Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Jaka Kordež

Vizualizacija podatkov LiDAR na spletu

diplomsko delo

univerzitetni študijski program
prve stopnje
računalništvo in informatika

Mentor: izr. prof. dr. Matija Marolt
Somentor: as. dr. Ciril Bohak

Ljubljana, 2018

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomske naloge preučite področje vizualizacije oblakov točk. Spoznajte se s spletnim pregledovalnikom oblakov točk Potree in ga nadgradite z možnostjo senčenja točk ob prikazu glede na položaj sonca. Izdelajte algoritem, ki bo v podatkih LiDAR za območje Slovenije zapolnil točke na vodni gladini. Podatkom dodate tudi barvno informacijo iz orto-foto posnetkov in izračunajte normalo v vsaki točki. Izdelajte celovit sistem za pretvorbo podatkov in njihov prikaz in ga ovrednotite.

Najprej se zahvaljujem mentorju Matiji Maroltu in somentorju Cirilu Bohaku za vso pomoč in koristne komentarje. Hvala tudi staršem za podporo med študijem. Nazadnje gre zahvala še Alenki Vurnik in Roku Hudobivniku za lektoriranje tega dela.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
LiDAR	Light Detection And Ranging	Svetlobno zaznavanje in merjenje
PCA	Principal Component Analysis	Analiza glavnih komponent

Povzetek

Naslov: Vizualizacija podatkov LiDAR na spletu

Avtor: Jaka Kordež

Tehnologija LiDAR je skupaj z večjimi zmogljivostmi računalnikov v zadnjem času omogočila hiter razvoj na področju zaznavanja in spremljanja okolice. Podatkom, pridobljenim na ta način, pogosto manjkajo dodatne informacije kot so barva in smer normale na površino. Diplomska naloga opisuje nadgradnjo spletnega pregledovalnika oblakov točk Potree in dodajanje informacij obstoječim podatkovnim zbirkam. Oblaku točk smo najprej s pomočjo topografske karte dodali manjkajoče točke na vodni gladini. Nato smo vsaki točki dodali barvno informacijo glede na orto-foto posnetke. Zadnji korak je bil izračun normal s pomočjo analize glavnih komponent. S tem pregledovalnik točke pravilno osenči glede na definiran položaj sonca. Ta se izračuna iz ure in datuma, ki ju uporabnik določi v pregledovalniku. Na koncu smo sistem še ovrednotili. Izmerili smo hitrost pretvorbe po posameznih korakih in zmogljivost upodabljanja. Pripravili smo podatke za del Slovenije in jih objavili na spletu.

Ključne besede: lidar, grafika, vizualizacija.

Abstract

Title: Visualisation of LiDAR data on the web

Author: Jaka Kordež

LiDAR technology and better computer capabilities have recently enabled rapid development in the area of environment detection and monitoring. However, such data often lacks additional information such as colour and surface normal direction. This thesis describes an upgrade of web point cloud viewer Potree and the process of adding information to existing datasets. The first step was to add missing points on the water surfaces to the point cloud using a topographic map. Then we added colour information from the orthophoto images. The last step was to calculate normals using the principal component analysis. The viewer can correctly shade the point cloud depending on the defined position of the sun. The position is calculated from the time and date, selected by the user. The system was also analysed. We measured transformation speed and rendering performance. Prepared data comprises a part of Slovenia and is available online.

Keywords: lidar, graphics, visualization.

Poglavje 1

Uvod

V zadnjem času tehnologija napreduje z vedno večjo hitrostjo. Računalniški sistemi postajajo vse bolj zmogljivi in omogočajo obdelavo velikih količin podatkov v kratkem času. Z večjo količino podatkov lahko zajeti model (npr. del zemeljskega površja) predstavimo natančneje in z večjo stopnjo podrobnosti. To nam pride prav pri obdelavi in prikazu različnih prostorskih modelov. Bolj natančna predstavitev naše okolice omogoča boljšo predstavo o izgledu in obliki ter predstavlja podlago za nadaljnje analize in raziskave. Senzorji LiDAR nam omogočajo, da iz letala posnamemo našo okolico in jo shranimo kot oblak točk v treh dimenzijah. Pomemben del obdelave takih podatkov predstavlja vizualizacija. Zanjo potrebujemo programsko opremo, ki omogoča učinkovit in uporabniku prijazen prikaz zajetih podatkov.

Za prikaz tovrstnih podatkov obstaja odprtokoden spletni pregledovalnik Potree [34], ki ga v svoji diplomski nalogi opisuje Markus Schütz [35]. Za potrebe prikaza posnetkov zemljinega površja je Jan Gašperlin pregledovalniku dodal podporo štiriškim drevesom [12].

Omenjeno implementacijo bomo še dodatno nadgradili z izračunom položaja sonca v določenem času in ustreznega senčenja. Točke nimajo priloženih podatkov o normalah, zato jih bo potrebno izračunati. Za izračun bomo uporabili analizo glavnih komponent (angl. Principal Component Analysis - PCA) in normale zapisali poleg vrednosti o barvi. Kot podlago za delo smo upora-

bili implementacijo dodajanja barv iz posnetkov Klemena Slemenika [36] in izračun normal Petra Fajdige [11].

Zaradi nepravilnega odboja svetlobe od vodnih površin je število zajetih točk na gladini jezer, rek in morja bistveno nižje kot drugod. Potrebno bo dopolnjevanje točk na vodni gladini in popravljanje napačnih klasifikacij. Za določanje vodnih površin bomo uporabili poligone iz topografske karte Slovenije [5].

Končni cilj je postavitev celotnega sistema za pripravo podatkov, procesiranje vzorčnega območja in postavitev strežnika za javni ogled rezultatov.

1.1 Povezana literatura

Za skeniranje svojega ozemlja s tehniko LiDAR se odloča vedno več držav. Trenutno imajo svojo zbirko zaključeno države: Kanada, Danska, Finska, Nizozemska, Švica in Slovenija. Nizozemci, ki so projekt izpeljali že leta 2003, so hkrati izdelali podoben spletni pregledovalnik[1]. Prav tako so uporabili Potree, le da so pretvorili celotno zbirko v eno osmiško drevo. To so dosegli z dodatnimi skriptami, ki podatke najprej razdelijo v manjše segmente, jih pretvorijo s Potree Converter-jem in jih na koncu združijo [25]. Dodali so tudi barvno informacijo iz orto-foto posnetkov [39].

