DGK Deutsche Geodätische Kommission
bei der Bayerischen Akademie der Wissenschaften

Reihe C | Dissertationen | Heft Nr. 648

Juilson José Jubanski

Monoplotting through Fusion of LIDAR Data and Low-Cost Digital Aerial Imagery

München 2010

Verlag der Bayerischen Akademie der Wissenschaften
in Kommission beim Verlag C. H. Beck

ISSN 0065-5325 ISBN 978-3-7696-5060-0

DGK Deutsche Geodätische Kommission
bei der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 648

Monoplotting through Fusion of LIDAR Data and Low-Cost Digital Aerial Imagery

Zur Erlangung des akademischen Grades eines
DOKTOR-INGENIEURS
von der Fakultät für
Bauingenieur- Geo- und Umweltwissenschaften
des
KIT - Universität des Landes Baden-Württemberg und
nationales Forschungszentrum in der Helmholtz-Gemeinschaft
genehmigte
DISSERTATION

von

M.Sc. Juilson José Jubanski

aus Curitiba – Brasilien

München 2010

Verlag der Bayerischen Akademie der Wissenschaften
in Kommission beim Verlag C. H. Beck

ISSN 0065-5325 ISBN 978-3-7696-5060-0

Adresse der Deutschen Geodätischen Kommission:

Deutsche Geodätische Kommission

Alfons-Goppel-Straße 11 ● D – 80 539 München

Telefon +49 – 89 – 23 031 1113 ● Telefax +49 – 89 – 23 031 - 1283/ - 1100

e-mail hornik@dgfi.badw.de ● http://www.dgk.badw.de

Hauptreferent: Prof. Dr.-Ing. habil. Dr. h.c. Hans-Peter Bähr

1. Korreferent: Prof. Dr.-Ing. Stefan Hinz

2. Korreferent: Prof. Dr. Edson Aparecido Mitishita

Tag der mündlichen Prüfung: 10.02.2010

ISSN 0065-5325 ISBN 978-3-7696-5060-0

Contents

List of Figures

List of Tables

Abstract

Photogrammetry is rapidly adapting itself to new digital technologies in order to become cheaper, more efficient and more competitive. This work proposes a semiautomatic monoplotting methodology based on LIDAR data and low-cost digital imagery. These are two complementary data sources of modern Photogrammetry. Monoplotting consists of the digitizing of interest elements from aerial imagery and their rectification using a digital surface model as altimetric data source. Precise altimetry data obtained with LIDAR and very good spatial resolution of photogrammetric digital imagery can be integrated in order to produce high quality urban maps. The importance of a well mapped urban environment is the possibility to make the planning of public and private actions viable. In the last two decades, advances in computer technology, earth observation sensors and GIS science led to the development of *Object-oriented Image Analysis*. A primary step for any object-oriented image processing is the *image segmentation*. This work presents a hybrid solution to image segmentation, filtering the image and finding seeds with the mean-shift procedure, and then growing segments with seeded region growing. Results show similar to expensive segmentation software. The most remarkable drawback is the difficulty to obtain good results in areas with shadows. LIDAR data is composed, basically, of huge amounts of 3D points, called *point clouds*. This work proposes a GIS based storage of the LIDAR data in the PostGres/PostGIS database environment. The performance shows very good, in the order of few milliseconds to find a point among about 14 million. This integration of LIDAR data into a GIS environment shows also very helpful during data fusion. This work proposes five stages for LIDAR processing: the *filtering*, which detects bare ground information; the *DSM generation*, which rasterizes the point cloud; the *segmentation*, which separates interest elements from bare ground; the *splitting*, which splits the segments into rough and smooth segments; and finally the *classification*, which classifies the segments of the split image into vegetation and edification. Buildings are, probably, the most important objects in an urban environment. This work proposes a building footprint detection through combination of LIDAR data and aerial imagery as a case study of the proposed data fusion methodology. A data fusion methodology in two steps is proposed. First, spectral data from the aerial imagery is integrated into the segmented DSM though orthorectification in order to permit a scene description for the classification. Then, the detected buildings in the DSM are projected into image space in order to detect the roofs in the aerial imagery. At last, the detected roofs are projected into object space. The methodology shows good results if data conditions are favorable: 90% in the edification classification and 82% of the polygon vertices with a precision better than 1m are correctly detected.

Zusammenfassung

Photogrammetrie stellt sich heutzutage rasch auf neue digitale Technologien ein, um kostengüstiger, effizienter und wettbewerbsfähiger zu werden. Diese Arbeit schlägt eine semi-automatische Monoauswertungs-Methodik für LIDAR-Daten und Low-Cost Digitalkameras vor. Dies sind zwei ergänzende Datenquellen der modernen Photogrammetrie. Monoauswertung besteht aus der Digitalisierung von Elementen aus Luftbildern sowie deren Entzerrung mittels eines digitalen Oberflächenmodelles als altimetrische Datenquelle. Genaue LIDAR Höhendaten und digitale photogrammetrische Bilder, sehr hoher räumlicher Auflösung, können integriert werden, um städtische Karten von hoher Qualität zu erzeugen. Die Bedeutung einer gut kartierten städtischen Umgebung besteht in der Möglichkeit, Planung von öffentlichen und privaten Initiativen durchführbar zu machen. Fortschritte in der Computertechnologie, Erdbeobachtung und GIS-Wissenschaften in den letzten zwei Jahrzehnten haben zur Entwicklung von *Objektorientierter Bildanalyse* geführt. Ein grundlegender Schritt für objektorientierte Bildverarbeitung ist die *Bildsegmentierung*. Diese Arbeit stellt eine hybride Bildsegmentierungslösung durch Filterung des Bildes und Seed Durchsuchung mit dem Mean-shift Verfahren und Segmentwachstum mit Seeded Region Growing vor. Die erzielten Ergebnisse ähneln den Ergebnissen kommerzieller Segmentierungsoftware. Der bemerkenswerteste Nachteil ist die Schwierigkeit, gute Ergebnisse in Bereichen mit Schatten zu erreichen. Da LIDAR-Daten aus großen Mengen von 3D-Punkten, den so genannten *Punktwolken*, bestehen, schlägt diese Arbeit eine GIS-basierte Speicherung der LIDAR-Daten in der Postgres/PostGIS Datenbank-Umgebung vor. Die beobachtete Leistung ist sehr gut. In der Größenordnung von wenigen Millisekunden kann ein Punkt unter etwa 14 Millionen Punkten gefunden werden. Die Integration von LIDAR Daten in eine GIS Umgebung ist auch sehr hilfreich bei der Datenfusion. Diese Arbeit schlägt fünf Schritte für die LIDAR Verarbeitung vor: die *Filterung*, zur Erkennung des Erdbodens; die *DSM Generation*, zur Rasterung der Punktwolke; die *Segmentierung*, zur Trennung von Objekten und Erdboden, das *Splitting*, zur Zerlegung von Objekten in raue und glatte Segmente; und schließlich die *Segmentklassifizierung*, zur Unterscheidung in Vegetation und Gebäuden. Gebäude sind wichtige Objekte in einer städtischen Umgebung. Daher werden in einer Fallstudie die entwickelte Datenfusionsmethodik zur Erkennung des Gebäudegrundrisses durch Kombination von LIDAR und Luftbildern getestet. Zuerst werden dabei die spektralen Daten aus den Luftbildern durch Orthorektifikation in die segmentierten Entfernungsbilder integriert, um eine bessere Szenebeschreibung zu ermöglichen. Dannach werden die erkannten Gebäude in den Bildraum projiziert, um die Dächer im Luftbild zu erkennen. Abschließend werden die erkannten Dächer zurück in den Objektraum projiziert. Die Methode ergibt gute Ergebnisse, wenn gute Daten vorhanden sind: ca. 90 % der Gebäude wurden korrekt klassifiziert und ca. 82 % der Gebäudekanten wurden mit einer Genauigkeit besser als 1m indentifiziert.

Resumo

A Fotogrametria está rapidamente se adaptando às novas tecnologias digitais de forma a se tornar mais barata, eficiente e competitiva. Este trabalho porpõe uma metodologia semi-automática de monorrestituição digital baseada em dados LIDAR e imagens aéreas digitais obtidas com câmeras de baixo custo. Estas são duas fontes de dados complementares na Fotogrametria moderna. Monorrestituição consiste na digitalização de feições de interesse a partir de uma aerofoto e sua retificação, utilizando como base de dados altimétrica um modelo digital de superfície. Dados altimétricos de alta precisão obtidos com LIDAR e a excelente resolução espacial oferecida por aerofotos digitais podem ser integradas no intento de se produzir mapas urbanos de alta qualidade. A importância de um ambiente urbano bem mapeado é a viabilização de projetos de planejamento públicos e privados. Nas últimas duas décadas, avanços nas ciências de computação, sensores imageadores e GIS levaram ao desenvolvimento da *Análise de Imagens Orientada a Objetos*. Uma etapa fundamental para o processamento digital de imagens orientada a objeto é a *Segmentação*. Este trabalho apresenta uma solução hibrida para segmentação de imagens, realizando uma filtragem e encontrando sementes com o algoritmo do deslocamento pela média e então realizando um crescimento de regiões a partir dessas sementes. Resultados similares a programas comerciais de segmentação foram obtidos. O maior problema apresentado foi nas regiões com sombras. Dados LIDAR consistem basicamente de grandes quantidades de pontos 3D, formando as chamadas *nuvens de pontos*. Este trabalho propõe um armazenanamento dessas nuvens de pontos baseado em GIS, utilizando-se do ambiente de banco de dados PostGres/PostGIS. A performance se mostrou muito boa, com tempos de busca na ordem de poucos milisegundos para um ponto entre aproximadamente 14 milhões. Essa integração de dados LIDAR num ambiente GIS se mostrou também muito interessante durante a fase da fusão dos dados. Este trabalho propõe cinco fases para o processamento dos dados LIDAR: a *filtragem*, que consiste na detecção de pontos ao nível do solo; a *geração da imagem laser*, onde a nuvem de pontos é rasterizada; a *segemtação da imagem laser*, onde as regiões altas são detectadas; a *separação*, que consiste em dividir os segmentos em regiões lisas e rugosas; e por fim a *classificação* que consiste em classificar os segmentos entre edificação e vegetação. Alguns problemas ainda ocorrem durante a etapa da flitragem. Edificações são, provavelmente, os objetos mais importantes da paisagem urbana. Este trabalho propõe uma metodologia para detecção de edificações através da fusão de dados LIDAR com imagens obtidas com câmeras de baixo custo como um estudo de caso da metodologia proposta. Esta metodologia se dá em duas etapas. Primeiramente, dados espectrais das aerofotos são levados à imagem LIDAR através de um procedimento de ortoretificação, de forma a se obter mais informações para a classificação. O proximo passo é a projeção dos edificios detectados para o espaço imagem e sua deteção na aerofoto segementada. Por último, os edificios detectados são reprojetados para o espaço objeto. A metodologia apresenta bons resultados com dados de qualidade: cerca de 90% das edificações são corretamente detectadas e 82% dos seus vértices são detectados com uma precisão melhor que 1m.

Introduction

Photogrammetry is rapidly adapting itself to new digital technologies in order to become cheaper, more efficient and more competitive. One example is the progress in LIDAR technology, which created new possibilities for the photogrammetric community. Another development concerns the use of images taken with low-cost digital cameras. The integration of these two methodologies is a promising research field. This work proposes a methodology for integration of LIDAR data with aerial images obtained with low-cost digital cameras. Because of the lack of updated mapping in many urban zones in developing countries - like Brazil - the fusion of those methods can become a solution to the enormous delay in urban cadastre. This work proposes methodologies for urban areas mapping, therefore it is important to define the term *city*. Bauer (2009) defines "city" as a complex arrangement of land uses, linked together by circulation systems and made viable by utilities systems as sewerage, water supply, electric power and telecommunications. The importance of a well mapped urban environment is the possibility to make the planning of public and private actions viable, which can help with the socioeconomic development of those spaces. *Urban planning* is a rational process that seeks the orderly, cost effective development of the urban environment. Furthermore, the orderly physical development of urban areas is the public interest, and public planning should be oriented to furthering public health, safety and general welfare. Maps provide the graphic representation of the planning area and relate pertinent planning data to geographic location (Bauer, 2009).

Aerial photographs have excellent spatial resolution and a well defined projective geometry. For these reasons, urban mapping is a traditional stereo photogrammetric task. However, stereo plotting is the slowest and most expensive step in a photogrammetric project. This is because highly specialized operators are needed, and also because of the complexity of the information contained in a stereo model, particularly in dense urbanized areas. Monoplotting consists of the digitizing of interest elements from photographic images and their rectification using a digital surface model (DSM) as altimetric data source (Makarovic, 1973). Using this technology, the need of highly specialized operators is avoided (Jauregui *et al.*, 2002; Mitishita, 1997). Monoplotting uses the inverse collinearity equations (see eq. 3.8) for the determination of planimetric coordinates of interest elements in a geodetic reference system; in function of the elements photogrammetric coordinates, its elevation, and the photograph's exterior orientation parameters. Nowadays, however, the available monoplotting systems depend on the hand-digitizing of the interest elements. Such issue must be solved by implementing automatic processes. The lack of altimetric data (DSM), necessary for the monoplotting process, used to be a problem, but, with the advent of LIDAR, elevation data became abundant and accurate. This work proposes a semiautomatic monoplotting methodology based on LIDAR data and low-cost digital imagery,"two complimentary data sources of modern Photogrammetry" (Schenk & Csathó, 2002).

In the last two decades, advances in computer technology, earth observation sensors and GIS science, led to the development of *Object-oriented Image Analysis*, whose main proposal is to use image context for its classification as an alternative to the traditional pixel-based approach (Gao & Mas, 2008). Traditional pixel-based image analysis is limited because of the following reasons: image pixels are not true geographical objects and the pixel topology is limited; pixel based image analysis largely neglects the spatial photo-interpretive elements such as texture, context, and shape; the increased variability implicit within high spatial resolution imagery confuses traditional pixel-based classifiers resulting in lower classification accuracies (Hay & Castilla, 2006). Object-oriented Image Analysis works on objects instead of single pixels. The idea to classify objects stems from the fact that most image data exhibit characteristic texture features which are neglected in conventional classifications (Blaschke & Strobl, 2001). The concept of *object* plays a central role in image interpretation. However, the determination of what constitutes an object is extremely difficult. A basic step for any object-oriented image processing is *image segmentation*. The major challenge to segmentation of the object-oriented pixel patches whose shapes resemble the shapes of "real-world" objects, is the high variability of relationships between the object and image context (background). While there has been considerable effort in the development of image segmentation, this problem remains a great challenge for computer vision (Skurikhin & Volegov, 2008). Comaniciu & Meer (2002) propose a segmentation technique based on the *mean-shift* procedure, which was first presented by Fukanaga & Hostetler (1975). A popular solution for image segmentation is the *seeded region growing* (SRG), first proposed by Adams & Bischof (1994). This work presents an alternative solution to image segmentation.

LIDAR data is composed, basically, of huge amounts of 3D points, called *point clouds*. Due to difficulties encountered to deal with point clouds, there is an emerging demand for automated data processing. The growing number of engineering applications using altimetric data shows that it is worth developing LIDAR data processing algorithms (Tovari, 2006). For example, knowledge of the geometric character of the bare ground is essential to many civil design and planning applications. This knowledge is acquired by building digital terrain models (DTM) from bare ground information, which can be derived from LIDAR data sets. Here, the difference between *bare ground* and *DTM* is that bare ground is a set of sampled ground points; and DTM is the combination of these points in a geometric structure that permit more in depth terrain analysis. This work proposes five stages for LIDAR processing: the *filtering*, which detects bare ground information; the *DSM generation*, which rasterizes the point cloud; the *segmentation*, which separates interest elements from bare ground; the *splitting*, which splits the segments into rough and smooth segments; and finally the *classification*, which classifies the segments of the split image into vegetation and edification.

Buildings are, probably, the most important objects in an urban environment. Building models are used e.g. to the system optimization of telecommunication antennas, for selection of photovoltaic devices or even to tourist information systems and energy demand approximation in mega cities (Tovari, 2006). However, buildings are complex objects. They can have very complex content and elevation variation in range data. They can be located at any places, and are often surrounded by other objects with similar radiometric properties such as roads or even trees. Many algorithms focus on the detection of building footprints in order to produce 3D city models (Baltsavias *et al.*, 1995; Hug & Wehr, 2007; Yoon *et al.*, 1999; Zhan *et al.*, 2002). LIDAR data can be used in the the separation of buildings from vegetation (Brunn & Weidner, 1997; Elberink & Mass, 2000; Henricson *et al.*, 1996; Hofamnn, 2001; Hu, 2003; Tovari, 2006). However, building footprints still cannot be detected fully automatically, and they are often assumed to be of simple shapes with orthogonal corners such as

rectangles or low-quality polygons (Vestri & Devernay, 2001; Vosselman, 1999; Wang & Schenk, 2000; Weidner & Förstner, 1995). This work proposes a building footprint detection methodology as a LIDAR and data fusion case study.

Precise altimetry data obtained with LIDAR and very good spatial resolution of photogrammetric digital imagery can be integrated in order to produce high quality urban maps. Since the amount of information contained in these data sets tends to be huge, automated procedures are indispensable to make this integration viable. LIDAR and aerial imagery must be preprocessed for the fusion: despite their complementarity, they have very different natures. In this work, the aerial imagery preparation means obtainment of exterior orientation parameters that correctly describe the relation between the LIDAR data set and the imagery; and then proceed a segmentation of the aerial images in order to produce objects that will be later classified. The preparation of the LIDAR data, in this work, means the creation of a segmented DSM, which will be classified with a fuzzy inference system into ground, vegetation and edification. The fusion methodology proposed in this work has two steps. First, spectral data from the aerial imagery is integrated to the segmented DSM though orthorectification in order to permit a better scene description for the classification. Then, the detected buildings in the DSM are projected into image space in order to detect the roofs in the aerial imagery. At last, the detected roofs are projected into object space. The objectives of this thesis can be summed up as follows:

Objectives

The **main objective** of this thesis is to implement a semi-automated monoplotting system for the integration of LIDAR data with aerial imagery.

The **specific objectives** are:

- To make possible a correlation algorithm between LASER images and aerial imagery;
- To study the integration of the results with GIS;
- To propose an integration methodology through image segmentation techniques and fuzzy logic classification;
- To implement a monoplotting system through the items mentioned above.

Thesis outline

This work is organized as follows. Chapter 1 shows and discusses main aspects of image processing issues used in this work. Section 1.1 shows a brief introduction to computer vision and Photogrammetry. Section 1.2 presents main aspects of aerial imagery orientation (interior and exterior). Section 1.3 presents the proposed mean-shift based segmentation algorithm. Chapter 2 presents the GIS oriented LIDAR data processing. Section 2.1 presents a brief introduction. Section 2.2 presents an overview of basic GIS concepts. Section 2.3 presents basics on LIDAR operation. Section 2.4 shows the filtering algorithm proposed by Brovelli *et al.* (2004) and the DTM generation. Section 2.5 presents the modified Araki algorithm for DSM generation and its segmentation. Section 2.6 presents the proposed height texture splitting algorithm. Section 2.7 presents the implemented LIDAR processing

at a glance. Chapter 3 presents the core of this work, showing how the laser-scanner and the images are fused. Section 3.1 shows the state-of-art on LIDAR and spectral data fusion. Section 3.2 presents the proposed fuzzy inference system for the split DSM classification. Section 3.3 shows the proposed monoplotting technique, with emphasis in building footprint detection. Chapter 4 shows the obtained results. Two data sets are used (UFPR and Biberach). Section 4.1 shows the methodology overview. Section 4.2 shows results obtained in image space. Section 4.3 shows results obtained in object space.

Chapter 1

From Photogrammetry to Computer Vision

1.1 Introduction

From the beginning of science, visual observation has played a major role. At that time, the only way to document the results of an experiment was by verbal description and manual drawings. The next major step was the invention of photography, which enabled results to be documented objectively. Three prominent examples of scientific applications of photography are astronomy, particle physics and photogrammetry. With help of photographs (but not only), astronomers were able to measure positions and magnitudes of stars, physicists discovered many elementary particles, and photogrammetrists produced topographic maps from aerial images. These were manual, expensive and time consuming procedures. Nowadays, we are in the middle of a second revolution sparked by the rapid progress in video and computer technology. The technology is now available to any scientist or engineer. In consequence, image processing has expanded and is further rapidly expanding from a few specialized applications into a standard scientific tool (Jähne, 2005).

Due to the progress cited above, the photogrammetric imaging techniques are changing from analog to digital very fast. The facilities and advantages of digital imaging are undeniable: lower material costs, better radiometric quality are two among many other advantages (Honkavaara *et al.*, 2006). These advantages permit an upgrade on the automation degree of the photogrammetric workflow, and it can be seen by the intensive use of automatic procedures for aerotriangulation and orthophoto production. Although the core of a photogrammetric project, the *stereoplotting* remains manual and very expensive in terms of time and resources. Making this step automatic requires many changes in the traditional photogrammetric workflow, starting from the data to be used.

LIDAR (Light Detection and Ranging) is an emerging technology on geographic data capture, and it is a candidate to help on the automation of photogrammetric tasks. The huge amount of 3D generated points are used to produce very precise DSM (Digital Surface Models), and after filtering processes, high resolution DTM (Digital Terrain Models). The challenge is to make two extremely distinct data sets compatible and interoperable, since 3D point clouds and aerial imagery have very distinct natures. Here, the use of large-format digital cameras would no longer be necessary, since the aperture angle of LIDAR systems are much smaller than large-format cameras. It makes more sense to use low-cost

digital cameras, which can deliver, operationally speaking, more compatible imagery for integration with LIDAR data. For more details, see chapter 2.

Increasing resolution and lower costs of off-the-shelf digital cameras are giving rise to their use in traditional and new photogrammetric activities (Habib *et al.*, 2006). The use of these cameras for photogrammetric purposes brings a relevant question: *"What is a low-cost camera?"*. DIN 18716-2 (2007) presents two definitions that can help to answer this question. The first definition is ***Messbild*** (metric image): *"Messbilder sind Bilder mit bekannter oder bestimmbarer innerer Orientierung zur Rekonstruktion des Aufnahmestrahlenbündels"* (metric images are images with known or estimable interior orientation that permit the reconstruction of the bundle of rays). The second definition is ***Messkamera*** (metric camera): *"Die Messkamera dient der Aufnahme von Messbildern"* (the metric camera serves the purpose of making metric images). Departing from these definitions, any camera whose interior parameters can be determined is a metric camera, no matter if it was projected or not for aerial mapping. Two main groups of low-cost cameras can be defined: the small-format consumer cameras; and the mid-format metric cameras, nowadays commonly installed on LIDAR systems.

Chapter 1 proposes image processing methodologies for preprocessing aerial images taken with low-cost cameras for its integration with LIDAR data. The steps are the image orientation (a classic photogrammetric issue) and image segmentation (a typical computer vision procedure). Section 1.2 shows the issues concerning image orientation (interior and exterior orientation). Section 1.3 shows the image segmentation procedure implemented in this work.

1.1.1 Equipment and Data Sets

This section presents the equipments and data sets used in this work. Two data sets are presented: fist a survey over the Centro Politécnico Campus of the Federal University of Paraná (UFPR), in Curitiba, Brazil; and second a survey over the Biberach downtown area in Baden-Württemberg, Germany.

UFPR Data Set Details

- **LIDAR System:** Optech ALTM 2050
- **Point Density:** $\sim 2 - 4\ points/m^2$
- **Camera Type:** Sony DSC-F717
- **Camera Resolution:** 5.2 Mpix
- **GSD:** $\sim 25cm$
- **Time Shift:** 2 months

(a) Sony DSC-F717 (b) Optech ALTM 2050

Figure 1.1: UFPR Data Set Systems

Biberach Data Set Details

- **LIDAR System:** Toposys Harrier 56
- **Point Density:** $\sim 4 - 8\ points/m^2$
- **Camera Type:** Applanix DSS-422
- **Camera Resolution:** 22Mpix
- **GSD:** $\sim 8cm$
- **Time Shift:** Simultaneous

(a) Applanix DSS-422 (b) Toposys Harrier 56

Figure 1.2: Biberach Data Set Systems

1.2 Image Orientation

Accordingly to Hartley & Zisserman (2004) *"a camera is a mapping between the 3D world (**object space**) and a 2D image (**image space**)"*. In Photogrammetry a *projective camera* model is commonly used. A projective camera P maps world points X to image points x according to $x = PX$. This is called image orientation, and is one of the most important tasks in any photogrammetric application. Traditionally it consists of two steps: the *interior orientation* (section 1.2.1), which recovers the image geometry through the knowledge of calibration parameters, and the *exterior orientation* (section 1.2.2), which determines the relation between the image space and the object space.

The *image space* is related to the *photogrammetric reference system*, an ideal three-dimensional reference system whose z-axis coincides with the camera's optical axis, and has as origin the principal point – the point where the optical axis crosses the image plane. The *raw-image reference system*, which is a new version of the fiducial reference system (since the CCD geometry substitutes the fiducial marks), is a two-dimensional reference system directly related to the image coordinates. The interior orientation transforms the coordinates from the raw-image reference system to the photogrammetric reference system.

The *object space* is related to a three-dimensional *real-world* reference system. In case of aerial photogrammetry this is a geodetic reference system. The exterior orientation transforms the coordinates from a real-world reference system to the photogrammetric reference system. This is the most important relation in Photogrammetry and it is done through the collinearity equations.

The Collinearity Equations

In Photogrammetry, the transformation ($x = PX$) of coordinates from object space (X, Y, Z) to the image space (x_p, y_p, c) is given by the collinearity equation. A total of six transformation parameters per image are determined: the projection center coordinates (X_0, Y_0, Z_0) and three rotations $(\omega, \varphi, \kappa)$. The rotations are combined to a matrix of rotations M, and the final form of the equations is:

$$\begin{aligned} x_p &= -c \cdot \frac{m_{11}(X - X_0) + m_{12}(Y - Y_0) + m_{13}(Z - Z_0)}{m_{31}(X - X_0) + m_{32}(Y - Y_0) + m_{33}(Z - Z_0)} \\ y_p &= -c \cdot \frac{m_{21}(X - X_0) + m_{22}(Y - Y_0) + m_{23}(Z - Z_0)}{m_{31}(X - X_0) + m_{32}(Y - Y_0) + m_{33}(Z - Z_0)} \end{aligned} \tag{1.1}$$

Where (X, Y, Z) are the coordinates of a point in the object space, (X_0, Y_0, Z_0) the coordinates of the perspective center and $(m_{11}$ ... $m_{33})$ the elements of the eulerian rotations matrix, given by the successive rotations in the three axes:

$$M = R_Z(\kappa) \cdot R_Y(\varphi) \cdot R_X(\omega) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \tag{1.2}$$

1.2.1 Interior Orientation and Geometric Calibration

The aim of the geometric calibration is to determine the interior orientation parameters of the camera in order to model systematic geometric deviations and distortions from the perspective geometry (Doerstel *et al.*, 2003). Several parameters are defined and calculated, based on physical models, which means that all components are derived from actual physical error sources (Machado *et al.*, 2004). The most popular set of parameters was presented in Brown (1971): the focal length (c), the principal point (x_0, y_0), the radial symmetric distortions (k_1, k_2, k_3) and the decentering distortions (P_1, P_2). There are other sets of parameters, like out-of-plane unflatness and in-plane distortion (Fraser, 1997), but these errors (and others) will be be directly treated during aerotriangulation in this work (section 1.2.2). The radial symmetric distortions are caused by fabrication failures which lead to errors in the lense shapes. The decentering distortions are caused by an eccentricity of individual lenses, amongst the lens compound, with respect to the optical axis. Given the raw coordinates of a point on one image (x_d, y_d), and the set of calibration parameters given above, the deviations caused by the radial symmetric distortions $(\delta r_x, \delta r_y)$ and the descentering distortions $(\delta d_x, \delta d_y)$ are given by (Brown, 1971):

$$\begin{aligned} \delta r_x &= (k_1 r^2 + k_2 r^4 + k_3 r^6)(x_d - x_0) \\ \delta r_y &= (k_1 r^2 + k_2 r^4 + k_3 r^6)(y_d - y_0) \\ \delta d_x &= P_1(r^2 + 2(x_d - x_0)^2) + 2P_2(x_d - x_0)(y_d - y_0) \\ \delta d_y &= P_2(r^2 + 2(y_d - y_0)^2) + 2P_1(x_d - x_0)(y_d - y_0) \end{aligned} \tag{1.3}$$

Where:

$$r = \sqrt{(x_d - x_0)^2 + (y_d - y_0)^2} \tag{1.4}$$

The corrections from the raw-image reference system to the photogrammetric reference system (x_p, y_p) are given on equation (1.5). These formulas are presented in (Moniwa, 1977):

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} - \begin{bmatrix} \delta r_x \\ \delta r_y \end{bmatrix} - \begin{bmatrix} \delta d_x \\ \delta d_y \end{bmatrix} \tag{1.5}$$

Departing from the classical collinearity equation (1.1), using (1.5) for (x_p, y_p), abbreviating the long terms on (1.1) by $(\Delta X, \Delta Y, \Delta Z)$ and grouping all unknowns on the right side of the equation one comes to:

$$\begin{aligned} x_d &= -c \cdot \frac{\Delta X}{\Delta Z} + x_0 + \delta r_x + \delta d_x \\ y_d &= -c \cdot \frac{\Delta Y}{\Delta Z} + y_0 + \delta r_y + \delta d_y \end{aligned} \tag{1.6}$$

In equation (1.6), the raw image coordinates are functions of the exterior orientation parameters and the calibration parameters. The solution by least-squares yields all calibration parameters. Photogrammetric cameras are mostly calibrated in a laboratory with the help of an optical goniometer (Kraus, 2007), but this method is not suitable for consumer low-cost cameras, and these cameras are mainly calibrated using close-range photogrammetric techniques. In this case, the most used methodology is the *self-calibration*. This term is used in computer vision when no calibration object is employed and the metric properties of the camera and of the imaged scene are recovered from a set of "uncalibrated" images, using constraints on the camera parameters or on the imaged scene (Remondino & Fraser, 2006). There is an extensive body of literature on low-cost digital camera calibration, like Cronk *et al.* (2006); Habib & Morgan (2005); Habib *et al.* (2006); Kunii & Chikatsu (2001); Läbe & Förstner (2004), just to mention a few. This number of works proves the potential of low-cost cameras for precision photogrammetric tasks. Table 1.1 shows the callibration parameters for the Sony DSC-717 – fig. 1.1(a).

Parameter	Value
c	$10.078mm$
x_0	$-0.246mm$
y_0	$-0.142mm$
k_1	$-2.25 \times 10^{-3} mm^{-2}$
k_2	$2.49 \times 10^{-5} mm^{-4}$
k_3	$1.29 \times 10^{-7} mm^{-6}$
P_1	$-1.17 \times 10^{-5} mm^{-1}$
P_2	$6.9 \times 10^{-5} mm^{-1}$

Table 1.1: Sony DSC-717 Calibration Parameters

For a typical low-cost camera calibration, several pictures are taken from a set of control points, and a bundle adjustment is performed, having as unknown the calibration parameters and the exterior orientation parameters (see eq. 1.6). Due to the high statistical correlation between calibration parameters and exterior orientation parameters, the self-calibration is only reliable when the network geometry is favorable: the camera station configuration comprises highly convergent images, orthogonal roll angles and a large number of well distributed points (Remondino & Fraser, 2006).

An important aspect of the suitability of low-cost digital cameras is its geometric stability (Läbe & Förstner, 2004). When used for aerial photogrammetric works, the cameras should be calibrated without auto-focus and with the focus set to infinite. The camera configuration during the flight must be the same as during the calibration. Läbe & Förstner (2004) show that the parameters determined by self-calibration may be used for photogrammetric applications. Another important point is the parameters temporal stability. In (Habib *et al.*, 2006) and (Habib & Morgan, 2005) several tests are performed and the conclusion is that the parameters are stable along the time. These two important conclusions: temporal stability and reliable mathematical modeling makes the low-cost cameras suitable for aerial photogrammetric projects.

1.2.2 Exterior Orientation and Aerotriangulation

Exterior orientation describes the location and orientation of an image in the object space. Traditionally, in order to georeference aerial photos, a bundle of rays from image points through the corresponding projection center to the ground is modeled by means of interior orientation while the exterior orientation is determined by using ground control points (Jacobsen, 2001).

Exterior orientation parameters of each image in a block can be determined through a block adjustment (aerotriangulation) using ground points as control. The mathematical model behind the block adjustment is based on the collinearity equations. Ground coordinates as well all exterior orientation parameters of each photo are the results of a block adjustment. The image orientation is either used directly or transformed into the required system. This methodology leads to an optimal fit of neighbored models and avoids additional efforts for model orientation (Jacobsen, 2001).

The traditional 8-parameter model of Brown (section 1.2.1) is normally not enough for a complete distortion modeling in one or more images. Errors like image non-flatness and the frequent change of the calibrated focal length are not included in the calibration certificate provided by the camera manufacturers. Changes of camera focal length occur due to a change of air pressure and temperature, which leads to deformations in the bundle of rays. These problems tend to be even more remarkable in consumer low-cost cameras. These deviations from the mathematical model tend to sum up causing deformations in the photogrammetric image blocks. Consequently, traditional bundle adjustment by ground control yields systematic deformations, especially in the altimetry, which is more sensitive to these effects (Jacobsen, 2001).

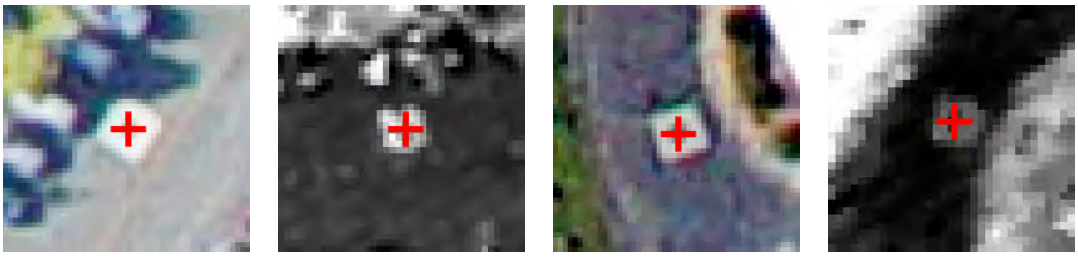
Instead of investigating the deviations from the mathematical models, it is possible to adapt the mathematical models in order to accommodate these deviations. This technique is called *bundle adjustment with self calibration by additional parameters*, and will be discussed later in this chapter. First some considerations about the control points collected from the LIDAR data are presented.

LIDAR Data for Exterior Orientation

This work as a whole aims the integration of LIDAR data and digital low cost imagery. In most modern laser scanners, a mid-format camera is integrated with the system and the exterior orientation parameters are directly determined by GPS and IMU. Using few ground control points, this orientation can be refined via an aerotriangulation. In these cases, the exterior orientation parameters are automatically compatible with the LIDAR data.

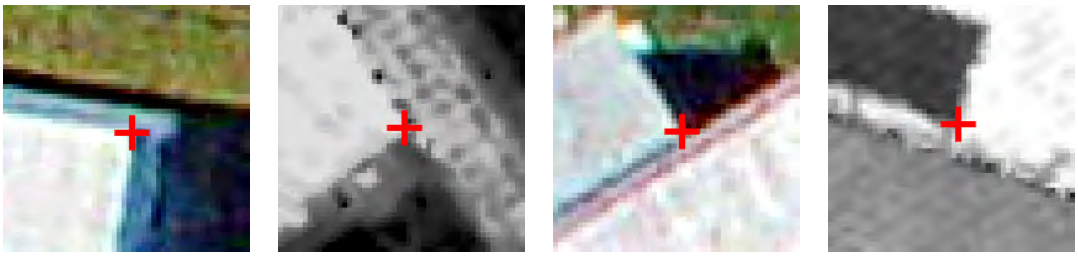
When imagery and LIDAR are not simultaneously surveyed, a geometric fit of both data sets is necessary. This integration, also called co-registration, is the target of several research studies (Habib *et al.*, 2008). In (Habib *et al.*, 2004) the co-registration is achieved through the use of linear features derived from LIDAR data as control information for aligning the model relative to the LIDAR reference frame. Santos *et al.* (2007) uses point correspondence between intensity LIDAR images and LIDAR images and aerial imagery based on corner and edge detection, area based matching, cross correlation coefficient and 2D transformation. Mitishita *et al.* (2008) proposes a methodology that uses the centroids of rectangular building roofs. Rönnholm *et al.* (2007) presents an overview of various co-registration approaches.

This work also uses the laser intensity image as a tool to collect ground control points for the external orientation. Since this orientation is not the main objective here, the points were simply collected manually. Two kind of points have been collected: ground control points for aerotriangulation, and check points for quality measurements. The ground control points are a mixture of pre-signalized (fig. 1.3) and well observable topographic points (fig. 1.4). The check points are exclusively pre-signalized points. Note in figure (1.3) the potential of pre-signalized points (white boxes with black circles inside) for co-registration when the camera is not integrated in the system.



(a) Pt. 1 – Photo (b) Pt. 1 – Intensity (c) Pt. 2 – Photo (d) Pt. 2 – Intensity

Figure 1.3: Examples of Pre-signalized Points



(a) Pt. 3 – Photo (b) Pt. 3 – Intensity (c) Pt. 4 – Photo (d) Pt. 4 – Intensity

Figure 1.4: Examples of Topographic Points

Aerotriangulation with Additional Parameters

The outline of performing an aerotriangulation with additional parameters is very similar to the self-calibration process mentioned in section (1.2.1). There is a confusion between self-calibration for calibration purposes and the use of additional parameters for a better fitting of not pre-refined errors. The first aims at obtaining of calibration parameters to be used in the interior orientation, and these are parameters that are used for long periods. The bundle adjustment with self calibration by additional parameters (also known as on-the-flight calibration) aims at reduction of systematic errors

which remains after interior orientation. Another main difference between the two usages of self-calibration is that, on the empirical models, the additional parameters are orthogonal to the exterior orientation ones, i.e. they are statistically independent from the exterior orientation parameters (EOP). For calibration, these parameters are strongly correlated with the EOP. There are many self-calibration models in use, e.g. the ones proposed by Ebner (1976), Grün (1978) and Jacobsen (1982). The most popular is the 12-parameter Ebner model (Kraus, 1996). Figure (1.5) shows this model.

This technique was developed for film cameras and has become standard in aerotriangulation with large-format digital cameras. This work applies the methodology to a consumer low-cost digital camera. First, a pre-refinement of the raw-image coordinates using the calibration parameters is performed (see section 1.2.1), and then, during the aerotriangulation, the rest of the errors are treated with Ebner's model. The functional model for the bundle adjustment with additional parameters is very similar to equation (1.6):

$$\begin{aligned} x_p &= -c \cdot \frac{\Delta X}{\Delta Z} + \delta a_x \\ y_p &= -c \cdot \frac{\Delta Y}{\Delta Z} + \delta a_y \end{aligned} \tag{1.7}$$

In equation (1.7) δa_x and δa_y represent the non pre-refined distortions. Figure (1.5) shows these terms as functions of 12 additional parameters ($b_1 ... b_{12}$) and the image dimensions (B_x, B_y) (Ebner, 1976). The pictures show the modeling effects on the images. After the adjustment, statistical tests are used to exclude additional parameters that show no significance. This procedure is iteratively done until all additional parameters show relevant.

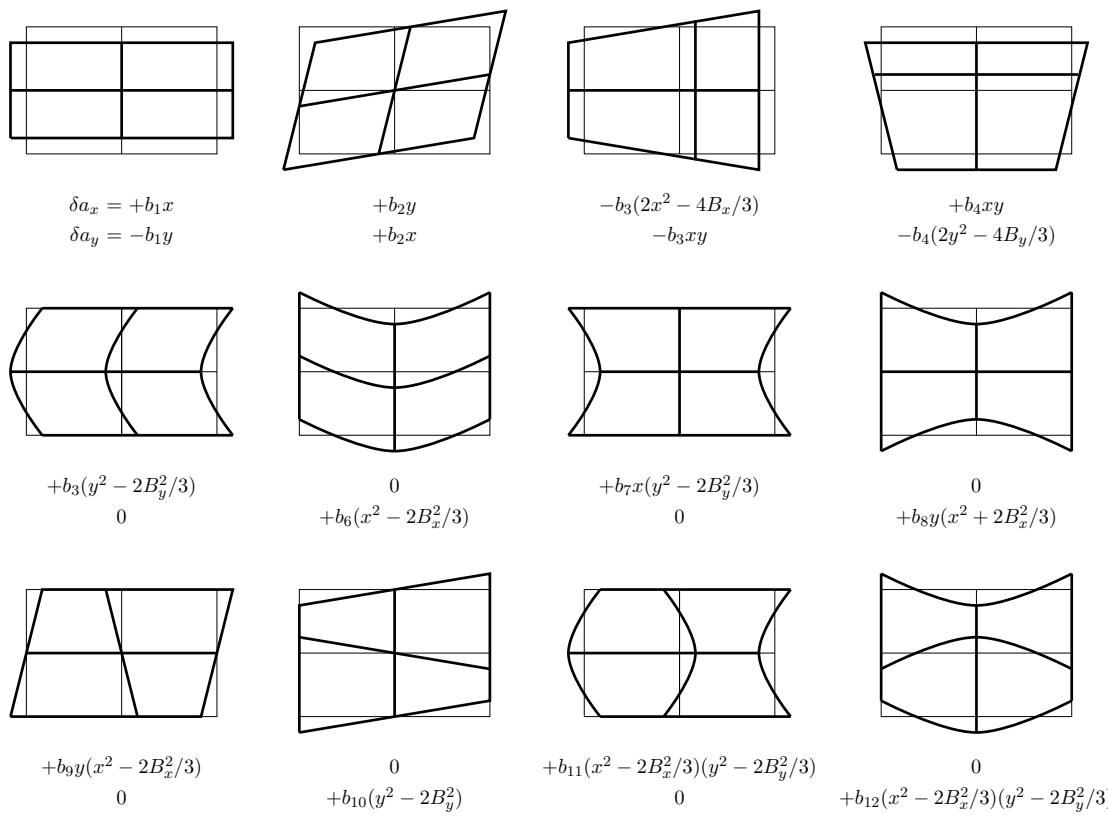


Figure 1.5: Ebner Additional Parameters for Aerotriangulation (Ebner, 1976)

UFPR Data Set Block Adjustment

This section shows results of a bundle adjustment with 13 images flown over the Centro Politécnico, in Curitiba (fig. 1.6), taken with the Sony DSC-717 camera – fig. 1.1(a). To study the effects of the additional parameters, two aerotriangulations are performed, with and without additional parameters. 10 pre-signalized check points (see fig. 1.3) are used to measure the quality of both adjustments. The lower the discrepancy, the better the aerotriangulation result. Figure (1.7) shows the obtained discrepancies in (X, Y, Z).

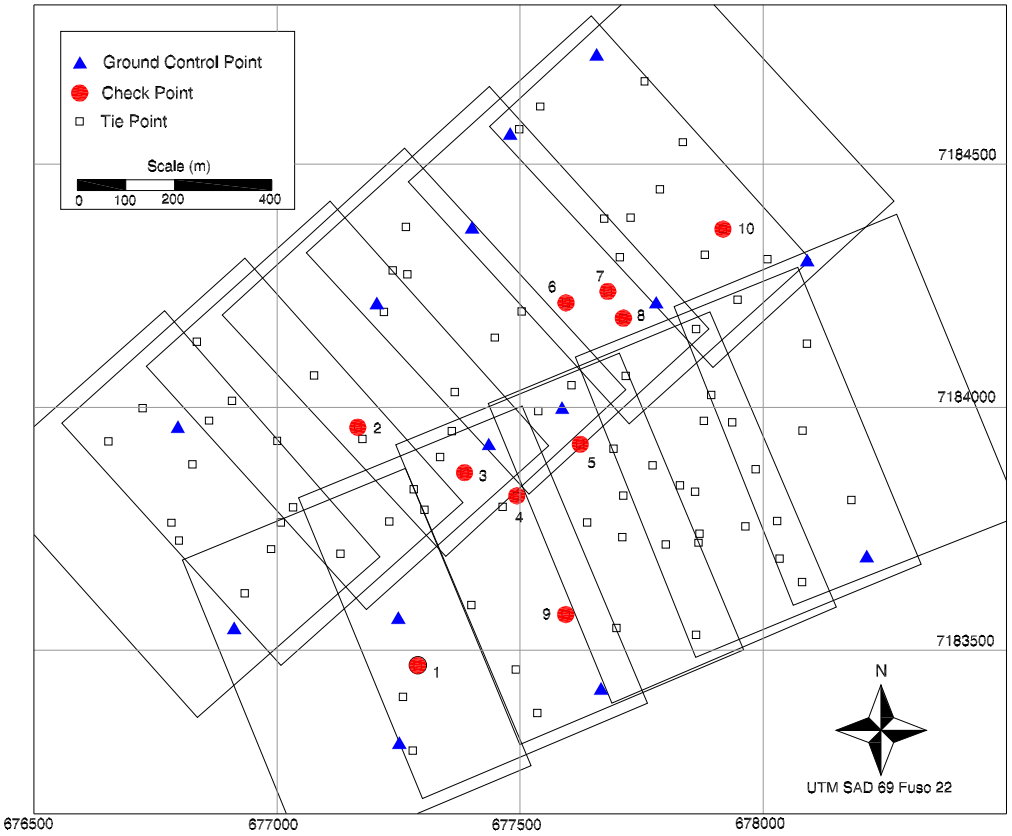


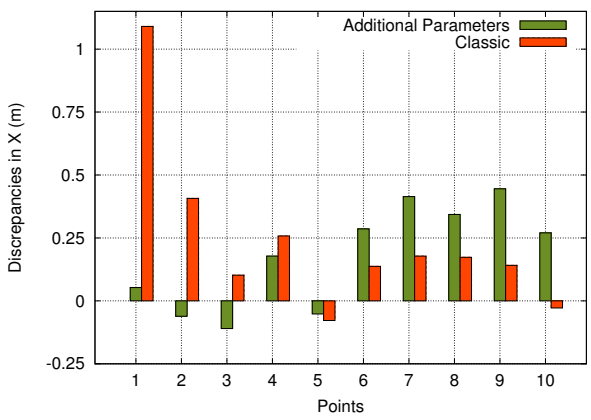
Figure 1.6: Centro Politécnico Block

This work uses the *t-Student* test to verify the additional parameter's relevance, as pointed out in Kraus (1996). The block has **659** observations (n), **405** parameters (u) and **254** degrees of freedom ($n - u$). For a 90% significance test, the $|t|$ value must be higher than **1.256**. It means that if a parameter has a value lower than this threshold, the hypothesis that this parameter is zero is not rejected. After the processing, the parameters $b_2, b_5, b_6, b_7, b_8, b_9$ and b_{10} showed significant. Table (1.2) shows the significant parameters.

