VILNIUS GEDIMINAS TECHNICAL UNIVERSITY
FACULTY OF ENVIRONMENTAL ENGINEERING
DEPARTMENT OF GEODESY AND CADASTRE

Eugenijus Streikus

3D MIESTO MODELIAVIMAS NAUDOJANT LiDAR DUOMENIS IR GIS TECHNOLOGIJŲ ANALIZĘ

3D CITY MODELLING USING LIDAR DATA AND GIS TECHNOLOGIES FEATURES ANALYSIS

Master‘s degree Thesis

Innovative Solutions in Geomatics, joint study programme, state code 6281EX001
Measurement engineering study field

Vilnius, 2019

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY
FACULTY OF ENVIRONMENTAL ENGINEERING
DEPARTMENT OF GEODESY AND CADASTRE

APPROVED BY
Head of Department
____
(Signature)
Jūratė Sužiedelytė-Visockienė
(Name, Surname)
2019-01-04
(Date)

Eugenijus Streikus

3D CITY MODELLING USING LIDAR DATA AND GIS TECHNOLOGIES FEATURES ANALYSIS

Master's degree Thesis

Innovative Solutions in Geomatics, joint study programme, state code 6281EX001
Measurement engineering study field

Supervisor prof. dr. Eimuntas Kazimieras Paršeliūnas ______ 2019 01 04
(Title, Name, Surname) (Signature) (Date)

Vilnius, 2019

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY
FACULTY OF ENVIRONMENTAL ENGINEERING
DEPARTMENT OF GEODESY AND CADASTRE

APPROVED BY
Head of department
(Signature)
Jūratė Sužiedelytė-Visockienė
(Name, Surname)
2018-11-29
(Date)

Measurement engineering study field

Innovative Solutions in Geomatics, joint study programme, state code 6281EX001

OBJECTIVES FOR MASTER THESIS

.................No. ....
Vilnius

For student Eugenijus Streikus
(Name, Surname)

Master Thesis title: 3D city modelling using LIDAR data and GIS technologies features analysis.

Approved on 29 November, 2018 by Dean‘s decree No. 192ap
(day, Month) (year)

The Final work has to be completed by ....4 January.................., 201.9.
(Day, Month) (Year)

THE OBJECTIVES:

Perform analysis of literary sources on the topic of Master's thesis. Examine the peculiarities and principles of airborne laser scanning. Review what affects data accuracy. Find out what LiDAR data formats are used in GIS software and compare scan data of 2007 and 2018. Perform and describe in detail the 3D city modelling process using existing LiDAR data and GIS technologies. Create scripts, models and CGA rules for 3D city modelling and adapt them using ArcGIS Pro and CityEngine environment...........................

Consultants of the Master Thesis: ....
(Title, Name, Surname)

Academic Supervisor .... prof. dr. Eimuntas Kazimieras Paršeliūnas
(Signature) (Title, Name, Surname)

Objectives accepted as a guidance for my Master Thesis

....
(Student‘s signature)
Eugenijus Streikus
(Student‘s Name, Surname)
2018-11-29
(Date)

Vilniaus Gedimino technikos universitetas
Aplinkos inžinerijos fakultetas
Geodezijos ir kadastro katedra

ISBN ISSN
Egz. sk. ....
Data ..........-.....-.....

Antrosios pakopos studijų **Inovatyvūs sprendimų geomatikoje** programos magistro baigiamasis darbas 3

Pavadinimas **3D miesto modeliavimas naudojant LiDAR duomenis ir GIS technologijų analizę**

Autorius **Eugenijus Streikus**

Vadovas **Eimuntas Kazimieras Paršeliūnas**

Kalba: anglų

Anotacija

Baigiamajame magistro darbe išanalizuoti LiDAR duomenų surinkimo iš orlaivio ypatumai ir principai. Apžvelgta kas turi įtakos duomenų tikslumui. Aprašytas LiDAR duomenų formatas naudojamas GIS programinėje įrangoje. Palyginti 2007 ir 2018 metų skenavimo duomenys. Atliktas ir detaliai apibūdintas 3D miesto modeliavimo procesas naudojant esamus LiDAR duomenis ir GIS technologijas. Panaudojus ArcGIS Pro ir CityEngine programinę įrangą sukurti ir pritaikyti skriptai, modeliai, ir CGA taisyklės reikalingos 3D miesto modeliavimui.

Darbą sudaro keturios dalys: įvadas, du skyriai, išvados, literatūros sąrašas.
Darbo apimtis - 58 p. teksto be priedų, 59 iliustr., 4 lent., 42 bibliografiniai šaltiniai.
Atskirai pridedami darbo priedai.

Prasminiai žodžiai: LiDAR, GIS, CGA, 3D miesto modeliavimas

Vilnius Gediminas Technical University
Faculty of Environmental Engineering
Department of Geodesy and Cadastre

ISBN ISSN
Copies No. ....
Date ..........-.....-.....

Master Degree Studies **Innovative Solutions in Geomatics** study programme Master Graduation Thesis 3

Title	**3D city modelling using LiDAR data and GIS technologies features analysis**
Author	**Eugenijus Streikus**
Academic supervisor	**Eimuntas Kazimieras Paršeliūnas**

Thesis language: English

Annotation

The final master's thesis analyzed the characteristics and principles of LiDAR data collection from aircraft. Reviewed what affects the accuracy of the data. Described LiDAR data format used GIS software. Compared of 2007 and 2018 scanning data. Performed and described in detail the 3D city modeling process using existing LiDAR data and GIS technology. Created scripts, models and CGA rules for 3D city modelling and adapted them using ArcGIS Pro and CityEngine environment.

The Final Thesis consists of four parts: introduction, two chapters, conclusions, bibliographic resources.
Scope of Thesis: 58 pages of text without appendixes, 59 figures, 4 tables, 42 bibliographic resources.
Appendixes included.

Keywords: LiDAR, GIS, CGA, 3D city modelling

Vilniaus Gedimino technikos universiteto
egzaminų sesijų ir baigiamųjų darbų rengimo
bei gynimo organizavimo tvarkos aprašo
2 priedas

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Eugenijus Streikus, 20093449

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Aplinkos inžinerijos fakultetas

(Fakultetas)

Inovatyvūs sprendimai geomatikoje, ISGmfu-17

(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO (PROJEKTO)
SĄŽININGUMO DEKLARACIJA

2019 m. sausio 03 d.

Patvirtinu, kad mano baigiamasis darbas tema „3D miesto modeliavimas naudojant LiDAR duomenis ir GIS technologijų analizę" patvirtintas 2018 m. lapkričio 29 d. dekano potvarkiu Nr. 192ap, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Parenkant ir įvertinant medžiagą bei rengiant baigiamąjį darbą, mane konsultavo mokslininkai ir specialistai: prof. dr. Eimuntas Kazimieras Paršeliūnas. Mano darbo vadovas prof. dr. Eimuntas Kazimieras Paršeliūnas.

Kitų asmenų indėlio į parengtą baigiamąjį darbą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).

(Parašas)	Eugenijus Streikus (Vardas ir pavardė)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

LIDAR	Light Detection and Ranging
GPS	Global Positioning System
DGPS	Differential Global Positioning System
IMU	Inertial Measurement Unit
RMSE	Root Mean Square Error
ASCII	American Standard Code for Information Interchange
ASPRS	American Society for Photogrammetry and Remote Sensing
VLR	Variable Length Record
EVLR	Extended Variable Length Record
GIS	Geographic Information System

INTRODUCTION

Relevance of the work. Within a time of only two decades airborne laser scanning have well established surveying techniques for the acquisition of (geo) spatial information. A wide variety of instruments is commercially available, and a large of companies operationally use airborne scanners, by many dedicated data acquisition, processing and visualisation software packages. The high quality 3D point clouds produced by laser scanners are nowadays routinely used for a diverse array of purposes including the production of digital terrain models and 3D City models, forestry sector, corridor mapping, etc. However the publicly accessible knowledge on laser scanning is distributed over a very large number of scientific publications, web pages and tutorials. With sensor resolution, point clouds acquired by airborne laser scanners nowadays dense enough to also retrieve detailed on detection and extraction 3D of buildings and other urban-specific features.

Every day, planners use geographic information system (GIS) technology to research, develop, implement, and monitor the progress of their plans. GIS provides planners, surveyors, and engineers with the tools they need to design and map their neighbourhoods and cities. Planners have the technical expertise, political savvy, and fiscal understanding to transform a vision of tomorrow into a strategic action plan for today, and they use GIS to facilitate the decision-making process.

Planners have always been involved in developing communities. Originally, this meant designing and maintaining cities and counties through land use regulation and infrastructure support. Agencies have had to balance the needs of residential neighbourhoods, agricultural areas, and business concerns. Now, in addition to that complex challenge, local governments must factor into these decisions the requirements of a growing list of regional, state, and federal agencies as well as special interest groups.

Rapidly changing economic conditions have further complicated the process by threatening the funding needed to carry out these functions. To date, local governments have been rightsized and downsized and have had budgets drastically cut while trying to maintain service levels. Information technology, especially GIS, has proven crucial in helping local governments cope in this environment.

ESRI software solutions help planning, building and safety, public works, and engineering professionals meet or exceed these demands. ESRI *ArcGIS Pro* and *CityEngine* software is one of the local governments' options for mapping and analysis. Using this software, planning agencies have already discovered how traditional tasks can be performed more efficiently and tasks - previously impractical or impossible - can be easily accomplished.

The objective of this work is to give a comprehensive overview of the principles how airborne laser scanning technology and 3D point clouds acquired by laser scanners can be served for 3D city modelling and urban features reconstruction using GIS technology.

Work objective. Perform an analysis of the features of 3D city modelling in GIS environment.

Work tasks:

1. Examine the peculiarities and principles of airborne laser scanning. Review what affects data accuracy. Find out what LiDAR data formats are used and compare scan data of 2007 and 2018.
2. Perform and describe in detail the 3D city modelling process using existing LiDAR data and GIS technologies.
3. Create scripts, models and CGA rules for 3D city modelling and adapt them using *ArcGIS Pro* and *CityEngine* environment.

Object of research. 1 km x 1 km area of Vilnius city center.

Research methodology. Practical application and processing of LiDAR data, and features extraction data using GIS technologies.

Scientific novelty. In Lithuania, 3D city GIS models are not used, offered technique and methodology how this can be done in practice.

Practical importance of scientific work, its applicability. Analyse and describe the 3D city modelling using existing LiDAR data and GIS technologies. Provide practical application of the obtained results.

Publications and reports on conferences were published on the topic of the final work. 21st conference for Junior Researchers "Science – Future of Lithuania", civil engineering and geodesy, on March 23, 2018, an article was published, entitled "3D building modelling by LiDAR data and GIS technologies". The article is included in the appendixes.

Scope and structure of the work. The Final Thesis consists of four parts: introduction, two chapters, conclusions, bibliographic resources.

Scope of Thesis: 58 pages of text without appendixes, 59 figures, 4 tables, 42 bibliographic resources.

Appendixes included.

1. PRINCIPLES OF LIDAR REMOTE SENSING

LiDAR is a remote sensing technology that detects the distances (or ranges), based on the time from laser signal transmission and reception. In practice, pulsed or continuous wave lasers are used: pulsed lasers transmit energy for a very short time and determine ranges based on amplitudes of received signals; on the contrary, continuous wave lasers detect ranges according to phase differences between transmitted and received signals (Baltsavis 1999b).

1.1. Basics of airborne laser scanning

Over the past decades, topographic laser profiling and scanning systems have evolved rapidly and have undoubtedly become an important technology for collecting geographic data. Equipped with both on-air and on-board platforms, these systems can collect clear 3D data in large quantities without any precedent. The complexity of the measured laser data processing is not high, which has led to a rapid spread of this technology to various applications. Although the invention of the laser begins in the early 1960s (Nelson 2014), the lack of a variety of supportive technologies has prevented the use of this device from mapping for decades.

At the end of the nineties, already having GPS, a method was developed to accurately determine the situation and orientation in large areas. By introducing a differential GPS (DGPS), the scanner's position has become known in horizontal and vertical coordinates in the subdecimeter range (Krabill 1989, Friess 1989). Improvements to DGPS technology, combined with Kalman filtering and IMU use, have been precise enough since the early 1980s (Lindenberger 1989, Schwarz et al 1994, Ackermann 1994). Standard height data accuracy increased by +10 cm and +50 cm. By the end of the 1980s, measurements were made by means of laser profilers (Ackermann 1988, Lindenberger 1993), providing laser pulses but without a scanning mechanism. Since early 1990s, profilers were replaced with scanning devices that generated from 5000 to 10,000 laser pulses per second. Nowadays, laser pulse rates can reach 300 kHz frequency.

1.2. Principle of airborne laser scanning

Airborne laser scanning is performed from a fixed wing aircraft or helicopter. This technique is based on two main components: 1. A laser scanner system that measures the distance to a location on the ground. 2. The GPS/IMU combination that accurately measures the position and orientation of the system. Laser scanning systems are relatively independent of the sun and can be performed during daytime or at night. This characteristic is a great advantage comparing to other ways of measuring landscapes.

The aircraft's laser scanner is completed by a GPS ground station. The ground station acts as a reference station for separate differential GPS (DGPS) calculations. DGPS is very important in compensating for atmospheric effects, preventing accurate positioning and achieving decimetre accuracy In order to cope with different atmospheric conditions and obtain better results, the distance between the aircraft and the GPS ground station must not exceed 30 km (although sometimes the correct accuracy is achieved over long distances).

Airborne laser scanners can be complemented by a digital camera system. Image data received along with range data makes it easier to interpret data in cases where it is difficult to recognize objects only from range data. Image data generally provides better spatial resolution and provides the basis for an integrated 3D point cloud and image processing. The optimal location for the camera is on the scanner assembly’s ground plate, because then the existing IMU registrations can be shared for georeferencing images.

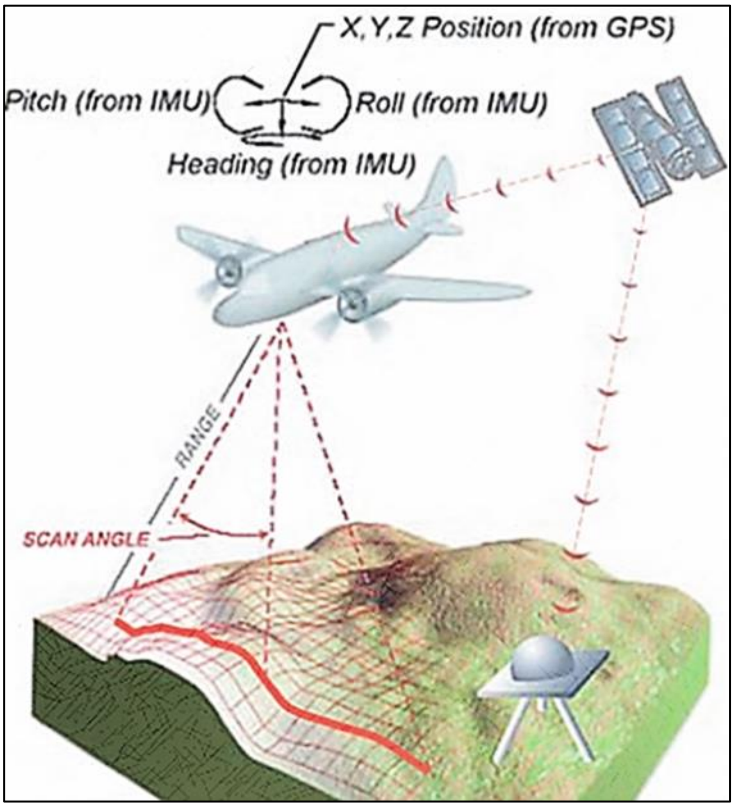


Figure 1. Principle scheme of airborne laser scanning (Petrie 2011)

1.3.Lidar accuracy

In practice, LIDAR accuracy is checked by comparing known points (measured) with laser scan data and defined as the standard deviation (σ^2) and root mean square error (RMSE) (Csanyi & Toth 2007).

According to Csanyi & Toth, modern LiDAR technology can reach several centimetres accuracy in ideal conditions, and this precision is sufficient for most engineering topographical works.

However, the desired accuracy is limited due to a navigation-based sensor platforms, as well as the LiDAR system consisting of a variety of multisensory and components, even after both the system and individual sensor calibration work can be identified certain errors which degrades the final accuracy of LiDAR data. The most common errors are navigation-based, these errors cannot be avoided without data control, so for LiDAR-specific ground control points is recommended to use to get the best results.

1.4.Lidar data formats

In the very first days when LiDAR data was collected and stored, many companies used a generic American Standard Code interchange system (ASCII). However, using this format has encountered some problems: (1) the reading and interpretation of ASCII data can be very slow and the file size can be very high even for small amounts of data, all of which greatly affects performance. (2) Loss of useful information is related to LiDAR data. (3) Format is not standard (Fig. 2).

a)			b)				
569999.790	6060193.120	96.810	599446.4570	1771.430	-2370.530	166.850	152.0
569999.250	6060193.080	96.810	599446.4570	1772.330	-2370.880	166.820	124.0
569998.690	6060193.040	96.780	599446.4570	1773.180	-2370.240	166.750	105.0
569998.170	6060193.000	96.790	599446.4570	1774.660	-2369.770	166.700	180.0
569996.780	6060193.150	96.530	599546.4770	1775.280	-2369.980	166.650	158.0
569997.890	6060193.240	96.760	599446.4770	1776.980	-2369.840	166.740	170.0
569998.430	6060193.280	96.780	599446.4770	1777.500	-2369.480	166.420	194.0
569998.950	6060193.330	96.750	599446.4770	1777.680	-2369.220	166.320	184.0

Figure 2. Example of LiDAR data in ASCII data files. The numbers in each row are: (a) x, y, z; (b) GPS time, x, y, z, intensity.

In order to better exchange LiDAR point cloud data, the American Society for Photogrammetry and Remote Sensing (ASPRS) introduced the LAS file format. This is binary file format that maintains information specific to the LiDAR nature of the data while being not very complex. The ASPRS LAS 1.0 Format Standard was released on May 9, 2003; LAS 1.1 on May 7, 2005; LAS 1.2 on September 2, 2008; LAS 1.3 on October 24, 2010; and LAS 1.4 On November 14, 2011. Since update (July 26 2013) of LAS 1.4 format the LiDAR mapping community has ability customize the LAS file format to meet their needs. The mechanism that makes this available is the LAS Domain Profile which is derivative of the base LAS 1.4 version specification that adds (but not removes alters) point classes and attributes. The point cloud files used in the project are all in LAS format. The up-to-date specifications of all LAS versions can be accessed on the website of ASPRS.

The structure of the LAS data file is described in detail in the LAS specification document (ASPRS 2011). Each LAS file consists of a public header block. Any number of Variable Length Records (VLRS), point data records, and any number of Extended Variable Length Records

(EVLRS). Tile public header block stores basic summary information such as boundary and the total number of the points. VLR consists of information about map projection and other metadata. EVLR is mainly used to store waveform data. EVLR started only from version 1.3 of the LAS. Public header block and point data records are required, but VLR and EVLR are optional. Therefore, LAS files do not necessarily include information such as the LiDAR data map projection. In this case, the user must get information from metadata files, reports, or by data provider.

Table 1.4.1. LAS 1.4 format definition

LAS file section	Note
Public header block	Required
Variable length records (VLRs)	Optional
Point data records	Required
Extended Variable Length Records (EVLRs)	Optional

Each point data record stores specific information such as the points x, y, z, intensity, return number, number of returns (given pulse), scan direction, edge of flight line, classification, GPS time, point source, etc. (Table 1.4.2). If the images were taken during the flight, the spectral values (e.g. red, green, blue) can be also stored at the point data record. Spectral values useful for realistic visualisation of the scanned landscapes (Figure 3).

Table 1.4.2. An example of point data record format

Item
X
Y
Z
Intensity
Return Number
Number of returns
Scan Direction Flag
Edge of Flight line
Classification
Scan Angle Rank (+90 to -90) -Left side
User Data
Point Source ID
Red
Green
Blue

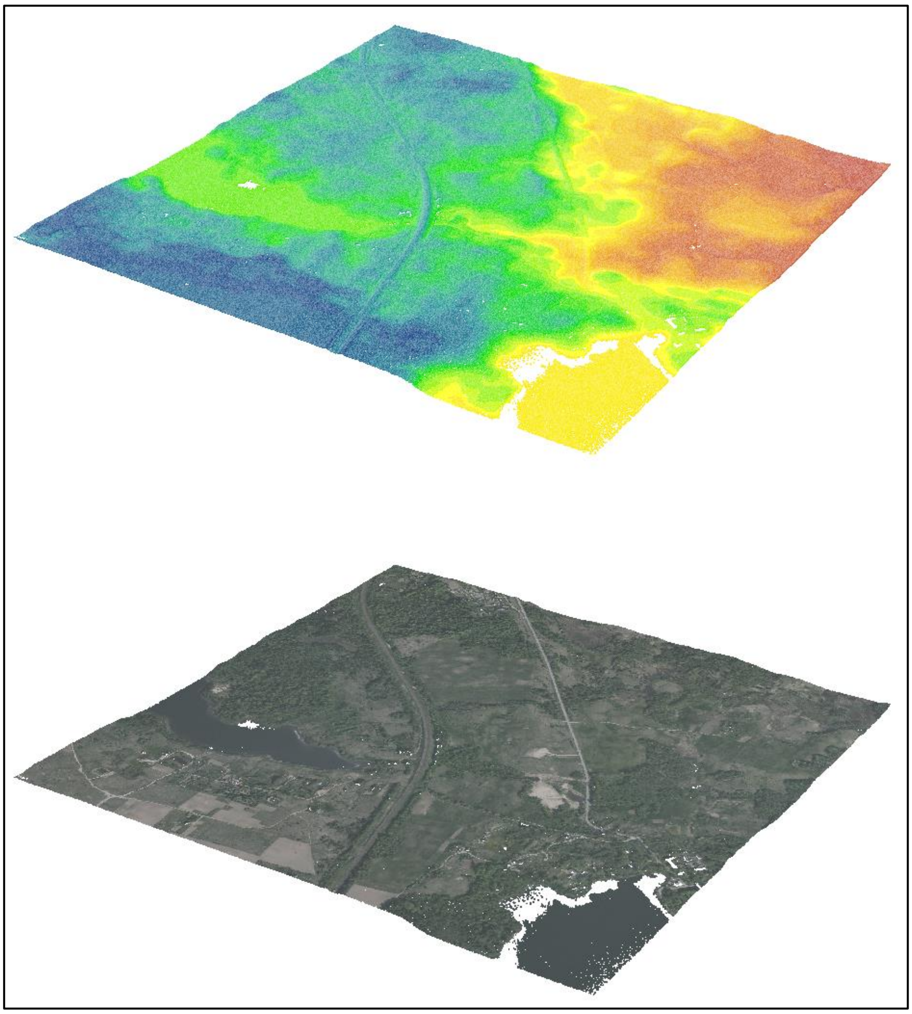


Figure 3. LiDAR points rendered by elevation (top) and RGB (bottom) spectral values

Regardless of the LAS file format, one of the most important stored attribute data is the classification. This attribute describes the object's laser beam reflection from the specific surface (for example, earth, building, vegetation, water, etc.). Table 1.4.3 is an example of a standard classification scheme, which is defined for LAS 1.4 data point data record formats 6-10. If the data is not classified (i.e. class 0 or 1), this significantly limits the resolution of the analysis and visualization tasks and vice versa. If the data is classified, digital terrain model generation (class 2) or building footprints extraction (class 6) can be performed more quickly, as well as other specific tasks.

