

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Egidij Egej Vencelj

**Analiza vertikalnosti stebrov  
visokonapetostnih omrežij s pomočjo  
obdelave oblaka točk**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM  
PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR:izr. prof. dr. Danijel Skočaj

Ljubljana, 2019

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Računalniški vid se vse bolj uveljavlja za reševanje različnih aplikativnih problemov. Eden izmed takšnih problemov je tudi avtomatski nadzor naklonov stebrov visokonapetostnega električnega omrežja. V diplomski nalogi razvijte postopek, ki s procesiranjem oblaka 3D točk, posnetega iz zraka s senzorjem LIDAR, avtomatsko določi središnico stebra ter izračuna naklon stebra oz. njegov odklon od vertikalne lege. Postopek implementirajte in aplicirajte na realnih podatkih. Dobljene rezultate primerjajte z ročno določenimi ter ocenite uspešnost delovanja predlaganega postopka.



*Zahvaljujem se bližnjim, ki so me tekom študija podpirali. Še posebna zahvala gre mojemu pokojnemu dedu, ki me je med študijem finančno in moralno podpiral. Prav tako se zahvaljujem družini Bernik za vso podporo v zadnjem letu študija. Zahvalil bi se tudi mentorjuizr. prof. dr. Danijelu Skočaju za mentorstvo in potrpljenje.*



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Motivacija . . . . .	1
1.2	Definicija problema . . . . .	2
1.3	Pregled ustaljenih metod in sorodne literature . . . . .	3
1.4	Predviden pristop . . . . .	4
<b>2</b>	<b>Vhodni podatki</b>	<b>7</b>
2.1	Zajem podatkov . . . . .	7
2.2	Oblika podatkov . . . . .	9
2.3	Koordinatni sistemi . . . . .	10
2.4	Opazovani objekti . . . . .	11
<b>3</b>	<b>Implementacija rešitve</b>	<b>15</b>
3.1	Iskanje in izbira relevantnih točk . . . . .	16
3.2	Določanje vrha stebra . . . . .	17
3.3	Določanje talne površine . . . . .	22
3.4	Določanje središča ob temelju stebra . . . . .	28
3.5	Implementacijske podrobnosti . . . . .	29
<b>4</b>	<b>Rezultati</b>	<b>33</b>
4.1	Testni podatki . . . . .	33

4.2	Merjenje napak . . . . .	34
4.3	Analiza napak pri določitvah karakterističnih točk . . . . .	36
4.4	Analiza napak pri klasifikaciji . . . . .	38
<b>5</b>	<b>Zaključek</b>	<b>41</b>
5.1	Nadaljnje delo in možne izboljšave . . . . .	41
5.2	Zaključne misli . . . . .	42
	<b>Literatura</b>	<b>44</b>

# Seznam uporabljenih kratic

<b>DBMS</b>	Database management system	Sistem za upravljanje s podatkovnimi zbirkami
<b>LiDAR</b>	Light detection and ranging	Svetlobno zaznavanje in merjenje
<b>GNSS</b>	Global navigation satellite system	Globalni satelitski sistem navigiranja
<b>GPS</b>	Global positioning system	Globalni sistem pozicioniranja
<b>DBSCAN</b>	Density-based spatial clustering of applications with noise	Prostorsko gručenje šumnih podatkov
<b>RANSAC</b>	Random sample consensus	Soglasje naključnih množic
<b>TLS</b>	Terrestrial lidar system	Talni LiDAR sistem
<b>ALS</b>	Airborne lidar system	Zračni LiDAR sistem
<b>INS</b>	Inertial navigation system	Inercijski navigacijski sistem



# Povzetek

**Naslov:** Analiza vertikalnosti stebrov visokonapetostnih omrežij s pomočjo obdelave oblaka točk

**Avtor:** Egidij Egej Vencelj

## **Povzetek:**

Za nemoteno oskrbo z električno energijo, ki je ključen element moderne infrastrukture, je potrebna tehnična brezhibnost električnih omrežij. Pregled električnih omrežij se zaradi njihove velikosti izvaja z letčega plovila s katerim zajemamo prostorske in druge podatke. Zaradi velikega števila prostorskih podatkov se pojavlja potreba po hitri in zanesljivi obdelavi, na podlagi katere določamo nekatere okvare omrežja. Samodejna določitev karakterističnih točk stebrov omrežja nam omogoča določitev primerov, kjer obstaja sum kritičnega odklona, torej okvare stebra omrežja. V diplomskem delu smo razvili postopek za samodejno določanje karakterističnih točk stebrov. Samodejno določitev karakterističnih točk smo dosegli z obdelavo oblaka točk s pomočjo kombinacije ustaljenih metod s področja računalniškega vida. V okviru diplomskega dela smo rezultate postopka primerjali z ročno določenimi točkami in na podlagi razlik ocenili uspešnost razvitega postopka.

**Ključne besede:** računalniški vid, napake, samodejno, zaznavanje, analiza, oblak točk.



# Abstract

**Title:** High voltage network pillar verticality analysis by means of point cloud processing

**Author:** Egidij Egej Vencelj

**Abstract:** Technical flawlessness of power grids is required for an uninterrupted flow of electricity, a key element of modern infrastructure. Due to their size, inspections of power grids are carried out with a flying vessel, which captures spatial and other data. Due to the large size of this spatial data, there is a need for fast and reliable processing, based on which we determine some of the network failures. The automatic determination of the characteristic points of the network pylons enables us to identify cases where a suspicion of a critical declination is present, ie. the failure of the network pylon. In this thesis we have developed a procedure for automatically determining the characteristic points of the pylons. The automatic determination of characteristic points was achieved by point cloud processing, using a combination of established methods in the field of computer vision. We compared the results of the procedure with manually determined points, and evaluated the success of the developed procedure on the basis of differences.

**Keywords:** computer vision, detection, fault, automatic, analysis, point cloud.



# Poglavje 1

## Uvod

### 1.1 Motivacija

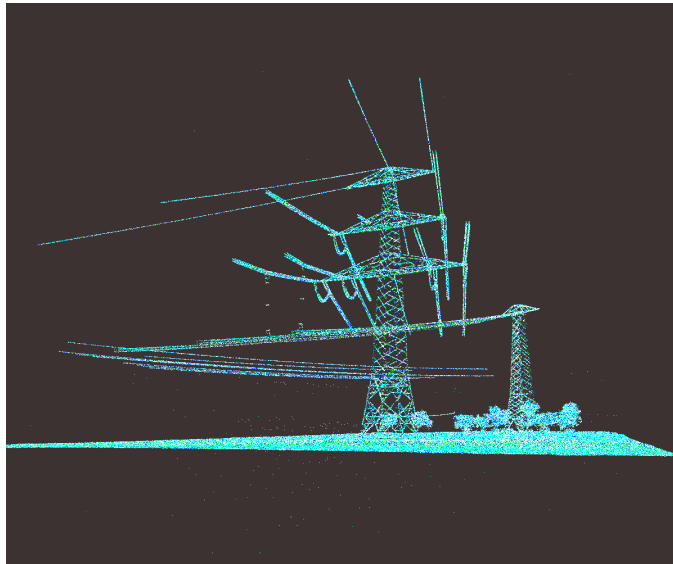
Za oskrbo z električno energijo so potrebna električna omrežja, ki omogočajo prenos električne energije od izvora do končnih uporabnikov. Na elementih omrežjih se pojavljajo napake, ki so posledica vremenskih in drugih mehanskih vplivov, konstrukcijskih napak, dotrajanosti posameznih elementov ali pa posledica slabega vzdrževanja koridorja (vpliv vegetacije). Pogosto je za vzdrževanje takih sistemov za celotno državo zadolžen en sam prenosni operater, zato poleg samega vzdrževanja pregled omrežja in analiza podatkov postaneta dodaten logističen problem. Zaradi tega prenosni operaterji za pregled omrežja pogosto najamejo zunanjega izvajalca.

Eden od produktov v okviru takega naročila je analiza prostorskih podatkov, zajetih s sistemom Light detection and ranging (LiDAR). Ena od možnih napak, s katero se srečujemo v okviru pregleda omrežja, je odklon oziroma ukrivljenost stebra, ki je lahko posledica prevelike natezne napetosti žic, plazenja ali pogrezanja tal ali pa gre za povsem namensko konstrukcijsko odločitev. Samodejni procesi v industriji omogočajo hitrejši in pogosto bolj konsistenten produkt, hkrati pa omogočajo povečanje produktivnosti in sočasno zmanjšanje obsega človeškega dela. V diplomskem delu se bomo osredotočili na samodejno analizo stebrov oziroma njihovih odklonov.

## 1.2 Definicija problema

Daljnovodi so infrastrukturni objekti, ki se raztezajo na ogromnih razdaljah. Tehnologija zajema podatkov že omogoča zajem podatkov v takem obsegu, pogosto pa se pojavljajo problemi pri interpretaciji tovrstnih podatkov zaradi njihove količine. Človeško delo predstavlja ozko grlo v postopku obdelave, saj je ročno določanje karakterističnih točk precej zamuden postopek. Samodejno določene karakteristične točke neposredno omogočajo tudi samodejno analizo odklonov stebrov. Zaradi omenjenih razlogov bi bilo smiselno čim večji del procesa opraviti samodejno.

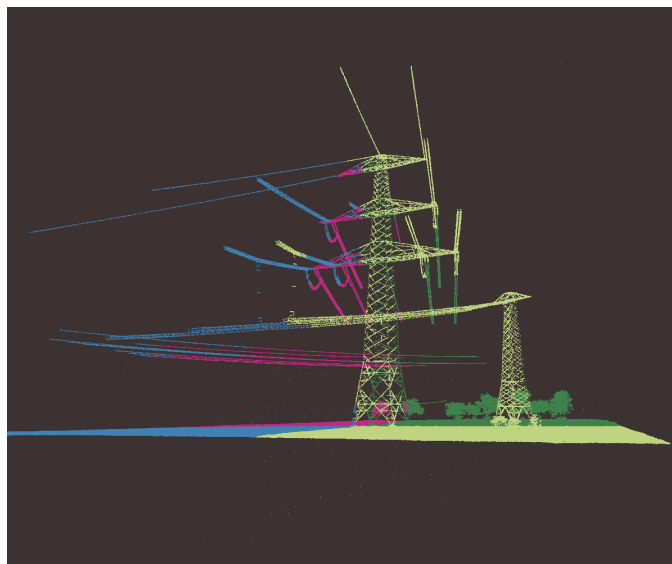
V diplomskem delu se posvečamo razvoju postopka za analizo napak s pomočjo obdelave oblaka točk. Primer oblaka točk je prikazan na sliki 1.1. Zasnovali bomo postopek samodejne določitve karakterističnih točk stebrov in analize njihovih odklonov. Postopek bo omogočil hitrejšo in bolj konsistentno obdelavo prostorskih podatkov.



Slika 1.1: Oblak točk v okolici stebra električnega omrežja.

### 1.3 Pregled ustaljenih metod in sorodne literature

Do sedaj sta se določitev in analiza vertikal ter njihovih odklonov opravljala ročno. Pred samim pregledom je bilo potrebno v program za prikaz prostorskih podatkov naložiti prave datoteke, saj pregled vseh podatkov sočasno zaradi njihove velike količine ni mogoč. Na sliki 1.2 je prikazan primer oblaka točk za steber, kjer je razvidno, da se zapisi nahajajo v različnih datotekah. Posamezna datoteka hrani točke za omejeno površino, zato se lahko zapisi točk za posamezen steber nahajajo v različnih datotekah.



