

# PORAZDELJENO UPODABLJANJE VOKSELIZIRANIH PODATKOV LIDAR

# DISTRIBUTED RENDERING OF VOXELIZED LIDAR DATA

*Miha Lunar, Ciril Bobak, Matija Marolt*

UDK: 528.83  
Klasifikacija prispevka po COBISS.SI: 1.04  
Prispelo: 11. 11. 2016  
Sprejeto: 9. 12. 2016

DOI: 10.15292/geodetski-vestnik.2016.04.615-626  
PROFESSIONAL ARTICLE  
Received: 11. 11. 2016  
Accepted: 9. 12. 2016

## IZVLEČEK

V članku predstavimo sistem za upodabljanje 3D-svetov, definiranih v vokselskem prostoru. Poleg upodabljanja sistem omogoča ustvarjanje takšnih svetov iz podatkov LiDAR in letalskih posnetkov površja, pri čemer se podatki diskretizirajo z ločljivostjo enega metra. Pri ustvarjanju svetov sistem uporabi osnovne pomenske označbe podatkov LiDAR (tla, rastje, stavbe ipd.), uvede pa tudi dodatne pomenske označbe, pridobljene na podlagi barvne informacije iz ustrezno poravnanih letalskih posnetkov (voda, betonske površine ipd.). Zaradi večje dostopnosti uporabnikom omogoča tako lokalno uporabo kot uporabo prek spleta. Za čim hitrejšo delovanje podpira uporabo večjedrnih in večprocesorskih računalnikov, omogoča pa tudi povezovanje več sistemov v gručo in s tem hitrejšo končno upodabljanje. V članku predstavimo rezultate – upodobitve izbranih območij – in možnosti uporabe razvitega sistema za druge namene.

## ABSTRACT

*In this paper, a system for rendering voxel (3D pixel)-based worlds is presented. The system enables the generation of such worlds from LiDAR and orthophoto data by discretizing LiDAR data with a resolution of 1 m. In world generation, the embedded basic semantic descriptors of LiDAR data are used, enhanced with additional classes acquired from colour information in geospatially aligned orthophoto images. The system allows local use as well as web-based use and supports multi-core and multi-processor hardware, as well as the clustering of multiple instances for faster rendering. We also present the results (renderings of selected areas) and describe other possible uses of the presented system.*

## KLJUČNE BESEDE

upodabljanje, vokseli, LiDAR, oblaki točk, ortofoto, sledenje žarkov

## KEY WORDS

rendering, voxels, LiDAR, point cloud, orthophoto, ray tracing

## 1 UVOD

Napredek v razvoju zemeljskih opazovanj omogoča zajem vse večje količine podrobnih podatkov, ki so uporabni na številnih področjih, kot so geodezija, geografija, geologija, in tudi drugod. Tehnologija LiDAR (angl. *Light Detection and Ranging*) omogoča zajem različnih plasti površja (od vrhov krošenj do trdnih tal) s precejšnjo natančnostjo. Od konca leta 2015 so podatki LiDAR za celotno Slovenijo (Geodetski inštitut Slovenije, 2015) javno dostopni in na voljo za splošno rabo. Dostopni so na spletnih straneh Agencije Republike Slovenije za okolje<sup>1</sup> (ARSO). Sestavljeni so iz oblakov točk (angl. *point cloud*), pri čemer posamezne točke poleg 3D-lokacije hranijo pomensko označbo površja, ki ga predstavljajo (tla, nizko/srednje/visoko rastje, stavbe, morje).

Za lažjo predstavo, kaj točno podatki zajemajo, jih je smiselno ustrezno upodobiti. Poleg podatkov LiDAR je za Slovenijo dostopen tudi državni ortofoto, ki je georefenciran fotografski prikaz površja in je na voljo prek portala Prostor Geodetske uprave RS<sup>2</sup>. Ideja, iz katere smo izhajali v okviru predstavljenega dela, je prikaz velikih realnih svetov v navidezni resničnosti, natančneje, v računalniških igrah. Najbolj priljubljena igra, ki igralca potopi v brezmejno, digitalno generirana prostranstva, je Minecraft<sup>3</sup>, prikazana je na sliki 1. Svet v igri je generiran samodejno in je sestavljen iz vokslov, odetih v različne teksture, ki jih pomensko opredeljujejo (na primer trava, grmovje, drevo, voda ipd.). Poleg samodejno generiranih svetov za Minecraft so veliko lastnih svetov ustvarili tudi sami igralci. Prav priljubljenost igre in njenega načina prikaza površja je glavni povod za razvoj lastnega upodobitvenega sistema, ki izkorišča zbirki podatkov LiDAR in državnega ortofota, izdelanega iz letalskih posnetkov Slovenije. Cilj je bil razviti sistem upodabljanja dejanske pokrajine na podlagi zajetih podatkov, z možnostjo uporabe upodobitvene tehnike metanja žarkov z naprednimi opcijami senčenja in osvetljevanja.



