



Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas

Objektų vektorizavimas naudojant erdvinių žemės paviršiaus lazerinio skenavimo duomenis

Baigiamasis magistro projektas

Paulius Savickas
Projekto autorius

dr. Mindaugas Kavaliauskas
Vadovas
dr. Kristina Kundelienė
Vadovė

Kaunas, 2018



Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas

Objektų vektorizavimas naudojant erdvinių žemės paviršiaus lazerinio skenavimo duomenis

Baigiamasis magistro projektas
Didžiųjų verslo duomenų analitika (621G12002)

Paulius Savickas
Projekto autorius

dr. Mindaugas Kavaliauskas
Vadovas
dr. Kristina Kundelienė
Vadovė

dr. Evaldas Vaičiukynas
Recenzentas
dr. Evaldas Stankevičius
Recenzentas

Kaunas, 2018



Kauno technologijos universitetas
Matematikos ir gamtos mokslų fakultetas
Paulius Savickas

Objektų vektorizavimas naudojant erdvinių žemės paviršiaus lazerinio skenavimo duomenis

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Pauliaus Savicko, baigiamasis projektas tema „Objektų vektorizavimas naudojant erdvinių žemės paviršiaus lazerinio skenavimo duomenis“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Turinys

Įvadas.....	12
1. Literatūros apžvalga	14
1.1. Praktinis panaudojamumas	14
1.2. Žemės paviršiaus skenavimo lazeriu iš orlaivio technologijos analizė	15
1.2.1. Vietovės paviršiaus skenavimas lazeriu	16
1.2.2. Lazerio skenerių tipai	18
1.2.3. Tikslumo analizė	19
1.3. Programinės įrangos apžvalga	20
1.4. Vektorizavimo metodų ir algoritmų apžvalga.....	22
1.4.1. Pastatų kraštinių gavimas taikant „Alpha Shapes“ algoritmą	22
1.4.2. Pastatų kraštinių gavimas taikant trianguliaciją / Voronojaus diagramas	24
1.4.3. Mažiausio ploto stačiakampio algoritmas	26
1.5. Algoritmų objektų karšinėms normalizuoti apžvalga.....	28
1.5.1. Kreivės aproksimacijos „Douglas-Peucker“ algoritmas	28
1.5.2. Figūros kampų aptikimo ir lokalizavimo „cornerity“ indekso algoritmas	29
1.5.3. Patobulintas „Douglas-Peucker“ algoritmas pastatų kraštiniams normalizuoti	31
1.6. Sprendinių optimizavimo ir kokybės bei paklaidos nustatymo metodų ir algoritmų apžvalga	32
1.6.1. Pareto fronto daugiakriterinis optimizavimo sprendimas.....	32
1.6.2. Žakardo indekso metodas aibių panašumui nustatyti.....	33
1.7. Literatūros apibendrinimas	34
2. Medžiagos ir tyrimų metodai.....	35
2.1. Tyrimui naudoti duomenys.....	36
2.2. Duomenų paruošimas	37
2.3. Lazerinio skenavimo taškų klasifikavimas.....	37
2.3.1. Žemės paviršiaus lygio nustatymas	38
2.3.2. Aukščio nustatymas taškams kurie nepriskirti jokiai klasei.....	42
2.3.3. Taškų klasifikavimas į pastatų arba aukštos augmenijos klases.....	42
2.4. Objektų kraštinių išgavimas	44
2.4.1. Objektų kraštinių išgavimas taikant „Alpha Shapes“ algoritmą visai imčiai iš karto	45
2.4.2. Objektų kraštinių išgavimas taikant „Alpha Shapes“ algoritmą kiekvienam objekto klasteriui atskirai	47

2.4.3. Objektų kraštinių išgavimas taikant mažiausio ploto stačiakampio algoritmą	50
2.5. Objektų kraštinių normalizavimas.....	50
2.5.1. Objektų kraštinių normalizavimas taikant RDP algoritmą.....	50
3. Tyrimų rezultatai ir jų aptarimas	52
3.1. Lazerinio skenavimo taškų klasifikavimo rezultatai ir jų aptarimas.....	52
3.1.1. Žemės paviršiaus lygio nustatymo rezultatai ir jų aptarimas	52
3.1.2. Pastatų ir medžių taškų nustatymo rezultatai ir jų aptarimas	53
3.2. Objektų kraštinių išgavimas	55
3.2.1. Objektų kraštinių išgavimas taikant „Alpha Shapes“ algoritmą visai imčiai iš karto	56
3.2.2. Objektų kraštinių išgavimas taikant „Alpha Shapes“ algoritmą kiekvienai taškų aibei atskirai	57
3.2.3. Objektų kraštinių išgavimas randant įgaubtąją daugdarą kiekvienai taškų aibei atskirai	58
3.2.4. Objektų kraštinių išgavimas taikant mažiausio ploto stačiakampio algoritmą	58
3.2.5. Objektų kraštinių išgavimo apibendrinimas	60
3.3. Objektų kraštinių normalizavimas.....	61
3.3.1. Objektų kraštinių normalizavimas taikant RDP algoritmą.....	61
3.3.2. Objektų kraštinių normalizavimas taikant Guru's „cornerity“ indekso algoritmą	63
3.3.3. Objektų kraštinių normalizavimo apibendrinimas	64
3.4. Ekonominis pagrindimas	65
3.4.1. Kylančios „Google Maps“ paslaugų kainos	65
3.4.2. Palydovinių vaizdų atpažinimo konkursas	66
Išvados	67
Literatūros sąrašas	68
Priedai.....	72
1 priedas. Rasti „Alpha Shapes“ optimalieji sprendiniai objektų kraštinių išgavimui	72
2 priedas. Rasti optimalūs sprendiniai objektų kraštinių normalizavimui taikant RDP algoritmą	72

Paveikslų sąrašas

1 pav. Skenavimo lazeriu principas: a – lazerio taškų paskirstymo žemės paviršiuje pėdsakas kreipiant spindulį veidrodžiu [1]	16
2 pav. Lazeriu nuskenuotų „taškų debesis“: Ringaudų gyvenvietė.....	17
3 pav. Vaizdai © CNES / Airbus, 2018 m., Žemėlapiu duomenys © Google, 2018 m. Ringaudų gyvenvietė.....	17
4 pav. Žemės paviršiaus pagrindu grįsta LiDAR sistema [9].....	18
5 pav. Kosminė LiDAR sistema	19
6 pav. „Alpha Shapes“ algoritmo principas [18].....	23
7 pav. Figūros detalumo lygis kintant α reikšmei.....	24
8 pav. LiDAR duomenys išreikšti Varanojaus diagrama	24
9 pav. Tikriniai vektoriai su mažiausiomis tikrinėmis reikšmėmis [19]	25
10 pav. Pastato segmentai kiekvienoje Voronojaus celėje [19]	25
11 pav. Rastos dvi stogo plokštumos [19].....	26
12 pav. 3D pastato modelis pritaikius Voronojaus diagramų metodą [19]	26
13 pav. Mažiausio ploto stačiakampis duotai taškų aibei.....	27
14 pav. Slankmačiais apribotas daugiakampis [21].....	28
15 pav. Algoritmo parinkti kampai su skirtingais pradžios taškais [23]	29
16 pav. Aptikti kampai su skirtingu parinktu lango pločiu [23].....	31
17 pav. Kampų aptikimo veiksmų seka taikant Li patobulintą „Douglas-Peucker“ algoritmą	32
18 pav. Pareto optimalieji sprendiniai	33
19 pav. Žakardo indeksas daugiakampių panašumui nustatyti.....	34
20 pav. Tyrimų etapai siekiant kartografuoti vietovę naudojant LiDAR duomenis	35
21 pav. Tyrimui naudoti NŽT LiDAR duomenys	36
22 pav. Parametrų derinimo procesas, žemės paviršiaus taškams nustatyti	38
23 pav. Taškai priskirti žemės paviršiui esantys tarp medžių	40
24 pav. „OpenStreetMap“ informacinės sistemos žemėlapiu žymėjimo netikslumai.....	41
25 pav. Paslinkusi „Bing“ ortofoto nuotrauka [31]	42
26 pav. Parametrų derinimo procesas, pastatų ir medžių taškams nustatyti.....	43
27 pav. „Alpha Shapes“ algoritmo parametrų derinimo procesas pastų kraštinėms apibrėžti visai lazerinio skenavimo taškų aibei iš karto.....	45
28 pav. „Alpha Shapes“ algoritmo klaida esant dideliu slenksčiui	47
29 pav. „Alpha Shapes“ algoritmo klaida esant mažam slenksčiui.....	47

30 pav. „Alpha Shapes“ algoritmo parametrų derinimo procesas, objektų kraštinėms apibrėžti kiekvienam klasteriui atskirai.....	49
31 pav. RDP algoritmo parametrų derinimo procesas.....	51
32 pav. Pareto frontas, pastatų ir aukštos augmenijos taškų klasifikavimo programos sprendiniams	54
33 pav. Grafiškai atvaizduoti suklasifikuoti LiDAR taškai.....	55
34 pav. Rasti „Alpha Shapes“ optimalieji sprendiniai objektų kraštinių išgavimui.....	56
35 pav. Išgautos pastatų kraštinės mažiausio ploto stačiakampio algoritmu	60
36 pav. Rasti optimalūs sprendiniai objektų kraštinių normalizavimui taikant RDP algoritmą	61
37 pav. Išgautos pastatų kraštinės taikant „Alpha Shapes“ algoritmą.....	62
38 pav. Normalizuotos pastatų kraštinės taikant RDP algoritmą	63
39 pav. Normalizuotos pastatų kraštinės taikant Guru „cornerity“ indekso algoritmą	64

Lentelių sąrašas

1 lentelė. Terminų ir santrumpų žodynas	11
2 lentelė. LiDAR jutiklių palyginimas	14
3 lentelė. LIDAR programinė įranga ir įrankiai [10].....	22
4 lentelė. Suklasifikuotų grupių duomenų reikšmės pagal ASPRS standartą	37
5 lentelė. Rasti optimalieji sprendiniai žemės taškų klasifikavimo kokybės parametrų.....	52
6 lentelė. Rasti Pareto fronto optimalūs sprendiniai pastatų taškų klasifikavimo kokybės parametrų	53
7 lentelė. Rasti „Alpha Shapes“ optimalūs sprendiniai objektų kraštinių išgavimui	57
8 lentelė. Rasti optimalūs sprendiniai objektų kraštinių išgavimui taikant įgaubtosios daugdaros algoritmą.....	58
9 lentelė. Rasti optimalieji sprendiniai objektų kraštinių išgavimui taikant mažiausio ploto stačiakampio algoritmą.....	59
10 lentelė. Algoritmų rezultatų kraštinėms išgauti palyginimas	60
11 lentelė. Optimalieji sprendiniai kraštinių normalizavimui taikant Guru „cornerity“ indekso algoritmą.....	63
12 lentelė. Algoritmų rezultatų kraštinėms išgauti palyginimas	65

Savickas, Paulius. Objektų vektorizavimas naudojant erdvinių žemės paviršiaus lazerinio skenavimo duomenis. Magistro baigiamasis projektas / vadovai dr. Mindaugas Kavaliauskas, dr. Kristina Kundelienė; Kauno technologijos universitetas, matematikos ir gamtos mokslų fakultetas.

Studijų kryptis ir sritis: Taikomoji matematika (A02), Matematikos mokslai (A).

Reikšminiai žodžiai: *Objektų vektorizavimas; pastatų kraštinių išgavimas; LiDAR.*

Kaunas, 2018. 71 p.

Santrauka

Nuolat kintant žemės paviršiui sunku visada žinoti realią situaciją apie didėjantį ar mažėjantį miškų plotą ar naujus pastatytus pastatus. Taip pat sunku užtikrinti jog ši, reali situacija, teisingai atvaizduota žemėlapiuose. Šiuos duomenis turi pastoviai fiksuoti bei registruoti tų sričių specialistai. Dažnai tai nebūna lengva užduotis ir yra daromi apytiksliai skaičiavimai.

Šiais laikais vis labiau tobulėjant mašininio mokymosi algoritmams bei įrankiams juos naudoti, taip pat pasitelkiant tokias technologijas kaip LiDAR (angl. Light Detection and Ranging) duomenims gauti, galima sukurti algoritmą, kuris galėtų išspręsti šias problemas.

LiDAR technologija yra naudojama iš orlaivio matuoti žemės paviršiaus ir jame esančių objektų aukščiui centimetrų tikslumu. Iš gautų duomenų, taikant šią technologiją, šiame baigiamajame darbe sukurtas algoritmas, kuris kartografuoja vietovę. Algoritmas atpažįsta žemės paviršių, pastatus, aukštą augmeniją. Vektorizuoja atpažintus žemės paviršiaus objektus.

Savickas, Paulius. Objects Vectorization Using Spatial Earth Surface Laser Scan Data. Master's Final Degree Project / supervisors assoc. dr. Mindaugas Kavaliauskas, dr. Kristina Kundelienė . The Faculty of Mathematics and Natural Sciences, Kaunas University of Technology.

Research area and field: Applied Mathematics (A02), Mathematical Sciences (A).

Key words: *Objects vectorization; building boundary extraction; LiDAR.*

Kaunas, 2018. 71 p.

Summary

It is difficult to always know the real situation regarding the increasing or decreasing forest area or newly constructed buildings on the ever-changing surface of the earth. It is difficult to ensure that this real situation is correctly represented on the maps. These data must be constantly recorded by specialists in those fields. Often this is not an easy task and rough estimates are made.

Nowadays, with the ever-expanding use of machine learning algorithms and tools, and with the use of technologies such as the light detection and ranging (LiDAR) to gather data, it is possible to create an algorithm that can solve these problems.

The LiDAR technology is used to measure the ground level and objects in the ground from airplane with an accuracy of centimeters. From this technology, in this final work, an algorithm prototype is created which maps the location automatically. The algorithm recognizes and vectorise the earth's surface, buildings and forests.

Terminų ir santrumpų žodynas

1 lentelė. Terminų ir santrumpų žodynas

LiDAR (angl. Light Detection and Ranging)	Metodas informacijai apie erdvinį žemės paviršių kaupti, vietovės skenavimo (zondavimo) lazeriu iš orlaivio [1]
3D (angl. three-dimensional)	Trimatė erdvė
RADAR (angl. Radio wave Detection and Ranging)	Radaras - įrenginys objektams erdvėje aptikti, koordinatėms nustatyti, matuoti objekto judesio parametrus radiolokacijos metodais.
DEM (angl. Digital Elevation Model)	Skaitmeninio aukščio modelis – 3D reljefo paviršiaus modelis [2]
Geometrinis primityvas	Geometriniai primityvai yra neskaidomi objektai, teikiantys informaciją apie geometrinę konfigūraciją. Tai gali būti taškai, kreivės, paviršiai, trimačiai kūnai.
NŽT	Nacionalinė žemės tarnyba
Geomorfologinis	Žemės plutos dalies pasikeitimas dėl geologinių veiksnių, pvz., tekamojo vandens, ledo, vėjo.
ASPRS (angl. American Society for Photogrammetry and Remote Sensing)	Amerikos fotogrametrijos ir nuotolinio stebėjimo draugija
Ortofoto	Žemės paviršiaus nuotrauka iš oro skirta topografinių žemėlapių sudarymui, miškotvarkos ir žemėtvarkos darbams.
CSV (angl. comma - separated values)	Bylų tekstinis formatas, kuriuose lentelių duomenys saugomos kableliais atskirtomis reikšmėmis.
GPS (angl. Global Positioning System)	Globali padėties nustatymo sistema.
GIS (angl. geographic information system)	Geografinė informacinė sistema.

Įvadas

Žemės paviršiaus lazerinio skenavimo (zondavimo) iš orlaivio metodas, žinomas kaip LiDAR, kuris suteikia modeliavimo ir tyrimo galimybes, kurios taikomos žemės paviršiaus, reljefo pastatų, miškų ir kitų objektų atpažinimui. Šiuo metodu gaunamas taškų debesis yra netolygus, nereguliarus, labai tankus, tai trijų dimensijų žemės paviršiaus objektų išdėstymą modeliuojanti geometrinė ir radiometrinė informacija [1, 3]. Dažniausiai šis metodas taikomas, kuriant aukštos raiškos žemėlapius su pakraipa į geodeziją, geomatiką, archeologiją, geografiją, geologiją, geomorfologiją, miškininkystę ir kt.

Informacinių technologijų progresas išplėtė LiDAR taikymo galimybes ir sričių įvairovę. Kasmet didėja ir besidominčių skaičius, norinčių pritaikyti LiDAR metodu gautus duomenis, savo darbo ar mokslo srityse. Tokie duomenys taikomi inovatyviuose projektuose tokiuose kaip autonominiai automobiliai, kuriuose šio metodo pagalba yra nustatomi aplinkui esantys objektai realiu laiku, ar virtualios realybės aplinkų kūrimui.

Skaitmeninės fotografijos bei piešiniai dažniausiai kompiuteriuose yra saugomi kaip taškiniai vaizdai. Vaizdai, kurie yra sudaryti iš individualių vaizdo elementų, atsižvelgiant į detalumo lygį, gali būti kartu privalumas ir trūkumas. Didesnio detalumo vaizdas yra aukštesnės raiškos, todėl užima daugiau vietos kietajame diske. Taip pat atliekant įvairius veiksmus su taškiniiais duomenimis pailgėja operacijų atlikimo laikas. Vektorinis formatas kompiuterinėje grafikoje yra žymiai patogesnis ir tinkamesnis saugoti, keistis bei atvaizduoti duomenis.

Vienas iš pagrindinių uždavinių – rasti metodą, kuris leistų nustatyti, LiDAR technologija surinktuose taškiniuose duomenyse esančius objektus bei juos vektorizuoti. Tokiu būdu pritaikyti gautus duomenis kartografuojant žemėlapius.

Darbo problematika ir aktualumas

Viena didžiausių šių dienų logistikos įmonių, naudojančių internetinius žemėlapius, problemų yra gauti kuo tikslesnę žemėlapių informaciją, tuo pačiu nepatiriant didelių kaštų mokant už šias paslaugas. Taip yra todėl, jog tokios įmonės dažnai taiko informacines sistemas, kuriuose planuojami maršrutai, paskirstomi turimi resursai, valdant maršruto apkrautumą, taikant internetinio žemėlapių paslaugas. Internetinių žemėlapių netikslumas įmonėm sukelia dideles problemas tokias kaip vėluojantys krovinių pristatymai klientams, didesnės transporto priemonių kuro sąnaudos bei didesnės vairuotojų darbo valandos. Įmonės siekdamos užkirsti kelią šioms problemoms perka

internetinio žemėlapio paslaugas iš tokių įmonių kaip „Google“ ar „Microsoft“, todėl atsiranda papildomos išlaidos. Pasaulyje yra sukurtų daug nemokamų, lengvai prieinamų internetinių žemėlapių, tokių kaip „OpenStreetMap“, tačiau tokie žemėlapiai nėra tikslūs, nes sistemoje objektai (keliai, pastatai, medžiai ir t.t.) yra apibrėžiami vartotojų, todėl natūralu, kad daugelis objektų nėra sužymėti.

Darbo tikslas ir uždaviniai

Tikslai

- Sukurti algoritmą, kuris kartografuoja vietovę, taikant LiDAR duomenis, atpažįsta žemės paviršių, pastatus bei aukštą augmeniją, atpažintus objektus vektorizuoja dvimatėje erdvėje.

Uždaviniai

- Rasti metodą klasifikuoti LiDAR duomenis.
- Rasti metodą klasterizuoti atskirus objektus LiDAR duomenyse.
- Atlikti taškinių duomenų vektorizavimo metodų analizę.
- Vektorizuoti objektus ir įvertinti jų tikslumą.
- Įvertinti realizuotą algoritmą, jo spragas ir privalumus.

Darbo rezultatai ir jų svarba

Suradus metodą iš LiDAR duomenų identifikuoti atskirus objektus bei juos vektorizuoti, šio metodo rezultatus galima panaudoti egzistuojančiuose „OpenStreetMap“ žemėlapių sprendimuose, pagerinant jų tikslumą ir detalumą.

1. Literatūros apžvalga

Šiame skyriuje analizuojama mokslinė literatūra bei bendra informacija apie LiDAR technologija, jos duomenų galimus klasifikavimo, klasterizavimo, vektorizavimo metodus ir algoritmus. Apžvelgiamos esamos technologijos skirtos darbui su LiDAR duomenimis. Pagal nusistatytus kriterijus pasirenkami metodai ir algoritmai. Apžvelgiami ar pasirinkti vektorizavimo metodai, algoritmai tinkami dirbti su LiDAR duomenimis. Šiame skyriuje bus apžvelgiami moksliniai straipsniai, vadovėliai, konferencijų medžiaga bei kita.

1.1. Praktinis panaudojamumas

Kompanija „Business Wire“ savo straipsnyje [4] teigia jog nuo 2016 m., kai pasaulinė LiDAR rinka sudarė 1 058,40 milijono JAV dolerių, ši rinka iki 2024 m. turėtų pasiekti 5 842,37 milijono JAV dolerių. Šio prognozuojamo laikotarpio nuo 2017 m. iki 2024 m. sudėtinė metinė augimo norma sudarys 24,1 %. Pagrindiniai faktoriai nulemiantys augimą rinkoje yra valstybių iniciatyva ir augantis prieinamumas, dronų panaudojimas įvairiuose srityse, augantis LiDAR projektų skaičius bei auganti technologijos pažanga. Tačiau proveržį kol kas šiek tiek stabdo didelė šios technologijos kaina.

Wang'as ir kiti autoriai [5] straipsnyje teigia jog šiuo metu svarbiausia tendencija yra sukurti pigias 3D skenavimo technologijas, kurios galėtų įsigyti paprasti vartotojai. Šiuo metu dažniausiai naudojami LiDAR jutikliai negali patenkinti šio poreikio. Atsižvelgdami į šią problemą šie autoriai pasiūlė savo sukurtą „RPLIDAR“, jų teigimu, pigų jutiklį, kuris galėtų būti taikomas vidaus sąlygomis ir palygino su kitais LiDAR jutikliais (2 lentelė.).

2 lentelė. LiDAR jutiklių palyginimas

	Hokuyo URG- 04LX	Lase ODS 1600 HT 2- select	AccuRange 4000-LIR	LMS200	Ibeo ALASCA XT	RPLIDAR
Kaina (\$)	1995	16100	9730	7605	13644	398,9
Diapazono atstumas (m)	0,02–5.6	0,8–2,4	Daugiau nei 16,5	8,19	0,3–200	0,2–6
Diapazono tikslumas (mm)	±10 arba 10 %	0,1	2,54	10	-	< 0,5

Bangos ilgis (mm)	785	670	780	905	905	775–795
Svoris (kg)	0.16	1,60	0,62	4,5	3,0	0,2

Deems'as [6] ir kiti autoriai straipsnyje teigia jog LiDAR suteikia vilčių naujiems atradimams hidrologijos srityje. Modernių jutiklių pagalba, taikant LiDAR technologiją, yra galimybė nustatyti iškritusio sniego kiekį kalnuotose arba (ir) miškinguose vietovėse. Šia technologija būtų galima atisakyti daug laiko atimančio rankinio mėginių ėmimo, nustatinėjant sniego kiekį. Ypač tai sunku nustatyti prašliaužus sniego lavinai kalnuotose vietovėse. Šių autorių pagrindinė idėja yra nustatyti žemės paviršiaus modelį be sniego ir esant sniegui. Išmatavus atstumą tarp šių dviejų paviršiaus modelių galima tiksliai nustatyti sniego kiekį kiekviename taške.

