

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Karl-Johan Pilve

Tööriist Lanelet2 täppiskaartide loomiseks

Bakalaureusetöö (9 EAP)

Juhendajad:

Tambet Matiisen, MSc

Edgar Sepp, MSc

Tartu 2021

Tööriist Lanelet2 täppiskaartide loomiseks

Lühikokkuvõte:

Tartu Ülikoolis tegeleb isejuhtivaid sõidukeid puudutava teadus- ja arendustööga isejuhtivate sõidukite labor. Kuna tänapäevased isejuhtivad autod vajavad sõitmiseks detailset täppiskaarti, siis loodi isejuhtivate sõidukite laboris töövahend selliste kaartide joonistamiseks ning kaardistati ka esmased testrajad Tartu ja Tallinna tänavatel. Selle vektorkaardi põhjal genereeritakse isejuhtiva sõiduki tarkvarale sobiv Autoware *vector map* formaadis kaart. Isejuhtivate sõidukite labor soovib testida Lanelet2 kaardiformaati, kuna sellel on mitmeid eeliseid Autoware *vector map* formaadi ees.

Käesoleva bakalaureusetöö raames loodi tööriist, mis teisendab isejuhtivate sõidukite labori vektorkaardi Lanelet2 formaadis täppiskaardiks. Töös antakse ülevaade täppiskaartide ja Lanelet2 formaadi olemusest. Seejärel kirjeldatakse loodud tööriista tehnilist teostust ja tuuakse välja arendustöö käigus tekkinud probleemid. Lõpetuseks tutvustatakse tööriistaga genereeritud kaardi testimist ja analüüsitakse võimalusi tööriista täiendamiseks.

Võtmesõnad:

Täppiskaardid, isejuhtivad sõidukid, Lanelet2

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Tool for Creating High Definition Maps in Lanelet2 format

Abstract:

At the University of Tartu, the Autonomous Driving Lab (ADL) conducts research and development in the field of self-driving vehicles. As modern self-driving vehicles need detailed high definition vector map for driving, the ADL has developed a tool for drawing such maps and mapped the first test tracks on the streets of Tartu and Tallinn. A high definition vector map in Autoware format, which is compatible with software for self-driving vehicles, is generated from ADL's vector map. ADL wishes to start testing with maps in Lanelet2 format because it has many advantages over Autoware vector map format.

This bachelor's thesis aimed to create a tool that would convert the existing ADL's vector map to Lanelet2 format. This thesis gives an overview of high definition maps and Lanelet2 map format. Then the technical implementation is explained, and an overview of problems during the development process is provided. Finally, the description of the testing process and suggestions on improving the tool are also given.

Keywords:

high-definition maps, autonomous driving, Lanelet2

CERCS: P170 (Computer science, numerical analysis, systems, control)

Sisukord

Sissejuhatus	5
1. Täppiskaardid	6
1.2 Täppiskaartide tutvustus	6
1.3 Semantilised kaardid	7
1.4 Lanelet2 kaardiformaat	8
1.4.2 Sõidulõigud	8
1.4.3 Lanelet2 primitiivid	9
1.5 Autoware	10
2. Tööriista ülevaade	11
2.1 Kasutuses olevad vektorkaardid	11
2.1.1 Isejuhtivate sõidukite labori vektorkaart	11
2.1.2 Autoware formaadi puudused	12
2.2 Tehniline lahendus	13
2.2.1 Paralleelsete joonte genereerimine	14
2.2.2 Sõidulõikude ühendamine	16
2.2.3 Valgusfoorid	18
2.2.4 OpenStreetMap formaat	19
2.3 Lahenduse loomisel ületatud takistused	20
3. Testimine ja edasine arendustöö	23
3.1 Valideerimine	24
3.2 Töökiirus	25
3.3 Edasine arendustöö	26
Kokkuvõte	28
Viidatud kirjandus	29
Lisad	31
I. Tööriista lähtekood	31
II. Litsents	31

