



UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Damir Balija

LiDAR kot spletna storitev

DIPLOMSKO DELO
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: dr. Andrej Brodnik

Ljubljana, 2011

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .



Št. naloge: 00098/2011

Datum: 04.04.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DAMIR BALIJA**

Naslov: **LIDAR KOT SPLETNA STORITEV**
LIDAR AS A WEB SERVICE

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Naprava LiDAR (Light Detection And Ranging) omogoča, podobno kot radar, merjenje razdalj. Pri tem se izkaže, da je celo zmogljivejša, saj lahko zaznava vrsto in velikost delcev, do katerih meri razdaljo. Na primer z napravo LiDAR lahko merimo kakovost zraka oziroma vsebnost trdnih delcev v njem na različnih razdaljah.

Z napravo LiDAR upravlja lidarski strežnik. V nalogi definirajte način upravljanja z lidarskim strežnikom, ki bo omogočal naročanje storitve meritev (LiDARski programa) ob določenem času na določenem mestu in zajem podatkov z naprave ob izvajanju programa. Na odjemalski strani začrtajte in implementirajte aplikacijo, ki bo poleg implementacije zgornjega dela s strežnikom omogočala hranjenje zajetih podatkov v standardiziranem LAS formatu, upravljanje s tako z zajetimi podatki in upravljanje z LiDARskimi programi. Vaša rešitev naj bo zasnovana na SOA paradigmi.

Mentor:

doc. dr. Andrej Brodnik

Dekan:

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Damir Baliija,

z vpisno številko 63070217,

sem avtor diplomskega dela z naslovom:

LiDAR kot spletna storitev

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Andrej Brodnik
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 23. 09. 2011

Podpis avtorja/-ice:

Zahvala

Zahvalo namenjam najprej mentorju, dr. Andreju Brodniku za vsestransko podporo, nasvete ter strokovno pomoč pri izdelavi diplomske naloge. Prav tako se zahvaljujem staršem, ki so mi študij omogočili in me v času študija podpirali. Na koncu bi se rad zahvalil še bratu, puncu, prijateljem in vsem za izkazano podporo, potrpežljivost in razumevanje med časom študija ter nastankom te diplomske naloge.

Najlepša hvala vsem!

Kazalo

Povzetek	1
Abstract	3
1 Uvod	5
1.1 Namen diplomskega dela	6
1.2 Očem varni LiDAR	7
2 Teoretične osnove za uresničitev cilja	11
2.1 Spletna storitev	11
2.1.1 SOAP	12
2.1.2 WSDL	15
2.2 XML	17
2.2.1 Sintaksa dokumenta XML	17
2.2.2 Vmesnik za procesiranje in razčlenjevanje XML	18
2.2.2.1 Objektni model dokumenta – DOM	19
2.3 Datotečni format LAS	19
2.4 Določanje položaja točk na Zemlji	23
2.4.1 Oblika in dimenzija zemlje	23
2.4.2 Koordinatni sistemi	25
2.4.2.1 Pravokotni koordinatni sistem v ravnini	26
2.4.2.2 Polarni koordinatni sistem v ravnini	27
2.4.2.3 Geografske koordinate na Zemlji kot krogli	28
2.4.3 Državni koordinatni sistem	28
2.4.3.1 Višinski državni koordinatni sistem	29
2.4.3.2 Gauss-Kruegerjeva projekcija	29
2.4.3.3 Projekcija TM	30
2.4.4 Metode določanja položaja točk	31
2.4.4.1 Absolutna določitev položaja	31

2.4.4.2	Relativna določitev položaja	31
2.4.4.3	Vincentyjeve formule	32
2.5	LiDARski lokalni koordinatni sistem	34
3	Programska oprema RhoLaaS za delo z LiDARskim sistemom	37
3.1	Uvod	37
3.2	Meritev, program in rezultat meritve	38
3.3	Scenarij (primer) uporabe protokola	40
3.4	Protokol povezave z LiDARskim strežnikom	41
3.4.1	Namestnik - Proxy	41
3.4.2	Spletna storitev SOAP	42
3.4.3	Merilni kanal	43
3.5	Opis in izvedba aplikacije RhoLaaS	46
3.5.1	Osnovno okno aplikacije	46
3.5.1.1	Arhiv programov meritev	51
3.5.1.2	Lastnosti glave datoteke formata LAS	53
3.5.1.3	Lastnosti točke iz datoteke formata LAS	55
3.5.2	Merjenje (tipi programov meritev)	56
3.5.2.1	Meritev v pravokotnem območju	56
3.5.2.2	Meritev v rezini	59
3.5.2.3	Hitra, osnovna meritev	60
3.5.2.4	Prikaz geografskih podatkov iz programa meritev	61
3.5.3	Vizualizacija meritev	61
3.5.3.1	Graf intenzitete	63
3.5.3.2	Graf v rezini	65
3.5.3.3	3D prikaz točk	66
4	Sklepne ugotovitve	69
4.1	Možnosti za nadaljnji razvoj in delo	70
A	Priloge	72
A.1	Tabele formatov zapisov točk v datoteki LAS	72
A.2	Tabela glave datoteke formata LAS	78
A.3	Opis elementov shranjenega programa meritve v formatu XML	79
	Seznam slik	80
	Seznam tabel	82

Seznam uporabljenih kratic in simbolov

Kratica	Pomen
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ASPRS	American Society of Photogrammetry and Remote Sensing
DOM	Document Object Model
ESRS	European Spatial Reference System
ETRS	European Terrestrial Reference System
EVLN	Extended Variable Length Record
EVRS	European Vertical Reference System
GIS	Geographic Information System
GPS	Global Positioning System
GRS	Geodetic Reference System
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
IR	Infrared
Laas	LiDAR as a Service
LAS	Lidar data exchange format
LIDAR	Light Detection And Ranging
RPC	Remote Procedure Call
SAX	Simple API for XML
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SOR	Start Of Record
TCP	Transmission Control Protocol
TOR	Type Of Record
UDDI	Universal Description, Discovery, and Integration

Kratika**Pomen**

URL

Uniform Resource Locators

UTM

Universal Transverse Mercator Projection

UUID

Universal Unique Identifier

UV

Ultraviolet

VLR

Variable Length Records

WGS

World Geodetic System

WS

Web Service

WSDL

Web Services Description Language

XML

eXtensible Markup Language

Povzetek

LiDAR (*Light Detection And Ranging*) je naprava, ki je po načinu delovanja podobna radarju, le da namesto radijskih valov uporablja svetlobo. Uporabimo ga lahko za detekcijo delcev v zraku, primarno različnih onesnaževalcev ali za merjenje vremenskih pojavov. LiDAR skupaj s podpornimi elementi tvori LiDARski sistem. V našem primeru bo uporaba LiDARskega sistema vezana na oddaljeno uporabo, in sicer na oddaljenega uporabnika.

- **Cilji:**

- (a) Načrtovati in zasnovati protokol povezave med oddaljenim odjemalcem (stranko) in LiDARsko napravo (strežnikom).
- (b) Razviti programsko opremo za oddaljeni nadzor LiDARske naprave, pošiljanje programov meritev, zajemanje, prikaz in shranjevanje programov meritev ter podatkov v standardnem formatu za shranjevanje podatkov takega tipa (LAS format).

- **Metode:**

Razvoj temelji na tehnološko modernemu razvojnemu okolju SOAP (*Simple Object Access Protocol*) in programskem jeziku Java, ki ga uporabimo za razvoj uporabniku prijaznega grafičnega vmesnika. Kot zahtevo za meritev oddaljeni uporabnik pripravi program meritve, ki ga kasneje pošlje LiDARski napravi (strežniku). Program vsebuje vse podatke, ki jih LiDARska naprava potrebuje za zajem podatkov iz njene okolice. LiDARska naprava s podatki iz prejetega programa izvede meritev, zajeti podatki pa so poslani nazaj oddaljenemu uporabniku. Za prikaz pridobljenih podatkov iz LiDARskega sistema uporabimo različne grafične kontrole.

- **Rezultati:**

Programski grafični vmesnik oz. aplikacija, katera s pomočjo protokola, ki ureja povezavo med oddaljenim odjemalcem (stranko) in LiDARsko napravo (strežnikom), omogoča naročanje storitve meritev (LiDARski program) ob določenem času na določenem mestu in zajem podatkov z naprave ob izva-
janju programa. Aplikacija omogoča tudi shranitev zajetih podatkov v stan-
dardnem formatu LAS, s pomočjo različnih grafičnih kontrol pa je podatke
možno tudi na pregleden način prikazati.

Implementirana aplikacija je namenjena upravljanju mobilnega LiDAR-
skega sistema, ki se ga namesti v avto ali prikolico, s čimer aplikacija tako
postane del mobilnega nadzornega okoljskega sistema. LiDARski sistem se bo
primarno uporabljal za zajem različnih vremenskih podatkov in podatkov o
kvaliteti zraka z namenom, da bi natančneje določili izvore onesnaževanja in
jih kvantitativno ovrednotili.

Ključne besede:

LiDAR, spletna storitev, onesnaženje zraka

Abstract

LiDAR(Light Detection And Ranging) is a device, which works in a similar way as radar, but instead of radio waves, it uses light. We can use it as a mean to detect particles in the air (mainly air pollution particles) or for measuring weather conditions. LiDAR with other supporting elements forms LiDAR system. In our example LiDAR system will be used remotely, in particular by remote user.

- **Goals:**

- (a) To plan and project protocol for exchanging messages between client (user, costumer) and LiDAR system (server).
- (b) To develop software for remote control of LiDAR device. This includes sending measurements programmes, capturing, presenting and storing measurements programmes and data in standard exchange format for this type of data (LAS format).

- **Methods:**

Software development is based programming language Java including graphical user interface, which is using Java's Swing library. Communication between server and client is arranged with custom developed protocol, which is based on SOAP protocol (Simple Object Access Protocol). Other Web Services technologies are also used.

- **Results:**

The outcome of project is GUI based software gateway (application) for using mobile LiDAR system. With this application user is able to prepare program of measurements, which is later send to the LiDAR device (server) as a request. Programme contains all the data needed for the LiDAR device, to carry out capturing data from its environment. LiDAR device on the specific place and time carries out measurements, according to received programme. Captured data, as a result of this measurements, are send back to the remote user, which can further manipulate with them. Application enables him to store data in standard LAS format and he can also visualize captured data with various graphical widgets. Application is meant to control mobile LiDAR system, which will be installed in car or trailer. In this way the application will become part of mobile control environmental system. LiDAR system will be primarily used as a mean to capture different kinds of weather data and data related to the quality of air. The purpose of this collecting is to accurately detect origins of pollution and to qualitatively evaluate collected data.

Key words:

LiDAR, web service, air pollution

Poglavje 1

Uvod

Emisije toplogrednih plinov in aerosolov (trda ali tekoča snov, manjši ali večji delci, na primer žveplov dioksid, saje, pepel), s katerimi onesnažujemo ozračje, spreminjajo sestavo ozračja, kar vpliva na toplotno bilanco Zemlje. Porast toplogrednih plinov povzroča učinek tople grede in posledično globalno segrevanje. Toplogredni plini so [1]:

- vodna para (H_2O - narašča z višanjem temperature površine),
- ogljikov dioksid (CO_2 - se sprošča pri naravnih procesih v rastlinskem in živalskem svetu),
- ozon (O_3 - nastaja pri fotokemičnih reakcijah),
- metan (CH_4 - nastaja z razpadanjem organskih snovi v okolju brez kisika),
- didušikov oksid (N_2O - prihaja iz prsti in oceanov),
- fluorirani ogljikovodiki (HFC), perfluorirani ogljikovodiki (PFC) in žveplov heksafluorid (SF_6) (novi industrijski plini).

Po drugi strani pa so aerosoli krivci tudi za globalno mračenje (uporablja se tudi termin senčenje). Aerosoli namreč odbijajo sončno svetlobo in toploto nazaj v vesolje in tako spreminjajo lastnosti oblakov (povečujejo število kapljic v oblakih, ki so zato manjši, in posledično odbijejo več sončnih žarkov nazaj v vesolje). Globalno mračenje prikriva dejansko moč globalnega segrevanja in če bi odpravili globalno mračenje, bi to pomenilo še hitrejše segrevanje Zemlje. In čeprav na prvi pogled zglada, da je globalno mračenje v resnici koristno, onesnaževalci zraka hkrati povzročajo tudi številne probleme za človeka in

okolje, kot so smog, težave z dihanjem, kisel dež in spremembe padavinskih vzorcev [2].

Za natančno oceno vsebnosti onesnaževanja v ozračju in na zemeljskem površju in z namenom zmanjšanja kratkoročnih in dolgoročnih negativnih posledic za okolje, je potrebno izvesti ustrezne meritve, s primerno prostorsko in časovno ločljivostjo. Način, ki ga predlagamo, je meritev na daljavo z uporabo LiDARja [3]. LiDAR je naprava, ki v osnovi deluje podobno kot radar, le da namesto radijskih valov uporablja svetlobo (manjša valovna dolžina), kar nam omogoča zaznavanje manjših delcev - aerosolov v zraku. Naprava s pomočjo laserja oddaja kratke svetlobne sunke, sprejme pa jih s pomočjo detektorja. Tako lahko analizira sipani del odbite svetlobe in iz nje izmeri koncentracijo različnih plinov (H_2O , CO_2 , CH_4 , NO_x , O_3 , itd.) ter aerosolov v ozračju [4].

1.1 Namen diplomskega dela

Diplomska naloga spada v kontekst večih projektov. Prvi je projekt z naslovom »Razvoj prenosnega, očem varnega LiDARja za nadzor onesnaževanja ozračja«, katerega nosilec je Univerza na Primorskem, Primorski inštitut za naravoslovne in tehnične vede (UP PINT) in projekta »Obdelava velikih količin geometrijskih podatkov LIDAR«, katerega nosilec je Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko v sodelovanju z UP PINT [5].

Osnovna struktura željenega sistema se je z razvojem precej spreminjala, vendar je primarni cilj ostal nespremenjen, torej izdelava sistema, ki bo omogočal oddaljeni nadzor in uporabo LiDARskega sistema za zajemanje, prikaz in shranjevanje podatkov za nadaljnjo analizo (programi meritev in datoteke LAS).

Da bi lahko omogočili oddaljeni nadzor smo LiDARski sistem implementirali kot spletno storitev. Za lažjo predstavbo: končni izdelek deluje tako, da pooblaščen stranka z dostopanjem preko spleta in s pomočjo aplikacije za oddaljen nadzor ter uporabo LiDARskega sistema, pripravi program meritve in zahtevo pošlje LiDARskemu strežniku. LiDARski strežnik preko spletnih storitev omogoča upravljanje LiDARske naprave, pri čemer so predhodno potrebni avtentikacija, preverjanje zasedenosti LiDARske naprave in vzpostavitev povezave za prenos izmerjenih podatkov (merilni kanal). Kot odgovor na zahtevo nato strežnik vrne odjemalcu odgovor kje in kako dobiti rezultate meritev (merilni kanal). Lidar nato obvesti pooblaščen osebo, da je prejel program meritve. Pooblaščen oseba poskrbi, da bo LiDAR ob določenem času

na željenem kraju in ob tej uri se odjemalec (stranka) poveže na LiDARski strežnik. Medtem, ko LiDAR izvaja meritev, odjemalec iz merilnega kanala zajema izmerjene podatke. Ko stranka prejme vse podatke, jih s pomočjo grafičnih orodij vizualizira ali pa jih shrani v binarno datoteko LAS (oblika tovarniškega standarda za shranjevanje podatkov pridobljenih iz LiDARja), za kasnejšo obdelavo oz. analizo.

Med izvajanjem diplomske naloge je LiDAR doživel nadgradnjo strojne opreme in sicer je dobil nov gibalni modul, ki je precej hitrejši od starega. Star sistem za krmiljenje LiDARja je uporabljal dva koračna motorja, ki sta bila mehansko vezana na pogonske osi preko reduktorjev zato je bilo premikanje LiDARja počasno, delo pa težavno. Da smo vsaj malo omilili problem počasnosti, smo si naprej pomagali z grobo ročno nastavitvijo.

Zaradi potrebe po implementaciji oddaljene uporabe LiDARskega sistema, si nismo smeli več privoščiti ročnih premikov oz. grobe ročne nastavitve pozicije, zato je bil nakup novega gibalnega modula nujen. S tem smo pridobili tudi na stabilnosti, praktičnosti in uporabnosti končnega izdelka in tako omogočili, da bo LiDARski sistem, čim bolje služil uporabnikom pri njihovem poslanstvu ohranjanja čistega in zdravega okolja.

1.2 Očem varni LiDAR

LiDAR (angl. *Light Detection And Ranging*) je naprava, ki s pomočjo lastnosti sipane oz. odbite svetlobe preiskuje strukturo oddaljenega cilja. Njegov osnovni namen je zaznavanje in merjenje oddaljenosti predmetov, njegov način delovanja pa precej podoben radarju. Za razliko od radijskih valov uporablja svetlobne sunke, ki se zaradi mnogo manjše valovne dolžine (IR, UV ali vidna svetloba) znatno sipajo tudi na mikroskopskih delcih, molekulah in atomih.

Na sliki 1.1 je iz dveh perspektiv predstavljena LiDARska naprava, z vsemi ključnimi sestavnimi deli:

- laserski oddajnik,
- sprejemnik sipane in odbite svetlobe (sprejemni teleskop),
- detektor signala.

karakteristike naštetih poglavitnih delov LiDARske naprave so zbrane v tabeli 1.1.

Grob princip delovanja LiDARja je sledeč (slika 1.2): laser oddaja kratke svetlobne sunke z dolžino tipično okoli 10 ns in vršno močjo nekaj MW. Ti se na

poti skozi ozračje sipajo in absorbirajo. Sprejemnik odbite in sipane svetlobe je v našem primeru standardni sprejemni zrcalni teleskop, ki svetlobo zbere na detektor. Iz časovne razlike med trenutkom oddaje svetlobnega sunka in časom sprejema odboja lahko izračunamo razdaljo. Naša naprava za vsak oddan sunek vzorči signal odbojev s frekvenco 7,37 MHz v 256 14 bitnih vzorcev, kar omogoča detekcijo do razdalje 5,2 km z dolžinsko ločljivostjo 20,3 m [4].



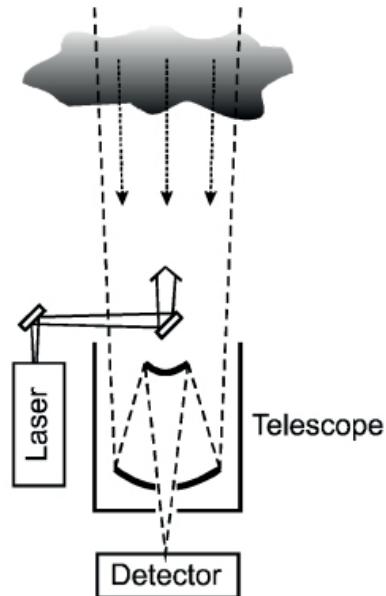
Slika 1.1: Poglavitni deli LiDARskega sistema so: laserski oddajnik (izvor svetlobe), sprejemni teleskop ter detektor signala.

Slabost navadnega LiDARja je ta, da je nevaren očem. Laserski sunek v vidnem delu spektra in s primerno intenziteto za praktične meritve je lahko ob direktnem ali indirektnem vstopu v oko usoden za opazovalčev vid. Zato se pri našem LiDARju uporablja svetloba valovne dolžine 1574 nm, ki izpolnjuje kriterij za varnost oči, saj je daleč izven vidnega dela spektra, torej v območju, ki je za oko neobčutljivo. Varnost še poveča periskop, ki razširi svetlobni žarek poslan iz LiDARja, in tako zmanjša tveganje za nevarnost laserskega sunka. Vendar zaradi spremembe valovne dolžine laserskega sunka, očem varni LiDAR ni tako učinkovit, kot bi bil lahko sicer.

LiDAR se poleg trivialne uporabe, uporablja na večih področjih, npr. tudi na področju določanje koncentracije plinov, za merjenje onesnaženosti zraka z različnimi aerosoli.

Pomanjkljivost našega očem varnega LiDARja je v nespremenljivi valovni dolžini izvora svetlobe. Običajni namenski LiDARji imajo spremenljivo valovno dolžino oddane svetlobe, kar jim omogoča zaznavanje delcev različnih velikosti, pri našem LiDARju pa smo omejeni na delce velikost $\sim 1,6 \mu\text{m}$. Ta

omejitev je sprejemljiva, saj je znano, da se svetloba najbolje siplje na delcih velikosti, podobne valovni dolžini svetlobe, s katero jih obsevamo [4].



Slika 1.2: Princip delovanja LiDARja. Laser oddaja kratke svetlobne sunke v ozračje. Teleskop prestreže del povratno sipane svetlobe. Časovno razločeno merjenje njene intenzitete daje krajevno razločeno informacijo o strukturi ozračja [6].

Laserski oddajnik	Nd: YAG + KTP OPO	
	Valovna dolžina	1574 nm
	Izhodna energija	9 mJ
	Repeticija	1 - 20 Hz
	Divergenca	6.8 mrad
	Širina žarka na izhodu	40 mm
Sprejemnik	Schmidt-Cassegrainov teleskop	
	Premer vhodne aperture	20 cm
	Vidni kot	1.8 mrad
Detektor	InGaAs PIN dioda	
	Premer	0.5 mm
	Interferenčni filter	
	Transmisija	73%
	Širina	15 nm
Zajemanje podatkov	Hitrost	10 MHz
	Resolucija	14 bit

Tabela 1.1: Karakteristike LiDARskega sistema [3].

Poglavje 2

Teoretične osnove za uresničitev cilja

2.1 Spletna storitev

Spletna storitev (angl. *Web service*) je način komunikacije med elektronskimi napravami preko svetovnega spleta (protokola HTTP). Gre za programske storitve, ki so ponujene na spletu in razvijalcem novih aplikacij omogočajo povpraševanje po določenih informacijah. Prednost tovrstnih spletnih storitev je v tem, da uporabljajo standarde neodvisne od platforme. Uporabnik spletne storitve mora poznati URL naslov storitve, podatkovne tipe, ki se uporabljajo pri klicih procedur, ni pa mu potrebno vedeti v katerem programskem jeziku je bila spletna storitev implementirana in na katerem računalniku se izvaja. Infrastruktura spletnih storitev temelji na treh tehnologijah [7]:

- UDDI (*Universal Description, Discovery, and Integration*),
- SOAP (*Simple Object Access Protocol*) in
- WSDL (*Web Services Description Language*).

UDDI je register, ki omogoča razvijalcem programske opreme poizvedovanje po spletnih storitvah, ki so na voljo. V registru UDDI lahko podjetja (uporabniki registra) objavljajo opise svojih spletnih storitev in vmesnikov za dostopanje do njih, druga podjetja pa lahko iz registra pridobivajo opise omenjenih storitev in nato avtomatično vzpostavijo povezave z njimi [7].

