BUILDING DETECTION FROM SATELLITE IMAGES USING SHADOW AND COLOR INFORMATION

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

BY

HASAN VOLKAN GÜDÜCÜ

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

AUGUST 2008

Approval of the thesis:

BUILDING DETECTION FROM SATELLITE IMAGES USING SHADOW AND COLOR IFORMATION

submitted by **HASAN VOLKAN GÜDÜCÜ** 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 Dept., METU**

Prof. Dr. Uğur HALICI ____________________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Kemal LEBLEBİCİOĞLU ____________________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Uğur HALICI ____________________
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Gözde BOZDAĞI AKAR ____________________
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. H. Şebnem DÜZGÜN ____________________
Mining Engineering Dept., METU

Asst. Prof. Dr. İlkay ULUSOY ____________________
Electrical and Electronics Engineering Dept., METU

Date: 26.08.2008

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 : Hasan Volkan Güdücü

Signature :

ABSTRACT

BUILDING DETECTION FROM SATELLITE IMAGES USING SHADOW AND COLOR INFORMATION

Güdücü, Hasan Volkan
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Uğur Halıcı

August 2008, 121 pages

A method for detecting buildings from satellite/aerial images is proposed in this study. The aim is to extract rectilinear buildings by using hypothesize first verify next manner. Hypothesis generation is accomplished by using edge detection and line generation stages. Hypothesis verification is carried out by using information obtained both from the color segmentation of HSV representation of the image and the shadow detection stages' output. Satellite/aerial image is firstly filtered to sharpen the edges. Then, edges are extracted using Canny edge detection algorithm. These edges are the input for the Hough Transform stage which will produce line segments according to these extracted edges. Then, extracted line segments are used to generate building hypotheses. Verification of these hypotheses makes use of the outputs of the HSV color segmentation and shadow detection stages. In this study, color segmentation is processed on the HSV representation of the satellite/aerial image which is less sensitive to illumination. In order to perform the shadow detection, the basic information which is shadow areas have higher value of saturation component and lower value of value component in HSV color space is used and according to this information a mask is applied to the HSV representation of the image to produce shadow pixels.

The proposed method is implemented as software written in MATLAB programming software. The approach was tested in several different areas. The results are encouraging

Keywords: Building Detection, Canny Edge Detection, Color Based Segmentation, Shadow Detection, Hough Transform.

ÖZ

UYDU GÖRÜNTÜLERİNDEN GÖLGE VE RENK BİLGİSİ KULLANILARAK BİNA BULUNMASI

Güdücü, Hasan Volkan
Y. Lisans, Elektrik-Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Uğur Halıcı

Ağustos 2008, 121 sayfa

Bu çalışmada hava/uydu görüntülerinden binaların bulunmasıyla ilgili bir yöntem önerilmiştir. Amaç, köşeli yapıdaki binaları önce hipotez geliştirip sonrasında bunu doğrulamak prensibine uyarak belirlemektir. Hipotez geliştirme; kenar bulma ve doğru oluşturma aşamalarından meydana gelmektedir. Hipotez doğrulama ise, resmin HSV uzayındaki rengine göre bölümleme ve gölge bulma aşamalarından elde edilen bilgi ile yapılmaktadır. Hava/uydu görüntüsü ilk olarak görüntüdeki kenarların daha belirginleşmesi için bir filtreden geçirilir. Sonrasında, kenarlar Canny kenar bulma algoritması ile çıkartılır. Bu kenarlar Hough Transform bölümünde kullanılarak doğru parçaları oluşturulur. Sonrasında, bu doğrular kullanılarak bina hipotezleri çıkartılır. Bu hipotezlerin doğrulanması için resmin HSV uzayındaki rengine göre bölümleme ve gölge bulma aşamalarının sonuçları kullanılmaktadır. Bu çalışmada renge göre bölümleme aşaması RGB uzayında bulunan görüntünün, ışığa RGB'ye göre daha az hassasiyet gösteren HSV uzayındaki eşdeğerine göre yapılmaktadır. Gölge bulma aşaması, HSV uzayında gölge olan yerlerin olmayan yerlere göre doygunluk parçasının daha yüksek ve değerlik parçasının daha düşük olduğu temel bilgisinin ışığında görüntüye basit bir maske uygulanarak yapılmaktadır.

Önerilen yöntem yazılım olarak MATLAB yazılımında yazılmıştır. Yöntem farklı görüntüler ile denenmiştir. Sonuçlar teşvik edicidir.

Anahtar Kelimeler: Bina Bulma, Canny Kenar Bulma, Renge Göre Bölümleme, Gölge Bulma, Hough Transform.

To My Family

ACKNOWLEDGMENTS

I express my deepest gratitude to Prof. Dr. Uğur HALICI for her guidance and support throughout this study and Asst. Prof. Dr. İlkay ULUSOY, Örsan Aytekin, METU Vision, and Computer Vision and Intelligent Systems Research laboratory.

I also thank to Assoc. Prof. Dr. H. Şebnem DÜZGÜN and Geodetic and Geographic Information Technologies research laboratory.

I could never achieve this goal without support of my wife and my family. I thank them for their encouragement and full support.

I would like to thank all my friends who helped in producing this thesis for their valuable suggestions and comments.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Object detection in satellite/aerial images has been an important research topic in computer vision for many years. Some useful applications of this subject are; updating of geographic information system (GIS) databases, urban city planning and land use analysis. The fundamental challenges that drive much of the research in this field are the edge or line extraction problems and segmentation problem, which is finding a desired object and separating it from the background in the presence of distractions caused by other features such as surface markings, vegetation, shadows, and highlights.

A large variety of building detection techniques and algorithms have been reported in the literature, but most detection algorithms rely on edge-based techniques that consist of linear feature detection, grouping for parallelogram structure extraction, and building polygons verification using knowledge such as geometric structure, shadow, and so forth. In order to solve this complex problem, integrating the power of multiple algorithms, cues, and available data sources is also implemented recently to improve the reliability and robustness of the extraction results. The recent availability of commercial high-resolution satellite imaging sensors such as IKONOS provides a new data source for building detection. The high spatial resolution of the imagery specifies very fine details in urban areas and facilitates the classification and detection of urban-related features such as roads and buildings. Since manual extraction of buildings from imagery is very slow, automated methods have been proposed to improve the speed and utility for urban map's production. Most of the recent work on building extraction from high resolution satellite images is based on supervised techniques. These techniques either require a classification based on initial training data to provide hypotheses for the positions and sizes of the candidate building features, or they use training sets or model databases to classify or match the buildings.

The difficult problem in building detection is performing automatic and accurate matching of the buildings in satellite/aerial imagery. The motivation behind this work is to use shadow and HSV (hue, saturation, intensity) color information gathered from the satellite/aerial image to verify the building hypotheses generated by using edge-based techniques. The approach has to be robust and accurate to be used in a system.

1.2 OBJECTIVE AND GOALS

The objective of this thesis is to develop a robust, reliable building detection algorithm, which is capable of detecting rectilinear buildings in complex environments and dense urban areas.

1.3 METHODOLOGY USED

The building detection approach used in this thesis is a model based method, the purpose of which is detecting rectilinear buildings. The buildings on the satellite/aerial images are 2D structures which are mostly made up of regions having borders being straight line. And these straight lines are the combinations of pixels in an image. So, the building detection algorithm firstly extracts the pixels that compose line segments by means of edge detection process. Then, these extracted pixels are grouped to form straight lines which will be searched for whether they form a rectilinear building or not. Then, verification of these possible buildings is done with the help of shadow and HSV color information extracted from the image.

Edge detection process is accomplished using Canny edge detection algorithm. Then, extracted edges are used to generate line segments through Hough Transform algorithm. After lines are generated, rectilinear building hypothesis generation is performed based on building models which are U-shaped, L-shaped and I-shaped. And finally, generated hypotheses are verified using the shadow image and HSV color segmentation image information .The color HSV segmented image is produced using k-means clustering algorithm and shadow image is produced by applying a mask to the image represented in the HSV space.

1.4 DATA

Nadir or near nadir view high resolution aerial images and satellite images are used in this thesis. The test images used in the thesis belong to the urban areas taken from Google Earth with an altitude of 1000m and the satellite images are taken from IKONOS satellite with 1m-pixel resolution.

1.5 CONTRIBUTION

The building detection system developed in this thesis work mainly contributes to reduce the false positives detection of buildings by using the shadow and the HSV color segmented image information extracted from the satellite/aerial imagery. The method proposed here is based on hypothesize and verify approach to detect rectilinear buildings.

1.6 THESIS OVERVIEW

The thesis chapters are organized as follows; Chapter 2 is devoted to background information on the buildings detection subject and the algorithms used throughout this study.

In Chapter 3, the method used in this study is explained in detail.

Chapter 4 covers the experimental results obtained from test cases.

Chapter 5 gives the summary, conclusion, and possible future work after this study.

CHAPTER 2

BACKGROUND

2.1 INTRODUCTION

In this chapter, the previous studies in the field of building detection are described briefly and the concepts used in this thesis are explained.

Building detection has attracted many researchers' attention in computer vision for many years. Detection of buildings automatically from satellite/aerial imagery involves several different problems related to computer vision since in urban areas there are many other subjects in close proximity such as trees, power lines vehicles, and parking lots. And these subjects may occlude the buildings' rooftops. Besides, the rooftops of the buildings may be composed of different surface materials with differing reflectance properties. The above mentioned problems make the automatic building detection is a challenging problem in the computer vision area.

In [2] researchers used snake-based approach to extract 2D building outlines from high resolution IKONOS satellite images and height data captured by airborne laser scanning system. A semiautomated approach is used in [36] based on active contour model (snakes) and the dynamic programming optimization technique. The method requires a digital surface model and an ortho-image. This approach can be more effective if applied after a human operator has manually determined seed points near the boundary of a desired feature.[37] extracts the principal contours of buildings in dense urban areas according to the radiometric behavior of buildings. The pros of this model to other snake-based approaches are; the new initiation criteria present a more reliable measure to select proper initial seeds, and sharply reduce the influence caused by spurs, the new external energy function consisted of corner feature and edge consistency makes the snake model more stable to converge to the true building contours and the post-treatment combining with illumination information reduces the constraints for initial snake model and the interference caused by illumination, sharply lessening the times of iteration. A different approach from the standard/traditional building extraction methods is using wavelet transform and image scaling [1] which uses both the high and the low frequency components of an aerial image. The wavelet transform extracts building edges corresponding to high frequencies while scaling provides abstraction by eliminating such frequency components. An attractive feature of the techniques is that they generate a family of images at different scales and or resolutions based on the original source image. There are a lot of systems that use widely available stereo images. Stereo images allow the determination of the third dimension by epipolar matching of different features extracted from both images. Multi-view strategies are advantageous in providing redundant information and improving the accuracy of the reconstruction. In [38] two or more panchromatic images which need not to be stereo pairs are used to extract flat or symmetric gable roof rectilinear buildings and constructs 3D models. This approach generates hypotheses for rectangular roof components by grouping lines in the image hierarchically and verifies hypotheses by searching for presence of predicted walls and shadows. This algorithm leaves the verification decision at the end, which is different from the general approach hypothesize and verify sequentially. Multiple and overlapping images of the scene are used in [39] to detect complex buildings with flat or complex rooftops. Perceptual grouping techniques are presented to group low-level features, such as edges and corners, into higher-level features by using a hierarchy of grouping processes with multiple levels of detail and use of probabilistic reasoning methods to select among the multiple hypotheses. Modeling the complex buildings is a major problem in this research. [40] uses both multi-view stereo and color information as cues for building detection in dense urban scenes. The significant improvement in this approach is detection rates is achieved by running the detection procedure on different views and combining the results.[41] focuses the problem of automatic extraction and modeling from stereoscopic pairs of high resolution (less than 10cm per ground pixel) aerial images. It is based on Digital Elevation Model (DEM) computation which involves a gradient correlation, contour-adaptive windows with a geodesic weighting, a multi-resolution coarse to fine scheme, and a two-way filtering symmetrical validation. [26] aims to detect buildings from a complete disparity map obtained by various stereo correlation techniques using 8-bit gray scale images. Approach used in [42] is based on fuzzy segmentation using aerial image and the GIS data (1:10000) as prior knowledge about the building. 2D GIS databases and domain knowledge is used with pairs of stereo images in [43] to reduce the complexity of the reconstruction by focusing on one building. [44] developed an approach to update the buildings existing vector database using normalized Digital Surface Model (nDSM) and spectral reflectance values of images. In this research since there is no perceptual grouping is done on vectorized edges, generated straight lines are ungrouped and could form wrong type of building model. In [45] three data sources; LiDAR, an aerial image and ground images is used to extract 3D view of the buildings. Using a hybrid model enables the researchers collect more accurate edge information and building details compared to using only LiDAR and more detailed surface information is gathered compared to using only stereo aerial images. This method is costly and needs ground images which are not available for the ones who just have only satellite or aerial images. Approach used in [46] for Level of Detail building model reconstruction is to use airborne LiDAR data and optical imagery. [3] uses probabilistic theory at its simplest level, assuming that the buildings probability distribution can be represented by a logistic function. In this approach buildings are regarded as individual objects in the image. And key points are to propose original features that characterize buildings and to achieve recognition by computing a probability measure on each individual object. This system has limitations on shadow model which is image dependent. [4] uses 3 layer perception neural network algorithm to improve the detection percentage of the building extraction from high-resolution satellite imagery, but the detection rate is approximately at %80 and needs human interaction during the learning phase of neural networks. [47] uses structural, contextual, and spectral information to automatically extract buildings from high-resolution satellite imagery. This approach's building extraction is benefited from three different extraction strategies which are; building hypotheses are generated and verified through shape analysis differential morphological profile (DMP), shadows are modeled as a part of the scene and provide reliable contextual information to independently hypothesize the position and the size of adjacent buildings and small bright buildings that cannot be reliably detected by structural and contextual information are detected using spectral information. Main drawback of this approach is gabled roof buildings are not extracted and successful detection rate is about %70.[48] uses a method to automatically detect houses and street networks in IKONOS multispectral images. This approach uses k-means clustering algorithm to extract possible houses and street networks by combining both spatial and spectral features. The novelty in this algorithm is its introduction of spatial coherence to clustering via connected components analysis. [49] uses a single Digital Elevation Model (DEM) as an input for automatic building modeling. Firstly, DEM is segmented into locally planar surfaces to recover the various facets of the buildings from the raw DEM. Then, in the second stage vectorization of the boundaries of each surface patch is done to obtain the model of the buildings. The performance of this system is dependent on the DEM quality. [50] uses a single DEM derived from optical stereo processing or active sensors, registered to a corresponding optical view of an urban site for the recognition and three dimensional reconstruction of buildings. The primary goal of the algorithm is to index into a set of many possible parametric surfaces that represent different rooftop shapes. Given the reduced set of surfaces that most closely resemble the DEM, a surface-fitting algorithm then fits each surface to the elevation data in order to select a final surface model and parameters for reconstructing the building rooftop. [51] uses single aerial image for automatic building detection by utilizing a graph-based approach. In this research, perceptual grouping concept is applied emphasizing the formulation of a graph. Although this system works reasonably well without any verification process, an improvement is needed in shadow verification analysis. Also, further work is also needed for line extraction process. As building hypotheses are generated by combining lines, the quality of line process determines the overall performance of building detection. [8] uses a monocular aerial image with a general viewpoint to detect buildings and construct 3D shape descriptions. Basic approach in this research is to use the geometric and projective constraints to make hypotheses for the presence of building roofs from the low-level features and to verify this by using available 3-D cues. This method assumes that building shapes to be rectilinear with flat roofs. [52] deals with the automatic extraction of building outlines using a pair of optical and synthetic aperture radar (SAR) images. This method has two main parts; first, extraction of partial potential building footprints on the SAR image, and then shapes detection on the optical one using the previously extracted primitives (lines). Main drawbacks of this approach are the SAR image could be used to validate the buildings detected in the optical image, the primitive SAR detection could be improved and large building detection algorithm be developed.

Some researchers in the building detection area have tried to work from a single image only ([5], [6], [7], [8], [9]). When there is only single image used as the data source, some ambiguities still remain unsolved, so these approaches' performance is limited. In order to improve the success rate of the algorithms', researchers benefited from shadows ([10], [11]) as the supplementary data source which is used to verify the buildings in automatic building detection methods.

In order to reduce the complexity of the detection problem, the methods used to detect buildings make use of more than one data sources. Mostly used data sources are Geographic Information System (GIS) database, Digital Surface Models (DSM), Digital Elevation Model (DEM), high resolution satellite/aerial image, stereo images, Laser induced detection and ranging (LiDAR), ground view images, spectral information of images, Synthetic Aperture Radar (SAR).

Multiple images have also been used in previous works ([12], [13], [14], [15], [16], [17], [18]). Most of these systems assume that the images are from nadir views and taken at nearly the same time, which simplifies the task of matching features.

There has also been work on using data with other data sources, such as color [19], digital surface models derived from intensity images or range sensors directly ([20], [21], [22], [23], [24], [25], [26]), or multispectral data, which can significantly simplify the task of building detection ([27], [28]).

The aim of this chapter is to present the algorithms and give explanations to the methods used throughout the thesis, which form the basic steps of the current study.

2.2 COLOR MODELS

Color has been widely used in machine-based vision systems for tasks such as image segmentation and object recognition. It offers several significant advantages over geometric cues and gray scale intensity such as computational simplicity, robustness under partial occlusion, rotation in depth, scale changes and resolution changes. Although color based object segmentation methods [29] proved to be efficient in a variety of vision applications, there are several problems associated with these methods, of which color constancy is one of the most important. A few factors that contribute to this problem include illumination changes, shadows and highlights, inter-reflection with other objects, and camera characteristics. The problem therefore is how to represent the object color robustly and efficiently.

Digital color images consist of color pixels, each of which is associated with a color feature vector. Color pixels usually contain Red, Green and Blue values each measured in 8 bits. A color model is a method for explaining the properties or behavior of color within some particular context [30]. The purpose of a color model is to allow convenient specification of colors within some color gamut, where the color gamut is a subset of all visible chromaticity [31].

2.2.1 RGB COLOR MODEL

The RGB (Red, Green, Blue) color model uses a Cartesian coordinate system and forms a unit cube shown in **Figure 2.1**

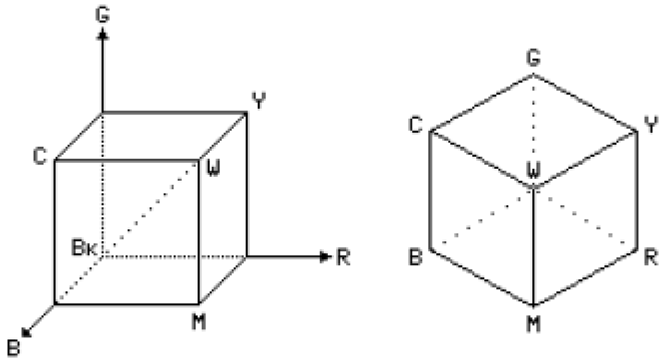

Figure 2.1 RGB Color Cube

The dotted main diagonal of the cube, with equal amounts of Red, Green and Blue, represents the gray levels. This diagonal is further referred to as the gray diagonal. The RGB color model is hardware oriented and is used in many image capturing, processing and rendering devices.

2.2.2 HSV COLOR MODEL

The HSV (Hue, Saturation, Value) color model suggested by A. Smith [32] is user oriented and is based on the intuitive appeal of the artist's tint, shade, and tone. The subspace within which the model is defined is a hexcone as shown in **Figure 2.2**.

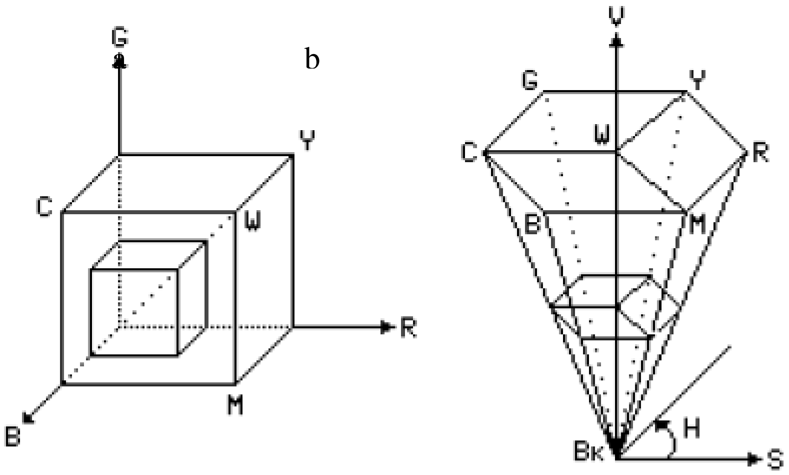

Figure 2.2 HSV Color Systems

Hue Saturation Value (HSV) color system corresponds to projecting standard Red, Green, Blue (RGB) color space along its principle diagonal from white to black. This results in the hex cone in **Figure 2.2**b. Descending the V axis in **Figure 2.2**b gives smaller hexcones corresponding to smaller (darker) RGB subcubes in **Figure 2.2**.

The top plane of the hexcone corresponds to V=1 and contains the maximum intensity colors. The point at the apex is black and has the coordinate V=0. Hue is the angle around the vertical (gray) axis, with Red at 0. The notion of Hue is very important and is used in many other color models. The complementary colors are 180° opposite to one another as measured by Hue. The value of S is a ratio, ranging from 0 on the centerline (V axis) to 1 on the triangular sides of the hexcone. The point S=0, V=1 is white. When S=0, H is irrelevant and is undefined. There is a geometrical correspondence between the HSV and the RGB color models. The top of the HSV hexcone corresponds to the surface seen by looking along the gray diagonal of the RGB color cube from white toward black as shown in **Figure 2.2**. The RGB cube has subcubes as shown in **Figure 2.2**. Each subcube when viewed along its gray diagonal appears like a hexcone. Each plane of constant V in HSV space corresponds to such a view of a subcube in the RGB space. In other words, the top surface of the RGB subcube orthogonal projected along the gray diagonal onto a plane becomes a plane of a constant V in the HSV hexcone.

2.3 COLOR IMAGE SEGMENTATION

One way of interpreting an image is to group and assign pixels to a region due to some criterion. This partitioning of the image into different regions is called segmentation [33]. Formally, segmentation can be defined as a method to partition an image into subimages called regions of common characteristics such that each segment is a discrete image.

Cluster analysis, also called segmentation analysis or taxonomy analysis, is a way to create groups of objects, or clusters, in such a way that the profiles of objects in the same cluster are very similar and the profiles of objects in different clusters are quite distinct.

2.3.1 CLUSTER ANALYSIS

A description of what constitutes a cluster, which probably agrees closely with our intuitive understanding of the term [34] is given by considering entities as points in a p-dimensional space, with each of the p variables being represented by one of the axis of this space. The variable values for each entity define a p-dimensional coordinate in this space. Clusters may now be described as continuous regions of this space containing a relatively high density of points, separated from other such regions by regions containing a relatively low density of points. This description matches the way we detect clusters visually in two or three-dimensional space.

The main purpose of clustering data is to reduce the size and complexity of the data set and to identify classes of similar entities. Data reduction may be accomplished by replacing the coordinates of each point in a cluster with the coordinates of that cluster's reference point, or assigning a point to a particular cluster. Clustered data require considerably less storage space and can be manipulated more quickly than the original data. The value of a particular clustering method depends on the application and may, for example, reflect how closely the reference points represent the data, or, how closely clusters represent specific shapes or volumes. An important factor also is the speed of the clustering algorithm.

Clustering techniques can be classified into two classes (**Table 2.1**) which are hierarchical and partitioning algorithms. The hierarchical algorithms can be divided into agglomerative (“bottom-up”) and splitting (“top-down”) procedures. The first type of hierarchical clustering starts from the finest partition possible (each observation forms a cluster) and groups them. The second type starts with the coarsest partition possible; one cluster contains all of the observations. It proceeds by splitting the single cluster up into smaller sized clusters. The partioning algorithms start from a given group definition and proceed by exchanging elements between groups until a certain score is optimized. The main difference between the two clustering techniques is that in hierarchical clustering once groups are found and elements are assigned to the groups, this assignment cannot be changed. In partitioning techniques, on the other hand, the assignment of objects into groups may change during the algorithm application.

Table 2.1 Clustering Techniques

CLASS NAME	METHOD
Hierarchical Techniques	The clusters themselves are classified into groups, the process being repeated at different levels to form a tree.
Partitioning Techniques	The clusters are formed by optimization of a "clustering criterion". The clusters are mutually exclusive, thus forming a partition of the set of entities.

Hierarchical clustering is appropriate for smaller samples (typically < 250). When sample number is large, the algorithm will be very slow to reach a solution and, indeed, may cause the computer to hang. To accomplish hierarchical clustering, the researcher must specify how similarity or distance is defined and how clusters are aggregated (or divided). Hierarchical clustering generates all possible clusters of sizes 1...K, but is used only for relatively small samples. In hierarchical clustering, the clusters are nested rather than being mutually exclusive, as is the usual case. That is, in hierarchical clustering, larger clusters created at later stages may contain smaller clusters created at earlier stages of agglomeration. One may wish to use the hierarchical cluster procedure on a sample of cases (ex., 200) to inspect results for different numbers of clusters. The optimum number of clusters depends on the research purpose. Identifying "typical" types may call for few clusters and identifying "exceptional" types may call for many clusters. After using hierarchical clustering to determine the desired number of clusters, the researcher may wish then to analyze the entire dataset with k-means clustering (aka, the Quick Cluster procedure: Analyze, Cluster, K-Means Cluster Analysis), specifying that number of clusters.

Partitional clustering is simple and fast so that it can be applied to large data sets. Main disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments. It minimizes intra-cluster variance, but does not ensure that the result has a global minimum of variance

K-means algorithm which belongs to partitional clustering is used in this thesis due to the speed concern which becomes an important parameter for large images.

2.3.2 K-MEANS ALGORITHM

Given S a set of N points and K the number of clusters, the algorithm chooses K reference points (e.g., at random) from S. Each reference point R_i defines a cluster C_i.

