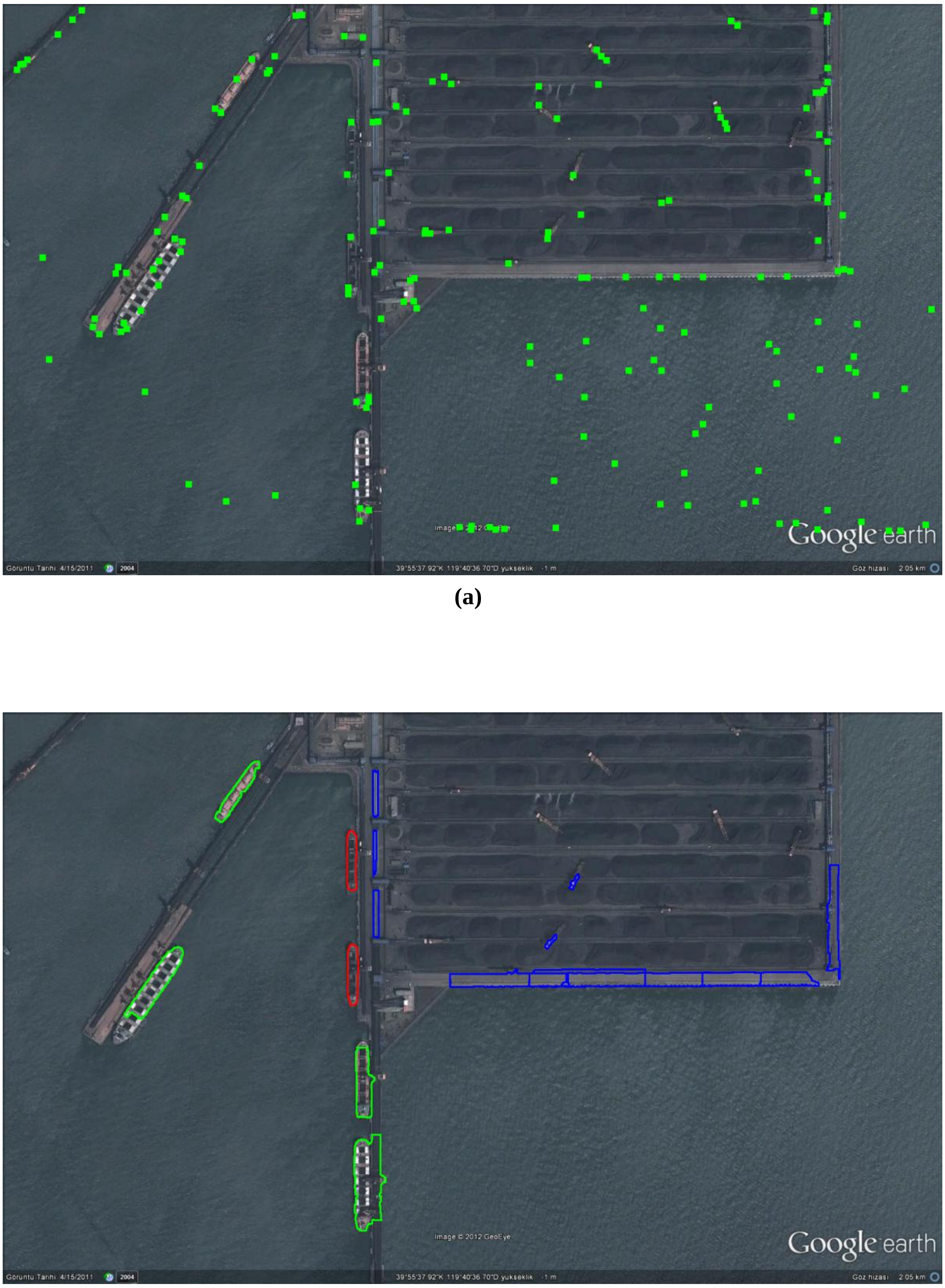
Figure 35: (a) Result of BoVW based detection algorithm (b) Results of the proposed algorithm for ship



(a)

(b)

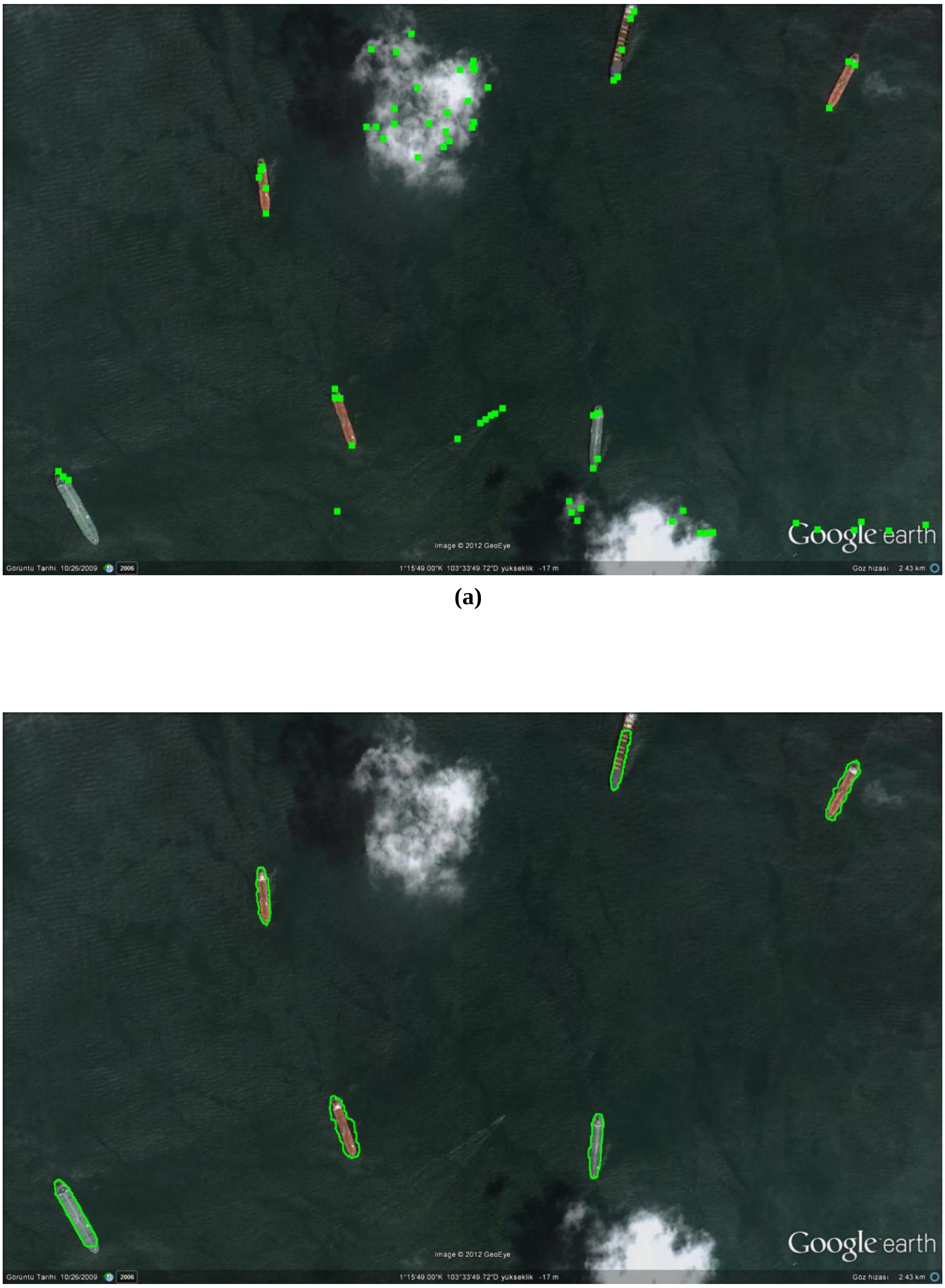
Figure 36: (a) Result of BoVW based detection algorithm (b) Results of the proposed algorithm for ship