2.1.1 SOAP

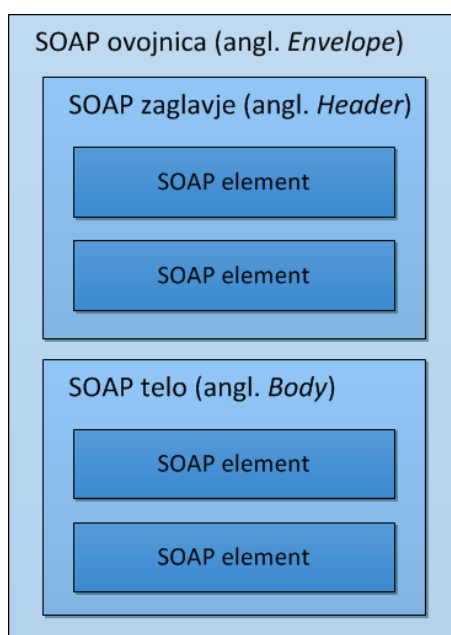
SOAP (*Simple Object Access Protocol*) je protokol, grajen na osnovi XML jezika in omogoča izmenjavo podatkov med aplikacijami na svetovnem spletu. Uporablja se za podajanje informacij (sporočil) tako v smeri strežnik-odjemalec kot tudi obratno. Deluje v povezavi s protokoli iz aplikacijske plasti (HTTP, SMTP ali RPC). Informacije, ki se prenašajo v sporočilu SOAP, so lahko v obliki dokumenta ali klica oddaljene procedure (RPC). Prav zaradi uporabe HTTP protokola se SOAP izogne težavam prehajanja preko požarnih zidov [7], [8].

Glavne prednosti SOAP protokola so [9]:

- **Ni tesno povezan z nobenim programskim jezikom.** Razvijalci programske opreme lahko uporabijo najnovejše programske jezike. Še posebno je dobrodošla neodvisnost od programskega jezika pri obstoječih aplikacijah.
- **Ni tesno povezan z določenim transportnim protokolom.** SOAP specifikacija določa način povezave SOAPa s protokoloma HTTP ali SMTP. SOAP sporočilo ni nič drugega kot XML dokument, ki ga lahko prenašamo z vsakim protokolom, ki podpira prenos znakov.
- **Ni tesno povezan z nobeno porazdeljeno objektno infrastrukturo.** Večino porazdeljenih objektnih sistemov lahko razširimo za podporo SOAPa.
- **Dopolnjuje obstoječe industrijske standarde.** SOAP uporablja XML za kodiranje sporočil. Tako SOAP, raje kot da bi uporabil sistem svoje vrste, dopolnjuje vrsto določil, ki so že definirana v specifikaciji XML Schema. SOAP lahko povežemo z obstoječim protokolom kot sta HTTP ali SMTP.
- **Omogoča interoperabilnost med različnimi okolji.** SOAP je zgrajen je obstoječih industrijskih standardih, kar omogoča aplikacijam na različnih platformah uspešno medsebojno komunikacijo, če le podpirajo te standarde. Na primer, namizna aplikacija, ki se uporablja na oseb- nem računalniku, lahko komunicira z aplikacijo zadnjega nivoja, ki teče na osrednjem računalniku, če le slednja zmore sprejemati in pošiljati XML prek HTTP protokola.

Standarden format SOAP protokola, vsebuje ovojnico in poljubno število morebitnih priponek, ki omogočajo, da se preko sporočila SOAP prenašajo tudi

podatki, ki sicer niso v obliki XML. SOAP ovojnica pa mora vselej uporabljati gramatiko XML - to med drugim omogoča tudi enostavno berljivost dokumenta.



Slika 2.1: Struktura sporočila SOAP.

Struktura SOAP sporočila vsebuje naslednje elemente [9] (primer slika 2.1):

- XML ovojnica (angl. »*Envelope*«), ki je korenski element sporočila, vsebuje glavo in telo sporočila. Namenjena je za prenos imen metod in parametrov do oddaljenih objektov.
- SOAP zaglavje (angl. »*Header*«) je neobvezno, vsebuje pa lahko razne napotke prejemniku, kot so informacije o transakcijah, pravila za koordiniranje aplikacijskih tipov objektov, ki se prenašajo kot argumenti, ter pravila (napotke) o varnosti.
- Telo (angl. »*Body*«) je obvezen del dokumenta in vsebuje podatke XML ali pa metode RPC in parametre, odvisno, ali gre za dokumentni ali pa za RPC tip sporočila SOAP. V telesu sporočila, je lahko neobvezno prisotna tudi napaka (»*SOAP Fault*«), s katero ponudnik storitve (SOAP prejemnik) sporoča, da je prišlo do napake.

SOAP omogoča odjemalcem klicanje oddaljenih procedur (RPC). Uporabniki izvedejo zahtevo s pošiljanjem sporočila SOAP na strežnik, ki ponuja spletno storitev. Strežnik se na zahtevo odzove z odgovorom, ki je prav tako SOAP-sporočilo [9]. V nadaljevanju sta predstavljeni dve SOAP sporočili, ki nastopita pri avtentikaciji oddaljenega uporabnika na LiDARski strežnik. Prvo sporočilo predstavlja poslano zahtevo, medtem ko drugo predstavlja odgovor na zahtevo.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
5   <soap:Body>
6     <Authenticate xmlns="http://tempuri.org/" />
7       <username>damir</username>
8       <password>geslo</password>
9     </Authenticate>
10  </soap:Body>
11 </soap:Envelope>

```

Slika 2.2: Primer zahteve v obliki sporočila SOAP.

Sporočilo na sliki 2.2 predstavlja SOAP zahtevo za sprožitev spletne storitve. V njem lahko opazimo, da ima ovojnica definiran URI imenskega prostora, glava sporočila je prazna, telo pa vsebuje ime operacije spletne storitve (*Authenticate*) ter vrednosti parametrov, za avtentikacijo spletne storitve (*username*, *password*).

Na sliki 2.3 je predstavljen odgovor, ki ga na prejšnjo zahtevo vrne spletna storitev. V njem so zapisane izhodne vrednosti klicane operacije (v našem primeru vrne varnostni ključ v formatu UUID¹), v formatu, ki ga predpisuje pripadajoča shema XML.

¹Enolična identifikacijska koda (angl. *Universal Unique Identifier*)

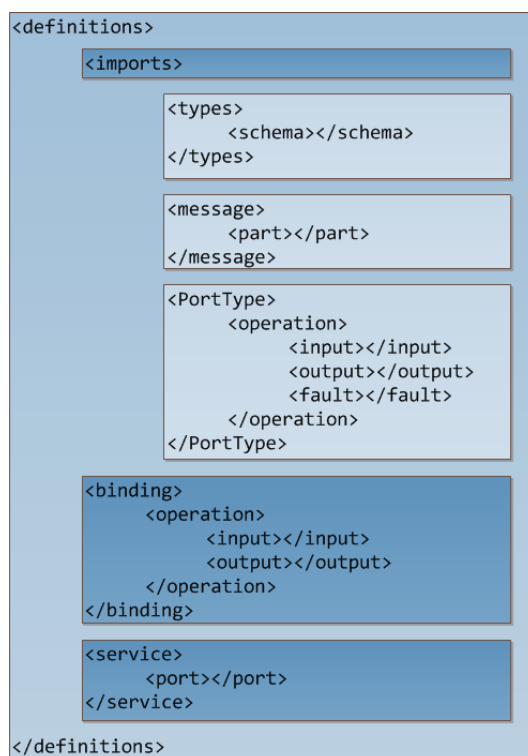
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:xsd="http://www.w3.org/2001/XMLSchema">
5   <soap:Body>
6     <AuthenticateResponse xmlns="http://tempuri.org/" />
7       <AuthenticateResult>
8         2d7437a6-b67c-4008-1375-f05549f763813
9       </AuthenticateResult>
10    </AuthenticateResponse>
11  </soap:Body>
12 </soap:Envelope>
```

Slika 2.3: Primer odgovora spletne storitve v obliki sporočila SOAP.

2.1.2 WSDL

WSDL (*Web Services Description Language*) je jezik za opis spletnih storitev in deluje kot vmesnik med odjemalcem in strežnikom. Protokol SOAP definira dostop do oddaljenih objektov in procedur, imena teh objektov in njihovi tipi pa so definirani v datotekah WSDL. Datoteke WSDL so praviloma shranjene na strežniku, ki ponuja določeno spletno storitev. Tudi te datoteke so shranjene v XML notaciji. Dokumenti WSDL se objavijo na spletu ali v registru UDDI, kjer so na voljo širšemu krogu uporabnikov. Za opis spletnih storitev WSDL jezik uporablja abstraktno predstavitev, ki se pri dostopu do spletne storitve preslika v točno določen mrežni protokol (SOAP, HTTP GET/POST) in v sporočila, ki se prenašajo po uporabljenem protokolu. Tako lahko razvijalec programske opreme s pomočjo namenskega programa (razvojnega orodja) iz WSDL datoteke zgradi vse potrebne objekte in funkcije, ki so potrebne za komunikacijo s spletno storitvijo [10].

WSDL spletno storitev opišemo z naborom WSDL elementov, dokument pa strukturiramo v jeziku XML. Glavni elementi, ki opisujejo WSDL storitev, so prikazani na sliki 2.4, podrobneje pa opisani v tabeli 2.1 [9].



Slika 2.4: Glavni elementi WSDL storitve.

Element	Opis
<i>types</i>	Podatkovni tipi, s katerimi so opisana sporočila.
<i>message</i>	Predstavlja definicijo sporočil, ki se preko operacij prenašajo med končnimi točkami. Sporočilo je lahko sestavljeno iz enega ali več logičnih delov, pri čemer je lahko vsak različnega tipa.
<i>portType</i>	Tipi končnih točk, ki predstavljajo nabor operacij, ki jih nudi spletna storitev. Operacija je lahko vhodni ali izhodni podatek, lahko pa predstavlja napako.
<i>binding</i>	Konkretni protokol in format podatkov, v katerem se prenašajo sporočila.
<i>service</i>	Storitev, katera predstavlja množico končnih točk z njihovimi naslovi.

Tabela 2.1: Opis glavnih elementov znotraj WDSL

2.2 XML

Tehnologija *eXtensible Markup Language* (XML) [11, 12] je preprost, prožen tekstovni format za opisovanje strukturiranih podatkov, kot so dokumenti, konfiguracije, transakcije, računi ipd.

Je označitveni jezik podoben HTMLju [13], vendar ne njegov nadomestek, temveč njegov komplement. XML lahko smatramo kot programsko in strojno neodvisen način prenašanja informacij. Na drugi strani je HTML jezik odgovoren le za prikaz informacij. Oznake (angl. »*tags*«) in sama struktura HTML dokumenta so vnaprej definirani oziroma standardizirani, in njihova uporaba je omejena le na njih (kot `< p>`, `<h1>`, ipd.).

Že iz samega imena lahko preberemo osnovni namen jezika. Jezik XML je razširljiv (angl. *extensible*), XML oznake niso vnaprej definirane, kar pomeni da je za definicijo strukture in oznak podatkov odgovoren uporabnik sam.

Pri izmenjavi datotek te vrste, je potrebno zagotoviti tudi možnost preverjanja ustreznosti zgradbe XML dokumenta (ne gre za preverjanje ustreznosti podatkov, gre le za preverjanje ustreznosti strukture). Za definicijo strukture (elementov, atributov in podatkovnih tipov znotraj njih), ter njihovo validacijo v XML dokumentih, se uporabljata standarda DTD ali XML Schema.

XML se pogosto uporablja tudi za hranjenje podatkov v datotekah ali celo podatkovnih bazah in za reševanje problemov kompatibilnosti. Eden od tipičnih problemov je nezmožnost izmenjave podatkov med takšnimi sistemi preko spleta. Hranjenje podatkov v XML datoteki nam omogoči, da lahko shranjene podatke uporabljajo različne vrste aplikacij, poleg tega pa XML format precej zmanjša kompleksnost teh podatkov.

2.2.1 Sintaksa dokumenta XML

Sintaksa XML je zelo enostavna in hkrati stroga. Pravila, ki določajo omenjeno sintakso, se je enostavno naučiti in uporabljati. Prav zaradi tega je tudi razvoj programske opreme, ki uporablja XML, preprost. Ob smiselnem poimenovanju elementov (oznak) je XML dokument lahko zelo enostaven in samoopisljiv, kar kaže tudi primer na sliki 2.5.

Primer XML dokumenta prikazuje slika 2.5. Na začetku XML datoteke se nahaja deklaracija XML dokumenta (angl. *XML declaration*), kjer navedemo različico XML standarda in način kodiranja znakov (angl. *character encoding*). Naslednja vrstica v našem primeru opisuje korenski element dokumenta. Korenska oznaka je prva oznaka v XML dokumentu in je, v pravilno formiranem XML dokumentu, neponovljiva. Vse ostale elemente moramo gnezditi pod to

oznako, torej znotraj korenkega elementa. V našem primeru so to: *to*, *from* in *body*. Zadnja vrstica zaključuje korenko oznako.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <note>
3   <to>Jure</to>
4   <from>Janez</from>
5   <body>Prosim, kupi mleko na poti domov.</body>
6 </note>

```

Slika 2.5: Primer XML dokumenta.

Vsak podatek v XML dokumentu je shranjen med začetno in zaključno oznako. Vsaka oznaka oziroma element ima lahko enega ali več atributov, ki jo podrobneje določajo. Oznake lahko gnezdimo, atributov pa ne smemo. Pri gnezdenju oznak je potrebno biti posebno pazljiv na pravilni vrstni red postavljanja zaključnih oznak. Vrednosti atributov morajo biti zapisani med dvema navednicama. Slika 2.6 prikazuje primer XML dokumenta z uporabo atributov.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <note to="Jure" from="Janez" body="Prosim, kupi mleko na poti domov." />

```

Slika 2.6: Primer XML dokumenta z atributi.

XML se razlikuje od HTML tudi v tem, da so oznake v XML občutljive na velikost črk. Tako je oznaka `<note>` različna od oznake `<Note>`. Zato moramo začetne in zaključne oznake dosledno pisati z enako velikostjo črk.

2.2.2 Vmesnik za procesiranje in razčlenjevanje XML

Za procesiranje in razčlenjevanje XML dokumentov obstaja vrsta tehnologij, ki se razlikujejo glede na način dostopa in predstavitve informacij. Med njimi sta dva glavna vmesnika:

- DOM (Document Object Model) je drevesni programski vmesnik v katerem je celotni dokument predstavljen kot drevesna struktura. Vozlišča znotraj drevesa omogočajo branje in urejanje XML dokumenta. Več o tem v nadaljevanju.

- SAX (Simple API for XML) je dogodkovni programski vmesnik. Za razliko od vmesnika DOM, SAX ne prebere najprej celotnega XML dokumenta v pomnilnik, ampak dokument procesira (bere) sekvenčno po posameznih delih (vozliščih). Med branjem dokumenta razčlenjevalnik proži dogodek, ki programu pove, da je prebral določen oz. željen simbol (začetek ali konec elementa).

2.2.2.1 Objektni model dokumenta – DOM

DOM (angl. *Document Object Model*) [14] je programski vmesnik, ki si pri procesiranju pomaga z drevesno strukturo XML dokumenta. Vmesnik DOM omogoča branje, iskanje in urejanje v XML dokumentih. Delovanje vmesnika DOM lahko opišemo v treh fazah: 1. Prebere dokument. 2. Razčleni dokument na posamezne elemente (besedilo, komentar, element, atribut, navodilo za procesiranje). 3. Dobljene elemente (iz točke 2) strukturira v drevo vozlišč, ki se nato shrani v pomnilnik.

Uporaba vmesnika DOM prinaša kar nekaj prednosti, ena izmed njih je ta, da je neodvisen od platforme in programskega jezika. Vmesnik DOM elemente XML dokumenta prebere v drevesno strukturo, uporabnik pa lahko te elemente kasneje naslavlja in z njimi manipulira, s pomočjo knjižnice poljubnega programskega jezika. Z urejanjem vsebine XML dokumenta s pomočjo vmesnika DOM, se spreminja drevesna struktura dokumenta, kar se izkaže za zelo uporabniško prijazen način interakcije s strukturo XML dokumenta. [15].

DOM razčlenjevalnik je manj primeren za delo z večjimi XML datotekami, saj celotno vsebino dokumenta XML najprej shrani v pomnilnik, in šele nato z njo manipulira. [15].

2.3 Datotečni format LAS

Format LAS [16], je javni datotečni format za izmenjavo podatkov pridobljenih iz LiDARske naprave. Namenjen je izmenjavi 3D točk, predstavljenimi s koordinatami x, y in z ter pripadajočimi skalarnimi vrednostmi. Zaradi potrebe po standardizaciji zapisa, iz LiDARske naprave pridobljenih podatkov, je društvo ASPRS [16], ki se ukvarja z uveljavitvijo tehnologij fotogrametrije in daljinskega zaznavanja na področju GIS, razvilo datotečni format LAS [17]. LAS danes predstavlja temeljni industrijski standard za izmenjavo LiDARskih podatkov. Gre za binarni format in kot tak, predstavlja alternativo tekstovnim (ASCII) in ostalim datotečnim formatom, ki so jih osnovala ter jih uporabljajo mnoga podjetja. Pri izmenjavi tekstovnih (ASCII) datotek obstajata

dve glavni težavi. Prva je velikost datotek, ki je lahko, tudi za majhne količine podatkov, zelo velika, njihovo branje in interpretacija pa zato počasna. Druga težava je nezanesljivost ASCII datotek pri prenosu, kjer se lahko informacije pri prenosu izgubijo. Ta težava je v formatu LAS rešena tako, da so polja, ki predstavljajo ključne informacije za LiDARske podatke obvezna, sicer datoteka ni veljavna. Tako naslovimo še en problem lastniških sistemov (datotek), in sicer prenosljivost med različnimi sistemi. Povzeto povedano: datoteka v formatu LAS je binarna datoteka, ki ohranja informacije specifične za LiDARsko napravo, brez pretirane kompleksnosti. Prav zato je shranjevanje datotek v formatu LAS hitro pridobilo podporo večine proizvajalcev strojne in programske opreme [18].

Danes poznamo več različic datotečnega formata LAS. Najpogosteje se uporabljajo formati verzije 1.0 [19], 1.1 [20], 1.2 [21] in 1.3 [18]. Na voljo je tudi že predlog formata 2.0 [22], ki je glede strukture podatkov in načina zapisa podatkov v datoteko svobodnejši.

Struktura formata LAS je sestavljena iz treh podatkovnih blokov (prikazani v tabeli 2.2 in na sliki 2.7), in sicer iz javne glave, zapisov spremenljive dolžine (VLR) in zapisov točk s koordinatami in pripadajočimi skalarnimi vrednostmi.

Blok javne glave
Zapisi spremenljive dolžine
Zapisi točk

Tabela 2.2: Struktura LAS datoteke

Datoteke LAS verzije 1.3 vsebujejo tudi podatke o celotni valovni obliki. Njihova struktura je sestavljena iz blokov prikazanih v tabeli 2.3.

Blok javne glave
Zapisi spremenljive dolžine
Zapisi točk
Razširjeni zapisi spremenljive dolžine (Paketi podatkov o celotni valovni obliki)

Tabela 2.3: Struktura LAS datoteke z podatki valovne oblike

Za številčenje bitov vseh vrednosti v datoteki formata LAS, uporabljamo način imenovan ozki-konec (angl. *little-endian*). Glava datoteke je sestavljena iz bloka javne glave, kateri sledi eden ali več zapisov spremenljive dolžine.

Javna glava vsebuje splošne podatke, kot so število točk v datoteki, število zapisov spremenljive dolžine, podatki za usklajevanje mej ipd. Zapisi spremenljive dolžine vsebujejo spremenljive vrste podatkov, vključno z informacijo o projekciji, metapodatke, podatke o paketih podatkov valovne oblike (angl. *waveform packet information*) in uporabniške podatke. Paketi podatkov valovne oblike hranijo podatke valovne oblike, ki pridejo iz same LiDARske naprave. V tabeli A.7 je podrobneje prikazana struktura javne glave datoteke [18]. V glavi LAS datoteke, najdemo tudi podatke s pomočjo katerih izračunamo absolutne koordinate točk. Posamezna koordinata x (tudi y in z) se s podatki, dobljenimi iz glave datoteke LAS, izračuna z enačbo 2.1.

$$x_{koordinata} = (x_{vrednost} * x_{skalirni faktor}) + x_{odmik} \quad (2.1)$$

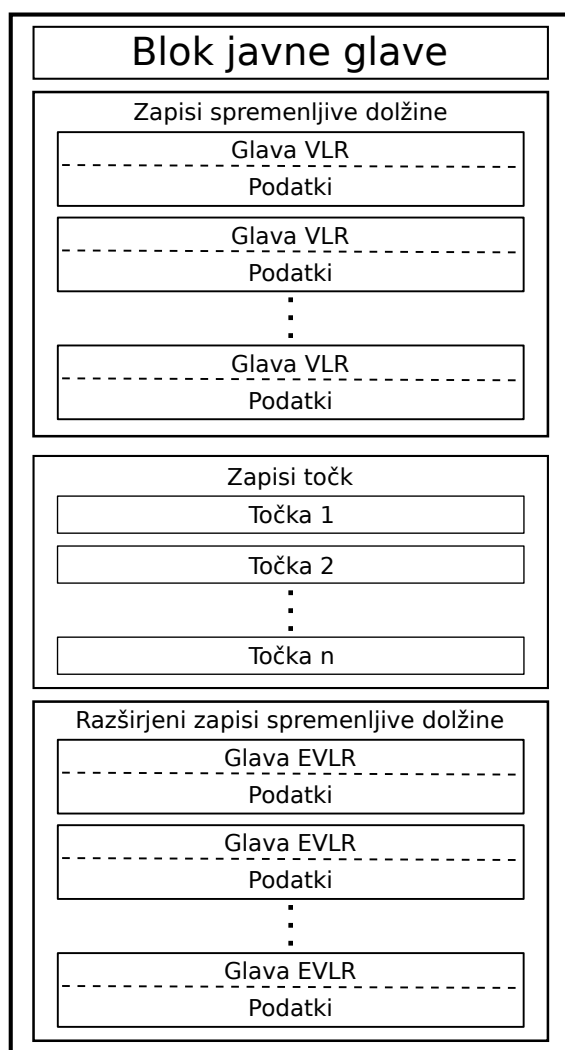
Takoj za blokom javne glave datoteke sledijo zapisi spremenljive dolžine (angl. *Variable Length Records - VLR*), katerih število je navedeno v glavi datoteke. Struktura tega bloka datoteke je prikazana v tabeli 2.4.