Then data points are partitioned into K clusters. Point p of S becomes a member of cluster C_i if p is closer in the underlying metric (e.g., the Euclidian distance) to R_i the reference point of C_i than to any other reference point. Closest means min $d(R_i,p)$, where $d(R_i,p)$ is a distance between point R_i of R and point p of S in the underlying metric. The centroid for each cluster is calculated and the centroid becomes a reference point of its cluster. During successive iterations, the centroids of each cluster are recalculated. During the iterations, the algorithm goes through all data points and determines if for point p in cluster C_i, the centroid of C_i is the nearest reference point. If so, no adjustments are made and the algorithm proceeds to the next data point. However, if the centroid of cluster C_j becomes the reference point closest to the data point p, then p is reassigned to cluster C_j, the centroids of the loosing cluster C_i (minus point p) and the gaining cluster C_j (plus point p) are recomputed, and the reference points of clusters C_i and C_j are moved to their new centroids. After each iteration, every one of the K reference points is a centroid, or mean, hence the name "k-means". The iterations proceed until, for all data points, no re-allocation of points from cluster to cluster is possible.

Finally, the distribution of points will correspond to the centroidal Voronoi configuration, where each data point is closer to the reference point of its cluster than to any other reference point, and each reference point is the centroid of its cluster. The algorithm runs in $O(N^* K)$ time.

The "k-means" algorithm does not guarantee the best possible partitioning, or finding the global minimum in terms of the error measure, but only provides a local minimum. However, the improvement of the partitioning and the convergence of the error measure to a local minimum are often quite fast, even when the initial reference points are badly chosen.

To summarize shortly; hierarchical techniques either aggregate or divide the data based on some proximity measure. The disadvantage is since they contain no provision for reallocation of entities, which may have, been poorly classified at an early stage in the analysis. The hierarchical methods tend to be computationally expensive and the definition of a meaningful stopping criterion for the fusion (or division) of the data is not straightforward.

Partitioning techniques, which seek to optimize some criterion, have a problem of finding a sub-optimal solution instead of a global solution. This is known as a local optima problem. Most optimization techniques also presume that the number of clusters is either known or given prior to clustering.

2.4 EDGE DETECTION

Edge detection is the process of finding sharp contrasts in intensities in an image. This process significantly reduces the amount of data in the image, while preserving the most important structural features of that image. An edge detection algorithm extracts the main properties from an image which are the building edges for the current study. **Figure 2.3** illustrates the edge detection algorithm applied to an image.

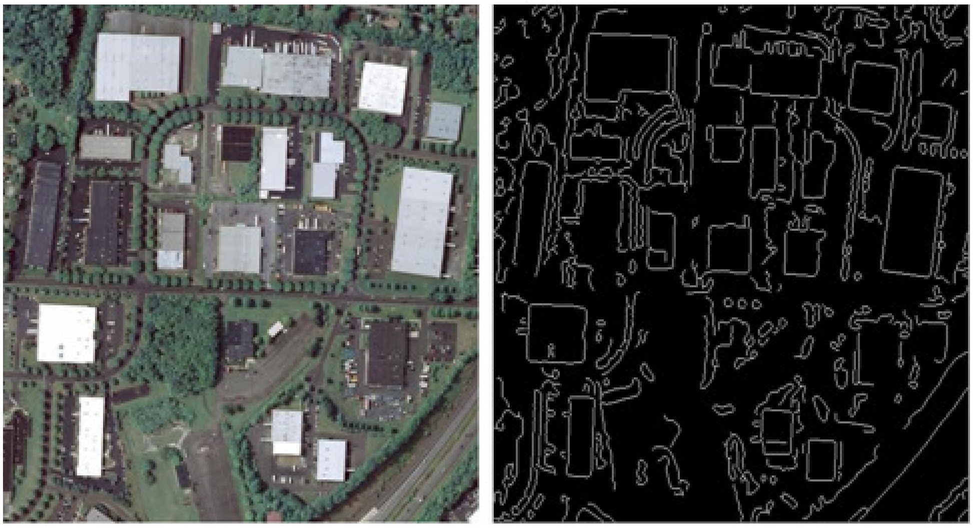

Figure 2.3 Edge Detection for an Aerial Image

There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories:

Gradient-Based Edge Detection: The gradient-based method detects the edges by searching for the maximum and minimum in the first derivative of the image.

Laplacian: The Laplacian method searches for zero crossings in the second derivative of the image to find edges.

An edge has the one-dimensional shape of a ramp and calculating the derivative of the image can highlight its location. To illustrate this idea suppose that there is a signal, with an edge shown by the jump in intensity (**Figure 2.4**).

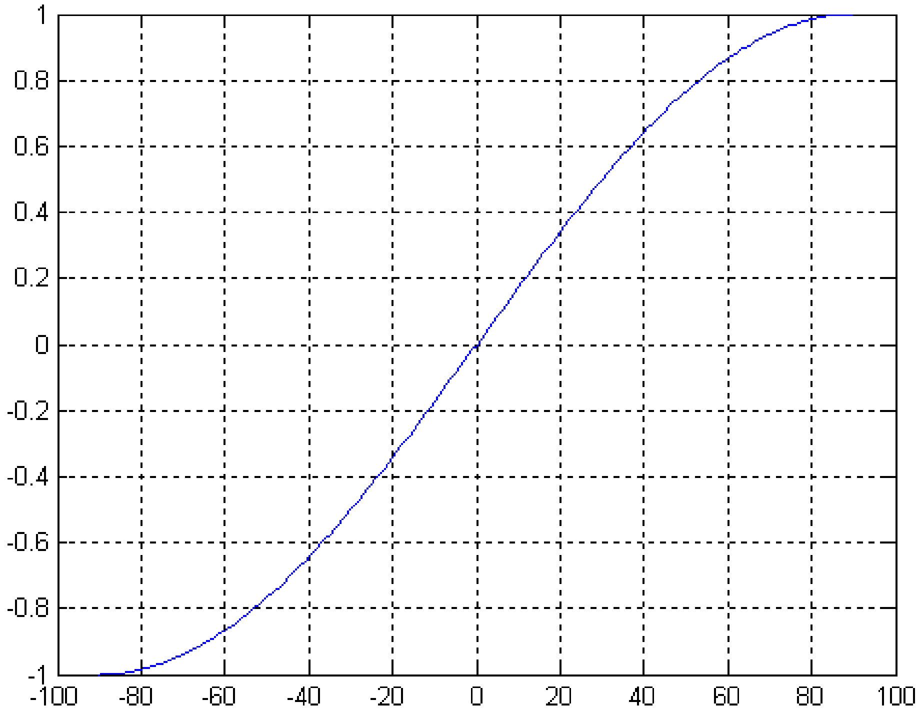

Figure 2.4 Signal Waveform

The gradient of this signal (which, in one dimension, is just the first derivative with respect to t) is shown in **Figure 2.5**.

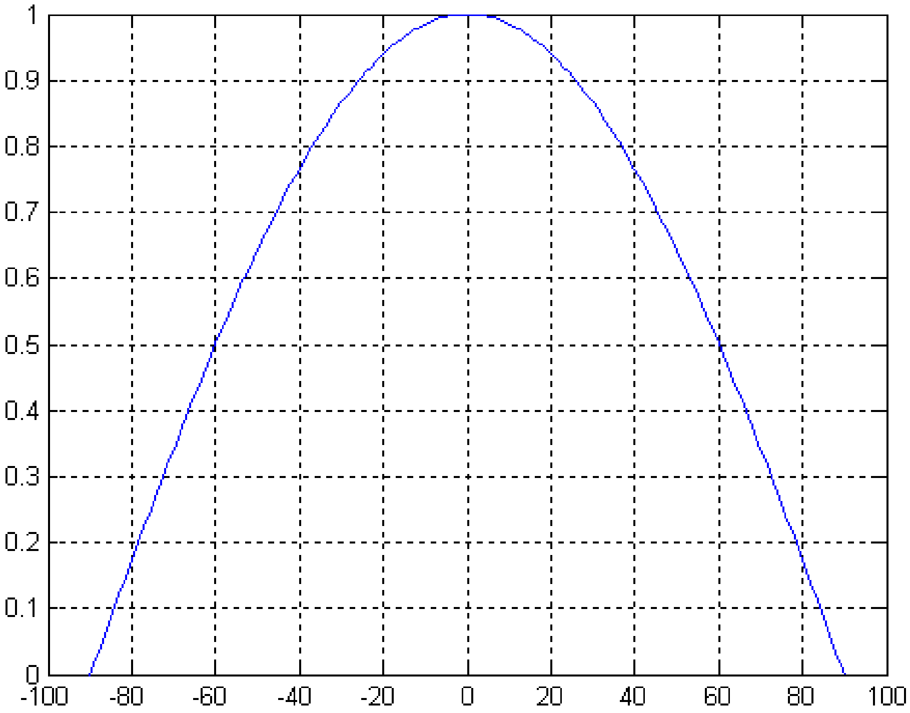

Figure 2.5 Derivative of the Signal

Clearly, the derivative shows a maximum located at the center of the edge in the original signal. This method of locating an edge is characteristic of the "gradient filter" family of edge detection filters. A pixel location is declared an edge location if the value of the gradient exceeds some threshold. So once a threshold is set, one can compare the gradient value to the threshold value and detect an edge whenever the threshold is exceeded. Furthermore, when the first derivative is at a maximum or minimum, the second derivative is zero. As a result, alternative way of finding the location of an edge is to locate the zeros in the second derivative. This method is known as the Laplacian and the second derivative of the signal is shown in **Figure 2.6**.

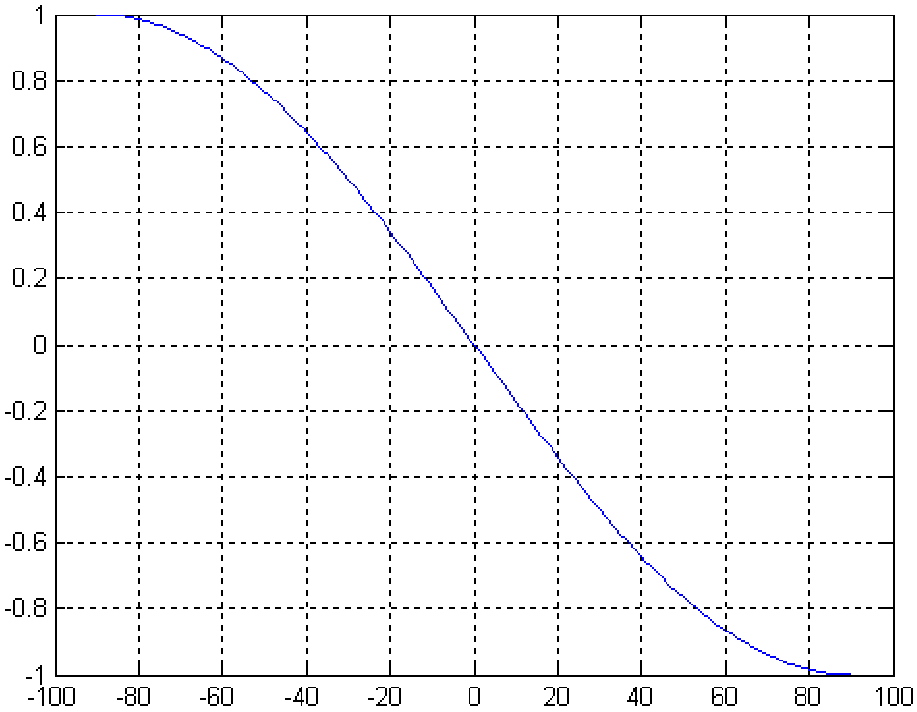

Figure 2.6 Second Derivative of the Signal

Most widely used gradient-based edge detection algorithms in computer vision area are Sobel, Roberts, and Canny operators whose brief descriptions are given below. Besides, Laplacian operator will be described shortly.

2.4.1 SOBEL OPERATOR

The operator consists of a pair of 3×3 convolution kernels as shown in **Figure 2.7**. One kernel is simply the other rotated by 90°.

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

Figure 2.7 Sobel Kernels

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (*Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given as

$$|G| = \sqrt{G_x^2 + G_y^2} \tag{2-1}$$

Typically, an approximate magnitude is computed using

$$|\mathrm{G}| = |\mathrm{G}_\mathrm{x}| + |\mathrm{G}_\mathrm{y}| \tag{2-2}$$

which is much faster to compute. A pixel location is declared an edge location if the value of the gradient exceeds some threshold. The angle of orientation of the edge giving rise to the spatial gradient is given by

$$\theta=\arctan\left(\frac{G_y}{G_x}\right) \tag{2-3}$$

2.4.2 ROBERT'S OPERATOR

The Roberts Cross operator performs a simple, 2-D spatial gradient measurement on an image. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point. The operator consists of a pair of 2×2 convolution kernels as shown in **Figure 2.8**. One kernel is simply the other rotated by 90°. This is very similar to the Sobel operator.

1	0
0	-1

0	1
-1	0

Figure 2.8 Robert's Kernels

These kernels are designed to respond maximally to edges running at 45° to the pixel grid but are not highly selective to orientation, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (*Gx* and *Gy*). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given as

$$|G| = \sqrt{G_x^2 + G_y^2} \tag{2-4}$$

Typically, an approximate magnitude is computed using

$$|\mathrm{G}| = |\mathrm{G_x}| + |\mathrm{G_y}| \tag{2-5}$$

which is much faster to compute.

The angle of orientation of the edge giving rise to the spatial gradient is given by:

$$\theta = \arctan\left(\frac{\mathrm{G_y}}{\mathrm{G_x}}\right) - \frac{3\pi}{4} \tag{2-6}$$

2.4.3 LAPLACIAN OF GAUSSIAN

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. The Laplacian is often applied to an image that has first been smoothed with something approximating a Gaussian Smoothing filter in order to reduce its sensitivity to noise. The operator normally takes a single gray level image as input and produces another gray level image as output.

The Laplacian *L(x,y)* of an image with pixel intensity values *I(x,y)* is given by:

$$\mathrm{L(x,y)} = \frac{\partial^2 \mathrm{I}}{\partial \mathrm{x}^2} + \frac{\partial^2 \mathrm{I}}{\partial \mathrm{y}^2} \tag{2-7}$$

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. Three commonly used small kernels are shown in **Figure 2.9**.

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

Figure 2.9 Three Commonly used Discrete Approximations to the Laplacian Filter

Because these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian Smoothed before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step.

In fact, since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter initially, and then convolve this hybrid filter with the image to achieve the required result. This will provide two advantages:

Since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations.

The LoG ('Laplacian of Gaussian') kernel can be precalculated in advance so only one convolution needs to be performed at run-time on the image.

The 2-D LoG function centered on zero and with Gaussian standard deviation σ has the form:

$$\mathrm{LoG}(x,y)=-\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad \textbf{(2-8)}$$

2.4.4 CANNY EDGE DETECTION ALGORITHM

Canny edge detection is considered to be optimal edge detection operator. Canny [48] defined optimal edge finding as a set of criteria that maximize the probability of detecting true edges while minimizing the probability of false edges. He found that the zero-crossings of the second directional derivative of a smoothed image were a reasonable measurement of actual edges. To smooth the image, the Canny edge detector uses Gaussian convolution, σ is the spread of the Gaussian and controls the degree of smoothing

$$g(m,n)=G_\sigma(m,n)*f(m,n) \qquad \textbf{(2-9)}$$

$$G_\sigma=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{m^2+n^2}{2\sigma^2}\right] \qquad \textbf{(2-10)}$$

Next, the image is convolved with a 2D first derivative operator to determine regions of sharp changes in intensities. The gradient magnitude and direction at each pixel are calculated in this step (**Figure 2.11**). Note that the maxima and minima of the first derivative gradient are the same as the zero-crossings of the second directional derivative.

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

Figure 2.10 Kernels

$$M(m,n)=\sqrt{g_m^2(m,n)+g_n^2(m,n)} \qquad \textbf{(2-11)}$$

$$\theta(m,n)=\tan^{-1}\left[\frac{g_n(m,n)}{g_m(m,n)}\right] \qquad \textbf{(2-12)}$$

Only the maxima crossings are of interest because these pixels represent the areas of the sharpest intensity changes in the image [34]. These zero-crossings are the ridge pixels that represent the set of possible edges. All other pixels are considered non-ridge and subsequently suppressed.

Finally, a two-threshold technique or hysteresis is performed along the ridge pixels to determine the final set of edges. Instead of using a single threshold value for filtering ridge pixels, the Canny algorithm implements a connected components analysis technique based on a hysteresis thresholding heuristic. This step uses two thresholds, t1, t2 where t1 > t2, to partition the ridge pixels into edges/non-edges. Pixels with gradient magnitudes above t1 are classified as definite edges. Pixels between t2 and t1 are classified as potential edges. Pixels under t2 are classified as non-edges. Next, all potential edges that can be traced back to a definite edge via adjacent potential edges are also marked as definite edges. The process solves some of the issues associated with edge streaking and discontinuity in the results achieved by simple detectors by identifying strong edges while accounting for comparatively weaker ones.

In this study t1 (*edge_Th_high*) and t2 (*edge_Th_low*) is set as 0.25 and 0.06 respectively.

The results of the described edge detection methods that are applied to an aerial image are given in **Figure 2.11- Figure 2.15**.

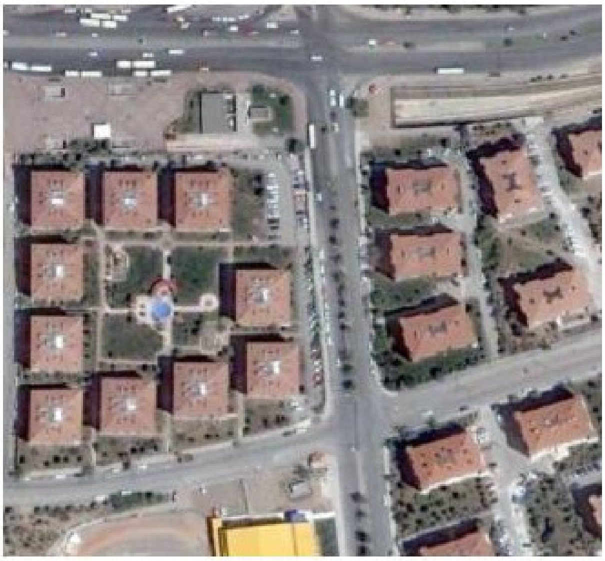

Figure 2.11 Original Image

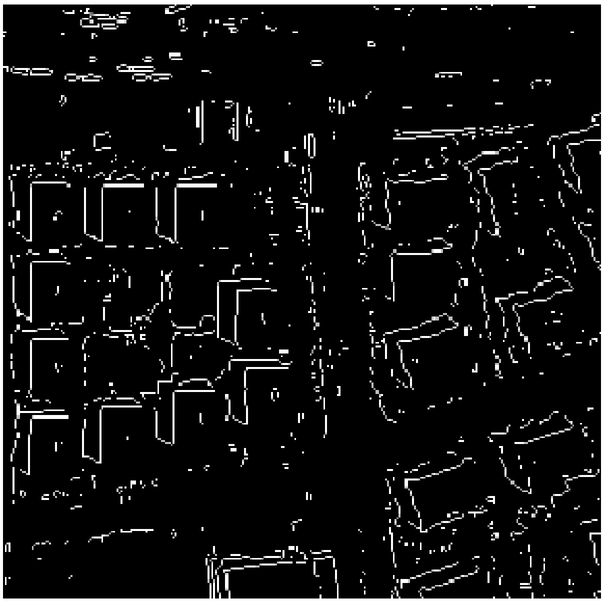

Figure 2.12 Sobel Edge Detected Image

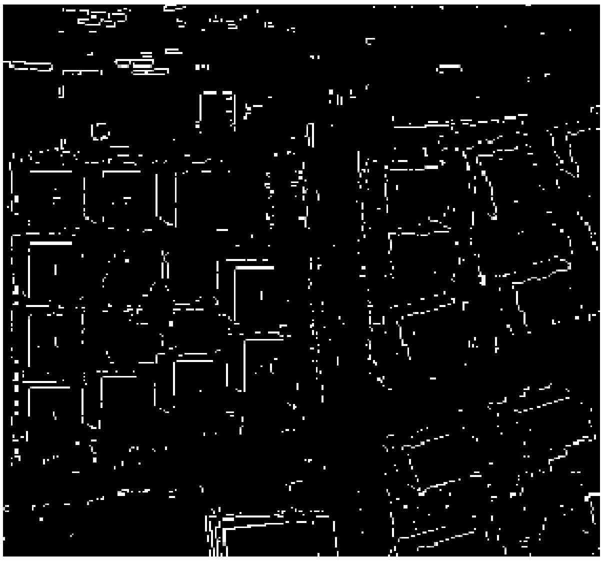

Figure 2.13 Roberts Edge Detected Image

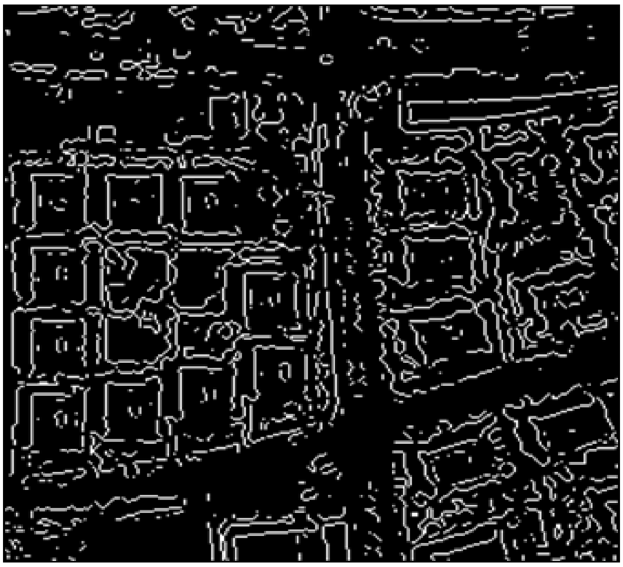

Figure 2.14 LoG Edge Detected Image

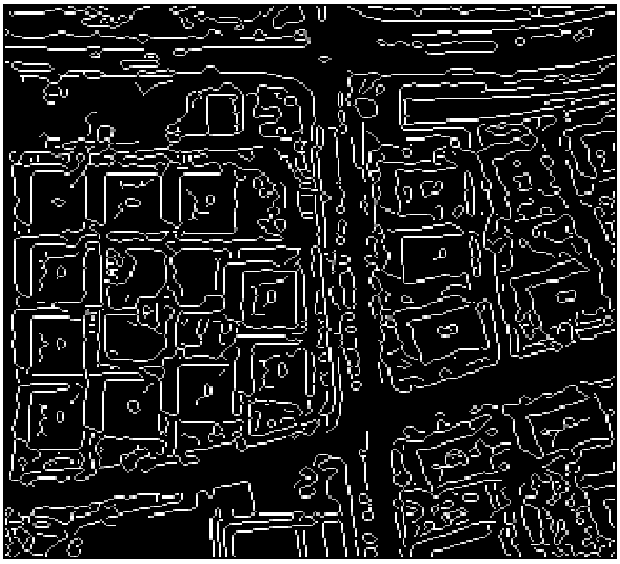

Figure 2.15 Canny Edge Detected Image

2.5 HOUGH TRANSFORM

After detecting the edges, the line forming process from the detected edges was carried out using the Hough Transform. The Hough Transform (HT) algorithm was developed by Paul Hough in 1962 ([35]). It is a method for detecting straight lines and curves, which can be represented by parametric equations, on gray level images. The underlying principle of the Hough Transform is that there are an infinite number of potential lines that pass through any point, each at a different orientation. The purpose of the transform is that it determines which of these lines pass through most features in an image -that is, which lines match most closely to the data in the image. In Hough Transform, each line is represented by two parameters that are commonly called *r* and θ representing the length and the angle of a normal from the origin to the line (**Figure 2.16**). Equation 2.13 represents a straight line for all (x,y) points satisfying the same (r,θ) values.

$$r = x \times \cos(\theta) + y \times \sin(\theta) \tag{2.13}$$

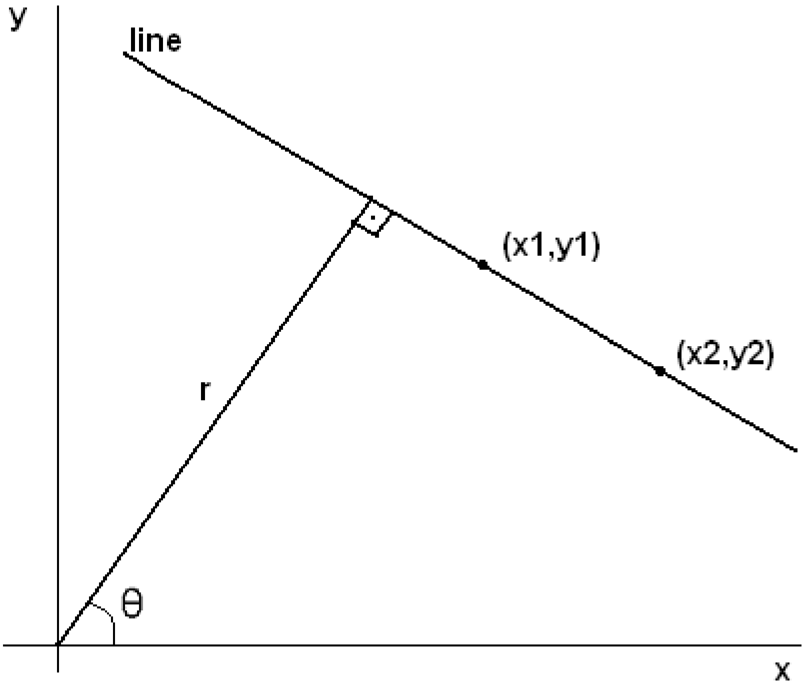

Figure 2.16 Parametric Description of a Straight Line

The aim here is to find the edge pixels that satisfy the same r and θ. For each edge pixel, θ values ranging from 1 to 180 is scanned and corresponding r value is stored in a vote matrix. Here the difference between the consecutive θ values which is a measure of the sensitivity of the Hough Transform is constant (*θdiff*). Smaller the value of this difference means more details can be detected but at the same time more noise will be present at the output and it is computationally inefficient. Contrary to this, larger the value of this parameter reduces the noise at the output but may miss some of the buildings' details in the image.