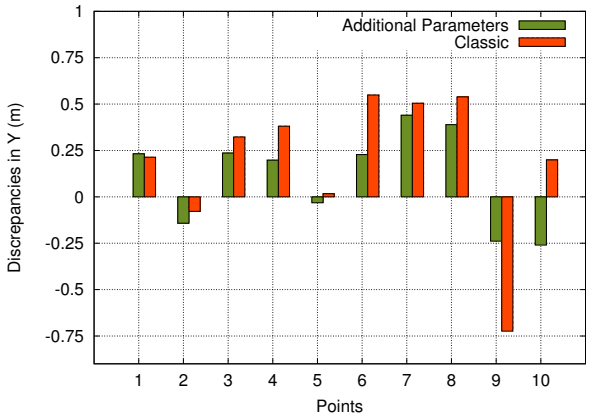
Parameter	Value	Standard Deviation	$\|t\|$
b_2	1.054853×10^{-3}	5.631074×10^{-4}	1.873
b_5	-1.148767×10^{-6}	5.291227×10^{-7}	2.173
b_6	-6.784402×10^{-7}	1.915882×10^{-7}	3.541
b_7	$-5.191415 \times 10^{-10}$	1.777683×10^{-10}	2.920
b_8	$-1.929325 \times 10^{-10}$	3.110714×10^{-11}	6.202
b_9	3.731200×10^{-10}	1.770648×10^{-10}	2.107
b_{10}	7.333639×10^{-11}	1.927014×10^{-11}	3.801

Table 1.2: UFPR Block – Significant Additional Parameters

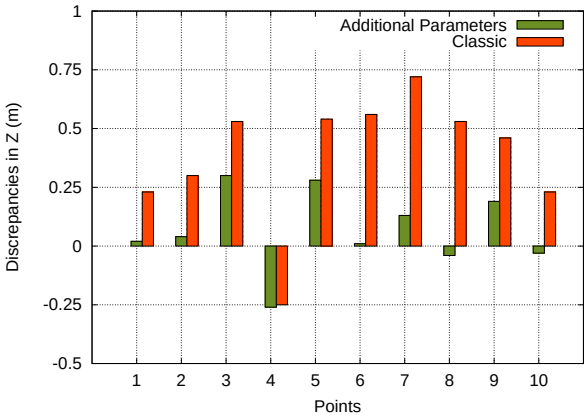
A better accuracy is expected when using additional parameters. This can be verified through the analysis of the check points discrepancies (fig. 1.7). Results show that the altimetric accuracy is much better when using additional parameters, as expected. The planimetric results also show better accuracy, but not as significantly as for the altimetry. These results prove the importance of a correct aerotriangulation modeling when working with low-cost digital cameras.



(a) Discrepancies in X



(b) Discrepancies in Y



(c) Discrepancies in Z

Figure 1.7: Check Points Discrepancies

1.2.3 Distortion-free Image Generation

Consumer low-cost digital cameras have lenses distortions in the order of 10 times larger than metric cameras. The optical systems of these cameras are projected to make beautiful and sharp images, and due to the "Scheimpflug Principle" the costs for sharp images are higher distortions. For aerial Photogrammetry these distortions are not acceptable: distortions can turn straight lines into curves, and this kind of effect can cause failures in automation algorithms.

In order to work with an image without distortions, this sections proposes the creation of a distortion-free image using the calibration parameters. The first step is to create a blank image which represents the planar coordinates (x_p, y_p) of the photogrammetric reference system. Then, every pixel on this image is mapped back to the distorted reference system, using the calibration parameters. This procedure of "adding" the distortions is widely used on orthophoto production.

The "distortion addition" can be achieved inverting the equation (1.5) into equation (1.8) below:

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} x_p \\ y_p \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \delta r_x \\ \delta r_y \end{bmatrix} + \begin{bmatrix} \delta d_x \\ \delta d_y \end{bmatrix} \tag{1.8}$$

Note that $\delta r_x, \delta r_y, \delta d_x, \delta d_y$ are functions of x_d, y_d (see equations 1.3 and 1.4). For metric cameras the equation (1.8) shows sufficient for the modeling, since the distortions have low values. For consumer low-cost cameras, though, a better mathematical modeling shows necessary (Machado *et al.*, 2004). If it is not done, the mapping from the photogrammetric reference system to the raw image coordinate system will not correctly produce a distortion-free image. This error is called *Inverse Mapping Error* and the figure (1.8) presented in Machado *et al.* (2004), shows the effects of this error, in pixels, for the Sony DSC-F717 – fig. 1.1(a).

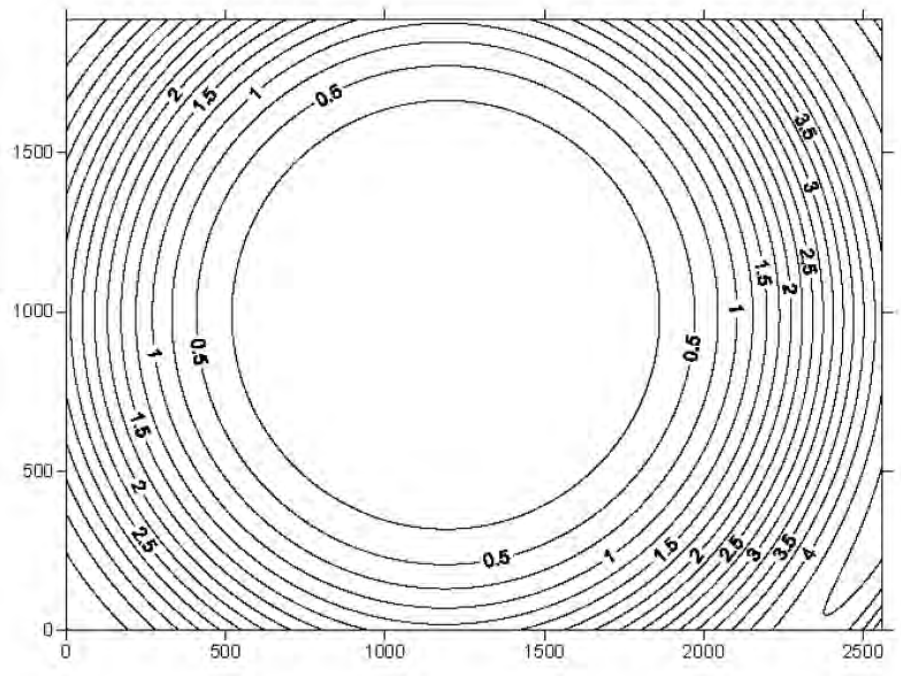


Figure 1.8: Sony DSC-F717 Inverse Mapping Errors in Pixel (Machado *et al.*, 2004)

In order to avoid the inverse mapping errors, Machado *et al.* (2004) propose the use of a numeric iterative method called *Newton-Raphson Method*. This method permits an iterative determination of coordinates in the raw-image reference system (x_d, y_d) based on the calibration parameters values and the photogrammetric coordinates (x_p, y_p).

If the logarithmic derivative of a function $f(x)$ can be simply determined, the real roots from $f(x) = 0$ may be determined fast through this method. Be a the approximation value of a root x, and h the correction to this value, then $x = a + h$ and the equation $f(x) = 0$ becomes $f(a + h) = 0$. Expanding by Taylor:

$$f(a+h) = f(a) + h.f'(a) + \frac{h^2}{2}(f''(a + \Theta h) = 0 \text{ with } 0 \leq \Theta \leq 1 \tag{1.9}$$

If h is a small value, the term containing h^2 can be eliminated:

$$f(a+h) \approx f(a) + h.f'(a) = 0 \tag{1.10}$$

For the first iteration, the value of h_1 is determined from eq. (1.10):

$$h_1 = -\frac{f(a)}{f'(a)} \tag{1.11}$$

The root's new value is given by:

$$a_1 = a + h_1 = a - \frac{f(a)}{f'(a)} \tag{1.12}$$

If n is the number of the iteration, the root's new values are given by:

$$a_n = a_{n\text{-}1} - \frac{f(a_{n\text{-}1})}{f'(a_{n\text{-}1})} \tag{1.13}$$

In the case of the generation of the distortion-free image, $f(x) = 0$ is determined by:

$$f(x_d, y_d) = \begin{bmatrix} x_p \\ y_p \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \delta r_x \\ \delta r_y \end{bmatrix} + \begin{bmatrix} \delta d_x \\ \delta d_y \end{bmatrix} - \begin{bmatrix} x_d \\ y_d \end{bmatrix} = 0 \tag{1.14}$$

Using as initial values the photogrammetric coordinates (x_p, y_p) and solving the equations in (1.14) using (1.13), the inverse mapping errors are avoided. The algorithm 1 presents the creation of a distortion free image. An example is shown in figure (1.12) page 44.

Algorithm 1: Distortion-Free Image Generation

Data: Original Image O_i, Calibration Parameters
Result: Distortion Free Image DF_i
Determine the Distortion-Free Image size $(ncol, nlin)$
for $col = 0 \rightarrow ncol$ **do**
  for $lin = 0 \rightarrow nlin$ **do**
    $x_p = col - \frac{ncol}{2}$
    $y_p = \frac{nlin}{2} - lin$
    Initialize $(x_d, y_d) = (x_p, y_p)$
    while $h < threshold$ **do**
      $h = \frac{f(x_d, y_d)}{f'(x_d, y_d)}$
      $(x_d, y_d) = (x_d, y_d) - h$
    end
    Interpolate color values from O_i using (x_d, y_d)
    Set color values in DF_i at (col, lin)
  end
end

1.3 Image Segmentation

Forsyth & Ponce (2002) define computer vision as *"an enterprize that uses statistical methods to disentangle data using models constructed with the aid of geometry, physics and learning theory"*. Image segmentation is an image processing technique, whose goal is to obtain a compact representation of what is helpful in one image, or recognize objects on it. This procedure is one of the most important add-ons of computer vision on the modern Photogrammetry, because it helps dealing with the huge amount of information that high resolution imagery brought (Forsyth & Ponce, 2002).

There are four main groups of techniques for image segmentation (also known as *paradigms of image segmentation*): the *pixel-based* methods, where just the values of the pixels are used; the *region-based* methods, where the pixel values are analyzed in larger areas; the *edge-based* methods, where the goal is to find edges and follow them; and at last the *model-based* methods, where geometric shapes of the objects can be used (Jähne, 2005).

Image segmentation is a challenging task. The richness of visual information makes bottom-up, solely image driven approaches always prone to errors. To be accurate, all current systems must be large and incorporate numerous ad-hoc procedures. Since perfect segmentation cannot be achieved without a top-down knowledge driven component, a bottom-up technique should only provide the input to the next stage where the task is accomplished using a priori knowledge about this goal; and eliminate as much as possible, the dependence on user parameter values (Comaniciu & Meer, 1997).

Despite some early research activities, image segmentation was established late in the field of geoinformation. First beginning with the availability of high resolution imagery ($< 1m$) and their characteristics (high level of detail, spectral variance, etc.) this method has become popular as a common variant of data interpretation (Meinel & Neubert, 2004). Nowadays, there are a plenty of segmentation techniques being applied to geoinformation sciences, since image segmentation became a crucial step within the object-based Remote Sensing information retrieval process. (Neubert *et al.*, 2008).

Meinel & Neubert (2004); Neubert *et al.* (2006, 2008) describe 15 software packages for image segmentation, using different algorithms. The listing following shows an abstract of these papers:

- *eCognition* from Definiens Imaging GmbH, Munich, Germany;
- *Image Segmentation for ERDAS Imagine* from USDA Forest Service, Remote Sensing Applications Center, Salt Lake City, USA;
- *Imagine WS for ERDAS Imagine* from the Austrian Academy of Sciences, Vienna, Austria;
- *HalconSEG* an adapted Lanser–segmentation algorithm for HALCON, MVTec GmbH, Munich, Germany;
- *SegSAR 1.0* from the National Institute for Space Research, São José dos Campos, Brazil;
- *ENVI Feature Extraction Module 4.4* from ITT Visual Information Solutions, Boulder, USA;
- *BerkeleyImgseg 0.54* from BETI – Berkeley Environmental Technology International, LLC, Berkeley, USA;
- *EDISON* from Rutgers University, Robust Image Understanding Lab.

In this work, the segmentation's objective is to obtain a reliable object-oriented representation of the images without loosing its geometric properties. The main objective is to use the high planimetric resolution from aerial imagery combined with LIDAR data. It means that the chosen techniques must generate objects keeping their edges as faithful as possible to the original image.

1.3.1 Mean-Shift Algorithm for Image Processing

One algorithm that shows good results regarding discontinuity preserving is the *Mean-Shift Image Segmentation Algorithm*, first described by Comaniciu & Meer (2002). A version of this algorithm is

implemented in *EDISON* and tested with aerial images in (Machado, 2006), achieving good results. For these reasons, this method was chosen and adapted in this work. An improved version is presented in (Comaniciu & Meer, 2002). This paper shows how to perform a feature space analysis in images using the mean-shift algorithm. Sudhamani & Venugopal (2006) presents an application for the algorithm. Park *et al.* (2009) and Cho *et al.* (2004) propose enhancements to the methodology by adding statistical model-based methods. Just to cite a few further applications, Zhou *et al.* (2009) use the mean-shift algorithm combined with a scale invariant feature transformation for feature tracking on video sequences. Collins (2003) uses a similar approach for 2D blobs through an image. Kim *et al.* (2003) uses mean-shift for text detection in images, and Yang & Pei (2008) proposes a landmarks corresponding estimation in multimodal medical image registration using the mean-shift algorithm.

This section presents the mean-shift procedure, the core of all methodologies cited above. Then a new *mean-shift seed-based region growing segmentation* methodology is presented.

The Mean-Shift Procedure

On an arbitrary set of points, the knowledge of its associated probability density function (pdf) is extremely relevant, since it can help on important physical interpretations of the observations. Sometimes this distribution is well known but many times the data behavior is not known, and one is impelled to use non-parametric approaches for data analysis. One of the most popular method is the *Kernel Density Estimator* (Silverman, 1986).

Given n data points $\{x_{\mathrm{i=1:n}}\}$ on a d-dimensional Euclidian space $\Re^d$, the multivariate kernel density estimate obtained with a kernel $K(x)$, of the the variable x, inside of a band-width h is:

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{\mathrm{i=1}}^{n} K\left(\frac{x - x_i}{h}\right) \tag{1.15}$$

The Epanechnikov kernel density estimation defined for 1.15 is given by:

$$\hat{\nabla} f(x) = \frac{n_x}{n(h^d c_d)} \cdot \frac{(d+2)}{h^2} \left(\frac{1}{n_x} \sum_{x_i \in S_h(x)} (x_i - x)\right) \tag{1.16}$$

Here $S_h(x)$ represents a sphere with radius h, volume $h^d c_d$, centered in x and containing n_x points. The first term of the eq. (1.16) is the representative quantity of the kernel density $\hat{f}(x)$, determinated inside the sphere $S_h(x)$ for a uniform kernel:

$$\hat{f}(x) = \frac{n_x}{n(h^d c_d)} \tag{1.17}$$

The last term of 1.16 is called *mean shift vector*:

$$M_h(x) = \frac{1}{n_x} \sum_{x_i \in S_h(x)} (x_i - x) \tag{1.18}$$

Combining the 1.16, 1.17 and 1.18 one comes to:

$$M_h(x) = \frac{h^2}{(d+2)} \cdot \frac{\hat{\nabla} f(x)}{\hat{f}(x)} \tag{1.19}$$

The algorithm 2 presents the pseudo-code implemented to find the high-density centers:

Algorithm 2: The Mean Shift Algorithm

Data: Band Width (h), Initial Point (x_i), Mean Shift threshold (M_t), Maximal Iterations Number ($it_{\max}$)

Result: Convergence Point (x)

Initialize localization window: $x \leftarrow x_i$

Set Iteration Number to zero: $it \leftarrow 0$

while $M_h(x) < (M_t)$ **do**

  Find all points within h

  Calculate $M_h(x)$

  $x \leftarrow x_i + M_h(x)$

  $it \leftarrow it + 1$

  if it_{max} **then**

    No convergence

  end

end

The mean-shift algorithm is a nonparametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters. The mean shift algorithm is a simple iterative procedure that shifts each data point to the average of data points in its neighborhood (Cheng, 1995). The mean-shift vector always points to the region with higher densities. When one "walks" in this direction, one tends to find the center of a cluster. The repeated movement of data points to the sample means is called *mean-shift algorithm* (Cheng, 1995). Figure (1.9) shows an example of two paths departing from points going to the center of a cluster. The blue circles represent the band-width (h).

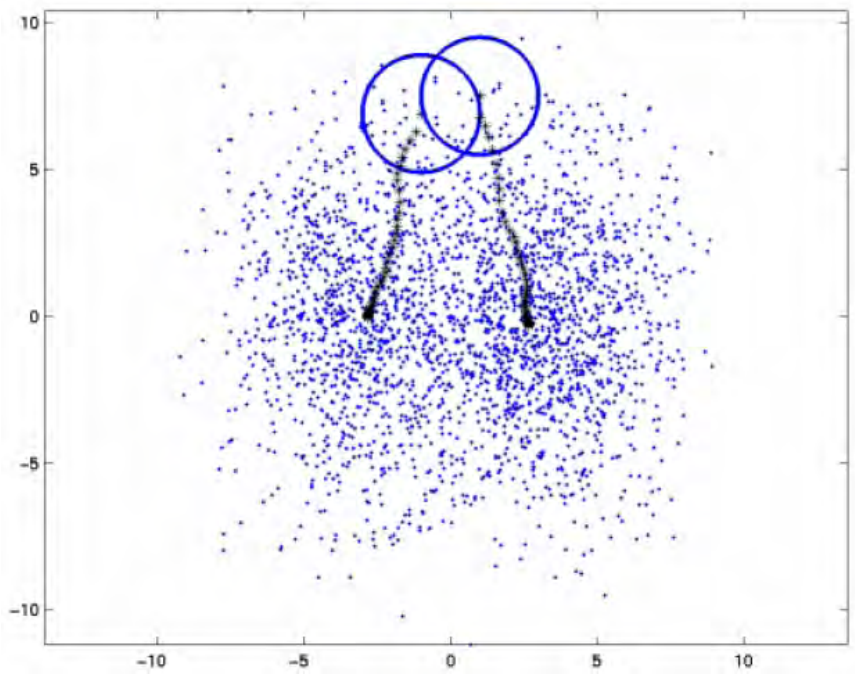


Figure 1.9: Mean-Shift Algorithm Path

Band Width and Segmentation Resolution

As one can see in the previous sections, the band width is a very important parameter to be determined prior to any usage of the mean-shift procedure. In image segmentation, the band width is related to the segmentation resolution, which is the most general parameter characterizing a segmentation technique. Comaniciu & Meer (1997) propose three resolution classes: *undersegmentation*, where just the dominant edges in the images are shown; *oversegmentation*, where the image is broken into many small regions from which any sought information can be assembled under knowledge control; and *quantization*, where the feature palette contains all the significant colors in the image.

The subjective definition of a homogeneous region on an image seems to depend on the *visual activity* in the image. Within the same segmentation class an image containing large homogeneous regions should be analyzed at higher resolution than an image with many textured areas. The simplest measure of the visual activity can be derived from the image's global covariance matrix (Comaniciu & Meer, 1997). This measure is the standard deviation (σ) of the image's histogram (Machado *et al.*, 2004). The band-width (h_r) for the colors in an image is taken proportional to σ. Table (1.3) presents rules proposed by Comaniciu & Meer (1997) for the three segmentation resolutions:

Segmentation Class	h_r
Undersegmentation	$0.4 \cdot \sigma$
Oversegmentation	$0.3 \cdot \sigma$
Quantization	$0.2 \cdot \sigma$

Table 1.3: Segmentation Class Parameters (Comaniciu & Meer, 1997)

Two other ways to determine the band width are proposed in (Comaniciu *et al.*, 2001). The first is based on the adaptive estimation of the normalized density gradient, and the second is a semi-parametric technique that imposes a local structure on the data to extract reliable scale information. Although, since aerial imagery has a very complex nature, it is better to keep a simple approach to the band width selection problem, and for this reason, in this work, the technique using the histogram will be applied.

An image is typically represented as a two-dimensional lattice of *d-dimensional* vectors (pixels), where $d = 1$ in the gray-level case, three for color images and $d > 3$ in the multispectral case. The space of the lattice is known as the *spatial domain* (s), while the spectral information is represented in the *range domain* (r). Since both domains have different natures, the band-width selection and the mean-shift procedure must be normalized. (Comaniciu & Meer, 2002) shows that a multivariate kernel can be defined as the product of two radially symmetric kernels K_r for the range domain and K_s for the spatial domain. Given a range band-width h_r and a spatial band-width h_s, the density gradient estimation (eq. 1.15) becomes:

$$\hat{f}(z_j^m) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h_r h_s} K_r\left(\frac{r_j^m - r_i^m}{h_r}\right) K_s\left(\frac{s_j^m - s_i^m}{h_s}\right) \tag{1.20}$$

The mean-shift algorithm proceeds on both domains separately:

$$r_j^{m+1} = r_j^m + M_{h_r,K_r}(r_j^m) \quad , \quad s_j^{m+1} = s_j^m + M_{h_s,K_s}(s_j^m) \tag{1.21}$$

The convergence occurs simultaneously in both domains when both mean-shift vector have a value smaller than a pre-defined threshold.

Mean-Shift Image Filtering

One important application of the mean-shift procedure in computer vision is the image *smoothing* or *filtering*. Smoothing through replacing the pixel in the center of a window by the average of the pixel in the window indiscriminately blurs the image, removing not only the noise but also salient information. Discontinuity preserving smoothing techniques adaptively reduce the amount of smoothing near abrupt changes in the local structure, i.e. edges (Comaniciu & Meer, 2002). The *mean-shift image filtering* is an option to perform this task.

Let x_i be the d-dimensional input, and $z_i, i = 1, ..., n$ the filtered image pixels in the joint spatial-range domain. The algorithm runs as follows:

Algorithm 3: Mean Shift Filtering

Data: Band Width (h_r, h_s), Original Image (O_i)
Result: Filtered Image (F_i)
for $col = 0 \rightarrow ncol$ **do**
  for $lin = 0 \rightarrow nlin$ **do**
    $j \leftarrow 1$
    $y_{i,\,j+1} \leftarrow x_i$
    Run the mean-shift procedure
    $y \leftarrow y_{i,\,c}$
    if *convergence* **then**
      Set $z_i = (x_i^s, y_{i,\,c}^r)$ in $F_i(lin, col)$
    else
      Set $O_i(lin, col)$ in $F_i(lin, col)$
    end
  end
end

Note that the spatial band-width has a distinct effect on the output when compared to the range (color) band-width. Only large features are represented in the filtered image when h_s increases. On the other hand, only features with high color contrast survive when h_r is large (Comaniciu & Meer, 2002).

1.3.2 Seeded Region Growing Mean-Shift Segmentation

This work uses the mean-shift algorithm as a tool to automate the seed location, and its band-width as threshold for the region growing. The seeded region growing (SRG) algorithm is very attractive for semantic image segmentation by involving high-level knowledge of image components in the seed selection procedure. However, the SRG algorithm also suffers from the problems of pixel sorting orders for labeling and automatic seed selection. An obvious way to improve the SRG algorithm is to provide more effective pixel labeling techniques and automate the process of seed selection (Fan *et al.*, 2005).

The classical approach to the seeded region growing algorithm attempts to segment an image into regions with respect to a set of q seeds (Adams & Bischof, 1994). The pixels in the same region are labeled by the same symbol and the pixels in different regions are labeled by different symbols. All these labeled pixels are called the allocated pixels, and the others are called the unallocated pixels (Fan *et al.*, 2005). Let H be the set of all unallocated pixels which are adjacent to at least one of the labeled regions and $N(x, y)$ the second-order neighborhood of the pixel (x, y):

$$H = \left\{ (x,y) \notin \bigcup_{i=1}^{q} R_i \mid N(x,y) \cap \bigcup_{i=1}^{q} R_i \neq \emptyset \right\} \tag{1.22}$$

Note that each pixel $(x, y) \in H$, $N(x, y)$ meets just one labeled image region R_i. Departing from this definition it is possible to define a function $\phi(x, y, R_i) \in \{1, 2, ..., q\}$ that shows the difference between the testing pixel at (x, y) and its adjacent labeled region R_i. $\phi(x, y, R_i)$ can be calculated as:

$$\phi(x,y,R_i) = |g(x,y) - g(X_i^c, Y_i^c)| \tag{1.23}$$

Here, $g(x, y)$ indicates the values of the three color components of the testing pixel (x, y), $g(X_i^c, Y_i^c)$ represents the average values of three components of the homogeneous region R_i, with (X_i^c, Y_i^c) the centroid of R_i. If $N(x, y)$ meets two or more of the labeled regions, $\phi(x, y, R_i)$ takes a value of i such that $N(x, y)$ meets R_i and $\phi(x, y, R_i)$ is minimized:

$$\phi(x,y,R_i) = \min_{(x,y) \in H} \{\phi(x,y,R_j) \mid j \in \{1,2,...,q\}\} \tag{1.24}$$

The automatic determination of seed locations is an extremely important step to a good image segmentation, since the seeds are the basis of all procedures. A poor starting estimate of region seeds or bad pixel sorting orders may result in an incorrect segmentation of an image (Fan *et al.*, 2005). To overcome these problems, Mehnert & Jackway (1997) introduce an improved seeded region growing algorithm, where the region centers are just updated after all the labels have been determined. Fan *et al.* (2001) propose an algorithm by integrating color-edge extraction and seeded region growing on the YUV color space. There, edges in Y, U and V are detected by an isotropic edge detector and then the centroids between adjacent edge regions are taken as the initial seeds. (Fan *et al.*, 2005) proposes also a simple solution via regular seed generation, using the center of regular rectangular regions as the seeds. Shih & Cheng (2005) propose advances on the methodologies proposed in (Fan *et al.*, 2001, 2005) by applying a region-merging algorithm.

Comaniciu & Meer (1997, 2002) propose two methodologies for image segmentation using the mean-shift algorithm. Comaniciu & Meer (1997) present a technique in which all pixels are mapped into feature space and through a random process the homogeneous areas are found and a feature pallete is set. When there are enough colors in the palette, all pixels are allocated according to the minimal distance from a relevant color, forming the segments in this way. The problem here is that the random search always produces different segmentations from the same image, which turns this solution inviable. To overcome this issue, Comaniciu & Meer (2002) propose a different solution, where all pixels are filtered using the mean-shift filtering algorithm and then they are clustered, and the final clusters are the image segments. The results are quite interesting, but the segmentation becomes dependent of the clustering technique chosen. To overcome this clustering dependence, this work proposes the use of the SRG as the "clustering" step.

Relevant regions in aerial imagery are relatively homogeneous. Roofs and streets are the most significant examples, since they are easily identified by any human operator. Departing from this concept, one can affirm that these regions build clusters on the images themselves, and that it is not necessary to map the image in a 5D feature space to find them, since they are almost defined.

The mean-shift filtering turns the image more homogeneous and the segments become easier to identify, but their edges remain untouched. If a pixel, after the filtering, lies in a very homogeneous region, one can use this pixel as a seed for region growing. So, in this way, one can join the advantages of mean-shift filtering with the simplicity of the seeded region growing algorithm.

This work presents a hybrid solution to image segmentation, filtering the image and finding seeds via the mean-shift procedure, and then growing the regions. Another difference from the classic SRG is that the seeds are not a priori defined, but since a homogeneous region has been found, the region grows through a flood fill algorithm – see (Burger & Burge, 2007) – and just after that the algorithm searches for a new seed. The equation (1.23) is changed into equation (1.25), and the mean of the already found pixels are not used, but the *homogeneous region significant color*, (S_c) determined via the mean-shift procedure.

$$\phi(x, y, R_i) = |g(x, y) - S_c| \tag{1.25}$$

A pixel lies inside the region if $\phi(x, y, R_i)$ is smaller than the band-width used in the mean-shift procedure. A region is considered homogeneous if more than a user-defined percentage of the pixels lie inside the band-width. This value is called *Homogeneity Threshold* (H_T). The proposed algorithm runs as follows:

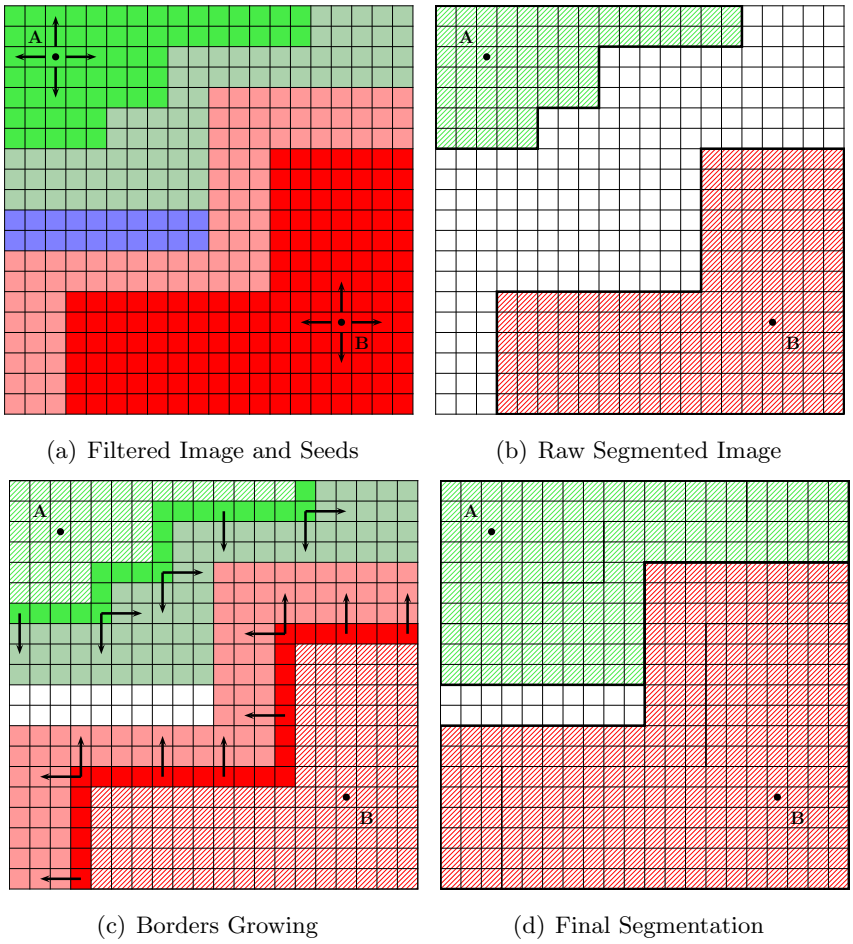
Algorithm 4: SRG Mean-Shift Segmentation

```
Data: Band Width (h_r, h_s), Original Image (O_i), Homogeneity
      Threshold (H_T)
Result: Segmented Image (S_i)
Run Mean-Shift Filtering
for col = 0 → ncol do
    for lin = 0 → nlin do
        c_(col,lin) ← count(pixels ∈ h_s ∩ h_r)
        p ← (c / h_r^2) × 100
        if p ≥ H_T then
            Run Mean-Shift Procedure
            Get S_c
            Run Flood Fill
            Get Region R
            for all pixels ∈ R do
                Set all found pixels to 0 on F_i
                Set S_c on S_i
            end
        end
    end
end
```

Segmentation Refinement

After the region growing processing, the segmentation is not complete yet. Many regions remain unsegmented, in the form of "black areas". It happens because pixels far away from the original seeds do not lie anymore inside the spectral band-width. This effect is more visible near to the borders of the segments, where the colors tend to be a mixture of the neighboring objects. This mixture can happen because of problems during the capture on the CCD, chromatic lense distortions or because of the compression method used by the camera manufacturer. This first result will be called *raw segmented image*.

Empiric observations on the raw segmented images show that these non-segmented areas still have, visually speaking, very similar colors to neighboring seed colors. The variations tend to be more in illumination than in the color components. Based on this facts, the proposed solution is simply to grow the regions from their borders in direction to the non-classified regions. Figure (1.10) shows the segmentation process.



(a) Filtered Image and Seeds (b) Raw Segmented Image

(c) Borders Growing (d) Final Segmentation

Figure 1.10: Segmentation Process

Since the non-classified region colors do not lie inside the band width of a seed color, the original color of the interface pixels (pixels which have unsegmented areas as neighbors) are used as new seed color to grow the segments. If a non-classified pixel lies on the band-width of an interface pixel, this pixel receives the color of the seed pixel. The non-classified regions tend to be small, then, in order to avoid excessive growing, a maximum growing distance M_d threshold is set. In other words, from an

interface pixel, the region can grow just up to a M_d distance from this pixel. This procedure is done for all interface pixels two times. In the first round, the segmentation band-width is used as color threshold and a large M_d is set (about 10 pixels). In the second, a higher color threshold is set, but M_d is smaller (about 3–5 pixels). All these thresholds can be freely set and depend on the camera type, and on which kind of image is being segmented.

Let figure 1.10(a) be an image after the mean-shift filtering. Two seed pixels (A and B) are found on this image and they grow to the raw segmented image on fig. 1.10(b). Note that there are more unsegmented pixels than acceptable, and that direct neighboring areas have similar color to the original seed, but lie outside the band-width. Using the interface pixels shown in figure 1.10(c), it is possible to grow beyond the original segments. Figure 1.10(d) shows the final image after the border growing. The blue region in figure 1.10(a) remains unsegmented, since no seed was found and the color is not similar to any other neighbors.

The last stage of the refinement is the elimination of small segments. This is done in two steps. First, all non-classified areas with less than a certain number of pixels are filled with the most abundant color in the neighborhood. At last, all colored areas with less than a minimum number of pixels are erased, and its color is substituted by the most similar color in the neighborhood.

1.4 Image Processing Overview

This section discusses the image processing implementation as a whole. Figure (1.11) shows the processing workflow.

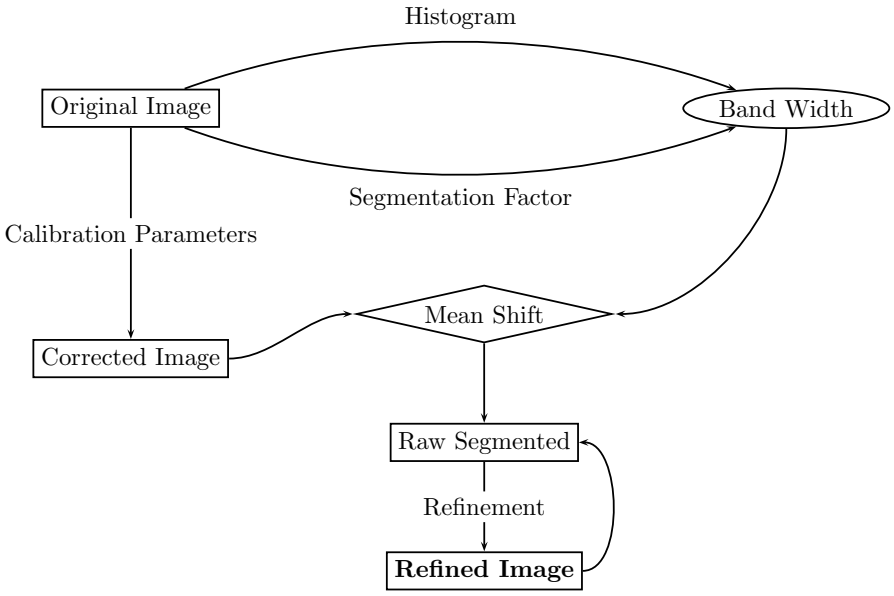


Figure 1.11: Image Processing Workflow

This workflow can be split into 3 main steps. The first step is the geometric correction of the original image, according to section (1.2.3). Figure (1.12) shows an example of the distortion free image generation. Figure 1.12(a) shows the original image and figure 1.12(b) shows the generated distortion free image. The image center corresponds exactly to the principal point. Looking the edges of the image, the effects of the distortions become remarkable.

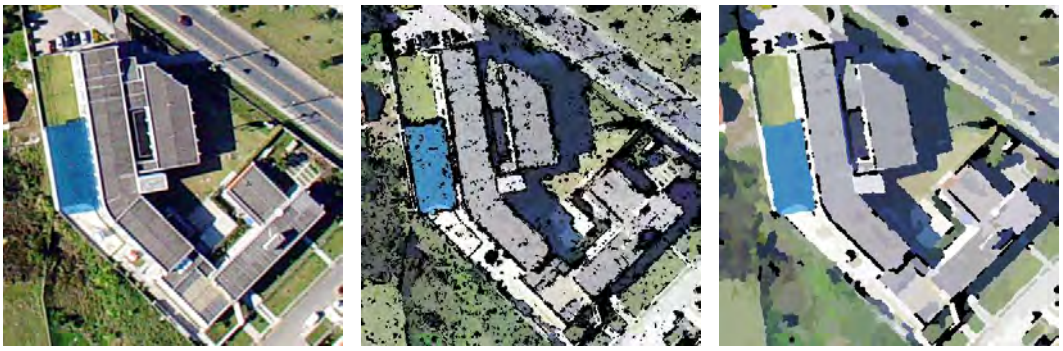
(a) Original Image (b) Distortion Free Image

Figure 1.12: Example of the Geometric Image Correction

The second step is to determine the band-width to be used in the further processing. Departing from the original image and its histogram, one can choose a segmentation factor as shown in table (1.3) and calculate the band-width. It is important to mention that the histogram is calculated on the basis of the image transformed to the LUV color space.

The third step is the image segmentation itself. The next figures show some results. Figures (1.13) and (1.14) present samples of a segmentation made from an image taken with the Sony DSC-F717 – see figure 1.1(a). The image has a GSD of about 25cm, and empirical tests show that a segmentation factor of 0.5 delivers best results. The spatial band-width is a 17x17 pixels window.

Figure (1.13) shows a complex building. Some areas stay unsegmented, but the overall result is good. Note that the most homogeneous roofs tend to show a better segmentation result. The small blue roof exemplifies this observation.



(a) Original Image (b) Raw Segmented (c) Refined

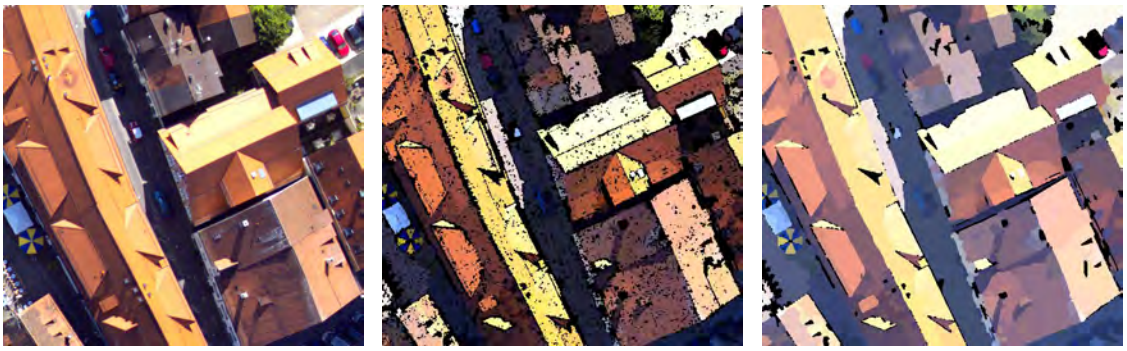
Figure 1.13: Segmentation Example 1

Figure (1.14) shows a high isolated building and a parking place. Note how the shadows of the building project and how the segmentation in this area are worse than in well illuminated parts. This is one of the most relevant problems in segmentation of aerial imagery. The well–illuminated roofs show a little oversegmented. The cars on the parking place as well some borders of the building remain unsegmented. It happens because no seed could be defined in these areas, using a 17 pixel spatial band–width.

(a) Original Image (b) Raw Segmented (c) Refined

Figure 1.14: Segmentation Example 2

Figures (1.15) and (1.16) present samples of a segmentation made from an image taken with the Applanix DSS – see figure 1.2(a). The image was taken over the downtown of Biberach, Germany, with a GSD of about 8cm. The segmentation factor was also 0.5 and the spatial band–width 19 pixels. Figure 1.15 shows a very complex set of roofs. Note that the illumination plays, again, a very important role in the segmentation quality.



(a) Original Image (b) Raw Segmented (c) Refined

Figure 1.15: Segmentation Example 3

Figure (1.16) shows a well illuminated and isolated building. Due to almost ideal conditions, the segmentation shows very good. Note how the borders grow over unsegmented areas after the raw segmentation step. Also the trees on the left side of the building show good results.

(a) Original Image (b) Raw Segmented (c) Refined

Figure 1.16: Segmentation Example 4

Chapter 2

LIDAR: A Powerful GIS Data Source

2.1 Introduction

Due to recent advances in Light Detection and Ranging (LIDAR) technologies, the acquisition of location an height information using laser scanners and other such equipments has become quite popular. Many companies utilize such LIDAR techniques to create three-dimensional point datasets in a variety of applications including city and object three-dimensional modeling. Since such point datasets are usually dense, they are referred as *point clouds*. In this three-dimensional modeling framework, the acquired three-dimensional datasets tend to be relatively large, typically in the order of hundreds of thousands of points per scan (Kothuri *et al.*, 2007). In order to deal with this amount of data, many companies developed systems based on a binary file extension (.las), which speeds up the access to the points in the cloud. The problem here is the integration of these datasets with other data sources, like vector maps and imagery. To overcome this issue, an option is to integrate the LIDAR data into GIS systems, since they offer several tools to deal with large datasets. This tendency can be proved by the LIDAR data tools newly included on the Oracle Spatial, the most popular commercial geodatabase on the market.

This chapter treats LIDAR data processing as a preprocessing step for the data fusion presented in chapter 3. An overview of relevant aspects on GIS and LIDAR technology is presented, as well a solution for LIDAR data storing and accessing using the PostgreSQL open source database. Furthermore, main aspects on LIDAR filtering, DTM and DSM generation are presented. At last, a height-texture based DSM segmentation methodology is proposed.

2.2 GIS Overview

Accordingly to Longley *et al.* (2005), *"the field of geographic information systems (GIS) is concerned with the description, explanation, and prediction of patterns and processes at geographic scales. GIS is a science, a technology, a discipline, and an applied problem solving methodology"*. Over the past decade, GIS have evolved from a highly specialized niche to a technology that affects nearly every aspect of our lives, from finding driving directions to managing natural disasters. While a few years ago the use of GIS was restricted to a group of researchers, planners and government workers, now almost everybody can create customized maps or overlay GIS data. On the other hand, many complex problems related to urban and regional planning, environmental protection, or business management,

require sophisticated tools and spatial expertise. Therefore the current GIS technology spans a wide range of applications from visualization to spatial analysis, modeling and simulations (Neteler & Mitasova, 2008).

As tool or as an information system, GIS technology has changed the entire approach to spatial data analysis. GIS has already been compared to not one but several simultaneous revolutionary changes in the way that data can be managed. The convergence of GIS with allied technologies, those of surveying, Remote Sensing, Photogrammetry, GPS, and mobile computing and communications has fed a spectacular growth of these technologies (Clarke, 2003).

Two methods are used to reduce geographic phenomena to forms that can be coded in computer databases, and we call these raster and vector. Accordingly to Longley *et al.* (2005), a raster representation space is divided into an array of rectangular cells, called pixels. All geographic information is then expressed by assigning properties or attributes to these pixels. Burrough & Mcdonnell (1998) show that in a vector structure, the units are represented as a crisp world objects using a coordinate space that is assumed to be continuous, not quantized as with the raster structure, allowing all positions, lengths and dimensions to be defined precisely . The relative merits of both systems have been summed up by Burrough & Mcdonnell (1998) as follows:

Raster Data Structures

Advantages

- Simple data structures;
- Location-specific manipulation of attribute data is easy;
- Many kinds of spatial analysis and filtering may be used;
- Easier mathematical modeling because spatial entities have a simple, regular shape;
- The technology is cheap;
- Many forms of data are available;

Disadvantages

- Large data volumes;
- Using large grid cells to reduce data volumes reduces spatial resolution, result in loss of information and an inability to recognize phenomenoligistically defined structures;
- Coordinate transformations are difficult and time consuming unless special algorithm and hardware are used and even that may result in loss of information or distortion of grid cell shape;

Vector Data Structures

Advantages

- Good representation of entity data models;
- Compact data structure;
- Topology can be described explicitly – therefore good for network analysis;
- Coordinate transformation and rubber sheeting is easy;
- Accurate graphic representation at all scales;
- Retrieval, updating and generalization of graphics and attributes are possible.