Element/polje	Opis	Tip	Velikost	Obvezen
Reserved	Rezervirano.	unsigned short	2 zloga	
User ID	ID uporabnika, ki je kreiral ta zapis spremenljive dolžine.	char[16]	16 zlogov	*
Record ID	Vrednost je odvisna od polja »User ID« in jo lahko določi uporabnik.	unsigned short	2 zloga	*
Record Length after Header	Število zlogov po glavi zapisa spremenljive dolžine.	unsigned short	2 zloga	*
Description	Opis podatkov.	char[32]	32 zlogov	

Tabela 2.4: Zapisi spremenljive dolžine.

V tretjem bloku, kjer so točke, se nahaja večina podatkov. V glavi datoteke se poleg števila točk nahaja tudi informacija o strukturi oz. formatu zapisa točk. Možnih je več načinov oz. formatov zapisa. Različica 1.3 definira pet različnih formatov zapisa točk [23]. Formata 0 in 1 (podrobneje predstavljena v prilogi v tabelah A.1 in A.2) sta skozi vse nove različice specifikacij LAS ostale nespremenjene. Zapisi točk formata 2 in 3 (podrobneje predstavljena v prilogi v tabelah A.3 in A.4), so bili dodani v verziji 1.2, zagotavljajo pa podporo dodatnim podatkom za slike. Dodana so bila namreč polja za rdeče, zelene in modre spektralne podatke v posamezni valovni dolžini. V specifikaciji LAS verzije 1.3, so bili dodani formati točk 4 in 5 (podrobneje predstavljena v prilogi v tabelah A.5 in A.6). Ta dva se od prejšnjih formatov razlikujeta v tem, da poleg zapisov spremenljivih dolžin omogočata shranjevanje LiDARskih

podatkov o njegovi celotni obliki odbitega valovanja (angl. *Full-Waveform LiDAR Data*).



Slika 2.7: Struktura formata LAS.

2.4 Določanje položaja točk na Zemlji

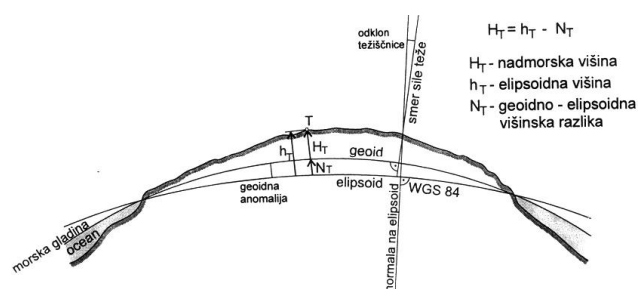
2.4.1 Oblika in dimenzija zemlje

Zgodnje predstave o obliki in velikosti Zemlje segajo že v Stari vek. Babilonci, Egipčani ter Arabci so si predstavljali Zemljo kot omejeno življenjsko območje - kot ravno okroglo ploščo, ki je obdana z vodo. Prvo pojmovanje Zemlje kot krogle, sega v obdobje helenistične Grčije (Aristotel, Eratosten). Od 17. do začetka 19. stoletja so bila izpeljana obsežna in vse bolj natančna merjenja, s katerimi so potrdili okroglo obliko Zemlje, in določili njeno velikost. Da ima Zemlja obliko rotacijskega elipsoida, pa sta že v 17. stoletju sklepala Anglež Newton in Nizozemec Hygens [24].

Fizična površina Zemlje je močno razčlenjena (velike spremembe v času) in zelo kompleksna, zato za njeno predstavitev uporabljamo poenostavljene modele in sicer glede na kontekst:

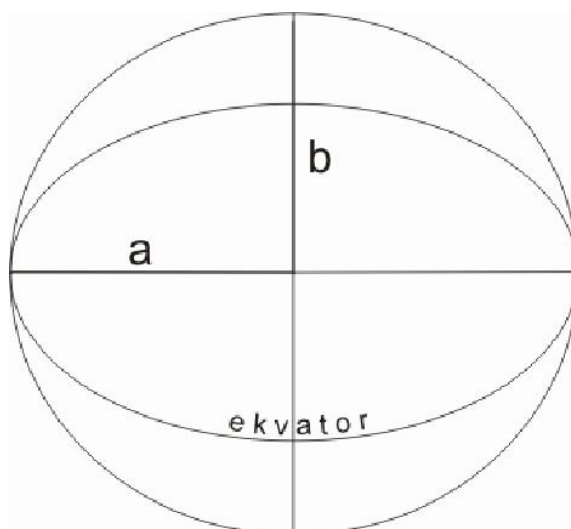
- Primerjalna ploskev je fizikalno definirana (fizikalni model - geoid), če se meritve nanašajo na zemeljsko težnost.
- Primerjalna ploskev je matematično definirana (elipsoid, krogla), če se izračuni izvajajo v geometričnem prostoru.

Zemljo smatramo kot geoid, to je referenčna ekvipotencialna (v vsaki točki pravokotna na vektor sile teže) ploskev. Ta v splošnem poteka pod kontinenti v globini in enaka nadmorski višini zemeljske površine [25].

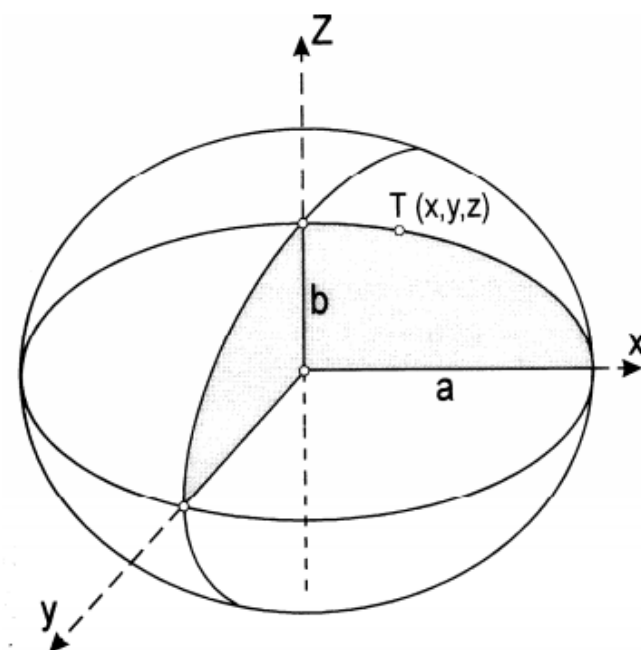


Slika 2.8: Povezava geoid in elipsoid [25].

Ker je privlačnost odvisna od nepravilne razporeditve zemeljskih mas tako na površini kot v notranjosti Zemlje, geoidu ni mogoče določiti matematične enačbe. Najboljši opis oblike zemeljske površine kot matematično določljivega geometrijskega telesa, je rotacijski elipsoid (elipsa zavrtena okoli krajše osi) [26].



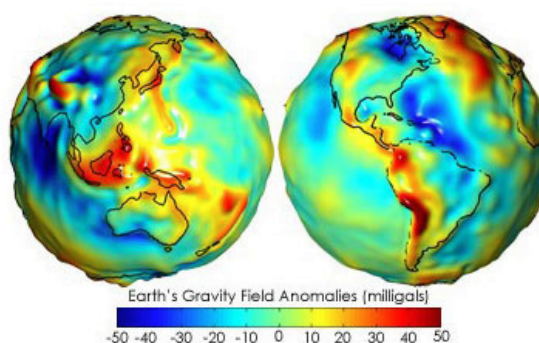
Slika 2.9: Rotacijski elipsoid z parametri a (velika polos), b (mala polos) in f (sploščenost elipsoida, ki je definirana z formulo $f = \frac{(a-b)}{a}$) [27].



Slika 2.10: Geodetski referenčni elipsoid [25].

Zaradi neenakomerne oblike Zemlje (oblike rotacijskega elipsoida), ni mogoče nedvoumno določiti rotacijskega elipsoida, ki bi se najbolje prilagodil obliki in velikosti Zemlje kot celote. Zato elipsoide ločimo v dve skupini:

- Globalni elipsoidi (GRS 1967, WGS 84, GRS80 ...)
- Lokalni elipsoidi (Bassel, Everest ...)



Slika 2.11: Vizualizacija valovanja geoida, z uporabo enot težnosti [28].

Da bi se rotacijski elipsoid najbolj prilegal določenemu območju, si je vsaka država za svoje potrebe izbrala svoj referenčni elipsoid. Geodetski referenčni elipsoidi, kateri določajo obliko in velikost Zemlje in se uporabljajo pri nas so Besselov, WGS84 in GRS80 (podrobneje so predstavljeni v tabeli 2.5).

Leto	ime	a	b	e
1841	Bessel	6.377.397,15500	6.356.078,96325	1/299,15
1984	WGS84	6.378.137,00000	6.356.752,30000	1/298,26
1980	GRS80	6.378.137,00000	6.356.752,31414	1/298,26

Tabela 2.5: Dimenzije elipsoidov Bessel, WGS84 in GRS80 [29, 27].

Obliko zemlje lahko predstavimo tudi kot kroglo. Krogla je najenostavnejši analitični opis oblike Zemlje. Tovrstna oblika Zemlje se uporablja za navigacijske, astronomske, geografske in geodezijske (kartografija) namene. Polmer krogle je izbran tako, da se najbolje prilega referenčnemu elipsoidu.

2.4.2 Koordinatni sistemi

Za potrebe določitve položaja točk (objektov) v prostoru mora definirati koordinatni sistem. Koordinatni sistem je sredstvo za podajanje geometrijskega

položaja točke (s pomočjo koordinat) v prostoru. Najbolj preprost in razširjen koordinatni sistem je t.i. »tridimenzionalni pravokotni kartezični« koordinatni sistem [27].

V geodeziji ločimo med koordinatnimi sistemi na elipsoidu ali krogli (prostorski koordinatni sistemi) in koordinatnimi sistemi v ravnini (ravninski koordinatni sistemi). Ravninski pravokotni sistemi so metode projiciranja Zemlje ali dela Zemlje na projekcijsko ploskev in so po metodi tudi poimenovani:

- Koordinatni sistem, ki izvira iz Univerzalne transverzalne Mercatorjeve projekcije (*Universal Transverse Mercator Projection* – UTM),
- mednarodni koordinatni sistem, ki določa dimenzijo in oznako mednarodne karte v merilu 1 : 1 000 000,
- Gauß-Krügerjev koordinatni sistem,
- avstro – ogrski lokalni koordinatni sistemi, ki sloni na Söldnerjevi projekciji.

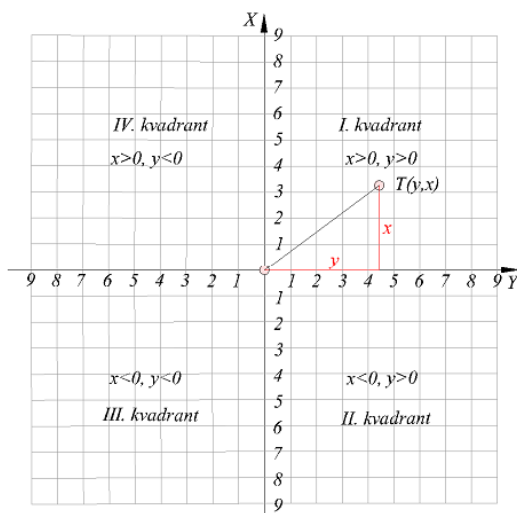
Med prostorskimi koordinatnimi sistemi največ uporabljamo [25]:

- geografski koordinatni sistem (geografska širina, geografska dolžina),
- WGS 84 (*World Geodetic System 1984*) – svetovni geodetski koordinatni sistem Navstar (geografska širina, geografska dolžina, nadmorska višina ali y , x , z).

2.4.2.1 Pravokotni (kartezični) koordinatni sistem v ravnini

Pravokotni kartezični koordinati točke, ki leži v koordinatnem sistemu, predstavljata razdalji te točke od koordinatnih osi X in Y . Pravokotni koordinatni sistem, je koordinatni sistem, ki je sestavljen iz dveh ali treh med seboj pravokotnih osi. Sestavne osi pravokotnega koordinatnega sistema imenujemo abscisna os (os X), ordinatna os (os Y) in aplikativna os (os Z). Presečišče vseh osi koordinatnega sistema imenujemo koordinatno izhodišče [30].

Na sliki 2.12 je prikazan geodetski pravokotni koordinatni sistem, ki se razlikuje od matematičnega. V geodetskem koordinatnem sistemu pozitivni del osi kaže proti severu (negativni proti jugu), pozitivni del ordinatne osi pa je usmerjen proti vzhodu (negativni proti zahodu). Geodetski koordinatni sistem se od matematičnega razlikuje tudi po tem, da je desnosični, kar pomeni, da koti naraščajo v desno (enako velja za kvadrante) [27].

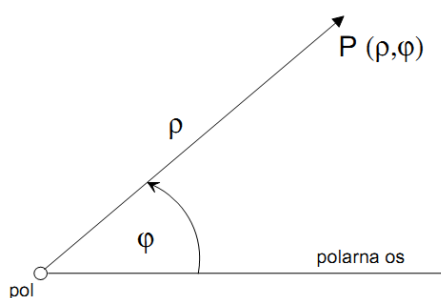


Slika 2.12: Pravokotni koordinatni sistem (razdelitev na kvadrante) [30].

2.4.2.2 Polarni koordinatni sistem v ravnini

Uporablja se kot alternativa pravokotnem koordinatnem sistemu [30]. Na sliki 2.13 je prikazan primer polarnega koordinatnega sistema. Polarni koordinati točke (na sliki 2.13 je to točka P), ki leži v koordinatnem sistemu predstavljata polarni koordinati:

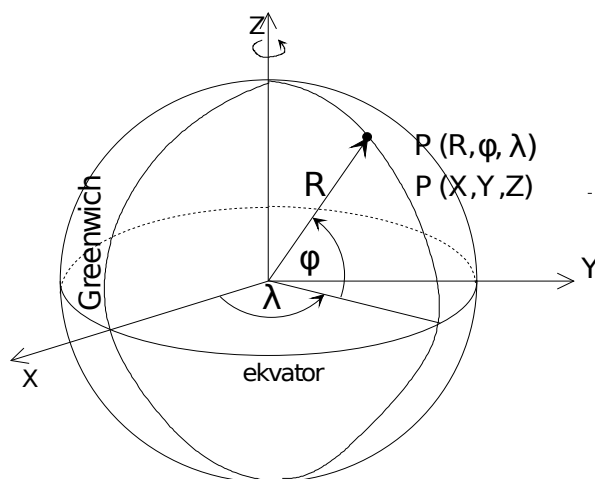
- radij vektor (ρ), to je vektor razdalje točke od koordinatnega izhodišča,
- polarni kot (φ), je kot med radij vektorjem in polarno osjo (izhodiščem za štetje kotov).



Slika 2.13: Primer polarnega koordinatnega sistema.

2.4.2.3 Geografske koordinate na Zemlji kot krogli

Geografski koordinatni sistem, s pomočjo koordinat geografske širine in geografske dolžine, določa lego točke na zemeljski površini. Geografska širina točke je njena sferna razdalja od ekvatorja, oziroma kot med normalo na površino elipsoida in med ravnino ekvatorja (slika 2.14). Izhodišče za računanje je ekvator (vrednost kota). Geografska dolžina točke je njena sferna razdalja od izhodiščnega meridiana (Greenwich) oziroma kot med ravnino začetnega meridiana in ravnino meridiana točke. Računa se od začetnega meridiana na vzhod in zahod (vrednost kota).



Slika 2.14: Krogla z radijem R (konstanta), kjer je točka podana s koordinatama φ (zemljepisna širina) in λ (zemljepisna dolžina) ali s koordinatama X , Y , Z . [27]

2.4.3 Državni koordinatni sistem

Državni koordinatni sistem je koordinatni sistem, ki je uradno veljaven na območju neke države [30, 31]. Za vzpostavljanje, vzdrževanje in vodenje državnega koordinatnega sistema je pristojna Geodetska uprava Republike Slovenije. V Sloveniji je od 1. 1. 2008 poleg starega v uporabi tudi novi horizontalni koordinatni sistem. Oba sta definirana na podlagi naslednjih kartografskih projekcij:

- GK - Gauß-Krügerjeva projekcija (stara),
- TM – transverzalna (prečna) Mercatorjeva projekcija (nova).

Trenutno so v Republiki Sloveniji predpisani naslednji ravninski koordinatni sistemi [32, 30]:

- višinski državni koordinatni sistem,
- horizontalni državni koordinatni sistem, ki ga označujemo z D48/GK (definiran na podlagi Gauß-Krügerjeve projekcije) in
- novi horizontalni državni koordinatni sistem, ki ga označujemo z D96/TM (definiran na podlagi transverzalne Mercatorjeve projekcije).

2.4.3.1 Višinski državni koordinatni sistem

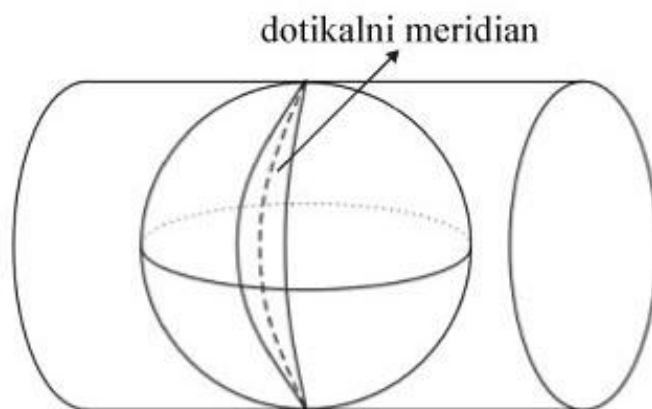
Referenčna ploskev višinskega koordinatnega sistema je najpogosteje geoid (kvazigeoid). Geodetska višinska mreža Republike Slovenije ima fizično izhodišče v normalnem reperju (izhodiščna višinska točka), ki se nahaja v naselju Smolnik v občini Ruše in je določena na podlagi večletne registracije srednje morske gladine s pomočjo mareografa [27].

2.4.3.2 Gauss-Kruegerjeva projekcija

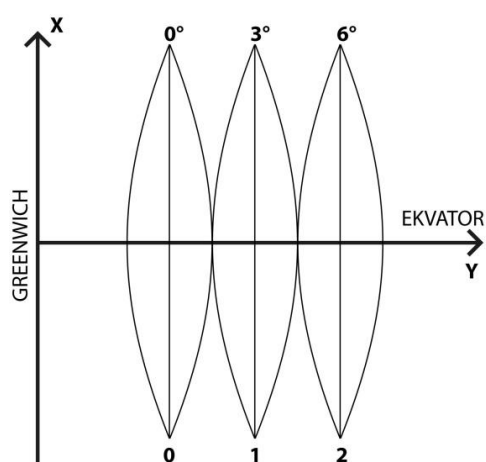
Gauss-Kruegerjeva projekcija je projekcija na Besselovem referenčnem elipsoidu in predstavlja matematično osnovo slovenskega kartografskega sistema D48/GK. Spada med projekcije TM (Transverzalna Mercatorjeva), in sicer:

- konformna projekcija - ohranja kote,
- prečna projekcija - valj se dotika elipsoida v dotikalnem meridianu,
- cilindrična projekcija - pomožna projekcijska ploskev je plašč valja (prečna in cilindrična projekcija je prikazana na sliki 2.15),
- centralna projekcija - projekcijski center je v središču elipsoida - Zemlje.

Gauss-Kruegerjeva projekcija razdeli zemeljski elipsoid na 120 meridianskih con oz. pasov (prikazani na sliki 2.16), ki potekajo od severnega do južnega pola in imajo širino 3° geografske dolžine. Izhodišče štetja meridianskih con je meridian Greenwich. Slovenija ima ugoden geografski položaj, saj poldnevnik 15° vzhodne geografske širine poteka skoraj povsem po sredini Slovenije. Zato lahko celotno območje Slovenije prikažemo v eni sami, 5. meridianski coni s srednjim poldnevnikom 15° [25].



Slika 2.15: Prikaz prečne cilindrične projekcije [27].



Slika 2.16: Prikaz meridianskih con Gauss-Kruegerjeve projekcije.

2.4.3.3 Projekcija TM

Projekcija TM (Transverzalna Mercatorjeva) je konformna (ohranja kote) cilindrična projekcija, pri kateri je cylinder rotiran za 90° (prečna oz. transverzalna) glede na ekvator. Referenčna ploskev v prečni Mercatorjevi projekciji je elipsoid GRS80. Vsi vzporedniki in poldnevnik, razen ekvatorja in dotikalnega meridiana, se v prečni Mercatorjevi projekciji preslikajo kot krivulje. Ekvator in dotikalni meridian se preslikata kot ravni, med seboj pravokotni liniji. Njuno presečišče predstavlja izhodišče pravokotnega koordinatnega sistema cone. Projekcija TM se uporablja za navigacijske karte.

Novejši slovenski horizontalni koordinatni sistem temelji na skupnem evropskem koord. sistemu ETRS89. Skupaj s slovenskim višinskim koordinatnim sistemom EVRS temeljita na evropskem prostorskem koord. sistemu ESRS [30].

2.4.4 Metode določanja položaja točk

Točkam na Zemljinem površju lahko določimo absolutni ali relativni položaj [33]. Med seboj se razlikujejo po natančnosti rezultatov meritev in vrsti uporabljene opreme. Pri absolutnem načinu določamo položaj posamezne točke v realnem času v svetovnem geodetskem sistemu WGS 84. Pri relativnem načinu določitve položaja točke določamo medsebojni položaj dveh ali več točk [24]. Položaj točk določimo na osnovi kotnih in faznih opazovanj.

2.4.4.1 Absolutna določitev položaja

Absolutna določitev položaja je najmanj natančna metoda. S to metodo je dosežena 10-metrška položajna in 20-metrška višinska natančnost. Rezultati te metode so primerni za navigacijo vozil in popotnikov. Uporabljajo se v navtiki, v turizmu, za mnoga GIS orodja in vse ostale uporabnike, ki jim ustreza ta stopnja natančnosti [33].

2.4.4.2 Relativna določitev položaja

Relativno določanje položaja pomeni določitev položaja ene točke s pomočjo druge točke, katere položaj že poznamo. Uporabimo lahko neposredna opazovanja med točkami ali posredna opazovanja s točk do ekstraterestičnih objektov [34, 33]. Pri določanju relativnega položaja se srečamo z dvema primeroma (angl. *Geodetic problems*), in sicer [24]:

- S pomočjo neposredne metode (angl. *direct solution*) se določi položaj druge točke glede na prvo, na osnovi znanih smeri (azimuta) in razdalje. Torej glede na podano zemljepisno širino in dolžino točke (φ_1, λ_1) , ki leži na površini elipsoida z azimutom α_1 in razdaljo s , se izračuna končna točka (φ_2, λ_2) z (povratnim) azimutom α_2 .
- Z inverzno metodo (angl. *inverse solution*) se z znanimi podatki obeh točk določi smer in razdalja med točkama. Med podanima točkama (φ_1, λ_1) in (φ_2, λ_2) , ki ležita na površini elipsoida, se izračuna geodetska razdalja s , ter azimuta α_1 in α_2 .

