

Univerza
v Ljubljani

*Fakulteta za
gradbeništvo in
geodezijo*



Kandidat/-ka:

Magistrsko delo št.:

Master thesis No.:

Mentor/-ica:

Predsednik komisije:

Somentor/-ica:

Član komisije:

Ljubljana, _____

STRAN ZA POPRAVKE

Stran z napako Vrstica z napako

Namesto

Naj bo

»Ta stran je namenoma prazna.«

BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK:	004.6:004.76:624(043.2)
Avtor:	Uroš Jotanović
Mentor:	prof. dr. Žiga Turk
Somentor:	asist. dr. Robert Klinc, asist. dr. Klemen Kregar
Naslov:	Tehnologije za zajem podatkov snovnega sveta v informacijska okolja za gradbeništvo
Tip dokumenta:	Magistrsko delo
Obseg in oprema:	62 str., 5 preg., 56 sl.
Ključne besede:	oblak točk, segmentacija, klasifikacija, fotogrametrija, terestrično lasersko skeniranje, BIM

Izvleček

Uporaba oblakov točk pri pridobivanju podatkov za informacijsko modeliranje stavb je v zadnjem času vedno bolj pogosto. Upravljalci in lastniki starejših stavb in objektov si prizadevajo za centralizacijo informacij. Dokumentacija o strojnih inštalacijah, cevovodih, električni opremi in preteklih posegih je pogosto shranjena na dislociranih medijih, v večini primerov pa še vedno na papirju. Oblak točk v procesu pretvorbe snovnega sveta v digitalni predstavlja izhodišče, v katerem so zajeti podatki o snovnem svetu, ki jih pridobimo na več različnih načinov, kot so fotogrametrija, terestrično lasersko skeniranje in zračni lidar. Ročno modeliranje BIM za upravljanje, vzdrževanje in nadaljnjo uporabo je zamuden postopek, nagnjen k napakam, zato želimo ta proces avtomatizirati in se tem napakam izogniti.

V zadnjem času se pojavlja vedno več samostojnih programov in dodatkov za obstoječe programe, ki omogočajo avtomatizirano, hitro in bolj natančno modeliranje na osnovi surovih podatkov iz oblakov točk. Princip delovanja teh programov in dodatkov izhaja iz modelov računalniškega vida, ki omogočajo poenostavitev dela in zmanjšanje vloge uporabnika v celotnem procesu.

V magistrski nalogi smo raziskovali možnosti za avtomatizacijo procesa izdelave BIM modelov iz surovih oblakov točk. V prvem delu naloge smo primerjali izmenjevalne formate za zapis oblakov točk, v katerih so na različne lahko načine zapisani prostorski podatki. Primerjali smo različno programsko opremo, ki omogoča delo z oblaki točk in modeliranje na njihovi osnovi. Raziskali smo trenutne algoritme računalniškega vida, ki so osnova teh programskih rešitev. V drugem delu smo na konkretnih primerih oblakov točk prikazali postopek modeliranja posameznih BIM elementov.

V postopku modeliranja smo rezultate za posamezne elemente (stene, cevi in stebre) grafično prikazali in dokazali visoko stopnjo natančnosti avtomatiziranega modeliranja. Kljub avtomatizaciji celotnega procesa je za doseganje dobrih rezultatov še vedno potrebna uporabniška interakcija, ki pa se z algoritmi globokega učenja v zadnjih letih zmanjšuje.

BIBLIOGRAPHIC AND DOCUMENTATION INFORMATION AND ABSTRACT

UDC:	004.6:004.76:624(043.2)
Author:	Uroš Jotanović
Supervisor:	Prof. Žiga Turk, Ph. D.
Co-advisor:	Assist. Prof. Rober Klinc, Ph. D., Assist. Prof. Klemen Kregar, Ph. D.
Title:	Technologies for capturing material world data into civil engineering information systems
Document type:	M. Thesis
Notes:	62 p., 5. tab., 56 fig.
Keywords:	point cloud, segmentation, classification, photogrammetry, terrestrial laser scanning, BIM

Abstract

The use of point clouds in data acquisition for building information modeling has become common practice in recent years. Managers and owners of older buildings strive to centralize information about their assets. Current documentation regarding mechanical installations, pipelines, electrical equipment, and past interventions is often stored on dislocated media or on paper in many cases. Point cloud represents the starting point in the process of transforming the material world into digital, which includes raw data about the material world, that is obtained in several different ways, such as photogrammetry, terrestrial laser scanning and air lidar. Manual BIM modeling for management, maintenance and further use is a time-consuming, error-prone process, that we strive to automate.

Recently, more stand-alone programs and add-ons for existing programs are emerging, allowing for fast, automated, and more accurate modeling based on raw point cloud data. The principles of these programs and add-ons are based on computer vision models that simplify the work and reduce the role of the user in the entire process.

In the master's thesis, we explored the possibilities of automating the creation process of BIM models from raw point clouds. In the first part of the thesis, we compared point cloud exchange formats, in which the same spatial data can be written in different ways. We compared different software that allows working with point clouds and modeling. We explored the current computer vision algorithms that serve as basis for these software solutions. In the second part, we presented the process of modeling individual BIM elements on point cloud examples.

During the modeling process, we graphically presented the results for individual elements (walls, pipes and columns) and proved the high level of accuracy for presented process. Despite the high level of automation during the process, user interaction is still required to achieve good results. The need for user interactions is lowering with the advancements of the deep learning algorithms in recent years.

ZAHVALA

Za strokovno pomoč in konstruktivne komentarje pri izdelavi magistrskega dela se zahvaljujem mentorju prof. dr. Žigi Turku in somentorjema asist. dr. Robertu Klincu ter asist. dr. Klemnu Kregarju.

Zahvaljujem Deanu Kobaletu in asist. dr. Klemnu Kregarju za laserske posnetke.

Rad bi se zahvalil tudi družini in sodelavcem za pomoč in podporo v času pisanja zaključnega dela.

KAZALO VSEBINE

1	UVOD	1
2	PREGLED STANJA	3
2.1	Metode zajema oblakov točk	3
2.1.1	Terestrično lasersko skeniranje	4
2.1.2	Mobilna laserska izmera	5
2.1.3	Zračno lasersko skeniranje	6
2.1.4	Fotogrametrija	7
2.2	Zapis oblakov točk	9
2.2.1	Formati za zapis oblakov točk	9
2.2.2	Interoperabilnost različnih formatov	12
2.2.3	Začetne zahteve za razvoj formata .e57	13
2.2.4	Struktura formata .e57	13
2.3	Vizualizacija in organizacija oblakov točk	15
2.3.1	Organizacija oblakov točk	15
2.3.2	Normale točk	15
2.3.3	Delaunayeva triangulacija	16
2.3.4	Octree podatkovna struktura	17
2.3.5	k-D tree podatkovna struktura	18
2.4	Segmentacija in klasifikacija oblakov točk	19
2.4.1	Segmentacija oblakov točk	19
2.4.2	Segmentacija na podlagi zaznavanja robov	19
2.4.3	Segmentacija na podlagi rasti regij	20
2.4.4	Klasifikacijske metode s prileganjem elementov	20
2.4.4.1	Houghova transformacija	21
2.4.4.2	Metoda naključnega vzorčenja	22
2.4.5	Nenadzorovano deljenje	23
2.4.5.1	Metoda grozdenja na osnovi K-srednjih vrednosti	23
2.4.5.2	Metoda grozdenja s premikom srednjih vrednosti	24
2.4.6	Supervokselizacija in predsegmentacija	24
2.4.7	Segmentacija na osnovi globokega učenja	25
2.4.7.1	Segmentacija na osnovi več pogledov	25
2.4.7.2	Segmentacija na osnovi vokselov	26
2.4.7.3	Segmentacija na osnovi točk	26
2.5	Programska oprema za delo z oblaki točk	27
2.5.1	Autodesk Recap	27
2.5.2	CloudCompare	27
2.5.3	Revit	27
2.5.4	Faro As-Built	28
2.5.5	Leica Cloudworx	29
2.5.6	ClearEdge3D EdgeWise	29
2.5.7	PointCab	30
2.5.8	Sklep	31
3	DELOTOK PRETVORBE OBLAKOV TOČK V BIM ELEMENTE	33
3.1	Uvod	33
3.2	Priprava oblakov točk	34
3.2.1	Priprava izrezov oblakov točk	34
3.2.2	Priprava podlag na osnovi izrezov	36
3.3	Avtomatizirano modeliranje sten	38
3.3.1	Izbira točk	38
3.3.2	Analiza odstopanja modela in oblaka točk	41
3.4	Avtomatizirano modeliranje cevi	44

3.4.1	Izbira točk	44
3.4.2	Povezave med elementi	47
3.4.3	Analiza odstopanja	48
3.5	Modeliranje nosilcev in stebrov	49
3.5.1	Izbira točk	50
3.5.2	Analiza odstopanja	53
4	ZAKLJUČEK	54
	VIRI	56

KAZALO SLIK

Slika 1 Shematski prikaz glavnih načinov zajema prostorskih podatkov.....	3
Slika 2 Terestrični laserski skener (levo) in prikaz delovanja mehanizma za pošiljanje in sprejem žarka (desno) (prirejeno po Oguchi, 2011)	5
Slika 3 Primerjava mobilnega laserskega skenerja (levo) in LiDAR (desno) na istem odseku ceste. Na levi sliki je večja gostota točk, hkrati pa so detajli bolj očitni (Wang, 2019).....	6
Slika 4 Digitalni model površine (DSM) (levo) in digitalni model očiščenega terena (DTM) (Vosselman in Maas, 2011).....	7
Slika 5 Princip delovanje fotogrametrije na osnovi stereoskopskega vida (Linder, 2016)	8
Slika 6 Primer ASCII zapisa točk z atributoma intenzitete in barve.	9
Slika 7 Oblak točk shranjen v formatu .e57 (Huber, 2011).....	12
Slika 8 Prikaz izseka XML dela zapisa znotraj formata .e57 (povzeto po libe57.org)	14
Slika 9 Težave pri prikazu oblakov točk. Na prvi sliki je opazovalec blizu objekta, velikost točk je majhna, zato se pojavijo praznine. Na drugi sliki so točke večje, prikaz pa je zato manj natančen. Na tretji sliki je opazovalec daleč, vidni pa so samo glavni detajli (Vosselman in Maas, 2011).....	15
Slika 10 Grafični prikaz prilagoditve normale n na točko q . Z metodo najmanjših kvadratov prilagodimo ploskev H na k okoliških točk (Vosselman in Maas, 2011)	16
Slika 11 Pogled na trianguliran teren z vrha (levo) in s strani (desno) (Vosselman in Maas, 2011).....	17
Slika 12 Rekurzivna razdelitev kocke na oktante (levo) in ustrezna oktree drevesna struktura (desno) (povzeto po Funk in Borner, 2016)	18
Slika 13 Rekurzivna delitev kvadrata (levo) in usrezna k-D tree drevesna struktura (desno) (povzeto po Teniente in Cetto, 2008).....	18
Slika 14 Model razdeljen v octree strukturi (levo) in k-D tree strukturi (desno) (Vosselman in Maas, 2011).....	19
Slika 15 Segmentacija stopnic z algoritmom rasti regij. Vsaka regija je v drugi barvi (Rabbani, 2006)	20
Slika 16 Prikaz zapisa točk v kartezičnem koordinatnem sistemu (levo) in parametričnem (desno) (Bergues in sod, 2019).....	21
Slika 17 Prikaz vhodnih podatkov v 2D prostoru (levo) in pravilnega izračuna (desno) (Wikipedia, 2020)	22
Slika 18 Prilagoditev stebra oblaku točk s pomočjo algoritma RANSAC (Grilli, 2017)	23
Slika 19 Deljenja oblaka točk v dveh dimenzijah s pomočjo algoritma za grodeznje na osnovi K-srednjih vrednosti. Trikotniki predstavljajo težišče grozda (Benavente, 2017)	24
Slika 20 Segmentacija oblakov točk s pomočjo globokega učenja (Qi, 2017).....	25
Slika 21 3D oblak točk je razdeljen na segmente 2D slik, ki se ločeno segmentirajo in klasificirano ter potem združijo v 3D (Boulch, 2018).....	26
Slika 22 Pojavno okno z napako pri koordinatah visokih vrednosti (povzeto po Faro, 2019).....	28
Slika 23 Model iz oblaka točk v programu Faro As – Built (Faro, 2019).....	28
Slika 24 Prikaz modeliranih cevi znotraj programa EdgeWise (ClearEdge3D, 2020).....	30
Slika 25 IDEF0 diagram pretvorbe snovnega sveta v BIM model.	33
Slika 26 Ilustracija postopka priprave oblakov točk za modeliranje sten znotraj tega poglavja.	34
Slika 27 Primerjava urejenega oblaka točk (levo) in neurejenega (desno).....	35
Slika 28 Primerjava različnih debelin prereza oblaka točk. Na levi je prerez tanjši in zato primernejši, na desni je debelejši in manj primeren.	36
Slika 29 Pojavno okno za rasterizacijo tlorisa.....	36

Slika 30 Primerjava surovega oblaka točk v tlorisu (zgoraj) in pripravljenega pogleda (spodaj).....	37
Slika 31 Ilustracija postopka avtomatiziranega modeliranja sten znotraj tega poglavja.....	38
Slika 32 Pojavno okno za detekcijo in umestitev sten	38
Slika 33 Prikaz delovanja algoritma za umestitev sten (povzeto po FARO, 2019)	39
Slika 34 Prikaz detekcije in umestitve stene na pripravljeno tlorisno podlago – primer 1	39
Slika 35 Prikaz detekcije in umestitve stene na pripravljeno tlorisno podlago - primer 2	40
Slika 36 Prikaz detekcije in umestitve stene na pripravljeno tlorisno podlago - primer 3	40
Slika 37 Grafični prikaz analize odstopanj točk glede na modelirano steno.....	41
Slika 38 Pojavno okno za analizo odstopanj	41
Slika 39 Grafični prikaz različnih vrst analiz odstopanj (povzeto po FARO, 2019).....	42
Slika 40 Ilustracija postopka avtomatiziranega modeliranja cevi znotraj tega poglavja.....	44
Slika 41 Izbira referenčnih točk na oblaku točk za umestitev cevi	45
Slika 42 Pojavno okno za modeliranje cevi	45
Slika 43 Prikaz modeliranja cevi na polovici znanih točk	46
Slika 44 Prikaz modeliranja cevi na četrtini znanih točk	46
Slika 45 Prikaz dveh cevi v prostoru. Na levi sliki šuma ni videti. Na desni je prikazan prečni prerez cevi, kjer je šum očiten.....	47
Slika 46 Izbira prostih robov, ki jih želimo medsebojno spojiti.....	47
Slika 47 Izbira prave povezave med cevmi. Na drugi sliki je pravilni kolenčni kos, na tretji nepravilni T kos	48
Slika 48 Prikaz nepravilnosti na spojih, ki je potrebno odpraviti.....	48
Slika 49 Prikaz pravilno modeliranih in povezanih cevi. Levo je prikaz z oblakom točk v ozadju, desno brez.....	49
Slika 50 Prikaz odstopanja modeliranih cevi od osnovnega oblaka točk.....	49
Slika 51 Ilustracija postopka avtomatiziranega modeliranja jeklenih elementov znotraj tega poglavja.	49
Slika 52 Pojavno okno za modeliranje stebrov in nosilcev	50
Slika 53 Pojavno okno za izbiro pravega elementa na osnovi prečnega prereza	51
Slika 54 Prikaz jeklenega profila z manjkajočim delom stojine (levo) in modeliran element na osnovi oblaka točk (desno)	52
Slika 55 Prikaz jeklenega profila z manjkajočo pasnico (levo) in prečni prerez (desno)	52
Slika 56 Prikaz odstopanja modeliranih jeklenih profilov od osnovnega oblaka točk.....	53

KAZALO PREGLEDNIC

Preglednica 1 Seznam formatov za zapis oblakov točk (povzeto po cloudcompare.org)	11
Preglednica 2 Tabelarični prikaz strukture zapisa znotraj formata .e57 (Huber, 2011)	14
Preglednica 3 Pregled programske opreme za modeliranje in delo z oblaki točk	31
Preglednica 4 Prikaz različnih vrst analiz, ki so mogoče znotraj programa As – Built (lasten vir)	43
Preglednica 5 SWOT analiza glavnih elementov pretvorbe snovnega sveta v digitalni.	55

1 UVOD

Trenutni napredki v geodeziji omogočajo vse bolj natančen in vsebinsko bogat zajem prostorskih podatkov v obliki oblakov točk. Oblak točk je množica podatkovnih točk v prostoru, ki predstavlja digitalni ekvivalent snovnega sveta. Zajeti oblaki točk redko predstavljajo končni produkt, zato se je na trgu v zadnjih letih pojavilo več komercialnih programov, ki na njihovi osnovi omogočajo modeliranje CAD in BIM.

Oblaki točk predstavljajo nestrukturirane in semantično prazne skupine točk, ki imajo velikokrat prisotne šume in podobne pomanjkljivosti, ki nastanejo kot posledica terenskega zajema. Uporabniki si oblake točk vizualno dobro predstavljamo, težava se pojavi pri pretvorbi teh preprostih točkovnih podatkov v informacijsko bogate modele. Sodobni algoritmi morajo biti sposobni preproste točke združiti v smiselne celote, te celote pa semantično in informacijsko obogatiti, s čimer zaključene celote dobijo smisel tudi v digitalnem okolju. Grajene strukture se razlikujejo od naravnih, zato so potrebni robustni algoritmi za delo s specifičnimi oblikami, s katerimi se srečujemo v grajenem okolju.

Za pretvorbo snovnega sveta v digitalni je potrebno izvesti segmentacijo, klasifikacijo, vizualizacijo in strukturiranje.

- V postopku segmentacije točke različnimi algoritmi razvrstimo v različna homogena območja, s čimer imajo točke v izolirani regiji podobne lastnosti.
- Pri klasifikaciji z različnimi algoritmi določimo semantične lastnosti posameznim izoliranim regijam. Skupina linijskih točk tako na primer ne predstavlja zgolj skupino brez pomenskih točk, ampak steno.
- V postopku vizualizacije točke na osnovi njihovih lastnosti prikazujemo v različnih pogledih in z različnimi grafičnimi vmesniki.
- S strukturiranjem oblake točk pametno in varčno zapisujemo v različne podatkovne strukture.

Proizvajalci programske opreme obljublajo poenostavitve pri obdelavi in izmenjavi podatkov ter različne stopnje avtomatizacije v procesu modeliranja iz oblakov točk. Da bi preverili te trditve, smo raziskali zapise oblakov točk in trenutno programsko opremo za modeliranje snovnega sveta. Z njihovo pomočjo smo izdelali informacijsko in semantično polne modele, kjer posamezni gradniki predstavljajo elemente gradbene stroke (stene, stebri, nosilci) in ne zgolj skupine osnovnih prostorskih podatkov, kot so koordinate in barve znotraj oblakov točk.

Primerjali smo konkurenčne programe za modeliranje in iskali prosto dostopne programske rešitve na tem področju. S programom Faro As – Built smo na različnih oblakih točk preverili, kakšna stopnja avtomatizacije je trenutno mogoča ter kakšen del procesa še vedno poteka ročno. Po navodilih proizvajalca smo na oblakih točk, zajetih z različni laserskimi skenerji, modelirali stene, stebre in cevi. Na koncu vsakega sklopa smo naredili analizo odstopanja in preverili natančnost modeliranja na podlagi referenčnega oblaka točk.

Naloga je sestavljena iz dveh glavnih poglavij in zaključka. V prvem poglavju je predstavljen trenutni tehnološki razvoj metod za delo z oblaki točk. V drugem poglavju pa je na konkretnih oblakih točk prikazan avtomatiziran delotok modeliranja s programom Faro As – Built.

Prvo poglavje je razdeljeno na pet podpoglavij. V prvem podpoglavju so predstavljene različne tehnologije zajema prostorskih podatkov. Temu sledi drugo podpoglavje, ki obravnava datotečni zapis zajetih prostorskih podatkov. V tretjem podpoglavju so predstavljene grafične in matematične metode

za vizualizacijo in strukturiranje oblakov točk. V četrtem podpoglavju so predstavljeni različni algoritmi segmentacije in klasifikacije oblakov točk. V zadnjem podpoglavju so predstavljeni različni programi za delo z oblaki točk in BIM modeliranje.

Na osnovi raziskanih postopkov iz prvega poglavja je v drugem poglavju na konkretnih primerih predstavljen avtomatiziran postopek modeliranja oblakov točk v BIM. To poglavje je vsebinsko razdeljeno na štiri dele glede na obravnavane vrste BIM elementov. V prvem podpoglavju je splošno prikazana predpriprava podatkov, v drugem modeliranje sten, v tretjem modeliranje cevi, v zadnjem pa je prikazano modeliranje stebrov in nosilcev.

V zaključku so povzeti rezultati postopkov iz drugega poglavja. Prikazane so njihove prednosti, slabosti, pomanjkljivosti in predlogi za dodatne izboljšave na tem področju.

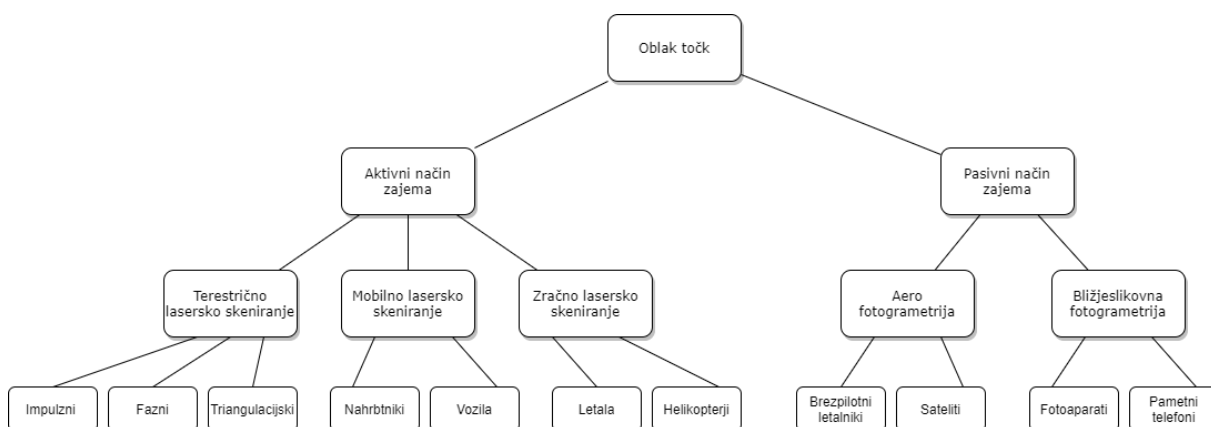
2 PREGLED STANJA

2.1 Metode zajema oblakov točk

Razvoj metod in tehnologij za zajem podatkov v geodeziji in geoinformatiki je v zadnjih letih omogočil zajem ogromnih količin terenskih podatkov o snovnem svetu. Ti podatki so niz točk P_i $i = 1, \dots, n$, ki se nahajajo v tridimenzionalnem kartezijanskem koordinatnem sistemu. Značilnosti posameznih točk pa so odvisne od vrste zajema prostorskih podatkov, ki je lahko aktiven ali pasiven. Aktivni način predstavlja vsakršno obliko laserskega zajema prostorskih podatkov, kjer so glavne značilnosti točk intenziteta odboja, red odboja in normale. Pri pasivnem zajemu so prostorski podatki pridobljeni z oblikami fotogrametrije, kjer so značilnosti barve, normale in kvaliteta preračuna. Različni načini zapisa in algoritmi branja teh zapisov, za zapis in branje so pogostokrat zapleteni, zato je ustrezno upravljanje teh naborov podatkov nujno.

Izraz oblak v besedni zvezi odraža neorganiziranost nabora podatkov, hkrati pa njihovo prostorsko povezanost. Zaradi svoje strukture imajo oblaki točk pogostokrat nejasno definiran rob. Georeferencirani oblaki točk so modeli resničnosti, povezani s točno določenim mestom, ki je zapisano v specifičnem koordinatnem sistemu. S tem so natančno umeščani v prostor. Vsaka točka P_i ima tri koordinate $(x_i; y_i; z_i) \in \mathbb{R}^3$, hkrati pa ima lahko še dodatne attribute, ki so odvisni od načina zajema. Dodatni atributi so lahko RGB barve posamezne točke, intenzivnost odboja in red odboja.

Oblake točk, ki opisujejo topografijo, lahko zajamemo z laserskim skeniranjem, fotogrametrijo, radargrametrijo, tomografijo SAR (angl. Synthetic Aperture Radar) in sonarjem MBES (angl. Multi-beam Echo Sounding). V literaturi ni točne opredelitve, ki bi določala velikost oblaka in razdaljo med posameznimi točkami. Vse je odvisno od metode in platforme za zajem podatkov. Najmanjše razdalje med predmetom in skenerjem so lahko velike nekaj centimetrov, zajamemo jih pa z industrijskimi triangulacijskimi laserskimi skenerji. Večji oblaki točk pa so lahko veliki več metrov in so pridobljeni s terestričnimi metodami skeniranja ali več sto metrov, če so pridobljeni z zračnim laserskim skeniranjem ali satelitsko altimetrijo. Posledica različnih načinov zajema je tudi različna gostota podatkov in stopnja njihove natančnosti. Primer zajema na večjem območju je mesto Dunaj s površino 400 km², ki je bilo leta 2008 skenirano z gostoto 20 točk na kvadratni meter, kar je skupaj zneslo 10 milijard točk. Za upravljanje tako velikega števila točk v oblaku potrebujemo ustrezne podatkovne strukture in algoritme za zapis, obdelavo in branje.



Slika 1 Shematski prikaz glavnih načinov zajema prostorskih podatkov.

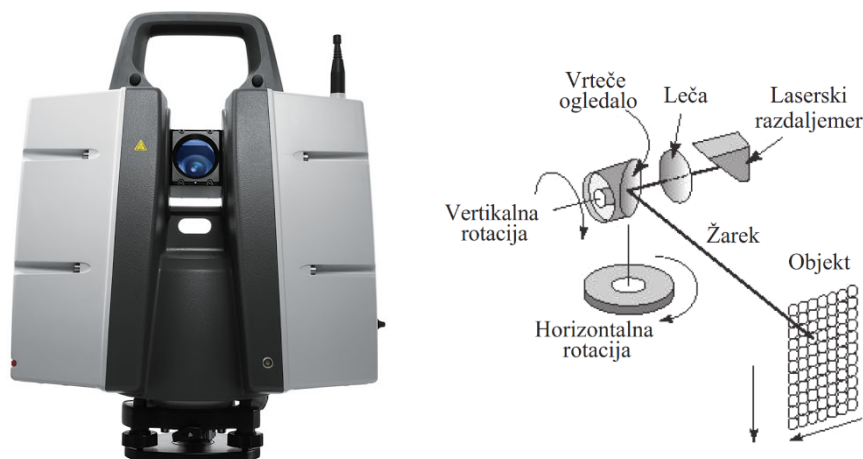
Oblakov točk ni mogoče uporabljati samo za vizualizacijo, ampak tudi za pridobivanje kvantitativnih informacij, kot so razdalje, koti in površine. Vse te količine dobimo iz oblaka točk direktno, brez posebne dodatne obdelave. Korak naprej je avtomatizirana analiza, vključno s semantično klasifikacijo in geometrijskim modeliranjem (Otepka in sod., 2013).

2.1.1 Terestrično lasersko skeniranje

S terestričnim laserskim skeniranjem (TLS), imenovanim tudi prizemni lidar ali topografski lidar, pridobivamo trirazsežnostne koordinate številnih točk z oddajanjem laserskih impulzov proti površinam in merjenjem razdalj od naprave do odbojnih površin (Vosselman in Maas, 2011). Proizvajalci sodobnih naprav TLS ponujajo merjenje 10^4 – 10^6 točk na sekundo z natančnostjo ~ 5 mm. Zaradi velike količine podatkov shranjenih v oblakih točk je za ogled, analizo in obdelavo potrebna namenska programska oprema. Mnogi skenerji v času skeniranja beležijo tudi digitalne fotografije. Iz fotografij lahko izvlečejo barvne vrednosti RGB in jih dodelijo vsaki točki. S takim postopkom nastane fotorealističen trirazsežnostni model. Ločljivost dobljenih podatkov je boljša od podatkov, pridobljenih z letalskimi in satelitskimi tehnikami zajema.