Disadvantages

- Combining several polygon networks by intersection and overlay may be difficult;
- Spatial analysis within basic units such as polygons is impossible without extra data because they are considered to be internally homogeneous;
- Simulation modeling of processes of spatial interaction over paths not defined by explicit topology is more difficult than with raster structures because each spatial entity has different shape and form.

Accordingly to Neteler & Mitasova (2008), GIS can be implemented as a comprehensive, multipurpose system; as a specialized, application oriented tool; or as a subsystem of a large software package supporting handling of geospatial data needed in its applications. GIS functionality is rapidly evolving and currently covers a wide range of areas, for example:

- Integration of geospatial data from various sources: projections and coordinate transformations, format conversions, spatial interpolation, transformation between data models;
- Visualization and communication of digital georeferenced data in form of digital and paper maps, animations, virtual reality (computer cartography);
- Spatial analysis: spatial query, spatial overlay (combination of spatial data to find locations with given properties), neighborhood operations, geostatistics and spatial statistics;
- Network analysis and optimization;
- Simulation of spatial processes: socioeconomic such as transportation, urban growth, population migration, as well as physical and biological, such as water and pollutant flow, ecosystem evolution, etc.;
- Image processing: satellite and airborne image processing, Remote Sensing applications.

In LIDAR applications both structures are used comprehensively. Some systems and application solutions are based on the raster structure, others on the point cloud. Data storage and processing systems have not been standardized, therefore, a lot of solutions exist and probably will be applied also in the future. Some process developers use raster data, because its advantages are favored, but on the other hand, some developers are committed to point cloud and TIN (Triangular Irregular Network). However TIN models have a better ability to describe precisely the surface, but, because TIN is based on the original points, it can not be smoothed. Raster has a smoother, more natural appearance and, for this reason raster is used usually for visualization, even if the computation uses the original point cloud. In TIN structure, the point density is variable, while in raster it is fixed. Due to the regular structure, simple computation processes can be used on raster and more complicated algorithms are necessary in the case of TIN. TIN structure is able to represent 3D models, while raster is limited to 2,5D. 3D models are important to represent surface overhangs. These surface elements can be described in raster data only with geometrical restrictions (Tovari, 2006). Both data structures can be converted easily to the other. Raster heights can be interpolated and stored within triangles in TIN structure. In raster to TIN conversion, each raster may be considered as a node point in the triangulation. Using a GIS approach to store and process LIDAR data gives one the option to use the advantages of both structures. For this reason, this work uses GIS tools to deal with LIDAR data. Section 2.4.2 discusses this question more in depth.

2.2.1 GIS and Databases

Per definition, GIS is built on the foundation of a geographic database. After people, the database is arguably the most important part of a GIS because of costs of collection and maintenance, and because the database forms the basis of all queries, analysis and decision making. Data storage within a GIS has historically been an issue of both space – usually how much disk space the system requires – and access, or how flexible a GIS is in terms of making the data available for use (Clarke, 2003). Today, all important GIS implementations store data in a database management system (DBMS), a specialist piece of software designed to handle multi-user access to an integrated set of data. Extending

standard DBMS to store geographic data raises several interesting challenges. Databases need to be designed with great care, and to be structured and indexed to provide efficient query and transaction performance (Longley *et al.*, 2005).

Data in a GIS database provide a simplified, digital representation of earth features for a given region. Georeferenced data can be organized within GIS using different criteria, for example, as thematic layers or spatial objects. Each thematic layer can be stored using an appropriate data model depending on the source of data and their potential use (Neteler & Mitasova, 2008). Building an accurate GIS database of spatial entities is an exacting task. Raw geographical data are available in many different analogue or digital forms, such as maps, aerial photographs, satellite images or tables. There are three, not mutually exclusive ways to create a digital geographical database: (a) acquire data in digital form from a data supplier, (b) digitize existing analogue data, and (c) carry out one's own digital survey. In all cases the data must be geometrically registered to a generally accepted and properly defined coordinate system (Burrough & Mcdonnell, 1998).

Longley *et al.* (2005) list the main advantages of the database approach to storing geographic data:

- Assembling all data at a single location (redundancy reduction);
- Maintenance costs decrease because of better organization and reduced data duplication;
- Applications become data independent so that multiple applications can use the same data and can evolve separately over time;
- Security and standards for data and data access can be established and enforced;
- DBMS are better suited to managing large numbers of concurrent users working with vast amounts of data.

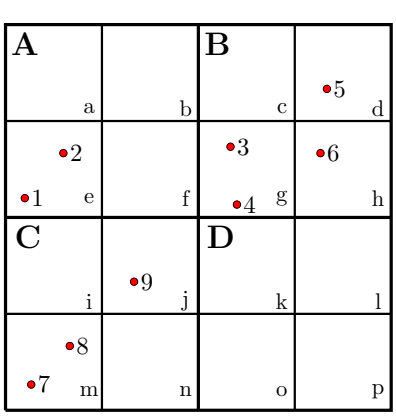
2.2.2 Clustering and Indexing Geodatabases

Geographic databases tend to be very large and geographic queries computationally expensive (Longley *et al.*, 2005). The vector data model is a relatively efficient means of storing geometric information of geographical data with only pertinent coordinate values recorded (see 2.2, page 48). The main problems have been associated with accessing the data (Burrough & Mcdonnell, 1998). Because of this, geographic queries can take a very long time (Longley *et al.*, 2005). The first attempts at improving database access times used 'brute-force' computing methods to scan the pointer arrays quickly, or to concentrate the master index array onto a small, contiguous area of disc or core storage (Burrough & Mcdonnell, 1998).

A more efficient way to speed up queries is to index a database and use the index to find data records (database table rows). A database index is, conceptually speaking, an ordered list derived from the data in a table. Using an index to find data reduces the number of computational tests that have to be performed to locate a given set of records. In DBMS jargon, indexes avoid expensive full-table scans by creating an index and storing it as a table column. A database index is a special representation of information about objects that improves searching. Three main methods of general practical importance have emerged in GIS: grid indexes, quadtrees and R-trees (Longley *et al.*, 2005). This work uses the grid index implemented upon PostGIS – **gist** – in order to improve the search for LIDAR points in the cloud.

A grid index can be thought of as a regular mesh placed over a layer of geographic objects. Figure (2.1) shows an example with 9 LIDAR indexed in two grid levels. The highest grid (Index 1) splits

the data set into four equal sized cells. Cell A includes the points 1 and 2, cell B the points 3, 4, 5 and 6; cell C 7, 8 and 9 and cell D has no points. The same process is repeated for the second level index (Index 2). A query to locate an object searches the indexed list first to find the object and then retrieves the object geometry or attributes for further analysis. These two tests are referred to as primary and secondary filters (Longley *et al.*, 2005).



Index 1	
A	1,2
B	3,4,5,6
C	7,8,9

Index 2	
d	5
e	1,2
g	3,4
h	6
j	9
m	7,8

Figure 2.1: A Multi-level Grid Geodatabase Index

The performance of an index is clearly related to the relationship between grid and object size, and object density. Grid indexes are one of the simplest and most robust indexing methods. They are fast to create and update and can handle a wide range of types and densities of data. For this reason they are widely used in GIS software systems (Longley *et al.*, 2005).

2.2.3 PostgreSQL, PostGIS and LIDAR Data

High resolution digital terrain models can be obtained using LIDAR data. There are many applications requiring such models, both civilian and military. Visual simulation and other types of 3D-visualizations are perhaps the most prominent ones due to the growth of easy accessible powerful 3D-computer graphics hardware. However, there are many other important applications, e.g. urban planning, command and control, mission planning and preparation and various terrain analysis problems. To support these applications development of new methods and algorithms for automatic terrain modeling, terrain feature analysis and databases are needed. Since data acquisition using airborne laser scanners usually entail huge data sets even for moderate areas it is important that computational efficiency, efficient storage and data access are considered (Elmqvist *et al.*, 2001).

This work uses the advantages of storing and accessing large LIDAR data sets in a geodatabase environment. The PostgreSQL was chosen as database for this task. The reasons are a very good performance, since PostreSQL is, nowadays, the most advanced open-source database, and the fact that it is a free open-source solution. Another reason to use this approach is the geographical extension for PostgreSQL, the PostGIS. This extension offers the most important functions for any GIS task, as grided indexing, clustering and many topological query functions. Since PostGIS offers a very reliable set of tools for working on vector data, all the vectorial data storage an managing in this work is done within the PostGIS environment.

2.3 LIDAR Operation Overview

LIDAR is a scanning and ranging method, which produces three-dimensional, highly accurate information and very high-resolution topographic models by direct measurement. The technology is also called Light Detection and Ranging (LIDAR). LIDAR has revolutionized both topographic and close range three-dimensional object recording. Particularly in analysis, there is much in common between laser scanner and Photogrammetry (Kraus, 2007). Using a laser scanner, points on the ground are sampled. With the aid of a narrow laser beam, a pulse of laser light from the scanner is diffusely reflected by a point on the ground surface. From the elapsed time between transmission and reception of the pulse, the distance between scanner and ground point can be determined. The laser beam in the laser scanner is deflected at right angles to the direction of flight and this angle of deflection is recordered (Kraus, 2007). Sometimes more than one echo is reflected back from different objects (e.g. tree-crown and ground), that's why some of the instruments can detect the second or further echoes as well. The principle of laser scanner can be seen on figure (2.2). This figure shows the scanner in 2 strips. Note that the laser beam crosses objects like trees and more than one echo can be recordered.

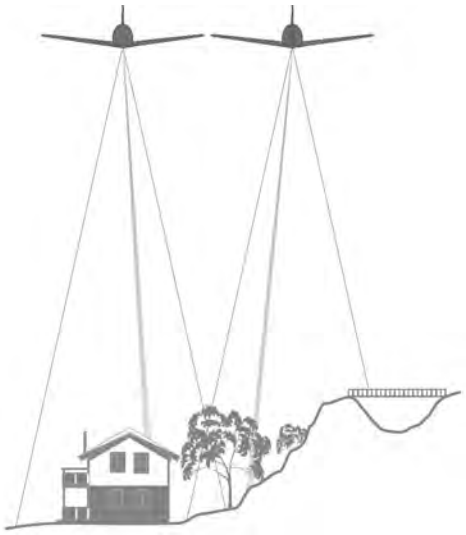


Figure 2.2: LIDAR Profiling (Sithole, 2005)

The coordinates of the laser scanner location and its orientation angles are required in order to convert the polar coordinates of the measured object point into (X, Y, Z) coordinates. These constantly changing values are determined by means of a dynamic POS (Position and Orientation System), consisting of GPS and an IMU (Kraus, 2007). The 6 orientation parameters of the laser scanner sensor $(\kappa, \varphi, \omega, X_0, Y_0, Z_0)$ are given by the orientation functions $\kappa(t), \varphi(t), \omega(t), X_0(t), Y_0(t), Z_0(t)$ and are determined using the GPS and IMU information an related to the corresponding polar coordinates $\alpha(t)$ and $s(t)$ through the synchronization time t. The resulting (X, Y, Z) coordinates of individual laser points are calculated in the object space with the eq. 2.3. Note that these equations are the same used for optic-mechanical scanners (Bähr, 1976).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0(t) \\ Y_0(t) \\ Z_0(t) \end{bmatrix} + \mathbf{R}_{\kappa(t)\cdot\varphi(t)\cdot\omega(t)} \begin{bmatrix} 0 \\ s(t)sin\alpha(t) \\ s(t)cos\alpha(t) \end{bmatrix} \tag{2.1}$$

2.4 Digital Terrain Model (DTM) Generation

Highly detailed and accurate terrain data is one of the most critical components of many mapping, engineering and natural resource management projects. The demand for high quality DTMs is increasing significantly as the GIS community advances toward 3-D technology and virtual-reality environments. Many geospatial applications, such as urban planning, landscape analysis, transportation and hydrological watershed analysis, need to use DTMs. Data used for DTM production include aerial and satellite images, IfSAR data, GIS data, and LIDAR data (Hu, 2003). To generate a DTM from LIDAR data, one has to remove points falling on above ground features, and to interpolate between the remaining terrain points falling on the bare ground surface. This is called LIDAR filtering. Then, using the bare ground points, one can interpolate the DTM. This work uses a DTM to segment a laser DSM (section 2.5). This section presents an overview on LIDAR filtering and DTM generation.

2.4.1 LIDAR Filtering

LIDAR filtering is a classification process to distinguish on-terrain points from off-terrain points within a cloud of LIDAR measurements. At the end of filtering, the DTM can be constructed by either removing off-terrain points or detecting on-terrain points (Sohn & Dowman, 2008). A number of algorithms have been reported in the literature, but most of them are not yet proved competent for industry production because of the complexity of the task and the need of much manual editing (Hu, 2003). Some algorithms work over DSM (see sec. 2.5), and others do it directly over the point clouds. This section presents a brief overview of the state-of-art on LIDAR filtering, as well presents a more detailed description of the chosen methodology.

Sithole (2005) conducted a study to evaluate the performance of several filtering algorithms. Seven characteristics, including data structure, test neighborhood, discontinuity, filter concept, single vs. iterative processing, replacement vs. culling and use of first pulse and reflectance data, are used to understand the behavior of those filter algorithms. The filtering results are compared against reference data that are generated by manually filtering raw lidar data. It is found that in general the filters perform well in landscapes of low complexity. However, complex landscapes as can be found in city areas and discontinuities in bare ground surfaces still pose challenges. It is suggested that future research be directed at heuristic classification of point clouds based on external data, quality reporting, improving the efficiency of filter strategies.

Hansen & Vögtle (1999) and Vögtle & Steinle (2003) developed a so called convex concave hull approach. First, a convex hull is set upwards with a triangulation method (e.g. Delaunay) to the data. In this process, the locally lowest points are selected, which are most probably ground points. These points are triangulated so that no points lie below the triangles. For each triangle of the convex hull, new points will be added that are located within the triangle and full certain criteria. The threshold depends on the size of the triangle, namely on its longest side. Other criteria can be applied as well, like maximum curvature. When a new point is accepted, the old triangle is divided and triangulated taking into account the new point. Triangles are densified until no additional points can be joined to them. This densification approximates in every step a more detailed terrain surface.

Based on the fact that laser footprints often are on the treetops in wooded areas, Kraus & Pfeifer (1998) formulated an asymmetric weight function by statistical analysis to filter the lidar data. The algorithm is based on the linear prediction with an individual accuracy for each point, and works

iteratively. The surface is firstly computed with equal weights for the elevation values of all points, and thus runs in an averaging way between terrain points and vegetation points. Then the weights are re-calculated based on the residuals relative to the surface, and are used for the next computation of the surface.

Petzold *et al.* (1999) proposed a filtering algorithm to distinguish points situated on buildings and on the vegetation from those expected to be on the ground. First, a rough terrain model is calculated using the lowest points found in a moving window of a rather large size that is based on the largest building size in region of interest. Then all points with a height difference exceeding a given threshold are filtered out and a more precise DTM is calculated. This step is repeated several times, reducing the window size and leading to the final DTM.

Masaharu & Ohtsubo (2002) developed a two-stage filtering method suitable for highly developed urban areas. The primary selection use the lowest point in regularly divided patches of the area, and a secondary selection is recursively applied to these points to remove points on building roofs and at the bottom of underground tunnels that may be remained in the primary selection. The secondary selection is based on whether the points in the primary selection are within one sigma from the mean of the neighboring lowest points of patches. If the point is judged out of the range by this statistical test, it is removed.

Vosselman (2000) and (Sithole, 2001) developed a slope-based filter, which is proved to be equivalent to the erosion operator in mathematical morphology. In this approach, the ground is defined as points within a given slope range. The thresholds are determined by a stochastic approach that needs training.

Sohn & Dowman (2008) developed an algorithm in which the lidar DSM is convolved with heterogeneous terrain slopes, and then fragmented into a set of homogeneous sub-regions, within which underlying terrain is characterized with a single terrain slope. Based upon this irregular terrain fragmentation, a lidar filtering technique, called recursive terrain fragmentation (RTF), was developed. The RTF filter employed an elementary terrain model for the reconstruction of a generic terrain surface. That elementary model is a planar terrain surface, which comprises on-terrain points with the same slope. The RTF filter reconstructs the DTM by obtaining plane terrain surfaces hidden in a cloud of lidar points. In a similar way to a deterministic filter, a single filtering criterion is adopted in order to differentiate on-terrain points from off-terrain ones, since the terrain favored by the RTF filter is simply modeled as a plane terrain surface.

Meng *et al.* (2009) presented a Multi-directional Ground Filtering (MGF) algorithm that combines advantages of the directional and neighborhood-based scanning. This technique explored the utility of identifying a variety of patterns in different directions across an image. The proposed MGF algorithm considers the slopes for neighboring pixels in up to four directions and the elevation difference between a pixel and the local minimum elevation within a two-dimensional and the nearest ground pixel.

All approaches cited above show strengths and weaknesses. LIDAR filtering is still an open research field, and no methodology can be defined as "standard". This work does not intent to make developments in this field, and for this reason an already implemented solution was chosen. The option is to use the solution from GRASS – GIS. This software applies the algorithm developed by Brovelli *et al.* (2004). This is a filtering algorithm based on spline interpolation and region growing techniques. It was designed for processing LIDAR data in urban areas, and shows good results. The next section presents the methodology.

Brovelli Algorithm

Viewed planimetrically – see figure (2.3) – objects in a landscape standout from the background (bare ground) by the fact that they have distinct edges that together form a closed boundary. Therefore, points within the closed boundaries are accepted as being part of an object. This is the concept of an *edge based filtering algorithm*, and it is used in the filter designed by Brovelli *et al.* (2004). This section presents the most important aspects of the algorithm proposed in this paper.

The first step of the algorithm, called spline step – S_0 – depends on the planimetric resolution of the raw data. A Tikhonov regularization parameter (λ_0) is introduced to avoid local and global singularity in the least square approach (in case of zones where observations are missing), and in order to assure the regularity of the surface, minimizing the curvature in empty areas. Imposing a high value for (λ_0) produces a surface with a behavior quite different from that obtained with an exact interpolator: the surface feels as little as possible the influence of possible outliers. Data corresponding to residuals exceeding a threshold (T_0) are considered as outliers and removed. Data are also automatically subdivided in tiles to avoid computational problems. This step allows the detection of the edges of the surface objects: an edge is a boundary between two different regions, i.e. a significant change in the height value corresponding to a small shift of the horizontal position.

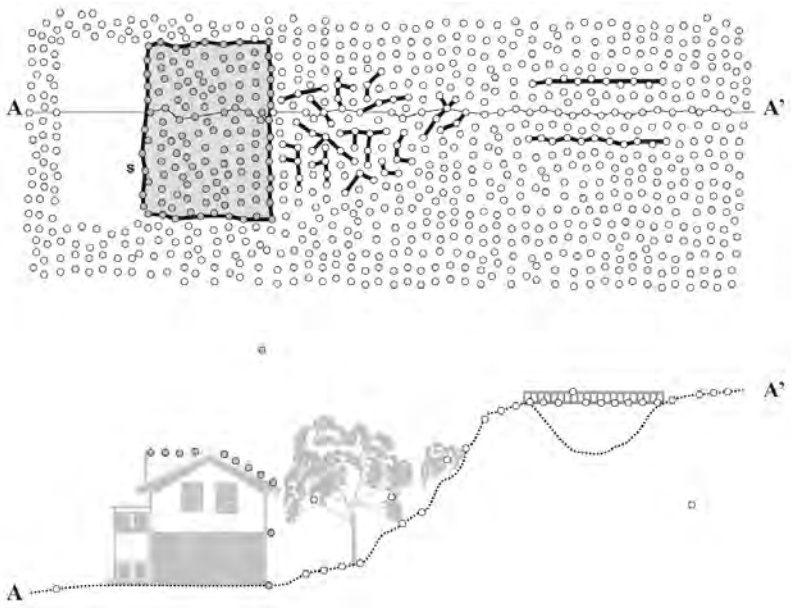


Figure 2.3: Brovelli Algorithm Concept (Sithole, 2005)

The implementation of an algorithm to detect edges is complicated by the non-regularity of the distribution of the observations. Two approximations of the DSM are then computed by means of bilinear (spline step S_g) and bicubic (spline step S_r) spline functions with Tikhonov regularization in a least squares approach. Theoretical considerations suggest that these surfaces should regularized in order to minimize their gradient and curvature. In the first case a low regularization parameter (λ_g) brings the interpolating functions as close as possible to the observations, whereas in the second one the choice of a high value for λ_r gives a rough and loose-fitting surface. Starting from the bilinear spline surface, where $\hat{a}, \hat{b}, \hat{c}, \hat{d}$ are the least square coefficients:

$$z(x, y) = \hat{a}x + \hat{b}y + \hat{c}xy + \hat{d} \tag{2.2}$$

The gradient magnitude G_m is computed as follows:

$$G_m = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2} = \sqrt{(\hat{a} + \hat{c}y)^2 + (\hat{b} + \hat{c}x)^2} \tag{2.3}$$

The imposition of a unique threshold to the gradient magnitude is not suitable because if we choose a low value we will not discriminate between an actual edge and possible measurement noise and if we choose a high value we will detect only very sharp height changes. The basic hypothesis is that noise corresponds mostly to an isolated observation or, at least, adjacent noises are generally not organized in a regular shape. In contrast, an edge shows a regular, chain-like behavior (see fig. 2.3). The computation of the direction of the edge vector (ϑ_P) can strengthen the classification:

$$\vartheta_P = \arctan\left(\frac{G_y}{G_x}\right) + \frac{\pi}{2} = \arctan\left(\frac{\hat{b} + \hat{c}x}{\hat{a} + \hat{c}y}\right) + \frac{\pi}{2} \tag{2.4}$$

Two thresholds for the magnitude gradient, the high (T_g) and low (t_g) thresholds, are set. Every point P where the magnitude gradient exceeds (T_g) is classified as a possible edge point. For every point where the magnitude gradient is lower than (T_g) but exceeds (t_g) we find, along the direction of the maximum direction of gradient rise (perpendicular to the direction of the edge vector), the two neighboring and opposite points P_1 and P_2. If , for a given threshold ϑ_g, these points have the same edge direction of P ($|\vartheta_{P_1} - \vartheta_P| < \vartheta_g, |\vartheta_{P_2} - \vartheta_P| < \vartheta_g$) and if the magnitude gradient for the eight nearest neighboring points exceeds T_g in at least two instances, the point will be classified as a possible edge point. In other cases it is classified as non-edge point.

Once the edges have been detected, the next step is to fill-in the objects they limit. The simplest idea is that the inner part of an object has generally a greater height than its edges. But this consideration cannot be true for vegetation, in some cases for buildings with particular types of roofs and because of the presence of random noises in the observations. Furthermore the edges, due to classification errors, sometimes do not represent closed lines. Some tests have to be added before we apply a region growing algorithm. Using the mean height value of the points within each cell the data are rasterized with a resolution r_d equal to the minimum data raw density. For each cell the presence of points with double pulse is evaluated (difference between first and last pulse greater than T_g. Starting from the cells classified as edges and with only one pulse, all the linked cells are found and a convex hull algorithm is applied on them, computing at the same time the mean of the corresponding heights (mean edge height). The points inside the convex hull are classified as objects in case their height is greater or equal to the previously mean computed edge height.

The algorithm fails in some cases: the simplest we can recall is the case where we have unusual roofs with pitches at different heights. In this example, part of the building was identified as object and part as terrain. Similar cases of terrain misclassification in the procedure output remain. To overcome these problems, a bilinear interpolation (spline step S_c) with Tikhonov regularizing parameter λ_c only on the points classified as ground has been performed. The analysis of the residuals (Δ) between the observations and the interpolated values compared with two thresholds t_c, T_c show four cases:

- if P is classified as ground and $\Delta > T_c$, it will be reclassified as object;
- if P is classified as double pulse ground and $\Delta > T_c$, it will be reclassified as double pulse object (edge or vegetation);
- if P is classified as object and $|\Delta| < t_c$, it will be reclassified as ground;
- if P is classified as double pulse object and $|\Delta| < t_c$, it will be reclassified as double pulse ground.

The procedure can be iterated multiple times until visual analysis indicates that the ambiguous cases are definitely solved or reduced. Using the DSM as auxiliary tool, a final manual editing is always necessary. Figure (2.4) shows results of the procedure. The most significant problems are ground areas with significant slope, where the algorithm classifies ground areas as objects, but for the use on this work, it does not affect the final results significatively, since after the DTM interpolation the height values will be similar to the real ones. The used methodology works well on dense urban areas (see upper left corner). Nevertheless, in order to obtain the shown results, some manual edition was necessary.

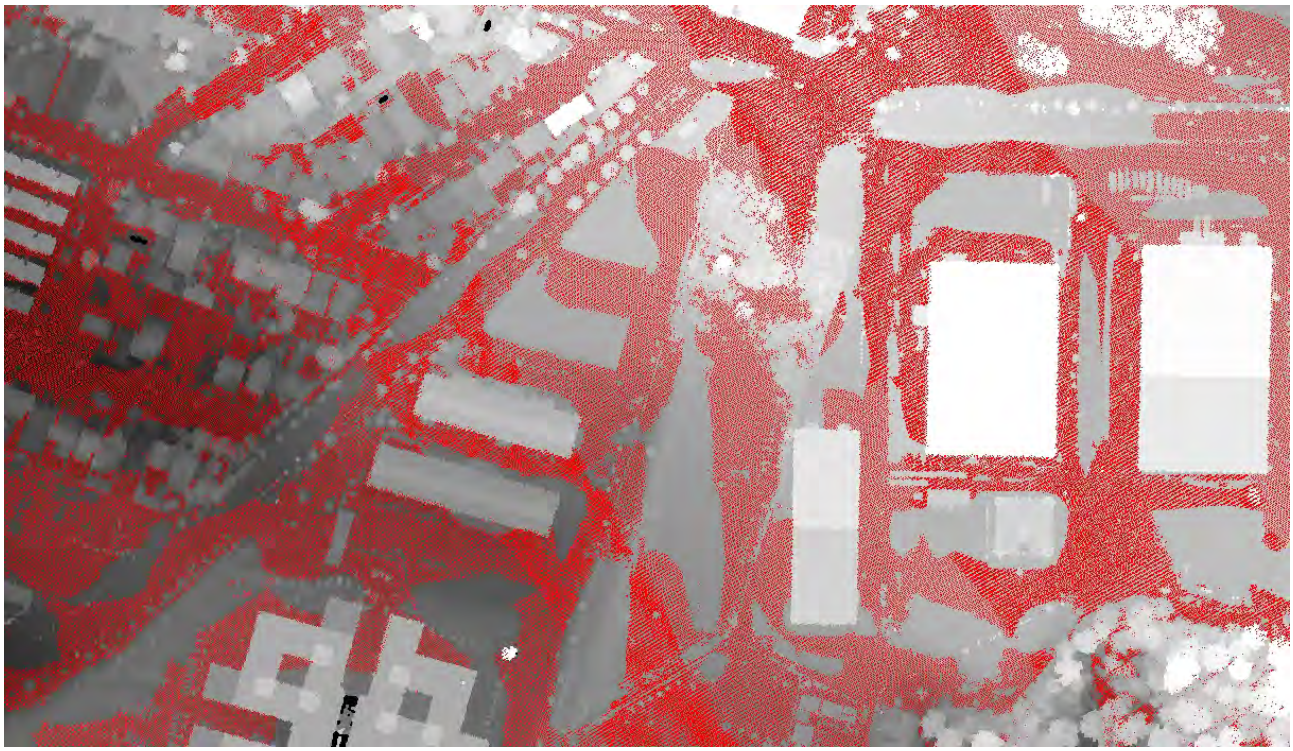


Figure 2.4: Example of the LIDAR Filtering

2.4.2 DTM Interpolation

Interpolation is the procedure of predicting the value of attributes at unsampled sites from measurements made at a point locations within the same area or region. This technique is used to convert data from point observations to continuous fields so that the spatial patterns sampled bay these measurements can be compared with the spatial patterns of other spatial entities. A Digital Terrain Model (DTM) is a special case of continuous surface created by interpolation . The variation of surface elevation can be modeled in many ways. DTMs can be represented either by mathematically defined surfaces or by point or line images. Line data can be used to represent contours and profiles. In GIS, DTMs are modeled by regular grids (altitude matrices) and TINs (Burrough & Mcdonnell, 1998).

Altitude matrices are the most common form of discretized elevation surface. Because of the easy with which matrices can be handled in the computer, in particular in raster based geographical information systems, the altitude matrix has become the most available form of DTM. They are the starting point for deriving much useful information about landform, such as slope, profile convexity, solar irradiance, lines of sight and surface topology (Burrough & Mcdonnell, 1998).

TIN was first presented by Fowler *et al.* (1978) as a digital terrain modeling that avoids the redundance of the altitude matrix and which at the same time would also be more efficient for many types of computation. A TIN is a terrain model that uses a sheet of of continuous, connected triangular facets based e.g. on a Delaunay triangulation of irregularly spaced nodes or observations points. TINs are also used to produce maps of slope, shaded relief, contour maps, profiles, horizons, block diagrams and line of sight maps (Burrough & Mcdonnell, 1998).

This work uses both structures to compose the DTM. First, all ground points from the LIDAR filtering are set into a GIS software (Spring - from INPE, Brazil). Then the Delaunay triangulation is computed. The next step is to perform a linear interpolation inside the triangles to obtain a regular grid. At the and the regular grid is exported as a raster file, to be used on the DSM segmentation.

2.5 Digital Surface Model (DSM) Generation and Segmentation

A DSM is a large matrix of altitudes. Each matrix element, or pixel, has a value representing a height value (z). A DSM provides geometric information about objects independent of their position, direction and intensity of light sources illuminating the scene or of reflectance properties. For these reasons, the DSM plays an important role in image understanding, three-dimensional object reconstruction, autonomous navigation, etc.

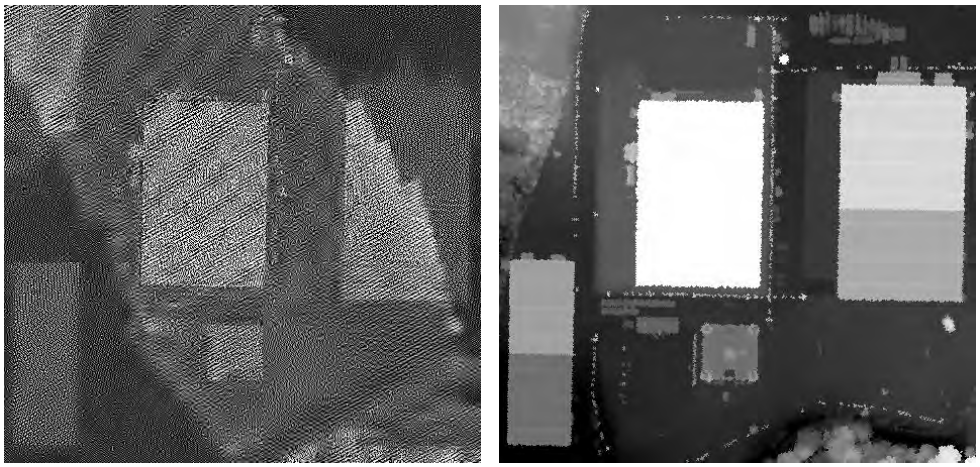
2.5.1 DSM Generation: Modified Araki Algorithm

The DSM creation is a very important step in many works with LIDAR data. Objective of this procedure is to create a regular grid of points departing from an irregular point cloud. The further processing is highly dependent on this procedure. DSM can be used to create DTMs and true orthophotos, for example. This work needs a DSM optimized for extraction of vegetation and buildings. It means that information concerning these objects must be preserved as realistically as possible. Araki (2005) proposes a methodology for DSM generation focused on true orthoimages generation from high-resolution satellite imagery. This methodology has the property of keeping buildings edges as sharp as possible, since no polynomial interpolation is used. The original Araki algorithm can be summarized in four steps:

1. Grid: The laser points are set to a grid;
2. Homogeneity Criterion: The non-filled pixels are set based on a homogeneity criterion. If the height difference between the lowest and highest point inside the search window is smaller than a homogeneity threshold H_t, the pixel is filled with the average of the neighboring pixels;
3. Closing Operator: The morphologic operator "closing" is applied on the image to close pixels still without data;
4. Final Treatment: The remaining regions without data, as occlusions and lakes are treated.

The methodology shows good results for its original purposes, but for this work some adaptations were necessary. The main reason for using a version of the Araki algorithm is the conservation of buildings borders, since it is a main issue in DSM segmentation (section 2.5.2) and classification (section 3.2.4). Two problems of this approach are that taking the average of neighbors at the step 2, and applying the closing operator for all pixels on step 3, creates a smooth image. This smoothness makes the roughness based segmentation splitting (section 2.6) much more difficult. To overcome these problems, a *modified Araki algorithm* is proposed:

1. Grid: Every pixel represents a small quadratic area on the object space. This area can be represented by a polygon. The modified algorithm searches for all points within each polygon, and if more than one point is found, the highest is chosen. This result is called the "high-image";
2. Homogeneity Criterion: The empty pixels are also filled with an homogeneity criterion. The difference is that the modified algorithm does not calculate the average, but simply takes the highest point of the neighborhood;
3. Closing Operator: The morphologic operator "closing" is applied on the image to close pixels still without data, but just on these pixels. The areas already filled remain untouched;
4. Final Treatment: If there is still any region without data, they are simply ignored. No final treatment is done.



(a) High Image (Step 1) (b) DSM (Step 4)

Figure 2.5: DSM Generation Example

Figure (2.5) shows how the modified Araki algorithm works. This example shows an image sample with 0.5m GSD. Note the high number of "empty" pixels in figure 2.5(a) and how the LIDAR data density varies due to overlapping strips. Figure 2.5(b) shows the final result.

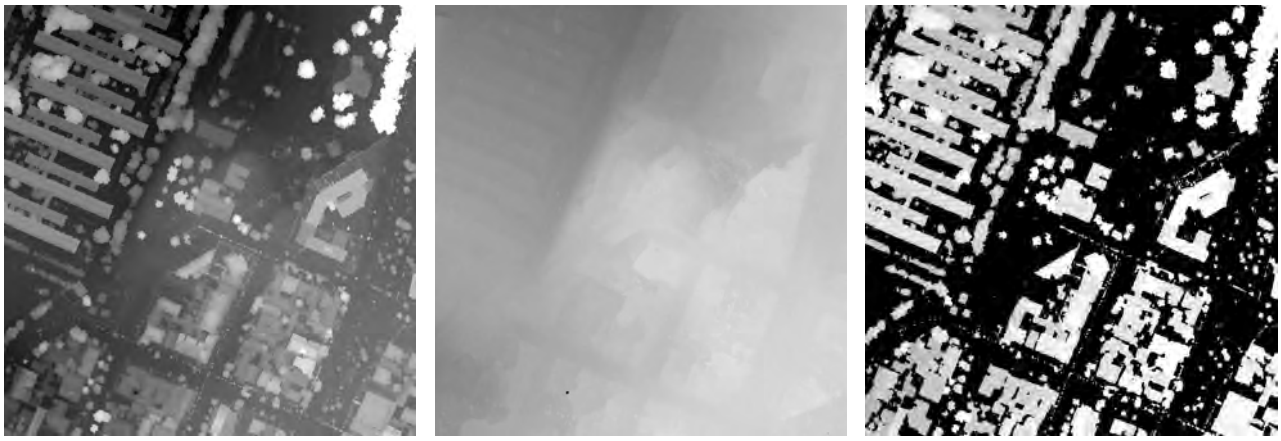
2.5.2 DSM Segmentation

The first step for classification of a DSM is its segmentation. There is a difference between LIDAR filtering and DSM segmentation. The first aims simply the bare ground point detection, while the second has more topological significance. A simple LIDAR filtering extracts all points above terrain, no matter if they have meaning or not. For example, all cars and small trees are classified as high points, but these objects have no practical importance in most applications.

This work proposes a DSM segmentation methodology based on the works of Tovari (2006) and Steinle (2005). These works use the normalized Digital Surface Models (nDSM) to process the DSM segmentation. Normalized digital surface models are derived from a DTM and DSM, i.e. it can be generated by subtracting the DTM from the DSM. The influence of topography on heights is excluded from the surface model. It contains all objects on the terrain surface, in an ideal case only the objects without any terrain influence. Since the heights are derived from a DTM, the inaccuracy – which is originated from the filtering and interpolation errors – appears in the nDSM as well. Although, contrary to (Tovari, 2006) and (Steinle, 2005), this work does not use the nDSM because it is necessary to have the actual altitude of the points in the further processing.

This work proposes the use of a LIDAR filtering (section 2.4.1) to produce a DTM (section 2.4.2). The DTM is exported as a raster image with the same resolution of the DSM generated (section 2.5.1). Significant off-terrain objects are such whose height difference ΔH (difference between real altitude H_P and the terrain altitude H_T) is higher than an empirically determined threshold τ:

$$\Delta H = H_P - H_T \begin{cases} \textbf{if } \Delta H > \tau \textbf{ then} \text{ is object} \\ \textbf{if } \Delta H < \tau \textbf{ then} \text{ is not object} \end{cases} \tag{2.5}$$



(a) DSM (b) DTM (c) Segmentation

Figure 2.6: DSM Segmentation Example

In urban environments, buildings and trees are important objects to be classified. This work uses a τ value of 2.5m, since almost all small objects are lower than this limit. Figure 2.6(a) shows a cut of a DSM generated with the modified Araki algorithm. Figure 2.6(b) shows the DTM generated with the methodology described in section (2.4). Figure 2.6(c) shows the resulting segmented DSM. Note that high trees and buildings are well separated from the bare ground.

2.6 DSM Height Texture Segmentation

Height texture is the variation of height values with respect to the neighboring pixels (Steinle, 2005). Since man made objects tend to be smoother than vegetation, the height texture is a possible approach for the distinction between vegetation and buildings. While artificial objects such as buildings consist of continuous, compact surfaces that are bounded by discontinuous edges, natural objects such as

vegetation have larger vertical variations throughout the objects since the beam can penetrate the canopy of trees (Elmqvist *et al.*, 2001).

As one can see in figure 2.6(c), several vegetation areas and buildings are melted together. A possible criterion to split these areas is the roughness of the surface measured by differential geometric quantities, like gradients or curvatures (Brunn & Weidner, 1997). Suitable results can be obtained by the Laplace operator (Maas, 1999) or by local curvature (Steinle & Vögtle, 2001), i.e. the difference of subsequent gradients in the four directions across a raster point (Tovari, 2006). Elmqvist *et al.* (2001) use the second derivative and the maximum slope of each pixel and its eight neighboring pixels. In vegetation areas, where the height between neighboring pixels considerably varies, the second derivative and slope are larger than within buildings where the change in height of a flat or tiled roof is small.

In this work, the objective of the DSM segmentation is to prepare the LIDAR data for the fusion with aerial imagery. We do not attempt, in this section, to propose a final object classification. This step of the work focuses on the splitting of smooth areas (which tend to be buildings), from rough areas (which tend to be vegetation). The key word here is "tend": it is not possible to affirm that a segment is building or vegetation just based on its height texture, since some complex buildings show really rough, and some dense trees show really smooth. Förstner (1994) presents an approach to feature extraction from digital images. A homogeneity measure h is used to distinguish between homogeneous and non-homogeneous regions. The image is classified into regions satisfying this homogeneity criterion. The homogeneity criterion used is related to the discontinuities on the image, which are a function of changes of the surface normals. This chapter proposes a statistical least-squares approach to the determination of the homogeneity measure. This is achieved through the use of the variance of unit weight a posteriori of the least-squares adjustment. The functional model — $f(z)$ — chosen is the equation of the plane on space:

$$f(z) = a \cdot x + b \cdot y + c \tag{2.6}$$

In equation (2.6), (a, b, c) are the parameters to be determined, (x, y) are the planimetric coordinates of a point on the image, and z its height. For a set of n points in space, one can build the following system of equations:

$$\begin{cases} z_1 = a \cdot x_1 + b \cdot y_1 + c \\ z_2 = a \cdot x_2 + b \cdot y_2 + c \\ \vdots \\ z_n = a \cdot x_n + b \cdot y_n + c \end{cases} \tag{2.7}$$

Equation (2.7) builds a linear system, which can be solved by least-squares. The first step is to define the matrix A, which is calculated from the derivatives of $f(z)$:

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial a} & \frac{\partial f_1}{\partial b} & \frac{\partial f_1}{\partial c} \\ \frac{\partial f_2}{\partial a} & \frac{\partial f_2}{\partial b} & \frac{\partial f_2}{\partial c} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial a} & \frac{\partial f_n}{\partial b} & \frac{\partial f_n}{\partial c} \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \tag{2.8}$$

The z coordinates are treated as observations, and they build the observations vector L_b. Since the standard deviations of the LIDAR points are known, one can also build the weight matrix P.

$$L_b = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \quad P = \begin{bmatrix} 1/\sigma_1^2 & 0 & \cdots & 0 \\ 0 & 1/\sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/\sigma_n^2 \end{bmatrix} \tag{2.9}$$

The parameters vector X is given by:

$$X = \left(A^T \cdot P \cdot A\right)^{-1} \cdot \left(A^T \cdot P \cdot L_b\right) = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{2.10}$$

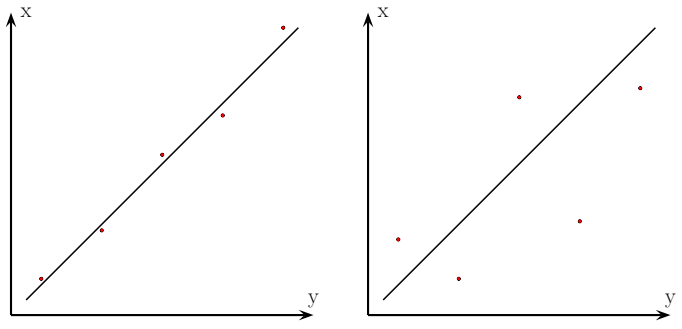
Next, the vector of residuals V of the observations L_b is calculated:

$$V = A \cdot X - L_b \tag{2.11}$$

The most important statistic of an adjustment, based on the residuals vector V is the *variance of unit weight a posteriori* ($\hat{\sigma}_0^2$). This value shows how the parameters fit to the model, based on the residuals of the observations. Rigourously speaking, the $\hat{\sigma}_0^2$ indicates the relation between the a priori precisions given to the observations (matrix P) and its residuals after the adjustment (vector V). In other words, if $\hat{\sigma}_0^2$ is about 1, the obtained residuals are compatible with the precisions given to the observations. Given the number of observations n and the number of parameters u, then $\hat{\sigma}_0^2$ is given by:

$$\hat{\sigma}_0^2 = \frac{V^T \cdot P \cdot V}{(n-u)} \tag{2.12}$$

Figure (2.7) shows a simple example of how $\hat{\sigma}_0^2$ behaves. Given a set of 5 points, one fits a line to theses points. If the points accurately model the line, like in figure 2.7(a), then $\hat{\sigma}_0^2 \approx 1$. If the points do not model a line, like figure 2.7(b), then $\hat{\sigma}_0^2 \gg 1$.