Za rešitev zgornjih dveh primerov se uporabljajo številne metode za računanje najkrajših razdalj (angl. *great-circle distance*) med dvema lokacijama (točkama) oziroma za določitev položaja (zemljepisna dolžina in širina) točk. Naš program uporablja neposredno Vincentyjevo metodo, ki glede na podano začetno točko, smer (azimut) in razdaljo izračuna zemljepisno širino in dolžino (angl. *latitude, longitude*) druge točke. V našem primeru je to z LiDARsko napravo izmerjena točka.

2.4.4.3 Vincentyjeve formule

Vincentyjeve formule [35, 36] zajemajo dve povezani iterativni metodi, ki se uporabljata za izračun razdalje med dvema točkama na površini sferoida. Razvil ju je Thaddeus Vincenty (1975) in temeljita na predpostavki, da ima Zemlja obliko rotacijskega elipsoida.

Vincentyjeve formule se množično uporabljajo v geodeziji, saj omogočajo točnost izračuna do 0,5 mm. Podobne metode za izračun razdalje med dvema točkama, ki temeljijo na sferičnem modelu Zemlje, so veliko preprostejše. Zato omogočajo manjšo natančnost, in sicer približno do 0,3%, kar pa je lahko še vedno dovolj dobro za normalno uporabo. Vincentyjeve formule so lahko programirljive, zato je njihova uporaba v računalništvu zelo razširjena. Te formule temeljijo na iterativnem izračunu, pri čemer je število iteracij potrebnih za rešitev majhno.

Kot smo že omenili Vincentyjeve formule zajemajo dve metodi. Prva je neposredna metoda (angl. *direct solution*), ki na podlagi znanih smeri (azimuta) in razdalje izračuna lokacijo (položaj) druge točke glede na prvo. Druga je inverzna metoda (angl. *inverse solution*), ki na podlagi podanih koordinat dveh lokacij (točk) izračuna geografsko razdaljo in smer (azimut) med dvema znanimi točkama. Za računanje pozicij izmerjenih točk z LiDARsko napravo, uporabimo Vincentyjevo neposredno metodo (angl. *direct solution*), ki uporablja zapise, kot so prikazani v tabeli 2.6.

V nadaljevanju je predstavljena Vincentyjeva neposredna metoda (enačbe od 2.2 do 2.13). O validaciji oz. potrditvi Vincetnyjeve formule si lahko več preberete v viru [37].

$$\sigma_1 = \arctan \left(\frac{\tan U_1}{\cos \alpha_1} \right) \quad (2.2)$$

$$\sin \alpha = \cos U_1 \sin \alpha_1; \quad \cos^2 \alpha = (1 - \sin \alpha)(1 + \sin \alpha) \quad (2.3)$$

$$A = 1 + \frac{u^2}{16384} \{4096 + u^2 [-768 + u^2(320 - 175u^2)]\} \quad (2.4)$$

$$B = \frac{u^2}{1024} \{256 + u^2 [-128 + u^2(74 - 47u^2)]\} \quad (2.5)$$

$$2\sigma_m = 2\sigma_1 + \sigma \quad (2.6)$$

$$\Delta\sigma = B \sin \sigma \left\{ \cos(2\sigma_m) + \frac{1}{4}B [\cos \sigma (-1 + 2 \cos^2(2\sigma_m)) - \frac{B}{6} \cos(2\sigma_m)(-3 + 4 \sin^2 \sigma)(-3 + 4 \cos^2(2\sigma_m))] \right\} \quad (2.7)$$

$$\sigma = \frac{s}{bA} + \Delta\sigma \quad (2.8)$$

Prvi izraz enačbe (enačba 2.8) se uporabi kot prvi približek k vrednosti σ . Nato se ostale enačbe (2.2 - 2.8) ponavljajo dokler spremembe vrednosti σ niso zanemarljivo majhne. Ko vrednost σ doseže zadostno natančnost, sledi naslednja ocena oz. izračun:

$$\phi_2 = \arctan \left(\frac{\sin U_1 \cos \sigma + \cos U_1 \sin \sigma \cos \alpha_1}{(1-f)\sqrt{\sin^2 \alpha + (\sin U_1 \sin \sigma - \cos U_1 \cos \sigma \cos \alpha_1)^2}} \right) \quad (2.9)$$

$$\lambda = \arctan \left(\frac{\sin \sigma \sin \alpha_1}{\cos U_1 \cos \sigma - \sin U_1 \sin \sigma \cos \alpha_1} \right) \quad (2.10)$$

$$C = \frac{f}{16} \cos^2 \alpha [4 + f(4 - 3 \cos^2 \alpha)] \quad (2.11)$$

$$L = \lambda - (1-C)f \sin \alpha \left\{ \sigma + C \sin \sigma [\cos(2\sigma_m + C \cos \sigma(-1 + 2 \cos^2(2\sigma_m)))] \right\} \quad (2.12)$$

$$\alpha_2 = \arctan \left(\frac{\sin \alpha}{-\sin U_1 \sin \sigma + \cos U_1 \cos \sigma \cos \alpha_1} \right) \quad (2.13)$$

zapis (angl. notation)	pomen
a	dolžina glavne osi elipsoida (polmer na ekvatorju - 6378137.0 metrov v WGS-84)
f	položni del referenčnega elipsoida (1/298.257223563 v WGS-84)
$b = (1 - f) \cdot a$	dolžina manjše osi elipsoida (polmer pri polih)
φ_1, φ_2	zemljepisna širina točk
$U1 = \arctan[(1 - f) \varphi_1]$	zmanjšana zemljepisna širina
$U2 = \arctan[(1 - f) \varphi_2]$	zmanjšana zemljepisna širina
$L = L2 - L1$	razlika v zemljepisni dolžini dveh točk
λ_1, λ_2	zemljepisne dolžine točk na pomožni sferi
α_1, α_2	začetni/končni azimut (smer)
α	azimut na ekvatorju
s	elipsoidna razdalja med dvema točkama
σ	dolžina loka med točkama na pomožni sferi

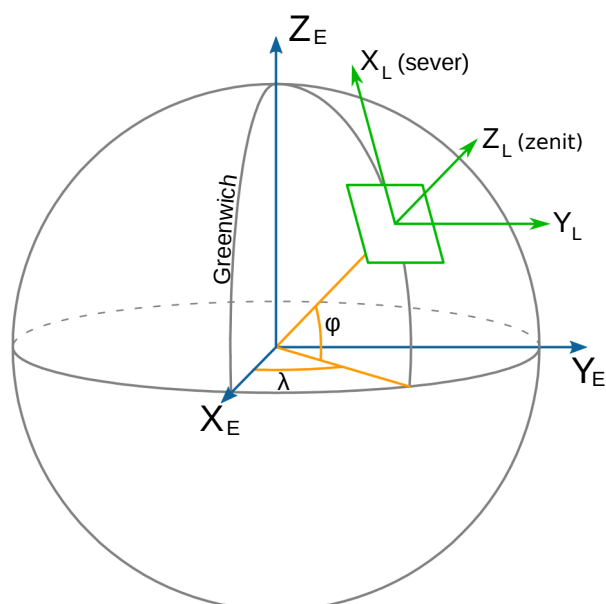
Tabela 2.6: Zapisi Vincenty-jeve neposredne metode.

2.5 LiDARski lokalni koordinatni sistem

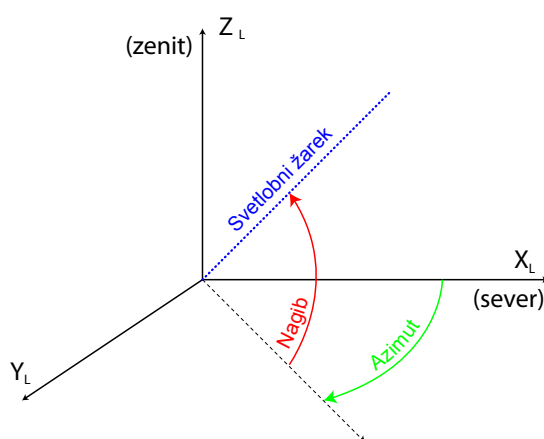
LiDARska naprava pri izvajanju meritev uporablja svoj lokalni (interni) koordinatni sistem. Tako lahko oddaljeni uporabnik (odjemalec), s pomočjo aplikacije za oddaljeno delo z LiDARsko napravo (*RhoLaaS*), prejete podatke (izmerjene točke) preslika v realni geometrijski sistem. Relacija med lokalnim in globalnim koordinatnim sistemom je prikazana na sliki 2.17.

Oddaljeni uporabnik (odjemalec), se pri določanju smeri meritve zanaša na predpostavko, da se LiDARska naprava zna postaviti v svoj koordinatni sistem (slika 2.18). Odmični kot meritve (angl. *Yaw*) se določi na podlagi azimuta. Azimut je kot med severom in izbrano smerjo. Merimo ga od severa v smeri urinega kazalca. S pomočjo osi Z, ki je usmerjena proti zenitu² se določi tudi nagibni kot, na podlagi katerega kasneje program za upravljanje z LiDARsko napravo izračuna oddaljenost izmerjene točke, kar je eden izmed parametrov za izračun njenega geografskega položaja (Vincentyjeva neposredna metoda).

²Točka na nebu, ki je navidezno neposredno nad točko opazovanja oziroma točka na nebesni kroglji z nebesno višino enako +90 stopinj.



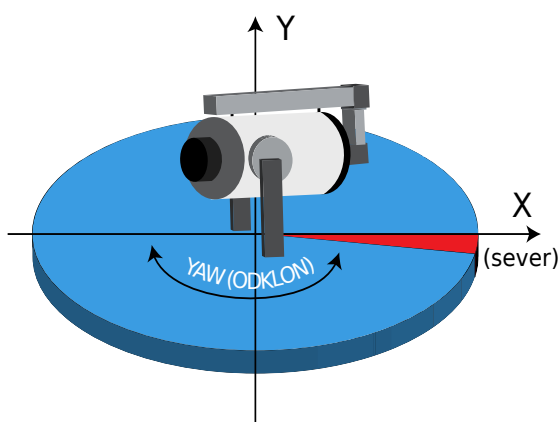
Slika 2.17: Povezava med globalnim koordinatnim sistemom in LiDARskim lokalnim koordinatni sistemom.



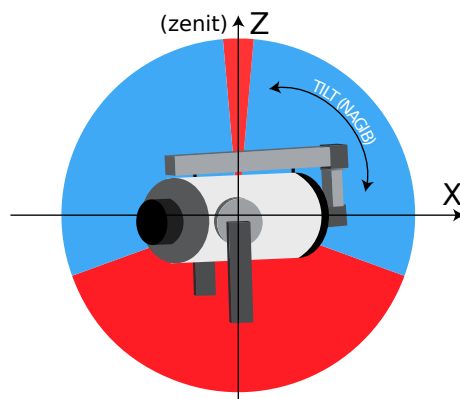
Slika 2.18: LiDARski interni koordinatni sistem. Izhodišče koordinatnega sistema predstavlja referenčno točko (položaj) LiDARske naprave, konkretno sta to podatka geografski dolžini in širini (φ , λ). Modra (pikčasta) črta na sliki predstavlja svetlobni žarek oddan iz LiDARske naprave, rdeča krivulja predstavlja nagibni kot (angl. *Tilt*), zelena krivulja pa azimut oz. odmični kot (angl. *Yaw*).

Da se LiDARska naprava ne bi v nedogled vrtela in se pri tem poškodovala, je z novo nadgradnjo strojne opreme dobila varovalke. V primeru, da se LiDARska naprava premakne izven meja delovnega področja, jo omenjene varovalke ustavijo. Prav zaradi naštetih novosti je LiDARska naprava pri izvajanju meritev omejena, na naslednja delovna področja:

- V odklonu (angl. *yaw*) od 0° do 350° (slika 2.19) in
- nagibu (angl. *tilt*) od -20° do 90° (slika 2.20).



Slika 2.19: Slika prikazuje omejeno delovno področje v smeri odklona (modra barva predstavlja dostopno delovno področje).



Slika 2.20: Slika prikazuje omejeno delovno področje v smeri nagiba (modra barva predstavlja dostopno delovno področje).

Poglavje 3

Programska oprema RhoLaaS za oddaljeno delo z LiDARskim sistemom

3.1 Uvod



Slika 3.1: Okence za informacije o aplikaciji *RhoLaaS*.

V tem poglavju so predstavljena splošna in podrobna navodila za uporabo programske opreme za oddaljeno uporabo LiDARskega sistema. Opisi so v obliki navodil in so namenjeni pretežno končnemu uporabniku. Omogočajo mu

lažjo uporabo aplikacije za pošiljanje in shranjevanje programov meritev, ter prikaz in shranjevanje dobljenih rezultatov meritev, ki so bile zajete s pomočjo LiDARskega sistema.

3.2 Meritev, program in rezultat meritve

V tem poglavju so predstavljeni osnovni pojmi s katerimi se uporabnik sreča pri delu z našim LiDARskim sistemom.

LiDARski strežnik omogoča spletno storitev naročanja meritve, ki temelji na SOAP protokolu. Uporabnik kot zahtevo na LiDARski strežnik pošlje **program meritve**, ki vsebuje vse potrebne podatke za to, da LiDARska naprava izvede **meritev**.

V naši aplikaciji je program meritve definiran kot objekt razreda *Program*, ki se ustvari na podlagi WSDL datoteke pridobljene iz LiDARskega strežnika.

Na sliki 3.2 so prikazani elementi meritve, ki so prav tako sestavni deli programa meritve. Program meritve najprej definirata elementa, ki sta namenjena njegovi identifikaciji. To sta element »*Author*«, ki predstavlja podpis avtorja programa meritve, ter element »*program ID*«, s katerim identificiramo program meritve pred zajemom izmerjenih podatkov iz LiDARske naprave.

Naslednja dva elementa določata nastavitve laserskega oddajnika, ki je eden izmed poglobitvinih delov LiDARske naprave. To sta element *EnergyLevel*, ki predstavlja intenziteto laserskega sunka (z vrednostjo med 1 in 20) ter element *PulsesPerMeasurement*, ki predstavlja število laserskih sunkov za posamezno meritev (vnose več kot 1 uporabimo, kadar želimo imeti meritve z manj šuma, saj se računa povprečna meritev).

Z elementom *measuringType* se določi tip meritve z naslednjimi vrednostmi: »*Data*«, kar pomeni da bomo iz LiDARske naprave zajemali celotne podatke (z ozadjem), »*Background*«, kar pomeni da bomo iz LiDARske naprave zajemali samo ozadje ali »*Data-Background*«, kar pomeni da bomo iz LiDARske naprave zajemali podatke brez ozadja.

Naslednji elementi določajo željeni čas in kraj meritve in sicer: element *ExecutionDateTime*, predstavlja željeni čas izvedbe meritve (datum in ura) ter elementi o zemljepisni širini (angl. *latitude*) in zemljepisni dolžini (angl. *longitude*) v formatu *DMS* (*Degree-Minute-Second*).

Elementi *YawStart*, *TiltStart*, *YawEnd*, *TiltEnd* so najpomembnejši podatki v programu meritve. Predstavljajo začetne in končne odmike ter nagibe za omejitvev področja, znotraj katerega želimo, da nam LiDARska naprava izvede meritve.

Z vrednostmi v elementih *YawResolution*, *TiltResolution* se določi odmična in nagibna gostota (število meritev v smeri odklona in nagiba), pri čemer za eno izmerjeno področje LiDARska naprava nato opravi skupaj *Yaw Resolution* x *Tilt Resolution* meritev.

V zadnjih dveh elementih *Repetitions* in *RepetitionDelay* se nahajata vrednosti, s katerima določimo število ponovitev, torej kolikokrat želimo ponoviti celoten niz meritve, in zakasnitev - koliko sekund naj aplikacija čaka med morebitnimi ponovitvami.

Rezultat izvedenega programa meritve in parametri programa samega, se zapišejo v datoteko formata LAS in sicer v njeno glavo in polja posameznih točk. Primeri polj v glavi datoteke LAS so: skupno število točk v datoteki, odmik do zapisa točk, »System Identifier«, »Generating Software«, ipd.

Prenos programa meritve na LiDARsko napravo in prenos izmerjenih vzorcev je določen s protokolom povezave med odjemalcem in LiDARsko napravo. Sledi razdelek v katerem je omenjeni protokol podrobneje opisan.



Slika 3.2: Struktura programa meritve.

3.3 Scenarij (primer) uporabe protokola

Za definicijo protokola in lažje razumevanje le tega, smo najprej določili primer scenarija, na katerem smo osnovali nadaljnji razvoj. Primer opisuje tako odjemalčevo kot strežnikovo stran in sicer:

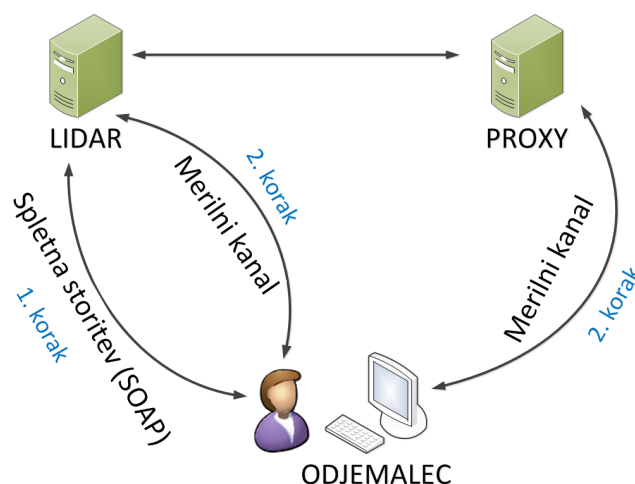
1. **Odjemalec** se dne 3.7.2011 ob 11:23, preko spletne storitve (SOAP), prijavi na LiDARski sistem.
2. **Strežnik**, kot odgovor na prijavo odjemalcu pošlje varnostni žeton, katerega bo potreboval za komunikacijo z LiDARskim strežnikom.
3. **Odjemalec** po prijavi, sporoči strežniku: »Dne 7.9.2011 ob 12:37 želim meritev. Izmerjeni podatki naj bodo brez ozadja. Meritev naj bo na GPS koordinatah 40°26'47" N zemljepisne širine in 79°58'36" W zemljepisne dolžine s prostorskim kotom (področje meritve), z začetnim odmikom -27° in začetnim nagibom 44°, ter z končnim odmikom 16° in končnim nagibom 33°. Meritev naj bo narejena z odmično gostoto 2 in nagibno gostoto 3. Ker želim imeti meritev z majhno količino šuma, naj bo število laserskih sunkov v vsako smer 3. Celotno meritev izvede dvakrat in med meritvami počakaj 10 sekund. Meritve posameznih točk izvede neposredno na robovih izbranega področja.«
4. **Strežnik** odgovori: »Zahteva sprejeta. Meritev bo na voljo ob času navedenem v programu meritve ter na IP naslovu 1.2.3.4 in vratih 5678.«
5. **Odjemalec** se ob določenem času priključi na merilni kanal (vtičnik angl. *socket*), ki se nahaja na prej vrnjenem naslovu 1.2.3.4/5678 (IP naslov/vrata).
6. **Strežnik** vpraša: »Varnostni žeton?«
7. **Odjemalec** odgovori z varnostnim žetonom in podatki na merilnem kanalu začnejo teči po vrstnem redu, ki ga določa protokol merilnega kanala.

Če je strežnikov odgovor negativen (pri 4. točki) to pomeni da strežnik ne bo mogel izvesti programa meritve. V tem primeru, odjemalec kot odgovor na zahtevo meritve prejme podatke za vir (angl. *data source*) in čas izvedbe kot negativno vrednost. Odjemalec zaprosi LiDARski strežnik za seznam zasedenosti, na podlagi katerega določi novi željeni čas izvedbe meritve. Nato odjemalec LiDARskemu strežniku, preko spletne storitve SOAP ponovno

pošlje program meritve. Po ponovnem naročilu storitve meritve se postopek, opisan od 4. do 7. točke, ponovi.

3.4 Protokol povezave z LiDARskim strežnikom

Protokol določa povezavo med odjemalcem (oddaljeno aplikacijo) in LiDARsko napravo s strežnikom. Kot sem že omenil, je protokol sestavljen iz dveh korakov, ki sta predstavljena na sliki 3.3. Prvi korak zajema naročanje storitve meritev oz. pošiljanje programa meritve na LiDARski strežnik. Izveden je z spletno storitvijo SOAP, ki jo ponuja LiDARski strežnik. Sledi mu drugi korak, ki je določen z merilnim kanalom (TCP/IP protokol). V drugem koraku se odjemalec, potem ko LiDARska naprava prične z izvajanjem meritve, poveže na LiDARski strežnik oz. na drugi vir (namestnik angl. *proxy*) iz katerega lahko zajema podatke, ki so rezultat meritve LiDARske naprave.



Slika 3.3: Diagram arhitekturne sheme LiDARskega sistema.

3.4.1 Namestnik - Proxy

Oddaljeni odjemalec lahko podatke zajete z LiDARsko napravo prejme tudi iz virov, ki niso LiDARski strežnik in sicer lahko njegovo nalogo prevzame namestnik. Naloga namestnika je, da transparentno distribuira podatke odjemalcem (oddaljenim strankam) in tako razbremeni LiDARski strežnik, ki ima omejeno pasovno širino (angl. *limited bandwidth*). Namestnik ne omogoča

upravljanja z LiDARsko napravo, temveč omogoča le vzpostavitev merilnega kanala (TCP/IP povezava), preko katerega se nato prenašajo podatki, ki so bili zajeti z LiDARsko napravo.

3.4.2 Spletna storitev SOAP

Spletna storitev vsebuje funkcije, ki odjemalcu omogočijo:

- avtentikacijo oddaljenega uporabnika in
- prenos programa meritve na LiDARski strežnik.