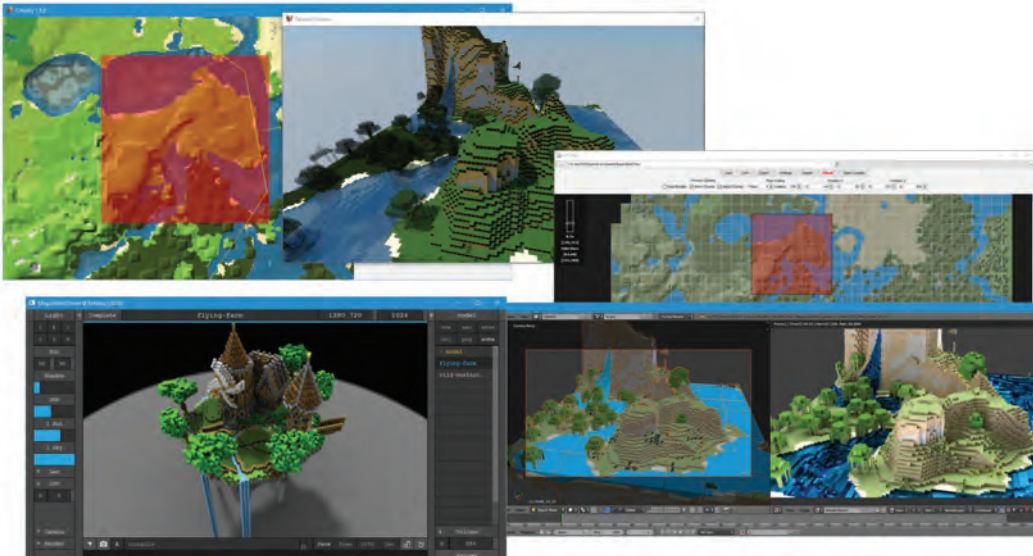
Slika 1: Igra Minecraft.

<sup>1</sup> <http://evode.arso.gov.si/indexd022.html>

<sup>2</sup> <http://www.e-prostor.gov.si/>

<sup>3</sup> <https://minecraft.net/>

Tako za upodabljanje svetov, predstavljenih z vokslami, kot za upodabljanje podatkov LiDAR obstaja precej rešitev, ki pa niso primerne za želeni način upodabljanja. Vokselne svetove je mogoče upodabljeti s programi za 3D-modeliranje (na primer Blender<sup>4</sup>, 3D Studio Max<sup>5</sup>, Maya<sup>6</sup> ipd.), v katere lahko uvozimo pretvorjene Minecraftove svetove z odprtokodnimi orodji (na primer mcobj<sup>7</sup>, mc2obj<sup>8</sup>, Jmc2Obj<sup>9</sup>). Prav tako obstajajo namenska orodja, kot sta Chunky<sup>10</sup>, ki sicer omogoča realistični izračun osvetlitve, a je pri upodabljanju večjih območij izredno počasen, in MagicaVoxel<sup>11</sup>, ki omogoča precej učinkovito upodabljanje svetov do velikosti 2048<sup>3</sup>. Vsem navedenim rešitvam, prikazanim na sliki 2, je skupno dejstvo, da ne omogočajo upodabljanja resnično velikih delov sveta, kar je cilj naše rešitve.



Slika 2: Chunky, Blender z Jmc2Obj, MagicaVoxel.

Obstaja tudi precej rešitev za neposredno upodabljanje podatkov LiDAR, kot na primer orodja zbirke LAStools<sup>12</sup> in LiDAR LiVE<sup>13</sup>, ki omogočajo neposredno upodobitev oblaka točk in lahko različno obarvajo točke z različnimi oznakami (tla, rastje, stavbe ipd.), kar je razvidno tudi s slike 3. Njihove glavne pomanjkljivosti so nezmožnost podrobne določitve videza posameznih tipov podatkov, nezmožnost popravil napak v podatkih ter nezmožnost uporabe naprednih osvetlitvenih in upodobitvenih tehnik. Ravno zaradi navedenih pomanjkljivosti tovrstne aplikacije niso primerne za doseg zastavljenih ciljev. Podatke LiDAR lahko uporabimo tudi za generiranje digitalnega modela reliefa (Mongus et al., 2013; Pečnik et al., 2009), ki se uporablja na številnih področjih in je lahko tudi podlaga za izgradnjo prikaza nekega območja.