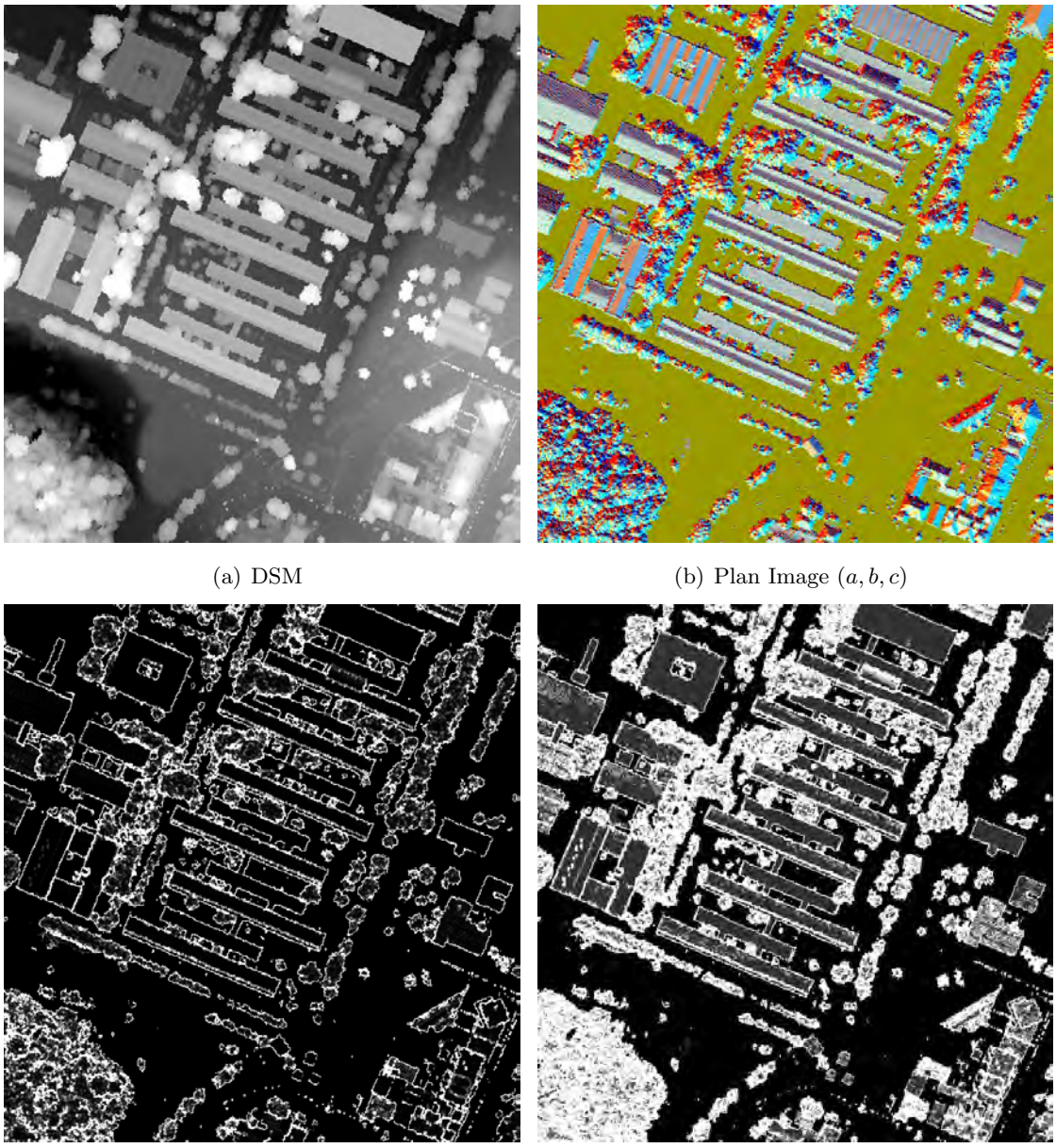


(a) $\hat{\sigma}_0^2 \approx 1$ (b) $\hat{\sigma}_0^2 \gg 1$

Figure 2.7: Expected $\hat{\sigma}_0^2$ Behavior

This work proposes to calculate $\hat{\sigma}_0^2$ for every pixel on a DSM, using the plane equation (eq. 2.6) as functional model. It delivers an overview of the rough and smooth areas: areas with lower $\hat{\sigma}_0^2$ values are smoother than areas with higher $\hat{\sigma}_0^2$. On the object borders, where the equation of the plane does

not fit at all, $\hat{\sigma}_0^2$ will show extremely high. On the other hand, on flat roofs, $\hat{\sigma}_0^2$ will show near to 1. To achieve these results, a window size is chosen and $(a, b, c, \hat{\sigma}_0^2)$ are determined for every pixel in the image. After that, the ground pixels are erased based on the segmentation result. At last, a preliminary classification based on $\hat{\sigma}_0^2$ is performed. Two thresholds are set: a low $\hat{\sigma}_0^2$ threshold (τ_l) and a high $\hat{\sigma}_0^2$ threshold (τ_h). If $\hat{\sigma}_0^2 < \tau_l$, the pixel is classified as smooth. If $\hat{\sigma}_0^2 > \tau_h$ the pixel is classified as rough. If $\tau_l \leq \hat{\sigma}_0^2 \leq \tau_h$, the pixel is classified as undefined. Figure (2.8) shows an example.



(a) DSM

(b) Plan Image (a, b, c)

(c) $\hat{\sigma}_0^2$ Image

(d) Preliminary Classification

Figure 2.8: $\hat{\sigma}_0^2$ Classification Methodology Example

Figure 2.8(a) shows the DSM. Figure 2.8(b) shows the *plan image*: an RGB composition of the obtained plane parameters (a, b, c). Figure shows 2.8(c) the $\hat{\sigma}_0^2$. White means a $\hat{\sigma}_0^2 > 1000$. Figure 2.8(d) shows the preliminary classification: black means ground, dark gray smooth, light gray means unclassified and white means rough. Note in figure 2.8(b) how the vegetation areas (down left corner) show much more texturized as the building roofs, and how it reflects directly in the preliminary classification.

Classification Refinement

DSM tend to be noisy. The noise interferes directly with the $\hat{\sigma}_0^2$ determination, making smooth areas show rougher as expected, especially on smaller roofs, as one can see on the down left corner of the fig. 2.8(d), where a residential area is shown. Because of these noise effects, the proposed classification must be refined in order to reduce the image fragmentation as seen in the fig. 2.8(d). This work proposes an iterative five step region growing solution for the classification refinement:

1. Same Plane Refinement: On this step, just the non-border unclassified pixels are analyzed. Given a window size, a smoothness threshold τ_s, a roughness threshold τ_s, and the standard deviation σ_p of the mean of the parameters a and b for all pixel inside the window; if $\sigma_p < \tau_s$, the pixel is reclassified as smooth. If $\sigma_p > \tau_r$, the pixel is reclassified as rough. If $\tau_s \geq \sigma_p \geq \tau_r$; the pixel stays unclassified.

2. First Inplane Refinement: Here just non-border unclassified pixels are analyzed. For a given unclassified pixel, if more than $N_{\min}$ pixels in the 8-neighborhood are smooth, the distances d_p between the point and all planes defined on the smooth pixels are determined. If more than $n_{\min}$ distances show smaller than a threshold τ_d, the pixel is reclassified as smooth. If more than $n_{\min}$ distances show bigger than τ_d, the pixel is reclassified as rough.

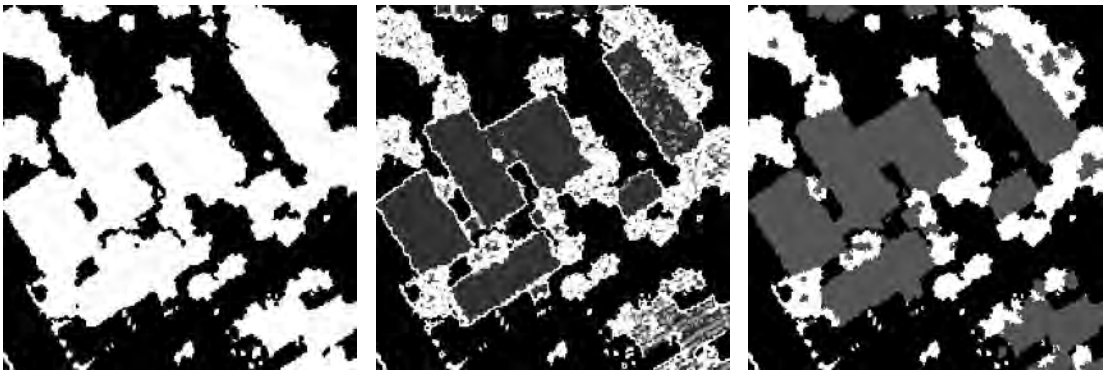
3. Second Inplane Refinement: In this step, the non-border rough pixels that interface smooth pixels are analyzed. The reclassification proceeds exactly as in the step 2.

4. Border Refinement: Here, the rough border pixels are analyzed and reclassified exactly as in the step 3.

5. Interface Refinement: After many reclassification iterations, it is possible that pixels on the interface between smooth and rough areas are misclassified. This misclassification is treated on this step. First, for all interface pixels classified as smooth, the height difference between the pixel and its smooth neighbors are calculated. If more than n height differences show higher than a threshold $\tau_{\Delta h}$, the pixel is reclassified as rough. Next, all rough pixels on the interfaces are analyzed in a similar way. For all interface pixels classified as rough, the height difference between the pixel and its smooth neighbors are calculated. If more than n height differences show lower than a threshold $\tau_{\Delta h}$, the pixel is reclassified as smooth.

This classification refinement does not use the flood-fill concept as in the aerial imagery segmentation. Here, pixels in the interfaces are iteratively analyzed. Furthermore, the five steps are not sequentially executed. First, the steps 1, 2 and 3 are iteratively run until a satisfactory number of non-border pixels are reclassified. After that, the borders are iteratively analyzed (step 4), and at last the step 5 is run.

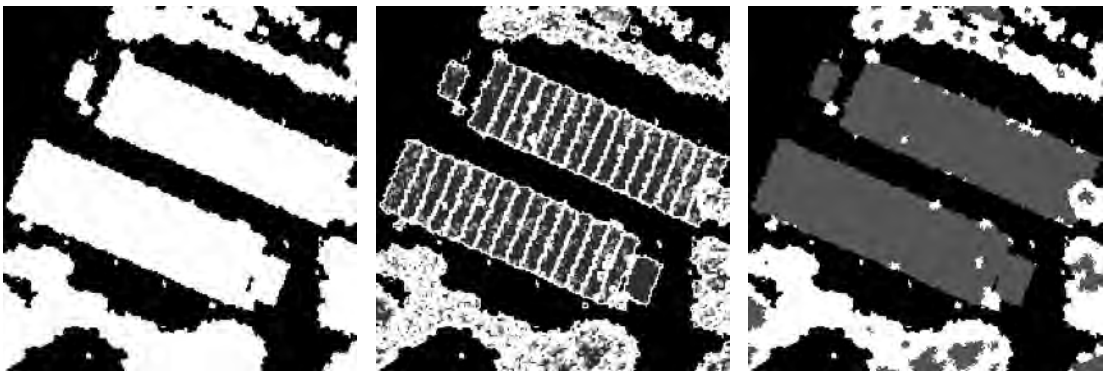
Figures (2.9), (2.10) and (2.11) show some examples of the full DSM segmentation. Figure 2.9(a) shows a complex segment, where vegetation and buildings are mixed. Figure 2.9(b) shows the preliminary $\hat{\sigma}_0^2$ classification. Finally, after the iterative refinement, figure 2.9(c) shows how the algorithm creates a segmentation based on the height texture: black areas mean ground, gray areas the smooth segments and the white areas the rough segments.



(a) Initial Segmentation (b) First Classification (c) Final Split Image

Figure 2.9: Split Process Example 1

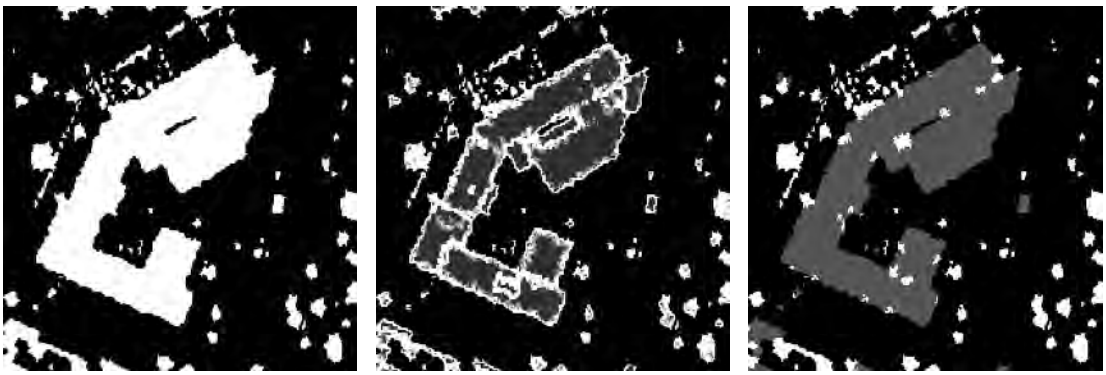
Figure (2.10) shows two isolated buildings and some dense vegetation areas. The roofs of the buildings are not flat, they show multiple faces, whose borders are quite visible in figure 2.10(b). After the refinement, the roofs show almost fully classified as smooth, as one can see in figure 2.10(c). Some dense tree tops are classified as smooth as well:



(a) Initial Segmentation (b) First Classification (c) Final Split Image

Figure 2.10: Split Process Example 2

Figure (2.11) shows the same building as figure (1.13). Note how almost all roof parts are joined in the final split in figure 2.11(c).



(a) Initial Segmentation (b) First Classification (c) Final Split Image

Figure 2.11: Split Process Example 3

2.7 LIDAR Processing Overview

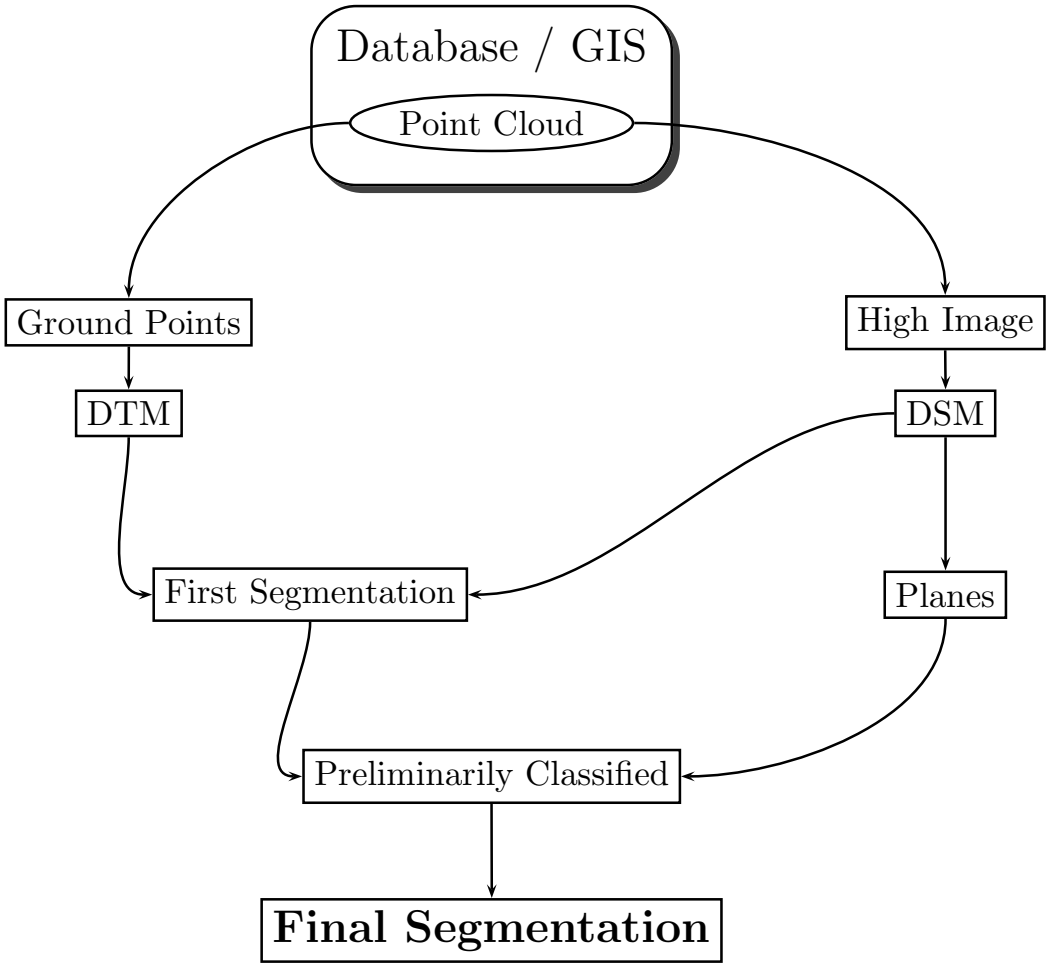
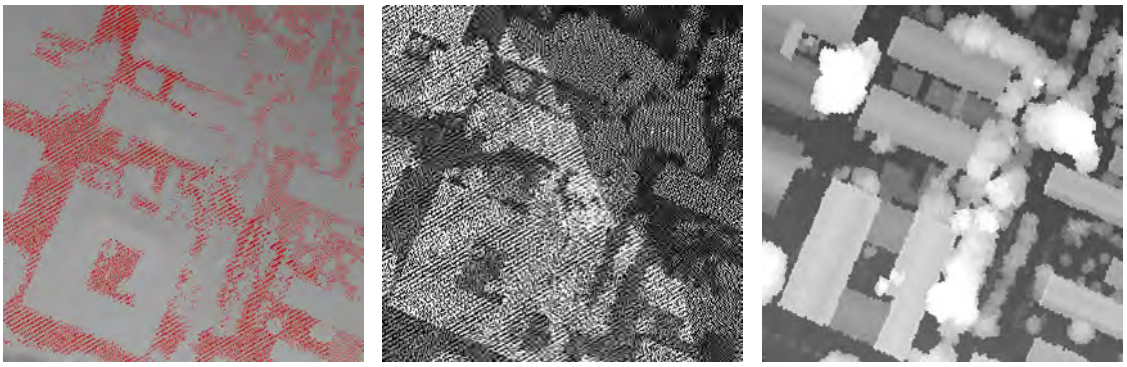


Figure 2.12: Image Processing Workflow

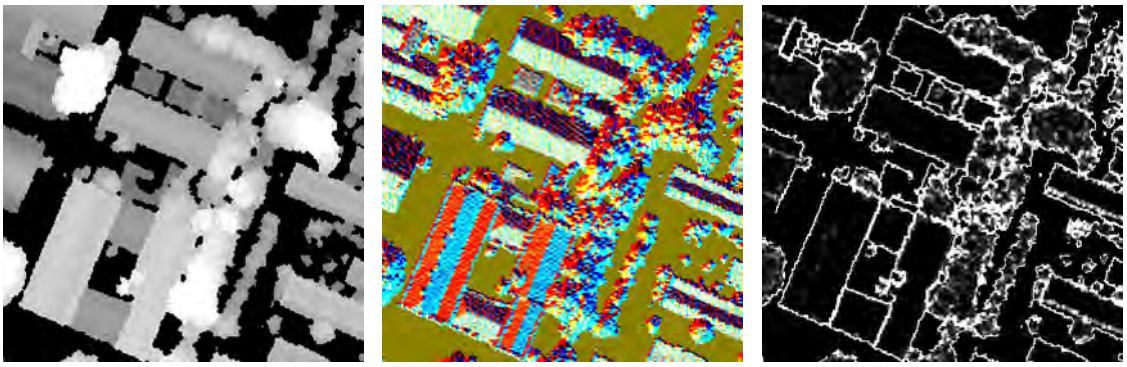
This section presents the proposed LIDAR processing at a glance. Figure (2.12) shows the workflow. We can split this workflow in 3 steps. In the *LIDAR preprocessing step* the point cloud is stored and indexed in a geodatabase, the LIDAR data is filtered and the DTM is generated. Also from the point cloud, the DSM is generated with the modified Araki algorithm. In the *plane processing step*, a segmentation image is generated, the values of a, b, c and $\hat{\sigma}_0^2$ are calculated from the DSM (DSM) and stored on the plan image. On the *segment splitting step*, using the segmentation and the $\hat{\sigma}_0^2$ values, the first classified image is generated. At last, this image is refined and the final segmented image is created.

Figures (2.13), (2.14) and (2.15) show a full example of the proposed workflow. The LIDAR system is the Optech ALTM 2050, flown over de UFPR Campus in Curitiba, Brazil. The images have a 0.5m GSD. Figure 2.13 shows the LIDAR pre-processing: the ground filtering and DTM generation in 2.13(a), high image in 2.13(b) and the DSM in 2.13(c). Figure (2.14) shows the plane processing: the preliminary segmented image in 2.14(a), the plane parameters (RGB composition of a, b, c) obtained in 2.14(b) and the $\hat{\sigma}_0^2$ in 2.14(c). Figure (2.15) shows the final segmentation: figure 2.15(a) shows the first classified image and figure 2.15(b) shows the final segmented image:



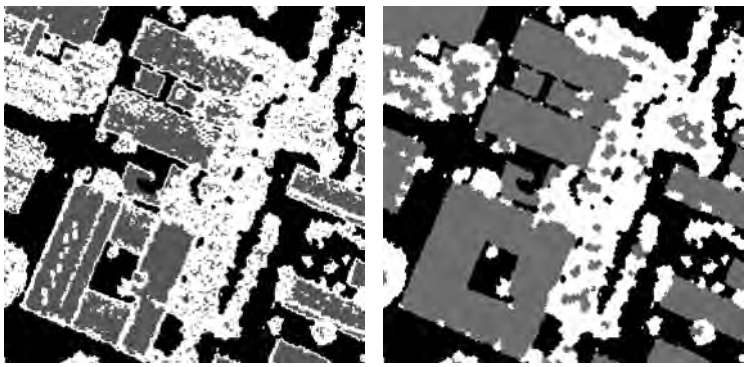
(a) Ground Points and DTM (b) High Image (c) DSM

Figure 2.13: Step 1 – LIDAR Preprocessing Step – UFPR



(a) Prel. Segmentation (b) Plane (a, b, c) (c) $\hat{\sigma}_0^2$

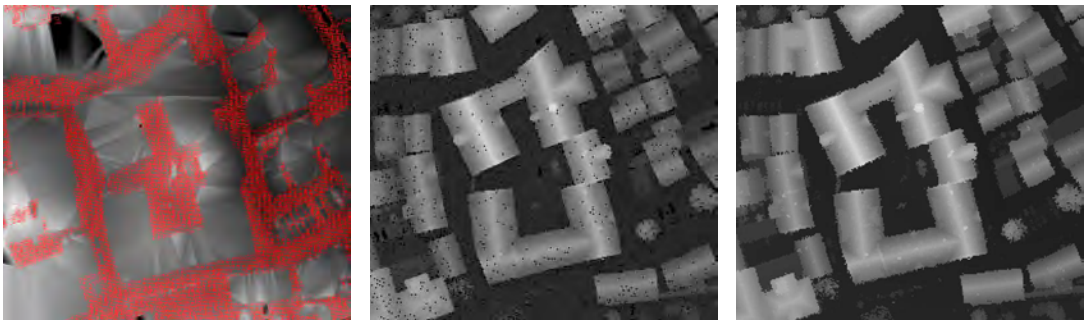
Figure 2.14: Step 2 – Plane Processing – UFPR



(a) First Classification (b) Final Segmentation

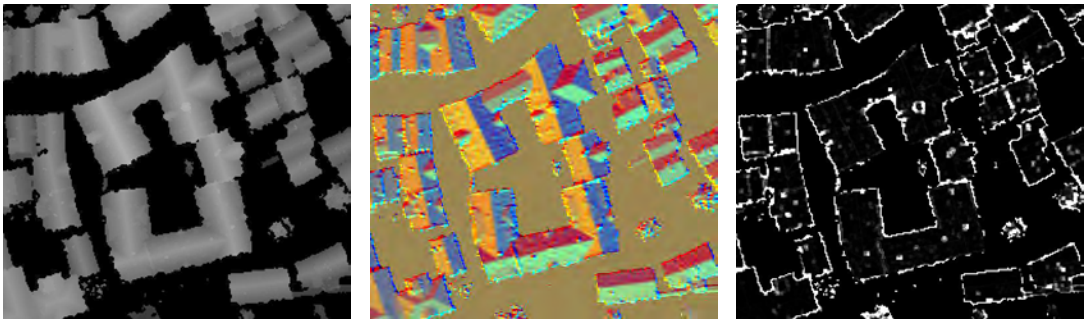
Figure 2.15: Step 3 – Segment Splitting Step – UFPR

Figures (2.16), (2.17) and (2.18) show another example of the proposed workflow. The laser scanner is a Toposys Harrior 56, flown over the Biberach downtown, in Germany. The images have 0.5m GSD. Figure (2.16) shows the LIDAR pre-processing: the ground filtering and DTM generation in 2.16(a), high image in 2.16(b) and the DSM in 2.16(c). Figure (2.17) shows the plane processing: the segmented image in 2.17(a), the plane parameters obtained in 2.17(b) and the $\hat{\sigma}_0^2$ in 2.17(c). Figure (2.18) shows the final segmentation: fig. 2.18(a) shows the first classified image and fig. 2.18(b) shows the segmented image:



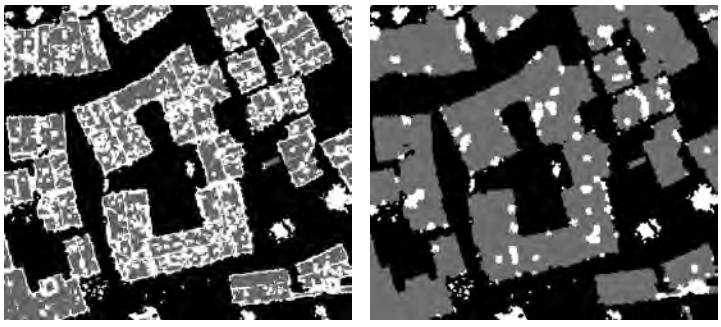
(a) Ground Points and DTM (b) High Image (c) DSM

Figure 2.16: Step 1 – LIDAR Preprocessing – Biberach



(a) Prel. Segmentation (b) Plane (a, b, c) (c) $\hat{\sigma}_0^2$

Figure 2.17: Step 2 – Plane Processing – Biberach



(a) First Classification (b) Final Segmentation

Figure 2.18: Step 3 – Segment Splitting – Biberach

The most interesting analysis comes from the split images in the fig. 2.15(b) and 2.18(b). Note that the proposed methodology shows efficient on the splitting of large DSM segments with significant mixture of buildings and dense vegetation, as one can see on fig. 2.15(b). One can also note that areas on the tops of dense trees are classified as smooth. On the other hand, one can see in 2.18(b) that there are many small rough segments inside the building, which occurs because of noise or because of the building complexity. Discrepancies in the results obtained from different sensors happen due to sensor configuration, plane calculation parametrization, noise, point density, object complexity, just to cite a few key reasons. Although the proposed methodology splits the segments quite well, it still does not classify the segments as vegetation or building. To overcome this issue, the chapter 3 shows a methodology that uses spectral information to classify the DSM segments.

Chapter 3

Digital Aerial Imagery and LIDAR Data Fusion

3.1 Introduction

The cartographic objective of urban mapping is the optimal presentation of urban structures with respect to the mapping purpose, like education, planning or navigation. The advances in GIS, Surveying, Remote Sensing and Photogrammetry in the last years made it possible for national mapping, space, postal or environmental agencies and other data providers to build up geographic databases with higher precision and resolution (Steiniger *et al.*, 2008). One example for a GIS data source is the LIDAR altimetry, which emerged in the last years as a leading technology for physical surfaces extraction. For urban mapping, this technology can be used to produce detailed surfaces, which can be used for 3D modeling, urban planning, telecommunication and real state management, among many others (Filin, 2004). Also digital photogrammetric methods have become widely used for providing DSMs and DTMs due to its efficiency and cost effectiveness. However, this efficiency decreases rapidly for complex urban scenes due to the failures of image matching, which are primarily caused by occlusions, depth discontinuities, shadows, textures, poor image quality, between others. Both LIDAR and classic photogrammetric procedures still need human-guided operations, which are costly and time-consuming. (Zhou *et al.*, 2004).

The future's information society will require up-to-date object-oriented three-dimensional geo-information. For an increasing number of applications, two-dimensional vector maps with update frequencies of a few years are already insufficient. Only automatic methodologies can satisfy these new demands and keep the production costs within reasonable bounds (Vosselman *et al.*, 2005). Thus, the automation of the generation of 3D city models has become lately a major focus of geoinformation research. Due to its advantages as an active technique for reliable 3D point determination, LIDAR has become a rather important source of information for the generation of this kind of model (Maas & Vosselman, 1999). One of the most serious problems in 3D mapping using solely LIDAR data as source is that the location accuracy of height jump edges tends to be quite low. Several algorithms use 2D ground plans containing the outlines of buildings, as pointed by Haala *et al.* (1998), Vosselman & Dijkman (2001) and Alexander *et al.* (2009). In the absence of building footprint data, building boundaries have been approximated from LIDAR data (Alharthy & Bethel, 2002; Cho *et al.*, 2004), or digitized from aerial photographs (Palmer & Shan, 2002).

This chapter presents an automatic building footprint detection technique supported by low-cost aerial imagery and LIDAR data. This sort of methodology can play an important role in 3D mapping, since in many urban environments, especially those in developing countries, no actual 2D cartography is available.

One of the most important steps on the modeling of urban environments is feature extraction from source data and its description. In general, features are distinguished as being either local or global. Local features are the classic cartographic elements: points, lines and areas. Larger features, also called structures are also known as global features, and these are composed of different local features. Relations between local features are introduced to characterize global features. These relations can be geometric (like distances and angles), radiometric (like gray value differences) or topological (like the notion that one feature is contained within another) (Wendt, 2007).

This work proposes a combined use of LIDAR data and aerial images to perform local feature extraction. The challenge here is to fuse two very distinct datasets. Accordingly to Baltsavias (1999), the major differences between Photogrammetry and LIDAR are: passive vs. active, high-power, collimated and monochromatic sensing; generally frame or linear sensors with perspective geometry vs. generally point sensors with polar geometry; full area coverage vs. pointwise sampling; indirect vs. direct acquisition or encoding of 3D coordinates; geometrically and radiometrically high quality images with multispectral capabilities vs. no imaging or monochromatic images of inferior quality; and ability for LIDAR to 'see' objects much smaller than the footprint. The features that can be extracted from each type of data source are distinct, but complementary. The complementary nature of the two methods is more evident when we attempt to describe the surface explicitly. Table 3.1 shows a comparison between surface properties obtained with LIDAR and aerial imagery trough image correlation techniques (Schenk & Csathó, 2002).

Surface Property	Point Cloud	Aerial Imagery
Patches	×	
Boundaries		×
Discontinuities		×
Roughness	×	

Table 3.1: Sources that Predominantly Determine Surface Properties (Schenk & Csathó, 2002)

Building detection essentially requires a classification of the input data, separating buildings from other objects (mainly trees). In order to accomplish this classification, parameters such as the height of LIDAR points above the terrain or the roughness of the surface can be used. Other information can be considered, like hight differences between first and last echoes and normalized difference vegetation index (NDVI) (Lu *et al.*, 2006; Rottensteiner *et al.*, 2007).

Various classification techniques have been applied for building detection, for example, unsupervised classification (Haala & Brenner, 1999), rule-based classification (Rottensteiner & Briese, 2002), Bayesian networks (Brunn & Weidner, 1997; Stassopoulou *et al.*, 2000), fuzzy logic (Matikainen & Hyyppa, 2003; Tovari, 2006; Vögtle & Steinle, 2003). Also hierarchical and multiscale approaches (Baltsavias, 2004; Hofmann *et al.*, 2002; Matikainen & Hyyppa, 2003; Vosselman *et al.*, 2005; Vu *et al.*, 2009) seem to offer promising solutions to accommodate more complex urban environments.

The fusion of aerial imagery and LIDAR offers interesting applications, like LIDAR data control, change detection (Schenk & Csathó, 2002) and creation of three-dimensional topological datasets (Vosselman *et al.*, 2005). This chapter proposes a fusion methodology in two main steps: first, the classification of the segmented DSM obtained in the chapter 2 using spectral information from the aerial imagery, and then a vector based integration of the segmented aerial images to perform edification boundaries detection. Conceptually, the proposed approach is close to the definition of fusion provided by Wald (1999): *"Fusion aims at obtaining information of greater quality"*.

3.2 DSM Classification with Fuzzy Logic

This section presents the proposed fuzzy based classification procedure. Section 3.2.1 shows an applied overview of fuzzy set theory. Section (3.2.2) proposes the use of orthorectification to bring spectral information to the segmented DSM. Section (3.2.3) presents new segment properties used to perform the fuzzy classification. At last, section (3.2.4) applies all knowledge from the previous sections to perform the classification itself.

3.2.1 Fuzzy Logic and Fuzzy Sets: An Applied Overview

Among various paradigmatic changes in computer science in the last years, one such change concerns the concept of *uncertainty*. In computer science, this change has been manifested by a gradual transition from the traditional view, which insists that uncertainty is undesirable and should be avoided by all possible means, to an alternative view, which is tolerant of uncertainty and insists that it is not possible to avoid it (Klir & Yuan, 1995). In geodesy, however, uncertainty has always been treated as major dimension of spatial data quality, arising from the granularity or resolution at which observations of phenomena are made, and from limitations imposed by computational representation and processing (Worboys, 1998), since uncertainty is part of the information. Therefore, the classification of geospatial datasets must take into account their uncertainty aspects.

The nature of uncertainty depends on the mathematical theory within which uncertainty pertaining to various problem-solving situations is formalized. Each formalization is a mathematical model of the situation. Assume that we can measure the amount of uncertainty involved in a problem-solving situation formalized in a particular mathematical theory. Assume further that the amount of uncertainty can be reduced by obtaining relevant information as a result of some action (adding observations, finding relevant new facts, performing relevant experiments and observing their outcomes, etc.). Then, the amount of information obtained by the action may be measured by the reduction of uncertainty that results from the action. However, information measured solely by uncertainty reduction does not capture the rich notion of information that human communication offers (Dubois & Prade, 2000). In order to overcome this issue, Zadeh (1965) proposed a method suitable for the mathematical modeling of vague human linguistic concepts such as "small", "approximately" or "similar", by means of *fuzziness*. In this paper, he introduced the theory whose objects (*fuzzy sets*) are sets with boundaries that are not precise. The membership in a fuzzy set is not a matter of affirmation or denial, but a matter of *degree*.

The significance of Zadeh's paper was that it challenged not only probability theory as the sole agent for uncertainty, but the very foundations upon which probability theory is based: the aristotelian

two-valued (boolean) logic. When A is a fuzzy set and x is a relevant object, the proposition "x is member of A" is not necessarily either true or false, but it may be true only to some degree, the degree to which x is actually a member of A. It is common to express degrees of membership in fuzzy sets as well as degrees of truth of the associated propositions by numbers in the closed unit interval $[0, 1]$. The values, 0 and 1, then represent, respectively, the total denial and affirmation of the membership in a given fuzzy set as well the falsity and truth of the associated proposition (Klir & Yuan, 1995). The function that expresses the degree of membership of an object x in a fuzzy set A is called *membership function*:

$$\mu(x) : X \rightarrow [0, 1] \tag{3.1}$$

Usually, we work in a quantitative setting, where the information is expressed by means of numerical values. However, many aspects of different activities in the real world cannot be assessed in a quantitative form, but rather in a qualitative one, with vague or imprecise knowledge. In that case a better approach may be to use linguistic assessments instead of numerical values (Marichal, 2002). The semantic of the linguistic terms is given by fuzzy numbers defined in the $[0, 1]$ interval. A way to characterize a fuzzy number is to use a representation based on parameters of its membership function (Bonissone & Decker, 1986). The simplest membership functions are formed using straight lines. Of these, the simplest is the triangular membership function which is nothing more than a collection of three points forming a triangle. A trapezoidal function is achieved by a 4-tuple (a, b, c, d) where the b and c indicates the interval in which the membership value is 1. The triangular membership function is a particular case of the trapezoidal case, where $b = c$ (Herrera *et al.*, 2002). Normally, membership functions are defined in an empirical way by means of training samples visually selected and interpreted by an operator (Tovari & Vögtle, 2004).

In order to make these concepts clear, we propose an example taken from the section 2.6. There, the segmented DSM was split into "rough" and "smooth" segments. This can be understood as a classic boolean classification. Obviously, the concept of roughness is not that simple. It makes more sense to think in roughness with linguistic terms as, for example, "rough" (R), "medium rough" (MR), "medium smooth" (MS) and "smooth" (S). We can use, for example, the percentage of pixels with high $\hat{\sigma}_0^2$ (eq. 2.6) as a numeric input for this classification, and create a set of tuples: S = $(0\%, 0\%, 20\%, 25\%)$; MS = $(15\%, 30\%, 45\%, 60\%)$; MR = $(40\%, 55\%, 70\%, 85\%)$ R = $(65\%, 80\%, 100\%, 100\%)$. Figure 3.1 shows a graphic representation of these membership functions.

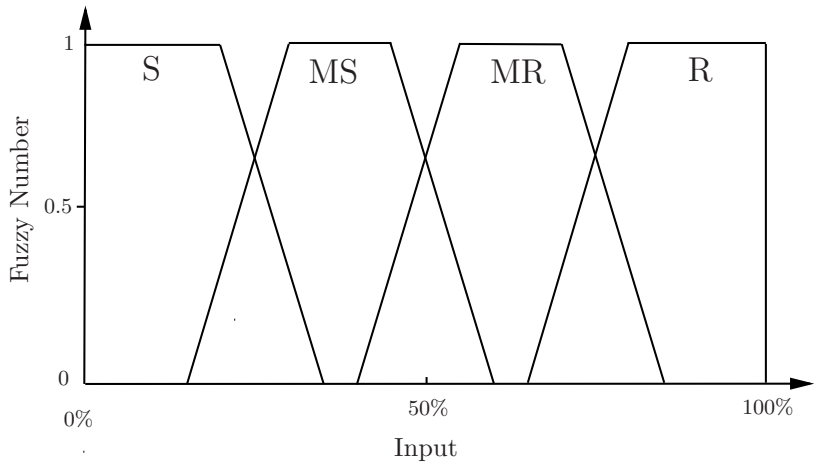


Figure 3.1: Membership Functions

Let us analyze a segment which has 23% of its pixels with high $\hat{\sigma}_0^2$ values. The interpretation of this fact using the membership function (fig. 3.1) is called *fuzzification*. This procedure transforms a real-world information (the percentage of pixels with high $\hat{\sigma}_0^2$) into a *fuzzy number* in the interval [0,1]. Figure 3.2 shows how fuzzification works: for the value of 23%, the segment has a degree of pertaining of 0.8 to the fuzzy set "smooth" and 0.5 for the fuzzy set "medium smooth".

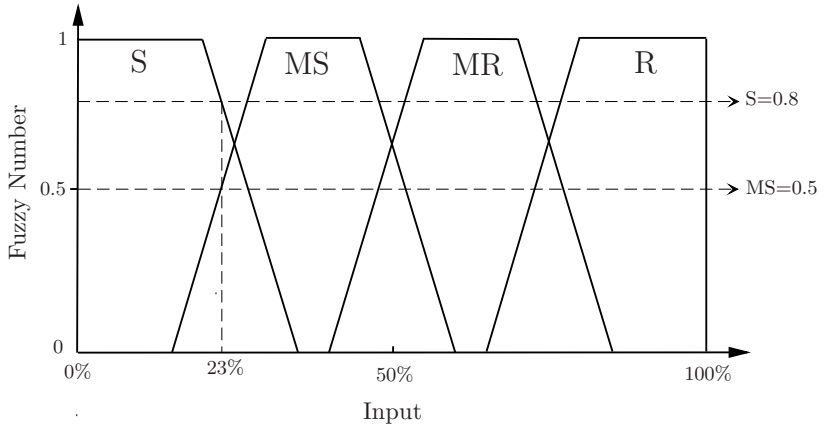


Figure 3.2: Fuzzification Example

After fuzzification it is necessary to interpret the meaning of the fuzzy number. This step is called *implication*. Implication shows the relation between two fuzzy sets: the antecedent (which was fuzzified), and the consequent (which will be deffuzified). For example: a smooth segment tends to pertain to a flat roof. This affirmation can be translated into a logic sentence, called *fuzzy rule*:

"**IF** a segment **IS** *smooth* **THEN** this segment **IS** a *flat roof*"

It is necessary to define the fuzzy set "flat roof" (FR) in terms of its roughness. We can explore the fact that the probability in % of a segment to be a flat roof declines with its roughness.

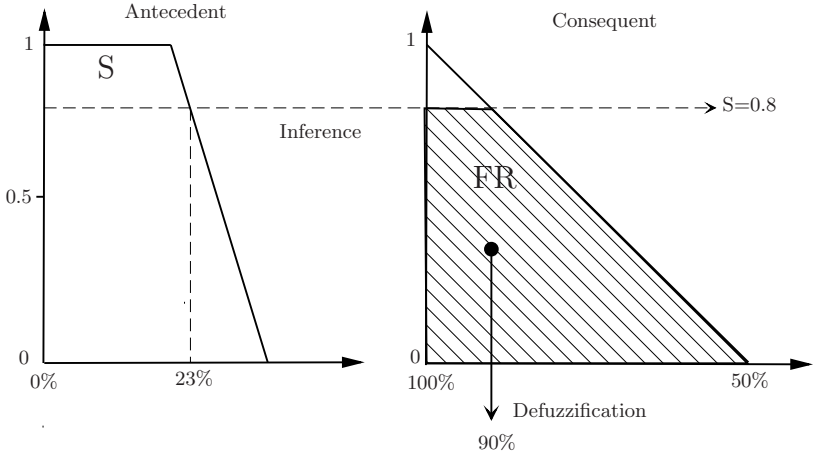
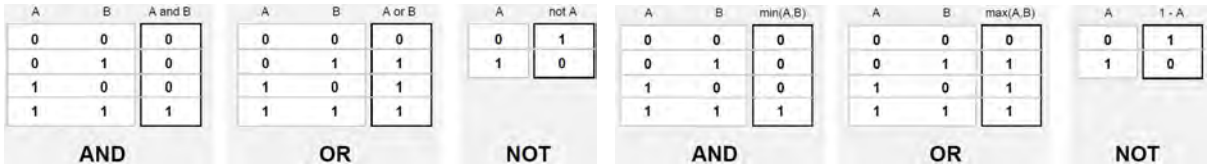


Figure 3.3: Implication Example

Figure (3.3) shows also how to transform the fuzzy numbers in "real-world" parameters. First, one cuts the polygon formed from the intersection of the consequent fuzzy set and the inference value. Then, the centroid of this area is calculated (or other method can be used). The abscissa of the centroid is

the defuzzified value. In this example, the result is that the segment has 90% probability to be a flat roof.

In most applications it is necessary to use multiple variables. This can be achieved by using *fuzzy reasoning operations*. An important thing to realize about fuzzy logical reasoning is the fact that it is a generalization of standard Boolean logic. In other words, if you keep the fuzzy values at their extremes of 1 (completely true), and 0 (completely false), standard logical operations will hold. Figure (3.4) shows the truth tables for boolean and fuzzy logic:



(a) Boolean (b) Fuzzy

Figure 3.4: Truth Tables (MathWorks, 2009)

In figure (3.5), the truth tables (figures 3.4(a) and 3.4(b)) are converted to a plot of two fuzzy sets applied together to create one fuzzy set. The upper part of the figure displays plots corresponding to the boolean truth tables, while the lower part of the figure displays how the operations work over a continuously varying range of truth values A and B according to the defined fuzzy operations.

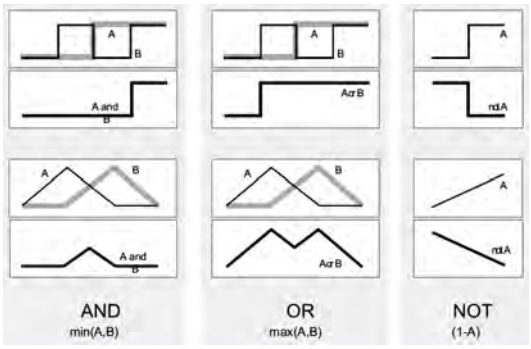


Figure 3.5: Comparison between Boolean and Fuzzy Logic (MathWorks, 2009)

Let us use a practical example to illustrate the fuzzy reasoning concept. Nowadays, many sensors deliver near infrared (NIR) information. This is very useful for vegetation detection. This detection is normally done through the normalized difference vegetation index ($NDVI$), which is determined using the red (R) and near infrared bands:

$$NDVI = \frac{NIR - R}{NIR + R} \tag{3.2}$$

$NDVI$ has a range [0,1], where 1 means very healthy vegetation, an 0 no vegetation at all. If we determine a mean NDVI of the split image segment, it is possible to set up a rule for a fuzzy set for low NDVI values called VL (*Vegetation Low*):

"**IF** the segment **IS** *smooth* **AND** its *VL* **IS** *low* **THEN** the segment is a *flat roof*"

Figure (3.6) shows graphically a fuzzy set (VL) for a low vegetation index, and how the fuzzy system proceeds for an NDVI of 0.12:

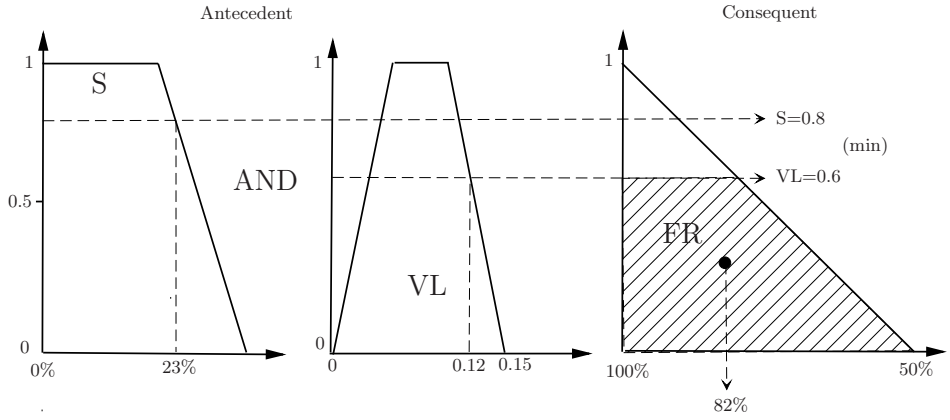


Figure 3.6: Fuzzy Reasoning Example

Normally it is necessary to aggregate more statements in order to obtain a final result. This is achieved by adding linguistic rules to the fuzzy system. A very useful parameter for a given segment is its size. Let us define the linguistic terms "large", "medium" or "small" for it. Let us also accept that, for a specific study area, we know that roofs have medium sizes. It is possible, now, to set up a membership function for medium areas (MA). The *rule block* (set of fuzzy rules) can be written as follows:

1 : "**IF** the segment **IS** *smooth* **AND** its *NDVI (VL)* **IS** *low* **THEN** the segment is a *flat roof*"

2 : "**IF** the *area* **IS** *medium* **THEN** the segment **IS** a *flat roof*"

The next step is to aggregate all rules into a single result (for example, the probability that a segment is a flat roof). This step is called *aggregation* process. Herrera *et al.* (2002) defines aggregation as the process in which the individual linguistic preference values are combined to obtain collective preference values. As long as the aggregation method is commutative (which it always should be), the order in which the rules are executed is unimportant. This work uses the operator "product" to aggregate the outputs, as pointed by Tovari (2006); Tovari & Vögtle (2004). Let the area of the segment be 300 pixels. Figure (3.7) shows graphically how the aggregation works. Both rules are combined with the "**product**" aggregation method:

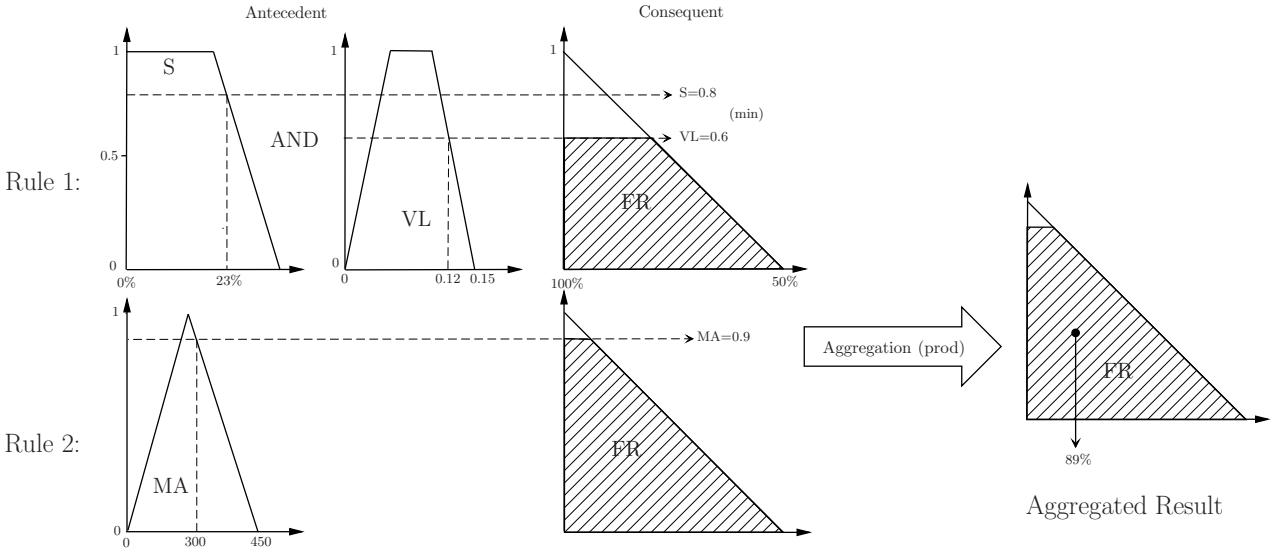


Figure 3.7: Aggregation Example

3.2.2 Spectral Information for DSM

Spectral information is widely used as a data source for mapping applications. Surface material information can be derived by traditional classification techniques from multispectral imagery and used for mapping of man-made structures and natural elements in complex urban scenes. Still the potential of spectral data is limited for these applications with respect to the accuracy and reliability of the results as well the possibility to discriminate a larger number of object categories (Haala & Brenner, 1999). A classic problem in multispectral data classification is the similar reflectance of grass-covered areas and trees. The same holds frequently true for streets and buildings. On the other hand, trees and buildings can be discriminated from grass-covered areas or streets using height data. For these reasons, imagery and height information can be used in a complementary way in order to enhance classification methodologies (Haala & Brenner, 1999).