Table 1.4.3. ASPRS Standard LiDAR Classes (Point data record formats 6-10)

Classification value	Meaning
0	Created, never classified
1	Unclassified
2	Ground
3	Low Vegetation
4	Medium Vegetation
5	High Vegetation

6	Building
7	Low Point (noise)
8	Reserved
9	Water
10	Rail
11	Road Surface
12	Reserved
13	Wire – Guard (Shield)
14	Wire – Conductor (Phase)
15	Transmission Tower
16	Wire-structure Connector (e.g. Insulator)
17	Bridge Deck
18	High Noise
19-63	Reserved
64-255	User definable

1.5.Overview of LiDAR data in Lithuania

In spring of 2007, at the order of the National Land Service under the Ministry of Agriculture, for the first time in Lithuania were made 10 largest cities - county centers: Vilnius, Kaunas, Marijampolė, Alytus, Klaipėda, Tauragė, Telšiai, Šiauliai, Panevėžys and Utena - a total of 2,475 km2 area (Žalnierukas et al 2009).

At the same time, a digital aerial photograph of large-scale urban areas was also made and orthophotographic maps produced. The aim of the work is the creation of 3D urban models using hybrid laser scanning and digital aerial photography technology. Airborne scans and orthophotographs were produced by the French FIT Conseil - Géométres Experts. Ground measurements were carried out by UAB InfoERA in Lithuania, and the quality control of production was carried out by the Institute of Aerogeodesis.

Urban areas are scanned by the Optech ALTM 3100 scanner from Antonov-2. For ground measurements, use the Z-Max GPS receivers (FIT Conseil 2007). The position is determined by the Trimble 750 GPS Navigation System - Applanix POS/AV IMU.

When compiling the 2007 LiDAR data, the technical requirements foresee that there should be 4 fixed points per square meter, the average distance between the points is 0.50 m. The height measurement accuracy shall not be less than 15 cm and the accuracy of the planimetric shall be 30 cm. The LiDAR laser radius measures from 1 to 3 cm to a hard surface (Kalantaitė et al 2010). Other data for the flight: speed - 205 km / h; distance between lanes - 300 m; side overlay - 30%; laser dot density 3-4 points per 1m2 (Žalnierukas et al 2009).

More than ten years later, the National Land Service under the Ministry of Agriculture carried out LiDAR scanning work with orthophoto images for the same 10 largest county centers.

In accordance with the technical requirements (National Land Service under Ministry of Agriculture, 2017), the number of points per square meter must be not less than 1.2, the average square error of the horizontal position of the point is not more than 30 cm, the average square error of the vertical position of the point is not more than 10 cm, scan angle - no more than +/- 25 degrees, side overlay is not less than 30 percent. The comparison between the 2007 and 2018 LiDAR data is shown in Table 1.5.1.

Table 1.5.1. Comparison of LiDAR data technical characteristics

Year of LiDAR data / Characteristics	2007	2018
File format	*.xyz (ASCII)	*.LAS (v1.2)
Classification	Separate files (4 classes)	10 classes
Number of points (1m^2)	4	12
Colorization	-	+
Horizontal mean square error	No more 30 cm	No more 30 cm
Vertical mean square error	No more 15 cm	No more 10 cm
Scanning bands overlapping	30 %	30 %
Aerial mapping date	2009	2018 (not yet produced)

2. 3D CITY MODELLING

The digital three-dimensional city model generally describes the geometry of the city's environment. Most commonly understood, the 3D city model combines information collections with certain objects that can be used in urban environments, expressed in three dimensional geometry (e.g., urban projects, surfaces, vegetation, and 3D construction models). These patterns are usually created through 3D reconstruction and data integration, for example, combining photogrammetry or laser scan data with Geographic Information System (GIS) data such as building tracks, unique ID numbers, or addresses.

3D city models can be used in many areas for tasks related to planning, environmental analysis, transportation, service of communication companies, risk management, or simply visualization of planned objects and decision-making. In addition, 3D city patterns can be seen as a factor in a smart city connecting the user to a city environment system or service platform. Together with advanced 3D data acquisition systems and applications, 3D city patterns have become routine resources for city geospatial data.

3D models are used in professional GIS tools or virtual globes. GIS platforms are typically designed for professional users, such as city planners, architects, etc. The virtual world has 3D spatial viewers, where the user can move freely and change the viewing angle or zooming (e.g., Google Earth, ArcGIS Online). Without the need for special software, browser-based virtual globes are most commonly used to target public audiences.

This 3D city modelling project was made using the 2018 LiDAR data. The analysis focuses on the central area of the city, due to the variety of building types and their high elevation, relief changes, and other specific features of the urban environment. The analysis of the territory of Vilnius city consists of four LAS data files measuring 0.5 km x 0.5 km (Fig. 4). The main topics analysed are digital terrain model generation, acquisition of 3D geometry of buildings, recognition of road infrastructure, extraction of individual trees, and publication of results on the internet.

The products used by *ESRI* are *ArcGIS Pro* and *CityEngine*. *ArcGIS Pro* is a professional 2D and 3D mapping tool with intuitive user interface for visualization, analysis, image and data processing as well as integration. *CityEngine* is an advanced 3D simulation software designed to create a vast, interactive and visual urban environment. With this software, mass modelling is done much faster than traditional methods.

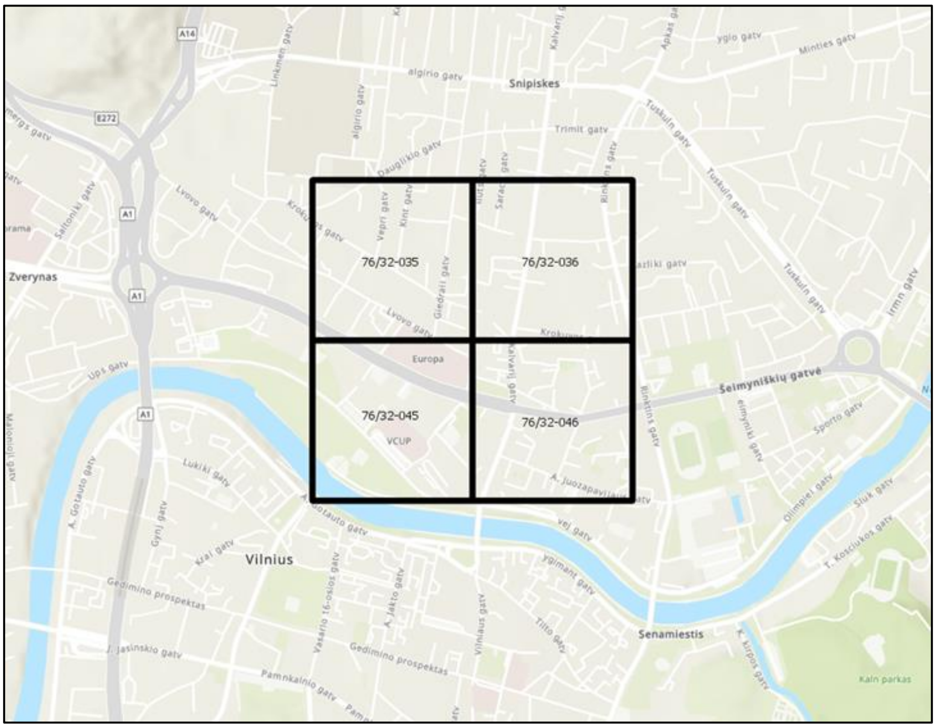


Figure 4. Case study area

2.1. Building extraction

Manual modelling of buildings is not timely and financially effective, so automated and semi-automatic processes are used to obtain results. Following the sequence of actions (Fig. 5), get 3D buildings using ArcGIS platform.

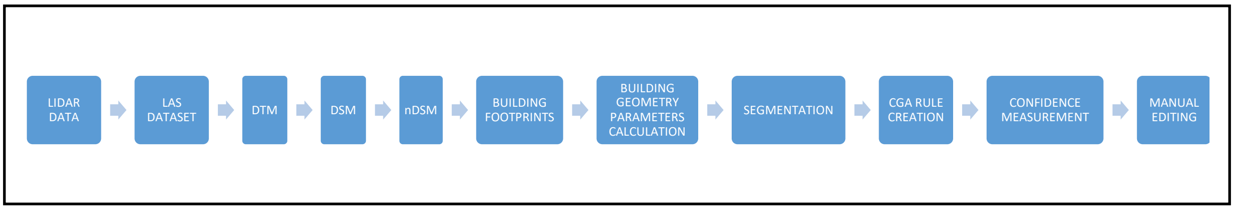


Figure 5. Flowchart of 3D building extraction

2.1.1. Creating a LAS dataset

A LAS dataset (.lasd) is a stand-alone file that stores reference to one or more LAS files on disk (What is LAS dataset? n.d.). A LAS dataset can also store reference to feature classes containing surface constraints which could be breaklines, water polygons, area boundaries, or any other type of surface feature enforced in the LAS dataset. There may be two additional files associated with an LAS database: LAS auxiliary file (.lasx), and projection file (.prj) (Create LAS Dataset n.d.). A .lasx file is created when statistics are calculated for any LAS file in an LAS dataset. The .lasx file provides a spatial index structure that helps improve the performance of an LAS dataset. If LAS files do not have a spatial reference or have an incorrect spatial reference defined in the header of the LAS file, a projection file (.prj) can be created for each LAS file. In that case, the new coordinate system information in the .prj file will take precedence over the spatial reference in the header section of the LAS file. Currently spatial reference system is defined already in the header of the LAS file, a projection file (.prj) is not needed, and additionally LAS dataset statistics were calculated.

Item	Category	Pt_Cnt	Percent	Z_Min	Z_Max	Intensity_Min	Intensity_Max	Synthetic_Pt_Cnt	Range_Min	Range_Max
First	Returns	46918623	82.94	86.6	263.11	<Null>	<Null>	<Null>	<Null>	<Null>
Second	Returns	6911577	12.22	86.67	261.43	<Null>	<Null>	<Null>	<Null>	<Null>
Third	Returns	2154711	3.81	87.26	253.57	<Null>	<Null>	<Null>	<Null>	<Null>
Fourth	Returns	521437	0.92	87.28	251.18	<Null>	<Null>	<Null>	<Null>	<Null>
Fifth	Returns	63659	0.11	87.38	241.62	<Null>	<Null>	<Null>	<Null>	<Null>
Last	Returns	46854517	82.83	86.6	262.13	<Null>	<Null>	<Null>	<Null>	<Null>
Single	Returns	39958447	70.64	86.6	262.13	<Null>	<Null>	<Null>	<Null>	<Null>
First_of_Many	Returns	6960176	12.3	87	263.11	<Null>	<Null>	<Null>	<Null>	<Null>
Last_of_Many	Returns	6896070	12.19	86.67	261.43	<Null>	<Null>	<Null>	<Null>	<Null>
All	Returns	56570007	100	86.6	263.11	<Null>	<Null>	<Null>	<Null>	<Null>
0_Created_Never_Cla...	ClassCodes	290052	0.51	92.49	153.38	0	65535	0	<Null>	<Null>
1_Unclassified	ClassCodes	2942438	5.2	87.39	263.11	0	65535	0	<Null>	<Null>
2_Ground	ClassCodes	23419742	41.4	87.4	115.08	0	65535	0	<Null>	<Null>
3_Low_Vegetation	ClassCodes	1703487	3.01	87.41	114.57	0	65535	0	<Null>	<Null>
4_Medium_Vegetation	ClassCodes	1788324	3.16	87.42	114.59	0	65535	0	<Null>	<Null>
5_High_Vegetation	ClassCodes	10029503	17.73	92.6	138.29	0	65535	0	<Null>	<Null>
6_Building	ClassCodes	12025965	21.26	90.67	194.81	0	65535	0	<Null>	<Null>
8_Model_Key_Point	ClassCodes	123	0	86.6	86.96	0	65535	0	<Null>	<Null>
9_Water	ClassCodes	94895	0.17	86.97	87.42	0	65535	0	<Null>	<Null>
11_Reserved	ClassCodes	4213826	7.45	91.69	110.71	0	65535	0	<Null>	<Null>
17_Reserved	ClassCodes	61652	0.11	95.36	97.7	0	65535	0	<Null>	<Null>
Return_No	Attributes	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	1	5
Intensity	Attributes	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	0	65535
Class_Code	Attributes	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	0	17
Scan_Angle	Attributes	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	-25	25
User_Data	Attributes	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	1	2
Point_Source	Attributes	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	25	83
Red	Attributes	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	4864	64000
Green	Attributes	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	4864	60416
Blue	Attributes	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	2560	55296
Model_Key	ClassFlags	53	0	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>
Synthetic	ClassFlags	0	0	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>
WithHeld	ClassFlags	0	0	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>
Overlap	ClassFlags	0	0	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>

Figure 6. Generated LAS dataset statistics of study area

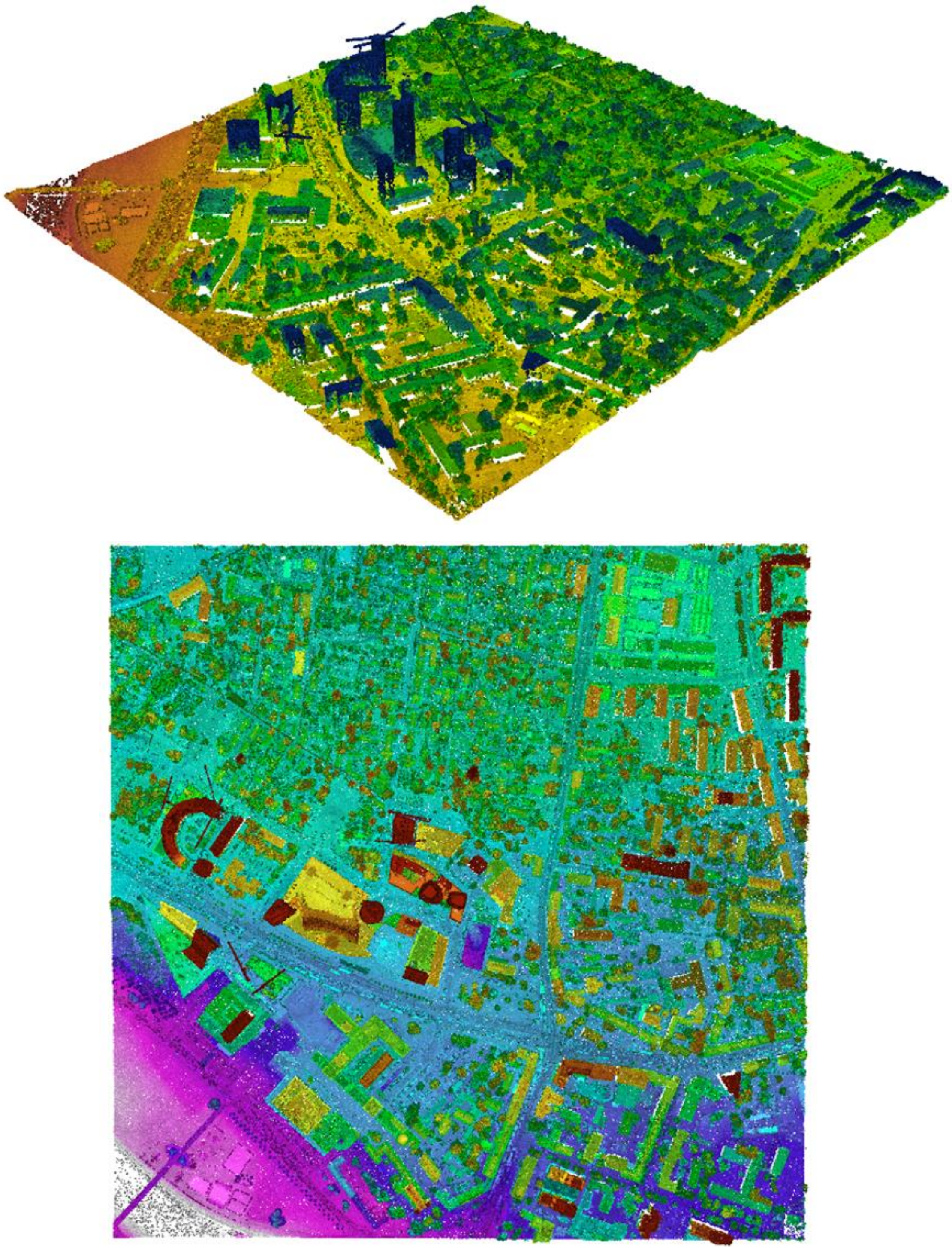


Figure 7. 2D and 3D view of LiDAR data (displayed by elevation and intensity)

2.1.2. DTM raster generation

A DTM is a 3D representation of a terrain's surface in digital form (Karan et al 2014). According GIS literature, the term DTM refers to the bare ground elevation points without any non-ground points (e.g., buildings and vegetation). Generating a DTM from LiDAR points initial step is to separate the points into ground and non-ground points. One of the most commonly used methods is to convert the LiDAR point data into a digital image by gridding the data points. The LiDAR points in each grid cell can have different elevations (z-values), for instance, the ground point elevation is lower than its neighbour object points. In the case of hitting the ground surface, the minimum z-value of the LiDAR points in a grid cell reflects the elevation of the bare ground.

Because a LiDAR point cloud is irregularly distributed, the smaller grid size results in more void areas (i.e., cells with no data point) within the model. Karan et al pays attention that to have a desired output, point density and number of LiDAR points should also be taken into account.

Interpolation is the procedure of predicting the value of attributes at unsampled site from measurements made at point locations within the same area or region (Zouaouid et al 2016).

As reported IDW interpolation method creates DEM with less overall error compared to other methods and the accuracy of DEM varies with the changes in terrain and land cover type. There are many opinions regarding on which interpolation method produces the highest accuracy LiDAR DEM. It can be seen that IDW and Kriging are the main competitors when it comes to producing DEM when comparing RMSEs.

For the generation of DTM in the ArcGIS Pro environment, *a LAS Dataset To Raster* geoprocessing tool was used. Main features described below:

- Classification codes used: *2 (Ground), 9 (Water), 11 (Road surface);*
- Return values: *1;*
- Value field: *Elevation;*
- Interpolation type: *Binning* (determines each output cell using the points that fall within its extent);
- Cell assignment: *IDW* (Inverse Distance Weighted interpolation);
- Void fill method: *Natural Neighbor* (finds the closest subset of input samples to a query point and applies weights to them based on proportionate areas to interpolate a value (Sibson 1981));
- Output data type: *Floating point* (32-bit floating point output raster);
- Sampling type: *Cell size* (Defines the cell size of the output raster);
- Sampling value: *0,2m* (Resolution of output raster);

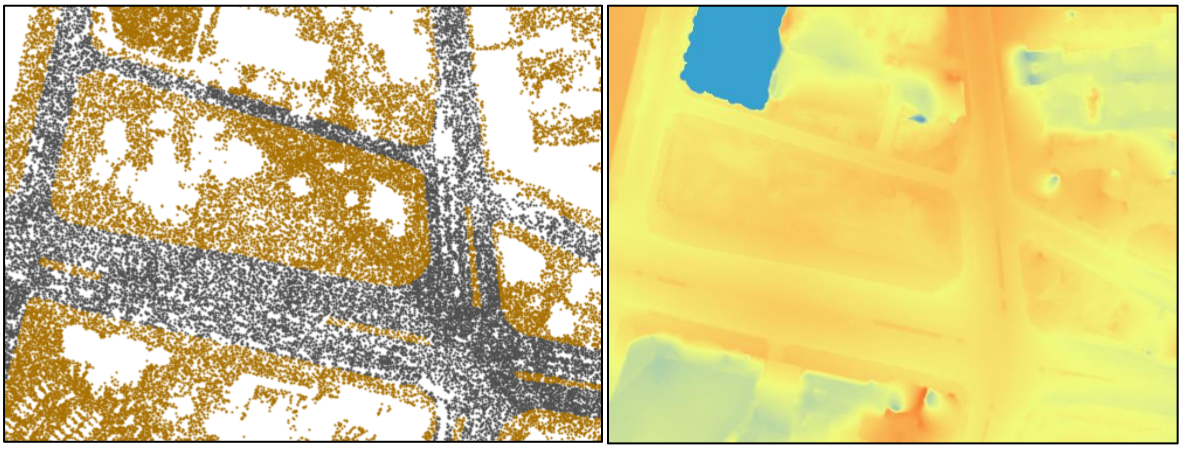


Figure 8. Comparing input LAS dataset (displayed by coding values) and output DTM raster (displayed by elevation values)

2.1.3. DSM raster generation

The DSM includes all objects on the earth's surface (Karan et al 2014). A terrain can be digitally modelled either by a series of regular grid points (altitude matrices) or as a triangulated irregular network (TIN). The former is a raster-based model consisting of a matrix of grids in which each grid contains a value representing surface elevation. A TIN is a vector-based model of the surface that is defined using a finite number of points in discrete formats in which each point is placed on the corner of triangles or rectangles by its coordinates (i.e., three-dimensional coordinates (x; y; z) or two-dimensional horizontal coordinates (x; y) and height (h)) (Wilson and Gallant 2000). Also, the data source (e.g., field surveying, aerial imagery, or satellite images) greatly affects the distribution of the reference points on the terrain.

For the generation of DSM with ArcGIS Pro, the same geoprocessing tool (LAS Dataset To Raster) was used. Main features described below:

- Classification codes used: *2 (Ground), 6 (Building), 9 (Water), 11 (Road surface);*
- Return values: *1;*
- Value field: *Elevation;*
- Interpolation type: *Binning* (determines each output cell using the points that fall within its extent);
- Cell assignment: *IDW* (Inverse Distance Weighted interpolation);
- Void fill method: *Natural Neighbor* (finds the closest subset of input samples to a query point and applies weights to them based on proportionate areas to interpolate a value (Sibson 1981));
- Output data type: *Floating point* (32-bit floating point output raster);
- Sampling type: *Cell size* (Defines the cell size of the output raster);
- Sampling value: *0,2m* (Resolution of output raster);

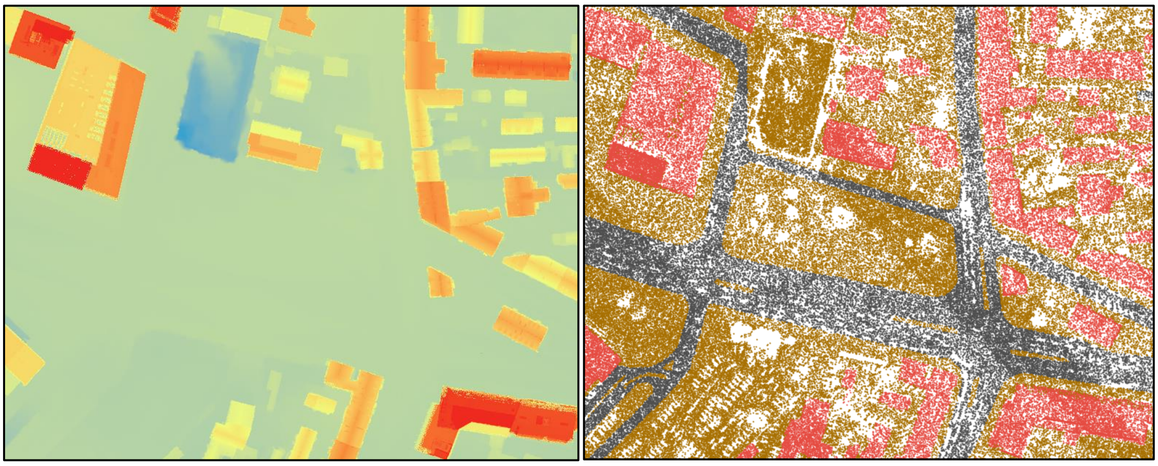


Figure 9. Comparing input LAS dataset (displayed by coding values) and output DSM raster (displayed by elevation values)

2.1.4. nDSM raster generation

Subtraction a DTM from a DSM of the same area produces nDSM (normalised digital surface model) (Mirosław-Świątek et al 2016) representing the heights of features from the surface. All features placed on zero-elevation surface.