(a)

(b)

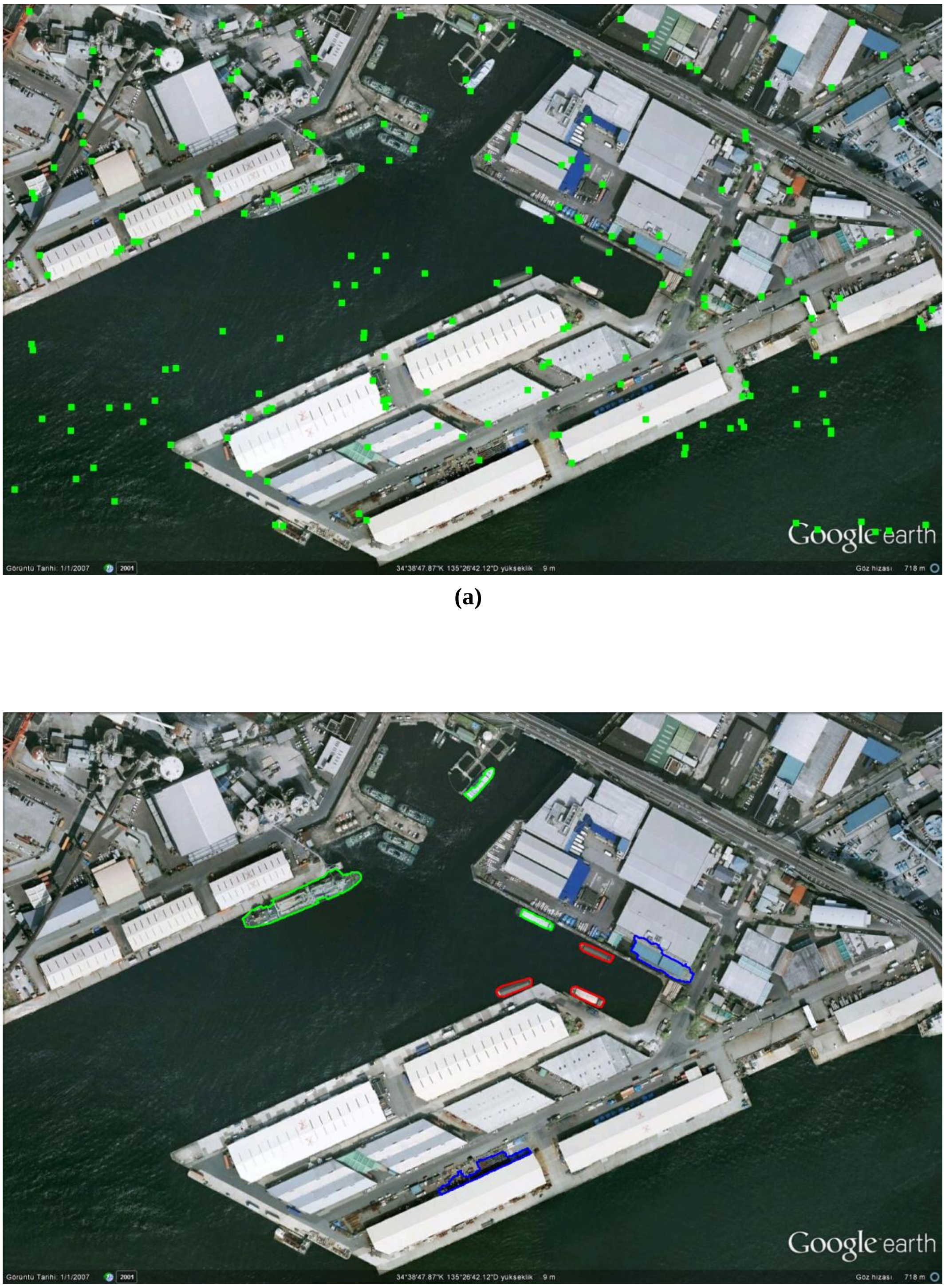
Figure 37: (a) Result of BoVW based detection algorithm (b) Results of the proposed algorithm for ship


Google earth
Image © 2012 GeoEye
Görüntü Tarihi: 10/26/2009
2006
1°15'49.00"K 103°33'49.72"D yükseklik -17 m
Göz hizası 2.43 km

(a)

(b)

Figure 38: (a) Result of BoVW based detection algorithm (b) Results of the proposed algorithm for ship



(a)

(b)

Figure 39: (a) Result of BoVW based detection algorithm (b) Results of the proposed algorithm for ship