This work proposes the orthorectification of the aerial imagery using as height information just the data from the segmented DSM (section 2.5.2). This aims to bring the spectral information contained on the aerial images to the LIDAR data. In this way, the height and roughness information can be combined with the spectral information to classify the segments obtained in the segmentation splitting process (section 2.6).

Orthophoto Production

Orthorectification aims to convert an image in the projective geometry into an orthogonal image, using the exterior orientation parameters and height information. Increasingly, digital orthophotos are employed as data in GIS. They are, therefore, an excellent orientation aid for GIS users (Kraus, 1996). The orthophoto production is implemented in an indirect way. First, a blank image is created in the orthophoto plane and corresponding matrix elements found in the reference image. Under the assumption of standard imaging conditions, the orthorectification process occurs as follows:

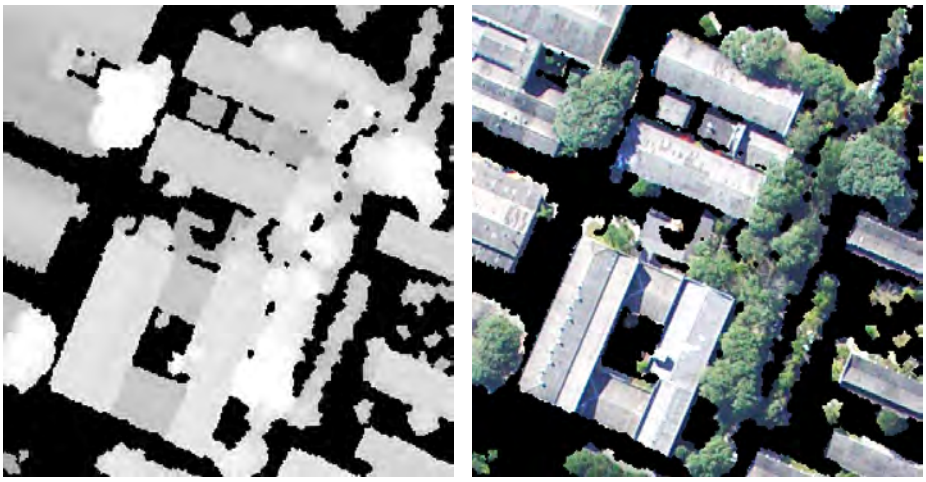
1. An image matrix is defined in the XY plane, called "orthophoto plane". The pixel spacing of the image matrix is usually significantly finer than the grid spacing of the terrain model. In this work, this spacing corresponds to the resolution of the DSM;

2. The Z coordinate of every pixel in the orthophoto matrix must be determined. In this work, no height interpolation is necessary, since the final orthophoto will have the same resolution as the DSM. This will give us the Z coordinates for all XY orthophoto pixels;

3. The image coordinates corresponding to the XYZ locations in the orthophoto grid are calculated using the collinearity equations (eq. 1.1) and the elements of interior and exterior orientation.

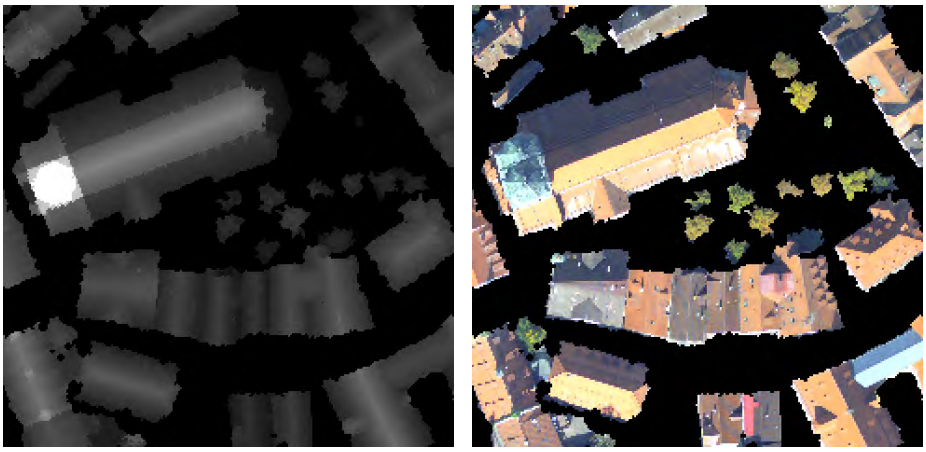
4. The gray value from the reference image corresponding to the photogrammetric coordinate pair is assigned to the corresponding XY position in the orthophoto matrix. Since the photogrammetric coordinate pair will not, in general, lie at the center of a pixel in the reference image, a gray value resampling is required. This work uses the bilinear resampling method.

Figures (3.8) and (3.9) show some examples, both with 0.5m GSD. Note the double mapping problem in the churches tower (fig. 3.9(b)), which occurs because of occlusions in the aerial imagery. This problem was not treated in this work.



(a) DSM Segmentation (b) Orthorectification

Figure 3.8: Orthorectification – UFPR



(a) DSM Segmentation (b) Orthorectification

Figure 3.9: Orthorectification – Biberach

3.2.3 Segment Properties

Inside each segment, specific properties for distinction of the relevant classes – buildings, vegetation and bare ground – are extracted. Rehor *et al.* (2008); Steinle (2005); Tovari (2006); Tovari & Vögtle (2004); Vögtle & Steinle (2003) proposed the following segment properties to perform the object classification: gradients on segment borders, height texture, first/last pulse differences, shape, size and laser pulse intensities.

Since LIDAR intensity images are quite noisy, this work uses as spectral data the information brought from the aerial imagery through orthorectification. This is the first real "data fusion" procedure, since both data sets are now fused to perform a better scene description. This work proposes also a different height texture parametrization (see section 2.6), which will be used in the classification. For the data sets used in this work, the first/last pulse differences show not significant and are not used. The proposed classification scheme also does not use the shape as parameter. It is important to mention that these not used parameters can be easily added in the future without great difficulties. The following sections show the proposed parametrization.

Green Index (G_i) and Neighborhood Green Index (NG_i)

Most digital cameras in use are not sensitive to NIR ($0.77\mu m - -0.89\mu m$), just to visible wavelengths: Blue ($0.45\mu m - -0.52\mu m$), Green ($0.52\mu m - -0.59\mu m$) and Red ($0.63\mu m - -0.69\mu m$) (INPE, 2009). It is expected that green areas represent vegetation. Departing from the split image, for every segment, a *green index* (G_i) and a *neighborhood green index* (NG_i) are calculated. The color analysis is performed on the LUV color space. Machado (2006) shows that the location of the green matices are dependent of the L component, and that they have negative U and positive V coordinates. Figure (3.10) shows an schematic example.

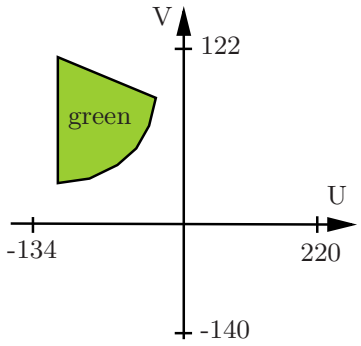
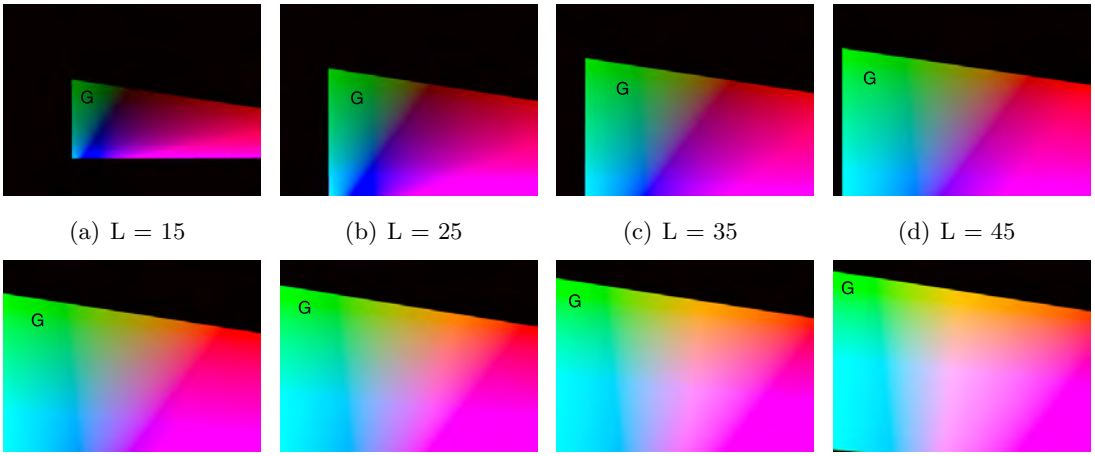


Figure 3.10: Green on LUV Color Space

This work divides the L component (which correspond to the luminosity and has a range from 0 to 100) in blocks of 10 units, starting from L=15 until L=85. The L=15 responds to the values from L=10 until L=20; the L=25 responds to L=20 until L=30 and so on. Values with L< 10 and L> 90 are ignored because it is not possible to clearly recognize "green colors". Then, a polygon containing the "green area" on UV plane for each L value is visually defined. Figure (3.11) presents the distribution of the colors for 8 L values. The "G" points the central green area.



(a) L = 15 (b) L = 25 (c) L = 35 (d) L = 45

(e) L = 55 (f) L = 65 (g) L = 75 (h) L = 85

Figure 3.11: Green Regions on the LUV Color Space

The "green test" consists in taking the RGB color from the orthorectified image, transforming these values into the LUV color space coordinates, and testing it based on the "green polygon" defined for the corresponding L block.

Let i be a pixel within a segment S with n pixels:

$$\forall\ i \in S \begin{cases} \textbf{if } i \text{ is green } \textbf{then } w_i = 1 \\ \textbf{else } w_i = 0 \end{cases}$$

The green index (G_i) is given by:

$$G_i = \frac{1}{n} \sum_{i=1}^{n} w_i \tag{3.3}$$

Next, the algorithm calculates the *neighborhood green index* (NG_i). Departing from the determined green indices, the algorithm searches for all pixels in the segment interface, similar as shown in figure 1.10(c), in chapter 1. For every pixel i in an interface with n pixels and a G_i green index, the neighborhood green index is given by:

$$NG_i = \frac{1}{n} \sum_{i=1}^{n} G_i \tag{3.4}$$

Performed tests show that actually few pixels are classified as green. In dense vegetation areas the G_i stays about 25%–30%. This effect is caused by noise in the image, bad radiometric sensor response (which are not radiometrically calibrated), low reflectance of vegetation in the visible wavelengths, and because the used image compression techniques. Many pixels stay in the edge between gray and green and are not correctly classified. G_i shows dependent of segment size: the bigger the segment, the better the probability of more green pixels to be correctly classified. This causes that large vegetation areas show good separation from large buildings, but for smaller segments, whose G_i have higher probability to miscalculated, worse results are expected.

Roughness Index (R_i)

In section (2.6) the segmented DSM was split into rough and smooth areas. Using a similar parametrization, it is possible to determine a *roughness index* for these segments, as discussed in section 3.2.1. Some rough areas are rougher than others, and the same occurs with smooth ones. For example, a smooth segment from a flat roof tends to be smoother than a smooth segment from a tree top. The roughness index is a weighted average based on the three-level classification shown in section (2.6).

Let i be a pixel within a segment S with n pixels and (τ_l,τ_h) the σ_0^2 thresholds defined in section 2.6:

$$\forall\ i \in S \begin{cases} \textbf{if } \hat{\sigma}_0^2 < \tau_l \textbf{ then } w_i = 0 \\ \textbf{if } \hat{\sigma}_0^2 \geq \tau_l \textbf{ and } \hat{\sigma}_0^2 \leq \tau_h \textbf{ then } w_i = 0.5 \\ \textbf{if } \hat{\sigma}_0^2 > \tau_h \textbf{ then } w_i = 1 \end{cases}$$

The roughness index (R_i) is given by:

$$R_i = \frac{1}{n} \sum_{i=1}^{n} w_i \tag{3.5}$$

Isolation Index (I_i)

Observing figures 2.9(c), 2.10(c), 2.11(c), 2.15(b) and 2.18(b) in chapter 2, it becomes clear that some segments are isolated (i.e. there are no interfacing neighbors), and that others lie completely inside another segment. This can be helpful during the classification, since it gives a better notion of the segment neighborhood. The isolation index (I_i) is determined based on the analysis of all interface pixels. Let i_{int} be an interface pixel within a segment S, N_i its 4-neighborhood, p_n a pixel within N_i and n the total number of inner interface pixels (interface pixels within the segment S):

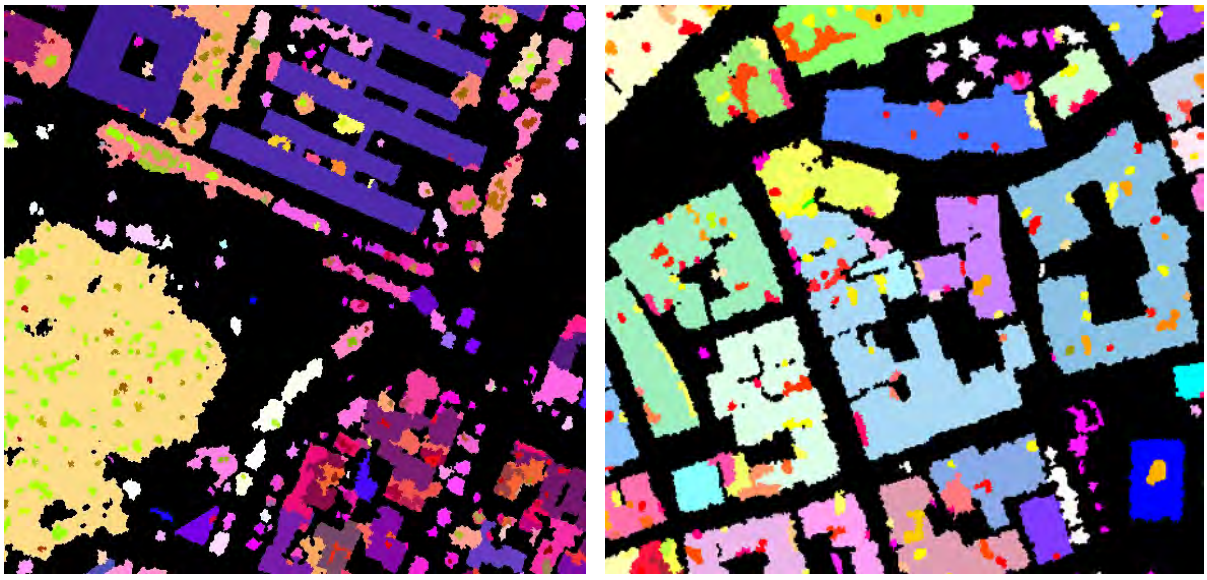
$$\forall\ i_{int} \in S \begin{cases} \textbf{if } \forall\ (p_n \in N_i) \nexists\ p_n \neq 0 \textbf{ then } w_i = 0 \\ \textbf{else } w_i = 1 \end{cases}$$

The isolation index (I_i) is given by:

$$I_i = \frac{1}{n_{int}} \sum_{i=1}^{n} w_i \tag{3.6}$$

3.2.4 Segment Classification

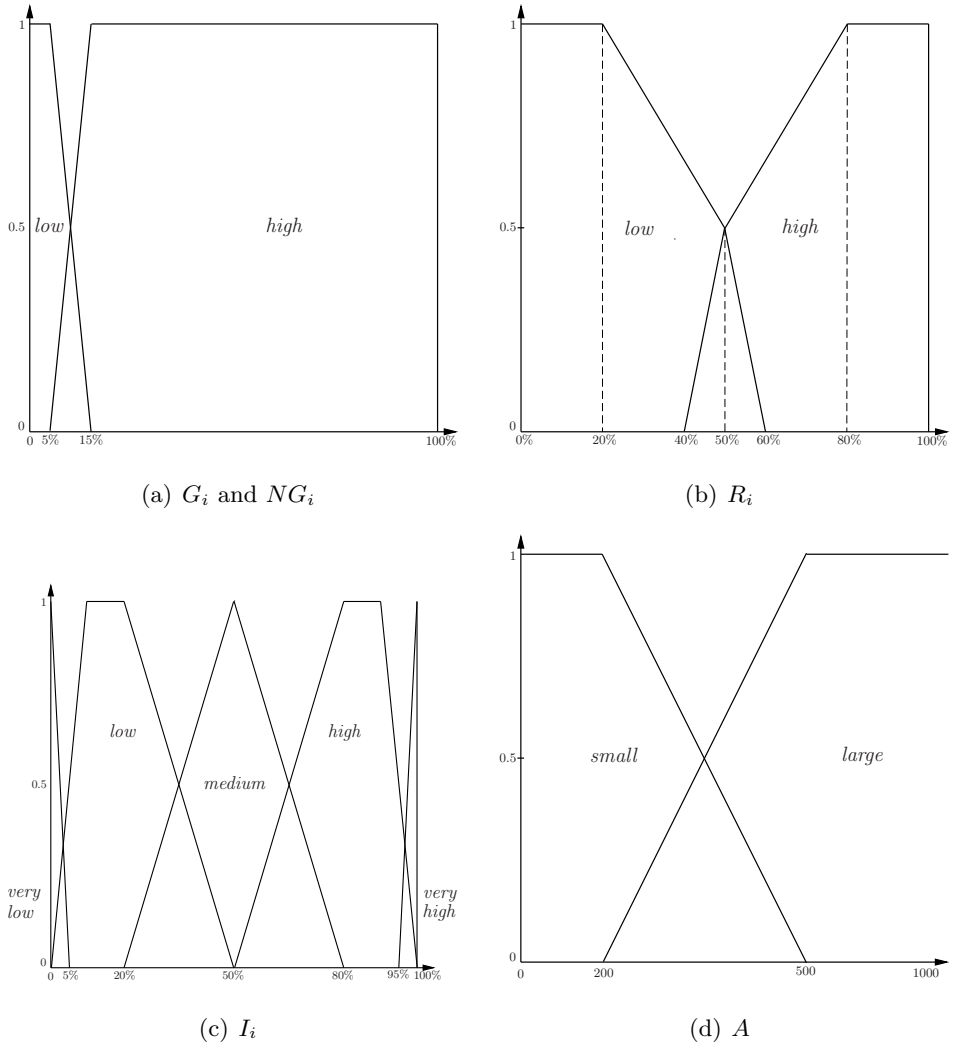
Urban environments have distinct natures, and data obtained from different sensors (laser scanners, digital cameras) have distinct properties. For every pair of urban environment and data configuration, an exclusive parametrization must be set in order to obtain a correct classification. This work focuses on usage of the proposed indices (G_i,NG_i,R_i and I_i) and the segment size (A) to perform separation from vegetation and building on the split DSM (section 2.5.2). These indices will be used as *fuzzy terms*. Other parameters (see section 3.2.3) can be easily added in future developments, given the fuzzy classification flexibility. Figure 3.12 shows RGB compositions of R_i G_i and I_i, respectively.



(a) UFPR (b) Biberach

Figure 3.12: Indexed Split Images

In order to proceed with the fuzzy inference schema as presented in section 3.2.1, it is necessary to create the membership functions for the fuzzy terms. The proposed antecedent membership functions (see section 3.2.1) are:



(a) G_i and NG_i (b) R_i

(c) I_i (d) A

Figure 3.13: Antecedent Membership Functions

It is also necessary to define the consequent membership functions (see section 3.2.1). To separate vegetation from building, this work proposes a *vegetation ratio* (R_{veg}). If R_{veg} is high, then the segment is classified as vegetation, otherwise as building. Figure (3.14) shows the implemented membership functions.

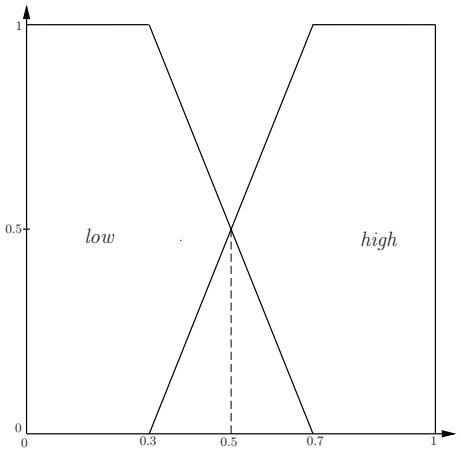
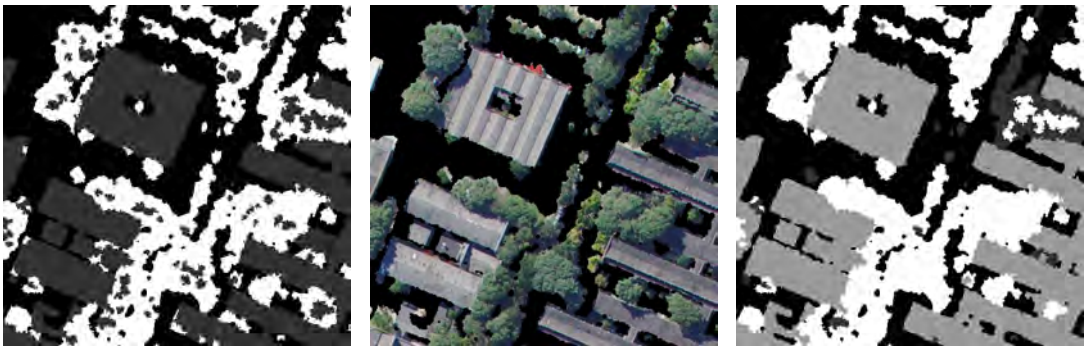


Figure 3.14: Consequent Membership Function R_{veg}

Given the membership functions (figures 3.13 and 3.14), it is necessary to determine the fuzzy rules:

1: IF I_i **IS** *high* **AND** R_i **IS** *high* **AND** G_i **IS** *high* **THEN** R_{veg} **IS** *high*;
2: IF I_i **IS** *high* **AND** R_i **IS** *low* **AND** G_i **IS** *low* **THEN** R_{veg} **IS** *low*;
3: IF I_i **IS** *low* **AND** R_i **IS** *high* **AND** G_i **IS** *low* **THEN** R_{veg} **IS** *low*;
4: IF I_i **IS** *very low* **AND** NG_i **IS** *high* **THEN** R_{veg} **IS** *high*;
5: IF I_i **IS** *very low* **AND** NG_i **IS** *low* **THEN** R_{veg} **IS** *low*;
6: IF I_i **IS** *very high* **AND** R_i **IS** *high* **AND** A **IS** *large* **AND** G_i **IS** *high* **THEN** R_{veg} **IS** *high*;
7: IF I_i **IS** *very high* **AND** R_i **IS** *low* **AND** A **IS** *large* **AND** G_i **IS** *low* **THEN** R_{veg} **IS** *low*;
8: IF NG_i **IS** *high* **AND** I_i **IS** *very high* **AND** A **IS** *small* **THEN** R_{veg} **IS** *high*;
9: IF NG_i **IS** *low* **AND** I_i **IS** *medium* **AND** G_i **IS** *low* **THEN** R_{veg} **IS** *low*;

Figures (3.15) and (3.16) show examples of obtained results. Figures 3.15(c) and 3.16(c) show the classification. Light gray areas mean edification, white areas vegetation, dark gray are non-classified areas and black bare ground.



(a) Split Image (b) Orthorectification (c) Classification

Figure 3.15: Fuzzy Classification Example – UFPR

(a) Split Image (b) Orthorectification (c) Classification

Figure 3.16: Fuzzy Classification Example – Biberach

Making a visual analysis using figure 3.15(b) as reference, most small "smooth isles" present within large vegetation areas in figure 3.15(a) were correctly classified as vegetation. On the other hand, making the same analysis using figure 3.16(b) as reference, most small "rough isles" present within large buildings were correctly classified as edification. See more visual and statistical analysis in chapter 4.

3.3 Geometric Data Fusion: The Roof Detection Case

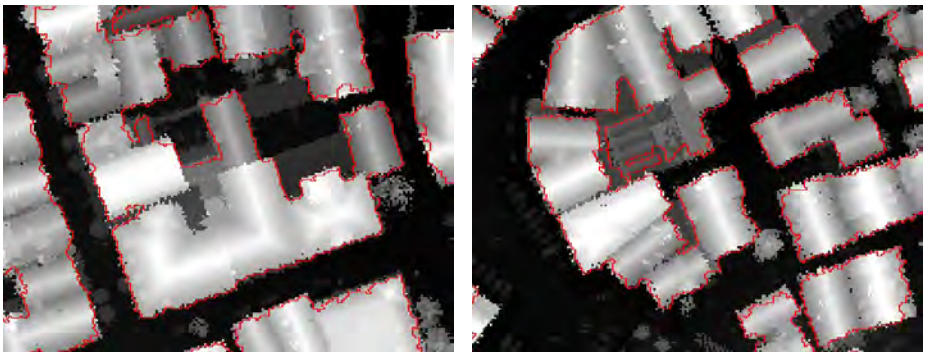
This section presents geometric aspects of the proposed data fusion methodology, which focuses on building detection.

3.3.1 Roof Detection from Segmented Aerial Imagery

Aim of this section is to obtain building data from segmented aerial images. These images are segmented in the photogrammetric reference system (see section 1.2). Since image orientation parameters are known and there is building information available from the fuzzy classification (section 3.2.4), it is possible to deal with both reference systems – the photogrammetric reference system and the geodetic reference system – trough the collinearity equations (1.1). The proposed detection procedure has 3 steps: the vectorization and projection of the detected buildings; the creation of a binary image from the vectorized buildings in the photogrammetric reference system; the classification of the aerial imagery segments.

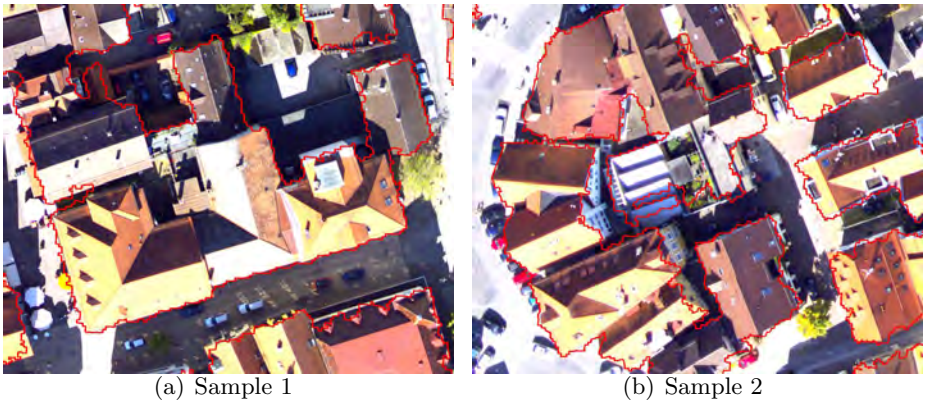
Building Vectorization and Projection

Building vectorization is a process to transform raster data (in this case the classified image) into vector data (see section 2.2). Using a vectorizing technique adapted from (Burger & Burge, 2007), 2D polygons from the classified buildings are stored in the database.



(a) Sample 1 (b) Sample 2

Figure 3.17: Vectorized Buildings and DSM (Object Space)



(a) Sample 1 (b) Sample 2

Figure 3.18: Projected Buildings (Image Space)

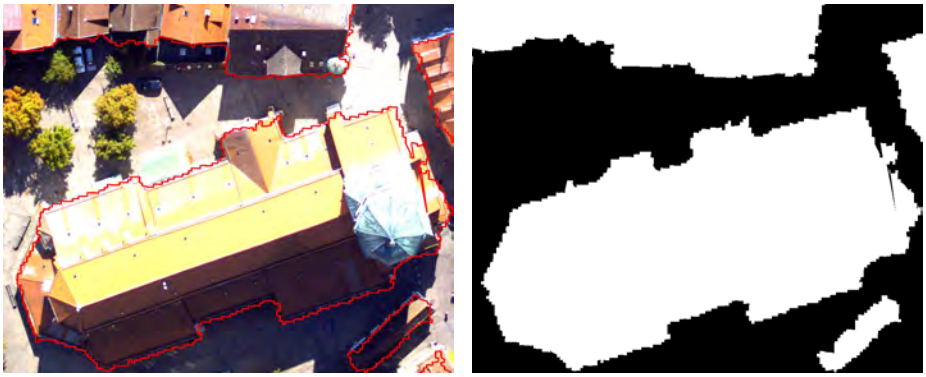
Simultaneously to the vectorization in the object space, the polygon is also created in the image space by applying the collinearity equations to every polygon vertex. The altimetric information comes from the DSM. Departing from this result, it is possible to test all segments and determine if they belong to a building or not. Figure 3.17 shows an the vectors (in red) plotted over the DSM. Figure 3.18 shows 2 examples in the photogrammetric reference system. Figures 3.18(a) and 3.18(b) were rotated by 180° in order to make the comparison with figures 3.17(a) and 3.17(b) simpler.

Building Binary Image Creation

Due to its high resolution and complexity, aerial imagery segmentation produces huge amounts of segments per image. In order to classify these segments, it is necessary to test them against the building polygons. The natural solution would be to automatically vectorize the segments and test them against the building polygons. However, this solution shows very bad performance, making this procedure unviable.

To overcome this issue this work proposes to create a *building binary image*. This image translates the segments information given by the fuzzy classification to the image space. The proposed solution consists in creating a *vector aerial image table* in the geodatabase, inserting a point for each pixel in the photogrammetric reference system in this table, and then indexing it as shown in section (2.2.2). Next, the algorithm takes the building polygons one by one, rapidly finds all "pixel-points" within the polygon and sets for each "pixel-point" the value 1 in the binary image.

The creation of the vector aerial image table has a long processing time due to the number of points to be created and inserted. However, this creation proceeds just once per camera. The table can be used multiple times and for any image taken with the respective camera. Another advantage of this approach is that complex overlay operations are avoided, which still lead to errors in the PostGIS (section 2.2.3) environment. Figure (3.19) shows an example.



(a) Building Vectors and Original Image (b) Building Binary Raster

Figure 3.19: Binarizing Example

Segment Indexing and Classification

The objective now is to define which segments are building parts. In order to achieve this, for every segment a *building index* (B_i) is calculated, using as reference the building binary image.

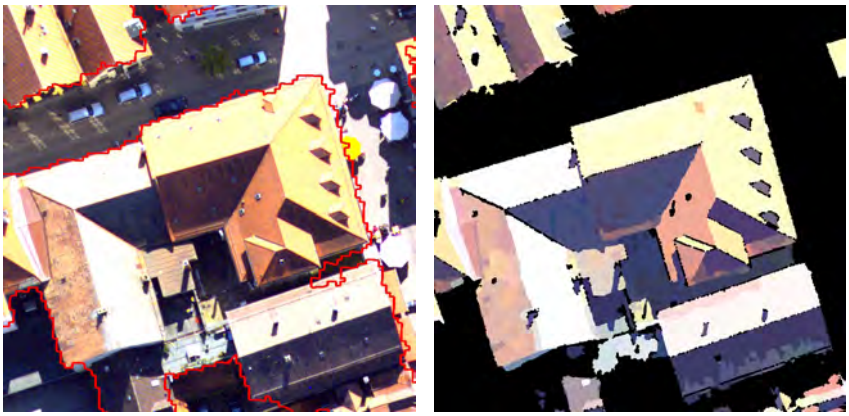
Let i be a pixel within a segment S with n pixels and I_{bin} the binary image:

$$\forall\ i \in S \begin{cases} \textbf{if } i = 1 \text{ in } I_{bin} \textbf{ then } w_i = 1 \\ \textbf{else } w_i = 0 \end{cases}$$

The building index (B_i) is given by:

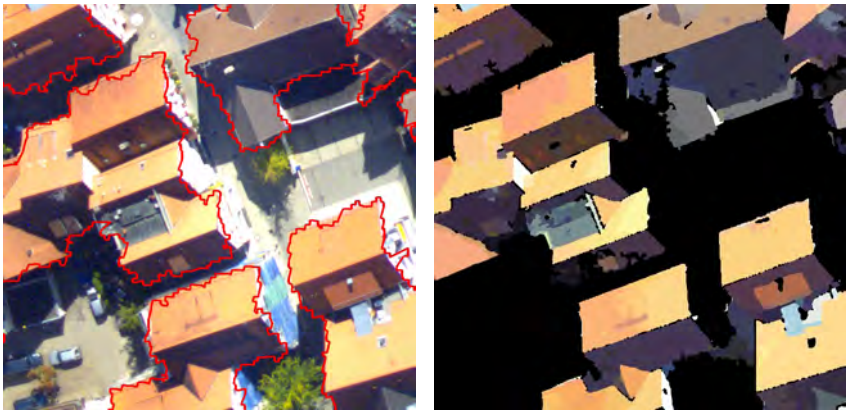
$$B_i = \frac{1}{n} \sum_{i=1}^{n} w_i \tag{3.7}$$

B_i indicates to what extent a segment lies inside a building. The next step consists in setting a threshold τ_b and selecting all segments which $B_i > \tau_b$. In other words, the specialist sets a limit to define a segment as building or not. Figures (3.20) and (3.21) show examples with $\tau_b = 0.5$ (all segments with at least 50% inside a building).



(a) Polygons and Original Image (b) Classified Building Segments

Figure 3.20: Classification Example 1



(a) Polygons and Original Image (b) Classified Building Segments

Figure 3.21: Classification Example 2

Note in figure (3.20) how the classified segments show a better building definition than just the polygon from the LIDAR classification. Figure (3.21) shows correctly detected roofs. Polygons from LIDAR classification show useful to detect roofs with good definition. Observe also that illumination plays a major role to a correct building detection: dark segments present worse results than well illuminated ones.

3.3.2 Building Footprint Monoplotting and Refinement

The procedure presented in section (3.3.1) takes place in the image space, but we are actually interested in features in the object space. This section proposes a solution to map the results from the segmentation classification results back to the geodetic reference system, using the inverse collinearity equations:

$$\begin{aligned} X &= X_0 + (Z - Z_0) \cdot \frac{m_{11}x + m_{21}y - m_{31}c}{m_{13}x + m_{23}y - m_{33}c} \\ Y &= Y_0 + (Z - Z_0) \cdot \frac{m_{12}x + m_{22}y - m_{32}c}{m_{13}x + m_{23}y - m_{33}c} \end{aligned} \tag{3.8}$$

Where (m_{11} ... m_{33}) are the elements of the matrix M (see eq. 1.2). The planimetric coordinates (X, Y) of a point in the object space are given as functions of their coordinates in image space (x, y), its altimetric coordinate Z and the exterior orientation parameters $(\kappa, \varphi, \omega, X_0, Y_0, Z_0)$. Makarovic (1973) conceived the *monoplotting*, which consists of producing planimetric information through the direct vectorization of features on a photographic image and rectifying it by utilizing the inverse collinearity equation. Although, limited computation power made these proposals stay forgotten for many years. Mitishita (1997) presented a monoplotting system using the principles proposed by (Makarovic, 1973) using a DTM as altimetric source. Jauregui *et al.* (2002) proposed a similar solution. The use of LIDAR data as altimetric data source in order to perform monoplotting was presented in Mitishita *et al.* (2004), and Machado (2006) proposed some automatic improvements to the methodology. This work proposes the use of the segment borders (inner interface pixels) obtained in section 3.3.1 as building roof edges. This determines the (x, y) coordinates. Since the EOP are known, the objective now is to define the Z coordinates for the roof edges, and then apply the equations (3.8). The next sections propose a Z determination methodology, a building footprint detection procedure, and a refinement technique to produce building polygons.

Roof LIDAR Data Projection

LIDAR data sets are very rich in altimetric information, and this section shows a methodology to detect roof information from the LIDAR point cloud and transform it to the image space. It is possible to simply use the classified DSM as altimetric data source, but after many processing steps, it is possible that altimetric information have lost accuracy.

(a) Example 1 (b) Example 2 (c) Example 3

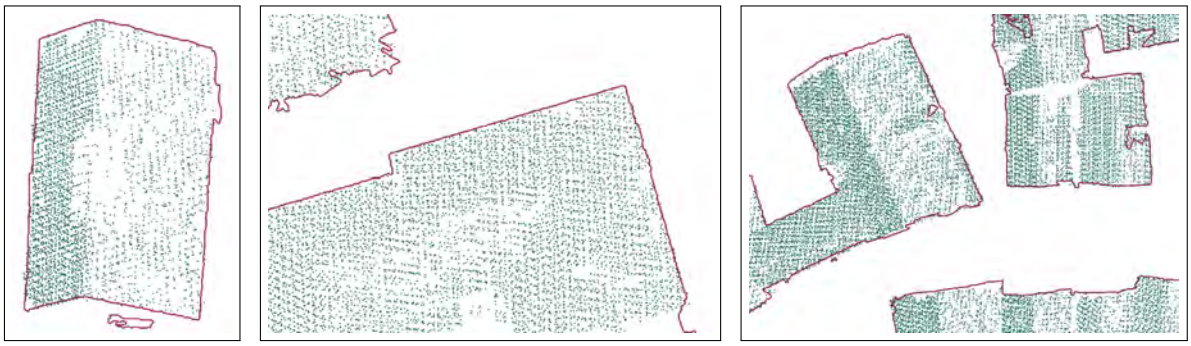
Figure 3.22: Projected Roof LIDAR Points

Departing from the building polygons in object space, all LIDAR points within the polygons are selected. Then, each selected LIDAR point is tested with the DTM, in order to verify if this point really is a "high point", exactly as done in section (2.5.2). If the point passes the test, the photogrammetric coordinates (x, y) are calculated using the collinearity equation (1.6), and a hybrid point with coordinates (x, y, Z) is stored in a table. At the end the table is indexed and clustered to permit fast access (see section 2.2.2). Figure (3.22) shows some examples.

Building Footprint Detection and Rectification

Buildings footprints are projections of the roofs on the ground. In this work, building roofs are determined in the image space, with the classification procedure proposed in section 3.3.1. The detected roofs are then used to determine building footprints. The first step is to analyze the classified segmentation and, for every pixel on the segment borders, a point with its (x, y) coordinates is stored in a table in the database. After the point collection, this table is clustered and indexed. Figure (3.23) shows the detected roofs (purple points) plotted over the original image.

Figure 3.23: Detected Borders (Image Space)



(a) Example 1 (b) Example 2 (c) Example 3

Figure 3.24: Laser Points and Roof Points (Image Space)

Next, it is possible to integrate the laser scanner information stored on the hybrid reference system determined in section (3.3.1) with the edge points obtained from the segmented aerial imagery. Figure (3.24) presents 3 examples of this integration. Purple points represent the detected edges and green points are the LIDAR data. The 3 areas are marked in red in figure (3.23).

It is necessary now to determine the coordinates of the roof edges in the object space. Mitishita *et al.* (2004) show the viability of non-iterative rectification of edification edges using LIDAR data as altimetric reference. This paper uses the Z from the nearest LIDAR point to calculate de planimetric coordinates using the inverse collinearity equations (see equation 3.8). Our work proposes also, in order to avoid extrapolations, a maximum search distance for the nearest laser point. Figure (3.25) presents an example of the proposed methodology, using a maximum distance of 10 pixels. The roof edge points rectified to the object space are plotted in red over the DSM.

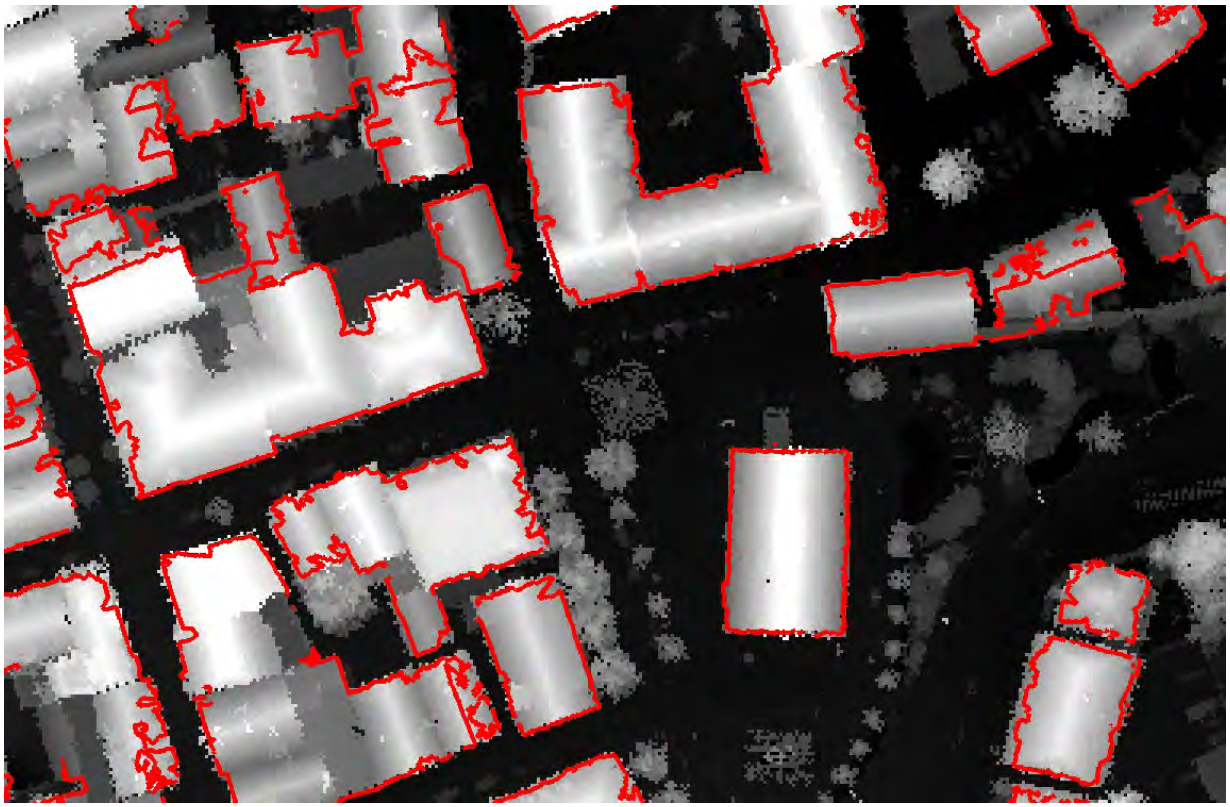
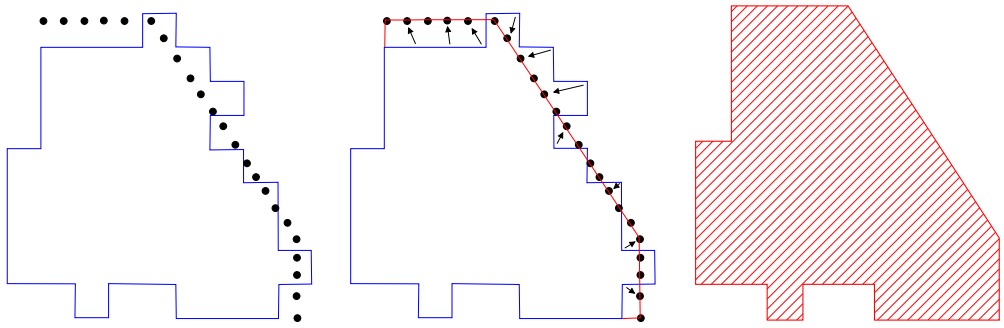


Figure 3.25: Detected Borders at the Object Space

Building Polygons Generation

Building information from aerial imagery is now available in form of points in the object space. In most GIS applications, buildings are represented as polygons. This work proposes a building polygon adjustment methodology that uses as start point the building polygons vectorized in section (3.3.1). The vertices of these polygons are analyzed, and if a rectified edge point is found within a given search radius, the vertex planimetric coordinates are translated to the rectified edge point. This solution was developed in order to keep original building information detected in section (3.3.1) in case no rectified edge point is available. That can happen in case of failures in the roof detection step (section 3.3.1). Figure (3.26) shows the proposed building adjustment technique. Figure 3.26(a) shows the original building polygon and the rectified edge points. Figure 3.26(b) shows the polygon adjustment by vertex translation. Figure 3.26(c) shows the refined polygon.



