

Univerza
v Ljubljani
Fakulteta
*za gradbeništvo
in geodezijo*

*Janova 2
1000 Ljubljana, Slovenija
telefon (01) 47 68 500
faks (01) 42 50 681
fgg@fgg.uni-lj.si*



Univerzitetni program Geodezija,
smer Geodezija

Kandidat:

Gregor Stavbar

Pregled odprtokodnih programov za obdelavo lidarskih podatkov

Diplomska naloga št.: 865

Mentor:
doc. dr. Mojca Kosmatin Fras

Ljubljana, 29. 6. 2011

STRAN ZA POPRAVKE, ERRATA

Stran z napako

Vrstica z napako

Namesto

Naj bo

IZJAVE

Podpisani Gregor Stavbar izjavljam, da sem avtor diplomskega dela z naslovom »Pregled odprtokodnih programov za obdelavo lidarskih podatkov«.

Izjavljam, da je elektronska različica v vsem enaka tiskani različici.

Izjavljam, da dovoljujem objavo elektronske različice v repozitoriju UL FGG.

Ljubljana, 15.6.2011

Gregor Stavbar

BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

- UDK:** 659.2:004:528.02(043.2)
- Avtor:** Gregor Stavbar
- Mentor:** doc. dr. Mojca Kosmatin Fras
- Naslov:** Pregled odprtokodnih programov za obdelavo lidarskih podatkov
- Obseg in oprema:** 79 str., 11 pregl., 19 sl., 1 en.
- Ključne besede:** prosto in odprtokodno programje, aerolasersko skeniranje, LAS, ASCII, digitalni model reliefa

Izveleček:

Glavna tema diplomske naloge je pregled in izbira proste in odprtokodne programske opreme za obdelavo lidarskih podatkov. Naloga v prvem delu zajema kratek opis osnovnih principov tehnologije aerolaserskega skeniranja ter opisuje in primerja standardna podatkovna formata za shranjevanje lidarskih podatkov (ASCII in LAS). V drugem delu se naloga posveča pregledu razvoja proste in odprte kode z opisi najpogosteje uporabljenih programskih licenc, pod katerimi je izdana prosta in odprtokodna programska oprema. Sledeča poglavja zajemajo pregled in analizo omenjene programske opreme. Izbira programov je opravljena na osnovi kriterijev, ki določajo primernost programske opreme iz vidika pedagoške in raziskovalne uporabe. Kriteriji so definirani ločeno za vsak namen uporabe. S pomočjo ocenitve programov je za vsak vidik uporabe priporočen posamezen program. Diplomska naloga se v prvi vrsti posveča pregledu programov, primernih za uporabo v pedagoški namen. S testnimi podatki iz obstoječega snemanja v Sloveniji (Koroška bela) je z uporabo izbranih programov SAGA GIS in FUSION/LDV predstavljen postopek obdelave lidarskih podatkov in izdelave digitalnega modela reliefa. Za oba izbrana programa so izdelana kratka navodila za uporabo programov pri izdelavi digitalnega modela reliefa.

BIBLIOGRAPHIC – DOCUMENTALISTIC INFORMATION

UDC: 659.2:004:528.02(043.2)

Author: Gregor Stavbar

Supervisor: Assist. Prof. Mojca Kosmatin Fras, Ph.D.

Title: Review of Open Source Software for Lidar Data Processing

Notes: 79 p., 11 tab., 19 fig., 1 eq.

Key words: free and open source software, airborne laser scanning, LAS, ASCII, digital terrain model

Abstract:

The thesis deals with free and open source software for lidar data processing. The first part includes descriptions of basic principles of airborne laser scanning technology, and describes and compares standard data formats for lidar data storage (ASCII and LAS). In the second part, development of free and open source software is investigated, describing most commonly used software licenses, under which free and open source software is issued. In the following chapters the aforementioned software is examined and analysed. Software is chosen based on criteria determining the adequacy of the software for pedagogical and research usages. The criteria are defined separately for each intended use. Through software assessment individual software is recommended for particular uses. The thesis primarily focuses on software review for pedagogical aspects of use. Test data derived from a recent scanning in Slovenia (Koroška bela) is used to introduce the procedure of lidar data processing, and elaboration of digital terrain model with software SAGA GIS and FUSION/LDV. Operating instructions for both programmes for digital terrain model elaboration (SAGA GIS and FUSION/LDV) are added.

ZAHVALA

Hvala doc. dr. Mojci Kosmatin Fras za pomoč in usmerjanje pri diplomski nalogi. Hvala za mentorstvo in podporo tekom študija.

Hvala mojim staršem, Vanji in Gabrijelu, da sta mi omogočila študij, me razumela in vzpodbujala. Hvala sestrama Sabini in Tini ter devru Blažu za brezpogojno podporo.

Hvala vsem prijateljem in sošolcem za zanimivo in razburljivo desetletje. Je hitro minilo.

Hvala tudi Lenki in tovarišu Petru za pomoč pri študiju, ko sem jo najbolj potreboval.

KAZALO VSEBINE

1	UVOD	1
2	AEROLASERSKO SKENIRANJE	3
2.1	Komponente merskega sistema ALS	3
2.2	Osnovni princip delovanja tehnologije LiDAR.....	5
2.3	Zajem podatkov	6
2.4	Obdelava podatkov.....	7
3	ORGANIZACIJA PODATKOV IN PODATKOVNI FORMAT	9
3.1	Organizacija podatkov v oblaku točk.....	9
3.2	Način zapisa podatkov v datoteko.....	10
3.2.1	Tekstovni ASCII podatkovni format.....	10
3.2.2	Binarni LAS podatkovni format.....	11
3.2.2.1	Razvoj LAS formata.....	11
3.2.2.2	Značilnosti LAS formata v primerjavi z ASCII formatom.....	12
3.2.2.3	Struktura LAS formata	13
4	PROSTA IN ODPRTOKODNA PROGRAMSKA OPREMA.....	15
4.1	Prosto programje oz. free software.....	15
4.2	Odprtokodno programje oz. open source software	16
4.3	Licence odprtokodnega in prostega programja.....	18
4.3.1	GNU GPL licenca	19
4.3.2	GNU LGPL licenca.....	21
4.3.3	BSD licenca.....	21
4.3.4	MIT licenca	22
4.3.5	MPL licenca	23
4.3.6	Apache License	24
5	IZBIRA PROGRAMOV IN OCENA PRIMERNOSTI ZA UPORABO	25
5.1	Iskanje ustreznih programov in izbira kandidatov za ocenitev	26
5.2	Kriteriji za ocenitev primernosti.....	29
5.3	Opis kriterijev in ocena programov.....	30

5.3.1	Zrelost projekta.....	30
5.3.2	Dokumentacija.....	31
5.3.2.1	Uporabniška dokumentacija.....	31
5.3.2.2	Razvijalska dokumentacija	32
5.3.3	Skupnost in podpora.....	32
5.3.3.1	Skupnost.....	32
5.3.3.2	Podpora	33
5.3.4	Prepoznavnost	34
5.3.5	Integracija.....	35
5.3.5.1	Modularnost.....	35
5.3.5.2	Standardi	36
5.3.5.3	Medopravilnost	36
5.3.5.4	Programske in strojne zahteve	36
5.3.6	Uporabniški nivo	37
5.3.7	Funkcionalnost	37
5.4	Vrednotenje kriterijev	38
5.5	Rezultati ocenitve programov	40
5.6	Izbira primernega kandidata.....	41
5.7	Opisi programov	42
5.7.1	Opis izbranih programov	42
5.7.1.1	SAGA GIS	42
5.7.1.2	FUSION/LDV	43
5.7.2	Opis ocenjevanih programov.....	45
5.7.2.1	GRASS GIS	45
5.7.2.2	gvSIG	46
5.7.2.3	ILWIS	48
5.7.2.4	MapWindow	48
5.7.2.5	Quantum GIS	50
5.7.2.6	QuickGrid	50
5.7.2.7	Whitebox GAT	51
5.7.3	Opis zanimivih, s selektivnimi kriteriji izločenih, programov in knjižnic	52
5.7.3.1	LAStools	52
5.7.3.2	libLAS.....	54
5.7.3.3	MCC-LIDAR	55
5.7.3.4	ALDPAT.....	55
5.7.3.5	GEON Points2Grid.....	56
5.7.3.6	FullAnalyze.....	57

6	NAVODILA ZA IZDELAVO DMR Z IZBRANIMA PROGRAMOMA.....	59
6.1	Izdelava DMR s programom FUSION/LDV.....	59
6.1.1	Prenos in namestitvev programa	59
6.1.2	Zagon programa in priprava projekta.....	59
6.1.3	Predogled podatkov.....	60
6.1.4	Konvertiranje podatkov.....	60
6.1.5	Izdelava referenčne podobe.....	61
6.1.6	Uvoz podatkov	63
6.1.7	Izdelava DMR	64
6.1.7.1	Uporaba orodja Groundfilter	64
6.1.7.2	Uporaba orodja GridSurfaceCreate	66
6.1.7.3	Izvoz podatkov	68
6.2	Izdelava DMR s programom SAGA	69
6.2.1	Prenos in namestitvev programa	69
6.2.2	Zagon programa in izdelava DMR.....	69
6.2.3	Filtriranje podatkov	70
6.2.4	Izdelava DMR	72
7	ZAKLJUČEK	73
	LITERATURA IN VIRI	75

KAZALO PREGLEDNIC

Preglednica 2-1: Klasifikacija zajetih lidarskih točk v razrede (ASPRS, 2009).....	8
Preglednica 3-1: Primer atributov lidarskih točk (Shan in Toth, 2009: 300).....	12
Preglednica 3-2: Primerjava kodiranja datoteke z 1 milijonom točk (Shan in Toth, 2009: 299)	13
Preglednica 3-3: Blokovna razdelitev formata LAS	14
Preglednica 5-1: Širok seznam programov, ki omogočajo manipulacijo lidarskih podatkov.....	27
Preglednica 5-2: Seznam kriterijev	30
Preglednica 5-3: Število možnih točk kriterijev ocenitve	38
Preglednica 5-4: Uteži kriterijev	39
Preglednica 5-5: Ocena FOSS programov za obdelavo lidarskih podatkov - pedagoški namen	40
Preglednica 5-6: Ocena FOSS programov za obdelavo lidarskih podatkov – raziskovalni namen.	40
Preglednica 5-7: Seznam nekaterih LAStools orodij	53

KAZALO SLIK

Slika 2-1: Sistem LiDAR (Zalokar, 2011: 5).....	4
Slika 2-2: Zajem podatkov v misiji ALS (Bitenc, 2006: 5).....	5
Slika 3-1: Primerjava pravilne mreže in oblaka točk (povzeto po Shan in Toth, 2009: 298).....	9
Slika 5-1: Prikaz uporabe programa SAGA.	43
Slika 5-2: 3D prikaz ploskve reliefa in oblak lidarskih točk	44
Slika 5-3: 3D pogled DMR-ja, izdelanega iz *.shp datoteke podatkov.....	47
Slika 5-4: Primer izdelave DMR iz *.shp datoteke podatkov	49
Slika 5-5: Primer uporabe programa QuikGrid	51
Slika 5-6: Primer uporabe programa Whitebox GAT.....	52
Slika 5-7: Prikaz oblaka terenskih točk z LASview orodjem.....	53
Slika 6-1: Primer izdelane referenčne podobe s pomočjo vrednosti intenzitete.....	62
Slika 6-2: Izbira simbola.....	63
Slika 6-3: Razlaga delov sintakse.....	65
Slika 6-4: Oblak točk po filtriranju s filtrom GroundFilter	66
Slika 6-5: Plastičen prikaz reliefa po uporabi orodja GridSurfaceCreate.....	68
Slika 6-6: Prikaz DMR-ja, uvoženega v SAGA-o iz programa FUSION/LDV	69
Slika 6-7: Oblak lidarskih točk.....	70
Slika 6-8: Sloj terenskih točk po filtriranju z modulom Subset Extractor.....	71
Slika 6-9: DMR izdelan s pomočjo metode interpolacije inverzne razdalje	72

SEZNAM PRILOG

Priloga A: Vprašanja za izpolnitev kriterijev in njihovo točkovno ovrednotenje

Priloga A.1: Vprašanja in rezultati vrednotenja kriterija Zrelost projekta

Priloga A.2: Vprašanja in rezultati vrednotenja kriterija Dokumentacija

Priloga A.3: Vprašanja in rezultati vrednotenja kriterija Skupnost in podpora

Priloga A.4: Vprašanja in rezultati vrednotenja kriterija Prepoznavnost

Priloga A.5: Vprašanja in rezultati vrednotenja kriterija Integracija

Priloga A.6: Vprašanja in rezultati vrednotenja kriterija Funkcionalnost

Priloga A.7: Vprašanja in rezultati vrednotenja kriterija Uporabniški nivo

OKRAJŠAVE

ALS	Aerolasersko skeniranje
ASCII	American standard code for information exchange
ASPRS	American society for photogrammetry and remote sensing
BSD	Berkeley software distribution license
CSV	Comma separated value
DGPS	Diferencialni GPS
DMP	Digitalni model površja
DMR	Digitalni model reliefa
DMV	Digitalni model višin
EMV	Elektromagnetno valovanje
FAQ	Frequently asked questions
FGG	Fakulteta za gradbeništvo in geodezijo
FOSS	Free and open source software
FSF	Free software foundation
FSM	Free software movement
GIS	Geografski informacijski sistem
GNU	Gnu's not Unix
GPL	General public license
GPS	Global positioning system
GUI	Graphic user interface
IBM	International business machines corporation
IMU	Inercialna merska enota
INS	Inercialni navigacijski sistem
LAS	Podatkovni format za shranjevanje lidarskih podatkov
LGPL	Lesser/library general public license
LiDAR	Light detection and ranging

LS	Laserski skener
MIT	Massachusetts institute of technology
MPL	Mozilla public license
OS	Operacijski sistem
OSGeo	Open source geospatial foundation
OSI	Open source initiative
POS	Sistem za določanje položaja in orientacije
TEC	Topographic engineering center
TIN	Neppravilna trikotniška mreža
VCT	Version control tool
XML	Extensible markup language

1 UVOD

Prosto in odprtokodno programje ali programska oprema (angl. free and open source software) je v zadnjih letih napravila velik korak v razvoju.

To področje računalništva in informatike ni novo, saj njegovi začetki segajo že v 60. leta 20. stoletja. Toda še preden je dobro zajelo zrak, se je znašlo v položaju, v katerem so vladala večja podjetja in korporacije, ki so hlepela le po čim večjem zaslužku. V zadnjih letih, ko svet z vzponom interneta in hitrim razvojem računalništva ter tehnologije nasploh postaja vsak dan manjši, je to programje začelo uživati vse več zaslužene pozornosti. Danes ne velja več le za trenutno tržno nišo, ampak za vedno bolj pomembno in razvito področje programske opreme, saj se zanj zanima vedno več uporabnikov kot tudi razvijalcev programske opreme, širom po svetu. Med najbolj znano prosto in odprtokodno programje (v nadaljevanju FOSS) spadajo programski paketi, kot so urejevalnik besedila OpenOffice.org, internetni brkljalnik Mozilla Firefox, operacijski sistem Linux, grafični urejevalnik Inkscape, znanstveni program za statistično računanje R Project, ipd.

Tako kot FOSS programje, je na področju fotogrametrije in daljinskega zaznavanja v vse večjem vzponu tudi izdelava topografskih podatkov z aerolaserskim skeniranjem, ki temelji na tehnologiji LiDAR (angl. Light Detection and Ranging).

S tehnologijo LiDAR (v nadaljevanju lidar) lahko iz zraka zajamemo ogromne količine podatkov o zemeljskem površju in objektih na njem. Ker tehnologija deluje na osnovi oddajanja elektromagnetnega valovanja (EMV) iz lastnega vira, to je aktivnega senzorja, zajem podatkov ni odvisen od vremenskih in časovnih omejitev. Zaradi ogromne količine zajetih podatkov in s tem povezane njihove obdelave kot npr. izdelava digitalnega modela reliefa (DMR), je potrebna uporaba zmogljivih računalnikov kot tudi posebne programske opreme. Danes je na voljo kar nekaj lastniške (angl. proprietary) kot tudi proste in odprtokodne programske opreme, ki je sposobna avtomatizirane obdelave omenjenih količin podatkov. Lidarska tehnologija ima kot rečeno kar nekaj pozitivnih lastnosti, vendar pa je potrebno izpostaviti, da velika količina podatkov še ne zagotavlja dobrih rezultatov. Z večjo količino podatkov se pojavijo tudi težave, predvsem pri njihovi obdelavi.

Povezava lidarske tehnologije in FOSS programske opreme, ki omogoča obdelavo lidarskih podatkov, je zanimiva za različne institucije, organizacije in podjetja kakor tudi za posameznika. FOSS programska oprema, kot pove ime samo, omogoča prosto porazdeljevanje in spreminjanje izvorne programske kode.

V diplomski nalogi sem se posvetil iskanju FOSS programske opreme, ki je sposobna obdelovati lidarske podatke. Zaradi količine ter geometrijskih in fizikalnih lastnosti z lidarjem zajetih podatkov so v programski opremi potrebni zmogljivi algoritmi, ki omogočajo njihovo obdelavo in izdelavo želenih prikazov površja kot je DMR.

Cilj moje diplomske naloge je bil poiskati FOSS programe, ki bi omogočali obdelavo lidarskih podatkov in hkrati bili primerni za uporabo in potrebe Fakultete za gradbeništvo in geodezijo Univerze v Ljubljani. S tem namenom sem se posvetil iskanju programov, ki bi omogočali študentu vpogled v obdelavo in spoznavanje uporabnosti surovih lidarskih podatkov. Želel sem sestaviti seznam programov, ki na kakršen koli način omogočajo obdelavo lidarskih podatkov ali le del postopka obdelave.

Za samo iskanje in predvsem izbiro teh specifičnih programov sem določil kriterije. S temi kriteriji, ki so mi bili v pomoč pri vrednotenju in ocenitvi primernosti programov za uporabo v pedagoške namene, sem želel doseči določeno stopnjo objektivnosti pri njihovi izbiri. Za sestavo kriterijev je potrebno dobro preučiti FOSS področje in ugotoviti določene zahteve, ki specifičen program izpostavijo iz preostale množice.

Prav tako sem vzporedno ocenjeval isto skupino programov za uporabnost v raziskovalne namene. Kriterije v ta namen sem dodatno prilagodil, saj so potrebe, želje in znanje raziskovalcev različne od potreb in želja študenta, ki je šele na začetku svoje akademske poti. Ko sem programe ocenil, sem na podlagi njihovih ocen in subjektivnega mnenja izbral dva kandidata, ki sta ustrezala potrebam pedagoškega in raziskovalnega namena.

Na koncu naloge sem predstavil njuno uporabnost na konkretnem primeru, to je izdelavi DMR-ja iz testnih podatkov oblaka lidarskih točk, ki so bili zajeti v Sloveniji. Hkrati sem za vsak program pripravil kratka navodila za izdelavo DMR-ja.

2 AEROLASERSKO SKENIRANJE

Aerolasersko skeniranje¹ (ALS) oziroma LiDAR (Light Detection And Ranging, v nadaljevanju lidar) je tehnika daljinskega zaznavanja, s katero iz zraka s pomočjo laserske svetlobe pridobivamo podatke o zemeljskem površju, objektih na njem in njihovih lastnostih. Začetek razvoja te tehnologije v poznih šestdesetih letih dvajsetega stoletja časovno sovпада z uporabo laserja za merjenje razdalj. V devetdesetih letih prejšnjega stoletja se je ALS začel hitro razvijati in širiti krog uporabe predvsem zaradi fizikalnih lastnosti laserske svetlobe in razvoja podpornih tehnologij kot sta GPS² in inercialni navigacijski sistem (INS), ki sta poglobitvena sestavna dela vsakega lidarskega sistema.

V nadaljevanju poglavja so na kratko, za potrebe razumevanja vsebine te diplomske naloge, predstavljeni princip delovanja sistema ALS, glavni sestavni deli tega sistema, zajem podatkov in njihove karakteristike s podrobnim opisom formata LAS.

2.1 Komponente merskega sistema ALS

Merski sistem ALS (Slika 2-1) sestavljajo tri glavne samostojne enote:

- laserski skener (LS), ki ga sestavljata laserski razdaljemer (LiDAR) in skener oz. naprava za odklon laserskih žarkov
- inercialni navigacijski sistem (INS)
- sistem GPS

Laserski razdaljemer je aktivni senzor³, ki omogoča merjenje razdalje od senzorja do tarče⁴ s pomočjo odboja laserskega žarka. Ker je hitrost svetlobe poznana, je za določitev razdalje do tarče potrebno izmeriti le čas potovanja laserskega žarka, ki ga predstavlja časovna razlika med trenutkom oddaje žarka in trenutkom prejema dela istega žarka, ki se je odbil od tarče na površju. Skener je elektrooptični sistem, ki usmerja laserski žarek proti zemeljskemu površju prečno na smer leta nosilca⁵ in meri odklon laserskega žarka (angl. scan angle).

Inercialni navigacijski sistem (INS) je merski sistem, ki definira smer laserskega žarka z določitvijo položaja, hitrosti, orientacije in kotne hitrosti nosilca laserskega sistema. INS vsebuje inercialno

¹ angl. Airborne Laser Scanning.

² Global Positioning System

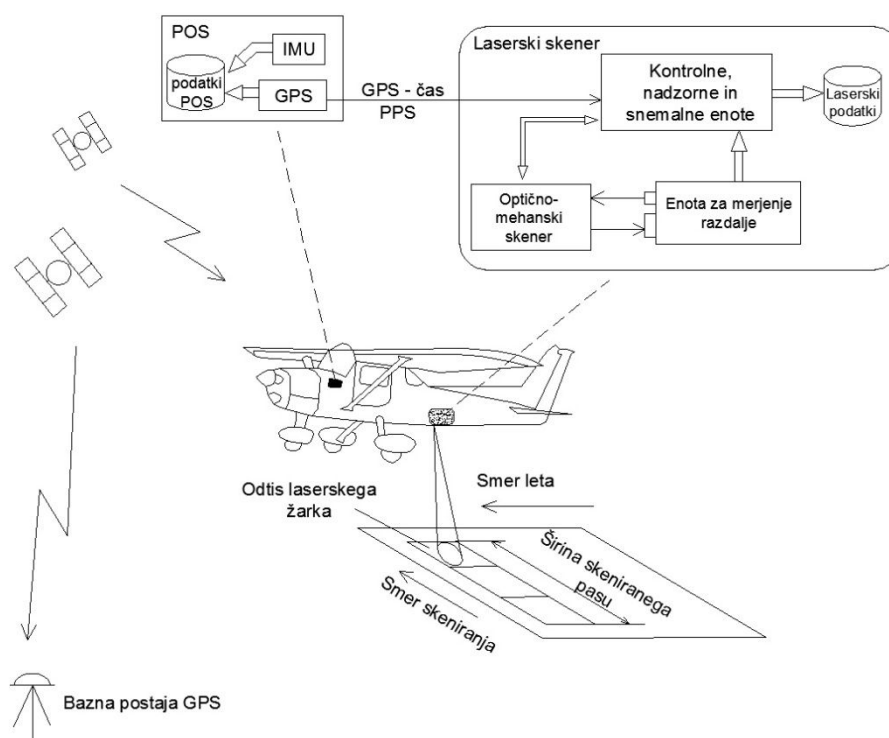
³ Aktivni senzor ima lasten vir elektromagnetnega valovanja (EMV), ki ga oddaja proti zemeljskemu površju in registrira njegov odboj, medtem ko pasivni senzor registrira v naravi prisotno elektromagnetno valovanje (npr. sončna svetloba). Pri aktivnem senzorju nismo odvisno od časa zajemanja in vremenskih razmer, ravno zaradi dejstva, da sam generira EMV, za razliko od pasivnega senzorja, kjer smo odvisno od obstoja EMV, ki ga zajemamo (npr. Sončna svetloba – meritve lahko izvajamo le podnevi, motijo nas oblaki ipd.).

⁴ Objekt na zemeljskem površju.

⁵ Letalo ali helikopter.

mersko enoto⁶ (IMU), ki jo sestavljajo trije merilniki pospeška in trije žiroskopi ter navigacijski računalnik, ki obdeluje podatke pridobljene z IMU. GPS določa položaj laserskega skenerja in beleži pot nosilca. Pri ALS se za izračun položaja LS v globalnem koordinatnem sistemu uporablja kinematična GPS metoda. Za doseg večje natančnosti meritev se izvajajo diferencialne GPS meritve (DGPS), kjer se uporabljajo permanentne referenčne postaje, ki so vključene v državno GPS mrežo ter ležijo znotraj območja snemanja.

INS in GPS sistema skupaj tvorita sistem za določanje položaja in orientacije (POS). POS je nujno potreben za določitev prostorskega položaja točke, saj je potrebno v trenutku meritve posamezne točke poznati položaj in orientacijo LS v izbranem koordinatnem sistemu (npr. WGS-84). Povezanost in sinhronost INS in GPS sistemov omogoča neposredno določitev elementov zunanje orientacije, kar posledično pomeni, da ni potrebe po določitvi terenskih oslonilnih točk za orientacijo lidarskih pasov.



Slika 2-1: Sistem LiDAR (Zalokar, 2011: 5)

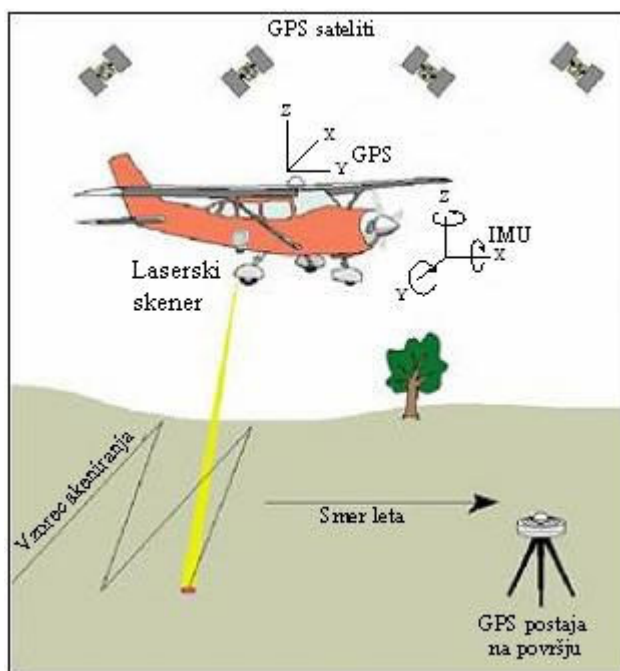
Sistemu ALS je največkrat dodana še digitalna kamera, ki omogoča simultano zajemanje vidne svetlobe. Rezultat so digitalne fotografije, ki jih lahko pretvorimo v ortofoto. Preostale komponente merskega sistema tudi drugi senzori po izbiri (digitalni CCD fotoaparati, fotogrametrični zračni fotoaparati, itd), računalnik, operacijski sistem in programi za kontroliranje zajema podatkov med letom, medij za shranjevanje laserskih meritev, podatkov skenerja, GPS-ja, INS-ja in alternativnih slikovnih podatkov, programi za planiranje leta in za postprocesiranje podatkov, GPS za navigacijo

⁶ Inertial Measurement Unit

leta nosilca sistema, komponente za montažo sistema ter neobvezno še merilci temperature in vlažnosti (Baltsavias, E., 1999, cit. po Bitenc, M., 2006).

2.2 Osnovni princip delovanja tehnologije LiDAR

Lidar je tehnologija daljinskega zaznavanja, ki s pomočjo aktivnega sensorja t.j. laserskega skenerja, vgrajenega v dno nosilca, oddaja signale proti zemeljskemu površju in objektom na njem ter prejme od njih odbit in razpršen signal (Slika 2-2).



Slika 2-2: Zajem podatkov v misiji ALS (Bitenc, 2006: 5)

Laserski razdaljemer odda svetlobo proti napravi za odklon žarkov, od katere se žarek odbije in usmeri proti zemeljskemu površju, na katerem se navidezno zariše vzorec skeniranja, prečno na smer leta. Signal (valovanje) se na tleh odbije in le del vrne nazaj v smeri sensorja. V LS vgrajen detektor vrnjen odboj zazna in registrira čas potovanja žarka, kar pomeni, da zabeleži čas oddaje signala aktivnega sensorja in čas vrnitve odboja. S pomočjo časa potovanja signala lahko posredno izmerimo dvojno razdaljo (poševno) od sensorja do tarče, od katere se je žarek odbil. V skenerju se hkrati zabeleži tudi kot odklona oddanega signala.

Za določitev prostorskih koordinat zajetih točk v referenčnem koordinatnem sistemu je potrebno poznati položaj in orientacijo sensorja LS v trenutku oddaje posameznega laserskega signala. Položaj je določen s simultanimi meritvami DGPS-ja, orientacija pa z INS meritvami treh kotov (ω , φ , κ) zasuka nosilca v prostoru. Potrebno je dodatno procesiranje podatkov, saj tehnologija ne omogoča določitve koordinat zajetih točk v realnem času. Tako za izračun koordinat točk potrebujemo merjene razdalje LS-ja, izhodiščni položaj žarka in vse tri kote zasuka plovila (INS).