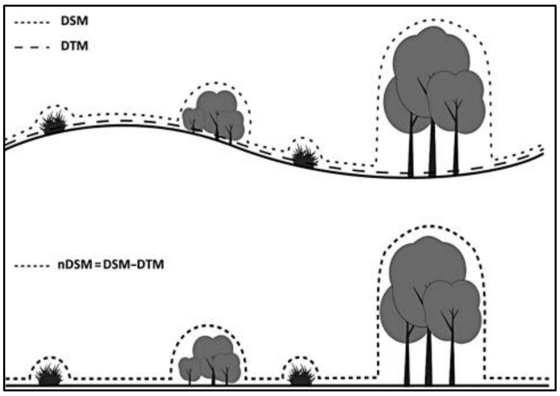


Figure 10. DSM, DTM, and nDSM (Mirosław-Świątek et al 2016)

Theoretically these features in nDSM should be zero or positive. However, it is very common to have negative values due to accuracy issues in DTM and DSM, interpolation process or other artefacts. In this case *Minus* geoprocessing tool was used for subtraction a DTM from a DSM. Generated nDSM negative values reaches up to -10 meters. To fix this issue *Raster Calculator* (mathematical computations tool) was used. Conditional calculation applied – if raster value < 0 applied value – 0, other values remains same. Newly generated nDSM raster elevation values varies from 0 to 149 meters.

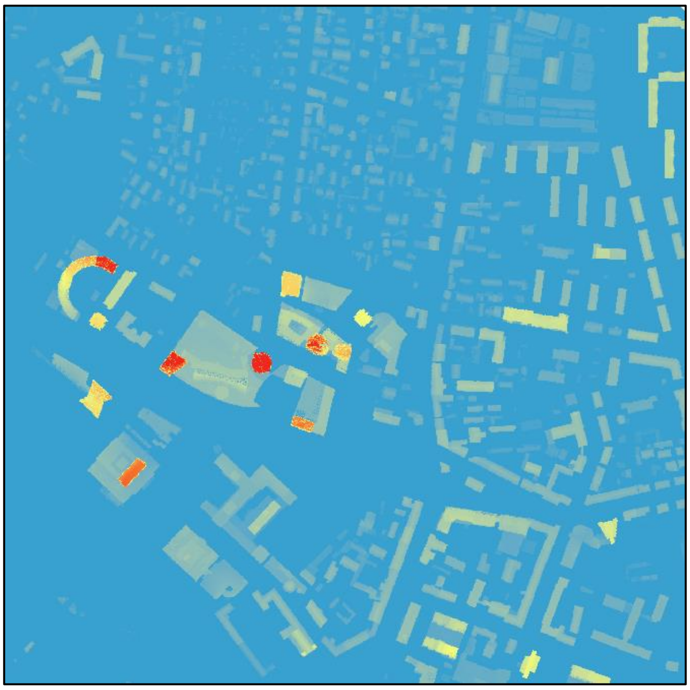


Figure 11. Generated nDSM raster (displayed by elevation)

2.1.5. Building footprints extraction

There are few approaches for building footprints extraction: 1. Using classified LiDAR data or 2. Using generated nDSM.

For generating building footprints from LiDAR data used *Las Dataset To Raster* geoprocessing tool. Main parameters described below:

- Classification codes used: *6 (Building)*
- Return values: *1;*
- Value field: *Elevation;*
- Interpolation type: *Binning* (determines each output cell using the points that fall within its extent);
- Cell assignment: *IDW* (Inverse Distance Weighted interpolation);
- Void fill method: *Simple* (Averages the values from data cells immediately surrounding a NoData cell to eliminate small voids);
- Output data type: *Floating point* (32-bit floating point output raster);
- Sampling type: *Cell size* (Defines the cell size of the output raster);
- Sampling value: *0,2m* (Resolution of output raster);

As a result we got floating point elevation raster which represents buildings. At the next step floating point raster values converted to single value with condition: if raster value < 1, set NoData values, else values transformed into a value of 1.

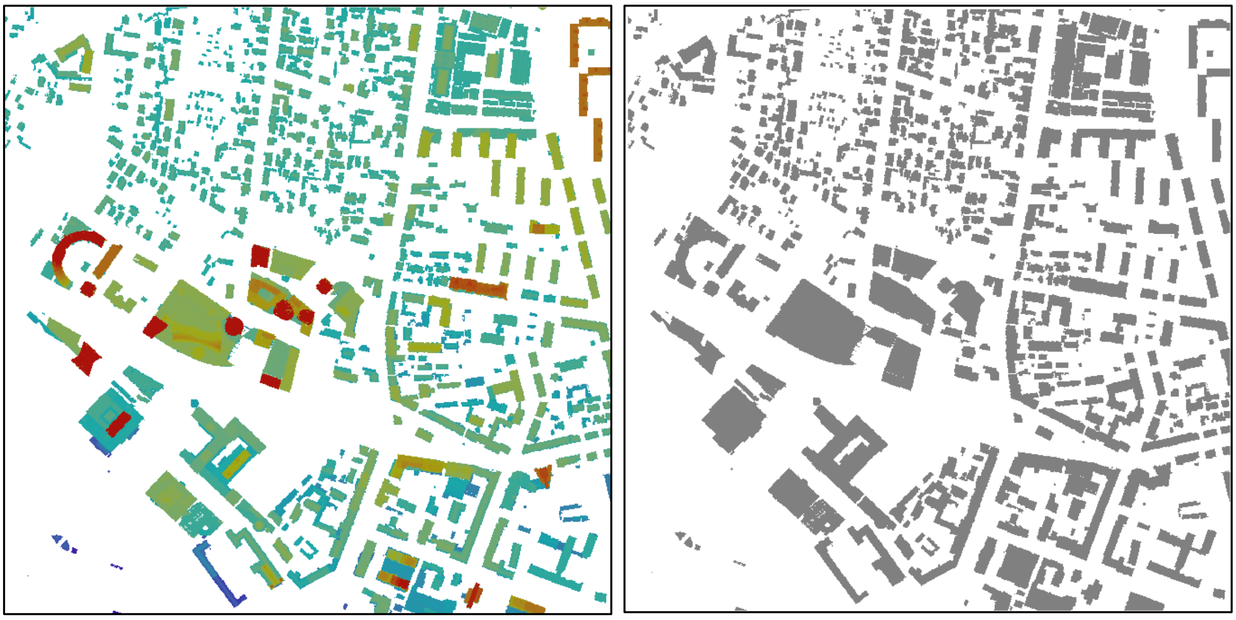


Figure 12. Rasters representing building footprints. Elevation raster (left), single value raster (right)

To convert raster dataset to polygon features – *Raster to Polygon* geoprocessing tool used, optionally polygons were simplified.

Figure 13. Extracted building polygon features

Second method which was used for results comparison is building footprint extraction of nDSM. To get results used *Raster Calculator* function. For input used nDSM and applied conditional rule: if raster values < 2 (buildings of which height is less than 2 meters to remove small buildings and other artefacts which may contain errors) set *NoData*, other remaining values set to 1. For final result used *Raster To Polygon* function to convert raster to polygon features. Results showed that some polygons are missing which was extracted from nDSM, at some cases building footprint geometry spatially inaccurate (Fig 14). For the further development of the project, building footprints used from extraction of classified LiDAR data.

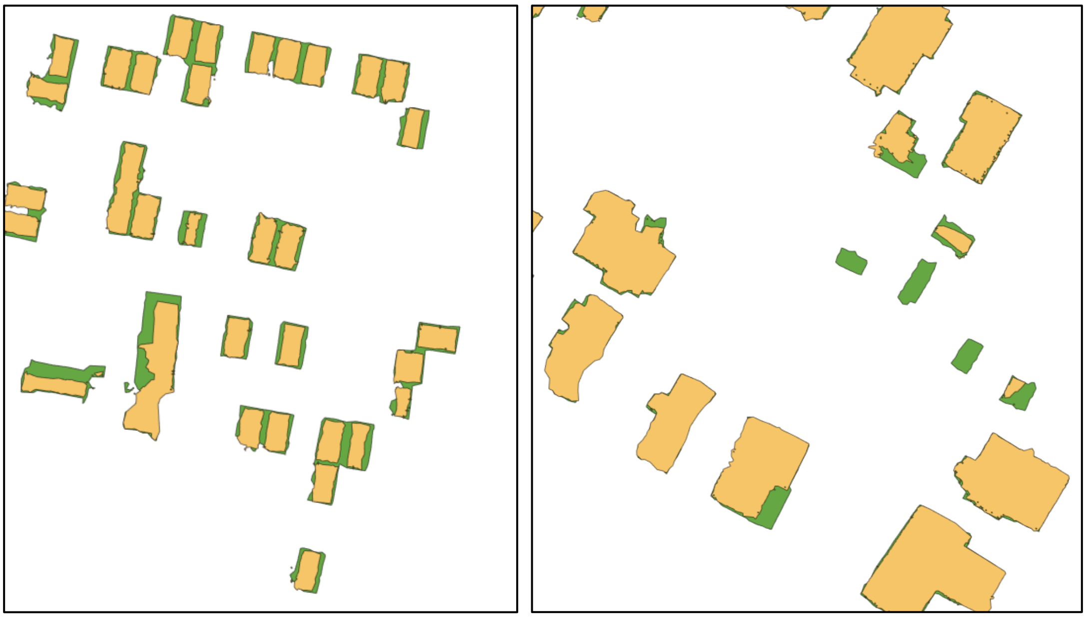


Figure 14. Comparison of extracted building footprints (orange – from nDSM, green – from classified LiDAR)

2.1.6. Calculation of building geometry parameters

The roof form extraction process uses LiDAR derived raster data and polygon building footprints to find both flat and sloped planar areas within an area of a building. The tool then estimates a standard architectural form for the roof based on the attributes collected from these planar surfaces. These attributes will then be used to procedurally create these features in 3D. The stand-alone script was created to combine number of calculations and tools into one.

Script input requires previously generated DTM, DSM, nDSM rasters, building polygon features, minimum of sloped and flat roof areas, and minimum building height.

First script step uses raster calculator which uses nDSM raster and removes values of which are lower than minimum building height value. In this case used value of 3 meters and nDSM values which are lower than 3 meters were set to null values.

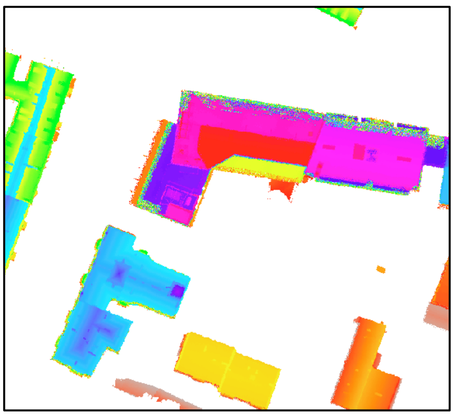


Figure 15. nDSM with elevation threshold

Next used *Slope* geoprocessing tool. This tool identifies the slope (gradient or steepness) from each cell of a raster. As an input used generated nDSM with building height threshold. Conditional rule was also applied to eliminate minimum and maximum values of sloped roof. Roof areas which slope angles were outside determined limits (less than 10 and more than 60 degrees of slope angle) were set as 0 (flat roof area).

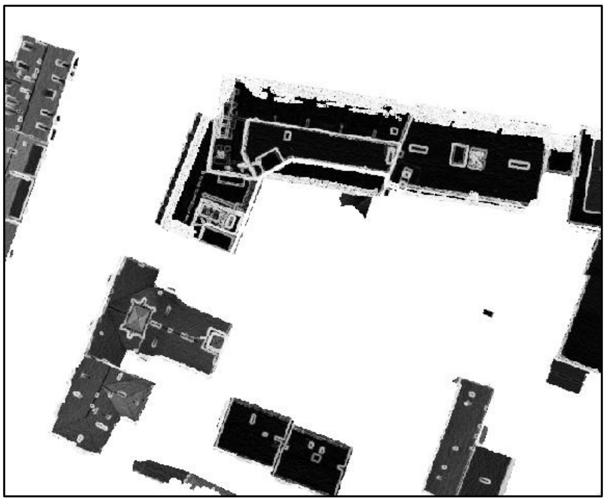


Figure 16. Slope raster of building areas

Aspect geoprocessing tool was used to identify the compass direction that the downhill slope faces for each location. Generated aspect sloped roof areas were reclassified – created simplified aspect raster which contains 9 classes (aspect values from 110 to 155 set to class 3, from 155 to 200 set to 4 and etc.). Reclassified aspect raster converted to polygon feature class with *Raster To Polygon* geoprocessing tool. Finally *Intersect* tool was used to calculate the geometric intersection of building footprints feature class and roof aspect polygon feature class. Attribute values from the input feature classes were copied to the output feature class. *Select* function was used to select sloped roof polygons by minimum area values which was determined at the beginning of the script.

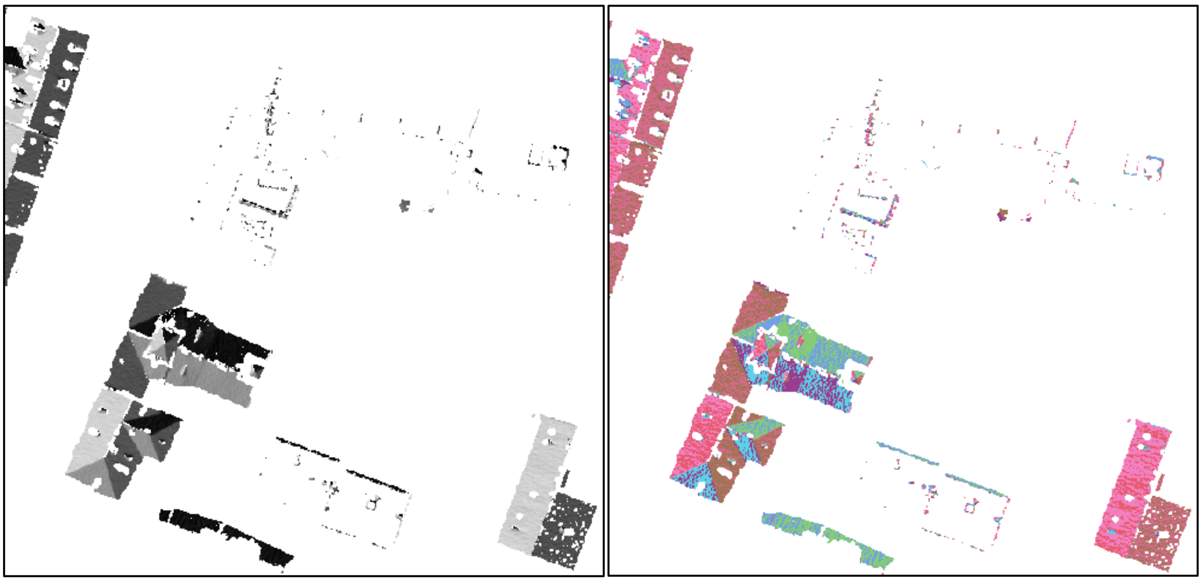


Figure 17. Aspect rasters (at left - general aspect raster, at right – reclassified aspect raster)

For calculation of building base elevation values *Zonal Statistics* was used. This tool calculates statistics on values of a raster within the zones of another dataset. As an input used building footprint polygon features and DTM raster. Statistics type was set to *minimum*, and as result get masked elevation raster which contains single elevation value for each building footprint. *Feature To Point* used to get building footprints as center points and *Add Surface Information* used to calculate elevation values for each point which represents center of building polygon.

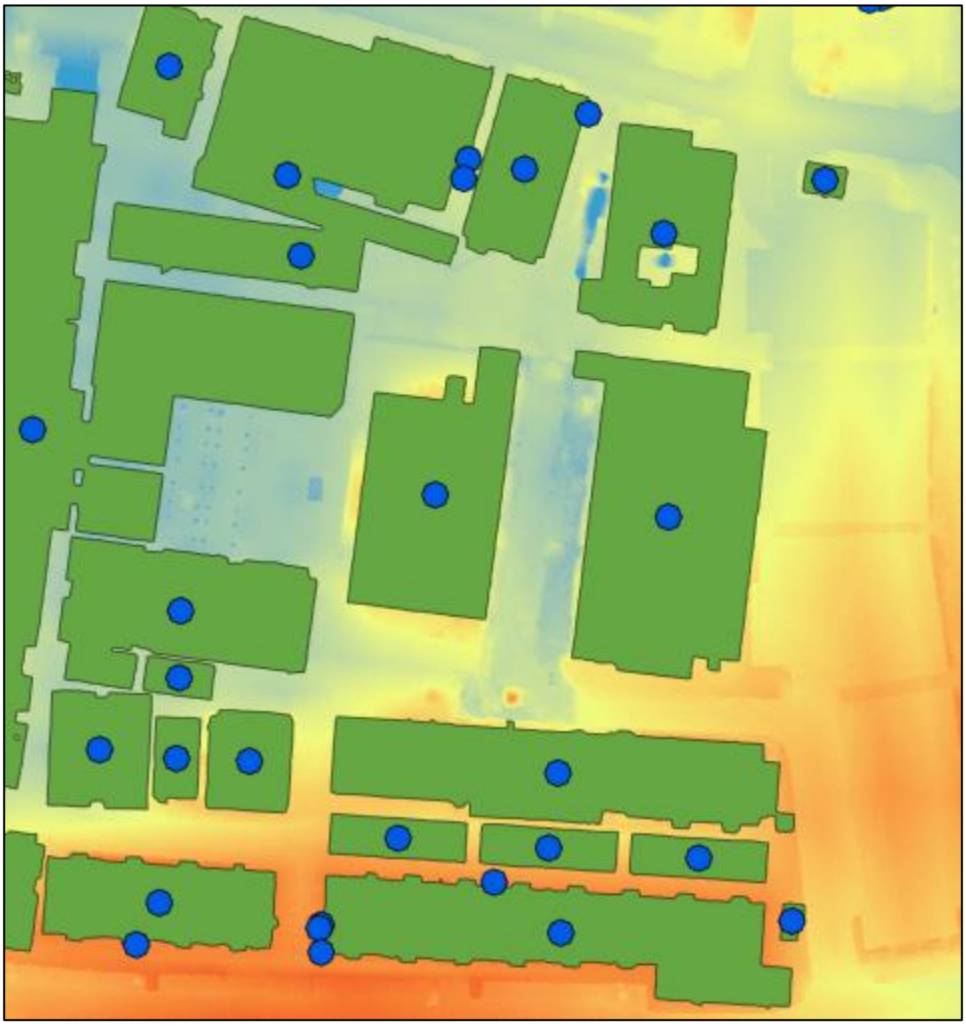


Figure 18. Building base elevation calculation (DTM in background, green – building footprints, blue circles – building centers)

Calculation of eave height values for sloped roofs are partly similar as in previous step. *Select* function was used for extraction of largest plane of the same building by area parameter, *Feature To Point* for converting largest building slope plane to point feature. All features except largest sloped plane areas were masked out of the DSM and *Zonal Statistics* generated raster with minimum value of DSM at largest sloped roof plane area. *Add Surface Information* used to calculate Z value.

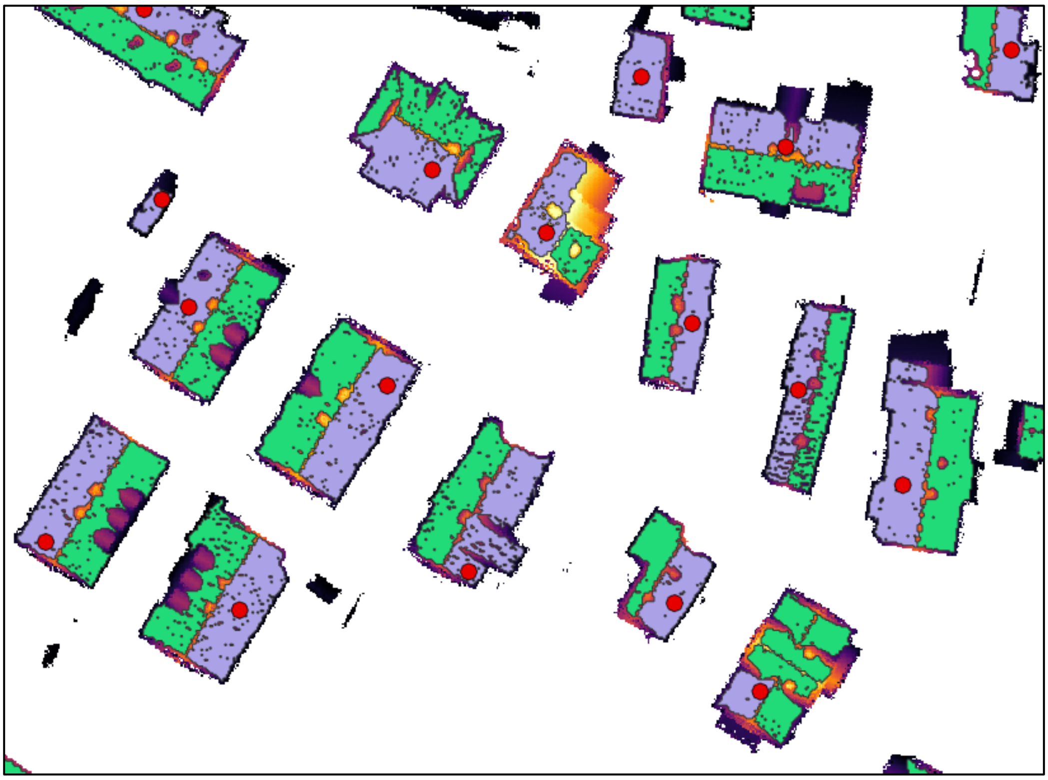


Figure 19. Eave height calculation (in background clipped building DSM raster, green – sloped roof planes, purple – largest roof slope plane, red circles – largest roof plane points)

In the calculation of flat roof parameters, the slope raster was mainly used. Roof slope raster converted to raster of 2 values using condition in *Raster Calculator*. If slope value less than threshold (10 degrees) convert values into 1, other values (not flat roof areas) remains as 0. Zero values converted to null values using *SetNull* function in raster calculator. Flat roof areas converted to polygon features using *Raster To Polygon* geoprocessing tool. *Intersect* tool removed artefacts outside building polygons and joined building attributes. *Select* tool used for remove small flat roof areas which are lower than given threshold, at this case used 250m^2 of minimum flat roof area.

