DETECTION OF AIRPORT RUNWAYS IN OPTICAL SATELLITE IMAGES

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

BY

UĞUR ZÖNGÜR

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

JULY 2009

Approval of the thesis:

DETECTION OF AIRPORT RUNWAYS IN OPTICAL SATELLITE IMAGES

submitted by **UĞUR ZÖNGÜR** in partial fulfillment of the requirements for the degree of
Master of Science in Electrical and Electronics Engineering Department, 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. Uğur Halıcı
Supervisor, **Electrical and Electronics Engineering Department** ________________

Asst. Prof. Dr. İlkay Ulusoy
Co-supervisor, **Electrical and Electronics Engineering Department** ________________

Examining Committee Members:

Prof. Dr. Kemal LEBLEBİCİOĞLU
METU, EE ________________

Prof. Dr. Uğur HALICI
METU, EE ________________

Prof. Dr. Gözde B. AKAR
METU, EE ________________

Assoc. Prof. Dr. Şebnem DÜZGÜN
METU, GGIT ________________

Asst. Prof. Dr. İlkay ULUSOY
METU, EE ________________

Date: ________________

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

Name, Last Name:	UĞUR ZÖNGÜR
Signature :	

ABSTRACT

DETECTION OF AIRPORT RUNWAYS IN OPTICAL SATELLITE IMAGES

Zöngür, Uğur
M.S., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Uğur Halıcı
Co-Supervisor : Asst. Prof. Dr. İlkay Ulusoy

July 2009, 99 pages

Advances in hardware and pattern recognition techniques, along with the widespread utilization of remote sensing satellites, have urged the development of automatic target detection systems. Automatic detection of airports is particularly essential, due to the strategic importance of these targets. In this thesis, a detection method is proposed for airport runways, which is the most distinguishing element of an airport. This method, which operates on large optical satellite images, is composed of a segmentation process based on textural properties, and a runway shape detection stage. In the segmentation process, several local textural features are extracted including not only low level features such as mean, standard deviation of image intensity and gradient, but also Zernike Moments, Circular-Mellin Features, Haralick Features, as well as features involving Gabor Filters, Wavelets and Fourier Power Spectrum Analysis. Since the subset of the mentioned features, which have a role in the discrimination of airport runways from other structures and landforms, cannot be predicted, Adaboost learning algorithm is employed for both classification and determining the feature subset, due to its feature selector nature. By means of the features chosen in this way, a coarse representation of possible runway locations is obtained, as a result of the segmentation operation. Subsequently, the runway shape detection stage, based on a novel form of Hough Transform, is performed over the possible runway locations, in order to obtain final runway positions. The proposed algorithm is examined with experimental work using a comprehensive data set consisting of large and high resolution satellite images and successful results are achieved.

Keywords: Airport runway detection, Textural features, Segmentation, Adaboost, Hough transform

ÖZ

OPTİK UYDU GÖRÜNTÜLERİNDE HAVAALANI PİSTLERİNİN TESPİT EDİLMESİ

Zöngür, Uğur
Yüksek Lisans, Elektrik ve Elektronik Mühendisligi Bölümü
Tez Yöneticisi : Prof. Dr. Uğur Halıcı
Ortak Tez Yöneticisi : Asst. Prof. Dr. İlkay Ulusoy

Temmuz 2009, 99 sayfa

Donanım ve örüntü tanıma tekniklerindeki gelişmeler, uzaktan algılama uydularının yaygın kullanımı ile beraber, otomatik hedef tespit sistemlerinin geliştirilmesini teşvik etmiştir. Havaalanlarının otomatik olarak tespiti, bu hedeflerin stratejik önemlerinden ötürü, özellikle gereklidir. Bu tezde, bir havaalanının en ayırt edici öğesi olan havaalanı pistleri için bir tespit yöntemi önerilmiştir. Büyük optik uydu görüntüleri üzerinde çalışan bu yöntem, dokusal özelliklere dayanan bir bölütleme sürecinden ve bir pist şekli tespit etme safhasından oluşmaktadır. Bölütleme sürecinde, sadece görüntü yoğunluğunun ve gradyanının ortalaması ve standart sapması gibi düşük seviyeli öznitelikler değil, Zernike Momentleri, Dairesel-Mellin Öznitelikleri, Haralick Öznitelikleri, bunlara ek olarak, Gabor Süzgeçleri, Dalgacıklar ve Fourier Güç Tayfı Çözümlemesi gerektiren öznitelikleri de içeren, bir takım yerel doku öznitelikleri çıkarılmıştır. Bahsedilen özniteliklerin, havaalanı pistinin, diğer yapılardan ve yer şekillerinden ayırt edilmesinde rol oynayan alt kümesi öngörülemediğinden, Adaboost öğrenme algoritması, öznitelik seçici doğasından dolayı, hem sınıflama, hem de öznitelik alt kümesinin bulunmasında kullanılmıştır. Bu yolla seçilen öznitelikler vasıtasıyla, bölütleme işleminin bir sonucu olarak, muhtemel pist konumlarının kaba bir gösterimi elde edilmiştir. Ardından, nihai pist pozisyonlarını elde etmek amacıyla, yeni bir Hough Dönüşüm biçimine dayanan, pist şekil tespiti aşaması yürütülmüştür. Önerilen algoritma, büyük ve yüksek çözünürlüklü görüntülerden oluşan, kapsamlı bir veri kümesi kullanılarak, deneysel çalışma ile incelenmiş ve başarılı sonuçlar elde edilmiştir.

Anahtar Kelimeler: Havaalanı pist tespiti, Dokusal Öznitelikler, Bölütleme, Adaboost, Hough dönüşümü

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Prof. Dr. Uğur Halıcı for her valuable guidance, advices and support throughout this study.

I am also grateful to ASELSAN Inc. and my colleagues for their valuable support.

I would like to express my appreciation to Fırat PARLATAN for his helps in obtaining the data set and TUBITAK for their support.

Finally, I thank my dear family and Gülden KARTAL for their unconditional love, support and trust.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Airports are important structures from both economical and military perspective. Economically, as fundamental cargo and passenger transportation stations, airports serve to attract and retain businesses with national and global ties. Therefore, airports are a major force in the local, regional, national and global economy, becoming increasingly significant in terms of financial reasons.

The military airports, i.e. airbases, are also critical strategic targets considering the importance of the aviation branch of a nation's defense forces. Airbases are used for not only take-off and landing of crucial bomber and fighter units, but also consequential support operations such as strategic and tactical airlift, combat airdrop and medical evacuation, promoting the worth of airports. From this point of view, automatic detection of airports can provide vital intelligence to take well-timed military measures in a state of war.

The technological improvements on both computational hardware and pattern recognition techniques made identification of airports an attainable objective. Besides, increasing number of countries that have their own satellites renders the problem even more attractive, by the supplied unbiased data to investigate. These reasons form the motivation of this thesis.

1.1 Previous Work

There exists a number of prior studies on the problem of airport detection. In addition to the fact that the quantity of these works is not satisfying, the qualities of some are also disputable. However, several ideas in (Liu, 2004) and (Han, 2002) are found convenient, and applied in this thesis.

In (Liu, 2004), a method, which is established on a texture based pre-screening step to obtain Regions of Interests (ROIs), and subseqeuently an elongated rectangle detection on those ROIs for finding runways is proposed. Kernel Matching Pursuits is used for the classification process in the first step. For the second step a Hough Transform based shape detection is employed. The proposed method is intended to find airport runways in large optical imagery (approximately 6500 × 7500 pixels) in a computationally efficient way. Example results are given using a dozen high resolution aerial images from Southern California.

In (Han, 2002), a method is proposed for finding airports in a hypothesis formation and verification manner. Hypotheses are formed by acquiring parallel lines in opposite orientation using edge tracking procedures on the edge detected image. Hypothesis verification is done by judging gray homogeneity, average gray value and contrast with background. Experiments are performed using more than a hundred but small (256 × 200) and low resolution images.

In (Qu, 2005), first a search for the elongated rectangles pertaining to runways is carried out, generating runway hypotheses. Contrary to (Liu, 2004), textural properties are utilized for hypothesis verication instead of obtaining ROIs, using Support Vector Machine classifier.

In (Gupta, 2007), a primitive segmentation is carried out before utilizing Hough Transform based potential runway selection. Hough Transform is also employed in (Wang, 2007) and (Pi, 2003) for the detection of airports, however both of these papers lack of an experimental work.

Due to the fact that the runways can be considered as road segments, runway detection has some relevancy to road detection. The encountered road detection algorithms in the literature are either semi-automatic (Niu, 2006)(Udomhunsakul, 2004), which needs a seed provided with human interaction (i.e. a part of a road and generally the initial direction), or automatic (Mena, 2005)(Mokhtarzade, 2008), which operates without such an interaction. Since runways are lack of a complex network structure unlike roads, application of semi-automatic algorithms to runway detection would obviously be infeasable. Similar to automaticity, there are many concepts that are not suitable or unnecessary for runway detection like road tracking methods or snake (active contour) applications. However, concepts involving textures and segmentation can be utilized. There are studies that uses Haralick features (Mena, 2005), (Fernandez, 1992), (Popescu, 2008), which are textural features and also employed in this study. (Mena, 2003) and (Mayer, 2006) provides a good bibliography of previous work about road detection.

1.2 Structure of an Airport

Airport is a structure, built for the take-off and landing of aircrafts. Airports can be classified as civil and military, in terms of utilization purpose.

Civil airports generally have a runway or multiple runways, taxiways, passenger and freight terminals, a control tower, and other minor buildings such as passenger walkways or parking areas. There can be small civil airports constructed specifically for gliders or flying boats, which have a grass or no permanent runways at all.

Military airports generally consist of a runway or multiple runways, taxiways, hangars, barracks, control tower, defensive structures such as SAM (Surface to Air Missile) sites, and other minor buildings such as shelters, administration building, fire-fighting station and ammunition storage. Military airports built for jet aircrafts tend to have long runways while the airfields constructed for transport aircrafts generally have large hangars.

Some airports can be combined airports used for both civil and military purposes. There can also be temporary airfields or emergency airfields, which are primarily long straight highway segments used as runways.

Runway, as a fundamental airport component, is a long strip that provides the necessary distance for aircrafts to reach the required speed for take-off, and to safely land. An airport may have single runway or many runways placed in various configurations, such as parallel, crossing, non-crossing (non-parallel at the same time) or any combination of these. The landing surface, may be a natural surface such as grass or gravel, or more commonly a man-made surface. These man-made surfaces can be asphalt (e.g. Helicopter Airfields), concrete, concrete with antiskid (e.g. Jet Airfields) or even pierced planking in case of temporary runways.

1.3 Scope of Thesis

This thesis refers to an airport detection algorithm, which is intended to work on large and high resolution optical images. The concerned optical satellite images that carry chromatic information that lies on the visible spectrum by the human eye and contain neither infrared nor any other remote sensed data channels except RGB (Red, Green, Blue). These images are approximately 13626×10862 in size, having 1 meter resolution.

On an aerial image, the most prominent part of an airport is absolutely the runway. Runways have a more distinguishable form, share more characteristics with other runways and cover the most area when compared to other airport structures. Therefore it can be concluded that airport detection problem can be reduced and interpreted as airport runway detection problem. For the rest of the thesis, airport and runway detection is used alternately.

1.4 Contribution of Thesis

In this thesis, a runway detection algorithm is proposed. This algorithm, in a coarse manner, involves a classification process preceded by extraction of various features, and runway shape detection by means of a Modified Hough Transform operation. In the classification part, the regions of interests, which is a rough guess of areas where runways might be, are determined based on textural properties. Afterwards runways, which has a thin, long rectangle shaped structure, are searched in those Regions of Interest. The contributions of this thesis, apart from the proposed algorithm itself that is explained in Chapter 3, are mentioned below.

1.4.1 Features Novel to Runway Detection Problem

In (Liu, 2004), Basic Statistical Features, Zernike Moments and Circular-Mellin features are utilized. It is also stated that Haralick features are used in various studies on road detection. In addition to these features, as one of the contributions of this thesis, a set of features is employed for segmentation, which has not been utilized for airport detection in previous studies. Particularly, these are features extracted from HSV (Hue, Saturation, Value) color space, and features involving Fourier Power Spectrum, Gabor Filters and Wavelet Decomposition. These features are explained in Section 2.1 and 3.1.1 in detail.

1.4.2 Use of Adaboost Algorithm for Segmentation

Another contribution of this thesis is the application of Adaboost algorithm on runway detection problem. Adaboost is a boosting algorithm, which aims to improve the performance of any other learning algorithm. When used together with a set of weak learners, it selects the desired number of most beneficial ones. Since finding features that can represent the genuine characteristics of the runway texture is not a straightforward task, a strategy that exploits the aforementioned property of Adaboost is utilized. This strategy is established upon finding as many features that could probably serve for discriminating runway textures, and let Adaboost algorithm to judge and decide which features to use based on the effectiveness of them.

1.4.3 Modified Hough Transform

The third contribution of this study is an improvement for the Hough Transform for runway detection. The improvement is achieved by an approach, which takes advantage of the relationship of a runway and its background. Runways generally have same background on either long side of their elongated rectangle shape. This phenomenon results in opposite mean gradient directions on those lines, which is called antiparallel lines. While there exist methods that make use of gradient information with hough transform (O'Gorman, 1976)(Ballard, 1987), or using antiparallel lines without hough transform (Scher, 1982)(Lin, 1998), the employed form in this study is original. This proposed form leads to an opposite peak-pair pattern on Hough parameter space, which is an effective, elegant and descriptive representation of line pairs, i.e. runway side lines. The details of this contibution are given in Section 3.2.2.

1.5 Organization of Thesis

This thesis is organized as follows. In Chapter 2, the fundamental concepts, which are essential for the proposed algorithm, are covered. In Chapter 3, the details of the proposed airport runway detection algorithm are described. Experimental results, concerning the parameter selection and performance evaluation of the proposed algorithm, are given in Chapter 4. Finally, the discussion of the obtained results, future work and concluding remarks are given in Chapter 5.

CHAPTER 2

BACKGROUND

In this chapter, the fundamental subjects, used in this study, are explained. In Section 2.1, features, utilized in this study, are covered. The Adaboost algorithm, employed with these features is described in Section 2.2. The Hough transform, on which the proposed Modified Hough Transform is based, is explained in Section 2.3. In Section 2.4, Morphological Image Operations, like opening and dilation, are covered. Finally, in Section 2.5, the computation of Canny Edge Operator is given.

2.1 Features

The features used in this study are explained below. Throughout this section, the concerned image is represented as

$$f(x, y)$$

which is assumed to be $N \times N$ in size.

2.1.1 Basic Features

Utilized basic statistical texture features, which are also used in detection of airports by (Liu, 2004), are explained below.

Mean of intensity image which represents how bright the pixels are in a block on average is given in Equation 2.1.

$$mean_f = \frac{1}{N^2}\sum_x\sum_y f(x,y) \tag{2.1}$$

Standard deviation of intensity image which indicates how dispersed the gray values are, i.e. contrast, is given in Equation 2.2

$$var_f = \sqrt{\frac{1}{N^2}\sum_x\sum_y (f(x,y) - mean_f)^2} \tag{2.2}$$

Gradient magnitude estimate for the calculation of basic features is defined in the Equation 2.3

$$\|\nabla f(x,y)\| = |f(x,y) - f(x-1,y) + j(f(x,y) - f(x,y-1))| \tag{2.3}$$

Mean of gradient magnitude, which provides information about average rate of gray level change between neighboring pixels, is given in Equation 2.4

$$mean_{\|\nabla f\|} = \frac{1}{N^2}\sum_x\sum_y \|\nabla f(x,y)\| \tag{2.4}$$

Standard deviation of gradient magnitude, which shows how variable this rate of change is, is given in equation 2.5

$$var_{\|\nabla f\|} = \sqrt{\frac{1}{N^2}\sum_x\sum_y (\|\nabla f(x,y)\| - mean_{\|\nabla f\|})^2} \tag{2.5}$$

2.1.2 Zernike Moments

Zernike moments are image moments, which are used in rotation invariant recognition of images (Khotanzad, 1990). Unlike regular image moments, they use a set of orthogonal basis functions, introduced by Zernike, which form a complete orthogonal set inside the unit circle. Mapping of image function, onto Zernike polynomials, results in no redundancy between different Zernike moments, as opposed to regular image moments. The basis functions are in the form of Equation 2.6, i.e. Zernike Polynomials.

$$V_{n,m}(x,y) = V_{n,m}(r,\theta) = R_{n,m}(r)e^{jm\theta} \tag{2.6}$$

Table 2.1: Zernike Moments and their corresponding orders

Order	**Moments**	**Number of Moments**
0	$Z_{0,0}$	1
1	$Z_{1,1}$	1
2	$Z_{2,0}$, $Z_{2,2}$	2
3	$Z_{3,1}$, $Z_{3,3}$	2
4	$Z_{4,0}$, $Z_{4,2}$, $Z_{4,4}$	3

In this equation j denotes the imaginary unit, $\sqrt{-1}$, n is a non-negative integer, m is an integer, $n - |m|$ is even, $|m| \leq n$ and finally r and θ are the magnitude and the angle of the vector from origin to (x, y) point respectively where $x^2 + y^2 \leq 1$. In the same equation, $R_{n,m}(r)$ denotes the radial polynomial, which is defined as

$$R_{n,m}(r) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s!(\frac{n+|m|}{2} - s)!(\frac{n-|m|}{2} - s)!} r^{n-2s} \tag{2.7}$$

Given these polynomials, Zernike Moments are defined as in Equation 2.8.

$$Z_{n,m}(f) = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r,\theta) V_{n,m}^*(r,\theta) r dr d\theta \tag{2.8}$$

Here, $*$ denotes complex conjugate. Since $f(x, y)$ is a real signal and $R_{n,m}(r) = R_{n,-m}(r)$, complex conjugate of $Z_{n,m,f}$, (i.e. $Z_{n,m}^*(f)$), is equal to $Z_{n,-m}(f)$, .

For a digital image, the expression above becomes

$$Z_{n,m}(f) = \frac{n+1}{\pi} \sum_x \sum_y f(x,y) V_{n,m}^*(r,\theta) \tag{2.9}$$

For the computation, the origin is taken to be the center of the image f, and pixel coordinates are mapped into the range of unit circle. The pixels outside this range are omitted.

In Table 2.1, Zernike Moments and their corresponding orders are given.

Zernike Moments are complex numbers, where rotation of the image f, results in a shift in the phase of these numbers, while their magnitude remains same as shown by (Khotanzad, 1990). Therefore obtaining rotation invariant features is possible by utilizing the magnitudes of Zernike Moments. It is also shown in the same study that under moderate noise, Zernike Features perform well for various classification tasks.

2.1.3 Circular-Mellin Features

Circular-Mellin Features are orthogonal texture features, which are orientation and scale invariant (Ravichandran, 1995). Rotation and scale invariance is obtained by the polar-log coordinate transformation. Cartesian to Polar-log coordinate conversion is as follows, where (x, y) is the coordinates of a point in Cartesian coordinate system and (λ, θ) in polar-log coordinate system,

$$e^{\lambda} = r = \sqrt{x^2 + y^2} \tag{2.10}$$

and

$$\theta = \arctan(y, x) \tag{2.11}$$

Correlation response of an image, $f(\lambda, \theta)$ with a filter $h(\lambda, \theta)$ is given as

$$\int_{\lambda}\int_{\theta} f(\lambda, \theta) h^{*}(\lambda, \theta) e^{2\lambda} d\lambda d\theta \tag{2.12}$$

If the filter h is in the form given in Equation 2.13, the magnitude of correlator output is shown to be rotation invariant in (Ravichandran, 1995) using Circular Harmonic Functions (CHF) where q is defined as the order of CHF or annular frequency.

$$h(\lambda, \theta) = h_q(\lambda) e^{jq\theta} \tag{2.13}$$

Likewise scale invariance can be obtained by using a filter in the form given in Equation 2.14 using Mellin Harmonic Functions (MHF) where p is defined as the order of MHF or radial frequency. However unlike rotation invariance, scale invariance is achieved indirectly. For various scales of $f(x, y)$, the correlator function generates different magnitude values, but the ratio between these values remains same. Thus scale invariance is obtained by comparing the ratios of outputs for distinct p values.

$$h(\lambda, \theta) = e^{-\lambda} h_p(\theta) e^{j2\pi p\lambda} \tag{2.14}$$

A filter function is written in Equation 2.15, which is consistent with the constraints of both Equations 2.13 and 2.14.

$$h_{p,q}(\lambda, \theta) = e^{-\lambda} e^{j2\pi p\lambda} e^{jq\theta} \tag{2.15}$$

With this equation the correlator function output becomes

$$C_{p,q}(f) = \int_\lambda \int_\theta f(\lambda, \theta) h^*_{p,q}(\lambda, \theta) e^{2\lambda} d\lambda d\theta \tag{2.16}$$

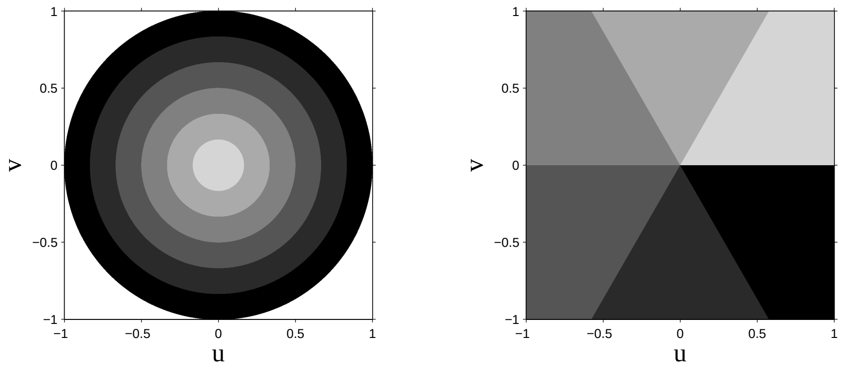
The magnitude of this correlator function, $|C_{p,q}(f)|$, is used as a feature.

2.1.4 Fourier Power Spectrum

Fourier analysis provides mathematical background for analysis of signals based on frequency. Let $f(x, y)$ be the signal representation of an image. Fourier transform of $f(x, y)$ is defined by

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-2\pi j(ux+vy)} dxdy \tag{2.17}$$

In Equation 2.17, j denotes imaginary unit, $\sqrt{-1}$. The Fourier power spectrum is calculated as $|F|^2 = FF^*$ where $*$ denotes complex conjugate. In (Augusteijn, 1995) and (Newsam, 2004) power spectrum is analyzed by ring or wedge shaped regions and four additional statistical features of the entire spectrum is defined. Ring and wedge shaped regions are given in Equations 2.18 and 2.19 respectively. Illustration of these regions are given in 2.1.



(a) Ring Shaped

(b) Wedge Shaped

Figure 2.1: Ring and Wedge shaped regions on Frequency domain. Different shades of gray denote different regions.

$$\phi_{r_1,r_2}(f) = \int_{r_1}^{r_2} \int_{0}^{2\pi} |F(r,\theta)|^2 d\theta dr \tag{2.18}$$

$$\phi_{\theta_1,\theta_2}(f) = \int_{\theta_1}^{\theta_2} \int_{0}^{\infty} |F(r,\theta)|^2 dr d\theta \tag{2.19}$$

Since coarse textures have high values of power spectrum near the origin and finer textures have a more spread out power spectrum (Weszka, 1976), ring shaped regions are associated with coarsness of the texture while wedge shape regions are associated with directionality. In the light of this information, it can be inferred that ring shaped regions are rotationally invariant and wedge shaped regions are not.

Since discrete images are our concern, Discrete Fourier Transform of the N by N image must be utilized, which is given in Equation 2.20.

$$F(u,v) = \frac{1}{N_x N_y} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) e^{-2\pi j \left(\frac{ux}{N} + \frac{vy}{N}\right)} \tag{2.20}$$

Ring shaped regions are approximated with their discrete counterparts and the aforementioned four additional statistical features are calculated as in equations 2.21, 2.22, 2.23 and 2.24 where N_F is the number of frequency components.

$$\text{Maximum Magnitude}(F) = \max\{|F(u,v)| : (u,v) \neq (0,0)\} \tag{2.21}$$

$$\text{Average Magnitude}(F) = \mu_F = \frac{1}{N_F}\sum_u\sum_v |F(u,v)| \tag{2.22}$$

$$\text{Energy of Magnitude}(F) = \sqrt{\sum_u\sum_v (|F(u,v)|)^2} \tag{2.23}$$

$$\text{Variance of Magnitude}(F) = \frac{1}{N_F}\sum_u\sum_v (|F(u,v)| - \mu_F)^2 \tag{2.24}$$

2.1.5 Gabor Filters

Gabor filters are linear filters composed of one harmonic and one Gaussian function. These filters have optimal localization in both spatial and frequency domain by minimizing the joint uncertainty in both domains which is an attractive property (Manjunath, 1996). An interesting fact about the relationship of Gabor filters and human perception is that the characteristics of cortical cells in the mammalian visual cortex can be approximated by Gabor filters (Rangayyan,2000). A two dimensional Gabor function is given in equation 2.25.

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}+2\pi jWx\right)\right]} \tag{2.25}$$

The Fourier transform of the same function is

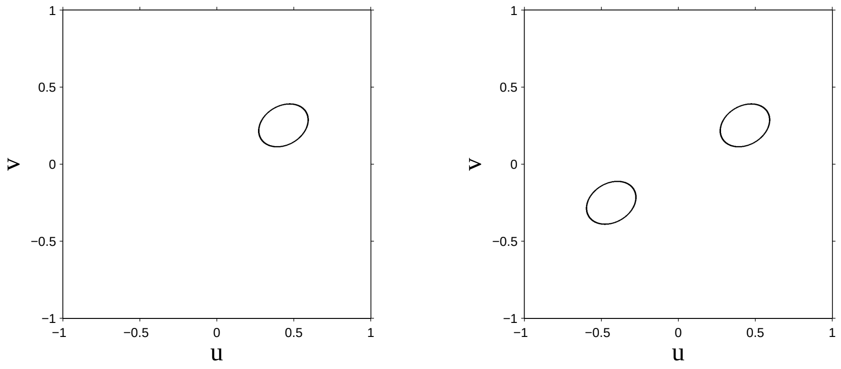
$$G(u,v) = e^{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]} \tag{2.26}$$

In these equations, W controls the modulation and the variances have the relationship given in 2.27 and 2.28.

$$\sigma_x = \frac{1}{2\pi\sigma_u} \tag{2.27}$$

$$\sigma_y = \frac{1}{2\pi\sigma_v} \tag{2.28}$$

Gabor functions form a complete but nonorthogonal set of basis functions. In order to get rid of the redundancy caused by nonorthogonality, these basis functions must



(a) Freq. response of $g_{k,s}(x, y)$

(b) Freq. response of $\Re\{g_{k,s}(x, y)\}$

Figure 2.2: Comparison of complex and real Gabor filters. Complex Gabor filters are not symmetric over frequency domain while their real counterparts are.

be scaled and rotated through a generating function given in Equation 2.29 in order to obtain a Gabor filter dictionary.

$$g_{k,s}(x, y) = a^{-s} g(x', y') \tag{2.29}$$

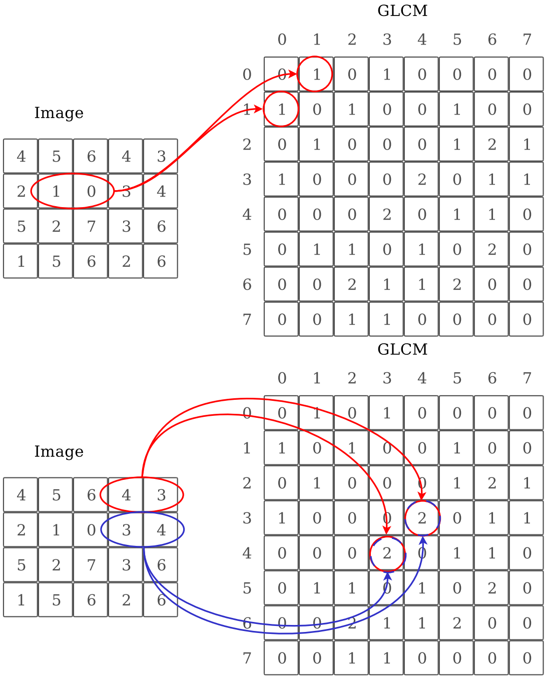
where

$$x' = a^{-s}(x \cos\theta + y \sin\theta) \tag{2.30}$$

$$y' = a^{-s}(-x \sin\theta + y \cos\theta) \tag{2.31}$$

$$\theta = \frac{k\pi}{K} \tag{2.32}$$

In these equations $0 \leq s \leq (S-1)$ and $0 \leq k \leq (K-1)$ where S and K are total number of scales and orientations respectively. Since the real part of the generating function, i.e. $\Re\{g_{k,s}(x, y)\}$, is used as filter, a symmetric frequency response is obtained for each orientation-scale pair. Half magnitudes of frequency responses of generating function and real part of the generating function is given in Figure 2.2 for an arbitrary (k, s) pair. The design of the employed gabor filter dictionary is given in detail in Section 3.1.