Po končani zračni misiji se v postopku naknadnega procesiranja podatkov na osnovi istega časovnega trenutka oddaje signala združijo podatki meritev LS (poševne razdalje) s podatki meritev sistema POS. Za izračun prostorskih koordinat zajetih točk v referenčnem koordinatnem sistemu sledita postopka kalibriranja merskega sistema ALS in georeferenciranja, ki ga omogočajo elementi zunanje orientacije. Ko je ta postopek končan, lahko te točke uporabimo za izdelavo digitalnega modela reliefa (DMR), digitalnega modela površja (DMP), ipd.

2.3 Zajem podatkov

Ker tehnika ALS sloni na uporabi aktivnega senzorja, meritve niso odvisne od časa zajema, kar pomeni, da bi jih načeloma lahko opravljali tudi ponoči. Prav tako na zajem ne vplivajo vremenske razmere (npr. oblaki, megla, smog ipd.). Določena količina žarkov se prebije tudi skozi krošnje dreves, s čimer pridobimo podatke o terenu tudi na z vegetacijo poraščenih območjih. Količina teh žarkov je odvisna od gostote poraščenosti terena.

Iz pridobljene razdalje med laserskim razdaljemerom in točkami, od katerih se je žarek odbil, pridobimo polarne koordinate točk v lokalnem sistemu lidarja. Koordinate točk so v osnovi predstavljene z geografsko širino, geografsko dolžino in elipsoidno višino ter temeljijo na WGS-84 referenčnem elipsoidu. Glede na potrebe uporabnika se jih lahko transformira v državni ali kak drugi koordinatni sistem. Istočasno so elipsoidne višine spremenjene v ortometrične višine, ki se nanašajo na izbrani višinski datum (Webster, Dias, Liu et al., cit. po Liu, X. 2008).

Laserski žarek se lahko na svoji poti odbije od različnih površin (tla, stavbe, listje, vodne površine, avtomobili, ipd.). Ko je objekt osvetljen z lasersko svetlobo, se ta večinoma disperzno (razpršeno) odbije ter vrne nazaj v sprejemnik. V primeru, ko žarek pade na gladko površino (npr. streha avtomobila), se od nje odbije zrcalno, kar pomeni, da se žarek ne vrne nazaj v sprejemnik. Slednji odboj lahko nadaljuje pot in se difuzno (v vse smeri) odbije še od nekega drugega objekta. V tem primeru ga sprejemnik lahko zazna, vendar je izmerjena razdalja predolga (angl. multi path) in predstavlja grobo napako. Kadar se signal odbije od mirnih vodnih površin, pride do pojava absorbcije. V tem primeru je odbiti signal prešibak, da bi ga sprejemnik lahko zaznal in registriral. Absorbcija je odvisna od vpadnega kota in je še posebej močna v primeru pravokotnega vpada signala (žarka) na mirno vodno površino. Posledično sprejemnik registrira manj točk, kar se pozna kasneje pri obdelavi, saj imamo območja brez podatkov.

Lidarski sistem je zmožen registrirati več vrnjenih signalov oz. odbojev (angl. return) enega oddanega signala. Večina lidarskih sistemov ponavadi registrira prvi in zadnji odboj, nekateri sistemi pa tudi več odbojev za posamezen signal. Več odbojev se pojavi, ko laserski signal zadene tarčo, ki njegove poti ne blokira popolnoma in zato del signala nadaljuje pot do nižje ležečega objekta. Taka situacija se

ponavadi pojavi na pogozdenih območjih, kjer so škrbine med vejami in listi dreves oz. vegetacije. Beleženje več odbojev je dokaj uporabno, predvsem pri topografskem kartiranju gozdnih območij in opisu stanja in sestave gozda (Wehr, Lohr, Reutebuch et al. cit. po Liu, X. 2008).

Poleg 3D položaja zajete točke detektor beleži tudi intenziteto odbitega laserskega žarka. Intenziteta se redko uporablja in predstavlja stranski produkt tehnike ALS. Na podlagi intenzitete lahko sklepamo o vrsti materiala objekta, od katerega se je laserski žarek odbil. To nam lahko pomaga pri obdelavi podatkov, predvsem v smislu prepoznavanja objektov, vendar pa so mnogokrat vrednosti intenzitete popačene zaradi različnih motečih faktorjev (Bitenc, M., 2006).

2.4 Obdelava podatkov

Najbolj pogost rezultat obdelave točk, zajetih z lidarjem, je digitalni model reliefa (DMR), katerega lahko uporabimo v GIS analizah in kartografiji. Lahko ga pridobimo tudi z drugimi merskimi tehnikami (digitaliziranje plastnic, slikovno ujemanje, ipd.), vendar ima izdelava DMR-ja iz lidarskih podatkov določene prednosti kot so hitrejši in avtomatiziran zajem podatkov, večja količina in s tem gostota podatkov, hitrejša obdelava podatkov s programsko opremo, ipd. Digitalni model reliefa je način opisa oblikovanosti zemeljskega površja, ki vključuje višinske točke, značilne črte in točke reliefa ter druge elemente, ki ga opisujejo (naklone, plastnice, padnice itd.).

Da lahko izdelamo DMR s pomočjo lidarskih točk, je potrebno podatke predhodno obdelati. Lidar zajame veliko količino točk, lahko tudi 10 milijonov točk na kvadratni kilometer, za katere se je uveljavil izraz oblak točk. Točke oblaka lahko preprosto delimo na terenske in ne-terenske; terenske točke predstavljajo tla, dejansko površje, ne-terenske točke pa vse ostalo (stavbe, vegetacija in drugi objekti, ki ne predstavljajo površja Zemlje). Bolj podrobno pa to razdelitev določa klasifikacija točk oblaka v razrede. Za izdelavo DMR-ja je potrebno iz oblaka točk odstraniti vse ne-terenske točke, čemur pravimo filtriranje točk. Večja količina oz. gostota podatkov vpliva na izdelavo natančnejšega prikaza reliefa (pridobimo bolj natančen DMR), vendar problem nastane na pogozdenih in z drugo vegetacijo poraščenih območjih, kjer običajno po filtriranju ne-terenskih točk ostane le majhno število terenskih točk. Število preostalih točk je odvisno od gostote poraščenosti terena in prebojnosti laserskega signala. Tako imenovane praznine določimo z metodami prostorske interpolacije na osnovi znanih terenskih točk

Podatki, ki sem jih v tej nalogi obdeloval, so bili že predhodno klasificirani, kar pomeni, da so točke razdeljene v določene razrede kot prikazuje Preglednica 2-1. S pomočjo klasifikacije točk v razrede lahko pri postopku filtriranja določimo katere razrede želimo uporabiti. Po odstranjenih ne-terenskih točkah pride do manjka točk na območjih, kjer se v naravi nahajajo stavbe, gozd, vegetacija in ostali, s

filtriranjem odstranjeni objekti. Na teh območjih podatke pridobimo z interpolacijo točk okoli omenjenih območij, zato DMR tu ni tako natančen kakor drugje in je zgolj približek dejanskega stanja.

Preglednica 2-1: Klasifikacija zajetih lidarskih točk v razrede (ASPRS, 2009)

Vrednost	Pomen	Meaning
0	Ustvarjen	Created, never classified
1	Neklasificirano	Unclassified
2	Tla	Ground
3	Nizka vegetacija	Low vegetation
4	Srednja vegetacija	Medium vegetation
5	Visoka vegetacija	High vegetation
6	Stavbe	Building
7	Šum	Low point (noise)
8	Ključna točka	Model key – point (mass point)
9	Voda	Water
10	Rezervirano za ASPRS definicijo	Reserved for ASPRS Definition
11	Rezervirano za ASPRS definicijo	Reserved for ASPRS Definition
12	Dvojne točke	Overlap Points
13 – 31	Rezervirano za ASPRS definicijo	Reserved for ASPRS Definition

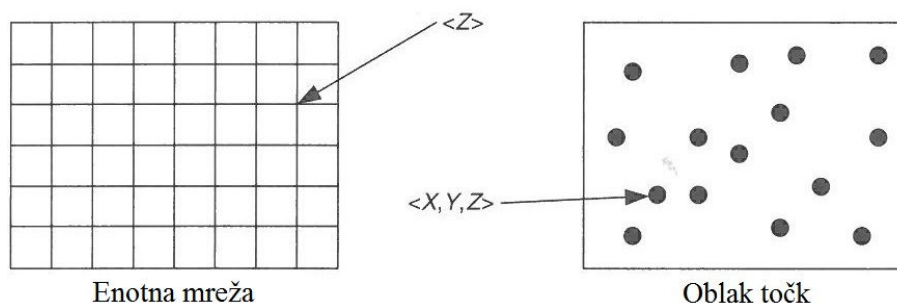
Tekom časa se je razvilo več algoritmov za izdelavo DMR-ja, katerih cilj je zagotoviti čim bolj natančen model reliefa. Ker v diplomski nalogi opisujem proste in odprtokodne programe za obdelavo lidarskih podatkov, je izbor »zmagovalnega« programa odvisen predvsem od algoritmov, ki jih posamezna programska oprema uporablja za interpolacijo manjkajočih točk.

3 ORGANIZACIJA PODATKOV IN PODATKOVNI FORMAT

Kadar izbiramo podatkovni format, moramo vnaprej predvideti, kaj želimo s podatki doseči oz. kaj je naš cilj, naš končni izdelek. Za potrebe diplomske naloge morajo podatki uporabljenim programom omogočati direkten dostop, brez predhodno izgubljenih informacij na račun podatkovnega formata. Ker želim v tej diplomski nalogi ugotoviti možnosti uporabe in obdelave lidarskih podatkov v prosti in odprtokodni programski opremi in z njeno pomočjo izdelati DMR, ki ga lahko naknadno uporabimo pri različnih GIS analizah, je potrebna uporaba podatkov z vsemi razpoložljivimi informacijami, ki jih zagotovi senzor med samim zajemom. V ta namen moramo izbrati podatkovni format, ki to omogoča. Lidarska skupnost se je sporazumela glede podatkovnega formata, ki je zelo zmogljiv oz. učinkovit za prenos zelo velikih podatkovnih nizov, naj spomnim, da se število točk v oblaku meri v milijonih, in je hkrati zadovoljiv za uporabnika in njegovo programsko opremo, s katero želi uporabiti podatke za svoje potrebe. Ta podatkovni format se imenuje LAS; je binaren in organiziran na osnovi oblaka točk.

3.1 Organizacija podatkov v oblaku točk

3D podatke, ki niso organizirani v enotno (angl. uniform) mrežo, pogosto imenujemo oblak točk, saj imajo slabo določene meje, podobne oblaku. Slika 3-1 prikazuje razliko med razporeditvijo podatkov v enotni (rastrski) mreži in oblaku točk.



Slika 3-1: Primerjava pravilne mreže in oblaka točk (povzeto po Shan in Toth, 2009: 298)

V mreži točk so koordinate posameznih točk znane, lokacija posamezne točke iz oblaka točk pa ni predhodno znana. Zato morajo biti vse tri koordinate preoblikovane v pravilen format za vsako posamezno točko. S točkami, organiziranimi v oblak točk, je težje izvajati določene operacije kot je npr. iskanje določene točke, interpoliranje lokacije, ki leži med točkami ipd. Točke, organizirane v oblak točk, so dokaj splošen način organiziranja točk, predvsem v geografskih informacijskih sistemih (GIS), ko so točke trikotno povezane z njihovimi sosedi. Takšen format imenujemo nepravilna trikotniška mreža oziroma TIN (angl. Triangulated Irregular Network). Izrazita prednost TIN mreže je ta, da omogoča natančno predstavitev nepravilne geomorfologije (grebeni, doline, ipd.) (Shan, J., Toth, C.K., 2009).

3.2 Način zapisa podatkov v datoteko

Ker točke, zajete z lidarsko tehnologijo, vsebujejo poleg položajnih podatkov tudi višinski podatek, torej vse tri prostorske koordinate, je potrebno pri izbiri formata zapisa podatkov nameniti velik del pozornosti tej tretji koordinati. Omenjeni podatki so po navadi kodirani v ASCII (American Standard Code for Information Exchange) ali v binarnem formatu. V prvem so atributi lidarskih točk ločeni z vejico. Te datoteke imenujemo CSV (comma-separated-value) datoteke. Drugi format pa omogoča shranjevanje lidarskih točk v binaren format. Potrebno je poznati razliko med omenjenima načinoma zapisov podatkov v datoteko.

3.2.1 Tekstovni ASCII podatkovni format

Tekstovne oz. ASCII datoteke so kodirane s pomočjo splošne angleške abecede in simbolov, kar pomeni, da končno ta način kodiranja uporablja 95 znakov. V ASCII formatu so podatki zapisani kot tekst (8 ali 16 bitov na posamezni znak). Te podatke lahko beremo in urejamo s preprostim programom za urejanje besedila (angl. word processor) kot je npr. Notepad. Vendar pa ASCII format ni najbolj primeren za shranjevanje lidarskih podatkov. Prvi razlog je prostor, ki ga zasede podatek v datoteki (angl. file space). Podatkovna vrednost "21432124" zasede v datoteki 9 bajtov. 8 bajtov je potrebnih za vseh 8 znakov podatkovne vrednosti in dodaten bajt za deveti znak, ki mu pravimo tudi "terminator". Terminator (ponavadi bajt, katerega vrednost je 0 oz. null) omogoča računalniškemu algoritmu ločevanje oz. razlikovanje med koncem ene podatkovne vrednosti in začetkom naslednje. Drugi razlog je, da računalniki v ASCII datotekah ne izvajajo numeričnih preračunavanj. To pomeni, da mora biti vsaka ASCII podatkovna vrednost najprej prebrana, opravljena analiza konca podatkovnega niza in nato preoblikovana v interni numerični format (32-bit integer), ki ga določen program lahko razume in obdela. Kot največjo pomanjkljivost shranjevanja podatkov v ASCII podatkovnem formatu je potrebno izpostaviti izgubo informacij, ki jih lidarski podatki vsebujejo in so kasneje pri njihovi obdelavi pomembne in v pomoč operaterju. Vpogled v količino izgubljenih podatkov omogoča Preglednica 3-1, ki sicer prikazuje podatke in njihove attribute, shranjene v LAS formatu, posredno pa prikazuje tudi, koliko podatkov se izgubi, kadar so lidarski podatki shranjeni v t.i. X, Y, Z ASCII formatu. Vse informacije, ki so ključnega pomena za klasifikacijo, se izgubijo.

Alternativo tekstovnemu kodiranju predstavlja XML (Extensible Markup Language) standardni jezik, ki zagotavlja pravilno branje podatkovnih polj pod predpostavko, da je podatkovna datoteka pravilno formirana. Prav tako kot ASCII format, ima tudi XML enake slabosti t.j. velikost porabljenega datotečnega prostora in potreba po pretvorbi teksta v binarni format (Shan, J., Toth, C.K., 2009).

3.2.2 Binarni LAS podatkovni format

LAS datotečni format je javen format za izmenjavo 3D podatkov oblaka točk med uporabniki. Čeprav je bil primarno razvit za izmenjavo lidarskih podatkov, format omogoča izmenjavo katerih koli nepravilno razporejenih tridimenzionalnih podatkov. LAS format je binarni datotečni format, ki ohranja oz. vzdržuje informacije, specifične lidarski naravi podatkov in hkrati ohranja zmerno raven kompleksnosti. Binarno kodiranje je kompaktno, kar pomeni, da za vsak posamezen podatek uporabi minimalen datotečni prostor. Binarni format je kodiran v računalniku razumljivem 4 bajtnem (heksadecimalnem) načinu predstavitve števila. Za uporabo binarnih formatov je potrebna t.i. skladnost "pisca" in "bralca" binarnega formata. Podatkovne datoteke vsebujoče lidarske podatke, shranjene v LAS datotečnem formatu, imajo končnico *.LAS. Lidarska strojna oprema shranjuje podatke v ta format, v katerem vključuje podatke komponent lidarja (GPS, INS in LS) in z njihovo pomočjo zagotovi X, Y in Z koordinate zajetih točk.

3.2.2.1 Razvoj LAS formata

Sistem LiDAR je poleg položaja točk sposoben zaznati tudi dodatne informacije, ki jih uporabnik potrebuje pri nadaljnji obdelavi. Na začetku razvoja lidarske tehnologije so vse, z lidarjem pridobljene podatke, shranjevali v preprosto berljivih, a hkrati v računalniškem smislu prostorsko požrešnih tekstovnih datotekah (npr. ASCII) ali pa v težko berljivih binarnih formatih. Največji problem je predstavljala izmenjava podatkov, saj je vsak ponudnik strojne in programske opreme uporabljal svoj interni format. Ponudnikom senzorjev (npr. Optech, Leica Geosystems) in ponudnikom programske opreme (npr. Terrasolid) je postajalo vse bolj jasno, da je potrebno razviti format, ki bo omogočal povezovanje strojne in programske opreme različnih proizvajalcev. Leta 2002 so predstavniki Z/I Imaging Corporation (danes del Intergraph Corporation), EnerQuest (danes Spectrum Mapping), Optech Corporation, Leica Geosystems in U.S. Army Corps of Engineers Topographic Engineering Center (TEC) ustanovili neformalen odbor za razvoj standardnega formata za izmenjavo lidarskih podatkov. EnerQuest je takrat za svoje potrebe uporabljal interni format, ki je omogočal izmenjavo podatkov med številnimi programskimi aplikacijami. Ta format, ki se je imenoval LAS (izhaja iz LASer), je bil binaren. EnerQuest in TEC sta se strinjala in donirala ta format javni uporabi. Za ta namen ustanovljen odbor je uporabil to originalno podatkovno specifikacijo za nastanek standarda, bolj znanega kot ASPRS (American Society for Photogrammetry and Remote Sensing) standard, za izmenjavo lidarskih podatkov. Po nekaj manjših popravkih je ASPRS LiDAR pododbor 9.3.2003 odobril prvo verzijo standarda i.s. LAS 1.0. To verzijo formata trenutno podpirajo vsi komercialni (lastniški in odprtokodni) ponudniki programske opreme za procesiranje lidarskih podatkov.

Kot sem že omenil je bila prva verzija formata izdana leta 2003, kateri je dve leti kasneje sledila prva revizija (LAS 1.1), v kateri so spremenili interpretacijo določenih polj in njihovo pozicijo. Sledili sta

še dve reviziji (LAS 1.2 in LAS 1.3) prve verzije formata, ki sta spremenili način zapisa GPS časa in točkam dodali barvno komponento. LAS 1.3 specifikacija je tudi zadnja, ki je bila 14.7.2009 odobrena s strani ASPRS odbora. Obstaja tudi predlog razvoja LAS 2.0 formata, ki bi naj predstavljal popolno prenavo formata, z namenom postati splošen standard za zapis oblakov točk in hkrati biti dovolj razširljiv, da bo omogočal izpeljavo lastniških formatov (ASPRS, 2011).

Ker so razlike med verzijami formata majhne, se bom v tej diplomski nalogi posvetil le formatu LAS 1.3.

3.2.2.2 Značilnosti LAS formata v primerjavi z ASCII formatom

V primerjavi s shranjevanjem lidarskih podatkov v ASCII datotečnem formatu, ima shranjevanje podatkov v LAS datotečnem formatu določene prednosti. Kot sem že omenil v poglavju so glavna prednost LAS formata informacije oz. atributi, ki jih format shrani za posamezno zajeto točko, kar prikazuje Preglednica 3-1.

Preglednica 3-1: Primer atributov lidarskih točk (Shan in Toth, 2009: 300)

Atribut	Opis
X, Y	Planimetrična lokacija točke na površju
Z	Višinska komponenta točke
Inzenziteteta	Intenziteta vrnjenega/odbitega laserskega pulza
GPS čas	Čas prejema vrnjenega pulza
Število odbojev	Število zaznanih odbojev za oddan pulz
Številka odboja	Številka odboja enega pulza (npr. drugi odboj od treh)
Kot zrcala	Kot zrcala skenerja v času oddanega pulza
Klasifikacija	Atribut površja dodeljen točki kot npr. vegetacija, relief, ipd.
ID točke	Unikaten identifikator točke

Kot vsak binaren zapis podatkov tudi LAS format omogoča varčevanje prostora, ki ga posamezna LAS datoteka zasede v računalniškem pomnilniku. Kot rečeno, vrednost "21432124", shranjena v ASCII formatu zasede 10 bajtov računalniškega pomnilnika, medtem ko ista vrednost, shranjena v binarnem LAS formatu, zasede le 4 bajte računalniškega pomnilnika. Glede na število atributov, shranjenih v ASCII tekstovni datoteki, lahko LAS format zmanjša njeno velikost za od 35% do 80%. Če upoštevamo, da lahko en projekt vsebuje tudi po več sto gigabajtov podatkov, potem ima ta

prihranek prostora v računalniškem pomnilniku toliko večjo težo. Primerjavo porabe prostora računalniškega pomnilnika različnih formatov kodiranja za datoteko z enim milijonom podatkov, ki za shranjevanje zahteva 32 – bitna cela števila (angl. integer), prikazuje Preglednica 3-2.

Preglednica 3-2: Primerjava kodiranja datoteke z 1 milijonom točk (Shan in Toth, 2009: 299)

Format	Primer kodiranja	Velikost v bajtih (byte)
Binarni (Int32)	iiii	4.000.000
Binarni (float)	ffff	4.000.000
Binarni (double)	dddddddd	8.000.000
ASCII (cm)	487600null	7.000.000
Fixed ASCII	0487600	7.000.000
XML	<ht>487600</ht>	15.000.000 +

Zmanjšanje velikosti datoteke z uporabo LAS formata vpliva med drugim tudi na hitrost procesiranja in branja datoteke v programski opremi za obdelavo lidarskih podatkov. Manj potrebnih bajtov na zajeto točko v datoteki pomeni tudi manj bajtov, ki jih mora program prebrati.

Kot pri vsakem podatkovnem standardu, gre tudi pri LAS formatu za kompromis med učinkovitim shranjevanjem in razkošjem podatkov, ki jih vsebuje datoteka. Algoritmi za procesiranje lidarskih podatkov so še vedno na stopnji, kjer je lahko vsak metapodatek o stanju senzorja med zajemom uporaben pri procesu klasifikacije. Na primer, če je kot skenerja določene točke enak nič, kar pomeni, da je bila točka zajeta v nadiru, potem se točka najverjetneje ni odbila od vertikalnega površja stavbe. LAS format omogoča bogato zbirko metapodatkov za posamezno točko. Po drugi strani pa si v LAS formatu točke (id) ne sledijo v določenem zaporedju, zato je pomembno, da ne domnevamo o redu točk. Smiselno so točke predstavljene le s pomočjo GPS časa. Takšna organizacija točk v datoteki omogoča hitro t.i. "real time" shranjevanje podatkov med zajemom. Temu lahko rečemo tudi "streaming format" (Shan, J., Toth, C.K., 2009).

3.2.2.3 Struktura LAS formata

Binarna LAS datoteka je v splošnem sestavljena iz treh blokov kot prikazuje Preglednica 3-3. Javna glava vsebuje splošne podatke kot so številke točk in koordinatne meje. Prvemu bloku sledijo zapisi s spremenljivo dolžino, ki vsebujejo spremenljive podatkovne tipe kot so projekcijske informacije, metapodatki, informacije paketov odbitega valovanja (angl. waveform packet information) in podatke

uporabniških aplikacij (angl. user application data). Zadnji blok, imenujmo ga točke, pa vsebuje zapise točk s koordinatami in pripadajočimi skalarnimi vrednostmi.

Preglednica 3-3: Blokovna razdelitev formata LAS

Slovensko poimenovanje	Angleško poimenovanje
JAVNA GLAVA	PUBLIC HEADER BLOCK
ZAPISI S SPREMENLJIVO DOLŽINO	VARIABLE LENGTH RECORDS
TOČKE	POINT DATA RECORDS

Namen te diplomske naloge je obdelava že predhodno pripravljenih oz. pred-procesiranih lidarskih podatkov v odprtokodnih in prosto dostopnih programih ter izdelava DMR-ja. Iz tega razloga se v nalogi nisem osredotočal na poglobljeno analizo strukture LAS formata. Vse specifikacije različnih verzij LAS formata so dostopne na straneh www.asprs.org in www.lasformat.org. Specifikacija definira binarni datotečni format za shranjevanje lidarskih podatkov.

4 PROSTA IN ODPRTOKODNA PROGRAMSKA OPREMA

Koncept odprte kode in proste delitve informacij je obstajal že davno pred računalniki, ko je človek delil svoje znanje o izdelavi orodja, pripravi ognja in hrane s sočlovekom, saj je le tako lahko preživel kot posameznik in kot del skupine. Čeprav se pojem odprta koda navezuje na več področij človekovega ustvarjanja, se v diplomski nalogi posvečam področju programske opreme t.j. GIS programske opreme oz. programja⁷, s katero je mogoče obdelovati geografske podatke, pridobljene z lidarsko tehnologijo.

Začetki prostega programja⁸ (angl. free software) segajo v 60. leta 20. stoletja, ko se je z IBM-ovimi računalniki razvila prosta programska oprema, katero so uporabniki lahko spreminjali, izboljševali in razdeljevali med seboj. Primer proste programske opreme, ki so jo uporabniki prejeli skupaj s kodo, je bil operacijski sistem Unix. V 70. letih 20. stoletja se je pojavila lastniška programska oprema, ki so jo z raznimi copyright⁹ omejitvami začeli vsiljevati proizvajalci, z namenom zaslužka. Richardu Stallmanu in univerzi Berkeley iz Kalifornije omejitve niso bile všeč, zato so konec 70. in v začetku 80. let 20. stoletja začeli izdajati programsko opremo pod drugačnimi licencami, z namenom njene osvoboditve. Gre za t.i. prosto dostopno programsko opremo. Sprva so se z njo ukvarjali predvsem raziskovalci in programerski navdušenci, kasneje, sredi 90. let 20. stoletja, pa so začela nastajati podjetja, ki so v njej spoznala poslovne zmožnosti.

4.1 Prosto programje oz. free software

Začetnik promocije prostega programja je Richard Stallman, v poznih 70. in začetku 80. let 20. stoletja programer na univerzi MIT¹⁰. V svojih programih je vedno pozival uporabnike, naj jih delijo z ostalimi, jih izboljšujejo, se iz njih učijo in svoje znanje delijo naprej, ko z njihovo uporabo končajo. V tistih časih je bilo deljenje programov in informacij med razvijalci in raziskovalci običajna praksa, saj so računalnike uporabljale predvsem univerze in njihovi raziskovalni laboratoriji.

Leta 1983 je lansiral GNU Project in skoval pojem free software. Znotraj projekta je začel razvijati prost in prosto dostopen operacijski sistem GNU¹¹ podoben sistemu Unix, z vsemi potrebnimi orodji za delo in vpeljal gibanje Free Software Movement (FSM). Dve leti kasneje je izdal GNU manifest, kjer je opisal GNU projekt in razložil pomembnost prostega programja ter ustanovil fundacijo Free Software Foundation (FSF), ki promovira splošno svobodo ustvarjanja, razdeljevanja in modificiranja

⁷ Sistemski in uporabniški programi kot del računalniškega sistema.

⁸ Prosto programje oz. prosto dostopna programska oprema.

⁹ Copyright je avtorska pravica, ki avtorju zagotavlja uresničevanje premoženjskih in osebnih interesov v zvezi z iskoriščanjem lastnega avtorskega dela. Je izključne narave in omejuje uporabo avtorskega dela brez soglasja avtorja.

¹⁰ Massachusetts Institute of Technology je raziskovalna ustanova in univerza, Cambridge, ZDA.

¹¹ Gnu's not Unix

računalniške programske opreme. V okviru FSF je prosto dostopna programska oprema definirana s štirimi lastnostmi oz. »svobodami« (angl. freedoms) kot jih imenuje R. Stallman (GNU, 2010):

Freedom 0: Svobodna uporaba programa za katerikoli namen¹².

Freedom 1: Svobodno preučevanje delovanja programa in prilagajanje potrebam uporabnika.

Freedom 2: Svobodno razdeljevanje in kopiranje programske opreme.

Freedom 3: Svoboda po izboljšanju programske opreme in objavi popravkov oz. modificiranih verzij v dobro skupnosti.

Prosto programje je programska oprema, ki vsakemu uporabniku omogoča uporabo, kopiranje in razdeljevanje osnovne ali modificirane programske verzije brezplačno ali proti plačilu. To pomeni, da mora biti izvorna programska koda uporabniku na razpolago. Takrat lahko program razvijamo, izboljšujemo, odpravljamo napake, ipd. Obstaja rek, ki opisuje celotno filozofijo na kratko: "*Think as a free speech, not as a free beer*", kar pomeni, da "*free*" (prost, svoboden, brezplačen,...) ni mišljen kot brezplačno pivo (angl. free beer), ampak kot svoboda (angl. freedom), zato se izraz "*free software*" nanaša na svobodo in ne na ceno, kot lahko zavede marsikoga, ki te tematike ne pozna dovolj dobro. Podjetja, ki izdelujejo lastniško programsko opremo, ponavadi uporabljajo izraz "*freeware*", kadar se sklicujejo na ceno, t.j. brezplačnost. Zaradi potencialne zmešnjave, ko neko podjetje ocenjuje svoj produkt kot "*free software*", je priporočljivo preveriti pogoje razdeljevanja, kjer lahko uporabnik vidi ali ima resnično vso svobodo, na katero programska oprema namiguje ali ne.