Initially, there is a two dimensional vote matrix corresponding to θ and r that are initialized to 0. If a pixel satisfies a (r; θ) pair, the corresponding entry in the matrix is increased by 1. After all the edge pixels have been processed, the values in the matrix are checked. If the value of a cell in the vote matrix is greater than a threshold value (*vote_th*), it is concluded that this (r; θ) pair represents a straight line who is passing through the satisfying (x,y) edge pixels.

CHAPTER 3

THE PROPOSED BUILDING DETECTION ALGORITHM

3.1 THE GENERAL STRUCTURE OF THE ALGORITHM

In this thesis a hypothesis based algorithm is proposed for building detection in satellite/aerial images. In the proposed approach, hypothesis generation finds out the building candidates and hypothesis verification eliminates the false ones. Line detection based feature extraction is employed for hypothesis generation, while color segmentation and shadow detection are employed for hypothesis verification.

In **Figure 3.1**, these steps and relation among them is shown as a flowchart. In the figure, the boxes represent the processes applied and ellipses represent the results.

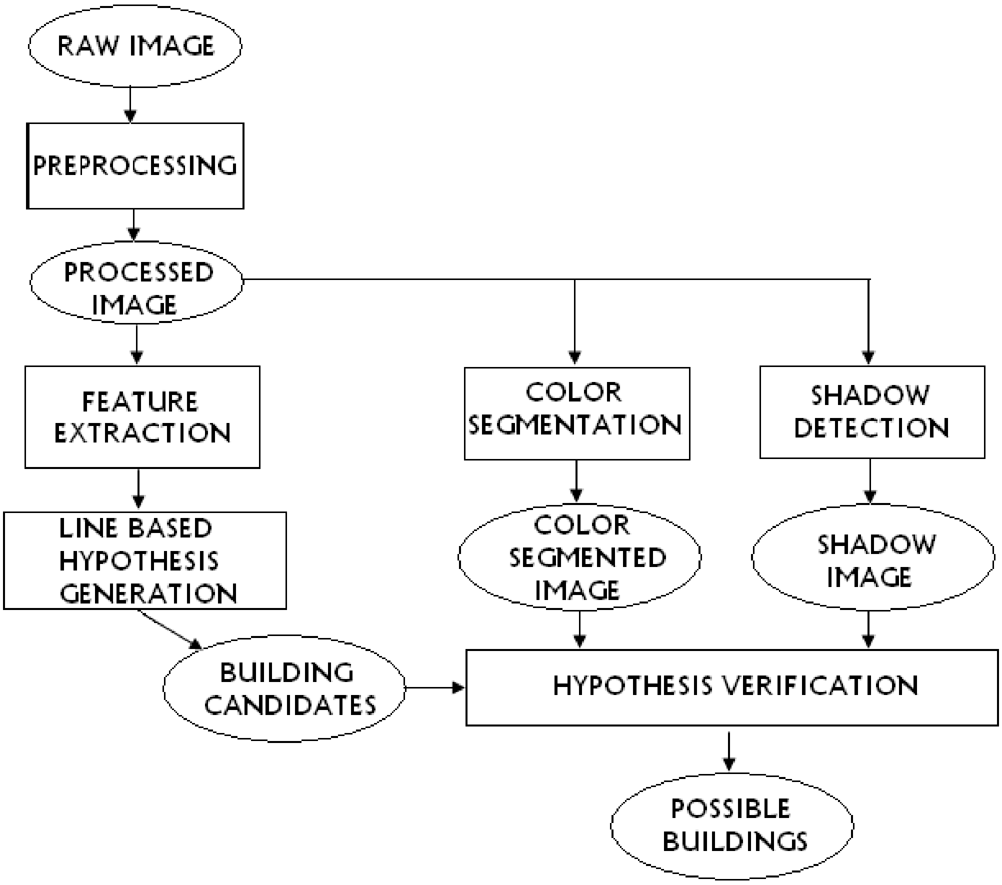

Figure 3.1 The Flowchart of the Proposed Building Detection Algorithm

A more detailed flowchart including the intermediate steps is shown in **Figure 3.2**. The preprocessing step consists of image sharpening and conversion of the image from RGB to HSV space. The processed image is obtained as the result of the preprocessing steps. In the figure boxes represent the methods used in the thesis and the ellipses represent the results.

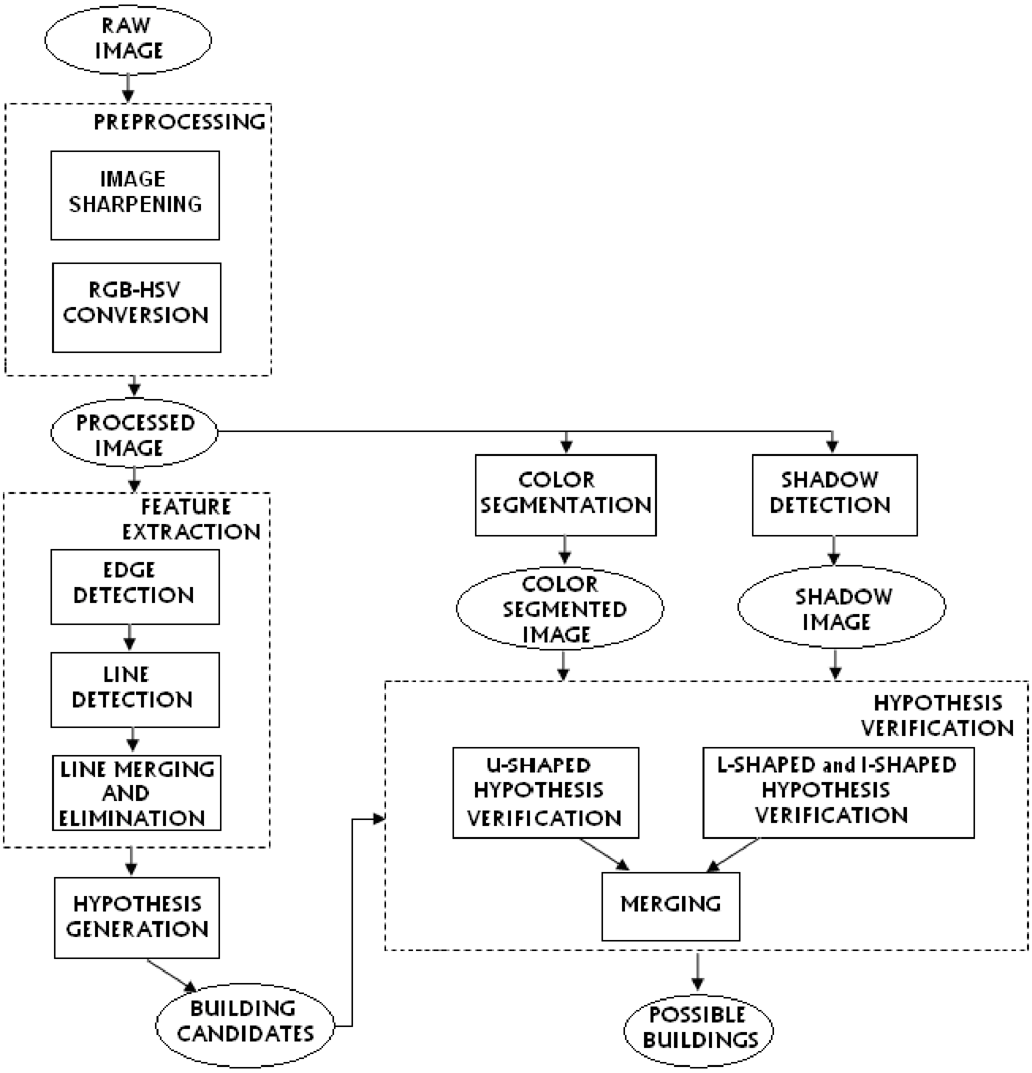

Figure 3.2 Detailed Flowchart of the Proposed Building Detection Algorithm

Feature extraction part consists of edge detection, line detection, line merging and elimination steps. The result of the feature extraction part is the lines which are possible borders of the buildings. In the hypothesis generation part, the building candidates are produced by considering the positions of the lines with respect to each other.

In the hypothesis verification part, both color segmented image obtained by color segmentation process and the shadow image obtained by shadow detection process are employed for the U-shaped hypothesis verification and also for the I and L-shaped hypothesis verifications. These candidate building borders are examined and the false candidates for each shape type are eliminated by the rules specific to each shape. Those candidates remaining after verification constitutes the possible buildings in the image. The details of all these steps are explained in detail in the following sections.

3.2 PREPROCESSING THE IMAGE

The performance of the proposed algorithm depends on the number of the successfully detected buildings, which is composed of edges. That is, the more building edges extracted from the image the more successful the algorithm. Therefore, a filter is applied initially to make the edges more apparent which will be detected by the edge detection process later on. In this study sharpening filter is applied in the preprocessing stage of the proposed algorithm to improve the edge detection performance (**Figure 3.3**).

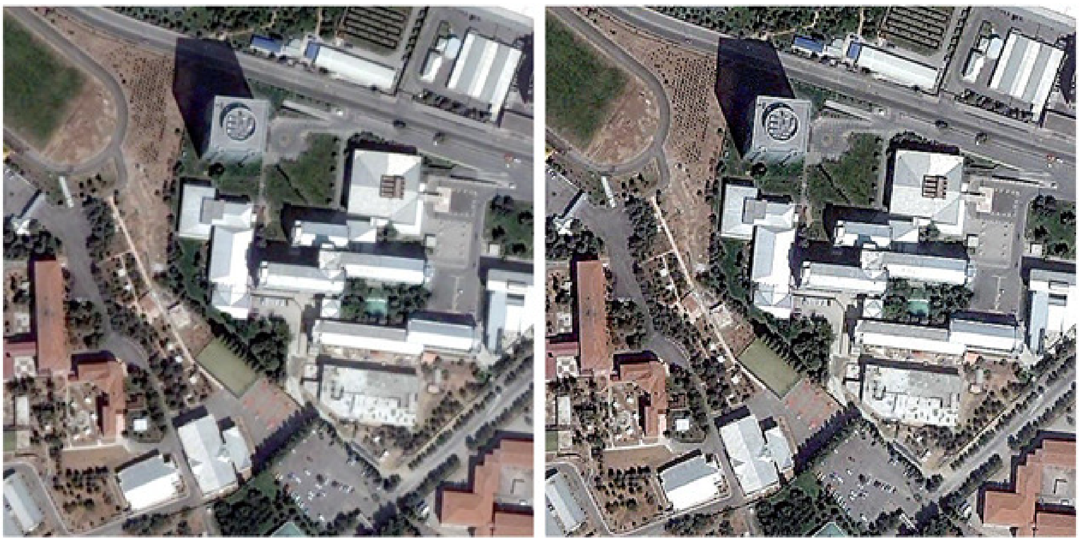

Figure 3.3 Original Image and Sharpening Filter Applied Image

3.3 COLOR SEGMENTATION

Performing the color segmentation in HSV space is better than doing so in RGB space since erroneous segmented regions may be produced in RGB space due to reflections or illuminations. In color segmentation stage, image is divided into three clusters according to the HSV-pixel values by performing k-means clustering algorithm. This algorithm used for segmenting the HSV representation of the image was explained in section 2.3 previously.

3.4 SHADOW DETECTION

The aim of shadow detection stage is to extract the shadow pixels from the image. RGB representation of the image is very sensitive to illumination and gives erroneous results when used to detect shadows. Therefore, the image is firstly converted to HSV representation from the RGB one. Then, a mask which filters the image to show higher values of saturation and lower values of value components on the HSV space. After applying this mask shadow pixels are extracted from the image (**Figure 3.4**)

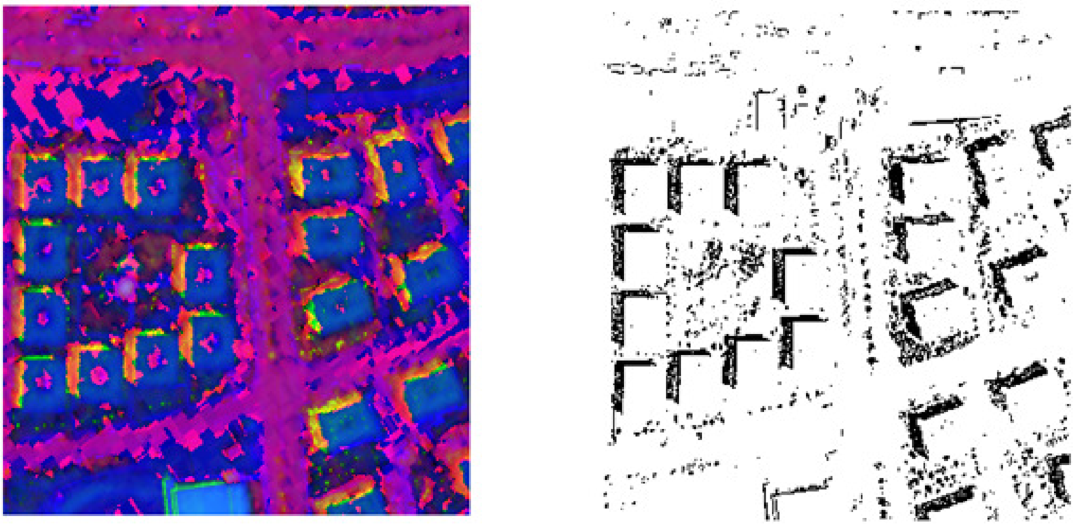

Figure 3.4 Shadow Detection Applied on an Image

The extracted shadow image consists of holes appearing among the shadow pixels. To fill these holes a morphological process consisting of erosion and dilation is applied which yields the image shown in **Figure 3.5**.

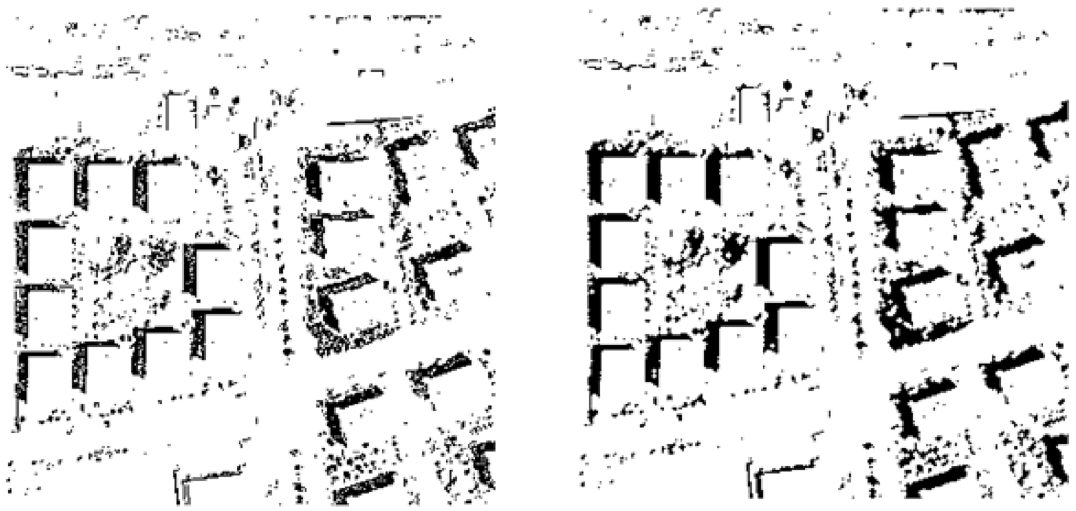

Figure 3.5 Shadow Image after Filling Holes

3.5 EDGE DETECTION

Edge detection stage extracts the edges from the image. Among the several known edge detection algorithms (Sobel, Canny, LoG, Roberts), "Canny Edge Detection" algorithm is used in this thesis to detect the edges. The Canny algorithm provides several advantages over the other edge detection techniques. It produces one pixel wide edges and connects the broken lines which are important for further processes. Canny edge detection algorithm was described in section 2.4.

3.6 HOUGH TRANSFORM

The edges detected in the previous step give us only the points with zero second derivates in the image. These points are not enough in detecting buildings.

In order to detect buildings, these discrete points be grouped to constitute lines which is what Hough Transform does. The Hough Transform was described in detail in section 2.5.

A brief explanation for line generation is as follows; for each edge pixel (x,y) in the image, there is a corresponding (Θ, r) pair. Hough transform states that any two or more points constituting a straight line has an intersection point in the (Θ, r) plane. This (Θ, r) point gives the corresponding line equation in x-y plane. Then for each edge pixel, algorithm searches for other nearby pixels that will form straight lines. The algorithm takes the distance between these pixels into account so that pixels with a distance larger than a threshold value (*line_min_dist*) which is set initially by the user, will not constitute a straight line. By this way, algorithm produces line segments whose start and the endpoints are known (**Figure 3.6**).

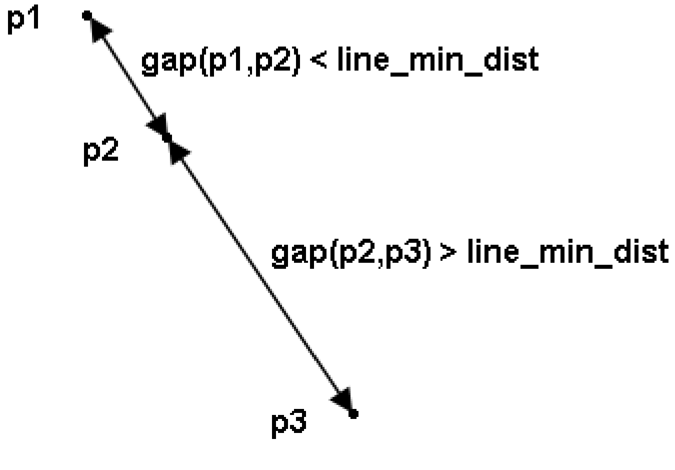

Figure 3.6 (p1, p2) Constitute Line; (p2, p3) Not Constitute Line

3.7 LINE MERGING AND ELIMINATION

The output of the Hough Transform stage is the line segments existing in the image. If the gap between the line segments is under a threshold value (*line_Th_merge*) then these lines are merged and the total number of lines extracted from the image is decreased. Therefore, the runtime of the algorithm which depends on the number of the lines is greatly improved by this stage. As well, the successful building detection rate is improved by line merging.

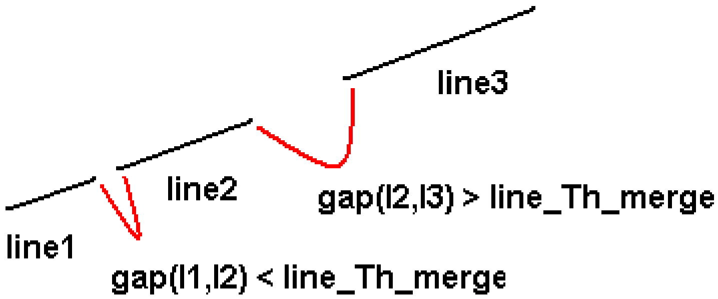

Figure 3.7 Line1 and Line2 are Merged; Line2 and Line3 not Merged

Line merging basically searches each line's distance to its neighbors. According to the threshold value whose default value is 2 and can be adjusted by the user, lines satisfying this threshold are merged. In addition to merging process, lines which are not merged and with a length lower or equal than 2 are eliminated (**Figure 3.7**). These lines are considered to be noise, and by eliminating them the runtime of the algorithm is also improved. (**Figure 3.8** , **Figure 3.9**)

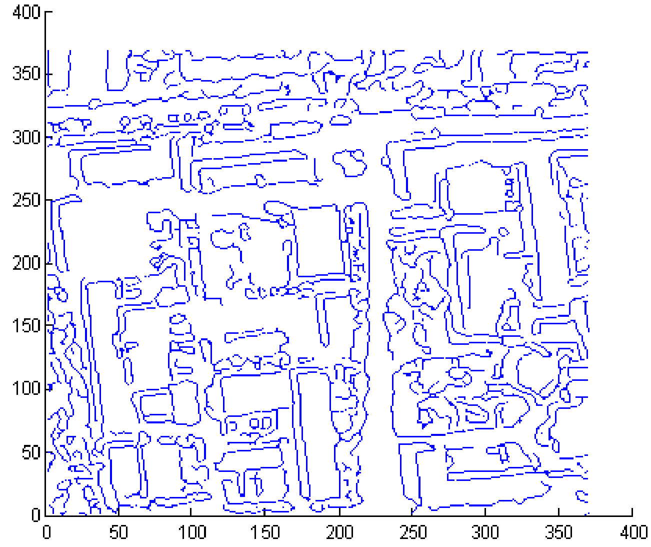

Figure 3.8 Lines Generated by Using Hough Transform

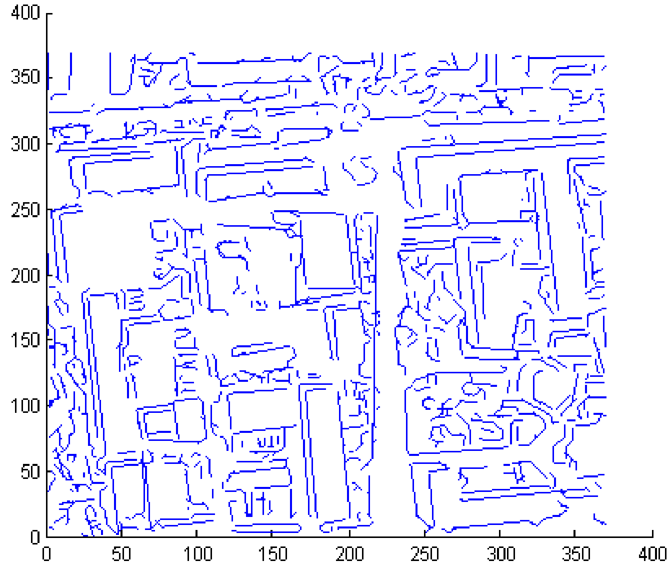

Figure 3.9 Merged and Eliminated Lines

3.8 HYPOTHESIS GENERATION

The goal of the hypothesis generation is to find candidate buildings in the image using the line information from the previous stage. In this thesis, only rectilinear shaped buildings are considered, so hypothesis generation stage searches for rectangular structures in the image. In order to improve the building detection rate, algorithm searches for 3-sided possible rectangles instead of searching for 4-sided rectangles taking into consideration that the fourth side may be lost during edge detection stage or some kind of occlusion due to trees or other structures in the image. In addition to this 3-sided rectangle search, algorithm searches for 2-sided possible rectangles, namely any "L-shapes" again assuming the remaining 2 sides may be lost similarly. Both of these searches assume that two straight line segments are nearly perpendicular to each other and the gap between the intersection points of these two lines is below a threshold value (*build_Th_inter*). And finally, algorithm also searches single line segments (I-shaped) that may be a part of a building and stores these selected lines as building hypotheses (**Figure 3.10**).

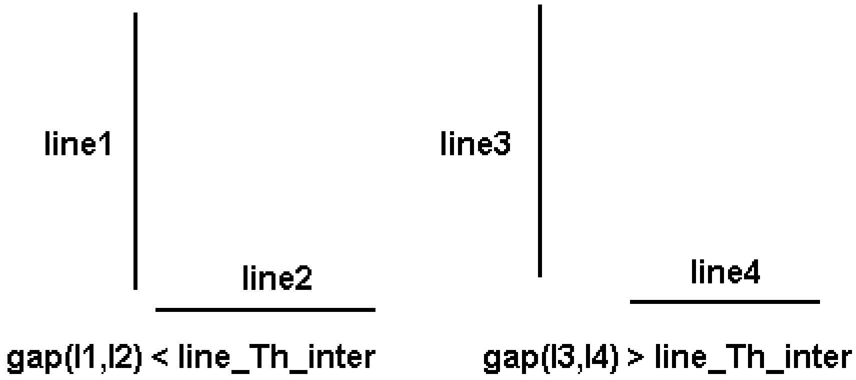

Figure 3.10 l1 and l2 Form a Corner; l3 and l4 are Separate

Here in this stage, algorithm generates the building hypotheses only if the line segments for each side of the candidate building are larger than the value of min_line_length parameter. By adjusting this parameter according to the image size, random line segments which are very small are discarded although they obey U, L or I-shaped hypotheses patterns (**Figure 3.11**)

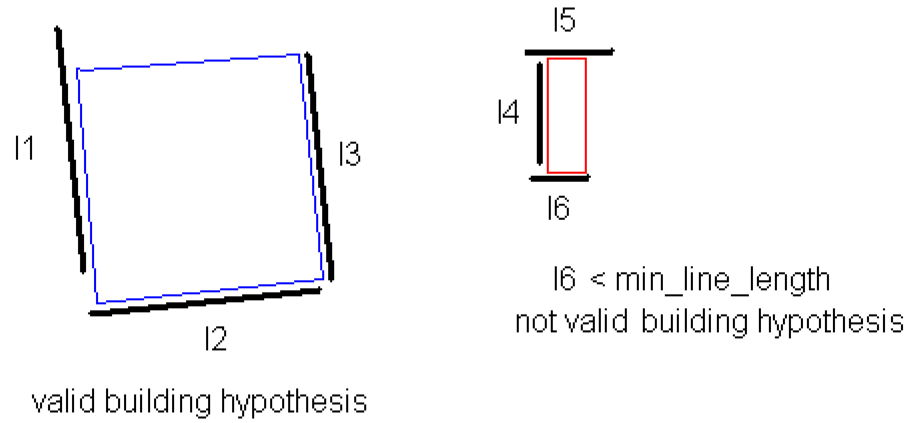

Figure 3.11 Valid and Invalid Building Hypotheses Based on min_lin_length Parameter

3.9 I-SHAPED AND L-SHAPED HYPOTHESIS VERIFICATION

This step uses shadow cast direction and the results of the three stages which are shadow detection, hsv-color segmentation and hypothesis generation to verify buildings in the image. Shadow cast direction is initially given as an input to the algorithm. From the shadow detection stage, algorithm has the shadow pixel coordinates. Hsv-color segmentation information is used to verify whether the candidate building's interior part lies mostly in one segment or not. The algorithm assumes nearly uniform color rooftops in the image and using HSV-color segmentation result some of the false hypotheses are eliminated. Majority of the pixels of the buildings' with nearly uniform color rooftops are represented in the hsv-color segmentation image in only one of the segments. Therefore, in the hypotheses verification stage, algorithm checks how much the candidate buildings' area lies in one of the segmented regions in the hsv-segmentation image. And if a predefined percentage of the area (*cluster_min_area*) lies in two or more segments, then the hypothesis is eliminated (**Figure 3.14**). From the hypothesis generation stage, algorithm has the 'I-shaped' and "L-shaped" line segments' coordinates. In order to state that a building exists there, shadow pixel coordinates should be adjacent to the "I-shaped" and "L-shaped" structures' coordinates according to the shadow cast direction. And the number of the shadow pixels that are adjacent, should be bigger than a threshold value (*sdw_line_rate*) which is a fraction of the line length in the "I" or "L-shaped" structure (**Figure 3.12**). According to this, algorithm eliminates the wrong hypotheses which violate this requirement. The remaining hypotheses indicate possible buildings in the image.