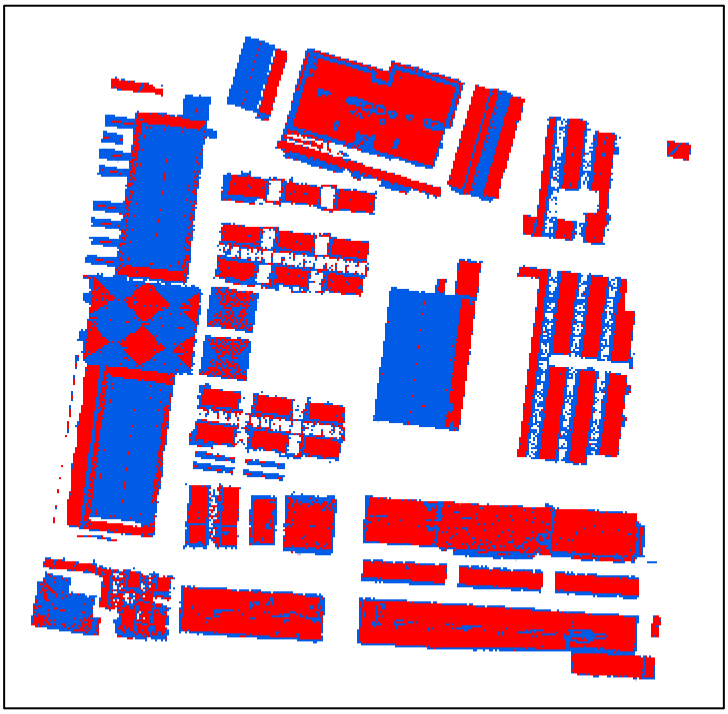


Figure 20. Flat (red colour) and not flat (blue colour) building areas

For final corrections all planes were merged together (aggregated) using *Dissolve* function. *Zonal statistics* clipped DSM raster by merged roof planes as *maximum* value within an area. *Feature To Point* converted planes into points and *Add Surface Information* calculated maximum values representing building heigh. Mathematical computations with polygon vertices and edges used to calculate additional parameters which describes roof orientation along building footprint.

2.1.7. Segmentation of building roof parts

Some buildings may have different roof types and roof parts. To represent these buildings more accurately, used a script which contains of few geoprocessing tools inside and helps automatically split building footprints into multiple roof segments.

At first *Select* tool copies building footprint areas where the roof type of the building attribute table described as flat. Using raster functions, raster values of DSM which are outside the building footprint polygon set to null.

Segment mean shift used to group DSM adjacent pixels that have similar spectral characteristics into segments. Segmentation described by three input parameters: Spectral detail, spatial detail, and minimum segment size in pixels. Spectral detail sets the level of importance to the spectral differences of features in imagery. Valid values range from 1 to 20 (in this case used value of 20). A higher value is appropriate for features to classify separately but have somewhat similar spectral characteristics. Smaller values create spectrally smoother outputs. Spatial detail sets the level of importance to the proximity between features in imagery. Valid values ranges also from 1 to 20 (used value of 20). A higher value is appropriate for features of interest are small and clustered together. Smaller values creates spatially smoother outputs. Minimum segment size in pixels merges segments which are smaller than input size with their best fitting neighbour segment. At this case used value of 500. *Raster To Polygon* used to convert segmented raster into polygon features.

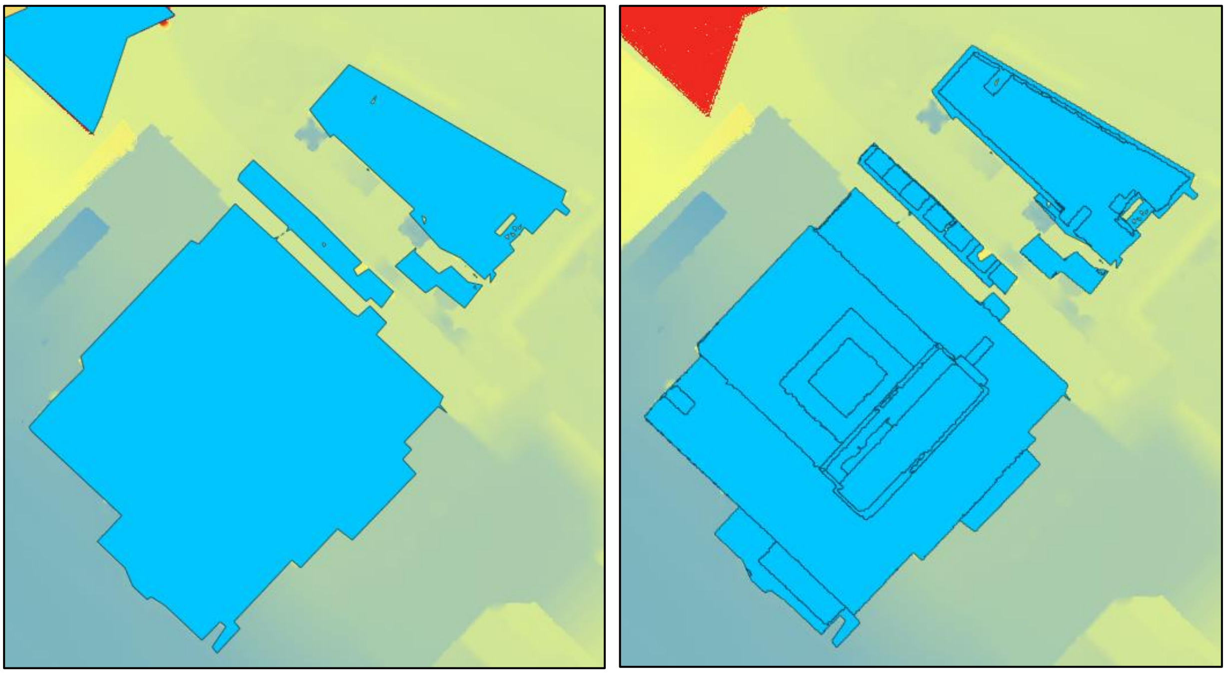


Figure 21. Input building footprint polygons and intermediate segmentation results

Segmented polygons which area was less than threshold (500 pixels) were removed. *Regularize Adjacent Building Footprint* geoprocessing tool used (method – right angles and diagonals and precision of 0.25) for regularization of building footprints that have common boundaries. As a result get segmented roof polygons for later processing.

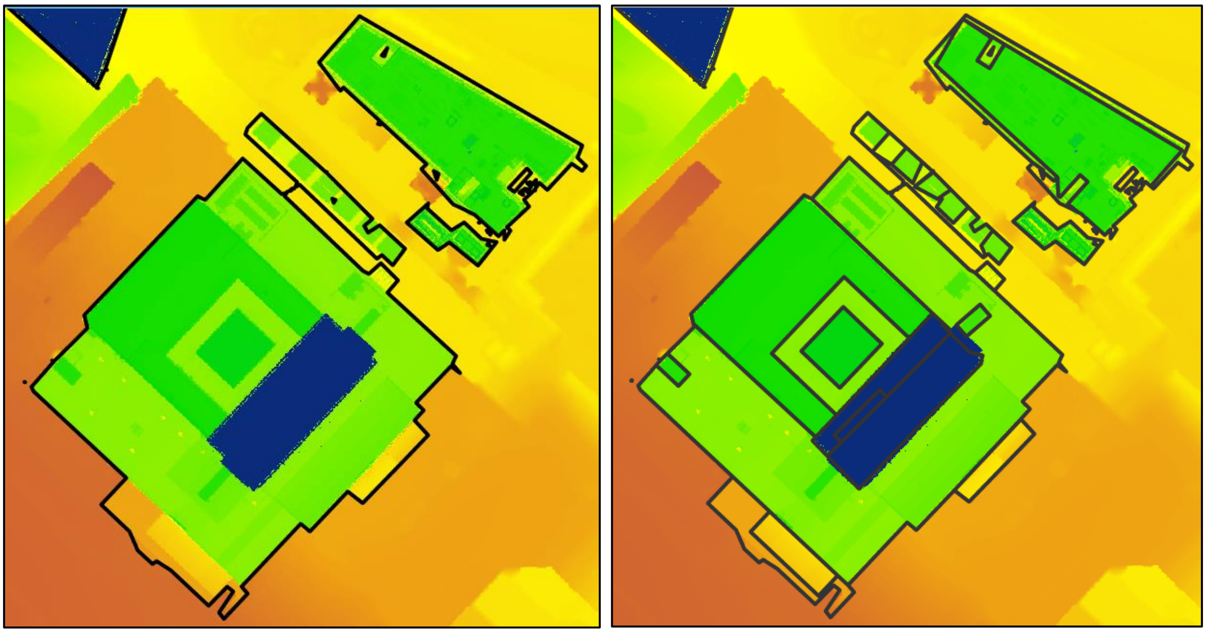


Figure 22. Input building footprints (left) and segmented building footprints (right)

2.1.8. CGA rule creation for buildings

CityEngine is a standalone computing application for designing, planning and modelling urban environments in 3D (Župan et al 2018). It was designed to help professional users in GIS, CAD and other 3D systems to:

- Quickly generate 3D cities from the existing 2D GIS data;
- Realize conceptual design in 3D based on the GIS data and procedural rules;
- Model virtual 3D urban environments for simulation and entertainment.

The concept of CityEngine is the "procedural" approach to modelling. The computer is a code that represents a number of commands - geometric modelling. Instead of "classical" user intervention, i.e. manual interaction with the model and 3D geometry modelling, the task is described as "abstract" in the "rule" file. Commands available in the CGA shape grammar format of CityEngine such as "extrude", "split" or "texture" are widely-known commands in most 3D applications so that each user can easily adopt and create complex architectural shapes in a relatively short period of time (ESRI CityEngine 2018).

One "rule" file can be used to generate many 3D models. For example, the rule may use information about attributes stored in the GIS data such as the number of building height, roof type, wall material type, etc. - to generate a range of alternative 3D models that accurately represent the properties of each feature. The more attributes are, the more accurate the model can be. The 3D model is nothing more than a 3D object that is the result of an extruded 2D form according to the rules defined in the CGA "rule" file. The origin of these 2D forms is variable:

- It can be imported from Esri Shapefiles or File Geodatabase;
- It can be manually build in CityEngine;
- It can be generated through CGA rules

Three-dimensional objects represented in CityEngine do not have to be all generated within CityEngine. CityEngine supports the variety of data formats for import and export (Fig. 23). CityEngine supports geographic data, 2D data, 3D data (polygonal networks) and Python scripting as input data types and offers a wide variety of formats for exporting 3D data.

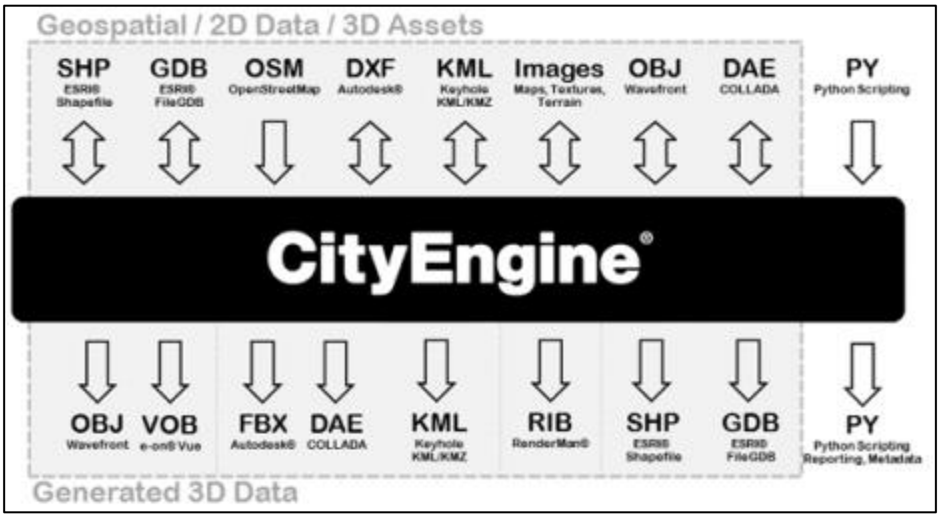


Figure 23. Supported CityEngine data formats (Ribeiro et al 2014)

However, only the geometry of objects from the multiple Esri Shapefiles or File Geodatabase can be edited and later upgraded to the original files. 3D models are 3D objects generated in CityEngine through procedural modelling. The remaining objects (3D or 2D) are called Shapes. By changing the urban design (i.e. by regenerating the "regenerate") commands, they are automatically applied and updated (Esri CityEngine 2018). The other advantage of procedural modelling is dynamically editing. For example, dynamically changing attribute values – generated model changes immediately. The lack of procedural modelling compared to manual modelling techniques is less "user-friendliness". The knowledge of CGA scripting for writing proper files makes the process of procedural modelling more complex at the beginning if the user has no experience with it. After organizing all data 3D modelling process is started. During 3D modelling process in CityEngine tutorials, materials and education videos were used.

For beginning of CGA procedural modelling a new City Engine project and scene was created with necessary coordinate system.

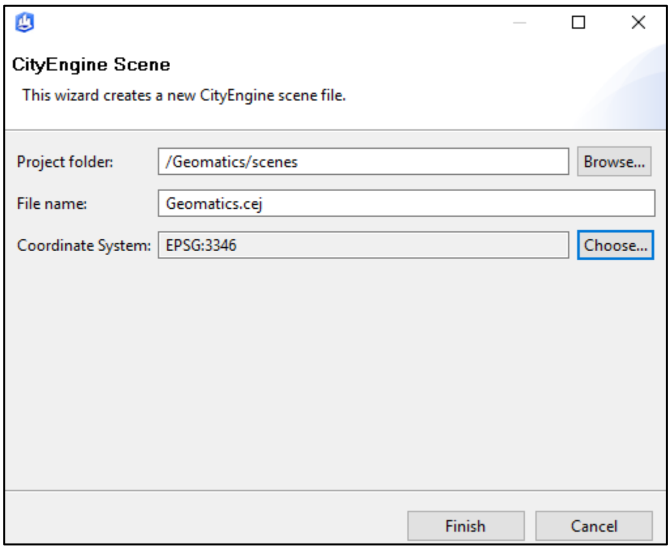


Figure 24. CityEngine scene creation wizard

As soon as scene created, previously generated shapefiles with their building and roof parameters were imported.

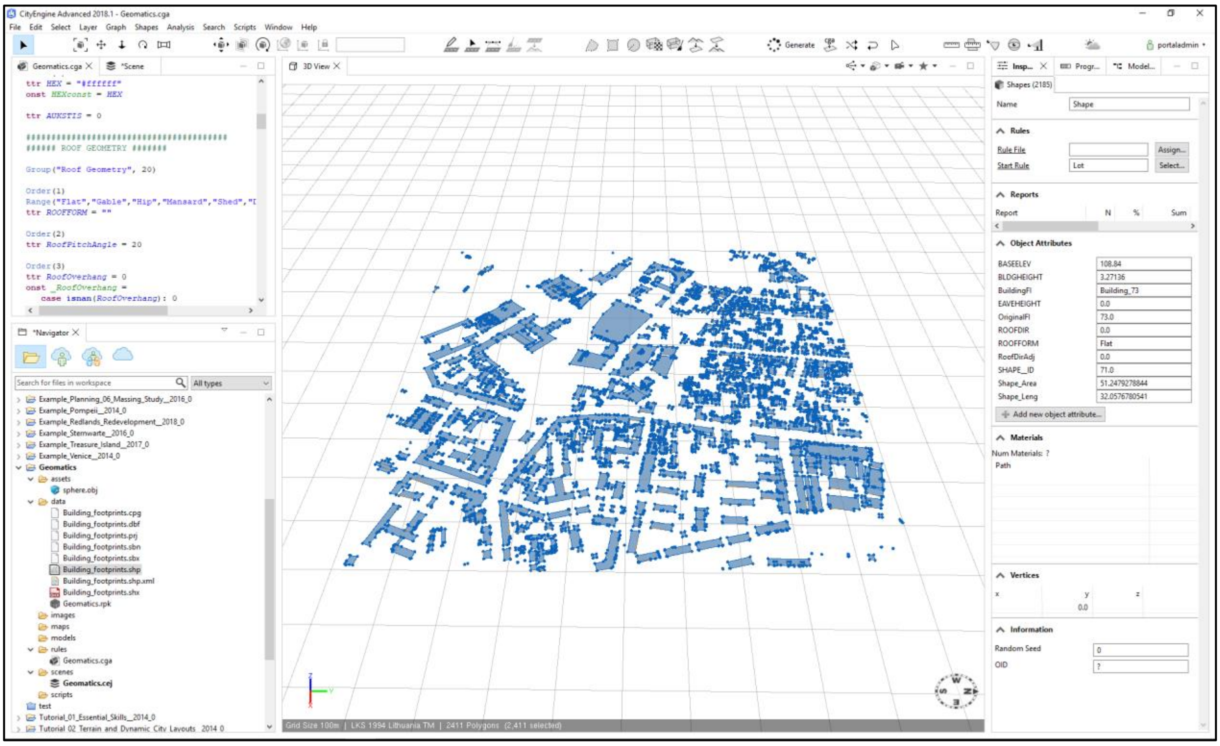


Figure 25. CityEngine main window

At next step new CGA shape grammar rule file created (*New-CityEngine-CGA Rule File*) and editor is opened. In the CGA editor, grammar authoring can now be started by defining the building parameters: building height, eave height, base elevation and etc.

```
/* Attributes *************************************/

########################################
####### BUILDING GEOMETRY #######
@Group("Building Geometry",10)

@Order(2)
attr BLDGHEIGHT = 0
const _BLDGHEIGHT =
    case isnan(BLDGHEIGHT): 0
    else: BLDGHEIGHT

@Order(3)
attr EAVEHEIGHT = 0
const _EAVEHEIGHT =
    case isnan(EAVEHEIGHT): 0
    else: EAVEHEIGHT

@Order(4)
attr BASEELEV = 0
const _BASEELEVATION =
    case isnan(BASEELEV): 0
    else: BASEELEV
```

Figure 26. Defining attributes

After attributes were defined the creation of the building starts. The first rule is called *Lot*. The mass model is created with the extrude operation as follows:

```
Extrusion -->
    cleanupGeometry(all,0.5)
    alignScopeToAxes(y)
    s('1,0,'1)
    extrude(facadeHeight)
    Building

LotInner --> NIL
```

Figure 27. Extrusion rule example

A mass model was divided into its facades by applying the component split. This rule splits the shape named Building, the mass model, into its faces by applying a component split. This results in a front shape, several side shapes as facades, and a roof shape.

```
# split the building geometry into its facade components
Building -->
        comp(f) { front : FrontFacade | side : SideFacade | top : Roof }
```

Figure 28. Building splitting into parts

After building was splitter into components, facades were set as solid components without any additional geometry.

```
# the front facade is subdivided into one front groundfloor
# and upper floors
FrontFacade -->
    setupProjection(0, scope.xy, scope.sx, scope.sy)
    split(y){{floorHeight : Floor}* }

Wall -->
projectUV(2)

SolidWall -->
    s('1,'1,0.4)
    t(0,0,-0.4)
    primitiveCube()
    projectUV(0)

SideFacade -->

    setupProjection(0, scope.xy, scope.sx, scope.sy)
    split(y){{floorHeight : Floor}* }

Floor -->
split(y) {~floorHeight - FloorLineThicknessconst : WallRoofLinesY | FloorLineThicknessconst : color( FloorLineColor ) FloorDivider.}

WallRoofLinesY -->
    projectUV(0)
```

Figure 29. Describing facades

For the start of roof modelling at first conditional rule was applied according roof type parameters.

```
Roof -->
    case _EAVEHEIGHT <=0 && ROOFFORM != "Flat": RoofPlanes # Same as flat roof.
    case ROOFFORM == "Flat" : RoofPlanes
    case ROOFFORM == "Gable" : roofGable(RoofPitchAngle, _RoofOverhang , _RoofOverhang , false, _RoofDirAdjust) RoofForm
    case ROOFFORM == "Hip" : roofHip(RoofPitchAngle, _RoofOverhang , true) RoofForm
    case ROOFFORM == "Mansard" : envelope(normal, roofHeight, 0, MansardAngle, 0, MansardAngle, 0, MansardAngle, 0, MansardAngle) RoofPlanes
    case ROOFFORM == "Shed" : RoofShedFix
    case ROOFFORM == "Dome" : DomeRoof
    case ROOFFORM == "Vault" : VaultRoof
    case ROOFFORM == "Spherical" : SphericalRoof
    else : RoofPlanes

    RoofForm -->
    alignScopeToAxes(y)
    s('1,roofHeight,'1)
    RoofPlanes
```

Figure 30. Defining roof form

For each roof type different rules were set. Some roof types (*Flat*, *Shed*, *Gabled*, *Hip*, *Pyramid*) are already supported by CityEngine without having use in-depth cga rules, however, for some types *mansard*, *dome*, *vault*, were created with more calculations and parameters.

```
RoofPlanes -->
		comp(f){bottom: NIL | all: RoofLinesY}
RoofLinesY -->
		offset(-FloorLineThickness/2)
		comp(f){border: color(FloorLineColor) X. | inside: Blank-
roofwithlines00}
Blankroofwithlines00 -->
color(HEXconst)
		setupProjection(0, scope.xy, 10, 10)
		projectUV(0)

@Hidden
attr eaveElevation = 0
@Hidden
attr tempRoofHeight = 0

RoofShedFix -->
		set(eaveElevation, scope.elevation)
		roofShed(10,_RoofDirAdjust)
		comp(f){side: NIL | bottom: NIL | all: RoofShedFixPart2}

RoofShedFixPart2 -->
		alignScopeToAxes(y)
		t(0, (eaveElevation - scope.elevation), 0)
		set(tempRoofHeight,scope.sy)
		extrude(world.y, -tempRoofHeight)
		split(y){'0.5: alignScopeToAxes(y) RoofForm | '0.5: NIL}
calcSegmentHt(n) = roofHeight * (cos(n*curvedAngleResolution) -
cos((n+1)*curvedAngleResolution))
const curvedAngleResolution = 10

VaultRoof -->
		VaultRoof(90/curvedAngleResolution-1)
VaultRoof(n) -->
		case n > 0: roofGable(n*curvedAngleResolution,0,0,false,0)
					split(y){ (calcSegmentHt(n)): RoofMass(n!=1)

					 comp(f){ bottom: NIL | horizontal: Vault-
Roof(n-1) } }
		else: NIL

DomeRoof -->
		DomeRoof(90/curvedAngleResolution-1)

DomeRoof(n) -->
		case n > 0: roofHip(n*curvedAngleResolution)
					split(y){ (calcSegmentHt(n)): RoofMass(n!=1)

					 comp(f){ bottom: NIL | horizontal: Do-
meRoof(n-1) } }
		else: NIL

RoofMass(removeBottomAndTop) -->
		case removeBottomAndTop:
				comp(f){ horizontal: NIL | vertical: RoofPlanes | all:
RoofPlanes }
		else: # remove only the bottom face
				comp(f){ bottom: NIL | vertical: RoofPlanes | all:
RoofPlanes }

@Hidden
attr smallerOfScopeXZ = 0
@Hidden
attr radius = 0

SphericalRoof -->
		alignScopeToAxes(y)
		LeaveDomeBaseHere. # This just saves the dome base in place.
		innerRectangle(edge) {shape : SphereBase | remainder : NIL}
SphereBase -->
		alignScopeToGeometry(yUp, any, longest)
		set(smallerOfScopeXZ, case scope.sx < scope.sz: scope.sx else:
scope.sz)
		s(smallerOfScopeXZ, 0, smallerOfScopeXZ)
		center(xz)
		i("builtin:cube")
		comp(f) {top: t(0,0,-scope.sy) Tank}
Tank -->
		alignScopeToAxes(y)
		i("sphere.obj")
		cleanupGeometry(edges, 0)
		set(radius, smallerOfScopeXZ / 2)
		alignScopeToGeometry(zUp, any, world.lowest)
		DomeTaper(1)

TankTopRadius = radius
TankTopHeight = roofHeight
steps = 10
splits = steps - 1
r = TankTopRadius
h = TankTopHeight
HH = sqrt(h*h + r*r)
H = HH / 2
R = 2*(H*H) / h
base = (R - h)
A = atan(r / base)
m = A / steps
AComp = 90 - A
a(i) = AComp + (m * i)
y(i) = R * sin(a(i))
x(i) = R * cos(a(i))
ylb(i) = y(i) - base
j(i) =
		case i == 0: 0
		else: ylb(i) - ylb(i-1)
k(i) = x(i-1) - x(i)
angle(i) = atan(j(i) / k(i))

DomeTaper(i) -->
		case i > steps:
				X.
		else:
				DomeLayer(angle(i),j(i))
				t(0,0,j(i))
				s(x(i)*2,x(i)*2,0)
				center(xy)
				DomeTaper(i+1)

DomeLayer(angle, j) -->
		roofPyramid(angle)
		split(y) {j: DomeFace.}

DomeFace -->
		offset(-0.06)
		comp(f) {border: color(0.6,0.6,0.6) X. | inside: X.}

End
```

Figure 31. Modelling rule for roof types

Created rule was applied for building footprint shapes to check if there are no mistakes in the same rule. Inspector window shows that attributes in rule and shapes connected and generation was applied.