4.2 Odprtokodno programje oz. open source software

Leta 1997 je Ric Raymond objavil analizo hekerske skupnosti in načela prostega programja pod naslovom "*The Cathedral and Bazaar*". Referat je v začetku leta 1998 prejel pomembno pozornost in je bil eden izmed razlogov za izdajo internetnega brskalnika Netscape Communicator kot prostega programja, podjetja Netscape Communications Corporation. Koda je danes bolj znana kot Mozilla Firefox in Thunderbird. Netscape-ova izdaja programa v takšni obliki je spodbudila Raymonda in ostale somišljenike, da so začeli razmišljati, kako približati principe in koristi prostega programja komercialni programski industriji. Sklenili so, da ideja, ki jo je zastopala FSF v smislu socialnega aktivizma, ni privlačna za podjetja kot je Netscape. Raymond in ostali¹³ so začeli iskati način za preoblikovanje gibanja FSM, z namenom poudariti poslovni potencial z razdeljevanjem izvorne programske kode. Izbrali so novo ime in sicer odprta koda (angl. open source). Istega leta je bila osnovana iniciativa Open Source Initiative (OSI) z namenom pospeševanja uporabe termina odprta koda in širjenje odprto kodnih načel.

¹² Ne glede na to ali je namen komercialne, izobraževalne ali raziskovalne narave.

¹³ Todd Anderson, Chris Peterson, John "maddog" Hall, Larry Augustin, Sam Ockman, Michael Tiemann

Richard Stallman in FSF sta nasprotovala novemu pristopu OSI organizacije, saj sta bila prepričana, da z njenim ozkim osredotočanjem na izvorno kodo uničuje filozofske in socialne vrednote prostega programja in zmanjšuje vrednost svobode računalniškega uporabnika. Kljub vsemu je Stallman mnenja, da so uporabniki obeh izrazov zavezniki v boju proti lastniški programski opremi. Termina, čeprav različna, opisujeta skoraj identično programsko opremo. Skoraj vsak odprtokodni program je prosto dostopen in skoraj vsak prosto dostopen program je odprtokoden.

Na prvi pogled se zdi, da oba izraza (prosto programje in odprtokodno programje) zastopata različni področji, kljub njunemu velikemu prekrivanju. Njuno pravo nasprotje je lastniško programje (angl. proprietary software).

Če se FSF zavzema predvsem za svobodno uporabo programske opreme, se na drugi strani OSI zavzema sicer za skorajda enake cilje, ampak iz bolj pragmatičnega vidika. Odklanja stališča, da je lastniško-komercialna programska oprema apriori nemoralna in poudarja predvsem tehnične pomanjkljivosti v primerjavi z odprtokodnim pristopom (Mlakar, M., 2009).

Odrpta koda opisuje splošno rabo v produkciji in razvoju, ki promovira dostop končnega izvora produkta. Nekateri jo smatrajo za filozofijo, drugi za pragmatično metodologijo. Odrpta koda ne pomeni samo dostop do izvorne kode. Je vsa programska oprema, ki je izdana pod OSI certifikatom, za pridobitev katerega mora njena programska licenca ustrezati 10 lastnostim. V splošnem so te lastnosti samo drugačna realizacija štirih »svobod«, ki jih določa FSF. Programska licenca odprtokodnega programa mora izpolnjevati naslednje kriterije (Benčina, 2003 cit. po Škrubej, U. 2009):

1. **Svobodna distribucija:** Licenca ne sme nikomur omejevati prodaje ali predaje programske opreme kot komponente združenih programskih paketov, ki vsebujejo več različnih kod, hkrati pa ne sme zahtevati nikakršnih plačil.
2. **Izvorna koda:** Program mora vsebovati izvorno kodo. V primeru da se program ne distribuira skupaj z izvorno kodo, mora biti ta brezplačna in javno dostopna. Oblika izvorne kode mora programerju omogočati njeno spreminjanje. Namerno zakrivanje ali drugačno oteževanje uporabe izvorne kode je prepovedano.
3. **Izpeljana dela:** Licenca mora dovoljevati spreminjanje in izpeljavo novih programskih rešitev, ki se morajo distribuirati pod enakimi licenčnimi pogoji kot originalna programska oprema.
4. **Integriteta avtorja izvorne kode:** Licenca lahko omejuje distribucijo modificirane izvorne kode samo, če licenca dovoljuje porazdeljevanje popravkov z namenom spreminjanja programa v času prevajanja. Dovoljevati mora distribucijo programske opreme izdelane iz

spremenjene izvorne kode. Lahko zahteva, da izpeljana dela nosijo drugačno ime oz. različno številko različice od originalne programske opreme.

5. **Prepoved diskriminacije posameznikov ali skupin:** Licenca ne sme diskriminirati nobenega posameznika ali skupine posameznikov.
6. **Prepoved diskriminacije posameznih področij dejavnosti:** Licenca ne sme nikogar omejevati pri uporabi programa na posameznem področju dejavnosti (npr. poslovni nameni, genetika, itd.).
7. **Distribucija licence:** Licenčne pravice se morajo nanašati na vse uporabnike, katerim je program razdeljen, brez dodatne potrebe po licenciranju.
8. **Licenca ne sme biti specifična za produkt:** Pravice, vezane na program, ne smejo biti odvisne od tega ali je program del določene programske distribucije. Če je program izločen iz distribucije in uporabljen oz. razdeljen naprej pod pogoji programske licence, morajo vse stranke, katerim je bil program razdeljen, imeti enake pravice kot tiste, ki so jim bile dodeljene pravice za prvotno programsko distribucijo.
9. **Licenca ne sme omejevati druge programske opreme:** Licenca ne sme postavljati omejitev nad drugo programsko opremo, ki je distribuirana z lastniško programsko opremo. Na primer, licenca ne sme zahtevati, da mora biti vsa ostala programska oprema, ki je porazdeljena na istem mediju, prav tako odprtokodna programska oprema.
10. **Licenca mora biti nevtralna do tehnologije:** Prevezemanje licence ne sme biti omejeno na posamezne tehnološke rešitve ali vmesnike.

4.3 Licence odprtokodnega in prostega programja

V zadnjih letih je nastala množica različnih licenc, ki sta jih odobrili FSF in OSI. Vsaka licenca, ki zagotavlja uporabniku specifične svoboščine¹⁴, je licenca prostega in odprtokodnega programja¹⁵. V diplomski nalogi so predstavljene samo licence, katerih pogoji določajo uporabo in omejitve programja, ki ga bom v nalogi opisal ali kasneje uporabil pri obdelavi lidarskih podatkov. Spodaj našteje licence lahko ponujajo različno stopnjo varovanja štirih »svobod«. Te licence so:

- GNU General Public License (GPL)
- GNU Lesser General Public License (LGPL)
- Berkeley Software Distribution License (BSD)
- Massachusetts Institute of Technology License (MIT)

¹⁴ Freedoms.

¹⁵ Prosto in odprtokodno programje ali free and open source software (FOSS).

- Mozilla Public License (MPL)
- Apache License

Značilnost oblike licenciranja prostega programja je ta, da se lahko uporabnik odloči ali bo licenco sprejel ali ne. Torej, uporabnik lahko uporablja program brez sprejetja licence oz. brez strinjanja z njenimi pogoji uporabe, toda, če želi uporabnik vršiti dodatne pravice, ki jih zagotavlja licenca prostega programja kot je porazdeljevanje programja, mora sprejeti in upoštevati pravice in dolžnosti, ki jih narekuje licenca.

Licence odprtokodnega programja lahko v splošnem delimo na dve skupini. V prvo skupino spadajo licence, s katerimi želimo tudi v bodoče ohraniti odprtost programja. Te licence so copyleft¹⁶ licence. V drugo skupino spadajo licence, ki dajejo uporabniku popolno svobodo. Te licence so permissive¹⁷ oz. non-copyleft licence.

4.3.1 GNU GPL licenca

Večina licenc programske opreme prepoveduje svobodno distribucijo in spreminjanje programja. Nasprotno pa licenca GNU GPL licenca (v nadaljevanju GPL) zagotavlja vse štiri »svobode«. Spada med copyleft licence in zagotavlja svobodo uporabe, kopiranja in distribuiranja prostega programja. GPL licence ne uporablja samo FSF za njene programe, ampak jo lahko uporabljajo tudi drugi avtorji, ki želijo izdati delo pod pogoji, ki jih navaja. Je najpomembnejša licenca prostega programja, saj je večina prostega programja distribuirana pod njenimi pogoji. Namenjena je zagotavljanju uporabnikove svobode pri spreminjanju vseh verzij programa ali uporabi njegove izvirne kode. Glavni razlog njene popularnosti je njen virusni efekt, to je koncept copyleft.

Prva verzija (v1) GPL licence je bila izdana leta 1989, katere uporaba ni več zaželjena, druga (v2) je izšla leta 1991, ki jo še vedno uporablja večina programja, leta 2007, pa je izšla tretja verzija (v3) GPL licence.

Pomembnejše lastnosti GPL v3 (FSF, 2010):

- Licenca se lahko uporabi za kakršnokoli programsko opremo, vendar se je ne sme spreminjati.

¹⁶ Copyleft je politika licenciranja, ki podeljuje vsakemu uporabniku pravico do rabe, porazdeljevanja in spreminjanja računalniških programov, dokumentov ipd. ter zahteva, da so novi izdelki ali modificirane verzije prav tako prosto dostopne. Je obratna načelu copyright, saj odpravlja vse pravice, ki se navezujejo na avtorje.

¹⁷ Permissive oz. non-copyleft licenca vsebuje minimalne zahteve glede oblike in načina porazdeljevanja programja. Kot copyleft licenca zagotavlja iste svoboščine glede uporabe, študija in privatnih modifikacij. Razlikujeta se le pri načelu porazdeljevanja modificiranih ali nemodificiranih verzij, kjer non-copyleft licenca uporabniku dovoljuje dodajanje novih omejitev.

- GPL licenca jamči svobodo pri porazdeljevanju in spreminjanju prostega programja in s tem zagotavlja, da ostane programje prosto za vse njegove uporabnike.
- Nanaša se na večino¹⁸ programja FSF-ja in na vse drugo programje, katerih avtorji so se zavezali k njeni uporabi.
- Dovoljuje razširjanje kopij prostega programja in tudi zaračunavanje za to storitev, če avtor tako želi.
- Spremembe programa mora urejevalec poslati nazaj avtorju. Redistribucija, vključno z vsemi spremembami, mora biti v skladu s pogoji GPL.
- Zavezuje avtorja k zagotovitvi izvirne kode uporabniku/stranki pri distribuciji programa. Licenca mora ponuditi oskrbo izvirne kode na zahtevo stranke za obdobje vsaj treh let.
- Dovoljuje spreminjanje programja ali uporabo njegovih delov v novem prostem programju v skladu s pogoji GPL.
- Varuje uporabnikove pravice z omejitvami, s katerimi prepoveduje vsakomur kratenje in zahtevo po predaji pravic. Te omejitve se preslikajo tudi v odgovornosti uporabnika, kadar razširja kopije programja ali ga spreminja. Na primer, če uporabnik razširja kopije, bodisi zastonj ali za plačilo, mora dati prejemnikom vse pravice, ki jih je imel sam. Zagotoviti jim mora dostop do izvirne kode in jim pokazati izvornik licence, da bodo poznali svoje pravice.
- Razvijalci, ki uporabljajo GNU GPL varujejo svoje pravice z dvema korakoma:
 - brani avtorske pravice nad programsko opremo
 - ponuja jim licenco, ki jim daje pravno dovoljenje za razmnoževanje, razširjanje in/ali spreminjanje programja.
- Za prosto programje pod GPL licenco ni nobenega jamstva, s čimer licenca ščiti vsakega avtorja programja in avtorja licence. V dobro uporabnika in avtorja programja zahteva GPL, da so vse modificirane verzije označene kot spremenjene. S tem se ne škodi ugledu prejšnjih avtorjev.
- Zaradi nevarnosti, da bi razširjevalci prostega programja s patenti naredili program lastniški (angl. proprietary), GPL jamči, da tak patent ne more biti licenciran kot prosto programje.

¹⁸ Nekatero programje licencirano po načelih FSF je zavezano licenci LGPL.

4.3.2 GNU LGPL licenca

Lesser General Public License, včasih znana kot GNU Library General Public License, je licenca za prosto programje, ki jo je izdal FSF. Izdelana je bila kot kompromis med čvrsto copyleft GPL in bolj dovoljujočimi licencami kot sta BSD in MIT licenci. LGPL je bila prvič izdana leta 1991, vendar poimenovana kot druga različica (verzija 2), zaradi sorodnosti z drugo verzijo GPL licence. Leta 1999 so na LGPL licenci odpravili manjše pomanjkljivosti, ji spremenili ime v današnjo obliko in izdali revidirano verzijo 2.1. S spremembo imena je FSF hotela poudariti, da licenca ni uporabna za vse programske knjižnice. Leta 2007 je FSF izdala zadnjo verzijo licence i.s. tretjo različico kot seznam dodatnih dovoljenj priloženih GPL licenci. Običajno se jo sicer uporablja za programske knjižnice (v nadaljevanju knjižnice), čeprav je tudi uporabna za samostojne aplikacije. Za knjižnice lahko uporabljamo tudi GPL licenco, vendar LGPL dovoljuje uporabo knjižnic v lastniških oz. komercialnih programih, medtem ko GPL spodbuja uporabo le-teh samo v prostem programju (FSF, 2010).

4.3.3 BSD licenca

Berkeley Software Distribution licence so družina svobodnih licenc za prosto programje. Originalno licenco so poimenovali po operacijskem sistemu Berkeley Software Distribution, podobnem Unix-u. Ta licenca je bila problematična, saj je vsebovala reklamno oz. oglaševalsko (angl. advertisement) klavzulo, ki je zahtevala, da mora vsaka reklama, ki omenja s to licenco pogojen program, vključiti stavek: »Ta produkt vsebuje program, ki ga je razvila Univerza v Kaliforniji, Berkely in njeni sodelavci« (angl. »*This product includes software developed by the University of California, Berkeley and its contributors.*«). Če bi licenco uporabljali samo za BSD operacijski sistem, ne bi bilo nobenih težav, vendar pa je licenco začelo uporabljati vedno več programskih razvijalcev, ki pa so v omenjeni klavzuli navedli svoje ime oz. institucijo. Posledično je s tem nastajalo vedno več podobnih licenc, kar je povzročalo zmedo in težave uporabnikom prostega programja. Po dogovoru razvijalcev BSD-ja in Richarda Stallmana so originalno licenco revidirali. Tako danes poznamo dve BSD licenci, ki spadata pod licence za prosto programje (OSI, 2010):

1. New BSD License¹⁹ / Modified BSD License²⁰ (leta 1999 popravljena originalna licenca)
2. Simplified BSD License¹⁹ / FreeBSD License²⁰

¹⁹ New BSD licenca kot jo imenuje iniciativa OSI

²⁰ Modified BSD licenca kot jo imenuje fundacija FSF

Originalne licence OSI ne smatra za odprtokodno, toda FSF jo vključuje med licence za prosto programje, vendar pa ne velja za kompatibilno²¹ z GPL licenco. Simplified oz. Free BSD licenco OSI smatra za odprtokodno in FSF za kompatibilno z GPL licenco.

Lastnosti:

- Spadajo pod non-copyleft oz. permissive licence, ki dovolijo spremenjenim programom vključitev v lastniške produkte in programe. Edina zahteva je, da se v izvorni kodi in dokumentaciji programa navede imena vseh avtorjev.
- Dovoljujejo uporabo in distribucijo programske opreme in izvorne kode.
- Lahko se spreminja pridobljeno izvorno kodo programa, licenciranega z BSD, ter se ga nato prodaja v binarni obliki, ne da bi se zraven distribuirala tudi spremenjena izvorna koda. Še vedno gre za odprto kodo, saj definicija odprte kode ne zahteva, da imajo izpeljana dela originalno licenco.
- Razlika med Modified BSD License in FreeBSD License je v tem, da se v slednji imen avtorjev ne sme uporabljati za promocijo izdelka brez predhodnega pisnega dovoljenja.

4.3.4 MIT licenca

Tudi MIT X license in X11 license. Zelo liberalna oz. svobodna angl. permissive, non-copyleft licenca za prosto programje, ki je bila razvita na Inštitutu za tehnologijo v Massachusettsu (angl. Massachusetts Institute of Technology). Je podobna BSD licencam in kompatibilna z GPL licenco, kar pomeni, da GPL dovoljuje kombiniranje in ponovno razdelitev s programi, ki uporabljajo MIT licenco. Ne ščiti avtorskih pravic, zato jo lahko ostali razvijalci po želji spreminjajo, glede na svoje potrebe (OSI, 2010).

Lastnosti:

- Zagotavlja brezplačnost programja in pripadajoče dokumentacije za kogarkoli, ki je dobil kopijo programa.
- Vsakdo lahko brez omejitev pravic uporablja, kopira, spreminja, združuje, objavlja, razdeljuje oz. distribuira, naprej licencira s podlicencami (angl. sublicense) in prodaja kopije programja in dokumentacije.
- Licenca mora biti dodana oz. vključena v vseh kopijah programja in dokumentacije.

²¹ Kompatibilnost dveh licenc pomeni, da lahko združujemo izvorni kodi pod različnima licencama v enem večjem programu

4.3.5 MPL licenca

Prvo različico (1.0) MPL licence je leta 1998 razvila odvetnica Mitchell Baker za korporacijo Netscape Communications Corporation, drugo različico (1.1), pa je razvila več kot deset let kasneje za potrebe Mozilla Foundation. Odobrili sta jo OSI in FSF, kar pomeni, da je označena kot licenca za prosto kot tudi za odprtokodno programje. Označena je kot križanec (hibrid) BSD in GNU GPL licence. Velja za šibko copyleft licenco, saj ima v nasprotju z MIT licenco kar nekaj kompleksnih omejitev, zaradi katerih je nekompatibilna z GNU GPL licenco. Toda različica 1.1 vsebuje predpis, ki omogoča prvotnemu razvijalcu, da določi del programa kot več licenčen (angl. multiple-licensed). To pomeni, da prvotni razvijalec dovoljuje uporabo dela originalne ali modificirane kode pod drugo, sicer določeno alternativno licenco. Za razliko od strong copyleft licenc lahko kodo, licencirano z MPL, združujemo z datotekami določenimi pod lastniškimi licencami v enem programu (OSI, 2010).

Lastnosti:

- Licenca omogoča binarnim²² datotekam, katerih izvorna koda ni bila vzeta iz originalnega programa t.j. gre za novo kodo, da ostanejo zasebne in se jih lahko prodaja.
- Omogoča, da določene modifikacije programa postanejo zasebne. Definicija modifikacije programa v MPL pravi, da je to vsaka sprememba v datoteki originalnega programa in vsaka nova datoteka, v katero je bil prepisan določen del izvorne kode originalnega programa. Iz tega sledi, da lahko ostanejo zasebne samo datoteke, ki v celoti vsebujejo povsem novo izvorno kodo.
- Izvorna koda, ki je kopirana ali spremenjena pod MPL licenco, mora ostati pod isto licenco.
- Dovoljuje uporabo, s to licenco licenciranega programja, vsakomur, tudi podjetjem, za kakršen koli namen.
- Zagotavlja pravico po spreminjanju in distribuciji za privatne potrebe, tudi znotraj podjetja ali organizacije. Prav tako je dovoljeno spreminjanje izvorne kode in uporaba le te v privatne namene.
- Dovoljuje distribucijo (izven organizacije) celotnega in nespremenjenega programa pod pogoji MPL, vendar pa je potrebno dodati vsaki kopiji izvorne kode kopijo licence.

²² Datoteka, v kateri so podatki ali strojni ukazi zapisani kot zaporedje bitov, ki jih ni mogoče smiselno tolmačiti v formatu ASCII, npr. izvedljiva datoteka.

4.3.6 Apache License

Apache licenca je licenca za prosto in odprtokodno programje. Njen avtor je Apache Software Foundation (ASF). Kot vsaka licenca zahteva ohranjanje avtorskih pravic (copyright). Dovoljuje uporabo izvorne kode za razvijanje lastniškega programja kot tudi prostega in odprtokodnega programja, kar pomeni, da ni copyleft licenca. Trenutno veljavna verzija licence je 2.0, ki je bila izdana leta 2004. Kot vse ostale non-copyleft licence tudi Apache licenca ne zahteva, da so vse nadaljne modificirane verzije različnega programja porazdeljene pod isto licenco. Licenca Apache License 2.0 je kompatibilna z GNU GPL ver3 licenco, toda ker ji je licenca GPL »nadrejena« (angl. superset), mora projekt, ki združuje GPL ver3 in Apache 2.0 licenci, biti licenciran pod GNU GPL ver3 licenco. FSF smatra vse Apache licence za nekompatibilne z GNU GPL licencami verzije 1 in 2.

Lastnosti (Apache License, 2011):

- Licenca dovoljuje, da lahko uporabnik prosto uporablja in prenaša programje licencirano z Apache licenco kot celoto ali kot del za kakeršen koli namen (osebni, interni, komercialni, podjetniški).
- Prepoveduje porazdeljevanje programja izdanega pod Apache licenco brez navedbe očetovstva.
- Zahteva, da se vsakemu Apache programju, ki se ga porazdeljuje, doda kopija licence in jasno navede očetovstvo ASF-ja.

Za uporabo FOSS programa na FGG kot tudi za raziskovalne namene je potrebno vedeti, kakšne so omejitve licenc pri porazdeljevanju in modificiranju programja. Pri uporabi programa v pedagoške namene na FGG gre predvsem za uporabo in porazdeljevanje programa med študente. Vse zgoraj opisane licence dovoljujejo prosto uporabo in porazdeljevanje programja, ki je izdano pod pogoji teh licenc. V nadaljevanju naloge sem predstavil določene programe bolj podrobno. Vsi ti programi so izdani pod opisanimi licencami. Z vidika licenciranja so primerni za pedagoške namene na FGG.

Za uporabo v raziskovalne namene, kjer lahko predvidevamo občasne modifikacije programa, pa se licence malenkostno razlikujejo. Copyleft licence določajo, da je potrebno vsako spremenjeno kodo ali le del kode določenega programa izdati pod isto licenco kot je bila izdana originalna koda. Non-copyleft licence pa te zahteve nimajo. Ko se uporabnik, institucija, organizacija ali podjetje odloči za razvoj določenega programa ali uporabo že obstoječe izvorne kode v lastne namene, je priporočeno licence preučiti bolj podrobno, najbolje s strokovnjakom prava, ki pozna področje prostih in odprtokodni licenc.

5 IZBIRA PROGRAMOV IN OCENA PRIMERNOSTI ZA UPORABO

Prostih in odprtokodnih programov²³, ki so sposobni obdelave različnih GIS podatkov je na FOSS tržišču mnogo, saj je področje FOSS GIS programov v zadnjih letih doživelo velik razmah. V tem poglavju je predstavljen način iskanja programov, primernih za obdelavo lidarskih podatkov, njihov pregled in analiza njihove primernosti za namen diplomske naloge. Na koncu poglavja sledi priporočilo najboljših kandidatov za praktično uporabo. Izbira najboljših kandidatov sloni na oceni, opravljeni s pomočjo kriterijev in subjektivnega mnenja avtorja diplomske naloge.

V prejšnjih poglavjih sem uporabljal izraz programje oz. programska oprema, saj sem s tem želel izpostaviti dejstvo, da govorimo o več programih hkrati. Na tem mestu želim opozoriti, da je v tem poglavju, med drugim, uporabljen tudi izraz projekt. Termin projekt opisuje ne samo lastnosti in funkcionalnosti posameznega programa, ampak tudi ostale, v svetu prostih in odprtokodnih programov, nujno potrebne sestavine, ki prispevajo k uspešnosti, kakovosti in praktični uporabnosti samega programa. Te sestavine so dokumentacija, skupnost, podpora, ipd. ter so podrobneje opisane v nadaljevanju poglavja. Vse te dodatne sestavine niso enako pomembne za končno oceno programa. Kriteriji za ocenjevanje programov so definirani tako, da se ocenjuje uporabnost samega programa in pa tudi prisotnost in kakovost dodatnih sestavin. Tako pridobimo bolj realistično oceno celotnega projekta, saj le-ta vsebuje na primer tudi kako dobro je sam program dokumentiran, kako dobro je organizirana pomoč uporabniku, ipd.

Kot lahko različne programe delimo na zmogljivejše in tiste manj zmogljive, lahko na podoben način ocenimo tudi nivo znanja uporabnikov. Na osnovi znanja in zahtev uporabnika sem za oceno projektov določil dve skupini uporabnikov: začetnik in raziskovalec ter s tem tudi dve skupini kriterijev, po katerih sem ocenjeval izbrane programe; kriteriji za pedagoški namen in kriteriji za raziskovalni namen. Pri pedagoškem namenu je potrebno upoštevati dejstvo, da se študent šele spoznava z lidarsko tehnologijo in uporabo programov v njen namen. Tako mora imeti program zraven vseh tehničnih faktorjev tudi določeno stopnjo intuitivnosti. Hkrati mora študentu omogočiti dober vpogled v samo obdelavo, saj mu tako program omogoča povezovanje znanja teorije in praktične uporabe lidarske tehnologije. Pri raziskovalnem namenu so zahteve in stopnja poznavanja tehnologije lidarja na višjem nivoju kot pri študentu, zato so tudi zahteve raziskovalca za izbiro določene programske opreme višje.

S skupinama kriterijev želim pokazati način ocenjevanja projektov za potrebe različnih tipov uporabnikov. V nalogi sem se posvetil bolj pedagoškemu namenu, vendar želim hkrati omeniti in

²³ Angl. free and open source software (FOSS). Angleško kratico FOSS uporabljam predvsem zaradi njene prepoznavnosti in pomena na področju odprte kode.

pokazati določene smernice in način ocenjevanja ter izbire programov tudi za raziskovalne namene. Tako sem zbrano skupino programov ocenil še z definiranimi kriterij za raziskovalni namen.

5.1 Iskanje ustreznih programov in izbira kandidatov za ocenitev

Iskanja FOSS GIS programov, ki omogočajo obdelavo lidarskih podatkov, sem se lotil brez posebnih izkušenj in znanja o samih FOSS programih in njihovi skupnosti. Najprej sem poiskal pomoč pri kolegih iz stroke, nato pa nadaljeval iskanje s pomočjo svetovnega spleta. Določil sem nekaj osnovnih točk oz. ciljev, ki sem jih uporabil kot ključne besede za poizvedovanje po spletu s pomočjo spletnih brskalnikov Google in Yahoo. S poizvedbo sem nadaljeval po spletnih straneh kot so Sourceforge.net, Freshmeat.net, Opensourcegis.org in FreeGIS.org, kjer je na enem mestu zbrana večina prostih in odprtokodnih programov. Wheeler (2010) pravi, da so ti arhivi dober pokazatelj uspešnosti posameznega projekta, saj poleg podatkov o projektu (št.prenosov, dokumentacija, licenca, kratek opis, ipd.) te strani vsebujejo tudi informacije o zrelosti projekta (Freshmeat.net), statusu razvoja programa (Sourceforge.net) ipd. Te informacije so začetnemu uporabniku FOSS programja v pomoč pri izbiri in odločitvi za posamezen program.

Cilji iskanja:

1. Zaradi velikega števila različnih FOSS programov, ki zastopajo različna področja informatike in računalništva, je bilo potrebno najprej določiti tip programja. Odločil sem se za splošno GIS programje.
2. Prva točka ni bila dovolj specifično definirana, saj sem dobil obsežen seznam različnih GIS programov, zato sem iskanje dodatno omejil na programe, ki omogočajo kakršnokoli manipuliranje ali obdelavo podatkov, pridobljenih z lidarsko tehnologijo.

Ko sem našel določen program, sem s pomočjo pogosto zastavljenih vprašanj (angl. FAQ²⁴) na domači spletni strani projekta in z zgoraj omenjenimi spletnimi arhivi, poskušal izvedeti ključne lastnosti programa. Če sem s tako pridobljenimi informacijami ugotovil ustreznost programa, sem ga uvrstil na seznam. Če program ni ustrežal potrebam, sem ga izločil, toda večkrat se je neustreznost izkazala šele kasneje, po podrobnem pregledu spletnih strani določenega projekta. Rezultat iskanja (z upoštevanjem zastavljenih ciljev) predstavlja seznam programov v preglednici Preglednica 5-1.

²⁴ Frequently Asked Questions

Preglednica 5-1: Širok seznam programov, ki omogočajo manipulacijo lidarskih podatkov

Št.	Ime	Spletni naslov
1	ALDPAT	http://lidar.ihrc.fiu.edu/lidartool.html
2	FullAnalyze	http://code.google.com/p/fullanalyze/
3	FUSION/LDV	http://forsys.cfr.washington.edu/fusion/fusionlatest.html
4	FWTools	http://fwtools.maptools.org/
5	GDAL	http://www.gdal.org/
6	Generic Mapping Tools	http://gmt.soest.hawaii.edu/
7	GEON Points2Grid Utility	http://lidar.asu.edu/points2grid.html
8	GRASS GIS	http://grass.osgeo.org/
9	gvSIG in DielmoOpenLidar	http://www.gvsig.org/ http://www.dielmo.com/
10	ILWIS	http://www.ilwis.org/
11	LAStools	http://www.cs.unc.edu/~isenburg/lastools/
12	libLAS	http://liblas.org/
13	MapWindow	http://www.mapwindow.org
14	MCC-LIDAR	http://sourceforge.net/projects/mcclidar/
15	MicroDEM	http://www.usna.edu/Users/oceano/pguth/website/microdem/microdem.htm
16	OPALS	http://www.ipf.tuwien.ac.at/opals/opals_docu/index.html
17	OSSIM	http://www.ossim.org/
18	Quantum GIS	http://www.qgis.org/
19	QuikGrid	http://www.galiander.ca/quikgrid/
20	SAGA GIS	http://www.saga-gis.org/
21	TNTmips free	http://www.microimages.com/tntlite/
22	uDig in JGrass	http://udig.refrations.net/ http://code.google.com/p/jgrass/

23	VTK	http://www.vtk.org/
24	Whitebox GAT	http://www.uoguelph.ca/~hydrogeo/Whitebox/index.html

Za oceno primernosti FOSS programov, s katerimi želimo obdelovati lidarske podatke in izdelati DMR, je potreben ozek seznam kandidatov, katere nato ocenimo bolj podrobno. Do ožjega seznama kandidatov lahko pridemo s pomočjo selekcije (van den Berg, K., 2005). Ker nisem vedel, ali vsi ti programi s seznama ustrezajo tudi potrebam diplomske naloge, sem določil selektivne kriterije. S temi kriteriji sem želel seznam skržiti na kandidate za podrobno ocenitev primernosti.