Pritaikę LiDAR technologiją Lukačas [7] ir kiti autoriai pasiūlė galimybę miesto zonose automatiškai nustatyti ilgą laiką saulės apšviečiamus pastatų stogus, įvertinant kitų pastatų bei medžių šešėlio įtaką. Pasaulyje yra sukurtų įvairių metrikų nustatinėjant fotovoltinių sistemų efektyvumui. Daugelį atvejų šie skaičiavimai yra atliekami esant puikioms oro sąlygoms bei maksimaliam saulės apšvietimui ir yra pateikiami kaip konstanta, tačiau realybėje taip nėra. Miesto zonose šioms sistemoms didelę įtaką daro aplinkui esantys pastatai ir medžiai, jų šešėliai. Pritaikius LiDAR sistemą šių šešėlių nustatymui yra galimybė apskaičiuoti tikslų fotovoltinės sistemos efektyvumą ją įrengiant ant vieno ar kito stogo.

Autoriai Du ir kiti bendraautoriai [8] fotogrametrijos ir nuotolinio stebėjimo žurnale teigia, kad automatinis pastatų kraštinių išgavimas iš LiDAR duomenų yra pagrindas programoms skirtoms trimatės erdvės (3D pagal angl. three-dimensional) pastatų modeliavimui, miesto planavimui, pavojų įvertinimui, skaitmeninių žemėlapių ir geografinių informacinių sistemų (GIS pagal angl. geographic information system) duomenų bazių atnaujinimui. Tai tapo viena aktualiausių temų tyrimuose, tokiuose kaip naujų jutiklių atradimuose bei pažangiuose apdorojimo technologijose. Tačiau pastatų kraštinių išgavimas trimatėje erdvėje vis dar tebėra sudėtingas uždavinys.

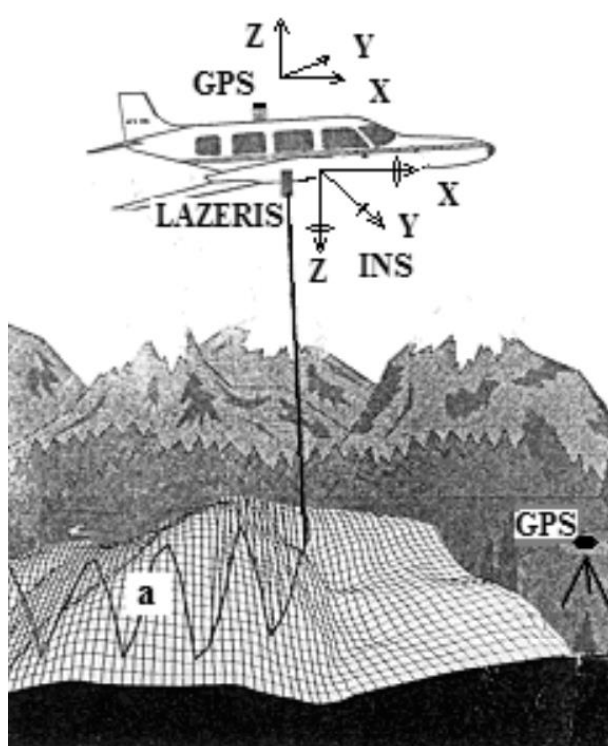
1.2. Žemės paviršiaus skenavimo lazeriu iš orlaivio technologijos analizė

LiDAR arba trimatės erdvės 3D skenavimo metodas buvo sukurtas 1960 metais povandeninių laivų aptikimui iš orlaivio. Šiais laikais aplinkos tyrimai yra sunkiai įsivaizduojami be nutolusiu stebėjimu grįstų metodų bei technologijų tokių kaip LiDAR ar radaro (RADAR pagal angl. Radio wave

Detection and Ranging). Aukštos rezoliucijos bei tikslumo duomenys suteikia galimybę stebėti supančią aplinką iš daugiau nei 100 km siekiančio aukščio virš jūros lygio [1, 9].

1.2.1. Vietovės paviršiaus skenavimas lazeriu

Lazerinio skenerio veikimo principas yra gana paprastas. Lazerinis skeneris įtaisomas orlaivyje. Skrendant orlaiviui registruojami sugeneruoti optinio elektroninio bloko lazerio impulsai ir jo aktyviais jutikliais išmatuojamas atstumas iki žemės paviršiaus, medžių lajos, pastatų stogų ir kitų objektų (1 pav.).



1 pav. Skenavimo lazeriu principas: a – lazerio taškų paskirstymo žemės paviršiuje pėdsakas kreipiant spindulį veidrodžiu [1]

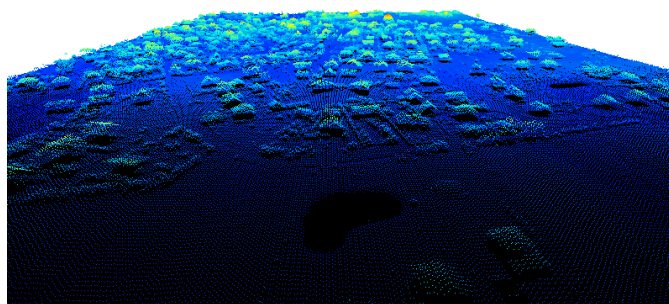
Norint išmatuoti atstumą iki objekto paviršiaus, skaičiuojama kiek laiko šviesos fotonas keliavo nuo orlaivio iki objekto. Tai galime apskaičiuoti naudojant šią formulę:

$$Atstumas = \frac{\text{Šviesos greitis} \times \text{Laikas}}{2} \quad (1)$$

Erdvinės polinės koordinatės nustatomos skenavimo lazeriu sistema. Jų pradžių fiksuoja įrengta globali padėties nustatymo sistema (GPS pagal angl. Global Positioning System). Įrengta inercinė

navigacinė sistema (INS) (1 pav.) skirta nustatyti kintančiam polinių koordinatų sistemos orientavimui erdvėje XYZ ašių atžvilgiu. Lazerinis skeneris, GPS ir INS veikia sinchroniškai. Taip pat reikalinga bent viena GPS referencinė stotis su žinomomis krašto koordinatėmis ir žinomu geoido aukščiu virš elipsoido. 3 tradiciniai linijiniai ir 3 kampiniai aeronuotruokos absoliutinio orientavimo elementai gaunami iš GPS ir INS įrenginių [1].

Taikant LiDAR skenavimo metodą sukuriama trijų dimensijų lazerio taškų visumos modelis („taškų debesis“) (2, 3 pav.). Ši technologija suteikia galimybę skenuoti žemės ar kito objekto paviršių. Nuskenuotų taškų tankis labai didelis, viename kvadratiniam paviršiaus metre gali būti iki keleto, keliolikos ar kelių dešimčių taškų. Taip sukaupiami šimtai gigabaitų informacijos apie nuskenuotos aplinkos paviršių [1, 9].



2 pav. Lazериu nuskenuotų „taškų debesis“: Ringaudų gyvenvietė



3 pav. Vaizdai © CNES / Airbus, 2018 m., Žemėlapiu duomenys © Google, 2018 m. Ringaudų gyvenvietė

Pavieniai lazerio spinduliai praeina pro medžių tarpus esančius tankiuose miškuose ir taip pasiekia žemės paviršių. Tokiu būdu galima gauti žemės paviršiaus modelį miškingose vietovėse. Nustatyta, kad priklausomai nuo medžių tankumo, spygliuočių medžių miškuose vidutinė lazerio spindulių skvarba iki žemės yra apie 35 %, o lapuočių miškuose vasarą apie 25 %, žiemą – iki 70 %.

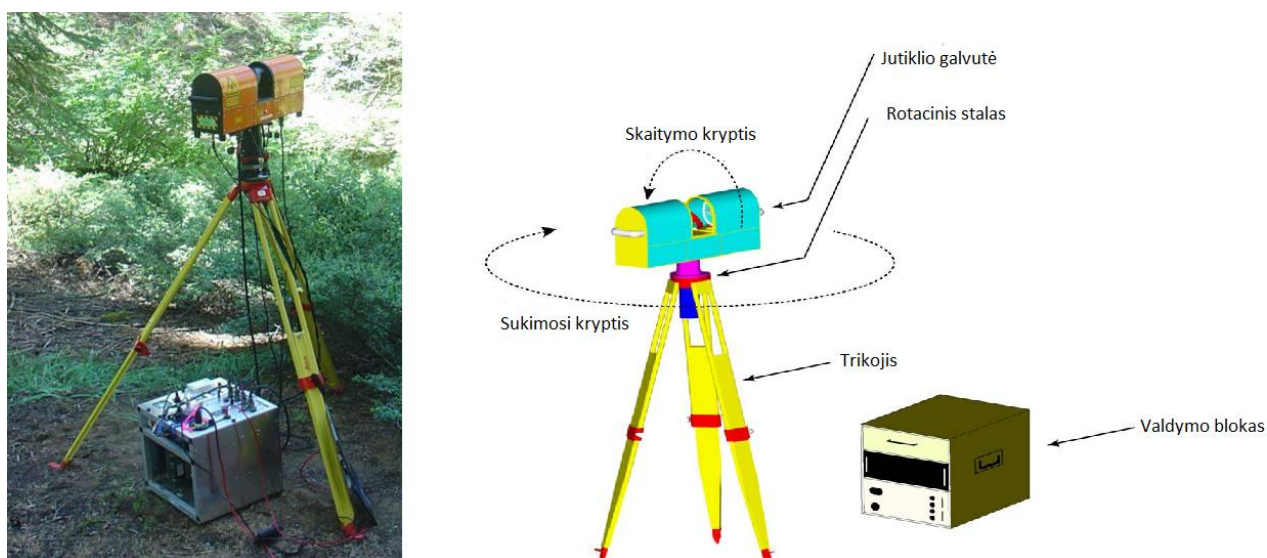
Skenuojant žemės paviršių lazeriu, gamtinių aplinkybių įtaka skenavimo kokybei nėra didelė. Matuoti galima apniukusią dieną, skrendant žemiau debesų, taip pat, bet kuriuo paros metu ar metų laiku. Didžiausią įtaką daro, priklausomai nuo lazerio šviesos šaltinio, migla bei rūkas. Lazeris gerai skenuoja šiuos paviršius: šviesų betoną, smėlį, jūros pakrančių ruožus, kartais vandenį, kitaip tariant, paviršius be tekstūros.

Taškų koordinatės užrašomos krašto koordinatinių sistemoje (Lietuvoje LKS 94), kurios yra gaunamos transformuojant, pagal žinomus parametrus bei geoido aukštį virš elipsoido, WGS 84 koordinatinių sistemoje užrašytas koordinatas. Šios būna perskaičiuotos iš polinių lazerinių taškų koordinatinių.

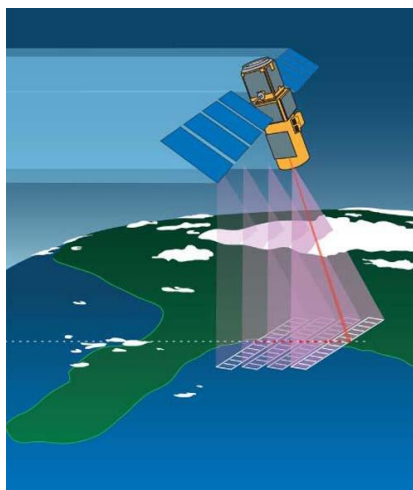
1.2.2. Lazerio skenerių tipai

LiDAR sistemos skirstomos į tris pagrindines sistemas:

- Žemės paviršiaus pagrindu grįstos LiDAR sistemos (4 pav.)
- Orlaiviais grįstos LiDAR sistemos (1 pav.)
- Kosminės LiDAR sistemos (5 pav.)



4 pav. Žemės paviršiaus pagrindu grįsta LiDAR sistema [9]



5 pav. Kosminė LiDAR sistema

1.2.3. Tikslumo analizė

Nuotolį iki kieto paviršiaus LiDAR lazerio spindulys išmatuoja 1–3 cm tikslumu. Albinas Žalnierukas [1] teigia:

vietovės skenavimo lazeriu tikslumą lemia skenerio savybės, skrydžio aukštis ir skenuojamojo ruožo plotis, skenerio pozicionavimo GPS ir orientavimo INS paklaidos, lazerio spindulių dažnis ir sklaida, atmosferos sąlygos ir ypač – atspindinčiojo paviršiaus savybės bei kiti veiksniai. Aukščių tikslumas (Z) didelis.

Taip pat šis autorius, savo straipsnyje, vidutinėms kvadratinėms žemės reljefo modelio, sudaryto pagal lazerio taškus, absoliutinių altitudžių paklaidoms σ_h apskaičiuoti rekomenduoja formulę:

$$\sigma_h = \pm \left(\frac{6}{\sqrt{n}} + 50 \operatorname{tg} \alpha \right) \text{cm}; \quad (2)$$

čia n – tinkamų taškų tankis, susijęs ir su skrydžio aukščiu (skaičius 1 m² plote); α – žemės paviršiaus nuolydis; 6 ir 50 – empiriniai parametrai.

Remiantis autoriaus straipsniu, vietovės paviršiaus modelio, sudaryto atliekant lazerinį skenavimą iš orlaivio, galima vidutinė kvadratinė aukščio paklaida 10–20 cm, o horizontalusis tikslumas – 1/1000–1/2000 skrydžio aukščio. Straipsnyje apie matavimo paklaidas teigiama: „Laikoma, kad lėktuvo kinematinė GPS pozicionavimo paklaida dėl dreifavimo yra apie ±10 cm (paklaida didėja ilgėjant skrydžio ruožui), o INS – ±0,01 gon, tai, esant 1000 m aukščiui, žemės paviršiuje sudarytų apie ±16 cm“ [1].

$$1 \text{ gon} = \left(\frac{\pi}{2}\right) 10 - 2 \text{ rad} \quad (3)$$

Čia gon – nesisteminis plokščiojo kampo vienetas.

1.3. Programinės įrangos apžvalga

Ramon Arrowsmith [10] savo straipsnyje aptaria kaip egzistuojančią programinę įrangą apdoroti LiDAR duomenis bei programinės įrangos poreikius, kurie dar nėra realizuoti. Autorius teigia, jog pasaulyje LiDAR duomenų apdorojimui reikalavimai programinei įrangai labai skirtingi, dėl didelės vartotojų bei jų poreikių įvairovės. Tačiau pirmieji žingsniai, atliekant LiDAR duomenų analizę, dauguma atvejų prasideda nuo taškų debesies, kuris buvo surinktas skenuojant lazeriu žemės paviršių. Todėl programinė įranga ir įrankiai turi palaikyti tokias bendras funkcijas kaip taškų debesies analizę, skaitmeninio aukščio modelio (DEM pagal angl. Digital Elevation Model) generavimą, kokybės užtikrinimą ir valdymą. Priklausomai nuo programos ar įrankio, DEM generavimas gali būti paskutinis žingsnis siekiant išsaugoti tikslumą atliekant aukščio ar kt. analizę. Nepaisant to kokiam procesui skirta programa ar įrankis, ji turėtų prisitaikyti prie daugumos vartotojų poreikių.

Dirbdami su LiDAR duomenimis, dauguma žmonių naudoja keletą programų ar įrankių, nes kiekviena turi savų pliusų ir minusų. Visi toliau aptarinėjami GIS arba nuotolinio stebėjimo programų paketai, kurie gali dirbti su taškiniais duomenimis, turi tam tikras LiDAR duomenų analizės ir atvaizdavimo funkcijas. Kaip matome lentelėje (3 lentelė.), programinės įrangos yra įvairios, nuo labai brangių komercinių programų, tokių kaip „Polyworks“ ar „TerraScan“, mažiau brangių, komercinių programų, tokių kaip „ArcGIS“ [10], nemokamų, komercinių, tokių kaip „LASTools“ [11], „GeomorphTools“ iki nemokamų atviro kodo programų, tokių kaip „GRASS“, „Points2Grid“, „LViz“ [12], „LidarViewer“ [13], „Real-time interactive Mapping System“.

„Points2Grid“ – tai geologų bei informacinių technologijų mokslininkų sukurtas, daug kompiuterio resursų nereikalaujantis įrankis DEM generavimui iš didelio kiekio LiDAR taškų debesies, naudojant vietinio elastingumo algoritmą (angl. local binning algorithm) [14].

„LViz“ – tai įrankis skirtas 3D LiDAR taškų debesies duomenų vizualizacijoms. Su šiuo įrankiu taip pat galima susieti duomenis su tekstūromis bei, vartotojo aplinkoje, valdyti šių duomenų atvaizdavimo nustatymus pvz., fono spalvą [12].

„LidarViewer“ – tai aplikacija skirta atvaizduoti ir atlikti skaičiavimus naudojant 3D taškų debesies duomenis. Programa buvo sukurta analizuoti milijardus LiDAR taškų išvengiant pasikartotino duomenų saugojimo, skaidymo mažesnėmis imtimis bei jų naikinimo, skaitmeninio paviršiaus

modeliavimo. Naudojant šią programą galima realiu laiku atvaizduoti interaktyvias vizualizacijas ir analizuoti duomenis geriau nei tai būtų galima atlikti naudojant vien kompiuterio pagrindinę atmintį. Tam pasiekti buvo optimizuotas duomenų valdymas pritaikant hierarchines duomenų struktūras. Saugant duomenis atsižvelgta į vaizdo priklausomybę nuo žiūrimo taško, kintamą skiriamąją gebą, algoritmus, skirtus apdoroti duomenis netelpančius į kompiuterio pagrindinę atmintį. Atvaizduotas taškų debesis visada gali būti pilnai manipuluojamas 3D erdvėje, todėl matavimai ir interpretacijos gali būti atliekamos iš skirtingų stebėjimo taškų. Interaktyvioje navigacijoje, „LidarViewer“ suteikia galimybę realiu laiku pasirinkti norimus taškus bei juos iškelti ar analizuoti atskirai. Taip pat išreikšti taškus geometriniais primityvais (linijomis, plokštumomis, figūromis, cilindrais), išmatuoti atstumą tarp jų bei nustatyti jų tikslią vietą [13].

„GRASS GIS“ – tai programa naudojama geografinių erdvių valdymui, analizei, nuotraukų generavimui, žemėlapių kūrimui, erdviniam modeliavimui ir atvaizdavimui. Taip pat ši programinė įranga turi pagrindinius įrankius LiDAR duomenų analizei.

„ArcGIS“ programos pagrindinė aplinka remiasi 2.5D kartografija ir daugelio pasaulio mokslininkų integruotais duomenimis. Tačiau 3D atvaizdavimo pajėgumas su dideliais LiDAR failais yra ribotas [10].

„QGIS“ – tai nemokama ir multiplatforminė geografinė, informacinė sistema. Ši programa suteikia geografinių bei erdviųjų duomenų peržiūrėjimo, redagavimo ir analizavimo galimybes. Ši programa yra labai universali, ji gali dirbti tiek su rastriniais tiek su vektoriniais duomenimis. Taip pat suteikia galimybę kaip pagrindo sluoksnį naudoti „Google“, „OpenStreetMap“, „Bing“ žemėlapius [15].

„LAStools“ – tai programų rinkinys, kurios pasižymi dideliu efektyvumu ir lankstumu dėl savo galimybių dirbti su komandine eilute. Šiuo rinkiniu galima klasifikuoti, konvertuoti, filtruoti, atlikti trianguliaciją, apibrėžti kontūrus, iškirpti bei atlikti poligonizaciją LiDAR duomenims. Ir tai tėra tik kelios išvardintos funkcijos iš viso rinkinio. Šiuos įrankius taip pat galima naudoti su „ArcGIS“, „QGIS“ [15] ir „ERDAS“ „IMAGINE“ programine įranga. „LAStools“ programinė įranga yra greičiausias ir efektyviausias, kompiuterio atminties atžvilgiu, sprendimas, naudojant komandinę eilutę apdorojant LiDAR duomenis. Ši sistema gali nesunkiai apdoroti milijardus taškų siekiančią informaciją [11].

Pavadinimas	Kūrėjas	Komercinis / nemokamas	Nuoroda
Points2Grid	GEON	Nemokamas	http://lidar.asu.edu/points2grid.html
LViz	GEON	Nemokamas	http://lidar.asu.edu/LViz.html
LidarViewer	KeckCAVES	Nemokamas	http://keckcaves.org/software/lidarviewer
GRASS GIS	OSGeo	Nemokamas	https://grass.osgeo.org
QGIS	OSGeo	Nemokamas	https://qgis.org
ArcGIS	ESRI	Komercinis	https://www.arcgis.com
Polyworks	InnovMetric	Komercinis	http://www.innovmetric.com
TerraScan	TerraSolid	Komercinis	http://www.terrasolid.fi
LAStools	rapidlasso GmbH	Komercinis	https://rapidlasso.com

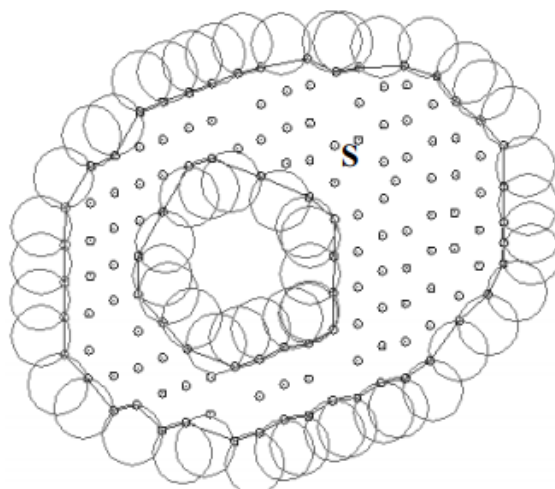
Taip pat reiktų paminėti atviro kodo „PostGIS“ [16] įrankį skirtą atviro kodo „PostgreSQL“ [17] reliacinei duomenų bazei. Šis įrankis suteikia galimybę saugoti geografinius objektus, tokius kaip, daugiakampis, linija, taškas ir pan. duomenų bazėje. Įrankis suteikia daug naudingų funkcijų užduotims su geografiniais objektais atlikti, tokių kaip, atstumo, perimetro, ploto apskaičiavimo. Taip pat yra galimybė atlikti veiksmus su aibėmis, tokius kaip, sąjungos ar sankirtos tarp geografinių objektų. Nesunkiai galima konvertuoti koordinates iš vienos į kitą koordinačių sistemą.

