# DETECTION OF EARTHQUAKE DAMAGED BUILDINGS FROM POST-EVENT PHOTOGRAPHS USING PERCEPTUAL GROUPING

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

BY

MUHAMMET ALİ GÜLER

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 Karslığlu

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: Muhammet Ali Güler

Signature :

# ABSTRACT

DETECTION OF EARTHQUAKE DAMAGED BUILDINGS FROM POST-EVENT PHOTOGRAPHS USING PERCEPTUAL GROUPING

Güler, Muhammet Ali

MSc, Department of Geodetic and Geographic Information Technologies

Supervisor: Asst. Prof. Dr. Mustafa Türker

MAY 2004, 72 pages

Two approaches were developed for detecting earthquake damaged buildings from post-event aerial photographs using shadow analysis and perceptual grouping. In the first approach, it is assumed that the vector boundaries of the buildings are not known a priori. Therefore, only the post-event aerial photographs were used to detect the collapsed buildings. The approach relies on an idea that if a building is fully damaged then, it will not generate a closed contour. First, a median filter is applied to remove the noise. Then, the edge pixels are detected through a Canny edge detector and the line segments are extracted from the output edge image using a raster-to-vector conversion process. After that, the line segments are grouped together using a three-level hierarchical perceptual grouping procedure to form a closed contour. The principles used in perceptual grouping include the proximity, the collinearity, the continuity and the perpendicularity. In the second approach, it is assumed that the vector boundaries of the buildings are known a priori. Therefore, this information is used as additional data source to detect the collapsed buildings. First, the edges are detected from the image through a Canny edge detector. Second, the line segments are extracted using a raster-to-vector conversion process. Then, a two-level hierarchical perceptual grouping procedure is used to group these line segments. The boundaries of the buildings are available and stored in a GIS as vector polygons. Therefore, after applying the perceptual grouping procedure, the damage conditions of the buildings are assessed on a building-by-building basis by measuring the agreement between the detected line segments and the vector building boundaries.

Both approaches were implemented in a selected area of the city of Golcuk, which is one of the urban areas most strongly affected by the Izmit, Turkey, 1999 earthquake. The area contains a total of 282 buildings, of which 79 are collapsed and 203 are uncollapsed. The results of the first approach is satisfactory. Of the total 203 uncollapsed buildings, 168 were detected correctly. For the second approach, the overall accuracy was computed to be 72.6%. Of the total 79 collapsed buildings, 63 were detected correctly.

Keywords: Perceptual grouping, building, Canny, damage assessment, edge detection

# ÖZ

ALGISAL GRUPLAMA KULLANARAK DEPREM SONRASI HAVA FOTOĞRAFLARINDAN ÇÖKEN BİNALARI TESPİT ETMEK

Güler, Muhammet Ali

Master, Jeodezi Ve Coğrafi Bilgi Teknelojileri Bölümü

Tez Yöneticisi: Yard. Doç. Dr. Mustafa Türker

Mayıs 2004, 72 sayfa

Depremde hasar görmüş binaların gölge bilgisi ve algısal gruplama kullanılarak hava fotoğraflarından tespiti için iki yaklaşım geliştirildi. Birinci yaklaşımda, binaların dış hatlarının bilinmediği varsayılır. Bundan dolayı, hasarlı binaların tespiti için sadece deprem sonrası hava fotoğrafları kullanıldı. Yaklaşım bina tamamen yıkıldıysa kapalı bir alan oluşturmayacagı fikrine dayanır. İlk olarak parazitleri yok etmek için bir median filtresi uygulanır. Sonra bir Canny kenar tespit yoluyla kenar noktalar tespit edilir ve bir raster-vektör dönüştürme işlemi kullanılarak kenar resminden çizgi parçaları elde edilir. Bundan sonra, çizgi parçaları kapalı bir alan oluşturmaları için üç aşamalı hiyerarşik algısal gruplama kullanılarak gruplandırılır. Algısal gruplamada kullanılan ilkeler yakınlık, doğrusallık, süreklilik ve dikliktir. İkinci yaklaşımda bina dış hatlarının bilindigi varsayılır. Bu yüzden, bu bilgi ek veri kaynağı olarak kullanılır. Öncelikle kenarlar bir Canny kenar tespit yöntemiyle bulunur. İkincil olarak, çizgi parçaları bir raster-vektör dönüştürme işlemi yoluyla elde edilir. Sonra, iki aşamalı hiyerarşik algısal gruplama kullanılarak bu çizgi parçaları gruplanır. Bina dış hatları kullanılabilir durumdadır ve vektör veri olarak bir GIS'te depolanır. Bundan dolayı, algısal gruplama prosedürü uygulandıktan sonra, binaların hasar durumları bina dış hatları ve elde edilen çizgi parçaları arasındaki benzerliği ölçerek teker teker değerlendirilir.

İki yaklaşımda 1999 İzmit Turkiye depreminde en fazla hasar gören kentsel alanlardan biri olan Gölcük'ten seçilen bir alanda uygulanmıştır. Seçilen alan 79 yıkık 203 sağlam toplam olarak 282 bina içermektedir. Birinci yaklaşımın sonuçları yeterlidir. 203 sağlam binadan 168 tanesi doğru olarak tespit edilmistir. İkinci yaklaşım için doğruluk 72.6% olarak hesaplanmıştır. 79 tane yıkık binadan 63 tanesi doğru olarak tespit edilmiştir.

Anahtar Kelimeler: Algısal gruplama, bina, Canny, hasar, hasar yönetimi, kenar bulma

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 angle Derya for her patience and support during my study,

my colleagues Burak Bilen, 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

An earthquake is a series of shock waves generated following the brittle failure of rocks within the earth's crust or upper mantle as a result of a build up of stress. Many earthquakes occur every year, on average greater than 800,000, but most are small and not felt by the humans. A severe earthquake, with a magnitude of greater than 8.0, can be expected every 8 to 10 years. But a significant number of smaller earthquakes that are still capable of destruction occur almost every year.

On 17 August, 1999 a strong earthquake struck north-west of Turkey caused tremendous damages in urban areas of Golcuk, Yalova , Izmit, and Istanbul, killed about 17,000 people and damaged thousands of buildings. After such a strong and deadly earthquake, the rapid and reliable damage assessment and the estimation of lost is very important to organize the emergency teams. If the location and the severity of the damage are known and available to responsible agencies, the emergency teams can be organized immediately and sent to the effected areas.

For this purpose, the aerial images are quite suitable data sources that can be used for detecting the damaged buildings after a severe earthquake. The aerial images can be immediately taken after an earthquake over the area and supplied to a damage assessment system. Then, by using image processing techniques and GIS utilities, the damaged buildings can be detected. The GIS data which provide the vector boundaries of the buildings are usually available. After finding the damaged areas, the damage assessment can be carried out using the integrated analysis of vector building boundaries and the post-earthquake aerial imagery.

The main objectives of this study are as follows:

- to develop approaches for detecting the collapsed buildings due to earthquake from post-event aerial imagery,
- to develop perceptual grouping procedures in order to find the closed contours, the linear groups and the corners, and
- to illustrate the advantages of integrating vector data and raster imagery in order to detect the collapsed buildings.

## 1.1 Data Sets and The Development Platform

The proposed approach was implemented in the C programming environment installed on a Linux operating system (Redhat 8.0). Gnu C complier (gcc 3.2) was used to compile the sources. For the image reading and the manipulating processes, the png (portable network graphics) library (libpng-1.2.2-8) was used. The png is a format for storing compressed raster images.

Figure 1.1: The study area

The study area is illustrated in Figure 1.1. An aerial photograph acquired over Golcuk after the Izmit Earthquake was used to implement the algorithm. There are mainly man-made structures in the selected area. The spatial resolution of the photograph is 50 cm. The selected area contains 282 buildings, of which 79 are collapsed. The buildings are mostly rectangular shaped. Within the building blocks, the orientation of the buildings are approximately in the same direction. The image was taken on a sunny day. Therefore, most of the buildings have the cast shadows.

## 1.2 Organization of the Thesis

The chapters of the thesis have been organized as follows:

In chapter 2, the previous works on damage detection are listed and the techniques used in these works are explained briefly. Then, the algorithms used in this thesis to detect the collapsed buildings are given.

In chapter 3, the first approach is described. In this first case, it is assumed that the vector boundaries of the buildings are not available and only the post-event aerial photograph is used for the detection of the damaged buildings. From this point of view, an approach is proposed.

In chapter 4, the second approach is provided. In the second approach it is assumed that the vector building polygons are available and stored in a GIS. The damage detection is carried out using a line matching algorithm and the perceptual grouping concept.

In chapter 5, the results of the proposed approaches are given. The two approaches are discussed separately. The final chapter (Chapter 6) contains the conclusions and the recommendations derived from this study.

# CHAPTER 2

# BACKGROUND

In this chapter, first the previous studies on damage detection techniques are given. Then, the approach proposed in this study is explained briefly and the algorithms used are described.

## 2.1 Previous Studies

It is a well known fact that the natural disasters that strike the countries cause enormous destruction, create human sufferings, and produce negative impacts on national economies. Therefore, the importance of a rapid and reliable damage assessment is realized. The advancement in the information technology including GIS, remote sensing and satellite/aerial imagery can help to develop a natural management system. A reliable post-disaster damage assessment would allow the guarantee of the rapid emergency response, and sensibly reduce the effects by providing an immediate estimate of the extent and location of the damaged area and making this knowledge available to the responsible agencies. To acquire information about the area where a disaster occurred, remote sensing techniques are widely used. Remotely sensed images provide sufficient geospatial information to estimate the extend and the location of the damaged area.

Change detection is one of the most widely used methods for damage detection. It is a powerful application of remote sensing, in that spectral resolution of multi-band sensors can be used to monitor both significant and subtle land cover changes over time [2]. In remote sensing, change detection provides information on the same geographic areas within a time interval. In other words, digital change detection is a process of identifying differences in the state of a geographic object by observing it at different times [3]. For this process, space and aerial imagery is quite suitable. The traditional change detection methods include image differencing [4], image rationing, principal components analysis [5], change vector analysis [6], and post-classification comparison [7].

By using change detection algorithms, several studies have been carried out. Gamba and Casciati [8] developed a procedure to detect damaged areas by utilizing image processing tools. Before the event, images and data about the buildings were collected and analyzed by using GIS capabilities. After the earthquake, the post-event images were taken and supplied to system. And change detection techniques were applied between both pre and post-event images. An image differencing algorithm was used by Masato Ishii [9]. He used a color based change detection detection algorithm. The pre and post-event aerial images subtracted from each other. The regions resulting in high difference in color classified as damaged.

Turker and San [10] used pre- and post-event SPOT HRV images to detect the Izmit earthquake induced changes. The change areas were detected by subtracting the near-infrared channel of the merged pre-event image from that of the post-event image. The overall accuracy for the change areas were found to be 83%.

In a recent study, Turker and Cetinkaya [11] detected the collapsed buildings caused by the 1999 Izmit, Turkey earthquake using digital elevation models (DEMs) created from the aerial photographs taken before (1994) and after (1999) the earthquake. The DEMs created from two epochs were differenced and the difference DEM was analyzed on a building-by-building basis for detecting the collapsed buildings. The producer's accuracy for collapsed buildings was computed as 84%. Further, Turker and San [12] utilized the cast shadows to detect the collapsed buildigns due to Izmit, Turkey earthquake. The avaiable vector building boundaries were used to match the shadow casting edges of the buildings with their correspondig shadows and to perform analysis in a building specific manner. Of the 80 collapsed buildings, 74 were detected correctly, providing 92.50% producer's accuracy.

In order to detect a building, texture and spatial data were taken into account in many applications. From this point of view, the shadows of the buildings point out that there may be a building adjacent to the cast shadows. This is because, the detection of the shadows are easier than to detect the buildings themselves [13]. Irvin and Mckeown [13] state that the shadows are usually provide the darkest reflectance therefore, the detection of the edges on the shadow area can be carried out by simple image processing techniques. In order to confirm the presence of the buildings and to estimate the height of them, Hertas and Nevatia [14] used the cast shadows.

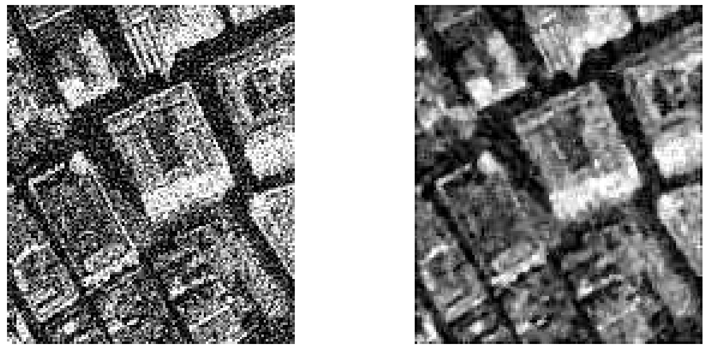
## 2.2 Algorithms used in this thesis

In this section, the algorithms used in this thesis are explained. The proposed approaches require a number of image processing techniques. First, a low-pass filter (median filter) is applied on the image to remove the noise. Then, an edge detection process (Canny) is performed to find the edge pixels. In order to extract the line segments from these edge pixels, the Douglas Peucker line simplification algorithm is used. Finally, the perceptual grouping algorithm is applied on the output of the Douglas Peucker algorithm.

### 2.2.1 Noise Removal

Images are often corrupted by impulse noise due to errors caused by the noisy sensors or communication channels. It is quite important to remove these impulse noise before further processing, such as edge detection, image segmentation, and object recognition. For this purpose, many approaches have been proposed [15]. Because of their simplicity and capability of preserving image edges the median-based filters have been widely used in the past two decades.

The median filter is a spatial filtering operation, which uses a 2-D mask to be applied to each pixel in the input image. To apply the mask means to center it in a pixel, evaluating the covered pixel brightnesses and determining which brightness value is the median value. Generally, a 3x3 mask is used for the operation. The median value is determined by placing the brightnesses in ascending or descending order and selecting the center value [15]. The median value will be the value for the center pixel in the mask of the output image. An example of the output of a median filter is given in Figure 2.1 (b).