<sup>4</sup> <https://www.blender.org/>

<sup>5</sup> <http://www.autodesk.com/products/3ds-max/>

<sup>6</sup> <http://www.autodesk.com/products/maya/>

<sup>7</sup> <https://github.com/quag/mcobj>

<sup>8</sup> <https://github.com/FalconNL/mc2obj>

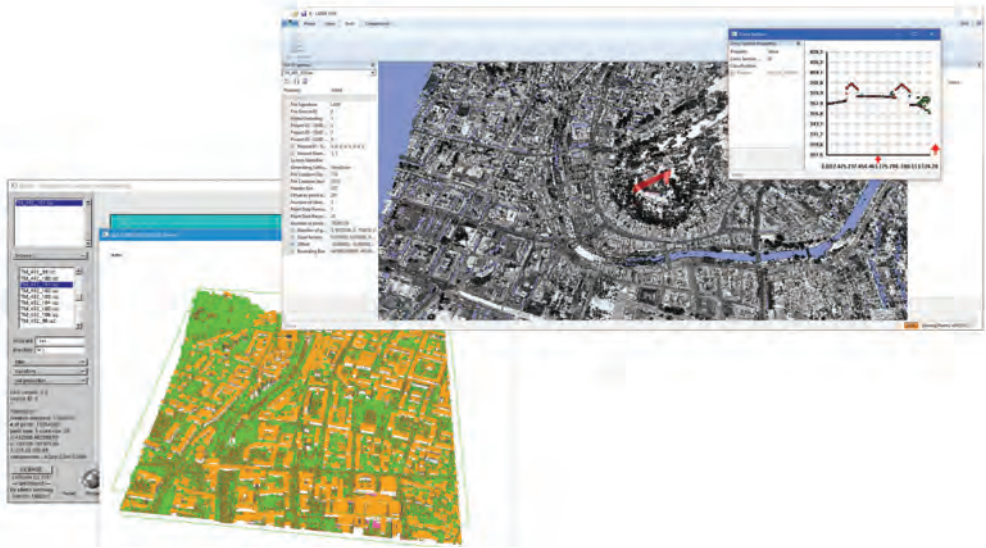
<sup>9</sup> <http://www.jmc2obj.net/>

<sup>10</sup> <http://chunky.l1bit.se/>

<sup>11</sup> <https://voxel.codeplex.com/>

<sup>12</sup> <https://rapidlasso.com/lastools/>

<sup>13</sup> <http://gemma.uni-mb.si/lidarlive/>

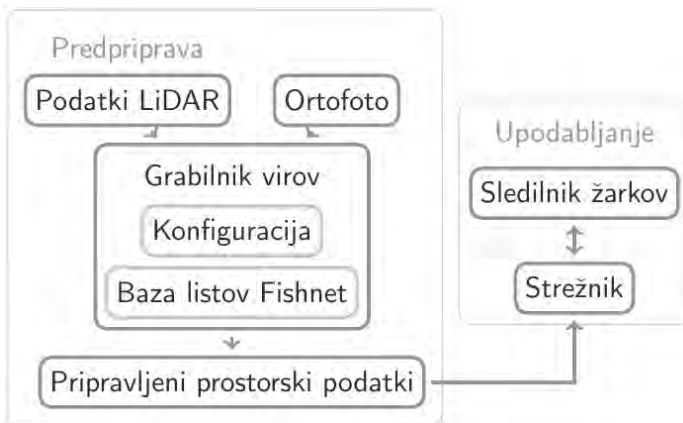


Slika 3: LAStools in LiDAR LIVE.

V nadaljevanju sta predstavljena arhitektura razvitega sistema in delovanje njegovih dveh glavnih delov, sledijo rezultati upodobitev, podane so ugotovitve in možnosti za drugačne rabe sistema.

## 2 ARHITEKTURA SISTEMA

Predstavljeni sistem deluje v dveh korakih: (1) priprava podatkov in (2) upodabljanje. Podrobnejša struktura je prikazana na sliki 4. Sistem uporablja dva osnovna vira podatkov, ki jih pretvori v obliko, primerno za upodabljanje: podatke LiDAR v obliki oblakov točk, ki so razdeljeni na kvadratna območja površine 1 km<sup>2</sup> in pokrivajo območje celotne Slovenije, ter ortofoto celotnega območja Slovenije. Obe zbirki podatkov sta georeferencirani glede na geodetski datum D96/TM.



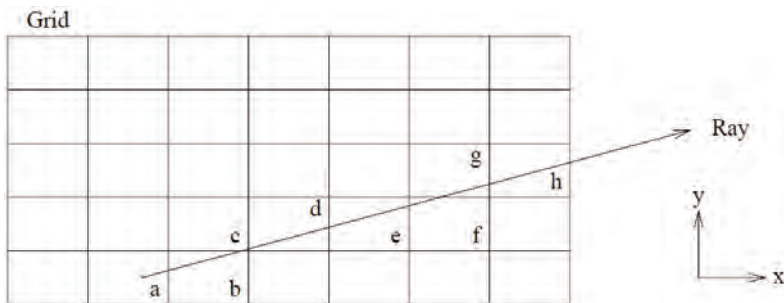
Slika 4: Arhitektura celotnega sistema.