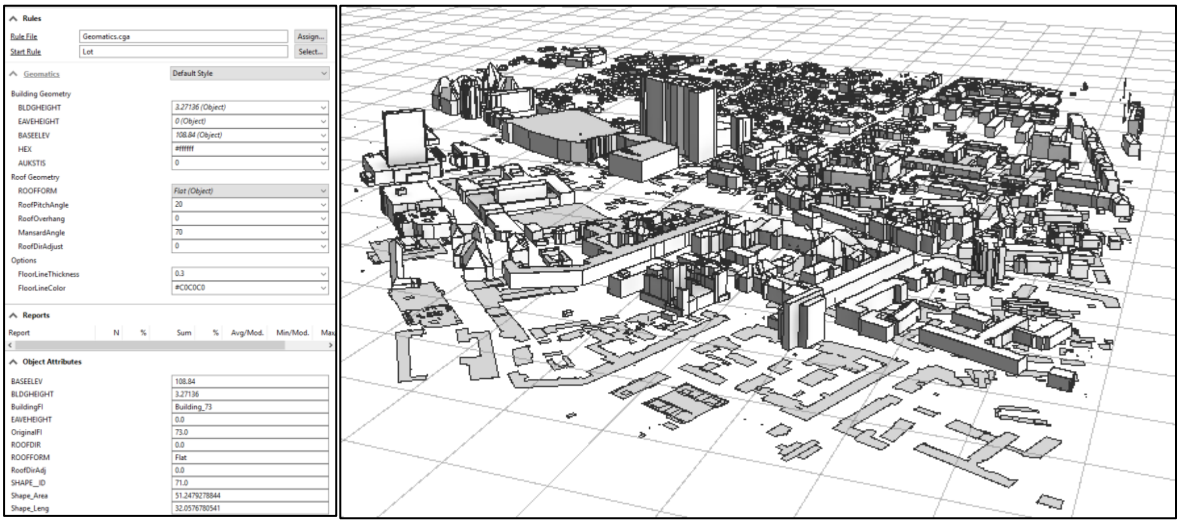


Figure 32. Inspection view and applied rule in CityEngine

The shape tree of a generated model interactively explored in model hierarchy window. Different part of model been analysed and reviewed.

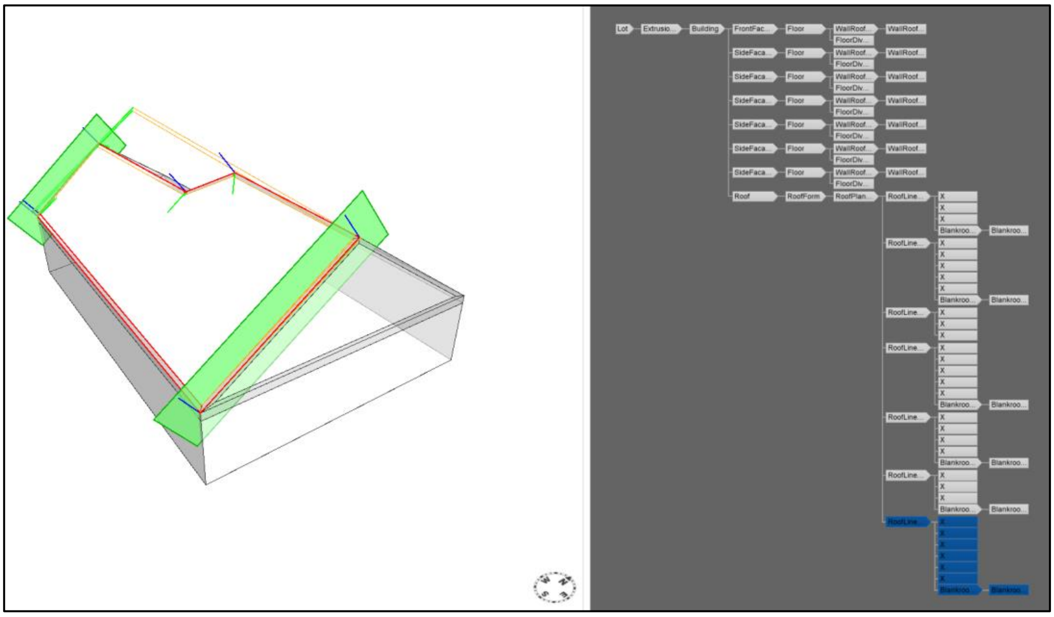


Figure 33. Individual model inspection

A 'rule package' (*.rpk) is a compressed package, containing compiled (binary) CGA rule files, plus all needed referenced assets and data. For finding the package content, the rules are analysed, to validate RPK for any errors or issues. If any issues are discovered, an error will be reported, after fixing errors, rule package can be exported.

2.1.9. Confidence measurement

Exported rule package applied in ArcGIS Pro environment by setting up property connections. Polygons according rule package set to 3D active layer.

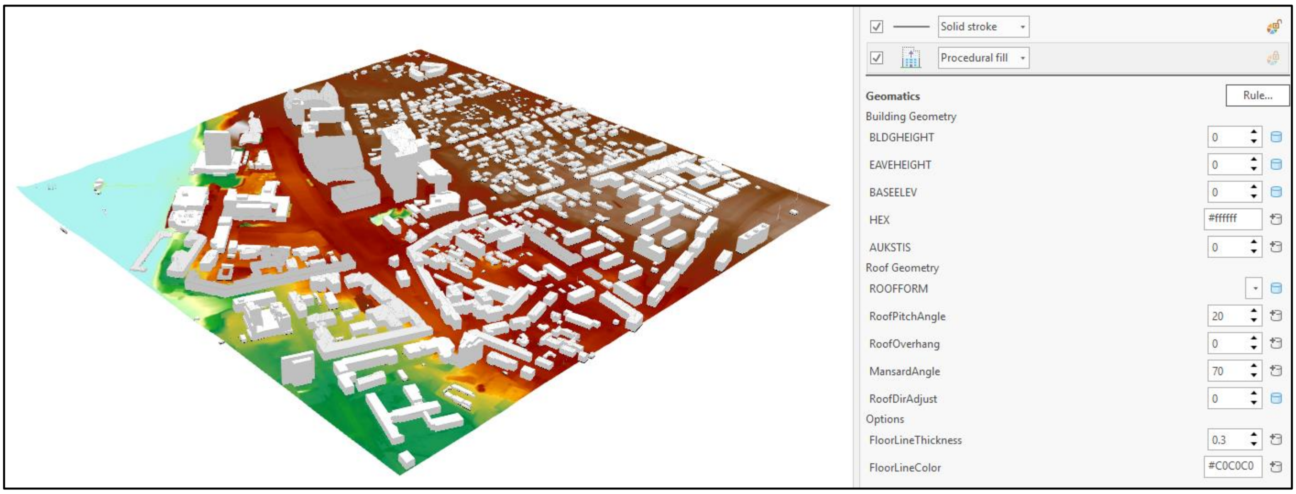


Figure 34. Active 3D building layer and attribute connections in ArcGIS Pro

Script of confidence measurement applied to perform a visual inspection of data. Script rasterizes building shells by *Multipatch to Raster* geoprocessing tool. This tool burn 3D building features (as multipatches) into a DTM. *Zonal statistics* applied to calculate mean values between original DSM and DTM raster with burned building footprints. Raster of RMSE converted into points and joined by unique building id with existing buildings. New attribute field created with RMSE values. Symbology of graduated colours applied to visually inspect extrusion quality.

RMSE field shows the root-mean-square-error of the generated building multipatch to the underlying surface model. The higher this number, the more likely that the roof-form extraction process encountered an error in classification. In general, a value of 1 meter or less is desirable, though this depends on the required application of the output features, and the resolution of the input data.

Figure 35. Building footprints visualisation by RMSE (from low to high (green to red))

2.1.10. Manual Editing

Visual comparison reveals that most of the buildings with high RMSE seem to have multiple parts with different heights or geometry errors caused by vegetation-covered roofs. The RMSE may have been caused by the building footprint not reflecting the actual extent of the building. Additionally, the footprint appears to be misaligned with the imagery.

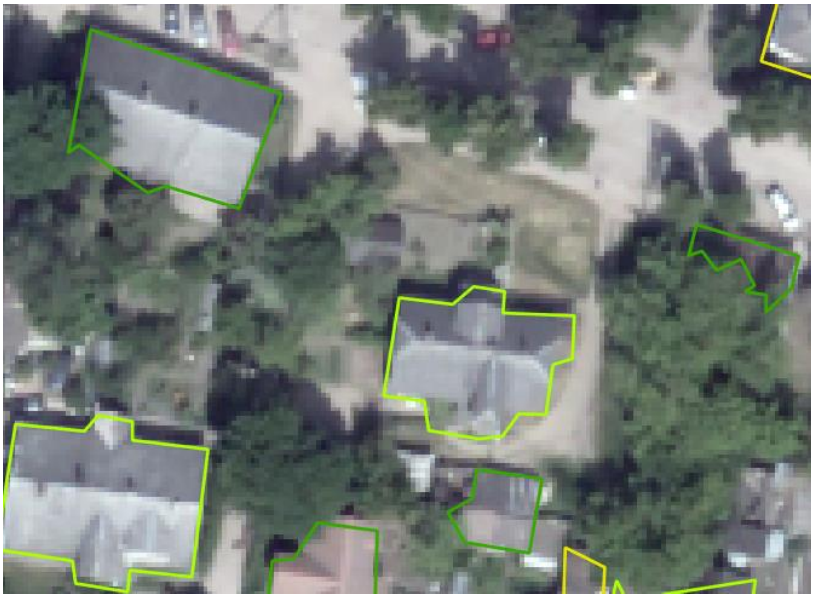


Figure 36. Building geometry errors caused by vegetation

One of the problems with the building was that the building footprint was misaligned with the imagery and elevation data. Before we made any other changes, vertices has been modified for better alignment. Turning on nDSM as background and standard editing tools used. Vertices were dragged to align them better with the building's location on the nDSM basemap.

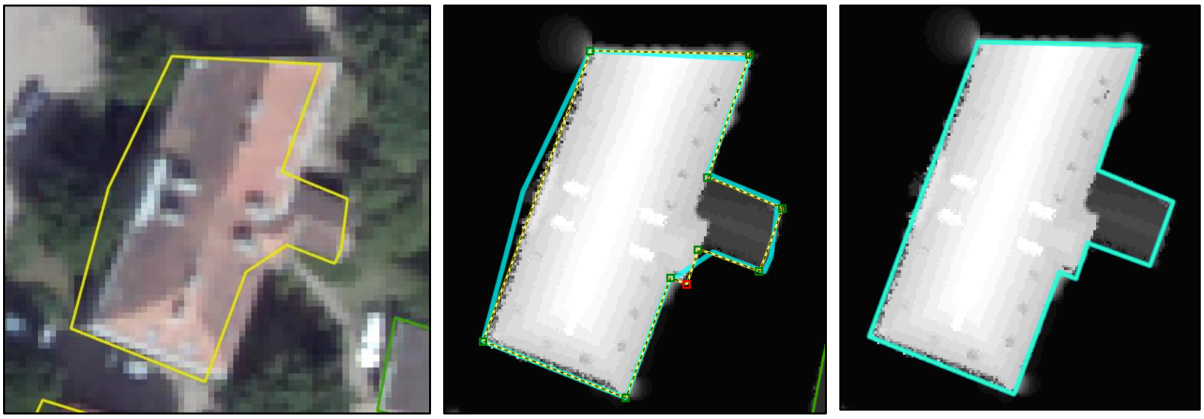


Figure 37. Misaligned building, editing footprint vertices, modified footprint

Second problem appears when comparing the building footprint to nDSM and the LAS point cloud, we see that the building footprint had one uniform height evenly spread across its area, the actual building had two parts with vastly different heights: the main tower of the building and part around it. To fix this problem, we split the building footprint into two features with standard editing tools, one to represent each part of the footprint.

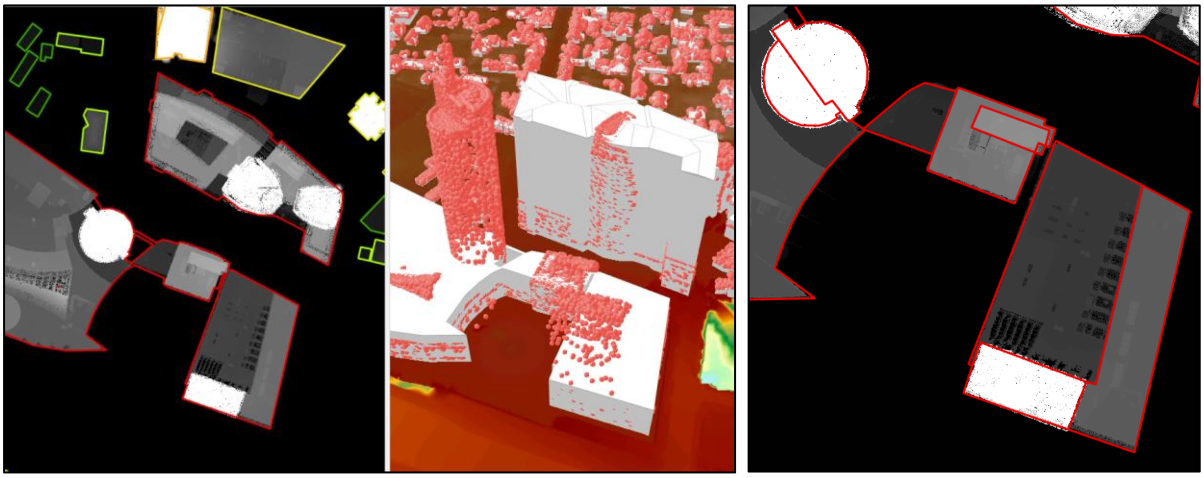


Figure 38. Multiple roof parts segmentation errors and modified segments

Features were splitted into multiple parts, but now all new features have the same attribute information as the original feature. The one building feature has the correct building height, but the feature that represents other building parts has the wrong height. In order to fix it we need to change attribute values of each part.

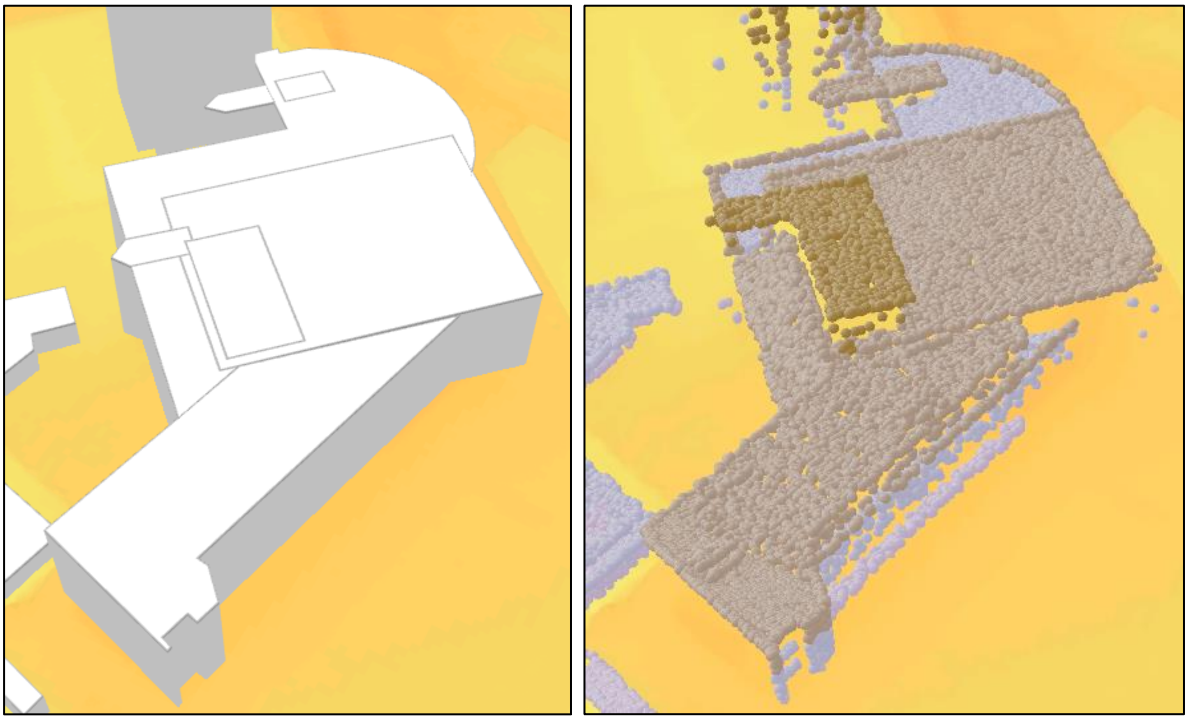


Figure 39. Segmented building with single height value and original LiDAR data

To identify segment height click on LiDAR point to open pop-up.

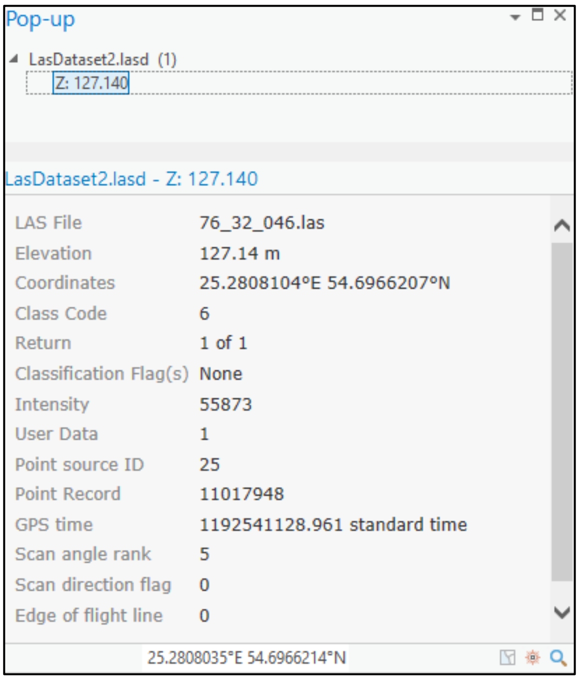


Figure 40. LiDAR point pop-up window

Depending on clicked location, the pop-up window shows that approximate building part elevation is about 127m. However, the LAS points have the true elevation of the point, the building height attribute in the roof form features has the maximum height of the building (which excludes the ground elevation). So we need to subtract the base elevation of the building from the elevation of the clicked point to get the elevation for the building height attribute.

OBJECTID	1267
BLDGHEIGHT	16.656815
EAVEHEIGHT	<Null>
ROOFFORM	Flat
BuildingFID	Building_1456
BASEELEV	105.3
ROOFDIR	<Null>
RoofDirAdjust	0
OriginalFID	1456
Shape_Length	197.101784
Shape_Area	1016.063289
RMSE	3.218307

Figure 41. Building segment attributes and modified building

The data is currently only a 2D feature class symbolized to look 3D (active 3D layer). After finished editing we need to convert these features to multipath for later use and sharing. Using *Layer 3D to feature class* converted existing features into multipath features and according unique building id to keep certain attributes like address, real estate number and etc. belonging to whole building.

2.2. Road extraction

Road information has an important role in many modern applications, including navigation, transportation and enables existing GIS databases to be uploaded efficiently (Matkan et al 2014). In GIS field, road extraction from digital images has received considerable attention in the past decades. Increasingly, LiDAR data has been used for road extraction (e.g., Boyko and Funkhouser 2011, Zuo & Quakenbush 2010, Matkan et al 2014). Fully automated road extraction in urban areas can be difficult due to the complexity of urban features, while manual digitizing of roads from images can be time consuming. In many cases, a semi-automated approach to road extraction can be implemented to improve the efficiency, accuracy, and cost-effectiveness of data development activities.

Ground features such as water bodies and asphalt pavement usually have very low intensity values, while some buildings roofs may also have similar intensity values. The approach of integration of LiDAR intensity data, DTM and DSM used for road extraction (Fig. 42).