4	5	6	4	3
2	1	0	3	4
5	2	7	3	6
1	5	6	2	6

	0	1	2	3	4	5	6	7
0	0	1	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0
2	0	1	0	0	0	1	2	1
3	1	0	0	0	2	0	1	1
4	0	0	0	2	0	1	1	0
5	0	1	1	0	1	0	2	0
6	0	0	2	1	1	2	0	0
7	0	0	1	1	0	0	0	0

4	5	6	4	3
2	1	0	3	4
5	2	7	3	6
1	5	6	2	6

	0	1	2	3	4	5	6	7
0	0	1	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0
2	0	1	0	0	0	1	2	1
3	1	0	0	0	2	0	1	1
4	0	0	0	2	0	1	1	0
5	0	1	1	0	1	0	2	0
6	0	0	2	1	1	2	0	0
7	0	0	1	1	0	0	0	0

Figure 2.3: Example calculation of GLCM matrix

2.1.6 Haralick Features

The features explained in this section are proposed by Haralick (Haralick,1973). These features are very popular for the textural analysis of remote sensed images and derived from a matrix called Gray-Level Co-occurrence Matrix (GLCM) which indicates the distribution of co-occurring values for a given offset. GLCM is $N_g \times N_g$ in size, where N_g is the number of gray levels and is calculated as follows.

$$GLCM_{\Delta x,\Delta y}(\alpha,\beta) = \sum_x \sum_y \begin{cases} 1 & \text{if } f(x,y) = \alpha \text{ and } f(x+\Delta x, y+\Delta y) = \beta \\ 0 & \text{otherwise} \end{cases} \tag{2.33}$$

In this equation, $f(x,y)$ is the image function, $(\Delta x, \Delta y)$ is the offset vector, and finally, since GLCM matrices are symmetric in (Haralick,1973), α and β are interchangeably row or column indices.

Illustration of an example calculation of GLCM matrix is given in Figure 2.3.

After calculation of GLCM, which is not so useful in its raw format, additional operations is required to extract useful information that is contained in this matrix. These operations are called Haralick features. Typically, a subset of Haralick features is employed after obtaining GLCM matrices. In this study, energy, contrast, homogeneity and correlation are used among the Haralick features.

Energy (Angular Second Moment) gives higher values for images having recurring gray levels thus having co-occurrences accumulated in a few gray level pairs and smaller values for noisy images which have a evenly distributed GLCM.

$$\zeta_{energy,\Delta x,\Delta y} = \sum_{\alpha}\sum_{\beta}\{GLCM_{\Delta x,\Delta y}(\alpha,\beta)\}^2 \tag{2.34}$$

Contrast has larger values if co-occurrences are piled up in gray-level pairs that are distant from each other, e.g. elements far from diagonal, and vice-versa. Contrast is defined as follows.

$$\zeta_{contrast,\Delta x,\Delta y} = \sum_{\alpha}\sum_{\beta}|\alpha-\beta|^2\, GLCM_{\Delta x,\Delta y}(\alpha,\beta) \tag{2.35}$$

Homogeneity (Inverse Difference Moment) is opposite of contrast in a way, since it has the difference of gray levels term in denominator, on the contrary of contrast which has it in its numerator.

$$\zeta_{homogeneity,\Delta x,\Delta y} = \sum_{\alpha}\sum_{\beta}\frac{GLCM_{\Delta x,\Delta y}(\alpha,\beta)}{1+|\alpha-\beta|} \tag{2.36}$$

Correlation has higher values if both α and β larger or smaller than their means, and lower values vice versa (Bevk, 2002).

$$\zeta_{correlation,\Delta x,\Delta y} = \sum_{\alpha}\sum_{\beta}\frac{(\alpha-\mu_\alpha)(\beta-\mu_\beta)GLCM_{\Delta x,\Delta y}(\alpha,\beta)}{\sigma_\alpha\sigma_\beta} \tag{2.37}$$

where μ_α, μ_β and σ_α, σ_β are means and standard deviations of marginal-probability matrices,

$$MPM_{\Delta x,\Delta y,\alpha}(\alpha) = \frac{1}{N_g}\sum_{\beta} GLCM_{\Delta x,\Delta y}(\alpha,\beta)$$

and

$$MPM_{\Delta x,\Delta y,\beta}(\beta) = \frac{1}{N_g}\sum_{\alpha} GLCM_{\Delta x,\Delta y}(\alpha,\beta)$$

respectively.

GLCM features' performances are demonstrated to be noteworthy, considering their relative simlicity, low extraction costs and compact feature vectors (Newsam,2004).

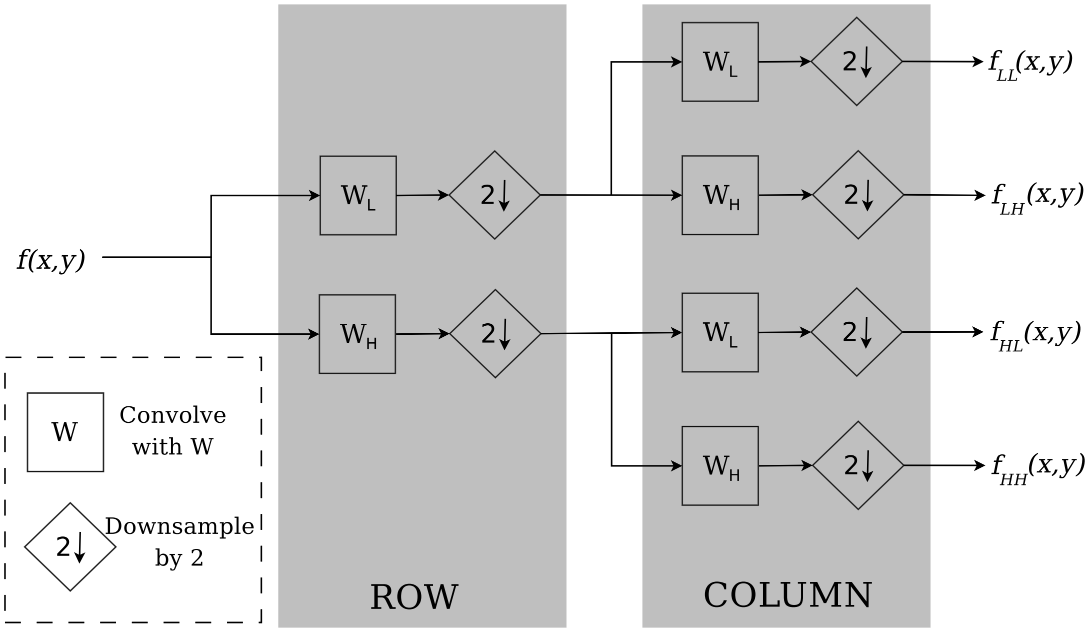


Figure 2.4: Single-level wavelet decomposition

2.1.7 Wavelet Features

Wavelet features are extracted using Discrete Wavelet Transform (DWT). DWT involves filtering and downsampling as shown in Figure 2.4. A multiresolution decomposition is obtained by applying the single-level wavelet decomposition recursively to low-frequency component (Figure 2.5). Multiresolution decomposition provides a basic hierarchical system in order to interpret both frequency and location based information contained within the image. Since utilized wavelet functions are orthogonal, each stage of decomposition finds out different periodical aspects of the image.

2.1.8 HSV Color Space

Features obtained from HSV color space are used in this thesis. The main idea of using features obtained from HSV color space, which is one of the contributions of this study, is to take advantage of the success of this color space in representing perceptual color relationships. These features are explained in Section 3.1 in detail. In order to define these features, images that are inherently in the RGB color space must be converted to

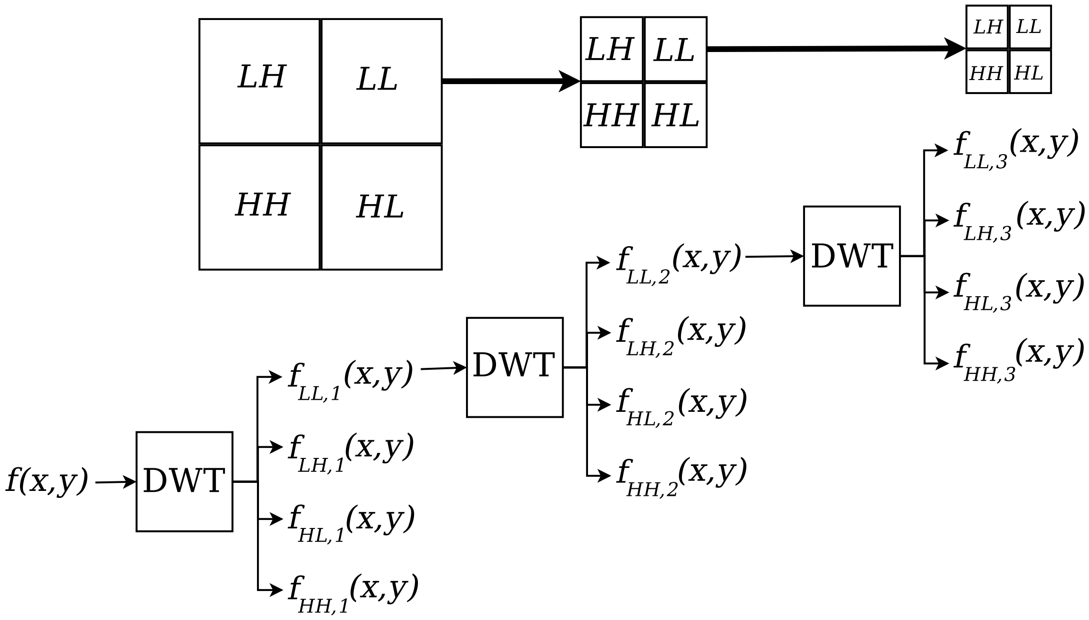


Figure 2.5: Multi-level wavelet decomposition

HSV color space. This is performed with the equations given in Equations 2.38 – 2.40 where $max = \max\{r(x,y), g(x,y), b(x,y)\}$, $min = \min\{r(x,y), g(x,y), b(x,y)\}$, and finally $r(x,y)$, $g(x,y)$, $b(x,y)$ and $h(x,y)$, $s(x,y)$, $v(x,y)$ functions denote channels of RGB and HSV respectively.

$$h(x,y) = \begin{cases} 0^\circ, & \text{if } max = min \\ \left(\frac{g(x,y)-b(x,y)}{max-min} 60^\circ + 360^\circ\right) \bmod 360^\circ, & \text{if } max = r(x,y) \\ \frac{b(x,y)-r(x,y)}{max-min} 60^\circ + 120^\circ, & \text{if } max = g(x,y) \\ \frac{r(x,y)-g(x,y)}{max-min} 60^\circ + 240^\circ, & \text{if } max = b(x,y) \end{cases} \tag{2.38}$$

$$s(x,y) = \begin{cases} 0, & \text{if } max = 0 \\ 1 - \frac{min}{max}, & \text{otherwise} \end{cases} \tag{2.39}$$

$$v(x,y) = max \tag{2.40}$$

2.2 Adaboost Learning Algorithm

Boosting is a general method to improve the performance of a learning algorithm (Freund, 1999). Adaboost (short for Adaptive Boosting) is a boosting algorithm, which takes a set of weak learners and constitutes a linear combination of them, in a number of iterations, to produce a strong classifier. A weak learner is a classifier, which gives weak hypotheses that are insufficient to solve the whole problem alone. These weak learners are often selected as threshold classifiers (Viola, 2001) which decide the output by judging the result of a comparison between input and a threshold. Such a threshold classifier, $h_j(x)$, is given in Equation 2.41.

$$h_j(x_j) = \begin{cases} +1 & \text{if } p_j x_j < p_j \theta_j \\ -1 & \text{otherwise} \end{cases} \tag{2.41}$$

In this equation x_j is the feature, θ_j is the threshold, p_j is the parity which decides the direction of inequality and $1 \leq j \leq K$ where K is the number of features. Every weak learner makes its decision by examining only one feature, so every classifier corresponds to a feature. Training of a weak learner, j, is given in Equation 2.42, and it means determining θ_j and p_j values that minimizes the classification error on the iteration t.

$$(\theta_j, p_j) = \underset{(\theta_j, p_j)}{\arg\min} \{\epsilon_{t,j}\} \tag{2.42}$$

This operation can be achieved simply by searching in intervals $\min(x) \leq \theta_j \leq \max(x)$ and $p_j = \{+1, -1\}$. The definition of $\epsilon_{t,j}$ is given in Equation 2.43 where y_i is the desired output label.

$$\epsilon_{t,j} = \sum_{i:h_j(x_i) \neq y_i} D_t(i) \tag{2.43}$$

In this equation, $D_t(i)$ is the distribution function over training samples on the t^{th} iteration. As it can be observed in the complete algorithm of the Adaboost, given in Figure 2.6, this distribution is utilized to emphasize the misclassified samples, forcing the algorithm to focus on the hard examples in the training set. $D_t(i)$ is initialized

Input: Training data are $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_m, y_m)$ where

- input feature vectors are $\mathbf{x}_i \in X_1 \times X_2 \times ... \times X_K$
- the j^{th} element of $\mathbf{x}_i$ is represented as $x_{i,j} \in X_j$
- the training data labels are $y_i \in Y = \{-1, +1\}$

Initialize: $D_1(i) = \frac{1}{m}$, for $1 \leq i \leq m$
Algorithm:
For $t = 1, ..., T$:

- Train weak classifiers finding (θ_j, p_j) pairs for $i = 1, ..., K$
- Get weak hypotheses $h_j(x_{i,j}) : X_j \to \{-1, +1\}$, for $j = 1, ..., K$
- Select the classifier with minimum error
  - $$j^*(t) = \arg\min_{(\theta_j, p_j)} \{\epsilon_{t,j}\} = \arg\min_j \sum_{i:h_j(x_{i,j}) \neq y_i} D_t(i)$$
  - Set $h_t^* = h_{\xi(t)}$
  - $$\text{Set } \epsilon_t = \epsilon_{t,j^*(t)} = \sum_{i:h_t^*(x_{i,j^*(t)}) \neq y_i} D_t(i)$$
- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$
- Update

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t^*(x_{i,j^*(t)}) = y_i \\ e^{\alpha_t} & \text{if } h_t'^*(x_{i,j^*(t)}) \neq y_i \end{cases} = \frac{D_t(i) e^{-\alpha_t y_i j^*(x_{i,\xi(t)})}}{Z_t}$$

where

$$Z_t = \left[\sum_i D_t(i) e^{-\alpha_t y_i h_t^*(x_{i,j^*(t)})} \right]^{-1}$$

Output:

$$H(\mathbf{x}) = \text{sign} \left(\sum_{t=1}^{T} \alpha_t h_t^*(\mathbf{x}) \right)$$

Figure 2.6: Adaboost Algorithm

to be uniform, and on every ireration it is updated in a way that the true classified samples' values are reduced and false classified samples' values are increased.

Adaboost has many advantages including that it is fast, simple and easy to program. It is also a nice property that it requires no parameters to be tuned, except the iteration count T. Adaboost provides a general and provably effective method for producing an accurate prediction rule by combining rough and moderately inaccurate weak learners. On the other hand, it should be noted that with overly complex or too weak hypotheses, boosting can fail, which is consistent with theory.

2.3 Hough Transform

Hough Transform is a technique which is used to extract arbitrary shapes, such as lines, circles, ellipses, out of an image. It is originally aimed to find lines and invented by Hough. However its most popular form is proposed in (Duda, 1972) using an angle-radius scheme, rather than a slope-intercept one, which is originally used by Hough. Computation principles of Hough transform, intended to find lines, is given below.

All straight lines on an image can be described using two parameters as shown in Figure 2.7, where θ is the angle of the normal of the straight line and ρ is the distance of the line from origin. Equation of this line, in the form $y = mx + n$, is

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \frac{\rho}{\sin\theta} \tag{2.44}$$

The same equation can be written for ρ, in terms of x, y and θ, as

$$\rho = x\cos\theta + y\sin\theta \tag{2.45}$$

For a specified (x_i, y_i) point, this equation becomes the sinusoid given in Equation 2.46, which represents the set of all possible (ρ, θ) that correspond to the specified point.

$$\rho = x_i\cos\theta + y_i\sin\theta \tag{2.46}$$

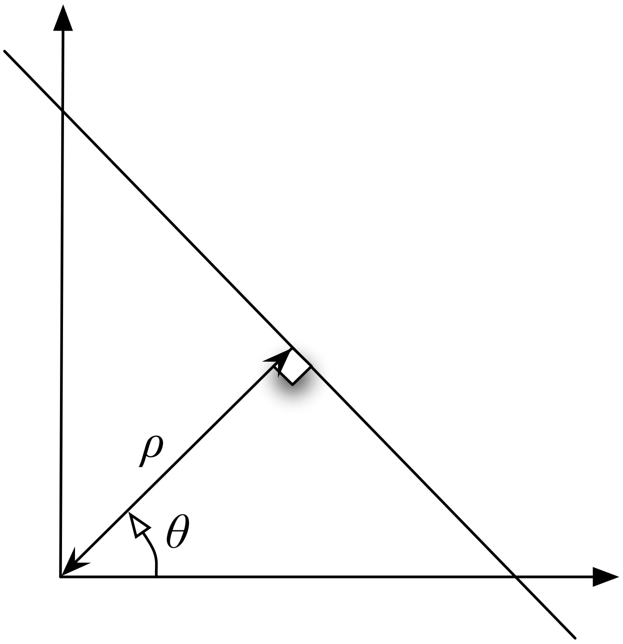
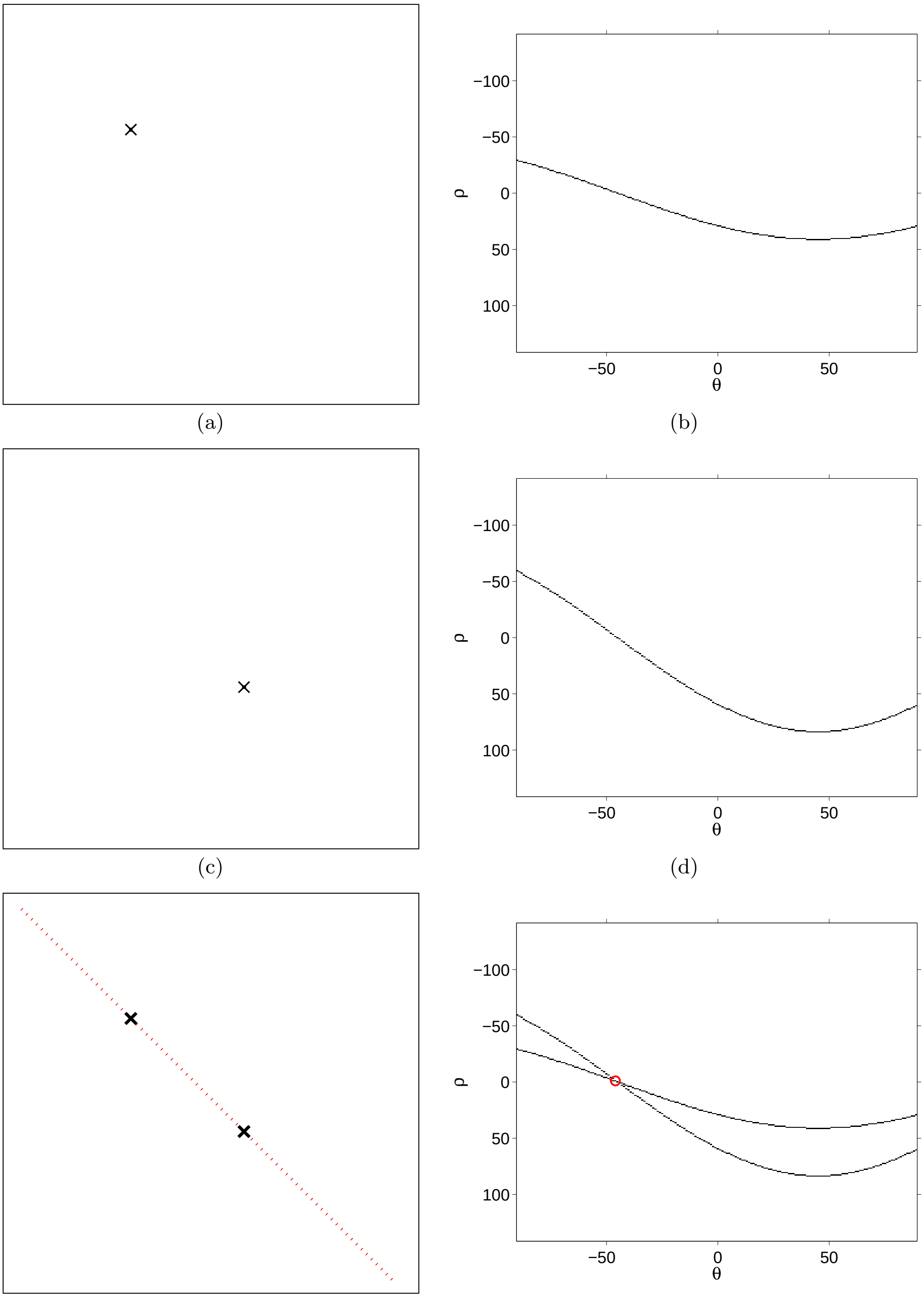


Figure 2.7: Line description for Hough Transform

Hough transform is based upon a voting mechanism, where every point (x_i, y_i) votes for a set of (ρ, θ) pairs, which is described by equation 2.46. If there is a collinear set of points, these votes intercept on a point on $\rho - \theta$ plane, where every (ρ, θ) represents a unique line if θ is given in the interval $[-\pi/2, \pi/2)$. The illustration of this process is provided in Figure 2.8. The Hough Transforms of the points (represented as crosses) in 2.8(a) and 2.8(c), are given respectively in 2.8(b) and 2.8(d). The Hough Transform of the image that has these two points combined, is given in 2.8(f). The intersection point of the sinusoids (i.e. peak point) is marked with a circle. The line corresponding to this peak (ρ, θ) pair is drawn with a dotted line in 2.8(e). It should be noted that every sinusoid belonging to a point on that line, intersects on the same marked peak, assuring that if there is a significant set of collinear points, i.e. lines, there will be a significant peak on the $\rho - \theta$ plane.

The computation of Hough Transform involves an accumulator matrix. The size of this matrix depends on the ρ and θ resolutions. For instance, a θ resolution of $1°$ results in 180 θ values that are $-90°, -89°, ..., 88°, 89°$. The ρ interval is dependent



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.8: Hough Transform

to image size and is in the range $\left[-\sqrt{N_y^2+N_x^2}, \sqrt{N_y^2+N_x^2}\right]$ for a $N_x \times N_y$ image. Computation cost is related to θ resolution, due to the fact that ρ is obtained by rounding up the term in Equation 2.46 for every possible θ on point (x_i, y_i). The elements of the matrix having coordinates θ's and such ρ's are increased by one.

Hough transform provides an efficient and effective method for finding straight lines on an image. Since it does not rely on the local continuity of the collinear pixels, it also achieves a good detection in the case of disconnected lines due to noise or other undesired effects.

2.4 Morphological Operations

Morphological image operations are used for the analysis and processing of geometrical structures on an image (Gonzalez, 2002). These operations are heavily based on set theory and applied on binary images. Two fundamental morphological operations *Dilation* and *Erosion* are as follows.

A digital binary image is defined as $f(x, y)$, where $(x, y) \in \mathbb{Z}^2$, $\mathbb{Z}$ is the set of integers, and since $f(x, y)$ is a binary image, $f(x, y) \to \{0, 1\}$. Let $A \subseteq \mathbb{Z}^2$, $\alpha \in A$ and $z \in \mathbb{Z}^2$. The translation of set A can be defined as

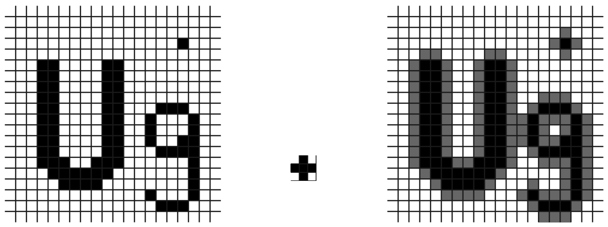
$$(A)_z = \{w | w = \alpha + z, \alpha \in A\}$$

Dilation operation thickens "the non-zero elements on the binary image", i.e. A, by means of the structuring element, B. Dilation of A with B, yields the origin locations of all possible translations of reflected Bs that has at least one common element with A. This operation can be expressed as

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} \tag{2.47}$$

where $B \subseteq \mathbb{Z}^2$ and

$$\hat{B} = \{w | w = -\beta, \beta \in B\}$$



(a) Original Image (b) (c) Dilated Image

Figure 2.9: Morphological dilation operation where (b) is the structuring element. Gray pixels in (c) are the added pixels after dilation operation.

$\hat{B}$ is the reflection (symmetric) of B. Since dilation does a similar job to convolution (Young, 1995), reflection serves as the reversal of one of the functions that takes place in convolution. Although dilation is a commutative operation, thus $A \oplus B = B \oplus A$, as a convention, first parameter is generally given the image and the second is given a structuring element. Dilation operation is visually explained in Figure 2.9.