1.4. Vektorizavimo metodų ir algoritmų apžvalga

1.4.1. Pastatų kraštinių gavimas taikant „Alpha Shapes“ algoritmą

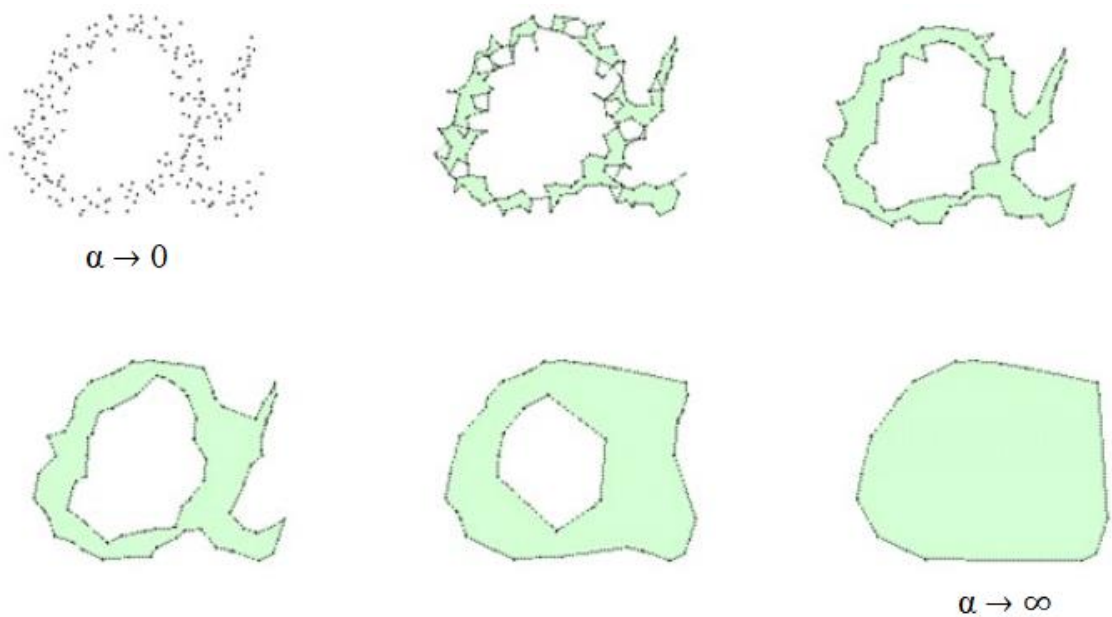
Pastatai yra pagrindiniai miestų elementai. Pastatų kraštinės yra miesto žemėlapių ir 3D pastatų modeliavimo pagrindas. Šiuo metu pastatų kraštinių gavimas naudojant LiDAR duomenis, yra viena iš populiariausių ir kartu sunkiausių temų, apdorojant LiDAR duomenis. Tai lemia atsitiktiniai taškų debesų duomenys ir pastatų sudėtingumas bei įvairovė. Šiame skyriuje aptarsime „Alpha Shapes“ algoritmą, kuris buvo pritaikytas išgauti pastatų kraštines iš taškų debesies.

Taškų debesis, kurio visi taškai priklauso vienam pastatui gali būti vaizduojama kaip 3D arba 2D taškų aibė S . Panaudojant „Alpha Shapes“ algoritmą galima rasti šios aibės kontūrą. Taigi „Alpha Shapes“ algoritmas gali būti naudojamas pastatų kraštinėms gauti (6 pav.).



6 pav. „Alpha Shapes“ algoritmo principas [18]

Tarkime, jog apskritimas su spinduliu α rieda aplink aibę S . Kai α reikšmė yra didesne nei slenkstis (riba), apskritimas nepakliūva į aibės S plotą. Šio apskritimo riedėjimo kelias sudarys taškų aibės S kontūrą. Kai α artėja į begalybę ($\alpha \rightarrow \infty$), tuomet α -figūra tampa iškilioji daugdara. O kai α yra labai mažas ($\alpha \rightarrow 0$), tuomet kiekvienas taškų aibės taškas tampa kraštine (7 pav.). Kai taškų aibę S sudaro tolygiai pasiskirstę taškai ir spindulys α artėja į optimalią reikšmę, „Alpha Shapes“ algoritmas gali nustatyti išorines ir vidines figūros kraštines [18].

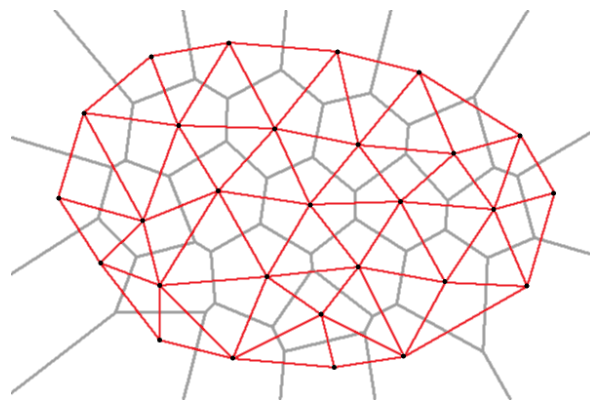


7 pav. Figūros detalumo lygis kintant α reikšmei

1.4.2. Pastatų kraštinių gavimas taikant trianguliaciją / Voronojaus diagramas

Tse [19] konferencijoje pristatė trianguliacija arba dar kitaip vadinamomis Varanojaus diagramomis grįstą algoritmą pastato kraštiniams išgauti.

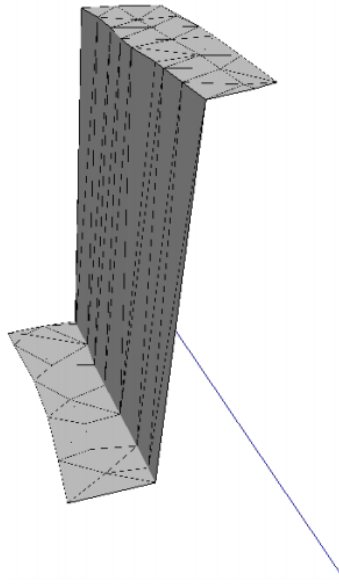
Šio algoritmo idėja yra konvertuoti LiDAR duomenis į Voronojaus diagramas (8 pav.), kuriuose kiekviena celė atitinka daug LiDAR duomenų taškų. Vienos celės atitinka žemės paviršiaus taškus, kitos – pastatų taškus.



8 pav. LiDAR duomenys išreikšti Varanojaus diagrama

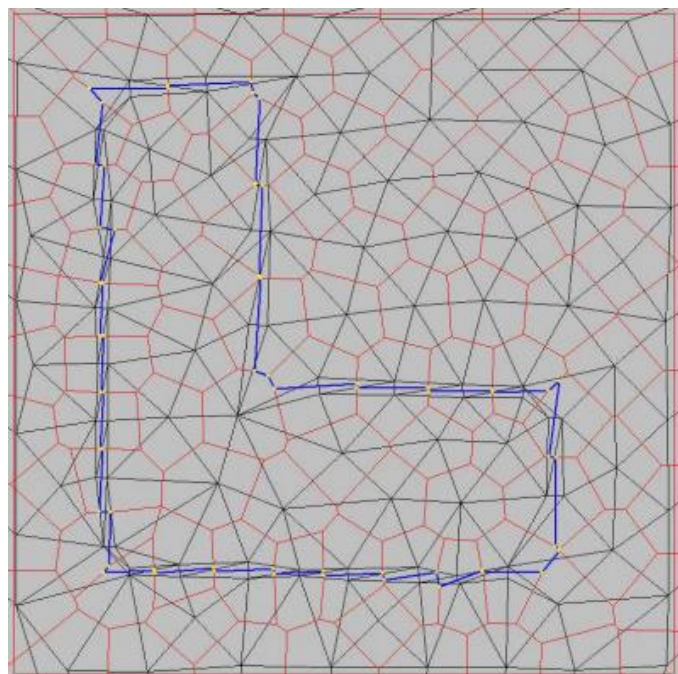
Panaudojant pagrindinių komponentų analizę yra randamos pastato celės. Iš atrinktų celių ištraukiami taškai ir apskaičiuojamos tikrinės reikšmės bei tikriniai vektoriai. Tikriniai vektoriai su

mažiausiomis tikrinėmis reikšmėmis suteikia matomą liniją bei orientaciją atskiriant aukštą plokštumą nuo žemos (9 pav.).



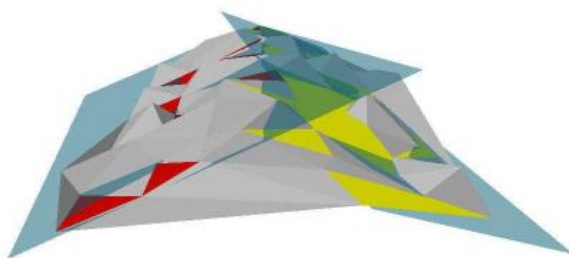
9 pav. Tikriniai vektoriai su mažiausiomis tikrinėmis reikšmėmis [19]

Remiantis tikrinių vektorių orientacija, galima rasti geriausią vietą padalinti ląstelę su maksimaliu skirtumu tarp aukščiausio ir žemiausio taško. Pridėjus po du taškus kirtimo vietoje ir padalinus celes į dvi dalis, gaunamas pastato kontūras (10 pav.).

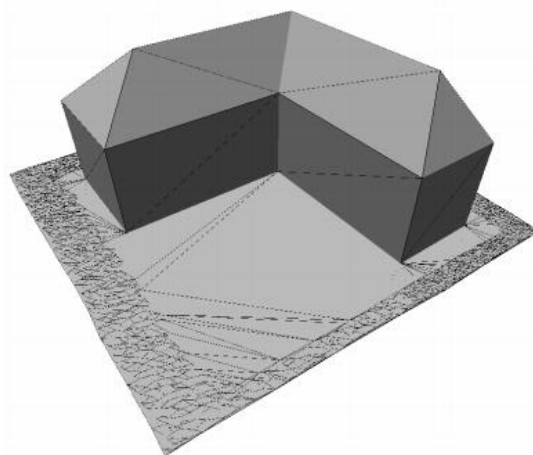


10 pav. Pastato segmentai kiekvienoje Voronojaus celėje [19]

Naudojant Delaunė trianguliaciją (nes, taip pat skiriasi plokštumų aukščiai) galima rasti ir pastato stogo plokštumas (11 pav.) ir taip sukurti pastato 3D modelį (12 pav.).



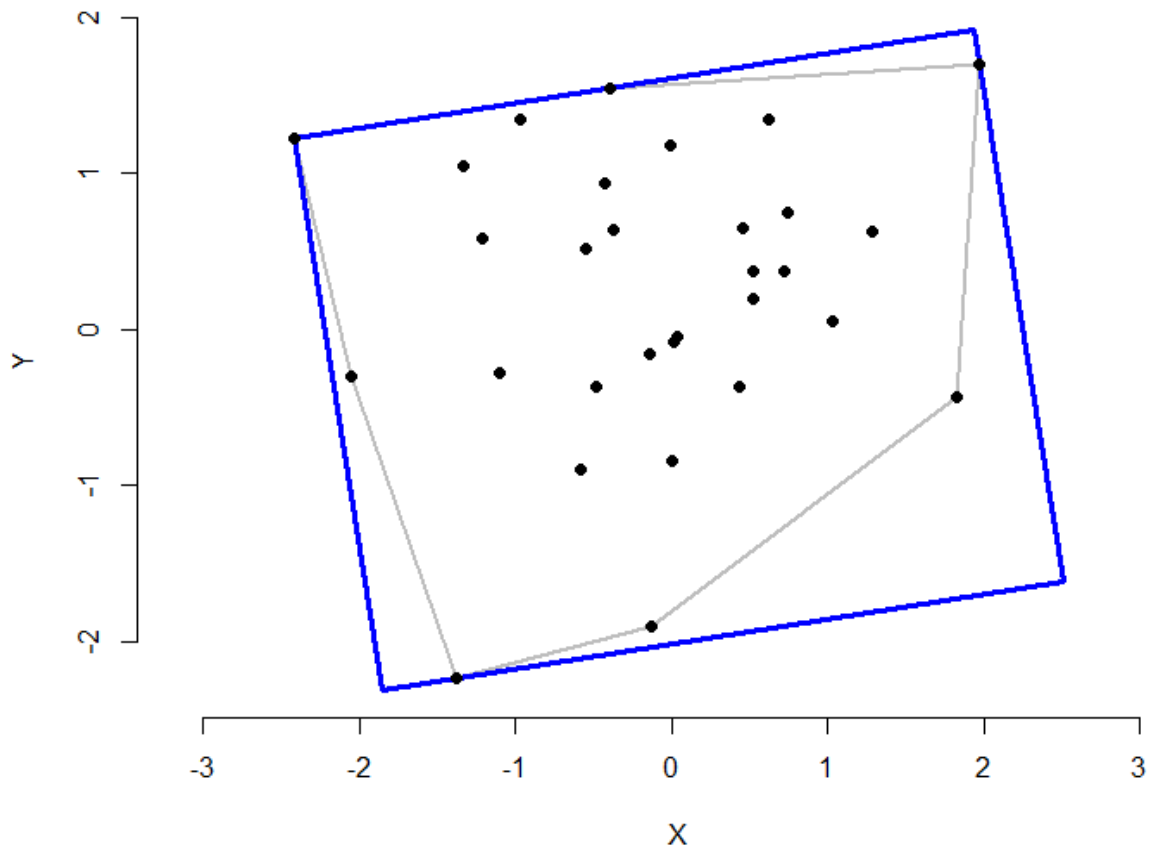
11 pav. Rastos dvi stogo plokštumos [19]



12 pav. 3D pastato modelis pritaikius Voronojaus diagramų metodą [19]

1.4.3. Mažiausio ploto stačiakampio algoritmas

Galime pastebėti, jog problema, duotoje taškų aibėje rasti tuos taškus apimantį stačiakampį, kurio plotas būtų mažiausias (pav. 13), yra aptinkamas įvairiose srityse, tokiose kaip, kompiuterinė geometrija ir pan.



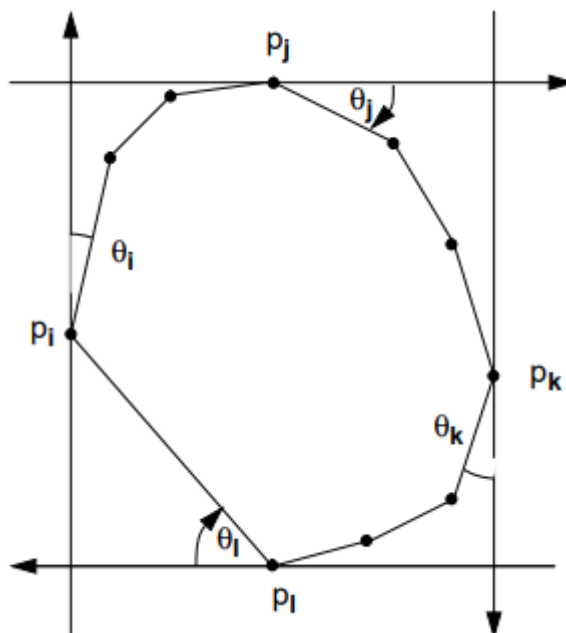
13 pav. Mažiausio ploto stačiakampis duotai taškų aibei

Duotai taškų aibei nesunkiai galime rasti uždarą daugiakampį (iškiliąją daugdarą). Taip pat nesunkiai galime nubrėžti mažiausio ploto stačiakampį, kurio kraštinės būtų lygiagrečios koordinatinių ašims. Tačiau sunkiausia dalis yra rasti ir nustatyti tokio daugiakampio, mažiausią plotą užimančio, stačiakampio orientaciją.

Šiai problemai spręsti Freemanas [20] žurnalo publikacijoje pateikė algoritmą, kurio idėja yra pirmiausia rasti mažiausią plotą užimančią daugiakampį. Tuomet iteratyviai kiekvienai daugiakampio kraštinei nubrėžti stačiakampį taip, jog jo bent viena kraštinė būtų lygiagreti daugiakampiui. Galiausiai pasirenkamas tas stačiakampis, kurio plotas mažiausias.

Taip pat Toussaintas [21] pristatė kitokį sprendimą šiai problemai spręsti. Naudojant šį sprendimą, taip pat, pirmiausia yra randama iškilioji daugdara, duotai taškų aibei. Mokslininkas pasiūlė naudoti du statmenus vienas kitam „slankmačius“, kurie apima daugiakampį (14 pav.). Pirmiausia

surandamos daugiakampio viršūnės su minimaliomis ir maksimaliomis x ir y koordinatėmis. Per šias viršūnes nubrėžiami slankmačiai. Tuomet slankmačiai yra pasukami mažiausiu kraštinės ir slankmačiu sudarančiu kampu. Taip pasuktų slankmačių apribotas plotas tampa mažiausio ploto stačiakampis.



14 pav. Slankmačiais apribotas daugiakampis [21]

1.5. Algoritmų objektų karšinėms normalizuoti apžvalga

1.5.1. Kreivės aproksimacijos „Douglas-Peucker“ algoritmas

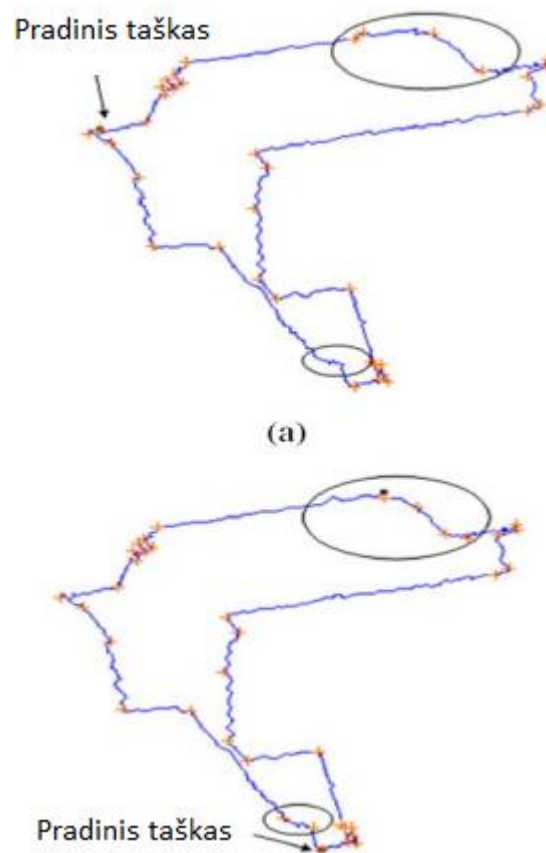
Rameras, Dauglasas ir Peuckeras [22] (RDP) pristatė rekursyvųjį algoritmą, kuris padalina kreivę į kelius segmentus tol, kol maksimalus atstumas iki kitų taškų yra mažesnis nei nurodytas slenkstis. Tarkime turime kreivę, kuria galime apibrėžti kaip aibę C :

$$C = \{P_i(x_i, y_i) | i = 1, 2, \dots, n\} \quad (4)$$

Čia n – yra taškų sekos numeris ir P_i yra kiekvienas taškas su koordinatėmis (x_i, y_i) . Šis metodas pasirenka tašką, kuris turi didžiausią atstumą iki linijos segmento ir yra tarpinis taškas tarp kreivės ir atstumo, didesnio nei nustatyta tolerancija δ_D . Linijos segmentas yra dalinamas į kelius segmentus tol, kol bus pasiekta konvergencija. Statmenas atstumas – D nuo bet kurio taško $P_k(x_k, y_k)$ iki tiesios linijos jungiančios tašką $P_i(x_i, y_i)$ ir $P_j(x_j, y_j)$ apskaičiuojamas taip:

$$D_k = \frac{(y_k - y_i)(x_j - x_i) - (x_k - x_i)(y_j - y_i)}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \quad (5)$$

Šis metodas yra paprastas ir gali būti naudojamas daugelio kreivių aproksimacijos problemoms spręsti. Taip pat šis algoritmas labai greitas, nes jam nereikia apskaičiuoti atskiro kreivės išlinkimo ir gali būti lygiagrečiai vykdomas su dideliais duomenų kiekiais. Tačiau yra keletas atvejų kai aproksimacijos kokybė netinkama tolimesnei analizei. Taip pat šis algoritmas yra jautrus pradinio ir galutinio taško pasirinkimui. Paveikslėlyje (15 pav.) apskritimais pažymėtose vietose galima pastebėti rezultato priklausomybę nuo pradinio ir galutinio taško pasirinkimo. Kryžiuokais pažymėtos vietos yra algoritmo parinkti figūros kampai.



15 pav. Algoritmo parinkti kampai su skirtingais pradžios taškais [23]

1.5.2. Figūros kampų aptikimo ir lokalizavimo „cornerity“ indekso algoritmas

Kampai yra labai svarbūs elementai norint suvokti pačias figūras. Jie yra naudojami norint aprašyti figūrą (trikampis, kvadratas ir pan.). Figūros kampų radimas taip pat svarbus uždavinys norint rasti normalizuotas pastato kraštines iš LiDAR taškų debesies. Guru [24] konferencijoje pristatė algoritmą

skirtą kampų aptikimui ir lokalizavimui. Algoritme, naujas matas, kampo indeksas (angl. cornerity index) yra gautas iš statistinių ir geometrinių ypatybių, susijusių su vidutinėmis X ir Y kreivės taškų koordinatų reikšmėmis. Kiekvienas kreivės taškas turi savo kreivės simetrinius segmentus $S_k(P_i)$ einančius per kaimyninius taškus, kurių dydis $2k + 1$. Kai k yra kaimyninio regiono dydis.

$$S_k(P_i) = \{P_j | j = i - k, \dots, i, \dots, i + k\} \quad (6)$$

Gaunamas taškas P_{ie} atitinka tašką P_i , kuris apibrėžiamas kaip kreivės segmento $S_k(P_i)$ geometrinis centroidas, kuris apskaičiuojamas taip:

$$P_{ie} = (x_{ie}, y_{ie}) \quad (7)$$

Čia x_{ie} ir y_{ie} yra vidutinės taškų X ir Y koordinatų reikšmės, kurios apskaičiuojamos taip:

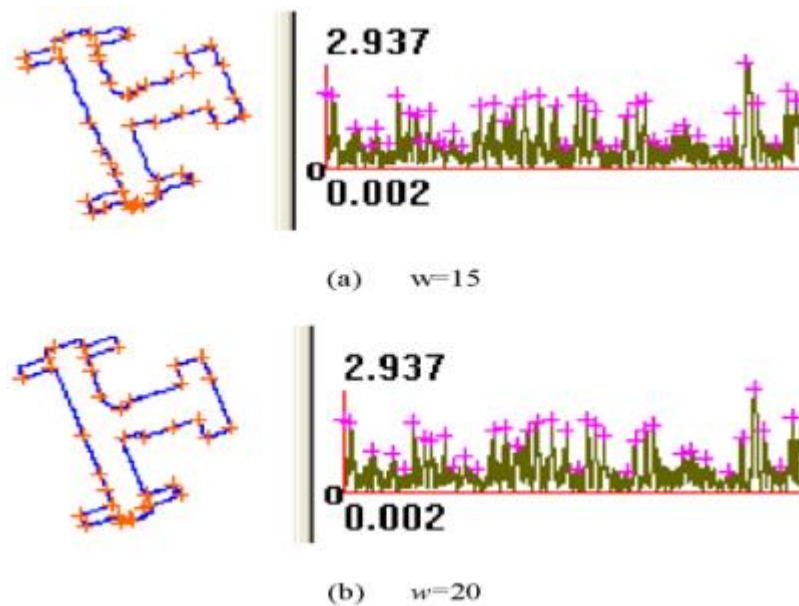
$$x_{ie} = \frac{1}{2k + 1} \sum_{j=i-k}^{j=i+k} x_j, \quad y_{ie} = \frac{1}{2k + 1} \sum_{j=i-k}^{j=i+k} y_j \quad (8)$$

Pakeitę kiekvieną tašką kitu apskaičiuotu pagal algoritmą tašku, gausime išlygintą primityviają kreivę. Apskaičiuotas taškas yra kampas tuomet, kai turi didesnę poslinkį lyginant su kitais kreivės taškais. Poslinkis apskaičiuojamas nustatant kampo indeksą, kuris apskaičiuojamas naudojant Euklido atstumą d tarp taško p_i ir apskaičiuoto taško p_{ie} :

$$d = \sqrt{(x_i - x_{ie})^2 + (y_i - y_{ie})^2} \quad (9)$$

Šiame Guru [24] metode tikimasi gauti aibę sujungtų taškų, kurių kampo indeksas būtų žymiai didesnis nei kitų taškų, taip identifikuojant juos kaip kampus.