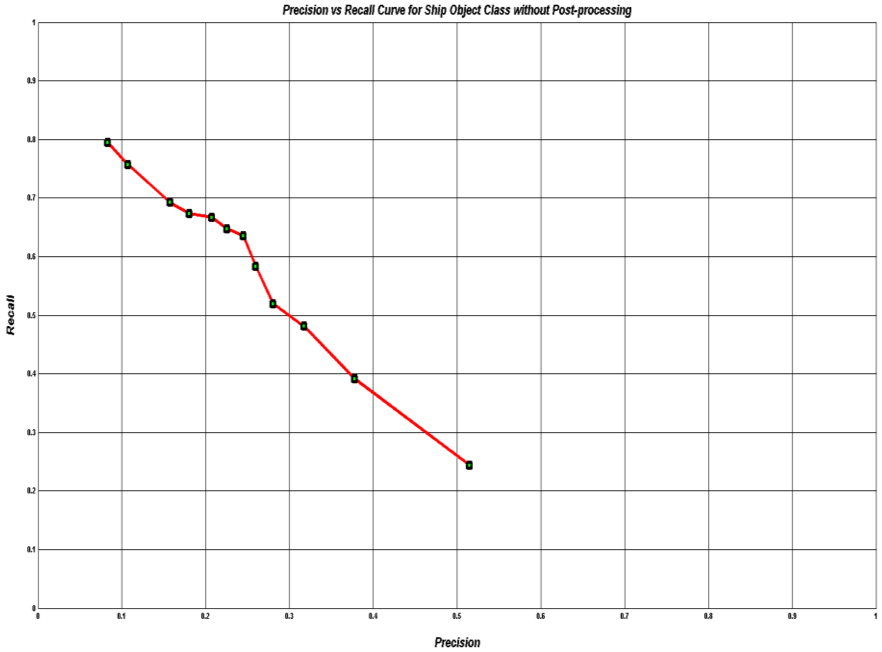


Figure 40: Precision vs. Recall curve without post-processing for ship class

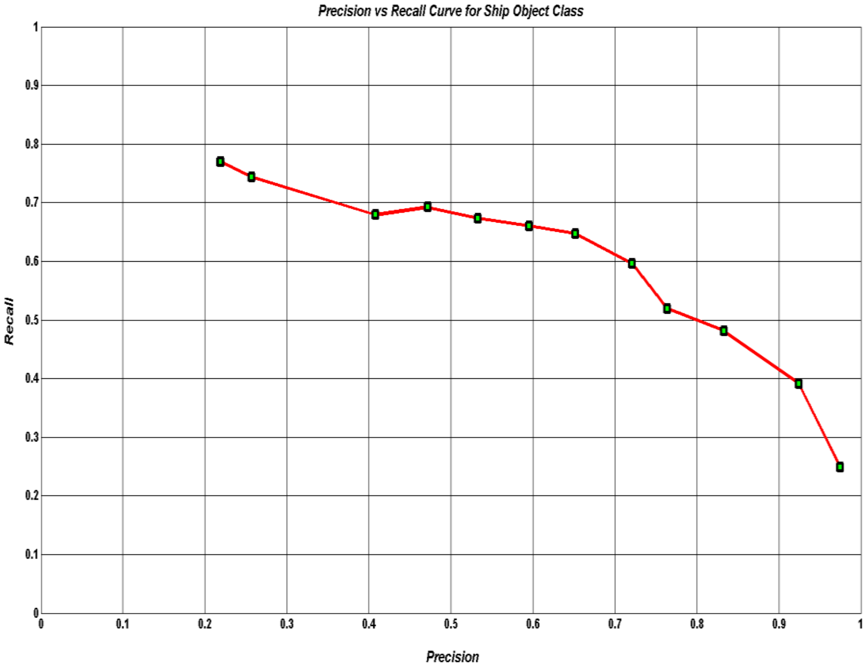


Figure 41: Precision vs. Recall curve for ship class

CHAPTER 5

CONCLUSION

5.1 Summary of Thesis

This thesis is devoted to the problem of geospatial object recognition from satellite imagery. An object recognition method is proposed to handle disadvantages of the classical object detection methods by exploiting the object shape characteristics.

The problem of image segmentation, which constitutes the key stage of the proposed system, is studied in Chapter 2. First of all, a detailed literature survey of image segmentation methods is presented with discussions about their strengths and weaknesses. Afterwards, normalized cut, k-means clustering and mean-shift methods for fully automatic image segmentation are investigated. Then, these algorithms are tested on several airplane and ship images and compared in terms of the performance of the foreground object extraction. Finally, because adequate efficiency cannot be acquired from these fully automated algorithms, multiple interactive or semi-automatic image segmentation techniques in the literature are researched. Interactive graph cut, GrabCut, and GrowCut image segmentation methods are analyzed within the context of this thesis. Their extraction performance and the user efforts needed for similar performance are compared in an objective and subjective manner in the final experimental result part.

Having established the foreground extraction, shape representation and description problem are discussed in Chapter 3. Initially, a comprehensive literature review for shape representation and description is introduced, and an organization of these techniques is given. Then, angular radial transform (ART) and geometric moment invariants for the region based methods and Fourier descriptor for contour based methods are analyzed in detail. In the final part, the performance comparison of these algorithms and all of their combinations is performed with NN and k-NN classification algorithms. For quantitative comparison, the confusion matrix, a specific table layout that allows visualization of the performance of any algorithm, is utilized.

In chapter 4, the proposed framework for geospatial object recognition is extensively explained. Next, the following sub-section mentions about the object detection algorithm, Bag of Visual Words, used in this thesis as hypothesis generation. After that, the classifier used in the proposed method, support vector machines (SVM), is reported in detail. Finally, the experimental results on two types of object classes, airplane and ship, are presented for the algorithm in a qualitative and quantitative approach.

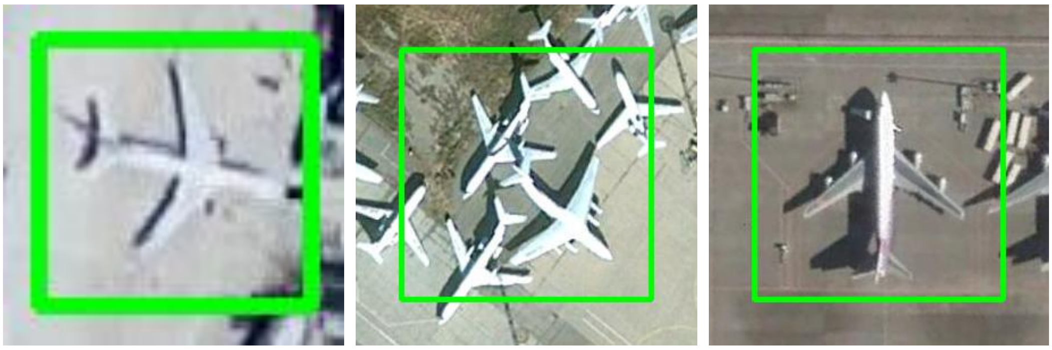
5.2 Discussions

Considering simulation results of the image segmentation algorithms presented in Chapter 2, it is obvious that the semi-automatic methods exhibit better performance than the fully automated ones in both objective (precision and recall values with ground truth) and subjective (visual inspection of the generated segments) evaluations. K-means clustering and normalized cut segmentation techniques in the fully automated does not produce smooth district boundaries overlapping with ground truth segment boundaries. Moreover, they depend on the number of clusters that should be determined beforehand. In the meantime; the fully automated mean-shift algorithm outperforms the previous two methods, and it relatively generates segmentation results which overlap with ground truth boundaries. However, the algorithm performance highly depends on the predefined parameter set such as minimum segment area, spatial and range bandwidth which should be precisely specified in order to obtain accurate and complete segmentation regions for each target image. Therefore, the target extraction with mean shift segmentation algorithm cannot be possible for the general case due to the parameter dependency and inadequacy of the method. On the contrary, interactive graph-cut and interactive GrowCut algorithms provide reliable segmentation outputs at the expense of high level user effort which is unfavorable for the proposed method. As a last method, interactive GrabCut outperforms the other fully and semi-automatic algorithms in terms of the quality of segmentation outputs and the desired user effort. Hence, it is selected as foreground extractor in the proposed method.

