MODEL BASED BUILDING EXTRACTION FROM HIGH RESOLUTION AERIAL IMAGES

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

BY

BURAK BİLEN

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

GEODETIC AND GEOGRAPHIC INFORMATION TECHNOLOGIES

MAY 2004

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Oğuz Işık
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Mustafa Türker
Supervisor

Examining Committee Members

Asst. Prof. Dr. M. Onur Karslıoğlu _____

Asst. Prof. Dr. Mustafa Türker _____

Dr. Uğur Murat Leloğlu _____

Dr. Jurgen Friedrich _____

MSc. Turan Yüksel _____

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: Burak Bilen

Signature :

ABSTRACT

MODEL BASED BUILDING EXTRACTION FROM HIGH RESOLUTION AERIAL IMAGES

Bilen, Burak

MSc, Department of Geodetic And Geographic Information Technologies

Supervisor: Asst. Prof. Dr. Mustafa Türker

May 2004, 70 pages

A method for detecting the buildings from high resolution aerial images is proposed. The aim is to extract the buildings from high resolution aerial images using the Hough transform and the model based perceptual grouping techniques. The edges detected from the image are the basic structures used in the building detection procedure. The method proposed in this thesis makes use of the basic image processing techniques. Noise removal and image sharpening techniques are used to enhance the input image. Then, the edges are extracted from the image using the Canny edge detection algorithm. The edges obtained are composed of discrete points. These discrete points are vectorized in order to generate straight line segments. This is performed with the use of the Hough transform and the perceptual grouping techniques. The straight line segments become the basic structures of the buildings. Finally, the straight line segments are grouped based on predefined model(s) using the model based perceptual grouping technique. The groups of straight line segments are the candidates for 2D structures that may be the buildings, the shadows or other man-made objects. The proposed method was implemented with a program written in C programming language. The approach was applied to several study areas. The results achieved are encouraging. The number of the extracted buildings increase if the orientation of the buildings are nearly the same and the Canny edge detector detects most of the building edges. If

the buildings have different orientations, some of the buildings may not be extracted with the proposed method. In addition to building orientation, the building size and the parameters used in the Hough transform and the perceptual grouping stages also affect the success of the proposed method.

Keywords: Building detection, edge detection, model, Hough Transform, perceptual grouping, Canny

ÖZ

YÜKSEK ÇÖZÜNÜRLÜKLÜ HAVA FOTOĞRAFLARINDAN MODEL TABANLI METODLA BİNALARIN BELİRLENMESİ

Bilen, Burak

Master, Jeodezi ve Coğrafi Bilgi Teknolojileri Bölümü

Tez Yöneticisi: Yard. Doç. Dr. Mustafa Türker

Mayıs 2004, 70 sayfa

Yüksek çözünürlüklü hava fotoğraflarından binaların bulunmasıyla ilgili bir yöntem önerilmektedir. Amaç, Hough dönüştürücüsünü ve model tabanlı algısal gruplama tekniğini kullanarak yüksek çözünürlüklü hava fotoğraflarından binaların belirlenmesidir. Resim üzerindeki kenarlar bina bulma yönteminde kullanılan temel yapı taşlarıdır. Tezde önerilen yöntemde temel görüntü işleme tekniklerinden yararlanılmaktadır. Resmi netleştirmek amacıyla parazit yoketme ve görüntü keskinleştirme teknikleri kullanılmaktadır. Kenarlar, Canny kenar bulma algoritması kullanılarak resimden çıkarılırlar. Elde edilen kenarlar ayrık noktalardan oluşmaktadırlar. Doğrusal çizgi parçalarını üretebilmek için, elde edilen ayrık noktalar yöneylendirilirler. Doğrusal çizgi parçası üretimi, Hough dönüştürücüsü ve algısal gruplama teknikleri kullanılarak

gerçekleştirilir. Doğrusal çizgi parçaları binaların temel yapı taşlarıdır. Son olarak, eldeki doğrusal çizgi parçaları önceden tanımlanmış modeller temel alınarak, model tabanlı algısal gruplama tekniğiyle gruplanırlar. Gruplanmış doğrusal çizgiler 2 boyutlu yapıları oluştururlar. Bu 2 boyutlu yapılar, bina, gölge veya insan yapımı başka bir nesne olabilir. Önerilen yöntemi hayata geçirmek için C programlama dili kullanılarak bir program yazıldı. Yöntem birkaç çalışma alanına uygulandı. Elde edilen sonuçlar daha sonraki çalışmalar için cesaret verici. Resim üzerindeki bina yönleri birbirine yakın olduğu ve Canny kenar bulucusu çoğu bina kenarlarını bulduğu durumda önerilen yöntemle belirlenen binaların sayısı artmaktadır. Bina yönleri farklı olduğu durumda ise bazı binalar belirlenememektedir. Bina yönlerinin yanında, bina boyutları ve Hough dönüştürücüsü ile algısal gruplama tekniğinde kullanılan parametreler önerilen yöntemin başarısını etkilemektedir.

Anahtar Kelimeler: Bina bulma, kenar bulma, model, Hough Dönüşümü, algısal gruplama, Canny

To my family

ACKNOWLEDGMENTS

I thank

my supervisor Asst. Prof. Dr. Mustafa Türker for his supervision, continuous support and guidance throughout the development of this thesis,

Assoc. Prof. Dr. Volkan Atalay for his valuable suggestions and comments,

my family for their endless love and support,

my colleagues Ali Güler, Tayfun Asker, Çağlar Bilir, Ahmet Öztürk, Feyza Taşkazan and Mustafa Atakan for their understanding and help when I am busy with my thesis,

METU Computer Center staff for the technical infrastructure they supply,

and all other friends who helped in producing this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Building detection from high resolution digital images has been a major research issue for more than two decades. Manual detection of buildings from images requires an unacceptable amount of effort both in terms of time and cost. With manual detection it is possible to detect all of the buildings accurately. However, the labor needed to detect the buildings increases as the study area worked on becomes larger and complex. Also the time needed in manual building detection increases which is not suitable for real-time applications. The studies on (semi-)automatic building detection aim to reduce this effort and produce reliable results. The purpose of (semi-)automatic building detection methods is to minimize the human intervention in the detection process and produce accurate results as in the manual detection method. Building detection study can be used in many applications such as military simulations and training ([1]), and city planning.

The previous building detection studies present semi-automatic or automatic methods. The building detection methods proposed in [2][3][4][5][6] are semi-automatic,

while the methods proposed in [1][7][8] are automatic. When compared with a semi-automatic building detection method, the construction of an automatic method is difficult and generally more information sources are needed. In semi-automatic building detection methods, the human intervention is involved in order to complete the building detection process. The automatic methods are much faster than semi-automatic methods but they may not produce the desired results. There may be missing buildings in the output and also there may be some false positives. The data used in this thesis may not be satisfactory for a successful fully automatic building detection process. In this thesis, a monoscopic high resolution pachromatic imagery is used to detect the buildings. From the image, the linear structures are extracted and the model(s) that determine the relations between these linear structures are used for detecting the buildings. Therefore, with this data only, it may not be possible to improve the success of building detection considerably and to eliminate the false positives. The use of more than one image, the intensity information or DEM (digital elevation model) information as supplementary data sources would increase the success and the accuracy of the proposed method. However, this is left as a future work.

This thesis presents a method that offers a solution to the problem of 2D building detection from high resolution aerial and space imagery. The proposed method is a model based method, the purpose of which is detecting 2D structures that fit the predefined model(s). The buildings are 2D structures that are composed of straight line segments. In digital images, the straight line segments are composed of pixels. Therefore, the building detection process starts with finding the edge pixels through an edge detection procedure. Then, these pixels are grouped into straight line seg-

ments and finally the straight line segments are converted into 2D structures based on the predefined model(s). To complete all these steps, the known algorithms are used.

In the present case, the edge detection process is carried out using the Canny edge detection algorithm. The straight line segments are then generated through Hough transform and the perceptual grouping. Finally, the grouping of the straight line segments is carried out using a model based perceptual grouping technique. The Hough transform is generally used for extracting the straight lines such as, roads ([9]), the edges of urban structures etc. from digital images. In building detection, the Hough transform has not been used widely. Several researchers have used Hough transform as a help function. For example, in [10], Hough transform was used for generating straight edges from irregular boundaries. In their method, the Hough transform was applied to a portion of the image that contains only a single building. In this study, the Hough transform was applied to a part of the image[1] that contains several buildings.

The main objectives of this thesis are as follows:

1. To develop a model based building extraction from high resolution images.

2. To use Hough transform for extracting straight line segments.

3. To show the usability of a model based perceptual grouping for grouping the straight line segments generated using the Hough transform.

[1] In some situations the image is divided into regular cells. The Hough transform is applied to these cells, not the whole image (Section 3.1.3)

1.1 Data

Nadir or near nadir view high resolution aerial images are used for testing the method. The aerial images used in the tests belong to the urban places which are acquired in 1995. The images are 1:4000 scale and the spatial resolution of the images are 50 cm per pixel.

1.2 Organization of the Thesis

The organization of the thesis is as follows: In Chapter 2, the related work on building detection are briefly reviewed. The algorithms used in this study are also briefly explained in Chapter 2. In Chapter 3, the method used in this thesis is explained in detail. In Chapter 4, the method is applied to several study areas. The results and the discussions related to each study area and the discussions regarding to all study areas are given in Chapter 4. Finally, the conclusions and the future works are provided in Chapter 5.

CHAPTER 2

BACKGROUND

In this chapter, the previous studies carried out on building detection are described and the concept used in this study is briefly explained. The previous studies were taken as a basis while preparing this study. Some of the algorithms, which form the basic steps of the previous studies, were also used in this study.

2.1 Building Detection

Detection of the buildings manually from high resolution images requires an unacceptable amount of effort both in terms of time and cost ([2]). Therefore, researchers have been working on semi-automated and automated building detection methods for more than two decades in order not to spend too much effort for manually detecting the buildings.

Early building detection systems used a single intensity image ([4][11]) and were effective for simple buildings only. These methods rely on edges or regions extracted from the image ([11]). Simple edge-based methods attempt to collect linked edge

curves into the desired object boundaries and succeed only for relatively simple scenes ([11]). Some of the edge-based methods have used a contour tracing technique ([12]). In [13], a method was described for accurately modeling buildings from a single image. A single image and the edges extracted from that image is not enough to detect the buildings accurately. Therefore, the studies that used a single image as the primary data source, used shadows ([5][11]) or hyperspectral data ([7]) as the supplementary data source. Shadows and hyperspectral data were used for verifying the buildings in automatic building detection methods. In semi-automatic building detection methods, human intervention is involved ([4]) in addition to the supplementary data used in automatic building detection methods. Human intervention is useful for detecting the missing buildings and eliminating the false positives. In general, building detection methods are divided into semi-automatic ([2][3][4][5][6]) and automatic ([1][7][8]) methods.

Semi-automatic building detection is performed with the help of an analyst. The analyst helps the computer to detect the buildings accurately. The analyst gives the clues to the computer and the computer detects the buildings with the help of these clues. The clues should be reliable and enough to detect the buildings accurately. The clues given by the analyst is very important because the rest of the work is performed by the computer with the use of these clues. The detection method may completely fail if the analyst makes a mistake. At the end of the detection process, some of the buildings may not be found or some other objects may be detected wrongly as buildings. Then, the analyst marks the missed buildings manually and removes the false positives. The semi-automatic building detection methods may produce good results

in terms of accuracy and efficiency if the methods are implemented properly. Since most of the work is performed by the computer, the effort spend by the analyst is reasonable ([2]).

Automatic building detection is performed without an intervention or with a small intervention of an analyst. Automatic building detection is a complex research area and generally requires more than one data source for detecting the buildings accurately. There are several difficulties in automatic building detection to deal with ([5]). For example, some of the parts of the buildings may be occluded by other buildings or trees, or the buildings may be darkened by shadows. In addition, the buildings may be in different shapes and sizes. Automatic building detection methods may not be able to handle all of these difficulties. Therefore, the methods may not be able to produce the reliable results.

Several automated building detection methods have been proposed in the literature using the hypothesis generation and verification for the presence of rooftops in the image ([14]). In hypothesis generation, 2D structures are detected in the input image using a data source. Edges, in the input image are the most widely used data source in the hypothesis generation. Edges are extracted with the use of an edge detector (see section 2.2.1). Then, these edges are brought into meaningful straight line segments (see section 2.2.2). The straight lines are later grouped based on predefined relations (see section 2.2.3) forming the 2D structures. The 2D structures are building hypothesis (BH), that is these 2D structures may be buildings or shadows or other man-made objects like cars, park areas etc. In order to mark the 2D structures as

buildings, the 2D structures should pass through a verification process.

