

**AUTOMATIC RECONSTRUCTION OF PHOTOREALISTIC 3-D BUILDING
MODELS FROM SATELLITE AND GROUND-LEVEL IMAGES**

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

BY

EMRE SÜMER

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
GEODETIC AND GEOGRAPHIC INFORMATION TECHNOLOGIES**

MARCH 2011

Approval of the thesis:

**AUTOMATIC RECONSTRUCTION OF PHOTOREALISTIC 3-D BUILDING
MODELS FROM SATELLITE AND GROUND-LEVEL IMAGES**

submitted by **EMRE SÜMER** in partial fulfillment of the requirements for the degree
of **Doctor of Philosophy in Geodetic and Geographic Information Technologies**
Department, Middle East Technical University by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Vedat Toprak
Head of Department, **Geological Engineering** _____

Prof. Dr. Volkan Atalay
Supervisor, **Computer Engineering Dept., METU** _____

Assoc. Prof. Dr. Mustafa Türker
Co-Supervisor, **Geodesy and Photogrammetry Eng., Hacettepe Univ.** _____

Examining Committee Members:

Prof. Dr. Vedat Toprak
Geological Engineering Dept., METU _____

Prof. Dr. Volkan Atalay
Computer Engineering Dept., METU _____

Assoc. Prof. Dr. M.Onur Karslıođlu
Civil Engineering Dept., METU _____

Assoc. Prof. Dr. Hasan Ođul
Computer Engineering Dept., Baskent Univ. _____

Assist. Prof. Dr. Yakup Özkazanç
Electrical and Electronics Engineering Dept., Hacettepe Univ. _____

Date: 15. 03. 2011

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: Emre SÜMER

Signature:

ABSTRACT

AUTOMATIC RECONSTRUCTION OF PHOTOREALISTIC 3-D BUILDING MODELS FROM SATELLITE AND GROUND-LEVEL IMAGES

Sümer, Emre

Ph.D., Department of Geodetic and Geographic Information Technologies

Supervisor: Prof. Dr. Volkan Atalay

Co-Supervisor: Assoc. Prof. Dr. Mustafa Türker

March 2011, 229 pages

This study presents an integrated framework for the automatic generation of the photorealistic 3-d building models from satellite and ground-level imagery. First, the 2-d building patches and the corresponding footprints are extracted from a high resolution imagery using an adaptive fuzzy-genetic algorithm approach. Next, the photorealistic facade textures are automatically extracted from the single ground-level building images using a developed approach, which includes facade image extraction, rectification, and occlusion removal. Finally, the textured 3-d building models are generated automatically by mapping the corresponding textures onto the facades of the models.

The developed 2-d building extraction and delineation approach was implemented on a selected urban area of the Batikent district of Ankara, Turkey. The building regions were extracted with an approximate detection rate of 93%. Moreover, the overall delineation accuracy was computed to be 3.9 meters. The developed concept for facade image extraction was tested on two distinct datasets. The facade image extraction accuracies were computed to be 82% and 81% for the Batikent and eTrims datasets,

respectively. As to rectification results, 60% and 80% of the facade images provided errors under ten pixels for the Batikent and eTrims datasets, respectively. In the evaluation of occlusion removal, the average scores were computed to be 2.58 and 2.28 for the Batikent and eTrims datasets, respectively. The scores are ranked between 1 (Excellent) to 6 (Unusable). The modeling of the total 110 single buildings with the photorealistic textures took about 50 minutes of processor running time and yielded a satisfactory level of accuracy.

Keywords: Building Detection, Facade Texture, Geometric Rectification, Occlusion Removal, 3-D Modeling

ÖZ

FOTOGERÇEKÇİ 3-B BİNA MODELLERİNİN UYDU VE YER SEVİYESİ GÖRÜNTÜLERİNDEN OTOMATİK OLARAK GERİÇATILIMI

Sümer, Emre

Doktora, Jeodezi ve Coğrafi Bilgi Teknolojileri E.A.B.D

Tez Yöneticisi: Prof. Dr. Volkan Atalay

Ortak Tez Yöneticisi: Doç. Dr. Mustafa Türker

Mart 2011, 229 sayfa

Bu çalışmada, uydu ve yer seviyesinden çekilmiş görüntülerden fotogerçekçi üç boyutlu bina modellemesini otomatik olarak yapmayı sağlayan bütünleşik bir yapı önerilmektedir. İlk önce, yüksek çözünürlüklü uydu görüntülerinden uyarlamalı bulanık-genetik algoritma yaklaşımı kullanılarak iki boyutlu bina bölge ve ayak izi çıkarımı gerçekleştirilir. Daha sonra, tekli yersel bina görüntüsünden; yüz görüntüsü çıkarımı, rektifikasyon ve engel giderme adımlarından oluşan bir yaklaşım kullanılarak fotogerçekçi yüz dokuları otomatik olarak çıkarılır. Son olarak, ilgili yüz dokularının model cephelerine kaplanması ile doku kaplı üç boyutlu modeller otomatik olarak üretilir.

İki boyutlu bina bölge ve ayak izi çıkarımı yaklaşımı Türkiye’de Ankara’nın Batıkent bölgesinden seçilmiş kentsel bir alan üzerinde uygulanmıştır. Bina bölgeleri yaklaşık %93’lük bir tespit oranı ile çıkarılmıştır. Bina sınırlarının çıkarılma doğruluğu ise 3.9 metre olarak hesaplanmıştır. Yüz dokularının elde edilmesine yönelik geliştirilen yaklaşım iki farklı veri kümesi üzerinde test edilmiştir. Yüz dokusu çıkarımı doğrulukları Batıkent ve eTrims veri kümeleri için sırasıyla %82 ve %81 olarak hesaplanmıştır. Rektifikasyon sonuçlarına göre, yüz dokularının Batıkent veri kümesi için %60’ının ve

eTrims veri kümesi için %80'inin 10 pikselin altında hataya sahip olduğu tespit edilmiştir. Engel giderme yaklaşımının değerlendirilmesinde ortalama skorlar Batikent veri kümesi için 2.58, eTrims veri kümesi için 2.28 olarak hesaplanmıştır. Bu skorlar 1 (Kusursuz) ile 6 (Kullanılamaz) değerleri arasında derecelendirilmektedir. Toplamda 110 binanın fotogerçekçi olarak modellenmesi 50 dakikalık bir işlemci yürütme süresi içerisinde ve tatmin edici bir başarı düzeyi ile gerçekleştirilmiştir.

Anahtar Kelimeler: Bina Tespiti, Yüz Dokusu, Geometrik Rektifikasyon, Engel Giderme, 3-B Modelleme

In Memory of My Dearest Father...

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my co-supervisor, Assoc. Prof. Dr. Mustafa TÜRKER, for his guidance, advice, criticism, encouragement and insight throughout my research. His intellect, attitude to research, and clarity of knowledge was a great source of inspiration for me. Special thanks are also due for all the fruitful brainstorming sessions and critical remarks he provided.

I would like to express my thanks to my supervisor Prof. Dr. Volkan ATALAY for giving me the opportunity to be under his supervision throughout my PhD. Special thanks are also due for his support and criticism during my thesis supervising committees.

I would also like to thank my supervising and examining committee members, Prof. Dr. Vedat TOPRAK, Assoc. Prof. Dr. Hasan OĞUL and Assist. Prof. Dr. Yakup ÖZKAZANÇ. Their insightful comments helped me to finish my PhD research and thesis.

I would like to express my sincere gratitude to my dean Prof. Dr. Berna DENGİZ, my former dean Prof. Dr. İmdat KARA, my chairman Prof. Dr. Ziya AKTAŞ, my colleagues Assist. Prof. Dr. Mustafa SERT and Assoc. Prof. Dr. Nizami GASILOV, and our department secretary Nihal UĞUR from Başkent University for their insight and patience throughout my study.

I would also thank to my friends Ali Özgün OK and Dilek KOÇ SAN for their suggestions, assistance and encouragement throughout my thesis.

I wish to express my sincere gratitude and appreciation to TÜBİTAK for its financial support during my thesis study.

I would also like to thank my parent for their affection, support and patience during my PhD program.

Finally, I cordially would like to thank my beloved wife. She is always encouraging and supporting me, and I could not finish this long journey without her.

TABLE OF CONTENTS

ABSTRACT.....	iv
ÖZ.....	vi
ACKNOWLEDGEMENTS.....	ix
TABLE OF CONTENTS.....	xi
LIST OF TABLES.....	xiv
LIST OF FIGURES.....	xvi
CHAPTER	
1. INTRODUCTION.....	1
1.1 Problem Definition and Motivations.....	1
1.2 Objectives and Contributions.....	3
1.3 Study Area and Data Description.....	5
1.4 Software and Hardware Requirements.....	12
1.5 Organization of the Thesis.....	16
2. BACKGROUND AND PAST STUDIES.....	18
2.1 Feature Extraction from Remote Sensing Images.....	18
2.2 2-D Building Acquisition and 3-D Modeling.....	21
2.2.1 2-D Building Acquisition and 3-D Modeling from Airborne Images.....	22
2.2.2 2-D Building Acquisition and 3-D Modeling from Spaceborne Images.....	32
2.3 Building Facade Texture Acquisition and Mapping	41
2.3.1 Airborne Studies Related to Building Facade Texture Acquisition and Mapping.....	42
2.3.2 Ground-based Studies Related to Building Facade Texture Acquisition and Mapping.....	43
2.3.2.1 Building Facade Texture Acquisition and Mapping from Laser Scanning Data.....	43

2.3.2.2	Building Facade Texture Acquisition and Mapping from Terrestrial Image Sequences.	45
2.3.2.3	Building Facade Texture Acquisition and Mapping from Single Image.....	49
2.3.3	Summary of Texture Occlusion Removal Techniques.....	51
3.	THE METHODOLOGY AND THE DEVELOPED SOFTWARE.....	54
3.1	The Methodology.....	54
3.2	The Developed Software.....	57
4.	2-D BUILDING EXTRACTION AND DELINEATION.....	67
4.1	The Proposed Workflow.....	67
4.2	Building Extraction.....	68
4.2.1	The Genetic-Image Component.....	69
4.2.2	Adaptive-Fuzzy Component.....	79
4.2.3	Genetic Algorithm Parameter Analysis.....	84
4.2.4	The Accuracy Assessment of Building Extraction.....	88
4.3	Building Delineation.....	89
4.3.1	The Preprocessing.....	90
4.3.2	Region Corner Detection.....	95
4.3.3	The Accuracy Assessment of Building Delineation....	98
5.	BUILDING FACADE TEXTURE ACQUISITION.....	101
5.1	The Proposed Workflow.....	101
5.2	Facade Image Extraction.....	102
5.2.1	Radiometric Pre-processing.....	104
5.2.2	Initialization of the Marker Pixels.....	105
5.2.3	Watershed Transform.....	106
5.2.4	The Accuracy Assessment of Facade Image Extraction.....	113
5.3	Facade Image Rectification.....	115
5.3.1	Facade Edge Analysis.....	117
5.3.2	End-Point Analysis.....	120
5.3.3	Projective Transformation.....	122

5.3.4	The Accuracy Assessment of Facade Image Rectification.....	127
5.4	Occlusion Removal.....	129
5.4.1	Facade Image Cropping.....	130
5.4.2	Occlusion Extraction and Texture Mending.....	133
5.4.3	Post-processing.....	135
5.4.4	The Accuracy Assessment of Occlusion Removal..	140
6.	3-D MODEL GENERATION AND TEXTURE MAPPING.....	142
6.1	Generating the Solid Building Models.....	142
6.2	Texture Selection and Mapping.....	149
7.	RESULTS AND DISCUSSION.....	163
7.1	Results and Discussion for 2-D Building Extraction and Delineation.....	163
7.2	Results and Discussion for Building Facade Texture Acquisition	178
7.3	Results and Discussion for 3-D Model Generation and Texture Mapping.....	197
8.	CONCLUSIONS AND RECOMMENDATIONS.....	202
8.1	Conclusions for 2-D Building Extraction and Delineation....	202
8.2	Conclusions for Building Facade Texture Acquisition.....	204
8.3	Conclusions for 3-D Model Generation and Texture Mapping.....	206
8.4	Recommendations.....	208
	REFERENCES.....	210
	APPENDICES	
A.	SOURCE CODE OF THE 2-D BUILDING EXTRACTION AND DELINEATION.....	223
B.	SOURCE CODE OF BUILDING FACADE TEXTURE ACQUISITION.	225
C.	SOURCE CODE OF 3-D MODEL GENERATION AND TEXTURE MAPPING.....	227
	CURRICULUM VITAE.....	229

LIST OF TABLES

TABLES

Table 4.1 The primitive image processing operations (the gene pool) (Harvey et al., 2002).....	72
Table 4.2 Fuzzy rules for the crossover operation.....	81
Table 4.3 Fuzzy rules for the mutation operation.....	81
Table 4.4 The selected step size and minimum/maximum values for determining the optimum population, chromosome, and training/test sample sizes.....	86
Table 4.5 The results of 360 individual runs, yielding the optimum population and chromosome sizes.....	85
Table 4.6 An example to clarify the peak values given in Table 4.5.....	87
Table 4.7 For a sample building the pixel errors on four corner points.....	100
Table 4.8 For a sample building, the accuracies under different confidence levels.....	100
Table 5.1 For a sample image, the pixel errors for the test points 1 to 4....	128
Table 5.2 For a sample image, the accuracies under different confidence levels.....	128
Table 5.3 Rating scale of the Television Allocations Study Organization (Frendendall and Behrend, 1960).....	141
Table 6.1 For a sample building, the corner coordinates and the building ID information.....	143
Table 6.2 The number of photos taken for the building blocks 1-6, and 8...	152
Table 6.3 The structure of the photo database.....	154
Table 7.1 For each run, the fitness values computed for 20 generations that do not include the adaptive-fuzzy component (the first variant)...	164
Table 7.2 For each run, the fitness values computed for 20 generations that include the adaptive-fuzzy component (the second variant).....	165

Table 7.3 The distance errors (pixel) computed for the corner points of the buildings.....	171
Table 7.4 The accuracies of the buildings under different confidence levels.....	173
Table 7.5 The quantitative results of the facade image extraction for the eTrims and Batikent datasets.....	179
Table 7.6 The pixel errors computed at the test points on the rectified facade images.....	182
Table 7.7 For the rectified facade images, the accuracies computed at 80%, 85%, 90%, and 95% confidence levels.....	183
Table 7.8 The ratings of ten viewers for the occlusion-free facade images.	185

LIST OF FIGURES

FIGURES

Figure 1.1 The Study Area.....	6
Figure 1.2 Pan-sharpened image of the sub-region used in the present study.....	7
Figure 1.3 (a) DTM and (b) DSM of the study area.....	8
Figure 1.4 The generation of nDSM: The gray areas represent the 3-d features after applying the threshold (Koc San, 2009).....	9
Figure 1.5 The workflow for generating the normalized digital surface model (nDSM) (Koc San, 2009).....	9
Figure 1.6 The generated nDSM of the study area.....	10
Figure 1.7 Scenes from the ground-level data collection process.....	11
Figure 1.8 The facade images captured in the Batikent district.....	11
Figure 1.9 The facade images taken from eTrims image database.....	12
Figure 1.10 The screenshot of the Cortona 3-d toolbar.....	13
Figure 1.11 Samsung WB 500 digital camera.....	15
Figure 1.12 Magellan Explorist XL handheld GPS receiver.....	15
Figure 2.1 The overall structure of GENIE (Perkins et al., 2000).....	39
Figure 2.2 The processing pipeline of Genie Pro (Perkins et al., 2005).....	40
Figure 2.3 The fuzzy adaptive genetic algorithm model (Herrera and Lozano 2003).....	41
Figure 3.1 The flowchart of the developed photorealistic 3-d building modeling from satellite and ground-level imagery.....	54
Figure 3.2 The interface of the developed software.....	58
Figure 3.3 (a) The display of a satellite image using the bands Red, Green, and Blue, and (b) the display of a facade in Red band only.....	59
Figure 3.4 A closer view of panel-3.....	60
Figure 3.5 The options of the “File” menu item.....	60
Figure 3.6 A snapshot from the option “Load Study Area”.....	61

Figure 3.7 The “Pre-processing” menu item and the options it contains.....	62
Figure 3.8 The option “Adjust Intensity” of the “Pre-processing” menu item..	62
Figure 3.9 The “2-D Delineation” menu item and the options it contains.....	63
Figure 3.10 The “Solid Modeling” menu item and the options it contains.....	64
Figure 3.11 The “Texture Generation” menu item and the options it contains.....	65
Figure 3.12 The “Photorealistic Modeling” menu item and the options it contains.....	66
Figure 4.1 The general workflow of the 2-D building extraction and delineation approach.....	68
Figure 4.2 The proposed 2-D building extraction technique.....	69
Figure 4.3 The structure of a population, which is composed of M chromosomes and N genes in each chromosome.....	70
Figure 4.4 The distribution of the training and test samples.....	71
Figure 4.5 The question dialog box which pops up before the training/test sample selection.....	72
Figure 4.6 The dimension reduction procedure using Fisher Linear Discriminant.....	75
Figure 4.7 The optimum direction ‘w’ in discriminating the points belonging to two different classes (red and black).....	76
Figure 4.8 An example for the crossover operation and the generated offsprings.....	78
Figure 4.9 An example for the mutation operation and the produced offspring.....	79
Figure 4.10 The parameter flow between the genetic image and the adaptive- fuzzy components (Herrera and Lozano, 2003).....	80
Figure 4.11 The membership functions for (a) e_1 , (b) e_2 and (c) $\Delta P_m(k)$	82
Figure 4.12 The screenshots of (a) the wait bar and (b) the “Layers” panel..	83
Figure 4.13 The parameters of the genetic algorithm with (a) the optimum values and (b) the arbitrary values.....	88
Figure 4.14 The workflow of the building delineation methodology.....	89
Figure 4.15 A disk-shaped structuring element with a radius of 3 (Matlab 2010a online documentation, 2010).....	91

Figure 4.16 (a) A selected building block from the study area, (b) the extracted building regions, and (c) the building regions after applying the opening and artifact removal operations.....	92
Figure 4.17 The building regions after applying the hole-filling operation.....	93
Figure 4.18 The final image after applying the pre-processing step of convex image generation.....	93
Figure 4.19 A sample configuration for the morphological pre-processing functions of the PREBUM software.....	94
Figure 4.20 The view of the “Layers” panel after the morphological pre-processing step.....	95
Figure 4.21 The generation of the best fitting ellipse (Matlab 2010a online documentation, 2010).....	96
Figure 4.22 The computed region corner points (1-4).....	96
Figure 4.23 The extracted building footprints overlaid with the image.....	98
Figure 4.24 For a selected sub area, (a) the extracted footprints and (b) the reference footprints. (c) The extracted corners and (d) the reference corners of a building.....	99
Figure 5.1 Building facade texture acquisition.....	102
Figure 5.2 The workflow of the proposed automatic facade image extraction technique.....	103
Figure 5.3 (a) A ground level photograph of a building having non-uniform illumination and (b) the region of interest in a closer view.....	104
Figure 5.4 (a) The building shown in Figure 5.3 after boosting the saturation and (b) the region of interest in a closer view.....	105
Figure 5.5 The initial markers for the foreground (red) and background (green) regions.....	106
Figure 5.6 Illustration of the concept of watershed segmentation: (a) The profile of gray level image; (b) Local minima of gray level yield catchment basins, local maxima define the watershed lines.....	106
Figure 5.7 The gradient image.....	107
Figure 5.8 The distributions of the marker pixels (left column) and the segments produced (right column) in the first (row-1), second (row-2), third (row-3) and fourth (row-4) iterations.....	108

Figure 5.9 The resulting segments after performing ten independent iterations.....	110
Figure 5.10 (a) The building facade patch extracted through intersecting the segments and (b) the extracted RGB facade image.....	111
Figure 5.11 The texture extraction parameters with (a) the predefined and (b) the arbitrary values.....	111
Figure 5.12 A screenshot of a segmentation, from which the current repetition count is 7 and the current iteration count is 1.....	112
Figure 5.13 A screenshot of the "Layers" panel after the facade texture extraction process.....	113
Figure 5.14 The four possible cases used in the accuracy assessment of facade image extraction procedure.....	114
Figure 5.15 The distortions with respect to horizontal and vertical reference lines.....	115
Figure 5.16 The workflow of the proposed automatic facade image rectification procedure.....	116
Figure 5.17 (a) The edges after applying the Canny edge detector, (b) the remaining edges after removing the horizontal edges and (c) the 3x3 vertical template, H.....	118
Figure 5.18 The Hough Transformation.....	119
Figure 5.19 The facade edges detected using the Hough transform with the line-cut threshold value of 150.....	119
Figure 5.20 The end-points of Hough lines located by the Harris corner detector.....	120
Figure 5.21 The estimated position of the erroneous end-point, which is shown in green.....	121
Figure 5.22 The orientations of (a) the initial trend lines and (b) the final trend lines.....	122
Figure 5.23 A projective transformation with center O, mapping xy plane to x'y' plane.....	123
Figure 5.24 For a 2D projective transformation example, the correspondences of four points.....	123

Figure 5.25 The alignment operation with respect to (a) vertical axis and (b) horizontal axis.....	125
Figure 5.26 The extracted building facades (a) before geometric rectification and (b) after the geometric rectification with the horizontal and vertical reference lines superimposed.....	126
Figure 5.27 A screenshot of "Layers" panel after the rectification process of the facade image.....	127
Figure 5.28 The test points on (a) the reference and (b) the rectified images.....	128
Figure 5.29 The steps of the proposed occlusion removal procedure.....	129
Figure 5.30 (a) The rectified facade image and (b) the set of corner points with the detected upper facade corners (A and B).....	130
Figure 5.31 (a) The delimiter lines for the sides (green) and the top (pink). (b) the facade image after cropping from the left, right, and the upper sides.....	131
Figure 5.32 After performing the cropping operation, (a) the unfilled image and (b) the filled image.....	132
Figure 5.33 A screenshot of the "Layers" panel after performing the facade texture cropping operation.....	132
Figure 5.34 (a) The selection of an occluded region that is enclosed by a dashed red rectangle. (b) The cropped patch of the occluded area.....	133
Figure 5.35 (a) The segmented image, in which the black pixels represent the occlusions and (b) the image patch with the occluded areas (black) superimposed.....	134
Figure 5.36 (a) The occluded patch, (b) the detected candidate patch, and (c) the mended facade image after fusing the patch detected in (b).....	135
Figure 5.37 (a) A sample building facade image patch with a seam pointed by the white arrows and (b) the reduced seam after applying the motion blurring in vertical direction.....	136
Figure 5.38 The abrupt illumination change (within red circle) on the building facade image.....	136

Figure 5.39 (a) The grayscale difference image, in which dark areas refer to minor or no change and (b) the updated candidate image.....	137
Figure 5.40 The image patch produced after applying the weighted averaging.....	137
Figure 5.41 The final building facade image after applying the post-processing steps.....	138
Figure 5.42 The building facade image before applying any post-processing.....	138
Figure 5.43 A wait bar that shows the correlation rate.....	139
Figure 5.44 (a) The question dialog box appeared after each occlusion removal cycle and (b) the screenshot of the “Layers” panel after performing the occlusion removal procedure.....	140
Figure 6.1 The boundaries of a quadrilateral shaped building.....	143
Figure 6.2 The translation of a building (red) on a base map.....	144
Figure 6.3 The buildings with different orientations.....	145
Figure 6.4 (a) The center point (blue cross) of a binary building patch and (b) the corresponding nDSM patch.....	145
Figure 6.5 (a) The height values falling within the 7x7 buffer zone. (b) The frequencies of the rounded height values and the selected height value of 14.7 m.....	146
Figure 6.6 The automatically generated 3-d models in a building block (Block-3).....	147
Figure 6.7 (a) The selection dialog box for the height files and (b) the subsequent warning message.....	148
Figure 6.8 The warning message after the generation of 3-D solid models.....	149
Figure 6.9 The workflow for the texture selection and mapping.....	150
Figure 6.10 The building blocks that contain (a) the rectangular shape buildings and (b) the square shaped buildings with the labels assigned to edges.....	151
Figure 6.11 The locations of photo shooting points and the directions for the building blocks considered.....	153
Figure 6.12 The 8-directions with the angles.....	154

Figure 6.13 The computation of Δy and Δx between A and B.....	155
Figure 6.14 (AB) Bearings in the (a) first, (b) second, (c) third, and (d) fourth quadrants.....	156
Figure 6.15 The facade photo assignment procedure for building block-2..	158
Figure 6.16 A screenshot of the textured buildings of building block-2.....	159
Figure 6.17 The illustrations (top) and the VRML models (bottom) of (a) gable and (b) hip roof types.....	160
Figure 6.18 (a) The roof pattern used at present study and (b) the resulting texture on a roof.....	160
Figure 6.19 The texture filling procedure for the gable-roofed buildings. (a) A building with the unfilled roof texture. (b) The same building with the filled roof texture.....	161
Figure 6.20 The generated photorealistic 3-D buildings of the study area from two different bird's-eye perspectives.....	162
Figure 6.21 The warning message after the generation of the photorealistic 3-d model.....	162
Figure 7.1 The performance curve of the first variant averaged over 10 runs.....	164
Figure 7.2 The performance curve of the second variant averaged over 10 runs.....	165
Figure 7.3 The building regions extracted using the proposed adaptive fuzzy-genetic classification approach with the approximate fitness value of 83.....	167
Figure 7.4 The building regions extracted using the proposed adaptive fuzzy-Genetic classification approach with the approximate fitness value of 93.....	167
Figure 7.5 (a) The detected building patches, on which the numbers represent building IDs after the morphological pre-processing, (b) the extracted footprints (green), and (c) the reference footprints (red).....	169

Figure 7.6 (a) The detected building patches, on which the numbers represent building IDs after the morphological pre-processing, (b) the extracted footprints (green), and (c) the reference footprints (red).....	170
Figure 7.7 (a) The image that contains closely located two buildings and (b) the result of segmentation which generated a merged single patch (colored in pink).....	177
Figure 7.8 The building boundaries (in green) detected from (a) a vertically elongated building and (b) a diagonally elongated building. The binary images represent the extracted building patches.....	177
Figure 7.9 The buildings with complex geometry	178
Figure 7.10 (a) A facade image from the eTrims dataset, (b) the reference patch generated manually, and (c) the facade patch extracted automatically through the developed approach. This facade image provided the highest QP value of 96.6%.....	180
Figure 7.11 (a) A facade image from the eTrims dataset, (b) the reference patch generated manually, and (c) the facade patch extracted automatically through the developed approach. This facade image provided the lowest QP value of 66.7%.....	180
Figure 7.12 (a) A facade image from the Batikent dataset, (b) the reference patch generated manually, and (c) the facade patch extracted automatically through the developed approach. This facade image provided the highest QP value of 97.8%.....	181
Figure 7.13 (a) A facade image from the Batikent dataset, (b) the reference patch generated manually, and (c) the facade patch extracted automatically through the developed approach. This facade image provided the lowest QP value of 53.5%.....	181
Figure 7.14 For the Trims dataset, the facade images with (a) the best rating and (b) the worst rating with an example of a failure outlined in red.....	187
Figure 7.15 For Batikent dataset, the facade images with (a) the best rating and (b) the worst rating with examples of failures outlined in red and cyan.....	187

Figure 7.16 The problem of under-segmentation. (a)-(b) The original images, and (c)-(d) the extracted patches suffering from under-segmentation.....	189
Figure 7.17 An example case showing the failure of the automated rectification approach. (a) The original image, (b) the result of facade extraction, (c) the detected edges by Canny operator, (d) the extracted Hough lines, and (e) the result of the rectification.....	191
Figure 7.18 A building facade image, in which the height difference between the building baseline and the pavement is evident.....	192
Figure 7.19 (a) Two buildings with identical textures that are located back to back and (b) the segmented facade with the over-segmentation.....	192
Figure 7.20 (a) The original image and (b) the generated occlusion-free image that contains superfluous information and incomplete structures.....	193
Figure 7.21 (a) A building facade located near the edge of the photograph and (b) the result of segmentation performed using the developed approach.....	194
Figure 7.22 Examples for showing the limitations of the developed occlusion removal technique.....	196
Figure 7.23 The adjacent facades of a building with different illuminations.....	198
Figure 7.24 A building suffering from a texture mismatch problem.....	199
Figure 7.25 For building block-1, the (a) nDSM data and (b) the R-G-B image.....	200
Figure 7.26 Mapping a circular facade texture on a quadrilateral plane (a and b), which generates an unrealistic view of the model.....	201

CHAPTER 1

INTRODUCTION

This study deals with the photorealistic 3-d modeling of urban buildings from space and ground level images. In the first section, the definition of the problem and the motivations are given. This is followed by the objectives and contributions of the study. Next, the study area is described together with the data used to test the developed integrated approach. After that, the software and hardware requirements are presented. In the final section, the thesis chapters are summarized as an outline.

1.1. Problem Definition and Motivations

Since the majority of the population lives in urban areas, many critical management issues involving geographical analysis such as urban planning, monitoring urban change and growth, civil protection, and environmental impact studies are required to be dealt with. In urban areas, land cover and land use change rapidly due to new construction of the buildings, roads, and other man-made objects. Therefore, monitoring these changes becomes an important issue and many Geographic Information System (GIS) applications suffer from the lack of timely land cover/use information. The geographic databases should be regularly updated with the changes occur on land surface. In order to do that the extraction of geographic features has long been performed manually by human operators, with high accuracy and reliability. However, manual delineation is a very time consuming operation and requires qualified people. For this reason, the automated object

extraction techniques from digital imagery have become a key concern for the modern geospatial applications.

The recent advances in the quality of data acquisition systems along with the desire to analyze data have improved the development of new image processing techniques for automatic object extraction and reconstruction. There exist many techniques to efficiently extract individual objects from images for the purpose of spatial analysis and object retrievals from large-scale image databases as summarized in Baltsavias (2004) and Mayer (2008). In these applications, the main interest was on man-made structures, especially the buildings.

One of the major research areas in urban remote sensing is the detection of buildings. At the first glance, buildings may seem to be simple objects, which can be easily identified and extracted. However, automatic extraction of buildings from large-scale images must handle several difficulties caused by different viewpoints and being in a complex shape and size. Building footprints are one of the fundamental GIS data components and they have been shown to be extremely useful in urban planning, infrastructure development, construction of telecommunication lines, pollution modeling, disaster planning, and many other kinds of urban simulations. In addition, building footprints not only localize the buildings, but also extract valuable information about the structure of building roofs and vertical walls that may not be visible to aerial sensors.

In parallel, with the rapid developments in information technologies, the term “visuality” has become an important fact. This is due to the fact that visualization of 3-d objects greatly improves the ability of human perception. Particularly, in geoscience applications, the spatial objects can be viewed from different perspectives or they can be examined through walking or flying using the virtual reality technology. Among the virtual worlds, virtual cities have become a popular phenomenon. A virtual city is composed of solid 3-d models of buildings, vegetation, terrain, and many more spatial objects.

Buildings are the most essential components of a virtual city model. The 3-d geometry of the buildings can be derived from the 2-d footprints of the buildings using various photogrammetric methods or they can be generated from ground-based laser scanning. Texturing of the building models also helps visualize the final model much better than a simple wireframe representation. Building textures are commonly captured from ground-based or occasionally from airborne remote sensing imagery. The high-quality texture acquisition and mapping is still a challenging task and serves as an important step for the construction of 3-d city models. Therefore, to provide more realistic views it is needed to extract “photorealistic” textures from building facade images. This is achieved by satisfying the elements of photorealism, such as texture/color fidelity, proportion, scale, edge transition, object relationship, etc. The vast majority of those elements are tightly connected with the occlusions, which can be identified as foreground objects that partially or completely obstruct building facades standing in the background. Since occlusions may cause lack of quality in the texturing procedure, the extraction and mapping of the occlusion-free facade textures have become an important priority.

1.2. Objectives and Contributions

The main objective of the present study is to develop an integrated framework for the automatic reconstruction of photorealistic 3-d building models from satellite and ground-level imagery. The supporting objectives can be listed as follows:

- Developing an approach for the automatic extraction of building areas and their corresponding 2-d footprints from high resolution space imagery based on an adaptive fuzzy-genetic technique.
- Automatic 3-d reconstruction of the buildings from 2-d footprints and the normalized digital elevation model (nDSM).

- Developing approaches for the automatic extraction of photorealistic building facade textures from ground-level photos by employing repetitive Watershed segmentation, image rectification, and texture cropping.
- Developing an approach for the semi-automatic removal of the occlusions based on a correlation-based image matching technique.
- Developing an approach that employs GPS coordinates and corresponding bearings of the photo shooting points for the automatic texture assignment and mapping of the occlusion-free images to building facades.

This study focused on those aspects that have received no or less attention in previous works. Specifically, the following contributions are listed:

- For building delineation, a genetic algorithm approach was adapted in order to achieve higher detection rates when compared with the conventional supervised classifiers.
- An adaptive-fuzzy extension was integrated with the conventional genetic algorithm in an attempt to improve the convergence rate by adjusting the genetic algorithm parameters.
- A repetitive watershed segmentation approach was introduced to extract more accurate texture information from the building facades by reducing the over-segmentation.
- A fully automated image rectification procedure was developed to rectify the building facade images that include perspective distortions.
- The removal of occlusions that block the building facades was achieved from a single ground-level image using a developed semi-

automated approach, which provides the means of seamless texture generation.

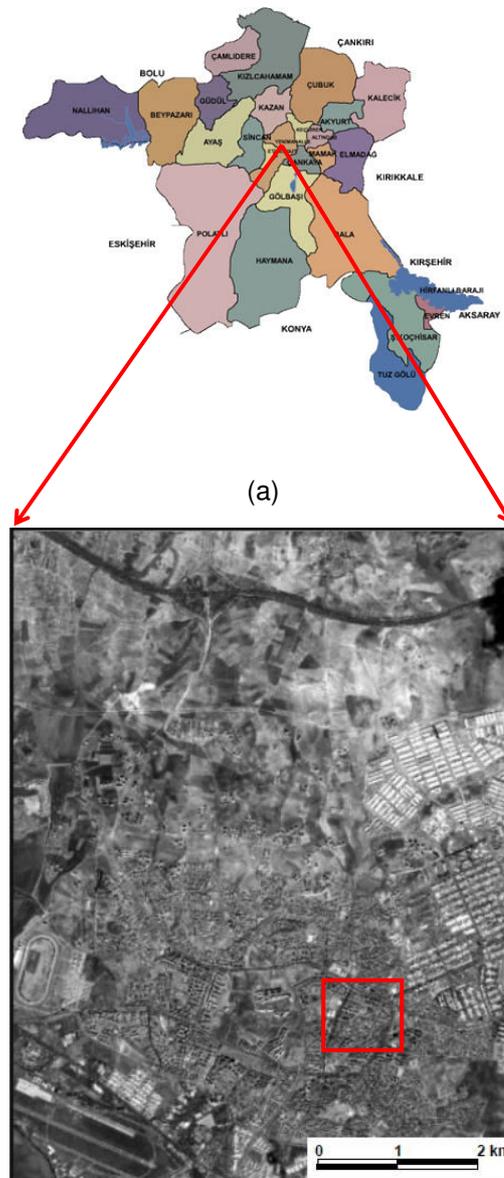
- To achieve fast and reliable texture mapping, an automatic approach that utilizes ground coordinates and bearings was developed for associating the building facades with their corresponding textures.
- For data acquisition, the utilization of low-cost equipments (a GPS receiver and a mid-price digital camera) was presented in the development of 3-d building models.
- Under certain assumptions, the generation of 3-d photorealistic building models was achieved using the minimum number of ground-level images.
- The overall study integrates the extraction of building footprints, retrieval of photorealistic facade textures, and the modeling of the textured buildings in 3-d with a satisfactory level of accuracy and the computational performance.

1.3. Study Area and Data Description

The developed methodology was implemented in a selected urban area of the Batikent district of the city of Ankara, Turkey. Batikent is a planned and regularly developed settlement area, which contains various types of buildings with different shapes and usage, such as residence, industrial, commercial, social, and cultural facilities. It is located on the western corridor of Ankara, lying over an area about 10,000,000 square meters. The district was a housing project of the 1980s, which was the biggest mass-housing project accomplished through cooperatives in Turkey. The project was planned for 50,000 housing units and 250,000 persons (European Resettlement Fund, 2007). The Batikent Housing Cooperation Union was

then founded and the area changed greatly by 1995. Figure 1.1 shows the study area with a 2-level map. Figure 1.1(a) illustrates the map showing the districts of Ankara, while (b) shows the panchromatic image of the whole study area captured by IKONOS. The red rectangle denotes the sub-region used in testing the developed approach at present study.

The data set used includes the IKONOS stereo panchromatic and pan-sharpened images in “Geo” data format acquired on August 4, 2002. The 1-m resolution pan-sharpened IKONOS image of a selected sub-region is illustrated in Figure 1.2.



(b)
Figure 1.1. The study area.



Figure 1.2. Pan-sharpened image of the sub-region used in the present study.

In addition to image data set, the existing digital vector database that contains 3-d lines and height points were also utilized to generate a digital terrain model (DTM). The 1:1000-scaled digital vector data, which was compiled by the Ankara Greater Municipality, General Directory of Water and Sewer System (ASKI) in 1999, covers the metropolitan area of Ankara. Furthermore, a digital topographic surface model (DSM), which was generated from the stereo pairs of the IKONOS satellite images using the OrthoEngine Module of PCI Geomatica image processing software, was also utilized. Both the DTM and DSM data sets were prepared by Koc San (2009) in a former study conducted in the department. Figure 1.3 illustrates the DTM and DSM data sets covering the study area.

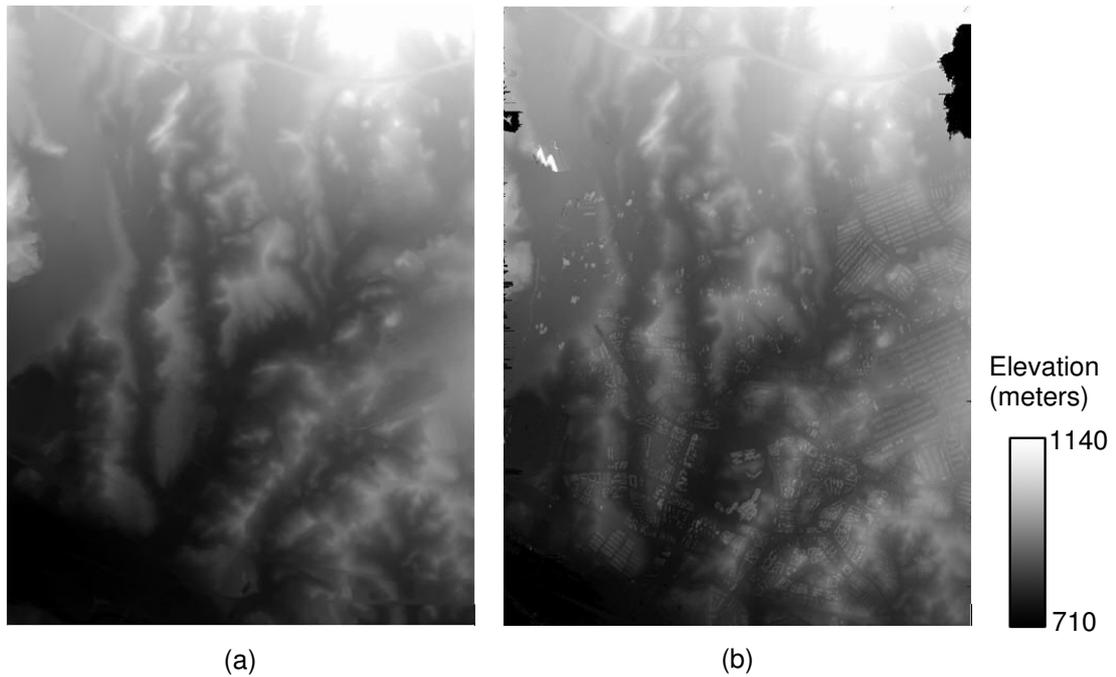


Figure 1.3. (a) DTM and (b) DSM of the study area.

A digital surface model (DSM) includes the objects with their heights above the ground level and the topography, while a digital terrain model is an elevation model of the landscape that does not include above ground objects. To be able to estimate the heights of the man made objects over the terrain, a normalized digital surface model (nDSM) was generated by subtracting DTM from DSM. Then, a threshold (3 m) was applied to nDSM to separate the man made objects. As illustrated in Figure 1.4, the values above 3 m represent the building areas. The workflow for the nDSM generation and the generated nDSM are shown in Figures 1.5 and 1.6, respectively.

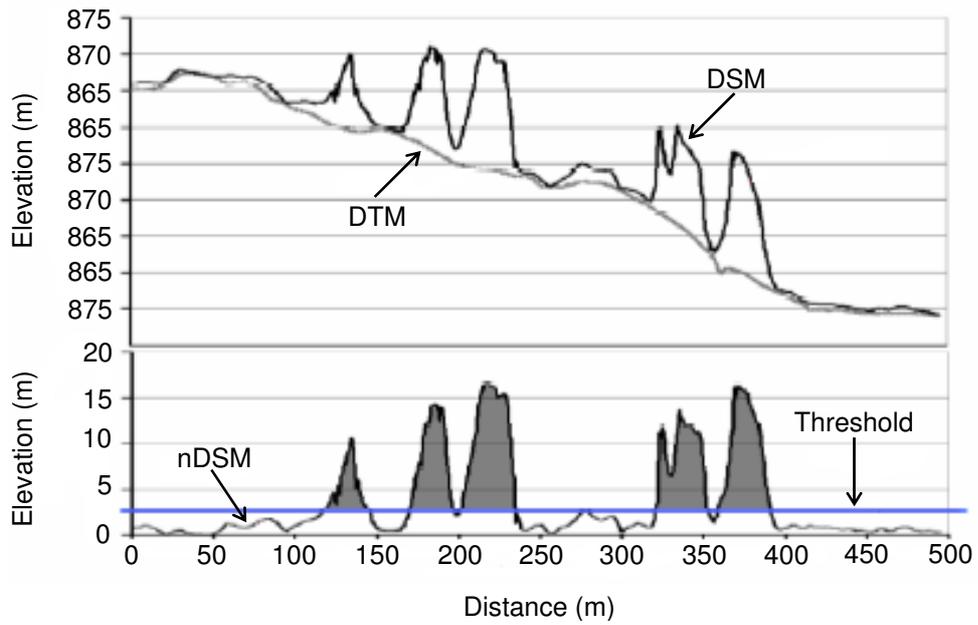


Figure 1.4. The generation of nDSM: The gray areas represent the 3-d features after applying the threshold (Koc San, 2009).

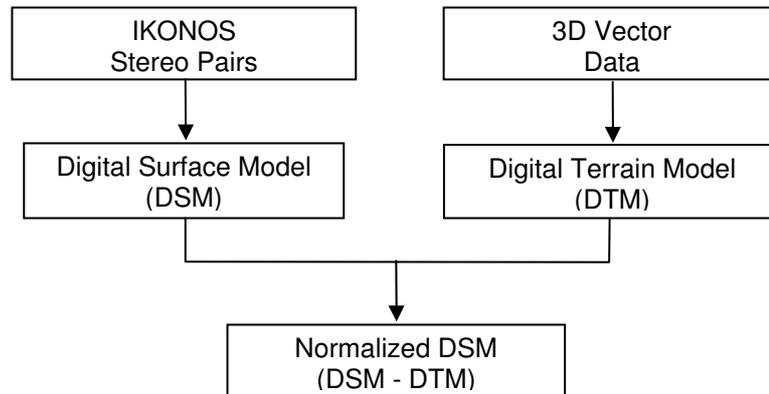


Figure 1.5. The workflow for generating the normalized digital surface model (nDSM) (Koc San, 2009).



Figure 1.6. The generated nDSM of the study area.

One other data source used in the study consists of the ground level images. They were employed for the extraction of the facade textures of the building models. At present study two sets of ground-level images were used. The first set of photos (15 in total) belongs to a sub-area selected in the Batikent district. These photos were acquired by a hand-held digital camera in a clear-sky day on January 9th, 2010. The coordinates and the bearings of the photo shooting points were also measured by a handheld GPS receiver. The scenes from the ground-level data collection procedure are shown in Figure 1.7. Several ground-level building facade images collected in the Batikent district are illustrated in Figure 1.8.



Figure 1.7. Scenes from the ground-level data collection process.



Figure 1.8. The facade images captured in the Batikent district.

The second data set includes the ground level images (5 in total) obtained from eTrims image database, which was created by Korč and Förstner (2009) along with the members of the eTRIMS consortium. The database contains annotated RGB-images of building facades over a hundred buildings captured in several major European cities. The ground truth data is also provided to serve as a basis for the evaluation and comparison of supervised learning approaches to image interpretation. The images selected from eTrims dataset are presented in Figure 1.9.



Figure 1.9. The facade images taken from eTrims image database.

1.4. Software and Hardware Requirements

All the implementations and processes were carried out using the commercial software packages. The implementations of the proposed methodologies were fully performed using the MATLAB programming environment. MATLAB, which stands for “MATrix LABORatory”, is a numerical computing environment and fourth-generation programming language (Matlab online documentation, 2010). It allows matrix manipulations, plotting of functions and data, implementation of algorithms, and creation of user interfaces. Besides, it provides interfacing with programs written in other languages including C, C++, and Fortran. MATLAB contains various toolboxes, which are specialized collections of M-files (MATLAB language programs) built for solving particular classes of problems. In the present study, Image Processing Toolbox™ was widely used in many operations for processing, analysis, and visualization of images along with the algorithm development. The Virtual Reality Toolbox™ was also employed in the

generation of 3-d building models. This toolbox lets the user view and interacts with dynamic system simulations in a 3-d virtual reality environment (Matlab online documentation, 2010).

The Virtual Reality Toolbox™ can be integrated with virtual reality modeling language (VRML) nodes using the structured programming ability of MATLAB. VRML is a standard file format used in the representation of the 3-d interactive vector graphics. GeoVRML, used in this study, is an extended version of VRML. It is designed to support geographic applications such as 3-d terrain modeling, urban planning, GPS data visualizations, etc. This standard is an official working group of the Web3D Consortium formed in 1998 and aims to represent geographic data using the VRML (Reddy *et al.*, 2001).

The generated 3-d models were visualized using the “Cortona” viewer. This viewer works as a VRML plug-in for the widely used internet browsers, such as Internet Explorer and Mozilla Firefox. It is not only a viewer but also it allows a wide-range of 3-d applications spanning from the visualization of the scientific data to advanced 3-d online services. The navigation modes of “flying”, “walking” and “studying” are available in Cortona together with the features, such as plan, pan, turn, and roll (Cortona3D Viewer User’s Guide, 2009). The interface of the basic functionalities is illustrated in Figure 1.10.

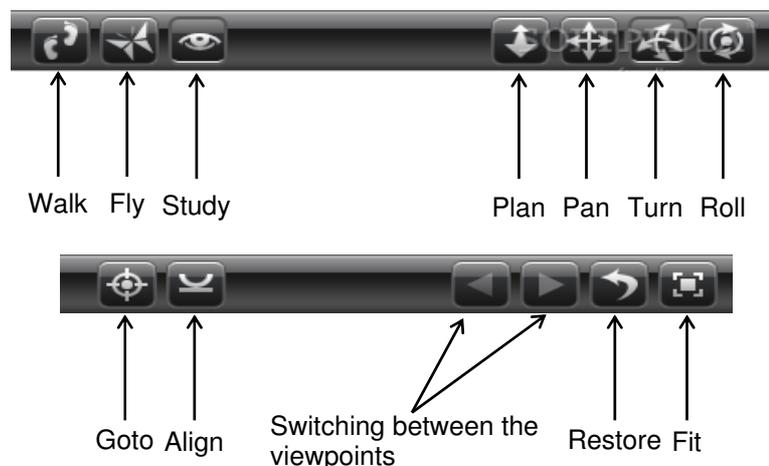


Figure 1.10. The screenshot of the Cortona 3-d toolbar.

GIMP is the other software package used in the present study. It is a free software for raster graphics editor, which is primarily used as an image retouching and editing tool. GIMP also accomplishes the fundamental image operations, such as resizing, editing, creating photos, combining multiple images, and conversion between different image formats (GNU Image Manipulation Program, 2010). In this study, the reference data needed for facade texture extraction and rectification were generated using the GIMP software.

The PCI Geomatica software was also used in the current study. PCI Geomatica is a complete and integrated desktop software that features remote sensing, digital photogrammetry, geospatial analysis, map production, mosaicing, and automated product systems capabilities. It also enables users to apply imagery in support of a wide range of applications such as the environment, aerospace and defense industries, and satellite receiving stations (PCI Geomatics, 2010). In the current study, PCI Geomatica was utilized in determining the building heights from nDSM.

In addition to commercial software packages used, two fundamental hardware equipments were employed for data acquisition. Samsung WB 500 digital camera having 10.2 mega pixels resolution and 10x optical zoom lens was used for collecting the ground-level building facade images (Figure 1.11). This digital camera has a focal length starting at 24mm making it one of the widest lenses ever found on a compact digital camera. Compared with the other digital cameras with longer zoom lenses, this model has been packed into a compact body size (Samsung WB500 Review, 2009).



Figure 1.11. Samsung WB 500 digital camera.

The Magellan Explorist XL handheld GPS receiver with 3-meter position accuracy was used for measuring the coordinates and bearings of the photo shooting points (Figure 1.12). This device has a practical usage that boasts a high contrast color TFT screen, a built-in compass, plus maps and expandability. Moreover, it has an ability to store an unlimited number of waypoints, routes and track logs through SD-card memory expansion. The device also possesses an advanced PC-style file management system and high-speed USB port allows users to download and access compatible Magellan mapping software (Magellan Basic User Manual, 2006).



Figure 1.12. Magellan Explorist XL handheld GPS receiver.

1.5. Organization of the Thesis

This thesis is composed of eight chapters. The next chapter (chapter 2) provides the background information and the literature review about 2-d building acquisition, 3-d modeling, facade texture acquisition, and mapping of buildings in terms of data requirements and the methodology.

The overall methodology, which comprises the inputs, outputs, and the proposed approaches of the framework, is presented in Chapter 3. Besides, the developed software **PhotoREAListic BUILDing Modeling (PREBUM)** is introduced. The panels and the menu items that the main graphical user interface of PREBUM contains are shown with the snapshots.

In chapter 4, the approaches developed for 2-d building extraction and delineation from high-resolution satellite imagery are presented. First, the genetic algorithm-based adaptive-fuzzy 2-d building extraction approach is given. Then, the assessment of the accuracy is explained. Second, the 2-d delineation approach, which is carried out based on image geometry and morphology, is emphasized. This is followed by the explanation of the accuracy assessment method based on positional accuracy.

Chapter 5 presents the building facade texture acquisition approach. It is initiated with a facade segmentation stage using the watershed transform. This is followed by the description of the automatic technique, which was developed for the rectification of the extracted facade images. Then, the occlusion-removal, which employs an image matching algorithm based on the correlation and post-processing stages, is provided. The methods used for assessing the accuracies are also presented.

In chapter 6, the 3-d modeling and texture mapping procedures are discussed. First, the generation of the untextured 3-d solid models is described. Then, the texture selection method based on GPS coordinates is explained along with the mapping procedures of the facades and roofs.

Finally in this chapter, the automatically generated photorealistic 3-d building models are presented.

In the next chapter (chapter 7), the experimental results and the performance issues of the developed approaches are given in detail. In addition, the discussions of the results are emphasized by presenting the shortcomings and the special cases that may make the approach fail.

The final chapter concludes the developed approaches and contains the recommendations that arise from this study.

CHAPTER 2

BACKGROUND AND PAST STUDIES

In this chapter, first the feature extraction basics related to various remote sensing technologies are given. Next, the past studies, which were conducted in the field of 2-d building extraction and 3-d modeling, are presented. Finally, the literature review about building texture acquisition and mapping is given. The occlusion removal techniques and the extraction of the facade textures are also described in this section. All these studies are classified from past to present with respect to different data sources.

2.1. Feature Extraction from Remote Sensing Images

In pattern recognition, feature extraction is defined as the transformation of the input data into the set of features. If the extracted features are carefully chosen, it is expected that the feature set extracts the relevant information from the input data in order to perform the desired task. Feature extraction can be used in the area of image processing, which involves algorithms to detect and isolate various desired portions or shapes (features) of a digital image. Extraction of geographic features, such as building footprints, roofs or facade textures is one of the practical applications of the geographic feature extraction process. It simply consists of the differentiation of the object (foreground region) from their surroundings (background region).

Large volumes of remotely-sensed data are being collected by an increasing number of sophisticated airborne, spaceborne or ground-based sensor systems. These data can be processed and analyzed manually and/or

automatically to extract specific features of interest. In the case of automatic building feature extraction, optimal spatial resolution, high spectral and temporal resolutions of the sensor are considered as the main requirements. To choose the optimum data source, it is important to know the data types with different characteristics. This is due to the fact that the images collected by various sensors have different spatial, spectral, and temporal resolutions. Of the various sensor types, the airborne remote sensing technology is one of the major data source for gathering information about the buildings due to high spatial resolution. Conventional aerial photographs, digital multispectral scanners, airborne SAR and airborne laser scanners are the main types of the airborne platforms.

Aerial photos possess detailed information due to its high spatial resolution. Thus, it is suitable for observing buildings or other infrastructures from the top view or side view (in facade texture acquisition purposes). In addition, stereo photographs give an opportunity to derive height information of the buildings. However, to get such detailed information, a high price must be paid. Digital multi-spectral scanners provide digital imagery using the CCD and CMOS sensors. These sensors have both high spatial resolution and multi-spectral information. Moreover, these scanners have various advantages, such as size, weight, dynamic range, and optical sensitivity. Airborne SAR is another platform, which is not widely used due to the complex image processing technologies in data acquisition and processing. In addition, the side looking characteristic of the system limits the radar applications for urban areas. Further, it provides considerable spatial resolution (0.5 m – 10 m), which allows nighttime data acquisition and operates in bad weather conditions to some extent (Lillesand *et al.*, 2007). Airborne laser scanners are used to measure three dimensional points, distributed over the terrain surface and on objects rising from the ground. For building extraction, information derived from the laser scanners is quite important in creating Digital Surface Model (DSM) of the build-up areas. Light Detection and Ranging (LIDAR) is a special airborne laser scanning technology that employs an airborne scanning laser rangefinder to produce detailed and accurate topographic

surveys. LIDAR can be used to accurately measure the topography of the ground, even where overlying vegetation is quite dense. In the case of building extraction, LIDAR can generate accurate information about building height and volume calculation. However, laser scanning produces blind images and needs to be integrated with the other data sources. Further disadvantages of this technology are the high cost and poor data availability.