Tačiau šis metodas turi trūkumų. Autorius nepateikė jokio būdo, kaip naudoti kokį nors slenkstį, atrenkant nustatytus kampų taškus. Be to, reikšmė k turi būti nustatyta iš anksto, todėl ji negali būti adaptyviai parinkta remiantis atstumu tarp kaimynystėje esančių taškų. Taip pat šis algoritmas labai priklausomas nuo parinkto lango dydžio (w). Kaip matote paveikslėlyje (16 pav.), parinkus lango plotį 20, gaunamas mažesnis skaičius klaidingai parinktų kampų.



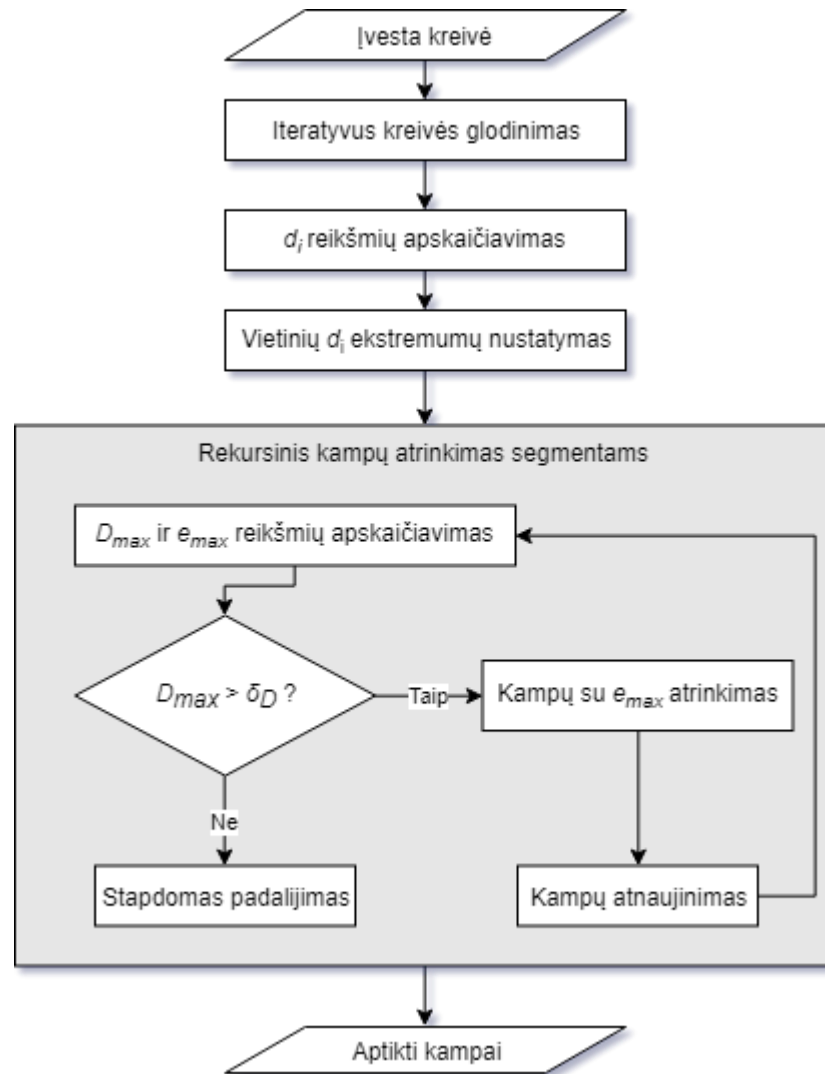
16 pav. Aptikti kampai su skirtingu parinktu lango pločiu [23]

1.5.3. Patobulintas „Douglas-Peucker“ algoritmas pastatų kraštinėms normalizuoti

Li [23] konferencijoje pristatė naują algoritmą figūrų kampams išgauti, kurio pagrindą sudaro dažniausiai pasaulyje naudojamas kreivių supaprastinimo algoritmas „Douglas-Peucker“ [25]. Šis algoritmas papildytas Guru [24] algoritmu kampams aptikti ir lokalizuoti, įvedant kampo indekso reikšmę, pagal kurią iteratyviai atrenkami taškai su didžiausiu lokaliu ekstremumu. Algoritmas ištestuotas įvairiomis figūromis ir palygintas su kitais metodais. Rezultatai parodė, jog šis algoritmas daug mažiau jautrus duomenų triukšmui.

Kampų aptikimo Guru kampo indekso algoritmas užfiksuoja kampų taškų lokalias geometrines savybes, tačiau jų tikslumas yra apribotas iš anksto nustatytais parametrais. „Douglas-Peucker“ algoritmas gerai sprendžia daugiakampio kraštinių aproksimacijos uždavinį, kuris yra labai paprastas, tačiau rezultatai yra labai jautrūs nuo pradinių ir galutinių parinktų taškų. Į šį algoritmą integravus Guru „cornerity“ indekso algoritmą, galima parinkti geriausias pradines reikšmes „Douglas-Peucker“ algoritmui.

Šio metodo veiksmų seka pateikta diagramoje (17 pav.). Pirmiausia iteratyviai glodinama kreivė, slenkančio vidurkio metodu. Tuomet randami kampų indeksai. Nustačius lokalius d_i ekstremumus parenkamos pradinės reikšmės „Douglas-Peucker“ algoritmui, kurios yra du didžiausi d_i ekstremumai. Toliau rekursyviai atliekami tie patys veiksmai su jau suskaidytais segmentais vidinių kampų radimui.



17 pav. Kampų aptikimo veiksmų seka taikant Li patobulintą „Douglas-Peucker“ algoritimą

1.6. Sprendinių optimizavimo ir kokybės bei paklaidos nustatymo metodų ir algoritmų apžvalga

1.6.1. Pareto fronto daugiakriterinis optimizavimo sprendimas

Miettinen [26] savo knygoje rašo jog pagrindinė multikriterinio optimizavimo problema tai, jog ji apima keletą tikslo funkcijų. Matematikoje ši problema gali būti išreiškiama taip:

$$\min(f_1(x), f_2(x), \dots, f_k(x)), \quad x \in X \quad (10)$$

Čia sveikasis skaičius $k \geq 2$ yra tikslo funkcijų skaičius, o aibė X yra optimalių sprendimų vektoriai. Ši aibė dažnai aprašoma kaip tam tikri funkcijos apribojimai. Vektoriais grįstos tikslo funkcijos dar dažnai aprašomos taip:

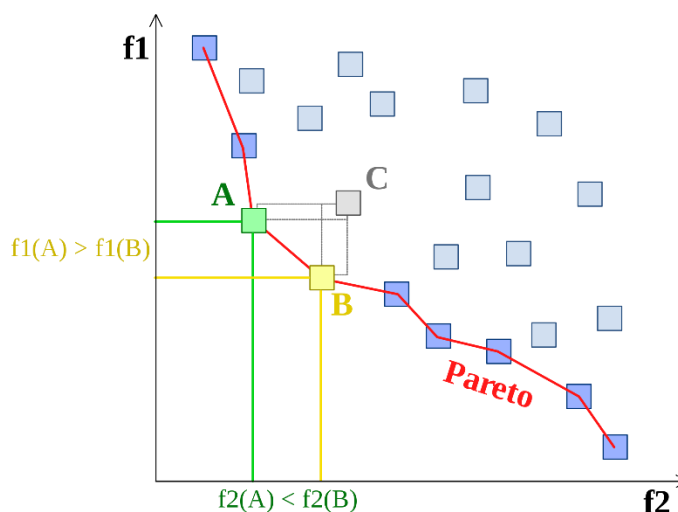
$$f: X \rightarrow R^k, f(x) = (f_1(x), \dots, f_k(x))^T \quad (11)$$

Jei tikslo funkcija maksimizuojama tai minimizuojamos jos neigiamos reikšmės.

Daugiakriteriniame optimizavime nėra tokio optimaliojo sprendinio, kuris minimizuotu visas tikslo funkcijas. Tokiais atvejais taikomas italų ekonomisto Vilfredo Pareto pasiūlytas metodas, Pareto optimaliems sprendiniams rasti (18 pav.). Tokie sprendiniai apibūdinami kaip optimalieji sprendiniai, kurių jau nebegalima pagerinti, nepabloginus kurios nors kitos tikslo funkcijos. Matematikoje išreiškiama taip, optimalusis sprendinys $x^1 \in X$ yra Pareto optimalusis sprendinys tada kai kitas sprendinys $x^2 \in X$, jeigu:

$$1. \quad f_i(x^1) \leq f_i(x^2), \quad \text{čia visi indeksai } i \in \{1, 2, \dots, k\} \text{ ir} \quad (12)$$

$$2. \quad f_j(x^1) < f_j(x^2), \quad \text{bent vienas indeksas } j \in \{1, 2, \dots, k\}. \quad (13)$$



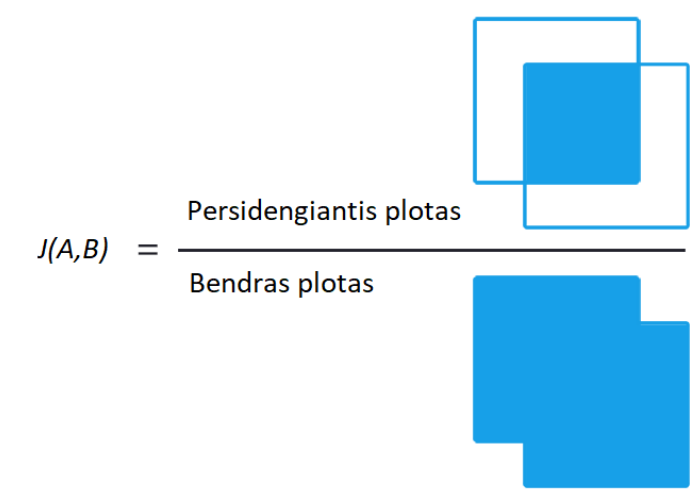
18 pav. Pareto optimalieji sprendiniai

1.6.2. Žakardo indekso metodas aibių panašumui nustatyti

Žakardas [27] buvo statistikas, kuris tirdamas floros pasiskirstymą, aprašė metodą aibių panašumui nustatyti. Šis apibrėžiamas kaip aibių A ir B sankirtos elementų skaičiaus santykis su A ir B sąjungos elementų skaičiumi:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (14)$$

Šis indeksas taip pat gali būti taikomas geometrijoje nustatinėjant dvejų daugiakampių panašumą pagal jų ploto sąjungą ir sankirtą (19 pav.).



19 pav. Žakardo indeksas daugiakampių panašumui nustatyti

1.7. Literatūros apibendrinimas

Apžvelgus literatūroje rastus, su LiDAR technologiją susijusius šaltinius, galime teigti jog ši technologija turi dideles perspektyvas įvairiuose srityse. Prognozuojama jog į šią technologiją bei su ja susijusias sritis bus investuojama vis daugiau, todėl sukurti geri sprendimai gali atnešti didelę ekonominę vertę įmonėms ar valstybei.

Apžvelgus mokslinę literatūrą galime rasti įvairių sprendimų skirtingiems LiDAR duomenų apdorojimo etapams. Tačiau sunku rasti straipsnių, kurie apimtų visą darbo procesą, siekiant kartografuoti vietovę naudojant LiDAR duomenis. Taip pat sunku rasti skirtingų metodų palyginimus. Mokslinėje literatūroje beveik nekalbama apie galutinį etapą, apdorojant LiDAR duomenis, kai iš išgautų objekto (pvz. pastato) taškų siekiama gauti objekto kraštines atitinkančius vektorius.

Apžvelgus literatūroje rastus algoritmus, siekiama aprašyti galimą procesą kartografuojant vietovę iš LiDAR duomenų. Palyginami skirtingi, literatūroje aprašyti, algoritmai objektų kraštinėms išgauti, kraštinėms vektorizuoti. Siekiant didesnio tikslumo, parenkami kiekviename etape tiksliausią rezultatą apskaičiuojantys algoritmai, pateikiama galimybė šiuos algoritmus kombinuoti.

2. Medžiagos ir tyrimų metodai

Šiame skyriuje aprašomi naudoti tyrimu metodai, pateikiami skaičiavimo modeliai ir metodai, aptariamoms neapibrėžtys. Visi etapai, kurie atliekami siekiant galutinio tikslo – kartografuoti vietovę naudojant LiDAR duomenis, pateikti diagramoje (20 pav.).

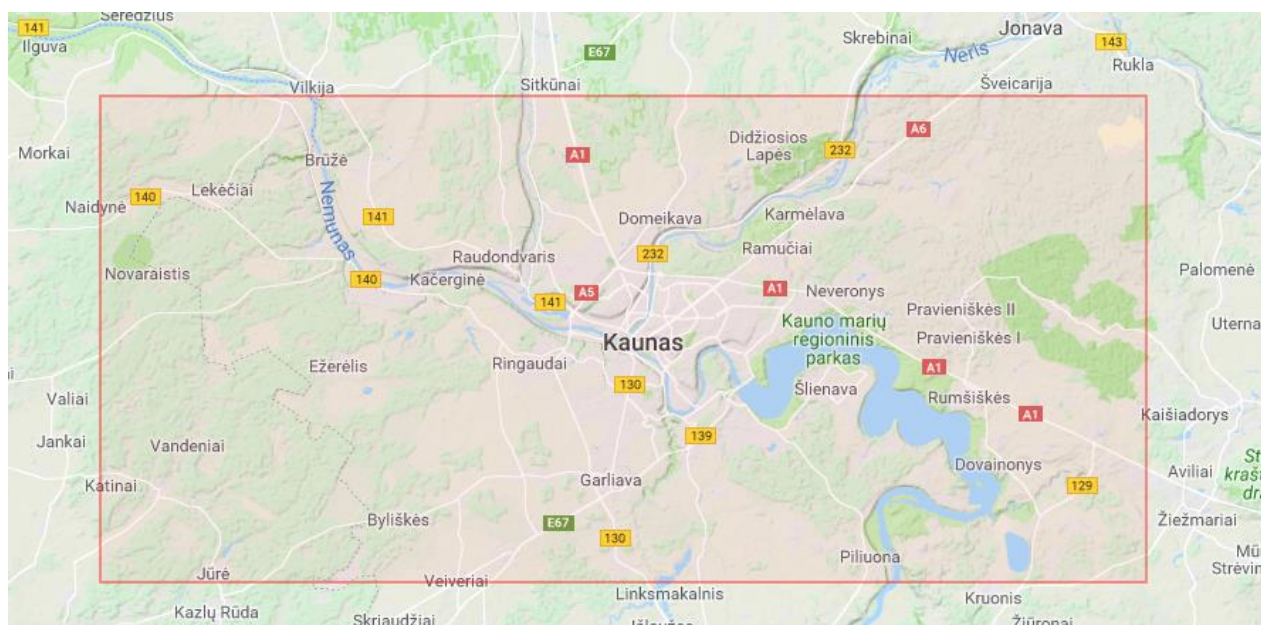


20 pav. Tyrimų etapai siekiant kartografuoti vietovę naudojant LiDAR duomenis

- Duomenų paruošimas – paruošiama LiDAR duomenų imtis, įkeliamą „OpenStreetMap“ duomenys į duomenų bazę.
- Klasifikavimas – klasifikuojami lazerinio skenavimo taškai į žemės paviršiaus, pastatų ir medžių klases. Randami optimalieji parametrai teikiantys geriausius rezultatus
- Klasterizavimas – klasterizuojami lazerinio skenavimo taškai, išskiriami kiekvieno objekto (pastato, aukštos augmenijos) taškų poaibiai.
- Vektorizavimas – išgaunamos objektų kraštinės, palyginami skirtingi metodai, randami optimalieji algoritimų parametrai.
- Normalizavimas – normalizuojamos pastatų kraštinės, palyginami skirtingi metodai, randami optimalieji algoritimų parametrai.

2.1. Tyrimui naudoti duomenys

Tyrimui iš nacionalinės žemės tarnybos (NŽT) gauti LiDAR duomenys apima Kauno, Kauno r., Kaišiadorių, Jonavos, Kazlų Rūdos savivaldybes. Iš viso 1950 km² (21 pav.). Duomenys buvo surinkti 2009 – 2010 metais. Nuskenuotą teritoriją yra dalinai užstatyta, pasižymi kontrastingomis geomorfologinėmis ypatybėmis. Duomenis sudaro virš 2,6 mlrd. taškų, kurie vidutiniškai yra pasiskirstę 1,39 taško į kvadratinį metrą. Tačiau taškai skirtinguose regionuose yra pasiskirstę ne vienodai. Kai kuriuose regionuose galima rasti ir tankesnę taškų pasiskirstymą, apie 9,98 taško į kvadratinį metrą. Taško informacija sudaro koordinatės, kurios užrašytos LKS 94 koordinatių sistemoje, bei taško aukštis virš jūros lygio.



21 pav. Tyrimui naudoti NŽT LiDAR duomenys

Siekiant optimizuoti algoritmų greitaveiką naudojant gijas ir taip išlygiagretinant procesus, taip pat ir geriau orientuotis plokštumoje bei optimizuoti duomenų atvaizdavimo greitaveiką, visas plotas buvo suskaldytas į mažesnius 42,25 hektaro plotus. Šiai užduočiai buvo panaudota tam skirta „LASTools“ programų rinkinio „lastile“ [11] programa, kurios pagalba išlygiagretinus procesus į 8 gijas, visas lazerinio skenavimo taškų apimantis plotas buvo suskaldytas į mažesnius 4747 plotus apytiksliai per 1 h.

2.2. Duomenų paruošimas

Norint nustatyti, kaip gerai buvo suklasifikuoti lazerinio skenavimo taškai ar kaip gerai buvo vektorizuoti objektai, reikia palyginti rezultatus su realiais duomenimis. Šiam tikslui buvo panaudoti atviros informacinės sistemos „OpenStreetMap“ internetinio žemėlapiu duomenys. Panaudojus programą „Osm2pgsql“ [28], žemėlapiu duomenys importuojami į atvirojo kodo, reliacinių duomenų bazių sistemą „PostgreSQL“ [17]. Norint atlikti įvairius veiksmus su žemėlapiu duomenimis, buvo panaudota atvirojo kodo, geografiniams objektams bei operacijomis su jais atlikti, plėtinys „PostGIS“ [16], kuris skirtas „PostgreSQL“ [17] duomenų bazei. Rezultatų analizavimui, interpretavimui ir vizualiam atvaizdavimui buvo naudota atvirojo kodo, geografinė informacinė sistema „QGIS“ [15].

2.3. Lazerinio skenavimo taškų klasifikavimas

Šiame etape siekiame suklasifikuoti duomenis kuo tiksliau į 4 grupes (4 lentelė.), pagal Amerikos fotogrametrijos ir nuotolinio stebėjimo draugijos standartą (ASPRS) [29].

4 lentelė. Suklasifikuotų grupių duomenų reikšmės pagal ASPRS standartą

Klasifikavimo reikšmė	Prasmė
1	Nepriskirtas taškas
2	Žemės paviršiaus taškas
...	...
5	Aukštos augmenijos (pvz. medžiai) taškas
6	Pastato taškas
...	...

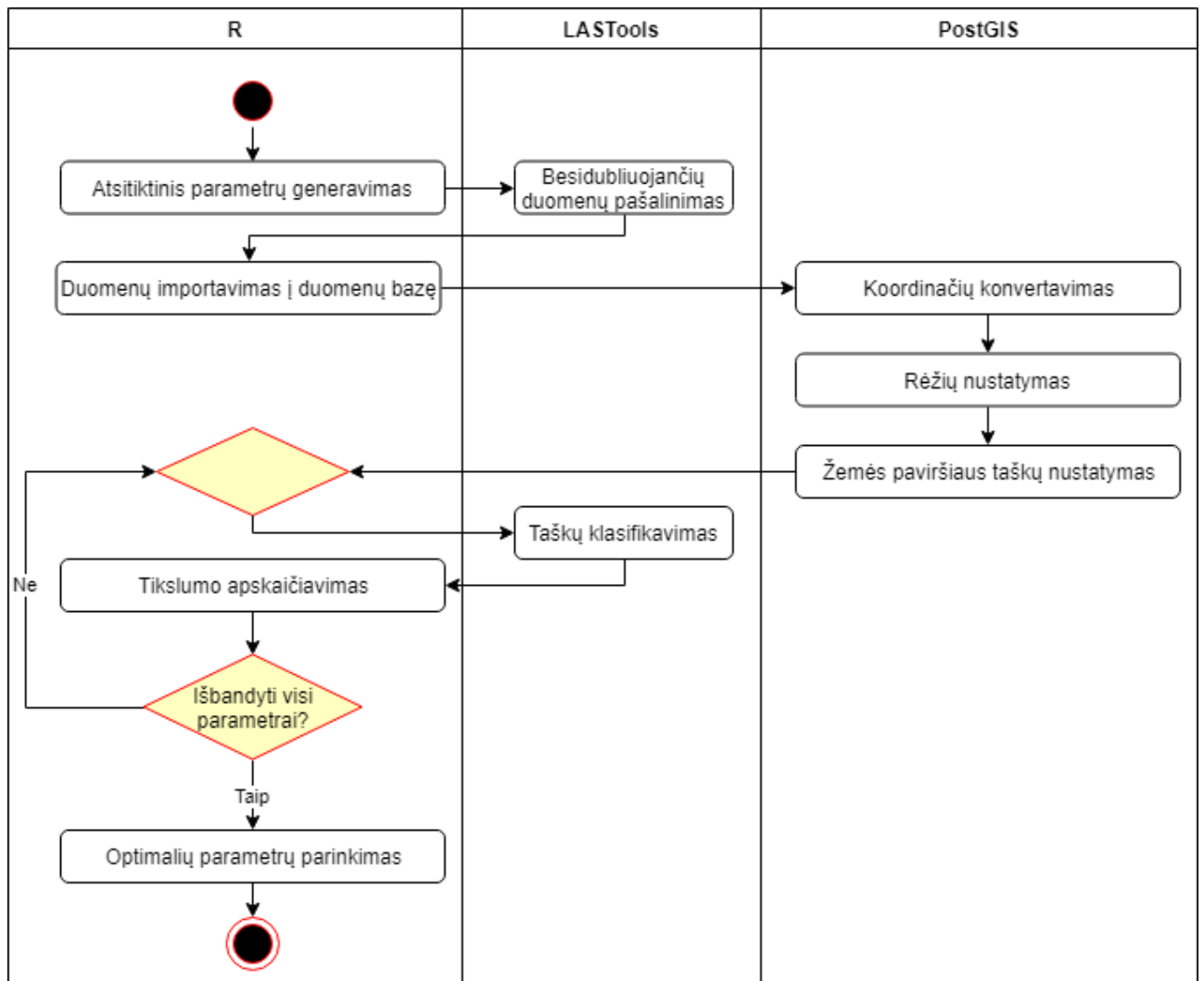
Klasifikavimo procesas susideda iš šių etapų:

- Žemės paviršiaus lygio nustatymas – duomenyse nustatomi lazerinio skenavimo taškai, kurie atitinka žemės paviršių.
- Aukščio nustatymas taškams, kurie nepriskirti jokiai klasei – pašalinamas triukšmas bei kiekvienam taškui apskaičiuojamas atstumas iki iš žemės paviršiaus taškų sukurtos trianguliacijos.
- Lazerinio skenavimo taškų klasifikavimas į pastatų arba aukštos augmenijos klases.

2.3.1. Žemės paviršiaus lygio nustatymas

Pasinaudoję „LASTools“ programų rinkinio „lasground_new“ [11] programa, pirmiausia nustatome žemės paviršiaus lygį. Ši programa klasifikuoja lazerinio skenavimo taškus į dvi grupes – žemės paviršiaus taškus ir nepriskirtus taškus. Programa tiria pasirinktą plotą pažingsniui (numatytoji reikšmė 25 m²) ir bando jame aptikti staigius aukščio šuolius (numatytoji reikšmė 50 cm) taip identifikuojant jog tai nėra žemės paviršiaus taškai. Taip pat yra derinamas maksimalus poslinkis metrais, iki kurio skaičiuojami taškai virš einamojo žemės lygio yra įtraukiami į žemės paviršių (numatytoji reikšmė 10 cm). Be to yra galimybė išjungti ar įjungti tam tikrus optimizavimo požymius, kas taip pat gali padaryti nemažą įtaką rezultatams.