Selektivni kriteriji:

1. FOSS namizni GIS programi (angl. desktop GIS²⁵) za obdelavo lidarskih podatkov: Cilj diplomske naloge narekuje izključitev določenega programja, ki ne ustreza FSF in OSI. Tako sem iz seznama izključil vse shareware²⁶ programe in freeware²⁷ programe (OPALS, TNTmips free, MICRODEM, ALDPAT, GEON Points2Grid). Izločil sem tudi programske knjižnice²⁸, saj niso namizni programi (LAsTools, libLAS, GDAL) in orodja za vizualizacijo ter pregledovanje (VTK in DielmoOpenLidar).
2. Namestitev programa v Windows operacijski sistem (OS): Izločil sem programe, ki ne omogočajo namestitve v Windows OS. Ta selektivni kriterij je pomemben iz dveh razlogov. Prvi razlog je namestitev potencialnega programa na Fakulteti za gradbeništvo in geodezijo Univerze v Ljubljani (FGG), kjer se na računalnikih v predavalnicah uporablja Windows 7 OS. Drugi razlog je navajenost študentov na uporabo Windows OS skozi celoten učni program na FGG. Posledično je bil iz seznama izločen program FullAnalyze.
3. Grafični vmesnik: Izločil sem programe, ki ne vsebujejo uporabniškega grafičnega vmesnika oz. GUI²⁹ in delujejo samo preko vmesnika z ukazno vrstico (angl. command-line). Obdelava podatkov v programu, ki ne vsebuje GUI za začetnega uporabnika ni prijazna (angl. user friendly). S tem selektivnim kriterijem sem izločil programe FWTools, Generic Mapping Tool in MCC-LIDAR.

²⁵ Program za kartiranje, ki je nameščen in deluje na osebem računalniku ter dovoljuje uporabniku prikazovanje, poizvedovanje, posodabljanje in analiziranje podatkov geografskih lokacij in informacij povezanih z njimi.

²⁶ Shareware ali preizkusni program je različica računalniškega programa, ki je uporabniku na voljo za preizkus pred odločitvijo za nakup. Ta program ima lahko v primerjavi s polno različico okrnjen nabor funkcij, ali pa je njegova uporaba časovno omejena in po preteku tega časa preneha delovati oz. deluje okrnjeno.

²⁷ Freeware ali brezplačno programje, za katerega uporabo nosilec materialne avtorske pravice ne zahteva nadomestila, lahko pa postavlja druge omejitve, npr. prepoved spreminjanja, prepoved porazdeljevanja.

²⁸ Programska knjižnica je v računalništvu zbirka podprogramov za pomoč pri izdelavi oz. razvoju programske opreme. Vsebuje kodo in podatke, ki se jih da uporabiti v neodvisnih programih. Nekatere izvršne datoteke npr. *.exe so lahko oboje; samostojni programi ali knjižnice, toda večina knjižnic ni izvršljivih.

²⁹ GUI oz. graphical user interface je grafični vmesnik v katerem komuniciranje med uporabnikom in računalnikom poteka s klikanjem in premikanjem gradnikov na zaslonu.

4. Subjektivni kriterij uporabnika: Kriterij se nanaša na intuitivnost programa in njegove enostavne namestitve v OS. Kompliciran postopek namestitve programa lahko uporabnika odvrne od nadaljnje uporabe programa, zlasti kadar se obenem sooči s slabo dokumentiranim opisom namestitve. Intuitivnost uporabe programa se nanaša na sposobnost uporabnika za neposredno dojetje operacij programa brez predznanja uporabe programa. Ko uporabnik prvič zažene program, lahko ugotovi, ali program omogoča takojšnjo in enostavno obdelavo podatkov. Pomembno je, da program uporabniku brez predznanja o uporabi programa v začetni fazi uporabe omogoča preizkus nekaterih enostavnih programskih operacij npr. uvoz podatkov. S tem kriterijem sem izločil program uDig, JGrass in OSSIM.

Obstaja tudi možnost, da sem s selektivnimi kriteriji izločil za raziskovalne namene dobre projekte, vendar, kot sem že omenil, sem se v nalogi osredotočal na neizkušenega uporabnika, začetnika.

Po izločitvi zgoraj omenjenih programov s pomočjo selektivnih kriterijev, sem dobil skrčen seznam programov, ki sem jih ocenil s kriteriji za ocenitev primernosti uporabe programov za potrebe diplomske naloge. V podrobno ocenitev sem tako vključil naslednje programe:

- | | |
|---------------|-----------------|
| 1. FUSION/LDV | 6. Quantum GIS |
| 2. GRASS GIS | 7. QuikGrid |
| 3. gvSIG | 8. SAGA GIS |
| 4. ILWIS | 9. Whitebox GAT |
| 5. MapWindow | |

5.2 Kriteriji za ocenitev primernosti

Ko predstavljamo različne projekte in njihove programe je pomembno, da ocenimo projekte s pomočjo enotnih kriterijev (Michlmayr, M., 2005), ki zagotavljajo objektivno oceno primernosti programa za namen uporabe. Ker je znanstvena literatura na temo FOSS GIS programja za obdelavo lidarskih podatkov kljub velikemu razmahu proste in odprtokodne industrije še vedno redkost, sem kriterije zaradi specifičnosti naloge definiral in združil iz različnih virov, ki se ukvarjajo z vrednotenjem FOSS GIS programov in FOSS programov na splošno. Za definiranje kriterijev (Preglednica 5-2) sem uporabil naslednje vire:

- CASCADOSS, 2007
- Chen, D. et al, 2008
- Michlmayr, M., 2005
- Ramsey, P., 2007

- Steiniger, S. in Bocher, E., 2008
- van den Berg, K., 2005
- Wheeler, D. A., 2010

Kot sem že omenil na začetku Poglavja 5 sem kriterije sestavil za pedagoške in raziskovalne namene. Preglednica 5-2 zajema vse uporabljene pedagoške kriterije in raziskovalne kriterije. Med raziskovalne kriterije nisem uvrstil kriterija Uporabniški nivo.

Preglednica 5-2: Seznam kriterijev

1	Zrelost projekta	5	Integracija
2	Dokumentacija	6	Uporabniški nivo
3	Skupnost in podpora	7	Funkcionalnost
4	Prepoznavnost		

Da sem lahko ugotovil v kakšni meri projekt zadovolji postavljenim kriterijem, sem znotraj vsakega kriterija določil vprašanja. Odgovore na vprašanja sem pridobil s pomočjo dokumentacije o projektu (FAQ, uporabniški priročnik, ipd.), z iskanjem po forumih, poštnih seznamih projektov, po svetovnem spletu ter s samo uporabo programov, kar je pomenilo namestitev programov na računalnik. Da sem lahko pridobil vse odgovore na postavljena vprašanja znotraj določenih kriterijev (npr. Funkcionalnost), sem programe tudi namestil na računalnik.

5.3 Opis kriterijev in ocena programov

5.3.1 Zrelost projekta

Zrelost projekta sporoča uporabniku za kako stabilen program se odloča in kakšne možnosti ima produkt za preživetje. Z zrelostjo projekta želimo oceniti na kakšni stopnji razvoja je projekt: koliko časa je že prisoten na tržišču, koliko časa je minilo od zadnje izdaje ipd. Ta kriterij je eden manj pomembnih kriterijev za ocenitev projekta, vendar pa ni zanemarljiv.

Starost projekta je časovna prisotnost produkta na tržišču. Projekt, ki se je ravnokar začel razvijati, vsebuje po vsej verjetnosti veliko napak in hroščev³⁰ (angl. bug). Po drugi strani pa za starejše projekte obstajajo možnosti za nenadno prekinitev delovanja. Čeprav se lahko projekt kot dokončana izdaja skozi čas stabilizira, ne bi smel noben projekt biti popolnoma statičen. Pomembno je, da je vzdrževan in da bo vzdrževan tudi v bodoče (Wheeler, D.A., 2010).

³⁰ Hrošč je napaka v računalniškem programu, ki je programer med svojim delom ni odkril.

Večina programskih projektov po določenem času izda novo verzijo programa s popravki zadnje verzije ali novimi dodatki oz. funkcijami. Število izdaj v določenem obdobju in njihova pomembnost kažejo na napredek, ki so ga razvijalci programa dosegli. Temu faktorju pravimo aktivnost izdaj. Pomembnost izdaje se kaže z velikostjo sprememb, saj gre lahko le za manjše popravke hroščev ali pa večje spremembe kot so dodatne oz. nove funkcionalnosti, ki jih program lahko izvaja. Spremembe so napisane v t.i. release notes. Pomembna lastnost tega kriterija je tako čas, ki je pretekel od zadnje izdane stabilne verzije programa. Ta faktor nam pove, ali je razvoj projekta še vedno aktiven in nam daje dober vpogled v resnost razvijalcev in sam projekt. Zadnja stabilna verzija naj ne bi bila starejša od dveh let [van den Berg, K., 2005].

FAQ je baza pogostih vprašanj. To je seznam, ki ga razvijalci programa načeloma oblikujejo na podlagi pomembnih in pogostih vprašanj njihovih uporabnikov. Dobra, uporabna in lahko dostopna baza pogostih vprašanj je tudi eden od pokazateljev pomembnosti projekta.

Kot sem že omenil je za raziskovalca pomembno, da si lahko program prilagodi in spremeni po svojih željah in potrebah. Tako sem v ocenitev programa za raziskovalni namen vključil tudi vprašanje o uporabi VCT³¹ orodja. Uporaba VCT-ja omogoča večjemu številu razvijalcev hkratno delo na izvorni kodi programa. Tudi lažja dostopnost do arhiva izvorne kode vzpodbuja iste razvijalce k pregledovanju kode, odstranjevanju napak in/ali dodajanju novih funkcionalnosti.

5.3.2 Dokumentacija

Dokumentacija je spremljajoč tekst programa in se ponavadi nahaja na spletni strani projekta ali pa je dodana k programskemu paketu. Z njo si lahko uporabnik pomaga pri namestitvi, uporabi in razvoju programa. Večji projekti imajo za dokumentacijo zadolženo posebno skupino ljudi, ki skrbi za njeno izdajanje in posodabljanje. Dokumentacija je pomembna za projekt, saj novi uporabnik ne bo znal uporabljati programa, če je le-ta slabo dokumentiran. Obstajata uporabniška in razvijalska dokumentacija. Za uporabnika – raziskovalca sta potrebni obe vrsti dokumentacije, medtem ko uporabnik – začetnik potrebuje le uporabniško dokumentacijo, saj se po vsej verjetnosti ne bo ukvarjal z razvojem programa.

5.3.2.1 Uporabniška dokumentacija

Uporabniška dokumentacija vsebuje dokumente, ki opisujejo, kako uporabljati program. Ponavadi vsebuje vsaj navodila za namestitev programa, saj je pri določenih programih le-ta specifična,

³¹ Version Control Tool je odlagališče datotek, ponavadi izvorne kode, z nadzorovanim dostopom, ki beleži vsako spremembo izvorne kode, skupaj z avtorjem spremembe, razlogom za spremembo in napotkom za odpravo problema.

predvsem pri dodajanju specifičnih modulov³², ki so potrebni za zahteve uporabnika. Potrebna dokumentacija za začetnega uporabnika:

- Uporabniški priročnik (angl. user guide) s splošnimi navodili za uporabo programa.
- Referenčni vodič (angl. reference guide), ki vsebuje opis orodij in funkcionalnosti programa.
- Priročnik za namestitev programa (angl. installation guide).
- Datoteke pomoči (angl. help files), ki uporabniku pomagajo pri delu z računalnikom.
- Vadnice (angl. tutorials), ki uporabniku pomagajo pri uporabi programa z navodili po korakih.

5.3.2.2 Razvijalska dokumentacija

Razvijalska dokumentacija ima v FOSS projektih pomembno vlogo. Dobra razvijalska dokumentacija vzpodbudi izkušene uporabnike, da se priključijo razvijalcem in prispevajo k razvoju programa. Potrebna dokumentacija za razvijalce:

- Dokumenti, ki opisujejo, kako postati udeležen pri razvijanju kode.
- Dokumenti, ki opisujejo, kako spreminjati in izpopolnjevati izvorno kodo.
- Datoteke z izvorno kodo in deli izvorne kode.
- Razvijalski FAQ in datoteke pomoči, spletna pomoč za razvijalce ipd.

Uspešen projekt mora zmanjšati težave, na katere lahko naletijo novi sodelavci, na minimum, da lahko poveča količino koristnih prizadevanj usmerjenih k projektu. Čas, ki se ga porabi za ugotavljanje nedokumentiranih detajlov oz. delov programa, je slabo porabljen čas, ki bi sicer moral biti namenjen in usmerjen produktivnemu razvoju kode (Ramsey, P., 2007).

5.3.3 Skupnost in podpora

5.3.3.1 Skupnost

Skupnost tvorijo uporabniki in razvijalci programa. Je gonilna sila projekta. Mišljena je kot tisti člen projekta, ki opravi veliko večino testiranj programa in zagotavlja kvaliteten odziv razvijalcem, ki so za uporabnike in razvoj programa ključnega pomena. V nasprotju z lastniškimi programi, kjer bi potrebovali finančne vire za testiranje, izkorišča FOSS projekt kot svoj vir svojo skupnost. Več kot je

³² Modul je skupek vsebinsko povezanih procedur, ki doda novo lastnost programskega sistema. Lahko tudi dodatek ali plug-in.

zainteresiranih ljudi za projekt, večja in aktivnejša je skupnost in posledično je tak program bolje sprejet. Skupnost zagotavlja pomoč uporabnikom vseh nivojev znanja uporabe programa (začetnik, napreden, izkušen, itd.) in je hkrati odgovorna za razvoj FOSS projekta.

Skupnost FOSS projekta sestavljajo ljudje, ki program uporabljajo ali pa na kakršenkoli način sodelujejo pri njegovem razvoju. To sodelovanje se lahko kaže v iskanju majnih napak kot so popravki teksta, iskanje in poročanje o programskih hroščih, prijava funkcionalnosti, ki ne delujejo pravilno, izrazu želje po določeni funkcionalnosti, ipd. Več kot je pristopov uporabnikov v skupnost, večji potencial ima skupnost za rast. Uporabnik z znanjem programskih jezikov, se lahko priključi razvijalski skupnosti. Za razvoj projekta je pomembno število sodelujočih razvijalcev. Razvijalec je lahko posameznik ali pa tudi različne organizacije in podjetja, kar daje skupnosti dodatno kredibilnost. Delovanje skupnosti je moč oceniti predvsem preko razprav na forumih in poštnih seznamih (Duijnhouwer, F. in Widdows, C., 2003; Crowston, K. et al., 2004; Nijdam, M., 2003; Golden, B., 2005, cit. po van den Berg, K., 2005).

Poštni sezname³³ in forumi predstavljajo učinkovit in hiter medij za komuniciranje in koordiniranje uporabnikov. Namenjeni so uporabnikom in razvijalcem, kjer si med seboj odgovarjajo na postavljena vprašanja. Poštni sezname se osredotočajo na določene teme (angl. topic). Obstajajo poštni sezname za uporabnike in razvijalce ter sezname za objave oz. obvestila. Zaželen je tudi arhiv poštnega seznama, saj lahko na ta način uporabniki in razvijalci dostopajo do razprav starejšega datuma, kjer lahko najdejo odgovore, ki so že bili podani na določena vprašanja.

Dober pokazatelj delovanja in živahnosti skupnosti je število objav in tem na forumih in poštnih seznamih v določenem časovnem obdobju. Viri navajajo, da je obdobje zadnjih šestih mesecev dovolj dolgo obdobje za oceno. V nalogi sem uporabil datum 15.5. 2011, tako da sem ocenjeval obdobje od 15.11.2010 do 15.5.2011.

5.3.3.2 Podpora

Podporo FOSS programom v večini primerov opravlja skupnost. Podpora skupnosti je neprecenljiv vir reševanja problemov. Dobra podpora je mogoča le ob dobrem sodelovanju razvijalske in uporabniške skupnosti (Golden, B., 2005, cit. po van den Berg, K., 2005).

Ločimo dva tipa podpore, ki jo zagotavlja skupnost:

³³ Razlika med poštnim seznamom in forumom je samo v formatu, saj oba vsebujeta enaka sporočila in informacije. Poštni seznam dostavi sporočilo v uporabnikov poštni predal, na forumu uporabnik dostopa do sporočil preko spleta.

1. Podpora uporabi (angl. usage support), kjer gre za pomoč končnemu uporabniku programa. V tem primeru skupnost pomaga uporabniku z odgovori na vprašanja, ki se navezujejo na samo uporabo programa.
2. Vzdrževanje oz. podpora ob napakah (angl. maintenance, failure support), s katero skupnost pomaga reševati programske probleme, predvsem odkrivanje in odpravljanje napak s pomočjo spletno dodanega sistema za sledenje napak (angl. bug tracker system³⁴).

S pomočjo stopnje razvitosti podpore določenemu programu, lahko ocenimo kako resno razvijalci opravljajo svoje delo na programu. Zraven podpore skupnosti obstaja tudi podpora tretje osebe (angl. third party support), ki jo uporabljajo bolj popularni projekti. Takšna podpora je za uporabnika plačljiva in se je pogosteje poslužujejo organizacije, podjetja in institucije.

5.3.4 Prepoznavnost

Prepoznavnost FOSS projekta je pomemben kriterij pri začetnem odločanju uporabnika o izbiri programa. Prepoznavnost projekta ocenjujemo s primerjanjem z drugimi projekti. Priporočeno je ugotoviti, kdo program uporablja (institucije, vladne agencije, podjetja, ipd.). Ponavadi se uspešni projekti na svoji spletni strani postavljajo s svojimi netipičnimi uporabniki. Prepoznavnost posledično vpliva tudi na interes skupnosti in privabljanje novih razvijalcev k projektu.

Prepoznavnost projekta lahko ocenimo s pomočjo:

- internetnih brskalnikov (Google, Yahoo, Najdi, ipd.), ki nam omogočajo hiter vpogled v priljubljenost posameznega projekta s številom zadetkov iskanja. Na tem mestu je potrebno opozoriti, da število zadetkov ni popolnoma objektivni pokazatelj, saj so lahko te informacije tudi zmanipulirane s strani "promotorjev" določenega programa.
- Wikipedije, kjer lahko izvemo osnovne informacije o projektu. Sicer Wikipedija v splošnem ne velja za relevanten vir informacij, toda pri pregledu in primerjavi domačih spletnih strani projekta s stranmi projektov na Wikipediji lahko trdim, da so pogosto informacije o projektu na Wikipediji bolj aktualne. To velja za seznam zadnjih stabilnih izdaj programov, vrste licence, ipd., po drugi strani pa so informacije o funkcionalnostih programa mnogokrat zastarele. V splošnem bo nov uporabnik dobil na straneh Wikipedije dovolj informacij, da bo vedel kako nadaljevati s svojo poizvedbo.
- specifičnih spletnih strani (Sourceforge.net, Freshmeat.net, ipd.), ki vodijo statistiko in informacije o projektu. Te strani, ki sem jih že omenil v Poglavju 5.1 so pomembne predvsem zaradi informacij, ki jih ponujajo o projektu kot so dokumentacija, izvorna koda, namestitvene

³⁴ Je programska aplikacija, ki pomaga razvijalcem slediti, s strani uporabnikov, prijavljenim napakam. Aplikacija omogoča uporabnikom prijavo napake neposredno.

datoteke programa, statistika priljubljenosti programa, ocene programov s strani uporabnikov, osnovne informacije o programu kot so licence, razvijalci, spletni naslovi do forumov in poštnih seznamov itd.

5.3.5 Integracija

Duijnhouwer in Widdows omenjata tri integracijske (povezovalne) podkriterije. Pomembni so pri programih, ki jih uporabnik uporablja v sodelovanju z drugimi programi ali pa namerava uporabnik (oseba, institucija, organizacija, ipd.) programu dodati nove funkcionalnosti, s katerimi program prilagodi potrebam svojega dela (Duijnhouwer, F. in Widdows, C., 2003, cit. po van den Berg, K. 2005).

5.3.5.1 Modularnost

Modularnost se navezuje na arhitekturo sestavljanja programa, kjer je ta sestavljen iz ločenih, medsebojno zamenljivih komponent t.j. modulov. S tem so programske funkcije razdeljene na module, kjer vsak modul izpolnjuje eno programsko funkcijo in vsebuje vse potrebno za njeno delovanje. Posledično je zaradi modularnosti programa lažje njegovo vzdrževanje, saj moduli predstavljajo logične meje med komponentami.

Takšna arhitektura ima določene prednosti (van den Berg, K., 2005):

- Lažje upravljanje s programom in dodajanje funkcionalnosti po meri brez spreminjanja jedra programa. Uporabnik, ki je razvil modul za določeno namen uporabe, ga lahko deli s skupnostjo, ki ga lahko prav tako uporablja.
- Modularna arhitektura programa omogoča izbiro za določeno uporabo potrebnih funkcionalnosti. Prav tako lahko uporabnik nepotrebne funkcionalnosti odstrani.
- Uporaba modulov FOSS programov v lastniških programih, kar dopuščajo non-copyleft licence.

Dodajanje že izdelanih modulov programa je uporabno za vse vrste uporabnikov. Zaželeno je, da obstaja arhiv modulov in da je metoda dodajanja modulov dobro opisana.

Ponavadi obstajajo za razvijanje modulov navodila in predloge³⁵ (angl. template), ki razvijalcu poenostavijo proces izdelave in prilagajanja programa njegovim potrebam. Te informacije in opis načina spreminjanja lahko najdemo v programski dokumentaciji (Wheeler, D.A., 2010).

³⁵ V naprej pripravljen dokument, ki ga dopolnimo z manjkajočimi podatki.

5.3.5.2 Standardi

Na tržišču se pojavlja vse več odprtih standardov, ki omogočajo lažje sodelovanje med različnimi programi. Programi, ki omogočajo uporabo teh standardov, bolje komunicirajo z ostalimi programskimi paketi. Uporaba trenutnih in odprtih standardov, ki jih uporablja FOSS program, je znak zrelosti programa (Duijnhouwer, F. in Widdows, C., 2003, cit. po van den Berg, K. 2005).

5.3.5.3 Medopravilnost

Medopravilnost pospešuje, omogoča in podpira delitev obdelave podatkov med različnimi programi, torej gre za medsebojno sodelovanje. Je tudi zmožnost komuniciranja, izvajanja programov in prenosa podatkov med različnimi funkcionalnimi enotami znotraj posameznega programa. Je tesno povezana s standardi.

5.3.5.4 Programske in strojne zahteve

Večina programov je napisana in izdelana samo za določene operacijske sisteme (OS), kar je lahko pri izbiri programa ključen faktor. Programske zahteve lahko navadno najdemo med dokumentacijo v namestitvenih dokumentih. Specifične programske zahteve FOSS programov lahko privedejo do problemov, kadar so nekompatibilne z OS, ki ga uporablja uporabnik. V primeru, ko se programske zmožnosti programa, ne ujemajo s programskimi zahtevami OS, se je potrebno odločiti za nadaljnji korak. Ali druga izbira programa ali zamenjava OS. Obe možnosti imata svoje slabosti. Slabost zamenjave OS je porabljen čas, ki ga uporabnik potrebuje, da se navadi in priuči uporabljati nov OS. Slabost zamenjave programa je lahko manjše število specifičnih funkcionalnosti novega programa v primerjavi s prvotno želenim. Kot prednost obeh primerov bi lahko izpostavil "širjenje obzorij", vendar je pri tem pomemben čas, ki ga ima uporabnik na voljo.

Ker je cilj diplomske naloge pregled programov za pedagoške namene na FGG, kjer so računalniki opremljeni z Windows OS, sem se odločil, da je prednost ocenjevanega programa nemoteno delovanje na omenjenem OS. Za raziskovalne namene predvidevam, da se uporabnik - raziskovalec, bolj posveti izbiri FOSS programa, ki mora ustrezati zahtevam njegovega dela. Hkrati se je iz istega razloga pripravljen naučiti uporabe drugih OS in programov. Problem manjšega števila funkcionalnosti programa lahko odpravi z dodajanjem in spreminjanjem modulov ter samega programa. Za to, pa potrebuje znanje s področja programiranja v specifičnem programskem jeziku.

Težave se lahko pojavijo tudi v odnosu strojne opreme in zahtevami programa. Te zahteve so ponavadi opisane v programski dokumentaciji. Težave so podobne kot v primeru programskih zahtev,

toda računalniki na FGG delujejo v Windows OS in so hkrati dovolj zmogljivi. Na osnovi OS računalnikov na FGG sem določil selektivni kriterij (Poglavje 5.1).

5.3.6 Uporabniški nivo

Kriterij opisuje primernost programa glede na uporabnikovo znanje za rokovanje z njim. Uporabnike lahko razdelimo v tri skupine. Raziskovalec, izkušen uporabnik in neizkušen uporabnik v smislu uporabe programov.

- Za raziskovalca lahko predvidevamo, da ima znanje potrebno za uporabo GIS programov in hkrati voljo po dodatnem izobraževanju na tem področju.
- Izkušenemu uporabniku GIS programi niso tuji.
- Neizkušen uporabnik – začetnik ni vajen pogoste uporabe GIS programov, a vendar mu GIS področje ni tuje.

S tem kriterijem določamo za kakšnega uporabnika je program primeren. Te informacije sem pridobil s spletnih forumov in poštinih seznamov kot tudi z lastno subjektivno izkušnjo. Kriterij je tesno povezan s kriterijem dokumentacije, saj je pomembno katero dokumentacijo projekt vsebuje in je v pomoč uporabniku, še posebej začetniku. Na vsa ta vprašanja lahko odgovori samo posamezen program, ki ga uporabnik želi preizkusiti oz. uporabiti za svoje delo. Kot pravi Wheeler (2010), je program z visoko stopnjo uporabnosti, enostaven za učenje in uporabo. Takšen program želim izpostaviti v tej nalogi.

5.3.7 Funkcionalnost

Za izpolnitev cilja diplomske naloge je ta kriterij najpomembnejši med vsemi. Program, ki ne omogoča določenih funkcionalnosti za obdelavo oblaka lidarskih točk in izdelavo modelov površja, ni primeren za uporabo na FGG in ga ni smiselno izpostavljati.

Seznami funkcionalnosti so ponavadi zbrani na spletnih straneh projektov. Problem se pojavi zaradi načina, ki ga uporabljajo odprtokodni projekti t.j. "Release early and often" (Raymond, E. S., 2002). Ta metoda sicer omogoča hitrejše odpravljanje napak, predvsem zaradi konstantnega in pogostega posodabljanja programov, vendar pa pogostokrat dokumentacija ne zmore slediti temu tempu posodabljanja. Pogostokrat se izkaže, da ponudnik programa na spletnih straneh projekta nima objavljenega novega ali celotnega seznama funkcionalnosti. Ker je zaradi pomanjkljivih informacij težko ugotoviti katere funkcionalnosti premore verzija programa, je program najbolje preizkusiti, toda ta postopek zahteva ogromno časa. Zaradi omenjenega problema sem poizkušal poiskati čim več

informacij s pomočjo skupnosti. Kljub vsemu sem preizkusil večino programov, saj nisem uspel zbrati zadovoljivo količino informacij glede funkcionalnosti programov.

Program mora za izdelavo DMR-ja vsebovati funkcionalnosti, ki omogočajo:

- uvoz podatkov v LAS in/ali ASCII (xyz,txt,asc) formatu
- filtriranje, s katerim v oblaku točk ločimo točke na terenske in ne-terenske
- interpolacijo točk
- izdelavo mrež (npr. TIN)
- izvoz podatkov v formatu, ki jih podpirajo ostali GIS programi (*.grd, *.shp, *.asc, ipd.)
- grafičen prikaz oblaka točk, saj nam ta prikaz služi kot skica oz. nam omogoča ogled območja obdelave

V zgornji seznam nisem vključil klasifikacije točk v razrede, saj sem za potrebe naloge dobil že klasificirane podatke. Tako smatram, da bi tudi študentje FGG prejeli v obdelavo že klasificirane podatke. Če bi hoteli točke klasificirati, bi si lahko pomagal s programom MCC-LIDAR, ki pa sem ga izključil že med selekcijo. Program klasificira točke na terenske in ne-terenske.

5.4 Vrednotenje kriterijev

Vsak kriterij vsebuje več vprašanj, na katera sem sposoben odgovoriti čim bolj objektivno in ne z osebno, subjektivno oceno. Vprašanja sem oblikoval tako, da sem lahko v večini primerov nanje odgovoril z "da" ali "ne" oziroma z "več" ali "manj kot". Vsakemu vprašanju ustreza samo en odgovor, ki je ovrednoten s točkami od 0 do 2. Odgovor z večjo vrednostjo bolje izpolnjuje zahteve kriterija, prinese višjo oceno projekta, in obratno, odgovor z manjšo vrednostjo slabše oz. ne izpolnjuje zahtev kriterija. Vsak kriterij ima različno število možnih točk kot prikazuje Preglednica 5-3.