Erosion operation, contrary to dilation, thins the non-zero elements on the binary image, A, by means of a structuring element, B. Erosion of A with B, yields the origin locations of all possible translations of Bs that has at least one common element with the background of A. This can be expressed, quite similar to dilation, as

$$A \ominus B = \{z | (B)_z \cap A^c \neq \emptyset\} \tag{2.48}$$

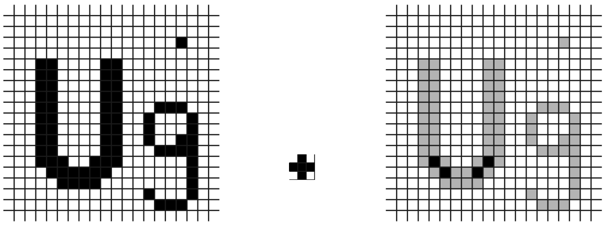
where A^c, the complement (background) of A is

$$A^c = \{w | w \notin A\}$$

Erosion operation is visually explained in Figure 2.10.

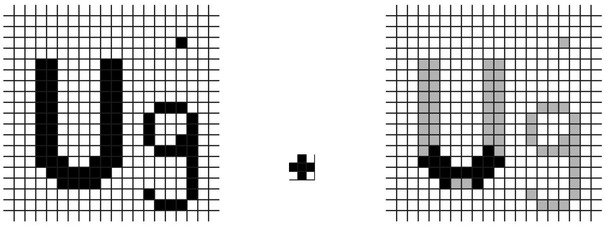
Utilization of dilation and erosion operations in combination, results in two additional operations, which are *Opening* and *Closing*.

Opening is the erosion operation followed by a dilation operation as given in Equation 2.49.



(a) Original Image (b) (c) Eroded Image

Figure 2.10: Morphological erosion operation where (b) is the structuring element. Gray pixels in (c) are the deleted pixels after erosion operation.



(a) Original Image (b) (c) Opened Image

Figure 2.11: Morphological opening operation where (b) is the structuring element. Gray pixels in (c) are the deleted pixels after opening operation.

$$A \circ B = (A \ominus B) \oplus B \tag{2.49}$$

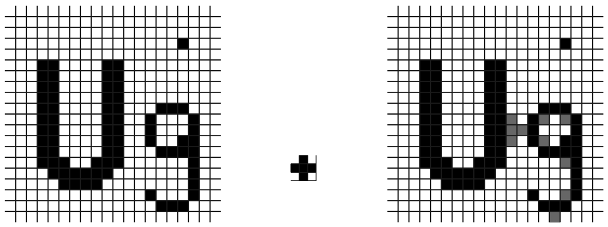
An alternate and more descriptive expression of this operation is

$$A \circ B = \bigcup \{(B)_z | (B)_z \subseteq A\} \tag{2.50}$$

where $\bigcup$ means the union of all sets. The meaning of this formula is the union of all possible translations of B, which can fit inside A. Opening operation is visually explained in Figure 2.11.

Closing is the dilation operation followed by a erosion operation as given in Equation 2.51.

$$A \bullet B = (A \oplus B) \ominus B \tag{2.51}$$



(a) Original Image (b) (c) Closed Image

Figure 2.12: Morphological closing operation where (b) is the structuring element. Gray pixels in (c) are the added pixels after closing operation.

Closing operation is visually explained in Figure 2.12.

2.5 Canny Edge Operator

In general, edge detection provides a way to reduce redundant information of an image, keeping its vital structural properties. Although there are many edge detection algorithms, Canny is a very popular one, because of its optimized nature. In (Canny, 1986), three performance criteria are defined, which are

- **Good detection**: A high rate of true edges and a low rate of false edges must be obtained.
- **Good localization**: Edges must be detected where they actually are.
- **Only one response to a single edge**: An actual edge must produce no more than one edge.

According to these criteria, an optimal function is obtained and approximated by the first derivative of Gaussian in the same study. Canny edge operator is applied to an image in following steps.

- **Noise reduction**: Image smoothing operation is done with a 9×9 Gaussian filter having $\sigma = 1$
- **Gradient Estimation**: 9×9 directional derivatives of 2D Gaussian functions,

which are in x and y directions, are convolved with the image to obtain directional gradient approximations $\frac{\partial f(x,y)}{\partial x}$ and $\frac{\partial f(x,y)}{\partial y}$. Gradient magnitude is

$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

and gradient direction is defined as

$$\angle\nabla f(x,y) = \arctan\left(\frac{\frac{\partial f(x,y)}{\partial y}}{\frac{\partial f(x,y)}{\partial x}}\right)$$

Gradient direction is quantized into 4 directions, which are “North–South”, “East–West”, “North East–South West” and “North West–South East”.

- **Non-maximum suppression**: If a pixel is not the local maximum in its gradient direction, among the neighboring pixels, it is set to ‘0’, yielding thin edge lines.

- **Edge Tracing and Thresholding**: Every edge pixel is traced and thresholding is applied on those edges, with a hysteresis mechanism having two thresholds.

Since, in this study, all the edge detecting operations are performed with Canny, figures showing both original and edge detected images can be found in Chapters 3 and 4 (for instance Figures 3.7(a) and 3.7(b)).

CHAPTER 3

PROPOSED RUNWAY DETECTION ALGORITHM

The proposed runway detection method basically consists of two main stages, which are binary classification of regions based on textural properties, and analysis of these regions based on shape. In the first stage a coarse segmentation is done on the satellite image, in order to find candidate regions for airport runway, based on the textural properties. This segmentation is a binary segmentation, where regions are labeled as either "probably belongs to a runway" or "probably does not belong to a runway". After this segmentation, only regions that possibly belong to a runway are considered and proceed to the second stage. In the second stage, a shape detection algorithm, which discovers long parallel line segments, is carried out on the "possibly runway" regions. These long parallel lines are considered as the identification marks of the two long sides of the elongated rectangle shape of the runway. The block diagram of the general structure of the proposed algorithm is shown in Figure 3.1.

3.1 Determining Runway Candidates Based on Textural Properties

In this study, texture segmentation is the process used for determining runway candidates, where image is partitioned into a possible runway (positive) and not runway (negative) segments. This part is composed of two sections: Feature Extraction and Classification. The block diagram of texture segmentation is given in Figure 3.2.

In order to perform the segmentation, first, satellite images, N_x pixels by N_y pixels in size, are divided into non-overlapping image blocks which are N pixels by N pixels in size. Throughout the segmentation process, these blocks are considered as the basic

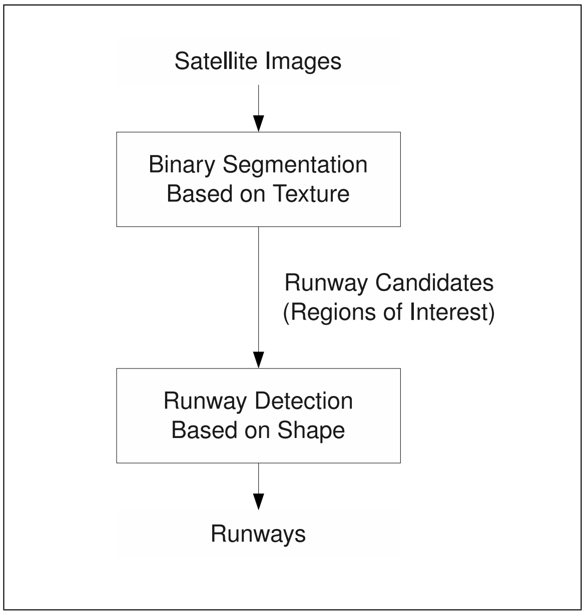

Satellite Images
Binary Segmentation Based on Texture
Runway Candidates (Regions of Interest)
Runway Detection Based on Shape
Runways

Figure 3.1: Block diagram of the general structure of the proposed algorithm

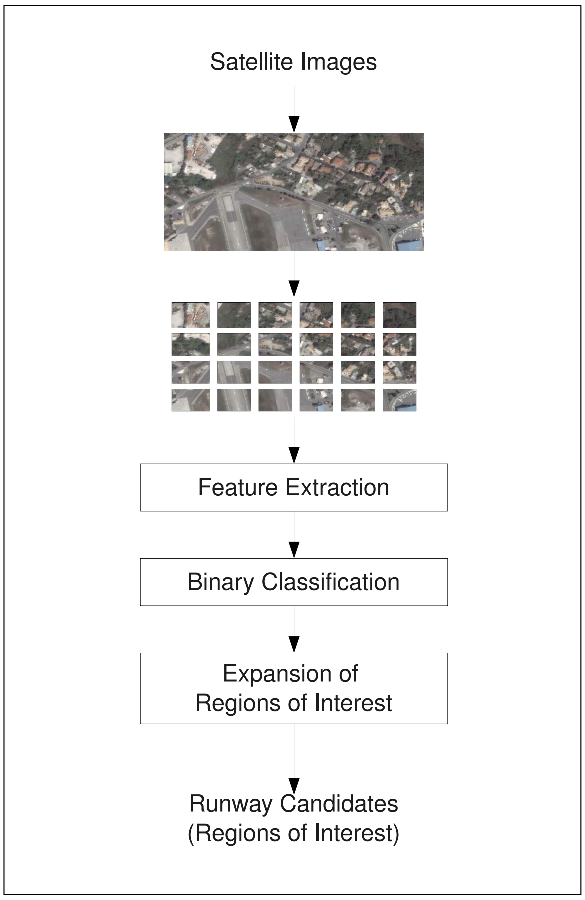

Satellite Images
Feature Extraction
Binary Classification
Expansion of Regions of Interest
Runway Candidates (Regions of Interest)

Figure 3.2: Block diagram of Texture Segmentation

elements to be classified and all the feature extraction and classification operations are executed over these blocks. In this thesis, signal representation of the image block in the b^{th} column and a^{th} row is given in Equation 3.1, where $f(x, y)$ is the gray-level intensity satellite image signal, a and b are Natural numbers that conforms $a < \frac{N_x}{N}$ and $b < \frac{N_y}{N}$. In this study, the block size, N, is set to be 32.

$$\gamma_{a,b}(x, y) = f(aN + x, bN + y) \tag{3.1}$$

After the division, these blocks are sent to the feature extractor where feature vectors are obtained. Feature vectors, which are extracted from the satellite images that belong to the training set, are training vectors. These vectors, along with the ground truth data, are utilized to teach the classification algorithm (Adaboost) the desired outputs. After acquisition of a trained classifier, runway candidates can be obtained by using solely the test vectors.

3.1.1 Feature Extraction

Each feature vector of a block consists of 137 features. Since superiority of a feature in distinguishing a runway is not trivial, this vector is chosen to be a large vector to exploit the feature selector property of the employed Adaboost learning algorithm. The calculation details of utilized features were stated before, in Section 2.1. However, how the feature vectors are formed and which features are used with which parameters, as well as their utilization reasons are covered in this section.

Basic Features

A subset of the utilized features, the basic features set, consists of means and variances of intensity and gradient of intensity of the concerned image block. These features are aligned as given in Equation 3.2. Since runways are generally brighter than the background, mean of intensity image is reasonable to include. Likewise, runways are generally uniform in terms of gray level; therefore variance is a valid feature for distinguishing runways from other landforms. In the same manner, since mean and variance of the gradient magnitude of gray level of image block denote how fast gray level changes and how much this speed varies between neighboring pixels respectively, it also makes sense to add these to the feature vector.

$$\mathbf{F}_{basic} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = \begin{bmatrix} mean_{\gamma} \\ var_{\gamma} \\ mean_{\|\nabla\gamma\|} \\ var_{\|\nabla\gamma\|} \end{bmatrix} \tag{3.2}$$

Zernike Moments

Zernike moments are also used as features in this study. As stated before these moments are rotation invariant image moments. Since the orientation of airport runway is unknown and can be at any angle, the rotation invariance is indispensable. These moments are primarily used and very effective in the field of letter recognition. Due to the fact that these moments can provide adequate information to recognize the letters, they can be useful to recognize runway markings as well.

Calculation of Zernike moments involves two parameters, which are n and m. The order of a Zernike moment is determined by the number n, and this number can take any integer from 0 to $+\infty$. This number must have an upper boundary because it is obviously infeasible to compute infinite number of moments. A method is proposed in (Khotanzad, 1990), which involves gray level thresholding over the original image and reconstructed image acquired from the real image's Zernike moments to obtain two binary images. Then, the Hamming distance between these two images is calculated. The order is selected to be the minimum order number where the Hamming distance is below a desired threshold. In the same paper, selection of this threshold is left uncertain. Because of this ambiguity, in this thesis, maximum order of moments is selected according to the restrictions in memory and computational time. The feature vector that contains Zernike moments used in this thesis is given in Equation 3.3.

$$\mathbf{F}_{zernike} = \begin{bmatrix} F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \\ F_{13} \end{bmatrix} = \begin{bmatrix} Z_{0,0}(\gamma) \\ Z_{1,1}(\gamma) \\ Z_{2,0}(\gamma) \\ Z_{2,2}(\gamma) \\ Z_{3,1}(\gamma) \\ Z_{3,3}(\gamma) \\ Z_{4,0}(\gamma) \\ Z_{4,2}(\gamma) \\ Z_{4,4}(\gamma) \end{bmatrix} \tag{3.3}$$

Circular-Mellin features

Another set of utilized features is Circular-Mellin features, which are also orientation and scale invariant. These features take advantage of two parameters: radial frequency (p) and annular frequency (q). Some experimental results are given in (Ravichandran, 1995) about the selection of these variables by a search algorithm. This search algorithm works as a feature selector and determines the best p and q combination. One of the experiments is based on an aerial image of pentagon, and the ability of a single Circular-Mellin feature ($p = 1$ and $q = 5$) to identify different wings of the structure is shown. The set of employed Circular-Mellin features is given in Equation 3.4 which includes the aforementioned $p = 1$, $q = 5$.

$$\mathbf{F}_{circular-mellin} = \begin{bmatrix} F_{14} \\ F_{15} \\ F_{16} \\ F_{17} \\ F_{18} \\ F_{19} \\ F_{20} \\ F_{21} \\ F_{22} \\ F_{23} \end{bmatrix} = \begin{bmatrix} C_{1,1}(\gamma) \\ C_{1,2}(\gamma) \\ C_{1,3}(\gamma) \\ C_{1,4}(\gamma) \\ C_{1,5}(\gamma) \\ C_{2,1}(\gamma) \\ C_{2,2}(\gamma) \\ C_{2,3}(\gamma) \\ C_{2,4}(\gamma) \\ C_{2,5}(\gamma) \end{bmatrix} \tag{3.4}$$

Fourier Power Spectrum

Fourier Power Spectrum is analyzed in order to extract features related with periodic image patterns. The power spectrum of the image block can be examined in ring shaped or wedge shaped regions as mentioned previously. Wedge shaped regions are orientation dependent and so they are irrational to use for runway detection. On the other hand, analysis over ring shaped regions can provide logical information based on recurring forms. In this thesis, power spectrum is divided into six equal ring shaped regions and the total powers comprised by each region are included as features. In addition to these features, maximum value, average value and variance of Discrete Fourier Transform magnitude of image block, as well as overall power spectrum energy are added to the feature vector. These features are formed as given in Equation 3.5 where $\Gamma(u,v) = DFT\,\{\gamma(x,y)\}$ and r_n's are equally spaced radii over frequency domain, r_6 being the maximum frequency.

$$\mathbf{F}_{fourier} = \begin{bmatrix} F_{24} \\ F_{25} \\ F_{26} \\ F_{27} \\ F_{28} \\ F_{29} \\ F_{30} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = \begin{bmatrix} \phi_{0,r_1}(\Gamma) \\ \phi_{r_1,r_2}(\Gamma) \\ \phi_{r_2,r_3}(\Gamma) \\ \phi_{r_3,r_4}(\Gamma) \\ \phi_{r_4,r_5}(\Gamma) \\ \phi_{r_5,r_6}(\Gamma) \\ \text{Maximum Magnitude}(\Gamma) \\ \text{Average Magnitude}(\Gamma) \\ \text{Energy of Magnitude}(\Gamma) \\ \text{Variance of Magnitude}(\Gamma) \end{bmatrix} \tag{3.5}$$

Gabor filters

A dictionary of Gabor Filters is designed in order to reduce redundancy with typical parameters: $K = 6$ orientations and $S = 4$ scales. Other parameters are chosen according to (Rangayyan, 2000) and given in Equations 3.6 − 3.9 where lower and upper center frequencies selected as $U_l = 0.06$ and $U_h = 0.5$.

$$W = U_h \tag{3.6}$$

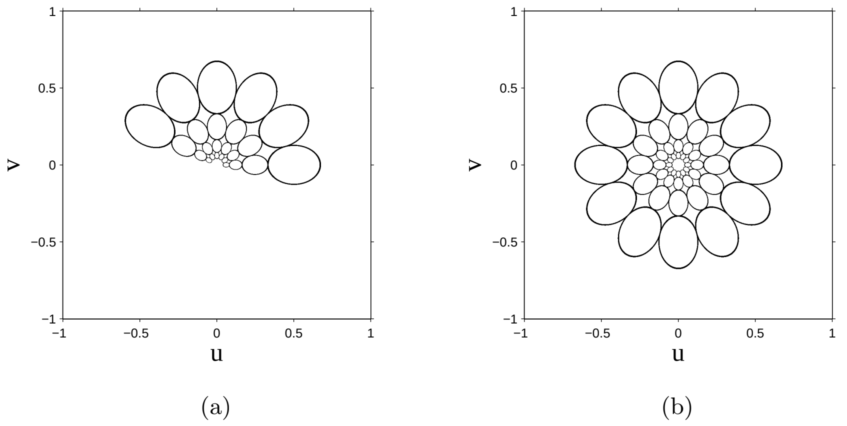


Figure 3.3: Half magnitude contours of designed Gabor filter dictionary. (a) Frequency responses of the generating function outputs (Complex), (b) Frequency responses of the real parts of the generating function outputs used as filters

$$a = \left(\frac{U_h}{U_l}\right)^{\frac{1}{S-1}} \tag{3.7}$$

$$\sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln(2)}} \tag{3.8}$$

$$\sigma_v = \frac{\tan(\frac{\pi}{2K})\left[U_h - (\frac{\sigma_u^2}{U_h})2\ln(2)\right]}{\sqrt{2\ln(2) - \frac{(2\ln(2))^2\sigma_u^2}{U_h^2}}} \tag{3.9}$$

Half magnitude contours of frequency responses of the designed Gabor filters are given in Figure 3.3. By examining the frequency coverage of the filters in the figure, it can be observed that Gabor filters which are shaped by given parameters have minimal redundancy of information.

Means and variances of the Gabor filtered output images are used as features, as shown in Equation 3.10 where $G_{k,s}(\gamma)$ denotes the γ signal, filtered with the Gabor filter, $g_{k,s}$.

$$\mathbf{F}_{gabor} = \begin{bmatrix} F_{34} \\ F_{35} \\ F_{36} \\ F_{37} \\ \vdots \\ F_{40} \\ F_{41} \\ \vdots \\ F_{80} \\ F_{81} \end{bmatrix} = \text{Circular Shift}\left(\begin{bmatrix} mean_{G_{1,1}(\gamma)} \\ var_{G_{1,1}(\gamma)} \\ mean_{G_{1,2}(\gamma)} \\ var_{G_{1,2}(\gamma)} \\ \vdots \\ mean_{G_{1,4}(\gamma)} \\ var_{G_{1,4}(\gamma)} \\ \vdots \\ mean_{G_{6,4}(\gamma)} \\ var_{G_{6,4}(\gamma)} \end{bmatrix} \right) \tag{3.10}$$

Gabor filters characteristically possess directionality which is not a desired property for runway detection problem. To overcome this difficulty and make Gabor filter outputs approximately rotation invariant, a simple method is proposed in (Newsam, 2005). According to this method, the vector given in Equation 3.10 is circularly shifted so that the scale-orientation pair having the maximum mean is located at the beginning of the vector. This method is shown to be effective by the experimental results in the same study.

Haralick Features

For the calculation of Haralick features, GLCM matrices are obtained. When no a priori information is available, it is common to use offsets $(1,0)$, $(1,-1)$, $(0,-1)$ and $(-1,-1)$, which correspond to adjacent pixels at $0°$, $45°$, $90°$ and $135°$ respectively. Therefore these offsets are utilized in thesis. As mentioned before, four Haralick features (energy, contrast, homogeneity and correlation) for four offsets (sixteen features in total) are employed. The feature vector is given in Equation 3.11 where $\zeta_{h,\Delta x,\Delta y}$ is the Haralick feature, h, calculated at the offset $(\Delta x, \Delta y)$

$$\mathbf{F}_{haralick} = \begin{bmatrix} F_{82} \\ F_{83} \\ F_{84} \\ F_{85} \\ \vdots \\ F_{94} \\ F_{95} \\ F_{96} \\ F_{97} \end{bmatrix} = \begin{bmatrix} \zeta_{energy,1,0} \\ \zeta_{contrast,1,0} \\ \zeta_{homogeneity,1,0} \\ \zeta_{correlation,1,0} \\ \vdots \\ \zeta_{energy,-1,-1} \\ \zeta_{contrast,-1,-1} \\ \zeta_{homogeneity,-1,-1} \\ \zeta_{correlation,-1,-1} \end{bmatrix} \tag{3.11}$$

Wavelet Analysis

Wavelet features are acquired using a multi resolution analysis and daubechies-4 wavelet. These features are expected to provide quantative description of textural properties related with frequency like Fourier Power Spectrum. Contrarily, Wavelet Analysis has localization in spatial domain too. A three level decomposition structure is employed and energies and standard deviations of four components (Low-Low, Low-High, High-Low, High-High) for three levels are used as features, making 24 features in total. The resulting feature vector is given in Equation 3.12 where $energy_{c,s}$ and $var_{c,s}$ are the energy and variance of the wavelet filtered signals of component, c at stage, s.

$$\mathbf{F}_{wavelet} = \begin{bmatrix} F_{98} \\ F_{99} \\ F_{100} \\ F_{101} \\ F_{102} \\ F_{103} \\ \vdots \\ F_{120} \\ F_{121} \end{bmatrix} = \begin{bmatrix} energy_{LL,1} \\ var_{LL,1} \\ energy_{LH,1} \\ var_{LH,1} \\ energy_{HL,1} \\ var_{HL,1} \\ \vdots \\ energy_{HH,3} \\ var_{HH,3} \end{bmatrix} \tag{3.12}$$

Features in HSV color space

This color space represents the color value in a very parallel way with human perception. Since human is considered to be successful at discriminating elements like runway textures, and by all means, the color is an evaluated aspect by humans, using HSV is advantageous for the problem for certain. For instance, saturation will surely provide very valuable information about a runway, bearing in mind that the runways tend to be in gray tones and colorfulness is a synonym for saturation. Likewise hue is closely related to dominant wavelength and although it is not so evident, dominant wavelength of the color of a runway might be useful in discrimination from other landforms. For these reasons, mean, variance, mean of gradient magnitude and variance of gradient magnitude as well as Zernike moment of order one and Circular-Mellin feature ($p = 1$, $q = 1$) for both Saturation and Value components are employed. Since these two components are linear data, common mean and variance formulas, given in Section 2.1.1 still apply. On the other side, Hue is an angular data and directional statistics is involved in mean and variance calculations which are given in Equations 3.13 – 3.16 where $h(x, y)$, $s(x, y)$ and $v(x, y)$ functions denote channels of HSV image and $\angle$ denotes the angle of complex number. Since Zernike and Circular-Mellin features inherently require magnitudes, rather than angles, Hue component is not utilized for these features. The resulting feature vector is given in Equation 3.17.

$$mean_h = \angle \left[\sum_x \sum_y \left(e^{2\pi j \ h(x,y)} \right) \right] \tag{3.13}$$

$$var_h = 1 - \frac{1}{N^2} \left| \sum_x \sum_y \left(e^{2\pi j \ h(x,y)} \right) \right| \tag{3.14}$$

$$\begin{aligned} d_{h,1}(x, y) &= (h(x, y) - h(x - 1, y) + 360^\circ) \bmod 360^\circ \\ d_{h,2}(x, y) &= (h(x, y) - h(x, y - 1) + 360^\circ) \bmod 360^\circ \\ \|\nabla h(x, y)\| &= \sqrt{d_{h,1}^2(x, y) + d_{h,2}^2(x, y)} \\ mean_{\|\nabla h\|} &= \frac{1}{N^2} \sum_x \sum_y \|\nabla h(x, y)\| \end{aligned} \tag{3.15}$$

$$var_{\|\nabla h\|} = \sqrt{\frac{1}{N^2} \sum_x \sum_y (\|\nabla h(x, y)\| - mean_{\|\nabla h\|})^2} \tag{3.16}$$

$$\mathbf{F}_{hsv} = \begin{bmatrix} F_{122} \\ F_{123} \\ F_{124} \\ F_{125} \\ F_{126} \\ F_{127} \\ F_{128} \\ F_{129} \\ F_{130} \\ F_{131} \\ F_{132} \\ F_{133} \\ F_{134} \\ F_{135} \\ F_{136} \\ F_{137} \end{bmatrix} = \begin{bmatrix} mean_h \\ var_h \\ mean_{\|\nabla h\|} \\ var_{\|\nabla h\|} \\ mean_s \\ var_s \\ mean_{\|\nabla s\|} \\ var_{\|\nabla s\|} \\ Z_{1,1}(s) \\ C_{1,1}(s) \\ mean_v \\ var_v \\ mean_{\|\nabla v\|} \\ var_{\|\nabla v\|} \\ Z_{1,1}(v) \\ C_{1,1}(v) \end{bmatrix} \tag{3.17}$$

Employing features from HSV color space for runway detection is a novel practice and it is shown to be very effective for the solution of the problem on Section 4.2.

3.1.2 Classification

The second step of the segmentation is the classification. In order to perform the classification, subsequent to obtaining features, large feature vectors are formed for each image block as shown in Equation 3.18.

