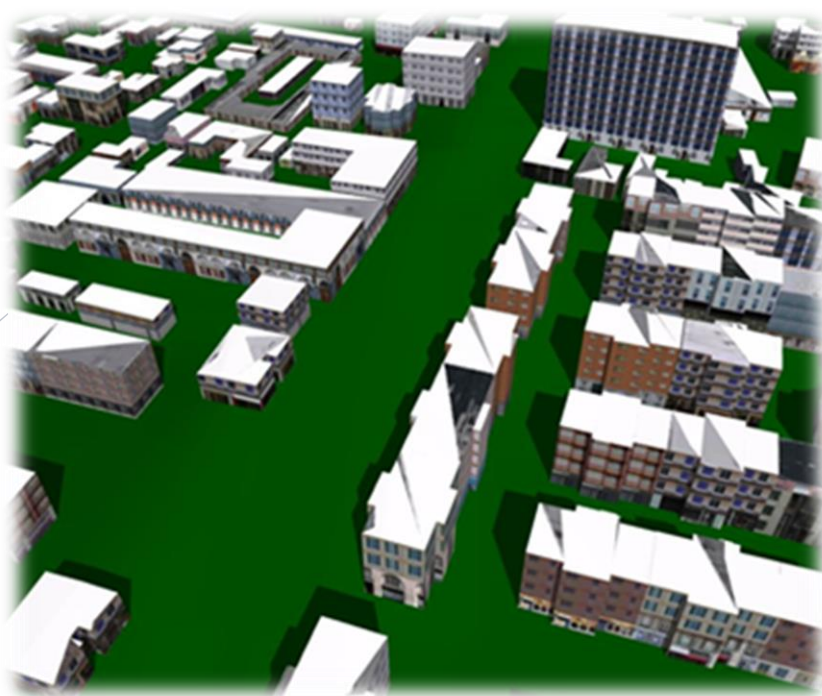


UTCB

Contributions to the automation of geospatial data extraction for 3D modelling

PhD Thesis - Summary



Supervisor: Prof.Univ.Ph.D. Petre Iuliu DRAGOMIR

Author: Iuliana Maria BINĂ (PÂRVU)

June 2021

Acknowledgment

The scientific elaboration of the present doctoral thesis represents the result of the researches carried out between October 2018 - May 2021, within the Technical University of Civil Engineering Bucharest, Faculty of Geodesy, in Civil Engineering.

First, I would like to thank my supervisor, Mr. Prof.Ph.D. Petre Iuliu DRAGOMIR, who guided my steps during my doctoral studies and offered me didactic support and useful ideas for the elaboration of my doctoral thesis. There were some difficult moments during the last year, in which the trust offered by him helped me to continue and reach the completion of this research.

I would like to thank the entire team of the Department of Topography and Cadastre, Faculty of Geodesy, Bucharest, for the scientific recommendations and assessments they offered me with the occasion of defending my doctoral thesis in the department.

I would like to thank the head of the *National Center of Cartography* and of the *Cartography and Photogrammetry Department* for their support in providing the data underlying the research and the opportunity to actively participate in the European research project - VOLTA. I would like to thank Mr. Fabio REMONDINO, head of the 3D Optical Metrology research unit of the Bruno Kessler Foundation (FBK), and Mr. Emre OZDEMIR, a Ph.D. student in his team, for their fruitful collaboration during the 2 months delegation at FBK (Trento, Italy).

Even though I didn't know him personally, but only through video courses, I want to thank Mr. Ian WITTEN of the University of Waikato, New Zealand, from whom I learned important things about *Data mining* and the Weka software.

A lot of thanks to the Department of Geodesy and Geoinformation - Research Groups Photogrammetry and Remote Sensing of the Technical University of Vienna for providing me the OPALS software, used in a step of the workflow proposed by me in the thesis.

I would like to thank my parents for their support and trust over the years of study. I want to tell my mother that what I am is due to her.

The most important supporters during my doctoral studies were my sons, Eduard (9 years old) and Mihnea (5 years old), and my husband, Adrian. Even though it was hard for them because we hadn't had so much time together, especially in the last year, they were confident in my abilities and always encouraged me. My husband gave me professional advice and helped me get through all the hard moments.

Table of contents

Table of contents.....	2
1. Introduction.....	7
1.1. General introduction	7
1.2. Motivation.....	7
1.3. Objectives and purpose	7
1.4. Organization of the Thesis	8
2. Geospatial Data Acquisition by Laser Scanning	10
2.1. Introduction	10
2.2. Short history	10
2.3. Introductory notions.....	10
2.4. Classification of laser scanning systems	10
2.5. Laser scanning applications	11
2.6. Traditional airborne laser scanning	11
2.6.1. Physical principles of ALS systems	11
2.6.2. ALS system components	12
2.6.2.1. Laser scanner	12
2.6.2.2. POS system	12
2.6.3. Reflectance and interaction with objects.....	12
2.6.4. The evolution of ALS systems	12
2.6.5. Airborne laser scanning workflow	13
2.6.5.1. Flight planning	13
2.6.5.2. Field measurements	13
2.6.5.3. Survey with ALS system	13
2.6.5.4. LiDAR data processing	13
2.6.5.5. Final products generation.....	14
2.7. Single Photon LiDAR	14
2.8. Geiger-mode scanner.....	14
2.9. LiDAR data for the case study.....	14
3. Acquisition of geospatial data by aerial photogrammetry	16
3.1. Introduction	16
3.2. Photogrammetric evolution.....	16
3.3. Photogrammetry principles.....	16

3.4.	Aerial photogrammetric system	17
3.4.1.	Vertical cameras	17
3.4.2.	Oblique cameras	17
3.5.	Photogrammetric workflow.....	17
3.5.1.	Mission planning.....	17
3.5.2.	Field measurements and image acquisition.....	18
3.5.3.	Tie points extraction.....	18
3.5.4.	Bundle adjustment.....	18
3.5.5.	DIM and DSM generation	18
3.5.6.	Orthophoto generation	18
3.6.	Aerial photogrammetry applications	19
3.7.	Photogrammetric data for the case study	19
4.	Extracting data from LiDAR point clouds	20
4.1.	Introduction and purpose	20
4.2.	Point cloud segmentation	20
4.2.1.	Edge based methods.....	20
4.2.2.	Region based methods.....	20
4.2.3.	Attributes based methods.....	20
4.2.4.	Model fitting methods	20
4.2.5.	Graph based methods	21
4.2.6.	Hybrid methods.....	21
4.2.7.	Machine learning based methods	21
4.3.	Point cloud classification based on Machine learning algorithms	21
4.3.1.	3D point features	22
4.3.1.1.	PCA, eigenvalues, and eigenvectors	22
4.3.1.2.	About point features	22
4.3.2.	Neighborhood selection	22
4.3.3.	3D feature extraction	23
4.3.4.	Feature selection	23
4.3.5.	Supervised classification	23
4.3.6.	Accuracy assessment	24
4.4.	Case study - point cloud classification.....	24
4.4.1.	Preparing the point cloud	24
4.4.2.	Dynamic neighborhood selection	25

4.4.3.	Feature extraction	26
4.4.4.	Classification and accuracy assessment	26
4.4.4.1.	Weka.....	26
4.4.4.2.	Training and testing data sets.....	26
4.4.4.3.	Machine learning classifiers	27
4.4.4.4.	Analysis of the results for the used classifiers.....	27
4.4.4.5.	Point cloud classification workflow.....	29
5.	Extracting data from orthorectified images	30
5.1.	Introduction and purpose	30
5.2.	Introductory notions.....	30
5.3.	Extracting data from images using Machine learning algorithms	30
5.4.	Extracting data from images using ANN.....	30
5.4.1.	Brief history of the development of neural networks.....	30
5.4.2.	ANN definitions	31
5.4.3.	Natural neuron versus artificial neuron.....	31
5.4.3.1.	Natural neuron.....	31
5.4.3.2.	Artificial neuron	31
5.4.4.	ANN architectures	32
5.4.5.	Deep neural network.....	32
5.4.6.	Learning process with ANN	33
5.4.7.	ANN applications	33
5.4.8.	Basic CNN architectures	33
5.4.8.1.	AlexNet.....	33
5.4.8.2.	VGGNet	33
5.4.8.3.	NiN Net.....	33
5.4.8.4.	GoogLeNet.....	34
5.4.8.5.	ResNet.....	34
5.4.8.6.	DenseNet	34
5.4.9.	Architectures used for semantic segmentation of images	34
5.4.9.1.	FCN.....	34
5.4.9.2.	SegNet.....	34
5.4.9.3.	U-net	34
5.4.9.4.	DeconvNet	34
5.5.	Accuracy assessment of image classification and segmentation	35

5.6.	ANN frameworks.....	35
5.7.	Case study - image semantic segmentation.....	35
5.7.1.	Image classification using Machine learning algorithms.....	35
5.7.1.1.	Supervised pixel-based classification	35
5.7.1.2.	Supervised object-based classification.....	37
5.7.2.	Image classification based on DNN.....	37
6.	3D Building Modelling based on point clouds.....	38
6.1.	Introduction and purpose	38
6.2.	3D building models generation	38
6.2.1.	Model-based generation of 3D building models.....	38
6.2.2.	Data-based generation of 3D building models	38
6.2.3.	Point cloud processing software.....	39
6.3.	CityGML level of detail.....	39
6.3.1.	CityGML 2.0.....	39
6.3.2.	CityGML 3.0.....	40
6.4.	Applications of 3D models of buildings	40
6.5.	The current international status in 3D building modelling	40
6.6.	Software for 3D building models generation	41
6.6.1.	BuildingReconstruction	41
6.6.2.	Esri CityEngine	41
6.6.3.	3dfier.....	41
6.6.4.	PolyFit.....	41
6.7.	Geometric and semantic errors in CityGML data sets	42
6.7.1.	Geometric and semantic types of errors	42
6.7.2.	Topology validation software	42
6.8.	3D GIS and BIM	42
6.9.	Case study – 3D building models generation.....	42
6.9.1.	Extraction of buildings footprints from LiDAR point clouds.....	42
6.9.1.1.	Generation of buildings footprints from segmented point clouds	42
6.9.1.2.	Post-processing of segmented polygons.....	44
6.9.2.	Generating 3D building models in 3dfier.....	45
6.9.3.	Generating 3D building models in CityEngine.....	45
6.9.3.1.	Buildings height extraction using centroid points	46
6.9.3.2.	Buildings height extraction using characteristic points	47

6.9.4. Accuracy assessment of the 3D building models	48
7. Generation and online distribution of 3D building models for the city of Arad.....	50
7.1. Introduction and purposes	50
7.2. 3D building models for the city of Arad.....	50
7.2.1. Input data	50
7.2.2. Data set processing.....	50
7.2.3. 3D building models generation in CityEngine	52
7.2.4. Quality evaluation of the 3D models generated for Arad.....	52
7.3. Online distribution of 3D building models	53
7.3.1. ArcGIS Online and CityEngine WebViewer	53
7.3.2. Data distributed online	53
8. General conclusions, original contributions, and future perspectives	55
8.1. General conclusions	55
8.2. Original contributions	59
8.3. Future perspectives	60
Bibliography	61
Published articles	66

1. Introduction

1.1. General introduction

No matter if we talk about LiDAR point clouds or orthorectified images, the extraction of geospatial data from photogrammetric or remote sensing products particularly interests researchers in our field. Until recently, the elements of the map were, exclusively, digitized, lately semi-automated or even automated extraction is the solution used by most companies.

The classification or segmentation of point clouds or images can be done by various techniques and based on a wide range of algorithms. The most used methods are those based on Machine learning or even on deep artificial neural networks.

Local administrations and not only have to perform complex tasks for urban planning. Therefore, they need a virtual representation of their city. Thus, the need to move from 2D to 3D modelling is obvious and extremely necessary. 3D models of cities are becoming a concern for the national mapping agencies in many countries all around the world.

1.2. Motivation

In most of the Ph.D. thesis I analyzed, as well as in articles, the topic of 3D building modelling is debated at individual building level (churches, monuments, etc.) or in small groups of buildings. In my doctoral thesis, I set out to generate the 3D building models for a large area, such as a big city, represented as the municipality of a county in Romania. The difficulty of generating 3D building models lies, in this case, a large amount of data and the increased need for hardware-software performance. Identifying a workflow that leads to good modelling quality is another big challenge.

1.3. Objectives and purpose

In this thesis, I set the focus on buildings.

The major objectives of the thesis are: automated extraction of building contours from both point clouds and orthorectified images and generation of 3D building models with the level of detail LoD1 and LoD2. Besides these objectives, which emerge from the title of the thesis, I set out to provide a specialised and current framework in the fields of laser scanning

and aerial photogrammetry in terms of workflows, data processing and state-of-the-art in 3D building modelling. Another goal is to identify an optimized workflow for classifying ALS point clouds.

I performed the tests on an area of Arad with low and medium height buildings. I chose this area because of the challenges that the vegetation creates both in extracting the buildings from point clouds and from photogrammetric images. Finally, I generated 3D building models on the entire surface of the Arad municipality.

The purpose of the doctoral thesis is to create a framework for the authorities to take the steps in order to launch a national project to generate 3D building models for all the cities in Romania and to publish them, as open data, on a national portal.

1.4. Organization of the Thesis

I structured the thesis into 8 chapters:

Chapter 1, called *Introduction*, contains a brief presentation of the Ph.D. thesis, the objectives, the primary motivation for the topic chosen, and the description of the structure at the chapter level.

Chapter 2, called *Geospatial Data Acquisition by Laser Scanning*, covers the history of LiDAR technology, the description of traditional airborne laser scanning systems (operating principle, the detailed panel of systems produced by the large companies in the field, their applications, complete workflow for obtaining spatial data by laser scanning, etc.), the description of the two modern scanning technologies: SPL (Single Photon LiDAR) and Geiger-mode, as well as the description of the LiDAR data set that I will use in the case study.

Chapter 3, entitled *Acquisition of geospatial data by aerial photogrammetry*, includes information about the photogrammetric evolution from analog-to-digital, the characteristics of nadir and oblique top cameras, the steps of a photogrammetric workflow, from flight planning to orthophotos generation (DTM/DSM orthophotos), and the description of the photogrammetric data set for the case study.

Chapter 4, called *Extracting data from LiDAR point clouds*, describes the segmentation and classification methods for point clouds, and the case study part, which will detail the proposed workflow for LiDAR point cloud classification, including the accuracy assessment based on various machine learning algorithms. In the workflow, I will describe my approach to the dynamic neighborhood selection, use a series of 3D point features for classification, and test the Weka software for supervised point cloud classification. Until now, this software was used to solve the problems of extracting patterns from large data sets, with applications in banking, diagnostics, marketing, and sales.

Chapter 5, called *Extracting data from orthorectified images*, describes the most popular methods of semantic segmentation of images based on Machine learning techniques and a review of the most popular artificial neural network (ANN) architectures for classification and semantic segmentation of images. In the case study part, I will describe all the stages of the workflow for the semantic segmentation of images, both with Machine learning algorithms and through deep neural networks.

Chapter 6, called *3D Building Modelling based on point clouds*, summarizes the general framework for 3D building modelling, starting with the generation methods, continuing with current information about the CityGML standard (City Geography Markup Language), detailing applications of 3D building modelling, and highlighting international state-of-the-art on the topic. As a preamble to the case study, I will describe some software that allows the generation of 3D models and I will identify the geometric and semantic errors encountered in 3D building modelling. The case study will include the extraction of the buildings footprints from the point clouds previously classified in Chapter 4 and the regularization of their shape. I will address the problem of 3D building modelling in two software: 3dfier and Esri CityEngine.

Chapter 7, called *Generation and online distribution of 3D building models for the city of Arad*, describes the 3D modelling at the city level, based on the data from the national database TopRo5 and from the LAKI II project. I will do the modelling in the CityEngine software. In the end, I will evaluate the quality of the 3D models and perform a study to compare the models generated using only LiDAR data and the ones generated with the current workflow. I will publish 3D building models for the city of Arad in the ArcGIS Online environment.

Chapter 8, entitled *General conclusions, original contributions, and future perspectives*, contains the general conclusions on the research carried out in this thesis, the original contributions made, as well as future research perspectives.

2. Geospatial Data Acquisition by Laser Scanning

2.1. Introduction

In this chapter, I will present the workflow for obtaining geospatial data by laser scanning, from the data acquisition flight planning to the generation of the final products. In the preamble, I will present the history of LiDAR (Light Detection and Ranging) technology, starting from 1930 until now, but also information related to the ALS systems (Airborne Laser Scanning) operation. I will detail the ALS systems produced by top companies in the field and their applications. In subchapters 2.7 and 2.8 I will present the two modern scanning technologies: SPL (Single Photon LiDAR) and Geiger-mode. In the end, I will describe the LiDAR data set for the city of Arad, the residence of Arad County, Romania, from the project *Geographic information for environment, climate change and EU integration - LAKE II* (Land Administration Knowledge Improvement) of the National Agency for Cadastre and Land Registration (ANCPI).

2.2. Short history

E.H. Synge introduced the LiDAR concept in the 1930s, and was supported by A. Einstein, in publishing his ideas for a new method of microscopic imaging in which an optical field scattered by a small gold particle could be used as a light source. The technology evolved a lot, so that in 1992, Geiger detectors were developed, by researcher R. Marino. In 2017, Leica Geosystems manufactures the SPL 100 scanner, which ensures large-area mapping at a minimal cost.

2.3. Introductory notions

LiDAR (Light Detection And Ranging), also known as LADAR, is an active remote sensing technology. The LiDAR scanner emits an intense and concentrated beam of light and measures the time required for reflections to be detected by the sensor. This information is used to determine the distance to objects (Carter, et al., 2012).

2.4. Classification of laser scanning systems

The classification of laser scanning systems can be done according to different features, such as:

1. the measurement technique: continuous wavelength or pulsed laser system;

2. the platform used: airborne, satellite, or terrestrial system;
3. the system type: simple or hybrid LiDAR system;
4. the receiver type: linear mode, Geiger (GmLiDAR) or single-photon system;
5. wavelengths used: topographic or bathymetric system;
6. the number of light sources: single or multispectral wavelength system.

2.5. Laser scanning applications

There are many applications of laser scanning, among which we can mention: large-area mapping, utility networks mapping, flood risk modelling and hazard maps creation, as well as 3D modelling, etc.

2.6. Traditional airborne laser scanning

Airborne laser scanning (ALS) enables point clouds acquisition, using aircraft, helicopter, or UAV platforms. In the following, I will present information about airborne laser scanning systems mounted on aircraft platform.