Preglednica 5-3: Število možnih točk kriterijev ocenitve

Kriterij	Pedagoški namen	Raziskovalni namen
Zrelost projekta	4	5
Dokumentacija	5	9
Skupnost in podpora	10	15
Prepoznavnost	4	4

Integracija	4	9
Funkcionalnost	7	7
Uporabniški nivo	4	/

Razmerje med dodeljenimi točkami in vsemi možnimi točkami kriterija predstavlja izpolnitev posameznega kriterija projekta. Vsakemu kriteriju je dodeljena utež, glede na njegovo pomembnost za posamezen namen (pedagoški in raziskovalni) kot prikazuje Preglednica 5-4.

Preglednica 5-4: Uteži kriterijev

Kriterij	Pedagoški namen	Raziskovalni namen
Zrelost projekta	5	10
Dokumentacija	25	20
Skupnost in podpora	10	20
Prepoznavnost	5	5
Integracija	10	15
Funkcionalnost	30	30
Uporabniški nivo	15	/

Uteži kriterijev za pedagoški namen se razlikujejo od uteži kriterijev za raziskovalni namen, saj so posamezni kriteriji za specifični namen različno pomembni. Določil sem jih na podlagi pomembnosti kriterija za doseg cilja te naloge – najti čim bolj primeren program za študente FGG (pedagoški namen). Očitno je, da na primer zrelost projekta ni primerljiva s funkcionalnostjo programa in zato ne moreta doprinesiti h končni oceni enakega števila točk (odstotkov). Uteži bi lahko bile določene tudi drugače, bolj objektivno; potrebno pa je nekako ločiti pomembnost kriterijev med seboj. Oceno primernosti projekta za omenjena namena sem izračunal po enačbi (1).

$$O = \sum_{i=1}^n \frac{p_i}{m_i} \cdot U_i \quad (1)$$

kjer so:

i ... kriterij

n ... število kriterijev

p_i ... prejeta število točk i-tega kriterija

m_i ... možno število točk i-tega kriterija

U_i ... utež i-tega kriterija

O ... ocena primernosti projekta

Kot sem že omenil, sem uteži določal glede na pomembnost posameznega kriterija kot tudi na zahteve določenega namena (pedagoški, raziskovalni). Zanašal sem se na občutek, ki sem ga dobil s prebiranjem v Poglavju 5.2 omenjene literature. Ta metoda določitve uteži in posledično ocene primernosti programa za potrebe naloge je subjektivna, na podlagi avtorjevih izkušenj, pridobljenih med študijem literature. Sicer obstajajo ocenitvene metode, s katerimi lahko dosežemo bolj objektivne rezultate (npr. Goal Question Metric), vendar podrobnejša obravnava tega problema presega meje te naloge.

5.5 Rezultati ocenitve programov

Točkovno ovrednotenje in ocene programov po posameznih kriterijih so podane v poglavju Priloge.

Preglednica 5-5 prikazuje končno oceno programov za obdelavo lidarskih podatkov, ocenjenih za pedagoški namen. Iz preglednice je razvidno, da je najbolje ocenjen program GRASS GIS.

Preglednica 5-5: Ocena FOSS programov za obdelavo lidarskih podatkov - pedagoški namen

	Program	Ocena		Program	Ocena
1	GRASS GIS	100,00	6	Whitebox GAT	72,96
2	SAGA GIS	94,00	7	ILWIS	69,07
3	MapWindow	84,46	8	gvSIG	68,64
4	FUSION/LDV	82,75	9	QuikGrid	54,86
5	Quantum GIS	76,14			

Preglednica 5-6 prikazuje končno oceno programov za obdelavo lidarskih podatkov, ocenjenih za raziskovalni namen. Iz preglednice je razvidno, da je tudi za raziskovalni namen najbolje ocenjen program GRASS GIS.

Preglednica 5-6: Ocena FOSS programov za obdelavo lidarskih podatkov – raziskovalni namen.

	Program	Ocena		Program	Ocena
1	GRASS GIS	100,00	6	FUSION/LDV	69,39

2	SAGA GIS	95,11	7	Whitebox GAT	65,88
3	MapWindow	83,02	8	ILWIS	59,29
4	Quantum GIS	81,37	9	QuikGrid	43,33
5	gvSIG	75,48			

5.6 Izbira primerne kandidata

S pomočjo ocenitve programov in subjektivne ocene sem za vsak namen izbral en program. Za pedagoški namen sem izbral program FUSION/LDV, za raziskovalni pa GIS program SAGA.

Program FUSION/LDV, ki po ocenitvi s kriteriji (glej Preglednica 5-5) zaseda »šele« 4. mesto sem izbral, ker omogoča uporabniku (študentu) bolj neposreden vpogled v samo obdelavo oblaka lidarskih točk. Po mojem mnenju tako uporabnik dobi bolj poglobljen vpogled v sam postopek obdelave podatkov in izdelave DMR-ja.

Program ni primerljiv s SAGA-o in GRASS-om, saj sta oba zelo zmogljiva GIS sistema. Če ga po ocenitvi in točkovnem vrednotenju primerjamo s programom MapWindow, ki je zasedel mesto višje, lahko opazimo, da je slednji na tem mestu predvsem zaradi več doseženih točk po kriteriju Skupnost in podpora. Projekt FUSION/LDV nima skupnosti, kar je slabost FOSS programa, vendar pa je zelo dobro dokumentiran. Dobra dokumentacija omogoča uporabniku takojšnjo uporabo in spoznavanje programa.

V FOSS GIS svetu sta SAGA in GRASS primerljiva programa z majhnimi razlikami. SAGA je v primerjavi z GRASS-om za odtenek bolj uporabniku prijazen program. Kot namiguje že samo ime programa (System for Automated Geo-Scientific Analysis) ima SAGA znanstveno ozadje. Zanj sem se odločil predvsem zaradi dobrega uporabniškega grafičnega vmesnika, same intuitivnosti programa in zmogljivosti. Ker sem hkrati želel izbrati program, ki je primeren tako za raziskovalce kot tudi za uporabnika – začetnika, sem se odločil zanj.

Po ocenitvi s kriteriji in točkovnem vrednotenju ima GRASS 100% oceno, kar je dokaj nenavadno. Razlog je verjetno premajhno število ocenitvenih kriterijev in vprašanj, ki bi omogočili še bolj podrobno analizo. V tem primeru je pomembno uporabnikovo subjektivno mnenje, ki si ga ustvari po namestitvi in uporabi programa. Pri odločitvi za določen program lahko upoštevamo ocene in mnenja drugih uporabnikov ter ocenitve programov s kriteriji, toda priporočeno je program tudi preizkusiti, sploh pri tako majhnih razlikah v ocenah, saj se bomo le tako lahko dokončno odločili za pravi program.

5.7 Opisi programov

V tem poglavju sledita opisa programov, ki sem jih izbral za obdelavo podatkov. Prav tako sem opisal tudi preostale ocenjene programe, ter nekaj, po mojem mnenju zanimivih programov, ki sem jih izključil iz seznama ocenitve s selektivnimi kriteriji.

5.7.1 Opis izbranih programov

5.7.1.1 SAGA GIS

Licenca: GNU GPL

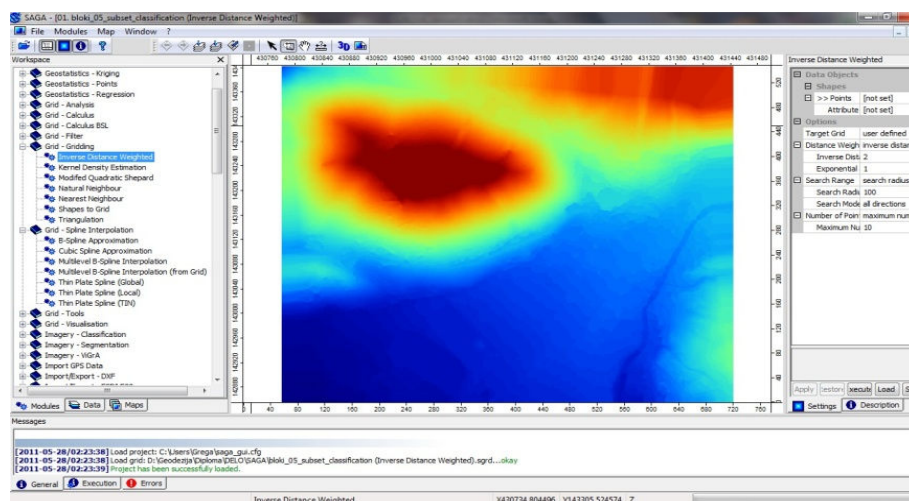
Programski jezik: C++

Platforma: Microsoft Windows, Linux, FreeBSD

Zadnja stabilna verzija: 2.0.7, **datum:** 17.5.2011

Spletna stran: <http://www.saga-gis.org/>

SAGA oz. System for Automated Geo-Scientific Analysis je geografski informacijski sistem, ki podpira vektorske in rastrske podatkovne formate. SAGA-o razvijajo že od leta 2001 na Univerzi v Göttingenu, Nemčija. Leta 2004 je bila večina izvorne kode programa izdana pod odprtokodno licenco kot verzija 1.1. Leta 2007 se je jedro razvijalske ekipe in s tem posledično tudi celoten projekt preselil v Hamburg, kjer sta na Oddelku za fizikalno geografijo Geografskega inštituta, Univerze v Hamburgu zaposlena J. Böhner in O. Conrad. Böhner in Conrad sestavljata jedro razvijalske ekipe, ki je odgovorna za vse prejšnje in zdajšnje različice programa. Ker je SAGA odprtokodni program, je mednarodna skupnost razvijalcev in vzdrževalcev programa vsak dan večja.



Slika 5-1: Prikaz uporabe programa SAGA.

SAGA je dobra izbira za uporabnika, ki se še ni srečal z GIS področjem kot tudi za uporabnika, ki želi izvajati zahtevne GIS analize. Vsebuje veliko zbirko geoznanstvenih algoritmov in je predvsem močno orodje za analize digitalnih modelov višin (DMV). Za uporabnika, ki že pozna osnovni koncept GIS analiz, SAGA ni le mogočno orodje, ampak tudi program, ob uporabi katerega bo užival. Kljub temu, da je program enostaven za uporabo, pa je SAGA zelo močno orodje za rastrske analize. Prav tako omogoča delo z vektorskimi podatki. Podpira večino popularnih in dobro poznanih podatkovnih formatov obeh podatkovnih tipov, kar omogoča uporabniku uporabo podatkov različnih virov. Za uvoz in izvoz podatkovnih formatov je na razpolago več modulov. Najbolj uporaben modul je Import/Export - GDAL/OGR, ki omogoča uvoz in izvoz znanih formatov knjižnic GDAL³⁶ in OGR³⁷. Z modulom Import/Export – LAS lahko uvozimo in izvozimo tudi LAS podatkovni format.

Je program, ki združuje mogočna orodja za geografske analize in intuitiven ter uporabniku prijazen GUI. Prav tako lahko uporabnik razvije svoje module in jih enostavno implementira v program. Slabost SAGA-e je ta, da ni znana širši GIS populaciji, kar pa je predvsem krivda pomanjkljive in slabe dokumentacije.

5.7.1.2 FUSION/LDV

Licenca: BSD

Programski jezik: C++

Platforma: Microsoft Windows,

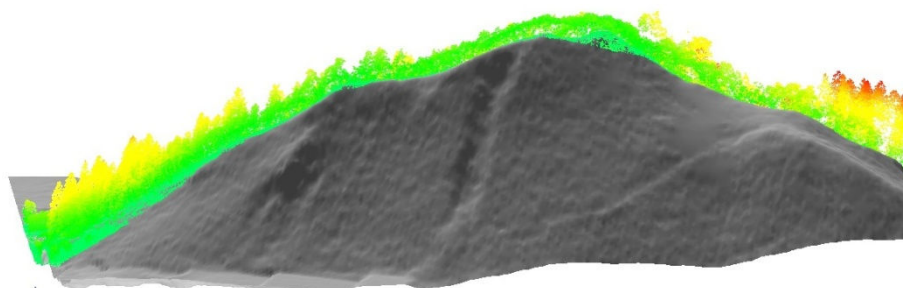
Zadnja stabilna verzija: 2.90, **datum:** 7.7.2010

³⁶ GDAL je knjižnica rastrskih podatkovnih formatov, glej http://www.gdal.org/formats_list.html.

³⁷ OGR je knjižnica vektorskih podatkovnih formatov, glej http://www.gdal.org/ogr/ogr_formats.html.

Spletna stran: <http://forsys.cfr.washington.edu/fusion/fusionlatest.html>

FUSION/LDV je programski paket za obdelavo in vizualizacijo oblaka lidarskih točk. Sestavljen je iz dveh programov (FUSION in LDV). FUSION vsebuje okno za 2D pregledovanje in prikazovanje različnih slojev podatkov in kontrolno okno. Podpira različne podatkovne tipe in formate. LDV oz. Lidar Data Viewer omogoča 3D okolje za vizualizacijo, ki temelji na OpenGL³⁸ knjižnici za pregledovanje prostorskih podatkovnih nizov. FUSION/LDV omogoča tudi možnosti specifičnih analiz in procesiranja lidarskih podatkov s pomočjo CMD programov ter BATCH³⁹ programiranjem.



Slika 5-2: 3D prikaz ploskve reliefa in oblak lidarskih točk

Program je bil razvit predvsem z namenom, da pomaga raziskovalcem razumeti, raziskovati in analizirati lidarske podatke. Razvili so ga znanstveniki na Pacific Northwest Research Station v sodelovanju z Univerzo iz Washingtona, ZDA pod okriljem Roberta J. McGaugheya.

Za razliko od komercialnih GIS programov lahko v FUSION/LDV programu brez predhodnega procesiranja enostavno pregledujemo večje zbirke lidarskih podatkov. Uporabniku omogoča hitro in enostavno izbiro ter prikaz podatkovnih podnizov lidarskih točk. Tem neobdelanim točkam lahko določi velikost, obliko in barvo ter izbere vzorec za grafični prikaz.

Uporabljeni algoritmi v programskih orodjih paketa FUSION omogočajo obdelavo in analizo lidarskih podatkov predvsem na pogozdenih območjih (npr. merjenje višin dreves, izdelava DMR).

Program podpira LAS in več ASCII formatov. Uporablja interni format *.LDA, ki je indeksiran binarni format in omogoča hiter dostop do podatkovnih nizov. LAS podatkovnih datotek za razliko od ASCII ni potrebno spreminjati v interni format.

³⁸ OpenGL (Open Graphics Library) je specifikacija standarda, ki določa programski vmesnik za pisanje programov, ki prikazujejo 3D in 2D računalniško grafiko.

³⁹ Batch datoteka (*.bat) je tekstovna datoteka, ki vsebuje zbirko ukazov. Ponavadi batch datoteke zaženemo s pomočjo programa cmd.exe, ki batch datoteko prebere in izvede njene ukaze.

Program za obdelavo podatkov zahteva predhodno naloženo georeferencirano podobo in world datoteko (*.tfw), ki vsebuje lokacijo, merilo in rotacijo TIFF (*.tif) podobe. Če te podobe in world datoteke nimamo, jo lahko v programu izdelamo sami, s pomočjo intenzitete lidarskih podatkov.

Uvoz podatkov: Kot sem že omenil lahko v program uvozimo LAS in ASCII format. FUSION bere modele površja (DMR in ostale) v PLANS DTM formatu (*.dtm) in zagotavlja orodja za spreminjanje in shranjevanje modelov površja ostalih formatov v PLANS format. Podprti formati za uvoz so USGS ASCII DEM, USGS SDTS, SURFER in ASCII Grid.

Izvoz podatkov: Program vključuje orodja za spreminjanje podatkov iz PLANS DTM formata (*.dtm) v druge formate kot so ASCII Raster format (*.asc), ENVI Standard format (*.nvi) za uporabo v programih ENVI in ArcInfo, TIFF (*.tif, *.tfw) in ASCII Text file (*.xyz, *.csv). Te formate lahko uporabljamo za nadaljnjo obdelavo v raznih GIS programih. Nekateri od njih so opisani v naslednjih poglavjih.

5.7.2 Opis ocenjevanih programov

5.7.2.1 GRASS GIS

Licenca: GNU GPL

Programski jezik: C, C++, Python, Tcl

Platforma: Microsoft Windows, Linux, Mac OS X, IBM AIX, BSD-Unix, itd.

Zadnja stabilna verzija: 6.4.1, **datum:** 12.4.2011

Spletna stran: <http://grass.osgeo.org/>

GRASS ali Geographic Resources Analysis Support System je rastrsko/vektorski geografski informacijski sistem za obdelavo in analize prostorskih podatkov, obdelavo podob, izdelavo kart, prostorsko modeliranje in vizualizacijo. Uporabljajo ga različni raziskovalni inštituti, univerze in podjetja, kot tudi vladne agencije. Je uraden projekt Open Source Geospatial Foundation⁴⁰ in je eden izmed najbolj celovitih, večnamenskih in zmogljivih orodij GIS, saj vsebuje več kot 350 različnih modulov in orodij, ki služijo za uporabo vektorskih, rastrskih in atributnih podatkov.

Razvijati so ga začeli za potrebe ameriške vojske v zgodnjih 80. letih 20. stoletja (GRASS, 2011). Danes pri njegovem razvoju sodelujejo razvijalci iz celega sveta, kar omogoča sistem za nadzor

⁴⁰ Open Source Geospatial Foundation (OSGeo) je neodvisna neprofitna organizacija, katere cilj je spodbujati in promovirati razvoj odprtokodnih tehnologij GIS in prostorskih podatkov (Open Source Geospatial Foundation, 2011)

izvirne kode (VCT), vendar pa še vedno nadzor in glavni razvoj opravlja razvojna ekipa GRASS-a raziskovalnega centra ITC⁴¹ (danes FBK⁴²).

Podpira velik nabor rastrskih in vektorskih formatov s pomočjo GDAL/OGR knjižnic. Te knjižnice podpirajo uvoz in izvoz 2D in 3D rastrskih ter topološko urejenih vektorskih podatkov.

GDAL knjižnico uporablja za manipulacijo rastrskih formatov kot so **ASCII**, **ARC/GRID**, E00, GIF, GMT, **TIF**, PNG, ERDAS LAN, SURFER (.grd), CEOS (SAR, SRTM, LANDSAT7 itd.), ERDAS LAN, HDF, LANDSAT TM/MSS, NHAP zračne podobe, SAR, SPOT, itd.

OGR knjižnico uporablja za manipulacijo vektorskih formatov kot so **ASCII**, ARC/INFO, ARC/INFO E00, ArcView SHAPE, BIL, DLG (U.S.), DXF, DXF3D, GMT, GPS-ASCII, **USGS-DEM**, IDRISI, MOSS, MapInfo MIF, TIGER, VRML, **XYZ ASCII**, **CSV**, dBase, itd.

Za potrebe diplomske naloge so predvsem uporabni formati v odebeljeni pisavi, s pomočjo katerih lahko prenašamo podatke iz enega programa v drug program. Za obdelavo lidarskih podatkov lahko v program neposredno uvozimo podatke v tekstovnem ASCII formatu. Za uvoz LAS formata v GRASS, pa je potrebno uporabiti knjižnico libLAS.

Na spletni strani <http://grass.osgeo.org/wiki/LIDAR> je zbranih še več informacij o samem uvozu in izvozu ter obdelavi in vizualizaciji lidarskih podatkov v programu GRASS. Za uporabnike, ki si želijo spoznati GRASS še bolj podrobno, pa priporočam knjigo avtorjev M. Neteler in H. Mitasova z naslovom OPEN SOURCE GIS, A GRASS GIS Approach, Third edition.

5.7.2.2 gvSIG

Licenca: GNU GPL

Programski jezik: Java

Platforma: Microsoft Windows, Linux, Mac OS X

Zadnja stabilna verzija: 1.11, **datum:** 18.4.2011

Spletna stran: <http://www.gvsig.org/>

Generalitat Valenciana, Sistema d'Informacio Geografica ali gvSIG je namizni GIS program. Je projekt, ki ga podpira OSGeo. Namenjen je urejanju, shranjevanju, analiziranju in porazdeljevanju različnih referenciranih geografskih informacij. Vsebuje uporabniku prijazen vmesnik. Uporabnik

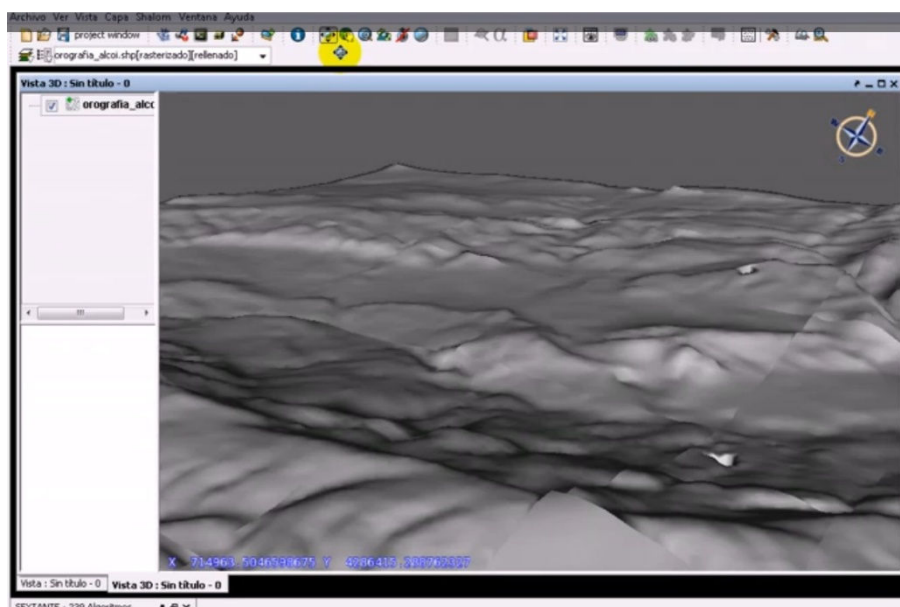
⁴¹ Centro per la Ricerca Scientifica e Tecnologica, Trento, Italija, ki je do danes podpirala razvoj več kot 60 modulov.

⁴² Raziskovalna organizacija avtonomne province Trento, ki promovira raziskave na področjih znanosti, tehnologije in humanistike.

lahko uporablja različne vektorske in rastrske formate. Program vsebuje širok nabor orodij za obdelavo in uporabo geografskih podatkov (OSGeo, 2011).

Razvit je bil za regionalni zbor za infrastrukturo in transport (CIT, Valencia, Španija), s ciljem zagotoviti program s funkcionalnostmi podobnimi ESRI-jevem ArcView GIS programu. Začetki razvoja programske kode, ki ga vodi podjetje IVER S.A. (Španija) segajo v leto 2003. V razvoj so vključene številne univerze in podjetja.

Kot sem omenil program podpira uporabo rastrskih in vektorskih podatkov. Izvorna koda rastrskih funkcionalnosti je razvita na osnovi algoritmov SAGA-e. Čeprav razvojna ekipa programa izhaja iz Španije, je uporabniška skupnost mednarodna. Kot slabost programa lahko omenim pomanjkanje dokumentacije za razvijalce.



Slika 5-3: 3D pogled DMR-ja, izdelanega iz *.shp datoteke podatkov

Podpira večino rastrskih in vektorskih formatov ter formatov podatkovnih baz (PostGIS, MySQL, Oracle, itd.). Prav tako podpira razširjene OGC standarde (WMS, WFS, itd.). Popoln seznam podprtih formatov je dostopen na spletni strani projekta, v nadaljevanju pa je naštetih le nekaj.

Vektorski formati: *.shp, *.gml, *.kml, *.dxf, *.dwg, itd.

Rastrski formati: *.asc, *.rst, *.tiff, *.png, *.lan, *.img, itd.

GvSIG lahko razširimo z dodatkom DielmoOpenLidar, ki omogoča upravljanje z lidarskimi podatki. DielmoOpenLidar je prav tako FOSS program in je licenciran z GNU GPL licenco. Omogoča dostop, vizualizacijo in analize lidarskih podatkov, toda trenutno še ne omogoča njihove obdelave. Tako lahko uporabnik le pregleduje neobdelane lidarske podatke skupaj z drugimi geografskimi podatki.

Razvijalci DielmoOpenLidar-ja obljublajo, da bo v prihodnosti mogoča tudi obdelava lidarskih podatkov in izdelava osnovnih produktov lidarske tehnologije kot je DMR.

5.7.2.3 ILWIS

Licenca: GPL

Programski jezik: Java

Platforma: Microsoft Windows

Zadnja stabilna verzija: 3.7.1, **datum:** 15.9.2010

Spletna stran: <http://www.ilwis.org/>

ILWIS oz. Integrated Land and Water Information System je program za GIS in daljinsko zaznavanje. Funkcionalnosti omogočajo digitalizacijo, urejanje, analiziranje, prikazovanje podatkov in izdelavo kvalitetnih kart. Program razvijajo na ITC Enschede⁴³ na Nizozemskem za potrebe raziskav in študentov. Čeprav ga razvijalci razvijajo že več kot dvajset let, so program licencirali kot prost program šele leta 2007. Program je v določenih pogledih podoben GRASS-u, vendar veliko bolj uporabniku prijazen. Slabost programa je ta, da ni sposoben procesiranja velikih količin lidarskih podatkov (ASCII). Podpira vektorske in rastrske formate s pomočjo uporabe GDAL knjižnice.

Rastrski formati: Arc/Info ASCII (*.asc, *.nas), Arc/Info .E00 (*.E00), GIF podobe (*.gif), Erdas .GIS in .LAN (*.gis, *.lan), IDA (*.img), Idrisi maps (*.doc), ILWIS ASCII raster (*.asc), Tagged Image File Format (*.tif), GeoTIFF (*.tif), itd.

Vektorski formati: Arc/Info .E00 (*.E00), Arc/View Shape (*.shp, *.shx, *.dbf), Atlas segments (*.bna), AutoCad (*.dxf), GARtrip Text (*.txt), ILWIS ASCII segments (*.smt), itd.

Med vsemi funkcionalnostmi, ki jih premore, omogoča tudi zbirko orodij za hidrološke analize na DMR-jih in DMV-jih, 3D vizualizacije, geostatistične analize (npr. kriging), interpolacije, izdelava in vizualizacija stereoparov, itd.

5.7.2.4 MapWindow

Licenca: MPL

Programski jezik: Microsoft .NET

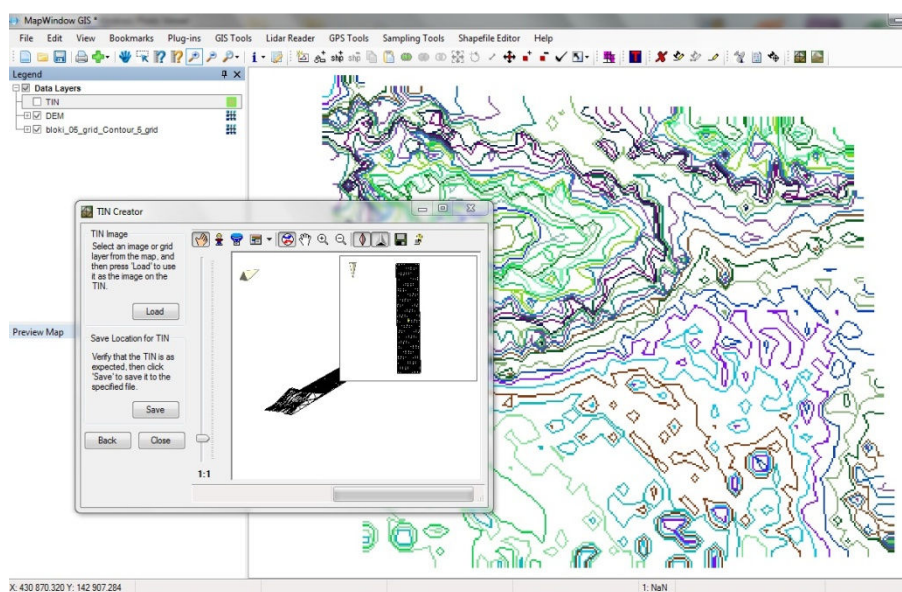
Platforma: Microsoft Windows

⁴³ International Institute for Geo-Information Science and Earth Observation.

Zadnja stabilna verzija: 4.7, **datum:** 28.11.2009

Spletna stran: <http://www.mapwindow.org/>

MapWindow je namizni GIS program, ki omogoča manipulacijo, analize in pregledovanje prostorskih podatkov in pripadajočih atributnih podatkov, s pomočjo mnogih standardnih podatkovnih formatov. Ta GIS projekt vsebuje namizno GIS aplikacijo s prožno arhitekturo dodatkov. Ker je odprtokodni program, katerega izvorna koda je dostopna na spletni strani projekta, omogoča spreminjanje programa za uporabnikove specifične naloge (MapWindow, 2011).