( a ) The noisy Image ( b ) The Filtered Image

Figure 2.1: A sample image: (a) Noisy, (b) Filtered

### 2.2.2 Edge Detection

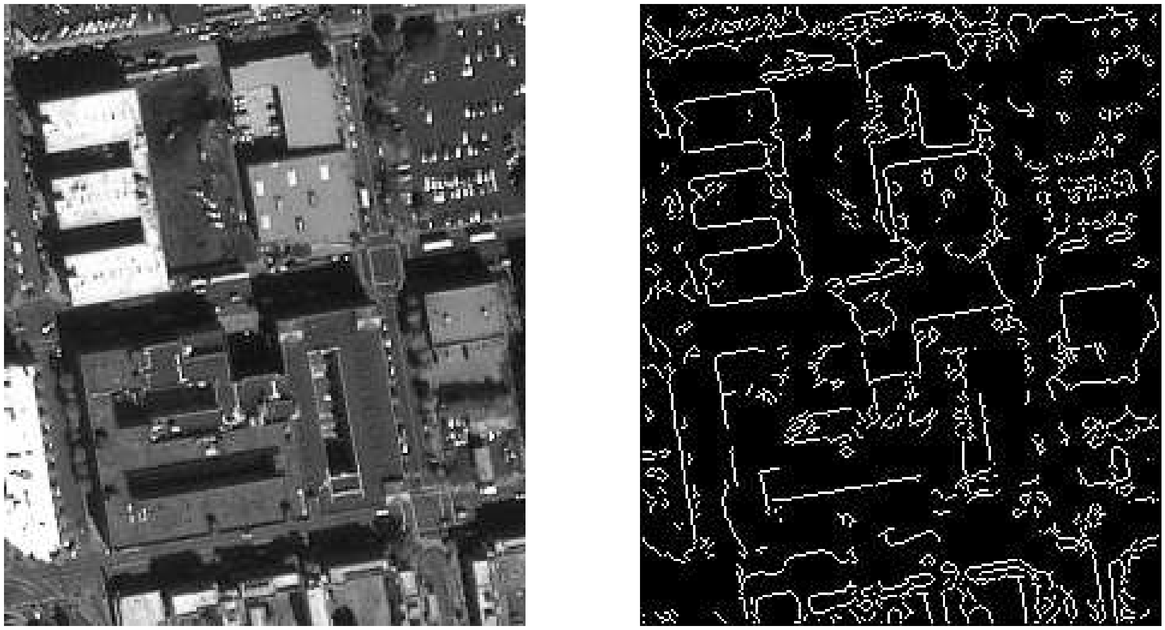
The edges in an image hold much information in the image. The edges tell where the objects are, their shapes and sizes, and also information about their texture. An edge is where the intensity of an image moves from a low value to a high value or vice versa. The edges characterize the boundaries. Therefore, it is a problem of fundamental importance in image processing. The edges are the areas with strong intensity contrasts. In other words, it is a jump in intensity from one pixel to the next. An edge detecting process significantly reduces the amount of data and removes useless information, while preserving the important structural properties in an image. There are many ways to perform edge detection. However, two major categories, gradient and Laplacian are widely used.

The gradient method detects the edges by looking to the maximum and minimum in the first derivative of the image. The idea is that, an edge occurs where there is a discontinuity in the intensity. In a discrete image, the gradient is calculated by simply differencing the gray values between the adjacent pixels. The well known gradient edge detectors are Roberts, Sobel, and Prewitt [16]. The Laplacian method searches for zero crossings in the second derivative of the image to find the edges. The Mexican Hat is an example of a second derivative operator [17].

Currently the Canny edge detection algorithm is very widely used around the world. The Canny edge detection method looks for the edges of the objects, and therefore, can be very useful for images containing the solid regions where the outlines are vectorized. This method attempts to find boundaries between poorly defined objects as well as the hard edges. An example for the output output of a Canny edge detector is given in Figure 2.2.

The Canny edge detection process works as follows:

- First, the image is smoothed by using Laplacian of Gaussian (LoG) filter to remove the noises. After this process, the gradient value is computed for each pixel.
- Then, non-maximal suppression is performed. Any gradient value that is not a local peak is set to zero. This process produces one pixel wide edges.
- Finally, hysteresis thresholding is applied. For this process, two threshold values are used. The higher threshold is usually three times as the lower threshold. Any pixel having a gradient greater than the higher threshold value are classed as a valid edge pixel. On the other hand, any pixels connected to these valid edge pixels that have a gradient value above the lower threshold value are also classed



( a ) Original Image ( b ) Canny output

Figure 2.2: An example of the output of a Canny edge detector.

as edge pixels.

### 2.2.3 Vectorization Process

Line detection and generalization is an important function in GIS operations. Therefore, many studies have been conducted using this operation. One of the most widely used line simplification algorithm is The Douglas Peucker algorithm. This algorithm tries to preserve the directional trends in a line by using a tolerance value which defines the number of the vertexes.

Figure 2.3 depicts an implementation example of the algorithm. The first pixel **a** is selected as the starting point of the line which will be extracted at the end of the algorithm, and **b** is selected as the final point of the line. The maximum distance perpendicular to ab is fc in Figure 2.3. This distance is compared with the user defined tolerance distance (td). If fc > td then, c becomes a new final point of the line. After that, the maximum distance from pixels to ac is computed. This distance is ed. This distance (ed) is also compared with the tolerance distance (td). If ed > td then, the new final point becomes d. This process is repeated until all the distances from each pixel points to the line are smaller than the tolerance distance.

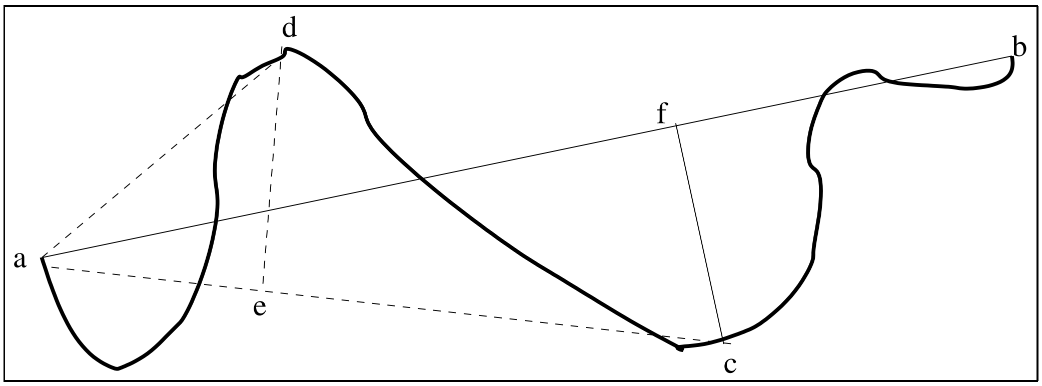



Figure 2.3: The Douglas Peucker Algorithm.

### 2.2.4 Perceptual Grouping

The perceptual grouping algorithm uses the gestalt theory [18]. By using the principles defined in the theory (Figure 2.4), a closed contour is formed from the line segments hierarchically. The principles used are as follows:

- *Proximity* means that the line segments which lie close to each other are grouped together.
- *Collinearity* means that the line segments which are on the same orientation are grouped together.
- *Curvilinearity* means that the line segments forming a curvilinear segment are grouped together.

- *Parallelism* means that the parallel line segments are grouped together.

- *Symmetry* means that the symmetric line segments are grouped together.

In order to overcome the fragmented lines and to obtain more abstract objects, the line segments are subsequently organized within a hierarchy of grouping hypotheses by using various Gestalt laws (Figure 2.4). In the lowest level, the line segments are combined to form collinear and curvilinear groups which are nearly straight lines or elliptical arcs. These groups are denoted as linear groups. In the next level, the symmetric and the parallel line segments are grouped. In the last level, these gropes are organized into closed contours.

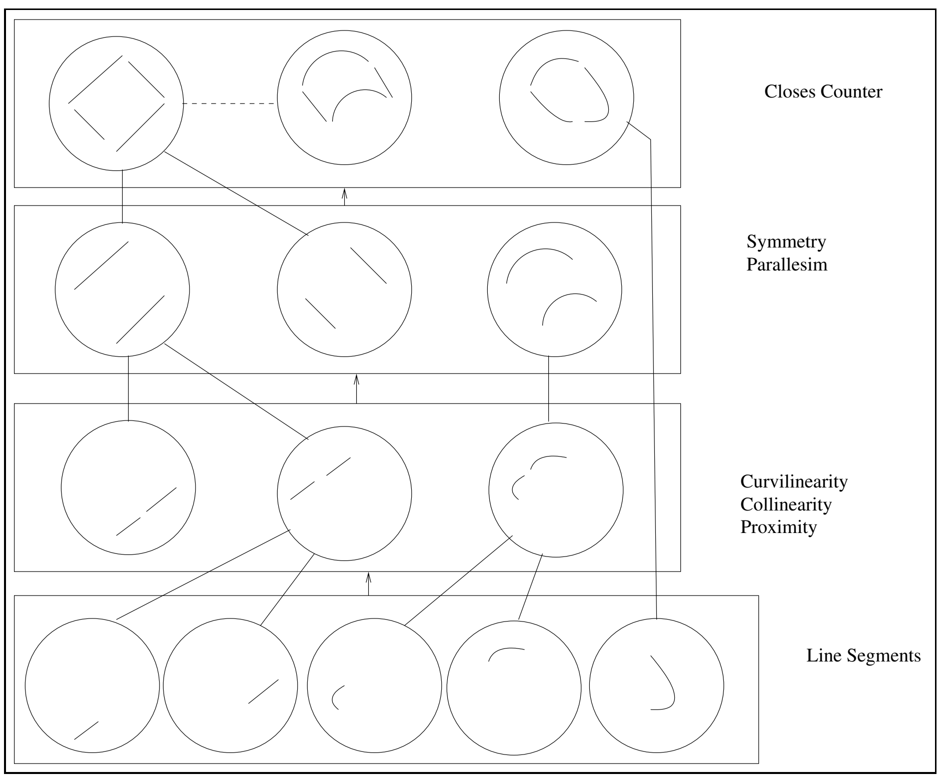



Figure 2.4: Hierarchy of The Grouping and The Gestalt Principles [1]

# CHAPTER 3

# DETECTION OF THE DAMAGED BUILDINGS THROUGH PERCEPTUAL GROUPING FROM POST-EVENT AERIAL IMAGERY

In this section, the concept of detecting the collapsed buildings through perceptual grouping from post-event aerial imagery is explained. The aim is to detect the damaged buildings from post-event aerial imagery by means of easy and simple procedures based on human perception. Before applying the perceptual grouping algorithm, several processings, described in the following sections, are needed to be applied. First, the edges are extracted from the post-event aerial imagery. Second, the line segments are generated. These line segments are then grouped by confirming in terms of continuity, proximity, perpendicularity, symmetry, and parallelism. The methodology is described in the following section. Next, the main processing stages are explained. Finally, the grouping algorithms are explained and, the detection processes of the collapsed buildings are given.

## 3.1 The Methodology

The main purpose of this study was to detect the collapsed buildings from the post-event aerial imagery. To achieve this objective, an approach was developed. The proposed approach is based on the utilization of the post-event aerial photos. It is assumed that the post-event aerial photos provide high spatial resolution and therefore allowed detailed spatial information about the urban structures.

The flow diagram of the methodology is given in Figure 3.1. The input imagery is a post-event aerial photograph that covers the study area of interest. The resolution and the quality of the aerial imagery is important for the success of the algorithm. First, a median filter is applied as a preprocessing operation on the aerial image. Then, a Canny edge detection algorithm is carried out to extract the edge pixels between the buildings and the surroundings. After extracting the edge pixels, the vectorization process is performed. Finally, a perceptual grouping algorithm is applied to detect the buildings. The existence of a building indicates that it is uncollapsed. On the other hand, the absence of a building means that the building is collapsed due to earthquake.

In order to apply the perceptual grouping algorithm, several processing steps are required to extract the line segments of the post-event aerial photograph. These processing stages include;

- Noise Removal,
- Detection of The Edge Pixels from Post-Event Aerial Photo, and
- Vectorization Process.

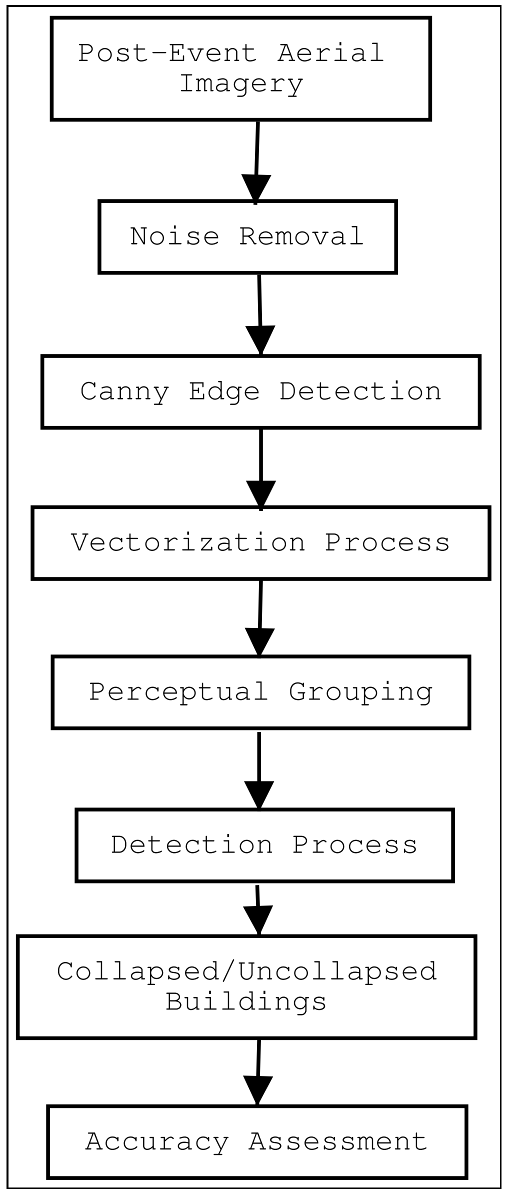



Figure 3.1: The processing stages of the proposed methodology

## 3.2 Noise Removal

The first process applied on the post-event aerial image was the median filtering. The goal of this filtering was to remove the noise. A median filter is a nonlinear filter generally used to remove impulsive noise from an image [19]. It is a more robust method than the traditional linear filtering because it preserves the sharp edges. For each pixel in the image, a median value is found. The median value is determined by placing the brightness values around the pixels (generally 3x3 mask is used) in ascending or descending order of magnitude and selecting the center value. The median value is set to the value for the center pixel of the mask in the output image.

## 3.3 Edge Detection

After performing the filtering process, the edge pixels were detected using the Canny edge detection algorithm. To apply the algorithm, a built-in function in Matlab 6.1.0 was used. The reason for choosing this algorithm was its efficiency and the output, it contains one pixel wide edges. The one pixel wide edges are used to vectorize the image. In section 2.2.2, the algorithm of the Canny edge detector is described. An output of a Canny edge detector is illustrated in figure 3.2.