2.6.1. Physical principles of ALS systems

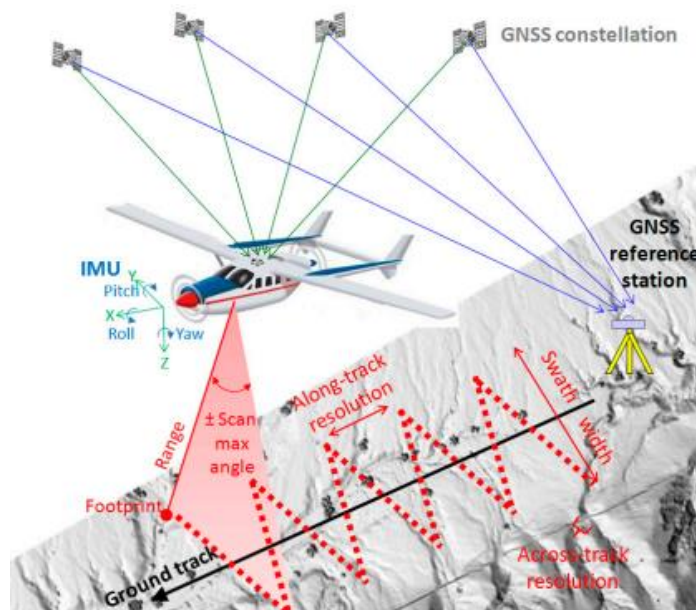


Figure 2.1 ALS system principle
(Fernandez-Diaz, Carter, Shrestha, & Glennie, 2014)

Figure 2.1 displays the measured scanning flight lines, using an oscillating mirror for the deflection of the laser beam, which determines the scan pattern in the form of the letter Z.

2.6.2. ALS system components

Figure 2.2 shows the basic components of an ALS system:

1. **airborne segment:** airborne platform, LiDAR, position, and orientation system (POS);
2. **ground segment:** GNSS reference stations and hardware-software processing component.

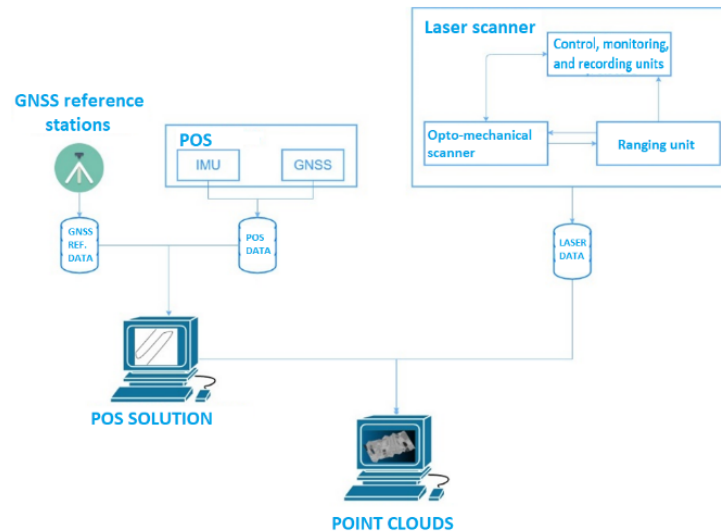


Figure 2.2 ALS system components

2.6.2.1. Laser scanner

The laser scanner comprises the ranging unit, the opto-mechanical scanner, and the control, monitoring, and recording unit.

2.6.2.2. POS system

The POS system comprises two components: IMU and GNSS. The GNSS system also uses ground reference stations, which must be on the scanning surface so that the distance between them and the ALS system does not exceed 25 km (Wehr, 2018).

2.6.3. Reflectance and interaction with objects

Reflectance is defined as the ratio of the incident radiation on a surface to the reflected radiation on that surface. In terms of LiDAR system performance, this surface feature is very important.

2.6.4. The evolution of ALS systems

The global LiDAR scanner market has grown significantly in recent years because of the sector of the autonomous vehicles. In a top 10 of LiDAR instrument manufacturers, developed by (Chakravarty, 2019), the following companies stand out:

- in Europa: 1st place - Hexagon Geosystems (Sweden), 3rd place - SICK AG (Germany), 6th place - RiegI (Austria);

- in North America: 2nd place - Velodyne LiDAR (California, USA), 4th place - TOPCON (California, USA), 5th place - Trimble (California, USA), Teledyne OPTech (Canada).

2.6.5. Airborne laser scanning workflow

The workflow for generating geospatial products through laser scanning involves planning the scanning mission, acquiring, pre-processing, and post-processing the data.

2.6.5.1. Flight planning

Based on the technical specifications that underlie any scanning project and with the help of a specialised program, the flight planning is performed. The first step is to choose the ALS system that is suitable with the project requirements.

Planning LiDAR scanning without overlapping swaths can lead to void areas; to prevent such a situation, it is recommended an overlap of at least 20%, and can even reach 50% where a high density of points is required (Heidemann, 2018). For a good planning we must consider the local meteorological reports, the big topographic changes, the existing water basins, and the land cover.

2.6.5.2. Field measurements

The field measurements comprises determining the coordinates of the points forming the control surfaces and of the checkpoints. We must survey these points in the field with a planimetric and altimetric accuracy superior to that of the LiDAR points.

2.6.5.3. Survey with ALS system

Before the actual scan, the instrument must be calibrated; this step is performed at each installation of the system in the aircraft or after any change in the system's installation, by measuring the 3D vectors of the lever arms. The pilot follows the flight plan, through the flight guidance software, and at the end of the scanning mission, the LiDAR and POS measurements are stored on memory cards or hard disks, and the measurements from the ground reference stations are stored locally.

2.6.5.4. LiDAR data processing

After the sequential processing and calibration of the data, the geocoded measurements are stored in chronological order, with the following attributes: $time_{GPS}$, X_{WGS84} , Y_{WGS84} , Z_{WGS84} , intensity.

To ensure data quality, strip adjustment should be performed based on the overlapping areas between the swaths (including the crosslines).

The standard point cloud storage format is the binary las format, defined by ASPRS (American Society for Photogrammetry and Remote Sensing). One attribute of the 3D points is the class. In version 1.4 of the las files, are presented 256 standard classes, divided in: 21 defined by ASPRS (eg Rail, Building), 42 reserved, and 203 user definable (ASPRS, 2019).

The absolute accuracy of the 3D points is influenced by the GNSS-IMU solution, the parameters of the laser scanner, the quality of the calibration, the quality of the strip adjustment and the accuracy of the coordinate transformation in the national reference system. In the subchapter I will describe the absolute planimetric accuracy, the relative and the absolute altimetric accuracies.

2.6.5.5. Final products generation

The final products that can be generated based on LiDAR 3D point clouds are 2.5D digital models (digital terrain models - DTM, digital surface models - DSM, normalized digital surface models - nDSM), contour lines, 2D vector elements (extracted: automatic, semi-automatic or digitized) and 3D models of objects. In this subchapter are being detailed the most common products: DTM, DSM, nDSM, and contour lines.

2.7. Single Photon LiDAR

SPL stands for Single Photon LiDAR. The operating principle of SPL instruments is to emit a single laser pulse and divide this beam through a diffractive optical element into multiple beamlets, to increase the density of the points. The advantages of SPL systems (Musäus, 2019) are:

- ✓ high flying heights, while maintaining the density of the points;
- ✓ bigger scanning rate than LiDAR linear systems ($> 450\text{km}^2/\text{h}$ at a density of $8\text{points}/\text{m}^2$);
- ✓ penetrates clouds and low altitude fog, allowing the extension of scanning intervals;
- ✓ the altimetric root mean square error $<10\text{cm}$, and the planimetric one $<15\text{cm}$.

2.8. Geiger-mode scanner

L3Harris Technologies Inc. developed, in 2015, the LiDAR instrument that uses GmAPD (Geiger-mode Avalanche Photo Diode), with a matrix of 32×128 detectors (SPAR 3D, 2016). The Geiger-mode scanner acquires much denser point clouds in a much shorter time and at reduced prices. Collecting data from multiple angles leads to improved vegetation penetration, elimination of shaded areas and data voids.

2.9. LiDAR data for the case study

The study area is the city of Arad, the residence of Arad County, Romania. LiDAR data are part of the LAKI II project, for which I performed quality control. One objective of this project (ANCPI, 2016) was to generate DTM and DSM by aerial scanning LiDAR with a density of $8\text{points}/\text{m}^2$ for areas with high risk of floods in Arad and Bihor counties, for an area of

approximately 10000km². They performed the flight with the RIEGL Q780 scanner, mounted on an aircraft platform. The point cloud for the city of Arad comprises over 2 billion points, with returns from 0 to 7. The point clouds were classified into: ground points (class 2), bridge points (class 10), and unclassified points (class 1). On the area of interest were generated 23 tiles of DTM, respectively DSM, with the resolution of 1m, in format geotiff.

3. Acquisition of geospatial data by aerial photogrammetry

3.1. Introduction

In this chapter, I will describe the stages of a photogrammetric workflow, from flight planning to the generation of final products. The traditional orthophoto based on DTM will be the preamble for the detailed description of the workflow for the true-orthophoto generation. The top nadir and oblique cameras will be also detailed. The focus will be set on oblique image acquisition and processing technology, given the innovative nature of this technology and the latest developments in terms of processing. The chapter ends with the presentation of the photogrammetric data set used.

3.2. Photogrammetric evolution

Based on the method of stereomodels exploitation, three techniques are distinguished, which also represent the three great epochs of photogrammetry: analog, analytical and digital.

Analog photogrammetry was used between 1900-1960 and involved the acquisition, with classic analog cameras, of images on film (format 19x19cm or 24x24cm). In Romania, this image exploitation technique was used until the advent of digital photogrammetry. Analytical photogrammetry was used between 1960-2000 and resembles analog photogrammetry, with the difference that the stereorestitution device is analytical (eg Aviolyt Wild Bc2) and uses a computer, on which the elements are drawn in vector format. The map was obtained on a plotter. The digital era began with the same analog images, but the processing was done with specialised software for digital stereorestitution. Within this era, the orientation is performed in automatic or semi-automatic mode, the exploitation of the stereomodel is interactive, and the photogrammetric products obtained are in a wide range. For some time, analog and digital cameras coexisted, simultaneously producing data for mapping purposes.

3.3. Photogrammetry principles

The basic principle of photogrammetry is triangulation. Thus, to determine the coordinates of an object in the field, images are acquired on the object from two perspectives. Currently, two digital image acquisition technologies are used: with CCD or CMOS sensors.

The classification of photogrammetric images (Upadhyay, 2013) can be done according to several criteria, including:

1. the location of the photogrammetric camera at the time of collection: aerial or terrestrial;
2. the angle at which the data is acquired from the nadir: vertical or tilted;
3. the field of view (FOV) angle of the camera: normal, wide or super-wide FOV.

Digital image capture can be done either sequentially with pushbroom cameras (eg Leica ADS40) or by single exposure (eg Leica DMC).

3.4. Aerial photogrammetric systems

3.4.1. Vertical cameras

Vertical images allow optimal observation of roofs, offer the advantages of equal illumination of the entire surveyed area and constant spatial resolution. The first commercial aerial images were acquired in England in the 1919s by the company Aerofilms, which later carried out extensive photography activities for mapping in Africa, Asia, and England (Aerial photography, 2017). Currently, there are many aerial camera companies on the photogrammetric market, such as Leica Geosystems, Vexcel Imaging, Phase One, Teledyne Optech.

3.4.2. Oblique cameras

Oblique cameras have been used in the military sector in the USA since the 1930s. Over the years there were developed several types of oblique cameras (Remondino, Toschi, Nex, & Gerke, 2018): single-camera (VisionMap), 2 cameras (IGI Dual), 3 cameras (DLR-3K), 4 cameras (Rolleimetric AIC x4), 5 cameras (Fairchild T-3A) or multiple cameras (Octoblque Midas). Oblique systems can have the following configurations:

- ✓ **Maltese-cross**: has a nadir-oriented camera and 4 tilted cameras on two perpendicular directions, which take oblique images with an inclination of 40°-50°;
- ✓ **Fan** (twin cameras - VisionMap A3 Edge);
- ✓ **Block** (4 cameras arranged in a block - IGI Quattro-DigiCAM).

3.5. Photogrammetric workflow

3.5.1. Mission planning

Based on the technical specifications underlying the aerial photogrammetric project, the suitable sensor and platform are selected, and the flight planning is performed. There are various flight planning software packages, either provided by sensor manufacturers or by independent developers (Pepe, Fregonese, & Scaioni, 2018).

3.5.2. Field measurements and image acquisition

The planned GCP (ground control points) and checkpoints are surveyed in the field, using proper equipment and methods, so that the accuracy of the points is better than the one needed for the final product. Choosing the optimal period for performing the photogrammetric flight must consider the following aspects:

- the sun angle must be greater than 30°, to limit the shadows in images;
- the land surface must not be covered by snow or floods;
- atmospheric conditions, such as fog, atmospheric veil, or dust, should be avoided.

3.5.3. Tie points extraction

This step consists of extracting the corresponding points in two or more images with overlap. Automating this step, for oblique images, is a laborious task, because of the different perspectives and lighting under which the images are taken. Tie point identification can be based on the brute force strategy, which involves comparing each descriptor of image A with each descriptor of image B. This approach is simple but quite inefficient. Examples of alternative strategies are: FLANN (Fast Library for Approximate Nearest Neighbors), which uses a quick search structure, and MatchMe, which uses similarity measurements between descriptors (Shragai, Barnea, & Even-Paz, 2012).

3.5.4. Bundle adjustment

Following the bundle adjustment, based on the GCP coordinates, the internal orientation parameters, the coordinates and the rotation angles measured for each projection center and the previously identified tie points, the 3D coordinates of the points, and the exterior orientation parameters will be determined. Both the tie points extraction and the bundle adjustment can be done with specialised software, such as Inpho Match-AT (Trimble), UltraMap (Vexcel), Leica Xpro (Hexagon), Pix4D packages (Pix4D).

3.5.5. DIM and DSM generation

After the orientation of the nadir and/or oblique images, the dense point cloud (DIM) is generated. With nadir images acquired at a GSD (ground sample distance) of 10 cm, a point cloud with a density of 100 points/m² can be generated. With oblique images, point clouds 3 times denser can be obtained (Rupnik, Nex, & Remondino, 2014).

3.5.6. Orthophoto generation

The orthophoto is the basic product of the photogrammetric workflow and represents an orthorectified image, where the perspective effects because of the acquisition and the relief have been eliminated. DTM was traditionally used to generate the orthophotos, but the use of DSM has become a standard in mapping. The accuracy of the DSM-orthophoto, also called true-orthophoto, depends on the quality of the DSM.

A well-known method for generating true-orthophotos is the Z-Buffer method, developed to solve the hidden surface problem (Catmull, 1974).

3.6. Aerial photogrammetry applications

The last step of the photogrammetric workflow is the extraction of vector data, either by 3D digitization in stereomodels or by 2D digitization on orthophotos, or by extracting the data by automatic/semi-automatic processes. These elements are subject to various photogrammetric applications, like urban planning, agriculture, emergencies responses.

3.7. Photogrammetric data for the case study

The study area is the same one, as presented in subchapter 2.9. Besides the already described objectives of the LAKI II project, we can mention the aerial photogrammetric flight and the orthophoto generation. The images were acquired with the GSD of 20 cm (ANCPI, 2016) and the aerial flight was performed with the Vexcel Ultracam Eagle Mark2 camera, mounted on an airplane. The aerial flight was performed in the East-West direction, in compliance with all meteorological requirements, when the sun angle was greater than 30°. All images were acquired in March 2017. The forward overlap was 65% and the side overlap was 35%. After running the photogrammetric workflow in UltraMap software, the RGB orthophoto was obtained with a spatial resolution of 20 cm and an absolute planimetric accuracy of $\pm 0.20\text{m}$.

4. Extracting data from LiDAR point clouds

4.1. Introduction and purpose

In this chapter, I will detail the most popular methods of point cloud segmentation. I will continue with the description of the 3D features and the methods of supervised classification of point clouds based on Machine learning algorithms. The purpose of the review of all segmentation and classification methods is to create a framework for the case study, where I will detail all the steps and results of the proposed workflow for the classification of LiDAR point clouds.

4.2. Point cloud segmentation

Point cloud segmentation is the process of classifying points into multiple homogeneous regions, based on similarity measurements of features (Nguyen & Le, 2013). The subchapter describes the segmentation methods presented by researchers in the field, among which (Sapkota, 2008), (Nguyen & Le, 2013), and (Xie, Tian, & Zhu, 2020).

4.2.1. Edge based methods

These methods comprise two stages: extracting the edges (boundaries between different regions) and grouping the points within these boundaries. The boundaries are identified based on changes in surface properties compared to an initially set threshold. One attribute used frequently is intensity.

4.2.2. Region based methods

The major advantage of this method is the robustness to noise points. A collection of *region growing* algorithms is available in the C++ code library for point cloud processing, named PCL (Rusu & Cousins, 2011).

4.2.3. Attributes based methods

Within these methods, the points are grouped based on the attributes. Thus, the first step is attributes computation, for example, surface texture measures (Filin S. , 2002). (Filin & Pfeifer, 2006) used the slope adaptive method to select the neighborhood based on the following point cloud attributes: point density, horizontal and vertical distribution, etc.

4.2.4. Model fitting methods

These methods use the idea that most artificial objects can be decomposed into primitive geometric shapes, such as planes, cylinders, spheres, and cones (Schnabel, Wahl, & Klein, 2006). Robust parameter estimation methods are the 3D Hough transformation and the RANSAC method (RANdom SAmple Consensus).

The RANSAC method (Fischler & Bolles, 1987) uses a minimum number of randomly chosen points to construct an initial solution, based on which it calculates the parameters of the initial model. This data set is grown with the points that fit in the original model. Thus, several primitive hypotheses are generated, and the choice of the optimal model is based on a voting scheme.

3D Hough transformation is based on the method of recognising 2D elements (lines and circles) in images (Hough, 1962). The transition from 2D space (images) to 3D space (point clouds) was made in several stages, and we can mention the research activity to detect arbitrary shapes (Ballard, 1991).

4.2.5. Graph based methods

In the simplest approach, graph based segmentation methods consider the point cloud as a graph, in which each point is a node, and the connections between certain pairs of neighboring points represent the edges. One approach is to create a 3D graph based on KNN method, where the neighbouring points are more closely connected (Golovinskiy & Funkhouser, 2009).

4.2.6. Hybrid methods

Hybrid methods use the combination of several segmentation methods and try to reduce the disadvantages of using one method by exploiting the advantages of another one.

4.2.7. Machine learning based methods

In the case of unsupervised learning methods that use *machine learning* algorithms, the problem consists of determining how data is organised. Some of the segmentation methods that are not based on training data are **K-means** (Zhu & Shahzad, 2014), **Fuzzy clustering** (Sampath & Shan, 2010), **Mean-shift** (Ukrainitz & Sarel, 2014), and **hierarchical clustering: SLink** (Sibson, 1973), **P-Linkage** (Lu, et al., 2016).

4.3. Point cloud classification based on Machine learning algorithms

Three approaches to the problem of semantic segmentation (or classification) are known (Grilli, Menna, & Remondino, 2017): supervised, unsupervised, and interactive. Figure 4.1 shows a workflow for the supervised classification of point clouds.