When shape description performances of ART, Hu moment invariants and Fourier descriptor methods are objectively evaluated in Chapter 3, it can be noticed that the integration of all of the three techniques outperforms the single methods and their different versions of the combination. The reason is that each shape descriptor method describes different 2D geometric properties and each one’s role are complementary for the description of the 2D shape. As shown in the experimental results in the Chapter 3, the retrieval performance of ART shape invariants increases at 2D shapes with complex structure, such as airplanes. Conversely, the Hu moment invariants efficiently describe elementary 2D geometric shapes, such as boats, unlike the complex shapes. On the other hand, the contour based shape invariants, Fourier descriptors, have powerful discrimination ability between the arbitrary shaped mask and the target objects. As a result, the integrated version of all these algorithms displays the properties of individual methods. Furthermore, these methods are robust to segmentation noises.

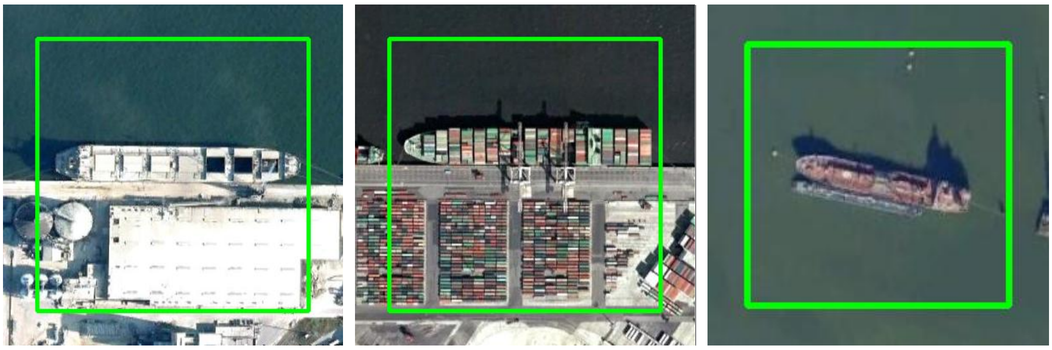
Experimental results indicate that the proposed geospatial object recognition algorithm is effective and promising. When the object masks are accurately and completely obtained, the false positives produced by any typical object detection methods can be successfully pruned for various object classes having characteristic shape. At the same time, the locations of target objects in the image are able to be determined exactly and accurately. Nevertheless, the size of the input rectangle used for specifying the foreground object should carefully be selected to be able to reproduce the entire object mask. The drawn rectangle should loosely encapsulate the target. Therefore, the rectangle size is nearly taken as two times of standard target object size.

Even though the proposed algorithm provides a great deal of performance increase in terms of the precision rate, there are still some problems from the viewpoint of the recall rate. These problems usually arise from the extraction of the foreground object mask step in the proposed system. One of the main problems related to image segmentation is about the foreground and background color distribution. This problem can generally occur in three different conditions [63]: (i) regions of low contrast at the transition from background to foreground (ii) camouflage, in which the true foreground and background distribution overlap partially in color space (iii) background material inside the user rectangle happens not to be adequately represented in the background region. For the third case, shadow is the most notable example excessively encountered in the real scenarios. Some examples for three cases are illustrated for aircraft and ship samples in Figure 42 and Figure 43, respectively.



(a) (b) (c)

Figure 42: Some illustration of the encountered problems in airplane class. (a) Regions of low contrast at transition (b) Camouflage (c) Background material inside the rectangle not enough represented in the background region.



(a) (b) (c)

Figure 43: Some illustration of the encountered problems for ship class. (a) Regions of low contrast at transition (b) Camouflage (c) Background material inside the rectangle not enough represented in the background region.

When the experimental results for object recognition in the aircraft and ship categories are examined, it can be seen that the precision rates of the airplane class are much better than the ship type' ones. The reason of this failure in the ship object is about the inadequacy of the shape representation methods in the description of the ship class. In other words, the ship masks and other masks extracted from rectangle shaped buildings, piers, etc. cannot sufficently be discriminated each other because of the closeness of their feature vectors in the feature space. Therefore, many places, such as building, pier, and rectangle field in the image can be recognized as a ship object at the end of the algorithm and so these false alarms reduce the precision rate significantly.

5.3 Future Work

The shadow can generate big troubles during the object extraction and these problems can be handled in several manners. Firstly, a shadow detection and restoration technique may be employed as a preprocessing step before segmentation algorithm. However, this idea can be discussed in terms of computational cost. Moreover, the performance of the shadow restoration algorithm considerably depends on the shadow detection part, which is still an unsolved problem in the literature. Another solution to be proposed is that, after shadow detection, the foreground extraction process is implemented by marking the found shadow pixels as background initially. In addition, this idea can be generalized into general background regions, e.g. sea, airfield, and port, existing in many cases by learning the backgrounds with GMM modeling technique.

The discrimination problem of the ship class can be tried to be solved by utilizing a cascade classifier. The first classifier is trained to separate the ship and similar masks from arbitrary shaped object masks. Then, the second classifier strictly distinguishes ship masks from the rectangular object masks.

REFERENCES

[1] Ç. Aytekin, "Geo-spatial Object Detection using Local Descriptors," Master of Science Thesis, METU, 2011.

[2] D. Arslan, "GLOBAL APPEARANCE BASED GEO-SPATIAL OBJECT DETECTION," Master of Science Thesis, METU, 2012.

[3] P. M. Roth and M. Winter, "Survey of Appearance-based Methods for Object Recognition," Technical Report, Inst. for Computer Graphics and Vision Graz University of Technology, Austria, 2008.

[4] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, no. 10, pp. 1615-1630, 2005.

[5] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky and L. V. G. T. Kadir, "A comparison of affine region detectors," *International Journal of Computer Vision,* vol. 65, no. 1/2, pp. 43-72, 2006.

[6] X. Sun, H. Wang and K. Fu, "Automatic Detection of Geospatial Objects Using Taxonomic Semantics," *IEEE Geosience Remote Sensing Letters,* vol. 7, no. 1, pp. 23-27, 2010.

[7] C. Tao, Y. Tan, H. Cai and J. Tian, "Airport Detection From Large IKONOS Images Using Clustered SIFT Keypoints and Region Information," *IEEE Transaction on Geoscience and Remote Sensing Letters,* vol. 8, pp. 23-27, 2011.