V koraku priprave podatkov s spleta prenesemo podatke LiDAR in ortofoto za želeno območje s komponento Grabilnik virov za dostop do podatkov (glej sliko 4), ki jo sestavljajo programska skripta za platformo Node.js in pomožni programi. Program iz lokalne baze prebere definirane naslove izbranih območij podatkov in jih prenese s strežnikov ARSO. Po prenosu podatkov LiDAR jih za lažjo uporabo pretvori iz lastniškega formata zLAS v odprti format LAZ. Pri tem uporablja asinhrono vrste ter lahko hkrati prenaša in pretvarja več kosov podatkov.

Upodabljanje se izvaja v drugem koraku. V ta namen smo razvili program za interaktivno upodabljanje zahtevanega območja sveta s tehniko sledenja žarkom. Vhodni podatki za upodabljanje predstavljajo svet, zapisan v prilagojeni obliki zapisa vokslov, za kar smo razvili namenski spletni vokselski strežnik, ki pretvarja in pomni geodetske podatke v obliki kosov sveta ter jih na zahtevo pošilja upodobljevalniku. Pri izračunu končne upodobitve lahko sodeluje več upodobljevalnikov, tako da vsak izračuna le del končne slike. S tovrstnim paralelnim računanjem skrajšamo čas, potreben za izračun zelene upodobitve. Končni rezultat sistema je slika izbranega dela pokrajine.

### 3 UPODOBLJEVALNIK

V procesu upodabljanja uporabljamo tehniko sledenja žarkom, kot je predstavljena v Pharr in Humphreys (2004) in katere shema je prikazana na sliki 5. Natančneje: za upodobitev uporabimo senčne žarke, odbite in lomljene žarke, žarke za izračun okoliškega zastiranja in simulacijo razpršitve ozračja. Sledilnik žarkov je razvit posebej za izračun prehodov skozi polja vokslov. V prvem koraku sprehoda izhodišče žarka pretvorimo v koordinatni sistem mreže vokslov, nato pa z algoritmom (Amanatides in Woo, 1987) izračunamo pot prehodov žarka skozi voksele z upoštevanjem robnih primerov in brez večkratnih prehodov skozi iste voksele.

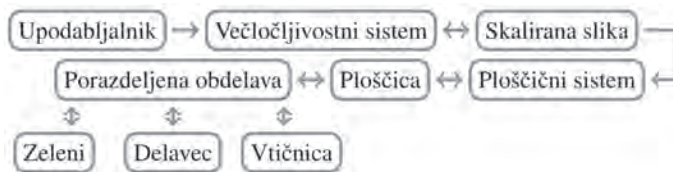


Slika 5: Algoritem korakov žarka skozi mrežo. (Vir: J. Amanatides, A. Woo: A Fast Voxel Traversal Algorithm for Ray Tracing)

V nadaljevanju predstavimo, kako v procesu upodabljanja končno sliko razbijemo na posamezne kose, ki jih obdelujejo proste procesne enote, ter kako posamezna procesna enota naloži in pomni svet vokslov za izračun sledenja žarkom.

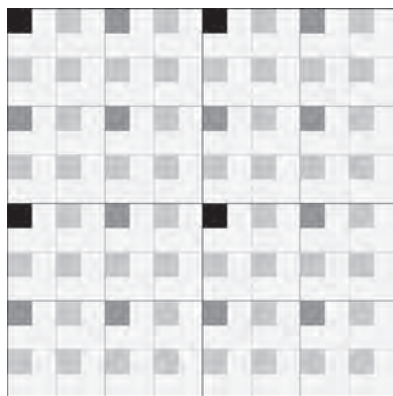
#### 3.1 Zgradba upodobljevalnika

Zgradba upodobljevalnika za izračun končne slike je predstavljena na sliki 6, kjer so prikazani posamezni koraki upodabljanja, ki jih podrobneje opišemo v nadaljevanju.



Slika 6: Pregled sestave upodobljevalnika.

Upodobljevalnik dopušča postopno upodabljanje, kar omogoča hiter interaktiven predogled izbranega prizora. Po končanem upodabljanju na najnižji ločljivosti upodobljevalnik izvaja upodobitve pri dvakrat višjih ločljivostih, dokler ne doseže zelene ciljne ločljivosti končne upodobitve. Ker je faktor med različnimi stopnjami ločljivosti ravno 2, lahko pri izračunu višje ločljivosti uporabimo že izračunano informacijo in dodatno izračunamo le še manjkajoče piksele slike. Primer slike pri različnih stopnjah ločljivosti je prikazan na sliki 7.



Slika 7: Štiri ločljivostne stopnje, prikazane z odtenki na končni sliki dimenzij 16 x 16. Višje ločljivosti so predstavljene s svetlejšimi odtenki sive.

