

Sadeq, Haval AbdulJabbar (2015) Merging digital surface models sourced from multi-satellite imagery and their consequent application in automating 3D building modelling. PhD thesis.

<http://theses.gla.ac.uk/6861/>

Copyright and moral rights for this thesis are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

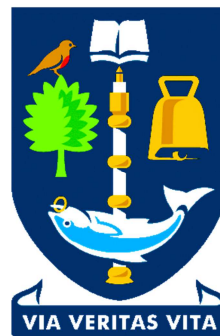
The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Merging Digital Surface Models Sourced from Multi-Satellite Imagery and their Consequent Application in Automating 3D Building Modelling

Haval AbdulJabbar Sadeq
BSc, MSc (Salahaddin University-Erbil)

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy



Geomatics Group
School of Geographical and Earth Sciences
College of Science and Engineering
University of Glasgow

December 2015

Abstract

Recently, especially within the last two decades, the demand for DSMs (Digital Surface Models) and 3D city models has increased dramatically. This has arisen due to the emergence of new applications beyond construction or analysis and consequently to a focus on accuracy and the cost.

This thesis addresses two linked subjects: first improving the quality of the DSM by merging different source DSMs using a Bayesian approach; and second, extracting building footprints using approaches, including Bayesian approaches, and producing 3D models.

Regarding the first topic, a probabilistic model has been generated based on the Bayesian approach in order to merge different source DSMs from different sensors. The Bayesian approach is specified to be ideal in the case when the data is limited and this can consequently be compensated by introducing the *a priori*. The implemented prior is based on the hypothesis that the building roof outlines are specified to be smooth, for that reason local entropy has been implemented in order to infer the *a priori* data. In addition to the *a priori* estimation, the quality of the DSMs is obtained by using field checkpoints from differential GNSS. The validation results have shown that the model was successfully able to improve the quality of the DSMs and improving some characteristics such as the roof surfaces, which consequently led to better representations. In addition to that, the developed model has been compared with the Maximum Likelihood model which showed similar quantitative statistical results and better qualitative results. Perhaps it is worth mentioning that, although the DSMs used in the merging have been produced using satellite images, the model can be applied on any type of DSM.

The second topic is building footprint extraction based on using satellite imagery. An efficient flow-line for automatic building footprint extraction and 3D model construction, from both stereo panchromatic and multispectral satellite imagery was developed. This flow-line has been applied in an area of different building types, with both hipped and sloped roofs. The flow line consisted of multi stages. First, data preparation, digital orthoimagery and DSMs are created from WorldView-1. Pleiades imagery is used to create a vegetation mask. The orthoimagery then undergoes binary classification into ‘foreground’ (including buildings, shadows, open-water, roads and trees) and ‘background’ (including grass, bare soil, and clay). From the foreground class, shadows and open water are removed after creating a shadow mask by thresholding the same orthoimagery. Likewise roads have been

removed, for the time being, after interactively creating a mask using the orthoimagery. NDVI processing of the Pleiades imagery has been used to create a mask for removing the trees. An 'edge map' is produced using Canny edge detection to define the exact building boundary outlines, from enhanced orthoimagery. A normalised digital surface model (nDSM) is produced from the original DSM using smoothing and subtracting techniques. Second, start Building Detection and Extraction. Buildings can be detected, in part, in the nDSM as isolated relatively elevated 'blobs'. These nDSM 'blobs' are uniquely labelled to identify rudimentary buildings. Each 'blob' is paired with its corresponding 'foreground' area from the orthoimagery. Each 'foreground' area is used as an initial building boundary, which is then vectorised and simplified. Some unnecessary details in the 'edge map', particularly on the roofs of the buildings can be removed using mathematical morphology. Some building edges are not detected in the 'edge map' due to low contrast in some parts of the orthoimagery. The 'edge map' is subsequently further improved also using mathematical morphology, leading to the 'modified edge map'. Finally, A Bayesian approach is used to find the most probable coordinates of the building footprints, based on the 'modified edge map'. The proposal that is made for the footprint *a priori* data is based on the creating a PDF which assumes that the probable footprint angle at the corner is 90° and along the edge is 180° , with a less probable value given to the other angles such as 45° and 135° . The 3D model is constructed by extracting the elevation of the buildings from the DSM and combining it with the regularized building boundary. Validation, both quantitatively and qualitatively has shown that the developed process and associated algorithms have successfully been able to extract building footprints and create 3D models.

List of publications

- Sadeq, H., Drummond, J. and Li, Z. (2012) “Evaluation of ASTER GDEM v. 2 using GPS checkpoints, OSGB DEM values and photogrammetrically derived DEMs,” in *XXII Conference of International Society of Photogrammetry and Remote Sensing*. Melbourne, Australia, pp. 295–300.
- Sadeq, H., Drummond, J. and Li, Z. (2014) “Bayesian Approach to Building Footprint extraction and 3D Modelling from High-resolution Satellite Imagery,” in *Proceedings of the GIS Research UK 22nd Annual Conference*. Glasgow-UK.

The above paper was awarded The Whittles Prize for the “best paper integrating Remote Sensing and GIS”, on a conference participant vote.

- Sadeq, H., Drummond, J. and Li, Z. (2014) “Bayesian approach to building footprint extraction and 3D modelling from high resolution satellite imagery,” *South-Eastern European Journal of Earth Observation and Geomatics*, 3(2S), pp. 185–189.
- Sadeq, H., Drummond, J. and Li, Z. (2015) “Building Footprint and 3-D Modelling using Bayesian Approaches and Satellite Images” submitted to *Photogrammetric Engineering & Remote Sensing*, under review.

Table of Contents

Abstract.....	II
Table of Contents	V
List of Tables	X
List of Figures.....	XI
Acknowledgment.....	XVII
Author's Declaration	XVIII
Abbreviations	XIX
Chapter 1 Introduction.....	1
1.1 Motivation.....	1
1.2 Problems Associated with Creating DSMs	5
1.3 Scope of the Research	8
1.4 Further Merging Considerations	9
1.5 Aim of the Research.....	10
1.6 Objectives of the Research.....	10
1.7 3D Building Modelling	11
1.8 Implemented Software	12
1.9 Thesis Structure	13
Chapter 2 Research Background.....	14
2.1 Overview of Satellite Images	14
2.2 DSM Generation Algorithms	14
2.3 Data Merging	17
2.3.1 DSM Merging	17
2.3.2 Multitemporal Effect.....	23
2.3.3 Multiresolution Data	24
2.3.4 Data Co-Registration.....	25
2.3.5 Probabilistic Method in Merging including maximising Entropy	26
2.4 Building Extraction Literature	33
2.4.1 Building Extraction from Satellite Imagery	33
2.4.2 Building Extraction from Aerial Imagery	44
2.4.3 Building Extraction from UAVs	45
2.4.4 Building Extraction from LiDAR	46
2.4.5 Building Extraction from DSMs	48
2.4.6 Building Extraction from Satellite and LiDAR	50
2.4.7 Building Extraction from Satellite and Maps.....	50
2.4.8 Building Extraction from LiDAR and Aerial Imagery	51
2.4.9 Building Extraction from DSM and Aerial Imagery.....	52

2.5 Summary	53
Chapter 3 Digital Surface Model (DSM)	55
3.1 DSM Format	56
3.2 Techniques for DSM Creation	57
3.2.1 Geodetic Survey	58
3.2.2 LiDAR Survey	58
3.2.3 Photogrammetry	59
3.2.4 InSAR	60
3.2.5 Digitizing Topographic Maps	61
3.3 Study area and Acquired Images Specification	62
3.3.1 Satellite Image Acquiring Geometry	64
3.3.2 WorldView-1 Satellite Imagery	66
3.3.3 Pleiades satellite imagery	67
3.3.4 RPC file	68
3.4 Ground Control Points	70
3.5 Satellite Image Processing	76
3.5.1 Interior Orientation, Exterior Orientation and Triangulation	77
3.6 DSM Generation	81
3.6.1 NGATE Technique	81
3.6.2 Area Based Matching	85
3.6.3 Edge Based Matching Technique	86
3.6.4 DSM Quality	86
3.7 Orthoimagery Generation	87
3.7.1 Quality of Orthoimagery	88
Chapter 4 Bayesian Statistics	90
4.1 Probability	90
4.2 Random Variables	91
4.3 Probability Distribution Functions	91
4.4 Statistical Inference Approaches	93
4.4.1 Likelihood Approach	94
4.4.2 Frequentist Approach	96
4.4.3 Bayesian Statistics	97
4.5 Bayes' Rule	97
4.6 Bayesian Inference	99
4.7 Bayesian Parameters	101
4.8 <i>A priori</i> Data	103
4.8.1 Subjectivity in <i>a priori</i> Data	104
4.8.2 Uniform <i>a priori</i> Data	104

4.8.3 Conjugate Prior	105
4.8.4 Improper Prior Distribution	105
4.9 Posterior Distribution.....	106
4.10 Data Inference.....	107
4.11 Applying a Bayesian approach	112
4.11.1 Bayesian Fusion Approach	112
4.12 Comparison between Bayesian and Frequentist Approaches	113
4.13 Summary	115
Chapter 5 Merging Digital Surface Models (DSMs) Using Bayesian Approaches	116
5.1 Test Site	117
5.2 Digital Surface Model (DSM) Quality Assessment.....	120
5.2.1 Blunders	123
5.3 DSM Formation Model.....	129
5.4 Merging DSMs.....	129
5.4.1 Merging Using Maximum Likelihood Method.....	131
5.4.2 Merging Using Bayesian Approach	134
5.5 Estimating the <i>a priori</i> using Maximum Entropy	140
5.6 Prior Optimization	142
5.7 The Implementation of Merging	143
5.8 Summary	143
Chapter 6 Result and Validation of Merged Digital Surface Models.....	145
6.1 Reference Data.....	145
6.2 Evaluation Indices.....	145
6.3 DSM Quantitative Assessment	148
6.3.1 Statistical Assessment.....	149
6.3.2 Correlation Assessment.....	152
6.4 DSM Qualitative Assessment	156
6.4.1 Slope Analysis	163
6.5 Summary	167
Chapter 7 Building Footprint and 3D Model Generation.....	169
7.1 Mathematical Morphology.....	170
7.1.1 Erosion	172
7.1.2 Dilation	173
7.1.3 Opening and Closing.....	173
7.2 Data Preparation.....	174
7.2.1 Image Segmentation.....	175
7.2.2 Producing nDSM.....	179
7.2.3 Uniquely Labelling nDSM Buildings	182

7.2.4	Shadow Detection	184
7.2.5	Road Extraction.....	185
7.2.6	Vegetation Area Masking	186
7.3	Producing an Edge Map.....	187
7.4	Building Detection	192
7.4.1	Enhancing Segmented Image	192
7.4.2	Detect Building Boundary	194
7.5	Building extraction.....	196
7.5.1	Extract initial Building Boundary	197
7.5.2	Sorting Initial Building Boundary.....	197
7.5.3	Approximate Initial Building Boundary	199
7.6	Building Footprint Construction	201
7.6.1	Modifying the Edge Map	201
7.6.2	Construct Individual Base Boundary	204
7.6.3	Building Boundary Optimization.....	204
7.7	3D Model Building Construction.....	214
7.8	Summary of the Developed Process and Associated Algorithms	215
7.9	Preliminary Assessment.....	217
7.10	Conclusion	220
Chapter 8 Result and Evaluation of Building Footprint and 3D Modelling		221
8.1	Study Area and Implemented Data	221
8.2	Reference Data.....	222
8.3	Qualitative Assessment for Building Footprints	223
8.4	Quantitative Assessment of Building Footprint.....	226
8.4.1	Quantitative Evaluation Indices	227
8.4.2	Corner Validation.....	227
8.4.3	Building's Centre Validation	230
8.4.4	Area Validation	232
8.5	Quality Property Assessment	233
8.6	3D model assessment	236
8.7	Conclusion	240
Chapter 9 Conclusions and future work.....		241
9.1	Conclusion	241
9.2	Future Work	243
9.2.1	Further work for Merging DSMs	243
9.2.2	Further work for building footprint extraction.....	245
9.3	Revisiting Objectives	247
References.....		248

Appendix A-C++ Code for merging Digital surface model using Maximum Likelihood approach	270
Appendix B-C++ Code for merging Digital surface model using Bayesian approach.....	273
Appendix C-C++ Code for labelling nDSM.....	278
Appendix D-C++ Code for Building footprint Extraction processes	281
Appendix E- Building footprint measured data.....	305

List of Tables

Table 1-1 Satellite image classification with respect to their resolution.....	4
Table 1-2 Characteristics of popular commercial optical very high resolution satellite image sensors. (Pan = panchromatic - a single band over the visible parts of the electromagnetic spectrum,; MS = multispectral - usually three or more bands in narrow wavelength ranges within the visible and near visible, including infra-red, parts of the electromagnetic spectrum)	4
Table 2-1 Comparison of Different DSM generation techniques.	16
Table 2-2 List of the summarised Building Extraction rate (BE) based on the data source.	44
Table 3-1 Characteristics of the satellite imageries used in the research.....	64
Table 3-2 List of the GCPs used in the triangulation of satellite imagery	71
Table 3-3 Comparison of Static and RTK measurement using different observation durations and at three different points. At the mid time of the field work for three different locations.....	73
Table 3-4 Static and RTK measurements made at the same point for different periods of time.....	74
Table 5-1 The quality characteristics of the two DSMs using the available checkpoints. A +ve discrepancy means the GNSS elevation is above the Pleiades or WV-1 elevation.....	128
Table 6-1 quality characteristics of the two digital surface models using the checkpoints. Red text indicates that the error in the point is greater than 1.96 x RMSE or outside the 95% confidence interval.....	151
Table 6-2 Merged and original slope map statistical analysis.	167
Table 8-1 Buildings Corners' planimetric quantitative statistical result from measuring 144 building corner points.	228
Table 8-2 X and Y Corners discrepancies of buildings' - statistical results obtained from measuring 144 building corner points.	229
Table 8-3 Planimetric statistical result for the discrepancy (m) at each building centre for 35 buildings.	231
Table 8-4 Individual X and Y coordinate discrepancies (m) of building centres - quantitative results for 35 buildings.	231
Table 8-5 Quantitative results for the discrepancies (m ²) of the building areas for 35 buildings.	232
Table 8-6 Quality Property Assessment	235
Table 8-7 3D statistical test using different sources for elevation for buildings with flat (one direction sloped) roofs only, showing discrepancies (d) obtained against measured points using a total station. Total number of measured buildings is 9.....	237
Table 8-8 3D statistical test using different sources for elevation for buildings with hipped roofs only, showing discrepancies (d) obtained against measured points using a total station. Total number of measured buildings is 26	237
Table 8-9 3D Statistical test using different sources for elevation for all buildings (hipped and one direction sloped roofs), showing discrepancies (d) obtained against measured points using a total station. Total number of measured buildings is 35.....	237
Table E-1 Corner planimetric discrepancies of true and extracted buildings.....	305
Table E-2 Corner planimetric discrepancies in X and Y direction of true and extracted buildings	306
Table E-3 Centre planimetric discrepancies in X and Y direction of true and extracted buildings.	307
Table E-4 measured area of the true and extracted buildings.....	308
Table E-5 Measured height of the true and extracted buildings of hipped buildings with hipped and one direction sloped roofs.....	309
Table E-6 Measured height of the true and extracted buildings of hipped roofs.....	310
Table E-7 Measured height of the true and extracted buildings of one direction slope roofs.....	310

List of Figures

Figure 1.1 Definition given by CityGML for different types of level of detail (LoD) for 3D model (Gröger and Plümer, 2012).	11
Figure 2.1 3D residuals map showing the effect of multitemporal data on the different DSMs (Papasaika et al., 2008).	23
Figure 2.2 Effect of co-registration of SRTM and ASTER DEM profiles, respect to the reference data (a) the ASTER DEM and SRTM profiles before the co-registration (b) the final profiles situation after co-registration operation.	25
Figure 2.3 Graph shows the value of the entropy for pairs of probabilities (Shannon, 1948).	32
Figure 2.4 The building extraction process from satellite imagery by the voting algorithm (Kim et al., 2005). Top row: the fragmented detected edge using the Burns et al. (1986) method. Bottom row: the constructed edges of the long side of the building.	34
Figure 2.5 Process of the Circular Cast Algorithm as it draws contour for extracting a building from satellite imagery (Theng, 2006).	35
Figure 2.6 Result of extracted buildings from high resolution satellite images (Liu et al. 2008).	35
Figure 2.7 Different sites showing extracted buildings, as reported by San and Turker (2010) from satellite imagery using the Hough transform, in which 80% of buildings were detected.	38
Figure 2.8 Building footprint extraction result using satellite imagery proposed by Dahiya et al. (2013), extracted buildings (purple) obtained from the applied algorithm compared with manually extracted buildings (black line).	39
Figure 2.9 Constructed 3D site using satellite imagery as shown in the research by Krauss et al. (2007).	41
Figure 2.10 Buildings extraction using the BUS (Building Unit Shape) algorithm implementing IKONOS imagery. The top row represents the satellite image, and bottom row represents the extracted building superimposed on the satellite image (Sohn and Dowman, 2001).	42
Figure 2.11 Applying artificial neural network for building footprint extraction (a) the data set used in experiment (b) the result of applying the algorithm (Lari and Ebadi, 2007).	43
Figure 2.12 Building construction implementing UAV images (a) 3D building model based on single UAV image (Jizhou et al., 2004) (b) LoD2 building using Google Sketchup software based on model created from UAV images.	45
Figure 2.13 Recommended algorithm by Wang et al.(2006) to extract buildings from LiDAR data using the Bayesian approach.	46
Figure 2.14 Automatically extracted buildings from LiDAR data as proposed by Dorninger and Pfeifer (2008).	48
Figure 2.15 Exploiting DSM for 3D city modelling based on object approach (Lafarge et al., 2008).	48
Figure 2.16 Building footprint extraction from satellite DSM using ‘energy’ modelling (Tournaire et al., 2010).	49
Figure 2.17 Urban 3D modelling using 3D satellite images’ data and maps (Tack et al., 2012).	51
Figure 2.18 Perspective view for the constructed model from LiDAR covered with texture from aerial imagery (Kokkas, 2008).	52
Figure 3.1 The comparison between DSM and DTM/DEM/DGM/DHM/DTEM.	56
Figure 3.2 Digital terrain model format categories (Li et al., 2005).	57
Figure 3.3 The principle of acquiring the ground coordinates using LiDAR system (ref: http://www.imagingnotes.com).	59
Figure 3.4 InSAR technique for DSM generation: (a) SRTM showing the used waves in DSM Generation http://www2.jpl.nasa.gov/srtm/instrumentinterferometry.html ; (b) the schematic diagram for a single spaceborne pass (SRTM). H is the sensor altitude, B is the baseline length, D is	

the slant range and δD is the slant range difference, θ is the look angle and z is the topographical point height (Abdelfattah and Nicolas, 2002)	61
Figure 3.5 The acquired satellite images for Glasgow WV-1 (background) and Pleiades satellite imagery (foreground), with the study area boundary superimposed in red.	63
Figure 3.6 Types of satellite sensor image scanning explaining along track (a) whisk-broom acquisition principle, (b) using push-broom acquisition, (c) panoramic image acquisition.	65
Figure 3.7 An illustration about determining the RPC using cube, shown object coordinate and image coordinate (Dowman et al., 2012).	69
Figure 3.8 Numbered and measured Ground Control Points (GCPs) and Checkpoints (CPs) over the study area, which are used as control points and quality assessment points, respectively.	70
Figure 3.9 The distribution of GCP and Checkpoints with respect to the GNSS Reference Station in the study area	75
Figure 3.10 a schematic diagram illustrating the main components of DPW (Agouris et al., 2004).	76
Figure 3.11 A sample of the generated DSM: (a) the Generated DSM from Satellite imagery, 0.5m resolution, using the NGATE approach; (b) the corresponding part of the DSM is shown on the satellite imagery.	83
Figure 3.12 Imagery for 3D point extractions experiment: (a) imagery showing complex buildings and flat surfaces with little variability; (b) the 3D points produced with the NGATE algorithm, the cyan line vertices are representing the ground truth points, the red dots are representing the 3D points produced by the NGATE algorithm. The algorithm successfully extracted points at highly variable locations or at the edges, while it failed at the locations of low variability (Zhang et al., 2007).....	85
Figure 3.13 The effect of using different resolution DSMs in orthoimage production: (a) orthoimage produced using a low resolution 10m DSM, where the building edges are straight; (b) orthoimage produced using a high resolution 50cm DSM, where the building edges are ‘wavy’.....	88
Figure 4.1. Gaussian Probability Density Function with mean (μ) equal to zero and variance (σ^2) equal to 1.....	92
Figure 4.2 Uniform <i>a priori</i> probability distribution, for the value of θ for interval [26,1500](Held and Bove, 2014).	105
Figure 4.3 The components and result of Bayesian statistics, <i>a posteriori</i> (or Posterior), likelihood and <i>a priori</i> (or Prior) distributions (Berry, 1997).....	107
Figure 4.4 Multi mode <i>a posteriori</i> probability (from Bernacchia, 2014).	109
Figure 4.5 the <i>a posteriori</i> distribution showing the conditional distribution and its relation to the joint marginal distribution (http://www.statisticalengineering.com/).	110
Figure 4.6 Illustration of Highest density interval (HDI) for measuring the parameter uncertainty (Kruschke, 2011).	111
Figure 5.1 Flow chart for DSM merging process.....	117
Figure 5.2 Study area and study data used in merging DSMs: (a) Pleiades satellite imagery with resolution 50cm; (b) WorldView-1 satellite imagery 50cm; (c) DSM 1m resolution produced from Pleiades stereo imager; and, (d) DSM with 1m resolution from WorldView-1 stereo satellite imagery.	119
Figure 5.3 The study area indicating the checkpoints used for quality assessment.....	123
Figure 5.4 Original DSMs error distribution histograms, DSM points against checkpoints in Table 5-1 (a) Pleiades DSM against RTK checkpoints, the histogram of the error appears not to be distributed normally (b) WV-1 DSM against RTK checkpoints, the histogram of the error appears to be distributed normally (c) the q-q plot for Pleiades discrepancy against measured checkpoints (d) the q-q plot for WV-1 discrepancy against measured checkpoints.	126
Figure 5.5 the explanation of probability parameters using Maximum Likelihood method (Zisserman, 2007).....	133
Figure 5.6 DSM shows different value of elevations used to represent the entropy (a) the entropy is low since the randomness in the elevations is high (b) the entropy is high since the variance in the	

elevation is low.	140
Figure 5.7 Hyperpixel size 3x3 (green pixels), that is used to find the <i>a priori</i> elevation (red pixel) and is passed over the DIGITAL SURFACE MODEL (grey pixels).	142
Figure 6.1 Validation checkpoints, the 31 checkpoints that have been used in the validation stage distributed over study area.	146
Figure 6.2 Comparison of the correlation, scatter plot and histograms for the input data and merging results using Maximum Likelihood and Bayesian techniques against checkpoints (CPs), where: the left column represents the correlation of the original and merged DSMs heights against CP heights; the middle column represents the scatter plot of the CP discrepancies and the original or merged DSMs, relative to the ellipsoidal height; and, the right column represents the frequency of the CP discrepancies (errors).	155
Figure 6.3 Orthoimagery for the study area indicating the specified profile location.....	156
Figure 6.4 Profile along line A-B as shown in Figure 6.3, showing the merging results using Maximum Likelihood, Bayesian approaches and the data sets: Pleiades DSM and WorldView-1 DSM. The Y-axis is elevation (m) and the X-axis distance (m) along the AB profile.....	158
Figure 6.5 Profile over the line A-B as shown in Figure 6.3. It shows the comparison between the merging results using the Bayesian approach with range $\pm 0.1\text{m}$. The Y-axis is elevation (m) and the X-axis distance (m) along the AB profile.	159
Figure 6.6 Profile over the line A-B as shown in Figure 6.3 it shows the comparison between the merging results using Bayesian approach with simulation range $\pm 0.25\text{m}$. The Y-axis is elevation (m) and the X-axis distance (m) along the AB profile.	160
Figure 6.7 A profile located over two checkpoints marked with red dots, being located at points No 26 and 29 in Table 6-1.....	161
Figure 6.8 Profile over the line A-B as shown in Figure 6.7. It shows the comparison between the merging results using the Bayesian approach with simulation range $\pm 0.25\text{m}$, and its relation with the checkpoints. The Y-axis is elevation (m) and the X-axis distance (m) along the A-B profile, passing through the checkpoints 26 and 29	162
Figure 6.9 Orthoimagery and 3D view to examine the effect of the merging over a flat surfaced structure – outlined in yellow.	163
Figure 6.10 Slope map analysis for merged DSMs, the white symbolization shows the effect of merging on removing the slope.	164
Figure 6.11 slope map visualisation for merged DSMs.	165
Figure 7.1 Flow chart illustrating the process for building footprint extraction, with the heading numbers at the left of the diagram.	170
Figure 7.2 Different types and size of structural elements with different location for reference points, red is foreground pixel, white background pixels and green is the reference point in the structural element.....	172
Figure 7.3 Applying Mean shift on orthoimagery for study area produced from WV-1 satellite imagery (a) original orthoimagery after applying sharp and smoothing operation (b) applying mean shift algorithm with radius 5 and colour difference 5 on image shown at ‘a’.....	177
Figure 7.4 Thresholded image using global thresholding method(a) thresholding the enhanced orthoimagery using minimum error thresholding (b)thresholding the enhanced orthoimagery after applying the mean shift algorithm, the noise has been removed and the holes have been eliminated.	178
Figure 7.5 Segmenting orthoimagery using different thresholding algorithms: (a) Minimum Error threshold, the buildings and roads are detected, except some buildings with high intensity; (b) Moments threshold, the high intensity buildings are detected, which had not been detected with the method used in the Figure 7.5(a).....	179
Figure 7.6 Process of producing an nDSM using MM: (a) 3d representation of the kernel, size 49x49, that is used in the smoothing with $\sigma=5$ pixels; (b) 3d representation of the kernel, size 49x49, that is used in the smoothing with $\sigma=10$ pixels; (c) DSM produced from merging satellite images; (d) applying a 49x49 kernel to the original DSM with $\sigma=10$ pixels; (e) thresholded nDSM - the top of	

the buildings have been detected; (f) thresholded DSM, the buildings on the upper left has been merged with higher elevation ground.	181
Figure 7.7 Overlaying the thresholded nDSM on orthoimagery: (a) a small part of the building has been detected in the nDSM; (b) nDSM has covered over two houses; (c) a part of the nDSM covers the building partially; (d) a green/vegetated area has been detected.	182
Figure 7.8 Object labelling process. (a) The msk used to label the region. Green is the reference pixel - a foreground pixel and red pixels are the connected neighbour pixels, used to check the connectivity of the pixels. (b) The original binary image with objects before labelling, both the objects having the same label. (c) Applying first pass of labelling, first object to the left has two labels, while the right has only one label. (d) Applying the second pass of the labelling on the object, the left object has only one label and the right object takes no different label.....	183
Figure 7.9. The result of labelling nDSM: a different colour value is given to each different object label, for better visualisation.....	184
Figure 7.10 Detecting and subtracting shadows (a) orthoimagery shows the shadow area (b) the detected shadows from the orthoimagery, (c) the refined shadows after applying Mathematical Morphology, leading to the production of solid objects.....	185
Figure 7.11 the digitized road using ArcGIS software.....	186
Figure 7.12 Green Mask production (a) Pleiades sensor's imagery including multispectral and infrared bands (b) NDVI map produced from using NDVI index (c) the vegetated (green) area which results from applying Minimum Error threshold on the NDVI map and overlaid over the Pleiades imagery.	187
Figure 7.13 Flow chart for Canny edge detection process.	189
Figure 7.14 The filters used to produce gradient map (a) gradient in x-direction (b) gradient in y-direction (Nixon and Aguado, 2008).	189
Figure 7.15 Illustrating non-maximum operation process (from Nixon and Aguado, 2008).	190
Figure 7.16 Illustrating non-maximum operation process with hysteresis thresholding (from Nixon and Aguado, 2008).....	191
Figure 7.17 producing edge map (a) the study area used to produce Canny edge map (b) result of applying Canny edge detection on 'a', using Gaussian kernel radius=1, low threshold=1 high threshold =2 in the study area	192
Figure 7.18 Process of refining the segmented orthoimagery which is produced from applying Minimum error thresholding (a) the output of subtracting the roads NDVI and shadows from the first segmented image which is obtained in sec 7.2.1 (b)Image erosion applying circular structural element(c) Using MM opening operation, apply structural element erosion twice and dilation twice, using square structural element(d) Image dilation applying circular structural element.	193
Figure 7.19 Process of refining the segmented orthoimagery which is produced from applying moment thresholding: (a) the output of subtracting the roads, NDVI and shadows from the second segmented image; (b) using MM opening operation, apply structural element erosion twice and dilation twice, using the square structural element, the noise has been eliminated and the objects separated.	194
Figure 7.20 The process of building detection. (a) The labelled nDSM overlaid on orthoimagery. (b) The segmented image. (c) The remaining corresponding part of the segmented image after dilating the labelled nDSM image. (d) The object that will be used in building extraction.....	195
Figure 7.21 The process of extracting building boundary: (a) the extracted building obtained from segmented image in the binary form; (b) the initial building boundary's pixels obtained from applying mathematical morphology operations on the binary image in (a).....	197
Figure 7.22 Two different types of buildings show the representation of the initial building's boundary points after converting them from pixel wise to vector. It shows the initial building's boundary sorted, based on the pixel readings.....	198
Figure 7.23 Sorted vectorised initial building's boundary vectors, obtained from applying the Nearest Neighbour algorithm.	199
Figure 7.24 Applying the Ramer-Douglas-Peucker algorithm to approximating the initial building	

boundary points that were produced in Figure 7.23 and using these as the initial building boundary.....	200
Figure 7.25 Explanation of the detected edge on the orthoimagery; (a) the orthoimagery that shows hipped building (b) the edge map which produced using orthoimagery with a lot of extra edges radius Gaussian=1, lower threshold=1, upper threshold=2 (c) the edge map which produced using same orthoimagery with very intense amount of edges, with radius Gaussian=0.5, lower threshold=0.1, upper threshold=0.2 (d) the edge map which produced using same orthoimagery with very few edges, with radius Gaussian=1, lower threshold=1, upper threshold=5.....	202
Figure 7.26 Comparisons of modified edge map (a) the original Canny map (b) the refined Canny map.	203
Figure 7.27 The cleaning filter which is used to remove the noise in the edge map. This filter traverses the edge map, and if all the red pixels overlies the background pixels of the edge map then any foreground pixels that are overlaid by green pixels will be removed from the edge map.	203
Figure 7.28 Modifying the edge map for the optimization purpose (a) Canny edge map (b) filling the gap by adding the thresholded image boundary to the Canny edge map (c) keeping the solo boundary by selecting the intersection point of the centre ray with the boundary in the image (b).....	204
Figure 7.29 The result of optimizing the boundary using Maximum Likelihood method (a) simulated boundary and constructed individual base boundary based on Canny edge map (b)the simulated boundary overlaid on the orthoimagery (c) the result of Maximum Likelihood method.....	207
Figure 7.30 The result of applying Maximum Likelihood method to regularize building footprints over the case study area.	207
Figure 7.31 Diagram showing the distance from selected pixel to the simulated boundary.	208
Figure 7.32 The prior probability function $f(\alpha)$ that used to find the penalty, 90° and 180° degrees gives the minimum penalty while the penalty for 45° and 135° is higher, and even higher for other angles.	210
Figure 7.33 (a) Initial building boundary overlaid over the edge map, the corner angles and the distances from the solo boundary to the initial building boundary can be seen; (b) result of regularized initial building boundary which is obtained by seeking the situation generating the minimum value for Eq. 2.	212
Figure 7.34 The result of applying Bayesian Approach to regularize building footprints over the case study area.	212
Figure 7.35 Comparison of initial and optimized building boundary: (a)&(d) initial and optimized building boundary sample overlaid over the edge map; (b)&(e) initial and optimized building boundary overlaid over individual base boundary; (c)&(f) initial and optimized building boundary overlaid over orthoimagery.	213
Figure 7.36 Vector map showing the result of applying Bayesian theorem (red) to regularize building boundary lines. Green lines represent the initial building boundary.	214
Figure 7.37 3D models for the study area based on extruding the building footprint based on maximum DSM height in the specified area.	215
Figure 7.38 Study area-1 and study area-2 inset in the WorldView-1 satellite imagery.....	218
Figure 7.39 Segmented image showing the effect of smoothing by replacing the median with its surrounding boundary (a) before enhancement (b) after enhancement. (Also see Figure 7.40) ..	219
Figure 7.40 Alternative study area used in applying the building footprint construction algorithm (a) orthoimagery used in the test (b) extracted building footprint overlaid on the orthoimage.....	220
Figure 8.1 The study area that has been used to test building extraction (a) study area-1, (b) study area-2	222
Figure 8.2 Transforming the British coordinate system (OSGB 1936) to UTM-30North.	223
Figure 8.3 The blue line is OS master map produced using transformation option in ArcGIS overlaid on the WV-1 orthoimagery. The green dots represent the measured field points with total station. Crown Copyright and Database Right 2015. Ordnance Survey.....	223
Figure 8.4 The extracted building footprint by applying Bayesian approach.	224

Figure 8.5 The extracted buildings in the study area.	225
Figure 8.6 Building Extraction Assessment (a) The buildings which were rotated or incorrectly constructed (b) true and incorrectly constructed buildings with their centre points.....	225
Figure 8.7 The original surveyed building with total station (blue) and the extracted buildings (black) in the study area.	226
Figure 8.8 The constructed and true building corners, with planimetric corner discrepancy measurements.....	228
Figure 8.9 The building corner error (discrepancy or residual) distribution for 144 corner measurements.....	229
Figure 8.10 The constructed and true building corners, with corner discrepancy measurements in x and y directions. (All dimensions and coordinates are meters).....	229
Figure 8.11 The corner error distribution of the buildings from measuring 144 corners.	230
Figure 8.12 The discrepancies between the constructed and true Centre of the buildings for 35 buildings.	231
Figure 8.13 The Centre error distribution of the 35 buildings.	232
Figure 8.14 The error distribution of the building areas for 35 buildings.....	233
Figure 8.15 The second study area: (a) the numbering system for true building areas overlaid on the extracted buildings; (b) the true buildings digitized using ArcGIS; (c) the extracted buildings overlaid on the true buildings.	234
Figure 8.16 The height error distribution and Cumulative Distribution Function (CDF) of the buildings: left column the histogram of error for the flat roofed (one-side slope) buildings - total 9 samples; middle column for the hipped roof buildings - total 26 samples; and, the right column for all buildings (one direction slope and hipped roof buildings) - total 35 samples.....	238
Figure A.1 The used input data in the merging (a) first input DSM from Pleiades with cell size 1m(b) second input DSM for WorldView-1 with cell size 1m.....	270
Figure B.1 A screen shot of merging process using Bayesian approach.	277
Figure B.2 The input data used in the building footprint extraction and 3D modelling.....	281
Figure C.1 A screen shot of the output of building footprint and 3D modelling is shown in text format.	304

Acknowledgment

First and foremost, I can't find words to express my gratitude to my Supervisor Dr. Jane Drummond. I would like to give immense thanks for her patience, support, encouragement and honourable contribution. Secondly, thank you to Professor Zhenhong Li for his supervision and advice during my study. Also I would like to express my gratefulness to Mrs. Anne Dunlop and Mr. Kenny Roberts for their assistance during field work and the School of Mathematics and Statistics at the University of Glasgow for coordinating my completing their Bayesian Statistics course, thereby increasing my knowledge of Bayesian approaches.

Thank you to my father and mother who raised me and helped me to reach to this level. I would like to show appreciation to my brothers, sisters, wife and my son who enthuse my life. Also thanks to relatives, friends and colleague who continuously contacted and encouraged me during my study.

I would like to express my thankfulness to colleagues and staff at the esteemed University of Glasgow especially the School of Geographical and Earth Sciences for their welcoming environment.

Also, I want to give my great thanks to my sponsor Kurdistan Regional Government for extensive support that has been provided during my study, without which I would not be able to come to the UK for the postgraduate study.

Finally, I would like also acknowledge Digital Globe for providing the WorldView-1 data and Airbus Defence and Space Ltd (Astrium) for making a special offer for providing the Pleiades data.

Disclaimer

Airbus Defence and Space Ltd, Geo-Intelligence have not repeated the experiments reported in this thesis and therefore cannot validate or endorse the results given in the thesis or the processing software used in the research.

Author's Declaration

I declare that the work recorded in this dissertation is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution.

Haval A. Sadeq

Abbreviations

ε	Noise or error in the DSM
H_f	value of the entropy
Δh	The difference between the measured checkpoint by GNSS and the checked elevation (i.e. usually the elevation obtained from the DSM)
$\operatorname{argmax} p(. .)$	Argument of the maximum of the probability
ASTER GDEM	Advanced Spaceborne Thermal Emission and Reflection Radiometer Global Digital Elevation Model
BDP	Buildings Detected Percentage
BE	Building Extraction rate
CInSAR	Cross-Interferometric Synthetic Aperture Radar
CM	Completeness rate
CPs	Checkpoints
DEM	Digital Elevation/Earth Model
DSM	Digital Surface Model
DTM	Digital Terrain Model
GIS	Geographic Information System
GNSS	Global Navigation Satellite System
IMU	Inertial Motion Unit/Inertial Mapping Unit
INS	Inertial Navigation System
InSAR	Interferometric Synthetic Aperture Radar
LiDAR	Light Detection and Ranging
LoD	Level of Detail
LSM	Least Squares Method
MAP	Maximum A Posteriori
$ML(\Theta)$	Maximum Likelihood of the required parameter
MLE	Maximum Likelihood estimation
MM	Mathematical Morphology
NDVI	Normalized Difference Vegetation Index
NGATE	Next Generation Automatic Terrain extraction
OSGB	Ordnance Survey of Great Britain

$P(x \Theta)$	Likelihood, probability of the unknown parameter (x) given the parameter value (Θ) used to represent the available data in probability form
$P(\Theta)$	Prior probability, or <i>a priori</i> , used to represent the prior information
$P(\Theta x)$	Posteriori, the probability of the unknown random variable given the experiment data
PCA	Principle Component Analysis
QP	Quality Percentage
r^2	determination coefficient
RADAR	Radio Detection And Ranging
RMSE	Root Mean Square Error
RPC	Rational Polynomial Coefficient
RTK	Real Time Kinematic
SAR	Synthetic Aperture Radar
σ	Standard Deviation
SOCET-GXP	SOftCopy Exploitation Toolkit - Geospatial eXploitation Products
SRTM SAR-X	Shuttle Radar Topography Mission Synthetic Aperture Radar band X
TIN	Triangular Irregular Network
TV	Total Variation
UAV/ UAS	Unmanned Aerial Vehicle/System
UTM	Universal Transverse Mercator
WGS	World Geodetic System

Chapter 1 Introduction

A Digital Surface Model (DSM, plural DSMs) represents the surface elevation of the ground including, most importantly, the features above it, such as roads, buildings, trees, and even smaller features such as parked cars and roadside bins. DSMs play a critical role in various applications including planning, 3D urban city maps, civilian emergencies, natural disaster management (e.g. flooding, earthquake, and landslides); military activities; airport management; and geographical analysis, such as in the geographies of health, crime and hazards (Saeedi and Zwick 2008). Moreover, rapid population increase, in developing countries such as Iraq and in particular Kurdistan, has led to a need to find the most cost-effective methods for modelling and mapping urban sites, among which building footprints have been shown to be extremely critical in planning and infrastructure development. DSMs can also be used to produce 3D models of an area.

The increasing efficiency of computers has led to the automation of much of the work that was previously achieved manually (Smith and Wagner, 2005), thereby increasing output and reducing execution time. Increased output has been achieved in the surveying field, notably in the production of DSMs.

The Bayesian approaches differ from the classical or frequentist methods in several important ways leading to their popularity (Berry, 1997; FDA, 2010; O'Hagan, 2004). The main reason for this greater popularity is that the Bayesian approaches accept prior information concerning the problem to hand. This feature has the following advantages: allows the use of smaller sample sizes; provides more informative results than frequentist method; and, determines the uncertainty of unknown parameters in addition to those parameters whose values are actively sought (FDA, 2010).

This research addresses the twin problems of increasing the quality of a DSM resulting from the merging of several DSMs and increasing the quality of building footprints automatically extracted from such a DSM, with particular attention being paid to the Bayesian approaches.

1.1 Motivation

The applications of satellite images have been, and still are, expanding into different fields especially in remote surveillance, including those outlined in the following

paragraphs (Dial et al., 2003).

Image analysis: this aids national or international intelligence, environmental observation, emergency response, etc. It is possible to use raw images with a low resolution at a low cost (Lillesand et al., 2015). Images can be provided as panchromatic or colour, for visual analysis (Olsen, 2007). It is also possible to obtain an infrared band, which is useful for vegetation based, and other, remote sensing applications (Dial et al., 2003).

Cartography: this involves designing map products. To implement this it is required to extract the information from the imagery related to particular maps such as topographical, hydrological, transportation and other different themes. It is assumed that the data gathered is three-dimensional, and for this purpose, a stereo image is required to obtain three-dimensional coordinates for the features of interest, which are needed for any 3-D geographic information systems (GISs), and contour maps (Poli et al., 2007).

Orthorectified Images: this product is crucial input to orthoimage maps, cartographic feature extraction and GIS database construction (Eisenbeiss et al., 2004). In many ways orthoimage maps are interchangeable with traditional maps with their specified regular scale, marginalia, and grid, except that traditional map data have undergone some selection, are represented by icons (or symbols) and vectors which have been corrected planimetrically with respect to the terrain. From orthoimagery all necessary data can be digitized, as required for the application purpose.

Digital Elevation Models (DEMs): although considered a significant product for assessing the danger of flooding, airport safety, planning and other applications it is also a vital component in orthorectifying imagery (Li et al., 2005). DEMs can be produced from stereo imagery; generally they are either edited to represent the 'bare' surface of the earth without buildings and trees, called DTMs, or left with building, trees, etc., and called DSMs. The first is useful for hydrological analysis and orthoimagemap production, while the second is useful for line-of-sight analysis such as applied at airports and by the military, and in planning and civil engineering in general (Li et al., 2005).

The applications that have been used in this research were limited to DSM production and orthoimagery, which will be illustrated in more detail in the following chapters. Presently available automated methods for creating DSMs and building footprints are facing difficulties with respect to cost, accuracy, acceptable building shape and extracting small buildings. Established methods for producing accurate footprints are often very labour intensive, using architectural blueprints, skilled personnel and much time to achieve an outcome. Source data currently used to extract building footprints accurately and with some level of automation, such as LiDAR and aerial images, are expensive and difficult to get. Satellite imagery is increasingly free (e.g. GLOVIS, 2013 URL: www.glovis.usgs.gov), and, assuming this trend continues then effective methods for its use must be developed.

Examining high-resolution commercial satellite imaging services, to find an alternative for costly LiDAR or aerial image data is valuable. The developments in satellite images have led to an ‘anytime and anywhere’ provision for many applications, and for some applications, can be available for many years. In addition, they can be available at a lower price per unit area and with better access to remote or restricted regions than aerial photography can provide. Based on the assumption that images taken from different directions and at different times, but of the same scene, will produce DSMs of varying characteristics, the proposed method aims to merge DSMs, from different images in order to produce an improved DSM and hence footprint. Specifically it is hoped this will overcome problems of DSM production such as: absence of sufficient texture; distinct object discontinuities; local object patches not being planar; repetitive objects; occlusions; moving objects; and, radiometric artefacts including specular reflections (Zhang and Gruen, 2006). All these problems will be considered.

The application of DSMs to the creation of topographic databases has become an important issue. A particular focus has been the use of DSMs for change detection, especially changes related to planning applications (Alobeid and Jacobsen, 2008) for which a requirement is the identification of accurate and well-defined building objects. When using remotely sensed imagery for the extraction of quantitative information, accuracy and clear definition is fundamental to the success of most applications.

According to the literature, satellite images have been classified according to the resolution of the sensors, such as Low, Medium, High and Very High resolution see

Table 1-1. Although different researchers have used different terms for specific resolutions, the most widely used terms, suggested by Dowman et al. (2012) as follows:

Table 1-1 Satellite image classification with respect to their resolution.	
Low Resolution	$\geq 30\text{m}$ and $< 300\text{m}$
Medium Resolution	$\geq 5\text{m}$ and $< 30\text{m}$
High Resolution	$\geq 1.0\text{m}$ and $< 5\text{m}$
Very High Resolution	$< 1.0\text{m}$

Table 1-2 shows the high resolution satellite optical image sensors that are available for providing imagery of the Earth's surface, based on sensing reflectance in the visible and near visible part of the electromagnetic spectrum and listing only Very High Resolution (see: Table 1-1) imagery. Only passive sensors are included. From the table it is clear that the maximum resolution (at the time of writing) of civilian use satellite imagery ('Res'. In Table 1-2) now reaches 0.31m, as has been achieved by the recently launched WorldView-3 (launched in August 2014). Since Very High Resolution satellite images have been used in this research, for that reason the table below is focused only on that type, although higher resolution data from other sources such as Aerial imagery or LiDAR are available.

Table 1-2 Characteristics of popular commercial optical very high resolution satellite image sensors. (Pan = panchromatic - a single band over the visible parts of the electromagnetic spectrum,; MS = multispectral - usually three or more bands in narrow wavelength ranges within the visible and near visible, including infra-red, parts of the electromagnetic spectrum)					
Satellite	Organization Country	Launch Date	Bands Pan/MS	Res. Pan/MS (m)	Height (km)
WorldView-1	DigitalGlobe, USA (DigitalGlobe Inc., 1992)	18 Sep, 2007	1/-	0.45/-	496
WorldView-2	DigitalGlobe, USA (DigitalGlobe Inc., 1992)	8 Oct, 2009	1/8	0.46/1.8	770
WorldView-3	DigitalGlobe, USA (DigitalGlobe Inc., 1992)	13 Aug, 2014	1/28	0.31/1.24	617
GeoEye-1	GeoEye Inc., USA (DigitalGlobe Inc., 1992)	6 Dec, 2008	1/4	0.41/1.65	684
Quick Bird	DigitalGlobe, USA (DigitalGlobe Inc., 1992)	18 Oct, 2001	1/4	0.61/2.44	450
IKONOS	GeoEye Inc., USA (DigitalGlobe Inc., 1992)	24 Sep, 1999	1/4	0.82/3.2	681
Pleiades-1A+ Pleiades-1B	Airbus Defence and Space France (Astrium Services, 2013)	16 Dec, 2011	¼	0.7/2.8	694
SkySat-1	Skybox Imaging-USA	21 Nov, 2013	1/4	0.9/2.0	572
SkySat-2	Skybox Imaging- USA	8 Jul, 2014	1/4	0.9/2.0	625
KOMPSAT 3	Korea Aerospace Research Institute- South Korea (SI Imaging Services, 2013)	17 May, 2012	1/4	0.7/2.8	685

The most significant information that can be seen in Table 1-2 is pixel resolution and band numbers. The highest satellite image resolution is from WorldView-3, which is 0.31m. All satellite images, except WorldView-1 images which are available in panchromatic only, provide infrared bands in addition to the colour (or visible) bands (blue, green, red). However, as shown in Table 1-2, satellite sensors Worldview-3 are providing even more bands than other sensors; for example, in addition to the panchromatic and visible bands, Worldview-3 provides eight short-wave infrared and 12 atmospheric compensation bands. The short-wave infrared bands are considered to be useful for more remote sensing applications such as modelling surface composition and mineral mapping (Kruse and Perry, 2013).

1.2 Problems Associated with Creating DSMs

The quality of any contributing DSM plays an important role in both the quality of the merged DSM and the extracted building footprint. The fundamental step in DSM generation from stereo imagery is image matching. A list of problems associated with the automation of DSM generation has been produced by Gruen et al. (2000). The nine items on this list are provided below.

Poor or no texture: Area Based Matching (ABM) mainly depends on image texture to support statistical pattern matching. Sometimes part of scene may have poor or no texture. The problem is mainly noticed in large-scale images of dense urban areas, consisting of extensive homogeneous areas, such as car-parks, flat warehouse roofs and sports fields.

This problem is also compounded, for example, with some modern buildings, which have the same wall as roofing material, making the matching process even more difficult. This has led to incorrect results when the roofs edges are excessively unclear. According to research conducted by Zhang and Gruen (2006) sometimes there are what should be conjugate edges but they can only be seen in one image and not in the other; it is difficult to construct the building if the edges are missing or fragmented (Liu et al. 2008). Aytekin et al. (2009) illustrated that building extraction is a challenging issue because buildings may appear similar to roads and pavements. Morgan et al. (2008) and Xiong and Zhang (2010) have mentioned that some inaccurate results appeared due to the lack of texture in specific topographically flat and featureless areas.

Occlusions: Alobeid and Jacobsen (2008) suggest the two main problems faced when producing a DSM are rapid change in elevation and occlusions, and both are serious in dense urban areas. Occlusions arise when the required detail in one image of a stereopair is hidden, such as by a deep roof eave, a high building or another building close by. They are a consequence of surface discontinuity. Occlusions can be considered a source of mismatching for almost all available matching methods. This is supported by a variety of researchers. For example, Tack et al. (2009) mentioned that sudden changes in height cause problems such as occlusions and shadow which lead to matching difficulties. Baltsavias et al. (2006) and Kraus et al. (2009) suggested that the main problems in creating DSMs are caused by matching long and deep shadows and occlusions, particularly in mountainous or snow-covered areas. Finally Aytekin et al. (2009) found that some objects may be difficult to extract due to occlusion by other structures or trees.

Habib et al. (2004) found matching becomes more difficult due to the different directions of view onto the objects, which need to be matched; one outcome is that, due to perspective projection, some edges will not have the same shape as their conjugate edge and another outcome of this is occlusion. The solution proposed by Habib et al. (2004) was to group building extraction into two tasks initially: low level and high level. The low level task was to identify the area for extraction after which the high level task (feature extraction and classification) could be performed. A multi-image matching approach may, also, address the problem of occlusions (as well, at the same time, as addressing other problems such as repetitive structures, multiple solutions and shadows (Zhang and Gruen, 2006)).

Distinct object discontinuities: Object discontinuities occur in large-scale images of built-up areas. As indicated in the previous section, these can cause occlusions. In general the required image segments are fragmented around the discontinuities. For example a small tree can fragment the depiction of a required roof-line in all images. This causes difficulty in getting an accurate result when building roof-lines according to Liu et al. (2008), especially in densely built-up areas (Alobeid and Jacobsen 2008). According to Tack et al. (2009) most difficulties in DSM generation are due to sudden discontinuities produced by buildings and other urban features, excessive variation in heights and the difference between the smooth ground surfaces and emerging buildings,

or other artificial features.

To overcome this problem Zhang and Gruen (2006) have suggested using multi image matching consequent on multiple-views of the object, leading to a decreased discontinuity effect.

A local object patch is not a planar surface: Some matching techniques, such as Cross-correlation and Least Squares Matching (LSM) are dependent on the assumption that it is possible to represent localised objects as inclined planar surfaces. Violations of this assumption cause low values for similarity measures when Area Based Match (ABM) methods are used, and thus result in mismatching.

Repetitive structure: A repetitive structure manifests itself as a periodic texture style in images. An example might be marked parking bays in a car park, or plough furrows in a farmer's field. These usually confuse the matcher because the matcher mixes up a feature in one image with a feature from a neighbouring part of the structure in the other image, especially if the image characteristics produced by these structures are similar compared with the discrepancies of the feature. Zhang and Gruen (2006) mentioned that there would be confusion due to repetition of objects that leads to confusion during searching for an edge.

Shadows: Shadows in images show little texture information and a lower signal-to-noise ratio. Both may result in numerous mismatches. According to Tack et al. (2009) the main source of error in edge detection is building shadows; when shadows show a high contrast with their surroundings pixels, edges will be detected at the shadow border. In addition to that, rooftops may reflect fragmented characteristics produced by shade (Aytekin et al., 2009).

Zhang and Gruen (2006) illustrate that if high resolution images have more than 8-bit pixel values, then this leads to improved image matching by reducing the number of "homogeneous" dark, or light, (shadowed) areas.

Moving objects: Moving objects usually disturb the image matching method.

Multi-layered and transparent objects: This phenomenon leads to occlusion problems and spatial ordering changes in different images. Generally this problem is

very difficult to handle in image matching, but is more related to the objects considered in close range photogrammetry (small glass objects, plastic bottles, etc.) than those addressed in creating topographic databases (Eren et al., 2009).

Radiometric artefacts: Examples of these are affects such as specular reflections and variations in reflected light. As perspective changes, radiometric artefacts are introduced. This problem can be solved by using high resolution satellite images, which have the ability to provide more data for specific items and, thus, provide more opportunity to achieve automatic matching. Such images offer better radiometric performance (i.e. higher dynamic range and signal-to-noise ratio). Using along-track stereo images from the same orbit has led to a reduction in radiometric differences and consequently better chances for achieving correlations (Zhang and Gruen 2006).

1.3 Scope of the Research

This research focuses on merging DSMs and 3D city modelling. The aim of merging the DSMs is to get qualitatively better DSMs that consequently can be applied city-wide, generally considered an appropriate application of satellite imagery (Partovi et al., 2014). So, this research is intended to supply, at city scale, three-dimensional products, in a less labour intensive and quicker manner than using aerial imagery. The products are limited to LoD0 and LoD1, as referred to in section 1.7 ('Objectives of the Research'). LoD1 (and LoD0, to an extent) can be used for applications such as planning and disaster management (Gröger and Plümer, 2012). The reason for the limitation to these two levels, and no higher, such as LoD2, is due to the satellite image resolution being about 50cm. Objects in this type of imagery have limited detail, therefore the products, such as DSMs and the building edges on the orthoimagery show fewer details than aerial imagery (Güdücü, 2008). Due to the relatively low resolution of satellite imagery compared to aerial imagery, generally, the extracted objects, based on VHR satellite imagery, are limited to buildings which are typically occupied, excluding those such as outhouses, garages, etc. (Dahiya et al., 2012; Kim et al., 2005; Mayunga et al., 2007), which are typically small and unoccupied. As a guideline, the Ordnance Survey of Great Britain regards buildings less than 0.1 ha, equivalent to 10m x 10m (Ordnance Survey, 2015), to be small and these have a low priority in revision; buildings less than 8m² are not mapped. In this research the smallest mapped building was 4.2m x 6.2m as shown in Figure 7.4 (b), implementing the proposed algorithm and based on 0.5m resolution data.

1.4 Further Merging Considerations

In this thesis three aspects of merging have not been taken into consideration, but their consideration has been recommended in the conclusions Chapter 9. These are: the multitemporal effect; the multiresolution effect and co-registration. Generally, prior to implementing merging, the above three factors should be considered (Papasaika-Hanusch, 2012).

The multitemporal effect is the difference or the change that has occurred in the DSMs due to change over time between the data sets. It is very common for the data that are used in merging to be multitemporal since they are from different sensors (i.e. sources), consequently leading to the data (i.e. the features) being affected by changes, especially when the gap between the data capture reaches many years. Therefore this point should be addressed during merging the DSMs (Bruzzone et al., 1999). However, in this research, the aim was focussing on examining the result of the merging only, therefore the multitemporal factor has not been taken into consideration and it is assumed, for the purposes of this research, that no change has occurred. In fact in the small study areas used this assumption can be made following the validations in the area which showed no temporal changes had arisen.

The multitemporal effect has been addressed by researchers, as further discussed in section 2.3.2, and further recommendations have been introduced in section 9.2.1. Although a DSM merging application would in practice be applied over a large area and using data captured at different times, this effect has been ignored by testing and validating only in areas where no changes have occurred during image capturing.

Different DSMs are produced with different grid size elements, causing a multiresolution effect on merging; also, frequently, the DSM data are generated from different sources which do not have the same resolution, therefore the multiresolution effect should be taken into the consideration (Ranchln and Wald, 2000). Nevertheless, in this research it is assumed that the multiresolution effect is not pertinent, and there is no need for resampling or otherwise addressing this issue, because the used data in the merging are generated using the same software, and generated to the same resolution. The effect has been further considered in section 2.3.3.

Finally, the last issue considered is co-registration. The process of correctly aligning

both of the DSMs to each other is called co-registration. This step is important, in merging, as reviewed in 2.3.4. In the co-registration process, the 3D separation between the DSMs is minimized (Akca, 2007). The DSMs that has better quality is specified as the “master”, this DSM is specified to be the base, and the other DSM specified as a search DSM and is called as “slave” DSM (Akca, 2007). The co-registration shifts the slave DSM horizontally and vertically and then any resampling is aligned to the master DSM (Li, 2007).

The used data in this research are assumed to be fully able to undergo co-registration, as suggested by the suppliers (United States Geological Survey, 2009). In this research co-registration was not carried out, and the native georegistration values were used. This decision may be questioned, but was taken to simplify the proposed merging process, which may be particularly suitable for situations where evidence may not exist to identify which data set is the most appropriate to act as “master”. As mentioned in the above paragraph, the co-registration process consists of resampling one of the DSMs to be aligned to the other DSM that is called the reference or “master” DSM. The latter is selected based on the quality. For example, in the event that the quality was unknown, selecting the wrong DSMs as a reference could add more error to the DSMs. The issue or co-registration is revisited in section 2.3.4.

1.5 Aim of the Research

The aim of this research is to contribute to the development of the efficient production of an improved DSM by merging existing surface models, each produced from different high resolution satellite image sensors, for the purpose of (automatically) generating reliable building footprints and 3D modelling, particularly investigating using Bayesian approaches.

1.6 Objectives of the Research

There are seven specific objectives, namely, to:

1. evaluate existing approaches to DSM merging and building extraction, essentially through a critical literature review;
2. generate DSMs using a technique, from among the techniques that are listed in Table 2-1, that can produce high resolution DSMs from high resolution images produced from different sensors, and is efficient;
3. develop an optimal procedure to **merge** the high resolution DSMs derived from

high resolution satellite imagery, particularly examining methods based on **Bayesian theory** (subsequently referred to as Bayesian merging), resulting in equal or better quality than the original DSMs;

4. validate the resultant DSMs arising from the proposed merging model using Bayesian approaches;
5. upgrade procedures to extract building footprints from high resolution satellite imagery;
6. apply the developed building extraction approach to the merged DSM to facilitate automated 3D model generation at LoD1; and,
7. validate the performance and quality of the developed approach with respect to building footprint extraction and 3D modelling at a selected test site.

1.7 3D Building Modelling

The application of the 3D building modelling has increased rapidly, either through the survey achieved by national government (e.g. municipality incorporation) or by commercial organization. In addition to the geometrical and graphical issues, it can be used for the semantic study for the objects. For instance, mapping and noise simulation propagation, planning for telecommunications, disaster management such as flood study and investigation (Gröger and Plümer, 2012). According to the literature five different levels have been defined, based on the amount of the details as shown in Figure 1. In this research the devoted levels is focused only on LoD0 which represents the building footprint which is represented by planimetric coordinates without height, and LoD1 which is represented by taking the building footprint “LoD0” and assigning maximum height within this area which leads to produce building prismatic shape.

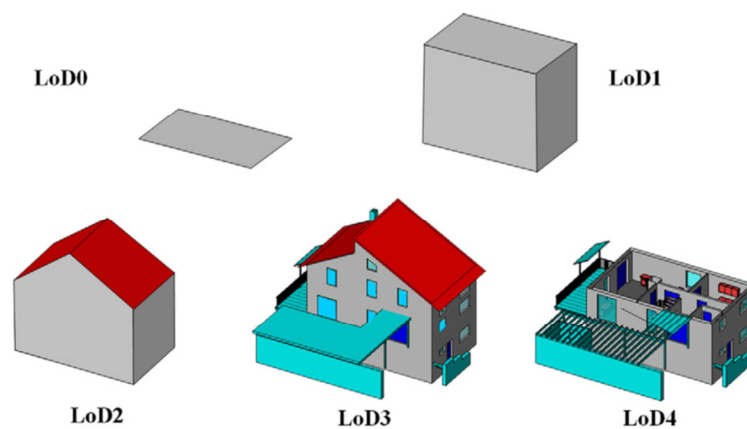


Figure 1.1 Definition given by CityGML for different types of level of detail (LoD) for 3D model (Gröger and Plümer, 2012).

The other levels such as LoD2 are difficult to construct because they need more accurate

and precise data such as that derived from LiDAR or even high resolution aerial imagery. However, the higher models LoD3 are composed of more intensive buildings measurements while LoD4 is based on interior building measurements.

1.8 Implemented Software

As well as using some scripts written by the author (see appendices A,B, C and D)) many software packages have been implemented in this research, either for processing the images for DSM generation and building footprint construction, or for the merging stage. In this section the main software that has been used will be discussed.

SOCET-GXP 4.1 has been used for processing the satellite images for the DSM and orthoimage generation. It is considered to be very powerful commercial software supporting the processing of acquired sensor images to generate geo spatial data, since it has robust techniques for DSM generation (Electronics & Integrated Solutions, 2008).

C++ has been used as programming language in order to develop the required code for merging DSMs and for building footprint extraction. Although this programming language is referred to as a naïve (or low level) language, because the programmer has to write many of the functions, rather than have them supplied from libraries, in order for them to be embedded in scripts, it has proved to be robust in implementing code rapidly and providing the result quickly.

ImageJ software: The author is most grateful that this software could be downloaded, free, from the web <http://imagej.nih.gov/ij>. This software is open source software written in Java, designed for image processing purposes and has been used at several different stages in the work, as well as for visualising outcomes. It is specified to be user friendly software since implementing an author's algorithm does not need much experience. Since the product of C++ (above) was a text file it was necessary to see the graphic result by using ImageJ. In addition, ImageJ has been used to enhance the orthoimage and to apply both convolution and mean-shift algorithms. Also, ImageJ software provides an edge detection tool based on the Canny edge algorithm, in order to find edges in the orthoimagery. The software was developed, in 1987, by the National Institutes of Health, for medical image processing, and is still undergoing continuous development (Schneider et al., 2012).

ArcGIS package: this is the final software package implemented in the research. It has been used as a complete system for the results' visualisation and for the analysis, for validation purposes. In addition it has been used for any necessary coordinate transformation of the products that have resulted from satellite images, and it has been used to create the final report of the results.

1.9 Thesis Structure

The dissertation is structured hierarchically, in **chapter 2** the background and the methods that are used to generate DSMs from satellite images and that are used in merging DSMs are briefly introduced. Also there is a focus on applying Bayesian approaches to image fusion and building footprint extraction. Pertinent photogrammetric principles and techniques for DSM generation are illustrated in **chapter 3**. The data that has been used and the techniques of DSM generation will be clarified. Also there will be some consideration of the principles of acquiring satellite imagery. The theory behind the implemented probabilistic methods used in merging the DSMs, including both Bayesian and classical methods, are discussed in **chapter 4**. Also the methods are compared. **Chapter 5** will show the methodology used in merging DSMs using probability, in two different ways (Bayesian and Maximum Likelihood approaches) in order to minimize the uncertainty of the merged DSMs. Also the *a priori* data, that is compulsory in the Bayesian approach, has been stated. **Chapter 6** is devoted to an explanation of the results and the validation of the original and merged DSMs. The data used in the experiment and the reference data used for the validation are introduced in this chapter. The building footprint extraction and 3D modelling are presented in **chapter 7**, the methodology for building footprint detection and the extraction which was later used for 3D model construction are shown in detail. For the developed process, all the techniques that have been implemented (successful or otherwise) have been illustrated. In addition to that it has been shown that 3D buildings have been created by introducing heights from the DSM. The results and validation of the extracted buildings and 3D models have been presented in **Chapter 8**. The conclusion, to which **Chapter 9** is devoted, summarises the proposed technology and how it can be used for further improvement of currently implemented methodologies, either with respect to merging the DSMs or for building footprint extraction. Furthermore, it presents the expected benefits from this research and highlights gaps where further research is needed.

Chapter 2 Research Background

The processing method that is proposed for this study comprises three main stages, DSM generation, DSM merging and building footprint extraction. The novelty of this research arises from merging the DSMs using Bayesian theory and regularizing building boundaries for 3D city modelling by also applying a Bayesian approach. The literature has been reviewed in a manner directed by these three stages. First a brief literature review focuses on the methods followed for DSM generation. This is followed by a critical review of the literature on the merging of DSMs (and DTMs) from different sources and an explanation of a probabilistic method implementing a Bayesian approach to merging different data. The last stage focuses on the literature related to building extraction from satellite imagery (including gaps) while also launching the approach that is subsequently implemented for extracting buildings using, simultaneously, different sources of data as input.

2.1 Overview of Satellite Images

Since 1972 satellite images have been available for general use by the public. This was started with the launch of Landsat-1, previously called ERTS-1, into orbit (Dowman et al., 2012). Since then different sensors have appeared; of these SPOT-1 HRV, launched in 1986, is the most important of the optical sensors, from the survey and mapping perspective, as it provided stereo-coverage and was designed to support topographic mapping and 3D map construction (Dowman et al., 2012). Later in 1999 IKONOS was launched. Since then different imaging satellites providing images with better resolutions have continued to be launched into orbit. Recently, in August 2014, a state-of-the-art satellite named WorldView-3 was launched into orbit, which can provide images with a resolution of up to 0.31meters. All these improvements have led to satellite imagery being widely available and used by many, for example for planning purposes, mapping and 3D city modelling. Moreover, the increased availability of images has encouraged competition among the providers to make the images available to the user at lower cost.

2.2 DSM Generation Algorithms

Different investigations have been carried out to examine the automatic production of accurate DSMs using images, and these have revealed many difficulties in producing reliable algorithms that recognize (especially complex) buildings formed from the geometric components of planes, curves, etc., and which deal with some of the problems

mentioned by Zhang and Gruen (2006), referred to in Section 1.2. The most crucial step in producing a DSM is image matching which, if achieved to a high standard, leads to a high quality DSM. Even the most robust approaches to image matching still face difficulties in achieving correct matching (Bertin et al., 2015; DeVenecia et al., 2007). Image matching when using satellite imagery to produce a DSM has been an important focus of research, due to the increased application of such imagery in different fields such as flood studies, change detection and planning. Table 2-1 summarises different methods, presented in the literature, which have been implemented for DSM generation; these are compared by indicating the published validation outcome for each approach.

Table 2-1 shows different algorithms which can be used to produce DSMs; from this table it can be inferred that all the techniques fail to construct small buildings and that the accuracies of the methods vary. The RMSE value for the DSM produced using BAE-System's NGATE approach is 0.98m using IKONOS 1m resolution imagery while the accuracy is 0.35 m using WorldView-1 0.5m resolution imagery, as explained in section 5.2, and this is the best compared to all other methods. The NGATE method is based on measuring the coordinates of each pixel, using area based matching and edge based matching, with each of them supporting the other. That is the matched pixels produced from area based matching will be used to support matching edge pixels, and additionally the results of edge matching will be used to support pixel matching, as will be described in more detail in section 3.6.1. Due to the high reported accuracy of the NGATE method, it was decided to use it in this research for producing the initial DSMs used in merging and also later as a primary component in building footprint extraction and 3D modelling.

Table 2-1 Comparison of Different DSM generation techniques.				
Algorithms	Incorporated Techniques	Validation: Best (i.e. smallest) cited height quality statistic	Advantages	Disadvantages
1-Least Squares Matching (LSM) (Alobeid, 2011)	Area based matching	RMSE: 0.6m (using satellite image of 0.5m resolution)	Getting High accuracy of building height	Building shapes are blurred. Searching in 2D, no need for epipolar images Require extra tie points. Small buildings are merged into one group
2-Dynamic Programming Matching Median Filtering (Alobeid, 2011)	. Measuring pixel disparity along the epipolar line with defined threshold	RMSE: 0.5m (using satellite image of 0.5m resolution)	Obtain sharp edge of the building outlines Getting High accuracy building height. Occlusion areas are identified	The produced disparity map contained striping. Small buildings are merged into one group
3-Semi-Global Matching (Alobeid, 2011)	Epipolar geometry. Implementing mutual information which depends on entropy and joint entropy.	RMSE: 0.4m (using satellite image of 0.5m resolution)	Obtain sharp edge of the building outlines Occlusion areas are identified. Getting High accuracy of building height	Small buildings are merged into one group
4-Semi-automatic DSM with triplets (Tack et al., 2009)	Epipolar geometry Feature matching Least squares method (MPGC) to detect mismatches	RMSE: 2.61 m stereoscopic 2.47 m tri-scopic, using satellite image of 1m resolution, tested against Checkpoints	Using more than two images Occlusions are treated	Low accuracy Small buildings are merged into one group
5-Dense stereo methods based on dynamic line warping and semiglobal matching. (Kraus et al., 2009)	Epipolar geometry, implementing the correlation between the lines and Mutual information	STD: 7.6m (using IKONS imagery of 2.5m resolution) tested against LiDAR of 1m resolution	Using more than two images Occlusion are treated	Very low accuracy Small buildings are merged into one group
6-Multiple Primitive Multi Image Matching (MPM) and Multiphoto Geometrically Constrained Matching (MPGC) (Zhang and Gruen 2006)	Area based matching Feature based matching least square method (MPGC) to detect mismatches	RMSE:2.83-3.34 m tested on IKONOS satellite image-GSD=1m, tested against airborne laser scanning	Using more than two images Occlusion are treated	Low accuracy Small buildings are merged into one group
7-NGATE (NEXT Generation Automatic Terrain Generation) (DeVenecia et al., 2007)	Epipolar geometry Area based matching Edge base matching	RMSE:0.98m using IKONOS satellite image-GSD=1m tested against Checkpoints.	Getting High accuracy building height. More robust because the result of area matching support edge matching and vice versa Different strategies can be used in different areas, consequently leads better results. More than two images can be used in matching process, so the occlusion can be detected.	Small buildings are merged into group The building facades are not vertical.

2.3 Data Merging

Merging data has become an important issue in different fields. It has been applied in the image fusion field to increase image detail by integrating the details from different sensors, to produce a single scene that contains more detail than any of the original separate images. Also it has helped to reduce the number of scenes stored for a specific area saving memory space. In the Survey and Mapping field, due to an increase in the number of techniques for data acquisition, which consequently has led to an increase in the sources for DSMs, there has been a motivation to combine these data sets into a single data set that thus contains more detail and of better quality – a productive synergy of the DSMs. The following sections will focus on the background to merging DSMs (and DTMs), which in turn will be followed by an explanation of the application of the Bayesian approach in merging different types of data.

2.3.1 DSM Merging

As indicated in the previous section, data merging has becoming increasingly important in different fields due to increasing amounts of data. Generally, when there were few DSM sources, little research was done into their merging. However, now this has changed and DSM merging has become an important topic as the range of sources of DSMs has increased and the need for better quality DSMs has emerged.

Data merging is complex if optimum results are to be obtained from the available data (Smith and Goldberg 1987). It can be used potentially to identify the highest quality data for the area, as well as to address problems of data volume. Data merging is also important as it fills the gaps and voids produced during constructing the original DEMs. The obvious solution for merging DEMs is to average tiles or strips from DEMs of the same area (Reuter et al., 2011), but this will not reflect the original data quality, because it gives the same weight to all data.

Dowman (2004) stressed the importance of data merging for improving the quality of the DSM prior to feature extraction. Moreover, he extolled the advantages of combining different DSMs such as: eliminating error and dealing with outliers; treating atmospheric corrections; and dealing with a lack of control points.

Reuter et al. (2011) conducted research to align DSM tiles horizontally and vertically,

and then, after that, merge them to produce a seamless DSM. They tried to solve the problems that appear in existing DSMs that have led, for instance, to the bias found in the 30m resolution ASTER GDEM v1 and striping in the 90m resolution SRTM datasets. These problems appear due to the horizontal and vertical misregistration of the data used and these researchers focussed their attention on this. The method Reuter et al. (2011) used comprises three stages. Firstly, horizontal alignment is achieved, by selecting a reference surface. Then the correlation between the reference surface and the DSM for different horizontal offsets is determined, prior to aligning the DSM to the location where the maximum correlation is found. Secondly, repeating the process but performing vertical alignment, using a vertical offset, which is calculated by sampling the rectified dataset with respect to the elevation of the reference surface. Following this process Reuter et al. (2011) achieved a mean displacement between the two data sets of less than a centimetre, whereas prior to their adjustment it was of the order -300m. Thirdly, they applied the LOESS filter (also known as local regression) to merge and mosaic each tile (or strip) into a single final DSM. The LOESS filter attributes each final individual pixel with the number of pixels employed in its estimation process and also provides an error estimation, using the minimum, average and maximum elevations for each individual pixel.

Wegmüller et al. (2010), focussed on the value of DSM merging to fill gaps, and developed a method for merging the ERS-2 ENVISAT Tandem (EET) CInSAR DSM with another existing DSM, such as the SRTM DSM. The methodology was used to fill in the gaps existing in the CInSAR DSM, due to deficiencies of EET CInSAR in mountainous areas or those of considerable slope, thus rendering the DSM more accurate and efficient. The methodology was based on using a weighting function, with the original CInSAR DSM used in the areas away from the gaps, whereas a smoothed weighted transition is applied at and near the gaps. The purpose of this weighted merging is to prevent elevation ‘steps’; this method was applied successfully, on an area in California, USA, which comprises flat and hilly areas; no ‘steps’ were detected near the flat areas in the merged DSM, but in the steeper areas some interactive editing was required to avoid ‘steps’.

Karkee et al. (2006) also carried out research into merging DSMs produced from InSAR techniques, i.e. the SRTM DSM, and an optical system, i.e. the ‘relative’ ASTER DSM.

The purpose of this research was to increase DSM accuracy, particularly filling DSM voids. The flowline of the algorithm initially involved registering the ASTER DSM to the SRTM DSM. The slope and aspect values from the surrounding pixels' elevations were used to fill the gaps by interpolation, as well as data from an additional DSM. During the merging, the incorrect elevations in both DSMs were removed by applying a Fast Fourier transform in the frequency domain (Karkee et al., 2006). The errors in the optical DSM showed high levels of variation, while errors in the SRTM DSM were limited to low levels of variation. The result has been evaluated by using a reference DTM derived from a contour map with scale 1:25000; the topographical (i.e. contour) map used had been generated by the Survey Department of Nepal with teamwork by the Finish International Development Agency. The results showed that the approach successfully increased the accuracy (i.e. reduced the RMSE) and filled the voids; it increased the accuracy by 44%. (The term 'relative' ASTER DSM has been used to refer to a DSM obtained from using image matching techniques without using GCPs (Tokunaga et al., 1996), and when using GCPs it has been called an 'absolute' ASTER DSM). Both data sets were registered using the same ground control points, resulting in RMSE values of several pixels in each set. The authors concluded that it would be more efficient to use image based correlation co-registration, with the expectation of less interactivity and greater sub-pixel accuracy.

Hosford et al. (2003) showed an approach for enhancing DSMs through a merging operation based on a geostatistical approach (i.e. capable of estimating an σ value). They used DSMs from stereo-radargrammetric SAR satellite and airborne laser altimeter data. The merged DSM has been evaluated against a DTM produced from a 1:25,000 scale topographic map and a helicopter-borne laser altimeter. The validation showed that the: i) estimated standard deviation; and, ii) RMSE with reference to the map of the fused DSM decreased from 21.2m to 14.0m and 27.2m to 23.2, respectively. For the helicopter-borne data error statistics decreased from 25.3m to 11.0m and 44.0 to 36.5 for the estimated σ and RMSE, respectively.

A study carried out by Papasaika et al. (2009), improved the details and accuracy of DSMs arising from poor image matching, by merging using extra data sources. The extra data was a DSM acquired from IKONOS and airborne LiDAR. The approach used addressed the merging based on each DSM's quality generating a DSM with blunders

and errors eliminated. Initially the DSMs are aligned to each other by using least squares 3D matching for the co-registration. Then, error of the DSMs was estimated through their geo-morphological characteristics such as slope and aspect without using measured ground truth. These characteristics later guided a fusion process based on an active contour model (Kass and Witkin, 1988) also called a 'snake'. An active contour tries to find the lines or edges in the image. A spline is defined in the image and the operation tries to shift the location of the spline onto the line or edge features. This operation involves two constraints - internal and external. The internal constraint is specified by the user to control the smoothness of the defined 'snake', while the external constraint is obtained from pixel intensity values (also referred to as 'energy'). The active contour method is trying to minimize the discrepancies between the spline and the pixels of the initial edges and lines. The results showed that the approach was able to merge DSMs successfully and also dealt with blunders, successfully removing artefacts in the LiDAR data.

Papasaiki and colleagues (Papasaika and Kokiopoulou, 2011; Papasaika-Hanusch, 2012) also conducted further research into mitigating the errors that were generated in DSMs during their construction, through developing a robust model based on sparse representation. The sparse representation model is capable of dealing with more than two DSMs, using "dictionaries". The components of such a dictionary, entitled atoms, are local terrain forms, constructed from training data. The DSM so produced from merging optical imagery and InSAR has been evaluated against a LiDAR DSM. Promising results were obtained and the sparse representation method gave good results compared to the weighted average method, in less steep areas. For the slopes that are greater than 30% the weighted average was better than sparse method while the result was the opposite when slope was less than 30%.

Costantini et al. (2006) conducted research into merging using different data sources (SRTM SAR-X, ERS SAR tandem data) for the purpose of image mosaicking. The focus was to manipulate discontinuities at the edge of the overlap area, arising from either horizontal or vertical systematic error. In their approach, first, they addressed the problem of systematic horizontal error, through co-registration. Later they addressed vertical systematic error. After the systematic error had been removed, the next stage was combining the DSMs, by weighted averaging based on the DSM's accuracy. The

subsequent interpolation filled any existing DSM voids. The validation result was RMSE values of 8.7m when compared to a higher accuracy source (having a 1.3m RMSE) suggesting the approach is capable of reducing the error inherent in the sources (for example the ERS SAR tandem data RMSE was 13.8m).

A study conducted by Schultz et al. (1999) generated DSMs from aerial imagery and fused them for mosaicking purposes. At the fusing stage, self-consistency was implemented to detect elevations that were unreliable in order to remove them. They claimed that their merging approach improved the merged DSMs by taking advantage of redundant elevations, removing outliers and handling geospatial uncertainty. A downside was the significant computational overload.

Lee et al. (2005) merged two different DSMs, from different sources, different dates and different resolutions (e.g. airborne and InSAR derived DSMs). They followed a neural network approach to merging in order to improve quality. The data that were used in this merging were specified to be multi-sensor, multi-resolution and multi-temporal. The outcome was successful to the extent that gaps were filled. However, the accuracy of the resulting DSM arising from fusing the two SAR derived DSMs using weights based on height error maps, derived from the interferometric coherence of the SAR data, could be considered low.

Podobnikar (2007) conducted research into merging DSMs from different sources for the purpose of mapping. The aim of the merging was to obtain higher quality DSMs through removing potential gross errors without using auxiliary data. The fusion was based on weighted averaging. Although the aim of the research was to address the enhancement of geomorphic characteristics leading to superior perception and thus more effective mapping, considerable consideration was given to the quality of the outcome, and he concentrated very much on the visualisation of the resulting DSMs, in order that blunders could be detected.

Fuss (2013) explained how DSM merging could reduce errors such as systematic errors and outliers, in addition to the offset that exists due to land use. In her thesis, she generated different DSMs from RADARSAT-2 imagery, a stereo-radargrammetric method being implemented for this purpose. In the fusion, elevation estimation based on

slope and elevation thresholding was later followed by k-means clustering. Finally, the merged DSM was filtered and smoothed. Although the accuracy increased globally, the precision decreased, and land cover was found to be particularly influential. The study indicated, strongly, that different fusion methods should be applied in areas of different land cover.

Tupin (2004) conducted research into merging SAR and optical imagery for the purpose of 3D building representation using a Region Adjacency Graph implementing a Markov Random Field by segmenting the optical images. This was based on using segmentation over the optical imagery for obtaining regions and then assigning the heights from the SAR data. The consequence of applying this method showed successful results with respect to large buildings, but failed with small or low height buildings.

A study was carried out by Reinartz et al. (2005) in order to generate and fuse DSMs from different sources (i.e. Spot-5 optical satellite imagery, SRTM C-band and X-band radar data) and also to model forest trees. To the extent that this study both looked at fusion of DSMs and extracting objects (trees) its aims were not unlike those of the building based study reported in this dissertation. In the merging process, they determined a height error from each DSM, based on the production process, which was used in order to provide a local estimation of the DSM's accuracy. The result has been evaluated using more accurate data sourced through laser scanning and aerial photography. The result showed that accuracy improved by merging these DSMs. Turning to the trees the difference between the surface model (DSM) and the bare earth model gives the tree heights. The SRTM derived DSMs consistently gave more accurate tree heights than SPOT-5 derived DSMs.

From the literature, DTMs generally and DSMs in particular have become an important issue in different applications especially in 3D modelling, therefore getting an accurate and detailed DSM has become a necessary and important issue. To obtain the highest quality DSM is still a challenge due to deficiencies in the applied algorithms and the resolution of satellite images (and not discounting the presence of random error); however, one approach to obtaining a good DSM is by merging different sources. The studies reviewed in this section do indicate the advantages of merging DSMs that can arise as follows: filling holes; improving the qualitative attributes; and, raising accuracy

by integrating the details from different DSMs and consequently removing potential gross errors (or blunders). But these accounts of research into DSM fusion do also indicate high computational overheads; some remaining interactive editing requirements; problems of registration even with a good supply of GCPs; slope and land cover having a differential affect on the quality of output; a desire to move away from an overall accuracy figure (e.g. for weighting) to pixel specific figures but an acknowledgement that these can be difficult to achieve; and, finally, that small objects can get lost in a DSM.

2.3.2 Multitemporal Effect

Usually the data used in merging are from different epochs, which means there is a high possibility of changes within the data sets' area. Thus, merging data, using an effective approach, from different sources at different times is an important challenge that faces researchers, due to the increasing number of different data sources such as optical satellite imagery, DSMs, LiDAR and InSAR available (Zhang, 2010a).

Different researchers have followed different procedures to address this problem. For instance Papasaika et al. (2008) introduced the following procedure: prior to fusion using weighted averaging, they produced a residuals map for each DSM, using a DSM of identical extent and resolution as shown in Figure 2.1. Then if the heights at each grid point were contradictory, they assigned the most recent DSM's height to the outcome.

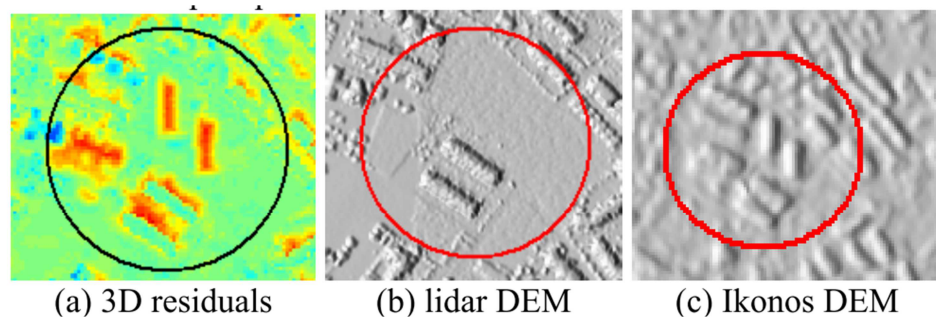


Figure 2.1 3D residuals map showing the effect of multitemporal data on the different DSMs (Papasaika et al., 2008).

Ghannam et al. (2014) introduced an approach for merging multitemporal data. Their approach was called the Wavelet-based Spatiotemporal Adaptive Data Fusion Model (WSAD-FM), based on wavelet transformation to merge MODIS low-resolution imagery with Landsat scenes, implementing a linear model. The aim of this approach is to increase the spatioresolution of satellite imagery captured at different periods of time.

The fusion (or merging) of multitemporal data has also been utilized to enhance image classification. Bruzzone et al. (1999) utilized data merging for the compound classification of remote-sensing data that was specified to be multisource and multitemporal. The merging approach was achieved implementing a Bayes approach.

Data merging, in the context of this research, aims to put heights together from different DSMs in order to get a better DSM, however it is very rare to have DSMs that are produced from data captured at exactly the same time; either the time gap is small, or large, perhaps reaching several years. Thus it is important to address the changes that have occurred during the time gap between different data sets.

2.3.3 Multiresolution Data

Since the merging operation is using data from different sensors, the data that are used in merging do not necessarily have the same resolution. For that reason, it is important to know how to deal with the effect of multi-resolution data.

According to the literature, it is preferred to resample the data to the higher resolution; in that case the feature on the ground will not be lost. This action is followed by Damron (1999) to merge 10m resolution DSMS from IFSAR DEM (Interferometric Synthetic Aperture Radar) which has been up-sampled to be the same as the LiDAR resolution, 2m. Ranchln and Wald (2000) also followed the same principle to up-sample the lower resolution data in order to merge SPOT multi-spectral data that has resolution 20m with higher resolution imagery that has a resolution of 10m. Kumar et al. (2009) also followed the same principle for merging panchromatic and multispectral images to obtain pansharpened image. They resampled low resolution multispectral data with 4m resolution into higher resolution data at 1m resolution.

The other sampling method, which is leads to reducing the resolution, is called downsampling. This type of resampling is not preferred and the disadvantage of using it as referred by Frajka and Zeger (2004) is causing to loss information in the data consequently is lead to reduce the quality of the data.

2.3.4 Data Co-registration

Prior to merging the data they should be georeferenced to the same reference plane, and co-registered (Costantini et al., 1997). Karkee et al. (2008) emphasize the importance of removing the shift that may exist between data sets prior to the merging. For the purpose of improving the accuracy of the DSMs by removing the voids in the data, Karkee et al. merged two public DSMs: SRTM and ASTER DEM. The ASTER DEM has been co-registered to the SRTM. In the co-registration process the ASTER data has been shifted to the mean elevation of SRTM as shown in Figure 2.2. The reference data for assessing the original and merged datasets has been obtained from contour maps at scale 1:25000.

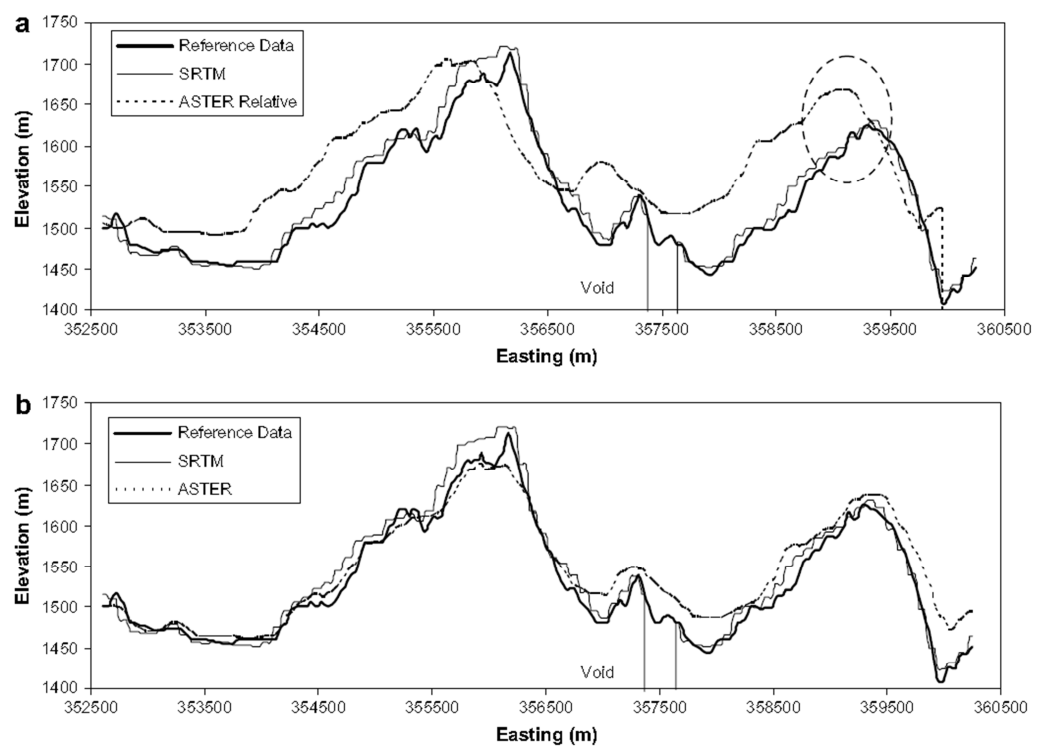


Figure 2.2 Effect of co-registration of SRTM and ASTER DEM profiles, respect to the reference data (a) the ASTER DEM and SRTM profiles before the co-registration (b) the final profiles situation after co-registration operation.

For co-registration, Papasaika-Hanusch (2012) implemented the LS3D algorithm of Gruen and Akca (2005). The co-registration was applied prior to the merging the DSMs. It aimed to remove the shifts, both horizontally or vertically and less probably the rotation and scale variances, between the DSMs. In the LS3D algorithm, the sum of the squares of the Euclidean distances between the DSMs points have minimized.

2.3.5 Probabilistic Method in Merging including maximising Entropy

One merging method to be investigated in this study will be based on a probabilistic method applying **Bayesian theory**. This aids decision by making use of new evidence, but taking into account previously existing evidence.

Within the surveying and mapping disciplines, merging or fusion has long been applied in the production of pansharpened images (Fasbender et al., 2008) and in the reduction of satellite image scene dimensionality through Principal Components Analysis (Kumar and Dass, 2009). More recently, as the literature cited in later paragraphs will show, Bayesian approaches are used, to some advantage, in both these areas. As a regular DSM is very similar to a single band from a multispectral image, except that every grid value represents elevation rather than reflectance, so, perhaps, Bayesian approaches could be applicable to DSM merging?

A considerable body of literature exists which shows how researchers have attempted to use Bayesian approaches to improve the results of data merging or fusion. Not all of these researchers have been working in the field of surveying and mapping, or Earth Observation. For example a dissertation by Punska (1999) shows merging data incorporating a Bayesian approach and Markov Chain Monte Carlo methods applied to different sources of data from different sensors, and particularly addressing problems arising from discontinuities, using an algorithm originally developed for speech recognition.

Rather than image bands or DSMs, Christensen (2005), looked at a small simulated data set, to compare the Fisherian, Neyman-Pearson (NP) and Bayesian approaches to hypothesis testing. Christensen (2005) showed that Bayesian approaches are influenced by *a priori* information, and the user can exploit different *a priori* information to reach a decision with respect to hypotheses that are mutually exclusive. The Fisher and Neyman-Pearson approaches to testing hypotheses are based on starting with a null hypothesis. In the Fisherian approach the distribution is selected and the data are evaluated as to whether the data suits the proposed model, or not. In the Neyman-Pearson approach two hypotheses are tested, a null and an alternative hypothesis, with the test based on rejecting the null hypothesis. Hypothesis testing is not a large component of the research being reported on in this dissertation, but, as well as considering the value of the

null-hypothesis, which for many is counter-intuitive, Christensen (2005) emphasises that the Bayesian approach to assigning confidence to an outcome is very intuitive, as it directly links confidence and probability. Christensen (2005) further, stresses the importance of good *a priori* information to achieve an appropriate result when using Bayesian approaches.

In a real, non-simulated, application, Jalobeanu et al. (2008) conducted research into merging multi-source data from astronomical images for the purpose of mosaicking, co-registration, improved resolution and, above all, to reduce the large number of redundant astronomic images, through fusion. Geometry, blur and noise statistics were taken into account, uncertainty was also addressed; and, then, a probabilistic method based on Bayesian inference to achieve the fusion, was tested, along with two other frequentist, purely image based solutions methods. The RMSE value after Bayesian fusion was half, or less, that of the other tested fusion methods. However, despite the higher accuracy, some noise in the final fused image was probably due to less than optimal *a priori* values.

Sharma et al. (2001) implemented a probabilistic model, based on the Bayesian approach, to fuse images from a variety of different noisy sensors, for aircraft landing guidance, rather than Earth Observation. For the parameter estimation least squares factor analysis was used; the merging implemented a locally weighted linear operation. For weighting, characteristics such as signal, noise and polarity change (necessary in the case that images have reverse contrast) were taken into consideration. As a result, the fused image, had improved features and contrast, and also the noise was reduced, however high computational overheads prevented its early practical implementation.

Mohammad-Djafari (2003) also published research details on using a Bayesian approach for merging data and images. He demonstrated different applications for applying Bayesian approaches such as merging images which are registered, and registering and merging images using 2D images to construct 3D views which have been linked with a DSM. He showed how to fuse different types of images such as ultrasound echo-graph data or X-ray radiograph data. He also implemented a Bayesian approach later with Féron (Féron and Mohammad-Djafari, 2005) for the purpose of automatic image fusion followed by segmentation, again not for Earth observation, but to fuse and interpret

X-ray security images; different images from different sensors or different bands were exploited in this research. Féron and Mohammad-Djafari (2005) obtained the final segmentation result following the merging of these different images. Results were satisfactory, with regard to the Bayesian approach, but the authors admitted to not having addressed the issue of automatic image co-registration in their otherwise automatic image fusion procedure. Turning to the segmentation aspect of this 2003 study, at more-or-less the same time Jones et al. (2003) fused images from high resolution visual and low-resolution thermal data incorporating a Bayesian approach, for surveillance purposes; the fused images giving better results during segmentation than any of the un-fused images.

A literature search has been conducted which has found at least thirteen applications of Bayesian approaches in the surveying and mapping field, related to the fusion of Earth Observation imagery. These will be considered in subsequent paragraphs.

The outcome of the supervised classification of a Bayesian fused image and the supervised multispectral classification of the contributing images reported by Shi and Manduchi (2003) showed disappointing results. It was shown that merging the output from various classifiers, using a Bayesian approach, gave a rather high misclassification result, when it was applied in a situation when more than one image characteristic is available (e.g. colour and texture). The misclassification rate for the Bayesian fused image was 38% while for the multispectral classification it was 27%. The advantage of working with only data set could be set against the poorer classification outcome, but Shi and Manduchi (2003) predicted improved results following further development.

Fasbender et al. (2008) implemented a Bayesian framework for fusing panchromatic and multispectral images prior to subsequent land cover classification. In this fusion process, Fasbender et al. (2008) depended on the statistical interaction between different bands, whether multispectral or panchromatic. The benefit of this method is that there is no requirement for hypothesis modelling, and it exploits visual or quantitative criteria to give weights to multispectral or panchromatic data. Consequently, this provides a result based on the study area and the user's requirements. The suggested algorithm proved to be successful; at 250,000 check pixels the correlation, with the classification under a Bayesian framework for four land-cover classes, was better than 0.9. In addition to the

multispectral images, it is also possible to fuse images from SAR and hyperspectral sources, using this algorithm. Fasbender et al. (2008) did not also include *a priori* values for the land cover classes, but recommended that they could in the future, despite the already very good results for the method tested.

Although unlikely to help in addressing this research's objectives, because the large number of bands involved is not an issue in the work reported in this thesis, a Bayesian approach implemented for segmenting hyperspectral images has been considered. Mohammadpour et al. (2004) carried out research in order to achieve segmentation by exploiting information from the heterogeneous bands that construct hyperspectral images. Their proposed algorithm was based on applying a Bayesian approach to the problem of joint segmentation of hyperspectral images. They tested their algorithm on a simulated image and found it to reduce by 80% the number of misclassified pixels.

Murthy et al. (2005) implemented a Monte Carlo Technique incorporated into a Bayesian framework for the purposes of image restoration while some parts of the images under consideration are left unchanged. They worked with simulated images, and much of their effort focussed on the issues of establishing both the *a priori* and likelihood values needed for the Bayes approach. Clear guidelines did not emerge from this article, as work was continuing, but the problem, particularly, of getting good *a priori* values was clearly stated. Kumar and Dass (2009), also concerned with getting a good *a priori* value to merge images that were obtained from different sources, implemented the total variation (TV) approach to give a value they used as a prior in pixelwise merging. They used the TV approach together with Principle Component Analysis to obtain an optimum image from source images. The benefit is that smoothing the image and while maintaining the image's discontinuities consequently led to sharp edges. This is essentially a Bayesian approach with a focus on smooth surfaces and sharp edges – again a value when considering the urban environment.

A Bayesian approach was used by Diebel et al., (2006) in 3D surface reconstruction and also subsequent decimation, from point cloud data or surface meshes of small real world objects, usually obtained from stereophotogrammetric procedures. Although the application area was computer animation, it was thought that the focus on real world objects might have some bearing on this research's intention to extract building models

from DSMs. The Diebel et al., (2006) approach was successfully applied on noisy data that had been combined with an *a priori* probabilistic model resulting in improved features through smoothing and establishing good corners. Results were not good where highly curved smooth surfaces were being re-constructed, but although the urban landscape has many smooth surfaces, these tend to be planar with pronounced corners. A useful point made by these authors was that a future research could focus on the noise systematically generated by each sensor.

