

Spletno upodabljanje LiDAR podatkov Slovenije z dodano barvno informacijo in senčenjem

Jaka Kordež, Ciril Bohak

Fakulteta za računalništvo in informatiko, Univerza v Ljubljani
E-pošta: jk5456@student.uni-lj.si, ciril.bohak@fri.uni-lj.si

Web-Based visualization of Slovenian LiDAR data with added color information and shading

In this paper we present an adaptation of web-based point-cloud data visualization framework Potree for use with Slovenian LiDAR data. We enriched LiDAR data with color information obtained from Ortho-Photo data. We supplemented the dataset with artificial points, where data was missing (water surfaces). For better spatial perception we also calculate the normals for individual points in the data and use them for shading. We evaluate the presented sistem step by step and present the final results.

Povzetek

V članku predstavimo prilagoditev spletnega ogrodja Potree za prikaz podatkov v obliki oblakov točk za uporabo s slovenskimi podatki LiDAR. V podatke smo dodali barvno informacijo iz zbirke orto-foto posnetkov. Podatke smo dopolnili z umetno ustvarjenimi točkami, kjer so bila manjkajoča območja (vodne površine). Za boljše prostorsko predstavo pri pregledovanju podatkov smo za posamezne točke v zbirki izračunali normale, ki jih uporabimo pri izračunu osvetlitve. Končni sistem smo korak za korakom performančno ovrednotili in predstavili končne rezultate.

1 Uvod

V zadnjem času se je zaradi napredka tehnologije zelo povečala količina zajetih okoljskih podatkov na različnih področjih. To velja tudi za področje geodezije, geografije in geologije, kjer si raziskovalci vse bolj pogosto pri snemanju zemeljskega površja pomagajo tudi s tehnologijo LiDAR (angl. Light Imaging, Detection and Ranging) [4]. Z uporabo senzorjev LiDAR, nameščenih na letala, lahko pridobimo izredno natančno informacijo o obliki površja, kot tudi o vrsti podlage od katere se laserski žarek odbije. Zbirko takšnih podatkov so pripravili tudi pri Agenciji Republike Slovenije za okolje [1].

Uporabnost takšne zbirke pa se močno poveča kadar lahko uporabniku njeno vsebino predstavimo na enostaven in prijazen način. Eden takšnih pristopov je vizualizacije takšne zbirke podatkov. Za široko dostopnost je dandanes smiselno uporabiti spletne tehnologije, saj so podprte na širokem naboru naprav (od namiznih računalnikov pa do mobilnih telefonov). Vizualizacija takšne

zbirke podatkov v obliki kock z robom 1 m z uporabo upodobljevalne metode sledenja žarkom je predstavljena v delu [6], kjer pa sistem ne omogoča interaktivnega vpogleda zaradi zahtevnega izračuna upodobitve. Za vizualizacijo takšnih podatkov je zaradi interaktivnosti smiselno uporabiti strojno pospešen izris. To nam omogoča uporaba vmesnika WebGL [5], ki ga uporablja tudi ogrodje Potree¹ [7], namenjeno prav vizualizaciji oblakov točk na spletu.

Ogrodje Potree je namenjeno prikazu podatkov, ki imajo točke razporejene po širšem 3D prostoru in za optimizacijo prikaza podatkov uporablja strukturo osmiškega drevesa (angl. octree). Za primer podatkov v že omenjeni zbirki velja, da so točke večinoma razporejene v tankem pasu v bližini površja in je posledično uporaba takšne podatkovne strukture nepotrebna in celo potratna, saj delitev prostora v tretji dimenziji ni potrebna. Prav to pa naslavlja v svojem delu Jan Gašperlin [3], ki je ogrodje Potree za namene prikaza površja prilagodil z implementacijo štiriškega drevesa za delitev prostora, kar se izkaže za dobro odločitev, tako v performančnem smislu kot pri prihranku prostora.