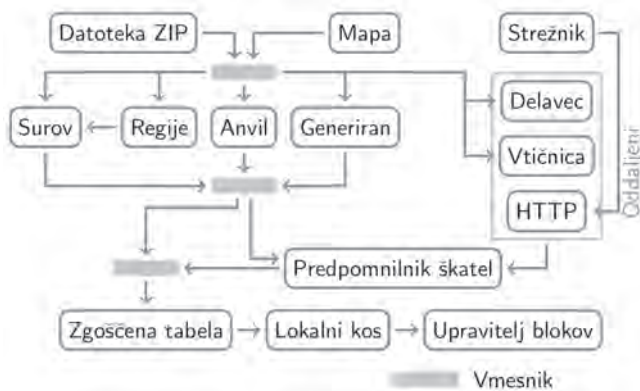
Sliko vsake ločljivosti razdelimo na posamezne ploščice (angl. *tiles*), ki vsebujejo množico pikselov določene velikosti (na primer  $64 \times 64$ ). S tem izboljšamo učinek lokalnega medpomnenja. Sistem podpira navpično in spiralno delitev na ploščice. Medtem ko je spiralna primernejša za interaktivne izrise, saj se v tem primeru središče slike izriše najprej, je vertikalna boljša v smislu učinkovitosti medpomnenja zaradi večje lokalnosti podatkov.

Posamezna ploščica se pošlje v porazdeljeno upodabljanje posameznemu prostemu procesorju glede na nastavitve sistema. V sistemu so implementirani procesorji različnih vrst: procesiranje v glavni niti, delovni procesor in vtični procesor. Delovni procesor lahko izračunava upodobitev v ločeni niti, vtični procesor pa lahko teče celo na oddaljeni lokaciji. Za izračun upodobitve je treba obema poslati ustrezne podatke.

### 3.2 Nalaganje sveta in medpomnjenje

Za upodabljanje sveta, sestavljenega iz vokslov, moramo najprej pridobiti zbirko podatkov, ki vsebuje želeno območje. Osnovna zgradba sistema za pridobivanje delov sveta je predstavljena na sliki 8.

V sistemu smo implementirali podporo za nalaganje svetov v treh različnih formatih, podprtih v igri Minecraft, ki so ogromen vir svetov, predstavljenih z vokslji. Podprte formate (Raw, Region in Anvil) sistem pretvori v lastno obliko zapisa kosov sveta, podobno tisti v igri Minecraft. Za upodabljanje omejenih območij brez navezave na zunanje podatkovne zbirke vsebuje sistem tudi generator okolja, ki teren ustvari z uporabo simpleksnega šuma. Za glavni vir podatkov v našem sistemu smo uporabili zbirko podatkov LiDAR in ortofota Slovenije. S tem namenom smo ustvarili spletni strežnik, ki vhodne podatke pretvori v predstavitev z označevanjem prostora in jih shrani v lastni obliki zapisa. Podrobnejše delovanje strežnika je predstavljeno v naslednjem poglavju.



Slika 8: Nalaganje in medpomnjenje sveta v upodobljevalniku.

Vsem vhodnim formatom je skupno izhodišče, da je svet razdeljen na posamezne navpične kose glede na vodoravno 2D-mrežo. To nam dopušča, da upodabljanje izvajamo s tako imenovano tehniko *out-of-core*, saj lahko posamezne kose sveta po potrebi nalagamo in odstranjujemo iz glavnega pomnilnika aplikacije, kar je potrebno predvsem pri upodabljanju velikih območij, ki jih v pomnilniku ne moremo hraniti naenkrat. Komponenta upravitelja blokov olajša nalaganje ustreznih kosov podatkov, njihovo učinkovito medpomnjenje z upoštevanjem geografske bližine in dostop do posameznih vokslov. V sistem smo dodali tudi pohitritve, ki upoštevajo domensko znanje, kot so izpuščanje kosov, kjer žarki ne bi zadeli nobenega voksla, in podobne optimizacije.

## 4 STREŽNIK VOKSLOV

Za pretvorbo velike količine točk iz zbirke podatkov LiDAR v obliko vokslov z robom en meter smo razvili namenski večnitni spletni strežnik. Ta na podlagi položaja in obsega območja, ki ga zahteva upodobljevalnik, zbere pripadajoče kose podatkov LiDAR ter izdela kos sveta v predstavitvi z vokslji in jo posreduje nazaj upodobljevalniku. V nadaljevanju podrobneje predstavimo posamezne korake pretvorbe podatkov iz izvornega oblaka točk v prečiščeno in prilagojeno predstavitev z vokslji. Prikaz delovanja posameznega koraka strežnika je prikazan na slikah 9 in 10.

#### 4.1 Medpomnjenje kosov podatkov

Iste podatke pogosto potrebuje več primerkov upodobljevalnika (primerki, ki upodablajo bližnja območja), zato jih je zaradi hitrejšega dostopa smiselno držati v medpomnilniku, kar se odraža v precej večji prepustnosti strežnika, ko posamezni primerki upodobljevalnika zahtevajo iste podatke.