Confirmation of the value of the Bayes approach to merging was provided by Ge et al. (2007) who used a probabilistic linear estimation method based on a Bayesian approach in order to merge multispectral Landsat ETM+ and panchromatic images for enhancing the multispectral image. According to Ge et al. (2007), the proposed method gives better results than the traditional methods (i.e. principle Component Analysis (PCA) and wavelets).

Providing further background, Bayesian approaches have also been used by Zhang (2011), and Zhang et al.(2008) to fuse hyperspectral and multispectral images; by Kotwal and Chaudhuri(2013) when fusing for improved visualisation; and, by Mascarenhas et al. (1992) who used a Bayesian framework to fuse SPOT satellite imagery after selecting the best characteristics from multispectral and panchromatic bands to obtain an ideal synthesised image. Finally, Sommer et al., (2009) evaluated the uncertainty in a data set after data fusion claiming that uncertainty can be robustly evaluated through using a Bayesian approach.

From the foregoing, it can be noted that Bayesian approaches have been able to solve the problems of merging **images** by including a proper prior. Identifying the proper prior is a challenge.

In the work reported on in this dissertation, it will be seen that the *a priori* probability that has been used is based on, and benefits from, the morphological characteristics of the ground and buildings particularly roughness, as suggested by Papasaika-Hanusch (2012). The index which is tested, in this research, to identify the proper prior is based on a roughness index using the entropy concept.

Several decades ago a clear relationship between maximum entropy and prior probability

was established (Jaynes, 1968) with the recommendation that despite the apparent subjectivity of *a priori* values, computing the maximum entropy could provide such a value, objectively.

The entropy concept was developed by the German physicist Rudolf Clausius ((Clausius, 1850) cited via (Downarowicz, 2011)) and means “turns into”. The word is used to express the ineffective heat that is released when heat is being produced from an energy system (Downarowicz, 2011). Shannon (1948) used entropy in probability and information theory, and later Downarowicz (2011) referred to him as the “Father of information theory”. Downarowicz (2011) advocated the use of the term entropy in other fields of science, not only physics or mathematics. For example, it is now being used in sociology. Generally the expression of entropy, can represent many conditions as indicated by (Downarowicz, 2011) such as:

“disorder”; “chaos”; or, “tendency toward uniform distribution of kinds”.

In information theory, concepts such as uncertainty, information and choice are influenced by entropy. Shannon (1948) mentioned that entropy, H_f , can be quantified, for a group of independent probabilities (n), using equation 2-1:

$$H_f = \sum_{i=1}^n f_i \log \left[\frac{1}{f_i} \right] = - \sum_{i=1}^n f_i \log f_i \quad 2-1$$

where:

H_f : local entropy value; and,

f_i : probability density for (n) values.

Cover and Thomas (1991 cited in Gill (1994)) maintain that, in information theory, entropy expresses the distribution of a specific random variable, based on the average information quantity. The value of entropy is limited to between 0 and $\log(n)$ and the measurement unit is referred to as either the ‘bit’ or the ‘Shannon’ (Downarowicz, 2011). When the state is dense, the value will be 0 or very close to 0, and when the state is diffuse, the value will be close to $\log(n)$, shows the entropy for pairs of probability values: p, and, (1-p).

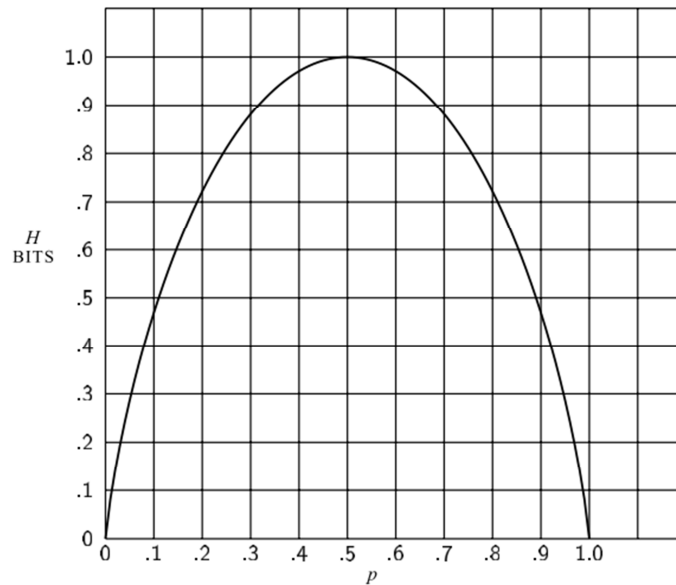


Figure 2.3 Graph shows the value of the entropy for pairs of probabilities (Shannon, 1948).

According to the graph, it can be noted that the value of entropy, represented by H , is a vital requirement regarding the choice of information. H achieves the minimum, or $H=0$, when there are certain choices or information about the situation, and one of the probabilities is zero, or while the other is unity. However $H=1$ when the values of both probabilities are equal, which in the case of information theory represents the situation when the information is not certain, and there is equality with regard to certain and uncertain probabilities. The value of entropy will be equal to $\log(n)$ when all probabilities are equal (i.e. having the same value). It should be noted that in the case of trying to increase the similarity or trying to minimize the difference between the probabilities, the value of the H will be increased, as implied by Shannon (1948).

As mentioned, entropy is linked to disorder. Gill (1994) illustrates that uncertainty in the probabilities can be evaluated using entropy, and, according to Shannon the entropy value H reaches a maximum when the values of the probabilities (p_1, p_2, \dots, p_n) are uniformly distributed, and its value increases monotonically with the value of n .

From the considerations of the reported work of others, provided in this section, it can be seen that Bayesian approaches have been widely and successfully applied in fusing greyscale or hyperspectral images for better visualization, segmentation or restoration. Some success has been achieved in smoothing planar surfaces and sharpening edges at discontinuities. However, the degree of success varies considerably amongst reported projects. Bayesian approaches can be used to minimize uncertainty in the data with a

proper model construction. Computational overheads are frequently reported as high, but these will tend to reduce over time.

Unfortunately, there have been no specific examples found of Bayesian approaches to merging DSMs, although the similarity between a DSM and a multispectral band is clear. The work of Diebel et al. (2006) with point clouds (or dense meshes) of small real world objects, and using Bayesian approaches to improve the modelling, is closely related. Thus, even if using Bayesian approaches for merging **DSMs, an objective of this research**, is considered novel it is likely that many of the findings with regard to merging images will be relevant.

2.4 Building Extraction Literature

Various research projects have been developed for detecting buildings, either automatically or semi-automatically, instead of using manual methods. Different sources of data are utilized for this purpose such as: aerial images; satellite images; LiDAR data; unmanned aerial vehicle (UAV) or unmanned aerial system (UAS) data; or, DSMs.

Extracting building footprints from satellite images has been a challenge for a long time, due to the low resolution of satellite images compared to aerial imagery. But, many researchers have focused on using satellite imagery for extracting building footprints possibly since its cost is very competitive compared to LiDAR and aerial imagery (Suárez et al., 2005), which are already commonly used for the purpose.

The following sections illustrate building footprint extraction for Survey and Mapping purposes. The sections are organised according to the type of input data implemented in the processing.

2.4.1 Building Extraction from Satellite Imagery

Kim et al. (2005) conducted research into extracting buildings using a ‘voting’ process rather than a ‘grouping’ strategy. A grouping strategy requires detecting all boundaries of the building in order to be able to extract the building. However, this method will fail when applied to high resolution images such as IKONOS; often, the long side of the building will be detected as disconnected pieces of lines and the short side of the building would not be detected. To overcome this difficulty Kim et al. (2005) introduced their

voting strategy algorithm. The algorithm concentrates on a building's 'lines' rather than the whole building. Kim et al. (2005) investigated images with 1m resolution of buildings of rectangular shape and relatively large size. The algorithm works by taking a point on the roof of the building to be considered as the starting point. After that, a small area is defined centred on the starting point. A line extraction algorithm by Burns et al. (1986) is used to extract lines near the starting point, after that their location and orientation is 'voted' based on an analysis of the orientation of neighbouring extracted lines. The line extraction algorithm was able to extract only the lines in the long side of the building Figure 2.4. The Building Extraction rate was 83%.

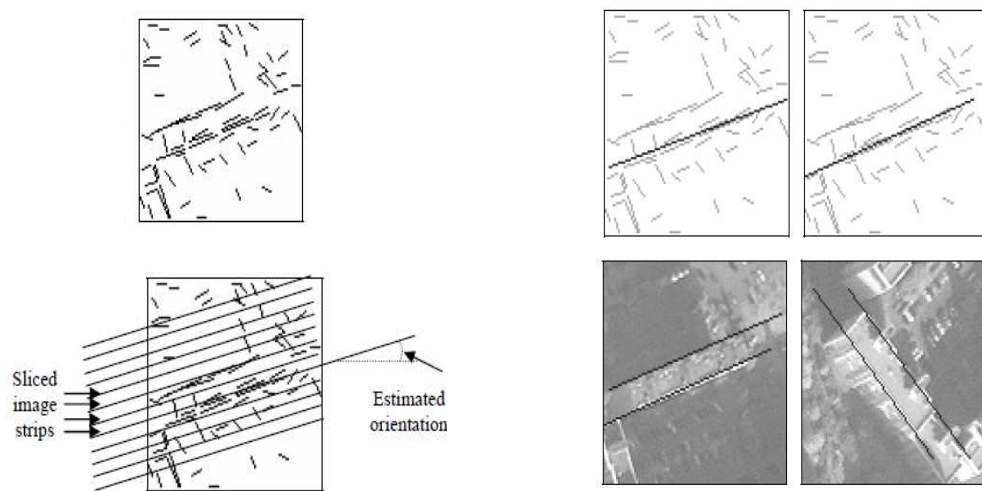


Figure 2.4 The building extraction process from satellite imagery by the voting algorithm (Kim et al., 2005). Top row: the fragmented detected edge using the Burns et al. (1986) method. Bottom row: the constructed edges of the long side of the building.

Research conducted by Theng (2006) to extract buildings from high resolution satellite imagery used the Circular Casting Algorithm. The implemented algorithm was developed as an alternative to the Radial Casting algorithm (Mayunga et al., 2005, 2007). Circular Casting has the ability to overcome the problems of complex buildings, by initializing an active contour. The initialization point can be at any point of the building, and it is not compulsory to select this point inside the building. Hence this leads to automation and the point can be picked by comparing pixels using corner detectors (Harris and Stephens 1988). Corner point detectors assume that corners are related to local maxima at each pixel of the image. If the local maximum is higher than a certain threshold, the pixel is declared as a corner. As a result, Circular Casting can start from any of the first corner pixels found. Figure 2.5 illustrates the steps for applying Circular

Casting to a building. As shown, the casting will start gradually from first detected point and will stop when the extracted building is closed.

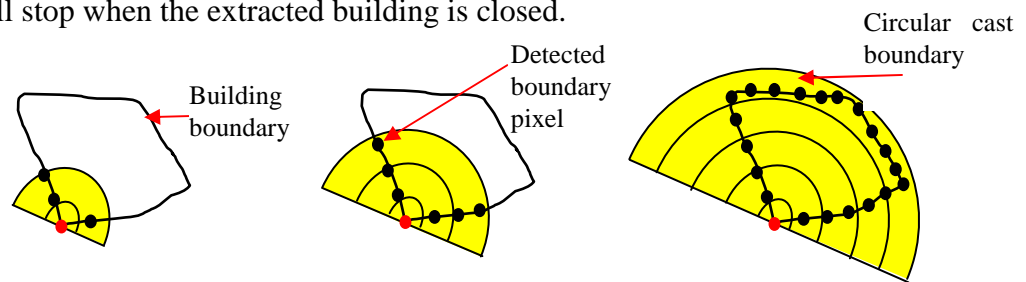


Figure 2.5 Process of the Circular Cast Algorithm as it draws contour for extracting a building from satellite imagery (Theng, 2006).

Liu et al. (2008) conducting research into building generation semi-automatically by selecting a building location manually, developed a novel method, based on feature extraction and region segmentation for extracting rooftops. They assumed that a building would be seen in high resolution satellite images as an individual building object. They also assumed that orthogonal corners of the buildings are connected with each other by straight outer lines. Generally, their hypothesis was based on the fact that a building has two parallel lines, which are connected to each other at orthogonal corners, and which can be classified as the dominant line set. Accordingly, the buildings can be depicted if the corners and lines are correctly selected, see Figure 2.6(a) and 2.3(b). For the extraction to be achieved, they merged the two ideas: first, effective segmentation of the building region has been used to extract components of a building roof's outline from its background; second, using pose clustering to adjust the direction of roof outline components and building corner locations. Pose clustering is based on a voting process in which the majority position of components in an edge are used to determine the position and location of that edge (Olson, 1997).

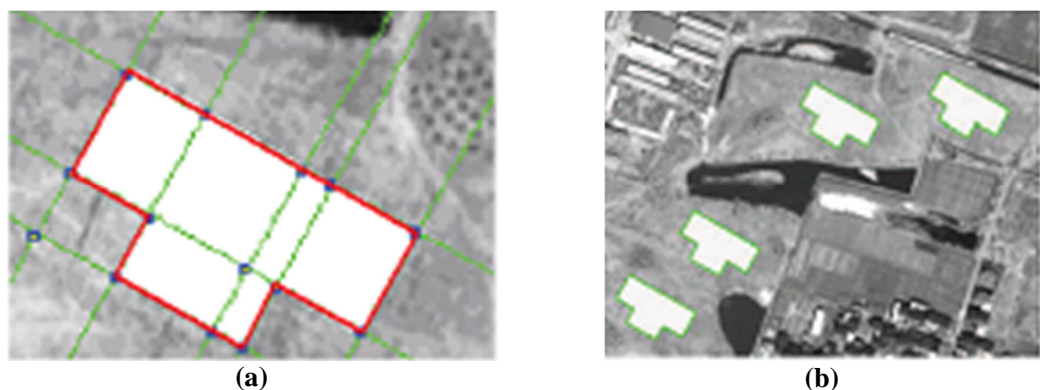


Figure 2.6 Result of extracted buildings from high resolution satellite images (Liu et al. 2008).

In order to achieve buildings' roof boundary extraction precisely, Liu et al. (2008) used

seeded region growth segmentation and multi scale object oriented segmentation methods, applied to Quickbird imagery. Based on image segmentation, by combining clustering and model matching techniques, they proposed a new method for building locating. This method finds building objects by considering spectral, geometric and textural characteristics. In order to attach the extracted building to a suitable shape, they created a template library with a collection of different types of buildings. Generally they depended on locating or extracting, or both, the attributes of the building using two different methods: first, region based building extraction or RBBE (combining seeded region growth where the user needs to calculate the limit of a building and manually define the threshold used in the growing process); and, second, localized multi-scale object oriented segmentation. The latter is efficient when applied to small and simple buildings, where the edge contrast is low, using high resolution images based on edge confidence and mean shift. Localized multi-scale object oriented segmentation also uses seeded region growth to gather pixels inside regions and approximate the shape of the building.

After developing RBBE, Liu et al. (2008), commenced feature based building extraction or FBBE. The output of this stage can be used in vector displays. Typically FBBE consists of feature extraction, which includes edge detection, corner detection, line detection and orthogonal corner detection. Several different algorithms are used; for example to locate the building, they used the ‘Voting Algorithm’ – discussed earlier. After locating the building Liu et al. (2008) employed the Hough Transform to extract the dominant line sets. This was followed by edge buffer analysis which was used to filter out (some) false edges. The building can be accurately constructed by using search algorithms for building corners based on the intersection of line sets. The result of the proposed method was successful when applied to rectangular shaped buildings’ roof scans, achieving a 75% building extraction rate (BE).

A study was conducted by Shaker et al., (2011) for 3D building extraction from IKONOS stereo satellite imagery after creating a DSM and orthoimagery. For the building extraction, the multispectral data underwent supervised classification. A Maximum Likelihood classifier was applied to classify the buildings based on the supplied training data, which was provided as four classes (bare soil, buildings, roads and vegetation). Finally, the classes were reclassified into two classes, buildings and other objects, based

on a majority filter. The planimetric buildings' data was obtained from detected edges from the images' building class; the imagery subsequently also stored the elevation from the DSM. The validation of the result has shown a good RMSE value for the height validation, which is 1.33m, and based on the number of buildings extracted correctly and incorrectly, the generated indices such as the Building Extraction (BE) rate reached 82.6% and the missing factor (MF) was 16%. BE, expressed as a percentage, is obtained from $TP/(TP+FP)$ and likewise MF from FN/TP : where TP (true positive) is a building that was a building and is marked as a building, in other words, there is no error in the extracted buildings; and, FN (false negative) records the buildings that exist but have not been extracted.

An approach to building footprint extraction developed by San and Turker (2010) is based on applying the Hough transform on a satellite image to automatically extract buildings. The approach aimed to extract circular and rectangular buildings from IKONOS satellite images Figure 2.7. The Support Vector Machine (SVM) (San and Turker, 2010) algorithm is based on building detection and delineation from the building patches defined in the nDSM. For the first stage, i.e. building detection, orthoimagery is used to detect the buildings based on the NDVI index. The nDSM and the NDVI index of the SVM algorithm implemented in ENVI software is used to classify the buildings. The SVM algorithm includes statistical learning for classifying the complex and noisy image data and supervised classification. Eventually manmade objects are converted into vector map data after distinguishing and then eliminating non-buildings areas. For that purpose the edges are detected, first, using Canny edge detection, and then converted into a vector map, based on the Hough transform that is specified to detect lines and curves. These lines and curves are eventually grouped together to construct building boundaries. The results in the study area have been classified into three types and shapes of buildings. The quantitative validation was based on BDP (Buildings Detected Percentage) which is obtained by $TP/(TP+FP+FN)\%$; and CM (Completeness) which is obtained by $TP/(TP+FN)\%$; where TP (true positive) is buildings that are originally buildings and marked as buildings, in other words, there is no error in the extracted buildings; FP (false positive) are the buildings that have incorrectly been extracted; FN (false negative) the buildings that exist but have not been extracted. The results were as follows: for the industrial rectangular shaped buildings the CM was 93.5% and the BDP was 79.5%; for

the residential rectangular buildings CM was 95.3% with a BDP of 79.1; and, for the residential circular buildings it was the lower CM was 78.7% with a BDP of 66.8%.

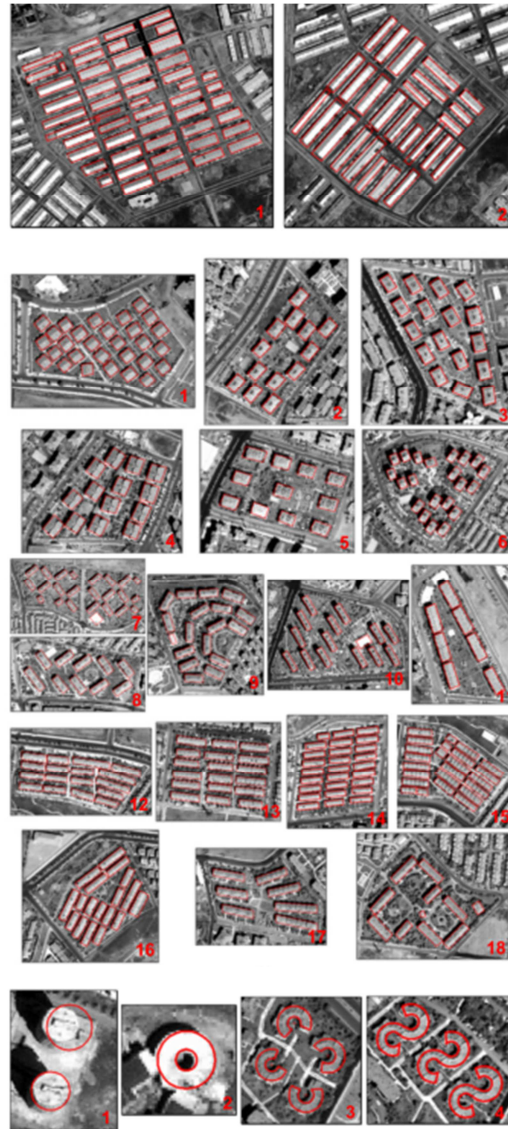


Figure 2.7 Different sites showing extracted buildings, as reported by San and Turker (2010) from satellite imagery using the Hough transform, in which 80% of buildings were detected.

Aytekin et al. (2009) used satellite images for the purpose of automatic building footprint extraction. The CM reached 80.8% and the percentage of BDP reached 84.5%. Their approach is based on classifying the satellite images into manmade and natural objects. For the natural object classification, which consisted of vegetation and shadow, they implemented the NDVI index and the shadows were detected based on converting the colour image's RGB space into YIQ (luminance (Y), hue (I), and saturation (Q)) colour space; shadow is specified by its high Q/I ratio. Finally the manmade structures were detected using the Otsu threshold, that is finding a threshold value in the grey intensity histogram after minimizing the intraclass variance, and maximizing the interclass

variance (Deserno, 2011). After eliminating the natural objects, the mean shift algorithm was applied to segment the rest of the objects, which consisted of man-made objects. The resulting objects suffered in that some parts were not related to buildings and consequently needed to be removed through a mathematical morphology operation. Other objects related to roads were removed based on using a shape ratio system, which is based on the road geometrical shape being different from building shape, in that roads have a longer shape and smaller width. Moreover, principle component analysis (PCA) was also implemented to remove the small irrelevant shapes or artefacts. Aytekin et al. (2009) implemented PCA in order to distinguish real artefacts from the road segments that are considered to be thin and long and not successfully detected earlier as road segments, based on their shape ratio.

Dahiya et al., (2013) extracted buildings automatically as vector shapes by using IKONOS high resolution satellite imagery. Their method was based on segmenting the image using a 'split and merge' segmentation method. Before converting the image into a vector map, numerous filters were applied which led to the removal of objects that were not specified to be buildings. The filters that were applied at this stage were probability, size and clump size. Eventually the vector map contained candidate buildings, but to further improve the accuracy, another group of operations were applied which comprised a probability and island filter, a smoothing operation, and enforcement of orthogonality. These operations were implemented in the ERDAS environment. The result showed that among the 122 known buildings, 24 buildings were not identified which means the extraction rate is 80%, but there was a further problem in that 18 building were identified incorrectly as buildings, see Figure 2.8.

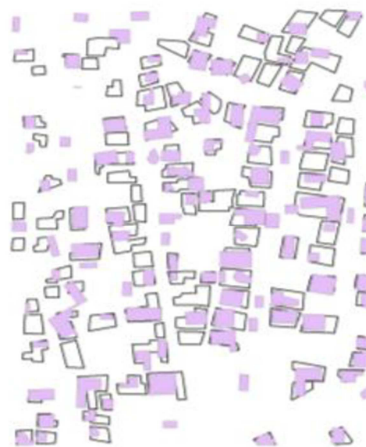


Figure 2.8 Building footprint extraction result using satellite imagery proposed by Dahiya et al. (2013), extracted buildings (purple) obtained from the applied algorithm compared with manually extracted buildings (black line).

Another study was also carried out by Dahiya and colleagues (Dahiya et al., 2012) for extracting buildings automatically from high resolution satellite imagery using an object oriented approach. First, they classified the satellite imagery using a pixel-based approach. The training data was used to compute the probability of the pixel being a building. Several samples of training data were carefully used for the tree study area, and not including any background. Later the produced image was segmented using the split and merge segmentation method, which led to an image with all pixels with similar characteristics being connected. After that a group of filters was applied (e.g. morphological operator, probability filter, size filter, reclump, dilate, erode, and clump size filter) in order to clean up the image from all unnecessary pixels and noise, remove small objects and smooth remaining objects. The resulting raster map was then converted to a vector map, which was filtered to achieve a cleanup. The result of the method suffered some errors that required further filtering; also, the method did not work on any image that had similarities between buildings and other objects. The quantitative validation indicated a good result. The approach was applied in three different areas that contained different numbers of buildings: 66, 94 and 102 with the overall accuracy BDF, based on the manually extracted buildings, for each group being 85.4%, 73.8% and 70.6% respectively.

Another study for building footprint extraction from satellite images was conducted by Krauss et al. (2007). IKONOS stereo satellite images were used for that. Prior to the production of the DSM and the DTM, the images were pre-processed. Later on, Krauss et al. (2007) classified the images using the nDSM that is obtained from subtracting the DTM from DSM. Then the remaining objects were divided into manmade and vegetation objects by using the NDVI index, since the imagery was multispectral. From the classified image, it was possible to model height objects as buildings or trees, see Figure 2.9 (where low height objects have been treated as the ground surface). All buildings in the small test area were extracted.

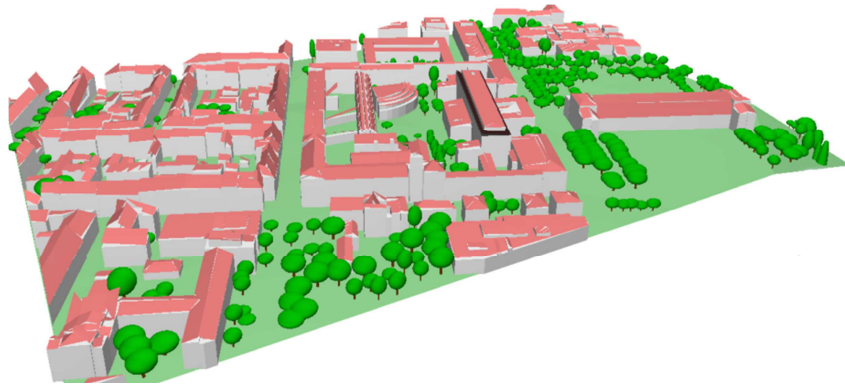


Figure 2.9 Constructed 3D site using satellite imagery as shown in the research by Krauss et al. (2007).

In addition to automatic building footprint extraction, some research was also conducted using satellite images for building footprint extraction but with some interactive input to the algorithm. Mayunga et al., (2007) developed a semi-automated algorithm for building extraction from the satellite imagery. The algorithm has been applied to an area that was specified to be a rapidly growing area in Dar es Salam city, Tanzania. The buildings and their distribution were irregular. The algorithm that was used was the ‘snake and radial casting’ algorithm. The suggested method depended on an initial point being located on a building then the boundary would be detected automatically. The results were validated using regional truth data, which was measured manually. Since the initial points were located manually for the building the efficiency of the building detection was high and it reached 90.5%, the measured RMSE accuracy reached 0.68m (well-defined features). The author advocated the method since it depended only on orthoimagery, did not need any auxiliary technology to be implemented, such as stereo viewing, and unskilled operators could implement it.

A study was conducted by Sohn and Dowman (2001) to exploit Ikonos high resolution satellite images and extract buildings that have rectilinear edges. It was implemented by picking a seed point inside the building, which is iteratively expanded until it reaches the edge of the building. The expansion is based on neighbour pixel values within a threshold limit. To infer an initial building shape quickly, a local Fourier analysis was implemented; this was assisted by analysing the principle direction angle in the building cluster. For the threshold limit, this expanded at each iterative stage until meeting the limit of a maximum number of iterations and the constraint of a fixed area as shown in Figure 2.10. The algorithm has the ability to overcome problems appearing during

building extraction using traditional area-based segmentation and consequently has led to successful building extraction. The achieved building extraction rate was 73%.

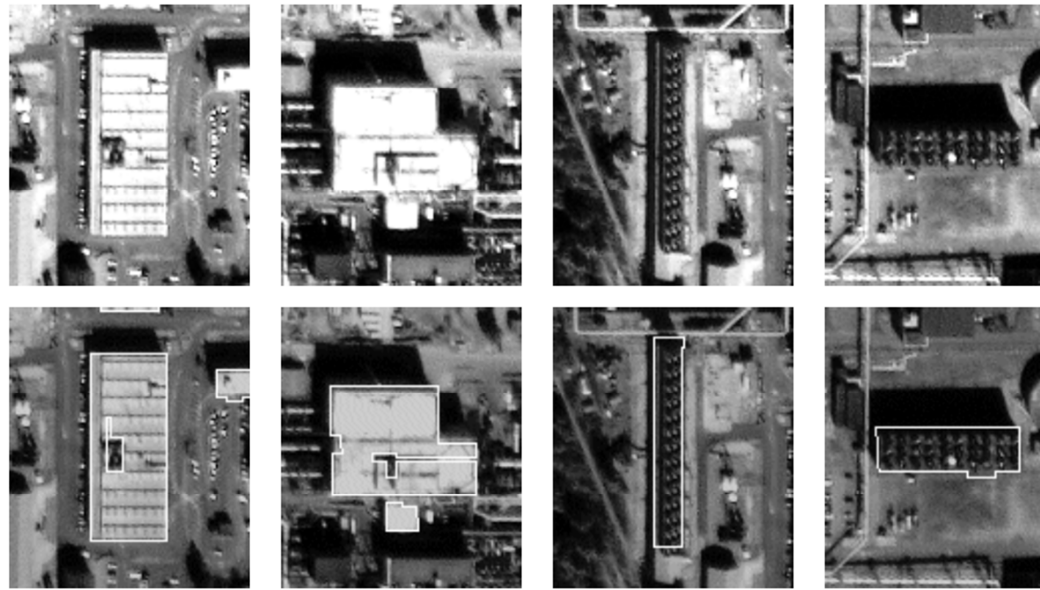


Figure 2.10 Buildings extraction using the BUS (Building Unit Shape) algorithm implementing IKONOS imagery. The top row represents the satellite image, and bottom row represents the extracted building superimposed on the satellite image (Sohn and Dowman, 2001).

Lari and Ebadi (2007) exploited high resolution satellite images in partnership with artificial neural networks for building extraction purposes. Their aim was to exploit selective information from both spectral and structural features. The proposed approach consisted of three stages: image processing and segmentation; feature extraction; and finally, applying an artificial neural network to decide whether the extracted edge belonged to the building or not. For the segmentation, they implemented a seeded region growing algorithm, based on using seed points assumed to be evenly distributed over the image. It is implemented by comparing the seed with its neighbour pixel intensities, if it is within the assumed threshold then it will be considered as the same region. Later on, a closing and opening operation was implemented to enhance the segmented image. The implemented artificial neural network is based on two main phases; the first phase is called the learning phase, uses training data that is saved manually, and is exploited to train for great accuracy; the second phase is called the application phase, where the data will be used to test new datasets. The input data of a neural network is based on extracted features consisting of mean colour and intensity, roundness, area, perimeter, compactness and the structural features of each segment. The algorithm when applied, gave a reasonable result, the detection percentage reached around 81% for urban areas, see: Figure 2.11.



Figure 2.11 Applying artificial neural network for building footprint extraction (a) the data set used in experiment (b) the result of applying the algorithm (Lari and Ebadi, 2007).

Jin and Davis (2005) carried out research in order to extract buildings automatically from high resolution satellite imagery (e.g. IKONOS imagery, 1 meter resolution) using information based on structural, contextual and spectral specifications. Initially a differential morphological profile (DMP) operation was carried out to supply image structural information. Information related to the building's assumed size and position is inferred from the DMP. The DMP is also used to detect the building's shadow, which is consequently used to provide appropriate data about the proposed size and position of the related building. At the final segmentation stage, building rectangles used as a seed are evaluated and expanded. The third stage is extracting the building that have a high intensity value based on spectral information. Eventually, the buildings extracted from the three operations are combined to represent the final result. The result of the implemented algorithm gave a 72.7% CM rate over a complex urban area, with a BDP of 58.8%.

Güdücü (2008) utilized both shadow and multispectral information for detecting buildings from the IKONOS satellite imagery, by exploiting these to detect buildings in dense urban regions. The assumption that was the basis of the algorithm is that building boundaries consist of rectilinear sides. Therefore, first, the building edges were detected using edge detection algorithms. Then, the lines were extracted using the Hough Transform algorithm and the edges that fell in a common straight line were selected as a building boundary. Shadows and HSV (hue, saturation, value) colour was also an aid. The validation has been based on a BDP and a Branch Factor (BF). Branch Factor is obtained from: $BF = FP / (TP + FP) \%$. The algorithm has been applied in nine different areas that contained different types of buildings and the results, based on number of

extracted building and not on calculated area have varied from 82%-100% with respect to the BDP and 0%-8.6% with respect to the Branch Factor (BF).

The research so far outlined in this section has specifically used satellite imagery only as data input, however some researchers have implemented algorithms that were applied on both satellite and aerial imagery. Satellite and aerial images have been integrated for the purpose of building extraction by Saeedi and Zwick (2008) who described an approach to automatically extracting building edges using either aerial or satellite images. Other research has been implemented using the Level Set function based on the active contour model, which detects buildings from mono-view aerial or satellite imagery (Karantzalos et al., 2007), using prior shapes.

The findings of this section are summarised in the Table 2-2, below.

Table 2-2 List of the summarised Building Extraction rate (BE) based on the data source.		
REFERENCE	DATA SOURCE	BE
Kim et al. (2005)	IKONOS	83%
Theng (2006)	QUICKBIRD	100%
Liu et al. (2008) - RBBE	QUICKBIRD	75%
Shaker et al., (2011)	IKONOS	83%
San &Turker (2010)	IKONOS	66 - 79%
Aytekin et al. (2009)	QUICKBIRD	84%
Dahiya et al. (2013)	IKONOS	80%
Dahiya et al. (2012)	IKONOS	71-85%
Krauss et al. (2007).	IKONOS	100%
Mayunga et al., (2007)	QUICKBIRD	91%
Sohn&Dowman (2001)	IKONOS	73%
Lari & Ebadi (2007)	IKONOS	81%
Jin and Davis (2005)	IKONOS	72%
Güdücü (2008)	IKONOS	82-100%
Saeedi & Zwick (2008)	QUICKBIRD	80-100%
Karantzalos et al. (2007)	IKONOS	NA

2.4.2 Building Extraction from Aerial Imagery

Some of the research has used exclusively aerial imagery. For instance Ahmadi et al. (2010) implemented an algorithm which is based on an active contour model for detecting and extracting building boundaries automatically using only aerial imagery (i.e.

no other data such as a DTM). The extraction method was able to overcome deficiencies that existed in classical active contouring for building extraction, which primarily arose because of sensitivity to the point of initialization and image noise. The results were acceptable, the validation has been conducted based on the true buildings in the region. The completeness factor (CM) was about 80%. Among 347 buildings, only 281 could be detected, while the others failed to be extracted due to the low radiometric discrimination between the building roof and the object background. 96% among the 281 building were extracted correctly (i.e. the extracted buildings were considered sufficiently identical to the real building shape).

2.4.3 Building Extraction from UAVs

In the last few years, data acquisition from UAV/UAS (Unmanned Aerial Vehicle or Unmanned Aerial System) has increased tremendously since it has been identified as low cost and high resolution. It has become an important source of data for rural areas but only sometimes (due to insurance restrictions), in urban areas. UAV images can reach a very high resolution, for example 1cm/pixel if flown at low altitude (Turner et al., 2012). Some research has addressed using images acquired from UAVs for building extraction purposes. Jizhou et al. (2004) utilized single UAV scenes for 3D model construction including building texture, and implemented qualitative validation of the algorithm based on the constructed models, Figure 2.12(a). Also other research carried out by Küng et al., (2011) created prismatic building shapes based on the building footprint and a constant height (e.g. LoD1, see section 7.7) from processing UAV stereo images, so it can be used for more detailed building construction incorporating, e.g., Google Sketchup software to create a more detailed building (i.e. LoD2), Figure 2.12(b).



(a)



(b)

Figure 2.12 Building construction implementing UAV images (a) 3D building model based on single UAV image (Jizhou et al., 2004) (b) LoD2 building using Google Sketchup software based on model created from UAV images.

2.4.4 Building Extraction from LiDAR

For more than two decades Aerial LiDAR has been very important to researchers, and has been implemented to extract buildings with very high accuracy, since the point cloud provided by LiDAR is very dense and accurate, and so it offered great potential regarding building footprints (Meng et al., 2007).

Wang et al. (2006) introduced a footprint extraction algorithm based on Bayesian techniques for the automatic construction of a building footprint from pre-classified LiDAR data. The algorithm consists of three steps. The first step finds a point on the approximate boundary. The second step establishes the approximate building footprint based on the found boundary point. The third step uses a Bayesian maximum *a posteriori* (MAP) estimation method, to find the most probable building footprint. A Bayesian method is used to enhance the footprint by integrating the goodness of fit to the data with *a priori* footprint shapes, see Figure 2.13, using the most probable angles for buildings, such as 180° for straight lines, and 90° for corners. A problem remains in that trees are often extracted as buildings in the areas when the buildings overhung with trees.

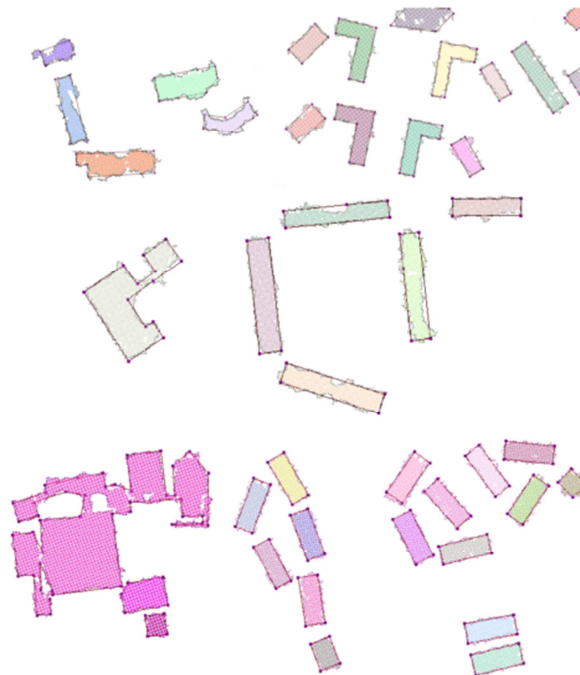


Figure 2.13 Recommended algorithm by Wang et al.(2006) to extract buildings from LiDAR data using the Bayesian approach.

Shen et al. (2011) developed an algorithm called ‘Alpha Shape’ to extract building boundaries from LiDAR Data. Their algorithm had the ability successfully to delineate the ‘inner’ and ‘outer’ boundaries of ‘welled’ buildings from point cloud data with

convex and concave polygon shapes. In addition to that, it was possible to retain the fine features of buildings, adaptively, and to distinguish the footprints of non-buildings. Alpha shapes can be used to delineate a building's shape using a randomly distributed set of data points and is characterised by high processing speeds. Moreover, it is possible to extract different shapes and sizes, selectively. Furthermore, the alpha shapes algorithm can remove noise and some trees by deciding the polygon's size and number of sides, and the number of points used to delineate the boundary. A downside of this work is that small irregular buildings were not successfully extracted.

Haithcoat et al. (2001) conducted research into automatic building footprint extraction and 3D reconstruction from LiDAR data. Their algorithm is based on detecting the objects that are higher than the ground surface, and general knowledge of building geometric characteristics, such as size, height and shape. The building extraction algorithm is based on the notion that buildings have geometric descriptions such as size, height and shape, for example, a height threshold will remove entities with lower height such as cars and grass; in addition, roads and empty land will be identified and removed also. By using size thresholds, it is possible to remove smaller entities such as single trees. Regarding the other objects such as vegetation areas and vegetation mixed with buildings, it is not possible to use height or size criteria to remove them. In this case, differential geometric criteria, based on using differential calculus or shape descriptors through derivatives, are used to distinguish between the objects. This is based on the roughness of measured surfaces, because buildings are polyhedrons consisting of flat planes while trees are irregularly shaped, lacking plane surfaces. Buildings with complex shapes were not always extracted correctly.

In addition to the above research, LiDAR has become an interesting and active research field for different researchers, briefly summarised here. Awrangjeb et al. (2013) extracted 3D building roofs automatically by integrating LiDAR data and multispectral imagery. Elhifnawy et al. (2011) utilized LiDAR data to extract the buildings by implementing a wavelet method. Kabolizade et al. (2012) used an algorithm composed of three models: firstly extracting the initial building boundary; secondly, improving the accuracy of the extracted boundary; and finally, the contour of the initial building was generalized and the building extracted, the data used in this research being limited to LiDAR. Sampath and Shan (2007) developed a set of rules to trace and optimize building

boundaries from LiDAR data. It has been possible to use LiDAR data to make a successful 3D city model for an area, for instance Dorninger and Pfeifer (2008) were able to construct 3D building accurately even where the roof of the building is sloped, which was successfully used in 3D city modelling see Figure 2.14.

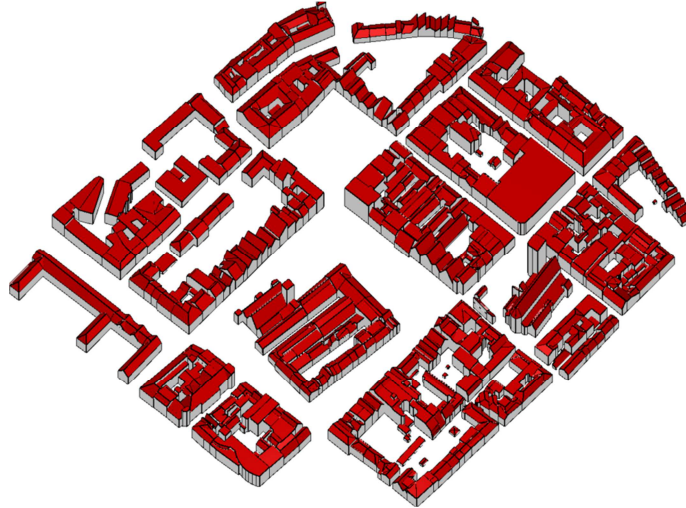


Figure 2.14 Automatically extracted buildings from LiDAR data as proposed by Dorninger and Pfeifer (2008).

2.4.5 Building Extraction from DSMs

Brunn and Weidner (1997) conducted early research into using DSM and input data for identifying and extracting buildings but their developments quickly ran up against the need for significant interactivity and trees and buildings were difficult to distinguish.

More recently, a DSM has also been utilized in an attempt to extract building footprints, Lafarge et al. (2008) used DSMs in order to extract buildings for 3D city modelling, see Figure 2.15. Parameters required are chosen by trial and error; each DSM requiring different ones. It is hoped that in the future, the estimation of these parameters will be a fully-automatic process.

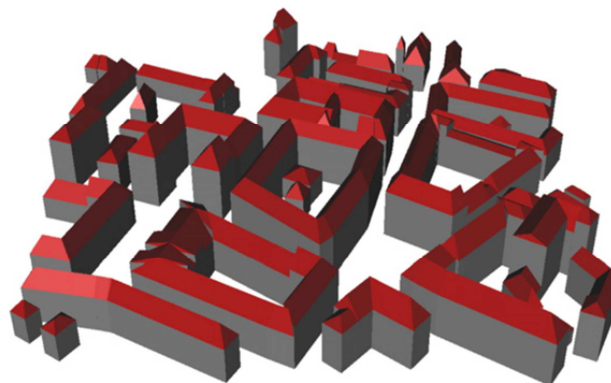


Figure 2.15 Exploiting DSM for 3D city modelling based on object approach (Lafarge et al., 2008).

Other research has been conducted by Tournaire et al. (2010) in order to extract building footprints from a DSM using an ‘energy’ model; the method has utilized ‘energy’ by minimizing it. The total ‘energy’ is the weighted sum of the ‘internal energy’ and the ‘external energy’: the ‘internal energy’ quantifies the objects’ collection and the ‘external energy’ is used to represent the suitability of the object with respect to the DTM. The input data was an urban DSM with resolution around 1m. The objective of the research was to use a group of rectangular building footprints to obtain authentic object configuration. This process has been achieved through using stochastic geometry in the operation. The process has been achieved through designing an energy function that has been divided into two terms. The first term is based on assessing the object’s fit to the available data, while the second term is based on *a priori* information such as overlapping parts or aligned objects in order to penalize or support a particular footprint composition. The Reversible Jump Monte Carlo Markov Chain (RJMCMC) joint with a simulated annealing algorithm has been used in the minimization process to find the optimal object composition. The validation has shown that the CM is 86.1% and BDP is 63.8%, this percentage could be increased by excluding the trees from the data using an NDVI mask in order to reach 83.8% and 73.3% for the CM and BDP respectively.

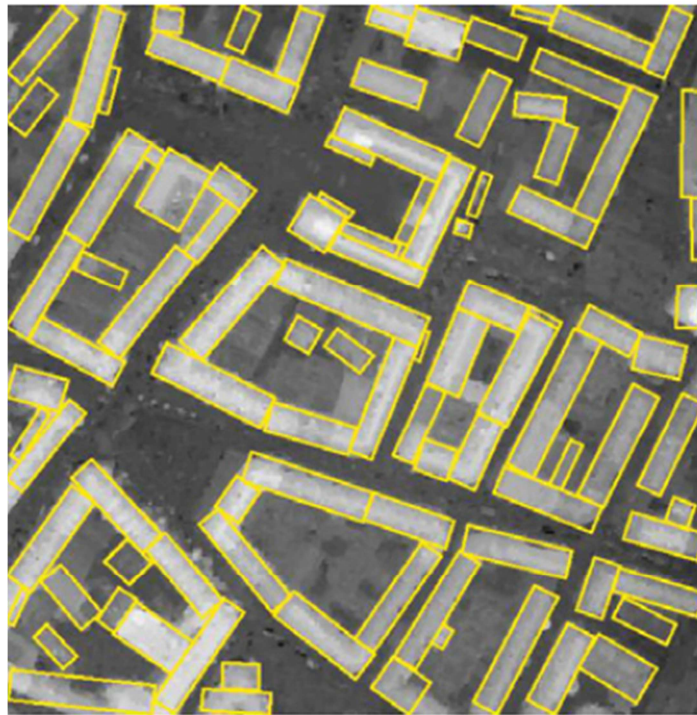


Figure 2.16 Building footprint extraction from satellite DSM using ‘energy’ modelling (Tournaire et al., 2010).

2.4.6 Building Extraction from Satellite and LiDAR

In addition to using satellite images as the only source for building extraction, many researchers used them in synthesis with other sources of data in order to get an optimal result for the building extraction.

LiDAR data with satellite imagery is a good example. Sohn and Dowman (2007) implemented an algorithm for automatically extracting buildings for the purpose of 3D city modelling. Their approach is based on using LiDAR data to identify building location with the support of NDVI index from multispectral bands of IKONOS. The building location that has been identified as the initial location has been validated using the edges obtained from the IKONOS satellite imagery. The building extraction rate for this project was ~94%, but building shapes were considered poor.

2.4.7 Building Extraction from Satellite and Maps

Another study for 3D building modelling, incorporating heights from processing the Ikonos satellite imagery and the building boundary from 2D planimetric maps, has been carried out by Tack et al. (2012). First, the location of the boundary has been identified using the cadastre map that is later matched to the refined DSM for the height extraction. The remaining DSM, after extracting each building boundary from it, has been used for constructing the DTM to represent the bare ground surface, which has been processed further to smooth it and remove the artefacts such as cars, local relief and vegetation. Since the building roofs were hipped (i.e. pitched with four slopes and a roof ridge), one elevation has been assigned to the building that is the maximum height. The results, which are seen in Figure 2.17, show that the algorithm has been successfully used for the modelling the city. Furthermore, the quantitative validation has shown that the accuracy of the method used with respect to the height for zone1 is 3.57m (which included houses and historical buildings covering an area 5.8km^2), and for zone2 is 3.62m (which included industrial buildings covering an area 9.4km^2). It seems the authors were slightly disappointed in these results, but expect that as DSMs get more and more dense, or converge with data clouds, the methodology developed will better support building extraction from DSMs.

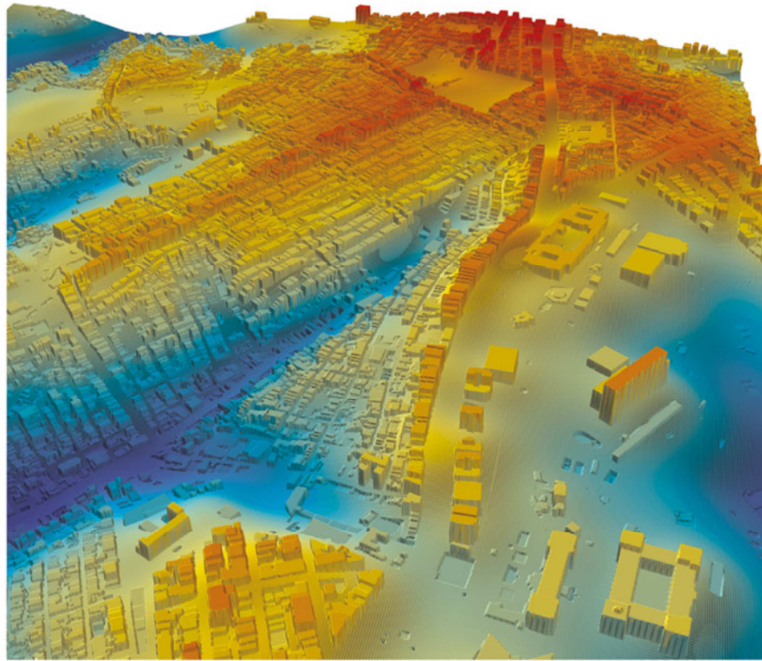


Figure 2.17 Urban 3D modelling using 3D satellite images' data and maps (Tack et al., 2012).

2.4.8 Building Extraction from LiDAR and Aerial Imagery

Some of the research has exploited different sources for data in order to extract buildings. Awrangjeb et al. (2013) utilized Orthorectified satellite imagery based on existing DTMs in addition to LiDAR for creating 3D models automatically, based on detecting the roofs of the building. The approach was based on using the building edges obtained from orthoimagery as a boundary to define the region of the LiDAR points of the roof. For the edge generation using Canny edge detection and entropy edge detection, from orthoimagery, the corners are detected also using the fast corner detector algorithm. The group of the points that is bounded by the extracted edge are expanded iteratively to construct the whole roof surface Figure 2.18. Some of the surfaces that represent the top of the trees have been removed, based on the characteristics of a geometric plane such as the size and the absence of spikes. For the validation, two different methods have been used: object based and pixel based, and for each of these two indices have been implemented, namely, detection rate or completeness (CM) (where $CM = TP / (TP + FN)$) and FP. For the object base the CM and FP were more than 98% while with respect to the pixel based approach the CM was 91% while FP was 95%.

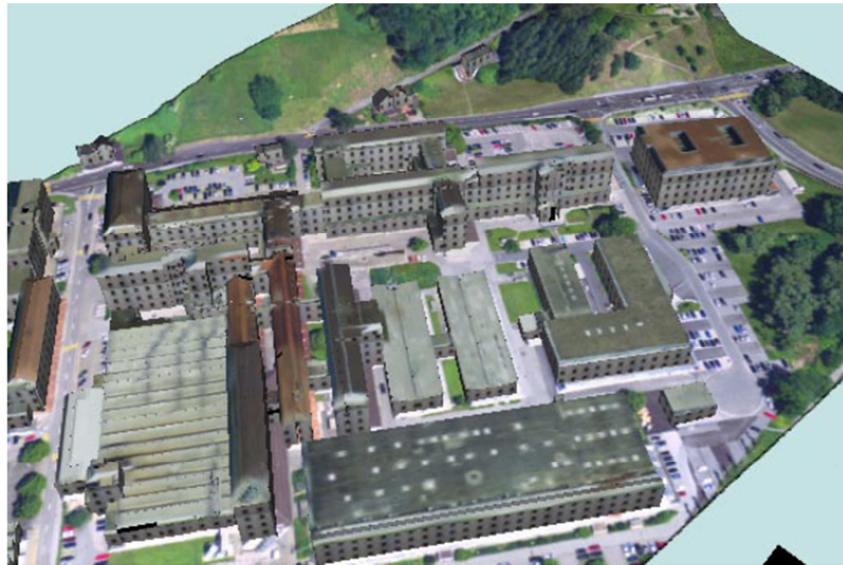


Figure 2.18 Perspective view for the constructed model from LiDAR covered with texture from aerial imagery (Kokkas, 2008).

Many researchers have focused on using LiDAR and aerial imagery since the first gives the best accuracy and the second can be used for mapping texture. These two sources are considered as the most important two sources for city modelling (Kokkas, 2008; Li et al., 2013a; Li et al., 2013c; Mao et al., 2009; Wang et al., 2006).

2.4.9 Building Extraction from DSM and Aerial Imagery

Kim and Nevatia (2004) conducted a study to find and describe complex buildings (with respect to their shape and roofs) by employing multiple, overlapping images of the site. They used the DSM as optional auxiliary data to find the location of the building in order to reduce the time for its complete matching; it was possible to locate the building without using the DSM, but it took longer. Therefore the DSM used need not be accurate (i.e. it is not necessary for the detailed shape of the building with its edges to be identified), a rough DSM, showing at least the presence of buildings, is acceptable. In addition they used numerous photos for the area in the extraction process. The operations started by extracting line features from the images, then the lines were associated with junctions or parallel relationships. After that, 3D features were generated from groups of matched image features to generate rooftops. Expandable Bayesian networks (EBN) are used to combine evidence from numerous images. The EBN method is specified, and is based on the structure of the input data at runtime, to instantiate the structure of the evidence (Kim and Nevatia, 2003)). Finally, overlap and rooftop analysis is completed to create the final building models, which, even on complicated buildings has been

successfully applied (Kim and Nevatia, 2004), although flat roofed buildings do present problems.

As well, Baltsavias et al. (1995) conducted research in order to use DSM and aerial image for building extraction. The algorithm follows the sequences described; first a DSM has been constructed from the images, and this is later used to specify a building's rudimentary location. Consequently, this provides information for matching building location with the extracted building feature from the image. These data provide information to make a rough assumption about buildings, which can then be exploited to assemble both features in 2D and 3D and construct the 3D roof. It is clear that these algorithms require a detailed DSM. However, the automatic extraction quality is dependent on the amount of human intervention.

This review of building footprint extraction and 3D modelling algorithms processing different sources such as LiDAR, aerial or UAV photography shows adequate results but they presented cost challenges or restrictions to commission. The literature shows that there is still no robust method for building footprint extraction or 3D construction using solely data based on satellite images.

2.5 Summary

This chapter has reviewed the literature related to the merging of DSMs and building footprint extraction and associated problems. Since the DSM has become an important product that can be used in different engineering fields, it is clear that there is an increasing amount of research regarding merging DSMs using different strategies. It can be inferred that none of the researchers have investigated Bayesian approaches for the merging, although the literature indicates that Bayesian approaches have a positive effect in minimizing uncertainty; merging data sets is inherently uncertain thus approaches, such as Bayesian approaches, which minimize uncertainty are worthy of investigation.

Regarding building footprint extraction and 3D city modelling, it can be noticed from the literature that the buildings extracted from satellite imagery usually included errors which may be too large (such as edge irregularities) for some 3D modelling applications. However, the other sources of data, excluding Satellite imagery, have been predominately successful for building extraction and 3D modelling. For that reason the

gap that existed due to irregular building extraction will be treated, in this work, by amending the algorithms that have been used where they were successfully applied for extracting building footprint such as that of Wang et al.(2006).

Chapter 3 Digital Surface Model (DSM)

A DSM is the elevation representation of both the terrain and the objects close to it (such as buildings, pylons and trees) represented digitally (numerically). The DSM concept emerged from the 1950's innovation by Miller and Laflamme of the Digital Terrain Model (Miller and Laflamme, 1958) cited via (Li et al., 2005)), which they proposed be implemented in civil engineering and to monitor the surface of the Earth. But, Miller and Laflamme (Doyle, 1978) used the term DTM to represent the ground surface using horizontal geospatial coordinates X,Y and vertical characteristics represented by the height, Z.

Doyle (1978) demonstrated that more characteristics than only height can be used in a DTM, such as soil texture, the value of the land, land use and others, but height has been most widely implemented. Li et al., (2005) support the view that DTMs were in use earlier, suggesting that in ancient times in order to manifest the real 3D surface of the terrain, semi-symbolic and semi-pictorial depictions were used.

It is obvious from the literature that the term to be used was not constrained to DTM; different terms evolved, in different countries, such as Digital Elevation Model (DEM), Digital Height Model (DHM), Digital Ground Model (DGM) and Digital Terrain Elevation Model (DTEM). All these names have been introduced to represent the spatial ground elevation (Petrie and Kennie, 1990). For instance, DEM is used in the USA, DHM in Germany, DGM in the UK, and DTED was invented and implemented in the USA by USGS and DMA (Defence Mapping Agency). There is some contradiction in the use of the term DEM: according to some references (Li et al., 2005) it records the greatest height in the measured area such as the buildings' roof and top of the tree, while for others (Peckham and Jordan, 2007) it is an alternative to DTM. A DSM can now be clearly distinguished from other products because it includes the objects above the ground such as houses, trees, poles, etc. as shown in Figure 3.1.

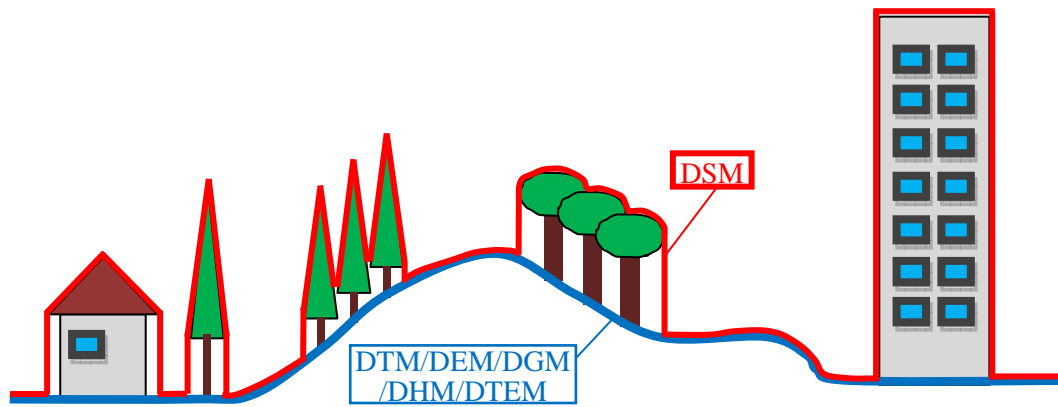


Figure 3.1 The comparison between DSM and DTM/DEM/DGM/DHM/DTEM.

This chapter will outline the sources of DSMs: the types of satellite images used; techniques used to acquire a DSM; processing the implemented data in the research; and measuring the uncertainty in the DSM.

3.1 DSM Format

The DSM may be said to be represented in an image format (i.e. graphically) or by a mathematical model (see Figure 3.2). However, numerous data structures have been implemented to store the data. The most common structures, and both depend on the point model to form the representation, are: rectangular grid representing the DSM in matrix form; or, as a TIN (Triangular Irregular Network) structure (Li et al., 2005). From these two methods, the grid data structure is considered the most common. The advantage of this structure is that the elevation can be allocated to a row or column. However, the advantage of the TIN structure, in which data are recorded as a triangular network, by connecting the points (of known X, Y and Z) by lines to form triangles, is that it can provide better information for an area that is multifaceted.

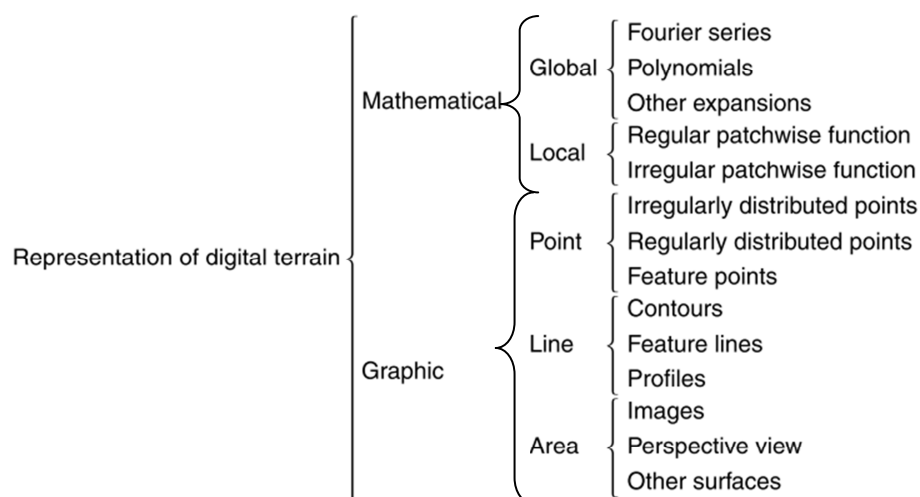


Figure 3.2 Digital terrain model format categories (Li et al., 2005).

Agouris et al. (2004) argue that the TIN method can be used to represent the ground more efficiently and realistically than the grid approach, because it provides elevation for critical points, as in the case of hilltops, and breaklines at elevation discontinuities, as in the case of building rooflines, valleys or ridges. In the grid method, it is difficult to sample the highest elevations as they rarely fall on the grid point, unless they coincide with the sample distance. The specific data structures that are used in this research's case studies are represented by a DSM with a grid spacing of 0.5m and 1.0 m. Although it is likely to be a more accurate model if the TIN is used, a grid is more convenient in data processing and it is assumed that the highest (and lowest) points are on the grid intersections.

3.2 Techniques for DSM Creation

Different systems can be used to acquire data for the DSMs. For instance a passive system, such as using aerial or satellite images records the reflected electromagnetic radiation from the object on the terrain surface or from the terrain surface itself, using sensors. Alternatively, an active system, such as LiDAR uses a device to transmit and receive the reflected electromagnetic waves of the object or terrain. LiDAR is suitable for areas of only small-to-medium size, due to its high cost. Images gathered from a passive system are suitable for all scales but are considered to have lower accuracy than LiDAR.

Technologies can be ranked by the accuracy of the data gathered which is subsequently reflected in the accuracy of the DSM produced. However, the accuracy of the produced DSM is also affected by external factors such as the project requirements and the allocated budget.

3.2.1 Geodetic Survey

A geodetic survey involves acquiring data from field measurements using traditional topographic survey instruments. Earlier, this method involved gridding the area and taking a measurement at each grid intersection point using a level or theodolite. The accuracy of the produced DSM, in this method, is considered to be very high, since it depends on precise instruments. Recently, accuracy and survey simplicity have been increased due to the application of computerized total stations for measuring the ground points and other objects. Moreover, the points can be referenced to the coordinate reference system using a GNSS station. The use of geodetic survey DSMs is limited to small projects or to evaluating other types of DSM, because of cost and time constraints. It is also possible to use RTK-GNSS to get direct measurements for the area of interest to a very high accuracy; consequently, this is currently considered the very best among all methods with respect to the accuracy. Furthermore, it is possible to use RTK GNSS points as a reference for evaluating DSMs produced from LiDAR sources (Chang et al., 2004).

3.2.2 LiDAR Survey

LiDAR (light detection and ranging), is considered to be cost effective with height accuracy RMSE values reaching at least 0.12m (Bilskie and Hagen, 2013) and even to better accuracy when the density of the laser pulses are increased either due to lower flight altitude, increasing the flight overlap over the specific area, decreasing aircraft speed or by increasing the pulse frequency (Jakubowski et al., 2013). The resulting DSM is considered to be the most accurate for topographical data acquisition (Polat and Uysal, 2015). This method is categorized as an active method; it depends on transmitting and receiving signals, and therefore it can be used at nighttime also. The acquisition of LiDAR data includes the incorporation of three different technologies (ASPRS, 2015): a laser scanning system; an inertial motion unit, IMU, (also known as an inertial mapping system - IMS, inertial mapping unit - IMU or inertial navigation system - INS); and, GNSS receivers, all of which are carried on an aircraft, see Figure 3.3.

The principle of measuring the position of a point using this technique is based on measuring the required time for the laser pulse, emitted from the sensor, to hit the object and return. The measured time for the laser pulse to travel, can be used to determine the distance. Meanwhile the angle that the laser is emitted can also be recorded. From these data, and using the position of the sensor, from the onboard IMU and GNSS, the 3D coordinates of the reflecting object can be determined (Carter et al., 2012).

The geolocation of the acquired LiDAR coordinates are calculated for the captured area, which is called a swath. LiDAR is considered suitable for large and small-scale projects, but the disadvantage is its high cost, and that it does not provide measurement specifically at the break lines, since it gives the coordinate of the transmitted laser point. In addition, it is affected by clouds (or their constituent water droplets) which reflect the pulses before they reach the ground surface. This technique can also be carried out from the ground using terrestrial laser scanning instruments, in order to construct a DSM for the area.

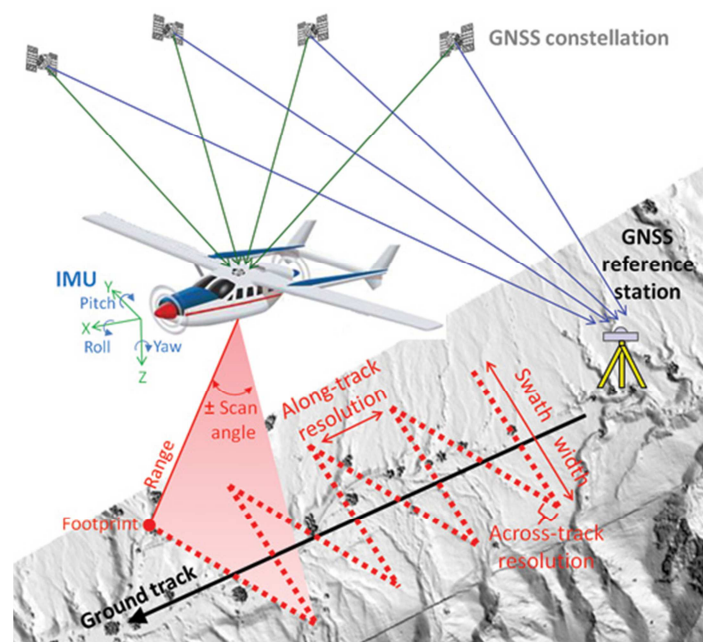


Figure 3.3 The principle of acquiring the ground coordinates using LiDAR system (ref: <http://www.imagingnotes.com>).

3.2.3 Photogrammetry

Photogrammetry yields quantitative and qualitative characteristics of specific objects from images. Its original use for mapping purposes was in 1849 (Wolf, 1983), and paralleled the introduction of photography, using elevated platforms or balloons, followed later by aerial photographs. By applying stereo-photogrammetric techniques, it is possible to produce 3D measurements. Different techniques and methods have been

used to generate the 3D data from the photography (or other imagery) starting from parallax bars and mirror stereoscopes through optical mechanical stereoplotters to digital photogrammetric workstations which exploit computerized algorithms (Leica Geosystems, 2013). In the 1970s, the introduction of civilian use satellite imagery, led to massive coverage. Miscellaneous satellite images can produce overlapping images for areas of interest, thus 3D measurements are generated. Images from such as SPOT, Ikonos, Geoeye, WorldView-1, -2 and -3, Pleiades, and others, have been continually improving.

Photogrammetric techniques, using aerial or space images, are more suitable than direct measurements for low or limited access areas, such as forest or politically unstable areas such as war zones, but the main shortcoming is that these techniques are affected by cloud and shadow.

3.2.4 InSAR

Interferometric Synthetic Aperture Radar (InSAR) is considered the most widely used technique to generate terrain models, particularly because of its use over large areas (Toutin and Gray, 2000; Toutin, 2000). The final product is partly dependent on the radar wavelength. Longer wavelengths (e.g. L- and P-bands) can produce DTMs whereas the X- and C-bands are more likely to produce DSMs (Papasaika-Hanusch, 2012). The active nature of a spaceborne radar system means that SAR images can be acquired at nighttime and are not restricted by cloud cover. The most recent generation of SAR satellites can now acquire SAR images with up to 1m resolution, such as those in the TerraSAR-X ‘Spotlight’ imaging mode. The Shuttle Radar Topographic Mission (SRTM) was an InSAR campaign, which is still a widely used source for global elevation data of high quality. The principle of obtaining 3D measurement from InSAR is based on recording signals for an area through two different passes or one pass using two different sensors separated by a baseline Figure 3.4(a). Data can be acquired from one pass as shown in the Figure 3.4(b) implementing the equations below (Abdelfattah and Nicolas, 2002):

$$\sin(\theta - \alpha) = \frac{(D + \delta D)^2 - D^2 - B^2}{2DB} \quad 3-1$$

$$z(D, \theta) = H - D \cos \theta \quad 3-2$$

InSAR products are almost global, following the SRTM campaign, which was completed within just 11 days in the year 2000. Currently these data are available at no cost and can be downloaded for most parts of the earth with resolution up to 30 m and an accuracy of 10-29m depending on the location (Li et al., 2013b). The TerraSAR-X satellite was launched in 2007, and later joined by its sister satellite, TanDEM-X, in 2010. The recent SAR accuracy has been raised to better than 10 m and 2 m regarding absolute and relative accuracy respectively (Dowman et al., 2012).

Limitations of InSAR can include geometrical distortions in mountainous terrain, as well as problems relating to path delays in the radar returns due to atmospheric water vapour. Different land cover can also determine the success of InSAR-generated models, with bare rock often generating better results compared to the less favourable conditions of dense vegetation.

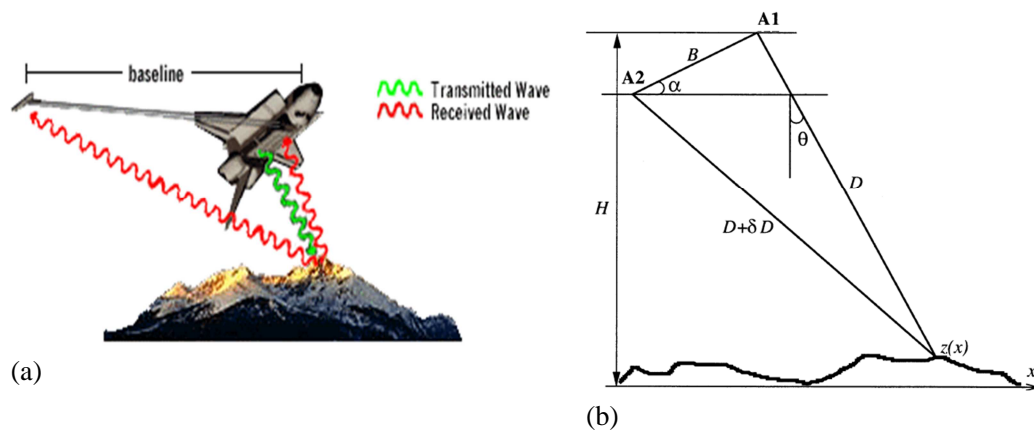


Figure 3.4 InSAR technique for DSM generation: (a) SRTM showing the used waves in DSM Generation <http://www2.jpl.nasa.gov/srtm/instrumentinterferometry.html>; (b) the schematic diagram for a single spaceborne pass (SRTM). H is the sensor altitude, B is the baseline length, D is the slant range and δD is the slant range difference, θ is the look angle and z is the topographical point height (Abdelfattah and Nicolas, 2002)

3.2.5 Digitizing Topographic Maps

Contour maps are the last source considered for a DTM. Contour maps are the most recognizable product representing the terrain, used since at least the 18th century (Li et al., 2005). The data are acquired by digitizing the contour lines from the map. This method is time consuming and the accuracy can be low in terms of repeating or missing information from the map (Doyle, 1978).

This source is limited to DTM production and does not include the buildings' cover, it can be modified to provide DSM production if the buildings' heights have been recorded also, at some stage. This method can be applied using large-scale topographic maps.

3.3 Study area and Acquired Images Specification

Since the main aim of this study is merging DSMs and the feature extraction that is related to buildings, the acquired images were of an urban area with different architectural styles of building. The area that has been selected is in Glasgow's 'West-end'. Further, the selected area was well within the city and it offered easy access for the processing and validation stages. It was originally specified that the area of interest be an urban area, include different types of structures such as small and large buildings, of simple and complex shape, and also contain some open areas such as roads, green areas and trees.

Two sources of satellite images have been acquired with the specifications as shown in the Table 3-1, the first source was from the Worldview-1 (WV-1) sensor from DigitalGlobe organization, dated 24May, 2012 covering 100km^2 , while the other was from the Pleiades sensor from Astrium Geo-information services, dated 09July, 2013, covering 25km^2 . However, for the purpose of the study a common part has been selected for the DSM merging, building footprint extraction and 3D modelling. The study area that has been selected for DSM generation and orthoimage production was limited to an area of 10.25km^2 . The coordinates of the area bounded by the Pleiades and WorldView-1 images are (417809 mE, 6191335 mN), (420528 mE, 6195105 mN) referenced to the UTM-Zone30 'North'.

For the purpose of minimizing processing time, the area selected, for both examining merging and building footprint extraction, is a representative sample of the whole area. Although somewhat suburban, the selected area has similar characteristics to many others in the UK and beyond, in terms of the range of building (multi-storey, detached, semi-detached, terraced and commercial or institutional) and land cover types (trees, open grass, sports areas, gardens, car parks, roads). Therefore, the algorithms developed in this work can be applied in the future in any other area, with different extents.

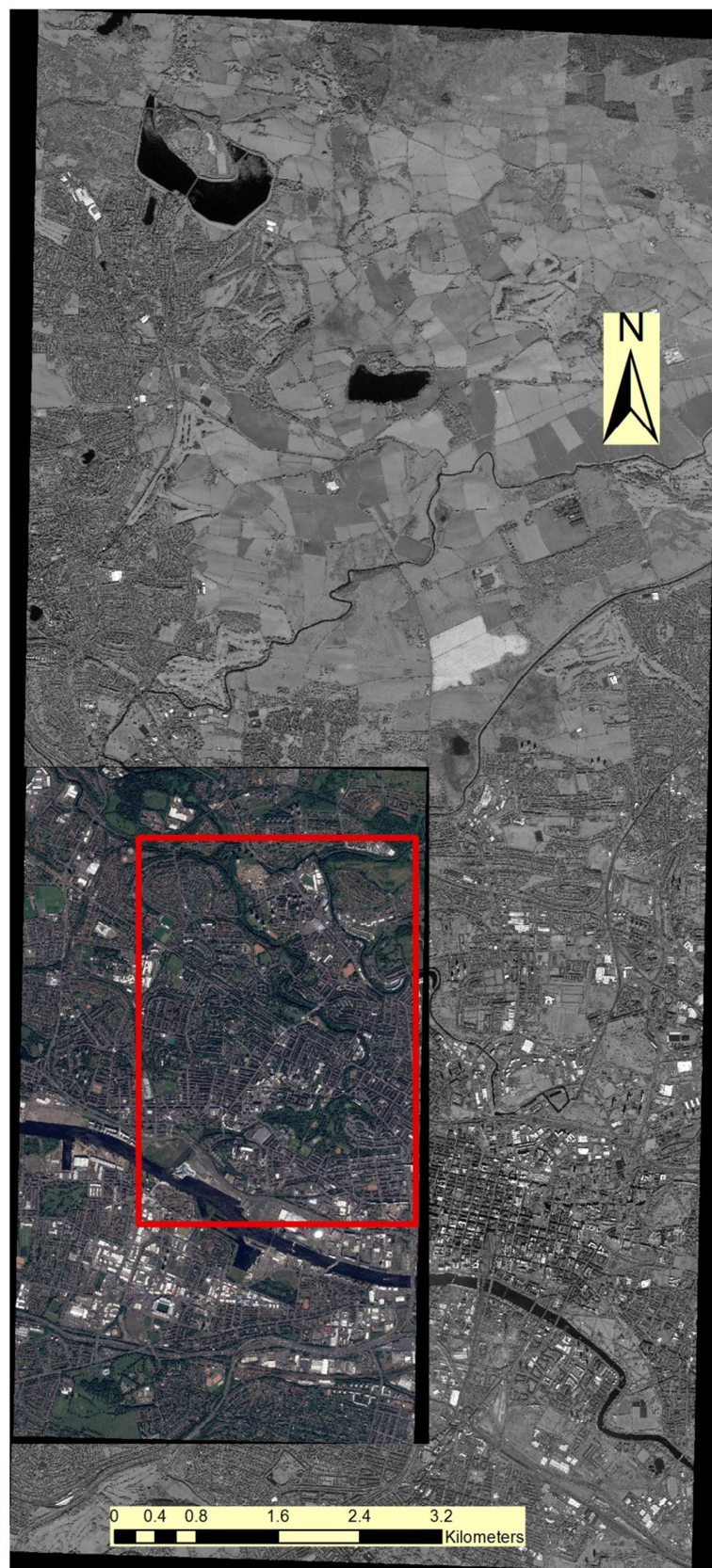


Figure 3.5 The acquired satellite images for Glasgow WV-1 (background) and Pleiades satellite imagery (foreground), with the study area boundary superimposed in red.

Table 3-1 Characteristics of the satellite imageries used in the research

Image ID	Acquisition Date	Max off Nadir(deg.)	Sun Elevation(deg.)	Image Type	Band Pan/MS	Res. GSD (m)
WV-1 image1	24/05/2012 11:42:49.42	17.33	54.59	Panchromatic	1/-	0.5
WV-1 image2	24/05/2012 11:43:28.78	21.01	54.63	Panchromatic	1/-	0.5
Pleiades image1	09/07/2013 11:35:44.1	11.91	55.46	Pansharpened	1/4	0.5
Pleiades image2	09/07/2013 11:36:25.8	14.03	55.51	Pansharpened	1/4	0.5

3.3.1 Satellite Image Acquiring Geometry

The data that have been employed in this research were sourced from satellite images. Two sources of satellite images have been used: WorldView-1 and Pleiades. Digital images can be classified into two main types. The first is static, the image that is acquired from aerial photography (either analogue or digital) which is also called frame imagery, and captures the whole scene at once. The other is dynamic, which is classified into three types (Mikhail et al., 2001). The first of these is the whisk-broom or point sensor type that acquires the image pixel by pixel, a solo pixel will be acquired continuously at a specific time, with the operational geometry working across the track pixel by pixel to the end of the track, then it starts again. The part of the sensor that is used to capture the image is called a framlet (Figure 3. (a)). This technique is used to acquire multispectral scans such as Landsat-1 and 3 satellite sensors; the technique is also followed in airborne LiDAR.

Another type of sensor is called a push-broom sensor. The geometry of the push-broom is shown in Figure 3.(b). It consists of a group of sensors arranged as a strip, and sensing across the track on the ground. In this technique, a group of sensors moves along the track and continuously acquires images until it has covered the required scene. The push-broom line-sensor is similar to acquiring a frame image but is moving continuously.

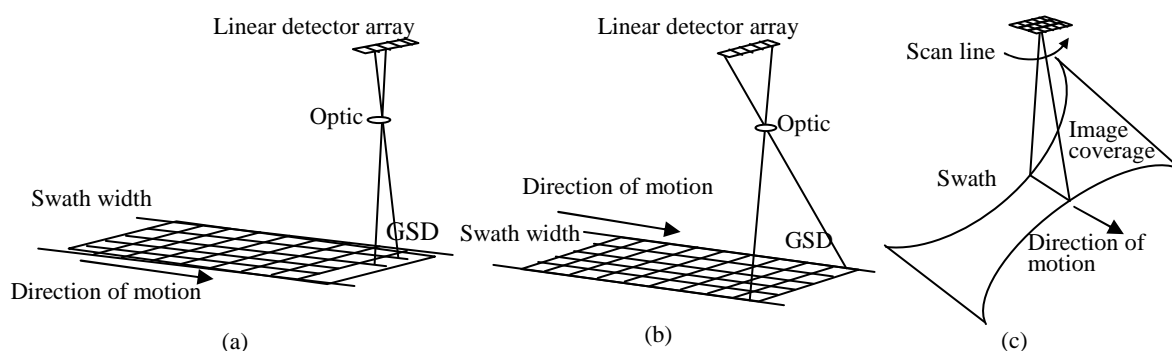


Figure 3.6 Types of satellite sensor image scanning explaining along track (a) whisk-broom acquisition principle, (b) using push-broom acquisition, (c) panoramic image acquisition.

A push broom image has a weak geometry, because each strip of the acquired image is independent from each other, although it is constructing the same scene, which, consequently, causes each strip to have its own position and orientation. This inconvenience can be slightly reduced by using GNSS (Global Navigation satellite System) receivers or an INS (Inertial Navigation system). Some agencies such as MOMS-02, have tried to solve the situation by activating more than one sensor, for example triple sensors, this leads to the capture of each pixel from three different locations to create a rigorous geometry for the calculation of point positions (Mikhail et al., 2001). As a result, this will permit the correction of any one part of the image and not require the whole scene, which may be expensive.

The last type uses a panoramic geometry Figure 3. (c). The sensor that is used is linear, but swings perpendicular to the along-track direction, which consequently causes it to image a larger area at the terminals of the swathe. Sometimes this imagery is specified to capture from horizon to horizon. This geometry was originally developed for reconnaissance purposes since it covers such a wide area (Mikhail et al., 2001).

As already stated, and more details will be given in the next two sections, two pairs of satellite images were acquired – one panchromatic band pair from the WorldView-1 satellite sensor of DigitalGlobe, and one Pansharpened pair (each made from four multispectral bands, i.e., three visible and one infra-red and one panchromatic band) from the Pleiades satellite sensor of Astrium, that cover a common area, with a set of auxiliary files that facilitated the processing job.

3.3.2 WorldView-1 Satellite Imagery

The first satellite sensor used in this research was WorldView01, of Digital Globe. This sensor was an enhancement of Quickbird-2. This satellite was launched in 2007, providing panchromatic images only with a resolution GSD equal to 0.46m at the nadir. It has an extremely high capability for image acquisition that reaches 750,000km² per day, with a swath width of 17.6km. According to its US licence, the resolution for commercial use is limited to only 0.5m although the acquired resolution is better than that and reaching 0.46m at GSD at nadir (DigitalGlobe Inc., 2013). The image acquisition system is based on the push-broom principle that is 'a strip at a time'.

According to DigitalGlobe Inc. (2013) there are five basic products of the WorldView-1 imagery. All five of them are radiometrically and sensor distortion (i.e. geometrically) corrected by adjusting the pixels' brightness values for errors due to the sensor used, solar radiation wavelength dependence and atmospheric consequences (Richards, 2013). The ephemeris data, internal calibration parameters and attitude measurements are employed to adjust the radiometric and sensor distortions in the imagery. The final products are as follows:

- Basic (1B) imagery: suitable for complex photogrammetric operations and preferred by customers who have superior capabilities with respect to image processing. No map projections are implemented.
- Basic (1B) Stereo Imagery: this type of product is suitable for organisations with considerable capability in image processing and powerful software; it is suitable for DSM and 3D feature extraction. This type is dedicated specifically for stereo imagery, the target area has been specified exactly and the data has been acquired on the same satellite orbit.
- Standard (2A) imagery: this type of product is limited to projects that do not require high accuracy. It is projected based on a specific map projection and datum. It has also been normalized to the reference ellipsoid by applying a coarse DTM. In addition, the produced image is not Orthorectified since the amount of normalization is small.

- OrthoReady standard (2A) imagery: it has the same specification that is available in standard imagery (2A), but it has been projected into a constant elevation and not a reference ellipsoid, and is ready for orthorectification. The relief displacement is still present in the image.
- OrthoReady (2A) Stereo Imagery: this type is exactly similar to the above product, the only difference is that the client is provided with the stereo-imagery to cover the Area Of Interest (AOI), and the data are thus ready for the orthorectification of a small area. The Basic Stereo imagery is thus more suitable a large sized area and needs more user expertise because it has not been orthorectified.

With most imagery types, a set of support files has been provided to facilitate the further image processing. The type of acquired image for this research is OrthoReady (2A) stereo imagery, which is limited to one band image considered to be a panchromatic image, see Figure 3.5.

3.3.3 Pleiades satellite imagery

The other source of imagery is the Pleiades satellite; it is a more recent source than WorldView-1 since it was launched on 16 Dec., 2011 while the latter was launched on 18 Sept., 2007. This constellation of satellites consists of twin satellites 1A and 1B. It was launched into orbit in 2011; the satellites (1A and 1B) are offset by 180° from each other on the same orbit. The altitude of the satellite is 694km and has the capability to cover 1 million square kilometres per day, with a coverage swathe of 20km. Pleiades sensors have the ability to provide panchromatic and multispectral images with delivery resolutions of 0.5m and 2.0m, respectively. The original resolutions were 0.7m for the panchromatic and for the multispectral 2.8m. An algorithm, implemented by the Centre National d'Etudes Spatiales (CNES) resampled the products to the 0.5m and 2.0m (Astrium Services, 2012). The products are panchromatic, multispectral and Pansharpened imagery. The wavelengths are as follows:

The panchromatic wavelength range is 0.47-0.83 μm .

The multispectral wavelength ranges are:

Blue: 0.43-0.55 μm ,

Green : 0.50-0.62 μm ,

Red: 0.59-0.71 μm ,

Near Infrared: 0.74-0.94 μm (Astrium Services, 2012).

The images are acquired based on the principle of push-broom systems. The ephemeris, altitude measurements and internal calibration parameters have been used in radiometrically and sensor distortion corrections applied on all of the products. The products that Pleiades can deliver are, according to Astrium Services (2012), Primary and Standard Ortho, only:

- **Primary:** this is the ‘raw’ satellite image, using ‘push-broom’ scanning technology. This product is ideal for advanced operations such as the orthorectification and DSM generation used in this study. This product has been delivered with RPC files and a Sensor model in order to guarantee fully automated operations. The primary product is supplied in the WGS84 geodetic coordinate reference system.
- **Standard Ortho:** is the other product of the orthoimagery. It has been geo-referenced, and, in addition to that, relief displacement has been removed. It has been made ready for direct use in engineering and mapping projects. For the rectification, the worldwide reference 3D dataset Astrium is implemented, if available, otherwise SRTM has been used. The reference 3D dataset used is part of Astrium’s “Elevation 20” suite (Astrium Services, 2012).

3.3.4 RPC file

The Rational Polynomial Coefficient (RPC) file contains information about the interior and exterior orientations. It is very useful because it minimizes the time for image processing due to not having to embark on the interior and exterior orientation processes. All the required information is saved in the files that are provided with the image. The only limitation is that the specific sensor must be supported by the software otherwise it will not be possible to process the satellite image.

An RPC file was defined first by IKONOS in 1999 (Xiong and Zhang, 2009). Satellite images, that were not rectified are usually accompanied by RPC files which include normalization parameters and coefficients which can reach to a total of 80 coefficients

(Dowman et al., 2012). It is a model that stores the geometry of the camera at the time of acquiring the image. This file represents the analytical model for linking the image space (X, Y), which is represented by rows and columns, to the ground space (E, N, H) which is represented by latitude, longitude and elevation. The geometric processing and orthorectification production will be facilitated by using the RPC file, implementing only the DSM file for providing the elevation. In addition to that, the DSM can be obtained from the primary images using the RPC file only without using GCPs for the area; however, the GCPs are recommended in order to increase the accuracy of the results.

The RPC file can be used as an alternative to provide information from the source such as camera geometry, position and orientation. The RPC file is calculated based on calculating the recognized interior and exterior sensor orientation using the global navigation system (i.e. GNSS) and attitude knowledge from star sensor and gyros, with the system calibration. Thus around the scene region detailed layers with object points will be generated in a cube model, see Figure 3.7. Thus for any coordinate on the ground in the cube of points the image coordinates are determined by implementing available data from the system calibration as well as straight sensor orientation. A considerable number, properly spread around the scene region, of 3D objects and their corresponding image coordinates will be used to adjust the RPC (Dowman et al., 2012).

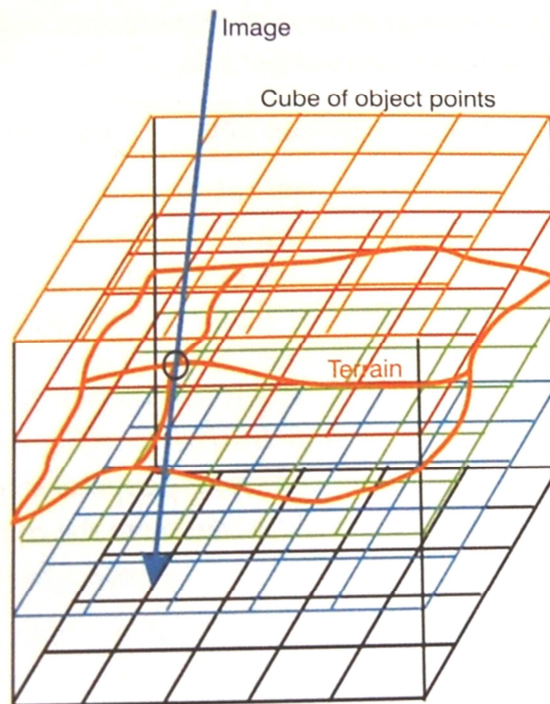


Figure 3.7 An illustration about determining the RPC using cube, shown object coordinate and image coordinate (Dowman et al., 2012).

3.4 Ground Control Points

For processing and validation purposes it was necessary to gather some control points in the study area, to be used as ground control point for the triangulation or as checkpoints for the quality assessment. For that purpose Leica Geo-system GNSS 1200 has been used to measure points on the ground as illustrated in Figure 3.8.

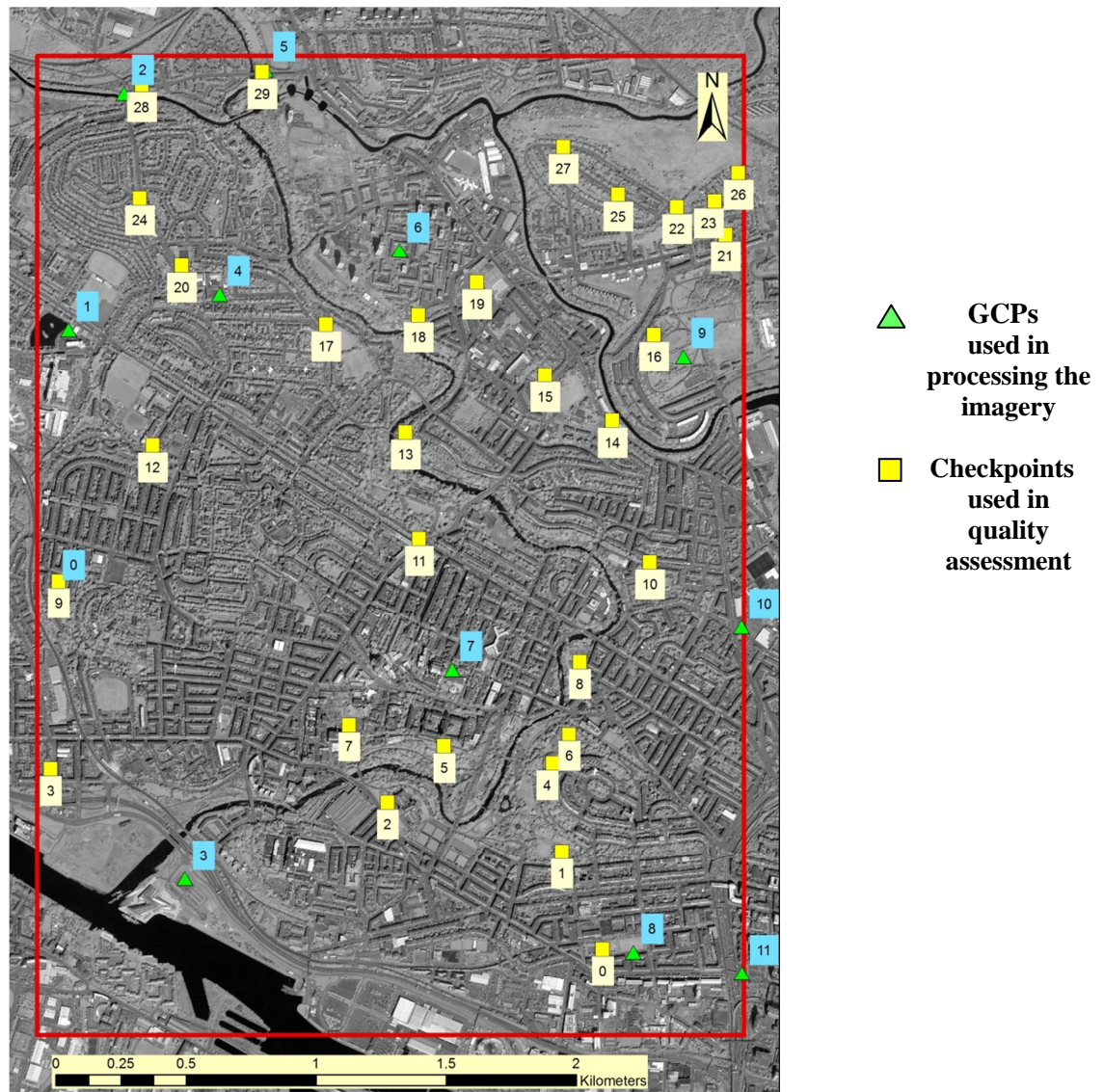


Figure 3.8 Numbered and measured Ground Control Points (GCPs) and Checkpoints (CPs) over the study area, which are used as control points and quality assessment points, respectively.

In total seventy-three (73) points were measured, see Figure 3.9. Among these 12 points were used in the processing of the satellite image at the triangulation stage, as listed in the Table 3-2, thirty-one (31) were used to determine RMSE for quality assessment for the subsequent weighting of the DSMs prior to merging and the remaining thirty (30) were used as checkpoints for evaluation (also referred to as validation) of the final merged result. The points were assigned Universal Transverse Mercator (UTM) zone 30N

Eastings and Northings coordinates and their heights referenced to the WGS84 ellipsoid. The reason for selecting this international coordinate reference system and not that of the Ordnance Survey Great Britain (OSGB) was to use the same coordinate system as the satellite images data files (i.e. RPC). Two types of points were measured at the site, namely Static GNSS and RTK (Real Time Kinematic) GNSS points. It had been the intention to use the Static GNSS approach for gathering GCPs for the satellite image registration using SOCET GXP, but a small evaluation was carried out at three different locations, in order to see the difference between the two approaches. The result of the evaluation showed that there is little difference between static GNSS observation for 15 minutes and an RTK GNSS observation for 1 minute, as shown in Table 3-3. In addition to the that test, another test was carried out on the University of Glasgow campus, also to see the differences between: long time static observation of around 7 hours; fifteen minute static observation; and, RTK GNSS observation. The results are as shown in Table 3-4.

pt.	E.(m)	N.(m)	GNSS Hgt.(m)	Description
0	417879.837	6193054.722	71.535	Corner at edge of Walkway in public park
1	417932.173	6194048.350	77.445	White road mark in car park
2	418144.800	6194961.668	94.350	White mark on the walkway
3	418378.985	6191933.824	59.765	White mark on the road
4	418512.838	6194183.439	107.536	White mark on the school's car park
5	418688.401	6195050.306	85.975	White mark on the road
6	419204.354	6194352.670	91.041	Corner at edge of walkway in public park
7	419407.195	6192741.306	101.148	White mark in the car park
8	420103.461	6191652.802	73.918	White mark in the car park
9	420296.281	6193946.721	129.709	Corner at edge of the grass in playground in the park
10	420519.729	6192905.267	78.478	White mark in the car park
11	420521.106	6191571.282	75.009	White mark of the bus stop on the road

The points that are used in the triangulation, quality measurement and validation were measured in two consecutive dates, 14 and 15 November, 2012. These points were measured after the WV-1 satellite imagery was acquired. The total number of points was 87. Regarding the Pleiades imagery, the same point which were measured for the WV-1 have been used. Among the points, only 73 were useful while the rest were difficult to

identify because they depended on road markings which had disappeared with time since the Pleiades was acquired eight months later, which was on 9 July, 2013. All these point were measured on a flat surface and far from buildings.

Table 3-3 Comparison of Static and RTK measurement using different observation durations and at three different points. At the mid time of the field work for three different locations

Point ID	GNSS surveying technique.	Date and duration	Easting (m)	Northing (m)	Ellip. Height (m)	Ortho. Height (m)	Geoid Sep (m)	σ -East. (m)	σ -North. (m)	σ -Height (m)	Posn+Hgt. Qlty (m)
Point_2	53B Static 15min	15/11/2012 11:29:54	257501.8897	665756.1575	73.2693	19.1011	54.1682	0.0002	0.0002	0.001	0.0011
Point_2	53b RTK 1Min	15/11/2012 11:51:02	257501.8859	665756.1599	73.2603	19.0921	54.1682	0.0028	0.0041	0.0072	0.0087
		Diff.	0.0038	-0.0024	0.009	0.009					

Point_4	12A Static 15mins	15/11/2012 12:32:04	255900.5349	666047.5545	59.7648	5.5720	54.1928	0.0001	0.0002	0.0009	0.001
Point_4	12a RTK 1min	15/11/2012 12:58:27	255900.5396	666047.5551	59.7679	5.5751	54.1928	0.0028	0.0035	0.0074	0.0087
		Diff.	-0.0047	-0.0006	-0.0031	-0.0031					

Point_3	40A Static 15mins	15/11/2012 13:32:04	255484.2526	668168.7766	77.4449	23.2642	54.1807	0.0003	0.0003	0.0022	0.0023
Point_3	40a RTK 1Min	15/11/2012 13:49:57	255484.2657	668168.7747	77.4169	23.2362	54.1807	0.0053	0.0056	0.0118	0.0141
		Diff.	-0.0131	0.0019	0.0280	0.0280					

Table 3-4 Static and RTK measurements made at the same point for different periods of time.