Zbirka podatkov LiDAR za posamezno točko vsebuje podatke o položaju, intenziteti odboja, zaporedni številki odboja in druge. Za dobro vizualizacijo površja pa bi bilo smiselno za posamezno točko hraniti tudi barvno informacijo. Ta problem je v svojem delu naslovil Matej Slemenik [8], ki je točkam pripisal barvo na podlagi poravnanih barvnih orto-foto posnetkov. Še boljše vizualizacijo pa lahko dosežemo z uporabo primerne metode za senčenje. Pri tem nam močno pomaga informacija o normalah, ki definira odboj svetlobe. To problematiko je v svojem delu naslovil Peter Fajdiga [2], ki za izračun normal v posamezni točki zbirke uporabi metodo analize glavnih komponent (angl. Principal Component Analysis - PCA) sosednjih točk. V našem sistemu to izkoristimo za možnost izračuna osvetlitve glede na položaj sonca, izbranega datuma in ure.

Za še boljši končni izgled vizualiziranih podatkov pa je smiselno nasloviti tudi napake oz. pomanjkljivosti v podatkih. Tako v pričujočem delu predstavimo tudi metodo za dopolnitev podatkov na območjih vodnih površin, kjer zaradi lastnosti površine površja senzorji podatkov niso zajeli. Za določanje vodnih površin uporabimo po-

¹<http://potree.org/>

datke iz Atlasa voda Ministrstva za okolja in prostor RS².

V članku v naslednjem poglavju predstavimo razviti sistem in njegove posamezne korake, v poglavju 3 razvit sistem performančno ovrednotimo ter predstavimo rezultate, na koncu pa v poglavju 4 podamo zaključke in možne korake za nadaljnje delo.

2 Spletni sistem za vizualizacijo terena

Celoten sistem je sestavljen iz dveh delov: (a) dela za pripravo podatkov in (b) uporabniškega dela za vizualizacijo podatkov. Prvi del teče na strežniku, kjer se podatki iterativno pripravljajo v obliko primerno za vizualizacijo. Tako pripravljene podatki so nato na voljo uporabniškemu delu sistema namenjenega vizualizaciji. Uporabniški del je razvit kot spletna aplikacija.

2.1 Priprava podatkov

Območje celotne Slovenije razdelimo v pakete, ki jih sestavljajo kvadrati s stranico dolžine 1 km tako, da so koordinate poravnane z Gauß-Krüger-jevim koordinatnim sistemom. Da bi zagotovili lažji nadzor nad procesiranjem podatkov, je sistem sestavljen iz posameznih stopenj, obdelava podatkov pa poteka cevovodno. Vsak paket mora skozi naslednje stopnje:

- *Prenos* - oblak točk se prenese iz strežnika.
- *Dodajanje točk* - dodajo se manjkajoče točke na vodnih površinah.
- *Dodajanje barve* - vsaka točka dobi barvno informacijo iz ustrezne slikovne pike na zračne posnetku površja.
- *Izračun normal* - izračun normal z metodo PCA.

Vsaka od stopenj sistema (prikazanega na sliki 1) se izvaja kot ločen proces in ima določeno vhodno in izhodno datoteko. Z vsem skupaj upravlja glavni program, ki zagotavlja tudi ponavljanje procesiranja v primeru napak. Za čim večjo učinkovitost se več paketov obdeluje sočasno. Vsaka od stopenj je podrobneje opisana v naslednjih podpoglavjih.