#### 4.2 Nalaganje točk terena

Prvi korak pri pripravi posameznega kosa podatkov je nalaganje točk zahtevanega območja iz zbirke podatkov LiDAR. Za branje je uporabljena programska knjižnica LASlib, ki omogoča izbor točk interesnega območja v obliki pravokotnika. Knjižnica prav tako poskrbi za razširjanje podatkov in njihovo branje v pomnilnik. Zaradi pogoste potrebe po sočasnem nalaganju točk iz različnih izvornih datotek zbirke sistem podpira sočasno branje.

#### 4.3 Kvantizacija

Za doseganje učinka kockastega sveta, kakršen je v igri Minecraft, je treba svet, predstavljen z oblakom točk iz zbirke podatkov LiDAR, kjer so točke v prostor postavljene z natančnostjo nekaj centimetrov, še dodatno kvantizirati. V predstavljenem primeru je rob kocke dolg en meter. Poleg informacije o položaju točke v prostoru hranimo tudi informacijo o kategoriji, v katero posamezna kocka spada.

#### 4.4 Zapolnitev stavb

V izvorni zbirki podatkov LiDAR so zgradbe večinoma predstavljene zgolj s strehami, saj laserski žarki ne prodrejo skozi in posledično ne morejo zaznati strukture pod streho. Za realnejšo predstavitev zgradb želimo dodati tudi njihovo vertikalno strukturo. Ker nimamo natančnejših podatkov o stenah zgradb, predpostavimo, da so navpične. Za vsak element (voksel), ki je označen kot zgradba, se v navpični smeri navzdol, do najnižje višine v okolici, dodajo elementi, označeni kot zgradba. Kot rezultat se vse strehe zgradb uporabijo kot meje za njihove stene. Izkaže se, da je to dober približek za večino zgradb.

#### 4.5 Zapolnitev tal

Zaradi netočnosti pri zajemu podatkov LiDAR tla niso povsod povsem zvezna, kar se kaže kot luknje v tleh. Dejstvo je tudi, da točke predstavljajo zgolj površino tal, ne pa tudi sestave tal pod površino. Zaradi zapolnitve lukenj v tleh pregledamo, kateri stolpci imajo višino večjo od 0. Zanje predvidevamo, da imajo definirano površino tal. Za preostale stolpce, ki imajo višino 0, pa izračunamo položaj novega elementa, označenega kot tla, na podlagi linearne aproksimacije iz položajev najbližjih sosedov. Poleg tega dodamo elemente, označene kot tla, do najnižje višine v izbranem kosu sveta.

#### 4.6 Specializacija

Osnovni nabor pomenskih označb v zbirki podatkov LiDAR je omejen, saj zajema: nizko rastje, srednje rastje, visoko rastje, tla, zgradbe in morje. Za bolj realistično upodobitev natančneje določimo pomenske označbe najvišje ležečih enot – vokslor. Za to uporabimo ortofoto in na njem za lokacijo posameznega voksla poiščemo barvne vrednosti pikselov v letalskih posnetkih. Za ta namen obe zbirki podatkov pre-



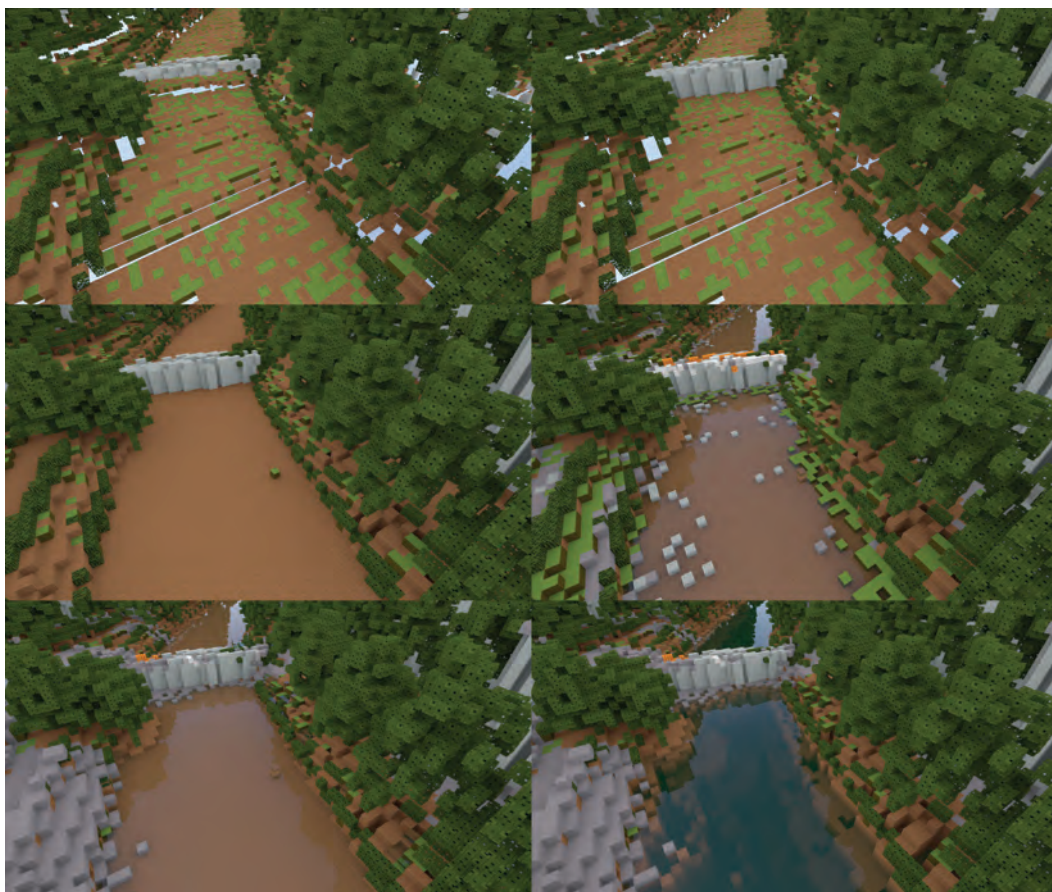
tvorimo v isti geodetski koordinatni sistem. Za izbrano množico barvnih vrednosti iz ortofota ustvarimo preslikovalno tabelo, ki posamezne barvne vrednosti poveže z ustrezno oznako. Vse barvne vrednosti povežemo z ročno določenimi vrednostmi na podlagi najkrajše evklidske razdalje v barvnem prostoru RGB. Za odpravo šuma, ki se pojavi pri specializaciji, nad podatki s posameznimi oznakami izvedemo filtriranje za odpravo lokalnih nepravilnosti.

