

AN AUTOMATED BUILDING EXTRACTION MODEL USING FUZZY  
K-NN CLASSIFIER FROM MONOCULAR AERIAL IMAGES

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

BY

ÇAĞLAR ŞENARAS

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
CIVIL ENGINEERING

SEPTEMBER 2007

Approval of the thesis:

**AN AUTOMATED BUILDING EXTRACTION MODEL USING FUZZY K-NN  
CLASSIFIER FROM MONOCULAR AERIAL IMAGES**

submitted by **ÇAĞLAR ŞENARAS** in partial fulfillment of the requirements for  
the degree of **Master of Science in Civil Engineering Department, Middle  
East Technical University** by,

Prof. Dr. Canan Özgen

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

Prof. Dr. Güney Özcebe

Head of Department, **Civil Engineering**

Assoc. Prof. Dr. Mahmut Onur Karslıoğlu

Supervisor, **Civil Engineering Dept., METU**

Prof. Dr. Fatoş Yarman Vural

Co-Supervisor, **Computer Eng. Dept., METU**

**Examining Committee Members:**

Asst.Prof.Dr. Murat Güler (CE, METU)

Assoc. Prof. Dr. Mahmut O. Karslıoğlu (CE, METU)

Prof. Dr. Fatoş Yarman Vural (CENG, METU)

Prof.Dr. Kerim Demirbaş (EE., METU)

Dr. Uğur Leloğlu (TÜBİTAK-UZAY)

**Date: 14.09.2007**

**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 : Çağlar Şenaras

Signature :

## **ABSTRACT**

### **AN AUTOMATED BUILDING EXTRACTION MODEL USING FUZZY K-NN CLASSIFIER FROM MONOCULAR AERIAL IMAGES**

Şenaras, Çağlar

MS., Department of Civil Engineering

Supervisor : Assoc. Prof. Dr. Mahmut Onur Karslıoğlu

Co-Supervisor : Prof Dr. Fatoş Yarman Vural

September 2007, 59 pages

The aim of this study is to develop an automated model to extract buildings from aerial images. The fuzzy k-Nearest Neighbor classification method is used to extract the buildings by using color information. Also, the advantages of the relevance feedback systems are discussed. The model is implemented in C# and a graphical user interface is developed for the user interaction. The model is evaluated in 5 different test areas with more than 600 buildings.

Keywords: building extraction, fuzzy k-NN, feedback

## ÖZ

### BULANIK ÖĞRENME YÖNTEMLERİ İLE HAVA FOTOĞRAFLARINDAKİ BINALARIN OTOMATİK BULUNMASI

Şenaras, Çağlar

Yüksek Lisans, İnşaat Mühendisliği Bölüm

Tez Yöneticisi : Assoc. Prof. Dr. Mahmut Onur Karslioğlu

Ortak Tez Yöneticisi: Prof. Dr. Fatoş Yarman Vuraş

Eylül 2007, 59 sayfa

Bu çalışmanın amacı hava fotoğraflarından binaların bulunması için otomatik bir model geliştirilmesidir. Bu model bulanık öğrenme yöntemleri ile uydu fotoğraflarındaki renk bilgisini kullanarak binaları bulmaya çalışmaktadır. Aynı zamanda kullanıcıdan aldığı geri besleme bilgileri ile zamanla hatalarını azaltıp, başarı performansını arttırmayı amaçlamaktadır. Model 5 farklı test alanında denenmiş ve performansı 600 den fazla bina üzerinde test edilmiştir.

Anahtar Kelimeler: uzaktan algılama, bina sezme, geri besleme, bulanık öğrenme yöntemleri

## **ACKNOWLEDGMENTS**

I am deeply indebted to my supervisor Assoc.Prof.Dr. Mahmut Onur Karslıoğlu, for his guidance, patience and motivation. I am sincerely in debt to him for his continuous support throughout this study.

I would like to thank to Prof. Dr. Fatoş Yarman Vural. I am fortune to be one of her student. She not only taught me a lot of things on computers sciences, but also affect my aspect of view to the life. I am sure that I would not be in such position without her guidance, motivation and delicious pies.

I also would like to thank to Özge, Cüneyt and all other image lab assistants for their valuable assistance and patience.

Finally, I would express my special thanks to my family and Eda for their patience and compassion throughout the study.

To My Parents

## TABLE OF CONTENTS

ABSTRACT .....	iv
ÖZ.....	v
ACKNOWLEDGMENTS .....	vi
TABLE OF CONTENTS .....	viii
CHAPTERS .....	<b>Error! Bookmark not defined.</b>
1.INTRODUCTION .....	1
1.1 Definition of Problem .....	1
1.2 Previous Works .....	3
1.3 Purpose of Thesis .....	4
2.BASIC CONCEPTS.....	5
2.1 Texture Features.....	5
2.2 Polygonal Approximation .....	6
2.3 Morphology .....	7
2.4 Chapter Summary .....	10
3.CLASSIFICATION.....	11
3.1 Unsupervised Classification Methods .....	12
3.2 Supervised Classification Methods.....	13
3.3 Chapter Summary .....	17
4.DEVELOPING THE MODEL .....	18
4.1 Feature Selection .....	18
4.2 Description of the Proposed Model .....	25
4.3 Overview of the Software Package.....	33
5.EXPERIMENTS AND RESULTS .....	38
5.1 Define Evaluation Criteria .....	38
5.2 Define Test Areas .....	40
5.3 Experiments .....	43
5.4 Results And Discussions .....	49
6.CONCLUSION.....	54
REFERENCES .....	56



# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Definition of Problem**

Extraction of geospatial data from the photogrammetric sensing images becomes more and more important with the advances in the technology. Today Geographic Information Systems are used in a large variety of applications in engineering, city planning and social sciences. Geospatial data like roads, buildings and rivers are the most critical feeds of a GIS database. For decades, this extraction of geospatial data process has been performed manually. However manual extraction is a very slow operation method and requires well-trained operators. The invested time and labor extremely increase the cost of operation. Because of these problems pattern recognition and photogrammetry communities are trying to find better solutions to replace human effort by automated systems.

Buildings in aerial images are the main type of geospatial data considered in the thesis. However, extracting this type of data is one of the most complex and challenging tasks [2]. The main problem is the variety of the type of buildings. The type of colors and also the shapes of the buildings are very inconstant. Also in some areas, the buildings are placed irregularly or too close to each other. Developing an automated model to overcome these problems is aimed in the thesis.

The degree of automation is one of the critical factors, which determines the cost of the effort spent by manual annotation of the geographical maps. Therefore, one needs to answer the following questions.

- What is the type of automation?
- What is the amount and type of human interaction?

This solution can be achieved in four different levels [4].

1. *Interactive systems*: All of the measurement tasks are done manually. There is no automation.

2. *Semi-automatic systems*: There is an interactive environment where human play a critical role in measurements and the automatic modules are integrated into main workflow. This interaction from human is required also when determining the parameters for the automated modules.

3. *Automated systems*: The main tasks are automatically performed by the system. The user interaction goes in for project setup before the automatic phase starts and also for visual verification, post editing and correcting the errors of the automated steps [4].

4. *Autonomous systems*: The system without any human interaction even for post-editing and extract the all requested objects after the first click. However the today's technology is far from this level of automation in cartographic maps.

Most of the recent researches and commercial software can be defined as a 2<sup>nd</sup> or 3<sup>rd</sup> level automatic system. In the following section a brief summary about the previous works will be presented.

## **1.2 Previous Works**

Cartographic feature extraction systems can be categorized by the type of sensor data used. Some of the researches concentrate on the fusion of more than one data sources. These methods usually use the advantages of height information in 3-d data set. One common approach is to use more than one aerial or satellite images and getting the height information using photogrammetric calculations [6, 7, 8, and 9]. Also using new technologies such as LIDAR, which provides high vertical accuracy and high point density, becomes popular. Some of these make use of the fusion of LIDAR and satellite image [10]. Also some approaches exist which only use the digital surface model calculated from the LIDAR data [11].

Another important category of the systems extracts objects by using monocular aerial or satellite imagery. Most of them are based on image processing methods. Tuytelaars (1998) presents an iterated Hough transform as a method for extracting regular geospatial objects. Jung and Schramm use windowed Hough transform to detect rectangular shapes on an image and perform experiments on aerial images to detect buildings. Wei and Zhao (2004) introduce an approach, where they first cluster the satellite image using an unsupervised learning method and use the shadow information to verify the existence of building. Then, for each building boundary, Canny operator is used for extracting edges and finally system detects lines using Hough transform. Mayungaa (2005) works on an active contour model which is commonly known as a snake algorithm for a semiautomatic building extraction method. In this method the user has to click to the approximate center of each building then the algorithm generates the border of this building. Stassopoulou and Caelli (2000) first segment the image into regions

and calculate the area and the mean gray-level intensity of these regions. Finally, they use Bayesian Networks to combine the information and classify the regions which are parts of a building. Chen and Blong (2002), use red/green/blue bands and also some simple texture features such as the mean value to extract buildings. They combined these inputs in an artificial neural network to classify pixels and then define building regions. Edge information is also used at the end of method to improve the classified image.

### **1.3 Purpose of Thesis**

The purpose of this thesis is to develop an efficient model for the automated systems to extract buildings. For this purpose, the performance of pixel based classification in Fuzzy k-NN is employed for segmentation of critical images. Then, a user interface, which allows adding, editing, and removing objects, is developed. The major contribution of this thesis is to introduce a relevance feedback mechanism to improve the result of the automatic annotation. As a result, the model can learn from the misclassifications by using the post-editing data and feedbacks the automated system.

The thesis organization is organized as follows:

In the second chapter, polygonal approximation methods, which are required for a better presentation and converting from raster to vector data and also morphological operators commonly used in image processing, are presented. In the third chapter some of the popular unsupervised and supervised classification methods like artificial neural network and k nearest neighbor method are introduced. The fourth chapter presents selection of features, implementation of the model and the software in detail. The last two chapters include experiment results and the conclusion.

## **CHAPTER 2**

### **BASIC CONCEPTS**

In this chapter some of the basic concepts, which are employed in this thesis, will be introduced. The first section overviews the texture features considered in chapter 4. The second section describes the polygonal approximation techniques used for representing the shape information in the aerial images. The last section explains some of the popular morphological functions for editing the images. These concepts have a great importance to understand the rest of the thesis.

#### **2.1 Texture Features**

As a basic definition, Texture is a function of the spatial variation in pixel values. This function is very useful in many image processing applications. Recognition of objects is one of the main applications of texture. For example, if we consider an image of a wooden surface, there is more than one variation of brown color and this significant spatial variation is the most important visual clue for the recognition of wood [18]. Because of these fundamental characteristics, texture information can be used also for recognizing the geospatial objects. A great number of researches on texture analysis are conducted. Some of the research is based on statistical methods [36]. These methods characterize an image or a region in terms of numerical features which consist of a vector. They can be derived from histogram information of images, co-occurrence matrices and run length coding. Also some approaches are based on

signal processing methods. Haar transform is one the most common methods in approaches based on signal processing [32, 35].

In the thesis, several texture descriptors are implemented and tested for the classification of geographical images. Some of these features and their results for the model are described in Chapter 4 in details.

## **2.2 Polygonal Approximation**

In most of the image classification systems, polygons are used to represent object boundaries. This representation is used for both smoothing the boundary surface and also compressing the data purposes. The building extraction models also need the polygonal approximation approach to represent the buildings boundary in the user interface and save them in a GIS data format.

The polygonal approximation can be basically defined as follows:

Let an object  $X$  has a boundary and the  $P_1 = \{p_{11}, p_{12}, p_{13} \dots p_{1n}\}$  is a set which includes the ordered set of points of this boundary. If another set of points  $P_2 = \{p_{21}, \dots p_{2m}\}$  exists, such that the elements of  $P_2$  is less than  $P_1$ 's and the differences between  $P_1$  and  $P_2$  is acceptable, then  $P_1$  is approximated by  $P_2$ . [19]

In other words, the aim of polygon approximation is to represent the initial boundary with a relatively small set of vertex points without losing the critical shape information. There are mainly two types of approaches. First one assumes the approximation as a minimum error problem. The initial polygon  $P_1$  with  $n$  vertices is approximated to a new polygon  $P_2$ , whose number of vertices is predefined as  $m$ , so that the approximation error is minimized. The other approach is using minimum vertex to present the polygon. The algorithm

approximates the polygon  $P_1$  by polygon  $P_2$  with fewest vertices until the error becomes smaller than a predefined tolerance value.