For the hypothesis verification, another data source is used. For example, DEM (digital elevation model) data, intensity data of the input image, or the shadow information can be used as additional data source. Intensity information can be used to discriminate the buildings from shadows. Shadows are dark regions and are represented with low intensity values ([5]) that make them easy to detect. If a shadow area is detected wrongly as a building, this can easily be understood with the intensity information. The intensity information may not be enough for hypothesis verification. For example, cars or other man-made objects may have the same intensity values as the buildings. In such a case additional data source should be provided to complete the verification process. The shadow information obtained from intensity information can be used as supplementary data source. For example, if a building's shadow is larger than a car's shadow, then a building and a car can be separated from each other with the help of the shadow information. After the verification process, each building hypothesis indicates a building or part of a building ([15]).

2.1.1 Previous Studies

The building detection methods are mainly based on grouping of primitives, like edges or segments or matching the predefined parametric models ([14]). Some of the proposed methodologies include the use of additional information, such as topographic maps ([16][17]), segmentation and Hough transform ([10]), multispectral and hyperspectral images ([18][19]), as well as various building models ([13][20]).

In [11], an automated building detection system from aerial images was proposed. To collect the edge segments and discard the edge segments that come from other sources, the perceptual grouping approach was used. The shape properties of the buildings were used for the perceptual grouping. The shadows and the walls were used to verify the hypotheses generated by the grouping process. In [11], 3D building reconstruction was proposed but the basic steps were the same as the 2D building detection methods. Their system uses the line segments that approximate the intensity boundaries to compute linear structures and the junctions among them. A hierarchy of features was constructed, leading to the formation of parallelogram hypotheses. Next, the promising hypotheses were selected and verified to correspond to roofs of buildings. The shadow and wall information, if available, was used to help form, select and verify the hypotheses. The proposed system's performance was not good since it finds all possible parallelograms and then tries to eliminate these parallelograms based on the shadow information. The results were satisfactory in terms of accuracy.

In [4], a semi-automated building detection method was proposed. The system was designed to find rectilinear building structures from a monocular image. Perceptual grouping techniques were used to find likely candidates for building roofs and the cast shadows were used to confirm them and to estimate the height. The system's performance was generally quite good when sufficient parts of the roof boundaries could be extracted, that is the extracted boundaries should be enough to detect L, T or I shaped buildings. However, in some cases, such as when the roof was dark, the boundary of the roof with the shadow was not detected and the system failed to confirm the presence of such buildings due to lack of sufficient evidence. In such cases, a

very simple guidance from the analyst, just indicating that in fact a dark building was present. The proposed method in [4] was tested on a number of examples provided by the RADIUS program and generally performed good results with a small interaction of the analyst.

In [2], a user assisted (semi-automatic) building detection method was proposed. The basic detection tasks were performed by an automatic system and the system received simple, but critical, assists from the user. The shapes of the buildings were restricted to be rectilinear. The basic steps of the system consist of forming the parallelogram hypotheses (to represent rectangular parts of roofs) in one image and inferring 3D shapes from them. A user can assist the system in the process of hypothesis formation as well as in making corrections to the resulting 3D models. A user interaction typically consists of the user pointing to a point or line feature. Each interaction from a user was called as a *click*. The system required two or more views of a scene with the associated camera geometry; however, all user interactions took place in one view only. The system was applied to a number of images. For flat roof buildings, two or three *clicks* from the user was enough to detect a building. For gabled roof buildings, three or four *clicks* were used to detect a building. The system's performance in detecting buildings correctly was good but the time required for this process was much more than an automated system.

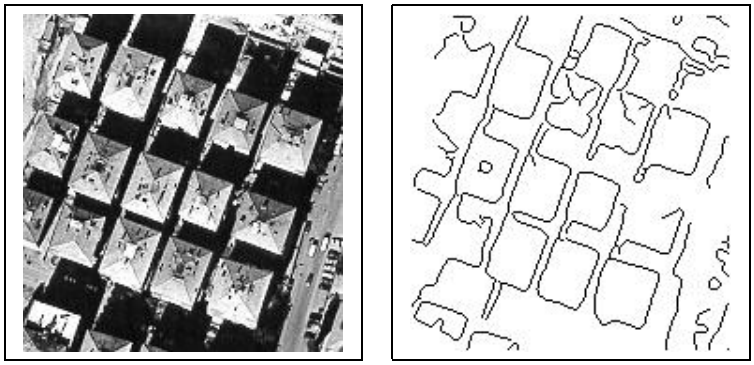
In [10], a different approach was proposed. In their approach, the classification output of an Ikonos multispectral imagery is used to provide approximate location and the shape for candidate buildings. The fine extraction was then carried out in the

corresponding panchromatic image through segmentation and squaring. A building squaring approach based on the Hough transform was developed that detect and form the rectilinear building boundaries. The buildings were composed of straight lines that meet at 90-degree angles. About 64.4% of the buildings were detected using the proposed method.

2.2 Concepts used in this Thesis

2.2.1 Edge Detection

Edges can be defined as the locations in an image where there is a sudden variation in the gray level or color of the pixels ([21]). An edge detection process extracts the seed properties of an image. The seed properties for the study carried out in this thesis are the building edges. Figure 2.1 illustrates the edge detection process on an image. The basic edge detectors are build on the estimation of grey level gradient at a pixel ([21]).



(a) Original image (b) Edge detected image

Figure 2.1: Edge detection

The most widely used edge detectors are Sobel, Prewitt, Roberts, Canny and Marr-Hildreth. Of these edge detectors, the Canny edge detector is considered to be optimal. It produces one pixel wide edges and connects the broken lines. Therefore, it is used in this thesis. The Canny edge detection is a step by step process. In the first step, the noise in the input image is filtered out. An appropriate Gaussian filter ([21]) is applied to the input image using the standard convolution methods for removing the noise and smoothing the image. A Gaussian filter is the combination of Laplacian and Gaussian functions. The σ in equation 2.1 determines the width of the filter and controls the amount of smoothing produced by the Gaussian component. The kernel used to implement a LoG filter should have a half-width of at least 3σ.

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\Big[1 - \frac{x^2+y^2}{2\sigma^2}\Big]e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.1}$$

After applying a Gaussian filter, the next step is to find the edge strength by taking the gradient of the image. The gradient magnitude is found by using the Sobel Filter. Figure 2.2 illustrates the Sobel filters used to find the gradient values in x and y directions.

Next, by using the gradient (G_x and G_y) values computed in the previous step, the

−1	0	1
−2	0	2
−1	0	1

a) G_x

−1	−2	−1
0	0	0
−1	2	1

b) G_y

Figure 2.2: The Sobel filter

edge direction (α) is found (Equation 2.2).

$$\alpha = invtan(G_y/G_x) \tag{2.2}$$

To be able to trace the edge direction, the edge direction should be converted to 0^0, 45^0, 90^0, or 135^0 by using the following rules. The conversion of α is necessary for non-maximum suppression.

$(0^0 \leq \alpha \leq 22.5^0$ or $157.5^0 \leq \alpha \leq 180^0) => \alpha = 0^0$

$(22.5^0 < \alpha < 67.5^0) => \alpha = 45^0$

$(67.5^0 \leq \alpha \leq 112.5^0) => \alpha = 90^0$

$(112.5^0 < \alpha < 157.5^0) => \alpha = 135^0$

The next step is the non-maximum suppression step. The purpose of the non-maximum suppression is to generate one-pixel wide edges. Algorithm 2.1 is used for non-maximum suppression.

Algorithm 2.1 Non-maximum suppression in the Canny edge detector ([21])

Create an output image, g_s, with the same dimensions as g

for all pixel coordinates, x and y, **do**

  if $g(x, y) < g$ at neighbor in direction α **or**

  $g(x, y) < g$ at neighbor in direction $\alpha + 180$ **then**

    $g_s(x, y) = 0$

  else

    $g_s(x, y) = g(x, y)$

  end if

end for

The final step in the Canny edge detector is the hysteresis thresholding. Hysteresis thresholding is different from the conventional thresholding procedure. In the hysteresis thresholding there are two threshold values. The pixels whose gradient values are larger than the larger threshold value (T1) are marked as edge pixels. If the remaining pixels' gradient values are larger than the smaller threshold value (T2) and if these pixels are neighbor to the edge pixels, then these pixels are also marked as edge pixels. Other pixels whose gradient values are smaller than T2 are non-edge pixels. Hysteresis thresholding helps to connect the broken lines in producing full edges in the output.

2.2.2 Hough Transform

The Hough Transform (HT) algorithm was developed by Paul Hough in 1962 ([22]). It is a method for detecting straight lines and curves, which have parametric representations, on gray level images. The output of an edge detection process (Canny edge detection) is used as the input for the HT. The output of the Canny edge detector consists of white pixels (pixel value=255) as edge pixels and black pixels (pixel value=0) as background pixels. Edge pixels are grouped into straight lines using the HT. The HT uses the following line equation for expressing the straight lines:

$$\rho = x * cos(\theta) + y * sin(\theta) \tag{2.3}$$

where; θ is the line orientation and ρ is the perpendicular distance from the origin to the line shown in Figure 2.3. The edge pixels in the output of the Canny edge detector are grouped according to equation 2.3. The purpose is to find the number of edge pixels that satisfy the same θ and ρ.

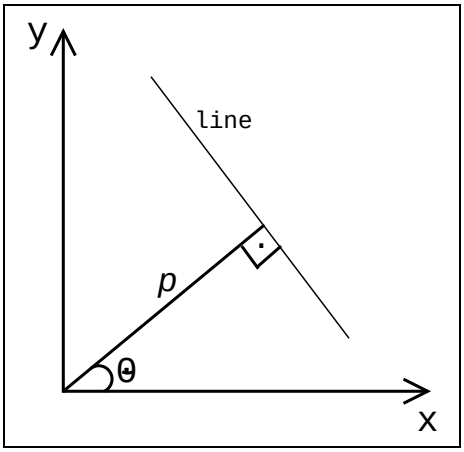


Figure 2.3: The parametric description of a straight line

All θ values between θ_1 and θ_n (Figure 2.4) are scanned for each pixel and the corresponding ρ value is found. θ_1 is greater than 0^o and θ_n is less than 180^o. The

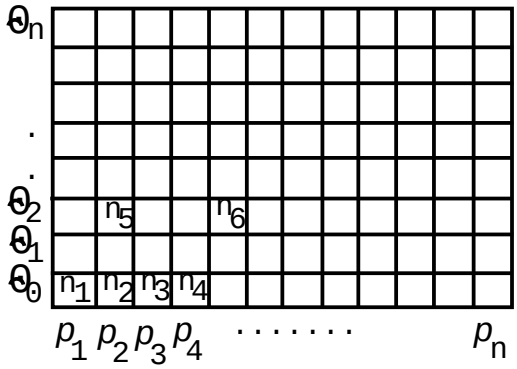


Figure 2.4: Hough accumulator array

difference between the consecutive θ values is constant and can take any integer value between 1 and $(\theta_n - \theta_1)$. The value of this constant (TINC in table 3.1) determines the sensitivity of the Hough transform. The smaller the value of TINC means the more details are detected but also the more noise can be present in the output. On the other hand, the larger the value of TINC the less noise can be present in the output image. However, in this case some important details may be lost and therefore, may result in missing some of the buildings.

At the beginning, there is a two dimensional accumulator array corresponding to θ and ρ that are initialized to 0. If a pixel satisfies a (θ, ρ) pair, the corresponding entry in the accumulator is increased by 1. After all the pixels are processed, the values in the accumulator array (Figure 2.4) are checked. If the value in the accumulator is greater than a threshold value, it is concluded that the (θ, ρ) pair represents a straight line. Figure 2.4 illustrates the relation between θ and ρ. Some of the cells in Figure 2.4 are empty. This means that there is no pixel found to satisfy the ρ value with the given θ.

2.2.3 Perceptual Grouping

Perceptual grouping is a technique to organize image primitives into higher level primitives using several predefined rules. The ideas of perceptual grouping for computer vision are based on the well known work of Gestalt psychologists. Gestalt is the German word for *form* and as it applied in gestalt psychology it means *unified whole* or *configuration*. The essential point of gestalt is that in perception the whole is different from the sum of its parts. There are several laws of Gestalt theory related to perception described in section 2.2.3.1. Some of these laws are used while implementing the perceptual grouping stage.

2.2.3.1 Laws of the Gestalt Theory

1. **Law of Proximity**

   The Gestalt law of proximity states that "objects or shapes that are close to one another appear to form groups" ([23]). Even if the shapes, sizes, and objects are radically different, they will appear as a group if they are close together.

2. **Law of Closure**

   The Gestalt theory seeks completeness; with shapes that are not closed, they seem incomplete and lead the learner to want to discover what is missing, rather than concentrating on the given instruction.

3. **Law of Symmetry**

   The Gestalt theory espoused the symmetrical so that the learner is not given the impression that something is out of balance, missing or wrong ([23]). Again, if an object is asymmetrical, the learner will waste time trying to find the problem instead of concentrating on the instruction. The chunking, or grouping, of information should follow a logical pattern.

4. **Figure-Ground Segregation**

   For a figure to be perceived, it must stand apart from its background.

5. **Law of Good Continuation**

   The Gestalt law states that learners "tend to continue shapes beyond their ending points" ([23]).