Parametrų derinimo procesas, žemės paviršiaus taškams nustatyti, susideda iš šių etapų (22 pav.):



22 pav. Parametrų derinimo procesas, žemės paviršiaus taškams nustatyti

- Atsitiktinis parametrų generavimas – sukuriamas sąrašas unikalių parametrų, kurie vėliau naudojami klasifikavimui atlikti taikant „LASTools“ programų rinkinio „lasground_new“ [11] programą.
 - Besidubliuojančių duomenų pašalinimas – pašalinami besidubliuojantys x, y, z ašių atžvilgiu, lazerinio skenavimo taškai, naudojant „LASTools“ programų rinkinio „lasduplicate“ [11] programą.
- Duomenų importavimas į duomenų bazę – LiDAR paviršiaus duomenys konvertuojami panaudojant „LASTools“ programų rinkinio „las2txt“ [11] programą, kuri duomenis konvertuoja į kableliais atskirtomis reikšmėmis (CSV pagal angl. comma - separated values) teksto formatą. Tuomet importuojamas CSV formato duomenų failas į „PostgreSQL“ duomenų bazę, panaudojant „R“ paketą „RPostgreSQL“ [30], kuris skirtas „R“ sąsajai su „PostgreSQL“ duomenų baze.
- Koordinačių konvertavimas – siekiant suvienodinti LiDAR duomenų ir „OpenStreetMaps“ sistemos duomenų koordinatinių sistemų, panaudojant „PostGIS“ [16] plėtinį, konvertuojamos LiDAR duomenų koordinatės iš „LKS 94“ koordinatinių sistemų į „WGS 84 / Pseudo-Mercator“ koordinatinių sistemą.
- Rėžių nustatymas – siekiant didesnės greಿತaveikos, panaudojant „PostGIS“ [16] plėtinį nustatomi koordinatinių rėžiai „OpenStreetMap“ sistemos duomenyse, kuriuose bus atliekami tolimesni veiksmai.
- Žemės paviršiaus taškų nustatymas – surandamos dvi LiDAR duomenų taškų aibės. Vieną taškų aibę sudaro taškai, kurie pakliūva, „OpenStreetMap“ informacinės sistemos žemėlapyje, į pastatų bei aukštos augmenijos plotus, bei kita aibė, kurią sudaro tie taškai, kurie nepakliūva. Taip suskirsčius taškus, daroma prielaida jog turimos tikslios žemės paviršiaus taškų ir pastatų bei aukštos augmenijos aibės.

Toliau iteratyviai atliekami veiksmai:

- Taškų klasifikavimas – panaudojus „LASTools“ programų rinkinio „lasground_new“ [11] programą suklasifikuojami taškai į žemės paviršiaus ir niekam nepriskirtų taškų klases.
- Tikslumo apskaičiavimas – randama LiDAR duomenų taškų aibė, kurioje taškai priskirti žemės paviršiaus klasei, taip pat šie taškai pakliūva į žemės paviršiaus plotą „OpenStreetMap“ duomenyse. Tuomet randama kita aibė, kurią sudaro visi LiDAR duomenų taškai, kurie priskirti žemės paviršiaus klasei ir taškai pakliūvantys į žemės paviršiaus plotą „OpenStreetMap“ duomenyse. Galiausiai apskaičiuojamas sprendinio tikslumas taikant

Žakardo indeksą [27]. Tai yra apskaičiuojant santykį tarp pirmosios ir antrosios aibės pagal formulę (14). Taip yra gaunamas sprendinio tikslumas, kuris parodo kaip tiksliai yra suklasifikuojami žemės paviršiaus taškai.

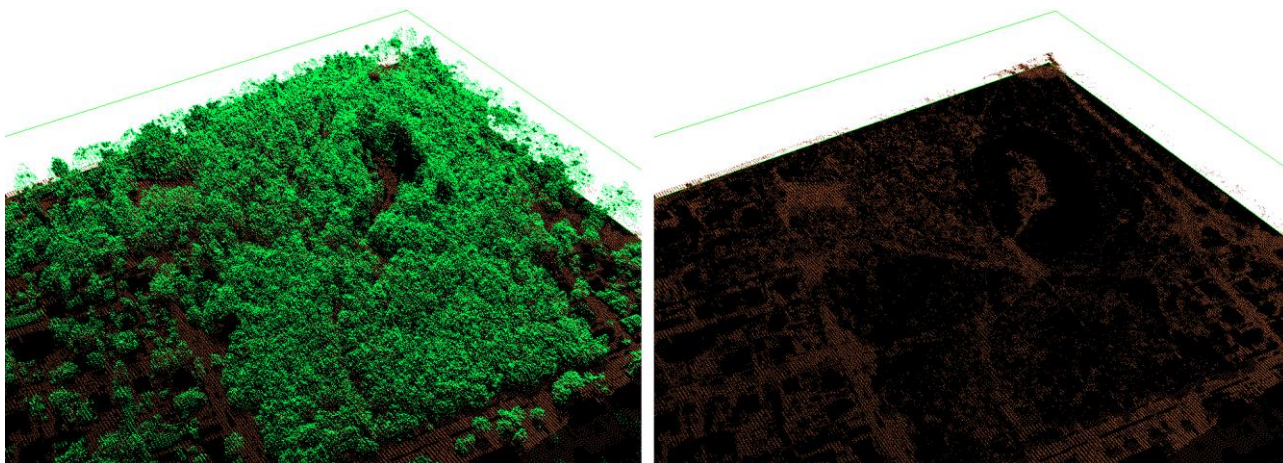
Tačiau mums svarbu ir kaip tiksliai suklasifikuojami, pastatų ir medžių taškai. Kitaip tariant, kaip tiksliai šie taškai yra priskirti nepriskirtųjų taškų klasei. Atitinkamai kaip buvo skaičiuojamas žemės paviršiaus tikslumas lygiai taip pat randamas ir nepriskirtųjų taškų tikslumas. Tik šiuo atveju naudojama LiDAR taškų aibė, kurią sudaro nepriskirti taškai bei aibė, kurioje taškai „OpenStreetMap“ sistemoje pakliūva į pastatų ir medžių plotus.

Galutinis sprendinio tikslumas nustatomas apskaičiavus šių ir prieš tai apskaičiuotų, tikslumo įverčių, aritmetinį vidurkį.

Pasibaigus ciklui atliekami veiksmai:

- Optimalių parametrų parinkimas – pasirenkamas tiksliausias, optimalusis sprendinys, kuris taikomas tolimesniuose skaičiavimuose.

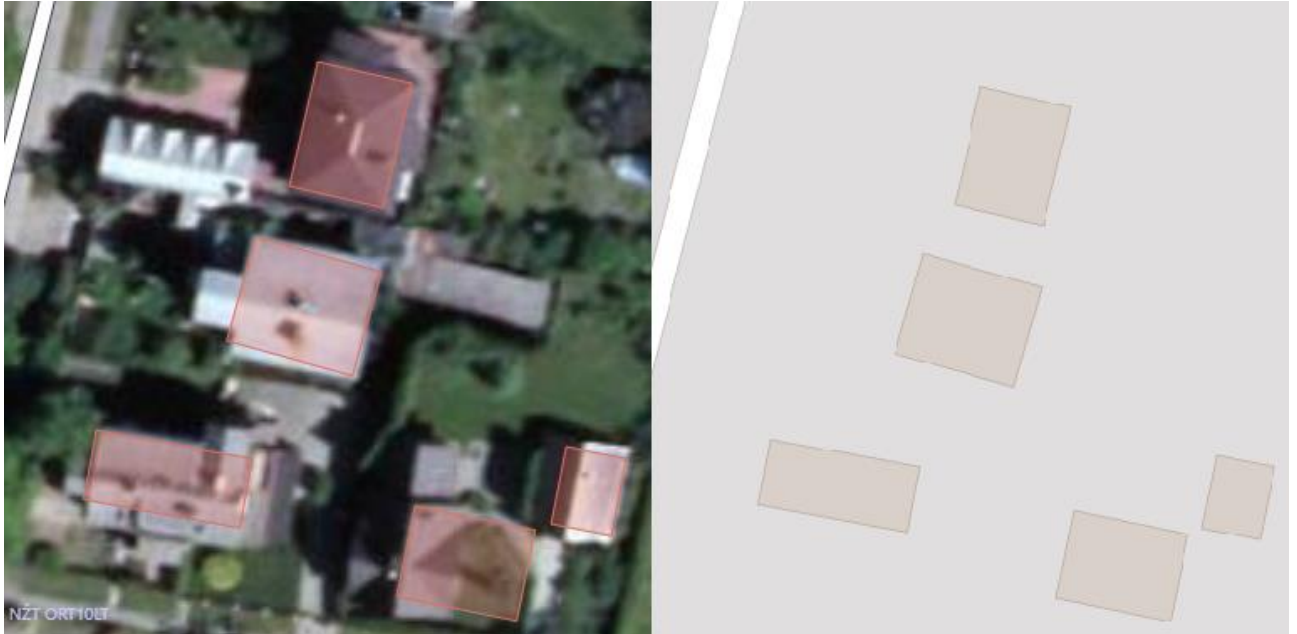
Šiame kokybės vertinime turime keletą neapibrėžtumų. Visų pirma, mes negalime patikrinti kiek neteisingai priskirtų taškų yra medžiais pažymėtuose plotuose, nes LiDAR technologija geba atrasti tarp medžių, šakų ir lapų tarpus, pro kuriuos lazeris pasiekia žemės paviršių. Būtent ši situacija pavaizduota paveikslėlyje (pav. 23), kuriame ta pati vieta pavaizduota atvaizduojant medžiams priskirtus taškus ir jų nerodant. Paveikslėlyje (pav. 23) rudi taškai – atitinka žemės paviršiaus taškus, o žali – aukštos augmenijos taškus.



23 pav. Taškai priskirti žemės paviršiui esantys tarp medžių

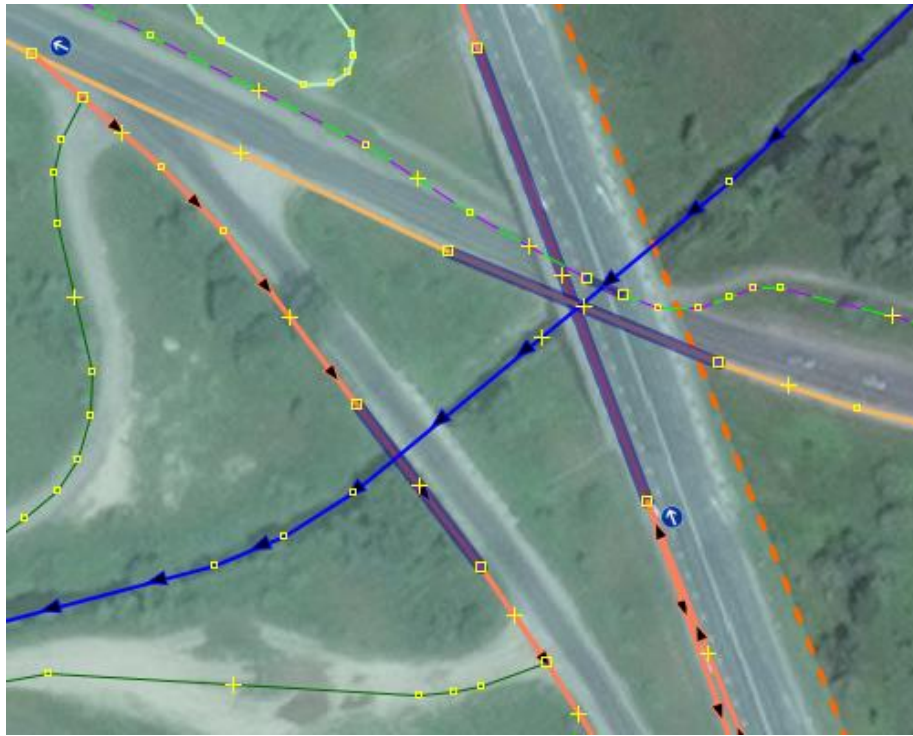
Visų antra, objektai, „OpenStreetMap“ informacinės sistemos žemėlapyje, yra apibrėžiami šios sistemos įvairių vartotojų, naudojant pelyte, braižant ant ortofoto nuotraukų, todėl objektai gali būti

apibrėžti su paklaida arba nepažymėti visai (24 pav.). Visu trečia gauti NŽT gauti LiDAR duomenys buvo surinkti 2009–2010 metais, o lyginami jie su dabartiniais „OpenStreetMap“ informacinės sistemos duomenimis, todėl per 8 metus tiek augmenijos, tiek pastatų situacija gali būti pasikeitusi pvz. iškirsti miškų plotai, pastatyti nauji pastatai.



24 pav. „OpenStreetMap“ informacinės sistemos žemėlapių žymėjimo netikslumai

Visų trečią kartais taip nutinka jog pati ortofoto nuotrauka yra paslinkusi nuo realių pozicijų kažkuria kryptimi (25 pav.). Ypač dažnai, ši situacija, pasitaikydavo iki 2013 metų kai „OpenStreetMap“ informacinėje sistemoje būdavo naudojami „Bing“ ir „Yahoo“ ortofoto nuotraukos. Tačiau nuo 2013 metų galima naudoti NŽT ortofoto nuotraukas, kurios yra itin tiksliai pozicionuotos [31]. Tačiau tam tikra dailis objektų, kurie buvo išbraižyti ant pasislinkusių ortofoto nuotraukų, gali būti nepataisyti į savo tikslias pozicijas, naudojant jau dabartines NŽT ortofoto nuotraukas.



25 pav. Paslinkusi „Bing“ ortofoto nuotrauka [31]

2.3.2. Aukščio nustatymas taškams kurie nepriskirti jokiai klasei

Šiame žingsnyje siekiame paruošti lazerinio skenavimo duomenis, kurie jau yra suklasifikuoti į žemės paviršiaus ir nepriskirtų taškų klases, tolimesniems etapams. Kitame žingsnyje sieksime suklasifikuoti taškus į pastatų ir aukštos augmenijos klases. Šiam tikslui reikia apskaičiuoti aukštį kiekvienam taškui, kuris nepriskirtas jokiai klasei. Tam yra naudojama „LASTools“ programų rinkinio „lasheight“ [11] programa. Ši programa žemės taškam sukuria trianguliacija, o tada nuo jos apskaičiuoja aukštį kiekvienam taškui, kuris nepriskirtas jokiai klasei.

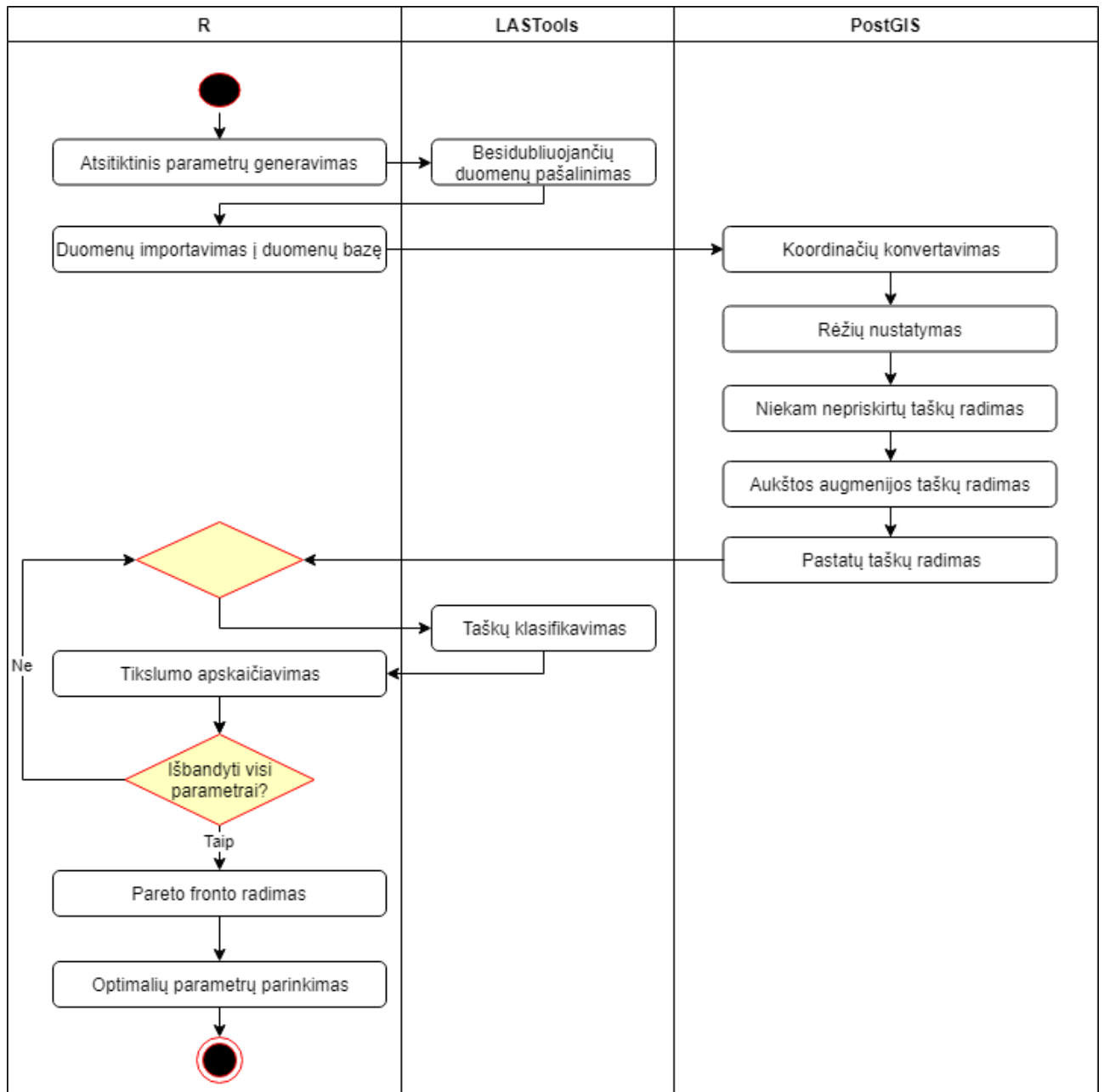
Apskaičiuojant aukštį programa taip pat išvalo duomenis nuo triukšmo – pavienius taškus esančius išskirtinai aukštai arba išskirtinai žemai. Šiuos aukščio nuo žemės paviršiaus duomenis naudos, kitame etape, klasifikavimui naudojama programa.

2.3.3. Taškų klasifikavimas į pastatų arba aukštos augmenijos klases

Kitas žingsnis klasifikuoti pastatus ir aukštą augmeniją (pvz. medžius). Šiam tikslui naudojama „LASTools“ programų rinkinio „lasclassify“ [11] programa, kuri bando pažingsniui aptikti taškus esančius pasirinktame plote (numatytoji reikšmė 2 m^2) ir pasirinktame aukštyje virš žemės paviršiaus (numatytoji reikšmė 2 m). Iš šių taškų bandoma suformuoti pasirinkto ploto (numatyta reikšmė $0,1$

m²) plokštumas (angl. planar), kurios atitinka stogus, arba pagal pasirinktą plotą (numatyta reikšmė 0,4 m²) nelygumus (angl. rugged), kurie atitinka aukštą augmeniją.

Parametrų derinimo procesas (26 pav.) aukštos augmenijos ir pastatų taškams nustatyti iš esmės labai nesikeičia, lyginant su parametrų derinimu, siekiant rasti optimalų žemės paviršiaus klasifikavimo sprendinį.



26 pav. Parametrų derinimo procesas, pastatų ir medžių taškams nustatyti

Skiriasi ieškomos taškų aibės, kurios pakliūva į atitinkamus „OpenStreetMap“ informacinės sistemos žemėlapiu plotus. Šiame parametrų derinimo procese anksčiau neaptarti etapai:

- Niekam nepriskirtų taškų nustatymas – surandama LiDAR duomenų taškų aibė, kurią sudaro taškai, kurie nepakliūva į „OpenStreetMap“ informacinės sistemos žemėlapyje, pastatų bei medžių plotus, taip pat ankstesniame etape nebuvo priskirti žemės paviršiui. Darome prielaidą, jog tokiu būdu, atrinktą poaibį sudaro taškai, kurie yra tikslūs nesuklasifikuotų taškų klasės taškai, kitaip tariant, turėtų likti nesuklasifikuoti po klasifikavimo etapo.
- Aukštos augmenijos taškų radimas – surandama LiDAR duomenų taškų aibė, kurią sudaro taškai, kurie pakliūva į „OpenStreetMap“ informacinės sistemos žemėlapyje, aukštos augmenijos plotus. Darome prielaidą, jog tai yra tikslūs aukštos augmenijos taškai.
- Pastatų taškų radimas – surandamas LiDAR duomenų taškų aibė, kurią sudaro taškai, kurie pakliūva į „OpenStreetMap“ informacinės sistemos žemėlapyje, pastatų plotus. Darome prielaidą, jog tai yra tikslūs pastatų taškai.
- Tikslumo apskaičiavimas – atitinkamai kaip buvo skaičiuojamas žemės paviršiaus tikslumas, taip randami LiDAR duomenų taškų poaibiai (nepriskirtų taškų, aukštos augmenijos, pastatų), ir jų sąjungos bei sankirtos su atitinkamais, anksčiau rastais, „OpenStreetMap“ poaibiais. Kiekvienai klasei apskaičiuojamas poaibių sankirtos ir sąjungos santykis, kuris ir parodo kiekvienos klasės suklasifikavimo tikslumą.
Bendras sprendinio tikslumas nustatomas apskaičiavus šių, prieš tai apskaičiuotų tikslumo įverčių, aritmetinį vidurkį.
- Pareto fronto radimas – kadangi mūsų tikslas yra rasti sprendinį, kurio būtų didžiausias pastatų ir medžių tikslumas, programos sprendiniams taikomas Pareto fronto daugiakriterinis optimizavimo algoritmas. Pareto fronto radimui panaudotas „R“ paketas „rPref“ [32].
- Optimalių parametrų pasirinkimas – pasirenkamas labiausiai tenkinantis Pareto fronto optimalusis sprendinys.