Program, ki ga pošlje odjemalec LiDARskemu strežniku s pomočjo spletne storitve je bil opisan v razdelku 3.2. V nadaljevanju so predstavljene glavne funkcije spletne storitve, in sicer:

- `string getAuthToken(string username, string password)`
Funkcija, ki omogoča prijavo v LiDARski sistem. Če je prijava uspešna LiDARski strežnik odjemalcu (oddaljeni stranki) vrne varnostni žeton.
- `DataSource addProgram(string authToken, Program program)`
Funkcija, ki omogoča, da odjemalec naloži program meritve (definiran v razdelku 3.2) na LiDARski strežnik. Program meritve je tipa `Program`, ta se ustvari s pomočjo dane WSDL datoteke iz spletnega servisa, ki ga ponuja LiDARski strežnik. Če se dodan program lahko izvede, se ustvari instanca razreda *DataSource*, v katerem so zapisani podatki za naslov merilnega kanala (IP naslov/vrata). Če pa dodanega programa LiDARski strežnik ne more izvesti, so omenjene vrednosti negativne.
- `ListOfBusyTimes getBusyList(string AuthToken)`
Funkcija, ki omogoča pridobitev seznama zasedenosti LiDARske naprave (ob klicu te funkcije se ustvari instanca razreda *ListOfBusyTimes*). S pomočjo teh podatkov oddaljeni uporabnik določi željeni čas izvedbe, ko bo LiDARska naprava prosta.

3.4.3 Merilni kanal

Merilni kanal je namenjen prenosu izmerjenih podatkov (meritev). Gre za vrsto povezave, po kateri se iz LiDARske naprave preko TCP/IP protokola do odjemalca prenašajo podatki zajeti z LiDARsko napravo.

Vsi prenosi iz smeri LiDARskega strežnika po merilnem kanalu so oblike prikazane na sliki 3.4. Vse vrednosti, ki se prenašajo po merilnem kanalu so predstavljeni z **32-bitnimi celimi števili** (angl. *integer*).

```
<Start Of Record>  
  <Type Of Record=Start/End/Measurement>  
    <Record Content>
```

Slika 3.4: Oblika prenosa po merilnem kanalu.

Odjemalec se mora pred zajemom podatkov na strežnik avtenticirati z žetonom. To stori tako, da žeton zapiše na odprto povezavo vtiča (angl. *socket*). Če je žeton pravilen, odjemalec lahko prične z branjem podatkov iz merilnega kanala, v nasprotnem primeru strežnik povezavo prekine.

Če se povezava merilnega kanala ohrani, odjemalec takoj po avtentikaciji iz merilnega kanala prebere vrednost SOR (angl. *Start of record*), ki nam pove zgolj to, da ji sledi naslednja kontrolna vrednost TOR (angl. *Type of record*), ki v sebi nosi vrednost za pričetek programa (vse kontrolne vrednosti so predstavljene v tabeli 3.1). Sledijo podatki iz merilnega kanala, ki predstavljajo glavo programa oz. splošne informacije o trenutno izvajajočem se programu (slika 3.5). Gre za podatke o trenutnem geografskem položaju LiDARske naprave (koordinate zemljepisne dolžine in širine) ter identifikacijsko vrednost programa meritve, ki se trenutno izvaja. Lahko se namreč zgodi, da začne odjemalec z branjem rezultata meritve na sredi izvajanja programa.

Glavi meritve sledijo podatki, ki so bili zajeti z LiDARsko napravo, na kar nas opozorita naslednja konstanta SOR ter vrednost *Measurement* konstante TOR (tabela 3.1). LiDARska naprava namreč za vsak oddan sunek (v eno smer) vrne 256 vzorcev. Vsak vzorec predstavlja vrednost intenzitete odbitega svetlobnega sunka z dolžinsko ločljivostjo 20,3 m.

Povzeto povedano: LiDARska naprava, preko merilnega kanala, za vsak oddan sunek (N različnih smeri - slika 3.7) odjemalcu pošlje obe kontrolni vrednosti (SOR, TOR z vrednostjo *Measurement*), naklonski in nagibni kot, ter vseh 256 vrednosti, ki predstavljajo intenzitete odbitega žarka (slika 3.6). Konec programa meritve označimo s kontrolno vrednostjo SOR in vrednostjo *End* kontrolne vrednosti TOR.

SOR	TOR	ID programa	Zemljepisna širina	Zemljepisna dolžina	Nadmorska višina naprave
-----	-----	-------------	--------------------	---------------------	--------------------------

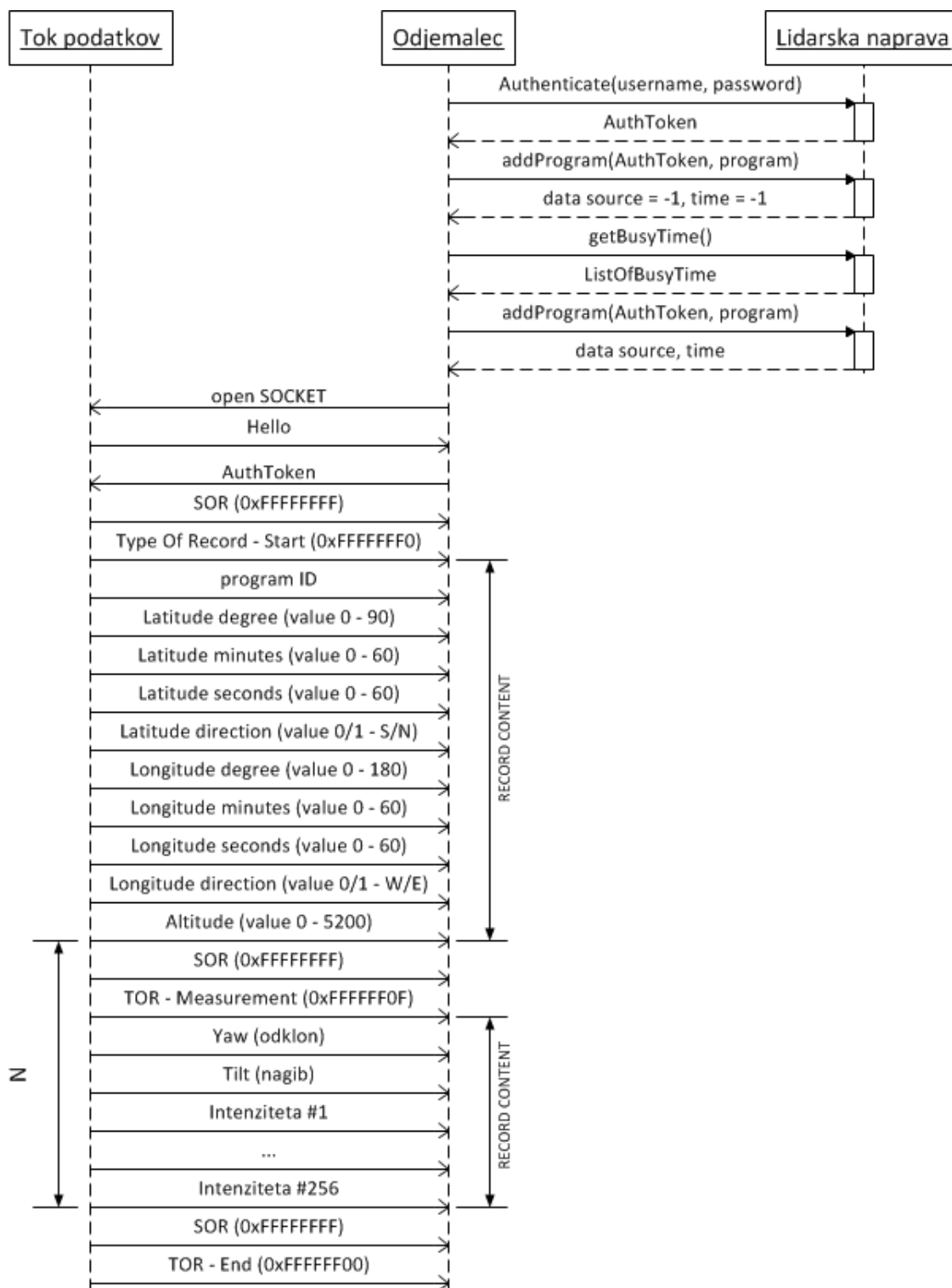
Slika 3.5: Zaporedje splošnih informacij izvedene meritve.

SOR	TOR	Odklon (angl. <i>yaw</i>)	Nagib (angl. <i>tilt</i>)	256 x vrednosti intenzitete
-----	-----	----------------------------	----------------------------	-----------------------------

Slika 3.6: Oblika meritve dobljena po merilnem kanalu.

Vrsta podatka	Vrednost
SOR (angl. <i>Start Of Record</i>)	0xFFFFFFFF
TOR (angl. <i>Type Of Record</i>) - Start	0xFFFFFFFF0
TOR (angl. <i>Type Of Record</i>) - Measurement	0xFFFFFFFF0F
TOR (angl. <i>Type Of Record</i>) - End	0xFFFFFFFF00

Tabela 3.1: Konstantne vrednosti za določanje vrste prebranega podatka.



Slika 3.7: Diagram protokola za komunikacijo z LiDARskim sistemom.

3.5 Opis in izvedba aplikacije RhoLaaS

Pri razvoju programske opreme za oddaljeno uporabo LiDARskega sistema, smo izhajali iz predhodno že razvite programske opreme, ki je bila napisana v programskem jeziku C#/.net. Obsegala je kontrole krmilnika za obračanje naprave, različne tipe meritve, shranjevanja in iskanja podatkov.

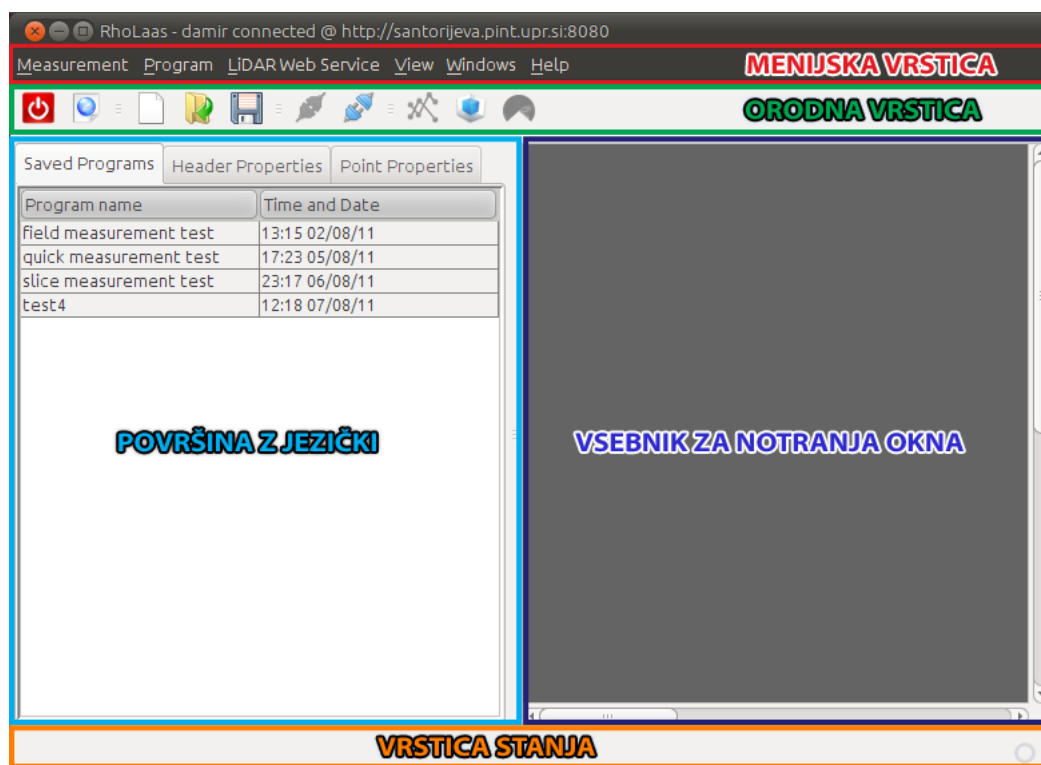
Za razliko od predhodne, novejša aplikacija omogoča oddaljeno uporabo LiDARske naprave, hranjenje meritev oz. z LiDARsko napravo zajetih točk v standarden format za LiDARske naprave (LAS format), hranjenje programov meritev ter več načinov prikazovanja podatkov iz LiDARske naprave.

Programski oprema je razvita v programskem jeziku Java. Zanj smo se odločili iz več razlogov. Java je trenutno najbolj popularen jezik, saj zasnova jezika omogoča hitro prevajanje, hitro iskanje in odpravljanje napak. Največja odlika Jave pa je neodvisnost od platforme - Java aplikacije tečejo na katerekoli operacijskem sistemu (Linux, Windows in Unix). Uporabniški vmesnik je razvit s pomočjo komponent javanske knjižnice Swing.

Še eno orodje s katerim smo se srečali pri razvoju aplikacije za delo z datotekami formata LAS je knjižnica LibLAS [38]. LibLAS za razvijalce programske opreme ponuja številne funkcije za vključevanje podpore LAS formata v svoje programske opreme. Je odprtokodna programska oprema, ki vsebuje mnogovrsten programski vmesnik (angl. *Multiple APIs*), kateri vključuje programske vmesnike za programske jezike C, C++, Ruby, Python in ogrodje .NET. Ker LibLAS ne ponuja programskega vmesnika za programski jezik Java, smo za delo z rezultati izvedenih meritev, ki so shranjeni v datotekah formata LAS, razvili lastno orodje, ki temelji prav na tem programskem jeziku.

3.5.1 Osnovno okno aplikacije

Osnovno okno v aplikaciji (prikazano na sliki 3.8) je namenjeno za pregled rezultatov meritev (branje datotek formata LAS), pa tudi za pregled shranjenih programov meritev. Predstavlja oporo za vsa ostala okna, ki lahko vsebujejo različne grafične kontrole ali različne tipe programov meritev. Poleg preglednega delovnega področja, osnovno okno v aplikaciji omogoča dostop do vseh funkcij, ki jih omogoča naša aplikacija. Na sliki 3.9 je prikazan diagram grafičnega vmesnika naše aplikacije, ki natančneje opisuje zgradbo grafičnega vmesnika in hkrati pove kako dostopati do željene funkcije v naši aplikaciji.



Slika 3.8: Osnovno okno programa *RhoLaaS*.

Na sliki 3.8 je predstavljeno osnovno okno naše aplikacije, pri čemer je vsak njen osnovni označen in obarvan. V nadaljevanju bom s pomočjo te slike opisal vse komponente osnovnega okna aplikacije *RhoLaaS*.

Skrajno zgoraj se nahaja **menijska vrstica**, ki nam služi za izbor vseh akcij, ki jih ponuja naša aplikacija. V meniju *Measurement* so na voljo funkcije za delo z rezultati meritev, in sicer:

- *New Scenario*, ki nam omogoča, da si ustvarimo novo delovno področje - zaprejo se vsa odprta okna, tabele, ki prikazujejo podatke iz prebrane LAS datoteke ter prekinemo povezavo s spletnim servisom (pozabimo *authToken*).
- *Open LAS*, s katerim odpremo predhodno shranjene meritve v formatu LAS ter
- *Save in Save As*, s katerima shranimo trenutno meritev tj. meritev, ki je sicer prikazana v podoknih *Header Properties* in *Point Properties*.

Naslednji meni v menijski vrstici z imenom *Program* vsebuje ukaze za delo s programi meritev. Z *New*, *Open*, *Save* in *Save As* lahko ustvarimo, odpremo in shranimo različne programe meritev (opisani v nadaljevanju in sicer v razdelku 3.5.2):

- hitra, osnovna meritev (*Quick Measurement*),
- meritev v rezini (*Slice Measurement*) in
- najbolj splošna meritev, meritev v pravokotnem območju (*Field Measurement*).

Poleg zgornjih možnosti za delo z lokalnimi programi meritev, imamo pri vzpostavitvi povezave s spletno storitvijo LiDARske naprave, na voljo tudi možnost *Open Executing program*, ki nam ob zasedenosti LiDARske naprave omogoča pregled programa meritve, ki ga LiDARska naprava trenutno izvaja. Varnostni žeton, ki je potreben za klic te funkcije se pridobi s predhodno avtentikacijo na LiDARski strežnik.

LiDAR Web Service je meni v menijski vrstici, z dvema izbirama, ki nam omogočata avtentikacijo oz. vzpostavitev in prekinitev povezave s spletno storitvijo. Ob uspešni vzpostavitvi povezave prejmemo varnostni žeton (*AuthToken*), ki ga za nadaljnje delo shranimo. Ob prekinitvi povezave varnostni žeton za trenutno sejo izgubi veljavnost.

V meniju *View* najdemo preklopne izbire, s katerimi lahko skrijemo oziroma prikažemo večje (glavne) dele osnovnega okna aplikacije, in sicer statusno vrstico, orodno vrstico, povišino z jezički za shranjene programe, lastnosti točk in lastnosti glave odprte LAS datoteke. Poleg naštetih možnosti se ob zagonu aplikacije tukaj namestijo različni videzi celotnega grafičnega vmesnika, med katerimi lahko med samo uporabo aplikacije preklapljam.

V naslednjem meniju z imenom *Windows* najdemo izbire za upravljanje z okni, ki se nahajajo v vsebniku za notranja okna. Tako odprta okna lahko razporedimo v prekrivajočo se vrsto (*Cascade Mode*) ali drugo ob drugem (*Tile*), lahko pa jih tudi skupaj zapremo in pomanjšamo.

Zadnji meni v menijski vrstici pod imenom *Help* vsebuje izbiro za prikaz okna (slika 3.1), ki nam poda informacijo avtorju in nekaj splošnih informacij programa.

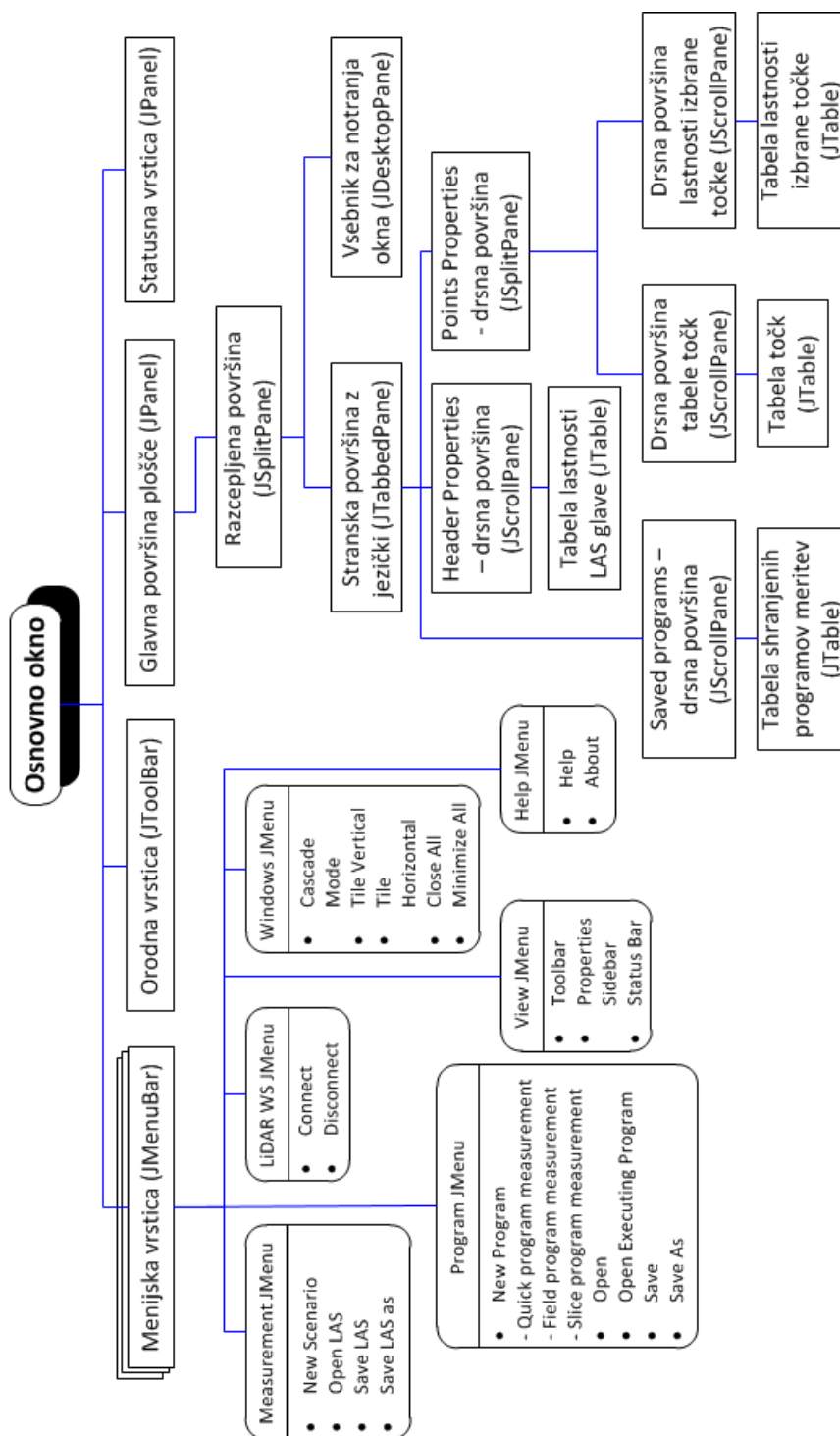
Pod menijsko vrstico se nahaja **orodna vrstica**, ki nam služi za lažjo navigacijo in hitrejši dostop do operacij, ki se najpogosteje uporabljajo. Orodno vrstico ter operacije, ki so prikazane v obliki gumbov lahko vidimo na sliki 3.8.

Večji del delovnega področja zaseda stransko okno, ki vsebuje **površino z naslednjimi jezički**:

- *Saved Programs*. Vsebuje seznam predhodno shranjenih programov meritve, iz katerega lahko izbran program meritve odpremo (z dvoklikom) ali po potrebi izbrišemo. S klikom na desni miškin gumb, se pojavi meni (slika 3.10), s pomočjo katerega lahko izberemo ali bomo izbran program odprli ali ga izbrisali.
- *Header Properties*. Vsebuje vse podatke glave, ki jih predpisuje standard datoteke formata LAS 1.3.
- Jeziček *Point Properties* je s pomočjo razcepljene površine (angl. *Split Pane*) razdeljen na dve podpovršini. Zgornji prostor zaseda tabela oz. seznam vseh točk iz prebrane LAS datoteke. V spodnjem prostoru pa se nahaja tabela lastnosti točke, ki je bila izbrana iz zgornjega seznama prebranih točk datoteke LAS.

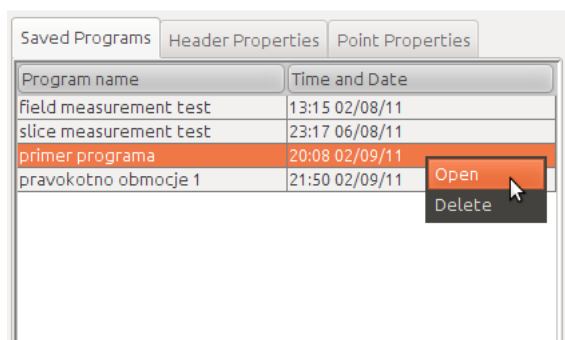
Na desni strani okna aplikacije se nahaja **vsebnik za notranja okna** (angl. *Desktop Pane*). Ta gradnik vsebuje vsa okna, ki se prikazujejo pri delu z našo aplikacijo in na pregleden način omogoča nadzor nad vsemi okni v trenutnem delovnem področju.