Spaceborne remote sensing technology is the other important data source in collecting information with respect to urban applications, such as in the detection of the building footprints. Its major superiority is the large area coverage, which is quite difficult to achieve in airborne remote sensing technology. This technology possesses several other advantages, such as repetitive and continuous coverage, having variety of sensors, possibility of acquiring stereo images and elevation data by using pointable sensors, fast production and low cost. Besides these advantages, spaceborne systems have some drawbacks, such as relatively low resolution data, delayed initial image acquisition, lack of central inventory of available satellite locations, large image file size, and frequent data incompatibility problems (Lillesand *et al.*, 2007).

Recently, the terrestrial data acquisition systems have become popular in urban applications. In particular, ground-level laser scanning systems are widely used and turned out to be a very promising alternative for many kinds of surveying applications. These systems provide the rapid acquisition of large 3-d data. Afterwards, this data can be often profitably combined with high resolution colored digital images to generate 3-d representation of the environment, such as building facade textures. The chief advantage of these systems is the more realistic representation of the models compared with the representation obtained by a single picture or collection of pictures. This is because of the high level of detail together with a good geometric accuracy. Nowadays, in virtual reality systems, the use of laser scanner based 3-d models opens new perspectives for urban modeling and texturing. However, the ground-based laser scanning systems have several drawbacks. The first

and the most important drawback is the cost. In terms of service fees, 3-d laser scanning costs much more than the other scanning systems. The second drawback is the huge size of the scanning files. The users must have powerful computer hardware to accommodate the significant memory requirements of the data (El-Hakim *et al.*, 2007).

Terrestrial video images also play an important role in many close-range urban applications. In the last decade, 3-d reconstruction and texturing of building objects from terrestrial video images has received much attention. The economic and flexible data acquisition is the main advantage of this technique. Besides, the occlusions are handled easily by using the multiple view ability. It also provides the capability of capturing building facades as seen from the street level, as well. The major drawback of the video data is the modeling and texturing difficulty of the building tops, as in the laser scanning systems. Moreover; due to different perspectives, scales, contrasts, color shadings, and other properties, these variations need to be adjusted in order to integrate them into a seamless texture mosaic. On the other hand, in a variety of studies, the extraction of building facades has been performed from a “single image” instead of sequence of images. A single image is cheaper to capture and requires less memory storage. It also provides higher resolution data, which has a crucial importance in visualization of textured building models. Moreover, a single photo reduces the number of scenes to be processed for texture mapping. However, despite many advantages, single images have a couple of disadvantages. The most important disadvantage is the difficulty of occlusion removal. Further, it takes too much time and effort to extract the facade structures for large urban areas. In addition, capturing the building roofs is a troublesome task.

2.2. 2-D Building Acquisition and 3-D Modeling

As a general tendency, large variety of aerial images in single, multiple or overlapping forms are employed in many studies. Besides, airborne LIDAR

and SAR applications have mostly been encountered in the recent studies. The hybrid methodologies that provide the integration of LIDAR and aerial imagery are commonly observed, as well. The use of ancillary data, such as digital surface model and topographic map information are also utilized by certain researchers. In this section, the spaceborne 2-d building acquisition and 3-d modeling studies are also summarized. It is observed that high resolution satellite images, such as IKONOS, SPOT, QuickBird, GeoEye, and WorldView are mostly used in recent studies. Depending on the application; the single and stereo use of panchromatic, multispectral, and pan-sharpened spaceborne imagery are encountered. The use of hybrid data set, such as the integrated SAR-optical imagery and LIDAR-optical imagery data sets are also experienced in certain studies. Moreover, in several studies the use of DEMs and DSMs, which are generated from different spaceborne sensors, can be observed.

2.2.1. 2-D Building Acquisition and 3-D Modeling from Airborne Images

In many applications of 2-d building acquisition, airborne imagery is widely used. Especially, in most of the early studies, black and white aerial images were used as a single data source. In a study conducted by Huertas and Nevatia (1988), a generic model of shapes of structures were used by assuming that the buildings are rectangular or composed of rectangular components such as box, 'E', 'T' and 'L' shapes. Their method was composed of four steps including line and corner detection, labeling of the corners based on shadows, tracing of object boundaries, and finally the verification of hypotheses. In addition, the shadows cast by the buildings were also utilized in verification of the buildings and estimation of the building heights. More generally, their method showed an example of how generic model knowledge can be used in extracting objects in real, outdoor scenes.

One of the frequently used applications of aerial photography is the extraction of the buildings from their shadows. Irvin and McKeown (1989)

state that the shadows are usually among the darkest areas in images and their extraction can be feasible using the image processing techniques. They developed four methods to estimate the grouping of the related structures. These are the prediction of structure shape, grouping of the related structures, verification of individual structures, and the structure height estimation. In each method, the main focus was on the relation between structures and their cast shadows. Their study showed that these techniques can be used to generate new pieces of information in the scene interpretation.

In a similar study conducted by Lin *et al.* (1994), a system was introduced for the detection and description of buildings in aerial scenes. The monocular views of arbitrary aerial scenes were utilized to detect and describe the buildings. Since low-level segmentation methods give highly fragmented segments, the perceptual grouping approach was used. The shape properties and shadow information was utilized to form and verify the hypotheses generated by the grouping process. This study also provided the 3-d descriptions of the buildings. The proposed system was tested on a number of examples and proved to be successful in overhead and oblique views.

A semi-automatic approach was proposed by Sahar and Krupnik (1999) for the 2-d extraction of buildings from large-scale images. Their goal was to reveal the 3-d outlines of the buildings. Their approach was composed of three stages, which are preprocessing, monoscopic processing, and stereoscopic processing. In the preprocessing stage, two different kinds of edge segments, one belonging to shadow areas and the other belonging to other features, were produced. The monoscopic processing stage was aimed at selecting topologically organized sets of edge segments. The potential building chains were created in this stage. In the last stage, the outlines of the buildings and the corner 3-d coordinates were extracted. The results were found to be quite encouraging in detecting outlines of buildings

accurately and reliably through an automatic procedure that exploits stereo, shadows, and a priori geometric information.

Further, Noronha and Nevatia (2001) performed the automatic detection and modeling of buildings from multiple aerial images. They described a system that detects and constructs 3-d models for rectilinear buildings. The main steps of the system include; the grouping of line segments, junctions, and parallels together with triples and U-contours, formation of roof hypotheses, wall and shadow verification. Their system was tested on a large number of real examples with successful results.

A subsequent study that utilizes aerial imagery in building detection was conducted by Gerke *et al.* (2001), who achieved the automatic extraction of buildings from the combination of aerial color infrared images and the digital surface model in an urban environment. The knowledge about the scene and the geometry of the objects were represented by means of a generic scene model. The buildings were reconstructed using invariant geometric moments leading to orthogonal geometric models.

Hofmann *et al.* (2002) utilized laser scanner data and topographic map information for detecting buildings using a knowledge-based approach. Their study was divided into two parts, which are segmentation and detection. First, the region based segmentation method was applied to delimit the objects. This was followed by locating the recorded houses by using a scanned topographic map, which was filtered and converted into a vector format. The study provided promising results where more than 95% of the buildings in the dataset were detected successfully.

The joint analysis of SAR, LIDAR, and aerial imagery was carried out by Gamba and Houshmand (2002) for the simultaneous extraction of land cover, DTM, and 3-d shapes of the buildings. The aim of this study was to discriminate between different objects in the scene using the 2-d and 3-d characteristics of the objects. This was achieved by a proposed algorithm,

which consists of grouping the pixels into segments, finding and enlarging the seeds for the planar surfaces, and connecting the planes into larger regions. The results were not completely satisfactory due to the LIDAR measurements, which suffer from uncertainties and errors in sharp boundary representation.

A novel method for building extraction from the concept of fitting CSG (Constructive Solid Geometry) primitives to aerial images was proposed by Tseng and Wang (2003). In this study, a semi-automatic procedure was adopted for performing high-level operations, such as building detection and model selection interactively by the operator and performing optimal model-image fitting automatically using a least-squares fitting algorithm. The test results were encouraging and supported the theory of model-based building extraction.

A model-based approach to recognize and reconstruct buildings from multiple aerial images was developed by Jaynes, *et al.* (2003). First, the buildings were segmented by detecting and grouping the lines that correspond to building rooftop boundaries. This was followed by the reconstruction process that makes use of the corresponding digital elevation (DEM) map. The approach reconstructed the various building types, such as peaked, flat, multi-level flat, and curved surfaces. The approach was evaluated on several datasets and it was concluded that the proposed two-phase approach leads to accurate reconstruction of a wide variety of building types while still retaining robustness.

Oriot (2003) performed a novel semi-automatic method to delineate buildings based on statistical active models (statistical snakes). This method purely utilized a stereoscopic pair of images in extracting the buildings. Besides, the method was proved to limit the number of interactions. The algorithm was successful to find solutions far away from the initialization area and buildings were described accurately in general.

Rottensteiner (2003) presented a methodology for the automated generation of complex 3-d building models from point clouds generated by the LIDAR sensors. The first step in their workflow was to estimate the DTM and DSM from LIDAR points. Then, the initial building patches were created by subtracting DTM from DSM. After that, the morphological operators and texture analysis were performed for eliminating the false alarm areas. After detecting the building regions, the following steps were accomplished by the proposed workflow: (i) detection and grouping of roof planes, (ii) model generation, (iii) consistent estimation of the model parameters, and (vi) model regularization. All buildings in the test area were detected, although their outlines had a rather complex shape. It was also reported that the constructed models resemble the roof shapes well according to a visual inspection.

In a more recent study conducted by Kim and Nevatia (2004), an approach was proposed for detecting and describing complex buildings with flat or complex rooftops by using the multiple overlapping images. Probabilistic reasoning, level-of-details, and cues (layers) derived from elevation data were used at different stages of the proposed approach in order to manage the huge search space for rooftop boundary hypotheses. The generated 3-d rooftop hypotheses were verified with the evidence collected from the images and the elevation data using Expandable Bayesian Networks (EBNs). As the final step, the overlap and rooftop analyses were performed for finding the final complex building models. The experimental results were reported to be promising for complex buildings.

Cho *et al.* (2004) proposed a practical method for 2-d building detection from airborne laser scanning data. They introduced a concept of pseudo-grid (binning) into raw laser scanning data to avoid the loss of information and accuracy due to interpolation. The proposed method comprised low and high level processes. The generation of pseudo-grid, noise removal and segmentation were included in the low level process, while the high level process basically consists of grouping, tree removal, and building detection

steps. According to experimental results, the proposed approach was found to be promising.

In a method presented by Peng and Liu (2005), the use of monocular urban aerial images without any prior knowledge of illumination was presented in the extraction of buildings in dense urban areas. Based on a building concept model, 2-d building extraction was performed in two stages that are (i) sunshine parts extraction, and (ii) self-shadow parts extraction. The images were initially simplified and segmented as sunshine parts, sunshine parts of high objects, and sunshine ground based on region-oriented radiometric features. To verify initial segmentations, a method was proposed for estimating the shadow cast direction by a shadow context model. Finally, refinement of the extracted buildings was accomplished with the aid of context and a modified partial snake model. The results showed that the approach can detect both flat and gable roofed buildings from a complicated background of images. For the shadows, the integrities of road were successfully maintained.

Characterization of buildings, which consist of building identification and height estimation using L-Band polarimetric interferometric synthetic aperture radar (SAR) data, was performed by Guillaso *et al.* (2005). Initially, a polarimetric interferometric segmentation was carried out to differentiate buildings from their surroundings. Three main classes were identified that are single bounce as surface, double bounce as building, and volume scattering as vegetation. Once the buildings were localized, the building heights were retrieved using a phase-to-height procedure. The height estimates of most of the buildings were reported to be acceptably accurate.

In a study conducted by Simonetto *et al.* (2005), the extraction of 3-d rectangular buildings from stereoscopic high resolution airborne radar images, recorded by a SAR airborne sensor was carried out. To detect the buildings, an adapted processing scheme was proposed that includes the Hough transform and a stereoscopic refinement stages. The building heights

were also measured by monoscopic and stereoscopic measures with the aid of previously extracted building footprints. Experiments were performed on images representing an industrial area. The results showed the potentiality of the method for rectangular building extraction and DEM generation.

An automated extraction procedure for 2-d building footprints from airborne LIDAR data was presented by Wang *et al.* (2006), who offered a novel Bayesian technique for automatically constructing building footprints from a pre-classified LIDAR point cloud. Initially, a bounded-error approximate building footprint was computed by the algorithm using an application of the shortest path algorithm. Then, the most probable building footprint was determined using linear optimization and simulated annealing techniques by maximizing the posterior probability. The algorithm was tested on 300 buildings in the data set and it was reported that the building footprints were obtained accurately.

Lu *et al.* (2006) proposed an approach for the automatic detection of buildings from aerial images using the combined analysis and interpretation techniques that include classification, shape modeling, and fusion. A dense DSM was obtained by stereo image matching and the preliminary building interest areas were revealed by using the results of multi-band classification, the DSM, and Normalized Difference Vegetation Index (NDVI). From these areas, a shape modeling algorithm was implemented in order to delineate their boundaries precisely. As a final step, Dempster-Shafer data fusion method was applied to detect buildings from the combination of three different outputs coming from DSM, K-Means clustering, and the delineated buildings. A number of test areas, which include buildings of different sizes, shape, and roof color were investigated and the results were found to be encouraging.

In a study conducted by Zhang *et al.* (2006), a framework was introduced for automatic extraction of 2-d building footprints from LIDAR measurements. As a first step, the ground and non-ground LIDAR measurements were

separated using a progressive morphological filter. Then, building measurements were identified from non-ground measurements by a region growing algorithm based on the plane-fitting technique. In the final step, the raw footprints were obtained by connecting the boundary points. The framework was tested on urbanized areas such as large institutional, commercial, and small residential buildings. A quantitative analysis revealed that the total of omission and commission errors for the extracted footprints was about 12% for both the institutional and residential areas.

Rottensteiner *et al.* (2007) developed a method for the detection of the buildings by the Dempster-Shafer fusion of airborne laser scanner (ALS) and multi-spectral images. The first image was an RGB orthophoto with a resolution of 0.15 m and the other was a geo-coded RGB together with a color infrared band with 0.5 m resolution. First, the image was classified based on a pixel-based approach. The classified image was improved by considering the uncertainty of the NDVI and by post-classification. It was concluded that the detection of the buildings are dependent on the building size. They also stated that the small buildings were detected incorrectly, in general.

In a study conducted by Xu and Jin (2007), the automatic detection and reconstruction of 3-d building objects from multi-aspect meter resolution SAR images were carried out. Initially, the imaging features of the object were generated as a priori knowledge. Then, the scattering image of the object were identified and extracted. Next, a statistical description of the object image and its coherency was given. Lastly, an automatic algorithm to match the object images of different aspects was designed and the reconstruction was performed. Reconstruction of building objects from their multi-aspect images revealed the fidelity of the whole process chain and the feasibility of 3-d objects.

In a very similar study, the recognition of urban buildings was carried out from multi-aspect high-resolution interferometric SAR (InSAR) data by Thiele

et al. (2007), who proposed two approaches for building detection and reconstruction. The first approach exploited the frequently observed lines of bright double-bounce scattering, while in the second, the extended buildings were extracted by grouping the primitives of different kinds. In short, both approaches focused on the recognition of buildings supported by knowledge-based analysis considering the SAR-specific effects, such as layover, radar shadow, and multipath signal propagation, observed in urban areas. The detection rates were computed to be 52% and 76%, while the false alarm rates were found to be 78% and 11% for the first and second approaches, respectively.

Further, Lee *et al.* (2008) proposed a new building detection and description algorithm from the fused LIDAR and photogrammetric data set. Their algorithm was composed of three steps. In the first step, initial building regions were extracted from LIDAR data. Next, the extraction of coarse building boundaries was carried out based on LIDAR results with region segmentation and merging from aerial imagery. In the third step, the precise building boundaries were extracted based on the coarse building boundaries using line segments matching and perceptual grouping. Experimental results demonstrated that the proposed algorithm yields accurate and reliable results on multi-sensor data.

The extraction of building features from remotely sensed elevation and spectral data based on mathematical morphology was presented in a study conducted by Vu *et al.* (2009). The elevation data derived from LIDAR were used as the primary data to delineate the structural information. On the other hand, the spectral data (a true color ortho-photo with 20 cm spatial resolution) was used as an additional source to remove vegetation and classify the building roof material. Object-based completeness and correctness were measured to report the accuracy of extraction in a quantitative way. Of the two test areas, first produced a correctness percentage of 83 while the second yielded 67. The completeness

percentages were found to be 79 and 73 for the first and second test areas, respectively.

In a very recent study conducted by Michaelsen *et al.* (2010), the extraction of building polygons was carried out from aerial SAR images. As a preprocessing step, the image was scaled down and then a morphological opening operation was applied to enhance and isolate the spot structures. Besides, the image was tiled into overlapping sub-images and on each image a morphological closing operation was applied to enhance thin line structures and close gaps. In the next step, short line segments were extracted from the sub-images using the squared averaged gradient filter. The third step was the grouping stage, in which the objects were accumulated. In the final step, the decision for the building polygons was constructed and the post-processing operations were performed. It was reported that promising results were obtained in recognition of the buildings from high resolution SAR data.

Wang *et al.* (2010) presented a new approach to extract buildings from high-resolution polarimetric synthetic aperture radar (PolSAR) data by employing both region-based and edge-based information. In the initial step, raw region and edge information were obtained by low-level detectors. Next, the rectangle features were extracted from the results of edge detection. In the final step, a novel Markov Random Field (MRF) framework was proposed for rectangles. Under this framework, the building rectangles were identified from the optimized rectangle candidates by minimizing the total energy. The effectiveness of the proposed method was tested using the real fully PolSAR data.

Ahmadi *et al.* (2010) proposed a new method for building boundary detection and extraction based on active contour model from high resolution aerial images. In their model, all building boundaries were detected by introducing certain points in the buildings' vicinity. Different from the traditional snake model, the proposed approach provided the detection of most relevant building boundaries without requiring height data and additional information

to distinguish between buildings and other features. Although the accuracy was found to be excellent in terms of building boundary extraction, some failures were also reported. This was due to the radiometric similarity between building roofs and the image background.

2.2.2. 2-D Building Acquisition and 3-D Modeling from Spaceborne Images

As in airborne imagery, recent high-resolution spaceborne images also provide a valuable data source for the acquisition of 2-d and 3-d building information. In a study conducted by Fraser *et al.* (2001), 3-d reconstruction of the buildings from high resolution IKONOS stereo imagery was reported. The focus of their study was on geopositioning accuracy, radiometric quality and attributes of the imagery that support building feature extraction. The results were reported to be successful such that the IKONOS stereo imagery has the potential to yield 1-m geopositioning accuracies and better in the context of building reconstruction.

Lee *et al.* (2003) presented a building extraction approach that utilizes the classification results conducted using both the multispectral and panchromatic IKONOS images. The classification results were then utilized for estimating the approximate locations and shapes of the candidate buildings. Their fine extraction was carried out in the corresponding panchromatic image through segmentation and squaring. The building squaring approach, which is based on the Hough transform, was also employed for detecting and forming the boundaries of rectilinear buildings. It was reported that 64.4% percent of the buildings were detected, extracted, and accurately formed.

Shackelford and Davis, (2003) developed a combined fuzzy pixel-based and object-based approach for discriminating the buildings from other urban land cover classes using the pan-sharpened multispectral IKONOS imagery. First,

a pixel-based hierarchical fuzzy classifier was carried out to categorize individual image pixels. Next, a multiresolution segmentation technique was used to segment the image to facilitate further object-based classification. Finally, an object-based fuzzy logic classification that utilizes shape, spectral, and neighborhood features was performed to differentiate between the buildings and the impervious surface classes in a dense urban environment. The classification accuracies were computed to be 76%, 81% and 99% for buildings, impervious surfaces, and roads, respectively.

The detection of building outlines based on the fusion of SAR and optical features were employed by Tupin and Roux, (2003). The aim of this study was to define areas of interest for building height reconstruction in radargrammetric or interferometric applications. The methodology was divided into two parts. In the first part, the extraction of partial potential building footprints (linear features) was carried out on a SAR image. Then, the shapes were detected on the optical image using the previously extracted lines. It was concluded that the detection of big buildings was difficult for many reasons. For the middle and small buildings, the detection was found to be rather satisfactory.

Wei *et al.* (2004) developed an algorithm that utilizes the clustering and edge detection operations to extract buildings from high-resolution panchromatic Quickbird images. First, an unsupervised clustering was performed to extract building shadows. Next, the candidate building objects were extracted from the clustering classes. Finally, the Canny edge detection operator was applied to detect the edges of the candidate building objects. Besides, the building boundaries were refined and some false building objects were excluded. The building extraction results were compared with the manually delineated results and the proposed algorithm was found to be quite efficient.

An integrated strategy for automatic extraction of buildings from 1-meter resolution satellite imagery of urban areas was demonstrated by Jin and Davis (2005). Buildings were extracted using the structural, contextual, and

spectral information. As the first step, a series of morphological opening and closing operations were applied to build a differential morphological profile (DMP), which consists of image structural information. Then, the shadow regions were extracted by DMP to provide reliable contextual information. This was followed by the extraction of the bright buildings using the spectral information. By combining the structural, contextual, and spectral information, about 72% of the building areas were extracted with a quality percentage of 58%.

Further, Kim *et al.* (2006) proposed a new algorithm for extracting building lines from monoscopic high-resolution satellite images. The approach was based on the extraction of the lines from rectangular-shaped building roofs with relatively large size. This was achieved by line voting and matching. The algorithm was initiated by an input point selected manually on a building roof. A region of interest was then defined, centered on the input point and within the region lines extracted. After that, a line voting process was applied to estimate the initial orientation and position of a building line. Finally, the orientation and position were refined using a least squares matching process. Two Ikonos images were used in the assessment of the performance and the proposed algorithm extracted 83% of the building lines.

A semi-automatic approach to extract buildings in structured and unstructured urban settlement areas from Quickbird imagery was presented by Mayunga *et al.* (2007). The method was based on radial casting algorithm to initialize snake contours. After employing the image pre-processing operations, the snakes contour was initialized. This was followed by the fine measurements of building outlines, which was performed by the modified snake model. In all test areas, buildings with different shapes and orientation were extracted with a reliable accuracy.

In a study conducted by Sohn and Dowman (2007), automatic extraction of building footprints was employed from the fusion of IKONOS imagery using the pan-sharpened multi-spectral bands and the airborne LIDAR data.

Initially, a laser point cloud in 3-d space was recognized as an isolated building object. Then, the rectilinear lines around building outlines were integrated to compensate the weakness of data-driven and model-driven methods. Finally, a full description of building outlines was accomplished by merging the convex polygons. The evaluations showed that the delineation performance was found to be around 0.11 (the branching factor), the detection percentage was computed to be 90% (the correctness), and the overall quality was reported to be about 80%.

Inglada (2007) proposed an image processing system for the detection and recognition of man-made objects from SPOT 5 supermode THR (Trés Haute Resolution or very high resolution) images having 2.5 m resolution. A supervised learning approach based on support vector machines was used to provide the learning of a generic model for each class of objects by utilizing a geometric characterization of the examples in the database. The main novelty of this study was the use of a high number of geometric image features in order to characterize several classes of objects that have different geometric properties. The results revealed the possibility of discriminating the several classes of objects with the classification rates higher than 80%.

In a more recent study, Liu *et al.* (2008) established a new general semi-automatic building extraction method from high resolution satellite imagery. To extract the precise building roof boundary, two different approaches (region-based and feature-based) were integrated. Region-based approach was used to discriminate the small and simple rectilinear rooftops from its background. On the other hand, in the feature-based approach, the precise positions of complex rooftops were delineated by employing pose clustering and model matching techniques. Integration of these approaches provided the extraction of buildings from simple rectangle rooftop to complicated ones. The tests on Quickbird imagery yielded an accuracy of 75% in the extraction of the regular building rooftops.

Sirmacek and Unsalan (2009) proposed methods to detect urban areas and buildings from panchromatic VHR IKONOS images. The methods were based on scale invariant feature transform (SIFT) keypoints, multiple subgraph matching, and graph-cut methods. The algorithm was initiated by picking two template building images with their corresponding SIFT keypoints, one representing dark buildings and the other representing bright buildings. The SIFT keypoints were also obtained for the test image. Next, the urban area was detected by applying a multiple subgraph matching between the template and test image SIFT keypoints. Then, separate buildings were detected from the formerly detected urban area using a novel graph-cut method. By using diverse and representative test sets, very promising results were reported on automatic detection of urban areas and buildings.

Ioannidis *et al.* (2009) proposed a knowledge-based method to control the suburban informal buildings. The method was based on the use of high resolution images and application of automatic change detection by computation and comparison of digital surface models and building extraction techniques. The basic idea behind the proposed approach was that the construction of a new building in the area of a construction site would appear as a “change” in DSM. For the cases where the area of interest was especially large, the satellite imagery was preferred. With the proposed algorithm, 72% of new buildings were detected using the fully automated procedures.

In a different change detection study, the automatic detection and delineation of the buildings from high resolution space images were carried out by Koc San (2009). The proposed approach was developed for updating the buildings of an existing vector database making use of spectral values, Digital Elevation Model (DEM) and model-based extraction techniques. Initially, the building areas were detected by image classification and normalized Digital Surface Model (nDSM), which was computed by subtracting Digital Terrain Model (DTM) from DSM. After that, the buildings in

the existing vector database were updated through evaluating the detected building areas and using the proposed model-based building extraction technique. The results showed that the proposed approach was quite satisfactory in detecting and delineating the buildings from high resolution space images.

In a study conducted by Karantzalos and Paragios (2009), the problem of automatic building extraction/segmentation from satellite images was aimed to solve. The main focus of the study was to overcome the limitations of the existing inaccurate data-driven segmentation that is caused by misleading low level information derived from shadows and occlusions. To do that, they introduced a novel recognition-driven framework that accounts for automatic and accurate building extraction. They also demonstrated the integration of prior knowledge on multiple building shapes into the segmentation process. The qualitative and quantitative evaluations demonstrated the potential of the proposed approach, which was supported by the very promising experimental results.

In a recent study, Lafarge *et al.* (2010) demonstrated a new approach for building reconstruction from satellite DSM having 0.7-m resolution. As a first step, the extraction of 2-d supports of the urban structures was carried out either automatically or interactively. Next, 3-d block was positioned on the 2-d supports using a Gibbs model. Afterwards, a Bayesian decision was used to find the optimal configuration of 3-d blocks using a Monte Carlo sampler. This method was validated on multiple data sets. It was reported that the proposed approach not only provides very good results from a single DSM, but also works well on various data resolutions.

Tournaire *et al.* (2010) extracted the building footprints from digital elevation models with various resolutions ranging from 10-cm to 1-m ground sample distance. A satellite DEM having 50 cm GSD was obtained with a graph-cut optimization approach in a multi-view framework. The proposed method was based on stochastic geometry and marked point processes of rectangles.

The aim was to obtain a reliable object configuration described by a collection of rectangular building footprints. Even if buildings were more complicated than a simple rectangle, they could be described by a union of rectangles which was able to generate the complex structures. The proposed methodology was compared with an existing methodology based on the same mathematical framework. The current study obtained similarly good results with a high computational efficiency using a simplified energy function.

Karantzalos and Paragios (2010) introduced a novel inferential framework for the reconstruction of 3-d buildings from the fused optical image and digital elevation model data sets. The proposed framework significantly extended the previous 3-d extraction and reconstruction efforts by accounting for shadows, occlusions, and other unfavorable conditions. The segmentation task was carried out in optical images and digital elevation maps. The competing priors to determine their pose and 3-d geometry from the observed data were also allowed by the integrated approach. Furthermore, a grammar-based building representation was introduced to efficiently describe the space of the solutions. Apart from the new building models, other terrain classes were added or removed from the database. The results were found to be very promising and the quantitative evaluation demonstrated the potential of the approach.

In 2-d building footprint extraction, several general purpose supervised learning strategies on multi-spectral imagery have been applied. The general approach used employs purely spectral input vectors built by the set of intensity values in each spectral channel for each pixel in the image. Although these vectors provide a suitable fixed-dimensionality space, in which the conventional classifiers often work well, it is evident that spatial relationships such as texture, proximity, or shape can be very informative in feature extraction. This kind of extra information can be added to spectral information. However, there exists a combinatorial huge choice for these additional vector dimensions (Harvey *et al.*, 2002). To deal with this problem,

a hybrid evolutionary algorithm called GENIE (GENetic Image Exploitation) was developed by Perkins *et al.* (2000). The algorithm searches a space of image processing operations for a set that can produce suitable feature planes, and a more conventional classifier which uses those feature planes to output a final classification. The overall structure of GENIE is shown in Figure 2.1.

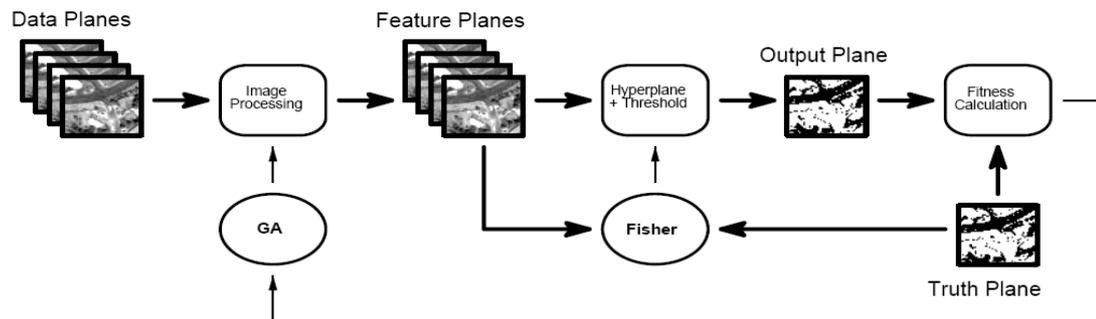


Figure 2.1. The overall structure of GENIE (Perkins *et al.*, 2000).

Initially, the training data was provided by marking up an image that shows both the locations of feature and the non-feature of interests. Then, the original data planes were transformed into a set of feature planes by a certain sequence of image processing operations (chromosomes). This was followed by a conventional supervised classification algorithm that was applied to feature planes in order to obtain the output image plane, which was an indicator for each pixel, whether that feature was there or not. Finally, a fitness value was calculated for each chromosome by comparing the output and the truth planes. The GENIE system was tested on various study areas for the automatic extraction of man-made features including buildings from high-resolution IKONOS images.

In a further study, Perkins *et al.* (2005) developed the system GENIE Pro. As in GENIE, this system was also a general purpose adaptive tool deriving automatic pixel classification algorithms for satellite and aerial imagery from training input. In particular, GENIE pro integrated spectral information and spatial cues such as texture, local morphology and large-scale shape

information, in a much more sophisticated way. The system architecture of GENIE Pro is shown Figure 2.2.

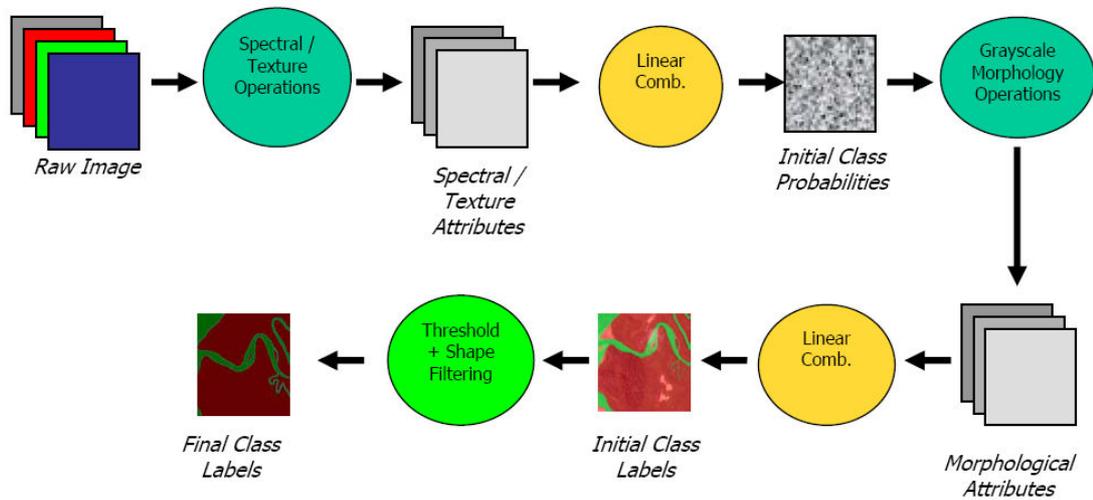


Figure 2.2. The processing pipeline of Genie Pro (Perkins *et al.*, 2005).

As in the system GENIE, the “Spectral/Texture” and “Grayscale Morphology” operations were the sets of attribute extractors. However, this time, each set was specialized for a specific task. For the extraction of spectral and texture attributes, the operators were designed to be simple and no complicated morphological operations were available. The aim was to highlight regions that might be the feature of interest but the result was likely to contain many false alarm areas. In the extraction of local morphological attributes, it was intended to further eliminate false alarms by employing morphological operations such as openings and closings with various different shaped structuring elements. The final stage of GENIE pro was composed of thresholding the grayscale image to obtain discrete labels, and performing an optional shape-filtering step.

The idea of fuzzy adaptive genetic algorithms is based on the adjustment of the selected control parameters or genetic operators during the evolution. These algorithms offer the most appropriate exploration and exploitation behavior to avoid premature convergence problem and improve the final

result (Herrera and Lozano 2003). The fuzzy logic based adaptive genetic algorithm model is shown Figure 2.3.

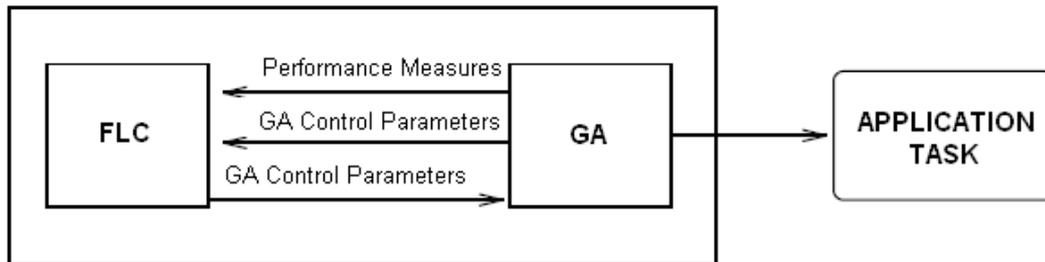


Figure 2.3. The fuzzy adaptive genetic algorithm model (Herrera and Lozano 2003).

In a study conducted by Liu *et al.* (2005), the use of a hybrid fuzzy genetic algorithm was presented. The crew grouping problem was solved using the fuzzy logic based controllers. The crossover and mutation probabilities were adjusted dynamically in order to improve the algorithm performance. When compared with the standard genetic algorithm, more satisfactory results were achieved with the fuzzy genetic approach.

2.3. Building Facade Texture Acquisition and Mapping

In this section, a review of studies focused on the acquisition and mapping of building facade textures is presented. First, the studies that include the capturing of the building facade textures from airborne sensors are summarized. It is shown that oblique and overlapping aerial images are commonly used. Next, the ground-based studies, which were conducted on building facade texture acquisition and mapping, are investigated. This section is further divided into three sub-parts which include the studies utilizing the terrestrial laser scanning technology, terrestrial image sequences that can be either a sequence of photos or a video data, and the studies conducted using the ground-level single images. It is reported that most widely used single images are those taken from panoramic, perspective, and

straight views. In the third part of this section, the investigation of the fundamental occlusion removal methods is provided briefly.

2.3.1. Airborne Studies Related to Building Facade Texture Acquisition and Mapping

The extraction of building facade textures from oblique airborne imagery is not commonly encountered in the literature due to its comparatively low resolution and poor depiction of facades. One of the related studies was conducted by Frueh *et al.* (2004), who employed the automatic texture selection for 3-d city models. First, the images were automatically registered by matching the 2-d image lines with projections of 3-d lines, derived from the city model. Next, the optimal image was selected by taking several factors into account, such as occlusion, image resolution, surface normal orientation, and coherence with neighboring triangles. Finally, for compact representation and efficient rendering, the utilized patches of textures were combined into a single texture atlas. They utilized 17 aerial images taken from a helicopter and the correct poses were found for all using the suitable parameters.

Lorenz and Döllner (2006) carried out facade texture mapping from overlapping high resolution monochrome aerial images. The proposed method was based on the rectified images that were extracted for each facade at a fixed spatial resolution. A quality map was also provided for encoding the actual effective spatial resolution of the projected aerial image. The rectified images were obtained by registering the multiple texture images to each others. As a result, the study was reported to be efficient in texturing the facades of large-scale city models.

Further, Wu *et al.* (2007) presented an approach for automatic retrieval of optimal texture from aerial video for the photo-realistic 3-d visualization of street environment. As an initial step, the texture distortions were eliminated by a fully-automatic orthorectification technique. Then, the normalized cross-

correlation operation was applied to ortho-rectified texture sub-windows in order to select a pair of texture having occlusion. Finally, the optimal texture was determined from two candidates based on an estimation method. The experimental results showed that the proposed approach increases eight times more efficiency and effectiveness than manual operation for urban texture reconstruction.

In a more recent study, the automated texture acquisition from oblique aerial images was performed by Wang *et al.* (2008). First, the image feature lines were extracted, which was followed by the matching of the extracted lines with the corresponding 3-d feature lines of object space from 3-d model. The interest image areas including building surfaces were then rectified with the refined external orientation (EO) parameters and the textures were obtained. The experimental results proved the correctness and robustness of the proposed approach.

2.3.2. Ground-based Studies Related to Building Facade Texture Acquisition and Mapping

In order to capture the building facade textures, various ground-based data sources were employed in the literature. The proposed techniques widely utilize the laser scanning data, terrestrial image sequences, and single images.

2.3.2.1. Building Facade Texture Acquisition and Mapping from Laser Scanning Data

A number of research projects have attempted to capture facade details from terrestrial laser scanning. One of these studies was proposed by Früh and Zakhor (2003), who automatically created textured 3-d city models using a mobile scanning system. They also managed to combine the images taken

from ground level and bird's eye perspective. To do that, a detailed model of the building facades were acquired by the ground based modeling process. The acquisition vehicle was localized using the DSM obtained from airborne laser scanning data. Then, the ground-based facades were registered to airborne model by means of Monte Carlo localization. Finally, to obtain a 3-d model the two models were merged with different resolutions. The study showed that the proposed approach results in visually acceptable models for downtown environments.

Böhm, (2008) demonstrated a new approach for facade detail extraction from range data in the case of incomplete texture information due to occlusions. The developed approach efficiently combined the coarse geometry of an existing building model (in the second level of detail – LOD2) with the detailed features from ground-based LIDAR data. To use a mapping for the integration of terrestrial LIDAR data, two-dimensional representation of the point cloud was derived. The point cloud that belongs to a particular facade was interpolated into a regular raster. The approach was found to be simple and efficient in enhancing prismatic building models using street-level LIDAR data.

Similarly, Carlberg *et al.* (2008) presented a general framework for facade surface reconstruction and segmentation using partially ordered 3-d point clouds composed of registered ground-based and airborne range and color data. The developed algorithm was capable of being applied to a large class of LIDAR data acquisition systems, in which ground-based data was obtained as a series of scan lines. Besides, the system was found to be efficient and scalable since it provided the reconstruction of the surfaces and the segmentation of the ground-based range data, simultaneously. Further, the merging of ground-based and airborne meshes, which exploits the locality of the ground-based mesh, was accomplished by a new algorithm.

In a more recent study conducted by Pu and Vosselman (2009), knowledge based reconstruction of building facade models from terrestrial laser

scanning data was presented. First, the important facade elements such as walls and roofs were distinguished as features. Then, knowledge about the size, position, orientation, and topology of the features was introduced to recognize them in a segmented laser point cloud. This was followed by the generation of an outline polygon of each feature using the least squares fitting, convex hull fitting or concave polygon fitting. Next, knowledge was employed to hypothesize the occluded parts from the directly extracted feature polygons. Finally, a polyhedron building model was used for the integration of extracted feature polygons with the hypothesized parts. The method was tested with two data sets containing various building shapes. The approach was found to be effective in automatic reconstruction of building models from terrestrial laser scanning data.

2.3.2.2. Building Facade Texture Acquisition and Mapping from Terrestrial Image Sequences

The vast majority of the studies have utilized the sequence of images in extracting the facade textures of the buildings. Of these, Faugeras *et al.* (1998) addressed the problem of the recovery of a realistic texture model of a scene from a sequence of ground-based images, without any prior knowledge about the parameters of the cameras or their motion. The approach was based on the correspondences between the images and the epipolar geometry. The geometry of the scene was reconstructed and the textures to be mapped on the scene polygons were extracted automatically from the images. Moreover, several images were combined through mosaicing them in order to remove the visual artifacts, such as pedestrians or trees from the textures. The whole system was evaluated and found to be efficient in production of scene models of high quality.

In a study conducted by Ortin and Remondino (2005), an occlusion-free image was generated for realistic texture mapping. The proposed method was used to determine the real appearance of facades by employing a

simple technique that relies on the homography between adjacent images. Thus, a new occlusion-free virtual texture was generated. The method could be directly applied to non-calibrated or even disoriented images acquired by an amateur camera. The test results were found to be satisfactory in recovery of the visual appearance of planar facades as well as complex 3-d objects.

Further, a highly automated facade texture generation and mapping system for 3-d building modeling was proposed by Tsai *et al.* (2007). The seamless and photo-realistic texture mosaics of building facades were generated from video sequences acquired with a digital camera and digital video camcorders. To provide the geometric alignment, the image frames were registered using an algorithm based on semi-automatically extracted interest points. Then, a polygon-based algorithm that utilizes alpha blending was applied to integrate colors and shadings over the overlapped regions of adjacent images. By doing this, continuous facade textures were produced in terms of geometric outlines and color domains. The occlusions of the generated texture mosaics were identified and mended using a series of morphological operations. Finally, the occlusion-free facade textures were mapped onto corresponding building facets. A test example showed that the resultant building model contained more complete and accurate texture features as well as a near-photorealistic appearance.

Mayer and Reznik (2007) performed the interpretation of building facades from uncalibrated wide-baseline image sequences. The proposed approach was composed of several novel features, such as the determination of facade planes by robust least square matching, learning of implicit shape models for objects, and the determination of windows by using Markov Chain Monte Carlo (MCMC) method, which employs an abstraction hierarchy based on mathematical morphology. The results were produced fully automatically, using only a few semantically meaningful thresholds and found to be fairly reasonable.

Korah and Rasmussen (2007) presented a technique for constructing “clean” texture map of a partially occluded building facade. A series of images taken from a moving camera were used as the main input stream. They applied a robust measure of spread to infer whether a particular mosaic pixel was occluded in a majority of views and proposed a novel spatio-temporal timeline-based inpainting algorithm. The algorithm utilized the appearance and motion cues to fill the texture map in highly occluded portions. Besides, the proposed method achieved to recognize foreground and background patches in a static imagery, as well. Satisfactory results were shown on building sequences.

In a more recent study conducted by Poullis and You (2009), the reconstruction of a photorealistic large-scale urban city model was presented. After the interactive reconstruction of complex linear and non-linear surfaces, the texturing problem was addressed. For the composition of photorealistic textures a rendering pipeline was proposed. The pipeline provided the recovery of missing or occluded texture portions in one image from another without doing any manual editing work. Apart from the images acquired from the ground-level, the aerial and satellite images were also employed for producing a set of view-independent seamless textures. The evaluation of the reconstructed 3d models could not be performed since there was no ground truth for comparison. Instead, model accuracy, realistic representation and level of detail, and scalability were considered. Their approach was found to be successful in creating the photorealistic large-scale virtual environments.

In an exceptional study conducted by Hoegner and Stilla (2009), 3-d building models were textured from the images recorded by infrared (IR) cameras. First, a relative orientation of the image sequences was generated. Then, the relative oriented image scene was matched. Next, the textures were extracted for each image and they were combined to create complete textures for the model surfaces. It was reported that the automated texturing of 3-d building models was achieved successfully.

Jang and Jung (2009) proposed a practical method for large scale building modeling with multiple image sequences. In their study, a single acquisition system, which is composed of a hand-held digital camera equipped with a GPS and a digital compass, was utilized. First, a scene was acquired by a pose camera. This step was used to annotate each image with navigation information acquired by a GPS and a digital compass. Then, an adjacency graph was built to determine the approximate spatial adjacency between each image sequences of roughly known poses. This was important with regard to organize the nodes into a set of sequences and access the nodes. Next, the camera positions were obtained by a standard "Structure from Motion" algorithm. Finally, the user-assisted modeling approach, which is based on a half-edge data structure, was employed. By using this method, suitable line segments were extracted for estimating the vanishing points and the corresponding corners.

In a very recent study, Tian *et al.* (2010) demonstrated an automatic knowledge-based method for the reconstruction of building facades from terrestrial video sequences. First, rules were applied to group the extracted features in a reasonable manner. Then, for each surface patch, a suitable outline and normal direction was specified. Using these patches, a hybrid model was employed to recover a building model from the extracted facade patches and hypothesized parts. The results showed that the proposed method correctly sets up topological relationships between the generated surface patches as well as the reasonable structure models in occluded areas.

In a study conducted by Kang *et al.* (2010), the automatic mosaicing of building facade textures was achieved from a sequence of monocular close-range images. The process was initiated by the computation of the camera parameters. The images were then rectified in order to eliminate the salient geometric distortions. Further, the relevant image segment was automatically retrieved using the detection range variance between the corresponding points for each of the facades. Afterwards, the entire building facade texture

was mosaiced for ortho-image generation. Finally, a dual post-processing step was employed to refine the mosaic image. The experimental results demonstrated that this method was widely acceptable in terms of automation level and applicability.

2.3.2.3. Building Facade Texture Acquisition and Mapping from Single Image

The utilization of single images has also been frequently met in the literature. One of the earliest studies was conducted by Song and Shan (2004). In a 3-d geospatial information system, wall textures were obtained from ground photography acquired by a digital camera and a 35mm roll film camera. Each wall was related to one individual image file and the wall texture was associated directly with the view plane without building the 3-d model. It was reported that the integration between the photorealistic visualization and 3-d GIS was successfully achieved.

Haala and Kada (2005) proposed a semi-automatic texture mapping procedure which uses panoramic scenes collected from a high-level system based on a rotating CCD line scanner. With this technology, large areas were able to be covered at high resolution and superb image quality. Besides, texture for a considerable number of buildings became available from a single scene. In addition to the reduced number of images to be processed, the use of panoramic scenes was able to minimize the changes in illumination since the image acquisitions at a different epoch was avoided.

In a study performed by Müller *et al.* (2007) the automatic derivation of 3-d models of high visual quality was carried out from rectified single facade images of arbitrary resolutions. After reconstructing the urban model, the procedural modeling of facades was performed using an algorithm developed. First, the facade structures such as floors, windows or doors were subdivided. Then, the subdivisions were segmented into smaller rectangles

using translational symmetry. Afterwards, the smaller rectangles with 3-d objects from a library of architectural elements were matched. Finally, various editing operations were performed by means of semantic facade interpretation along with the extraction of shape grammar rules. The tests were conducted on ground-based images. The comparisons showed how high resolution imagery can be visually enhanced with the proposed method.

Mohan and Murali (2007) presented a novel method for automatic modeling of planar surfaces with texture using the single view perspective images based on the edges and intensity representation. In their approach, the wire-frame building models were reconstructed. Next, perspective rectification was applied on every surface in the presence of symmetric objects. Afterwards, the wire-frame model was modified to have the true aspect ratio. Finally, the surfaces were rendered by piecewise texture mapping. Convincing photorealistic models were produced after rendering using texture mapping.

A novel approach was demonstrated by Laycock *et al.* (2007) for automatic generation and texturing of urban environment. The method was based on procedural texture generation, which permits the realization of a large variation in the building facade's appearance. The approach was efficient in terms of its memory usage and graphics hardware utilization. The procedural texture generation comprised the segmentation of a ground level image into three maps of wall, feature, and object. The wall map was used to refer to the wall material, such as flint or brick. The feature map was composed of a unique set of objects like windows or doors. On the other hand, the object lookup map was a template image representing the arrangement of the features on the facade. The OpenGL shading language was employed to combine these three maps during the rendering. An automatically generated and populated urban environment was preprocessed in less than ten minutes.

In a more recent study conducted by David (2008), an approach was proposed to detect building facades from the perspective view single images.

First, the image line segments were located. Then, the vanishing points of these segments were determined using the RANSAC robust estimation method. This was followed by the generation of local support for planar facades at different orientations. Next, the plane support points were clustered using an algorithm without having any knowledge about the number of clusters or their spatial proximity. Finally, the building facades were identified by fitting vanishing point-aligned quadrilaterals to the clustered support points. It was reported that the main contribution of this study was on the improved performance over existing approaches while placing no constraints on the facades.

Ripperda (2008) presented an approach for the extraction of building facade elements from ground-based images and range data. The reconstruction was a grammar based extraction approach which was guided by the reversible jump Markov Chain Monte Carlo (rjMCMC) method. This was due to the fact that the model was a structural description of the facade. Since the main interest in their study was the distribution of windows and doors, their widths and heights were also emphasized.

2.3.3. Summary of Texture Occlusion Removal Techniques

Apart from the occlusion removal techniques mentioned in the facade texture acquisition section, there have been several fundamental studies concentrated solely on texture occlusion removal. In one of these studies, the “image quilting” approach was developed by Efros and Freeman (2001), who presented a simple image-based method for generating a novel visual appearance, in which a new image was synthesized by stitching together the small patches of the existing images. The algorithm was composed of iteratively executed three steps. In the first step, the image was traced in raster scan order in steps of one block. Second, the input texture, which satisfies the overlap constraints within some error tolerance, was searched for every location. In the last step, the error surface between the newly

chosen block and the old blocks at the overlap region was computed. In addition, the minimum cost path along this surface was determined and the selected block was pasted onto the texture. More generally, the proposed algorithm was found to be fast and very simple in texture synthesis. Moreover, successful results were obtained for a wide range of textures.

Further, a region filling and object removal approach by an exemplar-based image inpainting was proposed by Criminisi *et al.* (2004). This study was proved to combine the advantages of two previous approaches that are texture synthesis and inpainting. The result was an image in which the selected object was replaced by a visually plausible background that mimics the appearance of the source region. For determining the fill order of the target region, an exemplar-based texture synthesis technique modulated by a unified scheme was employed. The fill priority was affected by a confidence value derived from the pixel values. The approach was also capable of propagating both linear structure and two-dimensional texture into the target region with a single simple algorithm. As a result, the approach was proved to be robust and effective.

In a different study conducted by Wilczkowiak *et al.* (2005), a hole filling algorithm, which can automatically locate candidate source patches and adhere to user specified constraints, was demonstrated. The source patches were either retrieved from elsewhere in the same image or from other images taken from different perspectives. The two significant contributions of this work were the automatic detection and adjustment of source patches, in which the macrostructure was compatible with the hole region. It was also reported that the proposed approach has a capability of handling macrostructure with an adjustable degree of automation.

In a recent study, a new image completion algorithm to work with the building facade textures was presented by Konushin and Vezhnevets (2007). Under the assumption of high periodicity of facade textures, the reconstruction of textures was carried out by an approach based on a cloning procedure. In

this procedure, the unknown (or occluded) area was filled with large patches as much as possible. In other words, a source patch was searched to cover the whole unknown area at once. When it was found, this patch was cloned into the hole. Otherwise, the hole was split in two, and the same procedure was repeated once more. The proposed algorithm was tested on more than 30 buildings of Moscow and Seoul. The results showed that the algorithm was able to keep the macrostructures of the buildings and deal with large resolution textures, as well.

CHAPTER 3

THE METHODOLOGY AND THE DEVELOPED SOFTWARE

In this chapter, first the methodology of the developed photorealistic 3-d building modeling framework is given. Then, the interface of the developed software **PhotoREalistic BUilding Modeling (PREBUM)** is described.

3.1. The Methodology

The methodology of the developed photorealistic 3-d building modeling from satellite and ground-level imagery is shown in Figure 3.1.

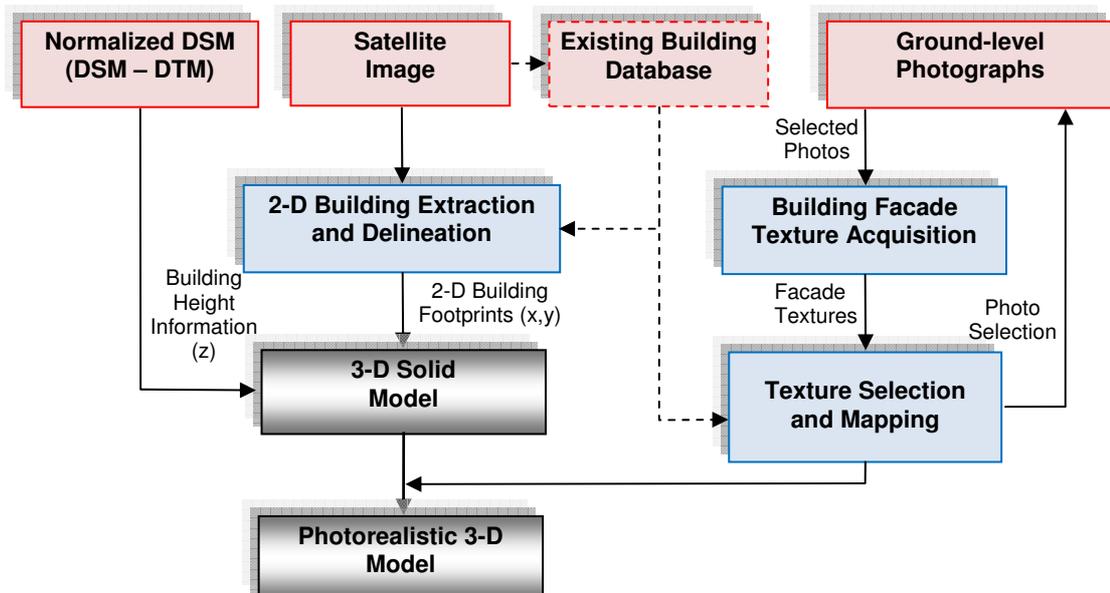


Figure 3.1. The flowchart of the developed photorealistic 3-d building modeling from satellite and ground-level imagery.