Point ID	GNSS surveying technique.	Date and duration	Easting (m)	Northing (m)	Ellip. Height (m)	Ortho. Height (m)	Geoid Sep (m)	σ -East. (m)	σ -North. (m)	σ -Heig ht (m)	Posn+Hgt. Qty (m)
Point_1	REF Static 7hr 1 min	14/11/2012 08:28:44	256902.4133	666545.6193	95.3638	41.1931	54.1707	0.0001	0.0001	0.0004	0.0005
Point_1	REF Static 23 mins	14/11/2012 08:02:14	256902.4105	666545.6209	95.3655	41.1948	54.1707	0.0003	0.0003	0.0014	0.0015
		Diff.	0.0028	-0.0016	-0.0017	-0.0017					

Point_1	REF Static 23 mins	14/11/2012 08:02:14	256902.4105	666545.6209	95.3655	41.1948	54.1707	0.0003	0.0003	0.0014	0.0015
Point_1	test RTK final	14/11/2012 15:37:01	256902.4121	666545.6068	95.3822	41.2115	54.1707	0.0044	0.0059	0.0121	0.0141
		Diff.	-0.0016	0.0141	-0.0167	-0.0167					

It can be noted that there is little difference between the static and RTK GNSS measurement, being less than 0.015 m in Easting and Northing, and less than 0.03 m in height. Moreover, the difference between the seven-hour static and approximately fifteen-minute static is less than 0.003m in all coordinates. Since the resolution of the satellite image is considered to be 0.5m, i.e. considerably more than the differences recorded between the GNSS approaches, and locating the GCP on the satellite image will also be within that 0.5m, it is therefore preferred to measure the rest of the points using the RTK GNSS approach, saving time and effort. It is assumed that the differences between static GNSS and RTK GNSS coordinates are small because the reference station, although south of the area, was very close to the measured points, being less than 4.5km distant with coordinates (418838.7349 mE, 6190590.333 mN), see Figure 3.9.



Figure 3.9 The distribution of GCP and Checkpoints with respect to the GNSS Reference Station in the study area

3.5 Satellite Image Processing

Processing satellite images has been achieved using techniques supported by Socet GXP software (Version 4.1) implemented in a digital photogrammetric workstation. Generally this involves two operations: (1) image acquiring, this includes selecting the required image type and the pre-processing stage that is needed to be applied prior to the production processes, and (2) image processing in order to generate the products that are needed for the project. Regarding the images acquired, the input products were OrthoReady (2A) Stereo Imagery from the WorldView1 sensor and Primary Type from the Pleiades sensor. Both of them came pre-processed and were accompanied by the necessary file (i.e. RPC file) for the subsequent production processes. Regarding the second operation, data processing was carried out with SOCET-GXP v4.1 software from BAE Systems.

The processing of satellite images is similar to the traditional photogrammetric flowline as shown in Figure 3.10. DPW processing comprises two main stages in order to make use of the imagery. The first stage includes orientation and triangulation, which implies finding the relation between the images and the ground. The second stage, which is found in the right column of Figure 3.10, includes DSM and Orthoimage generation in addition to the feature extraction for the application of updates and to generate GIS components. Although the above-mentioned stages are similar to those found in former photogrammetric approaches (i.e. analogue and analytical approaches) in DPW processing it is fully automated.

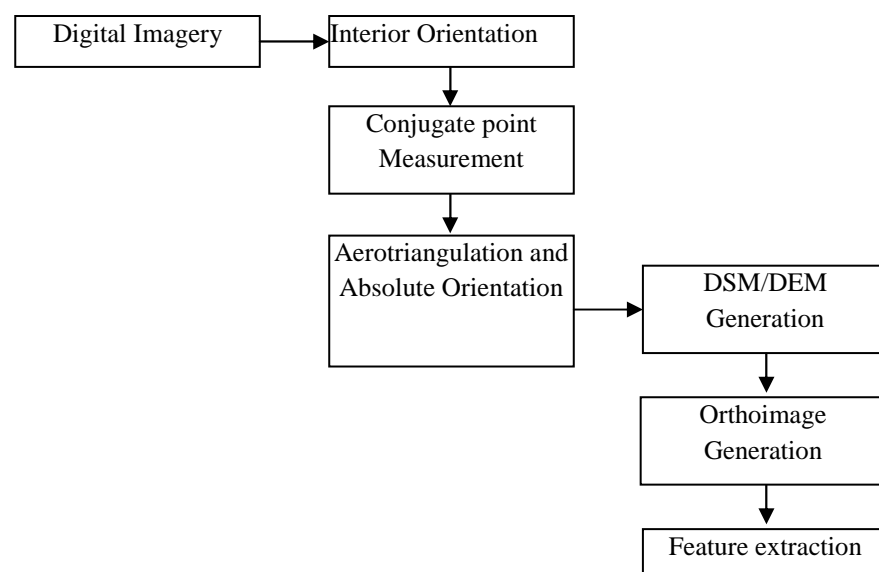


Figure 3.10 a schematic diagram illustrating the main components of DPW (Agouris et al., 2004).

The aim of the following sections is to illustrate the principle of photogrammetry with respect to satellite image processing and then the production stage.

3.5.1 Interior Orientation, Exterior Orientation and Triangulation

The first stage in processing images involves producing the sensor model that, among other things, defines the internal geometry of the sensor as it was during image capture. This definition, often called *interior orientation*, orients the coordinate system of the image pixels to the coordinate system of the image space. When traditional frame images are available, interior orientation uses fiducial marks and camera calibration will have taken place. Together the two processes provide data concerning: focal length; principle point location; optical distortions; and, the sensor location relative to the photo-coordinate system.

The second stage of the processing is exterior orientation when the camera station and its orientation, at the time of image capture, with respect to the coordinate reference system is calculated. Determining the exterior orientation elements (rotations of the sensor platform with respect to the coordinate reference system, and the coordinate reference system position of the sensor at the time of imaging) is also, in newer systems, considered a component of the triangulation process. Circumstances may mean that only initial approximate exterior orientation parameters are obtained, and later these parameters are refined.

In a satellite image the fiducial marks (found in traditional frame images) are not acquired, and an RPC file is provided instead giving the necessary interior and exterior orientation elements (Choi et al., 2012).

The sensor model, if defined completely, is also employed to support stereo processing in the image space. With a satellite image this definition is again found in the RPC file.

The purpose of a photogrammetric mapping project or photogrammetry in general, is to determine the ground coordinates of imaged points in the defined national or international coordinate reference system.

Early in the mapping project a data set, referred to as a ‘block’ of photography, must be created. Usually the ‘block’ includes two or more images that offer stereo (i.e. 3D) viewing through some common imagery. Attributes of the created ‘block’ comprise all the relevant details for a photogrammetric mapping project, including: the images that are employed in the project; data related to the images including their sensor models; camera/sensor orientations; mapping projection; reference spheroid; and, vertical datum. With these attributes, a mathematical relationship can be produced for the project, linking the images, their sensor models and ground control points, through a process called triangulation. Formerly the term triangulation, when referring to aerial photography would not have been applied to the very smallest blocks of (e.g.) only two images. But, now the term triangulation is used for simplicity (Gupta et al., 2013), regardless of whether the sensor platform was airborne or satellite borne, or whether the number of images is two, or more.

Triangulation uses first interior orientation results, which are defined by the user from external sources (e.g. focal length, radial lens distortion, principle point, fiducial coordinates from a camera calibration) and some image measurements, to finalise the image coordinates (x,y). Then, next, uses collinearity equations (e.g. 3-3 and 3-4 which present the relevant version of the collinearity condition) to solve for the exterior orientation elements (embedded in the a_i , b_i and c_i terms of the equations 3-3 and 3-4 below). However, in some satellite images (e.g. WorldView-1, Ikonos, Pleiades and some others) these elements are already provided in the RPC file, therefore the user does not need to determine them.

$$x = -f \frac{a_1(X_A - X_S) + b_1(Y_A - Y_S) + c_1(Z_A - Z_S)}{a_3(X_A - X_S) + b_3(Y_A - Y_S) + c_3(Z_A - Z_S)} \quad 3-3$$

$$y = -f \frac{a_2(X_A - X_S) + b_2(Y_A - Y_S) + c_2(Z_A - Z_S)}{a_3(X_A - X_S) + b_3(Y_A - Y_S) + c_3(Z_A - Z_S)} \quad 3-4$$

Where, for a particular point of interest:

x,y are its image coordinates;

X_A, Y_A, Z_A represent its ground coordinates in the selected coordinate reference system;

X_S, Y_S, Z_S represent the sensor’s ground coordinates in the coordinate reference system;

$$\begin{aligned}
a_1 &= \cos(\varphi).\cos(\kappa); \\
b_1 &= \cos(\omega).\sin(\kappa) + \sin(\omega).\sin(\varphi).\cos(\kappa); \\
c_1 &= \sin(\omega).\sin(\kappa) - \cos(\omega).\sin(\varphi).\cos(\kappa); \\
a_2 &= -\cos(\varphi).\sin(\kappa); \\
b_2 &= \cos(\omega).\cos(\kappa) - \sin(\omega).\sin(\varphi).\sin(\kappa); \\
c_2 &= \sin(\varphi).\cos(\kappa) + \cos(\omega).\sin(\varphi).\sin(\kappa); \\
a_3 &= \sin(\varphi); \\
b_3 &= -\sin(\omega).\cos(\varphi); \\
c_3 &= \cos(\omega).\cos(\varphi); \\
f &= \text{sensor's focal length};
\end{aligned}$$

where ω (or omega or tilt), φ (or phi, tip or pitch), κ (or kappa or swing) are the camera's tilts around the X, Y and Z axes of the coordinate reference system.

Triangulation for satellite images is similar to that for aerial images. The coordinates of many points should be measured on the ground; each ground point generating a pair of equations as above. The solution is strengthened if all equations for all points in an overlap area, existing between images, are simultaneously solved. The final result best represents the registration between the image space and the ground space within the area of overlapped images.

However, satellite imagery is characterised by a weaker geometry than aerial imagery, because it has been captured in space with a very high elevation, higher than aerial imagery. This consequently led to a lower base-to-height ratio, which leads to higher errors in elevations when the images are processed (Teo et al., 2010). From the triangulation viewpoint, satellite imagery can be considered to be time dependent, i.e. it is not a frame or instantaneously captured image. As a consequence this leads to an amendment of the collinearity equations for use with satellite images, and, therefore, a solution for different parameters needs to be achieved. With a frame image, only six parameters needed to be defined, three for the sensor location and three for the sensor tilt. However, for a typical image from a satellite borne sensor, many more parameters need to be determined, sometimes reaching ninety-four as in the case of Worldview01 images processed in SOCET-GXP. The reason is because a satellite image is captured line by line over a given time period, with each line being similar to a frame image and each line needing its own six parameters to define its exterior orientation. But for a satellite image,

a positive aspect is that the satellite platform is moving smoothly (mostly) which leads sensor location and sensor tilt parameters both being somewhat predictable functions of time, to be stored as RPC parameters hence supporting 3D point location determination (Xiong and Zhang, 2009).

The aim of triangulation is to solve for the exterior orientation parameters, including ω , ϕ , κ that are the camera's tilts around the X, Y and Z axes of the coordinate reference system, and the X, Y, Z coordinates of the camera station in the coordinate reference system. Initially approximate values are assigned to these parameters, but after a solution using a least adjustment technique a best estimate of these parameters is obtained.

The X, Y geodetic coordinates of points of interest are then obtained from their image coordinates through the application of collinearity equations. In this case, the collinearity condition is expressed by the collinearity equations:

$$X_A = X_S + (Z_A - Z_S) \frac{a_1(x) + a_2(y) + a_3f}{c_1(x) + c_2(y) + c_3f} \quad 3-5$$

$$Y_A = Y_S + (Z_A - Z_S) \frac{b_1(x) + b_2(y) + b_3f}{c_1(x) + c_2(y) + c_3f} \quad 3-6$$

where all terms are as defined for equations 3-3 and 3-4.

It should be noted that the elevation above vertical datum of each point of interest (i.e. Z_A) needs to be known to use equations 3-5 and 3-6 to determine X_A and Y_A . Z_A may be obtained through accessing an appropriate DSM, for example, or the prior application of the parallax equation (Wolf, 1983).

The process of triangulation that has been followed in this research was preceded by an additional preliminary step: digital relative orientation. In this step, the images are initially referenced to each other, not to the ground surface, through common tie-points. Searching for tie points may be achieved automatically using APM (automatic point measurement), or interactively. This process strengthens the application of the parallax equation, by removing y-parallax and improves the solution of Z_A .

3.6 DSM Generation

DSM production is the main task following the rectification of the stereo satellite images. It uses a correlation based matching technique. The produced DSM can either be represented in TIN or GRID format. Although the TIN format is more realistic, as has been mentioned earlier, in this research the focus will be on the grid since it is more suitable for the processing developed in this research. The main operator's task in digital photogrammetry is measuring the coordinates on the images. These points' positions are represented in rows and columns. The image matching operation requires finding a point in the second image whose location is known in the first image. Based on these points' coordinates the corresponding ground coordinates can be determined through an application of the parallax equation. In digital photogrammetry the searching method is fully automated. The DPW (digital photogrammetric workstation) comprises hardware and software, which represents the end-to-end requirements of the photogrammetric work such as: the ability to image process; 3D point measurement; automated and interactive point measurements; in addition to the conventionally required facilities such as data storage and the display of geospatial datasets and images.

It has been clear that the main challenge in photogrammetry is identifying matching points, which may be called corresponding or conjugate points, in both images automatically. This process is called image matching. This challenge is daunting because: the images used are taken from different viewpoints, which consequently causes the matched objects to be difficult to identify; some objects such as trees or buildings obstruct the view - which is called occlusion; and, points at elevation discontinuities are difficult to match. Shadow also has a great affect on hindering the matching. Gruen et al. (2000), claim shadows as the main obstacles for DSM generation, which consequently affects the precision and accuracy of the produced DSM.

The matching algorithm that has been followed in this research is NGATE (DeVenecia et al., 2007) which consists of blending two well-established techniques: area based matching and edge matching.

3.6.1 NGATE Technique

The algorithm that is used to create DSMs within the SOCET-GXP suite of tools is called NGATE (Next-Generation Automatic Terrain Extraction). As mentioned earlier in

section 2.2, it is based on the synthesis of two well-known techniques in computer vision, area based matching and feature based matching, each of them will be discussed later, individually. The reason for combining two methods is because area based matching cannot deal with the points at discontinuities, such as building edges or valleys. To overcome this problem feature or edge matching has been introduced, which is finding the corresponding edge points in both images (Zhang et al., 2007). This hybrid algorithm has lead NGATE to be considered a most powerful and favourable technique for urban and rural areas, in order to find very precise point coordinates for DSM creation. It compares favourably, both quantitatively and qualitatively, with algorithms that use area based matching techniques only (DeVenecia et al., 2007).

The NGATE algorithm is based on using both area based and feature based matching methods in a state that each of them supports the other, see Figure 3. The result of an area based matching technique leads to the edge matching process, because, without restraint, all the boundaries in the reference image can be considered as candidates of the corresponding edge in the search space, therefore it is necessary to limit the area of the search. Also it should be taken into consideration that it is very rare to have the same shape (length and orientation) for an edge in all the images, due to the different image perspectives and orientations. In addition to the already mentioned problem of occlusions and shadow, image noise can cause possible flaws in the edge detection algorithm itself. The area matching technique will initially create a rough DSM in the case where the conjugate pixel falls within a 5-10 pixels error; this will limit the search for the edge and cause it to fall within the same window. Meanwhile the result obtained in each matching will be used in order to partition the window based on the edge, and consequently, this will help to constrain the area matching.



Figure 3.11 A sample of the generated DSM: (a) the Generated DSM from Satellite imagery, 0.5m resolution, using the NGATE approach; (b) the corresponding part of the DSM is shown on the satellite imagery.

In addition to using both of the aforementioned techniques in elevation determination, there is still another challenge in image matching which may be considered to be a blunder in the measured elevations. Blunder detection and elimination has become a critical issue in automatic image matching and DSM production. To solve this problem, an algorithm called **back matching** is used in order to re-compute the elevation of the candidate pixel. This algorithm helps to identify blunders through verifying the fidelity of forward and backward matching. During the forward matching, a window is fixed in the left image and moving it along the epipolar line in the right image, until the maximum image correlation is found, and subsequently determining the elevation. While backward matching is fixing the window in the right image and moving it along the epipolar line in the left image in the same manner until getting the maximum correlation, then using this distance, the x-parallax, to find the elevation. If the difference of elevation at forward matching is significantly different from backward matching, based on a user defined threshold, it means there is a blunder and the elevation is unreliable (Zhang et al., 2006). Conversely, if the difference of elevation is zero or very small this means that the elevation is reliable and it can be used as DSM elevation. However if backward or forward matching failed to get a good correlation this means that it is difficult to match successfully, thus the point is unreliable also.

As has been stated NGATE is not efficient in areas of low texture variability, such as roofs of buildings (Zhang et al., 2007). However it provides robust 3D measurement at the locations with elevation discontinuities such as roof edges or corners, based on these discontinuity points the other intermediate roof surface points can be estimated using an interpolation method, see Figure 3.12 (Zhang et al., 2007). The NGATE technology is designed to very efficient when computing the elevation of each pixel in the image, and does not depend on post spacing (DeVenecia et al., 2007).



Figure 3.12 Imagery for 3D point extractions experiment: (a) imagery showing complex buildings and flat surfaces with little variability; (b) the 3D points produced with the NGATE algorithm, the cyan line vertices are representing the ground truth points, the red dots are representing the 3D points produced by the NGATE algorithm. The algorithm successfully extracted points at highly variable locations or at the edges, while it failed at the locations of low variability (Zhang et al., 2007)

3.6.2 Area Based Matching

It is clear that area based matching is well-established in image matching. It is based on a correlation, using a small window, usually 5×5 or larger, up to 15×15 . In this process, a window is fixed in one of the images and another window is traversed in the other image. In order to minimize the searching area and reduce the searching cost, the images are arranged so that the search is contained along a specific line, called the epipolar line. The window is moved pixel by pixel; at each location the normalised cross correlation value with the target window is recorded. Based on the result obtained from the correlation equation, the centre of the window location that gives the maximum correlation value will represent the conjugate point.

Area based matching is a well known algorithm and has been applied by many different researchers (Ackermann, 1984; Calitz and Ruther, 1996; Helava, 1988; Li, 1991; Okutomi and Kanade, 1992; Rosenholm, 1987; Zhang and Miller, 1997). The method that is implemented by NGATE is normalized cross correlation which is extensively used in computer vision (Vosselman et al., 2004). This method assumes that the intensity of the pixels is identical or similar in both target and reference images. Another assumption is the elevation inside the window is the same (e.g. there is similar x-parallax between the points). This is considered a limitation of this method in giving robust results. Moreover, regions with low texture intensity variation, such as roads, parking areas, building roofs, fail.

3.6.3 Edge Based Matching Technique

Edge matching, or feature matching, is used to improve the results of image matching. It is based on matching the points, along conjugate epipolar lines, in the images using an image-intensity matching technique. Zhang et al. (2006) referenced different algorithms that have been used in edge matching (Medioni and Nevatia, 1985; Ohta and Kanade, 1985). Although giving robust results when hybridised with area matching, it is not known to give as reliable a result as the image correlation method. This is because the edges are sometimes considered to be difficult to detect or it is not possible to detect them all.

In digital photogrammetry, there are some obstacles that face the edge-matching task, which can lead to fails or erroneous results. This is due to either no or a limited number of edges in the images, such as in flat areas. Popular algorithms (such as Canny Edge Detection) used in detecting the edges have not been considered robust and have not provided suitable edges, since the algorithm that is implemented in edge detection is based on the pixels' intensities and is still affected by noise and discontinuities. However, edge based matching, using an edge detection algorithm, is nevertheless considered an optimal approach, among all other approaches. NGATE combining area matching and edge matching gives an ideal result for matching either artificial edges (i.e. building, bridges) or natural edges discontinues (valleys and ridges).

3.6.4 DSM Quality

The 'quality' of the DSM is referring to the extent that the DSM is close to the truth (i.e. quality refers to the error that exists in the DSM). Terminology relating to 'quality' differs. Many (e.g. Weng (2002)) argue that the **errors** of the features that are represented in the DSM are impossible to measure. The reason behind this statement is that it is not possible to measure the true position of geometrical features that appear in geographical data sets. Instead, the term **uncertainty** is suggested (Weng, 2002) to represent the quality of the DSM. To determine uncertainty the data set obtained should be evaluated against reference data that is known to be more accurate, for example obtained from field measurements or LiDAR, which eventually will be used to find the differences in heights between the two data sets, to determine uncertainty. Since the DSM is used in different applications, so the quality of the employed DSM must be defined in order to evaluate the integrity of these applications. Karel et al. (2006) claim

that there are three elements that can be used to define DSM quality, and these are precision, reliability and accuracy. The last one is the most practical.

The interior quality or precision can be obtained from the estimated standard deviation, or estimated precision, of each value in the DSM, determined from a least squares adjustment procedure. Karel et al. (2006) recalls that the estimated precision can be controlled by redundancy, which minimizes and constrains the random error. Alternatively, redundancy can be used to measure reliability. The reliability can be represented by producing a Distance Map (Karel et al., 2006). The distance from input data and produced pixel can be represented in a Distance Map; the smaller the distance the more reliable the pixel value. The other type of the quality measurement is external quality assessment. This may be called accuracy measurement. The accuracy can represent the error. Generally, there are three types of errors: blunders, random errors and systematic errors.

Maune (2007) reminds us that only random error, in theory, is linked to accuracy. Li (1990) and Weng (2002) list some of the statistical measurements that can be used to estimate the uncertainty or accuracy of the DSM, such as RMSE, mean (i.e. bias), estimated Standard Deviation, and finally the maximum and minimum height differences between the reference and generated data set. These statistics should be determined after eliminating blunders from the dataset. However both Li (1990) and Weng (2002), extol RMSE. It is possible to derive the vertical accuracy from RMSE of the vertical at a recognized confidence level, which is generally 95% (Maune, 2007).

3.7 Orthoimagery Generation

The next stage after creating the DSM is orthoimage generation. In this research, the orthoimage has been used as a source to help detect and extract the building footprints, because the buildings in the DSM can be blurred and not clear - therefore difficult to detect. The orthoimage generation follows the DSM production because one of the requirements of orthoimage production is the availability of a DSM and the exterior orientation results. Orthoimagery implies generating a new image with particular geometric characteristics, from the original image, see Figure 3.3. Orthorectification comprises removing the influences of perspective and relief displacement that existed in the original image, in conjunction with image resampling, in order to produce an image

with a constant scale pixel.

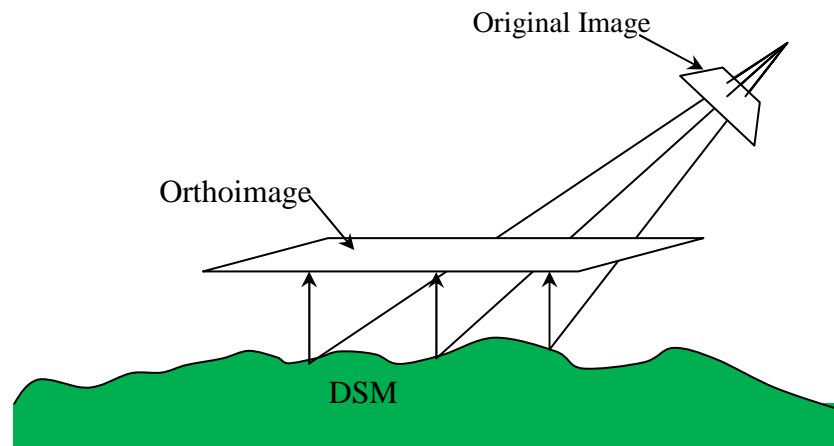


Figure 3.13 Illustration for the orthoimage production from the original image using the DSM ((BAE Systems, 2013) cited in Agouris et al., 2004)

3.7.1 Quality of Orthoimagery

Agouris et al., (2004) state that the main influences on the accuracy of generated orthoimagery are the spatial resolution of the original image, the accuracy of the exterior orientation, and the accuracy and resolution of the employed DSM.

For building detection, if the DSM to be used is produced from satellite imagery, then it is preferable to use a DSM with a lower resolution than the eventual orthoimagery. Practice has shown that, using higher resolution DSMs, the buildings edges are distorted, see Figure 3.13, and do not reflect the real situation (when straight edged building segments are expected).

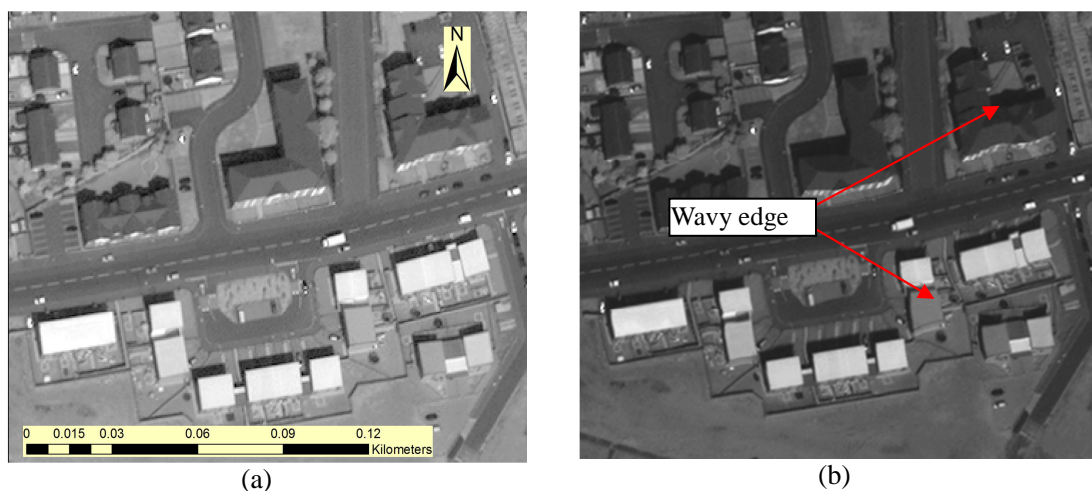


Figure 3.13 The effect of using different resolution DSMs in orthoimage production: (a) orthoimage produced using a low resolution 10m DSM, where the building edges are straight; (b) orthoimage produced using a high resolution 50cm DSM, where the building edges are ‘wavy’.

It is worth mentioning that there are some other problems arising in the orthoimagery related to buildings, which include the building's roof displacement and building lean. In order to remove the effect of the building lean and to shift the roof building to the true location, a detailed building feature that represents the building's model with its maximum height should be available (Agouris et al., 2004); obviously these may not be available. Mikhail et al., (2001) argue that the effect of relief displacement, and consequently building lean, can be minimized by using a camera with a longer focal length and working only on the inner section of the image rather than the outer part. They claim that a short focal length increases the effect of relief displacement. In this research, the imagery with lower nadir angle has been selected in order to minimize the effect of the building's lean.

Chapter 4 Bayesian Statistics

Statistical analyses are widely used by different researchers to summarize datasets through constructing probability models eliciting inferences related to current or future events. Such data analyses recognize that a population defined by a probability distribution function is expected to include any members of that population. The main methods that are applied in statistics are either the classical, or frequentist, methods or the Bayesian methods. But statisticians' attitudes towards Bayesian methods are still ambivalent, despite evidence of better results than when using frequentist methods in some cases (Carlin and Louis, 1996).

Statistical techniques are used when there is uncertainty in data. This uncertainty can be reduced by adding more data leading to it being minimized (FDA, 2010). However this continuous sampling of data may not be cost effective and, furthermore, it is sometimes difficult to provide more data. For that reason when the sample number is limited, it may be preferable to use a Bayesian approach and exploit *a priori* values in the calculation to minimizing the uncertainty (Berry, 1997; FDA, 2010). This chapter introduces the main statistical tools related to the Bayesian approach and the difference between the frequentist and Bayesian approaches.

The chapter will look at the basic elements of statistics that are considered as the main ingredients of Bayesian statistics. Furthermore, it addresses the difference between the Bayesian and the frequentist approach.

4.1 Probability

Probability, in daily life, is used to represent confidence or uncertainty regarding an unknown quantity, and is often used informally. However, these informal probability expressions can be transformed into formal mathematical expressions; uncertainty can be represented numerically by utilizing probability.

The basis of statistical techniques is probability. Statistics deal with mechanisms that are uncertain, and probability deals with events that are uncertain, which means probability and statistics are manipulating uncertainty. Both Bayesian and frequentist statistics use probability, but each in a different manner. The probability in frequentist statistics is utilized to exploit tools for statistical inference while the probability itself is used, in

Bayesian statistics, as a tool for statistical inference. This means that probability in Bayesian inference is refined using unknown parameters (Link and Barker, 2010). For both approaches, the value of probability, enclosed from 0 to 1, is shown in equation 4.1:

$$0 \leq P(E) \leq 1 \quad 4-1$$

where P represents the probability value of the event E . Assigning probability to unknown parameters is considered the most difficult stage in Bayesian statistics (Link and Barker, 2010).

4.2 Random Variables

In the Bayesian approach, the result is represented by some random variable, rather than a real number derived from observations. All variables in Bayesian statistical analysis are represented by random variables (Beaumont and Rannala, 2004; Gelman et al., 2004; Held and Bove, 2014). The value of a random variable X is a real number from the sample space obtained by using a function for that purpose (Kobayashi et al., 2012; Leon-Garcia, 1994).

The random variable, denoted by the capital letter X , usually reflects an actual probability distribution, $f(x)$ with parameters that define the distribution, such as μ and σ , as used to define normal distribution (Ang and Tang, 1975).

4.3 Probability Distribution Functions

In statistics, the probability density distribution function (pdf) is used either to represent the population, that is to make inference about uncommon measurements, or, to represent random variables. As mentioned, a random variable can be represented as a distribution function. In a discrete random variable, the probability distribution is a representation of possible quantities that correspond to a record of probabilities. In the case of a continuous random variable, the pdf is utilized to represent the random variable's probability at a given interval by integrating the pdf within that interval.

There are different types (or families) of probability distribution functions used in simulating random variables, such as Normal or Gaussian, Exponential, Bernoulli, Poisson, Binomial etc. (Gelman et al., 2004; Leon-Garcia, 1994). The focus, in this section, will be on the distribution that is very widely used, namely Gaussian.

Of the distribution families, Gaussian (or normal) distribution can be used to represent a number of measurements (n) of a set of random variables X_1, X_2, \dots, X_n , which when independently and identically distributed have mean, μ , and standard deviation, σ , under the condition that $-\infty < \mu < \infty$ and $0 < \sigma < \infty$. The function that represents the Gaussian distribution is:

$$f(x_i|\theta) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right) \quad 4-2$$

where θ represents the values of unknown parameters, in this Gaussian case μ and σ .

The Gaussian pdf is represented as a ‘bell shape’, see: Figure 4.1; the mean parameter, μ , which is sometimes also called the expected value, controls the location of the bell shape while the standard deviation, σ , governs the shape of the bell - whether wide or narrow. The shape is symmetrical, which means that the mode and median are located at the same location as the mean. The Gaussian pdf is called unimodal as it has a solo (peak) modal value. It is feasible to use a Gaussian distribution for the representation of the relative frequencies of x -values produced by a random process. Moreover, degree of belief (the amount of belief about the certainty regarding different propositions) can be represented by using a normal distribution for different x -values (Kruschke 2011). These two concepts: relative frequency and degree of belief can be used effectively when finding the parameters of the model, based on the assumption that errors are distributed using a Gaussian distribution.

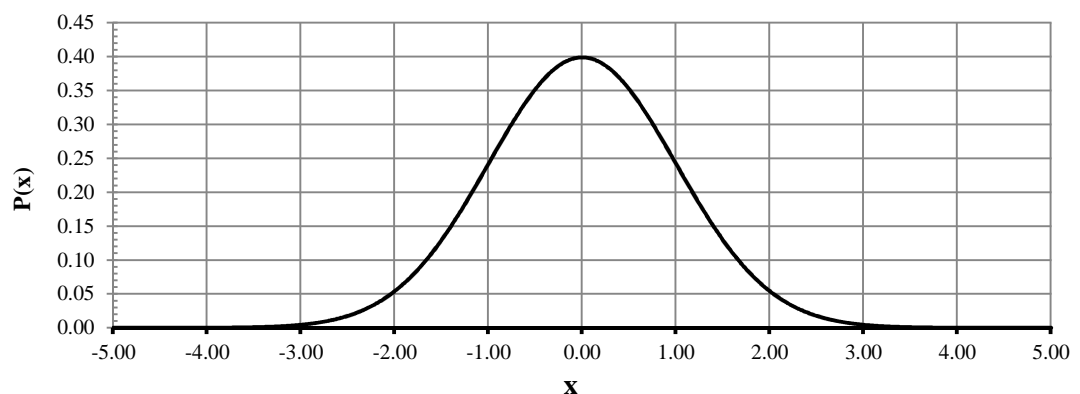


Figure 4.1. Gaussian Probability Density Function with mean (μ) equal to zero and variance (σ^2) equal to 1.

It is possible to use other forms of distribution instead of the Gaussian distribution, based on the situation and the circumstances of the problem, such as exponential, Bernoulli, Poisson and other distributions (Gelman et al., 2004).

4.4 Statistical Inference Approaches

Statistical inference requires that a model should be implemented and then the required parameters can be inferred. There is some variation in the use of the term ‘model’, but in most statistical situations, both the formula terms and the parameters are called the model. Selecting the most probable model best represents the situation being observed.

Statistical analyses, such as frequentist and Bayesian analyses are pervasive in different fields, and, although the frequentist dominates, the Bayesian approach is considered to be the most useful in complex situations (Shoemaker et al., 1999).

For long the frequentist, or classical, approach dominated, and has been widely used in various fields involving analyses and collection of data. In the late 20th century, due to the increased availability and power of computers, the Bayesian approach became very popular, since the simulation stage could now be carried out in a computer environment. It is still considered a state-of-the-art method, despite its 18th century origins.

Frequentist inference can be used to obtain sample data sets’ probabilities, by analysing the results of frequent surveys or tests. Frequentist inference gives precise results close to reality if applied correctly, however, it is sometimes applied incorrectly (Johnson, 1999). Bayesian inference can, on the other hand, be used to find directly the probability of the parameters of interest, depending on the sample space (Stauffer, 2008).

Supporters of the frequentist approach usually comment that the Bayesian approach is too sensitive to the *a priori* values and deals well with very specific problems but not all problems. Supporters of the Bayesian approach comment that the frequentist approach is inconsistent in the way it includes available data arising from observations to obtain the final result (Carlin and Louis 1996). Link and Barker (2010), when advocating Bayesian statistics to analyse data compared to the traditional frequentist method, identified the following advantages:

1. simplicity - it suits complicated models and data, and can be used in the situation where there is no frequentist solution available;
2. exactness - it can provide rational and reasonable results, even when dealing with a small sample size; and,
3. coherency – it is considered to use a straightforward interpretation technique with respect to the observed values while also being judged to be reliable in itself.

It is frequently quoted that “all models are wrong, but some are useful” (Box and Draper, 1987). This can be taken to mean that if the *a priori* data does not represent the prior belief accurately then it is regarded as “wrong”; nevertheless, the resulting *a posteriori* values are not necessarily lacking usefulness. The *a posteriori* data that arises from using assumed *a priori* data can still be considered to have the best possible probability of representing the unknown parameter (Hoff, 2009). This is justification for considering the Bayesian approach to be a more robust method for data inference than the frequentist approach.

4.4.1 Likelihood Approach

Likelihood represents the data revealed by all observations obtained in an experiment. It is expressed in probability form:

$$Likelihood = P(data|Model(parameters))$$

that is that likelihood is the probability of the parameters obtained from processing the observations, given a particular processing model. Among non-statisticians the terms ‘likelihood’ and ‘probability’ are often considered to be have the same meaning, however they are completely different among statisticians. Probability is used with parameters to determine the unknown results, and likelihood is used with the known results to determine the unknown parameters.

Probability is represented by $f(x|\theta)$. It is based on representing the random variable X (of which there are several measurements x_i) in order to calculate the parameters, represented by θ , which are considered to be fixed. In the case that there is more than one measurement, x_i , according to probability theory statistically independent

measurements of the pdf of the data $x = (x_1, \dots, x_n)$, given parameter θ , can be stated as the product of the *pdf* for individual measurements:

$$f(x = (x_1, \dots, x_n) | \theta) = f_1(x_1 | \theta) f_2(x_2 | \theta) \dots f_n(x_n | \theta) \quad 4-3$$

The pdf $f(x | \theta)$ and the likelihood function $L(\theta | x)$ are similar, but not the same. The total probability is between 0-1, but the total of the likelihood values does need not to sum to 1. The likelihood function, sometimes only called likelihood for simplicity, can be expressed as an inverse problem concerned with finding the pdf that represents the observed data, given the designed model. This is achieved by testing all the probabilities that the model produces and then selecting the density that most probably generated the data. The pdf, meanwhile, represents the distribution of the probability of continuous parameters (Held and Bove, 2014; Myung, 2003). According to James (2003) the principal behind the likelihood function $L(\theta | x)$ is that it offers information about the parameter(s) θ embedded in the experimental data $x = (x_1, \dots, x_n)$, by making inferences about the unknown parameter(s) θ .

The likelihood function, represented by $L(\theta; x_1, \dots, x_n)$, provides the means to find the parameter value θ given $p(x | \theta)$. In this case $p(x | \theta)$ is the joint probability mass function (pmf) or joint pdf of a collection of measurements x_1, \dots, x_n .

Maximum Likelihood Estimation (MLE) is used to examine the model that has produced estimated parameters. MLE uses a set of measurements and a model that can be exploited to represent the probability distribution of the variables of interest. MLE is used to make statistical inferences about the population of concern, using the likelihood model and the measured data. This is achieved by testing different parameter values and finding different probabilities then testing which of them agrees with the assumed probability distribution. According to Myung (2003) in the 1920's Fisher maintained that in principal MLE involves trying to find the probability that has the highest possibility of representing the observed data. This is accomplished by trying different parameters and selecting the one that gives the maximum result for the likelihood function $L(\theta | x)$. Thus the MLE estimate, equation 4-4, represents the parameter vector that was produced from searching the parameter space of the multi-dimensional data.

$$\bar{\theta}_{ML} = \arg \max_{\theta \in \Theta} L(\theta)$$

4-4

The Maximum Likelihood function behaves in the same manner if a logarithmic approach is taken. It is preferred to use the log of the likelihood function, for two main reasons. First it is clear that the probability value is between 0 - 1, therefore, especially for a large number of measurements, this leads to tremendous rounding and consequently truncation error, because the result of Maximum Likelihood is based on the multiplication of probabilities to produce a final value for the Maximum Likelihood; this problem exists even with most advanced computers. The second reason is that adopting a logarithmic approach transfers the processing to an addition task rather than a multiplication task, which is consequently easier to deal with it and simplifies the final equation. The logarithmic approach is further considered in section 5.4.2.

MLE is similar to least squares estimation (LSE) in the sense of finding the parameters to best fit the data, however there are two main differences between the techniques. First LSE tries to minimize the sum of squares of the predicted errors and there is no need to assume an error distribution while in MLE the result is obtained by maximizing the probability of fitting the model to the data having also modelled the behaviour of the data itself. LSE is based on making the fewest assumptions regarding the cause of the uncertainty during fitting the model to the data, to find the parameters. MLE, while considered a frequentist method, is considered to be more advanced in finding models' parameters; it tries to take into consideration the assumed error, as uncertainty, which is distributed based according to an assumed distribution model.

4.4.2 Frequentist Approach

The frequentist method is also known as the classical or standard method. As is clear from its name, it is based on frequencies in a long run. The result of the frequentist approach is a probability value (P) or a confidence interval (CI). The confidence interval is the range within which the unknown parameter lies with, e.g., 95% confidence (Carlin and Louis, 2000).

According to Held and Bove (2014) the probability associated with observed data, in frequentist analysis, is treated as random, while that associated with an unknown parameter is not treated as random.

The difference between data inference using MLE and the frequentist approach is that the latter takes all the values that are potentially expected to be true (Held and Bove, 2014), while the MLE is only giving one value that is potentially true.

4.4.3 Bayesian Statistics

Bayesian statistics, based on using Bayes' rule (which is illustrated in the next section), is an approach linking relevant prior available information to new available data, provided by observations, to infer the required parameters of interest. Likelihood provides the tool to incorporate data update before determining *a posteriori* values. This linking led to the Bayesian statistical approach becoming widespread among researchers (Anscombe, 1962). For example, the Bayesian statistical approach has emerged as a member of the machine learning family of techniques, and it has had a great impact on it (Kobayashi et al., 2012). Bayesian methods depend on prior information to infer new information, including new data from observations, encouraging the application of Bayesian approaches, over frequentist, in much research (O'Hagan, 2004).

For the incorporation of a Bayesian approach to generate an output parameter's *a posteriori* probability distribution, it is necessary to make assumptions of *a priori* probability distributions and likelihood, using the sample datasets. By analysing all the *a posteriori* parameters the user may be provided with more accurate details about the parameter of interest, than taking only the mean, mode, median or standard deviation. For example the *a posteriori* probability distribution can be used to estimate the parameters themselves; in Bayesian inference data are provided to enable the probability of the parameters to be determined, in order to achieve a solution (Stauffer, 2008). Or in other words, in the Bayesian approach, probabilities are fitted to all the information which has been presented in a particular situation (Gelman et al., 2004). The main challenge in using the Bayesian approach, according to FDA (2010), is the need for extensive understanding of the situation. It requires a thorough assessment at each stage, of the prior information and the information to be inferred from the observations, or of the model which is mathematically constructed to combine the two.

4.5 Bayes' Rule

Bayes' Rule, Bayes' Theorem or Bayes' Law, as it is variously named, can be used to define the Bayesian approach. The Bayesian approach involves determining the *a*

posteriori probability $f(\theta|X)$, by combining information on the data to hand using a model that is typically represented by a form of probability distribution $f(X|\theta)$. This probability distribution, $f(X|\theta)$, is the *likelihood* and represents the measured data $X = x_1, x_2, \dots, x_n$ given the vector of required parameters θ , and values based on θ expressed through *the a priori probability* and the *normalising constant* (or sometimes called *marginal likelihood* (Gelman et al., 2004) or evidence (Kruschke, 2011)).

The inferred outcome is obtained from the *a posteriori* probability distribution, focussed on the parameters, θ .

Specifically Bayes' rule stated:

$$a \text{ posteriori probabaility} = \frac{a \text{ priori probability} * \text{likelihood}}{\text{marginal likelihood}} \quad 4-5$$

Bayes' rule creates a joint conditional probability distribution for x and θ . The joint distribution used is the probability distribution function (pdf) or merely the distribution function $f(x)$ of a random variable; it is the function that produces $p(X \in R)$ with respect to the region R within the limit X when it is summed, and X is the function of the value of x under the condition that $f(x) > 0$ (Link and Barker, 2010).

Generally the joint probability distribution is created by the product of two distributions, here utilized by the data model-likelihood $p(x|\theta)$ and the *a priori* probability distribution $p(\theta)$:

$$p(\theta, x) = p(\theta)p(x|\theta) \quad 4-6$$

Taking θ conditioned on the observations and using the fundamental property of conditional probability, as recognised by Bayes' rule, leads to an *a posteriori* probability distribution (Carlin and Louis, 2000). According to Link and Barker (2010) the basis of Bayes' theorem is the relationship between the joint and conditional probabilities:

$$p(\theta|x) = \frac{p(\theta, x)}{p(x)} = \frac{p(\theta)p(x|\theta)}{p(x)} \quad 4-7$$

where $p(x)$ is called the normalization factor which is related to all possible values.

This can be calculated by summing all possible values $p(x) = \sum_{\theta} p(\theta)p(x|\theta)$ in the case of a discrete probability value. In the case of a continuous variable, the value will be $p(x) = \int p(\theta)p(x|\theta)d\theta$ using all possible values.

The normalization factor is used when it is required to know the absolute value of the probability, but if the situation only requires estimating the maximum or minimum relative probability it is not necessary to determine the normalization factor.

4.6 Bayesian Inference

One of the applications of Bayesian statistics is Bayesian inference. It can be used to infer information of interest from observations providing data concerning undetected quantities. Bayesian inference is used to obtain estimations of a parameter or unobserved data; usually probability statements are used to infer the estimations. The values that are used in Bayesian approaches, are random variables and provide uncertainties as output, represented as probability expressions. Bayesian inference is a statistical inference method where observations are utilized to determine the probability that assumptions are likely to be true, or to revise an already determined probability. In the applied field, Bayesian inference uses an *a priori* probability calculated from the likelihood of certain assumptions regarding the observations imported into a computation or process.

Bayesian inference is represented by constructing a model that adequately represents the situation from which information is to be inferred. The constructed model is based on Bayes' rule in which the results are represented by probability and called the *a posteriori* probability distribution. The input components of Bayesian inference are *a priori* probability and likelihood. The *a priori* probability represents the subjective belief regarding the situation before seeing any data; while the likelihood is the model that represents the data by fitting a probability distribution on it (Gelman et al., 2004).

Bayesian inference handles information generation from an appropriate model. The implementation begins by employing a particular belief about the unknown parameter that is called the *a priori* information. The *a priori* information could be any information, even ambiguous, about the unknown parameter. The certainty of the *a priori* information could, later, be increased by adding more measured data from the observations. In Bayesian statistics, to make an inference about an unknown parameter, it is necessary to

determine the whole *a posteriori* probability distribution, having been provided with the *a priori* probability of the parameter. Bayesian statistics procedures are based on determining probability of the ‘trueness’ (or accuracy) of the parameters (i.e. uncertainty is evaluated in Bayesian statistics by using probability).

The unknown parameters that are of interest are represented through the model that incorporates the parameters and their probability distributions. Modelling is used to infer the shape of the underlying procedure by examining the feasibility of specified models (Myung, 2003). Bayes’ approach involves measurements that have been observed and others that can be inferred or expressed through probability models. These probability models can be considered a basic tool in the Bayesian method and lead to uncertainty information originating from statistical analyses. The basic Bayesian analysis can be summarized in three main stages as follows (Gelman et al., 2004):

- 1-Utilize all observations and parameters associated with the problem and assign a full joint probability distribution model. The constructed model should be compatible with the data gathering method and the underlying problem.
- 2- From the above constructed model derive an *a posteriori* probability distribution, based on observed measurements and analyse it.
- 3-Asses the results’ goodness of fit and check whether the outcome represents the real facts and the resulting model corresponds to the data used. Check the result’s sensitivity to the assumed model in stage 1; sometimes it is necessary to amend or develop the model and re-evaluate (i.e. repeat) these three stages.

The above three stages are the core of Bayesian inference and estimation. However, the first stage is considered the main stage, and the main challenge in Bayesian analysis is how to produce a model and an *a priori* probability that fits the data.

It has been mentioned that it is possible to make a connection between probability and observations; also, it is possible to represent numerically beliefs regarding a situation. The expressed belief can be updated with new data by employing Bayes’ rule, which is considered to be delivering a sensible connection between beliefs and observations. On

the other hand Bayesian inference is considered as a means for using specific sets of facts to infer general knowledge (Hoff, 2009). Bayesian inference (or the Bayesian method) used in data analysis, offers the following advantages:

- it accompanies the estimation of the output parameter with useful statistical information;
- it provides superior statistical properties for the estimated parameter;
- it minimizes the number of measurements that need be used in the experiment to infer the result;
- it makes allowances for lost data;
- it estimates future data; and finally,
- it provides an environment for model prediction, choice and validation (Hoff, 2009).

The Bayes formula that is used in inference, for the continuous type of probability distribution, is of the form:

$$p(\theta|x) = \frac{p(x, \theta)}{p(x)} = \frac{p(\theta)p(x|\theta)}{\int p(\theta)p(x|\theta)d\theta} \quad 4-8$$

The term $\int p(\theta)p(x|\theta)d\theta = \int p(x, \theta) d\theta = p(x)$, is sometimes referred to as the marginal density distribution of the random variable X or its *a priori* predictive distribution (Boos and Stefanski, 2013). It is clear that this function is used to normalize the *a posteriori* distribution. It is obvious that θ need not be taken into account to find the final value of the marginal distribution $f(x)$. The marginal distribution is used to show the prediction based on random variable X , by considering the uncertainty of the random variables of the initial data, and the residual uncertainty of the random variable when θ is calculated. This marginal distribution is significant in model selection.

The main characteristics of Bayesian inference that made it a focal point for researchers are: the straightforward quantification of a measured parameter's uncertainty using probability; the removal of any obstacles to the number of parameters addressed; and, the applicability of joint probability density functions (Gelman et al., 2004).

4.7 Bayesian Parameters

A parameter can be considered to be a governor, or adjustor, of the influence input has on the resulting numerical probability that arises from the implementation of a model. It is

very rare for only one parameter to be considered. Most of the problems to which the Bayesian approach can be applied address more than one parameter. However, when the model consists of many unknown parameters it is not necessary to calculate all of them. Usually inferring only one (or just a few) of them will be enough. In all cases, the procedure is to calculate the joint *a posteriori* probability distribution of all the parameters that require to be determined and to assign a marginal distribution to the parameters that need not be calculated. A marginal distribution can be obtained by integration with respect to any unnecessary parameters' values. Parameters, not necessarily of direct interest to the situation, are often called nuisance parameters, such as standard deviation σ .

Gelman et al. (2004) explained the above as follows. Consider there is one unknown parameter θ composed from two unknown parameters, thus $\theta = (\theta_1, \theta_2)$. For the work which is the subject of the research reported in this thesis, let, θ_1 be μ (the average), and another parameter θ_2 , also called a 'nuisance' parameter, be σ (standard deviation). This can be stated as:

$$x|\mu, \sigma \quad 4-9$$

In the case of assigning normal distribution to the form of expression given in equation 4-9 (i.e. $x|\mu, \sigma$), in Chapter 5 $N(\mu, \sigma^2)$ is used.

The aim is to find $P(\theta_1|x)$, which is defined as the conditional probability distribution based on the existing data. The procedure for calculating the value of $P(\theta_1|x)$ involves:

calculating the joint *a posteriori* probability;

$$p(\theta_1, \theta_2|x) \propto p(x|\theta_1, \theta_2)p(\theta_1, \theta_2) \quad 4-10$$

averaging with respect to θ_2 ;

$$p(\theta_1|x) = \int p(\theta_1, \theta_2|x) d\theta_2 \quad 4-11$$

Substituting the averaged over θ_2 in the joint posterior probability distribution will produce;

$$p(\theta_1|x) = \int p(\theta_1|\theta_2, x) p(\theta_2|x) d\theta_2 \quad 4-12$$

Thus it can be shown that the parameter of interest, μ , which is represented above by θ_1 , is reflected in a conditional distribution function using the second parameter θ_2 , and it is obtained by synthesis of the conditional distributions, given θ_2 , which is a so-called nuisance parameter in this context.

4.8 *A priori* Data

The uncertainty of an unknown parameter is represented in the *a priori* data; for this it is possible to use any available precedent data about the unknown parameter (Boos and Stefanski, 2013). *A priori* data plays an important role in Bayesian inference, and it has a particularly large influence on a *posteriori* data when small data sets have been utilized. Therefore the *a priori* probability must be chosen judiciously, and the results (the *a posteriori* probability distribution) should be evaluated carefully for goodness of fit to estimate the dependability of the model's predictions (Stauffer, 2008).

According to Ang & Tang (1975) the parameters of the *a priori* probability distribution are defined by logic, experience or empirical considerations. Typically, the parameters of the *a priori* data are based on real practical information or personal belief about the situation. The *a priori* data should be selected carefully to convey the original situation because different outcomes result from using different *a priori* data. The *a priori* data become less doubtful if it is based on experimental data or on concrete assumptions. It may become suspect if it is based on personal conclusions (FDA, 2010). It is also possible to use a Bayesian approach without depending on *a priori* data (this is called the 'uniform prior' situation), by getting a temporary result, until it adjusts itself based on the arrival of *a priori* data (FDA, 2010)

Until now, the most serious obstacle to using Bayesian methods has been finding suitable *a priori* data to represent the situation. Although the *a priori* data are based on previous data or on a professional's subjective opinion, it is possible to simplify the selection of the *a priori* values by limiting them to a well-known distribution, to improve the data inference process and make the calculation task easier. It is possible to produce *a priori* data based on a few instructive observations, consequently this will help to make the study's input data the main source for producing the eventual *a posteriori* values (Carlin

and Louis, 1996).

Hoff (2009) has pointed out that the Bayes method does not help to specify any information about the *a priori* data, however it has the ability to modify the *a priori* data used based on the observed data.

Some important types and characteristics of *a priori* data are addressed in the following paragraphs.

4.8.1 Subjectivity in *a priori* Data

The Bayesian approach is considered to be subjective because it is based on assumed *a priori* data and that *a priori* data depends on personal evaluation, which varies from one person to another (Berry, 1997). In all cases, the *a priori* probability should reflect the problem. Researchers are always trying to minimize the disparity between the *a priori* probability and reality by invoking more evidence. Variance in the *a posteriori* probability distribution can be minimized by strengthening the model with more data.

The potential capability of Bayesian inference is embedded in quantifying all uncertainty using probability. This can be achieved by assigning *a priori* probability to each unknown parameter. According to Link and Barker (2010) the main aspect hampering the wider use of Bayesian inference is the subjectivity involved in deciding the *a priori* data.

4.8.2 Uniform *a priori* Data

If there is no information about the data that has been used in the problem, which means no prior knowledge, then, a value between the minimum and maximum, which is between 0 and 1, will be used. As shown in Figure 4.2, this gives the same properties to all values of the parameter, which in that case is equal to $P(\theta) = 1$. This type of prior information is useful when the problem uses very reliable measurement data. In that situation, *a priori* probability data has no significant effect and it is better to use ***uniform a priori probability*** or ***non-informative a priori probability*** instead of a conventionally constructed *a priori* probability. If the *a priori* probability that is used in the model is non-informative then it will have a very negligible effect on the *a posteriori* probability, and in that case all the results will be based on available data from the observations,

usually called *the likelihood*. The distinct method that is used to define the uniform *a priori* probability is called *Jeffrey's' prior probability* and can be adapted for multi-parameter situations although it is very commonly used with single parameter models. A uniform *a priori* probability can be used successfully in Bayesian approaches, but problems may arise when hypothesis testing or identifying the model.

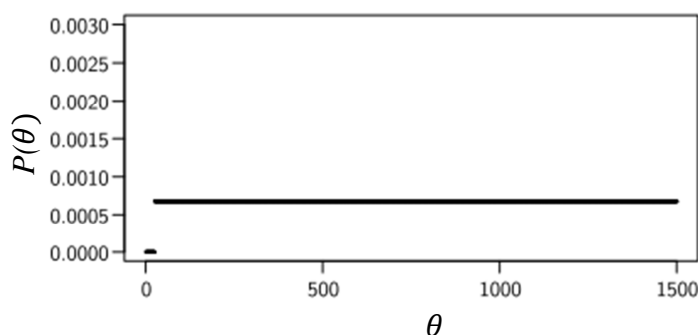


Figure 4.2 Uniform *a priori* probability distribution, for the value of θ for interval [26,1500](Held and Bove, 2014).

4.8.3 Conjugate Prior

Sometimes both the *a priori* probability distribution and the *a posteriori* probability distribution belong to the same family type (e.g. Gaussian, exponential, Poisson, etc.) and in that case, the *a priori* probability is referred to as **conjugate prior probability**. A *a posteriori* probability determination will be simpler when using a conjugate prior probability; it makes calculation easier. However, if the conjugate prior probability does not properly reflect the main *a priori* data, particularly if any new data arrives, then it is necessary to use a more genuine *a priori* probability. In this case it is better to use a ‘conjugate **prior probability**’ and, then, after that amend it to suit the situation (Held and Bove, 2014).

4.8.4 Improper Prior Distribution

A priori probability has a significant effect on the *a posteriori* probability distribution and thus on the final result. In order to minimize this effect it is common to use a “vague” or “improper” *a priori* distribution (represented by adding the proportional sign \propto and referred to as an improper prior); this prior could, for instance, be specified with a large variance (Held and Bove, 2014). Consequently, it is necessary to check that the *a posteriori* distribution is a proper distribution, and if so, then it is possible to use the improper prior in the Bayesian analysis.

4.9 Posterior Distribution

After data sampling, using the likelihood function (represented as a random variable and denoted by the density function $p(x|\theta)$) and deciding on the *a priori* probability, the next stage is to obtain the *a posteriori* probability by normalizing the product of the *a priori* probability and the likelihood. The *a posteriori* probability distribution so obtained is represented by the conditional probability distribution of θ given the random variable X (Boos and Stefanski, 2013):

$$P(\theta|x) = \frac{p(x|\theta)p(\theta)}{\int_{\theta} p(x|\theta)p(\theta) d\theta} = \frac{p(x|\theta)p(\theta)}{p(x)} \quad 4-13$$

The most significant output in Bayesian inference is the *a posteriori* probability distribution that incorporates all information about the unknown parameters using Bayes' rule. It is possible to use it in order to obtain point inference and interval estimates in addition to all other necessary information about any unknown parameter (Boos and Stefanski, 2013; Gelman et al., 2004; Hoff, 2009; Kobayashi et al., 2012)

It can be noted that $p(x)$ does not depend on θ , therefore the equation 4-13 can be simplified to produce an *a posteriori* distribution, by eliminating the fixed term $p(x)$, thus the equation for the *a posteriori* probability can be:

$$p(\theta|x) \propto p(\theta)p(x|\theta) \quad 4-14$$

or

$$Posteriori \propto Priori . Likelihood \quad 4-15$$

A posteriori probability $p(\theta|x)$ is proportional to the equation 4-14 and summarizes the substance of Bayesian theory. It can be noticed that the *a posteriori* probability distribution $p(\theta|x)$ is proportional, not equal, to the *a priori* $p(\theta)$ and likelihood model $p(x|\theta)$. The *a posteriori* probability distribution function represents the outcome of Bayesian inference. To give the full picture of the situation it may be useful to represent *a posteriori* distribution graphically, as shown in Figure 4.3.

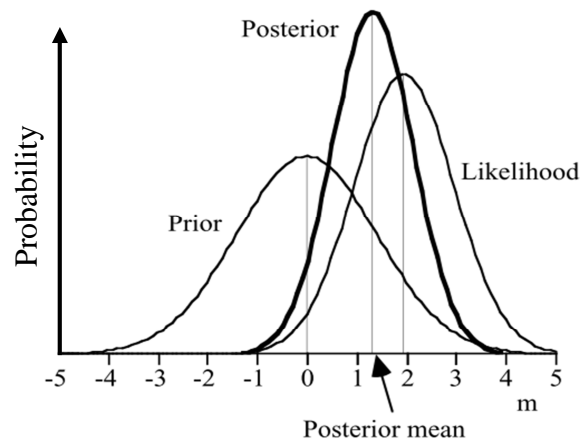


Figure 4.3 The components and result of Bayesian statistics, *a posteriori* (or Posterior), likelihood and *a priori* (or Prior) distributions (Berry, 1997).

However, a representation of the *a posteriori* probability distribution by numerical value is necessary for numerous practical implementations.

4.10 Data Inference

The final stage of the Bayesian inference approach is to infer information, such as an estimated parameter, from the *a posteriori* probability. Model selection and parameter estimation are considered to be two of the most important problems that researchers face in data analysis (Punska, 1999).

The most used statistics extracted from the summaries of the *a posteriori* distribution (referred to as point summaries) are: mean, mode; median; and, standard deviation. The mean and standard deviation play an extremely important role in deciding the final results of the original unknown parameters.

In Bayesian statistics, according to Gelman et al. (2004), there is relation between *a posteriori* and both *a priori* and likelihood. The average of all *a posterior* probabilities over the distribution of all data will lead to the mean *a priori* parameter that is known by θ . This can be shown in equation 4-16:

$$E(\theta) = E(E(\theta|x)) \quad 4-16$$

Variance is also important and shows how uncertainty can be reduced. The equation 4-17, manifests that relation and it shows that the variance of the derived θ is

less than the variance of the *a priori* θ .

$$Var(\theta) = E(Var(\theta|x)) + Var(E(\theta|x)) \quad 4-17$$

The variance in equation 4-17 is less than the variance of the original data represented by the *a priori* distribution due to incorporating the data. According to Gelman et al. (2004) the *a posteriori* distribution has greater impact on reducing the uncertainty when the original variance is large rather than small.

The main outcome of the Bayesian approach is reflected by sampling the *a posteriori* distribution. The procedure that is followed in the Bayesian approach for parameter estimation involves manipulating the parameter θ as a random variable accompanied by its *a priori* probability $p(\theta)$. The aim, in the Bayesian context, is to evaluate the *a posteriori* probability distribution, using *a posteriori* inference, by finding the *a posteriori* value of the parameter from the *a priori* data. To commence one starts by evaluating the conditional probability $p(\theta|D)$, by implementing Bayes rule:

$$P(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)} \quad 4-18$$

As has been mentioned earlier, $p(x)$ does not relate to the random variable θ , which means it is independent and it can be omitted in the proportionality function. The equation 4-14 shows that the *a posteriori* probability is specified by the multiplication of the likelihood function and the *a priori* probability.

In Bayesian analysis, assessing the maximum *a posteriori* (MAP) parameter values is considered a helpful means of relating model to data. The *a posteriori* distributions' mode can be evaluated through:

$$\begin{aligned} \theta_{MAP} &= \arg \max_{\theta} p(\theta|x) \\ &= \arg \max_{\theta} \frac{p(x|\theta)p(\theta)}{p(x)} = \arg \max_{\theta} p(x|\theta)p(\theta) \end{aligned} \quad 4-19$$

The evaluation of MAP and MLE are similar, with the difference being in the handling of the *a priori* probability. In the calculation during estimating the parameters, MAP links to

the *a priori* data that influences the result of the parameter estimation. In the work reported in this thesis the result, using MAP, is achieved by representing the models' parameter values with group measurements, which are a set of angles and distances.

The result of the MAP is very helpful, practical and clear, but the main negative aspect of using MAP, is in calculating only one mode if and when the *a posteriori* is multimode, as shown in the Figure 4.4 below, and this is then detrimentally neglected. Sometimes, when there is a model with many parameters and a high *a posteriori* probability results as a consequence of assigning a high value to one of the parameters, MAP does not give adequate information about the correlation of the parameters as is necessary in the Bayesian approach. Also sometimes, there is more than one outcome, rather than only a mode value, to be inferred from an *a posteriori* distribution, for example average, variance, marginal distribution etc. In that case, a statistical technique such as Markov Chain Monte Carlo (MCMC) estimation could be used to sample from the *a posteriori* distribution.

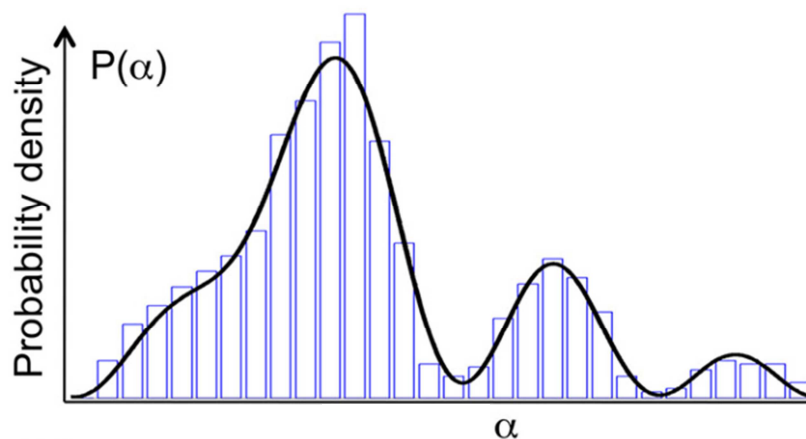


Figure 4.4 Multi mode *a posteriori* probability (from Bernacchia, 2014).

Bayesian inference involves determining the *a posteriori* probability distribution to achieve point estimation, interval estimation and model comparison, as summarised in the following paragraphs.

Point estimation. If the *a posteriori* distribution representing a univariate case in a continuous probability distribution function forms around θ , then to infer the result from the *a posteriori* data will be achieved by selecting the mode, the average or the median. Usually the mode is the simplest mean to be considered; for point estimation in a symmetric *a posteriori* distribution, the average and the median are matching values.

The mean usually represents the expectation (estimated value) of the parameter θ , with the most important data from the *a posteriori* distribution being the average and standard deviation along with the parameter θ , itself. However, the mode is important, especially in the case of complex problems, where it is difficult to calculate the average and the median. The probability density function (pdf) which is shown in the graph below, in Figure 4.5, illustrates an example of the value of the average from an *a posteriori* distribution.

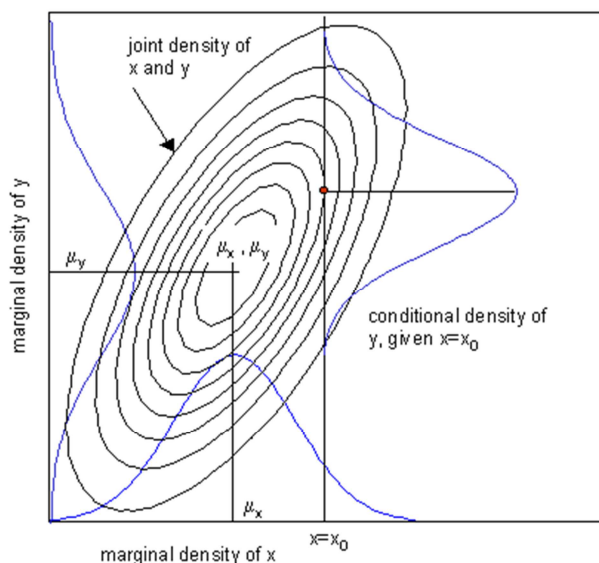


Figure 4.5 the *a posteriori* distribution showing the conditional distribution and its relation to the joint marginal distribution (<http://www.statisticalengineering.com/>).

Interval estimation. For an *a posteriori* interval, in addition to the point estimation, it is necessary to find out the uncertainty of the *a posteriori* data. This estimation calculates the value of the extent of the parameter θ . In recent studies involving Bayesian statistics, attention has turned to determining the probability interval estimation for the unknown quantity, consequently this gives further strength to the Bayesian perspective (Gelman et al., 2004). This limit of the value is also called the Bayesian confidence interval or *a posteriori* interval. When dealing with a univariate parameter it is called the confidence interval, and it is used to measure the uncertainty of the *a posteriori* distribution. This value is calculated by an interval representing 95% of the area between the ends of the *a posteriori* distribution (a, b), see Figure 4.6, for example $\Pr(\theta < a) = 0.025$ and $\Pr(\theta > b) = 0.025$, which is considered a MLE when a uniform *a priori* value is used in the Bayesian theorem. The property of 95%, which is also called Highest Density Interval (HDI), represents the values of the parameters that is more believed to be true than the points outside that limit.

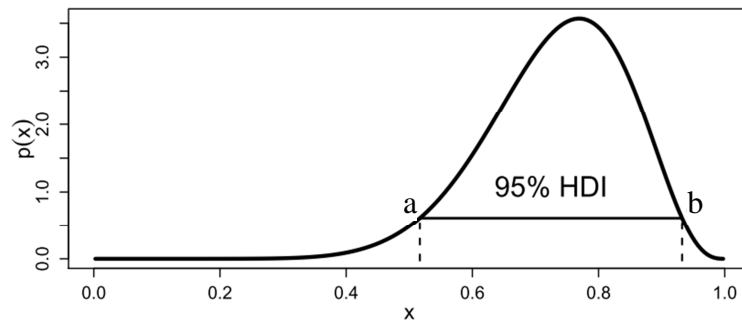


Figure 4.6 Illustration of Highest density interval (HDI) for measuring the parameter uncertainty (Kruschke, 2011).

Model Selection. This is also called model comparison, and is based on the selection of the most appropriate model, given the data. Model selection, is reflected in the *a posteriori* probability. However, the result does not give any information, directly, about the model and whether the model is correct or not; for that reason it is necessary to use a method that tests the model. Model testing can be achieved by expanding the range of the model and then later selecting the model that is the correct (or ‘best’) one. However, this method penalizes the models that contain numerous parameters. To perform model selection the *a posteriori* probability for each model is calculated based on equation 4-13 or equation 4-20 in case of discrete random variables:

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{\sum p(x|\theta)p(\theta)} \quad 4-20$$

In order to start model selection, and then to decide which of them is better, first the *a posteriori* probability for each model is determined.

Then, selection of the best model, between two models, for the calculation of the parameter(s) θ , can be based on the result of equation 4-21, to determine the posterior ratio. If the result of the posterior ratio, is bigger than **1** by an extreme value then the selection is based on the first model, $p(\theta_1|x)$. If the *posterior ratio* is much less than one, then the selection will be based on the second model, $p(\theta_2|x)$. If the result is unity, this probably means the data is not adequate to make the selection (Sivia, 1996).

$$\text{posterior ratio} = \frac{p(\theta_1|x)}{p(\theta_2|x)} = \frac{p(x|\theta_1)}{p(x|\theta_2)} \times \frac{p(\theta_1)}{p(\theta_2)} \quad 4-21$$

4.11 Applying a Bayesian approach

Bayesian techniques have been applied in two of the mathematical models implemented in the work reported on in this thesis: DSM merging and building footprint optimization. Regarding the merging, the implemented prior has been obtained using local entropy by assuming the building and ground surface in urban areas are smooth and not rough. Meanwhile, the prior that has been used in the regularization of a building footprint is based on the concept that the corner of building is orthogonal and edges are straight.

4.11.1 Bayesian Fusion Approach

A Bayesian approach has been successfully applied, by others, to improving the fusing of images, as can be seen from the various perspectives of improving a data set, reducing the size of a data set or enhancing it for purposes of investigation. Image fusion has been used in miscellaneous fields such as medicine, airport security, remote sensing and computer vision. Gheta *et al.* (2008) used a Bayesian approach to fuse images for the purpose of getting better depth information (i.e. 3D data). A Bayesian approach, incorporated with a Markov Chain Monte Carlo (MCMC) method, has been implemented to fuse MRI (Magnetic Resonance Imaging) and CT (Computed Tomography) data for medical images and for airport security checks, by improving segmentation (Feron and Mohammad-Djafari, 2004; Mohammad-Djafari, 2003; Punska, 1999; Shi and Manduchi, 2003). Sharma *et al.* (2001) had more robust results when using a Bayesian approach in the fusing of synthetic infrared and radar images for the purpose of enhanced airport landings. Jones *et al.* (2003) fused images of low resolution thermal data and high resolution visible range data to get better segmentation in a scene of interest. Bayesian approaches have been applied in the remote sensing field also, for example Zhang *et al.* (2008) and Zhang (2011) used Bayesian approaches to combine multispectral and hyperspectral images. Bayesian approaches were also applied to multispectral images to improve several principle components (Ge *et al.*, 2007; Mascarenhas *et al.*, 1992; Zaniboni and Mascarenhas, 1998; Zhang, 2010b). Bayesian approaches have been used to get an enhanced image for better visualization and to combine heterogeneous images in order to get one image with more features, or just to incorporate new incoming images (Mascarenhas *et al.*, 1996) or in the case of astronomical images to fuse either one-band astronomical images (Jalobeanu *et al.*, 2008) or hyperspectral images (Petremand *et al.*, 2012). A study by Sommer *et al.* (2009) showed the ability of the Bayesian approach to evaluate the uncertainty of fused data. It

has also been used to fuse panchromatic and multispectral images to obtain pansharpened images (Fasbender et al., 2008). A Bayesian approach can be used to achieve better image properties because it provides sharper and clearer edges on the image (Kotwal and Chaudhuri, 2013).

According to the literature, Bayesian approaches have been applied in remote sensing successfully, but no study has been found relating to their use in merging or fusing digital surface models. The common approach to fusing DSMs is based on using a weighted average, after assigning weights to the DSMs' points based on checking their fidelity or performing a "DSM accuracy assessment" by calculating some statistical measurements (i.e. RMSE).

The Bayesian approach utilizes all *a priori* and test data, and it does not eliminate or delete any *a priori* information. In the Bayesian approach, uncertainty is handled by employing *a priori* probability, consequently, it can be utilized for getting more accurate solutions in a decision making situation (O'Hagan, 2004). The *a priori* probability information in this study can be based on one of the original DSMs (e.g. that produced from the WorldView-1 sensor), and its uncertainty statistics.

It should be noted that the literature uses alternate words for the merging of Digital elevation models, such as fusion, combination, integration and synergy. These all can be considered to be a synonym for merging (Papasaika-Hanusch, 2012).

4.12 Comparison between Bayesian and Frequentist Approaches

According to a report by Berry (1997), a number of differences between Bayesian and frequentist approaches can be shown.

- Parameter probabilities. All parameters of interest are, in the Bayesian approach, fitted with a probability distribution, thus the Bayesian approach relates probability to the input parameters while the frequentist approach does not behave in the same way, and assigns probability just to the data that is produced from the experiment.

- Bayesian approaches can consider wide ranging facets of a process with the result being summarized by the *a posteriori* probability. The resulting probability distribution takes into account all the available facts regarding the unknown parameters, while the frequentist approach is just based on the data from a specific set of observations. This feature renders a Bayesian approach, potentially, more useful although it does meanwhile lead to a need for more judgments and quantification of the facts surrounding a situation and its assessment. Potentially this is the best approach for computing uncertainty and consequently it makes the Bayesian approach an important tool for decision making.
- Taking genuine observed outcomes, Bayesian inference is based on utilizing the outcomes from existing analyses only, whereas the frequentist approach is based on utilizing the probability that is assigned to the data.
- Flexibility. The most important characteristic of the Bayesian approach is that it accepts new data as the analytical process continues, which leads to a continuous update of the *a posteriori* probability, thus it is not necessary to specify the sample being analysed in advance. This makes the Bayesian approach very applicable to situations where it is not possible to use a frequentist approach, because the frequentist approach is only executed when all relevant data are completely available.
- Randomization. This is implemented in order to minimize the bias, and to produce balance between covariates (e.g. known and unknown variables). Randomization is considered to be the core in frequentist analysis, but is not required in the Bayesian approach.
- Predictive probability. The Bayesian approach has the ability to calculate the probability for unmeasured information. This feature is available in the frequentist approach as well, but in terms of conditions on specific parameters.
- Decision making. The Bayesian approach is considered very useful for establishing a decision for a particular situation or problem, because there is cost

and benefit associated with each problem and the Bayesian approach can support the evaluation of these costs and benefits for each likely future measurement event. In the frequentist approach this feature is not available, consequently this makes it weaker in decision making applications. In addition to that, the Bayesian approach provides more information through *a priori* probabilities which augments the available information (FDA, 2010).

4.13 Summary

The Bayesian model differs from the classical frequentist method, in that the unknown model parameters are random variables, while in the classical approach they are fixed constants. The Bayesian approach can give robust results compared to the classical method. Questions may arise concerning the veracity of the used *a priori* values since they may be based on personal opinion, and consequently judgements of subjectivity. The result of the Bayesian approach is totally affected by the implemented prior. Thus, results are effective if based on information correctly reflecting the situation in the problem. The Bayesian Approach can give reliable results when the data are limited to a few samples, for example merging DSMs if only two data sets are available, or building footprint extraction limited to only one data set.

Chapter 5 Merging Digital Surface Models (DSMs) Using Bayesian Approaches

Advances in data acquisition have led to a provision of heterogeneous data sources. Notionally, using many data sources is better than using only one, but the problem is dealing with all the data simultaneously. It is necessary to use an efficient method to merge them. Merging or fusing DSMs using different techniques is an active research field, due to increasing sources of DSMs. The candidate merging technique selected for this investigation utilizes Bayesian statistics; it has been selected because it offers the means to clearly consider the uncertainties in the estimated parameters.

The aim of merging is to obtain both more informative and more accurate data than originally existed (particularly with respect to the objects of interest and the Survey and Mapping application). The new merged DSM is obtained from data arising from the combination of two or more digital elevation models.

If a sound technique is used in the merging it will provide an informative result. However, when different images are used, to produce digital elevation models, the image matching will produce different results and somewhat different objects. Perhaps the simplest method of merging DSMs is averaging and giving the same weight to all sources. Even a weighted average method may not attain the most accurate results in the output merged digital surface model, and, in order to overcome predicted imperfections in the output, an approach more soundly based in theory has been established, for consideration.

As well as considering the well-established weighted-average approach (Maximum Likelihood Estimation) that is widely used for merging and is later used for comparison (see Table 6-1), this chapter will describe the methodology for merging DSMs using a Bayesian approach. The stages that are described (see Figure 5.1) support the merging of DSMs that have been produced from processing satellite images. Two approaches to merging: Maximum Likelihood Estimation (the weighted-average approach) and using a Bayesian approach will be tested. A quality assessment of the input data was essential for both methods and was executed prior to the merging process and also an *a priori* estimation of the elevation's probability distribution had to be performed in order to proceed with the merging method using the Bayesian approach.

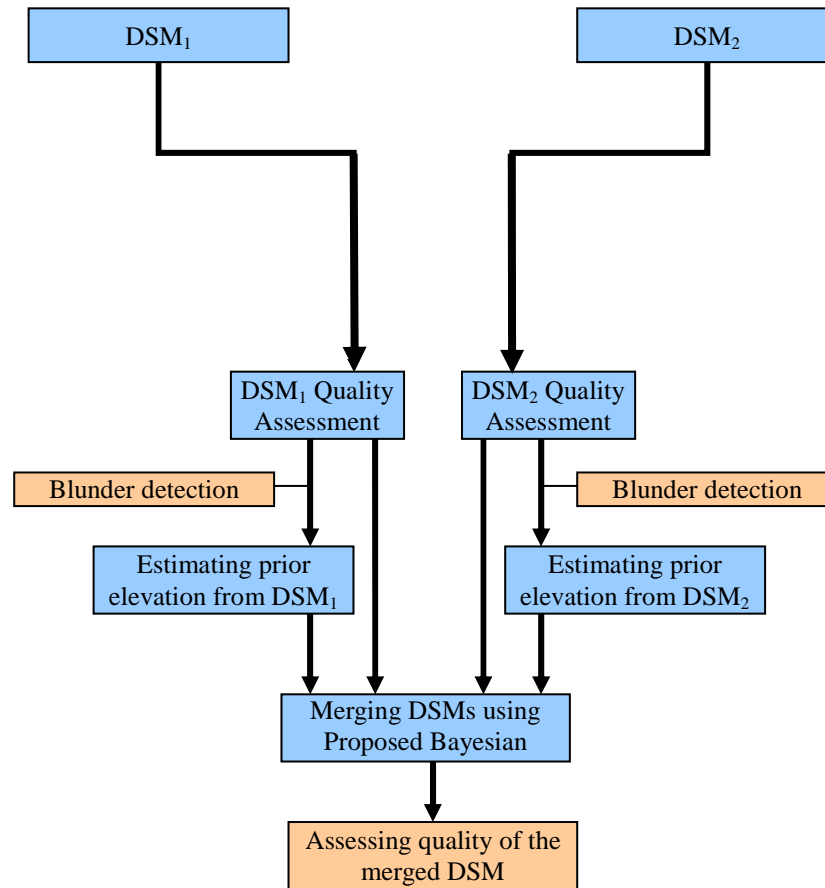


Figure 5.1 Flow chart for DSM merging process.

5.1 Test Site

The test area that is used to merge the DSMs is located in Glasgow, United Kingdom. The test area was on the Universal Transverse Mercator (UTM) projection, longitudinal zone 30 ‘North’ using the WGS84 figure of the Earth for both the vertical and horizontal datum. The area had coordinates, bottom left corner (417820 mE, 6191335 mN) and upper right (420527 mE, 6195090 mN), covering an area about 10km² that measured 2.7km by 3.75km. It is an urban area including different kinds of buildings, different types of vegetation including trees, and a river with steep banks in an otherwise gently sloping area. The elevation range varied from 49m to 152m, an elevation range of more than 100m. It was considered important to have a Glasgow location to enable a high level of local knowledge.

The study area that has been selected in this research is located in Glasgow, UK, which has high levels of cloud cover most of the year. Consequently, the number of archived satellite images for that area is very limited. Therefore, the only very high resolution satellite imagery available among that illustrated in Table 1-2 is limited to two stereo

scenes, one of them is Pleiades (Figure 5.2(a)) and the other is WorldView-1 (Figure 5.2(b)). They have been processed using Socet GXP in order to produce different DSMs (Figure 5.2(c)).

As shown in the Table 3-1, the first source was from the Worldview-1 (WV-1) sensor from the DigitalGlobe organization, dated 24 May, 2012 covering 100km^2 , while the other was from the Pleiades sensor from Astrium Geo-information services, dated 09 July, 2013, covering 25km^2 . Both products were provided by their suppliers with a resolution of 0.50m.

As shown in Table 3-1 the time gap between the satellite imagery is calculated to be around 14 months. Merged DSMs can be useful at the city-scale, thus it is normal that changes will have occurred in the area of concern over the period of the images' capture, as referred to it in 1.4. However it has been assumed, in this research, that no changes have arisen within the time period, therefore the merging in this research does not consider any multitemporal effect. This assumption has been made in order to focus on the effect of the merging using a Bayesian approach. For that reason, at the validation stage, the focus was only in an area in which no building changes had arisen during the image capturing period, which represents a gap of about 14 months. There is a seasonal gap of seven weeks, which in the west of Scotland, will result in slightly denser leaf coverage in early July than late May, so it might have been necessary to consider the multitemporal effect of increased leaf coverage during the merging process. It is suggested that prior to merging the possibility of change should be considered and change detection, perhaps arising from significant differences in vegetation cover or building development, introduced, if needed. In this research, any multitemporal effect arising from vegetation cover or building development has been neglected since the study area was for an urban area. During testing, it was noted that no building change occurred in the area, therefore no action has been taken with regard to this; otherwise, it is important to consider this factor.

Also, in this research, since the DSMs used have been produced with the same software it is assured that the resolution of the used data are identical, and the same resampling operation has been applied on the used data during the DSM generation stage. As mentioned in section 3.6.1, the techniques used in DSM generation, were based on

determining the coordinate at locations of high intensity variation or edges, only. Later resampling has been implemented to produce a grid pattern.

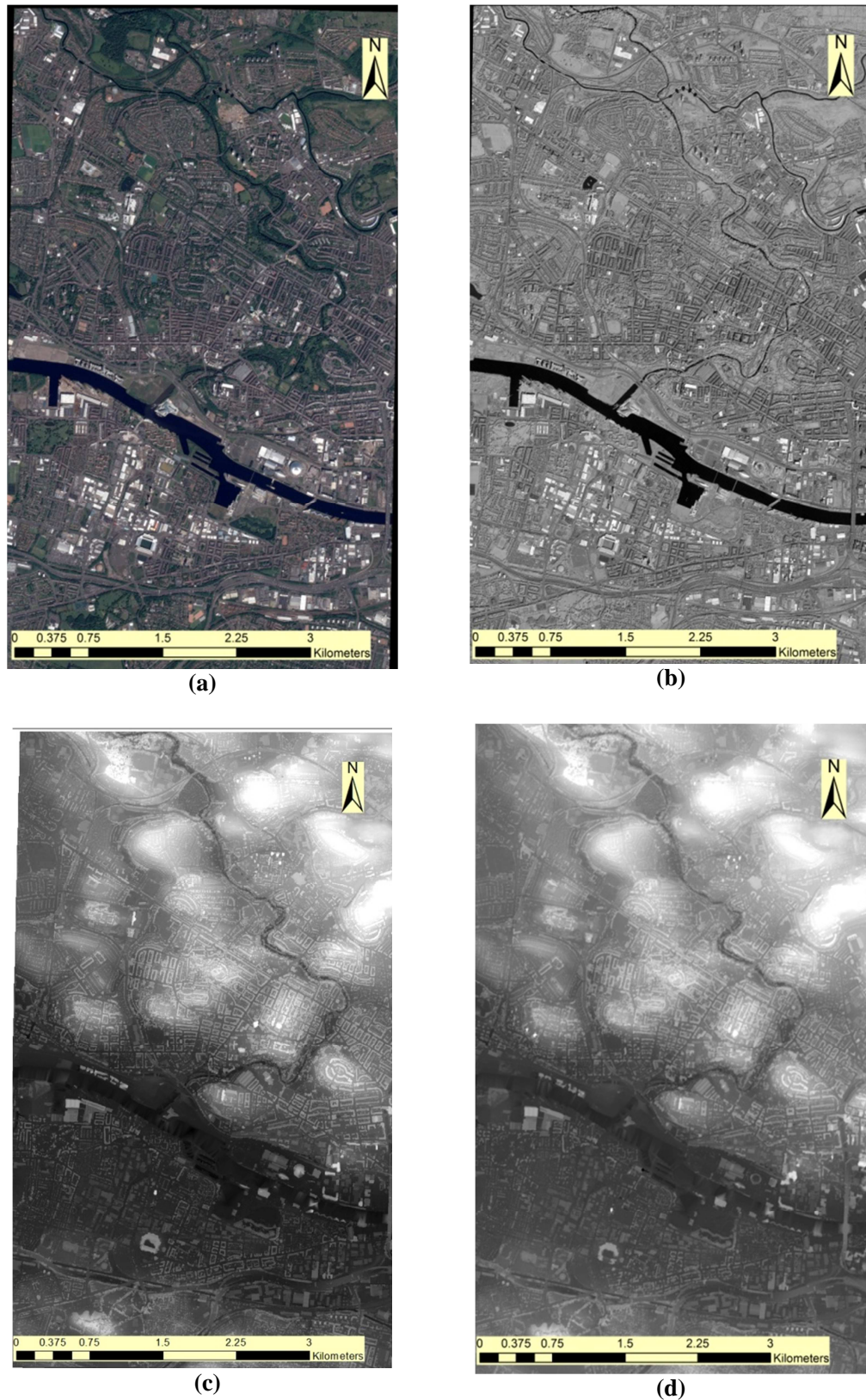


Figure 5.2 Study area and study data used in merging DSMs: (a) Pleiades satellite imagery with resolution 50cm; (b) WorldView-1 satellite imagery 50cm; (c) DSM 1m resolution produced form Pleiades stereo imager; and, (d) DSM with 1m resolution from WorldView-1 stereo satellite imagery.

The study area is specified to have varying morphology and types of built structures of different scales. In addition, land cover varied from green areas to road surfaces. The data have been processed and rectified for DSM production and orthoimagery using Socet GXP 4.1 software, with the aid of GCPs. Using SOCET-GXP, the resolution of the orthoimagery produced was 50cm, while the resolution of the DSMs was 1m. It is recommended by the SOCET GXP provider that the orthoimagery resolution be higher than the GSD of the DSM (Zhang and Smith, 2010), and the project reported in this thesis confirmed that as good practice.

5.2 Digital Surface Model (DSM) Quality Assessment

To start the merging process it is necessary to find out the quality of each digital surface model. DSM quality is an intensely researched topic commencing about 40 years ago, in 1972, led by Makarovic (Li, 1990), in The Netherlands. The quality of the DSM is based on measuring the error in DSM heights.

There are many factors affecting the accuracy of a DSM, according to Chen and Yue (2010), Li (1992) and Papasaika-Hanusch (2012), such as:

- distortion inherent in the sensor;
- the source data's attributes such as density and spread;
- surface or terrain features such as relief, land-cover, and texture;
- the mathematical approach that has been implemented to produce the DSM from the data source or the interpolation methods used; and,
- techniques used in map-digitization or field surveying.

common statistical measurements that have been used to describe the quality of the Digital Surface Models (DSMs) include the following (Papasaika-Hanusch, 2012):

Arithmetic Mean of the error

$$\mu = \frac{1}{n} \sum_{i=0}^n \Delta h_i \quad 5-1$$

Standard Deviation of unbiased error

$$\sigma = \sqrt{\frac{1}{(n-1)} \sum_{i=0}^n (\Delta h_i - \mu)^2} \quad 5-2$$

Root mean square error

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^n \Delta h_i^2} \quad 5-3$$

Mean absolute deviation

$$MAD0 = \frac{1}{n} \sum_{i=0}^n |\Delta h_i - \mu| \quad 5-4$$

Median absolute deviation

$$MAD1 = \text{median}|\Delta h_i - \mu| \quad 5-5$$

Where:

Δh_i is the difference between the checkpoint measured by GNSS and that obtained elevation from the DSM - i.e. the ‘error’ (or discrepancy);

μ is arithmetic mean of the error; and,

n is number of measurements or checkpoints.

Li (1990) evaluated the quality of the DSM experimentally using average error and RMSE, based on the difference between checkpoints and interpolated DSM points. To concur with Li, according to Chen & Yue (2010) the quality of the DSM is assessed by measuring the relation of the elevation of the produced DSM to the real elevation. The statistics that are often used to encapsulate the quality of a DSM are root mean square error (RMSE) and, sometimes, estimated standard deviation of the unbiased error; the latter may be considered when the bias is neglected in the data (Abdullah, 2013).

The accuracy of the DSM varies at each pixel and error is not distributed uniformly over all the DSM, it varies with changes of morphology and thus quality details could be given at each point (Karel and Kraus, 2006) in a DSM. However, the calculation of the quality of DSM at each pixel is considered to be complex and more challenging than determining a single RMSE value. According to Li (1988), Papasaika-Hanusch (2012), Torlegård *et al.* (1986), Wise (2000) and Yang and Hodler (2000), RMSE is considered to be the most common method to calculate the quality of the DSM.

But a problem with this single value, RMSE, is it is based on the assumption that the error in the data has no bias, which should not be always legitimate ((Li, 1993) cited in (Papasaika-Hanusch 2012)), furthermore it does not give any indication about the bias. In addition to that the RMSE is influenced by: the slope; reference data elevation distribution; and, the number of check measurements (Papasaika-Hanusch 2012).

To overcome this deficiency in the RMSE Papasaika-Hanusch (2012) suggested using other parameters that do consider outliers and biases, such as mean absolute deviation MAD0 and median absolute deviation MAD1.

The quality index that has been used foremost in this study to represent the quality of DSMs is a single RMSE value per terrain model, based on using the measured ground control points as shown in Figure 5.3. The points that have been measured are located far from buildings on the ground and on a flat surface, in order to reduce the errors arising during image matching. Some of them have been selected on the roads or walkways.

In addition to assuming the errors are distributed uniformly, it has been assumed the images are fully registered; this assumption arises from the situation where both DSMs are produced with the same grid spacing using the same software, same technique, and similar resolution satellite images. The only differences are sensor source and acquisition angle. These two factors (source and acquisition angle) cause the created DSMs to be different due to the image matching technique, thus identical features appear different in their respective DSMs.



Figure 5.3 The study area indicating the checkpoints used for quality assessment.

5.2.1 Blunders

The quality, or the uncertainty, of each DSM (i.e. produced from Pleiades and WorldView-1 imagery) is obtained from the RMSE equation 5-3. The sources of errors (whether blunders, systematic or random) in DSMs arise either during acquiring the data or during the modelling stage (i.e. interpolation). Although much research has been undertaken into how to remove blunders during DSM construction (Jancso, 2008; Milledge, 2009), still some of them exist, and in the case of this study it was necessary to detect and remove them. Before calculating the quality of the DSM, blunders, or outlier errors at checkpoints have been detected and eliminated.

The ground truth value, which is used to calculate error, was based on using checkpoints which have been obtained using differential GNSS (the points consisted of a mixture of RTK and Static GNSS points as mentioned in section 3.4), and each is referred to as ‘a true ground point’. The number of measured points is 61 points, and they have been located as well-defined points following an irregular pattern in the study area (Figure 5.3). Among those 61 points, only 30 points were used to evaluate the quality of the DSMs and the rest were left to assess and evaluate the results of merging.

The use of checkpoints to determine accuracy is well-established in Survey and Mapping, and the assumption is that both the set of checkpoints and the data being checked are blunder free, and that remaining errors are purely random, that is without outlier errors. Although all efforts should be made to avoid these, there may be blunders in the data being checked or even the checkpoints. Torlegård et al. (1986) introduced the term “unfiltered errors” for the difference between the elevation produced from photogrammetric techniques and checkpoints produced directly from measurements, whereas errors that remain after eliminating blunders, Torlegård et al. (1986) called “filtered errors”.

Ideally, “filtered errors” should be used for assessing accuracy; various approaches for eliminating outlier errors (i.e. blunders) have been proposed. An attractive approach, perhaps proposed because of its similarity to the removal of blunders when determining the precision of a set of repeat observations is the so-called “3 sigma” approach. Based on the assumption that the distribution of the errors following a Gaussian distribution, then sigma (σ) is the standard deviation of the error, thus “3 sigma” is a threshold to remove those checkpoints possibly associated with blunders, this means that 99.73 of the errors are lying within the often specified limits (e.g. ASPRS, 2015) of $\pm 3 \sigma$. In their publication, Höhle and Höhle (2009) demonstrate that such an approach lacks robustness when calculating RMSE and suggest alternatives, including median, normalized median, absolute deviation, absolute deviation and sample quantile. Nevertheless, a “3 sigma” approach in the project later reported in this thesis (see section 6.3.1) did reveal a single checkpoint associated with a high error, which when removed from the set of checkpoints produced a more encouraging RMSE value.

Although a comparison of data quality procedures is not a main focus of this research, one can concur with Hohle and Hohle, and acknowledge that removing checkpoints *post*

hoc lacks robustness. Checkpoints represent ‘the truth’; efforts should be made to ensure that checkpoints do so and contain no blunders. Furthermore, the points checked should also not be likely to included blunders. To quote the recent American Society for Photogrammetry and Remote Sensing (ASPRS) publication (ASPRS, 2015):

“Vertical checkpoints shall be established at locations that minimize interpolation errors when comparing elevations interpolated from the data set to the elevations of the checkpoints. Vertical checkpoints shall be surveyed on flat or uniformly-sloped open terrain and with slopes of 10% or less and should avoid vertical artefacts or abrupt changes in elevation.”

Considering the checkpoints used in this report, they were obtained using GNSS procedures. These procedures are susceptible to multipath errors introduced by nearby high buildings. Also, considering the points being checked, failures in image matching can arise near vertical objects, at steep slopes or where the height changes rapidly. As will be seen in Chapter 6, a ‘suspect’ checkpoint (point 20, in Table 6-1) was next to a 14m high building, and for this reason should not have been included in the checkpointing process. Not only might the GNSS observation at the point be of low quality because of multipath problems, but the image matching could have resulted in an estimated height value in the generated DSM with a low probability of being correct because of a nearby high vertical object. Based on the above ASPRS recommendations, at least, the suspect point can be dropped from the set of checkpoints, and indeed *should* have been prior to accuracy analysis.

In this research, all the errors in the DSMs, prior to merging, obtained from the satellite imagery and the measured checkpoints were within the limits defined by the threshold mentioned above. The histograms, in Figure 5.4, show the discrepancies for each original DSM against measured checkpoints. In order to estimate the distribution of the errors and to check whether they follow a normal distribution the q-q plot test has been applied. From Figure 5.4 (c and d) the q-q plots shows that the error does not deviate from the fitted line by a large amount and they are very close to the fitted line; this application of q-q plots was implemented by Bilskie and Hagen (2013) in their investigation of LiDAR DSM error over marshland and urban areas, where the close fit indicated that errors are normally distributed.

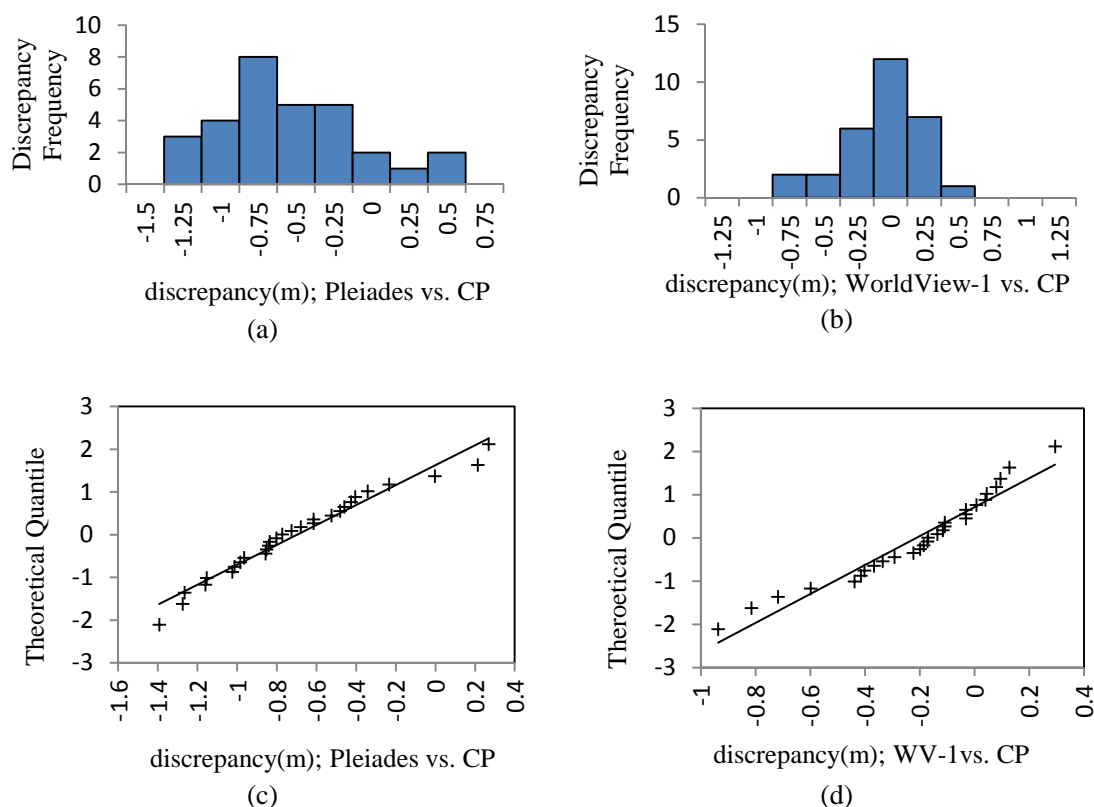


Figure 5.4 Original DSMs error distribution histograms, DSM points against checkpoints in Table 5-1 (a) Pleiades DSM against RTK checkpoints, the histogram of the error appears not to be distributed normally (b) WV-1 DSM against RTK checkpoints, the histogram of the error appears to be distributed normally (c) the q-q plot for Pleiades discrepancy against measured checkpoints (d) the q-q plot for WV-1 discrepancy against measured checkpoints.