V spodnjem robu okna se nahaja **vrstica stanja**. Desno se pri obdelavi datotek formata LAS prikaže časovno stanje poteka.

Slika 3.9: Diagram grafičnega vmesnika osnovnega okna programa *RhoLaaS*.

3.5.1.1 Arhiv programov meritev

Aplikacija omogoča hranjenje programov meritev. Vsi predhodno shranjeni programi so prikazani v zavihku *Saved Programs* (slika 3.10).



Slika 3.10: Arhiv programov meritev.

Shranjeni programi meritev se nahajajo v lokalnem pomnilniku. Zapisani so v datoteki formata XML in vsebujejo elemente, ki so bili že opisani v razdelku 3.2.

Na sliki 3.11 je prikazan primer shranjenega programa meritve, ki vsebuje elemente podrobneje opisane v prilogi A.3.

Poleg shranjenih programov meritev se v indeksni datoteki hrani tudi njihov seznam, s pomočjo katerega se ob zagonu aplikacije izpolni zavihek arhiv programov (slika 3.10). Po vsakem brisanju ter shranjevanju novega programa se seznam programov posodobi.

Vsi shranjeni programi meritev ter njihov seznam so shranjeni v lokalnem pomnilniku, na lokaciji določeni z relativno potjo */Archive/*, glede na korenski direktorij aplikacije.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <program>
3   <programType>field measurement</programType>
4   <name>primer programa</name>
5   <programID>2147483647</programID>
6   <energy>20</energy>
7   <measuringType>Data-Background</measuringType>
8   <latitude>45.548092</latitude>
9   <longitude>13.726816</longitude>
10  <yawStart>-1.8732</yawStart>
11  <tiltStart>45.548092</tiltStart>
12  <yawEnd>13.726816</yawEnd>
13  <tiltEnd>0.2818</tiltEnd>
14  <yawResolution>2</yawResolution>
15  <tiltResolution>2</tiltResolution>
16  <pulsesPMsm>21</pulsesPMsm>
17  <repetition>0</repetition>
18  <delay>30</delay>
19  <date>08 08 11</date>
20  <time>17 14</time>
21 </program>
```

Slika 3.11: Format shranjenega programa meritve.

3.5.1.2 Lastnosti glave datoteke formata LAS

Ob branju datotek formata LAS se prebrani podatki prikažejo v jezičkih:

- *Header Properties* ter
- *Point Properties*.

V jezičku *Header Properties* (slika 3.12) so prikazani podatki glave LAS datoteke, v skladu z verzijo 1.3 [18]. Kot je bilo že omenjeno v prejšnjem poglavju (2.3), lahko iz glave LAS datoteke preberemo podatke, ki so skupni vsem točkam, shranjenim v tej LAS datoteki. Tako lahko uporabnik pogleda v okviru katerega projekta je nastala datoteka (angl. *Project ID*), s pomočjo katere naprave in programske opreme je bila ustvarjena ipd.

Kot je bilo že omenjeno, nam LiDARska naprava kot **rezultat** za vsak oddan sunek vrne 256 14 bitnih vzorcev. Če upoštevamo, da LiDARska naprava vzorči signal odbojev s frekvenco 7,37 MHz in omogoča detekcijo do razdalje 5,2 km, dobljeni vzorci predstavljajo vrednost intenzitete odbitega svetlobnega sunka z dolžinsko ločljivostjo 20,3 m. Rezultat meritve se zapiše v datoteko LAS formata 1.3. Pri zapisu omenjene datoteke se sproti izračunajo vsi potrebni (obvezni) podatki (glej poglavje 2.3), ki jih je potrebno zapisati v glavo datoteke ter v polja posameznih točk.

Za lažje delo z datotekami LAS, ki so bile ustvarjene z našim LiDARskim sistemom se v polji »*Generating Software*« in »*System Identifier*«, ki se nahajata v glavi datoteke LAS, dodajo informacije o našem LiDARskem sistemu, in sicer ime LiDARske naprave (primer »*PINT LiDAR*«) ter ime aplikacije in tip programa meritve (primer »*RhoLaaS-slice*«), ki je bil uporabljen pri izvedbi meritve.

Header Properties	Value
File Signature ("LASF")	LASF
File Source ID	0
Reserved	0
Project ID - GUID data 1	0
Project ID - GUID data 2	0
Project ID - GUID data 3	0
Project ID - GUID data 4	
Version Major	1
Version Minor	1
System Identifier	PINT LIDAR
Generating Software	RhoLaaS-slice
File Creation Day of Year	6
File Creation Year	1
Header Size	227
Offset to point data	227
Number of variable length re...	0
Point Data Format ID (0-99 for...	1
Point Data Record Length	28
Number of point records	8192
Number of points by return	[0, 0, 0, 0, 0]
X scale factor	0.0010
Y scale factor	0.0010
Z scale factor	0.0010
X offset	0.0
Y offset	0.0
Z offset	0.0
Max X	45.594841604148975
Min X	45.54827094776766
Max Y	13.72707983215889
Min Y	13.726816062579617
Max Z	21.4021441541968
Min Z	0.01780235601951143

Slika 3.12: Lastnosti glave datoteke LAS.

3.5.1.3 Lastnosti točke iz datoteke formata LAS

Jeziček *Point Properties* je predstavljen z razcepljeno površino, in sicer ga sestavljata dve tabeli (slika 3.13). V zgornji tabeli so po straneh prikazane vse prebrane točke iz datoteke formata LAS. Kapaciteta vsake strani v tabeli se določi v odvisnosti od števila vseh točk v datoteki LAS, torej če je število vseh točk visoko bo tudi vsaka stran v tabeli vsebovala več točk. Po posameznih straneh se lahko pomikamo s pritiskom na gumb »page up« ali »page down« oz. z pritiskom na ikone nad oz. pod stranskim drsnikom tabele seznama točk.

V najbolj levem stolpcu *Points* so zapisani indeksi vsake točke. V naslednjih treh stolpcih (od leve proti desni) pa so z oznakami *X*, *Y* in *Z* navedene *x*, *y* in *z* vrednosti koordinat prebranih točk. V zadnjem stolpcu je prikazana še vrednost intenzitete vrnjenega žarka posamezne točke.

Z izborom točke iz zgornje tabele se preberejo vse preostale točke (256 – 1 točk), ki imajo isto smer meritve kot izbrana točka. S podatki prebranih točk se izriše graf intenzitete (predstavljen v poglavju 3.5.3.1).

Izbrano točko iz zgornje tabele je možno podrobneje analizirati v spodnji tabeli, ki prikazuje vse lastnosti točke po standardu LAS 1.3 [20]. V levem stolpcu *Point Properties* so prikazane lastnosti izbrane točke, v desnem *Value* pa so prikazane vrednosti posameznih lastnosti.

The screenshot shows a software window with two main sections. The top section is a table with columns: Point Index, X, Y, Z, and Intensity. The bottom section is a table with columns: Point Properties and Value.

Point Index	X	Y	Z	Intensity
0	-1,912900	-3,386750	1,071000	65535,000000
1	-1,916900	-3,382750	1,075000	65535,000000
2	-1,916900	-3,382750	1,072000	65535,000000
3	-1,914900	-3,384750	1,073000	65535,000000
4	-1,927900	-3,381750	0,914000	65535,000000
5	-1,926900	-3,380750	0,965000	65535,000000
6	-1,926900	-3,380750	1,066000	65535,000000
7	-1,923900	-3,381750	1,031000	65535,000000
8	-1,922900	-3,380750	1,034000	65535,000000
9	-1,929900	-3,378750	1,041000	65535,000000
10	-1,926900	-3,379750	1,007000	65535,000000
11	-1,925900	-3,379750	1,013000	65535,000000
12	-1,925900	-3,378750	0,971000	65535,000000

Point Properties	Value
X	-1,917
Y	-3,383
Z	1,075
Intensity	65.535
Return Number	0
Number of Returns(givenpulse)	0
ScanDirectionFlag	0
Edge of FlightLine	0
Classification	0
Scan Angle Rank(-90 to +90) – Leftside	0
UserData	0
Point Source ID	0
Red	61.200
Green	61.200
Blue	61.200

Slika 3.13: Podokno, lastnosti točk iz datoteke LAS.

3.5.2 Merjenje (tipi programov meritev)

Naša aplikacija omogoča tri tipe programov meritev, in sicer:

- **Program hitre (osnovne) meritve** omogoča meritev samo v eni izbrani smeri, torej s to meritvijo sprejmemo največ 1×256 točk. Takšno meritev smatramo kot **enostavno meritev**.
- **Program meritve v rezini** prav tako vsebuje več enostavnih (osnovnih) meritev, le da so te razporejene v rezino oz. v enakomernih kotnih razmikih od začetne do končne izbrane pozicije.
- **Program meritve v pravokotnem območju** je najbolj splošen program meritve in vsebuje več enostavnih meritev v izbranem pravokotnem območju.

Našteti programi meritve se med seboj razlikujejo le po določanju območja meritve. V nadaljevanju bodo predstavljeni vsi trije. Zaradi lažjega razumevanja, bo najprej predstavljen program meritve v izbranem pravokotnem območju. Pri naslednjih dveh tipih meritve bom predstavil zgolj razlike glede na njega.

3.5.2.1 Meritev v pravokotnem območju

Namen uporabe

Program meritve v pravokotnem območju uporabljamo, ko želimo opraviti več enostavnih meritev v nekem širšem pravokotnem območju. Najpogosteje se uporablja prav ta program meritve, saj je zaradi različnih odstopanj naprave (npr. v geografski poziciji) zelo težko pridobiti uporabne podatke z enim samim sunkom LiDARske naprave oziroma z eno samo enostavno meritvijo. Željeno ciljno območje izmerimo, tako da si izberemo neko pravokotno področje merjenja, znotraj katerega se nahaja tudi željeni cilj. Na sliki 3.14 je prikazano okno programa meritve v pravokotnem območju. V naši aplikaciji, lahko do njega dostopamo po naslednji poti:

```
Program : New : Field Program Measurement
```

Postopek uporabe

Postopek uporabe te meritve je v osnovi podoben ostalima dvema meritvama. Njihove razlike so navedene pri opisu postopka uporabe za posamezno meritev.

The screenshot shows a software window titled "Archive/programpravokotno obmocje 11314993011888.xml". The main content is a "Configure Field Measurement" dialog. It has the following fields and options:

- Name:** Input field containing "pravokotno obmocje 1".
- Yaw start:** Input field containing "-1.8732".
- Yaw end:** Input field containing "13.726816".
- Tilt start:** Input field containing "45.548092".
- Tilt end:** Input field containing "0.2818".
- Yaw resolution:** Input field containing "2".
- Tilt resolution:** Input field containing "2".
- Pulses/Msm.:** Input field containing "21".
- Energy:** Input field containing "20".
- Repetitions:** Input field containing "0".
- Delay [s]:** Input field containing "30".
- Measurement Type:** Radio buttons for "Data", "Background", and "Data - Background" (which is selected).
- Configure LIDAR Position:**
 - Latitude:** Input field containing "45.548092".
 - Longitude:** Input field containing "13.726816".
 - Buttons: "Open Google Maps" and "Show Map".
- Configure Execution Time:**
 - Date:** Input field containing "10.8.2011".
 - Time:** Time selection field containing "20:06:00".

At the bottom of the dialog are three buttons: "Save Program", "Send Program", and "Cancel".

Slika 3.14: Okno programa meritve v pravokotnem območju.

Meritev poteka **v treh korakih** (prva dva sta prikazana na sliki 3.3). **Prvi korak** zajema naročanje spletne storitve meritev, ki jo ponuja LiDARski strežnik. Ta korak ureja protokol SOAP, ki je opisan v razdelku 3.4.2. Preden se program meritve pošlje na LiDARski strežnik, ga je potrebno izpolniti (povezava med programom meritve in polji v datoteki formata las prikazuje tabela A.8). Opisan postopek je predstavljen v točkah od 1 do 9, in sicer:

1. Najprej si izberemo ime programa meritve za kasnejšo identifikacijo.
2. Nadalje izberemo tip meritve. Na voljo imamo, možnost da zajamemo zgolj podatke, zgolj ozadje, ali pa se odločimo za razliko podatkov in ozadja. Po navadi uporabljamo zadnjo možnost.
3. V naslednja polja vpišemo vrednosti za nastavitve laserskega oddajnika, ki je eden izmed poglobitvenih delov LiDARske naprave. V polje *Energy* vpišemo intenziteto laserskega sunka (z vrednostjo med 1 in 20). Nato izberemo še število laserskih sunkov za posamezno meritev (Pulses / Msm.). Vnose več kot 1 uporabimo, kadar želimo imeti meritve z manjšuma.

4. V polja *Yaw Start* (začetni odklon), *Tilt Start* (začetni nagib), *Yaw End* (končni odklon) ter *Tilt End* (končni nagib) vnesemo omejitve področja, na katerem želimo meriti.
5. Vnesemo še števili meritev v smeri odklona in nagiba. Iz njunega produkta se določi število opravljenih (*Yaw Resolution*) x (*Tilt Resolution*) meritev.
6. V polje *Repetitions* (ponovitve) vpišemo koliko področji želimo zajeti poleg prvega, torej kolikokrat želimo ponoviti celoten niz meritve. V polje *Delay* pa vpišemo, koliko sekund naj aplikacija čaka med posameznimi ponovitvami.
7. V polja *Latitude* in *Longitude* vnesemo željeni geografski položaj LiDARske naprave med izvajanjem meritve. Ta točka predstavlja koordinatno izhodišče za izračun geografske pozicije izmerjenih točk. S pritiskom na gumb *Open Google Maps* ali *Show Map* se nam odpre zemljevid, s katerim lažje določimo vrednosti polj *Latitude* in *Longitude*. Funkciji za prikaz zemljevidov so podrobneje opisani v razdelku 3.5.2.4.
8. V polja *Date* in *Time* vpišemo željeni čas za izvedbo meritve.
9. Trenutni program pošljemo na LiDARski strežnik s pritiskom na gumb *Send Program*. Pred pričetkom izvajanja programa meritve, se med našim programom in LiDARskim strežnikom vzpostavi merilni kanal, po katerem sprejmemo podatke s pomočjo LiDARske naprave izmerjenih točk.

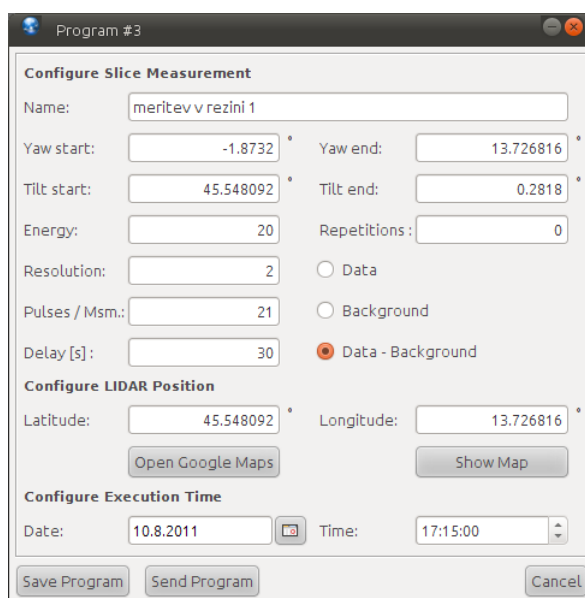
Drugi korak meritve obsega zajem podatkov, ki so rezultat izvedene meritve in shranjevanje dobljenih rezultatov v datoteko formata LAS. Urejen je s protokolom merilnega kanala (TCP/IP protokol), ki je opisan v razdelku 3.4.3. V naslednjih točkah je predstavljen tudi podroben postopek:

10. Ob vzpostavitvi merilnega kanala najprej preberemo ID programa (identifikacijska številka), ki ga primerjamo z identifikacijsko številko našega programa meritve. Poleg ID programa preberemo tudi koordinate *Latitude*, *Longitude* ter *Altitude*, ki nam povedo, kakšen je trenutni geografski položaj LiDARske naprave. S pomočjo prejetih koordinat in Vincentyjeve formule izračunamo geografski položaj izmerjenih točk, katerega kasneje zapišemo v datoteko formata LAS. V programu meritve se popravita polji za pozicijo, *Latitude* in *Longitude*.

11. Ko sprejmemo vse točke se pojavi opozorilno okno z vprašanjem, ali želimo prejete podatke shraniti v standard datoteke LAS formata 1.1 [20]. Če se odločimo za zapis točk v datoteko LAS, izberemo še lokacijo, kjer bomo datoteko hranili.

Zadnji, **tretji korak** zajema analizo in vizualizacijo prejetih oz. shranjenih podatkov iz LiDARske naprave. Po tem, ko smo datoteko formata LAS shranili, se ta takoj odpre, tako da je možno podatke za vse zajete točke trenutnega programa meritve, podrobneje analizirati. Rezultat izvedene meritve, s pomočjo programa meritve v pravokotnem območju, je možno prikazati le z 3D grafom in grafom intenzitete, ki sta podrobneje predstavljena v razdelku 3.5.3.

3.5.2.2 Meritev v rezini



Slika 3.15: Okno programa za meritev v rezini.

Namen uporabe

Kot je bilo že omenjeno meritev v rezini sestavlja več osnovnih (hitrih) meritev. Meritev po tem programu poteka z zajemom podatkov v enakomernih kotnih razmikih na poti od začetne (začetek meritve) do končne točke (zadnja meritev). Takšna meritev je priročna v primeru, ko želimo iz meritev konstruirati

sliko, podobno radarski. Na sliki 3.15 je prikazano okno programa za meritve v rezini. V naši aplikaciji, lahko do njega dostopamo po naslednji poti:

Program : New : Slice Program Measurement

Postopek uporabe

Postopek uporabe meritve v rezini je v osnovi precej podoben postopku meritve v pravokotnem območju. V **prvem koraku** se programa meritve razlikujeta le v poljih za število meritev v odklonu ter nagibu. Pri programu meritve v rezini ti dve polji nadomesti polje za število meritev v rezini - polje *Resolution*. **Drugi korak** pri postopku meritve v rezini se od drugega koraka postopka meritve v pravokotnem območju ne razlikuje, medtem, ko se **tretji korak** razlikuje zgolj v tem, da je mogoče rezultate meritve vizualizirati z vsemi tremi grafi, in sicer: 3D grafom, grafom intenzitete in grafom v rezini (predstavljeni v razdelku 3.5.3).

3.5.2.3 Hitra, osnovna meritve

Slika 3.16: Okno za hitro, osnovno meritve.

Namen uporabe

Osnovno meritev uporabljamo takrat, ko želimo izmeriti vrednosti samo v eni določeni smeri. Podatki, ki jih prejmemo se lahko predstavijo v grafu intenzitete v odvisnosti od razdalje ter v tridimenzionalnem grafu. Vsi ostali tipi programov meritev temeljijo na tem osnovnem programu meritve. Na sliki 3.16 je prikazano okno programa hitre (osnovne) meritve, do katerega lahko v naši aplikaciji, dostopamo po naslednji poti:

Program : New : Quick Program Measurement

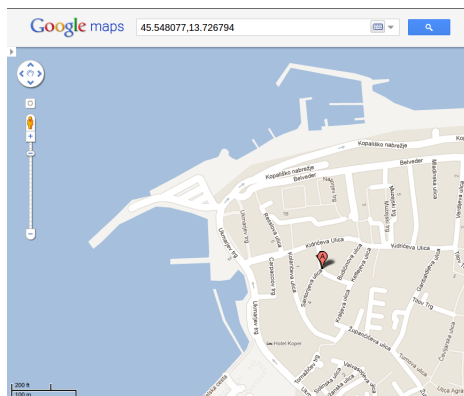
Postopek merjenja

Postopek uporabe programa za hitro (osnovno) meritev je najbolj preprost in za razliko od prejšnjih dveh programov, je v **prvem koraku** meritve potrebno določiti le eno smer meritve. Razlika je tudi v tem, da ta program ne vsebuje polj za resolucijo, saj se z uporabo tega programa izvede meritev v samo eni določeni smeri, ki jo določimo v poljih *Yaw*, *Tilt*. **Drugi korak** je enak kot pri prejšnjih dveh meritvah. **Tretji korak** je enak tretjemu koraku meritve v pravokotnem območju, rezultate obeh tipov meritev je možno prikazati le z 3D grafom in grafom intenzitete.

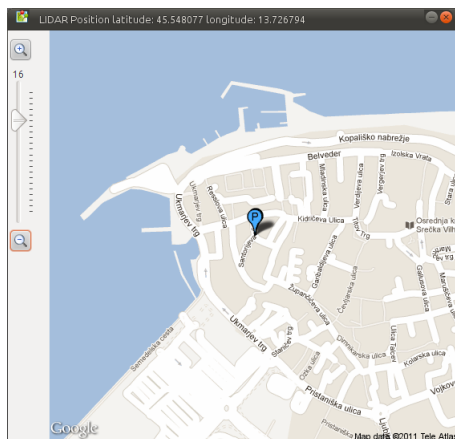
3.5.2.4 Prikaz geografskih podatkov iz programa meritev

Za lažjo določitev geografskega položaja, kje se bo LiDARska naprava med izvajanjem meritve nahajala, sta v programu meritev na voljo dve možnosti. Ob pritisku na gumb *Open Google Maps* se nam prikaže privzeti spletni brskalnik na naslovu z zemljevidom *Google Maps* [39]. S pomočjo zemljevida poiščemo poljubno lokacijo, ter dobljene geografske koordinate prenesemo v našo aplikacijo.

Druga možnost pa nam pravzaprav ne zagotavlja določanja geografskih točk, temveč samo preverjanje vnesenih koordinat v polja *Latitude* in *Longitude*. O pritisku na gumb *Show Map*, se odpre novo okno (slika 3.18), katerega večji del predstavlja statična slika pridobljena iz spletnega zemljevida *Google Maps*. Na levi strani okna se nahaja drsnik s pomočjo katerega lahko povečujemo in zmanjšujemo zemljevid.



Slika 3.17: Privzeti spletni brskalnik na naslovu GoogleMaps.



Slika 3.18: Okno za prikaz koordinat geografsko širino in dolžino (latitude, longitude).

3.5.3 Vizualizacija meritev

Vizualizacija meritev omogoča lažjo analizo izmerjenih točk. Za prikaz rezultatov meritev oz. točk prebranih iz datoteke formata LAS so v naši aplikaciji na voljo trije grafi, in sicer:

- graf intenzitete,
- graf v rezini in
- 3D prikaz točk.

Z grafom intenzitete in grafom v rezini je možno prikazati rezultate meritev oz. točke iz datotek formata LAS, zajetih in ustvarjenih le z našim LiDARskim sistemom. Za razliko od grafa intenzitete in grafa v rezini, 3D graf prikazuje vse datoteke formata LAS, ne glede na aplikacijo ali LiDARski sistem s katerim je bila ustvarjena. V razdelku 3.5.1.2 je opisano kako datoteke formata LAS, ki so bile ustvarjene z našo aplikacijo, ločimo od ostalih (»tujih«) LAS datotek. V nadaljevanju so podrobneje prikazani grafi, ki jih omogoča naša aplikacija.

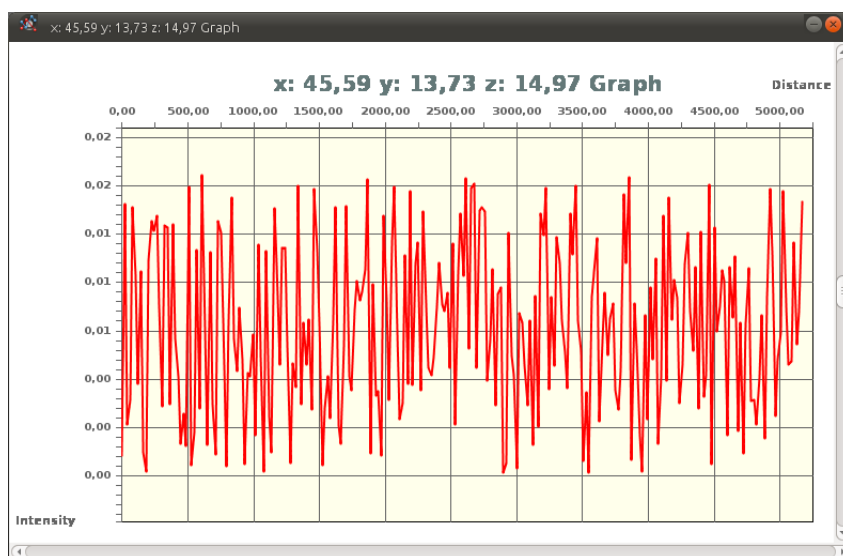
3.5.3.1 Graf intenzitete

Z grafom intenzitete lahko prikažemo točke, ki so bile pridobljene s pomočjo vseh treh tipov programa meritve (meritev v pravokotnem območju predstavljena v razdelku 3.5.2.1, meritev v rezini predstavljena v razdelku 3.5.2.2 in hitra meritev predstavljena v razdelku 3.5.2.3).

Za prikazovanje množice točk (skupaj 256 točk) s skupno smerjo meritve, ki so bile pridobljene iz LiDARskega strežnika, je bil razvit razred *Measurement Graph*, čigar izgled lahko vidimo na sliki 3.19.

Grafična kontrola je napisana s pomočjo knjižnice za dvodimenzionalno grafiko G, ki temelji na grafičnem paketu Java 2D. Grafična kontrola je sestavljena iz dveh grafičnih scen. Prva (*GWindow*) vsebuje celotno ozadje, torej obe osi (abscisna, ordinatna), naslov, ozadje grafa, mrežne črte itd. Ta scena pokriva celotno okno grafične kontrole. Druga scena (*GScene*) je v ospredju in vsebuje konkretne krivulje grafa. Ta scena vsebuje vidno polje, ki se ujema z poljem grafa, to pa nam omogoča, da lahko vrednosti točk, ki jih podamo z zaporedjem $T = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, neposredno rišemo. Grafična kontrola omogoča povečevanje in zmanjševanje grafa v poljubni točki.

V naslovni vrstici okna osnovne meritve (slika 3.19) se nahajajo vrednosti točke x, y in z, ki jo s trenutnim grafom analiziramo (za točke iz naše LiDARske naprave so to podatki o geografski dolžini, širini in višini). Na ordinatni



Slika 3.19: Okno grafičnega prikaza posamezne točke.

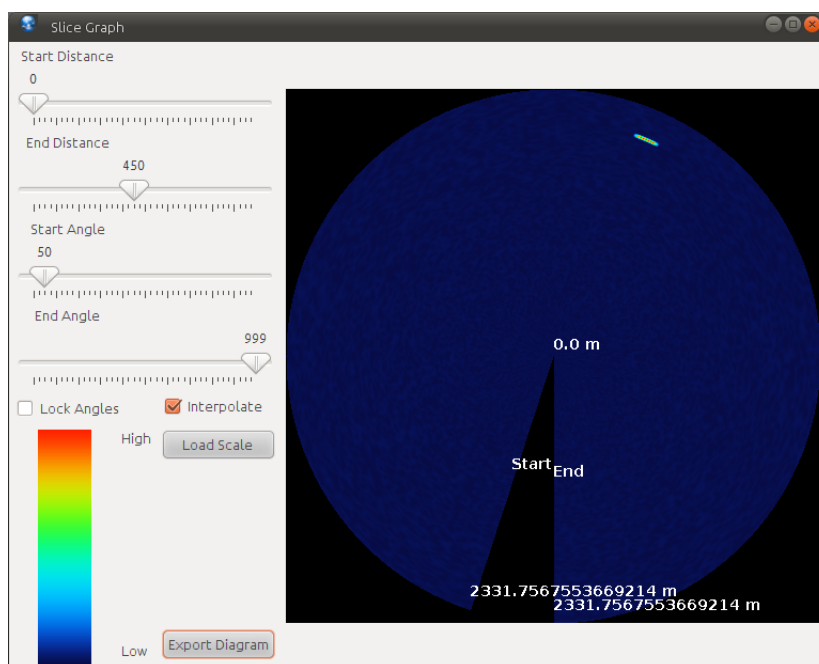
osi se nahaja intenziteta signala, na abscisi pa razdalja od izvora laserskih sunkov. Intenziteta signala je preračunana na interval $[0,1]$. Zaradi procesiranja podatkov v senzorju smo včasih priča tudi rahlo negativnim vrednostim. Z uporabo računalniške miške se lahko graf poljubno pomanjša ali poveča, omenjene akcije so podrobneje predstavljene v tabeli 3.2.

Akcija	Transformacija
levi miškin gumb	povečava (skaliranje angl. <i>zoom in</i>)
desni miškin gumb	zmanjšanje (skaliranje angl. <i>zoom out</i>)

Tabela 3.2: Uporaba miške pri grafu intenzitete.

3.5.3.2 Graf v rezini

Z grafom v rezini (prikazan na sliki 3.20) lahko prikažemo točke, ki so bile pridobljene z našim LiDARskim sistemom, s pomočjo programa za meritev v rezini (predstavljen v poglavju 3.5.2.2).



Slika 3.20: Okno za prikaz točk v rezini.

Diagram rezine v sredini okna je realiziran s kontrolo LidarScan, ki je obstajala že v predhodni verziji programske opreme za delo z LiDARsko napravo. Diagram rezine je sestavljen iz več osnovnih meritev, torej skupka točk, zajetega z enim samim LiDARskim sunkom. Meritve s poljubno odmično gostoto, ki sestavljajo diagram rezine, si sledijo v zaporedju glede na odmični kot. Prva meritev rezine je označena na robu diagrama z oznako *start*, zadnja meritev pa z oznako *end*. Vmes so enakomerno prikazane vmesne meritve. Na levi strani grafičnega okna, se nahajajo drsniki, ki omejujejo kote prikaza. Z drsniki *Start distance* in *End Distance* omejujemo prikaz meritev na določeno vzdolžno območje, katerega vrednosti so izpisane na robovih in na sredini diagrama. Z drsnikoma *Start Angle* in *End Angle* pa izbiramo začetni in končni kot prikaza rezine. Potrditveno polje *Lock angles* je namenjeno zaklepanju kotov, in sicer poskrbi za to, da sta kota (in posledično diagram) simetrična. Možnost *Interpolate* poskrbi za bolj gladek prikaz diagrama, tako da linearno

interpolira (tako radialno kot angularno) vrednosti med posameznimi meritvami. Pod drsniki se nahaja barvna intenzitetna skala, s pomočjo katere se izriše diagram rezine. Zgornji del barvne skale predstavlja vrednost točke z največjo intenziteto, spodnji del pa točko z najmanjšo vrednostjo. Barvno skalo je možno tudi zamenjati, ob pritisku na gumb *Load Scale* se prikaže pogovorno okno, s pomočjo katerega odpremo bitno sliko, ki bo predstavljala novo barvno skalo. Ob zamenjavi barvne skale se diagram rezine takoj obarva z novo barvno skalo. Z gumbom *Export Digram* lahko diagram v rezini izvozimo v slikovno datoteko. Ob pritisku na omenjen gumb, se prikaže pogovorno okno, s pomočjo katerega določimo slikovni datoteki ime in mesto, kjer jo bomo hranili.

3.5.3.3 3D prikaz točk



Slika 3.21: Okno za 3D prikaz točk. Okno prikazuje točke iz datoteke formata LAS, katera ni bila ustvarjena z našo aplikacijo.

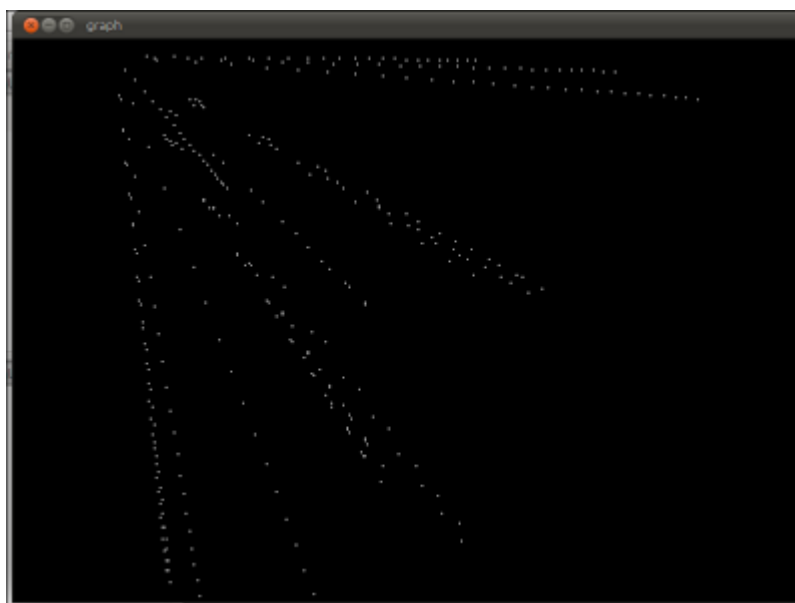
Grafična kontrola za prikaz vseh točk iz prebrane datoteke formata LAS, je napisana s pomočjo programskega vmesnika (angl. *Application Programming Interface - API*) Java 3D, ki za svoje delovanje potrebuje eno izmed grafičnih knjižnic OpenGL ali DirectX.

Animacije oz. grafične scene so strukturirane v obliki drevesa. Na sliki 3.23 je prikazana drevesna struktura grafične scene, ki je v naši aplikaciji uporabljena za prikaz tridimenzionalnih točk iz datoteke LAS. Izhodišče grafične scene je v *Virtual universe*. Vsaka scena v Java 3D je vsebovana v osnovni vejivni grupi *BranchGroup*, ki dopušča spreminjanje vsebine grupe. Na osnovno

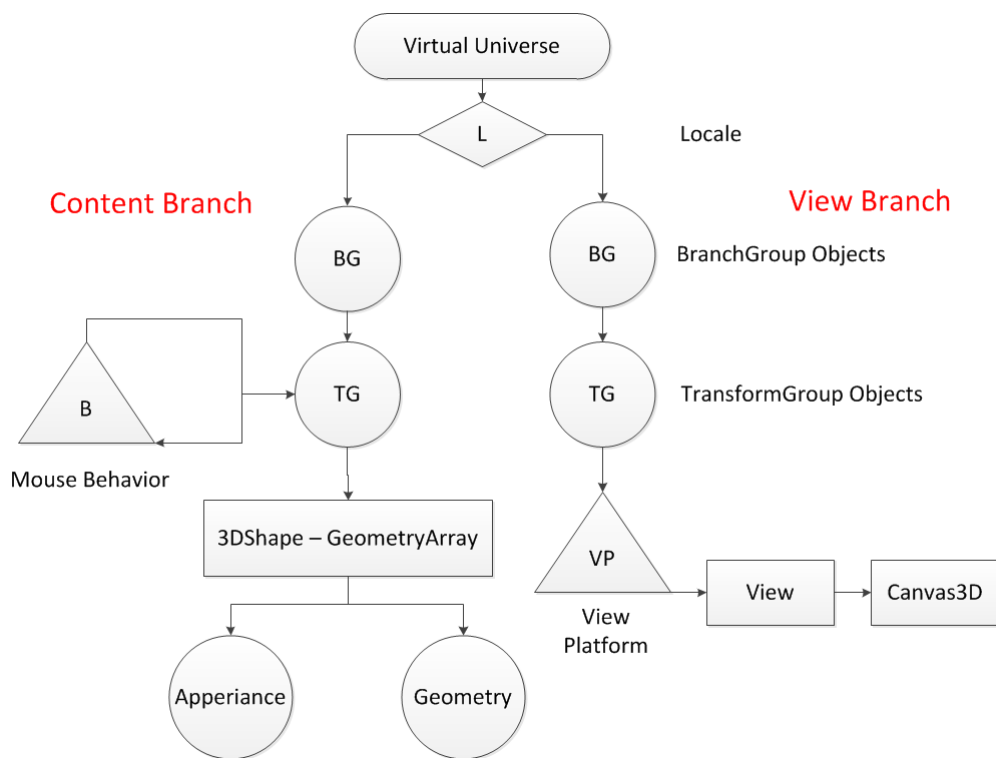
vejo smo vezali transformacijsko grupo, ki omogoča osnovne geometrične transformacije: rotacija, translacija in skaliranje. Pri gradnji scene smo nato na transformacijsko grupo vezali komponento Shape3D, ki predstavlja tabelo točk (*PointArray*) z vsemi prebranimi točkami iz datoteke formata LAS. Ker komponenta Shape3D ne dopušča osnovnih geometrijskih transformacij, smo uporabili vmesnik *Behavior*, s katerim smo omogočili transformacije z uporabo vhodnih naprav (npr. miška, tipkovnica, igralna palica). V našem primeru smo nanj vezali tri funkcije za translacijo, rotacijo in skaliranje z uporabo računalniške miške (tabela 3.3), s čimer smo zelo poenostavili ogled objektov v prostoru.

Akcija	Transformacija
levi miškin gumb	rotacija
desni miškin gumb	translacija
tipka Alt + levi miškin gumb	skaliranje (<i>zoom</i>)

Tabela 3.3: Uporaba miške pri tridimenzionalnem grafu.



Slika 3.22: Okno za 3D prikaz točk. Okno prikazuje točke iz datoteke formata LAS, ki je bila ustvarjena s programom meritve v polju (*Field Measurement*).



Slika 3.23: Drevesna struktura naše grafične scene.

Poglavje 4

Sklepne ugotovitve

Pri pisanju diplomske naloge, sem se srečal z veliko zanimivimi področij računalništva in informatike. Naj naštejemo le nekatere: uporaba različnih protokolov pri spletni storitvi, XML jezik, načrtovanje pri izvedbi projekta, razvoj protokola za spletni servis in merilni kanal, analiza standardov za uporabo datotek formata LAS, razvoj vmesnika (*API*) za branje ter pisanje datotek v formatu LAS, 2D in 3D grafika pri razvoju grafičnih kontrol za prikazovanje izvedenih meritev, matematika in algebra pri razvoju kontrol ter določanju geografskega položaja izmerjenih točk, uporaba geodetskih algoritmov za določanje točk v realnem koordinatnem sistem (določanje geografskega položaja izmerjenih točk)...

Moja glavna naloga je bila implementacija odjemalca za LiDAR kot spletno storitev in s to razvito aplikacijo lahko na oddaljen način upravljamo z LiDARskim strežnikom in preko njega tudi z LiDARsko napravo. Povezavo med odjemalcem in LiDARskim strežnikom ureja samostojno razvit protokol, ki omogoča naročanje storitve meritev (s pomočjo LiDARskega programa) ob določenem času na določenem kraju in zajem podatkov z naprave ob izvajanju programa. Implementirana aplikacija omogoča uporabo treh vrst meritev (meritev v pravokotnem področju, meritev v rezini in hitra meritev) v treh korakih. Prvi korak je naročanje storitve meritve, ki jo ureja SOAP protokol, drugi korak pa zajem podatkov iz LiDARske naprave ob izvajanju programa meritve. Pred shranjevanjem prejetih rezultatov meritev aplikacija s pomočjo Vincentyjeve formule za posamezno točko izračuna koordinate v realnem geografskem sistemu, šele potem se rezultat meritve shrani v standardiziranem LAS formatu na poljubni lokaciji v lokalnem pomnilniku. Sledi tretji korak v katerem je možno podatke iz prebrane datoteke formata LAS, ki predstavljajo rezultat izvedenih programov meritev, nazorno prikazati in podrobneje anali-

zirati. Aplikacija poleg sestavljanja in pošiljanja programov meritev omogoča njihovo shranjevanje in urejanje.

Pri delu s samo LiDARsko napravo smo med izvajanjem projekta naleteli na več, sicer kasneje rešenih težav, ki pa so nam vzele veliko časa. Projekt se je časovno zavlekel predvsem zaradi nadgradnje gibalnega modula LiDARske naprave, testiranja v tem času namreč niso bila možna. Vendar je bila nadgradnja nujna, saj smo le na ta način lahko odpravili pomanjkljivost gibalnega modula. Prejšnji gibalni modul je omogočal le majhne in počasne premike, zaradi česar je bilo delo težavno.

4.1 Možnosti za nadaljnji razvoj in delo

Kot možnosti za nadaljnji razvoj oz. dodelave in izboljšave v trenutni aplikaciji, bi predlagal naslednje:

- Razviti bi bilo potrebno protokol za interaktivni nadzor nad LiDARsko napravo.
- Izboljšati način prikaza izmerjenih točk, tako da bi neposredno na zemljevid risali izmerjene točke in njihovo intenziteto. To bi bilo možno doseči z uporabo Nasinega zemljevida *World Wind*, saj ponuja bogat vmesnik za programski jezik Java.
- Lahko bi razširili uporabo grafa v rezini, in sicer tako, da bi v tridimenzionalnem pogledu množice točk izbrali poljubno rezino (neko funkcijo). Glede na njo bi vse meritve, ki pripadajo tej funkciji izrisali v grafu v rezini.
- V trenutno aplikacijo bi lahko vključili orodje za brezizgubno stiskanje podatkov LiDAR (LAS datotek) [40].

Na področju strojne opreme LiDARske naprave bi predlagal naslednje izboljšave:

- Da bi dosegli natančnost izvedenih meritev, je potrebno LiDARsko napravo pred začetkom izvajanja meritve postaviti v vodoraven položaj in usmeriti proti severu. Zato bi bilo potrebno na sprejemnik (teleskop) LiDARske naprave namestiti GPS napravo, s katero bi v primeru, da je LiDARska naprava postavljena na hribu dosegli vodoraven položaj in njeno usmerjenost proti severu. Podatki iz GPS naprave bi se pošiljali po merilnem kanalu oddaljenemu uporabniku, z namenom izračuna realnih geografskih koordinat.

- Da bi bila LiDARska naprava povsem mobilna, je potrebno razmišljati o nakupu električnega generatorja in mobilne telekomunikacijske naprave za pošiljanje izmerjenih točk oddaljenemu uporabniku (odjemalcu).

Dodatek A

Priloge

A.1 Tabele formatov zapisov točk v datoteki LAS

Element/polje	Opis	Tip	Velikost	Obvezen
X	Koordinata x.	long	4 zlogi	*
Y	Koordinata y.	long	4 zlogi	*