The normalized digital surface model (nDSM), satellite image, and the ground-level photographs are the main inputs of the system. Existing building database is used as the secondary input data. When preparing this data, the buildings, which have identical appearance in terms of size and texture, are grouped together. This grouping is done as a result of building analysis from the satellite image and also from the field work. These groups are then subdivided into polygons and called “building blocks”. Three main approaches were developed for generating 3-d solid model as well as the photorealistic 3-d model. These approaches are; (i) 2-d building extraction and delineation, (ii) building facade texture acquisition, and (iii) Texture selection and mapping.

The 3-d solid building models are generated using two different data sets. The first data set contains 2-d building footprints (x-y), which are detected from high resolution pan-sharpened satellite image by employing the method developed for the 2-d building extraction and delineation part of the framework. In brief, the method first extracts the building patches from the imagery using an adaptive fuzzy-genetic approach. This stage is originated from a genetic algorithm based image exploitation system “GENIE”, which was developed by Perkins *et al.*, (2000). However, the method developed at present study brings an important novelty that is an adaptive-fuzzy module that fine-tunes the genetic algorithm parameters aiming to improve the feature extraction performance. The technique integrates the concepts of the well known genetic algorithm, such as population, chromosome, gene, crossover, and mutation into fundamental image processing concepts. The population is defined as the set of chromosomes, which consists of a predetermined number of image processing operations (genes). The genes are composed of the basic image processing operations. The algorithm is initiated by selecting the training samples for the building and non-building areas from the imagery. Next, the image processing operations are performed on a chromosome-by-chromosome basis to obtain the specific attribute planes, which are then fed into Fisher Linear Discriminant (FLD) module that finds an optimal discriminating hyper plane between the building

and non-building features. For each chromosome, the fitness values are calculated by analyzing the detection and mis-detection rates. After that, the crossover and mutation operations are applied to arbitrary chromosome(s) in order to create a better population in the next generation by diversifying the current population. At the end of each generation cycle, the crossover and the mutation probabilities are adjusted by the adaptive-fuzzy module for the next generation. The evolutionary process is repeated until a satisfactory level of iteration is reached.

In the second stage of the proposed method, building delineation operation is carried out. First, the morphological image processing operations are performed in order to enhance the extracted building polygons. Then, the building block to be processed is selected for further processing. For the selected building block, the corners of the building patches that fall within the block are detected. To do that, first the best fitting ellipse to each building patch is determined. Then, the maximum-area rectangle that can be inscribed in the ellipse is determined. In this manner, the corners of the rectangle can be used as the corners of the building patches. The building boundaries are then delineated by simply connecting the corner points in correct order.

The second data set used in the generation of 3-d solid models is the normalized digital surface model (nDSM), which is computed by subtracting digital terrain model (DTM) from digital surface model (DSM). The nDSM used at present study was generated in a former study conducted by Koc San (2009) in the department. By using the nDSM, the height information (z) of the building models is fetched easily within a satisfactory level of error margin. The detailed description about nDSM generation is given in data description section, in Chapter 1. In addition to three dimensional information (x-y-z), the size, translation, and orientation parameters of the building patches are also taken into account in the generation of untextured building models.

To extract the facade textures to be mapped on a solid model, a texture selection approach was developed. For each building block, two photos belonging to any two adjacent facades of any building are selected from ground-level photographs by means of proximity and bearing. Then, the selected photos are processed using the developed “Building Facade Texture Acquisition” technique to generate the facade textures. In brief, the developed technique provides the automatic retrieval of the photorealistic textures from single ground-level building images. First, the facade image is extracted using the Watershed segmentation technique. To initiate the segmentation the marker pixels are seeded automatically both for the foreground (facade) and background (sky, pavement and neighboring buildings) regions and the segmentation is carried out repetitively until the most successful foreground segment is extracted. Next, the extracted facade image is geometrically rectified using a developed technique that operates in an automated way. The main steps of the automatic rectification procedure includes; the Hough transformation of the segmented image, identification of the vertical end points of the detected Hough lines, estimation of the trend-lines, and the projective transformation. The geometric rectification is followed by the automated texture cropping and occlusion removal, which are also carried out automatically. In removing the occlusions a developed image-matching based approach is used. The approach consists of such operations that mark the occluded region, search the marked region in the remaining parts of the facade image, select a non-occluded patch, and copy it into marked region. Once the occlusion-free facade textures are generated they are mapped on the corresponding facades of the solid model and thus the photorealistic 3-d model is obtained.

3.2. The Developed Software

The software entitled **PhotoREalistic BUilding Modeling** (PREBUM) was developed within the scope of the study. The motivation of the software development has arisen from the need of integrating the main components of

the overall methodology and increasing the level of automation. This brings some advantages, such as speed, easiness, and compactness. Due to its user-friendly design supported by a graphical user interface (GUI) PREBUM is easy to use. The main interface of PREBUM is illustrated in Figure 3.2.

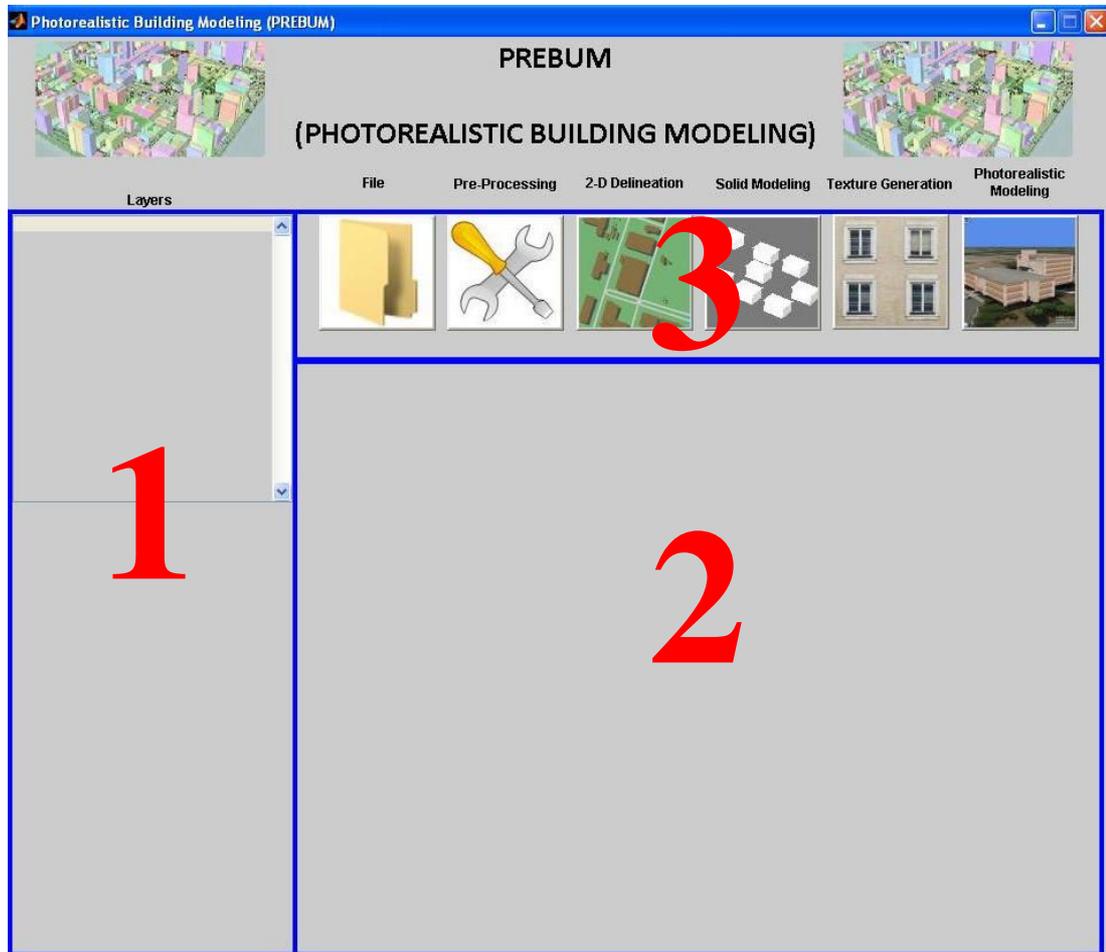


Figure 3.2. The interface of the developed software.

The interface is composed of three main panels fragmented by blue lines and numbered as 1, 2, and 3 in Figure 3.2. Panel -1 (Layers) is designed to keep track of input, output, and intermediate results of the analysis. The name of the output image files are simultaneously recorded on this panel and the user is allowed to scroll among them. The image files supported by the software include .jpg, .tif, .gif, .bmp, and .png. For the display of multispectral images, the desired band combination can be selected using the band-selection option before proceeding to next step. For this moment, the software is

limited to four bands, which are red (R), green (G), blue (B) and grayscale. Therefore, the user can select any combination of the bands Red ,Green, and Blue or a monochrome image. Figure 3.3 illustrates a couple of examples regarding the display of multispectral images.



Figure 3.3. (a) The display of a satellite image using the bands Red, Green, and Blue, and (b) the display of a facade image in Red band only.

Panel-2 is the view panel of the interface, in which the images are displayed. This panel is tightly connected with Panel-1 such that a selected file is visualized in this area immediately.

The final part of the GUI is the panel-3, which contains the major functions for data input and manipulation. Each function is performed by a popup menu item such as “File”, “Pre-Processing”, “2-D Delineation”, “Solid Modeling”, “Texture Generation” and “Photorealistic Modeling” (Figure 3.4).

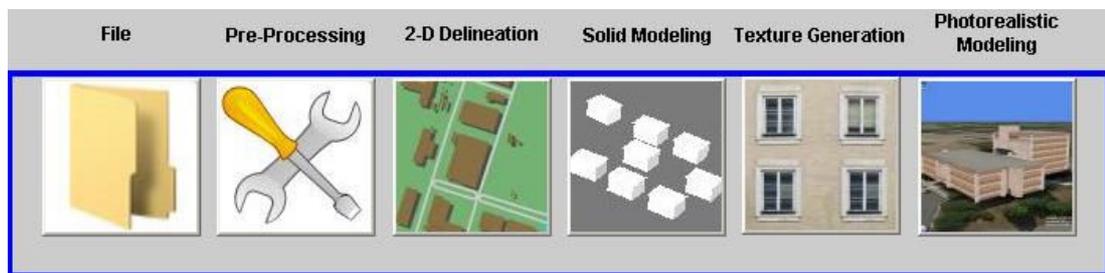


Figure 3.4. A closer view of panel-3.

“File” is the first menu item, which comprises four options: “Load Study Area”, “Delete All Layers”, “Delete Single Layer” and “Exit” (Figure 3.5).



Figure 3.5. The options of the “File” menu item.

The “Load Study” option provides the loading of the satellite image to layers panel. To do that, an open dialog box is initiated with the default file name as

“study_area.tif”. Besides, all other image file extensions mentioned above are also supported (Figure 3.6).

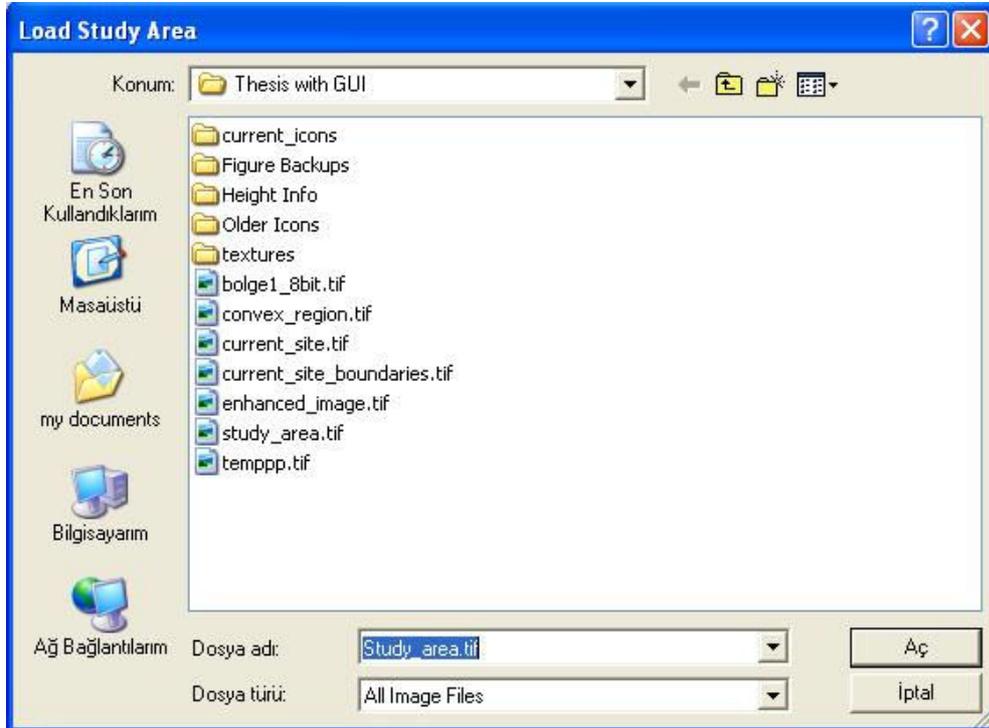


Figure 3.6. A snapshot from the option “Load Study Area”.

The second and third options are used to remove the files that are loaded previously to Panel-1. The only difference between the two options is that the former deletes the entire layers, while the latter deletes the selected layer only. The fourth option (Exit) terminates the program.

The second menu item “Pre-processing” consists of the fundamental image processing functions, which are used for enhancing the input imagery before proceeding to further stages. The input imagery can be a satellite image covering the study area or the ground-level photos of the building facades. The options contained within this menu item include “Adjust Intensity”, “Adjust Hue”, “Adjust Saturation”, “Histogram Equalization”, “Smooth”, and “Sharpen” (Figure 3.7).

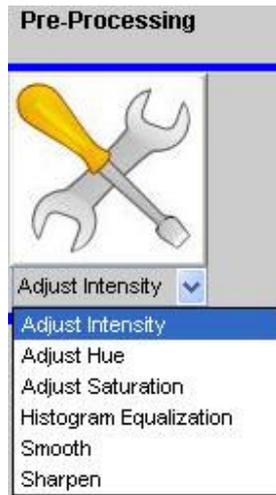


Figure 3.7. The “Pre-processing” menu item and the options it contains.

The first three options provide the adjustment of the input image in terms of hue, saturation and intensity (HSI). The features and the importance of HSI color space are given in Chapter 5. The interface “Adjust Intensity” is composed of “increase (+)”, “decrease (-)” and “reset” buttons along with an image preview window. The interfaces “Adjust Hue” and “Adjust “Saturation” are identical with the interface “Adjust Intensity”. Figure 3.8 shows the “Adjust Intensity” option of the “Pre-processing” menu item.



Figure 3.8. The option “Adjust Intensity” of the “Pre-processing” menu item.

The “Histogram Equalization” option of the “Pre-processing” menu item is employed to enhance the contrast of the image by means of a well-known contrast enhancement technique of histogram equalization. The fifth option, “Smooth” is used for eliminating the Gaussian-type noise from the imagery by smoothing it via a simple average filter. Finally, the last option “Sharpening” is carried out to sharpen the image by reducing the level of blurring.

The next menu item, “2-D Delineation”, includes five options that are “Set GA Parameters”, “Segmentation”, “Morphological Pre-Processing”, “Building Block Selection”, and “Delineation” (Figure 3.9).



Figure 3.9. The “2-D Delineation” menu item and the options it contains.

In this menu item, the building footprints are detected using a sequence of operations. The first option “Set GA Parameters” allows the initialization of the genetic algorithm parameters that are used in the segmentation of remote sensing imagery to detect the building patches. After segmenting the imagery, a pre-processing stage based on image morphology is carried out using the “Morphological Pre-Processing” option of the menu item “2-D Delineation”. This is followed by the selection of building blocks that contain buildings being in similar characteristics with respect to size and texture. Finally, the boundaries of the segmented patches are extracted using the option “Delineation”, which operates on an image geometry-based approach

developed at present study. The details of the options of the current menu item are emphasized in Chapter 4. The source code for the developed 2-d building extraction and delineation procedure is provided in Appendix A.

The menu item “Solid Modeling” is used in generating the untextured 3-d building models. It contains the options of “Input 3-D Height Information”, “Generate 3-D Models”, and “Show 3-D Solid Models” (Figure 3.10).

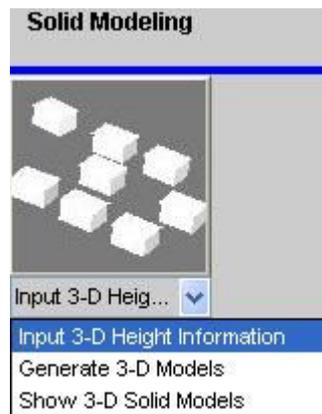


Figure 3.10. The “Solid Modeling” menu item and the options it contains.

In this menu item, the option “Input 3-D Height Information” provides data input for the building heights by overlaying the nDSM data with the detected building patches. For each building block, separate text files, which contain building height information belonging to each building contained within the block, are generated and then selected using this option. The option “Generate 3-D Models” integrates the height information with the previously determined 2-d coordinates. This process ends by generating the untextured 3-d models. The option “Show 3-D Solid Models” is used to visualize the building models using Cortona VRML viewer. A detailed description about the menu item “Solid Modeling” is given in Chapter 6. The source code for 3-d model generation is given in Appendix C.

The menu item “Texture Generation” is used for the generation of the building facade textures to be mapped on the solid model. The options contained within this item include “Load Facades and Set Parameters”,

“Texture Extraction”, “Rectification and Cropping”, and “Occlusion Removal” (Figure 3.11).



Figure 3.11. The “Texture Generation” menu item and the options it contains.

With the option “Load Facades and Set Parameters” the ground-level photos that are associated with the building blocks to be processed are loaded to the system. The parameters to be used in the texture extraction phase are also set using this option. The “Texture Extraction” option initiates the extraction of the facade textures from ground-level building photos using an approach that operates on iterative Watershed segmentation. The option “Rectification and Cropping” is used to perform the geometric rectification of the extracted facade textures and crop the rectified textures. The last option in this menu item is “Occlusion Removal”. It is used to remove the occlusions which may be present in the facade textures. A detailed description about this menu item is given in Chapters 5 and 6. The source code for the facade texture generation procedure is given in Appendix B.

The sixth and the last menu item of panel-3 is “Photorealistic Modeling”. The function of this menu item is twofold; the mapping of the facade textures and the visualization of the photorealistic model. The options contained within this menu item include “Map Textures” and “Show 3-D Photorealistic Models” (Figure 3.12). The first option “Map Textures” is used to map the previously generated facade textures on the solid model. Besides, it handles the roof

texturing with artificially produced tile patterns. In the second option “Show 3-D Photorealistic Models”, the visualization of the photorealistic model is carried out using the Cortona VRML viewer. A detailed description about this menu item is given in Chapter 6. The source code for the facade texture mapping is given in Appendix C.



Figure 3.12. The “Photorealistic Modeling” menu item and the options it contains.

In addition to the panels described above, the separate popup windows, warning dialog boxes, and wait bars are also employed in PREBUM to inform the user.

CHAPTER 4

2-D BUILDING EXTRACTION AND DELINEATION

In this chapter, a new technique developed for 2-D extraction and delineation of urban buildings from high resolution satellite imagery is described. In brief, the building regions are extracted using an adaptive fuzzy-genetic algorithm, while the footprints are detected through the morphological image processing operations. The overall methodology is given in the first section. Then, the steps of the extraction and delineation are examined thoroughly together with the accuracy assessment methods.

4.1. The Proposed Workflow

The steps of the proposed 2-D building extraction and delineation framework are given in Figure 4.1. The high resolution satellite image constitutes the input data. The 2-D building extraction is performed using an adaptive fuzzy-genetic approach, which combines the genetic algorithm concepts with the well-known image processing operations. The adaptive fuzzy part is used to improve the feature extraction performance by adjusting the genetic algorithm parameters. Next, the building delineation step is carried out to reveal the footprints of the extracted patches to be used in the generation of 3-D building models. This step comprises image morphology functions such as opening, hole-filling, and convex image generation along with the detection of the region corners of the binary images.

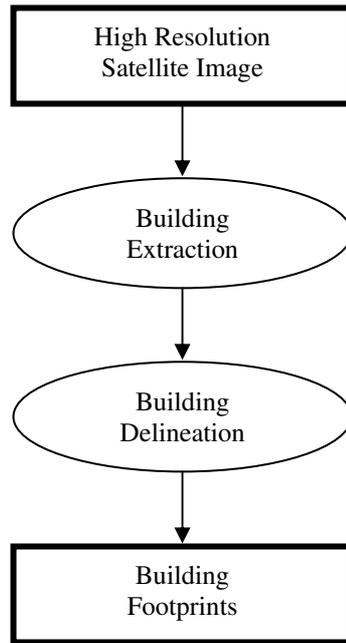


Figure 4.1. The general workflow of the 2-D building extraction and delineation approach.

4.2. Building Extraction

The processing steps of the proposed methodology for 2-D building extraction using the adaptive fuzzy-genetic approach are illustrated in Figure 4.2. First, the training and test regions are selected from the image both for the building and non-building classes. Next, the predetermined image processing operations are performed on the RGB image bands to obtain the spectral and texture attributes. These attributes are reduced into a single grayscale image band (the extracted building regions) by a Fisher Linear Discriminant Analysis. Then, the fitness values are computed by using the test samples. This is followed by performing several genetic algorithm operations in order to diversify the candidate solutions. In the last step, the genetic algorithm parameters are adjusted by an adaptive fuzzy logic controller to improve the performance of the methodology. This evolutionary process is repeated until a satisfactory level of accuracy is achieved.

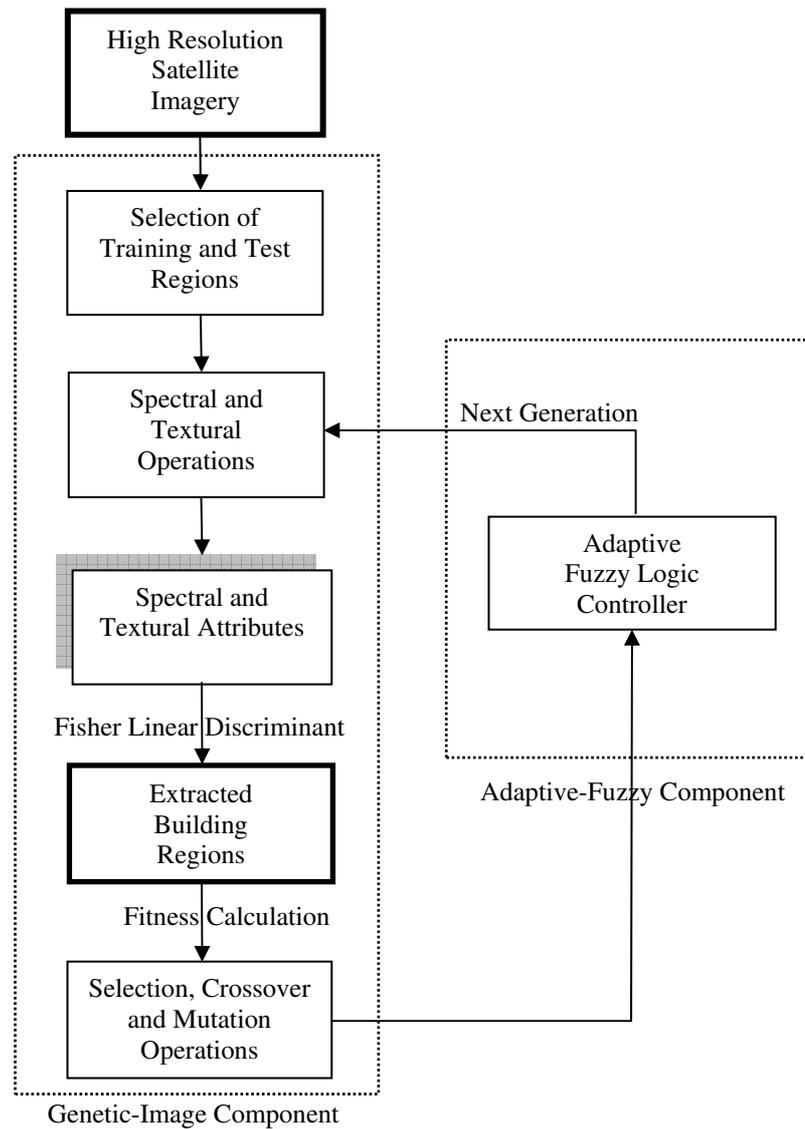


Figure 4.2. The proposed 2-D building extraction technique.

4.2.1. The Genetic-Image Component

Before describing the processing steps in detail, the fundamental image-based genetic algorithm (Perkins *et al.* 2000) basics are introduced first. In their design, the population of the genetic algorithm is generated from a predefined number of chromosomes, each of which can be seen as a candidate solution in extracting the building regions. The structure of a chromosome consists of a predetermined number of image processing

operations (genes). The genes are the well-known functions, such as basic mathematical, logical, thresholding operations, as well as the spectral and texture measures. The structure of a population, chromosomes, and the genes are illustrated in Figure 4.3.

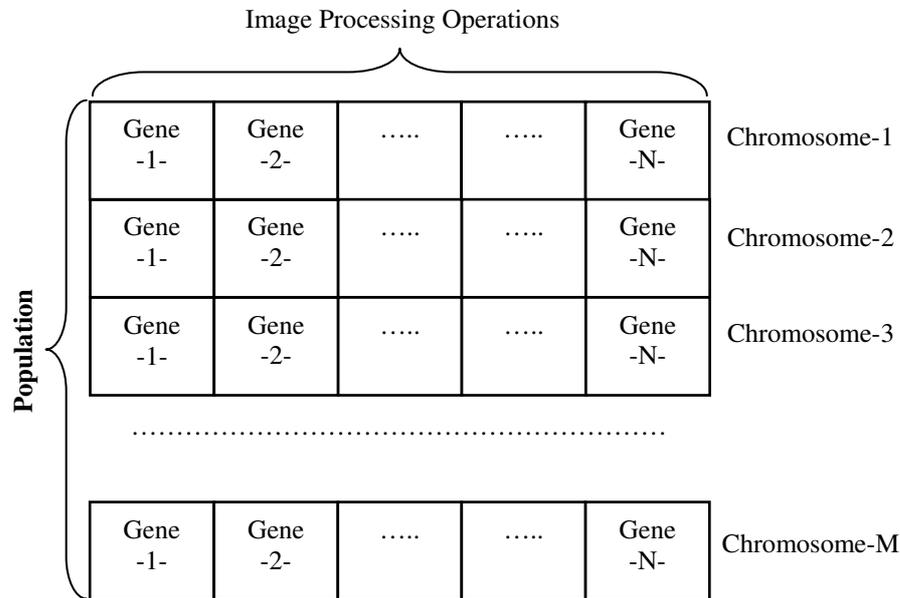


Figure 4.3. The structure of a population, which is composed of M chromosomes and N genes in each chromosome.

The first step is the selection of training and test samples from the satellite image. At present study, the aim was to discriminate buildings from the background. Therefore, two feature classes namely “building” and “non-building” were specified and the training samples were selected for each class. The sample size was determined as 50 equal size (10x10) square regions per class, which makes a total of 5000 pixels. The test samples were also collected to be used for assessing the accuracy of building extraction. Similar to training samples, 50 equal area test samples were marked for each of the “building” and “non-building” classes. The training and test samples were collected from different locations of the study area. The training samples were collected from the lower part of the study area, while the test samples were collected from the upper portion. The distribution of the test and training samples are illustrated in Figure 4.4. The red and green squares

denote the test samples for the building and non-building feature classes, while the blue and yellow squares correspond to training samples for the building and non-building classes, respectively.



Figure 4.4. The distribution of the training and test samples.

In the developed PREBUM software, the selection of the training and test samples is carried out using the “Segmentation” option of the “2-D Delineation” menu item. Before starting the selection, the user is given an option to either utilize the existing samples or to select the new ones before proceeding to further steps. The corresponding dialog box is shown in Figure 4.5.



Figure 4.5. The question dialog box which pops up before the training/test sample selection.

The next step is the initialization of the chromosomes with the image processing operations (genes). These operations are randomly selected from a gene pool. The complete list of the image processing functions, which are included in the gene pool, is shown in Table 4.1.

Table 4.1. The primitive image processing operations (the gene pool) (Harvey *et al.*, 2002).

Category	GENE ID	Gene Description	Input	
			# of Input Bands	# of Parameters
Basic Mathematical	1	Add Bands	2	0
	2	Add Scalar	1	1
	3	Subtract Bands	2	0
	4	Normalized Difference	2	0
	5	Multiply Bands	2	0
	6	Multiply by Scalar	1	1
	7	Negate Band	1	0
	8	Square Root	1	0
	9	Square	1	0
	10	Linear Scale	1	2
	11	Linear Combination	2	1
Logical	12	Minimum	2	0
	13	Maximum	2	0
	14	If Less Than Else	4	0
Thresholding	15	Clip High	1	1
	16	Clip Low	1	1
	17	Threshold	1	1
Texture	18	R5R5	1	0
	19	LAWB	1	0
	20	LAWD	1	0
	21	LAWF	1	0
	22	LAWH	1	0
Spectral	23	Distance Similarity	3	0
	24	Correlation Similarity	3	0
	25	Similarity Value	3	0

The first category in the gene pool is composed of basic mathematical operations. Gene#1 simply adds two bands, while Gene#2 adds a positive or

negative scalar parameter to a band. Gene#3 subtracts two bands and Gene#4 is similar to Gene#3 but divides the result by the sum of its two inputs. Gene#5 multiplies the pixel values of the bands, while Gene#6 scales the input band by a positive scalar. The genes #7, #8 and #9 perform the negation, square root, and square operations, respectively to a single input band. Gene#10 is similar to Gene#6 but takes an extra parameter, which is added on the scaled input. The last gene Gene#11 in this category outputs a linear combination of two inputs, in proportion specified by a parameter that takes a value between 0 and 1.

The second category comprises the fundamental logical operations. The genes #12 and #13 perform pixel-wise minimum and maximum for binary input. Gene#14 outputs its third input whenever the first input is less than its second input, and its fourth input elsewhere. The third category includes several basic thresholding operations. Falling in this category, Gene#15 truncates any pixel values above a value set by its parameter while Gene#16 does the reverse. In Gene#17 the values below its parameter are set to 0 (black), while the values above the parameter are set to 1 (white). The functions in the texture category apply Laws' texture energy measures to input bands. The fundamental (L3, E3, S3) and derived vectors (L5, E5, S5, W5, R5) are composed of 1-d convolution kernels, which are:

- $L3 = [1 \ 2 \ 1]$
- $E3 = [-1 \ 0 \ 1]$
- $S3 = [-1 \ 2 \ -1]$
- $L5 = [1 \ 4 \ 6 \ 4 \ 1]$
- $E5 = [-1 \ -2 \ 0 \ 2 \ 1]$
- $S5 = [-1 \ 0 \ 2 \ 0 \ -1]$
- $W5 = [-1 \ 2 \ 0 \ -2 \ 1]$
- $R5 = [1 \ -4 \ 6 \ -4 \ 1]$

where; the mnemonics stand for (L)evel, (E)dge, (S)pot, (W)ave and (R)ipple (Laws, 1980). In the current study, R5R5, LAWB, LAWD, LAWF and LAWH (Genes #18 - #22) are generated from the set of 1-d kernels given above, in which R5R5 corresponds to $R5^T \times R5$; LAWB, LAWD, LAWF and LAWH correspond to $S3^T \times L3$, $E3^T \times E3$, $L3^T \times S3$ and $S3^T \times S3$, respectively. In the

last category, the spectral similarity measurement within the input bands is provided by the distance and correlation similarities along with the similarity value.

All the chromosomes in the population have the same fixed number of genes. The optimum numbers for the chromosomes and genes will be discussed in section 4.2.3. An example chromosome with five genes can be illustrated as follows: [**3 7 17 7 21**], where the numbers denote the gene identification numbers. The image processing operations “Subtraction – Gene#3”, “Negation – Gene#7”, “Thresholding – Gene#17”, “Negation – Gene#7” and “Texture (LAWF) – Gene#21” are performed on randomly selected input and output bands. The input bands are the **R** (red), **G** (green), and **B** (blue) bands of the satellite image, while the output bands are the empty temporary bands. A temporary output band can be used as an input band after it is initialized by an operator. In other words, the temporary band should be non-empty. At present study, four temporary bands were determined and they were labeled “**temp1**”, “**temp2**”, “**temp3**”, and “**temp4**”.

According to our hypothetical chromosome described above, an example scenario works as follows: For Gene#3, let’s assume that the algorithm selects two input bands (**R**, **G**) and one output (**temp3**). The result of subtraction (**R - G**) is written to “**temp3**”. From now on, the band “**temp3**” can also be used as an input band. For the next gene (Gene#7), single input and output bands are selected, which are assumed to be “**G**” and “**temp1**”, respectively. The result of negating the green band is written to “**temp1**”. After that, for Gene#17, a single input band (**temp3**) and the output band (**temp2**) are selected together with a scalar parameter, which can be selected randomly between 0 and 255 for an 8-bit image. In the present example, the scalar parameter was defined as 135. Therefore, the pixel values of the band “**temp3**” staying above 135 are set to 255 and those staying below are set to 0. Then, the result is written to band “**temp2**”. Next, Gene#7 is used again to negate an input band (**B**) and the resultant image is written to an output band (**temp4**). In this case “**temp4**” is the solely

candidate output band since it is the only remaining empty band. Finally, for Gene#21, an input band (**temp1**) is selected, the LAWF texture mask is applied to this band, and the output is written to selected band (**temp4**).

At the end, all the temporary output bands are utilized that represent the spectral and textural attributes of a chromosome. Since the final classification produces the output of a single binary band (the extracted building regions), the dimension of the temporary output bands are reduced to one. This is carried out by means of a Fisher Linear Discriminant (FLD), which is a conventional classification algorithm (Figure 4.6).

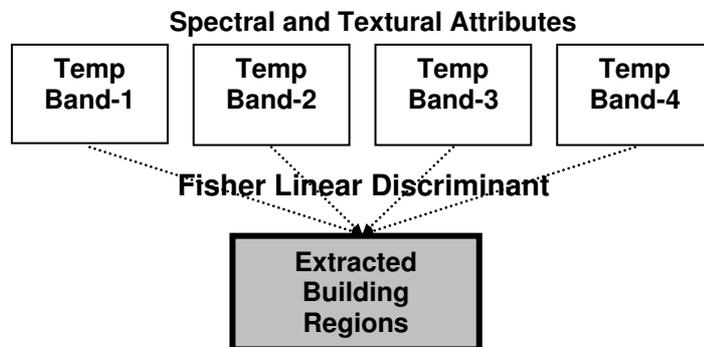


Figure 4.6. The dimension reduction procedure using Fisher Linear Discriminant.

The dimension reduction method provides a linear combination of the temporary output bands that maximize the mean separation between true (building) and false (non-building) pixels, normalized by the total variance in the projection defined by the linear combination. The result of the discriminant-finding phase is a gray-scale image, which is then reduced to a binary image by finding the threshold value that maximizes the "fitness". The dimension reduction phase is exemplified in Figure 4.7 (Joo, 2003).

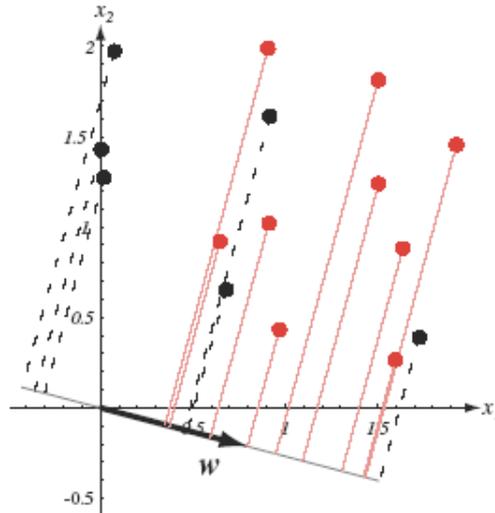


Figure 4.7. The optimum direction ' \mathbf{w} ' in discriminating the points belonging to two different classes (red and black).

In a projection onto one direction, \mathbf{w} (two class problem), the samples are \mathbf{n} d-dimensional vectors $\mathbf{x}_1 \dots \mathbf{x}_n$, which consist of two subsets D_1 and D_2 . The projected samples are computed by the below equation that consists of two subsets Y_1 and Y_2 .

$$y = \mathbf{w}^t \mathbf{x} \quad (\text{Equation - 1})$$

The criterion is to maximize the Fisher Linear Discriminant $J(\mathbf{w})$:

$$J(\mathbf{w}) = \frac{\mathbf{w}^t \mathbf{S}_B \mathbf{w}}{\mathbf{w}^t \mathbf{S}_w \mathbf{w}} \quad (\text{Equation - 2})$$

where, $\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2) \cdot (\mathbf{m}_1 - \mathbf{m}_2)^t$ is the between scatter matrix ($\mathbf{m}_i = \text{mean of } \mathbf{x} \in D_i$) and $\mathbf{S}_w = \mathbf{S}_1 + \mathbf{S}_2$ is the within scatter matrix, where,

$$\mathbf{S}_i = \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t \quad (\text{Equation - 3})$$

The optimal line direction, \mathbf{w} , can be computed as follows:

$$w = S_w^{-1}(m_1 - m_2) \quad (\text{Equation – 4})$$

Following the extraction of building regions in binary form, the next step is to evaluate the fitness of the candidate solution (chromosome). The details of fitness computation are discussed in section 4.2.4. After calculating the fitness values for all chromosomes in the population, the chromosomes are ranked from lowest to highest fitness. Then, only the best chromosomes are selected and the rest are discarded. The selection rate X_R is the fraction of the total population (N_{POP}) that survives for the next generation. The number of chromosomes to be kept (N_{KEPT}) is computed as follows:

$$N_{KEPT} = N_{POP} \times X_R \quad (\text{Equation – 5})$$

Of the total population of the chromosomes in a generation (N_{POP}), only the top N_{KEPT} are kept for mating, and the bottom ($N_{POP}-N_{KEPT}$) are discarded to allocate room for the new offsprings. Next, two chromosomes are selected from N_{KEPT} chromosomes to produce two new offsprings. The chromosomes are selected using a random pairing technique, which utilizes a uniform random number generator. The chromosome with the highest fitness value, which is named as “elite chromosome,” is excluded from this process to preserve the success rate for the next generations.

After selecting the parent chromosomes, the mating procedure is performed. Mating can be defined as the creation of one or more offspring from the selected parents. The most common forms of mating involve; the production of two offspring by two parents (crossover) and a single offspring by one parent (mutation). These operations are aimed to create a better population in the next generation by producing the altered offspring. The probabilities of the parent chromosomes to be involved in the crossover and mutation operations are set to P_C and P_m , respectively. In this study, the “single point” crossover operation is used. In this operation, a crossover point is randomly selected between the first and the last genes of the parents’ chromosomes. First, parent-1 copies its genes staying on the left of the crossover point to

offspring-1. Similarly, parent-2 copies its genes staying on the left of the same crossover point to offspring-2. Then, the genes staying on the right of the crossover point of parent-1 are moved to offspring-2 and parent-2 passes its genes to offspring-1 in the same manner (Figure 4.8).

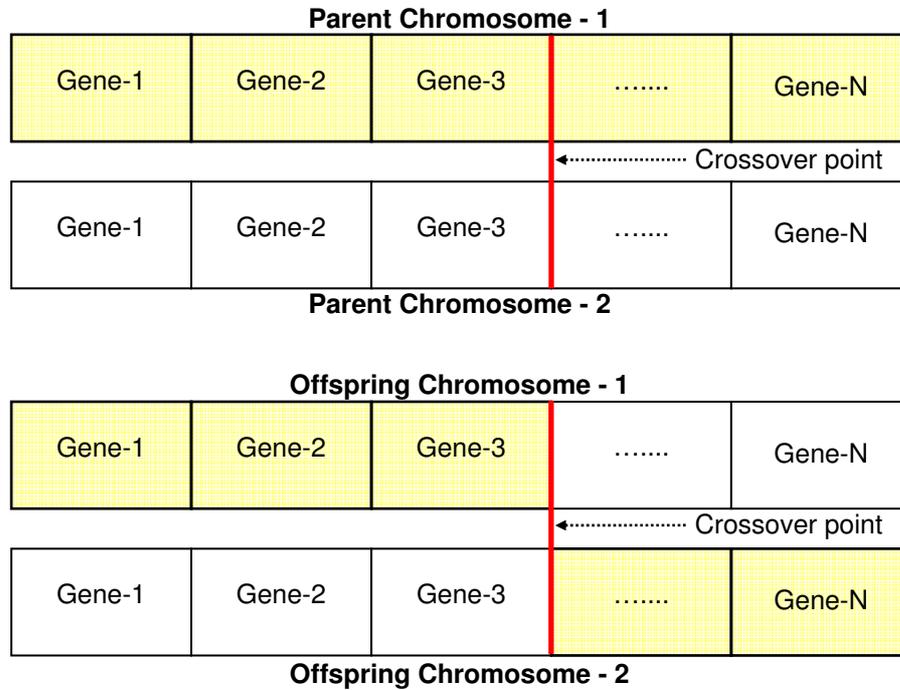


Figure 4.8. An example for the crossover operation and the generated offsprings.

Mutation is the second way to diversify the population. As in crossover, a single point mutation procedure is employed at present study. The gene to be mutated is randomly selected from the parent chromosome and exchanged by a gene arbitrarily selected from the gene pool (Figure 4.9).

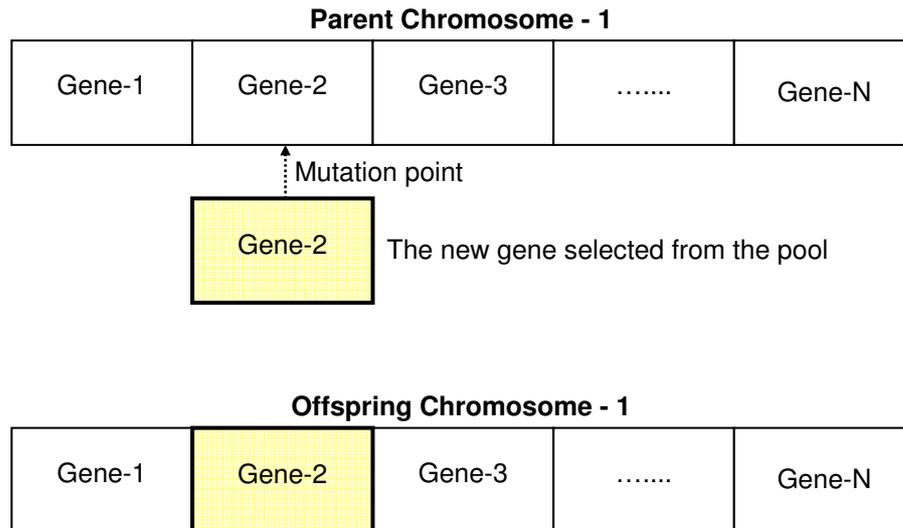


Figure 4.9. An example for the mutation operation and the produced offspring.

At the end of the crossover and mutation operations, the parents are expected to produce a total of $N_{POP} - N_{KEPT}$ offspring in order to keep the chromosome population as N_{POP} . To do that, the selection and mating procedures are repeated until the required number of offsprings is produced.

4.2.2. Adaptive-Fuzzy Component

Before proceeding to next generation of the genetic image component, an adaptive-fuzzy logic controller step is employed. The performance of the genetic algorithm is quite sensitive to control parameters. It is possible to destroy a well performing chromosome when the crossover probability is high. On the other hand, a low crossover probability may prevent obtaining better individuals and does not guarantee faster convergence. High mutation may cause too much diversity and takes longer time to reach the optimal solution, while low mutation tends to miss some near-optimal points. Thus, the use of fuzzy logic controllers to adapt genetic algorithm parameters is an important issue to improve the performance of the genetic algorithm.

The genetic algorithm performance measures, such as the average and maximum fitness values and the control parameters (crossover and mutation probabilities) are fed into the adaptive-fuzzy component. The controller sends back the adjusted parameters to be used in the next generation of the genetic algorithm cycle. The parameter flow between the two components is illustrated in Figure 4.10.

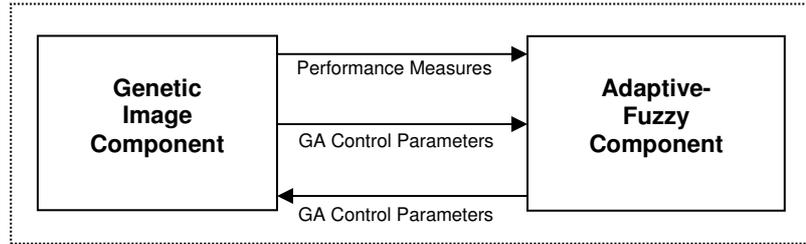


Figure 4.10. The parameter flow between the genetic image and the adaptive-fuzzy components (Herrera and Lozano, 2003).

The idea behind the adaptive-fuzzy component approach is as follows: The crossover and mutation probabilities (P_c and P_m) should increase if it consistently produces a better offspring. However, P_c should decrease and P_m should increase when $f_{ave}(k)$ (average fitness in k^{th} generation) approaches to $f_{max}(k)$ (maximum fitness in k^{th} generation) or $f_{ave}(k-1)$ approaches to $f_{ave}(k)$. This scheme is based on the fact that it encourages the well-performing genes to produce more offspring and reducing the chance for poorly performing genes to destroy the potential chromosomes during the crossover and mutation processes. In a study conducted by Liu *et al.*, (2005), two parameters (e_1 and e_2) were introduced to define the fuzzy rules for crossover and mutation operations (Equations 6 and 7).

$$e_1 = \frac{f_{max}(k) - f_{ave}(k)}{f_{max}(k)} \quad \text{(Equation – 6)}$$

$$e_2 = \frac{f_{ave}(k) - f_{ave}(k-1)}{f_{max}(k)} \quad \text{(Equation – 7)}$$

Using these parameters, the fuzzy rules are identified in order to describe the relation between the inputs e_1 , e_2 , and the output (the step size of the crossover or mutation probabilities) shown in Tables 4.2 and 4.3.

Table 4.2. Fuzzy rules for the crossover operation.

CROSSOVER ($\Delta P_c(k)$)	e_2				
	NL	NS	ZE	PS	PL
e₁	NL	NS	ZE	PS	PL
PL	<i>NS</i>	<i>ZE</i>	<i>NS</i>	<i>PS</i>	<i>PL</i>
PS	<i>ZE</i>	<i>ZE</i>	<i>NL</i>	<i>ZE</i>	<i>ZE</i>
ZE	<i>NS</i>	<i>NL</i>	<i>NL</i>	<i>NL</i>	<i>NL</i>

Table 4.3. Fuzzy rules for the mutation operation.

MUTATION ($\Delta P_m(k)$)	e_2				
	NL	NS	ZE	PS	PL
e₁	NL	NS	ZE	PS	PL
PL	<i>PS*</i>	<i>ZE*</i>	<i>PS*</i>	<i>NS*</i>	<i>NL*</i>
PS	<i>ZE*</i>	<i>ZE*</i>	<i>PL*</i>	<i>ZE*</i>	<i>NS*</i>
ZE	<i>PS*</i>	<i>PL*</i>	<i>PL*</i>	<i>PL*</i>	<i>PS*</i>

In these tables, the abbreviations NL, NS, ZE, PS and PL stand for “Negative Large”, “Negative Small”, “Zero”, “Positive Small” and “Positive Large”, respectively. The inputs of the mutation controller (e_1 and e_2) are same as the inputs of the crossover controller. However, the output values in table 4.3 that are illustrated by an asterisk (*) are scaled by 10% compared with the original output values given in table 4.2 (i.e. $PS^* = PS / 10$). The output values specify the step sizes of $\Delta P_c(k)$ and $\Delta P_m(k)$ for the crossover and mutation probabilities, respectively. By using the membership functions shown in Figure 4.11, the defuzzification process is performed by means of the centroid approach. In this technique, the fuzzy set membership function has the graph of a triangle as in the present case. If this triangle is to be cut in a straight horizontal line somewhere between the top and the bottom, and the top portion are to be removed, the remaining shape looks like a trapezoid. In the initial step of defuzzification, the parts of the graph are chopped off to form trapezoids. All of these trapezoids are then superimposed one another, forming a single geometric shape. Then, the

centroid of this shape is calculated and used as the defuzzified value. If the shape has a plate of equal density, the centroid is the point along the horizontal axis about which this shape would balance (Liu *et al.*, 2005).

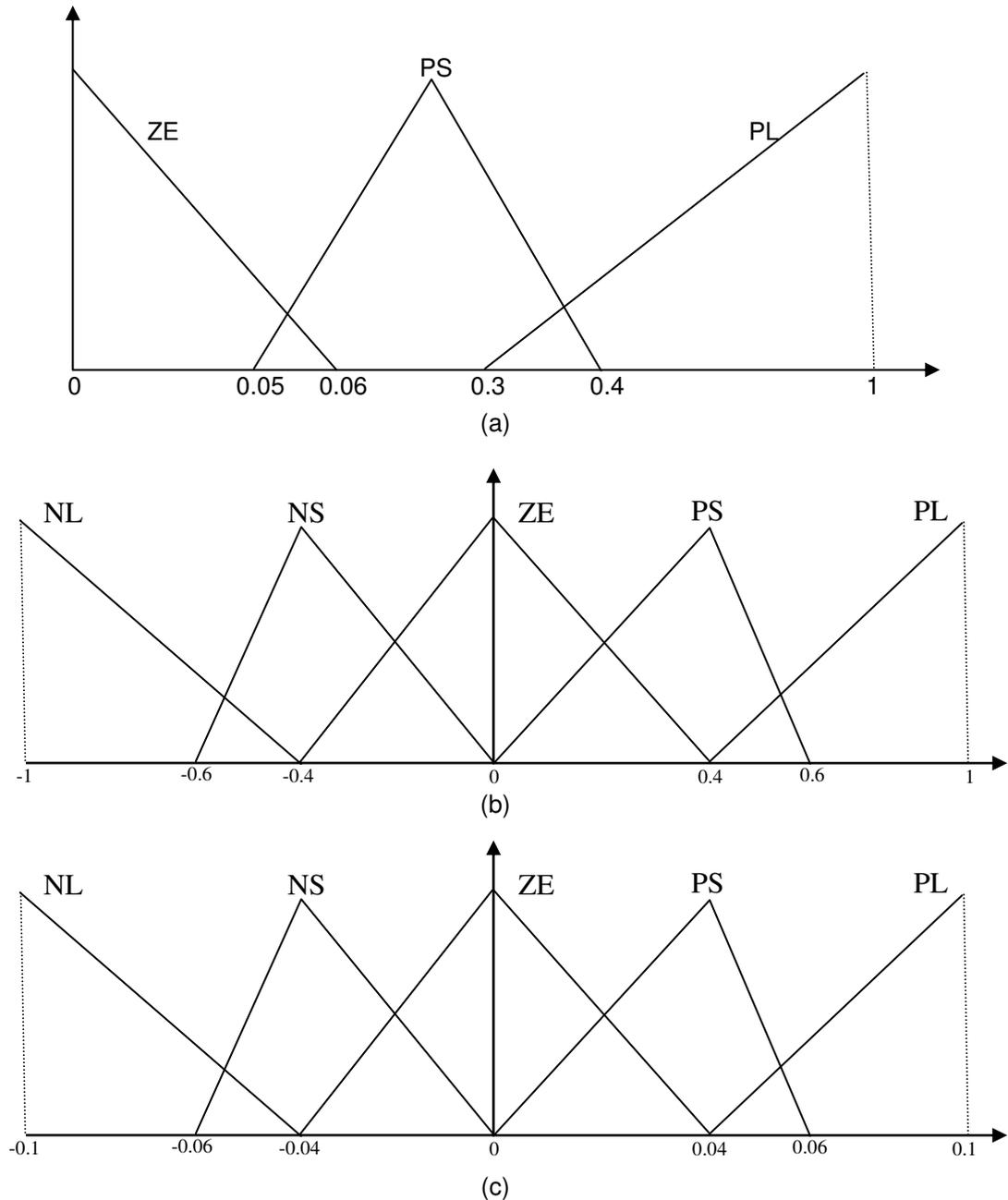


Figure 4.11. The membership functions for (a) e_1 , (b) e_2 and (c) $\Delta P_m(k)$.

By means of defuzzification process, the control parameters of the genetic algorithm are modified using the computed crisp values $\Delta P_c(k)$ and $\Delta P_m(k)$ (Equations 8 and 9)

$$P_c(k) = P_c(k-1) + \Delta P_c(k) \quad (\text{Equation – 8})$$

$$P_m(k) = P_m(k-1) + \Delta P_m(k) \quad (\text{Equation – 9})$$

After determining the new probabilities for crossover and mutation, the next generation is initiated with a renewed population. In the literature of genetic algorithm, the number of generations that evolve depends on whether an acceptable solution is reached or a set of iterations is exceeded. After a while, all the chromosomes and their fitness values would become the same. At this point, the algorithm should be stopped. In the present study, the genetic algorithm was stopped after the predetermined number of generations is reached.

The execution of the adaptive fuzzy-genetic approach in PREBUM is performed by choosing the “Segmentation” option under the “2-D Delineation” menu item. The progress of segmentation can be monitored by a wait bar. The segmentation result is stored in a file named “extracted_buildings.bmp” and is recorded on the “Layers” panel. The screenshots of the wait bar during segmentation and the “Layers” panel after segmentation are presented in Figure 4.12.

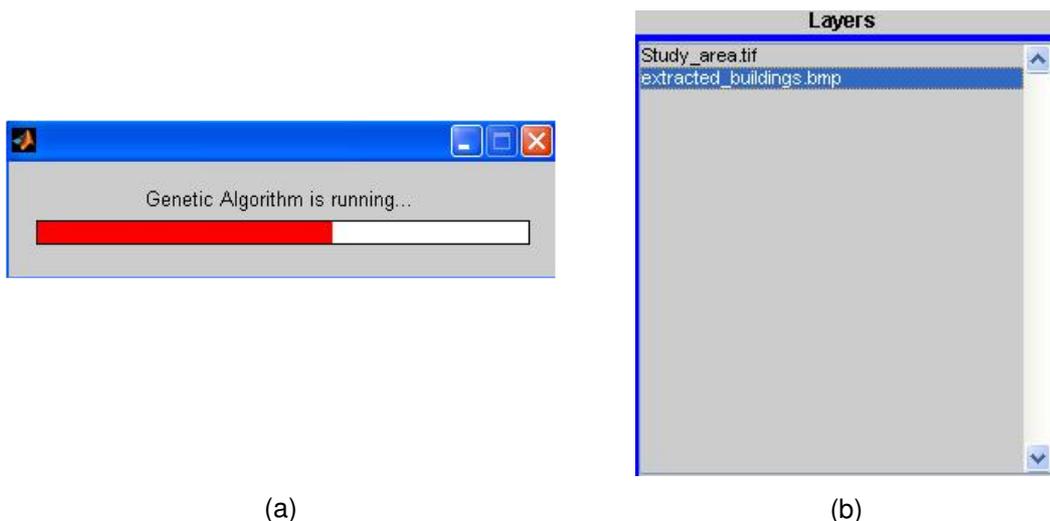


Figure 4.12. The screenshots of (a) the wait bar and (b) the “Layers” panel.