[8] S. Sahli, Y. Ouyang, Y. Sheng and D. A. Lavigne, "Robust Vehicle Detection in Low-Resolution Aerial Imagery," in *Proceedings of SPIE,*

2010.

[9] K. Rainey and J. Stastny, "Object recognition in ocean imagery using feature selection and compressive sensing," in *Applied Imagery Pattern Recognition Workshop (AIPR)*, 2011.

[10] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience,* vol. 3, no. 1, pp. 71-86, 1991.

[11] P. Viola and M. J. Jones, "Robust Real–Time Face Detection," *International Journal of Computer Vision,* vol. 57, no. 2, p. 137–154, 2004.

[12] C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *International Journal of Computer Vision,* vol. 38, no. 1, p. 15–33, 2000.

[13] B. Heisele, T. Serre, S. Mukherjee and T. Poggio, "Feature Reduction and Hierarchy of Classifiers for Fast Object Detection in Video Images," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition,* vol. 2, pp. 18–24,, 2001.

[14] X. Perrotton, M. Sturzel and M. Roux, "Automatic Object Detection On Aerial Images Using Local Descriptors And Image Synthesis," *Computer Vision Systems, Lecture Notes in Computer Science,* pp. 302-311, 2008.

[15] H. Cai and Y.Su, "Airplane Detection in Remote Sensing Image with a Circle-frequency Filter," in *Proceedings of the SPIE,* 2005.

[16] J. W. Hsieh, J. M. Chen, C. H. Chuang and K. C. Fan, "Aircraft Type Recognition in Satellite Images," *Processing of IEEE Vision, Image and Signal Processing,* vol. 152, no. 3, pp. 307-315, 2005.

[17] L. Eikvil, L. Aurdal and H. Koren, "Classification-Based Vehicle Detection in High-Resolution Satellite Images," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 64, no. 1, pp. 65-72, 2009.

[18] S. Bo and Y. Jing, "Region-based Airplane Detection in Remotely Sensed Imagery," in *International Congress on Image and Signal Processing* , 2010.

[19] J. Iisaka and T. Amano, "A Shape-based Object Recognition for Remote Sensing," in *Geoscience and Remote Sensing Symposium, IGARSS*, 2005.

[20] Y. Li, X. Sun, H. Wang, H. Sun and X. Li, "Automatic Target Detection in High-Resolution Remote Sensing Images Using a Contour-Based Spatial," *IEEE Geoscience and Remote Sensing Letters,* vol. 9, no. 5, pp. 886-890, 2012.

[21] K. Murphy, A. Torralba and W. Freeman, "Using the Forest to See the Trees: A Graphical Model Relating Features, Objects, and Scenes," in *In Advances in Neural Info. Proc. Systems*, 2003.

[22] K. Murphy, A. Torralba, D. Eaton and W. Freeman, "Object detection and localization using local and global features," in *Towards Category-Level Object Recognition*, 2005.

[23] L. Wang, J. Shi, G. Song and I.-f. Shen, "Object Detection Combining Recognition and Segmentation," in *Proceeding of the 8th Asian conference on Computer vision*, 2007.

[24] D. Ramanan, "Using segmentation to verify object hypotheses," in *CVPR*, 2007.

[25] I. Kokkinos, P. Maragos and A. Yuille, "Bottom-up & top-down object detection using primal sketch features and graphical models," in *CVPR*, 2006.

[26] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, New York: John Wiley&Sons, 1973.

[27] K. Fu and J. Mui, "A survey on image segmentation," *Pattern Recognition,* vol. 13, no. 1, pp. 3-16, 1981.

[28] R. Haralick and L. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics and Image Processing,* vol. 29, no. 1, pp. 100-132, 1985.

[29] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition,* vol. 26, no. 9, pp. 1277-1294, 1993.

[30] T. Zuva, O. O. Olugbara, S. O. Ojo and S. M. Ngwira, "Image Segmentation, Available Techniques, Developments and Open Issues," *Canadian Journal on Image Processing and Computer Vision,* vol. 2, no. 3, pp. 20-29, 2011.

[31] L. Spirkovska, "A Summary of Image Segmentation Techniques," NASA Technical Memorandum, California, 1993.

[32] D. D. Martin, C. C. Fowlkes, D. Tal and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE Int. Conf. on Computer Vision,* Vancouver, 2001.

[33] J. Freixenet, X. Mu˜noz, D. Raba and X. Cufi, "Yet Another Survey on Image Segmentation:Region and Boundary Information Integration," in *Proceeding of the European Conference on Computer Vision,* Copenhagen, 2002.

[34] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Addison-Wesley, 1993.

[35] Q. Wu and Y. Yu, "Two- Level Image Segmentation Based on Region and Edge Integration," in *Procedings of Digital Image Computing:Techniques and Applicaiton,* 2003.

[36] C. C. Chu and J. K. Aggarwal, "The Integration of Image Segmentation Maps Using Region and Edge Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, no. 12, pp. 1241-1252, 1993.

[37] J. Fan, Y. K. Yau, A. K. Elmagarmid and W. G. Aref, "Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing," *IEEE Transactions on Image Processing,* vol. 10, no. 10, pp. 1454-1466, 2001.

[38] S. Varshney, N. Rajpal and R. Purwar, "Comparative Study of Image Segmentation Techniques and Object Matching using Segmentation," in *International Conference on Methods and Models in Compuer Vision,* 2009.

[39] L. Lucchese and S. K. Mitra, "Color Image Segmentation: A State-of-the-Art Survey," *Proc. Indian Nat. Sci. Acad. (INSA-A),* Vols. 67-A, pp. 207 - 221, 2001 .

[40] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, 2000.

[41] S. Sarkar and P. Soundararajan, "Supervised Learning of Large Perceptual Organization: Graph Spectral Partitioning and Learning Automata," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 5, pp. 504-525, 2000.

[42] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, John Wiley& Sons, 2001.

[43] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 603- 619 , 2002.

[44] T. Kanungo, D. M. Mount, S. N. Netanyahu, C. D. Piatko, R. Silverman and A. Y. Wu, "An Efficient k-Means Clustering Algorithm : Analysis and Implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 7, pp. 881-892, 2002.

[45] S. Tatiraju and A. Mehta, "Image Segmentation using kmeans clustering, EM and Normalized Cuts," University of California, Irvine.

[46] C. M. Christoudia, B. Georgescu and P. Meer, "Synergism in Low Level Vision," in *Proc. Int. Conf. of Pattern Recognition*, Quebec City, 2001.

[47] I. Shimshoni, B. Georgescu and P. Meer, "Adaptive Mean Shift Based Clustering in High Dimensions," in *Computer*, 2003, pp. 456-475.

[48] C. Pantofaru and M. Hebert, "A Comparision of Image Segmentation Algorithms," Carnegie Mellon University Robotics Institute , Pennsylvania, 2005.