S pomočjo slovenskega oblaka točk je svojo diplomsko delo izdelal Miha Lunar [24]. Podatke je uporabil za prikaz terena s pomočjo tehnike sledenja žarkov (angl. raytracing). Pred prikazom oblak točk kvantizira in ga pretvori v množico vokslov. Na podoben način deluje tudi vizualizacija opisana v [37]. Druga možnost prikaza je s pretvorbo oblakov točk v klasičen model, definiran s trikotniki. Za tak prikaz je potreben algoritem za zaznavanje površin, končni rezultat pa deluje vsaj 50% hitreje [16].

Postopek vizualizacije oblakov točk je opisan tudi v delu [26]. Avtorja v delu opišeta združevanje različnih vrst podatkov in prikažeta možne primere uporabe. Pregledovalnik za prikaz oblakov točk iz različnih virov je

[1] `http://ahn2.pointclouds.nl/`

predstavljen tudi v poročilu [7].

Dela [27, 14, 31, 13] opisujejo algoritme za učinkovito izbiro prikazanih točk, glede na uporabnikov pogled. Bolj oddaljena območja lahko ponazorimo z manj točkami in s tem dosežemo hitrejše upodabljanje. Pri tem natančnost prikaza ostane enaka.

Podatki, zajeti s tehniko LiDAR, lahko zasedejo veliko prostora. Da se izognemo dolgotrajnemu kopiranju, je najboljša rešitev dostop na zahtevo do oddaljene shrambe, kot je to opisano v [17]. Za večji izkoristek pri hranjenju so potrebni tudi dobri algoritmi za kompresijo, opisani v delih [15, 19, 20, 29].

Na drugi strani veliko raziskovalcev uporablja oblake točk za pridobivanje informacij. Tak primer sta članka [28] in [30], ki opisujeta izdelavo natančnega modela reliefa. Na Univerzi Berkeley so s pomočjo statističnih modelov prepoznavali posamezne krošnje dreves [9], na dunajski univerzi pa stavbe [33]. Zaznane strehe stavb omogočajo tudi izračune ustreznosti za postavitev sončnih elektrarn [23, 21, 22].

Tehnika LiDAR se veliko uporablja v arheologiji, saj je končna podoba izdelana hitreje in natančneje kot z drugimi metodami, omogoča pa avtomatsko računalniško analizo [10, 38, 8].

V tem delu smo se odločili, da svojo vizualizacijo realiziramo z nadgradnjo pregledovalnika Potree. Pregledovalnik je dovolj razširjen, enostavno prilagodljiv, nizozemski projekt pa je pokazal, da je kos tudi večjim podatkovnim zbirkam.

1.2 Struktura dela

V 2. poglavju je najprej opisano obravnavano področje. Opisana so orodja in metode, ki se uporabljajo pri delu z oblaki točk. Sledi še podroben opis izdelanega sistema. Razložen je vsak korak, od podanega oblaka točk do končnega prikaza. Ovrednotenje sistema je predstavljeno v poglavju 3. Analiziramo tako časovno zahtevnost pretvorbe podatkov kot tudi zmogljivost končnega pregledovalnika. Priloženi so tudi zaslonski posnetki upodobitev

z različnimi nastavitvami. V poglavju 4 je podan zaključek in možnosti za nadaljnje delo.

Poglavje 2

Implementacija vizualizacijskega sistema

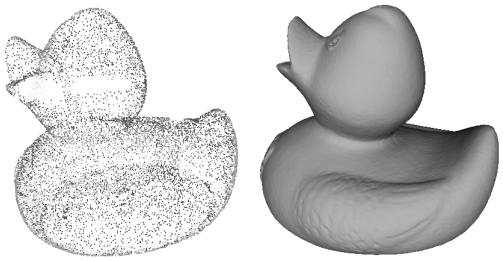
2.1 Tehnologije

2.1.1 Oblaki točk

Oblak točk sestavlja množica točk v prostoru, največkrat v treh dimenzijah. Najpogosteje jih srečamo pri uporabi 3d skenerjev. Ti namreč s pomočjo različnih metod merijo razdaljo in smer do ovire in vsako dobljeno točko dodajo oblaku. Vsaka točka ima lahko priložene dodatne informacije, kot so na primer intenziteta odboja, klasifikacija točke in smer normale. Prav slednja nam pomaga pri računalniški vizualizaciji, saj omogoča izračun osvetlitve.

2.1.2 LiDAR

LiDAR je metoda zaznavanja okolice s pomočjo laserskega žarka. Sistem odda žarek in meri čas, ki ga prepotuje od senzorja do odboja in nazaj. Iz časa izračuna razdaljo do ovire, skupaj s smerjo žarka pa dobi tudi točko v prostoru. Tehnologija se največ uporablja na področju geodezije, geologije, arhitekture in arheologije. Z njo lahko zelo natančno in učinkovito posnamemo določeno območje ali zajamemo obliko prostora.



Slika 2.1: Primerjava med modelom na levi prestavljenim z oblakom točk in na desni s trikotniki. Vir: [6]

Po naročilu Ministrstva za okolje in prostor Republike Slovenije je bilo v letih 2011 in 2012 izvedeno lasersko skeniranje in aerofotografiranje celotne Slovenije. Rezultat je oblak točk v treh gostotah: 2, 5 in 10 točk na m^2 in posnetki površja iz zraka ločljivosti 4 px/m^2. Podatki so prosto dostopni na strani Geodetske uprave RS.

Podatke, zajete s tehniko LiDAR, lahko shranimo v različnih formatih. Najbolj pogosta sta LAS in LAZ. LAS je osnovni standard združenja ASPRS[1], ki je izšel leta 2003, trenutno pa je aktualna različica 1.4 iz leta 2011 [1]. Pri delu z LAS datotekami nam pomaga zbirka orodij LAStools [2].

Da bi zmanjšali količino prostora, ki ga nek oblak točk zasede, so razvili format LAZ. Gre za brezizgubno kompresirano LAS datoteko. Delo s takimi datotekami nam omogočajo različne programske knjižnice, kot je npr. LASzip [3].

2.1.3 Potree

Potree je spletni pregledovalnik oblakov točk. Sestavljen je iz pretvornika, imenovanega Potree Converter, in spletne aplikacije Potree (slika 2.2). Oblak točk v datoteki oblike LAZ, LAS, binarni PLY, XYZ ali PTX odpremo s pretvornikom, ta pa nam izdela drevesno strukturo datotek, ki ustrezajo obliki, kot jo zna prikazati pregledovalnik. Generirane datoteke naložimo na polju-

[1] American Society for Photogrammetry and Remote Sensing

Slika 2.2: Uporabniški vmesnik pregledovalnika Potree.

ben spletni strežnik, skupaj s pregledovalnikom. Ko z brskalnikom odpremo generirano datoteko, nam Potree prebere parametre in prikaže izbran oblak točk. Potree Converter je napisan v programskem jeziku C++, Potree pregledovalnik pa v jeziku JavaScript in za izris uporablja okolje WebGL. Oba sta odprtokodna in dostopna na spletišču GitHub [4].