## 3.4 Vectorization

The above applied Canny edge detector produced the one pixel wide edges. However, in order to find the line segments, the connected edge pixels must be detected, through a raster-to-vector conversion process. There are many ways to perform this conversion process. In the present case, the Douglas Peucker algorithm was used both to extract

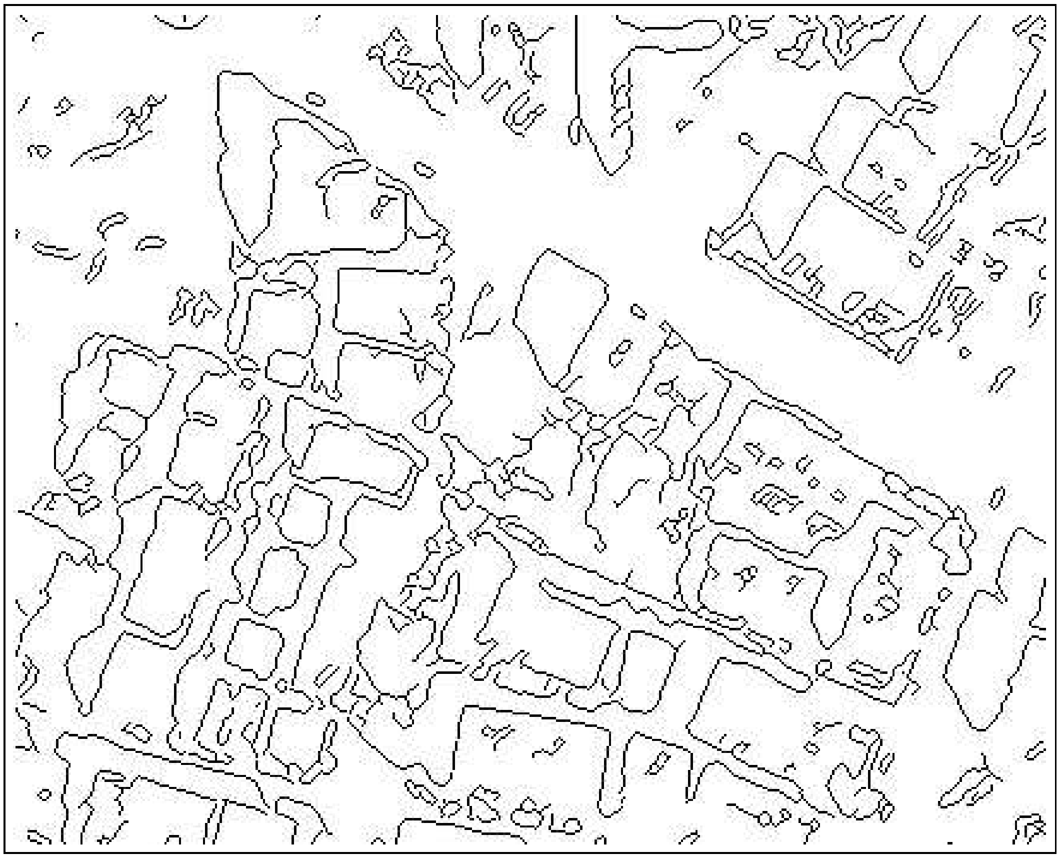


Figure 3.2: An example output of Canny edge detector

the line segments and to smooth the extracted line segments. The Douglas Peucker delivers the best perceptual representations of the original lines [20].

In the study area, most of the buildings are rectangular shaped. This means that each building has four corners and four straight edges. Therefore, during the vectorization process finding the line segment that is most similar to the original edge of the building is very important in order to detect the buildings correctly through perceptual grouping. From this point of view, the Douglas Peucker algorithm works well to implement the proposed methodology.

A line segment contains the connected edge pixels that are on the same direction with the direction of the line. Therefore, the edge pixels that are connected to each other are grouped together and named as edge pixel segments. The edge pixel segments can be seen in Figure 3.3, where each point represents a pixel.

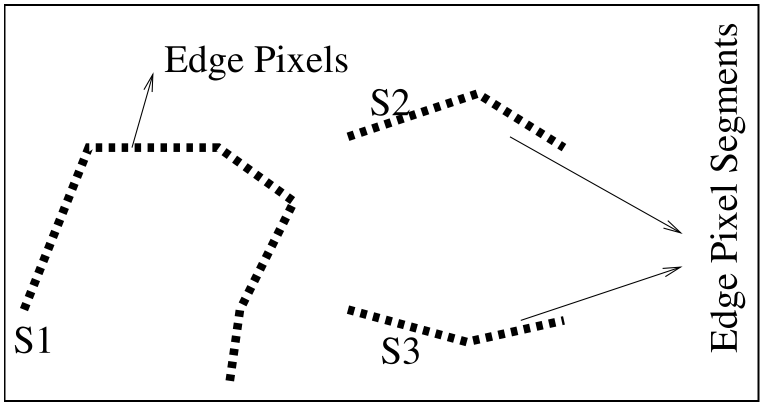



Figure 3.3: Edge pixels and edge pixel segments. Each point represent a pixel

A segment of connected pixels (pixel segment) may include more than one vertex, which yields more than one line segment. The location and the number of these vertexes were found using the Douglas Peucker algorithm. This algorithm tries to minimize the number of vertexes. In other words, it tries to find the minimum number of lines that represent the edge pixel segments. Therefore, the algorithm produces the polygonal objects (Figure 3.4), which are useful for performing the perceptual grouping procedure. This is because the rectangular shaped structures being a special kind of a polygonal shape are dealt with.

### 3.4.1 The Implementation of The Vectorization Process

The vectorization process finds the locations of the vertexes on the edge pixel segments. Adjacent two vertexes form a line segment, which may be a candidate for one of the edge of a building. In other words, each vertex defines a terminal point of a line

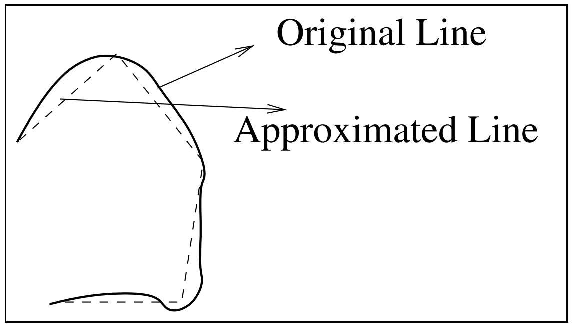



Figure 3.4: The vectorization process

segment. In order to run the algorithm on the edge pixel segments, two parameters are needed:

(i) Tolerance Value ($\epsilon$) and

(ii) N - The number of pixels to recognize a vertex.

The first parameter ($\epsilon$) is required to decide if an edge pixel is a vertex or not. In other words, this parameter is used to recognize whether there is an orientation difference between the first and the last pixel that violates the straight line segment. The second parameter (N) is used to define the minimum number of pixels to search a vertex. A line is drawn between the first pixel and the $N^{th}$ pixel. Then, a pixel that violates the straight line by means of the tolerance value is searched on the edge pixel segments (Figure 3.5). This process continues from the first to the last pixel of the edge pixel segments. After performing this process, the vertexes are located. In other words, the line segments are generated.

The above given two parameters used in the Douglas Peucker algorithm affect dramatically the line segments generated [21]. Therefore, finding the optimum values

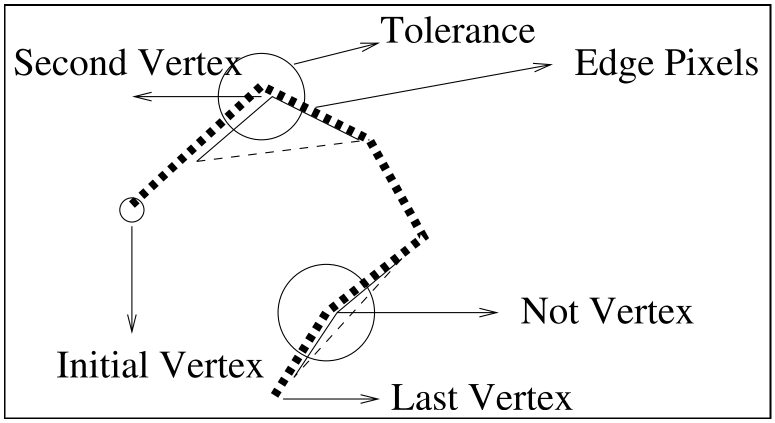



Figure 3.5: Finding the vertexes

for these parameters are important for the success of the proposed approach. The shape of the structures can be used to determine an optimum value for $\epsilon$. If there are circular structures in the selected area then, a small value must be chosen for $\epsilon$. This is because a small value for $\epsilon$ results in more vertexes and therefore, it represents a circular shape more effectively. In this study, a high value was selected for $\epsilon$ because the study area contains mainly the rectangular shaped buildings.

## 3.5 Perceptual Grouping

In a two dimensional digital imagery the objects are represented by line segments. In other words, the line segments form the objects in a digital image. If all the line segments of an object are grouped together to form the whole object then, the object can be recognized. Based on this statement therefore, the perceptual grouping algorithm was used to detect the buildings in the proposed approach.

The line segments generated through the vectorization process were available for the perceptual grouping algorithm. A hierarchical method was used to group the line segments. First, the collinear line segments were grouped together to see if they lie close to each other or not. This process was carried out to combine and obtain a full line, which might have been fragmented somehow in the edge detection and the vectorization process. This was a frequently faced situation. Then, the lines were grouped together to form a corner. Finally, the buildings were detected. A detailed explanation about this grouping algorithm is given in the following sections.

### 3.5.1 The First Level Grouping

The aim of the first level grouping was to combine the line segments that are the small parts of one of the edges of a building. An edge of a building can be fragmented into more than one line segments through an edge detection or a vectorization process. A line segment that is most similar to the original line was obtained by means of this level of grouping. This is illustrated in figure 3.6 with an example.

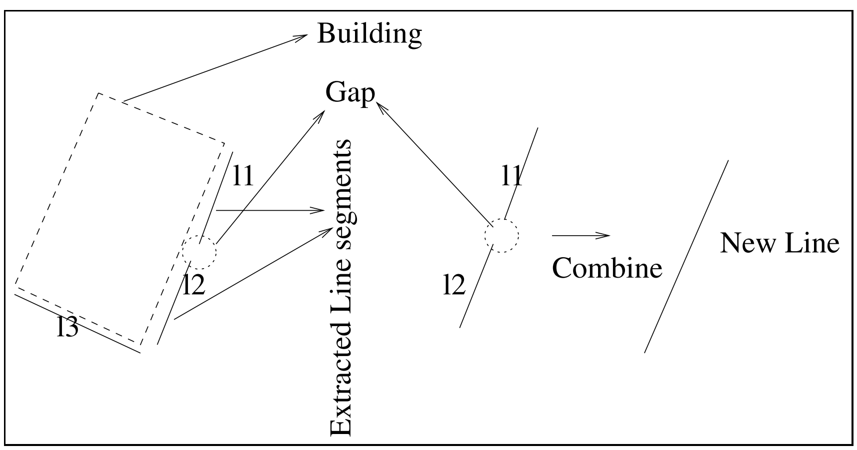



Figure 3.6: The first level grouping

In this level, two principles were verified, that are:

- *Proximity :* the line segments lie close to each other, and
- *Collinearity :* the line segments illustrate an orientation.

In order to verify these principles, two parameters must be chosen. These parameters are (i) the maximum distance between the two line segments and (ii) the allowed difference in the orientations. The maximum distance must be chosen less than the minimum distance between the buildings in the selected area. Otherwise, the two line segments that belong to different buildings can be combined.

If the difference in the orientation of the two line segments are less than the allowed difference and the gap between these line segments are smaller than the maximum distance then, the two line segments are combined (Figure 3.6).

### 3.5.2 The Second Level Grouping

Generally, the residential buildings are rectangular shaped and can be represented as vector polygons. The rectangular shaped buildings include corners with an angle in the range of $60^0$ to $120^0$. Based on this knowledge, a corner can be used as an indication of a building. In the following section, the utilization of the corner information is explained.

In the second level grouping, the line segments are group together according to the principles of the perpendicularity and proximity. The reason for using these two parameters is that most of the buildings in the selected area are rectangular shaped. Two parameters must be chosen for verifying the two principles. The first parameter is the maximum distance between the line segments to verify the proximity. The second parameter is the angle range at the corner to verify the perpendicularity. The second level grouping is illustrated in Figure 3.7. For each couple of line segments, the two principles are checked whether they are satisfied or not. If a couple is found then, the line segments are grouped together.

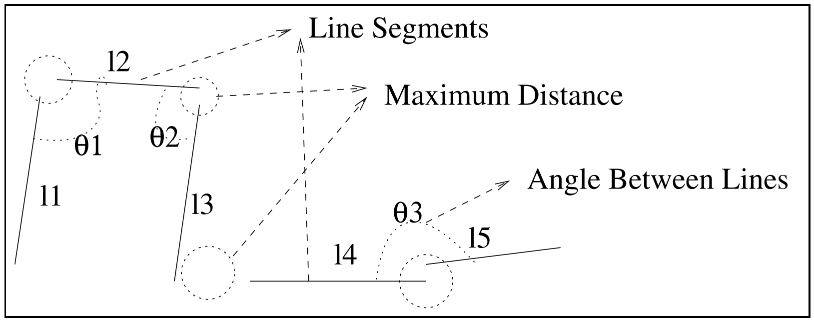



Figure 3.7: The second level grouping

### 3.5.3 The Third Level Grouping

Finally, in the third level grouping, the pairs of corners are investigated. As mentioned above, the study area, where the methodology was implemented, contains mainly the rectangular shaped buildings, each of which contains four corners. Therefore, the principles, which were used in this level, include the symmetry and the parallelism.

It is very well known that each corner of a rectangular shaped building has a pair of line segments that intersect at a corner (Figure 3.8). Based on this knowledge, the pairs of corners were grouped to find if they contain a common line segment (l2) and other two lines (l1 and l3) forming the corners are parallel and symmetric. This is illustrated with an example in Figure 3.8. In the figure, line l2 is common for both corner C1 and corner C2. The lines l1 and l3 are on the same side of l1 and they are parallel. Since the above two principles are verified, these two corners are grouped, therefore.

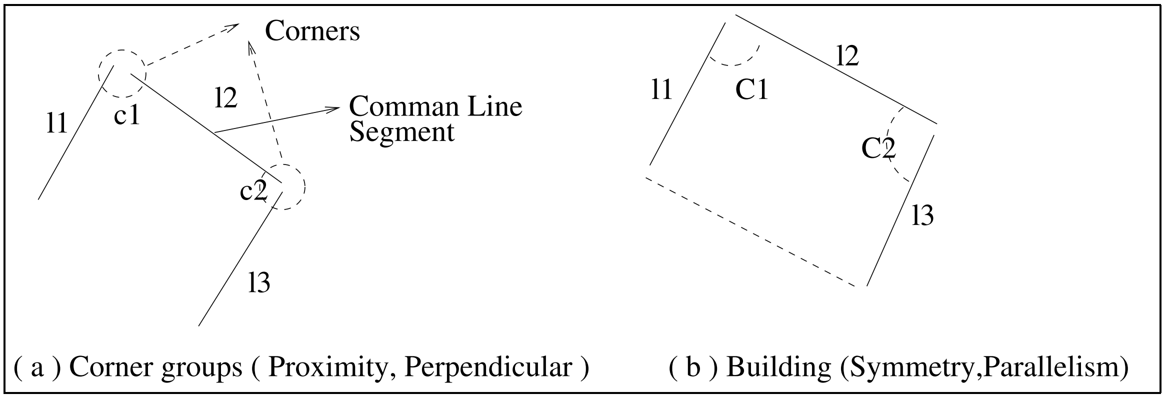



Figure 3.8: The third level grouping

## 3.6 Building Detection

In this section, the rules that were used to detect a building are explained. In an ideal situation, the corner groups form a closed contour. In such a situation, it is easy to detect a building. However, this may not always be the case. For this reason, the below given rules were defined. First, the description of the rules are given. Then, for each rule, the specific examples are provided.