Furthermore the Shapiro-Wilk test has been used for both datasets in order to confirm the normal distribution of the data analytically (Bilskie and Hagen, 2013; Shapiro and Wilk, 1965). The investigation determined that the Shapiro-Wilk value for the Pleiades data is 0.962 and the p-value is 0.370 and for the WV-1 is 0.944 and 0.124, respectively; it can be seen that both of the p-values are greater the 0.05. From the Shapiro-Wilk test it can be inferred that the error is normally distributed (Shapiro and Wilk, 1965).

Table 5-1 shows the ground checkpoints that have been used to evaluate the quality of the Digital elevation models from WorldView-1 and Pleiades after checking for the blunders. Although both of the images which have been used to produce the DSM have a 50 cm resolution, their panchromatic characteristics are different, as are their quality characteristics. The DSM produced from the pansharpened Pleiades images has an RMSE of 0.8m and 0% of errors in the checkpoint heights are blunders; the DSM produced from the panchromatic WorldView-1 images has an RMSE of 0.35m and 0% of

the errors in the checkpoint heights are blunders. As illustrated in the literature, see Table 1-1, the RMSE obtained is 0.6m from using satellite imagery with resolution 0.5m implementing an LSM technique, and the RMSE is 0.98 implementing an NGATE technique but using satellite imagery with resolution 1m. While the RMSE of the processing 0.5m resolution satellite imagery implementing SGM is 0.4m as shown in Table 1-1. However the result obtained from processing WV-1 satellite imagery is better, with an RMSE of 0.35m, while it is twice that with respect to the Pleiades result with an RMSE of 0.8m. The discrepancies are assumed to be due to the resolution of the satellite imagery as referred in the last paragraph of the section 8.6 and the geometry of the satellite platform as explained in 6.4.

Table 5-1 The quality characteristics of the two DSMs using the available checkpoints. A +ve discrepancy means the GNSS elevation is above the Pleiades or WV-1 elevation.

pt.	E.(m)	N.(m)	GNSS Hgt.(m)	Pleiades Hgt.(m)	WV-1 Hgt.(m)	Description
0	419984.044	6191665.674	73.269	72.848	73.067	corner's edge of walkway in public park
1	419827.489	6192039.621	75.220	75.008	75.093	corner's edge of walkway in public park
2	419157.261	6192231.164	72.249	72.592	72.154	Walkway edge in Kelvin Grove museum park
3	417861.733	6192359.230	61.611	62.447	62.330	Car park edge
4	419792.232	6192381.036	96.616	96.620	96.322	fence's corner of statue in public park
5	419374.942	6192448.093	95.256	95.780	95.658	white mark of road in car park
6	419855.809	6192494.495	87.011	87.852	87.043	white mark in football Stadium
7	419009.251	6192531.022	78.947	80.223	79.133	white mark in Car park
8	419896.604	6192772.322	68.180	68.796	68.173	white mark in Car park
9	417893.129	6193083.474	74.153	75.419	74.752	corner's edge of Walkway in public park
10	420166.325	6193155.845	81.503	82.118	81.613	Road white mark
11	419278.398	6193245.370	82.643	83.417	82.816	Texture variation in the walkway
12	418253.589	6193605.839	82.762	83.923	83.201	Public walkway corner
13	419226.770	6193655.078	72.136	73.529	72.428	Walkway's corner in the public park
14	420021.903	6193699.987	97.970	98.996	98.106	boundary's corner in Public park
15	419763.363	6193875.322	90.948	91.912	90.979	boundary's corner in Public park
16	420181.657	6194030.566	115.529	116.381	115.865	Boundary's corner in Playground
17	418920.726	6194071.048	81.444	82.302	81.668	slow down ground mark's corner in the road
18	419276.224	6194105.983	86.485	87.499	86.445	Residential area green park boundary's corner
19	419500.284	6194232.121	96.613	97.095	96.812	Pedestrian crossing white mark edge.
20	418365.912	6194296.239	104.684	105.486	104.715	Walk way corner to the schools tennis yard
21	420456.344	6194416.052	107.369	108.048	107.485	Corner of the slowdown ramp in the street
22	420270.417	6194523.306	120.904	121.364	121.074	Corner of the walkway to the entrance to the school
23	420415.829	6194545.737	112.530	113.257	112.486	White mark in the road
24	418204.263	6194556.991	111.292	111.697	111.212	Centre of the road round-about
25	420043.469	6194571.014	131.332	132.317	131.441	Pedestrian road crossing mark
26	420506.778	6194654.890	109.602	110.028	110.017	Corner of the slowdown ramp in the street
27	419833.300	6194754.782	123.470	123.705	123.837	White mark of intersection of two roads
28	418211.988	6194994.794	95.250	94.982	96.065	Corner of manhole cover on the walkway
29	418675.675	6195044.468	85.781	86.935	86.718	White mark on the ground
RMSE (m)				0.800	0.350	
number of points				30	30	
Arithmetic mean of the error (bias)(m)				-0.660	-0.198	
σ of error (m)				0.460	0.294	
Max +ve discrepancy (m)				0.421	0.294	
Max -ve discrepancy (m)				-1.393	-0.937	

5.3 DSM Formation Model

In order to achieve the merging operation using a Maximum Likelihood Estimate and a maximum *a posteriori* probability, it is necessary to establish a model that relates the DSM to the true underlying DSM. As has been mentioned earlier in section 5.2.1, the DSMs obtained from image matching or any other techniques do not typify the surface perfectly, because it has many errors (blunders, systematic, and random errors). The first two errors can be treated, however, the third one, random error, cannot. Thus, the underlying measured DSM had errors that were embedded in it.

The true DSM, which is denoted as $\overline{\text{DSM}}$, is related to the measured DSMs from the satellite image using equation 5-6.

$$Z_{(x,y)} = \bar{Z}_{(x,y)} + \varepsilon_{(x,y)} \quad 5-6$$

Where:

Z is the generated height from the DSM at location x,y ,

\bar{Z} is the true underlying height from the $\overline{\text{DSM}}$ at location x,y , and

ε is the height error in each DSM at location x,y .

Equation 5-6 shows the proposed forward model that relates the measured digital surface model that is obtained by combining the true (i.e. very accurate) digital surface model and latent error. The latter, is considered as the source of uncertainty in the DSM, determined through the inverse model of equation 5-6 (Sommer et al., 2009), which will be explained in the next section.

5.4 Merging DSMs

After constructing the model (in section 5.3), and despite its simplicity that may be considered the most challenging part of the whole process, the next stage is merging the DSMs to obtain the underlying true DSM, represented as $\overline{\text{DSM}}$. An algorithm for merging of differently sourced digital surface models is sought. Clearly, the proposed algorithm is used to reduce the number of data sources providing information related to substantial ground objects and to improve the quality of that information, and to produce more complete and reliable data.

Merging DSMs may present an ill-posed problem. According to Hadamard (1902), cited via Beyerer *et al.*(2011)) a problem is considered to be ill-posed if a solution: is not

unique; fails to exist; or the result is greatly affected by small changes in the input data. The reasons for considering the merging of DSMs as perhaps ill posed is that the results of merging are not unique (perhaps because of inherent random error). More generally, although the used DSMs are for the same area, due to occlusions, or other faults in the matching process, objects in some of the DSMs cannot be detected, which consequently will affect the merging.

The processing model used arises from considering the combination of the actual digital surface model values, represented by $\overline{\text{DSM}}$, and noise, through a simple transformation from the measured noisy digital surface model, represented by DSM , to the $\overline{\text{DSM}}$, at each point in the model. The model for generating the data, developed from equation 5-6, is shown in equation 5-7,

$$\begin{pmatrix} \text{DSM}_{1(x,y)} \\ \vdots \\ \vdots \\ \vdots \\ \text{DSM}_{k(x,y)} \end{pmatrix} = \overline{\text{DSM}}_{(x,y)} + \begin{pmatrix} \varepsilon_{1(x,y)} \\ \vdots \\ \vdots \\ \vdots \\ \varepsilon_{k(x,y)} \end{pmatrix} : k > 0 \quad 5-7$$

The above model can be inverted, to form a so-called inverse model, in order to infer the required DSM. DSM_1 to DSM_k represent the measured digital surface models, while $\overline{\text{DSM}}$ represents the underlying real or latent digital surface model values and (ε_1) to (ε_k) represents the error at each location in the measured DSMs. The calculation determines the DSM value for each DSM and the error as well at each grid location x,y . Regarding the errors, they are assumed to be random variables that are normally distributed, therefore, for each grid location's elevation in each DSM, an error can be calculated. This calculation, based on the normal distribution, requires a mean and a standard deviation. The mean will be 0 and the variance is assumed from the numeric value of the RMSE determined from the checkpoints for each DSM, and can be represented as $\varepsilon \sim N(0, \sigma^2)$; N is the representation of a Gaussian distribution with μ equal to zero and σ^2 equal to determined RMSE.

The error (RMSE) is assumed to be constant throughout the digital surface model, therefore $\varepsilon_k(i,j)$ is assumed to be constant overall the DSM which is represented by a single value (RMSE) of each DSM. The problem in this situation is to merge many noisy measurements for the purpose of determining the original DSM from these noisy measurements.

The aim of this merging is to evaluate contributing digital surface models using redundant measurements, with the expectation that this will lead to both a more accurate estimation of $\overline{\text{DSM}}$, and a more robust method for removing the error or noise. Maximum Likelihood and maximum *a posteriori* methods are used in order to get elevation values for the merged DSM. This is based on inverting the DSM formation model, see equation 5-7, to get the real digital surface model ($\overline{\text{DSM}}$).

As indicated, the merge procedures followed are based on a Maximum Likelihood approach and a Bayesian approach; the later approach requires *a priori* data, which can be any fact(s) about the situation. In the case reported on in this thesis, the fact that was used to establish the *a priori* data is based on the assumption that digital surface models of the ground represent a smooth and not a rough surface. This assumption helps to establish an *a priori* probability for the digital surface models based on local entropy; an extended explanation of this will be given in section 5.5.

The proposed approach is probabilistic, built on an assumption of noisy DSM data (Beyerer et al., 2011; Jalobeanu et al., 2008; Papasaika-Hanusch, 2012; Petremand et al., 2012; Sharma et al., 1999, 2001; Torlegård et al., 1986). The assumption that is made is that the underlying data (i.e. $\overline{\text{DSM}}$) records the true elevation which is represented by a linear function equation 5-6. The DSMs contains errors that are classified to be random which consequently result in uncertainty in the yielded data. For that reason, it is difficult to analyse problems that contain randomness or uncertainty using deterministic methods, therefore probabilistic methods appear to have an advantage in solving such a situation (Andrews and Phillips, 2003).

Thus a probabilistic (also referred to as frequentist or classical) approach providing the Maximum Likelihood for the true elevation and a Bayesian approach giving the maximum *a posteriori* probability (MAP) for the true elevation, based on solving the proposed inverse model (derived from 5-7), are both examined.

5.4.1 Merging Using Maximum Likelihood Method

A Maximum Likelihood method is considered the traditional method for estimating the results using noisy input data, in order to solve the proposed ill-posed inverse model, based on the assumption that noise is normally distributed within the data. The

Maximum Likelihood is based on maximizing the probability associated with the estimated value of a pixel in the merged DSM; that is the error between the elevations in the input pixel ($\overline{\text{DSM}}_{(x,y)}$) and the corresponding estimated pixel ($\text{DSM}_{k(x,y)}$) is required to be minimized. This can be summarized as estimating the parameters of the model, referred to as the inverse model, which has been formed for the merging task using the data that have been gathered. In summary, it is based on selecting the model's parameters in the case where the probability assigned to the measured data has produced a maximum probability. The function for estimating Maximum Likelihood, developed by Mittelhammer (2013), is represented by $L(\theta)$, and in the case that there are many sensors (i.e. two sensors), the likelihood function is:

$$L(\theta) = p(z_1, z_2 | \theta) = p(z_1 | \theta) p(z_2 | \theta) \quad 5-8$$

The probabilities $p(z_1 | \theta)$ and $p(z_2 | \theta)$ assumed to be normally distributed with mean z_1 and variance σ_1^2 and z_2 and variance σ_2^2 respectively;

$$p(z_1 | \theta) = \frac{1}{\sigma_1 \sqrt{2\pi}} e^{-\frac{(z_1 - z)^2}{2\sigma_1^2}} ; p(z_2 | \theta) = \frac{1}{\sigma_2 \sqrt{2\pi}} e^{-\frac{(z_2 - z)^2}{2\sigma_2^2}} \quad 5-9$$

The estimated value of z is obtained by Maximizing the Likelihood function Equation 5-8 with respect to z (Mittelhammer, 2013), which leads to:

$$\bar{z}_{MLE} = \underset{z}{\operatorname{argmax}} p(z_1, z_2 | \theta) = \frac{1}{\sigma_1 \sqrt{2\pi}} e^{-\frac{(z_1 - z)^2}{2\sigma_1^2}} \cdot \frac{1}{\sigma_2 \sqrt{2\pi}} e^{-\frac{(z_2 - z)^2}{2\sigma_2^2}} \quad 5-10$$

where:

e is: is the exponential function

σ is: is the standard deviation which is representing the Digital Surface Model quality

\bar{z}_{MLE} : is the value of the merged elevation using Maximum Likelihood.

Equation 5-10 yields Equation 5-11, after it is simplified and maximized with respect to z , in order to get the Maximum Likelihood estimate for z . This method requires the variances to be known for each measurement, which in this case is represented by the square value of the quality (using RMSE in this situation) of each digital surface model.

$$\bar{z}_{MLE} = \left[\frac{\frac{z_1}{\sigma_1^2} + \frac{z_2}{\sigma_2^2}}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} \right] \quad 5-11$$

Figure 5.5 illustrates the result of obtaining Maximum Likelihood from two measurements along the z axis to find the value of the \bar{z}_{MLE} (Zisserman, 2007)

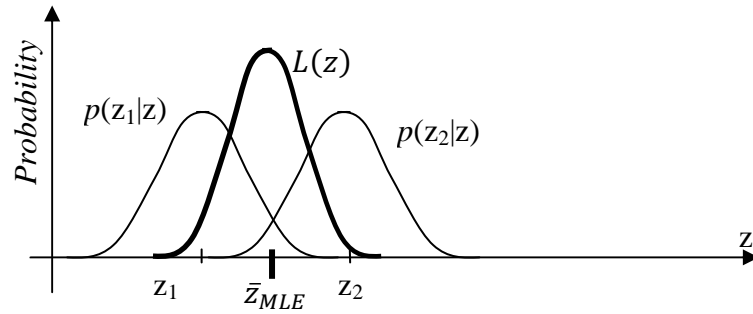


Figure 5.5 the explanation of probability parameters using Maximum Likelihood method (Zisserman, 2007).

In other words, when the likelihood function and the observations z (i.e. z_1, z_2) are given the estimated value is referred to as \bar{z}_{MLE} .

The code for the merging of the DSMs using C++ code is listed in appendix A.

The model can be extended to cover more than two sensors, which is the case if there is more than one set of measurements:

$$\bar{z}_{MLE} = \left[\frac{\frac{z_1}{\sigma_1^2} + \frac{z_2}{\sigma_2^2} + \frac{z_3}{\sigma_3^2} + \dots}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} + \frac{1}{\sigma_3^2} + \dots} \right] \quad 5-12$$

Equation 5-12 can be used for any number of sensors; this is called the weighted average; it merges the data based on the quality of each dataset, the better quality getting the more weight (or a lower standard deviation). This method is a widely used and popular method in data fusion and merging digital elevation models; it is considered to give appropriate results and is easy to use (Battiato et al., 2002; Costantini et al., 2006; Ferretti et al., 1999, 1998; Hoja and D'Angelo, 2009; Knöpfle et al., 1998; Podobnikar, 2007; Schultz et al., 1999; Torlegård et al., 1986; XU et al., 2010). The Maximum Likelihood approach has been said to offer a promising result when using it in merging satellite images (Schindler and Papasaika-Hanusch, 2011).

5.4.2 Merging Using Bayesian Approach

In the previous section 5.3 the inverse-model was defined; this relates the measured digital surface model ($DSM_{k(x,y)}$) to the actual digital surface model ($\overline{DSM}_{(x,y)}$) derived using Maximum Likelihood. The aim of merging, using a Bayesian approach, is to get an estimate of the real digital surface model by converting the inverse model. In this section the Bayesian approach is used to invert the forward model; this model is used to express the digital surface model formation, blended with uncertainty, while incorporating *a priori* knowledge about the digital surface model (i.e. its morphological properties). The most important benefit in a Bayesian approach and which does not exist in the other methods, is that the effect of noise is reduced by using a suitable *a priori* value. As already mentioned, it is assumed the errors are random and they can be represented by a Gaussian distribution with mean equal to elevation of the used DSM and variance numerically linked to the uncertainty in the digital surface model, measured by evaluating the quality, using checkpoints.

To effectively produce a merging technique that is efficient, dependable and economic, researchers have used probability models which help them to understand the situation of interest clearly (Leon-Garcia, 2008). The aim of using a Bayesian approach to merge the data is to increase the digital surface model quality and to solve problems that arise during the digital surface model production - such as occlusions and shadows, which are considered to be the main drawbacks in image matching when producing a digital surface model. Meanwhile as much topographic detail as possible should be kept in the resulting data (i.e. this can be produced by adding details from different Digital surface models into one, meanwhile minimizing the uncertainty in the produced digital surface model). The aim is to construct a new digital surface model through a probabilistic approach, incorporating *a priori* knowledge. The proposed technique is based on utilizing the image formation model. This is based on the idea that the underlying digital surface model is affected by noise (error) and a local linear function. Digital surface model merging can be considered an estimation problem and the image formation model is based on correlating the merged digital surface model with the measured digital surface model using a first order approximation.

The proposed method for merging digital surface models consists of two parts:

1. computation of the weights and *a priori* elevation for each digital surface model at each specific grid point;
2. merging the digital surface models using the Bayesian approach to get MAP.

Regarding the computation, included are:

1. quality assessment, at this stage the accuracy of each digital surface model is calculated individually;
2. *a priori* elevation calculation, the *a priori* elevation value has been calculated, and for each digital surface model the quality has been calculated individually and later has been added to the model; and finally,
3. implementation of the model, which is considered a trivial part.

To implement the model, both sets of input data should be at the same resolution (in this research this has been achieved during the digital surface model production stage; however it is possible to resample the digital surface model so they match each other), but this might well introduce additional uncertainties to the resampled Digital Surface Model due to the defects in the applied algorithm.

The uncertainty in the processed data can be considered to be of two types, either stochastic or systematic. The stochastic uncertainties are due to random processes arising from using a probabilistic model. The systematic uncertainties arise from using an imperfect model. For that reason since the Bayesian approach is using probability to evaluate the parameters, consequently for measuring uncertainty then it is possible to consider probability as an essential tool. For that reason a Bayesian approach has been considered for dealing with such uncertainties (Gelman et al., 2004). Also it is possible to use the probability distribution function to evaluate the degree of belief in the measurements (Beyerer et al., 2011).

Since merging deals with more than one digital surface model it is hoped that the uncertainty will eventually be decreased. According to Jalobeanu *et al.* (2008), a Bayesian approach can give a robust solution, in data merging, if judicious *a priori* information is employed, in contrast to a likelihood method. The traditional method for merging digital surface model is Maximum Likelihood and does not include perturbation in the computation. *A priori* information is necessary in the Bayesian approach, and the

assumption made to infer the *a priori* elevation that is used in model is based on the idea that buildings in the digital surface model specified have smooth surfaces. Based on this assumption a roughness index (i.e. local entropy) is used for maximizing local entropy. Without some assumptions it is impossible to make inference about new parameters (MacKay, 2003).

The Maximum Likelihood approach can be considered to deal with each pixel individually and does not take into consideration spatial correlation in the fused images' pixels. For that reason the DSM resulting from the fusion does not consider natural characteristics, such as smoothness, or other representations of the natural ground (Kotwal and Chaudhuri, 2013). So, in order to overcome the spatial correlation problem, it was necessary to introduce an *a priori* value which satisfactorily transformed the Maximum Likelihood value into a maximum *a posteriori* value, and transformed the problem from an ill-posed one into a well-posed one, by introducing an *a priori* value into the solution of equation 5-7, which leads to the merged digital surface model (Beyerer et al., 2011).

According to the Bayesian equation, the merging is based on multiplying the likelihood by the *a priori* elevation; the likelihood is based on maximizing the probability. Recalling that the Bayesian rule, used to get the *a posteriori* probability, is represented by:

$$p(\theta|z) = \frac{p(z|\theta)p(\theta)}{p(z)} = \frac{\text{likelihood} \times \text{prior}}{\text{normalizing factor}} \quad 5-13$$

Recalling equation 5-7, and assuming that the sensor error is normally distributed with mean z and variance σ^2 , $N(z, \sigma^2)$ (which represents the quality of the digital surface model) if there are two sensors, then:

$$z_1 = z + \varepsilon_1, \quad \varepsilon_1 \sim N(z_2, \sigma_1^2) \quad 5-14$$

$$z_2 = z + \varepsilon_2, \quad \varepsilon_2 \sim N(z_2, \sigma_2^2) \quad 5-15$$

where N is normal distribution.

From the joint likelihood function it is possible to get the following:

$$p(z_1, z_2|z) = p(z_1|z) \times p(z_2|z)$$

5-16

The elevations predicted through maximizing entropy have associated probabilities. Suppose that the elevation error is normally distributed, that the mean error is equal to mean elevation which is in this case is $(z_1, \text{ or } z_2)$, then the estimated *a priori* probability of the elevation, from $(DSM_{k(x,y)})$, can be based on a smoothed surface. This smoothing can be achieved by maximizing local entropy; three smoothed data sets are produced, based on a 3x3, a 5x5 and a 9x9 window respectively. For the two models being considered here, i.e. $(DSM_{k(x,y)})$ where $k = 1$ or $k = 2$, the estimated *a priori* probability of the elevation can be expressed:

$$p(z_1) \sim N(z_{1p}, \sigma_1^2) \quad 5-17$$

$$p(z_2) \sim N(z_{2p}, \sigma_2^2) \quad 5-18$$

where:

$p(.)$ is the probability distribution of the error which is normally distributed $N(. , .)$;

and,

σ^2 is the variance of the error which is represented by the square of the quality or RMSE of the DSM.

For each data set the probability can be calculated and, since they are independent, the product will be:

$$p(z_{1p}, z_{2p}) = p(z_{1p}) \times p(z_{2p}) \quad 5-19$$

The, *a priori* elevations are obtained by maximising the entropy index, as discussed in the next section in more detail. This is based on the assumption that the surface of the building and ground are smooth, therefore by maximizing the entropy index it is possible to get a value which can be used as an *a priori* elevation and which later can be used in a probability form.

Regarding variance, the other parameter of the normal distribution, it will be the quality parameter of the original digital surface model, although the quality of the digital surface

model could be considered to have changed slightly since assigning new heights to the candidate pixels in the model (that change will be very small and can be neglected since a large area is being dealt with, which consists of thousands of pixels, moreover that change is between -0.25m and 0.25m, therefore it will have very little effect on the overall quality of the digital surface model).

It was mentioned in section 4.9, that the Bayesian approach can be simplified to include only the likelihood and *a priori* data, and that $p(z)$, the normalization factor, can be eliminated since the situation is not dealing with finding the absolute value of the probability.

Recalling the simplified Bayesian equation:

$$p(z|z_1, z_2) = p(z|\theta)p(\theta) = p(z_1, z_2|z)p(z_{1p}, z_{2p}) \quad 5-20$$

and, assuming there are only two sensors, this results in:

$$\hat{z}_{MAP} = \operatorname{argmax} p(z|z_1, z_2) = (z|z_1, z_2) \sim e^{-\left(\frac{(z_1-z)^2}{2\sigma_1^2} + \frac{(z_2-z)^2}{2\sigma_2^2} + \frac{(z_{1p}-z)^2}{2\sigma_1^2} + \frac{(z_{2p}-z)^2}{2\sigma_2^2}\right)} \quad 5-21$$

Simplifying the equation and minimizing the result by taking the log of equation 5-21, the result of the Maximum A Posteriori (probability) (or MAP) which represents the merged digital surface models can be obtained from the following expression:

$$\hat{z}_{MAP} = \frac{\left[\frac{z_1}{\sigma_1^2} + \frac{z_2}{\sigma_2^2} + \frac{z_{1p}}{\sigma_1^2} + \frac{z_{2p}}{\sigma_2^2} \right]}{\left[\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} + \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right]} \quad 5-22$$

The result represents the value after merging two sensors. It is clear that the merging depends on the weight that is assigned to each DSM and this is obtained from assessing the quality of each digital surface model. It is clear that weight is inversely proportional to height and more weight is given to the lower variance or higher quality value and less to the low quality digital surface model. In addition to the weight, the other factor that influences the result is the value of the estimated *a priori* elevations, the derivation of which will be explained later, in section 5.5.

The code used for merging DSMs implementing Bayesian approaches is listed in Appendix B.

In the case of merging more than two sensors and accepting that the errors in the Digital elevation models are independently distributed, so the merging will be carried out by using the previous result as *a priori* and incorporating the new data in the Bayesian as likelihood (Beyerer et al., 2011). If and when a new digital surface model arrives, for example when there are three sets of data, the approach will be as follows:

use the result of merging the two data sets as obtained from equation 5-22, to give a *priori* probability of the elevation, and use the third data set (z_3) in its original form, with the merged data set (Z_{MAP_old}) in a weighted average operation, as shown in equation 5-23:

$$\hat{z}_{MAP_new} = \frac{\frac{z_3}{\sigma_3^2} + \frac{Z_{MAP_old}}{\sigma_{MAP_old}^2}}{\frac{1}{\sigma_3^2} + \frac{1}{\sigma_{MAP_old}^2}} \quad 5-23$$

The equation 5-23 leads to minimizing the uncertainty in the digital surface model, by using the merged digital surface model as *a priori* elevation data. The new digital surface model data (z_3) will be used to minimize that uncertainty. It is worth mentioning that in addition to obtaining an *a priori* value, it is important to determine the quality of the *a priori* elevation, also. It is clear that the quality of the prior digital surface model depends on the original quality of the input digital surface model; many approaches have been investigated to evaluate the data quality (σ). Torlegård *et al.*, (1986) argue that the total quality of the digital surface model can be determined by summing the quality of all the digital surface models and multiplying by a factor, which is either known or derived from photogrammetric input process, such as height of the flight, quality of the image, control points errors' propagation and operation and instrument precision. On the other hand, (Chunmei et al., 2013; Hu et al., 2009; Weng, 2002) all calculate the total quality of different digital surface models by adopting the less rigorous approach of adding the quality as shown in the equation 5-24:

$$\sigma_{MAP_old}^2 = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1^2 \sigma_2^2}$$

5-24

5.5 Estimating the *a priori* using Maximum Entropy

The above-mentioned characteristics can be exploited to measure surface roughness, while respecting multiple surfaces and distinguishing between rough and smooth surfaces. From this, it can be understood that entropy can be used to measure randomness with higher values indicating higher randomness and lower values representing lower randomness or lower disorder. From the viewpoint of energy, surface roughness can be represented by energy, with an excessive rate of similar minimal surface roughness values referring to a high value for entropy. The opposite is correct, when the values are not the same, or not close to each other; in that case roughness leads to low entropy. It is clear that entropy is representing the randomness in elevation variations which, consequently, can be employed to characterize the digital surface model. These characteristics can be used to manifest the representation of surface roughness, by assuming that the surface is smooth when the entropy is at the maximum and it is rough (or not smooth) when the entropy is lowest. Based on this assumption one can try to maximize the entropy by changing the value of the middle of the window. Maximizing the entropy by assigning different values will give an elevation that can be considered as the value of the *a priori* elevation.

According to the earlier discussion, about entropy and how entropy can be used to represent roughness, and as illustrated in Figure 5.6, maximum entropy is obtained when the probabilities are similar, and the uncertainty is high between the probabilities, i.e. the elevations are close to each other and the surface is smooth.

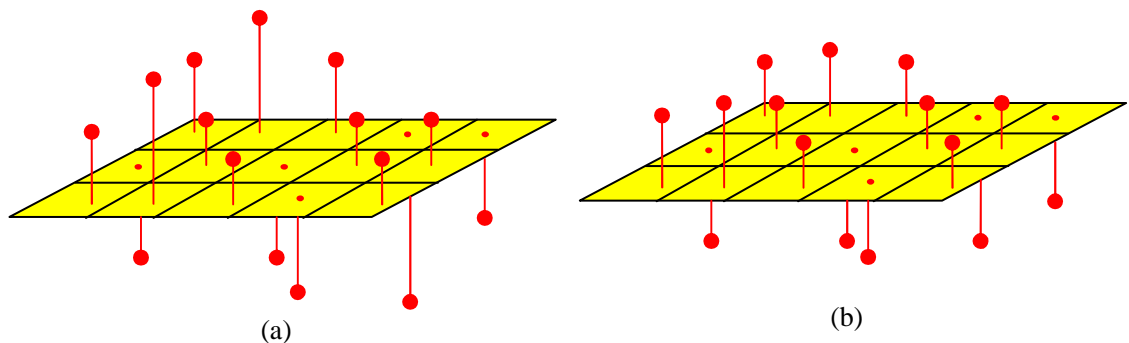


Figure 5.6 DSM shows different value of elevations used to represent the entropy (a) the entropy is low since the randomness in the elevations is high (b) the entropy is high since the variance in the elevation is low.

Shannon's approach to entropy has been used by researchers in different image analysis

applications. Telfer et al. (2003) implemented entropy in order to smooth the signal by removing the noise. Shiozaki, (1986) and (Dai and Wang, 2007) applied the principle of entropy to detected the edge based on using the pixel's intensity to detect the entropy and the edge is represent at the location where the entropy is high.

The assumption that is made when constructing the *a priori* value is to build on the assumption that building surfaces are smooth and have low roughness. This assumption can be quantified by using maximum entropy. The group of probabilities p_{ij} from p_{11} to p_{MN} in a specific window size MN , are represented by the value H_f . The equation 5-25 represents a one dimension signal, but it can be transformed to apply to a window.

$$H_f = \sum_{i=1}^M \sum_{j=1}^N (p_{ij} * \log p_{ij}) \quad 5-25$$

Where:

$$p_{ij} = f(x,y) / \sum_{i=1}^M \sum_{j=1}^N f(i,j) \quad 5-26$$

The values of the height have been transformed into probability with the specific window, MN . At each element of the window, the height probability has been evaluated by dividing each pixel value, f , by the total values of elevation within the specific window in order to find p_{ij} as shown in Figure 5.7.

In order to determine the *a priori* elevation value, the window is generated each time the maximum entropy is determined for a specific elevation. The model that is used for determining the entropy is illustrated in Equation 5-27.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \prod Hf(x) \quad 5-27$$

$\hat{\theta}$ in equation 5.19 represents the maximum entropy, and the value obtained can be considered an *initial* elevation, which will be used to create an *a priori* probability, this is obtained by maximizing each patch's entropy. This approach can be followed for each Digital Surface Model individually.

For the optimization, a window 3x3, see Figure 5., is traversed over the entire image in a

convolution operation, at each position the reference point $P_{i,j}$, the centre of the window, is simulated and is processed using equation 5-27 in order to find the maximum entropy. The height value (H) that produced the maximum entropy will be used as *a priori*. It is obvious that the value of $P_{i,j}$ is extremely dependent on the neighbour pixels and their pixel values.

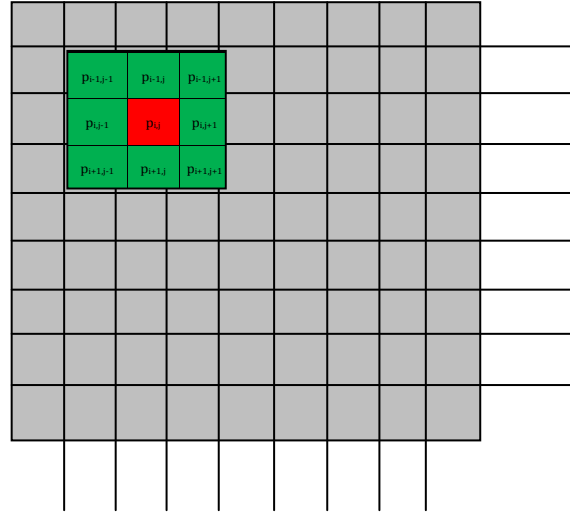


Figure 5.7 Hyperpixel size 3x3 (green pixels), that is used to find the *a priori* elevation (red pixel) and is passed over the DIGITAL SURFACE MODEL (grey pixels).

5.6 Prior Optimization

In optimizing the maximum entropy, it was difficult to use the Monte-Carlo approach to get a best value. The reason behind this is that the Monte-Carlo method depends on assigning a value randomly sampled from a distribution based on a specified mean denoted by its elevation and variance; sometimes, in the approach used in this work, the random number generated is too far from the mean because the assigned distribution has the limits $[-\infty, \infty]$. Consequently, this achieves an extreme value for the probability due to high roughness within the specified window, causing extreme modification of the ground morphology. This was happening because the entropy measurement was affected by boundary pixels. For that purpose, it was decided to find an alternative for simulating the error by creating a loop incrementing by 0.01m within a specific range. Two ranges were nominated: $\pm 0.1\text{m}$ and $\pm 0.25\text{m}$. In addition to that, the time cost has been reduced dramatically, with the Monte-Carlo method the time was 6 hours and 4 minutes for 1k iteration and 12 hours for 250,000 iteration using 3x3 windows, and using a 2 x CPU 2.3 GHz processor. However it cost less than 6 minutes to do the merging with a loop increment method, based on an increment value 0.01 which lead to the number of iterations being twenty for the $\pm 0.1\text{m}$ (i.e. -0.1m to $+0.1\text{m}$) range and fifty iterations for the $\pm 0.25\text{m}$ (i.e. -0.25m to $+0.25\text{m}$) range.

5.7 The Implementation of Merging

In the following section, the implementation process of the merging is outlined, by showing how the data are prepared and used in two different models for the merging: Maximum Likelihood (weighted average) and a Bayesian approach.

Regarding the merging using, the Maximum Likelihood approach (i.e. weighted average):

1. Calculate the quality of each DSM using the measured ground checkpoints as discussed in section 5.2.
2. Apply the weighted average model that is explained by equation 5-11 in the case of two data sets, and the equation 5-12 implemented in the case of more than two data sets, as discussed in section 5.4.1.
3. The format of the data and the C++ code used is shown in Appendix A.

Regarding the Merging using the Bayesian approaches (i.e. maximizing local entropy):

1. Find out the quality of each DSM using the measured checkpoints as discussed in section 5.2.
2. Find the *a priori* elevation using the model described in sec. 5.5 and the Equation 5-27. Different window sizes have been used to find the *a priori* value in order to estimate the prior elevation (i.e. windows of 3x3, 5,5 and 7x7, see sec. 5.5), and for each window size two ranges of elevation has been used in the optimization process, namely prior elevation $\pm 0.1\text{m}$ and prior elevation $\pm 0.25\text{m}$ as discussed in section. 5.6.
3. Use the model described by equation 5-22 to achieve the merging.
4. The processing code for the merging and for the *a priori* elevation calculation is shown in Appendix B.

5.8 Summary

The merging process has been examined in this chapter. The first stage was built on constructing a model representing the relationship between an ideal data set and one containing error; the second stage used Maximum Likelihood and a Bayesian approach, in order to find an estimate for the merged digital surface model. The result obtained was either the Maximum Likelihood or the maximum *a posteriori* probability of the real

digital surface model in the case of having provided an *a priori* probability distribution. In this chapter the following has been discussed: eliminating the blunders from checkpoints for assessing the quality of the digital surface models using RMSE, by assuming the errors are uniformly distributed over the digital surface models; and, how the *a priori* probability of elevation has been evaluated for digital surface models using a maximum entropy index based on measuring the surface roughness. The chapter discussed the optimization process for finding the merged digital surface models.

It has been assumed that the source digital surface models are correctly coordinated and there is no need for co-registration, although this assumption is not always valid and the DSMs should then be registered.

Chapter 6 Result and Validation of Merged Digital Surface Models

The evaluation of merging algorithms and models developed for this research is presented in this chapter. It includes the evaluation of each created and merged DSM that has been generated from optical satellite imagery implementing the developed models. For evaluation purposes check data have been used which were acquired by differential GNSS observations and are considered to be the ‘true’ ground elevations. Both quantitative and qualitative analyses have been used in this evaluation. Empirical equations and statistical tests have been used in the quantitative evaluation and with respect to the qualitative evaluation visual inspection has been used based on generated profiles and slope maps from the DSMs.

6.1 Reference Data

To proceed in the validation, it was necessary to get reference data of known accuracy. The data that were used for this purpose were limited to GNSS checkpoints. The checkpoints that have been used to validate the original and merged DSMs were acquired in the field using Leica Geosystem differential GNSS. The points are referenced to the Universal Transverse Mercator (UTM) projection zone 30 ‘North’ and the height datum referenced to ellipsoid WGS-84. The number of checkpoints measured was 61. The number of the points were used in the quality assessment were 30 and the rest have been used in the validation as shown in Figure 6.1.

6.2 Evaluation Indices

The accuracy and precision of the defined model has been evaluated using two types of assessments; Root Mean Square Error (RMSE) and ‘determination coefficient’, r^2 , see equation 6-1 and 6-2, respectively, for further clarification.



Figure 6.1 Validation checkpoints, the 31 checkpoints that have been used in the validation stage distributed over study area.

The RMSE has been used to give an indication of accuracy rather than precision, and because it retains the effect of any bias embedded in the error. For that reason the RMSE has been preferred to standard deviation of error (Abdullah, 2013). Podobnikar (2009) and Li et al., (2013b) extolled RMSE for evaluating a DSMs. By utilizing some checkpoints measured in the field it can express random and systematic errors in the DSMs as well as standard deviation of error and bias.

In determining RMSE, the difference of values between that predicted from a model and the ‘true’ value is measured. Each individual difference is a discrepancy, and may be considered an error. The RMSE value combines these errors into a single powerful

statistic. RMSE can be used to provide an estimation of error in any process model (Hansen and Schjoerring, 2003). As a means to represent error, RMSE has been increasingly used, according to Willmott et al., (1985).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Z_{observed,i} - Z_{model,i})^2}{n}} \quad 6-1$$

The RMSE index has been used in much research and it has been extolled by researchers because it is a straightforward calculation and not difficult for implementation. Nevertheless, it does not give an estimation for quality, at each individual post (i.e. interpolated grid point), in the surface that results from interpolation, therefore RMSE should be considered a global value for all DSMs (Wood, 1996), though unique to each.

There are three types of error in height developed during DSM construction, these are the well-established blunder, systematic and random error. RMSE can initially represent systematic and random errors, and can reveal systematic errors (Li et al., 2013b), prior to their removal. According to ASPRS Guidelines for height assessment (ASPRS, 2015) $1.9600 * RMSE$ is used to define vertical (or elevation) accuracy with a 95% Confidence Level (CL). This inference is based on the assumption that the discrepancies in elevation are random, follow a normal distribution and that neither systematic error nor blunders remain.

r^2 is the determination coefficient and helps to report the magnitude of the variance of the data as shown by the model (i.e. regression line). The coefficient of determination value ranges between 0 and 1 and the better agreement is indicated by a higher value (Legates and McCabe, 1999).

First the r value, the Pearson's correlation coefficient, will be determined using equation 6-2 (Legates and McCabe, 1999) and then from it r^2 is calculated:

$$r = \frac{n \sum xy - (\sum x) (\sum y)}{\sqrt{n(\sum x^2) - (\sum x)^2} \cdot \sqrt{n(\sum y^2) - (\sum y)^2}} \quad 6-2$$

where:

- x is the elevation value given in the first dataset
- y is the elevation value given in the second dataset
- n is the number of values

With r^2 the assumed linear relationship between two normally distributed variables can be represented; it gives an indication of the certainty with which data can be inferred from that linear relationship. Correlation between checkpoint heights and their DSM values has shown only a weak association with accuracy, which has led to it not being as popular as RMSE (Willmott et al., 1985). Nevertheless, it can be used to inspect the relation between the two variables (Li et al., 2013b), namely, the GCP height and the DSM height, see section 6.3.2.

6.3 DSM Quantitative Assessment

For the DSM assessment, a group of checkpoints (31 points) has been used to evaluate the quality of the DSMs. The merged DSMs can be classified into two types: Maximum Likelihood (i.e. weighted average) and Bayesian Merging. Within Bayesian Merging there were three groups derived from the implemented window size used in estimating the *a priori* probability of elevation, based on maximising local entropy, namely 3x3, 5x5 and 7x7 windows. Each of them has been evaluated based on two different iteration loops in a $\pm 0.1\text{m}$ range and a $\pm 0.5\text{m}$ range, using a 0.01m increment.

Merging (fusion) using Maximum Likelihood, as discussed in the previous chapter, consists of two steps, first determining the weight of each grid cell in each digital surface model, and then merging the digital surface models; the operation is performed grid wise. The first step is considered the most important; the second step is considered trivial and can be executed easily using equation 5-12. It should be noted that the Maximum Likelihood approach is applied when all the data are available (i.e. simultaneously) which is unlike the Bayesian approach, which allows use of the model even before obtaining all the data, as discussed in chapter four, section 04.11.

Instead of using checkpoints, some researchers have depended on using difference maps for evaluating the accuracy results (Karkee et al., 2006; Sadeq et al., 2012), in a raster environment, such as DSMs. This type of difference map is valid when a reference data set is available covering the whole study area and is not confined to only finite checkpoints. However, in the research that is the topic of this thesis, checkpoints have been used.

6.3.1 Statistical Assessment

For each DSM the RMSE, σ of error and maximum and minimum difference has been determined as shown in Table 6-1. The statistical tests that have been used in the DSM validation have been applied after eliminating blunders (point 21 is considered to be a blunder).

The coefficient r^2 has been used in order to test the correlation between the DSMs elevations and GCPs elevations. From the previous chapter it can be seen that the equation 5-22, represents the merged model using a Bayesian approach in the case when the measurements was limited to two digital surface models only. However, the equation 5-23 refers to a model supporting the continuous input of digital surface models, and is an update of the equation 5-13. The *a priori* probability of elevation is based on maximizing local entropy; in the entropy maximization stage two ranges have been used. The first range was 0.2m, from -0.1m to 0.1m, and the second range was 0.5m, from -0.25m to 0.25m.

During the quality assessment for the digital surface models produced by the Bayesian approach it is noticeable that the RMSE of the merged digital surface model with range $\pm 0.1m$ is smaller than the RMSE of the merged digital surface model with the $\pm 0.25m$ range for both the *a priori* probability of elevation obtained from the 3x3 and 5x5 windows, while the RMSE when the 7x7 window is used to obtain the *a priori* probability of elevation with the $\pm 0.25m$ range was better than that achieved with $\pm 0.1m$ range. The σ of error values were almost the same in all cases (i.e. all size windows and all increments for estimating *a priori* elevation), see Table 6-1.

From Table 6-1, the bias (or average discrepancy) in the merged digital surface models can be seen. The bias produced when merging using any Bayesian approach is larger than that resulting from the Maximum Likelihood merging method. Considering two Bayesian approaches, using a 3 x 3 window, the bias using the smaller range ($\pm 0.1m$) is less than when using the larger range ($\pm 0.25m$), being -0.255m and -0.292m respectively. It can be noticed that the errors (discrepancies), which are represented by RMSE, for the Maximum Likelihood case was 0.375m while for all the Bayesian approaches was larger, i.e. being 0.392m, or greater.

In the original source data, the maximum positive discrepancies were 0.391m and 0.221m in the digital surface models produced from WV-1 and Pleiades respectively, and the relevant maximum negative discrepancies were -0.766m and -1.348m.

It should be noted that the standard deviation of the error (or σ of discrepancies), which can be considered to be an unbiased representation of error, has been reduced in all the merged DSMs to be better than that for the original data. Initially it was 0.359m (Pleiades) and 0.320m (WorldView-1) and the merging has led it to be 0.306m in the worst case.

To estimate the *a priori* probability of elevation, local entropy has been utilized. This has been based on three different window sizes and for each window size two increment ranges, namely ± 0.1 m and ± 0.25 m, the increment being 0.01m.

From Table 6.1, it can be noticed that similar values for the RMSE are found in the three Bayesian approaches recorded in columns E, G and J, namely: 0.393m, 0.394m and 0.392m, respectively. For the Bayesian approach recorded in column F, the RMSE value was 0.418m. For the Bayesian approaches recorded in columns H and I, the RMSE value was 0.431m and 0.432m respectively.

With respect to the σ of the discrepancies, the values for the Bayesian merging approaches are slightly smaller than those for the Maximum Likelihood merging, indicating less variability in the discrepancies, in the Bayesian methods. The Bayesian approaches also had less variability in the discrepancies than the original DSMs from Pleiades and WorldView-1 (which had σ s of discrepancies of 0.359m and 0.320m respectively). It can be mentioned that the value of the σ of discrepancies was higher in WorldView-1 before removing the blunder, being 0.391m before removal and 0.320m after removal.

Table 6-1 quality characteristics of the two digital surface models using the checkpoints. Red text indicates that the error in the point is greater than 1.96 x RMSE or outside the 95% confidence interval

	A	B	C	D	E	F	G	H	I	J
Pt. #.	GNSS checkpoint elevation (m)	Pleiades checkpoint elevation (m)	WV-1 checkpoint elevation (m) without blunder	Max. Likel. checkpoint elevation (m)	Bayesian merging 3x3 window ±0.1m range checkpoint elevation (m)	Bayesian merging 3x3 window ±0.25m range checkpoint elevation (m)	Bayesian merging 5x5 window ±0.1m range checkpoint elevation (m)	Bayesian merging 5x5 window ±0.25m range checkpoint elevation (m)	Bayesian merging 7x7 window ±0.1m range checkpoint elevation (m)	Bayesian merging 7x7 window ±0.25m range checkpoint elevation (m)
0	76.237	76.080	76.159	76.119	76.148	76.183	76.169	76.223	76.239	76.169
1	98.334	98.998	99.049	99.039	99.059	99.094	99.016	99.061	99.012	98.995
2	72.050	72.249	72.075	72.077	72.104	72.139	72.114	72.163	72.180	72.127
3	71.025	72.071	71.510	71.638	71.664	71.699	71.634	71.682	71.655	71.613
4	62.967	63.627	62.793	62.953	62.997	63.032	63.003	63.073	63.073	63.003
5	96.169	96.532	96.478	96.505	96.555	96.613	96.555	96.625	96.625	96.555
6	95.382	95.161	95.169	94.857	94.886	94.923	94.907	94.977	94.977	94.907
7	76.409	77.137	76.602	76.695	76.729	76.765	76.741	76.790	76.811	76.745
8	83.341	84.145	83.046	83.264	83.303	83.338	83.300	83.345	83.354	83.299
9	68.287	69.468	68.216	68.361	68.388	68.423	68.397	68.451	68.463	68.396
10	88.579	89.453	88.802	88.877	88.915	88.950	88.927	88.972	88.994	88.927
11	87.847	89.195	88.163	88.298	88.325	88.360	88.340	88.388	88.393	88.336
12	85.504	86.239	85.499	85.491	85.538	85.586	85.541	85.609	85.611	85.541
13	85.256	86.303	85.523	85.643	85.651	85.686	85.664	85.709	85.712	85.657
14	97.737	98.507	97.804	97.952	97.970	98.005	97.962	98.007	98.007	97.953
15	85.755	86.603	85.847	85.913	85.938	85.973	85.959	86.011	86.029	85.963
16	111.676	112.618	112.130	112.191	112.212	112.247	112.233	112.279	112.299	112.241
17	88.241	89.067	88.606	88.724	88.774	88.809	88.774	88.844	88.844	88.774
18	87.633	88.103	87.561	87.609	87.645	87.680	87.659	87.716	87.729	87.659
19	93.710	94.549	94.109	94.105	94.142	94.177	94.146	94.192	94.197	94.147
20										
21	108.976	109.616	108.585	108.783	108.828	108.872	108.813	108.858	108.848	108.793
22	110.201	110.967	110.558	110.603	110.638	110.673	110.653	110.698	110.723	110.653
23	94.383	94.554	93.997	94.085	94.107	94.142	94.089	94.134	94.121	94.068
24	108.169	108.809	107.991	108.136	108.168	108.203	108.186	108.256	108.256	108.186
25	124.891	125.606	125.550	125.400	125.441	125.479	125.450	125.520	125.520	125.450
26	114.165	114.514	114.419	114.458	114.486	114.521	114.483	114.535	114.540	114.487
27	111.280	111.772	111.448	111.502	111.552	111.595	111.548	111.593	111.598	111.543
28	131.356	131.862	132.013	131.986	132.002	132.037	131.982	132.031	132.018	131.965
29	109.522	109.970	109.710	109.788	109.829	109.864	109.838	109.908	109.908	109.838
30	94.514	94.699	95.280	95.241	95.253	95.294	95.291	95.361	95.361	95.291
RMSE		0.721	0.357	0.375	0.393	0.418	0.394	0.431	0.432	0.392
95% Confidence level		1.414	0.701	0.735	0.770	0.819	0.772	0.845	0.846	0.768
number of points		30	30	30	30	30	30	30	30	30
Arithmetic mean of discrepancies or bias, (m)		-0.629	-0.170	-0.223	-0.255	-0.292	-0.259	-0.314	-0.317	-0.256
σ of discrepancies (m)		0.359	0.320	0.306	0.304	0.304	0.301	0.301	0.298	0.301
Max +ve discrepancy (m)		0.221	0.391	0.525	0.496	0.459	0.475	0.405	0.405	0.475
Max -ve discrepancy (m)		-1.348	-0.766	-0.727	-0.739	-0.779	-0.777	-0.847	-0.847	-0.777

6.3.2 Correlation Assessment

Further data analysis can be supported by Figure 6.2. Correlation (r^2) has recently been supported by the author and colleagues, as a means to quickly compare derived elevations and their checkpoints, Sadeq et al. (2012)

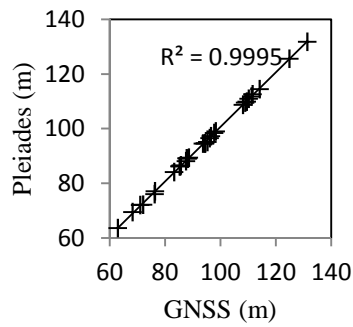
The correlation between the merged digital surface model's elevations values and those of the checkpoints has not decreased for any of the types of merging. Although the correlation was strong, especially for the WV-1_DSM where it was 0.9997 (for the Pleiades_DSM it was 0.9995), it has remained the same in the merged digital surface models where the correlation was 0.9997 with the Maximum Likelihood and Bayesian approaches for both the $\pm 0.1\text{m}$ and $\pm 0.25\text{m}$ ranges.

Figure 6.2 consists of twenty-seven diagrams, arranged as nine triplets. Each one of the nine sets of triplets validates a single method of determining elevations. The first two sets validate the original Pleiades and WorldView-1 sources and the remaining seven sets validate sources merged in different ways.

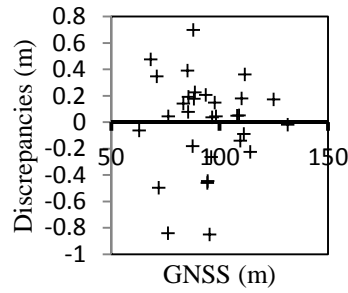
The leftmost diagram of each triplet, of Figure 6.2, shows the correlation (r^2) between checkpoint elevations (CPs) and the elevations provided by the method being considered. Checkpoint elevations are on the x-axis and those of the method being considered are shown on the y-axis.

The central diagram of each triplet, of Figure 6.2, shows a scatterplot of each CPs discrepancy.

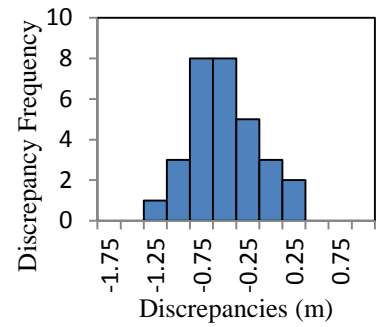
The rightmost diagram of each triplet (of Figure 6.2) uses histograms to show the distribution of discrepancies between the CPs and elevations provided by the method being considered, across the range of elevations being considered.



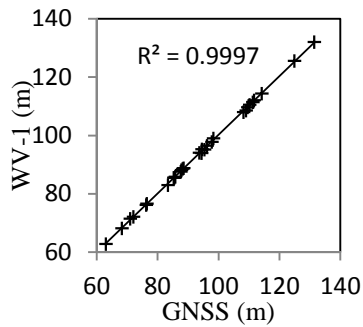
a-1) Pleiades vs. Checkpoints



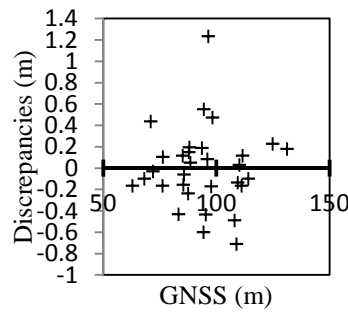
a-2) Pleiades and Checkpoint



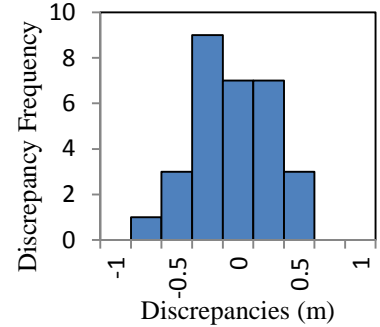
a-3) Pleiades and Checkpoint



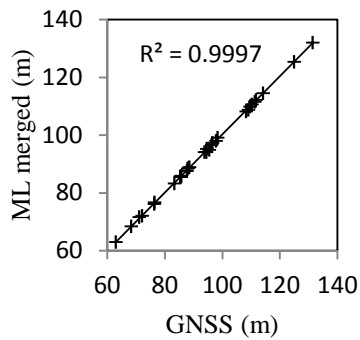
b-1) WV-1 vs. Checkpoints



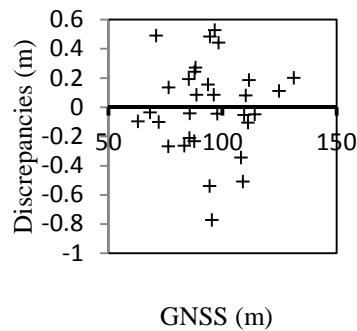
b-2) WV-1 vs. Checkpoints



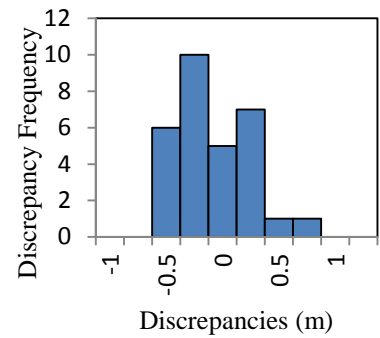
b-3) WV-1 vs. Checkpoints



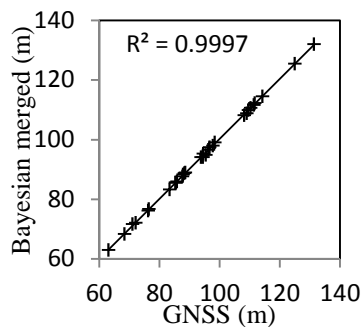
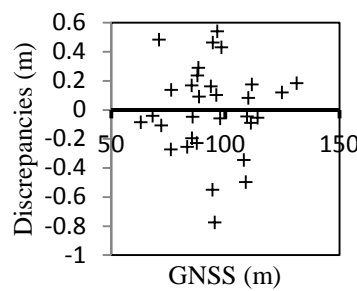
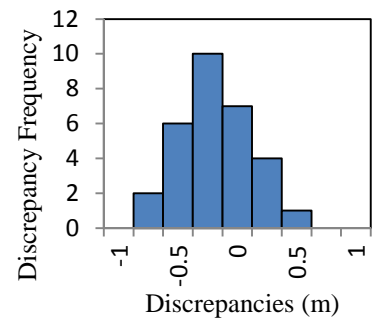
c-1) Maximum Likelihood (ML) merging vs. Checkpoints

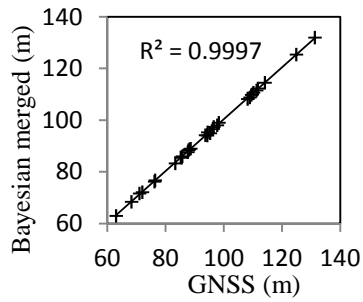


c-2) Maximum Likelihood (ML) merging vs. Checkpoints

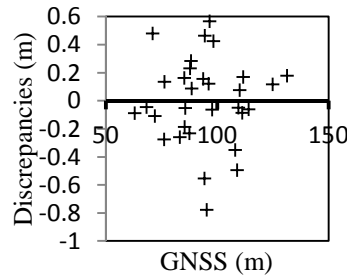


c-3) Maximum Likelihood (ML) merging vs. Checkpoints

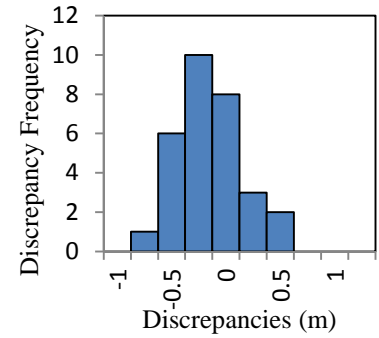
d-1) Bayesian merging - range $\pm 0.1m$ window 3×3 vs. Checkpoints.d-2) Bayesian merging - range $\pm 0.1m$ window 3×3 vs. Checkpoints.d-3) Bayesian merging - range $\pm 0.1m$ window 3×3 vs. Checkpoints.



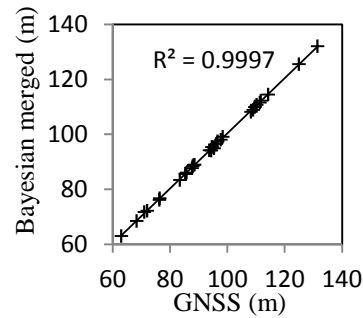
e-1) Bayesian merging - range $\pm 0.25\text{m}$ window 3×3 vs. Checkpoints



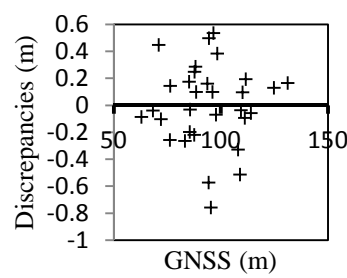
e-2) Bayesian merging - range $\pm 0.25\text{m}$ window 3×3 vs. Checkpoints



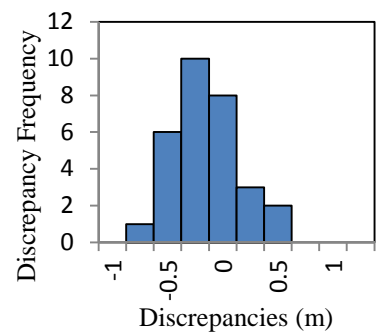
e-3) Bayesian merging - range $\pm 0.25\text{m}$ window 3×3 vs. Checkpoints



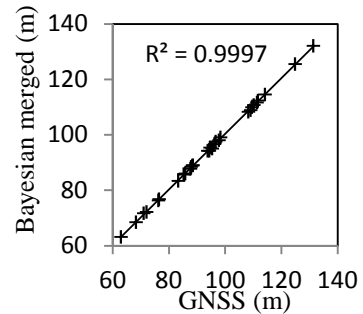
f-1) Bayesian merging - range $\pm 0.1\text{m}$ window 5×5 and Checkpoints.



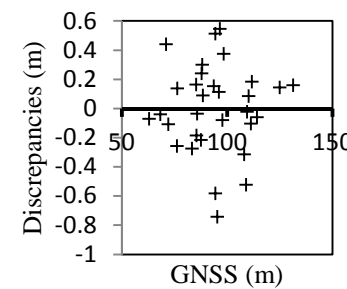
f-2) Bayesian merging - range $\pm 0.1\text{m}$ window 5×5 and Checkpoints.



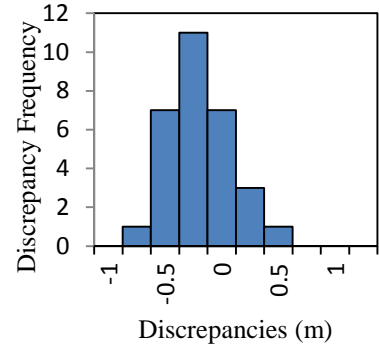
f-3) Bayesian merging - range $\pm 0.1\text{m}$ window 5×5 and Checkpoints.



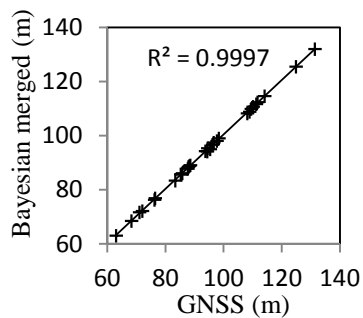
g-1) Bayesian merging - range $\pm 0.25\text{m}$ window 5×5 vs. Checkpoints



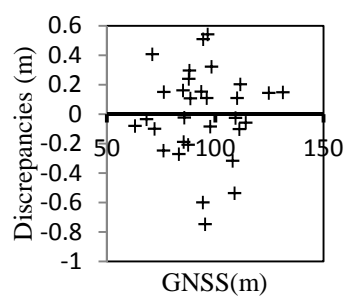
g-2) Bayesian merging - range $\pm 0.25\text{m}$ window 5×5 vs. Checkpoints



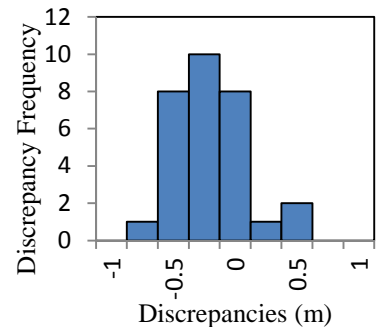
g-3) Bayesian merging - range $\pm 0.25\text{m}$ window 5×5 vs. Checkpoints



h-1) Bayesian merging - range $\pm 0.1\text{m}$ window 7×7 vs. Checkpoints



h-2) Bayesian merging - range $\pm 0.1\text{m}$ window 7×7 vs. Checkpoints



h-3) Bayesian merging - range $\pm 0.1\text{m}$ window 7×7 vs. Checkpoints

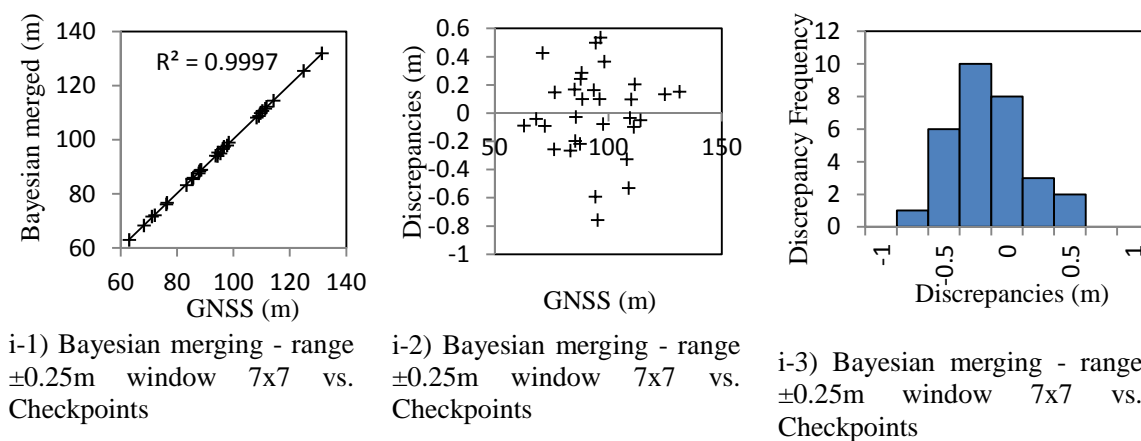


Figure 6.2 Comparison of the correlation, scatter plot and histograms for the input data and merging results using Maximum Likelihood and Bayesian techniques against checkpoints (CPs), where: the left column represents the correlation of the original and merged DSMs heights against CP heights; the middle column represents the scatter plot of the CP discrepancies and the original or merged DSMs, relative to the ellipsoidal height; and, the right column represents the frequency of the CP discrepancies (errors).

The error distribution has been assessed for each of the produced DSMs. It can be seen that in figures Figure 6.2 (rightmost diagrams) that the errors in the Bayesian approaches seem more normally distributed than the Maximum Likelihood and the standard deviation of the error in the Bayesian approaches were the same as or lower than for the Maximum Likelihood approaches, as can also be seen in Table 6-1.

The error illustrated in Figure 6.2, shows the differences between the DSMs either from the original or a merged DSM and the ‘true’ values. This has been measured by taking the difference from of the GCP elevations and the aforementioned DSMs at each checkpoint, as shown in Figure 6.1.

The Bayesian approach has considerable influence on imposing normality on the error distribution as can be seen from the plots of each type regardless of the window size and the iteration range used in looping for estimating the *a priori* probability of the elevation value. An analysis of the discrepancy scatter plots Figure 6.2 (middle column), as recommended by Rusling and Kumosinski (1996) shows that all the values are randomly distributed which further validates the linear relationship between all sets of interpolated heights and their checkpoint values.

By investigating the error of the Bayesian approach using an estimated *a priori* probability of elevation produced from different ranges $\pm 0.1\text{m}$ and $\pm 0.25\text{m}$, it indicates that the Bayesian approach was able to remove the systematic error during merging the

DSMs.

6.4 DSM Qualitative Assessment

In addition to the quantitative assessment, a qualitative assessment has been implemented, as well. This has been achieved by plotting the height profiles and comparing them for a specific location, also the slope map has been produced to be used in validation and to assess the effect of merging on smoothing the surface.

Figure 6.4 shows the profile (line A-B see Figure 6.3) that was produced at the merging stage, against the original profiles. The weighted average (Maximum Likelihood) approach enhances the DSM by removing irregularities in the underlying DSM. As has been shown earlier the RMSE value of the DSM produced from WorldView-1 is smaller than that produced by the Pleiades data, but the effect on the output DSM is not proportional, and this reflects the implemented equation, which is not linear. Figure 6.4 shows, graphically, that the WorldView-1 DSM has, through the weighting process based on the DSM's quality, more influence on the underlying DSM than does Pleiades.

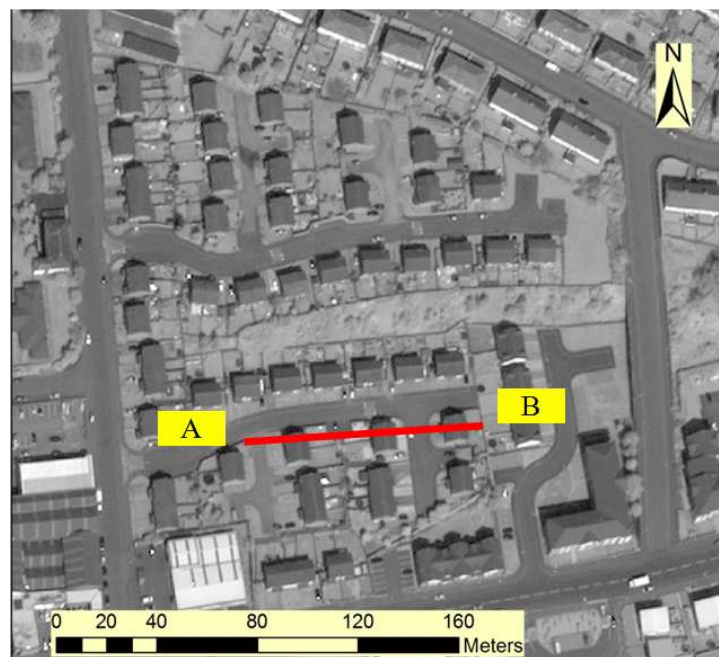


Figure 6.3 Orthoimagery for the study area indicating the specified profile location.

Due to variations in satellite geometry, it can be noticed from Figure 6.4 that there is a misregistration in the DSMs and consequently in the produced profile. Generally, since the geometry of satellite imagery is rather weak this causes extra errors in the generated elevation (Teo et al., 2010); moreover the Pleiades stereo imagery has a weaker geometry

than the WV-1 stereo imagery. From Table 1-2 it can be seen that the Pleiades sensor altitude is higher, by around 200km, than the WV-1 sensor. Moreover Table 3-1 shows that the incident image angle for Pleiades, $11.91^{\circ}+14.03^{\circ}=25.94^{\circ}$, is less than that for WV-1 which is $17.33^{\circ}+21.01^{\circ}=38.34^{\circ}$. This shows that the WV-1 base-to-height ratio is higher than Pleiades, which consequently leads to WV-1 having better accuracy than Pleiades (Teo et al., 2010).

Figure 6.4 shows the result of using a Bayesian approach to merging digital surface models. The Bayesian approach, with an estimated *a priori* probability of elevation, had more effect on smoothing than the Maximum Likelihood approach. Figure 6.4 shows how the Bayesian approach was able to remove the high peaks from the profile and increase the smoothness of the profile while the Maximum Likelihood approach was slightly less sensitive with respect to smoothing the surface of the DSM.

The merged digital surface model has a smoother surface than the original digital surface model. The Bayesian approach is able to remove the peaks from the building, and has more influence on the resulting digital surface model than Maximum Likelihood, especially when the range used to infer an *a priori* probability of elevation has been increased to $\pm 0.25\text{m}$ instead of $\pm 0.1\text{m}$.

The result of the merging using different window sizes has been investigated also. Figure 6.5 shows different profiles along the line A-B over the DSMs that were produced from using different window sizes 3x3, 5x5 and 7x7 with simulation range $\pm 0.1\text{m}$. It is clear from the profile that there is not much difference between these types, except the profile for the used window 5x5 has an average falling between the other two window sizes (i.e. 3x3 and 7x7). The other range for variance, simulation range $\pm 0.25\text{m}$, has been tested also as shown in Figure 6.6. The profiles show that the effects of the change of the window size with the specified variance are slight. Again, the window size 5x5 is close to the average of the windows 3x3 and 7x7, as in the case with the simulation range $\pm 0.1\text{m}$

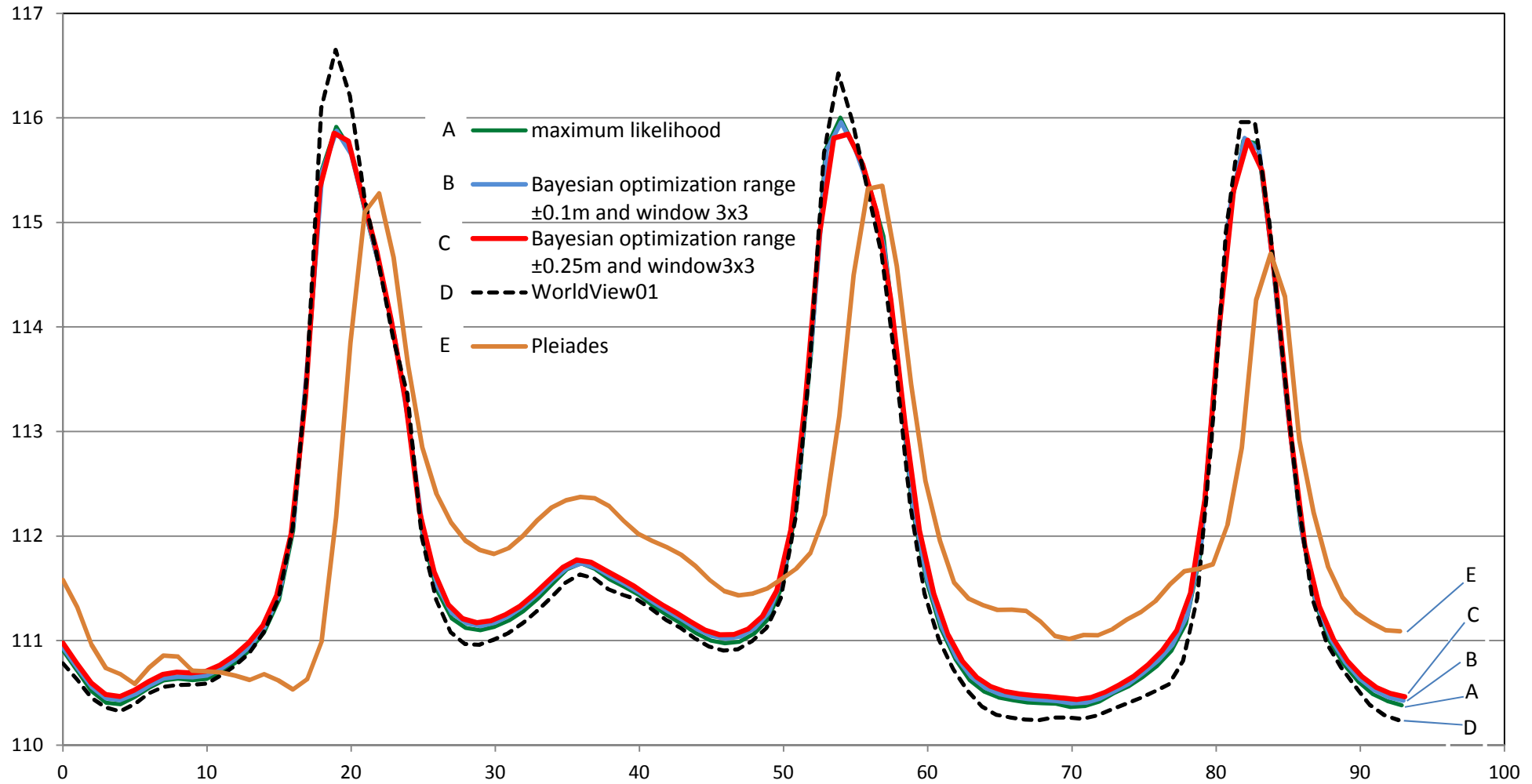


Figure 6.4 Profile along line A-B as shown in Figure 6.3, showing the merging results using Maximum Likelihood, Bayesian approaches and the data sets: Pleiades DSM and WorldView-1 DSM. The Y-axis is elevation (m) and the X-axis distance (m) along the AB profile.

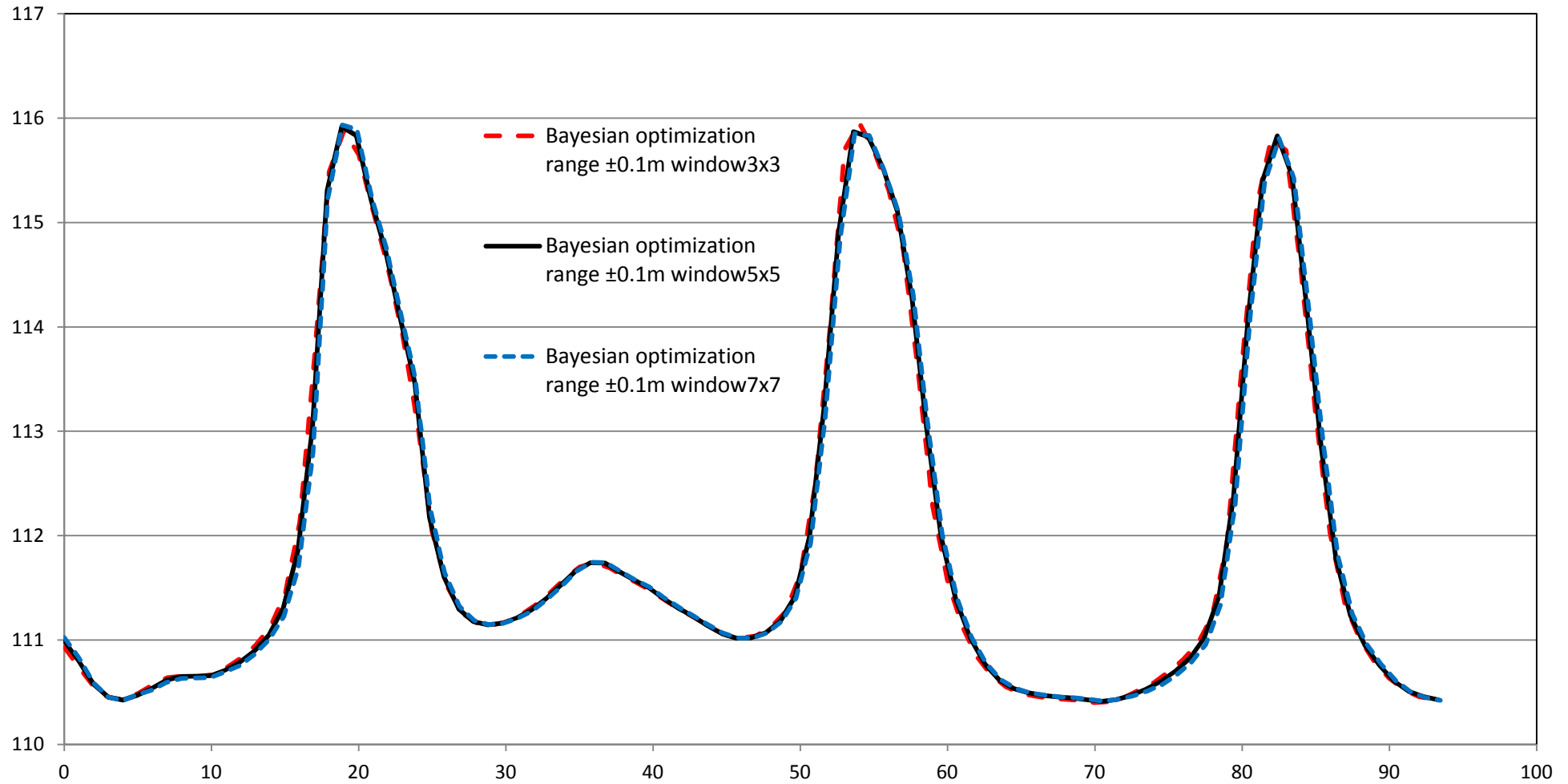


Figure 6.5 Profile over the line A-B as shown in Figure 6.3. It shows the comparison between the merging results using the Bayesian approach with range $\pm 0.1\text{m}$. The Y-axis is elevation (m) and the X-axis distance (m) along the AB profile.

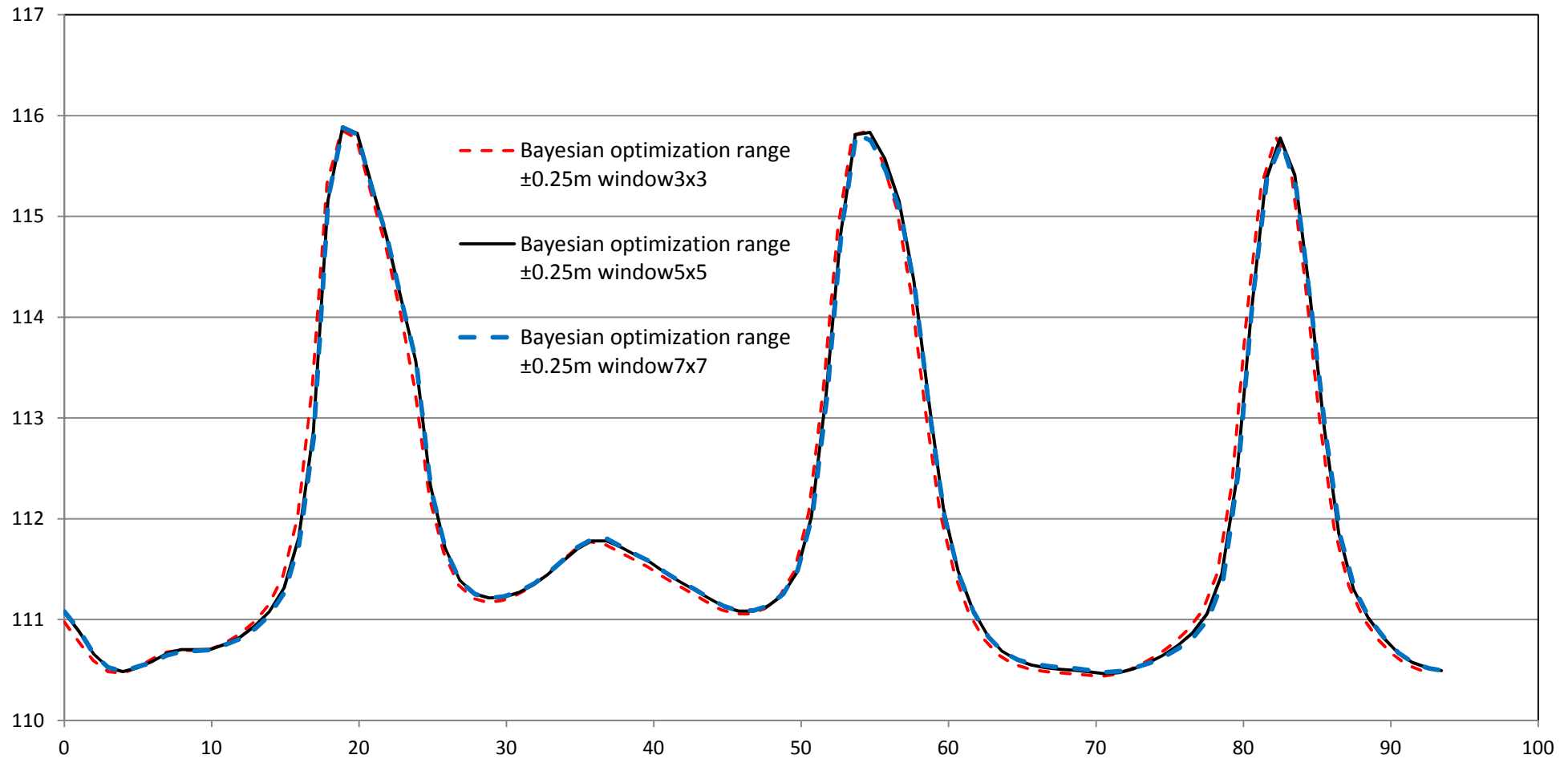


Figure 6.6 Profile over the line A-B as shown in Figure 6.3 it shows the comparison between the merging results using Bayesian approach with simulation range $\pm 0.25\text{m}$. The Y-axis is elevation (m) and the X-axis distance (m) along the AB profile.

Another profile has been taken at a different location as shown in Figure 6.7. This profile passes over two checkpoints. The first checkpoint is number 26 with elevation as shown in Table 6-1, which is equal to 114.165m, and the other is number 29 with elevation 109.522m.

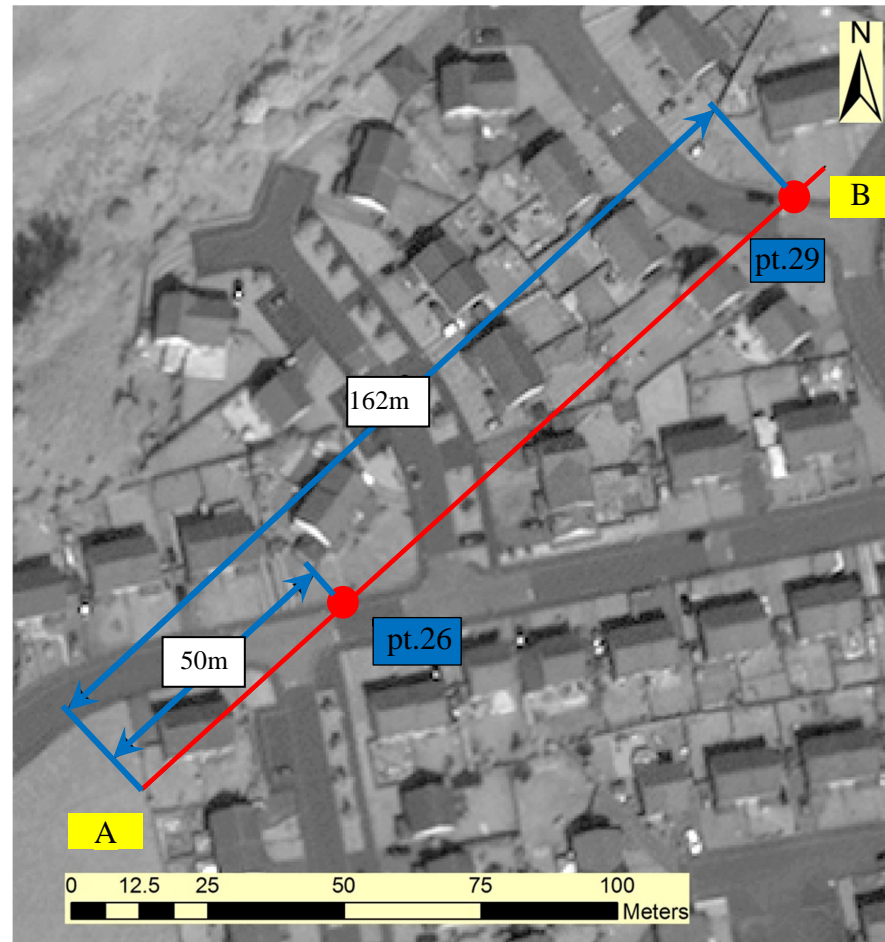


Figure 6.7 A profile located over two checkpoints marked with red dots, being located at points No 26 and 29 in Table 6-1.

The profile shows that there is a misregistration between the profiles from DSMs of Pleiades and WV-1. This is around 3m in the building location, as can be noticed in the profiles in the Figure 6.4 and Figure 6.8, however. The building near point A, in the Figure 6.7, has almost disappeared from the Pleiades DSM probably since it has low height and the profile passes through the edge of the building. As mentioned earlier in this section, the misregistration error is due to the Pleiades geometry being weaker than that of the WV-1 stereo imagery. The effect of the misregistration has not been further considered and the DSMs are treated as fully registered.

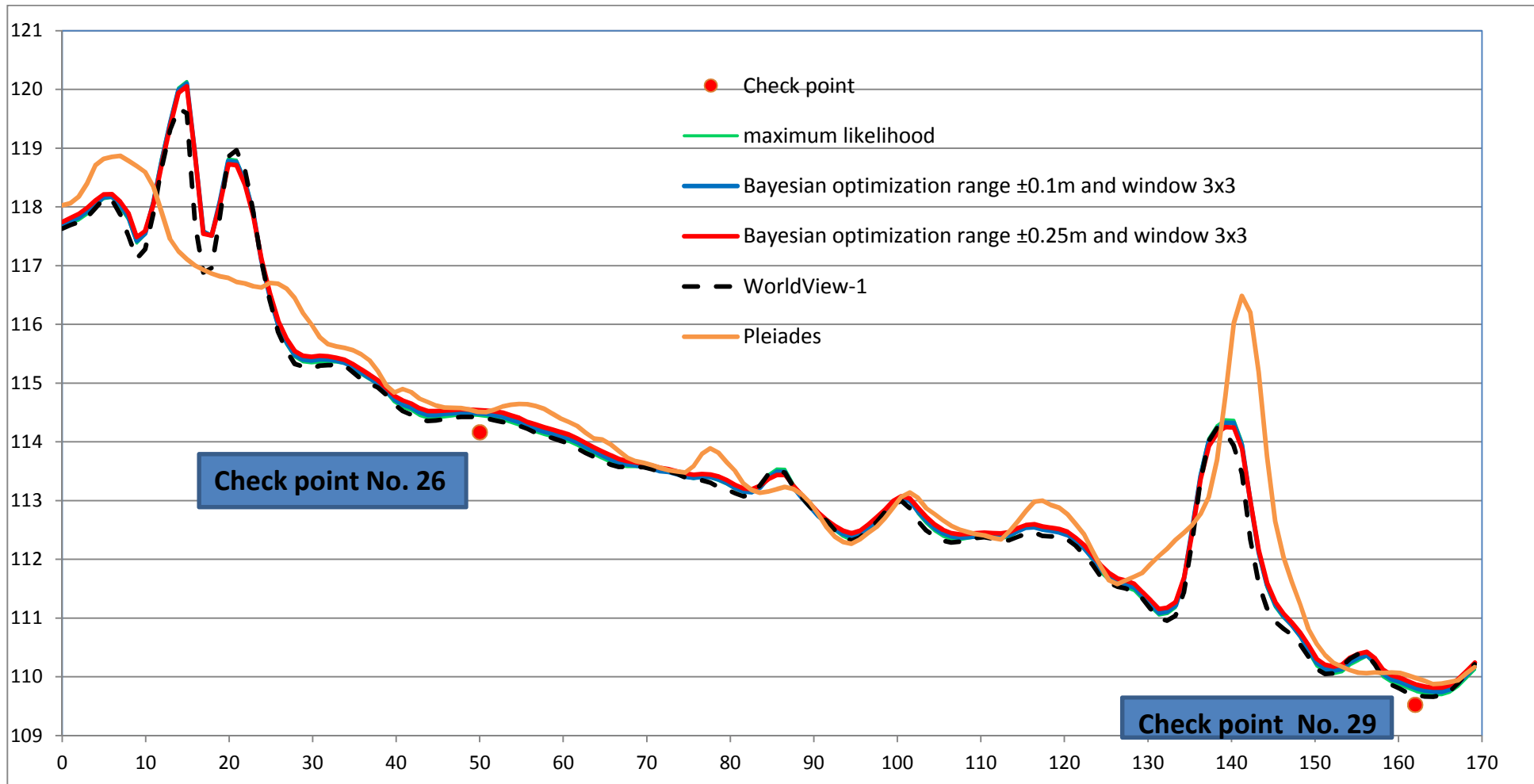
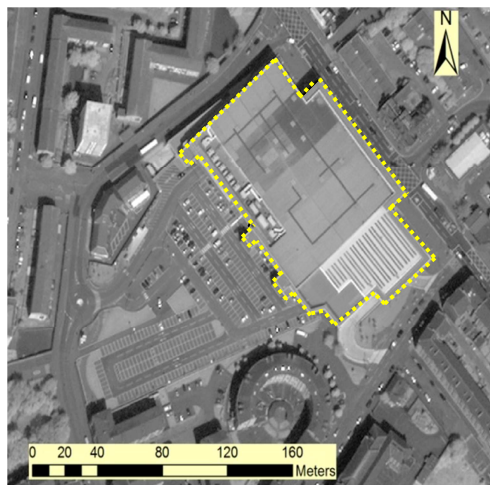


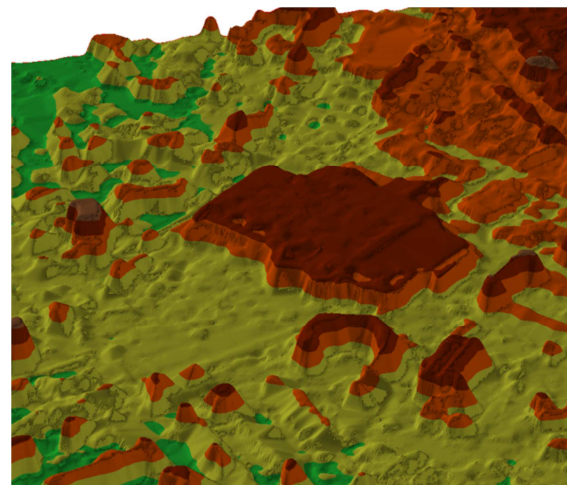
Figure 6.8 Profile over the line A-B as shown in Figure 6.7. It shows the comparison between the merging results using the Bayesian approach with simulation range $\pm 0.25\text{m}$, and its relation with the checkpoints. The Y-axis is elevation (m) and the X-axis distance (m) along the A-B profile, passing through the checkpoints 26 and 29

6.4.1 Slope Analysis

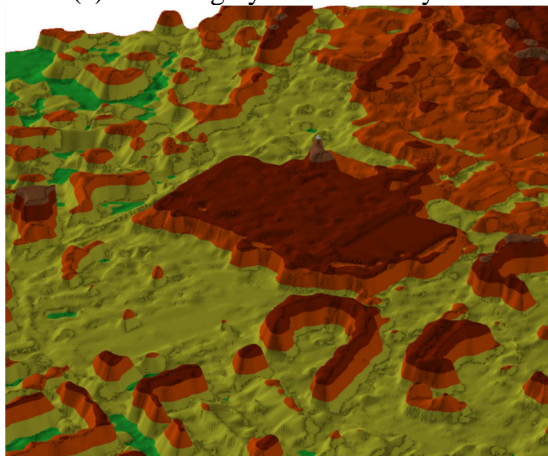
Further DSM assessment has been achieved by producing a slope map. In the analysis of the produced maps, it has been shown that the Bayesian approach has an active affect on smoothing the flat surfaces during the merging process. This can support the assumption that has been made during estimating the *a priori* probability of elevation used in the Bayesian approach. The assumption is based on the surface of structures (such as buildings and roads) having flat (i.e. planar) surfaces rather than rough surfaces, as has been shown in the DSMs produced from photogrammetric processing. The building roofs for a specific area are shown in Figure 6.9 (a and b), and they show the orthoimagery for the flat surface structure. The building roofs in the original DSM from Pleiades, which is used in the merging, are shown in Figure 6.9 (c) and are speckled and have a very rough surface while in reality the main central building has a flat surface and does not have any small irregularities. However, in the merging using the Bayesian approach, see Figure 6.9(d), it is obvious that the produced DSM does not suffer from much roughness and it almost represents the real structure without many affects of roughness.



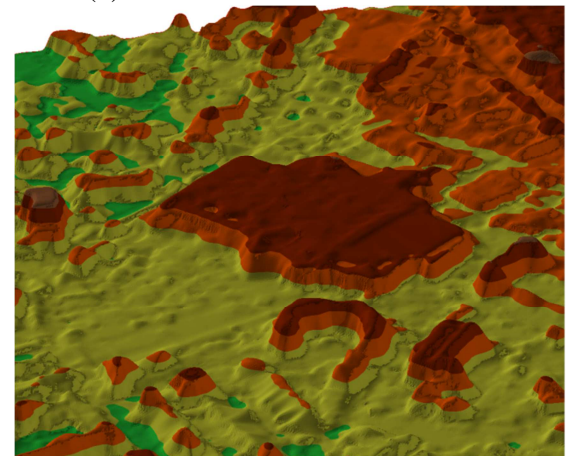
(a) orthoimagery over the study area



(b) WV-1 DSM in 3D view format



(c) Pleiades DSM in 3D view format



(d) Bayesian merged DSM in 3D view format

Figure 6.9 Orthoimagery and 3D view to examine the effect of the merging over a flat surfaced structure – outlined in yellow.

For further detailed qualitative validation, different slope maps have been produced either for the original DSMs or for the merged DSMs as shown in Figure 6.10, in order to find out the change in the surface slope between each stage. The circle referred to as “b” overlaid on the slope map see Figure 6.10 (a) and the circle referred to as “a” overlaid on the slope map Figure 6.10(b), show that some noise exists in the original DSMs used in the merging. This has been reduced in the DSMs merged using Maximum Likelihood and Bayesian approaches. with an optimized simulation range of $\pm 0.25\text{m}$ and a 3×3 kernel window, Figure 6.10 (c and d) for both of the marked circles “a” and “b” in the Figure 6.10 (c and d). From the Figure 6.10 (c and d), it is clear that the noise, which existed in the original DSM, has been minimized in the marked circles.