Slika 1.2: Oblak točk, kjer so točke obarvane glede na datoteke, v katerih se nahajajo njihovi zapisi.

Za tem sta se morala določiti vrh in središče pri temelju za posamezen steber. Prostorski podatki so na monitorju predstavljeni ploskovno, kar zahteva sočasno gledanje iz večih perspektiv. To povzroča odstopanja, saj se človek lahko osredotoči le na posamezen pogled v nekem trenutku. Zaradi omejene gostote točk, je na tem koraku potrebna človeška presoja in domensko znanje. Za tem se je moralo določiti še vertikalno stebra, torej daljico med

določenima točkama. Te podatke o karakteristikah stebra je bilo potrebno prek uporabniškega vmesnika Database management system (DBMS) vnesti v zaledno bazo podatkov. Tak postopek je bilo potrebno ponoviti za vsak steber, nato pa so se morali izračunati še odkloni stebrov glede na določene točke.

V okviru obdelave podatkov se je moralo odstraniti tudi šumne podatke in popravljati razrede slabo označenih točk. Šum se pojavlja pod talno površino, nad opazovanimi objekti, okrog reflektivnih objektov in tudi drugje. Šum se je odstranjeval kombinirano s samodejnimi [8] in ročnimi metodami.

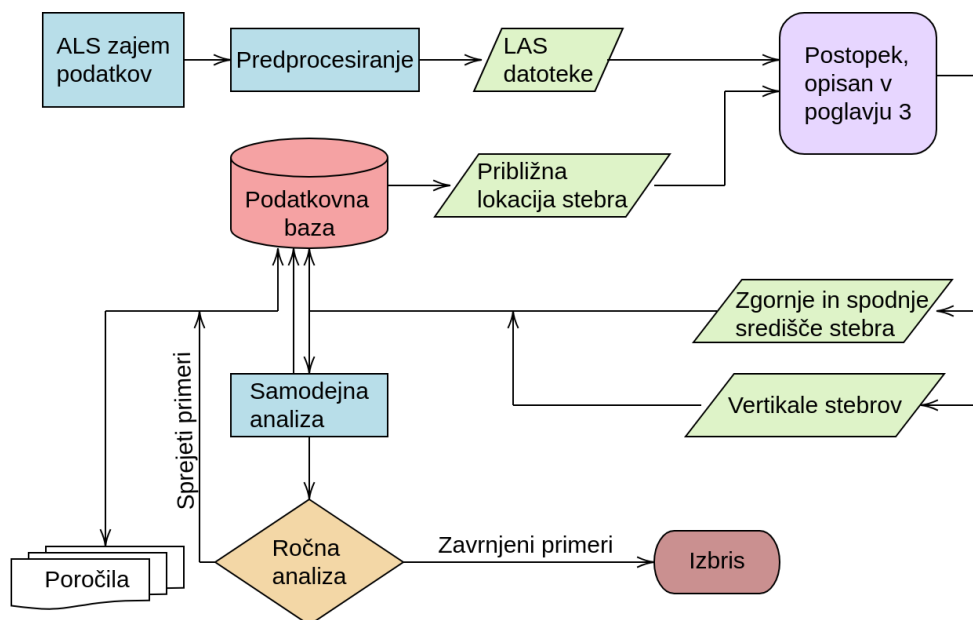
Za določitev talne površine so se in se še vedno uporabljajo različne metode glede na zahteve projekta in potreb [2, 3, 5, 17]. Določitev talne površine je pomembna za druge analize in generiranje rastrskih slik površja [15, 16, 22].

Zaradi ponavljajoče se narave dela in števila korakov v postopku je zaposlenim zbranost in posledično natančnost lahko upadla. Postopek je bil zaradi števila korakov in časa izvajanja posameznega koraka tudi časovno zamuden.

## 1.4 Predviden pristop

Delovni proces bomo nadgradili z vmesnim členom samodejne zaznave karakterističnih točk stebrov in njihovih vertikal. Posledica je sprememba poteka obdelave podatkov, ki je orisan na sliki 1.3. Namesto, da bi se pregledovalo vse stebre, želimo, da postopek sam izloči stebre, kjer pregled ni nujen. Samodejno želimo določiti karakteristične točke stebrov in opraviti analizo odklonov, preden zaposleni pričnejo interakcijo s podatki. Na sliki 1.3 je ta člen označen z vijolično barvo. Postopek bomo podrobneje predstavili v poglavju 3. Rezultati postopka se bodo zapisali v podatkovno bazo, kar nam bo omogočilo učinkovito izbiro primerov, ki jih je potrebno ročno preveriti (označeno oranžno).

V poglavju 2 bomo predstavili vhodne podatke. Predstavili bomo tehnologijo zajema, strukturo podatkov in njihove karakteristike. V tem poglavju



Slika 1.3: Grafični prikaz postopka od zajema podatkov, do izdelave poročila analize vertikalnosti stebrov visokonapetostnih omrežij.

so opisani tudi opazovani objekti in njihove značilnosti.

V poglavju 3 je podrobneje predstavljen postopek samodejne določitve karakterističnih točk stebrov visokonapetostnih omrežij. To zajema določitev talne površine ter vrha stebra in središča pri temelju. Predstavljeni so tudi pogosti problemi, ki jih srečujemo v praksi in predlagane rešitve nanje. Na koncu poglavja so opisane tudi implementacijske podrobnosti postopka.

V poglavju 4 ocenimo uspešnost postopka. Najprej predstavimo zbirko testnih podatkov, nato pa določimo metriko in postopek merjenja napak, razložimo in določimo pa tudi tipe napak. Za tem si ogledamo konkretne rezultate, na koncu pa jih še ocenimo.



# Poglavje 2

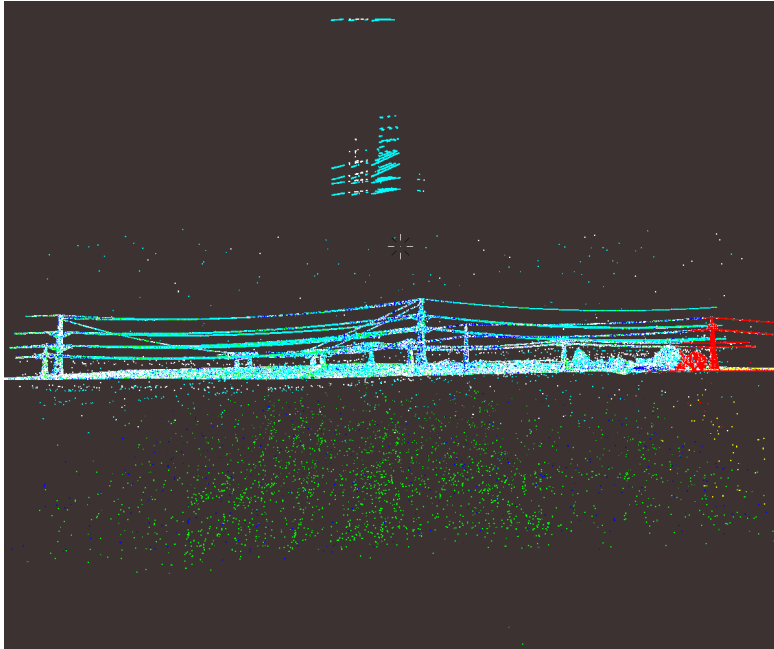
## Vhodni podatki

V tem poglavju je predstavljen način zajema podatkov in njihova struktura. Predstavljeni so tudi opazovani objekti in njihove značilnosti.

### 2.1 Zajem podatkov

Vhodne podatke za analizo, torej georeferenciran oblak točk, primer katerega je prikazan na sliki 2.1, je potrebno pridobiti z različnimi metodami daljinskega zaznavanja. Za zajem oblaka točk so na voljo različne vrste metodologij in senzorjev. Za tovrsten zajem podatkov se danes v industriji običajno uporabljajo senzori LiDAR. Tovrstni senzori so lahko talni (Terrestrial lidar system (TLS)) ali pa zračni (Airborne lidar system (ALS)). Pri sistemih TLS se podatke zajema iz posameznih stojišč, kjer je senzor LiDAR nameščen na ustrezne stativne, s sistemi ALS pa snemanje poteka iz zračnega plovila (letalo, helikopter, dron itd.). Obstajajo tudi sistemi, kjer je mogoče oblak točk generirati s pomočjo fotografij z znanim položajem in orientacijo v prostoru s tehniko iskanja presečišč žarkov skozi slikovne elemente slik, ki predstavljajo isto vsebino na dveh ali več posnetkih.

V našem primeru je vir podatkov senzor LiDAR (Slika 2.2), ki se v praksi pogosto kombinira še z drugimi senzori (različne kamere) z namenom povečanja količine zajetih podatkov, ki jih lahko med seboj primerjamo in



Slika 2.1: Stranski pogled na vhodni oblak točk, ki prikazuje dve razpetini daljnovoda. Nad daljnovodom je moč opaziti šum, barve točk pa so določene s tipom odboja, zapisanem v LAS datoteki.

kombiniramo ter tako izboljšamo njihovo kvaliteto.

Senzor LiDAR ne meri niti absolutnih niti relativnih koordinat točk, ampak meri odboje laserskih žarkov, njihove odbojne čase in jakosti. Gre za relativne meritve, ki neposredno niso uporabne, ker jih je potrebno še georeferencirati, hkrati pa doseči zadostno absolutno stopnjo natančnosti in homogenosti oblaka točk za konkreten namen snemanja. V ta namen imajo vsi sistemi, ki se uporabljajo za pridobivanje oblaka točk, skoraj po pravilu, dodan še navigacijski sistem. Navigacijski sistem je sestavljen iz satelitskega navigacijskega (Global navigation satellite system (GNSS)) in inercialnega (Inertial navigation system (INS)) podsistema. Absolutne koordinate točk se pridobi šele v postopku poobdelave, kjer se upošteva podatke vseh podsistemov.

Fizična implementacija sistema ALS je realizirana v obliki namenskega ohišja, ki vsebuje našete senzorje. Ohišje je pričvrščeno zračnemu plovilu



Slika 2.2: Uporabljen sistem ALS za zajem podatkov.

na način, ki omogoča neoviran pogled senzorjev proti tlor (Slika 2.3).

## 2.2 Oblika podatkov

Podatki se shranjujejo v t.i. bloke, ki pripadajo področjem vzdolž koridorja daljnovoda. Standardiziran format zapisa tako georeferenciranih laserskih točk je LAS ali njegova stisnjena različica LAZ, obstajajo pa tudi zaprti formati, tipično podani s strani proizvajalca posameznega senzorja.