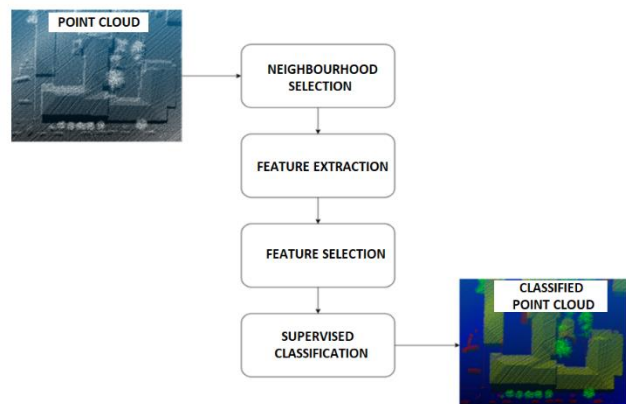


Figure 4.1 Supervised point cloud classification

4.3.1. 3D point features

4.3.1.1. PCA, eigenvalues, and eigenvectors

Principal Component Analysis (PCA) is a technique that performs the reduction of the data size by calculating the main components of the data - sets of eigenvalues and eigenvectors. The eigenvalues show the dimensions of a 3D ellipsoid along the three main axes and respect the inequality:

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0 \quad (4.1)$$

4.3.1.2. About point features

A feature is a specific representation of the data, which describes a property that allows the differentiation of certain elements within the scene (Weinmann M. , 2013). In the case of point clouds, the features can represent local or global geometric properties.

4.3.1.2.1. 3D features of LiDAR points

3D features can be structured into geometric and local-abstract features. The 3D geometric features can be point height, the local density of the points, the verticality, the maximum difference of height between the points in the neighborhood, the standard deviation of the height in the neighborhood. The best known local 3D features are the abstract eigenfeatures (Weinmann M. , 2016), computed based on the eigenvalues, like linearity, planarity, scattering.

- linearity
$$L = \frac{\lambda_1 - \lambda_2}{\lambda_1} \quad (4.2)$$

- planarity
$$P = \frac{\lambda_2 - \lambda_3}{\lambda_1} \quad (4.3)$$

4.3.1.2.2. 2D features of LiDAR points

The 2D features allow the identification of certain objects in the urban scene, which are almost perfectly vertical (building facades, pillars) and are complementary to the 3D ones.

4.3.2. Neighborhood selection

Defining the neighborhood of a 3D point, P, is the selection of points next to P, and this can be done using different strategies (Weinmann, et al., 2015), among which:

- a. **spherical/cylindrical neighborhood with fixed radius;**
- b. **k nearest neighbors in 3D:** comprising the nearest K 3D points around the P point, in terms of 3D or 2D metric distance.

Both approaches require prior knowledge of the analysed area and the definition of a constant value: either the radius of the sphere/cylinder or the number of neighbors.

4.3.3. 3D feature extraction

Most of the previously defined features are based on calculations within the defined neighborhood. For example, a point may have as attributes the following 16 computed features: $P(L, P, S, C, O, A, E, \Sigma, H, D_{3D}, V, \Delta H_{3D}, \sigma_H, D_{2D}, \Sigma_{2D}, R_{2D})$.

4.3.4. Feature selection

In selecting the optimal features to be used in the classification process, the Hughes phenomenon, also called the curse of dimensionality, must be considered. This phenomenon implies an increase in the classification's accuracy, followed by reaching a maximum and a slight decrease. This trend is also related to the number of features used in the classification process. One approach is to use, for this selection, the information gain (Weinmann, Jutzi, & Mallet, 2014), which is evaluated independently for each feature and those that get higher values are relevant and must be used in the classification process.

4.3.5. Supervised classification

Supervised classification can be done by several methods, but they all have in common the two sets:

$$\checkmark \text{ training } X = \{(x_i, l_i)\}, \quad i = 1 \dots N_x, \quad l_i \in \{1 \dots N_c\} \quad (4.4)$$

$$\checkmark \text{ testing } Y = \{y_j\}, \quad j = 1 \dots N_y \quad (4.5)$$

where,

x_i = vector of features for point i ;

l_i = corresponding class of point i ;

y_j = vector of features for point j

and the purpose of assigning a class l_j ($j = 1 \dots N_c$) to each j point in the testing data set.

Supervised classification methods can be differentiated into **standard classifiers**, which are based only on the extracted individual features, and **contextual classifiers**, which exploit both the extracted individual features and the relationships between the 3D points within the neighborhood. The most commonly used supervised classifiers are Nearest Neighbor, Decision Tree, Random Forest, AdaBoost, Support Vector Machines, and Maximum Likelihood (Waikato, 2021). Another approach is to classify point clouds using deep neural networks.

Nearest Neighbour classifier (NN) uses instance-based learning. The rule used is to compare the features of the new points with those of the points in the training set, for example, by using the Euclidean distances.

Decision Tree classifier (DT) uses rule-based learning. Based on the point features, a series of decisions are selected. Depending on the "answers" received to these "questions" a point is assigned to a certain class.

Random Forest classifier (RF) uses ensemble-based learning (Breiman L. , 2001). RF is represented by a set of binary decision trees, which based on the "answers" (true or false) at the level of each branch, partition the data set. The "root" node covers the entire predictor space, and the nodes that are no longer partitioned are leaves and are the final decisions of the classification.

AdaBoost classifier, within each iteration, creates a distribution of the training subset and applies a hypothesis with a low error in terms of the actual class. After running all iterations, the individual hypotheses must be combined into a single prediction rule (Freund & Schapire, 1997).

Support Vector Machines classifier (SVM) uses the algorithm of margins maximisation. SVM for multi-class classification can be regarded as a **one versus all** classifier (Morariu, 2007) or as a **one versus one** classifier (Chang & Lin, 2011).

Naive Bayesian is a probabilistic classifier that extracts the most probable class for each point based on Bayes' theorem and the assumption that all features are conditionally independent (John & Langley, 1995). To eliminate the problem of correlated features, the **Maximum Likelihood classifier** (ML) was developed, which is based on the hypothesis that classes can be represented by different Gaussian distributions.

4.3.6. Accuracy assessment

After classifying the testing data set, we can proceed to the last stage of the classification process, the assessment of the classification quality. This evaluation is performed using quality indicators (overall accuracy, classes recall, classes precision, etc.), based on the values extracted from the confusion matrix (Weinmann M. , 2019).

4.4. Case study - point cloud classification

4.4.1. Preparing the point cloud

Preparing the point cloud consists of selecting the training, testing, and validation areas, filtering the noise from the point cloud and RGB point clouds encoding. The division of these points into the three areas listed above and the percentage of points for each set are detailed in Figure 4.2.

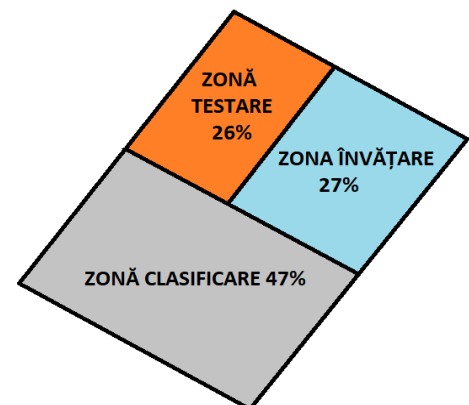


Figure 4.2 Dividing the point cloud
Blue - training data set; Orange - testing data set; Grey - area to classify

Noise filtering was done, using the **Statistical Outlier Removal** instrument, in **CloudCompare**. This filter eliminated 31444 points (2.8%). The total number of points used in this study case is 978337 (figure 4.3).

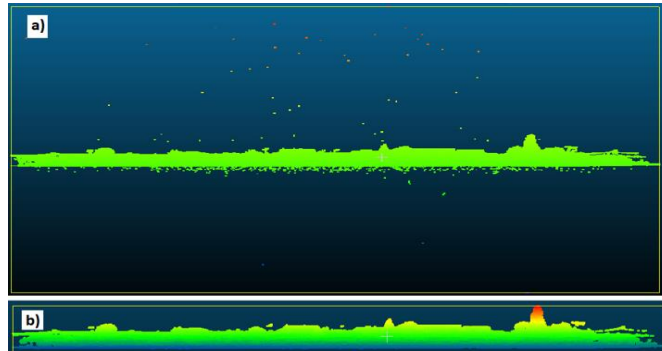


Figure 4.3 Point cloud
a) with noise; b) without noise

4.4.2. Dynamic neighborhood selection

In the literature, for the points neighborhood, fixed values are used for all the points (radius or number of close neighbors). An approach where the **K** parameter is flexible and can vary within the 3D point cloud is preferable. But, how can we select the neighborhood differently, dynamically? The answer would be to focus on a generic selection of individual neighborhoods described by an optimised scale parameter for each 3D point, thus completely avoiding the use of prior knowledge of the scene and/or of the data. In this sense, I used the **region growing** algorithm, and selected, dynamically, the nearest **K** neighbors of each point, by partitioning the point cloud into homogeneous and disjoint regions, starting from a seed point. Then, the growth of the region continued, based on similarity measurements.

Thus, I developed a code in the C++ language, based on the PCL code library. Based on several tests with various sets of segmentation parameters, I identified the optimal parameters (table 4.1). I carried this activity during the secondment at FBK (Bruno Kessler Foundation) research center, in Trento, Italy, within the VOLTA Project (GA nr. 734687 – VOLTA –H2020-MSCA-RISE-2016).

K - KDTree	50
Neighbours for Normals Computation	15
MinClusterSize	20
MaxClusterSize	100
NumberOfNeighbours	15
CurvatureThreshold	0.3
SmoothnessThreshold	$12.5/180 \cdot \pi$

Table 4.1 Segmentation parameters

4.4.3. Feature extraction

The computation of abstract eigenfeatures was implemented in a C++ code. I included also the code lines developed for the dynamic selection of K, after running and testing them separately. Besides the abstract features used in the literature, I have defined several new features. Through successive tests, I found that only the N_2 feature is relevant in the classification process.

$$N_2 = \frac{\lambda_1 + \lambda_2 + \lambda_3}{\lambda_1} \quad (4.6)$$

In figure 4.4, eigenfeatures such as linearity (a), planarity (b), surface variation (c) and N_2 (d) can be observed on the training data set.

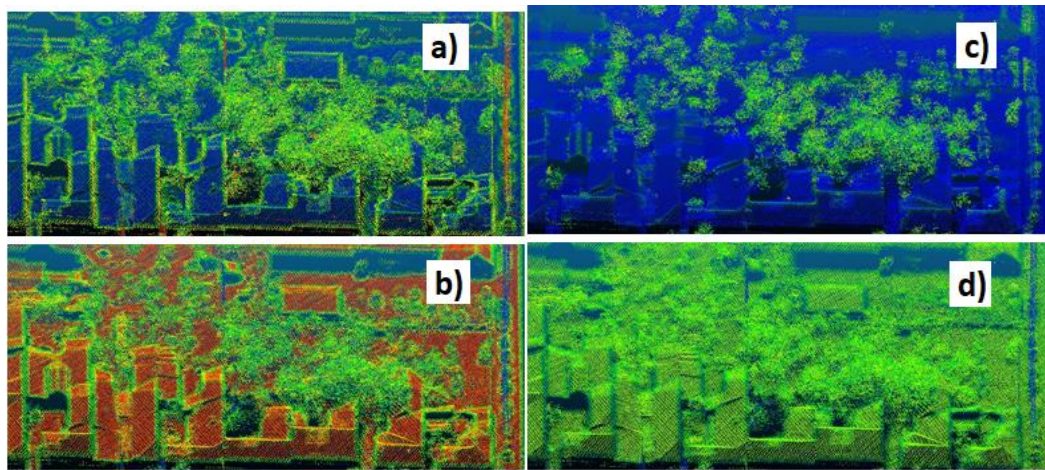


Figure 4.1 Some eigenfeatures

4.4.4. Classification and accuracy assessment

4.4.4.1. Weka

Weka is a software developed by the University of Waikato, New Zealand and contains a collection of **Machine learning** algorithms, written in Java language, for solving problems of **Data Mining** (Waikato, 2021).

4.4.4.2. Training and testing data sets

The point classes used for testing different classifiers are:

- ✓ 0 - **ground**;
- ✓ 1 - **vegetation**;
- ✓ 2 - **roof**;
- ✓ 3 - **others** (poles, power lines, cars, building facades, fences, etc.).

Based on the abstract eigenfeatures, but also on the attributes of the 3D points (including RGB values), I performed, in CloudCompare, the manual classification of the points from the training and testing data sets. First, I extracted the ground points automatically, using the **Cloth Simulation Filter** tool (Zhang, et al., 2016). From the rest of the non-ground points,

through a series of successive cuts and unions, I obtained the classified data sets. The manual classification was time consuming, the entire process took me approx. 40 hours.

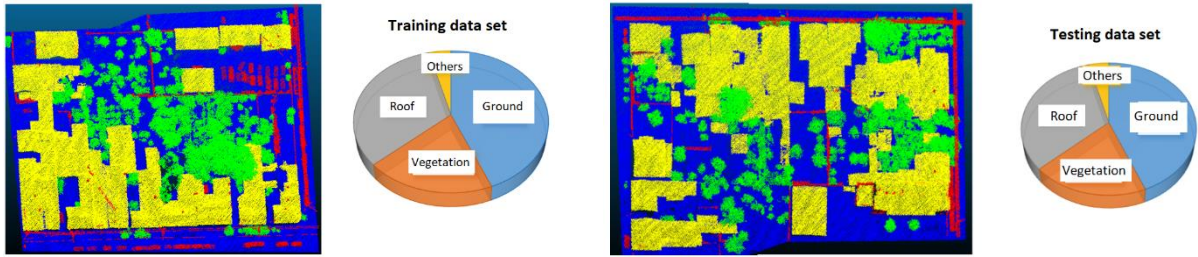


Figure 4.5 Training and testing data set

The features for all the 3D points are: translated 3D coordinates, RGB values, normal components in all three directions, curvature value, eigenvalues, abstract eigenfeatures, default point cloud features, unique ID, and class value.

4.4.4.3. Machine learning classifiers

In Machine learning the terms instance (3D point) and attribute (feature) are used. I trained 7 classifiers on 266987 instances, using the following attributes: normals (3), curvature, eigenvalues (3), abstract eigenfeatures (8), default point features (translated height, number of return and intensity), and class.

4.4.4.4. Analysis of the results for the used classifiers

From the analysis of the time required to create and test the model for the 7 classifiers run in the *Weka* software (figure 4.6), I conclude that the **IBk** (NN) classifier requires the longest time. The classifier that needs the least time is **Naive Bayes**. The **J48** classifier proved to be 6 times faster than the **Random Forest**, and the **Random Forest** and **SMO** classifiers had similar time requirements.

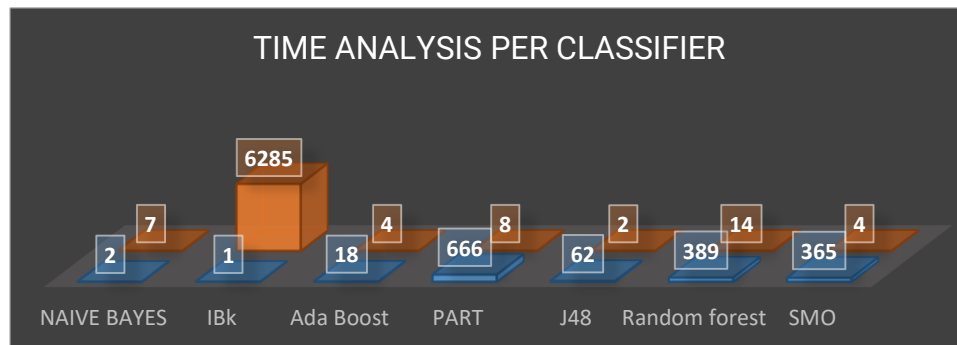


Figure 4.6 Time analysis per classifier in Weka (blue - creating the model; brown - testing the model)

In terms of classification quality (Figure 4.7), defined by the OA value (overall accuracy), the **SMO** classifier, followed by **Random Forest** and **Naive Bayes**, occupy the first three places, with values of approximately 93%. The **AdaBoost** classifier offers the lowest value of OA, only 76%.

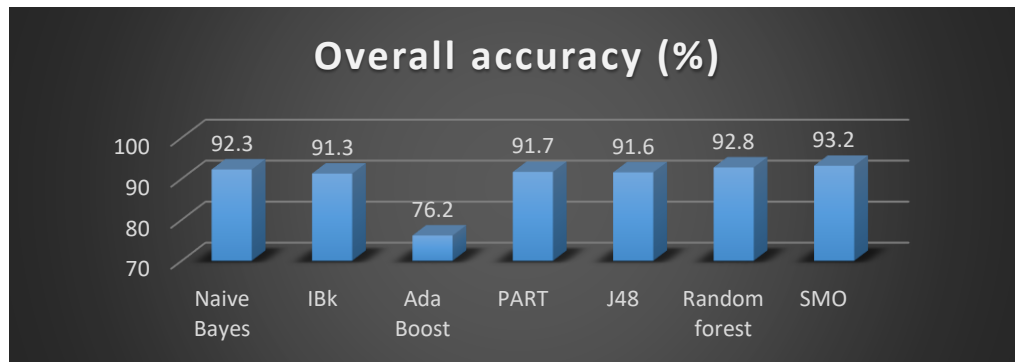


Figure 4.7 OA for the tested classifiers

From the analysis of the quality indicators for the *roof* class (figure 4.8), I conclude that the **Random Forest** and **SMO** classifiers offer the best results; the precision is about 98%, the recall is around 90%, and the F1 score is about 93%.

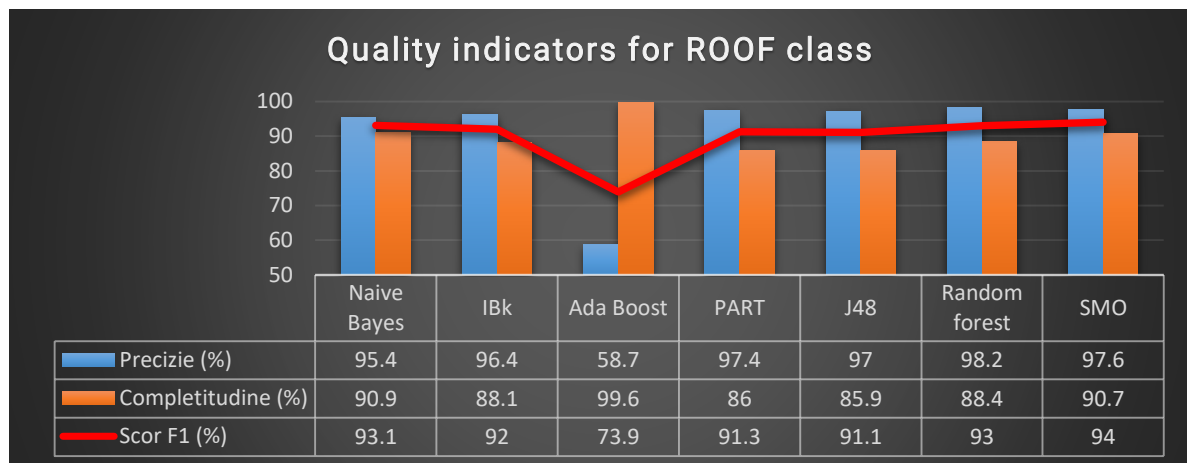


Figure 4.8 Quality indicators (blue - precision; brown - recall; red - F1 score)

By running the saved model of the Random Forest classifier, I obtained the results, displayed in figure 4.9, within 6 minutes.