**Rule 1:** *If four corners are grouped together then, it is a building.*

The working mechanism of this rule is illustrated with an example in Figure 3.9. If the corner groups constitute a closed contour then, the closed area is most probably a building. However, this statement may not be correct if the closed contour is a shadow area because the cast shadows of a building may produce corners that are also verified by this rule. In this case, the average brightness value of the closed area is used as additional information. If the average brightness value is less than a pre-defined threshold then, it is not considered to be a building.

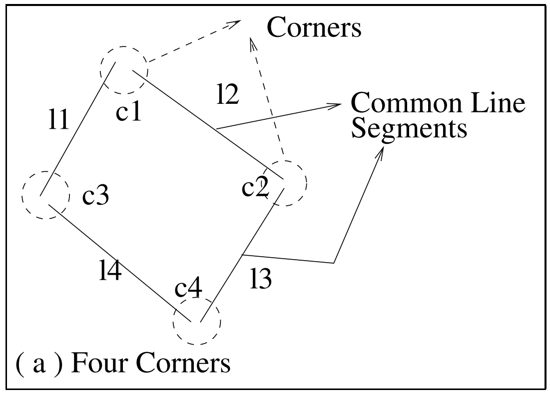



Figure 3.9: The illustration of rule 1

**Rule 2:** *If two or three corners are grouped together then, it is a building.*

Usually, because of the sun illumination effect, some of the edges of the buildings cannot be detected by the Canny edge detector. In such a case, the rule 1 does not work effectively and the building may be missed by the algorithm. Therefore, if two or three corners are grouped together then, this group is also labelled a building. This is based on an assumption that the detection of at least two or three corners would be enough to label the area as a building. This rule is illustrated in figure 3.10.

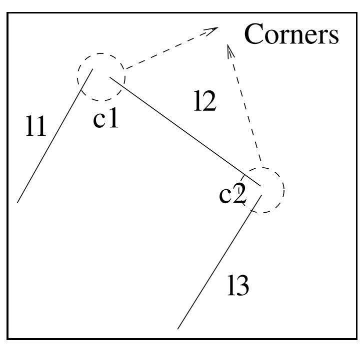



Figure 3.10: The illustration of rule 2

**Rule 3:** *If a shadow producing corner of a building and a real shadow corner are grouped together then, it is a building.*

Generally, the edges on the shadow producing corner of a building are detected accurately because there is a sharp difference in brightness values on the shadow side between the roof of the building and surroundings. This edges along the cast shadows are detected perfectly similar to shadow producing edges. This rule claims that if the corner formed by the edges along the cast shadows and the shadow producing corners are grouped together, then it is a building. This rule is illustrated in figure 3.11.

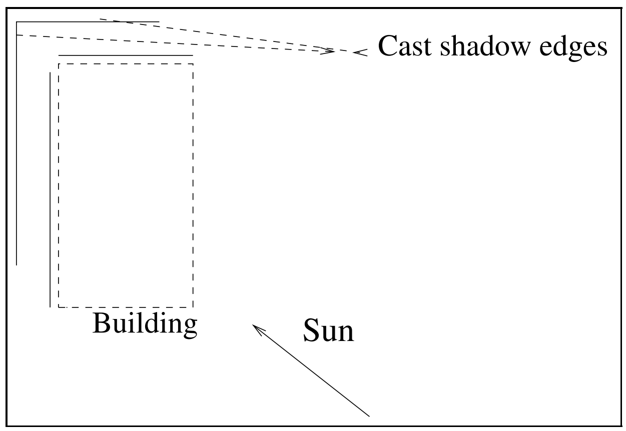



Figure 3.11: The illustration of rule 3

# CHAPTER 4

# GIS ASSISTED DETECTION OF THE DAMAGED BUILDINGS THROUGH PERCEPTUAL GROUPING FROM POST-EVENT AERIAL IMAGERY

In the previous chapter, the earthquake damaged buildings were detected through perceptual grouping without using GIS data. In this section, it is assumed that the building boundaries are available and stored in a GIS as vector polygons. The building boundaries are used as additional information for detecting the collapsed buildings. In this case, the damage detection is carried out in a building specific manner. The idea is that, if a building is uncollapsed, its edges should match with the line segments extracted through the vectorization process. Otherwise, the building is declared to be collapsed.

## 4.1 The Methodology

The main processing steps followed on the processing and analysis of the post-event aerial image to detect the collapsed buildings through GIS assisted perceptual group-

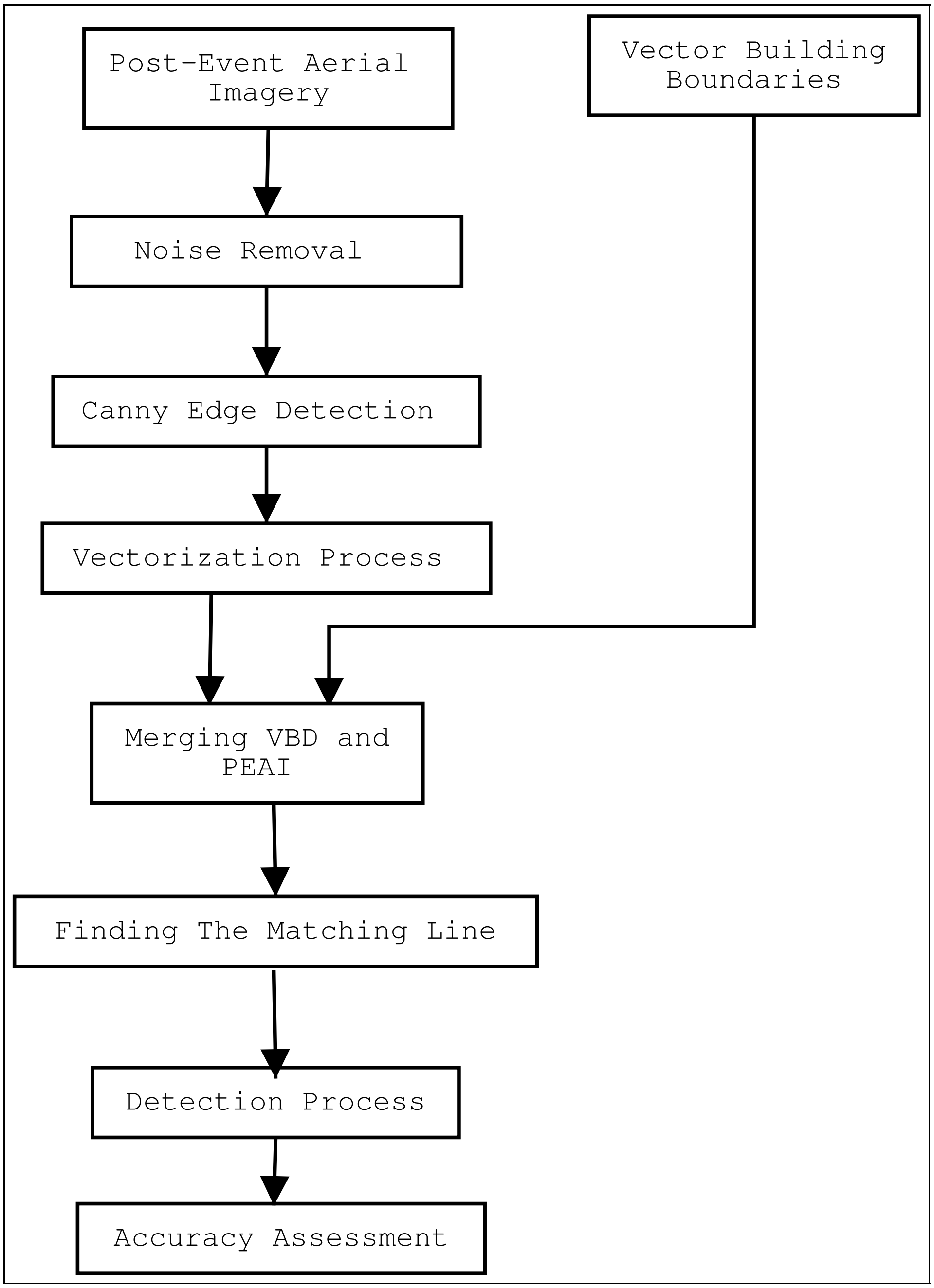



Figure 4.1: The flow diagram of the algorithm - VBD (Vector Building Boundaries) and PEAI (Post-Event Aerial Imagery)

ing is illustrated in figure 4.1. First, the median filter is applied on the post-event aerial image to remove the noise. Then, the Canny edge detection process is carried out to detect the edge pixels. After that, the line segments are extracted by applying the vectorization process. Upon extracting the line segments, the agreement is measured between the extracted line segments and the shadow casting edges of the vector building polygons. The decision about the damage condition of a building is made based on the degree of the match between the extracted line segments and the shadow casting edges of the building polygons. Finally, the accuracy assessment is performed to find the reliability of the proposed approach.

## 4.2 The First Processing Steps

The similar processings used in the previous chapter were also carried out in this approach. However, the vector building boundaries were used as additional information to increase the reliability of the damaged building detection. First, the median filter is applied to remove the noise from the post-event aerial image. This must be carried out before the edge detection process to overcome the noisy edge pixels. Then, the Canny edge detector is applied on the image to find the edge pixels. After that, the line segments are extracted from the edge pixels. Finally, using the perceptual grouping algorithm, the linear groups and the corners are found by verifying the proximity, the collinearity and the perpendicularity principles of the Gestalt law.

In this part of the study, the collapsed buildings were detected using a line matching algorithm between the vector boundaries of the buildings and the extracted linear groups. As an additional information, the shadow corners extracted through the perceptual grouping process were also taken into account. In the following sections, the detection procedure is explained in detail.

## 4.3 Integrating The Post-Event Aerial Imagery and The Vector Building Boundaries

In this study, it was assumed that, the digital vector boundaries of all the buildings falling within the area were known and stored in a GIS environment. Before performing any further processes, the extracted line segments and the vector boundaries of the buildings were integrated ( Figure 4.2 ). In the figure, the red colored polygons illustrate the vector building polygons, while the black colored lines illustrate the line segments extracted through the vectorization procedure.

## 4.4 Line Matching

The idea of the proposed approach is that *"if the edges of a building polygon match with the line segments extracted through the vectorization process then, the building is uncollapsed".*

### 4.4.1 Primitives

To match the lines, the following primitive parameters were used:

- Orientation,
- Length of the line segments, and
- Distance between the lines.

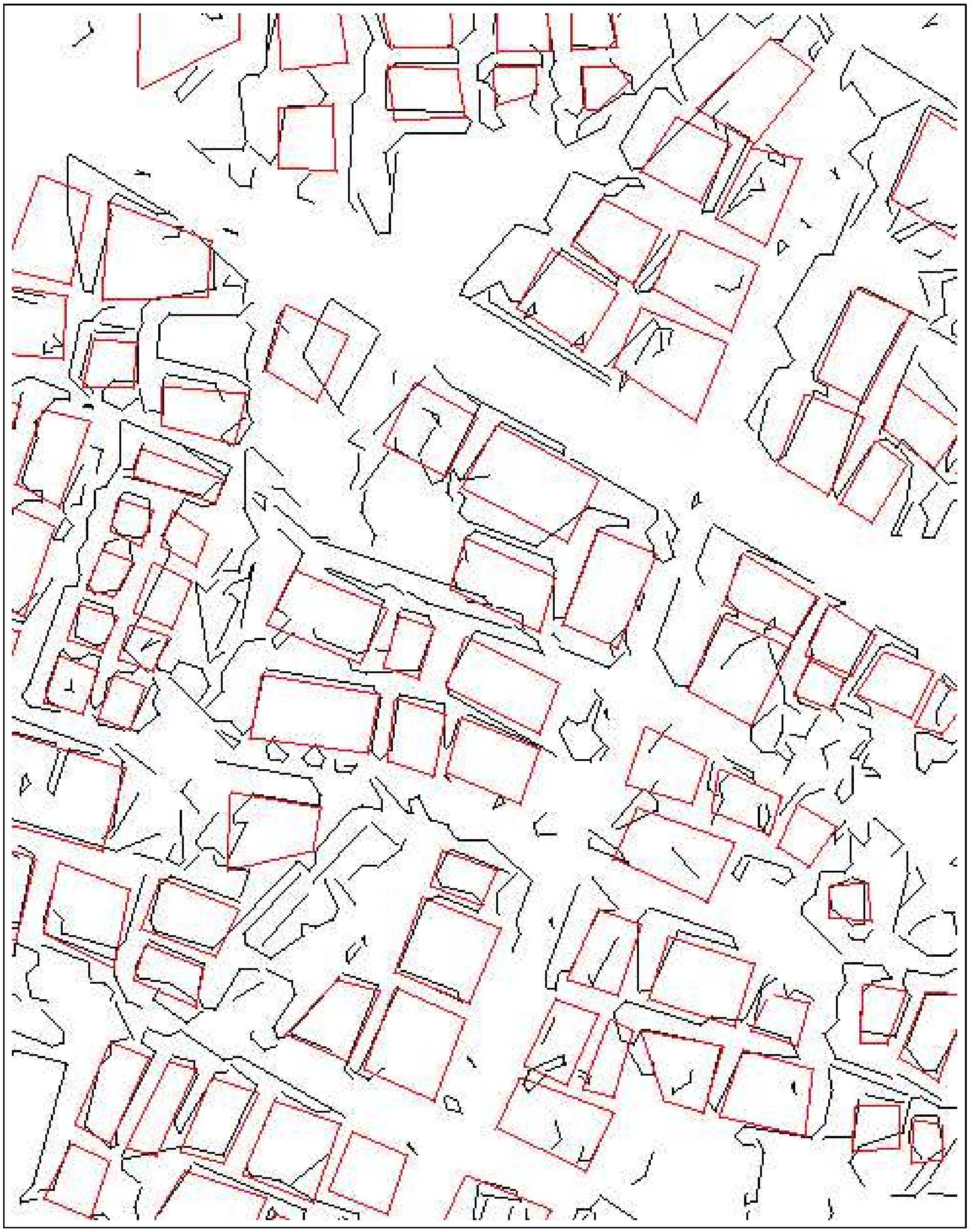


Figure 4.2: The vector building boundaries and the line segments (Vector boundaries in red)

In a Cartesian coordinate system, a line can be identified by its terminal points. For a line matching algorithm, the orientation and the distance parameters were needed. Therefore, the lines were represented in a polar coordinate system. In a polar coordinate system, the equation of a line becomes ( Equation 4.1) ;

$$x * cos(\theta) + y * sin(\theta) = r \tag{4.1}$$

In Equation 4.1, r represents the distance from the origin and $\theta$ represents an angle measured from the x axis ( Figure 4.3 ). In order to compute the similarity of the lines, the difference in their orientation is computed first. The orientation is one of the dominant parameters used to measure the similarity. In Equation 4.1, $\theta$ provides the orientation of the line. In the present case, the allowed difference was set for $\theta$ to $10^0$. As a distance parameter, r was used. The parameter r represents the difference between the origin and a line. The difference between the r values of the lines can be used as a distance parameter between the orientations of the line segments.