4.2.3. Genetic Algorithm Parameter Analysis

In the extraction of the building regions using the genetic algorithm, a number of parameters that include the training and test sample sizes, the selection rate (X_R), the number of generations, the population size (number of chromosomes), the chromosome size (number of genes), and the probabilities of crossover (P_c) and mutation (P_m) are used. For these parameters, the optimum values were determined.

To decide on a selection rate is somewhat arbitrary. Letting only a few chromosomes survive may limit the available genes while keeping too many chromosomes may result in a bad performance. As was the case in a study conducted by Haupt and Haupt (2004), X_R was kept to **50%** level in the natural selection process. The number of generations was kept **20**, which was found to be optimum for obtaining a barely changing value of the maximum fitness after performing several tests. The initial crossover and mutation probabilities were set to **0.8** and **0.2**, in parallel with the literature (Liu *et al.*, 2005; Haupt and Haupt, 2004; Perkins *et al.*, 2000).

To determine the optimum values for the population, the chromosome and training / test sample sizes, an accuracy test was carried out. To do that, the **average** and the **maximum fitness values** were computed under the varying parameters of training / test sample sizes, population size, and the chromosome size. For the training / test sample size, an initial step size of 5 was selected together with the minimum and maximum values of 5 and 50. Similarly, for the population and chromosome sizes, the step size, the minimum value, and the maximum value were set to 5, 5, and 30, respectively (Table 4.4). In this manner, a total number of 360 individual tests (algorithm runs) were performed, in which ten combinations for training / test sample sizes and six for each of the population and chromosome sizes ($10 * 6 * 6 = 360$).

Table 4.4. The selected step sizes and the minimum/maximum values for determining the optimum population, chromosome, and training/test sample sizes.

Parameters	Step Size	Minimum Value	Maximum Value
Training / Test Sample Size (number of regions)	5	5	50
Population Size (P) (number of chromosomes)	5	5	30
Chromosome Size (C) (number of genes)	5	5	30

After performing the tests, the highest peak value of **18** was calculated for the population size of **30** and the chromosome size of **5** (Table 4.5). In table 4.5, these values are highlighted in blue color. Finally, the optimum size for training and test samples was chosen to be **50** since among the other samples sizes the highest “maximum fitness” and “average fitness” values were reached using this value.

Table 4.5. The results of 360 individual runs, yielding the optimum population and chromosome sizes.

(Population,Chromosome) Pair	Training / Test Sample Sizes										TOTAL PEAKS
	5	10	15	20	25	30	35	40	45	50	
(p,c) = (5,5)	2	0	0	0	0	0	0	0	0	0	2
(p,c) = (5,10)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (5,15)	0	1	1	2	1	1	2	2	2	2	14
(p,c) = (5,20)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (5,25)	0	1	1	0	1	1	0	0	0	0	4
(p,c) = (5,30)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (10,5)	2	0	0	0	0	0	0	0	0	0	2
(p,c) = (10,10)	0	2	2	1	1	1	2	1	1	1	12
(p,c) = (10,15)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (10,20)	0	0	0	0	0	0	0	1	0	0	1
(p,c) = (10,25)	0	0	0	1	1	1	0	0	1	0	4
(p,c) = (10,30)	0	0	0	0	0	0	0	0	0	1	1
(p,c) = (15,5)	1	0	0	2	1	2	2	2	2	2	14
(p,c) = (15,10)	0	1	1	0	0	0	0	0	0	0	2
(p,c) = (15,15)	0	1	0	0	1	0	0	0	0	0	2
(p,c) = (15,20)	0	0	1	0	0	0	0	0	0	0	1
(p,c) = (15,25)	1	0	0	0	0	0	0	0	0	0	1
(p,c) = (15,30)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (20,5)	1	1	0	1	1	1	1	2	2	2	12
(p,c) = (20,10)	0	0	1	1	1	1	1	0	0	0	5
(p,c) = (20,15)	1	0	1	0	0	0	0	0	0	0	2
(p,c) = (20,20)	0	1	0	0	0	0	0	0	0	0	1
(p,c) = (20,25)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (20,30)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (25,5)	1	2	2	2	1	1	2	2	2	2	17
(p,c) = (25,10)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (25,15)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (25,20)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (25,25)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (25,30)	1	0	0	0	1	1	0	0	0	0	3
(p,c) = (30,5)	1	2	2	2	2	2	2	1	2	2	18
(p,c) = (30,10)	1	0	0	0	0	0	0	0	0	0	1
(p,c) = (30,15)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (30,20)	0	0	0	0	0	0	0	1	0	0	1
(p,c) = (30,25)	0	0	0	0	0	0	0	0	0	0	0
(p,c) = (30,30)	0	0	0	0	0	0	0	0	0	0	0

To clarify the procedure in determining the peak values, the following example is given using the fixed parameters of training / test sample sizes of 15 and the population size of 5. Further, the chromosome size is altered between 5 and 30 with a step size of 5 and the following fitness values are computed (Table 4.6).

Table 4.6. An example to clarify the peak values given in Table 4.5.

Training/Test Sample Size = 15, Population Size = 5	
<p><u>CASE 1:</u> Chromosome Size: 5 Maximum Fitness Value: 87.58 Average Fitness Value: 72.67</p>	<p><u>CASE 4:</u> Chromosome Size: 20 Maximum Fitness Value: 87.16 Average Fitness Value: 67.30</p>
<p><u>CASE 2:</u> Chromosome Size: 10 Maximum Fitness Value: 85.79 Average Fitness Value: 71.75</p>	<p><u>CASE 5:</u> Chromosome Size: 25 Maximum Fitness Value: 91.24 Average Fitness Value: 68.18</p>
<p><u>CASE 3:</u> Chromosome Size: 15 Maximum Fitness Value: 87.80 Average Fitness Value: 81.81</p>	<p><u>CASE 6:</u> Chromosome Size: 30 Maximum Fitness Value: 66.34 Average Fitness Value: 61.61</p>

Of the 6 different cases, the highest maximum fitness value of **91.24** was calculated for the chromosome size of 25 (case 5). Similarly, the highest average fitness value of **81.81** was computed for the Case 3, in which the chromosome size was 15. In Table 4.5, the corresponding cells are marked as **1** and highlighted in yellow color. The remaining peak values were determined in a similar way under different values of population, chromosome, and training / test sample sizes.

In the developed PREBUM software, the parameters of the genetic algorithm can be set by initiating the “Set GA Parameters” option under the “2-D Delineation” menu item. To retrieve the optimum parameters, the “Load Default Parameters” push button is clicked and the optimum values of the parameters are set to 50 for the training/test regions, to 20 for the number of generations, to 30 for the number of chromosomes, to 5 for the number of genes, to 0.8 for the crossover rate, and to 0.2 for the mutation rate. Alternatively, in each popup menu item these parameters can be set arbitrarily by selecting the predefined values. The “Apply” button is used to finalize the values by disabling the selection. The “Close” button closes the selection window. Figure 4.13 illustrates two different screenshots for the parameter selection of the genetic algorithm. In the figure, the left window (a)

illustrates the optimum parameters, while the right window (b) shows the arbitrary selection.

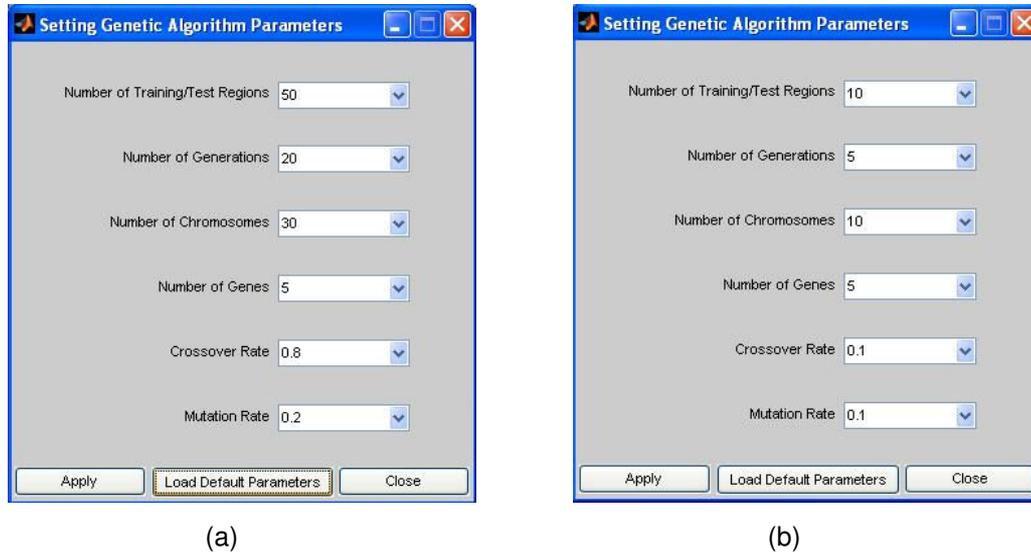


Figure 4.13. The parameters of the genetic algorithm with (a) the optimum values and (b) the arbitrary values.

4.2.4. The Accuracy Assessment of Building Extraction

The assessment of building extraction was carried out by the calculation of the fitness value for each chromosome. The fitness value (**FT**) of a chromosome can be defined by the degree of agreement between the final binary output and the test pixels of building and non-building regions. For each chromosome, the **FT** value was calculated using the detection (**D**) and misdetection (**MD**) rates (Equation 10).

$$FT = 50 \times (D + (1 - MD)) \quad (\text{Equation - 10})$$

where, **D** is the fraction of test pixels marked as “building” that the classifier marks as “building” (true positive), plus the fraction of test pixels marked as “non-building” that the classifier marks as “non-building” (true negative). On the other hand, **MD** is the fraction of test pixels marked as “building” that the classifier marks as “non-building” (false negative), plus the fraction of test

pixels marked as “non-building” that the classifier marks as “building” (false positive). For instance, if $D=1$ then, MD becomes 0 and FT is computed to be 100, which is the best case. In the worst case, FT becomes 0, for which $D=0$ and $MD=1$. Note that, a fitness score of 50 can be achieved with a classifier that identifies all pixels as “building” or “non-building”.

4.3. Building Delineation

In this section, the delineation of the above extracted buildings is described. Since, many false alarm areas are likely to appear along with the candidate building patches, a preprocessing step is employed using the morphological image processing operations. Then, the building boundaries are extracted using those functions that operate based on the measurement of the properties of the binary image regions. The steps of the proposed building delineation procedure are presented in Figure 4.14.

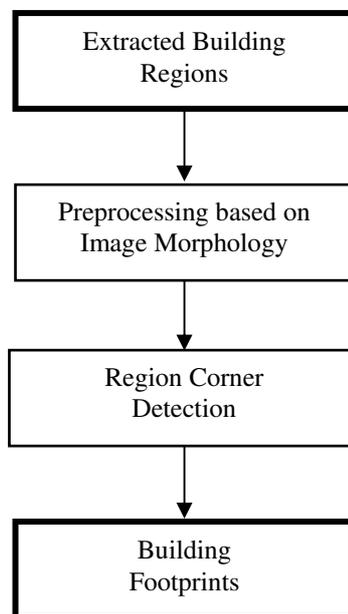


Figure 4.14. The workflow of the building delineation methodology.

4.3.1. The Preprocessing

The preprocessing operations carried out in the present study are based on image morphology, which is an important tool for extracting image components that are useful in the representation and description of region shape. Image morphology is based on set theory in mathematics and offers a powerful approach to numerous image processing problems. The morphological techniques are mostly used for pre- or post-processing, such as morphological filtering, thinning, thickening, and pruning (Gonzalez and Woods, 2008).

In the pre-processing stage, four fundamental morphological operations are employed in order to enhance the building regions. These are opening, artifact removal, hole filling, and convex image generation. First, the opening operation is performed to smooth the contours of the building regions and to eliminate the thin protrusions. The opening of set A by the structuring element B is denoted as $A \circ B$, which is formulated as

$$A \circ B = (A \ominus B) \oplus B \quad (\text{Equation – 11})$$

where, the symbols \ominus and \oplus denote the morphological erosion and dilation, respectively. Erosion tends to decrease the sizes of objects and remove small anomalies by subtracting objects with a radius smaller than the structuring element. On the contrary, dilation generally increases the sizes of objects and connecting areas that are separated by spaces smaller than the size of the structuring element.

In the present study, a disk-shaped structuring element with a radius (R) of 3 was used for the opening operation (Figure 4.15). This element was produced by the Matlab library function **strel('disk',3)**. Although other shapes of structuring elements, such as diamond, line, square and rectangle are also available, the disk-shape element with a radius of 3 was found to be more feasible in preserving the orientation of the building regions.

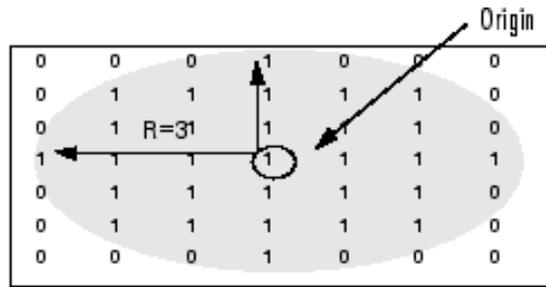


Figure 4.15. A disk-shaped structuring element with a radius of 3 (Matlab Online Documentation, 2010).

To implement the opening operation, Matlab’s **imopen(im,se)** library function was used. In this function, ‘*im*’ refers to input image, which contains the extracted building regions in the present case, and ‘*se*’ refers to formerly generated structuring element. To remove the isolated regions an artifact removal procedure was also performed immediately after the opening operation. For this operation, a threshold value of 250 pixels was selected. Therefore, the regions with the areas smaller than the threshold value were deleted from the binary image. A building block selected from the study area and the building regions extracted are illustrated in Figures 4.16(a) and 4.16(b), respectively. The composite effect of the opening and artifact removal operations are illustrated in Figure 4.16(c), where the isolated regions and small protrusions were eliminated to a great extent.

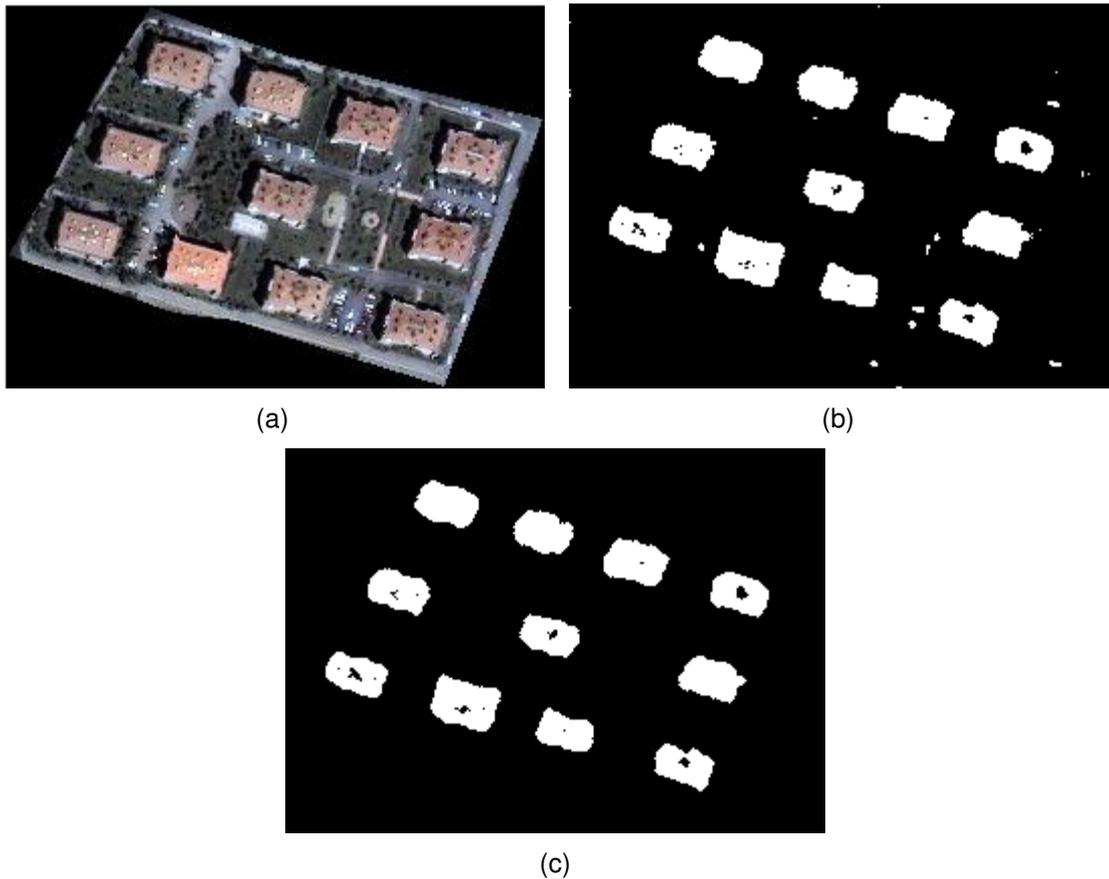


Figure 4.16. (a) A selected building block from the study area, (b) the extracted building regions, and (c) the building regions after applying the opening and artifact removal operations.

After applying the opening and artifact removal operations, the next step is hole filling. A hole is defined as a set of background pixels surrounded by a connected border of foreground pixels in a binary image. In general, the hole-filling algorithms are based on the combination of dilation, complementation, and intersection in an image (Gonzalez and Woods, 2008). In the present study, the hole-filling operation was carried out by Matlab's **imfill(bw,'holes')** library function, in which '*bw*' corresponds to binary image after opening process and '*holes*' refers to an optional argument. The binary image after filling the holes is shown in Figure 4.17. It is evident that the holes are removed successfully.



Figure 4.17. The building regions after applying the hole-filling operation.

The last pre-processing step was the convex image generation. In mathematical morphology, set **A** is defined to be convex if a straight line joining any two points in **A** lies within entirely **A**. The convex hull **H** of an arbitrary set **S** is the smallest convex polygon that can contain **S**. A convex image is a binary image that specifies the convex hull, with all pixels within the hull filled in. In this manner, the boundaries of greater complexity due to low image resolution are simplified. In the present study, the Matlab's **ConvexImage** property was used for obtaining the convex image. This function is the principle tool of Matlab's Image Processing Toolbox (IPT) for computing the region descriptors (Gonzales *et al.*, 2009). The building regions after converting to convex image are illustrated in Figure 4.18. As can be seen in the figure, the complex boundaries have become simpler that offers an advantage in extracting the footprints of the buildings.

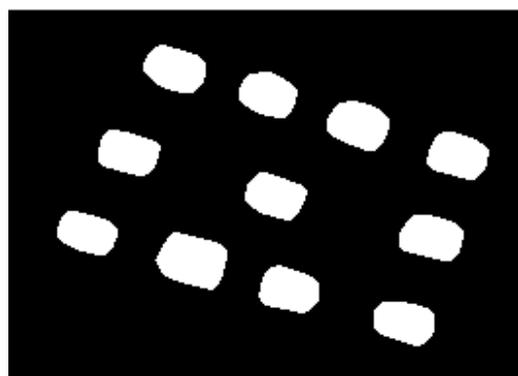


Figure 4.18. The final image after applying the pre-processing step of convex image generation.

The morphological pre-processing operations can also be implemented using the “Morphological Pre-Processing” option of the “2-D Delineation” menu item in PREBUM. To perform the aforementioned morphological operations, the push button “Load Default Parameters” is clicked. In addition to opening, threshold, hole filling and convex image generation functions, the software also includes the morphological functions of closing, erosion, and dilation along with their structuring element sizes. A sample configuration of the morphological functions is presented in Figure 4.19, in which the functions Opening and Threshold are initialized with certain values and Hole Filling and Convex Image Generation options are enabled. On the other hand, Closing, Erosion, and Dilation operations are disused.

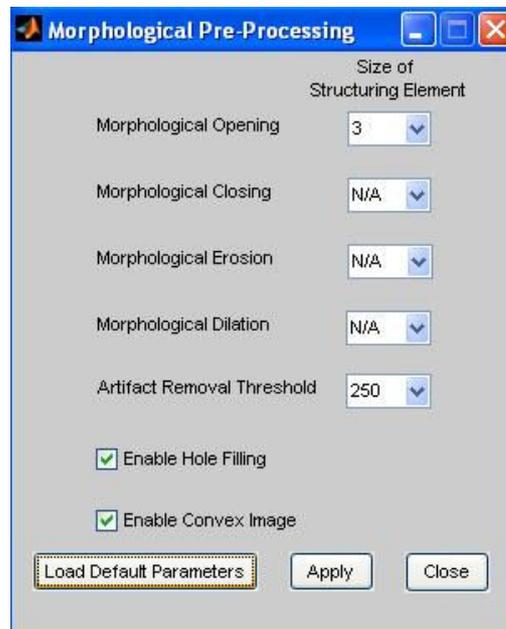


Figure 4.19. A sample configuration for the morphological pre-processing functions of the PREBUM software.

After the initialization, the enhancement process is triggered by clicking the “Apply” button. After a while, the enhancement step is terminated and the enhanced building patches are stored in a file named “enhanced_buildings.bmp” and recorded on the “Layers” panel (Figure 4.20).

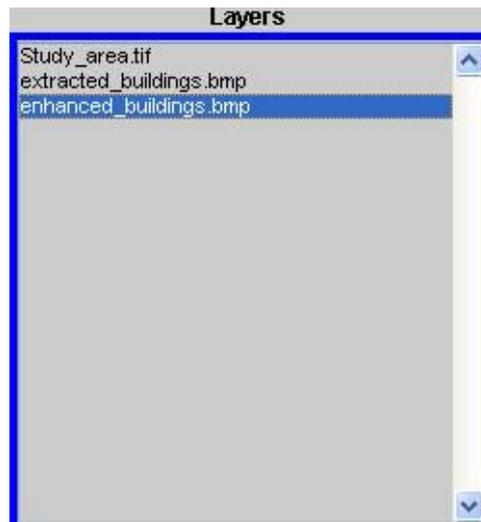


Figure 4.20. The view of the “Layers” panel after the morphological pre-processing step.

4.3.2. Region Corner Detection

The last step before producing the building footprints is to detect the corners of the extracted building patches. Since the quadrilateral buildings are considered in this study, four corner points are sufficient to delineate a boundary. To detect the corners, first the best fitting ellipse is generated for each building patch. This is performed by the **regionprops** function of Matlab. This function returns the ellipse parameters, such as orientation, foci, semi-minor axis and semi-major axis. An example for the best fitting ellipse is given in Figure 4.21, in which the left side shows an image region and its best fitting ellipse, while on the right side the parameters are indicated graphically, where the solid blue lines represent the axes and the red dots represent the foci. The orientation of the ellipse is computed by using the angle between the horizontal dashed line and the major axis.



Figure 4.21. The generation of the best fitting ellipse (Matlab Online Documentation, 2010).

In the next step, the rectangle having the largest area that can be inscribed in the ellipse is determined and the corners (1 to 4) of this rectangle are defined as the region corners (Figure 4.22).

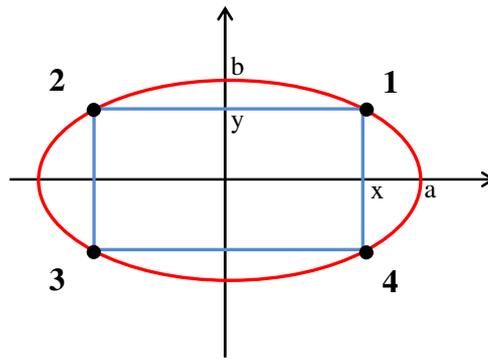


Figure 4.22. The computed region corner points (1-4).

To do that, ' y^2 ' is written in terms of ' x^2 ' using the squared area of the rectangle and the equation of a standard ellipse, where

$$(Area_{rect})^2 = (2x * 2y)^2 \quad (\text{Equation - 12})$$

$$Ellipse = \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1 \quad (\text{Equation - 13})$$

$$y^2 = b^2 \left(1 - \frac{x^2}{a^2}\right) = \frac{b^2}{a^2} (a^2 - x^2) \quad (\text{Equation - 14})$$

$$(Area_{rect})^2 = S(x) = 16 \frac{b^2}{a^2} x^2 (a^2 - x^2) \quad (\text{Equation - 15})$$

To maximize the area of the rectangle, the first derivative of $S(x)$ should be 0.

$$S'(x) = 0 \rightarrow 16 \frac{b^2}{a^2} [2x(a^2 - x^2) + x^2(-2x)] = 0$$

$$S'(x) = 32 \frac{b^2}{a^2} x(a^2 - 2x^2) = 0 \quad (\text{Equation – 16})$$

Either $(x = 0)$ or $(a^2 - 2x^2 = 0)$ should be satisfied. From here, 'x' is found to be $\frac{a}{\sqrt{2}}$. In a similar way, 'y' can be computed as $\frac{b}{\sqrt{2}}$ for the corner point

1. The remaining corners for point-2, point-3, and point-4 are found to be $\left(\frac{-a}{\sqrt{2}}, \frac{b}{\sqrt{2}}\right)$, $\left(\frac{-a}{\sqrt{2}}, \frac{-b}{\sqrt{2}}\right)$, $\left(\frac{a}{\sqrt{2}}, \frac{-b}{\sqrt{2}}\right)$, respectively. Since, the semi-major axis (a) and semi-minor axis (b) are known before, the corner points are computed easily, therefore.

However, the orientations of the building regions are generally different from the case given in Figure 4.22. Therefore, the patches are rotated by α radian in anti-clockwise direction. The rotation angle α is computed from the orientation parameters that are provided by the **regionprops** function. The rotated corner points (X_{Ri}, Y_{Ri}) are calculated by multiplying the original corner points (X_i, Y_i) by the rotation matrix. Then, the resultant matrix after multiplication is added to original corner points.

$$\begin{bmatrix} x_{Ri} \\ y_{Ri} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (\text{Equation – 17})$$

After detecting the corners of the building patches, the building boundaries are delineated by simply connecting the corners (vertices) in the correct order. This is achieved using the Bresenham's line algorithm, which is used to determine what points in an n-dimensional raster should be plotted in order to form a close approximation to a straight line between two given points (Bresenham, 1965). For a selected sub area, the extracted building footprints are illustrated in green color in Figure 4.23.



Figure 4.23. The extracted building footprints overlaid with the image.

4.3.3. The Accuracy Assessment of Building Delineation

The assessment of the delineated building footprints is carried out by means of computing the positional accuracy. The positional accuracy is the expected deviation of an object from its original location. It is generally measured by selecting a specified number of sample points in a prescribed manner and comparing the position coordinates with a reference source of information (Aronoff, 1989).

In the present case, to assess the positional accuracies of the building footprints detected, the reference source information was prepared using the GIMP-2 image processing software. Then, for each building, the coordinates of four corner points were measured on the reference source so that they could be compared with the corresponding coordinates in the output image. Figure 4.24 illustrates a sample building selected in which the reference and the output footprints are indicated by colored lines together with the corner coordinates.

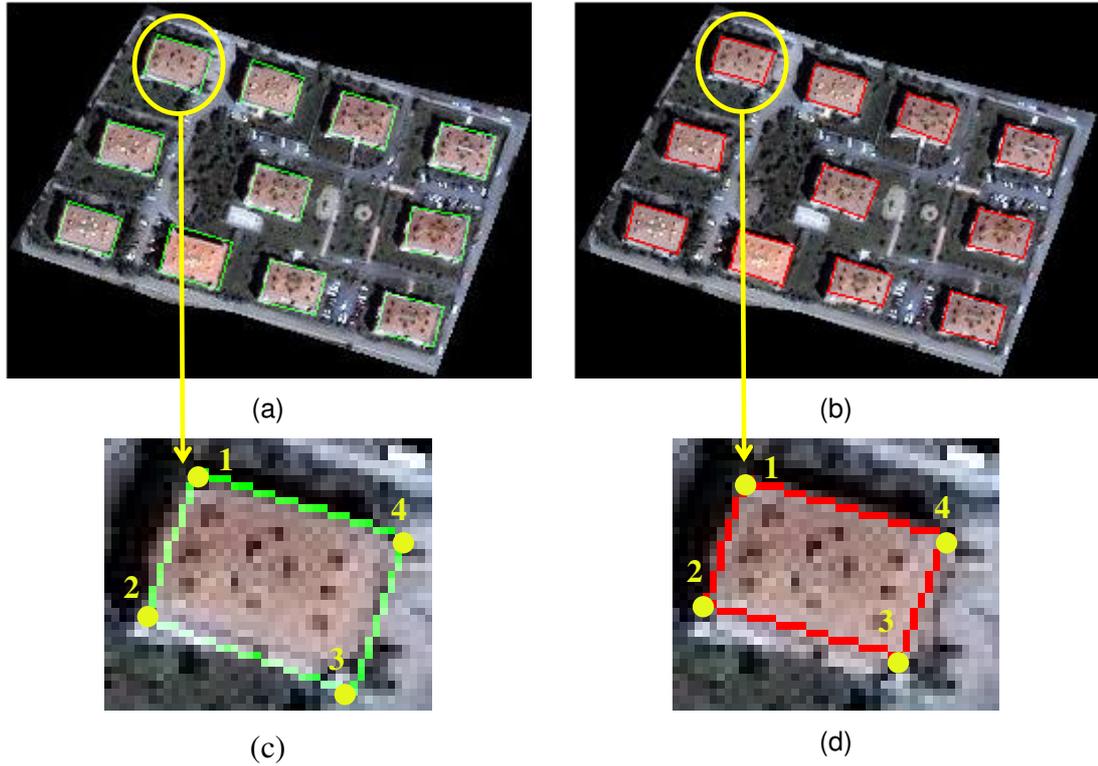


Figure 4.24. For a selected sub area, (a) the extracted footprints and (b) the reference footprints. (c) The extracted corners and (d) the reference corners of a building.

For each corner point, the pixel distance errors (*DistErr*) were computed from the specified (x,y) pairs using the following formula,

$$DistErr = \sqrt{(X_R - X_P)^2 + (Y_R - Y_P)^2} \quad (\text{Equation-18})$$

where, (X_R, Y_R) correspond to test points on the reference image and (X_P, Y_P) are the points selected from the output image. Without specifying the level of confidence, the stated distance errors become actually quite meaningless. Therefore, a normal distribution model was used and, for each building, the accuracies were computed at the confidence levels 80%, 85%, 90% and 95%, using the below given formula,

$$Accuracy = Z_Value * SD_DistErr + Mean_DistErr \quad (\text{Equation-19})$$

where, *Z_Value* denotes Z score for the corresponding level of confidence, *SD_DistErr* is the standard deviation of distance error and *Mean_DistErr* is the

mean of distance error. For the confidence levels 80%, 85%, 90% and 95%, respectively the Z scores of **0.84**, **1.045**, **1.28**, and **1.645** were used.

For a sample building cropped from the sub-image patch, the pixel errors of each corner point and the accuracies under different confidence levels are illustrated in Tables 4.7 and 4.8. In this case, *SD_DistErr* and *Mean_DistErr* were found to be **1.20** and **2.68**, respectively.

Table 4.7. For a sample building the pixel errors on four corner points.

CORNERS	X_R	Y_R	X_P	Y_P	DistErr
1	76	19	78	18	2,24
2	71	36	71	38	2,00
3	97	44	99	48	4,47
4	104	27	106	27	2,00

Table 4.8. For a sample building, the accuracies under different confidence levels.

CONFIDENCE LEVEL	Z_VALUE	ACCURACY
80%	0,84	3,69
85%	1,045	3,93
90%	1,28	4,22
95%	1,645	4,65

CHAPTER 5

BUILDING FACADE TEXTURE ACQUISITION

This chapter presents an approach for automatic acquisition of the realistic facade textures from ground-level building photographs. A series of methods were applied successively to obtain building facade texture images in order to map on 3-D building models. This chapter comprises; the proposed workflow, automatic texture extraction, and rectification along with the semi-automatic occlusion removal. Besides, the evaluation of the accuracy in each step is described.

5.1. The Proposed Workflow

The main steps followed in the proposed building facade texture acquisition approach are shown in Figure 5.1. The approach is initiated with the automatic texture extraction, which is carried out using a repetitive watershed segmentation. Then, the extracted facade texture images are geometrically rectified in an automated way. Finally, an occlusion removal step is employed to get rid of the obstacles that block the facade texture.

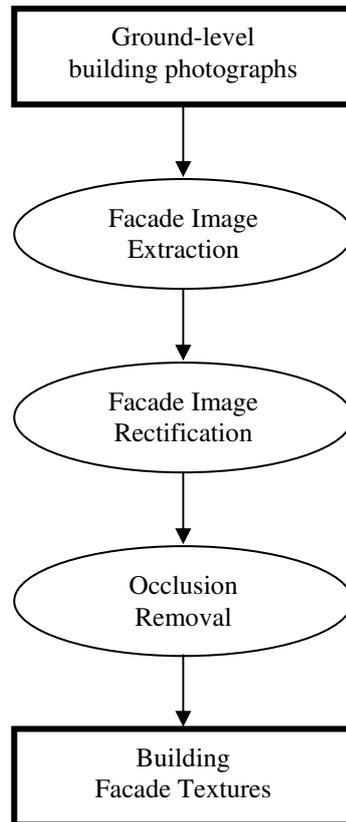


Figure 5.1. Building facade texture acquisition.

5.2. Facade Image Extraction

The building facade textures are automatically extracted from the ground-level building photographs. First, a preprocessing step is employed to reduce the non-uniform illumination effects on the facade images. Then, the marker pixels are initialized automatically both for the foreground (building) and the background (out of building) regions to initiate the segmentation. After that, the watershed segmentation is carried out repetitively until a stable foreground segment is obtained. These steps are repeated until a specified number of iterations are reached. To minimize the over-segmentation, the texture segments produced at the end of each iteration are overlaid using the and-logic. Figure 5.2 summarizes the proposed automatic texture extraction technique.

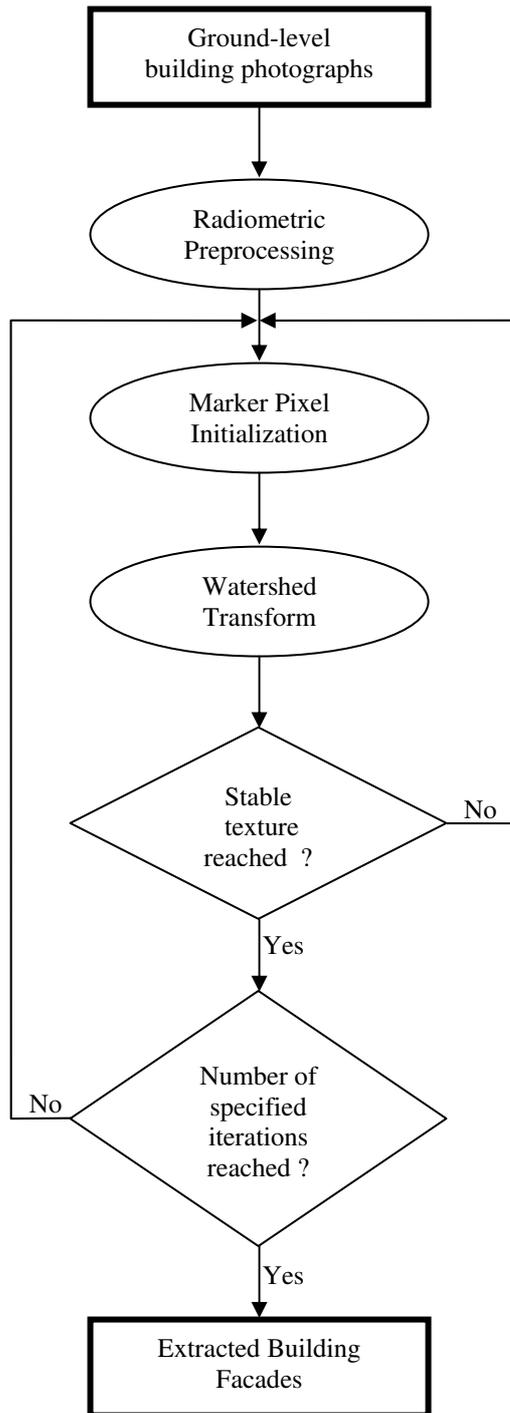


Figure 5.2. The workflow of the proposed automatic facade image extraction technique.

5.2.1. Radiometric Pre-processing

As is known good illumination conditions improve the efficiency of every image analysis application. However, uniform illumination condition is rarely obtained due to many factors, such as sun angle, casting shadows, etc. In a segmentation process, non-uniform illumination is one of the main reasons for the failure conditions causing over- and under-segmentation.

In this study, the building facade photographs acquired from ground level mostly suffer from the non-uniform illumination. A building with non-uniform illumination is shown in Figure 5.3.

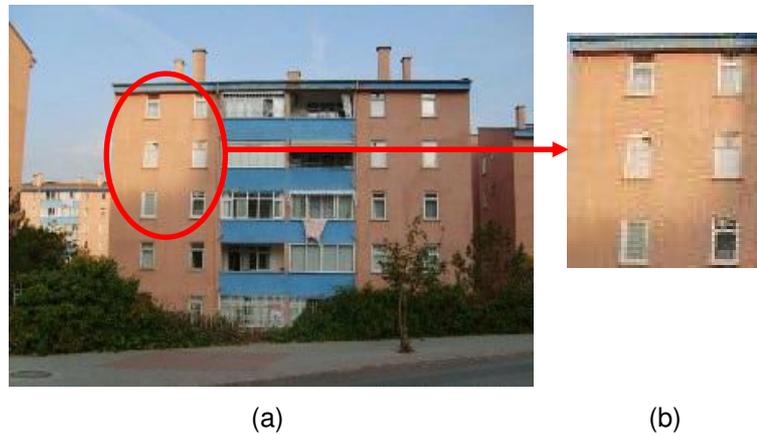


Figure 5.3. (a) A ground level photograph of a building having non-uniform illumination and (b) the region of interest in a closer view.

To minimize this effect, the RGB (**R**ed-**G**reen-**B**lue) image was transformed into HSI (**H**ue-**S**aturation-**I**ntensity) color space. The HSI color space is known as an ideal tool for developing image processing algorithms and also decouples intensity component from the color-carrying information in a color image. The adjustment of the saturation component was found to be useful in the pre-processing step due to the fact that it gives a measure of the degree to which a pure color is diluted by white light (Gonzales and Woods, 2008). The computation of the saturation component is as follows:

$$S = 1 - \frac{3}{(R + G + B)} [\min(R, G, B)] \quad (\text{Equation - 1})$$

The non-uniform illumination is reduced by multiplying the saturation component by a saturation coefficient (sc), of 5. At present study, this value was found to be optimum, as it provided the most successful segmentation accuracies after performing several experiments. In PREBUM, the saturation boosting is achieved by the “Adjust Saturation” option under the menu item of “Pre-processing”. To obtain the predetermined coefficient value, the “increase (+)” button is pressed five times, consecutively. Figure 5.4 illustrates the building shown in Figure 5.3 after boosting the saturation.

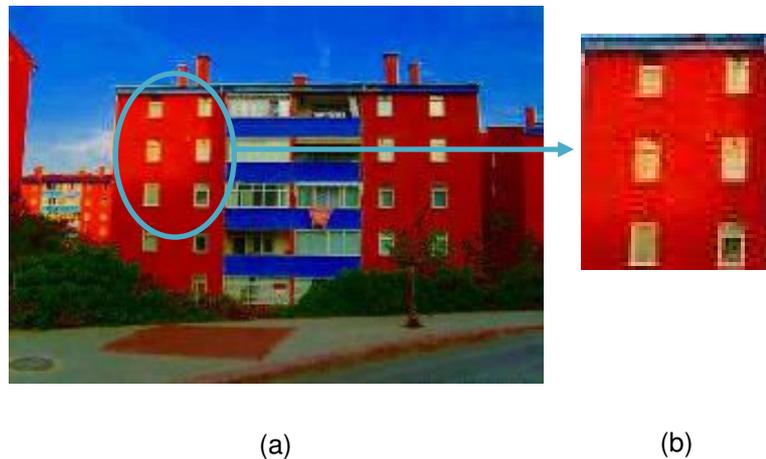


Figure 5.4. (a) The building shown in Figure 5.3 after boosting the saturation and (b) the region of interest in a closer view.

5.2.2. Initialization of the Marker Pixels

Before executing the watershed transform, the initial marker pixels (seeds) must be seeded in the input image. In this study, the initial markers are located automatically both for the foreground and background regions. It is assumed that for the foreground (building) region the initial marker pixels fall in the middle of the image frame. Similarly, the background seeds for the non-building objects (sky, pavement, neighboring buildings, etc.) are located near the edges of the image frame (Figure 5.5). The distribution of the new

markers in further stages of the repetitive watershed segmentation will be discussed in the following section.

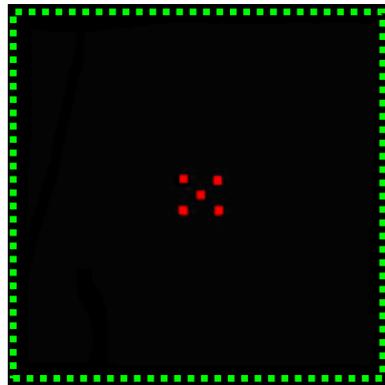


Figure 5.5. The initial markers for the foreground (red) and background (green) regions.

5.2.3. Watershed Transform

The watershed transform grows out of mathematical morphology and takes its inspiration from hydrology and the study of watersheds (Beucher and Meyer, 1992). Hydrological watersheds partition the landscape based on watershed lines (or ridges) and the valleys between them. When it rains, water will flow to low lying catchment basins of the landscape. The boundaries between these basins are called the watersheds (Figure 5.6).

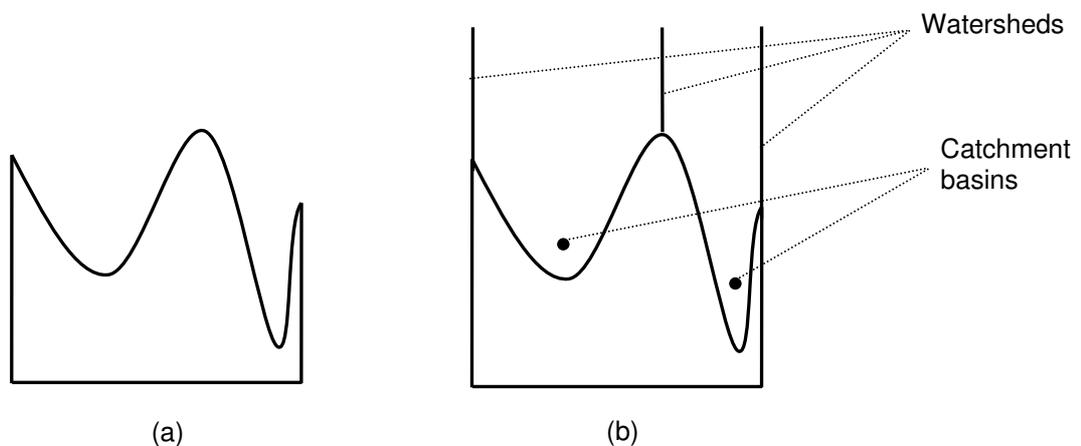


Figure 5.6. Illustration of the concept of watershed segmentation: (a) The profile of gray level image; (b) Local minima of gray level yield catchment basins, local maxima define the watershed lines.

Watershed transformation is the delineation of watershed lines from a gradient image derived from the input image based on immersion simulation. In the gradient image, high magnitudes correspond to distinct boundaries in the input image, whereas uniform smooth areas of the input image have low gradient that represent low magnitude surface (Figure 5.7). The algorithm imposes a discrete set of gray values on the image and then expands each catchment basin from its minimum grey level by iteratively merging the nearest connected-component regions of the next highest gray level. Any pixels that are equidistant from two basins are labeled as watershed boundaries (Vincent and Soille, 1991).



Figure 5.7. The gradient image.

In the present case, as a contribution to original watershed transform, a repetitive segmentation method is proposed, such that for each run, new foreground markers are automatically positioned randomly inside the newly segmented regions. The number of the new markers is computed as one percent of the total image area. That is, if the original image is 480×480 pixels, then the marker count becomes $480 \times 480 \times 0.01 = \mathbf{2304}$. By using the repetitive segmentation approach, more foreground information is extracted compared to original watershed method. This is because of the fact that the amount of texture produced in the next repetition is more than or equal to amount of texture in the current repetition. In Figure 5.8, the

distribution of the marker pixels and the corresponding building segments up to 4th repetition are illustrated.

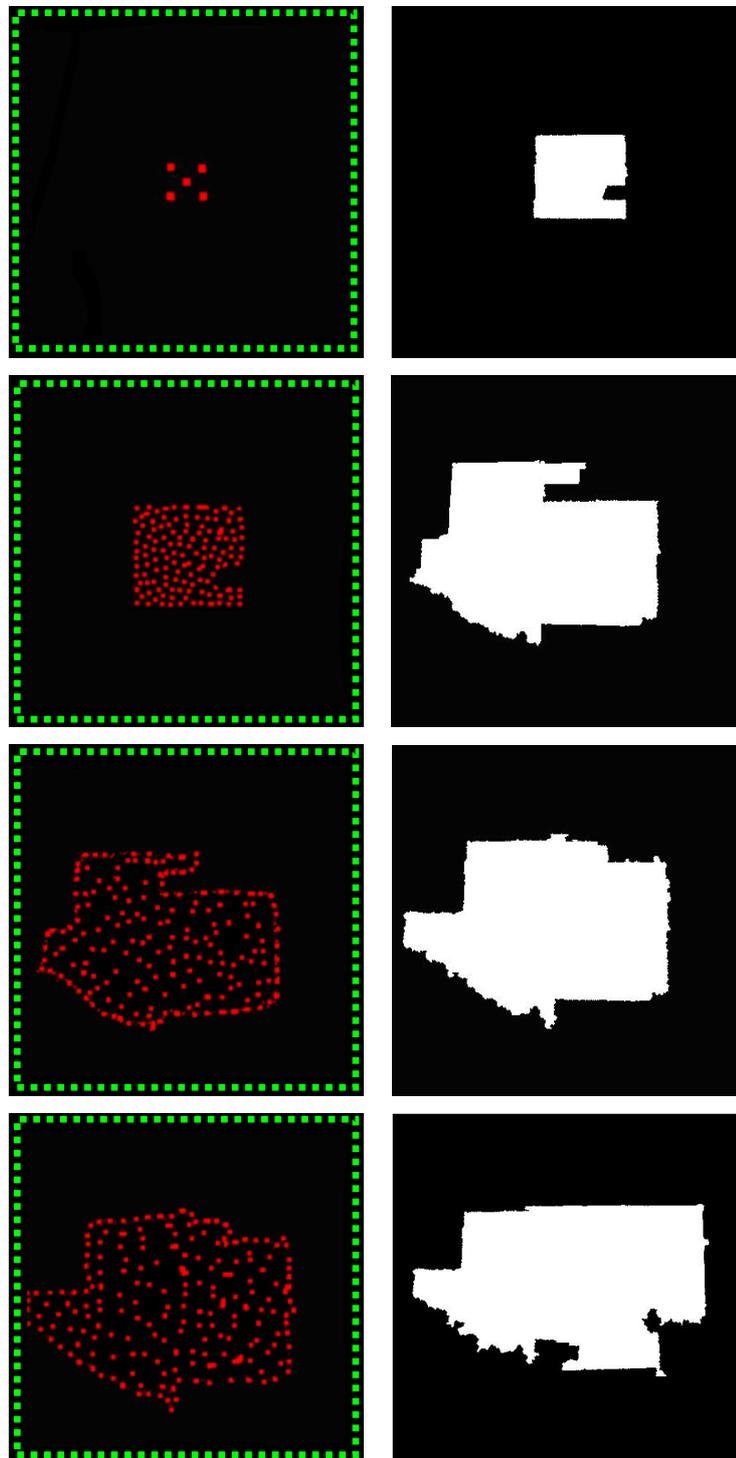


Figure 5.8. The distributions of the marker pixels (left column) and the segments produced (right column) in the first (row-1), second (row-2), third (row-3) and fourth (row-4) iterations.

To limit the segmentation, a stopping criterion was determined. Therefore, the execution procedure is terminated either by providing a stable texture ratio (tr) or by reaching a maximum repetition count (mrc). To check the first condition, “tr” (extracted pixels / total image pixels), is calculated for each segment. The differences between the consecutive ratios are also computed and stored. If any difference ratio is smaller than a predefined texture ratio threshold (trt) then the algorithm stops. This means that the growing of the foreground region is highly diminished or completely ended. In the present case the “trt” constant was set to **0.001**. In the second condition, the stopping decision is made based on “mrc”, which is bounded by **50**. This time, the segmentation is terminated at the 50th repetition, at worst. However, in general this case ends up with an over-segmentation problem and a post processing step is needed.

To reduce the over-segmentation problem, the whole watershed segmentation procedure is re-executed from the beginning and new segments are stored separately. This process is repeated until a number of specified iterations are reached. This parameter is called the maximum iteration count (mic). Based on a number of experiments, the optimum value was determined to be **10** for this parameter. Figure 5.9 illustrates the resulting segments after performing ten independent iterations.

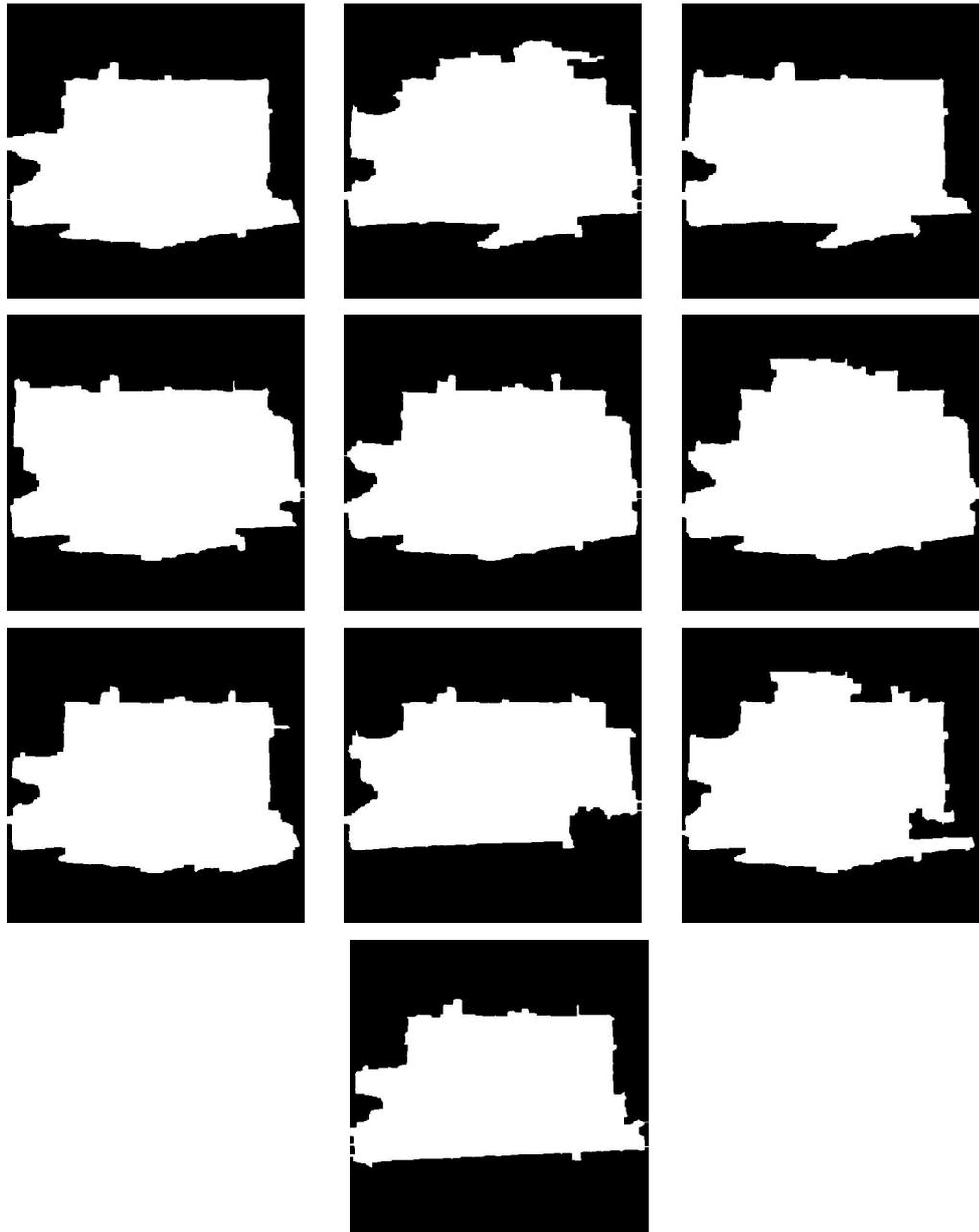


Figure 5.9. The resulting segments after performing ten independent iterations.

Next, the generated segments are overlaid and the intersections of the foreground regions are extracted. In this manner, the over-segmentation would likely be reduced to a reasonable level. The building facade patch in binary form and the extracted facade image in RGB form are shown in Figure 5.10. Note that the saturation value of the RGB image was reduced back to original value.