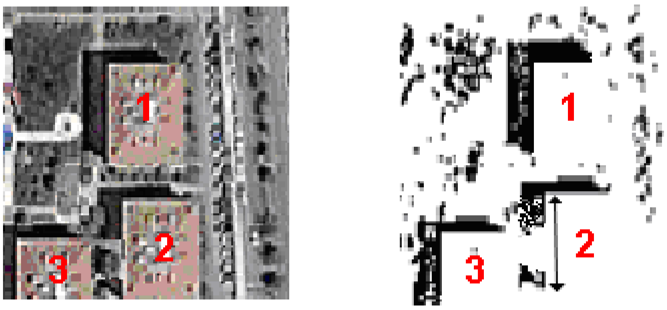

Figure 3.12 sdw_line_rate parameter effect on hypotheses

(Building #2's left line segment) x 0.85 > # of shadow pixels adjacent so, building #2 is eliminated.

For the "L-shaped" hypotheses, since only two line segments are known, the remaining other two lines are produced using the known ones. In other words, a symmetric "L-shaped" structure is drawn to the original "L-shaped" structure. For the "I-shaped" hypotheses, algorithm calculates the other three lines using the HSV color segmentation information. In other words, remaining three lines are the boundary of the HSV-color segment in which the original I-shaped line is the first part of the boundary. Here, the algorithm decides the three other boundaries using the parameter *cluster_color_diff*. Algorithm takes the I-shaped line as a basis and searches the adjacent hsv-color segment pixel values located at the other side from the shadow direction of this basis line. The search is performed perpendicular to the basis line and it starts from the basis line's start point and continues until the difference between two adjacent hsv-color segment pixel values are larger than the threshold *cluster_color_diff* value. This search is processed for each pixel value on the basis line. And according to the endpoints of the search, if the gap between start and endpoints are larger than the *min_line_length* threshold algorithm generates hypotheses (**Figure 3.13**).

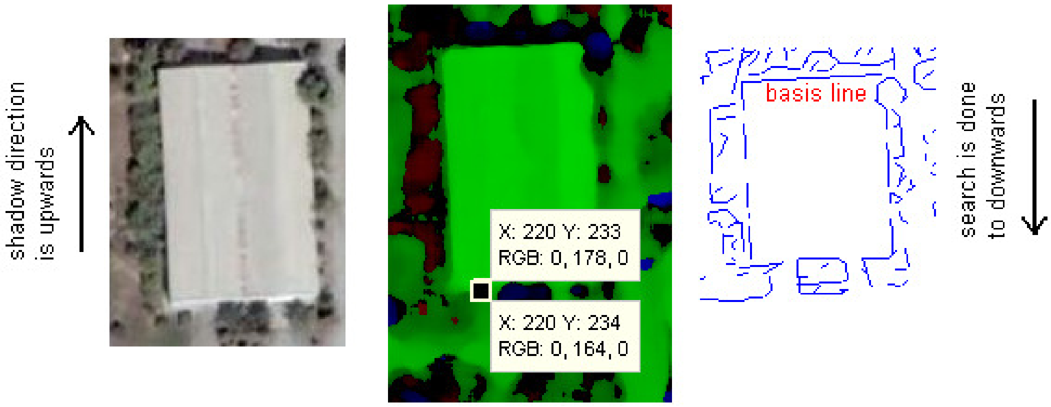

Figure 3.13 cluster_color_diff Effect on Hypotheses Verification

3.10 U-SHAPED HYPOTHESES VERIFICATION

This step uses the results of the three stages which are HSV-color segmentation, shadow detection and hypothesis generation to verify buildings in the image. The hypothesis generation stage extracts any “U-shaped” structures from the image some of which corresponding to buildings and remaining ones to false positives which are ground surfaces, parking lots, grass, playground etc. The algorithm assumes nearly uniform color rooftops in the image and using HSV-color segmentation result, and so some of the “U-shaped” false hypotheses are eliminated. Majority of the pixels of the buildings' with nearly uniform color rooftops are represented in the hsv-color segmentation image in only one of the segments. Therefore, in the hypotheses verification stage, algorithm checks how much the candidate buildings' area lies in one of the segmented regions in the hsv-segmentation image. And if a predefined percentage of the area (*cluster_min_area*) lies in two or more segments, then the hypothesis is eliminated (**Figure 3.14**)

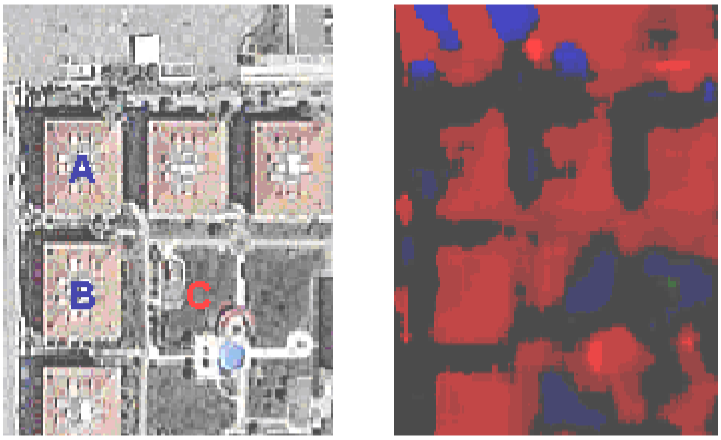

Figure 3.14 Effect of the cluster_min_area Effect on the Hypotheses

Majority of regionA and regionB pixels are in RED segment

Cluster_min_area_A > RED_segment_area_A

Cluster_min_area_A < BLUE_segment_area_A

Cluster_min_area_A < GREEN_segment_area_A

regionC pixels are in both RED and BLUE segments

Cluster_min_area_C > RED_segment_area_C

Cluster_min_area_C > BLUE_segment_area_C

Cluster_min_area_C < GREEN_segment_area_C

Besides, shadow detection stage output is also used for eliminating other false hypotheses. Line segments of the generated buildings' are searched if a predefined amount (sdw_line_rate) of shadow pixels are adjacent to them. And any "U-shaped" structure with more than %50 shadow pixels lying in this area is a false hypothesis and eliminated. The remaining hypotheses indicate possible buildings in the image.

CHAPTER 4

IMPLEMENTATION AND RESULTS

In this chapter, experimental results obtained by proposed building detection algorithm is evaluated under three different study areas.

The algorithm is implemented using MATLAB R2007a version on Windows XP Professional edition. Satellite/aerial images used in this thesis are bitmap images (.bmp), but any type image file, .jpg, .png etc. can be processed by the program.

The building detection performance of an aerial image is dependent to the image quality and the image characteristics, namely the building orientations, building sizes, cast shadow and environment features are the main elements that can change the success rate of the algorithm. In order to see the performance of the proposed algorithm, it is applied to 9 different areas each having different building orientations, cast shadows, building types and different surface materials with different reflectance properties. Detailed explanations for the first 3 of the images will be given in the following subsections, and also there will be images for each study area which are original image of the area, shadow detected image, hsv-color segmented image, edge detected image, Hough Transform applied image, generated hypotheses image and finally verified hypotheses image.

4.1 TEST AREA1

4.1.1 TEST AREA 1 PROPERTIES

First test image (**Figure 4.1**) is 290x290 pixels wide and was taken from Batikent, Ankara. The buildings are separated from each other, none of them is occluded, cast shadows are clear, orientation of the buildings are different, and all of them but one has the same rooftop color and type. Rooftops are not uniform most of the rooftops have different colored chimneys and the background is not heavily populated; some cars, subway station and parking lots add some ambiguity to the image.

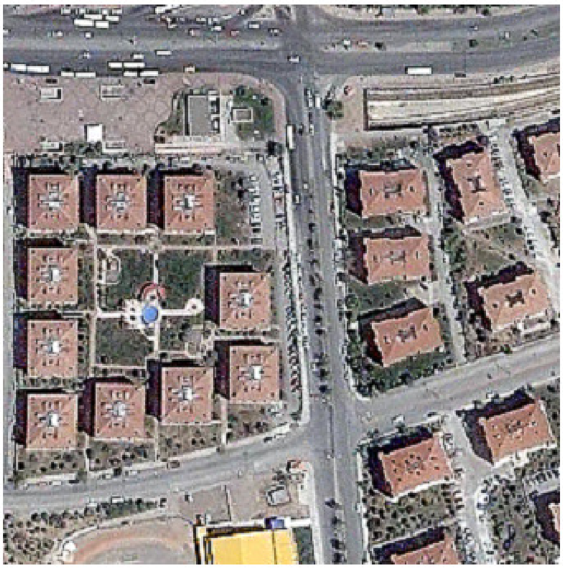

Figure 4.1 Original Image of the Test Area**1**

First, the algorithm converts the image into HSV representation and applies k-means algorithm to produce color (HSV) segmented image (**Figure 4.2**). In this figure, it is observed that the buildings have uniform segments except that the chimneys distract the uniformity.

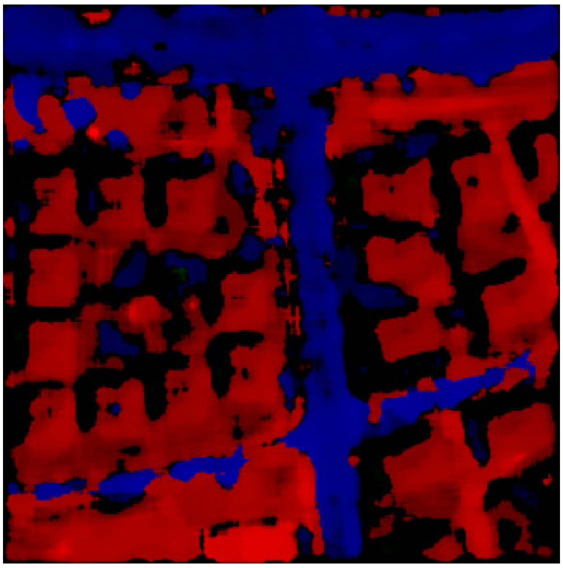

Figure 4.2 HSV Color Segmented Image of Test Area1; k=3

Then, shadow mask is performed to the original (RGB representation) image which yields shadow image (**Figure 4.3**). The shadow image nearly finds all the shadow pixels and there are very few non-shadow or false positive pixels.

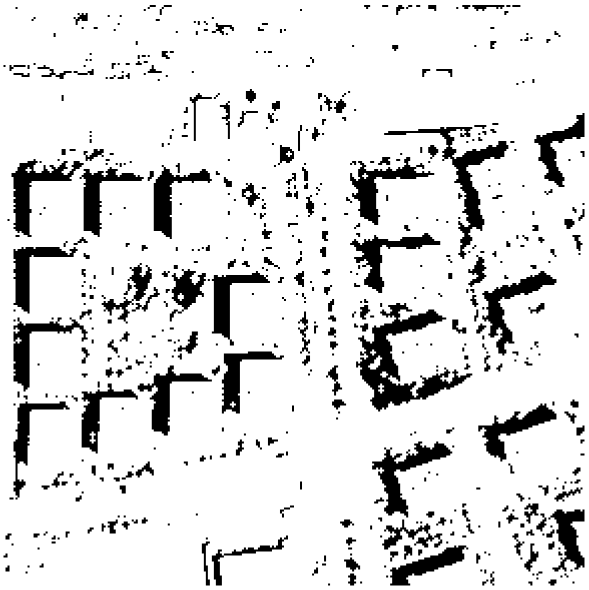

Figure 4.3 Shadow Image of the Test Area1

Next, the edge detection algorithm is applied to the image to produce edge pixels (**Figure 4.4**). It is seen from the figure that the edge detection algorithm functions well and finds nearly all the edges related with the buildings.

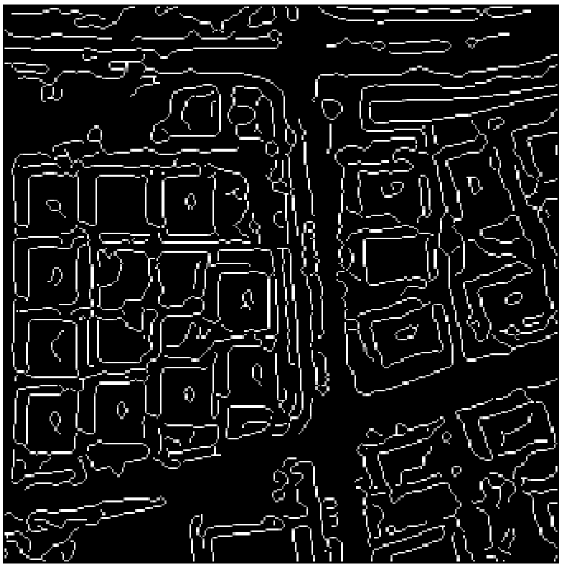

Figure 4.4 Edge Detected Image of Test Area1

After edge detection stage, Hough Transform is processed by using the edge information (**Figure 4.5**). Then, the lines are removed if their length is under a given threshold (threshold is 2) and the adjacent lines generated from the Hough Transform stage is merged if and only if the difference between them is under a given threshold, named line_Th_merge (threshold is assigned as 2) (**Figure 4.6**).

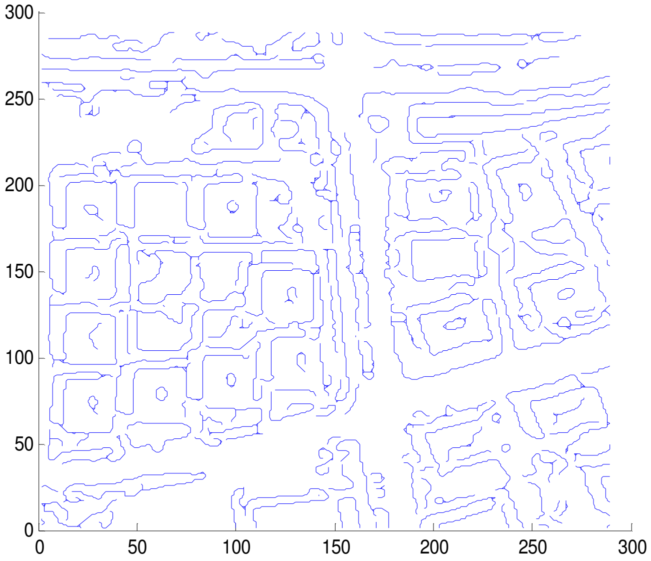

Figure 4.5 Hough Transform Applied to Test Area1

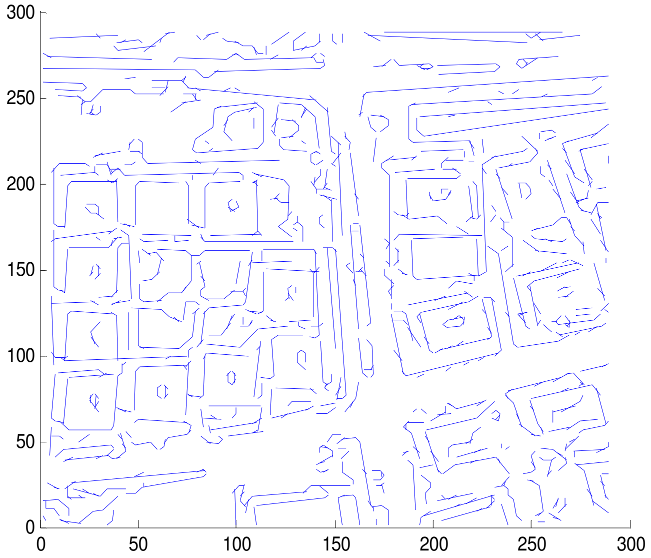

Figure 4.6 Test Area1 after Line Segments are Merged and Removed

After line segments are updated, the hypotheses are generated (**Figure 4.7**) and verified using the information of the HSV-color segmented image and shadow image (**Figure 4.8**). Here in the figures, the red colored boundaries correspond to the U-shape verified hypotheses and the blue colored boundaries correspond to the L-shaped and/or I-shape verified hypotheses.

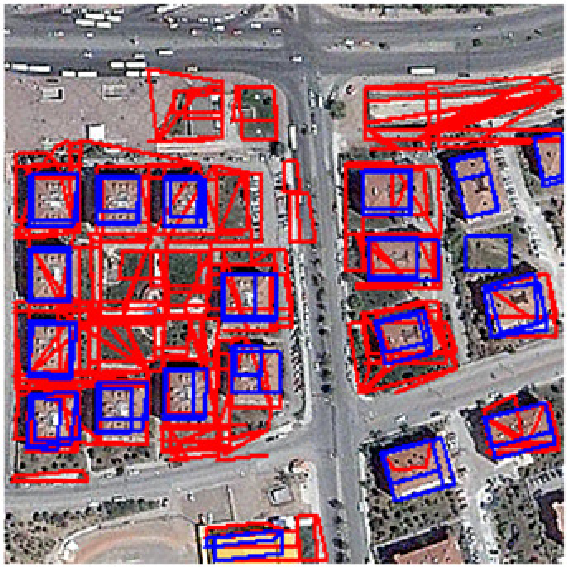

Figure 4.7 Generated Hypotheses for Test Area1

(U-shaped boundaries are shown by red line segments and L-shaped and/or I-shaped boundaries are shown by blue line segments.)

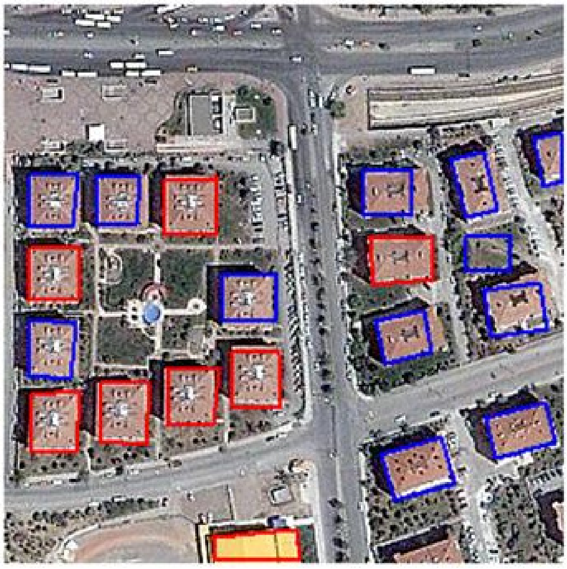

Figure 4.8 Verified Hypotheses for Test Area1

4.1.2 TEST AREA 1 RESULTS AND EVALUATION

For test area1 image, all of the buildings are successfully detected, one false positive is generated and no buildings are missed. False positive is generated because the line segment produced by the Hough Transform stage at that location in the image has enough shadow pixels adjacent to itself which is sufficient for the I-shape building hypothesis verification. Here the trees are the source of this shadow. This false positive can be avoided by changing the line shadow threshold value. For the present case, it is 0.85 times the length of the line segment.

4.1.3 RUN-TIME EVALUATION

The run-times for the system depend on the complexity of the images and the number of features in them. For the test area1 shown in **Figure 4.1-Figure 4.4** the total run time was approximately 189 seconds. Color segmentation and shadow extraction take 18 and 0.65 seconds respectively. Edge and line detection take 118 seconds. Hypothesis formation and verification required 40 and 12 seconds, respectively. The runtimes grow approximately exponentially with the size of the image for scenes of similar complexity.

4.1.4 FEATURES EVALUATION

The algorithm relies on several data to correctly detect the buildings from the satellite/aerial image. The most important two of these are the number of the extracted edges and the lines generated.

Table 4.1 gives these numbers for the test area1

Table 4.1 Test Area1 Features

	# of edge pixels extracted	# of lines generated
Testarea1	8710	3704

4.1.5 DETECTION EVALUATION

The following measures proposed in [8] are computed to evaluate the performance of the algorithm. Tp (True Positive) is a building in the test image and detected by the program. Fp (False Positive) is a building detected by the program but not present in the test image. Tn (True Negative) is a building in the test image but not detected by the program. These are combined to give the following two numbers:

$$\text{DetectionPercentage} = 100 \times \frac{T_p}{(T_p + T_n)} \qquad \textbf{(4.1)}$$

$$\text{BranchFactor} = 100 \times \frac{F_p}{\left(T_p + F_p\right)} \qquad \textbf{(4.2)}$$

In the above computations, a building is considered to be detected if a part of the building is detected by the system. The description of the detected building may not necessarily be correct. The accuracy of the shape is determined by counting correct building and nonbuilding pixels. In this study following pixel measures are used; the percentage of the number of pixels correctly labeled as building pixels over the number of building pixels in the image, the percentage of the number of pixels incorrectly labeled as building pixels over the number of pixels labeled as building pixels, and the percentage of the number of pixels correctly labeled as nonbuilding pixels over the number of nonbuilding pixels in the image.

The results for the test area1 is given in the **Table 4.2**

Table 4.2 Results for the Test Area1

	Tp	Tn	Fp	Detection Percentage	Branch Factor	Correct Building Pixels Per.	Incorrect Building Pixels Perc.	Correct Nonbuilding Pixels Perc.
ta1	19	0	1	100%	5%	90%	4.2%	98.6%

4.1.6 TEST AREA 1 PARAMETER EVALUATION

In this test case, parameters for various stages of the algorithm are the default values (**Table 4.3**)

Table 4.3 Some of the Default Parameter Values

Parameter	Value
min_line_length	image_width/35
edge_Th_high	0.25
edge_Th_low	0.06
cluster_color_diff	10

In order to compare the effect of the parameter values on the algorithm, some of the parameter values are changed. Firstly for the test area1, canny edge detection threshold values which are edge_Th_high and edge_Th_low are increased from 0.25 to 0.5 and from 0.06 to 0.125 respectively. The expected results of this change are; decrease in the number of the edges extracted, decrease in the number of generated lines, decrease in the number of detected buildings, decrease in the false positives and speedup in the execution time. The results of the features for this change are given in the **Table 4.4** .

Table 4.4 Results for the Modified Edge Threshold Values for the Extracted Features

	# of edge pixels extracted	# of lines generated	runtime(sec)
Default edge threshold values	8710	3704	189
Increased edge threshold values	5262	2196	93

The number of the extracted edge values directly affects the building detection performance; this is shown in **Figure 4.9** to **Figure 4.11** which shows the edge detection results, line generation results and verified buildings results respectively.