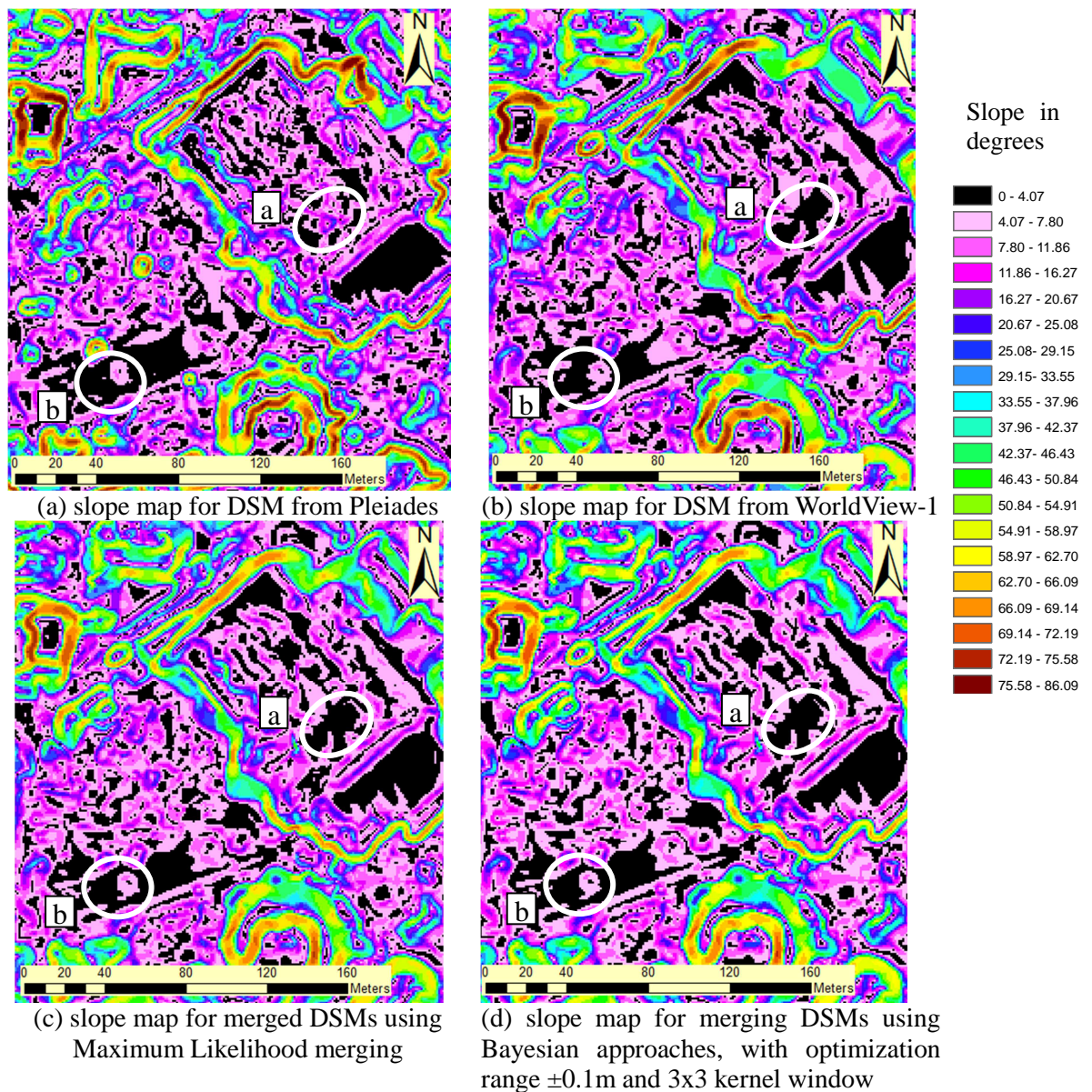


Figure 6.10 Slope map analysis for merged DSMs, the white symbolization shows the effect of merging on removing the slope.

The slope map has been used further in order to find out the changes between different DSMs that were produced using a Bayesian merging approach, by using different window

sizes and variance ranges during estimating an *a priori* probability for elevation. From visual scrutinizing of the slope map in Figure 6.11(a) to (e), it can be noticed that the Bayesian approach was able to flatten the surface of the structures in all cases. In addition, it can be seen that there are no dramatic differences among the different slope maps with regard to the flat-roofed building and almost all of the results are similar.

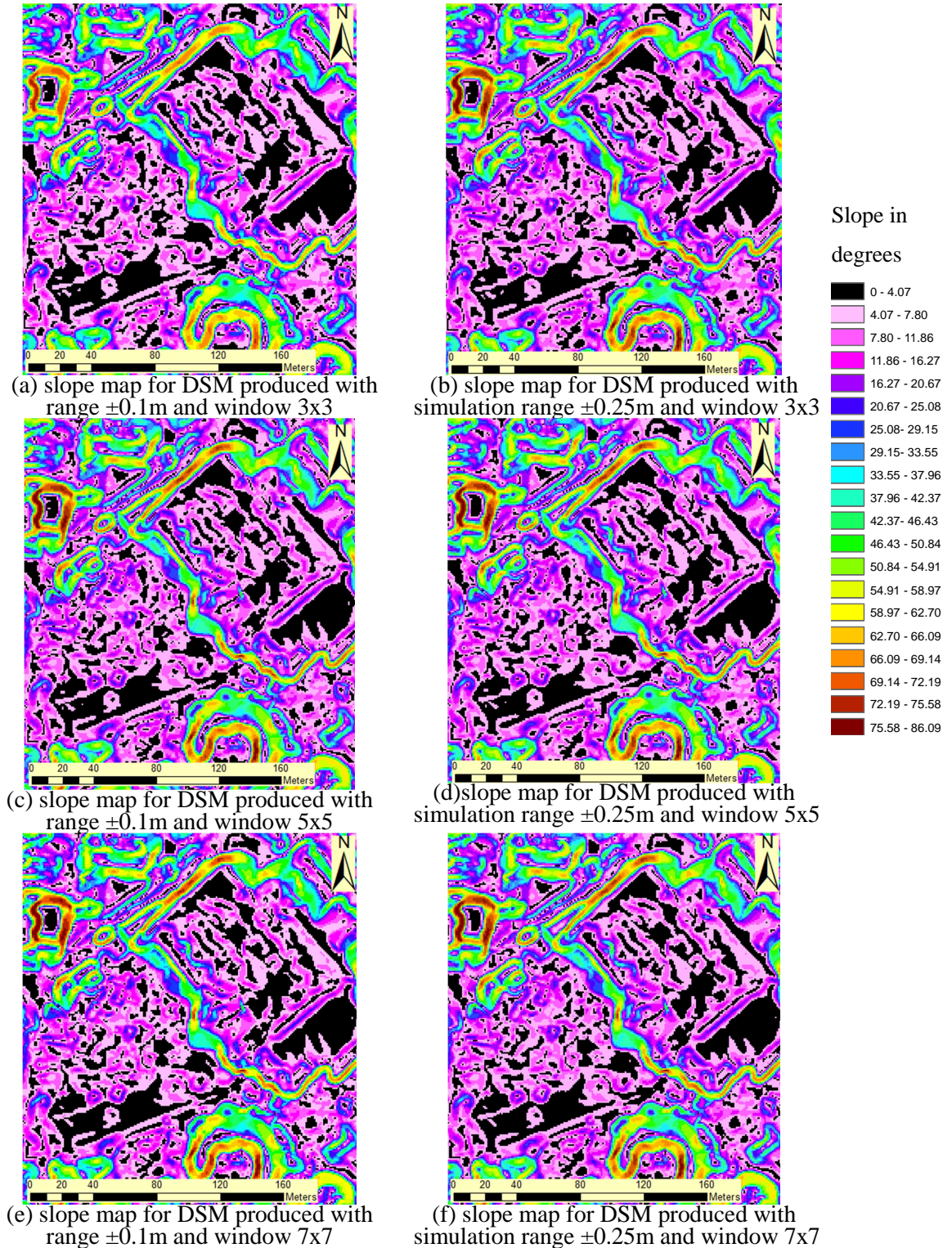


Figure 6.11 slope map visualisation for merged DSMs.

For further detailed analysis of the DSM's slope map, Table 6-2 shows the statistics analysis

for each slope map within the study area, explained in Figure 6.11. The statistical analysis is represented by the arithmetic mean of the slope values across the whole study area and the slope standard deviations across that area. In the slope map analysis, the average slope of the merged DSMs using the Bayesian approaches (e.g. all used window size and variances) is less than that for the slopes of the merged DSM achieved using the Maximum Likelihood approach. Moreover, the arithmetic mean values decrease with increasing the window size or with increasing the range value, because with a higher increment value, the patch that is used to infer an *a priori* probability of elevation has become smoother. On the other hand, the standard deviations of slope across the whole study area for the merged DSMs are less than the original DSMs used in the merging. The standard deviation of the merged DSMs using Maximum Likelihood method is less than the DSMs using Bayesian approaches. Moreover, the standard deviation of slope across the whole study area rose with increased the window size, and for each window size the standard deviation is increase with increasing the range (e.g. with simulation range of $\pm 0.25\text{m}$ rather than $\pm 0.1\text{m}$).

The standard deviation of the slope map of the WV-1 DSM is less than the Pleiades DSM. However, the standard deviation of the merged DSM is slightly less than the standard deviations of the slope maps of the original DSMs WV-1 and less than the Pleiades DSM., hence the merged DSM is smoother than original DSMs used in the merging. Based on the author's own local knowledge the smoother representation of roofs and ground surfaces is considered closer to the 'truth'.

In reality, the ground surface and the roofs of the buildings are considered to have smooth surfaces. This leads to the assumption that the pixel elevations in the windows used (e.g. size 3×3 , 5×5 or 7×7) change uniformly and there are no sudden changes in the surface elevations inside the windows. By simulating the elevation in the centre of the window and fixing the surrounding's elevation, the *a priori* elevation to be applied in merging the DSMs using the Bayesian approach, is obtained. During the simulation, the value of the elevation that optimizes the local entropy is selected as *a priori*, as discussed section 5.5.

Table 6-2 Merged and original slope map statistical analysis.		
Source of the DSM	Arithmetic mean of slope (degrees)	σ of slope (degrees)
WorldView-1 satellite imagery(A)	25.402	19.919
Pleiades satellite imagery(B)	27.740	21.199
Merging (A and B) with Maximum Likelihood	25.292	19.495
Merging (A and B) with Bayesian range $\pm 0.1\text{m}$ and 3x3window	25.279	19.628
Merging (A and B) with Bayesian simulation range $\pm 0.25\text{m}$ and 3x3window	25.220	19.515
Merging (A and B) with Bayesian simulation range $\pm 0.1\text{m}$ and 5x5window	25.196	19.725
Merging (A and B) with Bayesian simulation range $\pm 0.25\text{m}$ and 5x5window	25.037	19.617
Merging (A and B) with Bayesian simulation range $\pm 0.1\text{m}$ and 7x7window	25.135	19.778
Merging (A and B) with Bayesian simulation range $\pm 0.25\text{m}$ and 7x7window	25.892	19.705

6.5 Summary

The quantitative analysis have been implemented to evaluate the merged DSMs. The statistical analysis using RMSE, implementing field check points, has showed that there is no significant difference between the merged DSMs using Maximum Likelihood and Bayesian approach (using different window sizes 3x3, 5x5, and 7x7). The Maximum Likelihood was better just by 0.02m than Bayesian approach. Respect the σ of error, a quality indicator after the removal of bias, gave a similar result for the Maximum likelihood and Bayesian approach.

Furthermore, the investigation has showed that the correlation between maximum likelihood and Bayesian approach has been kept to be the same, with respect to all window sizes. However, the error distribution has been to be more normally distributed using the Bayesian approach than maximum likelihood, this means that the Bayesian was able to remove the systemic error from the DSMs via merging.

For intense evaluation, profiles have been generated from different DEMs at two different locations. The quantitative evaluation has showed that both methods, Maximum Likelihood and Bayesian approaches, has produced similar result, however, Bayesian Approach was able to produce smoother DSM to be close to the reality.

To proceed further in the smoothness investigation, the Slope Analysis of ESRI's ArcGIS has been utilized, and this showed that both methods are able to smooth the surface.

From this chapter it can be concluded that the Bayesian approach has a smoothing effect on the surface, and also quality can be increased using the Bayesian approach if bias can be removed – otherwise the resulting accuracy of the Maximum Likelihood approach is better.

Chapter 7 Building Footprint and 3D Model Generation

It can be assumed that Digital Surface Models (DSMs), constructed by merging individual DSMs each produced from stereo processing different satellite images and as discussed in Chapter 3, are imperfect, due to the low resolution of the satellite imagery compared to aerial imagery. Therefore, it is difficult to use such DSMs in building footprint extraction, because parts of buildings are not clearly identified. In addition to that, parts of buildings have been merged with some non-building parts that have similar heights, such as trees, consequently affecting the shape of the building.

This chapter examines a methodology developed for building footprint extraction and subsequent 3D model construction from an imperfect DSM and related imagery. The flow-line comprises four main stages.

First, **data preparation**, including DSM and orthoimagery construction. The normalized DSM (nDSM) is produced and labelled according to the Connected Component Labelling algorithm. The orthoimagery is segmented. Shadows are detected. The Normalized Difference Vegetation Index (NDVI) is produced and the edge map generated.

Second, **building detection and extraction**. At this stage, rudimentary building locations are detected using the nDSM and then the initial building boundary is extracted from the segmented orthoimagery. In addition, the edge map is modified, using Mathematical Morphology (or MM), for subsequent use.

Third, **building boundary regularization**. A Bayesian approach is used to find the most probable coordinates of the building footprints, based on the modified edge map.

Fourth, and finally, **the 3D model is constructed** by extracting the elevations of buildings from the DSM and combining these with the regularized building boundaries.

The implemented code, including the functions for the building footprint extraction, has been written in C++ and listed in Appendix D, except the code for labelling the nDSM, which is listed in Appendix C.

The above-mentioned stages are listed in flow chart see Figure 7.1.

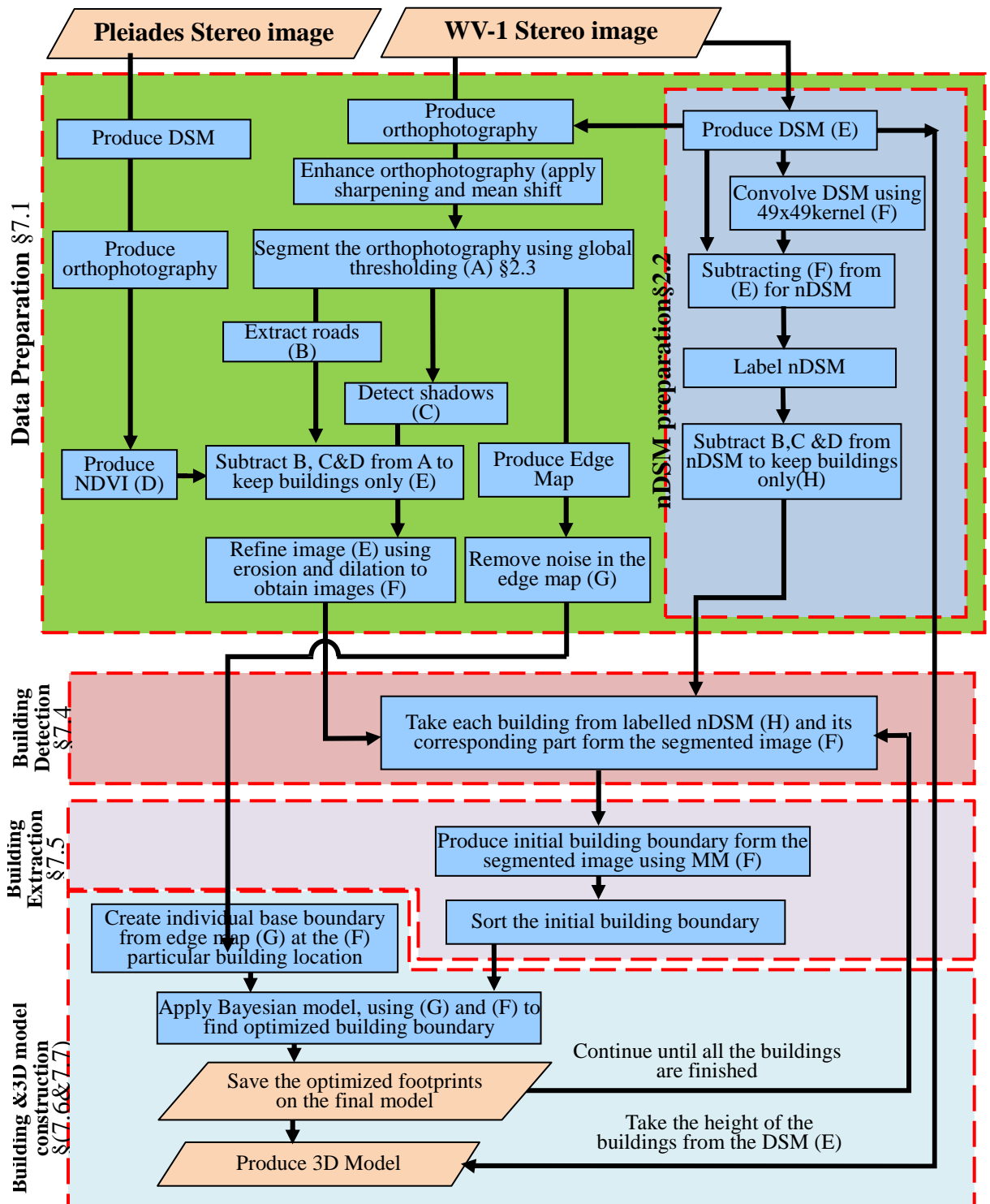


Figure 7.1 Flow chart illustrating the process for building footprint extraction, with the heading numbers at the left of the diagram.

7.1 Mathematical Morphology

Mathematical Morphology (MM), is a non-linear process commonly used in image analysis. It is based on modifying the geometrical shapes within the image, rather than pixel values. It was developed by Matheron and Serra in 1964 cited by Ronse et al. (2005), and subsequently gained popularity following three seminal publications from Matheron (1975) and Serra (1982 and 1988) also cited in Ronse et al. (2005). It has proved its value

in image analysis (Ronse et al., 2005). MM was originally created to deal with binary images, using sets to represent such images mathematically (Heijmans, 1992). It changes an image and makes it simpler, meanwhile retaining the main items in the image. It is worth mentioning that it can be applied on a greyscale image as well, but in this work only a binary image is considered. MM can be used at various stages in image processing such as, image pre-processing, image entity structure enhancement, image segmentation, and quantitative object description. The effects and results that can be obtained from using MM include: keeping the important characters of the objects in the image; filtering out irrelevant objects; thinning and thickening objects; object marking; finding the area and parameter of objects; extracting features from the image; detecting edges; removing noise; image restoration and reconstruction; etc. (Sonka et al., 1993).

In this section MM will be considered in more detail, since it has been used in many stages in developing the building extraction algorithm subsequently used in this work. The basic concept of MM is to test the geometrical composition of an image by matching it with small patterns at diverse positions in the image, while changing the dimensions and form of the “matching model” or “structuring element”, (see Figure 7.2(a,b,c and d)). The three most important factors in the structural element are: the shape of the structural element (circular, square, cross etc.); the size - either 3x3, 5x5, 7x7 etc.; and, the reference point of the structural element. All three have considerable effect on the result. The reference point is the basic point in the structural element that is used to assign it to the foreground pixel in the image; it is not compulsory for the reference point to be in the centre of the structural element, it could be anywhere. Also it not necessary that the structural element be symmetric, it can be non-symmetric as shown in the figures below (see Figure 7.2 (e, f, g and d)).

MM is based on set theory; the objects in binary images are represented by sets. All foreground pixels are represented as sets, which consist of a group of pixels that form a shape in the image. The basic operations that are introduced are: erosion; dilation; and opening and closing. These are considered in the following three sections.

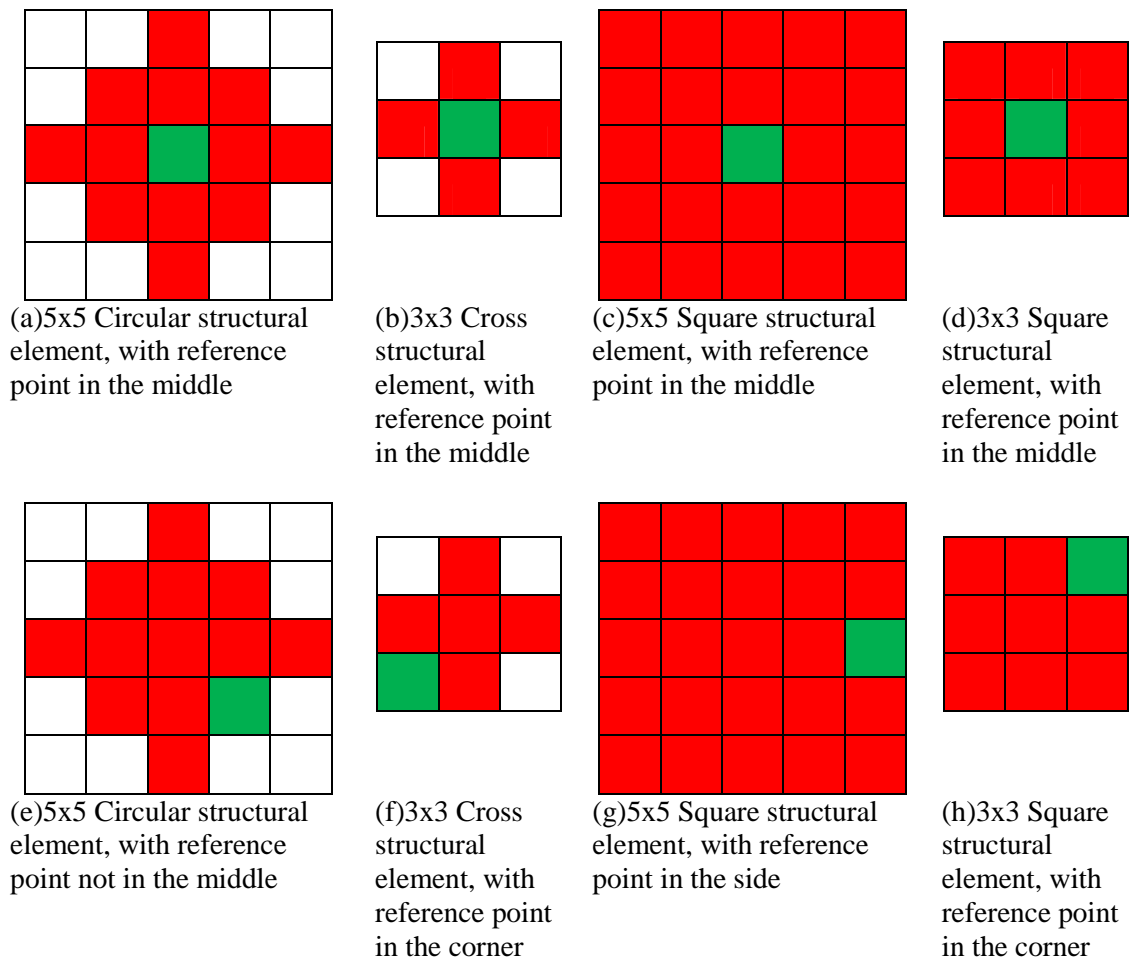


Figure 7.2 Different types and size of structural elements with different location for reference points, red is foreground pixel, white background pixels and green is the reference point in the structural element.

7.1.1 Erosion

Along with dilation, erosion is considered a fundamental action in MM (Pawar and Banga, 2012). This operation plays an important role in modifying the image. The consequence of this operation is to erode the margins of the objects in the binary image. This leads to a shrinking of the foreground area (i.e. reduces the size of the objects and thinning the size of linear objects); as a result the size of the holes will be increased. The operation is based on merging two sets using vector subtraction. The erosion of A (i.e. the binary image) by B (i.e. using structural element) to form A', can, be written as:

$$A' = A \ominus B \quad 7-1$$

The above formula can be understood to mean the erosion of the binary image A using structural element B to form A'; the symbol \ominus represents the erosion operation.

7.1.2 Dilation

The second most important operation in MM is dilation, which is the opposite of erosion. Applying this process will lead to an increase in the foreground pixels. The sizes of the objects are increased and the holes' sizes and gaps are decreased. In this operation, it is the union of the structural element pixel with the image pixel that represents dilation, which can be written as:

$$A' = A \oplus B \quad 7-2$$

The above formula is used to show that the binary image A is dilated with the structural element B, to form A'; the symbol \oplus is used to express the dilation operation.

Some of the structural elements that could be used in the erosion and dilation are listed in the Figure 7.2, and is it possible to use other types that may lead to better results.

7.1.3 Opening and Closing

When the image is eroded it is not possible to re-obtain the original image by dilating it, in other words the dilation is not the inverse of erosion and vice-versa. Alternatively, the images are simplified and will contain a smaller amount of detail. If an image is eroded and then dilated this is called opening, and can be represented by $X \circ B$, see equation 7-3. This equation represents dilating X by the structural element B, after having eroded it by the same structural element.

$$X \circ B = (X \ominus B) \oplus B \quad 7-3$$

The effect of opening is to smooth the contour of an image, and it isolates the objects from each other if they have been connected with a narrow strip of pixels. In summary opening removes small objects, such as noise, speckles and other objects smaller than the structural elements, and make the image simpler. In addition, the sizes of the holes are increased. The other operator is closing which is represented by $X \bullet B$, see equation (7-4), which means eroding the image with structural element B that had been dilated by the same structural element.

$$X \bullet B = (X \oplus B) \ominus B \quad 7-4$$

The effect of closing is on the hole, as it removes small holes, or their sizes are decreased. In addition, the closing operation smoothes the contours, as opening does, but the difference is that small break lines in the contours are joined.

The above operations were applied at different stages in the processing and were very effective during detecting and extracting buildings. They have been used to isolate and separate the buildings from each other and to smooth building boundaries, as well as extracting the buildings.

7.2 Data Preparation

The data, which are used in this stage, include the merged DSM and the orthoimagery. The DSM is obtained from implementing DSM merging using Bayesian theory as explained in Chapter 5. The constituent DSMs are produced from stereo satellite images, either WorldView-1 (or WV-1) or Pleiades, using photogrammetric techniques as outlined in Chapter 3. The merged DSM can be employed to obtain both the nDSM for rudimentary building location and the heights of the buildings for the 3D model construction. The other input is the orthoimagery, which is produced from the stereo satellite images by processing the satellite image from WV-1. WV-1 was chosen as that has better resolution, at 0.5m, than Pleiades at 0.7m. WV-1 offers stereo images with different viewing angles (or nadir angles) and in this case, nadir angles of 16.5 degrees and 17.2 degrees were used, and that with the lowest nadir angle (16.5 degrees) was actually used for the orthoimage production. In addition, the DSM that was used was that assessed to have a lower resolution - used because the higher resolution produced wavy edges. Since the classification is based on segmenting the orthoimagery, by extracting the roofs of the buildings, it is preferable to use the image that has a lower nadir angle, because near the nadir the building roofs show smaller shifts from their building footprints. This will affect the accuracy of the extracted buildings.

Making the orthoimagery, it was decided not to use a DSM that had a resolution of 0.5m, because it resulted in irregularities in the shape of the buildings (i.e. at the edges) in the orthoimagery (Zhou, 2010) (the irregularity arises because of using a DSM instead of a DTM). Generally, the DSM that is produced from photogrammetric techniques suffers from irregularity; this irregularity arises because the generated elevation value is based on image matching. The image matching is based on using correlation windows, with various window sizes between 5x5 pixels to 15x15 pixels. Due to different image perspective views (for example one image might contain more of one roof than the other might) it may happen that

matching based on correlation values does not quite identify conjugate pixels. Along a level feature such as a roof-line, where elevation values are expected to be constant, it may be that the calculated values following image matching are therefore uneven. For that reason, the produced edge will suffer from irregularity, and when the DSM is used in producing the orthoimagery the problem will be evident especially at the edges of buildings. If a lower resolution DSM has been applied during the production, using Socet GXP, with high smoothing the irregularity problem is overcome and produces better orthoimagery from the stereo images.

Therefore, the DSM that is used in producing the orthoimagery is produced from the same stereo images, as discussed in the Chapter 3, but with lower resolution (i.e. 10 meters). The comparison of two different types is shown in Figure 3.13. The other contributions to the algorithm, and obtained from the orthoimagery, are image segmentations, detected roads and an edge map. In addition, sourced from Pleiades, the NDVI image is available.

7.2.1 Image Segmentation

Image segmentation is considered to be the initial and most important part of image analysis (Gonzalez and Woods., 1992; Sonka et al., 1993). The aim of image segmentation is to partition the image into clusters, within the image, that have solid connections with objects or areas of the actual world. Segmentation is used to construct regional higher-level image structures, which probably correlate with structures in a higher-level object model. The segmentation process is executed before analyzing the shape of the object in 3-D; in general segmentation is considered to greatly influence success or failure in image analysis (Schalkoff, 1989). Russ (1999) indicates that image segmentation is a method used extensively in image analysis to reduce the amount of information needed for identifying image objects, as required.

As stated in section 1.6 a goal in this study is to ameliorate procedures to extract building footprints. Due to the difficulty of segmenting individual buildings, this amelioration may be achieved through partial segmentation. In partial segmentation the area does not represent the objects in the images completely, consequently, to obtain a complete segmentation, it is necessary to introduce a higher processing level (Sonka et al., 1993). In our case, the higher processing level is offered by MM and Bayesian statistics.

Autonomous segmentation, which is investigated here, was achieved by using a global thresholding method on the image - an important method in image segmentation. The idea is to analyse a grey-level histogram of the image so that the objects and the background are assigned to two grey levels. To distinguish one from the other, a threshold value (T) splits the image in such a way that pixels with a grey value greater than T are considered objects and a value '1' assigned to them, while the remaining pixels will have the value '0' assigned to them. The result will be a binary image with objects clearly identified on it.

The image used in segmentation, in this work, is an orthoimage, which as mentioned in section 7.2, is, along with the DSM, the main input data. Objects can be extracted from the orthoimage - such as buildings. It was necessary, prior to segmentation using global thresholding, to enhance the image for better results.

Several enhancements methods were used, as described in the following paragraphs.

First, image sharpening, this process started by convolving the orthoimagery using a 3×3 kernel, with all values equal to -1 and central value equal to 12. The sharpening leads to increased image contrast and gives emphasis to edges, however, it also give an emphasis to noise (Deserno, 2011). In this process, the value of the measured pixel is combined with its neighbouring pixel values based on the kernel used.

Second, image smoothing, which is achieved by passing a kernel 3×3 , with all values equal to 1, over the data in order to replace each pixel value by its neighbouring average. This will lead to a blurred image, but noise removal has been accentuated by the previous sharpening stage.

Next, the Mean-Shift Algorithm developed by Comaniciu et al. (2002), which is built-in to the ImageJ package, is applied. Its purpose is to delineate arbitrarily shaped clusters. The assigned values were spatial radius 5 representing the kernel dimension (i.e. the difference between the centre pixel and its neighbouring pixels) and colour distance which is used to fuse all the greyscale values that lie within the assigned colour limits and then assigning one value to them, which is represented by the mode. Figure 7.3 shows the comparison of the result of applying mean shift on the enhanced orthoimage (i.e. it is sharpened and smoothed).



Figure 7.3 Applying Mean shift on orthoimagery for study area produced from WV-1 satellite imagery (a) original orthoimagery after applying sharp and smoothing operation (b) applying mean shift algorithm with radius 5 and colour difference 5 on image shown at 'a'.

The Mean-Shift Algorithm is a non-parametric method (i.e. it works without needing to define any prior information and the shape of the clusters is not a required constraint). Typically, the algorithm is used to identify image clusters in Computer Vision and Image Analysis, and is based on defining a window with a specific circular size centred on a specific pixel in the data set. Later the defined window will be shifted based on the mean value of the pixels' intensity in the window; the calculated mean value will be used to find the magnitude and direction for shifting the original window for the next iteration. Repeating this until the result converges and the vector value change is so small that it does not need, effectively, to be shifted any more. Finally, the mean colour of the final iteration will be assigned to the starting location of the window's initial location.

The previous operation is trying to smooth the image by finding the mode inside a specific local region in the image, and later applying the mode value to the whole specific region. The Mean-Shift algorithm converts the image into homogenous tiles based on the closest neighbour pixels values and calculating the similarity inside each pixel group. It leads to better results in segmentation than by using a thresholding method because it is maintaining the edges of the objects and removing the noise that is not related to the building as shown in Figure 7.4.

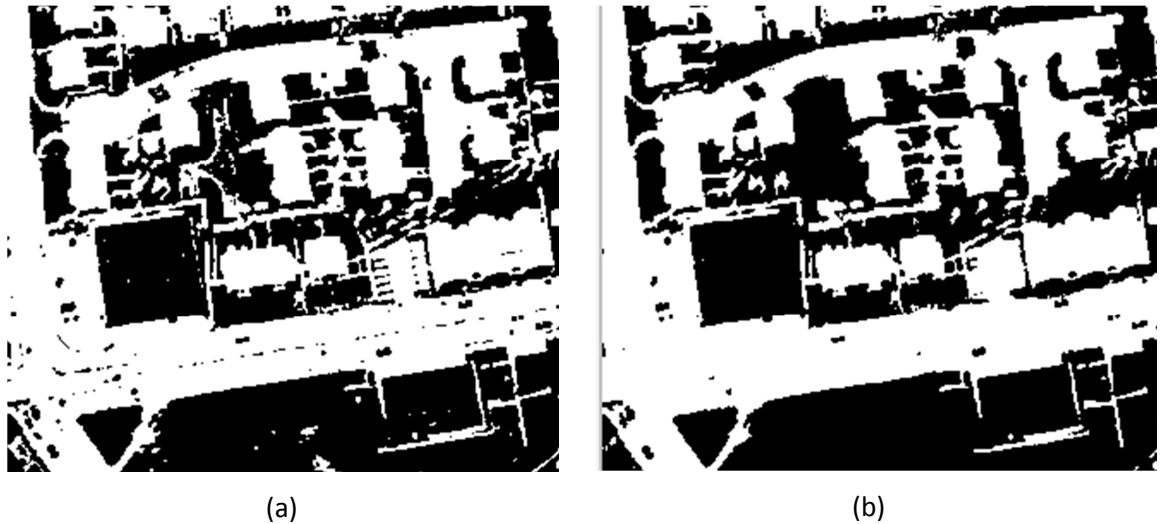


Figure 7.4 Thresholded image using global thresholding method(a) thresholding the enhanced orthoimagery using minimum error thresholding (b)thresholding the enhanced orthoimagery after applying the mean shift algorithm, the noise has been removed and the holes have been eliminated.

Then, the next stage, after applying the Mean-Shift Algorithm, is starting the segmentation, using the global thresholding method of Kittler and Illingworth (1986), namely their Minimum Error Algorithm. This method is based on the idea that it is possible to discriminate between the object and image background values utilizing the histograms of the image's grey intensity values. It is assumed that these histograms, for both object and background pixels, are normally distributed. Thus, a threshold value (or grey-level boundary value between the segments) is obtained in the overlap area of the two distributions (i.e. foreground and background pixels). Initially an arbitrary threshold value is assumed, later on the value is changed iteratively, thus, at each iteration, the histogram for the foreground (i.e. objects) and background is calculated, consequently an overlap is obtained from intersecting the tails of these two histograms (the normal distribution tails). Eventually, the value of the thresholded that gives the minimum the overlap area will be select as the optimized threshold value. The segmentation method used was able to detect the buildings and the roofs, however it was not able to detect the buildings that had high intensity greylevel values due to either their high reflectance, or that they contrasted highly with their surroundings, see Figure 7.3. Therefore it was necessary to use an additional threshold, determined using the Moments Algorithm (Tsai, 1985).

Moment thresholding (i.e. the basis of the Moments Algorithm, referred to above) is based on determining the mathematical moment for the greyscale image and comparing it with different values of the binary images produced from different values of thresholding. The moment is calculated deterministically using the equation 7-5. The value of the threshold

will be based on selecting the value that gives closest image moment to the original greyscale image.

$$m_i = \sum_{j=0}^1 p_j(z_j)^i, \quad i = 1, 2, 3, \dots \quad 7-5$$

Where:

m the image moment

i is the moment order

$p_j = n_j/n$, is the probability of the grey level and it is equal to

n_j is the number of pixels that has the value z_j in the image.

n is the total number of the pixels

j is the intensity value of the pixel

z_j pixel grey value

Moment thresholding can be considered to be an image transformation tool, and helps to retrieve the real image from the blurred image. With the moment threshold, the flat roofs were not detected, until an inverse image was produced, with a white background instead of a dark background, in order to make the objects visible (see Figure 7.5).

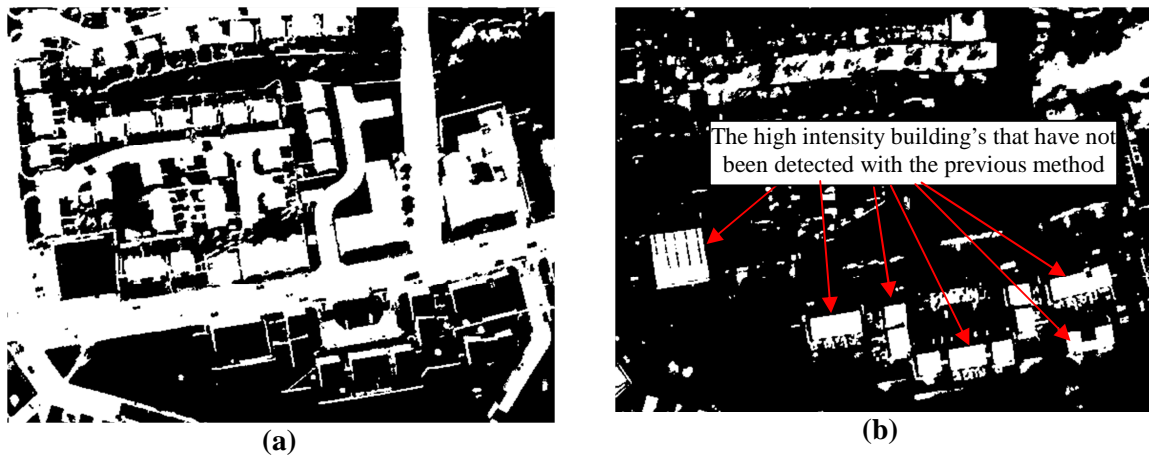


Figure 7.5 Segmenting orthoimagery using different thresholding algorithms: (a) Minimum Error threshold, the buildings and roads are detected, except some buildings with high intensity; (b) Moments threshold, the high intensity buildings are detected, which had not been detected with the method used in the Figure 7.5(a).

7.2.2 Producing nDSM

As explained in section 7.2 the main input data are DSMs and orthoimagery; they represent the basic data for the building footprint extraction. For example as well as providing height information, the DSMs are used for nDSM production, bringing all the buildings to one level ground surface. According to the literature (Dorninger and Pfeifer, 2008; Kocaman et al., 2006; Ma, 2004; Yang and Lin, 2009) an nDSM is produced by subtracting the DTM from the DSM, which leads to all the buildings being on one level ground surface. The

DSM is the digital ground representation including the buildings and trees and the DTM represents the bare terrain; this subtraction leads to a model that contains just buildings and trees (and possibly some noise arising from small features such as cars, bushes and bins) on one level ground surface, or 'zero datum'. However, in the absence of a DTM, Brunn and Weidner (1997) implemented a MM operation to obtain an approximate nDSM for building extraction purposes.

In order to make this research general and due to possible non-availability of any satisfactory DTM for this subtraction, an alternative method is followed to produce the nDSM. This alternative method commences by convolving the DSM using a 49x49 kernel with a standard deviation of 10 pixels, which leads to smoothing the buildings' heights and particularly their edges. The reason for using a 49x49 kernel is necessitated by the range of building sizes. The kernel that was used in the convolution was Gaussian see Figure 7.6 (a) and (b), with values in the kernel taken from that distribution using equation 7-6. If a small standard deviation (e.g. $\sigma=5$ pixels, see Figure 7.6(a)) was used this produced a kernel that had an effect on the middle part only without any significant effect on the edge parts of the kernel, because it had a very narrow bell shape. However, higher standard deviations, as used in this research to produce nDSM, (i.e. $\sigma=10$ pixels, see Figure 7.6(b)) created a wide bell shape which consequently had an effect on the outer parts of the kernel, and was better than $\sigma=5$ pixels. Therefore the kernel with $\sigma=10$ pixels has been selected in the smoothing process so it can be used in the nDSM production algorithm. To produce the nDSM, the smoothed DSM is subtracted from the original DSM, see Figure 7.6(c) and(d), leading to an nDSM see Figure 7.6(e), which, ideally includes the buildings and trees only, but also, possibly, some noise arising from small features.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2 + y^2}{2\sigma^2}\right)} \quad 7-6$$

where:

$G(x,y)$ is the pixel value at kernel point x,y

σ the standard deviation or parameter of smoothing

Concerning noise arising from small features, Kittler and Illingworth (1986) addressed this problem using their Minimum Error Thresholding Algorithm. This was implemented, and the nDSM was classified into background/foreground classes resulting in a 'background' class containing the ground and small features and a 'foreground' class containing buildings and trees. If the original DSM only had been thresholded it would have merged

the buildings placed in valleys or lower down hillsides with the highest elevation areas, see Figure 7.6(f), therefore it was necessary to produce the nDSM.

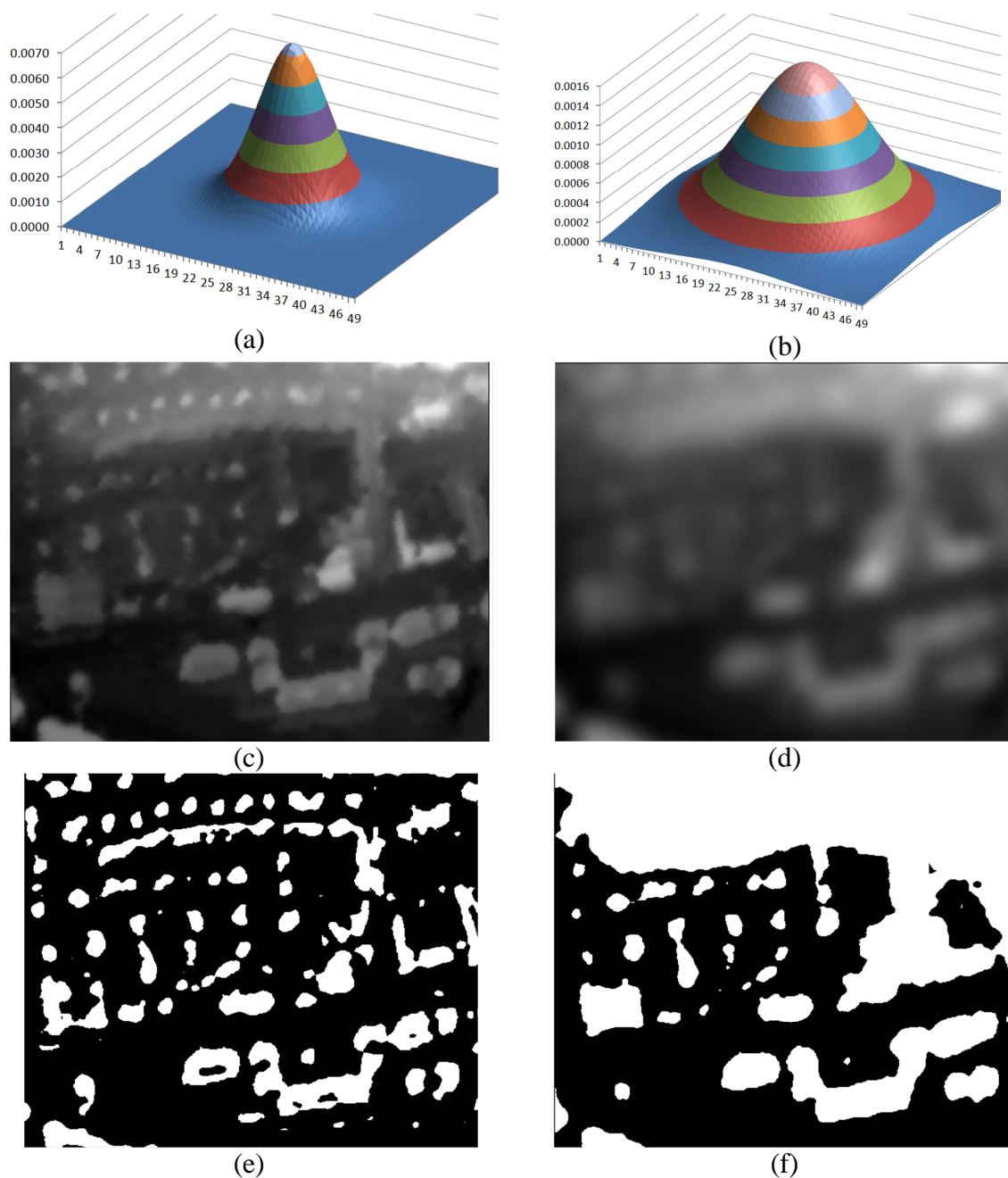


Figure 7.6 Process of producing an nDSM using MM: (a) 3d representation of the kernel, size 49x49, that is used in the smoothing with $\sigma=5$ pixels; (b) 3d representation of the kernel, size 49x49, that is used in the smoothing with $\sigma=10$ pixels; (c) DSM produced from merging satellite images; (d) applying a 49x49 kernel to the original DSM with $\sigma=10$ pixels; (e) thresholded nDSM - the top of the buildings have been detected; (f) thresholded DSM, the buildings on the upper left has been merged with higher elevation ground.

The 49x49 kernel leaves gaps and holes in the large buildings which can be overcome, see Figure 7.7, at 'a' for example, while it is ideal for small buildings, see Figure 7.7 'b' and 'c'.

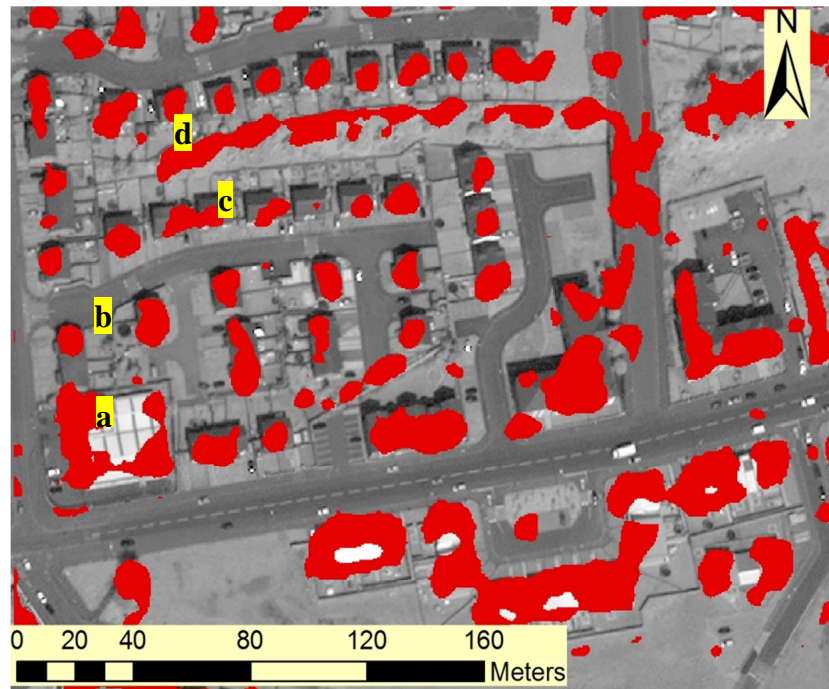


Figure 7.7 Overlaying the thresholded nDSM on orthoimagery: (a) a small part of the building has been detected in the nDSM; (b) nDSM has covered over two houses; (c) a part of the nDSM covers the building partially; (d) a green/vegetated area has been detected.

7.2.3 Uniquely Labelling nDSM Buildings

The nDSM for the study area contained more than one building, thus it was necessary to label each building individually. The final stage of nDSM production is, therefore, labelling the regions in order to represent each individual building. The labelling results in giving an integer tag to each group of pixels.

The labelling used the Connected Component Algorithm, described by Di Stefano and Bulgarelli (1999). The basic idea behind this algorithm is clustering the connected pixels and assigning an integer label to each cluster. It can be applied on grey scaled or binary images, but only binary image labelling is used here.

The labelling process began by labelling each cluster of a group of pixels connected to each other. This was achieved by scanning the binary image, to find the joined pixels, by examining their values (i.e. whether the pixels belonged to the foreground or the background).

The scanning process works from top to bottom and from left to right. The notion of this algorithm is to scan the binary image twice, passing a kernel over each pixel. The aim of the first scan is assigning provisional labels to each pixel based on their location and linking to the cluster structure. This labelling is done by traversing a mask, see Figure 7.8(a), over each pixel in the image, (see Figure 7.8(b)). The reference point $P(i,j)$ will be a foreground pixel, then, first it will check its left neighbour pixel $P(i,j-1)$ and upper neighbour pixel $P(i-1,j)$. If none of the neighbour pixels is foreground then it will assign a new label to the reference pixel. Otherwise, in the case, that one of them is foreground then it will take the label of that pixel. If both of them are foreground then it will take the label that has least value, and will save the label value that has the higher integer. Iteratively, it will continue until all the pixels in the image are scanned.

It is clear that some of the objects acquire two labels (see Figure 7.8(c), left object), therefore it is necessary to correct them. For that purpose a second pass is required; the objects that have more than one label will be adjusted using the data structure that has been saved in the first pass. The provisional second label values are replaced by the lowest provisional label value and thus each disjoint region will have a unique value, see Figure 7.8(d).

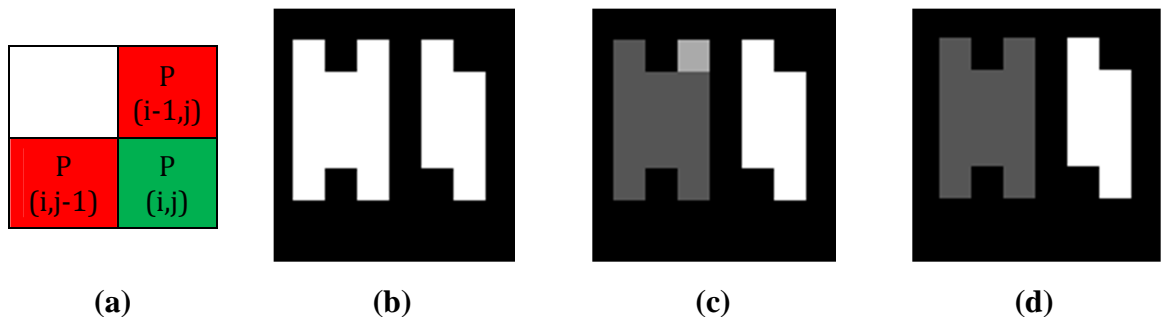


Figure 7.8 Object labelling process. (a) The mask used to label the region. Green is the reference pixel - a foreground pixel and red pixels are the connected neighbour pixels, used to check the connectivity of the pixels. (b) The original binary image with objects before labelling, both the objects having the same label. (c) Applying first pass of labelling, first object to the left has two labels, while the right has only one label. (d) Applying the second pass of the labelling on the object, the left object has only one label and the right object takes no different label.

The algorithm has been applied successfully on the nDSM binary image, and each object has taken a different label (see Figure 7.9) which can be used to identify each building individually for the purpose of rudimentary boundary extraction.

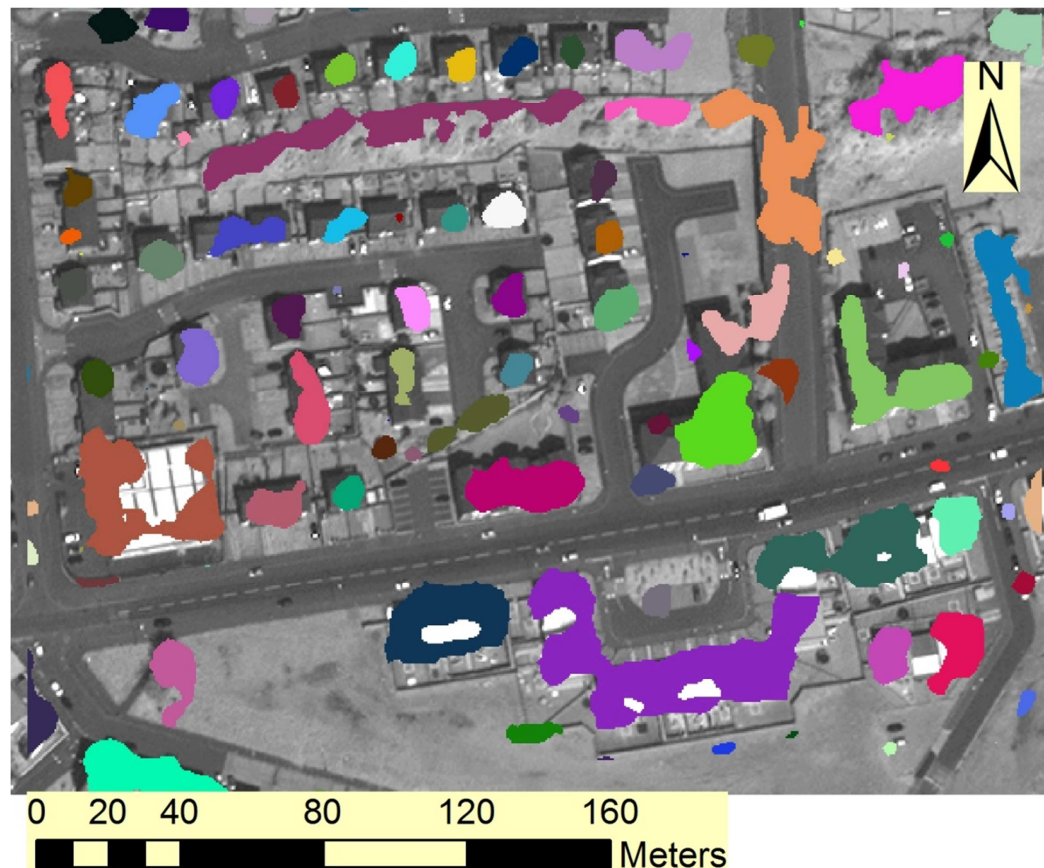


Figure 7.9. The result of labelling nDSM: a different colour value is given to each different object label, for better visualisation.

7.2.4 Shadow Detection

In the segmented image, the shadows are detected together with the buildings, due to the close similarity of texture values. The detected shadows were incorporated with the buildings, consequently adding additional parts to them. Since the detected buildings are used as initial building boundaries, it was necessary to eliminate the shadows in order to retain just the original building. The elimination process has been achieved by first detecting, then extracting the shadows, and then, later on, subtracting the shadows from the segmented image.

Different researchers have looked at the shadow problems in images. For instance Tsai (2006) proposed a method for shadow detection and its replacement by ‘foreground’ area using shape data. In the study reported in this thesis it has not been necessary to manipulate the shadows; it was enough just to detect and eliminate the shadows. The method which is followed here for panchromatic images, is similar to those of Yamazaki et al. (2009) and Liu et al. (2011). It is carried out by using a threshold value for shadow, by classifying the image into shadow and non-shadow area. It is based on the idea that pixel values less than the specified threshold are shadow and anything higher is non-shadow. The threshold value has been obtained by inspecting the image in shadow areas interactively, in the

orthoimagery, (see Figure 7.10(a)). The selected pixel greylevel value for shadow is 26 or less. This leads to detecting the shadows (see Figure 7.10(b)). The detected shadows initially consisted of discrete pixels from which regions were produced and contained many holes, for that reason Gdc (2008) suggested using the “opening” process of MM (dilation and erosion), see Sec.7.1. This process helped to join the pixels together and close the holes, consequently producing a solid object that represented each shadow Figure 7.10 (c).

The extracted shadows lead to a newly constructed shadows map. This new shadows map can be used as a mask, to subtract shadows from the segmented image, consequently leading to a new image with clearly identified boundaries, closer to real buildings. The intention of shadow detection in this research is just to rectify that error in segmented images. Therefore, it was not necessary to implement very accurate methods for shadow detection, because later on the Canny edge map is used to regularize the boundary of the building footprints.

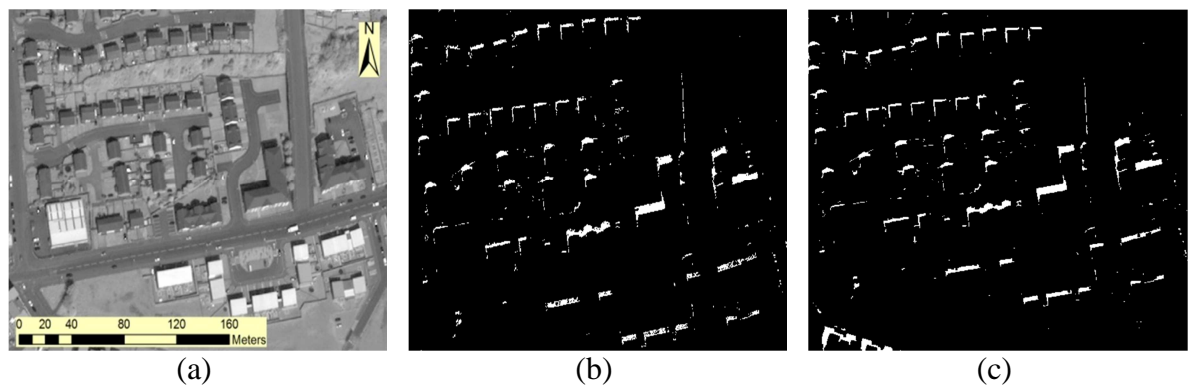


Figure 7.10 Detecting and subtracting shadows (a) orthoimagery shows the shadow area (b) the detected shadows from the orthoimagery, (c) the refined shadows after applying Mathematical Morphology, leading to the production of solid objects.

7.2.5 Road Extraction

The texture similarity of buildings and roads has encouraged an examination of building detection in the segmented image using the orthoimagery, in particular the segmented image resulting from applying a Minimum Error threshold, (see Figure 7.5(a)). The algorithm used causes the roads to be detected as buildings. Therefore, it was necessary to eliminate the roads from the segmented orthoimagery. Many different algorithms have been used to detect roads, some successfully, either from aerial images or satellite images (Ahmed and Rahman, 2011; Bacher and Mayer, 2005; Benkouider et al., 2011; Christophe and Inglada, 2007; Gecena and Sarpb, 2008; Hu et al., 2007; Kim et al., 2004; Liu et al., 2003; Long and Zhao, 2005; Ye et al., 2006; Yuan et al., 2009) but in this case, as road detection is not a focus of this research, the roads were initially extracted following their

manual digitizing using ArcGIS software (see Figure 7.11), and then the road pixels were subtracted from the segmented image. It is expected that an available automated algorithm can be implemented, by the author in the future, to do this.

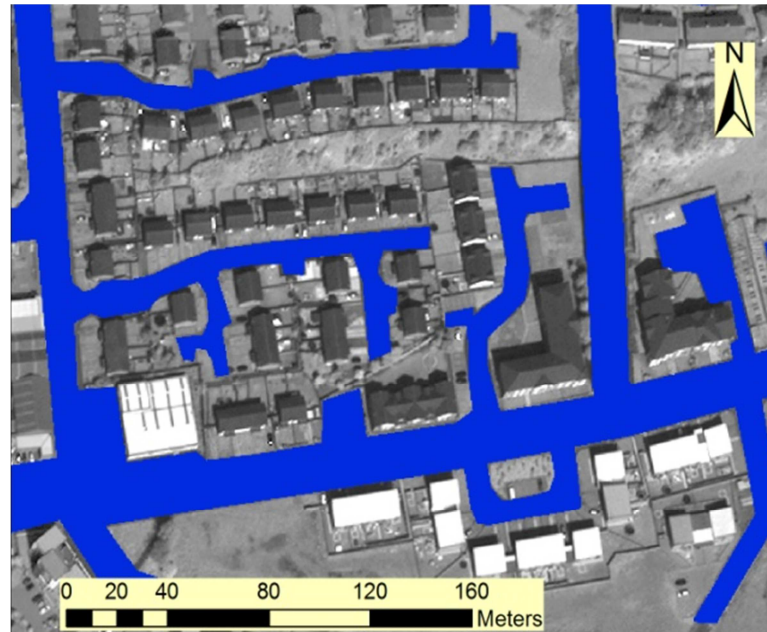


Figure 7.11 the digitized road using ArcGIS software.

7.2.6 Vegetation Area Masking

The DSM used to start the process contains all the objects above the ground, including building and non-building objects (such as cars and trees); all of them have height above the ground. Concerning the lower non-building objects such as cars, they can be removed by adding a height threshold to the nDSM. However, as tree heights are close to building heights it was necessary to find another way to discriminate them from the building objects, so they could be eliminated later. Different indices have been developed to indicate built up and vegetation areas such as Normalized Difference Vegetation Index (NDVI), Normalized Built-up area Index (NBI), Normalized Difference Impervious Surface index (NDIS), Normalized Difference Built-up Index (NDBI), Enhanced Built-Up and Bareness Index (EBBI), and they are described in general texts (e.g. Waqar et al., 2012). Studies, such as those by Waqar et al. (2012) and As-syakur et al. (2012) indicate that the above indices have the ability to discriminate built-up areas and vegetation. In addition to these, the study by Li and Liu (2008) advocates NDBI for finding the built-up areas, because it shows a stronger correlation between NDBI and land surface temperature, than NDVI, which leads to the NDBI giving more accurate result for the vegetation area. However, the images used in this research are pansharpened images, produced by using 50cm panchromatic images with 2m multispectral images from Pleiades having spectral bands Blue, Green, Red and Near-infrared. Specifically the pansharpened image has the

following spectral bands: panchromatic - 470 to 830nm; blue - 430 to 550nm; green - 500 to 620; and, near-infrared - 740 to 940nm (Astrium Services, 2013). It is clear that there is no shortwave infrared: 1550-1750nm which is necessary to compute the NDBI (As-syakur et al., 2012), see equation (7-7). For that reason, it was not possible to calculate the NDBI. Under the circumstances NDVI, is the only index that can be calculated, according to Deng & Wu (2012), using Equation (7-8).

$$NDBI = \frac{R_{SWIR} - R_{NIR}}{R_{SWIR} + R_{NIR}} \quad 7-7$$

$$NDVI = \frac{R_{NIR} - R_{red}}{R_{NIR} + R_{red}} \quad 7-8$$

The NDVI values ranges between -1 and +1; high values represent the vegetation area and the low values represent the built-up area. In order to segment the NDVI map, the Minimum Error Threshold method is suitable for segmenting the NDVI images into vegetation and non-vegetation areas. This method can be used to extract those vegetated areas identified in the NDVI map, (see Figure 7.12). The candidate thresholding method, using the Minimum Error Threshold, has been examined qualitatively, by the author, and it was found that it gave better results than other automatic threshold methods (i.e. Moments Algorithm (Tsai, 1985)). The nominated method distinguished between buildings and vegetation, however some of the smaller examples of vegetation have remained undetected, probably because the resolution of the Pleiades multispectral bands at 2m was resampled from the original 2.8m resolution, see sec 3.3.3.

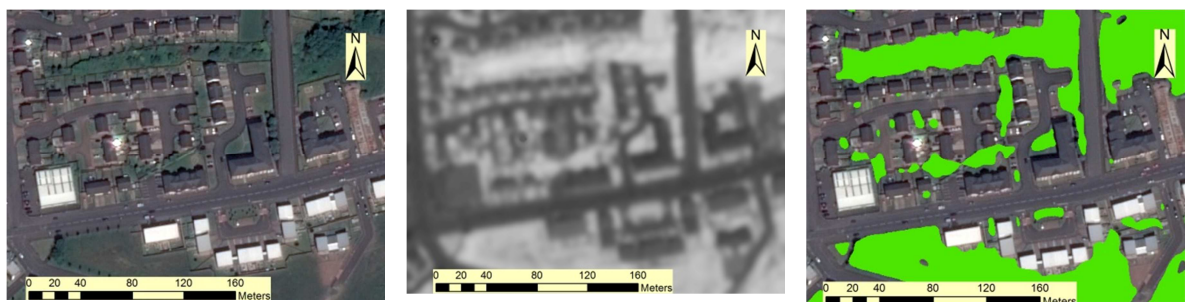


Figure 7.12 Green Mask production (a) Pleiades sensor's imagery including multispectral and infrared bands (b) NDVI map produced from using NDVI index (c) the vegetated (green) area which results from applying Minimum Error threshold on the NDVI map and overlaid over the Pleiades imagery.

7.3 Producing an Edge Map

Initially it was expected that segmentation could be followed by building extraction using MM. The segmented images show the buildings clearly (see Figure 7.4), but, the MM process, when implemented initially caused the buildings to lose their spatial resolution, because the operation is based on using circular and square structural elements, see

Figure 7.2(a) and (d). Therefore, it was necessary to improve the boundary definition of the buildings using edge mapping with Bayesian optimization. The selected operator used to produce an edge map is the Canny edge detector (Canny, 1986).

Based on the literature, it appears that the most powerful algorithm for edge detection in image processing and computer vision is Canny edge detection. Although it gives the best result if compared to the other known edge detectors such as Sobel, Robert cross and Prewitt edge detectors, its high response to noise is considered a drawback (Cardoso, 1999; Yang-Mao et al., 2008). Regardless of this downside, Canny edge detection is recognized as having the practical features that it can detect edges down to one pixel width and it has the ability to join scattered edges together (Cardoso, 1999; Yang-Mao et al., 2008). The Canny edge detector is known to be an ideal algorithm in low-level computer vision (i.e. detecting edges and corners in the image) for finding the main geometric features from the images (i.e. lines and corners) (Liu et al., 2008). However, as mentioned, it is influenced by noise and therefore a lot of artefacts are detected (Barry, 2011; Yang-Mao et al., 2008). These artefacts, consequently lead to incomplete buildings and inaccurate building boundaries, if, as is usual, the extracted edge refining the approximate boundary of a building has been obtained from segmenting orthoimagery using a Bayesian method. This will be discussed further 7.6.3(b).

The Canny algorithm uses several parameters and the success of edge detection depends on the good selection of these parameters. The algorithm is summarised in

Figure 7.13.

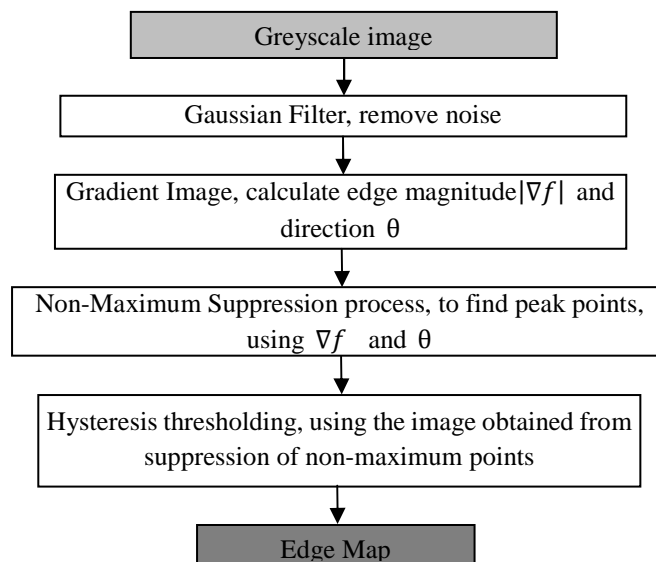


Figure 7.13 Flow chart for Canny edge detection process.

and works as described in the following paragraphs.

First, smooth the original image using a Gaussian Filter (equation 7-6). The purpose of this smoothing is to remove image noise. Because noise causes high gradient magnitudes it consequently produces edges that are not required. Canny (1986) greatly extolled the effect of the Gaussian kernel in reducing noise (Nixon and Aguado, 2008). The sigma (σ) value, in the Gaussian kernel, has a noticeable influence on image smoothing as does window (or kernel) size. If a window size is large, it will result in extra-smoothing, causing frail edges to be removed. Consequently, therefore, a smaller window size is recommended and is less expensive computationally. In summary a window size of 5x5 or 7x7 pixels is recommended (Worthington, 2002). If I is the image and $g(x, y)$ is the Gaussian filter then the smoothed image, Equation 7-9, $f(x, y)$ will be:

$$f(x, y) = I * g = \sum_k \sum_l I[k - x, l - y] * g[x, y] \quad 7-9$$

The new x,y pixel values in the image are calculated based on multiplying each pixel in the image with its corresponding part in the kernel. (It is referred to as a discrete convolution, i.e. the elevations are in grid format.)

Second, find the gradient values, G_x , and G_y , of the smoothed image at location (x,y), in both x and y directions respectively for each point. These are obtained, in both x and y directions by using the Sobel filter (Nixon and Aguado, 2008), see Figure 7.14. After



Figure 7.14 The filters used to produce gradient map (a) gradient in x-direction (b) gradient in y-direction (Nixon and Aguado, 2008).

getting the result, which is two gradient images, these will be used to calculate the magnitude of the gradient (its vector strength) and the direction of the gradient, which are considered the most important values in edge detection, using equations 7-10 and 7-11 (Nixon and Aguado, 2008).

$$|\nabla f| = \sqrt{G_x^2 + G_y^2} \quad 7-10$$

$$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad 7-11$$

where:

∇f represents the gradient value of the image at location (x,y);

G_x and G_y or $\partial f / \partial x$ and $\partial f / \partial y$ represent the gradient at location (x,y) in the x and y direction (using the preceding pixel in the x-direction and the preceding pixel in the y-direction), respectively; and

θ represents the direction of the gradient.

The third stage is non-maximum suppression. It has been shown that the Sobel operation was able to detect edges, but a problem was that the edge obtained contained an extensive range of points along the exact edge point, consequently it was a very irregular and thick representation of the edge in the image (Ali and Clausi, 2001). Therefore, it was necessary to increase the quality of the detected edge and limit it to only one pixel width. Consequently, the Canny edge detector became very popular. Non-maximum suppression detects points at the highest location of a ridge and neglects all the points beside these. Instead of being used to analyse ridges, it can be applied in edge processing. The principle of non-maximum suppression is explained in Figure 7.15 (Nixon and Aguado, 2008). First by checking a pixel, p, in a 3x3 window, then the values of the points p1 and p2 along the perpendicular of the edge direction at p will be calculated, by using interpolation from neighbour pixels. If the value of the pixel p is greater than p1 and p2 then pixel p will be labelled as 'maximum' and consequently as an edge. If not, it will be set to zero. Consequently, this operation preserves only the highest points on the edge (or ridge as originally implemented).

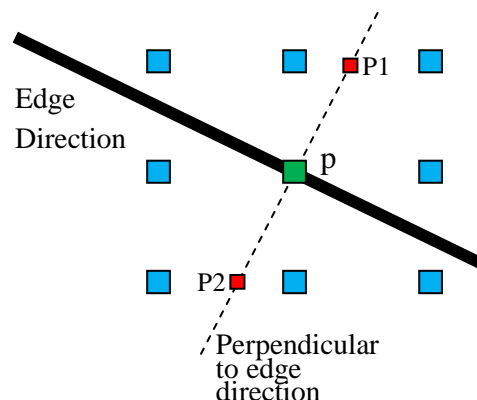


Figure 7.15 Illustrating non-maximum operation process (from Nixon and Aguado,

2008).

Finally, the image that is obtained from the suppression of non-maximal gradients contained only the peak along the ridges, representing either a weak or a strong edge. At this stage, the weak edges (small intensity change) will be eliminated through testing the edge points against the given thresholds, thus the remainder will be only the edges at the locations of large intensity change. In this operation two thresholds will be defined the upper and the lower thresholds. The operation that is used to keep the high intensity change edges while removing weak edges is called hysteresis thresholding. Figure 7.16 (based on Nixon and Aguado, 2008) shows how the hysteresis process is achieved on the image that is obtained from non-maximum suppression stage. First, the points that have magnitude above the upper threshold are set as a strong edge and given a value equal to 255, while the points below the lower threshold are considered as weak and set to zero. The hysteresis operation starts by first taking any point that is above the upper threshold and use it as a seed point. Next, the neighbours of that seed point will be set as an edge point if their value exceeds the lower threshold, then they are considered as seed points for other neighbour points. In this process, the investigation will be continued along the neighbour points until all the points are visited.

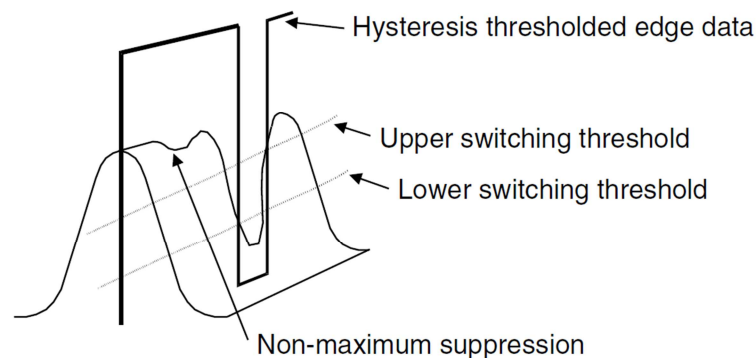


Figure 7.16 Illustrating non-maximum operation process with hysteresis thresholding (from Nixon and Aguado, 2008).

The flow chart of edge detection using the Canny approach was summarized in

Figure 7.13. The result of Canny edge detection is an edge map and among the edges we can find the building boundaries that will be used in the optimization process, see Figure 7.17.



Figure 7.17 producing edge map (a) the study area used to produce Canny edge map (b) result of applying Canny edge detection on ‘a’, using Gaussian kernel radius=1, low threshold=1 high threshold =2 in the study area

7.4 Building Detection

Before building extraction and construction, building detection is implemented. In this stage, the locations of the buildings are identified based on using the labelled nDSM and the segmented image, to deliver the buildings for the next stage.

7.4.1 Enhancing Segmented Image

The segmented image, either produced from the Minimum Error or the Moments algorithm and used to extract the building boundaries, has been refined by subtracting the roads, shadows and vegetation (using NDVI) from it. In addition to containing many irrelevant objects attached to the buildings, due to the segmentation process, the image also contains much inherited noise arising from the image refining subtraction process. These irrelevant objects and noise were causing false object representation, therefore, it was necessary to enhance the image and remove them prior to embarking on detecting the buildings and starting building footprint extraction.

In this enhancement, the refined segmented image that is obtained by using the Minimum Error algorithm has been used first. Two types of structural elements have been utilised, one circular, 5x5, and the other square, 3x3, (see Figure 7.2(a) and (d)).

Both the structural elements were used to achieve opening, but in a different order. First, the circular structure was used to achieve erosion using equation 7-1; this lead to simplifying the image, see Figure 7.18 (b). Later, the square structural 3x3 element was

used, twice, to achieve erosion followed by applying dilation, twice. Finally dilation was applied using the circular structural element to restore the spatial losses arising from erosion, in order to restore the rudimentary object, see Figure 7.18 (c) and (d). The reason for alternating between both structural elements arises because the circular element was able to remove the extra parts from the shapes of interest, as could the square structural element, but the square element was not able to break the links between the objects. The reason being that the square structural element, at 3x3 had a smaller size, and it was effective only on the objects that were parallel to the structural element. In addition, if a larger square structural element was used, then that caused excessive building erosion with consequent shape loss. It is important to retain the building shape as much as possible for the subsequent Bayesian processes and for edge enhancement.

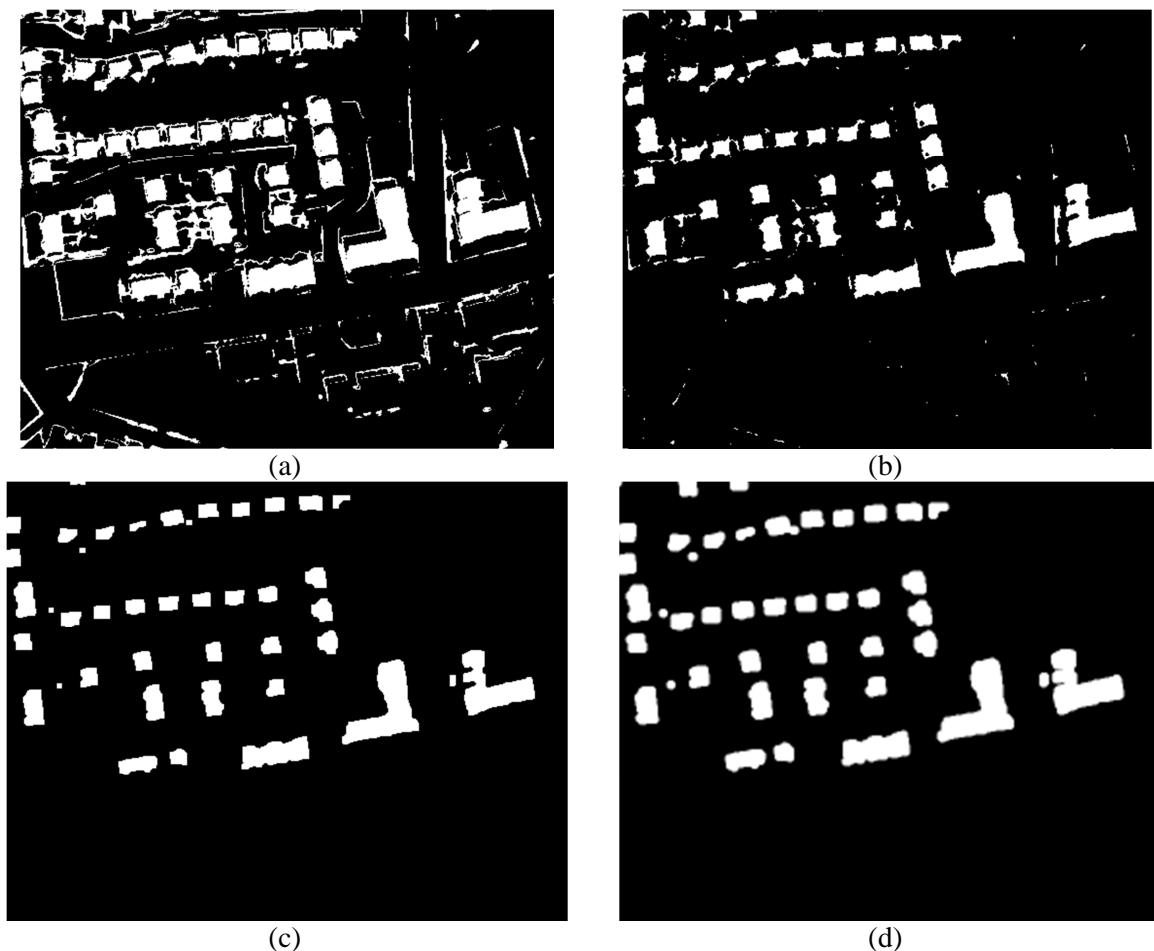


Figure 7.18 Process of refining the segmented orthoimagery which is produced from applying Minimum error thresholding (a) the output of subtracting the roads NDVI and shadows from the first segmented image which is obtained in sec 7.2.1 (b) Image erosion applying circular structural element (c) Using MM opening operation, apply structural element erosion twice and dilation twice, using square structural element (d) Image dilation applying circular structural element.

Regarding the segmented image, this has arisen from the processing of a high contrast image. This high contrast image has been produced following Moment thresholding,

subtraction of road, shadow and vegetation (using NDVI) and also MM treatment. It was clear that the square structural element was enough to remove the noise from the image by applying erosion twice followed by dilation twice. The result of the final image after applying MM is shown in Figure 7.19(a) and (b).

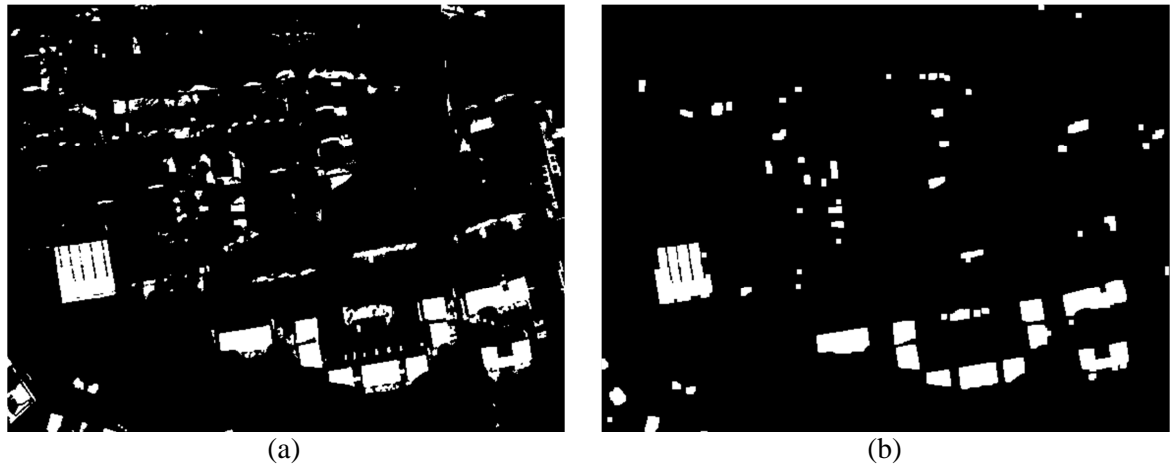


Figure 7.19 Process of refining the segmented orthoimagery which is produced from applying moment thresholding: (a) the output of subtracting the roads, NDVI and shadows from the second segmented image; (b) using MM opening operation, apply structural element erosion twice and dilation twice, using the square structural element, the noise has been eliminated and the objects separated.

The object segmentation of the binary image arising from the MM processing seemed promising. It could provide an initial building boundary, which is very important input for the Bayesian method.

7.4.2 Detect Building Boundary

In order to implement the buildings extraction process, it was necessary to detect the buildings in the image. This process has been carried out, first, by taking each object individually in the labelled nDSM, which was described earlier in section 7.2.3. It has been assumed that each label represents an object that has height - either a building or a tree. The trees have been eliminated by using the vegetation mask, and the remaining objects represent only the buildings.

In order to start building detection, each building is taken individually from the labelled nDSM. Later, the parts for the corresponding building are taken from the enhanced segmented image described in section 7.4.1, in order to produce an initial building boundary. However, the buildings in the labelled image are not complete and just represent a part of the building, consequently the initial building boundary will be imperfect (see

Figure 7.20 (a)). Therefore, it was necessary to increase the size of the detected building in the labelled image. The size is increased using the dilation operation that is explained in section 7.1.2, by dilating the specific object $\delta(X)$ 17 times, using a circular structural element. The number of dilations has been concluded experimentally.

As mentioned in section 7.4.1, two segmented images have been produced, due to differences in building texture, therefore it was necessary, before extracting the initial building boundary, to decide which of the segmented images better contains the building - either image A or B. The rule that is used is based on measuring the image areas that best correspond to the labelled areas in the nDSM. The area under the labelled nDSM is measured in both segmented images, and that giving the maximum area will be considered further.

The detected buildings at this stage are not limited to single specific buildings, but may cover parts of other buildings. Therefore, it was necessary to eliminate other areas by using 'common parts'. If there are any common pixels between the labelled nDSM and the segmented image then those common parts belong to a specific building, otherwise the pixels can be eliminated as belonging to other buildings, the process is illustrated in Figure 7.20.

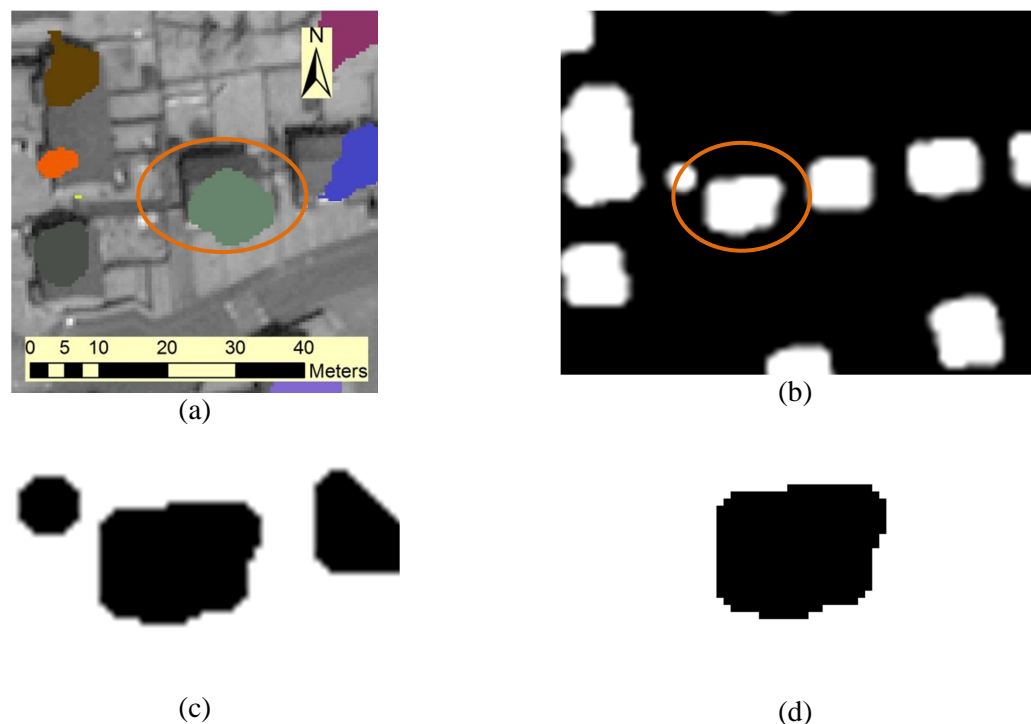


Figure 7.20 The process of building detection. (a) The labelled nDSM overlaid on orthoimagery. (b) The segmented image. (c) The remaining corresponding part of the segmented image after dilating the labelled nDSM image. (d) The object that will be used in building extraction.

7.5 Building extraction

Beyond building detection the boundary of the buildings are marked out giving the initial building boundary. This stage in the flow line concerns delineating the building footprint. Earlier, to transfer the initial building boundary to the final stage that consists of building boundary construction, initial building boundaries are vectorised and approximated in order to make them ready for the final stage.

In this section 7.5 and its sub-sections (7.5.1 – 7.5.3) and in section 7.6 (and its sub-sections 7.6.1 - 7.6.3) a number of boundaries are generated and processed. These are listed below and briefly defined. However, the reader will also find a fuller explanation of their meaning when they are introduced at the relevant points of the discussion.

- Initial Building boundary: the boundary, in pixel format, that is obtained after detecting the building in the segmented binary imagery.
- Vector boundary: converted initial building boundary from pixel format to vector
- Approximate initial building boundary: approximates the above, vector, boundary by minimizing the number of points at the border.
- Simulated boundary: the boundary generated by adding random numbers to the corner of approximated initial building boundary.
- Building footprint: the boundary of the building that is located under the roof.
- Individual base boundary: the remaining part of the edge map at the location of the building after removing the other pixels.
- Constructed individual building boundary or Solo boundary: the result that is obtained from the individual base boundary.
- Canny edge map: the obtained result from applying canny algorithm on orthoimagery.
- Modified edge : the enhanced edge map that is obtained from applying MM
- Regularized boundary, Optimised boundary or accurate building boundary: the best result of obtained boundary after applying Maximum Likelihood and Bayesian approach, this represents the most accurate building boundary that is obtained from applying the proposed algorithm.

7.5.1 Extract initial Building Boundary

The resulting image, which is produced from taking the corresponding segmented image in the previous section, contains the specific object that represents the building in image space, see Figure 7.21(a). At this stage, the initial building boundary of the building can be extracted using MM, as described in the next paragraph

The process for the initial building boundary extraction is as follows. First erode the image then subtract the resulting image from the original image X ; the whole operation can be represented in equation 7-12. The erosion has been achieved by using a square structural element 3×3 , and gives a promising result.

$$\beta(X) = X - (X \ominus B) \quad 7-12$$

The resulting boundary, see Figure 7.21(b) will be used as an initial building boundary and the pixels will be converted into the UTM zone 30 (North) projection grid coordinate system, on the WGS84 ellipsoid, and later on connected to each other to produce a vector map. Prior to the vectorizing operation, it was necessary to sort the points because the points are randomly distributed. For that purpose the procedure described in the next section was followed, to sort the points around the initial building boundary.



Figure 7.21 The process of extracting building boundary: (a) the extracted building obtained from segmented image in the binary form; (b) the initial building boundary's pixels obtained from applying mathematical morphology operations on the binary image in (a).

7.5.2 Sorting Initial Building Boundary

At this stage, the initial boundary of the buildings is represented by pixels in a binary image and the image contains only a specific building. It was necessary to convert the initial building boundary into vectors for the simulation requirements of the Bayesian approach. For the vectorizing, the coordinates of each pixel are converted into Cartesian coordinates from the row and column value of each pixel. The sorting of the points was

based on reading the image file, from top to bottom, and from right to left. As is clear in the figure the points are not ordered sequentially around the initial boundary, but are in their pixel (reading) order (see Figure 7.22).

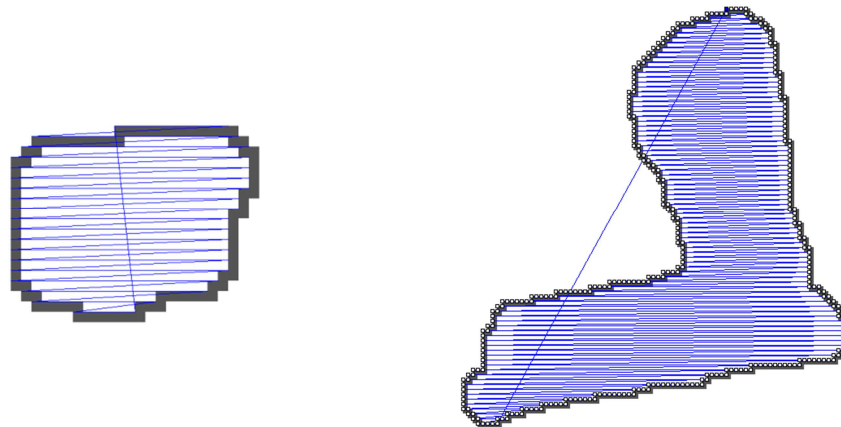


Figure 7.22 Two different types of buildings show the representation of the initial building's boundary points after converting them from pixel wise to vector. It shows the initial building's boundary sorted, based on the pixel readings.

The algorithm used for the sorting is the Nearest Neighbour Algorithm (NNA) used by Gutin et al. (2002) in the travelling salesman problem. The algorithm works as follows: first, any point is selected as the base point and is labelled as being visited; next, the distance from that point to all other points, not marked as visited, is measured. The point that gives the minimum distance will be assigned as the next point to the base point, and marked as visited. The new point is as the next base point and the procedure repeated. The process is continued until all points of the initial boundary of the building are visited.

The above method was very robust in the situation reported on in this work since the points are distributed close to each other and without gaps of varying size between them. The method has been applied successfully on buildings that are either square or L-shaped, see Figure 7.23, and without producing a convex hull shape. It is a simple algorithm which does not require additional parameters such as required by the Alpha Shape Algorithm (Bernardini and Bajaj, 1997), which needs the alpha parameter before sorting the initial boundary of the building and may produce a convex hull shape.

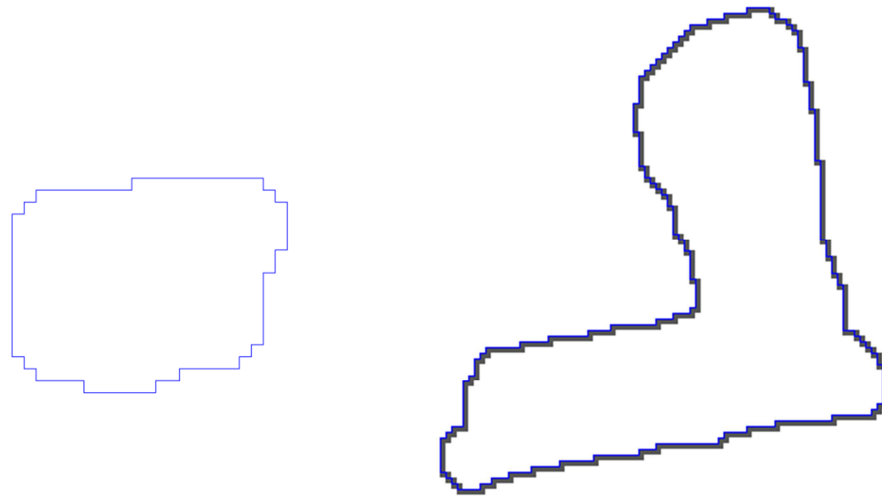


Figure 7.23 Sorted vectorised initial building's boundary vectors, obtained from applying the Nearest Neighbour algorithm.

7.5.3 Approximate Initial Building Boundary

Beyond sorting the points of the initial boundary of the buildings, it is possible to connect each coordinate to the next, to obtain the vector boundary. The initial boundary of the building obtained consists of a number of segments equal to the number of points minus one. For simulation purposes at the Bayesian optimization stage, the simulated boundary is generated based on applying random noise at the initial building boundary corners with specific variance. Later the optimized boundary based on the initial building boundary giving Maximum A Posteriori (MAP) probability is found. The large number of segments creates a cumbersome situation taking a lot of time to process, and potentially an unexpected outcome. In addition to the large number of segments representing the real continuous initial building boundary that has arisen from vectorization performed by connecting pixels, there will have been some irregularities in the initial building boundary, as a consequence of segmentation or MM.

It is necessary to join the segmented edges and remove the irregularities in the detected initial building boundary, meanwhile keeping the points of interest (i.e. the corners) of the initial building boundary. To join the segmented edges and remove the irregularities from the initial building boundary, an approximation method has been implemented, as proposed by Ramer (1972) and Douglas and Peucker (1973).

The suggested method minimizes the number of the points in the polygon based on a defined threshold and retains important bends/corners. The threshold, in this study, is 5 pixels and gives the best result, It is found experimentally that the threshold value which is

larger than 5 pixels will result in deleting important corner points while smaller values than 5 pixels will keep more points for the initial building boundary even if these are not main corner points.

Originally, the Ramer-Douglas-Peucker method was proposed for curve generalisation by iteratively reducing the number of points in the curve based on specified threshold(s). The Ramer-Douglas-Peucker method first checks the perpendicular offset distance between the vector connecting the two ends of a curve and other points that lie on the line between the end points. If the offsets are less than the threshold then their points will be deleted, otherwise it will make new vectors between the end points and the identified point that gave the maximum offset. This process can be repeated, using the end points and the identified point. This method was designed for open curves. In the first trial, the initial building boundary was treated as a curve, with terminal points $p(0)$ and $p(n)$. In the first trial, the approximation had not covered the part $p(1)-p(0)$, $p(0)-p(n)$, $p(n)-p(n-1)$. In order to apply the algorithm on the missed parts, it was necessary to operate the algorithm at different points, by assuming the curve terminal points are $p(3)$ and $p(4)$, instead of $p(0)$ and $p(n)$. It was noticeable that the second trial guarantees that the approximation has covered all the points and the result was promising, see Figure 7.24. It was also noticeable that some non-building objects have been detected with the building footprints, such as trees, but such areas were very small and after approximation consisted of just three points. Therefore in addition to the Ramer-Douglas-Peucker method a rule has been added, which is based on the number of points. If the total number of points is three or less then the initial building boundary will be neglected, based on the assumption that triangular buildings are extremely rare.

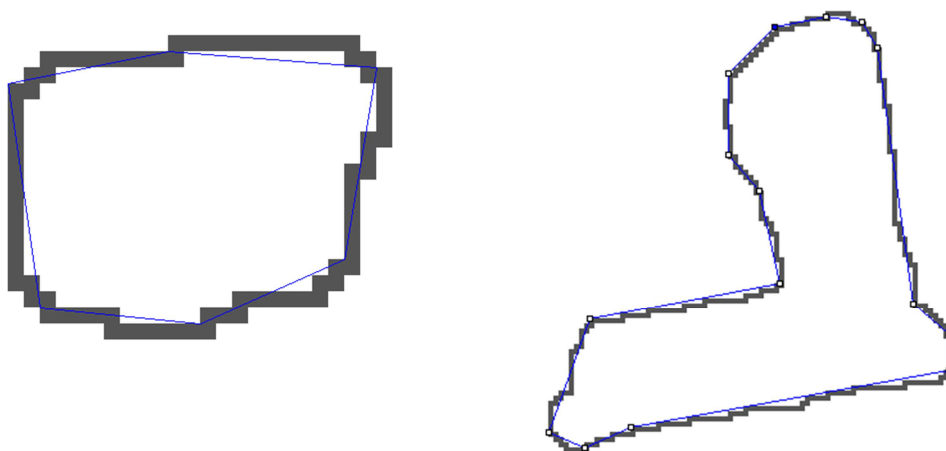


Figure 7.24 Applying the Ramer-Douglas-Peucker algorithm to approximating the initial building boundary points that were produced in Figure 7.23 and using these as the initial building boundary.

7.6 Building Footprint Construction

The extracted initial building boundary from the previous section is considered to be approximate; therefore, it needed to be enhanced. Two different methods have been used for this purpose: Bayesian and Maximum Likelihood. Moreover, the Canny edge map has been modified in order for it also to be implemented at the enhancement stage.