#### 4.7 Izenačevanje vodne gladine in poglobljanje

Vodo označimo šele v koraku specializacije (glej poglavje 4.6), zato je lahko ponekod zelo šumna in nenaravna, še posebej ob brežinah. Predpostavimo, da ima voda v nekem kosu sveta enako nadmorsko višino, ki jo izračunamo z mediano višin vseh vodnih elementov v posameznem kosu. Vso vodo nekega kosa sveta poravnamo na to višino. Ker elemente vode označimo šele v koraku specializacije, določajo zgolj površje vode, kar predstavlja globino enega metra. Za bolj realističen videz vodo umetno poglobimo glede na oddaljenost od najbližje obale.



Slika 9: Upodobljen oddaljen prizor Ljubljane po vsakem koraku strežnika. Zgoraj levo – kvantizacija, zgoraj desno – zapolnitev zgradb, sredina levo – zapolnitev tal, sredina desno – specializacija, spodaj levo – filtriranje ter spodaj desno – izenačevanje in poglobljanje vodne površine.



Slika 10: Upodobljen bližnji prizor Ljubljance po vsakem koraku strežnika. Zgoraj levo – kvantizacija, zgoraj desno – zapolnitev zgradb, sredina levo – zapolnitev tal, sredina desno – specializacija, spodaj levo – filtriranje ter spodaj desno – izenačevanje in poglobljanje vodne površine.

## 5 REZULTATI

Predstavljeni sistem omogoča upodobitev manjših in tudi večjih območij Slovenije. Izbira zelenega območja upodabljanja je mogoča prek preprostega uporabniškega vmesnika, ki omogoča tako uporabo vnaprej določenih predlog za upodabljanje kot tudi ročno nastavitve vseh parametrov upodabljanja. Na slikah od 11 do 16 so predstavljeni primeri upodobitev posameznih izbranih območij. Trajanje upodabljanja različnih območij je odvisno od velikosti izbranega območja in nastavitve parametrov upodabljanja. Ob posameznih slikah so opisane tudi težave, ki se pojavljajo v konkretnih primerih.



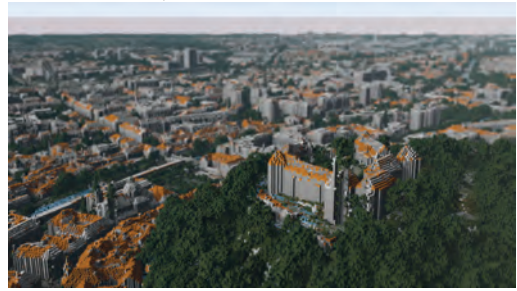
Slika 11: Ljubljana iz zraka. Upodobitev se približuje fotorealizmu. Na tej oddaljenosti večine težav, ki bi nastale ob upodobljanju, ne zaznamo.



Slika 12: Piran. Tudi v tem primeru se pojavljajo težave na vodni površini, kjer ponekod na morju točke močno odstopajo od dejanske gladine. Prav tako je pri stolpu piranske cerkve klasifikacija točk v izvornih podatkih LiDAR napačna.



Slika 13: Blejski otok. Na upodobitvi je lepo vidna napačna klasifikacija v izvornih podatkih LiDAR, saj je zvonik označen kot visoko rastje, in je tako tudi izrisan. Pri Blejskem jezeru se pojavljajo tudi težave na vodni površini, kjer v izvornih podatkih ni dovolj točk za posamezna območja. Takšne težave smo delno rešili z zapolnjevanjem lukenj na podlagi ortofota in digitalnega modela višin površja.