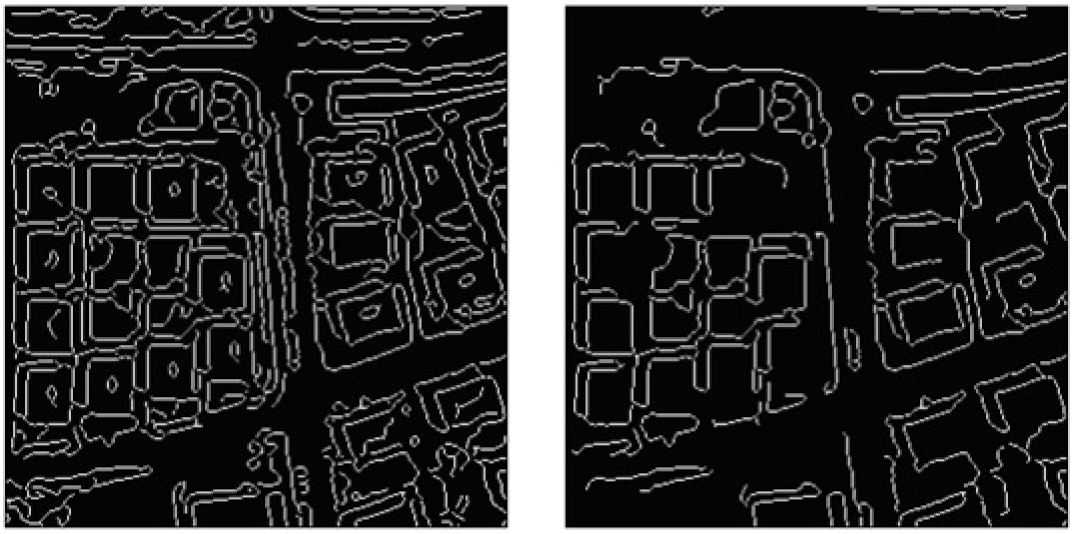

Figure 4.9 Edge Detection Results

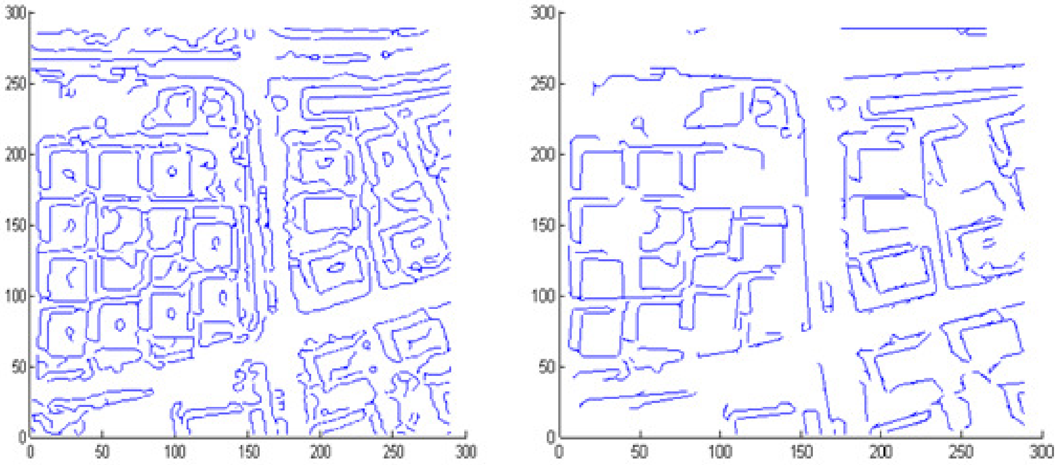

Figure 4.10 Generated Lines Results

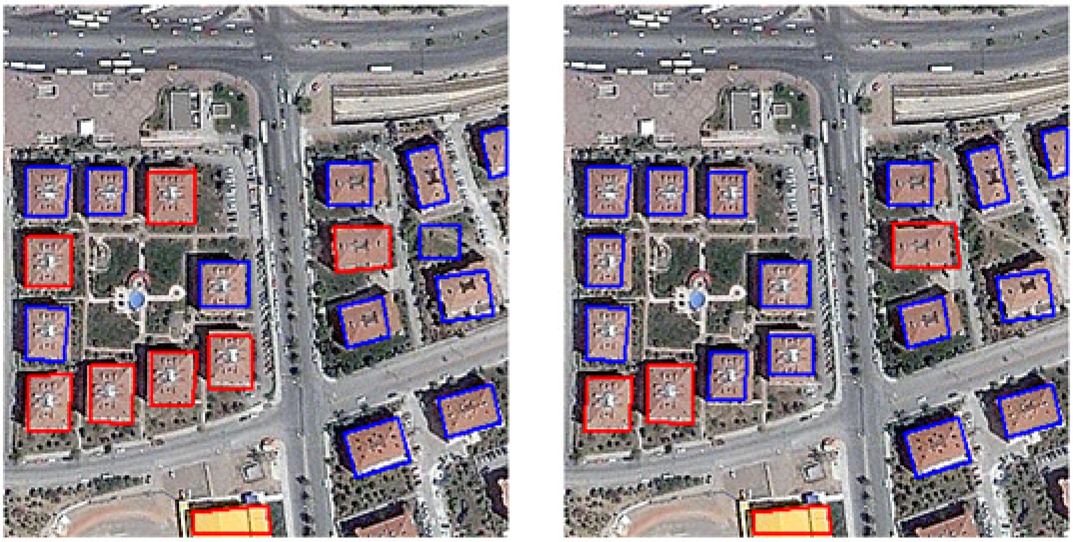

Figure 4.11 Verified Buildings Results

The results of this parameter change are consistent with the expectation. That is, number of extracted edges and the generated lines are decreased considerably which leads to a smaller runtime of the algorithm. Besides, the number of red-color plotted building boundaries which are created using U-shaped building hypotheses are decreased to 4 from 8 which means that some of the line segments constituting the U-shapes are not generated anymore. And finally, one false positive is eliminated due to line segment which is causing the false positive, is not generated with the increased edge threshold values.

Finally, a performance evaluation is given in **Table 4.5** to show the effect of the edge detection threshold change.

Table 4.5 Modified Edge Threshold Values for Performance Evaluation

	Tp	Tn	Fp	Det. Perc.	Br. Ft.	Correct Building Pixels Perc.	Incorrect Building Pixels Perc.	Correct Nonbuilding Pixels Perc.
Testarea1 (default)	19	0	1	100%	5%	90%	4.2%	98.6%
Testarea1 (modified)	19	0	0	100%	0%	88%	1.9%	99.1%

4.2 TEST AREA 2

4.2.1 TEST AREA 2 PROPERTIES

Second test image is 465x465 wide and was taken from Ankara (**Figure 4.12**). The test area2 dataset differs from the test area1 dataset in that buildings have varying shape, size, orientation, and roof intensity. The buildings have very distinct markings on the roofs that complicate the task of delineating them and rooftops are different from each other in terms of both color and shape. The site has as a small number of prominent roads that create distinct features in the images. Parking lot and the trees also add random line patterns and occlude some buildings.

Similarly the same steps are applied to the test area2 as applied to the test area1, which give hsv segmentation (**Figure 4.13**), shadow detection (**Figure 4.14**), edge detection (**Figure 4.15**), line generation (**Figure 4.16**), line removing and merging (**Figure 4.17**), building hypotheses generation (**Figure 4.18**) and building hypotheses verification images (**Figure 4.19**)

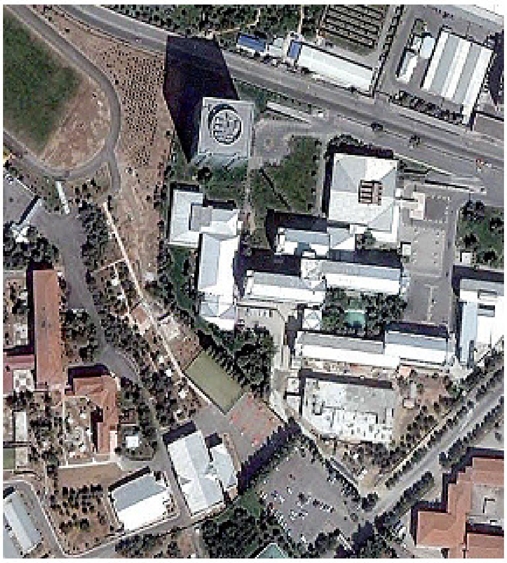

Figure 4.12 Original Image of the Test Area2

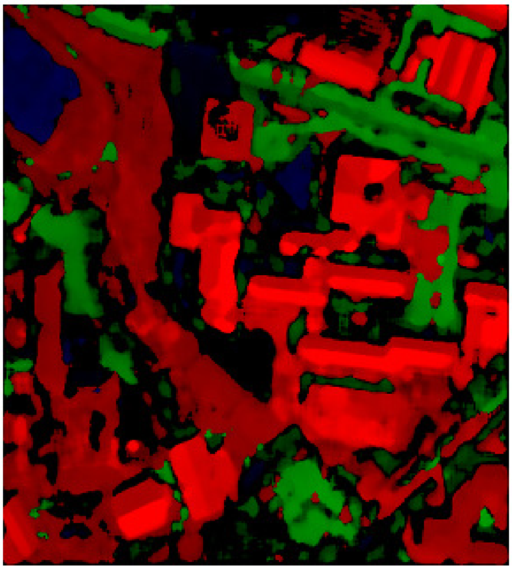

Figure 4.13 HSV Color Segmented Image of Test Area2; k=3

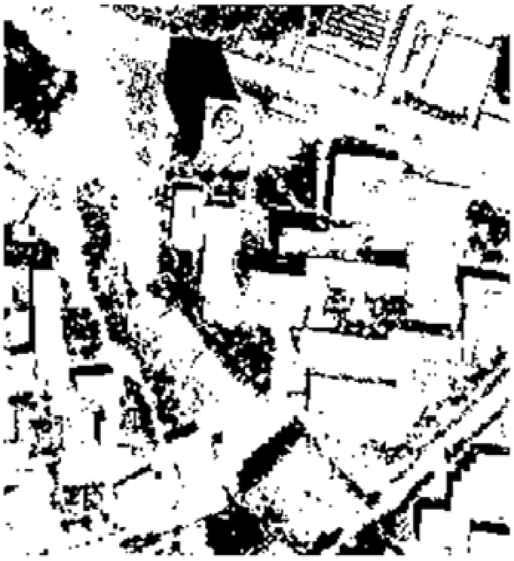

Figure 4.14 Shadow Image of the Test Area2

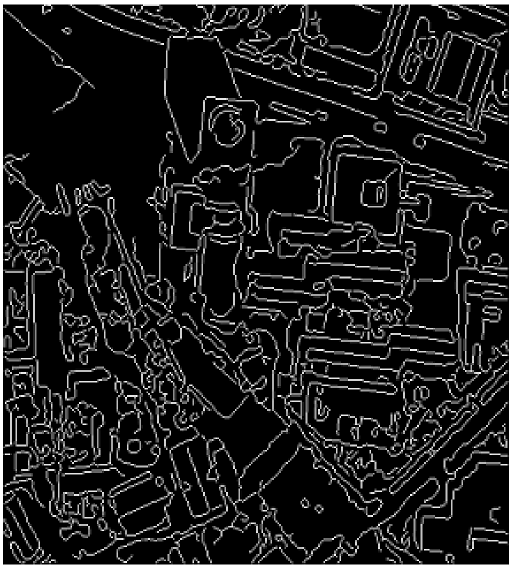

Figure 4.15 Edge Detected Image of Test Area2

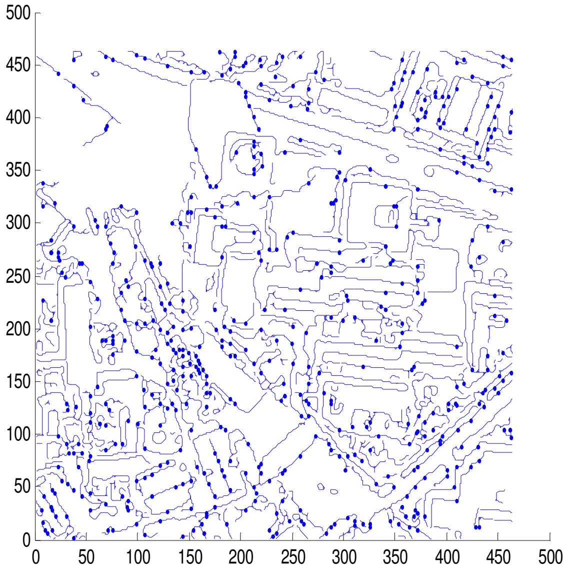

Figure 4.16 Hough Transform Applied to Test Area2

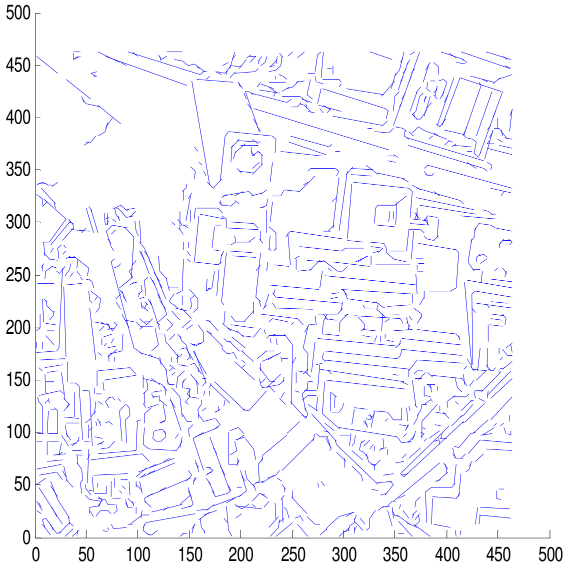

Figure 4.17 Test area2 after Line Segments are Merged and Removed

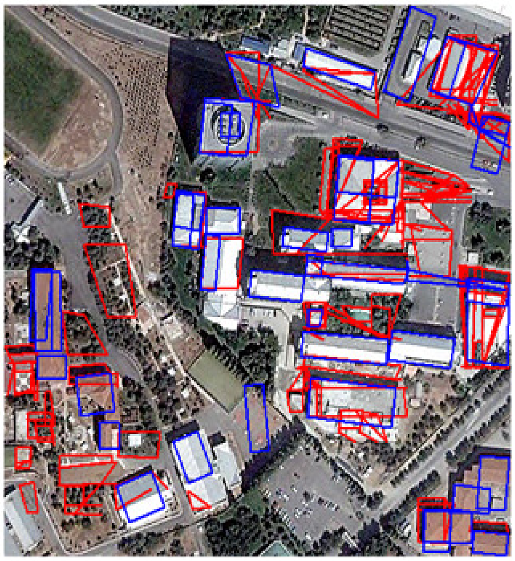

Figure 4.18 Generated Hypotheses for Test Area2

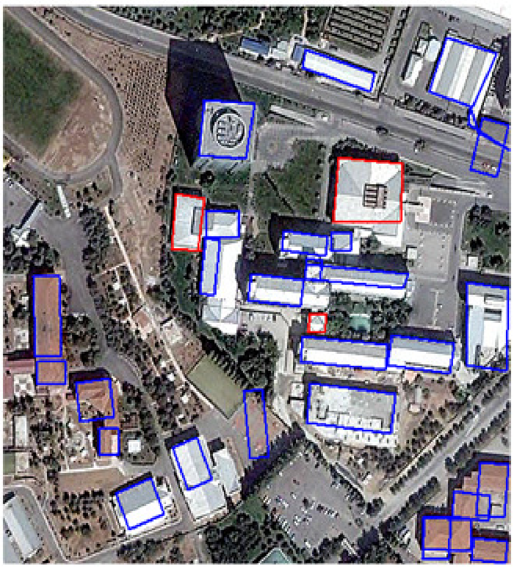

Figure 4.19 Verified Hypotheses for Test Area2

4.2.2 TEST AREA 2 RESULTS AND EVALUATION

The building detection performance for the test area2 image is not as high as the test area1 due to the increased scene complexity and the building size, shape, and number. And the image has also foliage and trees that clutter the background, as well as a number of manmade structures, such as car park areas and vehicles, which create accidental alignment of features that, may qualify as buildings.

For this image, 25 buildings are successfully detected, two false positives are generated and 4 buildings are missed. False positives are generated because the line segment produced by the Hough Transform stage at that location in the image has enough shadow pixels adjacent to itself which is sufficient for the I-shape building hypothesis verification. Here again the trees are the source of this shadow. This false positive can be avoided by changing the line shadow threshold value. For the present case, it is 0.85 times the length of the line segment. 2 buildings are missed due to the minimum building length threshold is not satisfied and the remaining 2 are missed since the area of the buildings' in the HSV-color segmentation image are not larger than the threshold value of the parameter cluster_min_area to be verified as a building.

4.2.3 RUN-TIME EVALUATION

The run-times for the system depend on the complexity of the images and the number of features in them. For the test area2 shown in **Figure 4.12** the total run time was approximately 750.61 seconds. Color segmentation and shadow extraction take 59 and 1.61 seconds respectively. Edge and line detection take 598 seconds. Hypotheses formation and verification required 74 and 18 seconds, respectively. The runtimes grow approximately exponentially with the size of the image for scenes of similar complexity.

4.2.4 FEATURES EVALUATION

The features for the test area2 image will be more than the test area1 since the image size is larger and the image is more complex. These numbers are given in **Table 4.6** .

Table 4.6 Test Area2 Features

	# of edge pixels extracted	# of lines generated
Testarea2	17651	8759

4.2.5 DETECTION EVALUATION

The same measures with the ones used in the test area1, are used to evaluate the performance of the algorithm applied to the test area2. Similarly, the statement which defines; a building is considered to be detected if a part of the building is detected by the system is valid. The description of the detected building may not necessarily be correct. The results for the test area2 are given in **Table 4.6**.

Table 4.7 Results for the Test Area2

	Tp	Tn	Fp	Detection Percentage	Branch Factor	Correct Building Pixels Perc.	Incorrect Building Pixels Perc.	Correct Nonbuilding Pixels Perc.
Ta2	25	4	2	86.2%	7.4%	81.1%	6.9%	95.8%

4.2.6 TEST AREA 2 PARAMETER EVALUATION

In this test case, parameters for various stages of the algorithm are the same as the test case1 but min_line_length is decreased (**Table 4.8**). In order to detect small buildings which are present on the test image2, algorithm should take into consideration smaller line lengths for candidate buildings. Therefore, initially min_line_length parameter is set as small in this test.

Table 4.8 Parameter Values for the Test Area2

Parameter	Value
min_line_length	image_width/40
edge_Th_high	0.25
edge_Th_low	0.06
cluster_color_diff	10

In order to compare the effect of the parameter value on the algorithm, min_line_length parameter value is changed. This parameter is increased to image_width/35 from image_width/40. The expected results of this change are; decrease in the number of the successfully detected buildings, decrease in the number of false positives, and speedup in the execution time.

min_line_length parameter value can be thought of as a tradeoff between the number of the false positives and number of successfully detected buildings. Decreasing this parameter value will probably increase both the successfully detected buildings and the false positives. Therefore, decreasing the value of this parameter should be decided according to the satellite/aerial image. If an image has a lot of small-sized buildings, then this parameter can be decreased so that the improvement on the detection percentage rate is well beyond the increase in the Branch Factor. For most of the satellite/aerial images default value for this parameter is at the optimum. The results of this parameter change are given in **Figure 4.20** which shows the verified buildings images.

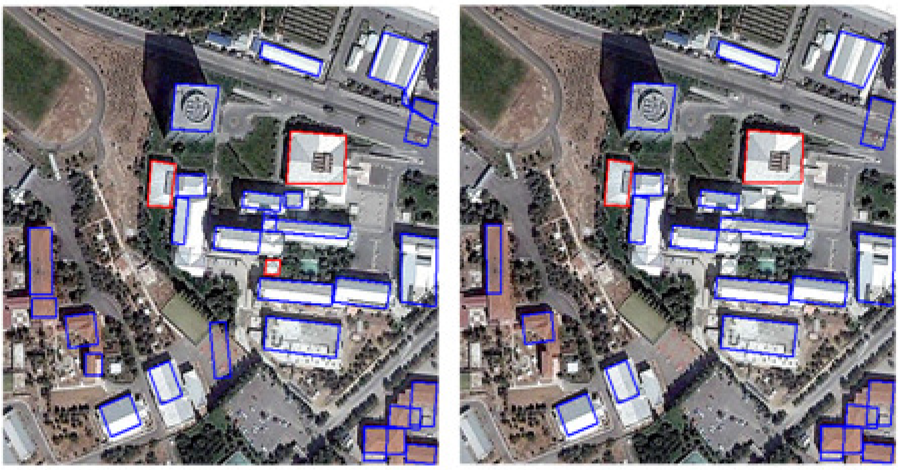

Figure 4.20 Verified Buildings Results

The result of this parameter change is consistent with the expectation. That is, number of successfully detected buildings and the false positives are decreased, and since there are fewer buildings detected, the runtime of the algorithm is improved approximately 20 seconds. Previously successfully detected 3 buildings are no longer detected with this parameter change. This change increased the min_line_length parameter value to 13 (imagewidth/35) from 11 which means these 3 missed buildings have line length lower than 13. Besides missed buildings, the number of the false positives is decreased to 1 which improves the Branch factor a little. Performance evaluation is given in **Table 4.9** to show the effect of the min_line_length threshold change.

Table 4.9 Results for the Modified min_line_length Value for Performance Evaluation

	Tp	Tn	Fp	Det. Perc.	Br. Ft.	Correct Building Pixels Perc.	Incorrect Building Pixels Perc.	Correct Non-building Pixels Perc.
testarea2 (default)	25	4	2	86.2%	7.4%	81.1%	6.9%	95.8%
testarea2 (modified)	22	7	1	75.8%	4.3%	72.7%	5.1%	96.4%

4.3 TEST AREA 3

4.3.1 TEST AREA 3 PROPERTIES

Third test image is 330x330 wide and was taken from Eskisehir (**Figure 4.21**). The test area3 dataset differs from the previous ones in that image is taken from IKONOS satellite with a resolution of 1m pixel wide. Image has uniform color (red) roof buildings with various orientation, shape and size. Upper part of the image is open land and field. Remaining parts include buildings, some trees and road network. Shadow cast direction is up-right.

After applying the algorithm to the test area3 (**Figure 4.21**), hsv segmentation (**Figure 4.22**), shadow detection (**Figure 4.23**), edge detection (**Figure 4.24**), line generation (**Figure 4.25**), line merging (**Figure 4.26**), generated hypotheses (**Figure 4.27**) and verified hypothesis (**Figure 4.28**) figures are obtained.

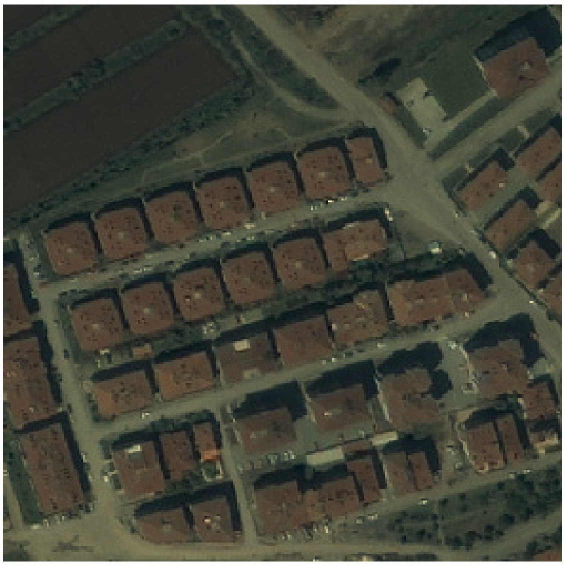

Figure 4.21 Original Image of the Test Area3

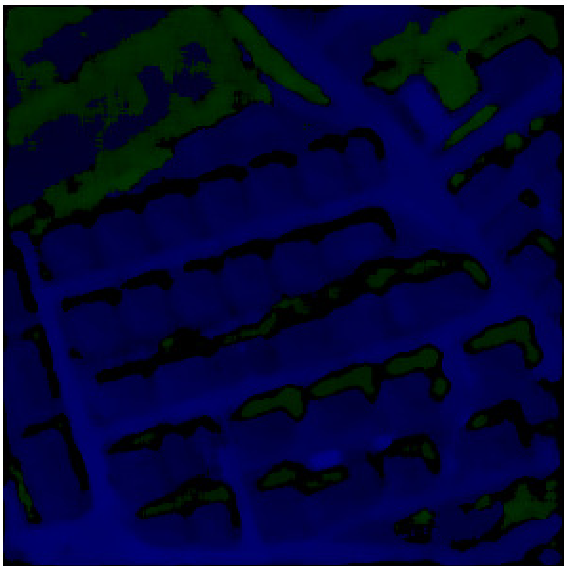

Figure 4.22 HSV Color Segmented Image of Test Area3

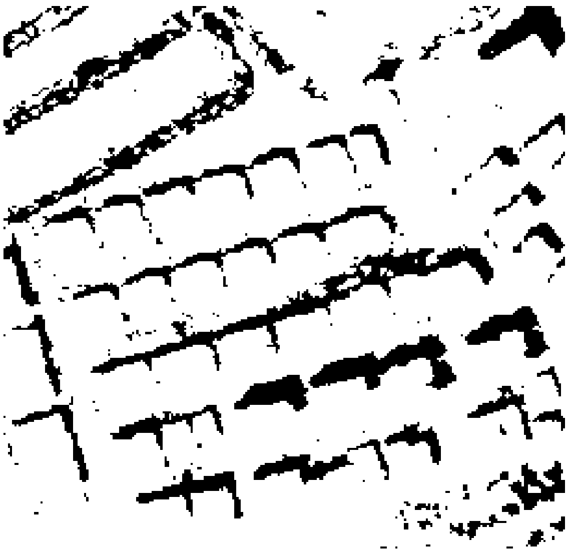

Figure 4.23 Shadow Image of the Test Area3

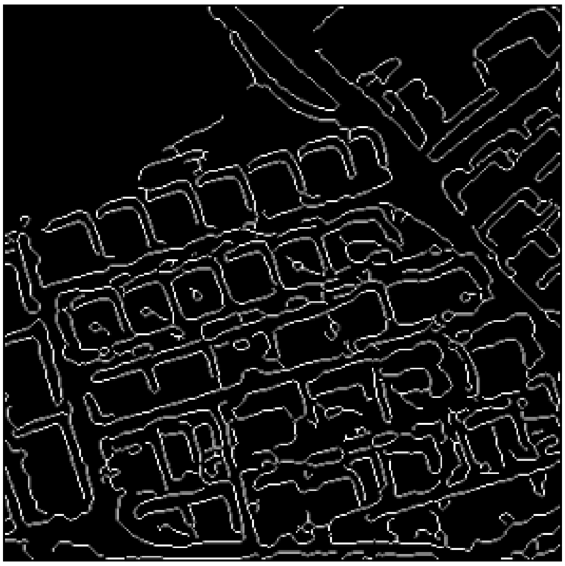

Figure 4.24 Edge Detected Image of Test Area3

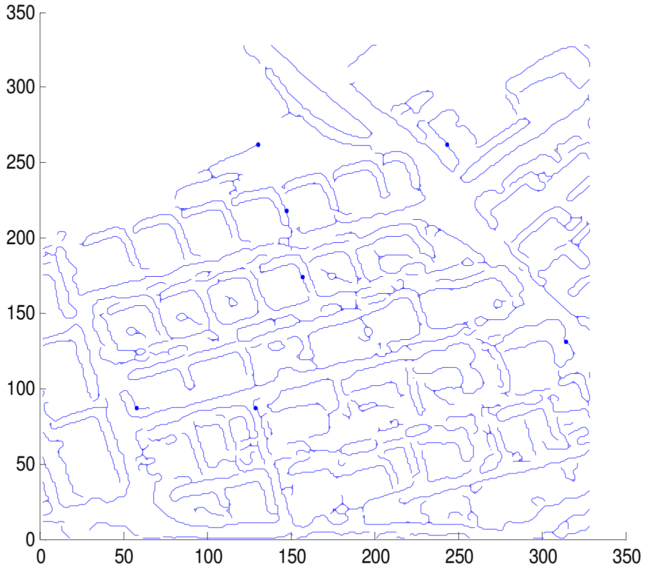

Figure 4.25 Hough Transform Applied to Test Area3

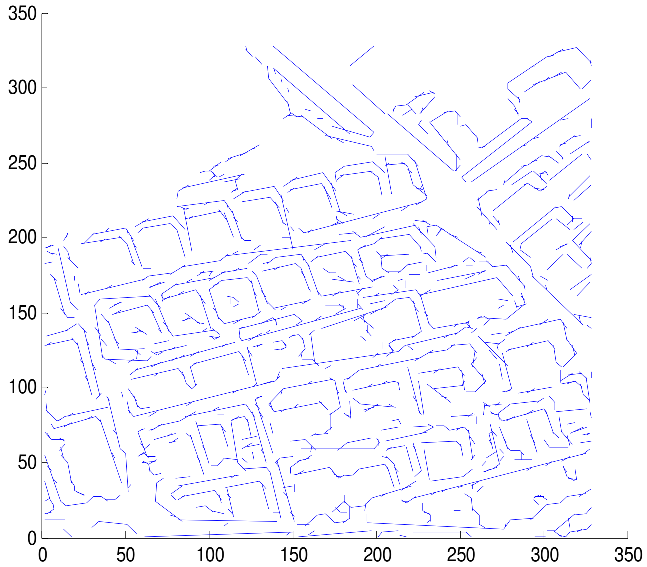

Figure 4.26 Test Area3 after Line Segments are Merged and Removed

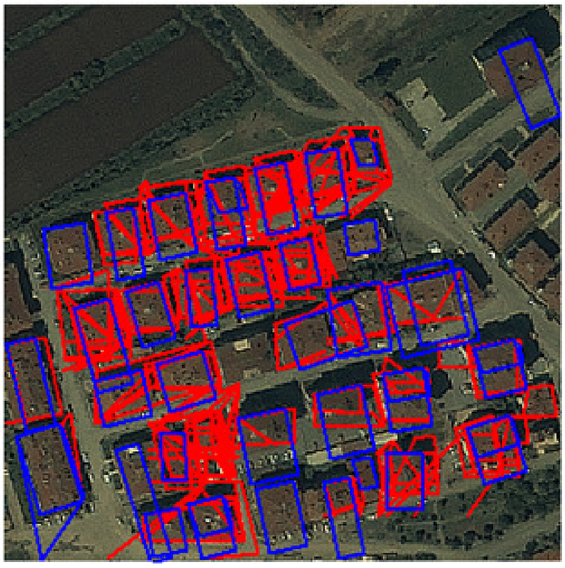

Figure 4.27 Generated Hypotheses for Test Area3

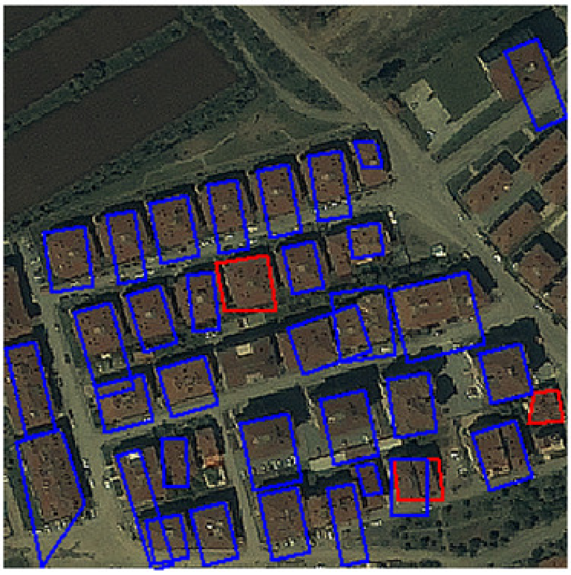

Figure 4.28 Verified Hypotheses for Test Area3

4.3.2 TEST AREA 3 RESULTS AND EVALUATION

The building detection performance for the test area3 image is high; however the generated boundaries (blue and red colored) are not one to one consistent with the buildings' contours. This is due to the hsv-color segmentation image has segments with close pixel values between buildings and road networks, so that buildings' boundaries mostly extend to the road networks. This situation can be avoided by adjusting cluster_color_diff parameter value accordingly (see section 3.9).