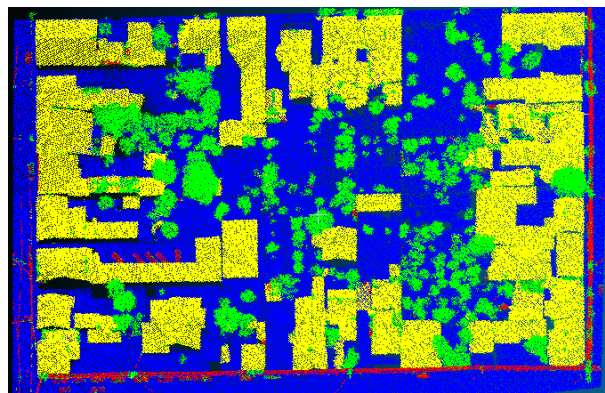


Figure 4.9 Classification results using the Random Forest classifier

4.4.4.5. Point cloud classification workflow

In figure 4.10 the workflow for classifying the point cloud using Machine Learning algorithms is schematically described.

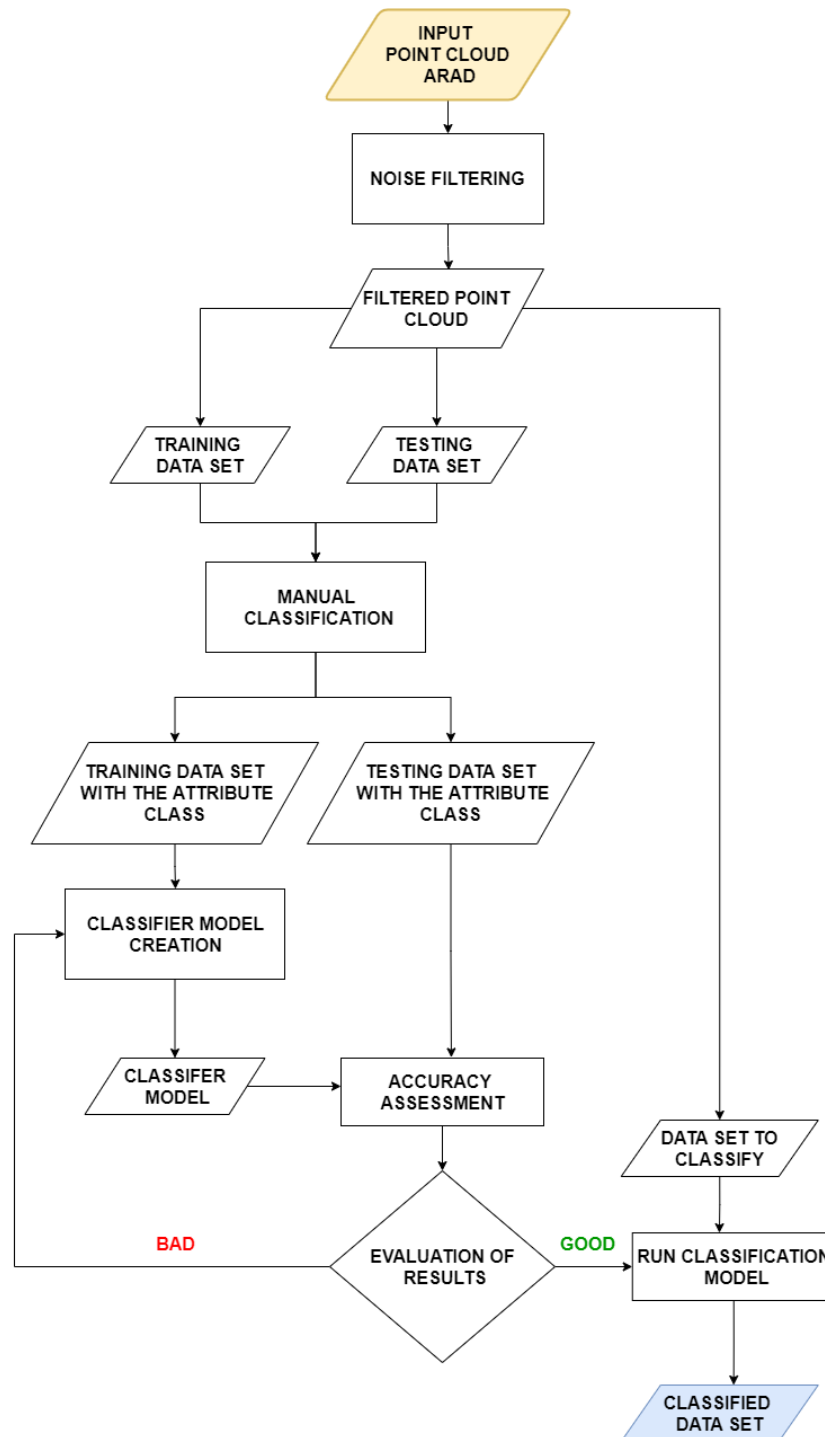


Figure 4.10 Classification workflow using Machine Learning classifiers

5. Extracting data from orthorectified images

5.1. Introduction and purpose

In this chapter, I will present the most popular methods of semantic segmentation of images. Then, I will briefly present the methods of supervised semantic segmentation of images based on Machine learning techniques and describe the most well-known artificial neural network architectures for image classification, from AlexNet to DenseNet and for semantic segmentation, such as FCN, SegNet, U-Net. In the case study, I will detail all the steps in the workflow for semantic segmentation, both with Machine Learning algorithms and with deep neural networks. I will perform the analysis on a subset of the orthophoto for the city of Arad, in ArcGIS Pro software.

5.2. Introductory notions

Artificial intelligence (AI) is the simulation of human intelligence by machines, which are programmed to think like a human. The term originated in 1956 is attributed to J. McCarthy from MIT. *Machine learning* is an application of AI, which contains algorithms that analyse data, learn about that data and apply what they have learned to make optimal decisions (Grossfeld, 2021). *Deep learning* is a sub-domain of *machine learning*, in which applications can learn and make their own decisions based on an artificial neural network (ANN).

5.3. Extracting data from images using Machine learning algorithms

Known *Machine Learning* algorithms, like **SVM**, **Random Forest**, **K-means clustering**, provide excellent results in the semantic segmentation of images.

5.4. Extracting data from images using ANN

5.4.1. Brief history of the development of neural networks

The idea of **neural networks** started from the research of W. McCullough and W. Pitts, from the University of Chicago, in 1943. Their work "A Logical Calculus of the Immanent Ideas in Nervous Activity" was the basis of the theory according to whom the activation function of a neuron is the basic unit of brain activity (McCullough & Pitts, 1943). The next step was the development of the **Perceptron** algorithm in 1958, by F. Rosenblatt, with the aim of supervised learning of binary classifiers. This field has undergone an amazing evolution, and the development of hardware and software components will certainly dictate the future of artificial neural networks.

5.4.2. ANN definitions

An **artificial neural network** (RNA) is “a distributed, massively parallel processor, composed of simple processing units with the natural ability to store new knowledge based on previous experience and make it available for later use.” (Haykin, 2008).

5.4.3. Natural neuron versus artificial neuron

5.4.3.1. Natural neuron

The human neuron is the morpho-functional unit of the nervous system and consists of the cell body and extensions (dendrites and axons). Figure 5.1 highlights the path taken by the stimulus.

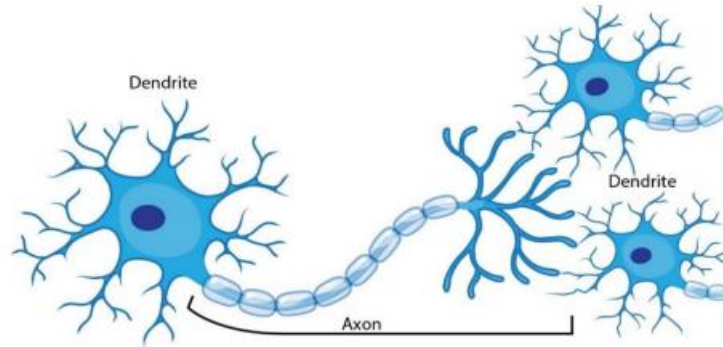


Figure 5.1 A network of human neurons
(Phillips, 2015)

5.4.3.2. Artificial neuron

An artificial neural network is a simplified model of the human neural network. Each artificial neuron comprises the following basic components (Figure 5.2):

- the **set of connections** or synapses between units, with the corresponding weights;
- the **summator**, which gathers the input signals, weighted with the synaptic power of the respective neuron;
- the **activation function**, which limits the amplitude of the output signal to a certain finite value;
- the **bias**, which is applied from outside the network and aims to increase or decrease the net input of the activation function (depending on the positive or negative value).

$$u_k = \sum_j w_{kj} \cdot x_j \quad (5.1)$$

$$y_k = \varphi(u_k + b_k) = \varphi(v_k) \quad (5.2)$$

where,

w_{kj} = the weights of neuron k ;

x_j = inputs ($j=1...m$);

u_k = the output obtained by summing the weighted input signals;

b_k = the bias of neuron k ;

v_k = the biased u_k output.

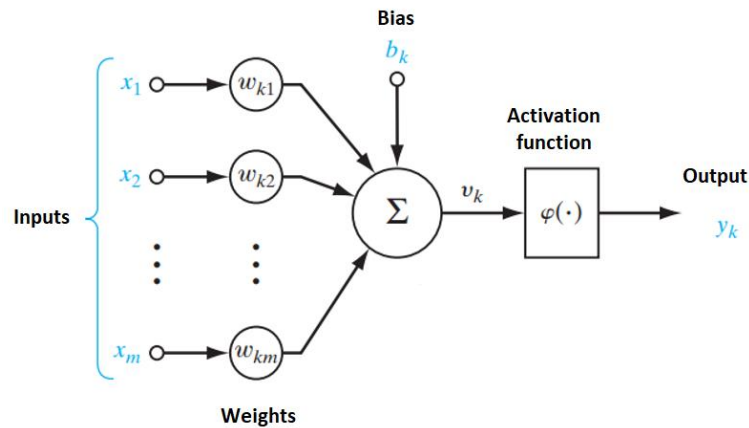


Figure 5.2 The model of neuron k
(After (Haykin, 2008))

5.4.4. ANN architectures

- A. **Single-layer feed-forward network** - causes the input layer to project directly onto the neuron output layer.
- B. **Multi-layer feed-forward network** - contains one or more hidden layers, comprising hidden units.
- C. **Feed-back network** - represents the most complex ANN architecture, which has at least one feedback loop.

5.4.5. Deep neural network

A deep neural network (DNN) is a complex neural network with multiple layers interposed between the input and output layers (figure 5.3). The activation function used is the non-linear rectified linear unit (ReLU). DNNs are used for text recognition (DeepText), facial recognition (DeepFace), etc.

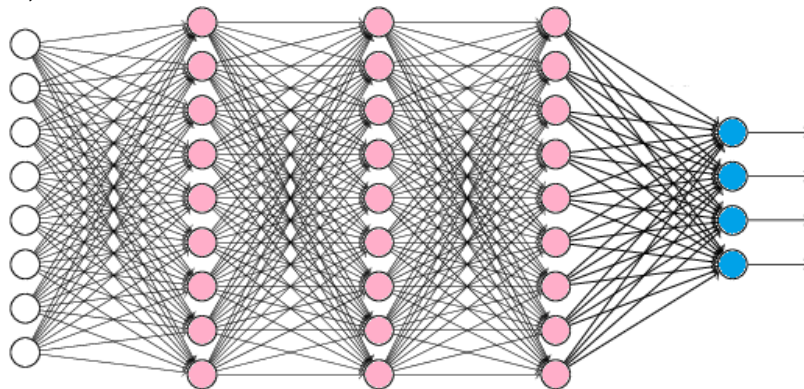


Figure 5.3 DNN with 3 hidden layers

A convolutional neural network (CNN) is a deep neural network, composed of convolutional layers, the aggregation layer, and one or more fully connected layers of the multilayer Perceptron type (Vrejoiu, 2019).

5.4.6. Learning process with ANN

Learning techniques can be classified into the following categories: supervised (figure 5.4), unsupervised, semi-supervised, or reinforced.

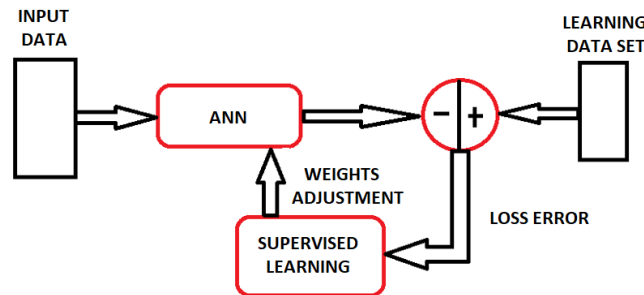


Figure 5.4 Supervised learning algorithms
(After (Yadav, Yadav, & Kumar, 2015))

5.4.7. ANN applications

The basic applications of ANN, in working with digital images, are semantic segmentation, classification, object detection, and instance segmentation (Agarwal, 2018).

5.4.8. Basic CNN architectures

5.4.8.1. AlexNet

AlexNet comprises five convolutional layers, three of them followed by max-pooling layers, and three fully connected layers. This architecture used about 60 million parameters and 650000 neurons. To make the learning process faster, unsaturated neurons and a highly efficient GPU implementation of the convolution operation were used (Krizhevsky, Sutskever, & Hinton, 2012).

5.4.8.2. VGGNet

The **VGGNet** neural network is made of 16 or 19 layers, the first is convolutional and the last 3 layers are fully connected. VGGNet is a block architecture, where each block comprises a succession of convolutional layers, followed by a max-pooling layer.

5.4.8.3. NiN Net

Based on **AlexNet** and **VGGNet** architectures, the **NiN** (Network in network) architecture was developed (Lin, Chen, & Yan, 2014). The fundamental differences between NiN and the previous architectures are the use of the *Mlpconv* layer (equivalent to a convolutional layer with a 1×1 kernel filter) instead of the linear convolutional layer and the global average-pooling layer instead of the fully connected layers.

5.4.8.4. GoogLeNet

Compared to the **AlexNet** architecture, **GoogLeNet** uses 12 times fewer parameters and provides higher accuracy results (Szegedy, et al., 2015). This architecture is based on the *Inception* blocks.

5.4.8.5. ResNet

The concept of **ResNet** (Residual Networks) architecture is based on the use of blocks that redirect input data and add them to the concept already learned from the previous layer (Residual Net, 2020).

Five ResNet architectures were created, starting with 18 layers, continuing with 34, 50, 101, and ending with 152 layers. The residual neural network with 152 layers was, in 2015, the deepest network presented at the *ImageNet* competition.

5.4.8.6. DenseNet

DenseNet is an improved version of ResNet, proposed in 2016 (Huang, Liu, Van Der Maaten, & Weinberger, 2017). The hyper-parameter used is the growth rate (k). Each layer reads the state from its preceding layer and writes it to the following layer. The growth rate adjusts the number of new information with which each layer contributes to the overall state.

5.4.9. Architectures used for semantic segmentation of images

5.4.9.1. FCN

FCN (Fully convolutional networks) are neural networks composed of convolutional layers, but which do not have fully connected final layers (Long, Shelhamer, & Darrell, 2015). The fundamental difference from CNN is that the decision layers at the end of the network are filters.

5.4.9.2. SegNet

SegNet is a deep CNN network comprising a network of encoders, a network of corresponding decoders, and the last layer of pixel-level classification.

5.4.9.3. U-net

The **U-net** architecture won the *ISBI* competition (International Symposium on Biomedical Imaging), in 2015. This architecture, based on FCN, was developed for the medical domain (Ronneberger, Fischer, & Brox, 2015).

5.4.9.4. DeconvNet

DeconvNet (Noh, Hong, & Han, 2015) uses convolution layers (which extract features from images) and deconvolution layers (generate probability maps). In the first step, the convolution network is applied and the layers, according to the VGG16 architecture, are used. In the deconvolution network, a series of unpooling layers and deconvolution ones are used.

5.5. Accuracy assessment of image classification and segmentation

The Jaccard index is the quantification of the percentage of overlap between the reference mask and the result of classification/segmentation. I have described the other quality indicators in subchapter 4.3.6.

5.6. ANN frameworks

The most popular frameworks with ANN are *TensorFlow*, *PyTorch*, *Keras*, *DL4J*, *Caffe*, *Microsoft Cognitive Toolkit*. In these frameworks, one can follow all the workflow steps for image classification or segmentation.

5.7. Case study - image semantic segmentation

I conducted the case study in *ArcGIS Pro* software, by testing the semantic segmentation capabilities of images through both Machine learning algorithms (SVM, Random Trees, and Maximum Likelihood) and deep artificial neural networks. I performed the tests on a subset of the orthophoto for the city of Arad, with the spatial resolution of 20 cm, and the spectral resolution of 32 bits.

5.7.1. Image classification using Machine learning algorithms

5.7.1.1. Supervised pixel-based classification

Based on visual image analysis and the need to extract the *buildings* class, I established the following classification scheme: *buildings*, *vegetation*, *roads*, and *bare-earth*. Using the *Training Samples Manager* tool, I collected 175 learning samples (circles with variable radius), so that their number allows a good classification, and the pixels related to the samples represent clearly and unequivocally the respective class. Initially, I ran the supervised learning process of the models without using auxiliary data, only based on the following attributes: colour, average digital number, standard deviation, number of pixels, and rectangularity. The supervised classifiers available in *ArcGIS Pro* are **Maximum Likelihood** (probabilistic classifier), **Random Trees** (ensemble classifier), and **Support Vector Machine**.

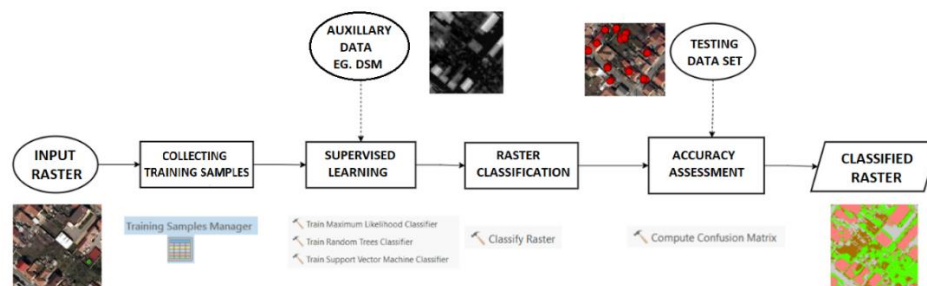


Figure 5.1 Supervised pixel-based classification in ArcGIS Pro

I used the following steps to compute the quality indicators of the classification:

- ✓ digitise the test samples and save the result in vector format;
- ✓ transform the vector file into a raster file;
- ✓ based on the raster, the desired number of test points is created (eg 5000), by applying a sampling strategy;
- ✓ the resulting feature class must be updated with the value corresponding to each point extracted from the classified raster;
- ✓ based on the test points and the classification result, the confusion matrix and quality indicators are computed.

Graph 5.6 details the quality indicators of the classification for the *buildings* class obtained for the three algorithms used.