Z	Koordinata z.	long	4 zlogi	*
Intensity	Intenziteta oz. intenzivnost vrnjene- nega žarka.	unsigned short	2 zloga	
Return Number	Kateri odboj za posamezen žarek.	3 bits	3 biti	*
Number of Returns	Skupno število odbojev žarka.	3 bits	3 biti	*
Scan Direction Flag	Smer, v katero je potovalo zrcalo laserja ob pošiljanju žarka.	1 bit	1 bit	*
Edge of Flight Line	Označuje zadnjo točko pri odčitavanju površja.	1 bit	1 bit	*
Classification	Klasifikacija posamezne točke (tla, nizka podrast, srednja podrast, ...).	unsigned char	1 zlog	*
Scan Angle Rank	Kot, pri katerem je bila točka za- jeta.	char	1 zlog	*
User Data	Poljubni podatki.	unsigned char	1 zlog	
Point Source ID	ID izvora/datoteke, iz katere je ta točka.	unsigned short	2 zloga	*

Tabela A.1: Format zapisa točk 0

Element/polje	Opis	Tip	Velikost	Obvezen
X	Koordinata x.	long	4 zlogi	*
Y	Koordinata y.	long	4 zlogi	*
Z	Koordinata z.	long	4 zlogi	*
Intensity	Intenziteta oz. intenzivnost vrjnega žarka.	unsigned short	2 zloga	
Return Number	Kateri odboj za posamezen žarek.	3 bits	3 biti	*
Number of Returns	Skupno število odbojev žarka.	3 bits	3 biti	*
Scan Direction Flag	Smer, v katero je potovalo zrcalo laserja ob pošiljanju žarka.	1 bit	1 bit	*
Edge of Flight Line	Označuje zadnjo točko pri prebiranju površja.	1 bit	1 bit	*
Classification	Klasifikacija posamezne točke (tla, nizka podrast, srednja podrast, ...).	unsigned char	1 zlog	*
Scan Angle Rank	Kot, pri katerem je bila točka zajeta.	char	1 zlog	*
User Data	Poljubni podatki.	unsigned char	1 zlog	
Point Source ID	ID izvora/datoteke, iz katere je ta točka.	unsigned short	2 zloga	*
GPS Time	Čas GPS, ko je bila točka zajeta.	double	8 zlogov	*

Tabela A.2: Format zapisa točk 1

Element/polje	Opis	Tip	Velikost	Obvezen
X	Koordinata x.	long	4 zlogi	*
Y	Koordinata y.	long	4 zlogi	*
Z	Koordinata z.	long	4 zlogi	*
Intensity	Intenziteta oz. intenzivnost vrnjene- nega žarka.	unsigned short	2 zloga	
Return Number	Kateri odboj za posamezen žarek.	3 bits	3 biti	*
Number of Returns	Skupno število odbojev žarka.	3 bits	3 biti	*
Scan Direction Flag	Smer, v katero je potovalo zrcalo laserja ob pošiljanju žarka.	1 bit	1 bit	*
Edge of Flight Line	Označuje zadnjo točko pri odčitavanju površja.	1 bit	1 bit	*
Classification	Klasifikacija posamezne točke (tla, nizka podrast, srednja podrast, ...).	unsigned char	1 zlog	*
Scan Angle Rank	Kot, pri katerem je bila točka za- jeta.	char	1 zlog	*
User Data	Poljubni podatki.	unsigned char	1 zlog	
Point Source ID	ID izvora/datoteke, iz katere je ta točka.	unsigned short	2 zloga	*
Red	Rdeča komponenta v sistemu RGB za točko.	unsigned short	2 zloga	*
Green	Zelena komponenta v sistemu RGB za točko.	unsigned short	2 zloga	*
Blue	Modra komponenta v sistemu RGB za točko.	unsigned short	2 zloga	*

Tabela A.3: Format zapisa točk 2

Element/polje	Opis	Tip	Velikost	Obvezen
X	Koordinata x.	long	4 zlogi	*
Y	Koordinata y.	long	4 zlogi	*
Z	Koordinata z.	long	4 zlogi	*
Intensity	Intenziteta oz. intenzivnost vrnjene- nega žarka.	unsigned short	2 zloga	
Return Number	Kateri odboj za posamezen žarek.	3 bits	3 biti	*
Number of Returns	Skupno število odbojev žarka.	3 bits	3 biti	*
Scan Direction Flag	Smer, v katero je potovalo zrcalo laserja ob pošiljanju žarka.	1 bit	1 bit	*
Edge of Flight Line	Označuje zadnjo točko pri prebi- ranju površja.	1 bit	1 bit	*
Classification	Klasifikacija posamezne točke (tla, nizka podrast, srednja podrast, ...).	unsigned char	1 zloga	*
Scan Angle Rank	Kot, pri katerem je bila točka za- jeta.	char	1 zlog	*
User Data	Poljubni podatki.	unsigned char	1 zlog	
Point Source ID	ID izvora/datoteke, iz katere je ta točka.	unsigned short	2 zloga	*
GPS Time	Čas GPS, ko je bila točka zajeta.	double	8 zlogov	*
Red	Rdeča komponenta v sistemu RGB za točko.	unsigned short	2 zloga	*
Green	Zelena komponenta v sistemu RGB za točko.	unsigned short	2 zloga	*
Blue	Modra komponenta v sistemu RGB za točko.	unsigned short	2 zloga	*

Tabela A.4: Format zapisa točk 3

Element/polje	Opis	Tip	Velikost	Obvezen
X	Koordinata x.	long	4 zlogi	*
Y	Koordinata y.	long	4 zlogi	*
Z	Koordinata z.	long	4 zlogi	*
Intensity	Intenziteta oz. intenzivnost vrnjene- nega žarka.	unsigned short	2 zloga	
Return Number	Kateri odboj za posamezen žarek.	3 bits	3 biti	*
Number of Returns	Skupno število odbojev žarka.	3 bits	3 biti	*
Scan Direction Flag	Smer, v katero je potovalo zrcalo laserja ob pošiljanju žarka.	1 bit	1 bit	*
Edge of Flight Line	Označuje zadnjo točko pri odčitavanju površja.	1 bit	1 bit	*
Classification	Klasifikacija posamezne točke (tla, nizka podrast, srednja podrast, ...).	unsigned char	1 zlog	*
Scan Angle Rank	Kot, pri katerem je bila točka za- jeta.	char	1 zlog	*
User Data	Poljubni podatki.	unsigned char	1 zlog	
Point Source ID	ID izvora/datoteke, iz katere je ta točka.	unsigned short	2 zloga	*
GPS Time	Čas GPS, ko je bila točka zajeta.	double	8 zlogov	*
Wave Packet Descrip- tor Index	Vrednost pokaže na uporabniško določen zapis, kateri opisuje pa- ket valovne oblike povezan s to točko.	unsigned char	1 zlog	*
Byte offset to wave- form data	Lokacija od začetka Paketov va- lovne oblike do glave paketa va- lovne oblike kateri je povezan s trenutno točko	unsigned long long	8 bytes	*
Waveform packet size in bytes	Velikost paketa valovne oblike ka- teri je povezan s trenutno točko	unsigned long	4 zlogi	*
Return Point Wave- form Location	Odmik v pikosekundah (10 na- 12) od prve digitalizirane vredno- sti znotraj paketa valovne oblike, katera je povezana s sunkom za- jemanja točke.	float	4 zlogi	*
X(t)	Parametri X(t), Y(t) in Z(t) opredeljujejo parametrično linijo enačbe za ekstrapolacijo točke vzdolž valovne oblike.	float	4 zlogi	*
Y(t)		float	4 zlogi	*
Z(t)		float	4 zlogi	*

Tabela A.5: Format zapisa točk 4

Element/polje	Opis	Tip	Velikost	Obvezen
X	Koordinata x.	long	4 zlogi	*
Y	Koordinata y.	long	4 zlogi	*
Z	Koordinata z.	long	4 zlogi	*
Intensity	Intenziteta oz. intenzivnost vrnjene- nega žarka.	unsigned short	2 zloga	
Return Number	Kateri odboj za posamezen žarek.	3 bits	3 biti	*
Number of Returns	Skupno število odbojev žarka.	3 bits	3 biti	*
Scan Direction Flag	Smer, v katero je potovalo zrcalo laserja ob pošiljanju žarka.	1 bit	1 bit	*
Edge of Flight Line	Označuje zadnjo točko pri odčitavanju površja.	1 bit	1 bit	*
Classification	Klasifikacija posamezne točke (tla, nizka podrast, srednja podrast, ...).	unsigned char	1 zlog	*
Scan Angle Rank	Kot, pri katerem je bila točka za- jeta.	char	1 zlog	*
User Data	Poljubni podatki.	unsigned char	1 zlog	
Point Source ID	ID izvora/datoteke, iz katere je ta točka.	unsigned short	2 zloga	*
GPS Time	Čas GPS, ko je bila točka zajeta.	double	8 zlogov	*
Red	Rdeča komponenta v sistemu RGB za točko.	unsigned short	2 zloga	*
Green	Zelena komponenta v sistemu RGB za točko.	unsigned short	2 zloga	*
Blue	Modra komponenta v sistemu RGB za točko.	unsigned short	2 zloga	*
Wave Packet Descrip- tor Index	Vrednost pokaže na uporabniško določen zapis, kateri opisuje pa- ket valovne oblike povezan s to točko.	unsigned char	1 zlog	*
Byte offset to wave- form data	Lokacija od začetka Paketov va- lovne oblike do glave paketa va- lovne oblike kateri je povezan s trenutno točko	unsigned long long	8 bytes	*
Waveform packet size in bytes	Velikost paketa valovne oblike ka- teri je povezan s trenutno točko	unsigned long	4 zlogi	*
Return Point Wave- form Location	Odmik v pikosekundah (10 na- 12) od prve digitalizirane vredno- sti znotraj paketa valovne oblike, katera je povezana s sunkom za- jemanja točke.	float	4 zlogi	*
X(t)	Parametri X(t), Y(t) in Z(t) opredeljujejo parametrično linijo enačbe za ekstrapolacijo točke vzdolž valovne oblike.	float	4 zlogi	*
Y(t)		float	4 zlogi	*
Z(t)		float	4 zlogi	*

Tabela A.6: Format zapisa točk 5

A.2 Tabela glave datoteke formata LAS

Element/polje	Opis	Tip	Velikost	Obvezen
File Signature	4 znaki »LASF«. Za hiter pregled formata datoteke.	char[4]	4 zlogi	*
File Source ID	ID številka letalske linije med 0 in 65535	unsigned short	2 zloga	*
Global encoding	Bitne vrednosti določajo nekatere podatke o datoteki.	unsigned short	2 zloga	*
Project ID – GUID data 1	ID projekta.	unsigned long	4 zlogi	
Project ID – GUID data 2	ID projekta.	unsigned short	2 zloga	
Project ID – GUID data 3	ID projekta.	unsigned short	2 zloga	
Project ID – GUID data 4	ID projekta.	unsigned char[8]	8 zlogov	
Version Major	Verzija datoteke. V kombinaciji s poljem »Version Minor«.	unsigned char	1 zlog	*
Version Minor	Verzija datoteke. V kombinaciji s poljem »Version Major«.	unsigned char	1 zlog	*
System Identifier	Identifikator, ki nam pove, na kakšen način so podatki nastali.	char[32]	32 zlogov	*
Generating Software	Informacija o programski opremi, ki je bila uporabljena pri generiranju.	char[32]	32 zlogov	*
File Creation Day of Year	Dan v letu, ko je bila datoteka kreirana.	unsigned short	2 zloga	
File Creation Year	Leto kreiranja, izraženo s štirimestnim številom.	unsigned short	2 zloga	
Header Size	Velikost glave datoteke v zlogih.	unsigned short	2 zloga	*
Offset to point data	Odmik v zlogih do prvega zapisa točke.	unsigned long	4 zlogi	*
Number of variable length records	Število zapisov s spremenljivo dolžino.	unsigned long	4 zlogi	*
Point data format ID	Format zapisa točk. Format 1.2 podpira formate 0, 1, 2 in 3. Zadnji 1.3 pa še 4 in 5.	unsigned char	1 zlog	*
Point data record length	Dolžina (v zlogih) zapisa ene točke.	unsigned short	2 zloga	*
Number of point records	Skupno število točk v datoteki.	unsigned long	4 zlogi	*
Number of points by return	Število odbojev, ki se upoštevajo pri odboju laserskega žarka.	unsigned long[5]	20 zlogov	*
X scale factor	Faktor normalizacije za komponento x.	double	8 zlogov	*
Y scale factor	Faktor normalizacije za komponento y.	double	8 zlogov	*
Z scale factor	Faktor normalizacije za komponento z.	double	8 zlogov	*
X offset	Odmik x.	double	8 zlogov	*
Y offset	Odmik y.	double	8 zlogov	*
Z offset	Odmik z.	double	8 zlogov	*
Max X	Maksimalna vrednost komponente x.	double	8 zlogov	*
Min X	Minimalna vrednost komponente x.	double	8 zlogov	*
Max Y	Maksimalna vrednost komponente y.	double	8 zlogov	*
Min Y	Minimalna vrednost komponente y.	double	8 zlogov	*
Max Z	Maksimalna vrednost komponente z.	double	8 zlogov	*
Min Z	Minimalna vrednost komponente z.	double	8 zlogov	*
Start of Waveform Data Packet Record	Odmik v zlogih od začetka LAS datoteke pa do prvega zloga paketa podatkov valovne oblike.	unsigned long long	8 zlogov	*

Tabela A.7: Glava LAS datoteke.

A.3 Opis elementov shranjenega programa meritve v formatu XML

Element	Opis	Polje v LAS
programType	Vrsta programa meritve. Možne vrednosti: »quick measurement«, »field measurement«in »slice measurement«	Generating Software
name	Ime programa meritve.	
programID	Vrednost za identifikacijo programa meritve.	
energy	Intenziteta laserskega sunka (vsebuje vrednost, ki je med 1 in 20).	
pulsesPMsm	Število laserskih sunkov za posamezno meritev (<i>Pulses / Msm</i>).	
measuringType	Tip meritve, ki lahko vsebuje vrednost »Data«, kar pomeni, da bomo zajemali samo podatke, ali vrednost »Background«, kar pomeni, da bomo zajemali samo ozadje, ali vrednost »Data-Background«, kar pomeni razlika podatkov in ozadja.	
latitude	Zemljepisna širina, željenega položaja LiDARske naprave.	
longitude	Zemljepisna dolžina, željenega položaja LiDARske naprave.	
yawStart	Začetni odmik področja meritve.	
tiltStart	Začetni nagib področja meritve.	
yawEnd	Končni odmik področja meritve.	
tiltEnd	Končni nagib področja meritve.	
yawResolution	Odmična gostota (število meritev v smeri odklona).	
tiltResolution	Nagibna gostota (število meritev v smeri nagiba).	
repetition	Število ponovitev celotnega niza meritve.	
delay	Koliko sekund naj aplikacija čaka med morebitnimi ponovitvami.	
date	Željen datum meritve.	File Creation Day of Year, File Creation Year
time	Željen čas meritve.	

Tabela A.8: Elementi programa meritve.

Slike

1.1	Poglavitni deli LiDARskega sistema.	8
1.2	Princip delovanja LiDARja.	9
2.1	Struktura sporočila SOAP.	13
2.2	Primer zahteve v obliki sporočila SOAP.	14
2.3	Primer odgovora spletne storitve v obliki sporočila SOAP.	15
2.4	Glavni elementi WSDL storitve.	16
2.5	Primer XML dokumenta.	18
2.6	Primer XML dokumenta z atributi.	18
2.7	Struktura formata LAS.	22
2.8	Povezava geoid in elipsoid [25].	23
2.9	Rotacijski elipsoid.	24
2.10	Geodetski referenčni elipsoid [25].	24
2.11	Geoid, gravitacijske anomalije	25
2.12	Pravokotni koordinatni sistem (razdelitev na kvadrante) [30].	27
2.13	Primer polarnega koordinatnega sistema.	27
2.14	kroglne koordinate	28
2.15	Prikaz prečne cilindrične projekcije [27].	30
2.16	Prikaz meridianskih con Gauss-Kruegerjeve projekcije.	30
2.17	Povezava med globalnim koordinatnim sistemom in LiDARskim lokalnim koordinatni sistemom.	35
2.18	LiDARski interni koordinatni sistem.	35
2.19	Omejitev delovnega področja LiDARske naprave v smeri odklona.	36
2.20	Omejitev delovnega področja LiDARske naprave v smeri nagiba.	36
3.1	Okence za informacije o aplikaciji <i>RhoLaaS</i>	37
3.2	Struktura programa meritve.	39
3.3	Diagram arhitekturne sheme LiDAR sistema.	41
3.4	Oblika prenosa po merilnem kanalu.	43
3.5	Zaporedje splošnih informacij izvedene meritve.	44

3.6	Oblika meritve dobljena po merilnem kanalu.	44
3.7	Diagram protokola za komunikacijo z LiDARskim sistemom. . .	45
3.8	Osnovno okno programa <i>RhoLaaS</i>	47
3.9	Diagram grafičnega vmesnika osnovnega okna programa <i>RhoLaaS</i> .	50
3.10	Arhiv programov meritev.	51
3.11	Format shranjenega programa meritve.	52
3.12	Lastnosti glave datoteke LAS.	54
3.13	Podokno, lastnosti točk iz datoteke LAS.	55
3.14	Okno programa meritve v pravokotnem območju.	57
3.15	Okno programa za meritev v rezini.	59
3.16	Okno za hitro, osnovno meritev.	60
3.17	Privzeti spletni brskalnik na naslovu GoogleMaps.	62
3.18	Okno za prikaz koordinat geografsko širino in dolžino (latitude, longitude).	62
3.19	Okno grafičnega prikaza posamezne točke.	64
3.20	Okno za prikaz točk v rezini.	65
3.21	Okno za 3D prikaz točk. Okno prikazuje točke iz datoteke for- mata LAS, katera ni bila ustvarjena z našo aplikacijo.	66
3.22	Okno za 3D prikaz točk. Okno prikazuje točke iz datoteke for- mata LAS, ki je bila ustvarjena s programom meritve v polju (<i>Field Measurement</i>).	67
3.23	Drevesna struktura naše grafične scene.	68

Tabele

1.1	Karakteristike LiDARskega sistema.	10
2.1	Opis glavnih elementov znotraj WDSL	16
2.2	Struktura LAS datoteke	20
2.3	Struktura LAS datoteke z podatki valovne oblike	20
2.4	Zapisi spremenljive dolžine.	21
2.5	Dimenzije elipsoidov Bessel, WGS84 in GRS80 [29, 27].	25
2.6	Zapisi Vincenty-jeve neposredne metode.	34
3.1	Tabela konstant za vrsto podatka.	44
3.2	Uporaba miške pri grafu intenzitete.	64
3.3	Uporaba miške pri tridimenzionalnem grafu.	67
A.1	Format zapisa točk 0	72
A.2	Format zapisa točk 1	73
A.3	Format zapisa točk 2	74
A.4	Format zapisa točk 3	75
A.5	Format zapisa točk 4	76
A.6	Format zapisa točk 5	77
A.7	Glava LAS datoteke.	78
A.8	Elementi programa meritve.	79

Literatura

- [1] A. Kranjec, A. Klemenc, and A. Keuc, “Fokus, društvo za sonaraven razvoj, spreminjam navade, ne pa podnebja! podnebne spremembe: priročnik”, Dostopno na: <http://www.focus.si/files/Publikacije/prirocnikCC.pdf>, maj 2005, Zajeto: 12. 9. 2011.
- [2] A. Shah, “Global dimming, global issues”, Dostopno na: <http://www.globalissues.org/article/529/global-dimming>, januar 2005, Zajeto: 1. 9. 2011.
- [3] B. Horvat, “Očem varni lidar”, Diplomsko delo, 2008, Univerza v Ljubljani, Fakulteta za matematiko in fiziko.
- [4] J. Bodlaj, “Prenosni, očem varni lidar z računalniškim krmiljenjem in podporo”, Diplomsko delo, 2008, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko.
- [5] D. Mongus (FERI, Uni Mb), T. Šukljan (PINT, UP), A. Brodnik (PINT, UP), A. Biasizzo (IJS, Lj), F. Novak (IJS, Lj), V. Podobnik (IGEA d.o.o., Lj), M. Kuder (XLAB d.o.o., Lj), B. Lipuš (FERI, Uni Mb), and B. Žalik (FERI, Uni Mb), “Obdelava velikih količin podatkov lidar”, Zbornik konference ERK.
- [6] G. Poberaj, I. Poberaj, and M. Zgonik, “Razvoj, očem varnega lidarja za nadzor onesnaženja ozračja”, Tehnično poročilo, 2003, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko.
- [7] A. Zrnec, “Spletne storitve niso dovolj”, *Elektrotehniški vestnik*, vol. 71, no. 5, november 2004.
- [8] T. Vidmar, *Informacijsko-komunikacijski sistem*, Pasadena, Ljubljana, 2002.

- [9] S. Short, *Building XML Web Services for the Microsoft .NET platform*, Microsoft Press, 2002.
- [10] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web services description language (wsdl) 1.1.", Tech. Rep., W3C, 2001.
- [11] E. Newcomer, *Understanding Web*, Addison-Wesley, maj 2003.
- [12] XML Core Working Group, "Extensible markup language (xml) 1.0", Tech. Rep., W3C, 2008.
- [13] W3C and WHATWG, "Html 4.01 specification", Tech. Rep., W3C, 1999.
- [14] DOM Working Group members, "Document object model (dom) level 2 core specification, version 1.0.", Tech. Rep., W3C, 2000.
- [15] R. Anderson, *Professional XML*, Birmingham, UK : Wrox Press, 2000.
- [16] "Asprs, the american society for photogrammetry & remote sensing", Dostopno na: <http://www.asprs.org/>, Zajeto: 12. 9. 2011.
- [17] A. Samberg, "An implementation of the asprs las standard", *IAPRS*, vol. 36, no. 3, 2007.
- [18] The American Society for Photogrammetry & Remote Sensing, Washington Rd, Princeton, ZDA, *LAS specification. Version 1.3.*, oktober 2010.
- [19] The American Society for Photogrammetry & Remote Sensing, Washington Rd, Princeton, ZDA, *LAS specification. Version 1.0.*, maj 2003.
- [20] The American Society for Photogrammetry & Remote Sensing, Washington Rd, Princeton, ZDA, *LAS specification. Version 1.1.*, maj 2005.
- [21] The American Society for Photogrammetry & Remote Sensing, Washington Rd, Princeton, ZDA, *LAS specification. Version 1.2.*, september 2008.
- [22] The American Society for Photogrammetry & Remote Sensing, Washington Rd, Princeton, ZDA, *LAS specification. Version 2.0.*, maj 207.
- [23] "Asprs standards committee, laser (las) file format exchange activities", Dostopno na: <http://www.asprs.org/>, Zajeto: 12. 9. 2011.
- [24] M. Juvančič, *Geodezija za gozdarje in krajinske arhitekte*, Ljubljana, Univerza v Ljubljani, Biotehniška fakulteta, Oddelek za gozdarstvo in obnovljive gozdne vire, 2000.

- [25] B. Kovačič, *Geodezija za gradbene inženirje*, Fakulteta za gradbeništvo Maribor, 2004.
- [26] M. Kuhar, *GEOFIZIKA, študijski pripomoček*, Fakulteta za gradbeništvo in geodezijo, Ljubljana, 2006.
- [27] M. Kuhar, "Geodezija, učno gradivo", Dostopno na: <http://www.fgg.uni-lj.si/~mkuhar/pouk/uvod/gradivo/>, 2008, Zajeto: 1. 9. 2011.
- [28] "Splung.com physics, gravitation", Dostopno na: <http://www.splung.com/>, Zajeto: 12. 9. 2011.
- [29] NIMA Geospatial Science Center, "Geodesy for layman", Tech. Rep., Defense Mapping Agency, WASHINGTON D C, ZDA, marec 1983.
- [30] A. Berdajs and M. Ulbl, *Inženirska geodezija*, Zavod IRC, Ljubljana, 2010.
- [31] B. Stopar, M. Kuhar, and B. Koler, "Novi koordinatni sistem v Sloveniji", 2007.
- [32] M. Brumec and J. Koleša, "Iz znanosti in stroke prilagoditev zemljiškokatstrskega prikaza pri izdelavi geodetskega načrta", *Geodetski vestnik*, vol. 55, no. 2, junij 2011.
- [33] K. Kozmus and B. Stopar, "Načini določanja položaja s satelitskimi tehnikami", *Geodetski vestnik*, vol. 47, no. 4, november 2003.
- [34] P. Vaniček, "An online tutorial in geodesy", *University of New Brunswick, Kanada: AcademicPress*, 2001.
- [35] R. Rapp, *Geometric Geodesy II*, Division of Geodesy and Spatial Science, Ohio State University, Columbus, ZDA, 1993.
- [36] T. Vincenty, *Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations*, Directorate of Overseas Surveys of the Ministry of Overseas Development Kingston Road. Tolworth, Surrey, april 1975.
- [37] C. M. Thomas and W. E. Featherstone, "Validation of vincenty's formulas for the geodesic using a new fourth-order extension of kivioja's formula", *Journal of Surveying Engineering*, vol. 131, no. 20, 2004.

- [38] “liblas”, Dostopno na: <http://liblas.org/>, Zajeto: 12. 9. 2011.
- [39] Google, “Google maps”, Dostopno na: <http://maps.google.com/>, Zajeto: 12. 9. 2011.
- [40] D. Mongus and B. Zalik, “Efficient method for lossless lidar data compression”, *International Journal of Remote Sensing*, vol. 32, no. 9, 2011.
- [41] C. F. F. Karney, “Geodesics on an ellipsoid of revolution.”, Tech. Rep., SRI International, Washington Rd, Princeton, ZDA, februar 2011.