6. **Law of Similarity**

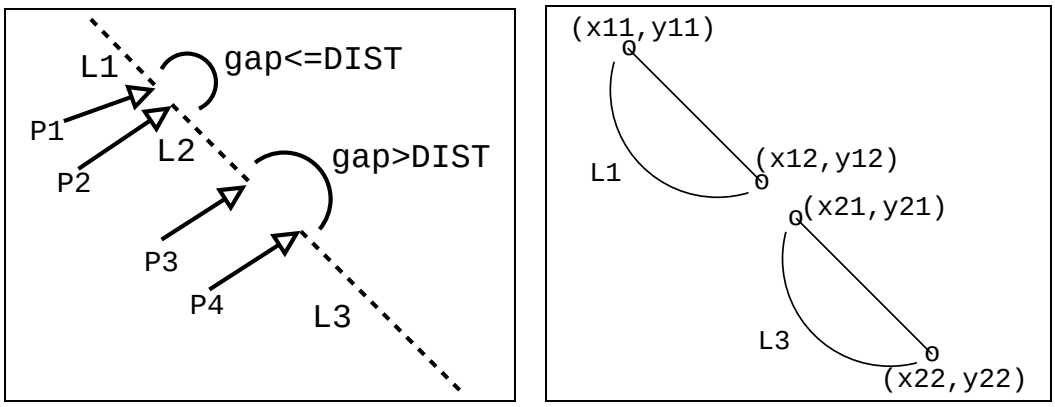
   The Gestalt theory states that objects that appear to be similar will be grouped together in the learner's mind ([23]). For example, for visual instruction, this can include the font styles, size, and color.

The Getstalt laws are adapted to the perceptual grouping and straight line segments are grouped based on some of the Gestalt laws. The perceptual grouping proposed in this thesis is a model based perceptual grouping. Therefore, the Gestalt laws are applied based on the building models.

2.2.3.2 Model Based Perceptual Grouping

Detecting buildings based on models is not a new approach. In several studies ([13][20]) models were used in the building detection procedure. It is not always possible to fit the models to the image and detect the buildings because the straight line segments in the image may not be detected properly and therefore, they may not be grouped into a model. In order to fit the model(s) to the image, the straight line segments should be grouped using the perceptual grouping process. While grouping the straight line segments perceptually, the parameters used in perceptual grouping should be flexible and appropriate tolerance values should be used in order to tolerate the parameters. In this thesis only the **rectangular** building model was used. The purpose is to group the straight line segments to generate the rectangular buildings.

The perceptual grouping stage starts with the straight line segment generation. If the end points of two straight line segments are closer (Law of Closure) to each other then, these lines are connected to form a larger straight line segment (Figure 2.5).



(a) Canny edge detector output

(b) Perceptual grouping output

Figure 2.5: Canny edge detector to perceptual grouping conversion

The generation of rectangle is performed by grouping the corners. The corners are represented by a pair of perpendicular and adjacent straight line segments (Figure 2.6). Two straight line segments are defined as perpendicular if the absolute value of the difference between their orientation is in the range of $90^0 \pm$ AT (AT: angle threshold). Adjacency (Law of Proximity) is defined by the distance between the ends of the lines. The straight line segments are considered adjacent if the distance between them is shorter than a predefined distance value DIST ([5]).

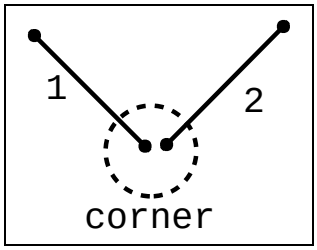
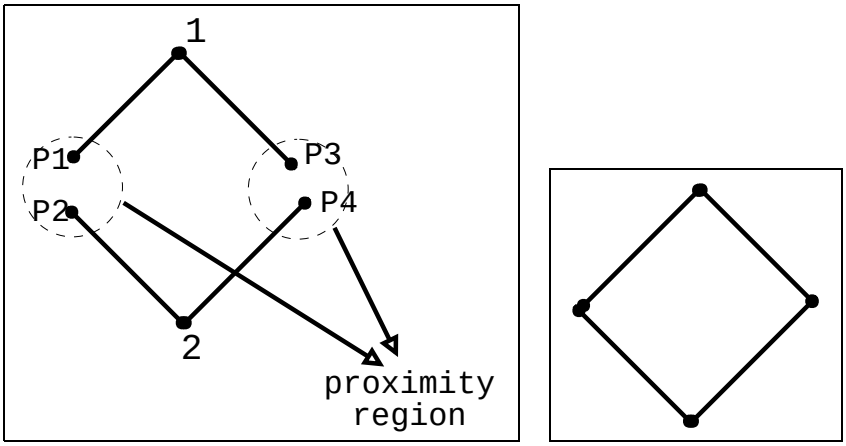


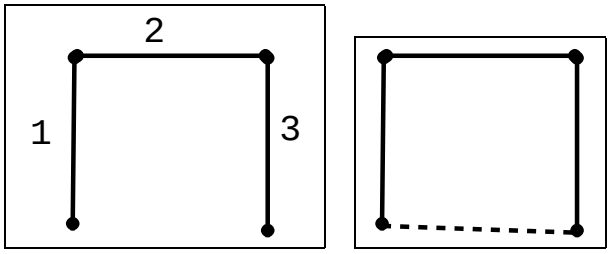
Figure 2.6: Corner detection

The generation of a rectangle is performed by grouping the corners obtained in the previous step. If four corners are detected, which are ordered as a rectangle, then these four corners represent a rectangular building. However, finding all of the four corners is generally not possible. Therefore, generation of a rectangle is performed with the pairs of corners that satisfy symmetrism (Law of Symmetry) (Figure 2.7) or parallelism (Figure 2.8). All of the corners are traced until all the rectangular buildings are detected. In order to fit a rectangle model to the image, it may be necessary to adjust AT and DIST parameters accordingly.



(a) (b)

Figure 2.7: The concept of symmetrism in perceptual grouping



(a) (b)

Figure 2.8: The concept of parallelism in perceptual grouping

CHAPTER 3

THE METHODOLOGY

In Chapter 2, the widely used building detection methods were explained. These methods have common or different parts and are generally the improvements of the old methods. They are basically the same but in order to increase the accuracy of the methods, some modifications have been made and new properties have been added to the methods. The rapid development of the computer technology increased the performance of the methods. High performance computers with high processing speeds and high physical memory have given opportunities to try more complex methods for researchers who have been working in the field of building detection. The method proposed in this thesis use the basic steps similar to the existing methods. However, the use of Hough transform in building detection is not very common.

3.1 The Methodology

The flow diagram of the proposed methodology is given in Figure 3.1. The concept is based on Hough transform and the perceptual grouping. The removal of the noise and the edge detection steps are carried out to prepare the input data for the Hough

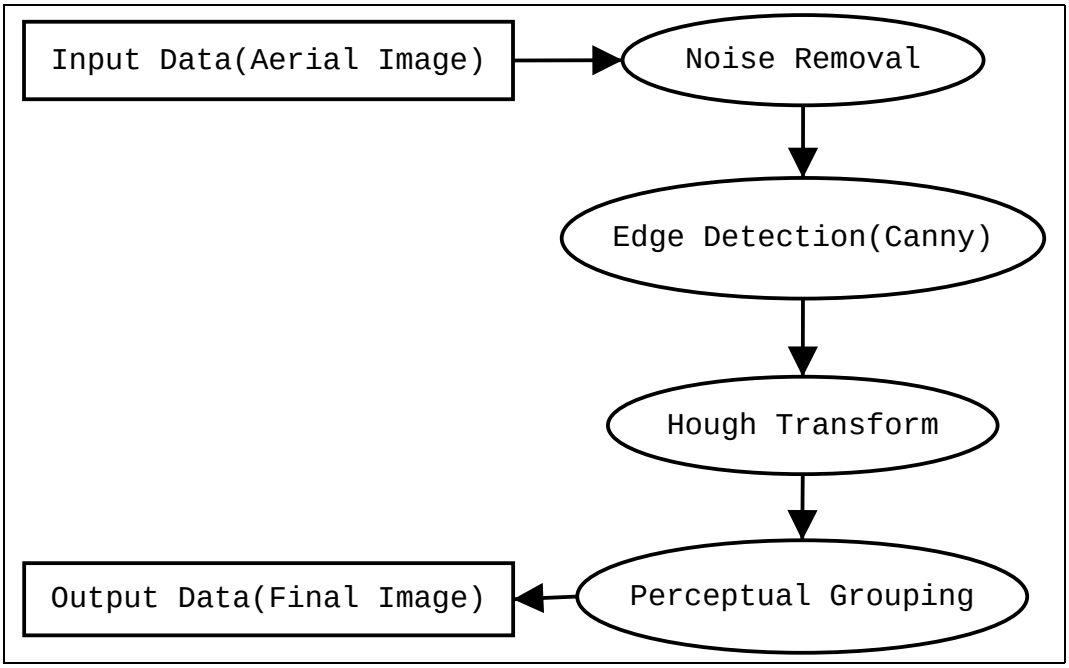


Figure 3.1: The flow diagram of the method

transform. In the edge detection stage, the edges are extracted from the input image. The output of the edge detection process is composed of discrete points that are not useful in detecting the buildings. To represent the building edges, the straight line segments are required. Therefore, these discrete points must be grouped into straight line segments. The generation of the straight line segments is performed through perceptual grouping that operates on the output of the Hough transform. These straight lines are then grouped based on several building models that compose the buildings. The grouping of the straight line segments is also carried out through perceptual grouping procedure.

3.1.1 Noise Removal

Because of the air conditions and the effects of the cast shadows, the aerial images generally contain noise. The noise affects the building detection process negatively. Therefore, as the first step, the noise is removed from the input aerial image. In the

present case a median filter was used to remove the noise. A median filter usually produces successful results. The filtering process was carried out using the Matlab software. In addition to noise, the aerial images may be blurred and the edges may be lost because of the blurring effect. Therefore, to remove the blurring effect, the image should be enhanced using an image sharpening method. GIMP (GNU Image Manipulation Program) software is used for sharpening the images. In the Noise Removal stage, both the median filtering and the enhancing procedures were carried out (Figure 3.1). The noise removal is considered to be an important stage in order to get good results during the edge detection stage.

3.1.2 Edge Detection

Performing an edge detection is the second processing stage in the building detection procedure. In this stage, the pixels forming the edges of the buildings are extracted. There are several known edge detection algorithms including Sobel, Prewitt, Roberts, Canny and Marr-Hildreth. Of these techniques, the Canny operator is used for detecting the edges in this thesis. The Canny edge detector produces one pixel wide edges, which is important for Hough transform in order to produce less noisy output. As well, the Canny edge detector is known to produce good results since it merges the broken lines. The Canny edge detection algorithm is described in section 2.2.1.

3.1.3 Hough Transform

The edges produced in the edge detection step are not meaningful for detecting the buildings. The output of the edge detection is composed of discrete points. These discrete points must be grouped into straight lines. This process is also called vectorization. The vectorization starts in the Hough transform stage and continues in the

perceptual grouping stage. The Hough transform produces (θ, ρ) pairs as the output. A (θ, ρ) pair means that a line cuts off the image from one edge and acrosses to the other edge. The Hough transform is described in section 2.2.2.

Good results can be obtained from Hough transform if the numbers in Hough accumulator array increases (Figure 2.4). There are several parameters (Table 3.1) that effect the success of the Hough transform. These parameters should be arranged according to the input image.

Table 3.1: Hough Transform Parameters

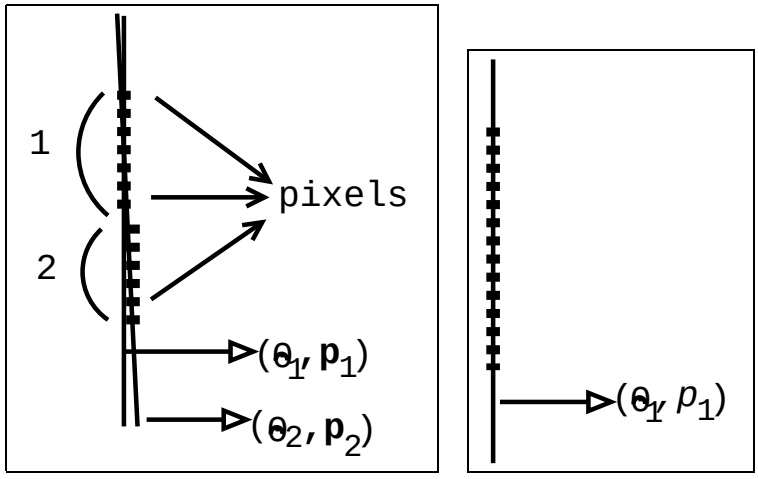
Parameter	Description
MULT	Multiplier, used to determine the minimum number of pixels representing a (θ, ρ) pair
TINC	Constant value between consecutive θ values
TOL	The deviation in the ρ of a straight line
XDIV	Divide the input image into XDIV parts in x-axis
YDIV	Divide the input image into YDIV parts in y-axis

Of these parameters, the threshold (minimum value in Hough accumulator array) is very important. If the threshold (T) is kept high, some straight lines may be lost. On the other hand, if the threshold is kept low, too much noise may be seen in the output. Therefore a relative threshold value is used in the implementation. That is the largest value ($n_{largest}$) is found in the Hough accumulator array and multiplied with a *multiplier* (MULT) in order to find the threshold (Equation 3.1). The multiplier can take any value in the range between 0.00 and 1.00. Since the multiplier is a floating point number the result in equation 3.1 is converted into an integer. It is more flexible to use a relative threshold because the user does not care about the values in Hough

accumulator array.

$$T = n_{largest} * MULT \tag{3.1}$$

The other parameter TINC (theta increment) describes the sensitivity of θ. If the value of 1 is selected for TINC, the image is scanned with the θ values of $0^0, 1^0, 2^0, ..., 179^0$. Similarly, if TINC is set to 5 then, the values of θ become $0^0, 5^0, 10^0, ..., 175^0$. The TINC has the similar effects as the threshold (T). If TINC decreases, the noise may increase. If TINC increases, some of the straight lines may be lost. If the value of TINC is low then, the same pixels may satisfy the consecutive θ values and nearly equal ρ values (Figure 3.2(a)).