7.6.1 Modifying the Edge Map

So far, the building boundary that is obtained by approximating the detected boundary can be considered as an initial building boundary and it is ready to be used in an optimization process using Bayesian theory. In the optimization, a Canny edge map has also been used. The optimization process is achieved by measuring the distance between the initial building boundary line and the Canny edge map. However, the edge map, whose creation was discussed in section 7.3, contains extra edges (see Figure 7.25). These extra edges were scattered in the roof area or connected to the edge of the buildings, either inside or outside of the building. The extra edges, produced due to building roof edges, caused confusion, because the measured distances were not representing the real boundary. This situation was complicated, and it had a significant effect on the result, because it caused the boundary of the building to shrink during the simulation stage while regularizing the initial building boundary.

The Canny edge shows the location of the path where there is contrast in pixel greyscale intensity values. The parameters of the Canny edge, such as the radius of the smoothing filter and hysteresis thresholds, control the intensity of these edges.

The quantity of detected edges in an edge map can be controlled by changing the parameters, if low values are used all small details will be detected as edges and vice versa, see Figure 7.25(c and d). So if a low threshold value is used a highly detailed edge map is generated. In this situation, the real building edges will be difficult to distinguish from other edges, see Figure 7.25(c). However, if a high threshold value is used a less detailed edge map is generated, see Figure 7.25(d), then the real building edge will be lost because only the edges with a very high difference in the pixels' intensity values will be detected as an edge (i.e. the boundary of the shadows will be detected as edge and the real building will be lost). For that reason the parameters that have been chosen are mid range, see Figure 7.25(b). It can be noticed that in addition to the main building roofs some small

details have also been detected as edges (i.e. shadows and some secondary roofs) and can be considered noise.

To avoid confusion and to ensure the real edges are measured it was necessary to remove all detail created inside the building. It was noticed that the extra details were numerous in the hipped roof buildings and when some structures, such as dormer windows and chimneys, create a large variance in contrast and produce shadows on the roof these will be detected as an edge. These edges have been removed by eroding the segmented image, see section 7.4.1, then taking its boundary using MM and finally subtracting it from the Canny edge map. The problem of extra details on the roof does not exist with flat roof slabs, whereas the problem was very clear in the hipped roof building, as shown in the Figure 7.26 (a), below.

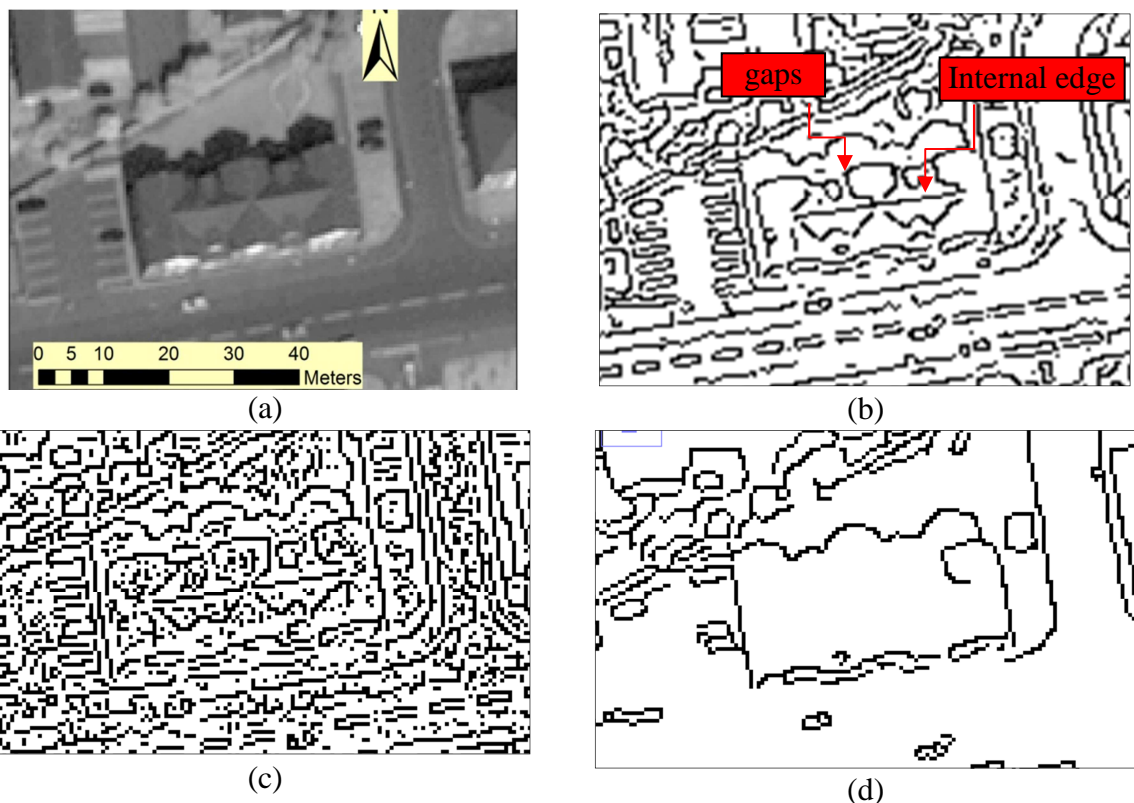


Figure 7.25 Explanation of the detected edge on the orthoimagery; (a) the orthoimagery that shows hipped building (b) the edge map which produced using orthoimagery with a lot of extra edges radius Gaussian=1, lower threshold=1, upper threshold=2 (c) the edge map which produced using same orthoimagery with very intense amount of edges, with radius Gaussian=0.5, lower threshold=0.1, upper threshold=0.2 (d) the edge map which produced using same orthoimagery with very few edges, with radius Gaussian=1, lower threshold=1, upper threshold=5.

To solve this problem the segmented image, which is discussed in section 7.2.1, has been eroded twice and then subtracted from edge map. The reason for the subtraction is in order not to remove some details from the boundary; however, this method was leaving some small amount of noise in the roof itself, although this was acceptable when considering the

result obtained. However, edge map modification was not stopped at this stage. It was necessary to perform another operation for filling gaps that existed in the Canny edge map, because the Canny algorithm failed in detecting some edges when the variation of the textures is low between two different areas, such as some (flat) slab roofs and the ground. For that purpose, the segmented image has been dilated twice, using MM. Later the boundary of each building, in the resulting segmented image after the double dilation, has been extracted using MM as well (see Figure 7.20). This operation has helped to fill the gaps that have been produced in the Canny edge map due to low contrast in the texture leading to some edges being undetected, causing a problem during constructing the individual base boundary (see section 7.6.2). Finally, the resulting map will be combined with the edge map and the result will be a new map where no gaps exist (see Figure 7.26(b)).

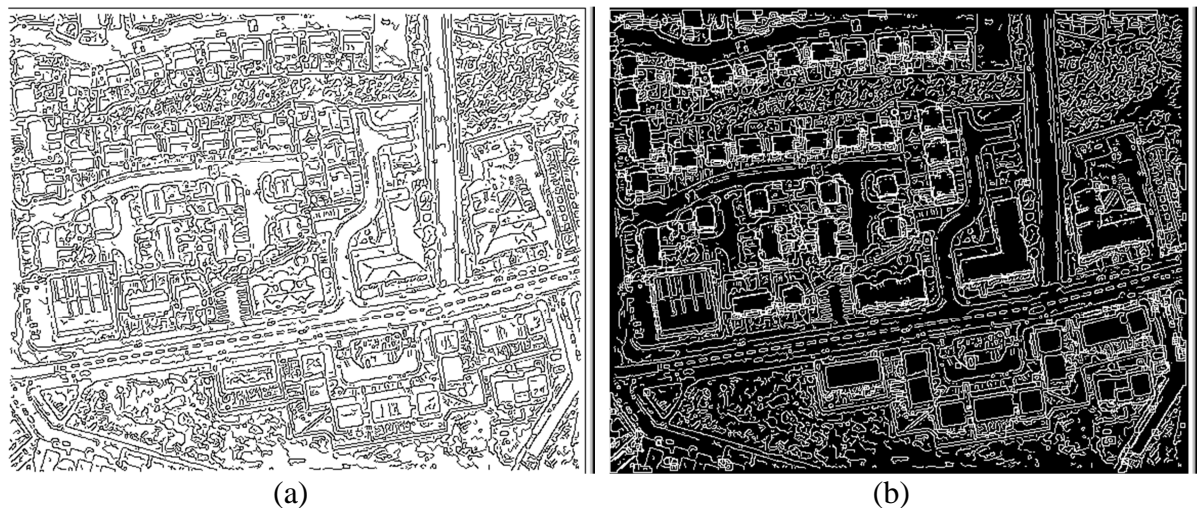


Figure 7.26 Comparisons of modified edge map (a) the original Canny map (b) the refined Canny map.

Despite removing the noise from the segmented image, still some noise (as extra edges) has been left, consequently causing problems. It was necessary to remove such noise from the edge map. At this stage, a 7x7 cleaning filter was passed over the edge map, see Figure 7.27.

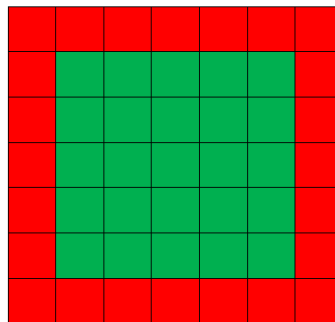


Figure 7.27 The cleaning filter which is used to remove the noise in the edge map. This filter traverses the edge map, and if all the red pixels overlies the background pixels of the edge map then any foreground pixels that are overlaid by green pixels will be removed from the edge map.

The image under the filter was examined; if the boundary of the pixel was background then the object inside the filter is removed, otherwise it shifted to the next position. Based on the idea that the foreground pixels are connected to each other, it was possible to remove the noise that was left inside the building, successfully.

7.6.2 Construct Individual Base Boundary

For the simulation purposes, it was necessary to measure the distance from each simulated boundary (obtained from adding random numbers to the initial building boundary) and the Canny edge map to be used, to find goodness of fit. However, the result was not as expected, because the distance was based on the Euclidean distance from the edge of the initial building boundary to the closest distance on the modified Canny edge map, and this distance varied during the measurement, which was consequently affecting the **argmin** result. For that purpose, it was necessary to create a new image space keeping only the building of interest and removing all the objects from the edge map except the boundary of interest, which is called solo boundary. To perform that operation, the nominated building in the segmented image described in section 7.4.1, has been dilated 10 times and then eroded 14 times to smooth the boundary and keep just the core of the building. Later on, it was necessary to minimize the number of core points, by taking the boundary of the core object. The final stage is to take each pixel from that boundary and scan around it for 360 degrees, record any intersected point on the modified edge map, and repeat this operation on all the boundary points, thus producing the solo boundary, see Figure 7.28 (c).

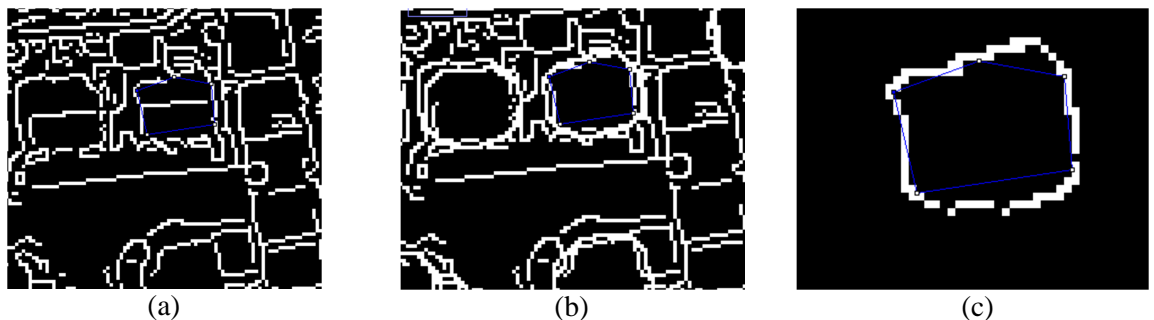


Figure 7.28 Modifying the edge map for the optimization purpose (a) Canny edge map (b) filling the gap by adding the thresholded image boundary to the Canny edge map (c) keeping the solo boundary by selecting the intersection point of the centre ray with the boundary in the image (b).

7.6.3 Building Boundary Optimization

For planning purposes, it is necessary to get an accurate building boundary with a regular shape. The initial building boundary, obtained from processing satellite image was not adequate for use in planning or city modelling purposes, due to loss of spatial resolution with the consequent effect on accuracy, and it was not sufficiently regular, due to applying

a MM operation. Therefore, it was necessary to add another operation to regularize the initial building boundary. Two methods have been tested, the Maximum Likelihood approach and a Bayesian approach. The first method depends on measuring the data while the later incorporates prior data to the measured data. The result from Maximum Likelihood was still not promising although it enhanced the initial building boundary but not with the required amount, therefore a Bayesian approach has been used to enhance the result. It was possible to gain a robust result by incorporating a proper and suitable prior in the calculation of the building footprint using a Bayesian approach. The methodology that is followed for the optimization process, in both Maximum Likelihood and Bayesian approaches, is based on Monte Carlo Simulation (Graham and Talay, 2013; Kruschke, 2011). This generates a sample of random numbers using normal distribution and a specific variance (in this case, the used variance was 3), and then adding these random numbers to the original coordinates of the initial building boundary to obtain new coordinates. The output involves producing a large number (n) of simulated boundaries, which represent the building footprints. The following sub-sections (a and b) illustrate these two methods applied to building boundary regularization.

a Maximum Likelihood Estimation

The likelihood function has been tested to optimize the initial building boundary using a Monte Carlo simulation procedure. The simulation was started by simulating each initial building boundary individually and measuring the distance from the solo boundary (produced from Canny edge map, see Figure 7.28 (c)) at each pixel, to the simulated initial building boundary.

The likelihood model used is represented in equation 7-13. It is based on modelling the estimated quality of the original approximate initial building boundary, following a Normal Distribution or Gaussian approach, assuming the errors (i.e. the distance between the original and simulated boundary) are normally distributed (i.e. zero mean and σ standard deviation of error) (Wang et al., 2006).

$$L(\theta) = P(D|\theta) = \prod_{i=1}^m P(d_i|X) = \prod_{i=1}^m \left\{ \frac{1}{\sqrt{2\pi}\sigma} \text{EXP} \left(-\frac{d_i^2}{2\sigma^2} \right) \right\} \quad 7-13$$

where:

$L(\theta)$ or $P(D|\theta)$, are the likelihood or the probability distribution of the data required in order to calculate the unknown parameters θ , the function that forms the observed data;

θ is the best value of the boundary based on maximizing the probability;

d is the distance between the boundary pixels on the solo map and the simulated building boundary at each i pixel location; and,

σ^2 is the variance of the of normal distribution.

The applied model is based on measuring the distance from each pixel in a constructed individual base boundary or solo boundary to the simulated boundary. The simulation is based on the Monte Carlo method. The distance from initial building boundary's pixel to the solo boundary is obtained by using the geometrical equation (negative values are given to the distances that lie outside the solo boundary). From the diagram Figure 7.31, the distance d can be inferred which represents the shortest distance from the pixel point (x_0, y_0) , in the solo boundary, to an intersection with the edge of the simulated boundary.

This distance (d) is obtained by applying the equation 7-14 (Weisstein, 2014). The equation is finding the projection from point P_0 to the line P_1P_2 . The distance is calculated for all segments of the simulated boundary and the minimum distance will be considered the required one.

$$d = \frac{|(x_2 - x_1)(y_1 - y_0) - (x_1 - x_0)(y_2 - y_1)|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \quad 7-14$$

The above procedure is repeated until all the pixels are visited in the individual base boundary as shown in Figure 7.29(a), and having found the value of $P(D|\theta)$ using the equation 7-13. The above procedure is repeated on each simulated building boundary, which in this research is 250k times, and calculating and recording the value of $P(D|\theta)$ at each time. Since the aim is to try to find the maximum probability according to Maximum Likelihood principles, the result of the Maximum Likelihood will be the simulated boundary that gives maximum value for $P(D|\theta)$. The Maximum Likelihood approach is trying to find the best fit for the initial building boundary given measured points from the Canny edge map Figure 7.29(c), which consequently represents the best goodness of fit to the data (Deserno, 2011).

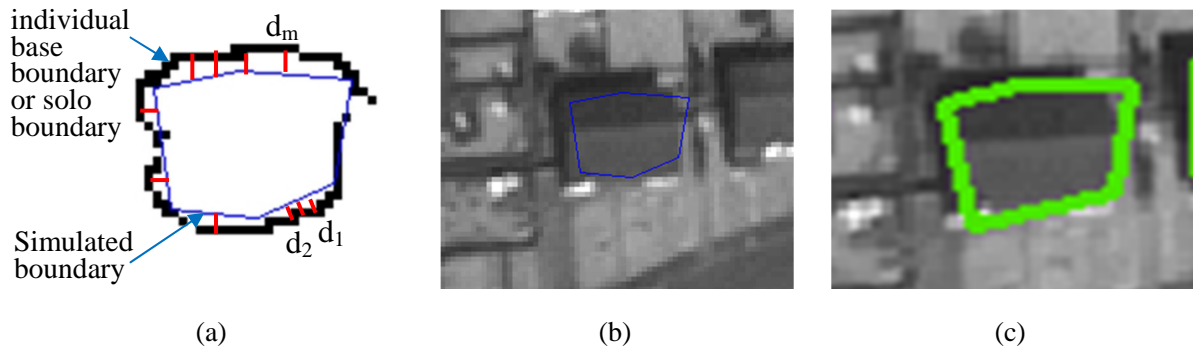


Figure 7.29 The result of optimizing the boundary using Maximum Likelihood method (a) simulated boundary and constructed individual base boundary based on Canny edge map (b) the simulated boundary overlaid on the orthoimagery (c) the result of Maximum Likelihood method.

By repeating the above-mentioned procedure on each building, the Maximum Likelihood for each of them was found in order to construct an overall result for the study area as shown in Figure 7.30.

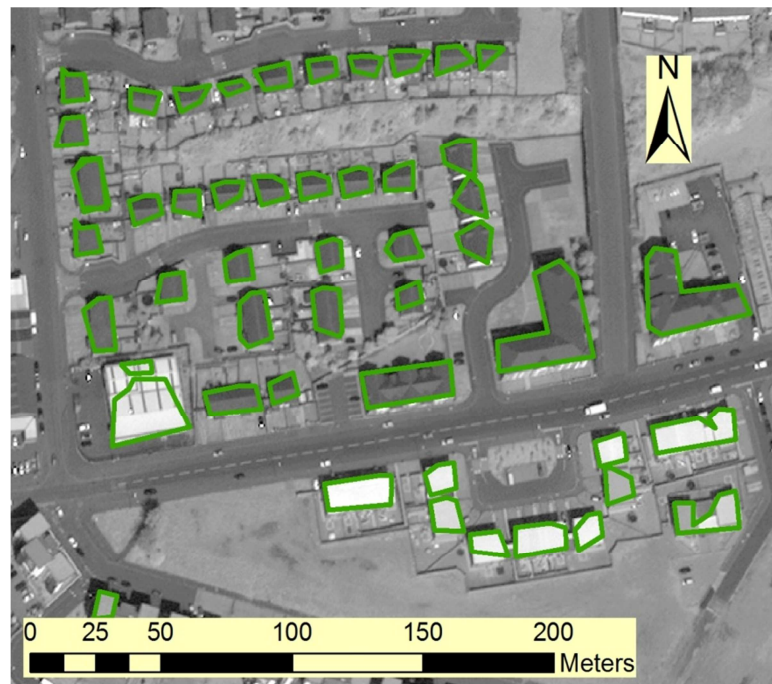


Figure 7.30 The result of applying Maximum Likelihood method to regularize building footprints over the case study area.

The role of Maximum Likelihood estimation is to try to maximize probability along the simulated building boundary using a normal distribution. At each pixel point on the individual building boundary the distance between the simulated and solo boundary is measured, later a normal distribution is fitted as shown in Figure 7.31. Using the measured distance (red colour), the probability is calculated at each point (yellow colour). Finally, the total probability is calculated using equation 7-13.

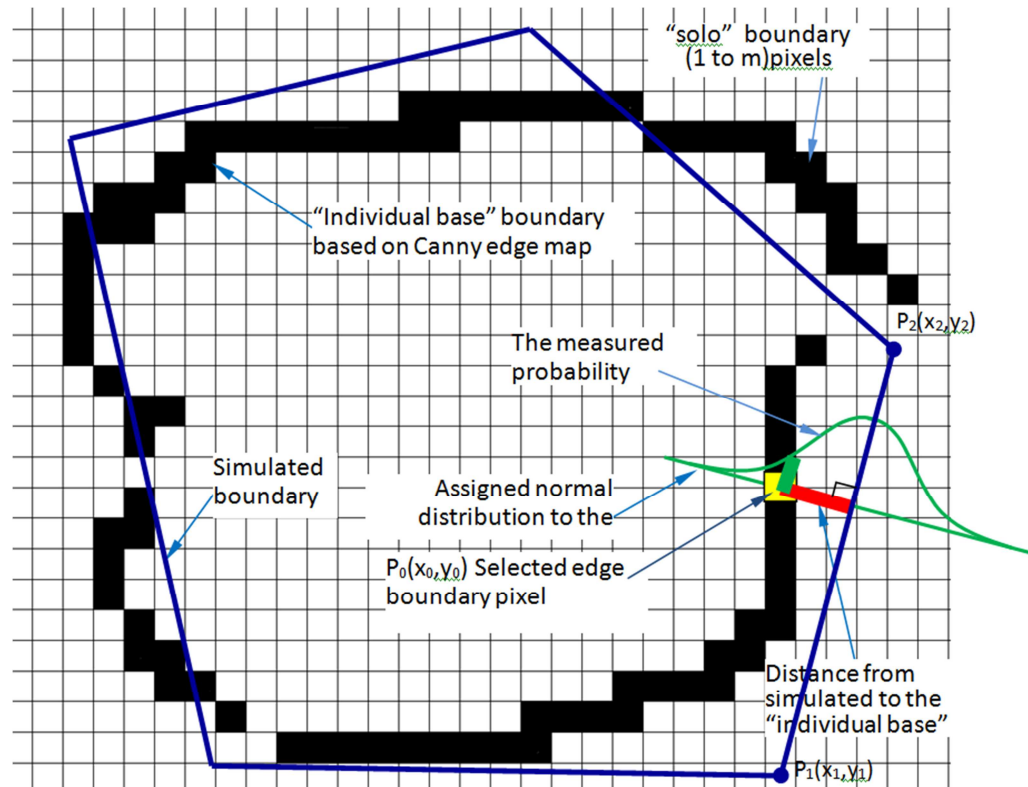


Figure 7.31 Diagram showing the distance from selected pixel to the simulated boundary.

b Bayesian Approach

Due to losing some of the spatial resolution from the initial building boundary, in the earlier stages, the inability of the Canny edge map to detect the building edges clearly and the modifications applied to the edge map to produce an individual base boundary, the result when using the Maximum Likelihood approach was not acceptable.

The next stage of building regularization is an examination using Bayesian statistics. The Bayesian approach is based on using prior probability, because it represents the expected situation and if the prior probability is a poor representation, the outcome may be unsatisfactory. Therefore the prior probability should reflect the expected situation as purely as possible.

In general, the Bayesian approach as discussed earlier, in chapter 4, consists of a joint probability distribution (which is often called the posterior probability) divided by probability term $p(X)$ for normalization purposes. This is shown in equation 7-15 (Kruschke, 2011), and is based on the normalised joint distribution of the likelihood and the prior probability (often referred to as *a priori* probability), which are both initially represented by their relevant probability density functions (pdf) of $p(\Theta)$ and $p(X|\Theta)$.

$$p(\theta|X) = \frac{p(\theta, X)}{p(X)} = \frac{p(\theta)p(X|\theta)}{p(X)} \quad 7-15$$

Where:

$p(\theta|X)$ the posterior probability, which is the probability of the hypothesis given the observed data.

$p(\theta, X)$ the joint distribution of the hypothesis and the observed data

$p(\theta)$ the prior probability, the probability of hypothesis is true before measuring any data.

$p(X|\theta)$ the likelihood, which is the probability of the observations arising given the hypothesis.

$p(X)$ the evidence, marginal likelihood, or normalization factor, which is used to normalize the Bayesian model.

Since the aim is to find a solution providing a relative maximum value, not finding the overall probability, and because the parameter θ does not interfere in the calculation of the normalization factor, therefore, it is not necessary to calculate the normalization factor $p(D)$, and thus the equation becomes (Link and Barker, 2010):

$$p(\theta|X) \propto p(\theta)p(X|\theta) \quad 7-16$$

The equation 7-16 can be translated into another form by substituting the θ as the building footprint and X as the location of the initial building boundary points.

$$p(\text{building footprint} | \text{location of initial building boundary points}) \propto p(\theta)p(X|\theta)$$

The aim is to find the building footprint boundary in terms of its x,y coordinates given the angle between the edges and the distance from the simulated boundary and the Canny edge map,

$$p(x_1, y_1, x_2, y_2, x_3, y_3, \dots | \alpha_1, \alpha_2, \alpha_3, \dots, d_1, d_2, d_3, \dots) \propto p(\theta)p(X|\theta) \quad 7-17$$

The result is obtained by finding the maximum of the argument (the product of prior probability and likelihood), or **argmax**, for several possibilities, using equation 7-18 (Tournaire et al., 2010).

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \{ p(\theta)p(X|\theta) \}$$

7-18

The author suggests that the Bayesian approach has been believed to be a more robust method for building footprint construction than Maximum Likelihood method since it is based on synthesising the model using prior information about the buildings and measurements.

The assumption (or expectation) used here for the prior probability is based on considering building footprints to have straight edges and right-angled corners, i.e. 180 degrees between the edge points (defining walls) and 90 degrees at corners, with low penalties applied to other angles such as 45 degrees. This assumption is based on work by Wang et al. (2006) who implemented a Bayesian approach with LiDAR data to extract building boundaries. By reflecting this assumption into a probability distribution form, it can be used in a Bayesian approach as shown in the following graph, Figure 7.32. From the graph, it can be seen that the value of the penalty near $90^\circ \pm 3^\circ$ and 177° to 180° is equal to 10^{-45} . Using such small number has a considerable effect during the optimization stage. The experimental results show that using other values rather than the abovementioned sometimes failed to achieve right-angled corners or straight lines for all building corners or edges, respectively.

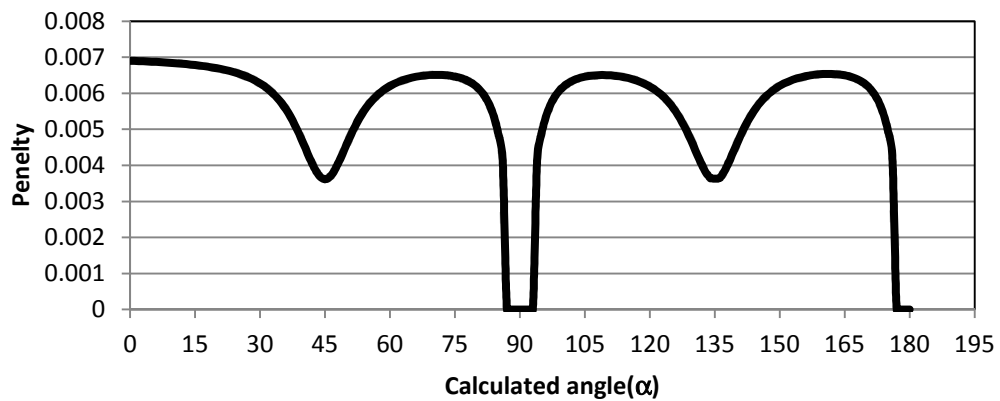


Figure 7.32 The prior probability function $f(\alpha)$ that used to find the penalty, 90° and 180° degrees gives the minimum penalty while the penalty for 45° and 135° is higher, and even higher for other angles.

The formula that has been used in the above graph is shown as equation 7-19 and has been obtained experimentally based on the assumption that made by (Wang et al., 2006), which is used to find the penalty for the angles obtained during the simulation stage.

$$f(\alpha) = \left\{ \begin{array}{ll} \frac{0.21 - \frac{6}{(\alpha-45)^2 + 60} - \frac{2}{(\alpha-90)^2 + 9.7} - \frac{6}{(\alpha-135)^2 + 60} - \frac{2}{(\alpha-180)^2 + 9.65}}{29.92} & \dots \text{if } 0 \leq \alpha < 87 \text{ or } 93 < \alpha < 177^\circ \\ 10^{-45} & \text{otherwise} \end{array} \right\} \quad 7-19$$

The multiplication of the prior probability and the likelihood will lead to a posterior probability. It is possible to find the maximum posterior probability, or take either the maximum of the log of the argument derived from equation 7-23 or the minimum of the negative log function (Diebel et al., 2006) It is preferable to take the minimum value since the values of the product of probabilities are very small. To derive the posterior probability the prior probability (7-20) is multiplied by the likelihood (7-21):

$$P(\theta) = \prod_{i=1}^k f(\alpha_i) \dots \dots \dots \text{prior} \quad 7-20$$

$$P(X|\theta) = \prod_{i=1}^m \left\{ \frac{1}{\sqrt{2\pi}\sigma} e \left(-\frac{d^2}{2\sigma^2} \right) \right\} \quad 7-21$$

To optimize the function (7-18) the above formulae are utilized to give:

$$\bar{\theta} = \operatorname{argmin}_{\theta} \{ -\log\{P(\theta) \cdot P(X|\theta)\} \} \quad 7-22$$

$$\bar{\theta} = \operatorname{argmin}_{\theta} \{ -\log [\prod_{i=1}^k f(\alpha_i)] - \log [\prod_{j=1}^m \left\{ \frac{1}{\sqrt{2\pi}\sigma} e \left(-\frac{d^2}{2\sigma^2} \right) \right\}] \} \quad 7-23$$

$$\bar{\theta} = \operatorname{argmin}_{\theta} \left\{ -\sum_{i=1}^k \log[f(\alpha_i)] - n \log \left[\frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{j=1}^m \log \left[e \left(-\frac{d^2}{2\sigma^2} \right) \right] \right] \right\} \quad 7-24$$

By simplifying the equation 7-24, opening the brackets and omitting the constants that do not affect the result, the final equation can be represented by equation 7-25.

$$\bar{\theta} = \operatorname{argmin}_{\theta} \left\{ -\sum_{i=1}^k \log f(\alpha_i) + \frac{1}{2\sigma^2} \sum_{j=1}^m d_j^2 \right\} \quad 7-25$$

The Monte Carlo simulation method has been used in the optimization process, as subsequently described. First, evaluate the distance of the simulated boundary to the actual building boundary or solo boundary derived from the Canny edge map. Then measuring

the corner angles, see Figure 7.33, for each simulated boundary calculate the penalty from the graph. Utilize the values obtained in Equation. 7-25 to evaluate the result, which is based on a minimum value of the result produced from the equation with all simulated values. The value of the σ^2 has been selected initially as 0.5 and increased gradually. It was found that the best value is considered to be 3, and it can be fixed at this value. However the larger value leads to neglecting the Maximum Likelihood cost consequently giving all the weight to the prior $\sum_{i=1}^k \log f(\alpha_i)$, and for that reason it was necessary to select a value which leads to achieve a balance between prior and Maximum Likelihood.

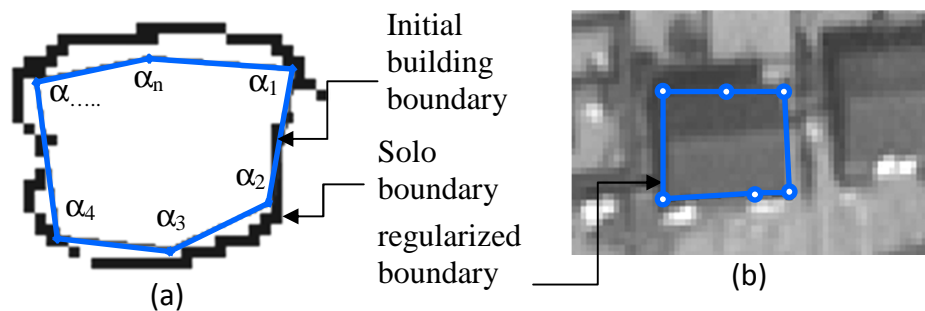


Figure 7.33 (a) Initial building boundary overlaid over the edge map, the corner angles and the distances from the solo boundary to the initial building boundary can be seen; (b) result of regularized initial building boundary which is obtained by seeking the situation generating the minimum value for Eq. 2.

The result of applying a Bayesian approach can be seen in Figure 7.34, which follows executing the model on all buildings, in order to find the optimized result for each building.

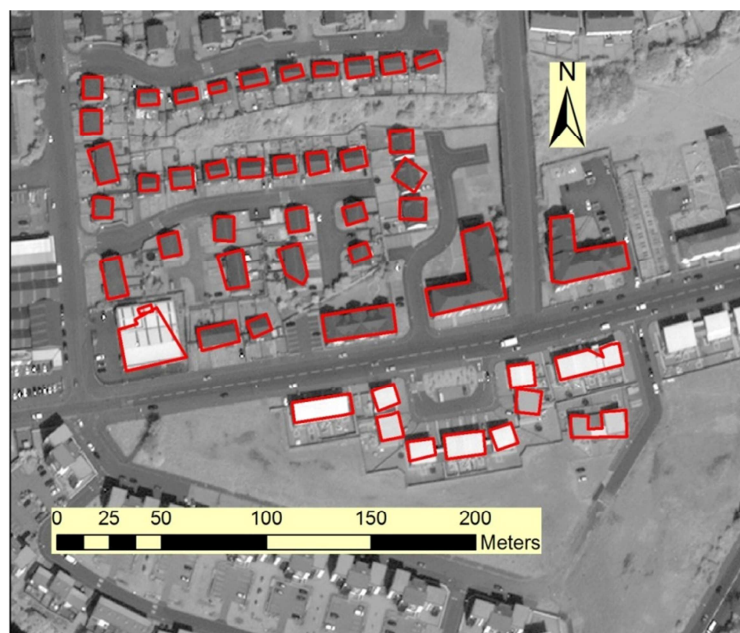


Figure 7.34 The result of applying Bayesian Approach to regularize building footprints over the case study area.

The Figure 7.35 shows the difference between the initial and optimized boundary using Bayesian theory. The initial building boundary first has been overlaid on the edge map, see Figure 7.35(a). Clearly, it was not possible to utilize this in the optimization process, because of so many edges in the area. During measuring the distance from the initial building boundary to the simulated footprint, other edges were constantly being intercepted consequently giving a wrong result. For that reason, the individual base boundary has been introduced. The boundary, which is used for the optimization, is shown in the Figure 7.35 (b). It is clear that only one boundary has been kept and the simulation will be against that boundary only. The images in the lower row of Figure 7.35 show the result of the optimization overlaid on the edge map, individual base boundary and orthomap. It is noticeable that the boundary that has been extracted has almost right-angled corners and the points on the edge have been straightened.

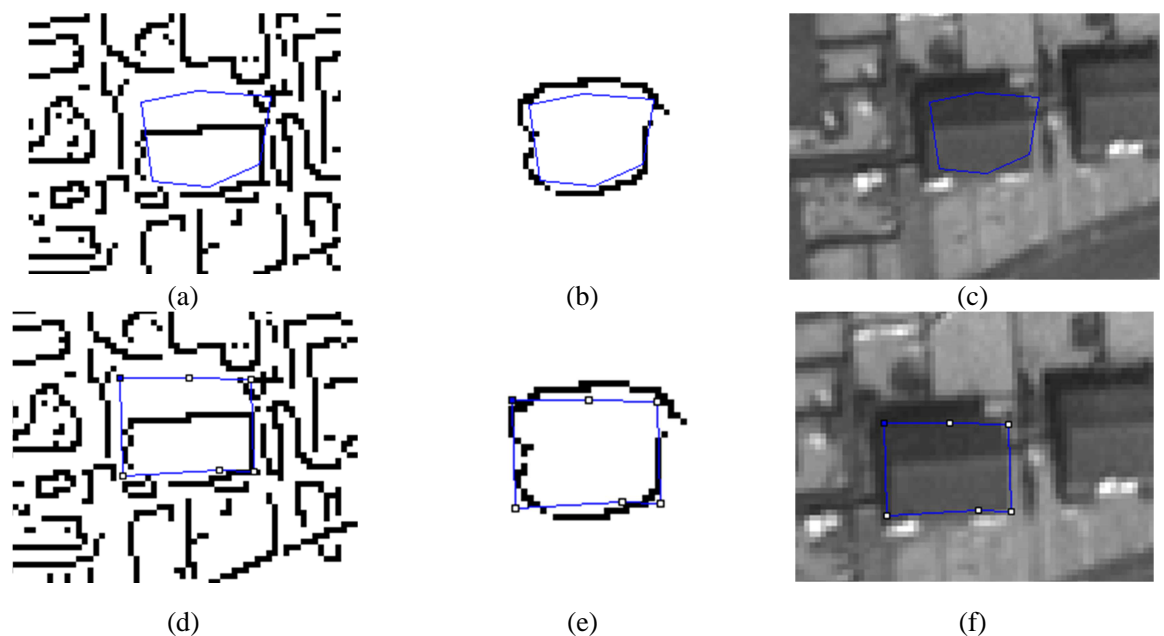


Figure 7.35 Comparison of initial and optimized building boundary: (a)&(d) initial and optimized building boundary sample overlaid over the edge map; (b)&(e) initial and optimized building boundary overlaid over individual base boundary; (c)&(f) initial and optimized building boundary overlaid over orthoimagery.

Figure 7.36 shows the initial building boundary (blue) and optimized building boundary (red), both are overlaid on the orthoimagery to see the effect of applying a Bayesian approach.

The processing time expended to achieve building footprint extraction and regularization using a 2xCPU 2.3 GHz processor, with either a Maximum Likelihood or a Bayesian

approach for the 49 buildings with 250k iterations, extended to 6 hours and 4 minutes. The choice of 250k iterations was selected experimentally: initially it is started at 100k iterations but the result of the optimized building boundary was not regular, therefore the number of iterations was been increased gradually to 250k iteration. At 250k iterations building boundaries were regular, that is the corners were right-angled and the edges were straight.

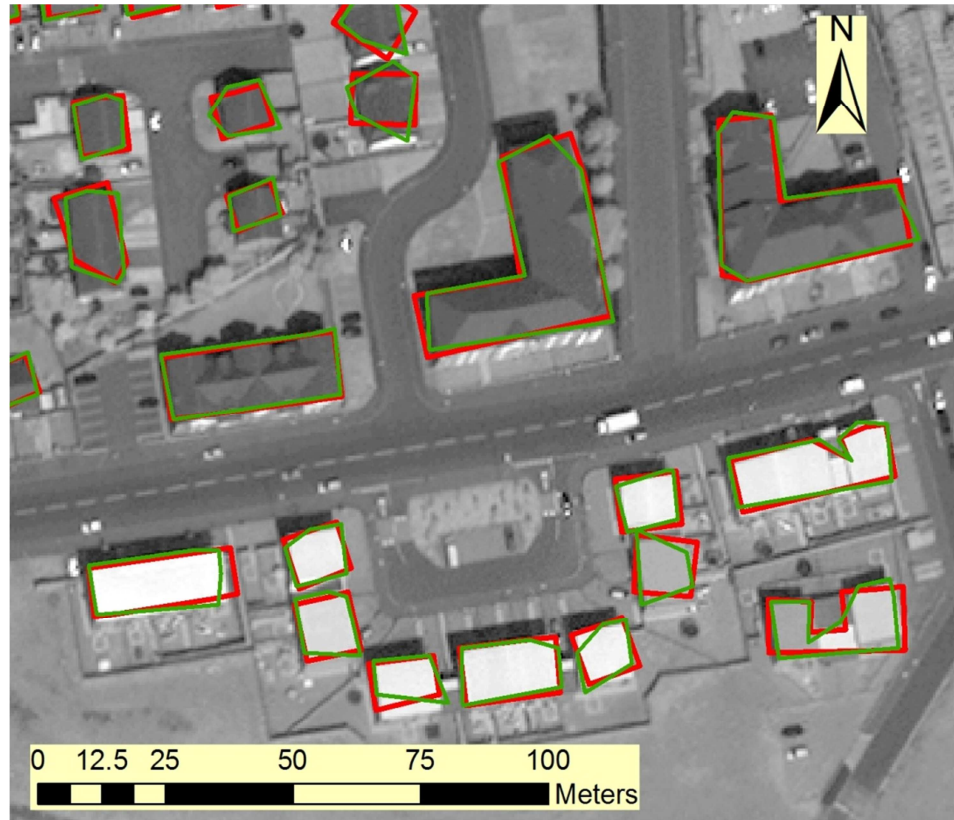


Figure 7.36 Vector map showing the result of applying Bayesian theorem (red) to regularize building boundary lines. Green lines represent the initial building boundary.

7.7 3D Model Building Construction

Once the building footprint has been produced for each building, the next stage is producing a 3D model utilizing the extracted height from the DSM. The constructed model is based on LoD1 which is based on taking the maximum height of the building according to the specifications of CityGML (see Figure 1.1) as mentioned by Gröger and Plümer (2012).

For the 3D model construction, the regularized building footprint has been extracted by taking the height in the corresponding part from the DSM. The above procedure has been applied on each building and saved as vector file so it can be used in the visualization, see Figure 7.37.

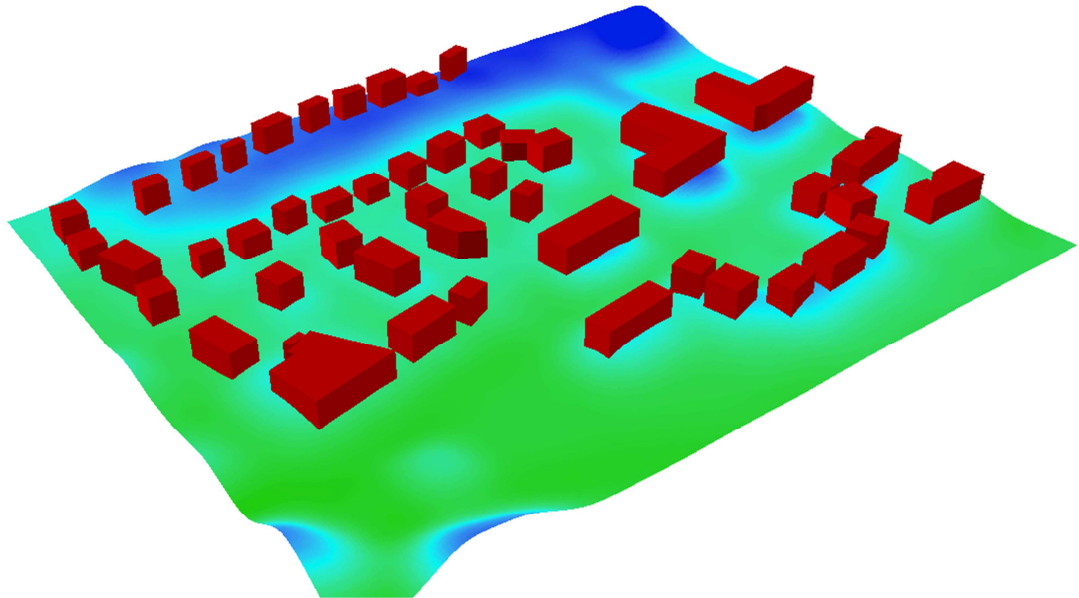


Figure 7.37 3D models for the study area based on extruding the building footprint based on maximum DSM height in the specified area.

For the purpose of visualization and to place the buildings on the ground, as shown in the above figures, the smoothed DSM that is utilized for the nDSM production in section 7.2.2 has been used. The resulting DTM was obtained by smoothing the DSM by implementing a 49x49 kernel. It is possible to use more exacting technique to produce a higher quality DTM such as mentioned by Krauss et al. (2007), but it was sufficient to use a rudimentary method.

7.8 Summary of the Developed Process and Associated Algorithms

The building extraction process and associated algorithms, which are shown in the flow chart Figure 7.1, are summarized below, in twenty-three steps, which include data processing, enhancement and building detection, extraction and construction.

1. Produce orthoimagery and DSM using Socet GXP from Satellite imagery.
2. Produce NDVI from multispectral imagery, using Pleiades satellite imagery, using ArcGIS software.
3. Extract the road from orthoimagery.
4. Enhance the orthoimagery by sharpening and applying a mean shift algorithms, using ImageJ software.
5. Produce segmented images and edge map from orthoimagery, and then produce nDSM from DSM, using ImageJ software.

Note: All subsequent operations have been implemented using C++ scripts written by the author

6. Label the nDSM using the connected component algorithm.

7. Detect the shadow by thresholding using pixel values less than 26. Then apply closing MM to produce mass shadow segments.
8. Subtract the road and NDVI from nDSM keeping the buildings only.
9. Subtract the roads, shadow and NDVI from the segmented images. Then apply minimum error threshold to obtain image-A and Moment threshold method to obtain image-B.
10. Erode the segmented image-A using a circular structural element then opening the image - using erosion twice and dilation twice. Then perform a further dilation using a circular structural element in order to remove noise and separate the objects from each other.
11. Apply MM opening on the segmented image-B using a square structural element, first erode the image twice then dilate it twice.
12. Enhance the edge map by removing the noise,
 - a. First dilate the segmented images A and B, just once.
 - b. Take the boundary and add it to the edge map, this process fill the gaps in the location where the Canny edge detection could not detect the edges, due to low contrast.
 - c. Erode the segmented images A and B twice then subtract them from the edge map, this process removes the noise from inside the building.
 - d. Pass a cleaning filter to remove the scattered object from the edge map
13. Take each region in the labelled nDSM individually. Since each building is detected as a 'blob', and just part of it, it is necessary to dilate each building 17 times in order to cover a whole building, and little more.
14. Take the corresponding part of the labelled image from the segmented images A and B, compare each segment and take the object that has the maximum area.
15. The area in the previous section consists of many parts, therefore these need to be labelled, using connected component labelling taking each part individually.
16. Compare each detected object in the segmented image with the objects in the nDSM and keep the common ones only.
17. Use the above segments to find each initial building's boundary using the MM method.
18. Sort the initial building boundary using the NNA algorithm and then approximate using the Ramer-Douglas-Peucker algorithm.
19. Create the individual base boundary, by taking the segmented image that is produced in section 14, and then erode it 5 times. From each pixel on the boundary

create a ray of lines, through 360 degrees and record wherever it intersects the modified Canny map, keep only this intersection and use it in the optimization process. Apply the Bayesian Model to find the best fit boundary with the most right-angled corners and straight edges.

20. Create a prior probability that represents our belief in the building boundary.
21. Use the initial building boundary, produced in item 17 and use the Monte Carlo method, simulating a random boundary and selecting MAP.
 - a. First, measure the internal angles of the simulated boundary and use it in the prior function, equation 7-19, to find prior penalty.
 - b. Measure the distance from each pixel In the individual base boundary - (see item 18) to the simulated boundary that produced
 - c. Plug-in the values from a and b in the Bayesian model and find MAP, the boundary that corresponding to the maximum value of the Bayesian will represent the candidate building boundary.
22. Take the boundary that created in item 19, and check the maximum height using DSM created in item 1.
23. Repeat items 19 and 20 on each building individually, covering all objects in the labelled nDSM.

7.9 Preliminary Assessment

The method was able to detect full 3D models by depending on the raw data only, without using any external data such as 2D Cadastre database, such as had been used by Tack et al. (2012). By applying the above-mentioned algorithm on the WV-1, it is noticed that the buildings are clearly extracted although some of them still have problems. 100% of the buildings have been detected. The algorithm applied gave as good as or a better result than other projects in the same field, Aytekin et al. (2009) showed in their paper that the extraction result was only about 81%. Meanwhile Dahiya et al. (2013), another study in the same field, exhibited the result that 24 building has not been extracted among 122 building, and moreover, 18 buildings were wrongly identified.

The algorithm has been checked against another study area and is marked as area-2, see Figure 7.38, with dimensions 203.5m by 271m that has different building orientations with low variability and the buildings being close to each other, see Figure 7.40. For that reason after the meanshift algorithm and before applying the minimum error threshold algorithm for the segmentation, the image has been smoothed by substituting each pixel's value by the median of the group of 5 x 5 pixels surrounding it, see Figure 7.39.

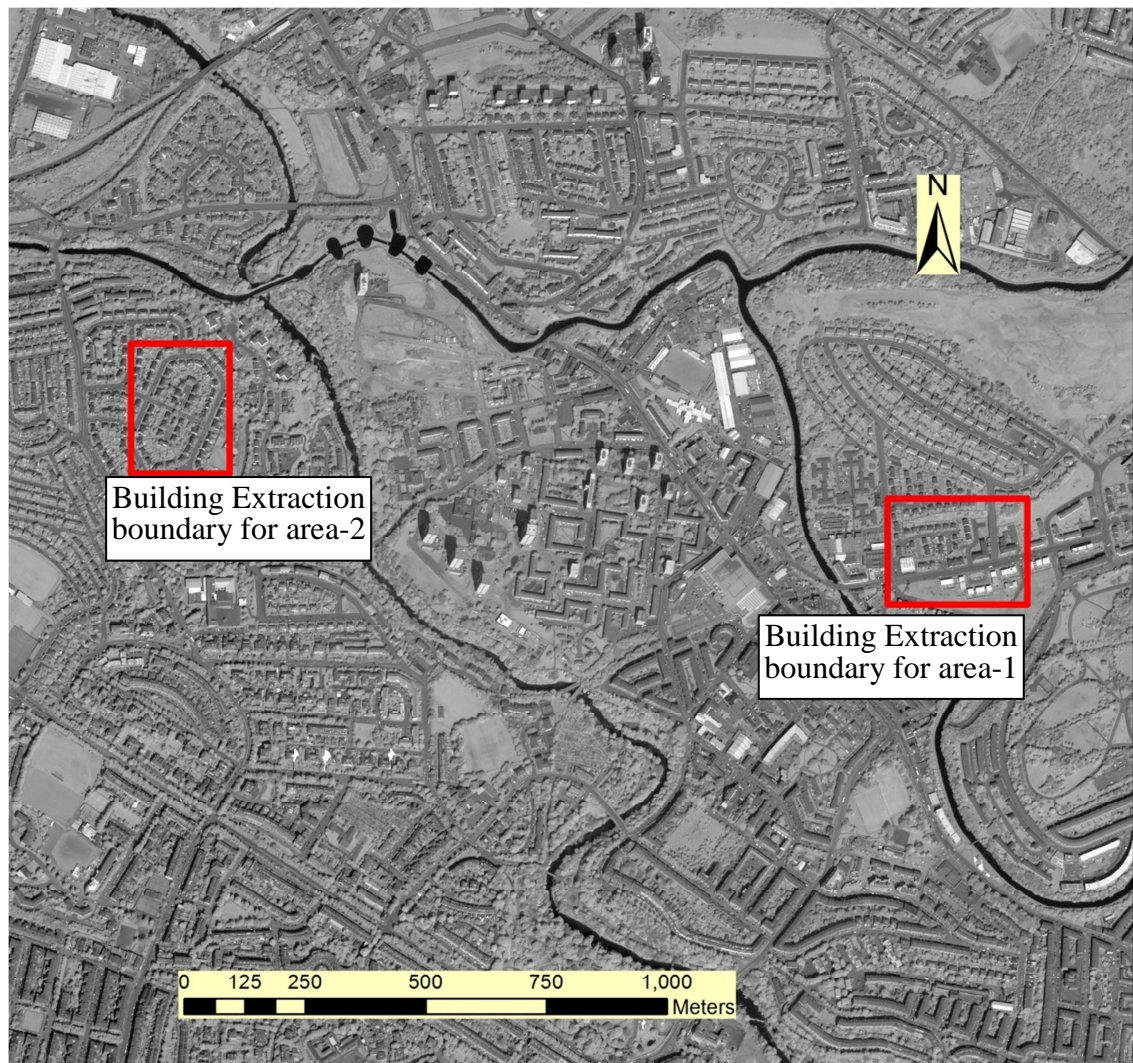


Figure 7.38 Study area-1 and study area-2 inset in the WorldView-1 satellite imagery.

It can be noticed that all the building have been detected successfully, except some of the buildings mostly at the border. It is worth mentioning that the buildings and surrounding texture were quite similar and therefore it was difficult to distinguish the buildings, precisely. Moreover, due to the building orientation being inclined the structural element was based on a cross structural element Figure 7.2(b) instead of square structural element, see Figure 7.2(d). Used were one Circular Erosion, followed directly by one Circular Dilation, followed by two Cross Erosions and finally then two Cross Dilations. In addition to that, the threshold value that has been used in smoothing (i.e. Ramer-Douglas-Poiker) the initial building boundary has been reduced to 3 pixels (1.5m) instead of 5 pixels (2.5m).

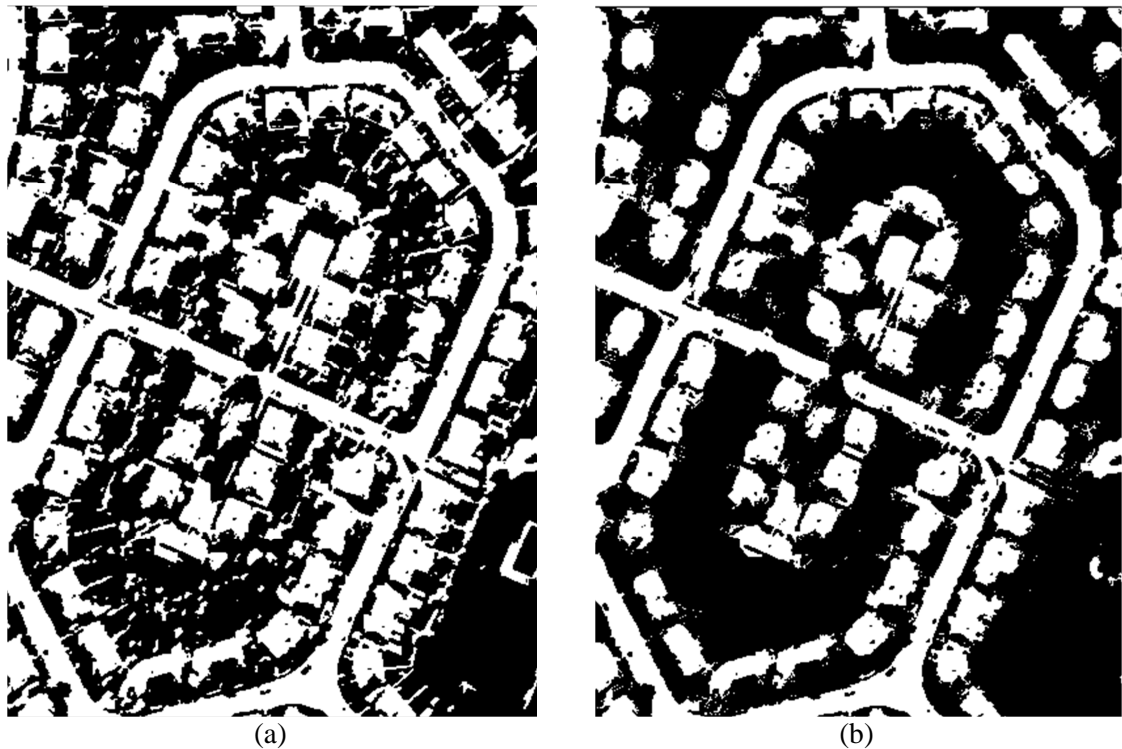


Figure 7.39 Segmented image showing the effect of smoothing by replacing the median with its surrounding boundary (a) before enhancement (b) after enhancement. (Also see Figure 7.40)

In the second test area of the fifty-eight buildings, ten were partially detected and one not detected at all. It is also worth reminding the reader that in the first test area among the forty-nine buildings all were detected with only four of them having been detected partially. In the first test area it is clear that 46 of the buildings have been detected very clearly, which gives a successful extraction percentage of 88%, although a few buildings are rotated slightly and have poor edges. This initially subjective evaluation will be extended, using more objective means, in chapter 8.

Some of the boundary has been detected more than once, namely the building at the bottom-left, as shown in Figure 7.30 and Figure 7.34. This is because the mentioned building has been extracted twice due to some objects remaining in the nDSM, see Figure 7.9. This consequently causes that object to appear as a separate building. These extra boundaries can be removed by applying some rules to the algorithm based on keeping the area that has biggest common area with initial building boundary and filter out the rest.

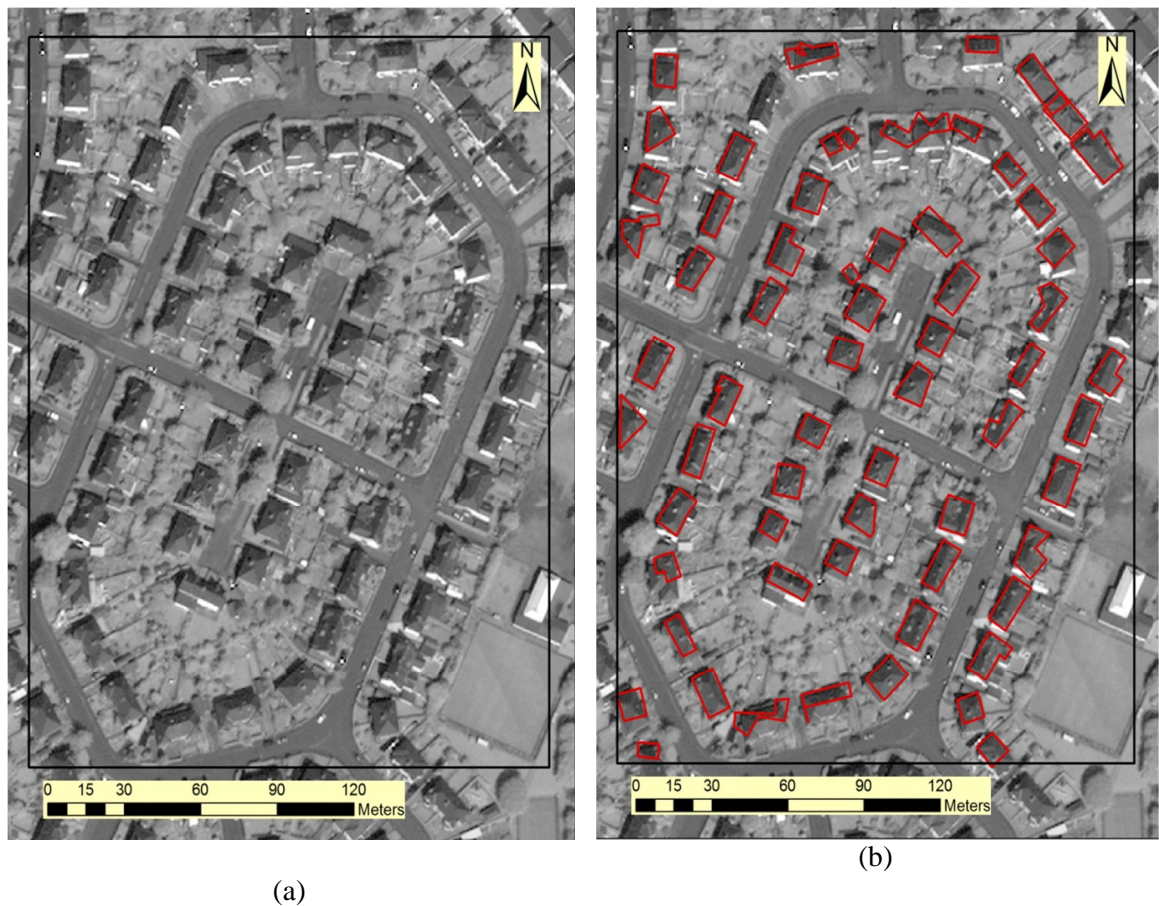


Figure 7.40 Alternative study area used in applying the building footprint construction algorithm (a) orthoimagery used in the test (b) extracted building footprint overlaid on the orthoimage.

7.10 Conclusion

The results obtained shows that satellite imagery can stand as a cost effective source for producing building footprints and 3D modelling. Both Maximum Likelihood and Bayesian approaches were able to extract building boundaries more effectively, when compared with the majority of approaches mentioned in section 7.9; for example the work reported here achieved 100% and 88% building detection in the two study areas, whereas earlier efforts did not achieve this, with Aytakin et al. (2009) achieving an extraction result of 81% and Dahiya et al. (2013), 80%. Furthermore, the Bayesian approach was able to improve building footprints noticeably. The suggested method is an almost fully automated and unsupervised method, except for the road detection part that is done manually since there has been considerable research in this field (and thus might have been done automatically). The applied algorithm can be implemented with few parameter changes, which consequently can be considered a good feature in terms of building footprint extraction.

Chapter 8 Result and Evaluation of Building Footprint and 3D Modelling

The process evaluation is presented in this chapter; it includes the evaluation of the constructed building footprint and 3D models. For the evaluation purpose the ‘true’ data that have been used were acquired from differential GNSS and total station observations of real ground elevations. In the assessment, both subjective and objective analyses have been used. Visual inspection has been used for the subjective evaluation. Regarding the objective validation, statistical analyses have been used such as RMSE and the percentage of false negative and false positive areas extracted.

8.1 Study Area and Implemented Data

For testing the building boundary regularization, two areas were selected for that purpose. Since the time used in regularizing the building boundaries was very large, due to simulating each building boundary through 250k iterations, small study areas were used in order to minimize the time of processing. The nature of the investigation reported in this thesis meant that the simulations were repeated many times as the method evolved. To make the investigation manageable relatively small study areas were used. The study areas that have been used in building extraction were within the 10km² that have been used in the merging operation, as shown in Figure 5.2. The study areas were representing the whole sample space which contained all types of buildings such as complex buildings, one direction sloped roof buildings and hipped roof buildings.

Two urban areas located in Glasgow were chosen, selected to facilitate evaluation purposes. The first area used to evaluate building footprints, called area-1, has the extents 419870 mE, 6194210 mN lower left and 420160 mE, 6194430 mN upper right, which covers an area 290m by 220m shown in Figure 8.1(a), and includes 49 different buildings with different heights, structures and roof slopes. The other area is called area-2, it has been having boundary coordinates as (418302.54 mE, 6194480.78 mN) lower left (418506.04 mE, 6194751.78 mN) upper right, the study area covers an area 203.5m by 271m as shown in Figure 8.1(b). The given coordinates are referenced to UTM zone 30 (North), using the WGS84 ellipsoid.

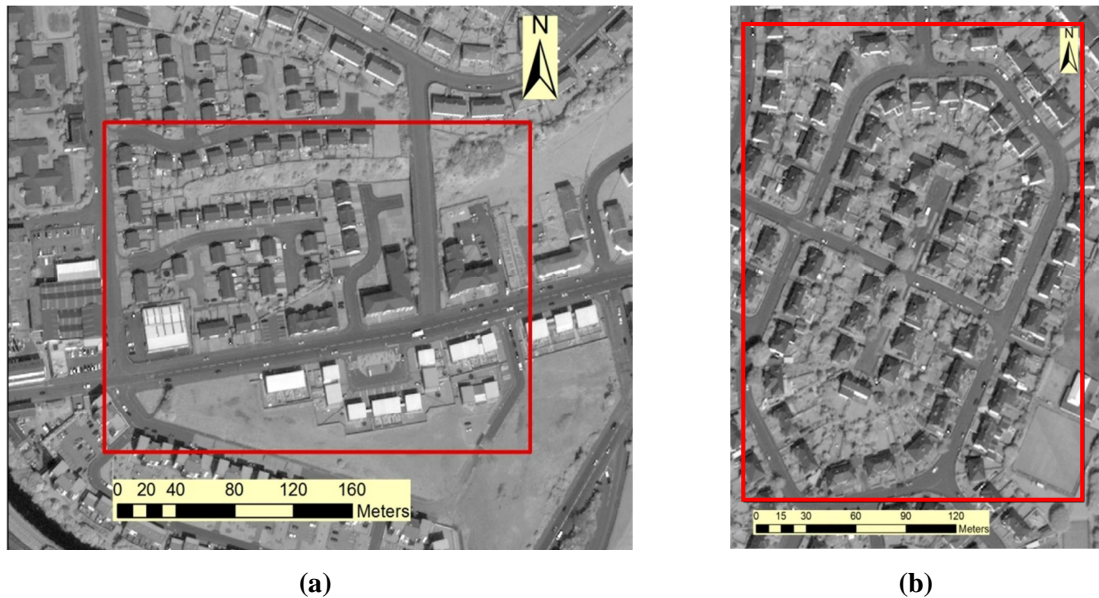


Figure 8.1 The study area that has been used to test building extraction (a) study area-1, (b) study area-2

8.2 Reference Data

The checkpoints that have been used to validate the extracted building footprints and 3Dmodels were acquired in the field using total station equipment. The observations were taken of the building corners and the highest elevation point on each rooftop using a reflector-less total station instrument. Prior to embarking on observing the field points, a set of GCPs using GNSS-RTK instrumentation has been established in the area.

Originally, the E and N coordinates for each point were in the OSGB coordinate system only, however the elevations were available on two vertical datums, Newlyn (Ordnance Survey (UK), 2013) and WGS84 ellipsoid. Since the coordinates of the orthoimagery and the DSMs that are implemented in the processing were on the UTM-Zone30 projection grid coordinate system and Eastings, Northings and height were all referenced to the WGS84 ellipsoid, as it was necessary for all coordinates to be on the same coordinate system, any British National Grid coordinates were transformed to UTMZone30 on WGS84. For the transformation, ArcGIS software has been implemented. Different projection systems exist in the ArcGIS environment that can be used in the transformation, WGS84-UTM30N (i.e. northern hemisphere) with the option “OSGB 1936 to ED 1950 UKOOA +ED 1950 to WGS 1984_1” selected.

Prior to assigning the projection UTM-30N to the coordinates, first, the data are imported to the ArcGIS and the OSGB coordinate system was assigned. Later it has been transformed using the available ArcGIS tool as shown in Figure 8.2, and displayed with the orthoimage Figure 8.3.

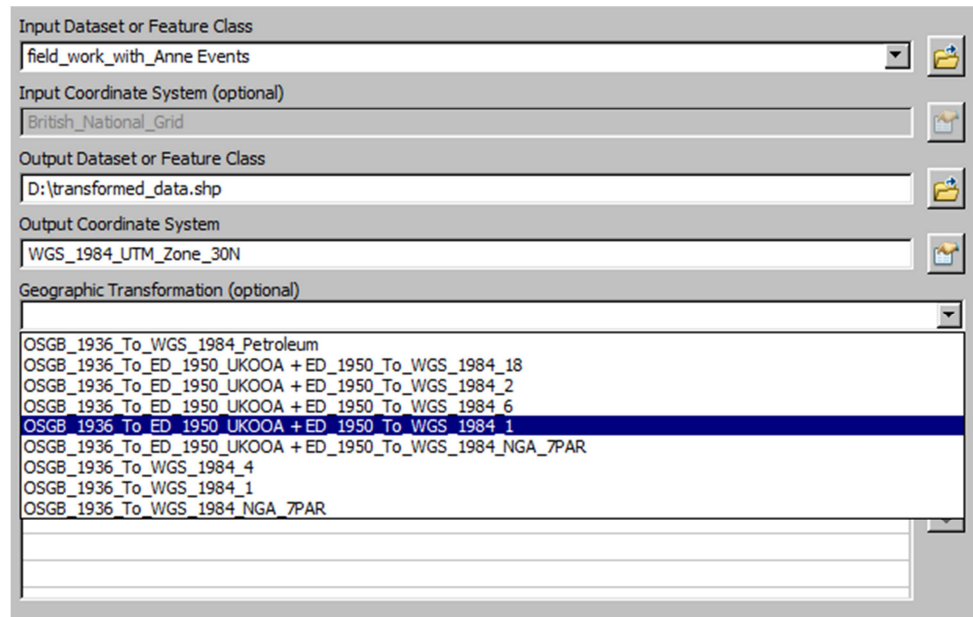


Figure 8.2 Transforming the British coordinate system (OSGB 1936) to UTM-30North.



Figure 8.3 The blue line is OS master map produced using transformation option in ArcGIS overlaid on the WV-1 orthoimagery. The green dots represent the measured filed points with total station. Crown Copyright and Database Right 2015. Ordnance Survey

8.3 Qualitative Assessment for Building Footprints

It was decided to use qualitative assessment to express the shape quality of extracted buildings. It is clear that the algorithm was able to detect the buildings and construct

geometrical building shapes close to the original shape. The visual inspection shows that the buildings were constructed as intended, based on extracting the building roofs Figure 8.4. However depending on the roofs, systematic error was found during the validation stage, which will be shown in the next section.

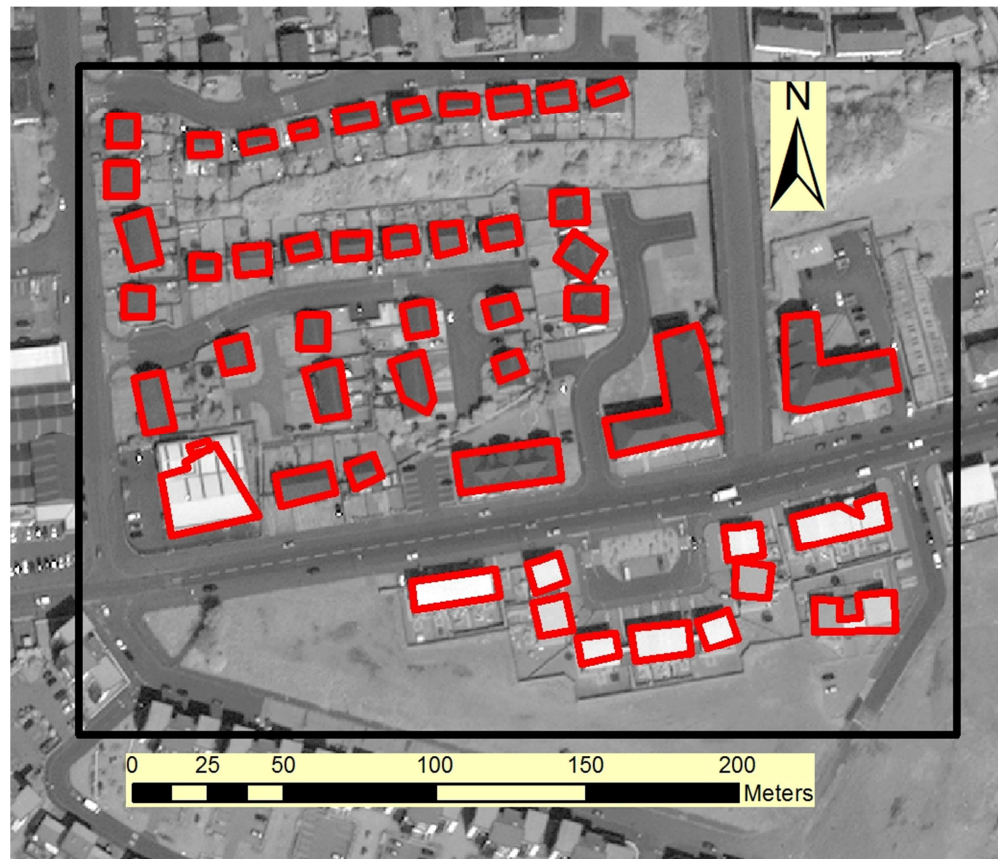


Figure 8.4 The extracted building footprint by applying Bayesian approach.

The qualitative assessment shows that the algorithm, in addition to detecting regular simple small buildings, was also able to detect and extract complex buildings. As illustrated in Figure 8.5 complex buildings have been clearly identified, as shown in the lower right. Such a building is very challenging for automatic construction because it specified to have complex fragmented shape. A building that has an L-shape also clearly was identified and extracted.



Figure 8.5 The extracted buildings in the study area.

In addition to the successful building shape extraction in some areas, it is necessary to mention that the algorithm had some shortcomings in extracting building successfully. In some areas although it was possible to extract buildings as they had been originally expressed, they have been rotated. As shown in Figure 8.6, some buildings were rotated and there is a building for which just a part has been extracted.

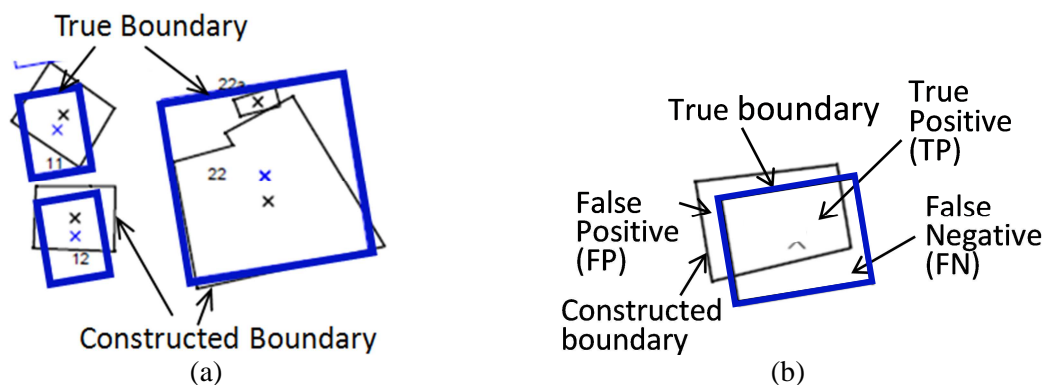


Figure 8.6 Building Extraction Assessment (a) The buildings which were rotated or incorrectly constructed (b) true and incorrectly constructed buildings with their centre points

It is clear that the delineated buildings are located at almost the correct building position with slight discrepancies, which could be neglected under some subjective circumstances, such as, when required to construct 3D model for the purpose of general planning or emergency response.

8.4.1 Quantitative Evaluation Indices

For validation purposes, the RMSE index has long been considered as the best index for measuring error, as reported by the ASPRS under the title “ASPRS Accuracy Standards for Digital Geospatial Data” (2013). The report praises the importance and effectiveness of RMSE rather than standard deviation (or σ) for representing error. The latter is used to give an indication as to how measurements cluster (Wood, 1996).

If no blunders or systematic error (systematic error is manifest as bias) exist in the data, then RMSE and the σ of unbiased error are expected to be numerically close in value, however if there is bias the RMSE will incorporate it while the σ of unbiased error will not, and for that reason the σ of unbiased error- cannot be considered to represent accuracy.

Bias is an important concept and the arithmetic mean of discrepancies gives an indication of its nature. The formula that is used in calculating the bias is showing in equation 5-1, while equation 5-2 refers to the standard deviation formula that is used to estimate the standard deviation (or σ) of unbiased error, and equation 5-3 refers to RMSE.

These three indices (subsequently referred to as: arithmetic mean discrepancy; σ of discrepancies; and RMSE), in addition to minimum and maximum error, have each been calculated for the corner coordinates, centre point coordinates, area dimensions and the height of the buildings.

8.4.2 Corner Validation

A quantitative test has been applied at the building corners in order to assess the accuracy of the constructed buildings. The accuracy has been estimated by measuring the discrepancy between the corner of the constructed building's footprint and the measured coordinate. The measured coordinates have been obtained from a total station referenced to the WGS-84 coordinate system using GNSS, and thus these coordinates can be considered as the 'truth'. The total station used was Leica TCR805 with 1-second angular resolution. The accuracy that has been evaluated in this section is limited to the positional planimetric accuracy derived from the discrepancies in the corner points, Figure 8.8. The discrepancy of each corner at each building has been measured individually. The RMSE and other statistical indices used to indicate the planimetric quality of each building is shown in Figure 8.6, and similar indices have been determined for the 35 buildings shown in (e.g.) Figure 8.7.

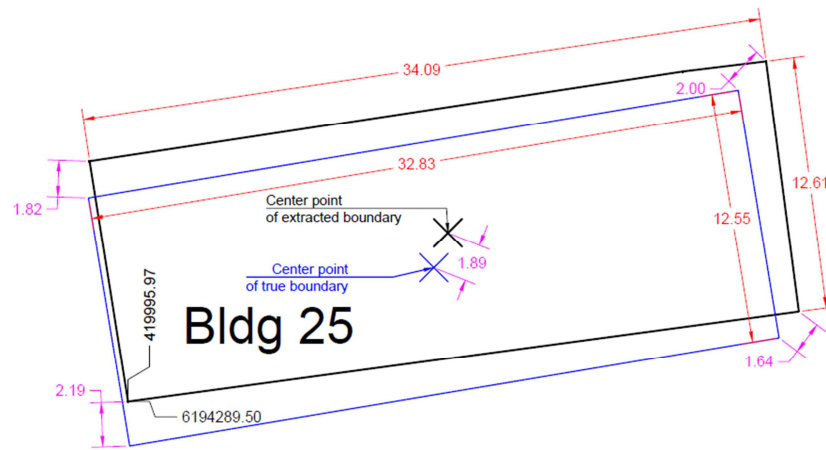


Figure 8.8 The constructed and true building corners, with planimetric corner discrepancy measurements.

The result of the discrepancies in the corners is summarized in Table 8.1 (see APPENDIX E). It shows that the value of the RMSE in addition to the other statistical measurements.

Table 8-1 Buildings Corners' planimetric quantitative statistical result from measuring 144 building corner points.	
Min. discrepancy(m)	0.12
Max. discrepancy(m)	9.65
Arithmetic Mean discrepancy (m)	2.23
σ of discrepancies (m)	1.62
RMSE (using discrepancies)(m)	2.88

Focusing on the roof of the building that exhibits lean, the building extraction has consequently led to a well-known systematic error. The issue of systematic building lean is well researched and commercial proprietary tools exist for its removal (Haskell and O'Donnell, 2001) should a project's quantitative specifications require it. Knowledge of a building's height is required to effect building lean removal.

The distribution of corner discrepancies (d) is shown in Figure 8.9. It is clear from the error distribution that the discrepancy bias (Arithmetic Mean of discrepancies) is about 2.23m.

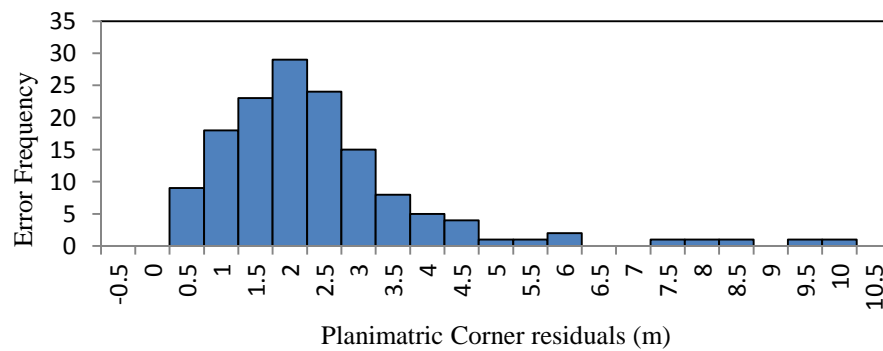


Figure 8.9 The building corner error (discrepancy or residual) distribution for 144 corner measurements.

These results were based on the planimetric discrepancies (residuals) of corners, but it is possible to go deeper into the analysis and consider the X, Y discrepancies of the corners individually, see Figure 8.10.

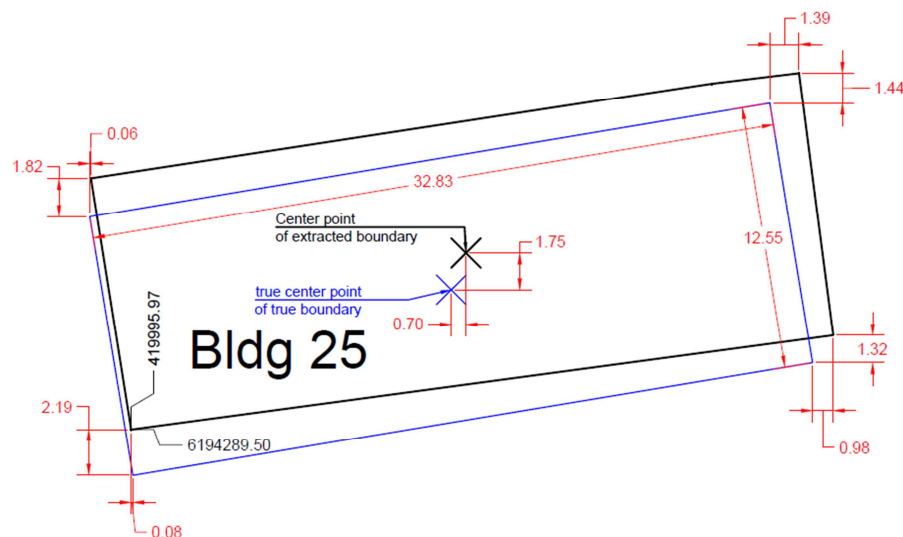


Figure 8.10 The constructed and true building corners, with corner discrepancy measurements in x and y directions. (All dimensions and coordinates are meters)

For each building the X and Y discrepancies have been measured, and the statistics of these measurements are shown in Table 8-2, see APPENDIX E.

Table 8-2 X and Y Corners discrepancies of buildings' - statistical results obtained from measuring 144 building corner points.		
Type of statistical test	X- discrepancy in meters	Y- discrepancy in meters
Min discrepancy (m) i.e. most negative	-4.76	-9.47
Max discrepancy (m) i.e. most positive	7.27	8.63
Mean discrepancy (m)	0.76	1.29
σ of discrepancy (m)	1.49	1.79
RMSE (m)	1.67	2.20

From the above analysis it is clear that the displacement or bias of X is smaller than Y. This is due to the lean of the building and the image exposure angle. The images used are not true orthoimages, which consequently leads to the roofs of the buildings being displaced from their original location. In addition the displacement value signs are positive, so the displacement is it to the north east. The RMSE value in X is smaller than in the Y direction, which is 1.67m in X and 2.20m in Y.

The histogram of the discrepancies has been evaluated as well, see Figure 8.11, in order to evaluate the distribution of the X and Y errors.

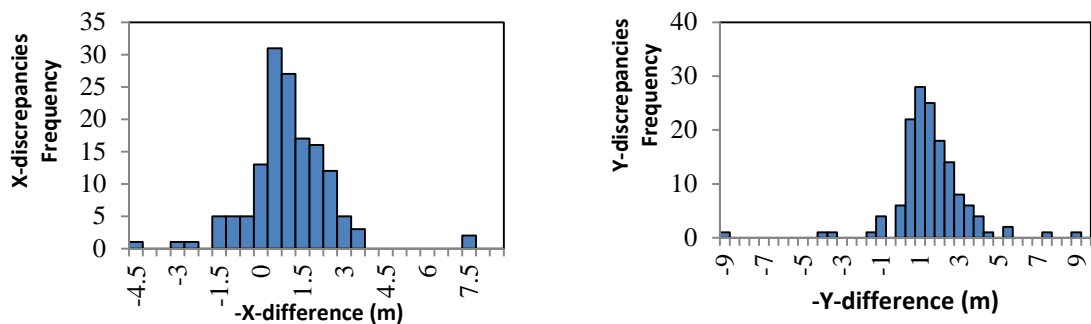


Figure 8.11 The corner error distribution of the buildings from measuring 144 corners.

The details of the measured values have been stated in Appendix E.

8.4.3 Building's Centre Validation

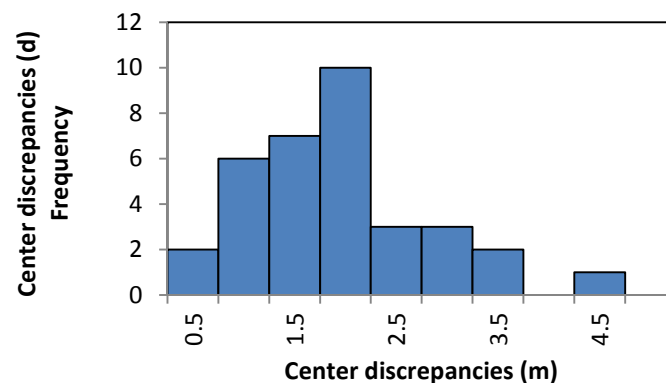
In addition to the corner displacement, it was also possible to apply a quantitative analysis of the error of the building's centre. The centre of each building, constructed and true, has been determined, by using AutoCAD software, and the discrepancy calculated. The planimetric discrepancy of the centre, as shown in the Figure 8.7, is mainly to south and west except for two buildings (bldg 26 and 27), whose discrepancy was only to the south.

As mentioned earlier this bias of the centre of the building is due to the acquisition angle of the satellite image and the fact that the building extraction process has focused on the roof of the buildings. The displacement of the centre has been evaluated twice, first the total planimetric discrepancy and then, individually, the discrepancies in X and Y. The overall validation results are shown in Table 8-3.

Table 8-3 Planimetric statistical result for the discrepancy (m) at each building centre for 35 buildings.

Min discrepancy (m)	0.35
Max discrepancy (m)	4.05
Arithmetic Mean discrepancy (m)	1.66
σ discrepancy (m)	0.87
RMSE (m)	1.84

The RMSE value is relatively small, at 1.84m, however there is still bias, as indicated by the arithmetic mean discrepancy which reaches to 1.66m - about three pixels. The histogram of the centre discrepancies can be shown in Figure 8.12.

**Figure 8.12 The discrepancies between the constructed and true Centre of the buildings for 35 buildings.**

The graph shows the error is not distributed normally, due probably to the different type of buildings, such as the L shaped building, existing among the rectangular shapes.

The other test on the centre that has been applied is the discrepancy of the centre X, and Y coordinates, individually. The individual X and Y discrepancies have been evaluated and the statistics are shown in Table 8-4,

Table 8-4 Individual X and Y coordinate discrepancies (m) of building centres - quantitative results for 35 buildings.

Type of statistical test	X- discrepancies (m)	Y- discrepancies(m)
Min discrepancy (m)	-0.96	-0.22
Max discrepancy (m)	2.15	4.01
Mean discrepancy (m)	0.61	1.38
σ of discrepancies (m)	0.59	0.95
RMSE (m)	0.84	1.64

The histogram of the discrepancies has also been evaluated in order to give an indication as to how the errors have been distributed. Figure 8.13 shows X and Y discrepancy distributions. It is clear from the histograms that the errors are not normally distributed, probably due to the buildings having different shapes therefore their centres discrepancy are different.

The details of the measured values have been stated in Appendix E.

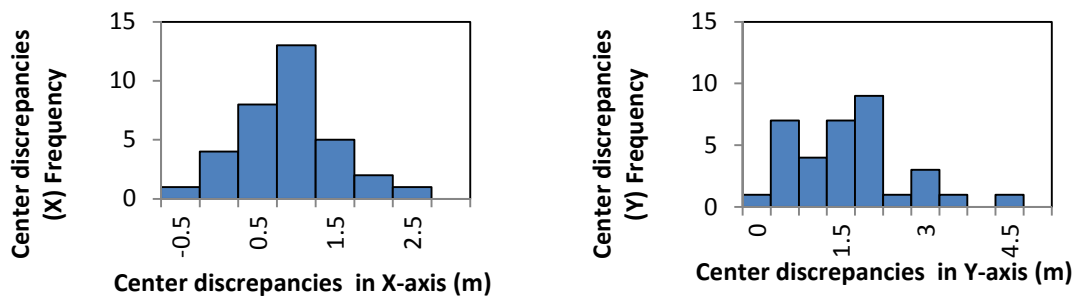


Figure 8.13 The Centre error distribution of the 35 buildings.

8.4.4 Area Validation

The next assessment is related to the percentage of extracted areas. Areas have been calculated in the AutoCAD environment. In this evaluation, the amount of the area that is represented as buildings is taken into consideration. At this stage the area that has been extracted as building through the process developed in this research is compared to the actual (or 'true') area, in order to test the percentage of the area that has been missed from or added to the total extracted area. The true area for the buildings in the first study area is calculated using the boundary that was obtained from total station coordinates, while for the second study area, they have been calculated by digitizing the boundary of the buildings on the ortho imagery. The quantitative tests applied are shown in the table below:

Table 8-5 Quantitative results for the discrepancies (m^2) of the building areas for 35 buildings.	
Min. area discrepancy of an extracted building (m^2) – i.e. most negative discrepancy	-210.66
Max. area discrepancy of an extracted building (m^2) – i.e. most positive discrepancy	154.50
Total 'true' area of buildings (m^2)	5912.91
Total extracted area of buildings (m^2)	6209.54
Difference (extracted - true) (m^2)	296.62
% Difference (extracted - true)	4.89% increase

The histogram of the difference in the area is shown in the below Figure 8.14

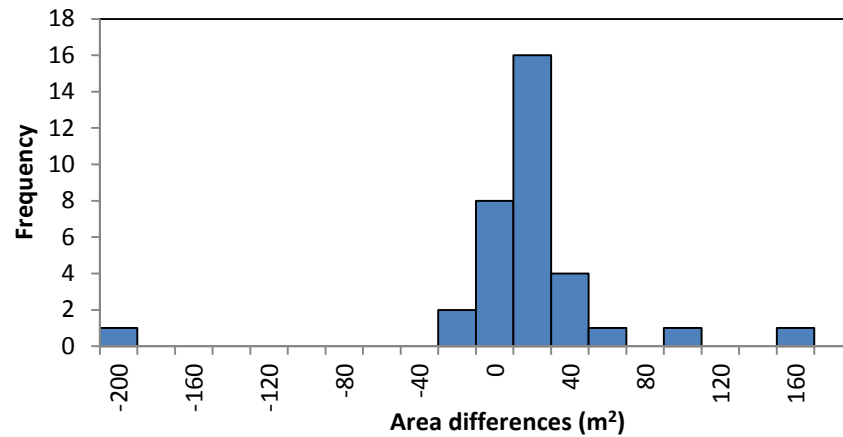


Figure 8.14 The error distribution of the building areas for 35 buildings.

It can be seen that the error is nearly normally distributed, but shows a bias and that the extracted area is consistently larger than the 'true' building area. The total area of the extracted buildings is higher than the 'true' area by 4.89%, which is larger than can be accounted for when considering that the extracted area is based on the roofline, whereas the 'true' building area uses the measured footprint.

The details of the measured values have been stated in Appendix E.

8.5 Quality Property Assessment

It is necessary to estimate, quantitatively, the practicability and the efficiency of the proposed process and its related algorithms. For that purpose, it was necessary to define some measurements that represent the operational quality of the developed process and its associated algorithms. Agouris et al., (2004) illustrated some accuracy measurements that have been used to evaluate building extraction rates. They show that some measurements should be defined, such as:

- the footprint area extracted correctly, denoted as true positive (TP);
 - the footprint area extracted incorrectly, denoted as false positive (FP);
- and finally,
- the footprint area that remained un-extracted incorrectly, denoted as false negative (FN).

From these measured values different quality indices can be obtained, for example

Correctness, Completeness and Quality, see equations 8-1, 8-2 and 8-3.

$$BE = \left(\frac{TP}{TP + FP} \right) \% \quad 8-1$$

$$CM = \left(\frac{TP}{TP + FN} \right) \% \quad 8-2$$

$$BDP = \left(\frac{TP}{TP + FP + FN} \right) \% \quad 8-3$$

Where:

BE is Building Extraction or Correctness;

CM is Completeness; and

BDP is Buildings Detected Percentage or Quality.

For the assessment of surface-area quality, two areas have been used in the quality assessment analysis, namely area-1 and area-2 as shown in Figure 8.1. For area-2 the true building boundaries have been obtained by digitizing the buildings using ArcGIS with the aid of a raster version of Ordnance Survey Master Map see Figure 8.15(b).

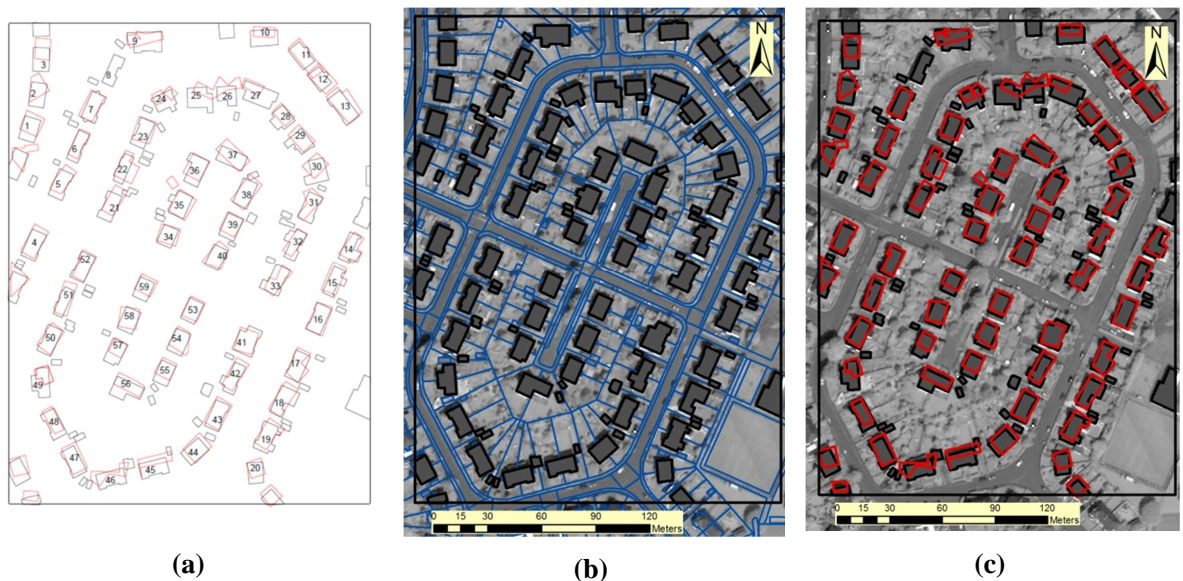


Figure 8.15 The second study area: (a) the numbering system for true building areas overlaid on the extracted buildings; (b) the true buildings digitized using ArcGIS; (c) the extracted buildings overlaid on the true buildings.

The correctness and completeness are complementary to each other and their values are between [0,1], or could be expressed in percentage terms. The Building Detected Percentage rate (BDP) statistic (equation 8-3) is considered more expressive than the others. These practicability and efficiency values have been calculated and presented in Table 8-6.

Table 8-6 Quality Property Assessment

Area	# of building	TP m ²	FP m ²	FN m ²	Correctness BE	Completeness CM	Quality BDP
1	35	4912.09	1019.119	1297.443	82.8%	79.1%	68.0%
2	58	5112.664	2451.737	1720.094	67.6%	74.8%	55.1%

The number of buildings extracted (excluding buildings located at the border of the study area, since they are not completely generated) and used in analysis for Area-1 is 35 buildings, see Figure 8.7, and for Area-2 is 58 buildings, excluding one building, number-8, as shown in Figure 8.16 (a). The analysis presented in Table 6-1 shows that the Quality Assessment statistics with respect to all indices (BE, CM and BDP) are higher in study Area-1 than study Area-2. This is due to the buildings in study Area-2 being densely distributed and very close to each other, as shown in Figure 7.40 (a). This consequently causes the initial building boundary produced by the segmentation process to be not fully extracted. Visual inspection shows that most of the buildings have been clearly identified.

The buildings that are detected are limited to buildings having a dimension larger than 4-meters, as shown Figure 8.4 and Figure 8.15 (c). This means that small buildings, as exist in area-2 Figure 8.15(c) are not detected, due to the resolution of the images being 50cm and the application of Mathematical Morphology (especially eroding more than once) causing small buildings to be missed.

The algorithm used is giving reasonable results if compared to other automatically applied algorithms. For instance the algorithm used by Aytekin et al. (2009) reached 80.8% with respect to CM (slightly higher than that achieved by the author's algorithm) and 34.5% with respect to BDP which is considerably less than that achieved by the algorithm used in this research.

The algorithm that has been implemented by Jin and Davis (2005) shows that the CM reached 72.7% and the BDP reached 58.5%.

However a better result has been obtained by applying supervised classification as the algorithm used by San and Turker (2010) had better accuracy since they used supervised classification and the CM reached 95.34% and the BDP reached 79.05%.

8.6 3D model assessment

Following the evaluation of planimetry described in preceding sections, the 3D models extracted following the proposed processes and their associated algorithms, now have their elevations assessed. The heights that are being assessed have been inferred from their DSMs. The DSMs being evaluated are from WorldView-1 and Pleiades as generated using Socet GXP, and three merged DSMs, namely, that resulting from merging using a Maximum Likelihood approach and two resulting from merging using a Bayesian approach. The summary of the quantitative tests for these five different 3D-models is shown in Table 8.7.

To achieve the validation for the constructed 3D buildings, a single measurement on the top of the building has been taken using a total station. The points that have been measured belonged to the highest building point, it having been selected in order to evaluate it against the highest elevation that will be taken using the constructed and merged DSMs.

As has been mentioned earlier, there was only one DSM generated with the Maximum Likelihood method, meanwhile, six types of DSM were generated using the Bayesian approach. In the validation just two DSMs have been used that belong to the Bayesian approach; one of them belongs to the DSM generated using the *a priori* range $\pm 0.1\text{m}$ and the window size 3x3 and the other were from *a priori* simulation range $\pm 0.25\text{m}$ with window size 7x7. The reason for selecting these two is because they have the lowest RMSE values: 0.420m for the window 3x3 and 0.419m for the window 7x7.