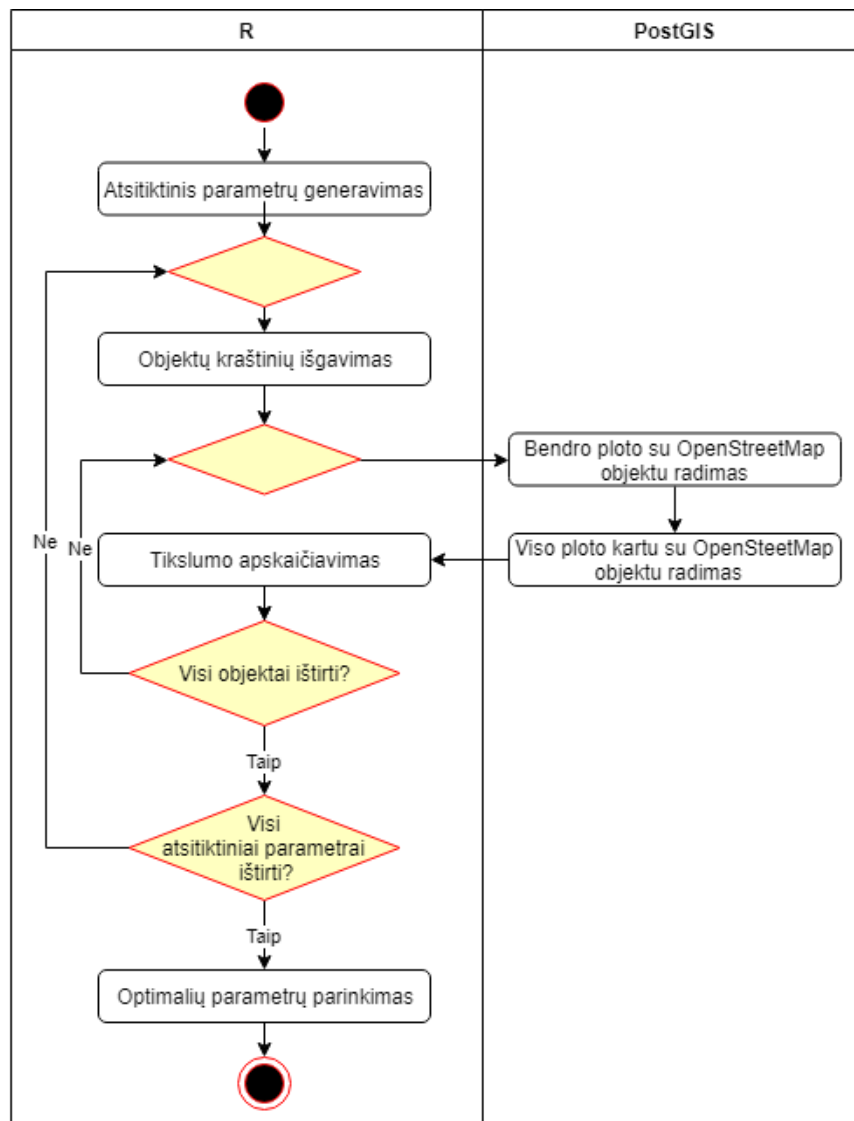
2.4. Objektų kraštinių išgavimas

Suklasifikavus LiDAR duomenis mes turime pavienius taškų debesis, kurie atitinka pastatus ir aukštą augmeniją. Kitas etapas yra juos vektorizuoti. Šiame tiriamajame darbe sieksime vektorizuoti duomenis dvimatėje erdvėje. Taigi mūsų tikslas išgauti pastato ar aukštos augmenijos, taškų užimamo ploto, kraštines.

Šis uždavinys gali būti sprendžiamas skirtingais būdais. Ieškoti objektų kraštinių visiems duomenims iš karto arba bandyti išgauti objekto kraštines kiekvieno atskiro objekto taškų aibei atskirai.

2.4.1. Objektų kraštinių išgavimas taikant „Alpha Shapes“ algoritmą visai imčiai iš karto

Pirmuoju atveju, kai siekiama rasti visas objektų kraštines iš karto, galima naudoti „Alpha Shapes“ algoritmą. Šiam algoritmui realizuoti naudojama „R“ paketo „alphahull“ [33] „ashape“ funkcija. Šioje funkcijoje galima derinti parametą – α , kuris atitinka apskritimo spindulį. Parametrų derinimo procesas „Alpha Shapes“ algoritmui, objektų kraštinių nustatymui, susideda iš šių etapų (27 pav.):



27 pav. „Alpha Shapes“ algoritmo parametrų derinimo procesas pastų kraštinių apibrėžti visai lazerinio skenavimo taškų aibei iš karto

Iteratyviai atliekami veiksmai su kiekvienu atsitiktiniu parametru:

- Objektų kraštinių išgavimas – siekiant išgauti objektų kraštines taikomas „Alpha Shapes“ algoritmas visai taškų aibei. Šią aibę sudaro jau suklasifikuoti lazerinio skenavimo taškai. Algoritmas yra taikomas kiekvienai klasei (aukšta augmenija, pastatai) atskirai.

Iteratyviai atliekami veiksmai su kiekvienu objektu, kuris buvo apibrėžtas „Alpha Shapes“ algoritmu.

- Bendro ploto su „OpenStreetMap“ objektu radimas – randamas „Alpha Shapes“ apibrėžto objekto ploto ir „OpenStreetMap“ žemėlapio objekto bendras plotas. Kitaip tariant šių dviejų objektų sankirta.
- Viso ploto kartu su „OpenStreetMap“ objektu radimas – randamas „Alpha Shapes“ apibrėžto objekto ploto ir „OpenStreetMap“ žemėlapio objekto kartu apimamas visas plotas. Kitaip tariant šių dviejų objektų sąjunga.
- Tikslumo apskaičiavimas - apskaičiuojamas objekto apibrėžimo tikslumas taikant Žakardo indeksą [27]. Kitaip tariant apskaičiuojant santykį tarp sankirtos ir sąjungos, apibrėžto ir „OpenStreetMap“ objektų. Šis tikslumas parodo kaip tiksliai yra apibrėžta objektą atitinkanti taškų aibė.

Pasibaigus ciklui atliekami veiksmai:

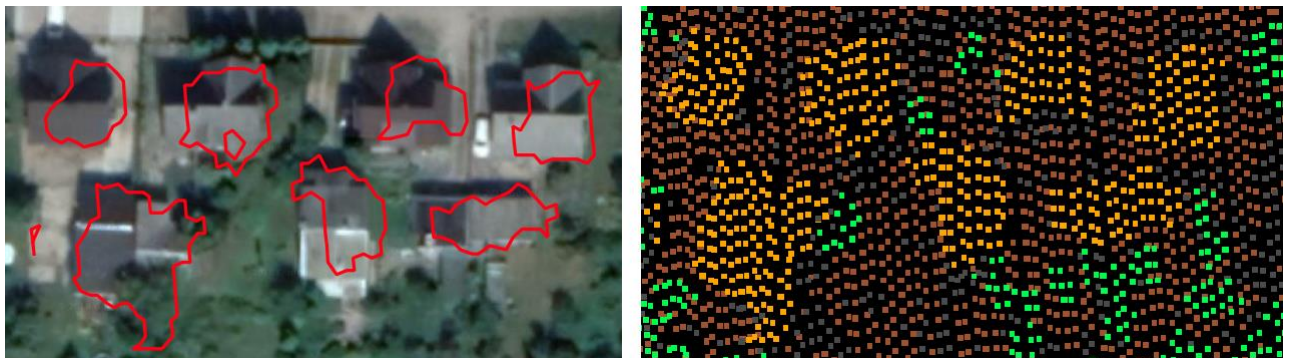
- Optimalių parametrų parinkimas – apskaičiuojamas aritmetinis vidurkis apibrėžtų objektų tikslumo įvertims. Tolimesniems etapams pasirenkamas didžiausią tikslumą turintis sprendinys.

Kai siekiama rasti visas objektų kraštines, „Alpha Shapes“ algoritmas suteikia mums galimybę rasti visų objektų kraštines iš karto, tačiau dėl savo veikimo specifikos jis gali padaryti nemažą paklaidą. Taip nutinka kai keletas objektų yra labai arti vienas kito (atstumas mažesnis nei algoritme naudojamas slenkstis). Tokiu atveju algoritmas apjungia keletą apibrėžti bandomų objektų į vieną bendrą objektą (28 pav.).



28 pav. „Alpha Shapes“ algoritmo klaida esant dideliam slenksčiui

Dėl šios priežasties naudojant mažesnę slenksčių šis algoritmas daro kitą klaidą. Dėl objekto toliau vienas nuo kito esančių taškų šis algoritmas suskaldo apibrėžti bandomą objektą į keletą atskirų objektų. Taip pat pasitaiko atvejų kai randama objekto viduje skylė, kurios neturi būti (29 pav.). Be to, kuo mažesnis slenksčių tuo išgautos kraštinės labiau kintančios, todėl tolimesniuose etapuose normalizuojant kraštines taip pat gaunamos didesnės paklaidos.



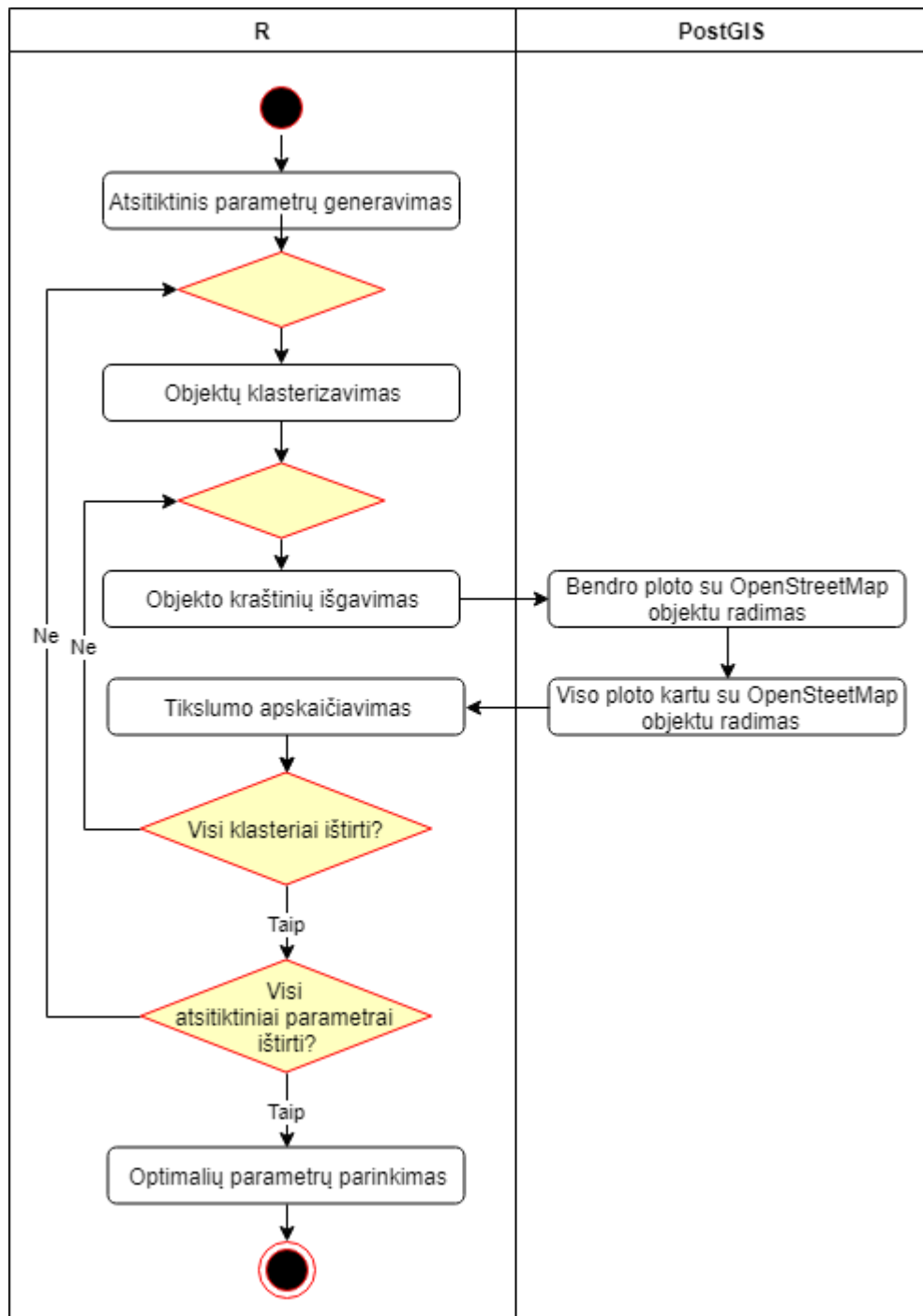
29 pav. „Alpha Shapes“ algoritmo klaida esant mažam slenksčiui

2.4.2. Objektų kraštinių išgavimas taikant „Alpha Shapes“ algoritmą kiekvienam objekto klasteriui atskirai

Antruoju atveju, norint nustatyti kiekvieną taškų aibę, kuri atitiktų objektą atskirai, reikia klasterizuoti duomenis. Kadangi lazerinio skenavimo taškai apytiksliai pasiskirstę panašiu tankumu, mums tiktų tankiu paremti klasterizavimo algoritmai. Vienas dažniausiai naudojamų tokių yra „DBSCAN“ algoritmas. Šio algoritmo realizacijai naudotas „R“ paketas „dbscan“ [34]. Šis algoritmas geba atpažinti įvairių formų klasterius, kuriuos atskiria pagal tarpus tarp jų arba retesniais lazerinio

skenavimo taškų pasiskirstymo regionais. Šis algoritmas ieško tokių klasterių, kurių kiekvieno taško plote pagal pasirinktą spindulį (parametras derinamas) bando aptikti minimalų taškų kiekį (parametras derinamas).

Šio algoritmo parametrų derinimo procesas (30 pav.) objektų kraštinėms išgauti iš esmės labai nesikeičia lyginant su parametrų derinimo procesu siekiant išgauti objektų kraštines iš visos aibės iš karto:



30 pav. „Alpha Shapes“ algoritmo parametru derinimo procesas, objektu kraštinems apibrėžti kiekvienam klasteriui atskirai

Šiame parametru derinimo procese anksčiau neaptarti etapai:

- Objektu klasterizavimas – siekiant išgauti atskirus objektu poaibius taikomas „DBSCAN“ algoritmas visai tašku aibei. Šią aibę sudaro jau suklasifikuoti lazerinio skenavimo taškai. Algoritmas yra taikomas kiekvienai klasei (aukšta augmenija, pastatai) atskirai.

- Objektų kraštinių išgavimas – siekiant išgauti klasterio, kuris atitinka objektą, kraštines taikomas „Alpha Shapes“ algoritmas.

2.4.3. Objektų kraštinių išgavimas taikant mažiausio ploto stačiakampio algoritmą

Turint suklastertuotus objektus atskirai galime taikyti ir kitą gerai žinomą algoritmą vadinamą mažiausio ploto stačiakampio algoritmu. Deja neretai namai būna kiek sudėtingesnės formos nei stačiakampis, nors rasime ir paprastų stačiakampių. Nustačius jog namas yra stačiakampis, šis algoritmas turėtų būti žymiai tikslesnis, nustatinėjant stačiakampio formos namus, nei kiti, nors tinka tik vienam namo formos tipui.

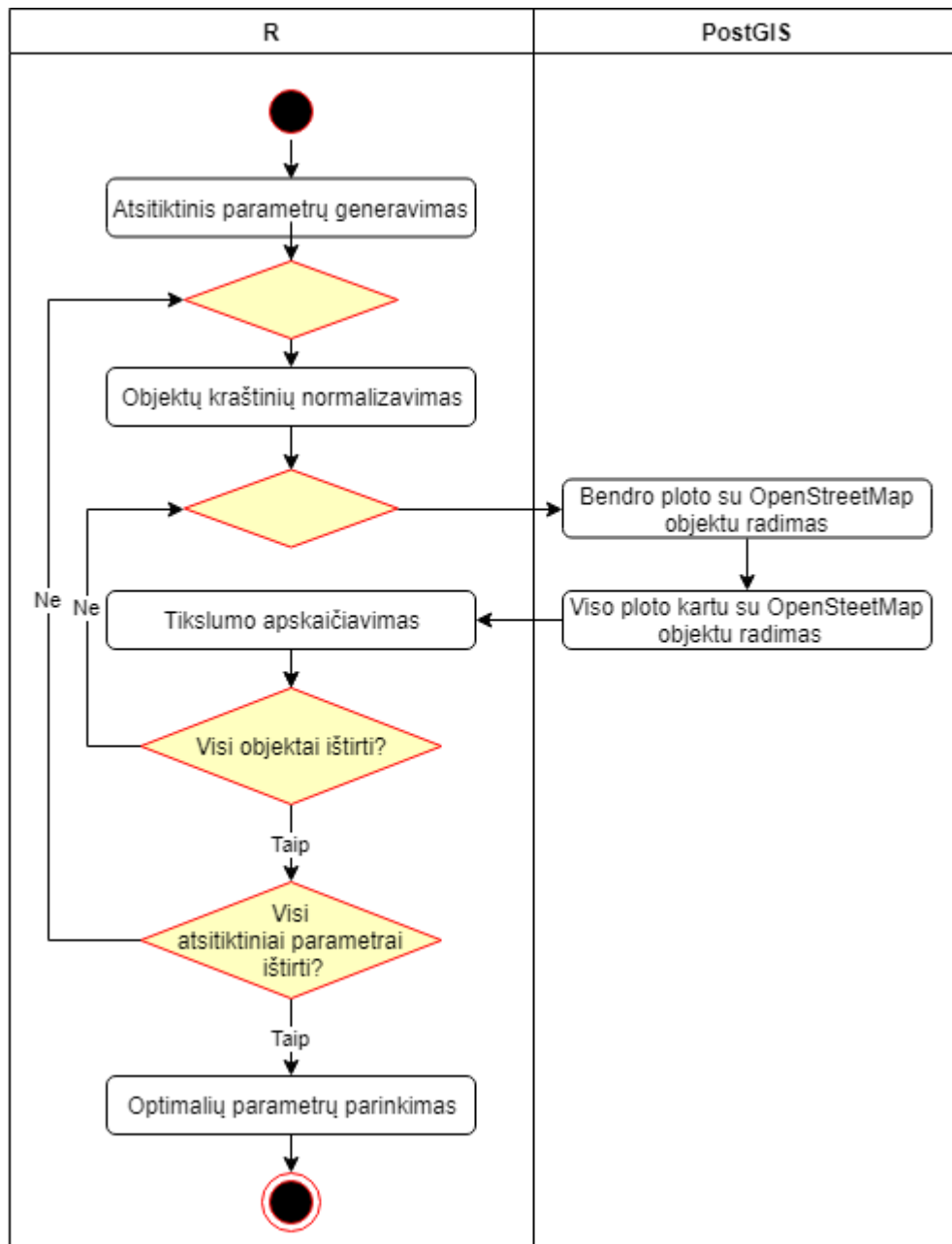
Mažiausio ploto stačiakampio algoritmui derinamų parametrų nėra, tačiau galime suderinti prieš tai taikomo klasterizavimo „DBSCAN“ algoritmo parametrus. Šio algoritmo parametrų derinimo procesas (30 pav.), objektų kraštinėms išgauti, iš esmės labai nesikeičia lyginant su parametrų derinimo procesu, siekiant išgauti objektų kraštines, taikant „Alpha Shapes“ algoritmą kiekvienai objekto taškų aibei atskirai. Esminis skirtumas jog objektų kraštinių išgavimo procese taikomas mažiausio ploto stačiakampio algoritmas.

2.5. Objektų kraštinių normalizavimas

Išgautos objektų kraštinės, taikant „Alpha Shapes“ algoritmą, eina per išorinius, objektą atitinkančios taškų debesies, taškus. LiDAR technologija skenuojant žemės paviršių, lazeris nebūtinai visada pataikys į pastato kraštą. Todėl taikant šį algoritmą mes neturime aiškios, tiesios objekto kraštinės. Dėl šios priežasties taikysime ir palyginsime kelis objektų kraštinėms normalizuoti tinkančius algoritmus.

2.5.1. Objektų kraštinių normalizavimas taikant RDP algoritmą

Vienas dažniausiai taikomų algoritmų kreivėms normalizuoti, yra RDP algoritmas. Šis algoritmas realizuotas „R“ paketo „kmlShape“ [35], „DouglasPeuckerEpsilon“ funkcija. Šiame algoritme yra derinamas parametras – ϵ , kuris atitinka didžiausią leistiną atstumą iki linijos (slenkstį). Šio algoritmo parametrų derinimo procesas (31 pav.), objektų kraštinėms išgauti, iš esmės labai nesikeičia lyginant su parametrų derinimo procesu, siekiant išgauti objektų kraštines, taikant „Alpha Shapes“ algoritmą kiekvienai objekto taškų aibei atskirai. Esminis skirtumas jog vietoje kraštinių išgavimo proceso taikomas objektų kraštinių normalizavimo procesas.



31 pav. RDP algoritmo parametru derinimo procesas

3. TYRIMŲ REZULTATAI IR JŲ APITARIMAS

3.1. Lazerinio skenavimo taškų klasifikavimo rezultatai ir jų aptarimas

3.1.1. Žemės paviršiaus lygio nustatymo rezultatai ir jų aptarimas

Šis etapas labai svarbus norint gauti kuo tikslesnį rezultatą galutiniame etape. Taip yra dėl to, kad neteisingai priskirti taškai žemės paviršiui, tolimesniuose etapuose nebebus bandomi priskirti pastatams ar medžiams. Ir priešingai, nepriskirti taškai nustatinėjant žemės paviršių, bus triukšmas, tarp kurio reikės atpažinti pastatus ir medžius tolimesniuose etapuose. Tai nulemtų tolimesnių etapų prastus skaičiavimų rezultatus.

Nustatyti geriausius žemės taškų klasifikavimo programos parametrus, algoritmas buvo paleistas 540 kartų su atsitiktiniais parametrais. Įvertinus tikslumą, taikant Žakardo indeksą [27] aibių panašumams nustatyti, buvo pasirinkti optimalieji parametrai su kuriais atliktas tiksliausias žemės paviršiaus klasifikavimas. Dešimt optimaliųjų sprendinių, bendro tikslumo mažėjimo tvarka, pateikti lentelėje (5 lentelė.).

5 lentelė. Rasti optimalieji sprendiniai žemės taškų klasifikavimo kokybės parametrams

Žemės paviršiaus tikslumas (%)	Nepriskirtų taškų tikslumas (%)	Bendras tikslumas (%)	Parametrai				
			step	spike down	spike	offset	bulge
24,486	36,523	30,505	12	2	0,5	0,15	1
24,486	36,523	30,504	12	2,5	0,5	0,15	1
24,486	36,523	30,504	12	3	0,5	0,15	1
24,492	36,514	30,503	12	2	0,5	0,15	1,66
24,492	36,514	30,503	12	2,5	0,5	0,15	1,66
24,492	36,514	30,503	12	3	0,5	0,15	1,66
24,488	36,513	30,501	12	2	0,5	0,15	1,33
24,488	36,513	30,501	12	2,5	0,5	0,15	1,33
24,488	36,513	30,501	12	3	0,5	0,15	1,33
24,488	36,505	30,496	12	2,5	0,5	0,15	1,99

Renkantis optimalųjį sprendinį žemės paviršiaus taškų klasifikavimui buvo pasirinktas didžiausią bendrą tikslumą turintis sprendinys. Šis optimalusis sprendinys, kurio didžiausias bendras tikslumas

30,505 %, lentelėje (5 lentelė.) pažymėtas žalia spalva. Šio sprendinio rezultatai bus naudojami kitame, medžių bei pastatų klasifikavimo etape.