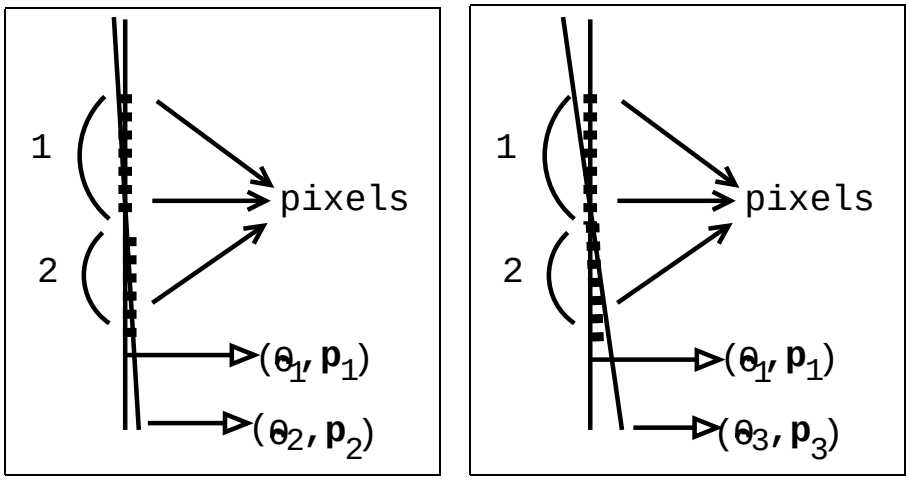
(a) (b)

Figure 3.2: The effect of TINC

The value of TINC is effective with the tolerance value (TOL). As can be seen in figure 3.2(a), both pixel group 1 and pixel group 2 satisfy θ_1 and θ_2 although the ρ values that correspond to these θ values are different in quantity. However, $\mid \rho_1 - \rho_2 \mid \leq TOL$ therefore, both pixel groups satisfy (θ_1, ρ_1) and (θ_2, ρ_2). If both (θ, ρ) values are selected, this causes noise in the output of the Hough transform.

Therefore, only (θ_1, ρ_1) is selected to represent the pixels (Figure 3.2(b)). For example, let us assume that there are 20 pixels satisfying (0^0,16.0) and the same pixels satisfying (1^0,16.4). If both of the (θ, ρ) pairs are used then, there is noise in the output. This means that the same pixels would belong to both (0^0,16.0) and (1^0,16.4). Therefore, one of the (θ, ρ) pairs should be eliminated and the appropriate (θ, ρ) pair should be used for the next stage. To solve this problem, larger TINC values can be selected and the pixels are made to appear in only one (θ, ρ) pair. This can be seen in figure 3.3. The value of TINC parameter ($\mid \theta_2 - \theta_1 \mid$) in figure 3.3(a) is smaller than the value of TINC parameter ($\mid \theta_3 - \theta_1 \mid$) in figure 3.3(b). In figure 3.3(a), pixel groups 1 and 2 are represented by both (θ_1, ρ_1) and (θ_2, ρ_2). But it is enough to represent the pixels with only (θ_1, ρ_1). Therefore, there would be noise if these pixels are represented with both (θ_1, ρ_1) and (θ_2, ρ_2) pairs in figure 3.3(a). In figure 3.3(b), pixel groups 1 and 2 are only represented by (θ_1, ρ_1). (θ_3, ρ_3) does not represent the pixel groups because it does not cover the top half of the pixel group 1 and bottom half of the pixel group 2. Therefore, by increasing the value of the TINC parameter, it is provided that the pixels in figure 3.3 be represented by only (θ_1, ρ_1). In the tests, the TINC value was set to 3 which generally performed good results.

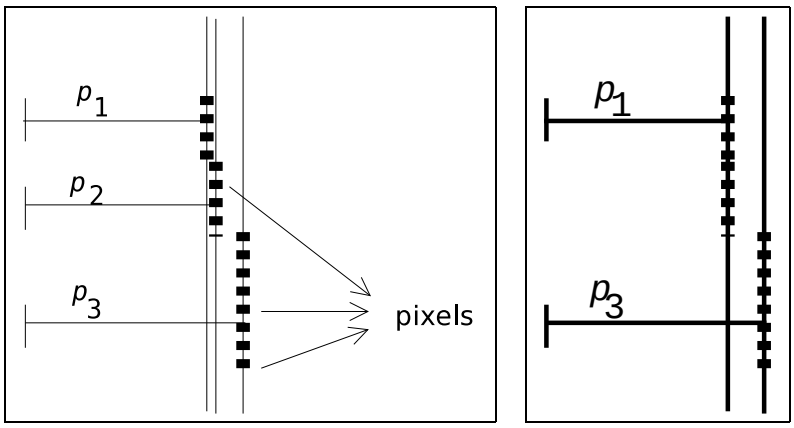
As can be seen in figure 3.4(a), there are three different pixel groups, in terms of ρ values. All the three pixel groups have the same θ value but different ρ values (ρ_1, ρ_2, ρ_3). If the difference between each ρ is smaller than or equal to a tolerance value (TOL) then, these ρ values may be assigned to a single ρ value and the pixels having these ρ values are merged. In figure 3.4(a), $\mid \rho_1 - \rho_2 \mid \leq TOL$ therefore, the pixels having the value of ρ_2 are assigned to ρ_1 and these pixels are merged (Figure



(a) (b)

Figure 3.3: The effect of small and large TINC values

3.4(b)). If $\mid \rho_1 - \rho_3 \mid > TOL$ then, the pixels on these edges are not merged into a single straight line (Figure 3.4(b)).



(a) (b)

Figure 3.4: The effect of TOL value

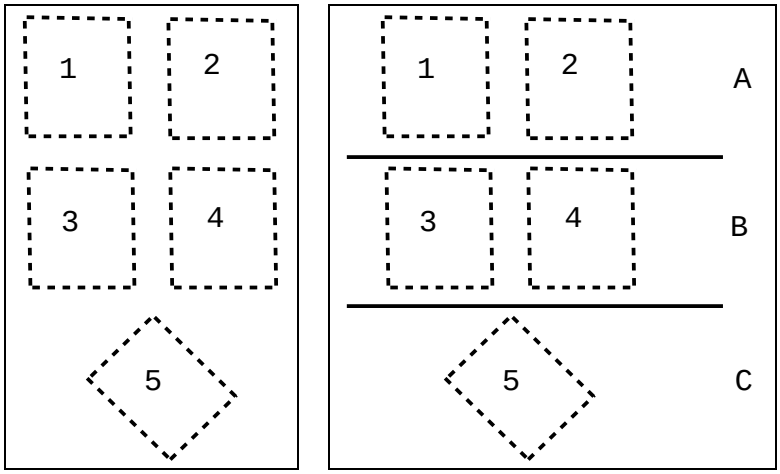
Applying a Hough transform to whole image may not always give satisfactory results. When there are buildings in different orientations (orientation means the directions of building edges), some of the buildings may not be detected correctly. If

the edges that have different orientations are represented in Hough accumulator array with small quantities, (they are under the threshold(T) value) then, these edges are not represented as straight lines. In order to preserve these edges in the final image, the input image was divided into meaningful and uniform pieces. Then, the Hough transform was applied to these pieces one-by-one. Each piece has its own Hough accumulator array and the parameters (Table 3.1) that affect each piece independently.

The edges represented in Global (accumulator of the whole image) Hough accumulator array with small numbers are generally represented with larger numbers in Local (accumulator of the piece) Hough accumulator. Using this way more buildings can be detected ([24]). In addition, the division process increases the speed of the processing. In Figure 3.5, an example is shown for the division process . In the figure, the orientation of building 5 is different from the others. Therefore, the pixels forming this building is represented with small quantity in the Global Hough accumulator array. In order to preserve this building in the final image, the input image is divided into 3 pieces (A,B,C) in y direction (Figure 3.5(b)). Then, the Hough transform is applied to each piece independently.

To divide the image into pieces, the XDIV and YDIV parameters were used. The XDIV parameter divides the image in x direction while the YDIV parameter divides the image in y direction. After applying the Hough transform to all the pieces and finding the edges, the pieces are merged and the perceptual grouping is applied to the whole image. The pieces are merged because an edge may cross over the neighboring pieces and therefore, the edge must be merged to make it complete. Care must be

taken while dividing the image into pieces. If the image is divided into too many pieces then, the meaningful straight line segments may not be found due to the noise. As well, some of the edges may be lost during the division and the merging processes.



(a) (b)

Figure 3.5: The effect of division of image

3.1.4 Perceptual Grouping

The goal of the perceptual grouping procedure is to organize image primitives into higher level primitives and thus, to explicitly represent the structures contained in the image. In this study, a *model based perceptual grouping* is proposed for grouping the straight line segments. In the present case only the rectangular shaped building models are taken into consideration. The aim is to group the straight line segments based on the model. Perceptual grouping is composed of two levels. In the first level, the pixels are grouped to find the straight line segments. The grouping of the pixel values are carried out according to the (θ, ρ) values found in the Hough transform step. In the next level, the straight line segments are grouped based on the model(s)

and the buildings are detected.

3.1.4.1 Finding The Straight Line Segments

The outputs of the Canny edge detector and the Hough transform are merged. Every (θ, ρ) pair found in the Hough transform is traced and those edge pixels from the output of the Canny edge detector that satisfy the (θ, ρ) are taken. Then, starting form the first pixel it is walked on these pixels. The first pixel is the starting point of the first straight line. It is then continued with the next pixel. If the distance between the consecutive pixels is below the preset threshold then, it is said that these pixels belong to the same straight line. Otherwise, the last point is the starting point of the next straight line and the pixel before the last pixel is the ending point of the previous straight line. By this way all the (θ, ρ) pairs were traced and the start and the end points of the straight lines were found.

This process is illustrated in figure 2.5 (Chapter 2) which shows the output of the Canny edge detector that contain the pixels with the same (θ, ρ) value. It was walked on these pixels and the straight line segments were generated. If the distance (gap) between the two consecutive pixels was below or equal to a threshold value (DIST) then, these pixels were considered belonging to the same straight line segment. In figure 2.5(a), the gap between P1 (pixel1) and P2 (pixel2) is smaller than or equal to DIST. Therefore, L1 and L2 are merged into a single line (L1). In contrast, the gap between P3 (pixel3) and P4 (pixel4) is greater than DIST. Therefore, L2 and L3 are not merged. As a consequence, the two straight line segments (L1 and L3) (Figure 2.5(b)) having the same (θ, ρ) value are obtained. This is the *proximity and collinearity* principles [25] of the perceptual grouping. The gaps like those between

P1 and P2 (Figure 2.5(a)) may occur because of the noise or the shadow effects.

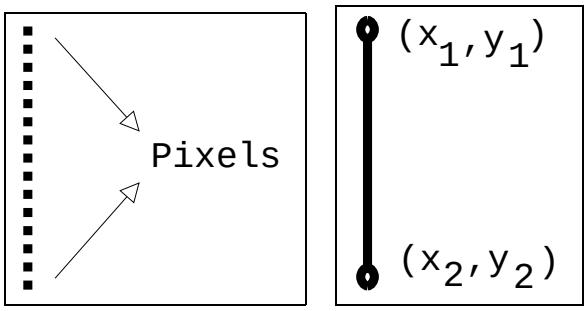
While grouping the pixels into straight line segments, the lengths of the straight line segments should be checked. The line segments with a length smaller than a threshold value are eliminated because they do not represent a building edge and therefore they are treated as noise. The MINLEN parameter is used as the threshold value for selecting the straight line segments which are the building edge candidates. The lengths of the line segments are in units of pixels. Therefore, MINLEN is the minimum number of pixels necessary to represent an edge of a building. The MINLEN parameter is calculated using the spatial resolution of the input image. For example, let us assume that the length of a building edge is 10 meters and the spatial resolution (pixel size) of the aerial image comprising this building is 50 cm. Then, edges of this building are represented by 20 pixels (Equation 3.2) on the aerial image. Therefore, in order to detect this building the MINLEN parameter should be set to 20.

$$MINLEN = L/SR \tag{3.2}$$

L: The length of a building edge

SR: Spatial resolution of the aerial image

The effect of the MINLEN parameter in straight line segment generation is shown in figure 3.6. In figure 3.6(a), there are several pixels having the same (θ, ρ) value. If the number of these pixels are bigger than the MINLEN then, these pixels are converted into a straight line segment as shown in figure 3.6(b). Dimensions of the buildings could be different from each other. Therefore, in order to detect the buildings that have smaller dimensions than other buildings, the value of the MINLEN



(a) (b)

Figure 3.6: The effect of the MINLEN parameter in straight line segment generation

parameter could be reduced. However, reducing the MINLEN parameter may cause to detect noisy parts as building edges. If the value of the MINLEN parameter is increased in order to remove the noise then, the building edges with smaller lengths may not be detected. Therefore, it may not always be possible to find an optimal MINLEN value.