## 2.3 Morphology

Basically, the mathematical morphology can be defined as the study of shapes. This important tool is commonly used in image processing applications for various tasks. In this thesis morphology operations are employed on object boundaries to smooth houses. Also it is used for eliminating small holes, filling gaps, shrinking or enlarging objects during the pre-processing or post-processing stages.

The mathematical morphology is based on the set theory. The sets in this tool represent the shape of objects in the image and Minkowski operators are used for morphological processes. Minkowski addition and dilation, is one of the simple and important set transformations. The main purpose of dilation is expending a set, in other words enlarging the boundaries of region, by a pre-defined structuring element [20, 21]. Mathematically this transformation can be defined as follows:

Suppose that,  $A$  is the set of coordinates for the input of a binary image and  $S$  is the set of coordinates corresponding to the structuring element. Let  $\hat{S}$  be the reflection of  $S$  about its origin and  $S_x$  denote the translation of  $S$  so that its origin is at  $x$ . Then, dilation of  $A$  by  $S$  is the set of all points  $x$ , such that the intersection of  $A$  with  $\hat{S}_x$  is non-empty [22]:

$$A \oplus S = \{x | \hat{S}_x \cap A \neq \emptyset\} \quad (2.1)$$

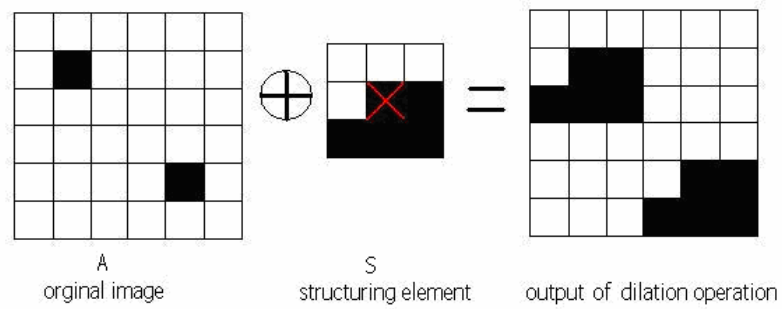


Figure 2.1 Example of dilation operation For the given original image, erosion operator gives the output using the structuring element.

Minkowski subtraction is another common operator used in mathematical morphology. This operator is, also, called as erosion. The main purpose of erosion is to shrink the boundary of a region by a pre-defined structuring element. Mathematically, erosion of A by the structuring element S is the set of all points x such that  $S_x$  is a subset of A.

$$A \ominus S = \{x \mid S_x \subseteq A\} \quad (2.2)$$

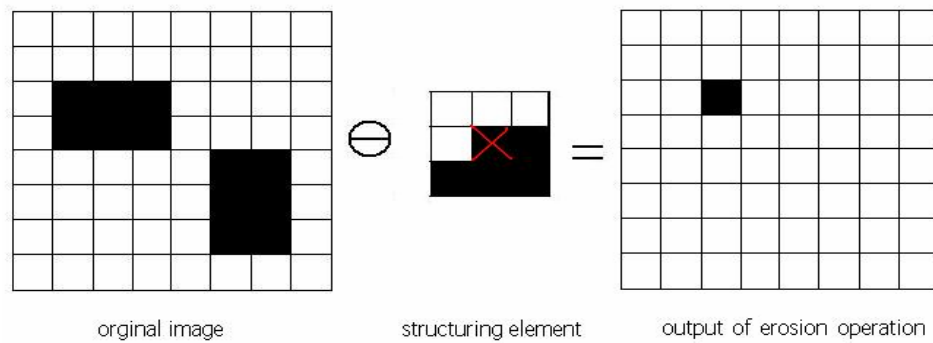


Figure 2.2. Example of erosion operation. For the given original image, erosion operator gives the output using the structuring element.

Also opening and closing are two important operations derived from erosion and dilation. The main purpose of opening is to smooth the



contour of an image by breaking narrow connection points or eliminating thin protrusions. On the other hand, closing is used for filling gaps, eliminating small holes or fusing narrow breaks [21].

The opening is defined as,

$$A \circ S = (A \ominus S) \oplus S$$

(2.3)

Which means opening of A by S is equal to dilation of the set, the result of erosion of A by S, by the structure element S again.

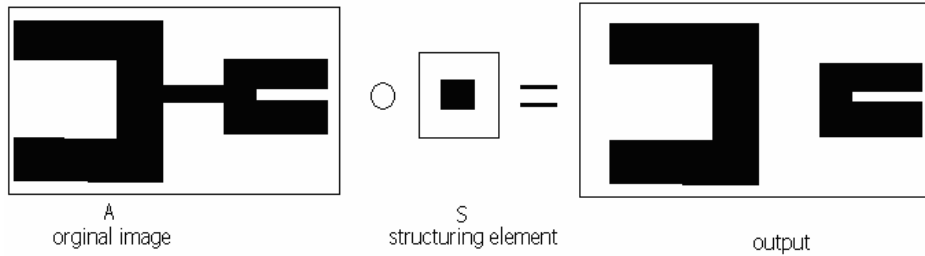


Figure 2.3. The example of opening operation the image. The opening operation is applied to the original image A using the structuring element S.

The closing of set A by S is defined as,

$$A \bullet S = (A \oplus S) \ominus S, \quad (2.4)$$

which means that the dilation of A by S is followed by the erosion by S is equals to closing of A by S.

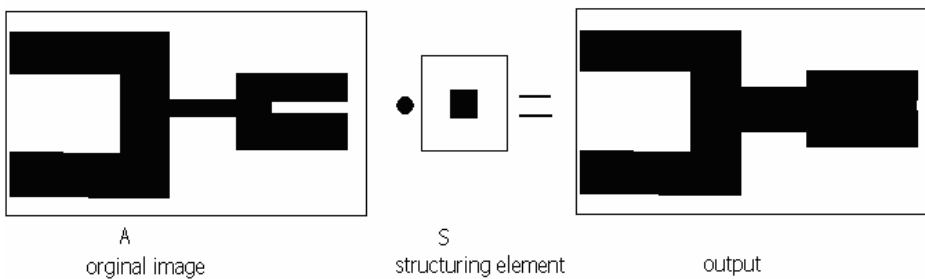


Figure 2.4 The example of closing operation the image with structuring element S.

There are also other types of operators in mathematical morphology such as skeletonization and thinning however these type of operators do not used in the thesis.

## **2.4 Chapter Summary**

In this chapter, some of the basic concepts which are employed in processing of the aerial images are reviewed. Also polygonal approximation concept, which is used for representing an initial polygonal shape with a relatively small set of vertices without losing the critical shape information, is presented shortly. Finally, the some of the morphological operators, which are used in the thesis for post-processing, are explained.

## **CHAPTER 3**

### **CLASSIFICATION**

Basically, object classification in aerial images assigns an object to a pre-defined category such as roads, houses and fields. During this process, an identifier called classification method is used. Some characteristic information is extracted from the categories using a set of descriptor, called feature vector, is fed to this classifier and determines the class of the object in the aerial image. Also some classifiers give a membership value for each object category instead of assigning a sample to the exact category. These methods can be either probabilistic or fuzzy approaches.

For aerial images, one of the major problems in classification is the vast amount of variations in feature vectors extracted from the objects of each class. The distinction among the feature vectors for the same category and also the similarity between the feature vectors of objects in different categories like trees and buildings makes the classification, a very complex problem. Also, the noise of any misapprehended property for a sensed data due to randomness of the world or sensor is a difficulty for the classification problem.

The difficulties mentioned above prevent using simple threshold methods or guessing the best classification decision ahead of time. Therefore, classification methods, which use a learning schema to create its own decision, need for extracting the buildings in an aerial image. These learning schemas can be performed either by an unsupervised manner or by the supervision of a teacher.

In the following sections, these learning approaches will be discussed. Moreover, for both of the learning approaches, some of the popular methods will be provided.

### **3.1 Unsupervised Classification Methods**

In some applications, only a collection of sample points, like several aerial images without their class information is provided and the classification has to be done only by using these images. Also, in some cases, unlabeled data is preferred because of the high cost of labeling a large set of sample data. Several unsupervised learning methods are developed to overcome the labeling operation. Clustering one form of unsupervised learning is commonly used for this purpose in remote sensing applications since the category information is not available. The main idea of clustering is to partition the sample data set into smaller sets, so that each subset contains a similar type of data in itself and also different from the other subsets. In this approach it is aimed to assign the similar objects in an aerial image to the same cluster. The similarity ties and differences are defined by some distance metrics. [24]

K-means clustering is one of the popular clustering methods. The main purpose of K-means is to partition the sample data set into k-cluster. For this purpose, the following procedure is applied.

1. Place randomly K points into the feature space from the sample dataset. These initial k-points are the initial centroids of clusters.
2. Add all points to the feature space.
3. Assign the points according to the nearest centroid.
4. Recalculate the new centroid of the clusters.

5. If the new centroid is changed return to step 3 and assign the sample points using the new centroids.
6. Stop.

## **3.2 Supervised Classification Methods**

When the labeled data is available, learning methods are supervised by these labeled samples. In this study, a set of training data is selected and the class labels are manually assigned to enable supervision. The training data contains sample objects and their desired classification labels. By using the training data, a model that maps the input samples to the class labels, is generated. After the training, the model predicts the label for an unknown sample. [26]

### **3.2.1 K-Nearest Neighbor Method**

K-nearest neighbor classification is one of the most popular methods. In spite of its simplicity it achieves high performance in many applications. The main idea for this classification method is to assign a sample point by examining the class labels of the nearest k-point.

In this supervised learning method, the training phase locates feature vectors of the sample objects into the sample space and labels their class information. In the classification phase, for each new object whose label is not known, its k-nearest neighbor is detected in the feature vector space. Then, by applying a majority voting for these neighbors, its class label is predicted.

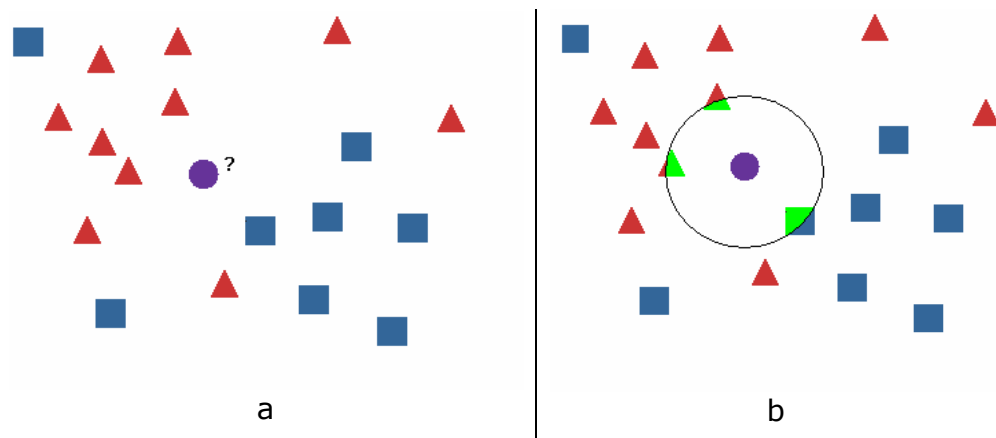


Figure 3.1 Classification with using k nearest neighbor algorithm. A given unlabeled circular object in Figure 3.1.a is classified where k is selected 3. 3 nearest neighbors of the unlabeled objects are shown in Figure 3.1.b 2 of which are triangles. Algorithm classifies it as a triangle after the majority voting.