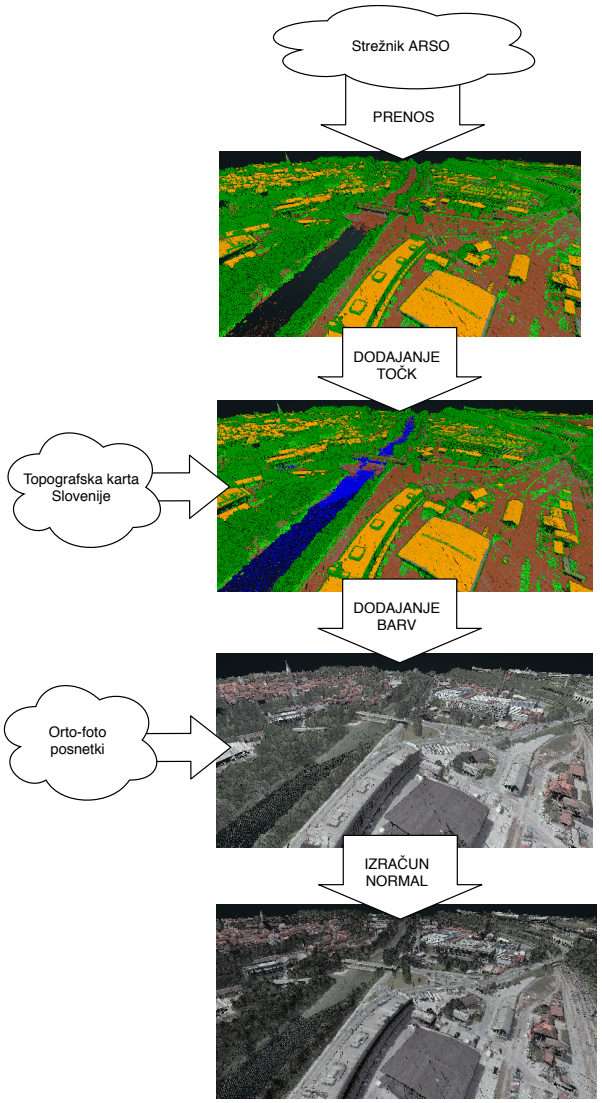
2.2 Priprava podatkov

Območje celotne Slovenije je razdeljeno v kvadrate - pakete točk s stranico dolžine 1 km, tako da so koordinate poravnane z Gauß-Krüger-jevim koordinatnim sistemom. Da bi zagotovil lažji nadzor nad procesiranjem podatkov, smo pripravo podatkov razdelili v štiri stopnje in jih povezali v celoten sistem. Obdelava paketov točk poteka cevovodno, vsak pa mora skozi sledeče stopnje:

- Prenos - oblak točk se prenese iz strežnika.
- Dodajanje točk - dodajo se točke na vodnih površinah.

- Dodajanje barve - vsaka točka dobi barvno informacijo iz soležne slikovne pike na zračnem posnetku površja.
- Dodajanje normal - izračun normal z metodo PCA.

Vsaka od stopenj sistema (prikazanega na sliki 2.3) se izvaja kot ločen proces in ima določeno vhodno in izhodno datoteko. Prve tri stopnje so implementirane v programskem jeziku C# in tečejo v okolju .NET. Izračun normal je implementiran v jeziku C++ in se prevede v strojno kodo sistema. Za izdelavo vzorčnega območja smo algoritem prevedli z prevajalnikom Visual C++. Z vsem skupaj upravlja glavni program (prav tako napisan v jeziku C#), ki zagotavlja tudi ponavljanje procesiranja v primeru napake. Za čim večjo učinkovitost se več datotek obdeluje sočasno. Vsaka od stopenj je podrobno opisana v naslednjih podpoglavjih.

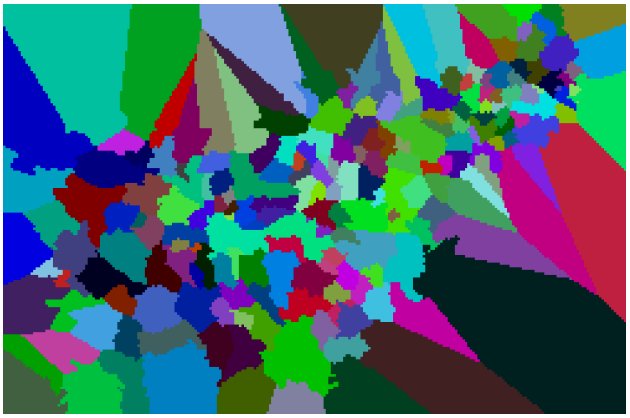


Slika 2.3: Prikaz zaporednih korakov cevovodnega delovanja sistema za pripravo podatkov.

Ko so vse datoteke opremljene z dodanimi informacijami, jih je potrebno preoblikovati v datotečno strukturo pregledovalnika Potree. Za to nalogo se uporablja program Potree Converter. Površina Slovenije znaša nekaj več kot 20.000 km^2, oblak točk pa vsebuje približno 411 milijard[2] točk. Ker Po-

[2]Preračunano iz povprečnih 20.570.000 točk na km^2

tree Converter ni namenjen tako velikim podatkovnim zbirkam, bi celotno Slovenijo težko pretvoril v enem kosu. Odločili smo se, da celotno površino razdelimo po občinah. Slovenija ima 212 občin, njihove meje pa so na voljo na strežniku Agencije Republike Slovenije za okolje. Razvili smo algoritem, ki prenese celoten zemljevid in vsako datoteko uvrsti v ustrezno občino. Rezultat je seznam datotek, ki ga uporabi program za nadzor prenosa podatkov tako, da datoteke iste občine shrani skupaj. Na sliki 2.4 so datoteke v isti občini pobarvane z enako barvo.



Slika 2.4: Prikaz datotek po pripadnosti občinam.