3.1.4.2 Grouping The Straight Line Segments

The straight line segments detected in the previous level (Section 3.1.4.1) are now grouped based on the rectangular building model. The corners are the key elements while grouping the straight line segments. If two straight line segments are perpendicular or nearly perpendicular to each other and the distance between one of their end points is smaller than a threshold value then, these straight line segments form a corner (Figure 2.6). The angle between each line segment is found by using the θ values obtained in the Hough transformation stage. The two straight line segments forming a corner are put in the corner array and then, these corners are grouped into rectangular buildings.

While grouping the corners, the symmetry [25] and the parallelism are used. Figure 2.7 (Chapter 2) illustrates the perceptual grouping of symmetric corners. Corner 1 and corner 2 are symmetric to each other and $\mid P1-P2 \mid \leq DIST$ and $\mid P3-P4 \mid \leq DIST$ (Figure 2.7(a)). Therefore, these two corner groups form a rectangular building (Figure 2.7(b)).

In parallelism, two corners share the same edge and other edges of the corners are parallel to each other. The concept of the parallelism is illustrated in figure 2.8 where the edges 1-2 and 2-3 form two corners and edge 2 is the common edge for these two corners. Therefore, the edges 1 and 3 are parallel to each other. As a consequence, these edges construct a building (Figure 2.8(b)).

CHAPTER 4

THE IMPLEMENTATION

The proposed approach was implemented using a program written in C programming language. To implement the Hough transform and the perceptual grouping, more than 1000 lines of C code was written (Appendices A and B). The enhancement of the input image is performed with the GIMP (GNU Image Manipulation Program). The noise removal and the Canny edge detection stages were performed with the Matlab software. Currently, the developed C program can only read the images of type png (portable network graphics). Therefore, the output of the Canny edge detector should be an eight-bit gray level "png" image.

The developed C program was applied to selected four study areas. The study areas are different from each other in terms of building orientation, the amount of the cast shadows, and the number and the distribution of the buildings. The properties of each study area are described in the following sections. For each study area, there are four images; (i) the original input image, (ii) the Canny applied image, (iii) the Hough transform applied image(s), and (iv) the perceptual grouping applied image(s). For

the second and the fourth study areas, the Hough transform was applied twice using different parameters. The Hough transform parameters that belong to each study area are shown in tables 4.1, 4.2, 4.3, and 4.4.

4.1 Study Area 1

4.1.1 Description of the study area

The image (Figure 4.1) is a part of an aerial image of Batıkent in Ankara, Turkey.

Figure 4.1: The original image of the study area

The study area is surrounded by the roads. The lower left of the study area is covered by an open land. Although the effect of the shadow is apparent in the image (Figure 4.1), the shadow effect is not quite clear in the Canny output provided in figure 4.2. Because of the sun illumination effect, the roofs of the buildings seem to be divided into two parts as if there are two buildings. The buildings in the image have nearly the same orientation (homogeneous). The orientation of the buildings is important for the success of the Hough transform. In homogeneous areas, good results

are expected from the Hough transform. The effect of the homogeneity of building orientation is seen more apparently in the Hough transform of the image provided in figure 4.3. The edges are in SE (south-east)- NW (north-west) direction and the edges perpendicular to these edges are in SW (south-west)-NE (north-east) direction. The distances between the buildings are enough to discriminate the buildings from each other.

4.1.2 Building Detection

The Hough transform parameters used are given in table 4.1. Since the output of the Canny edge detector (Figure 4.2) is not much noisy the Hough transform parameters were selected low in order not to miss most of the buildings. In the output of the Canny edge detector, one side of the buildings seem to be shorter than the other side. Therefore, the MINLEN was set low enough to catch the short sides and high enough not to include the noisy parts. The XDIV and YDIV values were set to 1. This means that the image was not divided into sub cells. The image was not divided because the area is homogeneous by means of building orientation. The outputs of the Hough transform and the perceptual grouping are given in figures 4.3 and 4.4 respectively.

Table 4.1: The Hough Transform Parameters

Parameter	Value
MINMULT	0.35
MINLEN	15
TINC	1
TOL	1.1
XDIV	1
YDIV	1

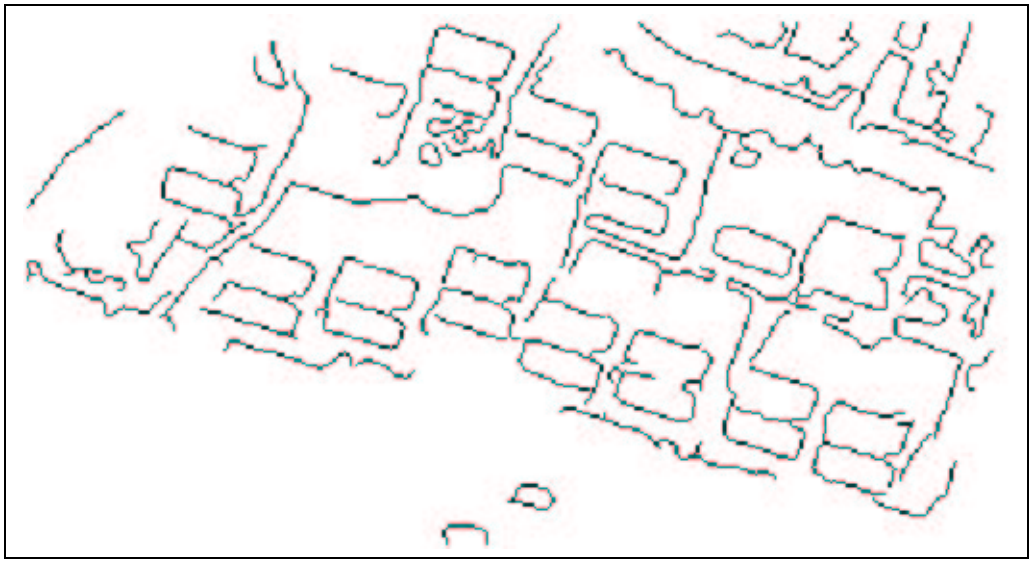


Figure 4.2: The output of the Canny edge detector

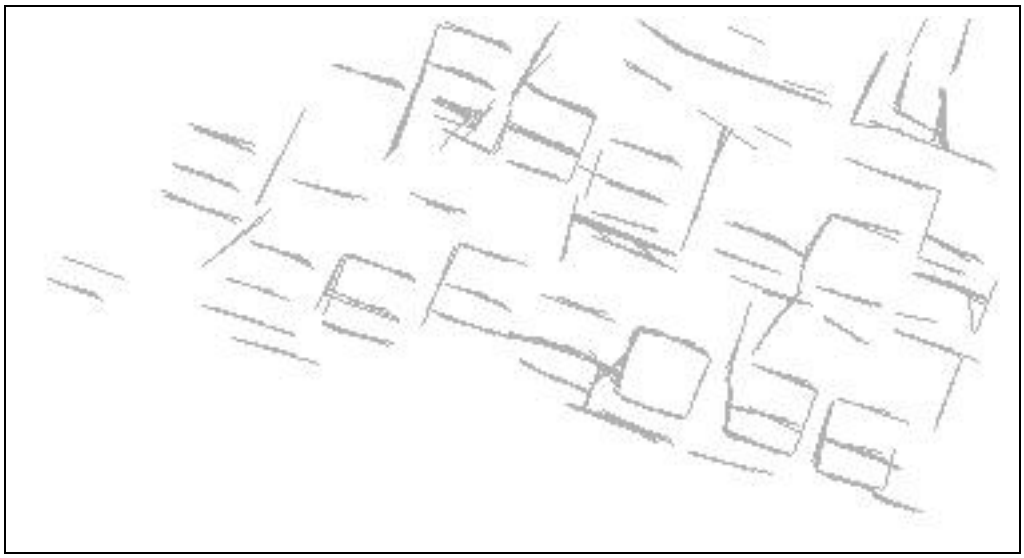


Figure 4.3: The output of the Hough transform

4.1.3 Results

As mentioned earlier, the success of the proposed method is affected by the quality of the input image and the success of each stage followed in the method. In this study area, some of the buildings were not detected, some of them were partially detected and some of them were successfully detected (Figure 4.4). As can be seen in the figure, there were some noise and false positives.

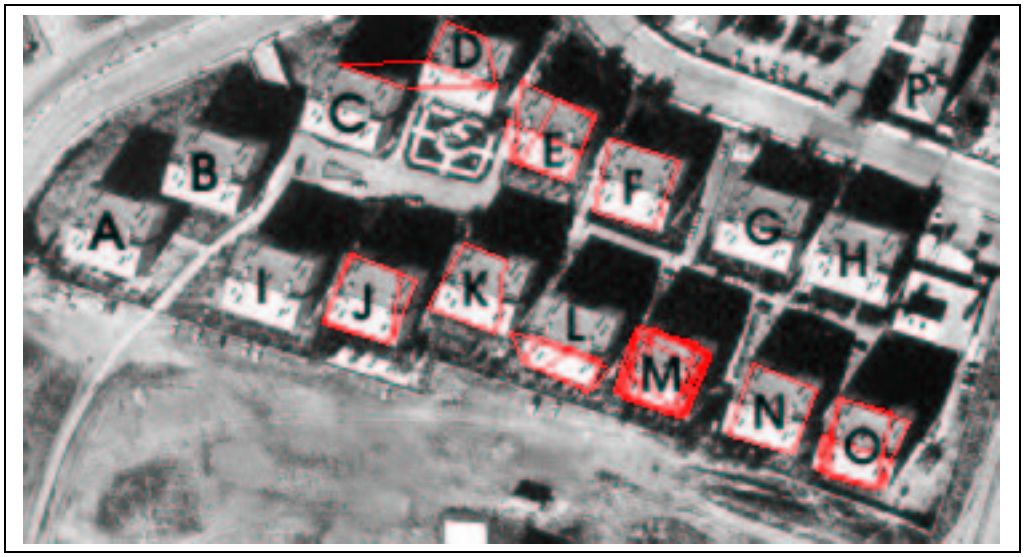


Figure 4.4: The result of perceptual grouping overlaid with the input image

Buildings A, B, C, G, H, I and P were not detected. Of these buildings, A and C were missed, because they were not represented in the output of the Canny edge detector (Figure 4.2). Buildings B, I and G were not detected because in the Hough transform stage only the parallel edges were detected. However, at least two corners are required in order to detect a building accurately. In figure 4.5, the edges of building B are shown in detail (the image is scaled by 2). As can be seen in the figure that building B is represented with three straight line segments parallel to each other. However, these line segments are not enough to detect a building using a model based perceptual grouping. Therefore, straight line segments perpendicular to these parallel line segments were not detected. This is because the (θ, ρ) pair representing these perpendicular line segments were represented in the Hough accumulator with values smaller than the threshold.

Three edges of building H can be seen visually in the output of the Hough transform (Figure 4.3). But, the distance between the bottom edge and the left edge of

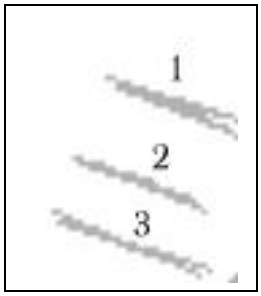


Figure 4.5: Building B after Hough transform

the building is greater than the threshold value (DIST). It appears that these edges were not connected in the perceptual grouping stage. The DIST parameter was used for discriminating two neighboring buildings from each other. If this value was set to a large value then, the two neighboring buildings would be detected as one single building.

Building P was missed in the perceptual grouping stage. The angle between the straight line segments 1 and 2 (Figure 4.6) are not in the range of $90^o \pm$ AT (AT: Angle threshold). Therefore, these two straight line segments were not grouped and as a result, building P was not detected by the proposed method.

Figure 4.6: Building P after Hough transform

Buildings D and L were partially detected. Lower part of building D and upper part of building L were not detected because there are missing straight line segments that are needed to form a corner. The straight line segments 2, 3 and 4 (Figure 4.7) were grouped into a building in perceptual grouping stage. However line segment 1

cannot be grouped with the other line segments.