Selecting the parameter k is important for the performance of algorithm. Because of the majority voting, usually odd numbers are selected. During the experiments it is shown that, the increment in the value of k up to 11 increases the performance of the classification because this makes the model more robust to noisy data. However, increasing the number of neighbors to be considered, increase the complexity of algorithm which means to run slower.

### 3.2.2 Fuzzy K-Nearest Neighbor Method

The classical K-Nearest Neighbor method assigns each object into a crisp class as a result of traditional set theory. However, the precision for a decision may not be warranted. Let's consider a two class classification problem where an object lies on the boundary. In such a case, the object is assigned to one of the classes without considering; it could be a member of other class. One of the common approaches is to use fuzzy sets to resolve this type of

ambiguities. Fuzzy set theory allows appraising the elements and giving them membership values in the real unit interval (value between 0 and 1) [25].

Fuzzy k-nearest neighbor is a classification technique, which provides the simplicity and the practicability of classical K nearest neighbor and also the advantages of using fuzzy logic. Fuzzy set theory gives a value for how close to the class for an unknown sample instead of a crisp membership of the class [27].

The main algorithm is very similar to K-NN. In the training procedure, the sample objects are located to the feature vector space and these samples are initialized in a fuzzy state. There are mainly two types of initializing:

1. Standard Crisp Initialization: Assign each sample to a class membership vector which is  $U_x = \{u_1, u_2, \dots, u_n\}$  where n is the number of class.  $u_j$  is assigned to 1 if x belongs to class  $c_j$ , otherwise it is assigned to 0.
2. K init Nearest Neighbor Initialization: For each sample data x, its k neighbors are found and then x's class membership value for each class j is assigned.

$$u_j(x) = \begin{cases} 0.51 + (n_j / K_{init}) \times 0.49, & \dots j = i \\ (n_j / k_{init}) \times 0.49, & \dots j \neq i \end{cases} \quad (3.1)$$

Where  $x \in \text{class}_i$ .

In the classification phase, for each new object x, it's K nearest neighbors are detected and then class membership values are calculated according to the following formula:

$$u_i(x) = \frac{\sum_{j=1}^K u_{ij} W_j}{\sum_{j=1}^K W_j} \quad (3.2)$$

Where,  $u_{ij}$  is the  $i^{\text{th}}$  class membership value of neighbor $_j$  and

$W_j$  is the weight of neighbor $_j$ :

$$W_j = \left( \frac{1}{\text{distance}^{\frac{2}{m-1}}} \right) \quad (3.3)$$

The value of  $m$ , used to scale the effect of the distance between  $x$  and neighbor $_j$ , is entirely arbitrary. As  $m$  approaches to  $\pm$  infinity, the results of classifier approach to K-nn. Kelle and Gray claim that, using  $m=2$  is a good approximation according to the results of their experiments, however they did not explain any theoretical or experimental data.

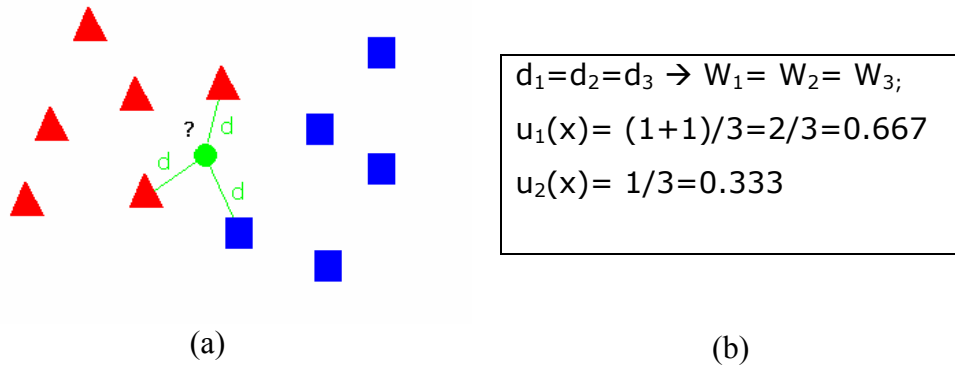


Figure 3.2. A classification example of an unlabeled object with using fuzzy k nearest neighbor algorithm. In the figure (a), the unlabeled green circular object's 3 neighbors are at the same distance. As displayed in the figure (b), the weights of these three neighbors are same due to the same distance. Figure 3.2.b shows the calculation of class membership values.



Fuzzy K-NN approach reduces the disadvantages of traditional K-NN approach and also it is a simple and effective solution for classification.

### **3.3 Chapter Summary**

In this chapter, the classification methods employed in this thesis is presented. Classification, assigning an object to a pre-defined category, is the main part for recognizing of an object. The classification methods can be categorized into two depending on the learning type. The first one is unsupervised learning, where the sample points does not have any class information. K-means is one of the popular unsupervised classification methods. The second type is supervised method, where the learning procedure is performed by the supervision of a teacher. The kNN and fuzzy kNN are two popular supervised learning methods which are discussed above.

## **CHAPTER 4**

### **DEVELOPING THE MODEL**

This chapter is devoted to the Automated Building Extraction with Relevance Feedback (ABE-RF). In the first section, different types of features, which are tested for the best performance, are discussed. The next section presents ABE-RF in details. Finally, the last chapter introduces the implemented software.

#### **4.1 Feature Selection and Classification**

For an effective building extraction method, the features which are used for classification are very important. Whichever classification method is used, if the selected features do not distinguish the differences and similarity of objects, classification method fails.

In this thesis, the performances of different types of features are tested on the fuzzy kNN method explained in the previous section. The method mainly classifies each pixel and creates homogenous regions using the class information. Thus, the selected feature has to be extracted for each pixel in the image.

Before starting the analyses, the training files are manually labeled and saved as an image file. During the analyses, features are implemented and their performances are tested by fuzzy K-NN in Matlab. First, training images and their labels are loaded and for each pixel its feature vector is calculated. Fuzzy K-NN is initialized

using this training data. Then, a test image is loaded and each pixel is classified by using its feature vector. Finally, the pixel labels are visualized as a new image. The ABE-RF has 4 main classes. Two of them are devoted to the different type of building. The other two classes are use for roads and other background objects like grass and trees. Blue color indicates the first type of buildings class, which include buildings with tile roofs; red ones indicates the second type of building class, buildings with more complex roof colors; green color indicates road classes and finally black color shows the background class.

#### 4.1.1 HSB Color Feature

Color is one of the most useful information in an image. To represent this information in a mathematical way, several models, called color spaces, are used. The HSB color space, which is one of the very important and attractive color models for image processing, represents colors very similar to color sensed by humans.

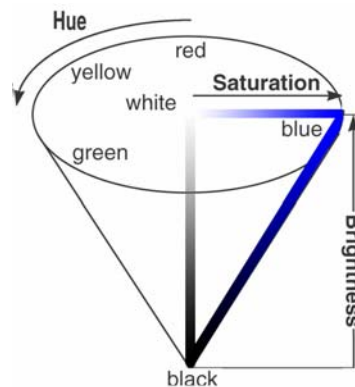


Figure 4.1. HSB color space. H stands for Hue, S stands for Saturation and B stands for brightness

In the HSB color model, Hue component is an angular measurement and describes the pure colors. 0 degree indicates the color red; 120 degree indicates green color and 240 degree means blue. Saturation component describes the colorfulness and ranges from 0 to 100%. The value of 0 indicates the color in grayscale and 100 indicates intense color. Brightness refers the amount of white in color. Also it ranges from 0 to 100% where 0 indicates black and 100% indicates the full of white [28].

In the analyses for each pixel, the values of the hue, saturation and brightness are calculated and normalized between 0 and 255 for an objective comparison. These values are used as a feature vector. Some examples of classification are given in figure 4.2 and 4.3

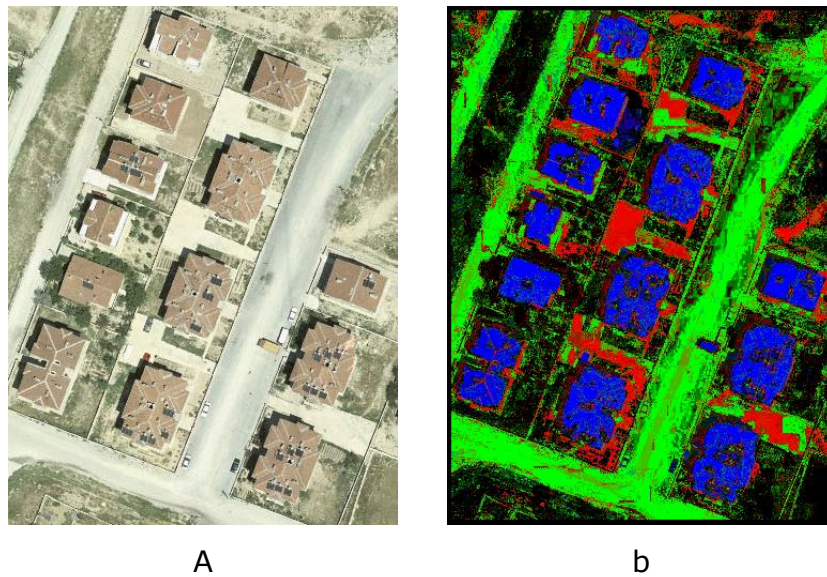


Figure 4.2 (a) Original image (b) The output of Fuzzy K-NN using Hue, Saturation and Brightness where blue pixels indicates type 1 buildings; red pixels indicates second type of buildings ; green pixels indicates the road and black represents other background objects.

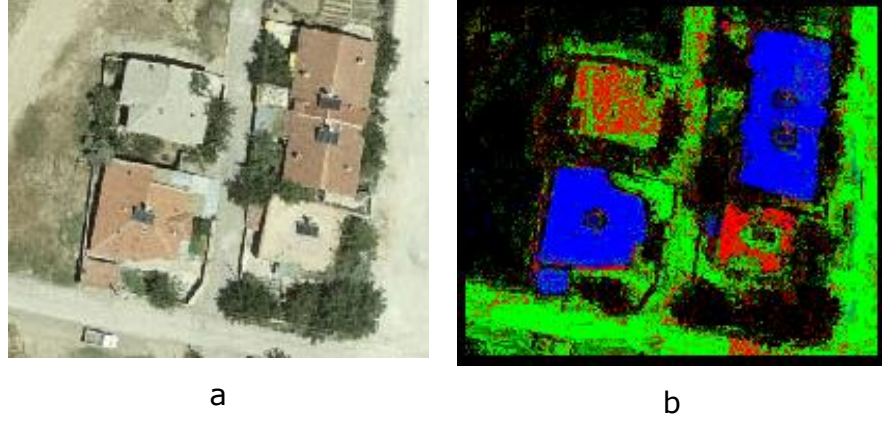


Figure 4.3 (a) Original image (b) The output of Fuzzy K-NN using Hue, Saturation and Brightness.

#### 4.1.2 Histogram Features

Histogram is the relative frequency of the pixel colors in a predefined window of an image. This basic representation is very useful for image processing to distinguish regions with different color characteristics. Also, it is used for classification in different types of applications as a basic texture descriptor [29]. In this thesis, a simple histogram feature is examined for classification of the pixels.

For each pixel  $P$ , the histogram of a small sub image with center  $P_x$ ,  $P_y$  and with a size of  $N \times N$  is generated. In this generating step, pixels of sub image, also called window, are used. The range of hue is divided into 15 bins and each pixel is put to the correct bin according to its hue value. Finally a feature vector is created:

$$F_{hue} = \{M_1, M_2, M_3, \dots, M_{15}\}$$

$$\text{Where } M_i = \frac{\# \text{ of pixels in } i^{\text{th}} \text{ bin}}{\# \text{ of total pixels}}$$

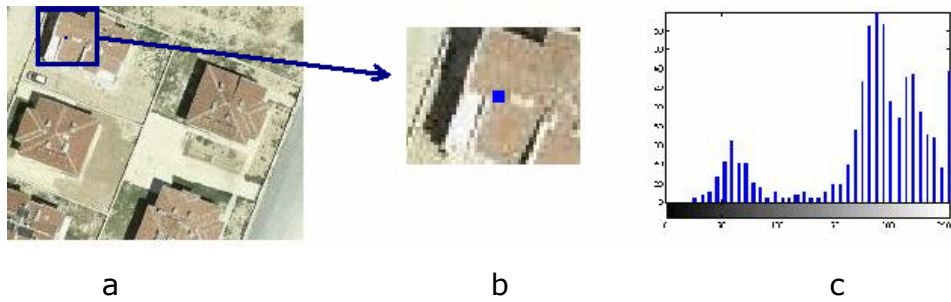


Figure 4.4. Demonstration of determining histogram feature. (a) The original image. (b) The window centered by the target pixel. (c) The histogram of the given window.