Since the image is taken from IKONOS satellite, it has a higher resolution and a better quality than the previous two test images. This leads to a better shadow detection and thus, very few false positives are generated. Besides, hsv-color segmentation is well benefited from high resolution image and resulted in accurate and uniform colored clusters for the buildings.

For this image, 35 buildings are successfully detected, no false positive is generated and 5 buildings are missed. 2 buildings are missed due to the minimum building length threshold is not satisfied (min_line_length) , 3 buildings are missed due to the orientation.

4.3.3 RUN-TIME EVALUATION

The run-time of the algorithm for the test area3 shown in **Figure 4.21** was approximately 260 seconds. Color segmentation and shadow extraction take 29 and 0.56 seconds respectively. Edge and line detection take 172 seconds. Hypothesis formation and verification required 39 and 17 seconds, respectively. The runtimes grow approximately exponentially with the size of the image for scenes of similar complexity.

4.3.4 FEATURES EVALUATION

The features for the test area3 image will be more or less equal to the test area1 since the image sizes and the scene structure are close to each other. These numbers are given in **Table 4.10**.

Table 4.10 Test Area3 Features

	# of edge pixels extracted	# of lines generated
Testarea3	8569	4684

4.3.5 DETECTION EVALUATION

The same measures with the ones used in the test area1 and test area2, are used to evaluate the performance of the algorithm applied to the test area3. Similarly, the statement which defines; a building is considered to be detected if a part of the building is detected by the system is valid. The description of the detected building may not necessarily be correct. The results for the test area3 are given in **Table 4.11**.

Table 4.11 Results for the Test Area3

	Tp	Tn	Fp	Detection Percentage	Branch Factor	Correct Building Pixels Per.	Incorrect Building Pixels Perc.	Correct Nonbuilding Pixels Perc.
ta3	35	5	0	87.5%	0%	78.8%	9.2%	93.6%

4.3.6 TEST AREA 3 PARAMETER EVALUATION

In this test case, parameters for various stages of the algorithm are the same as the test case1 but shadow mask is updated (**Table 4.12**). Since the image is taken from IKONOS satellite, it is high resolution. And thus, HSV components are more precisely represented which actually makes it easier to perform the shadow mask.

Other advantage of using high resolution image is hsv-color segmentation gives uniform color segments for the buildings.

In this test case, the boundaries of the buildings are plotted larger than their original sizes. In order to improve this situation, cluster_color_diff parameter is decreased, and the algorithm is applied again.

Table 4.12 Parameter Values for the Test Area3

Parameter	Value
min_line_length	image_width/35
edge_Th_high	0.25
edge_Th_low	0.06
cluster_color_diff	10

This parameter is decreased to 4 from 10. The expected result of this change is; plotted boundaries are more consistent with the real boundaries.

cluster_color_diff parameter’s default value is set as 10, which is optimum for most of the satellite/aerial images. This parameter is not effective on the algorithm’s detection performance unless it is set as lower than 3 or larger than 20. Decreasing the value of this parameter below 3 may cause to miss some of the buildings since with this parameter value, some small-sized buildings produced according to HSV-color segmentation information are eliminated because of violating min_line_length requirement. And increasing the value above 20 may lead to produce very large boundaries that may overlap or occupy other buildings boundaries so the small boundaries are eliminated due to hypothesis verification rules which in the end lead to the degradation in the detection performance.

The value of cluster_color_diff is effective on the shape of the boundary of the extracted building. That is, the larger the value of this parameter, the bigger the boundary plotted at the output, and similarly the lower the value of cluster_color_diff, the smaller the boundary plotted. If there is no requirement to the boundary of the detected buildings', this parameter value can be left as default.

The results of this change are given in **Figure 4.29** which shows the verified buildings figure.

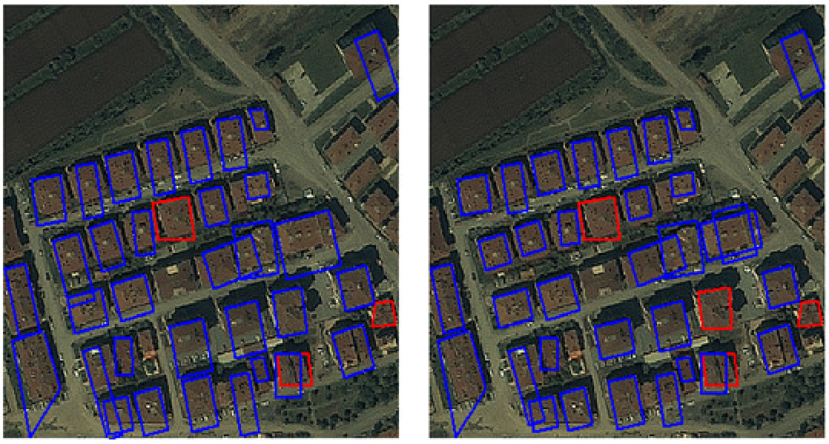

Figure 4.29 Verified Buildings Results

Finally, performance evaluation is given in **Table 4.13** to show the affect of the cluster_color_diff parameter change.

Table 4.13 Results for the Modified cluster_color_diff Value for Performance Evaluation

	Tp	Tn	Fp	Det. Perc.	Br. Ft.	Correct Building Pixels Perc.	Incorrect Building Pixels Perc.	Correct Non-building Pixels Perc.
testarea3 (default)	35	5	0	87.5%	0%	78.8%	9.2%	93.6%
testarea3 (modified)	35	5	0	87.5%	0%	78.8%	5.3%	96.4%

4.4 CHOICE OF PARAMETERS

The implemented algorithm needs to make several decisions based on incomplete evidence at various stages in the selection and verification processes. Evidence in various categories, such as from shadows and HSV-color segmented areas, is combined to make decisions on keeping or discarding hypotheses. Due to the complexity of the processes involved and lack of formal models for the contents of an image, it is difficult to find theoretically optimal solutions. Used procedure for such decision making is based on simplicity and intuitive judgments.

In the image preprocessing part, 3 filters are considered to improve the edge detection performance, which are anisotropic nonlinear diffusion, bilateral and sharpening filters. Anisotropic nonlinear diffusion filter proposed by Perona and Malik [53] basically smoothes the image while preserving the edges. However, this filter makes the edges of close buildings to disappear in a complex environment with buildings very close to each other which is inconvenient for the proposed algorithm (**Figure 4.30**).

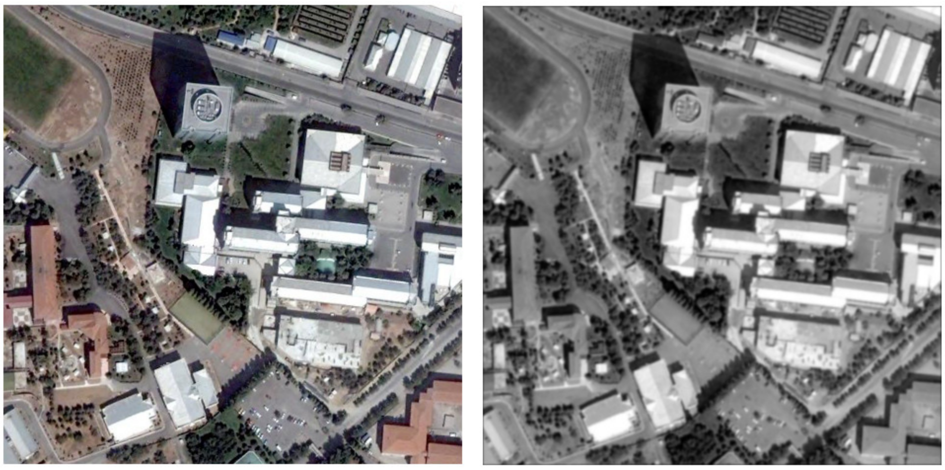

Figure 4.30 Original and Anisotropic Nonlinear Filter Applied Image

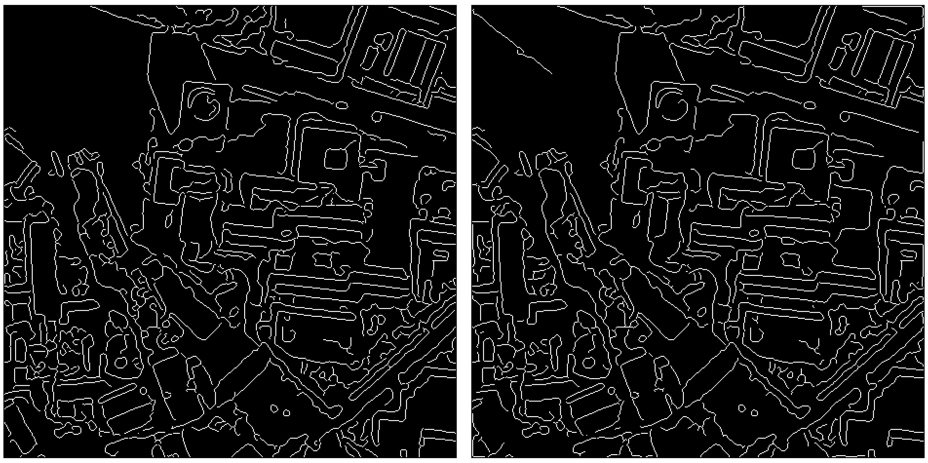

Figure 4.31 Edges Extracted from Original and Anisotropic Nonlinear Filtered Images

Bilateral filter proposed by Tomasi and Manduchi [54] basically takes into consideration the pixels' intensity values while averaging with its neighboring pixel values which preserves the sharp edges. Although this filter removes unnecessary details in the images which are grass or parking lot details and chimney discontinuities, it loses the edges of the buildings where buildings and their surroundings have close color values (**Figure 4.32**).

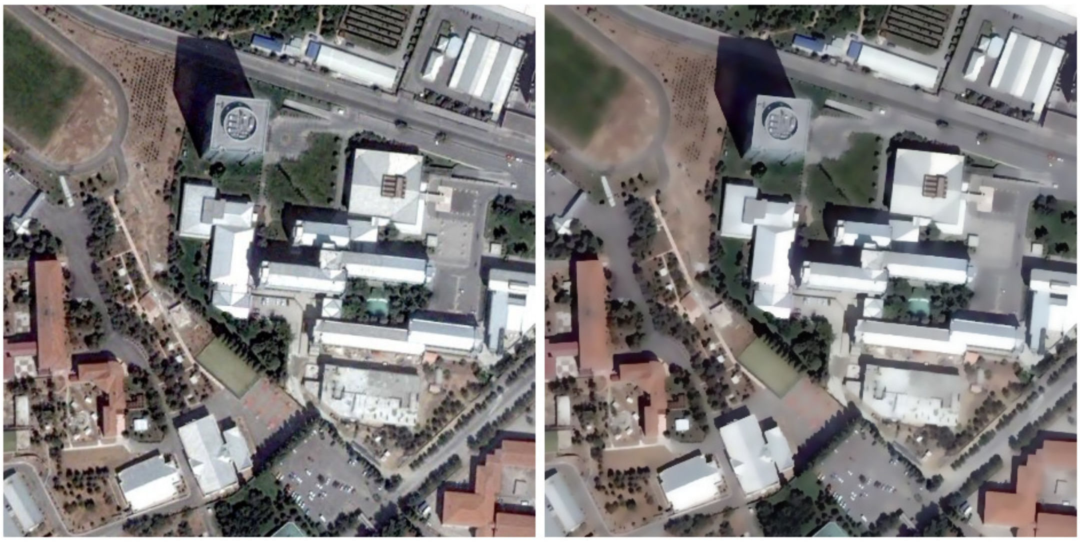

Figure 4.32 Original and Bilateral Filter Applied Images

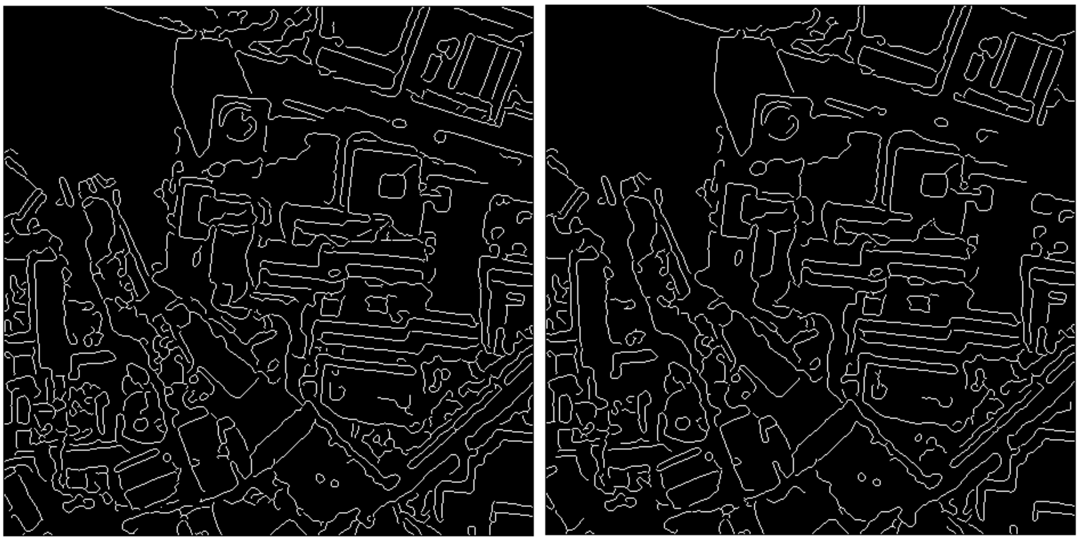

Figure 4.33 Edges Extracted from Original and Bilateral Filtered Images

The sharpening filter is basically a smoothed version of the original image that is inversely applied to the image. After applying the sharpening filter, the edges of the buildings are both preserved and strengthened which improves the edge detection performance of the proposed algorithm. Therefore, in this study sharpening filter is selected as the preprocessing filter (**Figure 4.34**).

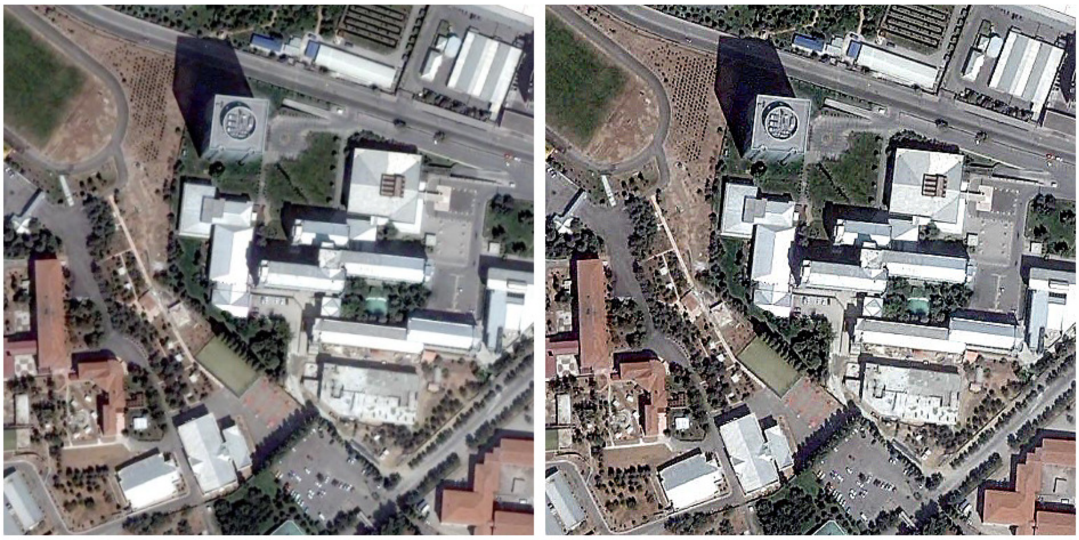

Figure 4.34 Original and Sharpening Filter Applied Images

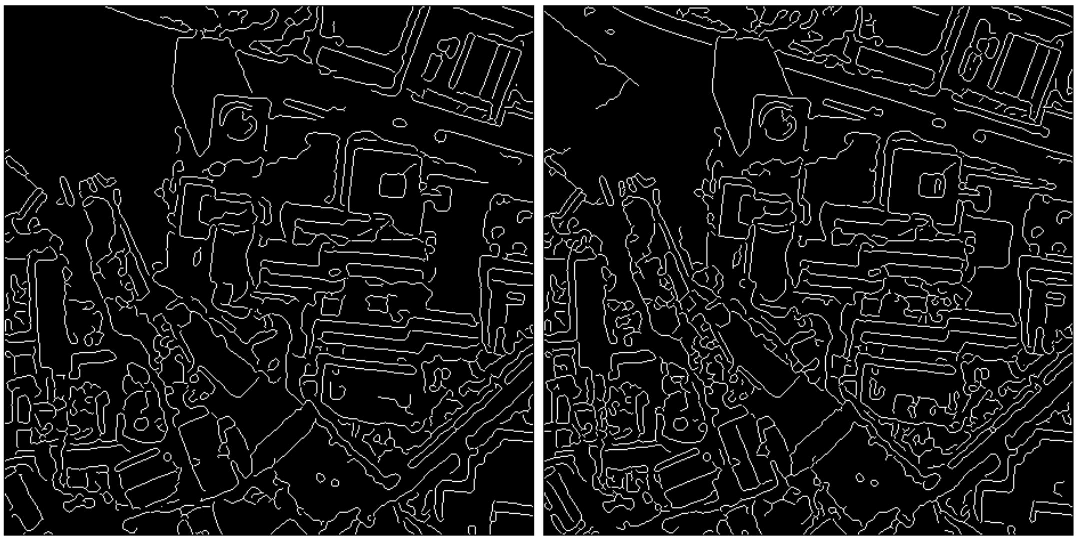

Figure 4.35 Edges Extracted from Original and Sharpening Filtered Images

The results of the verified building detection images after using each of these filters are shown in **Figure 4.36-Figure 4.38**. Moreover, in **Table 4.14** extracted features data are given to compare the filters performance.

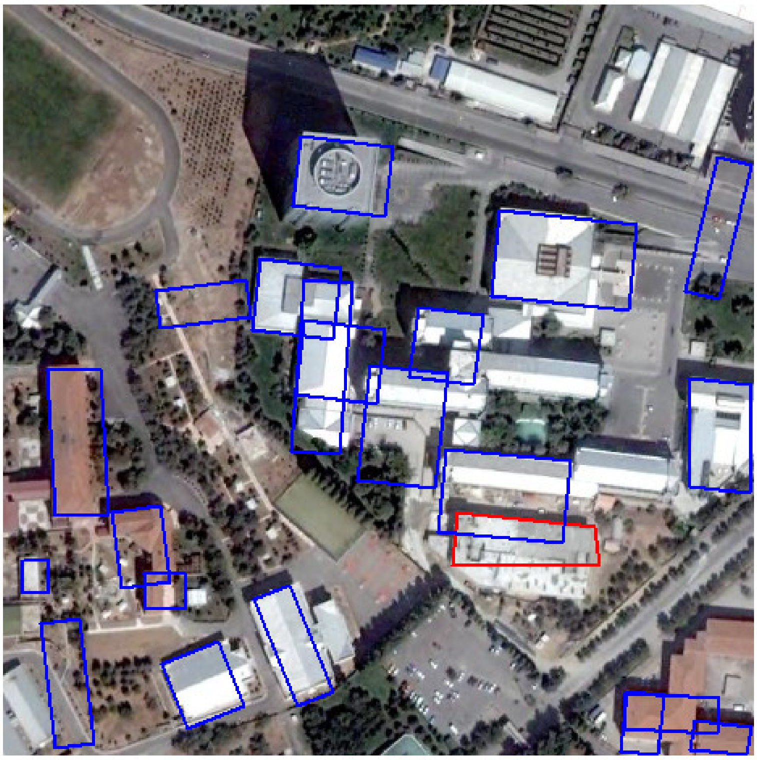

Figure 4.36 Verified Building Results using Anisotropic Nonlinear Diffusion Filter

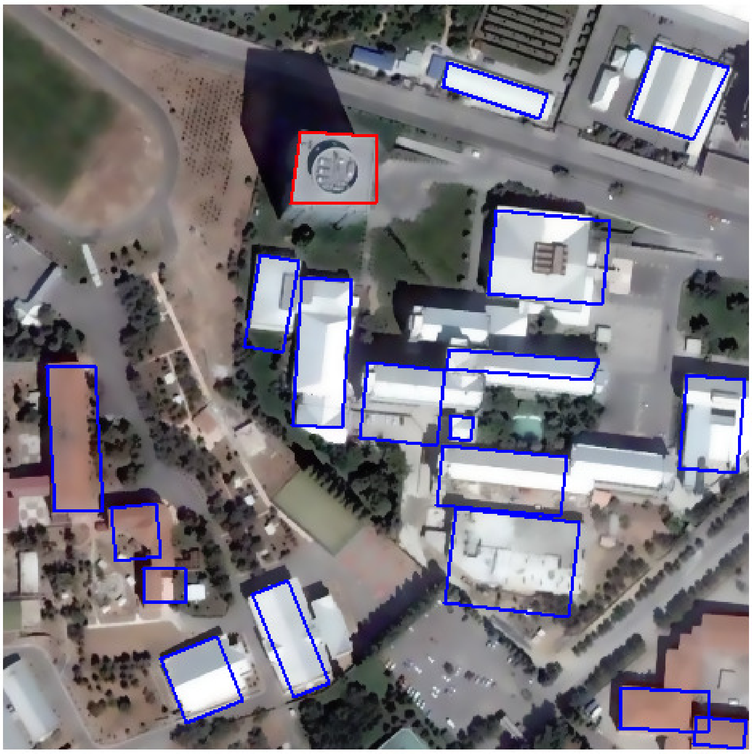

Figure 4.37 Verified Building Results using Bilateral Filter

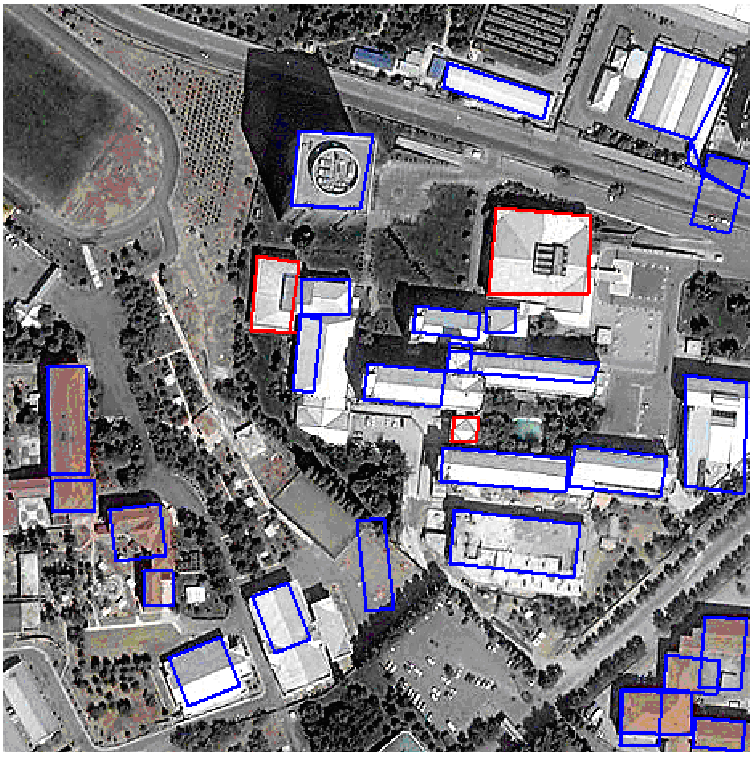

Figure 4.38 Verified Building Results using Sharpening Filter

Table 4.14 Results for the 3 Filters Considered

Testarea2	# of edge pixels extracted	# of lines generated	Tp	Tn	Fp	Det. Perc.	Br. Ft
Anisotropic nonlinear filter	14709	6974	18	11	3	62%	14%
Bilateral filter	13085	6068	20	9	0	68.9%	0%
Sharpening Filter	17651	8759	25	4	2	86.2%	7.4%

In the color segmentation stage of the proposed algorithm rbg, hsv and lab spaces are considered. That is, image is segmented in red, green, blue, hue, saturation and value spaces individually. And also rgb and hsv represented images are segmented in lab spaces. Due to illuminations and reflections, individual segmentations and rgb image represented in lab space have given poor results (**Figure 4.39-Figure 4.44**). In other words, buildings occupy two or more than two clusters. Besides, the well-known mean-shift segmentation has produced limited output which is some buildings occupy two segments (**Figure 4.45**). HSV represented image segmented in lab domain however, yields buildings to exist in only of the clusters which is required in the hypotheses verification part of the proposed algorithm.