Slika 5-4: Primer izdelave DMR iz *.shp datoteke podatkov

Na spletni strani se nahaja tudi arhiv dodatkov, ki se jih preprosto dodaja programu. Program je zelo intuitiven in enostaven za začetnega uporabnika. Omogoča obdelavo rastrskih in vektorskih podatkov kot tudi obdelavo lidarskih podatkov. Omogoča uvoz LAS formata. Za obdelavo lidarskih podatkov je potrebno v program vključiti določene dodatke npr. Lidar Reader, ki se nahajajo v arhivu dodatkov na spletni strani projekta. Uporablja pomožne knjižnice GDAL, GEOS, PROJ4, itd.

Program omogoča uvoz in izvoz različnih rastrskih in vektorskih formatov kot so ASCII Text (*.asc, *.arc), ESRI Shapefile (*.shp), ASPRS (*.las), USGS ASCII DEM (*.dem), ASCII Grid images (*.asc), GeoTIFF (*.tif), itd.

5.7.2.5 Quantum GIS

Licenca: GPL

Programski jezik: C++

Platforma: Microsoft Windows, Linux, Unix, Mac OS X

Zadnja stabilna verzija: 1.6.0, **datum:** 27.11.2010

Spletna stran: <http://www.qgis.org/>

Quantum GIS (QGIS) je uporabniku prijazen odprtokodni namizni GIS program. Kot odgovor na znanstveno orientiran GRASS je skupina prostovoljcev leta 2002 začela razvijati QGIS. Namen razvoja QGIS-a je bil zagotoviti geografski pregledovalnik, ki bo enostaven za uporabo. Medtem, ko se je QGIS projekt razvijal, je na dan priplavala ideja, da bi se QGIS uporabil kot enostaven uporabniški grafični vmesnik za GRASS. V povezavi z GRASS-om je omogočena obdelava lidarskih podatkov (Terrain-Profile Plugin). Ko so razvijalci programa dosegli prvotni namen, so pričeli z razvojem funkcionalnosti, ki bodo omogočale več kot le pregledovanje podatkov. Tako kot GRASS je tudi QGIS uradni projekt OSGeo.

Omogoča uporabo, editiranje, analiziranje podatkov ter izdelave kart. S QGIS-om lahko ustvarjamo različne vektorske in rastrske formate. Vektorski formati so podprti s knjižnico OGR, rastrski formati pa s knjižnico GDAL. Knjižnici nista integrirani v sam program in ju je potrebno dodatno namestiti. Podpira tudi OGC standarde npr. WMS in WFS.

5.7.2.6 QuickGrid

Licenca: GPL

Programski jezik: C++

Platforma: Microsoft Windows, ostali OS z uporabo Windows emulatorja⁴⁴

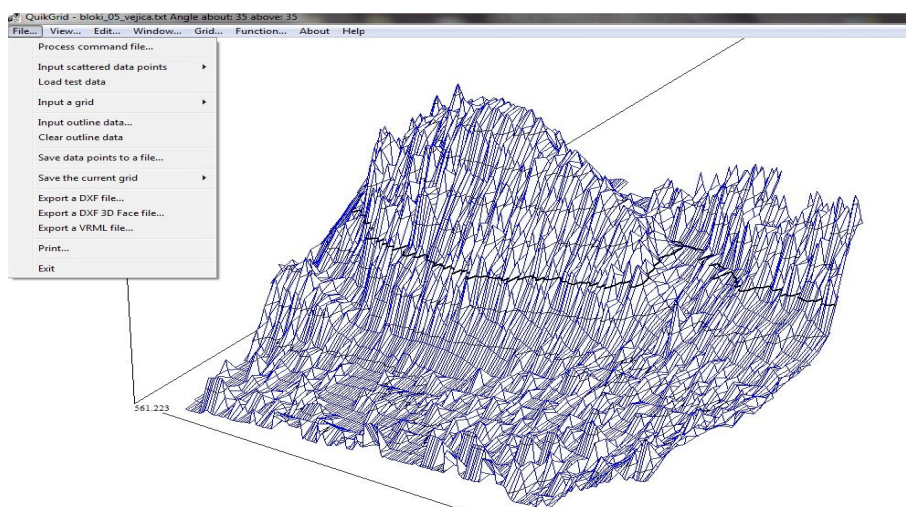
Zadnja stabilna verzija: 5.4, **datum:** 6.12.2009

Spletna stran: <http://www.galiander.ca/quikgrid/>

Program QuikGrid je enostaven program za uporabo, ki lahko v zelo kratkem času prebere veliko količino razpršenih točk (x, y, z), ki predstavljajo površje. Približno 5 milijonov točk prebere v 15 sekundah. Iz teh točk program generira mrežo in nato prikaže površje kot karto izohips (angl. contour)

⁴⁴ Windows emulator oz. WINE je prosta programska aplikacija, ki omogoča za Windows OS namenjenim računalniškim programom zagon na OS Unix-like.

ali kot 3D predstavitev. Odpira tekstovni ASCII format, kjer so podatki shranjeni kot X, Y, Z koordinate in ustvari mrežo iz podanih podatkov.



Slika 5-5: Primer uporabe programa QuikGrid

V program lahko uvozimo in izvozimo naslednje formate:

Uvoz: *.QCF, *.DCA, (USGS DEM) *.DEM, *.DXF, *.3DF, *.LL, *.NOS, *.OUT, (QuikGrid DEM) *.QG, *.XYZ

Izvoz: *.QG, *.ERS, *.DXF, *.3DF, *.XYZ, *.GRD, *.WRL

5.7.2.7 Whitebox GAT

Licenca: GNU GPL

Programski jezik: Visual Basic .NET, Python, C#

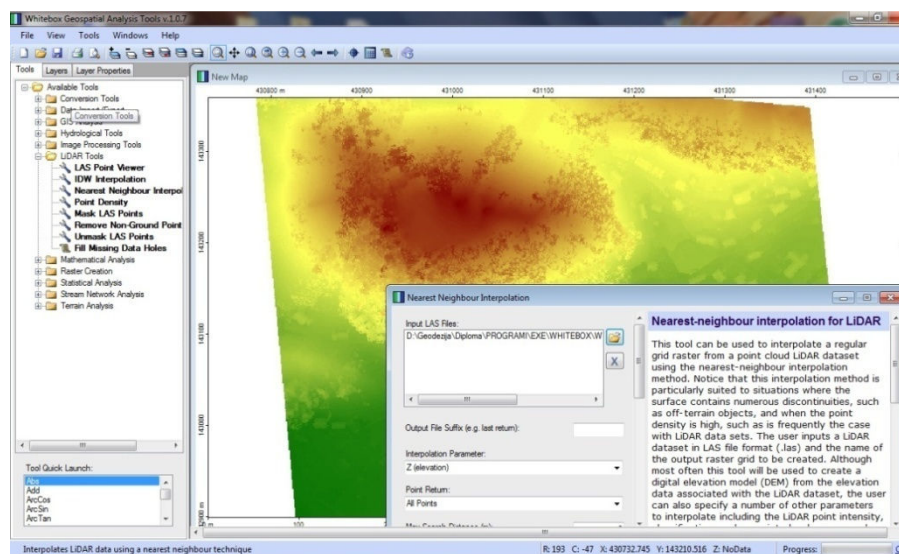
Platforma: Microsoft Windows

Zadnja stabilna verzija: 1.0.7, **datum:** 23.6.2010

Spletna stran: <http://www.uoguelph.ca/~hydrogeo/Whitebox/index.html>

Whitebox Geospatial Analysis Tools (GAT) je programski paket za GIS in daljinsko zaznavanje. Je enostaven za uporabo. Razvili so ga na Univerzi Guelph, Ontario, Kanada. Namenjen je raziskovalnim in izobraževalnim prostorskim analizam in vizualizaciji prostorskih podatkov. Program ne vsebuje namestitvene datoteke z namenom porazdeljevanja med čim več uporabnikov. Potrebno je zagnati le izvršljivo datoteko (*.exe). Vsebuje uporabniški grafični vmesnik. Vsebuje 180 orodij za obdelavo in analiziranje prostorskih podatkov. Opisi in razlage funkcionalnosti (orodij) so dodani v samem programu. Ko uporabnik požene določeno orodje, se v oknu pojavi razlaga. Nekatere funkcionalnosti

so razložene tudi v datotekah pomoči. Uporabniki lahko razvijajo in dodajajo orodja po lastnih potrebah. Projekt vsebuje tudi dobro opisane vadnice in navodila za uporabo.



Slika 5-6: Primer uporabe programa Whitebox GAT

Trenutna verzija še ne podpira vektorskih podatkovnih struktur, toda podpora Shapefile-ov (*.SHP) je že določena v razvojnem planu.

Program omogoča uvoz LAS formata, nudi vpogled v LAD datoteko, filtriranje točk glede na razrede klasifikacije, interpolacijo, izdelavo digitalnih modelov površja (DMR, DMV), glajenje ipd.

Omogoča izvoz in uvoz različnih formatov kot so: ArcGIS ASCII Grid (*.txt, *.asc), ArcGIS Grid (*.flt), IDRISI Raster (*.rst), SURFER ASCII Grid (Golden Software *.grd), CDED (*.dem), SRTM data (*.hgt), itd.

5.7.3 Opis zanimivih, s selektivnimi kriteriji izločenih, programov in knjižnic

5.7.3.1 LAStools

Licenca: LGPL

Programski jezik: C++

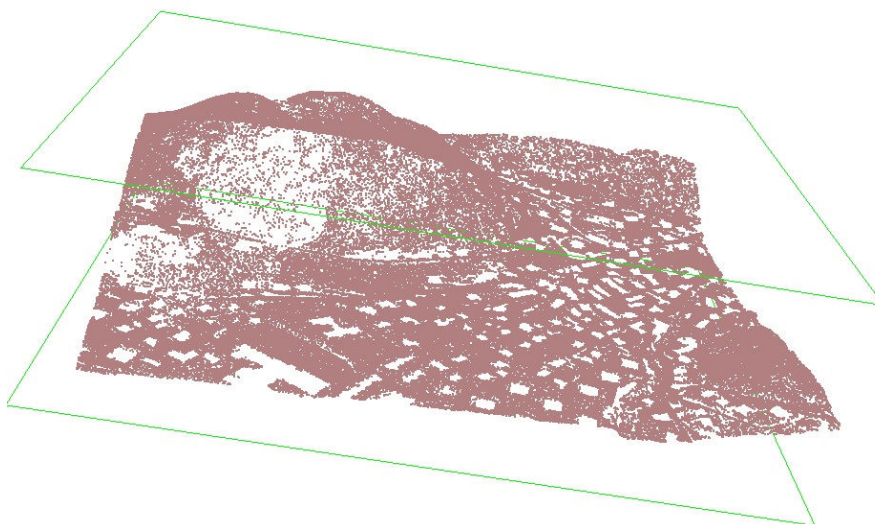
Platforma: Microsoft Windows, Linux, Unix

Zadnja stabilna verzija: ni podatka, **datum zadnje posodobitve:** 23.5.2011

Spletna stran: <http://www.cs.unc.edu/~isenburg/lastools/>

LAStools je zbirka orodij z ukazno vrstico (command-line tool), ki omogočajo preoblikovnanje (angl. converting), filtriranje, pregledovanje, procesiranje in kompresiranje lidarskih podatkov. S

programskim vmesnikom⁴⁵ LASlib, s pomočjo katerega se izvršuje branje in zapisovanje lidarskih podatkov iz in v ASPRS LAS format (verzije od 1.0 do 1.3). LAStools, zgrajeni na osnovi LASlib API vmesnika, predstavljajo visoko zmogljiva orodja z ukazno vrstico za procesiranje standardnega LAS in stisnjene LAZ formata. V preglednici Preglednica 5-7 je zbranih nekaj za obdelavo lidarskih podatkov zanimivih orodij, razširjen seznam in njihov opis z navodili za uporabo pa je zbran na spletni strani projekta (README.txt).



Slika 5-7: Prikaz oblaka terenskih točk z LASview orodjem

Preglednica 5-7: Seznam nekaterih LAStools orodij

Orodje	Kratek opis
<u>las2txt.exe</u>	Pretvori binarni LAS format v berljiv tekstovni ASCII format
<u>lasinfo.exe</u>	Omogoča splošen pogled vsebine LAS datoteke
<u>las2shp.exe</u>	Pretvori LAS format v ESRI-jev Shape format
<u>lasmerge.exe</u>	Združi več LAS datotek v eno datoteko
<u>las2las.exe</u>	Bere in zapisuje LAS/LAZ/ASCII datoteke, ki jih na kakršenkoli način spreminjamo (npr. filtriranje, transformiranje, ipd.)
<u>lasthin.exe</u>	Izvaja redčenje točk tako, da čez točke položi pravilno mrežo in znotraj celice ohrani samo točko z najnižjo, najvišjo ali naključno višino.
<u>las2tin.exe</u>	Izvede triangulacijo. Lahko določimo, da triangulira samo določene točke, glede na odboj (prvi, zadnji) ali klasifikacijski razred.

⁴⁵ API oz. Application Programming Interface ali programski vmesnik zagotavlja, da ima računalniški program na razpolago funkcije operacijskega sistema ali drugega računalniškega programa.

<u>las2dem.exe</u>	Točke začasno triangulira v TIN mrežo in nato rasterizira TIN v DMV. Izvozni format je lahko *.bil, *.asc, *.tif, *.png ali *.jpg
<u>las2iso.exe</u>	Izdela izolinije z izdelavo in interpolacijo začasne TIN mreže. Lahko triangulira samo določene točke (npr. prvi in zadnji odboj ter razred).
<u>lasview.exe</u>	Preprost pregledovalnik, ki temelji na OpenGL knjižnici. Vsebuje tudi interaktivne možnosti. Slika 5-7.

5.7.3.2 libLAS

Licenca: BSD

Programski jezik: C/C++

Platforma: Microsoft Windows, Linux, Unix, Mac OS X

Zadnja stabilna verzija: 1.6.0b4, **datum:** 7.1.2011

Spletna stran: <http://liblas.org/>

LibLAS je programska knjižnica za branje in pisanje LAS formata (verzije 1.0, 1.1 in 1.2). Primarno se posveča predvsem zagotavljanju »easy-to-program-with« knjižnico za razvijalce programov, ki želijo uporabiti LAS specifikacijo v svojem programu. LibLAS zagotavlja popolnoma odprtokodno knjižnico, zagotavlja napredno funkcionalnost in se v splošnem osredotoča predvsem na specifikacijo in ne na obdelavo lidarskih podatkov.

LibLAS predstavlja knjižnico, ki omogoča podporo pri branju in pisanju SPRS LAS formata v C/C++ aplikacijah (npr. SAGA GIS) in zbirko orodij z ukazno vrstico, ki temeljijo na LAsTools orodjih za manipulacijo in procesiranje LAS lidarskih podatkov. Ta orodja so zbrana na spletni strani projekta.

Knjižnica je razvita na osnovi LAsTools projekta. LibLAS različica LAsTools orodij ni več kompatibilna z ukazno vrstico. Ne glede na to, da imata LAsTools in libLAS skupno dediščino in da oba paketa kopirata drug drugega v smislu značilnosti in delovanja, sta danes precej različna.

LibLAS se posveča bolj namenu biti programska razvojna knjižnica, ki omogoča LAS procesiranje in ne procesna knjižnica kot taka.

5.7.3.3 MCC-LIDAR

Licenca: Apache License

Programski jezik: C++

Platforma: več OS (npr. Microsoft Windows)

Zadnja stabilna verzija: 1.0rc4, **datum:** 14.5.2011

Spletna stran: <http://sourceforge.net/projects/mcclidar/>

MCC-LIDAR je aplikacija za procesiranje lidarskih podatkov. Lidarske točke klasificira v LAS format kot terenske in ne-terenske z uporabo algoritma »Multiscale Curvature Classification algorithm«. Od tod tudi kratica MCC.

Program se izvaja iz konzole (angl. Command Prompt). Zahtevni del uporabe programa MCC-LIDAR je določitev dveh parametrov: merila (angl. scale) in ukrivljenosti (angl. curvature). Namestitev programa je enostavna. Priporočeno je v ukazno vrstico vpisati ukaz *mcc-lidar -help*, ki prikaže kratka navodila za uporabo.

Uvozimo lahko LAS format. Za izvozni format lahko določimo ASCII rastrski format (*.asc), ki izvozi rastrsko datoteko s površjem, interpoliranim iz terenskih točk in ASCII tekstovni format (*.csv), ki izvozi terenske točke, med seboj ločene z vejico.

5.7.3.4 ALDPAT

Licenca: freeware

Programski jezik: C++

Platforma: več OS (npr. Microsoft Windows)

Zadnja stabilna verzija: 1.0, **datum:** 26.4.2007

Spletna stran: <http://lidar.ihrc.fiu.edu/lidartool.html>

ALDPAT oz. Airborne LiDAR Data Processing Tool je projekt, ki je bil delno sponzoriran s strani National Science Foundation in National Oceanic and Atmospheric Administration.

ALDPAT je zbirka transparentnih in avtomatiziranih algoritmov za filtriranje zajetih lidarskih točk, ki klasificirajo zajete točke na terenske in ne-terenske. Program vsebuje GUI, ki poenostavlja uporabo algoritmov. Razvitje bil predvsem za raziskovalne namene.

Program omogoča izvajanje naslednjih nalog:

- razdelitev snemalnega pasu leta nosilca na več blokov (angl. split strip)
- ločitev prvega in zadnjega odboja meritev signala
- ocenitev meja blokov
- izdelava ESRI Shape file (*.shp) datoteke
- filtriranje lidarskih točk po geometrijskih lastnostih (ETEW Filter, Morph Filter, Morph Circle, Filter, Morph 2D filter, Slope Filter, Polynomial Filter, Polynomial 2 Surface Filter, Adaptive TIN Filter)

Program omogoča uvoz podatkov v ASCII tekstovnem formatu (*.txt), kjer so koordinate oz. atributi točk med seboj ločeni s presledkom ali vejico. Izvozni formati so lahko ASCII Ratser format (*.asc) ali SURFER format (*.grd). Program vsebuje tudi navodila za posamezno funkcionalnost.

5.7.3.5 GEON Points2Grid

Licenca: freeware

Programski jezik: ni podatka

Platforma: Microsoft Windows (Linux, Mac OS X)

Zadnja stabilna verzija: 1.3, **datum:** 24.4.2007

Spletna stran: <http://lidar.asu.edu/points2grid.html>

GEON Points2Grid Utility je orodje za generiranje DMV-jev iz oblaka lidarskih točk. Razvila ga je ekipa znanstvenikov dveh ameriških univerz in sicer Arizona State University in University of California, San Diego. V projekt je investirala National Science Foundation.

Orodje ponuja hitro generiranje DMV iz velike količine lidarskih podatkov. Program deluje na ASCII tekstovnem formatu, kjer so atributi ločeni med seboj z vejico. Izvozni format izdelanega DMV-ja je lahko v ESRI Arc ASCII Grid ali v standardnem ASCII Grid. Uporabnik izbere algoritem, s katerim želi izdelati DMV, resolucijo mreže in radij, saj algoritem uporabi pri izdelavi mreže le točke znotraj radija, katerega posredno določi uporabnik sam z izbiro resolucije mreže. Program se lahko uporablja za različne podatke, pomembno je le, da so organizirani v 3D oblak točk.

Po avtorjevih besedah je Windows različica zastarela. Razvijalci pripravljajo novo verzijo, ki pa bo licencirana pod BSD licenco. Trenutno je na spletni strani Sourceforge.net na razpolago nova verzija orodja, ki deluje Linux in Mac OS X operacijskih sistemih.

5.7.3.6 FullAnalyze

Licenca: GNU LGPL

Programski jezik: C++

Platforma: Linux

Zadnja stabilna verzija: 1.2, **datum:** 5.11.2009

Spletna stran: <https://www.ohloh.net/p/fullanalyze>

FullAnalyze je raziskovalno orodje, ki ga je razvil MATIS laboratorij na IGN⁴⁶ in CEMAGREF na Centru za daljinsko zaznavanje⁴⁷, Francija.

Program je namenjen obdelovanju, procesiranju in vizualizaciji lidarskih podatkov (oblak točk in waveform). Posebnost programa je, da lahko obdeluje polnovalovne (angl. full-waveform) lidarske podatke. Standardni format za polnovalovne podatke ne obstaja, poznamo pa LAS format za »večodbojne« (angl. multi-echo) lidarske podatke. Surovi podatki so shranjeni v lastniških formatih. Obdelava teh podatkov zato predstavlja izziv na področju lidarske tehnologije.

Fullanalyze je sposoben:

- georeferenciranja podatkov
- vizualizacije polnovalovnih podatkov, ki jih lahko s programom prikažemo kot 1D signal topologije senzorja (kot skenerja in čas), hkrati pa jih lahko prikažemo v 3D kartografskem pogledu, ki jih naložimo na klasičen 3D oblak točk za lažjo interpretacijo in razumevanje podatkov.
- procesiranja polnovalovnih podatkov, ki sloni na tehniki s parametričnim pristopom, ki skuša prilagoditi obliko vala in izvleči pomembne vrhove kot odboje. To omogoča pridobitev x, y, z lokacije kot tudi dodatne informacije o obliki tarče in njeni odbojni jakosti (angl. reflectance). Ta metoda je zelo učinkovita na pogozdenih območjih, saj omogoča zaznavanje več dodatnih odbojev v primerjavi z lidarjem (multi-echo). Na urbanih območjih lahko pridobimo boljše klasifikacijske rezultate.

Cilj programa FullAnalyze je tudi ta, da bi bil modularen z uporabo arhitekture dodatkov (pluginov), saj bi tako na enostaven način lahko uporabnik dodajal algoritme za obdelavo full-waveform lidarskih podatkov.

⁴⁶ Institut Géographique National oz. Nacionalna francoska kartografska agencija

⁴⁷ Maison de la Télédétection

FullAnalyze je bil uspešno uporabljen za obdelavo in procesiranje majhnega vzorca full-waveform lidarskih podatkov (Chauve, A. et al. 2009).

6 NAVODILA ZA IZDELAVO DMR Z IZBRANIMA PROGRAMOMA

V tem poglavju sem izdelal priročna navodila za uporabo iz Poglavja 5.6 izbranih dveh programov. V navodilih je opisan način izdelave DMR-ja iz oblaka lidarskih točk. V splošnem velja, da je potrebno oblak točk razdeliti na terenske (tla) in ne-terenske (stavbe, vegetacija, ipd) točke. Ker v oblaku točk terenske točke niso pravilno porazdeljene, jih je potrebno urediti v pravilno mrežo uniformnih celic enake oblike - raster. S pomočjo prostorske interpolacije določimo iskano vrednost (višino) vsem rastrskim celicam. Te vrednosti se definira z izračunom povprečja vrednosti bližnjih točk, katerih pomembnost oz. vpliv na določitev iskane vrednosti lahko definiramo z utežmi.

Podatke za obdelavo, ki sem jih uporabil v tem poglavju sem dobil na FGG. Avgusta, leta 2008 je podjetje Flycom opravilo snemanje na območju Koroške bele z Rieglovim inštrumentom. Podatke so uredili s programskim paketom TerraSolid in so shranjeni v formatu LAS 1.1.

6.1 Izdelava DMR s programom FUSION/LDV

6.1.1 Prenos in namestitvev programa

Z domače spletne strani⁴⁸ programa FUSION/LDV (v nadaljevanju FUSION) prenesemo namestitveno datoteko (Fusion_install.exe) in programski priročnik s kratkimi opisi programa in programskih orodij (Fusion_manual.pdf). Med namestitvijo sledimo navodilom in uporabimo priporočene nastavitve. Priporočeno je, da si izberemo direktorij oz. mapo (angl. folder), kjer bodo shranjene programske komponente s kratkim imenom in potjo do mape, saj bomo ime poti večkrat uporabili pri obdelavi podatkov. Izbral sem pot C:\Fusion\. Enako velja za mapo s podatki, ki jih bomo obdelovali npr. C:\lidar\bloki\.

6.1.2 Zagon programa in priprava projekta

- Zagon programa: Izberemo **Start | All Programs | FUSION | FUSION** ali s klikom na bližnico, če smo si jo ustvarili
- Shranimo projekt: **File | Save As |**
- Poimenujmo projekt npr. **Vaja**, izberemo končnico *.dvz, s katero FUSION ustvari projektno datoteko in ustvarimo novo mapo za shranjevanje projektov npr. **C:\lidar\Projekti**.

Na tem mestu lahko pričnemo z obdelavo podatkov.

⁴⁸ <http://forsys.cfr.washington.edu/fusion/fusionlatest.html>

6.1.3 Predogled podatkov

V mapo s podatki C:\lidar\bloki\ uvozimo surove lidarske podatke, ki smo jih dobili od ponudnika podatkov. S temi podatki npr. **bloki5.las** bomo izdelali DMR.

Primarna podatkovna formata, s katerima lahko izvajamo obdelavo v programu FUSION, sta formata s končnicami *.lda in *.las, kjer je LDA naravni format programa. V FUSION lahko uvozimo tudi druge formate kot sta ASCII Text format in ASCII CSV, v katerem so vrednosti atributov med seboj ločeni z vejico (angl. comma separated value). Ta dva formata morata za uporabo v programu FUSION biti preoblikovana oz. konvertirana v format LDA. Tudi LAS je lahko konvertiran v LDA, toda kot sem že omenil, ga lahko v programu neposredno uporabljamo. Potrebno je le ustvariti indeksno datoteko (*.idx), ki pa jo program ustvari samodejno takrat, ko je LAS datoteka prvič uporabljena.

Za predogled metapodatkov izberemo: **Tools | Miscellaneous utilities | Examine LAS Files**

Odpre se okno Ehamine LAS Files, kjer si lahko ogledamo, s katerim programom so bili podatki pripravljani, v katerem LAS formatu so zapisani, število zajetih točk, meje zajetega območja ipd.

6.1.4 Konvertiranje podatkov

Ker bomo v nadaljevanju obdelave potrebovali vrednosti intenzitete, si predhodno pripravimo CSV datoteko (*.csv), s pomočjo katere lahko izvemo dodatne informacije o podatkih v LAS datoteki.

- Izberemo: **Tools | Data Conversion | Export data from LAS or LDA files to other formats...**
- Odpre se okno **Convert and index LIDARDAT files**.
- Kot uvozno datoteko (angl. input file) izberemo želeno LAS datoteko, ki se nahaja v mapi C:\lidar\bloki\bloki5.las.
- V vrstici za izvozno datoteko (angl. output file) določimo pot do naše mape \bloki\ in ustvarimo datoteko s končnico *.csv. Izberemo ime in mu ročno pripišemo končnico *.csv.
- V padajočem seznamu okna **Convert and index LIDARDAT files** izberemo ASCII CSV format (XYZIPRS).
- Izberemo **Convert**.
- V naši mapi s podatki imamo novo datoteko npr. bloki5.csv, ki jo lahko odpremo s programom Microsoft Excel. Excel lahko odpre le del datoteke, saj je omejen z določenim

številom vrstic (1 048 576, Excel 2007), toda še vedno predstavlja dovolj dober vpogled v same podatke.

6.1.5 Izdelava referenčne podobe

Kot sem že omenil, potrebuje FUSION za uvoz LAS podatkov predhodno naloženo referenčno podobo (npr *.tif), ki zagotavlja koordinatni sistem in vizualno referenco za lidarske podatke. Čeprav so naši LAS podatki kompatibilni s programom FUSION, jih ne moremo videti, dokler ni naložena referenčna podoba.

Če imamo referenčno podobo, jo naložimo:

- V stolpcu z orodji izberemo **Image...**, poiščemo že pripravljeno referenčno podobo in izberemo **Open**.

Če referenčne podobe nimamo, jo lahko izdelamo s pomočjo intenzitete lidarskih podatkov:

- Izberemo: **Tools | Miscellaneous utilities | Create an image using LIDAR point data**
- Poiščemo naše podatke npr. bloki5.las, jih označimo in izberemo **Open**.
- Odpre se novo okno **Image Creator**.
- V vrstici **Image file name** izberemo **Browse**, kjer določimo lokacijo podobe, kamor jo želimo shraniti npr C:\lidar\bloki\ in ji ročno določimo ime s končnico npr. bloki5.bmp.
- Izberemo **Save**.
- V oknu **Image creator** lahko spreminjamo velikost piksla, barvno lestvico podobe, barvo ozadja in določamo meje podatkovnih vrednosti ter način izdelave podobe na osnovi izbranih podatkov npr. intenziteta.

Izdelali bomo podobo s pomočjo vrednosti intenzitete. Da lažje določimo meje njenih vrednosti, lahko uporabimo funkcijo »Scan for data ranges«. S pomočjo te funkcije lahko določimo razpon vrednosti intenzitete, ki so najpogosteje zastopane. V primeru podatkov bloki5.las nam funkcija določi vrednosti od 10 do 1151. Te vrednosti niso pravilno definirane, saj ustvarjamo barvno podobo, v našem primeru z barvno lestvico RGB, ki ima vrednosti razporejene od 0 do 255. Za premostitev te težave si pomagamo s CSV datoteko, ki smo jo ustvarili v Poglavju 6.4.

- Odpremo datoteko **bloki5.csv** v programu **Microsoft Excel**.