Also saturation and brightness values of pixels are used for creating a histogram. Histogram of these three components appended consecutively for creating new feature vectors. And their visual results are given in Figure 4.5. The histogram of the hue is a sufficient clue to detect most of the buildings with tile roofs; however it is not enough to distinguish the gray and the roads. Using the saturation and brightness information helps us to solve this problem and reduces the rate of the false alarm.

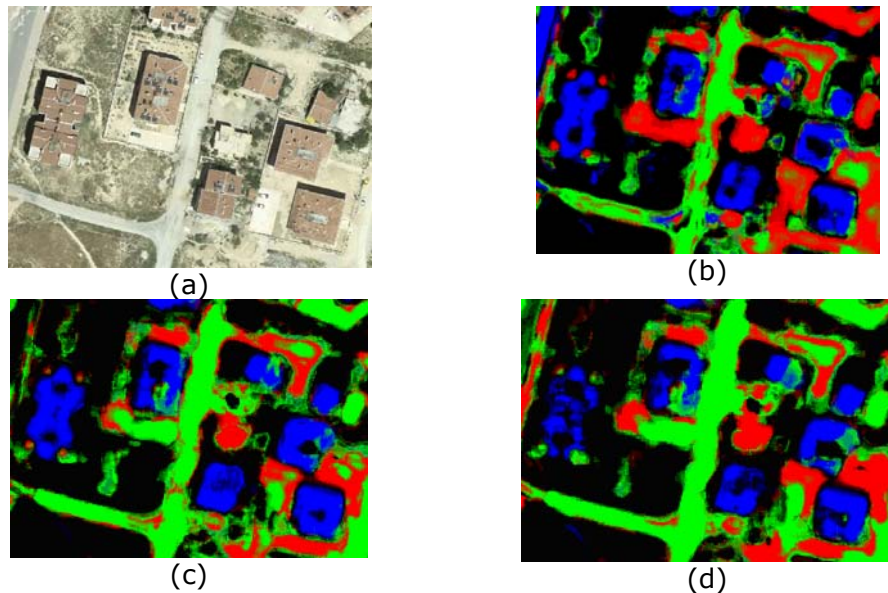


Figure 4.5. (a) The original image and classification results with using the given features in the order of (b) hue, (c) hue + saturation, (d) hue + brightness

### 4.1.3 Edge Histogram

Edge histogram feature is commonly used for extracting texture information [30] in pattern recognition. The basic idea is the extraction of the spatial distribution of edges for a given image. To extract the feature vector, first the image is divided into 16 sub-images. For each sub-image the members of 5 edge category: horizontal, vertical, 45°, 135° and isotropic, are counted and a histogram with 5 bins is created. The collection of all 80 bins (5 bins for each one of the 16 sub-images) forms the edge histogram feature vector.

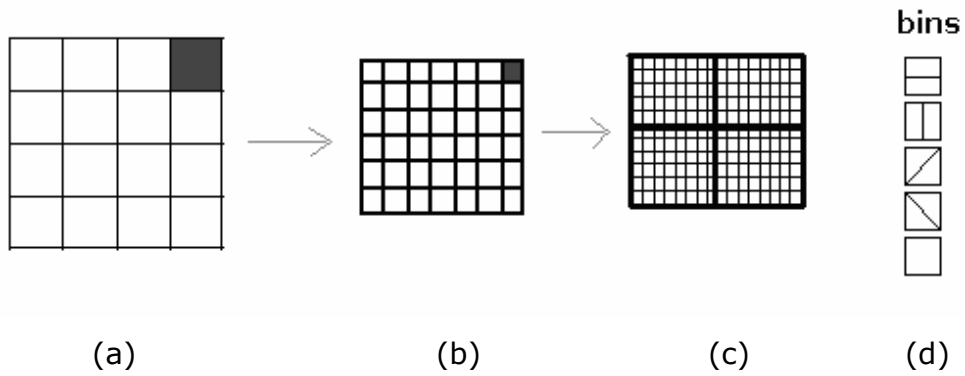


Figure 4.6 . Computing the edge histogram descriptor. The figure (a) illustrates the original image, which is divided 4x4 sub-images. Figure (b) illustrates the smaller blocks which are generated by dividing the sub-image into smaller block. In the figure (c), a smaller block, which is assumed as a 2x2 window, is given and the type of window edge type according to this assumption. Figure (d) shows the type of edges. Each detected edge increase the member of the related bin.

An implementation of the edge histogram is written in Matlab for this thesis [31]. In order to check its correctness the feature vectors are matched with other edge histogram implementations. Although this descriptor is commonly used, during the experiments it is observed that the edge histogram descriptor is not suitable for the desired model. In the training part the pixels which are near to



boundary surfaces with different labels cause noises because of very similar feature vectors and this makes the results very irrelevant.

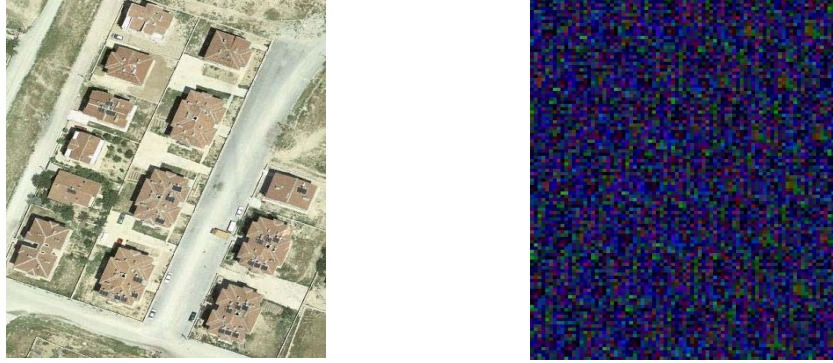


Figure 4.7. Sample classification result for edge histogram descriptor.

#### 4.1.4 Feature selection decision

In this thesis 5 feature descriptors are analyzed with different window size and different K values. Also, a Haar descriptor, which is implemented by Emre Akbaş according to the article [32], is used for classification. Although the outputs are meaningful, the performance over the experiments is not better than the other descriptors. Moreover, this descriptor runs slower than histogram and simple color feature.

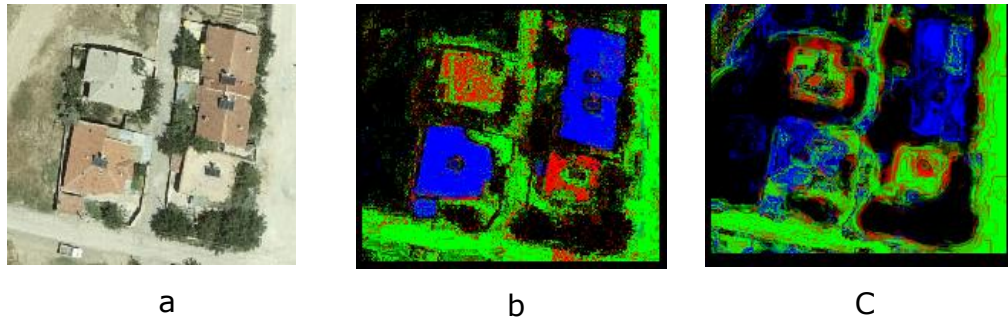


Figure 4.8. Output results of HSB color feature (fig. a) and Haar feature (fig. b).



After the visual analysis, it seems that the methodology of the model prevents to use some of the popular texture features like edge histogram. On the other hand, the simple texture features like color histogram and Haar descriptor do not provide a significant performance increment. So it is decided to use the HSB color feature due to its performance, simplicity and smaller feature vector size. The rest of this thesis is based on the results of HSB color features.

## **4.2 Description of the ABE-RF**

As described previously, the main aim of this thesis is developing an automated model for building extraction from remote sensing images, and visually presenting them in a user interface, where the user can edit these extracted information. Moreover, creating a relevance feedback mechanism, which uses edits made by the user, is one of the objectives. Also, the complexity of the algorithms included in the model is very important. The run time of the model has to be practically acceptable in the underlying application domain. During the developing stage of the model, these criteria mentioned above are elaborated.

The ABE-RF includes 4 main parts. The first part deals with the classification of pixels in the remotely sensed image using the fuzzy K- nearest neighbor method. The second part extracts the objects, which are the buildings, in the image. This task is accomplished by using class membership values of the pixel in the training space. The third part displays the buildings in a graphical user interface for interacting with the user. In this part, the user checks the results of the automatic classification and manually edits the detected building. The last part is the feedback mechanism, which employs the serious information from the user and re-trains the system accordingly.

Like the other supervised learning systems, before starting the classification, the fuzzy K-nearest neighbor classifier has to be trained. The model allows two different ways to train the system. In the first approach, remotely sensed images are manually labeled. Then, classifier reads each pixel together with its class value, extracted from the labeled file. In this labeled file, color values are used for each class, as described in Chapter 4.1. The model uses hue, saturation and brightness values as a 3-dimensional feature vector and creates the training space with these feature vectors. The second approach for completing the training procedure is using an already created train space. This property provides support for saving the experiences of the classifier for further usage.

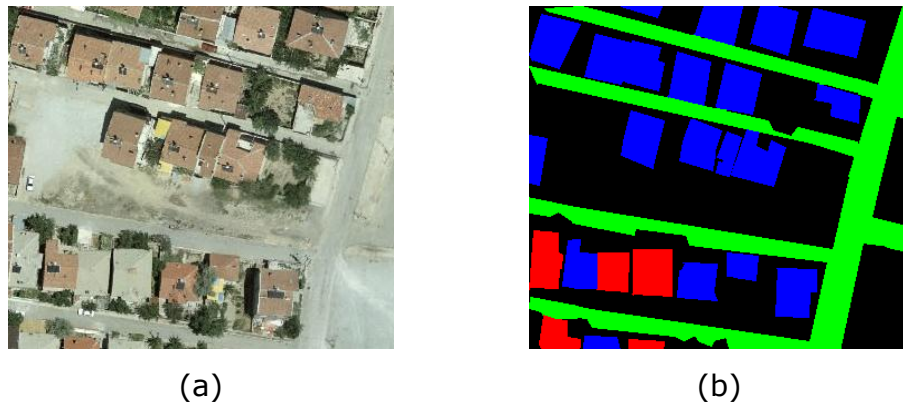


Figure 4.9. Training the classifier using manually labeled images. The figure (a) is the original image; the figure (b) is the manually labeled file of the original file.

In the proposed model, the fuzzy k-NN method is modified in order to overcome large training dataset. The classical approach checks every sample points in the data set to find the nearest ones. However, this makes the system slower with every added training image and editing made by the user. In order to make the program independent of the runtime with the training data size, the sample space is divided into sub-spaces. And each sub-space stores the

number of samples for each class in itself. So, for each training data instead of adding a point to the space with the selected class label, the proposed method just finds the related sub-space and increases the counter for this particular class.