Instrumente TLS lahko razdelimo v tri kategorije glede na razdaljo, ki jo lahko laserska svetloba prepotuje. Ločimo skenerje kratkega, srednjega in dolgega dosega. Naprave, optimizirane za dolge dosege (od nekaj sto metrov do nekaj kilometrov), se uporabljajo za merjenje prostorsko večjih površin (Hunter in sod., 2003; Abellán idr., 2006), medtem ko skenerji krajšega obsega merijo prostorsko manjše površine (do nekaj sto metrov) z večjo natančnostjo in več podrobnostmi (Heritage in Large, 2009). Pri skenerjih kratkega dosega je interval med sosednjimi točkami do 1 mm. V gradbeništvu pa taka gostota ne igra velike vloge.

Glede na princip merjenja dolžin instrumente razdelimo: impulzni (angl. Time of Flight), fazni (angl. Phase shift) in triangulacijski skenerji. Za tehnologijo snemanja se odločimo na podlagi različnih dejavnikov, ki jih želimo doseči: razdalja, hitrost, ločljivost in natančnost pridobljenih podatkov. Impulzni skenerji dosegajo največje razdalje, hkrati pa zaradi tehnologije pošiljanja žarkov pridobivajo podatke počasneje in so tudi manj natančni. Fazni skenerji so pogosto opredeljeni kot najhitrejši, saj oddajajo neprekinjen žarek in lahko posnamejo več kot 100.000 točk/sek z večjo natančnostjo kot impulzni. Zaradi tehnologije pa imajo krajši domet. Tako fazni kot impulzni instrumenti se uporabljajo pri terestričnem skeniranju, kjer je mogoče obdelovati večje površine ali strukture; vse od nekaj metrov pa do več kilometrov. Nasprotno pa triangulacijski skenerji običajno dosegajo manj kot 5 metrov zaradi omejenega vidnega polja laserja in kamere. Običajno se triangulacijski sistemi imenujejo sistemi kratkega dosega in so najbolj primerni za skeniranje manjših predmetov, ki so v velikosti od 1 cm do 2-3 metra (odvisno od instrumenta). Natančnost in preprostost tehnik TLS omogočajo raziskovalcem izvajanje ponavljajočih raziskav s primerljivimi rezultati (Oguchi in sod., 2011).



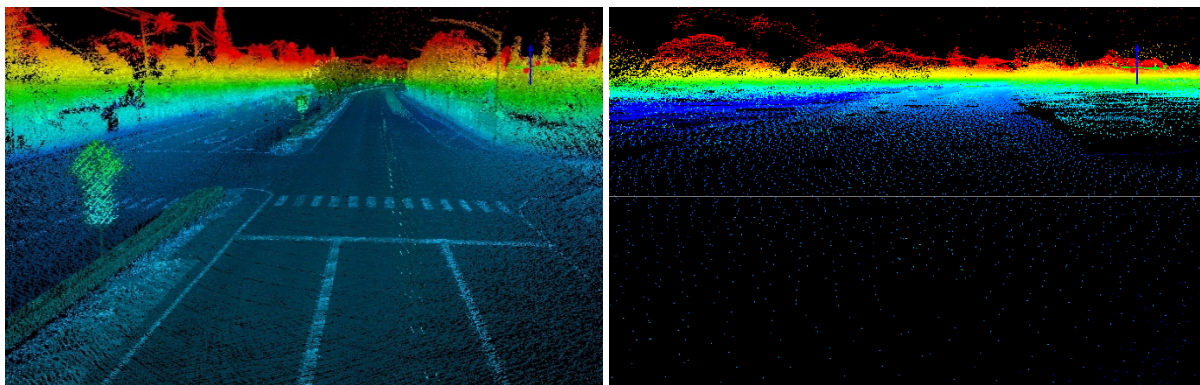
Slika 2 Terestrični laserski skener (levo) in prikaz delovanja mehanizma za pošiljanje in sprejem žarka (desno) (prirejeno po Oguchi, 2011)

2.1.2 Mobilna laserska izmera

Mobilno lasersko skeniranje (MLS) je tehnologija skeniranja z mobilne platforme (vozila). Natančni oblaki točk so pomemben vir podatkov za 3D mestne modele, ki se uporabljajo pri urbanističnem načrtovanju, simulacijah, kartiranju, vizualizaciji in avtonomnih vozilih. Pri uporabi in obdelavi teh podatkov so ključni načini združevanja, klasifikacije in segmentacije ogromnih količin podatkov.

Mobilni laserski skenerji se glede na namen uporabe lahko namestijo na vozila (avtomobil, vlak), nahrbtnike ali čolne. Ključne komponente mobilnega laserskega skenerja so laserski skener, GNSS (angl. Global Navigation Satellite System), IMU (angl. Inertial Measurement Unit) in fotoaparati. Takšne naprave so v praksi kompaktne, varne za uporabnika, lahke in pokrivajo celotno vidno polje v okolici. Mobilni laserski skenerji omogočajo zajem večje gostote točk v primerjavi z zračnimi laserskimi skenerji. Povprečna gostota točk pri mobilnih laserskih skenerjih je več tisoč točk na kvadratni meter, pri zračnem skeniranju pa je točk na kvadratni meter le nekaj deset. Hitrost zajema je zaradi kontinuiranega načina delovanja večja kot pri terestričnem laserskem skenerju, kjer je potrebno napravo vsakič ročno prestaviti in namestiti. Oblak točk, zajet na tak način, je dosti bolj homogen in gost. Kontinuirano zbiranje podatkov z visoko točkovno gostoto omogoča zajem podrobnih funkcij ceste, kot so robniki in površinske oznake ceste.

Podatke MLS uporabljamo pri razvrščanju cestnih površin, modeliranju cestnega okolja in okoliških objektov. Z raznimi algoritmi lahko določimo os ceste ali železniške proge, robove cest, ulične robnike in ostale obcestne objekte. Poleg cestnih objektov lahko zajamemo in naknadno rekonstruiramo tudi fasade ter obrise objektov z veliko natančnostjo. Znotraj objektov, kjer je GNSS lociranje oteženo, ali celo nemogoče, naprave za zajem (vozički, nahrbtniki) uporabljajo SLAM (angl. »Simultaneous Localisation and Mapping«) tehnologijo. Poleg prednosti, je pri MLS tudi nekaj omejitev. Posledice visoke gostote zajema so velike datoteke, ki so navadno tudi zahtevne za uporabo in obdelavo. Tako kot pri ostalih metodah je tudi pri MLS šum neizogiben problem. Skener ne more zajeti objektov, ki jih zakrivajo ovire, kot so drevesa, znaki, pešci, avtomobili in ostali obcestni objekti. Oblake točk pridobljene z MLS, je zato praktično kombinirati z drugimi (ALS, TLS) (Wang in sod., 2019).



Slika 3 Primerjava mobilnega laserskega skenerja (levo) in LiDAR (desno) na istem odseku ceste. Na levi sliki je večja gostota točk, hkrati pa so detajli bolj očitni (Wang, 2019)

2.1.3 Zračno lasersko skeniranje

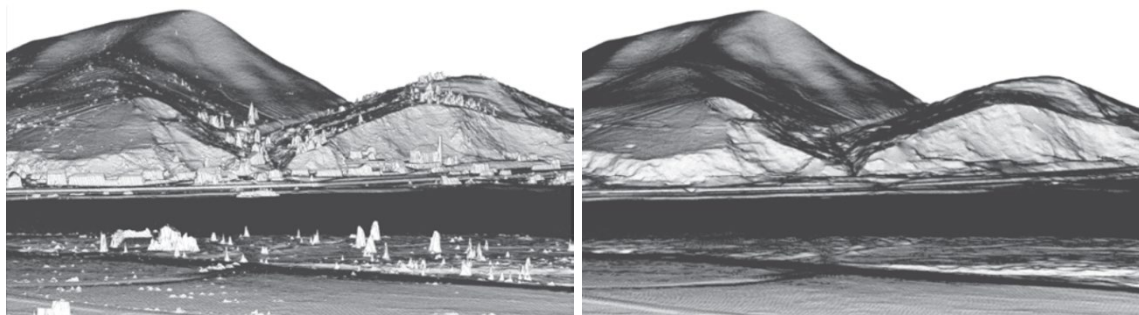
Lasersko skeniranje iz zraka se izvaja z letalom ali s helikopterjem. Tehnika temelji na dveh glavnih komponentah: sistemu laserskega skenerja, ki meri razdaljo do mesta na tleh v kombinaciji z GNSS in IMU enotami za natančno merjenje položaja in orientacijo sistema. Aktivni sistemi, ki temeljijo na laserskem skeniranju, so relativno neodvisni od sončne svetlobe. Uporabljajo se lahko podnevi ali ponoči. To pa je velika prednost v primerjavi z drugimi.

Zračni laserski skener je sestavljen iz laserskega skenerja, GNSS antene, IMU enote, enote za shranjevanje in sinhronizacijo podatkov in digitalne kamere. Laserski sistem (večinoma impulzni laser) je nameščen nad luknjo v trupu letala in neprestano pošilja laserske impulze proti terenu. Glede na hitrost zrakoplova in višino merjenja omogoča trenutna tehnologija merjenje z gostoto 50 točk/m². Standardna dvofrekvenčna GNSS antena je nameščena na izpostavljenem mestu na zrakoplovu, s tem pa omogoča nemoteno satelitsko povezavo. IMU je pritrjen neposredno na laserski skener ali na platformi v njegovi bližini. Navadno beleži podatke o pospešku in nagibu s frekvenco vzorčenja 200 Hz. Podatki o pospeških se lahko uporabijo kot pomoč pri interpolaciji položaja GNSS na poti, medtem ko se hitrost nagiba uporablja za določitev orientacije. Kombinacija podatkov GNSS in IMU omogoča rekonstrukcijo poti leta z natančnostjo znotraj 10 cm. Enota za nadzor in shranjevanje podatkov je odgovorna za časovno sinhronizacijo, nadzor in zapis podatkov. Poleg tega pa shranjuje metapodatke o snemanju in lokaciji, ki jih zbirajo laserski skener, IMU in GNSS. Sodobni laserski optični skenerji ustvarijo do 300 000 laserskih impulzov na sekundo in ustvarijo približno 20 GB podatkov na uro. GNSS in IMU enoti za primerjavo zapišeta približno 0,1 GB na uro.

Digitalne kamere pogostokrat dopolnjujejo sisteme zračnega laserskega skeniranja. Fotografije, posnete sočasno z ostalimi podatki o območju, lahko znatno izboljšajo interpretacijo podatkov v primerih, ko je prepoznavanje delov laserskih posnetkov oteženo. Največja gostota in kakovost podatkov sta pridobljeni s pulzi visokih frekvenc, nizko hitrostjo letenja, nizko višino letenja in pravim kotom snemanja.

Komercialni zračni laserski sistemi delujejo na valovnih dolžinah med 800 nm in 1550 nm. Spektralna širina je običajno med 0,1 in 0,5 nm. Ker je odbojnost predmeta odvisna od valovne dolžine žarka, imajo zato različni laserski sistemi svoje prednosti in slabosti pri skeniranju zemeljskega površja. Glede na pričakovano podlago snemanja je potrebno predvideti optimalno valovno dolžino. Pri valovnih dolžinah blizu vidnega dela spektra se žarek absorbira v vodi, zato so vodne površine pri teh valovnih dolžinah nepokrite. Poleg zagotavljanja podatkov na velikih površinah odprtega terena se zračno lasersko

skeniranje od drugih tehnologij daljinskega zaznavanja razlikuje tudi po zmožnosti merjenja terena pod krošnjami dreves in ostale vegetacije. Pri letu nad gozdnim območjem nekateri žarki prodrejo globje v drevesne krošnje in ustvarijo več odbojev. Število in faktor povratnih odbojev (oblika signala) sta odvisna od vrste in orientacije odbojne površine. Na ta način dobimo podatke o vrsti vegetacije in terenu pod njo. (Vosselman in Maas., 2010)



Slika 4 Digitalni model površine (DSM) (levo) in digitalni model očiščenega terena (DTM) (Vosselman in Maas, 2011)

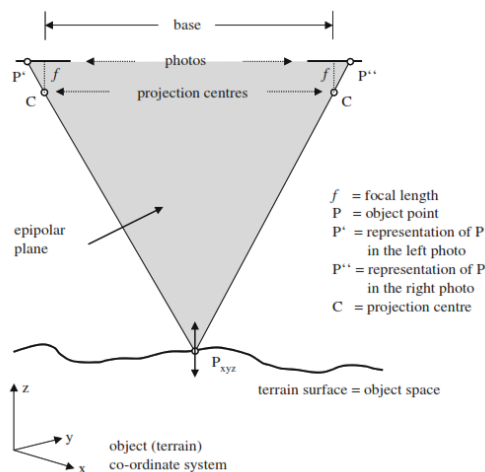
2.1.4 Fotogrametrija

Fotogrametrija je znanost in tehnologija pridobivanja zanesljivih informacij o fizičnih predmetih in okolju s postopki snemanja, merjenja in interpretacije fotografij (Wolf in Dewitt, 2000; McGlone, 2004). Fotogrametrija je stara skoraj toliko kot fotografija in se je od svojega razvoja pred približno 150 leti premaknila iz povsem analogne - optično-mehanske tehnike, v računalniško podprto digitalno fotogrametrijo, ki temelji na algoritmih računalniškega vida.

Fotogrametrija se ukvarja z natančnimi meritvami tridimenzionalnih predmetov in značilnosti terena na osnovi zajetih dvodimenzionalnih fotografij. Aplikacije vključujejo merjenje koordinat, razdalj, višin, površin, volumnov, pripravo topografskih zemljevidov in generiranje digitalnih modelov nadmorskih višin. Obstajata dve splošni vrsti fotogrametrije: aerofotogrametrija (s kamero v zraku) in bližjeslikovna fotogrametrija (s fotoaparatom v roki ali na stativu). V aerofotogrametriji je kamera nameščena na letalu (letalo, helikopter ali brezpilotni zrakoplov) in je običajno usmerjena navpično proti terenu. Letalo med letom snema medsebojno prekrivajoče se fotografije. Tradicionalno so bile te fotografije obdelane v stereo-ploterju (instrument, ki operaterju omogoča ogled dveh fotografij istočasno), zdaj pa jih obdelujejo namizni računalniki.

Pri bližneslikovni fotogrametriji je kamera nameščena na stativu ali v rokah uporabnika. Običajno je ta vrsta fotogrametrije netopografska, kar pomeni, da rezultati niso topografski izdelki, kot so modeli terena ali topografski zemljevidi, temveč so to risbe, oblaki točk ali 3D modeli. Vsakdanje kamere se lahko uporabljajo za zajemanje podatkov o zgradbah, inženirskih konstrukcijah, forenzičnih situacijah, nesrečah, rudnikih, zemeljskih delih, zalogah in arheoloških predmetih. Iz ene same fotografije lahko dobimo dvodimenzionalne koordinate, za pridobitev tretje dimenzije potrebujemo vsaj še eno dodatno fotografijo (Aber in sod., 2010). Metoda, s katero na podlagi vsaj dveh fotografij dobimo tretjo dimenzijo, se imenuje stereoskopski vid. Z dvema ali več fotografijami istega predmeta, posnetimi z različnih položajev, lahko s pomočjo preseka žarkov določimo katerokoli točko, ki je vidna na obeh fotografijah (Linder in sod., 2016). Za samodejno obdelavo fotografij se uporabljajo algoritmi gostega slikovnega ujemanja ali algoritmi za sočasno določitev zunanjih in notranjih parametrov orientacije ter izračuna oblaka točk (angl. Structure from motion). Za snemanje in rekonstrukcijo večjih površin se

uporabljajo algoritmi slikovnega ujemanja, kjer je potrebno na osnovi kalibracijskih parametrov fotoaparata določiti notranjo in zunanjo orientacijo. Pri obravnavanju manjših območij, kot so stavbe, ulice, kipi ali posamezni predmeti, se uporabljajo algoritmi za sočasno določitev zunanjih in notranjih parametrov orientacije. V tem primeru moramo za zagotavljanje natančnosti imeti veliko število fotografij, posnetih na različnih lokacijah in pod različnimi koti, na katerih so vidne skupne točke (Triglav Čekada, 2016).



Slika 5 Princip delovanje fotogrametrije na osnovi stereoskopskega vida (Linder, 2016)

2.2 Zapis oblakov točk

2.2.1 Formati za zapis oblakov točk

Trenutno je na trgu na voljo veliko različnih formatov in posledično datotečnih tipov za zapis oblakov točk. Različni načini zajema podatkov in strojna oprema tako ustvarjajo mnoge formate zapisa podatkov. Težava se pojavi pri programih za obdelavo, ki so sposobni uvažati in izvažati zgolj ozek izbor formatov. Posledično se proizvajalci in potrošniki podatkovnih sistemov za shranjevanje in izmenjavo podatkov zanašajo na lastne ali ad-hoc formate. Do informacij, shranjenih v lastniških oblikah, je težko dostopati, ad-hoc formati pa povečujejo stroške razvoja programske opreme in jih je težko razširiti na splošno uporabo. Podjetja, kot so Leica, Faro in Trimble, proizvajajo tako programske kot strojne rešitve za zajem in obdelavo podatkov, ki pa so usmerjene v lastne formate za povezavo svoje strojne in programske opreme. Na drugi strani pa podjetja, kot je Autodesk, ne proizvajajo strojne opreme, ampak samo programsko. Ponudba na trgu je velika, medsebojna povezanost pa ne, zato je nujna standardizacija, ki bi spodbujala interoperabilnost podatkov med programskimi rešitvami in strojno opremo za pridobivanje prostorskih podatkov (Huber, 2011).

Pri zapisu oblakov točk v datoteke ločimo binarni in ASCII zapis, nekateri formati pa omogočajo kombinacije obeh načinov. Zapis ASCII tako kot ostali jeziki bazira na binarni osnovi in znake zapisuje s sedmimi biti. V ASCII datoteki je tako oblak točk zapisan z znaki. Vsaka vrstica v zapisu predstavlja eno točko v oblaku (x, y, z prostorsko koordinato), poleg koordinate pa so zapisani še ostali atributi, ki so odvisni od načina zajema in vrste datoteke. Najbolj pogosti atributi so barve (RGB), intenziteta in klasifikacija odboja laserskega žarka. Na spodnji sliki so je prikazan izrez preprostega ASCII zapisa nekaj prostorskih točk. V prvih treh stolpcih so zapisane koordinate x, y in z, v četrtem je intenziteta odboja od površine, v zadnjih treh pa RGB vrednosti.

```
342684
4.246445 -6.426620 -50.214615 -399 66 50 83
-9.318283 -6.014053 -51.129257 -437 132 141 152
-4.043655 -14.000992 -65.560776 -332 69 61 85
-6.693771 -13.897171 -65.007004 -347 74 66 90
-6.574844 -13.897018 -65.017166 -363 78 69 92
-6.551651 -13.896622 -65.014236 -356 80 72 94
-6.482590 -13.906174 -65.046219 -357 76 68 88
-6.388870 -13.905197 -65.035995 -350 73 65 90
-6.329880 -13.904129 -65.019699 -327 78 70 93
-6.369919 -13.915909 -65.089279 -351 74 65 90
.
.
.
```

Slika 6 Primer ASCII zapisa točk z atributoma intenzitete in barve.

Prednost ASCII zapisa je njegova vsestranskost in dostopnost zaradi enostavnega načina zapisa. Taka oblika je enostavna za pregledovanje in urejanje, saj jo lahko odpremo že s programi kot je Notepad. Slabosti takega zapisa so večje datoteke, manjša količina metapodatkov in branje vsake vrstice posebej, kar zmanjša hitrost odpiranja datoteke v programih. Najbolj pogosti formati s takim zapisom so .xyz, .obj, .ptx in .asc. Binarni sistem shranjuje podatke v obliki binarnega zapisa, kjer števila niso predstavljena z znaki ampak zapisom mantise in eksponenta. Take datoteke so veliko bolj kompaktne (4-6 bajtov na število) in vsebujejo več informacij. Najbolj pogosti formati s takim zapisom so .fls, .pcd, .las. Tretja vrsta so formati, ki so zapisani kot kombinacija obeh načinov. Najbolj pogosti formati za kombiniran zapis so .ply, .fbx in .e57.

Glavni formati za zapis oblakov točk:

.obj Je preprost odprtokodni format za zapis geometrije, ki se lahko uporablja tudi za zapis oblakov točk. Točke se zapišejo s primitivnimi geometrijskimi oblikami (sfere). Vsebuje zapis oglišč, ogliščnih normal, projekcij, tekstur in poligonov na podlagi točk. Oglišča so v datoteki zapisana v nasprotni smeri urinega kazalca. Koordinate nimajo enot, lahko pa vsebujejo podatke o merilu.

.ply Format datotek je bil razvit v grafičnem laboratoriju Stanford za enostavno shranjevanje 3D grafik. Osnovni elementi v datoteki so oglišča, ki so lahko tudi povezana v trikotnike, s katerimi zapišemo površine. Glavne lastnosti elementov so njihove koordinate, poleg koordinat so lahko v zapisu vključene barve, normale ploskev, gostota, prosojnost in texture. V glavi datoteke so zapisani elementi in njihove lastnosti, v preostalem delu datoteke so številčni podatki. Format je opcijsko lahko zapisan binarno ali v ASCII zapisu.

.las Ta binarni format je razvilo Ameriško združenje za fotogrametrijo in daljinsko zaznavanje (ASPRS). Format je bil predviden za zapis podatkov, pridobljenih z Lidarjem, za kar trenutno tudi uporablja, omogoča pa zapis poljubnih oblakov točk. V formatu so zapisane koordinate, klasifikacija, ki je značilna za Lidar, intenziteta odboja laserskega žarka, RGB vrednost, številka zaporednega odboja in kot zajema. Kljub binarnemu zapisu, ki optimizira velikost datotek, je bil zaradi velike količine točk v datotekah razvit kompresijski format .laz. Ta format je priljubljen pri objavljanju in deljenju podatkov, saj so datoteke, zapisane v tem formatu, nekajkrat manjše od izvirnega formata. Težava tega formata je počasnejše odpiranje v programih, ker je potrebno podatke pred uporabo najprej dekompresirati v izvorni format.

.xyz Je nestandardizirana ASCII oblika zapisa oblakov točk v kartezičnem koordinatnem sistemu. Ravno zaradi pomanjkanja standardizacije ta format ne moremo obravnavati kot samostojen. Lahko ga uvrstimo v bolj splošno skupino ASCII oblakov točk, v katero spadajo .asc, .txt in .pts formati. Točke v tem formatu so zapisane v posameznih vrsticah. Poleg koordinat so v vrsticah druge lastnosti, ki jih specifikira avtor zapisa, so to lahko barve in intenziteta. Čeprav je združljivost tega tipa datoteke s programi velika zaradi pomanjkanja standardizacije to ni najbolj primerna oblika zapisa in prenosa podatkov.

.e57 Je standardiziran format za shranjevanje oblakov točk. Poleg shranjevanja točk se lahko uporablja za shranjevanje slik in metapodatkov o laserskih skenerjih ali drugih načinih zajema. Kot kombinacija binarnega in ASCII zapisa je format kompakten, hiter in vedno bolj razširjen. V datoteki so shranjene točke normale, barve in intenziteta. Čeprav je format predviden kot standarden, so bile knjižnice formata ustvarjene samo za Windows. Mac in Linux sta bila v začetnem razvoju izpuščena.

Seznam formatov, s katerimi lahko zapišemo oblake točk, je daljši od zgoraj opisanih. V preglednici spodaj strani so prikazni še ostali formati, ki se uporabljajo za shranjevanje oblakov točk. V preglednici so navedeni: format, njegova končnica, tip zapisa in število posameznih oblakov točk v datoteki. Število oblakov točk dejansko predstavlja število potencialnih stojišč laserskega skenerja. Ta stojišča so v datoteki medsebojno povezana v zaključeno celoto.

Preglednica 1 Seznam formatov za zapis oblakov točk (povzeto po cloudcompare.org)

Format	Končnica	Binarni/ASCII zapis	Št. oblakov točk v datoteki
BIN	.bin	Binarni	>1
SBF	.sbf	Binarni	1
ASCII	.asc,.txt,.xyz,.neu,.pts	ASCII	1
LAS	.las	Binarni	1
E57	.e57	Mešan zapis	>1
PTX	.ptx	ASCII	>1
FARO	.fls, *.fws	Binarni	>1
DP	.dp	Binarni	>1
PCD	.pcd	Binarni	>1
PLY	.ply	ASCII ali binarni	1
OBJ	.obj	ASCII	1
VTK	.vtk	ASCII	1
STL	.stl	ASCII	1
OFF	.off	ASCII	1
FBX	.fbx	ASCII ali binarni	1
OUT (Bundler)	.out	ASCII	1
PV	.pv	Binarni	1
PN	.pn	Binarni	1
SOI	.soi	ASCII	>1
POV	.pov	ASCII ali binarni	>1
ICM	.icm	ASCII ali binarni	1
Geo-Mascaret	.georef	ASCII	1

2.2.2 Interoperabilnost različnih formatov

Večina izmenjav 3D podatkov danes poteka z uporabo enega izmed treh tipov datotek: lastniški, odprti in ad-hoc formati. Dokumentacije odprtih formatov so prosto dostopne vsakomur. Lastniški formati pa dolgoročno niso učinkovit pristop k izmenjavi podatkov. Vsak proizvajalec strojne in programske opreme za svoje podatke običajno določi enega ali več lastniških formatov. Lastniški formati navadno ostanejo v domeni proizvajalcev strojne in programske opreme za ostale razvijalce pa so skrivnost. Takšen način lastninjenja formatov ustvarja ozko grlo pri deljenju datotek, kjer določeni programi omogočajo delo z omejenim številom specifičnih formatov. Pri prevedbi med formati se občasno informacije izgubijo, njihova natančnost se zmanjša ali pa se datoteke poškodujejo zaradi programskih napak in napačnega komuniciranja med zapisovalcem in bralcem podatkov. V primerih, ko neposreden prenos podatkov med programi ni mogoč, je potrebna pretvorba v niz vmesnih (posrednih) formatov. Takšna rešitev je dolgotrajna in nagnjena k napakam, s tem pa obstaja več možnosti za izgubo podatkov kot pri neposredni uporabi. Nenazadnje, lastniški formati niso vedno na voljo. Če so lastniški formati slučajno dostopni, pa morda niso dobro dokumentirani ali nimajo knjižnic programske opreme za branje in pisanje (Huber, 2011).



Slika 7 Oblak točk shranjen v formatu .e57 (Huber, 2011)

Format E57 poskuša odpraviti te težave z zagotavljanjem nevtralne oblike, ki bi v idealnem primeru odpravila potrebo po uvozu in izvozu številnih lastniških formatov. V nekaterih primerih programerji ustvarijo svoje ad-hoc formate za izmenjavo 3D podatkov. Takšni formati so pogosto omejeni na ASCII zapis s takim zapisom pa niso časovno in prostorsko učinkoviti. Uradna dokumentacija ad-hoc formatov je običajno omejena ali pa celo ne obstaja, kar dodatno povečuje možnosti za napake in izgubo podatkov. Ti formati običajno ne dosežejo široke uporabe, na koncu pa privedejo do dodatne zmede in odvečnega dela. Format E57 zmanjšuje potrebo po lastniških in ad-hoc formatih, saj so programske knjižnice prosto dostopne (Huber, 2011).

Oblika datoteke LAS je namenjena shranjevanju podatkov oblakov točk, ki jo je razvil ASPRS (angl. American Society for Photogrammetry and Remote Sensing). Prilagojen je potrebam aeroskeniranja (ALS), čeprav lahko format uporabimo za podatke terestričnega laserskega skenerja z ignoriranjem neuporabljenih polj. Format E57 naj bi bil splošnejši format, primeren za shranjevanje podatkov iz različnih virov. Na področjih, kjer se formata E57 in LAS prekrivata, pa imata različne rešitve za iste probleme. Format E57 vključuje tudi nekatere dodatne funkcije, ki niso na voljo v trenutni obliki LAS.