The validation has been achieved at three different stages on the study area, first covering all the buildings, then the buildings that have inclined roofs and finally the buildings with flat roofs (including among the ‘flat roofs’ those with a roof having a small incline to one side, only).

Table 8-7 3D statistical test using different sources for elevation for buildings with flat (one direction sloped) roofs only, showing discrepancies (d) obtained against measured points using a total station. Total number of measured buildings is 9

Type of statistical test	WV-1 DSM	Pleiades DSM	Maximum Likelihood merging DSM	Bayesian Merging DSM range $\pm 0.1\text{m}$	Bayesian Merging DSM range $\pm 0.25\text{m}$
Min. d (m)	-0.17	-3.69	-0.75	-0.73	-0.71
Max. d (m)	1.20	0.52	1.16	1.22	1.22
Arithmetic Mean d (m)	0.44	-1.48	0.36	0.40	0.39
σ of d (m)	0.52	1.33	0.68	0.68	0.68
RMSE using discrepancy (m)	0.66	1.94	0.73	0.76	0.75

Table 8-8 3D statistical test using different sources for elevation for buildings with hipped roofs only, showing discrepancies (d) obtained against measured points using a total station. Total number of measured buildings is 26

Type of statistical test	WV-1 DSM	Pleiades DSM	Maximum Likelihood merging DSM	Bayesian Merging DSM range $\pm 0.1\text{m}$	Bayesian Merging DSM range $\pm 0.25\text{m}$
Min. d (m)	-1.36	-2.80	-1.33	-1.28	-1.20
Max. d (m)	5.92	2.97	5.77	5.77	5.74
Mean d (m)	2.30	0.45	2.30	2.32	2.34
σ of d (m)	1.98	1.53	1.69	1.68	1.67
RMSE using discrepancy (m)	3.01	1.56	2.83	2.84	2.86

Table 8-9 3D Statistical test using different sources for elevation for all buildings (hipped and one direction sloped roofs), showing discrepancies (d) obtained against measured points using a total station. Total number of measured buildings is 35

Type of statistical test	WV-1 DSM	Pleiades DSM	Maximum Likelihood merging DSM	Bayesian Merging DSM range $\pm 0.1\text{m}$	Bayesian Merging DSM range $\pm 0.25\text{m}$
Min. d (m)	-1.36	-3.69	-1.33	-1.28	-1.20
Max. d (m)	5.92	2.97	5.77	5.77	5.74
Mean d (m)	1.82	-0.05	1.80	1.82	1.84
σ of d (m)	1.90	1.69	1.71	1.71	1.71
RMSE using discrepancy (m)	2.62	1.67	2.47	2.48	2.49

The histogram of the errors also has been analysed in order to see how the errors are distributed among each of the datasets. Figure 8.16 shows the distribution of error for each of the generated 3D models.

The details of the measured values have been stated in Appendix E.

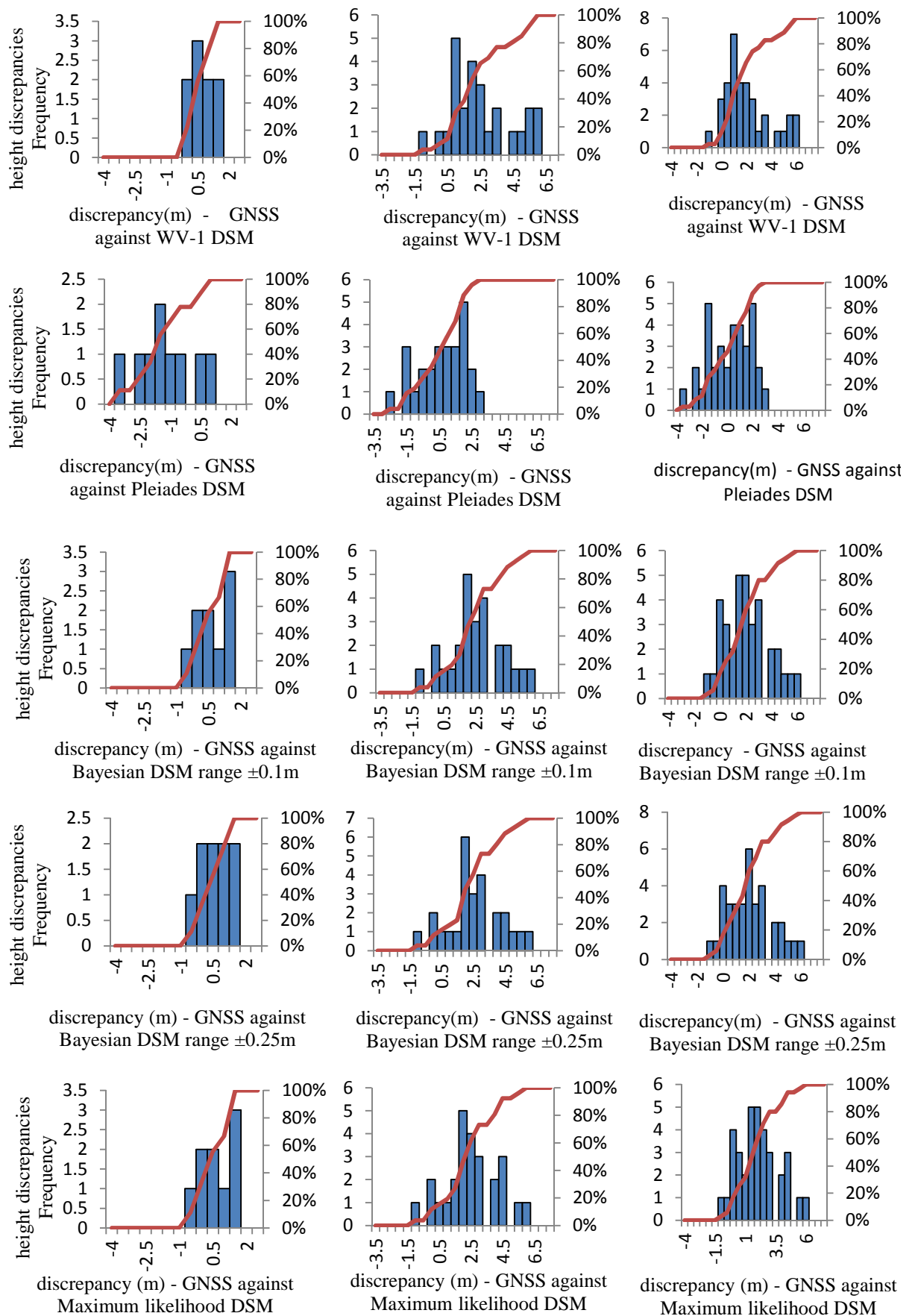


Figure 8.16 The height error distribution and Cumulative Distribution Function (CDF) of the buildings: left column the histogram of error for the flat roofed (one-side slope) buildings - total 9 samples; middle column for the hipped roof buildings - total 26 samples; and, the right column for all buildings (one direction slope and hipped roof buildings) - total 35 samples.

The frequency histogram shows that the error in the flat (or roof with almost no slope and that in only one direction) is less than for hipped roofed buildings and all buildings together. This high accuracy in the building with little roof's slope is because the image matching is more successful in the one-direction-slope near-flat roof.

From the Figure 8.16 (e.g. **Cumulative Distribution Function**) it can be noticed that the buildings obtained from Pleiades DSM having larger negative discrepancies than buildings from WV-1 DSM, with respect to all types of buildings (flat, hipped and mixed buildings). This is due to the large amount of bias in the Pleiades DSM, as illustrated in Table 6-1; it can be noticed that the bias of Pleiades is -0.629m compared to WV-1 DSM which is -0.170m.

Regarding the discrepancies in the buildings of the Merged DSMs, it can be noticed that with respect to 90% of buildings the residual is 1.5m for all buildings from three types of DSMs (i.e. Bayesian DSMs with both ranges $\pm 0.25\text{m}$, and $\pm 0.10\text{m}$ and Maximum likelihood DSM).

The discrepancies are different in the hipped roof buildings, 90% of building having residual 4.5m for both Bayesian DSM with range $\pm 0.10\text{m}$ and Maximum likelihood DSM while the discrepancy was larger reaching 5m with respect to the Bayesian DSM with range $\pm 0.10\text{m}$.

Similar discrepancies were found in the mixed type of buildings. The residual is 4.5m for 90% of buildings obtained using the Bayesian DSM with range $\pm 0.10\text{m}$. The residual has decreased to reach 4m with respect to 90% for buildings obtained from Bayesian DSM with range $\pm 0.25\text{m}$ and Maximum likelihood.

According to the study by Sadeq et al., (2012) correlation between the surface slope and the increased discrepancy has formerly been noted.

In addition, the WV-1 image has better results than the Pleiades image although the Pleiades image is multispectral; but the Pleiades GSD is 0.7m resampled to a resolution of 0.5m, whereas the original resolution of WV-1 is 0.5m. This leads to the image matching being more successful for WV-1 data since the original image resolution is higher there is greater image texture and better image matching - consequently more precise and accurate DSM can be produced.

8.7 Conclusion

This chapter has been limited to showing the results of analyses of the quality of the building footprints and 3D models, as generated using the processes and associated algorithms developed in this research. The visual inspection shows that the developed process detected different roof textures, i.e. either specified to be high intensity or low intensity, also detected whether the roof is highly sloped in both directions (hipped) or is flat (i.e. with a low slope in one direction).

Two different areas have been used to test the algorithm, although the average of the BDP is 61.55% it can be noted the average of the building extraction rate for both areas is much higher reaching 94%.

Also, the quantitative assessment shows a bias in both building footprint corners and building centres, which is due to building lean caused by the inclination of the satellite optical axis, and consequently this bias affects the result of quality assessment.

The height validation results, regarding the building 3D models, were better with respect to the flat roofed buildings than hipped roofed buildings.

The reason of the lower accuracy in the hipped roofed buildings is probably due to the relative high slope compared to the other buildings consequently affecting the DSM production using NGATE technique, which has been mentioned in section 3.6.1

Chapter 9 Conclusions and Future Work

This research has been devoted to developing a tool for merging numerous DSMs and constructing building footprints and their 3D models, aiming for an efficient method that can be applied worldwide. The research is based on using satellite image data, which is available for anywhere on earth. The tool was applied in the case study area and it gave reasonable results with respect to both qualitative and quantitative evaluations. This chapter will focus on the conclusion of the research and the future work that is expected to enhance the results of this research.

The proposed merging algorithm using Bayesian approach or ML can be used successfully in reducing data redundancy. In addition, the Bayesian approach can increase the quality of the merged data if the proposed recommendations in the forthcoming section are followed. Moreover, building footprint extraction and 3D modelling have shown reliable results based on satellite imagery. Also the highest accuracies have been shown to be among the areas that have flat roofs as in the case of the buildings in Iraq. Also it is possible to get more less-cloudy data in Iraq than the study area, Glasgow, consequently leading to a more up-to-date 3D city model.

9.1 Conclusion

In this research a statistical approach, based on probabilistic methods, has been investigated for the merging DSMs and enhancing building footprints.

Due to increasing the sources of DSM construction, merging DSMs can reduce data redundancy meanwhile improving the quality of the data. The applied statistical tests have shown that merging using a Bayesian method can provide DSM results similar to those achieved using a Maximum Likelihood method for merging, and it can be used for its intended purpose. It was hoped to obtain a DSM that had better characteristics than the original. Noticeable improvements in accuracy cannot, theoretically, be achieved, and purely on the positional accuracy basis the unmerged WorldView-1 DSM offers the greatest accuracy, but a more complete model can be achieved, and an approach is offered which can be used to detect blunders in a contributing DSM. Also the Bayesian approach has helped to smooth the surface of the generated structures so it may represent the real surfaces found in urban areas better. It can be acknowledged that the effect of misregistration has not been treated and the result would be more accurate if this was addressed. But the more complete DSM achieved is likely to assist greatly in a variety of applications, when this problem is addressed.

The source of the data used was limited to the very high resolution satellite images. The research commenced by acquiring and processing satellite images from WorldView-1 and Pleiades. For the purpose of the merging DSMs, the quality of each of them has been evaluated using checkpoints obtained from Differential GNSS for the purpose of the weighting the DSMs. It was noticed that the accuracy of the DSM of WV-1 is better than the DSM of Pleiades since the geometry (i.e. base-to-height ratio) of WV-1 is better than that of Pleiades. For the merging purpose, a model has been developed based on using a Bayesian approach, the *a priori* information that is required in a Bayesian approach has been established by maximizing entropy. In addition a more conventional merging method has been used also, based on the Maximum Likelihood method (i.e. weighted average).

The approach that has been developed is able to merge two or more DSMs, thus integrating the characteristics from various DSM into one single DSM. The suggested Bayesian method aimed to address and overcome the problems that arise from image matching during DSM production, in particular those caused by occlusions and shadow areas. From the application of the DSMs in 3D modelling, it can be noticed that there is very slight difference, in the produced 3D buildings model, using DSMs that obtained from ML and Bayesian approaches.

The scope of the research was extended in order to cover constructing rectilinear building footprints and 3D modelling. The algorithm, using a Bayesian approach, successfully constructed buildings footprints which represent LoD0 and constructed a 3D city model which represents LoD1. The algorithm has utilized both the DSM and orthoimagery. Regarding the 3D modelling, the original and merged DSMs have been used individually. Subsequently the algorithm was been evaluated qualitatively and quantitatively. Both evaluations gave promising results and consequently can be used at any place having similar characteristics to the study area regarding building roof intensity. Alternatively some of the parameters can be modified that relate to constructing the initial building footprint, which can be considered to be the key step in building detection. Moreover the Bayesian approach has proved its robustness in regularizing building footprints by producing shapes that have straight edges and right-angled corners, a common feature in most buildings, thus it is suitable for the purposes of planning and 3D modelling, especially for areas traditionally mapped at large scale.

The method developed is semi-automated and depends on just a few parameters to detect the building. The only part that is manual is interactive road detection, but as automated road detection is also a research area undergoing parallel development for which a successful outcome is expected, looking to the future this interactivity is unlikely to be required. Thus, this method is transferable to any area in the world using imagery that has similar sensor characteristics.

The novelty of this part of the study is introducing a Bayesian approach to refining building footprint boundaries using satellite image data. The approach was able to detect all the buildings, with different accuracy results, and, importantly, the benefit of this method is that it is, potentially relatively low cost since it is dependent on a single source of data for processing when incorporating the Bayesian approach. Due to dependency on image texture for segmentation, the algorithm was not able to detect the buildings by applying the same thresholding method in different areas. Additionally due to the buildings being oriented differently, it was necessary to use different structural elements with the mathematical morphology operations.

DSMs can be used in different planning, engineering and remote sensing applications, therefore merging DSMs increases their potential usefulness since aspects of quality and also completeness are increasing. However, it is expected that more robust results can be obtained by enhancing the approaches explored in this research following the suggestions and recommendations given in the following section.

9.2 Future Work

As it is clear from the dissertation and the conclusion, the work is pertinent to merging DSMs and building footprint extraction. Therefore the future work suggested here will address both aspects. There are many suggestions that can be made to fill the gaps that exist in this research in order to improve the result.

9.2.1 Further work for Merging DSMs

The merging technique used which is based on a probabilistic method, the Bayesian approach, can be improved by amending the ingredients that have been used in the model, such as the *a priori* information; further than that the model does not need to be modified. Although the cost of implementing the merging was ideal, the following section offers further experiments which might provide results leading to a better outcome:

1. DSM Quality can be considered as one of the main contributing factors of the

probabilistic method. It is assumed that quality is considered a constant over the whole DSM, since the quality is utilized to add weight to each DSM. However, *it could be better to make it pixel based*. Using error propagation to distribute the accuracy of the ground points over all a DSM, will obviate the need to depend on a global RMSE value.

2. In addition to using accuracy from ground points, *it is valuable to test the accuracy estimation using geo-morphological specifications such as slope and aspect*, and even land-cover in order to introduce more robust methods for assigning accuracy to the DSMs, instead of only depending on the ground points.
3. Concerning the *a priori* information, *investigate merging based on using total variation (TV)* to minimize the roughness in the surface instead of, or integrated with local entropy. TV is widely used in signal processing to smooth the signal meanwhile preserving the discontinuities in the data, which is in this case represented by change of the elevation.
4. The DSMs are based on global co-registration, however, there is potential displacement in the buildings which may be overcome by achieving *co-registration based on the patches* produced in an earlier step. Thus will minimize the variance that exists in each pixel. Consequently, this will lead to removing the systematic error from the merged DSMs, and consequently improve the result.
5. The study area was not suffered any changes during the period of the image taken, therefore the multi-temporal effect has not taken in consideration, however it is important to *consider including the multi-temporaleffect* in the model since the source DSMs may relate to different dates and seasons.

This problem probably can be solved, as referred in 2.3.2, by producing a residuals map, and if the height in the specific grid points are different then the height of the latest DSM will assigned to that specific grid point.

6. Since the DSMs used were produced from the same software, it was possible to control the resolution of the DSMs and make them uniform. However, it is important to *consider multi-resolution data* and to enable the use of different resolutions of data, by resampling the lower resolution data to the same resolution if the higher resolution.
7. Using information that has been generated during image matching and *incorporating the extra detail resulting from image matching into the DSM quality assessment* for each individual pixel instead of (or as well as) using ground truth data for the DSM quality assessment.

9.2.2 Further work for building footprint extraction

3D modelling has proved its importance for different engineering and environmental fields (Gröger and Plümer, 2012). The approach that has been presented in this thesis, which was based on satellite imagery only, was able to create a city model approximately based on levels of detail LoD0 and LoD1 (see 1.7).

The level of detail LoD0 represents volume objects with polygons and it is specified to be the coarsest representation of the city model, and for the level of detail LoD1 the buildings are represented by 3D block models consisting of a prismatic building type having a flat roof (Gröger and Plümer, 2012). Implementing the proposed approaches for 3D modelling will reduce the challenges currently existing for acquiring large scale 3D geodatabases.

LoD1 models can be used for flood simulation which is used for managing disasters. Monitoring and protecting against flood is becoming an important issue in developing and even in developed countries, causing death and displacement for large numbers of people, and billions of dollars of damage globally to property (Kokkas, 2008). LoD1 also can be used for preventing noise pollution. The LoD1 that has been generated from this research and overlaid on the DTM, can be used to produce a noise map when integrated with the roads, railways, airports and industrial sites (Czerwinski et al., 2007). Noise map analysis is considered as an important tool in the management of public health. LoD1 can also be implemented in urban planning for shadow analysis. Shadow analysis is important in order to find the relation between the planned building and the existing buildings and how this effect the aesthetics and suitability of the building with respect to its designed function, considered as necessary requirement for the design in some countries (Kokkas, 2008)

But in addition to the applications mentioned in the preceding paragraph, it must be remembered that there are other usages for 3D modelling which challenge the value of LoD1 because of the required higher level of detail which is currently limited to aerial imagery or LiDAR data.

Recently the 3D City model has been developed widely in tourism industry in order to help the tourist in order to plan their city visits. However for the virtual city modelling the more complex LoD3 or LoD4 models are required, although occasionally LoD2 is used (Kokkas, 2008). Also 3D modelling has been used for simulation and training purposes. But, for example, for simulating airport areas the level of detail is limited to LoD3 for simulating the

airport area while a lower level of detail, LoD2, was appropriate for other areas (Kokkas, 2008).

As shown in Fig. 7.37, data generated through the processes developed in this research can be directly imported into a standard GIS such as ArcGIS, for use by those involved in the implementation of flood mitigation, noise mapping, planning and tourism services.

However, the following suggestions could be examined to address deficiencies, so that a better accuracy can consequently be achieved and the products incorporated more effectively into a database.

1. The accuracy of the building footprint can be increased by *using orthoimagery that has a lower nadir angle*. Alternatively, by *removing the bias that exists due to building lean*, producing a so-called ‘true-orthoimage’, consequently leading to increased accuracy.
2. The time that is required to simulate the building can be decreased by *introducing some statistical methods for speeding up the iteration and getting results more quickly* such as a stochastic optimization method, for example simulated annealing (Chatterjee and Siarry, 2013). This method has the ability to find the global maximum rather than local optima only.
3. Examining image enhancing techniques in order to *improve the image contrast for the purpose of producing better building edges*, consequently leading to improved results during the optimization process.
4. Investigate more robust methods for object based analysis, e.g. *investigate the level set method for image segmentation* instead of the general threshold method for the purpose of getting initial building footprint, since it is considered to give better results than the classical threshold method due to the absence of noise and that there is no need for any morphological operation.
5. Convert the algorithm to full automation by *changing the interactive road detection into automated road detection*.
6. *Test the algorithm on higher resolution satellite imagery* (e.g. WorldView-3 resolution 0.31m, if available) in order to get better accuracy and better building shapes than in the current reported work.
7. *Introduce texture mapping* for the constructed 3D building by inferring it from the available stereo images, in order to make the 3D more realistic.
8. *Applying the algorithm to an extensive area* such as a whole city and incorporating the

result in topographical database for the purpose of GIS applications.

9.3 Revisiting Objectives

The general aim of this research has been to investigate merging DSMs using a Bayesian approach and extracting building footprints and 3D modelling based on satellite images also incorporating Bayesian approaches. Specifically all the objectives as presented in the introduction section 1.51.6, namely:

1. to evaluate existing approaches to DSM merging and building extraction, essentially through a critical literature review;
2. to generate DSMs using a technique, from among the techniques that are listed in Table 2-1, that can produce high resolution DSMs from high resolution images produced from different sensors, and is efficient;
3. to develop an optimal procedure to **merge** the high resolution DSMs derived from high resolution satellite imagery, particularly examining methods based on **Bayesian theory**, resulting in equal or better quality than the original DSMs;
4. to validate the resultant DSMs arising from the proposed merging model using Bayesian approaches;
5. to upgrade procedures to extract building footprints from high resolution satellite imagery;
6. to apply the developed building extraction approach to the merged DSM to facilitate automated 3D model generation at LoD1; and,
7. to validate the performance and quality of the developed approach at a selected test site.

This chapter has suggested that although all seven objectives have been addressed and some improvements discerned when compared to more established processes, better results could be obtained by pursuing some further experimentation as recommended in sections. 9.2.1 and 9.2.2.

References

- Abdelfattah, R. and Nicolas, J.M. (2002), "Topographic SAR interferometry formulation for high-precision DEM generation", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 40 No. 11, pp. 2415–2426.
- Abdullah, B.Q. (2013), "Mapping Matters", *American Society for Photogrammetry and Remote Sensing-webinar "Lidar Fundamental and Applications"*, Vol. 79 No. 4.
- Ackermann, F. (1984), "Digital Image Correlation: Performance and Potential Application in Photogrammetry", *Photogrammetric Record*, Vol. 11, pp. 429–439.
- Agouris, P., Doucette, P. and Stefanidis, A. (2004), "Automation and Digital Photogrammetric Workstation", *Manual of Photogrammetry, Fifth Edition, American Society for Photogrammetry and Remote Sensing (edited by McGlone, J.C., Mikhail E.M., and J. Bethel)*, pp. 949–981.
- Ahmadi, S., Zoej, M.J.V., Ebadi, H., Moghaddam, H.A. and Mohammadzadeh, A. (2010), "Automatic urban building boundary extraction from high resolution aerial images using an innovative model of active contours", *International Journal of Applied Earth Observation and Geoinformation*, Elsevier B.V., Vol. 12 No. 3, pp. 150–157.
- Ahmed, B. and Rahman, M. (2011), "Automatic road extractions from high resolution satellite imagery using road intersection model in urban areas", *Computer Engineering and Intelligent Systems*, Vol. 2 No. 4, pp. 82–94.
- Akca, D. (2007), *Least squares 3D surface matching. (PhD thesis)*, Swiss Federal Institute of Technology Zurich, available at:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.68.7716&rep=rep1&type=pdf> (accessed 31 January 2013).
- Ali, M. and Clausi, D. (2001), "Using the Canny edge detector for feature extraction and enhancement of remote sensing images", *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No.01CH37217)*, IEEE, Sydney, Australia, Vol. 05, pp. 2298–2300.
- Alobeid, A. (2011), *Assessment of Matching Algorithms for Urban DSM Generation from Very High Resolution Satellite Stereo Images. PhD Thesis, Wissenschaftliche Arbeiten der Fachrichtung Geodäsie und Geoinformatik der Leibniz Universität Hannover ; 293, Fachrichtung Geodäsie und Geoinformatik der Leibniz-Univ.*
- Alobeid, A. and Jacobsen, K. (2008), "Automatic generation of digital surface models from IKONOS stereo imagery and related application", *GORS 16th International Symposium, Damascus*.
- Andrews, L.C. and Phillips, R.L. (2003), *Mathematical Techniques for Engineers and Scientists*, SPIE, 1000 20th Street, Bellingham, WA 98227-0010 USA, doi:10.1117/3.467443.

- Ang, A.H.-S. and Tang, W.H. (1975), *Probability concepts in engineering planning and design*, John Wiley & Sons, New York, John Wiley & Sons, New York, Vol 1.
- Anscombe, F.J. (1962), “Bayesian Statistics”, *American Statistician*, Vol. 15.
- ASPRS. (2015), *ASPRS Positional Accuracy Standards for Digital Geospatial Data, Photogrammetric Engineering & Remote Sensing*, Vol. 81, doi:10.14358/PERS.81.3.A1-A26.
- “ASPRS Accuracy Standards for Digital Geospatial Data.” (2013), *American Society for Photogrammetry and Remote Sensing*, Vol. 79 No. 12, pp. 1073–1085.
- As-syakur, A.R., Adnyana, I.W.S., Arthana, I.W. and Nuarsa, I.W. (2012), “Enhanced Built-Up and Bareness Index (EBBI) for Mapping Built-Up and Bare Land in an Urban Area”, *Remote Sensing*, Vol. 4 No. 12, pp. 2957–2970.
- Astrium Services. (2012), “Pléiades Imagery User Guide”, available at: <http://www.astrium-geo.com/pleiades/> (accessed 7 November 2014).
- Astrium Services. (2013), “Pléiades Spot the Detail”, available at: http://www2.astrium-geo.com/files/pmedia/public/r61_9_geo_011_pleiades_en_low.pdf (accessed 9 October 2014).
- Awrangjeb, M., Zhang, C. and Fraser, C.S. (2013), “Automatic extraction of building roofs using LIDAR data and multispectral imagery”, *ISPRS Journal of Photogrammetry and Remote Sensing*, International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS), Vol. 83, pp. 1–18.
- Aytekin, O., Ulusoy, I., Erener, A. and Duzgun, H.S.B. (2009), “Automatic and unsupervised building extraction in complex urban environments from multi spectral satellite imagery”, *2009 4th International Conference on Recent Advances in Space Technologies*, IEEE, pp. 287–291.
- Bacher, U. and Mayer, H. (2005), “Automatic road extraction from multispectral high resolution satellite images”, *Stilla U, Rottensteiner F, Hinz S (Eds) CMRT05. IAPRS*, Vienna, Austria, Vol. XXXVI, pp. 29–34.
- Baltsavias, E., Li, Z. and Eisenbeiss, H. (2006), “DSM generation and interior orientation determination of IKONOS images using a testfield in Switzerland”, *Photogrammetrie, Fernerkundung, Geoinformation*, pp. 41–54.
- Baltsavias, E., Mason, S. and Stallmann, D. (1995), “Use of DTMs/DSMs and Orthoimages to Support Building Extraction”, in Gruen, A., Kuebler, O. and Agouris, P. (Eds.), *Automatic Extraction of Man-Made Objects from Aerial and Space Images*, Birkhäuser Basel, Basel, pp. 199–210.
- Barry, O. (2011), *Semi-Automatic Extraction of Information from Satellite Images (Master Thesis)*, Architecture, KTH University-Electrical Engineering.
- Battiato, S., Castorina, A. and Mancuso, M. (2002), “Method for merging digital images to obtain a high dynamic range digital image”, Vol. 1 No. 19, pp. 1389–1406.

- Beaumont, M. a and Rannala, B. (2004), “The Bayesian revolution in genetics.”, *Nature reviews. Genetics*, Vol. 5 No. 4, pp. 251–61.
- Benkouider, F., Hamami, L. and Abdellaoui, A. (2011), “Use of the Neural Net for Road Extraction from Satellite Images, Application in the City of Laghouat (Algria)”, *PIERS*, Vol. 7 No. 2, pp. 146–150.
- Bernacchia, A. (2014), “The interplay of plasticity and adaptation in neural circuits: a generative model”, *Frontiers in Synaptic Neuroscience*, Vol. 6 No. October, pp. 1–14.
- Bernardini, F. and Bajaj, C.L. (1997), “Sampling and Reconstructing Manifolds Using Alpha-Shapes.”
- Berry, D.A. (1997), *Using a Bayesian Approach in Medical Device Development, Institute of Statistics & Decision Sciences and Comprehensive Cancer Center, Duke University, Durham NC 27708-025*.
- Bertin, S., Friedrich, H., Delmas, P., Chan, E. and Gimel'farb, G. (2015), “Digital stereo photogrammetry for grain-scale monitoring of fluvial surfaces: Error evaluation and workflow optimisation”, *ISPRS Journal of Photogrammetry and Remote Sensing*, International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS), Vol. 101, pp. 193–208.
- Beyerer, J., Heizmann, M., Sander, J. and Gheta, I. (2011), “Bayesian methods for image fusion”, in Stathaki, T. (Ed.), *Image Fusion: Algorithms and Applications, Academic Press.*, pp. 157–192.
- Bilskie, M. V. and Hagen, S.C. (2013), “Topographic accuracy assessment of bare earth lidar-derived unstructured meshes”, *Advances in Water Resources*, Elsevier Ltd, Vol. 52, pp. 165–177.
- Boos, D.D. and Stefanski, L.A. (2013), *Essential Statistical Inference*, Springer Texts in Statistics, Springer New York, New York, NY, Vol. 120, doi:10.1007/978-1-4614-4818-1.
- Box, G.E.P. and Draper, N.R. (1987), “Empirical Model Building and Response Surfaces”, *New York. John Wiley and Sons, Inc.*
- Brunn, A. and Weidner, U. (1997), “Extracting buildings from digital surface models”, *International Archives of Photogrammetry and Remote Sensing, Part 3-4W2*, Stuttgart, Germany, Vol. 32, pp. 27–34.
- Bruzzone, L., Prieto, D.F. and Serpico, S.B. (1999), “A neural-statistical approach to multitemporal and multisource remote-sensing image classification”, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 37 No. 3, pp. 1350–1359.
- Burns, J.B., Hanson, A.R. and Riseman, E.M. (1986), “Extracting Straight Lines”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8 No. 4, pp. 425–455.
- Calitz, M.F. and Ruther, H. (1996), “Least Absolute Deviation (LAD) image matching”, *ISPRS Jo. of Photogrammetry and Remote Sensing*, Vol. 51 No. 5, pp. 223–229.

- Canny, J. (1986), "A computational approach to edge detection.", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 8 No. 6, pp. 679–98.
- Cardoso, L.A.L. da S. (1999), *Computer aided recognition of manmade structures*, MSc. dissertation, Monterey, USA.
- Carlin, B.P. and Louis, T.A. (1996), *Bayes and empirical Bayes methods for data analysis*, Chapman & Hall, London.
- Carlin, B.P. and Louis, T.A. (2000), *Bayes and Empirical Bayes Methods for Data Analysis*, London: Chapman & Hall, Second Edi.
- Carter, J., Schmid, K., Waters, K., Betzhold, L., Hadley, B., Mataosky, R. and Halleran., J. (2012), "Lidar 101: An Introduction to Lidar Technology, Data, and Applications.", (NOAA) Coastal Services Center, available at: http://coast.noaa.gov/digitalcoast/_/pdf/lidar101.pdf (accessed 9 October 2015).
- Chang, H., Ge, L. and Rizos, C. (2004), "Assessment of digital elevation models using RTK GPS", *Journal of Geospatial Engineering*, Vol. 6 No. 1.
- Chatterjee, A. and Siarry, P. (2013), *Computational Intelligence in Image Processing*, (Chatterjee, A. and Siarry, P.,Eds.), Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-642-30621-1.
- Chen, C. and Yue, T. (2010), "A method of DEM construction and related error analysis", *Computers & Geosciences*, Elsevier, Vol. 36 No. 6, pp. 717–725.
- Choi, S.Y., Kang, J.M. and Shin, D.S. (2012), "A Comparison Of Accuracies of the RPC Models: Homo- And Hetero-type Stereo Pairs Of Geoeye And Worldview Images", *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Melbourne, Australia, Vol. I–4, pp. 65–69.
- Christensen, R. (2005), "Testing Fisher, Neyman, Pearson, and Bayes", *The American Statistician*, Vol. 59 No. 2, pp. 121–126.
- Christophe, E. and Inglada, J. (2007), "Robust Road Extraction for High Resolution Satellite Images", *IEEE International Conference on Image Processing*, pp. V – 437–V – 440.
- Chunmei, W., Qinke, Y., Hongyan, L., Weiling, G., Jupp, D.L. and Rui, L. (2013), "Influence of resolution on elevation and slope at watershed scale in Loess Plateau.", *SpringerPlus*, Vol. 2 No. Suppl 1, p. S13.
- Comaniciu, D. and Meer, P. (2002), "Mean shift: a robust approach toward feature space analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24 No. 5, pp. 603–619.
- Costantini, M., Farina, A. and Zirilli, F. (1997), "The fusion of different resolution SAR images", *Proceedings of the IEEE*, Vol. 85 No. 1, pp. 139–146.
- Costantini, M., Malvarosa, F., Minati, E. and Zappitelli, E. (2006), "A Data Fusion Algorithm for DEM Mosaicking: Building a Global DEM with SRTM-X and ERS Data", *2006 IEEE International Symposium on Geoscience and Remote Sensing*, IEEE,

- pp. 3861–3864.
- Czerwinski, A., Sandmann, S., Stöcker-Meier, E. and Plümer, L. (2007), “Sustainable SDI for EU noise mapping in NRW – best practice for INSPIRE”, *International Journal for Spatial Data Infrastructure Research*, Vol. 2 No. 1.
- Dahiya, S., Garg, P.K. and Jat, M.K. (2012), “Object Oriented Approach for Building Extraction from High Resolution Satellite Images”, *International Advance Computing Conference (IACC)*, IEEE, pp. 1300–1305.
- Dahiya, S., Garg, P.K. and Jat, M.K. (2013), “Building Extraction from High Resolution Satellite Images”, *International Journal of Computing Science and Communication Technologies.*, Vol. 5 No. 2, pp. 829–834.
- Dai, W. and Wang, K. (2007), “An Image Edge Detection Algorithm Based on Local Entropy”, *2007 IEEE International Conference on Integration Technology*, Ieee, pp. 418–420.
- Damron, J.J. (1999), “Techniques for Digital Elevation Model (DEM) Fusion Using ARC/INFO: Using IFSAR /and LIDAR DEM Data”, *U.S. Army Topographic Engineering Center*.
- Deng, C. and Wu, C. (2012), “BCI: A biophysical composition index for remote sensing of urban environments”, *Remote Sensing of Environment*, Elsevier Inc., Vol. 127, pp. 247–259.
- DeVenecia, K., Walker, S. and Zhang, B. (2007), “New approaches to generating and processing high resolution elevation data with imagery”, *Photogramm. Week, Institute of Photogrammetry: Stuttgart, Germany.*
- Dial, G., Bowen, H., Gerlach, F., Grodecki, J. and Oleszczuk, R. (2003), “IKONOS satellite, imagery, and products”, *Remote Sensing of Environment*, Vol. 88 No. 1-2, pp. 23–36.
- Diebel, J.R., Thrun, S. and Brünig, M. (2006), “A Bayesian method for probable surface reconstruction and decimation”, *ACM Transactions on Graphics*, Vol. 25 No. 1, pp. 39–59.
- DigitalGlobe Inc. (1992), “DigitalGlobe”, available at: <http://www.digitalglobe.com/> (accessed 15 August 2014).
- DigitalGlobe Inc. (2013), *DigitalGlobe’s Core Imagery Product Guide*, available at: <http://www.digitalglobe.com/resources/product-data-sheets> (accessed 21 June 2014).
- Dorninger, P. and Pfeifer, N. (2008), “A Comprehensive Automated 3D Approach for Building Extraction, Reconstruction, and Regularization from Airborne Laser Scanning Point Clouds”, *Sensors*, Vol. 8 No. 11, pp. 7323–7343.
- Douglas, D.H. and Peucker, T.K. (1973), “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature”, *Cartographica: The International Journal for Geographic Information and Geovisualization*, Vol. 10 No. 2, pp. 112–122.

- Dowman, I. (2004), "Integration of LiDAR and IFSAR for mapping", *International Archives of Photogrammetry and Remote Sensing*, Istanbul, Turkey, Vol. XXXIV, pp. 90–100.
- Dowman, I., Jacobsen, K., Konecny, G. and Sandau, R. (2012), *High Resolution Optical Satellite Imagery*, Whittles Publishing, Scotland, UK.
- Downarowicz, T. (2011), *Entropy in dynamical systems*, *New Mathematical Monographs* 18, Cambridge University Press, Cambridge, available at: <http://www.maa.org/publications/maa-reviews/entropy-in-dynamical-systems> (accessed 19 June 2014).
- Doyle, F. (1978), "Digital Terrain Models : An Overview", *Photogrammetric Engineering and Remote Sensing*, Vol. 44 No. 12, pp. 1481–1485.
- Eisenbeiss, H., Baltsavias, E., Pateraki, M. and Zhang, L. (2004), "Potential of IKONOS and QuickBird imagery for accurate 3D point positioning, orthoimage and DSM generation", *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 35 No. Part B3.
- Electronics & Integrated Solutions. (2008), "Next-Generation Automatic Terrain Extraction (NGATE) Innovation in the cost-effective derivation of elevation data from imagery", *ASPRS 2006 Annual Conference, Reno, Nevada, 1 – 5 May*, BAE system.
- Elhifnawy, H., Elhabibya, M. and El-Sheimy, N. (2011), "Wavelet Building Extraction From Lidar Data", *Geomatics Technology in the City*.
- Eren, G., Aubreton, O., Meriaudeau, F., Sanchez, L.A., Fofi, D., Naskali, A.T., Truchetet, F., et al. (2009), "Scanning from heating : 3D shape estimation of transparent objects from local surface heating", *Opt. Express* 17, Vol. 17 No. 14, pp. 11457–11468.
- Fasbender, D., Radoux, J. and Bogaert, P. (2008), "Bayesian Data Fusion for Adaptable Image Pansharpening", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 46 No. 6, pp. 1847–1857.
- FDA. (2010), *Guidance for the Use of Bayesian Statistics in Medical Device Clinical Trials*, available at: www.fda.gov/OHRMS/DOCKETS/98fr/06d-0191-gdl0001.pdf (accessed 21 April 2014).
- Feron, O. and Mohammad-Djafari, A. (2004), "A Hidden Markov model for Bayesian data fusion of multivariate signals", *Journal of Electronic Imaging*, Data Analysis, Statistics and Probability, , Vol. 14 No. 2, pp. 1–14.
- Féron, O. and Mohammad-Djafari, A. (2005), "Image fusion and unsupervised joint segmentation using a HMM and MCMC algorithms", *Journal of Electronic Imaging*, Vol. 14 No. 2, doi:10.1117/1.1904066.
- Ferretti, a., Prati, C. and Rocca, F. (1999), "Multibaseline InSAR DEM reconstruction: the wavelet approach", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 37 No. 2, pp. 705–715.
- Ferretti, A., Monti-Guarnieri, A., Prati, C. and Rocca, F. (1998), "Multi-image DEM

- reconstruction”, *IGARSS '98. Sensing and Managing the Environment. 1998 IEEE International Geoscience and Remote Sensing. Symposium Proceedings. (Cat. No.98CH36174)*, IEEE, pp. 1367–1369 vol.3.
- Frajka, T. and Zeger, K. (2004), “Downsampling dependent upsampling of images”, *Signal Processing: Image Communication*, Vol. 19 No. 3, pp. 257–265.
- Fuss, C. (2013), *Digital Elevation Model Generation and Fusion*, Guelph, Ontario, Canada, available at: <https://atrium.lib.uoguelph.ca/xmlui/handle/10214/7571> (accessed 9 September 2014).
- Ge, Z., Wang, B. and Zhang, L. (2007), “Remote sensing image fusion based on Bayesian linear estimation”, *Science in China Series F: Information Sciences*, Vol. 50 No. 2, pp. 227–240.
- Gecena, R. and Sarpb, G. (2008), “Road Detection From High And Low Resolution Satellite Images”, *The International Archives of the Photogrammetry*, pp. 2–5.
- Gelman, A., Carlin, J., Stern, H. and Rubin, D. (2004), *Bayesian data analysis*, Chapman & Hall/CRC, 2nd ed.
- Ghannam, S., Awadallah, M., Abbott, A.L. and Wynne, R.H. (2014), “MULTISENSOR MULTITEMPORAL DATA FUSION USING THE WAVELET TRANSFORM”, *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.
- Gheta, I., Heizmann, M. and Beyerer, J. (2008), “Bayesian Fusion of Multivariate Image Series to Obtain Depth Information”, *Proceedings of Fusion*, Köln, pp. 1731–1737.
- Gill, J. (1994), “What is Entropy ? Entropy in Information Theory”, No. 1, pp. 1–5.
- Gonzalez, R.C. and Woods., R.E. (1992), *Digital image processing*, Addison-Wesley, Reading, MA ; Wokingham.
- Graham, C. and Talay, D. (2013), *Stochastic Simulation and Monte Carlo Methods*, Stochastic Modelling and Applied Probability, Springer Berlin Heidelberg, Berlin, Heidelberg, Vol. 68, doi:10.1007/978-3-642-39363-1.
- Gröger, G. and Plümer, L. (2012), “CityGML – Interoperable semantic 3D city models”, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 71, pp. 12–33.
- Gruen, A. and Akca, D. (2005), “Least squares 3D surface and curve matching”, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 59 No. 3, pp. 151–174.
- Gruen, A., Bär, S. and Bühner, T. (2000), “DTMs Derived Automatically From DIPS - Where Do We Stand?”, *Geoinformatics*, Vol. 3 No. 5, pp. 36–39.
- Güdücü, H.V. (2008), *Building Detection from Satellite Images Using shadow and Color Information*, MSc. dissertation, Electrical and Electronic Engineering- Middle East Technical University, Ankara, Turkey.
- Gupta, P., Kimothi, M. and Saklani, P. (2013), “Block Adjustment Technique, Processing & Geometric Quality Assessment of Cartosat-1 for Uttarakhand Himalayan Terrain”,

- International journal of Advancment in Remote Sensing, GIS and Geography*, Vol. 1 No. 1, pp. 49–60.
- Gutin, G., Yeo, A. and Zverovich, A. (2002), “Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP”, *Discrete Applied Mathematics*, Vol. 117 No. 1-3, pp. 81–86.
- Habib, A., Kim, E.M., Morgan, M. and Couloigner, I. (2004), “DEM generation from high resolution satellite imagery using parallel projection model”, *International Archives of Photogrammetry*, Vol. 35, pp. 393–398.
- Haithcoat, T.L., Song, W. and Hipple, J.D. (2001), “Building footprint extraction and 3-D reconstruction from LIDAR data”, *IEEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas (Cat. No.01EX482)*, Ieee, pp. 74–78.
- Hansen, P.M. and Schjoerring, J.K. (2003), “Reflectance measurement of canopy biomass and nitrogen status in wheat crops using normalized difference vegetation indices and partial least squares regression”, *Remote Sensing of Environment*, Vol. 86 No. 4, pp. 542–553.
- Harris, C. and Stephens, M. (1988), “A Combined Corner and Edge Detector”, *4th Alvey Vision Conference*, pp. 147–152.
- Haskell, L. and O'Donnell, R. (2001), “Stand Up Straight—Ortho Perspective on Downtown Denver”, available at: <http://www.esri.com/news/arcuser/1001/standup.html> (accessed 2 February 2015).
- Heijmans, H.J.A.M. (1992), “Mathematical morphology: a geometrical approach in image processing”, *Nieuw Archief voor Wiskunde*, Vol. 10, pp. 237–279.
- Helava, U.V. (1988), “Object-Space Least-Square Correlation”, *Photogrammetric Engineering and Remote Sensing*, Vol. 54 No. 6, pp. 711–714.
- Held, L. and Bove, D. (2014), *Applied Statistical Inference: Likelihood and Bayes*, Springer Heidelberg New York Dordrecht London, available at: <http://library.wur.nl/WebQuery/clc/2044986> (accessed 12 May 2014).
- Hoff, P.D. (2009), *A first course in Bayesian statistical methods*, Springer: New York ; London.
- Höhle, J. and Höhle, M. (2009), “Accuracy assessment of digital elevation models by means of robust statistical methods”, *ISPRS Journal of Photogrammetry and Remote Sensing*, Elsevier B.V., Vol. 64 No. 4, pp. 398–406.
- Hoja, D. and D'Angelo, P. (2009), “Analysis of DEM combination methods using high resolution optical stereo imagery and interferometric SAR data”, *ISPRS Hannover Workshop, High-resolution earth imaging for geospatial information*, International Society for Photogrammetry and Remote Sensing workshop, 2–5 June, Hannover, Germany, Vol. XXXVII–1–4.
- Honeyman-Buck, J. (2012), *Biomedical Image Processing*, (Deserno, T.M.,Ed.) *Journal of Digital Imaging*, Biological and Medical Physics, Biomedical Engineering, Springer

- Berlin Heidelberg, Berlin, Heidelberg, Vol. 25, doi:10.1007/s10278-012-9541-z.
- Hosford, S., Baghdadi, N., Bourguine, B., Daniels, P. and King, C. (2003), “Fusion of airborne laser altimeter and RADARSAT data for DEM generation”, *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No.03CH37477)*, IEEE, Vol. 2, pp. 806–808.
- Hu, J., Razdan, A., Femiani, J., Wonka, P. and Cui, M. (2007), “Fourier Shape Descriptors of Pixel Footprints for Road Extraction from Satellite Images”, *2007 IEEE International Conference on Image Processing*, IEEE, pp. I – 49–I – 52.
- Hu, P., Liu, X. and Hu, H. (2009), “Accuracy Assessment of Digital Elevation Models based on Approximation Theory”, *Photogrammetric Engineering & Remote Sensing*, Vol. 75 No. 1, pp. 49–56.
- Jakubowski, M.K., Guo, Q. and Kelly, M. (2013), “Tradeoffs between lidar pulse density and forest measurement accuracy”, *Remote Sensing of Environment*, Elsevier Inc., Vol. 130, pp. 245–253.
- Jalobeanu, A., Gutiérrez, J. a. and Slezak, E. (2008), “Multi-source data fusion and super-resolution from astronomical images”, *Statistical Methodology*, Vol. 5 No. 4, pp. 361–372.
- Jalobeanu, A., Gutiérrez, J.A. and Slezak, E. (2006), “Multi-source data fusion and super-resolution from astronomical images”, *Statistical Methodology*, Vol. 5 No. 4.
- James, S. (2003), *Subjective and Objective Bayesian Statistics*, (Press, S.J.,Ed.)Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2nd ed.
- Jancso, T. (2008), “Automated quality control of digital terrain models”, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVII par, pp. 863–868.
- Jaynes, E. (1968), “Prior Probabilities”, *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4 No. 3, pp. 227–241.
- 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 Advances in Signal Processing*, Vol. 2005 No. 14, pp. 2196–2206.
- Jizhou, W., Zongjian, L. and Chengming, L. (2004), “Reconstruction of buildings from a single UAV image”, *Int. Soc. Photogramm. Remote Sens. Congr., Istanbul, Turkey*, pp. 100–103.
- Johnson, D.H. (1999), “The Insignificance of Statistical Significance Testing.”, *Jo. of Wildlife management*, Vol. 63 No. 3, pp. 272–276.
- Jones, G.D., Allsop, R.E. and Gilby, J.H. (2003), “Bayesian analysis for fusion of data from disparate imaging systems for surveillance”, *Image and Vision Computing*, Vol. 21 No. 10, pp. 843–849.

- Kabolizade, M., Ebadi, H. and Mohammadzadeh, A. (2012), "Design and implementation of an algorithm for automatic 3D reconstruction of building models using genetic algorithm", *International Journal of Applied Earth Observation and Geoinformation*, Elsevier B.V., Vol. 19, pp. 104–114.
- Karantza, K., Papasakka, H. and Paragios, N. (2007), "Aerial and satellite image segmentation based on shape- constrained geodesic active contours", *ASPRS Annual Conference*, Bethesda, Maryland, pp. 339–346.
- Karel, W. and Kraus, K. (2006), "Quality parameters of digital terrain models", *The EuroSDR Test "Checking and Improving of Digital Terrian Models"*, EuroSDR Aalborg, Denmark.
- Karel, W., Pfeifer, N. and Briese, C. (2006), "DTM Quality Assessment", *ISPRS Technical Commission II Symposium, Vienna*, Vol. XXXVI-Part No. July, pp. 12–14.
- Karkee, M., Kusanagi, M. and Steward, B. (2006), "Fusion of Optical and InSAR DEMs: Improving the quality of free data", pp. 1–6.
- Karkee, M., Steward, B.L. and Aziz, S.A. (2008), "Improving quality of public domain digital elevation models through data fusion", *Biosystems Engineering*, Vol. 101 No. 3, pp. 293–305.
- Kass, M. and Witkin, A. (1988), "Snakes: Active contour models", *International journal of computer*, Vol. 1 No. 4, pp. 321–331.
- Kim, T., Lee, T., Lim, Y.J. and Kim, K. (2005), "The use of voting strategy for building extraction from high resolution satellite images", *IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS '05.*, Ieee, Vol. 2, pp. 1269–1272.
- Kim, T., Park, S.R., Kim, M.G., Jeong, S. and Kim, K.O. (2004), "Tracking road centerlines from high resolution remote sensing images by least squares correlation matching", *Photogrammetric engineering and remote sensing*, Vol. 70 No. 12, pp. 1417–1422.
- Kim, Z. and Nevatia, R. (2003), "Expandable bayesian networks for 3D object description from multiple views and multiple mode inputs", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25 No. 6, pp. 769–774.
- Kim, Z. and Nevatia, R. (2004), "Automatic description of complex buildings from multiple images", *Computer Vision and Image Understanding*, Vol. 96 No. 1, pp. 60–95.
- Kittler, J. and Illingworth, J. (1986), "Minimum Error Thresholding", *Pattern Recognition*, Vol. 19 No. 1, pp. 41–47.
- Knöpfle, W., Strunz, G. and Roth, A. (1998), "Mosaicking of digital elevation models derived by SAR interferometry", *ISPRS Commission IV Symposium, Stuttgart*, Vol. 32, pp. 306–313.
- Kobayashi, H., Mark, B.L. and Turin, W. (2012), *Probability, random processes, and statistical analysis*, Cambridge University Press.
- Kocaman, S., Zhang, L., Gruen, A. and Poli, D. (2006), "3D city modeling from

- high-resolution satellite images”, *Proceedings of ISPRS Workshop on Topographic Mapping from Space*.
- Kokkas, N. (2008), *An investigation into semi-automated 3D city modelling*, Ph.D. dissertation, University of Nottingham (UK).
- Kotwal, K. and Chaudhuri, S. (2013), “A Bayesian approach to visualization-oriented hyperspectral image fusion”, *Information Fusion*, Elsevier B.V., Vol. 14 No. 4, pp. 349–360.
- Kraus, T., Schneider, M. and Reinartz, P. (2009), “Orthorectification and DSM generation with ALOS-Prism data in urban areas”, *2009 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, pp. V–33–V–36.
- Krauss, T., Reinartz, P. and Stilla, U. (2007), “Extracting orthogonal building objects in urban areas from high resolution stereo satellite image pairs”, *ISPRS Photogrammetric Image Analysis*, Vol. 36 No. 3/W49B, pp. 77–82.
- Kruschke, J.K. (2011), *Doing Bayesian Data Analysis: A Tutorial with R and BUGS*, Academic Press.
- Kruse, F. a. and Perry, S.L. (2013), “Mineral mapping using simulated worldview-3 short-wave-infrared imagery”, *Remote Sensing*, Vol. 5 No. 6, pp. 2688–2703.
- Kumar, M. and Dass, S. (2009), “A total variation-based algorithm for pixel-level image fusion.”, *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, Vol. 18 No. 9, pp. 2137–43.
- Kumar, U., Mukhopadhyay, C. and Ramachandra, T. V. (2009), “Pixel based fusion using IKONOS imagery”, *international Journal of Recent Trends in Engineering*, Vol. 1 No. 1.
- Küng, O., Strecha, C., Fua, P., Gurdan, D., Achtelek, M., Doth, K.-M. and Stumpf, J. (2012), “Simplified Building Models Extraction from Ultra-Light UAV Imagery”, *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVIII-1/ No. September, pp. 217–222.
- Lafarge, F., Descombes, X., Zerubia, J. and Pierrot-Deseilligny, M. (2008), “Automatic building extraction from DEMs using an object approach and application to the 3D-city modeling”, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 63 No. 3, pp. 365–381.
- Lari, Z. and Ebadi, H. (2007), “Automated Building Extraction from High-Resolution Satellite Imagery Using Spectral and Structural Information Based on Artificial Neural Networks”, *International Conference on Object Extraction*, Vol. 21(2), pp. 23–27.
- Lee, S.H., Baek, S. and Shum, C.K. (2005), “Multi-temporal, multi-resolution data fusion for Antarctica DEM determination using InSAR and altimetry”, *Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International*, IEEE, Vol. 4, pp. 2827–2829.
- Legates, D.R. and McCabe, G.J. (1999), “Evaluating the use of ‘goodness-of-fit’ Measures

- in hydrologic and hydroclimatic model validation”, *Water Resources Research*, Vol. 35 No. 1, pp. 233–241.
- Leica Geosystems. (2013), *ERDAS Field Guide* TM, Leica Geosystems Geospatial Imaging, LLC.
- Leon-Garcia, A. (1994), *Probability and random processes for electrical engineering*, Addison-Wesley Longman, 2nded.
- Leon-Garcia, A. (2008), *Probabaility, Statistics, and Random Processes for Electrical Engineering*, Pearson Education, Inc., 3rded.
- Li, H. and Liu, Q. (2008), “Comparison of NDBI and NDVI as indicators of surface urban heat island effect in MODIS imagery Hua”, (Li, D., Gong, J. and Wu, H.,Eds.), Vol. 7285, pp. 728503–728503–10.
- Li, H., Zhong, C., Hu, X., Xiao, L. and Huang, X. (2013a), “New methodologies for precise building boundary extraction from LiDAR data and high resolution image”, *Sensor Review*, Vol. 33 No. 2, pp. 157–165.
- Li, P., Shi, C., Li, Z., Muller, J.-P., Drummond, J., Li, X., Li, T., et al. (2013b), “Evaluation of ASTER GDEM using GPS benchmarks and SRTM in China”, *International Journal of Remote Sensing*, Vol. 34 No. 5, pp. 1744–1771.
- Li, Y., Wu, H., An, R., Xu, H., He, Q. and Xu, J. (2013c), “An improved building boundary extraction algorithm based on fusion of optical imagery and LIDAR data”, *Optik - International Journal for Light and Electron Optics*, Vol. 124 No. 22, pp. 5357–5362.
- Li, Z. (1988), “On the measure of digital terrain model accuracy”, *The Photogrammetric Record*, Vol. 12 No. 72, pp. 873–877.
- Li, Z. (1990), *Sampling strategy and accuracy assessment for digital terrain modelling*, (PhD Thesis),University of Glasgow.
- Li, Z. (1991), “Effects of Check Points on the reliability of DTM accuracy estimates obtained from experimental tests”, *Photogrammetric Engineering and Remote Sensing*, Vol. 51 No. 10, pp. Pp1333–1340.
- Li, Z. (1992), “Variation of the accuracy of digital terrain models with sampling interval”, *The Photogrammetric Record*, Vol. 14 No. April, pp. 113–128.
- Li, Z. (2007), *Coregistration of Image pairs, DEM Alignment and Evaluation for SAR Interferometry. PhD thesis*, PURDUE UNIVERSITY.
- Li, Z., Zhu, C. and Gold, C. (2005), *Digital terrain modeling: principles and methodology*, Boca Raton, FL ; London : CRC Press, CRC Press, Boca Raton, London, New York, Washington, D.C.
- Lillesand, T., Kiefer, R.W. and Chipman, J. (2015), *Remote Sensing and Image Interpretation*, John Wiley & Sons, Inc., Seventh ed.
- Link, W.A. and Barker, R.J. (2010), *Bayesian Inference: with ecological applications*, Imprint: Academic Press.

- Liu, H., Li, J. and Chapman, M. (2003), “Automated road extraction from satellite imagery using hybrid genetic algorithms and cluster analysis”, *Journal of Environmental Informatics*, Vol. 1 No. 2, pp. 40–47.
- Liu, J., Fang, T. and Li, D. (2011), “Shadow Detection in Remotely Sensed Images Based on Self-Adaptive Feature Selection”, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 49 No. 12, pp. 5092–5103.
- Liu, Z., Cui, S. and Yan, Q. (2008), “Building extraction from high resolution satellite imagery based on multi-scale image segmentation and model matching”, *IEEE International Workshop on Earth Observation and Remote Sensing Applications*, pp. 1–7.
- Long, H. and Zhao, Z. (2005), “Urban road extraction from high-resolution optical satellite images”, *International Journal of Remote Sensing*, Vol. 26 No. 22, pp. 4907–4921.
- Ma, R. (2004), *Building Model Reconstruction From LiDAR Data and Aerial Photographs*, Ph.D. dissertation, The Ohio State University.
- MacKay, D. (2003), *Information theory, inference, and learning algorithms*, Cambridge University Press, available at:
<http://www.inference.phy.cam.ac.uk/mackay/itila/book.html> (accessed 10 October 2014).
- Mao, J., Liu, X. and Zeng, Q. (2009), “Building extraction by fusion of LIDAR data and aerial images”, *2009 Joint Urban Remote Sensing Event*, pp. 1–5.
- Mascarenhas, N.D.A., Banon, G.J.F. and Candeias, A.L.B. (1992), “Image Data Fusion under A Bayesian Approach”, *[Proceedings] IGARSS '92 International Geoscience and Remote Sensing Symposium*, IEEE, pp. 675–677.
- MASCARENHAS, N.D.A., BANON, G.J.F. and CANDEIAS, A.L.B. (1996), “Multispectral image data fusion under a Bayesian approach”, *International Journal of Remote Sensing*, Vol. 17 No. 8, pp. 1457–1471.
- Maune, D.F. (2007), *Digital elevation model technologies and applications : the DEM users manual*, American Society for Photogrammetry and Remote Sensing, 2nded.
- Mayunga, S., Zhang, Y. and Coleman, D. (2005), “Semi-automatic building extraction utilizing Quickbird imagery”, *Proc. ISPRS Workshop CMRT*, Vol. XXXVI,(3/W, pp. 131–136.
- 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*, Vol. 28 No. 10, pp. 2343–2357.
- Medioni, G. and Nevatia, R. (1985), “Segment based Stereo Matching.”, *Computer Vision, Graphics, and Image Processing*, Vol. 31 No. 1, pp. 2–18.
- Meng, L., Forberg, A. and Sarjakoski, T. (2007), “3D Building Generalisation”, *Challenges in the Portrayal of Geographic Information: Issues of Generalisation and Multi Scale Representation*, Elsevier Science Ltd., pp. Elsevier Science Ltd., Oxford, pp. 211–232.

- Mikhail, E., Bethel, J. and McGlone, J. (2001), *Introduction to modern photogrammetry*, Wiley.
- Milledge, D. (2009), “Optimization of stereo-matching algorithms using existing DEM data”, *Photogrammetric engineering and remote sensing*, Vol. 75 No. 3, pp. 323–333.
- Mittelhammer, R.C. (2013), *Mathematical Statistics for Economics and Business*, Springer New York, New York, NY, doi:10.1007/978-1-4614-5022-1.
- Mohammad-Djafari, A. (2003), “A Bayesian Approach for Data and Image Fusion”, *AIP Conference Proceedings*, AIP, Vol. 659, pp. 386–408.
- Mohammadpour, A., Féron, O. and Mohammad-djafari, A. (2004), “Bayesian segmentation of hyperspectral images”, *Bayesian Inference and Maximum Entropy Method in Science and Engineering : 24th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering 541-548 AIP*, AIP.
- Morgan, G.L.K., Liu, J.G. and Yan, H. (2008), “Sub-Pixel Stereo-Matching for DEM Generation from Narrow Baseline Stereo Imagery”, *IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, Vol. 3, pp. III – 1284–III – 1287.
- Murthy, K.P.N., Janani, M. and Priya, B.S. (2005), “Bayesian Restoration of Digital Images Employing Markov Chain Monte Carlo a Review”, *CoRR*, Computer Vision and Pattern Recognition; Statistical Mechanics; Computational Physics, , p. 42.
- Myung, I.J. (2003), “Tutorial on maximum likelihood estimation”, *Journal of Mathematical Psychology*, Vol. 47 No. 1, pp. 90–100.
- Nixon, M. and Aguado, A. (2008), *Feature extraction and image processing*, Academic Press, 2nded.
- O’Hagan, A. (2004), “Bayesian statistics: principles and benefits”, *Bayesian statistics and quality modelling in agro-food production chain (Wageningen UR Frontis Series)*, Springer., Dordrecht, Boekel MAJ., pp. 31–45.
- Ohta, Y. and Kanade, T. (1985), “Stereo by intra and inter-scanline search using dynamic programming”, *IEEE transactions on pattern Analysis and Machine Intelligence*, Vol. PAM1-7 No. 2, pp. 139 –153.
- Okutomi, M. and Kanade, T. (1992), “A locally adaptive window for signal matching”, *Int. Jo. of Computer Vision*, Vol. 7 No. 2, pp. 143–162.
- Olsen, R.C. (2007), *Remote Sensing from Air and Space*, SPIE, 1000 20th Street, Bellingham, WA 98227-0010 USA, doi:10.1117/3.673407.
- Olson, C.F. (1997), “Efficient Pose Clustering Using a Randomized Algorithm *”, *International Journal of Computer Vision*, Vol. 23 No. 2, pp. 131–147.
- Ordnance Survey. (2015), “Basic scale revision policy”, available at: <http://www.ordnancesurvey.co.uk/about/governance/policies/basic-scale-revision.html> (accessed 12 October 2015).

- Ordnance Survey (UK). (2013), "A guide to coordinate systems in Great Britain", <http://www.ordnancesurvey.co.uk>, available at: <http://www.ordnancesurvey.co.uk> (accessed 29 September 2014).
- Papasaika, H. and Kokiopoulou, E. (2011), "Fusion of digital elevation models using sparse representations", *ISPRS conference on Photogrammetric image analysis*, available at: http://link.springer.com/chapter/10.1007/978-3-642-24393-6_15 (accessed 9 September 2014).
- Papasaika, H., Poli, D. and Baltsavias, E. (2008), "A framework for the fusion of digital elevation models", *The International Archives of the Photogrammetry*, Vol. XXXVI No. Part B2, pp. 811–818.
- Papasaika, H., Poli, D. and Baltsavias, E. (2009), "Fusion of Digital Elevation Models from Various Data Sources", *2009 International Conference on Advanced Geographic Information Systems & Web Services*, IEEE, pp. 117–122.
- Papasaika-Hanusch, C. (2012), *Fusion of Digital Elevation Models*, PhD thesis, ETH Zurich.
- Partovi, T., Kraus, T., Arefi, H., Omidalizarandi, M. and Reinartz, P. (2014), "Model-driven 3D building reconstruction based on integration of DSM and spectral information of satellite images", *2014 IEEE Geoscience and Remote Sensing Symposium*, IEEE, pp. 3168–3171.
- Pawar, S. and Banga, V.K. (2012), "Morphology Approach in Image Processing", *International Conference on Intelligent Computational Systems (ICICS'2012) (Dubai)*, Dubai, pp. 148–150.
- Peckham, R.-J. and Jordan, G. (2007), *Digital Terrain Modelling*, Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-540-36731-4.
- Petremand, M., Jalobeanu, A. and Collet, C. (2012), "Optimal Bayesian fusion of large hyperspectral astronomical observations", *Statistical Methodology*, Elsevier B.V., Vol. 9 No. 1-2, pp. 44–54.
- Petrie, G. and Kennie, T. (1990), *Terrain modelling in surveying and civil engineering, Caithness : Whittles*, Caithness : Whittles, available at: <http://www.sciencedirect.com/science/article/pii/0010448587900662> (accessed 25 June 2014).
- Podobnikar, T. (2007), "DEM from various data sources and geomorphic details enhancement", *Proceedings of 5th Mountain Cartography Workshop, Bohinj, Slovenia. Zürich: International Cartographic Association, Commission on Mountain Cartography*, pp. 189–199.
- Podobnikar, T. (2009), "Methods for visual quality assessment of a digital terrain model", *Surveys and Perspectives. Integrating Environment and Society(SAPI EN. S.)*, Vol. 2 No. 2, available at: <http://sapiens.revues.org/738> (accessed 28 October 2014).

- Polat, N. and Uysal, M. (2015), “Investigating performance of Airborne LiDAR data filtering algorithms for DTM generation”, *Measurement*, Vol. 63, pp. 61–68.
- Poli, D., Wolff, K. and Gruen, A. (2007), “3D GEODATA RECOVERY FROM HIGH RESOLUTION SATELLITE IMAGERY”, *Technology*, pp. 2–5.
- Punska, O. (1999), *Bayesian Approaches to Multi-Sensor Data Fusion*, (MSc Thesis) University of Cambridge.
- Ramer, U. (1972), “An iterative procedure for the polygonal approximation of plane curves”, *Computer Graphics and Image Processing*, Vol. 1 No. 3, pp. 244–256.
- Ranchln, T. and Wald, L. (2000), “Fusion of High Spatial and Spectral Resolution Images : The ARSIS Concept and Its Implementation”, No. 1, pp. 49–61.
- Reinartz, P., Müller, R., Hoja, D., Lehner, M. and Schroeder, M. (2005), “Comparison and fusion of DEM derived from SPOT-5 HRS and SRTM data and estimation of forest heights”, *European Association of Remote Sensing Laboratories [Ed.]: Proceedings of the EARSeL Workshop on 3D Remote Sensing, Porto, Portugal*, available at: <http://m.www.eproceedings.org/SIG/3D/Dati/Porto/Reinartz.pdf> (accessed 9 September 2014).
- Reuter, H., Strobl, P. and Mehl, W. (2011), “How to merge a DEM?”, *Geomorphometry*, pp. 87–90.
- Richards, J.A. (2013), *Remote Sensing Digital Image Analysis*, Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-642-30062-2.
- Ronse, C., Najman, L. and Decencière, E. (2005), *Mathematical Morphology: 40 Years On*, (Ronse, C., Najman, L. and Decencière, E., Eds.) Computational Imaging and Vision, Springer-Verlag, Berlin/Heidelberg, Vol. 30, doi:10.1007/1-4020-3443-1.
- Rosenholm, D. (1987), “Least Squares matching Method: Some Experimental Results. Photogrammetric Record”, *Photogrammetric Record*, Vol. 12, pp. 493–512.
- Rusling, J.F. and Kumosinski, T.F. (1996), *Nonlinear computer modeling of chemical and biochemical data*, Academic Press.
- Russ, J.C. (1999), *The image processing handbook*, CRC Press, Boca Raton, FL.
- Sadeq, H., Drummond, J. and Li, Z. (2012), “Evaluation of ASTER GDEM v. 2 using GPS checkpoints, OSGB DEM values and photogrammetrically derived DEMs”, *XXII Conference of International Society of Photogrammetry and Remote Sensing*, Melbourne, Australia, Vol. XXXIX, pp. 295–300.
- Saeedi, P. and Zwick, H. (2008), “Automatic building detection in aerial and satellite images”, *IEEE 10th International Conference on Control, Automation, Robotics and Vision*, pp. 623–629.
- Sampath, A. and Shan, J. (2007), “Building Boundary Tracing and Regularization from Airborne Lidar Point Clouds”, *Photogrammetric Engineering & Remote Sensing*, Vol. 73 No. 7, pp. 805–812.

- San, D. and Turker, M. (2010), "Building extraction from high resolution satellite images using Hough transform", *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, Vol. XXXVIII No. 2001, pp. 1063–1068.
- Schalkoff, R. (1989), *Digital image processing and computer vision*, Wiley.
- Schindler, K. and Papasaika-Hanusch, H. (2011), "Improving Wide-Area DEMs Through Data Fusion-Chances and Limits", *Proceedings of the Photogrammetric Week'11*, pp. 159–170.
- Schneider, C. a, Rasband, W.S. and Eliceiri, K.W. (2012), "NIH Image to ImageJ: 25 years of image analysis", *Nature Methods*, Nature Publishing Group, Vol. 9 No. 7, pp. 671–675.
- Schultz, H., Riseman, E.M. and Stolle, F.R. (1999), "Error detection and DEM fusion using self-consistency", *Proceedings of the Seventh IEEE International Conference on Computer Vision*, IEEE, pp. 1174–1181 vol.2.
- Shaker, I.F., Abd-Elrahman, A., Abdel-Gawad, A.K. and Sherief, M. a. (2011), "Building Extraction from High Resolution Space Images in High Density Residential Areas in the Great Cairo Region", *Remote Sensing*, Vol. 3 No. 12, pp. 781–791.
- Shannon, C.E. (1948), "A Mathematical Theory of Communication", *Bell System Technical Journal*, Vol. 27 No. 3, pp. 379–423.
- Shapiro, S.S. and Wilk, M.B. (1965), "An analysis of variance test for normality (complete samples)", *Biometrika*, Vol. 52 No. 3-4, pp. 591–611.
- Sharma, R.K., Leen, T.K. and Pavel, M. (1999), "Probabilistic Image Sensor Fusion", Vol. 1.
- Sharma, R.K., Leen, T.K. and Pavel, M. (2001), "Bayesian sensor image fusion using local linear generative models", *Optical Engineering - OPT ENG*, Vol. 40 No. 7, pp. 1364–76.
- Shen, W., Zhang, J. and Yuan, F. (2011), "A new algorithm of building boundary extraction based on LIDAR data", *2011 19th International Conference on Geoinformatics*, IEEE, pp. 1–4.
- Shi, X. and Manduchi, R. (2003), "A Study on Bayes Feature Fusion for Image Classification", *2003 Conference on Computer Vision and Pattern Recognition Workshop*, IEEE, pp. 95–95.
- Shiozaki, A. (1986), "Edge extraction using entropy operator", *Computer Vision, Graphics, and Image Processing*, Vol. 36 No. 1, pp. 1–9.
- Shoemaker, J.S., Painter, I.S. and Weir, B.S. (1999), "Bayesian statistics in genetics a guide for the uninitiated", *Trends in Genetics*, Vol. 15 No. 9.
- SI Imaging Services. (2013), *Kompsat-3 Image Data Manual*, Korea Aerospace Research Institute, Vol. V1.01.
- Sivia, D. (1996), *Data analysis: a Bayesian tutorial*, Oxford : Clarendon Press ; New York : Oxford University Press.

- Smith, H.P. and Wagner, J.C. (2005), “A case study in manual and automated Monte Carlo variance reduction with a deep penetration reactor shielding problem”, *Nuclear science and engineering*, Vol. 149 No. 1, pp. 23–37.
- Smith, P.H. and Goldberg, M.J. (1987), *Problems in merging Earth sensing satellite data sets*.
- 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*, Vol. 62 No. 1, pp. 43–63.
- Sohn, G. and Dowman, I.J. (2001), “Extraction of buildings from high resolution satellite data”, *Automated Extraction of Man-Made Objects from Aerial and Space Images (III)*, Balkema Publishers, pp. 345–355.
- Sommer, K.-D., Kühn, O., León, F.P. and Siebert, B.R.L. (2009), “A Bayesian approach to information fusion for evaluating the measurement uncertainty”, *Robotics and Autonomous Systems*, Elsevier B.V., Vol. 57 No. 3, pp. 339–344.
- Sonka, M., Hlavac, V. and Boyle, R. (1993), *Image Processing, Analysis, and Machine Vision*, ITP, 2nded.
- Stauffer, H. (2008), *Contemporary Bayesian and frequentist statistical research methods for natural resource scientists*, John Wiley and Sons, Hoboken, New Jersey.
- Stefano, L. Di and Bulgarelli, A. (1999), “A simple and efficient connected components labeling algorithm”, *Proceedings 10th International Conference on Image Analysis and Processing*, IEEE Comput. Soc, pp. 322–327.
- Suárez, J.C., Smith, S., Bull, G., Malthus, T.J. and Knox, D. (2005), “The use of remote sensing techniques in operational forestry”, *Quarterly Journal of Forestry*, Vol. 99 No. 1, pp. 31–42.
- Tack, F., Buyuksalih, G. and Goossens, R. (2012), “3D building reconstruction based on given ground plan information and surface models extracted from spaceborne imagery”, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 67, pp. 52–64.
- Tack, F., Goossens, R. and Büyüksalih, G. (2009), “Semi-automatic city model extraction from tri-stereoscopic VHR satellite imagery”, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 38, pp. 89–96.
- Telfer, D.J., Spencer, J.W. and Jones, G.R. (2003), “An entropy-switched adaptive smoothing approach for time series data”, *Sensor Review*, Vol. 23 No. 1, pp. 40–43.
- Teo, T.A., Chen, L.C., Liu, C.L., Tung, Y.C. and Wu, W.Y. (2010), “DEM-aided block adjustment for satellite images with weak convergence geometry”, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 48 No. 4 PART 2, pp. 1907–1918.
- Theng, L.B. (2006), “Automatic Building Extraction from Satellite Imagery”, *International Journal of Computer Science*, Vol. 13 No. 3, pp. 255–259.

- Tokunaga, M., Hara, S., Miyazaki, Y. and Kaku, M. (1996), "Overview of DEM product generated by using ASTER data", in Kraus, K. and Waldhäusl, P. (Eds.), *International Archives of Photogrammetry and Remote Sensing*, Vienna, Austria, Vol. XXXI (B4), pp. 874–878.
- Torlegård, K., Östman, A. and Lindgren, R. (1986), "A comparative test of photogrammetrically sampled digital elevation models", *Photogrammetria*, Vol. 41 No. 1, pp. 1–16.
- Tournaire, O., Brédif, 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*, Elsevier B.V., Vol. 65 No. 4, pp. 317–327.
- Toutin, T. (2000), "Evaluation of radargrammetric DEM from RADARSAT images in high relief areas", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 38 No. 2, pp. 782–789.
- Toutin, T. and Gray, L. (2000), "State-of-the-art of elevation extraction from satellite SAR data", *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 55 No. 1, pp. 13–33.
- Tsai, V.J.D. (2006), "A comparative study on shadow compensation of color aerial images in invariant color models", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 44 No. 6, pp. 1661–1671.
- Tsai, W.-H. (1985), "Moment-preserving thresholding: A new approach", *Computer Vision, Graphics, and Image Processing*, Vol. 29 No. 3, pp. 377–393.
- Tupin, F. (2004), "Merging of SAR and optical features for 3D reconstruction in a radargrammetric framework", *IEEE International IEEE International IEEE International Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004*, IEEE, Vol. 1, pp. 89–92.
- Turner, D., Lucieer, A. and Watson, C. (2012), "An Automated Technique for Generating Georectified Mosaics from Ultra-High Resolution Unmanned Aerial Vehicle (UAV) Imagery, Based on Structure from Motion (SfM) Point Clouds", *Remote Sensing*, Vol. 4 No. 12, pp. 1392–1410.
- United States Geological Survey. (2009), "USGS Astrogeology Science Center Tutorial: Stereo Processing using HiRISE Stereo Imagery ISIS3 and SOCET SET", available at: ftp://pdsimage2.wr.usgs.gov/pub/pigpen/mars/tutorials/SS4HIRISE_JUNE2009/HiRISE_StereoProcessingTutorial_June2009.pdf (accessed 9 December 2014).
- Vosselman, G., Sester, M. and Mayer, H. (2004), "Basic computer vision techniques", *Manual of Photogrammetry, Fifth Edition, American Society for Photogrammetry and Remote Sensing (edited by McGlone, J.C., Mikhail E.M., and J. Bethel)*, pp. 445–504.
- Wang, O., Lodha, S.K. and Helmbold, D.P. (2006), "A Bayesian Approach to Building Footprint Extraction from Aerial LIDAR Data", *Third International Symposium on 3D*

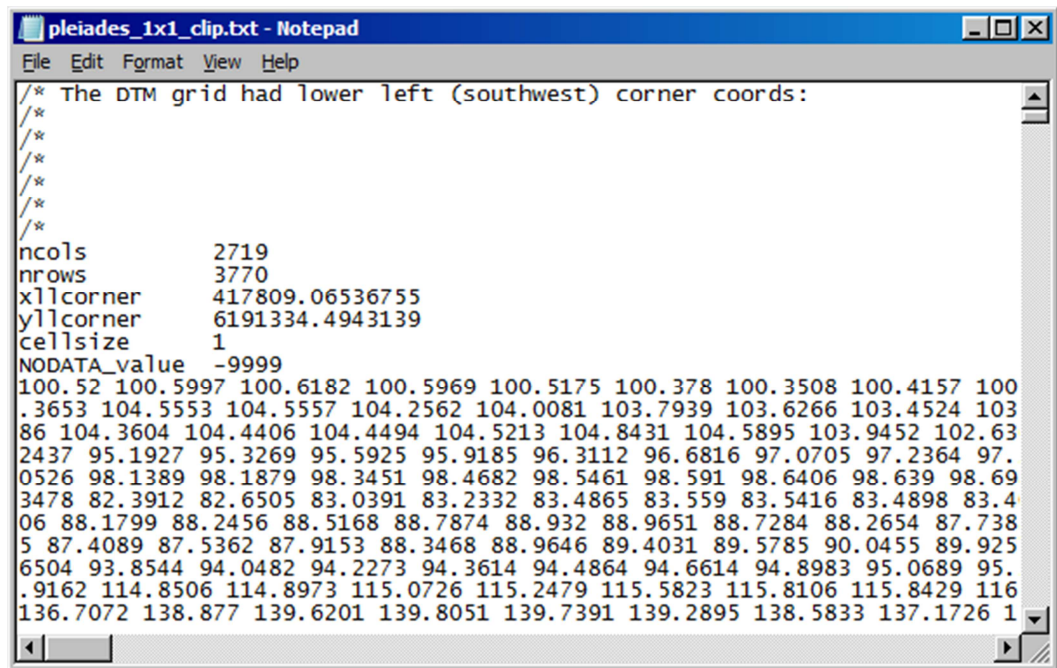
- Data Processing, Visualization, and Transmission (3DPVT'06)*, IEEE, pp. 192–199.
- Waqar, M.M., Mirza, J.F., Mumtaz, R. and Hussain, E. (2012), “Development of New Indices for Extraction of Built-Up Area & Bare Soil from Landsat Data”, *Journal of Geophysics & Remote Sensing*, Vol. 1 No. 1, pp. 1–4.
- Wegmüller, U., Wiesmann, A. and Santoro, M. (2010), “Merging of an EET CInSAR DEM with the SRTM DEM”, *Gamma Remote, Sensing Ag*, doi:10.1016/j.jappgeo.2009.04.002.
- Weisstein, E.W. (2014), “Point-Line Distance-2Dimensional”, *From MathWorld-A Wolfram Web Resource*, available at: <http://mathworld.wolfram.com/Point-LineDistance2-Dimensional.html> (accessed 4 November 2014).
- Weng, Q. (2002), “Quantifying uncertainty of digital elevation models derived from topographic maps”, *Richardson, D.E. and van Oosterom, O., editors, Advances in spatial data handling: 10th International Symposium on Spatial Data Handling, Berlin: Springer*, pp. 403–408.
- Willmott, C.J., Ackleson, S.G., Davis, R.E., Feddema, J.J., Klink, K.M., Legates, D.R., O'Donnell, J., et al. (1985), “Statistics for the evaluation and comparison of models”, *Journal of Geophysical Research*, Vol. 90 No. C5, p. 8995.
- Wise, S. (2000), “Assessing the quality for hydrological applications of digital elevation models derived from contours”, *Hydrological processes*, Vol. 14 No. 11-12, pp. 1909–1929.
- Wolf, P.R. (1983), *Elements of photogrammetry*, McGraw-Hill Education - Co.- Singapore, available at: <http://trid.trb.org/view.aspx?id=91357> (accessed 2 September 2014).
- Wood, J. (1996), *The Geomorphological Characterisation of Digital Elevation Models*, University of Leicester, Leicester, UK.
- Worthington, P.L. (2002), “Enhanced Canny edge detection using curvature consistency”, *Object recognition supported by user interaction for service robots*, IEEE Comput. Soc, Vol. 1, pp. 596–599.
- Xiong, Z. and Zhang, Y. (2009), “A Generic Method for RPC Refinement Using Ground Control Information”, *Photogrammetric engineering and remote sensing*, Vol. 75 No. 9, pp. 1083–1092.
- Xiong, Z. and Zhang, Y. (2010), “A New Strategy for DSM Generation from High Resolution Stereo Satellite Images Based on Control Network Interest Point Matching”, *ISPRS TC VII Symposium*, Vol. XXXVIII, pp. 658–663.
- XU, C., Wei, M., Griffiths, S., Mercer, B. and Abdoullaev, R. (2010), “Hybrid DEM generation and evaluation from spaceborne radargrammetric and optical stereoscopic DEMs”, *ISPRS Archives – Volume XXXVIII - Part 1*.
- Yamazaki, F., Liu, W. and Takasaki, M. (2009), “Characteristics of shadow and removal of its effects for remote sensing imagery”, *2009 IEEE International Geoscience and*

- Remote Sensing Symposium*, IEEE, pp. IV-426-IV-429.
- Yang, X. and Hodler, T. (2000), "Visual and Statistical Comparisons of Surface Modeling Techniques for Point-based Environmental Data", *Cartography and Geographic Information Science*, Vol. 27 No. 2, pp. 165-176.
- Yang, Y. and Lin, Y. (2009), "Object-based level set model for building detection in urban area", *2009 Joint Urban Remote Sensing Event*, IEEE, pp. 1-6.
- Yang-Mao, S.-F., Chan, Y.-K. and Chu, Y.-P. (2008), "Edge enhancement nucleus and cytoplasm contour detector of cervical smear images.", *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, Vol. 38 No. 2, pp. 353-366.
- Ye, F., Su, L. and Tang, J. (2006), "Automatic Road Extraction Using Particle Filters from High Resolution Images", *Journal of China University of Mining and Technology*, Vol. 16 No. 4, pp. 490-493.
- Yuan, J., Wang, D., Wu, B., Yan, L. and Li, R. (2009), "Automatic road extraction from satellite imagery using LEGION networks", *2009 International Joint Conference on Neural Networks*, IEEE, pp. 3471-3476.
- Zaniboni, G. and Mascarenhas, N. (1998), "Fusão bayesiana de imagens utilizando coeficientes de correlação localmente adaptáveis", *Simpósio Brasileiro de Sensoriamento Remoto, 9. (SBSR)*, pp. 1003-1014.
- Zhang, B., Manager, D., Miller, S. and Walker, S. (2007), "Next Generation Automatic Terrain Extraction using Microsoft UltraCam imagery", *ASPRS 2007 Annual Conference*, Tampa, Florida.
- Zhang, B. and Miller, S. (1997), "Adaptive automatic terrain extraction. Integrating photogrammetric techniques with scene analysis and machine vision", *SPIE Journal of Electronic Imaging*, No. 3072, pp. 27-36.
- Zhang, B., Miller, S., K., D. and Walker, S. (2006), "Automatic terrain extraction using multiple image pair and back matching", *ASPRS 2006 Annual Conference, Reno, Nevada*, available at:
<ftp://jetty.ecn.purdue.edu/jshan/proceedings/asprs2006/files/0180.pdf> (accessed 8 July 2014).
- Zhang, B. and Smith, W. (2010), "From Where to What: Image Understanding Through 3-D Geometric Shapes", *ASPRS 2010 Annual Conference*.
- Zhang, J. (2010a), "Multi-source remote sensing data fusion: status and trends", *International Journal of Image and Data Fusion*, Vol. 1 No. 1, pp. 5-24.
- Zhang, J. (2010b), "Multi-source remote sensing data fusion: status and trends", *International Journal of Image and Data Fusion*, Vol. 1 No. 1, pp. 5-24.
- Zhang, L. and Gruen, A. (2006), "Multi-image matching for DSM generation from IKONOS imagery", *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 60 No. 3, pp. 195-211.

- Zhang, Y. (2011), *EM-based Bayesian fusion of hyperspectral and multispectral images, Image Fusion*, InTech: Rijeka, Croatia.
- Zhang, Y., De Backer, S. and Scheunders, P. (2008), “Bayesian Fusion of Multispectral and Hyperspectral Image in Wavelet Domain”, *IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, pp. V – 69–V – 72.
- Zhou, G. (2010), *High-Resolution Urban True Orthoimagery*, Old Dominion University.
- Zisserman, A. (2007), “Lectures 3 & 4 ‘Estimators - ML, LS, MAP’”, *Estimation and Inference lectures Hilary, Robotics Research Group Department of Engineering Science, University of Oxford*, available at:
<http://www.robots.ox.ac.uk/~az/lectures/est/lect34.pdf> (accessed 12 February 2015).

Appendix A-C++ Code for merging Digital surface model using Maximum Likelihood approach

The following appendix is related to merging DSMs based on using Bayesian Approach, the number of required DSMs should be two and the format should be ascii as shown below:

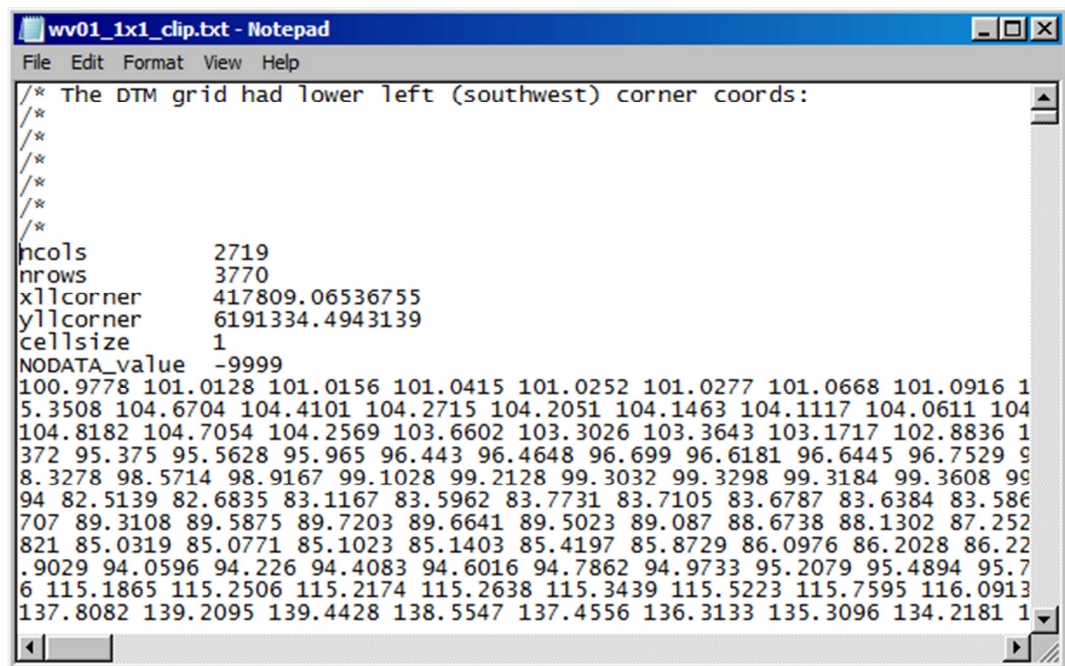


```

pleiades_1x1_clip.txt - Notepad
File Edit Format View Help
/* The DTM grid had lower left (southwest) corner coords:
/*
/*
/*
/*
/*
ncols      2719
nrows      3770
xllcorner  417809.06536755
yllcorner  6191334.4943139
cellsize   1
NODATA_value -9999
100.52 100.5997 100.6182 100.5969 100.5175 100.378 100.3508 100.4157 100
.3653 104.5553 104.5557 104.2562 104.0081 103.7939 103.6266 103.4524 103
86 104.3604 104.4406 104.4494 104.5213 104.8431 104.5895 103.9452 102.63
2437 95.1927 95.3269 95.5925 95.9185 96.3112 96.6816 97.0705 97.2364 97.
0526 98.1389 98.1879 98.3451 98.4682 98.5461 98.591 98.6406 98.639 98.69
3478 82.3912 82.6505 83.0391 83.2332 83.4865 83.559 83.5416 83.4898 83.4
06 88.1799 88.2456 88.5168 88.7874 88.932 88.9651 88.7284 88.2654 87.738
5 87.4089 87.5362 87.9153 88.3468 88.9646 89.4031 89.5785 90.0455 89.925
6504 93.8544 94.0482 94.2273 94.3614 94.4864 94.6614 94.8983 95.0689 95.
.9162 114.8506 114.8973 115.0726 115.2479 115.5823 115.8106 115.8429 116
136.7072 138.877 139.6201 139.8051 139.7391 139.2895 138.5833 137.1726 1

```

(a)



```

wv01_1x1_clip.txt - Notepad
File Edit Format View Help
/* The DTM grid had lower left (southwest) corner coords:
/*
/*
/*
/*
/*
ncols      2719
nrows      3770
xllcorner  417809.06536755
yllcorner  6191334.4943139
cellsize   1
NODATA_value -9999
100.9778 101.0128 101.0156 101.0415 101.0252 101.0277 101.0668 101.0916 1
5.3508 104.6704 104.4101 104.2715 104.2051 104.1463 104.1117 104.0611 104
104.8182 104.7054 104.2569 103.6602 103.3026 103.3643 103.1717 102.8836 1
372 95.375 95.5628 95.965 96.443 96.4648 96.699 96.6181 96.6445 96.7529 9
8.3278 98.5714 98.9167 99.1028 99.2128 99.3032 99.3298 99.3184 99.3608 99
94 82.5139 82.6835 83.1167 83.5962 83.7731 83.7105 83.6787 83.6384 83.586
707 89.3108 89.5875 89.7203 89.6641 89.5023 89.087 88.6738 88.1302 87.252
821 85.0319 85.0771 85.1023 85.1403 85.4197 85.8729 86.0976 86.2028 86.22
.9029 94.0596 94.226 94.4083 94.6016 94.7862 94.9733 95.2079 95.4894 95.7
6 115.1865 115.2506 115.2174 115.2638 115.3439 115.5223 115.7595 116.0913
137.8082 139.2095 139.4428 138.5547 137.4556 136.3133 135.3096 134.2181 1

```

(b)

Figure A.1 The used input data in the merging (a) first input DSM from Pleiades with cell size 1m(b) second input DSM for WorldView-1 with cell size 1m.