Vsaka LAS datoteka ima določeno strukturo atributov, ki je pogojena tako z različico formata kot tudi z zmožnostmi senzorja. Za reševanje problema v okviru diplomskega dela pa nas zanimajo:

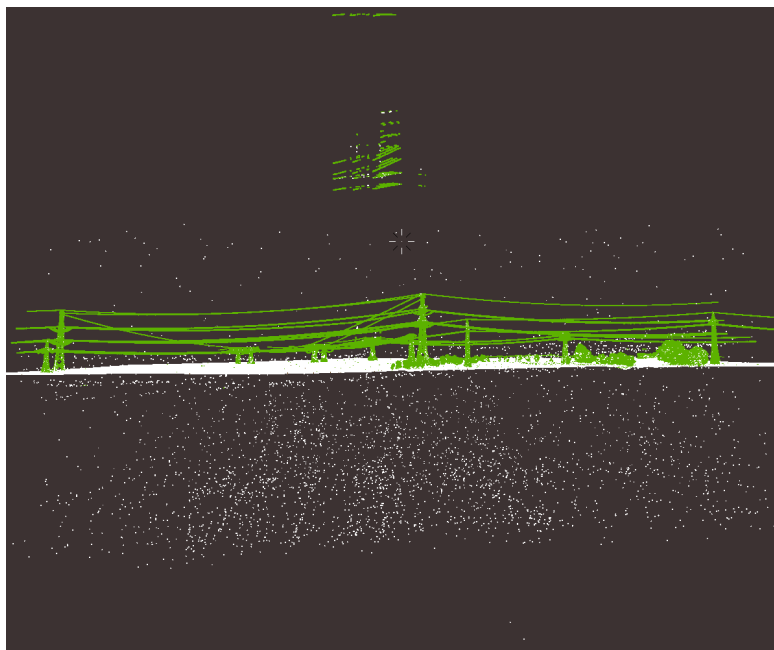
- x - koordinata x v koordinatnem sistemu
- y - koordinata y v koordinatnem sistemu
- z - koordinata z v koordinatnem sistemu
- class - razred točke (Slika 2.4)



Slika 2.3: Plovilo z nameščenim sistemom ALS.

## 2.3 Koordinatni sistemi

Ker se za osnovno georeferenciranje uporablja sistem GNSS, so začetni podatki (koordinate točk) podani v koordinatnem sistemu izbranega sistema GNSS (v praksi materializirani so vsem znani Global positioning system (GPS), ruski GLONASS, evropski GALILEO in kitajski BEIDOU). V konkretnem primeru se sicer za poobdelavo uporabljajo opazovanja satelitov tako omrežja GPS kot tudi omrežja GLONASS, vendar pa so rezultati preračunani v tako imenovani geodetski datum sistema GPS, poznanega pod imenom WGS84 (EPSG: 4326). To je geografski koordinatni sistem, kar pomeni, da je položaj podan z zemljepisno dolžino in širno, višina pa je določena geometrično in predstavlja oddaljenost od centra referenčnega telesa, ki je v tem primeru WGS84 rotacijski elipsoid (kot telo, ki aproksimira obliko Zemlje). Kakršnikoli preračuni razdalj, površin in drugih količin na rotacijskem elipsoidu so relativno kompleksni, zato se v geodeziji poslužujejo uporabe projiciranih - ravninskih koordinatnih sistemov. Projekcija je v tem kon-



Slika 2.4: Stranski pogled na oblak točk, kjer je barva določena z razredom točke.

tekstu funkcija, ki za izbrano lokalno območje določa projekcijsko ravnino (2D kartezični koordinatni sistem), ki zagotavlja enostavnost izračunov in kljub deformaciji še vedno zadostno natančnost za izbrani namen uporabe. V konkretnem primeru je bila izbrana projekcija UTM 32N (EPSG: 32632), saj se območje, kjer poteka daljnovod, nahaja na zahodnem delu Nemčije, ki geografsko gledano spada pod območje te projekcije.

## 2.4 Opazovani objekti

Hrbtenico električnega energetskega sistema vsake države (pa tudi širše) tvorijo daljnovodi. Osnovna karakteristika vsakega daljnovoda je njegova nazivna napetost. Ta pogojuje skoraj vse nadaljnje konstrukcijske lastnosti daljnovoda, kot so velikost, tip in debelina uporabljenih vodnikov, višina stebra, debelina paličja, uporabljen material stebra itd.

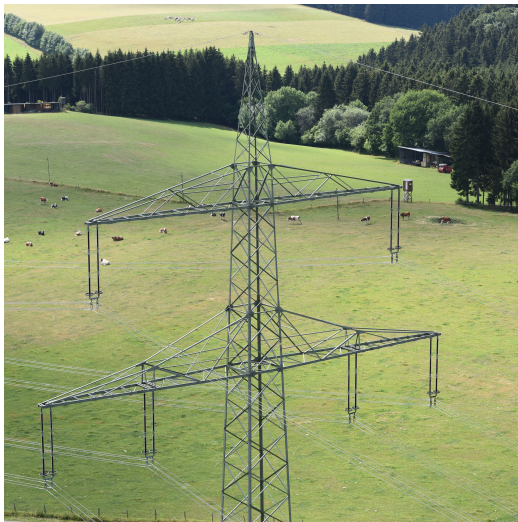
V osnovi delimo električno omrežje na prenosno (visokonapetostno) in

distribucijsko (srednje in nizkonapetostno). Prenosno omrežje je sestavljeno iz 110, 220 in 380 kV (v tujini tudi še 750 kV) daljnovodov in je namenjeno prenosu električne energije. Distribucijsko omrežje (20 in 40 kV) pa je namenjeno dovodu električne energije do posameznih naselij ali posameznih večjih industrijskih odjemalcev. Ostaja še nizkonapetostno omrežje, ki se običajno vije od hiše do hiše.

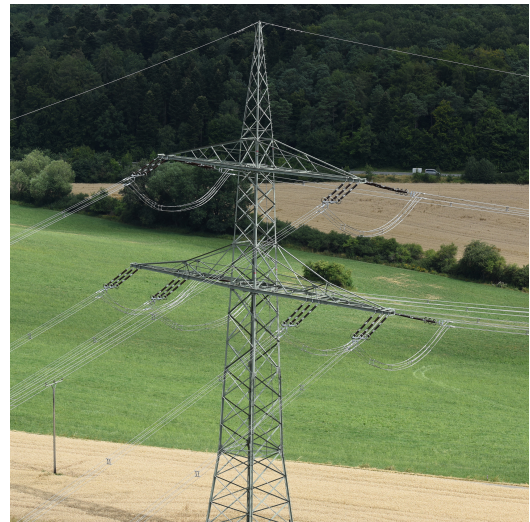
Vozlišča, kjer se napetosti pretvarjajo, daljnovodi pa začenjajo in končujejo, imenujemo razdelilne transformatorske postaje. V smislu topologije omrežja, lahko govorimo o hierarhiji iz velikega v malo. Prenosni daljnovodi povezujejo velike razdelilne transformatorske postaje in so običajno daljši od 20 km, dolgi pa so lahko vse do 150 km in več. Distribucijski vodi so krajši in se praviloma končajo v majhnih transformatorskih postajah, od koder se nadaljuje omrežje nizke napetosti. Vizualno distribucijske vode lahko ločimo od prenosnih že po tipu oporišča, ki je v Sloveniji praviloma leseno, v tujini pa se pojavljajo tudi železobetonski ali povsem kovinski stebri, ki običajno ne presežejo višine dreves. Stebri daljnovodov visokonapetostnega omrežja so praviloma zgrajeni v obliki paličja, njihova višina pa je okrog 60 m. Predmet opazovanja in analize, opisane v diplomski nalogi, so stebri visokonapetostnega omrežja.

Stebre lahko delimo tudi glede na njihovo funkcij v daljnovodu:

- *Portalni steber* - priključuje vodnike iz daljnovoda v razdelilno transformatorsko postajo. Primer takega stebra je prikazan na sliki 4.3.
- *Zatezni steber* - praviloma se pojavi povsod, kjer daljnovod spremeni smer, lahko pa tudi vmes, na dolgi ravnini. Na takem stebru se žice prekinejo in so preko tokovnega loka povezane z naslednjo razpetino. Izolatorji so na takem stebru montirani v smeri žice, kot je razvidno na sliki 2.5b.
- *Nosilni steber* - najbolj običajen steber, kjer žica zgolj prehaja skozi nosilno klemo pod izolatorje, ki so nameščeni vertikalno. Primer takega stebra je prikazan na sliki 2.5a.



(a) Nosilni steber.



(b) Zatezni steber.

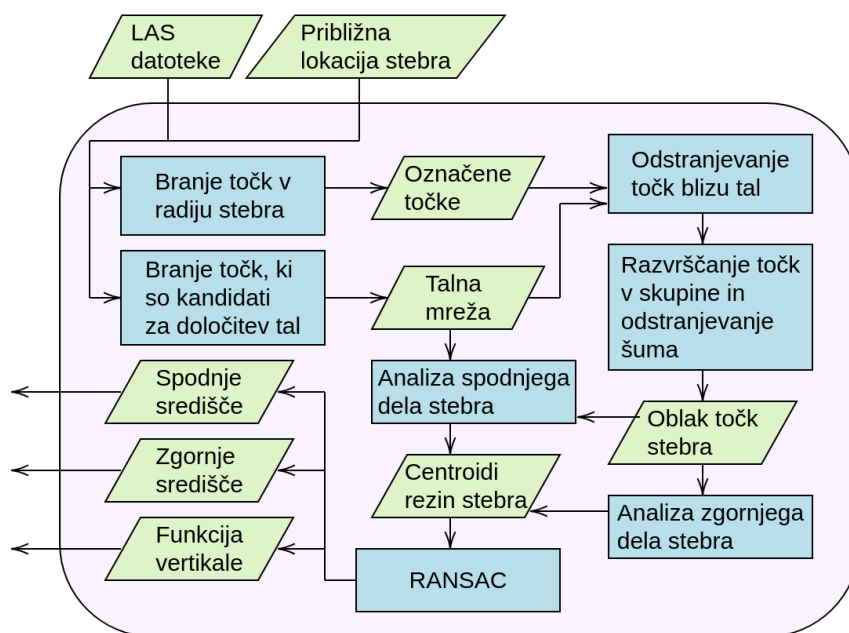
Slika 2.5: Fotografiji nosilnega in zateznega stebra.



## Poglavje 3

# Implementacija rešitve

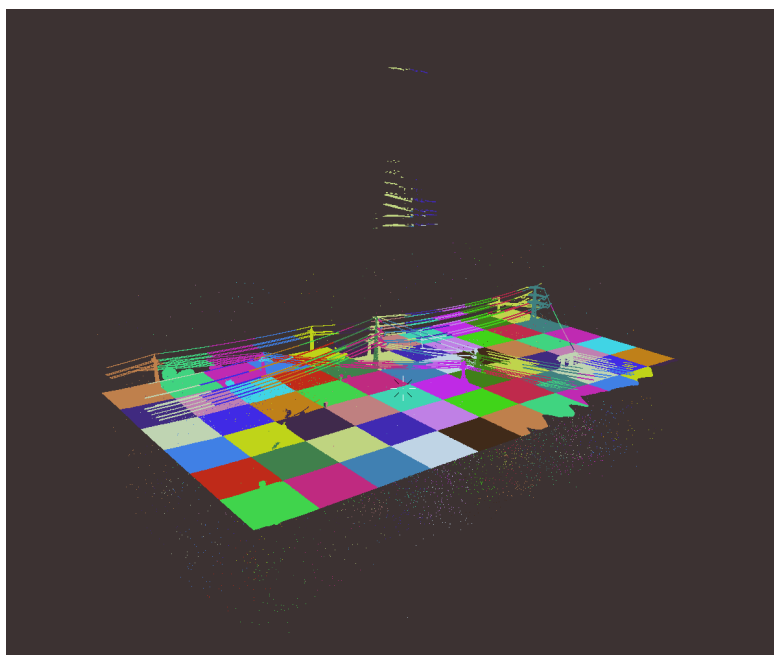
V tem poglavju je predstavljen postopek določanja karakterističnih točk stebrov, orisan pa je na sliki 3.1. Predstavljeni so tudi problemi, ki se v praksi pogosteje pojavljajo in predlagane možne rešitve zanje.