Sissejuhatus

Isejuhtivatel sõidukitel on suur potentsiaal muuta tulevikus liiklemine meie teedel kiiremaks, efektiivsemaks ja ohutumaks. Teadus- ja arendustöö isejuhtivate sõidukitega käib üle kogu maailma, mille tõttu loodi Tartu Ülikooli juurde isejuhtivate sõidukite labor, et ka Eestis katsetada maailmas välja töötatud tehnoloogilisi lahendusi. Üks peamisi etappe isejuhtivate sõidukite arendusprotsessis on võimalikult täpsete kaartide loomine. Selliseid väga täpseid kaarte nimetatakse täppiskaartideks. Isejuhtivate sõidukite laboril on valminud töövahend vektorkaartide joonistamiseks, mille abil on kaardistatud testimisrajad nii Tartus kui ka Tallinnas. Labori vektorkaart teisendatakse isejuhtiva sõiduki tarkvarale sobivaks Autoware *vector map* formaadis täppiskaardiks.

Käesoleva bakalaureusetöö eesmärk on luua tööriist, mis teisendaks isejuhtivate sõidukite labori vektorkaardi Lanelet2 formaadis täppiskaardiks. Isejuhtivate sõidukite labor soovib hakata testimas Lanelet2 formaadis kaartidega, sest seni kasutatud Autoware *vector map* formaadil on mitmeid puudusi ja Lanelet2 on üks peamisi alternatiive. Samas soovib labor endiselt teha testsõite ka olemasoleva kaardiga. Samade teede käsitsi kaardistamine kahe erineva kaardiformaadiga oleks väga töömahukas, eriti kui arvestada isejuhtivate sõidukite labori plaani ära kaardistada kogu Tartu linn. Bakalaureusetöö raames plaanitakse luua tööriist, mis võimaldaks luua Lanelet2 formaadis kaart olemasoleva vektorkaardi põhjal. See tähendaks, et käsitsi peaks muutma ainult olemasolevat kaarti.

Töö on jagatud kolmeks peatükiks. Esimeses peatükis selgitatakse lähemalt täppiskaartide olemust ja Lanelet2 kaardiformaati. Teises peatükis kirjeldatakse isejuhtivate sõidukite labori olemasolevat kaarti, tuuakse välja selle puudused, tehakse ülevaade bakalaureusetöö raames valminud tööriista tehnilisest lahendusest ja tuuakse välja arendustöö käigus tekkinud probleeme. Kolmandas peatükis kirjeldatakse tööriistaga genereeritud Lanelet2 kaardi testimist valideerimist ja töökiirust ning arutletakse, kuidas tööriista tulevikus täiendada.

1. Täppiskaardid

Kuna antud bakalaureusetöö eesmärgiks on luua tööriist, mis teisendab Tartu Ülikooli isejuhtivate sõidukite labori vektorkaardi Lanelet2 formaadis täppiskaardiks, siis kõigepealt on vaja tutvustada täppiskaartide olemust. Selles peatükis kirjeldatakse täpsemalt ka Lanelet2 kaardiformaati ning sõidulõike, mis on Lanelet2 formaadis kaardi tähtsaimaks elemendiks. Viimaks tutvustatakse isejuhtiva sõiduki opereerimise tarkvara Autoware ja selle tarkvara poolt kasutatavat kaardiformaati.

1.2 Täppiskaartide tutvustus

Võimalikult täpsete kaartide loomine on üks tähtsamaid aspekte isejuhtivate sõidukite arendamisel. Tänapäevaste veebis leiduvate navigatsioonikaartide asukohatäpsus ei ole piisav ohutuks sõiduks isejuhtiva autoga. Asukohatäpsus näitab asukoha erinevust tegelikust [1]. Tavaliselt on navigatsioonikaartide asukohatäpsus umbes 1 meeter. Kuna Eestis on ühe sõiduraja laius ligikaudu 3 meetrit ja rattaradade laius võib olla isegi alla ühe meetri, siis on vajalik anda isejuhtivatele sõidukitele ette kaart, mille asukohatäpsust saaks mõõta sentimeetrites. Selliseid väga täpseid vektorkujul kaarte nimetatakse täppiskaartideks (ingl *high definition vector maps*). Vektorkujuks nimetatakse andmete organiseerimise vormi, kus andmed kirjeldatakse punktide, joonte ja pindadena ning nende kuju ja vormi määravate matemaatiliste funktsioonidega ehk vektorandmetena [2].