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{basic} \\ \mathbf{F}_{zernike} \\ \mathbf{F}_{circular-mellin} \\ \mathbf{F}_{fourier} \\ \mathbf{F}_{gabor} \\ \mathbf{F}_{haralick} \\ \mathbf{F}_{wavelet} \\ \mathbf{F}_{hsv} \end{bmatrix} \tag{3.18}$$

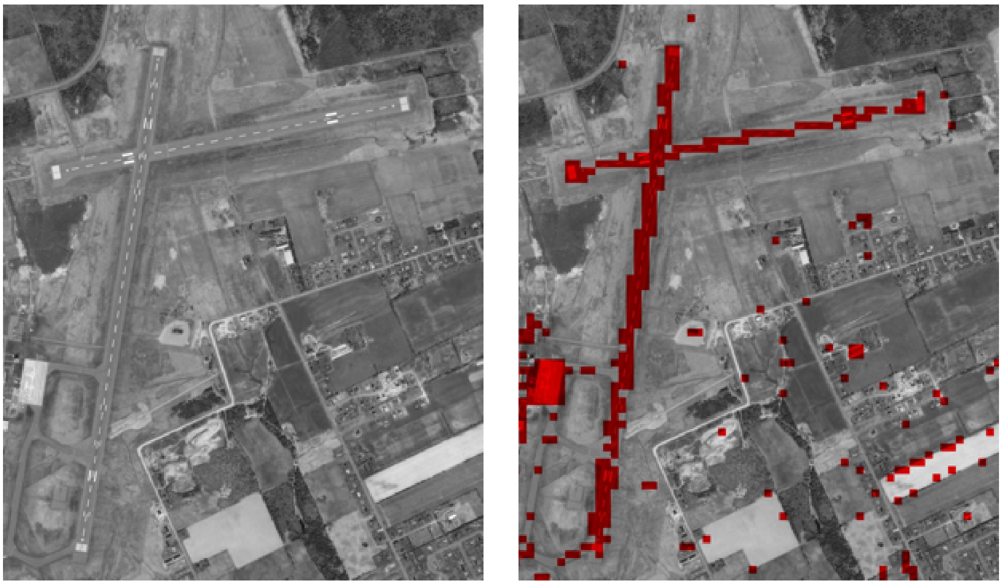
In order to clarify which feature is better than which one for the classification task of runway and other landforms, a considerable amount of features are selected in the previous section and Adaboost is employed because of its feature selector property. A weak classifier is utilized for every feature and Adaboost, as a meta-classifier, is asked to find the most helpful weighted combination of T weak classifiers. As mentioned before, Adaboost follows an iterative method and on every iteration, Adaboost selects a new weak classifier that minimizes the error for the training dataset, until the number of selected weak classifiers reaches the desired number, T. Experimental results are given in Chapter 4, for up to $T = 40$. Runway candidates, i.e. the image blocks which are probably belong to a runway, can be determined after training of Adaboost. The blocks labeled in this way are represented as

$$\Lambda(a,b) = \begin{cases} 1, & \text{if block represented by } \gamma_{a,b} \text{ is probably a runway} \\ 0, & \text{otherwise} \end{cases} \tag{3.19}$$

In Figure 3.4, an example outcome of this segmentation process given.

3.1.2.1 Expansion of Regions of Interest

Image blocks, which are on the boundary of runways and have mostly “non-runway” pixels, might be classified as negative despite they also have pixels which belong to a runway. Since they have edges which are very important for the rest of the algorithm, a countermeasure is needed. For this purpose, $\Lambda(a,b)$ function given in Equation 3.19 is considered as a binary image and a morphological dilation operation is performed with the structuring element consisting of all ones and 3×3 in size. This operation



(a) Original Image

(b) Segmentation Result

Figure 3.4: Result of a Segmentation Process. (a) is the original image and the red areas in (b) represents the blocks that are labeled as possible runway.

helps to include blocks containing runway edges, if they have not been labeled positive. The function of this operation is illustrated in Figure 3.5. Preservation of the runway side line edges can be observed in the figure.

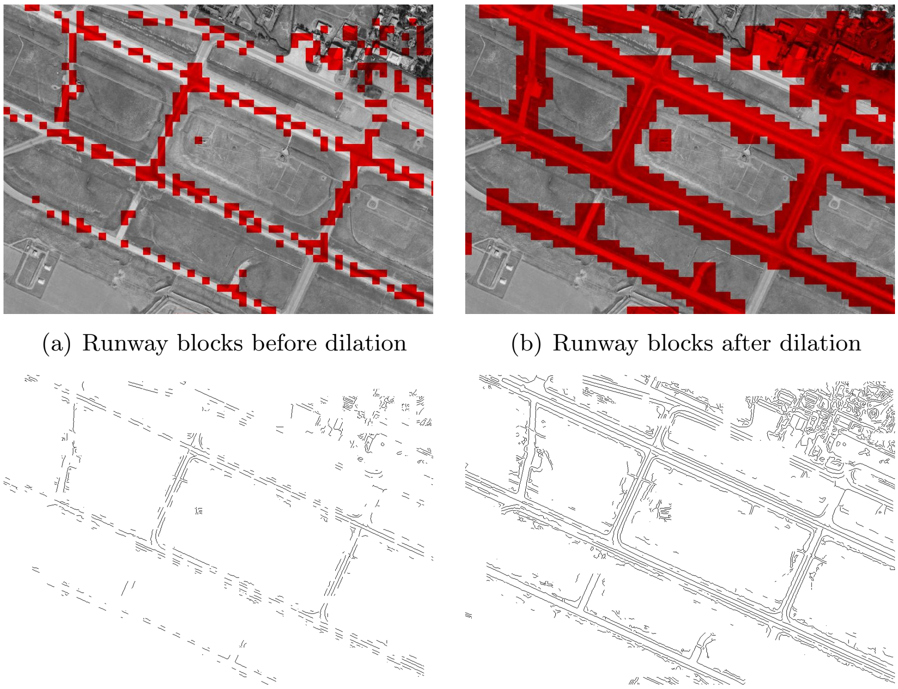
Let this dilation operation produces the function $\Lambda'(a,b)$. Then the regions of interest can be expressed by the set, given in Equation 3.20.

$$\left\{(x,y)|(aN \leq x < (a+1)N) \wedge (bN \leq y < (b+1)N \wedge (\Lambda'(a,b) = 1))\right\} \tag{3.20}$$

3.2 Runway Shape Detection

After discovering runway candidates, further processing is carried out in order to eliminate false positives, based on the shape analysis of the runway. This shape detection operation is shown in Figure 3.6 as block diagram.

Runway shape detection is composed of four main processes: Edge Detection, Modified Hough Transform, Peak Pair Detection and Line Segment Pair Detection.



(a) Runway blocks before dilation

(b) Runway blocks after dilation

(c) Edges before dilation

(d) Edges after dilation

Figure 3.5: Function of dilation operations

Prior to the shape detection, the satellite images are diminished by a ratio R for faster calculation and smaller memory footprint. The resizing operation is done by bicubic interpolation, where output pixel is weighted average of the pixels in the nearest 4×4 neighborhood. The images are shrinked to their half sizes in this thesis. Resized image $f'(x, y)$ is $N'_x \times N'_y$ in size, where $N'_x = \frac{N_x}{R}$ and $N'_y = \frac{N_y}{R}$. The block size for the resized image is $N' = \frac{N}{R}$.

3.2.1 Edge Detection

Airport runways are generally easily distinguished from background, yielding edges after Edge Detection. Therefore an Edge Detection operation is done for the regions of interest found on the previous step. Canny edge operator is selected for this operation because of its optimal solution for low defective edge rate, localization of edges and giving one response for a single edge (Canny, 1986).

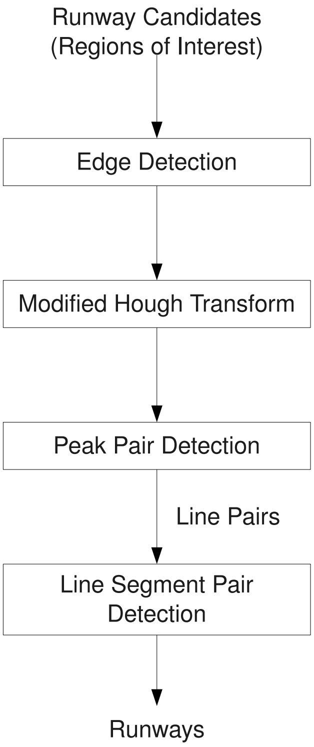
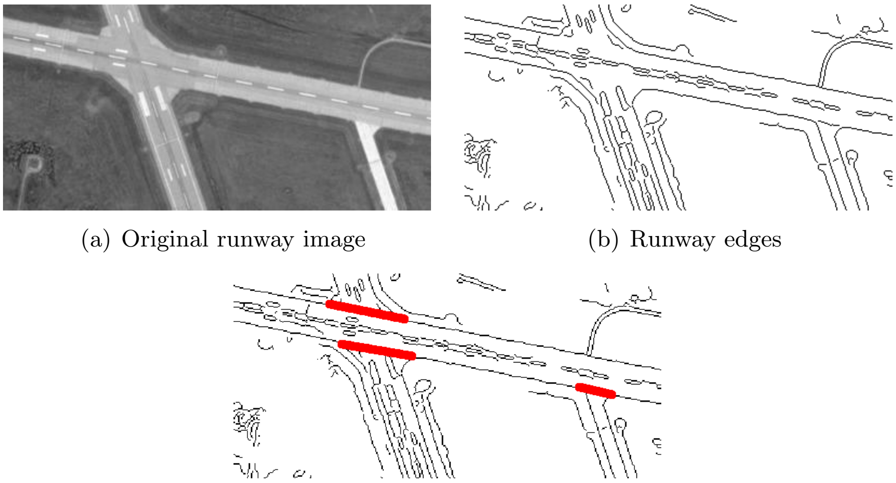


Figure 3.6: Block diagram of the runway shape detection operation



(a) Original runway image

(b) Runway edges

(c) Disjointed edges

Figure 3.7: Disjointed lines on airport edge image. (c) shows where discontinuity occurs.

3.2.2 Modified Hough Transform

Hough transform is a powerful tool used in finding lines in its most basic form. As previously mentioned, a nice property of Hough transform is that it still works even the edge is disconnected contrary to edge tracking algorithms. This is a desired attribute for the runway detection problem, considering cases like disjointed edges due to the junctions connecting main runway to taxiways which can be observed in Figure 3.7.

The proposed modified hough transform, which is one of the contributions of this thesis, is applied on the edge detected image. Regular hough transform is changed in order to be specialized in finding runway side lines. Runways generally have the same background on either side, which results in inverse mean gradient directions on opposing runway sides as shown in Figure 3.8. The modification is done to exploit this property.

Each edge point's contribution to the accumulator matrix is changed to be the term given in equation 3.21, which is the inner product of unit vector in the direction of θ, and unit vector in the gradient direction on that edge point instead of a constant.

$$\left\langle \vec{r}_\theta, \vec{r}_{\angle \nabla f(x,y)} \right\rangle = \cos(\theta - \angle \nabla f(x,y)) \tag{3.21}$$

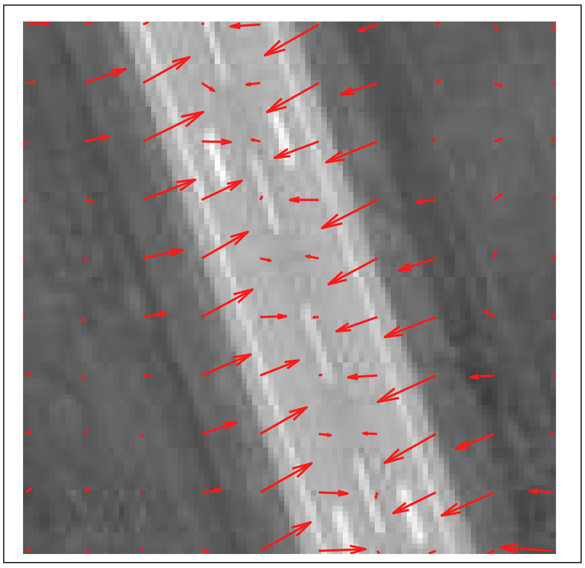


Figure 3.8: Gradient vectors in the vicinity of runway. Runway background is same on either side resulting in opposite gradient directions

In the equation 3.21, $\vec{r}_\theta$ and $\vec{r}_{\angle\nabla f(x,y)}$ denotes unit vectors in the direction of θ, and in the gradient direction on the edge point respectively. Since local gradient of the image intensity is orthogonal to the edge, irrelevant edge points, i.e. points that belong to another line or just noise and thus having gradient direction far from θ and $\theta + \pi$, are eliminated by this formula. Edge points belonging to a runway side having orientation θ will do the maximum contribution (approximately ∓ 1).

3.2.3 Peak Pair Detection

Since opposing long side lines of runways have inverse mean gradient directions, if points on one side contribute as 1, the contribution on the other side will necessarily be -1. Thus we expect to see one positive and one negative peak on accumulator array that are $\Delta\rho$ away from each other. Interval of possible $\Delta\rho$'s is related with minimum and maximum widths of airport runways desired to be detected and obviously the satellite image resolution. Likewise, the peak magnitude threshold, which determines the minimum magnitude of a peak to be counted as runway side line, is expected to be proportional with minimum runway length desired to be detected. However, due to the effects like gradient estimation error, it cannot be decided easily. For this purpose, an experiment is performed in Section 4.3, and the parameter is selected accordingly. Accumulator matrix is searched for such opposite peak pairs representing a runway. Each opposite peak pair, corresponds to a line pair, and these line pairs are saved for a further local analysis, which is line segment detection.

3.2.4 Line Segment Pair Detection

Lines are obtained previously by peak detection, and runway sides should be over those lines. Since the runway sides must inherently be represented as line segments instead of lines, an algorithm is designed for this purpose, as follows.

First, in order to get rid of the edge points that are in the unwanted directions, morphological image opening operation is performed as a preprocessor with an appropriate structuring element, which is consistent with θ. These structuring elements are given in Table 3.1. The decision of which structuring element will be used for opening operation, is made by judging the distances between the structuring element angle and θ, i.e. the i^{th} structuring element is utilized, given in the Equation 3.22.

$$\arg\min_i\{(\theta - \psi_i + 360^\circ) \bmod 360^\circ\} \tag{3.22}$$

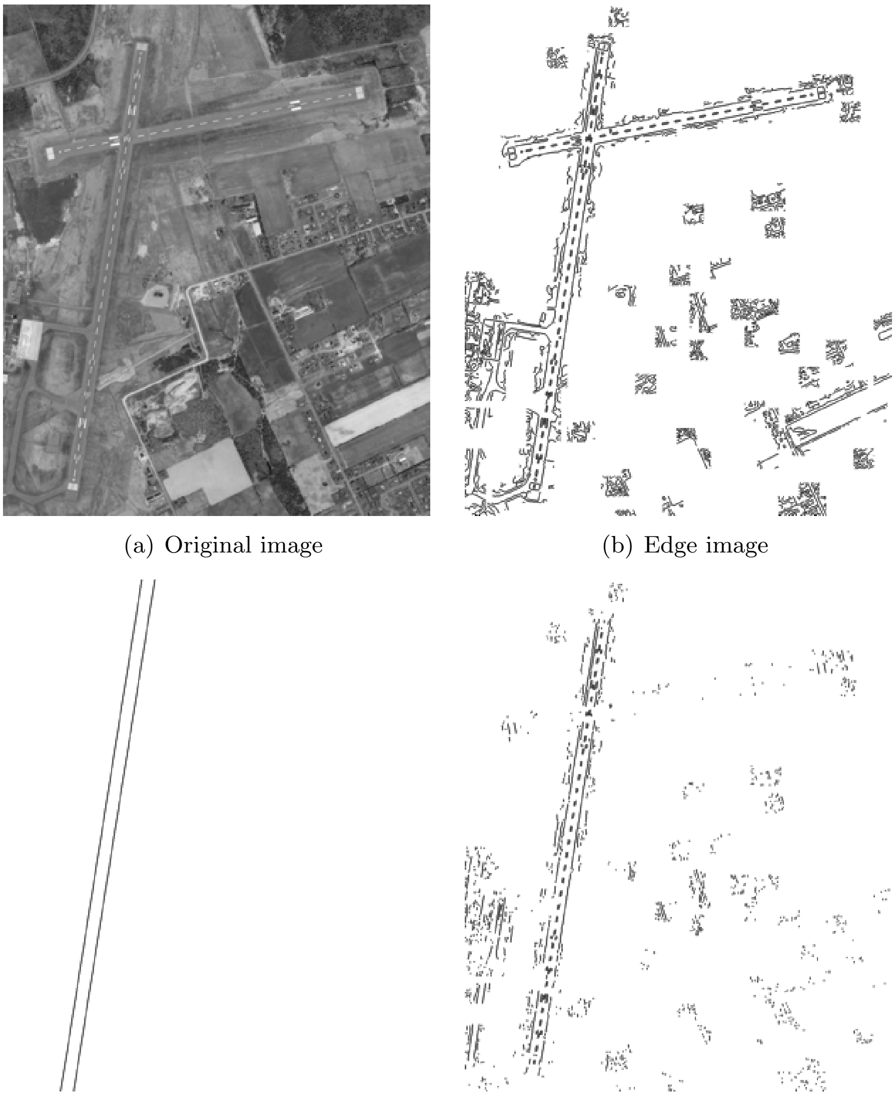
The function of this opening operation is given and explained visually in Figure 3.9, through an example.

After the opening operation, line is sequentially searched for “facing pixels”. A facing pixel can be defined as follows. Let line A and line B be a line pair, and let point α and point β be points on these lines respectively. If point α is the closest point to point β on the line A, then these points are facing pixels. If both of these facing pixels have non-zero value, these pixels are marked.

For the runway detection, longest continuous series of the marked pixels are considered. This longest continuous facing pixel set is merged with other continuous facing pixel sets that are at most “Maximum Tolerated Gap” away, until the distance between closest facing pixel set is greater than this threshold. After merge operation, if the resulting set is long enough, that is it has “Minimum Runway Length”, the

Table 3.1: Structuring elements used to eliminate false lines

Angle (°)	**Structuring Element**
0	□■□ □■□ □■□
18.4	■□□ □■□ □■□
45	■□□ □■□ □□■
56.3	□□□ ■■□ □□■
90	□□□ ■■■ □□□
-90	□□□ ■■■ □□□
-56.3	□□□ □■■ ■□□
-45	□□■ □■□ ■□□
-33.7	□□■ □■□ □■□



(a) Original image

(b) Edge image

(c) Detected Line Pair

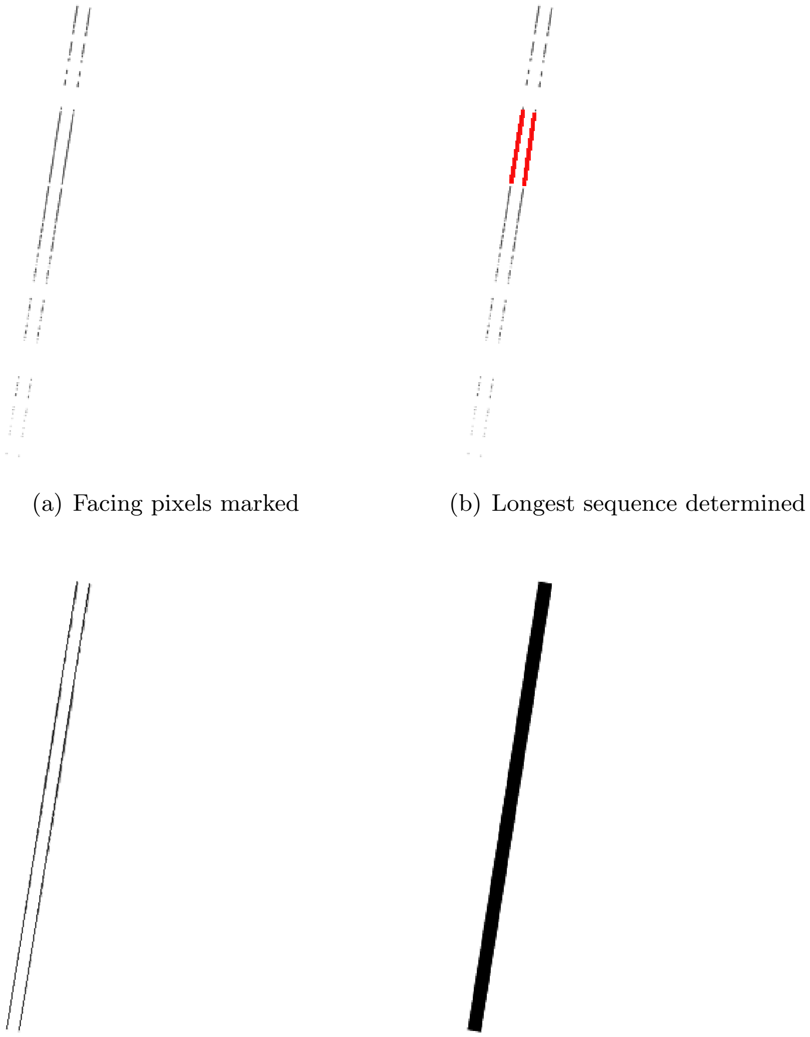
(d) Opened edge image

Figure 3.9: Image opening operation. (a) is the original image, (b) is the edge image, (c) is the line pair detected by hough transform and (d) is the image obtained by image opening operation with the 1^{st} structuring element in Table 3.1. One should notice the removal of the considerable amount of noise and the other runway present in (b) since their lines are in irrelative direction.

area between these line segment pairs is marked as runway. The selection of these "Maximum Tolerated Gap" and "Minimum Runway Length" parameters are given in Section 4.3.

The determining the longest facing pixel set and merging process is shown in Figure 3.10 on the same runway shown in Figure fig:imageOpeningOperation.

As a last remark, one may think that the selection of the longest facing pixel set may jeopardize the detection of another runway on the image. Since the length comparison is performed on just one (θ, ρ) pair, this proposition is not true. The only case this doubt may occur is when the two runway side lines fall into the same and only one (θ, ρ) bin on the Hough accumulator matrix. This has an extremely low probability, considering that even one runway generates more than one line pairs frequently. In other words, this case requires the odd case of second airport runway to be in the exact same orientation, have the exact same distance from origin and possess the exact same width with the first one.



(a) Facing pixels marked

(b) Longest sequence determined

(c) Merging performed

(d) Runway rectangle

Figure 3.10: Final phase of runway detection

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, the experimental results obtained by running the proposed algorithm are given. These experimental results are carried out with a dataset, consisting of 78, large satellite images having sizes of 13626×10862 on average and resolution of 1 meter that are obtained using Google Earth software. 30 of these images are randomly selected for training of Adaboost and 48 of them are reserved for testing. Each image is divided into blocks of size 32×32 and in this way 4422711 blocks are obtained for training. Each block of the training images is labeled as runway (positive) if more than half of its pixels belongs to main runway of an airport and labeled as non-runway (negative) if not. Such a labeling, in training images resulted in 5950 runway blocks and 4416761 non-runway blocks. For composition of the training set, 10% of the non-runway blocks are randomly selected for use because of memory constraints, while full set of the runway blocks are utilized. Ground truth data are constituted manually, by marking "only" the main runways.

4.1 The Performance Criteria

The performance evaluation is done on two types of results, where one of them is for the outcome of segmentation stage and the other is for the final result.

The performance evaluation for segmentation is carried out over the image blocks mentioned previously. In order to define performance equations, the ground truth data must be represented in a compatible way with Equation 3.19, which is

$$\Omega(a,b) = \begin{cases} 1, & \text{if block represented by } \gamma_{a,b} \text{ is truly a runway} \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

where "truly a runway" implies that "at least half of the pixels contained within the image block, belong to a runway".

The **True Positive Rate (Sensitivity)**, which is the ratio of correctly classified positive labeled samples to truly positive samples (indicates how much the algorithm fails to find an existing airport), can be expressed as

$$\text{True Positive Rate} = \frac{\sum_{(a,b):\Lambda(a,b)=\Omega(a,b)} \Omega(a,b)}{\sum_{(a,b)} \Omega(a,b)} \tag{4.2}$$

Likewise **True Negative Rate (Specificity)**, which is the ratio of correctly classified negative labeled samples to truly negative samples (indicates how much the algorithm fails to label a truly non-airport block as a non-airport block), can be written as

$$\text{True Negative Rate} = \frac{\sum_{(a,b):\Lambda(a,b)=\Omega(a,b)} 1 - \Omega(a,b)}{\sum_{(a,b)} 1 - \Omega(a,b)} \tag{4.3}$$

The **Positive Hit Rate (Positive Predictive Power)**, which is the ratio of correctly classified positive samples to all positive labeled samples (indicates how rare false alarms occur), can be expressed as

$$\text{Positive Hit Rate} = \frac{\sum_{(a,b):\Lambda(a,b)=\Omega(a,b)} \Omega(a,b)}{\sum_{(a,b)} \Lambda(a,b)} \tag{4.4}$$

Finally, **KHAT** (or $\hat{K}$) (Wilkinson,2005)(Congalton,1999), which is an approximation to a coefficient called Kappa that takes the correct classifications by chance into account and is defined as

$$\text{KHAT}_{seg} = \frac{N \sum_{i=1}^{2} X_{ii} - \sum_{i=1}^{2} (X_{i+} X_{+i})}{N^2 - \sum_{i=1}^{2} (X_{i+} X_{+i})} \tag{4.5}$$

where

- N is the total number of samples in error matrix,
- X_{ii} is the number of observations for i^{th} row and i^{th} column in error matrix,
- X_{i+} is the marginal total of row i in error matrix ($\sum_{j=1}^{2} X_{ij}$),
- X_{+i} is the marginal total of column i in error matrix ($\sum_{j=1}^{2} X_{ji}$),

and Segmentation Error Matrix is

$$\text{SEM} = \begin{bmatrix} \sum_{(a,b):\Lambda(a,b)=\Omega(a,b)} 1 - \Omega(a,b) & \sum_{(a,b):\Lambda(a,b)\neq\Omega(a,b)} \Omega(a,b) \\ \sum_{(a,b):\Lambda(a,b)\neq\Omega(a,b)} 1 - \Omega(a,b) & \sum_{(a,b):\Lambda(a,b)=\Omega(a,b)} \Omega(a,b) \end{bmatrix} \tag{4.6}$$

The final performance, i.e. the performance after Shape Detection stage, is measured in a similar manner. The only difference between the two is that in the first one, the samples are image blocks, on the other hand, in the latter one samples are pixels. The Equations 4.7 – 4.10 defines the final performance expressions where $\nu(x, y)$ denotes the classification results and $\tau(x, y)$ denotes the ground truth on pixel (x, y). Both $\nu(x, y)$ and $\tau(x, y)$ gives 1 for positive (pixel that belongs to a runway) and 0 for negative (pixel that does not belong to a runway) samples.

$$\text{True Positive Rate} = \frac{\sum_{(x,y):\nu(a,b)=\tau(x,y)} \tau(x,y)}{\sum_{(x,y)} \tau(x,y)} \tag{4.7}$$

$$\text{True Negative Rate} = \frac{\sum_{(x,y):\nu(x,y)=\tau(x,y)} 1 - \tau(x,y)}{\sum_{(x,y)} 1 - \tau(x,y)} \tag{4.8}$$

$$\text{Positive Hit Rate} = \frac{\sum_{(x,y):\nu(x,y)=\tau(x,y)} \tau(x,y)}{\sum_{(x,y)} \nu(x,y)} \tag{4.9}$$

$$\text{KHAT}_{fin} = \frac{N \sum_{i=1}^{2} X_{ii} - \sum_{i=1}^{2} (X_{i+} X_{+i})}{N^2 - \sum_{i=1}^{2} (X_{i+} X_{+i})} \tag{4.10}$$

where Final Error Matrix is

$$\text{FEM} = \begin{bmatrix} \sum_{(x,y):\nu(x,y)=\tau(x,y)} 1 - \tau(x,y) & \sum_{(x,y):\nu(x,y)\neq\tau(x,y)} \tau(x,y) \\ \sum_{(x,y):\nu(x,y)\neq\tau(x,y)} 1 - \tau(x,y) & \sum_{(x,y):\nu(x,y)=\tau(x,y)} \tau(x,y) \end{bmatrix} \tag{4.11}$$

4.2 Feature Selection

This section is about the feature selection of the employed Adaboost algorithm. As mentioned before, there are no parameters required to tune for Adaboost except the iteration count. Since the calculations on an iteration does not effect the computation results obtained on the previous iterations, this parameter just determines how many classifiers to be taken into account and it is always possible to perform the classification with less weak learners if the results for such a number of classifiers are satisfactory. This implies that the tuning of this parameter does not cause critical consequences and can be selected due to computational constraints. In this study, training of Adaboost is done for 40 iterations, which is shown to be an iteration count less than 40 is enough for the solution of this problem.