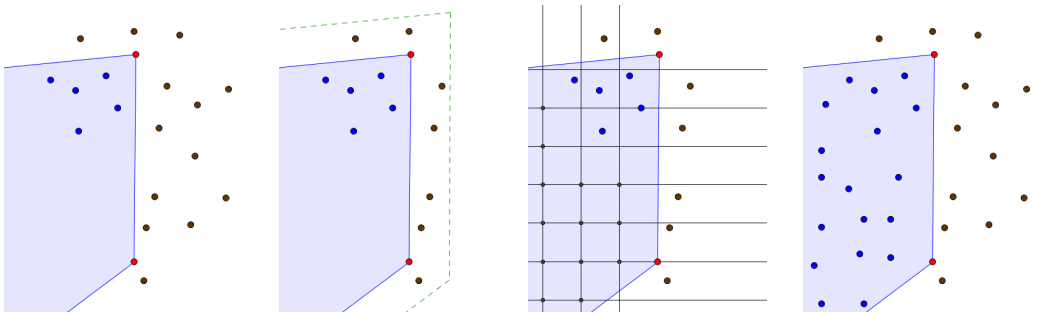
2.3 Dodajanje točk

S senzorjem LiDAR zaznavamo okolico na podlagi odboja laserskih žarkov. To pomeni, da se mora vsaj del žarka ob stiku s površino tudi odbiti nazaj proti senzorju. Tehnika deluje zelo dobro na trdnih površinah, na vodi pa zelo redko pride do ustreznega odboja. Rezultat so manjkajoče točke na jezerih in rekah. Da bi popravili oblake točk na teh območjih, moramo tam točke naknadno dodati.

Da bodo točke dodane na pravo mesto, potrebujemo podatke o vodnih površinah. V pomoč nam je topografska karta Slovenije, ki je v vektorski

digitalni obliki dostopna na strežniku Agencije Republike Slovenije za okolje[3]. Karta vsebuje tudi sloj, na katerem so vodne površine opisane v obliki dvodimenzionalnih poligonov. Vsak ima pripisano geografsko ime, širino, tip in še nekaj drugih podatkov.

Algoritem za vsak paket najprej prenese vse vodne površine za to območje v obliki JSON (slika 2.5a). Vsaka površina je predstavljena z množico točk, ki določajo poligon. Iz oblaka točk se vsakemu poligonu dodajo točke, ki glede na tloris ležijo dovolj blizu poligona in so označene kot tla (slika 2.5b). Točke, ki določajo poligon, so predstavljene z dvema koordinatama in nimajo nadmorske višine. V naslednjem koraku se zato vsaki priredi višina, ki je izračunana kot povprečje najbližjih nekaj točk iz izbranega dela oblaka točk. Da bi zapolnil manjkajoče vrzeli, program izriše navidezno mrežo točk preko



(a) (b) (c) (d)

Slika 2.5: Prikaz delovanja algoritma za dodajanje točk. (a) Poligon, ki označuje vodo, se prenese iz strežnika. (b) Obdržijo se točke, ki so klasificirane kot tla in dovolj blizu poligona. (c) Nove točke nastanejo na presečiščih mreže, ki so dovolj oddaljena od že obstoječih točk. (d) Vsaka dodana točka se naključno zamakne in doda oblaku točk.

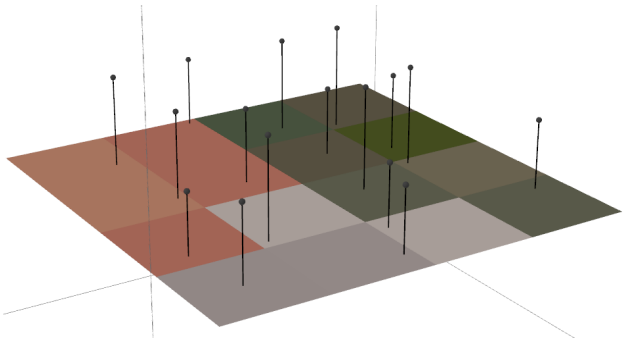
celotnega poligona z razmakom 1 m. V oblak se doda le tiste točke, ki so vsaj 1 m oddaljene od že obstoječih (slika 2.5c). Pred dodajanjem se višina določi kot povprečje višin najbližjih robnih točk poligona (označene z

[3] `http://gis.arso.gov.si/arcgis/rest/services/Topografske_karte_ARSO_nova/MapServer`

rdečo), koordinati X in Y pa se naključno zamakneta, da dobimo bolj naravno porazdelitev (slika 2.5d).

2.4 Dodajanje barv

Oblaki točk v zbirki LiDAR Ministrstva za okolje in prostor nimajo dodane informacije o barvi. To informacijo v podatke dodamo s pomočjo letalskih orto-foto posnetkov površja. Algoritem najprej prenese ustrezen posnetek iz strežnika Agencije RS za okolje[4]. Vsak posnetek ima dimenzije 2000×2000 pik, kar pomeni, da je ločljivost $4 pike/m^2$. Vsaki točki v oblaku se doda barva najbližjega slikovnega elementa iz orto-foto posnetka (slika 2.6).



Slika 2.6: Prikaz določanja ustrezne barve za posamezno točko.

2.5 Izračun normal

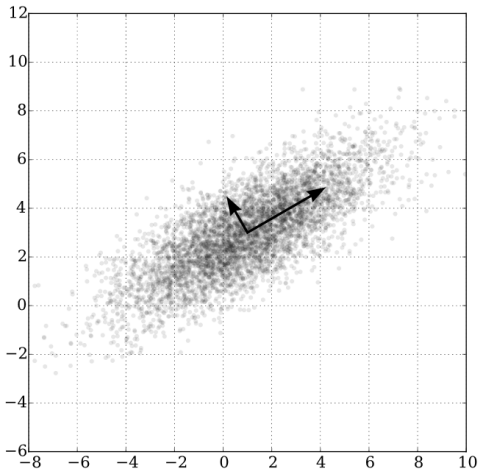
Oblak točk sam po sebi ne definira nobenih poligonov. Vsaka površina je zato predstavljena z nizom točk, ki ležijo na njeni ravnini. Da bi lahko model pravilno osenčili, je potrebno določiti normale za vsako točko. To najlažje storimo z uporabo metode PCA. Za vsako točko iz oblaka izberemo vse sosednje točke v določenem radiju in izračunamo skupno kovariančno matriko

[4] `http://gis.arso.gov.si/arcgis/rest/services/DOF_2016/MapServer`

po enačbi (2.1), kjer je k število sosednjih točk, p_i je i-ta sosednja točka in P točka normale.