Slika 3.1: Grafični prikaz postopka za določitev karakterističnih točk in normale stebra.

### 3.1 Iskanje in izbira relevantnih točk

Za določitev karakterističnih točk posameznega stebra je potrebno iz vseh podatkov, ki so bili zajeti tekom projekta, izbrati podmnožico, ki je primerna za obdelavo. Posamezne bloke v koraku predobdelave razdelimo na manjše podbloke, kar je razvidno na sliki 3.2. V bazi hranimo zapise poligonov, ki jih posamezni podbloki geografsko pokrivajo. Za vsak steber poznamo tudi približno lokacijo, ki ni tako natančna kot naši georeferencirani oblaki točk. Najprej poiščemo bloke, ki se nahajajo v okolici lokacije obdelovanega stebra in jih naložimo v delovni spomin. Iz vsakega izmed naloženih blokov izberemo točke, ki se nahajajo znotraj izbrane razdalje od približne lokacije stebra in jih shranimo v združeno tabelo. V tabelo shranimo koordinate  $x$ ,  $y$  in  $z$ , če se tudi razred točk ujema s podanim argumentom.



Slika 3.2: Stranski pogled na oblak točk, kjer je barva točk določena z datoteko, v kateri se točke nahajajo. Točke, katerih podatki se nahajajo v istih datotekah, imajo enako barvo.

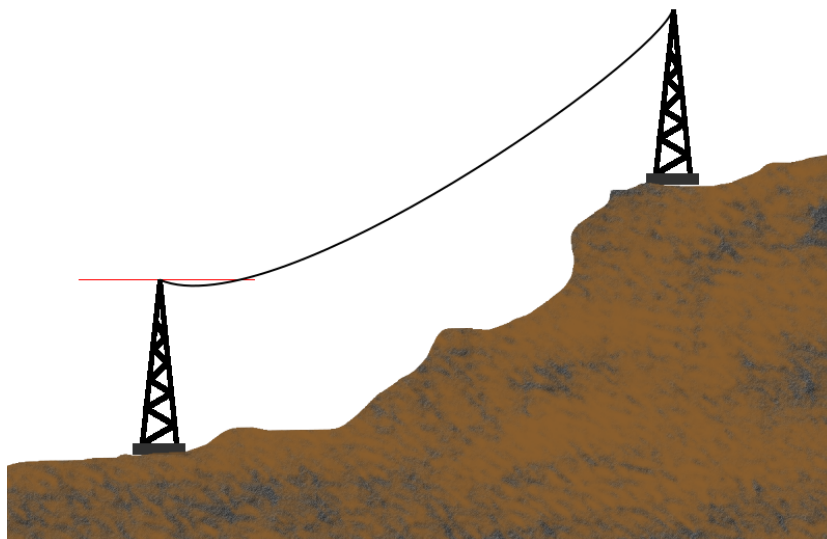
## 3.2 Določanje vrha stebra

Ker v tem koraku še ne poznamo bolj natančne lokacije stebra, je potrebno trenutni približek izboljšati. Pričakujemo, da vrhovi stebrov predstavljajo lokalni maksimum, saj stebri ne smejo biti neposredno obdani z drevesi ali drugimi objekti. Na žalost se v praksi izkaže, da temu ni vedno tako.

### 3.2.1 Pogosti problemi

#### Obravnavanje žic pri iskanju lokalnega maksimuma

Ker sta lahko dva zaporedna stebra na izrazito različnih višinah, se lahko zgodi, da žice, ki ju povezujejo, predstavljajo lokalni maksimum enega od stebrov (Slika 3.3). Take napake so sicer redke, a se občasno vseeno pojavijo.



Slika 3.3: Primer, kjer zaradi lastnosti terena vrh stebra ni lokalni maksimum.

Problem rešimo tako, da točke najprej razvrstimo padajoče po njihovih koordinatah  $Z$ . Za vsako izmed točk v izbrani okolici preverimo število točk

in prostornino njihove konveksne ovojnice. Če imamo opravka z žico, bo prostornina majhna, prav tako pa bo majhno tudi število točk v okolici. Zanko ustavimo, ko je število točk dovolj veliko in prostornina njihove konveksne ovojnice dovolj velika. Izbrana točka določi dimenzijo  $Z$  vrha in začetni približek po dimenzijah  $X$  in  $Y$  - steber vrha pogosto nima v eni točki, ker gre za ploskev, katere centroid je potrebno še določiti.

### Zaznavanje in odstranjevanje šuma

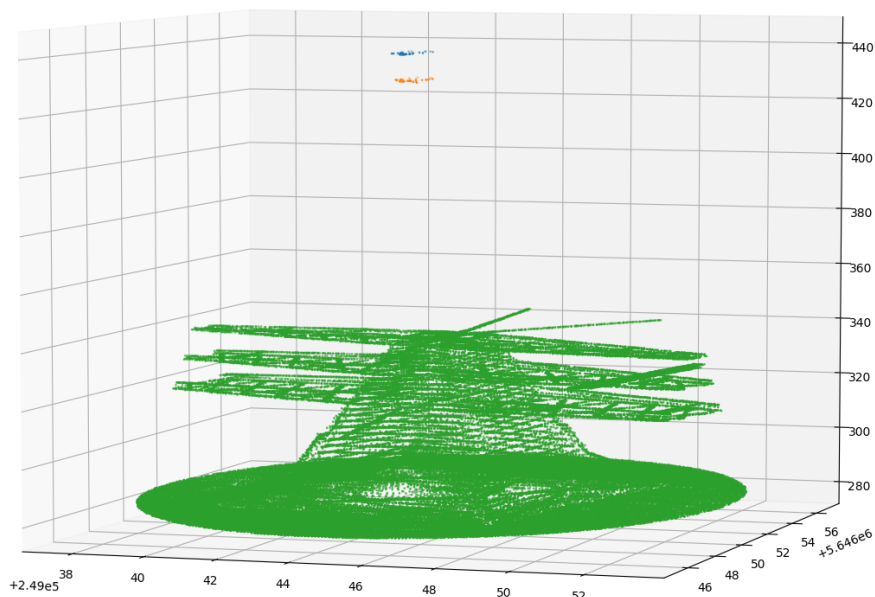
Občasno se zgodi, da so lokalni maksimum šumni podatki, ki nastajajo zaradi vlage v ozračju ali pa odbojev od izrazito gladkih površin (vodne površine, steklo, občasno strehe, mokre ceste itd.). Šum lahko delno omejimo s pravilno izbiro časa zajema podatkov, ko je relativna vlažnost zraka nižja.

Kljub preventivnim metodam, se šum v podatkih še vedno lahko pojavi, zato ga moramo robustno zaznati in odstraniti. Podatke razvrstimo v razrede z algoritmom Density-based spatial clustering of applications with noise (DBSCAN) [7], pri čemer so parametri izbrani tako, da celoten steber postane svoj razred, višje locirani šumni podatki pa običajno že v tem koraku klasificirajo kot šum. Občasno je šumnih podatkov še vedno dovolj, da jih algoritem obravnava kot svoj razred (Slika 3.4). Določitev konveksne ovojnice in njene prostornine nam omogoča izločitev takih gruč, saj iz vseh gruč izberemo tisto, ki ima največji volumen, ostale pa zavržemo. Tovrstni primeri se pojavljajo relativno redko, a vseeno dovolj pogosto, da je dodatna obravnava potrebna.

Podatki, pridobljeni v tem koraku, se uporabljajo kot osnova za vse nadaljnje postopke, ki zadevajo zaznavo delov stebra.

### 3.2.2 Natančno določanje vrha

Za bolj natančno določanje središča na horizontalni ravnini vrha je smiselno, da uporabimo vse razpoložljive informacije, ki nam jih oblak točk ponuja. Navdih dobimo iz človeškega dojemanja oblik. Steber dojemamo kot piramido. Stebru bi bilo smiselno določiti vertikalno.



Slika 3.4: Primer pojavitve šumnih podatkov visoko nad stebrom.

Steber razdelimo na rezine in za vsako določimo centroid. Iz množice tako določenih središč lahko določimo sredinsko vertikalno stebra, kar nam omogoča tudi izločanje slabo določenih centroidov s pomočjo metode Random sample consensus (RANSAC) [10]. Izračun površine in centroida posamezne rezine oziroma njene preslikave v 2D podprostoru izračunamo po postopkih, opisanih v Bourke [1].

Kot vhodne podatke podamo:

- *points*: Točke, s katerimi operacija poteka
- *start*: Začetna točka

- *step*: Smerni vektor premika točke za korak
- *depth*: Debelina rezine
- *grow*: Koliko širitve na korak dovoljujemo - torej koliko je maksimalni odmik posameznega točkovnega zapisa od stranice 2D konveksne ovojnice iz prejšnjega koraka

Za začetno točko izberemo kar centroid konveksne ovojnice zgornjega dela stebra, smerni vektor kaže navzgor ( $[0, 0, 0.25]$ ), debelina rezine pa je 1,5 m. Za vsak korak dovolimo 5 cm horizontalne rasti - to pomeni, da sprejmemo tudi točke, ki se v horizontalni preslikavi nahajajo do največ 5 cm izven prejšnje ovojnice. Parametri so izbrani glede na realne dimenzije.