Format E57 podpira več koordinatnih sistemov (vključno s kartezičnimi in sferičnimi), zapis fotografij, vgrajeno zaznavanje napak in razvrščanje točk v vrstice, stolpce ali uporabniško definirane skupine. Format E57 določa tudi mehanizem razširitve, ki uporabnikom omogoča, da razvijajo razširitve po meri, ki niso bile predvidene v prvotni zasnovi standarda. Poleg tega pa ima skoraj neomejeno velikost datoteke in števila zapisov – 18 eksabajtov (Huber, 2011).

2.2.3 Začetne zahteve za razvoj formata .e57

Zadnja leta skupina prostovoljcev deluje v okviru odbora ASTM E57 za sisteme prostorskih podatkov, kjer razvijajo odprt standard za izmenjavo podatkov oblakov točk. Pododbor E57.04 za interoperabilnost podatkov je del mednarodne organizacije za standardizacijo ASTM International. Ta odbor vključuje predstavnike številnih največjih proizvajalcev sistemov za lasersko skeniranje, proizvajalce programske opreme za obdelavo oblakov točk, ponudnike storitev laserskega skeniranja, industrijske svetovalce in akademske raziskovalce. Rezultat prizadevanj tega odbora je format E57 za izmenjavo podatkov oblakov točk. Format E57 je zasnovan za splošni namen shranjevanja podatkov, ki jih proizvajajo različni sistemi za zajem prostorskih podatkov. Datoteka lahko zapiše podatke oblakov točk iz laserskih skenerjev, fotogrametričnih sistemov in ostalih načinov zajema. Poleg shranjevanja meritev 3D točk lahko v datoteko zapišemo tudi 2D fotografije, pa tudi osnovne metapodatke, povezane z 2D slikami in 3D točkami (Huber, 2011).

Zahteve za obliko zapisa datotek E57 so razvili s soglasjem sodelujočih predstavnikov industrije (razvijalci strojne in programske opreme). Na začetku so bile določene glavne značilnosti, ki bi omogočile globalno sprejetje in uporabo standarda. Kasneje bi sledili še dodatni elementi, ki bi izboljšali zmogljivost in uporabnost. Standard bi se osredotočal na shranjevanje 3D podatkov in na zanesljivo izmenjavo podatkov med različnimi udeleženci.

Zahteve po standardu so organizirane okoli petih vodilnih načel:

- Zanesljiva interoperabilnost - Podatki morajo biti prenosljivi med različnimi udeleženci v procesu dela z oblakom točk.
- Odprtost - Standard in njegovo izvajanje morata biti javno dostopna, nevtralna in dobro dokumentirana brez kakršnih koli omejitev do uporabnikov.
- Minimalne ovire za razvijalce - Razvojni stroški za razvijalce, ki bi implementirali format v svoje programe, morajo biti minimalni.
- Minimalistična zasnova - Oblikovanje mora biti čim bolj preprosto, hkrati pa mora dosegati zahtevane cilje.
- Razširljivost - Zasnova mora biti razširljiva, da bi se v prihodnosti lahko dodale nove zmogljivosti. Pri tem se ne sme zmanjšati funkcionalnost.

2.2.4 Struktura formata .e57

Na globalni ravni temelji struktura zapisa na podatkovnem formatu XML. Zapis v čisti obliki XML bi neučinkovito shranjeval velike nabore podatkov, zajete z laserskimi skenerji. Datoteke bi bile nesprejemljivo velike, uvoz in izvoz pa bi bila počasna. Na nizki ravni zapisa so dejanski podatki o točkah predstavljeni s stisnjeno binarno obliko. Drugi veliki bloki podatkov, kot so slike, so prav tako učinkovito zapisani v binarni obliki. Takšna oblika zapisa podpira fleksibilnost in razširljivost z uporabo besedilnega XML, hkrati pa omogoča učinkovit uvoz in izvoz. Datoteka E57 je razdeljena na tri dele: glavo, niz izbirnih binarnih odsekov in odsek XML. Glava je majhna, 48-bajtna binarna struktura, ki vsebuje ključne podatke o datoteki, kot sta številka različice in lokacija odseka XML. Odsek XML je

zapisan v hierarhični drevesni strukturi. Če datoteka vsebuje točkovne podatke ali slike, se na te dele hierarhije sklicuje odsek XML, dejanski podatki pa se shranijo v binarnih razdelkih z ločenimi referencami za vsak niz točk ali slik.

Preglednica 2 Tabelarni prikaz strukture zapisa znotraj formata .e57 (Huber, 2011)

Glava
Binarni odsek (točke)
Binarni odsek (točke)
...
Binarni odsek (slike)
XML odsek

Oblika formata E57 omogoča njegovo razširitev z novimi zmogljivostmi, hkrati pa ohranja združljivost s predhodnimi različicami. Če uporabniška skupnost razširitev dobro sprejme, bo prihodnja revizija vključila to razširitev v nabor glavnih funkcij. Pomembno je, da nove funkcije ne preprečujejo branja in uporabi datotek v starejših različicah. Razširitve so definirane s pomočjo imenskih prostorov XML, ki enolično identificirajo vsa nova imena elementov, ki so definirana v novi razširitvi. Ideja je, da ko bralnik datotek E57 naleti na element z imenskim prostorom, ki ga ne razume, ta element preprosto ignorira. Na ta način lahko starejši programi lahko berejo in obdelujejo datoteke E57, izpustijo pa nove elemente, ki jih določajo nove razširitve.

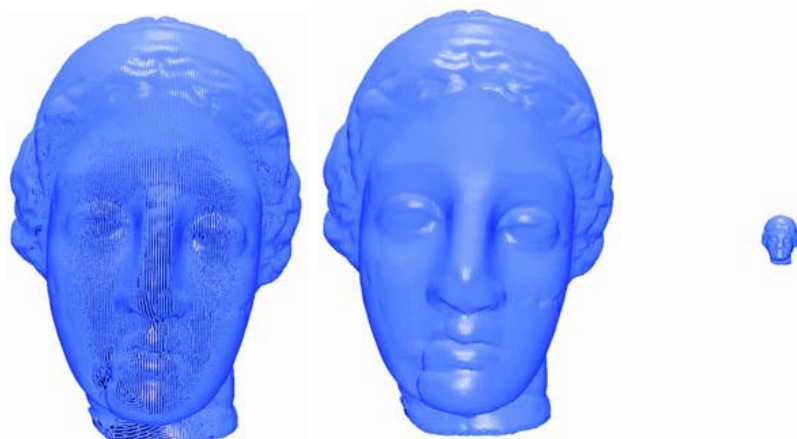
```
<?xml version="1.0" encoding="UTF-8"?>
<e57Root type="Structure" xmlns="http://www.astm.org/COMMIT/E57/2010-e57-v1.0">
  <formatName type="String"><![CDATA[ASTM E57 3D Imaging Data File]]></formatName>
  <guid type="String"><![CDATA[{56D8F874-3656-4CA8-BA17-88F9757063D0}]]></guid>
  <versionMajor type="Integer">1</versionMajor>
  <versionMinor type="Integer"/>
  <e57LibraryVersion type="String"><![CDATA[InteliSum-LD3-Studio-V5.1-E57RefImpl-0.3.173-x86-windows]]></e57LibraryVersion>
  <coordinateMetadata type="String"/>
  <creationDateTime type="Structure">
    <dateTimeValue type="Float">9.6758299097398019e+008</dateTimeValue>
    <isAtomicClockReferenced type="Integer"/>
  </creationDateTime>
  <data3D type="Vector" allowHeterogeneousChildren="1">
    <vectorChild type="Structure">
      <guid type="String"><![CDATA[{D2629847-A1DB-4579-9E86-5D146B688EA5}]]></guid>
      <name type="String"><![CDATA[sp412]]></name>
      <description type="String"/>
      <originalGuids type="Vector" allowHeterogeneousChildren="0">
        <vectorChild type="String"><![CDATA[{D2629847-A1DB-4579-9E86-5D146B688EA5}]]></vectorChild>
      </originalGuids>
      <sensorVendor type="String"><![CDATA[Z+F]]></sensorVendor>
      <sensorModel type="String"><![CDATA[I5006]]></sensorModel>
      <sensorSerialNumber type="String"><![CDATA[749]]></sensorSerialNumber>
      <sensorHardwareVersion type="String"/>
      <sensorSoftwareVersion type="String"><![CDATA[4.4.0.452]]></sensorSoftwareVersion>
      <sensorFirmwareVersion type="String"><![CDATA[7.3.2.4067-534]]></sensorFirmwareVersion>
      <temperature type="Float"/>
      <relativeHumidity type="Float"/>
      <atmosphericPressure type="Float"/>
      <indexBounds type="Structure">
        <rowMinimum type="Integer"/>
        <rowMaximum type="Integer">543</rowMaximum>
        <columnMinimum type="Integer"/>
        <columnMaximum type="Integer">2473</columnMaximum>
        <returnMinimum type="Integer"/>
        <returnMaximum type="Integer">1</returnMaximum>
      </indexBounds>
      <intensityLimits type="Structure">
        <intensityMaximum type="ScaledInteger" minimum="0" maximum="32767" scale="3.0518509475997192e-005">32767</intensityMaximum>
        <intensityMinimum type="ScaledInteger" minimum="0" maximum="32767" scale="3.0518509475997192e-005">0</intensityMinimum>
      </intensityLimits>
      <colorLimits type="Structure">
        <colorRedMaximum type="Integer">255</colorRedMaximum>
        <colorRedMinimum type="Integer">0</colorRedMinimum>
        <colorGreenMaximum type="Integer">255</colorGreenMaximum>
        <colorGreenMinimum type="Integer">0</colorGreenMinimum>
        <colorBlueMaximum type="Integer">255</colorBlueMaximum>
        <colorBlueMinimum type="Integer">0</colorBlueMinimum>
      </colorLimits>
    </vectorChild>
  </data3D>
</e57Root>
```

Slika 8 Prikaz izseka XML dela zapisa znotraj formata .e57 (povzeto po libe57.org)

2.3 Vizualizacija in organizacija oblakov točk

2.3.1 Organizacija oblakov točk

Z laserskim skenerjem pridobljen oblak točk je le redko končni izdelek, saj v večini primerov predstavlja le vmesni korak celotnega procesa. Poleg tega lahko že na začetku ocenimo kakovost posnetka in vsebino, hkrati pa preverimo, če kaj manjka pred nadaljevanjem procesa. Za boljšo kontrolo in iskanje pomanjkljivosti je mogoče uporabiti različne tehnike, kot sta barvanje ali senčenje oblaka. Čeprav je vizualizacija oblakov točk uporabna za številne namene, večina projektov zahteva nadaljnjo obdelavo za pridobivanje dodatnih informacij. Te dodatne informacije pa so odvisne od vrste nadaljnje uporabe. Za lažje delo in nadaljevanje je oblak točk praktično urediti v smiselno strukturo, saj so oblaki točk v večini primerov prikazani kot dolgi sezname x, y in z koordinat, v nekaterih primerih pa so v datotekah zapisani tudi dodani atributi. Čeprav so točke običajno zapisane v vrstnem redu njihovega zajema, nam to ne omogoča, da hitro določimo vse sosednje točke znotraj poljubnega radija. Za izdelavo takšnih poizvedb je potrebno podatke zapisati v strukture. Točkam je smiselno tudi določiti normale ali pa jih na podlagi okoliških točk povezati v Delunayeve trikotnike. Večje količine je smiselno strukturirati v octree in k-D strukture. Ko je podatkovna struktura pripravljena, je lažje določiti sosednje točke, ki pripadajo isti ploskvi. Strukture, pridobljene s takšno obdelavo, se lahko uporabljajo za številne namene, kot so združevanje več oblakov skupaj, segmentiranje točk in rekonstrukcije stavb. Dandanes standardni grafični vmesniki, kot sta OpenGL ali DirectX, omogočajo prikaz točk kot samostojnih enot (primitivov). Poleg koordinat pa omogočajo tudi določitev velikosti, barv in normal točk, ni pa mogoče določiti tekstur. Odvisno od grafične strojne opreme, je točka lahko upodobljena kot štirikotnik ali krogla v prostoru. Zaradi strojne podpore je upodabljanje oblaka točk hitro in enostavno, s čimer je trenutno mogoče prikazati več deset milijonov točk na sekundo (Vosselman in Maas, 2011). Na spodnji sliki je prikazana primerjava velikosti točk in različne oddaljenosti med opazovalcem in objektom.



Slika 9 Težave pri prikazu oblakov točk. Na prvi sliki je opazovalec blizu objekta, velikost točk je majhna, zato se pojavijo praznine. Na drugi sliki so točke večje, prikaz pa je zato manj natančen. Na tretji sliki je opazovalec daleč, vidni pa so samo glavni detajli (Vosselman in Maas, 2011)

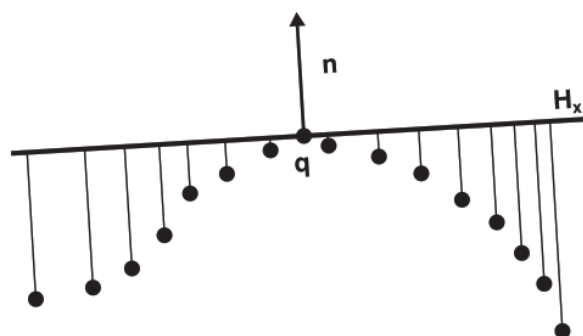
2.3.2 Normale točk

Ker tehnike skeniranja ne določajo normal točkam, je treba za vsako točko oceniti normale v postprocesiranju surovega oblaka točk. Določanje normal točkam je osnovna naloga v postopku obdelave oblakov točk. V preteklih letih je bilo za to postopek razvitih kar nekaj metod (Zhao in sod., 2019).

Metode za določitev normal točk:

- Metode, ki temeljijo na regresiji,
- metode, ki temeljijo na Voronijevih območjih,
- metode, ki temeljijo na globokem učenju.

Regresijska metoda deluje ob predpostavki, da je površina predmeta povsod enakomerna. Pri izračunu se upošteva lokalna okolica računske točke, za katero se izračunava normala. Okrog računske točke se skozi njene okoliške točke lokalno napne ravnina. Z analizo glavnih komponent (PCA) se izračunana normala napete ravnine upošteva kot normala računske točke. Ker je metoda enostavna in hitra, se pogosto uporablja za obdelavo oblakov točk. Pomanjkljivosti se pokažejo pri objektih z ostrimi robovi. Na robove je težko napeti kakršnekoli ravnine, vrednosti analize glavnih komponent so posledično nizke, rezultati pa zato slabi. Poleg tega je metoda občutljiva na šum in odvečne točke (angl. outliers), ki jih običajno vsebuje vsak oblak točk. Za izboljšanje metode in povečanje robustnosti so bile razvite nekatere metode (Zhao in sod., 2019). Na osnovi Houghove transformacije in enotne strategije vzorčenja je mogoče zmanjšati vpliv neenakomerne gostote podatkov (Boulch in sod., 2012). Za doseganje optimalnih rezultatov takšne tehnike pogostokrat zahtevajo ročno izbiro najbolj primernih parametrov, z zamudnim postopkom poskušanja in popravljanja. Poleg tega so raziskovalci v zadnjih letih začeli ocenjevati normale za nestrukturirane oblake točk z uporabo tehnik globokega učenja (Boulch in Marlet, 2016).

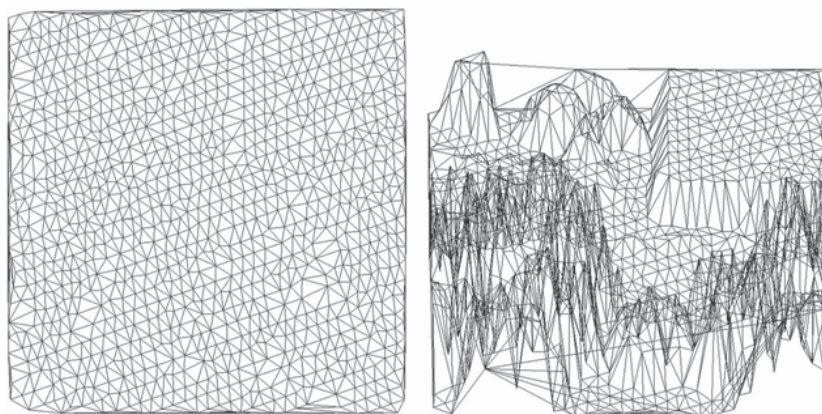


Slika 10 Grafični prikaz prilagoditve normale n na točko q . Z metodo najmanjših kvadratov prilagodimo ploskev H na k okoliških točk (Vosselman in Maas, 2011)

2.3.3 Delaunayeva triangulacija

Metoda Delaunayeve triangulacije poveže skupino točk v skupino trikotnikov, kjer obravnavane točke predstavljajo vozlišča trikotnikov. Metodo je leta 1934 razvil ruski matematik Boris Delaunay. Metoda je ena bolj preprostih in ima to lastnost, da se znotraj kroga, očrtanega kateremukoli trikotniku, ne nahaja nobena druga točka. S to metodo so vse točke povezane v kompaktne trikotnike z najmanjšimi notranjimi koti. Trikotniki so si zaradi tega podobni, med njimi pa ni prevelikih odstopanj. Prve sosednje točke z neposredno povezane v trikotnike. Večina trikotnikov je enakostraničnih s kratkimi robovi, le v bližini konveksne ovojnice in na robovih se lahko pojavijo podolgovati, enakokraki trikotniki z dolgimi robovi. Za triangulacijo so bili razviti številni algoritmi, kot so algoritem obračanja, inkrementalni algoritem, algoritem deli in vladaj, algoritem s prebirno premico (angl. sweep line) in algoritem preverjanja točk. Intuitivno najpreprostejši je algoritem obračanja, ki se opira na pogoj, da znotraj kroga,

očrtanega kateremukoli trikotniku, ni četrte točke. V takšni situaciji zamenja skupni stranici paru trikotnikov, s tem pa ustvari nov, drugačen par trikotnikov, kjer je pogoj očrtane krožnice izpolnjen. Optimalna triangulacija se izvede v navidezni ravnini, kjer so točke medsebojno blizu (primer zemeljskega površja). Višinske razlike točk niso težava, dokler višinska razlika ni neprimerno večja od medsebojne oddaljenosti ostalih točk. Primer neugodne višinske triangulacije je prikazan na sliki 8. Podobna težava se pojavi tudi pri hkratni obdelavi površin, ki se nahajajo ena nad drugo. Takrat se točke, ki pripadajo posamezni površini, ne povežejo samo medsebojno, ampak tudi s točkami drugih ravnin. Taka triangulacija izračuna dolge enakokrake trikotnike, ki ne prikazujejo pravega stanja (Vosselman in sod., 2010).



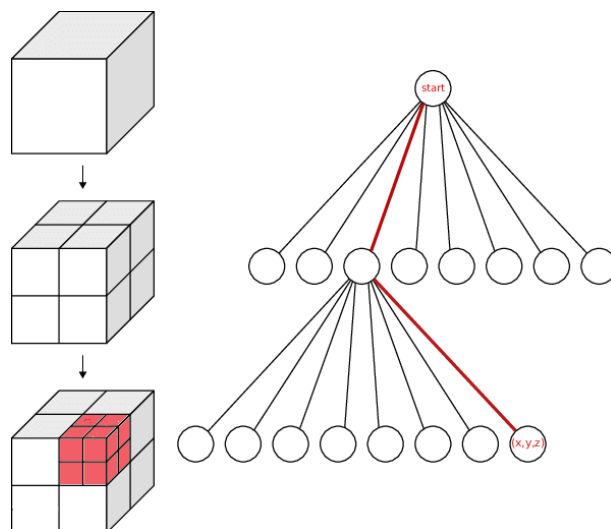
Slika 11 Pogled na trianguliran teren z vrha (levo) in s strani (desno) (Vosselman in Maas, 2011)

Na tak način lahko pripravimo triangulirano mrežo ali TIN (angl. triangulated irregular network), s katero lahko vektorsko prikažemo neprekinjene površine, ki so v celoti sestavljene iz trikotnikov. Model TIN je bil razvit v zgodnjih sedemdesetih letih kot preprost način prikaza površin iz niza nepravilno razporejenih točk. Digitalne podatkovne strukture TIN se uporabljajo v različnih aplikacijah, vključno z geografskimi informacijskimi sistemi (GIS) in računalniško podprto grafiko (CAD) za vizualizacijo površin. Tako kot oblak točk TIN ne predstavlja vedno končnega produkta in ga lahko dodatno obdelujemo.

2.3.4 Octree podatkovna struktura

Octree strukture so učinkovite drevesne strukture za delo z velikimi količinami podatkov. Poleg prostorskega indeksiranja jih v okviru obdelave oblakov točk lahko uporabljamo za kompresiranje. S kompresiranjem spreminjamo stopnjo podrobnosti (angl. level of detail) in odstranjujemo objektov izven polja opazovanja (angl. viewing frustum). Octree predstavljajo analogijo 2D quadtree strukture v 3D prostoru. Vsako vozlišče drevesne strukture predstavlja samostojen kubični prostorski volumen, ki ga imenujemo celica. Začenši z največjo celico, ki zajema vse točke znotraj obravnavanega prostora, se vsaka naslednja celica, ki še vsebuje točke, razdeli na največ naslednjih osem celic z razpolovitvijo celic v vsaki od treh dimenzij. Če v nekem območju ni točk, se tam celice ne delijo naprej in ostanejo prazne. Na ta način sta lokacija in nivo posameznih celic enolično določena. Delitev celic v octree strukturi je podobna binarnemu drevesu, kjer se strukture namesto dvakrat dveh opcij (levo-desno in spredaj-zadaj) delijo na osem delov (levo-desno, spredaj-zadaj in zgoraj-spodaj). Octree struktura je zaključena, ko

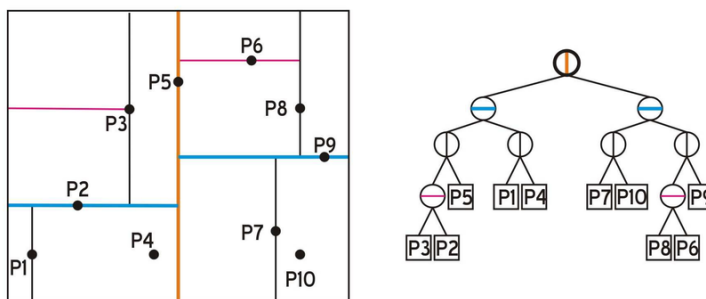
celice na zadnjem nivoju vsebujejo malo točk (Vosselman in sod., 2010). Na spodnji sliki je prikazana rekurzivna delitev kocke v dveh nivojih na oktante in ustrezna octree struktura.



Slika 12 Rekurzivna razdelitev kocke na oktante (levo) in ustrezna octree drevesna struktura (desno) (povzeto po Funk in Borner, 2016)

2.3.5 k-D tree podatkovna struktura

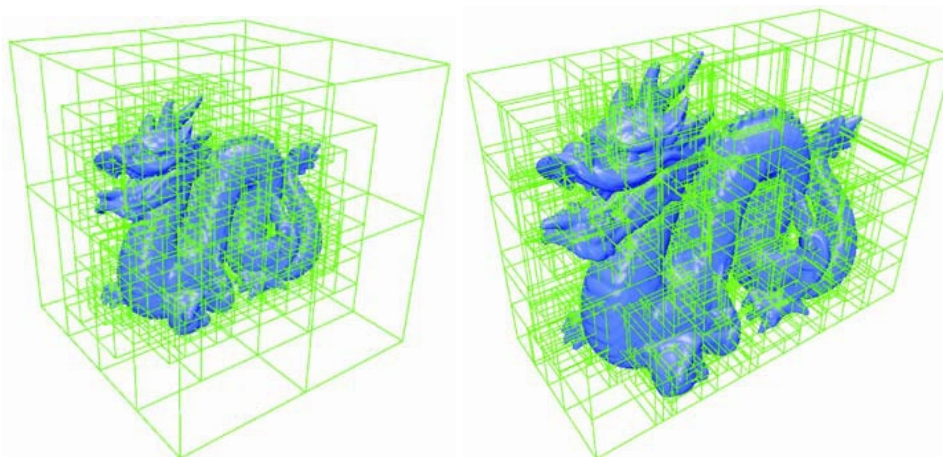
Za razliko od octree strukture zagotavlja k-D drevesna struktura bolj uravnoteženo hierarhično strukturiranje podatkov. Ime k-D predstavlja število dimenzij, na katere se delijo posamezne celice. Struktura k-D je kompaktna prostorska podatkovna struktura, ki je poleg funkcij, ki jih omogoča octree struktura, primerna tudi kot optimizacijski algoritem iskanja najbližjih sosedov. Vsako vozlišče predstavlja celico ene izmed dimenzij, ki se naknadno delita na novi podvozlišči. Vse točke v levem podvozlišču se nahajajo v prvi celici, točke v desnem podvozlišču pa v drugi. V tej se ne shranjuje deliteljev podvozlišč, zato je podatkovno varčnejša. V k-D strukturi je korenski element 1. V vsakem naslednjem elementu i je levi podelument $2i$, desni pa $2i+1$. Izpuščanje deliteljev prihrani 8 bajtov na element, kar pri velikem številu delitev prihrani kar nekaj prostora. Izdelava uravnoteženega k-D drevesa se začne z celico, ki zajema vse točke in rekurzivno deli trenutno celico vzdolž ene dimenzije na dve celici.



Slika 13 Rekurzivna delitev kvadrata (levo) in ustrezna k-D tree drevesna struktura (desno) (povzeto po Teniente in Cetto, 2008)

V 3D oblaku točk je začetna os delitve x , druga delitev je v smeri osi y , naslednja v smeri z . Po delitvi v smeri z se postopek vrne na delitev v smeri x in nadaljuje v prej opisanem zaporedju. Postopek se ponavlja, dokler se ne zapolnijo vse particije. Namesto večkratnega kroženja po posameznih dimenzijah se pri k-D dosežejo boljši rezultati ravno z delitvijo prostorske dimenzije, kjer so podatki najbolj

razpršeni. Vsaka izmed particij vsebuje enako število elementov, določenih na podlagi srednje vrednosti. Na ta način doseže k-D struktura boljšo stopnjo zapolnjenosti z uporabo manjšega števila particij kot octree struktura (Vosselman in sod., 2010). Na sliki 12 je na modelu prikazana medsebojna primerjava prostorskih struktur octree in k-D tree.



Slika 14 Model razdeljen v octree strukturi (levo) in k-D tree strukturi (desno) (Vosselman in Maas, 2011)

2.4 Segmentacija in klasifikacija oblakov točk

2.4.1 Segmentacija oblakov točk

Oblaki točk se pogosto uporabljajo izključno za vizualizacijo ali preproste meritve, kot so merjenje dolžin, kotov in površin. Z oblaki točk lahko zajamemo in segmentiramo dolge cestne odseke (Clode in sod., 2004; Boyko in Funkhouser, 2011), forenzično pregledujemo nesreče (Laefer in Pradhan, 2006) in zajamemo stavbe kulturne dediščine (Pu in Vosselman, 2009). Merjene stavbe so večinoma velike in kompleksne, zato je potrebno pridobljene podatke smiselno segmentirati. Segmentacija je v procesu dela z oblaki točk pomembna naloga, saj so nabori podatkov oblakov točk pogostokrat nestrukturirani in ogromni. Številni segmentacijski algoritmi zahtevajo veliko ročnih nastavitev začetnih parametrov, zaradi česar so postopki segmentacije velikokrat dolgotrajni. Hitrost izračuna je tu ključna, pri hitrejših algoritmih lahko uporabniki preizkusijo več različnih začetnih parametrov v primerjavi s počasnejšimi, s tem pa dosežejo bolj pravilne rezultate.