$$C = \sum_{i=1}^{k} (p_i - P)(p_i - P)^T \tag{2.1}$$

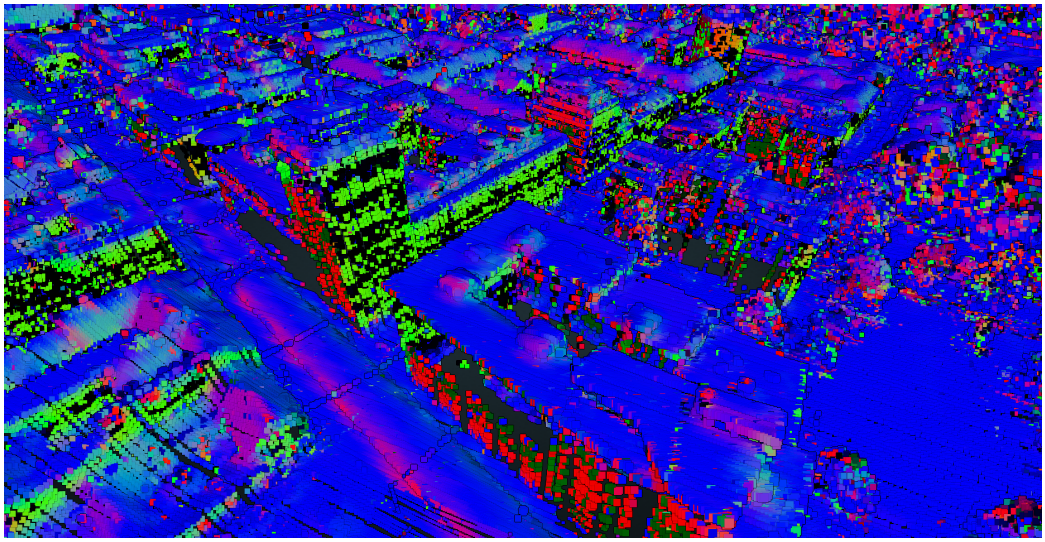
Nad to matriko se izvede razcep na lastne vrednosti in lastne vektorje. Ker so vektorji in matrika tridimenzionalni, dobimo 3 lastne vrednosti in 3 lastne vektorje, ki so med seboj ortogonalni. Tisti z največjo lastno vrednostjo kaže v smeri osi, vzdolž katere je varianca točk največja. Na drugi strani lastni vektor, katerega lastna vrednost je najmanjša, najmanj sovpada s točkami v oblaku. V primeru, ko točke predstavljajo ravno površino, to pomeni, da je pravokoten nanjo in je enak smeri normale.



Slika 2.7: Primer dobljenih lastnih vektorjev za množico točk v dveh dimenzijah. Vir: [40]

Laserski žarek se vedno odbija od površine, zato je naš model okolice votel. Pri izračunu normale torej nimamo podatka o tem, katera stran površine je zunanja in katera notranja. Lahko se zgodi, da normala kaže navzdol, to pa povzroči napačno senčenje. Podatki so posneti iz letala, zato ni mogoče, da bi kakšno površino zaznali s spodnje strani. V takem primeru zato algoritem obrne vektor v nasprotno smer.

Zadnji korak našega sistema dobi kot vhod datoteko oblike LAZ z barvnimi informacijami. Tej datoteki mora dodati izračunane normale in jo vrniti kot izhod. Datoteka LAZ vsebuje kompresiran zapis oblike LAS, a ta ne predvideva določenih polj za zapis normal. Za potrebe shranjevanja dodatnih informacij definira t. i. zapise spremenljivih dolžin (angl. Variable Length Records). Ti omogočajo poljubno dolg zapis poleg vsake točke in bi bili primerni tudi za hranjenje normal. Da bi čim bolj prihranili s prostorom, smo se raje odločili za uporabo polja za barvno informacijo. Vsaka točka ima za vsak barvni kanal rezerviranih 16 bitov, na zračnih posnetkih pa ima vsak barvni kanal samo 8 bitov informacije. Vsako od treh koordinat vektorja normale predstavimo kot predznačeno 8 bitno število in ga zapišemo v spodnjih 8 bitov informacije o barvi.

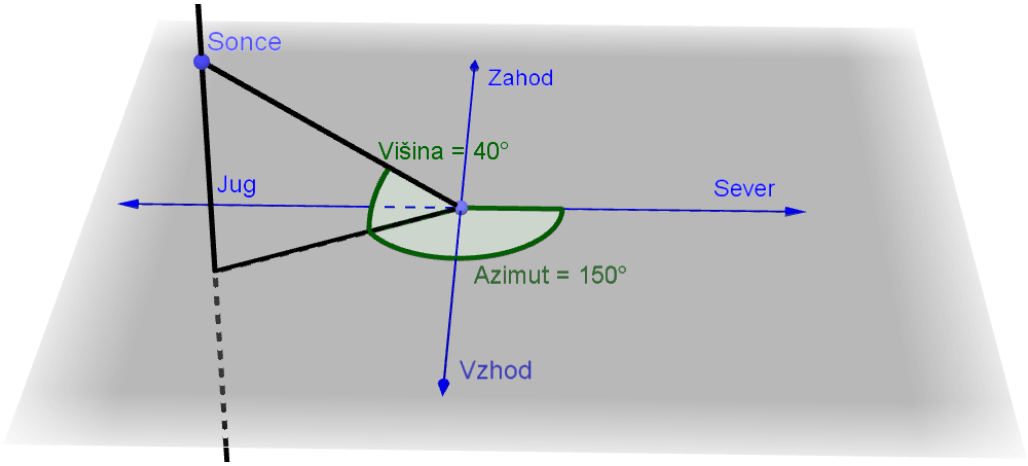


Slika 2.8: Predstavitev smeri normal z barvo.

2.6 Senčenje

Izračunane normale nam omogočajo, da površine ustrezno senčimo. Da bi bil prikaz čim bolj podoben naravni podobi, simuliramo svetlobo, ki prihaja

s sonca. Smer in jakost te svetlobe je odvisna predvsem od časa v dnevu in datuma. Za določanje smeri smo Potree pregledovalniku dodali knjižnico Suncalc[5], ki omogoča izračun azimuta in višine sonca na nebu (slika 2.9) za določen čas in geografski položaj. V tlorisni projekciji je azimut kot med smerjo proti severu in smerjo proti točki na nebu. Podobno velja za višino, ki je kot med vodoravno površino in smerjo proti točki v stranskem risu. Iz pridobljenih kotov lahko konstruiramo vektor proti soncu, ga obrnemo in



Slika 2.9: Prikaz predstavitve položaja sonca.

tako dobimo smer svetlobe, ki prihaja od sonca.