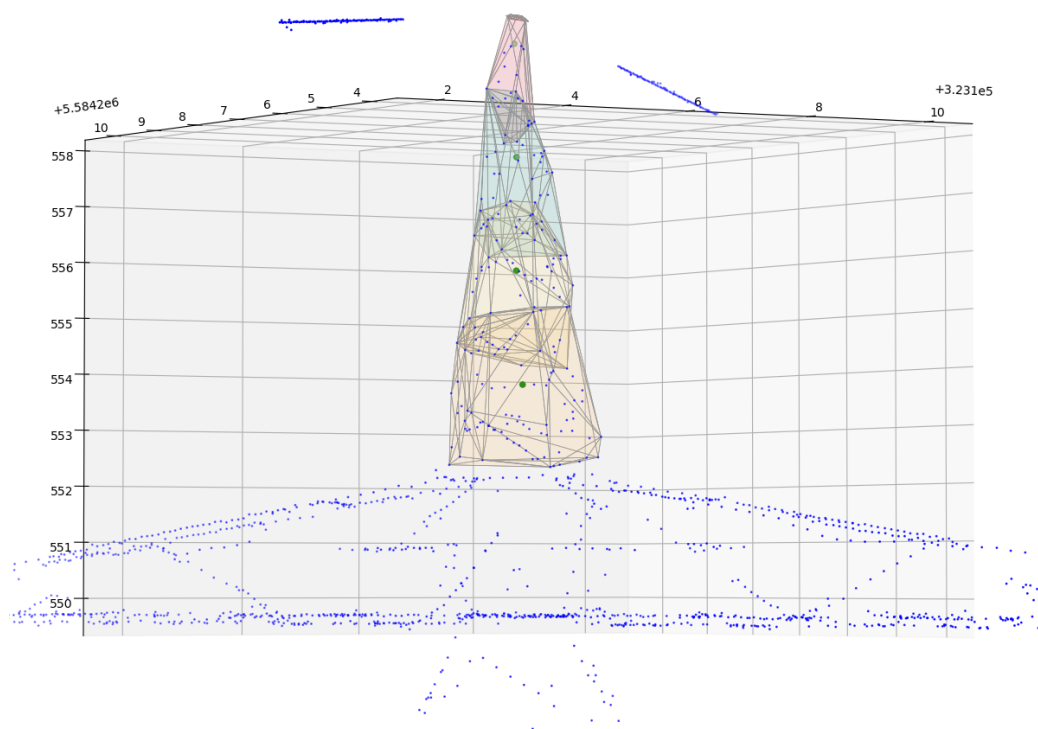
Pred prvo iteracijo glede na podane parametre poiščemo točke v okolici začetne točke, izračunamo konveksno ovojnico in točke v tej rezini shranimo v seznam rezin, nato pa pričnemo iteracijo.

V vsaki iteraciji najprej poiščemo točke okrog izbrane točke (*POI*). Za tem izračunamo horizontalno 2D konveksno ovojnico točk iz prejšnje iteracije. Izločimo vse točke v trenutni iteraciji, ki se v 2D podprostoru preslikajo zunaj konveksne ovojnice iz prejšnje iteracije. Če je preostalih točk premalo, prekinemo iteracijo, sicer pa poračunamo konveksno ovojnico trenutne iteracije. Centroid in točke rezine za trenutno iteracijo shranimo v svoja seznama. Če je točk za izračun konveksne ovojnice premalo, iteracijo na tem koraku prekinemo.

Kot rezultat funkcije prejmemo seznam rezin (tj. več oblakov točk) in seznam centroidov teh rezin.

Čeprav so stene stebra precej votle, jih ljudje interpretiramo kot ploskve. Postopek moramo primerno prilagoditi, da lahko steber obdelujemo kot telo s ploskvami. Posamezna rezina v preseku na določeni višini ne zglada pravokotna, pač pa gre za poligon, ki je odvisen od palične razporeditve na tej višini stebra (Slika 3.5).

V izogib deformaciji mreže stebra zagotovimo, da se rezine deloma prekrivajo, nato pa jih spojimo. Ker plašč, ki ga opiše unija robnih točk kon-

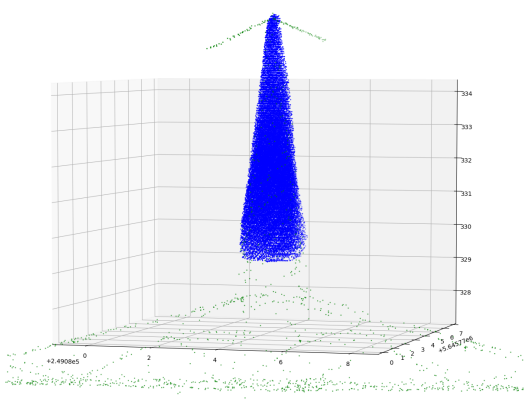


Slika 3.5: Primer nekoliko slabše porazdelitve paličja, nižje gostote točk in posledičnih deformacij rezin. Za lažjo predstavo je prikazana vsaka osma rezina, da se prikazane rezine medsebojno ne prekrivajo.

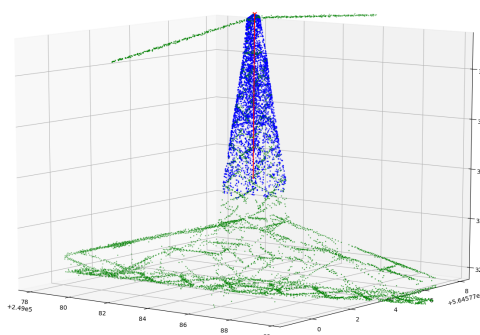
veksnih ovojnici rezin ni enaka konveksni ovojnici unije robnih točk ovojnic, moramo plašč stebra določiti drugače. V območju vsake rezine generiramo gosto mrežo sintetičnih točk (Slika 3.6a), nato pa odstranimo tiste, ki se nahajajo izven njene konveksne ovojnice (Slika 3.6b). Ohranimo lahko samo vozlišča ovojnic in tako bistveno zmanjšamo število točk, na katerih nadaljnje operacije potekajo. V zadnjem koraku točke združimo in vrnemo mrežo (Algoritem 3), ki je primerna za obdelavo.

Na tako generiranem oblaku točk lahko poženemo postopek za rezanje generirane mreže in določitev centroidov rezin (Algoritem 2), ki je bolj pre-

prosta verzija postopka za razrez stebra na rezine in iskanje centroidov (Algoritem 1). Glavna razlika med njima je, da za pri poenostavljenem postopku za posamezno rezino ni potrebno računati konveksne ovojnice v dveh ali treh dimenzijah, saj je oblika plašča že pripravljena (generirana v prejšnjem koraku). Algoritem lahko poženemo z bolj strogimi parametri in tako dobimo več bolj določenih rezin in pripadajočih centroidov.



(a) Polna mreža točk (prikazana vsaka 20. točka).



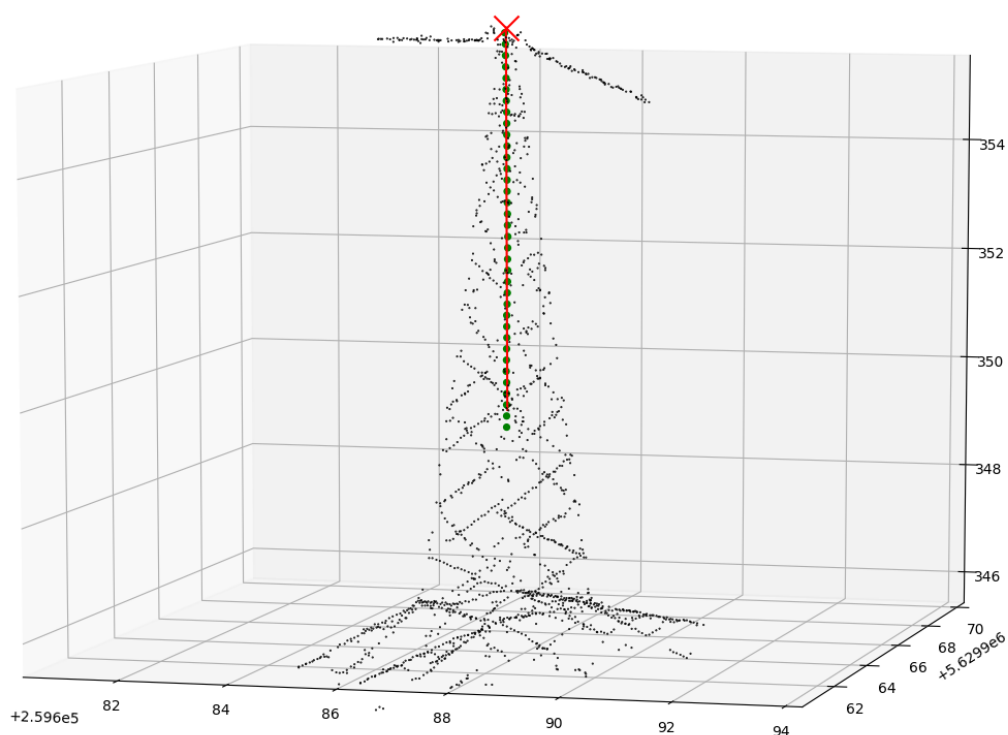
(b) Prikaz točk, ki definirajo plašč zgornjega dela stebra (brez redčenja v prikazu).

Slika 3.6: Prikaz generirane mreže plašča zgornjega dela stebra.

Iz pridobljenih točk je možno bolj robustno določiti pravo središče stebra oziroma njegovega vrha (Slika 3.7). Za ta namen na točke vertikalne prilagajamo funkcijo premice s pomočjo algoritma RANSAC [9, 21]. Funkcija vrne izhodiščno točko in smerni vektor premice.

### 3.3 Določanje talne površine

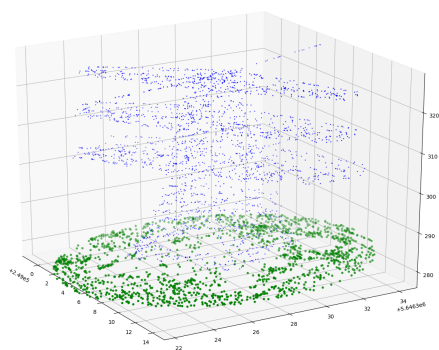
Pred določanjem centroida ob temelju stebra, je potrebno določiti še talno mrežo, saj se mora centroid nahajati na presečišču talne površine in vertikalne stebra. V datotekah LAS se nahajajo tudi zapisi točk talne površine (Slika 3.8a) oziroma kandidati, ki bodo kasneje uporabljeni v postopkih za



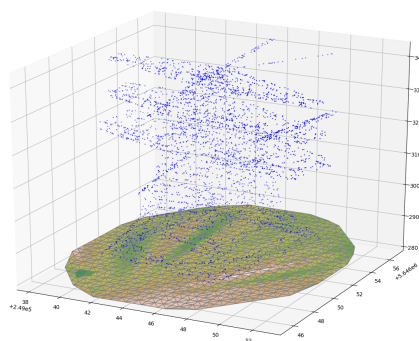
Slika 3.7: Primer zaznane vrhnje točke, označene z rdečim X. Z rdečo črto je označena daljica na premici, ki je rezultat algoritma RANSAC. Z zeleno so označene točke centroidov, ki pripadajo posameznim rezinam.

določanje talne površine. Obstaja mnogo avtomatskih in polavtomatskih postopkov za bolj robustno in natančno določitev talne površine [17, 5], za rešitev konkretnega problema pa bo približek dovolj.

Da lahko izračunamo presečišče, je potrebno določiti talno mrežo, s katero lahko vertikalno premico stebra sekamo. Zaradi spremenljive gostote točk generiramo točkovno mrežo z gostoto 1 m in tretjo dimenzijo interpoliramo. Kot izhod iz funkcije prejmemo enakomerno porazdeljeno mrežo glede na dimenziji X in Y, dimenzija Z pa se interpolira glede na vhodne podatke. Interpolacija ravnine je primerna, ker v opazovani okolici ni previsov ali izrazito razgibanega terena. Nad generirano mrežo izvedemo Delaunayovo triangulacijo [6, 12] (Slika 3.8b). Izhod ima več atributov, zanimajo pa nas kazalci na točke, ki sestavljajo trikotnike talne mreže.



(a) Vhodni podatki za določanje talne površine. Kandidati za talne točke označeni zeleno.



(b) Interpolacija talne površine.

Slika 3.8: Določanje talne površine.