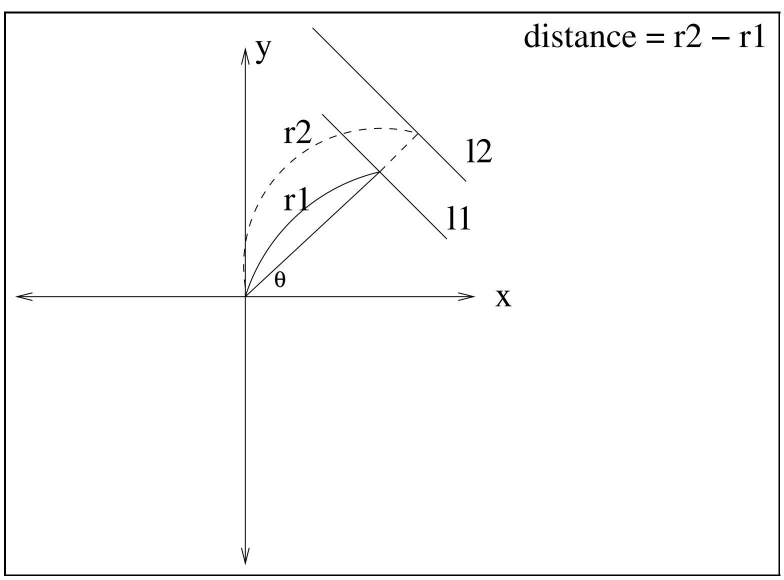



Figure 4.3: The distance between the line segments

The final parameter used was the length of the line segments. This primitive parameter was computed as the length of the overlapping part of the two lines (Figure 4.4). In Figure 4.4, l1, l2, l3, l4 and l5 represent the line segments that satisfy the orientation and the distance parameters. For each satisfying line, the overlapping parts on the corresponding edges of the vector polygon boundaries are computed. The overlapping parts with the vector polygon boundaries are shown in Figure 4.4. If the total length of the overlapping parts on the vector boundaries of a building is higher than %75 of the total length of the correspondig edge then, this edge is considered to be a matching line segment of the building under consideration.

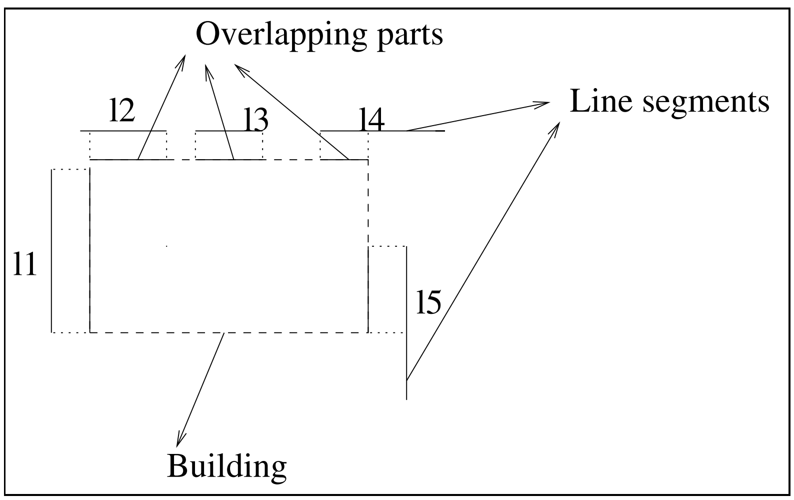



Figure 4.4: The overlapping parts between the line segments and vector polygon boundaries

## 4.5 Building Detection

The vector polygon boundaries of the buildings were available. Therefore, the algorithm was developed based on utilizing this boundary information. Using this information, the decision rules were defined. The decision chart is illustrated in Figure 4.5, where several rules are set for detecting the damaged buildings.

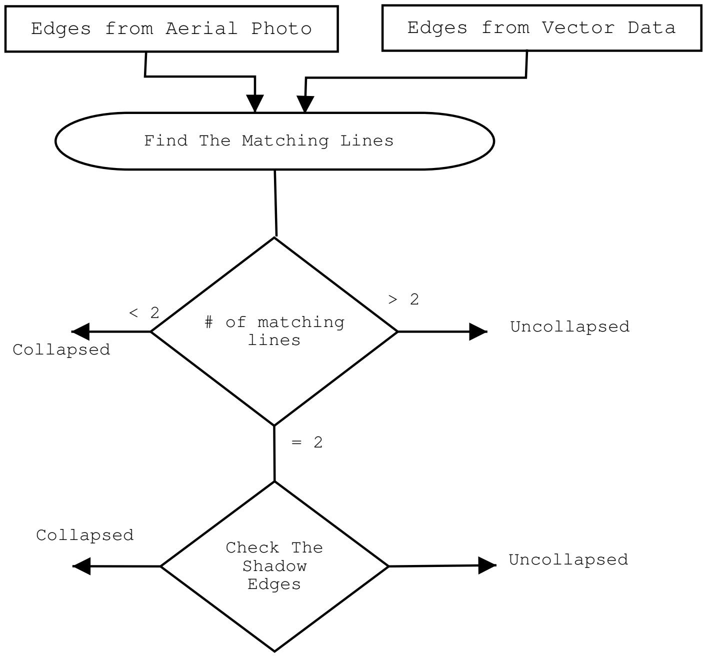



Figure 4.5: The decision chart

**Rule 1:** *If all the edges of a building polygon match with the extracted line segments then, the building is labelled uncollapsed.*

This was an obvious case. If a building is uncollapsed then, all the edge pixels should be detected by the Canny edge detector. And, all the line segments should be extracted after applying the vectorization process on these edge pixels. In such a case, all the vector boundaries of the building polygon match with the extracted line segments and the building is labelled uncollapsed.

**Rule 2:** *If at least three edges of a building polygon match with the line segments then, the building is labelled uncollapsed.*

Because of the sun illumination effect, the roofs of the buildings and the surroundings reflect approximately the same intensity. Therefore, the line segments that are facing the sun may not be detected correctly by the Canny edge detector. This is illustrated in Figure 4.6. As can be seen in the figure, some of edges of the vector boundaries that correspond to these undetected line segments do not match. However, it is a fact that this building is uncollapsed. For this condition, the second rule is defined. The rule is that *"if more than the half of the edges match a line segment then, the building is labelled uncollapsed."*

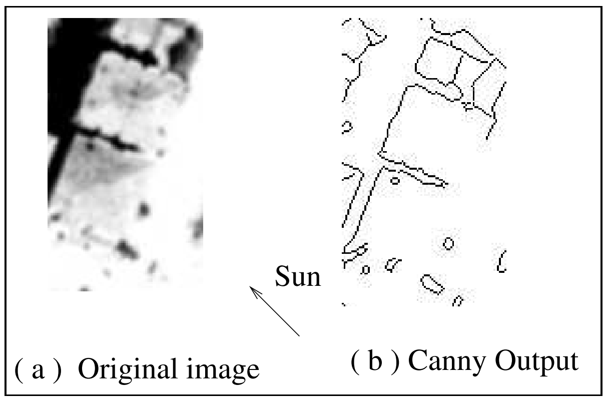



Figure 4.6: (a)The original imagery and (b) the output of the Canny edge detector

**Rule 3 :** *If the shadow producing edges of a building match with the line segments and there is a shadow corner then, the building is labelled uncollapsed.*

Because of the brightness difference between the roofs of the buildings and their cast shadows, the edge pixels were detected perfectly along the shadow producing edges. Therefore, most of the cases, this information was enough to label a building as collapsed or uncollapsed. If the two shadow producing edges match a line segment and there is a shadow corner formed by the edges along the cast shadow then, the building is labelled uncollapsed.

The cast shadows of a building produce perfect edges along the sun's illumination direction. The edges of the shadows are grouped to form a shadow corner in the perceptual grouping process. The shadow corner is illustrated in Figure 4.7. The cast shadow corner and the corner on the shadow producing edges are grouped to form a building.

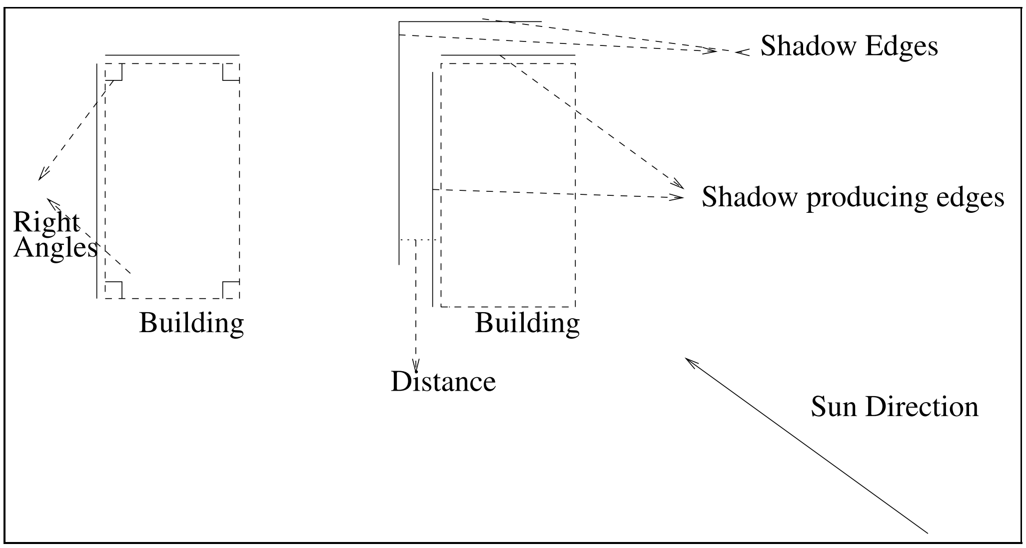



Figure 4.7: ( a ) A rectangular building,( b ) shadow corners

# CHAPTER 5

# RESULTS and DISCUSSIONS

In this chapter, the results of the proposed damage detection methods that were implemented in the selected study area of Golcuk are given. First, the results of the vectorization process are provided. Because the output of the vectorization process becomes the input for the damage detection algorithms its accuracy therefore affects the overall accuracy of the algorithm. This is followed by explaining the results of the perceptual grouping process. After that, the results of the two detection algorithms as well as their comparative evaluations are given. In addition, the difficulties and the problems for applying the proposed approaches are discussed.

## 5.1 The Edge Detection and The Vectorization Process

The first process applied on the image was to detect the edge pixels. A Canny edge detector was used for this purpose. Instead of implementing the Canny detector, a built-in edge function in mathlab 6.1.0 was used to detect the edges. This function takes a parameter (SIGMA) which is used as standard deviation of the Canny edge detection algorithm. Increasing this parameter reduces the sensitivity to noise, at the expense of losing detailed line segments. The optimum value for this parameter must be found therefore. The default value of SIGMA is 1, but in the present case it was set to 0.3. The edge pixels of a part of the study area are illustrated in Figure 5.1(b).

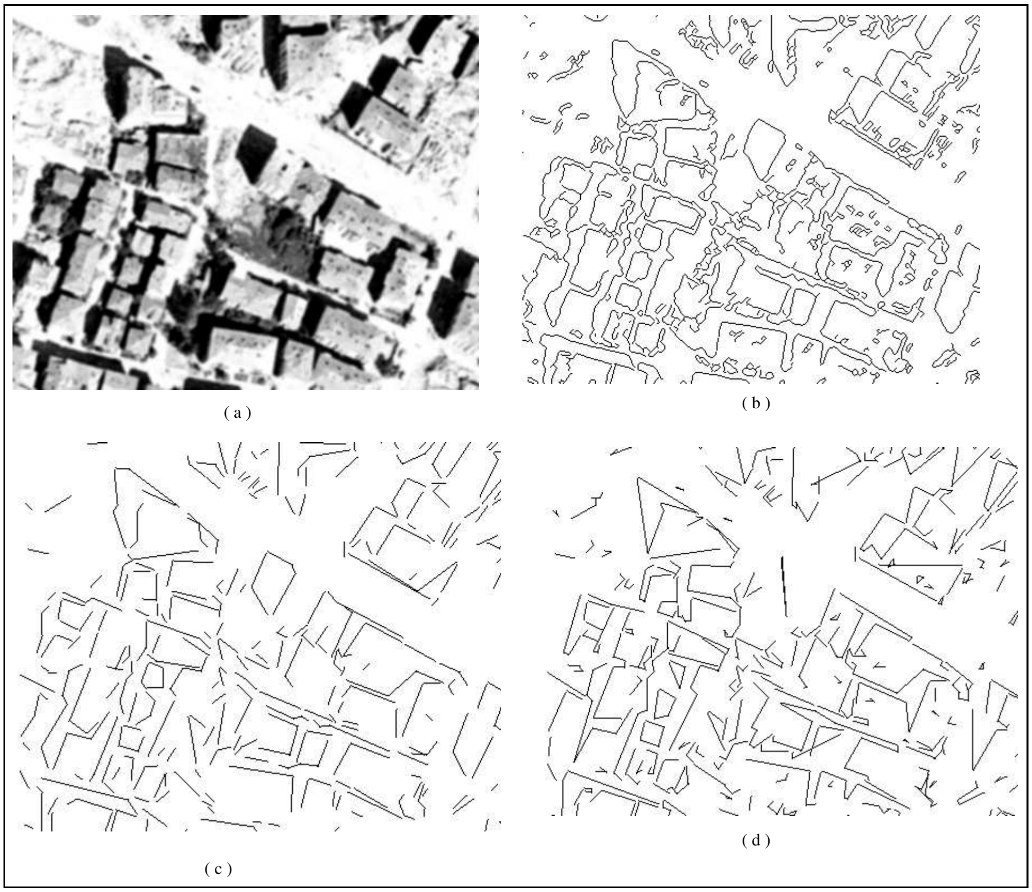


Figure 5.1: ( a ) Original image ( b ) Canny output ( c ) Small tolerance ( d ) Large tolerance

As explained in section 2.2.2, there are several edge detection algorithms. But in this study, the Canny detector was used because of its efficiency and the output it generates one pixel wide edges. The results of the Canny edge detector was quite efficient for both approaches proposed in this study. Most of the edges of the buildings were found successfully with the SIGMA value of 0.3. The edges of the study area is depicted in figure 5.2.