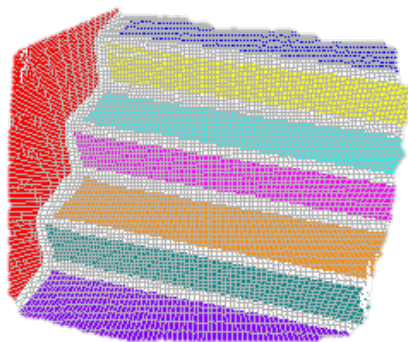
2.4.2 Segmentacija na podlagi zaznavanja robov

Segmentacija na podlagi zaznavanja robov (angl. edge based) je skupina matematičnih algoritmov, katerih cilj je prepoznavanje robov v digitalni sliki. Algoritmi določajo robove v sliki na podlagi sprememb svetlosti in barv. Točke, na katerih se svetlost ali barva spremenita, so običajno organizirane v niz medsebojno povezanih točkovnih segmentov - robov. Segmentacija na podlagi zaznavanja robov je starejša metoda, prenesena neposredno iz 2D rastrskih slik v 3D oblake točk, ki se je uporabljala v zgodnjih fazah segmentacije. Podobno kot pri digitalnih slikah so robovi v oblakih točk določeni na podlagi sprememb lastnosti sosednjih točk. Med te lastnosti spadajo barve, gradienti, intenziteta odboja in normale točk. Algoritem zaznavanja robov v oblakih točk se deli na dve glavni stopnji; zaznavanje robov različnih regij točk in združevanje točk znotraj teh robov (Rabani in sod., 2006) in algoritme, ki v primerih z malo šuma in enakomerno gostoto omogočajo hitro segmentacijo. Poleg tega v 3D prostoru

takšne metode pogosto ustvarijo prekinjene robove in jih zato brez dodajanja polnila ali dodatnih interakcij ne moremo neposredno uporabljati za identifikacijo zaprtih segmentov (Castillo in sod., 2013, Grilli in sod., 2017).

2.4.3 Segmentacija na podlagi rasti regij

Metode rasti regij (angl. region growing) se začnejo z izbiro ene ali več začetnih točk (angl. seed points) s specifičnimi lastnostmi. Na podlagi kriterijev, ki upoštevajo lastnosti začetnih točk, se med seboj povezujejo točke, ki so medsebojno blizu in so znotraj začetnih kriterijev. Kjer so točke med seboj oddaljene preveč ali pa njihove lastnosti preveč odstopajo, se regija zaključi. Tako kot segmentacija na podlagi zaznavanja robov tudi metoda rasti regij izhaja iz analize digitalnih slik. Prvi korak algoritma rasti regij je prepoznavanje začetnih točk na podlagi začetnih parametrov. Prvotni algoritem sta predstavila Besl in Jain leta 1988, skozi leta pa je bilo s strani drugih raziskovalcev predstavljenih več različic in izboljšav. Za pravilno delovanje algoritma so ključni trije dejavniki: kriteriji podobnosti točk, korak rasti regij in izbiro začetnih točk. Med kriterije podobnosti spadajo geometrijske značilnosti, kot so razdalja med točkami in njihove normale. (Ning in sod., Dong in sod.). Poleg geometrijskih lastnosti lahko kot kriterij uporabimo tudi barve točk (Vosselman in sod., 2004). Kot enote rasti lahko uporabimo posamezne točke, vokselske mreže ali kombinacije obeh. Za povečanje učinkovitosti je smiselno nadomestiti točk z vokseli v mrežni strukturi. Število vokselov je manjše od števila točk v oblaku, zato je proces hitrejši (Deschaut in sod. 2010). Natančnost teh algoritmov temelji na pravilno določenih kriterijih rasti in lokaciji začetnih točk, ki jih je treba vnaprej določiti in prilagoditi različnim naborom podatkov. Algoritmi niso pretirano robustni in so občutljivi na šume v oblakih točk (Woo in sod., 2002, Teboul in sod 2010). Poleg tega so algoritmi računsko zahtevni, z zmanjšanjem števila točk pa vodijo v kompromis med natančnostjo in učinkovitostjo (Woo in sod., 2002, Boulaassal in sod., 2007).



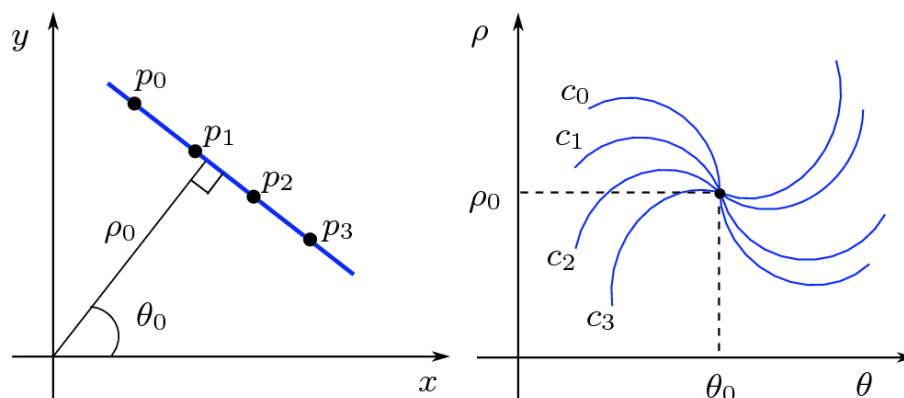
Slika 15 Segmentacija stopnic z algoritmom rasti regij. Vsaka regija je v drugi barvi (Rabbani, 2006)

2.4.4 Klasifikacijske metode s prileganjem elementov

Metode klasifikacije s prileganjem elementov slonijo na ideji, da veliko u predmetov v naravi temelji na preprostih geometrijskih elementih, kot so krogle, valji in plošče. Geometrijske oblike se prilegajo podatkom v oblaku točk, ki najbolj ustrezajo njihovem matematičnemu zapisu. V okviru te kategorije sta v uporabi Houghova transformacija (ang. Hough transformation - HT) (Ballard, 1981) in metoda naključnega vzorčenja (angl. random sample consensus - RANSAC) (Fischler, Bolles, 1981).

2.4.4.1 Houghova transformacija

Metoda Houghove transformacije je bila v osnovi razvita za zaznavanje črt v 2D slikah (Hough, 1962). in je sestavljena iz treh glavnih korakov. Prvi korak je preslikava vzorcev (npr. pikslov ali prostorskih točk) iz prvotnega prostora v diskretni parametrični prostor. Temu sledi akumuliranje rezultatov v parametričnem prostoru v obliki celic. Zadnji korak je izbira celice z lokalno največjo oceno, ki vsebuje največ akumuliranih rezultatov, katerih koordinate se v parametričnem prostoru uporabljajo za predstavljanje geometrijskih segmentov iz prvotnega prostora. Osnovna različica je generalizirana Houghova transformacija (angl. Generalized Hough transform - GHT) (Duda in Hart, 1972). Za poenostavitev računanja in izogib nekaterim računskih težavam se pri GHT uporablja polarni zapis premice namesto eksplcitne oblike. GHT temelji na enačbi $r = x \sin \theta + y \cos \theta$, kjer sta x in y koordinati posameznega vzorčnega piksla, r pa je pravokotna razdalja med izhodiščem koordinatnega sistema in premico skozi izbrano točko, kot θ pa je kot med normalo premice in osjo x . Parametrizacija kota in polmera se lahko razširi tudi v 3D prostor in se lahko uporablja pri zaznavanju prostorskih lastnosti in segmentaciji geometrijskih struktur v oblakih točk. V primerjavi z dvodimenzionalno enačbo je v 3D prostoru dodan še parameter φ , zaradi česar se enačba spremeni $r = x \cos(\theta) \sin(\varphi) + y \sin(\theta) \sin(\varphi) + z \cos(\varphi)$.



Slika 16 Prikaz zapisa točk v kartezičnem koordinatnem sistemu (levo) in parametričnem (desno) (Bergues in sod, 2019)

Pomanjkljivost metode GHT je brezmejnost parametričnega prostora, kar vodi do velike porabe pomnilnika in dolgega računskega časa (Kaiser in sod., 2019). Skozi leta je bilo izvedenih kar nekaj raziskav za izboljšanje metode GHT z izboljšanjem računanja.

Izboljšane metode GHT so:

- probabilistična Houghova transformacija (PHT) (Kiryati in sod., 2019),
- adaptivna probabilistična Houghova transformacija (APHT) (Kiryati in sod., 1994),
- progresivna probabilistična Houghova transformacija (PPHT) (Galamhos in sod., 1999),
- naključna Houghova transformacija (RHT) (Xu in sod., 1990),
- Houghova transformacija na osnovi jeder (KHT) (Fernandes in sod., 2008).

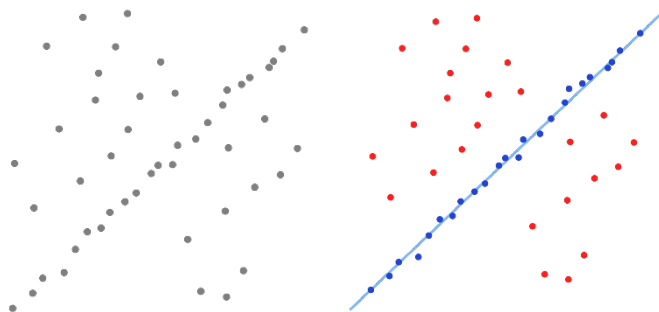
Poleg računskega vpliva je ustrezna izbira oblike akumuliranja rezultatov tudi način za optimizacijo učinkovitosti HT (Bormann, 2011). Na področju gradbeništva so najpogostejše raziskovane geometrijske

oblike prav ploskve. Poleg ploskev lahko segmentiramo tudi druge osnovne geometrijske elemente, kot so valji (Rabbani in Heuvel, 2005) in sfere (Camurri in sod., 2011).

2.4.4.2 Metoda naključnega vzorčenja

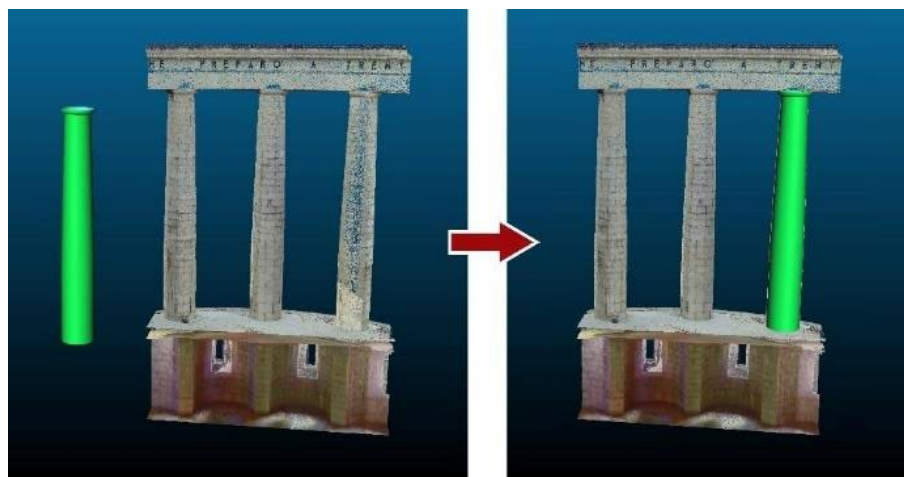
Metoda naključnega vzorčenja (RANSAC) je iterativna metoda ocenjevanja parametrov matematičnega modela iz vhodnih podatkov, ki velikokrat vsebujejo šum in napake (Strutz, 2016). S tem pristopom želimo zmanjšati ali pa celo izničiti vpliv šuma in napak v vhodnih podatkih. Algoritem je sestavljen iz dveh glavnih korakov, ki se iterativno ponavljata. Za kakovostne rezultate je potrebno izvesti zadostno število iteracij. V prvem koraku je podmnožica, ki vsebuje vsaj minimalno količino podatkov za izračun, izbrana naključno iz začetnega nabora podatkov. Primeren model in ustrezni parametri modela se izračunajo z uporabo elementov iz te podmnožice. V drugem koraku algoritem preveri, kateri elementi iz celotnega nabora podatkov so v okviru določene tolerance in skladni z modelom iz prvega koraka. Podatki, ki se nahajajo znotraj tolerance, se štejejo kot ustrezni, zunanji podatki (angl. outliers) so neustrezni. Po zaključenih N iteracijah imamo večje število naključno izbranih vzorcev z različnim naborom podatkov. Vzorec z največ podatki znotraj tolerance je reprezentativen vzorec (Fischler in Bolles, 1981).

Preprost primer je prilagoditev premice naboru podatkov v dveh dimenzijah. Ob predpostavki, da so podatki razdeljeni na prave podatke in šum. Pravi podatki so tisti, skozi katere lahko potegnemo premico. Na spodnji sliki je prikazana premica, ki poteka skozi prave podatke obarvane v modro, rdeči podatki so šum.



Slika 17 Prikaz vhodnih podatkov v 2D prostoru (levo) in pravilnega izračuna (desno) (Wikipedia, 2020)

Na nivoju oblakov točk je glavna prednost metode RANSAC njena robustnost in zmožnost zanesljive ocene parametrov, kar omogoča kvalitetno oceno parametrov ob prisotnem šumu (Schnabel, 2007). Algoritem ne potrebuje zapletene optimizacije in velikih pomnilniških kapacitet v primerjavi s HT metodo (Tarsha-Kurdi, 2007). Poleg ploskev je z algoritmom mogoče zaznavati sfere, cilindre, stožce in toruse (Schnabel, 2007). Kljub temu ima algoritem tudi nekaj pomanjkljivosti. Za določitev elementov algoritem uporablja lokacijo točk, na ta način pa lahko ustvari elemente, ki v naravi ne obstajajo. Slabše deluje pri tudi veliki gostoti točk in kompleksnih geometrijah (Awad in sod., 2010, Tarsha-Kurdi, 2007). V praksi je število iteracij omejeno, izračunani model ni vedno optimalen in ne ustreza začetnemu naboru podatkov. Z večanjem števila iteracij se poveča verjetnost za izračuna pravilnega modela, za kar pa je potrebnega več časa. Poleg tega pa RANSAC v omejenem številu iteracij ne najde vedno optimalnega modela, ko je šuma več od 50% vseh podatkov (Hast, 2007).



Slika 18 Prilagoditev stebra oblaku točk s pomočjo algoritma RANSAC (Grilli, 2017)

2.4.5 Nenadzorovano deljenje

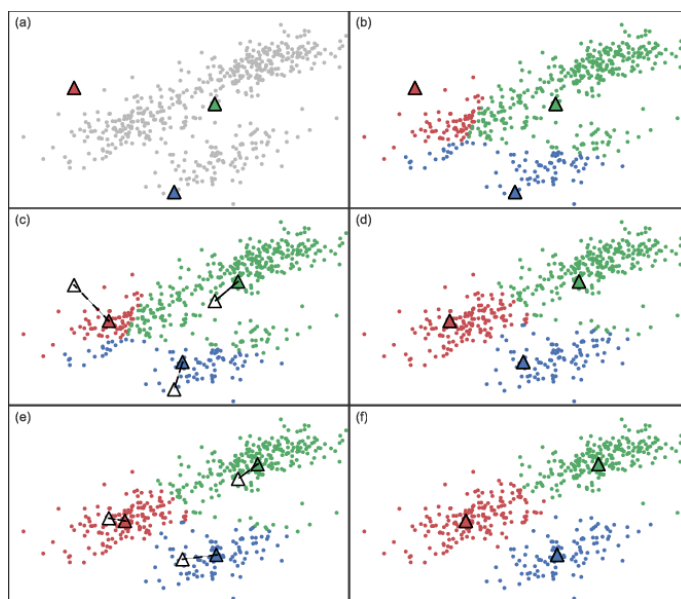
Metode, ki temeljijo na grozdenju (angl. clustering) se uporabljajo za nenadzorovano deljenje. V to skupino spadajo metode, katerih cilj je združevanje točk s podobnimi geometrijskimi značilnostmi in prostorsko porazdelitvijo v isto homogeno skupino. Za razliko od algoritmov rasti regij in metod s prilagajanjem elementov teh vzorcev običajno ne določimo vnaprej, zato se lahko algoritmi grozdenja uporabljajo za segmentacijo nepravilnih oblik, kot je vegetacija. Pri segmentaciji je potrebno izbrati lastnost, na podlagi katere se podatki delijo.

Metode nenadzorovanega deljenja:

- metoda grozdenja na osnovi K-srednjih vrednosti,
- metoda grozdenja s premikom srednjih vrednosti,
- metoda grozdenja z zabrisanimi mejami.

2.4.5.1 Metoda grozdenja na osnovi K-srednjih vrednosti

Metoda K-srednjih vrednosti je iterativen algoritem, ki razdeli nabor točk v različne skupine. Skupine se med seboj ne prekrivajo, vsaka točka pa pripada samo eni skupini. Metoda je sestavljena iz dveh glavnih korakov, ki se iterativno ponavljata. Prvi korak je določitev K začetnih točk, iz katerih na koncu računanja dobimo K skupin. Temu sledi izračun vsote najmanjših kvadratov med vsemi točkami in centriidi posameznih skupin. Točke, ki so najbližje posameznemu centroidu, tvorijo skupino. Po vsaki iteraciji so v skupinah nove točke, s tem pa se spremenijo tudi centriidi. Iteracije se zaključijo, ko ni več sprememb znotraj posameznih skupin, algoritem pa zato spada v metodo grozdenja in ne rasti regij. Z grozdenjem je mogoče segmentirati krošnje dreves (Morsdorf in sod., 2004) in strehe stavb (Sampath in sod., 2006) iz zračno zajetih podatkov. Raziskovalci so uporabili metodo tudi za segmentacijo fasad stavb na osnovi podatkov TomoSAR (Shahzad in Zhu, 2012). Ena od prednosti metode K-srednjih vrednosti je, da jo je mogoče enostavno prilagoditi vsem vrstam lastnosti in jo lahko uporabimo v večdimenzionalnem prostoru. Glavna pomanjkljivost te metode je težka določitev pravilnega števila K. Metodo je za izračun optimalnih rezultatov potrebno večkrat ponoviti. Nekoliko izboljšana različica metode K-srednjih vrednosti je metoda grozdenja z zabrisanimi mejami, kjer so lahko točke za razliko od osnovne metode grozdenja v več skupinah hkrati.



Slika 19 Deljenja oblaka točk v dveh dimenzijah s pomočjo algoritma za grodeznje na osnovi K-srednjih vrednosti. Trikotniki predstavljajo težišče grozda (Benavente, 2017)

2.4.5.2 Metoda grozdenja s premikom srednjih vrednosti

V nasprotju z grozdenjem na osnovi K-srednjih vrednosti je metoda grozdenja s premikom srednjih vrednosti algoritem združevanja, pri katerem se za razliko od klasterizacije izognemo določanju števila skupin že na začetku postopka. Število skupin določi algoritem glede na vhodne podatke. Osnova za algoritem je poljubno jedro (krog ali podobna geometrijska oblika), ki se iterativno premika proti območju največje gostote točk. Pri vsaki iteraciji se jedro premakne na sredino točk znotraj njega, vsak premik pa je definiran z vektorjem. Način izračuna tega povprečja je odvisen od izbire jedra. Učinkovito se uporablja na podatkih ALS v mestnih in gozdnih terenih (Lodha in sod., 2007, Morsdorff in sod., 2004). Algoritmi premika srednje vrednosti so bili uporabljeni tudi na oblakih točk TomoSAR, s čimer so bile segmentirane fasade in drevesa (Shahzad in sod., 2015). Ker sta število skupin in oblika vsake skupine neznani, obstaja velika verjetnost prevelikega števila segmentov pri tem algoritmu zato se po navadi uporablja kot korak predsegmentacije ali rafiniranja podatkov.

2.4.6 Supervokselizacija in predsegmentacija

Pogosta strategija pri segmentiranju je deljenje surovega oblaka točk v manjše skupine. S tem zmanjšamo računski čas in negativne vplive šuma. Najmanjše osnovne regije so vokseli, ki se združujejo v večje enote, imenovane supervokseli. Ker supervokseli lahko še dodatno poenostavijo situacijo z minimalno izgubo informacij in minimalnim prekrivanjem, jih lahko uporabimo v predsegmentaciji. Namesto surovih oblakov točk so te enote vhodni podatki za algoritme, ki delujejo na podlagi vokselov in supervokselov. En izmed bolj razvitih algoritmov supervokselizacije je Voxel Cloud Connectivity Segmentation (VCCS) (Papon in sod., 2013)

2.4.7 Segmentacija na osnovi globokega učenja

Globoko učenje (angl. deep learning) je najpomembnejša in najhitreje rastoča tehnika prepoznavanja vzorcev, računalniškega vida in analize podatkov (Zhu in sod., 2017). Pred uporabo v 3D podatkih se je globoko učenje pokazalo kot učinkovita metoda pri različnih nalogah v 2D podatkih na področju računalniškega vida in obdelave slik. Kljub podobnosti pa zasnove iz 2D ni mogoče enostavno uporabiti v 3D oblakih točk, saj so ti navadno neurejeni, nestrukturirani, nepravilni in vsebujejo šum. Za rešitev tega problema je bistvena transformacija surovega oblaka točk v nekakšno delovno obliko. Glede na obliko podatkov jih lahko razdelimo v tri glavne kategorije.

Glavne kategorije:

- Segmentacija na osnovi več pogledov,
- segmentacija na osnovi vokselov,
- segmentacija na osnovi točk.



Slika 20 Segmentacija oblakov točk s pomočjo globokega učenja (Qi, 2017)

2.4.7.1 Segmentacija na osnovi več pogledov

Segmentacija na osnovi več pogledov (angl. multiview based) je ena izmed prvih rešitev globokega učenja z zmanjševanjem števila dimenzij. 3D prostorski podatki so pri tej metodi predstavljeni z 2D slikami, ki jih je mogoče obdelati z 2D nevronske mreže. Rezultati se po obdelavi sestavijo v nazaj v 3D. Metode, ki temeljijo na več pogledih in zmanjševanju števila dimenzij, so na začetku relativno dobro rešile težave s strukturiranjem prostorskih podatkov v oblaku točk, hkrati pa so imele tudi nekaj pomanjkljivosti. S spremembo dimenzij se omejuje in izgubijo številne geometrijske strukture, saj so 2D slike samo približek 3D prostorskih podatkov. Bolj zapletene naloge, kot so semantična segmentacija, v takih primerih pripeljejo do omejenih in nezadovoljivih rezultatov. Večkrat projicirane slike naj bi v teoriji zajele vse točke in poglede. Pri velikih in zapletenih oblakih točk je pogostokrat težko izbrati dovolj primernih prizorov, zaradi česar so rezultati pomanjkljivi. Ena bolj razvitih metod, ki se uporablja, je SnapNet (Boulch in sod., 2018).



Slika 21 3D oblak točk je razdeljen na segmente 2D slik, ki se ločeno segmentirajo in klasificirano ter potem združijo v 3D (Boulch, 2018)

2.4.7.2 Segmentacija na osnovi vokselov

Segmentacija na osnovi vokselov (angl. voxelized) s 3D nevronskimi mrežami je drugi pristop semantične segmentacije oblakov točk. Postopek vokselizacije reši del težav surovih, neurejenih in nestrukturiranih oblakov točk. Glavna težava je še izguba določenih podatkov v procesu vokselizacije. Vokselska struktura podatkov je v primerjavi z oblaki točk nižje ločljivosti. Poleg tega pa vokselizirane strukture ne shranjujejo samo zasedenih, ampak tudi prazne prostore, zaradi se računski čas podaljša. Ena prvih nevronskih mrež je bila VoxNet (Maturana in Scherer, 2015), ki je bila testirana na objektih knjižnice SegCloud je bil pred leti najsodobnejši pristop z dobrimi rezultati na testnih primerih na semantic3D.net. Težave so se pojavljale v obliki dolgih računskih časov in velike porabe spomina zaradi vokselske strukture. Kot izboljšani različici sta se pojavila OctNet (Riegler in sod., 2017) in O-CNN (Wang in sod., 2015).

2.4.7.3 Segmentacija na osnovi točk

Pri metodah, ki temeljijo na večkratnem gledanju kot tudi na vokselih, obstajajo omejitve (izguba in posplošitev podatkov) je smiselna uporaba metod, ki temeljijo na segmentaciji točk kot osnovnih elementov. Do sedaj se je pojavilo veliko pristopov in se še vedno pojavljajo (Qi in sod., 2017). Za razliko od uporabe ločenih operacij transformacije v primerih, ki temeljijo na več pogledih in vokselih, je v teh pristopih kanonizacija zavezujoča z nevronske mrežno arhitekturo. PointNet (Qi in sod., 2017) je pionirski okvir globokega učenja, ki je bil izveden neposredno na oblakih točk.

2.5 Programska oprema za delo z oblaki točk

2.5.1 Autodesk Recap

Autodeskova komercialna programska oprema ReCap (ReCap, 2021) je preprosta programska rešitev, ki omogoča neposredno odpiranje datotek oblakov točk. S pomočjo nekaj prilagodljivih nastavitev uvoza omogoča filtriranje odvečnih podatkov. Datoteke formatov .rcp in .rcs, ustvarjene z ReCapom, je mogoče enostavno uporabljati v vseh ostalih programih podjetja Autodesk. ReCap tako deluje kot podporni program v procesu dela z oblaki točk, s katerim lahko uredimo, očistimo, obrežemo in sestavimo oblake točke pred nadaljnjo uporabo. Laserske posnetke z različnih stojišč lahko povežemo v zaključene celote in jih georeferenciramo. Oblake točk v programu lahko prikazujemo glede na barve (RGB) in intenziteto, hkrati pa lahko merimo dolžine, kote, površine in koordinate posameznih točk. Urejen oblak točk lahko iz ReCapa prenesemo v Civil 3D, Revit ali Navisworks in ga uporabimo kot podlago za nadaljnje delo.

2.5.2 CloudCompare

CloudCompare je prosto dostopna programska oprema za obdelavo oblakov točk in trikotniških mrež (TIN). Projekt CloudCompare (CloudCompare, 2021) je bil ustvarjen kot produkt sodelovanja dveh francoskih podjetij Telecom ParisTech in oddelka za raziskave in razvoj na Electricite de France (EDF) leta 2003. Takrat je bil namenjen hitri zaznavi sprememb in razlik oblakov točk visoke gostote, pridobljenih z laserskimi skenerji v industrijskih objektih ali na gradbiščih. Skozi čas se je razvil v napredno prosto dostopno programsko opremo za obdelavo oblakov točk in trikotniških mrež. Za razliko od Autodeskovega ReCapa, ki omogoča osnovne funkcije obdelave oblakov točk, nam CloudCompare poleg osnovnih orodij ponuja več naprednih algoritmov. Omogoča medsebojne primerjave različnih oblakov točk, izračun njihovih projekcij, segmentacijo, različne statistične in geometrijske izračune, kot so normale, ukrivljenost in gostoto.

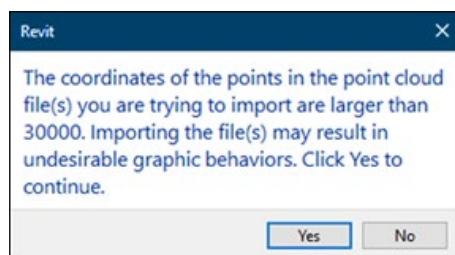
2.5.3 Revit

Autodesk Revit (Revit, 2021) je komercialna programska oprema za modeliranje zgradb, namenjena arhitektom, krajinskim arhitektom, gradbenim inženirjem, inženirjem strojnih, električnih in vodovodnih inštalacij, oblikovalcem in izvajalcem. Izvirno programsko opremo je razvil Charles River Software, podjetje, ustanovljeno leta 1997, leta 2002 pa je postala del Autodesk. Programska oprema uporabnikom omogoča modeliranje elementov v 3D, pripravo načrtov v 2D na osnovi modelov, vnos in dostop do informacij o posameznih elementih. Revit je BIM orodje za modeliranje zgradb, kjer lahko z različnimi orodji tudi načrtujemo in spremljamo življenjski cikel zgradb.

Delovno okolje programa je prilagojeno ustvarjanju in manipulaciji BIM elementov, njihovih posameznih delov in celotnih gradbenih sklopov. Orodja za modeliranje lahko uporabimo za delo z obstoječimi, novo ustvarjenimi in uvoženimi elementi (angl. families). Uporabniški vmesnik je razdeljen na sklope, namenjene posameznim disciplinam v procesu modeliranja.