Kumar Chellapilla [3] toob oma artiklis välja neli printsiipi, mis defineerivad täppiskaardi. Esiteks peaks sõiduk saama kaardilt võimalikult palju andmeid ümbruse kohta, kuna sellisel juhul ei ole vaja neid andmeid koguda sõidu ajal. Näiteks teede, ristmike ja liiklusmärkide tuvastamine on täpsem, kui kasutada kaardil olevaid andmeid. Teiseks peaksid täppiskaardid suurendama turvalisust. Näiteks ei peaks olema kaardile märgitud mitte ainult sõiduraja kiiruspiirang, vaid ka optimaalne sõidukiirus sellel sõidurajal. Kolmandaks peaks saama täppiskaarti kasutada ainulaadse sensorina, mille ulatusel piire ei ole. Kuna tavaliselt on isejuhtivate sõidukite sensorite ulatus 100-200 meetrit, siis täppiskaart võimaldab sõidukil “näha” ka kaugemal asuvaid objekte. Neljandaks peaks täppiskaarti saama lugeda ja uuendada korraka mitmed isejuhtivad sõidukid. Nii saavad sõidukid reaajas omavahel suhelda saavutades sellega suurema turvalisuse ja efektiivsuse.

Kumar Chellapilla [3] artiklis on välja toodud ka kihid, millest täppiskaart koosneb. Ta väidab, et kaardi jagamine kihtideks võimaldab iga kihti iseseisvalt koostada, uuendada ja testida. Artikli autor toob välja, et täppiskaartide baaskihiks on tavaline teedevõrgu kaart ja

iga järgnev kiht lisab kaardile rohkem detaile. Ta mainib, et baaskihi peale tuleb veel neli kihti: geomeetiline kiht (ingl *geometric map layer*), semantiline kiht (ingl *semantic map layer*), kaardi eelteadmiste kiht (ingl *map priors layer*) ja reaalaaja kiht (ingl *real-time layer*). Geomeetiline kiht koosneb 3D ruumiandmetest, mida kogutakse lidari, GPS-i ja erinevate kaameratega. Saadud andmetest genereeritakse 3D punktipilv. Semantilisel kihil kujutatakse mitmeid 2D ja 3D objekte nagu sõiduradade ääred, ristmikud, ülekäigurajad, liiklusmärgid ja nii edasi. Lisaks on semantilise kihi objektidel ka hulga metaandmeid nagu näiteks kiiruse piirangud ja pöörangute keelud. Kaardi eelteadmiste kihile märgitakse dünaamiliste elementide andmed ja ka informatsioon inimeste sõiduharjumuste kohta. Näiteks on kihile märgitud soovituslikud kiiruse profiilid, erinevad peatumiskohad tee andmiseks, mis ei ole stoppjoontega tähistatud, ja tulede järjestus, mida fooritulede tsükkel läbib. Neljas ja viimane kiht on reaalaaja kiht. See on ainuke kiht, mida saab muuta sõidu ajal. Kihile märgitakse liiklusummikud, uued teetööde piirkonnad ja muud takistused.

1.3 Semantilised kaardid

Nagu eelnevalt mainitud, märgitakse täppiskaardi semantilisele kihile sõidurajad, ülekäigurajad, liiklusmärgid ja kõik muu, mis aitab isejuhtival sõidukil teedel liigelda vastavalt liikluseeskirjadele. Kris Efland ja Holger Rapp [4] tutvustavad oma artiklis lähemalt semantilise kihi olemust ja selles lõigus lähtume nende käsitlusest. Semantilise kaardi tähtsaim element on teedevõrk, kuhu on märgitud sõiduradade arv ja nende suund. Lisaks märgitakse teedevõrgu kihile ka kohad, kus sõiduk peab jalakäijatele või teistele sõidukitele teed andma. Teedevõrgu kihist järgmine kiht on sõiduraja geomeetria kiht, mille täpsus peaks olema sentimeetrites ja kuhu märgitakse andmed teemärgistuse ja muude liikluskorraldusvahendite kohta, nagu teekünnised, teemärgistuse värv ja piirkonnad, kus on lubatud sõidurada vahetada. Kirjeldatud kahe kihi peamine erinevus seisnebki selles, et teedevõrgu kiht aitab planeerida pikemaids marsuute, näiteks kuidas sõita ühest linnast teise, kuid sõiduradade geomeetria kiht aitab teha detailsemaid otsuseid sõidutrajektoori kohta, nagu sõiduraja valik pööramisel.