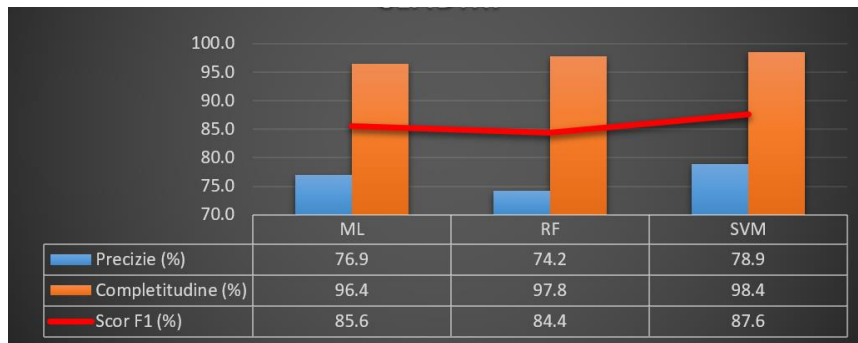


Figure 5.6 Quality indicators – buildings class
(blue - precision; brown - recall; red - F1 score)

Figure 5.7 Classified raster using SVM

I included the DSM raster as auxiliary data in the classification process. For this analysis, I applied only the SVM classifier (resulting SVM-V2 approach). This approach has led to an increase in the accuracy of the classification. The values of the quality indicators are shown in figure 5.8.

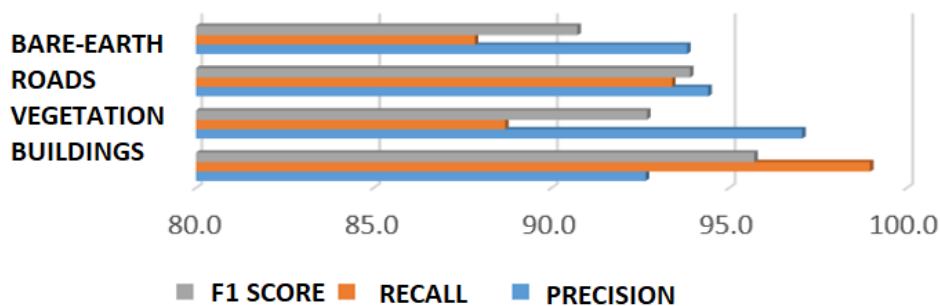


Figure 5.8 Quality indicators for SVM-V2

Comparing the two approaches used for image classification with the SVM algorithm: with or without the information from the DSM, I observed an increase in the quality of the classification by at least 10% for each indicator analysed (figure 5.9).

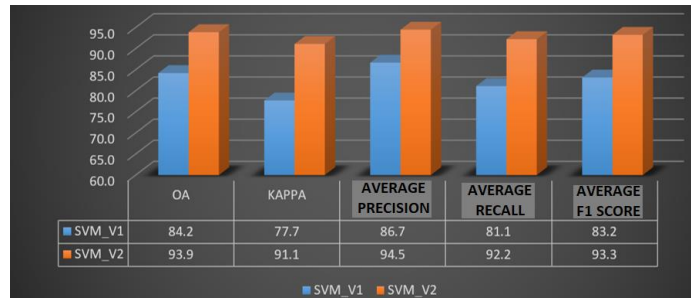


Figure 5.10 Quality indicators for image classification

5.7.1.2. Supervised object-based classification

Compared to the pixel-based classification, this approach differs by including in the workflow the following steps: image segmentation, based on the *Mean Shift* algorithm, and the selection of samples, based on the resulting segments.

5.7.2. Image classification based on DNN

After preparing the image, I created the training samples, digitizing the outline of 138 roofs. These polygon samples, were transformed into a typical ANN format, using the *Export Training Data* tool. The result was composed of 52 images, each image having 3 spectral bands and the size of 256x256 pixels. ANN extracted 556 features, with an average of 10.69 features/image. The next step was to train the model. Based on the learning data set, *ArcGIS Pro* created the network model, using the **U-Net** network type with the **ResNet34** structural model. I applied the model on the test data set and got, for the only class created - BUILDINGS, the precision of 90.4%, the recall of 62.7%, and the F1 score of 73.8%. The last step of the workflow is the pixel-based classification of the entire image.

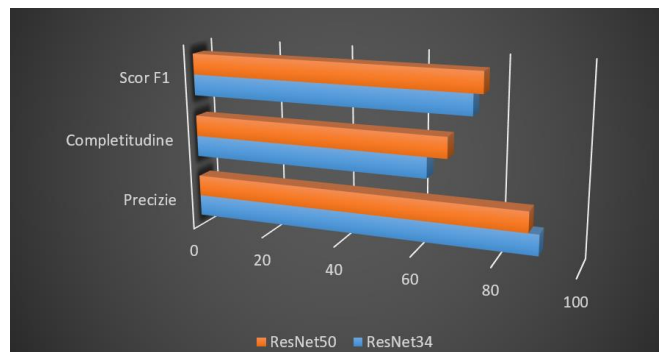
Figure 5.2 Semantic segmentation result
ResNet50

Figure 5.11 Quality indicators using ResNet34 and

I resumed the ANN training using the **U-Net** network type with the **ResNet50** structural model. The time required to create the model was 15 hours. I got, for the BUILDINGS class, the precision of 87.9%, the recall of 67.8% and the F1 score of 76.2%. I conclude that the time required to create the model using ResNet50 is too long, and the quality indicators did not experience a significant increase.

6. 3D Building Modelling based on point clouds

6.1. Introduction and purpose

Local and central administrations need a virtual representation of their cities. The elements in the 3D city models that I will emphasise in this chapter are represented by buildings. DSM is usually used for 3D building models. Unfortunately, these models do not contain semantic or geometric information and do not allow the differentiation between elements and the realisation of analyses, simulations, etc. 3D models in which each building is individually represented can be reconstructed from photogrammetric images, point clouds (either LiDAR or photogrammetric derived), ground footprints or a combination of these types of data.

6.2. 3D building models generation

3D city models comprise 3D models of buildings, vegetation, street elements, urban infrastructure (eg hydrographic constructions, high voltage lines), and urban objects (eg traffic signs, monuments, statues, fountains). DSM at higher resolutions include all these elements, but do not distinguish between individual objects. Therefore, generating 3D models from point clouds is the preferred option, as it preserves the accuracy and facilitates analysis.

6.2.1. Model-based generation of 3D building models

In this situation, a building is considered to be composed of simple primitives. These primitives are stored in a library of pre-defined models. With complex shaped buildings, the solution for a reconstruction close to reality is to decompose the building into multiple shapes and match the models to these shapes.

6.2.2. Data-based generation of 3D building models

Using this approach, no assumptions are made about the shape of the buildings, thus eliminating the problems encountered with model-based methods. Data-based methods use point cloud segmentation, followed by individual segment assembly and model reconstruction. The methods used for segmentation have been detailed in subchapter 4.2. and include algorithms such as region growing, RANSAC.

(Dorninger & Pfeifer, 2008) proposed the use of the α Shape algorithm (Edelsbrunner, Kirkpatrick, & Seidel, 1983) for generating the buildings footprints. An α Shape 2D is a polygonal boundary of a set of points (figure 6.1).

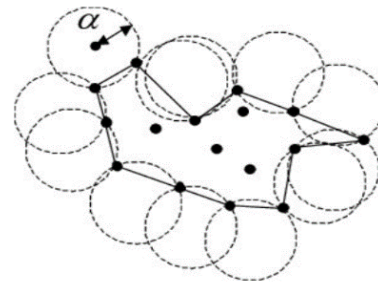


Figure 6.1 α Shape 2D
(Ohbuchi & Takei, 2003)

6.2.3. Point cloud processing softwares

There are various point cloud processing softwares, which can be divided into:

- provided by LiDAR system manufacturers, eg RiPROCESS developed by Riegl (RIEGL, RiProcess_Datasheet, 2021);
- academic or open-source: LAS tools (Hug, Krzystek, & Fuchs, 2004), CloudCompare (Girardeau-Montaut, 2019);
- commercial software for performing the entire workflow, eg TerraScan (Terrasolid, 2021).

OPALS (Orientation and Processing of Airborne Laser Scanning) is also part of the group of academic software. This modular program system was developed within the Research Groups Photogrammetry and Remote Sensing, Technical University of Vienna (OPALS, 2019a). The modules available in OPALS allow both LiDAR data processing, raw POS (direct georeferencing, strip adjustment, calibration), as well as classification of point clouds, geomorphological modelling and generation of DTM and DSM (Pfeifer, Mandlbauer, Otepka, & Karel, 2014).

6.3. CityGML level of detail

The geometric and semantic detailing of a 3D model is known as the level of detail (LoD). Standardisation defined by the Open Geospatial Consortium (OGC) is known as the CityGML standard. CityGML is an open data model, used for storing and exchanging virtual 3D city models, based on the ISO 19100 international standards. The OGC adopted the standard CityGML, version 2.0, in March 2012 and it is still in use. Version 3.0 is in progress since 2013 and has not been released yet.

6.3.1. CityGML 2.0

In the CityGML 2.0 are defined 5 standard classes with different LoDs (figure 6.2). Higher levels have a higher degree of detail, are more precise and have a structure of greater complexity.



Figure 6.2 LoDs in CityGML 2.0
(Biljecki, Ledoux, & Stoter, 2016)

In the CityGML standard, objects present semantic and geometric definition models. The **semantic model** is based on the ISO 19109 and the GML3 standard and contains the definition of classes. Using semantics differentiates typical 3D models from standardised CityGML models. At the semantic level, features represent real-world objects, such as buildings, walls, windows. The **geometric model** is based on the ISO 19107 standard and represents the spatial properties of the geometry of 3D objects. If a 3D model has both types of hierarchies, then these two must be coherent and in correspondence.

6.3.2. CityGML 3.0

CityGML 3.0 will bring major improvements to previous versions, by splitting the building models into floors, removing LoD4 and including interior, and exterior specifications for all levels of detail.

6.4. Applications of 3D building models

3D city models, through their basic component - 3D building models, cover a wide range of applications. The most known applications are estimating the potential of solar energy, estimating the energy required for individual buildings, creating the 3D cadastre, visibility analysis, prompt response to emergencies, and studying the impact of floods, earthquakes.

6.5. The current international status in 3D building modelling

The website <https://3dbag.nl/> contains sets of 3D building models for all the cities in the Netherlands. The buildings are available at various levels of detail and are regularly updated.

The 3D city models for Singapore were launched as open data in 2019. The buildings footprints available in OpenStreetMap were extruded based on information on the number of floors provided by the HDB (Housing and Development Council).

Switzerland is another country that has moved to 3D topographic maps, with total coverage of the territory with 3D objects.

6.6. Softwares for the generation of 3D building models

6.6.1. BuildingReconstruction

The software *BuildingReconstruction*, developed by *virtualcitySYSTEMS* (2018) generates digital building models with LoD1 or LoD2. For each 2D building polygon, a separate 3D model is produced. The building volume corresponds to the 2D geometry of the footprint and the height of the building extracted from the DSM.

6.6.2. Esri CityEngine

The software *CityEngine*, developed by *Esri*, uses a 3D generation procedure based on a series of rule files, written in a unique programming language called CGA (Computer Generated Architecture). These rules underlie the extrusion of polygons and the generation of 3D building models (Figure 6.3). Textures for buildings can be added to the rules file.

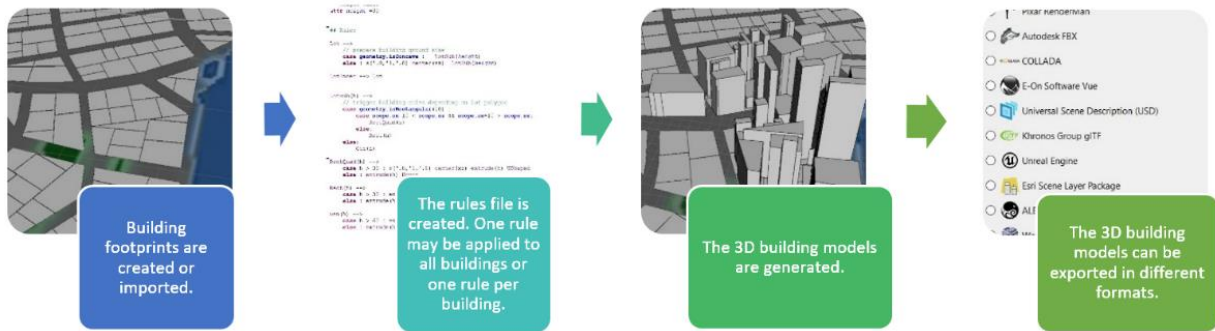


Figure 6.3 3D building models using CityEngine

6.6.3. 3dfier

3dfier is an open-source tool for creating 3D models, developed at Delft University of Technology, Netherlands. 3D building models are generated based on the 2D polygons and the point cloud related to the area (TUDelft, 2021). The buildings footprints are extruded at the height given by the 3D point cloud (figure 6.4).

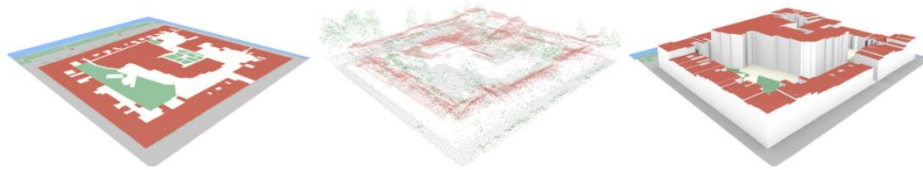


Figure 6.4 3D building models using 3dfier (TUDelft, 2021)

6.6.4. PolyFit

The *PolyFit* software performs surface reconstruction based on assumption and selection strategies, generating 3D building models with as few planar surfaces as possible.

6.7. Geometric and semantic errors in CityGML data sets

6.7.1. Geometric and semantic types of errors

CityGML uses geometric primitives to represent objects according to the ISO 19107 standard: a 0D primitive is a point, a 1D primitive is a linear ring, a 2D primitive is a polygon, and a 3D primitive is a solid (Biljecki F. , et al., 2016). Geometric primitives with the same dimensionality can be combined to form aggregates or composites. They can be structured in four classes: multi-surface, composite surface, shell, composite solid.

Each surface can store attributes, and this is recommended to facilitate further analysis. To highlight and correct semantic errors, classes can be symbolised differently and thus errors can be easily noticed on a visual inspection.

6.7.2. Topology validation software

The *val3dity* open-source software allows 3D primitive validation under the international standard, ISO 19107 - *Geographic information - Spatial scheme*, using the command line. GML, CityJSON, CityGML, OBJ files can be used as input data. The validation process is hierarchical, so that the low-dimension primitives are validated first.

6.8. 3D GIS and BIM

Regarding the life cycle of a building, the first step should be to generate a BIM (Building Information Models). The IFC (Industry Foundation Classes) standard represents all the components of a building (eg *IfcDoorType* - door type). Therefore, constructions built based on a BIM no longer need a 3D model based on the elements measured in the field, but only to perform the semantic mapping and geometric-semantic conversion from the IFC standard to the GML. Some softwares allow this conversion, like *Building Information Modelserver* (BIMserver, 2021), *IFCExplorer* (SECOM, 2020), *FME* (SAFE, 2021).

6.9. Case study – 3D building models generation

The first step in generating 3D building models is to extract buildings footprints from point clouds in OPALS software. Subsequently, I applied two approaches to determine the building's heights, directly from the point cloud, or based on information extracted from the nDSM.

6.9.1. Extraction of buildings footprints from LiDAR point clouds

6.9.1.1. Generation of buildings footprints from segmented point clouds

For the case study, I used only the class of *roof* from the classified point cloud. The first step, in OPALS, was to transform the point cloud from las to odm format.

6.9.1.1.1. Conditional clustering

Based on the examples available on the OPALS website, <https://opals.geo.tuwien.ac.at/>, I created a Python code and applied two different grouping criteria for segmentation based on conditional clustering.

Conditional clustering - v1

The homogeneity criterion is given by the z-coordinate differences of the neighboring points. If this difference is less than 10cm, then the points belong to the same segment.

Following the segmentation process, I got 44 segments. For each segment, the software displays the plan parameters and the sigma0 value. By applying the alpha shape algorithm, 57 polygons resulted. Some polygons contain several buildings, but most of them have been extracted correctly.

Conditional clustering - v2

Here, the angles between the normal vectors gives the homogeneity criterion at the neighboring points. For the angle, I used the value of 10°. Following the segmentation process, 64 segments resulted. By applying the alpha shape algorithm, 71 polygons resulted.

6.9.1.1.2. Plane extraction

Based on the examples available on the OPALS website, <https://opals.geo.tuwien.ac.at/>, I created a Python code to extract the buildings footprints. After performing several tests, I used the following parameters for segmentation:

- ✓ searchRadius = 1;
- ✓ minSegSize = 50;
- ✓ maxDist = 0.25;
- ✓ maxSigma = 0.15;
- ✓ seedCalculator = "NormalSigma0 < 0.1 ? NormalSigma0 : invalid";
- ✓ alphaRadius = 1.

After performing the segmentation and the alpha shape algorithm, I got 210 segments and 222 polygons. From the results obtained, I can state that the segmentation that groups points belonging to a single roof is performed by the method of plane extraction. The disadvantage of this method is the lack of polygons on the buildings ridges. Figure 6.5 shows the comparative analysis of the tested segmentation methods. Regardless of the method chosen, GIS operations for polygons regularisation must be applied.

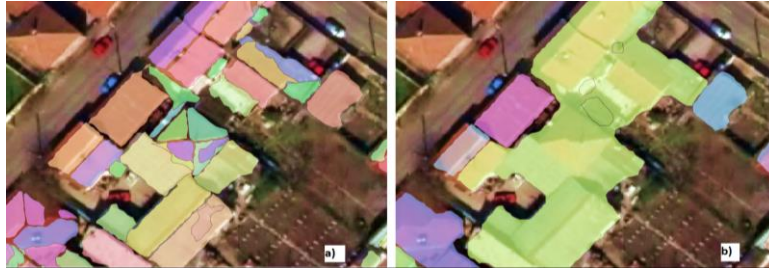


Figure 6.5 Building outlines based on segmentation
a) plane extraction; b) conditional clustering – v2

Following the analysis of the results, I concluded that the method of plane extraction provides the best results and will be used in the workflow.

6.9.1.2. Post-processing of segmented polygons

Buildings footprints generated by segmentation using plane extraction method require simplification and regularisation operations in a GIS environment, in order to use them as a basis for 3D building models. The proposed workflow for regularising polygons was applied in *ArcGIS Pro*. I detailed the instruments used in Figure 6.6.

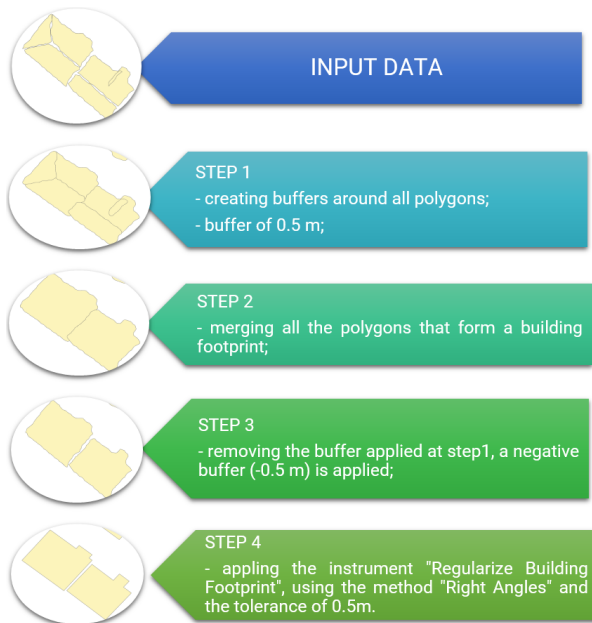


Figure 6.6 Regularization process applied



Figure 6.7 Buildings footprints
a) segmentation results ; b) postprocessing results

After GIS processing, I performed the filtering of buildings with areas smaller than 20m^2 . Finally, for the data set used, only 53 polygons remained, representing the buildings footprints. The data set must be checked and corrected for topological errors such as overlaps or gaps. I ran the topology and identified 24 errors, which I corrected.