V osnovi je Revit prilagojen ustvarjanju novih in obdelavi obstoječih projektov, nekoliko manj pa obratnemu inženirstvu, kjer iz laserskih posnetkov poustvarjamo obstoječe stanje. Obstoječe funkcije znotraj programa trenutno ne omogočajo poenostavljenega, hitrega in natančnega obratnega modeliranja na osnovi oblakov točk. Modeliranje poteka ročno, z izbiro elementov in njihovo umestitvijo v oblak

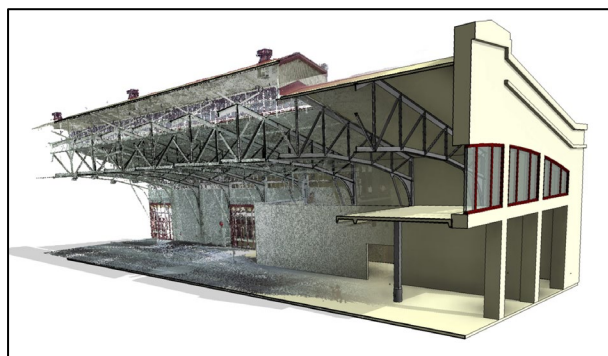
točk. Ročni postopek je zamuden in daje nenatančne rezultate, zato so se razvili dodatki za avtomatizacijo tega procesa. Prepoznavanje elementov znotraj oblaka točk je tako izključno v domeni uporabnika brez pomoči programa. Dodatne težave znotraj Revita se pojavljajo tudi pri georeferenciranju modelov znotraj oblaka točk. Program ni prilagojen delu z velikimi koordinatami znotraj koordinatnih sistemov, kot je Gauss–Krüger, se pojavljajo težave z glavnimi funkcijami (offset, snap). Ozko grlo v celotnem procesu pa predstavlja omejeno število datotečnih formatov, s katerimi operira Revit.



Slika 22 Pojavno okno z napako pri koordinatah visokih vrednosti (povzeto po Faro, 2019)

2.5.4 Faro As-Built

Komercialna programska oprema FARO As-Built (Faro, 2019) ponuja enostavno, natančno, hitro in učinkovito obdelavo laserskih posnetkov v modele CAD in BIM. As-Built podpira običajne datotečne formate in omogoča delo z 3D modeli v vseh panogah arhitekture, inženiringa in gradbeništva. Program lahko uporabljamo samostojno ali kot dodatek Autodeskovima Revitu in AutoCAD. Oba vtičnika se brez težav integrirata v Revit in AutoCAD, s tem pa dodatno razširita funkcionalnosti osnovnih programov na tak način, da je oblake točk mogoče obdelovati neposredno.



Slika 23 Model iz oblaka točk v programu Faro As – Built (Faro, 2019)

S številnimi funkcijami omogoča prikaz in navigacijo po oblakih točk. Oblake točk je mogoče poljubno obrezati, obrezane segmente pa smiselno uporabiti. Z orodjem za avtomatizirano zaznavanje in ustvarjanje elementov, kot so stene, cevi, stebri, nosilci, lahko te elemente hitro in natančno priredimo oblakom točk. Poleg dela z obstoječimi elementi omogoča dodatek tudi ustvarjanje novih Revitovih elementov (angl. families) na osnovi oblakov točk. Na ta način lahko preprosto ustvarimo vrata, okna, stebre in ostale dodatke, ki so specifični in jih ni v obstoječih knjižnicah. Poleg ustvarjanja elementov omogoča tudi kvalitativno analizo in zaznavanje trkov (angl. clash detection) med modelom in oblakom točk.

Zaradi preprostega uporabniškega vmesnika omogoča hiter delotok brez predhodnega poznavanja osnovnega programa Revit ali AutoCAD. Kljub preprostosti so rezultati natančni, njihovo natančnost pa vedno lahko preverimo kvalitativno. Pri delu z uporabniškim vmesnikom so datotečni formati omejeni na osnovni program, samostojni program omogoča nekoliko več svobode pri izbiri formatov.

2.5.5 Leica Cloudworx

Komercialni vtičnik Leica CloudWorx omogoča preprosto in učinkovito delo z obsežnimi oblaki točk. Leica poleg strojne opreme ponuja programske rešitve za geodetsko delo, kar se pokaže v kakovosti in številu funkcij, ki jih s tem programskim orodjem zajame. Program deluje samostojno, lahko pa se, tako kot Faro, integrira v AutoCAD, Revit, Navisworks, BricsCAD, Solidworks in Benteley.

Številni programi v svoji osnovi omogočajo delo z oblaki točk, ki pa v večini primerov ni prilagojeno velikim količinam podatkov. Ena pomembnejših funkcij, ki jih Leica s svojim programskim vtičnikom nudi, je zmogljivo in učinkovito obdelovanje obsežnih oblakov točk. Cloudworx poskrbi za poenostavljeno delo in hitro prikazovanje oblakov točk, ki bi bili za osnovne programe preveliki. Za boljšo predstavbo ponuja 360 stopinjske poglede znotraj oblakov točk. Poleg tega omogoča poljubno obrezovanje oblakov točk in avtomatizirano modeliranje sten, cevi, stebrov nosilcev in ostalih gradbenih elementov. Tako kot Faro ponuja tudi Leica kvalitativno analizo in zaznavanje trkov.

Uporabniški vmesnik je nekoliko neurejen, kar pa je glede na število funkcij razumljivo. Krivulja časa učenja je zato nekoliko bolj strma kot pri ostalih programih. Pri modeliranju sten pa je korak pred ostalimi, ker omogoča modeliranje sten, ki niso popolnoma vertikalne. Tako kot pri ostalih vmesnikih je glavna težava povezana z datotečnimi formati, saj so le ti vezani na osnovni program. S samostojnim programom lahko operiramo z večjim številom izmenjevalnih formatov (Leica CloudWorx, 2021).

2.5.6 ClearEdge3D EdgeWise

Programska rešitev z bolj preprostim uporabniškim vmesnikom, ki hkrati omogoča največ avtomatizacije pri modeliranju elementov, je EdgeWise (ClearEdge3D, 2021). Program vsebuje algoritme za avtomatizacijo, ki omogočajo samodejno ekstrakcijo skeleta zgradb in cevi, kar je velika prednost v primerjavi s konkurenčnimi programi na trgu. Podjetje ClearEdge3D tako kot ostali ponudniki ponuja samostojno različico programa ali pa uporabniški vmesnik Autodeskovemu Revitu. Podjetje so leta 2006 ustanovili priznani strokovnjaki s področja računalniškega vida, hiperspektralne analize podatkov in analize podatkov lidar. Prva različica EdgeWise Building, ki se je osredotočila na modeliranje zgradb in terenov, je bila predstavljena leta 2009.



Slika 24 Prikaz modeliranih cevi znotraj programa EdgeWise (ClearEdge3D, 2020)

Veliko stopnja avtomatizacije omogoča velik prihranek časa. Poleg avtomatizacije je omogoča tudi ročno ali z avtomatizacijo podprta modeliranje elementov BIM in CAD. Licenciranje je razdeljeno v sklope, s tem pa je uporabnikom omogočen nakup tistega modula, ki ga za delo potrebujejo. Enako je z ostalimi sklopi modeliranja cevi, jeklenih elementov in kontrole trkov.

2.5.7 PointCab

Nekoliko preprostejša programska rešitev z najmanj avtomatizacije je PointCab (PointCab, 2021). Poleg modeliranja zgradb omogoča obdelavo terenskih podatkov in delo z digitalnimi modeli terena, za razliko od ostalih programov, ki so usmerjeni v modeliranje konstrukcijskih elementov. Program, tako kot vsi prejšnji, deluje samostojno, lahko pa se uporablja kot razširitev Revitu ali Archicadu, s tem pa omogoča modeliranje CAD in BIM. Programske funkcije delujejo na osnovi modulov, tako kot pri EdgeWise, kjer uporabnik zakupi tiste, ki jih za delo potrebuje.

Za razliko od prej opisanih avtomatiziranih programskih rešitev PointCab omogoča tudi modeliranje različnih masnih elementov. Delotok je nekoliko počasnejši v primerjavi z ostalimi programi, ni pa zato nič manj natančen. Glavna prednost je modeliranje cilindričnih, poligonalnih, zakrivljenih in ostalih nepravilnih oblik, ki se pojavljajo v gradbeništvu in arhitekturi. Kvalitativni prikaz deformacij in odstopanj od oblaka točk ponuja modul za analizo, ki pa ne ponuja zaznavanja trkov.

Je ena redkih programskih rešitev, ki razširi funkcionalnost dela z oblaki točk v Archicadu. Z razširitvijo omogoča delo z večjim številom izmenjevalnih formatov, kot jih Archicad v osnovi omogoča. Uporabniški vmesnik je glede na število funkcij primerno enostaven in strukturiran tako, da uporabnika smiselno vodi skozi delotok.

2.5.8 Sklep

Pri izmenjavi formatov je potrebno poznati koncept strukturiranih in nestrukturiranih zbirk podatkov v posameznih formatih. Strukturirane zbirke podatkov imajo znotraj datoteke zapisane lokacije stojišč laserskega skenerja v prostoru. S temi informacijami lahko v programih vidimo, kateri deli oblaka točk so nastali znotraj posameznih stojišč. Nestrukturirani podatki pa združujejo posamezne posnetke v zvezno celoto. Praviloma je pri delu mogoče preiti iz strukturiranih podatkov v nestrukturirane, vendar ne obratno.

V kombinaciji z lastniškimi tipi formatov, ki so bili opisani v petem poglavju, se močno zmanjša interoperabilnost formatov. Tako lahko lastniške strukturirane formate, kot so .ptg (Leica), .fls (Faro), .tzf (Trimble) uvozimo različne programe, ne moremo pa jih zapisati v iste. Težava je enaka pri obratnem načinu dela. Število formatov s katerimi lahko prosto operiramo in jih medsebojno izmenjujemo se tako zmanjša na .e57 in ASCII formate .txt.

V preglednici spodaj so strnjene lastnosti posameznih programov in izmenjevalni formati, ki jih podpirajo.

Preglednica 3 Pregled programske opreme za modeliranje in delo z oblaki točk

	Prednosti	Omejitve	Izmenjevalni format	
			Uvoz	Izvoz
ReCap	<ul style="list-style-type: none"> Preprost uporabniški vmesnik Obdelava obsežnih oblakov točk 	<ul style="list-style-type: none"> Malo izmenjevalnih formatov Osnovne funkcije obdelave (obrezovanje) 	<ul style="list-style-type: none"> .e57 .pts .rcs 	<ul style="list-style-type: none"> .e57 .pts .rcs
CloudCompare	<ul style="list-style-type: none"> Algoritmi za delo z oblaki (izračun normal, segmentacija) Prosto dostopen 		<ul style="list-style-type: none"> .e57 .las .pts .flw .ptg .zfs .dp . 	<ul style="list-style-type: none"> .e57 .las .pts .obj .ply .stl .pcd
Revit	<ul style="list-style-type: none"> BIM modeliranje zgradb 	<ul style="list-style-type: none"> Malo izmenjevalnih formatov Ni prilagojen samostojnemu modeliranju na oblakih točk 	<ul style="list-style-type: none"> .e57 .pts 	/
As - Built	<ul style="list-style-type: none"> Preprost uporabniški vmesnik Kvalitativna analiza Detekcija trkov 	<ul style="list-style-type: none"> Programski vmesnik samo za Revit 	<ul style="list-style-type: none"> .e57 .pts .flw .fls .fws 	<ul style="list-style-type: none"> .e57 .pts .flw .fls .fws
Cloudworx	<ul style="list-style-type: none"> Modeliranje poševnih sten Programski vmesnik za veliko programov (Benteley, Navisworks, Solidworks) 	<ul style="list-style-type: none"> Nepregleden uporabniški vmesnik 	<ul style="list-style-type: none"> .e57 .pts .las .prj .ptg .ptx .dp 	<ul style="list-style-type: none"> .e57 .pts .prj .ptg .ptx

	<ul style="list-style-type: none"> • Ogled 360° fotografij • Kvalitativna analiza in detekcija trkov 		<ul style="list-style-type: none"> • .zfc 	
EdgeWise	<ul style="list-style-type: none"> • Največja stopnja avtomatizacije (skelet objektov) • Preprost uporabniški vmesnik • Detekcija trkov • Kvalitativna analiza in detekcija trkov 	<ul style="list-style-type: none"> • Programski vmesnik samo za Revit 	<ul style="list-style-type: none"> • .e57 • .pts • .dp • .fls • .rcp • .ptg • .zfs 	<ul style="list-style-type: none"> • .e57 • .pts
PointCab	<ul style="list-style-type: none"> • Modeliranje masnih elementov nepravilnih oblik • Kvalitativna analiza • Programski vmesnik za Archicad 	<ul style="list-style-type: none"> • Najmanj avtomatizacije • Ne omogoča detekcije trkov 	<ul style="list-style-type: none"> • .e57 • .pts • .las 	<ul style="list-style-type: none"> • .e57 • .pts

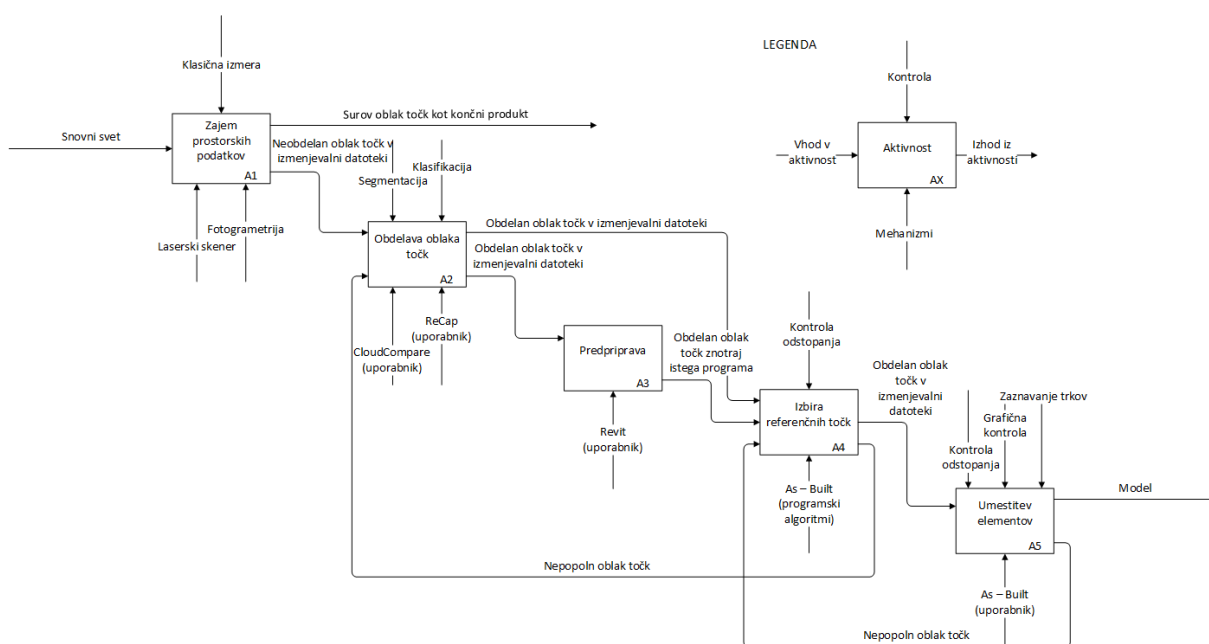
3 DELOTOK PRETVORBE OBLAKOV TOČK V BIM ELEMENTE

3.1 Uvod

V strokovni literaturi se v zadnjih letih pojavljajo načini avtomatiziranega modeliranja iz oblakov točk. Celoten proces pretvorbe snovnega sveta v BIM modele ima več korakov.

- V prvem koraku zajamemo prostorske podatke z laserskim skenerjem ali metodami fotogrametrije. Zajet in neobdelan oblak točk je lahko v tej točki tudi končni produkt, lahko pa ga tako kot v našem primeru, uporabimo kot osnovo za nadaljnje delo.
- V drugem koraku oblak točk obdelamo za nadaljnje delo. S programom CloudCompare in Recap ga segmentiramo, klasificiramo, obrežemo in očistimo šum.
- Obdelan oblak točk je osnova za tretji korak predpriprave izrezov v Revitu za modeliranje sten. Ta korak je opcijski in ga pri modeliranju stebrov, cevi in nosilcev lahko izpustimo.
- V četrtem koraku z dodatkom As – Built znotraj Revita izberemo referenčne točke, ki so osnova algoritmom v petem koraku za določitev in prostorsko umestitev elementov.
- Znotraj zadnjega koraka kontroliramo element, izvajamo grafično kontrolo odstopanja elementa od oblaka točk in zaznavamo potencialne trke. V primeru slabih rezultatov se lahko vrnemo v četrti korak in ponovimo izbiro referenčnih točk ali pa na drugi korak obdelave, kjer oblak točk dodatno uredimo. Končni rezultat je natančen model, narejen na osnovi oblakov točk. V spodnjem IDEF0 diagramu je prikazan celoten postopek avtomatiziranega modeliranja. Prvi in drugi korak smo v nalogi izpustili, ker smo prejeli obdelane oblake točk.

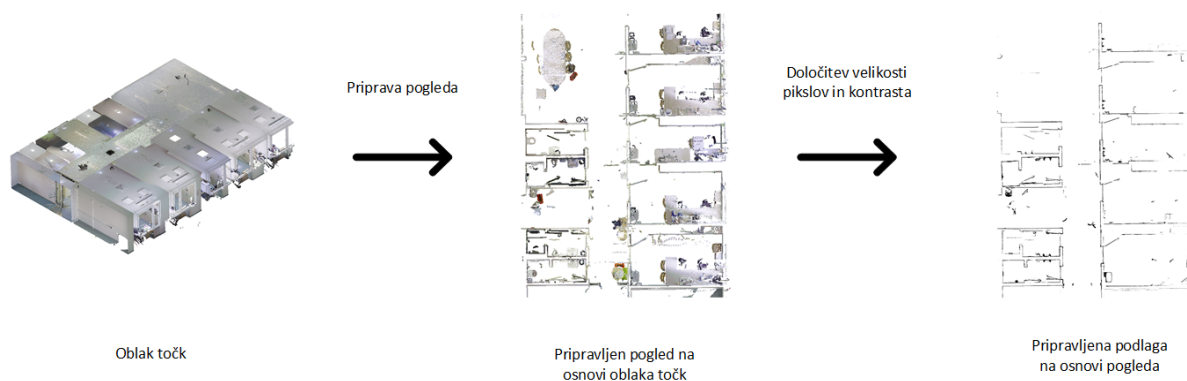
Za modeliranje smo v nalogi uporabili Autodeskov komercialni program Revit in dodatek za avtomatizirano modeliranje As – Built, podjetja Faro, ki deluje kot nadgradnja osnovnega programa. Za ta dodatek smo se odločili, ker omogoča avtomatizirano podporo pri modeliranju z izbiro nekaj referenčnih točk. Strukturiran uporabniški vmesnik omogoča preprosto delo v znotraj posameznega sklopa.



Slika 25 IDEF0 diagram pretvorbe snovnega sveta v BIM model.

V nadaljevanju poglavja so v štirih sklopih zajeti glavni procesi pri modeliranju. Proces zajema prostorskih podatkov (A1) je v nalogi zgolj omenjen, ker smo vse zajete oblake točk prejeli v nestrukturirani obliki iz zunanjih virov. Delotok smo v prvem poglavju (poglavje 3.2) začeli s procesom obdelave oblaka točk (A2), ki mu je sledil korak predpriprave (A3). Dobra predpriprava je osnova za kvalitetne in natančne modele sten. Modeliranje sten, nosilcev in cevi je razdeljeno na tri ločena podpoglavja za vsak obravnavan sklop. Modeliranje sten na predpripravljeni podlagi je opisano v drugem podpoglavju. V tretjem poglavju je prikazano modeliranje cevi in njihovih spojev. V zadnjem poglavju je obdelano modeliranje nosilcev in stebrov. Znotraj vseh treh podpoglavij (poglavja 3.3, 3.4 in 3.5) je na nekaj primerih prikazan postopek izbire referenčnih točk (A4) in umestitev elementov (A5) z vsemi analizami znotraj posameznih korakov.

3.2 Priprava oblakov točk



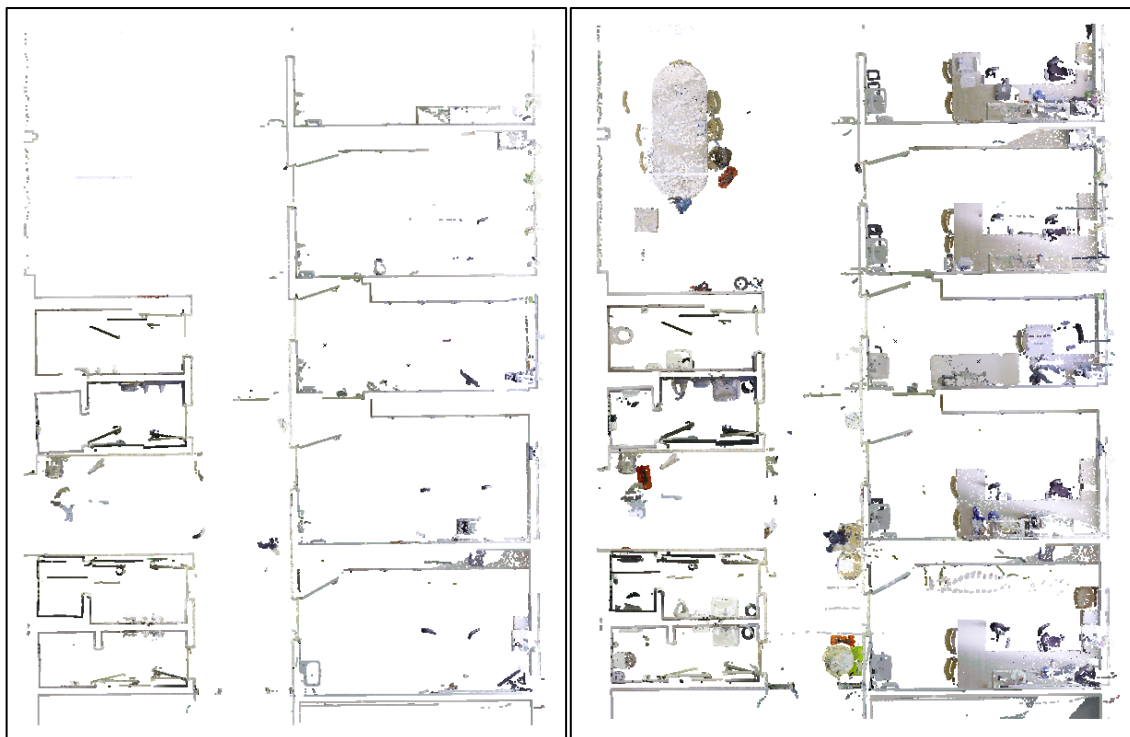
Slika 26 Ilustracija postopka priprave oblakov točk za modeliranje sten znotraj tega poglavja.

Pred modeliranjem zidov je treba oblak točk prilagoditi in pripraviti za to nalogo. Osnova je v tem primeru algoritem detekcije robov, ki iz pripravljenega pogleda izriše robove posameznih elementov in jih prikaže kot skelet. Oblak točk je tako preslikan iz treh dimenzij v dve, točke oblaka pa v piksele. Za boljše zaznavanje robov je smiselno določiti stvari, katere želimo, da so v procesu modeliranja vidne. Želimo imeti čim manj šuma in senc, ki jih povzročajo predmeti v prostoru. V naslednjem podpoglavju je prikazana razlika med pogledom, ki je za modeliranje bolj ustrezen, in pogledom, ki je zaradi prisotnega šuma nekoliko manj ustrezen. Razlike v modeliranju na obeh primerih pa so prikazane v nadaljevanju poglavja.

3.2.1 Priprava izrezov oblakov točk

Posamezno nadstropje je potrebno pripraviti tako, da vsebuje čim manj odvečnih objektov in šuma, ki bi lahko motili algoritme in uporabnika pri modeliranju. V podoknu Properties -> View range, programa Revit, lahko uredimo poglede, v katerih delamo. V tem podoknu iz oblaka točk na željeni višini izrežemo del. Izogibamo točkam na tleh, zato izrez pripravimo nad tlemi posameznih nadstropij. Po nekaj poskusih na različnih oblakih točk se je pokazalo, da je najprimernejša višina spodnjega dela izreza 1,5 m, zgornja pa 2 m. S tem se izognemo različnim predmetom, ki so v prostorih postavljeni na človeku dostopnih višinah. Taki predmeti so stoli, mize, umivalniki, radiatorji. Nekaterim predmetom, kot so omare, ki potekajo skoraj po celotni višini prostora, se je v tem delu procesa skoraj nemogoče izogniti,

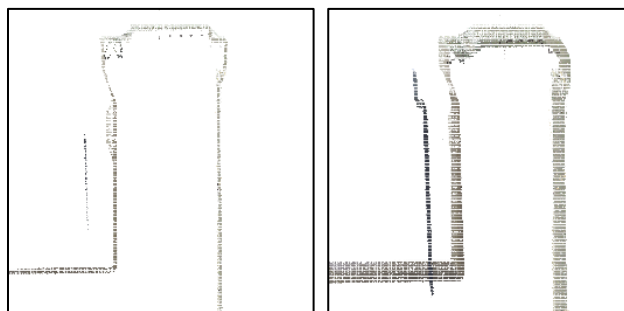
zato pa na tej višini izreza zajamemo okna in parapetne zidove. Pri delu se je potrebno zavedati, da popolnega primera ni nujno mogoče doseči.



Slika 27 Primerjava urejenega oblaka točk (levo) in neurejenega (desno)

Na zgornji sliki sta prikazana dva izreza na različnih višinah, oba debeline 30 cm. Na levem je spodnja višina izreza 1,8 m, kjer lahko vidimo skelet prostorov in nekaj omar. S takim izrezom je prostorski prikaz bolj pregleden. Višina desnega prereza je 1,2 m, v njem pa se nahaja veliko več motečih predmetov. Poleg višine izreza je pri tem delu pomembna tudi njegova debelina. Večja kot je debelina izreza, več točk iz oblaka je prikazanih, zaradi česar je v primeru manj natančnega laserskega posnetka lahko prisotnega več šuma, ki lahko moti detekcijo in zmanjšuje natančnost modeliranja. Po drugi strani pa premalo točk ne prikaže celotne situacije.

Na spodnjih slikah je prikazan detajl vratnega podboja in vogala med dvema stenama. V levem primeru je debelina izreza 30 cm, kjer je mogoče prepoznati glavne karakteristike. Na desnem je debelina reza 1,2 m in je zato prikazanega tudi več šuma na stenah. Karakteristike je na desnem še mogoče prepoznati, niso pa tako izrazite kot v levem primeru, kjer je rez tanjši in bolj skoncentriran. V desnem primeru je prisoten tudi šum oz. napaka pri skeniranju. Takim napakam se želimo v čim večji meri izogibati, ker dajejo napačne prostorske informacije.

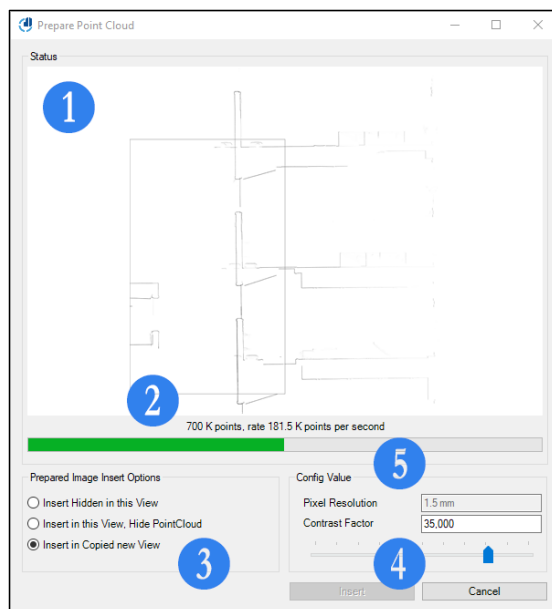


Slika 28 Primerjava različnih debelin prereza oblaka točk. Na levi je prerez tanjši in zato primernejši, na desni je debelejši in manj primeren.

Idealnega recepta za določitev debeline reza in njegovih višin ni, v vsakem primeru je potrebno poznati celotno sliko skeniranega območja in temu primerno prirediti parametre. Vsak oblak je v programih za manipulacijo mogoče dodatno obrezati in očistiti šum, s tem pa kakovost še izboljšati.