---

**Algoritem 1:** razrez stebra na rezine in iskanje centroidov

---

```
1 function hull_projection_scan(points, poi, step, grow = 0);
  Input  : points, poi, step, depth, grow
  Output: slices, centroids
2 begin
3   init center_pts, slices
4   hull ← ConvexHull(points_around(poi, points, dz=depth))
5   while True do
6     buff ← points_around(Poi, points, dz = depth)
7     if len(buff) < 5 then
8       | break #V opazovani okolici ni dovolj točk
9     end
10    slice_pts ← buff
11    #Create 2D hull and eliminate points outside of it
12    hull2d ← ConvexHull(hull.points[:, 0 : 2])
13    hull_2d_path ← Path(hull2d.points[hull2d.vertices])
14    pts_in ←
15    hull_2d_path.contains_points(slice_pts[:, 0:2], radius=grow)
16    if sum(pts_in) < 4 then
17      | break #Na voljo ni dovolj točk za določitev konveksne
18      |   ovojnice
19    end
20    slice_pts ← slice_pts[pts_in]
21    c ← poly2d_centroid(slice_pts)
22    c ← np.hstack([c, poi [2]])
23    hull ← ConvexHull(slice_pts)
24    slices.append(slice_pts)
25    center_pts.append(c)
26    poi = poi + step
27 end
return np.array(slices), np.array(center_pts)
```

---

---

**Algoritem 2:** rezanje generirane mreže in določitev centroidov rezin
 

---

```

1 function z_slice_center_scan(points, start, step, depth);
   Input  : points, start, step, depth
   Output: slices, centroids
2 begin
3   init center_pts, slices
4   poi ← start
5   while True do
6     buff ← points_around(Poi, points, dz = depth)
7     if len(buff) < 10 then
8       | break #V izbrani okolici ni dovolj točk
9     end
10    slice_pts ← buff c ← poly2d_centroid(slice_pts)
11    c ← np.hstack([c, poi [2]])
12    hull ← ConvexHull(slice_pts)
13    slices.append(slice_pts)
14    center_pts.append(c)
15    poi = poi + step
16  end
17  return np.array(slices), np.array(center_pts)
18 end

```

---

---

**Algoritem 3:** generiranje goste mreže

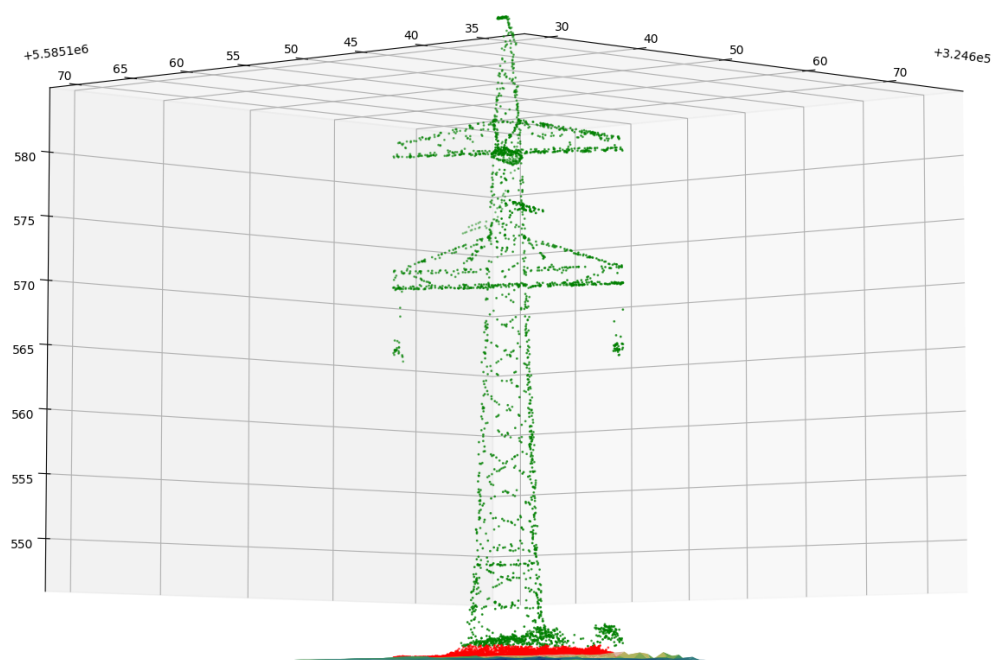
---

```
1 function generate_grid(slices, density);  
   Input  : slices, density  
   Output: grid  
2 begin  
3   init i, grid  
4   for s in slices do  
5     h ← ConvexHull(s)  
6     g = meshgrid(h, density) #Generiraj mrežo točk s podano  
   gostoto v mejnem okviru rezine  
7     pts_in ← in_hull(g, h.points[h.vertices])  
8     h ← ConvexHull(pts_in)  
9     grid.append(h.points[h.vertices])  
10  end  
11  return grid  
12 end
```

---

### 3.4 Določanje središča ob temelju stebra

Za obdelavo spodnjega dela stebra je potrebno odstraniti točke, ki pripadajo tlom in nizkemu rastju. Za vsako izmed točk določimo, nad katerim trikotnikom talne površine se nahaja. V kolikor je razdalja med točko in centroidom pripadajočega trikotnika manjša od željene meje, to točko odstranimo (Slika 3.9).

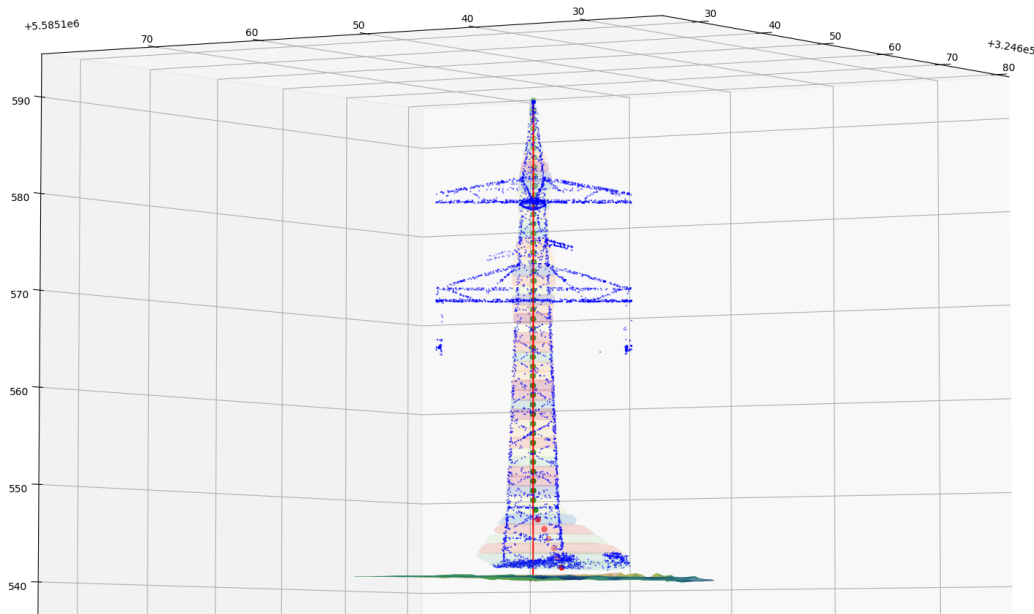


Slika 3.9: Z rdečo barvo so označene točke, ki jih odstranimo.

Nad pridobljenimi točkami poženemo postopek za razrez stebra na rezine in iskanje centroidov (Algoritem 2). Začetno točko poračunamo kot presek premice, določene v koraku določanja vrha stebra in talne površine. Postopek nam zagotovi odstranitev skoraj vseh točk, ki pripadajo rokam daljnovoda, ostane pa nam samo osrednji del stebra.

Tako kot v postopku iskanja vrhnje točke, tudi tu generiramo gostejši sintetičen oblak točk za stene stebra in na njem izvedemo postopek za rezanje generirane mreže in določitev centroidov rezin (Algoritem 2), rezultat katerega je večje število rezin in pripadajočih centroidov.

Obstaja verjetnost, da so centriodi ob vznožju stebra slabo določeni, saj se lahko v tem območju še vedno pojavijo ovire (Slika 3.10). Z metodo RAN-SAC tako izločimo osamelce (outlier) in določimo model vertikalne premice stebra. Točko, ki določa spodnje središče stebra, določimo kot presečišče vertikalne stebra in talne površine.



Slika 3.10: Primer stebra z motečo okolico, določeni osamelci in vertikala stebra.

### 3.5 Implementacijske podrobnosti

V poglavju je omenjenih nekaj postopkov in podatkovnih struktur, katerih implementacija je ponujena v zunanjih knjižnicah. V tem razdelku so pred-

stavljene pogosteje uporabljene zunanje knjižnice in funkcije. Predstavljenih je tudi nekaj bolj splošnih tehnologij, ki so bile uporabljene v realizaciji postopka.

### 3.5.1 Programsko okolje

Postopek je bil realiziran v programskem jeziku Python [19]. Glavni razlog za izbiro programskega jezika je bil nabor ostale programske opreme v ostalih procesih, ki je v veliki večini prav tako implementirana s programskim jezikom Python. Dodatni razlogi za izbiro programskega jezika so bili tudi hitrost prototipiranja in širok nabor knjižnic.

### 3.5.2 Zunanje knjižnice

V programskem jeziku Python se za delo z numeričnimi podatki pogosto uporablja knjižnica NumPy [18], ki jo uporabljamo tudi v implementaciji postopka, opisanega v diplomskem delu. Knjižnica med drugim omogoča optimizirane postopke za delo z velikimi podatki. Ponuja mnogo matričnih in vektorskih operacij, kar znatno izboljša berljivost kode in čas implementacije. Za branje LiDAR podatkov smo uporabili knjižnico Laspy [4], ki nam vrača podatke v obliki NumPy podatkovnih struktur.

V okviru implementacije je uporabljena tudi zaledna baza podatkov. Gre za bazo PostgreSQL, uporabljamo pa tudi razširitev PostGIS, ki nam med drugim omogoča shranjevanje geometrijskih in geografskih podatkov ter izvajanje operacij nad njimi. V okviru postopka, opisanega v diplomskem delu, uporabljamo klice razširitve PostGIS, ki nam za določene stebre omogočajo iskanje poligonov, ki jih opišejo datoteke LAS na geografskem območju. Tako lahko poiščemo datoteke LAS, v katerih se nahajajo zapisi točk v okolici opazovanih objektov.

V razdelku 3.3 je omenjena interpolacija talne mreže. Interpolacija je realizirana v knjižnici SciPy [14]. Funkcija `scipy.interpolate.griddata` omogoča interpolacijo podatkov poljubne dimenzije. Kot argumente podamo

koordinate točk  $x$  in  $y$  ter kot njihove vrednosti dimenzijo  $z$ . Funkcija kot argument prejme še točke, za katere želimo interpolacijo izračunati, in željeno metodo izračuna interpolacije.

Omenjena je tudi Delaunayeva triangulacija [11] (Razdelek 3.3). Funkcija `matplotlib.tri.triangulation` za izračun triangulacije je implementirana v zunanji knjižnici `matplotlib` [13]. Uporabimo jo za pridobitev triangulacije talne mreže, saj nam vrnjen objekt ponuja tudi možnost iskanja trikotnikov glede na podane točke. To funkcionalnost uporabimo za iskanje razdalje oblaka točk od talne površine. Eden od atributov vrnjenega objekta je tudi njegova konveksna ovojnica, ki jo pogosto uporabljamo v postopku, opisanem v diplomskem delu.