Figure 4.1 shows the performance results on every iteration for the training set mentioned in the beginning of this chapter. As one may notice, the x-axis values of these results indicate the number of features (weak learners) utilized, while y-axis values show the corresponding performances (True Positive and True Negative rates) for that number. The "Number of Features" does not bear adequate information about which features are used. Therefore, features chosen by Adaboost algorithm for this run, are given in Table 4.1. In the table, names and parameters of the selected features are given with selection orders, indices in feature vector, Adaboost weights and classifier rules. The threshold values of classifier rules are normalized according to the minimum and maximum values of the corresponding feature in the training feature set in order to provide a more explanatory expression.

While it is not easy to state the possible selection reasonings for every selected feature, some of them are explicit. For instance the first feature in the table, the mean of Saturation, denotes how colorful the image block is, in average. Observing the classifier rule, the weak learner output is positive, if input is less colorful than the 16% of the saturation range. Since airport runways are not colorful structures and they tend to be in grayscale, this outcome is consistent with common sense. Likewise the second selected feature, variance of intensity gradient magnitude, which is selected multiple times, denotes how variable the rate of intensity level change is, in an image block. Higher values means the block must have abruptly changing neighboring pixels, as

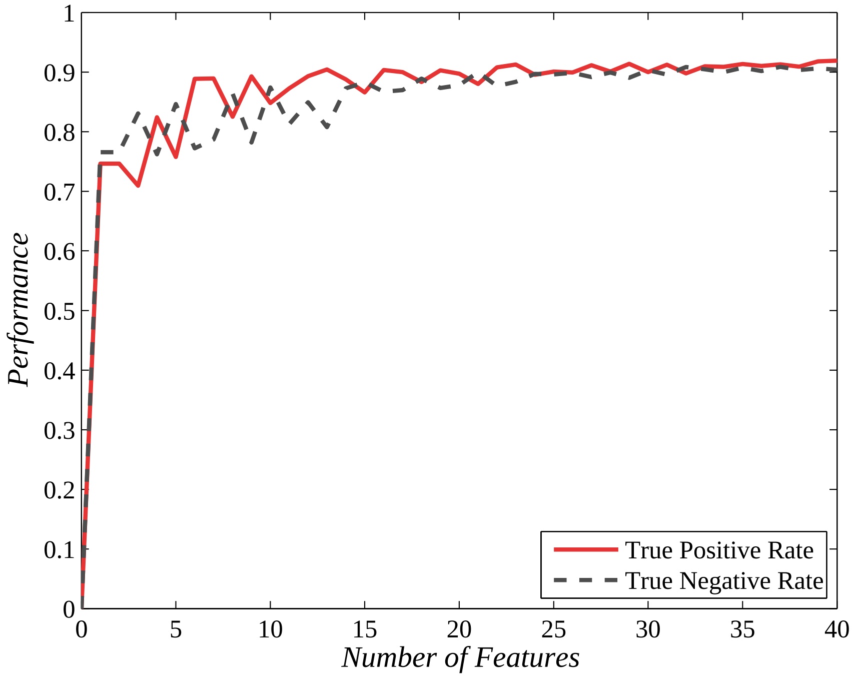


Figure 4.1: Performance vs Number of Features (Training Set)

well as uniform areas at the same time. This is the case when there is a runway sign or runway edge on a block (Figure A.1(a)).

Selection of duplicate features are possible in Adaboost. In the first glance this may seem to be causing a redundancy since there is no new information provided. However, in this way, more sophisticated classification rules can be established, because the thresholds and/or parities change on every iteration due to the training of weak classifiers.

For the computation of the performance of the test set, the same procedures applied on the training set are performed. Each one of the 48 test images is divided into blocks of size 32×32 and in this way 6869023 blocks are obtained for testing. The ground truth data, utilized for evaluation of classification results, contains 10403 runway blocks and 6858620 non-runway blocks. For composition of the test set, 7% of the negative blocks are randomly selected for use because of memory constraints, while full set of the positive blocks are utilized. Obtained classification results are compared to the

Table 4.1: Features selected by Adaboost and their selection orders

Order	**Index**	**Name and Parameter**	**Weight**	**Classifier Rule**
1	126	Mean of Saturation	1.13	< 0.16
2	4	Var. of Intensity Grad. Mag.	0.77	> 0.16
3	93	Haralick - Correlation (at 90°)	0.65	> 0.92
4	88	Haralick - Homogeneity (at 45°)	0.77	> 0.77
5	4	Var. of Intensity Grad. Mag.	0.51	> 0.22
6	129	Var. of Saturation Grad. Mag.	0.56	< 0.08
7	1	Mean of Intensity	0.49	> 0.58
8	131	Circ.-Mellin of Sat. (p=1, q=1)	0.33	> 0.09
9	86	Haralick - Energy (at 45°)	0.37	> 0.16
10	135	Var. of Value Grad. Mag.	0.43	> 0.33
11	59	Var. of Gabor Filt. Output (13^{th})	0.36	< 0.07
12	4	Var. of Intensity Grad. Mag.	0.4	> 0.11
13	35	Var. of Gabor Filt. Output (1^{st})	0.4	< 0.05
14	4	Var. of Intensity Grad. Mag.	0.37	> 0.26
15	128	Mean of Saturation Grad. Mag.	0.32	< 0.03
16	4	Var. of Intensity Grad. Mag.	0.35	> 0.09
17	77	Var. of Gabor Filt. Output (22^{th})	0.32	< 0.06
18	89	Haralick - Correlation (at 45°)	0.33	> 0.93
19	31	Average of DFT magnitude	0.27	> 0.18
20	53	Var. of Gabor Filt. Output (10^{th})	0.3	< 0.05
21	4	Var. of Intensity Grad. Mag.	0.29	> 0.42
22	115	Var. of Wavelet Output (St.3, LL)	0.31	> 0.06
23	124	Mean of Hue Grad. Mag.	0.28	< 0.11
24	126	Mean of Saturation	0.29	< 0.05
25	4	Var. of Intensity Grad. Mag.	0.27	> 0.14
26	59	Var. of Gabor Filt. Output (13^{th})	0.29	< 0.05
27	125	Var. of Hue Grad. Mag.	0.22	> 0.03
28	113	Var. of Wavelet Output (St.2, HH)	0.25	> 0.24
29	129	Var. of Saturation Grad. Mag.	0.27	< 0.1
30	131	Circ.-Mellin of Sat. (p=1, q=1)	0.27	> 0.19
31	31	Average of DFT magnitude	0.25	< 0.46
32	4	Var. of Intensity Grad. Mag.	0.23	> 0.48
33	122	Mean of Hue	0.23	> 0.11
34	108	Mean of Wavelet Output (St.2, LH)	0.18	< 0.01
35	4	Var. of Intensity Grad. Mag.	0.24	> 0.2
36	103	Var. of Wavelet Output (St.1, HL)	0.25	< 0.07
37	105	Var. of Wavelet Output (St.1, HH)	0.21	> 0.09
38	75	Var. of Gabor Filt. Output (21^{th})	0.2	< 0.08
39	115	Var. of Wavelet Output (St.3, LL)	0.17	> 0.13
40	123	Variance of Hue	0.19	< 0.03

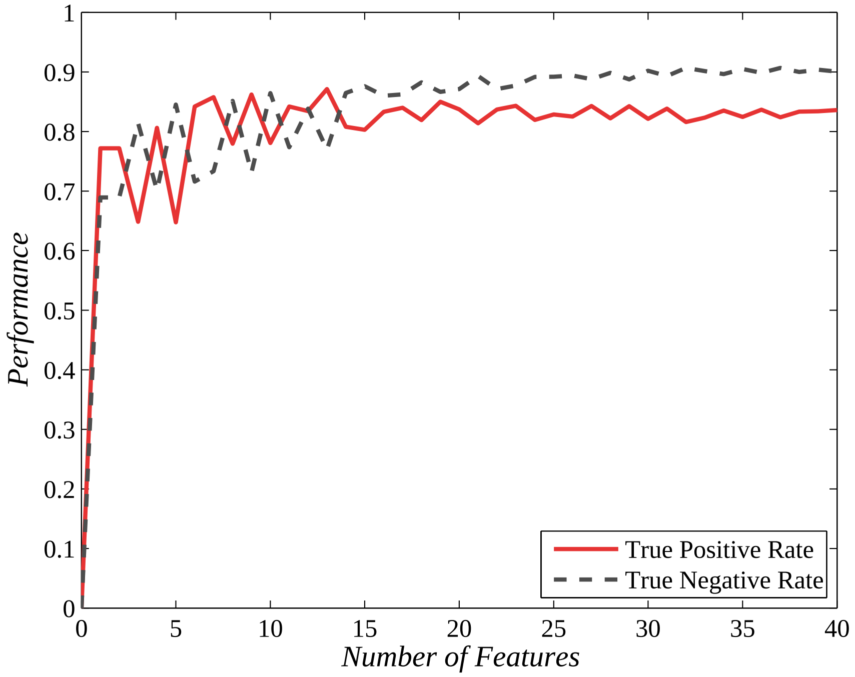


Figure 4.2: Performance vs Number of Features (Testing Set)

ground truth data to get performance results.

The performance results for the testing set are given in Figure 4.2. While the True Negative Rates are very close to the ones achieved for the training set, the True Positive Rate is slightly lower.

The outcomes of both sets show that a value of number of iterations around 16–20 gives comparable results with the value of 40. That is to say, after approximately 16–20 features, adding more features to the feature vector does not provide significant performance improvement. It is a question of computation capacity, whether to include remaining features after stated number or not. In this study, the subsequent stages are performed with data, generated by a classification process employing all of the features included.

4.3 Parameter Selection for Shape Detection

As mentioned previously, there exists a number of parameters utilized at the shape detection phase of the proposed algorithm. An analytic optimization of these parameters is quite difficult, because of the complexity of the proposed system. Therefore a set of experiments are carried out on a dozen of randomly selected satellite images to examine the consequences of the various selection of these parameters. The experimented parameters are Hough Peak Threshold, which determines the minimum peak magnitude for line pair detection in the Hough parameter space, Maximum Tolerated Gap, which determines the maximum gap permitted between two line segments for joining, and Minimum Runway Length, which determines the minimum length of a line pair to be counted as a runway.

Two criteria are considered for the evaluation of the experiments: True Positive Rate and Positive Hit Rate. True Negative Rate is excluded because of its insensitiveness due to the excessive amount of negative samples.

For the selection of Hough Peak Threshold parameter, three values are experimented: 100, 150 and 200. Maximum Tolerated Gap and Minimum Runway Length are kept constant between the experiments, which are chosen to be 100 and 400 respectively. The resulting performance graph is given in Figure 4.3. By examining the figure it can be observed that with the increasing values of Hough Peak Threshold, True Positive Rate decreases while Positive Hit Rate increases. Since both criteria are desired to be maximized, there is a trade off and a decision must be made. The value, where True Positive Rate is above 90% and Positive Hit Rate is above 10%, is chosen, which corresponds to a Hough Peak Threshold value of 150.

A similar procedure is followed for the Selection of Maximum Tolerated Gap parameter. Three values of Maximum Tolerated Gap are experimented: 50, 100 and 150. Hough Peak Threshold and Minimum Runway Length are kept constant between the experiments, which are chosen to be 150 and 400 respectively. The results are given in Figure 4.4. Examining the figure, yields that with the increasing values of Maximum Tolerated Gap, Positive Hit Rate decreases and True Positive Rate increases. A decision is made in a similar fashion with Hough Peak Threshold, which corresponds

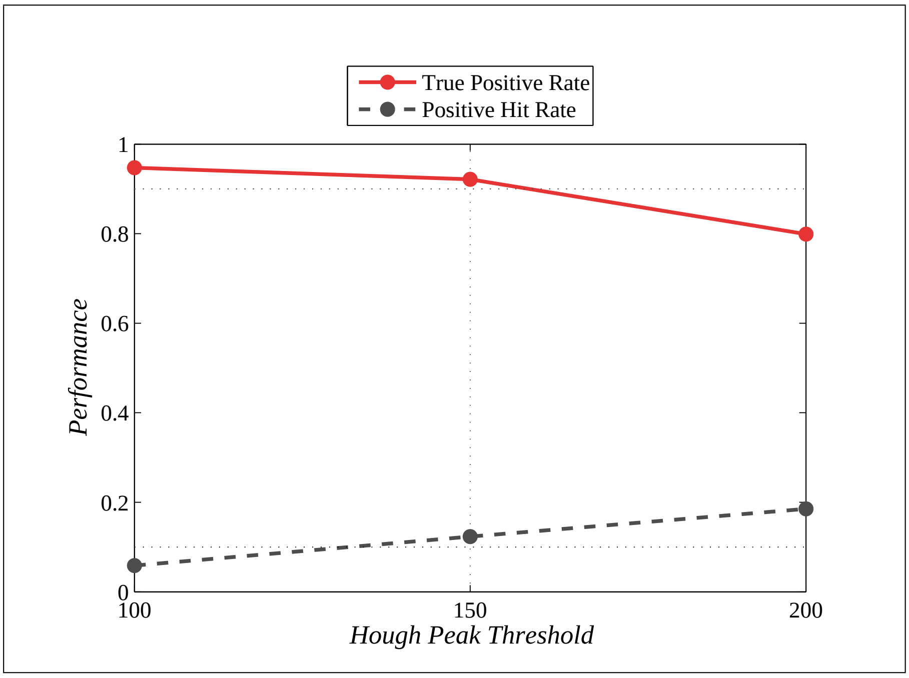


Figure 4.3: Performance vs Hough Peak Threshold

to a Maximum Tolerated Gap value of 100.

Selection of Minimum Runway Length parameter is made like the previous two experiments. Three values of Minimum Runway Length are experimented: 250, 400 and 550. Hough Peak Threshold and Maximum Tolerated Gap are kept constant between the experiments, which are chosen to be 150 and 100 respectively. In Figure 4.5, it can be observed that with the increasing values of Minimum Runway Length, True Positive Rate decreases and Positive Hit Rate increases. The decision of the value of Minimum Runway Length is made to be 400 according to the same reasonings given in the previous experiments.

4.4 Overall Performance

After determining the algorithm parameters, the proposed method is executed over the entire data set with these values and the results are obtained. Performances are calculated by using the formulae given in Section 4.1. The True Positive Rates, for both the results obtained after Segmentation and Shape Detection, are given in Table

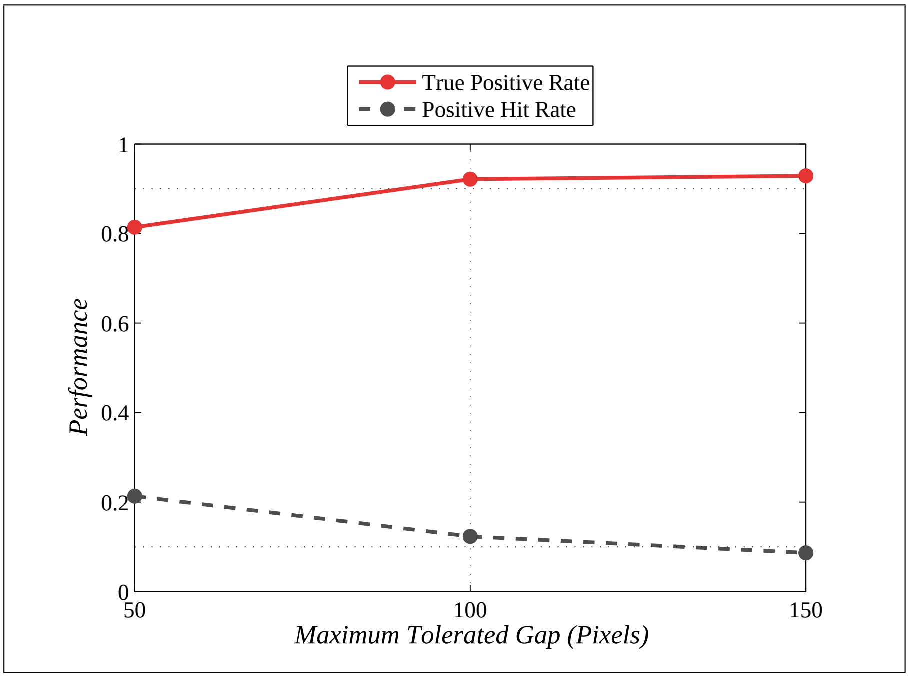


Figure 4.4: Performance vs Maximum Tolerated Gap

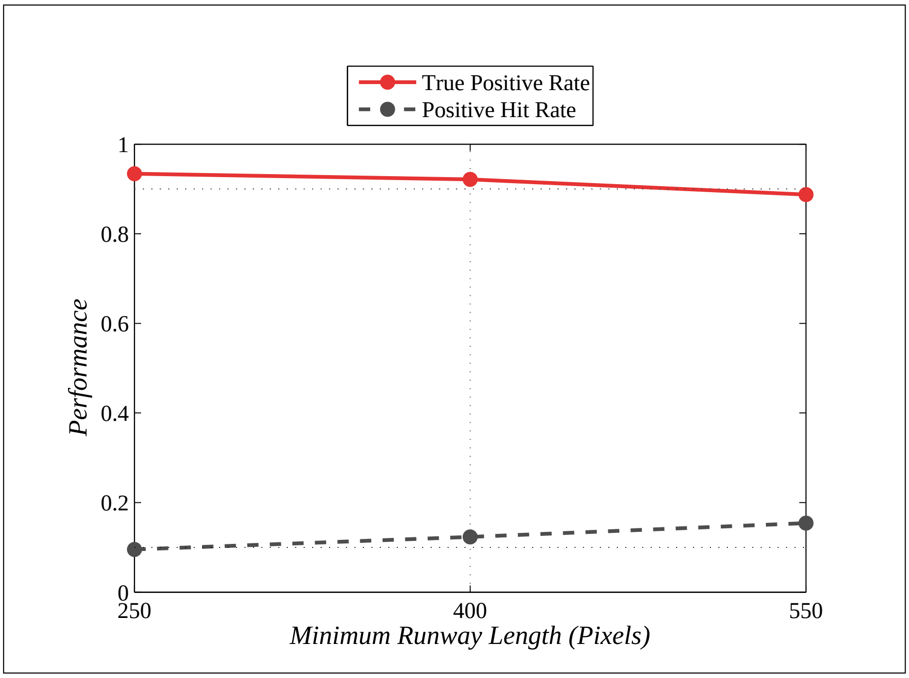


Figure 4.5: Performance vs Minimum Runway Length

Table 4.2: True Positive Rate

Data Set	Segmentation	Final
Training	.91899	.85759
Testing	.83601	.82613
Overall	.8662	.83754

Table 4.3: True Negative Rate

Data Set	Segmentation	Final
Training	.90378	.99632
Testing	.90002	.98996
Overall	.9015	.99245

4.2. The performances over the training and testing sets are also available in the table individually.

True Negative, Positive Hit Rate and KHAT are also given in Tables 4.3, 4.4 and 4.5 respectively.

In Tables 4.6 and 4.7 the results of segmentation stage is given in terms of number of image blocks for the first 40 and 20 features selected by Adaboost respectively.

In Table 4.8 the result of the final stage is given in terms of pixel numbers. It should be noted that the pixels belong to resized images as explained in Section 3.2. For comparison purposes with Table 4.8, the segmentation result is given in terms pixels, in Table 4.9.

Table 4.4: Positive Hit Rate

Data Set	Segmentation	Final
Training	.012703	.23856
Testing	.012525	.11131
Overall	.012593	.13881

Table 4.5: KHAT

Data Set	Segmentation	Final
Training	.0225	.3720
Testing	.0218	.1940
Overall	.0220	.2362

Table 4.6: Segmentation Results with the first 40 features (In terms of image blocks)

Training Set	Truly Positive	Truly Negative
Classified as Positive	5468	424968
Classified as Negative	482	3991793
Test Set	**Truly Positive**	**Truly Negative**
Classified as Positive	8697	685693
Classified as Negative	1706	6172927

Table 4.7: Segmentation Results with the first 20 features (In terms of image blocks)

Training Set	Truly Positive	Truly Negative
Classified as Positive	5338	539921
Classified as Negative	612	3876840
Test Set	**Truly Positive**	**Truly Negative**
Classified as Positive	8709	885979
Classified as Negative	1694	5972641

Table 4.8: Final results (after shape detection) with the first 40 features in terms of pixels)

Training Set	Truly Positive	Truly Negative
Classified as Positive	1303979	4162158
Classified as Negative	216544	1126531335
Test Set	**Truly Positive**	**Truly Negative**
Classified as Positive	2207062	17621540
Classified as Negative	464513	1738176773
Overall	**Truly Positive**	**Truly Negative**
Classified as Positive	3511041	21783698
Classified as Negative	681057	2864708108

Table 4.9: Segmentation Results with the first 40 features (In terms of pixels)

Training Set	**Truly Positive**	**Truly Negative**
Classified as Positive	1399808	108791808
Classified as Negative	123392	1021899008
Test Set	**Truly Positive**	**Truly Negative**
Classified as Positive	2226432	175537408
Classified as Negative	436736	1580269312

4.5 Example Results

Apart from the performance evaluation, three example results are given with step by step images to provide a better comprehension about the internals of the algorithm. The images are selected so that one of them is a good example, one of them has a poor positive hit rate (has numerous false alarms), and the other has a deficient true positive rate (detection of runway failed). These examples are Charlottetown Airport (CYYG) in Canada, Bagotville Airport (CYBR) in Canada and Coleman Aaf Airport (ETOR) in Germany. The four letter codes given in parantheses are International Civil Aviation Organization (ICAO) codes, representing the airports uniquely. Since the images are considerably large, these are given in Appendix A with the comments. The complete list of airports used in the data set are given in Appendix B, with ICAO code, IATA (International Air Transportation Association) code, name, city, country, coordinate and image size informations.

CHAPTER 5

CONCLUSION

A method for the detection of airport runways is proposed in this study. This method is based on an approach, which involves a segmentation process and a subsequent geometric analysis on the aerial image. In the segmentation phase, textural properties are considered, and mostly prevalent textural features that are used for segmentation in the literature are employed. Geometric examination stage is established upon a modified hough transform analysis which is a novel technique. In addition to that, employing Adaboost learning algorithm and utilization of features obtained by using HSV color space, Gabor Filters, Fourier Power Spectrum Analysis and Wavelets, are also original works for the airport runway detection problem.

It is observed that in the Ababoost training process, the majority of selected features (24 out of 40) are the mentioned new features (Table 4.1), demonstrating that they are convinient features for runway detection. Haralick features that are previously used for automatic road detection applications are also selected by Adaboost (4 out of 40), which is an expected result due to the similarity of textural properties of roads and runways. In Table 4.1, one may notice there are 26 different features, and repetitions exist. The repeated occurances of some features, at first, may be interpreted as redundant selections. However, the utilized weak classifiers are threshold classifiers explained in Section 2.2, which decides by judging whether the value of input is "above" or "below" a threshold. Since the weak classifiers are redetermined on every iteration, multiple selection allows the definition of more complex rules.

The experiments are carried out on a 64-bit MATLAB environment, on a dual Xeon 2.0 GHz workstation with 4 GBs of memory, running Linux x64 operating system.

Extracting all of the 137 features from a 14400 × 11247 image, takes approximately 115 minutes, where extracting the selected 26 unique features, which is the case for all of the images after training, takes 78 minutes by using this system. Keeping the size of the data and the platform in mind, algorithm performs fairly well. A performance improvement alternative is given in Section 5.1. It is also always possible to decide the number of features to be used with a trade off between performance and computation time, due to the scalability provided by Adaboost. After the training, all blocks of an image takes approximately 5 seconds to be classified.

Hough transform, which the proposed modified hough transform is based upon, is a relatively a cost-effective method for line detection. Together with the convenient techniques employed succeeding to hough transform, running times of approximately 7 minutes (including edge detection) is measured. It can be inferred that the proposed algorithm is computationally efficient, again considering the amount of data processed.

Unlike the previous studies in the literature, the proposed method is experimented with a large dataset consisting of heavily negative samples. Promising True Positive Rate and True Negative Rates are obtained and given in Section 4.4 in terms of training, test and combined sets that are used in Adaboost. However, the Positive Hit Rate mentioned in the same section indicates that only one runway out of approximately seven detected runways is a true runway. Although this result may seem poor, when the dominance of negative samples are taken into account, such an outcome is tolerable and it can be concluded that the proposed method is successful.

5.1 Future Work

As future work, further shape analysis can be carried out in order to eliminate false positives. Since it is observed that the algorithm occasionally misinterprets the highways or other wide roads as runways, these structures' interconnected network may be detected and eliminated by such an analysis. For another instance, fundamental airport buildings, such as control towers, terminal buildings or hangars, may be searched to conclude whether the detected structure is truly an airport or not.

Since stereographic image pairs, or remote sensed data like infrared, LIDAR (Light

Detection and Ranging) or SAR (Synthetic Aperture Radar) significantly helps by providing informations such as elevation, presence of vegetation or material properties, inclusion of such channels in the feature extraction process would surely contribute the accuracy of the outcomes of the segmentation stage. The importance of this concept can be observed in the example given by Figures A.10 – A.17.

The proposed system is suitable for parallel processing in terms of feature extraction process. Since the non-overlapping image blocks, used for segmentation, are independent from each other, the feature extraction can be carried out by different processing units for faster calculation.

Segmentation process can also be modified with a multi-class Adaboost learning algorithm, so that it can serve as a general purpose region of interest detector, for a multipurpose automatic target detection system. This improvement provides an efficiency enhancement due to the unification of the detection of the regions of interest operations for various targets.

REFERENCES

[1] M.F. Augusteijn, L.E. Clemens, and K.A. Shaw. Performance evaluation of texture measures for ground cover identification in satellite images by means of a neural network classifier. *Geoscience and Remote Sensing, IEEE Transactions on*, 33(3):616–626, May 1995.

[2] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 714–725, 1987.

[3] Matjaz Bevk and Igor Kononenko. A statistical approach to texture description of medical images: A preliminary study. *Computer-Based Medical Systems, IEEE Symposium on*, 0:239, 2002.

[4] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[5] Russell G. Congalton and Kass Green. *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices.* Lewis Publishers, New York, NY, USA, 1999.

[6] Richard O. Duda and Peter E. Hart. Use of hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11 – 15, 1972.

[7] C. Fernandez-Maloigne, D. Laugier, and A. Bekkhoucha. Texture analysis for road detection. In *Intelligent Vehicles '92 Symposium., Proceedings of the*, pages 219–224, Jun-1 Jul 1992.

[8] Yoav Freund and Robert E. Schapire. A short introduction to boosting. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kaufmann, 1999.

[9] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[10] Pranav Gupta and Anupam Agrawal. Airport detection in high-resolution panchromatic satellite images. *Journal of Institution of Engineers (India)*, 88(5):3–9, 2007.