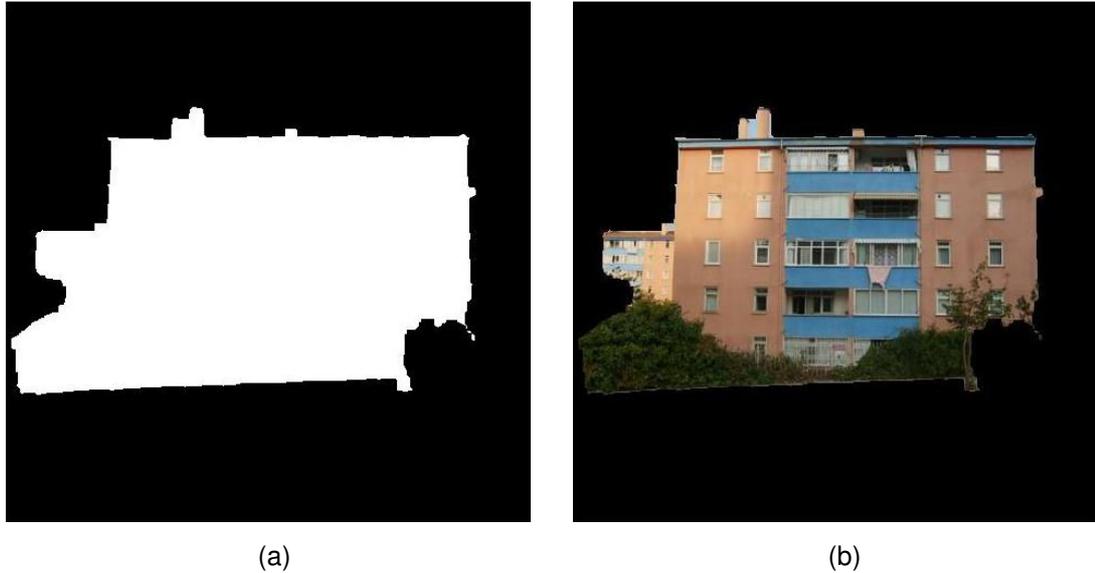


Figure 5.10. (a) The building facade patch extracted through intersecting the segments and (b) the extracted RGB facade image.

In PREBUM, the initialization of the facade segmentation parameters “mrc” and “mic” is carried out using the “Load Facades and Set Parameters” option under the menu item of “Texture Generation”. The predefined values of 50 and 10, respectively can be set for the parameters “mrc” and “mic”, by clicking the “Load Default Parameters” button. Other than these values, the user is allowed to select any other values using the corresponding popup menus. A screenshot that shows two different parameter settings is given in Figure 5.11.



Figure 5.11. The texture extraction parameters with (a) the predefined values and (b) the arbitrary values.

After the parameters are set, the facade segmentation procedure is initiated using the “Texture Extraction” option under the same menu item. The

growing images in each repetition of watershed segmentation are shown on the view panel. During segmentation, the current counts of repetition and iteration parameters are also demonstrated in a textbox. Figure 5.12 illustrates an example that shows the status of these parameters along with the growing image, in which the instantaneous iteration and repetition counts are shown as 1 and 7, respectively.

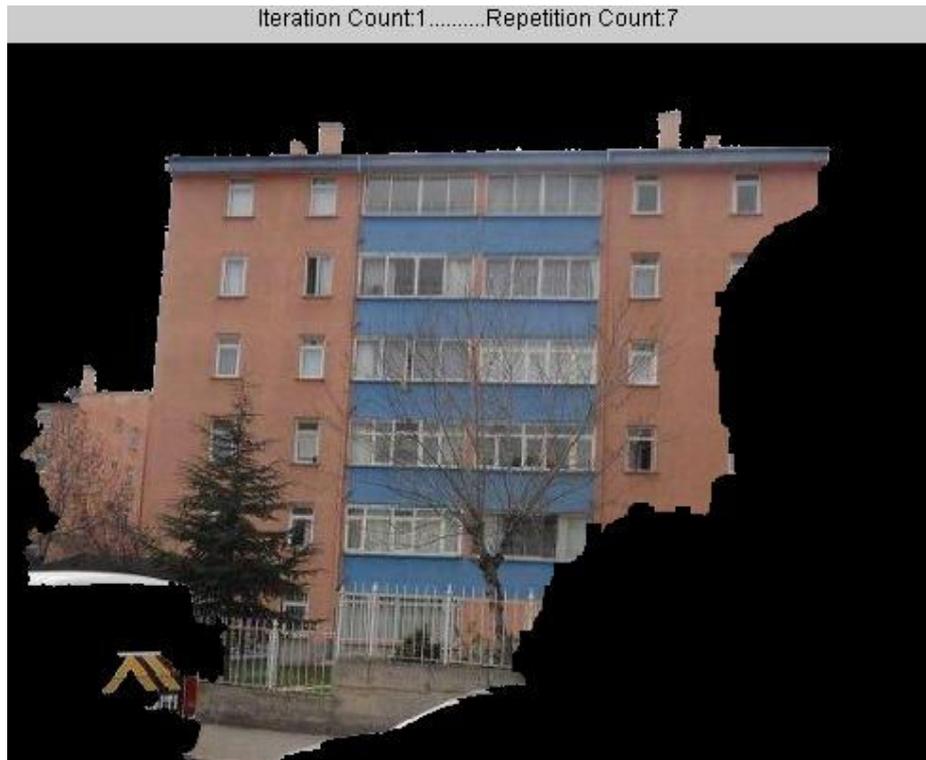


Figure 5.12. A screenshot of a segmentation, from which the current repetition count is 7 and the current iteration count is 1.

At the end of the facade segmentation process, the resulting facade image is stored in a file named "Facade_extracted_image.jpg" and recorded on the "Layers" panel. Two identification suffixes (1 or 2) are appended at the end of the file names in order to distinguish the two adjacent facades belonging to the same building. A screenshot of "Layers" panel after the facade texture extraction is given in Figure 5.13. Note that the original facade images are stored as "Facade_image_1.jpg" and "Facade_image_2.jpg", while the extracted facade image files are "facade_extracted_image_1.jpg" and

“facade_extracted_image_2.jpg”. “1” and “2” are the file name suffixes which indicate the adjacent facade ids of a single building.

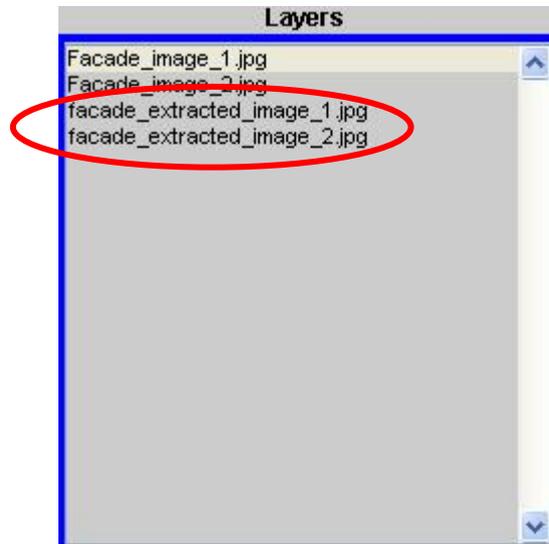


Figure 5.13. A screenshot of the "Layers" panel after the facade texture extraction process.

5.2.4. The Accuracy Assessment of Facade Image Extraction

The segmented facade images are assessed by a quantitative evaluation metric developed by Shufelt and McKeown (1993). This metric is based on labeling of pixels by comparing the output of the proposed methodology with the reference data. For labeling, four possible categories may occur for each pixel that are; true positive (**tp**), true negative (**tn**), false positive (**fp**) and false negative (**fn**). In true positive case, both the analysis results and the reference data label the pixels as belonging to foreground (building facade). In true negative case, the background is labeled both by the analysis results and the reference data. In false positive case, the analysis results label the pixels as belonging to foreground, while the reference data labels them as background. The false negative case is the exact opposite of the false positive case. These four cases are illustrated in Figure 5.14, where “A” (rectangle) is the reference region and “B” (trapezoid) is the extracted region. The green area corresponds to “**tp**” pixels, while the yellow refers to “**tn**”. On

the other hand, the blue and the orange areas correspond to “fn” and “fp” pixels, respectively.

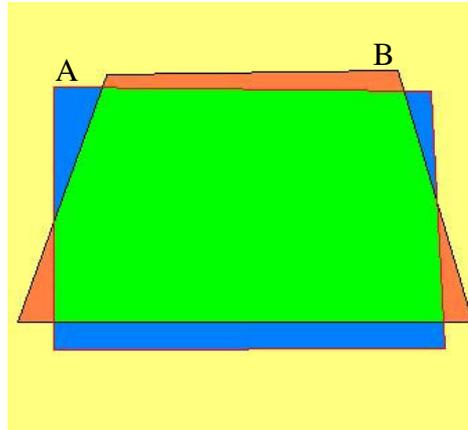


Figure 5.14. The four possible cases used in the accuracy assessment of the facade image extraction procedure.

To evaluate the performance, the counts of “tp”, “tn”, “fp” and “fn” pixels are calculated and facade detection percentage (**fdp**), branching factor (**bf**), and quality percentage (**qp**) metrics are computed.

$$fdp = \frac{100 * tp}{tp + fn}, \quad bf = \frac{fp}{tp}, \quad qp = \frac{100 * tp}{tp + fp + fn} \quad (\text{Equation-2})$$

The metric “fdp” can be treated as a measure of object detection performance. It evaluates the fraction of reference pixels labeled as building facade pixels by the proposed technique. The metric “bf” is the measure of over-segmentation, in which the analysis labels background pixels as foreground pixels. If there is no over-segmentation, the value of “bf” would become zero. On the other hand, the case with the “bf” value of one would incorrectly label a background pixel as a foreground pixel for every foreground pixel it correctly detected. Finally, the metric “qp” was used to measure the absolute quality of the proposed method. In order to obtain 100 percent quality, the proposed method must produce a perfect segmentation of the foreground with respect to reference data. In other words, the

proposed method must correctly label every pixel, without missing any ($fn=0$) and without mislabeling any background pixels ($fp = 0$).

5.3. Facade Image Rectification

After extracting the building facades, the next step is the automatic rectification of the facade images. The purpose of the rectification procedure is to remove or at least reduce the perspective distortions due to improper viewing angles of the camera. Perspective distortion is known as warping of an object and its surrounding area that differs significantly from what the object looks like with a normal focal length. As can be easily seen in Figure 5.15, the extracted facade image contains vertical and horizontal distortions.



Figure 5.15. The distortions with respect to horizontal and vertical reference lines.

In the present case, the proposed approach reduces the perspective distortion in an automated way. In order to do that a series of operations are

employed. These include the detection of the edges using the Canny operator, the extraction of the strongest vertical lines through Hough transformation, the end-point detection, the adjustment of the vertical facade edges, and the projective transformation. The workflow of the proposed approach to perform facade image rectification is given in Figure 5.16.

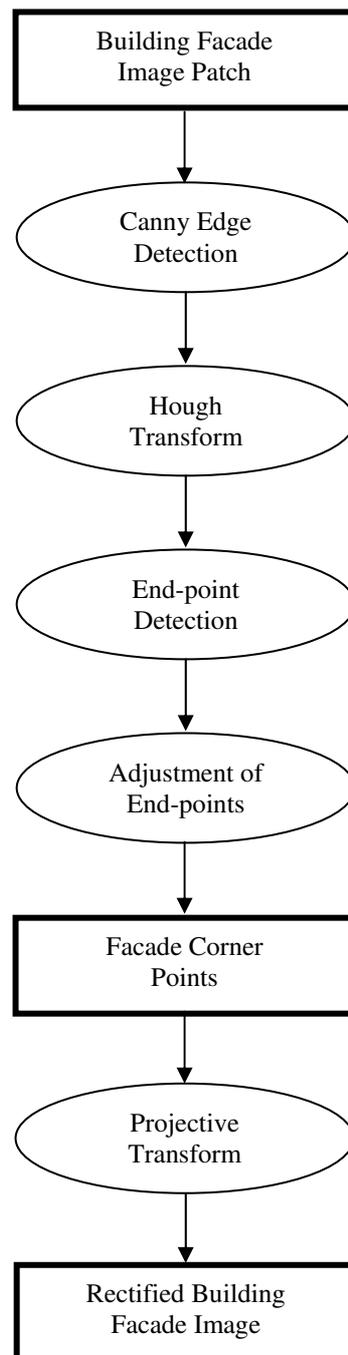
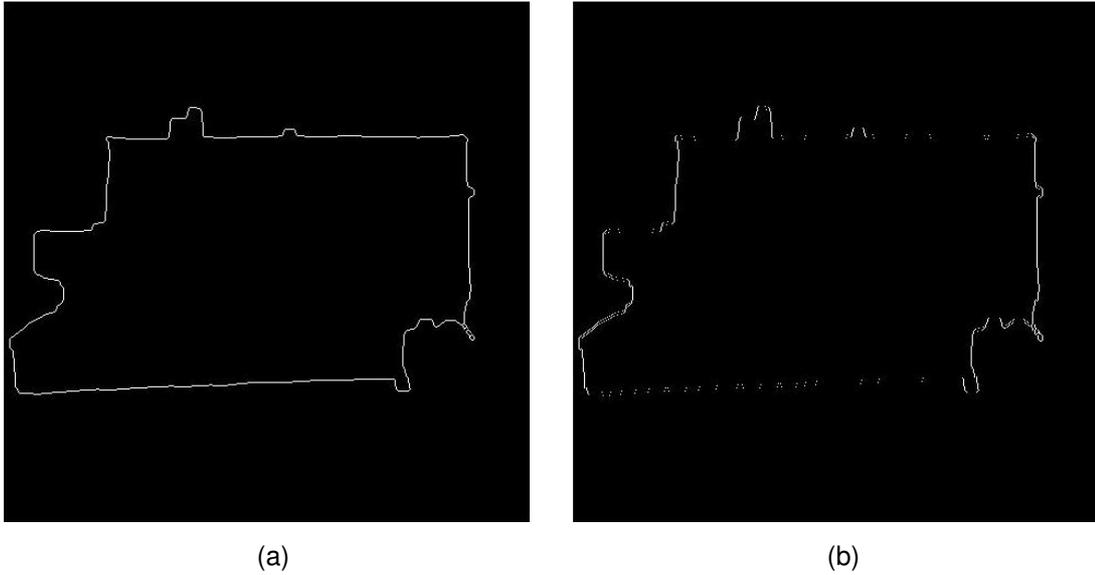


Figure 5.16. The workflow of the proposed automatic facade image rectification procedure.

5.3.1. Facade Edge Analysis

The strongest vertical edges of the building facade patch are extracted using the Canny edge detector and the Hough transform. For a ground level image, the vertical edges can be said to be more important than the horizontal edges. This is due to the fact that the occlusions usually block the lower part of a building facade and therefore, the extraction of the bottom edges become almost impossible. On the other hand, the vertical facade edges can be recovered totally or partially, at worst. Thus, it would be adequate to detect the facade corner points.

The Canny edge detector is a powerful multi-stage algorithm in detecting wide range of edges in many image processing applications. The algorithm includes; noise reduction through a Gaussian filtering, determination of the intensity gradient, non-maximum suppression and edge tracing, and hysteresis thresholding (Canny, 1986). Therefore, in the present case the Canny edge detector was applied for extracting the edges of the building facade patches. Since the vertical edges are of more importance, the horizontal edges are eliminated using a 3x3 filter, **H** (Figure 5.17).



H:

-1	2	-1
-1	2	-1
-1	2	-1

(c)

Figure 5.17. (a) The edges after applying the Canny edge detector, (b) the remaining edges after removing the horizontal edges and (c) the 3x3 vertical template, H.

After detecting the vertical edges, the next step is to reveal the strongest edge information. This is performed using the Hough transformation, which is based on the mapping the information in the image (feature space) into a parameter space. The following equation and Figure 5.18 give the basis of the process (Vozikis,2009).

$$H(\theta, \rho) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(x, y) \delta(\rho - x \cos(\theta) - y \sin(\theta)) dx dy \quad (\text{Equation - 3})$$

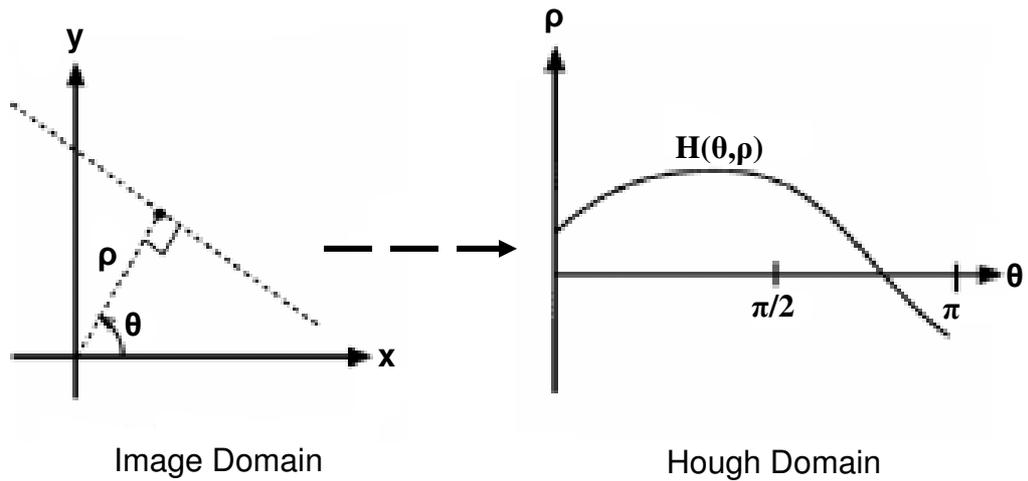


Figure 5.18. The Hough Transformation.

where, δ is the Dirac delta function, (θ, ρ) and (x, y) are Hough and image domains, respectively. Each point (x, y) in the original image $F(x, y)$ is transformed into a sinusoid $\rho = x \cos(\theta) - y \sin(\theta)$ and $H(\theta, \rho)$, which gives the total number of sinusoids that intersect at point (θ, ρ) . Hence, it gives the total number of points making up the line in the original image. By choosing a threshold T for $H(\theta, \rho)$ and using the inverse Hough transform, the original image is filtered so that the lines containing at least T points are kept only.

Figure 5.19 illustrates the result of the Hough transform applied on the binary image shown in Figure 5.17(b) using a line-cut threshold (T) value of 150. Note that the resulting Hough lines were thickened by one pixel for the adjustment of end points, which will be described in the following section.

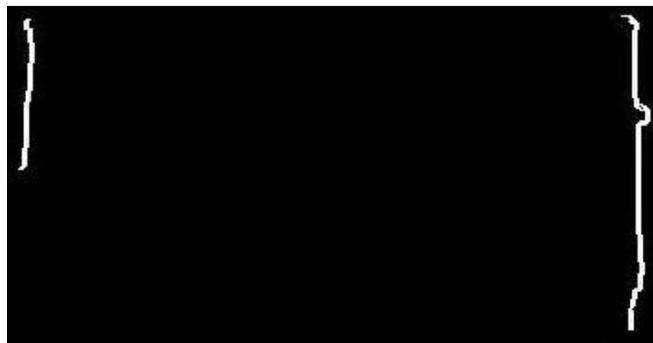


Figure 5.19. The facade edges detected using the Hough transform with the line-cut threshold value of 150.

5.3.2. End-Point Analysis

After detecting the Hough lines, the next step is to detect the end-points of the vertical edges. This process is regarded as the preliminary step before the detection of the facade corners.

The end-points are detected using the Harris corner detector, which is a widely accepted method for interest point detection. The method defines a corner to be a point with low self-similarity. The similarity is measured by taking the weighted sum of squared differences (SSD) between two image patches. A corner is characterized by a large variation of SSD in all directions of the vector $(x \ y)$ (Harris and Stephens, 1988).

The detected four end-points (colored in red) of the Hough lines are shown in Figure 5.20. As can be predicted easily, there may be a number of cases, in which more than four end-points can be detected. In these cases, the closest four end-points to image corners (upper left, upper right, lower left and lower right) are selected. Since in this study the rectangle facade images are considered, four end-points are needed inherently.

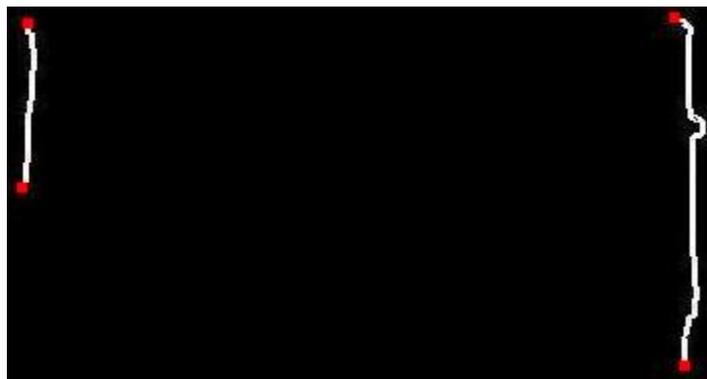


Figure 5.20. The end-points of Hough lines located by the Harris corner detector.

Due to the occluded areas on the building facade or the segmentation problems, the facade edges may not be extracted accurately. In parallel, the end-points may not be detected correctly. As can be seen in Figure 5.20, the

position of the lower end-point of the left edge is not correct. To fix this problem, the erroneous end-point is moved downward to the level of the right lower end-point. By doing this the approximate position of the erroneous point is estimated (Figure 5.21).

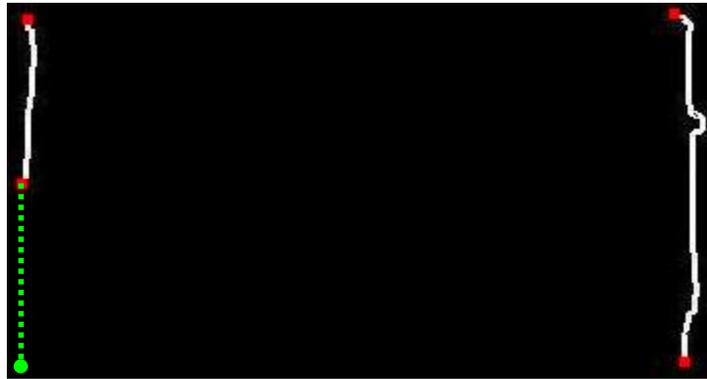
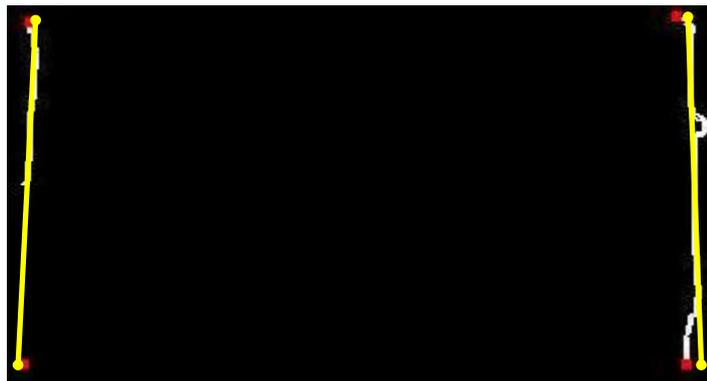


Figure 5.21. The estimated position of the erroneous end-point, which is shown in green.

Because of the curvilinear nature of the Hough lines, the current end-points are not yet the final positions for performing the geometric rectification. Therefore, to identify the facade corner points a trend line is estimated in an iterated way. To do that first, the lines that connect the upper and lower end-points of the facade edge are drawn (Figure 5.22(a)). Then, a wide range of different lines are drawn in a similar way by moving the upper and lower end-points in the horizontal direction. For each line drawn, the agreement is measured with the actual facade edges by counting the coinciding pixels. And, the line having the maximum coinciding pixels is selected as the trend line. The initial (blue) and the final (yellow) trend lines for the left and right facade edges are shown in Figure 5.22. Note that, the yellow dots illustrate the final positions of the end-points (facade corner points).



(a)



(b)

Figure 5.22. The orientations of (a) the initial trend lines and (b) the final trend lines.

5.3.3. Projective Transformation

The last step of the facade image rectification is the projective transformation, which maps lines to lines but does not necessarily preserve parallelism. Any plane projective transformation can be expressed by a 3x3 non-singular homogenous coordinates. In other words, any non-singular 3x3 matrix defines a projective transformation of the plane. The computation of a projective transformation is illustrated in Figure 5.23 and Equation 4.

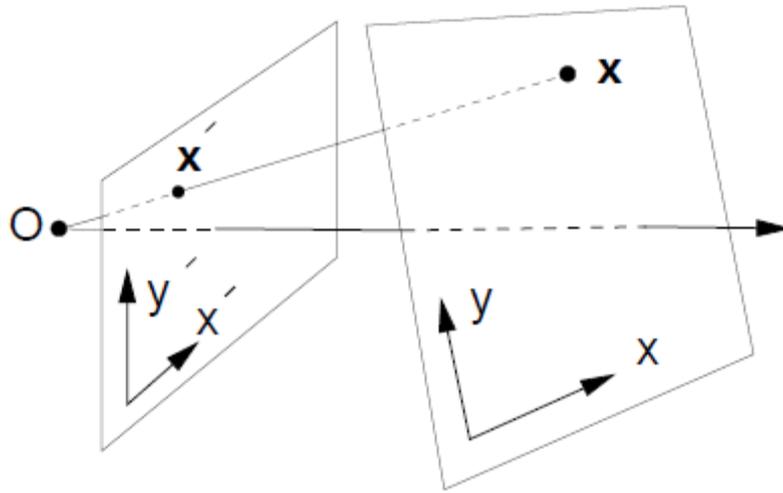


Figure 5.23. A projective transformation with center O, mapping xy plane to x'y' plane.

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}}_H \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (\text{Equation - 4})$$

where, H is a 3x3 non-singular homogenous matrix, $(X_1 X_2 X_3)$ corresponds to XYZ plane coordinates and $(X'_1 X'_2 X'_3)$ corresponds to the other plane coordinates, which is X'Y'Z'.

In the following example, the homogenous matrix H can be determined by using the correspondences $(x,y) \rightarrow (x',y')$ for **four** points, shown in Figure 5.24. Since the transformation is carried out in 2D, the values for X_3 and X'_3 are taken as 1 (Reid, 2003).

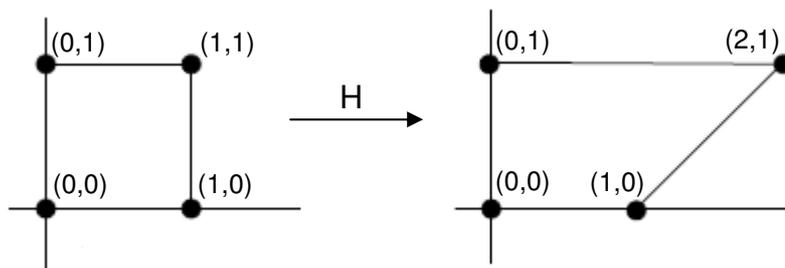


Figure 5.24. For a 2D projective transformation example, the correspondences of four points.

For the first correspondence $(0,0) \rightarrow (0,0)$

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

yields $h_{13} = h_{23} = 0$;

The second correspondence $(1,0) \rightarrow (1,0)$

$$\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & 0 \\ h_{21} & h_{22} & 0 \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

yields $h_{21} = 0$ and $h_{11} = h_{31} + h_{33}$;

The third correspondence $(0,1) \rightarrow (0,1)$

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{bmatrix} h_{31} + h_{33} & h_{12} & 0 \\ 0 & h_{22} & 0 \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

yields $h_{12} = 0$ and $h_{22} = h_{32} + h_{33}$;

The fourth correspondence $(1,1) \rightarrow (2,1)$

$$\begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = \begin{bmatrix} h_{31} + h_{33} & 0 & 0 \\ 0 & h_{32} + h_{33} & 0 \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

yields $h_{31} + h_{33} = 2$, $h_{32} + h_{33} = 1$ and $h_{31} + h_{32} + h_{33} = 1$

By solving the equations using the matrix elements only, H can be found as

$$H = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 2 \end{bmatrix}$$

In the present case, the final positions of the facade corners correspond to points on \mathbf{xy} plane, while the aligned positions refer to $\mathbf{x'y'}$. Therefore, to compute the homogenous matrix shown in the previous example, a similar transformation can be carried out.

The projective transformation is carried out by mapping the trend lines onto newly determined aligned positions that are parallel to vertical axis. To do that the angle between a trend line and the vertical axis (colored in green) is calculated and the aligned positions are computed using this angle (Figure 5.25(a)). For the horizontal direction, the imaginary dashed lines (colored in red) are drawn between the upper and lower end-point pairs. Then, the angles between the imaginary lines and the horizontal axes (colored in blue) are calculated and the aligned positions $((X_1', Y_1'), (X_2', Y_2'), (X_3', Y_3'), (X_4', Y_4'))$ are computed (Figure 5.25(b)).

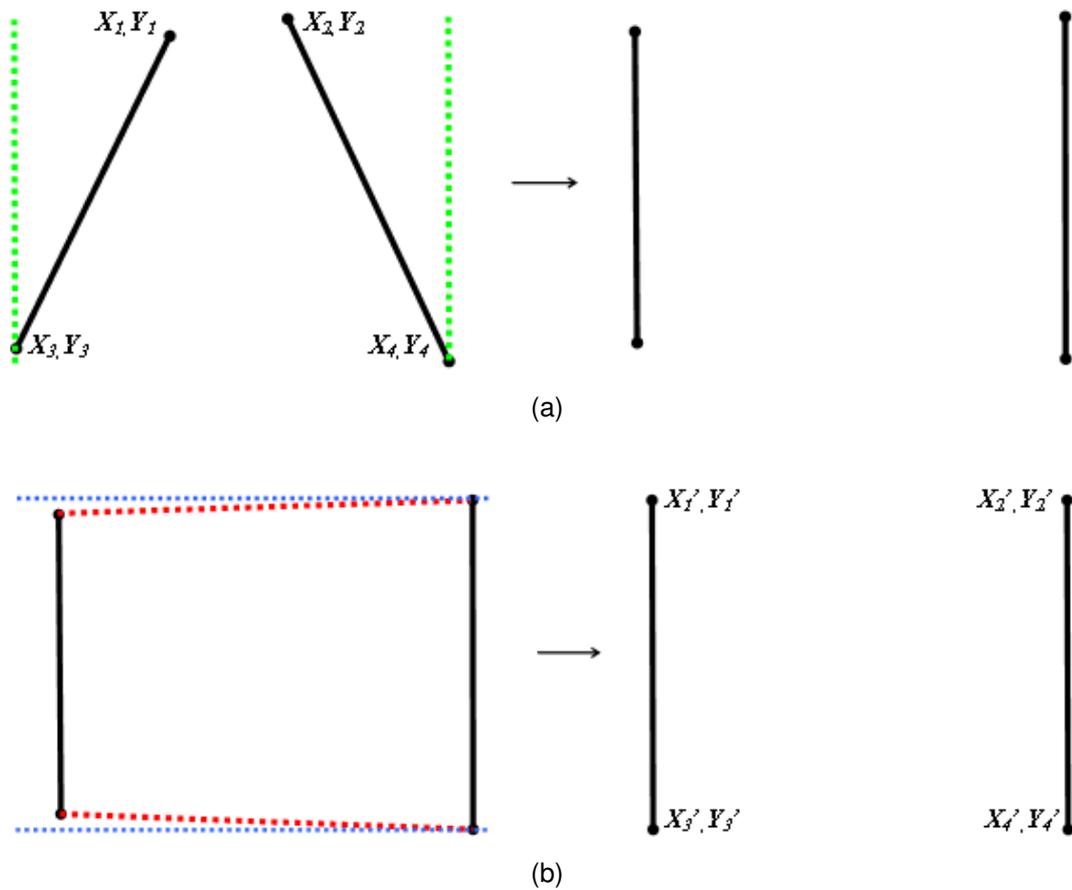


Figure 5.25. The alignment operation with respect to (a) vertical axis and (b) horizontal axis.

After finding the locations of the end-points, the projective transformation is carried out using the MATLAB image processing toolbox functions. Then, the resulting image is resampled using the bicubic interpolation method that involves the sixteen nearest neighbors of a point. In this technique, the intensity value assigned to a point (x,y) is obtained using the equation:

$$v(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j \quad (\text{Equation - 5})$$

where, the sixteen coefficients (a_{ij}) are computed from sixteen equations in sixteen unknowns (Gonzalez and Woods, 2008). By applying the proposed geometric rectification procedure, the perspective distortions inherent in the building facade image are reduced to a reasonable level. For the facade image of a building selected from the study area, the extracted building facades before and after geometric rectification are illustrated in Figure 5.26.

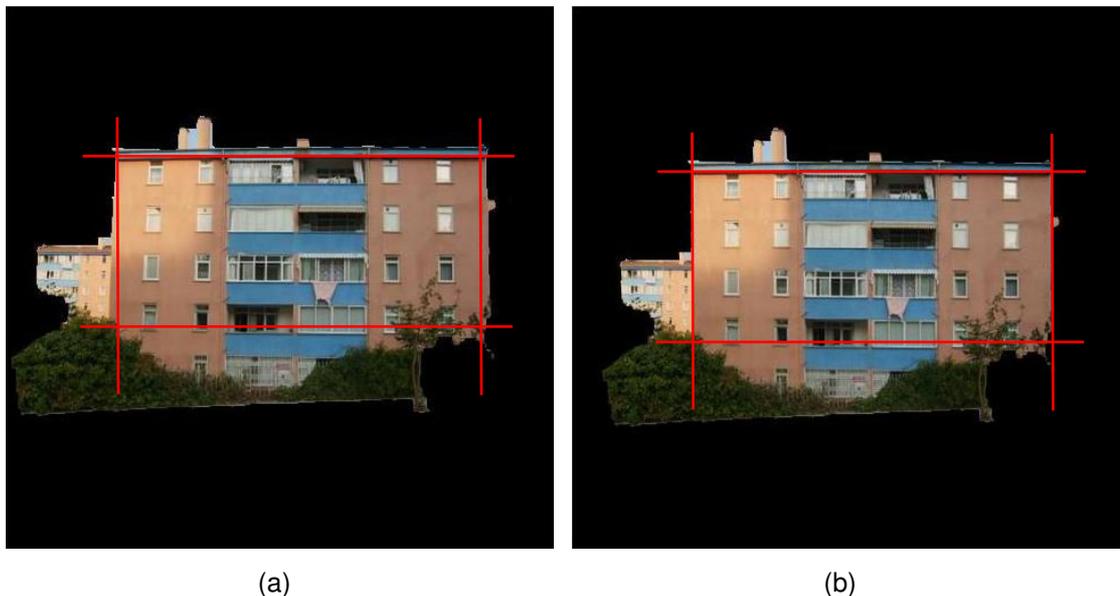


Figure 5.26. The extracted building facades (a) before geometric rectification and (b) after geometric rectification with the horizontal and vertical reference lines superimposed.

In the developed PREBUM software, the rectification process is carried out using the “Rectification and Cropping” option under the menu item of “Texture Generation”. The detected edges, vertical Hough lines, the detected

end-points, and the trend lines are demonstrated in a separate popup window. The facade image after performing the rectification is stored in a file named “facade_rectified_image.jpg” and recorded on the “Layers” panel. As in the previous section, the identification suffixes are appended at the end of the file names. The screenshot of the “Layers” panel after performing the rectification of the facade image is given in Figure 5.27, in which the file names that are circled in red correspond to adjacent facades of a single building.

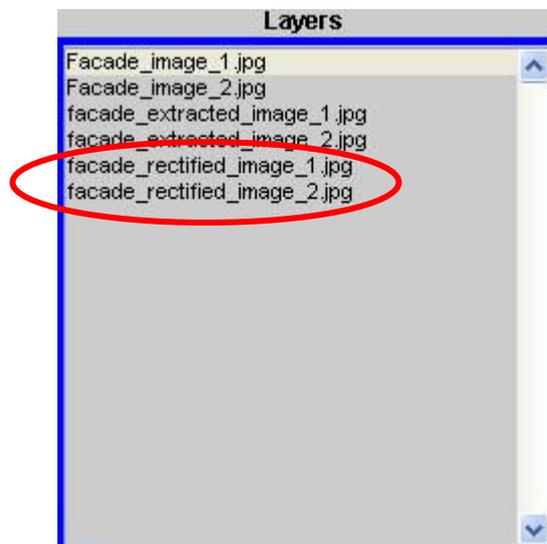


Figure 5.27. A screenshot of "Layers" panel after the rectification process of the facade image.

5.3.4. The Accuracy Assessment of Facade Image Rectification

After performing the geometric rectification on the extracted facade images, the results are evaluated using the same quantitative metric described in section 4.3.3. For each facade image, the corresponding reference data is prepared by performing the geometric rectification manually using the GIMP-2 image processing software. Then, four test points are determined both on the reference image and the rectified image. The locations of the test points (red color) on reference image and the test points (green color) on rectified image are illustrated in Figure 5.28.



Figure 5.28. The test points on (a) the reference and (b) the rectified images.

As described in section 4.3.3, the pixel distance errors (*DistErr*) are computed for the specified (x,y) pairs. In addition, the accuracies are computed in a similar way at the confidence levels 80%, 85%, 90% and 95% with the corresponding Z scores of **0.84**, **1.045**, **1.28** and **1.645**. For a sample image, the pixel errors for each test point and the accuracies under different confidence levels are given in Tables 5.1 and 5.2. Note that the standard deviation (*SD_DistErr*) and the mean of the distance error (*Mean_DistErr*) are computed to be **0.68** and **3.83**.

Table 5.1. For a sample image, the pixel errors for the test points 1 to 4.

TEST POINTS	X _R	Y _R	X _P	Y _P	DistErr
1	100	136	103	139	4,24
2	67	272	70	275	4,24
3	417	295	415	297	2,83
4	420	138	420	142	4,00

Table 5.2. For a sample image, the accuracies under different confidence levels.

CONFIDENCE LEVEL	Z_VALUE	ACCURACY
80%	0,84	4,40
85%	1,045	4,54
90%	1,28	4,69
95%	1,645	4,94

5.4. Occlusion Removal

Before the acquisition of the final building facade textures, the last step is the removal of the occlusions. An occlusion can be defined as a foreground object that partially or completely obstructs an object standing in the background. These objects are generally composed of vegetation, such as trees, bushes, etc. However, other sort of occlusions can also be encountered rarely, such as lamp posts, parked vehicles or pedestrians. The workflow of the developed occlusion removal procedure is illustrated in Figure 5.29.

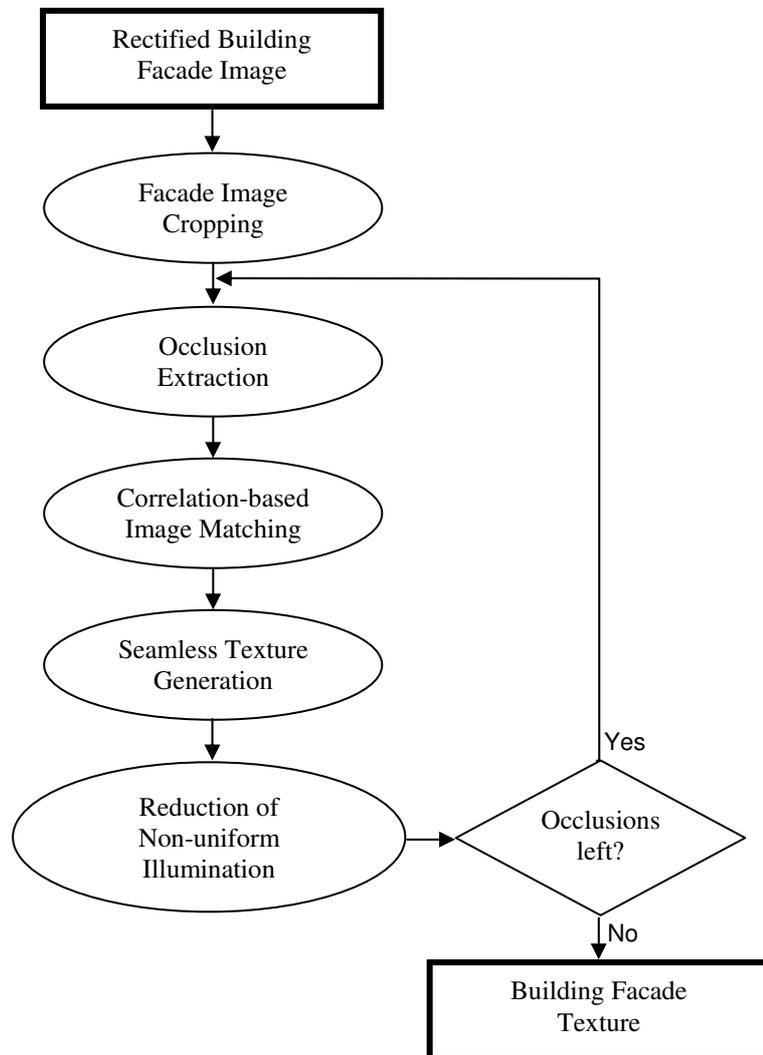


Figure 5.29. The steps of the proposed occlusion removal procedure.

The first step of the developed occlusion removal procedure is the automatic cropping of the rectified facade images. This step is necessary since the current facade images may still suffer from the over-segmentation problem. Therefore, the remaining background regions are completely removed from the facade image. Then, the occluded area is marked on the facade image interactively by means of defining the upper-lower and the left-right margins. Next, a correlation-based image matching is performed to mend the occluded region. This is followed by two post-processing steps, one dealing with the texture seams and the other reducing the illumination differences between the occluded and the candidate facade image patches. Once the whole steps are completed, the user can stop or restart the same procedure until a satisfactory result is achieved. At the end of this procedure the final product becomes a building facade image to be mapped on 3-d building facades.

5.4.1. Facade Image Cropping

In this pre-processing step, the facade image is separated from the background. To do that, first, the corner points on the rectified facade image are identified using the Harris corner detection algorithm. Then, two corner points, one closest to upper left corner (A) and the other closest to upper right corner (B) of the image, are selected as the upper facade corners (Figure 5.30).

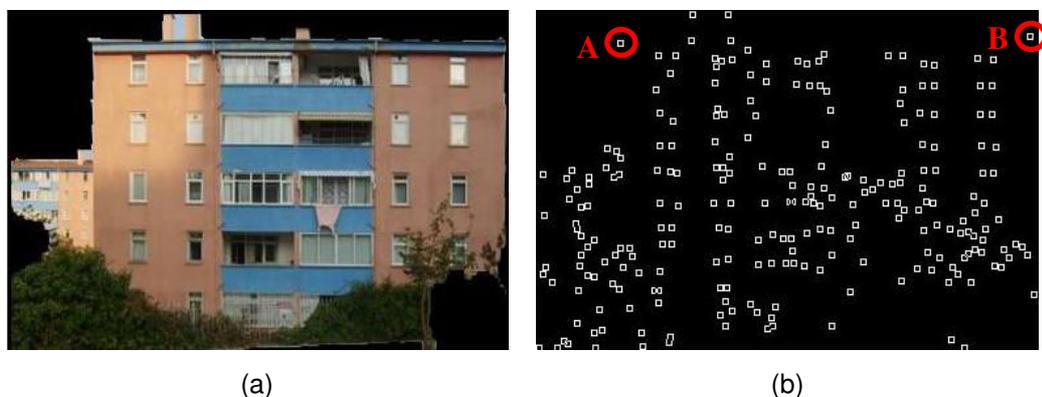


Figure 5.30. (a) The rectified facade image and (b) the set of corner points with the detected upper facade corners (A and B).

By locating these two points, the size of cropping from the sides and the top are determined. For the sides, the vertical lines (green color) that pass through the facade corners determine the cropping limit. On the other hand, a horizontal line (pink color) that passes through the lower facade point (A) identifies the height to be removed from the top. These lines and the resulting cropped image are shown in Figure 5.31.

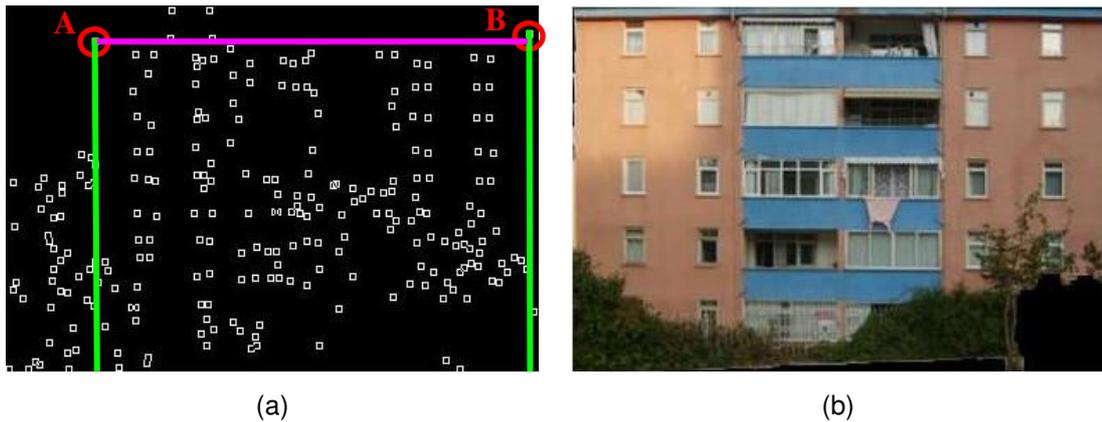


Figure 5.31. (a) The delimiter lines for the sides (green) and the top (pink). (b) The facade image after cropping from the left, right, and the upper sides.

The last and the most troublesome task in finalizing the facade image cropping is to find the cropping limit for the bottom of the image. To do that each row is analyzed one by one starting from the bottom of the RGB image and, for each row, the most frequently occurring brightness value is computed. If this value is (0,0,0) for red, green and blue bands, then the background (black) region is supposed to be dominant and the row is removed from the facade image. On the other hand; if the value is other than (0,0,0), then the row is kept as a foreground information. Further, the remaining background pixels after cropping are filled with the original RGB values. After performing the cropping operation from four sides, the unfilled and the filled images are illustrated in Figures 5.32(a) and (b), respectively. Note that the filled regions are indicated by red circles in Figure 5.32(a).



Figure 5.32. After performing the cropping operation, (a) the unfilled image and (b) the filled image.

In the developed PREBUM software, the facade image cropping operation is performed using the “Rectification and Cropping” option under the menu item “Texture Generation”. The aforementioned cropping steps are achieved automatically and illustrated on the view panel. The cropped facade image is stored in a file named “facade_rectified_cropped_image.jpg” and recorded on the “Layers” panel. As in the previous two sections, the identification suffixes are also appended at the end of the file names. Figure 5.33 illustrates a screenshot of the “Layers” panel, on which the file names of the cropped facade images are circled in red.

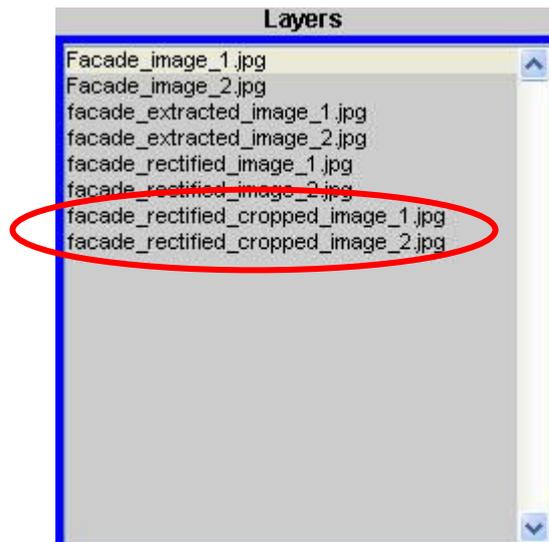


Figure 5.33. A screenshot of the “Layers” panel after performing the facade texture cropping operation.

5.4.2. Occlusion Extraction and Texture Mending

After the cropping operation, the next step is the extraction of the occlusions. In the present study, the occlusion extraction is performed by employing a semi-automated technique aiming to reveal the occluded regions of the facade image. To do that the user is enabled to draw a quadrilateral that encloses the occlusions. The patch enclosed by a red dashed box shown in Figure 5.34(a) represents the occluded area on the facade image. Note that, the selected quadrilateral region also contains non-occluded texture that belongs to building facade. This is because of the fact that inclusion of non-occluded texture increases the chance of finding a highly correlated patch to be replaced with the occluded patch.



Figure 5.34. (a) The selection of an occluded region that is enclosed by a dashed red rectangle. (b) The cropped patch of the occluded area.

In order to reveal the occlusions, an occlusion threshold (ot) is determined by Otsu's method (Otsu, 1979), which is based on minimizing the intra-class variance in determining a global threshold value. Next, the patch is segmented into two classes, one (the black areas) representing the occlusions and the other (the white areas) representing the building regions (Figure 5.35). As can be seen in the figure, several parts that belong to building facade are segmented as occlusions. However, the majority of the building facade elements including windows, balconies, and the wall texture are recovered successfully.



Figure 5.35. (a) The segmented image, in which the black pixels represent the occlusions and (b) the image patch with the occluded areas (black) superimposed.

The extraction of the occlusions is followed by the texture mending procedure, which is based on the image matching approach. In this procedure, the most similar patch to selected patch (all but itself) is searched within the cropped facade image. For measuring the similarity, the correlation metric is used for the non-occluded (segmented as white) regions only. The correlation coefficient (r) can be computed as follows:

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (\text{Equation - 6})$$

where; A_{mn} and B_{mn} are the brightness values for the matrices A and B at the coordinates (m,n) , \bar{A} and \bar{B} are the mean values of the matrices A and B , respectively.

Since the correlation coefficient is computed for a single image band it is applied to R-G-B bands separately, therefore. Then, the average value of the three is computed and accepted as the correlation value. Finally, the patch having the highest correlation with the occluded patch (candidate patch) is detected and fused to cropped facade image (Figure 5.36).

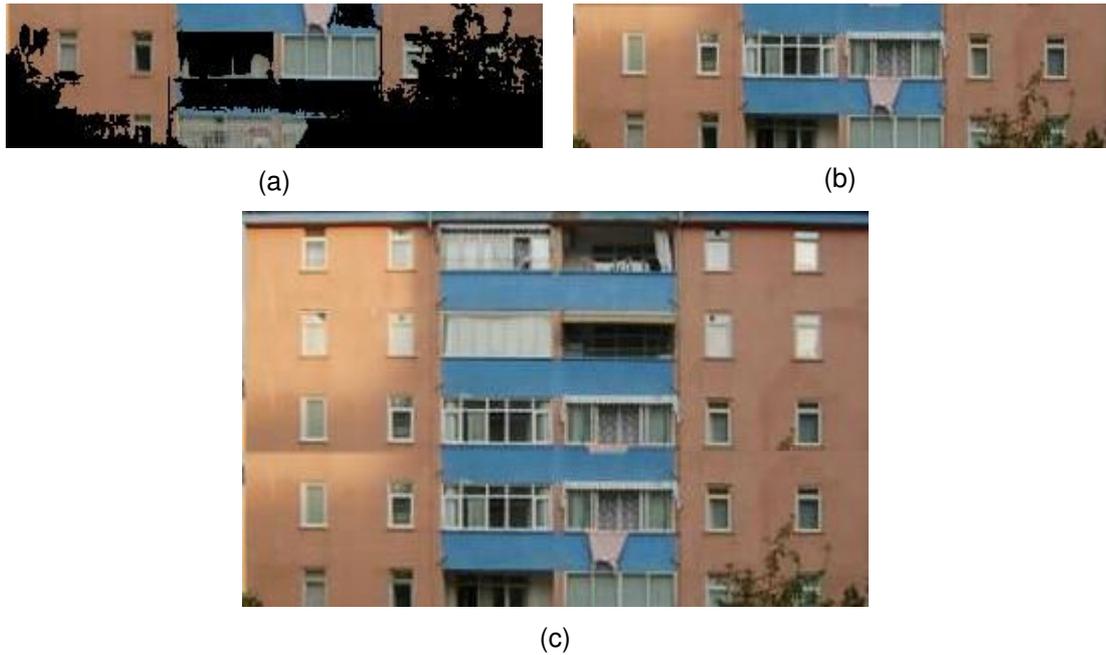


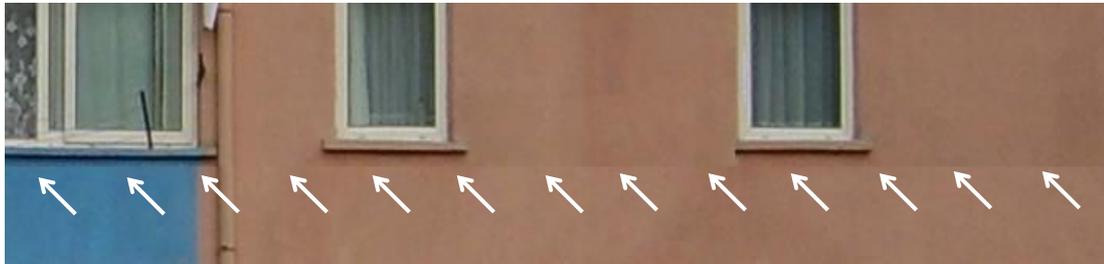
Figure 5.36. (a) The occluded patch, (b) the detected candidate patch, and (c) the mended facade image after fusing the patch detected in (b).

5.4.3. Post-processing

Although the occlusions are eliminated from the facade image, two more issues remain to be handled before the acquisition of the final building facade texture. The first issue is the sharp transitions (seams) at the borders of the fused patches. This problem can be solved by applying a motion blurring filter, which creates a movement blur. This filter is capable of linear, radial, and zoom movements. The blurring size and direction can be altered by adjusting the length and angle parameters. In this study, the linear type of filter was used so that the blurring occurs in a single direction either horizontally or vertically. On the other hand, the value for the parameter motion blurring length (mbl), which represents the blur intensity, was kept as 5. Moreover, the values for the motion blurring angle (mba) were taken to be 90° for the horizontal and 0° for the vertical seams.

The filter is applied within the three pixel wide buffer zone which is generated around the seam instead of applying it to whole image. In general motion

blurring has negative effects in many image processing applications. In the present case however, it is utilized for generating the seamless textures. A small section of a sample building facade image before and after the motion blurring is illustrated in Figure 5.37.



(a)



(b)

Figure 5.37. (a) A sample building facade image patch with a seam pointed by the white arrows and (b) the reduced seam after applying the motion blurring in vertical direction.

The second problem is the non-uniformity of the illumination such that the candidate patch may have different illumination characteristics with respect to neighborhood (Figure 5.38).



Figure 5.38. The abrupt illumination change (within red circle) on the building facade image.

To reduce this effect, a couple of operations that are image differencing and the weighted averaging are carried out. First, a difference image is calculated by subtracting the occluded patch (Figure 5.36(a)) from the candidate patch (Figure 5.36(b)) after converting both to grayscale. Then, a threshold (image difference threshold - idt) is defined to determine the pixels of the occluded patch to be assigned to candidate patch. The assignment is performed based on the following condition: “if the pixel value in difference image is less than “ idt ” then, assign the corresponding pixel in the occluded patch to same position in the candidate patch”. Otherwise, the original pixel value in the candidate patch remains unchanged. In the present case the value for “ idt ” was defined as 50, above which the difference is considered to be a structural difference rather than a radiometric difference. The difference image and the updated candidate patch are illustrated in Figure 5.39.



Figure 5.39. (a) The grayscale difference image, in which dark areas refer to minor or no change and (b) the updated candidate image.