Za prilagajanje funkcije premice na množico točk uporabljamo algoritem RANSAC. Algoritem RANSAC je iterativni algoritem za določanje parametrov modela na podlagi množice točk z naključnim izbiranjem podmnožic in iskanjem soglasja. Funkcija je implementirana v knjižnici `scikit-image` Van der Walt et al. [20].



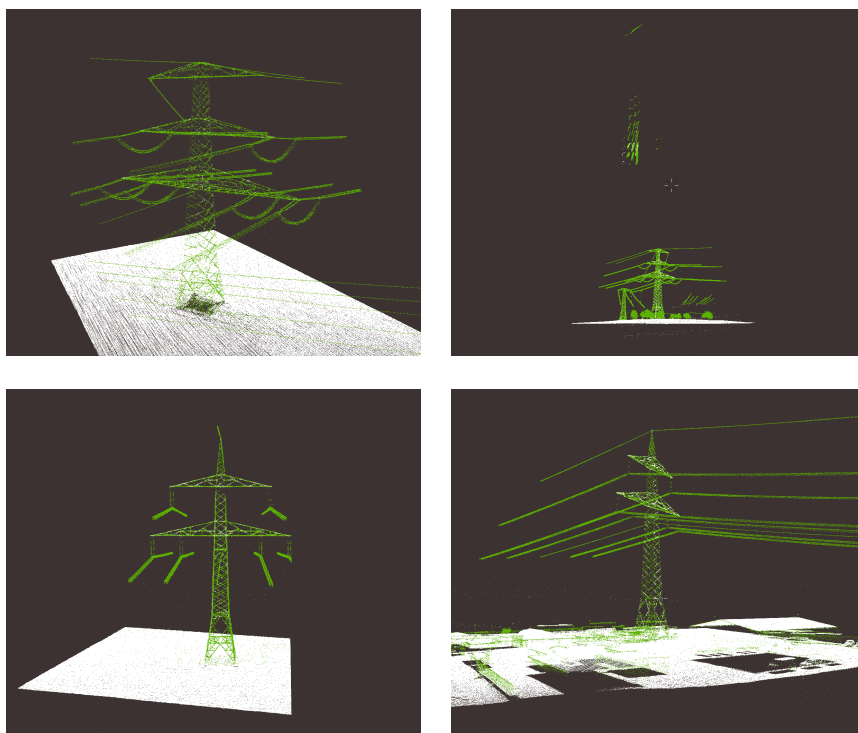
# Poglavje 4

## Rezultati

V tem poglavju bomo definirali načine merjenja napak in tipe napak, ki se pojavljajo. Na podlagi statistike o napakah bomo ovrednotili uspešnost postopka.

### 4.1 Testni podatki

Rešitev je bila preizkušena na zbirki podatkov s 157 primeri objektov, ki so geografsko razporejeni na približno 60 km dolgem odseku visokonapetostnega daljnovoda. Nekaj primerov objektov je prikazanih na sliki 4.1. Podatki se nahajajo v 5381 datotekah v skupni velikosti 157,2 GB. Velikost posamezne LAS datoteke je odvisna od števila točk, ki so zajete na geografskem območju, ki ga pokriva. Najmanjše datoteke vsebujejo zapise le nekaj točk, največja pa vsebuje 3,1 milijona zapisov točk. V zbirki podatkov se nahajajo zapisi 2,4 milijarde točk. Testni podatki še niso bili ročno očiščeni, zato je bil prisoten tudi šum. Nad podatki je bila izvedena predobdelava, torej georeferenciranje in osnovna klasifikacija. Sprva so se ročno določile karakteristične točke objektov, ročno pa je bila izvedena tudi analiza odklonov za vse objekte. Za tem je bila izvedena še samodejna določitev karakterističnih točk in njihova analiza. Za analizo uspešnosti postopka smo primerjali ročno in samodejno pridobljene rezultate.



Slika 4.1: Oblaki točk za nekaj primerov opazovanih objektov, obarvani glede na razred, ki je določen v stopnji predobdelave.

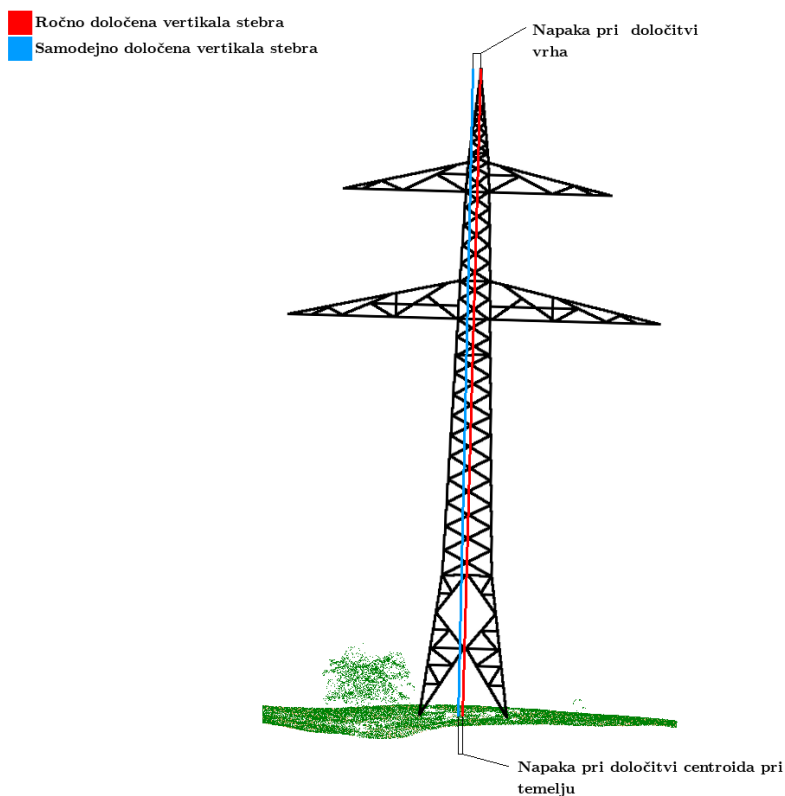
## 4.2 Merjenje napak

Za vrednotenje napak smo uporabili evklidsko metriko. Velikost napake se izračuna kot razdalja med opazovanima točkama  $p$  in  $q$  (Enačba 4.1).

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \quad (4.1)$$

Za naročnika je pomemben predvsem prečni in vzdolžni odmik vrha glede na potek daljnovoda in pa velikost odklonskega kota stebra glede na navpičnico (Slika 4.2). V okviru analize rezultatov postopka smo gledali največje absolutne oziroma relativne odmike, ne glede na smer.

Postopek smo ovrednotili glede na dve karakteristiki. V prvem koraku smo primerjali ročno določene točke s samodejno določenimi točkami, ki smo jih določili s postopkom, opisanim v diplomskem delu. Napako smo izmerili,



Slika 4.2: Skica stebra z označeno ročno in s samodejno določeno vertikalno.

kot razdaljo med točkama na horizontalni ravnini (2D preslikava).

V drugem delu analize smo ovrednotili zaznave relativnih odmikov, torej horizontalni odmik med centroidom pri vrhu in pri temelju. Velikost tovrstnega odmika določi primere, pri katerih obstaja sum odklona stebra. Pri tej analizi želimo imeti čim manj negativno določenih pravih primerov, torej primerov, kjer se pojavlja odklon stebra, postopek pa zaradi napak pri določitvi karakterističnih točk primer označi kot izpraven.

### 4.3 Analiza napak pri določitvah karakterističnih točk

V praksi sistem za zajemanje podatkov v optimalnih pogojih zagotavlja natančnost nekje med 2 in 8 cm, zaradi okoljskih dejavnikov pa prihaja tudi do večjih odstopanj. Napaka pri zajemu podatkov je odvisna predvsem od natančnosti geolociranja senzorja LiDAR. Prav tako je pri interpretaciji potrebno upoštevati lastnosti opazovanih objektov. Debelina nosilnih delov stebra je okrog 10 cm, prečno paličje pa okrog 5 cm, kar pomeni, da je napaka pri določitvi karakterističnih točk v razponu nekaj centimetrov zanemarljiva.



Slika 4.3: Primer objekta, ki ni primeren za avtomatsko analizo.

Ciljna in zadostna natančnost rezultatov je zaradi naštetih dejavnikov okrog 10 cm, glede na ročno določene rezultate (Slika 4.2). Upoštevati je potrebno tudi možnost napake pri ročni določitvi točk, zato je tudi večja napaka lahko dovoljena. Prav tako absolutna napaka pri določitvi karakterističnih točk ni nujno kritična. Če ima napaka pri določitvi karakterističnih točk podobno velikost in smer tako pri vrhu, kot pri temelju, to ne vpliva na analizo odklona. Bolj kritična je torej razlika razdalj med ročno določenima in med samodejno določenima točkama. Obstaja tudi mnogo primerov, kjer avtomatska zaznava zaradi specifičnosti objekta ni mogoča, saj je opazovani objekt unikatni ali pa sploh ni del analize in se v zbirki podatkov pojavlja zaradi topoloških razlogov. Primer takega objekta je prikazan na sliki 4.3.

### 4.3.1 Rezultati

Čeprav je povprečna napaka (Tabela 4.1) precej blizu željene natančnosti, bi bilo potrebno popraviti le devet samodejno določenih vertikal (Tabela 4.2). Odstopanja se v največji meri pojavljajo zaradi večje razgibanosti terena, nesorazmerne gostote točk na različnih straneh stebra. Včasih je prisotna celo ukrivljenost konstrukcije stebra.

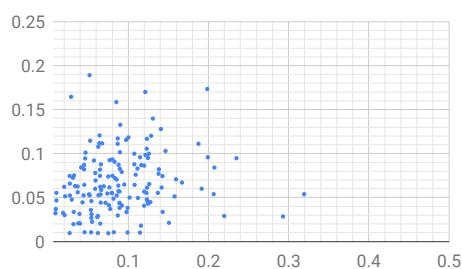
	Povprečje	Standardni odklon
<b>Vrh</b>	6,66	3,53
<b>Temelj</b>	8,90	5,20
<b>Razlika razdalj</b>	5,22	8,45

Tabela 4.1: Povprečja in standardni odkloni napak pri določitvi karakterističnih točk v centimetrih.

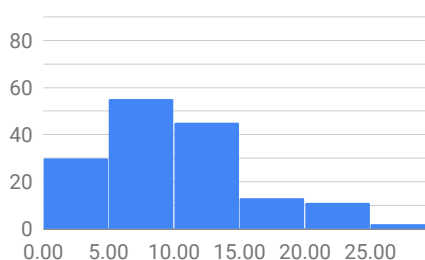
Kljub napakam pri določitvi karakterističnih točk in potrebnim popravkom se napake običajno v podobni meri izražajo na obeh določenih točkah. Efektivno to pomeni, da se relativna napaka izniči oziroma zmanjša v primeru, ko sta smer in velikost napake pri določitvi obeh točk podobni. Posledica je, da napaka pri določitvi posamezne točke v takih primerih zgolj omejeno vpliva na analizo odklona. Če primerjamo velikosti napak zgornjega in spodnjega središča, lahko opazimo, da velikost (Slika 4.4a) in tudi smer (Slika 4.4b) rahlo korelirata, kar nakazuje, da so take napake pogoste.