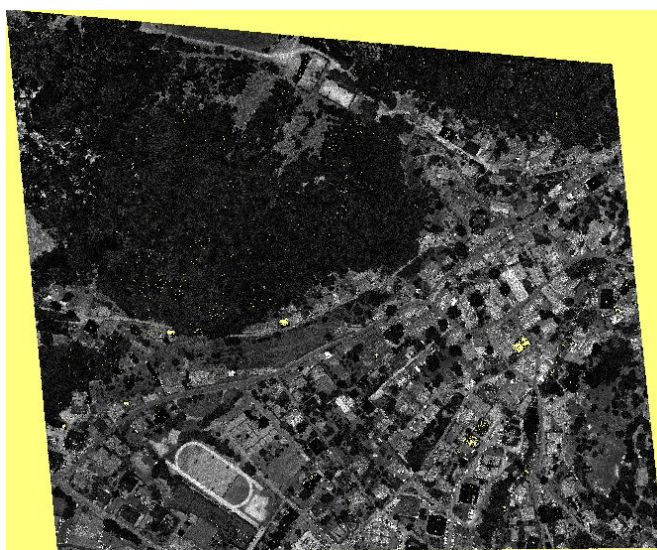
Program nas opozori, da ne more prikazati vseh podatkov. Prikaže lahko »le« približno milijon vrstic (ena vrstica za vsako zajeto točko). Glede na to, da je v bloku podatkov zajetih približno 5 milijonov točk (Examine LAS files), nam vseeno omogoča dober vpogled v dodeljene vrednosti intenzitete.

- Izberemo stolpec z vrednostmi intenzitete in ga po vrednostih uredimo naraščajoče ali padajoče.
- Približno ocenimo vzorec razpona vrednosti, kjer ne upoštevamo vrednosti, ki drastično izstopajo.
- V oknu **Image creator** programa FUSION, označimo **Clamp data** in v polja vpišemo mejne vrednosti.
- Izberemo **Create image**.

Opomba: Če prejmemo obvestilo, da se program ne odziva, je potrebno počakati, saj se to obvestilo pojavi večkrat in sicer, ko program obdeluje podatke. Verjetno napaka programa, vendar bodo podatki zagotovo obdelani, kar program sporoči v statusni vrstici okna **Image creator**.

- Ko je podoba izdelana, izberemo **Close**, da zapremo okno **Image creator**.
- Izdelano podobo naložimo z izbiro **Image...** v stolpcu z orodji. Če se podoba ne pojavi v glavnem oknu, je potrebno imeti označen kvadrataček pred gumbom Image...

Če s podobo nismo zadovoljni, lahko izdelavo ponovimo, ter spremenimo mejne vrednosti. Za primer bloki5.las sem uporabil vrednosti intenzitete od 10 do 106, izbral ozadje ter velikost piksla 1. Za vsako izdelano podobo program avtomatsko izdela world datoteko. Za BMP datoteke je ta datoteka s končnico *.bmpw. Slika 6-1 prikazuje primer izdelane podobe.



Slika 6-1: Primer izdelane referenčne podobe s pomočjo vrednosti intenzitete

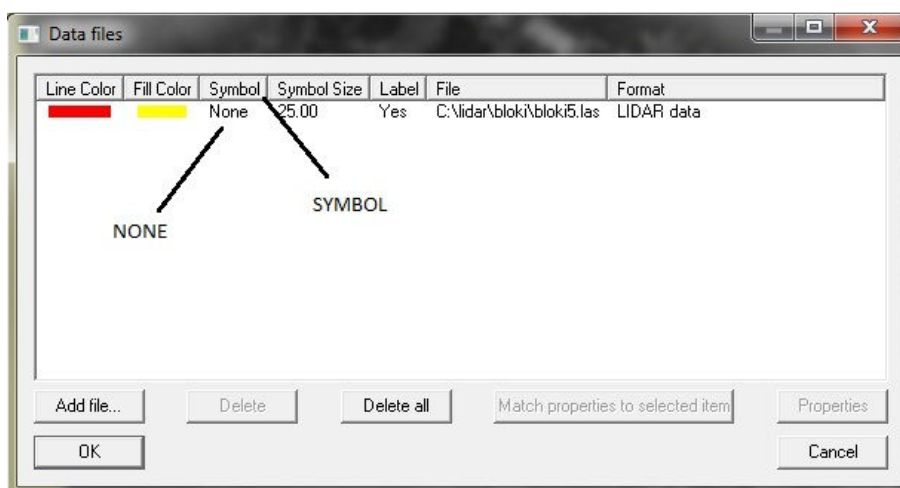
6.1.6 Uvoz podatkov

Ko naložimo podobo, lahko naložimo tudi LAS podatke:

- V stolpcu z orodji izberemo **RAW data...**
- Odpre se okno **Data files**, kjer z izbiro **Add file...** uvozimo podatke.
- Ko izberemo datoteko s podatki, preverimo, če je pod seznamom Symbol v vrstici z izbranimi podatki, izbrana možnost **None** kot prikazuje Slika 6-2. Če ni, dvakrat kliknemo na vrstico in to možnost izberemo. To izberemo, ker ne želimo prikazovanja lidarskih podatkov v glavnem oknu programa, ker je pri vsaki spremembi pogleda dolgotrajen postopek prikaza točk.
- Kvadrataček pred gumbom **RAW data...** pustimo prazen, saj je količina podatkov prevelika, da bi jih prikazovali v glavnem oknu.
- Za ogled lidarskih podatkov v glavnem oknu z **levim klikom** miške povlečemo izbrano območje, ki si ga želimo ogledati.

Za začetek izberemo majhno območje, saj smo v program uvozili veliko količino podatkov. Z izbiro gumba **Sample options** lahko izberemo obliko vzorca, prikaz števila točk ipd.

Ko smo izbrali vzorec oz. območje, ki ga želimo pregledati, se pojavi novo okno in sicer, LIDAR Data Viewer (LDV). Če izberemo vprašaj v orodni vrstici LDV-ja, se pojavijo navodila za uporabo pregledovalnika LDV. Z desnim klikom miške, v glavnem oknu LDV, se pojavi seznam dodatnih funkcionalnosti, ki služijo prikazovanju različnih slojev podatkov.



Slika 6-2: Izbira simbola

Ko imamo naloženo referenčno podobo in surove podatke, lahko pričnemo z izdelavo površja.

6.1.7 Izdelava DMR

Izdelava DMR-ja poteka v dveh korakih. Najprej bomo podatke filtrirali, s čimer izločimo iz nadaljnje obdelave vse ne-terenske točke, nato pa s pomočjo mreženja izdelali DMR iz terenskih točk.

Za filtriranje točk bomo uporabili orodje GroundFilter (v nadaljevanju filter) z ukazno vrstico. Algoritem, ki ga orodje uporablja, je razvit na osnovi teorije K. Krausa in N. Pfeiferja. Filter filtrira oblak točk tako, da prepozna točke, ki ležijo na verjetnem površju tal. Algoritem je osnovan na linearni predikciji s posamezno točnostjo za vsako meritev. Deluje iterativno, kjer v vsaki iteraciji izračuna površje z enakimi utežmi za vse točke oz. točneje z-meritve. To površje poteka v povprečju med vegetacijo in terenskimi točkami (Kraus, K., Pfeifer, N., 1998).

Slabost tega filtra je, da ne izdelava dovršenega niza terenskih točk, saj popolnoma ne odstrani ostalih (ne-terenskih) odbojev. Ta lastnost je značilna predvsem na večjih, relativno ravnih urbanih območjih, zato ni primeren za obdelavo točk, kjer je cilj izdelava DMR-ja. V splošnem ta filter proizvede točkovne nize, s katerimi lahko izdelamo digitalne modele površja, ki jih lahko uporabimo pri določanju višine vegetacije. Filter opravlja svojo nalogo dobro predvsem na pogozdenih območjih, za kar je program FUSION tudi specializiran.

Teorija je opisana v članku »Determination of terrain models in wooded areas with airborne laser scanner data« (Kraus, K., Pfeifer, N., 1998).

Za primerjavo terena pogozdenih in urbanih območij po uporabi filtra lahko vseeno izdelamo DMR.

6.1.7.1 Uporaba orodja Groundfilter

Postopek filtriranja izvedemo v orodju z ukazno vrstico GroundFilter.

- Odpremo DOS ukazno okno: **Start** | v iskalno vrstico vpišemo **cmd** in izberemo **cmd.exe**.
- Določimo pot do mape FUSION, ki smo jo ustvarili med namestitvijo programa.
- Ukaz: **cd c:\fusion** oz. pot do mape, kjer je shranjena mapa FUSION
- Vpišemo: **GroundFilter**, da dobimo opis celotne sintakse⁴⁹.

Opomba: Uporaba velikih črk ni potrebna.

⁴⁹ Sintaksa je niz pravil, ki določajo strukturo stavkov, zaporedja besed in odnosa med njimi.

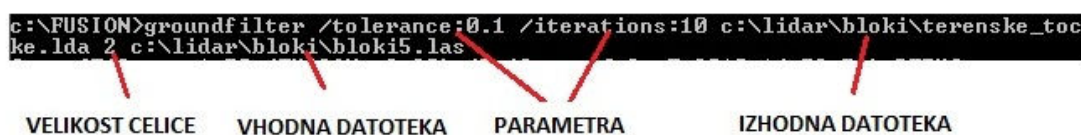
Uporabimo orodje **GroundFilter** za izdelavo datoteke s terenskimi točkami iz podatkovne datoteke bloki5.las. Če uporabimo določene predhodno nastavljene parametre, za katere avtorji algoritma trdijo, da so optimalni za večino območij raziskave, moramo določiti le še naslednje parametre in ukaze:

- **toleranca**, kjer določimo vrednost za končno filtriranje terenskih točk. Samo točke znotraj vrednosti bodo vključene v izhodni datoteki. Če tolerance ne določimo, bo za vsako točko v postopku filtriranja določena utežna vrednost
- **iteracija**, kjer določimo število ponovitev izvedbe algoritma
- **velikost celice**
- **vhodno in izhodno datoteko** s podatki

Sintaksa v vrstici C:\FUSION> mora izgledati tako kot kaže Slika 6-3.

```
groundfilter /tolerance: 0.1 /iterations:10 c:\lidar\bloki\terenske_tocke.lda 2 c:\lidar\bloki\bloki5.las
```

Toleranco, velikost celice in iteracije, lahko poljubno spreminjamo in opazujemo spremembe. Pozorni moramo biti na vse presledke in znake, ki jih vpisujemo, da bo program razumel naš ukaz.

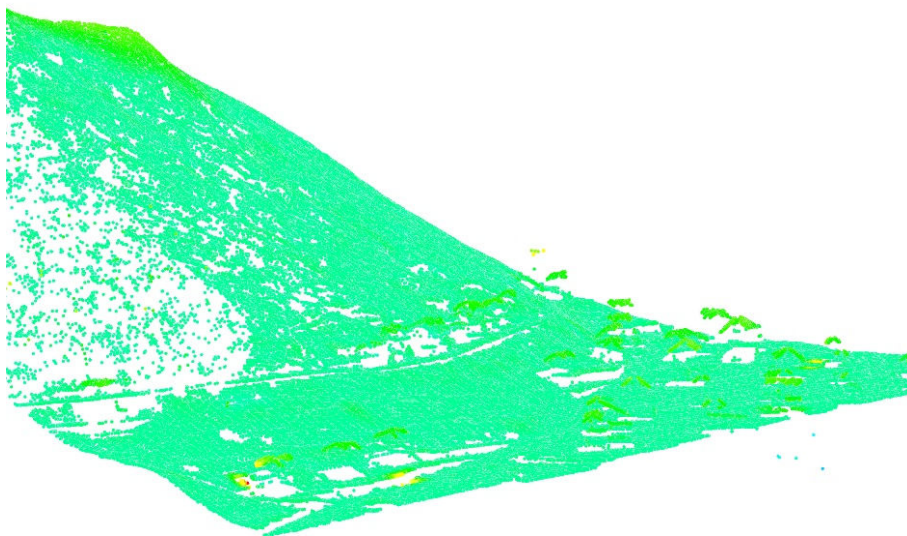


Slika 6-3: Razlaga delov sintakse

Podrobni opisi vseh ukazov in switch⁵⁰-ev so zbrani v datoteki FUSION_manual.pdf, ki je dostopna na spletni strani projekta.

Orodje izdelava datoteko **terenske_tocke.lda**, ki jo lahko uvozimo v FUSION, tako da izberemo **Raw data..** v ukaznem stolpcu. V LDV-ju si lahko ogledamo rezultat filtriranja točk, ki ga prikazuje Slika 6-4.

⁵⁰ Switch je pri orodjih z ukazno vrstico namig uporabnika računalniškemu programu, da naj spremeni predhodno nastavljene lastnosti izvedbe ukazov.



Slika 6-4: Oblak točk po filtriranju s filtrom GroundFilter

Kot je iz Slika 6-4 razvidno, so v oblaku točk, kjer je urbano območje, še vedno prisotne točke, ki določajo strehe stavb. Filter je, kot opisuje teorija, dobro odstranil ne-terenske točke na pogozdem območju (na sliki levi zgornji kot).

6.1.7.2 Uporaba orodja GridSurfaceCreate

GridSurfaceCreate izdelava mrežni model površja z uporabo diskretno razporejenih točk. Izhodna datoteka je shranjena v PLANS DTM formatu, ki uporablja višinske vrednosti točk. Posamezna višina celice je določena z izračunom povprečja višin vseh točk, ki se nahajajo znotraj celice. Orodje se ponavadi uporablja po postopku filtriranja ali z uporabo podatkovne datoteke terenskih točk, ki nam jih zagotovi ponudnik.

Ker v nadaljnjo uporabo ni smiselno vključiti z GroundFilter orodjem filtriranih točk oblaka, bomo to storili neposredno v GridSurfaceCreate orodju, saj je še vedno potrebno ne-terenske točke ločiti od terenskih.

Digitalni model reliefa bomo izdelali po naslednjem postopku:

- Odpremo DOS ukazno okno: **Start** | v iskalno vrstico vpišemo **cmd** in izberemo **cmd.exe**.
- Določimo pot do mape FUSION, ki smo jo ustvarili med namestitvijo programa.
- Ukaz: **cd c:\fusion** oz. pot do mape, kjer je shranjena mapa FUSION.
- Vpišemo: **GridSurfaceCreate**, da dobimo opis celotne sintakse.

Za izdelavo DMR-ja in predhodno filtriranje oblaka točk nastavimo naslednje parametre:

- **spike**; Filtrira končno površje, da odstrani špice oz. vrhove. Določimo vrednost v odstotkih, nad katero izloči vse navpične vrhove. Določimo, da naj odstrani vse vrhove, ki imajo več kot 70% naklon.
- **class**; Uporabi samo odboje s klasifikacijsko vrednostjo. Tlom je dodeljen 2. klasifikacijski razred v LAS formatu, zato izberemo vrednost 2. To stikalo (angl. switch) je mogoče uporabiti samo pri vhodnih podatkih v LAS podatkovnem formatu.
- **class:string**; Stikalo, ki ga lahko uporabimo samo pri LAS vhodnih podatkih. Omogoča uvrstitev tistih točk v vzorec, ki zasedajo klasifikacijske vrednosti, določene s strani uporabnika. Določimo vrednosti 2, 3, 4 in 5, ki predstavljajo tla in vegetacijo.
- **minimum**; Program uporabi najnižjo višino vseh točk v celici kot višino celice. Zaradi tega parametra moramo posledično določiti majhno velikost celice, da se izognemo izdelavi površja, ki leži pod dejanskim terenom.
- **velikost celice**; Je enakih enot kot lidarski podatki. Določimo vrednost 1.
- **xy enota**; M –meter
- **z enota**; M –meter
- **projekcija, cona, vertikalni datum in horizontalni datum**; Ker program ne omogoča izbire teh parametrov za območje Slovenije, pri vseh določimo 0.
- **vhodna in izhodna datoteka** s podatki

V splošnem GridSurfaceCreate uporablja vhodne podatke v LDA podatkovnem formatu, vendar zaradi potrebe po izločanju ne-terenskih točk iz izdelave DMR-ja neposredno uporabimo LAS format.

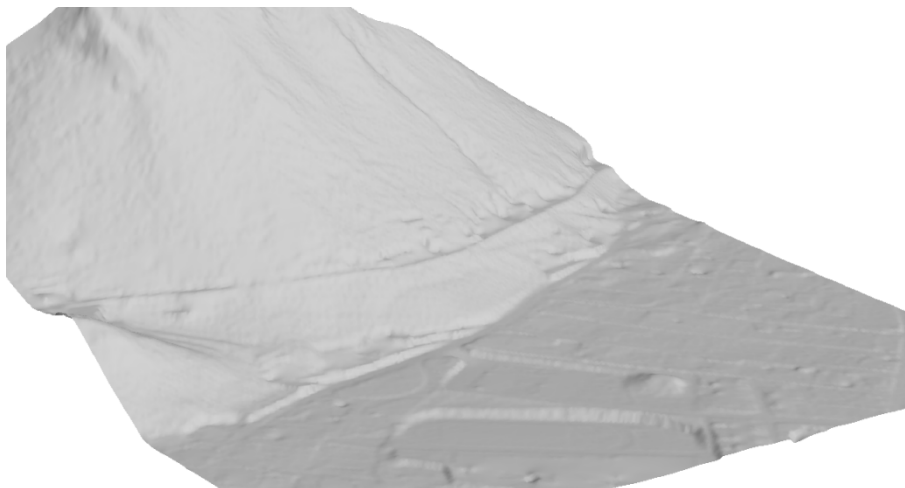
Sintaksa v vrstici C:\FUSION> izgleda:

```
gridsurfacecreate /spike:70 /class:2 /class:string2,3,4,5 /minimum c:\lidar\bloki\DMR.dtm 1 M M 0 0  
0 0 c:\lidar\bloki\bloki5.las
```

Ko orodje GridSurfaceCreate izdelava DMR.dtm, ga lahko uvozimo v FUSION.

- Izberemo **Bare earth...** v ukaznem stolpcu.
- Pojavi se novo okno **Surface model**.
- Izberemo zeleno datoteko.
- Po želji spreminjamo barve, število in ekvidistanco izohips.
- V glavnem oknu programa se izrišejo izohipse. Če se ne, imamo verjetno izključen sloj **Bare earth...**

- Izberemo območje z levim klikom miške, da se odpre **LDV**.
- V **LDV**-ju lahko s desnim klikom vklopimo **seznam** dodatnih programskih orodij, s katerimi preklapljamo med sloji in prikazom.

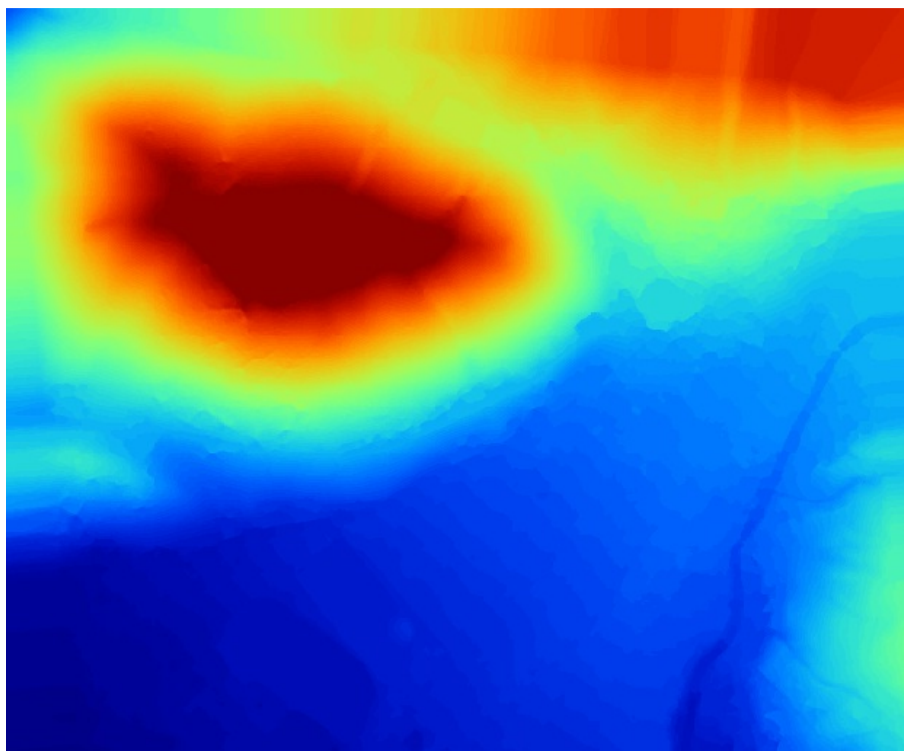


Slika 6-5: Plastičen prikaz reliefa po uporabi orodja GridSurfaceCreate

6.1.7.3 Izvoz podatkov

Ker PLANS DTM podatkovni format ni uporaben v večini GIS programov, ga lahko izvozimo v ASCII Grid podatkovni format. To storimo na naslednji način:

- Izberemo: **Tools | Terrain mode | Export model**
- Pojavi se novo okno **Export DTM**
- V prvi vrstici določimo model, ki ga želimo izvoziti npr. **DMR.dtm**
- V drugi vrstici določimo ime izhodne datoteke, kateri ročno dodamo končnico ***.asc**
- V oknu **Export DTM** pod poglavjem **Format** izberemo ASCII grid file
- Izberemo **Export**
- Program nas obvesti, da je ustvaril ASCII Raster grid format
- ASC podatkovni format lahko uvozimo v druge GIS programe (ArcGis, SAGA, GRASS, ipd.), kar prikazuje Slika 6-6.



Slika 6-6: Prikaz DMR-ja, uvoženega v SAGA-o iz programa FUSION/LDV

6.2 Izdelava DMR s programom SAGA

6.2.1 Prenos in namestitev programa

Namestitveno datoteko programa SAGA (`saga_2.0.7_win32_setup.exe`) lahko neposredno prenesemo iz spletne strani Sourceforge.net ali pa posredno preko povezave iz domače spletne strani⁵¹ projekta SAGA. Prav tako je na spletni strani Sourceforge.net zbrana dokumentacija. Tudi na povezavi Wiki na domači spletni strani lahko najdemo dokumentacijo, ki nam je lahko v pomoč pri uporabi programa.

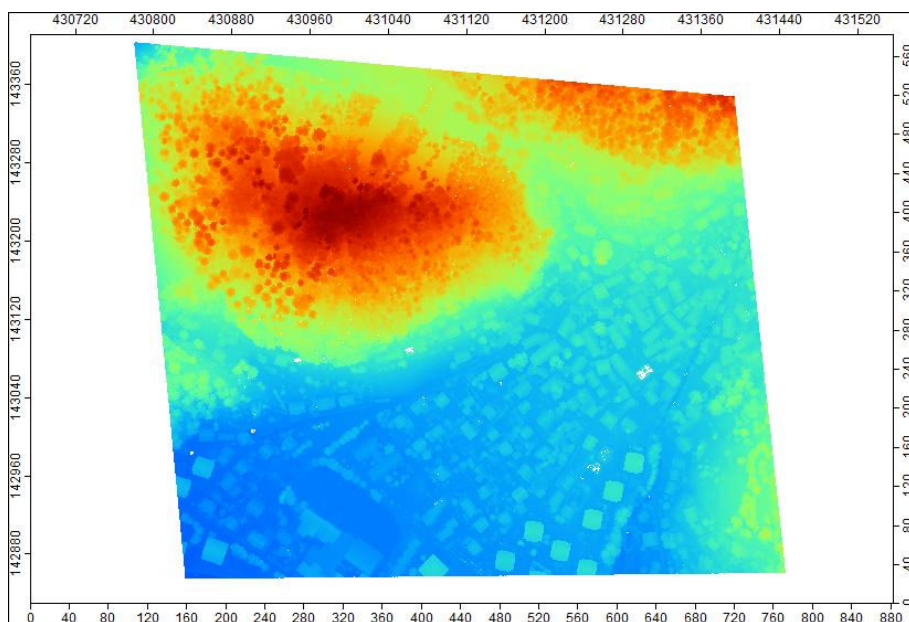
Med namestitvijo programa sledimo navodilom in uporabimo priporočene nastavitve, poljubno pa lahko spremenimo pot do namestitvenega direktorija.

6.2.2 Zagon programa in izdelava DMR

- Izberemo: **Start | All Programs | SAGA GIS** ali pa uporabimo bližnjico na namizju
- V **Delovnem okolju** (angl. Workspace) so zbrani vsi moduli, s katerimi lahko opravljamo različne operacije.
- Za ovoz podatkov v LAS podatkovnem formatu izberemo: **Import/Export – LAS** in nato **Import LAS files**, ki se pojavi v novem oknu.

⁵¹ <http://www.saga-gis.org/>

- V vrstici **Input File** izberemo podatke, ki jih želimo obdelati npr. bloki5.las
- Pod razdelkom **Attributes to import besides x, y, z...** lahko izbiramo attribute, ki jih želimo uporabiti in so zbrani c LAS datoteki. V našem primeru izberemo **intenziteto** in **klasifikacijo** tako, da označimo prazna kvadratna polja.
- Izberemo **Okay**



Slika 6-7: Oblak lidarskih točk

V **Oknu z obvestili** (angl. Message window) pod glavnim oknom programa lahko vidimo, če se je modul pravilno izvršil.

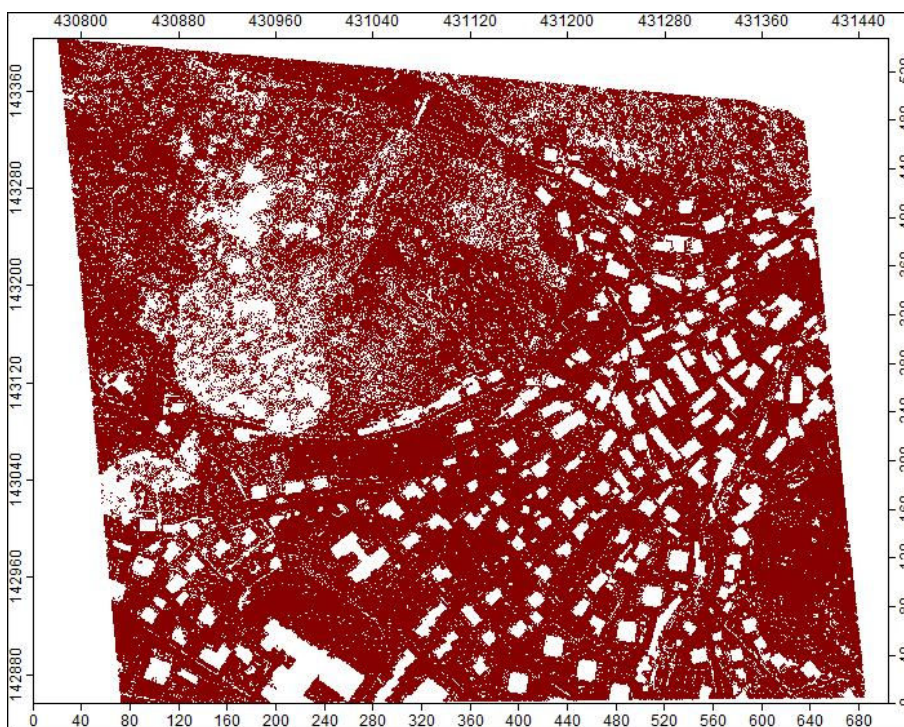
Delovno okolje sestavljajo trije zavihki **Modules**, **Data** in **Maps**. Pod zavihkom **Maps** se je po uvozu podatkov pojavil naš podatkovni točkovni sloj 01.bloki5 (Slika 6-7). Lahko se prepričamo, da smo uspešno uvozili podatke v program.

6.2.3 Filtriranje podatkov

Značilnost programa SAGA je, da vsebuje več modulov, ki v splošnem izvedejo podobne operacije s podatki. Uporabnik se lahko odloči, kateri modul bo uporabil.

Za filtriranje podatkov, kjer bomo ločili terenske točke od ne-terenskih točk bomo uporabili naslednji modul:

- Izberemo: **Shapes – Point Clouds | Point Cloud Reclassifier / Subset Extractor**, ki se pojavi v novem oknu.
- Pod **Data Objects | Point Cloud** izberemo našo datoteko s podatki, ki jo želimo filtrirati.
- Pod **Data Objects | Attribute** izberemo v padajočem seznamu **klasifikacijo**, saj bomo podatke med seboj ločili glede na razrede klasifikacije.
- Tretjo vrstico pustimo prazno, saj bomo šele izdelali nov sloj.
- Pod **Options | Mode of operation** izberemo **Extract Subset**, saj bomo ustvarili nov podniz podatkov, v katerem bodo samo terenske točke.
- Pod **Options | Method** pustimo v padajočem seznamu single. Pod **Method single** za **staro vrednost** (angl. old value) določimo vrednost **2** (razred tal), za **novi vrednost** (angl. new value), pa lahko pustimo **1**. Ker bo ena vrednost zamenjala drugo vrednost, za operator izberemo **enačaj**. Ostala polja pustimo.
- Izberemo **Okay**
- V zavihku **Data Delovnega** okolja lahko opazimo nov podatkovni sloj. Če ga označimo z desnim gumbom miške in izberemo **Add to map**, se v glavnem oknu programa pojavi izdelan sloj terenskih točk npr. 01.bloki_subset_classification (Slika 6-8).