The implemented Code in the merging with Maximum Likelihood approach is:

```

//This program is written using Microsoft's Visual Studio 2010 C++
//The author ship of the code is Haval A.Sadeq as a part of the Ph.D. dissertation
//Ph.D. in Geomatics, college of sciences and Engineering, University of Glasgow,
Scotland,UK
//Email
h.sadeq.1@research.gla.ac.uk,website:http://www.gla.ac.uk/schools/ges/pgresearch/hav
alsadeq/
//This code is for merging two DSMs using Maximum Likelihood (i.e. weighted average)
// It is completed and evaluated 15/10/2013
//Define header files
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include<float.h>
using namespace std;
#define a 2719// Number of rows in the data
#define b 3770//Number of rows in the data
int main()
{float i1,i2, mle;
  ofstream myfile1;
  string
line1,line2,line3,line4,line5,line6,line7,line8,line9,line10,line11,line12,line13;//
reading the header of the file
  float r1[a][b],r2[a][b],//the array for saving the DSM
merged[a][b]; //the array for saving the merged DSM
  //////////opening and reading first file//////////
  char *iname1 = "pleiades_1x1_clip.txt";
  ifstream infile1(iname1);
  if (!infile1) {
    cout << "There was a problem opening file "
    << iname1
    << " for reading."
    << endl; return 0; }
  cout << "Opened " << iname1 << " for reading." << endl;
  getline (infile1,line1);
  getline (infile1,line2);
  getline (infile1,line3);
  getline (infile1,line4);
  getline (infile1,line5);
  getline (infile1,line6);
  getline (infile1,line7);
  getline (infile1,line8);
  getline (infile1,line9);
  getline (infile1,line10);
  getline (infile1,line11);
  getline (infile1,line12);
  getline (infile1,line13);
  cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
line11 << endl<< line12 << endl<< line13 << endl;
  ////////////
  int j=0, k=0;
  while (infile1 >> i1)
  {if (j==(b-1))
    {r1[k][j]=i1;j=0; k++;}
    else{r1[k][j]=i1;j++;}
    ;}
  ////////////////////////////////////////////
  //////////opening and reading second file//////////////////////////////////////////
  char *iname2 = "wv01_1x1_clip.txt";
  ifstream infile2(iname2);
  if (!infile2) {

```

```

        cout << "There was a problem opening file "
              << inname2
              << " for reading."
              << endl;          return 0;    }

    cout << "Opening Second File "<<endl;
    cout << "Opened " << inname2 << " for reading." << endl;
    getline (infile2,line1);
    getline (infile2,line2);
    getline (infile2,line3);
    getline (infile2,line4);
    getline (infile2,line5);
    getline (infile2,line6);
    getline (infile2,line7);
    getline (infile2,line8);
    getline (infile2,line9);
    getline (infile2,line10);
    getline (infile2,line11);
    getline (infile2,line12);
    getline (infile2,line13);
    cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
    endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
    line11 << endl<< line12 << endl<< line13 << endl;
    ///////////////////////////////////
    j=0, k=0;
    while (infile2 >> i2)
    {if (j==(b-1))
        {r2[k][j]=i2;j=0; k++;}
        else{r2[k][j]=i2;j++;}
        ;}
    cout << "start merging "<<endl;
    ///////////////////////////////////
    //define sigma value based on the RMSE calculated from GNSS checkpoints
    float sigma1=0.8, sigma2=0.35, n3=0;
    ///////////////////////////////////
    for (k=0;k<a;++k)
    {
        for (j=0;j<b;++j)
        {
            //the model used in the merging based on Maximum Likelihood (weighted average)
            merged[k][j]=sigma2*sigma2/(sigma1*sigma1+sigma2*sigma2)*r1[k][j]
                +sigma1*sigma1/(sigma1*sigma1+sigma2*sigma2)*r2[k][j];
        }
    }

    cout<<std::fixed<<std::setprecision(3)<<(float)k/a*100<<"%"<<endl;// percentage of
    progress
    }
    //saving the merged data to a file
    myfile1.open ("merged_dsm_likelihood.txt");
    myfile1 << line1<<endl;
    myfile1 << line2<<endl;myfile1 << line3<<endl;myfile1 << line4<<endl; myfile1 <<
    line5<<endl; myfile1 << line6<<endl; myfile1 << line7<<endl; myfile1 << line8<<endl;
    myfile1 << line9<<endl; myfile1 << line10<<endl;  myfile1 << line11<<endl;
    myfile1 << line12<<endl;  myfile1 << line13<<endl;
        for (k=0;k<a;++k)
            for (j=0;j<b;++j)
            {
    myfile1<< std::fixed << std::setprecision(4)<< merged[k][j]<<' ';// controlling the
    precision of the number

            }
    myfile1.close( ); infile1.close( ); infile2.close( );//close the file

    return 0;}
    ///////////////////////////////////End of the code////////////////////////////////////

```

Appendix B-C++ Code for merging Digital surface model using Bayesian approach

The following code used in the merging with Bayesian approach, using the same format of data that is shown in figure A.1:

```

//This program is written using Microsoft's Visual Studio 2010 C++
//The authorship of the code is Haval A.Sadeq as a part of the Ph.D. dissertation
//Ph.D. in Geomatics, college of sciences and Engineering, University of Glasgow,
Scotland, UK
//Email
h.sadeq.1@research.gla.ac.uk,website:http://www.gla.ac.uk/schools/ges/pgresearch/hav
alsadeq/
//This code is for merging DSMs using Bayesian approach from two different sources with
equal or different quality
// It is completed and evaluated 20/01/2014

//Define header files
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include <float.h>
#include <random>
using namespace std;

//not when finish from the processing deduct 4 from col and row in the output file
#define b 2719//Number of rows in the data
#define a 3770 //Number of rows in the data
#define var .1 //the range used in the local entropy simulation

double entropy(int k11, int j11, double r[a][b]);
int main()
{
//define the variables
double i1,i2,i3, mle;
ofstream myfile1;
string
line1,line2,line3,line4,line5,line6,line7,line8,line9,line10,line11,line12,line13;
double r1[a][b],r2[a][b],pij_temp[a][b],
merged_dem[a][b],r1_sim[a][b],r2_sim[a][b]

;

//////////opening first DSM//////////
cout<<"start reading first file"<<endl;
char *iname1 = "pleiades_1x1_clip.txt";

ifstream infile1(iname1);
if (!infile1) {
cout << "There was a problem opening file "
<< iname1
<< " for reading."
<< endl; return 0; }
cout << "Opened " << iname1 << " for reading." << endl;
getline (infile1,line1);
getline (infile1,line2);
getline (infile1,line3);
getline (infile1,line4);

```

```

getline (infile1,line5);
getline (infile1,line6);
getline (infile1,line7);
getline (infile1,line8);
getline (infile1,line9);
getline (infile1,line10);
getline (infile1,line11);
getline (infile1,line12);
getline (infile1,line13);
cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
line11 << endl<< line12 << endl<< line13 << endl;
//////////Reading first file and saving to array
int j=0, k=0;
while (infile1 >> i1)
{if (j==(b-1))
    {r1[k][j]=i1;j=0; k++;}
    else{r1[k][j]=i1;j++;}
    ;}

//////////opening second DSM//////////
cout<<"start reading second file"<<endl;
char *iname2 = "wv01_1x1_clip.txt";
ifstream infile2(iname2);
if (!infile2) {
    cout << "There was a problem opening file "
    << iname2
    << " for reading."
    << endl;    return 0;    }
cout << "Opened " << iname2 << " for reading." << endl;
getline (infile2,line1);
getline (infile2,line2);
getline (infile2,line3);
getline (infile2,line4);
getline (infile2,line5);
getline (infile2,line6);
getline (infile2,line7);
getline (infile2,line8);
getline (infile2,line9);
getline (infile2,line10);
getline (infile2,line11);
getline (infile2,line12);
getline (infile2,line13);
cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
line11 << endl<< line12 << endl<< line13 << endl;

//////////Reading second file and saving to array
j=0, k=0;
while (infile2 >> i2)
{if (j==(b-1))
    {r2[k][j]=i2;j=0; k++;}
    else{r2[k][j]=i2;j++;}
    ;}

//////////
cout<<"start mearing the DSMs using Bayesian approach"<<endl;

double entropy_edge[a][b],pij[a][b],z;
for (k=0;k<a;++k)
    for (j=0;j<b;++j)
        {entropy_edge[k][j] =
0;merged_dem[k][j]=0;r1_sim[k][j]=r1[k][j];r2_sim[k][j]=r2[k][j];}
//selecting the window size for the for a priori elevation estimation using maximum
entropy

```

```

        int wind=1;//for window size 3x3
        //int wind=2;//for window size 5x5
        //int wind=3;//for window size 7x7

        //merging the elevations

j=0, k=0;
double optim_max1,optim_max2,ent_dem1,ent_dem2,prior1,prior2;

for (k=wind;k<a-wind;++k)
{
    for (j=wind;j<b-wind;++j)
    {
        int itr1=0,itr2=0;
        double vv1=r1[k][j],vv2=r2[k][j],vvv1,vvv2;

        optim_max1=-100000000,optim_max2=-100000000;
        int i=0;

        for (float v=-var;v<var;v=v+0.01)
        {

            double err1,err2;

            //calculate new value of maximum entropy
            r1_sim[k][j]=r1[k][j]+v;
            r2_sim[k][j]=r2[k][j]+v;

            ent_dem1=entropy( k, j, r1_sim);
            ent_dem2=entropy( k, j, r2_sim);

            if (ent_dem1 >= optim_max1) {optim_max1=ent_dem1;itr1=i;vvv1=v;}
            if (ent_dem2 >= optim_max2) {optim_max2=ent_dem2;itr2=i;vvv2=v;}

            i=i+1;

        }

        prior1=vv1+vvv1;//the prior elevation for the first DSM
        prior2=vv2+vvv2;//the prior elevation for the second DSM

        double var1=0.8*0.8,//the quality of first DSM is 0.8m
            var2=0.35*0.35;//the quality of first DSM is 0.35m
        //define merging model based on Bayesian approach after determining the prior elevation
        //fro the function using the maximum entropy
        merged_dem[k][j]=(r1[k][j]/var1+r2[k][j]/var2+prior1/var1+prior2/var2)/
        (1/var1+1/var2+1/var1+1/var2);

    }

    cout<<std::fixed<<std::setprecision(2)<<(float)k/a*100<<"%"<<endl;// percentage of
    progress

}

//start writing the result to a file
cout<<"start writing the result to a file"<<endl;

myfile1.open ("merged_dsm_.1var_high_entropy_wind_3x3.txt");
myfile1 << line1<<endl;
    myfile1 << line2<<endl;myfile1 << line3<<endl;myfile1 << line4<<endl; myfile1 <<
line5<<endl; myfile1 << line6<<endl; myfile1 << line7<<endl; myfile1 << line8<<endl;
    myfile1 << line9<<endl; myfile1 << line10<<endl; myfile1 << line11<<endl;
    myfile1 << line12<<endl; myfile1 << line13<<endl;

```

```

//write the result to a file
for (k=0;k<a;++k)
    for (j=0;j<b;++j)
    {
        myfile1<< std::fixed << std::setprecision(4)<< merged_dem[k][j]<< ' ';
    }

myfile1.close( );  infile1.close( );  infile2.close( ); //close the files
return 0;}

//////////end of teh program for merging DSMs//////////

//////////functions using for the prior extimation based on maximum entropy
double entropy(int k11, int j11, double r[a][b])
{
    ////////////defining the window size//////////
    double pij_temp[3][3],r_wind[3][3],sum_wind=0,Hij;    int wind=1;//for window
size 3x3
    //double pij_temp[5][5],r_wind[5][5],sum_wind=0,Hij;    int wind=2;//for window
size 5x5
    //double pij_temp[7][7],r_wind[7][7],sum_wind=0,Hij;    int wind=3;//for window
size 7x7

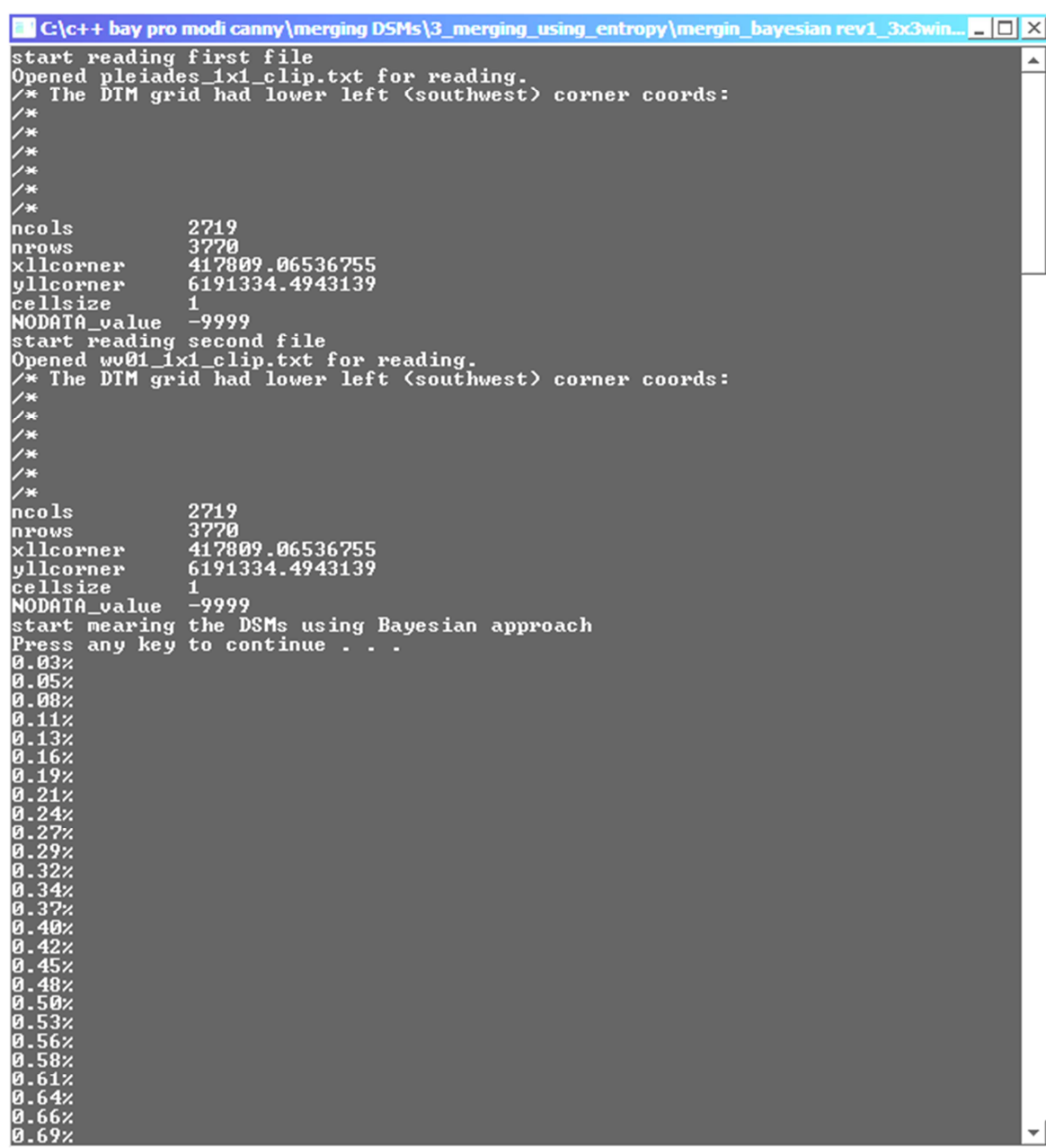
    int co_k=0,co_j=0;
    ////////////
    for(int k1=k11-wind;k1<=k11+wind;k1++)
    {co_j=0;
        for(int j1=j11-wind;j1<=j11+wind;j1++)
        {
            r_wind[co_k][co_j]=r[k1][j1];
            co_j=co_j+1;
        }
        co_k=co_k+1;
    }
for(int k1=0;k1<=1+wind;k1++)
    for(int j1=0;j1<=1+wind;j1++)
    {
        sum_wind=sum_wind+r_wind[k1][j1];
    }
    //////////// find probability for each pixel
    for(int k2=0;k2<=1+wind;k2++)
    for(int j2=0;j2<=1+wind;j2++)
    {pij_temp[k2][j2]=r_wind[k2][j2]/sum_wind;}
    //////////// find the entropy for the group
Hij=0;
    for(int k3=0;k3<=1+wind;k3++)
    for(int j3=0;j3<=1+wind;j3++)
    {
        Hij=Hij+pij_temp[k3][j3]*log10(pij_temp[k3][j3]);
    }

    return (-Hij)*1000000;}

//////////end of the function//////////

```


The processing window is as shown below indicating the percentage completed. The used parameter merging is 3x3 window and the range is $\pm 0.1\text{m}$. At the end the output will be an ASCII file:



```

C:\c++ bay pro modi canny\merging DSMs\3_merging_using_entropy\mergin_bayesian rev1_3x3win...
start reading first file
Opened pleiades_1x1_clip.txt for reading.
/* The DTM grid had lower left (southwest) corner coords:
/*
/*
/*
/*
/*
ncols          2719
nrows          3770
xllcorner      417809.06536755
yllcorner      6191334.4943139
cellsize       1
NODATA_value   -9999
start reading second file
Opened wv01_1x1_clip.txt for reading.
/* The DTM grid had lower left (southwest) corner coords:
/*
/*
/*
/*
/*
ncols          2719
nrows          3770
xllcorner      417809.06536755
yllcorner      6191334.4943139
cellsize       1
NODATA_value   -9999
start mearing the DSMs using Bayesian approach
Press any key to continue . . .
0.03%
0.05%
0.08%
0.11%
0.13%
0.16%
0.19%
0.21%
0.24%
0.27%
0.29%
0.32%
0.34%
0.37%
0.40%
0.42%
0.45%
0.48%
0.50%
0.53%
0.56%
0.58%
0.61%
0.64%
0.66%
0.69%

```

Figure B.1 A screen shot of merging process using Bayesian approach.

Appendix C-C++ Code for labelling nDSM

This code is used in order to label the nDSM that produced from subtracting the original DSM from the convolved DSM as illustrated in 7.2.2

Note: the nDSM labelling code has not been embedded in the main code which is shown in appendix D, because it was taking a long time and it is preferred to run this separately so it will not affect the time needed for building footprint extraction and 3D modelling.

```

//This program is written using Microsoft's Visual Studio 2010 C++.
//The author ship of the code is Haval A.Sadeq as a part of the Ph.D. dissertation.
//Ph.D. in Geomatics, college of sciences and Engineering, University of Glasgow,
Scotland, UK.
//Email
h.sadeq.1@research.gla.ac.uk,website:http://www.gla.ac.uk/schools/ges/pgresearch/hav
alsadeq/
//This code is for labelling the binary images using the connected component labelling.
// It is completed and evaluated 15/8/2013.

//Define the header files
#include <stdio.h>
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include<float.h>
using namespace std;
#define b 704//Number of columns in the data
#define a 381 //Number of rows in the data

int main()
{
//define the variables
float i;
ofstream myfile1;
ofstream myfile2;
int c1[1000000],c2[1000000];
string
line1,line2,line3,line4,line5,line6,line7,line8,line9,line10,line11,line12,line13;
float r[a][b],r_new[a+2][b+2];
int label[a][b];
float thr_level[a][b];

int j=0, k=0;

char *iname = "ndsmgrid.txt";// define the file name

ifstream infile(inname);
if (!infile) {
    cout << "There was a problem opening file "
    << inname
    << " for reading."
    << endl;    return 0;    }
cout << "Opened " << inname << " for reading." << endl;
getline (infile,line1);
getline (infile,line2);
getline (infile,line3);
getline (infile,line4);
getline (infile,line5);
getline (infile,line6);
getline (infile,line7);

```

```

getline (infile,line8);
getline (infile,line9);
getline (infile,line10);
getline (infile,line11);
getline (infile,line12);
getline (infile,line13);
cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
line11 << endl<< line12 << endl<< line13 << endl;
//////////
while (infile >> i)

{if (j==(b-1))
    {r[k][j]=i;j=0; k++;}
else{r[k][j]=i;j++;}
};
//////////calculate the threshold at each level and save the data into layers

    { int level=1;
for(k=0;k<a;k++){
    for(j=0;j<b;j++){
        if(r[k][j] < level) thr_level[k][j] = 0;
        else thr_level[k][j] = 1;    } } }
    //////////start labelling//////////

{    int lab=0, newlabel;
    if (thr_level[0][0] == 1) label[0][0]=1;else label[0][0]=0;//check first
corner
    for (j=1;j<b;++j)
    {
if (thr_level[0][j] == 1)// 1 is foreground , 0 is for background
{ if (thr_level[0][j] == thr_level[0][j-1] ) { label[0][j]= label[0][j-1];//check
frist row
    }
    else {++lab; label[0][j]=lab;}    }

else label[0][j]=0; }
////start the other rows
    int ch=0;
    for (k=1;k<a;++k)

    for (j=0;j<b;++j)
    {

if (thr_level[k][j] == 1)
{
    if (j == 0) if (thr_level[k-1][j]== 1 ) { label[k][j] = label[k-1][j];};//specify the
first column
    else {++lab; label[k][j]=lab;}}

//////////start first pass
//for the other columns rather than first column
//case1
    {
    if (thr_level[k][j] == thr_level[k][j-1] && thr_level[k][j] == thr_level[k-1][j]
)
    { if ( label[k-1][j] > label[k][j-1])
    { label[k][j]= label[k][j-1];ch=ch+1;;c1[ch]= label[k-1][j];c2[ch]=
label[k][j-1];    }

    else {label[k][j]= label[k-1][j];ch=ch+1;;c1[ch]= label[k][j-1];c2[ch]=
label[k-1][j];}}

//case2
if (thr_level[k][j] != thr_level[k][j-1] && thr_level[k][j] == thr_level[k-1][j] )
    label[k][j]= label[k-1][j];

```

```

//case3
if (thr_level[k][j] == thr_level[k][j-1] && thr_level[k][j] != thr_level[k-1][j] )
    label[k][j]= label[k][j-1];

//case 4
if (thr_level[k][j] != thr_level[k][j-1] && thr_level[k][j] != thr_level[k-1][j] )
    {++lab; label[k][j]=lab;}}

else label[k][j]=0; }

//////////start second pass

for (int ch1=0;ch1<=ch;++ch1)
{
    if (c1[ch1] != c2[ch1] )
    {
        for (k=0;k<a;++k)
            for (j=0;j<b;++j)
            {
                if (label[k][j]==c2[ch1] && c1[ch1] > c2[ch1]) label[k][j]=c1[ch1];
                else if (label[k][j]==c1[ch1] && c1[ch1] < c2[ch1]) label[k][j]=c2[ch1];
            }}
}

//////////scan the image for discrepancy in the labels
for (int ss=0;ss<100;++ss)
{
    for (k=0;k<a;++k)
        for (j=0;j<b;++j)
        if (thr_level[k][j] == 1)
        {
            if (thr_level[k][j] == thr_level[k-1][j-1] )
                if ( label[k][j] > label[k-1][j-1]) label[k][j]=label[k-1][j-1];
            if (thr_level[k][j] == thr_level[k-1][j] )
                if ( label[k][j] > label[k-1][j]) label[k][j]=label[k-1][j];
            if (thr_level[k][j] == thr_level[k-1][j+1] )
                if ( label[k][j] > label[k-1][j+1]) label[k][j]=label[k-1][j+1];
            if (thr_level[k][j] == thr_level[k][j-1] )
                if ( label[k][j] > label[k][j-1]) label[k][j]=label[k][j-1];
            if (thr_level[k][j] == thr_level[k][j+1] )
                if ( label[k][j] > label[k][j+1]) label[k][j]=label[k][j+1];
            if (thr_level[k][j] == thr_level[k+1][j-1] )
                if ( label[k][j] > label[k+1][j-1]) label[k][j]=label[k+1][j-1];
            if (thr_level[k][j] == thr_level[k+1][j] )
                if ( label[k][j] > label[k+1][j]) label[k][j]=label[k+1][j];
            if (thr_level[k][j] == thr_level[k+1][j+1] )
                if ( label[k][j] > label[k+1][j+1]) label[k][j]=label[k+1][j+1];
        }
    }

}

////////saving the labelled nDSM data to a file so it can be used in the building footprint
extraction
myfile1.open ("label.txt");

myfile1 << line1<<endl;
myfile1 << line2<<endl;myfile1 << line3<<endl; myfile1 << line4<<endl; myfile1
<< line5<<endl;myfile1 << line6<<endl; myfile1 << line7<<endl; myfile1 <<
line8<<endl; myfile1 << line9<<endl; myfile1 << line10<<endl; myfile1 <<
line11<<endl;myfile1 << line12<<endl; myfile1 << line13<<endl;
{
    for (k=0;k<a;++k)
        for (j=0;j<b;++j)
            myfile1<< label[k][j]<< ' ';}}

//close the file
myfile1.close( );
infile.close( );
return 0;}

//////////end of the program for labelling nDSM//////////

```

Appendix D-C++ Code for Building footprint Extraction processes

Appendix C is related to the building footprint extraction,

```

/* The DTM grid had lower left (southwest) corner coords:
/*
/*
/*
/*
ncols      580
nrows      440
xllcorner  419870
yllcorner  6194210
cellsize    0.5
NODATA_value -9999
111.9639 111.9825 111.9993 112.0157 112.0316 112.0455 112.
115.6677 115.6609 115.6635 115.6819 115.707 115.7436 115.
193 117.2026 117.2117 117.2238 117.241 117.2631 117.294 11
118.8094 118.8138 118.8149 118.8155 118.8154 118.8172 118.
0.8627 120.9021 120.9323 120.9467 120.9548 120.9521 120.9

```

-The used DSM for height elevation

```

/* The DTM grid had lower left (southwest) corner coords:
/*
/*
/*
/*
ncols      580
nrows      440
xllcorner  419870
yllcorner  6194210
cellsize    0.5
NODATA_value -9999
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

-Labelled nDSM that produced from using code in appendix C

```

/* The DTM grid had lower left (southwest) corner coords:
/*
/*
/*
/*
ncols      580
nrows      440
xllcorner  419870
yllcorner  6194210
cellsize    0.5
NODATA_value -9999
39 37 36 36 36 35 38 39 39 40 42 36 32 40 47 46 45 47 50 4
4 36 37 35 34 34 35 36 35 35 34 35 36 35 35 35 34 34 36
34 36 37 36 40 43 44 46 47 47 46 45 46 51 53 51 49 50 52 5
3 35 35 34 34 33 33 35 36 35 34 36 35 34 34 35 36 34 35 35
37 35 37 37 42 50 51 50 51 50 49 49 49 53 55 52 51 51 53 5

```

-Ortho images

```

/* The DTM grid had lower left (southwest) corner coords:
/*
/*
/*
/*
ncols      580
nrows      440
xllcorner  419870
yllcorner  6194210
cellsize    0.5
NODATA_value -9999
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

-Segmented ortho photography with minimum error

```

/* The DTM grid had lower left (southwest) corner coords:
/*
/*
/*
/*
ncols      580
nrows      440
xllcorner  419870
yllcorner  6194210
cellsize    0.5
NODATA_value -9999
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

```

-Segmented orthoimagery with the moment segmentation method

```

/* The DTM grid had lower left (southwest) corner coords:
/*
/*
/*
/*
ncols      580
nrows      440
xllcorner  419870
yllcorner  6194210
cellsize    0.5
NODATA_value -9999
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

-Extracted road using ArcGIS software

```

/* The DTM grid had lower left (southwest) corner coords:
/*
/*
/*
/*
ncols      580
nrows      440
xllcorner  419870
yllcorner  6194210
cellsize    0.5
NODATA_value -9999
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

-Produced edge map using Canny edge detection

```

/* The DTM grid had lower left (southwest) corner coords:
/*
/*
/*
/*
ncols      580
nrows      440
xllcorner  419870
yllcorner  6194210
cellsize    0.5
NODATA_value -9999
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

```

-NDVI vegetation map

Figure B.23 The input data used in the building footprint extraction and 3D modelling.

The implemented code for building footprint extraction and 3D construction is shown below,

```

//This program is written using Microsoft's Visual Studio 2010 C++
//The author ship of the code is Haval A.Sadeq as a part of the Ph.D. dissertation
//Ph.D. in Geomatics, college of sciences and Engineering, University of Glasgow,
Scotland, UK
//Email
h.sadeq.1@research.gla.ac.uk,website:http://www.gla.ac.uk/schools/ges/pgresearch/hav
alsadeq/
//This code is for building extraction from satellite image
// It is completed and evaluated 03/02/2014
//the input data consisted from eight types:
//1-Digital Surface Model(DSM) of the area produced from Socet GXP
//2-the Labelled nDSM produced from code in Appendix C
//3-the orthoimagery for the area from Socet SET
//4-segmented image with minimum error thresholding method using ImageJ software
//5-segmented image with moment error thresholding method using ImageJ software
//6-extracted road file using the ArcGIS
//7-edge map using Canny edge map algorithm using ImageJ software
//8-NDVI map produce using infrared band exist in the Pleiades satellite imagery using
ArcGIS

//Define header files
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include <float.h>
#include <sstream>
#include <random>
using namespace std;
int c1[1000000],c2[1000000];//define parameters for sub labelling
const double pi=3.14159265358979323846264338327950;//
void sort_nna(int point , int (&point_x)[1000], int (&point_y)[1000]) ;
void doglus_appr(int point , int point_x[1000], int point_y[1000], int
sort_point_x[1000],int sort_point_y[1000], float& thre_dis);
#define rnd 250000//Number of iteration for the simulation purposes
#define points 150//Number of maximum points for building boundary
#define b 580//Number of columns in the data
#define a 440// Number of rows in the data
void erosion_square(int r[a][b] , int (&erosion)[a][b] );// define the MM-erosion
function using square S.E.
void dilation_square(int r[a][b] , int (&dilation)[a][b] );// define the MM
function-dilation using square S.E.
void erosion_circular(int r[a][b] , int (&erosion)[a][b] );// define the MM
function-erosion using square C.E.
void dilation_circular(int r[a][b] , int (&dilation)[a][b] );// define the MM
function-dilation using square C.E.
void labeling( int thr_level[a][b] , int (&label)[a][b], int& max_lab
);//labelling function
void boundary1(int r[a][b] , int (&boundary)[a][b] );//Finding the boundary
using MM the boundary
int boun(int loop, int& min_x,int& min_y, int& max_x, int& max_y ,int& area,int
label[a][b]);// determine the corners of the work area to minimize the time
int main()
{float var1=3;//variance value sigma^2, the variance of the normal probability
distribution that used to estimate random points
float i0,likelihod_d;//define float variables
int i1,i2,i3,i4,i5,i6,i7,i8;//define integer variables
long double rn[points][rnd];//this array is for the generated random variables
rn[number of points][rnd]

```

```

float candidate_p_x[points][rnd],candidate_p_y[points][rnd];
int point, v1; int j, k; int max_lab=-1e10;
//define files for reading the data and saving the output
ofstream myfile1,myfile2,myfile3,myfile4,myfile5;
myfile2.open ("bayesian_result_v3_th_5_5men_10.txt");
string
line1,line2,line3,line4,line5,line6,line7,line8,line9,line10,line11,line12,line13;
// defining arrays for saving the data during the processing
float r0[a][b],r00[a][b],final_dsm[a][b];
int
r1[a][b],r2[a][b],r3[a][b],r3a[a][b],r3b[a][b],r4[a][b],r5[a][b],r01[a][b],shadow
[a][b],temp_r1[a][b],

temp1[a][b],temp1a[a][b],temp1b[a][b],temp2[a][b],temp3[a][b],label[a][b],r0_indv
i[a][b],

intial_boundary[a][b],temp_r[a][b],dilation[a][b],erosion[a][b],harris[a][b],harr
is1[a][b],
harris_co[a][b],canny_specific[a][b],new_canny[a][b];

////////opening first file-this file is related to DSM//////////
char *iname0 = "DSM_Bayesain_merg_resampled_50cm_bilinear_3x3_rmse_0.420m.txt";
//the Used DSM in the Merging
ifstream infile0(iname0);
if (!infile0) {
    cout << "There was a problem opening file "
    << iname0
    << " for reading."
    << endl; return 0; }
cout << "Opened " << iname0 << " for reading." << endl;
getline (infile0,line1);
getline (infile0,line2);
getline (infile0,line3);
getline (infile0,line4);
getline (infile0,line5);
getline (infile0,line6);
getline (infile0,line7);
getline (infile0,line8);
getline (infile0,line9);
getline (infile0,line10);
getline (infile0,line11);
getline (infile0,line12);
getline (infile0,line13);
cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
line11 << endl<< line12 << endl<< line13 << endl;
//////////Reading first file and saveing to array
j=0, k=0;
while (infile0 >> i0)
{if (j==(b-1))
    {r0[k][j]=i0;j=0; k++;}
    else{r0[k][j]=i0;j++;}
    ;}
//////////start extract coordinates of the corner//////////
float x_cor_val,y_cor_val;string txt;

stringstream x_cor (line10);//extracat the coordinates
x_cor>> txt>>x_cor_val;

stringstream y_cor (line11);//extracat the coordinates
y_cor>> txt>>y_cor_val;

////////opening the second file//////////the labelled nDSM from the previous code described
in APPENDIXC C in the thesis//
char *iname1 = "labeled_nDSM.txt";
ifstream infile1(iname1);

```



```

    if (!infile1) {
        cout << "There was a problem opening file "
            << inname1
            << " for reading."
            << endl;
        return 0;
    }
    cout << "Opened " << inname1 << " for reading." << endl;
    getline (infile1,line1);
    getline (infile1,line2);
    getline (infile1,line3);
    getline (infile1,line4);
    getline (infile1,line5);
    getline (infile1,line6);
    getline (infile1,line7);
    getline (infile1,line8);
    getline (infile1,line9);
    getline (infile1,line10);
    getline (infile1,line11);
    getline (infile1,line12);
    getline (infile1,line13);
    cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
    endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
    line11 << endl<< line12 << endl<< line13 << endl;
    //Reading second file and saving to array
    j=0, k=0;
    while (infile1 >> i1)
    {if (j==(b-1))
        {r1[k][j]=i1;j=0; k++;}
        else{r1[k][j]=i1;j++;}
        ;}
    //opening the third file//////////the orthoimagery
    char *inname2 = "ortho_image.txt";
    ifstream infile2(inname2);
    if (!infile2) {
        cout << "There was a problem opening file "
            << inname2
            << " for reading."
            << endl;
        return 0;
    }
    cout << "Opened " << inname2 << " for reading." << endl;
    getline (infile2,line1);
    getline (infile2,line2);
    getline (infile2,line3);
    getline (infile2,line4);
    getline (infile2,line5);
    getline (infile2,line6);
    getline (infile2,line7);
    getline (infile2,line8);
    getline (infile2,line9);
    getline (infile2,line10);
    getline (infile2,line11);
    getline (infile2,line12);
    getline (infile2,line13);
    cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
    endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
    line11 << endl<< line12 << endl<< line13 << endl;
    //Reading third file and saving to array
    j=0, k=0;

    while (infile2 >> i2)
    {if (j==(b-1))
        {r2[k][j]=i2;j=0; k++;}
        else{r2[k][j]=i2;j++;}
        ;}

    //opening fourth file//////////the segmented image with minimum error

    char *inname3 = "segmented_ortho_min_error.txt";

```

```

    ifstream infile3(inname3);
    if (!infile3) {
        cout << "There was a problem opening file "
             << inname3
             << " for reading."
             << endl;
        return 0;
    }
    cout << "Opened " << inname3 << " for reading." << endl;
    getline (infile3,line1);
    getline (infile3,line2);
    getline (infile3,line3);
    getline (infile3,line4);
    getline (infile3,line5);
    getline (infile3,line6);
    getline (infile3,line7);
    getline (infile3,line8);
    getline (infile3,line9);
    getline (infile3,line10);
    getline (infile3,line11);
    getline (infile3,line12);
    getline (infile3,line13);
    cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
    endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
    line11 << endl<< line12 << endl<< line13 << endl;
    //Reading fourth file and saving to array
    j=0, k=0;

    while (infile3 >> i3)
    {if (j==(b-1))
        {r3[k][j]=i3;j=0; k++;}
        else{r3[k][j]=i3;j++;}
        ;}
    //opening fifth file////////the segmented image with moment
    char *inname4 = "segmented_ortho_moment.txt";
    ifstream infile4(inname4);
    if (!infile4) {
        cout << "There was a problem opening file "
             << inname4
             << " for reading."
             << endl;
        return 0;
    }
    cout << "Opened " << inname4 << " for reading." << endl;
    getline (infile4,line1);
    getline (infile4,line2);
    getline (infile4,line3);
    getline (infile4,line4);
    getline (infile4,line5);
    getline (infile4,line6);
    getline (infile4,line7);
    getline (infile4,line8);
    getline (infile4,line9);
    getline (infile4,line10);
    getline (infile4,line11);
    getline (infile4,line12);
    getline (infile4,line13);
    cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
    endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
    line11 << endl<< line12 << endl<< line13 << endl;
    //Reading fifth file and saving to array
    j=0, k=0;

    while (infile4 >> i4)
    {if (j==(b-1))
        {r4[k][j]=i4;j=0; k++;}
        else{r4[k][j]=i4;j++;}
        ;}

```



```

/////opening sixth file//////////the digitized road raster file
char *iname5 = "extracted_road.txt";
ifstream infile5(inname5);
if (!infile5) {
    cout << "There was a problem opening file "
        << inname5
        << " for reading."
        << endl;    return 0;    }
cout << "Opened " << inname5 << " for reading." << endl;
getline (infile5,line1);
getline (infile5,line2);
getline (infile5,line3);
getline (infile5,line4);
getline (infile5,line5);
getline (infile5,line6);
getline (infile5,line7);
getline (infile5,line8);
getline (infile5,line9);
getline (infile5,line10);
getline (infile5,line11);
getline (infile5,line12);
getline (infile5,line13);
cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
line11 << endl<< line12 << endl<< line13 << endl;
//////////Reading sixth file and saving to array
j=0, k=0;

while (infile5 >> i5)
{if (j==(b-1))
    {r5[k][j]=i5;j=0; k++;}
    else{r5[k][j]=i5;j++;}
    ;}
////////opening seventh file////////// //the edge map produced with Canny algorithm

char *iname6 = "edgemap_produced_from_orthophotography.txt";
ifstream infile6(inname6);
if (!infile6) {
    cout << "There was a problem opening file "
        << inname6
        << " for reading."
        << endl;    return 0;    }
cout << "Opened " << inname6 << " for reading." << endl;
getline (infile6,line1);
getline (infile6,line2);
getline (infile6,line3);
getline (infile6,line4);
getline (infile6,line5);
getline (infile6,line6);
getline (infile6,line7);
getline (infile6,line8);
getline (infile6,line9);
getline (infile6,line10);
getline (infile6,line11);
getline (infile6,line12);
getline (infile6,line13);
cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
line11 << endl<< line12 << endl<< line13 << endl;
//////////Reading seventh file and saving to array
j=0, k=0;
while (infile6 >> i6)
{if (j==(b-1))
    {r01[k][j]=i6;j=0; k++;}
    else{r01[k][j]=i6;j++;}
    ;}

```

```

////////opening eighth file//////////the NDVI, vegetation map that produced using
Pleiades

char *iname7 = "ndvi_vegetation_map.txt";
ifstream infile7(inname7);
if (!infile7) {
    cout << "There was a problem opening file "
        << inname7
        << " for reading."
        << endl;    return 0;    }
cout << "Opened " << inname7 << " for reading." << endl;
getline (infile7,line1);
getline (infile7,line2);
getline (infile7,line3);
getline (infile7,line4);
getline (infile7,line5);
getline (infile7,line6);
getline (infile7,line7);
getline (infile7,line8);
getline (infile7,line9);
getline (infile7,line10);
getline (infile7,line11);
getline (infile7,line12);
getline (infile7,line13);
cout << line1 << endl;cout << line2 << endl<< line3 << endl<< line4 << endl<< line5 <<
endl<< line6 << endl<< line7 << endl<< line8<< endl<< line9 << endl<< line10 << endl<<
line11 << endl<< line12 << endl<< line13 << endl;
//////////Reading eighth file and saving to array
j=0, k=0;

while (infile7 >> i7)
{if (j==(b-1))
    {r0_indvi[k][j]=i7;j=0; k++;}
    else{r0_indvi[k][j]=i7;j++;}
    ;}
//////////

//this part to remove shadow because it has pixel less than 25, using ortho photo
for (k=0;k<a;++k)
    for (j=0;j<b;++j)
        if (r2[k][j]<=25)shadow[k][j]= 1;else shadow[k][j]=0;
dilation_circular( shadow , dilation ) ;//to join the parts together
    for (k=0;k<a;++k)for (j=0;j<b;++j)shadow[k][j]=dilation[k][j];
erosion_circular( shadow , erosion ) ;//
for (k=0;k<a;++k)for (j=0;j<b;++j)shadow[k][j]=erosion[k][j];
//////////
for (k=0;k<a;++k)
    for (j=0;j<b;++j)

    {

r3a[k][j]= r3[k][j]-r5[k][j]-shadow[k][j]-r0_indvi[k][j]; ;//subtract road from the
segmented images
r3b[k][j]= r4[k][j]-r5[k][j]-shadow[k][j]-r0_indvi[k][j]; ;//subtract road from the
segmented images

    harris[k][j]=0;
    harris1[k][j]=0;
    harris_co[k][j]=0;

    }

//////////start to subtract NDVI and produce binary image
for (k=0;k<a;++k)
    for (j=0;j<b;++j)
    {
        if (r3a[k][j]==-1 || r3a[k][j]==-2|| r3a[k][j]==-3) r3a[k][j]=0;
    }

```

```

        if (r3b[k][j]==-1 || r3b[k][j]==-2 || r3b[k][j]==-3)r3b[k][j]=0;

        if (r0_indvi[k][j]!=0) r1[k][j]=0; //subtract the NDVI from label images
    }
    ///////////////Apply Mathematical morphology(MM) on the segmented images to remove noise
    and enhance the image
    //first segmented image
    erosion_circular( r3a , erosion ) ;//Apply MM erosion with C.E. once
    for (k=0;k<a;++k)for (j=0;j<b;++j)r3a[k][j]=erosion[k][j];

    erosion_square( r3a , erosion ) ;//Apply MM erosion with S.E. once
    for (k=0;k<a;++k)for (j=0;j<b;++j)r3a[k][j]=erosion[k][j];
    erosion_square( r3a , erosion ) ;//Apply MM erosion with S.E. once
    for (k=0;k<a;++k)for (j=0;j<b;++j)r3a[k][j]=erosion[k][j];

    dilation_square( r3a , dilation ) ;//Apply MM dialtion with S.E. once
    for (k=0;k<a;++k)for (j=0;j<b;++j)r3a[k][j]=dilation[k][j];
    dilation_square( r3a , dilation ) ;//Apply MM dilation with S.E. once
    for (k=0;k<a;++k)for (j=0;j<b;++j)r3a[k][j]=dilation[k][j];

    ///////////////
    dilation_circular( r3a , dilation ) ;//Apply MM dilation with C.E. once
    for (k=0;k<a;++k)for (j=0;j<b;++j)r3a[k][j]=dilation[k][j];// the r3a used as
    inital boundary

    dilation_circular( r3a , temp2 ) ;//Apply MM dilation with C.E. once

    boundary1( temp2 , temp2 ) ;//boundary function to be used to modify Canny edge map

    //Second segmented image
    erosion_square( r3b , erosion ) ;//Apply MM erosion with S.E. once
    for (k=0;k<a;++k)for (j=0;j<b;++j)r3b[k][j]=erosion[k][j];
    erosion_square( r3b , erosion ) ;//Apply MM erosion with S.E. once
    for (k=0;k<a;++k)for (j=0;j<b;++j)r3b[k][j]=erosion[k][j];
    dilation_square( r3b , dilation ) ;//Apply MM dilation with S.E. once
    for (k=0;k<a;++k)for (j=0;j<b;++j)r3b[k][j]=dilation[k][j];
    dilation_square( r3b , dilation ) ;//Apply MM dilation with S.E. once
    for (k=0;k<a;++k)for (j=0;j<b;++j)r3b[k][j]=dilation[k][j];

    ///////////////
    dilation_circular( r3b , temp3 ) ;////Apply MM dilation with C.E. once
    boundary1( temp3 , temp3 ) ;//boundary function to be used to modify Canny edge map

    for (k=0;k<a;++k)for (j=0;j<b;++j)
    if (temp2[k][j]==1 ||temp3[k][j]==1 )r01[k][j]=1;//merge the produced above boundary
    with the Canny edge map

    ///////////////
    //decrease the size of the segmented images and the subtract them the above Canny
    edge map
    erosion_circular( r3a , erosion ) ;//Apply MM erosion with C.E. once
    for (k=0;k<a;++k)for (j=0;j<b;++j)temp2[k][j]=erosion[k][j];
    erosion_circular( r3b , erosion ) ;//Apply MM erosion with C.E. once
    for (k=0;k<a;++k)for (j=0;j<b;++j)temp3[k][j]=erosion[k][j];

    ///////////////
    //Vacant data in the Canny edge map whenever there is data in the segmnted image
    for (k=0;k<a;++k)for (j=0;j<b;++j)
    if (temp2[k][j]==1 || temp3[k][j]==1)r01[k][j]=0;

    ///////////////clean the edge map by removing noise

```

```

//////////make a mask 7x7 to clean the objects in the 5x5 area
    for (k=0;k<a-6;++k)
        for (j=0;j<b-6;++j)
if (r01[k][j]==0 && r01[k][j+1]==0 && r01[k][j+2]==0 && r01[k][j+3]==0 &&
r01[k][j+4]==0 && r01[k][j+5]==0 && r01[k][j+6]==0 &&
    r01[k+1][j]==0 &&
r01[k+1][j+6]==0 &&
    r01[k+2][j]==0 &&
r01[k+2][j+6]==0 &&
    r01[k+3][j]==0 &&
r01[k+3][j+6]==0 &&
    r01[k+4][j]==0 &&
r01[k+4][j+6]==0 &&
    r01[k+5][j]==0 &&
r01[k+5][j+6]==0 &&
    r01[k+6][j]==0 && r01[k+6][j+1]==0 && r01[k+6][j+2]==0 && r01[k+6][j+3]==0 &&
r01[k+6][j+4]==0 && r01[k+6][j+5]==0 && r01[k+6][j+6]==0 )

    {
        r01[k+1][j+1]=0 ; r01[k+1][j+2]=0 ; r01[k+1][j+3]=0 ; r01[k+1][j+4]=0 ;
r01[k+1][j+5]=0 ;
        r01[k+2][j+1]=0 ; r01[k+2][j+2]=0 ; r01[k+2][j+3]=0 ; r01[k+2][j+4]=0 ;
r01[k+2][j+5]=0 ;
        r01[k+3][j+1]=0 ; r01[k+3][j+2]=0 ; r01[k+3][j+3]=0 ; r01[k+3][j+4]=0 ;
r01[k+3][j+5]=0 ;
        r01[k+4][j+1]=0 ; r01[k+4][j+2]=0 ; r01[k+4][j+3]=0 ; r01[k+4][j+4]=0 ;
r01[k+4][j+5]=0 ;
        r01[k+5][j+1]=0 ; r01[k+5][j+2]=0 ; r01[k+5][j+3]=0 ; r01[k+5][j+4]=0 ;
r01[k+5][j+5]=0 ;
    }

//////////assign the value zero to the a new array
    int v;
    for (k=0;k<a;++k)
        for (j=0;j<b;++j)
            final_dsm[k][j]=0;

//////////stat building extraction
    //estimate the number of labels in the label layer
    int max_lab_count=0;
    for (k=0;k<a;++k)
        for (j=0;j<b;++j)
            if (r1[k][j]>max_lab_count) max_lab_count=r1[k][j];
    cout <<"max_lab_count"<<max_lab_count<<endl;

    for (int lab_co=1;lab_co<=max_lab_count;++lab_co) // start reading each label(building)
    individually
    {
        for (k=0;k<a;++k)
            for (j=0;j<b;++j)
                if (r1[k][j]==lab_co){temp_r[k][j]=1;temp_r1[k][j]=1;}// produce a new layer
                containing only one label
                else {temp_r[k][j]=0;temp_r1[k][j]=0;}

//////////
        for (k=0;k<a;++k)
            for (j=0;j<b;++j)
                {if (temp_r[k][j]!=0){v=0;goto start;}}//this to check whether there is a label or
                not
                {v=1; goto continu1;}
        //////////
    start:
        cout <<"building label no.: "<<lab_co<<endl;

```

```

    for (int dil=0;dil<17;++dil)
    {cout<<"dilate"<<endl;
    dilation_circular( temp_r , dilation );//Apply MM dilation with C.E. on the labelled
    object seventeen times
        for (k=0;k<a;++k)for (j=0;j<b;++j)temp_r[k][j]=dilation[k][j];
    }
    //////////////find the boundary of the selected region to minimize the time
    int max_xb=1e-10,max_yb=1e-10,min_xb=1e10,min_yb=1e10,areab=0;
    boun( 1, min_xb, min_yb, max_xb, max_yb , areab ,temp_r);//use function to calculate
    the corners of the work area
    int a_start=0;
    int a_end=a;
    int b_start=0;
    int b_end=b;
    //////////////find the corresponding part from the segmented image
    for (k=a_start;k<a_end;++k)
        for (j=b_start;j<b_end;++j)
        {
            if (temp_r[k][j]==1)
            {temp1a[k][j]=r3a[k][j];temp1b[k][j]=r3b[k][j];}
            else
            {temp1a[k][j]=0;temp1b[k][j]=0;} }
    ////////////////check the area and pick the biggest////////////////////
    //then element the other parts          //////////////////
    int area_a=0, area_b=0;
    for (k=a_start;k<a_end;++k)
    for (j=b_start;j<b_end;++j)
    {
        if (temp1a[k][j]==1)area_a=area_a+1;if (temp1b[k][j]==1)area_b=area_b+1;
    }
    ////////////////add the boundary either from the segmented image 1(area_a) or from
    segmented
    ////////////////image 2(area_b) and save it to the new array called
    temp1////////////////////

        if (area_a>area_b)
        {
            for (k=a_start;k<a_end;++k)
            for (j=b_start;j<b_end;++j)
                temp1[k][j]=temp1a[k][j];
            }
        else
        {
            for (k=a_start;k<a_end;++k)
            for (j=b_start;j<b_end;++j)
                temp1[k][j]=temp1b[k][j];
            }

        labeling( temp1 , label , max_lab );// label the objects in array1
    ///create new array with only the final object and taking the elevation of maximum from
    corresponding original dsm
    int k1,j1;
    float height,min_height,max_height ;
    for (int lab=1;lab<=max_lab;++lab)

    {
        max_height = -1e10;min_height=1e10;int area_matching=0;
        ///create new blank arrayies
    for (k=a_start;k<a_end;++k)
    for (j=b_start;j<b_end;++j)
        {harris[k][j]=0; new_canny[k][j]=0;intial_boundary[k][j]=0;}

    for (k=a_start;k<a_end;++k)
    for (j=b_start;j<b_end;++j)
    {
        if (label[k][j]==lab)

```

```

        {if (r0[k][j]>max_height) max_height=r0[k][j];//find the max height at each
sub-label
        if (r0[k][j]<min_height) min_height=r0[k][j];//find the min height at each
sub-label
        harris[k][j]=1; //create an array with value 1
        intial_boundary[k][j]=1;//create an array with value 1
        }
    }
    ///////////////17/12/2013 pick the common segment on the
    for (k=a_start;k<a_end;++k)
        for (j=b_start;j<b_end;++j)
// find weather the sub-label area and the segmented image are common

        if (harris[k][j]!=0 && temp_r1[k][j]!=0){area_matching=1;}

        if (area_matching==1){ goto start2;}//if common goto to start2 in order to
implement boundary regularization
        ///////////////otherwise skip this sub-label and check the other sublabel
        else { goto continu1;}

    ///////////////
    start2:

//dilate and erode each building individually to close the holes
start1:

    for (int dil=0;dil<10;++dil)
    {
dilation_square ( harris , dilation );//Apply MM dilation with S.E. ten times
        for (k=a_start;k<a_end;++k)
            for (j=b_start;j<b_end;++j)
                harris[k][j]=dilation[k][j];
    }

//find the centre of the area
    int cen_k=0,cen_j=0,p_k=0,p_j=0;
    for (k=a_start;k<a_end;++k)
        for (j=b_start;j<b_end;++j)
            if (harris[k][j]==1)
                { cen_k=cen_k+k;p_k=p_k+1;
                  cen_j=cen_j+j;p_j=p_j+1;

                }
    if (p_k==0 ||p_j==0) goto end_this_loop1;
    cen_k=cen_k/p_k;cen_j=cen_j/p_j;
    cout<<cen_k<< " " <<cen_j<<endl;

    end_this_loop1;;
    for (int ero=0;ero<14;++ero)
        {erosion_square( harris , erosion) ;//Apply MM dilation with S.E. fourteen times
            for (k=a_start;k<a_end;++k)for
(j=b_start;j<b_end;++j)harris[k][j]=erosion[k][j];
        }

    for (k=a_start;k<a_end;++k)
        for (j=b_start;j<b_end;++j)
            {if (harris[k][j]==1)goto end_this_loop2;
              else harris[cen_k][cen_j]=1;
            }
    end_this_loop2;;

    ///////////////record the result in new array which is called final_dsm[ ][ ]
    { for (k1=0;k1<a;++k1)
        for (j1=0;j1<b;++j1)
            if (harris1[k1][j1]==1)final_dsm[k1][j1]=max_height;
    }

```

```

    }
    ////////////////detect the boundary
    //create new edge map contain only the candidate image,

    for (k=a_start;k<a_end;++k)
        for (j=b_start;j<b_end;++j)
            canny_specific[k][j]=r01[k][j];
    for (k=a_start;k<a_end;++k)
        for (j=b_start;j<b_end;++j)
            if (harris[k][j]==1)canny_specific[k][j]=0;
    //the following steps is to produce a 'solo' edge map
    boundary1 ( harris , harris) ;//boundary function

float tk,tj;int tk_int,tj_int;
    for (k=a_start;k<a_end;++k)
        for (j=b_start;j<b_end;++j)
            if (harris[k][j]==1)// if any pixel detected then start to detect points on canny
            edge detector
            {
                //this part is used to create a new array with just boundary pixels
                for (int theta =0;theta<360;theta=theta+20)
                    //int theta =0;
                    for (float dist =0;dist<a;dist=dist+0.5)
                        {
                            tj=j+cos(theta*pi/180)*dist;
                            tk=k+sin(theta*pi/180)*dist;

            ///convert the the numbers to integer and remove the decimals in order to fill in the
            pixel exactly
            if( (tk + 0.5) >= (int(tk) + 1) )
                tk_int= int(tk)+1;
            else tk_int=int(tk);

            if( (tj + 0.5) >= (int(tj) + 1) )
                tj_int = int(tj)+1;
            else
                tj_int=int(tj);
            ////////////////

            if (canny_specific[tk_int][tj_int]==1) {new_canny[tk_int][tj_int]=1;
;break; }
            if (canny_specific[tk_int][tj_int+1]==1)
            {new_canny[tk_int][tj_int+1]=1;;break;}
            if (canny_specific[tk_int+1][tj_int+1]==1)
            {new_canny[tk_int+1][tj_int+1]=1;;break;}
            if (canny_specific[tk_int+1][tj_int]==1)
            {new_canny[tk_int+1][tj_int]=1;;break;}

            }

        }
    }
    //fill the gaps in the generated canny
    int pixel,temp_k, temp_j;
    for (k=a_start;k<a_end;++k)
        for (j=b_start;j<b_end;++j)
        {
            if (new_canny[k][j]==0 && new_canny[k][j-1]==1 &&
new_canny[k][j+1]==1)new_canny[k][j]=1;
            if (new_canny[k][j]==0 && new_canny[k][j+1]==1 &&
new_canny[k+1][j-1]==1)new_canny[k][j]=1;
            if (new_canny[k][j]==0 && new_canny[k-1][j-1]==1 &&
new_canny[k+1][j+1]==1)new_canny[k][j]=1;
            if (new_canny[k][j]==0 && new_canny[k-1][j-1]==1 &&
new_canny[k+1][j]==1)new_canny[k][j]=1;
            if (new_canny[k][j]==0 && new_canny[k-1][j+1]==1 &&
new_canny[k+1][j]==1)new_canny[k][j]=1;

```

```

}

//////////connect the pixels using mathematical morphology to produce
initial building boundary for the optimization

dilation_square ( intial_boundary , dilation );//Apply MM dilation with S.E.
for (k=a_start;k<a_end;++k)for
(j=b_start;j<b_end;++j)intial_boundary[k][j]=dilation[k][j];

erosion_square ( intial_boundary , erosion );//Apply MM erosion with S.E.
for (k=a_start;k<a_end;++k)for
(j=b_start;j<b_end;++j)intial_boundary[k][j]=erosion[k][j];

    boundary1(intial_boundary,intial_boundary);//find the boundary
    ////////////get the common parts in order to find the exterior
boundary only

    int point_x[10000],point_y[10000] ;
    float thresh;
    point=0;
for (k=a_start;k<a_end;++k)
    for (j=b_start;j<b_end;++j)

        if(intial_boundary[k][j]!=0){point_x[point]=j;point_y[point]=k;point=point+1;
//new_canny_original[k][j]=0;
}

        if (point <= 3) {goto continu11;}//rule: if the number of the points in the
initial boundary is less or equal to 3 then neglect this label and goto the another one
sort_nna( point , point_x, point_y);//sort the boundary

    ////////////
    thresh=5;// boundary approximation threshold
    doglus_appr(point, point_x, point_y, point_x, point_y,thresh);//approximate the
boundary

    int new_points=0,temp_x0,temp_y0,temp_x1,temp_y1;
    for (int s1=0;s1<point;++s1)
    if(point_x[s1] !=-100000000)new_points=new_points+1;
    point=new_points;

if (point <= 3) {goto continu11;}
//////////
thresh=5;
    doglus_appr(point, point_x, point_y, point_x, point_y,thresh);
    new_points=0;
    for (int s1=0;s1<point;++s1)
    if(point_x[s1] !=-100000000)new_points=new_points+1;
    point=new_points;

if (point <= 3) {goto continu11;}
//////////
    for (int s1=0;s1<point;++s1)
    {
cout<<point_x[s1]<<" " <<point_y[s1]<<" " <<max_height<<endl;// print the
coordinates on the screen
// save the initial boundary coordinate to a file
    myfile5<< std::fixed << std::setprecision(2)<<s1<<" "<<lab_co<<lab<<" "<<
point_x[s1]/2+x_cor_val<<" "<<(a-point_y[s1])/2+y_cor_val<<" "<<max_height<<endl;
    }

    //start Bayesian
    //create new array for the r0 and for canny
    for (k=a_start;k<a_end;++k)
    for (j=b_start;j<b_end;++j)
        if(harris1[k][j]==1)r00[k][j]=1;else r00[k][j]=0;

```



```

int px,py,itr=0;
float new_point_x[points],new_point_y[points],minimum_value_arg=1e20, argmin[rnd];
//generate random points for the corner boundary simulation
std::default_random_engine generator;
std::normal_distribution<double> distribution(0,var1);

for (int i=0; i<rnd; i=i+1)
{
    for (int s=0;s<point;++s)
    {new_point_x[s]=point_x[s]+
    distribution(generator);candidate_p_x[s][i]=new_point_x[s];
    new_point_y[s]=point_y[s]+
    distribution(generator);candidate_p_y[s][i]=new_point_y[s];

    }

    float d[points],mean,var;

float px0,py0,px1,py1,d_lik_seg, m, m_per, px_new, py_new,
    px_new_per_out,py_new_per_out,px_new_per_in,py_new_per_in,d_likelihood[100000];//
d_likelihood=0;
int px_out,py_out,px_in,py_in,d_cou=0,dis,sign_v1,sign_v2,sign,dis_sign;

    //find the distance in the new_canny to each line and find the sign then find
    for each line
    for (k=0;k<a;++k)
        for (j=0;j<b;++j)

            if (new_canny[k][j]==1)
            {
                //find the distance for k,j to the

float px2=j,px3=j+1;
float py2=k,py3=k;

                //
float min_dist=10000;
int cou_poi=0;
for (int s=0;s<point;++s)
    {
        if(s!=point-1){
            px0=new_point_x[s];py0=new_point_y[s];px1=new_point_x[s+1];py1=new_point_y[s+1];}
        if(s==point-1){
            px0=new_point_x[point-1];py0=new_point_y[point-1];px1=new_point_x[0];py1=new_point_y
            [0];}

            d_lik_seg=sqrt((px0-px1)*(px0-px1)+(py0-py1)*(py0-py1));
            m=(py1-py0)/(px1-px0);//slope
            m_per=-1/m;//slope of perpendicular line
            //
float di=((px0-px1)*(py2-py3)-(py0-py1)*(px2-px3));
            if (di==0) continue; //no intersection are parallel
            else {
                float
distance=abs((px1-px0)*(py0-k)-(px0-j)*(py1-py0))/sqrt((px1-px0)*(px1-px0)+(py1-py0)
*(py1-py0)) ;
                if (distance<=min_dist)min_dist=distance;
                float pre=(px0*py1)-(py0*px1), post=(px2*py3)-(py2*px3);
                float x =(pre*(px2-px3)-(px0-px1)*post)/di;
                float y =(pre*(py2-py3)-(py0-py1)*post)/di;

```

```

        if (x>px3&& y<=py1 && y>=py0)cou_poi=cou_poi+1;

    }

}
if (cou_poi==0 ||cou_poi==2 ||cou_poi==4||cou_poi==6 ||cou_poi==8
||cou_poi==10||cou_poi==12||cou_poi==14)min_dist=-1*min_dist;
    else min_dist=abs(min_dist);

    d_likelihood[d_cou]=min_dist;
    d_cou=d_cou+1;
}

    likelihhod_d=0;
for (int ii=0; ii<d_cou; ++ii)
{

    likelihhod_d = likelihhod_d + d_likelihood[ii]*d_likelihood[ii];// likelihood
    probabaility value
}

//////////
//////////start Bayesian theorem
//////////calculate the angles between points of the new profiles for the prior calculation

long double edg_a,edg_b,edg_c,ang[points],penalty, prior;;

edg_a=sqrt((( new_point_x[1]-new_point_x[0])*( new_point_x[1] -new_point_x[0])) + ((
new_point_y[1]-new_point_y[0])*( new_point_y[1] -new_point_y[0])));
edg_b=sqrt((( new_point_x[0]-new_point_x[point-1])*( new_point_x[0]
-new_point_x[point-1])) + (( new_point_y[0]-new_point_y[point-1])*( new_point_y[0]
-new_point_y[point-1])));
edg_c=sqrt((( new_point_x[1]-new_point_x[point-1])*( new_point_x[1]
-new_point_x[point-1])) + (( new_point_y[1]-new_point_y[point-1])*( new_point_y[1]
-new_point_y[point-1])));

ang[0]=acos((edg_a*edg_a+edg_b*edg_b-edg_c*edg_c)/(2*edg_a*edg_b) ) * 180 / pi;// angle
in degrees

edg_a=sqrt((( new_point_x[0]-new_point_x[point-1])*( new_point_x[0]
-new_point_x[point-1])) + (( new_point_y[0]-new_point_y[point-1])*( new_point_y[0]
-new_point_y[point-1])));
edg_b=sqrt((( new_point_x[point-2]-new_point_x[point-1])*( new_point_x[point-2]
-new_point_x[point-1])) + (( new_point_y[point-2]-new_point_y[point-1])*(
new_point_y[point-2] -new_point_y[point-1])));
edg_c=sqrt((( new_point_x[0]-new_point_x[point-2])*( new_point_x[0]
-new_point_x[point-2])) + (( new_point_y[0]-new_point_y[point-2])*( new_point_y[0]
-new_point_y[point-2])));

    ang[point-1]=acos((edg_a*edg_a+edg_b*edg_b-edg_c*edg_c)/(2*edg_a*edg_b) ) * 180 /
pi;// angle in degrees

    for (int s=1;s<point-1;++s)
    {

edg_a=sqrt((( new_point_x[s]-new_point_x[s-1] ) * ( new_point_x[s]
-new_point_x[s-1])) + (( new_point_y[s]-new_point_y[s-1])*( new_point_y[s]
-new_point_y[s-1])));
edg_b=sqrt((( new_point_x[s]-new_point_x[s+1] ) * (new_point_x[s] -new_point_x[s+1]))
+ (( new_point_y[s]-new_point_y[s+1])*( new_point_y[s] -new_point_y[s+1])));
edg_c=sqrt((( new_point_x[s-1]-new_point_x[s+1]) * ( new_point_x[s-1]
-new_point_x[s+1])) + (( new_point_y[s-1]-new_point_y[s+1])*( new_point_y[s-1]
-new_point_y[s+1])));

```

```

    ang[s]=acos((edg_a*edg_a+edg_b*edg_b-edg_c*edg_c)/(2*edg_a*edg_b) ) * 180 / pi;//
    convert angle to degrees

}

//////////define the penalty using the defined equation in the
thesis

prior=0;
for (int s=0;s<point;++s)

{
    if(ang[s]<87 && ang[s]>0)    penalty=

        ((0.21-6/((ang[s]-45)*(ang[s]-45)+60))
        -2/((ang[s]-90)*(ang[s]-90)+9.7)
        -6/((ang[s]-135)*(ang[s]-135)+60)
        -2/((ang[s]-180)*(ang[s]-180)+9.65))/29.92;

    if(ang[s]<177 && ang[s]>93)    penalty= ((0.21-6/((ang[s]-45)*(ang[s]-45)+60))
        -2/((ang[s]-90)*(ang[s]-90)+9.7)
        -6/((ang[s]-135)*(ang[s]-135)+60)
        -2/((ang[s]-180)*(ang[s]-180)+9.65))/29.92;

    if(ang[s]<=93 && ang[s]>=87) penalty= 1e-45;
    if(ang[s]<=180 && ang[s]>=177) penalty= 1e-45;
    prior=prior+(log10(penalty));

}

//////////end priori//////////
argmin[i]=1*likelihod_d/(2*var1)+1*prior;//find the Maximum a Posteriori
if (argmin[i]<minimum_value_arg) {minimum_value_arg=argmin[i];itr=i;};
}
cout<<"itr:"<<itr<<"      "<<argmin[itr]<<"      "<<minimum_value_arg<<endl<<endl;//
prnt the result on the screen

// test if the points are three nodes or not, after applying doglus approximate

thresh=0.5;
new_points=0,temp_x0,temp_y0,temp_x1,temp_y1;
for (int s1=0;s1<point;++s1)
    if(point_x[s1] !=-100000000)new_points=new_points+1;
point=new_points;
//////////
new_points=0,temp_x0,temp_y0,temp_x1,temp_y1;
for (int s1=0;s1<point;++s1)
    if(point_x[s1] !=-100000000)new_points=new_points+1;
point=new_points;
//////////
for (int s=0;s<point;++s)
{
    new_point_x[s]=candidate_p_x[s][itr];new_point_y[s]=candidate_p_y[s][itr];

    cout<<"Bayesian result boundary"<<" "<<s<<" "<<lab_co<<lab<<" "<< new_point_x[s]<<
    " "<<new_point_y[s]<<" "<<max_height<<endl;

    new_point_x[s]=candidate_p_x[s][itr]/2+x_cor_val;// this is real coordinate
system
    new_point_y[s]=(a-candidate_p_y[s][itr])/2+y_cor_val;
    myfile2<< std::fixed << std::setprecision(2)<<s<<" "<<lab_co<<lab<<" "<<
    new_point_x[s]<<" "<<new_point_y[s]<<" "<<max_height<<endl;

}
continu11::if (point <=3 )continue;//when it reaches here it will still in the loop

```

```

therefore i used continue to skip this loop

}
continu1::if (v==1 )continue ;
}
/////close the files
myfile2.close( );
infile0.close( );  infile1.close( );  infile2.close( );  infile3.close( );
infile4.close( );  infile5.close( );  infile6.close( );  infile7.close( );

return 0;}
//*****
//*****
//*****
//////////Functions that used in building extraction code //////////

////////// start calculate the erosion
void erosion_square(int r[a][b] , int (&erosion)[a][b] )
{int r_temp[a][b];
for(int k=0;k<a;k++)
    for(int j=0;j<b;j++)
        erosion[k][j]=0;

//////////square structure
for(int k=1;k<a-1;k++)
    for(int j=1;j<b-1;j++)
    {

if (r[k][j]==r[k-1][j] && r[k][j]==r[k+1][j] && r[k][j]==r[k][j-1] &&
r[k][j]==r[k][j+1] && //box element
    r[k][j]==r[k-1][j-1]&& r[k][j]==r[k-1][j+1]&& r[k][j]==r[k+1][j-1]&&
r[k][j]==r[k+1][j+1] )//first upper left corner calculator

        erosion[k][j]= r[k][j];
    else erosion[k][j]=0;    }

}
////////// End calculate the erosion
////////// start calculate the erosion Circular 5x5
void erosion_circular(int r[a][b] , int (&erosion)[a][b] )
{int r_temp[a][b];
for(int k=0;k<a;k++)
    for(int j=0;j<b;j++)
        erosion[k][j]=0;
//////////square structure
for(int k=2;k<a-2;k++)
    for(int j=2;j<b-2;j++)
    {

if (r[k][j]==r[k-1][j] && r[k][j]==r[k+1][j] && r[k][j]==r[k][j-1] &&
r[k][j]==r[k][j+1] && //box element
    r[k][j]==r[k-1][j-1]&& r[k][j]==r[k-1][j+1]&& r[k][j]==r[k+1][j-1]&&
r[k][j]==r[k+1][j+1] && //box element
    r[k][j]==r[k-2][j]&& r[k][j]==r[k+2][j]&& r[k][j]==r[k][j-2]&&
r[k][j]==r[k+2][j]

        )//first upper left corner calculator

        erosion[k][j]= r[k][j];
    else erosion[k][j]=0;    }

} //////////// End calculate the erosion circular SE 5x5
////////// start calculate the function
void dilation_square(int r[a][b] , int (&dilation)[a][b] )
{

```

```

    for(int k=0;k<a;k++)
{ dilation[k][0]=0; dilation[k][b-1]=0;}
for(int j=0;j<b;j++)
{ dilation[0][j]=0; dilation[a-1][j]=0;}

for(int k=1;k<a-1;k++)

    for(int j=1;j<b-1;j++)
    {
        if (r[k-1][j-1]==1 || r[k-1][j]==1|| r[k-1][j+1]==1|| r[k][j-1]==1||
r[k][j]==1|| r[k][j+1]==1|| r[k+1][j-1]==1 || r[k+1][j]==1|| r[k+1][j+1]==1)//box
element

        dilation[k][j]=1;else dilation[k][j]=0 ;}
    }
    //start calculate the function circular SE 5x5

void dilation_circular(int r[a][b] , int (&dilation)[a][b] )
{for(int k=0;k<a;k++)
    for(int j=0;j<b;j++)
        dilation[k][j]=0;

for(int k=2;k<a-2;k++)

    for(int j=2;j<b-2;j++)
    {
        if (r[k-1][j-1]==1 || r[k-1][j]==1|| r[k-1][j+1]==1|| r[k][j-1]==1||
r[k][j]==1||
        r[k][j+1]==1|| r[k+1][j-1]==1 || r[k+1][j]==1|| r[k+1][j+1]==1||
r[k-2][j]==1|| r[k+2][j]==1 || r[k][j-2]==1|| r[k][j+2]==1 )
            dilation[k][j]=1;else dilation[k][j]=0 ;
    }
}
//start labelling//
void labeling( int thr_level[a][b] , int (&label)[a][b] , int& max_lab )

{ int j; int lab=0, newlabel;
    if (thr_level[0][0] == 1) label[0][0]=1;else label[0][0]=0;//check first
corner

    for (j=0;j<b;++j)
    {
        if (thr_level[0][j] == 1)// 1 is foreground , 0 is for background
        { if (thr_level[0][j] == thr_level[0][j-1] )
        {
            label[0][j]= label[0][j-1];
        }
            else {++lab; label[0][j]=lab;}
        }
    }
    else label[0][j]=0;
}

//for the other rows
int ch=0,k;
for (k=0;k<a;++k)
    for (j=0;j<b;++j)
    {

if (thr_level[k][j] == 1)
{
    if (j == 0) if (thr_level[k-1][j]== 1 ) { label[k][j] = label[k-1][j];};//test the
first column
    else {++lab; label[k][j]=lab;}}
}

```

```

//////////for the other columns rather than first column
//case1
//else
{
    if (thr_level[k][j] == thr_level[k][j-1] && thr_level[k][j] == thr_level[k-1][j]
)
    { if ( label[k-1][j] > label[k][j-1])
      { label[k][j]= label[k][j-1];ch=ch+1;;c1[ch]= label[k-1][j];c2[ch]=
label[k][j-1];    }

      else {label[k][j]= label[k-1][j];ch=ch+1;;c1[ch]= label[k][j-1];c2[ch]=
label[k-1][j];}}

//case2
if (thr_level[k][j] != thr_level[k][j-1] && thr_level[k][j] == thr_level[k-1][j] )
    label[k][j]= label[k-1][j];

//case3
if (thr_level[k][j] == thr_level[k][j-1] && thr_level[k][j] != thr_level[k-1][j] )
    label[k][j]= label[k][j-1];

//case 4
if (thr_level[k][j] != thr_level[k][j-1] && thr_level[k][j] != thr_level[k-1][j] )
    {++lab; label[k][j]=lab;}
//

    }}

else label[k][j]=0; }

//////////start second pass

for (int ch1=0;ch1<=ch;++ch1)
{   if (c1[ch1] != c2[ch1] )
    {
        for (k=0;k<a;++k)
        for (j=0;j<b;++j)
        {
            if (label[k][j]==c2[ch1] && c1[ch1] > c2[ch1]) label[k][j]=c1[ch1];
            else if (label[k][j]==c1[ch1] && c1[ch1] < c2[ch1]) label[k][j]=c2[ch1];

        }}

//////////scan the image for discrepancy in the labels
for (int ss=0;ss<25;++ss)// number of loops this should be treated automatically
{
    for (k=0;k<a;++k)
    for (j=0;j<b;++j)
    if (thr_level[k][j] == 1)
    {
        if (thr_level[k][j] == thr_level[k-1][j-1] )
            if ( label[k][j] > label[k-1][j-1]) label[k][j]=label[k-1][j-1];
        if (thr_level[k][j] == thr_level[k-1][j] )
            if ( label[k][j] > label[k-1][j]) label[k][j]=label[k-1][j];
        if (thr_level[k][j] == thr_level[k-1][j+1] )
            if ( label[k][j] > label[k-1][j+1]) label[k][j]=label[k-1][j+1];
        if (thr_level[k][j] == thr_level[k][j-1] )
            if ( label[k][j] > label[k][j-1]) label[k][j]=label[k][j-1];
        if (thr_level[k][j] == thr_level[k][j+1] )
            if ( label[k][j] > label[k][j+1]) label[k][j]=label[k][j+1];
        if (thr_level[k][j] == thr_level[k+1][j-1] )
            if ( label[k][j] > label[k+1][j-1]) label[k][j]=label[k+1][j-1];
        if (thr_level[k][j] == thr_level[k+1][j] )

```

```

        if ( label[k][j] > label[k+1][j]) label[k][j]=label[k+1][j];
    if (thr_level[k][j] == thr_level[k+1][j+1] )
        if ( label[k][j] > label[k+1][j+1]) label[k][j]=label[k+1][j+1];
    }
}

for (k=0;k<a;++k)
    for (j=0;j<b;++j)
        { if (label[k][j]>max_lab) max_lab=label[k][j];}

}
//////////end of labelling

//////////sort the points using Nearest neighbour algorithm
void sort_nna(int point , int (&point_x)[1000], int (&point_y)[1000])
{
    int vx[1000],vy[1000];
    vx[0]=point_x[0],vy[0]=point_y[0];point_x[0]=-1000;point_y[0]=-1000;float
d;

    for (int s=0;s<point;++s)
    { d=10e10;

        for (int s1=0;s1<point;++s1)
        {

            if (vx[s]==point_x[s1] &&
vy[s]==point_y[s1]){point_x[s1]=-1000;point_y[s1]=-1000; }//to make the point visited
        }
        for (int s1=0;s1<point;++s1)
        {

            if (point_x[s1]!=-1000 && point_y[s1]!=-1000 )
            {
                float
d1=sqrt((float)(vx[s]-point_x[s1])*(vx[s]-point_x[s1])+(vy[s]-point_y[s1])*(vy[s]-po
int_y[s1]));
                if (d1<= d){d=d1;vx[s+1]=point_x[s1];vy[s+1]=point_y[s1]; }

            }

        }

    }
    for (int s=0;s<point;++s)
    {point_x[s]=vx[s];point_y[s]=vy[s];}

}

//////////
//////////Start doglus approximate algorithm to minimize number of point in the boundary
//////////
void doglus_appr(int point , int point_x[1000], int point_y[1000], int
sort_point_x[1000],int sort_point_y[1000],float& thre_dis)
{
    int temp_x0,temp_y0;
    temp_x0=point_x[0];temp_y0=point_y[0];
    for (int s1=0;s1<point;++s1)
    {point_x[s1]=point_x[s1+1];point_y[s1]=point_y[s1+1];}
    point_x[point-1]=temp_x0;point_y[point-1]=temp_y0;

    float thresh;
    thresh=thre_dis;
    int point1, st;
    point1=point;
    int s1=0,end1;

```

```

float m,b1,d,px0,py0,px1,py1,px,py;
int index,temp,temp1;

float end= point1;

st=s1;
end1=point1-1;

do{
start_dog1:

    px0=point_x[st];
    py0=point_y[st];
    px1=point_x[end1];
    py1=point_y[end1];

    m=(py1-py0)/(px1-px0);//slope
    b1=py1-m*px1 ;

    float max_dis=-10000;
    index=0;
    for (int s11=st;s11<end1 ;++s11)

    {

        px=point_x[s11];// the coordinate at each point
        py=point_y[s11];
        d=abs(py-m*px-b1)/sqrt(m*m+1);// calculate the distance from point to the line

        if (px0==px1) d= abs(px1-px);

        if (d< 0.0001) d=0;
        if(d>max_dis){max_dis=d;index=s11;}//find maximum distance
    }
    if ((st== index) && (st+1 == end1)) {st=st+1;goto skip_parameter;}

    if
    (max_dis>thresh){
        end1=index;goto start_dog1;}
    else
    {
        for (int s11=st+1;s11<end1 ;++s11)
        { point_x[s11]=-100000000;point_y[s11]=-100000000;
        }
        ////////////////calculate the points number again
        ///1-sort the points
        ///2-calculate the number of points
        for (int i=0;i<point1;++i)
        for (int s=0;s<point1-1;++s)
        if(point_x[s] ==-100000000)
        {
            temp =point_x[s];temp1 =point_y[s];
            point_x[s]=point_x[s+1];point_y[s]=point_y[s+1];
            point_x[s+1]=temp;point_y[s+1]=temp1;
        }

        int new_points=0;
        for (int s1=0;s1<point1;++s1)
        if(point_x[s1] !=-100000000)new_points=new_points+1;
        point1=new_points;
        st=0;
        ///////////////////////////////////////////////////
        goto skip_parameter;

    }
}

```



```

st=index;
skip_parameter;;
end1=point1-1;
    } while ( st!=end1);

    ////////////sort
        for (int i=0;i<point1;++i)
    for (int s=0;s<point1-1;++s)
        if(point_x[s] ==-100000000)
        {
            temp =point_x[s];temp1 =point_y[s];
            point_x[s]=point_x[s+1];point_y[s]=point_y[s+1];
            point_x[s+1]=temp;point_y[s+1]=temp1;
        }

    ////////////

        for(int i=0;i<point;++i)//sort the points
    for (int s=0;s<point-1;++s)
        if(point_x[s] ==-100000000)
        {
            temp =point_x[s];temp1 =point_y[s];
            point_x[s]=point_x[s+1];point_y[s]=point_y[s+1];
            point_x[s+1]=temp;point_y[s+1]=temp1;
        }
    }
    ////////////calculate the number of real points
        int new_points=0;
        for (int s1=0;s1<point;++s1)
            if(point_x[s1] !=-100000000)new_points=new_points+1;
        point=new_points;

    ////////////check angle filter and eliminate the angles which is less than 25
    ////////////calculate the angles between points of the new profiles
    double edg_a,edg_b,edg_c,ang[1000];;
    edg_a=sqrt((float)(( point_x[1]-point_x[0])*( point_x[1] -point_x[0])) + ((
    point_y[1]-point_y[0])*( point_y[1] -point_y[0])));
    edg_b=sqrt((float)(( point_x[0]-point_x[point-1])*( point_x[0] -point_x[point-1])) +
    (( point_y[0]-point_y[point-1])*( point_y[0] -point_y[point-1])));
    edg_c=sqrt((float)(( point_x[1]-point_x[point-1])*( point_x[1] -point_x[point-1])) +
    (( point_y[1]-point_y[point-1])*( point_y[1] -point_y[point-1])));

    ang[0]=acos((edg_a*edg_a+edg_b*edg_b-edg_c*edg_c)/(2*edg_a*edg_b) ) * 180 / pi;// angle
    in degrees

    edg_a=sqrt((float)(( point_x[0]-point_x[point-1])*( point_x[0] -point_x[point-1])) +
    (( point_y[0]-point_y[point-1])*( point_y[0] -point_y[point-1])));
    edg_b=sqrt((float)(( point_x[point-2]-point_x[point-1])*( point_x[point-2]
    -point_x[point-1])) + (( point_y[point-2]-point_y[point-1])*( point_y[point-2]
    -point_y[point-1])));
    edg_c=sqrt((float)(( point_x[0]-point_x[point-2])*( point_x[0] -point_x[point-2])) +
    (( point_y[0]-point_y[point-2])*( point_y[0] -point_y[point-2])));

    ang[point-1]=acos((edg_a*edg_a+edg_b*edg_b-edg_c*edg_c)/(2*edg_a*edg_b) ) * 180 /
    pi;// angle in degrees

    for (int s=1;s<point-1;++s)
    {

    edg_a=sqrt((float)(( point_x[s]-point_x[s-1] ) * ( point_x[s] -point_x[s-1])) + ((
    point_y[s]-point_y[s-1])*( point_y[s] -point_y[s-1])));
    edg_b=sqrt((float)(( point_x[s]-point_x[s+1] ) * (point_x[s] -point_x[s+1])) + ((
    point_y[s]-point_y[s+1])*( point_y[s] -point_y[s+1])));
    edg_c=sqrt((float)(( point_x[s-1]-point_x[s+1]) * ( point_x[s-1] -point_x[s+1])) + ((
    point_y[s-1]-point_y[s+1])*( point_y[s-1] -point_y[s+1])));

```

```

    ang[s]=acos((edg_a*edg_a+edg_b*edg_b-edg_c*edg_c)/(2*edg_a*edg_b) ) * 180 / pi;//
    angle in degrees

}
//////////
for (int s=0;s<point;++s)
{
    if (ang[s]<=45)point_x[s]=-100000000;
}
//////////

for(int i=0;i<point;++i)//sort the points
for (int s=0;s<point-1;++s)
if(point_x[s] ==-100000000)
{
    temp =point_x[s];temp1 =point_y[s];
    point_x[s]=point_x[s+1];point_y[s]=point_y[s+1];
    point_x[s+1]=temp;point_y[s+1]=temp1;
}
}
//////////End doglus approximate //////////
//////////calculate the the boundary of the object using mathematical morphology
void boundary1(int r[a][b] , int (&boundary)[a][b] )
{int r_temp[a][b];
for(int k=0;k<a;k++)
    for(int j=0;j<b;j++)
        r_temp[k][j]=0;

for(int k=0;k<a;k++)
{   r_temp[k][0]=0; r_temp[k][b-1]=0;}
for(int j=0;j<b;j++)
{   r_temp[0][j]=0; r_temp[a-1][j]=0;}

//////////square structure
for(int k=1;k<a-1;k++)
    for(int j=1;j<b-1;j++)
    {   if (r[k][j]==r[k-1][j] && r[k][j]==r[k+1][j] && r[k][j]==r[k][j-1] &&
r[k][j]==r[k][j+1] &&
        r[k][j]==r[k-1][j-1]&& r[k][j]==r[k-1][j+1]&& r[k][j]==r[k+1][j-1]&&
r[k][j]==r[k+1][j+1] )//first upper left corner calculator
        r_temp[k][j]= r[k][j];
        else r_temp[k][j]=0;    }

    for(int k=0;k<a;k++)
        for(int j=0;j<b;j++)
            boundary[k][j]=r[k][j]- r_temp[k][j];
}
////////// End calculate the dilation
////////// detection of the buildings to locate the limits of the objects
int boun(int loop, int& min_x,int& min_y, int& max_x, int& max_y ,int& area, int
label[a][b])
{
    for (int k=0;k<a;++k)
    for (int j=0;j<b;++j)
        if (label[k][j]==1)
        {area=area+1;if (k>=max_y)max_y=k;if (k<=min_y)min_y=k;
if (j>=max_x)max_x=j;if (j<=min_x)min_x=j;}
    return 0;
}
//////////detection of the buildings to locate the limits of the objects
//////////End of the function part//////////

```

The result of the processing the code will be an ASCII file that contains five columns, and listed as follow: first column is for the sequence of the building boundary nodes; the second column is for the building boundary label; the third, fourth and fifth indicates the coordinates with respect to the UTM for each corner individually.

```

File Edit Format View Help
0 734 420038.50 6194422.00 124.71
1 734 420040.06 6194417.50 124.71
2 734 420050.66 6194421.00 124.71
3 734 420048.84 6194426.00 124.71
0 959 420022.16 6194422.50 119.38
1 959 420023.41 6194415.00 119.38
2 959 420033.81 6194416.50 119.38
3 959 420032.50 6194424.50 119.38
0 12012 419990.06 6194420.00 123.71
1 12012 419990.31 6194414.00 123.71
2 12012 420001.16 6194415.00 123.71
3 12012 420001.06 6194420.00 123.71
4 12012 419994.88 6194420.00 123.71
0 1298 419974.50 6194417.50 123.73
1 1298 419975.72 6194412.00 123.73
2 1298 419985.25 6194414.50 123.73
3 1298 419983.66 6194419.50 123.73
4 1298 419979.59 6194418.50 123.73
0 13012 420005.38 6194421.50 124.48
1 13012 420006.56 6194413.00 124.48
2 13012 420018.69 6194415.00 124.48
3 13012 420017.72 6194423.00 124.48
4 13012 420009.31 6194422.00 124.48
0 13811 419954.38 6194415.00 123.65
1 13811 419955.81 6194408.50 123.65

```

Figure C.1 A screen shot of the output of building footprint and 3D modelling is shown in text format.

Appendix E- Building footprint measured data

Table E-1 Corner planimetric discrepancies of true and extracted buildings.						
Building #	planimetric corners(m)					
1	1.59	1.61	0.79	1.43		
2	3.29	1.17	3.57	1.35		
3	2.49	0.12	2.24	2.70		
4	0.94	1.37	0.76	0.26		
5	1.60	1.35	3.35	3.32		
6	1.24	1.76	1.44	0.90		
7	1.38	0.87	1.61	1.44		
8	1.44	1.09	0.45	1.90		
9	0.89	2.10	2.73	2.15		
10	1.67	1.68	2.23	3.76		
11	4.25	5.83	9.20	7.19		
12	2.39	2.75	2.99	5.85		
13	1.96	1.95	2.16	1.57		
14	0.75	2.02	2.47	1.11		
15	2.57	2.07	0.14	1.96		
16	1.10	2.07	2.26	1.24		
17	1.59	0.71	2.81	0.59		
18	1.87	0.36	2.36	0.74		
19	3.41	1.62	2.75	2.81		
20	2.48	2.69	1.94	7.73		
21	0.80	2.44	5.50	2.95		
22	9.65	8.21	2.59	1.84		
23	1.07	1.78	1.10	2.00		
24	0.29	2.38	3.40	1.73		
25	1.82	2.00	1.64	2.19		
26	2.87	3.18	0.87	3.91	4.34	1.88
27	3.81	2.48	4.07	4.25	3.09	4.69
28	1.52	2.64	2.80	1.45		
29	0.20	1.94	1.58	2.20		
30	0.79	1.32	2.34	1.34		
31	1.39	0.99	2.92	1.96		
32	1.80	0.98	0.50	0.61		
33	0.49	2.18	2.44	2.33		
34	2.08	3.42	0.79	3.94		
35	1.39	1.49	1.38	1.00		
Min(m)	0.12					
Max(m)	9.65					
Mean(m)	2.23					
σ (m)	1.62					
RMSE(m)	2.88					

Table E-2 Corner planimetric discrepancies in X and Y direction of true and extracted buildings													
Building #	X corners_residual						Y corners_residual						
1	-0.27	1.42	0.26	-0.96			1.57	0.75	0.75	1.06			
2	-2.85	-0.32	2.14	0.79			-1.64	1.12	2.86	1.09			
3	1.91	0.01	-0.91	0.95			1.60	0.12	2.05	2.53			
4	-0.10	0.95	0.76	0.22			0.94	0.99	0.09	0.14			
5	1.59	0.20	0.94	2.48			-0.18	1.33	3.22	2.21			
6	1.02	1.74	1.21	0.45			0.71	0.23	0.79	0.77			
7	1.33	0.75	1.22	1.36			0.38	0.45	1.05	0.48			
8	1.04	-0.42	0.29	1.78			0.99	1.01	0.34	0.68			
9	0.88	1.54	2.40	2.14			-0.21	1.42	1.31	-0.19			
10	-0.14	1.62	0.14	-1.71			1.67	0.47	2.23	3.34			
11	-1.45	-4.76	3.19	7.09			-4.00	3.36	8.63	1.19			
12	-0.43	2.83	0.34	2.89			2.35	0.75	2.98	5.09			
13	0.60	1.83	0.60	0.14			1.87	0.66	1.45	2.15			
14	-0.58	0.46	1.50	0.67			0.47	1.96	1.97	0.89			
15	1.02	1.68	0.11	-0.85			2.36	1.21	0.09	1.77			
16	0.01	1.15	1.50	0.71			1.10	1.73	1.69	1.01			
17	-1.30	0.38	1.53	0.19			0.91	0.59	2.55	0.56			
18	-1.47	0.34	2.19	0.62			1.15	0.12	0.87	-0.40			
19	-3.40	0.35	-1.52	0.72			-0.27	1.58	2.65	2.37			
20	2.44	0.12	0.14	0.47			0.48	2.69	1.91	7.18			
21	-0.67	-0.21	1.04	0.51			0.43	2.43	5.40	2.91			
22	1.85	7.27	2.49	1.17			-9.47	-3.82	0.73	-1.42			
23	0.79	0.20	1.05	1.61			0.71	1.76	0.34	-1.20			
24	-0.03	0.80	0.83	1.51			0.29	2.24	3.30	0.84			
25	0.06	1.39	0.98	-0.08			1.82	1.44	1.32	2.19			
26	-1.66	-1.47	-0.02	2.22	2.56	-1.24	2.33	2.82	0.87	3.22	3.51	1.41	
27	2.07	1.95	2.17	3.46	2.98	-1.69	3.20	1.54	3.95	2.19	0.82	4.37	
28	0.95	2.51	2.45	0.58			1.19	0.81	1.36	1.33			
29	0.08	0.32	1.22	1.85			0.19	1.91	1.00	-1.19			
30	0.42	1.30	1.84	0.81			0.67	-0.25	1.44	1.07			
31	0.24	0.48	0.11	0.36			1.37	0.87	3.60	1.92			
32	0.83	0.86	0.18	0.59			1.60	0.47	0.47	0.15			
33	0.28	0.28	1.66	2.29			0.40	2.16	2.22	0.47			
34	1.00	3.16	-0.20	-1.77			1.83	-1.29	0.77	3.52			
35	-0.08	0.90	0.69	-0.10			1.38	0.90	1.19	1.00			
Min(m)	-4.76						-9.47						
Max(m)	7.27						8.63						
Mean(m)	0.76						1.29						
σ (m)	1.49						1.79						
RMSE(m)	1.67						2.20						

Table E-3 Centre planimetric discrepancies in X and Y direction of true and extracted buildings.

Building #	Planemtaric centre residual		X centre residual		Y centre residual
1	1.05		0.21		1.03
2	1.05		-0.08		1.05
3	1.67		0.38		1.62
4	0.64		0.44		0.46
5	2.12		1.35		1.64
6	1.18		1.05		0.55
7	1.26		1.11		0.60
8	0.7		0.60		0.36
9	1.85		1.71		0.70
10	1.87		-0.02		1.87
11	2.47		0.96		2.28
12	2.77		-0.03		2.77
13	1.75		0.68		1.62
14	1.46		0.53		1.36
15	1.42		0.53		1.32
16	1.67		0.83		1.45
17	0.64		0.07		0.63
18	0.35		0.27		-0.22
19	2.09		-0.96		1.85
20	3.18		0.31		3.16
21	2.77		0.13		2.76
22	4.05		0.51		4.01
23	0.91		0.84		0.35
24	1.67		0.34		1.63
25	1.89		0.70		1.75
26	3.15		1.06		2.96
27	2.82		2.15		1.82
28	1.92		1.56		1.12
29	0.8		0.65		0.47
30	1.23		1.14		0.47
31	1.74		-0.20		1.73
32	0.71		0.53		0.47
33	1.61		0.99		1.26
34	1.36		0.53		1.26
35	0.36		0.36		0.06
Min(m)	0.35		-0.96		-0.22
Max(m)	4.05		2.15		4.01
Mean(m)	1.66		0.61		1.38
σ (m)	0.87		0.59		0.95
RMSE(m)f	1.84		0.84		1.64

Table E-4 measured area of the true and extracted buildings.						
Building #	area			common area (TP)	wrongly extracted (FN)	missed area (FP)
	true area	extracted area	Diff.			
1	89.6546	103.675	14.0204	81.56	22.115	8.0946
2	152.7253	179.385	26.6597	137.332	42.053	15.3933
3	88.811	60.715	-28.096	53.3	7.415	35.511
4	78.409	96.23	17.821	77.53	18.7	0.879
5	87.9964	58.7975	-29.1989	49.72	9.0775	38.2764
6	90.762	91.4175	0.6555	77.12	14.2975	13.642
7	80.1817	75.615	-4.5667	63.9	11.715	16.2817
8	88.0907	87.4975	-0.5932	75.52	11.9775	12.5707
9	88.1016	98.3925	10.2909	73.72	24.6725	14.3816
10	117.4319	116.2375	-1.1944	90.54	25.6975	26.8919
11	117.4993	133.07	15.5707	89.33	43.74	28.1693
12	117.4085	121.6525	4.244	78.94	42.7125	38.4685
13	79.657	84.71	5.053	62.47	22.24	17.187
14	88.7595	100.6875	11.928	76.53	24.1575	12.2295
15	88.741	103.4025	14.6615	78.45	24.9525	10.291
16	89.3722	102.3775	13.0053	74.79	27.5875	14.5822
17	77.1729	80.835	3.6621	64.79	16.045	12.3829
18	152.9226	177.705	24.7824	147.77	29.935	5.1526
19	153.2231	194.3725	41.1494	132.39	61.9825	20.8331
20	153.5556	169.585	16.0294	115.29	54.295	38.2656
21	77.2712	65.33	-11.9412	45.57	19.76	31.7012
22	774.9193	564.2625	-210.657	536.47	27.7925	238.4493
22a		18.3	18.3	18.3	0	0
23	154.4471	183.83	29.3829	145.19	38.64	9.2571
24	80.6323	76.285	-4.3473	59.33	16.955	21.3023
25	411.729	421.7075	9.9785	353.39	68.3175	58.339
26	646.312	800.815	154.503	575.11	225.705	71.202
27	590.3626	680.01	89.6474	526.19	153.82	64.1726
28	258.2512	266.0175	7.7663	223.05	42.9675	35.2012
29	90.6505	107.2825	16.632	83.71	23.5725	6.9405
30	125.553	123.4775	-2.0755	103.7	19.7775	21.853
31	117.8535	106.935	-10.9185	88.76	18.175	29.0935
32	185.4073	204.3525	18.9452	177.48	26.8725	7.9273
33	103.7491	105.7975	2.0484	83.42	22.3775	20.3291
34	124.5665	124.41	-0.1565	100.81	23.6	23.7565
35	90.731	124.3625	33.6315	90.62	33.7425	0.111
total	5912.9115	6209.535	296.6235	4912.092	1297.443	1019.119

Table E-5 Measured height of the true and extracted buildings of hipped buildings with hipped and one direction sloped roofs.						
Building #	WGS-Ele v.	WV-1 DSM	Pleiades DSM	Maximum Likelihood Merging DSM	Bayesian Merging DSM range $\pm 0.1\text{m}$	Bayesian Merging DSM range $\pm 0.25\text{m}$
1	118.16	113.38	115.19	113.70	113.66	113.63
2	117.34	115.02	118.33	114.89	114.84	114.76
3	118.09	114.70	118.09	115.26	115.23	115.15
4	118.15	114.08	116.86	114.56	114.53	114.46
5	118.61	115.70	118.37	115.92	115.99	116.03
6	118.66	112.86	118.19	113.64	113.63	113.65
7	118.55	112.63	116.64	112.78	112.78	112.81
8	118.64	115.16	118.06	115.04	114.99	114.94
9	118.00	116.92	117.59	116.66	116.72	116.77
10	118.59	113.10	120.43	114.11	114.11	114.12
11	118.49	113.06	119.66	114.06	114.06	114.07
12	118.37	118.66	119.13	118.39	118.39	118.37
13	116.54	114.22	114.67	113.98	113.93	113.88
14	117.71	115.77	116.95	115.95	115.95	115.97
15	118.25	117.58	116.26	116.69	116.69	116.62
16	118.43	117.06	116.36	116.16	116.10	116.05
17	117.92	117.45	116.05	116.52	116.45	116.40
18	115.83	113.74	114.47	113.55	113.50	113.47
19	117.85	117.34	119.71	117.99	117.95	117.92
20	117.73	116.92	116.15	115.87	115.81	115.76
21	117.64	116.04	115.57	115.47	115.42	115.37
22	114.24	114.24	115.26	114.42	114.41	114.44
23	115.62	114.76	116.02	114.82	114.77	114.71
24	115.83	113.95	115.01	114.11	114.07	113.99
25	119.80	119.26	122.60	119.38	119.38	119.42
26	125.86	124.06	124.57	124.16	124.16	124.19
27	122.99	124.35	124.98	124.32	124.27	124.19
28	116.23	116.14	118.52	116.57	116.53	116.55
29	116.64	116.81	120.33	117.39	117.37	117.35
30	118.88	117.77	118.36	117.87	117.84	117.88
31	118.88	117.68	121.42	118.28	118.26	118.28
32	118.89	118.82	118.74	117.73	117.67	117.67
33	118.89	118.73	119.81	117.82	117.78	117.79
34	119.14	118.60	120.71	118.87	118.83	118.81
35	118.85	117.92	120.84	118.40	118.37	118.37
Min. d(m)		-1.36	-3.69	-1.33	-1.28	-1.20
Max. d(m)		5.92	2.97	5.77	5.77	5.74
Mean d(m)		1.82	-0.05	1.80	1.82	1.84
σ of d (m)		1.90	1.69	1.71	1.71	1.71
RMSE using discrepancy(m)		2.62	1.67	2.47	2.48	2.49

Table E-6 Measured height of the true and extracted buildings of hipped roofs.						
Building #	WGS-Ele v.	WV-1 DSM	Pleiades DSM	Maximum Likelihood Merging DSM	Bayesian Merging DSM range $\pm 0.1\text{m}$	Bayesian Merging DSM range $\pm 0.25\text{m}$
1	118.16	113.38	115.19	113.70	113.66	113.63
2	117.34	115.02	118.33	114.89	114.84	114.76
3	118.09	114.70	118.09	115.26	115.23	115.15
4	118.15	114.08	116.86	114.56	114.53	114.46
5	118.61	115.70	118.37	115.92	115.99	116.03
6	118.66	112.86	118.19	113.64	113.63	113.65
7	118.55	112.63	116.64	112.78	112.78	112.81
8	118.64	115.16	118.06	115.04	114.99	114.94
9	118.00	116.92	117.59	116.66	116.72	116.77
10	118.59	113.10	120.43	114.11	114.11	114.12
11	118.49	113.06	119.66	114.06	114.06	114.07
12	118.37	118.66	119.13	118.39	118.39	118.37
13	116.54	114.22	114.67	113.98	113.93	113.88
14	117.71	115.77	116.95	115.95	115.95	115.97
15	118.25	117.58	116.26	116.69	116.69	116.62
16	118.43	117.06	116.36	116.16	116.10	116.05
17	117.92	117.45	116.05	116.52	116.45	116.40
18	115.83	113.74	114.47	113.55	113.50	113.47
19	117.85	117.34	119.71	117.99	117.95	117.92
20	117.73	116.92	116.15	115.87	115.81	115.76
21	117.64	116.04	115.57	115.47	115.42	115.37
23	115.62	114.76	116.02	114.82	114.77	114.71
24	115.83	113.95	115.01	114.11	114.07	113.99
25	119.80	119.26	122.60	119.38	119.38	119.42
26	125.86	124.06	124.57	124.16	124.16	124.19
27	122.99	124.35	124.98	124.32	124.27	124.19
Min. d(m)		-1.36	-2.80	-1.33	-1.28	-1.20
Max. d(m)		5.92	2.97	5.77	5.77	5.74
Mean d(m)		2.30	0.45	2.30	2.32	2.34
σ of d (m)		1.98	1.53	1.69	1.68	1.67
RMSE using discrepancy(m)		3.01	1.56	2.83	2.84	2.86

Table E-7 Measured height of the true and extracted buildings of one direction slope roofs.						
Building #	WGS-Ele v.	WV-1 DSM	Pleiades DSM	Maximum Likelihood Merging DSM	Bayesian Merging DSM range $\pm 0.1\text{m}$	Bayesian Merging DSM range $\pm 0.25\text{m}$
22	114.24	114.24	115.26	114.42	114.41	114.44
28	116.23	116.14	118.52	116.57	116.53	116.55
29	116.64	116.81	120.33	117.39	117.37	117.35
30	118.88	117.77	118.36	117.87	117.84	117.88
31	118.88	117.68	121.42	118.28	118.26	118.28
32	118.89	118.82	118.74	117.73	117.67	117.67
33	118.89	118.73	119.81	117.82	117.78	117.79
34	119.14	118.60	120.71	118.87	118.83	118.81
35	118.85	117.92	120.84	118.40	118.37	118.37
Min. d(m)		-0.17	-3.69	-0.75	-0.73	-0.71
Max. d(m)		1.20	0.52	1.16	1.22	1.22
Mean d(m)		0.44	-1.48	0.36	0.40	0.39
σ of d (m)		0.52	1.33	0.68	0.68	0.68
RMSE using discrepancy(m)		0.66	1.94	0.73	0.76	0.75