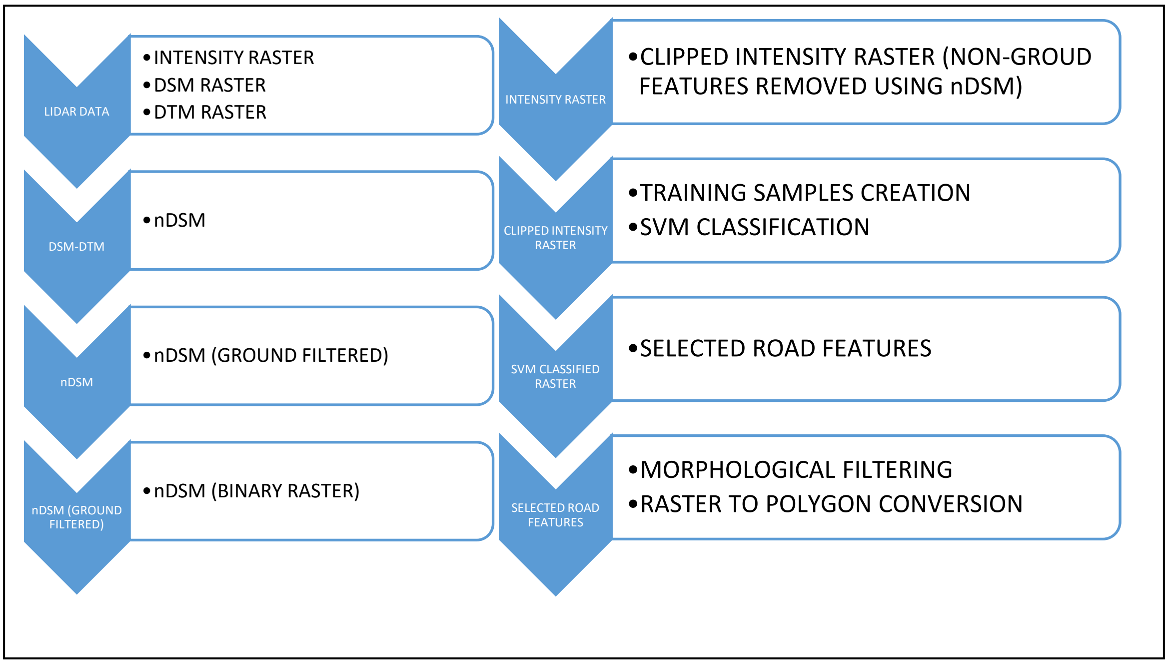


Figure 42. Flowchart of roads extraction

At first step LiDAR intensity image was created using *LAS Dataset To Raster*. Used first return values, Binning Inverse Distance Weighted interpolation method, raster cell size set to 0.2. Generated intensity image converted from 16-bit into 8-bit raster image for later processing.

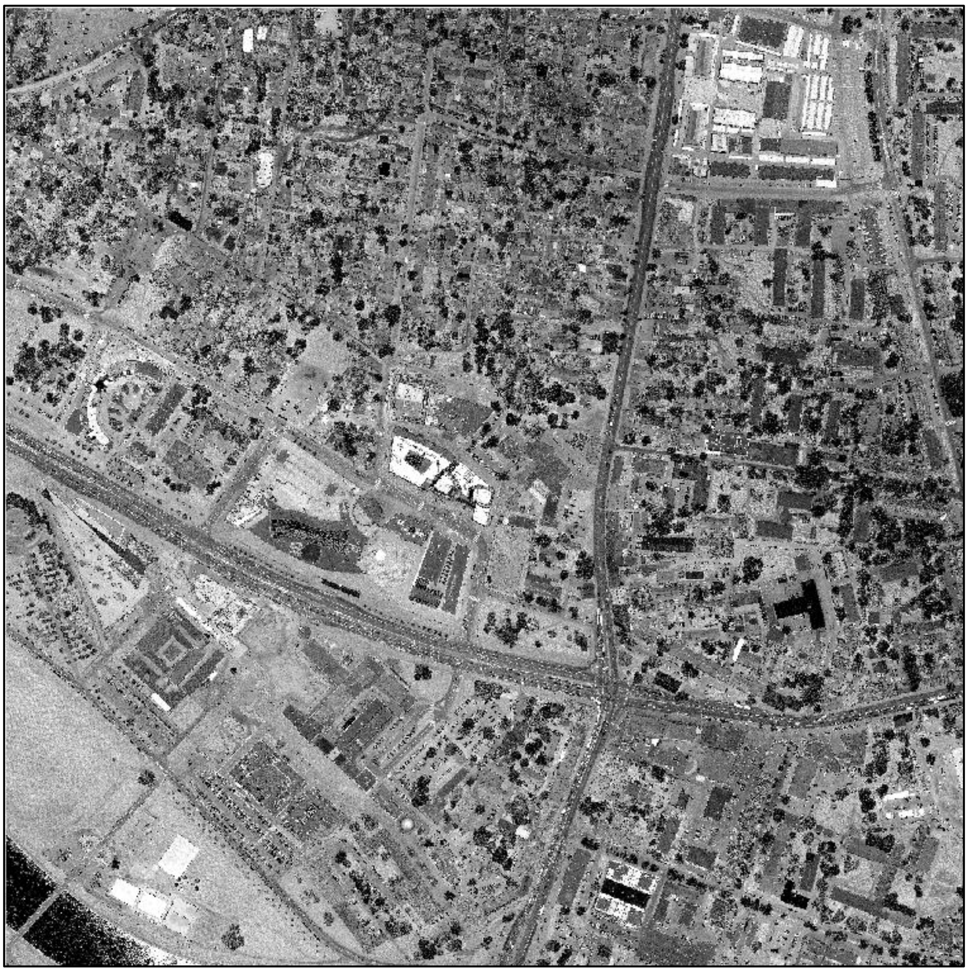


Figure 43. Intensity raster

Then a DSM and DTM rasters were created from LiDAR points by *LAS Dataset To Raster*. Elevation values was used for raster generation, interpolation type set to binning, cell size set to 0.2, *Inverse Distance Weighted* point values assigned and voids were eliminated using *linear* void fill method. Using *minus* function created nDSM raster. Using threshold (from -1 to 1) new binary raster was generated out of nDSM. Value of 0 represents ground features and value of 1 other features like buildings, trees and etc. Values that represents other non-ground features were set to *NoData* using raster calculator.

Figure 44. Binary image of ground points with NoData values (white)

Using raster calculator intensity raster values that overlaps nDSM raster were extracted. As result new intensity image was generated without buildings, trees and etc.

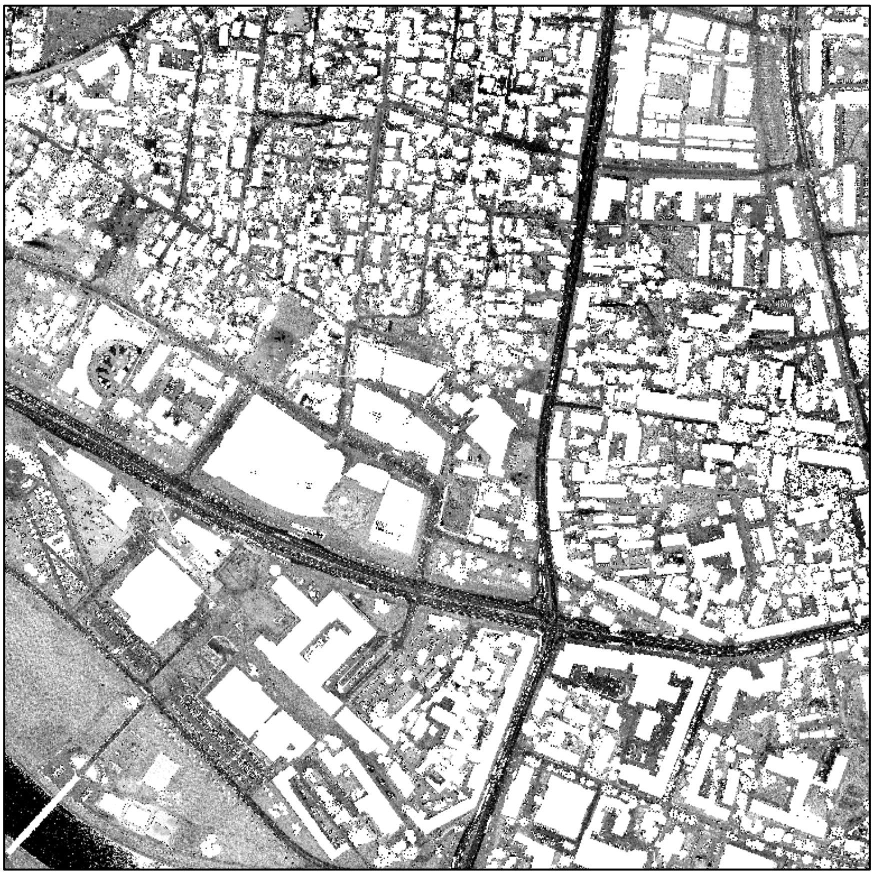


Figure 45. Intensity raster of ground points

At next step used image classification *training samples manager*. With this tool selected representative sites for each land cover class in the image. These sites are called training samples. A training sample has location information (point or polygon) and associated land cover class. The image classification algorithm uses the training samples to identify the land cover classes in the entire image. Training samples are saved as a feature class locally. Total five classes were used as training samples: asphalt, concrete, grassland, gravel, water.

Support Vector Machine (SVM) classification performed which mapped input data vectors into a higher dimensional feature space and optimally separated the data into the different classes. *Support vector machines* can handle very large images, and is less susceptible to noise, correlated bands, or an unbalanced number or size of training sites within each class. As result segmented image was generated.

Figure 46. Segmented image by SVM classifier

Segmented image reclassified with raster calculator to keep only values that represent roads. Focal statistics applied to calculate for each input cell location a mean statistics of the values within a circular (radius = 5 cells) neighbourhood around it.

Figure 47. Road network with calculated mean values of segmented raster

For final result raster calculator applied to remove values which are higher than threshold (0.45). Generated raster converted to polygonal features and morphological filters used to eliminate rough inconsistencies.

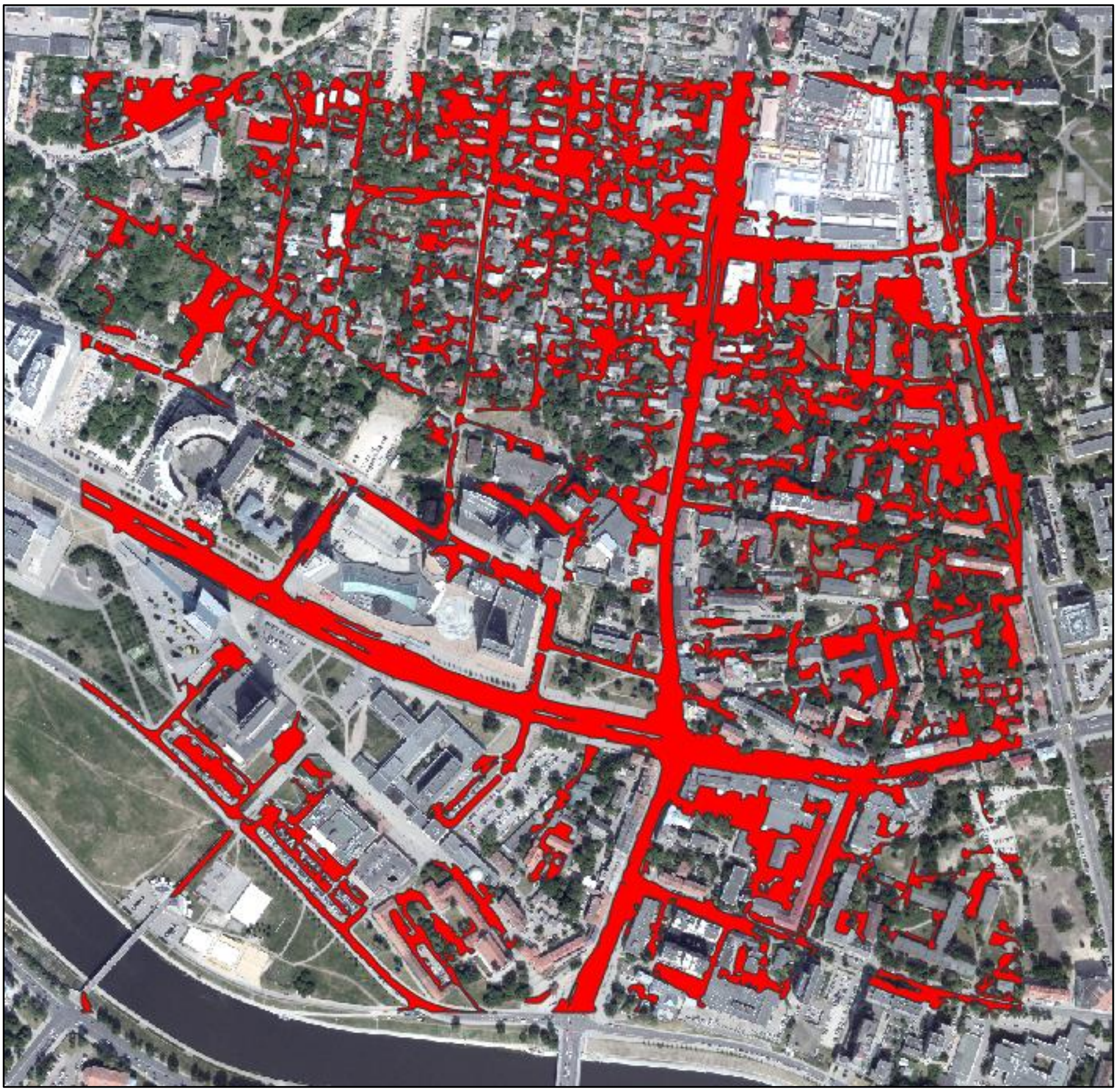


Figure 48. Final result of extracted road areas (and some other ground features)

Final road network can be corrected through interactive editing and converted into line features (centerlines). Attribute information can be filled for both lines and polygon features to better represent information related to roads and road network.

2.3. Vegetation extraction

Urban forests are relatively large financial investment for cities. Despite efforts and resources used for tree care, many cities often lack detailed information about their condition. LiDAR remote sensing technology, already used in commercial forest management and shows great potential as a forest monitoring tool. LiDAR data provides three-dimensional information directly and receives multiple return signals from vegetation (Fig. 49).

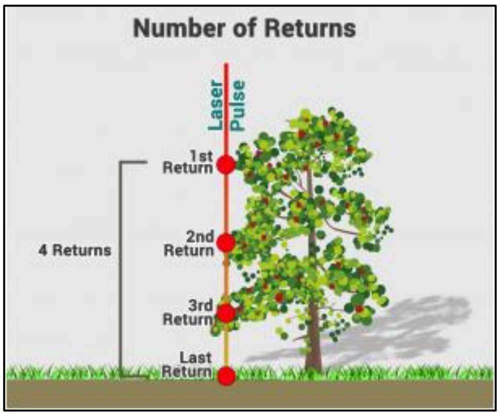


Figure 49. Multiple return explanation (A Complete Guide to LiDAR 2018)

The single tree detection using airborne LiDAR data was primarily proposed by Brandtberg (1999), Hyyppä and Inkinen (1999) and Samberg and Hyyppä, (1999), which later became an accepted topic for researchers (Hyyppä et al. 2008). Most published algorithms detect individual trees from an interpolated raster surface from LiDAR points that hit on the tree canopy surface (Moradi et al 2016) (Fig. 50)

Algorithm	Reference	Forest type	Leaf condition (leaf on or off)	Point density (points/m2)	Accuracy (In %)
Local maxima	Persson et al., 2002	Coniferous and Deciduous	-----	-----	71
Scale-space theory	Brandtberg et al., 2003	Deciduous	off	12	----
Pouring algorithm	Koch et al., 2006	Deciduous	on	5-10	72.73 (avg.)
Normalized cut	Reitberger et al., 2009	Coniferous and Deciduous	on and off	10-25	66
Watershed	Alonzo et al., 2014	Urban Forest	on	22	83
Watershed	Duncanson et al., 2014	Coniferous and Deciduous	on and off	50	70
Hierarchy-Directed Acyclic Graph	Strîmbu, F. and Strîmbu, M., 2015	Coniferous and Deciduous	off	30	61.1-98.8

Figure 50. Different individual tree detection methods based on LiDAR-derived raster surface (Moradi et al 2016)

On the basis of studies and analyses carried out by Plowright (2015), Moradi et al (2016), Argamosa et al (2016), Quinto et al (2017) and Liu (2013) the algorithm (Fig. 51) was created to delineate individual tree crowns and information (tree height, canopy width) using ArcGIS tools.

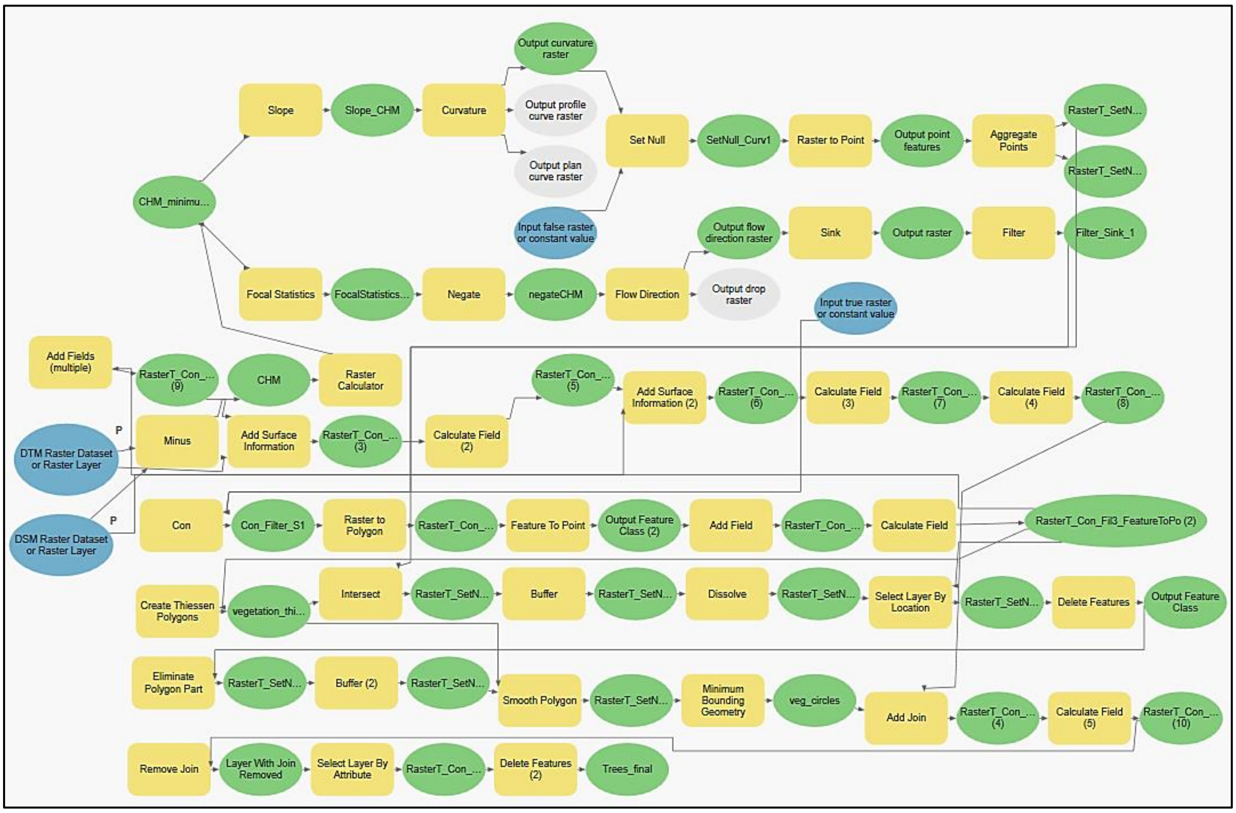


Figure 51. ArcGIS pro model for individual tree extraction

Canopy height model (CHM) is the difference between the DSM (with tops of the trees) and the DTM. The CHM represents the actual height of trees with the influence of ground elevation removed. CHM raster was created from DSM by subtracting DTM raster values (Fig. 53).

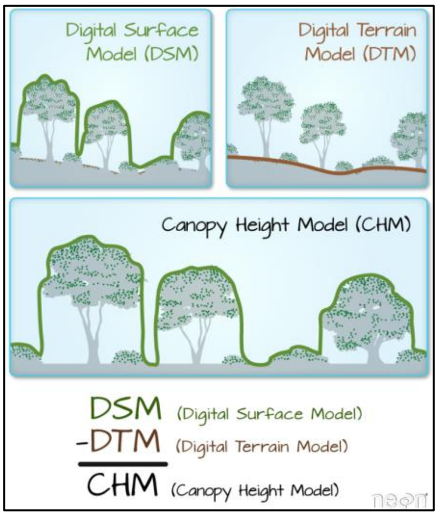


Figure 52. CHM explanation (Wasser 2018)

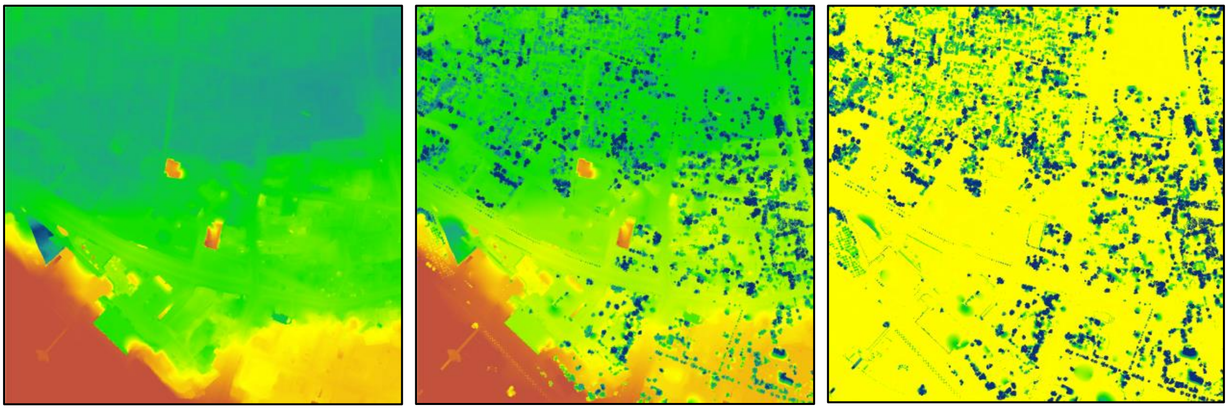


Figure 53. Generated DTM, DSM and CHM rasters

The minimum height of the tree (2 meters) was determined for CHM raster using raster calculator. Taking the CHM as an input, pixels with elevation values greater than or equal to 2 meters were extracted, other values set to null. The maximum value of a 1 m radius in a cell was obtained from this raster using Focal Statistics tool.