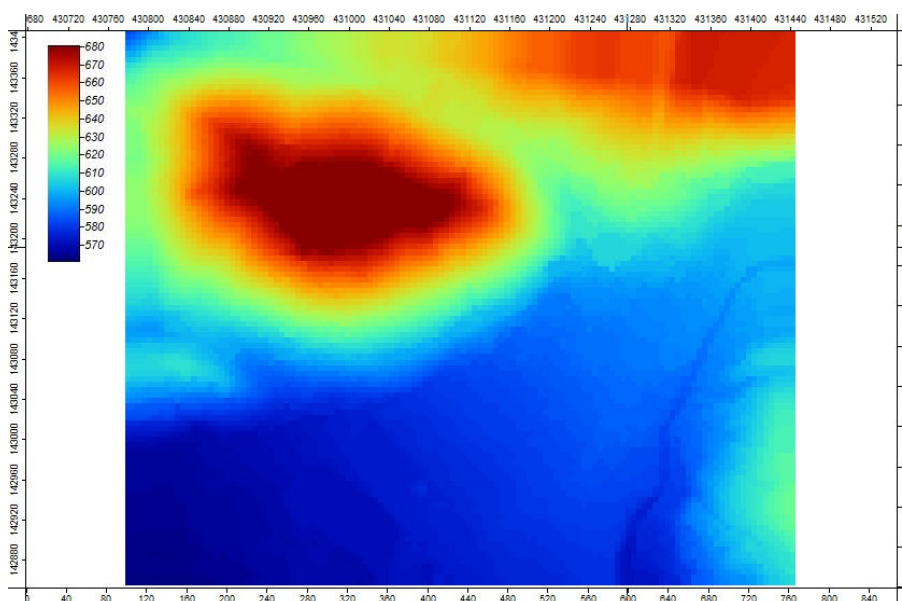


Slika 6-8: Sloj terenskih točk po filtriranju z modulom Subset Extractor

6.2.4 Izdelava DMR

V Delovnem okolju izberemo Grid – Gridding, kjer so zbrani različni moduli, ki predstavljajo različne načine modeliranja ploskev.

- Izberemo modul **Inverse Distance Weighted**, ki se pojavi v novem oknu.
- Pod **Shapes | Points** izberemo sloj, ki smo ga izdelali v prejšnjem poglavju npr. 01.bloki_subset_classification in za polje **Attribute** izberemo višino, saj želimo izvesti interpolacijo višin.
- Pod **Options | Target grid** izberemo iz padajočega seznama **user defined**.
- Ostale parametre pustimo nespremenjen.
- Izberemo **Okay** in še enkrat potrdimo izbiro.
- Modul je izdelal DMR (Slika 6-9).



Slika 6-9: DMR izdelan s pomočjo metode interpolacije inverzne razdalje

Postopek izdelave DMR-ja v programu SAGA je bistveno krajši kot v programu FUSION/LDV. Sicer uporabnik nima vpogleda v samo obdelavo ima pa zato na izbiro več metod obdelave podatkov, ki jih omogočajo različni moduli. V vaji smo uporabili večinoma predhodno nastavljene nastavitve, ki ji ponujajo moduli. Kot smo lahko videli je parametrov in nastavitvev dovolj, za določitev in izdelavo digitalnega modela terena po svojih potrebah.

7 ZAKLJUČEK

Osnovni cilji diplomske naloge so bili pregled, analiza in priporočilo za uporabo prostih in odprtokodnih programov, ki omogočajo obdelavo lidarskih podatkov.

K izbiri in analizi, v diplomski nalogi obravnavanih programov, je po mojem mnenju potrebno pristopiti sistematično, saj je potrebno upoštevati več zahtev in želja različnih tipov uporabnikov omenjenih programov. Prav tako je potrebno posvetiti pozornost tudi v proces izbire vpletenim programom, ki imajo tako kot uporabniki svoje zahteve, lastnosti, sposobnosti in omejitve.

Ker obstaja želja po uporabi FOSS programov v pedagoške namene, se je potrebno spoznati z osnovnimi pravili porazdeljevanja in uporabe programov. Odprtokodna skupnost vzpodbuja delitev informacij in znanja med vse ljudi, vendar temu znanju ne odvzema zasluženih avtorskih pravic. Iz tega razloga je potrebno pred uporabo programske opreme v širše pedagoške namene dobro preučiti pravice in dolžnosti uporabnikov. Te pravice določajo licence za prosto in odprtokodno programje. V nalogi sem opisal najpogosteje uporabljene licence in tiste, ki se navezujejo na izbrane programe. V splošnem velja, da omogočajo prosto porazdeljevanje in uporabo v kakršne koli namene, torej tudi pedagoške. Več omejitev in zahtev izražajo pri samem spreminjanju izvorne programske kode.

Za iskanje primerne programske opreme sem izkoristil možnosti, ki jih ponuja internet, saj sta od njega v veliki meri odvisni prepoznavnost in priljubljenost odprtokodnih programskih projektov. S pomočjo različnih brskalnikov, forumov, spletnih arhivov za shranjevanje FOSS programov in domačih spletnih strani projektov sem sestavil seznam programov, za katere sem ugotovil, da na nek način izpolnjujejo zahteve uporabnika za uporabo podatkov, pridobljenih s pomočjo lidarja. Začetni seznam sem najprej skrčil s selektivnimi kriteriji. Z njimi sem določil zahteve posameznega uporabnika in zahteve za uporabo programov v pedagoške namene, torej potrebe fakultete. Programe iz skrčenega seznama sem nato namestil na računalnik, jih preizkusil in vključil v proces ocenitve primernosti, kjer sem določil dodatne, bolj specifične kriterije. Kriterije sem definiriral na osnovi zahtev, katere naj bi uspešen FOSS program izpolnjeval. Nato sem jih točkovno ovrednotil in jim dodelil uteži glede na njihovo pomembnost. Največjo težo sem seveda namenil kriteriju funkcionalnosti, saj je od njega v veliki meri odvisna primernost programov za pedagoški in raziskovalni namen. Rezultati ocenitve so izpostavili programe, ki v odprtokodni skupnosti veljajo za zmogljive pri obdelavi različnih geografskih podatkov. Glede na oceno sta se za najbolj primerna programa izkazala projekta GRASS GIS in SAGA GIS, ki sta vsekakor primerna izbira, predvsem za raziskovalne potrebe. Toda za pedagoški namen, kjer so zahteve in želje uporabnika drugačne v primerjavi z raziskovalnimi, sem se odločil izpostaviti programa FUSION/LDV in SAGA GIS. Oba sta primerna za pedagoške namene. Prvi omogoča po mojem mnenju boljši vpogled v samo obdelavo lidarskih podatkov, slednji pa izpolnjuje zahteve začetnega uporabnika kot tudi raziskovalca.

Za oba priporočena programa sem izdelal kratka navodila za izdelavo DMR-ja, s katerimi želim potencialnega uporabnika seznaniti z osnovami uporabe programa za izdelavo modelov površja.

V nalogi sem na kratko predstavil še preostale programe iz končnega seznama kot tudi določene programe, ki sem jih izločil na samem začetku selekcije. Ti programi niso bili izločeni, ker bi bili neuporabni oz. zaradi njihovih slabosti. Izločeni so bili, ker niso izpolnjevali ene od zahtev naloge, to je uporabe v pedagoške namene. Vseeno pa ti programi omogočajo obdelavo lidarskih podatkov. Nekateri omogočajo le del obdelave, vendar bi jih bilo vredno bolje preučiti in jih uporabiti pri kakšnih bolj specifičnih nalogah.

Z ocenitvijo programov sem želel nakazati, da obstajajo metode, s katerimi lahko uporabniku pomagamo pri odločitvi za določen program. Ker je postopek pridobivanja informacij o posameznem projektu dolgotrajen in zamuden se ponavadi zanj odločajo podjetja, institucije in organizacije, ki želijo namesto lastniških programov uporabljati FOSS programe. Znan je projekt CASCADOSS, ki je podrobno ocenjeval FOSS GIS programe z namenom seznaniti potencialne uporabnike z vlogo in možnostmi za uporabo prostorskih odprtokodnih projektov.

Brez sistematičnega pristopa bi lahko izbral program popolnoma instinktivno, ga uporabil, vendar bi na koncu lahko ugotovil, da ne ustreza cilju diplomske naloge. Kot sem omenil, je načinov in modelov ocene primernosti programov za določeno uporabo več. Obstajajo subjektivne kot tudi objektivne metrične metode ocenjevanja primernosti programske opreme. V tej nalogi sem združil obe ideji odločanja, saj je vrednotenje programov bilo namenjeno lažji izbiri primernosti programa za končno uporabo in izdelavo DMR-ja.

Menim, da bi bilo smiselno področju proste in odprte kode v prihodnosti nameniti več pozornosti, saj ima v primerjavi z lastniškimi programi določene prednosti (finančni vidik, hitrejši razvoj, globalna razvijalska skupnost, itd.). Vsekakor je prednost uporabe FOSS programov v pedagoške namene v primerjavi z lastniškimi ta, da si lahko študent namesti program na svoj osebni računalnik. S tem se lahko bolj poglobljeno posveti spoznavanju in uporabi programa, saj ni časovno in prostorsko omejen z računalniškimi učilnicami na fakulteti.

LITERATURA IN VIRI

American Society for Photogrammetry and Remote Sensing, (ASPRS). 2009. LAS Specification version 1.3 – R10

http://www.asprs.org/a/society/committees/standards/LAS_1_3_r11.pdf (22. 5. 2011).

Apache License, Version 2.0

<http://www.apache.org/licenses/> (20. 3. 2011).

ASPRS Standards Committee, LASer (LAS) File Format Exchange Activities. 2011.

http://www.asprs.org/a/society/committees/standards/lidar_exchange_format.html (15. 5. 2011).

Bajcer, D. 2007. Ali lahko odprtokodni sistemi za upravljanje spletnih vsebin uspešno podprejo podjetniške iniciative, ki temeljijo na novih spletnih poslovnih modelih. Diplomski naloga. Ljubljana, Univerza v Ljubljani, Ekonomska fakulteta: 46 f.

http://www.cek.ef.uni-lj.si/u_diplome/bajcer2817.pdf (14. 12. 2010).

Basili, V. R., Caldiera, B., Rombach, H. D. 1994. Goal Question Metric Approach. Encyclopedia of Software Engineering. John Wiley & Sons, Inc.: str. 528-532.

<ftp://ftp.cs.umd.edu/pub/sel/papers/gqm.pdf> (22. 2. 2011).

Bitenc, M. 2006. Analiza podatkov in izdelkov zračnega laserskega skeniranja na projektu Neusiedler See. Diplomski naloga. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo: 123 f.

CASCADOSS. 2007. Evaluation criteria for open source GIS and RS software. Development of transnational cascade training programme on Open Source GIS&RS Software for environmental applications.

<http://www.cascadoss.eu/en/> (19. 1. 2011).

Chauve, A., Bretar, F., Durrieu, S., Pierrot – Deseilligny, M., Puech, W. 2009. Fullanalyze: A research tool for handling, processing and analyzing full-wave-form lidar data.

http://recherche.ign.fr/labos/matis/pdf/articles_conf/2009_chauve_igarss.pdf (23. 3. 2011).

Chen, D., Carmona – Moreno, C., Leone, A., Shams, S. 2008. Assessment of open source GIS software for water resources management in developing countries. JRC Scientific and Technical Reports (EUR collection).

http://publications.jrc.ec.europa.eu/repository/bitstream/111111111/10555/1/reqno_jrc49291_version%20approved%20by%20opoce.pdf (20. 1. 2011).

ESRI, GIS Dictionary

<http://support.esri.com/en/knowledgebase/GISDictionary/search> (20. 5. 2011).

Freshmeat

<http://freshmeat.net/> (21. 3. 2011).

FreeBSD license

<http://www.freebsd.org/copyright/freebsd-license.html> (18. 12. 2010).

Free Software Foundation (FSF)

<http://www.fsf.org/> (10. 12. 2010).

GNU Operating System, Philosophy of the GNU Project

<http://www.gnu.org/philosophy/> (12. 12. 2010).

GNU Operating System, GNU Lesser General Public License

<http://www.gnu.org/copyleft/lesser.html> (17. 12. 2010).

GNU Operating System, The BSD License Problem

<http://www.gnu.org/philosophy/bsd.html> (18. 12. 2010).

GNU Operating System, The Free Software Definition

<http://www.gnu.org/philosophy/free-sw.html> (12. 12. 2010).

GRASS GIS

<http://grass.osgeo.org/> (24. 5. 2011).

gvSIG

<http://www.gvsig.org/web/> (25. 5. 2011).

islovar

<http://www.islovar.org/> (21. 11. 2010)

Kocmut, M. 2009. Stiskanje podatkov LiDAR. Diplomaska naloga. Maribor, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko: 51 f.

Kovač, B. 2008. Vizualizacija velikih količin podatkov LiDAR. Diplomaska naloga. Maribor, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko: 58 f.

Kraus, K., Pfeifer, N. 1998. Determination of terrain models in wooded areas with airborne laser scanner data. ISPRS Journal of Photogrammetry and Remote Sensing 53,4: 193-203.
<http://www.martinkodde.nl/glacier/data/bibliography/2526743040kraus.pdf> (24. 5. 2011).

Liu, X. 2008. Airborne LiDAR for DEM generation: some critical issues. Progress in Physical Geography 32,1: 31-49.
<http://ppg.sagepub.com/content/32/1.toc> (14. 4. 2011).

MapWindow

<http://www.mapwindow.org/> (25. 5. 2011).

McGaughey, R. J., 2010. FUSION/LDV Software for LiDAR data analysis and visualization. Priročnik k programu FUSION/LDV verzije 2.90
http://forsys.cfr.washington.edu/fusion/FUSION_manual.pdf (13. 4. 2011).

Michlmayr, M. 2005. Software process maturity and the success of free software projects. Zieliński, K. (ur.), Szmuc, T. (ur.), Software Engineering: Evolution and Emerging Technologies 130: 3-14.
http://www.cyrius.com/publications/michlmayr-process_maturity_success.pdf (15. 1. 2011).

Mlakar, M. 2009. Uporaba odprte kode v okoljih GIS. Diplomaska naloga. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo: 72 f.

Mozilla Public License

<http://www.mozilla.org/MPL/> (9. 1. 2011).

OpenGL

<http://www.opengl.org/> (28. 5. 2011).

Open Source Geospatial Foundation

<http://www.osgeo.org/> (24. 5. 2011).

Open Source Initiative

<http://www.opensource.org/> (8. 12. 2010).

Open Source Initiative, GNU General Public License, version 3 (GPL-3.0)

<http://www.opensource.org/licenses/gpl-3.0.html> (16. 12. 2010).

Open Source Initiative OSI, The BSD License: Licensing, The BSD 2-Clause License

<http://www.opensource.org/licenses/bsd-license.php> (17. 12. 2010).

Open Source Initiative OSI, The MIT License (MIT): Licensing, The MIT License (MIT)

<http://www.opensource.org/licenses/mit-license.php> (18. 12. 2010).

OSS Watch; open source software advisory service, Top tips for selecting open source software

<http://www.oss-watch.ac.uk/resources/tips.xml> (19. 3. 2011).

Pečnik, S. 2009. Interaktivno orodje za obdelavo podatkov LiDAR. Diplomaska naloga. Maribor, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko: 56 f.

Perens, B. 1999. The Open Source Definition

<http://oreilly.com/catalog/opensources/book/perens.html> (12. 1. 2011).

Primožič, P. 2005. Uporaba odprte kode kot osnova za razvoj programske opreme. Diplomaska naloga. Ljubljana, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko: 71 f.

http://eprints.fri.uni-lj.si/165/1/PP_diploma.pdf (14. 12. 2010).

Ramsey, P. 2007. The state of open source GIS.

<http://www.refrations.net/expertise/whitepapers/opensourcesurvey/survey-open-source-2007-12.pdf>
(13. 1. 2011).

Raymond, E. S. 2002. The Cathedral and the Bazaar.

<http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html> (30. 3. 2011).

SAGA GIS, System fo Automated Geoscientific Analysis

<http://www.saga-gis.org/> (15. 9. 2010).

Shan, J., Toth, C.K.. (ur.). 2009. Topographic laser ranging and scanning. Principles and processing. New York, Taylor & Francis Group: 590 str.

Sourceforge

<https://sourceforge.net/> (21. 3. 2011).

Steiniger, S., Bocher, E. 2008. An overview on current free and open source desktop GIS developments. International Journal of Geographical Information Science 23,10: 1345-1370.

http://terragis.net/docs/presentations/sstein_foss_desktop_gis_overview.pdf (15. 2. 2011).

Szejko, S., 2002. Requirements driven quality control

<http://www.cse.dmu.ac.uk/COMPSAC/wimpe/secretpath/scratch/papers/paper.123.pdf> (2. 3. 2011).

Škrubej, U. 2009. Uporaba odprtokodnih rešitev v spletnih trgovinah malih podjetij. Magistrsko delo. Maribor, Univerza v Mariboru, Ekonomsko – poslovna fakulteta: 79 f.

<http://www.epf.uni-mb.si/ediplome/pdfs/skrubej-uros-mag.pdf> (14. 12. 2010).

Šumrada, R. 2005. Strukture podatkov in prostorske analize. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo: 284 str.

The Linux Information Project

<http://www.linfo.org/mitlicense.html> (18. 12. 2010).

Zalokar, M. 2011. Trirazsežno modeliranje zgradb iz lidarskih podatkov na primeru mesta Domžale. Diplomaska naloga. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo: 52 f.

van den Berg, K. 2005. Finding open options. An open source software evaluation model with a case study on course management systems. Master thesis. Tilburg, Tilburg University, Information Management and Science program: 92 f.

<http://www.karinvandenberg.nl/Thesis.pdf> (17. 2. 2011).

Wheeler, D. A. 2010. How to Evaluate Open Source Software / Free Software (OSS/FS) Programs

http://www.dwheeler.com/oss_fs_eval.html (10. 3. 2011).

PRILOGE

Priloga A: Vprašanja za izpolnitev kriterijev in njihovo točkovno ovrednotenje

Priloga A.1: Vprašanja in rezultati vrednotenja kriterija Zrelost projekta

P – pedagoški namen, R – raziskovalni namen, X – vprašanje se za ta namen ne ocenjuje

Št.	Vprašanje	Ovrednotenje	P	R
1	Koliko časa je preteklo od prve pojave produkta na trgu?	0 – manj kot 1 leto 1 – od 1 do 5 let 2 – več kot 5 let		
2	Ali je minilo več kot dve leti od zadnje stabilne verzije programa?	0 – da 1 – ne		
3	Ali obstaja baza pogostih vprašanj (FAQ)?	0 – ne 1 – da		
4	Ali projekt uporablja VCT?	0 – ne 1 – da	X	

Zrelost projekta – točkovno ovrednotenje

Vprašanje / Program	FUSION / LDV	gvSIG	ILWIS	MapWindow	QuikGrid	Whitebox GAT	GRASS	SAGA	QGIS
1	2	2	2	2	2	1	2	2	2
2	1	1	1	1	1	1	1	1	1
3	1	0	1	1	1	0	1	1	1
4	1	1	0	0	0	0	1	1	1
Skupaj P	4	3	4	4	4	2	4	4	4
Skupaj R	5	4	4	4	4	2	5	5	5

Priloga A.2: Vprašanja in rezultati vrednotenja kriterija Dokumentacija

P – pedagoški namen, R – raziskovalni namen, X – vprašanje se za ta namen ne ocenjuje

Št.	Vprašanje	Ovrednotenje	P	R
5	Ali program vsebuje uporabniški priročnik?	0 – ne 1 – da		
6	Ali obstaja zadovoljiv opis orodij oz. funkcionalnosti za obdelavo lidarskih podatkov?	0 – ne 1 – da		
7	Ali obstaja priročnik za namestitev programa?	0 – ne 1 – da		
8	Ali program vsebuje datoteke pomoči (help files)?	0 – ne 1 – da		
9	Ali obstajajo vadnice za obdelavo lidarskih podatkov?	0 – ne 1 – da		
10	Ali je na voljo zunanja dokumentacija (knjige, članki, poročila, ipd.)?	0 – ne 1 – da	X	
11	Ali je preko spleta omogočen direkten dostop do izvorne kode in delov izvorne kode?	0 – ne 1 – da	X	
12	Ali obstajajo dokumenti, ki opisujejo, kako biti udeležen pri razvoju projekta in programske opreme?	0 – ne 1 – da	X	
13	Ali obstaja dokumentacija za razvijanje programa (razvijalski FAQ, datoteke pomoči za razvijalce, ipd.)?	0 – ne 1 – da	X	

Dokumentacija - točkovno ovrednotenje

Vprašanje / Program	FUSION / LDV	gvSIG	ILWIS	MapWindow	QuikGrid	Whitebox GAT	GRASS	SAGA	QGIS
5	1	1	1	1	0	1	1	1	1
6	1	1	1	1	1	1	1	1	0
7	1	1	1	0	1	1	1	1	1
8	0	0	1	1	1	1	1	1	1
9	1	0	1	0	0	1	1	0	0
10	0	0	0	1	0	1	1	1	1
11	0	1	1	1	1	1	1	1	1
12	0	1	1	1	0	1	1	1	1
13	0	1	0	1	0	1	1	1	1
Skupaj P	4	3	5	3	3	5	5	4	3
Skupaj R	4	6	7	7	4	9	9	8	7

Priloga A.3: Vprašanja in rezultati vrednotenja kriterija Skupnost in podpora

P – pedagoški namen, R – raziskovalni namen, X – vprašanje se za ta namen ne ocenjuje

Št.	Vprašanje	Ovrednotenje	P	R
14	Ali je na spletni strani projekta skupnost jasno vidna?	0 – ne 1 – da		
15	Ali obstaja poštni seznam ali forum za uporabnike?	0 – ne 1 – da		
16	Ali obstaja poštni seznam ali forum za razvijalce?	0 – ne 1 – da	X	
17	Ali je razvijalski poštni seznam / forum javen?	0 – ne 1 – da	X	
18	Ali obstaja arhiv poštnega seznama / foruma?	0 – ne 1 – da		
19	Povprečno število objav (angl. post) na forumu ali poštmem seznamu (zadnjih 6 mesecev)?	0 – manj kot 5 1 – med 5 in 50 2 – več kot 50		
20	Število objavljenih tem (angl. topic) na forumu ali poštmem seznamu v zadnjih 6 mesecih?	0 – manj kot 5 1 – več kot 5		
21	Ali pri projektu sodeluje več razvijalcev?	0 – en razvijalec 1 – več razvijalcev	X	
22	Ali je jasno kdo sestavlja razvijalsko ekipo?	0 – ne 1 - da	X	
23	Ali obstaja drugačen način podpore uporabi, ki ni forum oz. poštni seznam (npr. neposredno po mailu)?	0 – ne 1 - da		
24	Ali obstaja podpora ob napakah?	0 – ne 1 - da		
25	Koliko institucij podpira / sodeluje pri projektu?	0 – nobena 1 – ena 2 – dve ali več		
26	Ali je jedro razvijalske ekipe pri delu finančno podprto s strani organizacij?	0 – ne 1 - da	X	

Skupnost in podpora - točkovno ovrednotenje

Vprašanje / Program	FUSION / LDV	gvSIG	ILWIS	MapWindow	QuikGrid	Whitebox GAT	GRASS	SAGA	QGIS
14	0	1	1	1	0	0	1	1	1
15	0	1	1	1	0	0	1	1	1
16	0	1	0	1	0	0	1	1	1
17	0	1	0	1	0	0	1	0	1
18	0	1	1	1	0	0	1	1	1
19	0	2	1	2	0	0	2	2	2
20	0	1	1	1	0	0	1	1	1
21	0	1	1	1	0	1	1	1	1
22	1	1	1	1	1	1	1	1	1
23	1	1	1	1	1	0	1	1	0
24	1	0	0	1	1	0	1	1	1
25	2	2	2	2	0	1	2	1	2
26	1	1	1	1	0	1	1	1	1
Skupaj P	4	9	8	10	2	1	10	9	9
Skupaj R	6	14	11	15	3	4	15	13	14

Priloga A.4: Vprašanja in rezultati vrednotenja kriterija Prepoznavnost

P – pedagoški namen, R – raziskovalni namen, X – vprašanje se za ta namen ne ocenjuje

Št.	Vprašanje	Ovrednotenje	P	R
27	Ali je projekt omenjen v Wikipediji?	0 – ne 1 – da		
28	Koliko relevantnih / ustreznih / primernih zadetkov o projektu najdemo v brskalniku kot je npr. Google?	0 – manj kot 10 1 – več kot 10		
29	Ali je statistika popularnosti vodena na spletni strani freshmeat.net?	0 – ne 1 – da		
30	Ali je aktivna statistika projekta vodena na strani sourceforge.net?	0 – ne 1 – da		

Prepoznavnost – točkovno ovrednotenje

Vprašanje / Program	FUSION / LDV	gvSIG	ILWIS	MapWindow	QuikGrid	Whitebox GAT	GRASS	SAGA	QGIS
27	0	1	1	1	0	1	1	1	1
28	1	1	1	1	1	1	1	1	1
29	0	1	0	0	0	0	1	1	1
30	1	1	0	1	0	0	1	1	1
Skupaj P	2	4	2	3	1	2	4	4	4
Skupaj R	2	4	2	3	1	2	4	4	4

Priloga A.5: Vprašanja in rezultati vrednotenja kriterija Integracija

P – pedagoški namen, R – raziskovalni namen, X – vprašanje se za ta namen ne ocenjuje

Št.	Vprašanje	Ovrednotenje	P	R
31	Ali program omogoča razvoj in dodajanje modulov in dodatkov za potrebe uporabnika?	0 – ne 1 – da	X	
32	Ali obstaja jasna metoda za dodajanje novih funkcionalnosti, ki ne spremeni jedra programa?	0 – ne 1 – da	X	
33	Ali obstaja knjižnica (arhiv) že prispevanih dodatkov, ki jih vodi uporabniška ali razvijalska skupnost? (arhiv)	0 – ne 1 – da		
34	Ali produkt uporablja ali izvaja sorodne mednarodne standarde (OGC, ISO)?	0 – ne 1 – da		
35	Ali produkt omogoča prenos podatkov iz/v drug softverski produkt?	0 – ne 1 – da		
36	Ali ima program posebne programske zahteve?	0 – da 1 – ne		
37	Na koliko platformah (operacijski sistem) lahko program deluje?	0 – na 1 1 – na 2 2 – 3 in več	X	
38	Ali program omogoča izbiro operacijskega sistema za inštalacijo?	0 – ne 1 – da	X	

Integracija – točkovno ovrednotenje

Vprašanje / Program	FUSION / LDV	gvSIG	ILWIS	MapWindow	QuikGrid	Whitebox GAT	GRASS	SAGA	QGIS
31	0	1	1	1	0	1	1	1	1
32	0	1	0	1	0	1	1	1	1
33	1	1	0	1	0	0	1	1	1
34	1	1	1	1	1	1	1	1	1
35	1	1	1	1	1	1	1	1	1
36	1	0	0	1	1	0	1	1	1
37	2	2	2	0	2	1	2	2	2
38	0	1	1	0	0	0	1	1	1
Skupaj P	4	3	2	4	3	2	4	4	4
Skupaj R	6	8	6	6	5	5	9	9	9

Priloga A.6: Vprašanja in rezultati vrednotenja kriterija Funkcionalnost

P – pedagoški namen, R – raziskovalni namen, X – vprašanje se za ta namen ne ocenjuje

Št.	Vprašanje	Ovrednotenje	P	R
39	Ali lahko v program uvozimo podatke v LAS formatu?	0 – ne 1 – da		
40	Ali lahko v program uvozimo podatke v ASCII formatu (txt, asc)?	0 – ne 1 – da		
41	Ali lahko iz programa izvozimo podatke v formatih, ki jih podpirajo ostali FOSS/GIS programi (shp, asc, grd, dem, txt, ...)?	0 – ne 1 – da		
42	Ali program omogoča filtriranje oblaka točk?	0 – ne 1 – da		
43	Ali program omogoča interpolacijo točk?	0 – ne 1 – da		
44	Ali program omogoča izdelavo mrež (TIN, kvadratna, ipd.)?	0 – ne 1 – da		
45	Ali je možen prikaz oblaka točk?	0 – ne 1 – da		

Funkcionalnost – točkovno ovrednotenje

Vprašanje / Program	FUSION / LDV	gvSIG	ILWIS	MapWindow	QuikGrid	Whitebox GAT	GRASS	SAGA	QGIS
39	1	0	0	1	0	1	1	1	0
40	1	1	1	1	1	1	1	1	1
41	1	1	0	1	1	1	1	1	1
42	1	0	0	1	0	1	1	1	0
43	1	1	1	1	0	1	1	1	1
44	1	1	0	1	1	1	1	1	1
45	1	0	0	0	0	0	1	1	0
Skupaj P	7	4	2	6	3	6	7	7	4
Skupaj R	7	4	2	6	3	6	7	7	4

Priloga A.7: Vprašanja in rezultati vrednotenja kriterija Uporabniški nivo

P – pedagoški namen, R – raziskovalni namen, X – vprašanje se za ta namen ne ocenjuje

Št.	Vprašanje	Ovrednotenje	P	R
46	Ali je namestitev za neizkušenega uporabnika enostavna?	0 – zahtevna 1 – enostavna		X
47	Za kakšnega uporabnika je program primeren?	0 – izkušen 1 – neizkušen		X
48	Ali ima produkt grafični uporabniški vmesnik (GUI)?	0 – ne 1 – da		X
49	Ali obstaja možnost za prilagoditev programa (plug-in, dodajanje novih modulov...)?	0 – ne 1 – da		X

Uporabniški nivo – točkovno ovrednotenje

Vprašanje / Program	FUSION / LDV	gvSIG	ILWIS	MapWindow	QuikGrid	Whitebox GAT	GRASS	SAGA	QGIS
46	1	1	1	1	1	1	1	1	1
47	1	0	1	1	1	1	1	1	1
48	1	1	1	1	1	1	1	1	1
49	0	1	1	1	0	0	1	1	1
Skupaj P	3	3	4	4	3	3	4	4	4