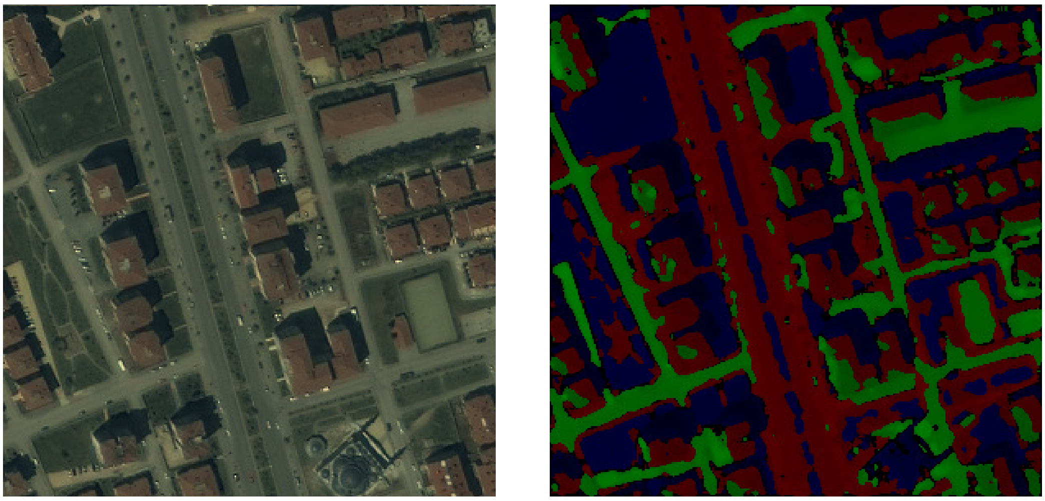

Figure 4.39 Original Image and Segmentation Applied on "red" Space Image

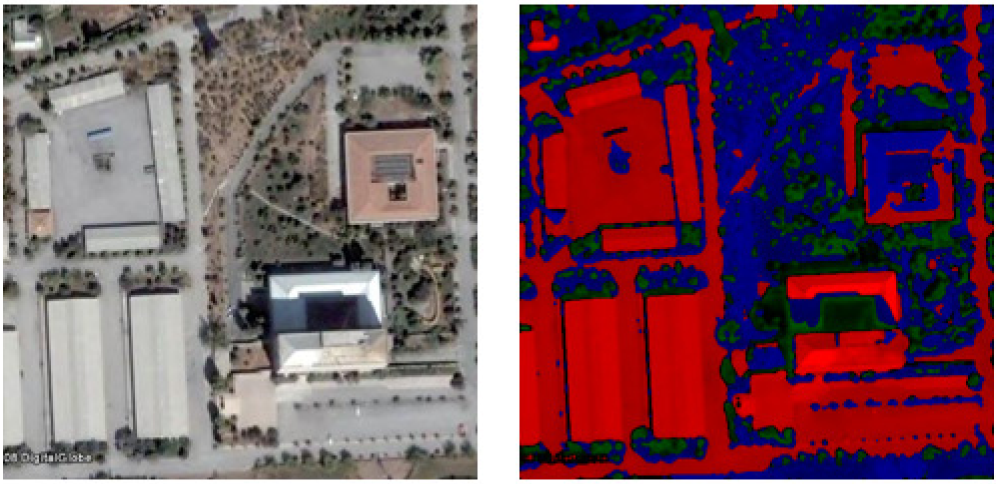

Figure 4.40 Original Image and Segmentation Applied on “green” Space Image

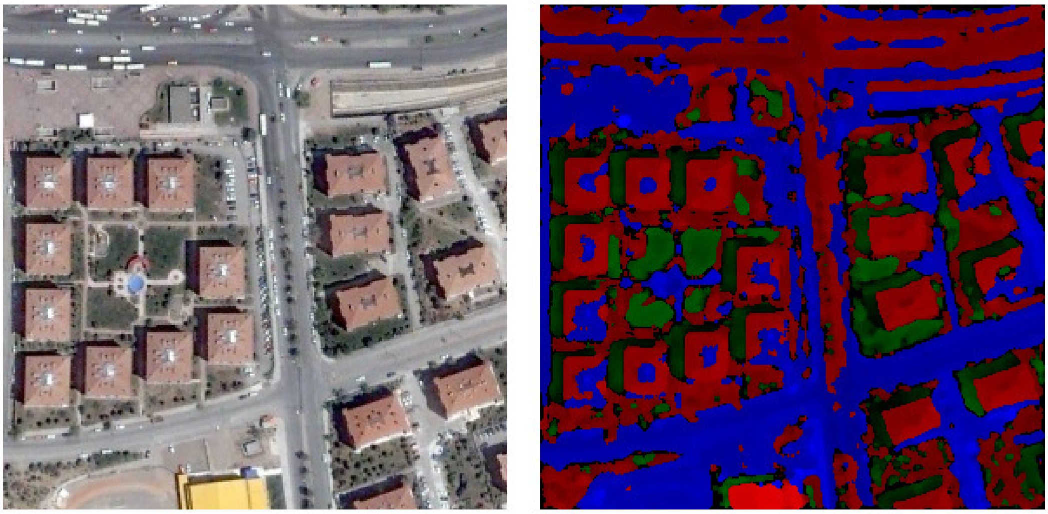

Figure 4.41 Original Image and Segmentation Applied on “blue” Space Image

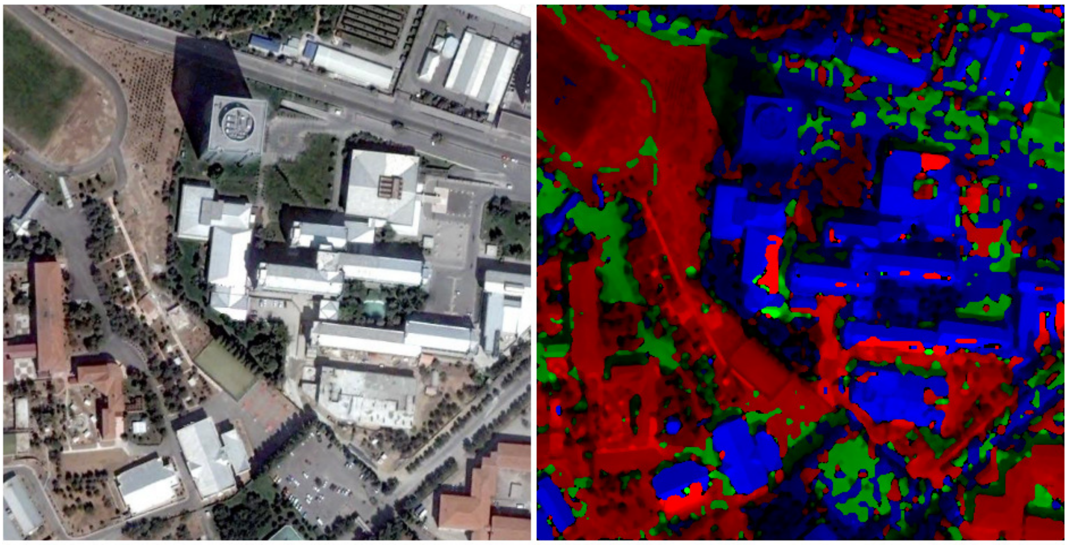

Figure 4.42 Original Image and Segmentation Applied on “hue” Space Image

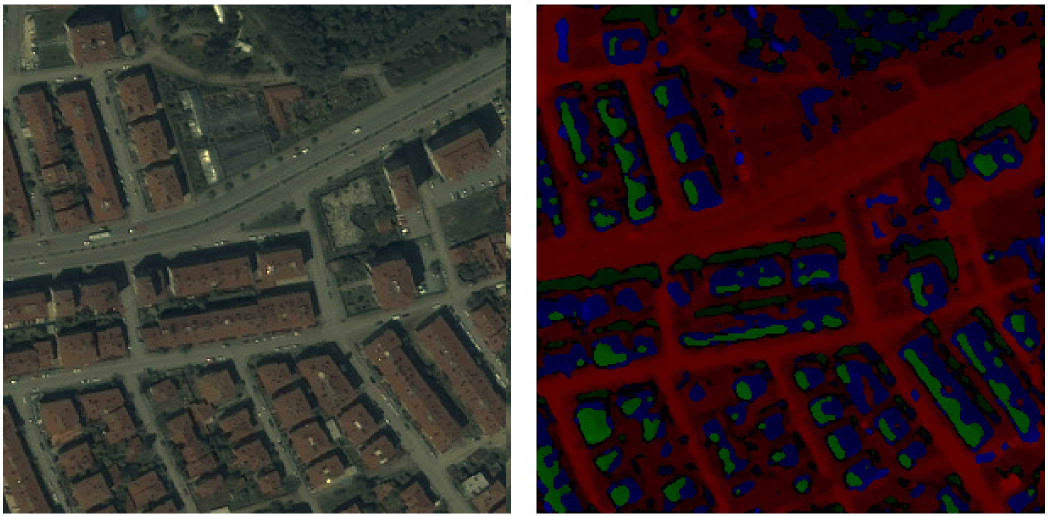

Figure 4.43 Original Image and Segmentation Applied on “saturation” Space Image

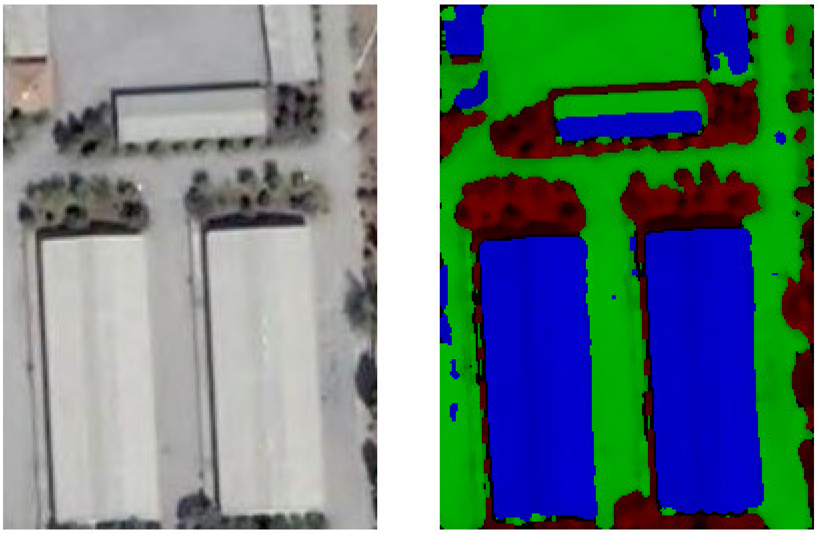

Figure 4.44 Original Image and Segmentation Applied on "value" Space Image

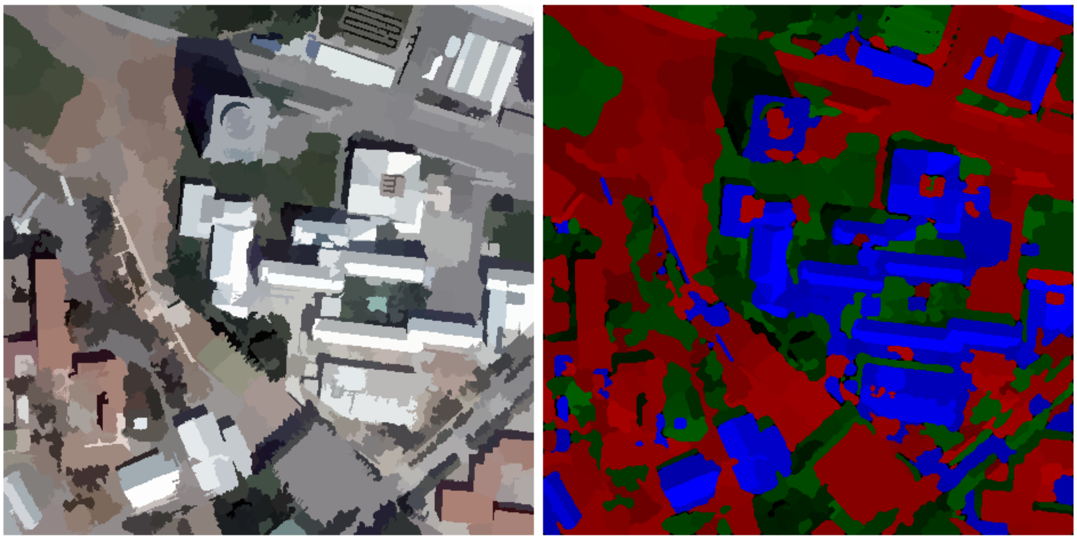

Figure 4.45 Mean-shift Segmentation Image and its k-means Applied Form

The results of the verified building detection images after segmentation is carried out on each of these spaces are shown in **Figure4.46-Figure4.49**. Moreover, in **Table 4.15** the extracted features are given to compare the filters performance.

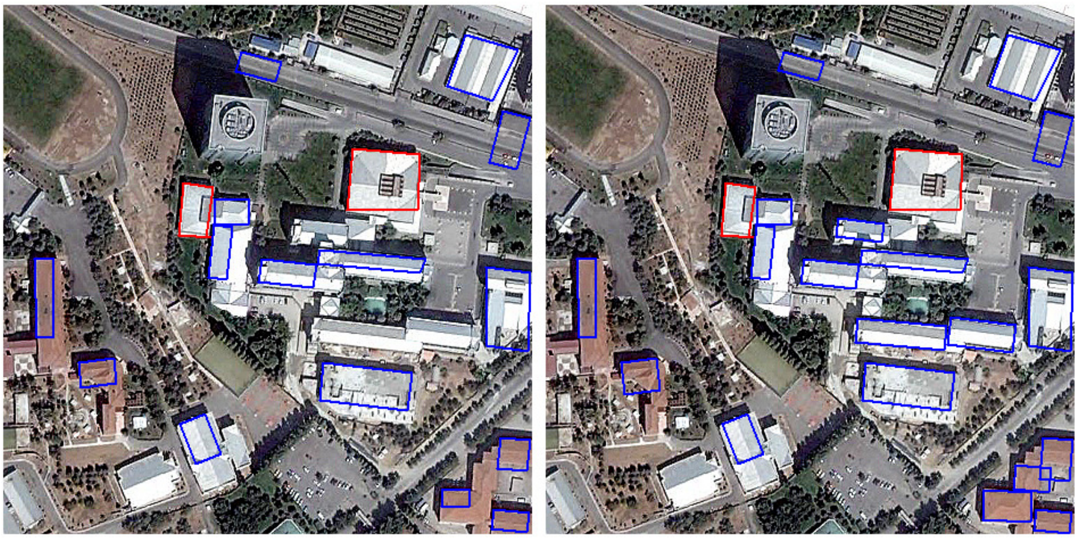

Figure 4.46 Verified Building Results after Segmenting on "red" and "green" Spaces Respectively

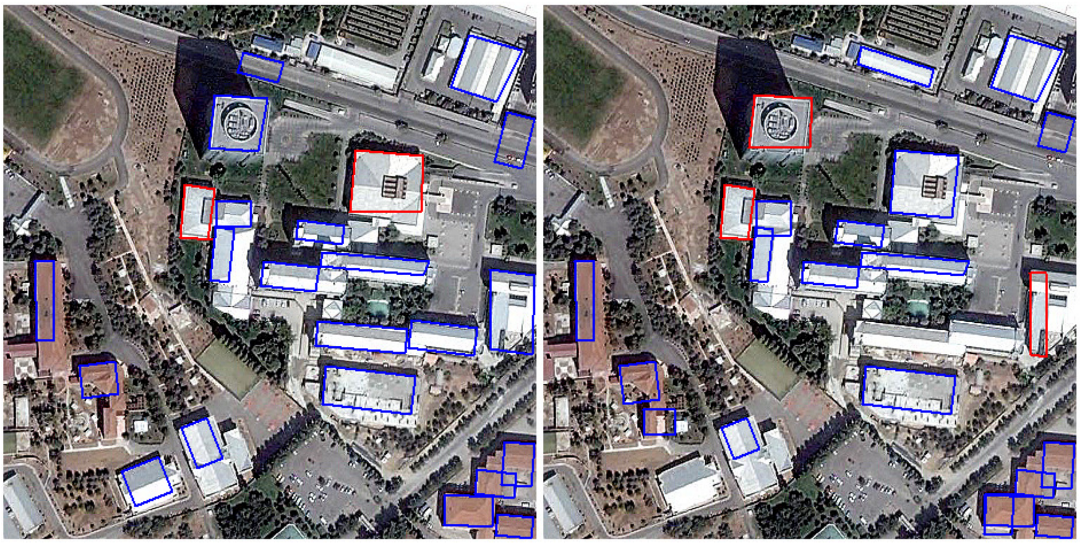

Figure 4.47 Verified Building Results after Segmenting on "blue" and "hue" Spaces Respectively

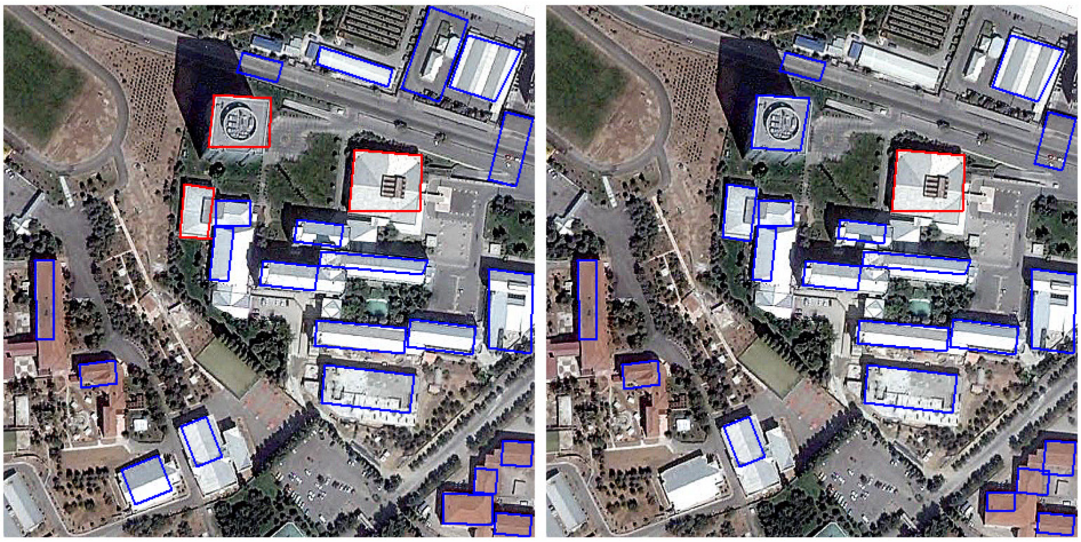

Figure 4.48 Verified Building Results after Segmenting on "saturation" and "value" Spaces Respectively

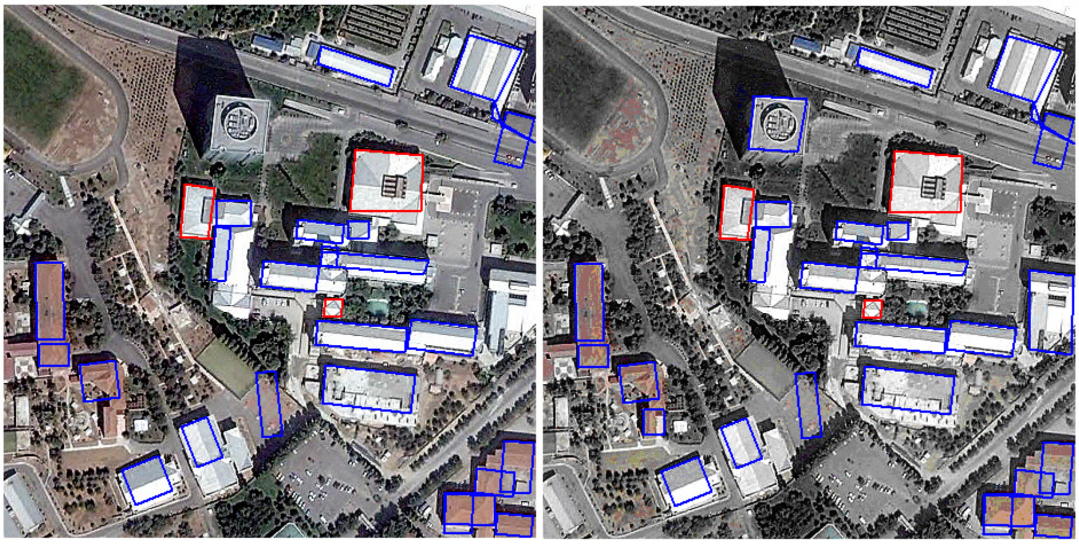

Figure 4.49 Verified Building Results after mean-shift Segmentation and hsv Segmentation is Performed Respectively

Table 4.15 Results for the Segmentations Performed on Different Spaces

Testarea2	Tp	Tn	Fp	Detection. Percantage	Branch. Factor
“red” space	15	14	2	51.7%	11.7%
“green” space	18	11	1	62%	5%
“blue” space	22	7	2	75.8%	8%
“hue” space	18	11	1	62%	5%
“saturation” space	23	6	3	79.3%	11%
“value” space	20	9	2	68.9%	9%
Mean-shift applied on “rgb” space	23	6	2	79.3%	8%
Hsv space	25	4	2	86.2%	7.4%

The verified buildings' boundaries are not one-to-one consistent with the real buildings due to poor performance of the line generation stage. In order to overcome this problem, region growing is considered to make the boundaries more consistent. However, region growing algorithms produce worse boundaries for the buildings that have close color pixel values with their surroundings and buildings with oblique rooftops. Therefore, region growing is not implemented in this study (**Figure 4.50**).

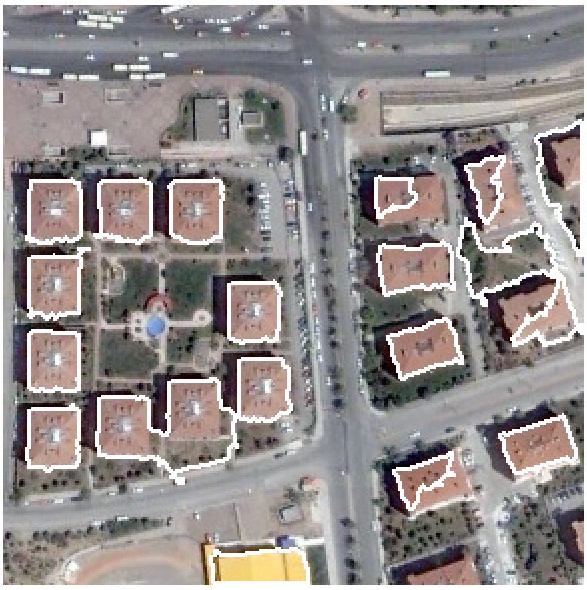

Figure 4.50 Region Growing Applied Image

In the proposed algorithm, to reduce the number of false hypotheses, edge verification for the verified building hypotheses is considered. In other words, each line segment of the verified building hypothesis is investigated whether there are edge pixels present adjacent or not. This verification works fine for the images with buildings away from each other, but for the images with buildings are close to each other, some of the true positive buildings are eliminated. Therefore, this verification stage is not included in the proposed algorithm.

4.5 PERFORMANCE RESULTS

Algorithm is applied to 9 images. First 2 images are the Google Earth images and the remaining 7 of them are IKONOS satellite images showing Eskisehir city. Table 4.14 gives the performance results of the algorithm for each image as well as the image sizes and, runtimes. In this table, first 3 rows correspond to the three test areas which are investigated in detail in section 4. For the last 6 rows, the building detected images are given in **Figure 4.51** to **Figure 4.56**.