3.1.2. Pastatų ir medžių taškų nustatymo rezultatai ir jų aptarimas

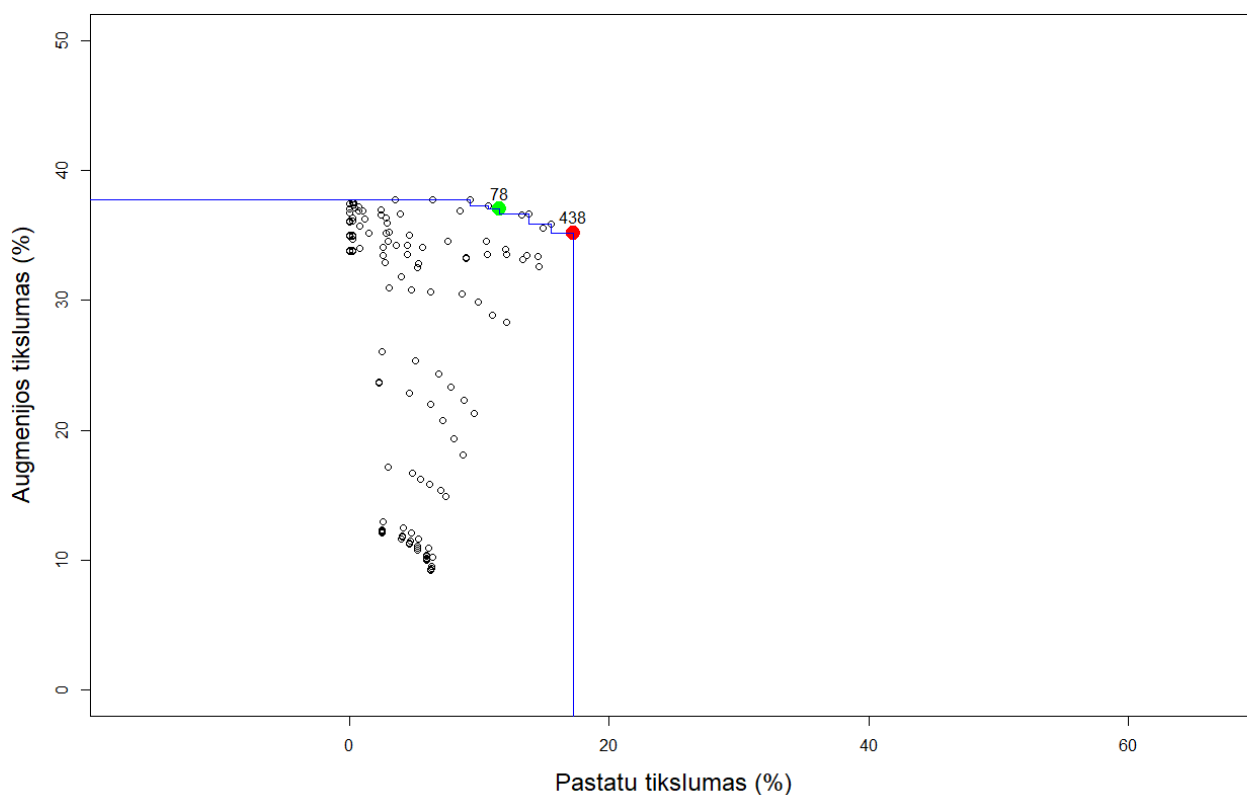
Nustatyti optimalius pastatų ir medžių taškų klasifikavimo programos parametrus, algoritmas buvo paleistas 572 kartus su atsitiktiniais parametrais. Įvertinus bendrą tikslumą, taikant Žakardo indeksą [27] aibių panašumams nustatyti, buvo rasti optimalieji sprendiniai su kuriais suklasifikuoti aukštos augmenijos ir pastatų taškai tiksliausiai. Dešimt optimaliųjų sprendinių bendro tikslumo mažėjimo tvarka pateikti lentelėje (6 lentelė.). Remiantis bendruoju tikslumu rastas optimalusis sprendinys. Šis optimalusis sprendinys, kurio bendras tikslumas 25,365 %, lentelėje (6 lentelė.) ir diagramoje (32 pav.) pažymėtas žalia spalva.

6 lentelė. Rasti Pareto fronto optimalūs sprendiniai pastatų taškų klasifikavimo kokybės parametrams

Pastatų tikslumas (%)	Nepriskirtų taškų tikslumas (%)	Medžių tikslumas (%)	Bendras tikslumas (%)	Parametrai			
				step	planar	ground offset	rugged
11,559	27,511	37,024	25,365	4	0,5	5	0,1
11,559	27,511	37,024	25,365	4	0,5	5	0,3
11,559	27,511	37,024	25,365	4	0,5	5	0,5
11,559	27,509	37,025	25,365	4	0,5	5	0,7
13,851	25,235	36,647	25,245	4	0,5	4	0,1
13,851	25,235	36,647	25,245	4	0,5	4	0,3
13,851	25,235	36,647	25,245	4	0,5	4	0,5
13,851	25,232	36,649	25,244	4	0,5	4	0,7
9,279	27,442	37,736	24,819	4	0,25	5	0,1
...
17,271	15,581	35,190	22,681	4	0,5	2	0,7

Tačiau šiame etape pastatų bei aukštos augmenijos tikslumas mums yra daug aktualesnis nei nepriskirtų taškų tikslumas. Norint rasti optimaliuosius sprendinius, atsižvelgiant tik į pastatų ir

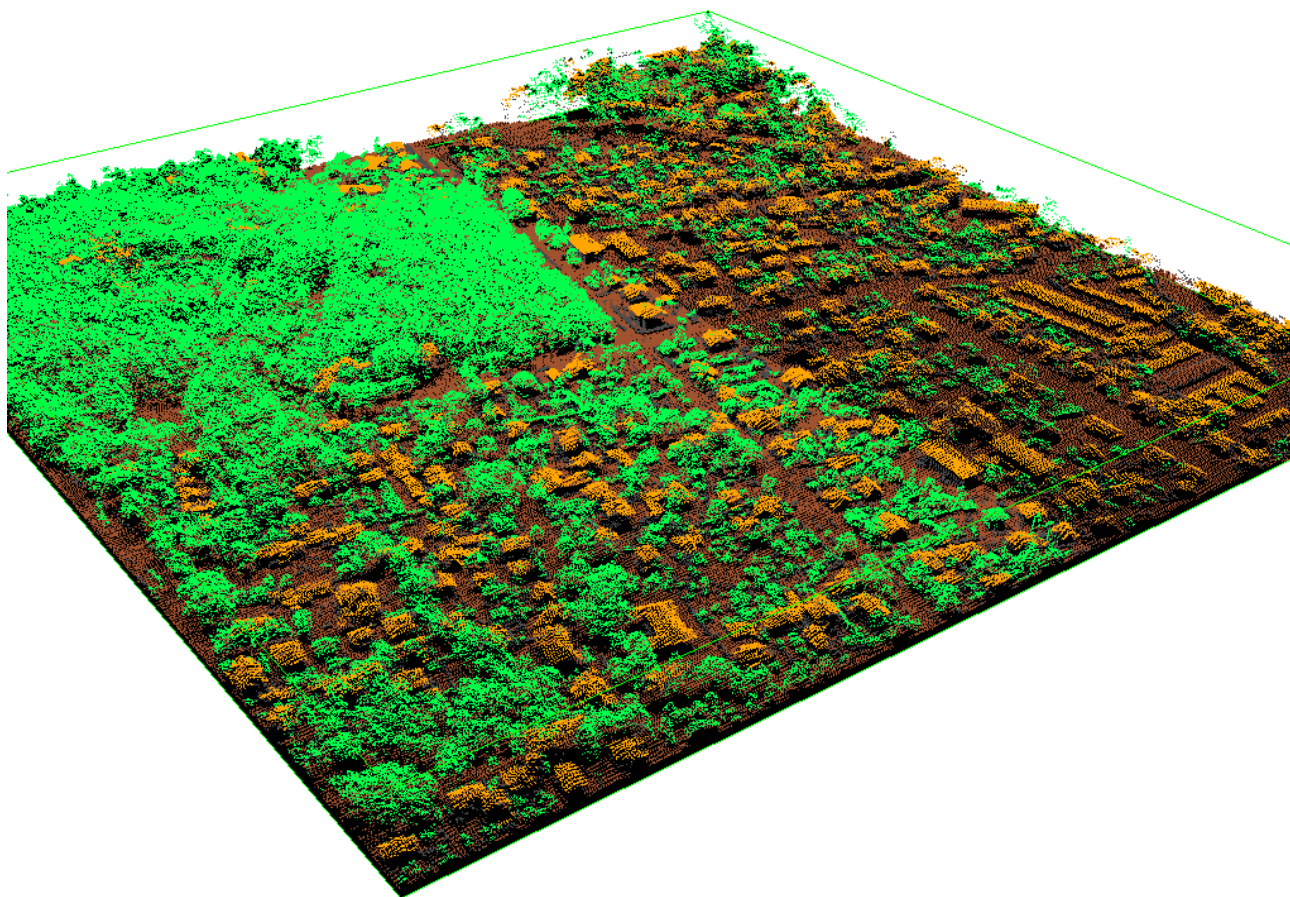
aukštos augmenijos tikslumą, šiems programoms sprendiniams buvo taikomas Pareto fronto daugiakriterinis optimizavimo algoritmas (32 pav.).



32 pav. Pareto frontas, pastatų ir aukštos augmenijos taškų klasifikavimo programos sprendiniams

Šiuo algoritmu buvo rasti 7 optimalieji sprendiniai. Taigi atsižvelgus į tai jog pastatų prastas klasifikavimas daug labiau jaučiamas, žiūrint į rezultatą, galutiniam vartotojui, taip pat diagramoje (32 pav.) galime pastebėti jog pastatų tikslumas didėja labiau nei mažėja aukštos augmenijos tikslumas, pasirinktas Pareto fronto optimalusis sprendinys, kuris kartu yra ir tiksliausias sprendinys pastatų tikslumo atžvilgiu. Šio optimaliojo sprendinio tikslumas pastatų atžvilgiu yra 17,271 %, o tikslumas aukštos augmenijos atžvilgiu 35,19 %, kuris lentelėje (6 lentelė.) ir diagramoje (32 pav.) pažymėtas raudona spalva. Tolimesniuose etapuose bus naudojami šio sprendinio rezultatai.

Paveikslėlyje (33 pav.) pateikti šio sprendinio suklasifikuoti taškai grafiškai. Šiame paveikslėlyje oranžiniai taškai atitinka pastato taškus, žali – aukštos augmenijos taškus, rudi – žemės paviršiaus taškus, o pilki niekam nepriskirtus taškus.



33 pav. Grafiškai atvaizduoti suklasifikuoti LiDAR taškai

Iš pirmo žvilgsnio rezultatas atrodo neblogai, tačiau įdėmiau įsižiūrėjus, galima rasti klasifikavimo klaidų. Pvz. aukštos augmenijos taškai priskirti pastatui ir atvirkščiai. Tai turės įtakos tolimesniems skaičiavimams.

3.2. Objektų kraštinių išgavimas

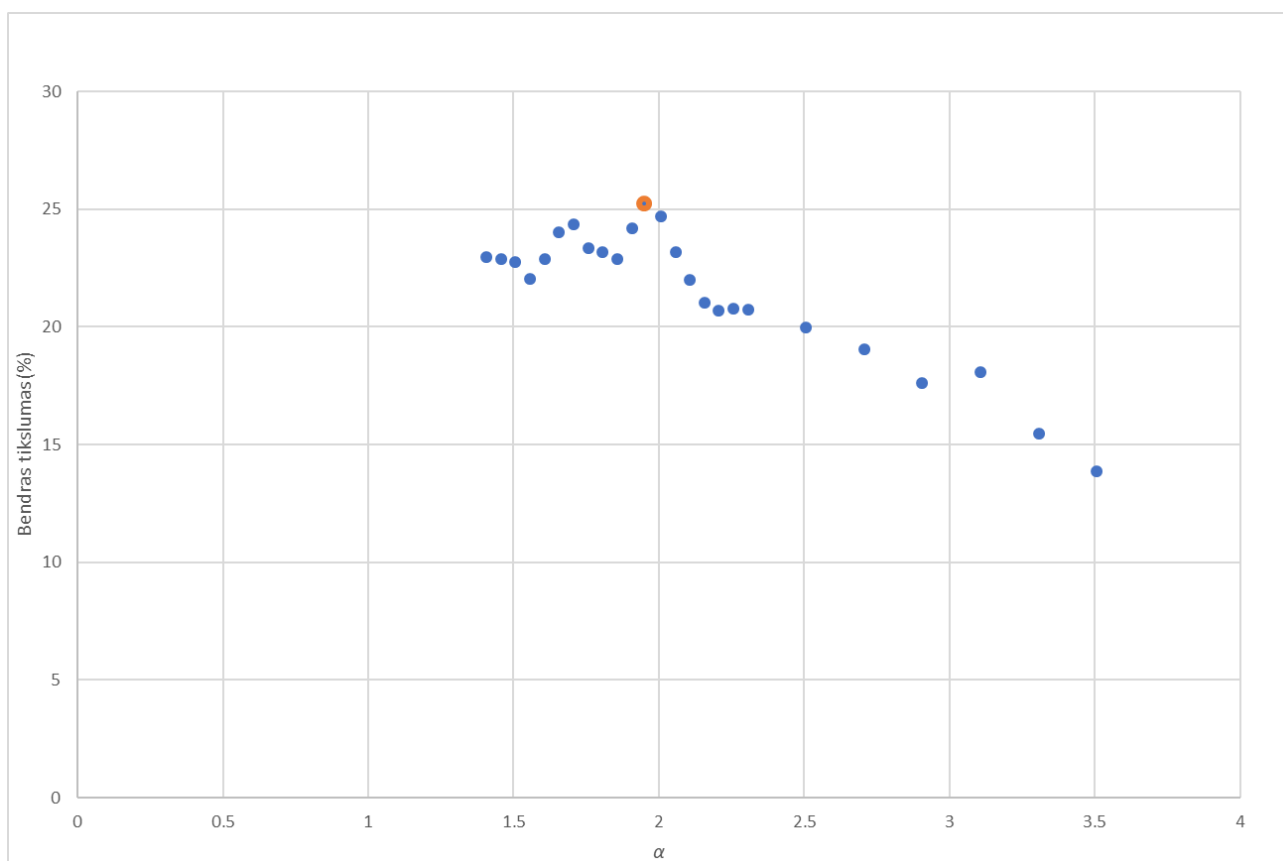
Norint nustatyti geriausią metodą ar algoritmą objektų kraštinių išgavimui, buvo realizuota keletas metodų bei algoritmų. Tiriant kiekvieną metodą ir algoritmą visada buvo naudojama ta pati suklasifikuota imtis ir rezultatai lyginami su tais pačiais „OpenStreetMap“ informacinės sistemos žemėlapiu duomenimis. Todėl darome prielaidą jog galime lyginti šių metodų bei algoritmų rezultatus ir pagal tai spęsti, kuris yra geriausias.

Šiame etape tyrimui naudojamai aibei nebegalime tirti aukštos augmenijos ploto kraštinės išgavimo tikslumo. Taip yra, nes anksčiau naudotas klasifikavimo algoritmas geba atpažinti pavienius medžius ar medžių plotelius, kurių nėra apibrėžta „OpenStreetMap“ žemėlapyje. O dideliuose aukštos augmenijos plotuose, yra randamos medžių proskynos, todėl yra lyginami sąlyginai maži apibrėžti

aukštos augmenijos plotai su dideliu aukštos augmenijos plotu apibrėžtu „OpenStreetMap“ informacinės sistemos žemėlapyje. Todėl toliau tyrėme tik pastatų apibrėžimo tikslumą.

3.2.1. Objektų kraštinių išgavimas taikant „Alpha Shapes“ algoritmą visai imčiai iš karto

Norint nustatyti optimalųjį „Alpha Shapes“ algoritmo slenksčio – α reikšmę, kai algoritmas taikomas visai, suklasifikuotų lazerinio skenavimo taškų, aibei iš karto, algoritmas buvo paleistas 25 kartus su atsitiktine α reikšme. Įvertinus pastatų apibrėžimo tikslumą, taikant Žakardo indeksą [27] aibių panašumams nustatyti, buvo rasti optimalieji sprendiniai su kuriais suklasifikuoti pastatų taškai tiksliausiai. Dešimt optimaliųjų sprendinių pastatų tikslumo mažėjimo tvarka pateikti lentelėje (1 priedas.). Remiantis pastatų apibrėžimo tikslumu optimalusis yra sprendinys, kuris lentelėje (1 priedas.) pažymėtas žalia spalva. Šis optimalusis sprendinys pažymėtas oranžine spalva diagramoje (34 pav.). Šio optimaliojo sprendinio bendras tikslumas yra 25,219 %, kai α reikšmė yra 1,95.



34 pav. Rasti „Alpha Shapes“ optimalieji sprendiniai objektų kraštinių išgavimui

3.2.2. Objektų kraštinių išgavimas taikant „Alpha Shapes“ algoritimą kiekvienai taškų aibei atskirai

Norint nustatyti optimaliuosius „Alpha Shapes“ algoritmo parametrus, kai algoritmas taikomas kiekvienam, objektą atitinkančiam, klasteriui, algoritmas buvo paleistas 25 kartus su atsitiktiniais parametrais. Įvertinus pastatų apibrėžimo tikslumą, taikant Žakardo indeksą [27] aibių panašumams nustatyti, buvo rasti optimalieji sprendiniai su kuriais suklasifikuoti pastatų taškai tiksliausiai. Dešimt optimaliųjų sprendinių pastatų tikslumo mažėjimo tvarka pateikti lentelėje (7 lentelė.). Remiantis pastatų apibrėžimo tikslumu optimalusis yra pirmas sprendinys, kuris lentelėje (7 lentelė.) pažymėtas žalia spalva.

7 lentelė. Rasti „Alpha Shapes“ optimalūs sprendiniai objektų kraštinių išgavimui

Vidutinis kampų skaičius	Pastatų tikslumas (%)	Aukštos augmenijos (%)	Bendras tikslumas (%)	Parametrai		
				alph a	eps	minPt s
49,106	51,258	0,594	25,926	1,9	1,9	4
49,718	50,912	0,786	25,849	1,8	1,8	3
21,913	25,080	0,012	12,546	1,8	1,95	5
21,913	25,080	0,031	12,555	5	1,95	3
21,913	25,080	0,014	12,547	1,9	2,1	4
21,913	25,080	0,055	12,568	2	1,8	5
31,755	22,996	0,012	11,504	1,8	2	5
21,852	21,586	0,012	10,799	1,8	5	5
27,091	12,833	0,012	6,423	1,85	1,8	3
27,091	12,833	0,055	6,444	2	2	5

Kaip ir buvo tikėtasi, objektų apibrėžimas taikant „Alpha Shapes“ algoritimą, prieš tai juos klasterizavus į atskirus taškų klasterius, yra tikslesnis nei taikant algoritimą naudojant visą taškų aibę iš karto. Galime pastebėti jog optimalioji α reikšmė abejais atvejais yra labai panaši. Taigi galime daryti prielaidą jog šio algoritmo taikymo metodas yra tikslesnis dėl atlikto klasterizavimo prieš tai. Turbūt tai lemia tai jog „DBSCAN“ algoritmas išvalo šiek tiek triukšmo, kurį priskiria triukšmo klasteriui.

3.2.3. Objektų kraštinių išgavimas randant įgaubtąją daugdarą kiekvienai taškų aibei atskirai

Norint nustatyti optimaliuosius klasterizavimo parametrus, kai kiekvienam objektą atitinkančiam klasteriui, kraštinės išgaunamos taikant įgaubtosios daugdaros algoritimą, algoritmas buvo paleistas 25 kartus su atsitiktiniais klasterizavimo parametrais. Įvertinus pastatų apibrėžimo tikslumą, taikant Žakardo indeksą [27] aibių panašumams nustatyti, buvo rasti optimalieji sprendiniai su kuriais apibrėžtos pastatų kraštinės tiksliausiai. Dešimt optimaliųjų sprendinių, pastatų tikslumo mažėjimo tvarka, pateikti lentelėje (8 lentelė.). Remiantis pastatų apibrėžimo tikslumu optimalusis yra pirmas sprendinys, kuris lentelėje (8 lentelė.) pažymėtas žalia spalva.

8 lentelė. Rasti optimalūs sprendiniai objektų kraštinių išgavimui taikant įgaubtosios daugdaros algoritimą

Vidutinis kampų skaičius	Pastatų tikslumas (%)	Medžių tikslumas (%)	Bendras tikslumas (%)	Parametrai	
				eps	minPts
49,106	45,371	0,004	22,687	2	4
49,718	43,681	0,007	21,844	2,5	5
21,913	7,815	0,005	3,910	2,2	4
21,913	4,735	0,004	2,369	2,4	4
21,913	4,308	0,006	2,157	2	5
21,913	4,308	0,004	2,156	2,6	4
31,755	4,308	0,004	2,156	2,8	4
21,852	4,308	0,004	2,156	3	4
27,091	0,802	0,013	0,407	2	6
27,091	0,802	0,006	0,404	2,2	5

3.2.4. Objektų kraštinių išgavimas taikant mažiausio ploto stačiakampio algoritimą

Norint nustatyti optimaliuosius klasterizavimo parametrus, kai kiekvienam objektą atitinkančiam klasteriui, kraštinės išgaunamos taikant mažiausio ploto stačiakampio algoritimą, algoritmas buvo paleistas 25 kartus su atsitiktiniais klasterizavimo parametrais. Šio algoritmo netaikome aukštos augmenijos plotų kraštinėms išgauti, nes labai retas atvejis kai juos galime žymėti stačiakampiu. Taip pat, stačiakampiu pažymėti pavieniai medžiai, atrodo labai nepriimtinais galutiniam vartotojui. Įvertinus pastatų apibrėžimo tikslumą, taikant Žakardo indeksą [27] aibių panašumams nustatyti, buvo rasti optimalieji sprendiniai su kuriais apibrėžtos pastatų kraštinės tiksliausiai. Dešimt

optimaliųjų sprendinių, pastatų tikslumo mažėjimo tvarka, pateikti lentelėje (9 lentelė.). Remiantis pastatų apibrėžimo tikslumu optimalusis yra pirmas sprendinys, kuris lentelėje (9 lentelė.) pažymėtas žalia spalva.

9 lentelė. Rasti optimalieji sprendiniai objektų kraštinių išgavimui taikant mažiausio ploto stačiakampio algoritmą

Vidutinis kampų skaičius	Pastatų tikslumas (%)	Parametrai	
		eps	minPts
4	42,105	2,2	7
4	41,684	1,7	8
4	41,502	2,1	6
4	41,228	2,4	8
4	41,069	2,5	8
4	40,892	2,3	8
4	40,839	2	5
4	40,656	2,2	6
4	40,515	1,6	7
4	40,507	2	6

Kaip ir buvo tikėtasi šiuo algoritmu išgautos kraštinės neblogai atvaizduoja stačiakampio formos pastatus, tačiau daug prasčiau sekasi su kitokios formos pastatais, o dar blogiau su didelę plotą užimančiais ir ne stačiakampio formos pastatais (35 pav.). Tačiau šio algoritmo rezultatų jau nebereikia normalizuoti, nes yra išgaunamos tiesios pastatų kraštinės.

Galime pastebėti jog šio algoritmo kokybei didelę įtaką daro klasterizavimo algoritmo „DBSCAN“ pradinės reikšmės („eps“, „minPts“). Kadangi mažiausio ploto stačiakampio algoritmas yra jautrus duomenų triukšmui, jis žymiai tiksliau išgauna pastato kraštines, kai klasterizavimo parametrai yra parinkti taip, jog būtų atfiltruota kuo daugiau duomenų triukšmo.

Šis algoritmas galėtų būti taikomas kombinuotai su kitais algoritmais, kurie geba išgauti ne tik stačiakampio formos pastatų kraštines, jei būtų galimybė nustatyti ar pastatas yra stačiakampio formos.



35 pav. Išgautos pastatų kraštinės mažiausio ploto stačiakampio algoritmu

3.2.5. Objektų kraštinių išgavimo apibendrinimas

Apibendrinus visus kraštines išgauti tirtus algoritmus tiksliausią rezultatą parodė „Alpha Shapes“ algoritmas, kuris buvo taikytas kiekvienam klasterizuotam objektui atskirai. Šio algoritmo tikslumo rezultatai lentelėje (10 lentelė.) pažymėti žalia spalva. Šio algoritmo pastatų apibrėžimo rezultatai bus taikomi tolimesniuose tyrimuose normalizuojant kraštines.