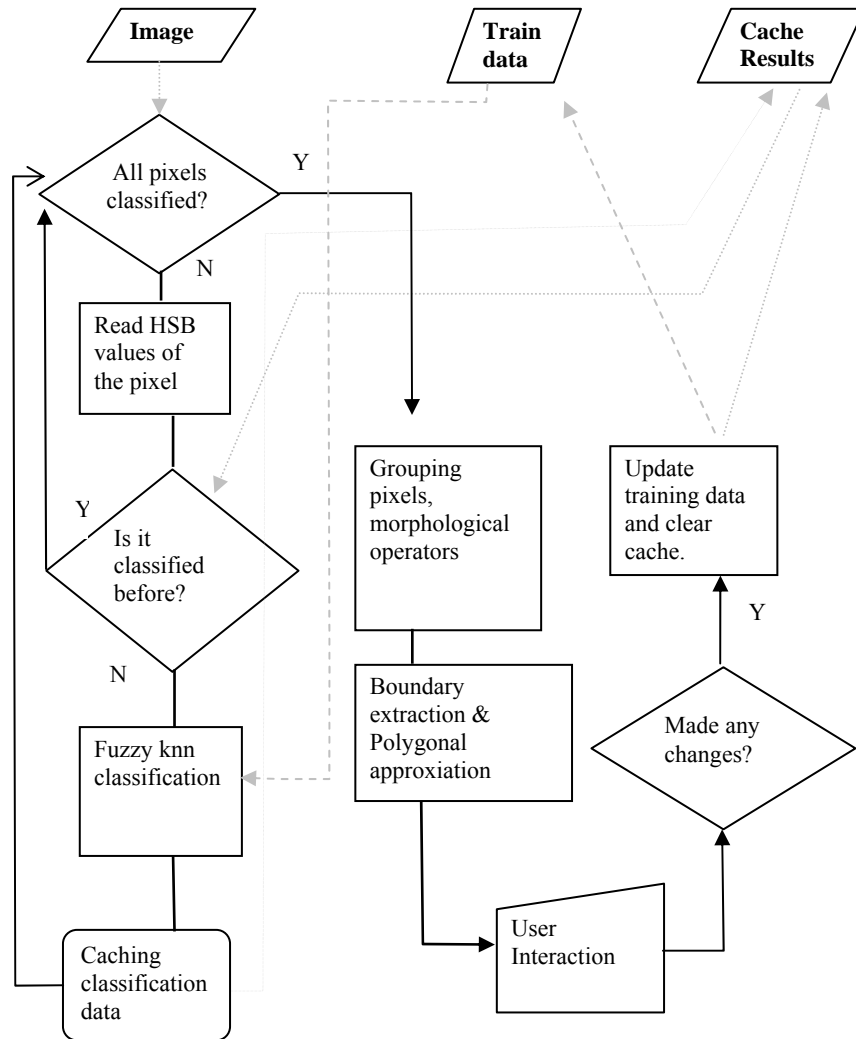


Figure 4.10. Flow diagram of the ABE-RF.

After the model is trained, the user requests to extract the buildings in the image. The model starts to read color information from the upper left pixel and read each pixel's HSB values, uses them as a feature vector. The classifier finds the K nearest sub-space and calculates the class membership values by considering

both the distances of neighbor sub-spaces and also the number of elements in these sub-spaces. The class membership of point  $X$  for the class  $i$  is calculated according to the following formula

$$u_i(x) = \sum_{j=1}^K \frac{W_j \times M_{ji}}{M_j}$$

Where  $M_{ji}$  is the number of samples in the neighbor sub-space  $j$  with the class label  $i$  ;  $M_j$  is the total number of samples in neighbor  $j$  and  $W_j$  is the weight of the sub-space derived from (3.3).

Moreover, a cache mechanism is designed for a better performance. For each HSB value, the model checks the cache memory. If membership class values are available in the cache for this vector, the model directly uses these values otherwise classifies with fuzzy knn and stores the result in the cache for further requests.

The next step is detecting the buildings using the classification result. Membership values of building classes are considered for this purpose. In order to remove noises, first a mean filter is applied to the membership values. The idea of the mean filtering is simply to reassign each pixel values with the average value of its neighbors' values. The kernel, which defines the size and the shape of the neighborhood, is selected as a 7x7 square in the model. After the mean filter is applied to all pixels, the pixels with very small membership values are neglected and one of the popular morphological operators, the opening operator is applied with a same sized kernel. Detailed information about this operator is given in Chapter 2.3. A sample output for these processes is given in Figure 4.11.

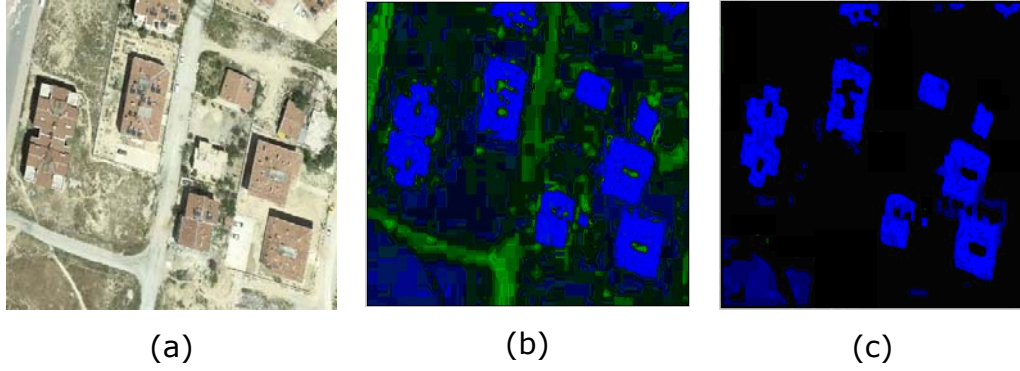


Figure 4.11. The figure (b) is the fuzzy knn classification result of the original aerial image given in figure (a). After the classification, the model applies threshold, morphological operators and creates the output in figure (c).

After the simple image processing filters, the model searches for groups of pixels whose membership values are more than a threshold. The threshold value is selected depending on the type of building class. For building class with tile roofs, the classifier is more confidential. So, pixels with a less membership values can be accepted as a building. On the other hand, the feature vectors for the class of buildings with complex roofs is very similar to class of road's feature vectors which causes a requirement for a high membership value to confide.

Pseudo code for the second step is as follows:

```

Function FindObjects( $U_i$ )
{
    MeanFiltering( $U_i$ )
    Erosion ( $U_i$ )
    Dilation( $U_i$ )
    Foreach unlabeled pixel, x
        If  $u_i(x) > \text{threshold}$ 
            {

```

```

        CreateNewObject(Objn)
        AddMember (x , Objn)
        If MaxArea > Obj.Area > MinArea
        {
            FillHoles( Objn)
            FindBoundary( Objn)
            FindCorners(Objn)    // Polygon Approximation
        }
    }
}

```

In the thesis, object term is used to refer to the group of pixels. After detecting each object, its area is calculated and if the area of the object is in the limit of acceptance, then the object is accepted as a building. The model prepares these accepted buildings for presenting them in the user interface. First, the holes, i.e. pixels which can be detected as non-building due to dish antennas or reservoirs, are filled. Secondly, the boundary information for the building is extracted. Finally, these boundaries are approximated to the polygons for a better and more compact representation. The polygonal approximation allows the user to easily edit and connect the output of the automatic classification. It is important to approximate the polygon with the minimum number of vertices and also it is required to preserve the original shape. Here the Iri-Imai algorithm is chosen due to its simplicity and available implemented source codes [33]. These source codes are modified according to the requirements of the model.

The polygonal approximation algorithm constructs a graph  $G$ , whose nodes correspond to members of boundary of the building. In this graph two nodes are linked to each other only if the vertices

between them satisfy an error tolerance which is related to the distance between two vertices.

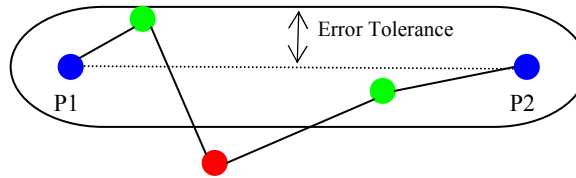


Figure 4.12. 1 of 3 vertexes between P1 and P2 is not within the tolerance. Thus, the nodes of P1 and P2 can not link to each other.

Each link, called an edge, has a weight, which shows the number of nodes between the parent nodes. After calculating the weights, the algorithm uses a standard shortest path to find the maximum weighted path from the start point to the end point. This shortest path is the optimal approximation for the given chain.

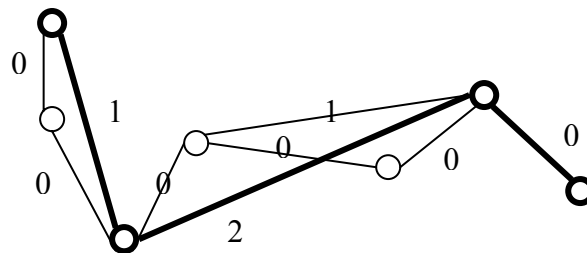


Figure 4.13. A sample graph for the Iri-Imai algorithm.

However, this method has some disadvantages: Firstly, it is a brute force algorithm, which is not a quick solution. Secondly, the starting and ending points are defined before the approximation. The algorithm approximates the polygon considering these points and if the points are not determinative, then the approximated polygon will not be optimal. In order to overcome these problems, the algorithm is modified. Actually, a set of pre-processing technique

can resolve the problems for detecting more determinative boundary points and using the points as the nodes of the graph.

The algorithm of pre-processing is inspired from [34] and [19]. The idea is that the sharp curvatures are more significant to form the shape of the polygon. In order to determine the significance of the curvature at boundary points, their K-slopes, defined as the slope between  $K^{\text{th}}$  left neighbor, point, and  $K^{\text{th}}$  right neighbor, are calculated for different K values between  $N/8 - N/24$ . Then, peak points are selected according to the average of K-slopes. Using the average of different K values prevents spurious peaks. Although this inspired approach seems enough to approximate a polygon, the result of the tests done in this thesis, shows that required number of peaks to represent the original shape can be excessive up to 30. This large number makes the model useless for the editing step. For this reason, first the pre-processing finds the peak 30 point, then these selected peak points are given to Iri-Imai algorithm and a polygon is approximated with this algorithm. Also the initial point problem for the Iri-Imai problem, as mentioned above, is solved by using these peak points.

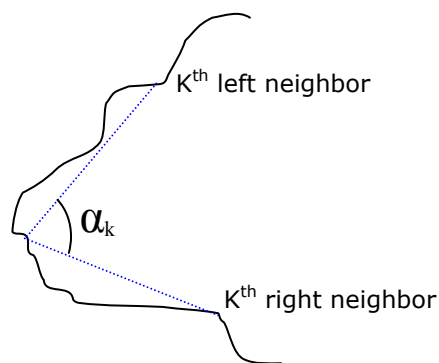


Figure 4.14 . Illustration of K-Slope



In the third step, the model displays the created polygons over the original image and, like other automated systems, allows user to manually edit these results. The user can reshape, delete and add building in this step; also redefining the class of building is allowed. Detailed information about the step will be explained in the Chapter 4.3.

The last part of the ABE-RF is the feedback mechanism, which is also very important for a better recognition rate for further runs. Like all other building extraction models, the performance for extraction is not fully satisfactory. Some of the buildings are missed, and also some of the non-building objects are classified as a building. The aim of this part is attempting to change the sample space for the fuzzy k nearest neighbor by using feedback information and reducing the misclassification problem. For this purpose, after user finished the editing part, the model compares the current polygons with the previous polygons which are saved before editing and finds the misclassified pixels. Then, for each misclassified pixel, the model finds related sub-space using the HSB value and increases the counter for the desired class in this sub-space.

### **4.3 Overview of the Developed Software**

The ABE-RF is implemented in C# using Visual Studio 2005. The captured videos of the program are uploaded to <http://www.ce.metu.edu.tr/~caglar>. These representative videos demonstrate how to use the software. However in this part of the thesis brief information is given about the software.