Although the level of illumination seemed to be normalized considerably, the noisy pixels may arise. Therefore, the noisy pixels are removed using the weighted image averaging method. To do that, the candidate and the updated candidate images are summed up with the equal weights of 0.5. By doing this, the balance between the noise level and the non-uniformity of the illumination are preserved to a certain extent. The resulting image patch after removing the noisy pixels is illustrated in Figure 5.40.



Figure 5.40. The image patch produced after applying the weighted averaging.

As mentioned earlier, the extraction of the occlusions is carried out using a developed semi-automatic procedure. Therefore, the user can stop the execution if the results are satisfactory or can resume the process to obtain better results. It should be remembered that removing the occlusions entirely may not be possible in the first iteration. As can be seen in Figure 5.40, upon performing the first iteration, occlusions (a branch of a tree) still remain in the lower right corner of the facade image. Therefore, to remove the remaining occlusions the same steps are repeated and the final building facade texture is generated as shown in Figure 5.41. The building facade image before applying any post-processing operation is illustrated in Figure 5.42, in which the texture seams and the non-uniform illumination are quite noticeable.



Figure 5.41. The final building facade image after applying the post-processing steps.



Figure 5.42. The building facade image before applying any post-processing.

The occlusion removal in the developed PREBUM software is performed by choosing the “Occlusion Removal” option under the menu item “Texture Generation”. The selection of the occluded quadrilateral region is carried out by the user. During the matching process a wait bar is visualized along with a message showing the instantaneous maximum correlation rate (Figure 5.43).

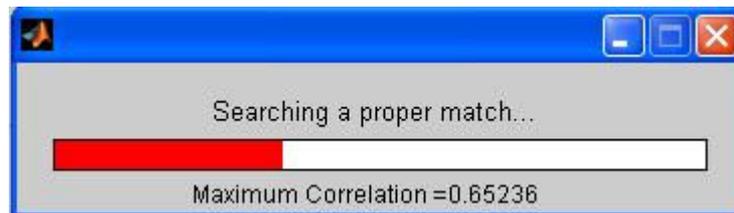


Figure 5.43. A wait bar that shows the correlation rate.

At the end of each occlusion removal cycle, a decision is asked via a question dialog box (Figure 5.44(a)). Therefore, the user can either stop the execution or resume the occlusion removal procedure to achieve better results. After terminating the procedure, the cleared facade image is stored in a file named “facade_cleared_image.jpg” and recorded on the “Layers” panel. The identification suffixes are appended at the end of the file names. A screenshot of the “Layers” panel after applying the occlusion removal procedure is shown in Figure 5.44(b), in which the file names (circled in red) correspond to cleared facade images (final textures).

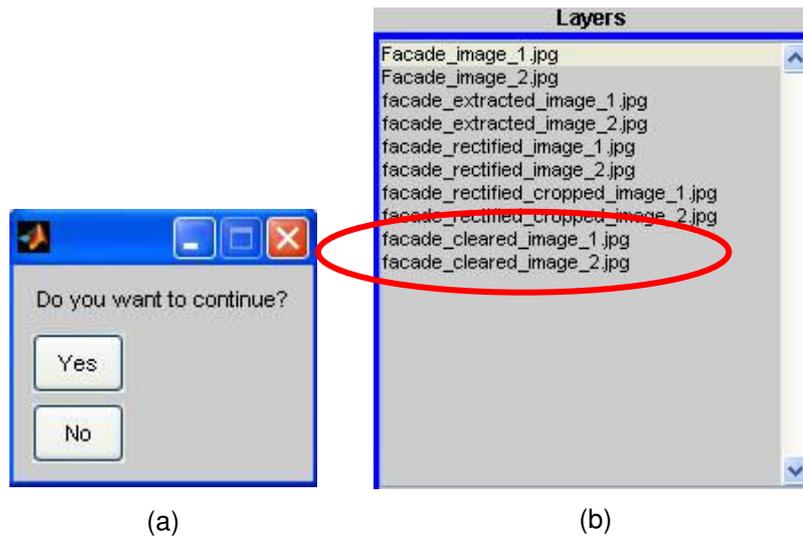


Figure 5.44. (a) The question dialog box appeared after each occlusion removal cycle and (b) the screenshot of the “Layers” panel after performing the occlusion removal procedure.

5.4.4 The Accuracy Assessment of Occlusion Removal

After applying the occlusion removal procedure, the results were assessed by a subjective metric. Although the objective methods offer a simple and convenient way, the subjective assessment of the final image quality is found to be more appropriate. The assessment was carried out by presenting an occlusion-free image to a cross-section of viewers and averaging their evaluations. In the present case, the evaluations were performed using an absolute rating scale, illustrated in Table 5.3. The rating values change between 1 and 6, in which 1 is categorized as “Excellent”, while 6 corresponds to “Unusable”. After all, the final ratings of each facade texture were computed by averaging the individual ratings of each viewer.

Table 5.3. Rating scale of the Television Allocations Study Organization (Frendendall and Behrend, 1960).

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

CHAPTER 6

3-D MODEL GENERATION AND TEXTURE MAPPING

In this chapter, the integration of 3-d building models with the facade images is explained. First, the generation of the untextured 3-d solid model is described. The 3-d solid models are generated by means of integrating 2-d footprint information and the normalized digital surface model (nDSM). Then, a texture selection and mapping procedure is employed to construct the textured model. To improve the level of reality the synthetic roof models and textures are also utilized. In the last section, the example models of various building blocks are presented.

6.1. Generating the Solid Building Models

The solid building models are generated from both the quadrilateral shaped 2-d boundary information and the height information. Quadrilateral is a polygon with four edges (sides) and four vertices (corners), which is defined in Euclidean plane geometry. In this study, the square and rectangle shaped buildings are analyzed. An example building boundary (green) with the numbered corner points (yellow) and the mid-points of the edges (red) are illustrated in Figure 6.1.

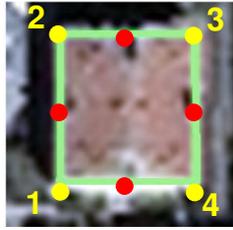


Figure 6.1. The boundaries of a quadrilateral shaped building.

The pixel coordinates (x,y) of the corners are automatically extracted using the approach described in chapter 4. The geographic coordinates in terms of easting and northing are also computed from the satellite image, where easting refers to the eastward-measured distance and northing refers to northward-measured distance in meters. For a sample building, the list of pixel coordinates and the geographic coordinates for four corners together with the building identification number (ID) and the building block ID information are illustrated in Table 6.1.

Table 6.1. For a sample building, the corner coordinates and the building ID information.

Building Block ID	Building ID	Building Corner ID	Pixel Coordinate for X	Pixel Coordinate for Y	Easting (meters)	Northing (meters)
3	1	1	581	594	477067	4425613
		2	581	572	477067	4425635
		3	601	572	477087	4425635
		4	601	594	477087	4425613

As can be seen in the table, the sample building belongs to third block (Block ID=3) and is labeled as the first building (Building ID=1). It is evident from the corner coordinates that the building has a rectangular shape with the dimensions of 20x22 m that is computed from the absolute difference between the easting ($|477067 - 477087| = 20$ m) and the northing ($|4425613 - 4425635| = 22$ m).

After computing the size, the next step is to determine the location of the building in the model. In order to provide the correct position of the buildings, the translation of each building must be carried out on the x-y plane.

Translating a set of points means adding a predetermined pair of constants (X,Y) to Cartesian coordinates for every point in the set. If the original coordinates of a point are (x, y) then, the translation is carried out as follows (Equation 1):

$$(x', y') = (x + X, y + Y) \quad (\text{Equation - 1})$$

where, (x', y') is the translated coordinates. This is illustrated in Figure 6.2, in which the corner coordinates of a building (red color) are shifted on the base map with respect to pixel coordinates based on the information given in Table 6.1.



Figure 6.2. The translation of a building (red) on a base map.

Next, the rotation on 2-d plane is carried out. The information for rotation is derived from the orientation parameter, which was discussed in chapter 4.

The orientations of the buildings were already computed during the extraction of 2-d footprints. Buildings with various orientations are presented in Figure 6.3.

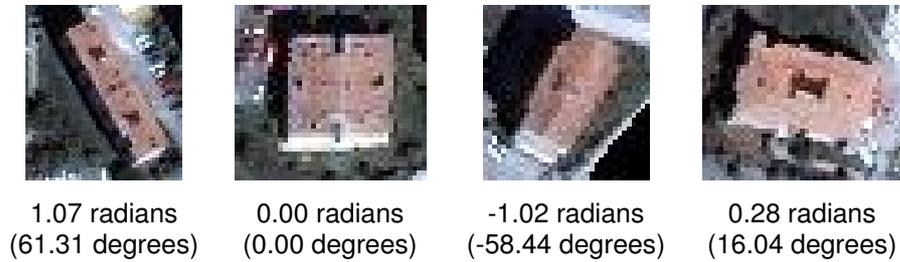


Figure 6.3. The buildings with different orientations.

After obtaining the orientation information, the 2-d characteristics of the buildings are determined. Then, the building heights are estimated from nDSM and the 2-d quadrilaterals are extruded along the “z” axis. To do that the centroid coordinates of the buildings are used. These coordinates are computed from the intersecting point of the minor and major axes of the best fitting ellipse which was explained in chapter 4. For a building, the centroid point (blue cross) and its projection onto corresponding nDSM patch are presented in Figure 6.4.

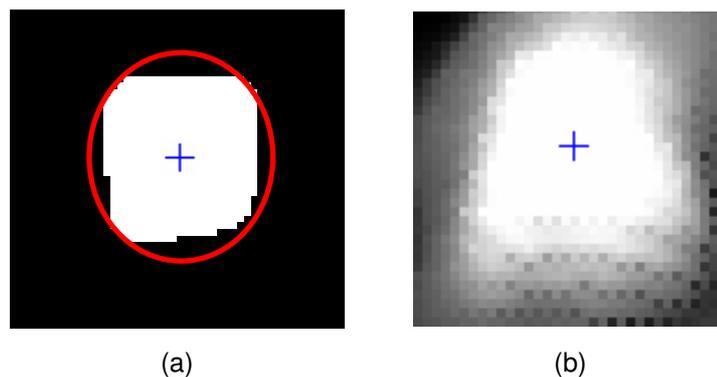


Figure 6.4. (a) The center point (blue cross) of a binary building patch and (b) the corresponding nDSM patch.

Next, a 7x7 buffer zone from the center is generated to capture the height information. The size of the zone is a default value specified by the PCI Geomatica software. The values falling within the buffer zone are rounded to

the nearest real numbers with one decimal place. This is followed by the computation of the majority height value. The advantage of using the majority value is that it avoids the selection of the extreme values in determining the building heights. The computed majority value is accepted as the estimated height of the building. In case the majority value is more than one, either is selected arbitrarily. For a sample building, the captured height values and the frequencies after rounding are illustrated in Figure 6.5. In the given example, the most occurring (with 17 times) height value of 14.7 m was selected as the building height.

	4224	4225	4226	4227	4228	4229	4230
6060	14.3849	14.5138	14.6319	14.6968	14.6844	14.6171	14.3165
6061	14.4501	14.7687	14.7072	14.9858	14.7648	14.9012	14.3983
6062	14.5587	14.6561	14.8148	14.8425	14.9109	14.7943	14.7083
6063	14.5662	14.8108	14.806	14.967	14.8807	14.999	14.6861
6064	14.6	14.6906	14.7891	14.8173	14.8717	14.7772	14.7185
6065	14.6695	14.6772	14.6978	14.777	14.7458	14.715	14.5232
6066	14.5465	14.6359	14.6851	14.7176	14.724	14.6621	14.5416

(a)

Rounded height values (m)	Frequency
14.3	1
14.4	2
14.5	5
14.6	6
14.7	17
14.8	11
14.9	4
15.0	3
TOTAL	49

(b)

Figure 6.5. (a) The height values falling within the 7x7 buffer zone. (b) The frequencies of the rounded height values and the selected height value of 14.7 m.

After computing the estimated height value, the solid model becomes ready to construct. For creating the model, a crucial assumption is hypothesized and therefore, a generic size is determined for each building in the block. To do that, the x-y-z dimensions are analyzed separately and the most frequently occurring extents are determined and assigned to all buildings in the block. This assumption was found to be plausible since all buildings in a block have identical dimensions very often in reality. In this manner, the errors due to small variations in size are eliminated. The untextured 3-d model of a building block (Block-3) is presented in Figure 6.6. Note that the roof shapes were assumed to be flat for this moment.

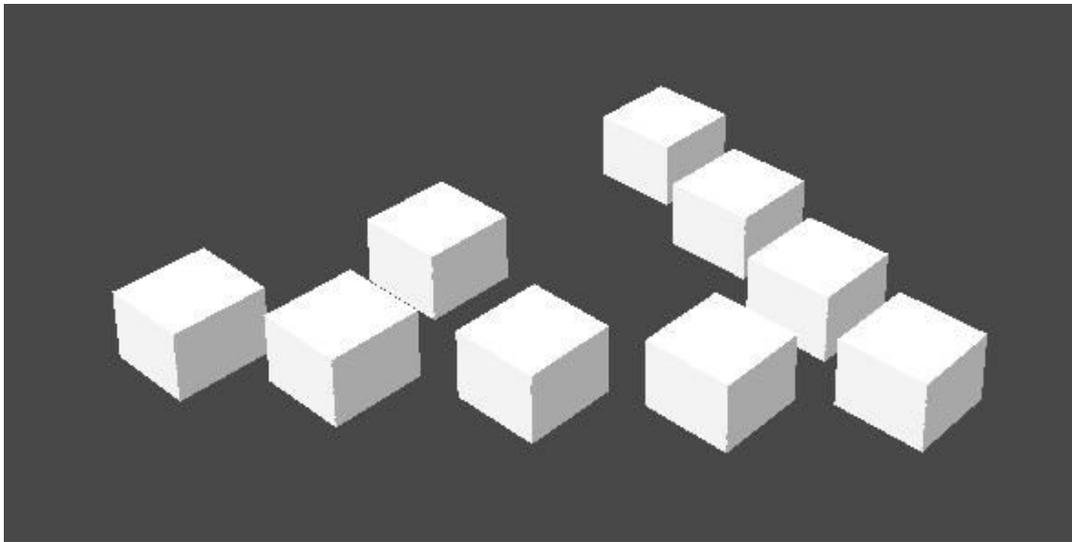
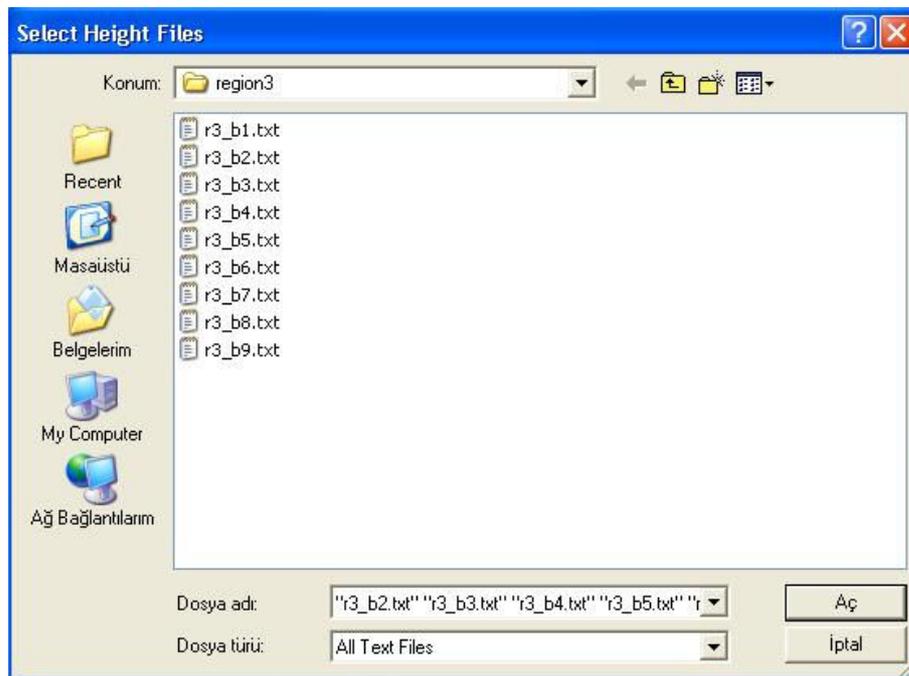


Figure 6.6. The automatically generated 3-d models in a building block (Block-3).

In the developed PREBUM software, the 3-d solid models are generated using the “Solid Modeling” menu item. First, for the block being considered, the heights of the buildings are initiated using the “Input 3-D Height Information” option. The current block information is retrieved automatically and the related height files are listed. The name of a height file consists of a block id along with a building id. For example, the height file, “r3_b1.txt”, keeps the rounded height values of building # 1 in block # 3. The extension is chosen as “.txt”, which is easily recognized by many programming

languages. Figure 6.7 illustrates the dialog box which includes the selected height files and a warning message after the user selection.

Then, the height information is integrated with the 2-d coordinates by using the menu option of “Generate 3-D Models”. And, with the extrusion process the 3-d solid model is generated. This process ends up with a warning message that informs the user (Figure 6.8). The last option “Show 3-D Solid Models” is initiated to visualize the building models using the Cortona viewer.



(a)



(b)

Figure 6.7. (a) The selection dialog box for the height files and (b) the subsequent warning message.



Figure 6.8. The warning message after the generation of 3-D solid models.

6.2. Texture Selection and Mapping

After generating 3-d building models, the next step is to map the extracted textures on the building facades. The workflow of the developed texture selection and mapping procedure is illustrated in Figure 6.9. The first step is the labeling of the edges of the buildings falling within the selected block. Then, two photos are selected from database with respect to proximity and orientation. This is followed by the extraction, rectification, and occlusion removal of the images, discussed in chapter 5. Finally, the texture mapping procedure is carried out.

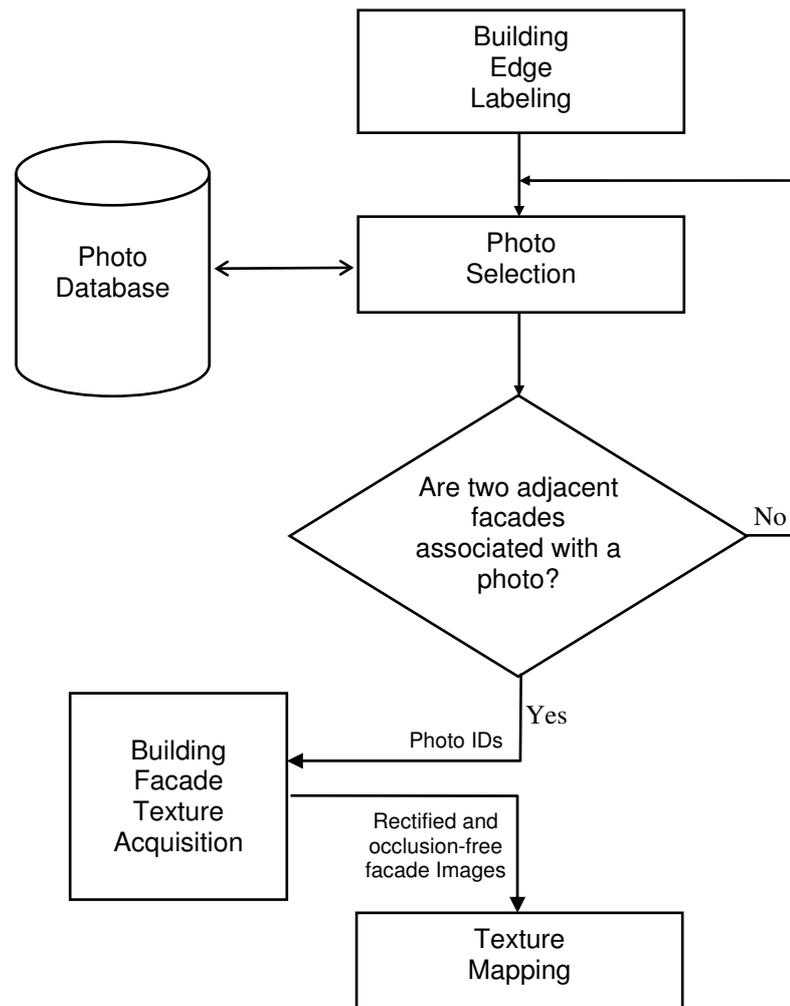


Figure 6.9. The workflow for the texture selection and mapping.

In the edge labeling step, the shapes of the buildings are analyzed using the 2-d information. In the present case, the shape is considered to be rectangle or square. For the rectangle shaped buildings, two different labels, such as “a” and “b” are assigned to adjacent edges. And, the similar labeling is repeated for the opposing edges. For the square shaped buildings, the labeling of the edges using the labels “a” and “b” is carried out arbitrarily. The proposed automated labeling process for each building is shown in Figure 6.10. Note that within a selected building block, associating the correct textures with the corresponding facades of the buildings is an important step. This is based on the assumption that all the buildings falling within a block have identical sizes and therefore identical opposing facade textures. As an exceptional case; if the buildings that fall within a block are square in shape

and they have different orientations, then the labeling operation can be performed manually.

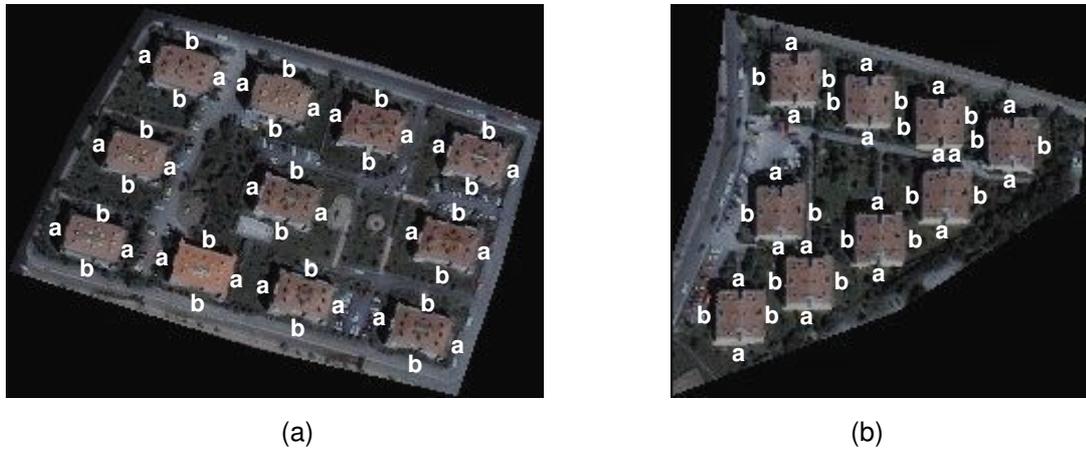


Figure 6.10. The building blocks that contain (a) the rectangular shape buildings and (b) the square shaped buildings with the labels assigned to edges.

Next, the corresponding facade photographs (ground-level photographs) are selected from the photo database. As mentioned earlier, 15 photos were taken in the Batikent district in a clear-sky day using Samsung WB500 digital camera. For each building block, at least two photos were taken from the points having suitable shooting conditions, such as sufficient angle of view and reasonable amount of occlusions. The reason for taking at least two photos is originated from the assumption that each quadrilateral building has only two opposing facades and therefore, the opposed facades are mapped with the same texture. The number of photos taken for each building block is given in Table 6.2. The photos taken for Block - 7 were excluded from the database since the building facade structure in this block was quite complex and therefore, the proposed building facade texture extraction approach would fail, accordingly.

Table 6.2. The number of photos taken for the building blocks 1 – 6, and 8.

Building Block	# of photos taken
1	2
2	3
3	2
4	2
5	2
6	2
8	2
TOTAL	15

In parallel, the coordinates of the photo shooting points were also measured by a handheld GPS receiver. As a general policy, the facades facing to street were photographed because of the accessibility issues. Besides, the shooting points were selected carefully, in which the GPS signals captured were rather strong in these points.

In the study area, the photo shootings and the coordinate measurements for eight building blocks were completed within an hour. For shooting the photos no vehicle was used. Instead, the facades were photographed on foot. The locations of the photo shoots and their directions are illustrated in Figure 6.11.



Figure 6.11. The locations of photo shooting points and the directions for the building blocks considered.

The structure of the photo database is given in Table 6.3, in which four attributes are stored. The first attribute “Photo ID” uniquely identifies each facade photo. The second and third attributes are the earth coordinates of the shooting points in terms of easting and northing. And the fourth attribute is the bearing, which is measured by the built-in compass of the handheld GPS receiver. The actual values of the bearings are rounded to nearest direction according to Figure 6.12.

Table 6.3. The structure of the photo database.

Photo ID	Easting	Northing	GPS bearing (in degrees)
1	476650	4425958	90
2	476672	4425995	90
3	476728	4426027	180
4	476977	4425978	135
5	477012	4425947	0
6	477066	4425793	90
7	477126	4425752	0
8	477101	4425759	180
9	477070	4425729	90
10	477069	4425697	270
11	476952	4425585	0
12	476767	4425703	135
13	476782	4425745	315
14	476768	4425846	180
15	476727	4425735	225

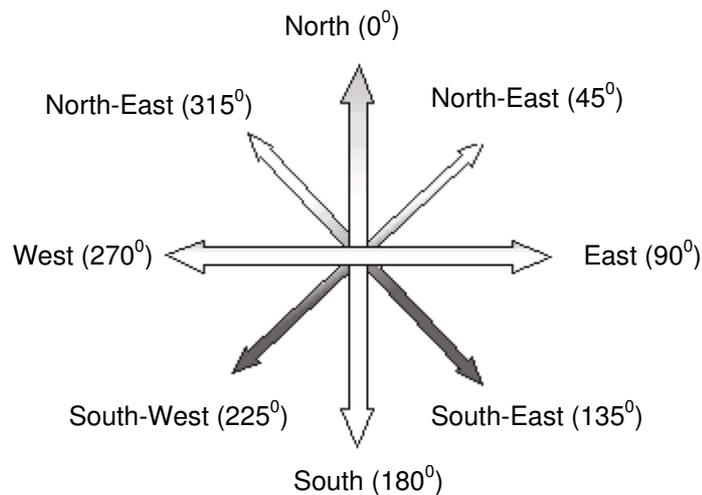


Figure 6.12. The 8-directions with the angles.

The next step is the selection of the corresponding photos. This is carried out by analyzing the proximity and the bearing of the shooting points with respect to facade mid-points. First, for a selected building block, the proximities of all shooting points to facade mid-points are calculated. For the Batikent study area, this computation was performed based on the Euclidean distance between each of the 15 shooting points and the facade mid-points of the

buildings falling within the block. For example, Block-2 contains 11 buildings, for which the number of facade midpoints was 44. Thus, a total of 660 distance values ($15 \times 44 = 660$) must be calculated. Then, these distance values are sorted in the ascending order. Next, starting from the closest, the correspondences between the photos and the facades are analyzed. The photo, for which the computed bearing of the shooting point and the bearing measured by GPS are the same, is selected as a facade texture.

Bearing to a point is the angle measured in degrees in a clockwise direction from the north. Thus, four possibilities arise depending on the four quadrants. To determine the quadrant; the signs of Δx and Δy should be examined. For example, in computation of the angle between A and B; Δy and Δx are found to be $(y_b - y_a) > 0$ and $(x_b - x_a) > 0$, which are in the first quadrant (Figure 6.13).

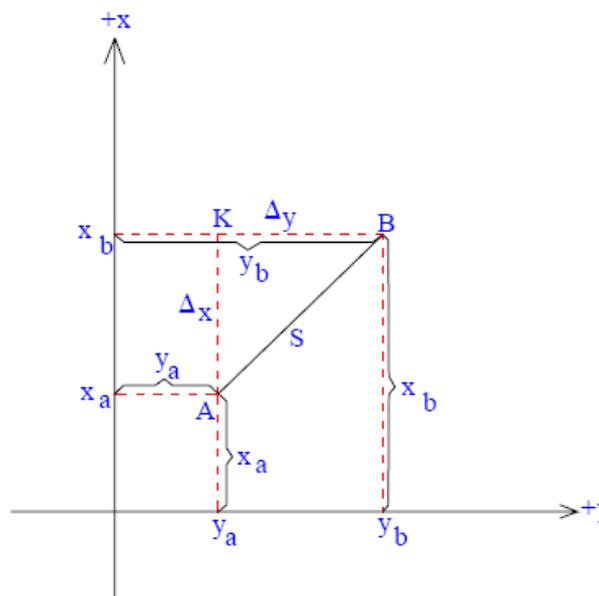


Figure 6.13. The computation of Δy and Δx between A and B.

The complete set of possibilities for the signs of $(\Delta y, \Delta x)$ and the corresponding (AB) bearings are illustrated in Figure 6.14. Note that, the value of bearing can be computed as follows (Equation 2).

$$AB = \arctan\left(\frac{\Delta y}{\Delta x}\right)$$

(Equation – 2)

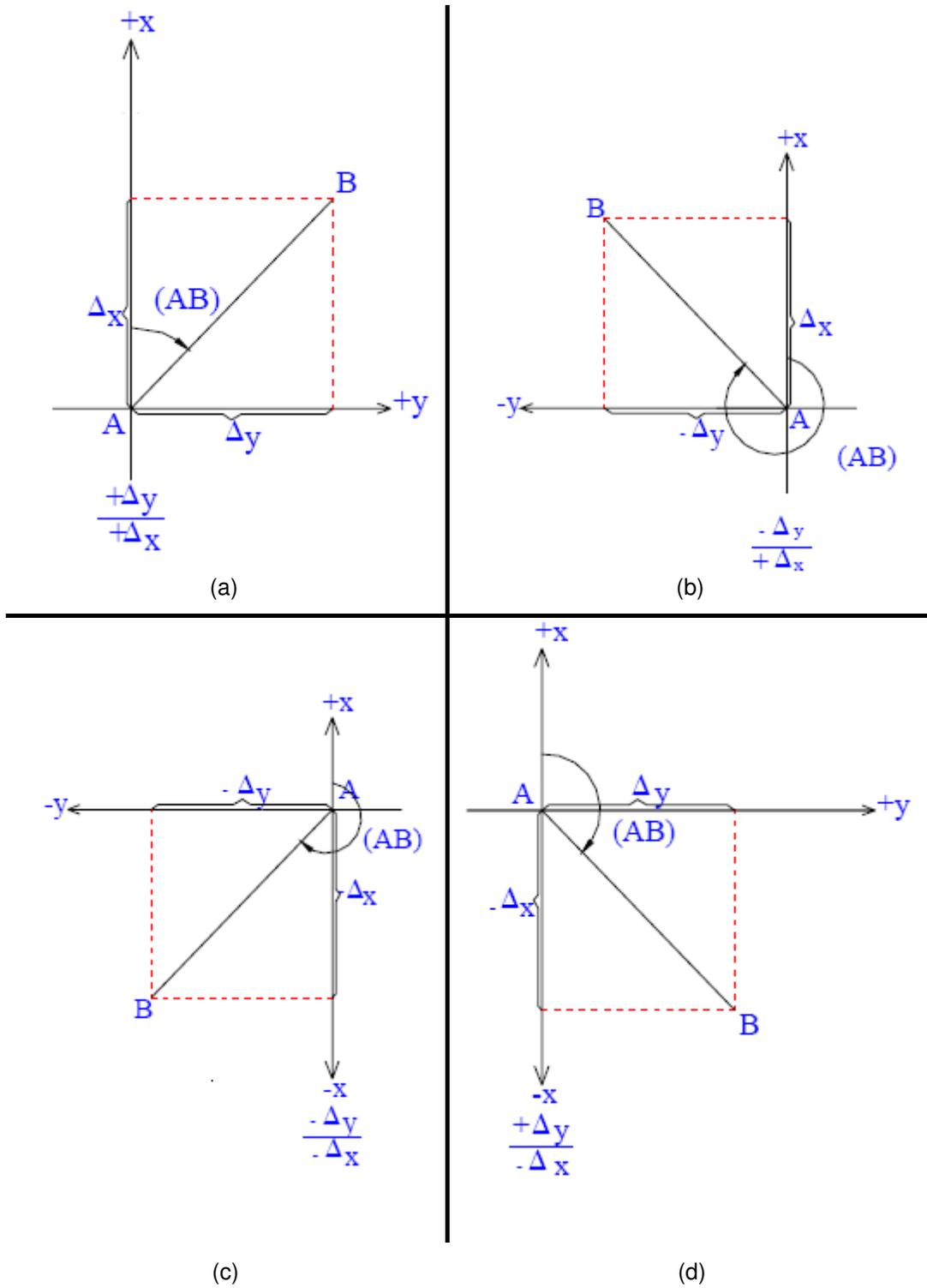


Figure 6.14. (AB) Bearings in the (a) first, (b) second, (c) third, and (d) fourth quadrants.

After computing the bearings, a set of adjustments with respect to quadrants are carried out in order to normalize the computed bearings. To do that, for the second and third quadrants, 180° is added to original bearings. Similarly, 360° is added to bearings in the fourth quadrant.

In the present case, the bearings are calculated with respect to two points. Of these points, the first is the photo shooting point and the other is the facade mid-point falling within the block. As in the GPS bearings the computed angles are rounded to nearest 8-direction. Those photos having the same bearings with the GPS are assigned to corresponding facade as well as to other facades with the same label in the building block being considered. This process is repeated until the texture assignment procedure is completed. To map the facade images for the whole study area, the algorithm is run for each building block falling within the study area. In the present case the algorithm was run seven times, one for each block (except block-7).

The facade texture mapping of the buildings in Block-2 is illustrated in Figure 6.15. In this example, the photo closest to facade mid-point (having the proximity ranking of 1) was analyzed first (Figure 6.15(a)). It was found that photo#1 satisfied the condition of having the same bearing as measured by the GPS. Therefore, the texture of the facade with label "a" in building block-2 was mapped using photo#1. Next, the photo having the proximity ranking of 2 was analyzed (Figure 6.15(b)). The bearing values of the photo and GPS (180° and 0°) did not match and therefore, photo#14 was rejected. It was also realized that photo#14 does not belong to Block-2. Next, the photo in the third place in terms of proximity was analyzed (Figure 6.15(c)). Although there was a match between two bearings, photo#2 was also rejected. This is due to the fact that a photo having the label "a" was already assigned to the facade. In the fourth iteration, the photo with the proximity ranking of four was analyzed (Figure 6.15(d)). In this case, the photo was selected due to the match between two bearings. In addition to that the label of the facade, "b", was not occupied before. The execution of the algorithm is terminated and

the remaining candidate photos are discarded in case two photos are selected for any two adjacent facades (“a” and “b” in Figure 6.10). In the present case, the photos **#1** and **#3** were selected to be used in the texture mapping of the 3-d solid building models belonging to building block #2.



Photo ID: **1**
 Facade Label: **a**
 GPS bearing: **90°**
 Computed bearing: **90°**
 Proximity ranking : **1**

(a)



Photo ID: **14**
 Facade Label: **a**
 GPS bearing: **180°**
 Computed bearing: **0°**
 Proximity ranking : **2**

(b)



Photo ID: **2**
 Facade Label: **a**
 GPS bearing: **90°**
 Computed bearing: **90°**
 Proximity ranking : **3**

(c)



Photo ID: **3**
 Facade Label: **b**
 GPS bearing: **180°**
 Computed bearing: **180°**
 Proximity ranking : **4**

(d)

Figure 6.15. The facade photo assignment procedure for building block-2.

After selecting the photos, the facade textures to be mapped on the solid model are extracted using the proposed method described in chapter 5. In texture mapping, the first step is to calculate the size of the texture to be mapped using the 2-d texture coordinates. Then, the binding of texture to a

face is carried out. In this manner, each corner of the texture piece is bound to a 3-d coordinate on the facade. If necessary, the texture image is stretched to fit the face. The mapping procedure is fully performed in VRML through an automated way. To do that, VRML's "ImageTexture" node is employed by giving the path of the texture image files to "URL" field. For building block-2, the 3-d buildings after mapping the facade textures are shown in Figure 6.16.



Figure 6.16. A screenshot of the textured buildings of building block-2.

To improve the level of reality, the building roofs are also modeled in VRML environment. To do that, the actual roof types of the building blocks were visually investigated from the Google Earth images, which provide higher spatial resolution than the original pan-sharpened imagery. Consequently, two different roof types that are "gable" and "hip" were examined to be used in the study area. The illustrations of the roof types and the corresponding VRML models are presented in Figure 6.17. Besides, the overhanging parts of the roofs were also taken into account to preserve the fidelity, which is shown in Figure 6.16.

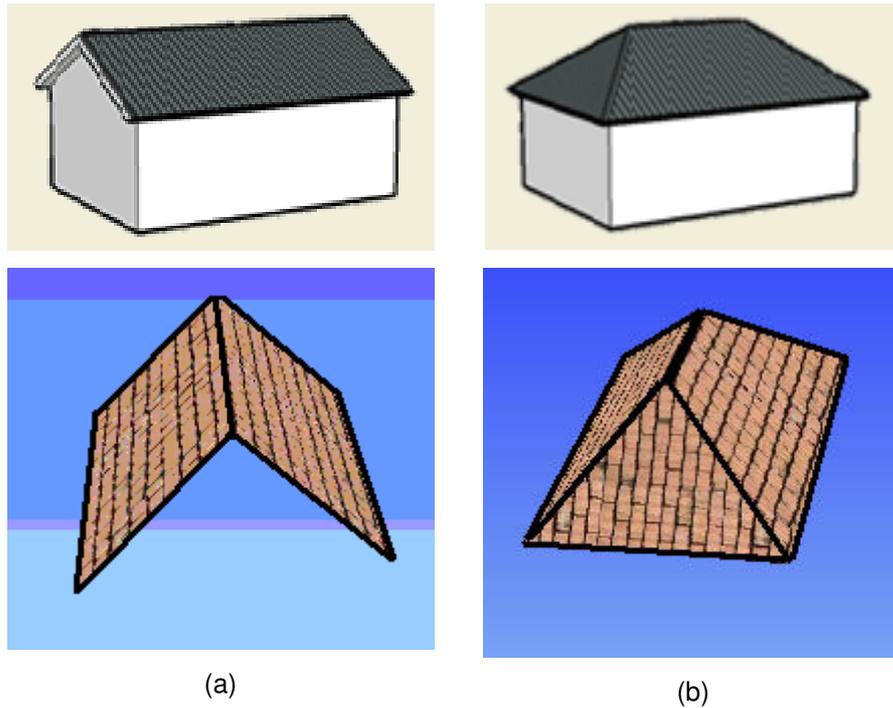


Figure 6.17. The illustrations (top) and the VRML models (bottom) of (a) gable and (b) hip roof types.

For texturing the roof models, a unit pattern (tile) cropped from the high resolution Google Earth imagery is employed. To provide a realistic appearance the selected pattern is repeated all over the roof surface. The unit pattern and the produced roof texture are illustrated in Figures 6.18 (a) and (b), respectively.

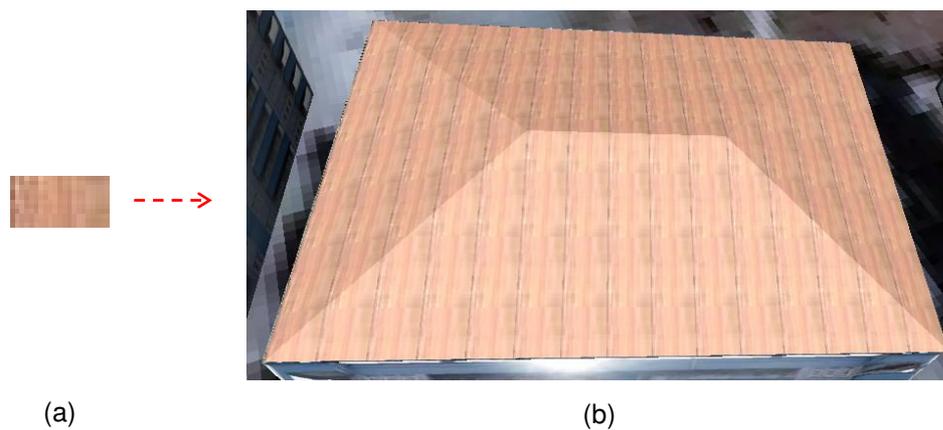


Figure 6.18. (a) The roof pattern used at present study and (b) the resulting texture on a roof.

However, for the gable type of roofs, a gap (indicated by a circle) occurs between the roof and the top of the building facade (Figure 6.19(a)). This gap is filled with a homogenous texture pattern cropped from the facade texture (Figure 6.19(b)).

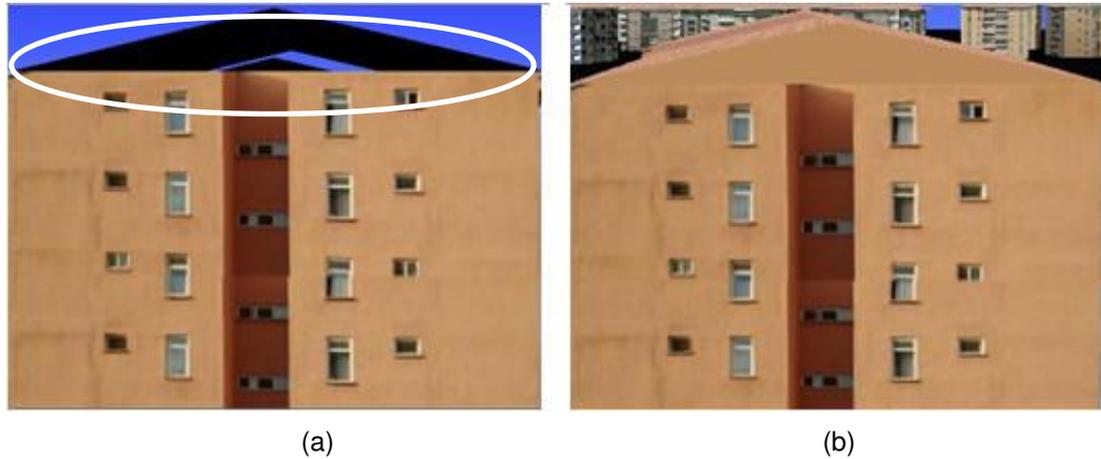


Figure 6.19. The texture filling procedure for the gable-roofed buildings. (a) A building with the unfilled roof texture. (b) The same building with the filled roof texture.

Finally, in the selected area of study, which is composed of eight building blocks, the photorealistic building models were generated. Due to the problems mentioned earlier, the facade textures for those buildings falling within Block-7 were extracted and mapped manually. Two different bird's-eye perspectives of the study area are illustrated in Figure 6.20.



Figure 6.20. The generated photorealistic 3-D buildings of the study area from two different bird's-eye perspectives.

In the developed PREBUM software, the selection and mapping of the facade textures are implemented by employing the menu item of "Photorealistic Modeling". Under this menu item, the first option, "Map Textures", fulfills all the operations mentioned in this section. A warning message (Figure 6.21) appears immediately after the completion of the mapping process. In the second option, "Show 3-D Photorealistic Models", the textured 3-d building models can be visualized using the Cortona viewer.

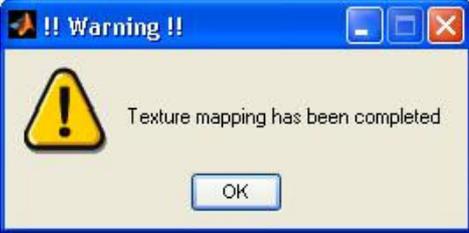


Figure 6.21. The warning message after the generation of the photorealistic 3-d model.

CHAPTER 7

RESULTS AND DISCUSSION

This chapter contains the experimental results of the implementation of the proposed approaches described in chapters 4 and 5. For each approach, the results were generated using the accuracy assessment techniques described earlier. Besides, the time complexities of the algorithms are also reported. In the discussion sections, the shortcomings and the limitations, which may make the proposed approaches fail, are revealed by providing examples for each case. The results and discussions explained in the following sections are grouped into three main categories: (i) results and discussion for 2-d building extraction and delineation, (ii) results and discussion for building facade texture acquisition, and (iii) discussion for 3-d model generation and texture mapping.

7.1. Results and Discussion for 2-D Building Extraction and Delineation

The developed automatic 2-D building extraction approach, described in section 4.2, was implemented in the Batikent study area. The system was run for two different variants by using the predetermined optimum genetic algorithm parameters that are Training/Test Sample Size = 50, Number of Generations = 20, Population Size = 30, Chromosome Size = 5, Crossover Probability = 0.8, and Mutation Probability = 0.2. In the first variant, the adaptive-fuzzy component was excluded, which means that the crossover and the mutation probabilities were kept fixed. For each chromosome

(candidate solution), the fitness values were calculated by using the accuracy assessment method explained in section 4.2.4. After 10 individual runs of the algorithm, the fitness value of **91.86** (averaged over 10 runs) was reached in the last generation. For each run, the computed fitness values and the performance curve are presented in Table 7.1 and Figure 7.1, respectively.

Table 7.1. For each run, the fitness values computed for 20 generations that do not include the adaptive-fuzzy component (the first variant).

Number of Generations	Test -1	Test -2	Test -3	Test -4	Test -5	Test -6	Test -7	Test -8	Test -9	Test -10	Average Fitness
1	88,97	90,91	82,34	90,96	89,22	80,04	88,03	84,03	90,96	87,67	87,31
2	90,56	91,33	82,34	91,00	91,20	81,21	90,01	85,61	91,00	88,87	88,31
3	90,56	91,33	85,74	91,00	91,37	85,39	90,18	87,78	91,00	89,65	89,40
4	91,28	91,33	85,74	91,48	91,49	85,41	90,30	87,85	91,48	89,88	89,62
5	91,32	91,33	85,74	91,48	91,66	85,41	90,46	87,94	91,48	89,96	89,68
6	91,32	91,33	85,74	91,48	92,23	90,91	91,04	90,97	91,48	91,16	90,77
7	91,32	91,33	85,74	92,86	92,23	90,91	91,04	90,97	92,86	91,62	91,09
8	91,32	91,33	88,50	92,86	92,35	90,99	91,16	91,08	92,86	91,70	91,41
9	91,50	91,33	88,50	92,86	92,35	90,99	91,16	91,08	92,86	91,70	91,43
10	91,50	91,58	90,91	92,86	92,35	90,99	91,16	91,08	92,86	91,70	91,70
11	91,50	91,58	90,91	92,86	92,35	91,26	91,16	91,21	92,86	91,74	91,74
12	91,50	91,68	91,04	92,86	92,35	91,48	91,16	91,32	92,86	91,78	91,80
13	91,50	91,68	91,04	92,86	92,35	91,48	91,16	91,32	92,86	91,78	91,80
14	91,50	91,68	91,46	92,86	92,35	91,48	91,16	91,32	92,86	91,78	91,84
15	91,64	91,68	91,46	92,86	92,35	91,48	91,16	91,32	92,86	91,78	91,86
16	91,64	91,68	91,46	92,86	92,35	91,48	91,16	91,32	92,86	91,78	91,86
17	91,64	91,68	91,46	92,86	92,35	91,48	91,16	91,32	92,86	91,78	91,86
18	91,64	91,68	91,46	92,86	92,35	91,48	91,16	91,32	92,86	91,78	91,86
19	91,64	91,68	91,46	92,86	92,35	91,48	91,16	91,32	92,86	91,78	91,86
20	91,64	91,68	91,46	92,86	92,35	91,48	91,16	91,32	92,86	91,78	91,86

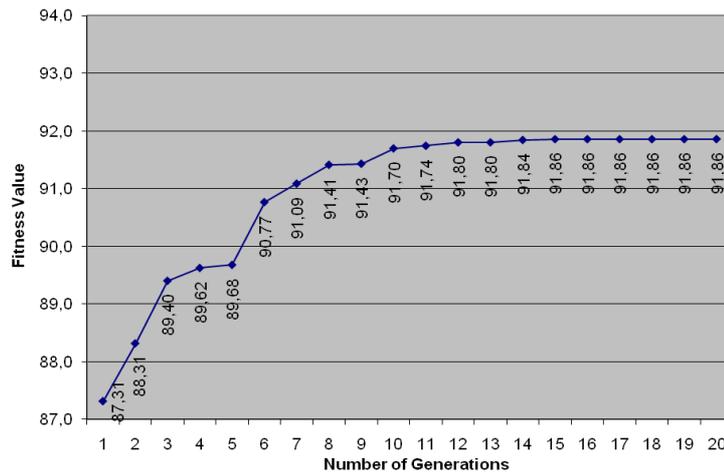


Figure 7.1. The performance curve of the first variant averaged over 10 runs.

In the second variant, the adaptive-fuzzy component was included into the system. In other words, the proposed approach was fully executed. That is,

the crossover and mutation probabilities were adjusted adaptively with respect to performance measures mentioned in section 4.2.2. After 10 individual runs, the highest fitness value of **93.13** was found in the 20th generation, averaged over 10 runs. The fitness values and the performance curve of the second variant are illustrated in Table 7.2 and Figure 7.2, respectively.

Table 7.2. For each run, the fitness values computed for 20 generations that include the adaptive-fuzzy component (the second variant).

Number of Generations	Test -1	Test -2	Test -3	Test -4	Test -5	Test -6	Test -7	Test -8	Test -9	Test -10	Average Fitness
1	88,72	92,32	85,11	81,34	88,72	90,27	83,64	91,24	85,11	86,66	87,31
2	92,26	92,75	85,11	82,51	92,26	91,86	83,64	92,25	85,11	87,00	88,47
3	92,60	92,75	85,11	86,69	92,60	91,86	87,04	92,25	85,11	88,13	89,41
4	92,83	92,75	92,96	86,71	92,83	92,58	87,04	92,60	92,96	90,87	91,41
5	92,83	93,08	92,96	86,71	92,83	92,62	87,04	92,62	92,96	90,87	91,45
6	92,83	94,23	92,96	92,21	92,83	92,62	87,04	92,62	92,96	90,87	92,12
7	92,83	94,23	92,96	92,21	92,83	92,62	87,04	92,62	92,96	90,87	92,12
8	93,08	94,23	92,96	92,29	93,08	92,62	89,80	92,62	92,96	91,79	92,54
9	93,08	94,23	93,06	92,29	93,08	92,80	89,80	92,72	93,06	91,86	92,60
10	93,08	94,23	93,06	92,29	93,08	92,80	92,21	92,84	93,06	92,70	92,93
11	93,08	94,23	93,06	92,56	93,08	92,80	92,21	92,84	93,06	92,70	92,96
12	93,08	94,23	93,06	92,78	93,08	92,80	92,34	92,89	93,06	92,76	93,01
13	93,08	94,23	93,06	92,78	93,08	92,80	92,34	92,89	93,06	92,76	93,01
14	93,08	94,23	93,06	92,78	93,08	92,80	92,76	92,89	93,06	92,90	93,06
15	93,08	94,23	93,06	92,78	93,08	92,94	92,76	92,96	93,06	92,93	93,09
16	93,08	94,23	93,06	92,78	93,08	92,94	92,76	92,96	93,06	92,93	93,09
17	93,08	94,23	93,06	92,78	93,08	92,94	92,76	92,96	93,06	92,93	93,09
18	93,08	94,23	93,06	92,78	93,08	92,94	93,06	92,96	93,06	93,03	93,13
19	93,08	94,23	93,06	92,78	93,08	92,94	93,06	92,96	93,06	93,03	93,13
20	93,08	94,23	93,06	92,78	93,08	92,94	93,06	92,96	93,06	93,03	93,13

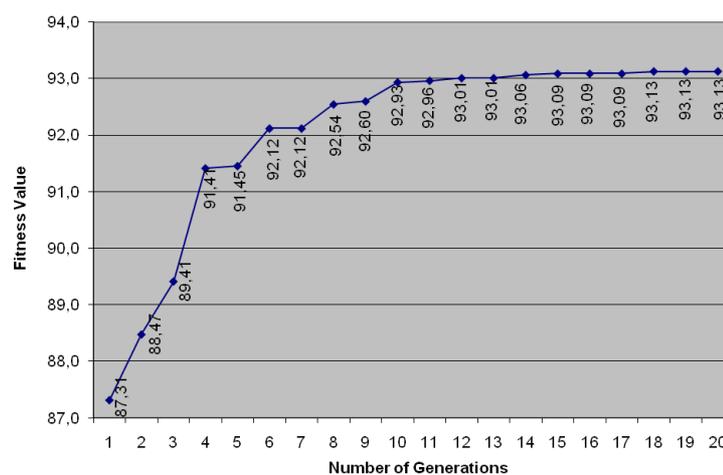


Figure 7.2. The performance curve of the second variant averaged over 10 runs.

In order to compare and analyze the performance curves together, the starting fitness values of the two variants were fixed to 87.31. When the results of the two variants are compared, the conventional genetic algorithm approach (variant - 1) yields 4.55 (91.86 – 87.31) progress in terms of fitness between the first and the last generations. On the other hand, the adaptive-fuzzy genetic algorithm approach (variant - 2) yields a better progress of 5.82 (93.13 – 87.31). Further, the second variant proved to develop by reaching an almost perfect score (92.12%) only after six generations, while the first variant required longer periods of evolution to get close to this level of accuracy. Besides, the maximum fitness value of the second variant was found to be just a little more successful than the first variant at the end of 20 generations. The reason for the slower convergence of the first variant is due to the fixed initial probabilities of the crossover and mutation, which has a potential risk to get trapped in local minimum solution.

The building areas extracted using the second variant of the proposed approach with the approximate fitness values of 83 and 93 are illustrated in Figures 7.3 and 7.4, respectively. As expected, “non-building” regions, which were labeled as “building” (false alarm regions) in Figure 7.3, are eliminated to a large extent in Figure 7.4. Several examples of these areas are illustrated in both figures by red circles numbered 1 to 5.



Figure 7.3. The building regions extracted using the proposed adaptive fuzzy-genetic classification approach with the approximate fitness value of **83**.

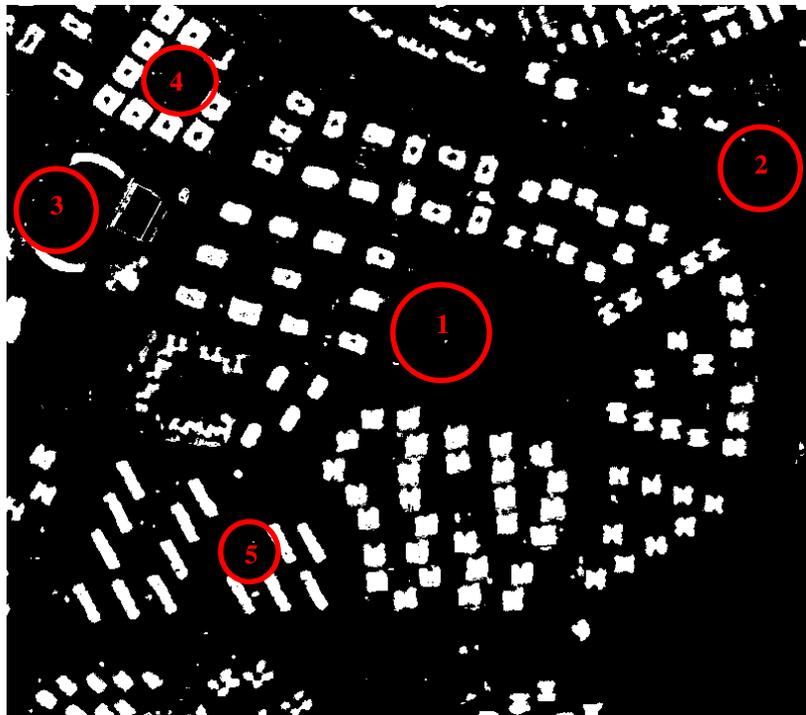


Figure 7.4. The building regions extracted using the proposed adaptive fuzzy-genetic classification approach with the approximate fitness value of **93**.