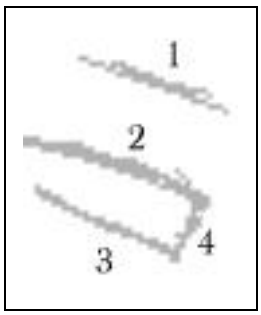


Figure 4.7: Building L after Hough transform

Some noise has occurred while detecting the lower part of building L. This is because 2, 3 and 4 (Figure 4.7) are not one pixel wide straight line segments. Therefore, these line segments were used more than once in the perceptual grouping stage.

Buildings E, F, J, K, M, N and O were correctly detected. Except buildings F and N, some noise has occurred while detecting the buildings correctly. The cause of the noise is the straight line segments which have widths larger than one pixel (thick straight line segments). In fact, the thick straight line segments are composed of more than one straight line segment. Therefore, these thick line segments that cause noise are used more than once in the perceptual grouping stage.

4.2 Study Area 2

4.2.1 Description of the study area

The image (Figure 4.8) is a part of an aerial image of Batıkent in Ankara, Turkey. The image is a nadir view of the area. Because of the sun angle, the shadows of the buildings are quite clear in the image. Except for small regions situated in the upper left, the lower left and the lower right of the study area, the area is full of the same

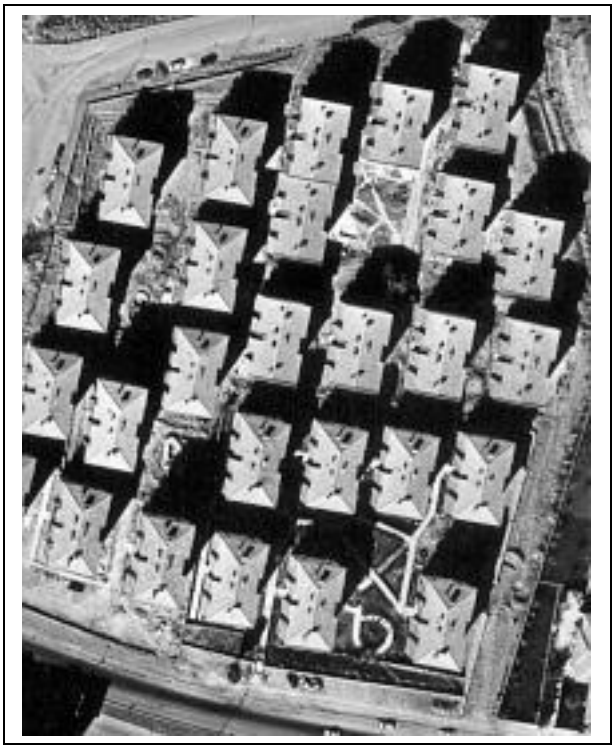


Figure 4.8: The original image of the study area

type of buildings. The buildings are near to each other, which may make the building detection process more difficult.

4.2.2 Building Detection

The Hough transform was applied twice to this study area with different parameters. The parameters are provided in table 4.2. In this study area, the orientation of the buildings are similar, however the orientation of the cast shadows are different from the building orientation. Therefore, the input image was divided into 9 cells (see section 3.1.3 for the purpose of image division) by setting XDIV=3 and YDIV=3. The Hough transform parameters in table 4.2 differ only in the TINC value. Therefore, the effect of TINC would be seen in the output images.

The outputs of the Canny edge detector, the Hough transform and the perceptual

Table 4.2: The Hough Transform Parameters

Parameter	Value1	Value2
MINMULT	0.30	0.30
MINLEN	15	15
TINC	3	1
TOL	1.1	1.1
XDIV	3	3
YDIV	3	3

grouping are provided in figures 4.9, 4.10, 4.11, 4.12, 4.13 respectively.

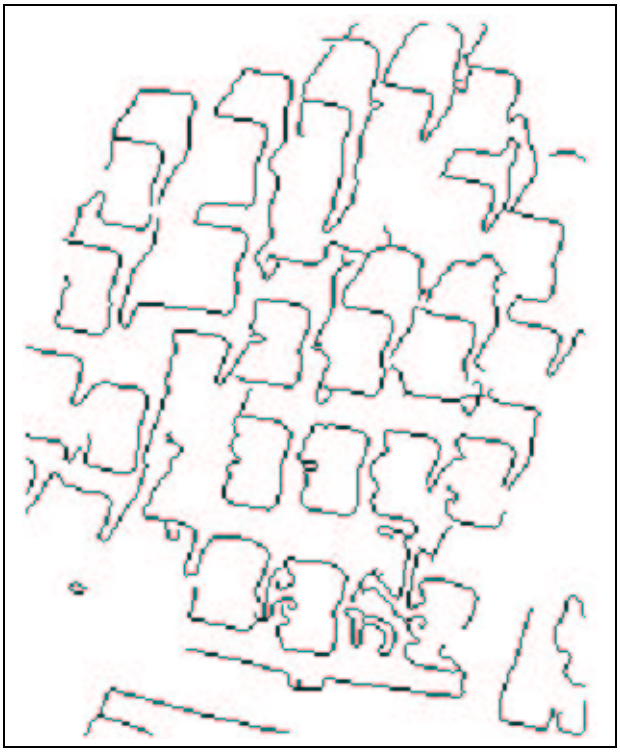


Figure 4.9: The output of the Canny edge detector

4.2.3 Results

To see the effect of TINC explicitly, the Hough transform was applied to the study area using TINC=3 and TINC=1. By decreasing the value of TINC, the number of the detected buildings increased. But, the decrease in the value of TINC also increased

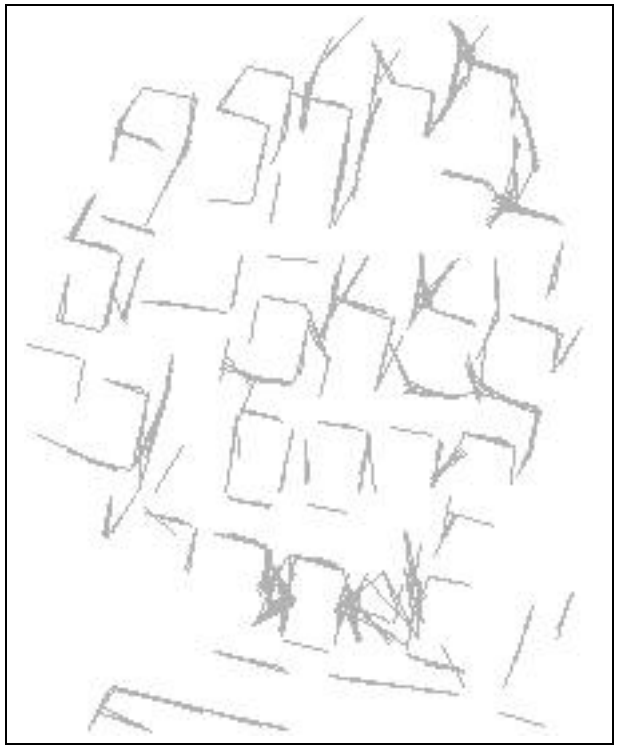


Figure 4.10: The output of the Hough transform (TINC=3)

Figure 4.11: The result of perceptual grouping overlaid with the input image (TINC=3)

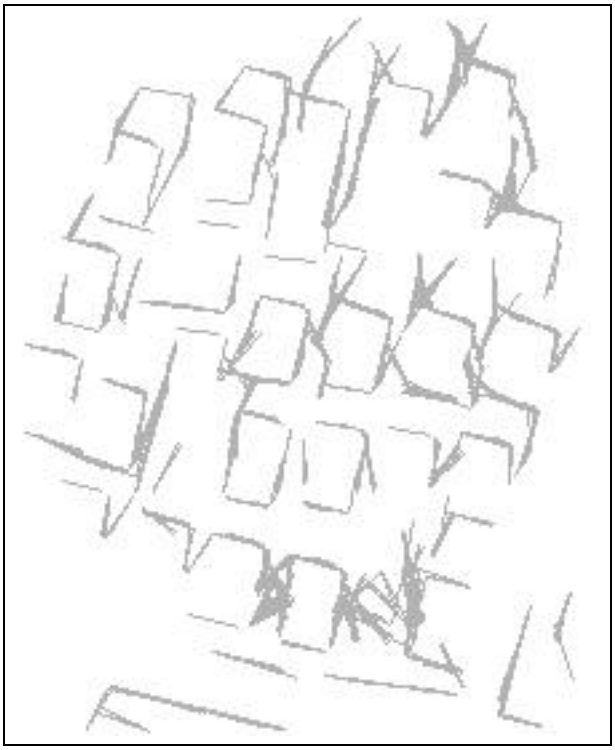


Figure 4.12: The output of the Hough transform (TINC=1)

Figure 4.13: The result of perceptual grouping overlaid with the input image (TINC=1)

the noise. The effects of changing the value of TINC can be seen in figures 4.11 and 4.13. The results can be examined using both of these figures.

4.2.3.1 First case (TINC=3)

As can be seen in figure 4.11, there are undetected, partially detected and correctly detected buildings. Some of the undetected buildings were detected after changing the value of the TINC parameter. Unfortunately, buildings A, C, E, F, G, H, I, K, O, R, U, V, W, X, Z1, Z2 were not detected. As stated before, these buildings might have been missed at one of the processing stages. Therefore, the cause of the failures should be examined closely and then the necessary precautions should be taken.

Buildings C, E, I, U, Z1 and Z2 were not detected by the Canny edge detector. Because the proposed method is an edge-based method there is no way to detect the buildings which are not detected during the edge detection stage. Buildings A, F, G, H and R were detected by the Canny edge detector. However, these buildings were missed in the Hough transform stage. Of these buildings, the edges of A, G and H were lost during the Hough transform stage. These edges were lost because their orientations were a little bit different than the other edges. Therefore, (θ, ρ) pairs representing these edges were represented with the small values in the Hough accumulator array and left under the threshold value.

Building R has only one corner after the Hough transform. The straight line segments 1 and 2 (Figure 4.14) form a corner. The distance between the end points of the line segments 3 and 2 is greater than the threshold value (DIST). Therefore,

these two line segments do not form a corner. Since building R has only one corner it was not detected, therefore.

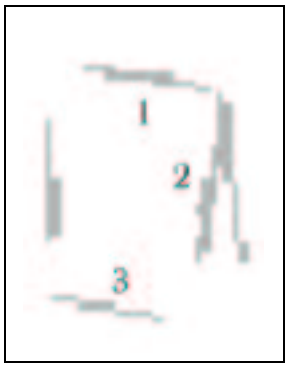


Figure 4.14: Building R after Hough transform

It is assumed that the angle between two neighboring edges of a building is in the range of [90^o-AT , 90^o+AT] (AT is the angle threshold). In figure 4.15, the angle between line segments 1 and 2 is not in the desired range. Therefore, these edges were not grouped as a building corner in the perceptual grouping stage and as a result, building F was not detected.

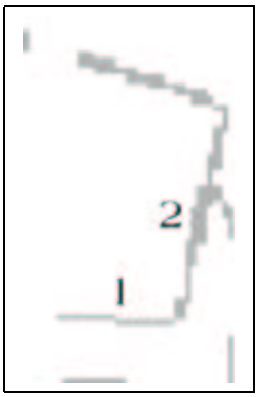


Figure 4.15: Building F after Hough transform

The angle threshold (AT) value was determined using TINC. The angle threshold value used in the tests is TINC*3. Therefore, the angle between the two neighboring edges of a building should be in range of 87 and 93 for this study area. This range is calculated as follows:

TINC = 1 (Value2 column of table 4.2)

AT = TINC*3

AT = 3

range => [90-3 , 90+3] => [87,93]

Buildings L, N, P, T and Y are partially detected. The buildings in this study area are located close to each other. As well the shadows of the buildings cast onto other buildings. Therefore, there is some noise in the partially detected buildings. Buildings L and P are partially detected. While detecting these buildings, noise occurred because they are close to each other and they are surrounded by the cast shadows.

Building N was also partially detected. However, building N was detected using the edges of the neighbor building I. In figure 4.16, the line segments 1 and 2 form a corner. The line segments 1 and 3 also form a corner because the distance between the end points (black points) of these two line segments is smaller than the threshold value (DIST). Then, using the principle of parallelism, these three line segments (1, 2, 3) are grouped into a building.

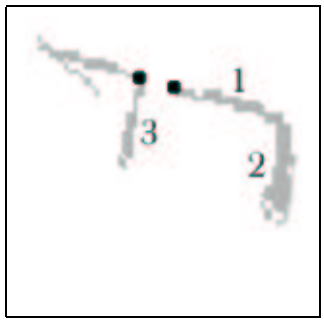


Figure 4.16: Building N after Hough transform

Buildings B, D, M and S were detected correctly. Although there is some noise,

buildings B, D and M were detected without any problem. Building S was also detected correctly but there is some noise inside and around this building, which is caused by the noisy line segments detected by the Hough transform.