3.2.2 Priprava podlag na osnovi izrezov

Iz pripravljenega pogleda, opisanega v prejšnjem podpoglavju, lahko v zavihku As-Built – Object tools – Prepare point cloud, pripravimo podlago za modeliranje. Algoritem pretvori izbrani pogled iz 3D prostorskih točk v 2D sliko, ki služi kot osnova za avtomatizirano modeliranje. Sliko lahko vstavimo v poljubne poglede, uporabimo pa jo lahko tudi kot končni izdelek, ki služi kot grob shematski prikaz. V kombinaciji z dobro pripravljenimi izrezi iz prejšnjega poglavja in nastavitvami v tem lahko pripravimo kvaliteto podlago, ki deluje kot dobra osnova za avtomatizirano modeliranje sten. Na naslednji sliki je prikazano pojavno okno za pripravo podlage.



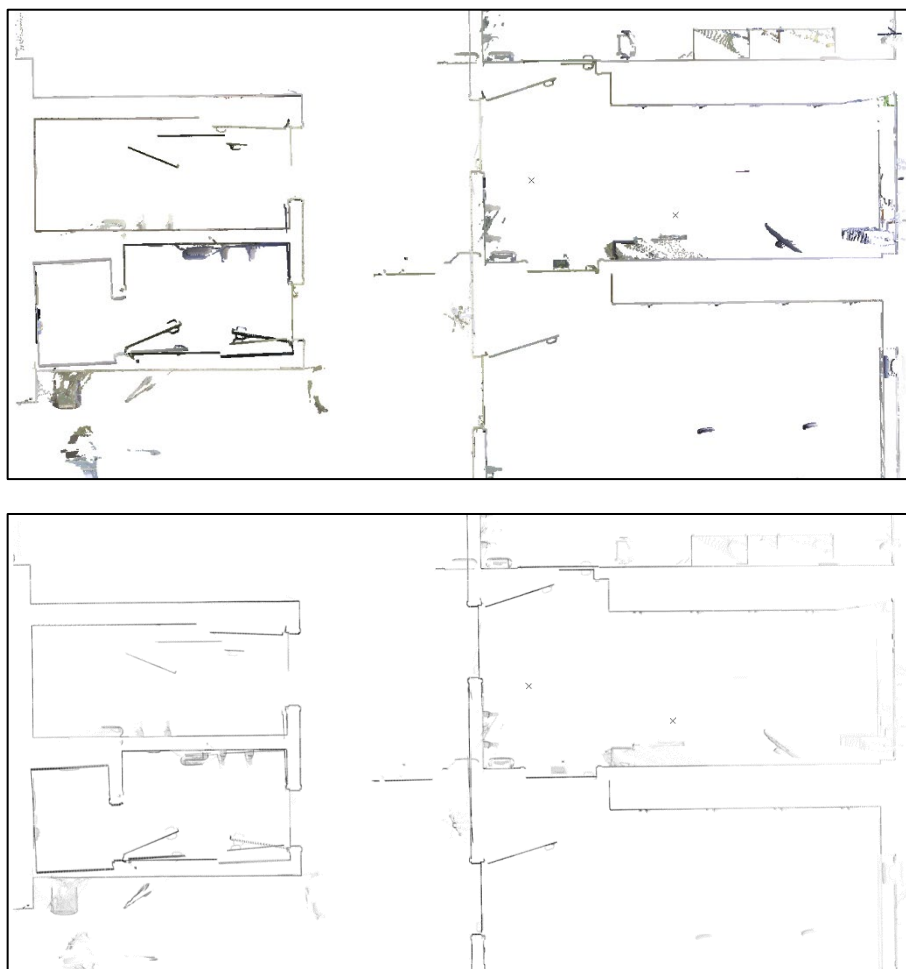
Slika 29 Pojavno okno za rasterizacijo tlorisa.

Pregled pojavnega okna

1. Okno za prikaz zaznanih elementov, ki se sproti posodoblja glede na zaznane elemente.

2. Status zaznavanja, hitrost zaznavanja in skupno število do sedaj zaznanih točk.
3. Izbira pogleda, v katerem se prikaže pripravljena podlaga. Za samo delo je v večini primerov najboljša izbira »Insert in Copied new View«, ki ustvari nov pogled, ločen od oblaka točk. Ta pogled je tudi najbolj pregleden za uporabnika in za avtomatizirano modeliranje.
4. Nastavitev kontrasta omogoča bolj izrazit prikaz robov. Z višjo stopnjo kontrasta robovi bolj izstopajo, s tem pa so bolj pregledni za uporabnika. Robove z večjim kontrastom algoritem zaznava robov lažje prepozna, saj izstopajo iz ozadja.
5. Nastavitev omogoča prilagoditev velikosti prikazanih točk. Z manjšimi točkami so robovi veliko bolj ostri, natančnost modeliranja pa posledično večja.

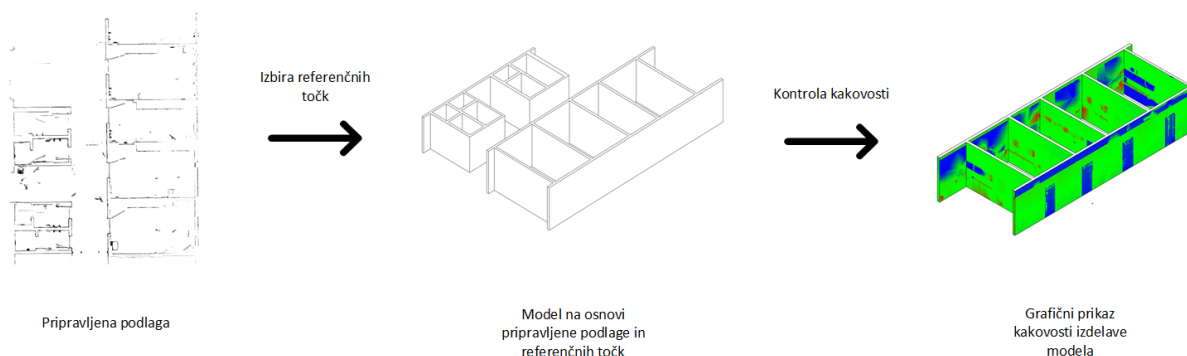
Natančnost prikaza lahko dodatno nastavljamo tudi v Revitu. V zavihku Detail level lahko izbiramo med tremi nastavitvami Coarse, Medium in Fine. Natančnost prikaza je tudi odvisna od oddaljenosti in kontrasta, ki smo ga določili v postopku priprave podlage.



Slika 30 Primerjava surovega oblaka točk v tlorisu (zgoraj) in pripravljenega pogleda (spodaj)

Zgornja slika prikazuje pogled surovega oblaka točk, spodnja pa obdelano podlago za modeliranje. Sivinski kontrast, določen na podlagi gostote in intenzitete točk, v situaciji bolj odstopa kot barvna slika. Na ta način lahko vidimo dele, kjer je gostota večja, obarvane v črno. Na delih, kjer so točke dlje od stojišča laserskega skenerja in jih je zato manj, so te točke svetlejše.

3.3 Avtomatizirano modeliranje sten



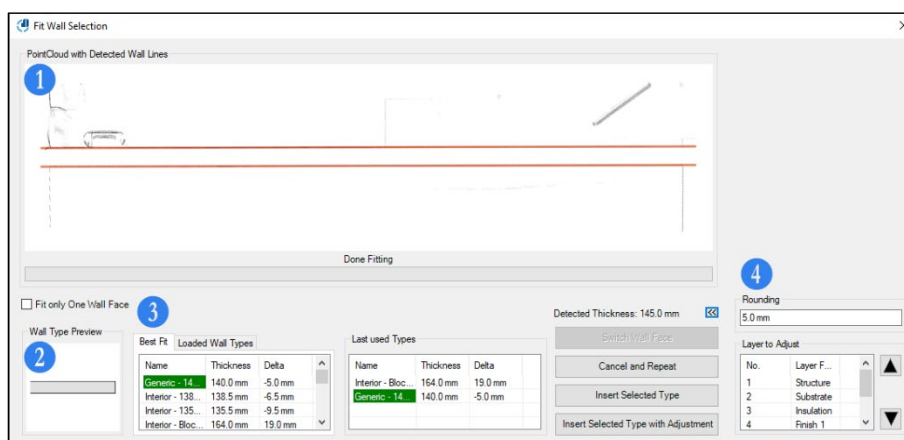
Slika 31 Ilustracija postopka avtomatiziranega modeliranja sten znotraj tega poglavja.

3.3.1 Izbira točk

Predhodno pripravljena podlaga nam v tem poglavju služi kot osnova za avtomatizirano modeliranje sten, smeri, višin in njihovega odstopanja. V zavihku As Built – Fit wall se odpre pojavno okno, kjer določimo zgornje nadstropje modelirane stene ali pa njeno višino, ko omejitev zgornjega nadstropja ni določena. Na mestih pomanjkljivih podatkov omogoča tudi določitev stene na osnovi ene posnete strani. Če je zaradi ovir vidna samo ena stran stene, lahko to s steno kljub temu modeliramo. Ta opcija se izkaže nadvse uporabna v primerih, kjer so laserski posnetki narejeni znotraj ene sobe, povezav z ostalimi prostori pa ni. V naslednjem koraku izberemo začetno in končno točko stene. Algoritem deluje na osnovi uporabnikove izbire dveh ali več točk, ki služijo kot referenca za zaznavo. Ker so stene linearni elementi, za njihovo zaznavo v dobro pripravljenih oblakih točk zadostujeta že dve točki. V primerih, kjer je šuma in okoliških predmetov več, program pa z dvema določenima točkama ne uspe zaznati poteka in debeline stene, mu lahko pomagamo z izbiro več točk.

Pojavno okno spodaj prikazuje primer modeliranja stene znotraj konture oblaka točk. Stena je ravna, brez lomov, šuma in objektov, ki bi zakrivali njen potek. Na levi strani stene je zaradi bližine stojišča laserskega skenerja gostota točk večja kot na desni. Na podlagi izbranih nastavitev program prikaže izračunano odstopanje med steno in oblakom točk.

Pregled pojavnega okna

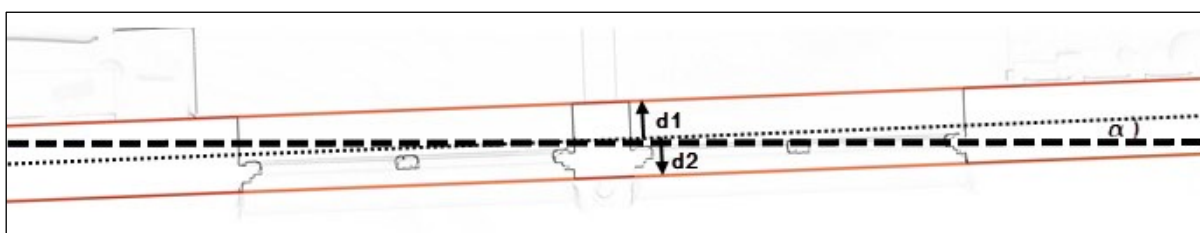


Slika 32 Pojavno okno za detekcijo in umestitev sten

1. Okno za prikaz zaznanih sten, ki prikazuje potek umeščene stene v modelu.

2. Podokno za grafični prikaz elementov.
3. Okno za izbiro vrste zidu s stolpcema za prikaz debeline in odstopanja. V prvem stolpcu je navedena debelina osnovnega Revit elementa, v drugem pa razlika med dimenzijo osnovnega elementa iz programske knjižnice in dimenzijo zaznanega elementa.
4. Nastavitev zaokrožitve debeline sten in sloja, ki se lahko prilagaja.

Črtkana črna na spodnji sliki predstavlja os stene, ki jo določimo z izbiro dveh točk znotraj oblaka točk. Oranžna črta predstavlja zaznane točke, ki na obeh straneh osi ležijo na izračunani razdalji $d1$ in $d2$. Izračunane točke predstavljajo stranice stene. Na osnovi izbranih točk program izračuna tudi kot rotacije stene v prostoru. Na spodnji sliki je ta kot označen z α . Rezultat avtomatskega procesa je stena z natančno izračunano debelino in kotom rotacije v prostoru. Taka stena je na podlagi tega natančno pozicionirana v prostoru.

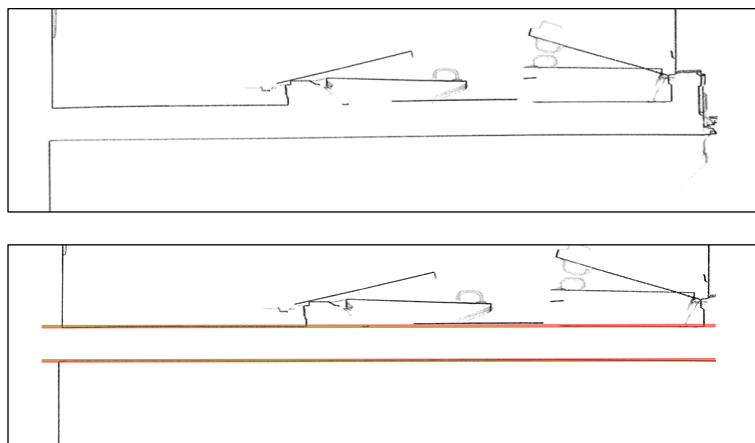


Slika 33 Prikaz delovanja algoritma za umestitev sten (povzeto po FARO, 2019)

Prekinitve zaradi pomanjkljivih laserskih posnetkov, шумov ali senc ne vplivajo bistveno na zaznavo, kar je pomembno v notranjih prostorih, kjer pohištvo prekriva določen del stene in ga v času skeniranja ni potrebno premikati. Prekinitve zaradi odprtín, kot so vrata in okna, ne vplivajo na potek, ker je za izračun poteka stene dovolj že nekaj točk, s katerimi program izračuna smer, pozicijo in debelino stene.

Primer 1 – Prekinitve zaradi predmetov v prostoru

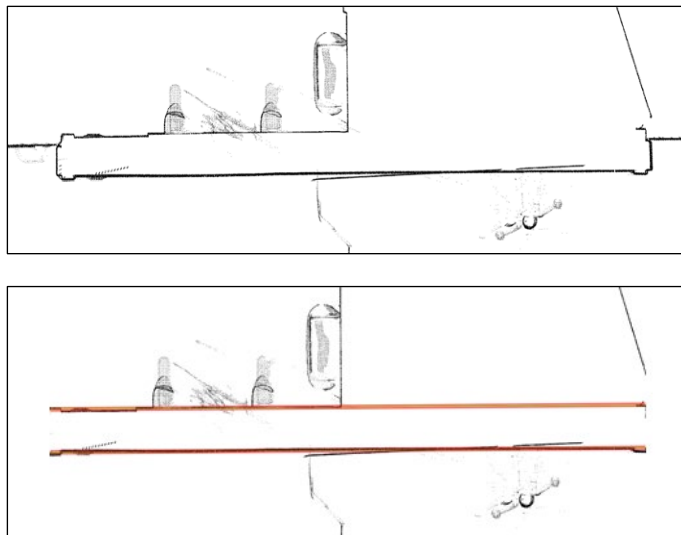
Nekateri predmeti v času skeniranja prekrivajo stene in jih tudi v postopku obdelave ni mogoče te težave rešiti v celoti. Na spodnjih slikah lahko vidimo, kako omare z vrati ustvarjajo prekinitve oblaka točk. V tem primeru smo za umestitev stene v prostor določili dve točki na obeh stran. Algoritem poveže steno tudi na mestih prekinitvev, s tem pa v programu dobimo neprekinjeno steno. Na spodnji sliki je z oranžno barvo prikazan potek stene.



Slika 34 Prikaz detekcije in umestitve stene na pripravljeno tlorisno podlago – primer 1

Primer 2 – Prekinitev zaradi sence pri laserskem posnetku

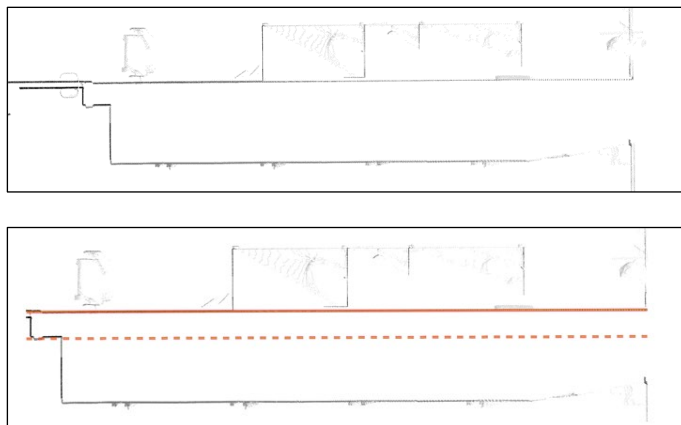
V drugem primeru so v času skeniranja vrata zakrila polovico stene in ustvarila senco v oblaku točk. Na podlagi situacije vemo, da se stena nadaljuje tudi na mestu, kjer točke niso bile posnete. Algoritem poveže steno v zaključeno celoto, njeno debelino pa določi na podlagi znanega dela, ki je v celoti viden na levi strani. Na spodnji sliki je z oranžno barvo prikazan potek stene.



Slika 35 Prikaz detekcije in umestitve stene na pripravljeno tlorisno podlago - primer 2

Primer 3 – Zaznava na osnovi ene vidne stranice

Ko steno prekrivajo objekti po njeni celi dolžini in jih zaradi njihove konfiguracije v času skeniranja ni potrebno premikati, lahko uporabimo algoritem zaznave na osnovi ene vidne stranice. V tem primeru smo za določitev smeri in debeline uporabili kratek, neprikrit del stene na levi stran. Polna oranžna črta na spodnji sliki prikazuje stranico, na podlagi katere je določena stena, črtkana črta pa približek na osnovi vidnega dela na levi strani.

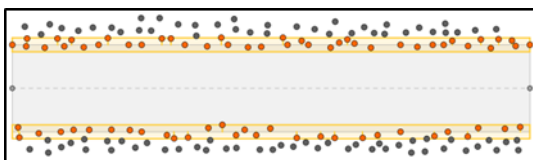


Slika 36 Prikaz detekcije in umestitve stene na pripravljeno tlorisno podlago - primer 3

3.3.2 Analiza odstopanja modela in oblaka točk

Na podlagi analize odstopanja lahko uporabnik prepozna različne kategorije napak pri modeliranju, ki jih za lažje razumevanje razdelimo v tri skupine: lokacijske, geometrijske in orientacijske napake.

Lokacijske napake se lahko pojavijo, če so vzdolž celotne stene konstantne vrednosti odstopanj. Če so na primer pozitivna odstopanja na eni strani stene in negativna na drugi strani, pomeni, da je model stene zamaknjen. V tem primeru bi jo bilo potrebno zamenjati ali prestaviti. Če je debelina modela stene napačna, je to geometrijska napaka. To napako lahko prepoznamo, če so vrednosti odstopanj na obeh straneh pozitivne, kar pomeni, da je model stene premajhnih dimenzij. Negativne vrednosti na obeh straneh pomenijo, da je model stene predebel. Rešitev za tako geometrijsko napako bi bila sprememba debeline. Napačno orientacijo je mogoče prepoznati, če se vrednosti odstopanja povečujejo ali zmanjšujejo z ene strani stene proti drugi. Model stene je v tem primeru zasukan in bi ga bilo potrebno rotirati.

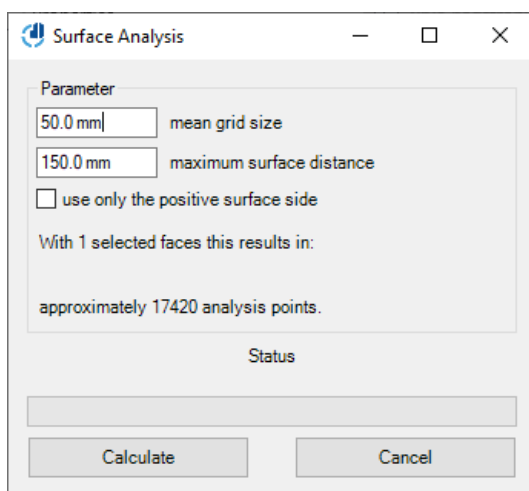


Slika 37 Grafični prikaz analize odstopanj točk glede na modelirano steno

Pomembno vlogo pri vsem tem igra tudi lokacijska kakovost oblaka točk. Kljub zadostni gostoti točk je potrebno zagotoviti kvaliteto in pravilno umestitev oblata točk v prostoru. Ta namreč ne sme biti zasukan v katerikoli smeri ali kako drugače popačen.

Pregled pojavnega okna

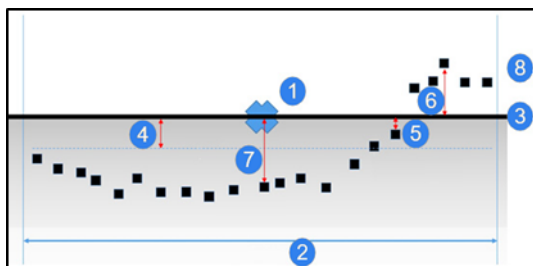
Vtičnik za analizo odstopanja modela in oblaka točk je enostaven za uporabo. Pred izračunom je potrebno izbrati stene in pripadajoči oblak točk. Program razdeli stenski element na celice in izračuna odstopanje za vsako od njih. Manjše celice omogočajo bolj natančen izračun, posledično pa daljši računski čas. Velikost celic je smiselno prilagoditi velikosti analiziranih elementov.



Slika 38 Pojavno okno za analizo odstopanj

1. Raster celic določa višino in širino vsake računske celice. Na ukrivljenih površinah se te vrednosti prilagodijo ukrivljenosti elementov.
2. Največja razdalja od računske površine določa pravokotni odmik od površine, v katerem se nahajajo računske točke za analizo odstopanja.
3. Izračun na pozitivni strani elementa omogoča izračun odstopanj samo na zunanji površini.
4. Okvirno število točk za izračun.

Vizualizacija različnih tipov analiz

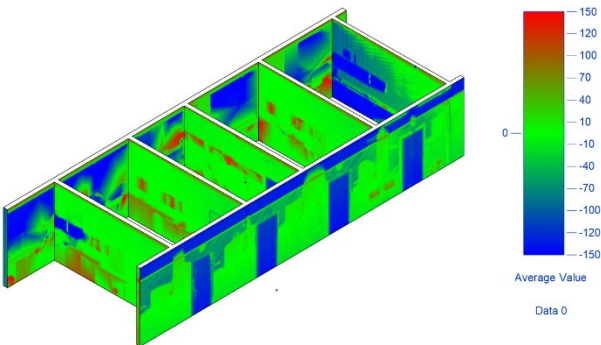
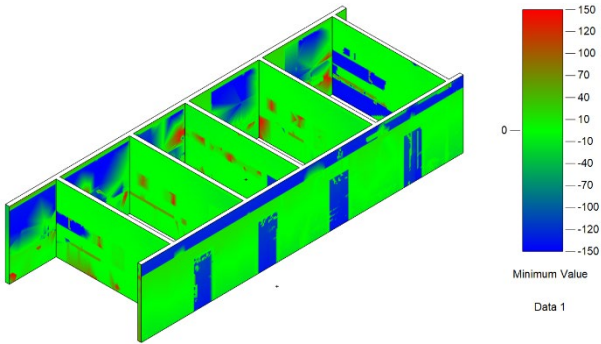
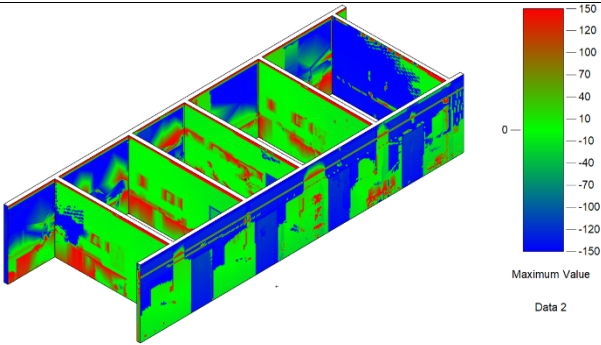


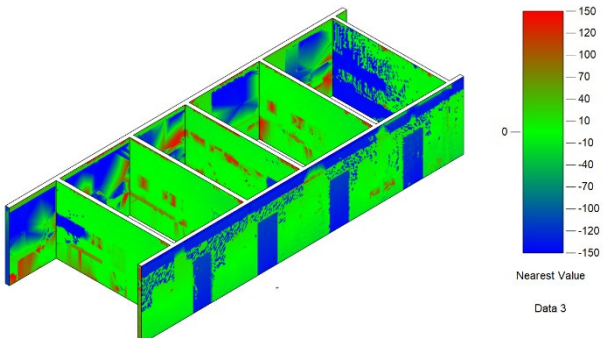
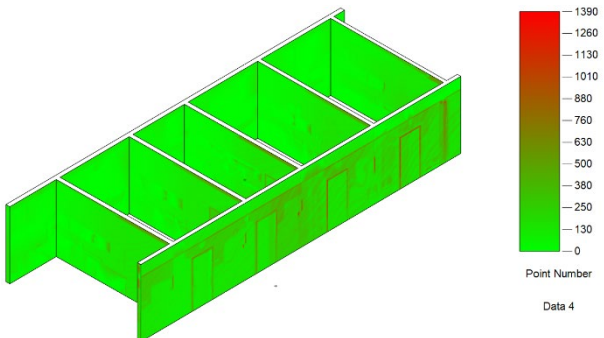
Slika 39 Grafični prikaz različnih vrst analiz odstopanj (povzeto po FARO, 2019)

1. Sredina računske celice v rastru.
2. Dimenzija celice v rastru.
3. Računski rob elementa.
4. Srednja vrednost odstopanja predstavlja povprečno razdaljo med vsemi točkami oblaka točk znotraj posamezne računske celice.
5. Minimalna vrednost odstopanja je najmanjše odstopanje znotraj posamezne računske celice. Pozitivne vrednosti ležijo zunaj elementa, negativne pa znotraj.
6. Največja vrednost odstopanja predstavlja največje odstopanje znotraj posamezne računske celice. Pozitivne vrednosti ležijo zunaj elementa, negativne pa znotraj.
7. Najbližja vrednost je pravokotna razdalja med središčem posamezne računske celice in točko, ki se nahaja najbližje središču.
8. Gostota točk je število točk v znotraj posamezne računske celice.

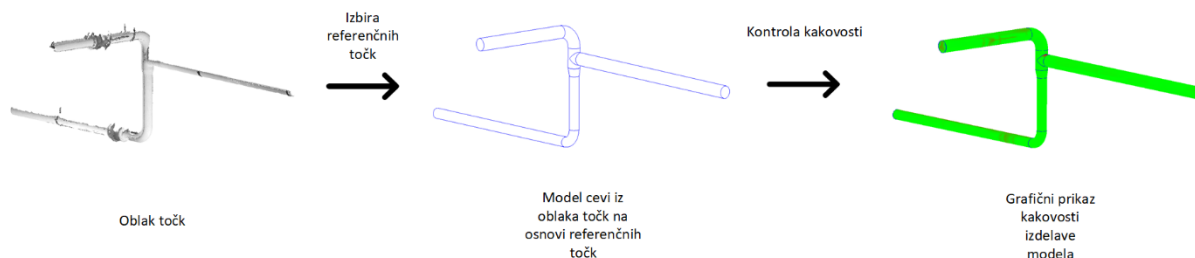
V razpredelnici na naslednji strani so različni grafični prikazi modeliranega nadstropja. Na desni strani posamezne slike je legenda natančnosti modela v primerjavi z referenčnim oblakom točk. Različni odtenki rdeče barve v tem primeru predstavljajo pozitivno odstopanje v vrednosti 70 in 150 mm med modelom in oblakom točk. Različni odtenki zelene prikazujejo odstopanje v vrednosti med 70 in -70 mm. Svetlo zelena barva je vrednost odstopanja okrog 0 mm. Na drugi strani odtenki modre predstavljajo odstopanja med -70 in -150 mm.