Kris Efland ja Holger Rapp [4] toovad välja ka semantilise kaardi loomisprotsessi. Kõigepealt võetakse tavalise GPS navigeerimisel kasutatava kaardi andmed. Need andmed moodustavad teedevõrgu kaardikihi aluse, millele lisatakse andmeid, mis saadakse tänavaid ja teid isejuhtivate sõidukitega läbi sõites. Läbisõidu käigus korjatakse ka esmased andmed sõiduradade geomeetria kihile, kasutades erinevaid sensoreid ja masinõpet. Seejärel andmeid

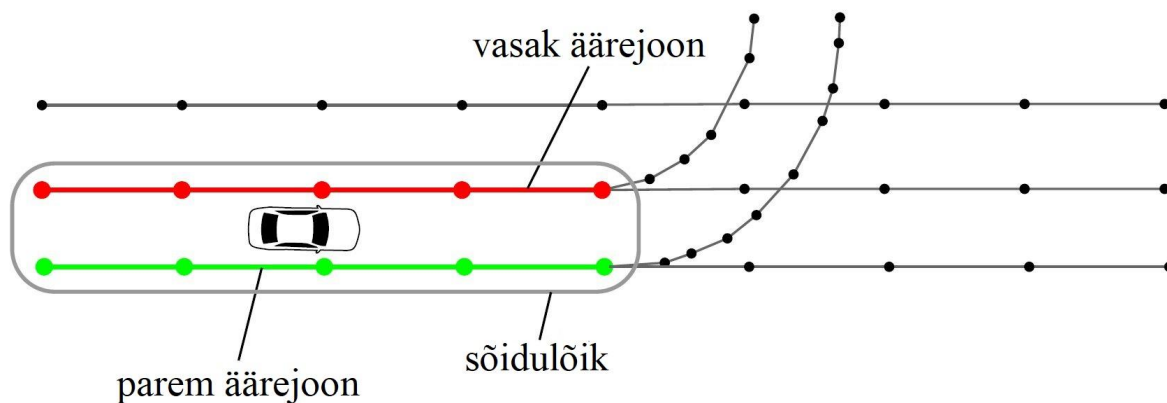
analüüsitakse ja need pannakse kaardile isejuhtivale sõidukile arusaadavalt. Viimaks kontrollitakse kaartide kvaliteeti, et kaardid oleksid sentimeetrise täpsusega. Kaarte kontrollivad nii inimesed kui ka erinevad veatuvastusalgoritmid. Enne isejuhtiva sõidukiga tänaval sõitmist testitakse kaarte tavaliselt ka simulatsioonis. Seejärel tehakse testsõidud tänaval ning alles peale edukat testimist lubatakse isejuhtivatel sõidukitel kaarti kasutada.

1.4 Lanelet2 kaardiformaat

Lanelet2 on täppiskaartide formaat, mida tutvustasid oma artiklis Fabian Poggenhans ja tema kaasautorid [5]. Autorite sõnul võib maanteedel ja teistel asulavälistel teedel piisata ainult sõiduradade asukoha märkimisest kaardile, kuid suuremates linnades peaks olema võimalus kaardile panna ka jalgteed, jalgrattarajad ja keerukama liikluskorralduse, näiteks milline foor on kehtiv antud sõidurajal. Lisaks peaks kaardil olema informatsioon ka sõiduradadest kaugemal olevate objektide kohta, et sõiduk oskaks hädaolukorras paremini käituda. See tagab allika väitel turvalise autonoomse sõitmise isegi keerulistes liiklusoludes suuremates linnades. Artikli autorid nimetavad kaarte, mis lubavad turvalist kõrgel tasemel autonoomset sõitu (ingl *highly automated driving (HAD)*) HAD kaartideks. Allika sõnul polnud artikli ilmumise ajal olemas ühtegi HAD kaardi nõtule vastavat kaardiformaati ja seetõttu otsustati välja töötada Lanelet2 formaat.