	Število primerov	Delež
<b>Potrebni popravki</b>	9	5,8%
<b>Zadovoljiva natančnost</b>	146	94,2%

Tabela 4.2: Uspešnost določitve karakterističnih točk stebra.



(a) Velikosti napak pri določitvi vrha proti velikostim napak pri določitvi napak centroida pri temelju stebra.



(b) Histogram kotov med smerjo napak pri določitvi centroida vrha in temelja.

Slika 4.4: Grafični prikaz korelacije napak pri določitvi karakterističnih točk.

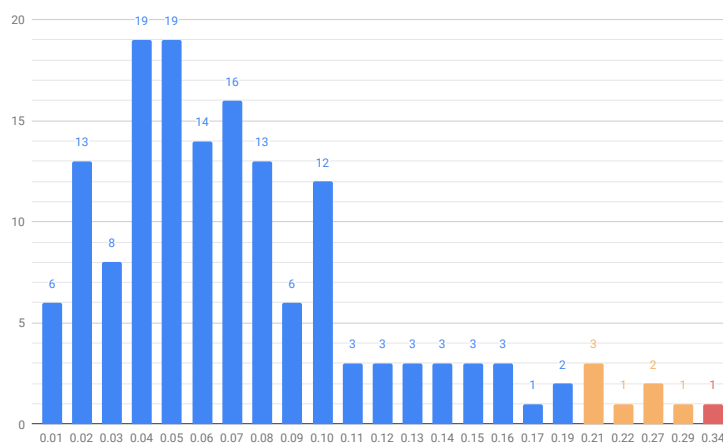
## 4.4 Analiza napak pri klasifikaciji

Postopek ponuja možnost določitve meje za samodejno označitev suma odklona vertikalne stebra od navpičnice. Velikost napake stebra se določi z velikostjo horizontalnega odmika med središčem pri temelju in vrhom stebra. Kot možne kandidate običajno označimo primere, kjer je horizontalni odmik 20 cm ali več. Označene primere je potrebno tudi ročno preveriti. Ta korak je obvezen, saj je za potrditev ali opustitev hipoteze o kritičnosti odklona stebra potrebno domensko znanje. Na končno odločitev o brezhibnosti stebra namreč vplivajo tudi drugi dejavniki, ne le velikost odmika. Nad množico izbranih kandidatov smo opravili tudi dodatno analizo napak pri določitvi karakterističnih točk po enakem postopku kot v prejšnjem poglavju.

### 4.4.1 Rezultati

Pri analizi odklonov stebrov je bilo izmed 157 zaznanih osem potencialno okvarjenih stebrov. Po ročnem pregledu strokovnjaka je bil kot kritično odklonjen steber sprejet le eden od kandidatov, predlaganih v rezultatu postopka. Rezultati so grafično prikazani na sliki 4.5. Okvarjenih stebrov, ki

bi jih algoritem označil kot brezhibne, ni bilo, kot je razvidno v tabeli 4.3. Okvarjen primer je imel zaznan odmik 34 cm, dejanski odmik pa je bil 39 cm.



Slika 4.5: Frekvenca pojavitve posamezne velikosti odklikov, zaokroženih na dve decimalni mesti. Primeri, označeni z modro, niso potrebovali dodatne obravnave. Hipoteza o napaki je bila pri oranžno označenih primerih ovržena, pri rdeče označenih pa sprejeta.

		Napoved	
		Potreben pregled	Odklon sprejemljiv
Resnica	Odklon kritičen	1	0
	Odklon sprejemljiv	7	147

Tabela 4.3: Matrika zamenjav

Pri analizi napak pri določitvi karakterističnih točk nad množico potencialnih kandidatov smo izračunali povprečno napako 5,41 cm in standardni odklon 2,77 cm za velikost odklona relativno na ročno določeno vertikalno.



# Poglavje 5

## Zaključek

V okviru diplomskega dela smo predstavili problematiko analize prostorskih podatkov visokonapetostnih omrežij. Diplomsko delo predstavi pogoste probleme, ki jih srečujemo v praksi in predlaga možne pristope za samodejno obravnavo tovrstnih problemov. V delu je opisan postopek samodejne določitve karakterističnih točk stebrov, ki so potrebne za analizo vertikalnosti stebrov omrežij. Postopek smo preizkusili z obdelavo podatkov s terena, njegova uspešnost pa je bila tudi izmerjena in ovrednotena. Za ocenitev uspešnosti postopka smo rezultate postopka primerjali z ročno pridobljenimi rezultati. Primerjali smo relativno napako pri določitvi karakterističnih točk stebrov, ovrednotili pa smo tudi delež pravilno označenih primerov stebrov, kjer obstaja sum okvare, torej odklon stebra. Rezultati kažejo, da je metoda uspešna, saj je postopek v množici kandidatov, kjer obstaja sum okvare, zajel vse dejansko okvarjene stebre, hkrati pa je bila množica označenih primerov majhna, v primerjavi s celotno množico primerov, kar občutno zmanjša obseg dela.

### 5.1 Nadaljnje delo in možne izboljšave

Pri postopku imamo opravka z delno surovimi podatki. To pomeni, da talna površina še ni dokončno določena in operiramo le s kandidati za določitev

talne površine. Obstaja sum, da se lahko pojavi šumna točka, se klasificira kot kandidat za določitev talne površine in pokvari relief. V praksi se tak primer še ni pojavil. Postopek seveda deluje tudi na očiščenih podatkih, ki so na voljo v kasnejših korakih celotne analize zajetih podatkov.

Med dejanskimi točkami in sintetično mrežo plašča osrednjega dela stebra prihaja do manjših odstopanj. Do teh odstopanj pride, ker se manjša napaka lahko nekajkrat propagira vzdolž rezin stebra. Ta napaka bistveno ne vpliva na določitev centroida, saj je običajno simetrična glede na središče. Vseeno bi bilo smiselno v prihodnosti poiskati še boljši pristop za generiranje plašča osrednjega dela stebra.

Trenutno postopek določi vertikalo stebra, saj predpostavlja ravnost stebra. V praksi se lahko zgodi, da je konstrukcija stebra ukrivljena. Morda bi lahko v prihodnosti zaznavali tudi taka odstopanja.

## 5.2 Zaključne misli

Najbolj pomembna lastnost postopka je njegova hitrost v primerjavi s človeško analizo. Za analizo enakega obsega dela je bilo doslej potrebnih več tednov dela večih zaposlenih, postopek, opisan v diplomskem delu, pa nam omogoča ogled rezultatov že v nekaj urah, odvisno od količine zajetih podatkov. Neposredna posledica tega je, da je analizo mogoče izvajati že na terenu, kar predstavlja dodatno fleksibilnost izvajalca in možnost dodatne inšpekcije na terenu. Implementacija rešitve omogoča, da se strokovnjaki z domenskim znanjem lahko posvetijo primerom, kjer je potrebna človeška presoja.

Kljub določeni meri skepticizma se je postopek v okviru projekta, kjer je bil preizkušen, izkazal izvrstno. Seveda je pri tovrstnih raziskavah samodejno dobljene rezultate obvezno potrebno še enkrat preveriti. Postopek učinkovito zmanjša obseg človeškega dela, kar je bil tudi cilj razvoja postopka.

Nedvomno je bil postopek razvit uspešno, saj se že sedaj uporablja v praksi, z njim pa so zelo zadovoljni predvsem zaposleni.





# Literatura

- [1] Paul Bourke. Calculating the area and centroid of a polygon. *Swinburne Univ. of Technology*, 1988.
- [2] R. Brennan and T L Webster. Object-oriented land cover classification of lidar-derived surfaces. *Canadian Journal of Remote Sensing*, 32(2): 162–172, 2006. doi: 10.5589/m06-015. URL <https://doi.org/10.5589/m06-015>.
- [3] R Brennan and TL Webster. Object-oriented land cover classification of lidar-derived surfaces. *Canadian journal of remote sensing*, 32(2): 162–172, 2006.
- [4] Grant Brown. Laspy. <https://github.com/laspy/laspy>, 2012–.
- [5] Amin P Charaniya, Roberto Manduchi, and Suresh K Lodha. Supervised parametric classification of aerial lidar data. In *null*, page 30. IEEE, 2004.
- [6] B. Delaunay. Sur la sphère vide. A la mémoire de Georges Voronoï. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, (6):793–800, 1934.
- [7] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

- 
- [8] Hai-Tao Fang and De-Shuang Huang. Noise reduction in lidar signal based on discrete wavelet transform. *Optics Communications*, 233(1-3): 67–76, 2004.
- [9] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL <http://doi.acm.org/10.1145/358669.358692>.
- [10] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [11] P.L. George and H. Borouchaki. *Delaunay Triangulation and Meshing: Application to Finite Elements*. Butterworth-Heinemann, 1998. ISBN 9782866016920. URL <https://books.google.si/books?id=HZGfI61PSUQC>.
- [12] Leonidas J. Guibas, Donald E. Knuth, and Micha Sharir. Randomized incremental construction of delaunay and voronoi diagrams. *Algorithmica*, 7(1):381–413, Jun 1992. ISSN 1432-0541. doi: 10.1007/BF01758770. URL <https://doi.org/10.1007/BF01758770>.
- [13] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science Engineering*, 9(3):90–95, May 2007. ISSN 1521-9615. doi: 10.1109/MCSE.2007.55.
- [14] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL <http://www.scipy.org/>. [Online; accessed 1. Jan 2019].
- [15] Anahita Khosravipour, Andrew K Skidmore, Martin Isenburg, Tiejun Wang, and Yousif A Hussin. Generating pit-free canopy height models

- from airborne lidar. *Photogrammetric Engineering & Remote Sensing*, 80(9):863–872, 2014.
- [16] Anahita Khosravipour, Andrew K Skidmore, Tiejun Wang, Martin Isenburg, and Kouros Khoshelham. Effect of slope on treetop detection using a lidar canopy height model. *ISPRS journal of photogrammetry and remote sensing*, 104:44–52, 2015.
- [17] Xuelian Meng, Nate Currit, and Kaiguang Zhao. Ground filtering algorithms for airborne lidar data: A review of critical issues. *Remote Sensing*, 2(3):833–860, 2010.
- [18] Travis Oliphant. NumPy: A guide to NumPy. USA: Trelgol Publishing, 2006–. URL <http://www.numpy.org/>. [Online; accessed 1. Jan 2019].
- [19] Guido Rossum. Python reference manual. Technical report, Amsterdam, The Netherlands, 1995.
- [20] Stéfan Van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014. ISSN 2167-8359. doi: 10.7717/peerj.453. URL <https://doi.org/10.7717/peerj.453>.
- [21] Wikipedia contributors. Random sample consensus, 2004. URL <https://en.wikipedia.org/w/index.php?title=Plagiarism&oldid=5139350>. [Online; accessed 12-June-2018].
- [22] K Zhang. Airborne lidar data processing and analysis tools. In *AGU Fall Meeting Abstracts*, 2007.