The experimental results of the building delineation approach were assessed by employing the accuracy assessment technique described in section 4.3.3. In order to avoid the delineation of the misclassified structures or any remaining artifacts, the procedure was executed for the selected building blocks only. Therefore, eight building blocks of the existing building database that contain the quadrilateral buildings were used for this purpose, and each building block was analyzed independently. As mentioned in the previous chapters, an assumption was made that in each block, the buildings are identical with respect to size, except their orientations. For blocks 1, 2, 3, and 4, the extracted patches (Figure 7.5(a)), the extracted footprints (Figure 7.5(b)) and the reference footprints (Figure 7.5(c)) falling within the blocks are illustrated. Similarly, for blocks 5, 6, 7, and 8, the corresponding outputs are shown in Figure 7.6.

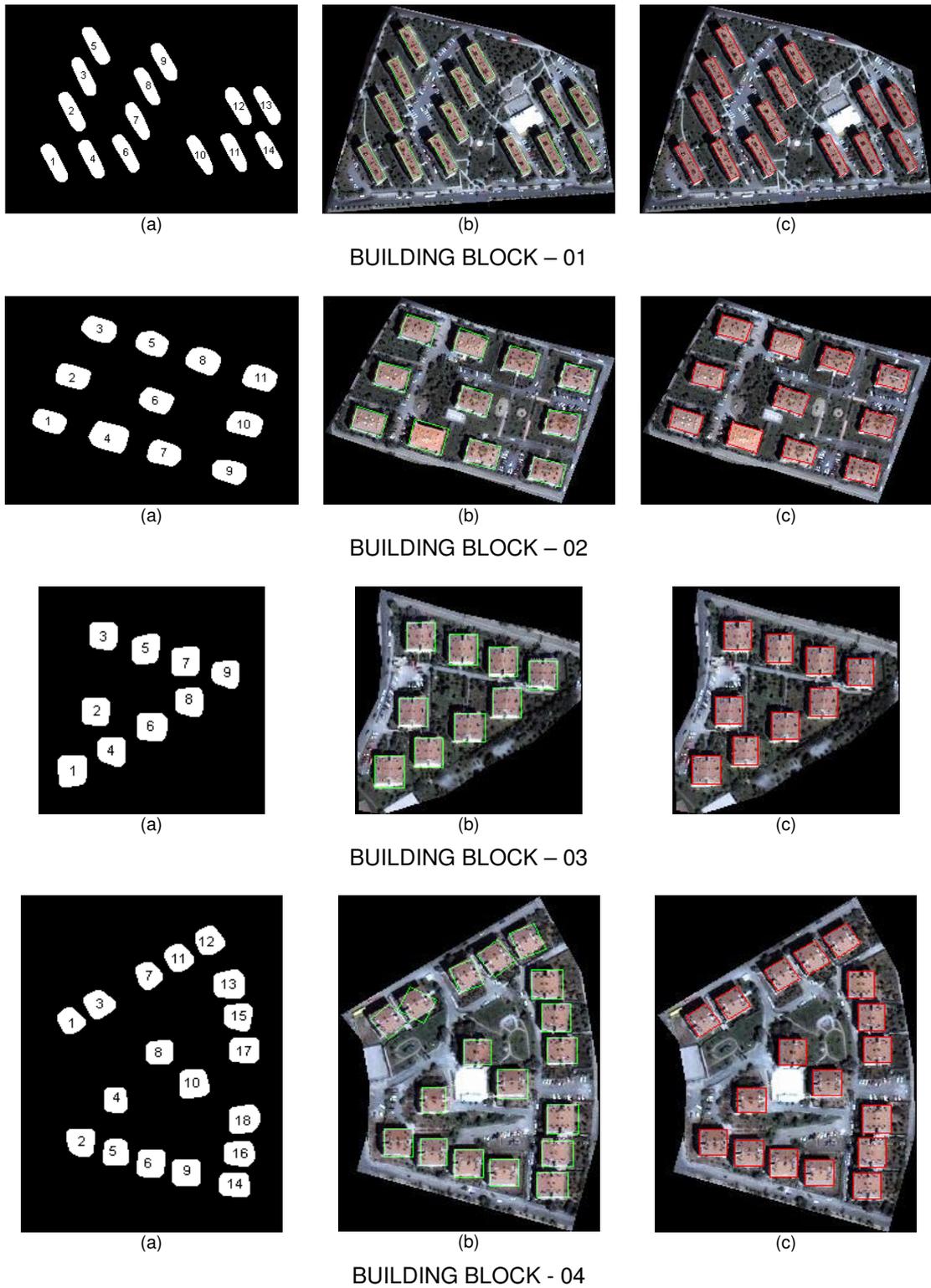


Figure 7.5. (a) The detected building patches, on which the numbers represent building IDs after the morphological pre-processing, (b) the extracted footprints (green), and (c) the reference footprints (red).

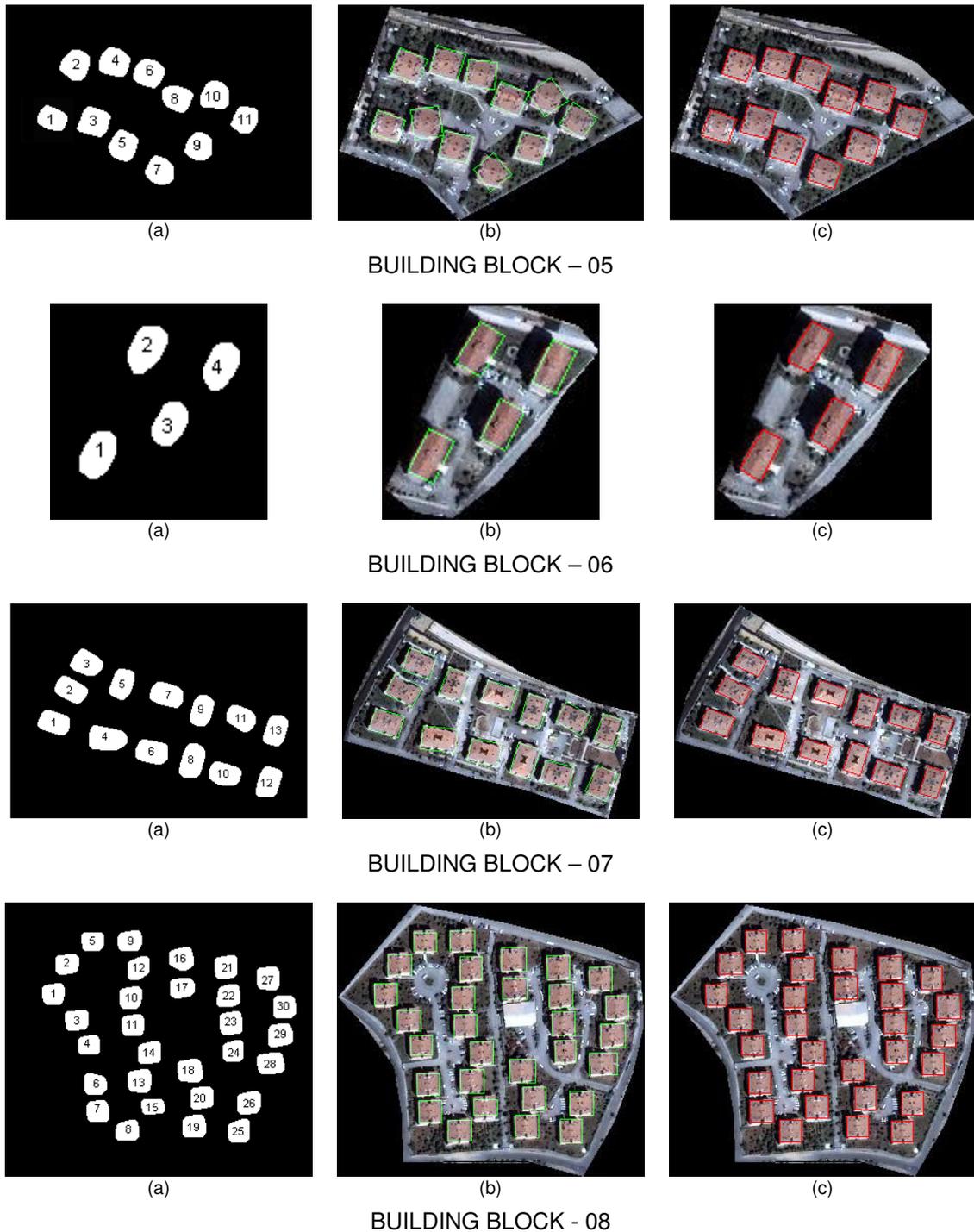


Figure 7.6. (a) The detected building patches, on which the numbers represent building IDs after the morphological pre-processing, (b) the extracted footprints (green), and (c) the reference footprints (red).

For each building falling within each building block, the distance errors computed at the corner points are given in Table 7.3. The accuracies computed under different confidence levels are given in Table 7.4.

Table 7.3. The distance errors (pixel) computed for the corner points of the buildings.

BUILDING BLOCK	Building No	DistErr (Pixel)					
		Corner Point-1	Corner Point-2	Corner Point-3	Corner Point-4	Mean_DistErr	SD_DistErr
1	1	1.41	1.41	1.41	0.00	1.06	0.71
	2	1.00	1.00	1.00	1.00	1.00	0.00
	3	1.00	2.00	2.00	1.00	1.50	0.58
	4	1.41	2.24	2.00	1.00	1.66	0.56
	5	0.00	2.24	1.41	1.41	1.27	0.93
	6	2.83	1.00	1.00	3.00	1.96	1.11
	7	3.61	2.24	2.24	3.16	2.81	0.69
	8	2.24	1.00	0.00	1.41	1.16	0.93
	9	2.24	1.41	1.41	1.00	1.52	0.52
	10	2.24	1.41	1.41	1.00	1.52	0.52
	11	2.00	1.00	0.00	2.00	1.25	0.96
	12	2.24	2.24	1.41	1.41	1.83	0.47
	13	1.41	3.00	2.24	1.00	1.91	0.89
	14	2.83	1.41	2.24	3.16	2.41	0.77
2	1	2.24	2.00	4.47	2.00	2.68	1.20
	2	1.00	2.00	3.16	3.00	2.29	1.00
	3	1.41	2.00	1.00	1.00	1.35	0.47
	4	3.00	2.24	2.24	3.61	2.77	0.66
	5	0.00	3.16	1.14	1.00	1.39	1.32
	6	2.24	1.00	1.41	2.24	1.72	0.62
	7	1.41	4.12	4.47	2.24	3.06	1.47
	8	1.00	5.00	4.00	1.00	2.75	2.06
	9	1.41	1.41	2.24	2.00	1.77	0.42
	10	1.00	1.41	2.24	3.16	1.95	0.96
	11	1.00	2.00	3.16	2.24	2.10	0.89
3	1	3.00	0.00	1.00	3.16	1.79	1.55
	2	2.24	1.41	2.24	2.83	2.18	0.58
	3	1.00	1.00	1.41	1.41	1.21	0.24
	4	2.24	1.41	2.24	2.83	2.18	0.58
	5	2.00	0.00	2.00	2.83	1.71	1.20
	6	2.00	1.00	3.16	3.61	2.44	1.18
	7	2.00	0.00	0.00	2.00	1.00	1.15
	8	1.00	0.00	1.00	1.41	0.85	0.60
	9	3.16	2.24	2.24	3.16	2.70	0.53
4	1	4.47	3.00	0.00	3.00	2.62	1.88
	2	2.24	1.00	2.00	2.83	2.02	0.76
	3	9.85	9.06	8.00	5.39	8.07	1.94
	4	1.41	1.00	3.00	3.16	2.14	1.10
	5	2.00	2.24	1.00	0.00	1.31	1.02
	6	1.41	1.41	2.24	2.24	1.83	0.47
	7	2.24	1.41	3.00	5.00	2.91	1.53
	8	1.00	1.41	1.41	1.00	1.21	0.24
	9	1.41	1.00	1.00	1.41	1.21	0.24
	10	3.16	1.00	2.00	3.61	2.44	1.18
	11	2.83	2.00	1.00	2.24	2.02	0.76
	12	2.00	3.61	3.00	5.39	3.50	1.42
	13	1.00	1.00	4.12	4.12	2.56	1.80
	14	1.00	0.00	3.00	3.16	1.79	1.55
	15	1.00	0.00	2.00	2.24	1.31	1.02
	16	0.00	3.00	4.24	3.00	2.56	1.80
	17	1.00	1.00	2.24	2.24	1.62	0.71
	18	1.41	1.00	3.00	3.16	2.14	1.10

5	1	1.00	3.16	1.00	1.00	1.54	1.08
	2	2.83	1.41	1.41	3.61	2.32	1.09
	3	5.39	7.21	6.40	7.07	6.52	0.83
	4	2.24	1.00	2.24	2.24	1.93	0.62
	5	2.24	0.00	1.00	2.00	1.31	1.02
	6	5.10	4.47	2.83	3.16	3.89	1.07
	7	9.22	8.25	7.28	7.07	7.95	0.99
	8	3.00	1.00	2.00	2.83	2.21	0.92
	9	1.00	1.41	2.24	1.00	1.41	0.58
	10	9.49	8.06	10.20	10.44	9.55	1.07
	11	3.00	3.16	5.39	5.39	4.23	1.33
6	1	3.16	1.00	1.00	2.24	1.85	1.05
	2	3.61	1.00	0.00	2.24	1.71	1.56
	3	4.00	1.00	1.00	1.41	1.85	1.44
	4	3.61	1.00	1.41	3.00	2.25	1.25
7	1	2.24	2.00	1.00	1.41	1.66	0.56
	2	2.83	2.00	3.00	2.83	2.66	0.45
	3	3.61	5.00	2.00	4.47	3.77	1.31
	4	5.39	4.00	1.00	4.47	3.71	1.90
	5	3.00	2.00	1.41	2.24	2.16	0.66
	6	4.24	5.10	1.41	3.61	3.59	1.57
	7	4.00	3.16	0.00	2.24	2.35	1.72
	8	4.47	3.61	2.00	4.47	3.64	1.17
	9	1.41	2.24	1.00	1.41	1.52	0.52
	10	2.24	2.00	0.00	3.00	1.81	1.28
	11	6.40	2.24	5.83	2.83	4.32	2.10
	12	3.16	2.83	2.24	4.47	3.17	0.95
	13	3.16	2.24	1.00	2.24	2.16	0.89
8	1	1.00	1.41	2.24	2.00	1.66	0.56
	2	1.41	1.41	1.41	1.41	1.41	0.00
	3	1.00	2.00	2.24	1.41	1.66	0.56
	4	1.00	1.41	1.41	1.00	1.21	0.24
	5	0.00	2.00	2.83	2.00	1.71	1.20
	6	2.00	1.00	1.41	2.24	1.66	0.56
	7	1.00	1.00	0.00	0.00	0.50	0.58
	8	1.00	0.00	2.00	2.24	1.31	1.02
	9	1.41	1.41	3.16	3.16	2.29	1.01
	10	2.00	0.00	1.00	2.24	1.31	1.02
	11	1.00	1.00	2.24	2.24	1.62	0.71
	12	1.00	1.41	1.00	0.00	0.85	0.60
	13	3.00	0.00	2.00	3.61	2.15	1.58
	14	1.41	1.00	1.00	1.41	1.21	0.24
	15	3.00	4.00	4.12	3.16	3.57	0.57
	16	2.24	1.41	3.16	3.61	2.60	0.98
	17	0.00	2.00	2.83	2.00	1.71	1.20
	18	2.24	1.41	3.16	3.61	2.60	0.98
	19	1.00	1.00	2.24	2.24	1.62	0.71
	20	0.00	0.00	2.00	2.00	1.00	1.15
	21	0.00	1.00	1.41	1.00	0.85	0.60
	22	0.00	1.00	1.41	1.00	0.85	0.60
	23	2.00	1.00	1.41	2.24	1.66	0.56
	24	1.41	1.00	0.00	1.00	0.85	0.60
	25	1.41	2.24	2.24	1.41	1.83	0.47
	26	0.00	1.00	2.24	2.00	1.31	1.02
	27	1.00	0.00	2.00	2.24	1.31	1.02
	28	1.00	1.00	5.00	5.00	3.00	2.31
	29	1.00	1.00	2.24	2.24	1.62	0.71
	30	1.00	1.41	1.00	0.00	0.85	0.60

Table 7.4. The accuracies of the buildings under different confidence levels.

BUILDING BLOCK	Building No	Accuracy Under Different Confidence Levels			
		80%	85%	90%	95%
1	1	1.65	1.80	1.97	2.22
	2	1.00	1.00	1.00	1.00
	3	1.98	2.10	2.24	2.45
	4	2.13	2.25	2.38	2.59
	5	2.05	2.24	2.45	2.79
	6	2.89	3.11	3.37	3.78
	7	3.39	3.53	3.69	3.94
	8	1.94	2.13	2.35	2.69
	9	1.95	2.06	2.18	2.37
	10	1.95	2.06	2.18	2.37
	11	2.05	2.25	2.48	2.82
	12	2.22	2.32	2.43	2.61
	13	2.66	2.84	3.05	3.37
	14	3.05	3.21	3.39	3.67
2	1	3.69	3.93	4.22	4.65
	2	3.13	3.34	3.57	3.94
	3	1.75	1.85	1.96	2.13
	4	3.33	3.46	3.62	3.86
	5	2.50	2.77	3.08	3.57
	6	2.24	2.37	2.51	2.74
	7	4.30	4.60	4.95	5.48
	8	4.48	4.90	5.39	6.14
	9	2.12	2.20	2.30	2.45
	10	2.76	2.95	3.18	3.53
	11	2.85	3.03	3.24	3.56
3	1	3.09	3.41	3.77	4.33
	2	2.67	2.79	2.92	3.13
	3	1.41	1.46	1.51	1.60
	4	2.67	2.79	2.92	3.13
	5	2.72	2.96	3.25	3.69
	6	3.43	3.67	3.95	4.38
	7	1.97	2.21	2.48	2.90
	8	1.36	1.48	1.62	1.84
	9	3.15	3.26	3.38	3.58
4	1	4.20	4.58	5.02	5.71
	2	2.66	2.81	2.99	3.27
	3	9.71	10.10	10.56	11.27
	4	3.07	3.29	3.55	3.95
	5	2.17	2.38	2.62	2.99
	6	2.22	2.32	2.43	2.61
	7	4.20	4.52	4.88	5.44
	8	1.41	1.46	1.51	1.60
	9	1.41	1.46	1.51	1.60
	10	3.43	3.67	3.95	4.38
	11	2.66	2.81	2.99	3.27
	12	4.69	4.98	5.32	5.84
	13	4.08	4.45	4.87	5.53
	14	3.09	3.41	3.77	4.33
	15	2.17	2.38	2.62	2.99
	16	4.08	4.45	4.87	5.53
	17	2.22	2.36	2.53	2.79
	18	3.07	3.29	3.55	3.95

5	1	2.45	2.67	2.92	3.32
	2	3.23	3.45	3.71	4.11
	3	7.22	7.39	7.58	7.89
	4	2.45	2.57	2.72	2.94
	5	2.17	2.38	2.62	2.99
	6	4.79	5.01	5.26	5.66
	7	8.78	8.99	9.22	9.58
	8	2.98	3.16	3.38	3.71
	9	1.90	2.02	2.16	2.37
	10	10.45	10.66	10.92	11.31
	11	5.35	5.62	5.94	6.42
6	1	2.73	2.95	3.20	3.58
	2	3.02	3.34	3.71	4.28
	3	3.07	3.36	3.70	4.23
	4	3.30	3.56	3.85	4.31
7	1	2.13	2.25	2.38	2.59
	2	3.04	3.13	3.24	3.40
	3	4.87	5.14	5.45	5.93
	4	5.31	5.70	6.14	6.84
	5	2.71	2.85	3.00	3.24
	6	4.91	5.24	5.61	6.18
	7	3.80	4.15	4.56	5.19
	8	4.62	4.86	5.13	5.55
	9	1.95	2.06	2.18	2.37
	10	2.88	3.15	3.45	3.91
	11	6.09	6.52	7.01	7.77
	12	3.97	4.16	4.39	4.73
	13	2.90	3.09	3.29	3.62
8	1	2.13	2.25	2.38	2.59
	2	1.41	1.41	1.41	1.41
	3	2.13	2.25	2.38	2.59
	4	1.41	1.46	1.51	1.60
	5	2.72	2.96	3.25	3.69
	6	2.13	2.25	2.38	2.59
	7	0.98	1.10	1.24	1.45
	8	2.17	2.38	2.62	2.99
	9	3.14	3.34	3.58	3.95
	10	2.17	2.38	2.62	2.99
	11	2.22	2.36	2.53	2.79
	12	1.36	1.48	1.62	1.84
	13	3.48	3.80	4.17	4.75
	14	1.41	1.46	1.51	1.60
	15	4.05	4.17	4.30	4.51
	16	3.43	3.63	3.86	4.21
	17	2.72	2.96	3.25	3.69
	18	3.43	3.63	3.86	4.21
	19	2.22	2.36	2.53	2.79
	20	1.97	2.21	2.48	2.90
	21	1.36	1.48	1.62	1.84
	22	1.36	1.48	1.62	1.84
	23	2.13	2.25	2.38	2.59
	24	1.36	1.48	1.62	1.84
	25	2.22	2.32	2.43	2.61
	26	2.17	2.38	2.62	2.99
	27	2.17	2.38	2.62	2.99
	28	4.94	5.41	5.96	6.80
	29	2.22	2.36	2.53	2.79
	30	1.36	1.48	1.62	1.84

A total of **110** buildings that fall in eight blocks were analyzed. The minimum and maximum values for the mean distance errors were computed to be **0.5**

and **9.55**, respectively. Similarly, the extreme values of the standard deviation were computed to be **0** and **2.31**, respectively. The variation of the accuracy values were found to be in the range **0.98 - 10.45** at the confidence level 80%, in the range **1.00 - 10.66** at the confidence level 85%, in the range **1.00 - 10.92** at the confidence level 90%, and in the range **1.00 - 11.31** at the confidence level 95%. The average values of the mean distance errors were computed as **1.63** for block 1, **2.16** for block 2, **1.78** for block 3, **2.40** for block 4, **3.89** for block 5, **1.91** for block 6, **2.80** for block 7, and **1.59** for block 8. In addition, the accuracies under the confidence level of 95% were also investigated. This confidence level is a typically used confidence level in the vast majority of the scientific studies. The average accuracies were computed as **2.76** for block 1, **3.82** for block 2, **3.18** for block 3, **4.28** for block 4, **5.48** for block 5, **4.10** for block 6, **4.72** for block 7, and **2.91** for block 8 at the specified confidence level. Since the spatial resolution of the satellite image is 1-meter, the unit of errors may be given as “meters” instead of “pixels”. Of the building blocks, block#1 provided the highest average accuracy of 2.76 meters at the confidence level of 95%. This means that we are confident that 95% of the time the observed errors will be 2.76 meters or less. Conversely, we are also accepting that 5% of the time errors exceeding 2.76 meters will occur. The second highest accuracy of 2.91 pixels was provided by block#8. The accuracies computed for the remaining blocks were ranked as follows: block#3 with 3.18 meters, block#2 with 3.82 meters, block#6 with 4.10 meters, block#4 with 4.28 meters, block#7 with 4.72 meters, and block#5 with 5.48 meters.

Based on the experimental results it can be stated that the developed building delineation approach was quite successful for blocks #1, #8, #3 and #2, for which the errors were computed to be less than 4 meters at 95% confidence level. Further, when the buildings in these blocks are visually examined, the boundary agreements are found to be quite successful. On the contrary, the accuracies were found to be relatively low for the building blocks #6, #4, #7 and #5. The visual examination also supports the

experimental findings for the buildings falling within these blocks, in which a few of the boundaries are observed to be mis-orientated.

Besides the experimental results, the running times, which are required by the computer to perform the developed algorithms, were also computed. The “genetic algorithm parameter analysis” step took the running time of **120 hours (5 days)**. This step was performed once in the beginning of the methodology and composed of several experiments in order to estimate the optimum genetic algorithm parameters, such as chromosome and gene sizes together with the number of the training and test regions. After that, the adaptive fuzzy-genetic approach was executed with the optimum parameters to extract the 2-d building patches. This step took an average time of **24 minutes** for the whole study area. In the final step, the delineation of the building regions was carried out on block by block basis. For each block, the delineation time took only **2 seconds** on average. As a result, the total elapsed time for the delineation of eight building blocks was computed to be **16 seconds** on average. If the optimal parameter analysis stage is excluded, the building extraction and delineation step would take less than **25 minutes** for the eight building blocks containing a total of 110 single buildings. All these performance measurements were operated on Windows XP operating system and executed on a Pentium Core 2 1.86 GHz processor with 2 GB RAM.

In most cases the proposed 2-d building extraction and delineation approach was successful for detecting the 2-d building boundaries. However, several shortcomings were evident in certain cases. After investigating the reasons of the failures, it was found that the success of building extraction highly affects the accuracy of building delineation. One reason was the over segmentation of the image, which may occur due to the spectral confusion among the feature classes. In the present study, this problem is mostly encountered for closely located buildings. The above mentioned segmentation problem is illustrated in Figure 7.7, in which the patches of two closely located buildings were detected as a merged single patch.

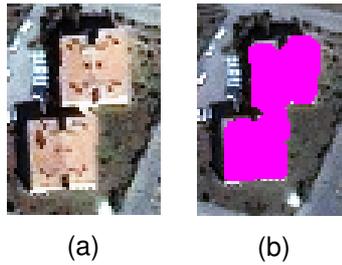


Figure 7.7. (a) The image that contains closely located two buildings and (b) the result of segmentation which generated a merged single patch (colored in pink).

A different type of problem was faced during the boundary delineation of the buildings. For the square-shaped buildings with diagonal orientation, the building boundaries were delineated wrongly in several cases. This shortcoming is due to the mis-orientation of the best fitting ellipse, which arises from the irregular shape of the binary patch that have more than four vertices. For the square-shaped buildings that are elongating vertically, this problem is solved to a certain extent by using a minimum bounding box (mbb). The orientation of these buildings can be easily determined by counting the pixels falling inside the mbb. If these pixels are more than a predefined threshold value, then the buildings are accepted to be vertically elongated. This type of error was generally observed in building blocks #4 and #5. In Figure 7.8, two distinct square-shaped buildings with different orientations are illustrated. The vertically elongated building (Figure 7.8(a)) was delineated correctly after applying the method based on mbb, while the diagonally elongated building (Figure 7.8(b)) was wrongly delineated because of the above mentioned reason.

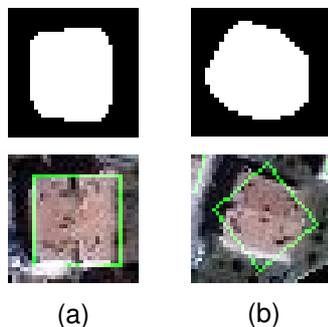


Figure 7.8. The building boundaries (in green) detected from (a) a vertically elongated building and (b) a diagonally elongated building. The binary images represent the extracted building patches.

The developed 2-d building extraction and delineation approach has a limitation, which is that the approach solely operates on the quadrilateral-shaped buildings. Therefore, in this study, the polygonal shaped buildings having more than four edges are kept out of the consideration. As described above, the proposed delineation approach is based on finding the rectangle with the largest area, which can be inscribed in the best-fitting ellipse. Therefore, a building patch must have four edges. To exclude the complex structures, the buildings are analyzed on block-by-block basis. Figure 7.9 illustrates two different examples for the complex-shaped buildings, which are not included in the analyses.



Figure 7.9. The buildings with complex geometry.

7.2. Results and Discussion for Building Facade Texture Acquisition

The developed approach was tested on both the Batikent and eTrims datasets. After performing the accuracy assessment explained in section 5.2, the quantitative evaluation results were obtained. For both datasets, the evaluation results are provided in Table 7.5.

Table 7.5. The quantitative results of the facade image extraction for the eTrims and Batikent datasets.

Dataset	Building No	TP	TN	FP	FN	FDP(%)	BF	QP(%)
eTRIMS	1	30494	24271	137	10634	74.1	0.004	73.9
	2	31372	24409	350	9405	76.9	0.011	76.3
	3	34607	13634	62	17233	66.8	0.002	66.7
	4	39143	23589	145	2659	93.6	0.004	93.3
	5	44334	19658	127	1417	96.9	0.003	96.6
	<i>Average:</i>						81.7	0.005
BATIKENT	1	70929	147223	9778	2470	96.6	0.138	85.3
	2	71047	144305	14339	709	99.0	0.202	82.5
	3	72034	156757	1358	251	99.7	0.019	97.8
	4	73297	152183	4917	3	100	0.067	93.7
	5	83568	125890	20457	485	99.4	0.245	80.0
	6	110366	89258	3056	27720	79.9	0.028	78.2
	7	69718	149359	11308	15	100	0.162	86.0
	8	81713	77551	528	70608	53.6	0.006	53.5
	9	71495	148445	3797	6663	91.5	0.053	87.2
	10	47413	175553	736	6698	87.6	0.016	86.4
	11	85193	116449	3357	25401	77.0	0.039	74.8
	12	117029	57081	502	55788	67.7	0.004	67.5
	13	68217	151735	3823	6625	91.1	0.056	86.7
	14	90673	118861	20678	188	99.8	0.228	81.3
	15	88658	135845	5680	217	99.8	0.064	93.8
	<i>Average:</i>						89.5	0.088

For the eTrims dataset, the computed facade detection percentages (FDP) were in the range **74.1% - 96.9%**. Similarly, the branching factors (BF) and the quality percentages (QP) were found to be between **0.002 - 0.011** and **66.7% - 96.6%**, respectively. The average “FDP”, “BF”, and “QP” values were found to be **81.7%**, **0.005**, and **81.4%**, respectively. The “QP” values were close to “FDP” values since the branching factors were relatively small. This means that the over-segmentation problem is almost never encountered in the extraction of the building facades. The building facade patches of eTrims data set with the highest and lowest quality percentages are illustrated in Figures 7.10 and 7.11, respectively.

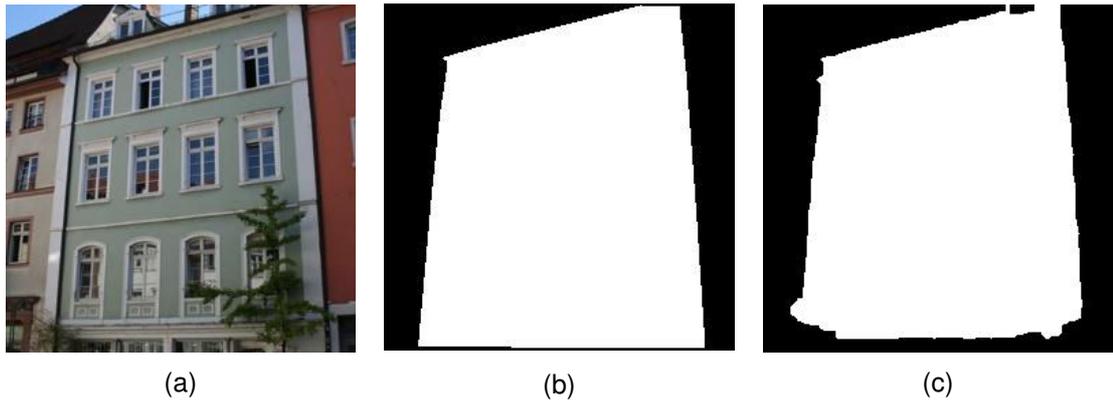


Figure 7.10. (a) A facade image from the eTrims dataset, (b) the reference patch generated manually, and (c) the facade patch extracted automatically through the developed approach.

This facade image provided the highest QP value of 96.6%.

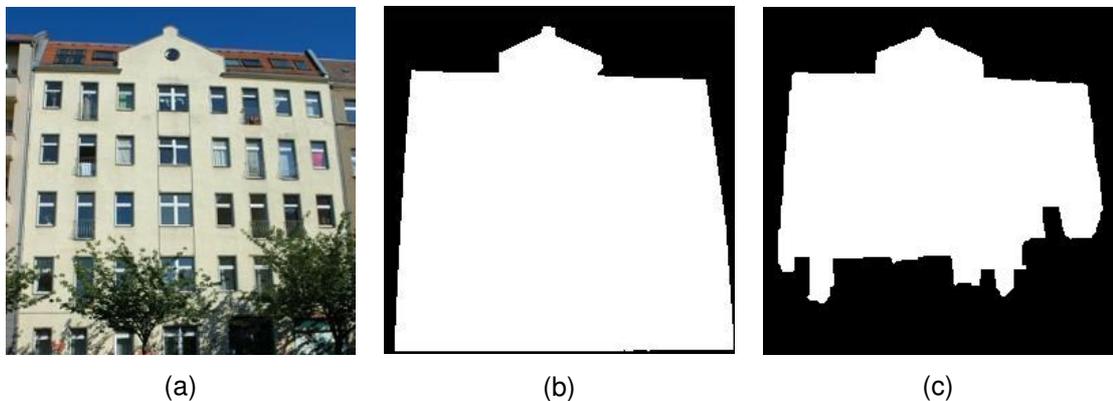


Figure 7.11. (a) A facade image from the eTrims dataset, (b) the reference patch generated manually, and (c) the facade patch extracted automatically through the developed approach.

This facade image provided the lowest QP value of 66.7%.

For the Batikent dataset (Table 7.5), the “FDP” values were found to be in the range **53.6% - 100%**. In the same way, the branching factors (BF) and the quality percentages (QP) were computed to be in the range **0.004 - 0.245** and **53.5% - 97.8%**, respectively. The average facade detection percentage was computed to be **89.5%**, while the average “BF” and “QP” values were found to be **0.088** and **82.3%**, respectively. Different from the eTrims dataset, there was a significant gap between the average values of “FDP” and “QP”. It is believed that this was due to the high branching factor values compared to values of the eTrims dataset. This means that the over-segmentation problem is evident in the Batikent data set, in which the analysis labels background pixels as foreground pixels. It is also observed that the computed

facade detection percentages of **53.6%** (building #8) and **67.7%** (building #12) were found to be considerably low. It is believed that these low accuracy values were due to the high number of false negative (FN) pixels. If these buildings are excluded from the dataset, then the average facade detection accuracy can reach to **93.9%**. From the Batikent dataset, the facade patches of two buildings with the highest and lowest quality percentage values are shown in Figures 7.12 and 7.13, respectively.

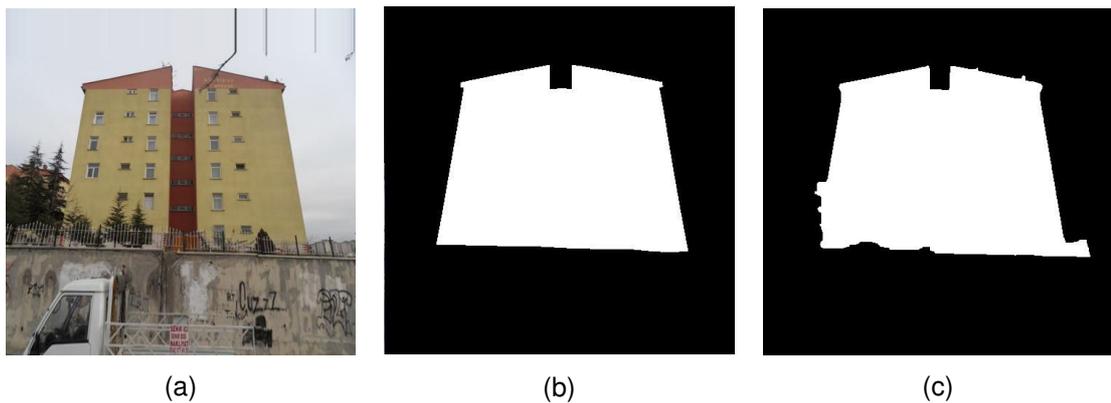


Figure 7.12. (a) A facade image from the Batikent dataset, (b) the reference patch generated manually, and (c) the facade patch extracted automatically through the developed approach.

This facade image provided the highest QP value of 97.8%.

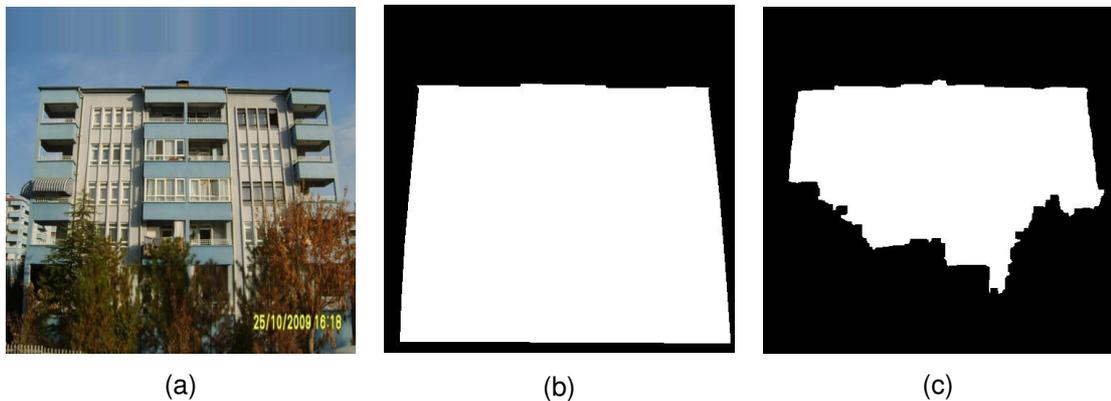


Figure 7.13. (a) A facade image from the Batikent dataset, (b) the reference patch generated manually, and (c) the facade patch extracted automatically through the developed approach.

This facade image provided the lowest QP value of 53.5%.

The developed approach for facade image rectification (section 5.3) was tested on the extracted facade images of both datasets. The accuracy assessment of the rectified images was carried out using the technique

described in section 5.3.4. For both data sets, the assessment results that contain a complete list of pixel distance errors are given in Table 7.6. The accuracies computed at 80%, 85%, 90% and 95% confidence levels are given in Table 7.7.

Table 7.6. The pixel errors computed at the test points on the rectified facade images.

Dataset	Building No	DistErr (pixel)					
		Test Point #1	Test Point #2	Test Point #3	Test Point #4	Mean_DistErr	SD_DistErr
eTRIMS	1	3.61	5.10	1.41	3.00	3.28	1.52
	2	1.41	2.24	2.24	1.41	1.83	0.47
	3	2.00	1.00	2.24	1.41	1.66	0.56
	4	4.47	9.49	16.55	8.54	9.76	5.02
	5	2.83	1.41	3.00	7.28	3.63	2.53
	<i>Average:</i>					4.03	2.02
BATIKENT	1	4.24	4.24	2.83	4.00	3.83	0.68
	2	1.00	3.61	3.16	1.00	2.19	1.39
	3	14.14	21.40	17.00	16.76	17.33	3.01
	4	2.24	2.00	6.08	6.40	4.18	2.39
	5	5.83	8.49	6.40	4.47	6.30	1.67
	6	12.04	2.83	11.40	9.22	8.87	4.21
	7	20.52	19.72	4.24	13.93	14.60	7.51
	8	6.00	8.06	8.06	5.83	6.99	1.24
	9	7.07	16.55	10.30	7.07	10.25	4.47
	10	5.83	3.61	3.16	5.83	4.61	1.42
	11	5.83	12.37	1.41	5.00	6.15	4.57
	12	15.30	9.22	11.40	11.40	11.83	2.53
	13	9.22	6.71	7.81	6.40	7.54	1.28
	14	2.83	6.40	3.61	1.41	3.56	2.10
	15	3.61	8.06	4.47	4.00	5.03	2.05
<i>Average:</i>					7.55	2.70	

Table 7.7. For the rectified facade images, the accuracies computed at 80%, 85%, 90%, and 95% confidence levels.

Dataset	Building No	Accuracy Under Different Confidence Levels			
		80%	85%	90%	95%
eTRIMS	1	4.56	4.87	5.23	5.79
	2	2.22	2.32	2.43	2.61
	3	2.13	2.25	2.38	2.59
	4	13.98	15.01	16.19	18.03
	5	5.76	6.28	6.28	7.80
	<i>Average:</i>	5.73	6.15	6.50	7.36
BATIKENT	1	4.40	4.54	4.69	4.94
	2	3.36	3.64	3.97	4.48
	3	19.85	20.47	21.18	22.28
	4	6.19	6.67	7.24	8.11
	5	7.70	8.04	8.43	9.04
	6	12.41	13.27	14.26	15.79
	7	20.91	22.45	24.21	26.95
	8	8.03	8.29	8.58	9.03
	9	14.00	14.92	15.97	17.60
	10	5.80	6.10	6.43	6.95
	11	9.99	10.92	12.00	13.66
	12	13.96	14.47	15.07	15.99
	13	8.61	8.87	9.17	9.63
	14	5.33	5.76	6.25	7.02
	15	6.76	7.18	7.66	8.41
<i>Average:</i>	9.82	10.37	11.01	11.99	

For the eTrims dataset, the minimum and maximum values for the mean distance errors were computed to be **1.66** and **9.76**, respectively. Similarly, the minimum and maximum values of the standard deviation were computed to be **0.47** and **5.02**, respectively (Table 7.6). The variation of the accuracy values were computed to be in the range **2.13 - 13.98** at the confidence level 80%, in the range **2.25 - 15.01** at the confidence level 85%, in the range **2.38 - 16.19** at the confidence level 90%, and in the range **2.59 - 18.03** at the confidence level 95% (Table 7.7). The average value for the mean distance errors were computed as **4.03** in the eTrims dataset. In addition, the accuracy under the confidence level of 95% was also investigated. At this confidence level, the average value was found to be **7.36**. Buildings #3 and

#2 provided the best accuracies, which were 2.59 and 2.61, respectively. These buildings were followed by building #1 with the accuracy of 5.79 and building #5 with the accuracy of 7.80. Of the used five test buildings, building #4 provided the worst results. For this building the mean error value and the standard deviation were computed as 9.76 and 5.02 pixels (Table 7.6). The reason why this building revealed the lowest accuracy was investigated. It was found that on the right facade border, the output of the Hough transformation produced too many short line segments with different orientations. Therefore, the optimal coinciding trend line generated from these Hough line segments was not correct. The rectification process was highly affected by the incorrectly determined trend line, which boosts the positional error.

When the results of the Batikent dataset were analyzed, the minimum and maximum values for the mean distance errors were computed to be **2.19** and **17.33**, respectively. Likewise, the extreme values of the standard deviation were found to be **0.68** and **7.51**, respectively (Table 7.6). The variation of the accuracy values were computed to be in the range **3.36 - 20.91** at the confidence level 80%, in the range **3.64 - 22.45** at the confidence level 85%, in the range **3.97 - 24.21** at the confidence level 90%, and in the range **4.48 - 26.95** at the confidence level 95% (Table 7.7). The average value of the mean distance errors were computed as **7.55** in this dataset. In addition, the accuracy under the confidence level of 95% was also investigated. At this confidence level, the average value of 15 buildings was found to be **11.99**. It was found that buildings #2, #1, #10, #14, and #4 provided the lowest errors with the accuracies ranging from 4.48 to 8.11 pixels. The next five buildings that provided the lowest errors were building #15, #8, #5, #13, and #11, for which the accuracies were computed in the range 8.41 – 13.66. The worst five buildings (Buildings #6, #12, #9, #3, and #7) provided the highest errors with accuracies ranging from 15.79 to 26.95 at the confidence level 95% (Table 7.7). The reasons for the failures were investigated and it was found that the main reasons for the high error values were the deficiencies and the redundancies of the extracted facade border lines. These cases likely occur

due to the protrusions, such as balconies, pavilions and overhanging roofs. Since the facade extraction algorithm segments the protruded regions as foreground texture the original facade edges get lost, therefore.

The developed approach for occlusion removal was tested on the rectified images of both datasets. Using the qualitative accuracy assessment method described in section 5.4.4, the occlusion-free facade images of the buildings that fall within the datasets were evaluated by ten viewers. The viewer profiles are quite dissimilar, such as from senior students in the field of computer science, research assistants, professors and the people working in the field of image processing. For both datasets, the ratings of the viewers are provided in Table 7.8.

Table 7.8. The ratings of ten viewers for the occlusion-free facade images.

Dataset	Building No	VIEWER #										Average Rating
		1	2	3	4	5	6	7	8	9	10	
eTRIMS	1	2	1	2	2	1	1	2	2	2	1	1.6
	2	2	3	2	2	3	1	2	1	2	2	2.0
	3	2	2	3	3	2	1	3	1	3	1	2.1
	4	3	3	4	3	3	2	3	2	3	2	2.8
	5	3	2	4	4	2	3	3	2	3	3	2.9
	<i>Average:</i>											2.28
BATIKENT	1	3	3	5	4	4	2	4	3	4	2	3.4
	2	3	2	4	3	3	2	4	3	3	2	2.9
	3	3	3	2	3	2	1	3	2	3	1	2.3
	4	3	4	3	3	3	2	4	3	3	2	3.0
	5	2	3	2	2	3	1	3	2	3	2	2.3
	6	2	3	2	3	2	2	3	2	2	1	2.2
	7	3	3	2	2	3	1	2	3	2	1	2.2
	8	3	4	5	3	4	2	3	4	3	3	3.4
	9	1	2	3	2	2	1	2	2	3	2	2.0
	10	3	2	4	2	3	2	2	4	2	2	2.6
	11	2	2	3	2	3	2	2	3	3	2	2.4
	12	2	2	3	2	2	1	2	3	2	1	2.0
	13	4	4	5	3	5	3	3	5	3	4	3.9
	14	1	2	3	1	2	1	2	2	2	1	1.7
	15	2	4	3	2	2	1	4	2	2	2	2.4
<i>Average:</i>											2.58	

For the eTrims dataset, the average ratings of the facade images were in the range **1.6 – 2.9**. Buildings #1, #2, #3, #4 and #5 comprised the standing from the best to worst, respectively. The average value was computed to be **2.28**, which can be identified as “Fine” with respect to rating scale given in section 5.4.4. This means that the reviewers were satisfied with the results and found the occlusion removal quality nearly high. On the other hand, for the Batikent dataset, the average ratings were in the range **1.7 – 3.9**. The best rating was computed for building #14. The subsequent best ratings were computed for buildings #9, #12, #6, #7, #3, #5, #11, #15, #10, #2, #4, #1, #8, and # 13. The average of the whole dataset was calculated as **2.58**, which is staying in the middle of the criterions “Fine” and “Passable”. This can be interpreted as the occlusion-free images have an acceptable quality. For each dataset, the facade images with the best and the worst ratings are shown in Figures 7.14 and 7.15. The main reasons for the failures were investigated and it was found that the major problems were the excessive blurring of the seams, discontinuities between the objects (e.g. shifted windows or broken lines), and the occlusions that cannot be removed completely. For instance, building #5 of eTrims database suffers from a broken line problem, circled in red color (Figure 7.14(b)). It is evident that this problem is due to the erroneous rectification. Similarly, building #13 of the Batikent data set has the worst rating of all the buildings in two data sets. This building contains two major problems, one is the shifted window (outlined in red), and the other is the lamppost that cannot be removed (outlined in cyan) (Figure 7.15(b)). It is believed that these failures are mainly caused by the problems occurred in the matching step of the proposed occlusion removal.



(a)

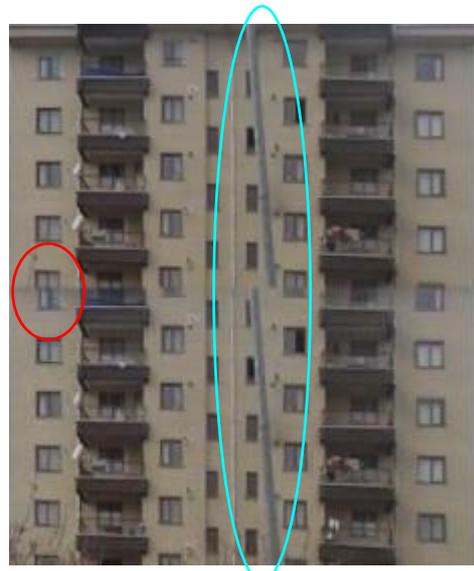


(b)

Figure 7.14. For the eTrims dataset, the facade images with (a) the best rating and (b) the worst rating with an example of a failure outlined in red.



(a)



(b)

Figure 7.15. For Batikent dataset, the facade images with (a) the best rating and (b) the worst rating with examples of failures outlined in red and cyan.

In addition to the experimental results, the running time calculations of the algorithms were also computed for the building facade texture acquisition approach. The running time for the extraction of one facade was found to be **1 minute** on average. Therefore, under the assumption of identical opposing facade textures of a building, the time required for extracting all the facades was about **2 minutes**. After that, the rectification performances were assessed. The proposed algorithm took approximately **15 seconds** for the rectification of a single facade and **30 seconds** for the rectification of the

whole building using the above assumption. The occlusion removal stage, which include the auto-cropping, image matching and post-processing steps, was also analyzed in terms of the computational performance. This stage took about **10 seconds** per facade and **20 seconds** for the entire building based on the aforementioned assumption and by excluding the user-intervention for manually selecting the occluded area. To sum up, the entire approach took about **3 minutes** (2 mins + 30 secs + 20 secs) for producing the rectified occlusion-free facade textures for a single building. As an overall performance, building facade texture acquisition took about **24 minutes** for the whole study area that contain a total of 110 single buildings falling within 8 building blocks under the assumption that the buildings in a block have identical appearance. In the evaluation of the performance, the operating system and the hardware configuration used were the same as in the 2-d building extraction and delineation approach.

In general, the developed approaches for building facade texture acquisition yielded satisfactory results. However, several shortcomings and limitations were also evident. One problem is the under-segmentation, which occurs during the facade image extraction process through Watershed segmentation. This problem is typically faced if the photographed facade is blocked by a large amount of occlusion or the facade texture is extremely heterogeneous due to the casting shadow or the non-uniform illumination. In each case, the segmentation algorithm may not be able to partition the facade image correctly and therefore, the accuracy of the extracted facade image drops. Figure 7.16 illustrates these two different cases. In the first case, the building facade is occluded by the vegetation, pedestrian, and a billboard (Figure 7.16(a)). In the second case, the facade texture has become heterogeneous due to the shadow cast on the facade of the building (Figure 7.16(b)). For both cases, the extracted patches are shown in Figures 7.16(c) and (d).



(a)



(b)



(c)



(d)

Figure 7.16. The problem of under-segmentation. (a)-(b) The original images, and (c)-(d) the extracted patches suffering from under-segmentation.

Under-segmentation negatively affects the further steps of the developed approaches. In the present case, the extraction of the vertical facade edges is the main concern since the vertical facade edges have a vital importance in the developed automatic rectification procedure. However, it was observed that in some cases the extraction of the vertical facade edges cannot be possible. One case was the existence of the occlusions and texture heterogeneity, which were already discussed above. The other case is arisen from the protrusions, such as pavilions and balconies on the facades. Since the facade extraction algorithm segments the protruded regions as the

foreground texture, the original facade edges are missed, therefore causing the fail of the automated rectification approach. An example case is illustrated in Figure 7.17. First, the original facade image given in Figure 7.17(a) is segmented. The result of the segmentation is given in Figure 7.17(b). Then, the vertical edges (Figure 7.17(c)) are detected by the Canny edge detector and the Hough lines (Figure 7.17(d)) are extracted. As can be seen in Figure 7.17(d), the Hough transformation was not able to detect all the vertical edges and therefore, some of the vertical edges were missed. This is due to the fact that the Hough transform merely preserves the strong edges. If the detected vertical edges are not enough for generating the trend lines to be used in the rectification procedure, the result of face image rectification would be inaccurate. Therefore, the proposed automatic rectification procedure would fail. Figure 7.17(e) illustrates the rectified facade image, for which the failure of the rectification procedure is obvious. For these reasons, the facade textures that belong to building block-7 were rectified manually.

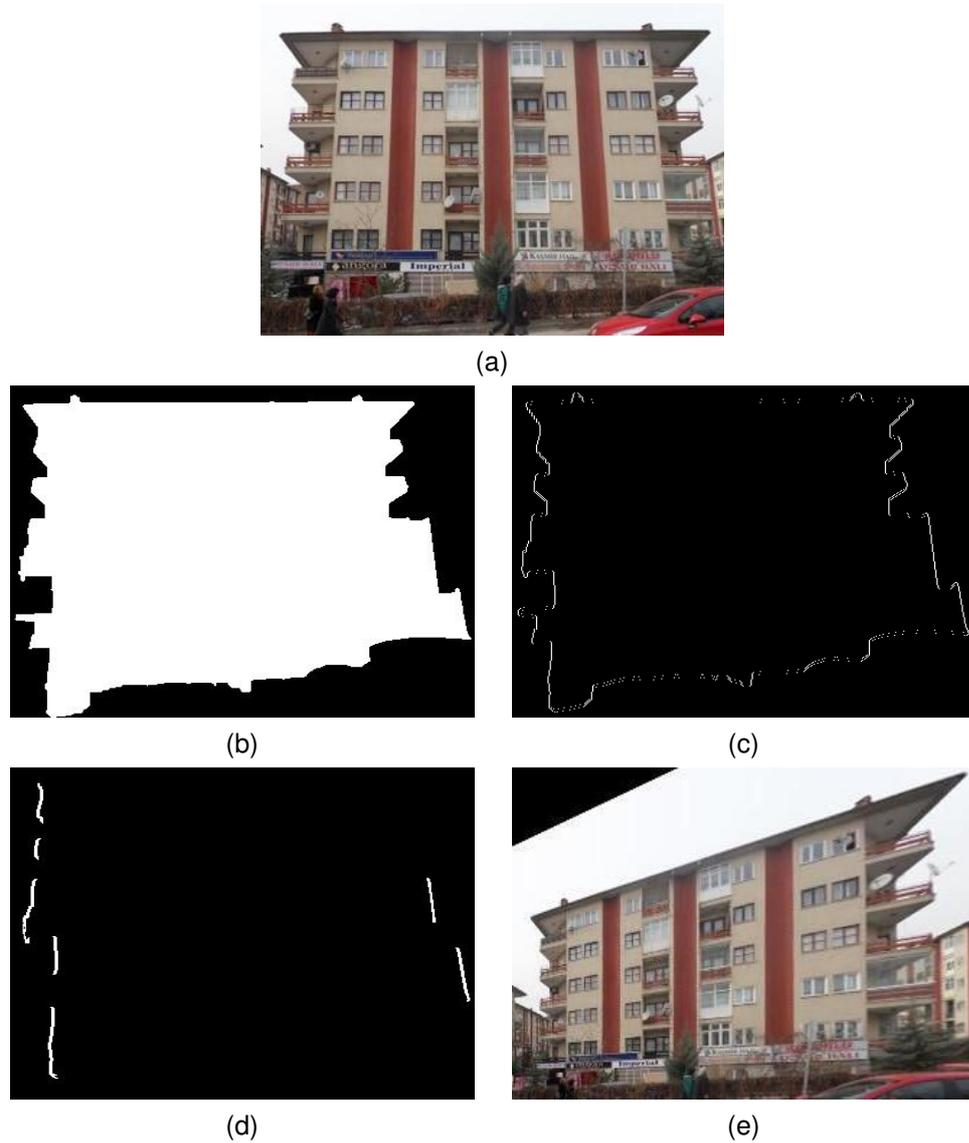


Figure 7.17. An example case showing the failure of the automated rectification approach. (a) The original image, (b) the result of facade extraction, (c) the detected edges by Canny operator, (d) the extracted Hough lines, and (e) the result of the rectification.