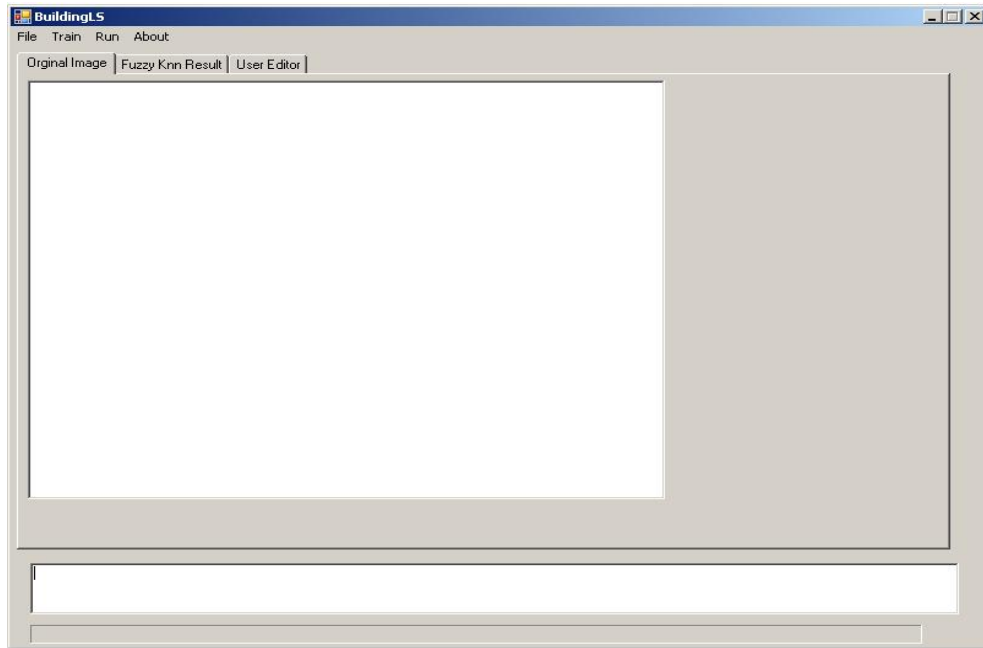


Figure 4.15. The first screen of BuildingLS.

Before starting to extract building, the user has to train the system from the *train* menu. The user has two alternatives. The first one is selecting “from labeled image” and using manually labeled files as described in chapter 4.2. The other one is selecting “from data file” and loading previously used sample space. Also from this menu the user can save sample space for further usage purposes by selecting “save data file”.

After the system is trained, the user loads the desired remote sensing image for extracting buildings from the File menu. Also from this menu the user can capture a screen shot of detected buildings or save them in XML format. For the current version, BuildingLS does not support any GIS XML format, but this module can be imported for the next versions.

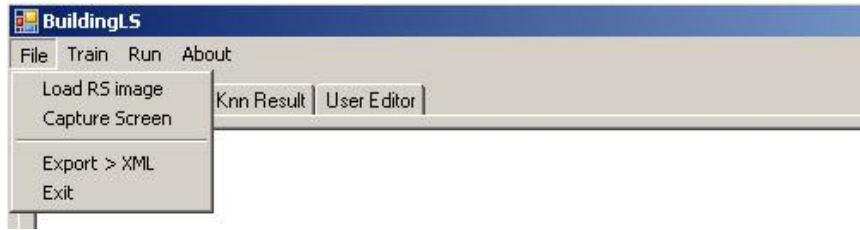


Figure 4.16. File menu items

The extraction begins with the clicking *run* from the menu. At the bottom of the window, a process bar shows the percentage of the completed parts. When processing finishes, the text box at the bottom shows the number of extracted buildings.

The user interface is divided into 3 tabs:

1- Original Image: This tab shows the remote sensing image without any polygon or modification.

2- User Editor: This tab is designed for the user interaction, which is very important for an automated system. In the left of the tab, extracted buildings are presented in a layer using polygons. The selected building is presented with a red polygon, others are with green polygons. Also each vertex of the polygon is presented by a 5x5 red square. The user can reshape the polygons with the moving vertex or deleting a vertex by right clicking.

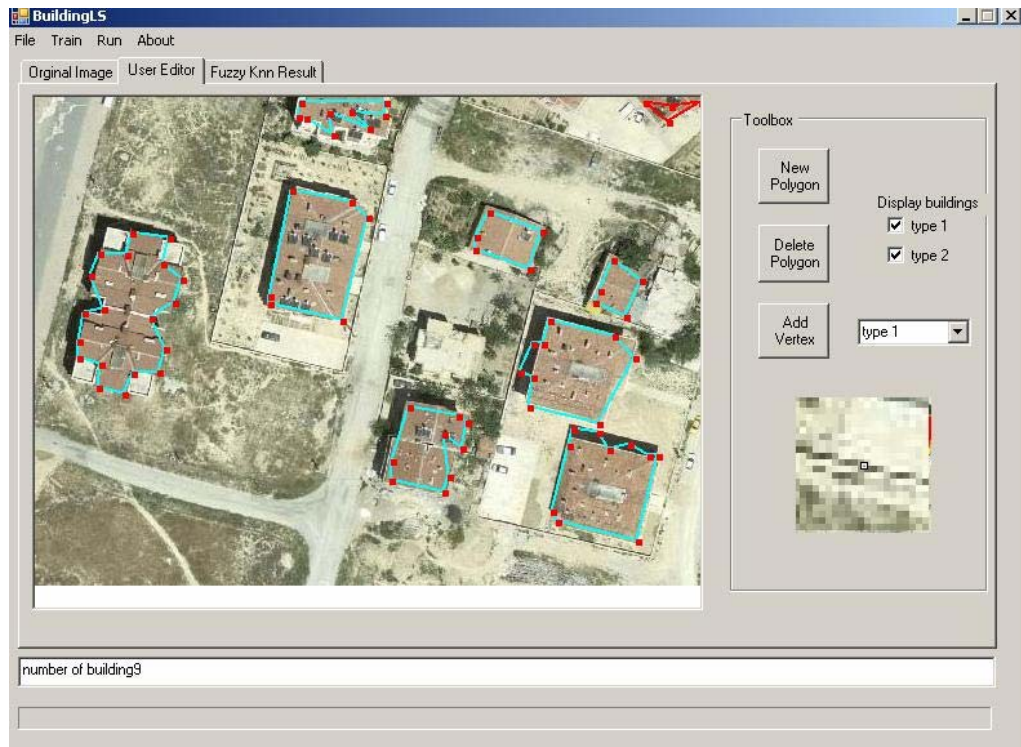


Figure 4.17. User Editor Tab

In the right side of the tab, there is a toolbox for editing. The user can select the type of buildings which will be displayed in the left side by this toolbox. Also the user can change type of the selected building, or add a new vertex on it. Moreover the user can draw a new polygon or delete a previously created polygon by using this toolbox.

3- Fuzzy Knn: The last tab is the most innovative property of the software. In this part the button labeled Update System, is used for starting feedback mechanism. The upper side of the screen is divided into two images. The left one shows the extracted buildings according to the user's current modifications. The right one is the previous state of extraction for comparison. If the user does not satisfy with the latest update, he can return the previous state by clicking Undo. Besides to analyze, the user can see pixels' class

membership values for the first type of buildings by clicking show membership values.



Figure 4.18. Fuzzy Knn results tab

## **CHAPTER 5**

### **EXPERIMENTS AND RESULTS**

In this chapter, the experiments which are performed to evaluate the proposed model and their results are discussed. For the best indication of performance, several evaluation criteria are used. Also, building extraction procedure is done for 5 different test areas. Thus, the performance for different type of regions and building characteristics is analyzed.

#### **5.1 Definition of Evaluation Criteria**

Since there is a wide range of errors in identifying the buildings in an aerial image, the evaluation criteria has a critical role for the performance of the system. The data table, which presents the experiment results, has to include required information like the performance of the model, errors and these errors types. For the evaluation of experiments, 5 different criteria are used and the detected buildings according to these criteria are counted.

- Fully Detected: the buildings, whose borders are correctly defined and do not need any modification by user.
- Missed: buildings which can not be detected. The Main reason for missing a building is the color similarity of the roof with any other object.
- Merged: The group of detached buildings , which are detected as a single building.

- Imperfect Detection: the buildings, which are detected by the model but needs small user edits to a better presentation. Usually the different type of objects on the roof like dish antennas causes these imperfections.
- False Alarm: the non-building areas, which are detected as a building by the model. The reason of this type of problem is also the color similarity. In some cases, the color of the some non-building objects can be very similar to the other objects. In such cases the model can extracted it as a building.

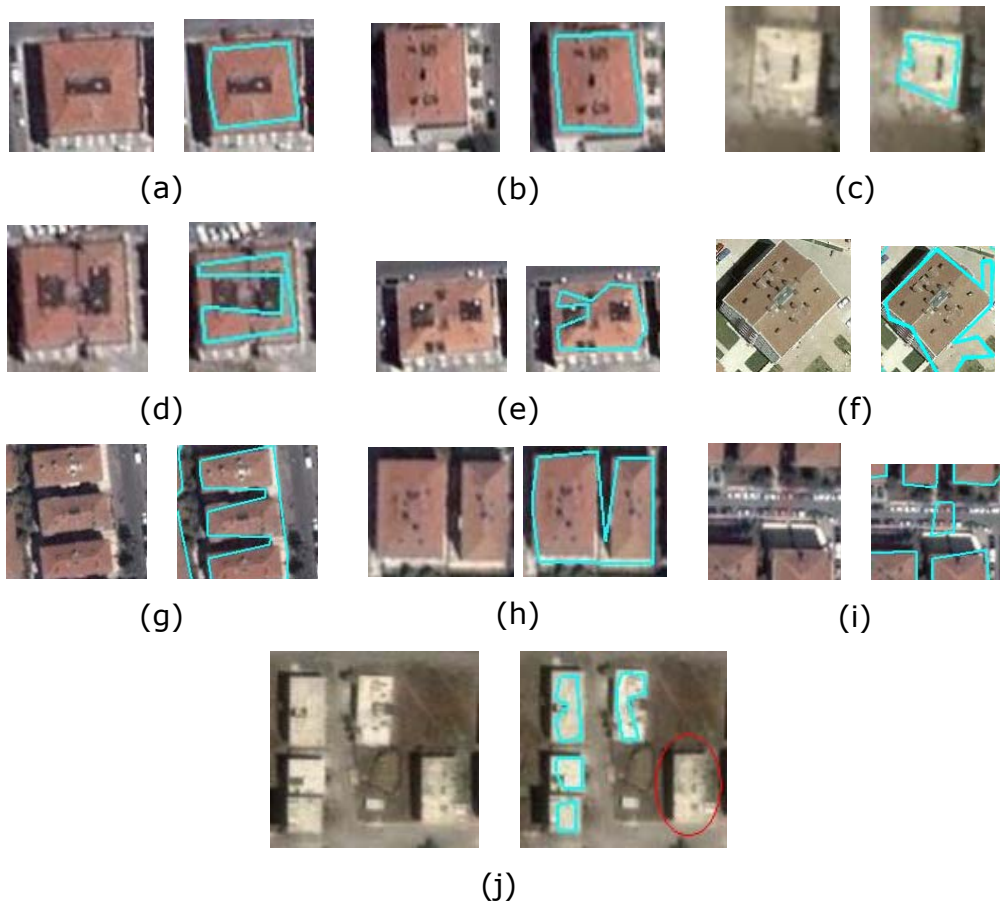


Figure 5.1. The examples of evaluation criteria. For each figure the first image shows the original image the second one is the output image. (a, b, c) Example of fully detected buildings. (d, e, f) Example of imperfect detection. (f, g) Example of merge problem. (i) Example of false alarm. (j) Example of missed building.



## 5.2 Define Test Areas

ABE-RF is tested in different test areas for a more realistic evaluation. While selecting these test areas, the characteristic of the environment and the buildings are considered. For each test area 6 images are captured. One of these six images is used for training. The others are used for performance testing.

Test Area 1:

- Appertain to Konya
- Suburban area
- Steppe climate is dominant
- Attached and detached buildings
- Different roof types



Figure 5.2. Sample images from Test Area 1. In this test area, many different types of roofs exist. Also some of the buildings are too close to each other such that they can not easily distinguished by the operator.

Test Area 2:

- Appertain to Emek – Ankara
- Urban area
- Usually Well planed, detached building
- Tile roofs





Figure 5.3 Sample images from Test Area 2 which is an urban area. The buildings are regularly placed and the roofs of the buildings are more significant.