From the analysis of the data set, I can conclude that over 90% of the buildings have correct footprints. The difference of 10% of non-compliant footprints is due to errors in the classification or/and in the segmentation of the point cloud.

6.9.2. Generating 3D building models in 3dfier

To run the generation in *3dfier* software, it is necessary to prepare the working environment (figure 6.8). As described in the previous subchapter, the buildings footprints were regularised and corrected for all topological errors before being introduced into the 3D model generation software. The point cloud used consists only of the *ground* and *roof* point classes. I updated the model configuration file with the corresponding information, based on the data set. The software can be run from the Command Prompt. After navigating to the location where *3dfier* is saved, I executed the command `.. | 3dfier.exe testarea_config.yml -OBJ output | ARAD.obj`.

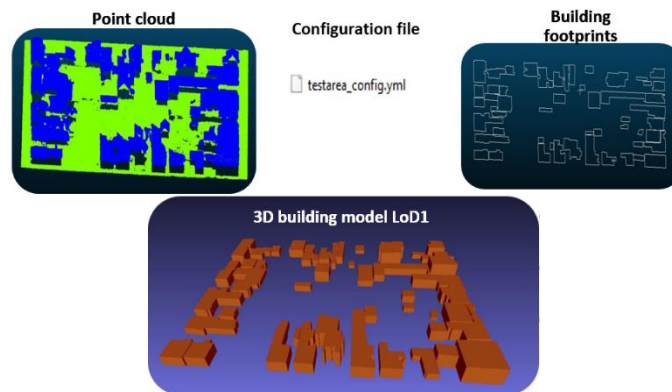


Figure 6.8 Input data for 3dfier

ARAD.obj consists of 1030 vertices and 1976 faces. For viewing, the 3D building models with LoD1, I uploaded the file in MeshLab software (figure 6.9).

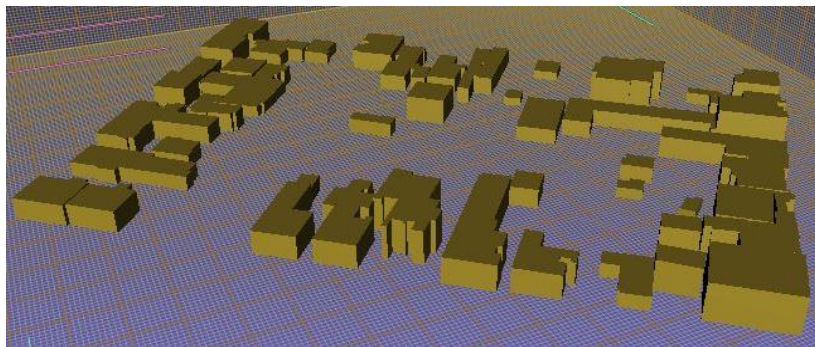


Figure 6.9 3D building models generated in 3dfier

6.9.3. Generating 3D building models in CityEngine

This approach requires information about the height of each building. I chose to use these heights in the form of attributes of the buildings footprints. Thus, based on the previously generated buildings footprints, the height extracted from nDSM, and the CGA rules, 3D building models with LoD1 or LoD2 were generated.

6.9.3.1. Buildings height extraction using centroid points

I generated, in *Global Mapper*, on the area of interest, the following models: DTM - from *ground* points, DSM - from *ground* and *roof* points, and nDSM - based on the difference between DSM and DTM. I generated these models with the spatial resolution of 1m, using the bilinear interpolation method.

I performed the transfer of the values from nDSM to the polygons representing the buildings footprints in *ArcGIS Pro*, following the steps below:

- ✓ generating centroid points of buildings (figure 6.10);
- ✓ extracting the pixel height value, corresponding to the centroid point, from nDSM;
- ✓ creating the Height field (type Numeric - Double);
- ✓ creating a spatial join between the centroid points and the buildings footprints ;
- ✓ attribute transfer and population of the Height field.

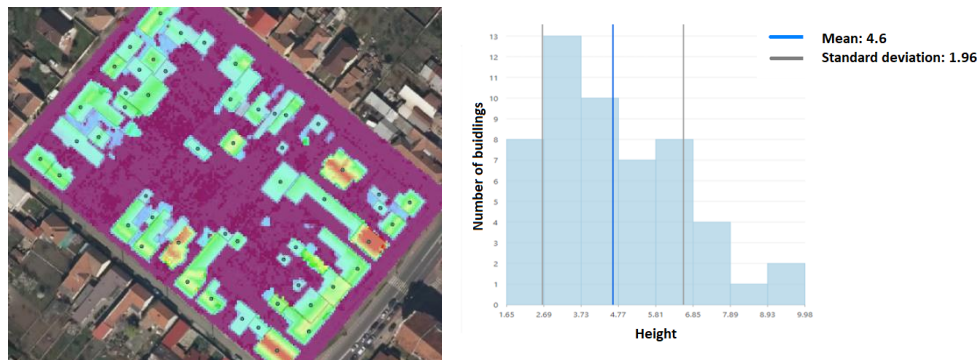


Figure 6.10 Centroid points and statistical data

From figure 6.10, we can observe that the average height of the buildings is 4.6 meters and that most buildings have heights below 7 meters.

In CityEngine, I defined the CGA rules: the type of roof (gable roof), the inclination angle (22.5°), and the texture used for building facades. Analysing the heights of the buildings in the models and in the classified point clouds, I observed big differences for some buildings whose roofs are very steep (figure 6.11).

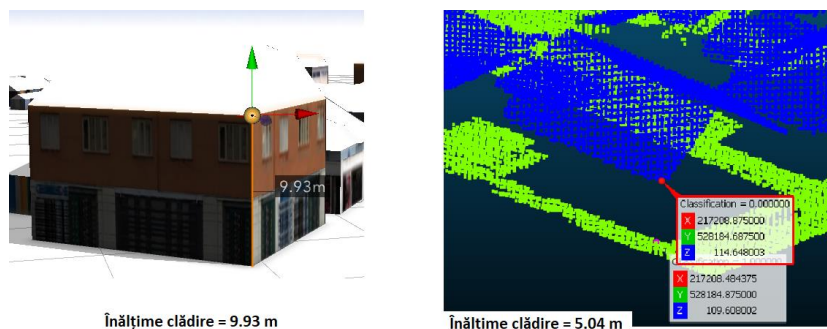


Figure 6.11 Differences between the 3D models and the point clouds

These errors are because by the way I extracted the height of the buildings from the nDSM, namely using the buildings centroids, which in most cases, are on the ridge of the roofs, at the highest points.

6.9.3.2. Buildings height extraction using characteristic points

To avoid the above problem, I applied another building height extraction workflow:

- ✓ applying a negative buffer (-1m) to the building polygons;
- ✓ transforming the contour of the building polygons into a line type;
- ✓ generating points on each contour of the buildings (figure 6.12);
- ✓ extracting the pixel height value, corresponding to the characteristic point, from nDSM;
- ✓ creating the Height field (type Numeric - Double);
- ✓ creating a spatial join between the characteristic points and the buildings footprints ;
- ✓ attribute transfer and population of the Height field.

By applying this new workflow, I moved from the centroid points of the buildings to the characteristic points on the roofs, much closer to the lower edges of the roof.

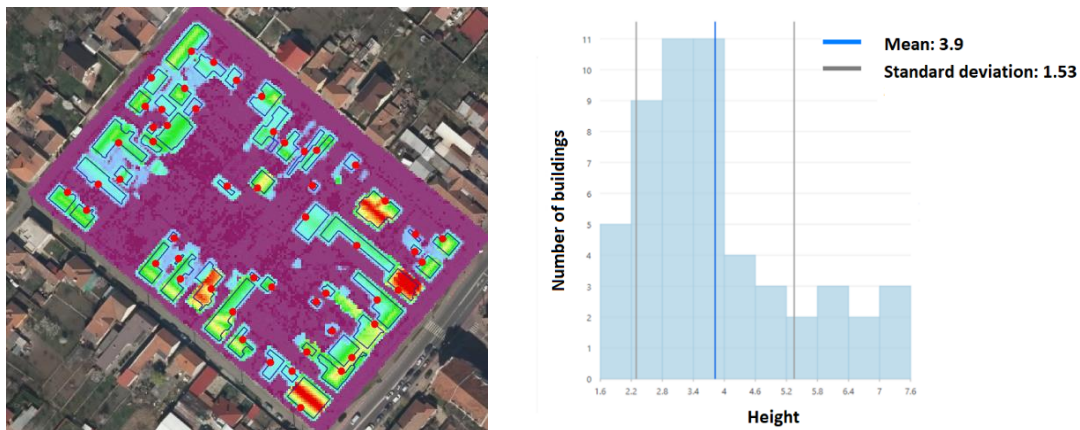


Figure 6.12 Characteristic points and statistical data

This alternative approach leads to the generation of realistic 3D models with improved building heights (figure 6.13). Figure 6.12 shows that the tallest building is 7.6 meters and that the average height of the buildings is 3.9 meters, compared to 4.6 meters in the previous approach.

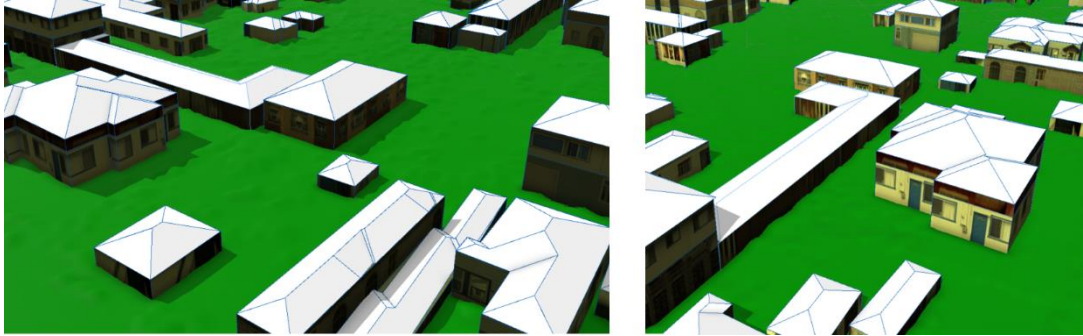


Figure 6.13 3D optimised building model - details

6.9.4. Accuracy assessment of the 3D building models

The 3D building models resulting in *3dfier* have the level of detail LoD1, and the ones generated in *CityEngine* have the level of detail LoD2. Therefore, for the analysis of the quality of the generated models, I used the 3D models generated by using the optimised heights determined from the characteristic points of the buildings in CityEngine.

To evaluate the quality of the 3D models, I analysed the following indicators:

A. planimetric root mean square error

I chose 17 points randomly distributed, located both at the base of the building and in the upper part of it. I compared the planimetric coordinates of these points with the corresponding (closest) points in the point cloud. The resulting planimetric RMSE is 0.38m.

B. altimetric root mean square error

I chose 22 points randomly distributed, located in the upper part of the roofs. I compared the altimetric coordinates of these points (building height) with the pixel values in the nDSM. It should be noted that all roofs were generated with the same angle of inclination. Therefore, block roofs or roofs with slopes in other directions can generate errors in the RMSE computation. From the data analysis, I observed big differences for 6 points that were further analysed.

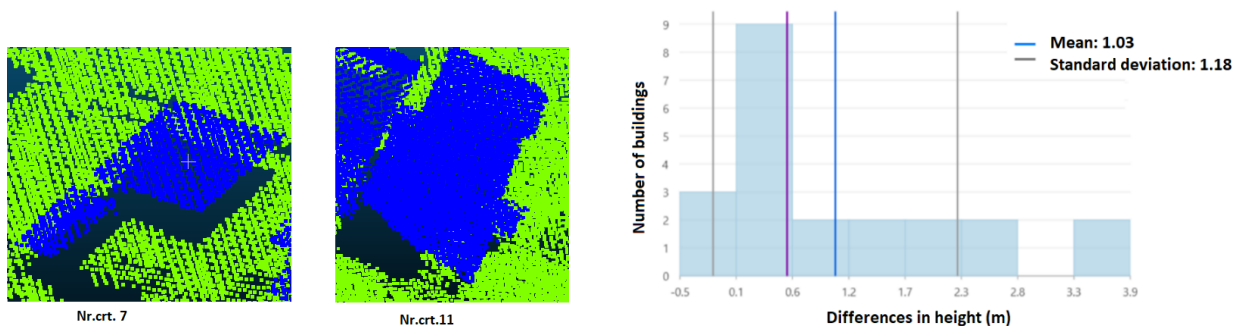


Figure 6.14 Visual and statistical analysis

The situations presented in figure 6.14 show that the large differences identified are due to the block-type roofs (No.crt.7) and the roofs with slopes in other directions (No.crt.11). Eliminating these situations, I obtained an altimetric RMSE of 0.63m.

The 3D RMSE is calculated based on the three RMSE of the coordinates.

$$\sigma_{3D} = \sqrt{\sigma_X^2 + \sigma_Y^2 + \sigma_H^2} = 0.73m$$

From the literature, the 3D accuracy of the points in the 3D building models with LoD2 is 2 meters (Jokela, 2016), therefore, it can be said that the 3D RMSE obtained on the analysed data set (0.73 meters) falls within the standards in the field.

Considering all the steps taken, and the results obtained in Chapters 4 and 6, the workflow proposed from the unclassified point clouds to the 3D building models is displayed in Figure 6.15.

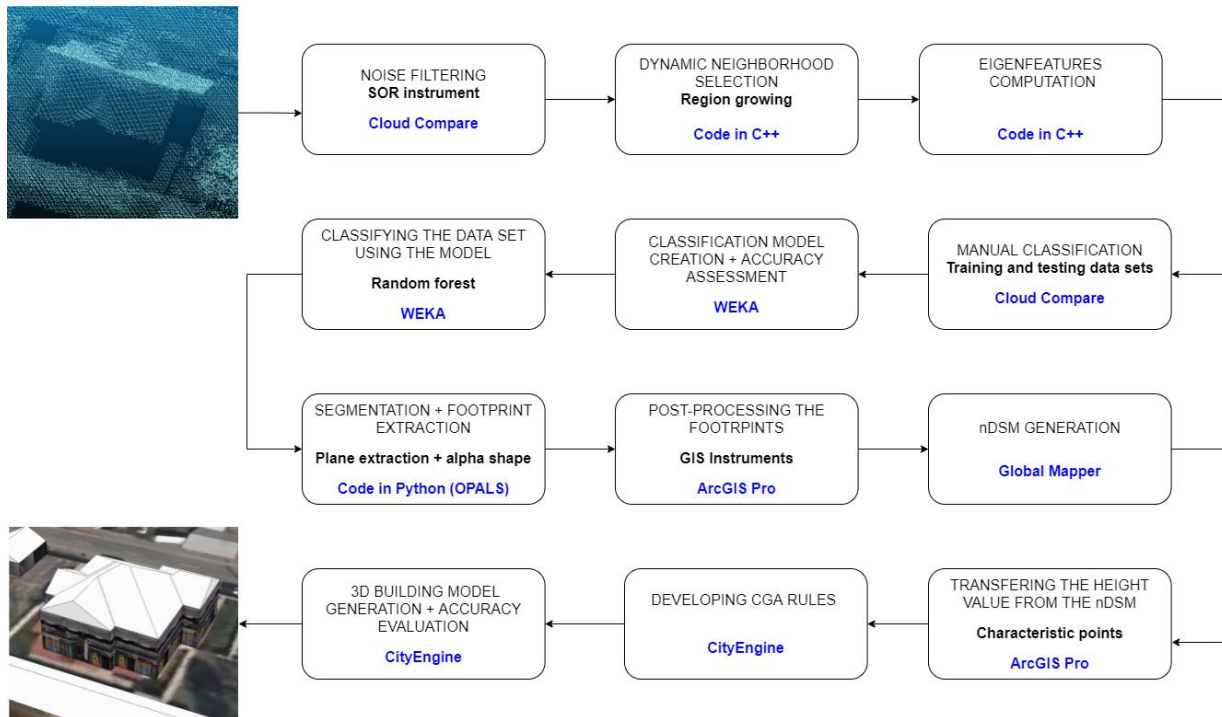


Figure 6.15 Proposed workflow for 3D building models generation

7. Generation and online distribution of 3D building models for the city of Arad

7.1. Introduction and purposes

3D building modelling for an entire city can provide useful data for the municipality. The workflow presented in Chapter 6 can lead to a correct and consistent generation of 3D models. To exemplify modelling at the city scale, I used the national database TopRo5 and data related to the building's height extracted from the LAKI II project.

7.2. 3D building models for the city of Arad

7.2.1. Input data

As input data for the 3D building modelling for the city of Arad, I used:

- ✓ the feature class BUILDINGS from the TopRo5 database;
- ✓ DTM and DSM from the LAKI II Project.

TopRo5 is Romania's reference topographic map in digital format, corresponding to the scale 1: 5000. It was populated with data and is constantly updated within the National Center of Cartography, currently being at version 6 (CNC, 2020a). The data is mostly digitized on the orthophotos. Additionally, external sources are used, such as the buildings contours from the *eTerra* application (ANCPI, 2021), vector data output from other projects in Romania.

7.2.2. Data set processing

The first filtering of the elements extracted from TopRo5 was the elimination of buildings with areas smaller than 20m², resulting in 50390 building polygons.

Based on DMS and DTM, I generated, in the Global Mapper software, the nDSM, using the tool *Combine/Compare Terrain Layers*. All 23 nDSM models contain above-ground points and were exported at a spatial resolution of 1m (Figure 7.1).

Even if there is a time difference between the photogrammetric data acquisition (2016) and the LiDAR acquisition (2017), over 90% of the buildings do not show changes over time that may affect the 3D modelling.