Da bi uporabnik lahko sam izbral čas osvetlitve, smo v Potree pregledovalnik dodali tudi grafične kontrolnike za izbiro ure in datuma. V glavnem meniju je nov razdelek *Time and date* z drsnikoma (prikazano na sliki 2.10). Prvi omogoča nastavitev ure na minuto natančno, drugi pa datuma.

Slika 2.10: Glavni meni spletnega pregledovalnika z novimi kontrolniki.

[5] `https://github.com/mourner/suncalc`

Potree za prikaz oblakov točk uporablja knjižnico WebGL. Vsaka točka je predstavljena z enim ogliščem (angl. vertex). Za izračun barve se uporablja senčilnik oglišč, saj želimo, da je kvadrat, ki predstavlja posamezno točko, enotne barve. Normale se preberejo kot spodnjih 8 bitov barvne informacije in prenesejo v senčnilnik, skupaj z ostalimi atributi (barva, klasifikacija ...). Senčenje se izračuna po Lambertovem modelu [18]. Ta je podoben Phongovemu [32], vendar upošteva le ambientno in razpršeno komponento. Model je predstavljen z enačbo 2.2, kjer C_m predstavlja barvo materiala, w_a delež ambientne svetlobe, R smer svetlobe in N vektor normale.

$$C = C_m(w_a + (1 - w_a)(R \bullet N)) \tag{2.2}$$

Senčilnik barvo materiala in smer normale dobi iz barve točke, smer svetlobe je določena glede na datum in uro, delež ambientne svetlobe pa je 15 %.

Poglavje 3

Ovrednotenje razvitega sistema

Rezultat celotnega diplomskega dela je delujoč sistem za prikaz površja Slovenije v obliki oblaka točk. Točke imajo dodano barvno informacijo in normale. Na vodnih površinah so manjkajoče točke dodane. Prikazovalnik dinamično nalaga datoteke iz HTTP strežnika in jih prikazuje glede na izbrane nastavitve. Izdelano območje obsega gorenjsko in obalno-kraško statistično regijo. Območje ima površino 3508 km^2, je pokrito z 68 milijard točkami in je veliko 640 GB. V nadaljevanju bo opisana analiza delovanja vsakega sklopa. Procesiranje se je izvajalo na več računalnikih. Za lažjo primerjavo so v naslednjih poglavjih opisani le časi, dobljeni na napravi s procesorjem AMD Ryzen 1700 3,1 GHz in 8 GB delovnega pomnilnika.

3.1 Priprava podatkov

Sistem za pripravo podatkov po korakih, opisanih v poglavjih 2.3, 2.4 in 2.5, je za izračun izdelanega območja porabil skupno 7148 ur. Naenkrat je na 8-jedrnem procesorju teklo šest procesov. V tabeli 3.1 so zapisani dobljeni povprečni časi, standardni odkloni in velikosti vzorcev za posamezen korak.

Izkaže se, da je najbolj zahteven drugi korak, dodajanje točk. Najbolj zamudni del je branje celotnega oblaka in ugotoavljanje pripadnosti točk vodnim površinam. V ta namen uporablja klasičen algoritem, ki preverja

Korak	Povp. čas	St. odklon	Velikost vzorca
Prenos	1 min 13 s	26 s	162
Dodajanje točk	95 min 35 s	124 min 38 s	230
Izračun normal	20 min 43 s	7 min 38 s	272
Dodajanje barv	1 min 34 s	31 s	272

Tabela 3.1: Dobljeni časi za vsak korak priprave podatkov.

parnost presečišč med žarkom iz točke in robovi poligona. Opozoriti velja tudi na velik standardni odklon pri tem koraku. Kadar na območju datoteke ni vodnih površin, program to ugotovi in zaključi že v nekaj sekundah. Takih primerov je bilo v naši testni množici kar 40%.

Seštevek vseh povprečnih časov je 119 minut in 6 sekund. Z njim lahko napovemo tudi čas obdelave za celotno Slovenijo. Ta znaša približno 1692 dni za zaporedno procesiranje oz. 6768 ur z uporabo šestih jeder.

3.2 Pretvorba v datotečno strukturo Potree

Pretvorba se je izvajala po občinah. Za vsako občino je bil ustvarjen nov oblak točk v datotečni strukturi, namenjeni pregledovalniku Potree. Potree Converter se je pri večjih občinah (več kot 3 milijarde točk) pogosto nepredvideno zaustavljal, zato smo največje občine razdelili na vzhodni in zahodni del (označeno z "V" in "Z"). Celoten postopek je trajal 53061 ur. Od vseh 31 pretvorjenih občin smo jih za analizo izbrali le 9. Sledijo površina, število točk, čas pretvarjanja in končno velikost za posamezno občino v tabeli 3.2. Velja omeniti, da sistem vsako datoteko (s površino 1 km^2) uvrsti v občino, ki jo najbolj prekriva. Zato se površina pretvorjene občine ne sklada nujno s pravo površino v naravi.

Če iz tabele 3.2 vzamemo število točk in ga delimo s časom v sekundah dobimo hitrost pretvarjanja, približno 29.000 $točk/s$. Iz tega lahko sklepamo, da bi pretvorba celotne države (411 milijard točk) zaporedoma na enem računalniku trajala približno 164 dni.

Občina	Površina	Čas	Št. točk	Velikost
Ankaran	19 km^2	1 ur 16 min	129 mio.	839 MB
Izola	36 km^2	3 ur 40 min	379 mio.	2,7 GB
Piran	64 km^2	5 ur 25 min	549 mio.	3,9 GB
Komen	118 km^2	29 ur 9 min	2.925 mio.	27,6 GB
Sežana Z	120 km^2	23 ur 25 min	2.630 mio.	25,2 GB
Kranjska Gora V	137 km^2	29 ur 12 min	3.037 mio.	30,2 GB
Divača	145 km^2	31 ur 49 min	3.454 mio.	33,7 GB
Kranj	152 km^2	28 ur 30 min	3.026 mio.	27.3 GB
Kranjska Gora Z	152 km^2	28 ur 13 min	3.036 mio.	29,0 GB

Tabela 3.2: Podatki o pretvorbi za izbrane občine

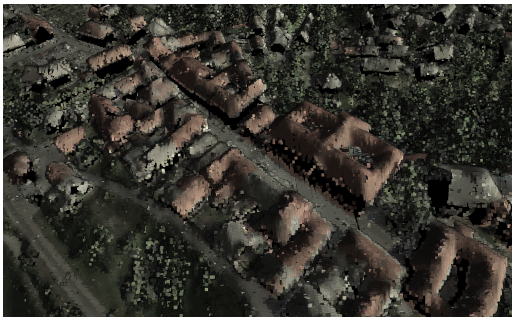
Naloga pretvornika je, da iz vseh podanih točk zgradi štiriško drevo in ga zapiše v datoteke. Algoritem je računsko zahteven, zato je bila frekvenca procesorja ozko grlo našega testnega sistema. Program za delovanje izkorišča le eno nit in dosega največ 6 % izkoristka, pri 8-jedrnem procesorju s 16 nitmi. Za hitrejše procesiranje bi torej potrebovali hitrejši procesor oz. paralelizacijo algoritma.