[11] JunWei Han, Lei Guo, and YongSheng Bao. A method of automatic finding airport runways in aerial images. In *Signal Processing, 2002 6th International Conference on*, volume 1, pages 731–734 vol.1, Aug. 2002.

[12] Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6):610–621, Nov. 1973.

[13] A. Khotanzad and Y.H. Hong. Invariant image recognition by zernike moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(5):489–497, May 1990.

[14] Chungan Lin and Ramakant Nevatia. Building detection and description from a single intensity image. *Comput. Vis. Image Underst.*, 72(2):101–121, 1998.

[15] Dehong Liu, Lihan He, and L. Carin. Airport detection in large aerial optical imagery. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 5, pages V–761–4 vol.5, May 2004.

[16] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(8):837–842, Aug 1996.

[17] H. Mayer, S. Hinz, U. Bacher, and E. Baltsavias. A test of automatic road extraction approaches. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 209 – 214, 2006.

[18] J. B. Mena. State of the art on automatic road extraction for gis update: a novel classification. *Pattern Recognition Letters*, 24(16):3037 – 3058, 2003.

[19] J. B. Mena and J. A. Malpica. An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery. *Pattern Recogn. Lett.*, 26(9):1201–1220, 2005.

[20] M. Mokhtarzade, M.J.V. Zoej, and H. Ebadi. Automatic road extraction from high resolution satellite images using neural networks, texture analysis, fuzzy clustering and genetic algorithms. In *The international archives of the photogrammetry, remote sensing and spatial information sciences, ISPRS Congress Beijing*, page B3b: 549 ff, 2008.

[21] S. D. Newsam and C. Kamath. Retrieval using texture features in high resolution multi-spectral satellite imagery. In *SPIE Defense and Security Symposium, Data Mining and Knowledge Discovery: Theory, Tools, and Technology VI*, 2004.

[22] S. D. Newsam and C. Kamath. Comparing shape and texture features for pattern recognition in simulation data. In E. R. Dougherty, J. T. Astola, and K. O. Egiazarian, editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5672 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 106–117, March 2005.

[23] X. Niu. A semi-automatic framework for highway extraction and vehicle detection based on a geometric deformable model. *International Journal of Photogrammetry and Remote Sensing*, 61:170–186, December 2006.

[24] F. O'Gorman and M.B. Clowes. Finding picture edges through collinearity of feature points. *Computers, IEEE Transactions on*, C-25(4):449–456, April 1976.

[25] Yiming Pi, Luhong Fan, and Xiaobo Yang. Airport detection and runway recognition in sar images. In *Geoscience and Remote Sensing Symposium, 2003. IGARSS '03. Proceedings. 2003 IEEE International*, volume 6, pages 4007–4009 vol.6, July 2003.

[26] Dan Popescu, Radu Dobrescu, and Daniel Merezeanu. Road analysis based on texture similarity evaluation. In *SIP'08: Proceedings of the 7th WSEAS International Conference on Signal Processing*, pages 47–51, Stevens Point, Wisconsin, USA, 2008. World Scientific and Engineering Academy and Society (WSEAS).

[27] Yanyun Qu, Cuihua Li, and Nanning Zheng. Airport detection base on support vector machine from a single image. In *Information, Communications and Signal Processing, 2005 Fifth International Conference on*, pages 546–549, 0-0 2005.

[28] R.M. Rangayyan, R.J. Ferrari, J.E.L. Desautels, and A.F. Frere. Directional analysis of images with gabor wavelets. In *Computer Graphics and Image Processing, 2000. Proceedings XIII Brazilian Symposium on*, pages 170–177, 2000.

[29] G. Ravichandran and M.M. Trivedi. Circular-mellin features for texture segmentation. *Image Processing, IEEE Transactions on*, 4(12):1629–1640, Dec 1995.

[30] Ann Scher, Michael Shneier, and Azriel Rosenfeld. A method for finding pairs of antiparallel straight lines. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-4(3):316–323, May 1982.

[31] S. Udomhunsakul. Semi-automatic road detection from satellite imagery. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 3, pages 1723–1726 Vol. 3, Oct. 2004.

[32] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[33] Qiu-Ting Wang and Ye Bin. A new recognizing and understanding method of military airfield image based on geometric structure. In *Wavelet Analysis and Pattern Recognition, 2007. ICWAPR '07. International Conference on*, volume 3, pages 1241–1246, Nov. 2007.

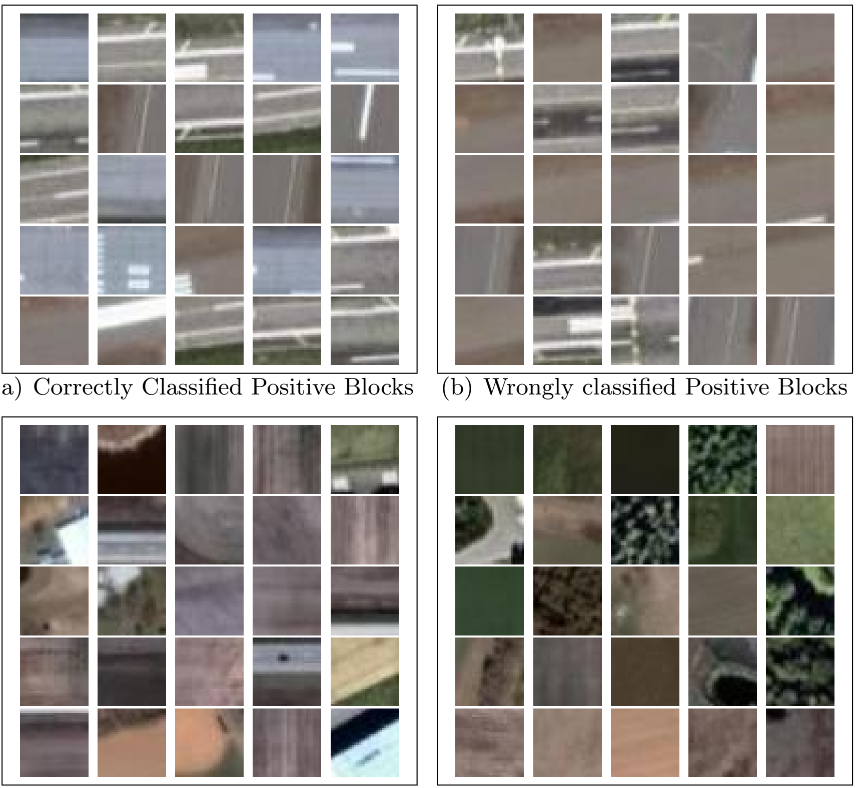
[34] J. S. Weszka and A. Rosenfeld. A comparative study of texture measures for terrain classification. *NASA STI/Recon Technical Report N*, 76, March 1975.

[35] G.G. Wilkinson. Results and implications of a study of fifteen years of satellite image classification experiments. *Geoscience and Remote Sensing, IEEE Transactions on*, 43(3):433–440, March 2005.

[36] I. Young, J. Gerbrands, and L. van Vliet. Fundamentals of image processing, 1995.

APPENDIX A

IMAGES OF EXAMPLE RESULTS



(a) Correctly Classified Positive Blocks (b) Wrongly classified Positive Blocks

(c) Wrongly Classified Negative Blocks (d) Correctly classified Negative Blocks

Figure A.1: Example classified blocks.

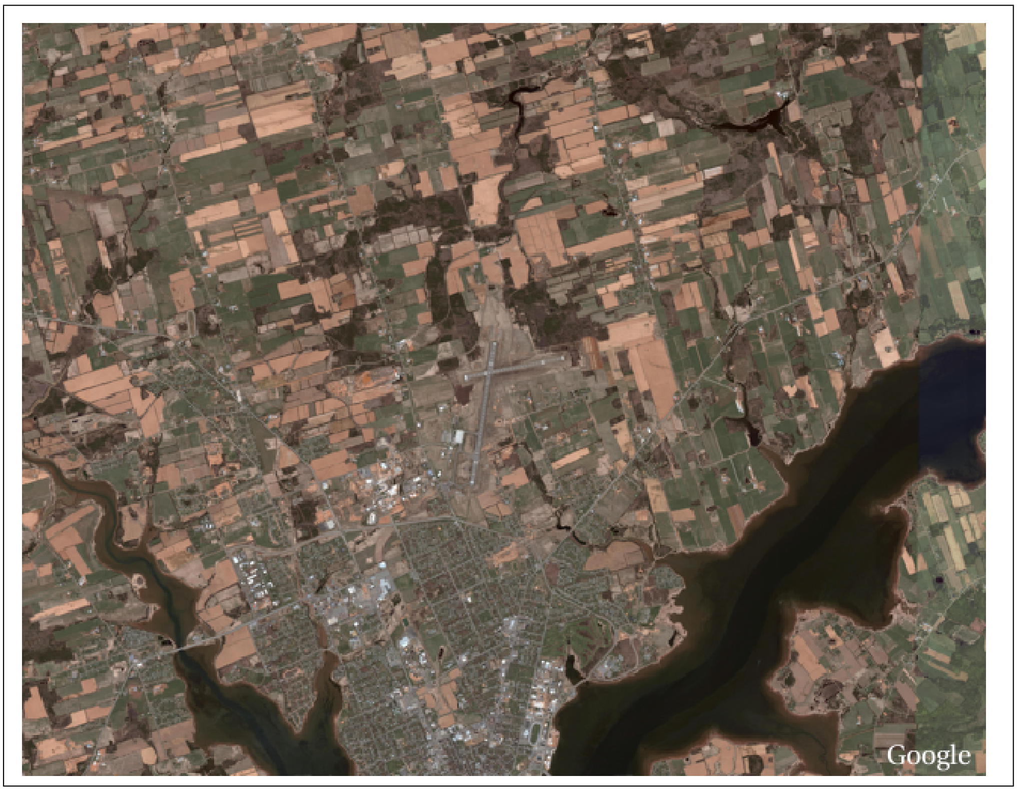


Figure A.2: CYYG – Original Image. (©2007 Google ^{TM} – Imagery ©2009 DigitalGlobe, GeoEye)

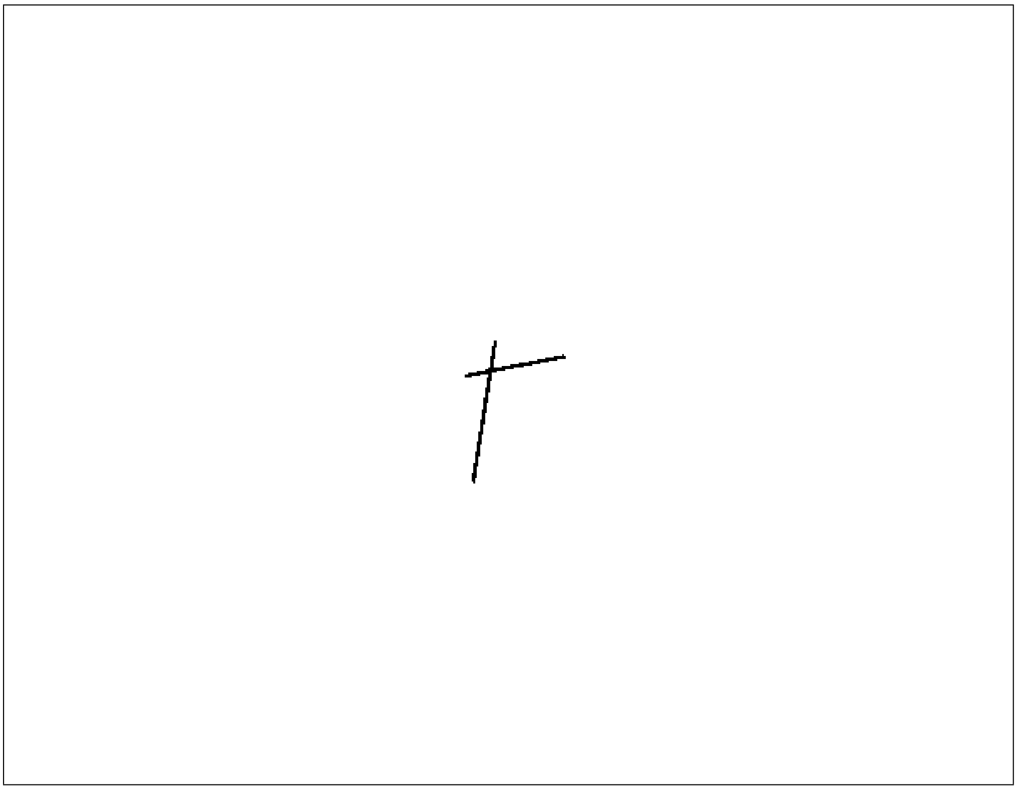


Figure A.3: CYYG – Ground truth data

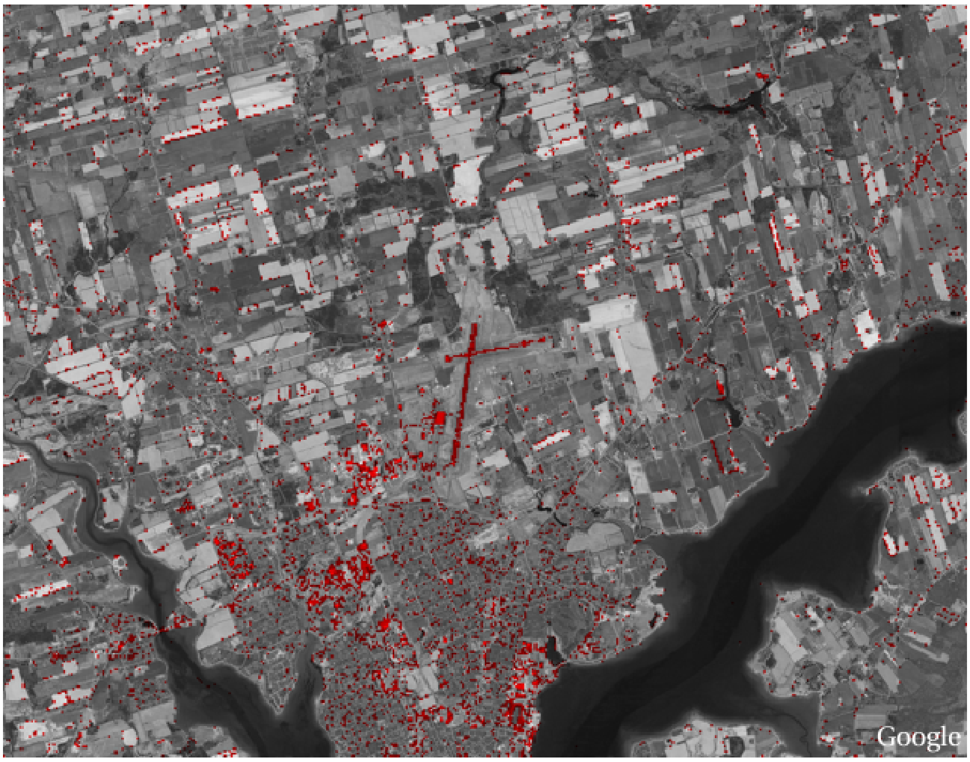


Figure A.4: CYYG – Segmentation result

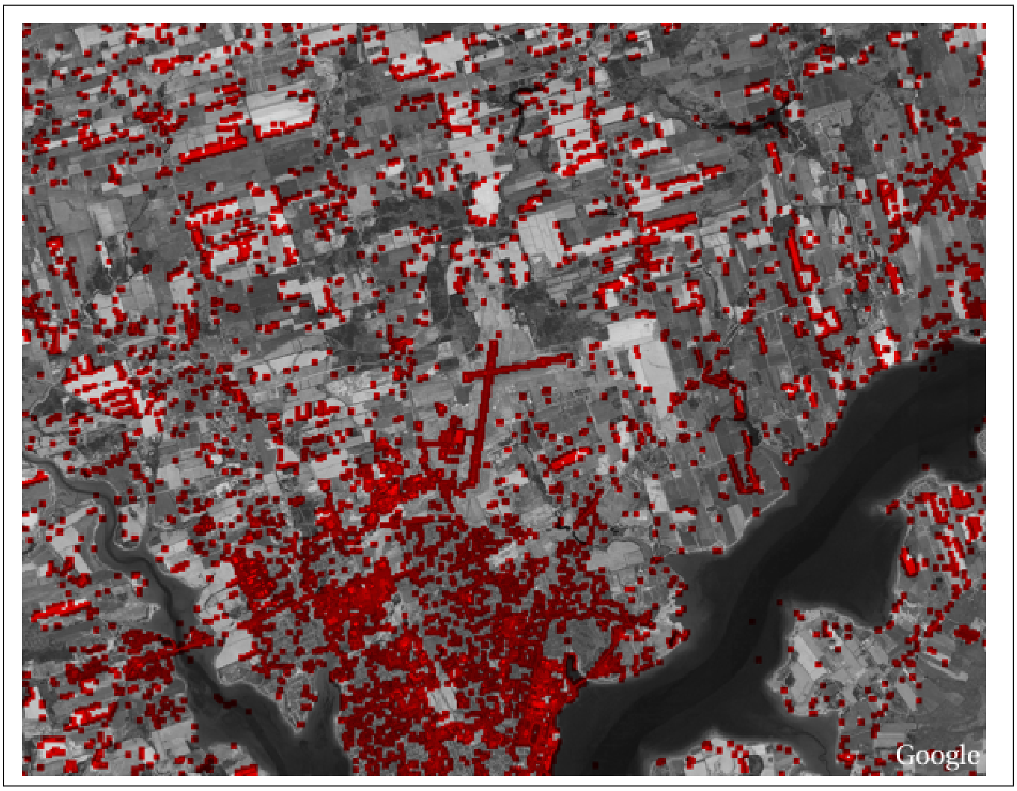


Figure A.5: CYYG – Expansion of regions of interest

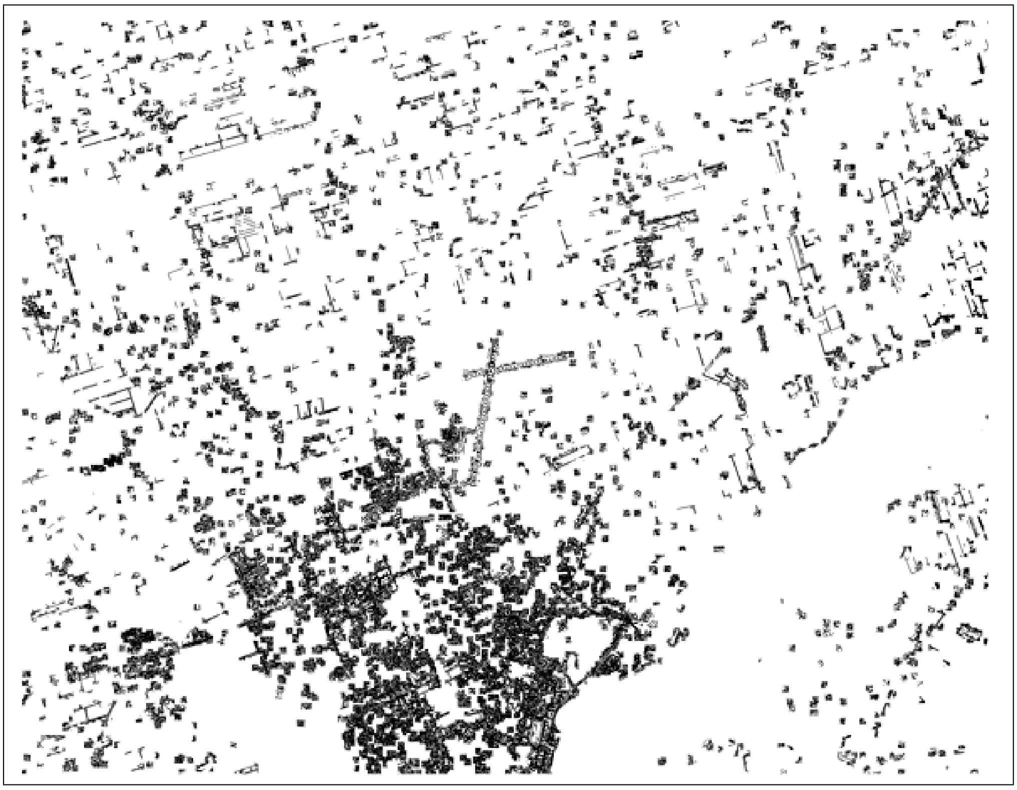


Figure A.6: CYYG – Edge detection result

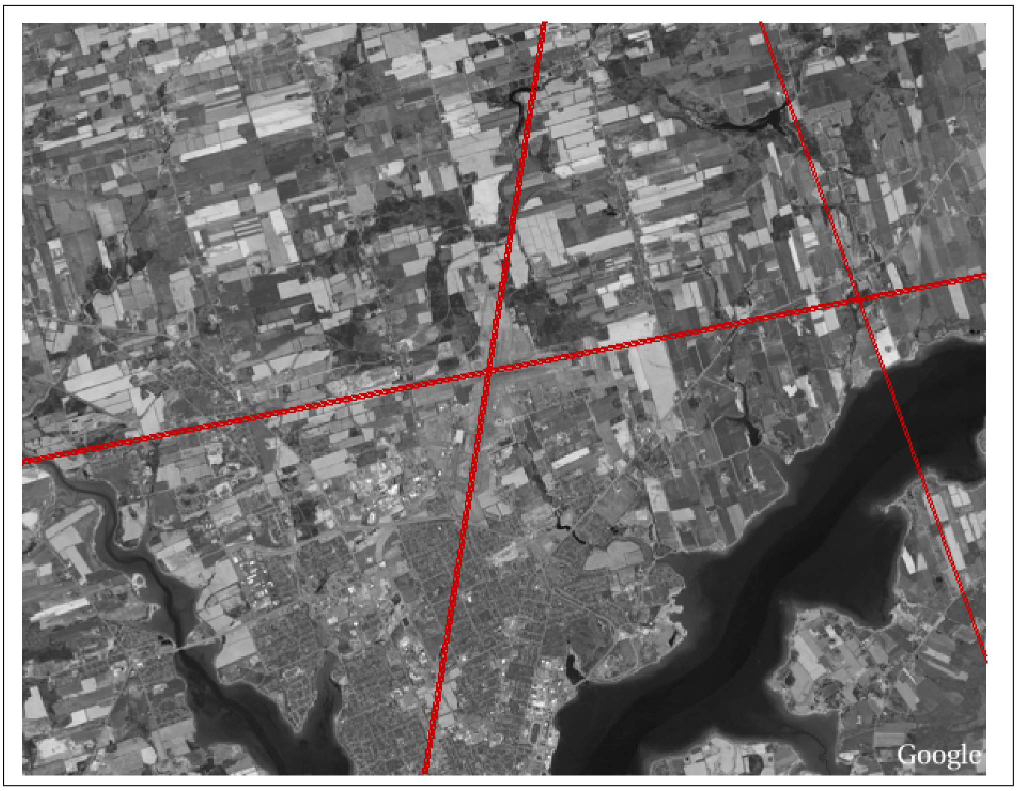


Figure A.7: CYYG – Detected line pairs (overlaid on original image)

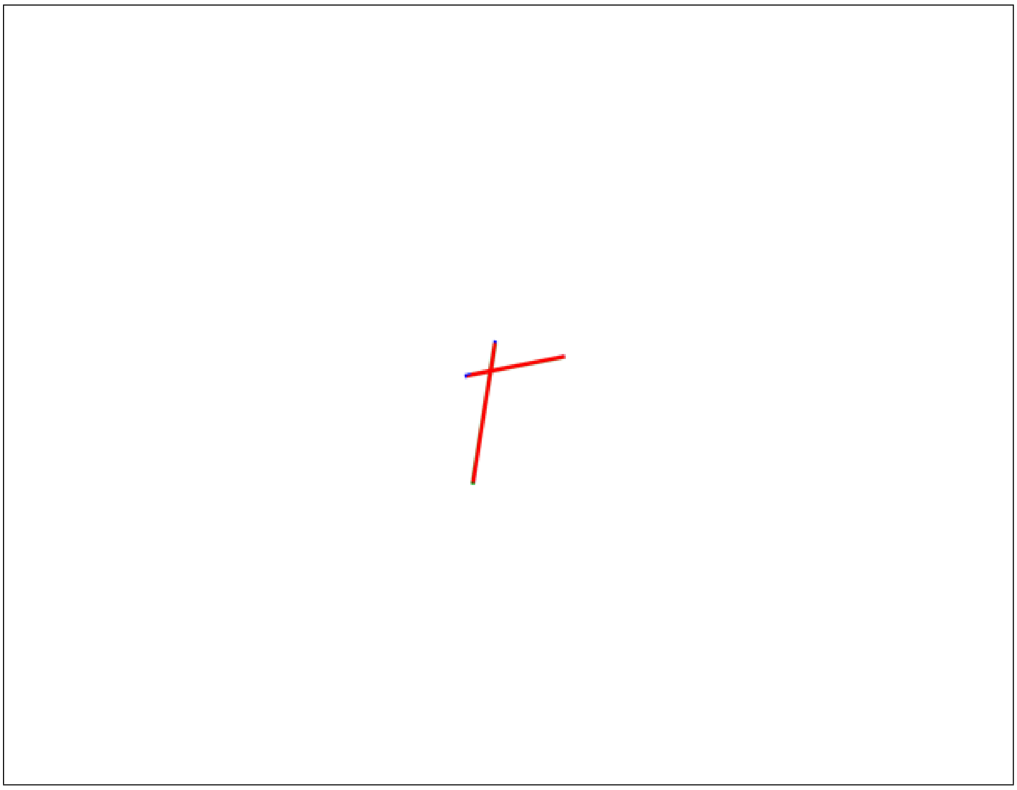


Figure A.8: CYYG – Final result (Red: True Positive, Green: False Positive, Blue: False Negative, White: True Negative)

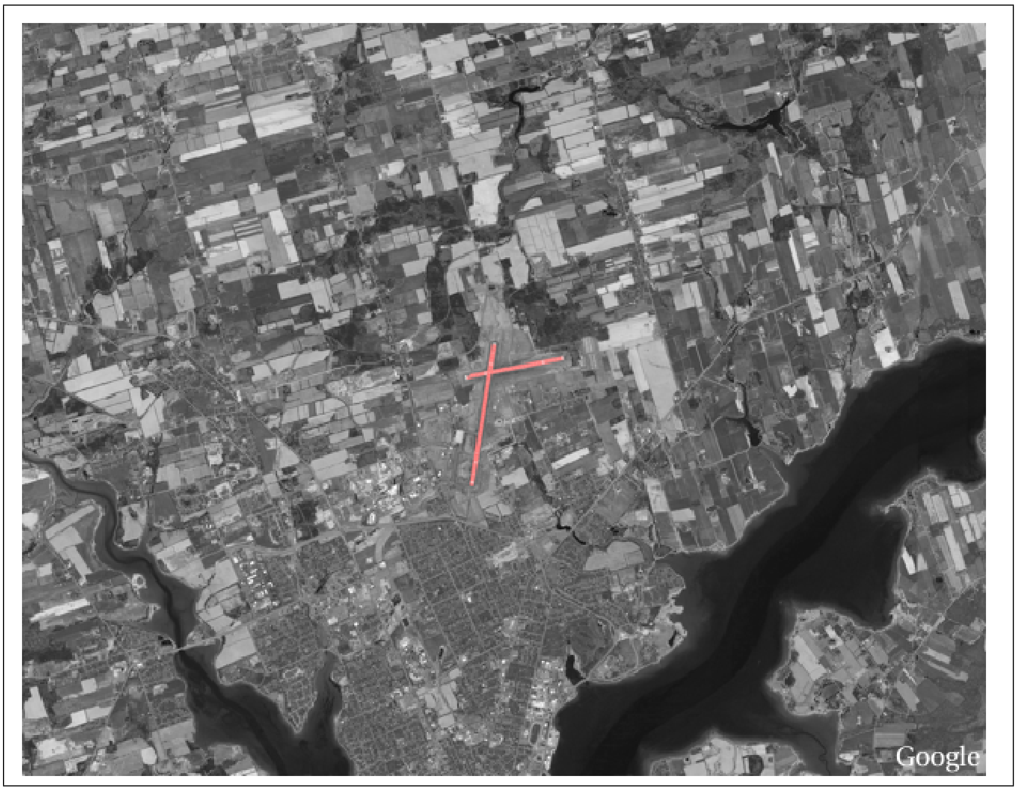


Figure A.9: CYYG – Final result (overlaid on original image). *A quite satisfying result containing almost no false positives or false negatives.*