[49] [Online]. Available: http://en.wikipedia.org/wiki/Graph_theory. [Accessed

29 June 2012].

[50] W. Yair, "Segmentation using eigenvectors: a unifying view," in *Proceedings of the International Conference on Computer Vision*, 1999.

[51] Z. Wu and R. Leahy, "An optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 15, no. 11, pp. 1101-1113, 1993.

[52] M. Stoer and F. Wagner, "A simple minimum cut," *Journal of the ACM,* vol. 44, no. 4, pp. 585 - 591 , 1997.

[53] P. Soundararajan and S. Sarkar, "Analysis of MinCut, Average Cut, and Normalized Cut Measures," in *Proc. Third Workshop Perceptual Organization in Computer Vision*, 2001.

[54] G. H. Golub and C. F. Van Loan, Matrix Computations, John Hopkins Press, 1989.

[55] T. Cour, S. Yu and J. Shi, 2004. [Online]. Available: http://www.cis.upenn.edu/~jshi/software/. [Accessed 29 June 2012].

[56] *Adobe Photoshop 7 : Magic Wand,* Adobe Sytem Incorp., 2002.

[57] E. N. Mortensen and W. A. Barrett, "Interactive Segmentation with IntelligentScissors," *Graphical Models and Image Processing,* vol. 60, no. 5, p. 349–384, 1998.

[58] L. Grady and G. Funka-Lea, "Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials.," *ECCV Workshops*, p. 230–245, 2004.

[59] T. Heimann, M. Thorn, T. Kunert and H. P. Meinzer, "New methods for leak detection and contour correction in seeded region growing segmentation," *International Archives of Photogrammetry and Remote Sensing,* vol. XXXV, p. 317–322, 2004.

[60] Y. Y. Boykov and M. P. Jolly, "Interactive Graph Cuts for Optimal

Boundary & Region Segmentation of Objects in N-D Images," in *Proceedings of International Conference on Computer Vision*, Vancouver, 2001.

[61] Y. Li, J. Sun, C. K. Tang and H. Y. Shum, "Lazy Snapping," in *SIGGRAPH*, 2004.

[62] V. Vezhnevets and V. Konouchine, "“GrowCut” - Interactive Multi-Label N-D Image Segmentation By Cellular Automata," in *Proc. Graphicon*, 2005.

[63] C. Rother, V. Kolmogorov and A. Blake, ""Grabcut" - Interactive Foreground Extraction using Iterated Graph Cuts," in *Proc. ACM SIGGRAPG*, 2004.

[64] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision,* pp. 321-331, 1988.

[65] K. Liu, J. Tian, Y. Lu, C. Qin, X. Yang, S. Zhu and X. Zhang, "A fast bioluminescent source localization method based on generalized graph cuts with mouse model validations," *Virtual Journal for Biomedical Optics,* vol. 18, no. 4, pp. 3732-3745, 2010.

[66] Y. Boykov and V. Kolmogorov, "An Experimental Comparision of Min-Cut / Max-Flow Algorithms for Energy Minimizaiton in Vision," *IEEE Transactions Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1124- 1137 , 2004.

[67] J. F. Talbot and X. Xu, "Implementing Grabcut," 2006.

[68] J. Von Neumann, Theory of Self-Reproducing Automata, University of Illınouis Press, 1966.

[69] V. Kolmogorov and Y. Boykov. [Online]. Available: http://www.csd.uwo.ca/faculty/yuri/Abstracts/pami04-abs.html. [Accessed 29 06 2012].

[70] A. Amanatiadis, V. Kaburlasos, A. Gasteratos and S. Papadakis, "A comparative study of invariant descriptors for shape retrieval," *IST 2009 -*

International Workshop on Imaging Systems and Techniques, p. 391–394, 2009.

[71] M. Bober, F. Preteux and Y. M. Kim, "MPEG-7 Visual Shape Descriptors: Cover Sheet," Mitsubichi Electric, 2001.

[72] C. Martinez-Ortiz*, "2D and 3D Shape Descriptors" Phd thesis,* Department of Computer Science in University of Exeter, 2010.

[73] D. Zhang and G. Lu, "A Comparative Study of Three Region Shape Descriptors," in *DICTA2002: Digital Image Computing Techniques and Applications*, Melbourne, 2002.

[74] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition,* vol. 37, pp. 1-19, 2004.

[75] I. Yong, J. Walker and J. Bowie, "An analysis technique for biological shape," *Computer Graphics Image Processing,* vol. 25, pp. 357-370, 1974.

[76] E. R. Davies, Machine Vision: Theory, Algorithms, Practicalities, New York: Academica Press, 1997.

[77] P. J. van Otterloo, A contour-Oriented Approach to Shape Anaşysis, Englewood Cliffs: Prentice-Hall International, 1991.

[78] K. H., S. T. and P. M., "An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification," *Pattern Analysis and Machine Intelligence,* vol. 17, no. 2, p. 201–207, 1995.

[79] F. Mokhtarian and A. Machworth, "Scale-based description and recognition pf planar curves and two-dimensional shapes," *IEEE Pattern Analysis and Machine Intelligence,* vol. 8, no. 1, pp. 34-43, 1986.

[80] A. S., M. F. and K. J., "Curvature scale space image in shape similarity retrieval," *Multimedia Systems,* vol. 7, p. 467–476, 1999.

[81] F. Berrada, D. Aboutajdine, S. E. Ouatik and A. Lachkar, "Review Of 2D Shape Descriptors Based On The Curvature Scale Space Approach," in *Multimedia Computing and Systems (ICMCS),* 2011.

[82] R. Z. Roskies and D. S. Zahn, "Fourier descriptors for plane closed curves," *IEEE Transactions on Computers,* vol. 3, pp. 269-281, 1972.

[83] Eric, P. Fu and F. King-Sun, "Shape Discrimination Using Fourier Descriptors," *Pattern Analysis and Machine Intelligence,* vol. 8, no. 3, pp. 388- 397 , 1986.

[84] D. Zhang and G. Lu, "Comparision of Shape Retrieval using Fourier Descriptors and Short-time Fourier Descriptors," in *Pacific-Rim Conference on Multimedia,* Beijing, 2001.

[85] W. W. Boles and Q. M. Tieng, "Recognition of 2D object contours using wavelet transform zero-crossing representation," *IEEE Pattern Recognition and Machine Intelligence,* vol. 19, no. 7, pp. 910-916, 1997.

[86] H. S. Yang, S. U. Lee and K. M. Lee, "Recognition of 2D object contours using starting-point-independent wavelet coefficent matching," *Visual Communication Image Representation,* vol. 9, no. 2, pp. 171-181, 1998.