#### Test Area 3:

- Appertain to Bostanlı –İzmir
- Urban area
- Both well planed and closely spaced buildings
- Tile roofs



Figure 5.4 Sample images from Test Area 3 which is also another urban area. City planning is better than the first test area and also regular type of roofs are used.

Test Area 4:

- Appertain to Levent- İstanbul
- Residential buildings
- Verdure
- Tile roofs



Figure 5.5 Sample images from Test Area 4.

Test Area 5:

- Appertain to Esenyurt- İstanbul
- Suburban area
- Steppe climate is dominant
- Different roof types



Figure 5.6. Sample images from Test Area 5 which is a hard test area. The problems in test area 1 are also exist in this test area. Also some of the buildings do not have any roof material and this makes the extraction problem more serious.

### 5.3 Experiments

As mentioned before for each test area 6 different images are captured. The experiments are performed in two cases. In the first case, one of these six images is manually labeled and the model is trained using this image, called training image. After the system is trained, software extracts buildings in the rest of the images.

The aim of the second case is performing feedback and checking the final improvement in the extraction. Software extracts the buildings in one of these five test image using the training data. Then, the user corrects some of the mistakes by the user interface and feedback the software for re-training. A sample illustration is given in figure 5.6.

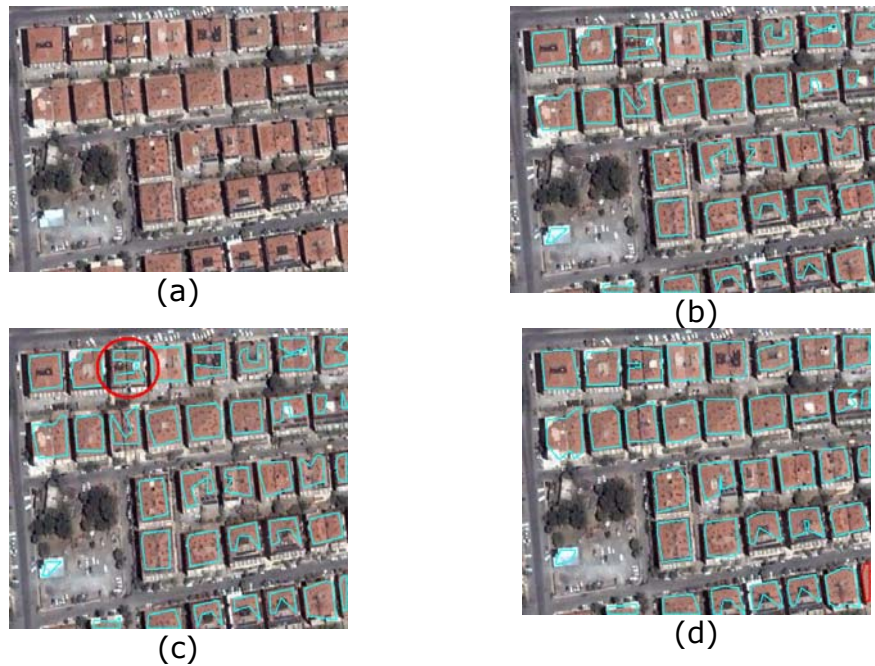


Figure 5.7. Illustration of Feedback. (a) The original image, (b) the initial extraction of the building. (c) The building which is manually edited. (c) the extraction of buildings using the feedback information

The user checks for the improvement and continues to editing until the performance is reached to the desired level or the feedback mechanism starts to reduce the extraction performance due to perplexing the training data. Then the software extracts the buildings in other images using this feedback information and re-train the system.





(a)



Total	20
Detection	13
Missed	0
Merge	0
Imperfect	7
False Alarm	3

(b)



Total	20
Detection	20
Missed	0
Merge	0
Imperfect	0
False Alarm	0

(c)

Figure 5.8. Sample test result for test area 1: The figure (a) is the original image. Figure (b) is the output of the model using the training data. Then the user manually fixes the problems in another image and feedback the model. The figure (c) is the new output of the model which uses the feedback information. The feedback information solves the imperfect and false alarm problems in this image.



(a)



Total	32
Detection	19
Missed	1
Merge	7
Imperfect	5
False Alarm	3

(b)



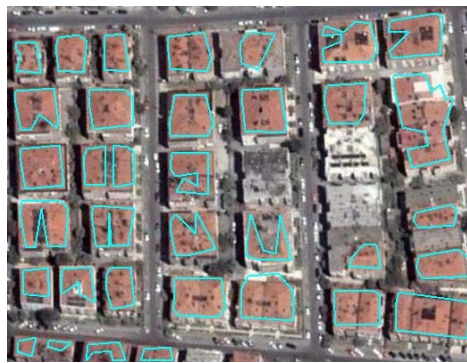
Total	32
Detection	29
Missed	1
Merge	0
Imperfect	2
False Alarm	0

(c)

Figure 5.9. Sample test result for test area 2. The figure (a) is the original image. Figure (b) is the output of the model using the training data. Then the user manually fixes the problems in another image and feedback the model. The figure (c) is the new output of the model which uses the feedback information. The feedback information reduces the imperfection and merging problems. However the feedback mechanism does not sufficient to detect missed buildings.



(a)



Total	30
Detection	17
Missed	3
Merge	0
Imperfect	10
False Alarm	0

(b)



Total	30
Detection	21
Missed	3
Merge	0
Imperfect	6
False Alarm	0

(c)

Figure 5.10. Sample test result for test area 3. The figure (a) is the original image. Figure (b) is the output of the model using the training data. Then the user manually fixes the problems in another image and feedback the model. The figure (c) is the new output of the model which uses the feedback information. The feedback information reduces the number of imperfectly detected buildings.





(a)



Total	22
Detection	16
Missed	0
Merge	0
Imperfect	6
False Alarm	2

(b)



Total	22
Detection	21
Missed	0
Merge	0
Imperfect	1
False Alarm	0

(c)

Figure 5.11. Sample test result for test area 4: (a) Original image (b) Case 1 - without feedback (c) Case 2 - after feedback



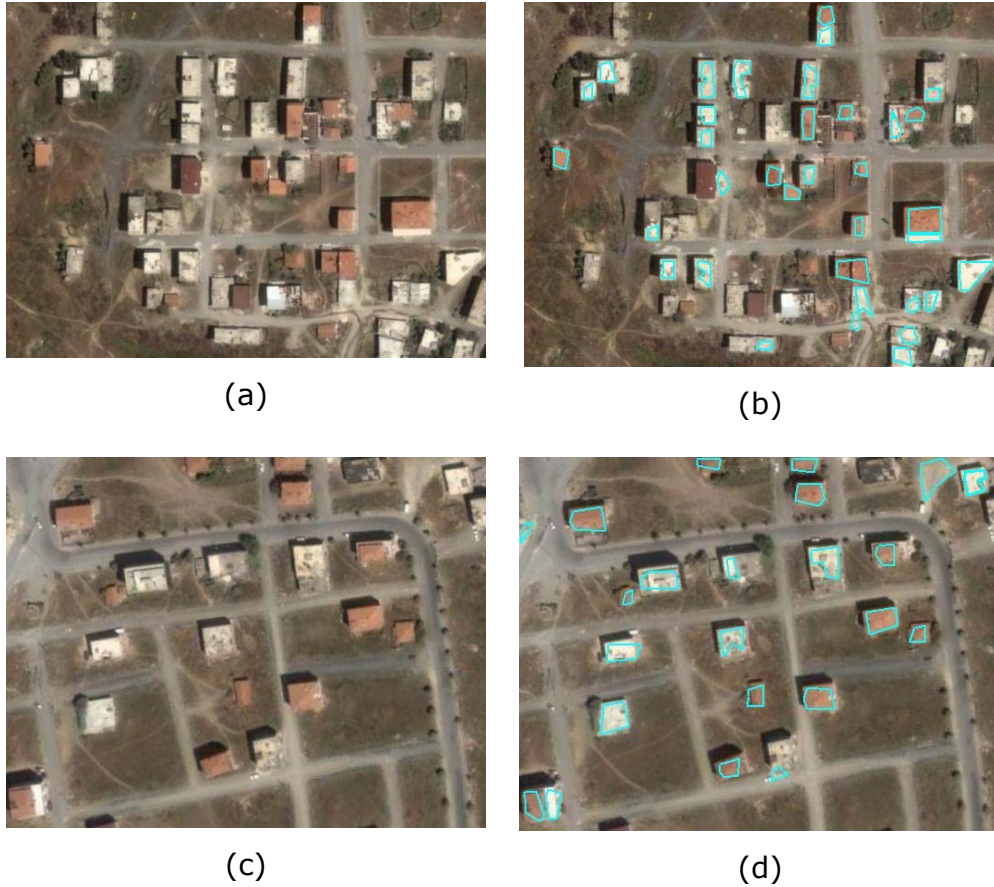


Figure 5.12. Two sample results for the test area 5. The figure (a) and (c) are the original images. Figure (b) and figure (d) shows the extracted buildings. We usually face with the missed problem in this test area and the feedback mechanism does not provide a significant improvement for this type of error.

## 5.4 Results and Discussions

In the following tables, the experiment results are given for the following two cases;

Case 1: The system detects the building using the training data

Case 2: The system detects the building using the training data and also the feedback information.

Table 5.1. Experiment results for Test Area 1.

Image Id	Total	Fully Detection		Imperfect Detection		Merge		Missed		False Alarm	
		Case1	Case2	Case1	Case2	Case1	Cas 2	Case1	Case2	Case1	Case2
2	12	6	8	4	2	2	2	0	0	5	1
4	18	8	17	9	0	0	0	1	1	2	1
5	8	4	3	2	4	0	0	2	1	4	4
6	37	13	16	6	3	11	12	7	6	4	1
1	20	14	20	6	0	0	0	0	0	3	0
<b>Total</b>	<b>95</b>	<b>45</b>	<b>64</b>	<b>27</b>	<b>9</b>	<b>13</b>	<b>14</b>	<b>10</b>	<b>8</b>	<b>18</b>	<b>7</b>

Table 5.2. Experiment results for Test Area 2.

Image Id	Total	Fully Detection		Imperfect Detection		Merge		Missed		False Alarm	
		Case1	Case2	Case1	Case2	Case1	Cas 2	Case1	Case2	Case1	Case2
1	33	19	28	3	5	11	0	0	0	0	1
3	37	30	34	3	3	4	0	0	0	1	1
4	32	19	29	5	2	7	0	1	1	3	0
5	46	30	36	3	4	13	6	0	0	1	0
6	16	0	16	0	0	16	0	0	0	0	0
<b>Total</b>	<b>164</b>	<b>98</b>	<b>143</b>	<b>14</b>	<b>14</b>	<b>51</b>	<b>6</b>	<b>1</b>	<b>1</b>	<b>5</b>	<b>2</b>

Table 5.3. Experiment results for Test Area 3

Image Id	Total	Fully Detection		Imperfect Detection		Merge		Missed		False Alarm	
		Case1	Case2	Case1	Case2	Case1	Cas 2	Case1	Case2	Case1	Case2
6	30	14	22	16	8	0	0	0	0	1	1
2	20	15	18	5	2	0	0	0	0	0	0
3	34	26	26	8	5	0	3	0	0	0	0
4	31	21	26	8	2	2	2	0	1	0	0
5	30	17	21	10	6	0	0	3	3	0	0
<b>Total</b>	<b>145</b>	<b>93</b>	<b>113</b>	<b>47</b>	<b>23</b>	<b>2</b>	<b>5</b>	<b>3</b>	<b>4</b>	<b>1</b>	<b>1</b>

Table 5.4. Experiment results for Test Area 4.

Image Id	Total	Fully Detection		Imperfect Detection		Merge		Missed		False Alarm	
		Case1	Case2	Case1	Case2	Case1	Cas 2	Case1	Case2	Case1	Case2
2	18	18	16	0	2	0	0	0	0	3	1
3	38	33	31	2	4	3	3	0	0	0	0
5	22	16	21	6	1	0	0	0	0	2	1
6	13	13	12	0	1	0	0	0	0	2	0
1	23	9	23	0	0	14	0	0	0	0	0
<b>Total</b>	<b>114</b>	<b>89</b>	<b>103</b>	<b>8</b>	<b>8</b>	<b>17</b>	<b>3</b>	<b>0</b>	<b>0</b>	<b>7</b>	<b>2</b>

Table 5.5. Experiment results for Test Area 5.