Some of the regions were detected wrongly as buildings. For example, the shadow regions above building F and building Z1 were detected as buildings. With the proposed method, there is no way to eliminate the false positives because only the straight line segments were used to detect the buildings. Neither the shadow nor the intensity information were used in the method.

4.2.3.2 Second case (TINC=1)

By decreasing the value of TINC, more straight line segments could be detected. Therefore, in order to detect the missing buildings, the value of TINC was decreased. The decrease of the value of the TINC parameter would increase the noise. In the first case (TINC=3), buildings A, O, V and Z2 were not detected. After decreasing the value of TINC, buildings A, O and V were partially detected while building Z2 was correctly detected. In addition, those buildings that were partially detected in the first case (L, N and P), were correctly detected in this case.

The increase of the noise can be clearly seen if the figures 4.11 and 4.13 are compared. For example, there is much more noise in building M in figure 4.13 than in figure 4.11.

4.3 Study Area 3

4.3.1 Description of the study area

The image (Figure 4.17) is a part of an aerial image of Batıkent in Ankara, Turkey. In this study area, the buildings are grouped around empty regions. Some of the buildings are single detached buildings and some of them are stuck to each other. Also the orientations of the buildings differ from each other. The sun illumination cast shadows in the direction of the illumination.

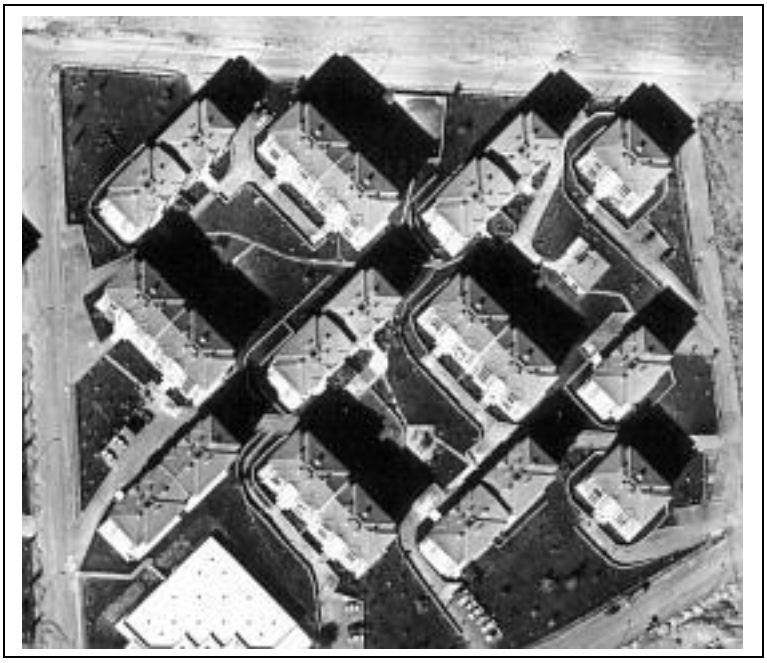


Figure 4.17: The original image of the study area

4.3.2 Building Detection

Since the orientations of the buildings differ from each other the input image was subdivided into cells in order to detect as many buildings as possible. The Hough transform parameters used are given in table 4.3. The values of the parameters given in the table were found to be optimal after several trials. The outputs of the Canny

edge detection, Hough transform and the perceptual grouping are provided in figures 4.18, 4.19, 4.20 respectively.

Table 4.3: The Hough Transform Parameters

Parameter	Value
MINMULT	0.25
MINLEN	15
TINC	3
TOL	1.1
XDIV	4
YDIV	3

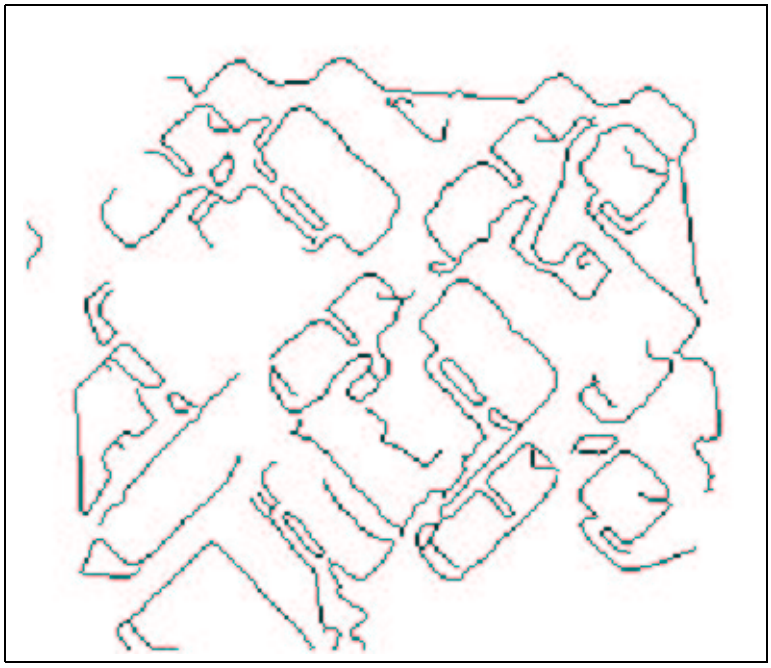


Figure 4.18: The output of the Canny edge detector

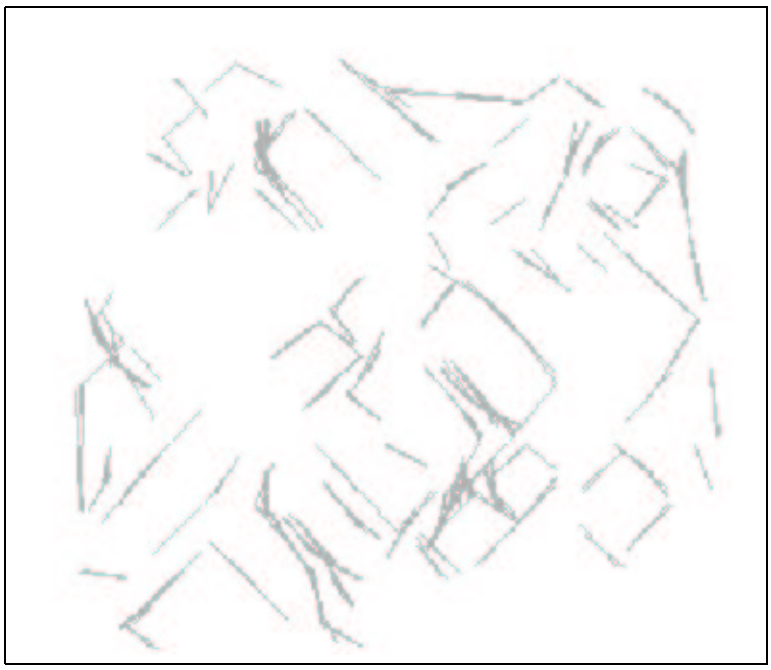


Figure 4.19: The output of the Hough transform

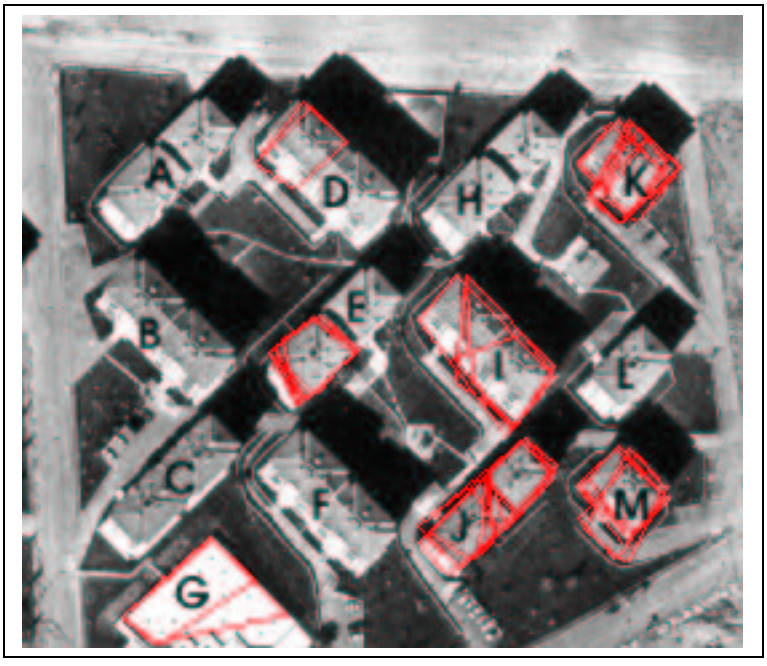


Figure 4.20: The result of perceptual grouping overlaid with the input image

4.3.3 Results

There are 17 buildings in the input image (Figure 4.17). Some of the buildings are attached to each other. Therefore, they are counted as a single building. The accuracy

of the proposed method was not as high for this study area as it was expected. The main reason for the failure is attributed to the Canny edge detection algorithm. The edges of several buildings were lost during the Canny edge detection stage. Also, there were noisy edges belonging to the objects that are in the open areas of the input image. Some of the building edges were lost and some noisy edges occurred in the Canny edge detection stage. This inevitably affected the result of the Hough transform negatively causing to miss some of the straight line segments (Figure 4.19). Although some of the building edges were interpreted correctly during the Canny edge detection stage, the edges were lost during Hough transform. This is because the pixels belonging to these edges were represented in the Hough accumulator array with lower values than the threshold. Because decreasing the threshold values would cause too much noise they were not decreased, therefore.

Buildings A, B, C, F, H and L were not detected. Of these buildings, B and L were missed because they were not represented by the Canny edge detector. Building A was missed during Hough transform. The image was divided into 12 cells (XDIV=4, YDIV=3). In the division and the merging phases, the edges of building A was lost and therefore, it was missed. Buildings C and F were represented with only two parallel straight line segments after the Hough transform. One of the short edges of these building was lost by the Canny edge detector stage, while the other was lost during Hough transform. The existance of two parallel lines is not enough to detect a building through a model based perceptual grouping. For this reason, buildings C and F were missed. Buildings D and E were partially detected because they were partially represented after the Hough transform. Buildings G, I, J, M and K were correctly

detected but some noise occurred in the output. This is because of the thick straight line segments obtained in the Hough transform stage.

4.4 Study Area 4

4.4.1 Description of the study area

The image (Figure 4.21) is a part of an aerial image of Batıkent in Ankara, Turkey. In this study area, the buildings are sparse and scattered. The size and the orientation of the buildings are generally different from each other. Most of the study area is covered by the open lands. The cast shadows and the roads may make it difficult to detect the buildings correctly.

Figure 4.21: The original image of the study area

4.4.2 Building Detection

Since the size and the orientations of the buildings are generally different form each other the MINLEN, XDIV and YDIV parameters are expected to affect the results. To see the effect of the division of the image, two configurations were used for the

Hough transform parameters in this study area. The Hough transform parameters used are shown in table 4.4.

Table 4.4: The Hough Transform Parameters

Parameter	Value1	Value2
MINMULT	0.25	0.25
MINLEN	15	15
TINC	1	1
TOL	1.1	1.1
XDIV	3	1
YDIV	3	1

The outputs of the Canny edge detection, Hough transform and the perceptual grouping are provided in figures 4.22, 4.23, 4.24, 4.25 and 4.26 respectively.

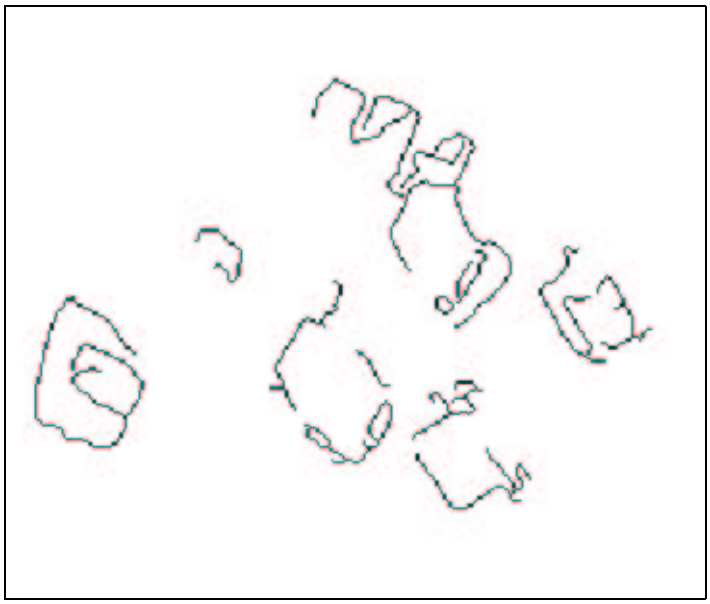


Figure 4.22: The output of the Canny edge detector

Figure 4.23: The output of the Hough transform (XDIV=3, YDIV=3)

Figure 4.24: The result of perceptual grouping overlaid with the input image (XDIV=3, YDIV=3)

Figure 4.25: The output of the Hough transform (XDIV=1, YDIV=1)

Figure 4.26: The result of perceptual grouping overlaid with the input image (XDIV=1, YDIV=1)

4.4.3 Results

For this study area, the final output was mostly affected by the Canny edge detector. Buildings B and G (Figures 4.24 and 4.26) were missed during the Canny edge detection stage (Figure 4.22). Therefore, there is no way to detect these buildings in the subsequent stages followed by the Canny edge detection stage. Building C (Figure 4.26) was represented by the Canny edge detector. However, this building was eliminated in the Hough transform because the sizes of the edges of this building were smaller than MINLEN value. By decreasing the value of the MINLEN it would be possible to represent building C in the final output. However, this would cause considerable noise to appear in the final output.