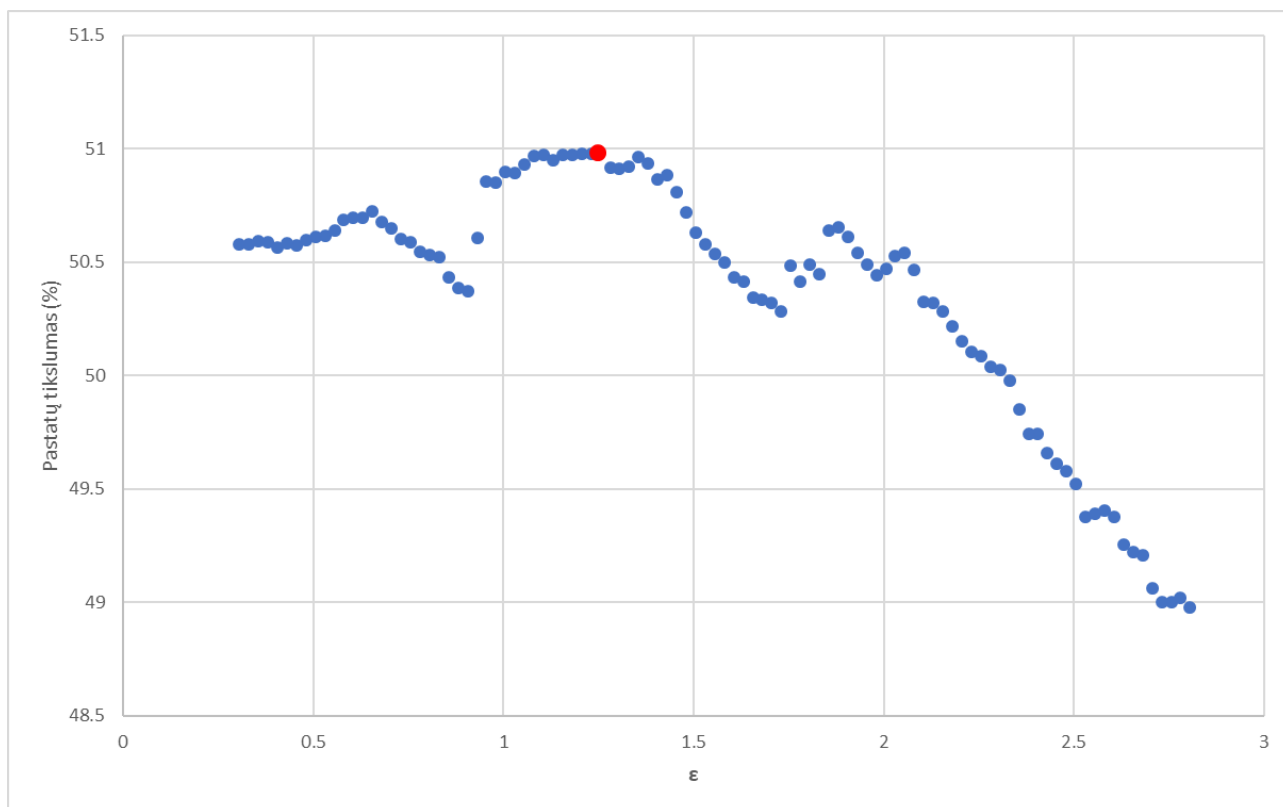
10 lentelė. Algoritmų rezultatų kraštinėms išgauti palyginimas

Algoritmas	Pastatų tikslumas (%)	Medžių tikslumas (%)	Bendras tikslumas (%)
„Alpha Shapes“ visai imčiai iš karto	50,119	0,319	25,219
„Alpha Shapes“ kiekvienam objektui atskirai	51,258	0,594	25,926
Įgaubtosios daugdaros	45,371	0,004	22,687
Mažiausio ploto stačiakampio	42,105	-	-

3.3. Objektų kraštinių normalizavimas

3.3.1. Objektų kraštinių normalizavimas taikant RDP algoritmą

Norint nustatyti optimaliuosius normalizavimo parametrus, kai kiekvienas objektas yra jau apibrėžtas „Alpha Shapes“ algoritmu, kraštinės normalizuojamos taikant RDP algoritmą, algoritmas buvo paleistas 101 kartą su atsitiktiniais parametrais. Įvertinus pastatų apibrėžimo tikslumą, taikant Žakardo indeksą [27] aibių panašumams nustatyti, buvo rasti optimalieji sprendiniai, su kuriais normalizuotos pastatų kraštinės tiksliausiai, kitaip tariant, dėl kraštinių normalizavimo, prieš tai išgautų kraštinių tikslumas sumažėja mažiausiai. Dešimt optimaliųjų sprendinių, pastatų tikslumo mažėjimo tvarka, pateikti lentelėje (2 priedas.). Remiantis pastatų apibrėžimo tikslumu optimalusis yra sprendinys, kuris lentelėje (2 priedas.) pažymėtas žalia spalva. Šis optimalusis sprendinys pažymėtas raudona spalva diagramoje (36 pav.). Šio optimaliojo sprendinio pastatų tikslumas yra 50,987 %, kai ϵ reikšmė yra 1,225.



36 pav. Rasti optimalūs sprendiniai objektų kraštinių normalizavimui taikant RDP algoritmą

Lyginant išgautas pastatų kraštines, taikant „Alpha Shapes“ algoritmą, su po to normalizuotomis pastatų kraštinėmis RDP algoritmu (37, 38 pav.), kraštinės po normalizavimo žymiai mažiau

kintančios, pastatus sudaro daug mažiau kampų, vaizdas atrodo daug priimtinau galutiniam vartotojui.



37 pav. Išgautos pastatų kraštinės taikant „Alpha Shapes“ algoritimą



38 pav. Normalizuotos pastatų kraštinės taikant RDP algoritimą

3.3.2. Objektų kraštinių normalizavimas taikant Guru's „cornerity“ indekso algoritimą

Norint nustatyti optimaliuosius normalizavimo parametrus, kai kiekvienas objektas yra jau apibrėžtas „Alpha Shapes“ algoritmu, kraštinės normalizuojamos taikant Guru „cornerity“ indekso algoritimą. Algoritmas buvo paleistas 196 kartus su atsitiktiniais parametrais. Įvertinus pastatų apibrėžimo tikslumą, taikant Žakardo indeksą [27] aibių panašumams nustatyti, buvo rasti optimalieji sprendiniai, su kuriais normalizuotos pastatų kraštinės tiksliausiai, kitaip tariant, dėl kraštinių normalizavimo, prieš tai išgautų kraštinių tikslumas sumažėja mažiausiai. Dešimt optimaliųjų sprendinių, pastatų tikslumo mažėjimo tvarka, pateikti lentelėje (11 lentelė.). Remiantis pastatų apibrėžimo tikslumu optimalusis yra sprendinys, kuris lentelėje (11 lentelė.) pažymėtas žalia spalva.

11 lentelė. Optimalieji sprendiniai kraštinių normalizavimui taikant Guru „cornerity“ indekso algoritimą

Vidutinis kampų skaičius	Pastatų tikslumas (%)	Medžių tikslumas (%)	Bendras tikslumas (%)	Parametrai		
				i	k	w
27,047	50,929	3,165	27,047	9	2	3
27,028	50,927	3,130	27,028	4	1	3
27,030	50,922	3,138	27,030	10	3	3

27,022	50,904	3,140	27,022	9	3	3
27,030	50,904	3,157	27,030	7	3	3
27,030	50,895	3,165	27,030	13	2	3
27,029	50,895	3,162	27,029	4	3	3
27,035	50,893	3,178	27,035	12	3	3
27,027	50,888	3,166	27,027	6	3	3
27,018	50,888	3,148	27,018	10	2	3

Lyginant normalizuotas pastatų kraštines, taikant RDP algoritmą, su normalizuotomis pastatų kraštinėmis Guru algoritmu (38, 39 pav.), kraštinės po normalizavimo žymiai mažiau kintančios, pastatus sudaro daug mažiau kampų, vaizdas atrodo daug priimtinau galutiniam vartotojui.



39 pav. Normalizuotos pastatų kraštinės taikant Guru „cornerity“ indekso algoritmą

3.3.3. Objektų kraštinių normalizavimo apibendrinimas

Apibendrinus visus kraštines vektorizuoti tirtus algoritmus abu algoritmai parodė panašų rezultatą. Šių algoritmų rezultatai pateikti lentelėje (12 lentelė.). Šių algoritmų esminis skirtumas yra vidutinis

kampų skaičius, kuris parodo jog Guru „cornerity“ indekso algoritmu, objektų kraštinės yra žymiai mažiau normalizuotos, labiau kintančios.

12 lentelė. Algoritmų rezultatų kraštinėms išgauti palyginimas

Algoritmas	Vidutinis kampų skaičius	Pastatų tikslumas (%)	Medžių tikslumas (%)	Bendras tikslumas (%)
RDP	13,424	50,987	2,782	26,885
Guru „cornerity“ indekso	27,047	50,929	3,165	27,047

3.4. Ekonominis pagrindimas

3.4.1. Kylančios „Google Maps“ paslaugų kainos

Nuo 2018 metų liepos 11 keičiasi „Google Maps“ mokėjimo planas [36]. Nuo šios datos „Google API“ prieiga be rakto nebebus palaikoma. Užklausos į „Maps JavaScript API“ ir „Street View API“ gražins mažos skiriamosios gebos žemėlapius su vandens ženklu, kuriame parašyta jog žemėlapis atvaizduojamas tik kūrimo tikslams.

„Google API“ paslaugos nebebus galimos tol kol nebus užregistruota kreditinės kortelės informacija „Google“ paskyroje. Ir tai galios visiems šių paslaugų vartotojams, net tiems, kurie atvaizduoja žemėlapi tik savo internetinės svetainės kontaktų puslapyje.

Natūralu jog „Google“ nenuskriaus mažųjų savo klientų, todėl suteiks 200 \$ nemokamai kiekvieną mėnesį. Už šią sumą vartotojas galės gauti 28 tūkst. užklausų per mėnesį. Viršijus ši kiekį bus mokama 14 \$ už kiekvieną 100 tūkst. užklausų, viršijus 100 tūkst., 11,20 \$. Tačiau šiuo metu vartotojai gauna 25 tūkst. nemokamų užklausų per dieną. Taigi vartotojui, kuris šį dienos limitą išnaudodavo, bet neviršydavo ir naudojosi paslaugomis nemokamai, ši paslauga jam kainuos apie 87 \$ per mėnesį.

Atsižvelgus į šią naujieną galime teigti jog nemaža dalis vartotojų nebus linkę mokėti pinigų už paslaugą, kuria naudojosi nemokamai. Todėl natūralu jog šie vartotojai ieškos alternatyvų. Viena iš tokių alternatyvų yra nemokama „OpneStreetMap“ arba šia sistema grįsti sprendimai. Nemaža vartotojų dalis pasirinktų šią informacinę žemėlapių sistemą jei ji atitiktų vartotojų kokybės reikalavimus. Taigi, pritaikius, baigiamajame darbe aprašytus, sprendimus galime šių žemėlapių

kokybę pagerinti ir tokiu būdu suteikti galimybę, daugiau vartotojų naudotis nemokamomis paslaugomis, taip nepadidinant išlaidas savo versluose.

3.4.2. Palydovinių vaizdų atpažinimo konkursas

Prognozavimo, modeliavimo bei analizės konkursų platformoje „Kaggle“, 2017 m. buvo rengtas konkursas palydoviniams vaizdams atpažinti [37]. Šį konkursą organizavo gynybos mokslo ir technologijų kompanija „dstl“. Šio konkurso idėja surasti naują sprendimą, kuris sumažintų našumą analitikams, atpažinti pastatus, įvairias dirbtines konstrukcijas, kelius, medžius, dirbamus laukus, upes, stovinčius vandenį, didelius bei mažus automobilius, naudojant palydovinius vaizdus. Šio konkurso nugalėtojai gavo 50 tūkst. \$, o antros ir trečios vietos nugalėtojai, atitinkamai 30 tūkst. \$ bei 20 tūkst. \$.

Galime teigti jog tokie konkursai parodo objektų atpažinimo iš nutolusių skanavimo šaltinių, metodų svarbą ne tik verslui, bet ir valstybei bei jos krašto apsaugai. Baigiamajame darbe pateikti vektorizavimo metodai taip pat galėtų būti taikomi ir kitų objektų (automobilių, kelių ir pan.) išgavimui iš taškinių LiDAR duomenų jei būtų surastas metodas juos klasifikuoti.

Išvados

- Siekiant ištirti aukštos augmenijos objektų apibrėžimo tikslumą reiktų pasirinkti tirti tokią vietą žemėlapyje, kurioje tiksliai būtų išbraižyti aukštos augmenijos plotai „OpenStreetMap“ informacinėje sistemoje.
- Geriausias algoritmas pastatų kraštinėms išgauti yra „Alpha Shapes“ algoritmas, kuris taikomas kiekvienam objekto klasteriui, kuris gautas taikant „DBSCAN“ klasterizavimo algoritmą.
- Normalizuojant objekto kraštines šiek tiek sumažėja objekto apibrėžimo tikslumas, tačiau vartotojui šis rezultatas labiau priimtinas, o taip pat gaunami žymiai mažesni vektorių dydžiai.
- Siekiant geresnių vektorizavimo rezultatų reiktų visus tirtus algoritmus kombinuoti pagal savybes, kuriuose jie buvo tiksliausi.

Literatūros sąrašas

1. ŽALNIERUKAS, A. and ČYPAS, K. Žemes Paviršiaus Skenavimo Lazeriu Iš Orlaivio Technologijos Analize. *Geodezija Ir Kartografija*, 2006, vol. 32, no. 4. pp. 101-105 ISSN 1648-3502.
2. *Digital Elevation Model – DEM*. Sinergise. [žiūrėta 2018-04-09]. Prieiga per: <https://sentinel-hub.com/blog/digital-elevation-model-dem>.
3. ŽALNIERUKAS, A., RUZGIENE, B., KALANTAITE, A. and VALAITIENE, R. Miestu Skenavimo LIDAR Metodu Tikslumo Analize. *Geodezija Ir Kartografija*, 01/01, 2009, vol. 35, no. 2. pp. 55-60 ISSN 1392-1541.
4. RESEARCH, a.M. Global LIDAR Market Forecasts to 2024 - Reliability Issues in Airborne Lidar - Research and Markets. *Business Wire (English)*, 2017.
5. WANG, H., et al. *Validation of a Low-Cost 2D Laser Scanner in Development of a More-Affordable Mobile Terrestrial Proximal Sensing System for 3D Plant Structure Phenotyping in Indoor Environment*. , August 2017, 2017 Prieiga per: <http://www.sciencedirect.com/science/article/pii/S0168169916307797> ISBN 0168-1699. Prieiga per: DOI <https://doi.org/10.1016/j.compag.2017.06.002>.
6. DEEMS, J.S., PAINTER, T.H. and FINNEGAN, D.C. Lidar Measurement of Snow Depth: A Review. *Journal of Glaciology*, 2017/07/10, 2013, vol. 59, no. 215 [žiūrėta 2018/05/30]. pp. 467-479. Prieiga per: <https://www.cambridge.org/core/article/lidar-measurement-of-snow-depth-a-review/4419DF5C778946103080CB6187D434C0> Cambridge Core. ISSN 0022-1430. Prieiga per: DOI: 10.3189/2013JoG12J154.
7. LUKAČ, N., et al. *Buildings Roofs Photovoltaic Potential Assessment Based on LiDAR (Light Detection and Ranging) Data*. , 1 March 2014, 2014 Prieiga per: <http://www.sciencedirect.com/science/article/pii/S0360544213011365> ISBN 0360-5442. Prieiga per: DOI <https://doi.org/10.1016/j.energy.2013.12.066>.
8. DU, S., et al. *Automatic Building Extraction from LiDAR Data Fusion of Point and Grid-Based Features*. , August 2017, 2017 Prieiga per:

<http://www.sciencedirect.com/science/article/pii/S0924271616306438> ISBN 0924-2716. Prieiga per: DOI <https://doi.org/10.1016/j.isprsjprs.2017.06.005>.

9. T. AGARWAL. *All You Know about LIDAR Systems and Applications*. [žiūrėta 2018-04-09]. Prieiga per: <https://www.elprocus.com/lidar-light-detection-and-ranging-working-application/>.

10. ARROWSMITH, R. Current Capabilities and Community Needs for Software Tools and Educational Resources for use with LiDAR High Resolution Topography Data, 09/08, 2008.

11. *LAStools*. rapidlasso GmbH. [žiūrėta 2018-04-09]. Prieiga per: <https://rapidlasso.com/lastools/>.

12. C. CROSBY. *LViz - LiDAR / ALSM 3D Visualization Tool*. 2007-01-08, 2007 [žiūrėta 2018-04-09]. Prieiga per: <http://lidar.asu.edu/LViz.html>.

13. *LidarViewer*. KeckCAVES. 2016-07-18, 2016 [žiūrėta 2018-04-09]. Prieiga per: <http://keckcaves.org/software/lidarviewer>.

14. C. CROSBY. *GEON Points2Grid Utility*. , 2007-06-24 [žiūrėta 2018-04-09]. Prieiga per: <http://lidar.asu.edu/points2grid.html>.

15. *QGIS - A Free and Open Source Geographic Information System*. 2018-05-23, 2018 [žiūrėta 2018-05-23]. Prieiga per: <https://qgis.org>.

16. *About PostGIS*. [žiūrėta 2018-05-12]. Prieiga per: <https://postgis.net/>.

17. *Postgresql: The World's most Advanced Open Source Relational Database*. , 2018 [žiūrėta 2018-05-23]. Prieiga per: <https://www.postgresql.org/>.

18. SHEN, W., ZHANG, J. and YUAN, F. *A New Algorithm of Building Boundary Extraction Based on LIDAR Data*. , 2011 ISBN 2161-024X. Prieiga per: DOI 10.1109/GeoInformatics.2011.5981049.

19. TSE, R., GOLD, C. and KIDNER, D. *Using the Delaunay Triangulation/ Voronoi Diagram to Extract Building Information from Raw LIDAR Data*. , 2007 Prieiga per: DOI 10.1109/ISVD.2007.40.

20. FREEMAN, H. and SHAPIRA, R. Determining the Minimum-Area Encasing Rectangle for an Arbitrary Closed Curve. *Commun.ACM*, jul, 1975, vol. 18, no. 7. pp. 409-413 ISSN 0001-0782.
21. TOUSSAINT, G. *Solving Geometric Problems with the Rotating Calipers.* , 1983.
22. DOUGLAS, D.H. and PEUCKER, T.K. *Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line Or its Caricature.* , 03/18; 2018/04, 2011 ISBN 9780470669488. Prieiga per: DOI doi:10.1002/9780470669488.ch2.
23. LI, L. and JIANG, W. *An Improved Douglas-Peucker Algorithm for Fast Curve Approximation.* , 2010 Prieiga per: DOI 10.1109/CISP.2010.5647972.
24. GURU, D.S., DINESH, R. and NAGABHUSHAN, P. *Boundary Based Corner Detection and Localization using New 'Cornerity' Index: A Robust Approach.* , 2004 Prieiga per: DOI 10.1109/CCCRV.2004.1301477.
25. HECKBERT, P.S. and GARLAND, M. *Survey of Polygonal Surface Simplification Algorithms,* 1997.
26. MIETTINEN, K. *Nonlinear Multiobjective Optimization.* Springer US, 2012 ISBN 9781461555636.
27. PAUL, J. The Distribution of the Flora in the Alpine Zone.1. *New Phytologist*, 02/01; 2018/05, 1912, vol. 11, no. 2. pp. 37-50 ISSN 0028-646X.
28. *Osm2pgsql.* 2018-01-27, 2018 [žiūrēta 2018-05-23]. Prieiga per:<https://wiki.openstreetmap.org/wiki/Osm2pgsql>.
29. *Lidar Point Classification.* Environmental Systems Research Institute. [žiūrēta 2018-05-12]. Prieiga per:<http://desktop.arcgis.com/en/arcmap/10.3/manage-data/las-dataset/lidar-point-classification.htm>.
30. J. CONWAY, et al. *R Interface to the 'PostgreSQL' Database System.* 2017-06-24, 2017 [žiūrēta 2018-05-12]. Prieiga per:<https://cran.r-project.org/web/packages/RPostgreSQL/RPostgreSQL.pdf>.

31. Ortofoto. 2013-06-05, 2013 [viewed 2018-05-12]. Available from:
<https://blog.openmap.lt/2013/06/05/ortofoto/>.
32. ROOCKS, P. Computing Pareto Frontiers and Database Preferences with the rPref Package. *Rj*, 2016, vol. 8, no. 2. pp. 393-404.
33. B. PATEIRO-LÓPEZ and R. ALBERTO. *Generalizing the Convex Hull of a Sample: The R Package Alphahull*. 2016-02-10, 2016 [žiūrėta 2018-05-19]. Prieiga per:<https://cran.r-project.org/web/packages/alphahull/alphahull.pdf>.
34. M. HAHLER, M. PIEKENBROCK, S. ARYA and D. MOUNT. *Density Based Clustering of Applications with Noise (DBSCAN) and Related Algorithms*. 2018-05-19, 2018 Prieiga per:<https://cran.r-project.org/web/packages/dbscan/dbscan.pdf>.
35. C. GENOLINI and E. GUICHARD. *K-Means for Longitudinal Data using Shape-Respecting Distance*. 2016-03-08, 2016 [žiūrėta 2018-05-22]. Prieiga per:<ftp://cran.r-project.org/pub/R/web/packages/kmlShape/kmlShape.pdf>.
36. *Pricing that Scales to Fit Your Needs*. [žiūrėta 2018-05-31]. Prieiga per:<https://cloud.google.com/maps-platform/pricing/>.
37. *Dstl Satellite Imagery Feature Detection*. [žiūrėta 2018-05-31]. Prieiga per:<https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection>.

Priedai

1 priedas. Rasti „Alpha Shapes“ optimalieji sprendiniai objektų kraštinių išgavimui

Vidutinis kampų skaičius	Pastatų tikslumas (%)	Aukštos augmenijos tikslumas (%)	Bendras tikslumas (%)	Parametras alpha
51,181	50,119	0,319	25,219	1,95
50,682	49,140	0,420	24,780	2
47,723	48,595	0,283	24,439	1,7
50,983	48,084	0,414	24,249	1,9
48,354	47,947	0,275	24,111	1,65
48,361	46,507	0,380	23,443	1,75
53,153	46,207	0,346	23,276	2,05
48,451	46,120	0,434	23,277	1,8
48,845	45,925	0,158	23,041	1,4
48,574	45,811	0,152	22,981	1,45

2 priedas. Rasti optimalūs sprendiniai objektų kraštinių normalizavimui taikant RDP algoritimą

Vidutinis kampų skaičius	Pastatų tikslumas (%)	Medžių tikslumas (%)	Bendras tikslumas (%)	Parametras ϵ
13,424	50,987	2,782	26,885	1,225
13,607	50,985	2,784	26,885	1,2
13,984	50,984	2,800	26,892	1,15
14,377	50,984	2,810	26,897	1,1
13,770	50,982	2,789	26,885	1,175
13,215	50,979	2,821	26,900	1,25
14,555	50,976	2,817	26,897	1,075
12,660	50,973	2,758	26,866	1,35
14,173	50,959	2,807	26,883	1,125
12,550	50,942	2,459	26,701	1,375