Table 4.16 Performance Evaluation of the Algorithm

	Tp	Tn	Fp	Det. Perc.	Br. Ft.	Run time	Image Size	#of extracted edge pixels& lines
ta1	19	0	1	100%	5%	189	290x290	(8710,3704)
ta2	25	4	2	86.2%	7.4%	750	465x465	(17651,8759)
ta3	35	5	0	87.5%	0%	260	330x330	(8569,4684)
ta4	53	2	5	96.3%	8.6%	945	364x364	(15608,8437)
ta5	72	7	1	91.1%	1%	1340	400x400	(17358,10173)
ta6	137	30	5	82.0%	3%	1405	400x400	(19573,9267)
ta7	24	6	1	80%	4%	230	300x300	(6434,3906)
ta8	34	4	1	89.4%	2%	475	300x300	(10502,5820)
ta9	132	28	6	82.5%	4%	2717	500x500	(24856,14300)

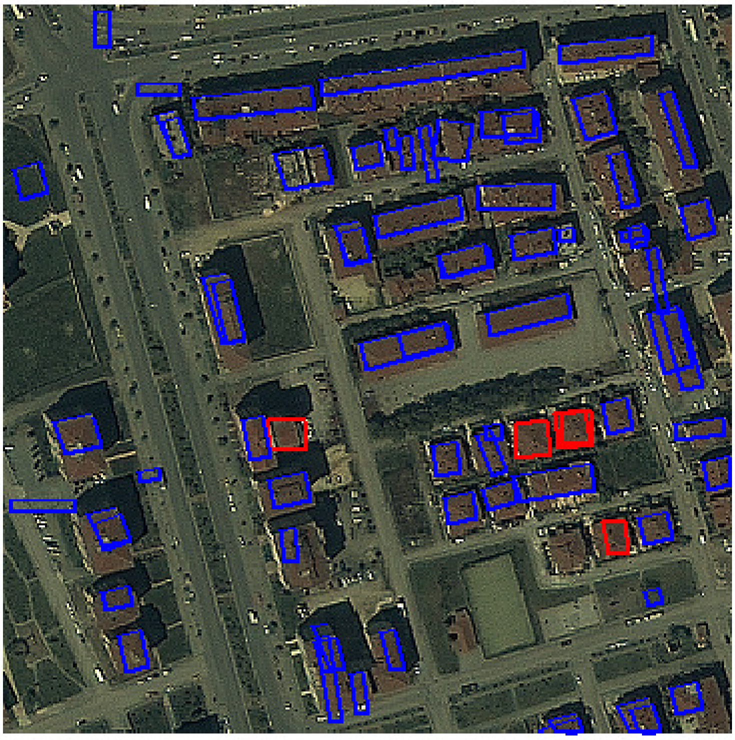

Figure 4.51 Verified Buildings Result for Test Area4

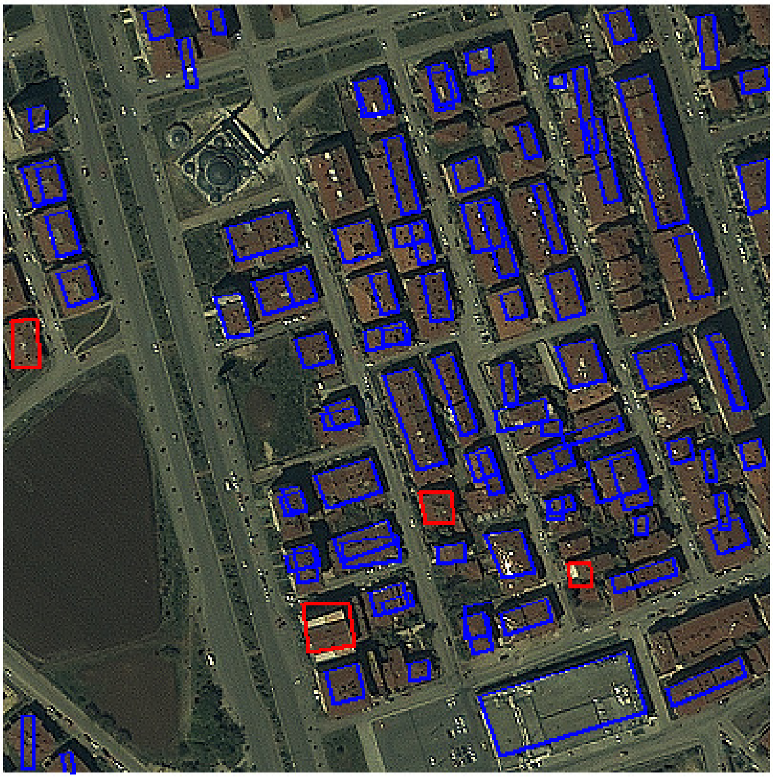

Figure 4.52 Verified Buildings Result for Test Area5

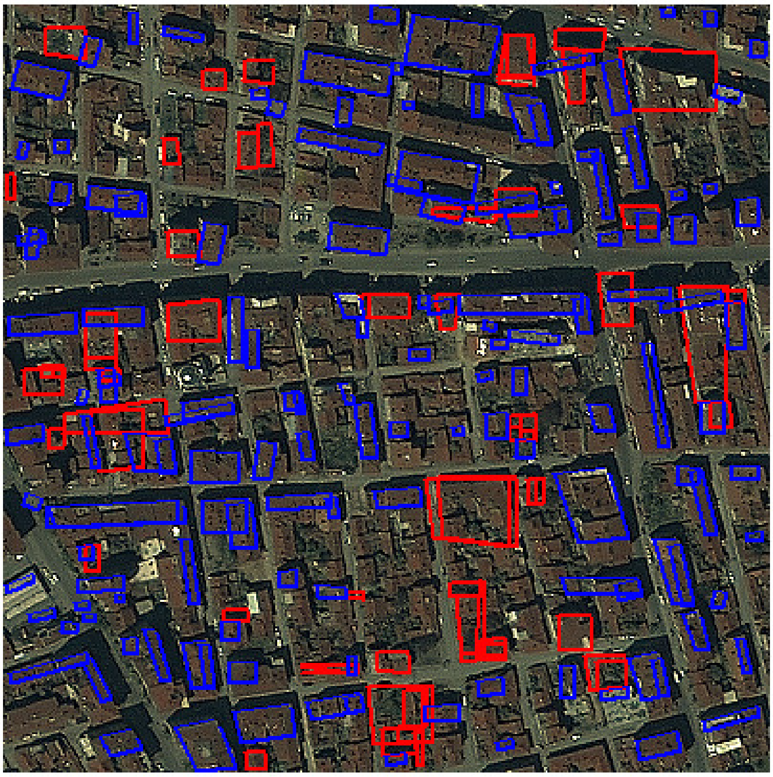

Figure 4.53 Verified Buildings Result for Test Area6

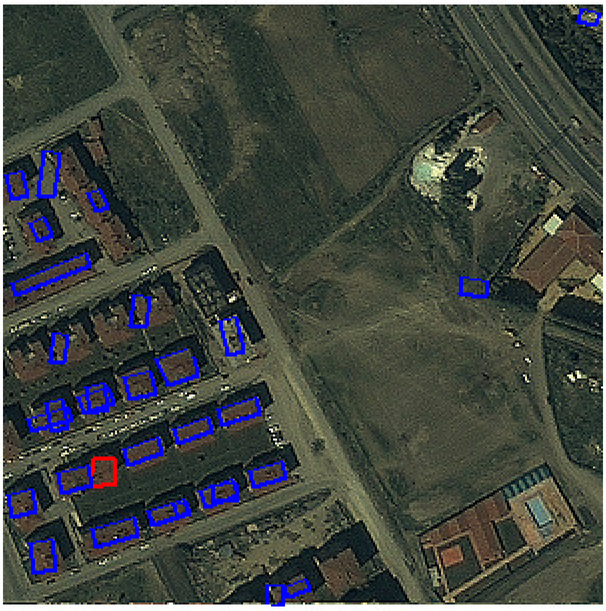

Figure 4.54 Verified Buildings Result for Test Area7

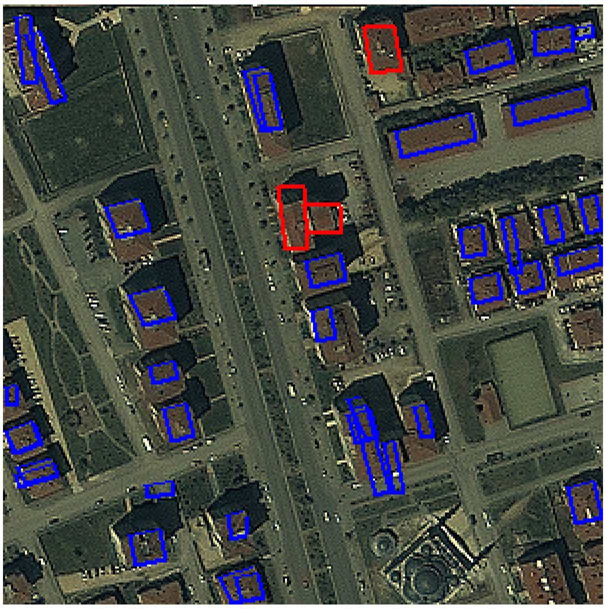

Figure 4.55 Verified Buildings Result for Test Area8

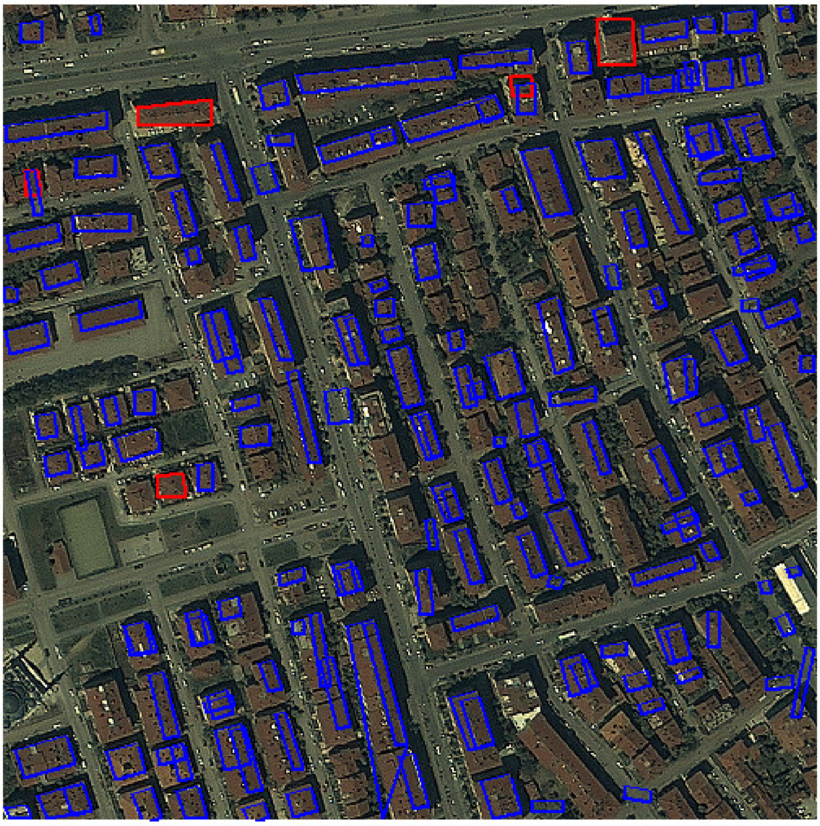

Figure 4.56 Verified Buildings Result for Test Area9

CHAPTER 5

CONCLUSIONS

5.1 SUMMARY

In this study, building detection algorithm from aerial or satellite imagery is proposed and implemented. Proposed algorithm is based on hypothesis generation and verification for rectilinear buildings. HSV-color segmentation and shadow information of the image is used in this thesis to verify the building hypothesis which is different from the previous studies.

A cascaded system is implemented in this thesis. Firstly, the satellite/aerial image is filtered to remove noise and make the edges stronger so that most of them can be detected by edge detection. Next, edge detection is performed using Canny edge detection algorithm. The parameters used in the Canny edge detection stage is critical since these parameters determine the number of points extracted from the image some of which constitute the buildings' parts. These parameters are set to optimum values so as to find maximum building parts and minimum noise and/or spurious points. Then, these extracted edge points are used to generate line segments by using Hough Transform algorithm. In order to correctly generate these line segments, Hough Transform's parameters are optimized so that the algorithm does not produce spurious/false line segments. Next, building hypotheses are formed from these lines which obey the rectilinear building models. Finally, the hypotheses are verified using both the hsv-color segmentation and the shadow information extracted from the satellite/aerial image which eliminates majority of the false positive hypotheses. Here in this thesis, the use of both the HSV-color segmentation and the shadow information to verify these hypotheses is developed which has not been applied in the literature yet.

The results of the implemented algorithm are satisfactory for both urban and sparse environments. Building detection rate is above %80 for most of the images.

5.2 DISCUSSION

In this thesis, a cascaded algorithm is implemented to detect the buildings in a satellite/aerial image. In other words, the output of a stage in the algorithm is the input to its successor which means that overall performance of the algorithm is actually determined by the poorest stage. First stage in the algorithm is the satellite/aerial image, the quality of the image is important since HSV-color segmentation, shadow detection and edge detection stages take this image as an input. Therefore, these stages' performance depends on the image quality. Shadow detection is carried out by using a simple mask to the HSV representation of the image; this mask functions well for most of the test cases but may produce poor performance images having dark-colored rooftop buildings unless images are not taken from satellite. Then, edge detection stage plays a key role in the algorithm. The parameters (low threshold, high threshold, and sigma) determine the success rate of this stage. The building detection rate can be improved or degraded by adjusting these parameters, such as by lowering the threshold value more buildings can be detected in the image but a lower threshold also catches more noise in the scene which will result in false positives and an increased runtime of the algorithm. Another important stage in the algorithm is the line merging and removing stage. The threshold parameters to merge lines and remove lines can be adjusted to perform better for one image, but these values may not perform well for another image. In other words, it is difficult to find theoretically optimal solutions satisfying for all the images. Hypothesis generation and verification stages' parameters are set according to the possible building sizes which are defined based on the information of the altitude image is taken or image's meter-pixel resolution. In this study, images are either taken at 1000m altitude or have 1m-pixel resolution. Therefore, the parameters for the algorithm is set so as to find as much as buildings and as low as nonbuildings at the output.

5.3 FUTURE WORK

A first step forward from this work is to make use of other data sources such as range data (LiDAR) or stereo images which will provide the height data of the buildings so that 3D representations of these structures can be modeled. These data sources also eliminate the generated false positive building hypotheses.

Verification using the shadow information part of the implemented system can be improved by using a better shadow detection algorithm which will greatly reduce the false positive building hypotheses.

In this system, the shadow cast direction is needed for the system to correctly verify the hypotheses, but by implementing the necessary software codes this information can be extracted from the image which will reduce the user interaction.

The color segmentation part in this algorithm is implemented to produce 3 distinct regions, however, this number can be increased to 4 or 5 regions which will provide the system to model the buildings more accurately. But, increasing the number of regions may cause to miss some of the buildings, in order to prevent this loss, verification part of the algorithm should be updated accordingly.

This building detection system takes the image as a whole and processes all of the algorithms on this single image which is computationally inefficient. To improve the runtime of the system, the input image may be divided into small sub images and intra algorithms (edge detection, line generation) be applied to theses sub images. After these intra algorithms are finished, the results of all the sub images should be merged correctly. This divide and process method will greatly reduce the runtime of the algorithm since most of the time is consumed during edge detection and line generation parts of the algorithm.

The approach presented here can be easily generalized for more complex, but specific, shapes (such as for regular polygon shaped roofs), but would require major modifications to handle more general cases. The system is also limited in its ability to handle buildings which are rectilinear but highly complex, such as consisting of many wings, or cases where the buildings are very close to each other. Modeling of more complex buildings in more complex surrounds remains a topic for future research.

REFERENCES

[1] S. Levitt. and F. Aghdasi. "An investigation into the use of wavelets and scaling for the extraction of buildings in aerial images", Proceedings of the 1998 South African Symposium on Communications and Signal Processing, 1998. COMSIG '98, pp. 133-138

[2] T. Guo, and Y. Yasuoka, "Snake-based approach for building extraction from high-resolution satellite images and height data in urban areas", Proceedings of 23rd Asian Conference on Remote Sensing, 2002

[3] Wei Liu; Prinet V. "Building detection from high-resolution satellite image using probability model" Geoscience and Remote Sensing Symposium, 2005 IGARSS apos;05 volume 6 issue pp. 2888-2891

[4] Z. Lari and H. Ebadi, "Automated building from high-resolution satellite imagery using spectral and structural information based on artificial neural networks" Conference on Information Extraction from SAR and Optical Data, with Emphasis on Developing Countries, 2007

[5] A. Huertas and R. Nevatia, "Detecting Buildings in Aerial Images," Computer Vision, Graphics and Image Processing, vol. 41, no. 2, pp. 131-152, Feb. 1988.

[6] R. Irving and D. McKeown, "Methods for Exploiting the Relationship between Buildings and Their Shadows in Aerial Imagery," IEEE Trans. Systems, Man, and Cybernetics, vol. 19, no. 6, pp. 1564-1575, Nov./Dec. 1989.

[7] J. Shufelt and D. McKeown, “Fusion of Monocular Cues to Detect Man-Made Structures in Aerial Imagery,” Computer Vision, Graphics, and Image Processing, vol. 57, no. 3, pp. 307-330, May 1993.

[8] C.A. Lin and R. Nevatia, “Building Detection and Description from a Single Intensity Image,” Computer Vision and Image Understanding, vol. 72, no. 2, pp. 101-121, Nov. 1998.

[9] J.C. McGlone and J. Shufelt, “Projective and Object Space Geometry for Monocular Building Extraction,” Proc. Conf. Computer Vision and Pattern Recognition, pp. 54-61, June 1994.

[10] L. Sahar and A. Krupnik, “Semiautomatic extraction of building outlines from large-scale aerial images," vol. 65, pp. 459-465, Department of Civil Engineering, Technion - Israel Institute of Technology, Haifa, 32000 Israel, Photogrammetric Engineering & Remote Sensing, April 1999.

[11] C. Lin, A. Huertas, and R. Nevatia, “Detection of buildings from monocular images," Institute for Robotics and Intelligent Systems University of Southern California Los Angeles, California 90089-0273, U.S.A., 1995.

[12] Special Issue on Automatic Building Extraction from Aerial Images, Computer Vision and Image Understanding, A. GruÈn and R. Nevatia, eds., vol. 72, no. 2, Nov. 1998.

[13] M. Roux and D. McKeown, “Feature Matching for Building Extraction from Multiple Views,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 46-53, June 1994.

[14] M. Herman and T. Kanade, “Incremental Reconstruction of 3D Scenes from Multiple, Complex Images,” Artificial Intelligence, vol. 30, no. 3, pp. 289-341, Dec. 1986.

[15] R. Collins, C. Jaynes, Y.Q. Cheng, X. Wang, F. Stolle, E. Riseman, and A. Hanson, “The Ascender System: Automated Site Modeling from Multiple Aerial Images,” Computer Vision and Image Understanding, vol. 72, no. 2, pp. 143-162, Nov. 1998.

[16] P. Fua, “Model-Based Optimization: Accurate and Consistent Site Modeling,” Proc. 18th ISPRS Congress, Comm. III, WG 2, pp. 222- 233, 1996.

[17] O. Faugeras, S. Laveau, L. Robert, G. Csurka, and C. Zeller, “3D Reconstruction of Urban Scenes from Sequences of Images,” Proc. Automatic Extraction of Man-Made Objects from Aerial and Space Images, pp. 145-168, 1995.

[18] A. Fischer, T. Kolbe, F. Lang, A. Cremers, W. Forstner, L. Plumer, and V. Steinhage, “Extracting Buildings from Aerial Images Using Hierarchical Aggregation in 2D and 3D,” Computer Vision and Image Understanding, vol. 72, no. 2, pp. 185-203, Nov. 1998.

[19] O. Henricsson, “The Role of Color Attributes and Similarity Grouping in 3D Building Reconstruction,” Computer Vision and Image Understanding, vol. 72, no. 2, pp. 163-184, Nov. 1998.

[20] N. Haala and M. Hahn, “Data Fusion for the Detection and Reconstruction of Buildings,” Proc. Automatic Extraction of Man- Made Objects from Aerial and Space Images, pp. 211-220, 1995.

[21] E. Baltsavias, S. Mason, and D. Stallmann, “Use of DTMs/DSMs and Orthoimages to Support Building Extraction,” Proc. Automatic Extraction of Man-Made Objects from Aerial and Space Images, pp. 199- 210, 1995.

[22] U. Weidner, “An Approach to Building Extraction from Digital Surface Models,” Proc 18th ISPRS Congress, Comm. III, WG 2 pp. 924-929, 1996.

[23] T. Kim and J. Muller, “Building Extraction and Verification from Spaceborne and Aerial Imagery Using Image Understanding Data Fusion Techniques,” Proc. Automatic Extraction of Man-Made Objects from Aerial and Space Images, pp. 221-230, 1995.

[24] A. Huertas, Z. Kim, and R. Nevatia, “Multisensor Integration for Building Modeling,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 203-210, June 2000.

[25] M. Berthod, L. Gabet, G. Giraudon, and J. Lotti, “High Resolution Stereo for the Detection of Buildings,” Proc. Automatic Extraction of Man-Made Objects from Aerial and Space Images, pp.135-144, 1995.

[26] N. Paparoditis, M. Cord, M. Jordan, and J.P. Cocquerez, “Building Detection and Reconstruction from Mid- and High Resolution Aerial Images,” Computer Vision and Image Understanding, vol. 72, no. 2 pp. 122-142, Nov. 1998.

[27] D. McKeown, C. McGlone, S. Cochran, W. Harvey, J. Shufelt, and D. Yocum, “Automated Cartographic Feature Attribution Using Panchromatic and Hyperspectral Imagery,” Proc. DARPA Image Understanding Workshop, pp. 517-536, Nov. 1998.

[28] A. Huertas, R. Nevatia, and D. Landgrebe, “Use of Hyperspectral Data with Intensity Images for Automatic Building Modeling,” Proc. Second Int'l Conf. Information Fusion, pp. 680-687, July 1999.

[29] J.R. Beveridge, J. Griffith, R.R. Kohler, A.R. Hanson, E.M. Riseman, “Segmenting images using localized histograms and region merging”, Int'l. J. of Comp. Vis., vol. 2, 311 – 347, 1989.

[30] D. Hearn, M.P. Baker, Computer Graphics, Prentice Hall, 1994

[31] J.D. Foley, A. Van Dam, S.K. Feiner, J.F. Hughes, R.L. Philips, Introduction to Computer Graphics, Addison-Wesley, 1997

[32] A.R. Smith, Color Gamut Transform Pairs, Proc. SIGGRAPH 78, pp. 12-19, 1978

[33] R. Jain, R. Kasturi, B. G. Schunk, “Machine Vision”, McGraw-Hill Inc., 1995

[34] J. Canny, “A Computational Approach to Edge Detection,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679-698, June 1986.

[35] Q. Ji and R. M. Haralick, “Error propagation for the hough transform," Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA and Department of Electrical Engineering, University of Washington, USA, 2000.

[36] H.Rüther, H. M. Martine and E.G. Mtalo “Application of snakes and dynamic programming optimization technique in modeling of buildings in informal

settlement areas" ISPRS Journal of Photogrammetry & Remote Sensing 56 (2002) 269-282.

[37] J. Peng, D. Zhang and Y. Liu "An improved snake model for building detection from urban aerial images" Pattern Recognition Letters 26 (2005) 587-595.

[38] S. Noronha and R. Nevatia "Detection and modeling of buildings from multiple aerial images" IEEE Transactions on pattern analysis and machine intelligence, vol.23 no.5, may2001

[39] Z. Kim and R. Nevatia "Automatic description of complex buildings from multiple images" Computer vision and image understanding 96 (2004) 60-95.

[40] M. Fradkin, H. maitre and M. Roux "Building detection from multiple aerial images in dense urban areas" Computer vision and image understanding 82, 181-207 (2001).

[41] M. Cord, M. Jordan and J. Cocquerez "Accurate building structure recovery from high resolution aerial imagery" Computer vision and image understanding 82, 138-173 (2001).

[42] D. Jinghui, P. Veronique and L. Hanqing " Building extraction in urban areas from satellite images using GIS data as prior information" Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings.2004 IEEE International.

[43] I. Suveg and G. Vosselman "Reconstruction of 3D building models from aerial images and maps" ISPRS Journal of Photogrammetry & Remote Sensing 58 (2004) 202-224

[44] D. Koç and M. Türker “Automatic building detection from high resolution satellite images” Proceedings on 2nd International Conference on Recent Advances in Space Technologies, 2005. RAST2005, pages 617-622.

[45] J.Hu, S. You and U. Neumann “Integrating LiDAR, aerial image and ground images for complete urban building modeling” Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT’06).

[46] F.O. Chikomo, J.P. Mills and S.L. Barr “An integrated approach to level-of-detail building extraction using airborne lidar and optical imagery” PIA07 Photogrammetric Image Analysis, Munich, Germany, September 19-21, 2007

[47] X. Jin and C.H. Davis “Automated building extraction from high-resolution satellite imagery in urban areas using structural, contextual, and spectral information” EURASIP Journal on Applied Signal Processing 2005:14, 2196-2206

[48] C. Unsalan and K.L. Boyer “A system to detect houses and residential street networks in multispectral satellite images” Computer Vision and Image Understanding 98 (2005) 423-461

[49] C. Vestri “Using range data in automatic modeling of buildings” Image abd Vision Computing 24 (2006) 709-719

[50] C. Jaynes, E. Riseman and A. Hanson “Recognition and reconstruction of buildings from multiple aerial images” Computer Vision and Image Understanding 90 (2003) 68-98

[51] T. Kim and J.P. Muller "Development of a graph-based approach for building detection" Image and Vision Computing 17 (1999) 3-14

[52] F. Tupin and M. Roux "Detection of building outlines based on the fusion of SAR and optical features" ISPRS Journal of Photogrammetry & Remote Sensing 58 (2003) 71-82

[53] P. Perona and J. Malik "Scale-space and edge detection using anisotropic diffusion" IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(7):629-639, July 1990.

[54] C. Tomasi and R. Manduchi. "Bilateral Filtering for Gray and Color Images". Proceedings of the IEEE International Conference on Computer Vision, 1998.