[87] H. Freeman, "On the encoding of the arbitrary geometric configurations," *IRE Trans. Electron. Comput. ,* vol. 10, pp. 260-268, 1961.

[88] W. J. Groskey and R. Mehrotra, "Index-based object recognition in pictorial data management," *Compuer Vision Graphics Image Process,* vol. 52, pp. 416-436, 1990.

[89] S. Berretti, A. D. Bimbo and P. Pala, "Retrieval by shape similarity with perceptual distance and effective indexing," *IEEE Transaction Multimedia,* vol. 4, pp. 225-239, 2000.

[90] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory,* vol. 8, no. 2, pp. 179-187, 1962.

[91] P. M. Liao S. X., "On image analysis by moments.," *Pattern Analysis and Machine Intelligence,* vol. 18, no. 3, pp. 254-266, 1996.

[92] C.-H. Teh and R. Chin, "On image analysis by the methods of moments," *Pattern Analysis and Machine Intelligence,* vol. 10, no. 4, pp. 496- 513 , 1988.

[93] T. M. R., "Image analysis via the general theory of moments," *J Opt Soc Am,* vol. 70, no. 8, pp. 920-930, 1980.

[94] M. Bober, "MPEG-7 VISUAL SHAPE DESCRIPTORS," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 11, no. 6, pp. 716- 719, 2001.

[95] D. Zhang and G. Lu, "Generic Fourier Descriptor for Shape-based Image Retrieval," *IEEE International Conference on Multimedia and Expo ICME '02.,* vol. 1, pp. 425- 428, 2002.

[96] G. Lu and A. Sajjanhar, "Region-based shape representation and similarity measure for content-based image retrieval," *Multimedia Systems,* vol. 7, no. 2, pp. 165-174, 1999.

[97] A. Goshtasby, "Description and discrimination of planar shapes using shape matrices," *IEEE Trans. Pattern Analysis and Machine Intelligence,* pp. 738-743, 1985.

[98] Y. Mingqiang, K. Kidiyo and R. Joseph, "A survey of shape feature extraction techniques," *Pattern Recognition,* pp. 43-90, 2008.

[99] L. N. A. S. a. P. S. A. Kadir, "A Comparative Experiment of Several Shape Methods in Recognizing Plants," *International Journal of Computer Science & Information Technology (IJCSIT),* vol. 3, no. 3, 2011.

[100] D. Zhang and G. Lu, "A Comparative Study of Curvature Scale Space and Fourier Descriptors for Shape-based Image Retrieval Abstract," *Journal of Visual Communication and Image Representation,* vol. 14, no. 1, p. 39–57, 2003.

[101] O. Terrades, S. Tabbone and E. Valveny, "A Review of Shape Descriptors for Document Analysis," *International Conference on Document Analysis and Recognition,* vol. 1, pp. 227- 231, 2007.

[102] S. Loncaric, "A survey of shape analysis techniques," *Pattern Recognition,* vol. 31, no. 8, p. 983–1001, 1998.

[103] A. Amanatiadis, V. Kaburlasos, A. Gasteratos and S. Papadakis,

"Evaluation of shape descriptors for shape-based image," *IET Image Processing*, 2011.

[104] H. Z., "Analysis of Hu's moment invariants on image scaling and rotation," in *International Conference on Computer Engineering and Technology (ICCET)*, 2010.

[105] L. Kotoulas and I. Andreadis, "An efficient technique for the computation of ART," *IEEE Trans. Circuits and Systems for Video Technolog*, vol. 18, no. 5, p. 682–686, 2008.

[106] D. Zhang and G. Lu, "A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures," *Journal of Visual Communication and Image Representation,* vol. 14, no. 1, pp. 41-60, 2003.

[107] R. Kohavi and F. Provost, "Glossary of Terms," *Machine Learning - Special issue on applications of machine learning and the knowledge discovery process,* vol. 30, no. 2-3, pp. 271-274, 1998.

[108] A. F. Smeaton, P. Over and W. Kraaij, "Evaluation campaigns and TRECVid," *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval (MIR),* pp. 321-330, 2006.

[109] M. Everingham, L. Gool, C. K. Williams, J. Winn and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision,* vol. 88, no. 2, p. 303–338, 2010.

[110] C. Harris and M. Stephens, "A combined corner and edge detector," *Proc. of the 4th Alvey Vision Conference,* p. 147–151, 1988.

[111] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[112] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision,* vol. 43, no. 1, pp. 29-44, 2001.

[113] V. Vapnik and A. Lerner, "Pattern recognition using generalized portrait method," *Automation and Remote Control*, p. 774–780, 1963.

[114] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning,* vol. 20, pp. 273-297, 1995.

[115] B. E. Boser, I. M. Guyon and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," *COLT '92: Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, p. 144–152, 1992.

[116] P. V. a. M. J. Jones, "Robust Real-Time Object Detection," in *Proceedings of IEEE Workshop on Statistical and Computational Theories of Vision*, 2001.

APPENDIX A

NORMALIZED CUTS FORMULATION

Let's assume that a graph, *V*, is partitioned into two disjoint sets *A* and *B*, and let *x* be an *N* = |*V*| dimensional indicator vector, such that x_i = 1 if node *i* is in A and x_i = -1 if otherwise. The total connection from node i to all other nodes is defined by $d(i)$ such that $d(i) = \sum_j w(i,j)$, where $w(i,j)$ indicate the link weight between node i and j. With the definitions of *x* and *d*, the normalized cut value can be rewritten as:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

$$= \frac{\sum_{(x_i>0,x_j<0)} -w_{ij}.x_i.x_j}{\sum_{x_i>0} d_i} + \frac{\sum_{(x_i<0,x_j>0)} -w_{ij}.x_i.x_j}{\sum_{x_i<0} d_i}$$

Let *D* be an *N x N* diagonal matrix whose entries are $d(i)$'s and *W* be an *N x N* symmetric matrix with *W (i, j) =* w_{ij} , then, define *k* as:

$$k = \frac{\sum_{x_i>0} d(i)}{\sum_i d(i)}$$

and $\bar{1}$ be an Nx1 vector of all ones. For $x_i > 0$ and $x_j < 0$, the indicator vector can be defined as $\frac{\bar{1}+x}{2}$ and $\frac{\bar{1}-x}{2}$, respectively. Thus, $Ncut(A,B)$ can be written as:

$$= \frac{1}{4}\ (\frac{(\bar{1}+x)^T(D-W)(\bar{1}+x)}{k\bar{1}^T D\bar{1}} + \frac{(\bar{1}-x)^T(D-W)(\bar{1}-x)}{(1-k)\bar{1}^T D\bar{1}}$$

$$= \frac{1}{4}(\frac{x^T(D-W)x + \bar{1}^T(D-W)\bar{1}}{k(1-k)\bar{1}^T D\bar{1}} + \frac{2(1-2k)\bar{1}^T(D-W)x}{k(1-k)\bar{1}^T D\bar{1}})$$