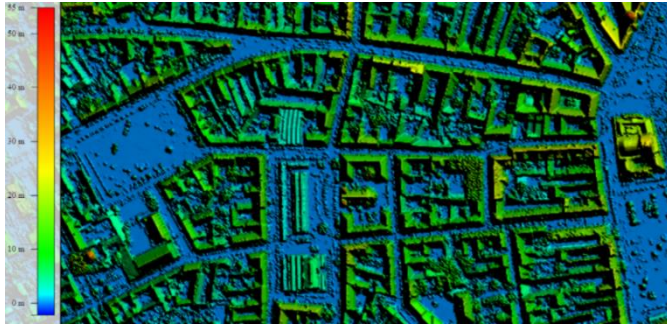


Figure 7.1 Detailed view of a nDSM tile

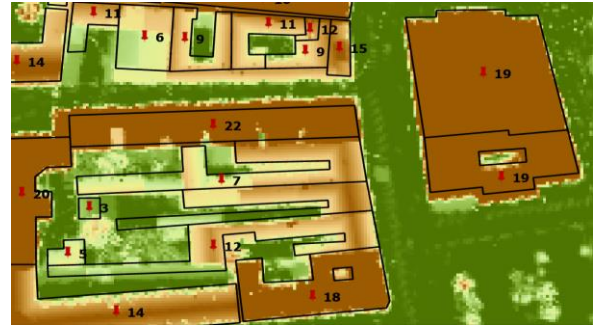


Figure 7.2 Centroid building points with the attribute height

For all the buildings polygons, I generated centroid points, to which I transferred the height from the normalized model of the surfaces (figure 7.2). Following a brief analysis of the values transferred from nDSM, I found that some centroid points have a height value equal to 0m and decided to do a thorough check at the tile level. From the analyses performed, heights of 0 meters were found for 1119 centroid points. Therefore, these buildings were either demolished, or their digitization was not done exactly with the real situation in the field (figure 7.4). 6669 buildings have heights smaller than 2 meters and I considered them annex buildings and removed them from future analyses (figure 7.3)

Figure 7.3 Statistical analysis of building heights

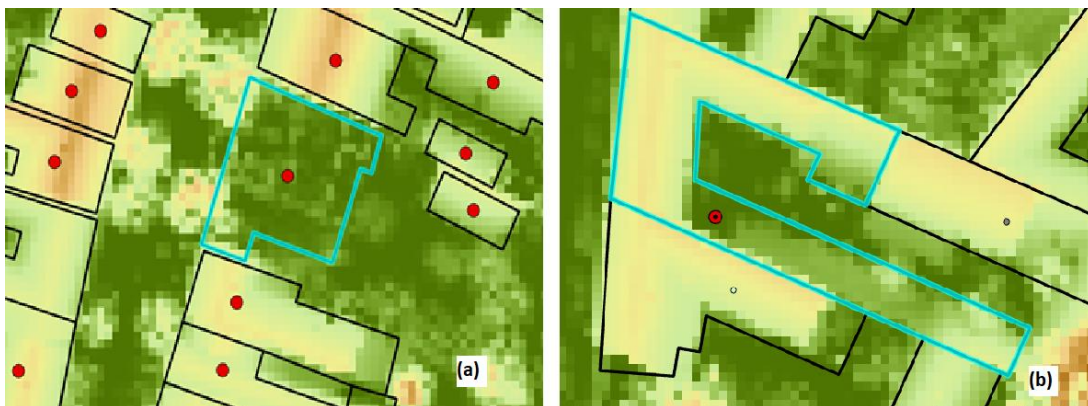
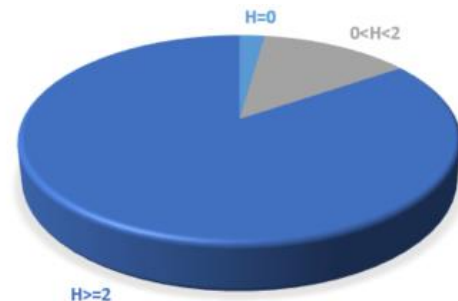


Figure 7.4 buildings with height equal to 0 m
(a) demolished building; (b) improper digitized building

7.2.3. 3D building models generation in CityEngine

Based on the buildings footprints, with the corresponding height attribute, the DTM, and the CGA rules, I generated, in CityEngine, the 3D building models for the city of Arad. Starting with the 42602 **2D buildings polygons**, after applying the modelling process 525613 **3D faces** resulted (figures 7.5). By setting the roof angle to 0°, I generated block-type buildings with flat roofs for all buildings in Arad. It should be noted that churches and other buildings with complex shapes were not treated differently from the rest, their 3D models will not be in line with the real 3D shape.



Figure 7.5 Overview - 3D building models for Arad

7.2.4. Quality evaluation of the 3D models generated for Arad

I chose to conduct a comparative study between the 3D models generated in Chapter 6 and the models obtained in this chapter. The analysis of the distances from 3D models to the point cloud (only the *roof* class) was performed in CloudCompare, using the *Cloud/Mesh Distances* tool. Following the evaluation, statistical data were obtained, based on the approximate distances for 147565 points (table 7.1). Thus, the standard deviation in the case of models generated based on footprints from TopRo5 is 1.57m, compared to 1.08m for the situation of models derived only from LiDAR data.

3D building models	Average distance (m)	Standard deviation (m)
Derived from LiDAR data	0.26	1.08
Generated based on footprints from TopRo5	0.42	1.57

Table 7.1 Quality analysis

The two sets of 3D building models are also differentiated by the number of polygonal faces resulting from the generation: 1187 faces - model from TopRo5 and 3084 faces - model generated from LiDAR data (figure 7.6).

Based on the analyses presented above, I can conclude that the models generated only from LiDAR data, through the workflow proposed in Chapter 6, provide superior quality results.

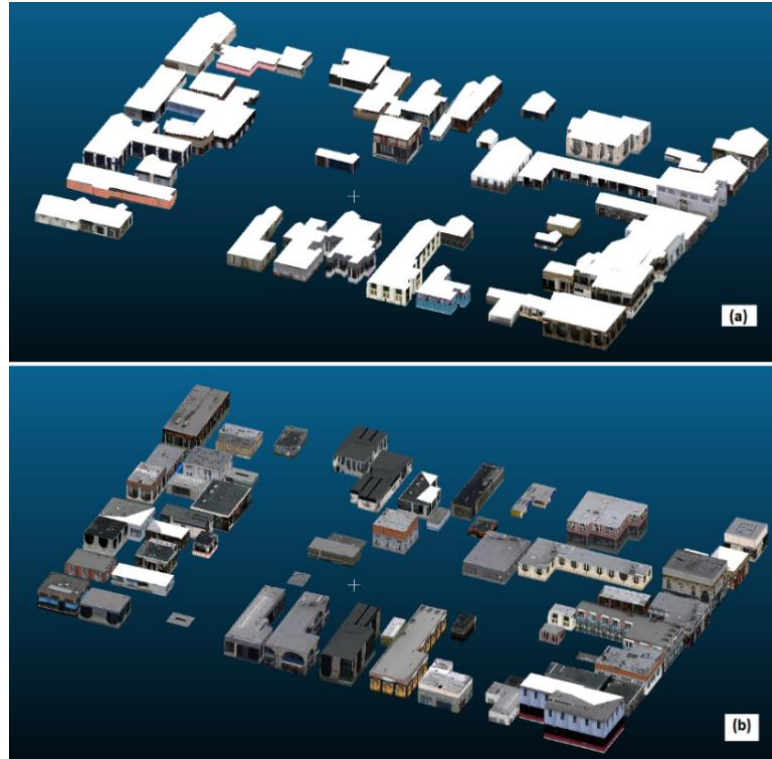


Figure 7.6 3D building models
(a) derived exclusively from LiDAR data; (b) generated based on footprints from TopRo5

7.3. Online distribution of 3D building models

7.3.1. ArcGIS Online and CityEngine WebViewer

ArcGIS Online is a software-available-as-a-service (SaaS) that provides the user with a series of tools for mapping and analysing 2D and 3D data (ESRI, 2021).

7.3.2. Data distributed online

Based on the documentation started during the writing of the article about *2D maps and web applications for Oradea city* (Pârnu I. M., Cuibac Picu, Dragomir, & Pantan, 2021), I decided to continue studying the capabilities of *ArcGIS Online* in working with 3D data. In this sense, I exported the scene from *CityEngine*, which contained the 3D building models for the city of Arad and the ground represented by the DTM, in *CityEngine WebScene* format.

The new file must be stored on the *ArcGIS Online* platform via the *Share As* option. Then, the data is available in the Content tab of *ArcGIS Online* and can be viewed and analysed in the *CityEngine Web Viewer* software (figure 7.7).

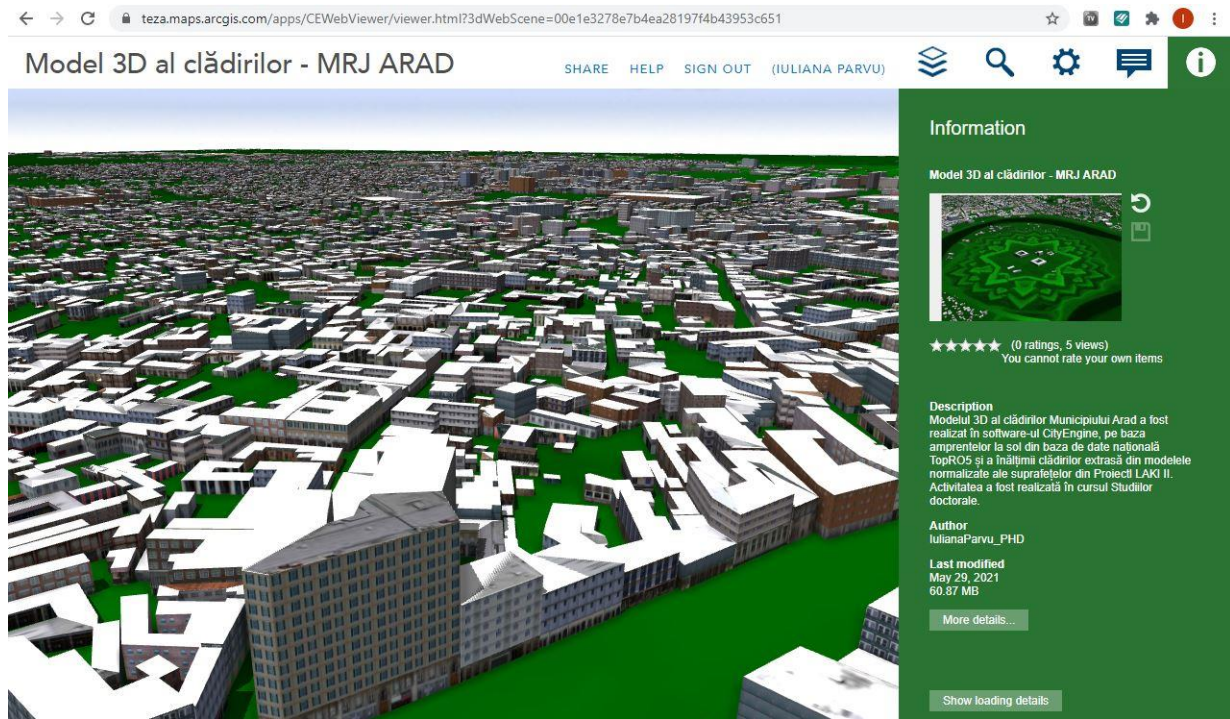


Figure 7.7 3D building models in CityEngine WebViewer

8. General conclusions, original contributions, and future perspectives

8.1. General conclusions

Conclusions on the acquisition of geospatial data by remote sensing and aerial photogrammetry

In terms of LiDAR technology, conventional linear scanners offer clear performance and, with the technological evolution offered by SPL and Geiger-mode scanners, capabilities and productivity have increased considerably. From the analysis of the performances of the three types of LiDAR instruments (linear scanners, SPL and Geiger-mode), it is observed that the scanning rate of a SPL system is 3 times higher than that of a linear system, and the scanning rate of a Geiger-mode system is 2 times larger than of a SPL system.

For every project, LiDAR scanning or aerial photogrammetry, the planning part is extremely important, and if this stage is not treated seriously, the data processing can be much difficult or even impossible to accomplish.

Point cloud processing workflow involves data calibration, strip adjustment, transformation into the coordinate system and point cloud classification. ASPRS defined, in 2019, the latest version of the LAS file, version 1.4, with 256 classes. Depending on the type of project, it is necessary to identify the optimal classes to use in the classification process. One should know, that a large number of classes complicates the classification process and increases confusion between classes (Pârvu I. M., Cuibac Picu, Dragomir, & Poli, 2020).

Currently, top photogrammetric cameras, produced by companies such as Leica Geosystems, Vexcel Imagine, are used. These can take nadir and oblique images. Compared to traditional photogrammetry, oblique imaging technology offers several advantages, such as: very good visibility of facades, multiple views and easy interpretation of buildings. The disadvantages of using oblique images are the different lighting of objects, the different spatial resolutions within the images and the increased hardware-software requirements. The most common configuration for oblique cameras is *Maltese-cross*.

In this thesis, I have detailed the photogrammetric workflow, presenting the latest information related to the production of true orthophotos from oblique images. Regarding the extraction of tie points from oblique images, the process is extremely laborious and computationally demanding. Using the SIFT operator facilitated the extraction process.

With images with a spatial resolution of 10 cm, the resulting dense point cloud (DIM) from nadir images has a density of 100 points/m², compared to the DIM obtained from nadir and oblique images that provide a triple density because of the points on the facades of

buildings. Even if the point cloud is much denser, problems arise because of the high percentage of noise points. To reduce this effect, 80% forward and side overlaps.

For the precise extraction of buildings in a semi-automatic or automatic way, the true orthophoto is the optimal choice, because the artificial objects are positioned correctly on the ground and the obstructed areas are eliminated.

Conclusions regarding the classification of LiDAR point clouds and the semantic segmentation of orthorectified images

I analyzed the segmentation methods of point clouds, starting with edge-based segmentation, continuing with region-based segmentation, arriving at model fitting methods and ending with Machine learning-based segmentation.

Starting with the basic workflow for the supervised classification of point clouds, I developed my workflow. The first step consists in selecting the neighborhood, based on which the eigenfeatures of the points will be computed. In the literature are used the following ways of selecting the neighborhood: choosing the value of the radius of the sphere or cylinder that determines the neighborhood around the point and choosing/computing the value of K nearest neighbors. Analyzing the options, I decided that using the number of K nearest neighbors is the optimal choice. In this sense, I proposed and implemented in a C++ code the use of the dynamic neighborhood of K, based on the region growing algorithm. Thus, each point will have its own K number of neighbors, depending on the similarity measurements typical of the region-based segmentation method.

The most well-known local 3D features are abstract eigenfaetures, which are determined based on the eigenvalues. Thus, I developed a C++ code for computing features, such as linearity, planarity, sphericity, anisotropy. Besides these 3D features described in the literature, I have defined several features, of which the N_2 feature has proven relevant in the classification process.

In the supervised classification, the assignment of the class is done by learning the model based on a set of training data and by running the model created on the entire data set. The training data set must represent the data very well and its classification must be extremely precise. The testing data set is used to assess the quality of the classification. Both data sets can be classified manually, automatically or by the crowdsourcing method.

The **Weka** software, developed by the University of Waikato, New Zealand, has implemented a collection of Machine learning algorithms for solving large-scale data extraction patterns with applications in banking, diagnostics, marketing and sales. In the thesis, I used this software for the supervised classification of LiDAR point clouds.

In the tests performed, I started the analysis using three classes (*ground, vegetation* and *roof*), but the classification results had very low quality indicators, because all points such as power lines, cars, fences, were classified in one of the 3 classes mentioned. Thus, I

included the *other* class, which contains points that do not belong to the ground, vegetation or roof class.

From the analyses regarding the time required to create and test the model for the 7 classifiers run in the Weka software, the longest time was required for the IBk classifier (1 hour and 45 minutes) and the shortest, for the Naive Bayes classifier (9 seconds) . The Random Forest classifier created and tested the model in just 7 minutes.

The ranking, in terms of classification quality indicators, is: 1st place - SMO, 2nd place - Random Forest and 3rd place - Naive Bayes. The overall classification accuracy obtained by these classifiers is approximately 93%. Analyzing the quality indicators for the *roof* class, I demonstrated that the Random Forest and SMO methods offer the best results, with values of the F1 score of approximately 93%.

After all the tests, I can say that Weka is a high-performance work environment and allows the precise classification of point clouds.

Regarding the extraction of the buildings footprints from orthorectified images, I analyzed the semantic segmentation based on Machine learning algorithms (Maximum Likelihood, Random Trees and SVM) and on Deep Artificial Neural Networks, in *ArcGIS Pro*. Even though the semantic segmentation quality indicators were close, for the Machine learning tests, SVM generated the best results. In order to increase the quality of the classification, I introduced, as auxiliary data, the DSM. Thus, the overall accuracy of the classification reached 93.9%, and the *buildings* class generated a F1 score of 93.3%, with 7% better than the approach without DSM.

I tested the ANN capabilities using the U-Net network with the ResNet architecture and ran the classification on CPU. The results obtained for the *BUILDINGS* class, using ResNet34 and ResNet50 were similar, observing a small decrease in precision (2.5%) and a slight increase in recall (5%). For the only class created, *BUILDINGS*, I obtained using ResNet34, the precision of 90.4%, the recall of 62.7% and the F1 score of 73.8%. The time required to create the ResNet50 model was 15 hours, 7 times longer than the one needed to create the ResNet34.

I recommend the use of graphics processing units (GPUs), which reduces the processing time.

Unfortunately, the resulting buildings footprints cannot be used in the process of generating 3D building models, because the orthorectified image used is an orthophoto, where most buildings appear tilted in the image. Therefore, in order to automatically extract the outline of buildings, it is recommended to classify true orthophotos.

Conclusions on 3D building modeling

The need to move from 2D maps to 3D models is important, both in terms of alignment with international standards and the usefulness of applications such as: estimating the potential of solar energy, estimating the energy requirements for individual buildings, prompt response to emergencies and study of the impact of floods, earthquakes.

Some countries that represent international models to follow in terms of 3D modelling are the Netherlands, Singapore, Switzerland, Estonia. In these countries, were developed 3D models for all buildings, using LoD1 or LoD2, and the data was published as open data on national portals. I believe Romania can and must move to the generation of 3D building models for its cities. The studies and results that I presented in this thesis can be the basis of a new national project of 3D building modelling.

In the case study, I started from the point clouds previously classified and extracted the buildings footprints, without manual intervention. I developed a Python code, in which I calculated the normals and applied the segmentation based on region growing, using the two approaches: conditional clustering and plane extraction. The buildings footprints were extracted using the alpha shape algorithm. The proposed workflow for the transition from segmented polygons to the regularised buildings footprints was done in ArcGIS Pro.

I tested the 3D building models generation in two software: 3dfier and CityEngine.

In 3dfier, I introduced the buildings footprints, the point cloud consisting points on the *ground* and on *roofs*, and the updated configuration file. The result was an obj file with 3D building models (LoD1).

Running the modelling process in CityEngine requires height information of the buildings. I extracted these data based on the generated nDSM, using the following approaches: based on the centroid points of the buildings footprints and based on characteristic points on the roof, much closer to its lower edges. From the analyses performed, the approach that used the characteristic points generated the best results. I generated the 3D building models (LoD2) based on the CGA rules, such as the type of roof, its angle of inclination, the texture that will be used for building facades.

For the quality assessment of the 3D models, I evaluated the planimetric and altimetric component. The planimetric RMSE was 0.38 meters. The altimetric RMSE was 1.55 meters. From the detailed analysis of the situations that generated this big value, I can say that some points are located on buildings with different types of roof than the one used by the CGA rules (flat, not inclined) or with a sloped roof in another direction. By removing these points from the computation, an altimetric RMSE of 0.63m is obtained. From the literature, the 3D accuracy in the 3D building models (LoD2) is 2 meters, therefore, I can state that the 3D RMSE obtained on the analysed data set (0.73 meters) falls within the standards in the field.