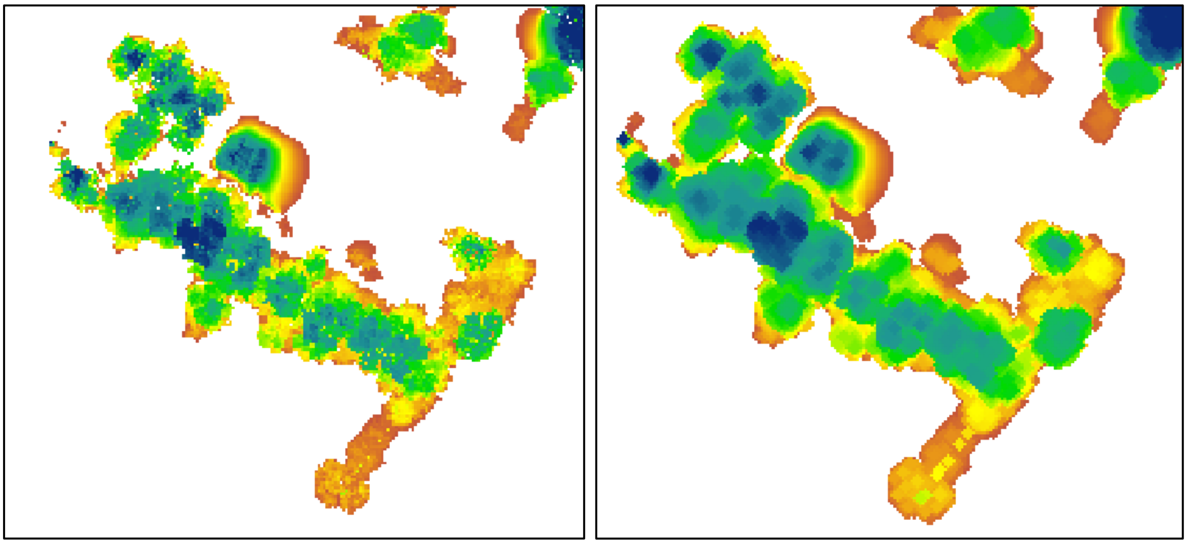


Figure 54. Part of CHM raster before (left) and after (right) applying local maxima

Next part takes the CHM as an input to compute for the slope (*Slope* tool) and setting output measurement units as degrees and planar method. Generated slope raster used to compute for curvature using *Curvature* geoprocessing tool. The profile curvature was computed by getting the curve parallel to the maximum slope and values less than 0 were set to null. Generated curvature raster converted to points (*Raster To Point*) and later to polygons (*Aggregate*). Polygons were created around clusters of three or more points within the aggregation distance of 1 meter.

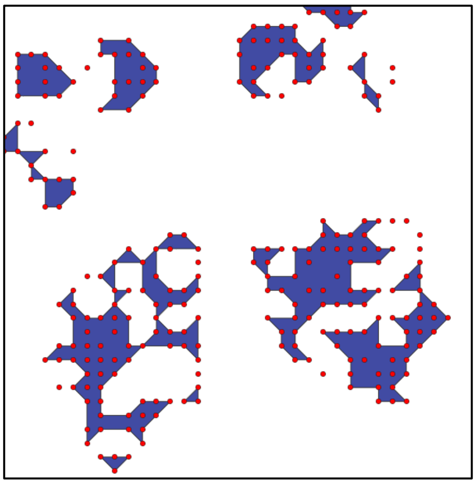


Figure 55. Curvature raster points and aggregated polygons

CHM with previously calculated elevation threshold and local maxima used to find treetops. Using *Negate* function changed cell values from positive to negative on a cell-by-cell basis. Using negative CHM raster as an input created a raster of flow direction from each cell to its downslope neighbours cells using D8 method. Using *Sink* function all sinks or areas of internal drainage were recognized. *Filter* function performed for edge-enhancing (High pass) and conditional rule (value is equal to 0) applied for extraction of lowest areas from adjacent values. Filtered values converted to points (Feature to point) with unique ids.

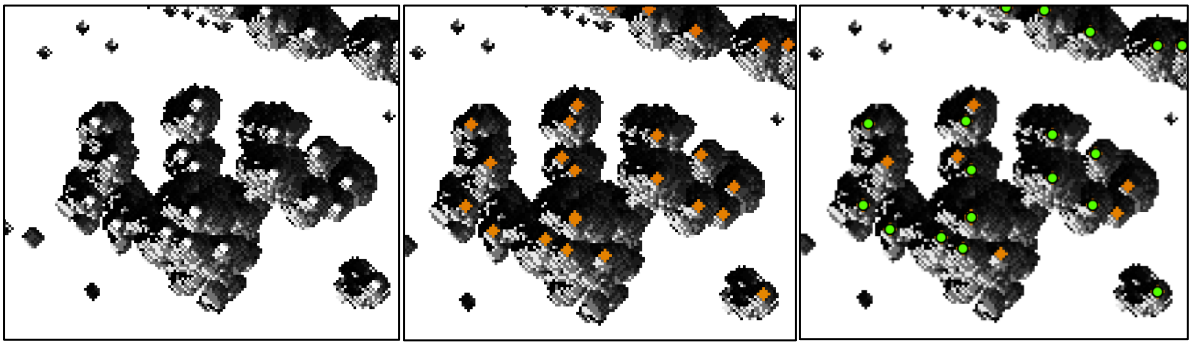


Figure 56. Flow direction raster, flow direction raster overlaid with sink raster (orange), filtered treetop points

Using the previously generated treetop representing points Thiessen polygons were created. *Create Thiessen polygons* tool was used to divide the area covered by the input point features into Thiessen or proximal zones. These zones represent full areas where any location within the zone is closer to its associated input point than to any other input point. Thiessen polygons used as a boundary of each tree crown.

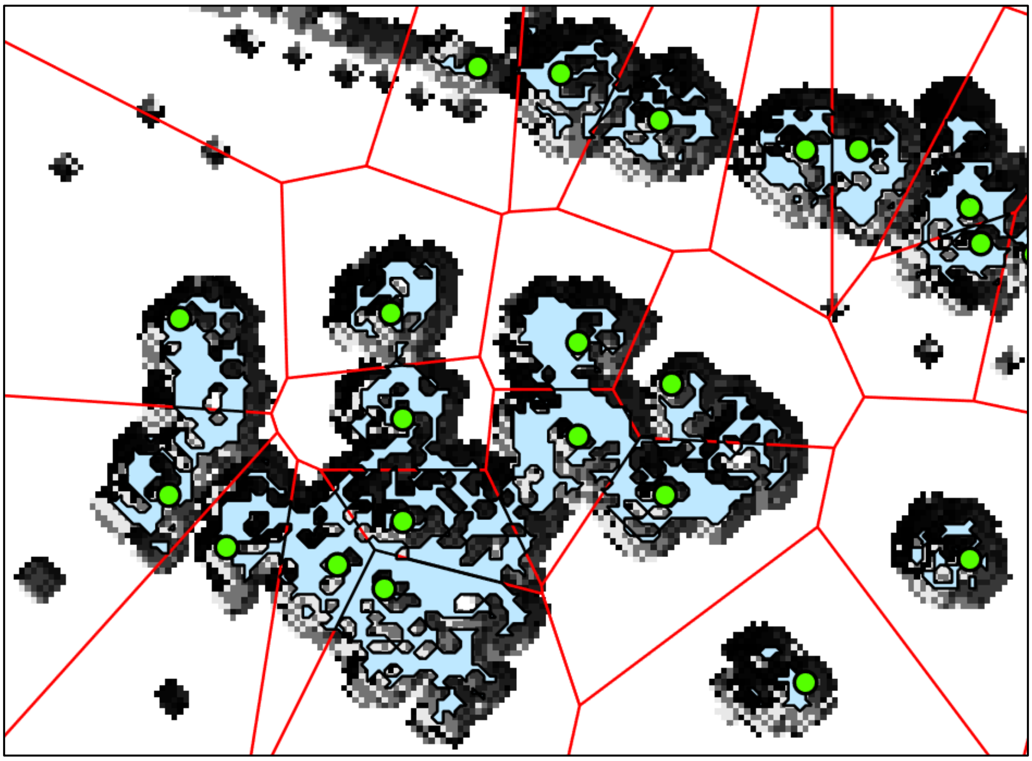


Figure 57. Thiessen (red) polygons separating tree (green) crowns (blue)

The parts of the tree crown that were not spatially connected to the tree to points were removed. Circles were created with *Minimal Boundary Geometry* to obtain information on the diameter of the tree crown. *Add Surface Information* tool was used to gather information of tree height. Each feature class layer with information was joined by unique attribute (id) and using multiple attribute calculators' information was stored into specific feature class fields.

Figure 58. Principle of tree crown diameter information extraction

As a final result point feature class was generated with an attribute information of ground elevation, height, and crown diameter. *Preset Layers* integrated in ArcGIS Pro is another way to display and share data in three-dimensional environment. *Realistic Trees* preset layers used to show trees in real-world form and textures. At this case the tree type connected to the attribute fields which contains the height and crown width information. Generated 3D trees used for visualisation and can be shared across platform.

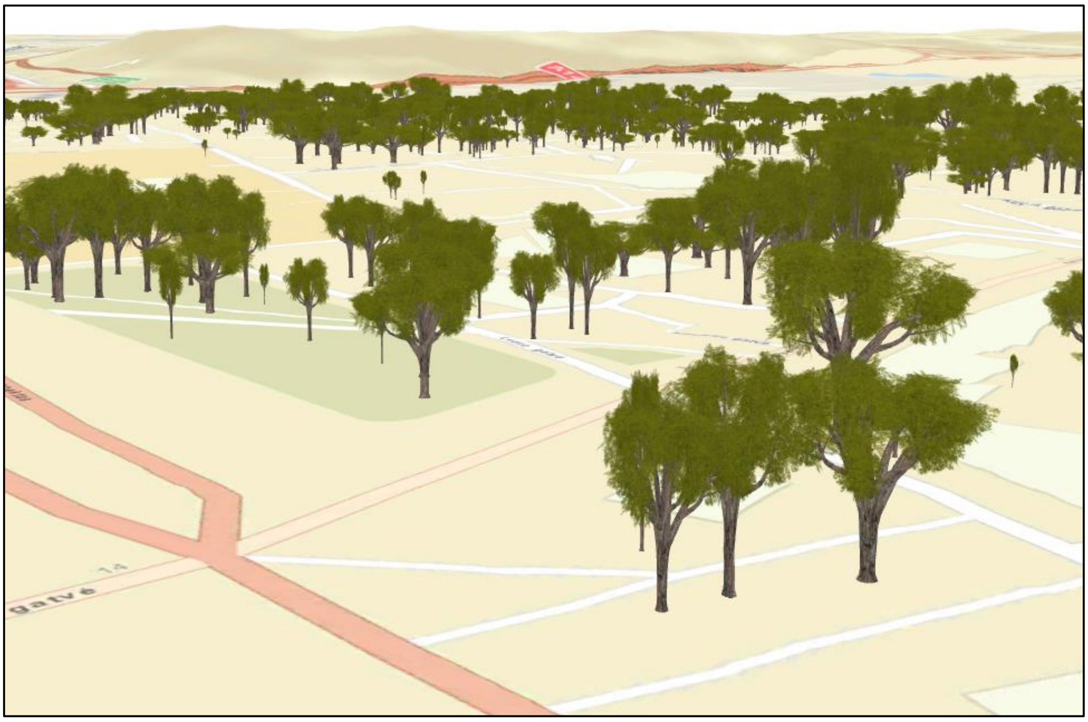


Figure 59. Extracted 3D trees

2.4. Sharing

There are few solutions for sharing data across ArcGIS platform like ArcGIS *Online* or *Portal*.

ArcGIS *Online* is a cloud-based mapping and analysis solution. Through *Map Viewer* and *3D Scene Viewer*, information of basemaps, styles templates and widgets can be accessed.

Portal for ArcGIS is a component of ArcGIS Enterprise that allows to share maps, scenes, apps, and other geographic information with other people in organization.

CONCLUSION AND DISCUSSION

In order to use LiDAR data, we need to know the basic principles of LiDAR remote sensing. Airborne laser scanning is performed from a fixed wing aircraft or helicopter. Technique is based on two main components: a laser scanner system that measures the distance to a location on the ground and the GPS/IMU combination that measures the position and orientation of the system. LiDAR accuracy is checked by comparing measured points with laser scan data and defined as the standard deviation (σ^2) and root mean square error (RMSE). The most common errors are navigation-based, these errors cannot be avoided without data control, so for LiDAR-specific ground control points is recommended to improve final accuracy. Received LIDAR data may be provided in a variety of data formats depending on the software platform being used. ESRI's company ArcGIS software uses the *.las data format, so knowing the structure (version and type) of this format, we know what information the data contains (number of classification classes and other attributes).

In Lithuania, LiDAR scanning were performed twice (in 2007 and 2018). Recent data is presented in a more compatible data format (LAS) than before (XYZ). Although the accuracy of the data obtained has slightly improved, however, the increased density of points and the number of classes make it possible to use the data more widely and efficiently.

3D city model combines information collections of certain objects and expressed in three dimensional geometry. Two ESRI products (ArcGIS Pro and CityEngine) used for main topics features analysis. Features of 3D building generation, recognition of road infrastructure and extraction of individual trees are analysed and described. 1 km x 1 km of central area of the city was analysed, due to the variety and height of the buildings, relief changes, and other specific features of the urban environment.

3D buildings were extracted in semi-automatic way. Created Las dataset for use of point clouds on ArcGIS platform, as well as visualization and statistical calculations. DTM and DSM rasters were used as a basis for extraction of 3D buildings. Building geometry parameters for extracted footprints calculated using ArcGIS tools and mathematical computations. Segmentation (optional) process used for complex rooftops extraction. CityEngine platform used for CGA rule creation and visualization of buildings by attribute data. Buildings with the highest RMSE value were edited manually.

To build road representing polygons used intensity, DTM and DSM rasters. Non-ground features were removed from intensity image. SVM classification technique used for segmentation and feature extraction.

For individual tree extraction a model was created in ArcGIS Pro environment. DTM and DSM used as a basis. As a result point feature class generated with attribute values of base elevation, tree height and crown diameter. Results displayed in 3D using Preset layers in ArcGIS Pro.

The workflow traditionally used to reconstruct 3D buildings from airborne LiDAR is relatively straight-forward: the LiDAR point-cloud is transformed into a DTM and DSM rasters, then inspected by human editors for buildings present. If a building is found, one or more polygon describing the roof form of the building is manually digitized, e.g. if it is a large hip roof with two gable outlets, there will be three polygons (one hip and two gables on top) drawn by the editor or recognized in the segmentation process. Once all the roofs are described that way, a set of ArcGIS Procedural rules is applied to extrude the building models using the roof segments, with heights and ridge directions computed from the DSM.

Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) have been quite popular in the recent years. Success stories from self-driving cars to medical imaging are filling the media. GIS is not falling behind: having efficient and intelligent tools will significantly cut costs of creating and maintaining GIS content and bring its quality to higher standards.

Artificial Intelligence is a broad family of computer algorithms and methodologies which encompasses fields like Machine Learning (based on statistical methods), and, as we will be diving into here, Deep Learning (based on deep artificial neural networks). The unique and valuable feature of these approaches is the ability to learn and self-improve over time. Now, with deep convolutional neural networks (CNN), that is no longer needed; the network will learn these coefficients by itself during training, which is done by providing it with enough examples. There are experiments and examples presented by Esri to explore ways to streamline the creation of 3D content using Artificial Intelligence.

REFERENCES

1. Baltsavias, E., 1999. ‘Airborne laser scanning: basic relations and formulas’, *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 54, no. 2-3, pp. 199-214, doi: 10.1016/S0924-2716(99)00015-5
2. American Society for Photogrammetry and Remote Sensing (ASPRS) n.d., *Laser (Las) File Format Exchange Activities,* viewed 6 May 2018, https://www.asprs.org/divisions-committees/LiDAR-division/laser-las-file-format-exchange-activities
3. American Society for Photogrammetry and Remote Sensing (ASPRS) 2011, *LAS specification, version 1.4 - R13,* viewed 10 May 2018, https://www.asprs.org/wp-content/uploads/2010/12/LAS_1_4_r13.pdf.
4. Csanyi, N & Toth, CK 2007, ‘Improvement of LiDAR Data Accuracy Using LiDAR-Specific Ground Targets’, *Photogrammetric Engineering & Remote Sensing,* vol. 73, no. 4, pp. 385-396, doi: 10.14358/PERS.73.4.385
5. Žalnierukas, A, Ruzgienė, B, Kalantaitė, A & Valaitienė, R 2009, ‘Miestų skenavimo LiDAR metodu tikslumo analizė‘, *Geodesy and Cartography,* vol. 35, no. 2, pp. 55-60, doi: 10.3846/1392-1541.2009.35.55-60
6. Kalantaitė, A, Putrimas, R & Šlikas, D 2010, ‘Erdvinių skenavimo duomenų taikymas vietovės trimačiams modeliams generuoti‘, *Geodesy and Cartography,* vol. 36, no. 4, pp. 151-155. doi: 10.3846/gc.2010.24
7. FIT Conseil (Géométres Experts), UAB InfoERA, 2007, ‘*Lietuvos Respublikos detalaus erdvinio modelio sudarymas, iš LIDAR skrydžių ir duomenų kaupimo kokybės kontrolės ataskaita*‘, 13 p.
8. National Land Service under Ministry of Agriculture, 2017, *‘Lietuvos Respublikos apskričių centrų lazerinio skenavimo taškų erdvinių duomenų rinkinio sudarymo kokybės kontrolės paslaugų techninė specifikacija’,* viewed 20 September 2018, http://www.cvpp.lt/index.php?option=com_vpt&theme=new&task=view&tender_id=340613.
9. Nelson, R 2014, ‘How did we get here? An early history of forestry lidar 1’, *Canadian Journal of Remote Sensing*, vol. 39, no. S1, pp. 6-17, doi: 10.5589/m13-011
10. Krabill, WB 1989, ‘GPS applications to laser profiling and laser scanning for digital terrain models’, In *Photogrammetric Week 1989*, Wichmann Verlag, Heidelberg, Germany.
11. Friess, P 1989, ‘Empirical accuracy of positions computed from airborne GPS data’, In *High Precision Navigation*, Linkwitz, K, & Hangleiter, U (Eds.), Springer Verlag, Berlin and Heidelberg, Germany, pp. 163 - 175.

12. Lindenberger, J 1989, ‘Test results of laser profiling for topographic terrain survey’, Proceedings *42nd Photogrammetric Week*, Stuttgart, Germany, Wichmann Verlag, Heidelberg, Germany, pp. 25 - 39.
13. Schwarz, KP, Chapman, M, Cannon, M, Gong, P & Cosandier, D 1994, ‘A precise positioning/attitude system in support of airborne remote sensing’, Proceedings of the *GIS/ISPRS Conference*, 6 - 10 June, Ottawa, Canada.
14. Ackermann, E 1994, ‘On the status and accuracy performance of GPS photogrammetry’, Proceedings Of *Mapping and Remote Sensing Tools for the 21st Century*, 26 - 29 August, Washington, DC, pp. 80 - 90.
15. Ackermann, E 1988, ‘Digital terrain models of forest areas by airborne profiling’, In High precision Navigation, Linknvitz, K, & Hangleiter, U (Eds.), pp. 239—250, Springer-Verlag, New York, NY, USA.
16. Lindenberger, J 1993, ‘Laser-Profilmessung zur topographischen Geländeaufnahme’, PhD dissertation, Deutsche Geodätische Kommission bei der Bayerischen Akademie der Wissenschaften.
17. Petrie, G 2011, ‘Airborne Topographic Laser Scanners: Current Developments in the Technology’, *GEO Informatics*, vol. 14, pp. 34 – 44, viewed 10 July, https://www.researchgate.net/publication/281309092_Airborne_Topographic_Laser_Scanners_Current_Developments_in_the_Technology.
18. ArcGIS Pro help n.d., *What is a LAS dataset?*, viewed 11 November 2018, http://pro.arcgis.com/en/pro-app/help/data/las-dataset/what-is-a-las-dataset-.htm.
19. ArcGIS Pro help n.d., *Create LAS Dataset*, viewed 11 November 2018 http://pro.arcgis.com/en/pro-app/tool-reference/data-management/create-las-dataset.htm.
20. Karan, PE, Sivakumar, R, Irizarry, J & Guhathakurta, S 2014, ‘Digital Modeling of Construction Site Terrain Using Remotely Sensed Data and Geographic Information Systems Analyses’, *Journal of Construction Engineering and Management*, vol. 140, no. 3, doi: 10.1061/(ASCE)CO.1943-7862.0000822
21. Zouaouid, I, Khanan, A, Faisal, M, Omar, Z, Abdul Rahman FZ, Salleh, M & Radhie, M, 2016, ‘Evaluating Error of LiDar Derived DEM Interpolation for Vegetation Area’, *International archives of the photogrammetry, remote sensing and spatial information sciences - ISPRS archives*, XLII-4/W1, pp. 141 – 150, doi: 10.5194/isprs-archives-XLII-4-W1-141-2016
22. Sibson, R 1981, ‘A Brief Description of Natural Neighbor Interpolation’, chapter 2 in *Interpolating Multivariate Data*, New York: John Wiley & Sons, 1981, pp. 21 - 36.