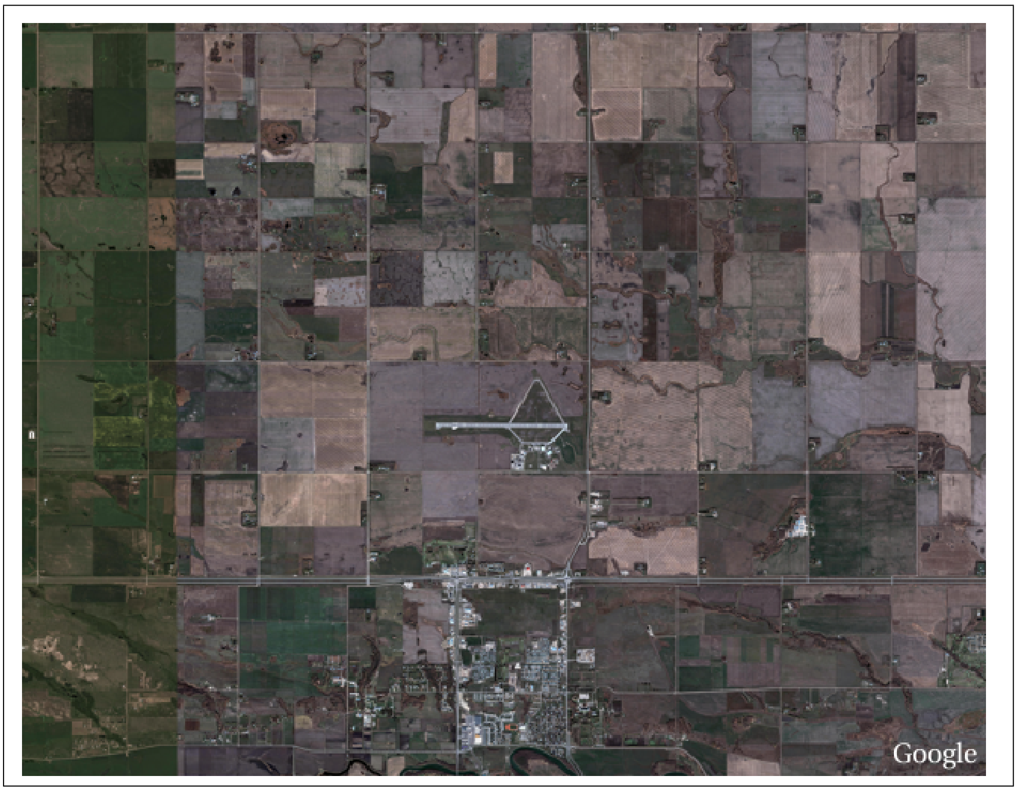


Figure A.10: CYBR – Original Image. (©2007 Google ^{TM} – Imagery ©2009 DigitalGlobe)

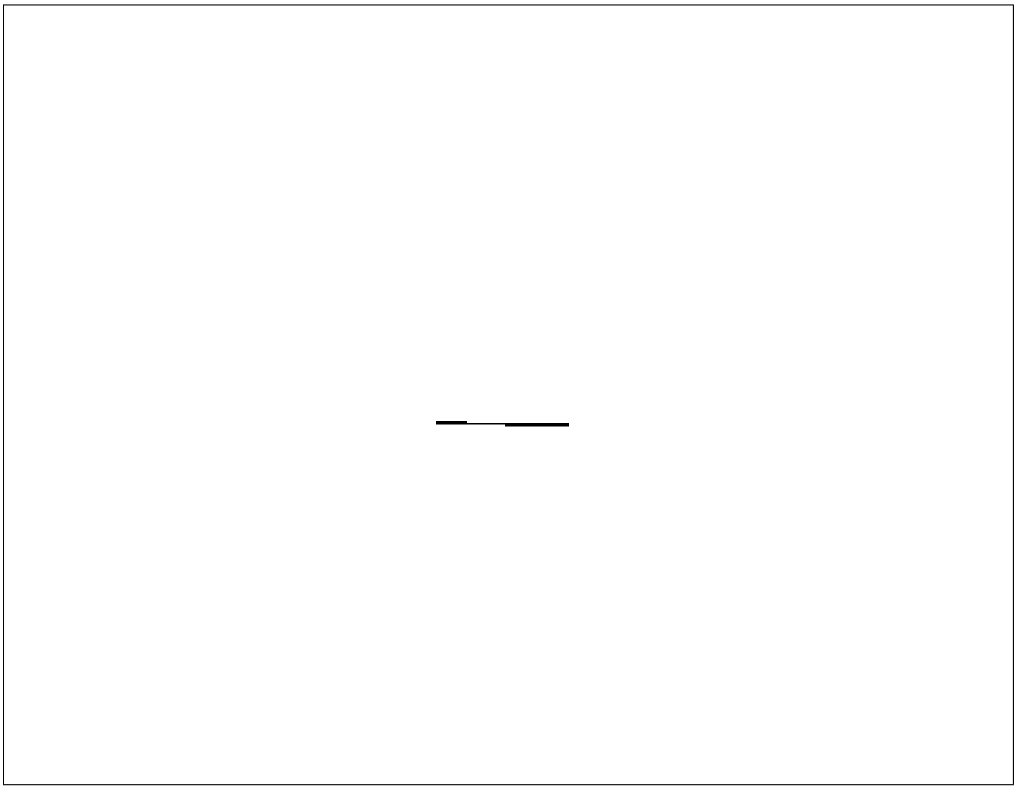


Figure A.11: CYBR – Ground truth data

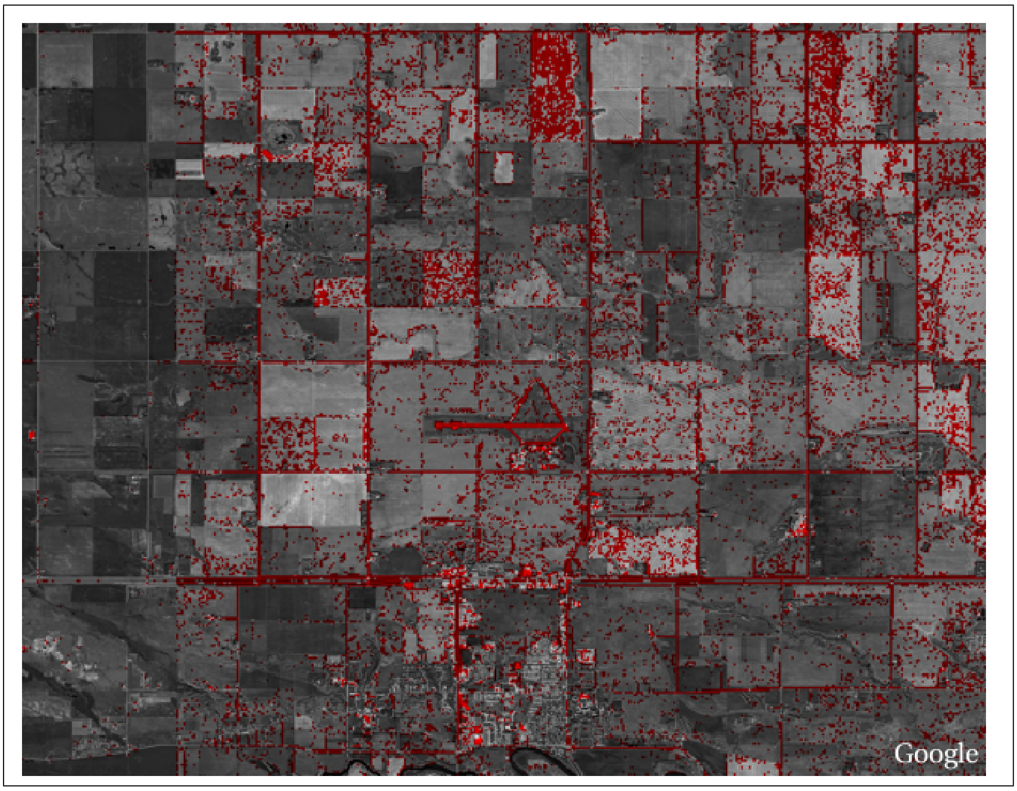


Figure A.12: CYBR – Segmentation result. *A considerable amount of false positive blocks, are on vegetation areas and other roads.*

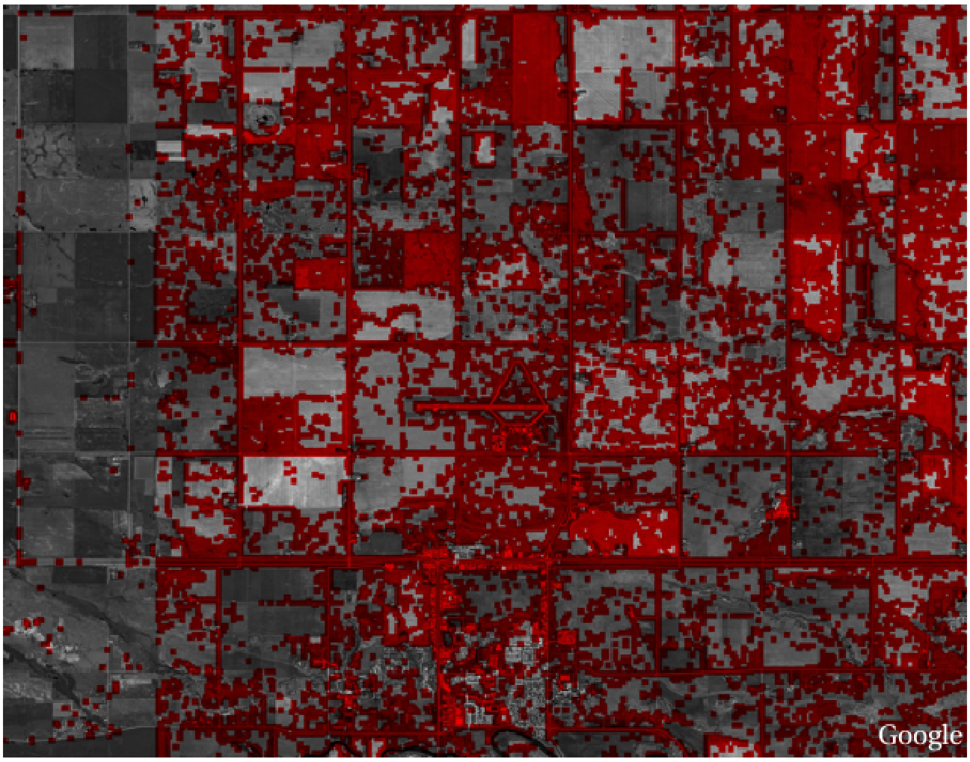


Figure A.13: CYBR – Expansion of regions of interest

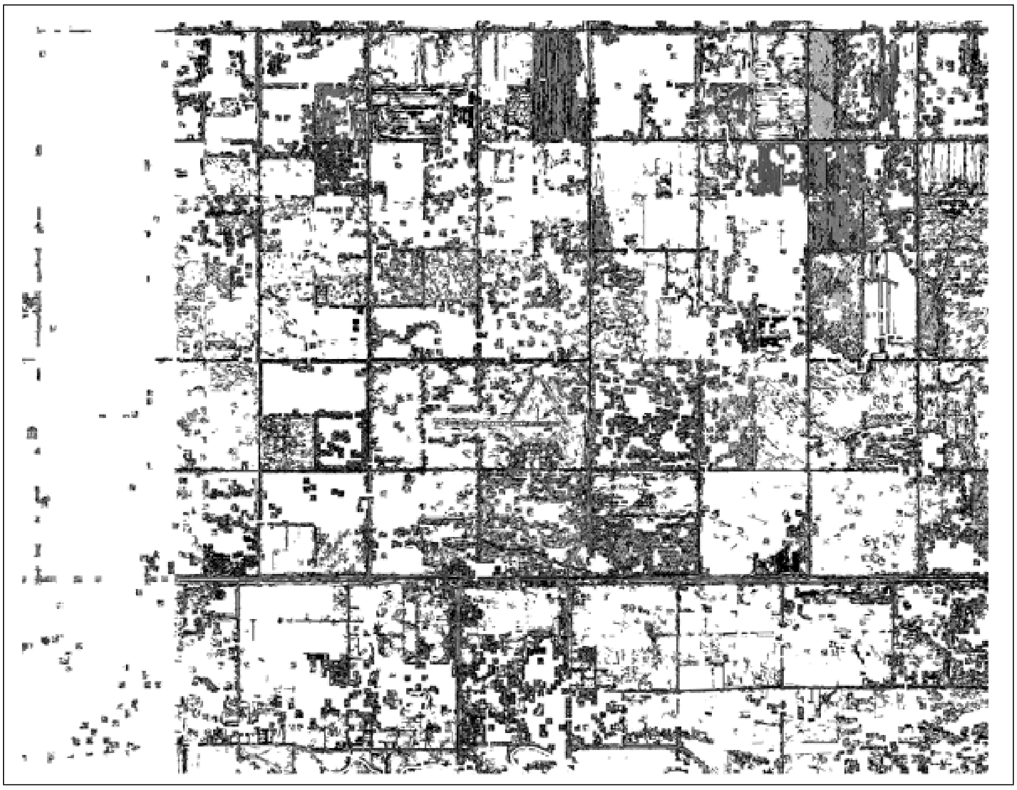


Figure A.14: CYBR – Edge detection result.

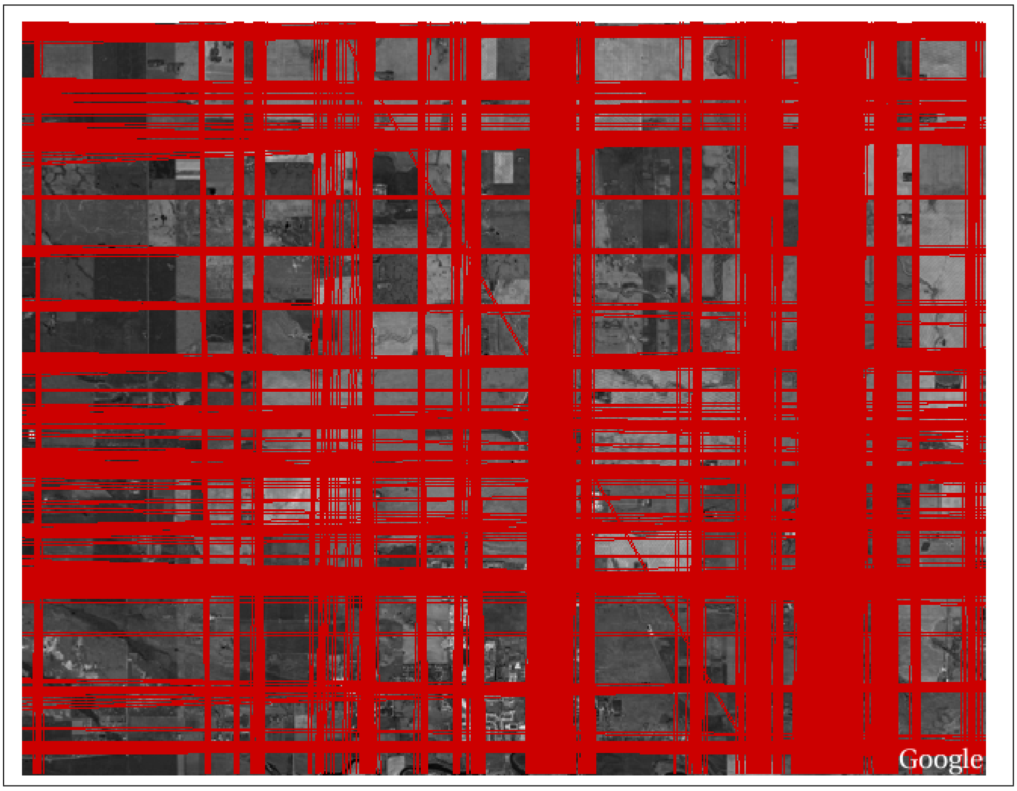


Figure A.15: CYBR – Detected line pairs (overlaid on original image). *The long straight roads and systematically plowed fields in the area generate numerous sets of collinear edge points, which results in excessive number of lines.*

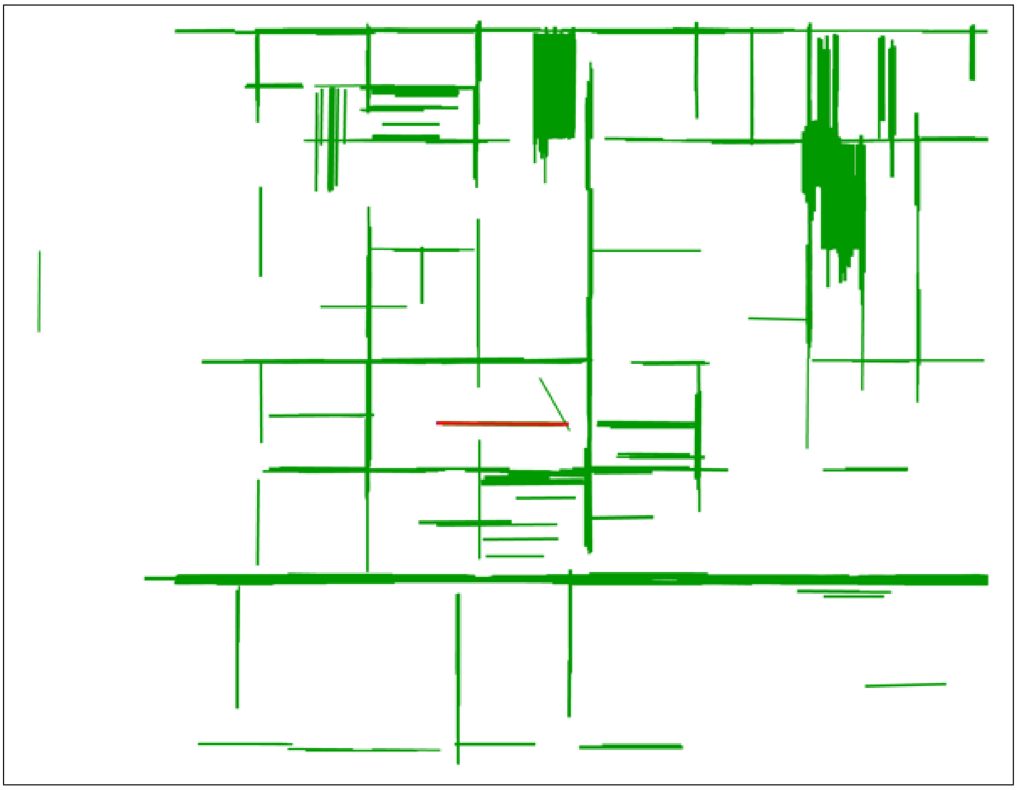


Figure A.16: CYBR – Final result (Red: True Positive, Green: False Positive, Blue: False Negative, White: True Negative)

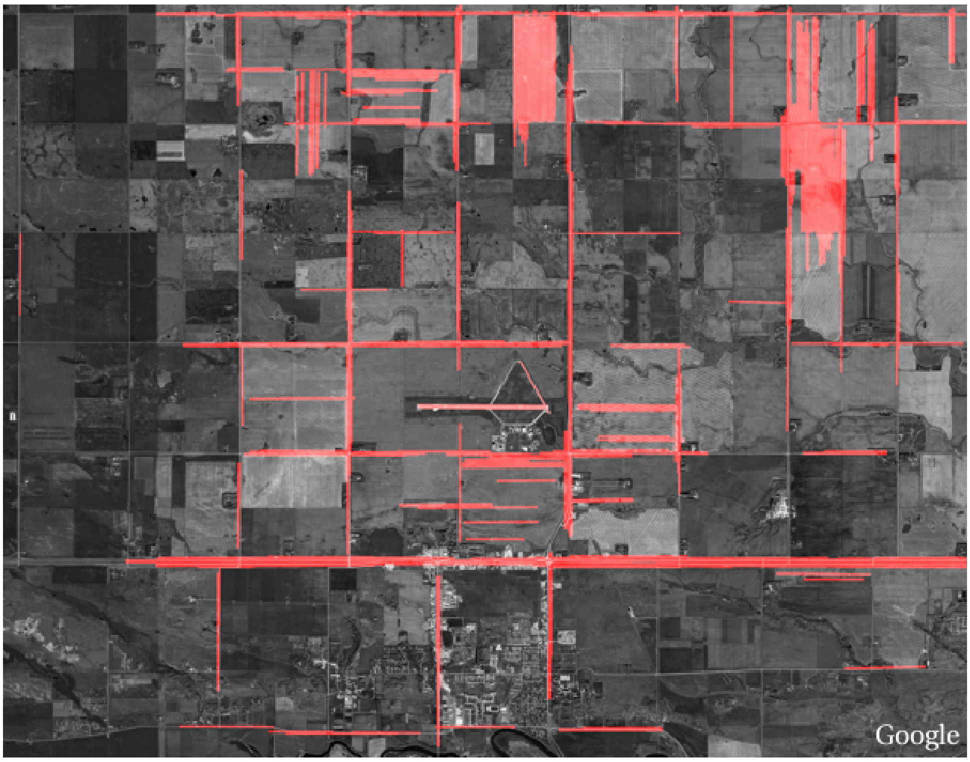


Figure A.17: CYBR – Final result (overlaid on original image). *Large number of lines, results in large number of line segments and thus false positives.*

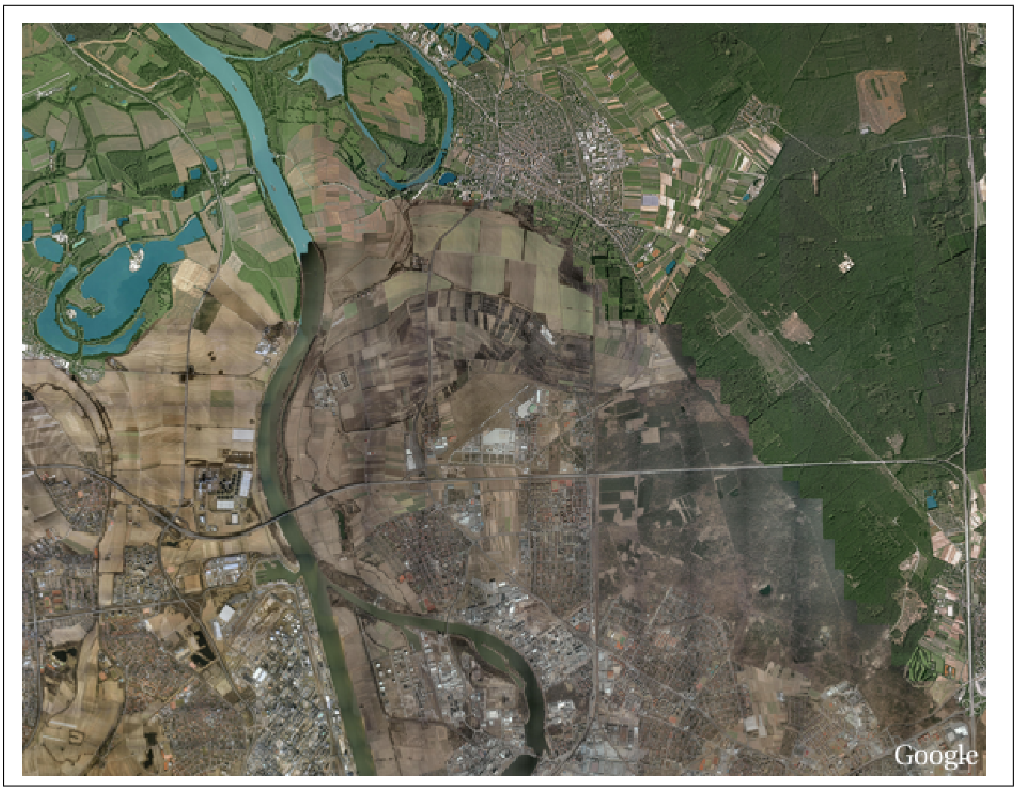


Figure A.18: ETOR – Original Image. (©2007 Google ^{TM} – Imagery ©2009 GeoContent, AeroWest)