Slika 14: Pogled na Ljubljano in Ljubljanski grad. V upodobitvi so dobro vidne težave z barvanjem strešnih elementov. Prav tako v koraku specializacije napačno označimo okolico gradu kot vodno površino, ker je barva travnate podlage v senci preveč podobna izbranim barvam, ki jih uporabimo za klasifikacijo vodnih površin. Prikazan je tudi učinek globinske ostrine, saj je v fokusu grad, okolica pa je z oddaljenostjo vse bolj zamegljena.



Slika 15: Ljubljana. Na tej upodobitvi vidimo, kakšne težave nastajajo pri mostovih. Mostovi so klasificirani kot zgradbe in jih v koraku zapolnitve zgradb poglobimo, kar seveda ne ustreza dejanskemu stanju. Prav tako je v upodobitvi razvidno, da lahko ob robovih struge višina vodne gladine precej odstopa od pričakovane.



Slika 16: Šmarna gora. Na sliki je prikazano, kako se na upodobitvi odraža dviganje terena. Lepo so vidna območja z enako nadmorsko višino, če niso prekrita z gozdno površino. Gozdna površina lepo zmešča stopničasto vzpenjanje, k čemur zelo veliko pripomorejo predvsem sence. Lepo so razvidni tudi koridorji daljnovidov.

## 6 ZAKLJUČEK

Upodobitve, ustvarjene s predstavljenim sistemom, so nova možnost za prikaz slovenske krajine v obliki, ki je v zadnjem času zelo priljubljena – z uporabo vokslov. S tem dobimo drugačen pogled na pokrajino. S sistemom lahko upodabljamo tudi večja območja, kar je kljub velikosti osnovnega elementa en meter prepričljivo realistična upodobitev. Podatke, kot jih za upodabljanje pripravi sistem, lahko uporabimo tudi v drugačnem scenariju, na primer kot vhod za ustvarjanje sveta v prvoosebni pustolovski igri, postavljeni v slovenski prostor. Več o podrobnostih sistema lahko najdete v delih Lunar et al. (2016), Lunar in Marolt (2016).

### Literatura in viri:

- Geodetski inštitut Slovenije (2015). Izvedba laserskega skeniranja Slovenije. Blok 35 – tehnično poročilo o izdelavi izdelkov. Geodetski inštitut Slovenije. [http://gis.arso.gov.si/related/lidar\\_porocila/b\\_35\\_izdelava\\_izdelkov.pdf](http://gis.arso.gov.si/related/lidar_porocila/b_35_izdelava_izdelkov.pdf), pridobljeno 21. 8. 2015.
- Mongus, D., Čekada, M. T., Žalik, B. (2013). Analiza samodejne metode za generiranje digitalnih modelov reliefa iz podatkov lidar na območju Slovenije. Geodetski vestnik, 57 (2), 245.
- Pečnik, S., Žalik, B., Mongus, D. (2009). Interaktivno orodje za obdelavo podatkov LiDAR. Diplomsko naloga univerzitetnega študijskega programa. Maribor: Univerza v Mariboru Fakulteta za elektrotehniko, računalništvo in informatiko.
- Pharr, M., Humphreys, G. (2004). Physically based rendering: From theory to implementation. Morgan Kaufmann.
- Amanatides, J., Woo, A. (1987). A fast voxel traversal algorithm for ray tracing. Eurographics, 87 (3), 3–10.
- Lunar, M., Bohak, C., Marolt, M. (2016). Distributed ray tracing for rendering voxelized LIDAR geospatial data. V: Zbornik petindvajsete mednarodne Elektrotehniške in računalniške konference ERK 2016, Portorož, str. 55–58.
- Lunar, M., Marolt, M. (2016). Porazdeljeno sledenje žarkov za upodabljanje lasersko zajetih prostorskih podatkov. Diplomsko naloga. Ljubljana: Univerza v Ljubljani, Fakulteta za računalništvo in informatiko.



Lunar, M., Bohak, C., Marolt, M. (2016). Porazdeljeno upodabljanje vokseliziranih podatkov LiDAR. Geodetski vestnik, 60 (4): 615–626. DOI: 10.15292/geodetski-vestnik.2016.04.615-626

**Miha Lunar, univ. dipl. inž. rač. in inf.**  
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko  
Večna pot 113, SI-1000 Ljubljana  
e-naslov: mlunar@gmail.com

**Dr. Ciril Bohak, univ. dipl. inž. rač. in inf.**  
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko  
Večna pot 113, SI-1000 Ljubljana  
e-naslov: ciril.bohak@fri.uni-lj.si

**Doc. dr. Matija Marolt, univ. dipl. inž. rač. in inf.**  
Univerza v Ljubljani, Fakulteta za računalništvo in informatiko  
Večna pot 113, SI-1000 Ljubljana  
e-naslov: matija.marolt@fri.uni-lj.si