1.4.2 Sõidulõigud

Sõidulõikude (ingl *lanelets*) kontseptsiooni tutvustasid esimest korda oma artiklis Philipp Bender, Julius Ziegler ja Christoph Stiller [6]. Autorid kirjeldavad sõidulõike kui tee segmente, millele on lisatud andmed kirjeldamiseks ümbritsevaid tingimusi. Sõidulõik on kaardil kujutatud kahe murdjoonega, millest üks märgib sõiduraja vasakpoolset äärt ja teine parempoolset äärt ning selleks, et kahte sõidulõiku ühendada peavad mõlema sõidulõigu äärejoonte otspunktid olema üksteisega ühendatud. Sõidusuuna määrab sõidulõigu äärejoone roll. Äärejoonele saab anda parema või vasaku rolli. Kaks sõidulõiku on üksteise naabruses, kui sõidulõikude otspunktid on identsed ja üleminek ühelt sõidulõigult teisele on piisavalt lauge. Naabruses olevad sõidulõigud moodustavad graafi, kus graafi tippudeks on sõidulõigud. Graafi kahe tipu vahel on serv siis, kui tippudele vastavad sõidulõigud on üksteisega naabruses. Saadud graaf võimaldab isejuhtival sõidukil planeerida teekonda. Sõidulõikudele on võimalik lisada kiiruspiiranguid ja muid liikluskorraldust puudutavaid andmeid.



Joonis 1. Sõidulõikude näidis [6].

1.4.3 Lanelet2 primitiivid

Peatükis on refereeritud Fabian Poggenhans ja tema kaasautorite artiklit [5], kus on kirjeldatud Lanelet2 kaardiformaati lähemalt. Lanelet2 formaadis kaart koosneb viiest primitiivist. Geomeetriline primitiiv on ruumilise nähtuse digitaalse vektoresituse element – kas joon, punkt või pind [2]. Lanelet2 formaadi primitiivideks on punktid, jooned, sõidulõigud, alad ja regulatiivsed elemendid. Igal primitiivil peab olema unikaalne ID ning sildid lisatakse primitiivile võtme ja väärtuste paaridena. Igale primitiivile on mõned sildid (ingl *tag*) kohustuslikud, kuid uusi silte saab kaardi täiustamiseks ka ise lisada. Kogu Lanelet2 kaardi struktuur salvestatakse OpenStreetMap (OSM) vormingus XML faili.

Punktid (ingl *points*) on kõige lihtsamad kaardi elemendid. Nad võivad kujutada vertikaalseid objekte, kuid enamasti moodustatakse punktide abil murdjooni. Punktid on ainsad kaardi elemendid, millele on lisatud asukohtaandmed.

Murdjooned (ingl *linestrings*) on järjestatud punktide massiiv. Nende abil märgitakse kaardile tee ääred, äärekivid, teekattemärgistus ja muud sarnased objektid. Murdjooned võivad olla ka virtuaalsed, kui nad kujutavad mingit mõttelist piiri või joont teel.

Sõidulõigud (ingl *lanelets*) on väikseimad kaardi elemendid, kus toimub suunatud liikumine. Sõidulõigud moodustavad tavaliselt sõiduraja, kuid ka jalakäijate ülekäiguradade ja rööbasteede kujutamisel kasutatakse sõidulõike. Väikseima segmendi all peetakse silmas seda, et ühe sõidulõigu piires liikluskorraldus ei muutu. Sõidulõigule on võimalik lisada regulatiivseid elemente, mille abil märgitakse antud sõidulõigule kehtivad liikluseeskirjad. Sõidulõikudest on pikemalt kirjutatud peatükis 1.4.2.