Preglednica 4 Prikaz različnih vrst analiz, ki so mogoče znotraj programa As – Built (lasten vir)

Analiza	Grafični prikaz
<p>Povprečna vrednost</p> <p>Velik del sten je v svetlo zeleni barvi, kar prikazuje minimalne vrednosti odstopanja sten od oblaka točk znotraj nekaj milimetrov. Na mestih predmetov v prostoru so barve rdeče, na mestih odprtih pa temno modre. Te deli so pomanjkljivosti oblaka točk v času skeniranja, ki se jim ni mogoče izogniti in ne predstavljajo dejanskih napak v modelu.</p>	
<p>Minimalna vrednost</p> <p>Desna slika prikazuje minimalno razdaljo med najbližjo točko oblaka točk in računsko celico znotraj modela. Večina sten je v svetlo zeleni barvi, kar pomeni, da so točke zelo blizu stene. V kombinaciji s prikazom povprečnih vrednosti predpostavimo milimetrsko natančnost modela v primerjavi z oblakom točk. Modre lise so odprtine, ki ponovno predstavljajo pomanjkljivosti podatkov.</p>	
<p>Maksimalna vrednost</p> <p>Maksimalna vrednost predstavlja razdaljo med računsko celico modela in najbolj oddaljeno točko znotraj te računske celice. Večina sten je ponovno v zeleni barvi, kar pomeni, da tudi najbolj oddaljene referenčne točke niso oddaljene za več kot nekaj milimetrov od modela. Rdeče barve ponovno predstavljajo anomalije zaradi pohištva, modre pa zaradi odprtih.</p>	

<p>Najbližja vrednost</p> <p>Točka, ki je pravokotno najbližje sredini računske celice je z barvami prikazana v desni sliki. Ker so odstopanja med oblakom točk in modelom majhna, je barvna analiza podobna analizi minimalnih vrednosti. Model je obarvan zeleno, odstopanja pa so le nekaj milimetrov.</p>	
<p>Število točk</p> <p>Število točk predstavlja gostoto točk znotraj ene računske celice. V našem primeru se vrednosti gibajo med 100 in 200. Razlog za tako število je lahko redek oblak točk ali pa večja površina računskih celic. V našem primeru smo za izračun uporabili celice z rastrom 50 x 50 mm. Velikost rastra je smiselno prilagoditi dimenzijam obravnavnega oblaka točk.</p>	

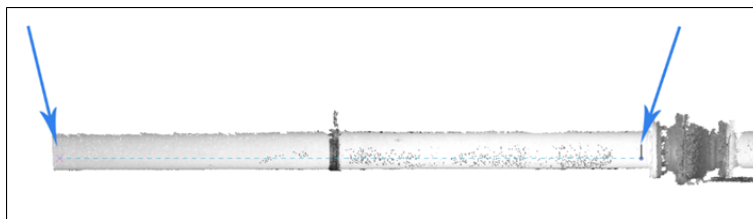
3.4 Avtomatizirano modeliranje cevi



Slika 40 Ilustracija postopka avtomatiziranega modeliranja cevi znotraj tega poglavja.

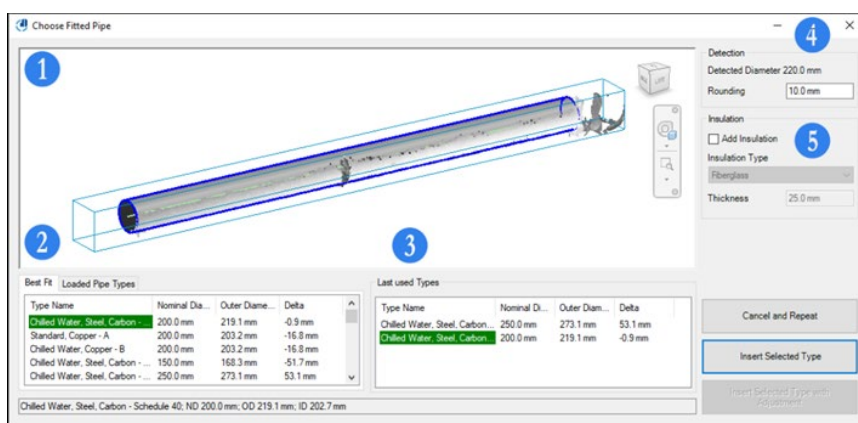
3.4.1 Izbira točk

Pri modeliranju cevi v prostoru ne potrebujemo predpriprave kot pri modeliranju sten v prejšnjem poglavju. Dovolj je, če obravnavane cevi izoliramo od ostalih objektov. Cevi zaradi medsebojnega prekrivanja, vertikalnih in horizontalnih potekov, težko prikažemo v 2D in jih zato obravnavamo kot 3D prostorski problem. V zavijku As Built – Pipe fitting se odpre pojavno okno, kjer določimo tip cevi in nadstropje, v katerem se modelirana cev nahaja. Za tem izberemo začetno in končno točko cevi. Program na osnovi izbranih točk določi potek cevi na referenčnem oblaku točk, kot je prikazano na spodnji sliki. Algoritem na podlagi oboda točk določi zunanji premer cevi in jo prostorsko umesti. Oblak točk v veliko primerih ni idealen, ker se cevi nahajajo v vogalih sten in so tesno ob stenah ali pa se zaradi medsebojnega prepletanja prekrivajo. Z laserskim skenerjem na terenu skoraj nikoli ne dobimo popolnih podatkov o ceveh, večinoma samo njihove dele. Pomanjkanje podatkov je pri modeliranju potrebno pametno izkoristiti, če želimo doseči dobre rezultate.



Slika 41 Izbira referenčnih točk na oblaku točk za umestitev cevi

Pregled pojavnega okna



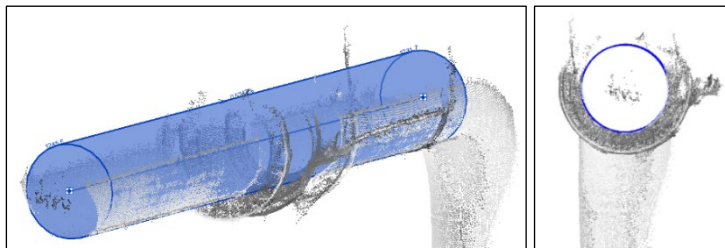
Slika 42 Pojavno okno za modeliranje cevi

1. Okno za prikaz cevi, ki prikazuje potek umeščene cevi glede na oblak točk v prostoru.
2. Okno za izbiro vrste cevi s stolpcema za prikaz premera in odstopanja. V prvem stolpcu je naveden premer osnovnega Revit elementa, v drugem pa razlika med dimenzijo osnovnega elementa iz knjižnice in izbranega.
3. Prikaz izbrane vrste cevi iz knjižnice programa Revit.
4. Nastavitev zaokrožitve zunanje premera cevi v mm.
5. Možnost dodajanja različnih tipov izolacije okrog cevi in njihove debeline.

Primer 1 – Polovica cevi

Če se cev nahaja pod stropom, njen zgornji del pa zaradi lokacije zajema podatkov ni viden, lahko program na osnovi dostopne polovice preračuna radij in dokonča cev ter jo natančno umesti v prostor. Na spodnjih slikah lahko vidimo natančnost umeščenega modela cevi znotraj nepopolnega referenčnega oblaka točk. Na nekaterih mestih je prisoten minimalen šum, ki pa v tem primeru ni dovolj izrazit, da bi zmotil algoritem in pokvaril natančnost zaznave. V več primerih smo z nepopolnimi podatki uspešno zaključili modele cevi.

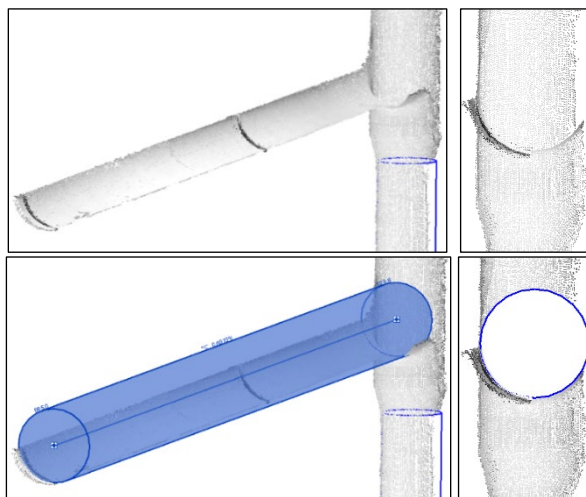




Slika 43 Prikaz modeliranja cevi na polovici znanih točk

Primer 2 – Četrtnina cevi

Če se cev nahaja v vogalu, lahko z laserskim skenerjem zajamemo zgolj četrtnino njenega premera. Z izbiro dveh točk na referenčnem oblaku točk program na podlagi dostopnih podatkov natančno določi in umesti model cevi v prostor. Podobno kot v prvem primeru smo kljub nepopolnim podatkom element uspešno umestili v prostor. Na spodnjih slikah je prikazan model cevi znotraj oblaka točk.



Slika 44 Prikaz modeliranja cevi na četrtnini znanih točk

Primer 3 – Šum

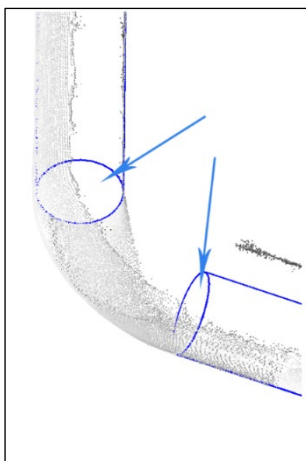
V tem primeru je bil oblak točk zajet z drugim laserskim skenerjem. V prostoru se pod stropom nahajata dve izolirani cevi, ki se večkrat kolenčno zlomita. Šum v tem primeru programu onemogoči določitev premera in lokacije cevi. Rešitev za to je ročno modeliranje, ki pa ni tako hitro in natančno kot avtomatizirano.



Slika 45 Prikaz dveh cevi v prostoru. Na levi sliki šuma ni videti. Na desni je prikazan prečni prerez cevi, kjer je šum očiten.

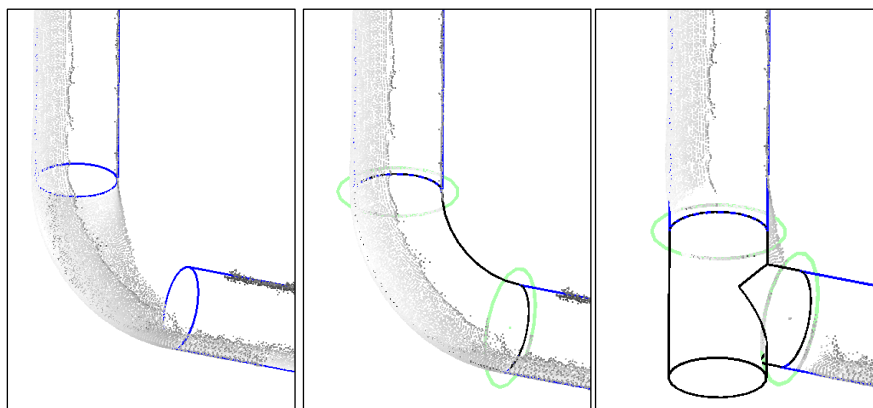
3.4.2 Povezave med elementi

Cevi v prostoru je za dokončanje potrebno povezati v smiselno celoto, ki bi v programu delovala brez napak. Za to uporabimo ukaz As Built – Place pipe fitting. Ko prvič zaženemo ta ukaz v projektu, program analizira vse družine naloženih cevi. Analiza določi spremenljive parametre v družinah, ki jih je mogoče po tem uporabiti pri povezavah. Rezultati analize se shranijo v projektu. Ob uvozu dodatnih družin, ki niso bile vključene v prvotno analizo, program ponovno analizira dodatne cevi. Za povezavo ločenih delov je potrebno izbrati proste konce cevi, ki jih želimo povezati med sabo. Izbrani konci cevi se obarvajo, s tem pa se vizualno ločijo od ostalih cevi. Na spodnji sliki sta s puščicama označena izbrana konca cevi, ki ju želimo povezati med seboj.



Slika 46 Izbira prostih robov, ki jih želimo medsebojno spojiti

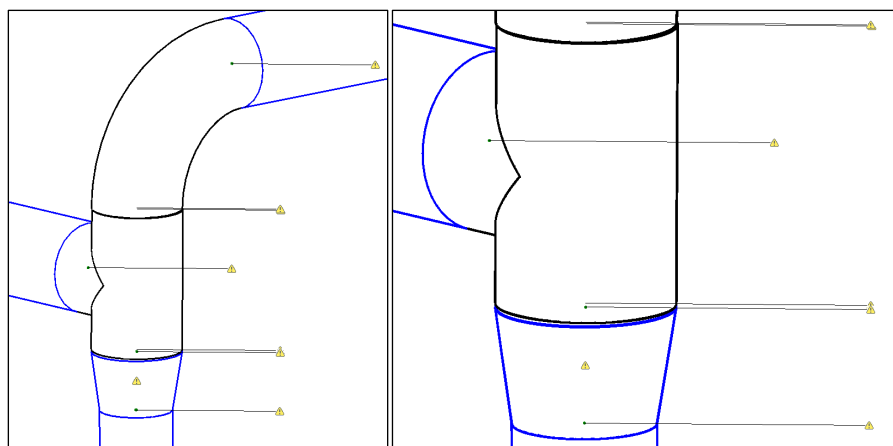
Na podlagi vhodnih točk, ki ležita pravokotno ena na drugo, program izbere kolenski kos s prilegajočim se radijem, ki se natančno prilega oblaku točk. Druga možnost, ki jo program na osnovi vhodnih parametrov ponudi, je T kos, ki pa za obravnavano situacijo ni primeren. Glede na vhodne podatke nam program ponudi kolenski kos kot najbolj primerno povezavo. Postopek lahko z izbiro prostih koncev cevi ponavljamo še za ostale nepovezane dele.



Slika 47 Izbira prave povezave med cevmi. Na drugi sliki je pravilni kolenčni kos, na tretji nepravilni T kos

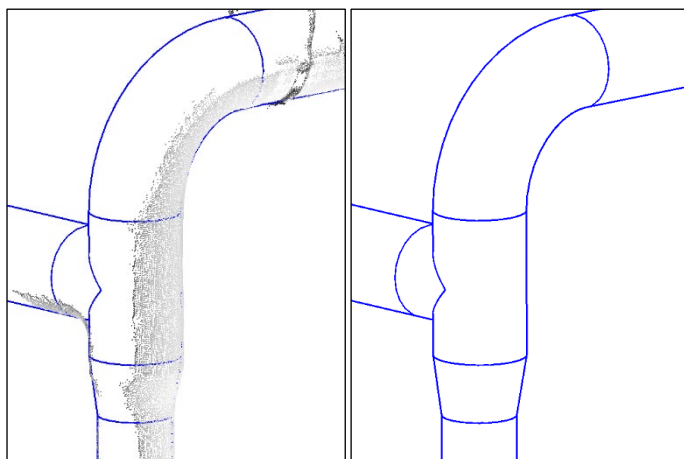
3.4.3 Analiza odstopanja

Namen algoritmov v prejšnjem podpoglavju je iskanje in modeliranje povezav, da so čim bolj podobne oblaku točk, iz katerega izhajamo. Po končanem modeliranju pa model v večini primerov še ne izpolnjuje zahtevanih omejitev skladnosti. Ker vsako komponento modeliramo posebej, se pokažejo manjša odstopanja in neskladnosti, saj komponente niso povezane v zaključeno celoto. Do tega pride, če so komponente postavljene pod napačnimi koti, njihovi deli štrlijo ali pa imajo med seboj vrzeli. Na to nas opozori Revit, ki te nedoslednosti označi. Napačni stiki so označeni s klicaji in različnimi barvami, kjer črna barva predstavlja nepopoln in nepovezan stik. Spodnja slika prikazuje primer nepopolnega kolenčnega stika in T kosa z opozorili.



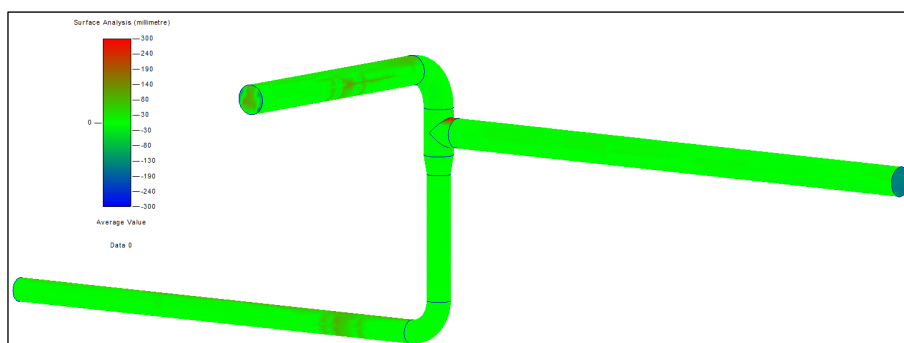
Slika 48 Prikaz nepravilnosti na spojih, ki je potrebno odpraviti

Algoritem za korekcijo spremeni položaj osi cevi. Osi minimalno premakne in zasuka povezave med cevmi, s tem pa ustvari natančno povezan model. Iteracija se zaustavi, ko je izpolnjeno vsaj eno od dveh meril zaustavitve. Prvi pogoj je dosežena najmanjša tolerančna vrednost med osjo spoja in cevi, ki znaša $0,01^\circ$. Drugi pogoj pa je preseženih 10000 iteracij na posameznem spoju. Če rezultati po teh pogojih niso doseženi, je potrebno nekoliko zmanjšati natančnost iteracijskih parametrov, da dosežemo povezavo posameznih elementov v zaključeno celoto. Rezultati so zaradi tega nekoliko slabši, vendar še vedno sprejemljivi. Popolni stiki med cevmi so pomembni zaradi analize toka skozi njih.



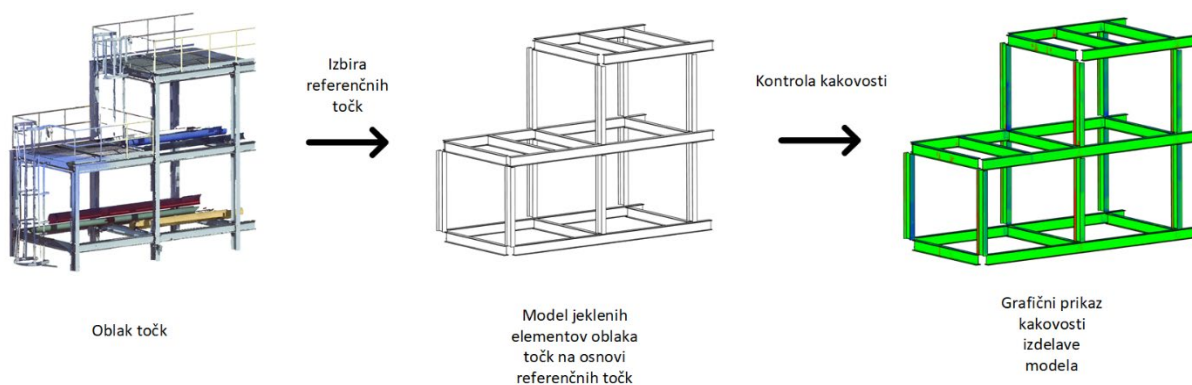
Slika 49 Prikaz pravilno modeliranih in povezanih cevi. Levo je prikaz z oblakom točk v ozadju, desno brez

Tako kot pri stenah v prejšnjem podpoglavju lahko tudi pri ceveh in njihovih stikih določimo natančnost modela v primerjavi z referenčnim oblakom točk. Ideja in metoda izračuna sta enaki kot pri stenah. Raster računskih celic se v tem primeru prilagodi valjasti obliki cevi. Na spodnji sliki so s svetlo zeleno barvo so označena mesta na elementu, ki se od oblaka točk razlikujejo za nekaj milimetrov. Kljub nepopolnim oblakom točk, ki vsebujejo nekaj senc in šuma, lahko pripravimo BIM modele z maksimalno natančnostjo, ki jo lahko v celotnem procesu nadziramo.



Slika 50 Prikaz odstopanja modeliranih cevi od osnovnega oblaka točk

3.5 Modeliranje nosilcev in stebrov



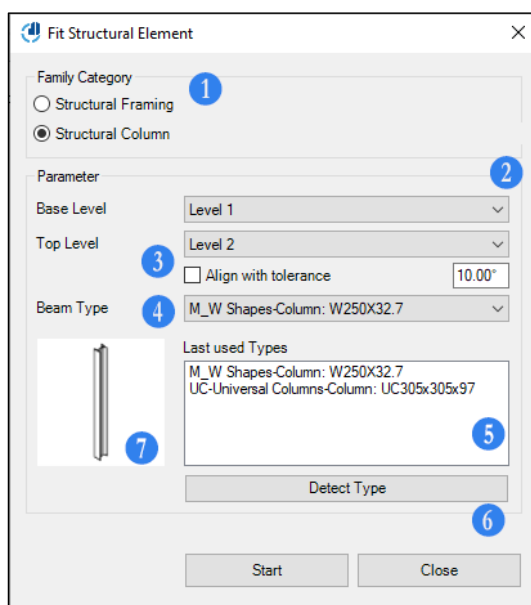
Slika 51 Ilustracija postopka avtomatiziranega modeliranja jeklenih elementov znotraj tega poglavja.

3.5.1 Izbira točk

Delotok pri modeliranju stebrov ali nosilcev je podoben kot pri modeliranju cevi v prejšnjem poglavju. Tako kot pri ceveh tudi pri stebrih in nosilcih pred začetkom ne potrebujemo pripravljene 2D podlage. Dovolj je, da obravnavane dele oblakov točk, na katerih želimo modelirati elemente, obrežemo in začasno izoliramo iz okolice, da ne motijo prostorske predstave uporabnika in algoritma za modeliranje.

V zavihku As-Built – Fit structural element se odpre pojavno okno, kjer v prvem koraku izberemo tip elementa, ki ga nameravamo modelirati. Na podlagi izbranega tipa elementa se temu primerno prilagodijo nastavitve v pojavnem oknu. Pri izbiri stebra se pojavi dodatna možnost izbire spodnjega in zgornjega nivoja stebra. Pri izbiri nosilca lahko določimo samo spodnji nivo. Za modeliranje elementa izberemo dve referenčni točki na oblaku točk, ki predstavljata začetek in konec elementa. Program na podlagi prečnega prereza oblaka točk poišče v knjižnici tisti element, ki najbolj ustreza prečnemu prerezu. Po končani primerjavi se prikaže pojavno okno z najbolj primernim elementom glede na izbrane točke v oblaku točk.

Pregled pojavnega okna

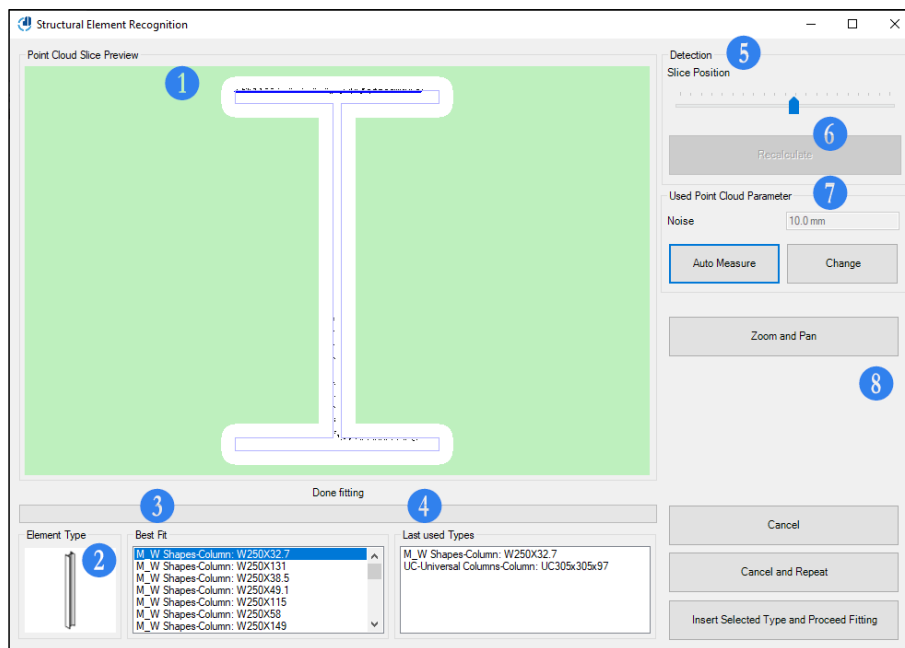


Slika 52 Pojavno okno za modeliranje stebrov in nosilcev

1. Izbira tipa elementa – nosilec ali steber.
2. Izbira referenčnih ravnin za omejitve elementa.
3. Toleranca, znotraj katere je umestitev elementa še mogoča. Večja kot je vrednost tolerančice, manj natančen je model.
4. Ročna izbira vrste elementa, če ga poznamo.
5. Seznam predhodno uporabljenih elementov.
6. Dodatno podokno za avtomatično določitev tipa elementa na podlagi prereza oblaka točk
7. Predogled izbranega elementa.

Pregled pojavnega podokna

Ko modeliranega elementa ne poznamo, ga lahko določimo iz prečnega prereza oblaka točk. Orodje je praktično, ker omogoča natančno določitev modeliranja vrst jeklenih profilov (H profil, I profil, C profil itd.) in betonskih stebrov.

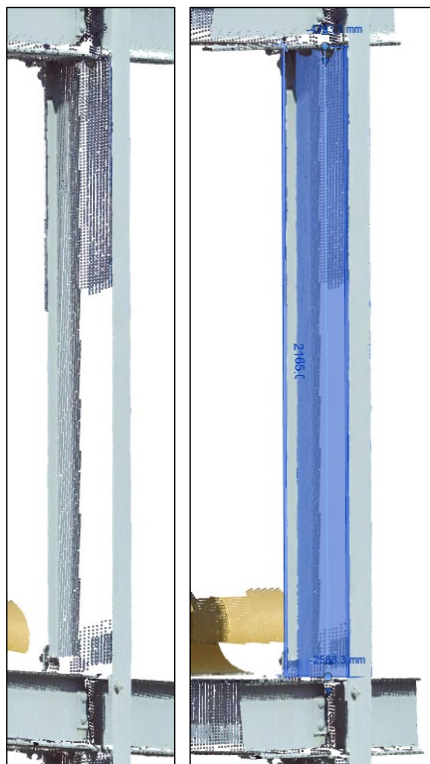


Slika 53 Pojavno okno za izbiro pravega elementa na osnovi prečnega prereza

1. Prikaz zaznanega elementa na osnovi prečnega prereza oblaka točk. S temno modro črto je označena stranica, na kateri sta bili izbrani referenčni točki. Zazan element je označen s svetlo modro barvo. Z belo barvo je označeno sprejemljivo območje šuma. V tem podoknu lahko uporabnik prosto izbriše šum ali odvečne točke, ki motijo zaznavo.
2. Predogled izbranega elementa iz Revit knjižnice.
3. Prikaz elementa, ki se najboljše prilega. Elementi so sistematično razdeljeni glede na kakovost prileganja.
4. Seznam predhodno uporabljenih elementov.
5. Prepoznavanje vzorca se izvede na izrezu oblaka točk med obema izbranimi točkama. Z drsnikom se ta lokacija lahko spreminja vzdolž osi elementa in s tem izboljšamo natančnost določitve. To lahko izniči točke ali popravi primer, ko je bilo zaradi senčenja odkritih premalo točk.
6. Povrnitev prvotnih nastavitev.
7. Določitev mejnih vrednosti šuma.
8. Prikaz lokacije elementa v modelu.

Primer 1 – Manjkajoča stojina

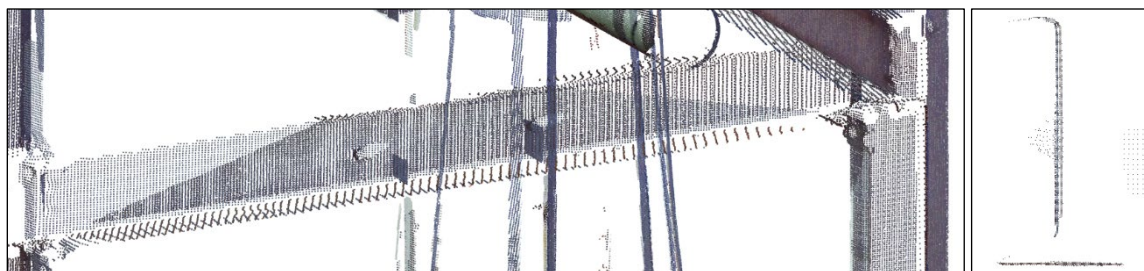
V primeru, kjer so podatki znotraj oblaka točk pomanjkljivi, si lahko pomagamo z delom elementa, kjer te podatke imamo. Tip jeklenega profila in dimenzije so vzdolž njegove osi nespremenljive. Če poznamo širino obeh pasnic in njuno medsebojno razdaljo, lahko določimo dimenzije stojine. Na spodnji sliki je prikazan primer jeklenega I profila, kjer zaradi ovir pri zajemu manjka polovica stojine.