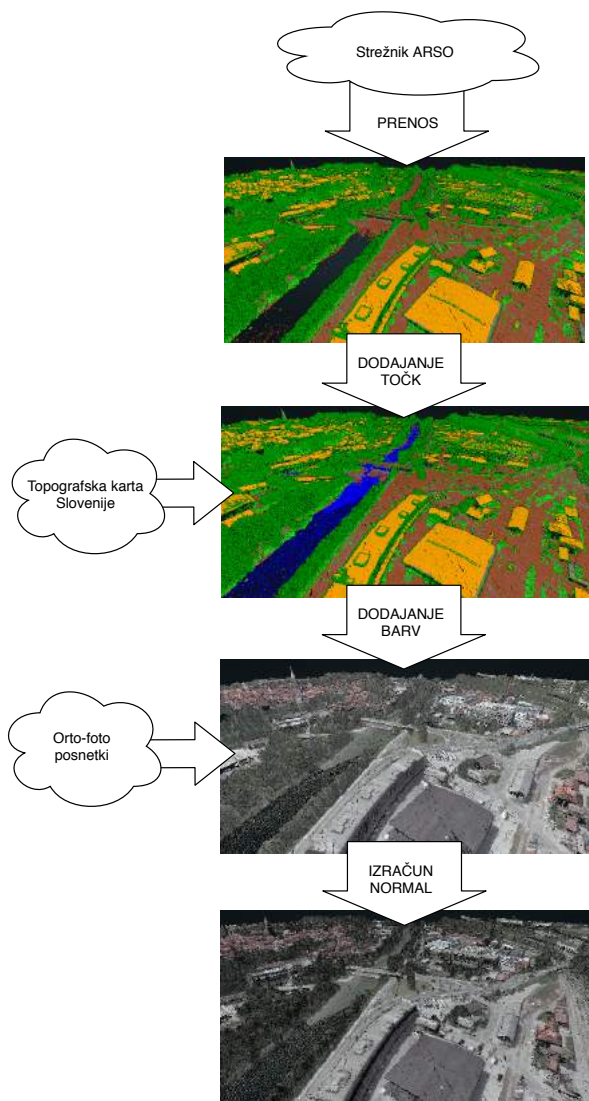
2.1.1 Dodajanje manjkajočih točk

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 torej naknadno dodati.

Da bi točke dodali 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 okolje³. Karta vsebuje tudi sloj na katerem so vodne površine

²<http://www.evode.gov.si/atlas-voda/>

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



Slika 1: Prikaz zaporednih korakov cevovodnega delovanja sistema za pripravo podatkov.

opisane v obliki dvodimenzionalnih poligonov. Vsak ima pripisano geografsko ime, širino, tip in še druge podatke.

Algoritem za vsak paket najprej prenese vse vodne površine za to območje v obliki JSON. Vsaka površina je predstavljena z množico točk, ki določajo poligon. Iz oblaka točk se vsakemu poligonu dodajo točke, ki glede na tloris ležijo znotraj poligona in so označene kot tla. Točke, ki določajo poligon so predstavljene z dvema koordinatama in nimajo nadmorske višine. V naslednjem koraku se zato vsaki priredi višina izračunana kot povprečje najbližjih nekaj točk iz izbranega dela oblaka točk. Da bi zapolnil manjkajoče vrzeli, program izriše navidežno mrežo točk preko celotnega poligona z razmakom 1 m. V oblak se doda le tiste točke, ki so vsaj 1 m oddaljene od že obstoječih. Pred dodajanjem se višina določi kot povprečje višin najbližjih robnih točk poligona, koordinati X in Y pa se naključno zamakneta da dobimo bolj naravno porazdelitev.

2.1.2 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 prenese ustrezen posnetek iz strežnika Agencije RS za okolje⁴ in vsaki točki v oblaku doda barvo najbližjega slikovnega elementa iz orto-foto posnetka.

2.1.3 Senčenje

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 po enačbi (1), kjer je k število sosednjih točk, p_i je i -ta sosednja točka in P pa točka normale.

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

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. Normala je tisti lastni vektor, katerega lastna vrednost je najmanjša, saj najmanj sovpada z sosednjimi točkami.

2.2 Vizualizacija podatkov

Uporabniški del aplikacije je razvit kot spletna aplikacija in za prikaz uporablja prilagojeno različico že omenjenega ogrodja Potree. Ogrodje Potree smo prilagodili tako, da namesto osmiške delitve prostora zaradi učinkovitosti uporablja štiriško delitev.

Na strežniku pripravljene podatki izbranega območja se pretočijo k uporabniku, kjer se uporabniku prikažejo. Izris je strojno pospešen in izrablja vmesnik WebGL. Spletna aplikacija omogoča prikaz različnih lastnosti terena shranjenih v podatkih, podpira pa tudi njihovo združevanje. Nekaj primerov vizualizacije istih podatkov je prikazanih na sliki 2.

3 Evalvacija in rezultati

Da bi ocenili hitrost delovanja celotnega sistema smo poskusno pognali celoten sistem in merili čas izračuna. V tabeli 1 so prikazani povprečni izmerjeni časi posameznih stopenj cevovoda. Izmerjeni so bili na vzorcu 12 različnih datotek. Povprečno število točk na datoteko je bilo 14.829.274. Celotna obdelava paketov je v povprečju trajala 10 minut in 29 sekund. To pomeni, da je bilo na sekundo obdelanih 22.998 točk. Testiranje je potekalo na procesorju AMD Ryzen 1700 pri frekvenci 3,1 GHz.

Ker vsaka datoteka pokriva območje 1 km^2 , površina Slovenije pa je nekaj več kot 20.000 km^2 , pričakujemo, da bo celotno območje obdelano v približno 3605 urah.