3.3 Senčenje

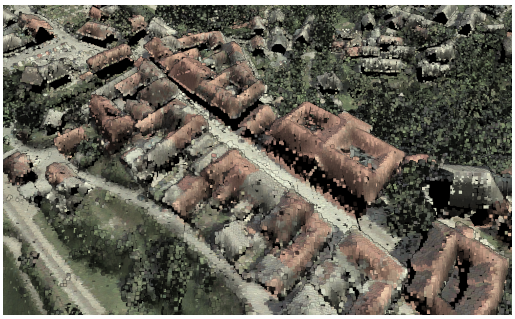
Da bi ugotovili kolikšno dodatno breme predstavlja senčenje za grafični pospeševalnik, smo izvedli še performančni test spletnega pregledovalnika. V našem testnem računalniku je Nvidia GT 430, testirali pa smo v brskalniku Firefox 61 na operacijskem sistemu Windows 10. Pregledovalnik je bil nastavljen na upodabljanje enega milijona točk. En oblak točk smo prikazali iz desetih različnih kotov in vsakič zapisali dobljeno število sličic na sekundo (angl. frames per second - FPS) za prikaz osenčenega in neosenčenega modela. V povprečju je izris neosenčenega modela potekal z 28,66 FPS, osenčenega modela pa z 27,5 FPS. Dodatno senčenje tako zateva $4,2$ % več časa.

Sledi še nekaj zaslonskih posnetkov končnega rezultata. Zraven vsakega posnetka je zapisana lokacija in uporabljene nastavitve pregledovalnika.

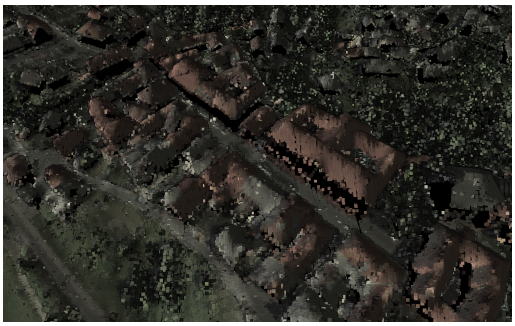
(a) Prikaz barve brez senčenja.



(b) Senčenje ob 5. uri.

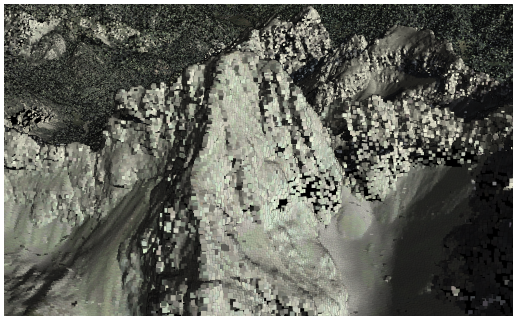


(c) Senčenje ob 12. uri.

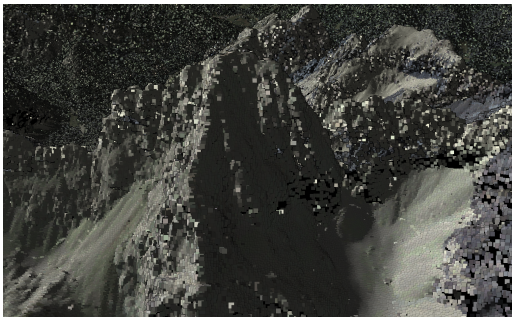


(d) Senčenje ob 18. uri.

Slika 3.1: Upodobitev Radovljice ob različnih urah v dnevu.



(a) Triglav ob 5. uri.



(b) Triglav ob 17. uri.

(c) Koper s prikazom barve.

(d) Koper s senčenjem.

Slika 3.2: Upodobitev Kopra in Triglava z različnimi nastavitvami.

Poglavje 4

Sklepne ugotovitve

Oblaku točk smo v skladu s cilji uspešno dodali manjkajoče točke na vodnih površinah, barvno informacijo iz orto-foto posnetkov in izračunane normale. Za vsakega od treh korakov je bil razvit algoritem, ki doda podatke v oblak točk. Glavni program poganja in nadzira delovanje teh algoritmov. V primeru napak omogoča tudi ponovno izvajanje. V drugem delu smo nadgradili spletni pregledovalnik Potree. Ta s pomočjo izračunanih normal, določenega datuma in ure pravilno osenči posamezne točke. Težave so se pojavile le pri uporabi orodja Potree Converter, saj to ni namenjeno pretvorbi tako velikih podatkovnih zbirk. Ena od možnih rešitev bi bila uporaba skript nizozemskih raziskovalcev [25], vendar pa te niso namenjene implementaciji štiriških dreves Jana Gašperlina [12].

Algoritem za dodajanje točk bi lahko nadgradili s prepoznavo in ustrezno obravnavo mostov in drugih objektov nad vodo. Vsi mostovi so klasificirani kot stavbe ali rastje, zato na teh mestih program ne najde ustrezne višine in ne dodaja točk. Možna rešitev bi bila uporaba sosednjih, že določenih poligonov. Napake se včasih pojavijo tudi pri raznih jezovih in slapovih, ponekod tudi zaradi napačne klasifikacije.

V prihodnje si želimo pretvoriti in objaviti podatke za celotno Slovenijo. Upamo, da bo opravljeno delo spodbudilo čim več ljudi k razpravi o tovrstni tehnologiji in pomagalo pri nadaljnjih raziskavah na tem področju.

Literatura

[1] Laser (las) file format exchange activities. https://www.asprs.org/divisions-committees/lidar-division/laser-las-file-format-exchange-activities. Dostopano: 27. 6. 2018.

[2] Lastools. https://github.com/LAStools/LAStools. Dostopano: 18. 8. 2018.

[3] Laszip - free and lossless lidar compression. https://laszip.org/. Dostopano: 27. 6. 2018.

[4] Potree. https://github.com/potree. Dostopano: 27. 6. 2018.