Slika 54 Prikaz jeklenega profila z manjkajočim delom stojine (levo) in modeliran element na osnovi oblaka točk (desno)

Primer 2 – Manjkajoči del pasnice

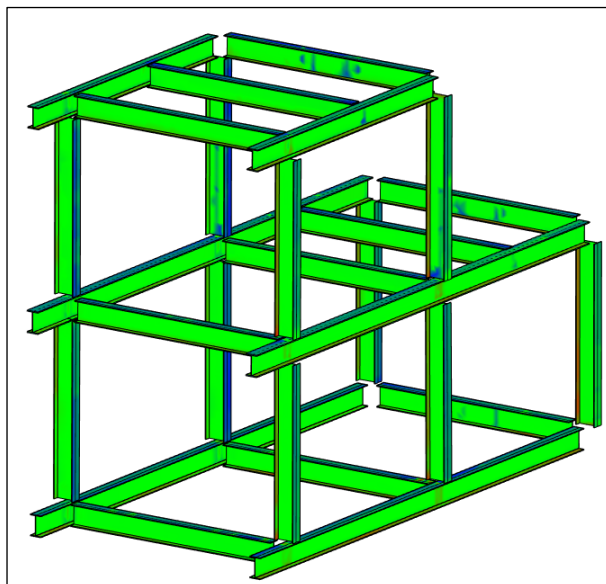
Zaradi oddaljenosti in kota med virom zajema in jeklenim profilom je kakovost podatkov zato slabša. Točke so veliko redkejše, zaradi ovir pa ni zajet del pasnice. Dimenzij zgornje pasnice ne poznamo, poznamo pa dimenzije stojine in spodnje pasnice. Program z znanimi dimenzijami določa še manjkajoče, če so zrcalne znanim. Na osnovi dveh izbranih točk program določi elementa njegove dimenzije in lokacijo v prostoru.



Slika 55 Prikaz jeklenega profila z manjkajočo pasnico (levo) in prečni prerez (desno)

3.5.2 Analiza odstopanja

Na spodnji sliki lahko vidimo prikaz analize natančnosti umestitve in odstopanj elementov znotraj oblaka točk. S svetlo zeleno barvo so označeni deli elementov, kjer je natančnost znotraj nekaj milimetrov. Deli, obarvani z modro barvo, so rezultat pomanjkljivih podatkov v oblakih točk.



Slika 56 Prikaz odstopanja modeliranih jeklenih profilov od osnovnega oblaka točk

4 ZAKLJUČEK

V poglavju 3.3, znotraj praktičnega dela smo preverjali modeliranje sten na osnovi referenčnih oblakov točk. Zgradbe in stene znotraj njih niso vedno medsebojno pravokotne in vzporedne, kar je pri modeliranju dejanskega stanja treba upoštevati. V času zajema podatkov je v zgradbah prisotnih veliko ovir, ki povzročajo sence in šum. Prednost avtomatiziranega vmesnika je algoritem, ki na osnovi referenčnih uporabniških točk določi dimenzije sten in jih umesti znotraj oblaka točk. V prvem primeru znotraj poglavja 3.3 smo uspešno modelirali stene, ki so bile v večji meri zakrite s sencami drugih predmetov. V drugem primeru znotraj poglavja 3.3 smo modelirali stene, kjer je bila zajeta samo ena njihova stranica. Algoritem na podlagi znanih podatkov predvidi potek sten in jih zaključiti. Avtomatiziran način dela omogoča večjo hitrost in natančnost dela kot ročno modeliranje, kot so s podobnimi primeri potrdili tudi pri ClearEdge3D (ClearEdge3D Case Studies, 2021). Kljub uporabi drugega programa so s podobnim delotokom dosegli 60 % hitrejša končna rezultata kot pri konvencionalnem ročnem modeliranju. Kakovost modela smo v procesu modeliranja kontrolirali, na koncu sklopa pa smo rezultate tudi grafično prikazali. Dimenzije modelov sten in njihova prostorska lokacija so bili določeni z milimetro natančnostjo v primerjavi z referenčnim oblakom točk. Glavne omejitve obstoječih algoritmov so se pokazale v obliki nepravilnih in nagnjenih sten ter sten s spremenljivo debelino, ki jih programu ni uspelo prepoznati in umestiti.

Drugi sklop znotraj poglavja 3.4 je obsegal modeliranje cevi in njihovih spojev. Cevi so v prostorih velikokrat medsebojno prepletene in se medsebojno prekrivajo, zato pri terenskem zajemu ni mogoče zajeti vsega. Glavna prednost programskega vmesnika je algoritem, ki interpolira premer cevi in jih natančno umesti znotraj znanih podatkov. S pomočjo programa smo določili premere cevi in jih natančno umestili v prostor tudi v primerih, kjer sta bili zajeti samo polovica ali pa četrtnina cevi. S tako programsko podporo se proces precej skrajša. Na terenu ni treba posvetiti toliko časa zajemu vseh podatkov, skrajša pa se tudi faza modeliranja. Na primerih se je pokazalo, da hitrost modeliranja ne zmanjšuje natančnosti modela. Poleg modeliranja program omogoča kontrolo kakovosti modela v primerjavi z referenčnim oblakom točk in grafično prikaže različna odstopanja. Prostorska natančnost in dimenzije modelov so bile določene z milimetro natančnostjo v primerjavi z referenčnim oblakom točk. Analizo natančnosti modeliranja smo na koncu sklopa tudi grafično prikazali. Težava se je pokazala v obliki šuma v zajetih podatkih. Šum je v naših primerih popačil obliko cevi, zaradi česar program ni mogel zaznati in določiti cevi.

Modeliranje stebrov in nosilcev je potekalo v tretjem sklopu znotraj poglavja 3.5. V času zajema podatkov se tako kot cevi tudi stebri medsebojno prekrivajo. Pri betonskih stebrih sta velikokrat zajeti dve stranici, pri jeklenih nosilcih pa samo stojine ali pasnice. Z izbiro referenčnih točk znotraj oblaka točk je program analiziral oblak točk in določil vrste stebrov in nosilcev. V prvem primeru smo s programom določili dimenzije stebrov, kjer sta bili zajeti stojini, manjkala pa je pasnica. V drugem primeru je bila situacija podobna, znani sta bili ena pasnica in stojina. Z avtomatizacijo je tok dela hitrejši, določitev elementov pa bolj natančna. Kakovost modela smo v procesu kontrolirali in grafično pokazali na koncu sklopa. Vsi elementi modela so bili določeni z milimetro natančnostjo glede na referenčni oblak točk.

V vseh sklopih smo potrdili določeno stopnjo avtomatizacije, ki uporabnikom omogoča hitrejšo in bolj natančno zaznavo, kar potrjujejo tudi podobni primeri v literaturi. Kot je bilo omenjeno že v prvem odstavku tega poglavja, so hitrejši tok dela pokazali pri ClearEdge3D (ClearEdge3D, 2021), kjer so ocenili povprečno 60 % večjo hitrost modeliranja na podobnih primerih. Natančnost avtomatiziranega procesa pretvorbe oblakov točk BIM v primerjavi z ročnim procesom so na podobnih primerih dokazali

tudi številni raziskovalci v preteklih letih, ki so primerjali kvalitativne rezultate istih elementov, modeliranih ročno in avtomatsko (Patraucean in sod., 2015, Qu in Sun, 2015, Macher in sod., 2017). Trenutni postopek zaradi uporabniške interakcije težko označimo kot avtomatiziran, kljub temu da ga tako označujejo proizvajalci programske opreme. Trenutni proces dela je polavtomatski, kjer uporabnik z izbiro vrste elementov in referenčnih točk znotraj oblaka sproži avtomatsko obdelavo. Popolna avtomatizacija, kjer bi program brez interakcije uporabnika iz oblaka točk semantično prepoznal določene dele zgradb, trenutno ni mogoča v celoti. Najbližje temu je programska oprema EdgeWise podjetja ClearEdge3D, ki omogoča avtomatsko modeliranje stenskega skeleta zgradb. Natančnost tega je še vedno slaba, rezultat pa bolj predstavlja shematski model zgradbe kot pa podroben BIM model. Vodilni proizvajalci programske opreme (Autodesk in Nemetschek group) trenutno kažejo počasen razvoj vmesnikov za delo z oblaki točk znotraj svojih programov na področju gradbeništva. Pot do popolne avtomatizacije se zdi najbolj obetavna z algoritmi globokega učenja, ki bi na osnovi obsežnih podatkovnih baz prepoznavali posamezne elemente znotraj oblaka točk, jih prostorsko umestili, povezali v celote in v toku celotnega procesa izvajali kontrolo kakovosti. Kljub napredku algoritmov globokega učenja pa teoretična meja črpanja informacij iz oblakov točk ostaja. Tako na primer kljub novim algoritmom ne bo mogoče črpati določenih informacij iz oblaka točk, ko je na primer starost stene zgolj na podlagi primitivnih podatkov znotraj oblaka točk.

Za lažjo in bolj strukturirano predstavo je v spodnji preglednici prikazana SWOT (Strengths, Weaknesses, Opportunities, Threats) analiza z glavnimi rezultati.

Preglednica 5 SWOT analiza glavnih elementov pretvorbe snovnega sveta v digitalni.

Strengths - prednosti	Weaknesses - slabosti
<ul style="list-style-type: none"> • Večja hitrost pri obdelavi podatkov in modeliranju • Večja natančnost pri obdelavi in modeliranju • Sprotna analiza kakovosti izdelanih modelov 	<ul style="list-style-type: none"> • Omejitve pri modeliranju ne tipičnih konstrukcijskih elementov • Kljub avtomatizaciji je še vedno prisotno veliko uporabniške interakcije • Slaba interdisciplinarna povezava (datotečni formati) • Pasivnost glavnih proizvajalcev programske opreme (Autodesk in Nemetschek) pri razvoju programov v smeri dela z oblaki točk
Opportunities – možnosti	Threats – nevarnosti
<ul style="list-style-type: none"> • Dodatno zmanjšanje uporabniške interakcije • Uporaba boljših algoritmov – globoko učenje • Programska oprema (programski paket), ki bi omogočala celoten delotok od zajema do končnega modela znotraj enega programa 	<ul style="list-style-type: none"> • Veliko število lastniških izmenjevalnih formatov, ki se z večanjem števila programskih in strojnih ponudnikov v zadnjih letih le še povečuje • Omejitve pri črpanju informacij iz oblakov točk (starost stene)

VIRI

- Abellán, A., Vilaplana, J.M., Martínez, J., 2006. Application of a long-range Terrestrial Laser Scanner to a detailed rockfall study at Vall de Núria (Eastern Pyrenees, Spain). *Engineering Geology* 88, 136–148.
- Aber, J.S., Marzolf, I., Ries, J.B., 2010. *Small-format aerial photography: principles, techniques and geoscience applications*, 1. ed. ed. Elsevier, Amsterdam.
- Antolin, I., 2015. Uporaba oblakov 3D-točk v gradbeništvu 43.
- Arslan, A.E., Kalkan, K., 2013. Comparison of working efficiency of terrestrial laser scanner in day and night conditions. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XL-7/W2, 19–21.
- Badenko, V., Fedotov, A., Zotov, D., 2018. Extracting features from laser scanning point cloud. *SHS Web of Conf.* 44, 00013.
- Ballard, D.H., 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition* 13, 111–122.
- Barrile, V., Candela, G., Fotia, A., 2019. Point cloud segmentation using image processing techniques for structural analysis. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLII-2/W11, 187–193.
- Bassier, M., Mattheuwsen, L., Vergauwen, M., 2019. BIM reconstruction: automated procedural modeling from point cloud data. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLII-2/W17, 53–60.
- Blazek, J., 2015. *Computational fluid dynamics: principles and applications*, Third edition. ed. Butterworth-Heinemann, Amsterdam.
- Borrmann, D., Elseberg, J., Lingemann, K., Nüchter, A., 2011. The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design. *3D Res* 2, 3.
- Boulaassal, H., Landes, T., Grussenmeyer, P., Tarsha-Kurdi, F., 2007. Automatic segmentation of building facades using Terrestrial Laser Data 7.
- Boulch, A., Guerry, J., Le Saux, B., Audebert, N., 2018. SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers & Graphics* 71, 189–198.
- Boulch, A., Marlet, R., 2012. Fast and Robust Normal Estimation for Point Clouds with Sharp Features. *Computer Graphics Forum* 31, 1765–1774.
- Boyko, A., Funkhouser, T., 2011. Extracting roads from dense point clouds in large scale urban environment. *ISPRS Journal of Photogrammetry and Remote Sensing* 66, S2–S12.
- Buckley, S.J., Kurz, T.H., Schneider, D., 2012. The benefits of terrestrial laser scanning and hyperspectral data fusion products. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XXXIX-B7, 541–546.
- Camurri, M., Vezzani, R., Cucchiara, R., 2014. 3D Hough transform for sphere recognition on point clouds: A systematic study and a new method proposal. *Machine Vision and Applications* 25, 1877–1891.

Castillo, E., Zhao, H., n.d. Point Cloud Segmentation via Constrained Nonlinear Least Squares Surface Normal Estimates 6.

Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, pp. 77–85.

ClearEdge3D, 2021. Case studies.

<https://www.clearedge3d.com/resources/#CaseStudies> (pridobljeno 16.4.2021)

CloudCompare, 2021. User manual.

<http://www.cloudcompare.org/doc/qCC/CloudCompare%20v2.6.1%20-%20User%20manual.pdf>
(pridobljeno 16.4.2021)

Cheok, G.S., Lytle, A.M., Saidi, K.S., 2008. ASTM E57 3D imaging systems committee: an update, in: Turner, M.D., Kamerman, G.W. (Eds.), . Presented at the SPIE Defense and Security Symposium, Orlando, FL, p. 69500J.

Deschud, J.-E., Goulette, F., n.d. A Fast and Accurate Plane Detection Algorithm for Large Noisy Point Clouds Using Filtered Normals and Voxel Growing 10.

Dong, X., Xu, Q., Huang, R., Liu, Q., Kieffer, D.S., 2020. Reconstruction of Surficial Rock Blocks by Means of Rock Structure Modelling of 3D TLS Point Clouds: The 2013 Long-Chang Rockfall. *Rock Mech Rock Eng* 53, 671–689.

Dong, Z., Yang, B., Hu, P., Scherer, S., 2018. An efficient global energy optimization approach for robust 3D plane segmentation of point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 137, 112–133.

Duda, R.O., Hart, P.E., 1972. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM* 15, 11–15.

Fernandes, L.A.F., Oliveira, M.M., 2008. Real-time line detection through an improved Hough transform voting scheme. *Pattern Recognition* 41, 299–314.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395.

Foster, S., Halbstein, D., 2014. Integrating 3D Modeling, Photogrammetry and Design, *SpringerBriefs in Computer Science*. Springer London, London.

Grilli, E., Menna, F., Remondino, F., 2017. A review of point clouds segmentation and classification algorithms. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLII-2/W3, 339–344.

Hadjimitsis, D.G., Themistocleous, K., Cuca, B., Agapiou, A., Lysandrou, V., Lasaponara, R., Masini, N., Schreier, G. (Eds.), 2020. Remote Sensing for Archaeology and Cultural Landscapes: Best Practices and Perspectives Across Europe and the Middle East, *Springer Remote Sensing/Photogrammetry*. Springer International Publishing, Cham.

Hast, A., Nysjö, J., Marchetti, A., 2013. Optimal RANSAC - Towards a Repeatable Algorithm for Finding the Optimal Set. *Journal of WSCG* 21, 10.

Heritage, G.L., Large, A.R.G. (Eds.), 2009. Laser scanning for the environmental sciences. Wiley-Blackwell, Chichester, UK ; Hoboken, NJ.

Huber, D., 2011. The ASTM E57 file format for 3D imaging data exchange, in: Beraldin, J.A., Cheok, G.S., McCarthy, M.B., Neuschaefer-Rube, U., Baskurt, A.M., McDowall, I.E., Dolinsky, M. (Eds.), . Presented at the IS&T/SPIE Electronic Imaging, San Francisco Airport, California, USA, p. 78640A.

Hunter, G., Pinkerton, H., Airey, R., Calvari, S., 2003. The application of a long-range laser scanner for monitoring volcanic activity on Mount Etna. *Journal of Volcanology and Geothermal Research* 123, 203–210.

Jemec, L., 2016. Izdelava modela konstrukcij BIM na osnovi oblaka točk.

Kaiser, A., Ybanez Zepeda, J.A., Boubekur, T., 2019. A Survey of Simple Geometric Primitives Detection Methods for Captured 3D Data. *Computer Graphics Forum* 38, 167–196.

Kastelic, M., 2010. Obdelava podatkov laserskega skeniranja v programu Geomagic na primeru Mislejevega portala.

Kharroubi, A., Hajji, R., Billen, R., Poux, F., 2019. Classification and integration of massive 3d points clouds in a virtual reality (vr) environment. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLII-2/W17, 165–171.

Kiryati, N., Eldar, Y., Bruckstein, A.M., 1991. A probabilistic Hough transform. *Pattern Recognition* 24, 303–316.

Kwadjo, D.T., Tchinda, E.N., Bobda, C., Menadjou, N., Fotsing, C., Nziengam, N., Nziengam, N., 2019. From PC2BIM: Automatic Model generation from Indoor Point Cloud, in: *Proceedings of the 13th International Conference on Distributed Smart Cameras*. Presented at the ICDSC 2019: 13th International Conference on Distributed Smart Cameras, ACM, Trento Italy, pp. 1–6.

Laefer, D.F., Koss, A., Pradhan, A., 2006. The Need for Baseline Data Characteristics for GIS-Based Disaster Management Systems. *J. Urban Plann. Dev.* 132, 115–119.

Leica CloudWorx, 2021. Case Studies.

<https://leica-geosystems.com/products/laser-scanners/software/leica-cloudworx> (pridobljeno 16.4.2021)

Lennon, R., 2017. Automatic Validation of As-Is and As-Generated IFC BIMs for Advanced Scan-to-BIM Methods 10.

Limberger, F.A., Oliveira, M.M., 2016. Real-Time Detection of Planar Regions in Unorganized Point Clouds 6.

Linder, W., 2016. *Digital Photogrammetry*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Macher, H., Chow, L., Fai, S., 2019. Automating the verification of heritage building information models created from point cloud data. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLII-2/W9, 455–460.

Malinverni, E.S., Pierdicca, R., Paolanti, M., Martini, M., Morbidoni, C., Matrone, F., Lingua, A., 2019. Deep learning for semantic segmentation of 3d point cloud. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLII-2/W15, 735–742.

Matas, J., Galambos, C., Kittler, J., 1998. Progressive Probabilistic Hough Transform, in: Proceedings of the British Machine Vision Conference 1998. Presented at the British Machine Vision Conference 1998, British Machine Vision Association, Southampton, p. 26.1-26.10.

Maturana, D., Scherer, S., 2015. VoxNet: A 3D Convolutional Neural Network for real-time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Presented at the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Hamburg, Germany, pp. 922–928.

Mills, J.P., Chandler, J.H., 2007. ISPRS Commission V Symposium: Image Engineering And Vision Metrology. Photogrammetric Record 22, 94–96.

Morsdorf, F., Meier, E., Kötz, B., Itten, K.I., Dobbertin, M., Allgöwer, B., 2004. LIDAR-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management. Remote Sensing of Environment 92, 353–362.

Mozetič, B., 2004. Terestrično 3d (trirazsežno) lasersko skeniranje. Geodetski vestnik 12 str.

Müller, C., Kohoutek, T.K., Theiler, P.W., 2012. Automated Site Plan Extraction from TLS Point Clouds 10.

Mura, C., Wyss, G., Pajarola, R., 2018. Robust normal estimation in unstructured 3D point clouds by selective normal space exploration. Vis Comput 34, 961–971.

Ni, H., Lin, X., Ning, X., Zhang, J., 2016. Edge Detection and Feature Line Tracing in 3D-Point Clouds by Analyzing Geometric Properties of Neighborhoods. Remote Sensing 8, 710.

Ning, X., Zhang, X., Wang, Y., Jaeger, M., 2009. Segmentation of architecture shape information from 3D point cloud, in: Proceedings of the 8th International Conference on Virtual Reality Continuum and Its Applications in Industry - VRCAI '09. Presented at the the 8th International Conference, ACM Press, Yokohama, Japan, p. 127.

O'Donnell, J., Truong-Hong, L., Boyle, N., Corry, E., Cao, J., Laefer, D.F., 2019. LiDAR point-cloud mapping of building façades for building energy performance simulation. Automation in Construction 107, 102905.

Oguchi, T., Hayakawa, Y., S., Wasklewicz, T., 2011. Geographic information systems in geomorphology, in: Developments in Earth Surface Processes. Elsevier, pp. 189–224.

Otepka, J., Ghuffar, S., Waldhauser, C., Hochreiter, R., Pfeifer, N., 2013. Georeferenced Point Clouds: A Survey of Features and Point Cloud Management. IJGI 2, 1038–1065.

Özdemir, E., Remondino, F., Golkar, A., 2019. Aerial point cloud classification with deep learning and machine learning algorithms. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-4/W18, 843–849.

Papon, J., Abramov, A., Schoeler, M., Worgotter, F., 2013. Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition. Presented at the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Portland, OR, USA, pp. 2027–2034.

Pavol Jozef Šafárik, Gallay, M., Kaňuk, J., Institute of Geography, Pavol Jozef Šafárik University in Košice, Slovakia, Hochmuth, Z., Institute of Geography, Pavol Jozef Šafárik University in Košice, Slovakia, Meneely, J., Queen's University Belfast, UK, Hofierka, J., Institute of Geography, Pavol Jozef Šafárik University in Košice, Slovakia, Sedlák, V., Institute of Geography, Pavol Jozef Šafárik University in Košice, Slovakia, 2015. Large-scale and high-resolution 3-D cave mapping by terrestrial laser scanning: a case study of the Domica Cave, Slovakia. *IJS* 44, 277–291.

Petrič, A., 2013. Vrednotenje in primerjava geodetskih merskih tehnologij lidar in 3d laserskega skeniranja. 30 str.

Petrie, G., Toth, C.K., 2009. Terrestrial Laser Scanners. *Topographic Laser Ranging and Scanning* 43.

Pfeifer, N., Briese, C., 2007. Geometrical aspects of airborne laser scanning and terrestrial laser scanning 9.

PointCab Software, 2021. Applications

<https://pointcab-software.com/en/applications/> (pridobljeno 16.4.2021)

Pu, S., Vosselman, G., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing* 64, 575–584.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space.

Rabbani, T., 2005. Efficient hough transform for automatic detection of cylinders in point clouds 6.

Recap, 2021. Features.

<https://www.autodesk.com/products/recap/features> (pridobljeno 16.4.2021)

Riegler, G., Ulusoy, A.O., Geiger, A., 2017. OctNet: Learning Deep 3D Representations at High Resolutions, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, pp. 6620–6629.

Rodríguez-Moreno, C., Reinoso-Gordo, J.F., Rivas-López, E., Gómez-Blanco, A., Ariza-López, F.J., Ariza-López, I., 2018. From point cloud to BIM: an integrated workflow for documentation, research and modelling of architectural heritage. *Survey Review* 50, 212–231.

Rossmann, J., Schluse, M., Bücken, A., Krahwinkler, P., 2007. Using airborne laser scanner data in forestry management: a novel approach to single tree delineation 6.

Rutzinger, M., Höfle, B., Oude Elberink, S., Vosselman, G., 2011. Feasibility of Facade Footprint Extraction from Mobile Laser Scanning Data. *pfg* 2011, 97–107.

Sampath, A., Jie Shan, 2010. Segmentation and Reconstruction of Polyhedral Building Roofs From Aerial Lidar Point Clouds. *IEEE Trans. Geosci. Remote Sensing* 48, 1554–1567.

San José Alonso, J.I., Martínez Rubio, J., Fernández Martín, J.J., García Fernández, J., 2012. Comparing time-of-flight and phase-shift. The survey of the royal pantheon in the Basilica of San Isidoro (León). *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XXXVIII-5/W16, 377–385.

Schmitt, M., Shahzad, M., Zhu, X.X., 2015. Reconstruction of individual trees from multi-aspect TomoSAR data. *Remote Sensing of Environment* 165, 175–185.

Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum* 26, 214–226.

Shan, J., Toth, C.K., 2018. *Topographic laser ranging and scanning: principles and processing*, Second edition. ed. Taylor & Francis, CRC Press, Boca Raton.

Smogavec, L., 2015. Uporabnost terestričnega laserskega skeniranja pri izdelavi geodetskega načrta. Dipl. nal. – UNI. Ljubljana, UL FGG, Odd. za geodezijo, Geodezija v inženirstvu. 100.

Spina, S., Debattista, K., Bugeja, K., Chalmers, A., 2011. Point Cloud Segmentation for Cultural Heritage Sites. *VAST: International Symposium on Virtual Reality* 8 pages.

Staiger, R., 2003. *Terrestrial Laser Scanning Technology, Systems and Applications* 10.

Števancec, T., 2016. *Ravninska Delaunayeva triangulacija*.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., 2007. Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data 6.

Thomson, C., 2016. From Point Cloud to Building Information Model: Capturing and Processing Survey Data Towards Automation for High Quality 3D Models to Aid a BIM Process.

Thomson, C., Boehm, J., 2015. Automatic Geometry Generation from Point Clouds for BIM. *Remote Sensing* 7, 11753–11775.

Tran, H., Khoshelham, K., 2019. A stochastic approach to automated reconstruction of 3d models of interior spaces from point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* IV-2/W5, 299–306.

Triglav Čekada, M., 2017. Fotogrametrični in lidarski oblaki točk. *Geogr. vestn.* 89.

Vo, A.-V., Truong-Hong, L., Laefer, D.F., Bertolotto, M., 2015. Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing* 104, 88–100.

Vosselman, G., Gorte, B.G.H., Sithole, G., Rabbani, T., n.d. Recognising structure in laser scanner point clouds 6.

Vosselman, G., Maas, H.-G. (Eds.), 2011. *Airborne and terrestrial laser scanning*, repr. ed. Whittles, Dunbeath.

Wang, Chen, Zhu, Liu, Li, Zheng, 2019. A Survey of Mobile Laser Scanning Applications and Key Techniques over Urban Areas. *Remote Sensing* 11, 1540.

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., Tong, X., 2017. O-CNN: octree-based convolutional neural networks for 3D shape analysis. *ACM Trans. Graph.* 36, 1–11.

Woo, H., Kang, E., Wang, S., Lee, K.H., 2002. A new segmentation method for point cloud data. *International Journal of Machine Tools and Manufacture* 42, 167–178.

Xie, Y., Tian, J., Zhu, X.X., 2020. Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE Geosci. Remote Sens. Mag.* 8, 38–59.

Xu, L., Oja, E., Kultanen, P., 1990. A new curve detection method: Randomized Hough Transform (RHT). *Pattern recognition letters* 11, 8.

Yla-Jaaski, A., Kiryati, N., 1994. Adaptive termination of voting in the probabilistic circular Hough transform. *IEEE Trans. Pattern Anal. Machine Intell.* 16, 911–915.

Zhao, R., Pang, M., Liu, C., Zhang, Y., 2019. Robust Normal Estimation for 3D LiDAR Point Clouds in Urban Environments. *Sensors* 19, 1248.

Zhu, X.X., Shahzad, M., 2014. Facade Reconstruction Using Multiview Spaceborne TomoSAR Point Clouds. *IEEE Trans. Geosci. Remote Sensing* 52, 3541–3552.

Zhu, X.X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geosci. Remote Sens. Mag.* 5, 8–36.