(a) Polygon and Edge Points (b) Polygon Adjustment (c) Refined Polygon

Figure 3.26: Building Polygon Refinement

Figure (3.27) shows a practical example of the proposed methodology. The polygons are plotted over the DSM in red.

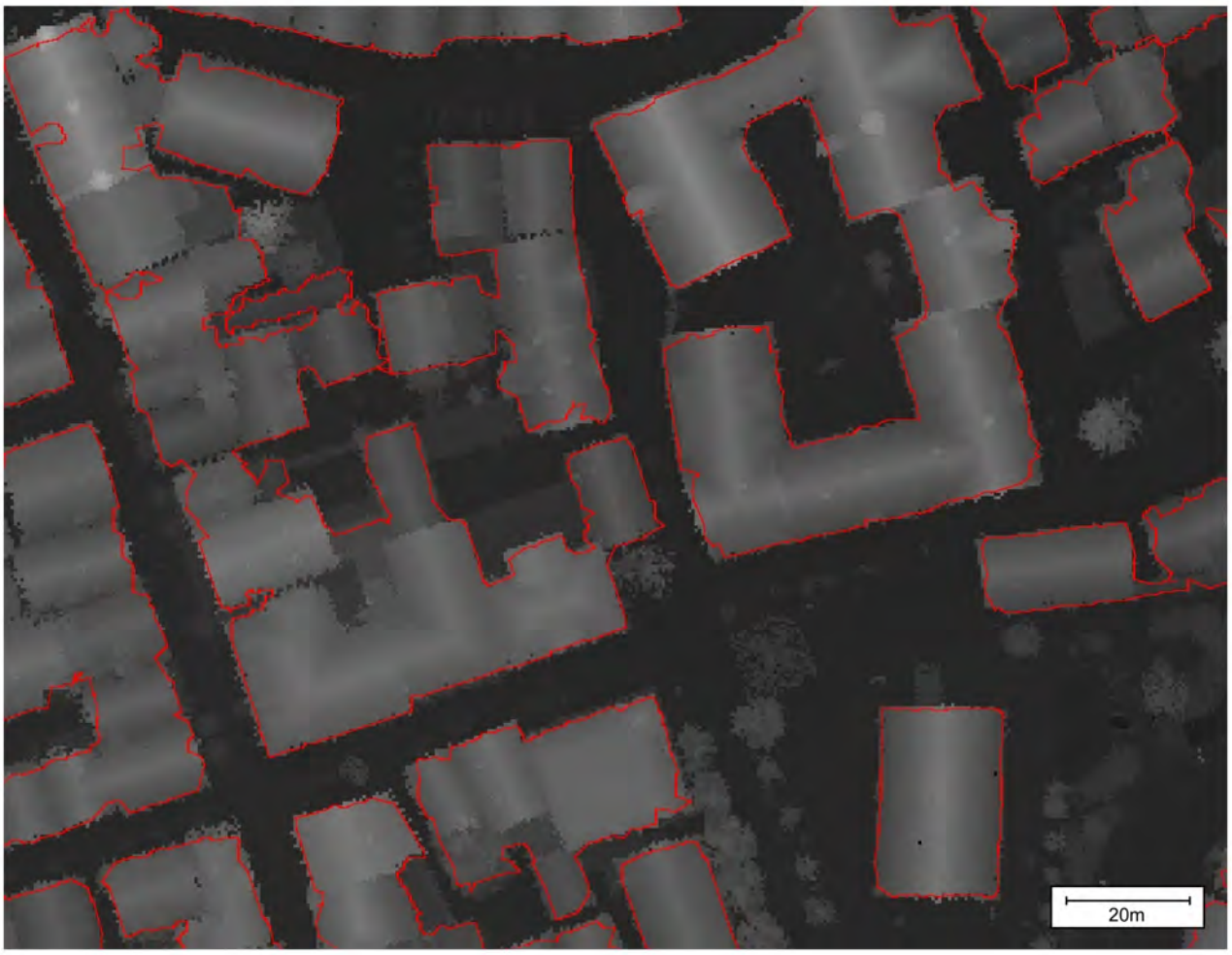


Figure 3.27: Refined Building Polygons

Both data sets have been fused: building information from the LIDAR classification and aerial imagery segmentation are used together to make a better building description. More detailed analysis in chapter 4.

Chapter 4

Tests and Results

4.1 Methodology Overview

Chapters 1, 2 and 3 presented a set of methodologies to process aerial imagery and LIDAR data in order to fuse these pieces of information. This chapter discusses the obtained results of two data sets. One data set was obtained by surveying the Centro Politécnico Campus of the Universidade Federal do Paraná (UFPR), in Curitiba, Brazil. The other data set was obtained by surveying the downtown area of Biberach, in Baden-Würtemberg, Germany. Both data sets consist of aerial imagery and LIDAR data. Figure (4.1) shows the methodology overview at a glance.

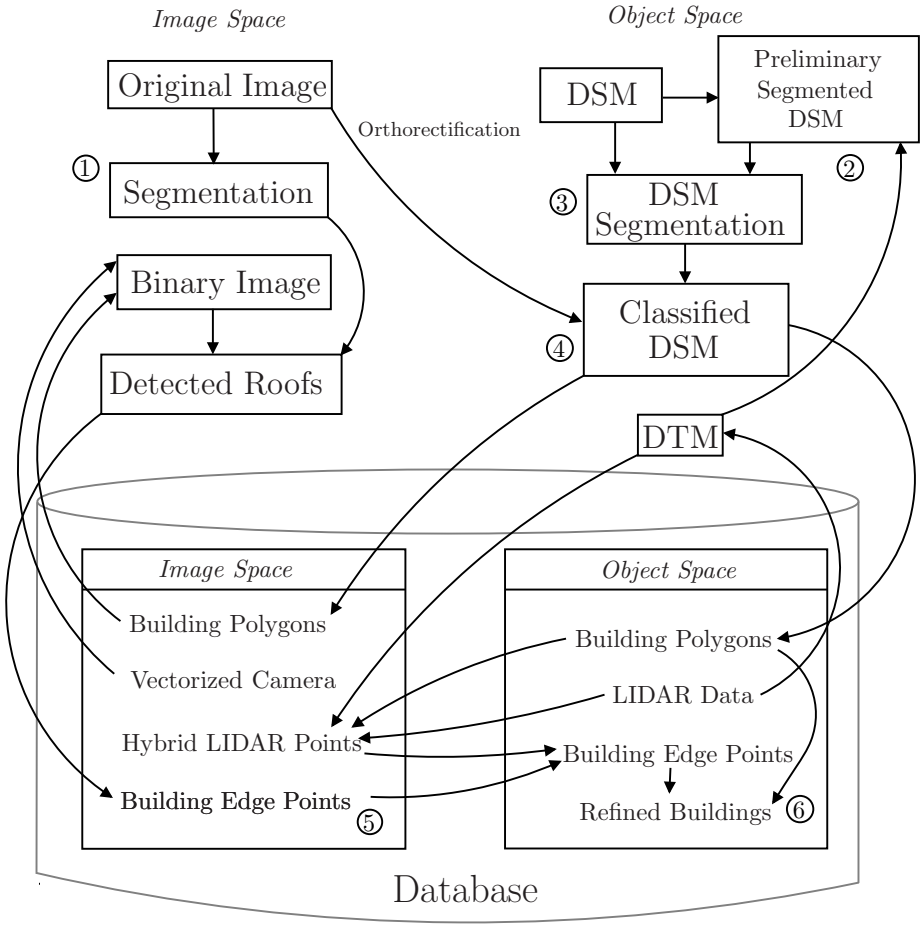


Figure 4.1: Methodology Overview

Steps marked with numbers ① to ⑥ are the key processing steps. Their results will be presented and discussed in this chapter. These steps are:

① **Segmentation:** Results from the aerial image segmentation with the proposed SRG methodology are shown in section (1.3.2).

② **Preliminary Segmented DSM:** Results from the DSM segmentation algorithm are presented in section (2.5.2).

③ **Segmented DSM:** Results of the $\hat{\sigma}_0^2$ based split methodology are presented in section (2.6).

④ **Classified DSM:** Results of the fuzzy classification methodology are presented in section (3.2.4).

⑤ **Building Edge Points:** Results of the roof detection methodology in the image space are presented in section (3.3.1).

⑥ **Refined Buildings:** Results of the building monoplotting and refinement are proposed in section (3.3.2).

This chapter is organized as follows. Section (4.2) presents results of the processing in image space. Section (4.2.1) presents a visual analysis of the segmentation (①) and roof detection (⑤). Section (4.2.2) presents a statistical analysis of the roof vectorization (⑤). Section (4.3) presents results obtained in the object space. Section (4.3.1) presents a visual analysis of both datasets concerning the steps ②,③,④ and ⑥. Section (4.3.2) presents a statistical analysis of the fuzzy classification (④) and the obtained buildings (⑥).

4.2 Image Space Analysis

Image segmentation results are difficult to be statistically analyzed (Neubert *et al.*, 2006), but through visual analysis it is possible to detect weaknesses and strengths. Section (4.2.1) proceeds a visual analysis of the segmentation obtained with the images from both data sets (UFPR and Biberach, step ①). Section (4.2.2) presents a statistical analysis of the proposed vectorization technique (⑤)

4.2.1 Segmentation and Roof Detection Visual Analysis

This section presents a segmentation and roof detection visual analysis from both datasets. First, two of 13 images from the UFPR block (section 1.2.2) flown on June 27, 2003 with the Sony DSC-F717 – fig. 1.1(a) camera over the Centro Politéctico Campus of the UFPR in Curitiba is presented. Second, tests with an image taken in September 2006 with the 22Mpix Applanix DSS-22M – fig. 1.2(a), taken over the Biberach downtown in Germany are shown.

UFPR Images

Figure (4.2) shows the image 195 from the UFPR block. The analyzed areas are drawn in the image. Figures (4.3) to (4.7) show results in detail.

Figure 4.2: UFPR Block – Photo 195 (Original)

(a) Original (b) Segmentation (c) Detected Roofs

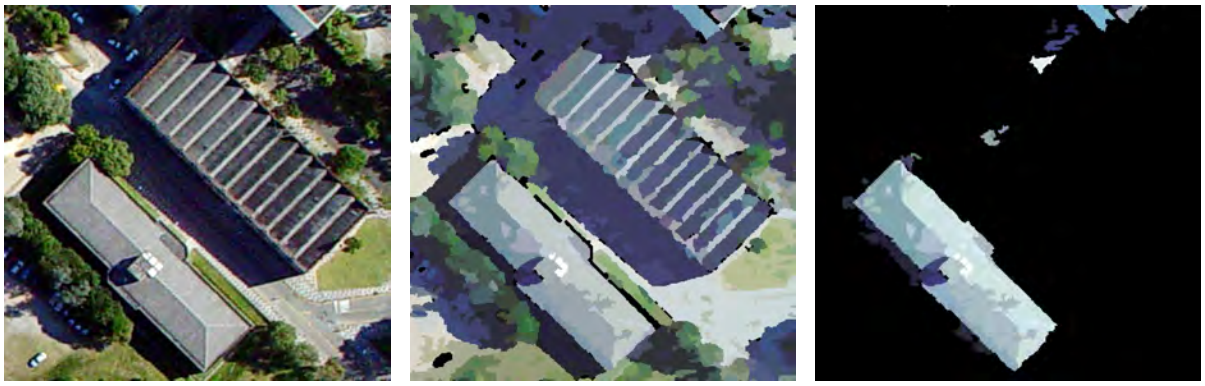
Figure 4.3: Area 1 – Image Space – UFPR

(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.4: Area 2 – Image Space – UFPR

Figure (4.3) shows a residential area. Note that edification close together build “edification blocks”. Most roofs were correctly identified, but shadows make some buildings show incomplete.

Figure (4.4) shows a single complex building. Most roof edges are correctly determined. Some shadows in the upper left corner of the building were classified as high segment due to image orientation problems. This example illustrates the edge conserving principle of the mean-shift segmentation.



(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.5: Area 3 – Image Space – UFPR

Figure (4.5) shows 2 buildings. One was correctly detected and the other was not. The reason was wrong classification of the upper right building during the fuzzy classification step. Once again, edges show well defined. A good separation between building and vegetation is shown.

Figure (4.6) shows another complex building with large trees nearby. Most roof segments were correctly detected. Also roofs in the shadow were correctly detected.

(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.6: Area 4 – Image Space – UFPR

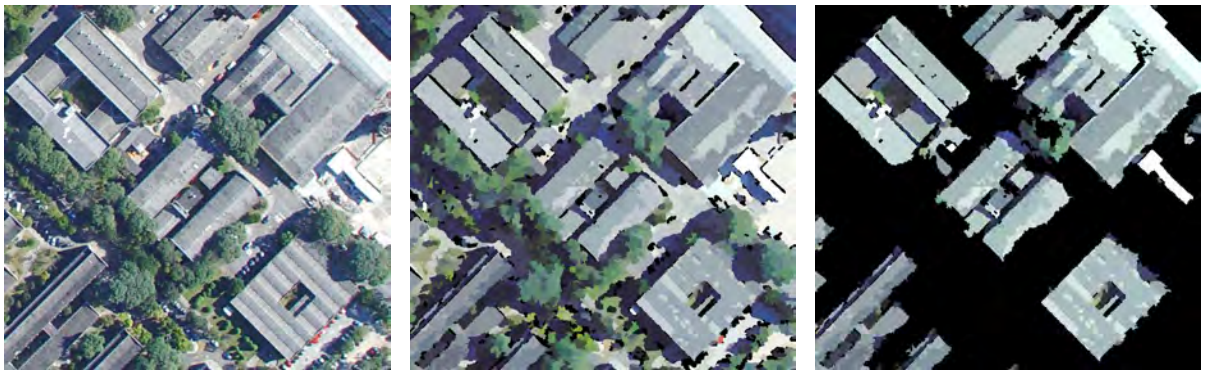
(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.7: Area 5 – Image Space – UFPR

Figure (4.7) shows 3 buildings and some isolated trees close to them. Two small green areas were detected as roofs. A part of the central building was confused with the ground during the segmentation and was misclassified. Some small shadow segments were misclassified as roof due to image orientation problems. These orientation problems occurred due to problems in the LIDAR data used as reference for the aerotriangulation. Some strips show deformations, and since the control points come from several strips, these deformations reflected in the exterior orientation parameters of the images.

Figure (4.8) shows the image 197 from the UFPR block. The analyzed areas are drawn in the image. Figures (4.9) to (4.13) show the results in detail.

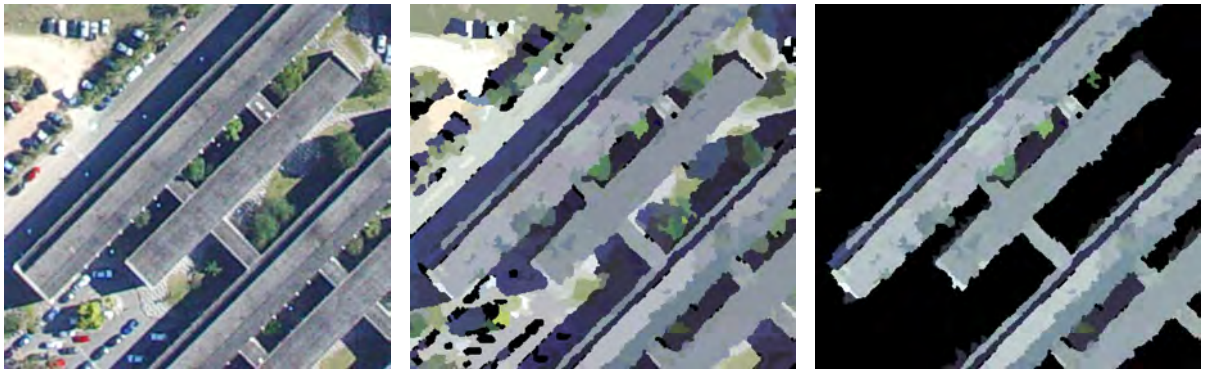
Figure 4.8: UFPR Block – Photo 197 (Original)



(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.9: Area 6 – Image Space – UFPR

Figure (4.9) shows a very mixed area with high trees and buildings in between. The segmentation shows again to conserve the edges as expected. Most roof areas were correctly detected, just one tree was misclassified as roof. The internal building area detection has yet to be implemented. The edges show very well defined.



(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.10: Area 7 – Image Space – UFPR

Figure (4.10) shows some connected buildings with some small trees close to them. The edges do not show very sharp due to small shadow segments misclassified as roof. This happens because of image orientation problems and/or bad fitting of laser strips.

Figure (4.11) shows 3 isolated buildings. Two of them were correctly detected. The upper right one has a part misclassified, due to its complexity. See figure (4.30) in page 108 for more details.

Figures (4.12) and (4.13) show residential areas. A good number of roofs were correctly detected – see fig. 4.12(c). The mixture of high trees and smaller buildings produces bad results. Some roofs are just partially detected if the trees shadows are too dark. Also LIDAR data resolution plays an important role, since the smaller the roofs, the higher the point density necessary to produce correct results.

(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.11: Area 8 – Image Space – UFPR

(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.12: Area 9 – Image Space – UFPR

(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.13: Area 10 – Image Space – UFPR

Biberach Image

Figure (4.14) shows the Biberach image. This image has a 22Mpix resolution and was taken with a metric mid-format camera, so the results will show better than in the UFPR examples. The analyzed areas are drawn in the image. Figures (4.15) to (4.20) show the results in detail.

Figure 4.14: Biberach Image

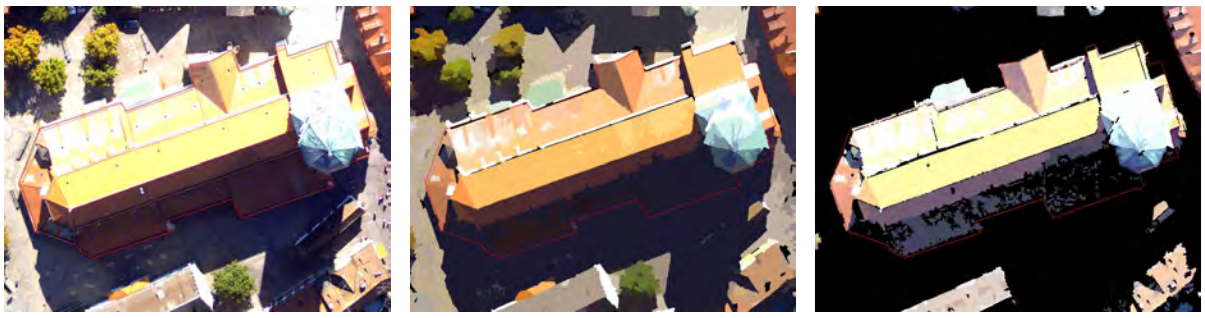
(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.15: Area 1 – Image Space – Biberach

(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.16: Area 2 – Image Space – Biberach

Figures (4.15) and (4.16) show some correctly detected roofs. The Biberach LIDAR data set has a higher point density, and for this reason better results are expected. Also smaller roofs can be detected: note some small roof parts in figure (4.16), lower left corner. As expected, regions with shadows tend to show bad results due to insufficient radiometric resolution and bad signal noise ratio. The bandwidth (see section 1.3.1) calculated for the segmentation is higher then the color differences in these areas, so the regions tend to grow more than expected and cause some roof detection problems.



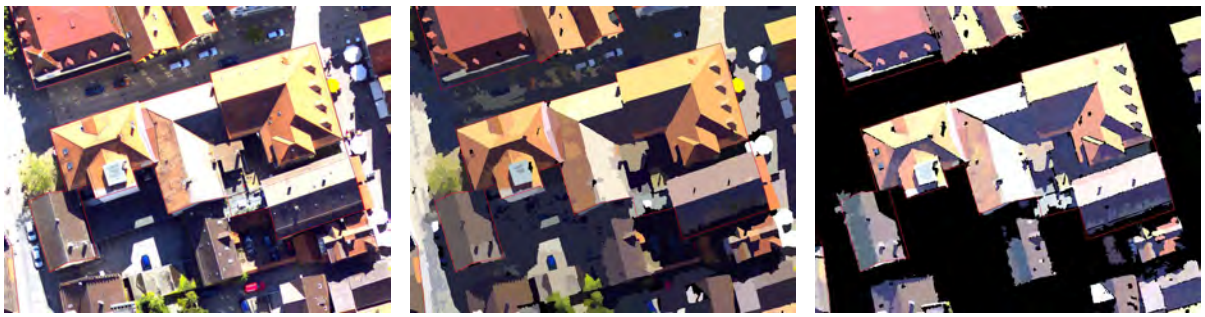
(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.17: Area 3 – Image Space – Biberach

Figure (4.17) shows the Biberach church. Here, the effects of shadow are exemplary. The bright side is well segmented and detected. The dark side shows a confusion between ground and dark roofs due to bad illumination.

(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.18: Area 4 – Image Space – Biberach



(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.19: Area 5 – Image Space – Biberach

Figures (4.18) and (4.19) show complex sets of roofs that were correctly detected. It becomes clear that high LIDAR point density and good illumination condition lead to the best results. Note, particularly, the almost perfectly detected roof edge in fig. 4.18(c).

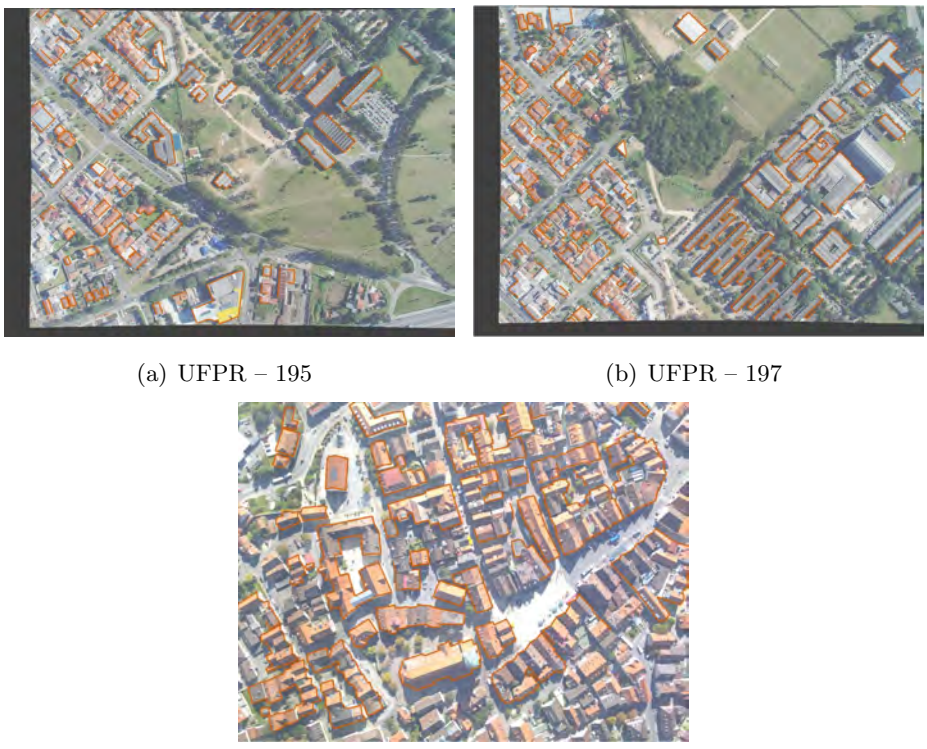
(a) Original (b) Segmentation (c) Detected Roofs

Figure 4.20: Area 6 – Image Space – Biberach

Figure (4.20) shows another complex set of buildings. Note that, again, illumination conditions play a major role in the segmentation and roof detection: some dark roofs have very similar spectral response as shaded bare ground areas. This issue causes confusion and misclassification.

4.2.2 Vectorization Statistical Analysis

The detected roof segments are used to perform automatic roof vectorization (see section 3.3.2). This section presents a vectorization quality analysis using as reference a set of well defined building edges manually vectorized. Figure (4.21) shows the reference vectors (in orange) for the three tested images.



(a) UFPR – 195 (b) UFPR – 197

(c) Biberach

Figure 4.21: Roofs Manually Vectorized

The building edges are detected pixel by pixel in form of points (see section 3.3.2). On the other hand, the manual vectorized edges are represented by lines. In order to make both data compatible, the lines are split into points with 1 pixel resolution. These will be the reference points. Given a search radius, the algorithm searches for every reference point, the nearest neighbor between the automatically vectorized points. If a point is found, we can say that for a given precision (the search radius), the automatic point is correctly determined. The relation between the correct points found and the total number of reference points is used as parameter for the analysis. This work analyzes 10 search radii: 0.5, 1, 1.5, 2, 3, 5, 10, 15, 20 and 25 pixels. Figure (4.22) shows the obtained results:

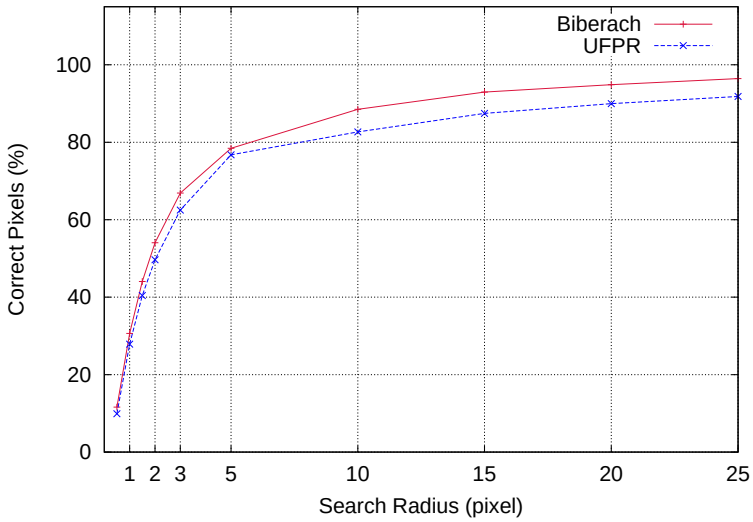


Figure 4.22: Vectorization Quality Analysis

Note that the Biberach image shows systematically better results than the UFPR images. This quality difference was already noticed in the visual analysis (section 4.2.1). A key value in the graphic is the 10 pixel mark, since it was the value used in the point edge rectification (see section 3.3.2). The Biberach image shows a performance 6% better than the UFPR images. Considering an 8cm GSD for the Biberach image and a 25cm GSD for the UFPR images, the approximate difference, in m, for 10 pixels is about 1.70m. For a 5 pixel search radius, both data sets show about 80% correctness, but considering the GSD of both images, the Biberach data set shows results 85cm better than the UFPR data set. This causes the building footprints from the Biberach data set to be better determined than in the UFPR data set.

4.3 Object Space Analysis

The UFPR LIDAR data set consists in 5 flight strips containing about 15 million points, flown with an Optech ALS 2050 Laser Scanner, fig. 1.1(b), on May 9, 2003. Figure (4.23) shows a cut of the generated DSM. The Biberach LIDAR data set consists in 8 flight strips containing about 14 million points, flown with a Toposys Harrier 56 fig. 1.2(b), Laser Scanner in September 2006. Figure (4.24) shows a cut of the generated DSM. Both images were created with the modified Araki algorithm presented in section (2.5.1). The GSD is 50cm for both images. The red areas are presented in detail in section (4.3.1). Note that the Biberach DSM (fig. 4.24) shows upside down in relation to the original image (fig. 4.14) due to the flight direction. A similar effect occurs in the UFPR block.

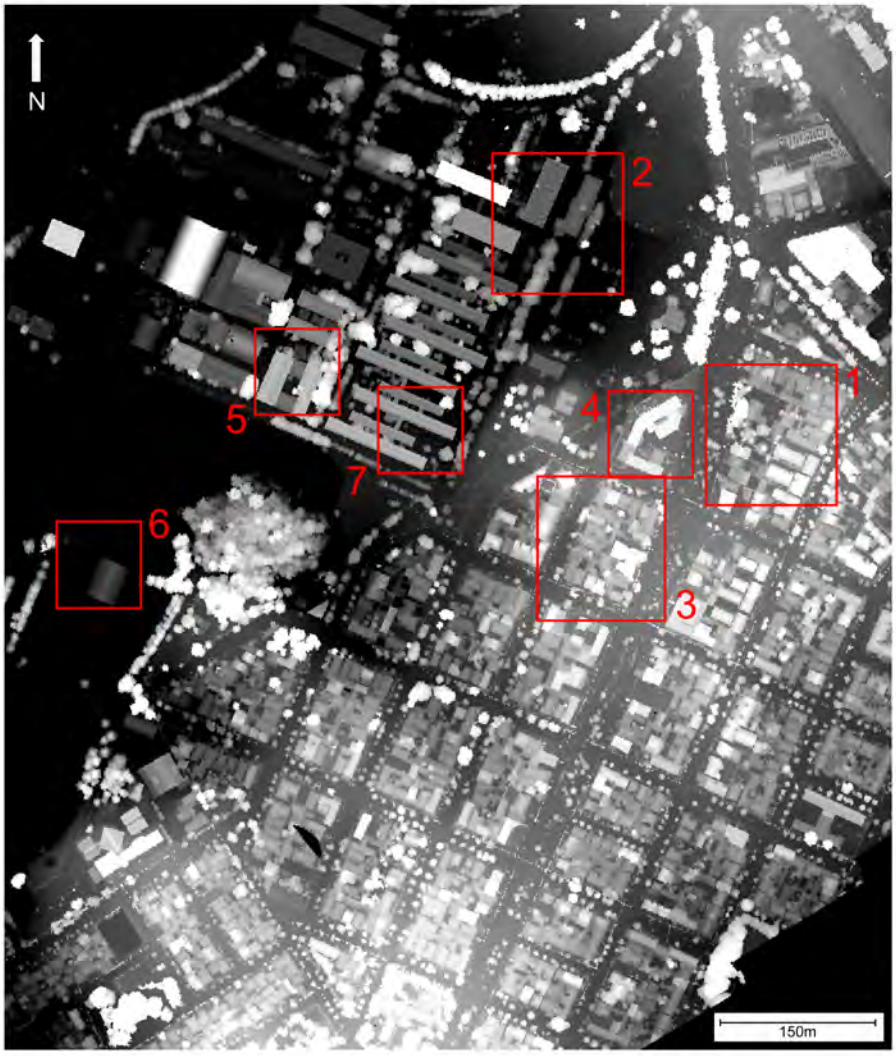


Figure 4.23: UFPR DSM

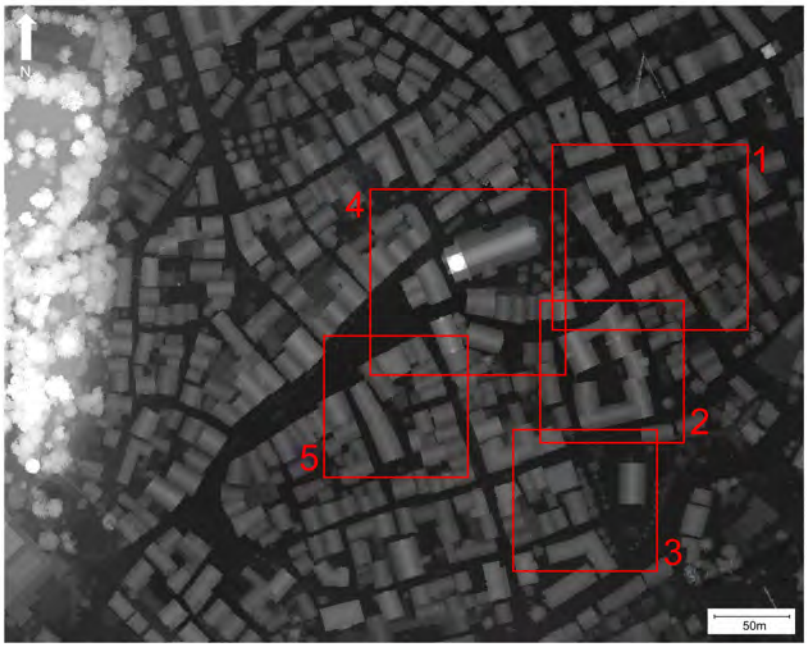
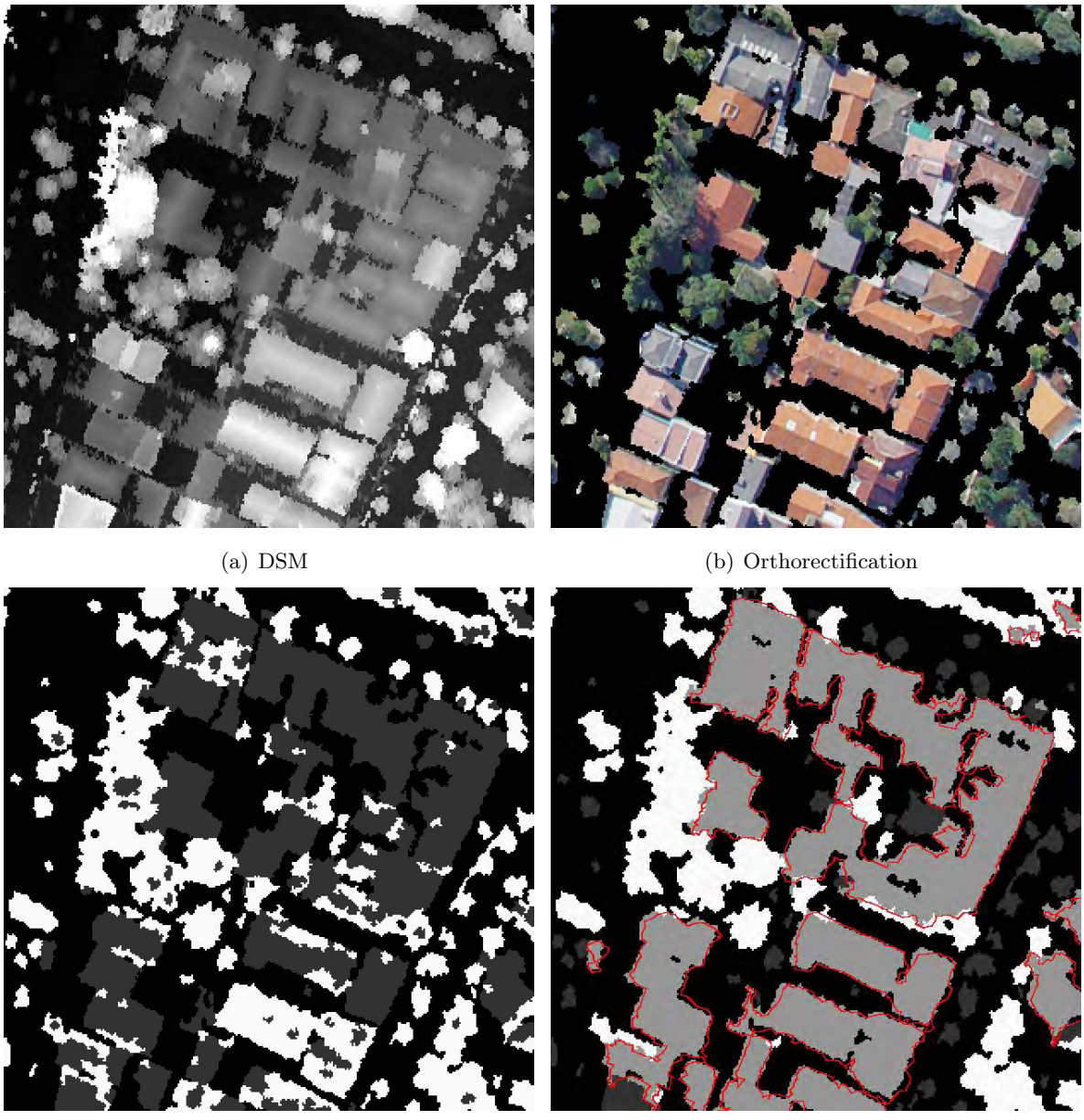


Figure 4.24: Biberach DSM

4.3.1 Overall Visual Analysis

This section presents a visual analysis from the steps ②, ③, ④ and ⑥ shown in figure 4.1.

UFPR Data Set

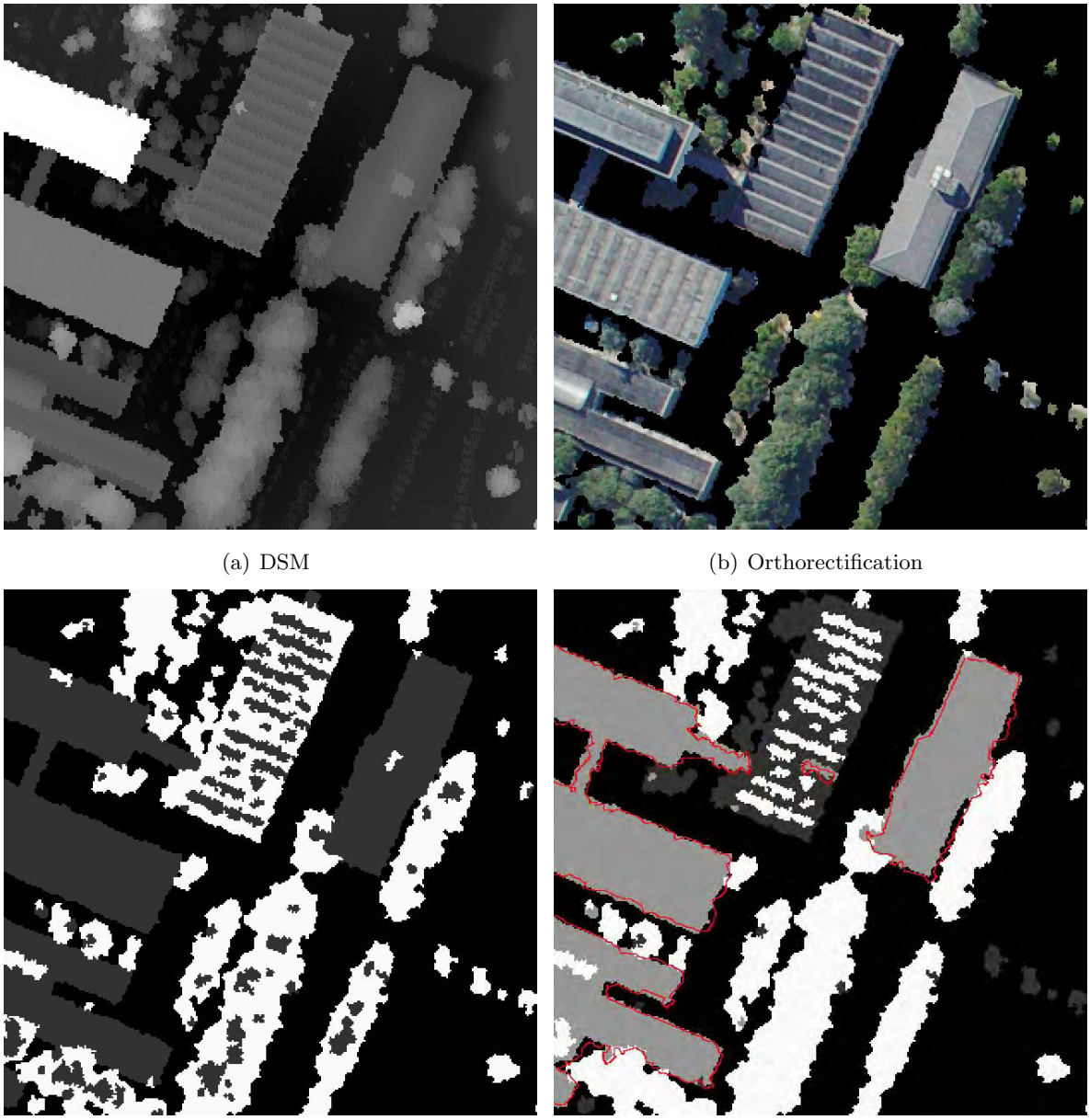


(a) DSM (b) Orthorectification

(c) Split Image (d) Classification and Vectors

Figure 4.25: Area 1 – Object Space – UFPR

Figure (4.25) shows a residential area. Some DSM segmentation problems are observed in fig. 4.25(b). Note also that the DSM shows noisy. Some flat roofs were detected as rough segments as one can see in fig. 4.25(c). They show unclassified in fig. 4.25(d). The overall classification result is good, but the vectorization still shows many problems. Note that misclassification in building edges leads to worse vectorization results.

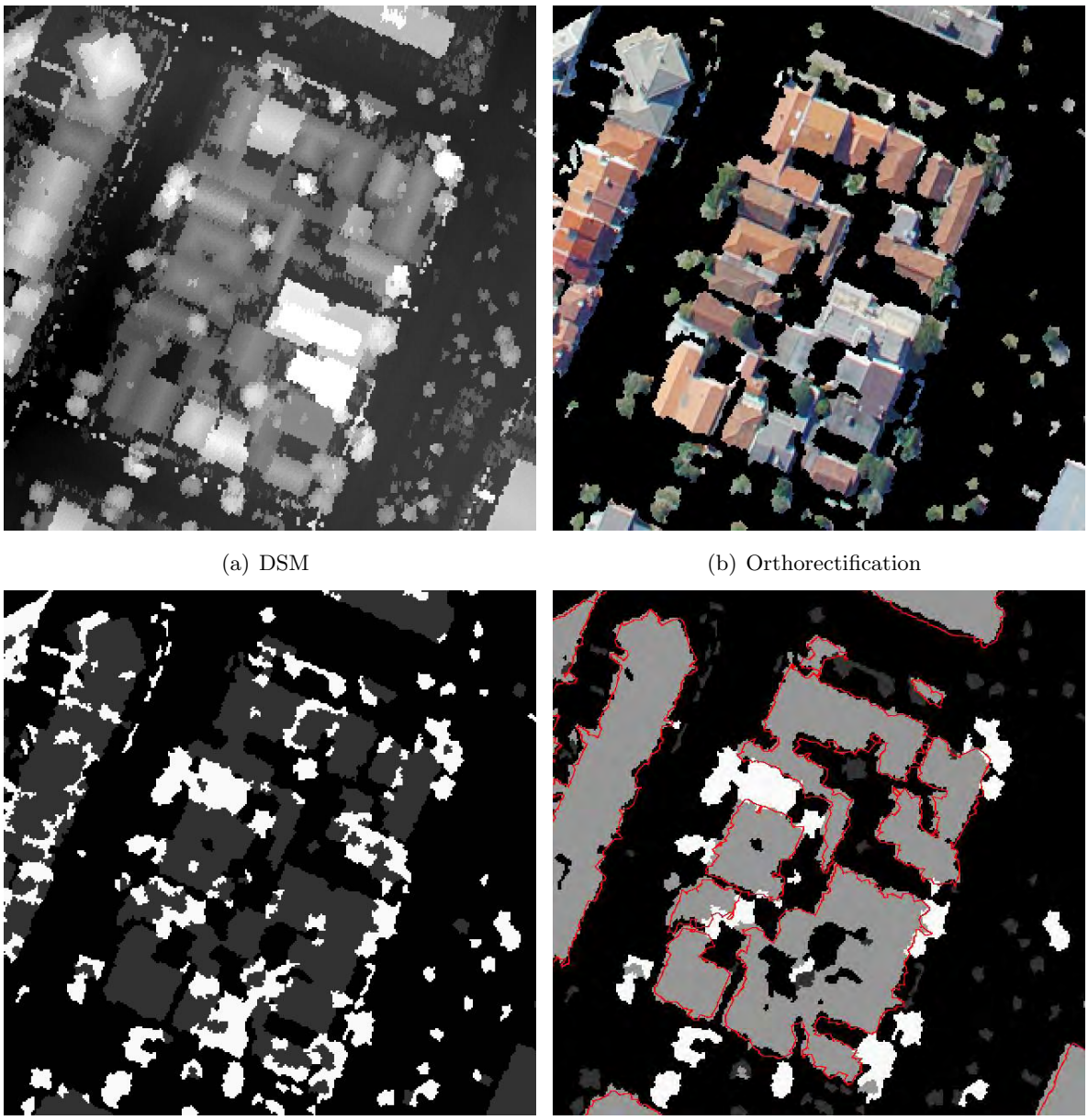


(a) DSM (b) Orthorectification

(c) Split Image (d) Classification and Vectors

Figure 4.26: Area 2 – Object Space – UFPR

Figure (4.26) shows a mixture of buildings and trees. An entire building was misclassified between vegetation and unclassified segments. Note how the smooth isles inside tree tops in fig. 4.26(c) were correctly classified as vegetation. It is possible to note some orientation problems in fig. 4.26(b). This kind of problem leads to bad vectorization results as seen in fig. 4.26(d).