[5] Topografska karta republike slovenije. http://gis.arso.gov.si/arcgis/rest/services/Topografske_karte_ARSO_nova/MapServer.

[6] Abderrahim, Elmoataz, François Lozes. Pdes and variational method on 3-d surfaces and point clouds. [Dostopano 29. 8. 2018].

[7] Ciril Bohak, Byeong Hak Kim, and Min Young Kim. Web-based visualization of real-time ladar datawith support for collaboration. Technical report, KNU, Daegu, The Republic of Korea, 2017.

[8] Keith Challis, Paolo Forlin, and Mark Kincey. A generic toolkit for the visualization of archaeological features on airborne lidar elevation data. *Archaeological Prospection*, 18(4):279–289, 2011.

[9] Qi Chen, Dennis Baldocchi, Peng Gong, and Maggi Kelly. Isolating individual trees in a savanna woodland using small footprint lidar data. *Photogrammetric Engineering & Remote Sensing*, 72(8):923–932, 2006.

[10] Michael Doneus. Openness as visualization technique for interpretative mapping of airborne lidar derived digital terrain models. *Remote sensing*, 5(12):6427–6442, 2013.

[11] Peter Fajdiga. Združevanje ortofora in podatkov LiDAR v oblak točk z informacijo o barvi in normali - Poročilo seminarja. Technical report, University of Ljubljana, Faculty of Computer and Information Science, 2018.

[12] Jan Gašperlin. Web-based visualization of large terrain point cloud dataset - Seminar report. Technical report, University of Ljubljana, Faculty of Computer and Information Science, 2018.

[13] Boštjan Kovac. An approach to visualization of large data sets from lidar. 2009.

[14] Boštjan Kovač and Borut Žalik. Visualization of lidar datasets using point-based rendering technique. *Computers & Geosciences*, 36(11):1443–1450, 2010.

[15] M Kuder and B Žalik. Improving lidar compression efficiency on small packets. *Electronics Letters*, 49(25):1637–1638, 2013.

[16] Marko Kuder, Marjan Šterk, and Borut Žalik. Point-based rendering optimization with textured meshes for fast lidar visualization. *Computers & geosciences*, 59:181–190, 2013.

[17] Marko Kuder and Borut Žalik. Web-based lidar visualization with point-based rendering. In *Signal-Image Technology and Internet-Based Systems (SITIS), 2011 Seventh International Conference on*, pages 38–45. IEEE, 2011.

[18] Johann Heinrich Lambert. *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. Klett, 1760.

[19] B Lipuš and B Žalik. Lossy las file compression using uniform space division. *Electronics letters*, 48(20):1278–1279, 2012.

[20] Bogdan Lipuš and Borut Žalik. Lossless progressive compression of lidar data using hierarchical grid level distribution. *Remote Sensing Letters*, 6(3):190–198, 2015.

[21] Niko Lukač, Sebastijan Seme, Danijel Žlaus, Gorazd Štumberger, and Borut Žalik. Buildings roofs photovoltaic potential assessment based on lidar (light detection and ranging) data. *Energy*, 66:598–609, 2014.

[22] Niko Lukač and Borut Žalik. Gpu-based roofs' solar potential estimation using lidar data. *Computers & Geosciences*, 52:34–41, 2013.

[23] Niko Lukač, Danijel Žlaus, Sebastijan Seme, Borut Žalik, and Gorazd Štumberger. Rating of roofs' surfaces regarding their solar potential and suitability for pv systems, based on lidar data. *Applied energy*, 102:803–812, 2013.

[24] Miha Lunar. *Porazdeljeno sledenje žarkov za upodabljanje lasersko zajetih prostorskih podatkov*. PhD thesis, Univerza v Ljubljani, 2016.

[25] Oscar Martinez-Rubi, Stefan Verhoeven, Maarten Van Meersbergen, M Schûtz, Peter Van Oosterom, Romulo Gonçalves, and Theo Tijssen. Taming the beast: Free and open-source massive point cloud web visualization. In *Capturing Reality Forum 2015, 23-25 November 2015, Salzburg, Austria*. The Servey Association, 2015.

[26] Robert J McGaughey and Ward W Carson. Fusing lidar data, photographs, and other data using 2d and 3d visualization techniques. *Proceedings of terrain data: applications and visualization–making the Connection*, pages 28–30, 2003.

[27] Domen Mongus, S Pečnik, and B Žalik. Efficient visualization of lidar datasets. In *2009 International Conference on Optical Instruments and Technology: Optoelectronic Imaging and Process Technology*, volume 7513, page 75130M. International Society for Optics and Photonics, 2009.

[28] Domen Mongus, Mihaela Triglav Čekada, and Borut Žalik. Analiza samodejne metode za generiranje digitalnih modelov reliefa iz podatkov lidar na območju slovenije. *Geodetski vestnik*, 57(2), 2013.

[29] Domen Mongus and Borut Žalik. Efficient method for lossless lidar data compression. *International journal of remote sensing*, 32(9):2507–2518, 2011.

[30] Saso Pecnik, Domen Mongus, and Borut Zalik. System for digital elevation model generation from lidar data. *International Multidisciplinary Scientific GeoConference: SGEM: Surveying Geology & mining Ecology Management*, 1:957, 2010.

[31] Sašo Pečnik and Borut Žalik. Real-time visualization using gpu-accelerated filtering of lidar data. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 8(12):2108–2112, 2014.

[32] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[33] Franz Rottensteiner and Christian Briese. A new method for building extraction in urban areas from high-resolution lidar data. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):295–301, 2002.

[34] Markus Schütz. Potree. `http://potree.org/`. Dostopano: 27. 6. 2018.

[35] Markus Schütz. Potree: Rendering large point clouds in web browsers. 2016.

[36] Matej Slemenik. Združevanje ortofotov in lidar podatkov za izgradnjo oblaka točk z informacijami o barvi in normali - poročilo seminarja. Technical report, University of Ljubljana, Faculty of Computer and Information Science, 2018.

[37] Jason Stoker. Visualization of multiple-return lidar data: Using voxels. *Photogramm. Eng. Remote Sens*, 75(2):109–112, 2009.

[38] Benjamin Štular, Žiga Kokalj, Krištof Oštir, and Laure Nuninger. Visualization of lidar-derived relief models for detection of archaeological features. *Journal of archaeological science*, 39(11):3354–3360, 2012.

[39] Adriaan van Natijne and RC Lindenbergh. Web based visualisation of 3d radar and lidar data. 2017.

[40] Wikipedia, the free encyclopedia. Pca of a multivariate gaussian distribution. [Dostopano 22. 8. 2018].