One other shortcoming of the building facade texture acquisition approach is the partial loss of the texture information due to the level of the building bases, which may be lower than the level of the pavement. In this case, the camera cannot record the undersides of the facades. Depending on the buildings, the size of texture loss can be small or as much as the whole story of a building. In the facade image of a building shown in Figure 7.18, a significant part (under the red line) cannot be acquired due to the height difference between the building baseline and the pavement.



Figure 7.18. A building facade image, in which the height difference between the building baseline and the pavement is evident.

Another shortcoming to be mentioned is the over-segmentation, which is caused by the facades of the attached buildings that have identical textures. In this case, the facades of the attached buildings are extracted as a single facade due to the texture similarity between two facades. During data collection, the facades of disjoint buildings should be attempted to be photographed separately. However, in dense settlement areas this may not always be possible. This shortcoming is illustrated in Figure 7.19, in which two buildings with identical textures are located back to back. Therefore, the facades of these buildings were extracted as a joint single facade image resulting in over-segmentation.



(a)



(b)

Figure 7.19. (a) Two buildings with identical textures that are located back to back and (b) the segmented facade with facade over-segmentation.

Another exceptional case is encountered in the developed occlusion removal approach. As described in section 5.4.2, the occluded area on a facade image is replaced by an occlusion-free patch detected based on a correlation-based image matching algorithm. However, the occlusion-free patch, which is fused into the occluded area, may sometimes be unrealistic. This is due to the fact that the structures, such as windows or doors may be shifted from their original locations on the occlusion removed image. Besides, some superfluous objects may appear on the image. In order not to increase the complexity, the integrity and consistency of the facades were not considered in the present study. These problematic cases are illustrated with an example in Figure 7.20. Note that, the original image shown in Figure 7.20(a) refers to a 5-story building, whereas the generated occlusion free image, which is shown in Figure 7.20(b), has an additional story at the bottom, outlined in red. In addition, the windows at the bottom are superfluous.



Figure 7.20. (a) The original image and (b) the generated occlusion-free image that contains superfluous information and incomplete structures.

In addition to above described shortcomings the developed texture acquisition approach has a couple of limitations. The first limitation is the shooting constraint of the facade photographs. As described in section 5.2.2, the facade extraction algorithm automatically seeds the initial foreground

markers to be located in the center of the image. To achieve this, the region of interest (the building facade) must be centered in the photograph. Otherwise, the segmentation algorithm may fail in extracting the building facade from the photograph. This is illustrated in Figure 7.21, in which the segmentation algorithm was not able to extract the building facade successfully (Figure 7.21(b)) as the building facade in the photograph (Figure 7.21(a)) was located off the center of the photograph.

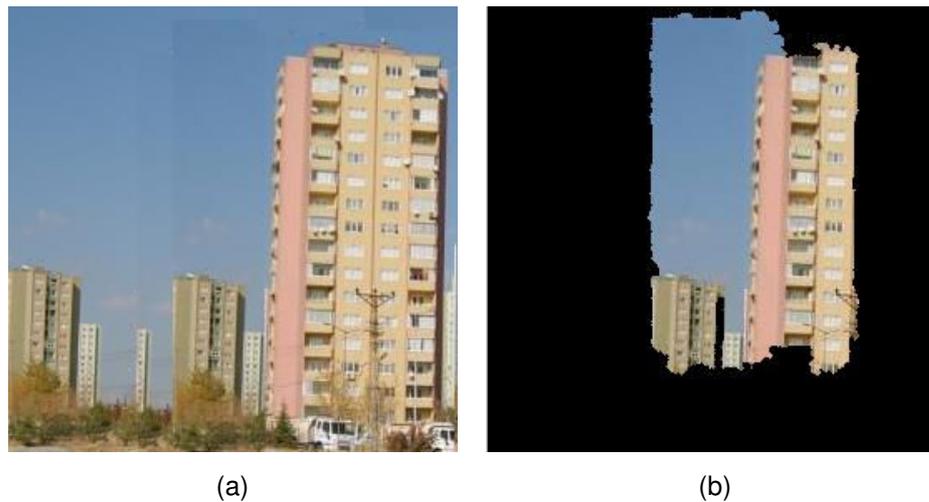


Figure 7.21. (a) A building facade located near the edge of the photograph and (b) the result of segmentation performed using the developed approach.

The second limitation is observed in the occlusion removal stage. For the building facades having heterogeneous textures the algorithm works more efficiently if the building facade contains a repetitive pattern. This is due to the fact that within a building facade with heterogeneous texture, the probability of finding a patch that is correlated to occluded patch is high. One other factor that affects the success rate of the occlusion removal algorithm is the number of story of the buildings. The higher the building is the more chance that a patch highly correlated to the occluded patch can be found. Another factor that affects the success rate of the occlusion removing algorithm is the occlusion size. In order to obtain successful results, the size of the occlusions should not be larger than one half of the building facade with respect to width or height. Otherwise, the developed occlusion removal

algorithm may not be able to detect an occlusion-free patch within the building facade image being searched.

The above described limitations are illustrated with an example in Figure 7.22. In Figures 7.22(a), (c), and (e) the red frames represent the occluded regions to be replaced with the occlusion-free counterparts. The building shown in Figure 7.22(a) is an example of a multistory building with a moderate occlusion size. It is evident that the pattern of the facade is repetitive. Therefore, as expected, the detected facade patch for replacing the occluded area was quite satisfactory (Figure 7.22(b)). On the building facade given in Figure 7.22(c) the occlusion spreads out over the surface of the facade. Although the facade pattern is repetitive and the building has multiple stories, the algorithm failed for finding a patch highly correlated to occluded area (Figure 7.22(d)). The building shown in Figure 7.22(e) has only two stories and the occlusion size is moderate. However, as in the previous case the result was not satisfactory since a patch highly correlated to occluded area was not able to be found (Figure 7.22(f)).



(a)



(b)



(c)



(d)



(e)



(f)

Figure 7.22. Examples for showing the limitations of the developed occlusion removal technique.

7.3. Results and Discussion for 3-D Model Generation and Texture Mapping

In this section, the quantitative or qualitative results were not obtained. Instead, the algorithm performances were evaluated in order to find the running times for generating the textured 3-d building models. The total elapse time for generating 3-d solid models together with the texture selection and mapping was computed to be about **3 seconds** per block and **24 seconds** for 8 blocks under the same operating system and hardware configurations used in the performance analyses of 2-d building extraction and delineation, and the building facade texture acquisition approaches.

A number of shortcomings and limitations were also faced during the generation of 3-d solid models and texture mapping. The first shortcoming is the brightness difference between the adjacent facades of the buildings. Since the acquisition of the facade images is carried out in different illumination conditions the global brightness of the images are tend to be different, therefore. This shortcoming is illustrated in Figure 7.23, in which the difference in the illumination of the adjacent facades is obvious (Figures 7.23(a) and (b)). Note that the average values for the blue image bands were computed to be 140 and 160 for the facade images (a) and (b), respectively.



(a)



(b)

Figure 7.23. The adjacent facades of a building with different illuminations.

The other shortcoming is the mismatching of the adjacent facade textures. This problem may occur due to several reasons. The first reason would be the shooting distance of a facade photograph. In general, the photos are taken from a constant distance. However, in some cases this may not be possible due to the obstacles. In these cases, the dimensions of the building facades photographed would tend to change due to shooting points with varying distances. This also affects the dimensions of the foreground textures being extracted. In particular, for the adjacent facade images of a single building, in which each facade is photographed from varying distances, a texture mis-matching problem likely occurs between the adjacent facade textures. As a consequence, this problem may reduce the visual quality of the textured building models. The second reason would be the varying level of the building bases. As discussed in the previous section, the textures of the adjacent facades may not be consistent due to different amount of texture that may be extracted for each facade. The third reason is the effect of

projective transformation used in the rectification stage. As is known, the size of an image after projective transformation may not be the same with the original size before the transformation. The amount of change is proportional with the amount of perspective distortion, in which highly distorted images would tend to diminish in size after the rectification. Since the effect of perspective distortions may not be same for each adjacent facade, the size of each rectified texture may change, therefore. This case also results with the texture mis-matching problem. Figure 7.24 illustrates a modeled building that suffers from the texture mis-match. It is also apparent that the number stories reflected by the facade textures are different.



Figure 7.24. A building suffering from a texture mismatch problem.

The last shortcoming is caused by the z-component (height) of 3-d building models. Due to errors contained by the normalized digital surface models, some of the building heights were not be determined. For example, for block-1 the height information was missing (Figure 7.25(a)). It is predicted that these deficiencies were caused by the matching problems occurred during the generation of nDSM. In the present study, for the buildings in block-1, the height information was estimated from the number of floors that the buildings contained. To do that, the height of each floor, which was assumed to be 2.5 m, was multiplied by the number of floors. That is, for a 10-story

building, the height was estimated to be 25 meters. The problematic patch of nDSM data that corresponds to block-1 is illustrated in Figure 7.25(a), while the corresponding R-G-B image is illustrated in Figure 7.25(b). Note that, on nDSM data the color at building locations appear black, which means that image correlation failed at these positions and therefore, produced zero elevation.

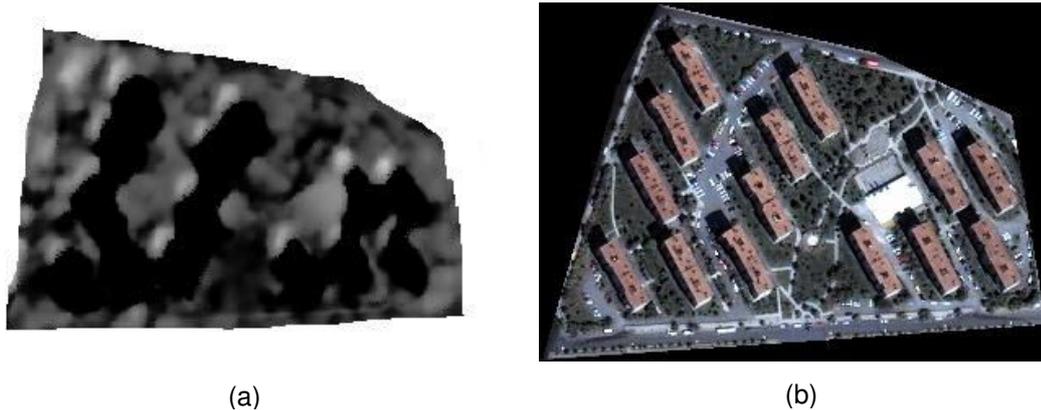
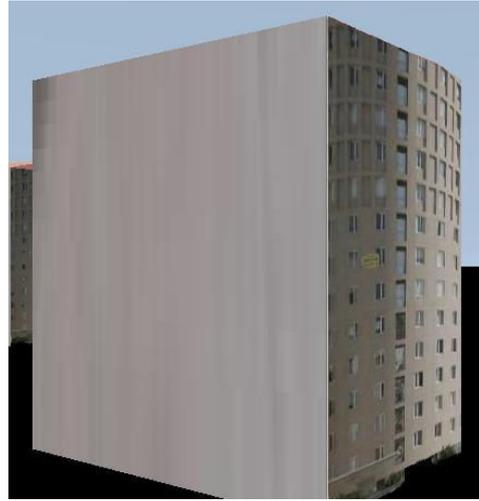


Figure 7.25. For building block-1, the (a) nDSM data and (b) the R-G-B image.

In addition to aforementioned shortcomings, the modeling part has an important limitation, which is related with the shape of the facade images to be mapped on 3-d models. Since the quadrilateral facades are considered in this study, only the rectangular-shaped textures are captured and processed, accordingly. The facade textures having different geometry such as polygonal, circular, etc. other than quadrilateral are kept out of the scope in the present study. As is known, all the techniques developed in the generation of the facade texture were already adapted to produce quadrilateral texture patches. As illustrated in Figure 7.26, a circular texture is forced to map onto a quadrilateral building facade, which results in an unrealistic view of the model.



(a)



(b)

Figure 7.26. Mapping a circular facade texture on a quadrilateral plane (a and b), which generates an unrealistic view of the model.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

In this chapter, first the conclusions of the developed integrated framework for 2-d building extraction and delineation, building facade texture acquisition, 3-d model generation, and texture mapping are presented. Next, the future directions for the study are discussed.

8.1. Conclusions for 2-D Building Extraction and Delineation

The developed approach for 2-d extraction and delineation of buildings from high resolution space imagery reveals the following conclusions:

- The conventional genetic algorithm-based approach for the extraction of 2-d building patches provided higher detection rates when compared to Fisher Linear Discriminant (FLD) classifier. It was observed that the detection rates of FLD hardly reached to the detection rates, which were generated in the first generation of the genetic algorithm-based approach.
- The detection rate of the conventional genetic algorithm was computed to be 91.86% (averaged over 10 runs) although the convergence rate was found to be relatively low. On the other hand, the proposed adaptive fuzzy-genetic algorithm converged earlier than the conventional algorithm within the first a few generations yielding an average fitness rate of 93.13%. The reason for the slow convergence of the conventional algorithm

- was due to the fixed initial probabilities of crossover and mutation operations, which highly increase the risk to get trapped in local minimum solution.
- The selection of the optimum initial values for the genetic algorithm parameters, such as the selection rate of 50%, the population size of 30, the chromosome size of 5, and the training/test sample sizes of 50 was found to be very efficient in reaching the maximum fitness values.
- When the accuracy of building delineation was evaluated at 95% confidence level, the mean errors were computed to be less than 4 meters for the total 110 buildings falling within 8 building blocks.
- The pre-processing stage before running the delineation algorithm eliminated the false alarm areas successfully. The morphological opening operation with a disk-shaped structuring element having a radius of **3** removed the isolated regions and small protrusions to a large extent. The radius of the structuring element was proved to preserve the orientation of the building patches. Further, the hole-filling morphological operation achieved the removal of every hole in all binary patches. In addition, the convex binary patch generation proved to simplify the protruded region boundaries before extracting the building footprints.
- It is observed that performing the processing and analysis operations one block at a time improved the building delineation accuracy by excluding the misclassified structures, complex-shaped buildings, or any artifacts, which were kept out of consideration in the present study.

- According to running time performances, the elapsed time of 24 minutes for the building extraction and the elapsed time of 16 seconds for the delineation of 2-d buildings were achieved for the whole study area that includes 110 single buildings. On the other hand, the determination of the optimum genetic algorithm parameters took 120 hours of running time, the step of which was performed once in the beginning of the methodology only.

8.2. Conclusions for Building Facade Texture Acquisition

The conclusions derived from the automatic extraction of the ground-level facade textures are as follows:

- For both data sets, the average quality percentage values of the facade image extraction were computed to be over 80% using a fully automated approach.
- In the pre-processing step, the analysis performed for the saturation component proved to reduce the non-uniform illumination on building facade images. After conducting several interactive experiments, the saturation coefficient of five was accepted to be the optimum value at present study. However, this value should not be considered global.
- In facade image extraction, the use of an initial marker template increased the level of automation in watershed segmentation by setting sufficient foreground and background markers at right locations.
- The “repetitive” aspect of watershed segmentation produced much more texture information than the conventional watershed algorithm. Further, the over-segmentation problem was overcome

to a large extent by means of intersecting the intermediate segments and using the segment of the intersections as the final segment.

- The developed automatic technique for texture rectification reduced the perspective distortions and yielded fairly encouraging positional accuracies. For the Batikent dataset, nine facade images out of fifteen were at the level of error bound under ten pixels. In the eTrims dataset, four out of five images yielded positional error rates no more than ten pixels. All assessments were performed at 95% confidence level.
- The combined use of Canny edge detector and Hough transformation was successful in detecting the vertical strong line segments of facades, which have a crucial importance in the rectification of the facade image. Besides, the best fitting lines (trend lines) were estimated from these line segments in order to fix the discontinuities and define the degree of distortion.
- The qualitative results of the occlusion removal were satisfactory. For the Batikent dataset, the average of the ratings was computed to be 2.58 staying in the middle of the criteria “Fine” and “Passable”. In parallel, the buildings in the eTrims dataset were included into “Fine” category having an average subjective rating of 2.28.
- The cropping operation, which was conducted as a pre-processing step in the occlusion removal procedure, was proved to reveal the foreground region (facade image) and discard the background region most of the time.
- In the texture mending stage, the correlation based image matching approach was found to be quite successful for finding

the occlusion-free counterpart of the occluded region. The experiments conducted revealed that the image matching process commonly terminates at the correlation rates ranging from **+0.90** to **+0.99** for the successful cases.

- Of the post-processing steps applied in occlusion removal, the motion blurring technique enhanced the results by removing the sharp transitions (seams) at the borders of the pasted texture patches. Moreover, applying an algorithm that is based on image differencing and weighted averaging reduced the illumination differences remarkably.
- The running time performance of facade image acquisition stage was computed. On average, the extraction of one facade was performed within a minute, while the rectification of a single facade took about 15 seconds. The occlusion removal of a single facade took approximately 10 seconds, as well.

8.3. Conclusions for 3-D Model Generation and Texture Mapping

The conclusions of 3-d solid model generation and texture mapping of the reconstructed models are as follows:

- It is evident that the previously extracted 2-d building footprints facilitated the determination of the coordinates (x,y) of the building patch corners and the mid-points of the edges. Further, the use of nDSM data was proved to be helpful in extruding the solid building models automatically by providing the approximate heights (z) of the buildings.
- In the modeling phase, the assumption “The buildings in a block have identical size” provided the elimination of certain positional

errors due to varying building dimensions. Moreover, this assumption improved the model generation performance to a large extent.

- The assumption made “The opposing facades have identical textures” was proved to be quite functional in texture mapping. This assumption not only improved the mapping performance in terms of speed, but also reduced the number of photos taken from the study area. In parallel, the coordinates of the shooting points collected by a manual GPS was also diminished.
- The analysis of the computed and measured bearings of the shooting points was found to be very successful in boosting the texture mapping accuracy. With the developed algorithm, all the facade textures were mapped properly onto solid models.
- The user-assisted roof modeling (by taking the realistic roof types and overhanging roofs into account) improved the level of reality of the building models. Further, the use of high-resolution roof texture patches also enhanced the quality of the building models.
- The reconstruction of a single building block in 3-d and texture mapping took about a few seconds. More generally, the overall running time performance of the photo-realistic 3-d building modeling of the study area, which contains a total of 110 single buildings falling within eight blocks, was computed to be about 50 minutes. The total elapsed time is accumulated by adding the elapsed times of 2-d building extraction and delineation (25 minutes), building facade texture acquisition (24 minutes) and 3-d model generation and texture mapping (24 seconds).

As a final concluding remark, the overall results revealed that a considerable progress has been made towards the automated reconstruction and representation of photorealistic large scale urban environments.

8.4. Recommendations

Although the proposed 3-d photorealistic building modeling approach demonstrates promising results, several priorities for future research remain. For instance, higher resolution satellite imagery or different data sources such as aerial photos or airborne laser scanning systems can be employed to extract complex buildings and detailed 3-d roof models with high quality roof textures as well as the city furniture such as roads, landmarks and vegetation.

Alternatively, the fusion of multi-source imagery to generate accurate and detailed building models with high quality textures, together with the high precision digital elevation / surface information for a more intelligent and robust perception should be the essential future research direction. Specifically, the use of ground-level laser scanning system might be helpful in detecting and removing the occlusions in an automated manner by using the range information.

New techniques will be required to facilitate the removal of the occlusions having different orientations and size. To do that, more sophisticated building structure analysis including automatic detection of windows and other structural elements can be deliberated. This additional information might enable the robust reconstruction of occluded building texture in more challenging cases.

Finally, mosaicing of facade textures will be an alternative research issue in the future. Generation of image mosaics for simple (planar) or complex (non-

planar) facades might be helpful in the extraction of the facade textures from the close-range digital image sequence.

REFERENCES

Ahmadi, S., Zoj, M.J.V., Ebadi, H., Moghaddam, H.A. and Mohammedzadeh, A., (2010), Automatic Urban Building Extraction from High Resolution Aerial Images Using an Innovative Model of Active Contours, *International Journal of Applied Earth Observation and Geoinformation*, 12, pp.150-157.

Aronoff, S., (1989), *Geographic Information Systems: A Management Perspective*, WDL Publications, pp. 133-148, Ottawa, CANADA.

Baltsavias, E.P., (2004), Object Extraction and Revision by Image Analysis Using Existing Geodata and Knowledge: Current Status and Steps Towards Operational Systems, *ISPRS Journal of Photogrammetry and Remote Sensing*, 58, pp.129-151.

Beucher, S. and Meyer, F., (1992), The Morphological Approach of Segmentation: The Watershed Transformation. In: *Mathematical Morphology in Image Processing*, E. Dougherty, Ed., chapter 12, pp. 433-481. Marcel Dekker, New York, USA.

Böhm, J., (2008), Facade Detail from Incomplete Range Data, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVII, Part B5, Beijing, CHINA.

Bresenham, J. E., (1965), Algorithm for Computer Control of a Digital Plotter, *IBM Systems Journal*, 4(1), pp.25-30.

Canny, J.F., (1986), A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), pp.679-698.

Carlberg, M., Andrews, J., Gao, P. and Zakhor, A., (2008), Fast Surface Reconstruction and Segmentation with Ground-Based and Airborne LIDAR Range Data, *Proceedings of 4th International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'08)*, Atlanta, Georgia, USA.

Cho, W., Jwa, Y.S., Chang, H.J. and Lee, S.H., (2004), Pseudo-Grid Based Building Extraction using Airborne LIDAR Data, Proceedings of International Society for Photogrammetry and Remote Sensing (ISPRS'04) Congress, Istanbul, TURKEY.

Cortona3D Viewer User's Guide, (2009), ParallelGraphics. *Retrieved 28 June 2010*, [Url: <http://www.cortona3d.com/Products/Cortona-3D-Viewer.aspx>]

Criminisi, A., Perez, P. and Toyama, K., (2004), Region Filling and Object Removal by Exemplar-Based Image Inpainting, IEEE Transactions on Image Processing, 13(9), pp.1200-1212.

David, P., (2008), Detection of Building Facades in Urban Environments, Proceedings of SPIE, Vol. 6978, pp: 139-148.

Efros, A.A. and Freeman, W.T., (2001), Image Quilting for Texture Synthesis and Transfer, Proceedings of SIGGRAPH 2001, Los Angeles, California, USA.

El-Hakim, S., Gonzo, L., Voltolini, F., Girardi, S., Rizzi, A., Remondino, F. and Whiting, E., (2007), Detailed 3-D Modeling of Castles, International Journal of Architectural Computing, 5(2), pp.199-220.

European Resettlement Fund, Batikent Project-TURKEY. *Retrieved 08 November 2007*,
[Url: <http://vcn.bc.ca/citizens-handbook/unesco/most/easteur1.html>]

Faugeras, O., Robert, L., Laveau, S., Csurka, G., Zeller, C., Gauclin, C. and Zoghlami, I., (1998), 3-D Reconstruction of Urban Scenes from Image Sequences, Computer Vision and Image Understanding, 69(3), pp.292-309.

Fraser, C.S., Baltsavias, E. and Gruen, A., (2001), 3-D Building Reconstruction from High-resolution Ikonos Stereo-imagery. In: Automatic Extraction of Man-Made Objects From Aerial and Space Images (III), Edited by E.P.Baltsavias, A.Gruen, and L.V. Gool, pp. 331-344, Balkema, Tokyo, JAPAN.

Frendendall, G.L. and Behrend, W.L., (1960), Picture Quality – Procedures for Evaluating Subjective Effects of Interference. Proceedings of Institute of Radio Engineers (IRE), vol. 48, pp.1030-1034.

Frueh, C., Sammon, R. and Zakhor, A., (2004), Automated Texture Mapping of 3D City Models With Oblique Aerial Imagery, Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'04), Thessaloniki, GREECE.

Frueh, C. and Zakhor, A., (2003), Constructing 3-D City Models by Merging Aerial and Ground Views, IEEE Transactions on Computer Graphics and Applications, Vol. 23(6), pp: 52-61.

Gamba, P. and Houshmand, B., (2002), Joint Analysis of SAR, LIDAR and Aerial Imagery for Simultaneous Extraction of Land Cover, DTM and 3D Shape of Buildings, International Journal of Remote Sensing, 23(20), pp. 4439-4450.

Gerke, M., Heipke, C. and Straub, B.M., (2001), Building Extraction from Aerial Imagery Using a Generic Scene Model and Invariant Geometric Moments, IEEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas, pp.85-89, Rome, ITALY.

GNU Image Manipulation Program, (2010), *Retrieved 14 August 2010*, [Url: <http://en.wikipedia.org/wiki/GIMP>]

Gonzalez, R.C., Woods, R.E. and Eddins, S.L., (2009), Digital Image Processing Using Matlab, 2nd Edition, Gatesmark Publishing, LLC, USA.

Gonzalez, R.C. and Woods, R.E., (2008), Digital Image Processing, 3rd Edition, Pearson Education Inc., New Jersey, USA.

Guillaso, S., Ferro-Famil, L., Reigber, A. and Pottier, E., (2005), Building Characterization Using L-Band Polarimetric Interferometric SAR Data, IEEE Geoscience and Remote Sensing Letters, 2(3), pp.347-351.

Haala, N. and Kada, M., (2005), Panoramic Scenes for Texture Mapping of 3D City Models, Proceedings of ISPRS WG V/5 'Panoramic Photogrammetry Workshop', Berlin, GERMANY.

Harris, C. and Stephens, M.J., (1988), A Combined Corner and Edge Detector. Proceedings of the 4th Alvey Vision Conference, pp.147-152, Manchester, ENGLAND.

Harvey, N.R., Theiler, J., Brumby, S.P., Perkins, S., Szymanski, J.J., Bloch, J.J., Porter, R.B., Galassi, M. and Young A.C., (2002), Comparison of GENIE and Conventional Supervised Classifiers for Multispectral Image Feature Extraction, IEEE Transactions on Geoscience and Remote Sensing, 40(2), pp.393-404.

Haupt, R.L. and Haupt S.E., (2004), Practical Genetic Algorithms, 2nd Edition, Wiley & Sons Inc. Publications, New Jersey, USA.

Herrera, F. and Lozano, M., (2003), Fuzzy Adaptive Genetic Algorithms: Design, Taxonomy and Future Directions, Soft Computing, Springer-Verlag, Vol.7, pp.545-562.

Hoegner, L. and Stilla, U., (2009), Thermal Leakage Detection on Building Facades Using Infrared Textures Generated by Mobile Mapping, Proceedings of 2009 Urban Remote Sensing Joint Event, Shanghai, CHINA.

Hofmann, A.D., Maas, H.G. and Streilein, A., (2002), Knowledge-based Building Detection Based on Laser Scanner Data and Topographic Map Information, Proceedings of the Photogrammetric Computer Vision, ISPRS Commission III Symposium, PCV'02, Graz, AUSTRIA.

Huertas, A. and Nevatia, R., (1988), Detecting Buildings in Aerial Images, Computer Vision, Graphics, and Image Processing, Vol.41, pp.131-152.

Inglada, J., (2007), Automatic Recognition of Man-made Objects in High Resolution Optical Remote Sensing Images by SVM Classification of Geometric Image Features, ISPRS Journal of Photogrammetry and Remote Sensing, 62(3), pp.236-248.

Ioannidis, C., Psaltis, C. and Potsiou, C., (2009), Towards a Strategy for Control of Suburban Informal Buildings Through Automatic Change Detection, *Computers, Environment and Urban Systems*, Vol.33, pp.64-74.

Irvin, R.B. and McKeown, D.M., (1989), Methods for Exploiting the Relationship Between Buildings and Their Shadows in Aerial Imagery, *IEEE Transactions On Systems, Man, and Cybernetics*, 19(6), pp.1564-1575.

Jang, K.H. and Jung, S.K., (2009), Practical Modeling Technique for Large-scale 3-D Building Models from Ground Images, *Pattern Recognition Letters*, Vol.30, pp.861-869.

Jaynes, C., Riseman, E. and Hanson, A., (2003), Recognition and Reconstruction of Buildings from Multiple Aerial Images, *Computer Vision and Image Understanding*, Vol.90, pp.68-98.

Jin, X. and Davis, C.H., (2005), Automated Building Extraction from High-Resolution Satellite Imagery in Urban Areas Using Structural, Contextual, and Spectral Information, *EURASIP Journal on Applied Signal Processing*, Vol.14, pp.2196-2206.

Joo, S.W., (2003), Lecture Notes on Linear Discriminant Functions and SVM, University of Maryland Institute for Advanced Computer Studies. *Retrieved 08 January 2011*, [Url: http://www.umiacs.umd.edu/~knkim/KG_VISA/LDA_SVM]

Kang, Z., Zhang, L., Zlatanova, S. and Li, J., (2010), An Automatic Mosaicking Method for Building Facade Texture Mapping Using a Monocular Close-range Image Sequences, *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(3), pp.282-293.

Karantzas, K. and Paragios, N., (2010), Large Scale Building Reconstruction Through Information Fusion and 3-D Priors, *IEEE Transactions on Geoscience and Remote Sensing*, 48(5), pp.2283-2296.

Karantzas, K. and Paragios, N., (2009), Recognition-Driven Two-Dimensional Competing Priors Toward Automatic and Accurate Building Detection, *IEEE Transactions on Geoscience and Remote Sensing*, 47(1), pp.133-144.

Kim, T., Lee, T.Y. and Kim, K.O., (2006), Semiautomatic Building Line Extraction from Ikonos Images Through Monoscopic Line Analysis, *Photogrammetric Engineering and Remote Sensing*, 72(5), pp.541-549.

Kim, Z. and Nevatia, R., (2004), Automatic Description of Complex Buildings from Multiple Images, *Computer Vision and Image Understanding*, Vol.96, pp.60-95.

Koc San, D., (2009), Approaches for Automatic Urban Building Extraction and Updating from High Resolution Satellite Imagery, PhD Thesis submitted to the Graduate School of Natural and Applied Sciences, Middle East Technical University, TURKEY.

Konushin, V. and Vezhnevets, V., (2007), Automatic Building Texture Completion, *Proceedings of GraphiCon'2007*, Moscow, RUSSIA.

Korah, T. and Rasmussen, C., (2007), Spatiotemporal Inpainting for Recovering Texture Maps of Occluded Building Facades, *IEEE Transactions on Image Processing*, 16(9), pp.2262-271.

Korč, F. and Förstner, W., (2009), eTRIMS Image Database for Interpreting Images of Man-Made Scenes. Technical report TR-IGG-P-2009-01, University of Bonn, Department of Photogrammetry, GERMANY.

Lafarge, F., Descombes, X., Zerubia, J. and Pierrot-Deseilligny, M., (2010), Structural Approach for Building Reconstruction from a Single DSM, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1), pp.135-146.

Laws, K. I., (1980), Rapid Texture Identification, *Proceedings of SPIE*, Vol. 238, pp.376–380.

Laycock, R.G., Ryder, G.D.G. and Day, A.M., (2007), Automatic Generation, Texturing and Population of a Reflective Real-Time Urban Environment, *Computers and Graphics*, Vol. 31, pp: 625-635.

Lee, D.H., Lee, K.M. and Lee, S.U., (2008), Fusion of Lidar and Imagery for Reliable Building Extraction, Photogrammetric Engineering and Remote Sensing, 74(2), pp.215-225.

Lee, D.S., Shan, J. and Bethel, J.S., (2003), Class-Guided Building Extraction from Ikonos Imagery, Photogrammetric Engineering and Remote Sensing, 69(2), pp.143-150.

Lillesand, T., Kiefer, R. and Chipman, J., (2007), Remote Sensing and Image Interpretation, 6th Edition, John Wiley and Sons Inc., New York, USA.

Lin, C., Huertas, A. and Nevatia, R., (1994), Detection of Buildings Using Perceptual Grouping and Shadows, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'94, pp.62-69, Seattle, USA.

Liu, Z., Cui, S. and Yan, Q., (2008), Building Extraction from High Resolution Satellite Imagery Based on Multi-scale Image Segmentation and Model Matching, Proceedings of 2008 International Workshop on Earth Observation and Remote Sensing Applications, Beijing, CHINA.

Liu, H., Xu, Z. and Abraham, A., (2005), Hybrid Fuzzy-Genetic Algorithm Approach for Crew Grouping, Proceedings of 5th International Conference on Intelligent Systems Design and Applications, ISDA'05, pp.332-337, Wroclaw, POLAND.

Lorenz, H. and Döllner, J., (2006), Towards Automating the Generation of Facade Textures of Virtual City Models, ISPRS Commission II, WG II/5 Workshop, Vienna, AUSTRIA.

Lu, Y.H., Trinder, J.C. and Kubik, K., (2006), Automatic Building Detection Using the Dempster-Shafer Algorithm, Photogrammetric Engineering and Remote Sensing, 72(4), pp.395-403.

Magellan Basic User Manual, (2006), Thales Navigation Inc., San Dimas, CA, USA.

MATLAB Online Documentation, (2010), Documentation for Mathworks Products. Retrieved 19 May 2010, [Url: <http://www.mathworks.com/help/techdoc>]

Mayer, H., (2008), Object Extraction in Photogrammetric Computer Vision, ISPRS Journal of Photogrammetry and Remote Sensing, 63(2), pp.213-222.

Mayer, H. and Reznik, S., (2007), Building Facade Interpretation from Uncalibrated Wide-baseline Image Sequences, ISPRS Journal of Photogrammetry and Remote Sensing, 61(6), pp.371-380.

Mayunga, S.D., Coleman, D.J. and Zhang, Y., (2007), A Semi-automated Approach for Extracting Buildings from Quickbird Imagery Applied to Informal Settlement Mapping, International Journal of Remote Sensing, 28(10), pp.2343-2357.

Michaelsen, E., Stilla, U., Soergel, U. and Doktorski, L., (2010), Extraction of Building Polygons from SAR Images: Grouping and Decision-level in the GESTALT System, Pattern Recognition Letters, Vol.31, pp.1071-1076.

Mohan, S. and Murali, S., (2007), Automated 3D Modeling and Rendering from Single View Images, Proceedings of International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), Vol.4, pp.476-480, Tamil Nadu, INDIA.

Müller, P., Zeng, G., Wonka, P. and Van Gool, L., (2007), Image-based Procedural Modeling of Facades, Proceedings of SIGGRAPGH 2007, San Diego, California, USA.

Noronha, S. and Nevatia, R., (2001), Detection and Modeling of Buildings from Multiple Aerial Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(5), pp.501-518.

Oriot, H., (2003), Statistical Snakes for Building Extraction from Stereoscopic Aerial Images, ISPRS Archives, Vol.XXXIV, Part 3/W8, Munich, GERMANY.

Ortin, D. and Remondino, F., (2005), Occlusion-Free Image Generation for Realistic Texture Mapping, Proceedings of the ISPRS WG V/4 Virtual Reconstruction and Visualization of Complex Architectures, Venice, ITALY.

Otsu, N., (1979), A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics, 15, pp.62-66.

PCI Geomatics, (2010), *Retrieved 14 August 2010*,
[Url: <http://www.pcigeomatics.com>].

Peng, J. and Liu, Y.C., (2005), Model and Context-driven Building Extraction in Dense Urban Aerial Images, International Journal of Remote Sensing, 26(7), pp. 1289-1307.

Perkins, S., Edlund, K., Esch-Mosher, D., Eads, D., Harvey, N. and Brumby, S., (2005), Genie Pro: Robust Image Classification Using Shape, Texture and Spectral Information, Proceedings of SPIE, Vol.5806. pp.139-148.

Perkins, S., Theiler, J., Brumby, S.P., Harvey, N.R., Porter, R., Szymanski, J.J. and Bloch, J.J., (2000), GENIE: A Hybrid Genetic Algorithm for Feature Classification in Multi-Spectral Images, Proceedings of SPIE, Vol.4120, pp.52-62.

Poullis, C. and You, S., (2009), Photorealistic Large-Scale Urban City Model Reconstruction, IEEE Transactions on Visualization and Computer Graphics, 15(4), pp.654-669.

Pu, S. and Vosselman, G., (2009), Knowledge-based Reconstruction of Building Models from Terrestrial Laser Scanning Data, ISPRS Journal of Photogrammetry and Remote Sensing, 64(6), pp.575-584.

Reddy, M., Iverson, L., Leclerc, Y. and Heller, A., (2001), GeoVRML: Open Web-based 3-D Cartography, Proceedings of the International Cartographic Conference (ICC2001), Beijing, CHINA.

Reid, I., (2003), Computational Geometry Lecture Notes. Robotics Research Group, University of Oxford. *Retrieved 23 March 2010*, [Url: <http://www.robots.ox.ac.uk/~ian/Teaching/CompGeom/lec2.pdf>].

Ripperda, N., (2008). Determination of Facade Attributes for Facade Reconstruction, Proceedings of International Society for Photogrammetry and Remote Sensing (ISPRS'08) Congress, Beijing, China.

Rottensteiner, F., Trinder, J., Clode, S. and Kubik, K., (2007), Building Detection by Fusion of Airborne Laser Scanner Data and Multi-spectral Images: Performance Evaluation and Sensitivity Analysis, ISPRS Journal of Photogrammetry and Remote Sensing, 62(2), pp.135-149.

Rottensteiner, F., (2003), Automatic Generation of High-Quality Building Models from Lidar Data, IEEE Computer Graphics and Applications, 23(6), pp.42-50.

Sahar, L. and Krupnik, A., (1999), Semiautomatic Extraction of Building Outlines from Large-Scale Aerial Images, Photogrammetric Engineering and Remote Sensing, 65(4), pp.459-465.

Samsung WB500 Review, (2009). *Retrieved 15 August 2010*, [Url: http://www.photographyblog.com/reviews/samsung_wb500_review]

Shackelford, A.K. and Davis, C.H., (2003), A Combined Fuzzy Pixel-Based and Object-Based Approach for Classification of High-Resolution Multispectral Data Over Urban Areas, IEEE Transactions on Geoscience and Remote Sensing, 41(10), pp.2354-2363.

Shufelt, J.A. and McKeown, D.M., (1993), Fusion of Monocular Cues to Detect Man-made Structures in Aerial Imagery. Computer Vision, Graphics and Image Processing: Image Understanding, 57(3), pp.307-330.

Simonetto, E., Oriot, H. and Garello, R., (2005), Rectangular Building Extraction from Stereoscopic Airborne Radar Images, IEEE Transactions on Geoscience and Remote Sensing, 43(10), pp.2386-2395.

Sirmacek, B. and Unsalan, C., (2009), Urban-Area and Building Detection Using SIFT Keypoints and Graph Theory, *IEEE Transactions on Geoscience and Remote Sensing*, 47(4), pp.1156-1167.

Sohn, G. and Dowman, I., (2007), Data Fusion of High-resolution Satellite Imagery and LIDAR Data for Automatic Building Extraction, *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(1), pp.43-63.

Song, Y. and Shan, J., (2004), Photorealistic Building Modeling and Visualization in 3-D Geospatial Information System, *Proceedings of International Society for Photogrammetry and Remote Sensing (ISPRS'04) Congress, Istanbul, TURKEY*.

Thiele, A., Cadario, E., Schulz, K., Thönnessen, U. and Soergel, U., (2007), Building Reconstruction from Multi-Aspect High-Resolution InSAR Data in Urban Areas, *IEEE Transactions on Geoscience and Remote Sensing*, 45(11), pp.3583-3593.

Tian, Y., Gerke, M., Vosselman, G. and Zhu, Q., (2010), Knowledge-based Building Reconstruction from Terrestrial Video Sequences, *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(4), pp.395-408.

Tournaire, O., Bredif, M., Boldo, D. and Durupt, M., (2010), An Efficient Stochastic Approach for Building Footprint Extraction from Digital Elevation Models, *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(4), pp.317-327.

Tsai, F., Lin, H.C. and Chen, C.H., (2007), Facade Texture Generation and Mapping Using Digital Videos, *ISPRS Journal of Photogrammetry and Remote Sensing*, 12(1), pp.83-92.

Tseng, Y. and Wang, S., (2003), Semiautomated Building Extraction Based on CSG Model-Image Fitting, *Photogrammetric Engineering and Remote Sensing*, 69(2), pp.171-180.

Tupin, F. and Roux, M., (2003), Detection of Building Outlines Based on the Fusion of SAR and Optical Features, *ISPRS Journal of Photogrammetry and Remote Sensing*, 58(2), pp.71-82.

Vincent, L. and Soille, P., (1991), Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6), pp. 583-598.

Vozikis, G., (2009), Comparison of Methods for Automated Building Extraction from High Resolution Image Data. *City Models, Roads and Traffic (CMRT09)*, International Archives of Photogrammetry and Remote Sensing, Vol. XXXVIII, Part 3/W4, Paris, FRANCE.

Vu, T.T., Yamazaki, F. and Matsuoka, M., (2010), Multi-scale Solution for Building Extraction from Lidar and Image Data, *International Journal of Applied Earth Observation and Geoinformation*, Vol.11, pp.281-289.

Wang, Y., Tupin, F. and Han, C., (2010), Building Detection from High-resolution PolSAR Data at the Rectangle Level by Combining Region and Edge Information, *Pattern Recognition Letters*, Vol.31, pp.1077-1088.

Wang, M., Bai, H. and Hu, F., (2008), Automatic Texture Acquisition for 3D Model Using Oblique Aerial Images, 1st International Conference on Intelligent Networks and Intelligent Systems (ICINIS 2008), pp.495-498, Wuhan, CHINA.

Wang, O., Lodha, S.K. and Helmbold, D.P., (2006), A Bayesian Approach to Building Footprint Extraction from Aerial LIDAR Data, *Proceedings of the International Symposium on 3D Data Processing, Visualization, and Transmission*, pp.192-199, NC, USA.

Wei, Y., Zhao, Z. and Song, J., (2004), Urban Building Extraction from High-Resolution Satellite Panchromatic Image Using Clustering and Edge Detection, *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, pp.2008-2010, Anchorage, Alaska, USA.

Wilczkowiak, M., Brostow, G.J., Tordoff, B. and Cipolla, R., (2005), Hole Filling Through Photomontage, *Proceedings of the 16th British Machine Vision Conference*, pp.492-501, Oxford, U.K.

Wu, J., Zhou, G., Ding, F. and Liu, Z., (2007), Automatic Retrieval of Optimal Texture from Aerial Video for Photo-realistic 3D Visualization of Street Environment, Proceedings of 4th International Conference on Image and Graphics, Chengdu, Sichuan, CHINA.

Xu, F. and Jin, Y.Q., (2007), Automatic Reconstruction of Building Objects From Multiaspect Meter-Resolution SAR Images, IEEE Transactions on Geoscience and Remote Sensing, 45(7), pp.2336-2353.

Zhang, K., Yan, J. and Chen, S.C., (2006), Automatic Construction of Building Footprints from Airborne LIDAR Data, IEEE Transactions on Geoscience and Remote Sensing, 44(9), pp.2523-2533.

APPENDIX A: SOURCE CODE OF THE 2-D BUILDING EXTRACTION AND DELINEATION

```
number_of_generations = ga_parameters(2);
population_size = ga_parameters(3);
number_of_genes = ga_parameters(4);
crossover_rate = ga_parameters(5);
mutation_rate = ga_parameters(6);

number_of_data_planes = 3; %Original data planes
number_of_scratch_planes = 4; %Temporary planes
gene_info_size = 7; % Maximum number of input & output planes

fprintf(fid, 'Population Size: %d\n',population_size);
fprintf(fid, 'Number of Genes: %d\n',number_of_genes);

%% Building initial scratch planes (All are composed of zeros)
for i=1:number_of_scratch_planes
    scratch(:,i) = study_area(:,1) - study_area(:,1);
end

building_samples = zeros(number_of_bina_samples, number_of_scratch_planes);
non_building_samples = zeros(number_of_non_bina_samples,
                             number_of_scratch_planes);

%% Building the initial population
population = zeros(population_size,number_of_genes);

for j=1:population_size
    for k=1:number_of_genes
        population(j,k) = 1 + round(rand*22);
    end
end
```

```

[row,col] = size(study_area(:,:,1)); % Size of the original image
study_area_size = size(study_area,3); % Size of the bands
output_plane = zeros(row,col);

% CONSTRUCTION OF GENE INFO MATRIX, initially all (-1)
gene_info = -ones(population_size, gene_info_size, number_of_genes);

for i=1:population_size
    for j=1:number_of_genes
        gene_info = generate_gene_info(population(i,j), scratch, study_area_size, gene_info, i, j);
        gene_info(i,1,j) = 1 + round(rand * (number_of_scratch_planes - 1));
    end
end

average_fitness_values = zeros(number_of_generations,1);
maximum_fitness_values = zeros(number_of_generations,1);

% GENETIC ALGORITHM CYCLE STARTS HERE
chromosome_fits = zeros(number_of_generations*population_size,1);
h = waitbar(0,'Genetic Algorithm is running...');
for generation=1:number_of_generations %Number of Generations
    total_time = number_of_generations * population_size * number_of_genes;
    chromosome_index = 1;
    view_count = 1;
    for i=1:population_size
        for j=1:number_of_genes
            scratch(:,:,gene_info(i,1,j)) = evaluate_genes(population(i,j), scratch, study_area,
                gene_info, i, j);
waitbar((((generation-1)*number_of_genes*population_size) + ((i-1)*number_of_genes) + j) /
total_time)
        end
        for k=1 : number_of_scratch_planes
            if (max(max(scratch(:,:,k)))==0) | (min(min(scratch(:,:,k)))==255)
                scratch(:,:,k) = study_area(:,:,1);
            end
        end
    end
end

```

APPENDIX B: SOURCE CODE OF BUILDING FACADE TEXTURE ACQUISITION

```
%Determining new markers for the foreground
while marker_count < max_marker_count
    rw = ceil(r*rand);
    cl = ceil(c*rand);
    if (yeni2(rw,cl) ~= 0)
        if (marker(rw,cl) ~= 1)
            marker(rw,cl) = 1;
            marker_count = marker_count + 1;
        end
    end
end

if e_trims == 1
    marker(2:2,4:c-2)=2;
    marker(2:c-2,r-2:c-2)=2;
    marker(r-floor(r/50):c-2,2:c-2)=2;
    marker(2:c-2,3:3)=2;
else
    marker(2:2,4:c-2)=2;
    marker(2:c-2,r-2:c-2)=2;
    marker(r-floor(r/6):c-2,2:c-2)=2;
    marker(2:c-2,3:3)=2;
end

end

disp('Applying watershed segmentation by flooding from marked sources...')
imout=watershedmex(g,marker);
imwrite(palettemex(imout),'watershed_output.bmp');

temp_mask = yeni2; % Previous Mask
yeni = imread('watershed_output.bmp');
yeni1 = rgb2gray(yeni);
```

```

thr = graythresh(yeni1);
yeni2 = im2bw(yeni1,thr);
yeni2 = 1 - yeni2; % Current Mask
yeni2 = or(temp_mask,yeni2);

mark = imread('marker_output.bmp');
mark1 = rgb2gray(mark);
overlapped = yeni1 + mark1;
segmented_texture(:,:,1) = im(:,:,1).* yeni2;
segmented_texture(:,:,2) = im(:,:,2).* yeni2;
segmented_texture(:,:,3) = im(:,:,3).* yeni2;
imwrite(segmented_texture, 'cropped.jpg');

L = bwlabel(yeni2);
stats1 = regionprops(L,'Area');
stats2 = regionprops(L,'Extent');
mask_library(:,iteration_number) = yeni2;
mask_stats(iteration_number,1) = (stats1(1).Area) / (r*c); % Area proportion in original image
mask_stats(iteration_number,2) = stats2(1).Extent; % Area proportion in MBB

if(mask_stats(iteration_number,1)) <= 0.5 %Ignore the small (less than 50%) regions
    mask_stats(iteration_number,2) = 0; %Nullify the extent
end

%Stopping Criteria
diff = abs(mask_stats(iteration_number,1) - mask_stats(iteration_number-1,1));
if(diff <= change_area_quantity)
    i_t = i_t + 1; % No or minor change
else
    i_t = 0; % Significant change
end
if(i_t == iteration_tolerance)
    iteration_number = ic+1; % Stop the iteration
else
    iteration_number = iteration_number + 1; % Carry on iteration
end

```

APPENDIX C: SOURCE CODE OF 3-D MODEL GENERATION AND TEXTURE MAPPING

```
min_photo = zeros(size(site_building_coords,1),7);
min_photo_dist = zeros(size(site_building_coords,1),1);

finish_photo_selection = 0; %Initially unfinished
short_edge_slot = 0; %Initially short building edge photo is empty
long_edge_slot = 0; %Initially long building edge photo is empty
selected_short_edge_photo_id = 0;
selected_long_edge_photo_id = 0;

for i=1:size(site_building_coords,1)
    for j=1:size(temp_building_midpoint_coords,1)
        dist = sqrt((site_building_coords(i,3) - temp_building_midpoint_coords(j,6))^2 ...
            + (site_building_coords(i,4) - temp_building_midpoint_coords(j,7))^2);
        if dist < min_dist
            min_dist = dist;
            min_index = j;
        end
    end
    min_photo(i,:) = temp_building_midpoint_coords(min_index,:);
    min_photo_dist(i) = sqrt((site_building_coords(i,3) -
        temp_building_midpoint_coords(min_index,6))^2 +
        (site_building_coords(i,4) - mp_building_midpoint_coords(min_index,7))^2);
    min_dist = 100000;
end

if reg_ID ~= 7
    while(finish_photo_selection == 0)
        [m_val m_ind] = min(min_photo_dist);
        photo_id = mod(site_building_coords(m_ind,2),100);
        edge_id = site_building_coords(m_ind,1);
        photo_easting = site_building_coords(m_ind,3);
        photo_northing = site_building_coords(m_ind,4);
```

```

degree_photo = site_building_coords(m_ind,5);
facade_easting = min_photo(m_ind,6);
facade_northing = min_photo(m_ind,7);

delta_X = photo_easting - facade_easting;
delta_Y = photo_northing - facade_northing;
temp_alpha = atan(delta_Y / delta_X);

if delta_Y >= 0 & delta_X >= 0 %First Quadrant
    alpha = temp_alpha
elseif delta_Y >= 0 & delta_X < 0 %Second Quadrant
    alpha = temp_alpha + pi;
elseif delta_Y < 0 & delta_X < 0 %Third Quadrant
    alpha = temp_alpha + pi;
elseif delta_Y < 0 & delta_X >= 0 %Fourth Quadrant
    alpha = temp_alpha + (pi*2);
end

degree_measured = (alpha/(2*pi)*360);
% The computed angle is approximated to one of the nearest 8-neighbor
if degree_measured > 0 & degree_measured < 45
    diff0 = degree_measured - 0;
    diff45 = 45 - degree_measured;
    if diff0 < diff45
        rounded_degree_measured = 0;
    else
        rounded_degree_measured = 45;
    end

elseif degree_measured > 45 & degree_measured < 90
    diff45 = degree_measured - 45;
    diff90 = 90 - degree_measured;
    if diff45 < diff90
        rounded_degree_measured = 45;
    else
        rounded_degree_measured = 90;
    end
end

```

CURRICULUM VITAE

PERSONEL INFORMATION

Surname, Name: Sümer, Emre
Nationality: Turkish (TC)
Date and Place of Birth: 2 February 1979, Ankara
Marital Status: Married
Phone: +90 312 234 10 10 / 1305
Fax: +90 312 234 10 51
Email: esumer@baskent.edu.tr

EDUCATION

Degree	Institution	Year of Graduation
Ph.D.	METU GGIT	2011
M.Sc.	METU GGIT	2004
B.Sc.	Baskent University Computer Engineering	2001
High School	Büyük Kolej, Ankara	1997

WORK EXPERIENCE

Year	Place	Enrollment
2005 – Present	Baskent University – Dept. of Computer Eng.	Lecturer
2001 – 2005	Baskent University – Dept. of Computer Eng.	Research Asst.

FOREIGN LANGUAGES

Advanced English

PUBLICATIONS

1. M.Turker and **Sumer, E.**, 2010, "Automatic Retrieval of Near Photo-realistic Textures from Single Ground-Level Building Images", Proceedings of International Conference on Geographic Object-based Image Analysis (GEOBIA 2010), Ghent, BELGIUM.
2. M.Turker and **Sumer, E.**, 2009, "Towards Automatic Extraction for 3D Building Modelling", Proceedings of 8th Annual Asian Conference and Exhibition on Geospatial Information, Technology and Applications (Map Asia 2009), Singapore, SINGAPORE.
3. **E.Sumer** and Turker, M., 2008, "Building Detection from High-Resolution Satellite Imagery using Adaptive Fuzzy-Genetic Approach", Proceedings of International Conference on Geographic Object-based Image Analysis (GEOBIA'08), Calgary, CANADA.
4. M.Turker and **Sumer, E.**, 2008, "Building-based Damage Detection due to Earthquake using the Watershed Segmentation of Post-event Aerial Images", International Journal of Remote Sensing, Vol.29, No.11, pp.3073-3089.

HOBBIES

Computer Technologies, Movies, Snooker, Bowling, Basketball