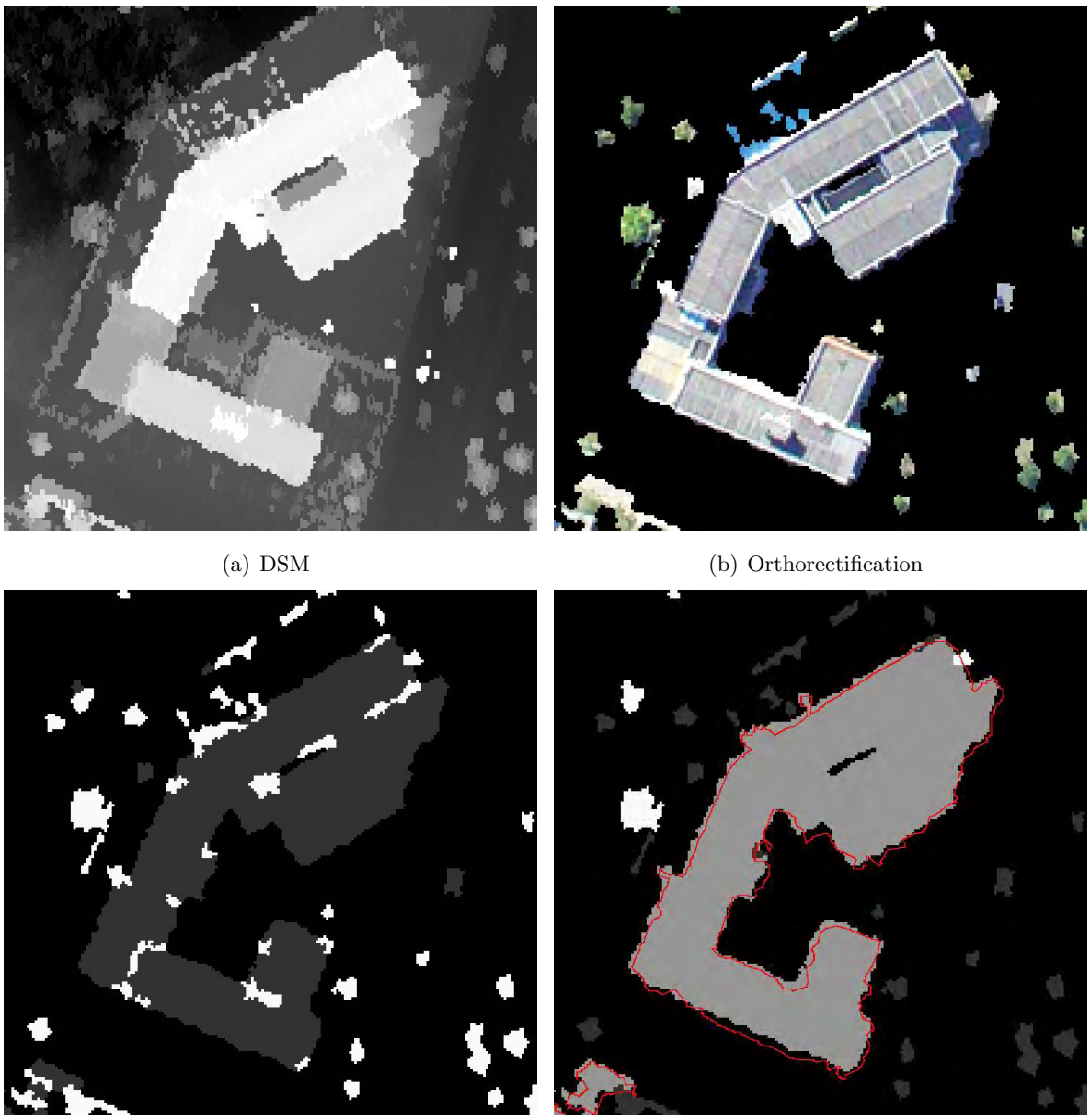


(a) DSM (b) Orthorectification

(c) Split Image (d) Classification and Vectors

Figure 4.27: Area 3 – Object Space – UFPR

Figure (4.27) shows one more residential area. Note how the long building block on the upper left corner was correctly classified, even with many rough segments. Note also the noise in the DSM that looks like a texture in some flat roofs. This kind of effect is quite normal in the UFPR LIDAR data set and leads to bad segmentation results and bad height texture determination.

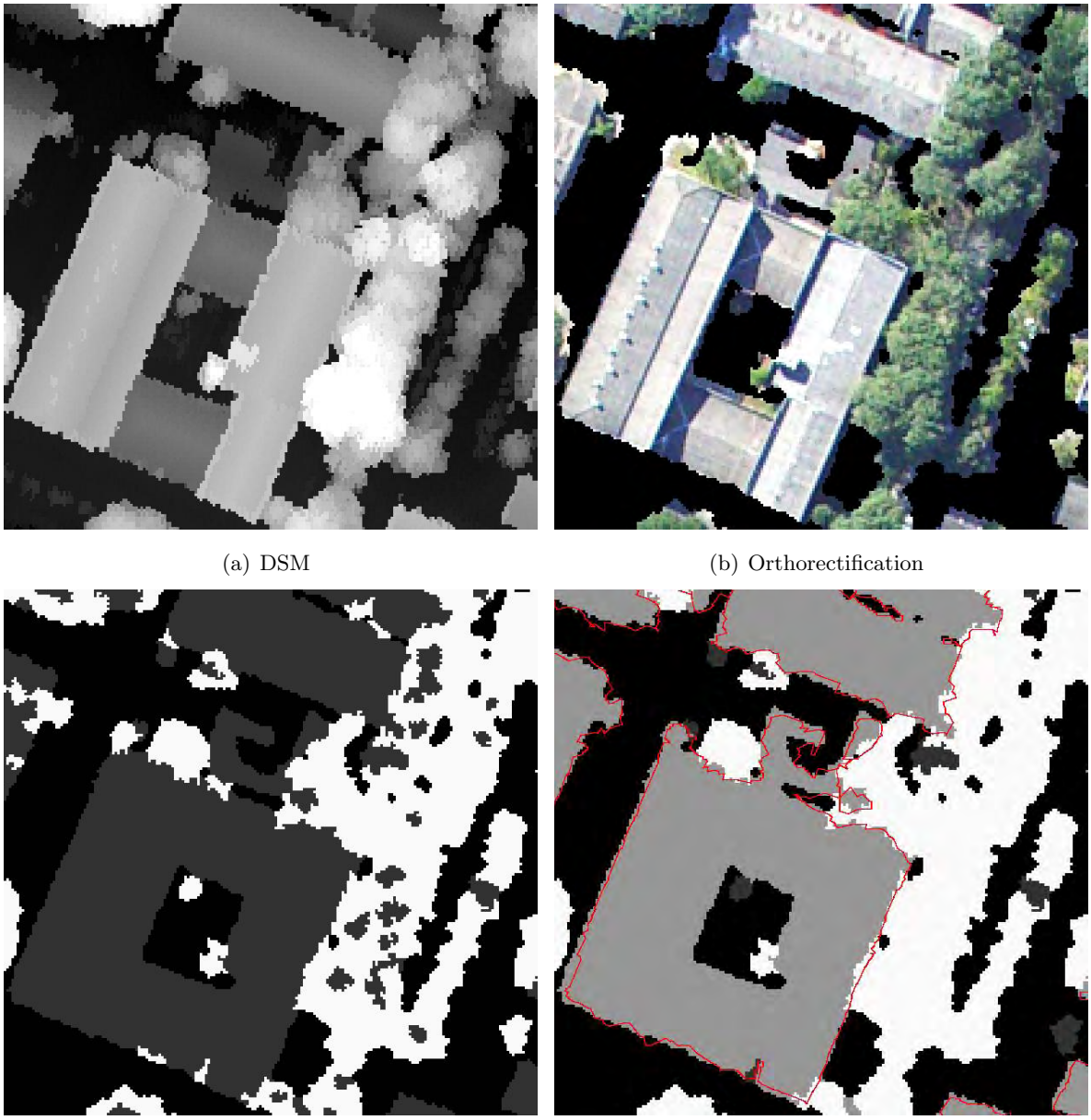


(a) DSM (b) Orthorectification

(c) Split Image (d) Classification and Vectors

Figure 4.28: Area 4 – Object Space – UFPR

Figure (4.28) shows a single complex building. Here, the LIDAR data do not show noisy and the split image shows good results. The internal rough areas are correctly classified as edification. The problems in this example are mostly related to the image orientation. These problems lead to the incorrect vectorization of the down left building edge, for example. The problems with image orientation in the UFPR data set are caused by distortions and in the LIDAR data, which was used as reference.

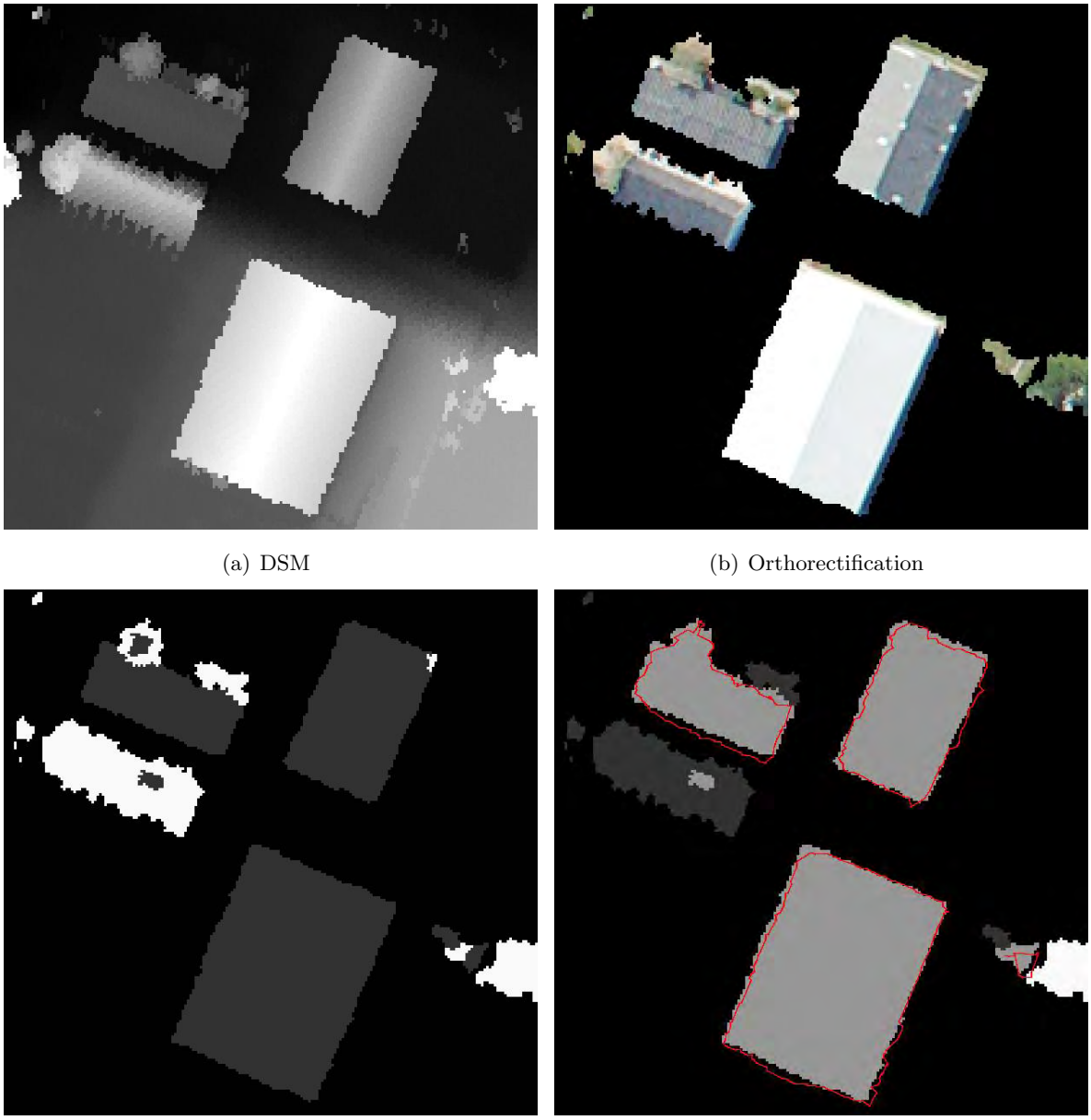


(a) DSM (b) Orthorectification

(c) Split Image (d) Classification and Vectors

Figure 4.29: Area 5 – Object Space – UFPR

Figure (4.29) shows another set of buildings mixed with trees. Here the noise is not remarkable and the image orientation shows good. These two aspects lead to a better image segmentation, as seen in fig. 4.29(b), a good image splitting – fig. 4.29(c) – and a good classification and vectorization as noticed in fig. 4.29(d). Note also that the vectorization shows good results: the left edge is almost perfectly identified as a straight line.

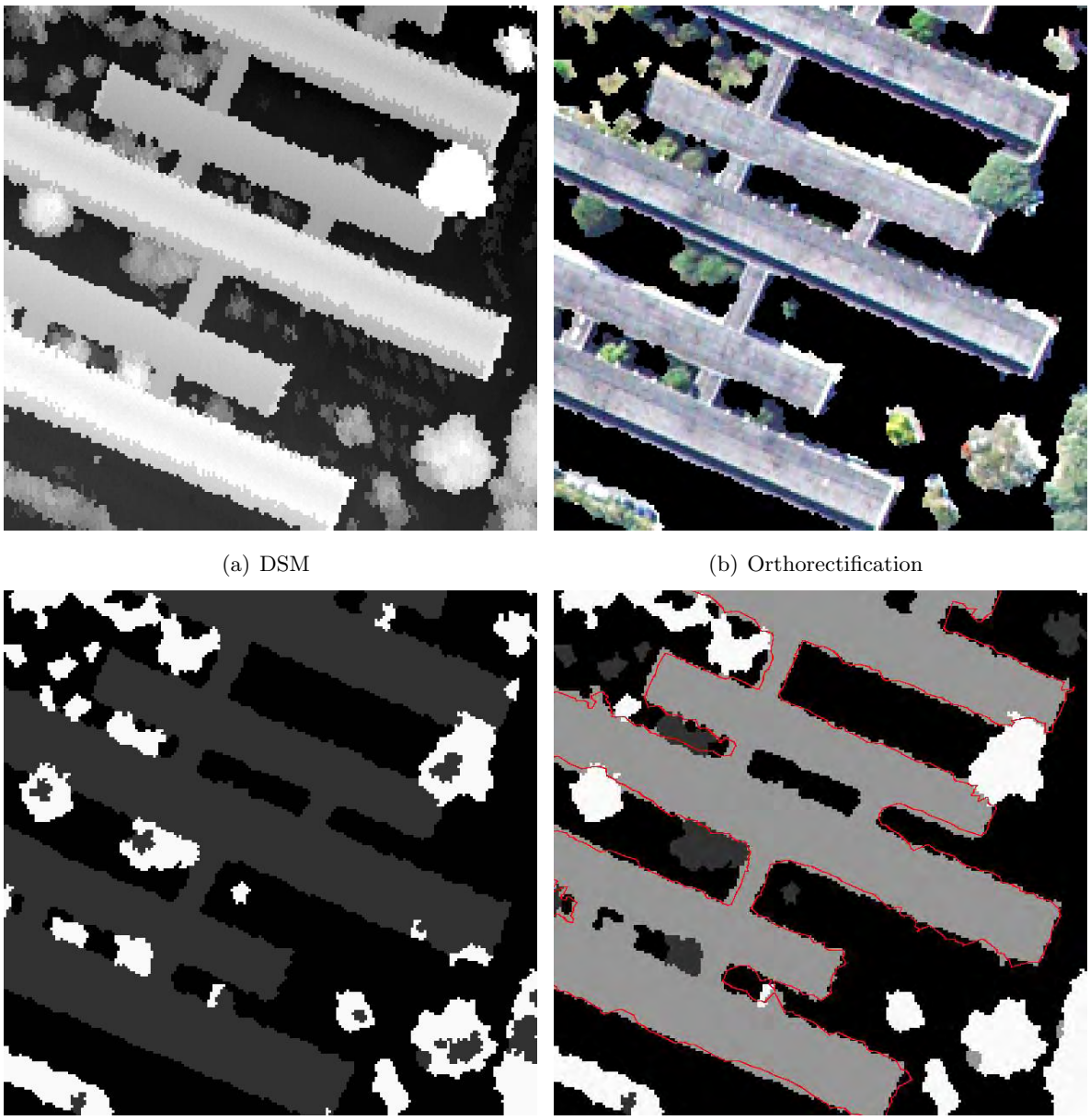


(a) DSM (b) Orthorectification

(c) Split Image (d) Classification and Vectors

Figure 4.30: Area 6 – Object Space – UFPR

Figure (4.30) shows 4 buildings. Three of them were well detected and classified. The vectorization shows some problems due to image orientation. Note the small green strip in the upper side of the white roof, this proves an orientation problem. The fourth building shows a very complex shape and is not classified.



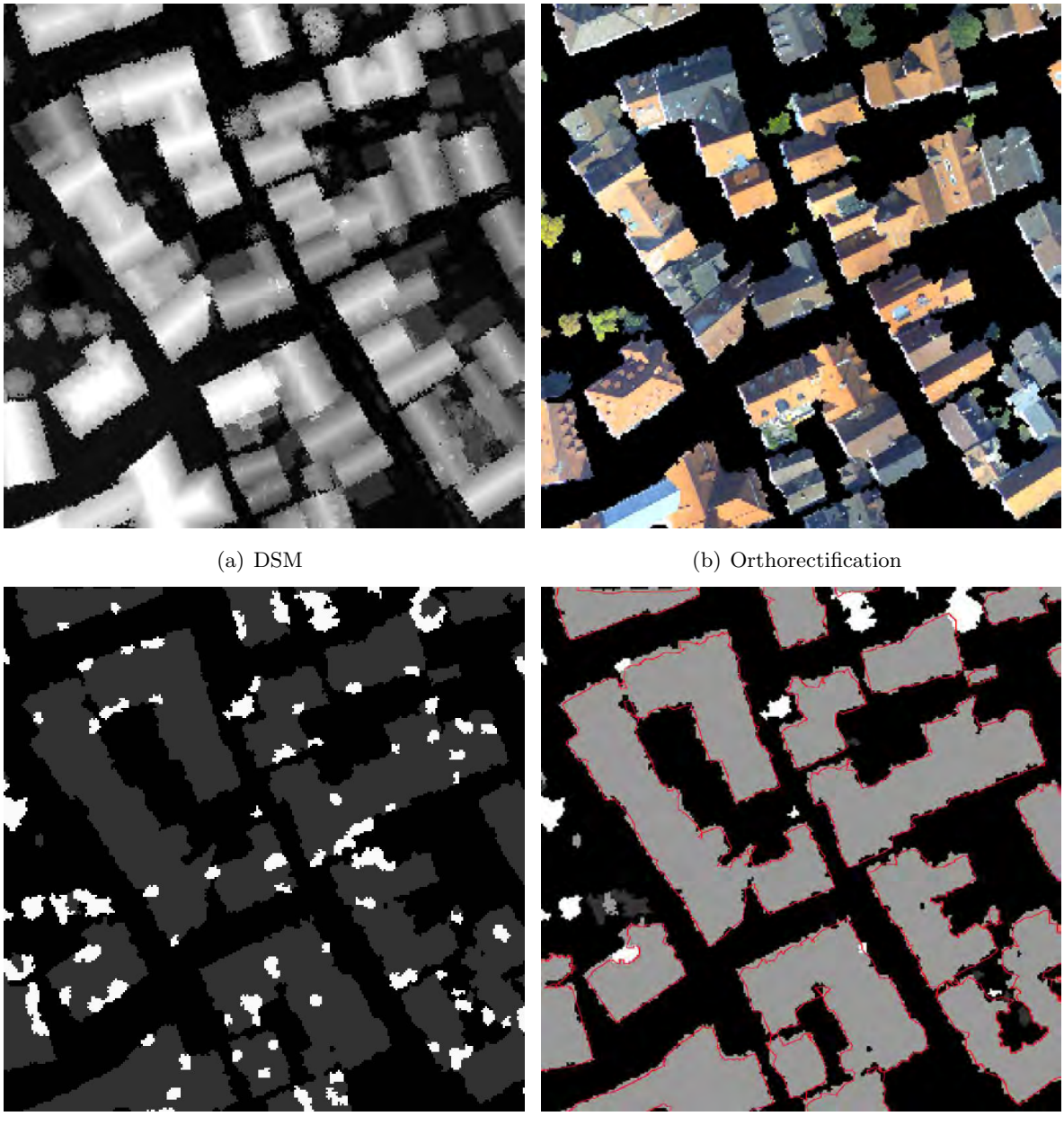
(a) DSM (b) Orthorectification

(c) Split Image (d) Classification and Vectors

Figure 4.31: Area 7 – Object Space – UFPR

Figure (4.31) shows a set of connected buildings. The results are good: the DSM is correctly segmented, split and classified. Some orientation problems can be noticed. The vectorization shows also good results. The internal areas are not treated in this work, just the external boundaries.

Biberach Data Set

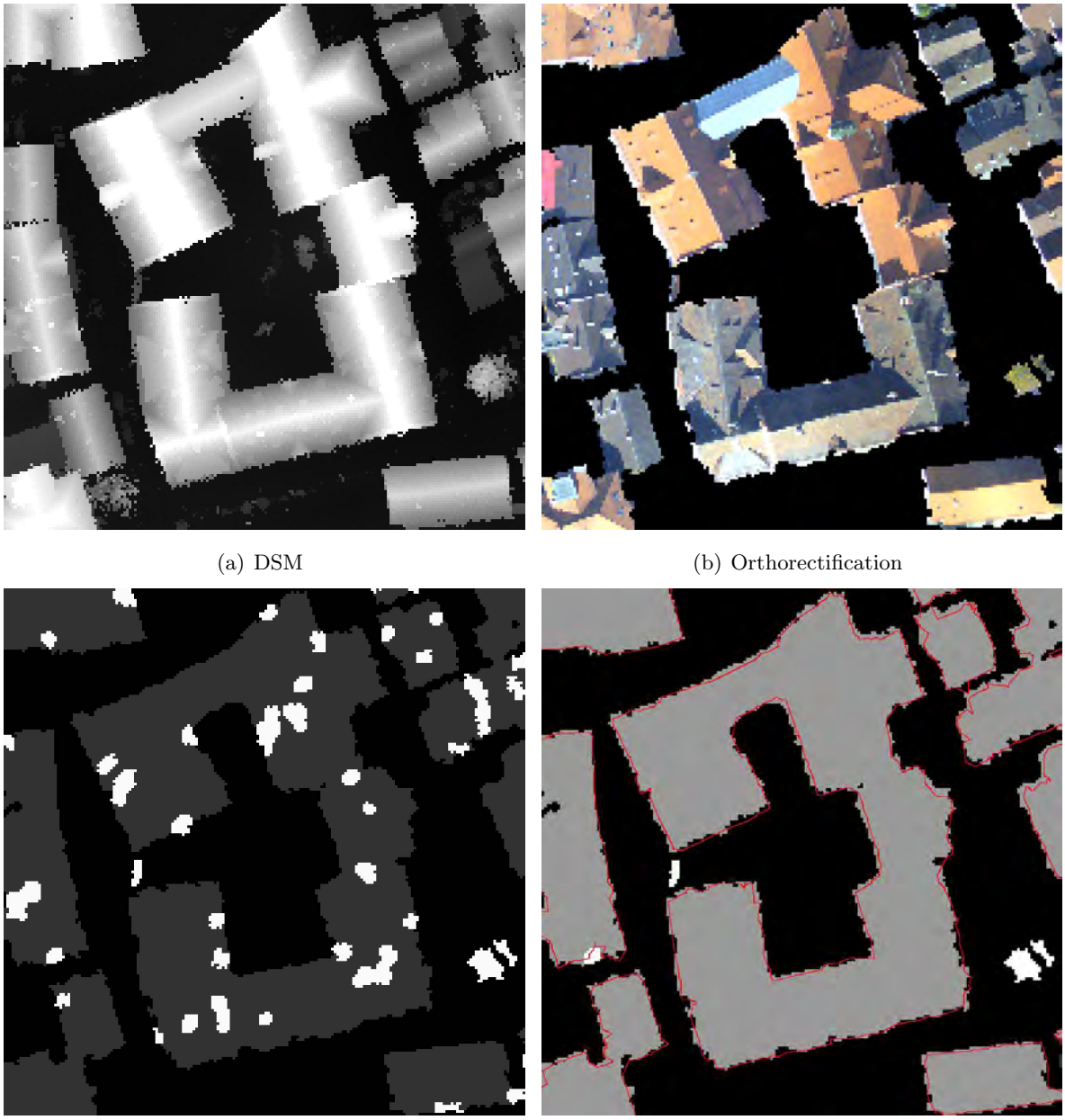


(a) DSM (b) Orthorectification

(c) Split Image (d) Classification and Vectors

Figure 4.32: Area 1 – Object Space – Biberach

Figure (4.32) shows some building blocks. There is almost no noise in the DSM. The orientation of the Biberach image is very good, since the orientation parameters were measured at the same time as the LIDAR data. Note that almost all rough isles inside building blocks were correctly classified as edification. Note also that the misclassified segments tend to be on the building edges, which leads to vectorization errors.

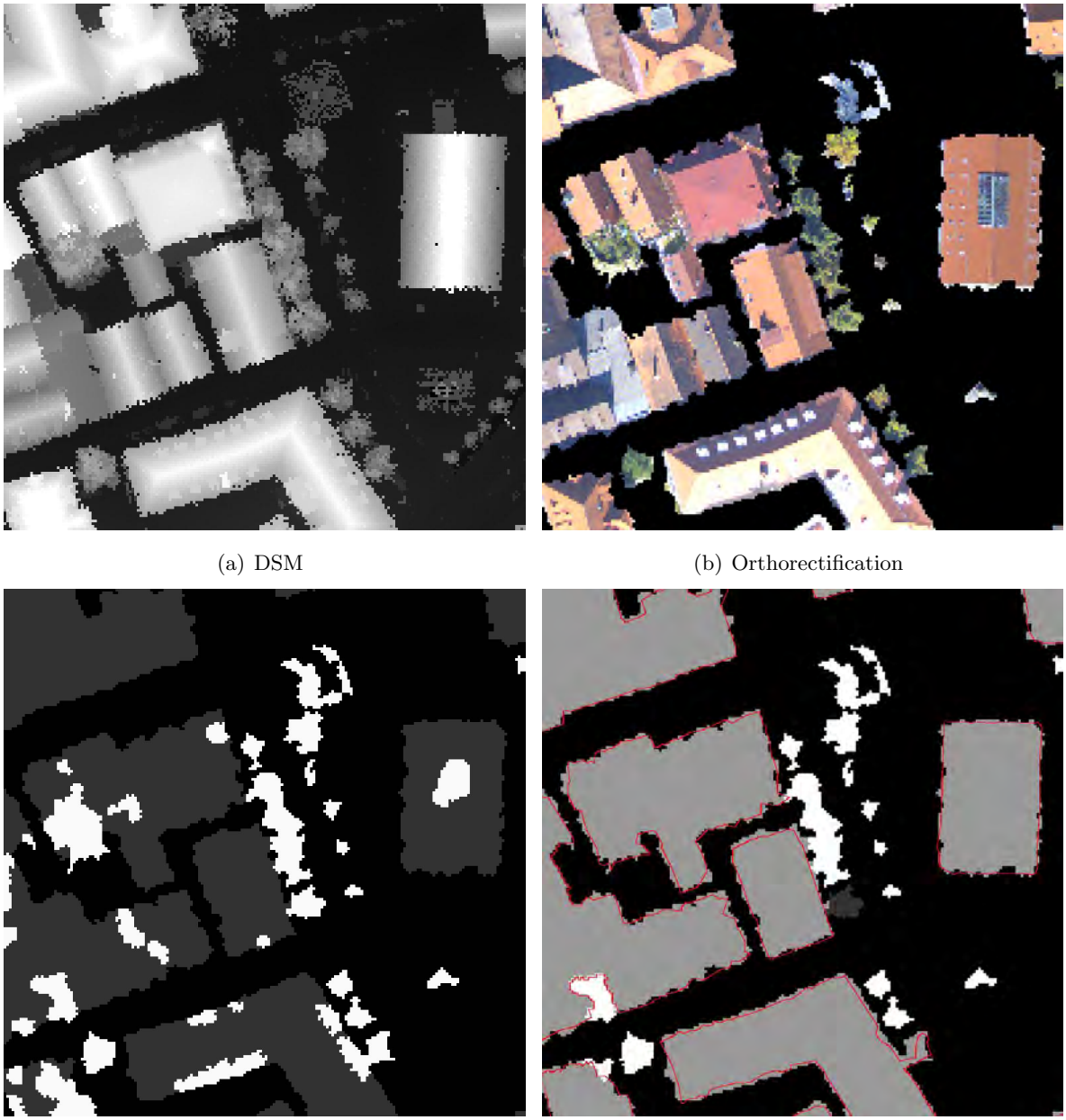


(a) DSM (b) Orthorectification

(c) Split Image (d) Classification and Vectors

Figure 4.33: Area 2 – Object Space – Biberach

Figure (4.33) shows a complex building. All rough isles were correctly classified as edification. Note that the vectorization shows a very realistic result, with a better edge definition than in the segmentation result. This example proves the potential of the methodology, since a complex building block was correctly detected and vectorized.

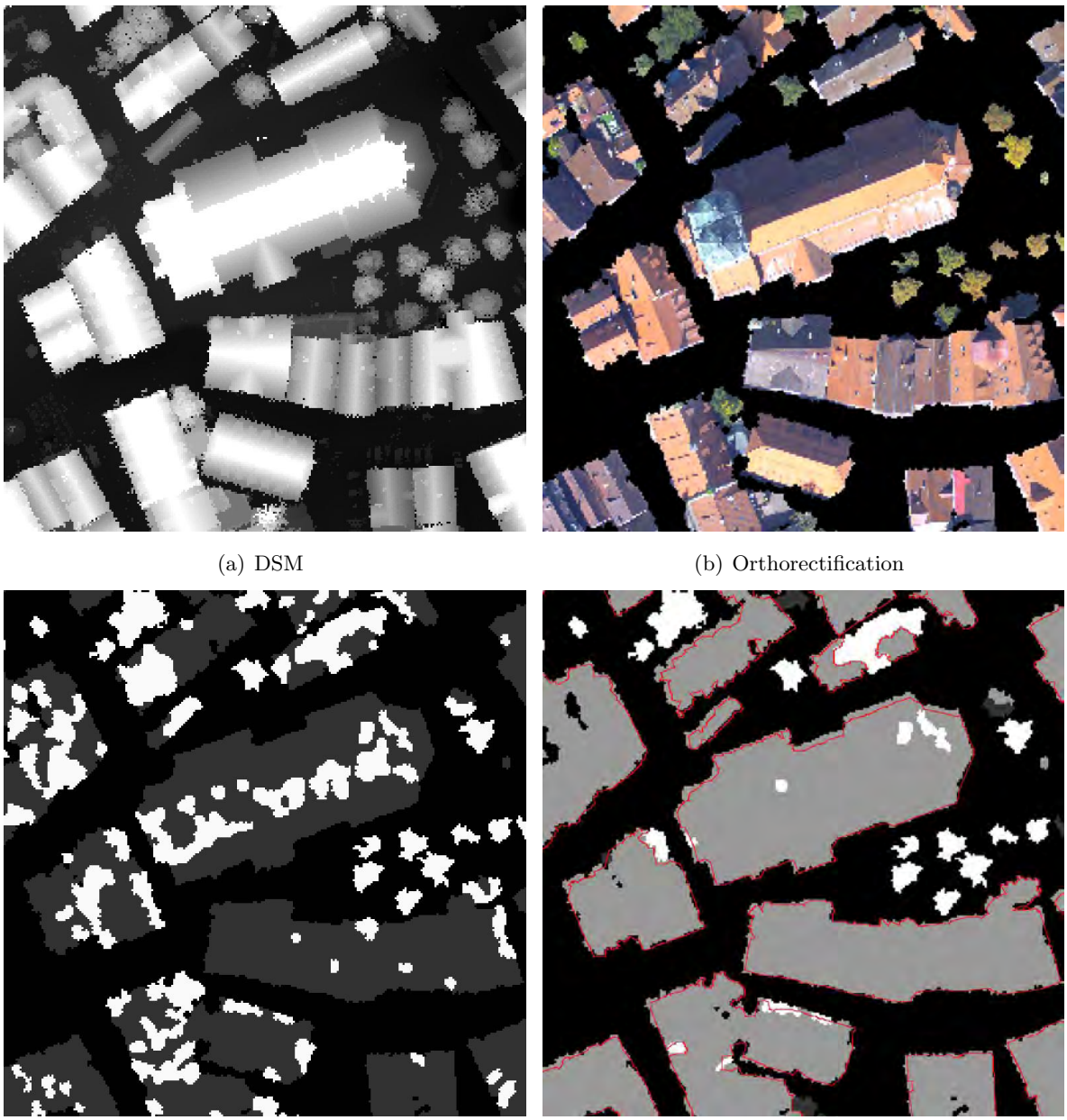


(a) DSM (b) Orthorectification

(c) Split Image (d) Classification and Vectors

Figure 4.34: Area 3 – Object Space – Biberach

Figure (4.34) shows one more set of buildings. Note the building on the right side, and how the vectorization shows a very good result. Note also that most trees were correctly classified. A small error occurs in the large building in the lower left corner. In the left side one misclassification occurs, which leads to a wrong vectorization.

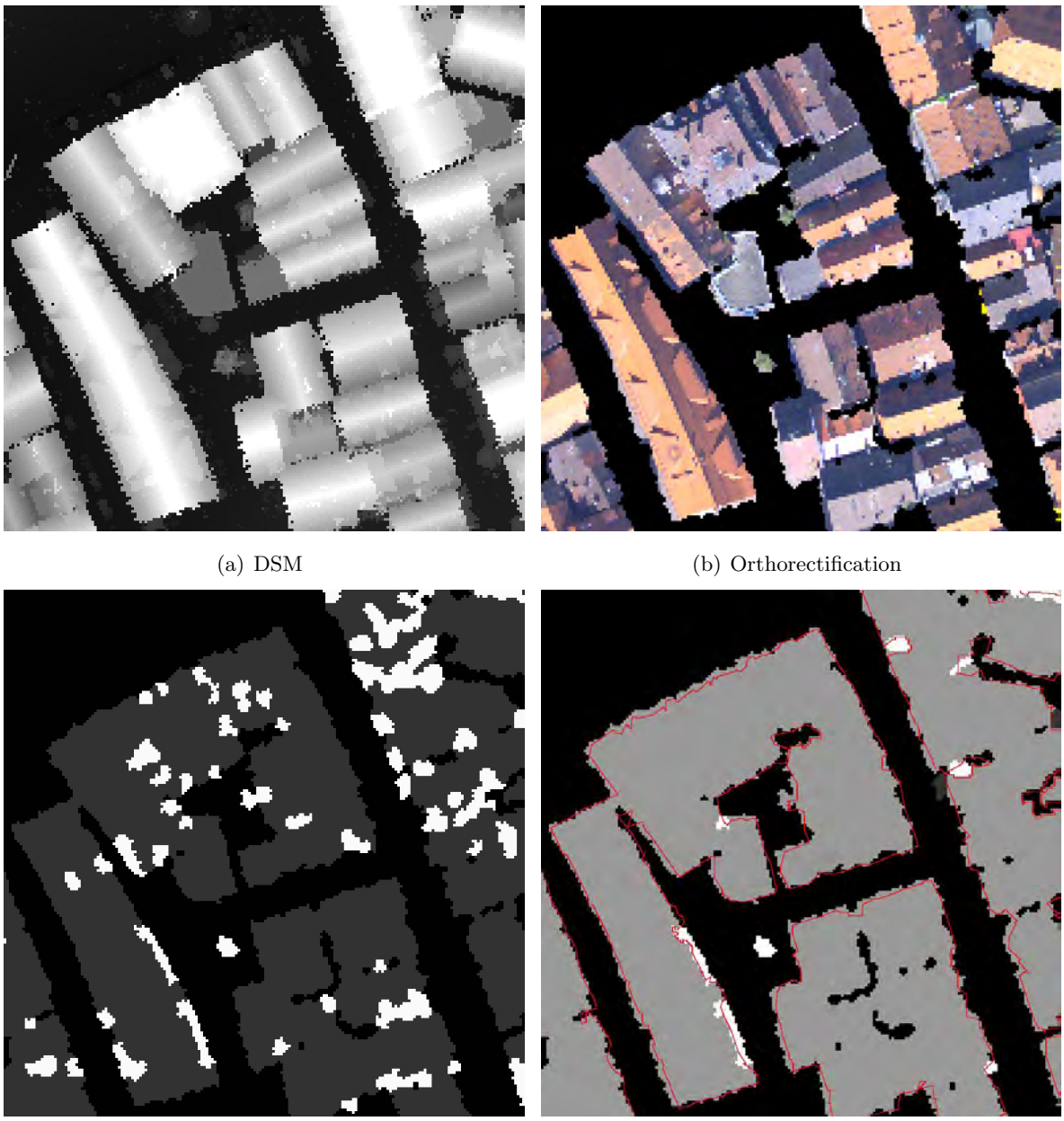


(a) DSM (b) Orthorectification

(c) Split Image (d) Classification and Vectors

Figure 4.35: Area 4 – Object Space – Biberach

Figure (4.35) shows some problematic cases. Note again how the misclassified segments in the edges lead to worse vectorization results. Note also that here the illumination shows bad conditions, the methodology has higher probability of failure. Nevertheless, most trees were correctly classified. Once again, the proposed methodology shows good results for complex areas, since enough quality data is available.



(a) DSM (b) Orthorectification

(c) Split Image (d) Classification and Vectors

Figure 4.36: Area 5 – Object Space – Biberach

Figure (4.36) shows a last example from the Biberach data set. A set of dark and bright roofs is shown. It becomes quite clear that under favorable illumination conditions, see the long building in the left side, the methodology shows good results. Note in this building how in the brighter side (left) shows good classification and vectorization, and the darker side (right) shows worse results.

4.3.2 Classification and Building Vectorization Analysis

In a similar way to the image space, it is necessary to make a statistical analysis of the obtained results in the object space. From the UFPR data set 39 buildings (figures 4.38, page 118 and 4.39, page 119) and 54 vegetation areas (figure 4.40, page 120) were tested. From Biberach, 42 buildings (figure 4.41, page 121) were tested. These areas were manually digitized over the DSM with the help of the aerial imagery to be used as reference vectors. All pixels from the fuzzy classification that lie inside the digitized edification and vegetation areas are tested. The ideal result would be 100% for the given class (vegetation or edification). Table (4.3.2) presents the obtained results:

Tested Data	Edification (%)	Vegetation (%)	Ground (%)	Not Classified (%)
UFPR – Edification	89.19	2.09	6.55	2.17
UFPR – Vegetation	2.73	87.40	7.69	2.18
Biberach – Edification	90.85	0.64	8.32	0.19

Table 4.1: Pixels from Fuzzy Classification Statistics

The results show that the classification correctness is similar for both data sets. The main difference is in the distribution of the misclassified pixels. In the Biberach data set, almost all errors (8.32%) are from pixels classified as ground. In the UFPR data set, more pixels are misclassified as vegetation or not classified. This effect happens because there is, indeed, more vegetation in the data set. These misclassifications lead to a worse edge detection, since the misclassified pixels tend to be on the building borders, as seen in section (4.3.1). The pixels misclassified as ground, in both data sets, are errors from the DSM preliminary segmentation, as shown in section (2.5).

In order to prove the estimation that the Biberach data set shows better results than the UFPR data set, a second test is proposed. All detected building vertexes are tested against the digitized polygons, as done in section (4.2.2). The manually digitized vectors are split in 10cm resolution, and these points are used as reference. Tables (4.2, page 117) and (4.3, page 117) show the percentage of correctly detected vertexes for three precision levels (search radius of 0.5m, 1m and 1.5m). The tested buildings and the reference vectors are shown in figures (4.38), (4.39) and (4.41). Note that the UFPR data set shows worse vertex quality than the Biberach data set. It was already expected given the visual and numeric analysis results. Figure (4.37) summarizes the tables, showing the mean results for the three precision levels (search radius):

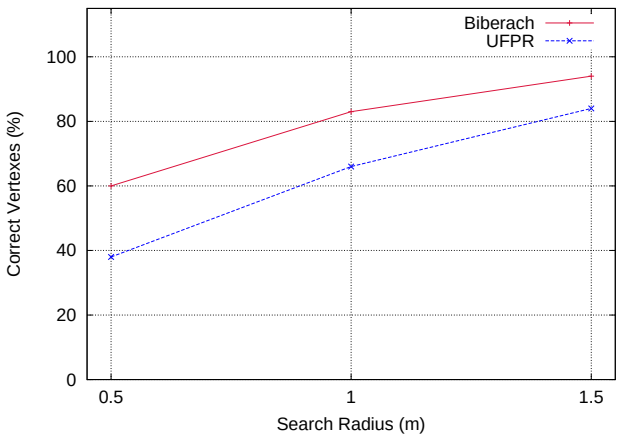


Figure 4.37: Building Detection Overall Results

Figure (4.37) shows that the classification quality is a main issue for the correctness of the building footprint detection. Note a systematic difference of about 20% in the vertex quality caused by misclassified segments on the edification borders (about 4% not taking in account the pixels misclassified as ground, since these errors are not caused by the classification itself). Also the 6% difference in the vectorization (figure 4.22) play an important role on the final results. As one can see in figure 4.1, the processing steps are interdependent and lead to error accumulation. If one analyzes the steps separately, the UFPR data set shows always slightly worse results than the Biberach data set. At the end, these small differences sum up and a considerably difference in the vertex quality is noticed. The most important conclusion is that data quality plays the major role in the final results. This conclusion and other remarks will be presented and discussed in the conclusions chapter.