23. Wilson, JP & Gallant, JC 2000, 'Terrain analysis: Principles and applications', Wiley, New York.
24. Guo, T & Yasuoka, Y 2002, 'Snake-based approach for building extraction from high-resolution satellite images and height data in urban areas', Proceedings of the *23rd Asian Conference on Remote Sensing*.
25. Mirosław-Świątek, D, Szporak-Wasilewska, S, Michałowski, R, Kardel, I & Grygoruk, M 2016, 'Developing an algorithm for enhancement of a digital terrain model for a densely vegetated floodplain wetland', *Journal of applied remote sensing*, vol. 10, issue 3, doi: 10.1117/1.JRS.10.036013.
26. Župan, R, Župan, K, Frangeš, S & Vinković, A 2018, *'Automatic procedural 3d modelling of buildings', Technical Journal*, vol. 12, no. 3, doi: 10.31803/tg-20180713143638
27. ESRI CityEngine 2018, *Advanced 3D city design software*, viewed 10 December 2018, https://www.esri.com/en-us/arcgis/products/esri-cityengine/overview
28. Ribeiro, A, Duarte De Almeida, JP & Ellul, C 2014, 'Exploring CityEngine as a Visualisation Tool for 3D Cadastre', *4th International FIG 3D Cadastre Workshop*, 9-11 November 2014, Dubai, United Arab Emirates, viewed 2 December 2018, http://www.gdmc.nl/3dcadastre/literature/3Dcad_2014_25.pdf.
29. Matkan, A, Hajeb, M & Sadeghian, S 2014, 'Road Extraction from Lidar Data Using Support Vector Machine Classification', *Photogrammetric Engineering & Remote Sensing*, vol. 80, pp. 409 – 422, doi: 10.14358/PERS.80.5.409
30. Boyko, A & Funkhouser, T 2011, 'Extracting roads from dense point clouds in large scale urban environment', *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66:S2-S12, doi: 10.1016/j.isprsjprs.2011.09.009
31. Zuo, Y & Quackenbush, L 2010, 'Road extraction form lidar data in residential and commercial areas of Oneida County, New York', *ASPRS 2010 Annual Conference*, San Diego, California, April 26 - 30, 2010, viewed 5 November 2018, https://pdfs.semanticscholar.org/74eb/172c7857aad4fe5ba4e13718d9f37afba717.pdf
32. GIS Geography, 2018, *A Complete Guide to LiDAR: Light Detection and Ranging*, GIS Geography, viewed 3 December 2018, https://gisgeography.com/lidar-light-detection-and-ranging/
33. Hyyppä, J, Hyyppä, H, Leckie, D, Gougeon, F, Yu, X & Maltamo, M 2008, 'Review of methods of small-footprint airborne laser scanning for extracting forest inventory data in boreal forests', *International Journal of Remote Sensing*, vol. 29, pp. 1339-1366, doi: 10.1080/01431160701736489

34. Brandtberg, T 1999, ‘Automatic Individual Tree-based Analysis of High Spatial Resolution Remotely Sensed Data’ Ph.D. dissertation, Acta Universitatis, Agriculutae Sueciae, Silvestria 118, Swedish University of Agricultural Sciences, Uppsala, Sweden, viewed 13 September 2018, https://www.researchgate.net/publication/35985290_Automatic_individual_tree-based_analysis_of_high_spatial_resolution_remotely_sensed_data
35. Hyyppa, J, & Inkinen, M 1999, ‘Detecting and estimating attributes for single trees using laser scanner’, *Photogrammetric Engineering & Remote Sensing*, vol. 68, no. 9, September 2002, pp. 925-932, viewed on 22 November 2018, https://pdfs.semanticscholar.org/f4c8/638742e1ae48f13eb4e6f98688cc45394721.pdf
36. Samberg, A, Hyyppä, J, Ruppert, G, Hyyppä, H, Ziegler, M, Schardt, M & Soininen, A 1999, ‘Evaluation of the laser scanner data in the forest area’, *Laser Radar Technology and Applications IV*, Proceedings vol. 3707, pp. 570-581, https://doi.org/10.1117/12.351385
37. Moradi, A, Satari, M & Momeni, M 2016, ‘Individual tree of urban forest extraction from very high density lidar data’, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B3, 2016 XXIII ISPRS Congress, 12 - 19 July 2016, Prague, Czech Republic, doi: 10.5194/isprsarchives-XLI-B3-337-2016
38. Plowright, A 2015, *Extracting trees in an urban environment using airborne LiDAR Report*, 9 March 2015, viewed 26 September 2018, https://open.library.ubc.ca/cIRcle/collections/graduateresearch/42591/items/1.0075836
39. Argamosa, R, Paringit, E, Quinton, KR, AM Tandoc, F, Faelga, RA, A.G. Ibañez, C, A.V. Posilero, M & Zaragosa, G 2016, ‘Fully automated gis-based individual tree crown delineation based on curvature values from a lidar derived canopy height model in a coniferous plantation’, *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B8, pp. 563-569, doi: 10.5194/isprs-archives-XLI-B8-563-2016
40. Quinto, M, Beronilla, R, Damian, G, Camaso, E & Alberto, R, 2017, 'Extraction of Forest Plantation Resources in Selected Forest of San Manuel, Pangasinan, Philippines Using LiDAR Data for Forest Status Assessment', *World Academy of Science, Engineering and Technology, International Science Index 127, International Journal of Biological, Biomolecular, Agricultural, Food and Biotechnological Engineering*, vol. 11(7), pp. 530 – 534, doi: doi.org/10.5281/zenodo.1131786
41. Liu, H 2013, ‘Automated Treetop Detection and Tree Crown Identification Using Discrete-return Lidar Data’, thesis, Denton, Texas, viewed 13 October 2018, https://digital.library.unt.edu/ark:/67531/metadc271858/

42. Wasser, L 2018, ‘Lesson 4. Create a Canopy Height Model With Lidar Data’, image, viewed 15 December 2018, https://www.earthdatascience.org/courses/earth-analytics/lidar-raster-data-r/lidar-chm-dem-dsm/

APPENDIXES

21-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“ teminė konferencija
Proceedings of the 21th Conference for Junior Researchers „Science – Future of Lithuania“

CIVILINĖ INŽINERIJA IR GEODEZIJA / CIVIL ENGINEERING AND GEODESY

2018 m. kovo 23 d. Vilnius
23 March, Vilnius, Lithuania

eISSN 2029-7149
eISBN 978-609-476-114-0
https://doi.org/10.3846/geo.2018.012

http://jmk.msk.vgtu.lt

3D PASTATŲ MODELIAVIMAS NAUDOJANT LIDAR DUOMENIS IR GIS TECHNOLOGIJAS

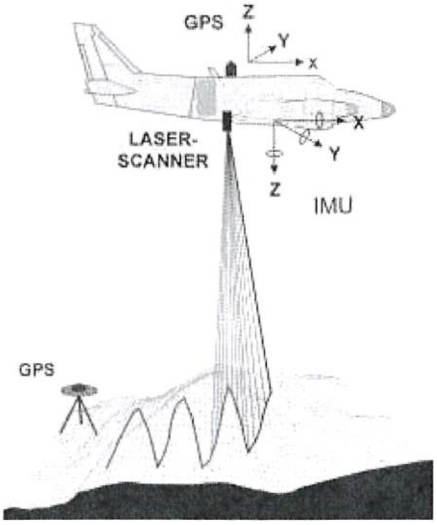
Eugenijus Streikus[1], Eimuntas Kazimieras Paršeliūnas[2]
Vilniaus Gedimino technikos universitetas,
El. p. [1]eugenijus.streikus@stud.vgtu.lt; [2]eimuntas.parseliunas@vgtu.lt

Anotacija. LiDAR (*light detection and ranging*) – tai nuolat populiarėjantis duomenų apie žemės paviršių surinkimo būdas naudojant orlaivyje sumontuotą lazerį. Pagrindiniai šio metodo privalumas – greitas 3D duomenų surinkimas dideliame plote. Užfiksuotus 3D duomenis galima panaudoti topografinę išraišką turinčių savybių objektams atpažinti, analizuoti ar atvaizduoti trimačiame žemėlapyje. 3D pastatams sukurti panaudoti 2007 m. LiDAR skenavimo duomenys, pirminis duomenų apdorojimas atliktas *LAStools* (rapidlasso GmbH), galutiniai 3D pastatų modeliai sukurti naudojantis ESRI kompanijos produktais – *ArcGIS Pro* ir *CityEngine*.

Reikšminiai žodžiai: geografinės informacinės sistemos, LiDAR, skaitmeninis reljefo modelis, skaitmeninis paviršiaus modelis.

Įvadas

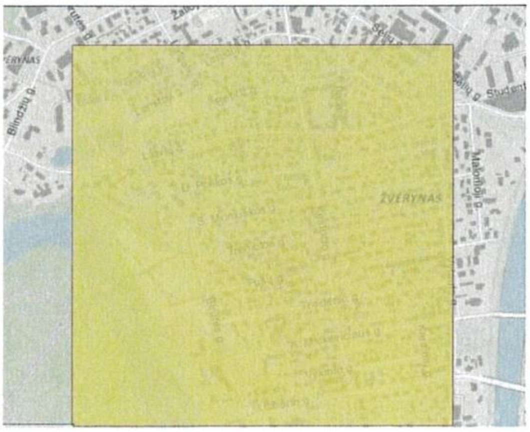
Pasaulyje naudojami įvairūs metodai 3D modeliams kurti. Kiekvienas jų turi savo taikymo sritį, privalumų ir trūkumų. Kuriant 3D modelius naudojami skirtingais metodais gaunami pradiniai duomenys, vieni iš jų – LiDAR (angl. *Light Detection And Ranging*). LiDAR technologijos ir principai aprašyti A. Žalnieruko ir K. Čypo (2006). Šių duomenų surinkimo tipai gali būti skirstomi į antžeminius (angl. *Terrestial Laser scanning – TLS*) arba skanuojant iš orlaivio (angl. *Airborne Laser scanning – ALS*) (Crosby, 2016) (1 paveikslas).



1 paveikslas. LiDAR duomenų surinkimo iš orlaivio iliustracija

2007 m. pirmą kartą Lietuvoje buvo sudaryti dešimties didžiausių miestų – Vilniaus, Kauno, Šiaulių, Panevėžio, Klaipėdos, Marijampolės, Alytaus, Tauragės, Telšių, Utenos – LiDAR duomenų rinkiniai. Bendras užimamos teritorijos duomenų kiekis apima 2475 km^2. Šie duomenys pateikiami atskiruose duomenų failuose klasifikuoti į keturis tipus – pastatai, žemės paviršius, augalija, tiltai (Žalnierukas, Ruzginė, Kalantaitė ir Valaitienė, 2009).

Nagrinėjamu atveju pasirinkta 1 km^2 Vilniaus miesto Žvėryno mikrorajono teritorija (2 paveikslas).



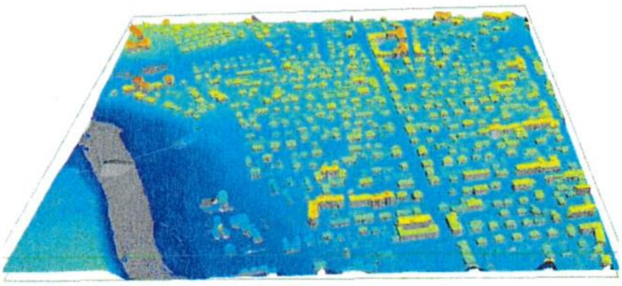
2 paveikslas. Analizuojama teritorija

Tyrimo tikslas – naudojant esamus LiDAR duomenis ir pritaikant geografinių informacinių sistemų (GIS) technologijas automatizuotai sukurti trimačius pastatų modelius ir atlikti gautų rezultatų analizę.

LiDAR duomenų paruošimas įkėlimui į GIS

Visi 2007 m. skanavimo duomenys, parsisiųsti iš Lietuvos erdvinės informacinės (LEI) sistemos, pateikiami XYZ formatu. Šis duomenų formato tipas nepalaikomas vėliau naudojamoje ArcGIS Pro programinėje įrangoje (PĮ), todėl jį būtina konvertuoti į LAS. Konvertuoti iš vieno formato į kitą naudotas kompanijos Rapidlasso atvirojo kodo programinės įrangos paketas LAStools.

Naudojant šiuos įrankius žemės paviršiaus ir pastatų duomenys konvertuoti į LAS duomenų formatą, taip pat sujungti du duomenų rinkiniai – pastatų ir žemės paviršiaus grupė į vieną. Šio žingsnio rezultatas – skaitmeninio žemės paviršiaus modelio (angl. *Digital surface model*, DSM) (3 paveikslas) ir skaitmeninio reljefo modelio duomenys (angl. *Digital terrain model*, DTM).

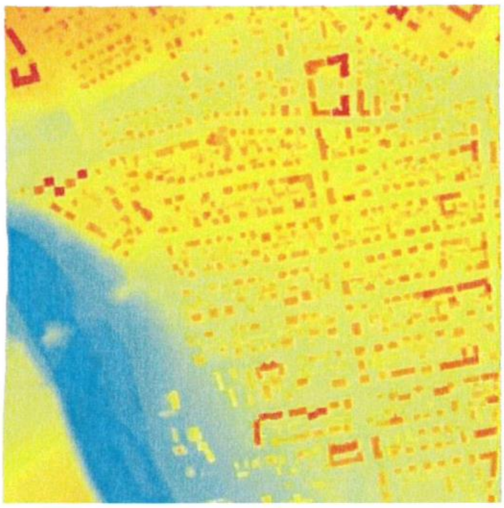


3 paveikslas. DSM LiDAR duomenys

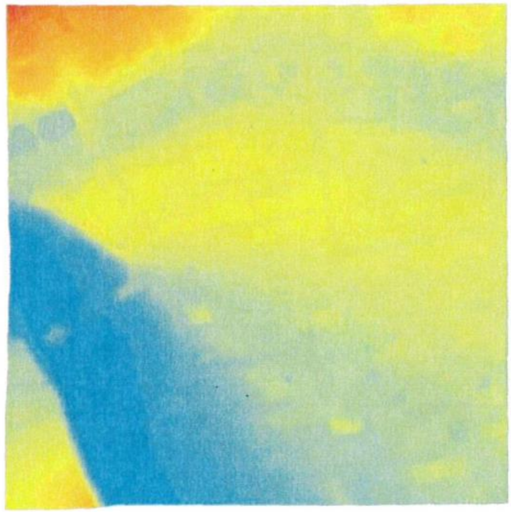
LiDAR duomenų apdorojimas GIS

Tinkamo formato (LAS) duomenys, importuoti į GIS programinę įrangą, ir sukurti rastrinio tipo failai atvaizduojantys skaitmeninius reljefo ir paviršiaus modelius (DTM ir DSM) (4 ir 5 paveikslai). Turimų duomenų taškų sklaida 1 m^2 plote yra 4 taškai, todėl generuojamų rastrų ląstelės dydis (angl. *Cell size*) – 0,5 m – yra pakankamas. Generuojant rastrus linijinis interpoliavimo metodas, o aukščio reikšmė perduodama pagal į rastro ląstelę patenkančių taškų vidutinę aukščio reikšmę. Gautųjų ląstelių skaičius abiejuose modeliuose sudaro 4 mln. (2000×2000).

Iš esamų paviršiaus ir reljefo rastrų sukurtas kitas rastras, kuris reikalingas pastato altitudei nuo žemės paviršiaus ir stogo formos parametrams gauti. Tai atliekama iš skaitmeninio paviršiaus modelio atimant skaitmeninio reljefo modelio rastro ląstelių vertes.



4 paveikslas. Sugeneruotas DSM rastras



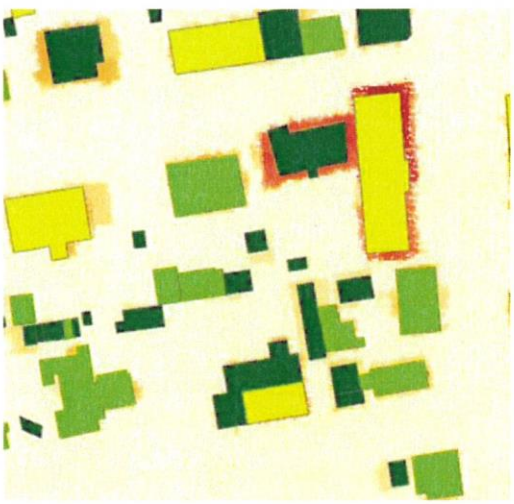
5 paveikslas. Sugeneruotas DTM rastras

6 paveikslas. Sugeneruotas nDSM rastras

Gautas normalizuotas paviršiaus modelio (angl. *normalised Digital surface model*, nDSM) rastras (6 paveikslas), kurio žemiausia vertė turi būti kuo artimesnė 0 – ji nurodo pastato nulinę altitudę, aukščiausia ląstelės vertė nusako pastato aukštį nuo žemės paviršiaus.

Kitam žingsniui svarbu turėti pagrindą, kuris bus naudojamas atributinei informacijai saugoti apie 3D objekto parametrus. Tai gali būti topografinių duomenų rinkiniai, vektorizuotos ortofotografinės nuotraukos pagrindu sudaryti poligonai arba gauto rastro (nDSM) konvertavimas į poligonus (7 paveikslas).

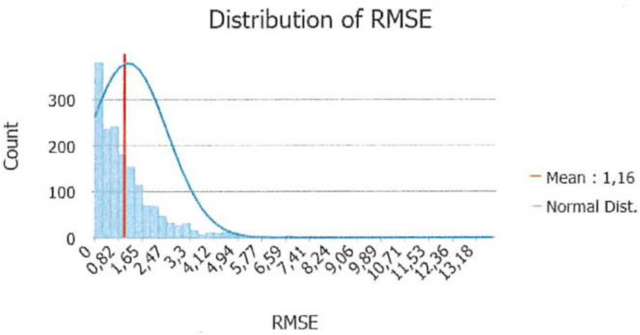
7 paveikslas. Analizei naudojami pastatų poligonų sluoksniai



8 paveikslas. Rastro ir esamo poligono palyginimas

Šiuo atveju naudojant esamus topografinius duomenis ir GIS įrankius apskaičiuoti pagrindiniai pastato ir stogo parametrai: pastato aukštis, aukštis iki karnizo, stogo formos ir nuolydžio kampai, pastato nulinė altitudė. Šiurkščioms klaidoms išvengi taip pat apskaičiuota vidutinė kvadratinė klaida (angl. *Root Mean Square Error*, RMSE). Statistiškai šis rodiklis įvertina gautus poligono sluoksnius sukuriant juos iš normalizuoto rastro nDSM ir lyginant jo atitikmenį su esamais duomenimis, pvz., gautais iš topografinių duomenų bazės. Atvaizduojant poligonus pagal šias reikšmes, galima juos keisti naudojant standartinius redagavimo įrankius (8 paveikslas).

Bendras analizuojamų plotų skaičius siekia 1677. RMSE rodikliai analizuojamoje teritorijoje svyruoja nuo 0 iki 13,73 (9 paveikslas). Kuo mažesnis šis rodiklis, tuo didesnis sutapimas tarp esamų ir gautų generuojant juos ir rastro. Vidutinis šio koeficiento rodiklis siekia 1,16.



9 paveikslas. RMSE rodiklių verčių pasiskirstymas

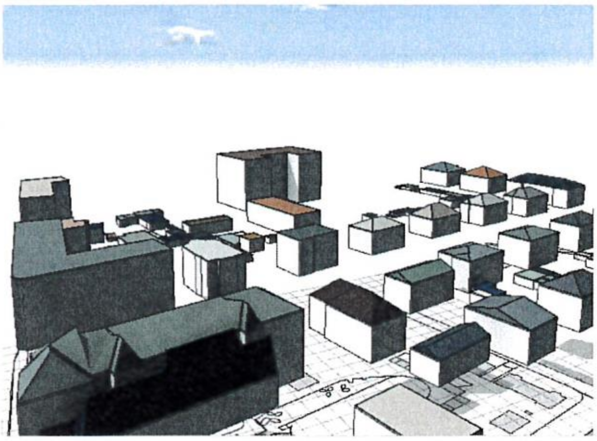
Procedūrinis modeliavimas

Apskaičiavus minėtus rodiklius sukurtas procedūrinio modeliavimo taisyklės. Tam naudojama kita programinė įranga – ESRI CityEngine. Ši PĮ skirta 3D miestui modeliuoti ir vizualizuoti naudojant kompiuterio generavimo architektūros (angl. *Computer Generated Architecture,* CGA) taisykles (10 paveikslas). Šio 3D modeliavimo idėja – apibrėžti taisykles, kurios pakartotinai tobulina dizainą, sukuriant vis daugiau detalesnių objektų.

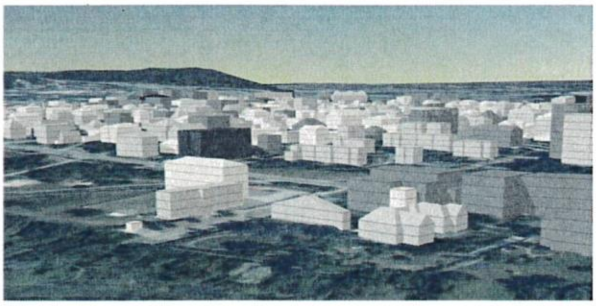
```
const FloorLineThicknessconst = FloorLineThickness
const roofHeight = ( _BLDGHEIGHT - _EAVEHEIGHT )
const facadeHeight = case _EAVEHEIGHT > 0 : _EAVEHEIGHT else :_BLDGHEIGHT
const floorCountInt = case AUKSTIS < 3: 1 else: AUKSTIS
const floorHeightFloat = facadeHeight / floorCountInt
const floorHeight = floorHeightFloat
const floorCount = AUKSTIS


@startrule
Lot -->
    t(0,( _BASEELEVATION -(scope.elevation)), 0)
    Extrusion

Extrusion -->
    cleanupGeometry(all,0.5)
    alignScopeToAxes(y)
    s('1,0,'1)
    extrude(facadeHeight)
    Building

LotInner --> NIL

# split the building geometry into its facade components
Building -->
        comp(f) { front : FrontFacade | side : SideFacade | top : Roof }

# the front facade is subdivided into one front groundfloor
```

10 paveikslas. CGA taisyklės pavyzdys

Šios teritorijos 3D atvaizduoti CGA taisyklėje nurodyti pagrindiniai parametrai. Pastato nulinė altitudė, nuo kurios generuojamas pastato aukštis vertikalia kryptimi, stogo formos tipai ir nuolydžiai. Pritaikius naujai sukurtą taisyklę atvaizduoti gauti 3D modeliai (11 paveikslas). Taisyklių kūrimo metu galima intuityviai keisti gautus atributinius parametrus, atlikti matematinius veiksmus, pritaikyti tekstūras sukurtiems objektams. Taisyklė taikoma esamiems plotams peržiūrėti ir įsitikinti, ar nėra padaryta klaidų pačioje taisyklėje



11 paveikslas. CGA taisyklės pritaikymo poligonams pavyzdys Nr. 1



12 paveikslas. CGA taisyklės pritaikymo poligonams pavyzdys Nr. 2

Sukurta taisyklė eksportuota į *.RPK (angl. *Rule package*) tipo failą. Trimatėje GIS aplinkoje ši taisyklė pritaikyta topografiniams plotams atvaizduoti (12 paveikslas).

Išvados

1. Naudojant esamus LiDAR duomenis ir GIS sugeneruoti 3D pastatai, kurie naudingi atliekant įvairias analizes (vizualines, matomumo, insoliacijos, projektinių statinių, aplinkos ir kt.). Šie procesai gali būti automatizuoti.

2. Sukurtos taisyklės gali būti naudojamos turint naujus ar kitos teritorijos LiDAR duomenis.

3. Tobulinant taisykles gaunami rezultatai gali būti tikslesni ir realistiškesni.

Padėkos

Straipsnio autoriai dėkoja SĮ „Vilniaus planui" už suteiktus topografinius duomenis.

Literatūra

Crosby, C. (2016). *Introduction to Lidar, Terrestrial Laser Scanning Applications*. Prieiga per internetą: https://cloud.sdsc.edu/v1/AUTH_opentopography/www/shortcourses/16UNAVCO/16UNAVCO_Crosby_introALS_TLS.pdf

Žalnierukas, A. ir Čypas, K. (2006). Žemės skenavimo lazeriu iš orlaivio technologijos analizė. *Geodesy and Cartography*, *32*(4), 101-105.

Žalnierukas, A., Ruzginė, B., Kalantaitė. A. ir Valaitienė, R. (2009). Miestų skenavimo LIDAR metodu tikslumo analizė. *Geodezija ir Kartografija*, *35*(2), 55-60. https://doi.org/10.3846/1392-1541.2009.35.55-60

3D BUILDING MODELING BY LIDAR DATA AND GIS TECHNOLOGIES

E. Streikus, E. K. Paršeliūnas

Summary

LiDAR (Light Detection and Ranging) is an ever-growing way of collecting data on ground surfaces using an aircraft-mounted laser. The main advantage of this method is the rapid acquisition of 3D data in a large area. The captured 3D data can be used to recognize, analyze, or map objects of topographic expressions on a 3D map. The LiDAR scan data for 2007 was used to create 3D buildings, the initial data processing was performed by LAStools (rapidlasso GmbH), the final 3D building models developed using ESRI products – ArcGIS Pro and CityEngine.

Keywords: Geographic Information System, LiDAR, Digital Relief Model, Digital Surface Model.