The effect of dividing the image into regular cells is evident in building I. If the input image was divided into 9 cells by setting the XDIV and YDIV parameters to 3 then, building I would be detected in the final output (Figure 4.24). Without dividing the study area into cells, it would not be possible to detect building I (Figure 4.26).

4.5 Discussion

Because the proposed method is hierarchic the output of each stage affects the success of the next stage directly. Therefore, stating that the success of the method is directly related to the quality of the image is not wrong. First of all, the quality of the aerial image is very important. The contrast of the image should be high enough to discriminate the buildings from the background. The effects of the atmospheric conditions should be minimum in order to get a clean image. If the image is of good quality, there is not much thing to do in the noise removal stage.

The Canny edge detection stage is also very important for the success of the methodology. In the selected study areas given in this thesis, some of the buildings were lost during the Canny edge detection stage. Especially in study area-4, nearly half of the buildings were not represented after the Canny edge detection stage was applied. Therefore, these buildings were not detected in the final output.

The Canny edge detection parameters (see section 2.2.1) determine the success of the Canny edge detector output. By adjusting these parameters, more buildings can be represented in the Canny edge detection output. But unwanted noise may occur in the Canny edge detection output. The Hough transform is quite sensitive to the noise. Therefore the unwanted noise present in the Canny edge detection stage may cause noisy figures in the Hough transform output. In addition, this unwanted noise may be represented as false negatives in the Hough transform output.

The method used in this thesis is a parametric method. Therefore, the parameters used in each stage affect the result directly, either negatively or positively. It is therefore quite important to find the optimal values for the parameters. The effects of the Hough transform and the perceptual grouping parameters were examined in this chapter in the selected four study areas.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

In this thesis, a method was proposed to detect the buildings automatically from high resolution aerial images. The Hough transform, which is used for detecting structures that have parametric representation like linear line, circle and ellipse, was applied for vectorizing the edge pixels. The use of the Hough transform in building detection is not very common. In the previous studies, the Hough transform was generally used in the detection of a single object form an input image. In this thesis, the Hough transform was applied to high resolution images that contain residential buildings.

The Hough transform generated reasonable results in those images where the building edges were extracted well by the Canny edge detector, which provided relatively noise free outputs. The parameters of the Canny edge detector were set, as far as possible, to optimum values to preserve as many edges as possible while reducing the noise as much as possible. In the noisy images, the Hough transform did not produce the desired results. If there is too much noise in the output of the Canny edge detector, the parameters of the Hough transform should be set to strict values in order

to reduce the negative effects of the noise. However, setting the parameters to strict values may cause to miss too many buildings in the final output. It is therefore, not suitable to apply the Hough transform to noisy images.

The Hough transform is not an efficient algorithm in terms of the CPU (central processing unit) and memory usage. All of the edge pixels in the Canny edge detector output are traced for each θ value and these results are stored in the Hough accumulator array. If there are too many edge pixels in the Canny edge detector output, the processing time and the memory usage of the Hough transform increase. Dividing the input image uniformly into pieces increase the performance of the method and decrease the processing time. The division of the image into regular cells puts an overhead to the implementation of the method and, in some situations, the result is affected negatively. The pieces of the outputs of the Hough transform should be merged before the perceptual grouping stage. Some of the straight lines may be lost in the division and the merging process which may result in missing some of the buildings present in the input image.

The model based perceptual grouping implemented in this study produced reasonable results where the buildings were in shape of the model. The shapes of the buildings must not be completely the same as the shapes of the model. The model is flexible up to a degree. In aerial images where the buildings resemble to the model, successful results were generated.

The Hough transform and the perceptual grouping algorithms are parametric algo-

rithms. It is generally not easy to find the optimal parameters and generate successful results. The analyst who applies the method to high resolution images should change the parameters until the satisfactory results are found.

In this thesis only the rectangular models were used. It is planned to increase the number of the models and to apply the method on aerial images that contain more complex shaped buildings. On the other hand, only the edge information (one source of information) was used to detect the buildings. In the automatic building detection methods, a single source of information may not be enough to detect the buildings accurately and to remove the false positives. It is therefore recommended that new information such as, the intensity, the shadows, the heights of the buildings, etc. should be included in the processings.

REFERENCES

[1] Z. Kim, A. Huertas, and R. Nevatia, "A model-based approach for multi-view complex building description," Institute for Robotics and Intelligent Systems, University of Southern California, 2001.

[2] J. Li, R. Nevatia, and S. Nornoha, "User assisted modeling of buildings form aerial images," Institute for Robotics and Intelligent Systems, University of Southern California Los Angeles, California 90089, 1999.

[3] S. C. Lee, A. Huertas, and R. Nevatia, "Modeling 3-d complex buildings with user assistance," Institute for Robotics and Intelligent Systems, University of Southern California Los Angeles, California 90089, 2000.

[4] S. Heuel and R. Nevatia, "Including interaction in an automated modelling system," Institute for Robotics and Intelligent Systems, University of Southern California Los Angeles, California 90089, IEEE, 1995.

[5] 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.

[6] V. K. Shettigara, S. G. Kempinger, and R. Aitchison, "Semi-automatic detection and extraction of man-made objects in multispectral aerial and satllite images," Defence Science and Technology Organisation PO Box 1500 Salisbury SA 5108 Australia, 1995.

[7] A. Huertas, R. Nevatia, and D. Landgrebe, "Use of hyperspectral data with intensity images for automatic building modeling," Institute for Robotics and Intelligent Systems, University of Southern California Los Angeles, California 90089 and Electrical and Computer Engineering Purdue University West Lafayette, Indiana 47907-1285, 1999.

[8] R. T. Collins, A. R. Hanson, E. M. Riseman, and H. Schultz, "Automatic extraction of buildings and terrain from aerial images," Department of Computer Science Lederle Graduate Research Center University of Massachusets Amherst, MA USA 01003-4610, 1995.

[9] M. Bicego, S. Dalfini, G. Vernazza, and V. Murino, "Automatic road extraction from aerial images by probabilistic contour tracking," Dipartimento di Informatica, Università di Verona Ca' Vignal 2, Strada Le Grazie 15, 37134 Verona, Italy

and Dipartimento di Ingegneria Biofisica ed Elettronica (DIBE), Università di Genova Via all'Opera Pia 11A, 16145 Genova, Italy, IEEE, 2003.

[10] D. S. Lee, J. Shan, and J. S. Bethel, "Class-guided building extraction from ikonos imagery," vol. 69, pp. 143–150, Geomatics Engineering, School of Civil Engineering, Purdue University, West Lafayette, IN 47907-1284, Photogrammetric Engineering & Remote Sensing, February 2003.

[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] A. Huertas and R. Nevatia, "Detecting buildings in aerial images," vol. 41(2), pp. 131–152, Computer Vision, Graphics and Image Processing, 1988.

[13] C. Braun, T. Kolbe, F. Lang, W. Schickler, V. Steinhage, A. Cremers, W. Förstner, and L. Plumer, "Models for photogrammetric building reconstruction," vol. 19(1), pp. 109–118, Computers & Graphics, 1995.

[14] A. Katartzis, H. Sahli, E. Nyssen, and J. Cornelis, "Detection of buildings from a single airborne image using a markov random field model," Vrije Universiteit Brussel, ETRO department, Pleinlaan 2, 1050 Brussels, Belgium., 2001.

[15] T. Kim and J.-P. Muller, "Building extraction and verification from spaceborne and aerial imagery using image understanding fusion techniques," Department of Photogrammetry and Surveying University Colloge London Gower St., London WC1E 6BT, UK, 1995.

[16] R. Haala, "Combining multiple data sources for urban data acquisition," pp. 329–340, Wichmann Verlag, Heidelberg, Germany, Photogrammetric Week, 1999.

[17] W. Förstner and L. Pluemer, Semantic Modeling for the Acquisition of Topographic Information from Images and Maps, p. 277. Birkhaeuser Verlag, Berlin, Germany, 1997.

[18] A. Huertas, R. Nevatia, and D. Landgrebe, "Use of hyperspectral data with intensity images for automatic building modeling," Sunnyvale, California, Proceedings of the Second International Conference on Information Fusion, July 1999.

[19] D. McKeown, S. Cochran, S. Ford, C.McGlone, J. Shufelt, and D. Yocum, "Fusion of hydice hyperspectral data with panchromatic feature extraction," vol. 37(3), pp. 1261–1277, IEEE Transactions on Geoscience and Remote Sensing, 1999.

[20] W. Förstner, "3d-city models: Automatic and semiautomatic acquisition methods," pp. 291–303, Photogrammetric Week, 1999.

[21] N. Efford, Digital Image Processing: a practical introduction using Java, pp. 164–174. Pearson Education Limited, 2000.

[22] 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.

[23] P. Moore and C. Fitz, "Gestalt theory and instructional design," vol. 23(2), pp. 137–157, Technical Writing and Communication, 1993.

[24] A. Karnieli, A. Meisels, L. Fisher, and Y. Arkin, "Automatic extraction and evaluation of geological linear features from digital remote sensing data using a hough transform," vol. 62, pp. 525–531, The Jacob Blaustein Institute for Desert Research, Sede Boker Campus Ben-Gurion University of the Negev 84990, Israel and Geological Survey of Israel, 30 Malkhe Yisrael St., 95501 Jerusalem, Israel, Photogrammetric Engineering & Remote Sensing, May 1996.

[25] F. Ackermann, A. Maßmann, S. Posch, G. Sagerer, and D. Schlüter, "Perceptual grouping of contour segments using markov random fields," AG für Angewandte Informatik Universität Bielefeld Postfach 100131, 33501 Bielefeld Germany, 1997.

APPENDIX A

STRUCTURES USED IN THE SOURCE CODE

```
typedef struct {
    double r;   /* rho value */
    int num;    /* number of pixels satisfying the rho */
} r_num;


typedef struct {
    int theta;  /* theta value */
    r_num *rn;  /* corresponding r_num for theta */
    int diffr;  /* different rho values corresponding to theta */
} trn;


typedef struct {
    int x;  /* x-coordinate of a pixel */
    int y;  /* y-coordinate of a pixel */
} coord;
```

```
typedef struct { /* coordinates of pixels satisfying
                    the rho values */
    r_num rn;
    coord *c;
} rnc;


typedef struct { /* theta value and corresponding rho and
                    pixel coordinates */
    int theta;
    int diffr;
    rnc *myrnc;
} trnc;


typedef struct {
    int type;   /* type of the corner point */
    coord c[4]; /* start-end points of two lines
                   composing the corner */
} corner;


typedef struct {
    int type;   /* type of the building */
    coord c[4]; /* start-end points of two opposite lines
                   composing the building */
```

```
} building;
```

APPENDIX B

A PART OF THE CODE IMPLEMENTED IN THE STUDY

The following are the two functions used in the Hough transform code. The **sinlookup** and **coslookup** functions increase the performance of the code. For every pixel, $sin(\theta)$ and $cos(\theta)$ values are needed. Instead of calculating $sin(\theta)$ and $cos(\theta)$ values each time for each pixel, they are calculated once and kept in the memory of the computer and used as needed.

```
double *sinlookup(int t1,int t2,int tinc)
{
    double *sinarr;
    int i;
    sinarr=(double *)malloc(sizeof(double)*((int)((t2-t1)/tinc)+2));
    for(i=0;i<=(t2-t1);i+=tinc) {
        sinarr[i/tinc]=sin(D2R(t1+i));
```

```
    }
    return sinarr;
}

double *coslookup(int t1,int t2,int tinc)
{
    double *cosarr;
    int i;
    cosarr=(double *)malloc(sizeof(double)*((int)((t2-t1)/tinc)+2));
    for(i=0;i<=(t2-t1);i+=tinc) {
        cosarr[i/tinc]=cos(D2R(t1+i));
    }
    return cosarr;
}
```

The Hough transform parameters are kept in a file. The Hough transform can be applied by changing the parameters in this file without recompiling the source code. The following is an example Hough transform configuration file.

```
THETA1  0       # starting angle

THETA2  179     # ending angle

MINMULT 0.35    # multiplier used to find the threshold value in the

                # accumulator array

DIST    5.0     # the distance to separate two straight line segments

MINLEN  15      # the minimum straight line segment length

TINC    3       # constant value between consecutive theta's

TOL     1.1     # the deviation in the rho of a straight line

XDIV    2       # divide the input image into XDIV parts in x-axis

YDIV    3       # divide the input image into YDIV parts in y-axis
```