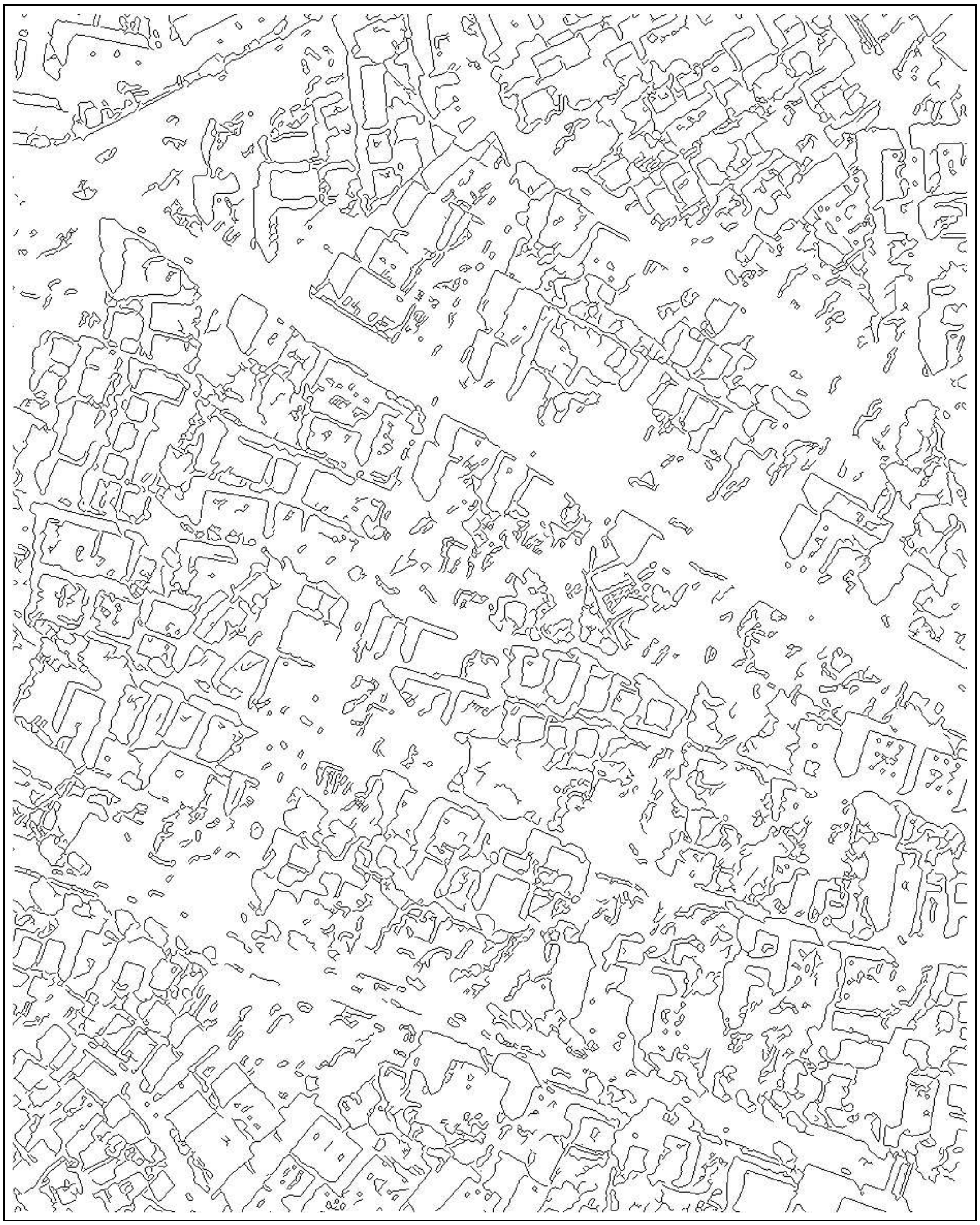


Figure 5.2: The edges of the study area

As can be seen in figure 5.1 (b), the output of the Canny edge detector contains noisy lines which cannot be an edge of an urban structures. These noisy lines should be removed before performing the perceptual grouping procedure. The trees, small objects cause these line segments to be generated by the Canny edge detector. Because of this, they should be removed. The removing process is carried out during the vectorization process. If an extracted line segment is small and there is no line segments lie close to it then, this line segment is removed. This result is illustrated in figure 5.1(b),(c).

As explained in section 3.4, two parameters $\epsilon$ and N were used to perform this process. The parameter of $\epsilon$ presents the tolerance value that determines the number of vertexes and the length of them. Figure 5.2(c), (d) shows the output of the Canny edge detector when the vectorization is performed using a large and a small tolerance values. The two results are specifically given for the comparison purpose. It can be observed in Figure 5.1 (b) that the output with the small tolerance value provides high detail. However, it contains more noisy lines. Therefore, the optimum tolerance value must be found to keep the line segments that correspond to the edges of the buildings. The line segments of the study area is illustrated in figure 5.3.

## 5.2 The Perceptual Grouping

After extracting the line segments, they were grouped together through a hierarchical perceptual grouping procedure. The corners were detected and labelled according to their directions. Figure 5.4 depicts the corners in different colors. The yellow colored corners are in south-east direction, the green color represents the corners along the

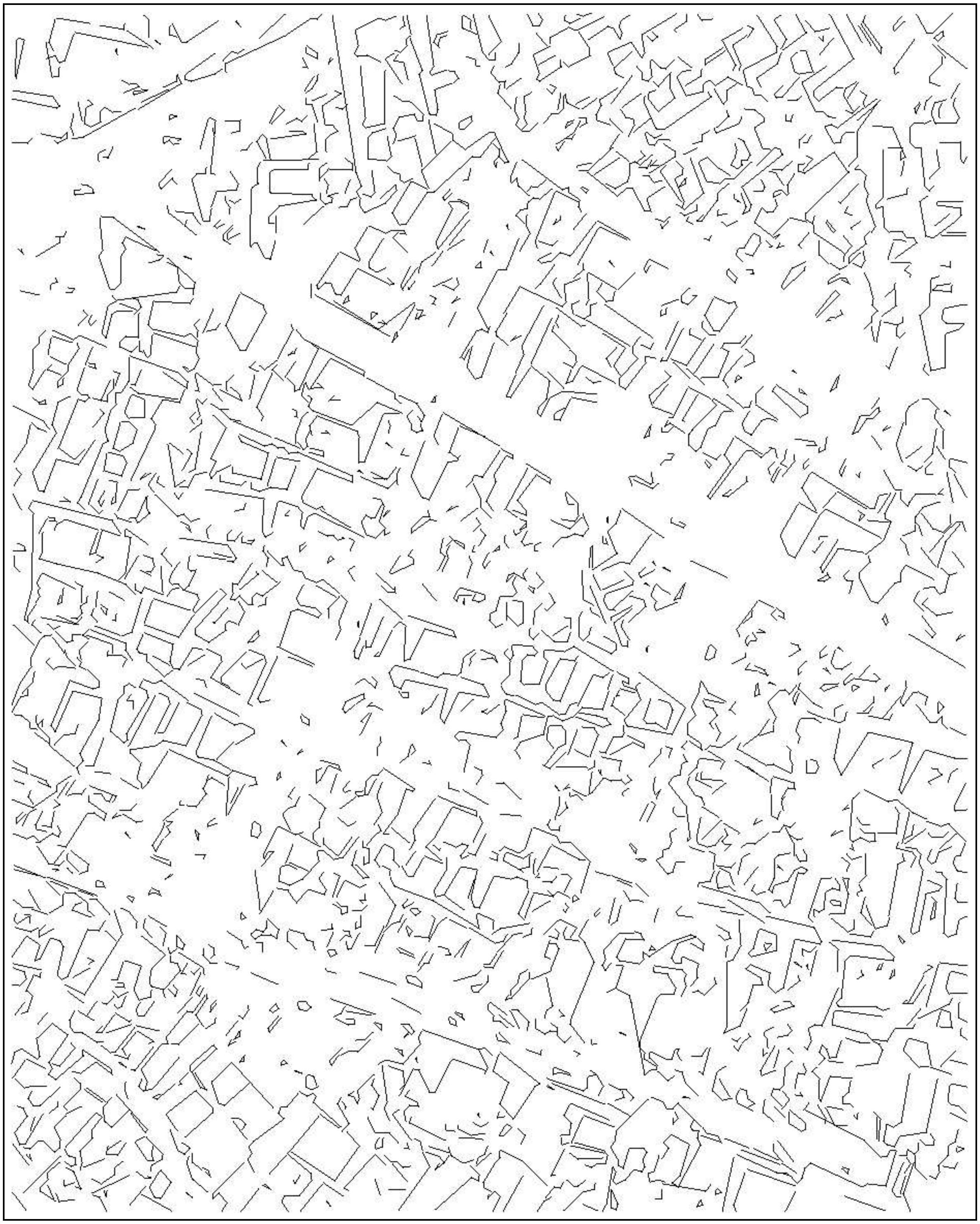


Figure 5.3: The line segments of the study area

Figure 5.4: Perceptual grouping output

north-east, the blue color and the red color depicts the corners along south-west and north-west respectively. As mentioned earlier, the cast shadows were used as an indication of a building. The shadow corners of the buildings are illustrated in figure 5.5.

As can be seen in figure 5.4, the corners were detected effectively. Most of the corners in the opposite direction of the illumination were found correctly. However, because of the small objects, a number of line segments that were not a part of a build-

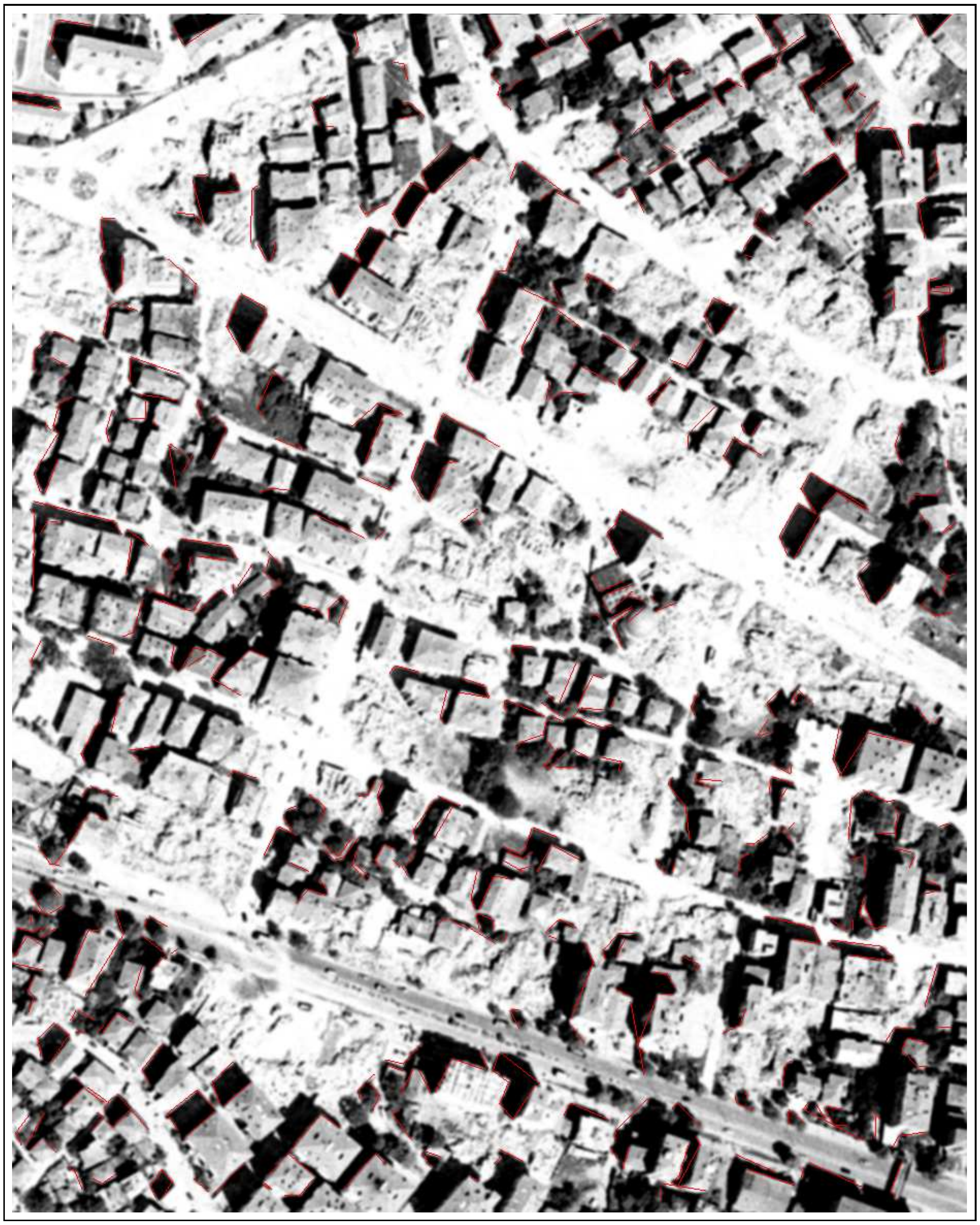


Figure 5.5: The shadow corners of the buildings

ing were grouped together to form a corner. This is illustrated in figure 5.6, where the objects labelled A, B, C and D are trees. The line segments extracted through vectorization process can be seen in figure 5.6 (b). Since, these line segments were verified by the principles perpendicularity and proximity, they were grouped together. These corner groups may be detected as a building if they form a closed counters. Therefore, these objects reduce the accuracy of the first approach. If a colored aerial image was available then, these objects should be removed. The park areas, the trees etc. may be mask out easily by using the color information. After that, the algorithm would be applied on the unmasked part of the image.

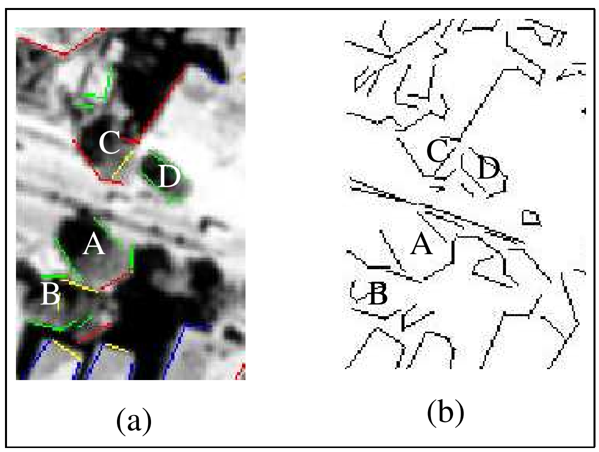



Figure 5.6: (a) The original image (b) the line segments

## 5.3 The Results of The First Approach

In the first approach, three rules were defined for building detection. The first rule verifies that whether the corners constitute a closed contour or not. This rule traces the conditions which produce perfect line segments on the edges of the buildings. The experimental results indicate that this rule works quite efficient on an area that contains sparsely located rectangular shaped buildings. As well, the area should not contain small objects which cause shadows on the buildings, such as the trees etc. On the contrary, for a dense area, the efficiency of this rule decreases sharply, which was the case in the study study area used in this thesis.

After applying the perceptual grouping process using the first rule, a total of 32 closed contours were found. However, as explained in 3.6, this information was not enough to detect all of the buildings because the cast shadows of the buildings cause the closed contours as well. Therefore, the closed contours extracted must also be checked whether they are shadow contours or not. After computing the average intensity for all the contours, 18 out of 32 closed contours were labelled as shadow contours. As a result, 14 buildings were detected with the use of this rule. Based on this result, it can be concluded that using the first rule alone is not sufficient to detect all the buildings, particularly in a dense urban area. Therefore, additional rules were defined and implemented in the study area.

The second rule traces the conditions that one or more line segments on the edges of the buildings are missing, due to the Canny edge detector that cannot detect the edge pixels. This was a quite common situation for most of the buildings in the study area. This can be seen in the output of the Canny edge detector (Figure 5.2). By using the second rule, 166 corners were grouped. However, as in the previous section, due to the shadows of the buildings and the small objects, there were a number of false positives. The shadow corner groups were 24 and the objects that are not urban structures were 12. In other words, by using the second rule in addition to first one, 130 buildings were detected.