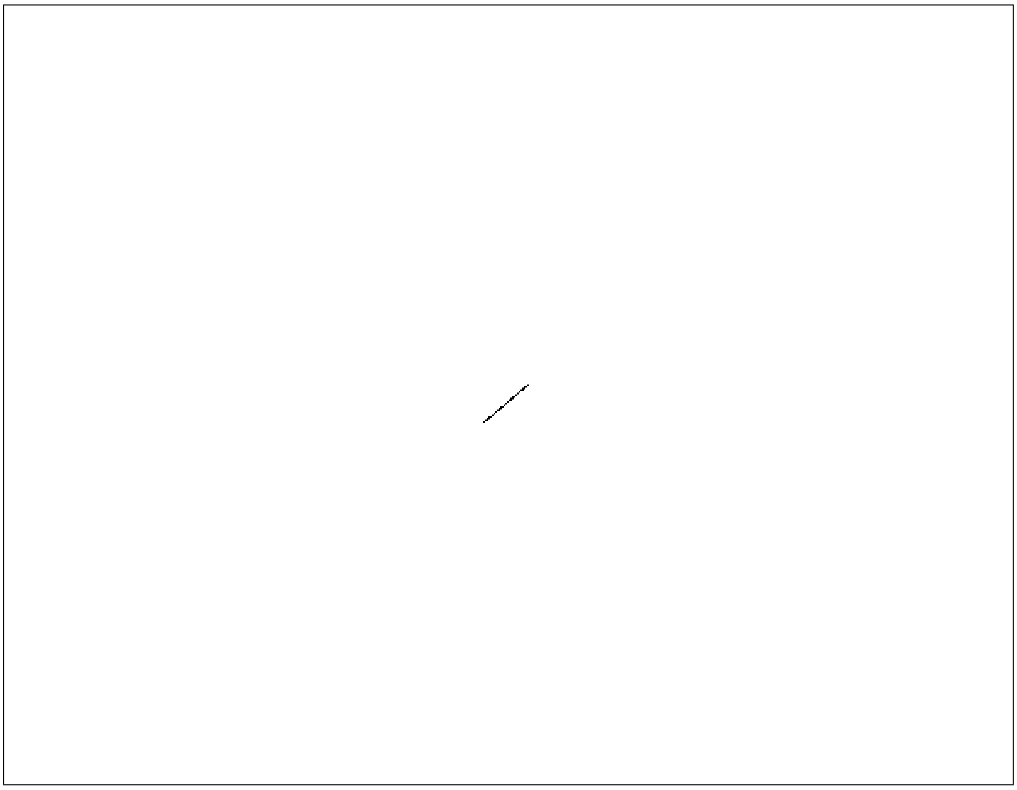


Figure A.19: ETOR – Ground truth data

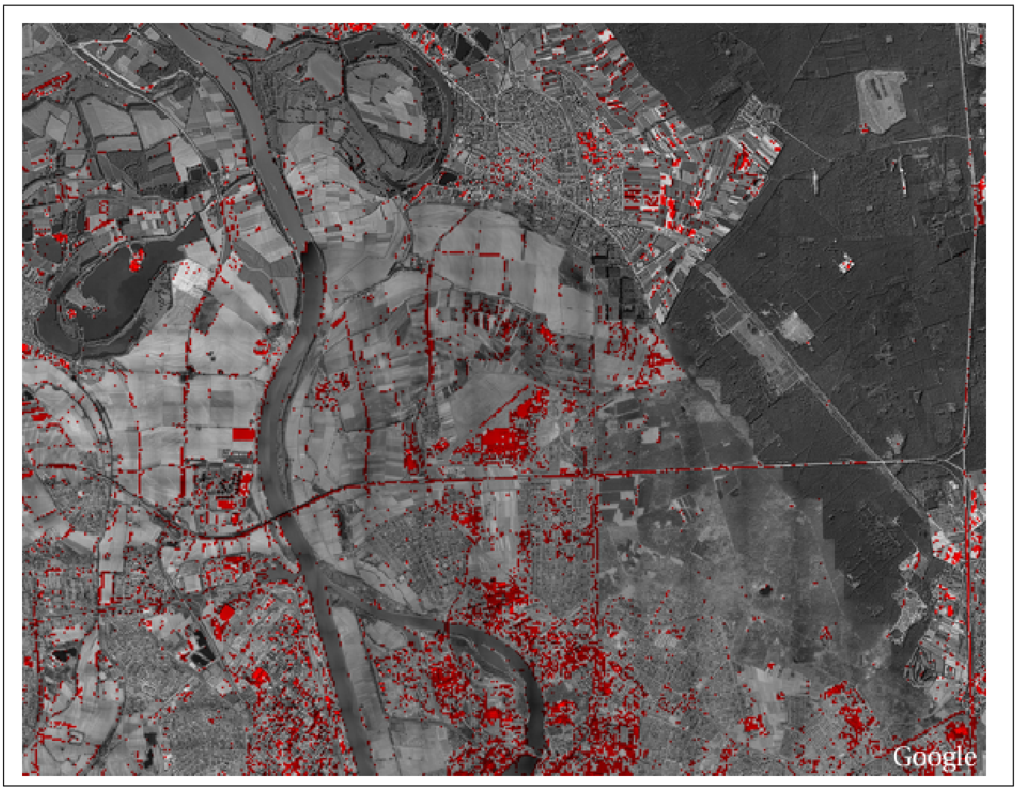


Figure A.20: ETOR – Segmentation result

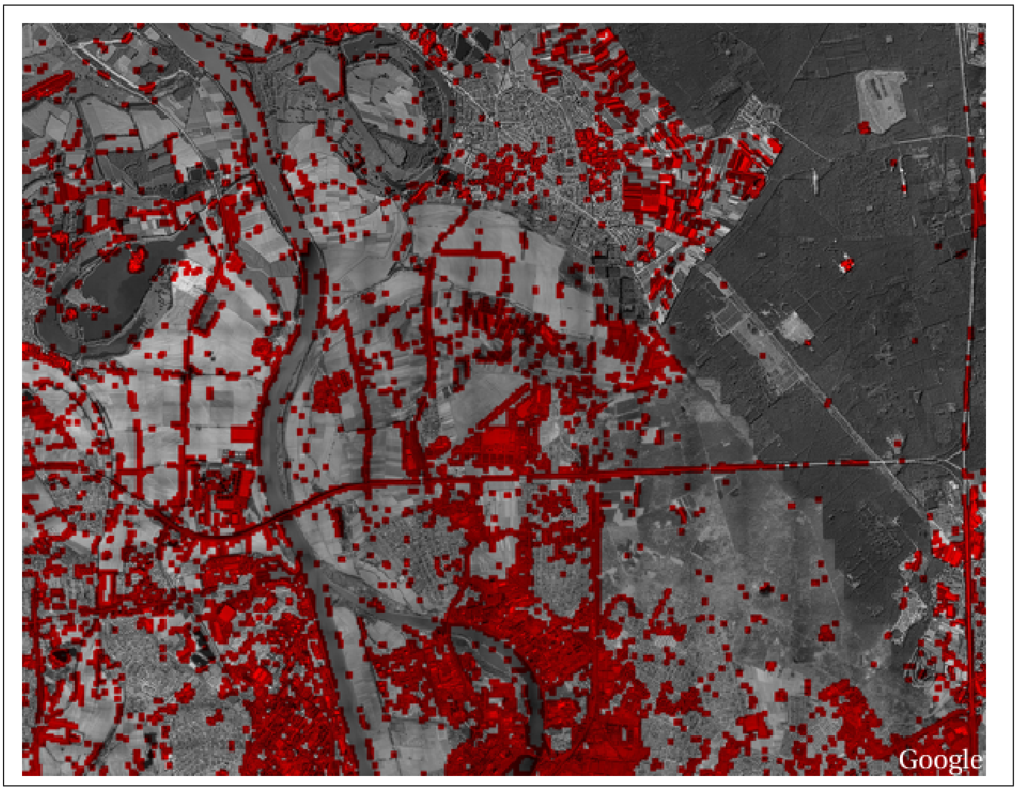
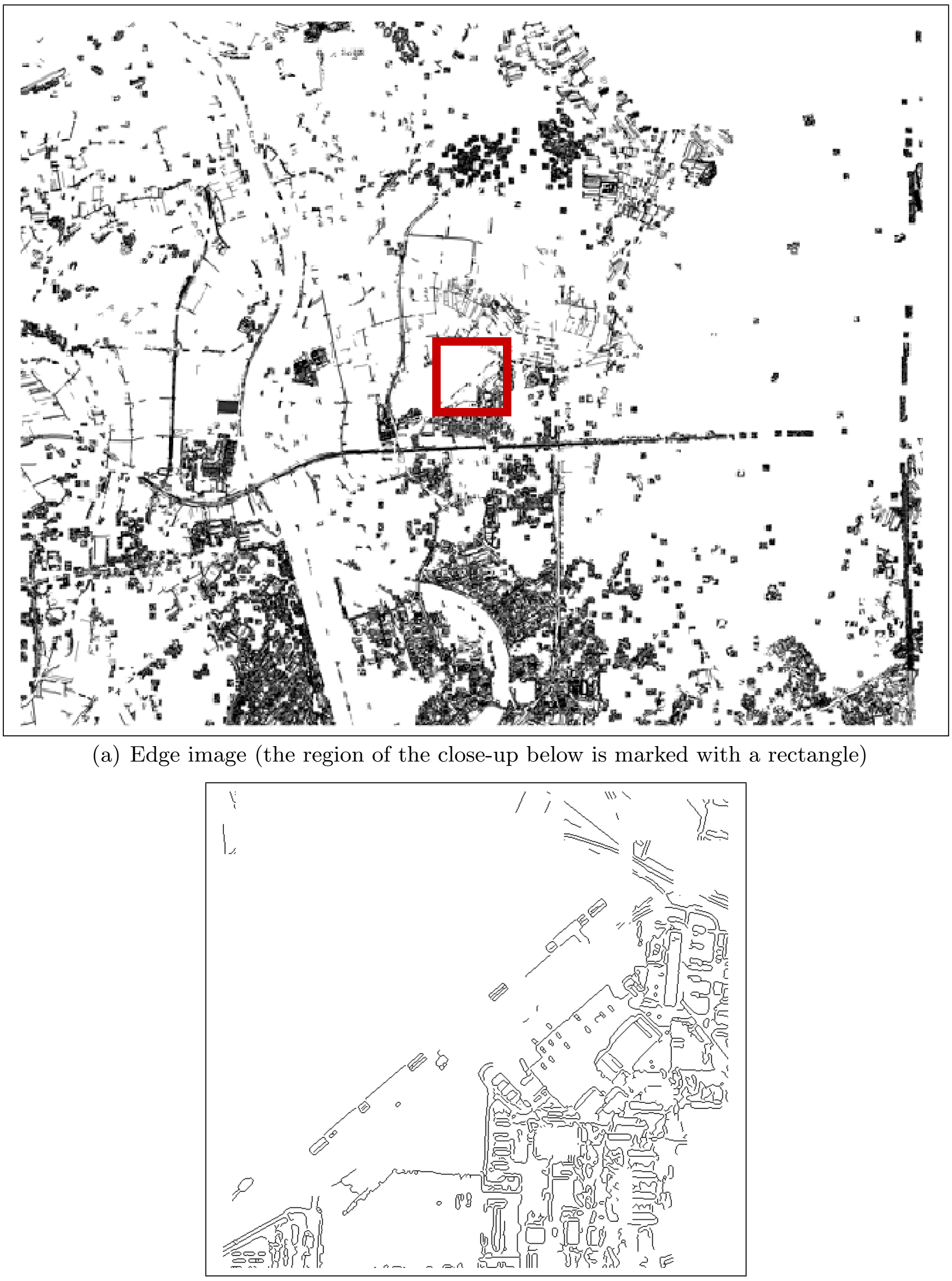


Figure A.21: ETOR – Expansion of regions of interest. *The runway is successfully included in the Region of Interest.*



(a) Edge image (the region of the close-up below is marked with a rectangle)

(b) Airport region close-up

Figure A.22: ETOR – Edge detection result. *In this example, Canny edge detector fails to generate edges around the runway.*

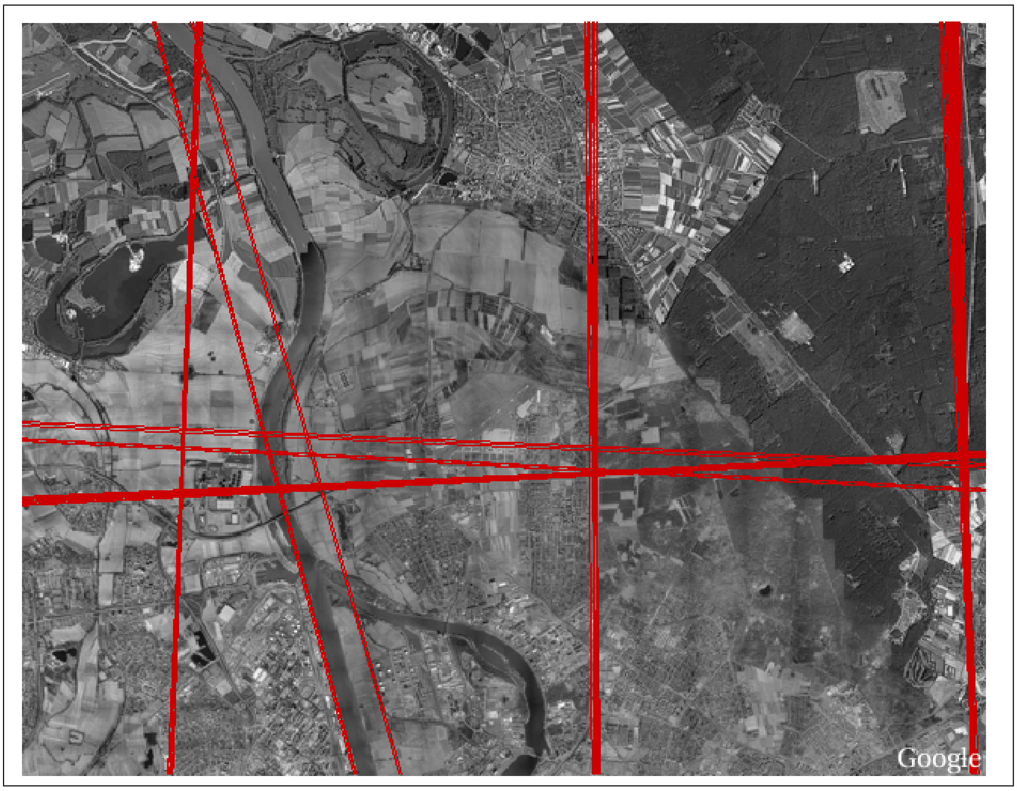


Figure A.23: ETOR – Detected line pairs (overlaid on original image). *Since line detection relies on the edge data, the runway line cannot be detected.*

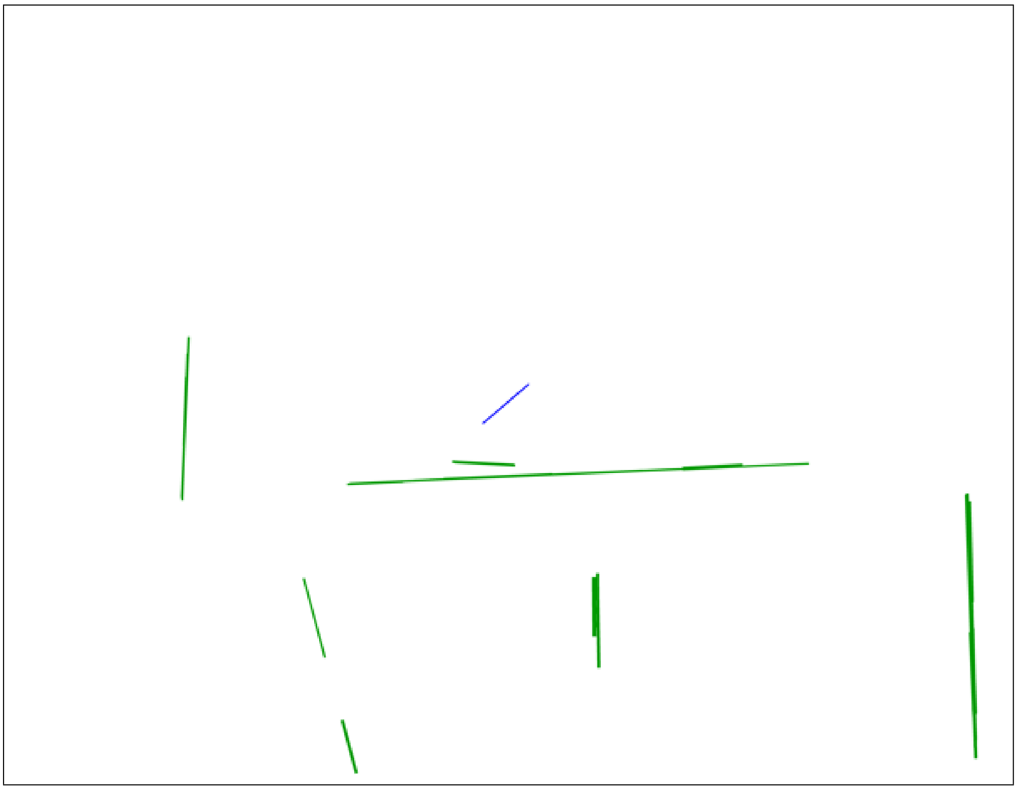


Figure A.24: ETOR – Final result (Red: True Positive, Green: False Positive, Blue: False Negative, White: True Negative)

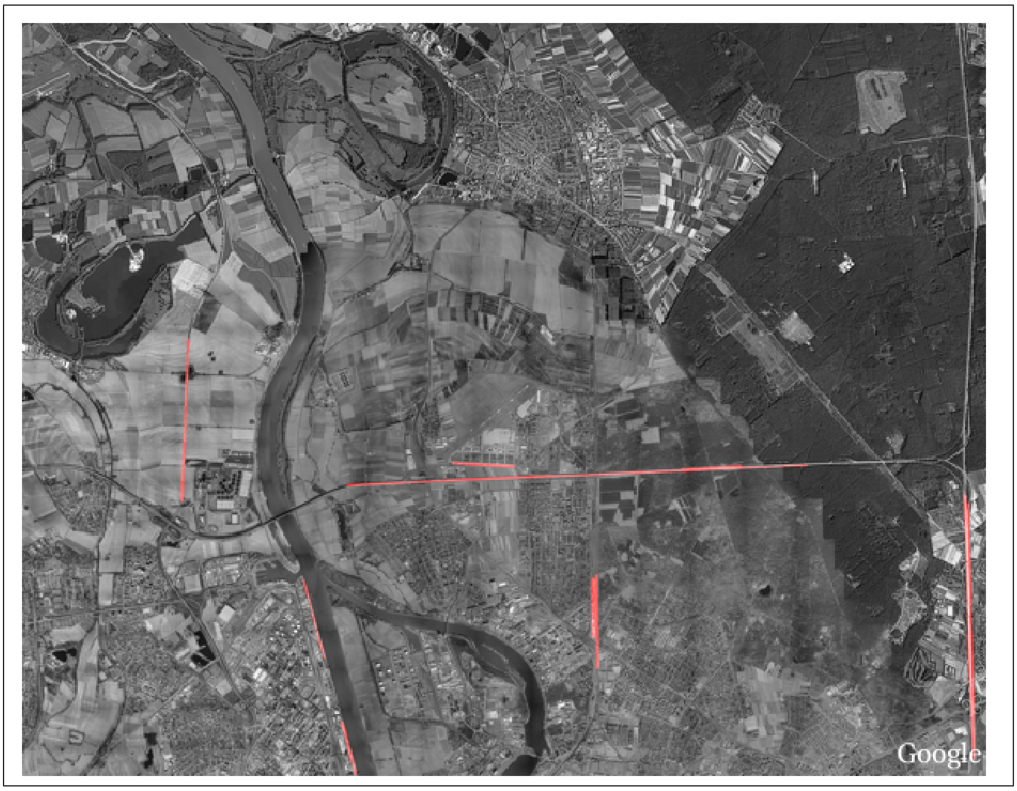


Figure A.25: ETOR – Final result (overlaid on original image). *Instead of the runway a series of long straight road segments are detected.*

APPENDIX B

DATA SET INFORMATION

Table B.1: Airport and image informations used in dataset

ICAO	Name	City	Country	IATA	Lat.	Long.	Width	Height
CYBG	Bagotville	Bagotville	Canada	YBG	48.3	-71.0	14400	11247
CYBR	Brandon Muni	Brandon	Canada	YBR	49.9	-100.0	14400	11247
CYXE	Saskatoon J G Diefenbaker International	Saskatoon	Canada	YXE	52.2	-106.7	14400	11247
CYYG	Charlottetown	Charlottetown	Canada	YYG	46.3	-63.1	14400	11247
DRRN	Diori Hamani	Niamey	Niger	NIM	13.5	2.2	14400	11247
DTTJ	Zarzis	Djerba	Tunisia	DJE	33.9	10.8	14400	11247
ETAR	Ramstein Ab	Ramstein	Germany	RMS	49.4	7.6	14400	11247

Continued on Next Page. . .

Table B.1 – Continued

ICAO	Name	City	Country	IATA	Lat.	Long.	Width	Height
ETNS	Schleswig	Schleswig	Germany	N/A	54.5	9.5	14400	11247
ETOR	Coleman Aaf	Coleman	Germany	N/A	49.6	8.5	14400	11247
ETSL	Lechfeld	Lechfeld	Germany	N/A	48.2	10.9	14400	11247
FABE	Bisho	Bisho	South Africa	VIY	-32.9	27.3	14400	11247
FAPE	Port Elizabeth	Port Elizabeth	South Africa	PLZ	-34.0	25.6	14400	11247
FAUP	Upington	Upington	South Africa	UTN	-28.4	21.3	14400	10896
FNLU	Luanda 4 De Fevereiro	Luanda	Angola	LAD	-8.9	13.2	14400	10704
GCLA	La Palma	Santa Cruz De La Palma	Spain	SPC	28.6	-17.8	9168	10704
GGOV	Bissau Oswaldo Vieira International	Bissau	Guinea Bissau	BXO	11.9	-15.7	14400	10704
GMMC	Anfa	Casablanca	Morocco	CAS	33.6	-7.7	14400	10704
GMME	Sale	Rabat	Morocco	RBA	34.1	-6.8	14400	10869
HEAT	Asyut International	Asyut	Egypt	N/A	27.0	31.0	14400	10704
KCXO	Montgomery Co	Conroe	USA	CXO	30.4	-95.4	14400	10704
KFLV	Sherman Aaf	Fort Leavenworth	USA	FLV	39.4	-94.9	14400	10704
KFMY	Page Fld	Fort Myers	USA	FMY	26.6	-81.9	14400	10704
KFOK	The Francis S Gabreski	West Hampton Beach	USA	FOK	40.8	-72.6	14400	10704
KGEG	Spokane International	Spokane	USA	GEG	47.6	-117.5	14400	10704
KHUF	Terre Haute International Hulman Fld	Terre Haute	USA	HUF	39.5	-87.3	14400	10875

Continued on Next Page. . .

Table B.1 – Continued

ICAO	Name	City	Country	IATA	Lat.	Long.	Width	Height
KILM	Wilmington International	Wilmington	USA	ILM	34.3	-77.9	14400	10875
KLNK	Lincoln Muni	Lincoln	USA	LNK	40.9	-96.8	14400	10875
KMCF	Macdill Afb	Tampa	USA	MCF	27.8	-82.5	6636	8945
KMCI	Kansas City International	Kansas City	USA	MCI	39.3	-94.7	14400	10875
KMKE	General Mitchell International	Milwaukee	USA	MKE	42.9	-87.9	14400	10875
KNMM	Meridian Nas	Meridian	USA	NMM	32.6	-88.6	14400	10875
KNTD	Point Mugu Nas	Point Mugu	USA	NTD	34.1	-119.1	14400	10875
KSFF	Felts Fld	Spokane	USA	SFF	47.7	-117.3	14400	10875
KSHV	Shreveport Rgnl	Shreveport	USA	SHV	32.4	-93.8	14400	10896
KSZL	Whiteman Afb	Knobnoster	USA	SZL	38.7	-93.5	14400	10875
KTEB	Teterboro	Teterboro	USA	TEB	40.8	-74.1	14400	10875
LDSP	Split	Split	Croatia	SPU	43.5	16.3	6165	6781
LELN	Leon	Leon	Spain	N/A	42.6	-5.7	14400	11211
LFBF	Francazal	Toulouse	France	N/A	43.5	1.4	14400	10896
LFOH	Octeville	Le Havre	France	LEH	49.5	0.1	13584	8813
LGKR	Ioannis Kapodistrias International	Kerkyra/Corfu	Greece	CFU	39.6	19.9	8712	10896
LGTL	Kasteli	Kasteli	Greece	N/A	35.2	25.3	12480	10896
LICZ	Sigonella	Sigonella	Italy	NSY	37.4	14.9	14400	10896

Continued on Next Page…

Table B.1 – Continued

ICAO	**Name**	**City**	**Country**	**IATA**	**Lat.**	**Long.**	**Width**	**Height**
LPAR	Alverca	Alverca	Acores	N/A	38.9	-9.0	14400	10896
LPVR	Vila Real	Vila Real	Acores	VRL	41.3	-7.7	14400	10896
LSZH	Zurich	Zurich	Switzerland	ZRH	47.5	8.5	14400	10896
LTAG	Incirlik Ab	Adana	Turkey	ADA	37.0	35.4	14400	10896
LTBG	Bandirma	Bandirma	Turkey	BDM	40.3	28.0	14400	10875
LZIB	M R Stefanik	Bratislava	Slovakia	BTS	48.2	17.2	14400	10875
LZKZ	Kosice	Kosice	Slovakia	KSC	48.7	21.2	14400	10896
MMCN	Ciudad Obregon International	Ciudad Obregon	Mexico	CEN	27.4	-109.8	14400	10896
MMNL	Quetzalcoatl International	Nuevo Laredo	Mexico	NLD	27.4	-99.6	14144	8999
MMPR	Licenciado Gustavo Diaz Ordaz International	Puerto Vallarta	Mexico	PVR	20.7	-105.3	14400	13698
MUNG	Rafael Perez	Nueva Gerona	Cuba	GER	21.8	-82.8	14400	10875
OERK	King Khaled International	Riyadh	Saudi Arabia	RUH	25.0	46.7	14400	10875
OISS	Shiraz Shahid Dastghaib International	Shiraz	Iran	SYZ	29.5	52.6	14400	10875
OKBK	Kuwait International	Kuwait	Kuwait	KWI	29.2	48.0	14400	10875
PAKN	King Salmon	King Salmon	USA	AKN	58.7	-156.6	14400	10875
PASI	Sitka Rocky Gutierrez	Sitka	USA	SIT	57.0	-135.4	14400	7806
RCQC	Makung	Makung	Taiwan	MZG	23.6	119.6	14088	10896
SANL	La Rioja	La Rioja	Argentina	IRJ	-29.4	-66.8	14400	10896

Continued on Next Page…

Table B.1 – Continued

ICAO	Name	City	Country	IATA	Lat.	Long.	Width	Height
SANU	San Juan	San Julian	Argentina	UAQ	-31.6	-68.4	13260	10896
SBPN	Porto Nacional	Porto Nacional	Brazil	PNB	-10.7	-48.4	14400	10875
SBPP	Ponta Pora	Ponta Pora	Brazil	PMG	-22.5	-55.7	8964	11247
SBUR	Uberaba	Uberaba	Brazil	UBA	-19.8	-48.0	14400	11247
SEGU	Simon Bolivar International	Guayaquil	Ecuador	GYE	-2.2	-79.9	14400	11247
SEMT	Eloy Alfaro International	Manta	Ecuador	MEC	-0.9	-80.7	10512	11247
SPQT	Coronel Fap Francisco Secada Vignetta	Iquitos	Peru	IQT	-3.8	-73.3	14400	11247
SVBC	General Jose Antonio Anzoategui International	Barcelona	Venezuela	BLA	10.1	-64.7	14400	11247
SVMG	Del Caribe International Gen Santago Marino	Porlamar	Venezuela	PMV	10.9	-64.0	14400	11247
SVSO	Mayor Buenaventura Vivas	Santo Domingo	Venezuela	STD	7.6	-72.0	14400	11247
TJBQ	Rafael Hernandez	Aguadilla	Puerto Rico	BQN	18.5	-67.1	10056	11499
VABB	Chhatrapati Shivaji International	Bombay	India	BOM	19.1	72.9	14400	11247
VGZR	Zia International	Dhaka	Bangladesh	DAC	23.8	90.4	4812	11247
VTPI	Takhli	Nakhon Sawan	Thailand	N/A	15.3	100.3	14400	11247
WABP	Moses Kilangin	Timika	Indonesia	TIM	-4.5	136.9	8628	11247
YBTL	Townsville	Townsville	Australia	TSV	-19.3	146.8	14400	11247
YSSY	Kingsford Smith International Airport	Sydney	Australia	SYD	-33.9	151.2	14400	11247