Image Id	Total	Fully Detection		Imperfect Detection		Merged		Missed		False Alarm	
		Case1	Case2	Case1	Case2	Case1	Case2	Case1	Case2	Case1	Case2
1	43	31	-	0	-	12	-	12	-	0	-
2	29	19	-	6	-	0	-	4	-	9	-
4	30	19	-	6	-	0	-	5	-	1	-
5	13	7	-	6	-	0	-	0	-	2	-
6	31	16	-	10	-	2	-	3	-	1	-
<b>Total</b>	<b>146</b>	<b>92</b>	<b>-</b>	<b>28</b>	<b>-</b>	<b>2</b>	<b>-</b>	<b>24</b>	<b>-</b>	<b>13</b>	<b>-</b>

The performance results of the test areas are given in the tables from Table 5.1 to Table 5.5. In the first test area, 10.53% of the buildings are missed due to variety in the type of the material of the roofs for this region. There are 3 major types of roofs for this region, which are tile, concrete and metallic. Also, some of the roofs contain more than one of these materials. The tile and the metallic roofs are extracted easier than the concrete roofs, whose color feature information is very similar to road class. The percentage of fully detected buildings, which do not need any correction, is 47.37%, however using the feedback information; the model increases this rate up to 67.37%. These rates are lower than most of the other test areas. One of the reason is the model does not

show a good performance on the roofs with mixed materials. One of the other reason is the disturbed shape of the roofs due to dish antennas and large water reservoirs on the roofs.

Table 5.6. Experiment results for whole test areas

	Without Feedback		Feedback	
	#	%	#	%
<b>Total</b>	664	100 %	664	100 %
<b>Fully Detected</b>	417	62,80%	515	77,56%
<b>Imperfect Detected</b>	124	18,67%	82	12,35%
<b>Merged</b>	85	12,80%	30	4,52%
<b>Missed</b>	38	5,72%	37	5,57%
<b>False Alarm</b>	44		25	

On the other hand, the test results for urban areas, which are Emek, Bostanlı and Levent, are better than the first one. Only 5 of the 423 building, which means 1.2% of the buildings, are missed in these test areas. Also the performance for fully detection is better: The model fully detects up to 67% of the buildings without the feedback. Model usually faced with merging problem in these dense populated areas. This fully detection rate is increased to 85% after the feedback information.

The last area also has very high error rates. The model misses 16% of the buildings. Additionally, the performances of the model could not be improved by the feedback. The sources of errors are very similar to the errors in the first test area. Some of the roofs are not covered by any material and the colors of these roofs, which cause errors, are very similar to the color of road or some other background objects. Recall that the implemented model uses only the color features, and the performance of the proposed system becomes insufficient in such areas.

The experimental results give us some important information about the model. The model is based on the classification of the pixels by the color information using a training space, and updates itself with the previous experiences. This basic assumption is quite valid in well planned urban areas. However in the undeveloped areas, usually a specific tile or metallic type of roof covering material is not used, they are usually concrete and using only the color information becomes insufficient for these roofs. A second important finding is, the relevance feedback creates a significant advantage on extraction of the buildings in case of the feature is enough to differentiate the classes. In the 664 test building, the feedback approach increases the fully detected buildings from 417 to 515. Also the merged buildings are decreased from 85 to 30. Moreover the model shows 50% improvement for false alarm with feedback. However the feedback approach does not work to recover the missed buildings. The main reason for this is, as discussed above, the insufficiency of the feature descriptor. The buildings, which are missed, have very similar feature vectors for other objects and if the user persists to try to correct this missing and feed backs the model, some of the roads or background objects are classified as a building.

## **CHAPTER 6**

### **CONCLUSION**

In this thesis, an automated building extraction method with learning capabilities is proposed. For a given aerial image, model classifies each pixel using fuzzy k-NN according to its HSB values and then extracts the objects using these classified pixels and some basic morphological operations. Then for each created object, model extracts the border and represents it using polygonal approximation method. The polygons used for representing buildings, are shown in the user interface, which is designed for interacting with the user. The user interface allows adding, editing and removing the objects. After the user completes editing, the model analyzes the changes made by user, uses them as a feedback and updates itself to improve its learning capabilities.

To achieve a reliable performance grade, results of the aerial images of different areas are examined. The performances of the model using HSB as a feature vector and the feedback mechanism are analyzed by the experiments. It is showed that the model is very satisfied in the urban areas. The main reason is that several type of roof covering materials are used in these areas. The model learns the color characteristics of these covering materials and extracts objects which carry the characteristics. On the other hand, in the rural or sub-urban areas, there is no standardization for roof covering; moreover some of the buildings do not have any roof covering materials at all. In such areas the model based on the color information shows a lower performance.

The relevance feedback is the major contribution of this thesis. It provides a considerable improvement for building extraction. However in the model, the performance of the feedback mechanism is related to the correctness of the information given by the user. Therefore, the user has to be trained before using the developed software.

As future work, it is planned to use more than one feature to extract the buildings. Although color information is a very important clue for detecting buildings, the texture information has to be included into the model for stable and satisfying results. Furthermore a better relevance feedback mechanism which will ignore the mistakes of the user is intended.

## REFERENCES

- [1] Easa, Said, & Chan, Yupo .Urban Planning and Development Applications of GIS. Reston, Virginia: American Society of Civil Engineers, pages 1-10, 2000
- [2] S.D Mayunga, Dr. Y. Zhang, and Dr. D.J. Coleman. Semi-Automatic Building Extraction Utilizing Quickbird Imagery. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences VOLUME XXXVI, PART 3/W24*, pages 131-136, 2005
- [3] Gülch E, Digital Systems for Automated Cartographic Feature Extraction. In *Proceedings of the XIX<sup>th</sup> ISPRS Congress*, volume XXXIII-B2 of *International Archives of Photogrammetry and Remote Sensing*, pages 241-256, Amsterdam.
- [4] F. Rottensteiner. Semi-automatic Extraction of Buildings Based on Hybrid Adjustment Using 3D Surface Models and Management of Building Data in a TIS. *PhD thesis, Vienna University of Technology*, 2001.
- [5] N. Haala. Extraction of buildings and trees in urban environments, *ISPRS Journal of Photogrammetry & Remote Sensing* 54, pages 130–137, 1999
- [6] Henricsson O. The Role of Color Attributes and Similarity Grouping in 3-D Building Reconstruction, *Computer Vision and Image Understanding*, 1998
- [7] A. Fischer, T.H.Kolbe, F. Lang, A. B. Cremers, W. Föorstner, L. Plümer, and V. Steinhage, Extracting buildings from aerial images using hierarchical aggregation in 2D and 3D, *Computer Vision and Image Understanding*, pages 185–203, 1998.



- [8] T. Moons, D. Frère, J. Vandekerckhove, and L. Van Gool, Automatic modeling and 3D reconstruction of urban house roofs from high resolution aerial imagery. In *Proceedings, Fifth European Conference on Computer Vision, Vol. I*, pages. 410–425, 1998.
- [9] D. Koc San and M. Turker. Automatic Building Extraction From High Resolution Stereo Satellite Images, *ISPRS commission VII*, 2007
- [10] G. Sohn and J. Dowman, "Building Extraction Using LIDAR DEMs and IKONOS Images", *ISPRS, Volume XXXIV, PART 3/W13*, 2003.
- [11] F. Rottensteiner and C. Briese. A New Method for Building Extraction in Urban Areas from High-Resolution LIDAR Data. *Symposium der ISPRS-Comm. III. International Archives of Photogrammetry and Remote Sensing, Volume XXXIV / 3A*, pp. 295 – 301, 2002
- [12] Tuytelaars, T. Van Gool, L. Proesmans, M. Moons, T. The cascaded Hough transform as an aid in aerial image interpretation, *Computer Vision. Sixth International Conference*, pages 67-72, 1998.
- [13] R. Jung and R. Schramm, Rectangle Detection based on a Windowed Hough Transform, In *Proc. 17th Brazilian Symposium on Computer Graphics and Image Processing*, Curitiba, Brazil, 2004.
- [14] S.D Mayungaa , Dr. Y. Zhanga, and Dr. D.J. Coleman, Semi-Automatic Building Extraction Utilizing Quickbird Imagery, *IAPRS Vol. XXXVI, Part 3/W24*, 2005
- [15] A. Stassopoulou, T. Caelli, Building Detection Using Bayesian Networks, *International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14, No. 6*, 2000
- [16] Yanfeng Wei, Zhongming Zhao, and Jianghong Song. Urban building extraction from high-resolution satellite panchromatic image using clustering and edge detection , *Geoscience and Remote Sensing Symposium*, 2005
- [17] Keping Chen Blong, R. , Extracting building features from high resolution aerial imagery for natural hazards risk assessment, *Geoscience and Remote Sensing Symposium*, 2002
- [18] M. Tuceryan, K. Jain, Texture analysis , *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, by C.

H. Chen, L. F. Pau, P. S. P. Wang (eds.), pp. 207-248, World Scientific Publishing Co., 1998.

- [19] E. Şaykol, G. Güleşir, KiMPA: A Kinematics-Based Method for Polygon Approximation, *Advances in Information Systems: Second International Conference, ADVIS 2002*, Izmir, Turkey, October 23-25, 2002.
- [20] F. Y. Vural, Image Processing Lecture Notes, 2004
- [21] R.C Gonzalez, E. Woods, Digital Image Processing, 1993 pp 518-530
- [22] Morphology, Dilation , last access: August 2007 ,<http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>
- [23] Richard O. Duda, Peter E. Hart, David G. Stork - Technology - 2004, Pattern Recognition
- [24] Wikipedia - Unsupervised learning , last access: August 2007 [http://en.wikipedia.org/wiki/Unsupervised\\_learning](http://en.wikipedia.org/wiki/Unsupervised_learning)
- [25] Matthew Kerwin, A Fuzzy K-Nearest Neighbor Algorithm review and Critical Analysis, 2005
- [26] Wikipedia - Supervised learning , last access: August 2007 [http://en.wikipedia.org/wiki/Supervised\\_learning](http://en.wikipedia.org/wiki/Supervised_learning)
- [27] J. M. Keller, M. R. Gray, and J. A. Givens, Jr., "A Fuzzy K-Nearest Neighbor Algorithm", *IEEE Transactions on Systems*, 1985
- [28] Wikipedia - HSV color space, [http://en.wikipedia.org/wiki/HSV\\_color\\_space](http://en.wikipedia.org/wiki/HSV_color_space)
- [29] E. Montiel, S. Aguado, M. Nixon, Texture classification via conditional histograms
- [30] B.S. Manjutah, J. Ohm, V. Vasudevan and A. Yamada, Color and Texture Descriptors, *IEEE Transactions On Circuits And Systems For Video Technology*, vol. 11, no. 6, 2001
- [31] Implementation of Edge Histogram Descriptor by Çağlar Şenaras in Matlab, <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=14128&objectType=File>
- [32] James Z. Wang, Gio Wiederhold, Oscar Firschein and Sha Xin Wei, Content-Based Image Indexing and Searching Using Daubechies' Wavelets, *International Journal on Digital Libraries*, vol. 1, no. 4, pages. 311-328 , 1998.

- [33] Polygonal Chain Algorithm, last access: August 2007  
<http://www.cs.mcgill.ca/~yfaras/cs644/project.html>
- [34] Nafiz Arica, Fatos T. Yarman-Vural, BAS : A Perceptual Shape Descriptor Based on the Beam Angle Statistics, *Pattern Recognition Letters*, Vol. 24, pages. 1627-1639, 2003.
- [35] Turner, M.R., Texture Discrimination by Gabor Functions, *Biological Cybernetics*, 55, pp. 71-82, 1986.
- [36] R.M. Haralick. Statistical Image Texture Analysis. *Handbook of Pattern Recognition and Image Processing*, volume 86, pages 247--279. Academic Press, 1986.