The third rule states that if a building is uncollapsed then, it should cast shadows. If a shadow corner and the corner which is not a shadow lie in the same direction but opposite to the sun illumination direction then, this corner is an indication of a building. By applying this rule, 30 corners were grouped. However, 6 of these corner groups were false positives.

### 5.3.1 The Accuracy of The First Approach

The evaluation of the first approach is given in table 5.1. The algorithm developed detected a total of 210 buildings. Of these buildings, 42 were false-positive. The detected buildings are illustrated in figure 5.7. In the selected study area, there are 205 uncollapsed buildings. According to these results, it can be said that this approach seems to be effective for detecting the uncollapsed buildings. However, a number of small objects that are not urban structures were also detected as buildings. In other words, for a heterogeneous areas that contain small objects such as tress, rectangular structures other than the residential buildings, the accuracy of the approach may be low. To overcome this problem, color infrared images should be used. In a color infrared image, the vegetated areas can be masked out and excluded from further processing operations. The proposed approach would then be applied on the unmasked parts of the image.

One of the disadvantageous of this approach is that it may not be effective for non-rectangular shaped buildings. Since the algorithm detects the corners and then decides whether the closed contours are buildings or not the non-rectangular shaped

Figure 5.7: The Detected Buildings

Table 5.1: The error matrix for damage detection from post-event aerial photos

| | Detected Buildings | False-Positive | Total |
|---|---|---|---|
| Rule 1 | 14 | - | 14 |
| Rule 2 | 130 | 12 | 142 |
| Rule 3 | 24 | 30 | 54 |
| Total | 168 | 42 | 210 |

buildings may not be detected. On the other hand, the procedure seems to be suitable for homogeneous areas that contain rectangular shaped buildings.

## 5.4 The Results of The Second Approach

In this approach, the similarity between the line segments extracted through the vectorization process and vector boundaries were measured. Then, according to the results, the buildings were labeled collapsed or uncollapsed. As mentioned in chapter 4, three rules were defined to classify the buildings as collapsed or uncollapsed. The first rule verifies the perfect conditions, which means that all the edges of a building match a line segment extracted through the vectorization process. The second rule verifies that more than half of the edges of the buildings match a line segment. Finally, the shadow information is taken into account. This rule checks the condition that the shadow producing edges of a building match a line segment and a shadow corner exists in the opposite direction of the illumination according to shadow producing edges of the buildings.

Figure 5.8 (a) depicts a part of the study area which contains the collapsed and uncollapsed buildings while figure 5.8 (b) shows the vector boundaries superimposed on the line segments. As can be seen in the figure, the buildings labeled A and B are totally collapsed. The buildings labeled C, D, E, F and G are uncollapsed buildings. In figure 5.8 (b), the collapsed buildings are represented in red color while the uncollapsed buildings are shown in green color. As can be seen in the figure, the line segments corresponding to the edges of the uncollapsed buildings were extracted. However, there are no line segments for the collapsed buildings. Figure 5.8 (c) and (d) represent the superimposition of the line segments on vector building boundaries and the final output, respectively. Since no line segments match with the edges of building A and only two line segments match with the edges of the building B, these two buildings (A and B) were correctly classified as collapsed. As can be seen in figure 5.8, buildings C, D, E and F were correctly classified as uncollapsed. However, although building G is uncollapsed, it was labeled as collapsed, because only one of the edges of this building matches a line segment. The conditions in which the approach fails are explained detail in the following section.

The overlapping parts between the line segments and the vector building polygons and the final output of the study area are illustrated in figures 5.9 and 5.10. After labeling all the buildings according to defined rules, each building was investigated to see whether it was collapsed or not. The results of this investigation are given in table 5.2.

As mentioned in chapter 4, three rules were defined to categorize buildings as collapsed or uncollapsed. The first rule verifies the perfect conditions, which means that all the edges of a building match a line segment extracted through the vectorization process. The second rule verifies that more than half of the edges of the buildings match a line segment. Finally, the shadow information is taken into account. This

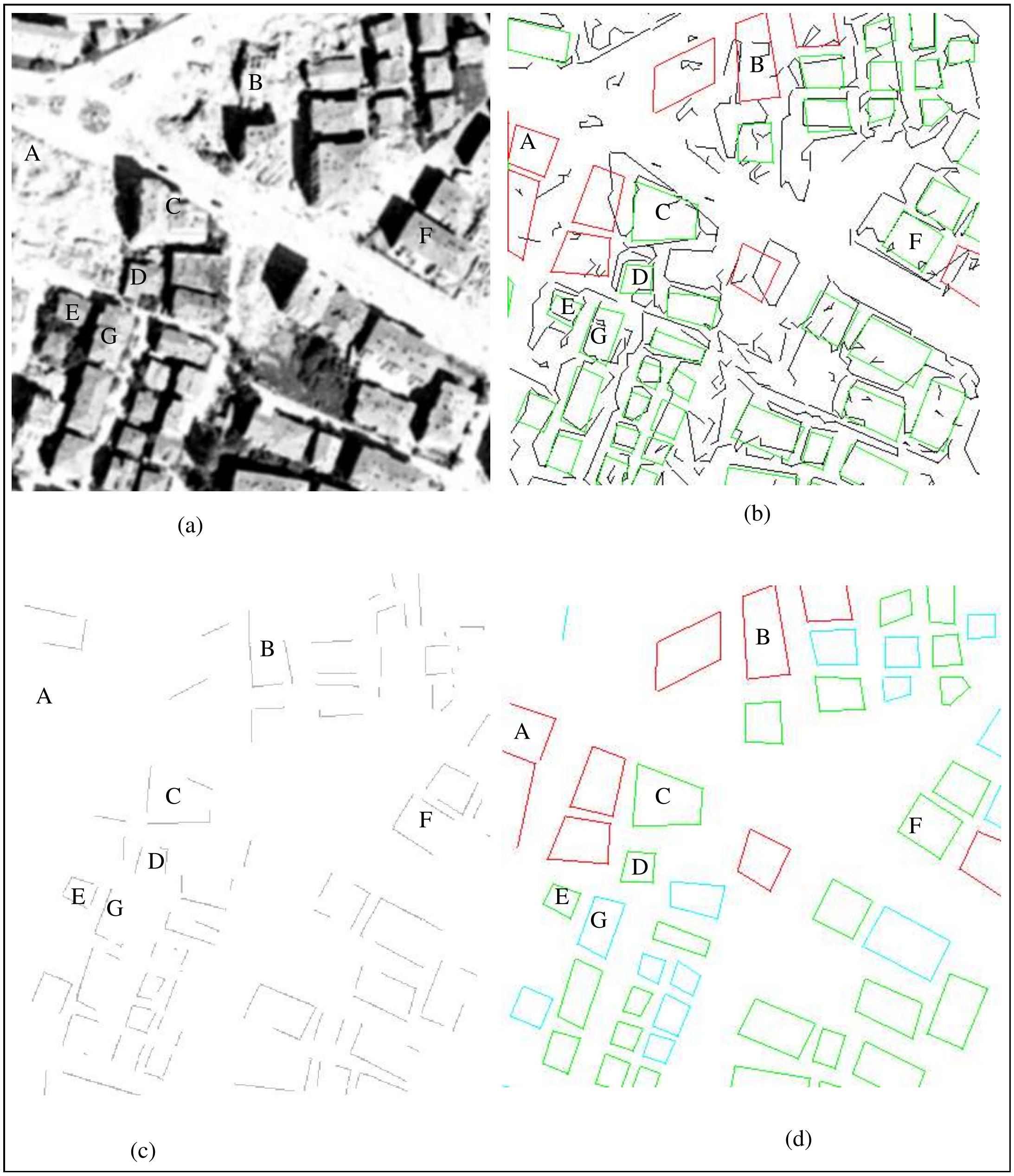



Figure 5.8: (a) The original image (b) the line segments and the vector boundaries - green represents the uncollapsed buildings, while red represents the collapsed buildings (c) the overlapping parts (d) the output image

rule checks the condition that the shadow producing edges match a line segment and a shadow corner exists in the opposite direction of the illumination according to these edges. The results are given in table 5.2. Based on these rules, to perform line matching, the overlapping parts between the vector building boundaries and the extracted line segments were computed. The results are illustrated in figure 5.9.

Table 5.2: The results of building detection after applying the rules

| | Uncollapsed | False-Positives | Total |
|---|---|---|---|
| Rule 1 | 41 | - | 41 |
| Rule 2 | 77 | 8 | 85 |
| Rule 3 | 24 | 8 | 32 |
| Total | 142 | 16 | 158 |

In figure 5.10, the buildings that were labeled collapsed and uncollapsed are illustrated in different colors. The red color depicts the correctly labeled collapsed buildings while the green color represents the correctly labelled uncollapsed buildings. On the other hand, the false-positives, and the false-negatives are represented as yellow and blue colors, respectively.

Based on the reference data, the results were evaluated (Table 5.3). The total (overall) accuracy was computed as 72.7%. The overall accuracy is the total number

Table 5.3: An error matrix for the detection of the buildings using the second approach

| | Collapsed | Uncollapsed | Total |
|---|---|---|---|
| Collapsed | 63 | 61 | 124 |
| Uncollapsed | 16 | 142 | 158 |
| Total | 79 | 203 | 282 |
| Producer's % | 79.7 | 70.0 | |
| User's % | 50.8 | 89.9 | |
| Total % | 72.69 | | |

Figure 5.9: The overlapping parts between the line segments and the vector building polygons

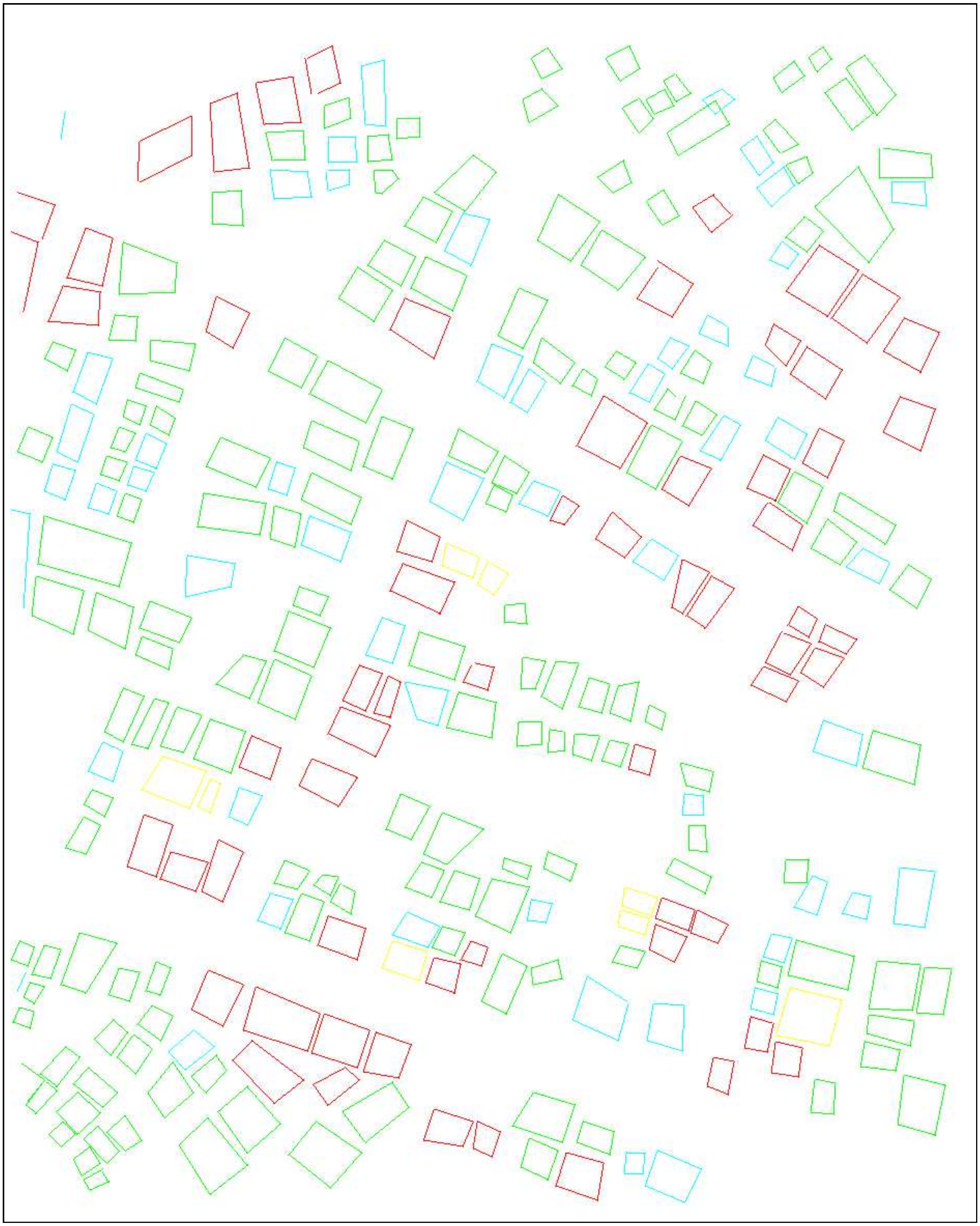


Figure 5.10: The output of the second approach

of correctly classified samples ( the sum of the elements along the diagonal ) divided by the total number of samples. The producer's accuracy was obtained as 79.7% for the collapsed and 70.0% for the uncollapsed category. This is the number of correctly classified samples of a particular class diveded by the total number of reference samples for that class. The user's accuracy was drived as 50.8% for the collapsed and 89.9% for the uncollapsed buildings. This is the number of correctly classified samples of a particular class divided by the total number of samples being classified as that class.

### 5.4.1 Discussions for The Second Approach

As can be seen in the previous section, some of the buildings were mislabeled. Of the total 79 collapsed buildings, 16 were labelled uncollapsed. Similarly, of the total 203 uncollapsed buildings, 61 were labeled collapsed. There are a number of reasons that cause this wrong categorization. In order to label the buildings, the agreement between the line segments and the edges of the buildings obtained from GIS data were measured. In other words, to label a building uncollapsed, there should be agreement between the line segments and the edges of the building. However, for 61 uncollapsed buildings, there were not line segments that match their vector boundaries. Therefore, they were labeled collapsed. In figure 5.11, two sample buildings are illustrated. As can be seen in the figure, buildings A and B were uncollapsed. However, their edges were not detected correctly. The reason for this is that the contrast between the roof of the buildings and the surroundings is low. Therefore, the edges of the buildings were not detected by the Canny detector. Because of this reason, 38 uncollapsed buildings were mislabeled.

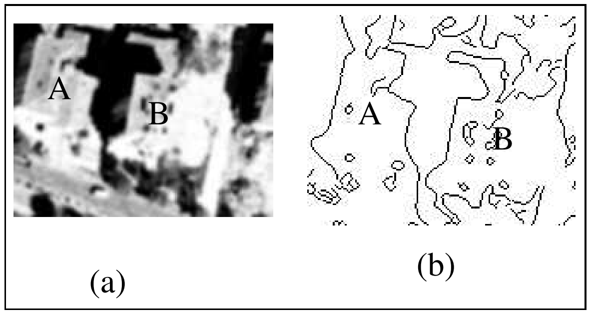



Figure 5.11: (a) The original image and (b) the edges extracted from the image

During the vectorization process, in order to find the line segments that were most similar to the edges of the buildings, the minimum number of vertexes were found. This process modified the orientation of some of the line segments. Therefore, the algorithm failed for matching the line segments with the corresponding vector building boundaries. This case is depicted in figure 5.12. As can be seen in the figure, the line segment on the right side of building B was modified. Therefore, both buildings A and B were labeled as collapsed, although they were uncollapsed. Because of this reason, 23 uncollapsed buildings were wrongly labeled collapsed.