Let’s define the following auxiliary variables,

$$a(x) = \ x^T(D-W)x$$

$$\beta(x) = \bar{1}^T(D-W)x$$

$$\gamma = \bar{1}^T(D-W)\bar{1}$$

and

$$M = \bar{1}^T D\bar{1}$$

Then, the above equation can be expressed as follows:

$$= \frac{(a(x)+\gamma) + 2(1-2k)\beta(x)}{k(1-k)M}$$

$$= \frac{(a(x)+\gamma) + 2(1-2k)\beta(x)}{k(1-k)M} - \frac{2(a(x)+\gamma)}{M} + \frac{2a(x)}{M} + \frac{2\gamma}{M}$$

The last constant term is dropped since it is equal to zero in this case,

$$= \frac{(1-2k+2k^2)(a(x)+\gamma) + 2(1-2k)\beta(x)}{k(1-k)M} + \frac{2a(x)}{M}$$

$$= \frac{\frac{(1-2k+2k^2)}{(1-k)^2}(a(x)+\gamma) + \frac{2(1-2k)}{(1-k)^2}\beta(x)}{\frac{k}{1-k}M} + \frac{2a(x)}{M}$$

Setting $b = \frac{k}{1-k}$, then

$$= \frac{(1+b^2)(a(x)+\gamma) + 2(1-b^2)\beta(x)}{bM} + \frac{2ba(x)}{bM}$$

$$= \frac{(1+b^2)(a(x)+\gamma)}{bM} + \frac{2(1-b^2)\beta(x)}{bM} + \frac{2ba(x)}{bM} - \frac{2b\gamma}{bM}$$

$$= \frac{(1+b^2)(x^T(D-W)x + \bar{1}^T(D-W)\bar{1})}{b\bar{1}^T D\bar{1}} + \frac{2(1-b^2)\bar{1}^T(D-W)\bar{1}}{b\bar{1}^T D\bar{1}}$$

$$+ \frac{2bx^T(D-W)x}{b\bar{1}^T D\bar{1}} - \frac{2b\bar{1}^T(D-W)\bar{1}}{b\bar{1}^T D\bar{1}}$$

$$= \frac{\big((1+x)^T(D-W)(1+x)\big)}{b\bar{1}^T D\bar{1}} + \frac{b^2(1-x)^T(D-W)(1-x)}{b\overline{1b\ \sum_{x_\imath<0} d_\imath + b^2 \sum_{x_\imath<0} d_\imath}^T\ D\bar{1}}$$

$$- \frac{(2b(1-x)^T(D-W)(1+x))}{b\bar{1}^T D\bar{1}}$$

$$= \frac{[(1+x) - b(1-x)]^T(D-W)[(1+x) - b(1-x)]}{b\bar{1}^T D\bar{1}}$$

Setting $y = (1+x) - b(1-x)$, it is easy to see that

$$y^T D\bar{1} = \sum_{x_i>0} d_i - b \sum_{x_i<0} d_i = 0$$

Since $b = \frac{k}{1-k} = \frac{\sum_{x_i>0} d_i}{\sum_{x_i<0} d_i}$ and

$$y^T D\bar{1} = \sum_{x_i>0} d_i + b^2 \sum_{x_i<0} d_i$$

$$= b \sum_{x_i<0} d_i + b^2 \sum_{x_i<0} d_i$$

$$= b\left(\sum_{x_i<0} d_i + b \sum_{x_i<0} d_i\right)$$

$$= b\bar{1}^T D\bar{1}$$

By putting everything together, the normalized cut formulation can be expressed as:

$$Ncut(A,B) = \frac{y^T(D-W)y}{y^T D y}$$

with the condition $y(i) \in \{1, -b\}$ and $y^T D \bar{1} = 0$.

APPENDIX B

GRAPH CUT ALGORITHM

In the interactive graph cut segmentation algorithm, the main goal is to compute the global minimum of the cost function defining the soft constraints among all segmentation results that satisfy additional hard constraint imposed by a user. This computation can be carried out by finding the global minimum cost cut on a graph with two terminals. Below it is shown how the minimum cut $\hat{C}$ defines segmentation $\hat{A}$ and that this segmentation is optimal. Assume that $\mathcal{F}$ denotes a set of *feasible* cuts C on graph $\mathcal{G}$ such that

- C severs exactly one t-links at each p
- $\{p, q\} \in C$ iff p, q are t-linked to different terminals
- If $p \in \mathcal{O}$, then $\{p, T\} \in C$
- If $p \in \mathcal{B}$, then $\{p, S\} \in C$

Therefore, the minimum cut $\hat{C}$ on graph is *feasible* $(\hat{C} \in \mathcal{F})$ according to the definition of the *feasible* cut $\mathcal{F}$. Moreover, a unique corresponding segmentation $A(C)$ for any feasible cut $C \in \mathcal{F}$ can be defined such that

$$A_p(C) = \begin{cases} \text{"obj"}, & if\ \{p, T\} \in C \\ \text{"bkg"}, & if\ \{p, S\} \in C \end{cases} \tag{1}$$

In light of the abovementioned facts, a corresponding segmentation $\hat{A} = A(\hat{C})$ can be defined for the minimum feasible cut $\hat{C}$ and this segmentation output minimizes the cost function defined by (2.19) among all segmentation satisfying constraints in (2.24) and (2.25). The proof of the theorem can be explained as follows:

By using the table of edge weights defined in Table 7, definition of feasible cuts $\mathcal{F}$, and mathematical expression of the segmentation defined in (1), a cost of any $C \in \mathcal{F}$ is

$$|C| = \sum_{p \notin \mathcal{O} \cup \mathcal{B}} \lambda . R_p \left(A_p(C) \right) + \sum_{\{p,q\} \epsilon N} B_{\{p,q\}} \, . \, \delta \left(A_p(C), A_q(C) \right)$$

$$= E\big(A(C)\big) - \sum_{p \epsilon \mathcal{O}} \lambda . R_p \, (\text{"}obj\text{"}) - \sum_{p \epsilon \mathcal{B}} \lambda . R_p \, (\text{"}bkg\text{"})$$

Thus, $|C| = E\big(A(C)\big) - const(C)$. In fact, the equation (1) gives one-to-one correspondence between the set of all feasible cuts in $\mathcal{F}$ and the set $\mathcal{H}$ of all assignments A that satisfy hard constraint in (2.24) and (2.25). Then,

$$E\big(\hat{A}\big) = \big|\hat{C}\big| + const \; = \; \min_{C \in \mathcal{F}} |C| + const = \; \min_{C \in \mathcal{F}} E\big(A(C)\big) = \; \min_{C \in \mathcal{H}} E(A)$$

and this expression proves that the results of the algorithm are optimal.