It is important to remember that the 3D building models are not exactly like in reality, but in order to have an excellent quality, they must be up-to-date, consistent and without topological or semantic errors.

In Chapter 7, I generated 3D models of buildings based on the data from the national database (TopRo5 - 2016) and the heights from nDSM (from the LAKI II Project - 2017). If the buildings had been digitized based on the orthophotos in LAKI II Project, then a 1: 1 correspondence could be made between the digitized buildings and the buildings in nDSM. This improvement would have allowed 3D modelling of all the buildings and would have led to an increase in the 3D accuracy of the models.

I set the angle of the roofs to 0°, which led to the generation of 3D block-type models, with flat roofs, for all buildings on the surface of the municipality.

Comparing the models generated only from LiDAR data, through the workflow proposed in Chapter 6, with the models resulted based on the buildings footprints from the national database and the heights from nDSM, I can conclude that:

- models generated only from LiDAR data offer superior results in terms of quality;
- models generated based on the data from the national database can successfully generate 3D models of buildings with the level of detail LoD1 and a standard deviation of about 1.5 meters.

The 3D building models generated for the city of Arad are stored in the ArcGIS Online platform, at <https://bit.ly/3fzndY2>, but I did not publish them as open data because of the copyright on the raw data.

3D building models of high quality can be obtained by topologically correcting the footprints, by including attributes that contain information about the type of roof, the slope, etc. for every building and by segmenting buildings with several levels of height.

8.2. Original contributions

The major topics of my thesis, namely the automatic extraction of geospatial data and 3D modelling, are of great interest because they present current problems in our field of research. As personal contributions I can mention:

- Performing a detailed analysis of the latest technological achievements in terms of equipment and data processing for aerial laser scanning and aerial photogrammetry;
- Performing the quality control of the data from the LAKI II project, from the planning stage to the generation of the final products;
- Use of the most current LiDAR data set at the national level;

- Carrying out a detailed analysis of the methods for classifying point clouds, focusing on the latest achievements;
- Carrying out a detailed analysis of the methods of semantic segmentation of images, emphasizing Machine learning techniques and deep neural networks architectures;
- Implementation of the region growing method for the dynamic selection of the neighborhood, to perform the supervised classification of point clouds;
- Use of Weka software for supervised classification of point clouds and comparative analysis of the machine learning algorithms implemented;
- Making the workflow from the polygons segmented in the OPALS software to the footprints of the regularised buildings;
- Transfer of building height values from nDSM using the approaches: based on the centroid points of the footprints and based on characteristic points on the roof, much closer to the lower edges of the roof;
- Development of a workflow for generating 3D building models using only LiDAR data;
- Generation of 3D models for all the building in Arad.

8.3. Future perspectives

The studies carried out in this thesis can be continued. I will mention some future research directions:

- Extracting the buildings footprints using the proposed workflow and generating the 3D building models, only from LiDAR data, for the entire surface of Arad;
- Including more attributes to the buildings, to generate, in CityEngine, roofs following their actual shape in the field;
- Extracting buildings footprints from true orthophotos and analysing in comparison the results with the footprints extracted from LiDAR data.

Bibliography

- Aerial photography. (2017). În F. Ramon (Ed.), *Basic Photography in 180 Days* (pg. 107-123).
- Agarwal, R. (2018, September 22). *Object Detection: An End to End Theoretical Perspective*. Preluat pe March 23, 2021, de pe towardsdatascience.com: <https://towardsdatascience.com/object-detection-using-deep-learning-approaches-an-end-to-end-theoretical-perspective-4ca27eee8a9a>
- ANCPI. (2016). *Informații generale proiect*. Retrieved from <http://www.ancpi.ro/>: <http://www.ancpi.ro/laki2/>
- ANCPI. (2021). *Portal ANCPI*. Preluat pe May 28, 2021, de pe <https://geoportal.ancpi.ro/>: <https://geoportal.ancpi.ro/geoportal/imobile/Harta.html>
- ASPRS. (2019). *LAS Specification 1.4 - R14*. Bethesda, Maryland: The American Society for Photogrammetry & Remote Sensing.
- Ballard, D. H. (1991). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2), 183-194.
- Biljecki, F., Ledoux, H., & Stoter, J. (2016). An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*. 59, pg. 25-37. Elsevier.
- Biljecki, F., Ledoux, H., Du, X., Stoter, J., Soon, K. H., & Khoo, V. H. (2016). THE MOST COMMON GEOMETRIC AND SEMANTIC ERRORS IN CITYGML DATASETS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W1, 13-22. doi:10.5194/isprs-annals-IV-2-W1-13-2016
- BIMserver. (2021, March 16). Preluat pe April 30, 2021, de pe <https://github.com/>: <https://github.com/opensourceBIM/BIMserver#readme>
- Breiman, L. (2001). Random Forests. *Machine Learning*, 5-32.
- Carter, J., Schmid, K., Waters, K., Betzhold, L., Hadley, B., Mataosky, R., & Halleran, J. (2012). *Lidar 101: An Introduction to Lidar Technology, Data, and Applications*. National Oceanic and Atmospheric Administration (NOAA) Coastal Services Center.
- Catmull, E. (1974). *A subdivision algorithm for computer display of curved surfaces*. Technical report, DTIC Document.
- Chakravarty, S. (2019, April 22). *Top 10 LiDAR Manufacturers in World*. Preluat de pe <https://www.marketresearchreports.com/>: https://www.marketresearchreports.com/blog/2019/04/22/top-10-lidar-manufacturers-world?fbclid=IwAR0jTBdNFGnkylFYFRlr54eRw19VbHt_XaEHZyVH0yUSTMBQPctPIBaYE8
- Chang, C. C., & Lin, C. J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on intelligent systems and technology*, 2(3).
- CNC. (2020a). *TOPRO5*. Preluat pe May 2021, de pe <https://cngcft.ro/>: <https://cngcft.ro/index.php/ro/portofoliu/lucrari-finalizate/topro5>
- Dorning, P., & Pfeifer, N. (2008). A Comprehensive Automated 3D Approach for Building Extraction, Reconstruction, and Regularization from Airborne Laser Scanning Point Clouds. *Sensors*, 7323-7343.
- Edelsbrunner, H., Kirkpatrick, D. G., & Seidel, R. (1983). On the Shape of a Set of Points in the Plane. *IEEE TRANSACTIONS ON INFORMATION THEORY*, IT-29(4), 551-559.

- ESRI. (2021). *ArcGIS Online*. Preluat pe May 30, 2021, de pe <https://www.esri.com/>: <https://www.esri.com/ro-ro/arcgis/products/arcgis-online/overview>
- Fernandez-Diaz, J. C., Carter, W. E., Shrestha, R. L., & Glennie, C. L. (2014). Now You See It... Now You Don't: Understanding Airborne Mapping LiDAR Collection and Data Product Generation for Archaeological Research in Mesoamerica. *Remote Sensing*, 6, 9951-10001. doi:10.3390/rs6109951
- Filin, S. (2002). SURFACE CLUSTERING FROM AIRBORNE LASER SCANNING DATA. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXIV, 119-124.
- Filin, S., & Pfeifer, N. (2006). Segmentation of airborne laser scanning data using a slope adaptive neighborhood. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(2), 71-80.
- Fischler, M. A., & Bolles, R. C. (1987). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. (M. A. Fischler, & O. Firschein, Ed.) *Readings in Computer Vision*, 726-740. doi:10.1016/B978-0-08-051581-6.50070-2
- Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of computer and system sciences*, 55, 119-139.
- Girardeau-Montaut, D. (2019, December 4-5). CloudCompare. *Point Cloud Processing Workshop*. Stuttgart.
- Golovinskiy, A., & Funkhouser, T. (2009). Min-cut based segmentation of point clouds. *IEEE Workshop on Search in 3D and Video at ICCV*.
- Grilli, E., Menna, F., & Remondino, F. (2017). A REVIEW OF POINT CLOUDS SEGMENTATION AND CLASSIFICATION ALGORITHMS. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W3*, 339-344.
- Grossfeld, B. (2021, April 7). *Deep learning vs machine learning: a simple way to learn the difference*. Preluat pe April 19, 2021, de pe www.zendesk.com: <https://www.zendesk.com/blog/machine-learning-and-deep-learning/>
- Haykin, S. (2008). *NEURAL NETWORKS AND LEARNING MACHINES*. New Jersey, United States of America: Pearson Education.
- Heidemann, H. K. (2018). Lidar Base Specification. *Collection and Delineation of Spatial Data*.
- Hough, P. (1962). METHOD AND MEANS FOR RECOGNIZING COMPLEX PATTERNS.
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *IEEE Conference on Computer Vision and Pattern Recognition*, (pg. 2261-2269). Honolulu.
- Hug, C., Krzystek, P., & Fuchs, W. (2004). ADVANCED LIDAR DATA PROCESSING WITH LASTOOLS. *ISPRS*.
- John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. *Proceedings of the conference on uncertainty in artificial intelligence*, 338-345.
- Jokela, J. (2016). *CityGML building model production from airborne laser scanning*. Master Thesis - Science in Technology, Aalto University, School of Engineering, Espoo.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). IMAGENET CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS. *Advances in neural information processing systems*, 25(2). doi:10.1145/3065386
- Lin, M., Chen, Q., & Yan, S. (2014). NETWORK IN NETWORK. *ICLR 2013*. arXiv:1312.4400v3.

- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition*. Boston, MA, USA: IEEE.
- Lu, X., Yao, J., Tu, J., Li, K., Li, L., & Liu, Y. (2016). PAIRWISE LINKAGE FOR POINT CLOUD SEGMENTATION. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, III-3*, 201-208.
- McCullough, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5*, 115-133.
- Morariu, I. D. (2007). *Contribuții la extragerea automată de cunoștințe din masive de date*. Teză de doctorat, Universitatea Lucian Blaga, Facultatea de Inginerie Hermann Oberth, Sibiu, România.
- Musäus, S. (2019). The SPL 100 in Europe. *EUROSDR Workshop on Single Photon and Geiger-Mode LiDAR*. Barcelona, Spain.
- Nguyen, A., & Le, B. (2013). 3D point cloud segmentation: A survey. *6th International Conference on Robotics, Automation and Mechatronics*, 225-230.
- Noh, H., Hong, S., & Han, B. (2015). LEARNING DECONVOLUTION NETWORK FOR SEMANTIC SEGMENTATION. *IEEE International Conference on Computer Vision*, 1520-1528.
- Ohbuchi, R., & Takei, T. (2003). Shape-Similarity Comparison of 3D Models. *Pacific Graphics*. Canmore, Canada.
- Pârvu, I. M., Cuibac Picu, I. A., Dragomir, P. I., & Pantan, R. (2021). Promoting Open Data services to decision-makers: Providing interactive data through Web Maps and Web Applications for Oradea city and Bihor county. *IOP Conference Series: Earth and Environmental Science, 664*. doi:10.1088/1755-1315/664/1/012073
- Pârvu, I. M., Cuibac Picu, I. A., Dragomir, P. I., & Poli, D. (2020). URBAN CLASSIFICATION FROM AERIAL AND SATELLITE IMAGES. *JOURNAL OF APPLIED ENGINEERING SCIENCES, 10(23)(2/2020)*, 163-172.
- Pepe, M., Fregonese, L., & Scaioni, M. (2018). Planning airborne photogrammetry and remote-sensing missions with modern platforms and sensors. *European Journal of Remote Sensing, 51(1)*, 412-436. doi:10.1080/22797254.2018.1444945
- Pfeifer, N., Mandlbauer, G., Otepka, J., & Karel, W. (2014). OPALS - A framework for Airborne Laser Scanning data analysis. *Computers, Environment and Urban Systems, 45*, 125 - 136.
- Phillips, D. K. (2015, May 13). *Speed of the Human Brain*. Preluat pe April 16, 2021, de pe <https://askabiologist.asu.edu/>: <https://askabiologist.asu.edu/plosable/speed-human-brain>
- Remondino, F., Toschi, I., Nex, F., & Gerke, M. (2018). *Oblique Aerial Camera Systems for Mapping Purposes*. e-learning course: EuroSDR.
- Residual Net*. (2020). Preluat pe April 10, 2021, de pe <https://leonardoaraujosantos.gitbook.io/>: https://leonardoaraujosantos.gitbook.io/artificial-intelligence/machine_learning/deep_learning/residual_net
- RIEGL. (2021, May 17). *RiProcess_Datasheet*. Preluat pe May 19, 2021, de pe [riegl.com](http://www.riegl.com/): http://www.riegl.com/uploads/tx_pxpriegl/downloads/RiProcess_Datasheet_2021-05-17.pdf
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *MICCAI 2015*, 234-241. doi:10.1007/978-3-319-24574-4_28

- Rupnik, E., Nex, F., & Remondino, F. (2014). OBLIQUE MULTI-CAMERA SYSTEMS - ORIENTATION AND DENSE MATCHING ISSUES. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-3/W1*, 107-114. doi:10.5194/isprsarchives-XL-3-W1-107-2014
- Rusu, R. B., & Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). *IEEE International Conference on Robotics and Automation*, 1-4.
- SAFE. (2021). *FME Transformer Gallery*. Preluat pe April 30, 2021, de pe <https://www.safe.com/>: <https://www.safe.com/transformers/?datatype=BIM>
- Sampath, A., & Shan, J. (2010). Segmentation and reconstruction of polyhedral building roofs from aerial lidar point clouds. *IEEE Transactions on geoscience and remote sensing*, 48(3), 1554–1567.
- Sapkota, P. P. (2008). *Segmentation of coloured point cloud data*. Thesis for Master of Science, International Institute Geo-Information Science and Earth Observation, Enschede, Netherlands.
- Schnabel, R., Wahl, R., & Klein, R. (2006). Shape Detection in Point Clouds. *SIGGRAPH 2006*. Preluat de pe <https://cg.cs.uni-bonn.de/aigaion2root/attachments/cg-2006-2.pdf>
- SECOM. (2020, July 30). *IFCExplorer 1.0*. Preluat pe April 30, 2021, de pe <https://ifcexplorer.software.informer.com/>: <https://ifcexplorer.software.informer.com/>
- Shragai, Z., Barnea, S., & Even-Paz, A. (2012). Automatic Tie-point Extraction. *Gim International*.
- Sibson, R. (1973). SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1), 30-34.
- SPAR 3D. (2016). *Answers, Not Data: A Year of Harris' Commercial Geiger-Mode LiDAR*. Preluat de pe [spar3d.com](https://www.spar3d.com/): <https://www.spar3d.com/news/lidar/answers-not-data-a-year-of-harris-commercial-geiger-mode-lidar/>
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9. doi:10.1109/CVPR.2015.7298594
- Terrasolid. (2021). *TerraScan*. Preluat pe May 17, 2021, de pe <https://terrasolid.com/>: <https://terrasolid.com/products/terrascan/>
- TUdelft. (2021, February 26). *The open-source tool for creation of 3D models*. Preluat pe May 9, 2021, de pe <http://tudelft3d.github.io/>: <http://tudelft3d.github.io/3dfier/>
- Ukrainitz, Y., & Sarel, B. (2014, November 20). *Segmentation and Clustering*. Preluat pe April 15, 2021, de pe <https://www.slideserve.com/>: <https://www.slideserve.com/hamilton-moore/cse-473-573-computer-vision-and-image-processing-cvip>
- Upadhyay, N. (2013). *Basics of Photogrammetry*. Preluat de pe <https://www.gisresources.com/>: https://www.gisresources.com/basic-of-photogrammetry_2/
- Vrejoiu, M. H. (2019). REȚELE NEURONALE CONVOLUȚIONALE, BIG DATA ȘI DEEP LEARNING ÎN ANALIZA AUTOMATĂ DE IMAGINI. *Romanian Journal of Information Technology and Automatic Control*, 29(1), 91-114.
- Waikato, U. o. (2021). *WEKA*. Preluat de pe www.cs.waikato.ac.nz: <https://www.cs.waikato.ac.nz/ml/weka/>
- Wehr, A. (2018). LIDAR Systems and Calibration. În J. Shan, & C. Toth, *Topographic Laser Ranging and Scanning: principles and processing* (pg. 159-200). CRC Press.

- Weinmann, M. (2013). Visual Features—From Early Concepts to Modern Computer Vision. În *Advanced Topics in Computer Vision* (pg. 1-34). London: Springer-Verlag.
- Weinmann, M. (2016). *Reconstruction and Analysis of 3D Scenes*. Springer International Publishing Switzerland.
- Weinmann, M. (2019). *An Introduction to Classification*. EuroSDR EduServ17.
- Weinmann, M., Jutzi, B., & Mallet, C. (2014). SEMANTIC 3D SCENE INTERPRETATION: A FRAMEWORK COMBINING OPTIMAL NEIGHBORHOOD SIZE SELECTION WITH RELEVANT FEATURES. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3, 181-188.
- Weinmann, M., Schmidt, A., Mallet, C., Hinz, S., Rottensteiner, F., & Jutzi, B. (2015). CONTEXTUAL CLASSIFICATION OF POINT CLOUD DATA BY EXPLOITING INDIVIDUAL 3D NEIGHBOURHOODS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W4, 271-278. doi:10.5194/isprsannals-II-3-W4-271-2015
- Xie, Y., Tian, J., & Zhu, X. X. (2020). Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE GEOSCIENCE AND REMOTE SENSING MAGZINE*.
- Yadav, N., Yadav, A., & Kumar, M. (2015). *An Introduction to Neural Network Methods for Differential Equations*. Springer.
- Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., & Yan, G. (2016). An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sensing*, 8(6). doi:10.3390/rs8060501
- Zhu, X. X., & Shahzad, M. (2014). Facade reconstruction using multiview spaceborne tomosar point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 52(6), 3541–3552.

Published articles

- Pârvu, I. M.,** Cuibac Picu, I. A., Dragomir, P. I. (2018). Pixel Based Classification of Images. *RevCAD 25/2018*, 83-90.
- Pârvu, I. M.,** Remondino, F., & Özdemir, E. (2018). LOD2 building generation experiences and comparisons. *JOURNAL OF APPLIED ENGINEERING SCIENCES*. doi:10.2478/jaes-2018-0019.
- Pârvu, I. M.,** Özdemir, E., & Remondino, F. (2020). Aerial point cloud classification using an alternative approach for the dynamic computation of k-Nearest Neighbors. *JOURNAL OF APPLIED ENGINEERING SCIENCES*, 10(23)(2/2020), 155-162.
- Pârvu, I. M.,** Cuibac Picu, I. A., Dragomir, P. I., & Poli, D. (2020). Urban classification from aerial and satellite images. *JOURNAL OF APPLIED ENGINEERING SCIENCES*, 10(23)(2/2020), 163-172.
- Pârvu, I. M.,** Cuibac Picu, I. C., Dragomir, P. I., & Pantan, R. (2021). Promoting Open Data services to decision-makers: Providing interactive data through Web Maps and Web Applications for Oradea city and Bihor county. *IOP CONFERENCE SERIES: EARTH AND ENVIRONMETAL SCIENCE*, 664. doi:10.1088/1755-1315/664/1/012073.