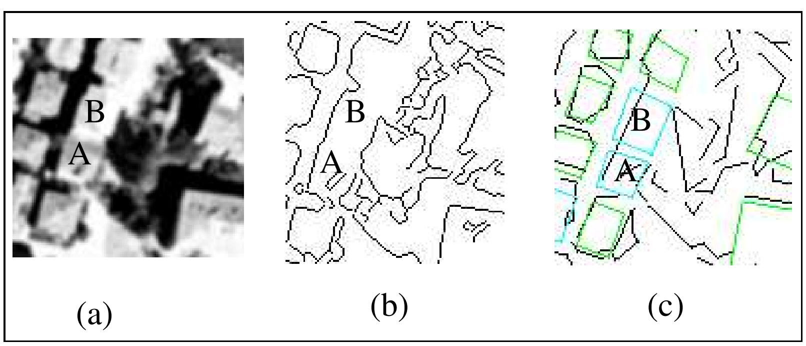



Figure 5.12: (a) The original image, (b) the edges of the image, (c) the line segments and the vector boundaries

As can be seen in table 5.3, 16 collapsed buildings were wrongly labeled uncollapsed. This is because of the locations of the buildings and the noisy lines that were detected around the small shadow areas on collapsed buildings. The line segments that correspond to an edge of a cast shadow of a neighboring building were wrongly matched with the boundaries of the current building. Similarly, the line segments extracted from the small uniform objects were matched. Therefore, the collapsed buildings were labelled uncollapsed. These conditions can be seen in figure 5.13, where there are two collapsed buildings (A and B). The line segments extracted from the edges around building A (Figure 5.13 (c)) match with the vector boundaries of this building. Therefore building A was labeled uncollapsed. For building B, a narrow cast shadow around this collapsed building (Figure 5.13 (b)) resulted in line segments. Therefore, building B was also labeled uncollapsed.

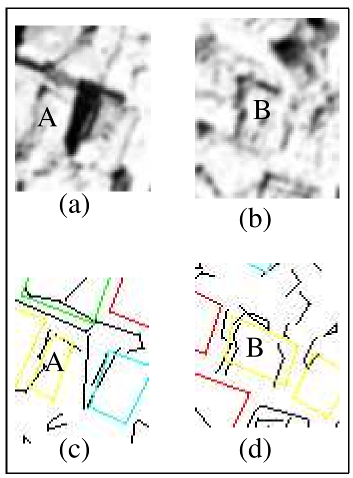



Figure 5.13: (a and b) The original images, and (c and d) the line segments with the vector building boundaries overlaid

The success of the proposed approach depends on several parameters including $\epsilon$ to find the locations of the vertexes, allowed distance and allowed orientation to group the line segments. Therefore, the accuracy of the algorithm varies depending on the user defined values for these parameters. Before applying the algorithm, the optimum values must be selected for these parameters to obtain reliable results.

According to the results, this approach seems to be quite suitable if the vector building boundaries are known a priori in addition to post-event aerial images. However, the approach may not be able to detect the non-rectangular shaped buildings. Therefore, in heterogeneous areas, the accuracy of the algorithm may decrease.

# CHAPTER 6

# CONCLUSIONS and RECOMMENDATIONS

In this study, two approaches were presented for detecting the collapsed buildings due to earthquake. Both approaches are based on the perceptual grouping and they utilize the relationship between the buildings and the cast shadows. In the first approach, only the post-event aerial photographs were used to detect the collapsed buildings. In the second approach, it was assumed that the vector boundaries of the buildings were known a priori. Therefore, this supportive information was also used for detecting the collapsed buildings.

In the first approach, of the total 203 uncollapsed buildings, 168 were correctly detected. On the other hand, there were 42 false-positives. Based on the results and the achievements during the study, the following conclusions were reached for the first approach:

- The use of the post-event aerial photographs was effective to detect the uncollapsed buildings.

- The perceptual grouping algorithm is quite effective for finding the corners and for grouping the line segments in order to form a closed contour.

- The approach is limited to detect the rectangular shaped buildings. Therefore, it may not be effective to detect the non-rectangular shaped buildings.

- This approach seems to be quite effective for those study areas, where the rectangular shaped buildings are distinctly separated each other.

- Processing only the post-event aerial photographs is not enough to detect the collapsed buildings.

- If the pre-event aerial photographs were available then, this approach could be more effective to detect the collapsed buildings due to earthquake.

In the second approach, the detection of the collapsed buildings using the existing vector building boundaries was more successful compared with the first approach. The results reveal that of the 79 collapsed buildings, 63 were correctly detected, providing 79.7% producer's accuracy. Compared with the collapsed buildings, a lower degree of agreement is evident between the assessment results and the reference data for uncollapsed buildings. Of the total 203 uncollapsed buildings, 142 were detected, providing 70.7% producer's accuracy. The overall accuracy was computed as 72.7%. The following conclusions were reached for the second approach:

- The post-event aerial photographs and the existing vector building boundaries can be effectively used together to detect the collapsed buildings due to earthquake.

- In order to make a decision about a building if it is collapsed or uncollapsed, the use of the line segments that correspond to vector boundaries of the buildings was quite effective.

- The integration of remote sensing and geographical information system (GIS) was successfully utilized for detecting the collapsed buildings.

For both approaches, several processings including edge detection, vectorization and perceptual grouping were carried out. During these processes, the following conclusions were reached:

- The accuracy of the edge detector affects the overall accuracy. Therefore, in order to detect the correct edges that represent the boundaries of the buildings, the optimum values should be selected for the parameters of edge detection.

- Several parameters were used for the vectorization and the perceptual grouping processes. For these parameters, the optimum values should be selected in order to obtain satisfactory results. For this purpose, the physical properties of the buildings falling within the study area can be used.

- The shadow corners were successfully detected through the perceptual grouping procedure. And, the use of these corners were quite effective to label the buildings as uncollapsed.

## 6.1 Recommendations

In order to increase the accuracy of the proposed approaches, a color infrared imagery can be used. By using a color infrared imagery, the vegetation can be masked out as well as those objects which differ from urban structures with their colors. Therefore, the line segments that increase the complexity in the study area can be removed.

The building models can be used to detect the buildings in the first approach. If the building models are supplied then, the line segments can be compared with these models and according to results, a number of decision rules can be defined. Further, the building block segments can be used. If the pre-event aerial image does not exist but the boundaries of the building blocks are known then, the open areas can be classified as the damaged regions based on an idea that the building blocks do not contain open areas. In addition, a region growing algorithm can be taken into account. If a uniform surface can be generated from an area labeled as a building by applying a region growing algorithm then, this area is most probably an urban structure.

As a further process, the shadow information can be used more effectively. If the height of the buildings are known and the locations of the shadows are estimated then, the line segments extracted from the cast shadows can be compared with the edges of the estimated shadow area.

# REFERENCES

[1] D. Schluter and S. Posch, "Combining contour and region information for perceptual grouping," pp. 393–401, Informatik Aktuell, 1998.

[2] R. Jhonson, "Change vector analysis for disaster assessment: A case stduy of hurricane Andrew, remote sensing images and technical notes," pp. 41–45, Geocarto International, 1, 1994.

[3] A. Singh, "Digitial change detection techniques using remotely sensed data," pp. 989–1013, International Journal of Remote Sensing, 10, 1989.

[4] R. Jensen and D. Toll, "Detecting residental land-use development at the urban fringe," pp. 629–643, Photogrammetic Engineering and Remote Sensing, 48, 1982.

[5] G. Byrne, P. Crapper, and K. Mayo, "Monitoring Land-cover Change by Principal Component Analysis of Multitemporal Landsat Data," pp. 175–184, Remote Sensing of Environment, 10, 1980.

[6] W. Malila, "Change Vector Analysis: An Approach For Detecting Forest Changes with Landsat," pp. 629–643, Proceedings of the Sixth Annual Symposium of Machine Proccessing of Remotely Sensed Data, 1980.

[7] J. Howarth and G. Wickware, "Procedure For Change Detection Using Landsat Digital Data," pp. 277–291, International Journal of Remote Sensing, 2, 1981.

[8] P. Gamba and F. Casciati, "GIS and image understanding for near-real-time earthquake damage assesment," pp. 987–994, Photogrammetric Engineering and Remote Sensing, 64, 1998.

[9] M. Ishii, T. Goto, T. Sugiyama, H. Saji, and A. Keiichi, "Detection of earthquake damaged areas from aerial photographs by using color and edge information," Dept. of Computer Secience,Faculty of Information,Shizuoka University.

[10] M. Turker and B.T. San, "SPOT HRV data analysis for detecting earthquake-induced changes in Izmit,Turkey," pp. 2439–2450, International Journal of Remote Sensing, 24, 2003.

[11] M. Turker and B. Cetinkaya, "Automatic detection of earthqauke damaged buildings using DEMs from pre and post-earthquake stere aerial photographs," International Journal of Remote Sensing, in press.

[12] M. Turker and B.T. San, "Detection of collapsed buildings caused by the 1999 Izmit,Turkey earthquake through digital analysis of post-event aerial photographs," International Journal of Remote Sensing, in press.

[13] R. Irvin and D. McKeown, "Methods for exploiting the relationship between buildings and their shadows in aerial imagery," pp. 1564–1575, IEEE Transactions On Systems, Man and Cybernetics, 19, 1989.

[14] A. Huertas and R. Nevaita, "Detecting changes in aerial views of man-made structures," pp. 583–596, Image Vision Computing, 18, 2000.

[15] P. Mather, Computer Processing of Remotely-Sensed Images. John Wiley & Sons, 1989.

[16] D. Ballard and M. Brown, Computer Vision. Prentice-Hall, 1982.

[17] N. Efford, Digital Image Processing. Pearson Education Limited, 2000.

[18] F. Ackermann, A. Masmann, S. Posch, G. Sagerer, and Schlüter, "Perceptual grouping of contour segments using markov random fields," pp. 11–17, Pattern Recognition and Image Analysis, 7, 1997.

[19] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: A tutorial," pp. 157–192, IEEE Trans. on Circuits and Systems, 43, 1996.

[20] W. Shin and M. R. G. Marque, "A non-self-intersection douglas-peucker algorithm," State University of Campinas, 1999.

[21] G. Jenks, "Geographic logic in line generalisation," pp. 27–42, Cartographica, 26, 1989.

# APPENDIX A

# STRUCTURES USED IN THE SOURCE CODE

```
struct bina {

        int id;

        short int durum;

};

struct bina binalar[500];

typedef short int Bool;

struct point {

        int x;

        int y;

};

struct kosep {

        struct point pt;

        short int bakis;

        int yakin;

        int uzak;
```

```
        int dik;
        int yatay;
        struct point duzak;
        struct point dyakin;
        struct point yyakin;
        struct point yuzak;
        short int golge;
};
struct line_of_loop {
        int lineid;
        struct point next;
};


struct lpoint {
        struct point pt;
        struct lpoint *ptr;
};
struct linep {
        struct point start;
        struct point end;
        double m;
        double distance;
        double constant;
        double theta;
```

```
        double r;

        int id;

        int taken;

        int deleted;

        double xd;

        double yd;

};

struct line {

        struct point end;

        struct point start;

        struct lpoint *ptr;

};
```

# APPENDIX B

# A PART OF THE CODE IMPLEMENTED IN THE STUDY

```
saglamcount=0;
for ( i=0; i < numofvector; i++ ){
  int k,edge[7],t;
  count=0;
  uygunkose=0;
  j=0;
  for ( t=0 ; t < 7 ; t++){
    edge[t] = -1;
  }
  while ( vector_data[i][j].start.x > -1 ) {
    if ( (k=findLine(linearr,vector_data[i][j],
        linecount,img_renkli,W,H)) ) {
      count++;
```

```
    edge[j] = k;

  }

    j++;


}

j=0;

if ( count == 2 ) {

  short int dur=0;

  for ( u =0; u < cornernum && !dur; u++ ) {

    if ( findDistance2(kose[u].pt,vector_data[i][0].start)

    < 35 && kose[u].pt.x < vector_data[i][0].start.x

      && kose[u].golge ) {

      ayakta++;

  if ( binalar[i].durum == 0 ) {

    falsep++;

  }

      saglam[i]=SAGLAM;

      dur=1;

      j=0;

    while ( vector_data[i][j].start.x != -1  ){

      if ( vector_data[i][j].start.x > 0 &&

        vector_data[i][j].start.y > 0 &&

        vector_data[i][j].end.x >0 &&

        vector_data[i][j].end.y > 0 ){
```

```
      drawLineRenkli(img_son,vector_data[i][j].start,
        vector_data[i][j].end,250,0,250,W,H);
      }
      j++;
    }
    }
  }
}
j=0;
if ( ( count > 2 ) ){
  saglam[i]=i;
  while ( vector_data[i][j].start.x != -1  ){
    if ( vector_data[i][j].start.x > 0 &&
      vector_data[i][j].start.y > 0 &&
      vector_data[i][j].end.x >0 &&
      vector_data[i][j].end.y > 0 ){
      printf("i %d durum :%d  \n",i,binalar[i].durum);

      if ( binalar[i].durum == 0 ){
      drawLineRenkli(img_son,vector_data[i][j].start,
        vector_data[i][j].end,0,0,250,W,H);
      }
      if ( binalar[i].durum==1 ) {
      drawLineRenkli(img_son,vector_data[i][j].start,
```

```
          vector_data[i][j].end,250,0,250,W,H);
         }


       }
       j++;
     }
     ayakta++;
     if ( binalar[i].durum == 0 ) {
       falsep++;
     }
   }
   else if ( saglam[i] != SAGLAM ) saglam[i]=0;


 }
 for ( i=0; i < numofvector; i++ ){
   if ( !saglam[i] ) {
     j=0;
     while ( vector_data[i][j].start.x > -1  ){
       if ( vector_data[i][j].start.x > 0 &&
         vector_data[i][j].start.y > 0 &&
         vector_data[i][j].end.x >0 &&
         vector_data[i][j].end.y > 0 ){
         if ( binalar[i].durum == 1 ){
         drawLineRenkli(img_son,vector_data[i][j].start,
```

```
          vector_data[i][j].end,250,0,0,W,H);
        }
        else {
        drawLineRenkli(img_son,vector_data[i][j].start,
          vector_data[i][j].end,2,250,250,W,H);
        }


      }
      j++;
    }
    if ( binalar[i].durum == 1 ) falsen++;
    yikik++;
  }
}
writePNG(argv[5],img_son,ImgWidth,ImgHeight,24);
printf("ayakta degil %d yikilmis degil %d ayakta buldu
  %d yikilmis buldu %d \n", falsep,falsen,ayakta,yikik);
```