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



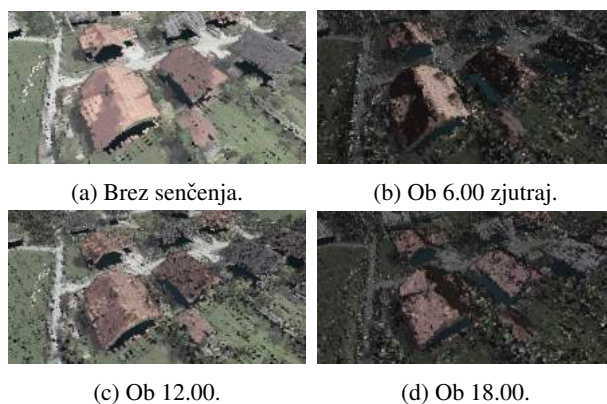
Slika 2: Prikaz različnih lastnosti shranjenih za posamezno točko in njihove kombinacije.

Tabela 1: Povprečni časi stopenj cevovoda

KORAK	ČAS
Prenos	35 s
Dodajanje točk	46 s
Dodajanje barv	47 s
Izračun normal	8 min 42 s
Skupaj	10 min 49 s

Program lahko izkoristi več jeder sistema s tem, da obdeluje več datotek hkrati. Na ta način lahko z uporabo npr. 8 jeder zaključimo v približno 450 urah oz. v 18 dneh in 18 urah.

Z izračunanimi normalami lahko v koraku senčenja izračunamo osvetlitev v različnih delih dneva. Na sliki 3 je tako prikazan del podatkov brez izračuna senčenja in nato še z izračunom senčenja ob treh delih dneva: zjutraj, popoldne in zvečer.



Slika 3: Senčenje terena ob različnih urah v dnevu.

Na slikah 4a - 4d je razviden končni rezultat vizualizacije podatkov obdelanih s predstavljenim sistemom za izbrana področja Slovenije.



(a) Pogled na Ljubljano.



(b) Pogled na Ljubljano.



(c) Pogled na Piran.



(d) Pogled na Triglav.

Slika 4: Prikaz končnih upodobitev podatkov pripravljenih in vizualiziranih s predstavljenim sistemom.

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 zabeležili 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.

4 Zaključki in nadaljnje delo

Predstavljen sistem je bil razvit z namenom izdelave dostopnega orodja za prikaz izbranih geodetskih podatkov za območje Slovenije. Uporabniški del razvitega sistema omogoča vpogled v podatke tako na namiznem računalniku kot na mobilnih telefonih in tablicah. Uporabniku ponuja interaktiven, odziven in intuitiven vmesnik, kjer lahko izbrane parametre prilagaja med samo uporabo.

V okviru nadaljnjega dela želimo s predstavljenim sistemom pripraviti podatke za celotno Slovenijo in jih ponuditi na vpogled širši javnosti. Prav tako želimo sistem nadgraditi z dodatnimi funkcijami in podatke dodatno avtomatsko obdelati kjer prihaja do artefaktov (npr. pri mostovih in jezovih). Sistem je enostavno prilagodljiv in nadgradljiv in upamo, da pritegne širši nabor uporabnikov.

Literatura

- [1] Geodetski inštitut slovenije. izvedba laserskega skeniranja slovenije. blok 35 – tehnično poročilo o izdelavi izdelkov. geodetski inštitut slovenije. Technical report, 2015.
- [2] 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.
- [3] 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.
- [4] Norbert Haala, Michael Peter, Jens Kremer, and Graham Hunter. Mobile LiDAR Mapping for 3D Point Cloud Collection in Urban Areas — A Performance Test. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37:1119–1127, 2008.
- [5] Dean Jackson and Jeff Gilbert. WebGL specification. Technical report, The Khronos Group Inc., 2015.
- [6] Miha Lunar, Ciril Bohak, and Matija Marolt. Distributed ray tracing for rendering voxelized lidar geospatial data. In *Zbornik petindvajsete mednarodne Elektrotehniške in računalniške konference ERK 2016, 19. - 21. september 2016, Portorož, Slovenija*, pages 55–58, 2016.
- [7] Markus Schütz. Potree: Rendering large point clouds in web browsers. Master's thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, 2016.
- [8] 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.