Photo 195

ID	$0.5m$	$1.0m$	$1.5m$
1	40%	68%	87%
14	38%	68%	87%
10	43%	72%	98%
13	29%	64%	92%
8	45%	70%	85%
7	50%	70%	90%
5	38%	66%	89%
18	36%	70%	92%
15	52%	80%	99%
19	32%	54%	81%
9	33%	60%	78%
22	38%	77%	94%
26	49%	74%	100%
27	32%	61%	85%
46	43%	75%	97%
47	26%	48%	73%
52	35%	55%	70%
40	26%	48%	70%
42	44%	70%	80%
50	51%	79%	95%
43	38%	66%	84%
34	31%	51%	68%

Photo 197

ID	$0.5m$	$1.0m$	$1.5m$
23	34%	71%	95%
18	30%	52%	75%
9	56%	75%	91%
45	32%	63%	93%
49	31%	65%	92%
57	19%	40%	66%
55	42%	77%	93%
52	43%	75%	92%
51	48%	81%	100%
58	34%	64%	86%
54	39%	71%	93%
39	32%	49%	62%
47	30%	59%	85%
53	51%	75%	92%
22	31%	51%	70%
24	28%	51%	80%
15	57%	93%	100%

Table 4.2: UFPR Building Extraction Statistics

ID	$0.5m$	$1.0m$	$1.5m$	ID	$0.5m$	$1.0m$	$1.5m$
122	41%	64%	88%	**154**	85%	99%	100%
138	57%	77%	98%	**144**	37%	59%	73%
141	60%	83%	88%	**148**	56%	78%	88%
136	48%	65%	75%	**133**	53%	73%	84%
180	67%	90%	87%	**131**	58%	87%	98%
156	58%	76%	85%	**136**	57%	77%	90%
177	65%	91%	99%	**128**	53%	81%	94%
161	48%	70%	87%	**111**	71%	90%	99%
160	60%	84%	100%	**121**	75%	93%	100%
157	52%	84%	96%	**123**	77%	98%	100%
155	42%	76%	97%	**125**	74%	94%	100%
137	61%	82%	91%	**126**	75%	93%	98%
108	67%	88%	94%	**105**	38%	54%	66%
175	60%	85%	99%	**116**	49%	72%	93%
127	73%	92%	98%	**117**	45%	64%	75%
129	43%	66%	79%	**120**	63%	78%	80%
170	46%	75%	91%	**113**	57%	81%	96%
112	45%	70%	83%	**103**	49%	76%	94%
169	55%	89%	100%	**104**	66%	89%	99%
172	71%	91%	100%	**109**	49%	84%	100%
157	93%	100%	100%	**114**	70%	83%	92%

Table 4.3: Biberach Building Extraction Statistics

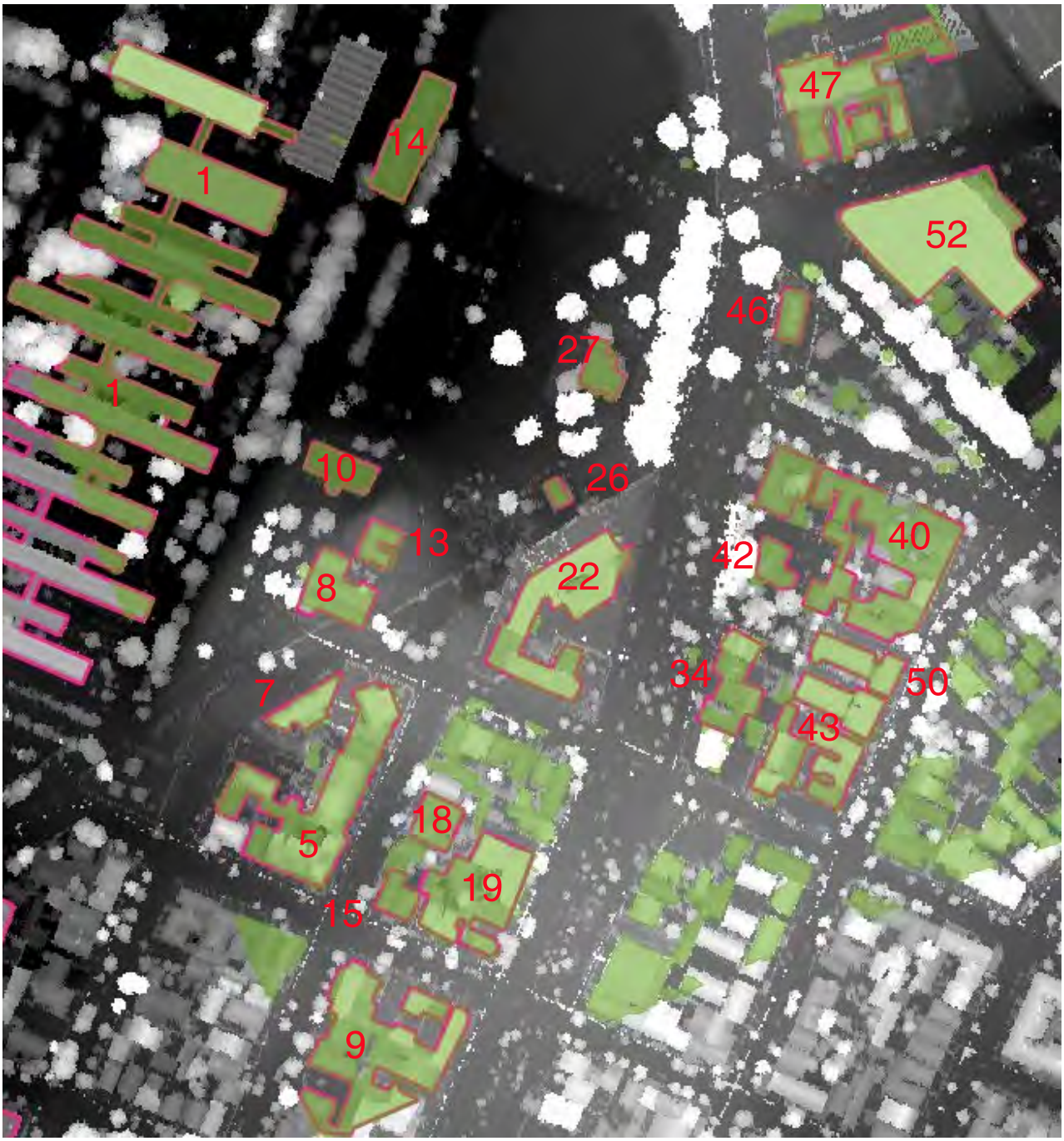


Figure 4.38: UFPR Analyzed Buildings – Photo 195

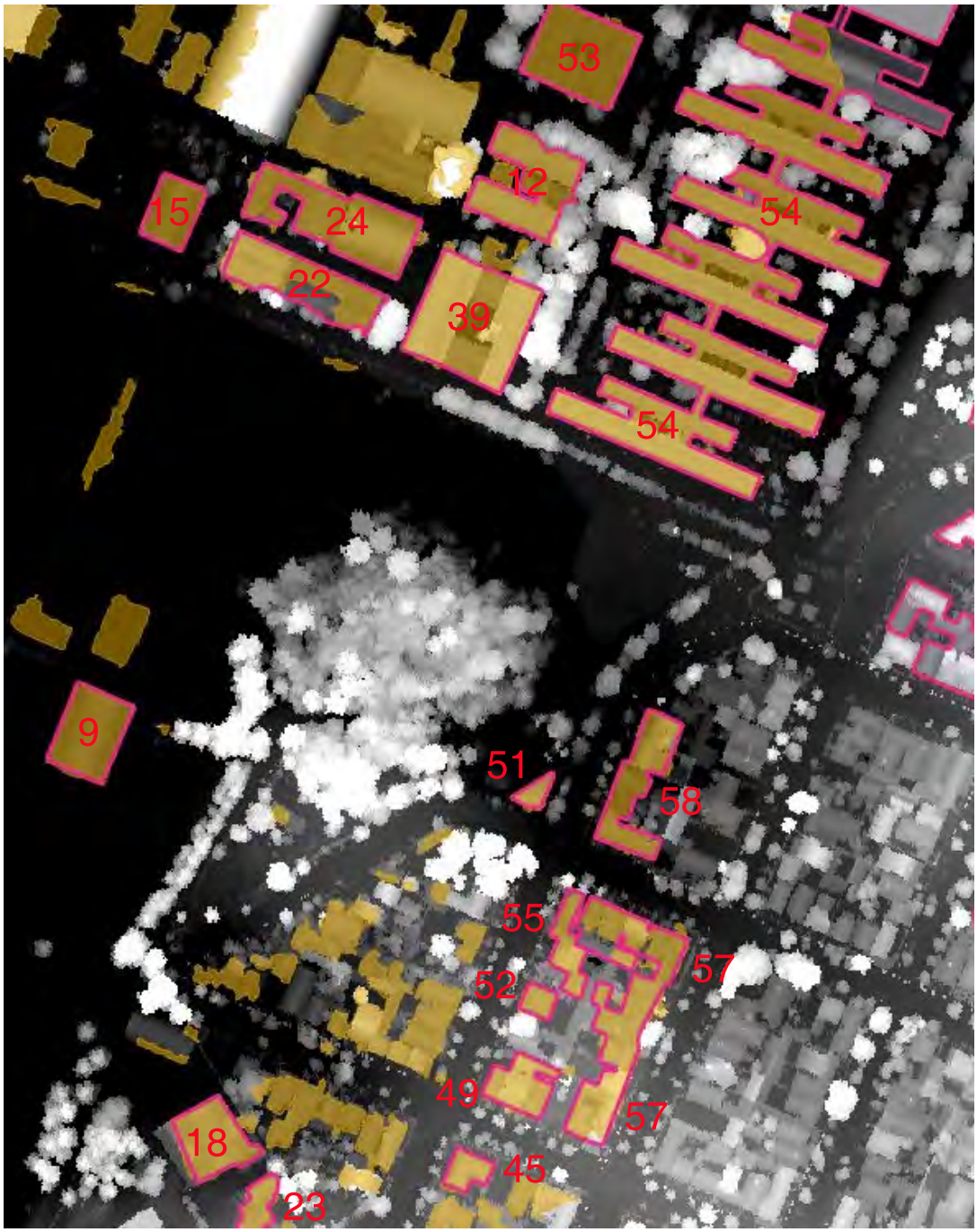


Figure 4.39: UFPR Analyzed Buildings – Photo 197

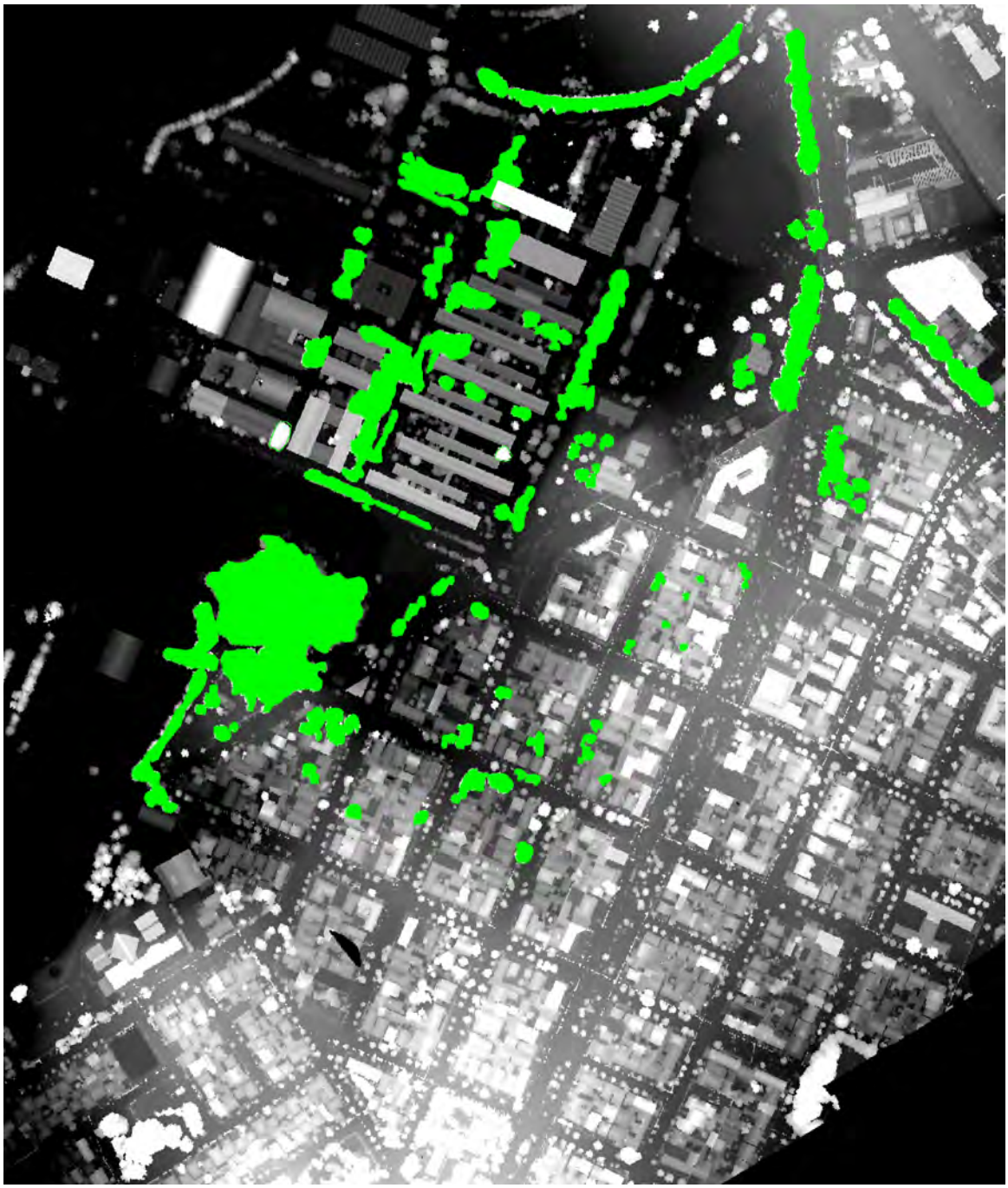


Figure 4.40: UFPR Analyzed Vegetation

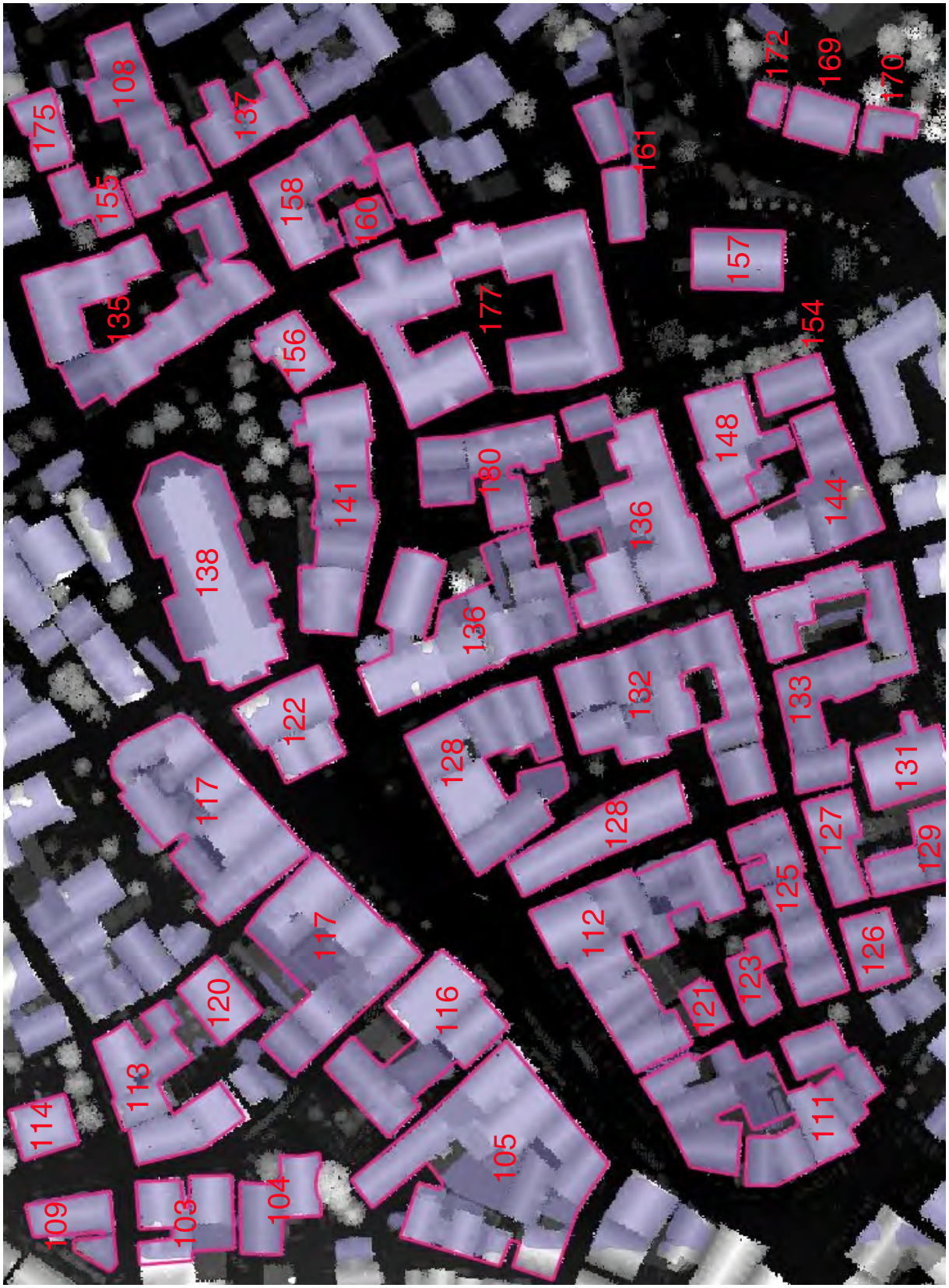


Figure 4.41: Biberach Analyzed Buildings

Conclusions and Future Work

The main objective of this work, to implement a semi-automated monoplotting system for the integration of LIDAR data with aerial imagery, was achieved. Also the specific objectives pointed in the introduction were fulfilled.

The proposed image orientation procedures showed good results. The created distortion-free image showed very helpful among the development. An image segmentation methodology called *seeded region growing mean-shift segmentation* was proposed, and its results showed similar to commercial segmentation packages. The technique showed "easy to tune" – it is simple to set parameters in order to achieve desired results. The technique showed also very flexible during the development, and it can be easily enhanced. Some future work are the use of a near-infrared band during the segmentation process and perform tests directly in RGB color space.

There are some drawbacks in the proposed segmentation methodology. Most remarkable is the difficulty to obtain good results in areas with shadows. This problem leads to wrong roof detection. A solution would be to take images with 11bit radiometric resolution, since this option is already available in most mid-format cameras installed with LIDAR systems. Also a fourth band (NIR) could help in the segmentation in areas with shadows. The analyzed images showed also some pixel mixture problems in building borders. This effect caused failures during the segmentation processes. Another drawback in the proposed segmentation methodology was the computational performance. It showed acceptable for the UFPR images (5.2Mpix), but for the Biberach image (22Mpix) the processing time was already long. Tests made with a 81Mpix large frame scanned air photo showed unviable. Since image resolution tends to grow, and multi core computers become widely used, multi core parallel programming techniques could reduce segmentation processing time.

Most image processing techniques used in geoscience were developed for Remote Sensing applications. This is easy to understand, since Remote Sensing is a digital technology since its very beginning. Geoscience is living a "high-resolution revolution", and image processing is adapting itself to this new reality. In Remote Sensing, high-resolution means about 50cm GSD, but in modern digital Photogrammetry high-resolution means about 5cm GSD. Nevertheless, the radiometric quality of digital airborne sensors is rapidly raising. Due to lower atmospheric noise, digital photogrammetric images show, nowadays, a much better radiometric quality as orbital imagery. These resolution gaps must be taken in account when developing automatic photogrammetric procedures, such as image segmentation. The proposed question is: *"Are the momentary image processing tools sufficient for the raising resolution and quality of digital photogrammetric imagery?"*

This work proposed a GIS based storage of the LIDAR data in the PostGres/PostGIS database environment. The intention was to achieve efficient storage and access to LIDAR data. The performance showed very good, in the order of milliseconds to find a point between about 14 millions. This integra-

tion of LIDAR data into a GIS environment showed also very helpful during data fusion, when LIDAR data must be accessed multiple times to be transformed into image space coordinates. This storage approach showed also good performance during DSM creation. The modified Araki algorithm proposed for DSM creation fulfills its conceptual proposals, that is to be a simple and efficient building-edge conserving algorithm. The height texture based segmentation showed also good results. This initial classification showed fundamental to the success of the proposed fuzzy inference system during the fusion stage. The plane parametrization obtained during the height texture processing shows useful for future work in other applications, as change detection, 3D city modeling and roof detection for photovoltaic plates.

The main LIDAR data processing problem occurred during the filtering process. It was not an objective of this work to make developments in this area, and the solution implemented within the GRASS GIS was adopted. The results presented in this work were manually refined in order to obtain an acceptable bare ground description. Nevertheless, some residual errors led to failure in the DTM generation, which led to errors in the DSM segmentation, and so on. Another drawback occurred during the UFPR block image orientation process. The LIDAR strips are not perfectly adjusted, and the image orientation depends on coordinates from multiple LIDAR strips. Even though aerotriangulation with additional parameters showed good results, some distortions have been noticed during the fusion stage. This stage is strongly dependent on the exterior orientation parameters, since it uses the normal and the inverse collinearity equations. Simultaneous flight of LIDAR and imagery (as in the Biberach data set), profiting of the IMU and GNSS systems from the LIDAR sensor, shows best results. It is possible to implement most of the proposed procedures directly in the point clouds, in order to avoid distortions that can occur during DSM creation. The biggest challenge is polygon delineation directly from point clouds, which can be developed in future work.

This work proposed a data fusion methodology in two steps: a *spectral fusion* through orthorectification and fuzzy classification, and a *geometric fusion* for building footprint extraction. The proposed methodology shows good results if data conditions are favorable: 90% in the edification classification and 82% of the polygon vertices with about 1m precision. Also complex sets of roofs are correctly detected and vectorized. For an individual building description, additional data shows necessary, as the parcel divisions, for example. The fuzzy inference system showed high flexibility during development, and a more complete parametrization (i.e. the use of the shape as classification parameter) can be easily implemented in future work. The green index proposed to help in vegetation detection showed interesting results, even if many “green pixels” were wrongly classified. Since most low-cost cameras do not have a NIR sensor, the green index use shows relevant in vegetation detection.

The developed data fusion methodology shows very dependent on data quality, both imagery and LIDAR. All processing stages are strongly correlated, and even small errors in individual steps can lead to relevant errors in the final result. For example, the roof detection from the aerial imagery is strongly dependent on the image segmentation, which is strongly dependent on the image quality. Here, image quality is understood as a combination of geometric resolution and contrast between neighboring areas. The DSM segmentation is strongly dependent on the DTM quality, which strongly depends on LIDAR filtering results. These error accumulations can be reduced by data processing improvements to be implemented in future work.

A bottom line of this work is about the relation between data quality and resolution. During the development of this work, it became clear that good automation results are highly dependent on good

data quality. It is relevant, then, to discuss *photogrammetric data quality*. Accordingly to the DIN EN ISO 9000 (2005) *"quality is the degree to which a set of inherent characteristics fulfils requirement"*. In Remote Sensing, data quality is usually related to data resolution: geometric, radiometric, spectral and temporal. With the advent of digital Photogrammetry, these formerly separated geosciences – Photogrammetry and Remote Sensing – are nowadays more likely to pertain to the same "fuzzy set". It is possible to borrow concepts of Remote Sensing data quality to digital Photogrammetry. Nevertheless, some adaptation must be done. The most relevant points to be discussed are about geometric and temporal resolution. In Remote Sensing geometric resolution is usually related to the GSD. For photogrammetric sensors, the geometric resolution must take into account the camera interior and exterior orientation accuracy: a 5cm GSD image resolution makes no sense if the image orientation has only 1m accuracy. For LIDAR data, it makes no sense to have 10 points per square meter in strip overlapping areas, if the strips have 1m shift. Temporal resolution in classic Remote Sensing jargon is related to the time shift between two passages of the sensor over the same scene. In our data fusion application, this definition must be adapted to the time shift between the aerial imagery flight and the LIDAR flight. Now, a good temporal resolution means simultaneous LIDAR and imagery flights, and the longer this time shift, the more problems and errors will occur. It becomes very clear when comparing results from the Biberach data set (simultaneous flight) and the UFPR data set (about 2 months time shift). Data quality is a resolution matter. Just controlling data quality during the surveying phase can lead to good data sets, which is a prerequisite for the success of automation procedures in modern digital Photogrammetry.

Bibliography

Adams, R., & Bischof, L. 1994. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**(6), 641–647.

Alexander, C., Smith-Voysey, S., Jarvis, C., & Tansey, K. 2009. Integrating building footprints and LiDAR elevation data to classify roof structures and visualise buildings. *Computers, Environment and Urban Systems*, **33**(4), 285–292.

Alharthy, A., & Bethel, J. 2002. Heuristic filtering and 3D feature extraction from LIDAR data. *Pages 29–34 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXIV 3A.

Araki, I. 2005. *Fusão de Informações Espectrias, Altimétrica e de Dados Auxiliares na Classificação de Imagens de Alta Resolução Espacial*. Curso de Pós-Graduação em Ciências Geodésicas - Universidade Federal do Paraná.

Baltsavias, E. 1999. A comparison between photogrammetry and laser scanning. *ISPRS Journal of Photogrammetry and Remote Sensing*, **54**(2-3), 83–94.

Baltsavias, E. 2004. Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. *ISPRS Journal of Photogrammetry and Remote Sensing*, **58**(3-4), 129–151.

Baltsavias, E., Mason, S., & Stallmann, D. 1995. Use of DTMs/DSMs and orthoimages to support building extraction. *In: Workshop on AEMOASI, Basel*.

Bauer, K. 2009. *City planning for civil engineers, environmental engineers, and surveyors*. CRC Press, Boca Raton.

Bähr, H.-P. 1976. *Analyse der Geometrie auf Photodetektoren abgetasteter Aufnahmen von Erderkundungssatelliten*. Lehrstühle für Geodäsie, Photogrammetrie und Kartographie an der Technischen Universität Hannover.

Blaschke, T., & Strobl, J. 2001. What's wrong with pixels? Some recent development interfacing remote sensing and GIS. *GeoBIT/GIS*, **6**, 12–17.

Bonissone, P., & Decker, K. 1986. *Uncertainty in Artificcial Inteligence*. North-Holland. Chap. Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-off Precision and Complexity.

Brovelli, M., Longoni, U., & Cannata, M. 2004. LIDAR data filtering and DTM interpolation within GRASS. *Transactions in GIS*, **8**(2), 155–174.

Brown, D. 1971. Close-range camera calibration. *Photogrammetric Engineering*, **37**(8), 855–866.

Brunn, A., & Weidner, U. 1997. Extratiing buildings from digital surface models. *In: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXII-3/4W2 Stuttgart.

Burger, W., & Burge, M. 2007. *Digital Image Processing: An Algorithmic Introduction using Java*. Springer.

Burrough, P., & Mcdonnell, R. 1998. *Principles of Geographical Information Systems*. Oxford University Press, USA.

Cheng, Y. 1995. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(8), 790–799.

Cho, W., Jwa, Y., Chang, H., & Lee, S. 2004. Pseudo-grid based building extraction using airborne LIDAR data. *Pages 378–381 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXV(b3).

Clarke, K. 2003. *Getting Started with Geographic Information Systems*. 4. ed. edn. Prentice Hall series in geographic information science. Prentice Hall.

Collins, R. 2003 (June). Mean-shift blob tracking through scale space. *Pages 234–240 of: Computer Vision and Pattern Recognition, 2003*, vol. 2.

Comaniciu, D., & Meer, P. 1997. Robust Analysis of Feature Spaces: Color Image Segmentation. *Pages 750–755 of: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*

Comaniciu, D., & Meer, P. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(5), 603–619.

Comaniciu, D., Ramesh, V., & Meer, P. 2001. The variable bandwidth mean-shift and data-driven scale selection. *Pages 438–445 of: Proceedings of the 8th International Conference on Computer Vision.*

Cronk, S., Fraser, C., & Hanley, H. 2006. Automated metric calibration of colour digital cameras. *The Photogrammetric Record*, **21**(116), 355–372.

DIN 18716-2. 2007. *Deutsches Institut für Normung — Photogrammetrie und Fernerkundung — Teil 2: Besondere Begriffe der photogrammetrischen Auswertung.*

DIN EN ISO 9000. 2005. *Deutsches Institut für Normung — Quality management systems — Fundamentals and vocabulary.*

Doerstel, C., Jacobsen, K., & Stallmann, D. 2003. DMC - Photogrammetric accuracy - Calibration aspects and generation of synthetic DMC images. *Optical 3-D Measurement Techniques VI, Institut for Geodesy and Photogrammetry (Zurich)*, **I**, 74–88.

Dubois, D., & Prade, H. 2000. *Fundamentals of fuzzy sets.* Kluwer Academic Publishers Group, Dordrecht.

Ebner, H. 1976. Self calibrating block adjustment. *In: International Archives of the XIIth ISPRS Congress.*

Elberink, S., & Mass, H. 2000. The use of anisotropic height texture measures for the segmentation of airborne laser scanner data. *Pages 678–684 of: International Archieves of Photogrammetry and Remote Sensings*, vol. XXXIII B3.

Elmqvist, M., Jungert, E., Lantz, F., Persson, Å., & Söderman, U. 2001. Terrain modelling and analysis using laser scanner data. *Pages 219–226 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXIV-3/W4 Annapolis.

Fan, J., Yau, D., Elmagarmid, A., & Aref, W. 2001. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Transactions On Image Processing*, **10**, 1454–1466.

Fan, J., Zeng, G., Body, M., & Hacid, M-S. 2005. Seeded region growing: an extensive and comparative study. *Pattern Recognition Letters*, **26**(8), 1139–1156.

Filin, S. 2004. Surface classification from airborne laser scanning data,. *Computers & Geosciences*, **30**(9-10), 1033–1041.

Forsyth, D., & Ponce, J. 2002. *Computer Vision: A Modern Approach.* Prentice Hall.

Fowler, R., Little, J., & Mark, D. 1978. The triangulated irregular network. *Pages 24–31 of: ASPRS Congress on Survey and Mapping, St. Louis.*

Fraser, C. 1997. Digital camera celf-calibration. *ISPRS Journal of Photogrammetry and Remote Sensing*, **52**(4), 149–159.

Förstner, W. 1994. A framework for low level feature extraction. *Pages 383–394 of: Computer Vision - ECCV 94*, vol. II.

Fukanaga, K., & Hostetler, L. 1975. The estimation of tge gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, **21**, 32–40.

Gao, Y., & Mas, J. 2008. A comparison of the performance of pixel-based and object-based classifications over images with various spatial resolution. *In: International Archieves of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII 4/C1.

Grün, A. 1978. Progress in photogrammetric point determination by compensation of systematic errors and detection of gross errors. *Nachrichten aus dem Karten- und Vermessungswesen Series II*, **36**, 113–140.

Haala, N., & Brenner, C. 1999. Extraction of buildings and trees in urban environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, **54**(2-3), 130–137.

Haala, N., Brenner, C., & Anders, K. 1998. 3D urban GIS from laser altimeter an 2D map data. *Pages*

339–346 of: International Archieves of Photogrammetry and Remote Sensing, vol. XXXII 3/1.

Habib, A., & Morgan, M. 2005. Stability analysis and geometric calibration of off-the-shelf digital cameras. *Photogrammetric Engineering and Remote Sensing*, 733–741.

Habib, A., Ghanma, M., & Mitishita, E. 2004. Co-registration of photogrammetric and LIDAR data: methodology and case study. *Revista Brasileira de Cartografia*.

Habib, A., Pullivelli, A., Mitishita, E., Ghanma, M., & Kim, E-M. 2006. Stability analysis of low-cost digital cameras for aerial mapping using different georeferencing techniques. *The Photogrammetric Record*, **21**(March), 29–43.

Habib, A., Kersting, J., McCaffrey, M., & Jarvis, A. 2008. Integration of lidar and airborne imagery for realistic visualization of 3d urban environments. *Pages 617–624 of: Proceedinds of the ISPRS Congress Beijing 2008*, vol. Comission III, WG II/5.

Hansen, W. von, & Vögtle, T. 1999. Extraktion der Geländeoberfläche aus flugzeuggetragenen Laserscanner-Aufnahmen. *Photogrammetrie Fernerkundung Geoinformation*, **4**, 229–236.

Hartley, R., & Zisserman, A. 2004. *Multiple View Geometry in Computer Vision*. Second edn. Cambridge University Press.

Hay, G., & Castilla, G. 2006. Object-based image analysis: strength, weakness, opportunities, and threats (SWOT). *In: 1st. International Conference on Object-based Image Analysis (OBIA 2006)*.

Henricson, O., Bignone, F., Willuhn, W., Ade, F., Kubler, O., Baltsavias, E., Mason, S., & Gruen, A. 1996. Project Amobe: strategies, current status and future work. *Pages 321–330 of: International Archieves of Photogrammetry and Remote Sensings*, vol. XXXI.

Herrera, F., Herrera-Viedma, E., & Martínez, L. 2002. *Aggregation Operators: New Trends and Applications*. Physica-Verlag, Heidelberg. Chap. Representation Models for Aggregating Linguistic Information: Issues and Analysis.

Hofamnn, P. 2001. Detecting buildings and roads from IKONOS data using additional elevation information. *GIS*, **6**, 28–33.

Hofmann, A., Maas, H-G., & Streilein, A. 2002. Knowledge-based building detection based on laser scanner data and topographic map information. *Pages 169–174 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXIV 3A.

Honkavaara, E., Ahokas, E., Hyyppä, J., Jaakkola, J., Kaartinen, H., Kuittinen, R., Markelin, L., & Nurminen, K. 2006. Geometric test field calibration of digital photogrammetric sensors. *ISPRS Journal of Photogrammetry and Remote Sensing*, **60**(6), 387–399.

Hu, Y. 2003. *Automated extraction of digital terrain models, roads and building using airborne LIDAR data*. Department of Geomatic Engineering - University of Calgary.

Hug, C., & Wehr, A. 2007. Detecting and identifying topographic objects in imaging laser altimeter data. *Pages 19–26 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXVII.

INPE, Instituto Nacional de Pesquisas Espaciais. 2009. *Engenharia de Satélites. URL: http://www.inpe.br/ Acessed in 6.11.2009*.

Jacobsen, K. 1982. Programmgesteuerte Auswahl Zusätzlicher Parameter. *Bildmessung und Luftbildwesen*, **50**, 213–217.

Jacobsen, K. 2001. Direct georeferencing. *Photogrammetric Engineering and Remote Sensing*, **67**(12).

Jauregui, M., Vílchez, J., & Chacón, L. 2002. A procedure for map updating using digital mono-plotting. *Computers & Geosciences*, **28**(4), 513–523.

Jähne, B. 2005. *Digital Image Processing*. 6. rev. and extended ed. edn. Berlin: Springer.

Kim, K., Jung, K., & Kim, J. 2003. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(12), 631–1639.

Klir, G., & Yuan, B. 1995. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice-Hall, London.

Kothuri, R., Godfrind, A., & Beinat, E. 2007. *Pro Oracle Spatial for Oracle Database 11g*. Springer-Verlag New York.

Kraus, K. 1996. *Photogrammetrie Band 2: Verfeinerte Methoden und Anwendungen*. Bonn: Dümmlers Verlag.

Kraus, K. 2007. *Photogrammetry : Geometry from Images and Laser Scans*. 2. ed. edn. de Gruyter Textbook. Berlin: de Gruyter.

Kraus, K., & Pfeifer, N. 1998. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, **53**, 193–203.

Kunii, Y., & Chikatsu, H. 2001. On the application of 3 million consumer digital camera to digital photogrammetry. *Pages 278–287 of: Proceedigs of the SPIE Videometrics VII*, vol. 4309.

Läbe, T., & Förstner, W. 2004. Geometric stability of low-cost digital consumer cameras. *Pages 528–535 of: Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Sciennces*, vol. 35(5).

Longley, P., Goodchild, M., Maguire, D., & Rhind, D. 2005. *Geographic Information Systems and Science*. 2 edn. New York, NY: John Wiley & Sons.

Lu, Y., Trinder, J., & Kubik, K. 2006. Automatic building detection using the Dempster-Shafer algorithm. *Photogrammetric Engineering and Remote Sensing*, **72**(4), 395–404.

Maas, H-G. 1999. The potential of height texture measures for the segmentation of airborne laserscanner data. *Fourth International Airborne Remote Sensing Conference and Exhibition*, Ottawa, Ontario, Canada.

Maas, H-G., & Vosselman, G. 1999. Two algorithms for extracting building models from raw laser altimetry data. *ISPRS Journal of Photogrammetry and Remote Sensing*, **54**(2-3), 153 – 163.

Machado, A. 2006. *Extração Automática de Contornos de Edificações Utulizando Imagem Gerada por Câmara Digital de Pequeno Formato e Dados Laser*. Curso de Pós-Graduação em Ciências Geodésicas - Universidade Federal do Paraná.

Machado, A., Mitishita, E., & Santos, R. 2004 (September). A case study: Digital camera systematic errors. *In: Anais do I Simpósio das Ciências Geodésicas e Tecnologias da Geoinformação*. Universidade Federal do Recife.

Makarovic, B. 1973. Digital mono-plotters. *I.T.C. Journal*, **1**, 101–122.

Marichal, J-L. 2002. *Aggregation Operators: New Trends and Applications*. Physica-Verlag, Heidelberg. Chap. Aggregation of Interacting Criteria by Means of Discrete Choquet Integral.

Masaharu, H., & Ohtsubo, K. 2002. A filtering method of airborne laser scanner data for complex terrain. *In: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXIV, part 3A/B.

MathWorks. 2009. *Matlab Fuzzy Logic Toolbox User's Guide*.

Matikainen, L., & Hyyppa, J. 2003. Automatic detection of buildong from laser scanner data for map updating. *Pages 218–224 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXIV 3/W13.

Mehnert, A., & Jackway, P. 1997. An improved seeded region growing algorithm. *Pattern Recognition Letters*, **18**(10), 1065–1071.

Meinel, G., & Neubert, M. 2004 (July). A Comparision of segmentation programs for high resolution remote sensing data. *Pages 1097–1102 of: Proceedings XXth ISPRS Congress, Istanbul*, vol. XXXV-B4.

Meng, X., Wang, L., Silván-Cárdenas, J., & Currit, N. 2009. A multi-directional ground filtering algorithm for airborne LIDAR. *ISPRS Journal of Photogrammetry and Remote Sensing*, **64**(1), 117–124.

Mitishita, E. 1997. *Monorestituição Digital de Aerofotos, Associada com Sistema de Computação Gráfica C.A.D., para fins de Mapeamento na Área Florestal*. Setor de Ciências Agrárias - Universidade Federal do Paraná.

Mitishita, E., Machado, A., Habib, A., & Gonçalves, G. 2004. 3D monocular restitution applied to small format digital airphoto and laser scanner data. *In: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXV III/1.

Mitishita, E., Habib, A., Centeno, J., Machado, A., Lay, J., & Wong, C. 2008. Photogrammetric and lidar data integration using the centroid of a rectangular roof as a control point. *The Photogrammetric Record*, **23**(121), 19–35.

Moniwa, H. 1977. *Analytical photogrammetric system with self-calibration and its applications*. Department of Surveying Engineering - University of New Brunswick.

Neteler, M., & Mitasova, H. 2008. *Open Source GIS: A GRASS GIS Approach*. Third edn. New York: Springer.

Neubert, M., Herold, H., & Meinel, G. 2006 (July). Evaluation of remote sensing image segmentation quality - Further results and concepts. *In: Proceedings 1st International Conference on Object-based Image Analysis (OBIA 2006), Salzburg*, vol. XXXVI-4/C42.

Neubert, M., Herold, H., & Meinel, G. 2008 (August). Assessment of remote sensing image segmentation quality. *In: Proceedings GEOBIA 2008, Calgary, Canada*, vol. XXXVIII-4/C1.

Palmer, T., & Shan, J. 2002. A comparative study on urban visualization using LIDAR data and GIS. *URISA Journal*, **14**(2), 19–25.

Park, J., Lee, G., & Park, S. 2009. Color image segmentation using adaptive mean shift and statistical model-based methods. *Computers and Mathematics with Applications*, **57**(6), 970–980. Advances in Fuzzy Sets and Knowledge Discovery.

Petzold, B., Reiss, P., & Stossel, W. 1999. Laser scanning - surveying and mapping agencies are using a new technique for the derivation of digital terrain models. *ISPRS Journal of Photogrammetry and Remote Sensing*, **54**, 95–104.

Rehor, M., Bähr, H.-P., Tarsha-Kurdi, F., Landes, T., & Grussenmeyer, P. 2008. Contribution of two plane detection algorithms to recognition ofintact and damaged buildings in LIDAR data. *The Photogrammetric Record*, **23**(124), 441–456.

Remondino, F., & Fraser, C. 2006. Digital camera calibration methods: Considerations and comparisons. *In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. Vol. XXXVI.

Rönnholm, P., Honkavaara, E., Litkey, P., Hyyppä, H., & Hyyppä, J. 2007. Integration of laser scanning and photogrammetry. *Pages 355–362 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXVI, Part 3 / W52.

Rottensteiner, F., & Briese, C. 2002. A new method for building extraction in urban areas from high-resolution LIDAR data. *Pages 295–301 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXIV 3A.

Rottensteiner, F., Trinder, J., Clode, S., & Kubik, K. 2007. Building detection by fusion of airborne laser scanner data and multi-spectral images: Performance evaluation and sensitivity analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, **62**(2), 135–149.

Santos, R., Mitishita, E., Bähr, H-P., & Vögtle, T. 2007. Methodolgy to automatic integration of digital aerial images and LIDAR data. *Boletim de Ciências Geodésicas*, **13**(2), 369–394.

Schenk, T., & Csathó, B. 2002. Fusion of LIDAR data and aerial imagery for a more complete surface description. *Pages 310–317 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXIV 3A.

Shih, F., & Cheng, S. 2005. Automatic seeded region growing for color image segmentation. *Image and Vision Computing*, **23**(10), 877–886.

Silverman, B. 1986. *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

Sithole, G. 2001. Filtering of laser altimetry data using a slope adaptive filter. *Pages 203–210 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXIV, part 3/W4.

Sithole, G. 2005. *Segmentation and classification of laser scanner data*. Ph.D. Thesis. Publications on Geodesy, 59. Publication of Netherlands Geodetic Commision.

Skurikhin, A., & Volegov, P. 2008. Object-oriented hierarchical image vectorization. *In: International Archieves of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII 4/C1.

Sohn, G., & Dowman, I. 2008. A model-based approach for reconstructing a terrain surface from airborne LIDAR data. *The Photogrammetric Record*, **23**(122), 170–193.

Stassopoulou, A., Caelli, T., & Ramirez, J. 2000. Automatic extraction of building statistics from digital orthophotos. *International Journal of Geographical Information Science*, **14**(8), 795–814.

Steiniger, S., Lange, T., Burghardt, D., & Weibel, R. 2008. An approach for the classification of urban building structures based on discriminant analysis techniques. *Transactions in GIS*, **12**(1), 31–59.

Steinle, E. 2005. *Gebäude-Modellierung und -Änderungserkennung basierend auf multitemporalen Laserscanningdatensätzen*. Intitut of Photogrammetry and Remote Sensing - University of Karlsruhe.

Steinle, E., & Vögtle, T. 2001. Automated extraction and reconstruction of buildings in laserscanning data for disaster management. *Pages 309–318 of: Automatic Extraction of Man-Made Objects from Aerial and Space Images (III), E. Baltsavias et al. (eds.),Swets & Zeitlinger, Lisse, The Netherlands*.

Sudhamani, M., & Venugopal, C. 2006. Segmentation of color images using mean shift algorithm for feature extraction. *Pages 241–242 of: 9th International Conference on Information Technology 2006 ICIT'06*.

Tovari, D. 2006. *Segmentation Based Classification of Airborne Laser Scanner Data*. Intitut of Photogrammetry and Remote Sensing - University of Karlsruhe.

Tovari, D., & Vögtle, T. 2004. Object classification in laser scanner data. *In: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXIV, VI,III/4.

Vestri, C., & Devernay, F. 2001. Using robust methods for automatic extraction of buildings. *Pages 133–138 of: CVPR*, vol. 1.

Vögtle, T., & Steinle, E. 2003. On the quality of object classification and automated building modelling based on laser scanning data. *Pages 149–155 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXIV, part 3/W13.

Vosselman, G. 1999. Building reconstruction using planar faces in very high-density data. *In: International Archieves of Photogrammetry and Remote Sensings*, vol. XXXII.

Vosselman, G. 2000. Slope based filtering of laser altimetry data. *Pages 935–942 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXIII, part B3/2.

Vosselman, G., & Dijkman, S. 2001. 3D urban GIS from laser altimeter an 2D map data. *Pages 37–43 of: International Archieves of Photogrammetry and Remote Sensing*, vol. XXXIV 3/W4.

Vosselman, G., Kessels, P., & Gorte, B. 2005. The utilisation of airborne laser scanning for mapping. *International Journal of Applied Earth Observation and Geoinformation*, **6**(3-4), 177 – 186.

Vu, T., Yamazaki, F., & Matsuoka, M. 2009. Multi-scale solution for building extraction from LiDAR and image data. *International Journal of Applied Earth Observation and Geoinformation*, **11**(4), 281–289.

Wald, L. 1999. Some terms of reference in data fusion. *IEEE Transactions on Geoscience and Remote Sensing*, **37**(3).

Wang, Z., & Schenk, T. 2000. Building extraction and reconstruction from lidar data. *Pages 958–964 of: International Archieves of Photogrammetry and Remote Sensings*, vol. XXXIII B3.

Weidner, U., & Förstner, W. 1995. Towards automatic building extraction from high resolution digital elevation models. *ISPRS Journal of Photogrammetry and Remote Sensing*, **50**, 38–49.

Wendt, A. 2007. A concept for feature based data registration by simultaneous consideration of laser scanner data and photogrammetric images. *ISPRS Journal of Photogrammetry and Remote Sensing*, **62**(2), 122 – 134.

Worboys, M. 1998. Computation with imprecise geospatial data. *Comput., Environ., and Urban Systems*, **22**(2), 85–105.

Yang, X., & Pei, J. 2008. Multimodal medical image elastic registration using mean shift. *International Conference on Natural Computation*, **4**, 177–181.

Yoon, T., Kim, T., Park, W., & Kim, T. 1999. Building segmentation using active contour model. *Pages 27–30 of: Joint ISPRS Workshop on Sensors and Mapping from Space - Hannover*.

Zadeh, L. 1965. Fuzzy sets. *Informat. Control*, **8**, 338–353.

Zhan, Q., Molenaar, M., & Tempfli, K. 2002. Building extraction from laser data by reasoning on image segments in elevation slices. *Pages 305–308 of: International Archieves of Photogrammetry and Remote Sensings*, vol. XXXIV.

Zhou, G., Song, C., Simmers, J., & Cheng, P. 2004. Urban 3D GIS From LiDAR and digital aerial images. *Computers & Geosciences*, **30**(4), 345–353.

Zhou, H., Yuan, Y., & Shi, C. 2009. Object tracking using SIFT features and mean shift. *Computer Vision and Image Understanding*, **113**(3), 345–352. Special Issue on Video Analysis.

Acknowledgments

First and foremost, I must thank my parents, especially my mother, for their great support and understanding. I would like to thank God for life and its opportunities. I would thank all colleagues here at KIT (former University of Karlsruhe), in the IPF (Institute of Photogrammetry and Remote Sensing). Special thanks go to my advisors Prof. Dr. Hans-Peter Bähr and Prof. Dr. Stefan Hinz (in Karlsruhe) and Prof. Dr. Edson Mitishita (in Curitiba), for their great help. Special thanks go also for my IPF colleague Patrick Bradley, for the careful text review. I would thank the LACTEC and Toposys for the UFPR and Biberach data sets respectively. Support from DAAD (Deutscher Akademischer Austausch Dienst – in Germany) and CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – in Brazil) is gratefully appreciated.