Alad (ingl *areas*) on kaardi elemendid, kus ei toimu suunatud liikumist või pole liikumine üldse võimalik. Aladega kujutatakse parkimiskohti, väljakuid, rohealaseid ja hooneid. Alad koosnevad vähemalt ühest murdjoonest, mis moodustab ala välimise piiri. Lisaks võib ala

koosneda ka ühest või mitmest murdjoonest, mis moodustavad ala sisemisi piire ja kujutavad sellega alas olevaid auke. Ka aladele võib lisada regulatiivseid elemente.

Regulatiivsed elemendid (ingl *regulatory elements*) defineerivad sõidulõikudele või aladele kehtivad liikluseeskirjad, nagu valgusfoorid, eesõiguse reeglid ja kiirusepiirangud. Regulatiivsed elemendid viitavad tavaliselt ka kaardi elemendile, mis liikluseeskirja kehtestas, nagu näiteks liiklusmärgile või valgusfoorile. Lisaks võib regulatiivne element sisaldada viidet ka stoppjoonele.

1.5 Autoware

Autoware on avatud lähtekoodiga tarkvara isejuhtiva sõiduki opereerimiseks. Autoware'i tutvustasid esimest korda oma artiklis Shinpei Kato ja tema kaasautorid [7]. Autoware on arendatud robotite operatsioonisüsteemi (ROS) platvormile. Autorite sõnul võimaldab Autoware tarkvara sooritada kõiki autonoomse sõidu komponente – ümbruskonna tuvastamist, teekonna planeerimist ja sõiduki juhtimist.

Wai Nwe Tun ja teiste [8] artikli kohaselt kasutab Autoware oma kohandatud vektorkaardi formaati, mida järgnevas lõigus kirjeldatakse. Autoware'i vektorkaart on vormindatud komaga eraldatud CSV failidesse ja jagatud 32 erinevaks kategooriaks, millest tähtsamad on punkt (ingl *point*), sõlm (ingl *node*), sõidurada (ingl *lane*) ja sõidujoon (*drlane*). Punktide failis on peamiselt asukohaandmed, sõlmede failis on punktide ID-d, sõidujoonte failis on ära toodud sõiduradade keskjooned ning sõiduradade failis on kirjeldatud, millised sõidujooned on omavahel ühendatud.

Autoware tarkvarale on arendatud ka Lanelet2 laiendus, et lihtsustada Lanelet2 vektorkaardi kasutamist koos Autoware tarkvaraga. Nõuded Lanelet2 kaardile Autoware tarkvaral kasutamiseks on ära toodud laienduse dokumentatsioonis [9]. Dokumentatsiooni kohaselt on võimalik Lanelet2 kaardile lisada lokaalsed koordinaatide ja kõrguse andmeid sisaldavad sildid. Samuti võimaldab Autoware Lanelet2 kaardile märkida valgusfooride lampide asukohad.

2. Tööriista ülevaade

Selles peatükis antakse ülevaade bakalaureusetöö raames valminud tööriistast. Kõigepealt tutvustatakse Tartu Ülikooli isejuhtivate sõidukite labori vektorkaarti ja sellest Autoware'i vektorkaardi genereerimist. Seejärel tuuakse välja Autoware'i formaadi puudused ja põhjendatakse, miks soovib isejuhtivate sõidukite labor katsetada Lanelet2 kaardiformaati. Seejärel kirjeldatakse täpsemalt tööriista tehnilist lahendust ja tuuakse välja arendustöö käigus ületatud takistused.

2.1 Kasutuses olevad vektorkaardid

Seni on Tartu Ülikooli isejuhtivate sõidukite labor autonoomsel sõidul kasutanud Autoware CSV formaadis vektorkaarti. Autoware formaadis kaart genereeritakse labori enda loodud vektorkaardi põhjal. Järgnevalt tutvustataksegi lähemalt labori loodud vektorkaarti ja tuuakse välja Autoware kaardi puudused, mille tõttu soovib labor hakata katsetama ka Lanelet2 formaadis kaardiga.

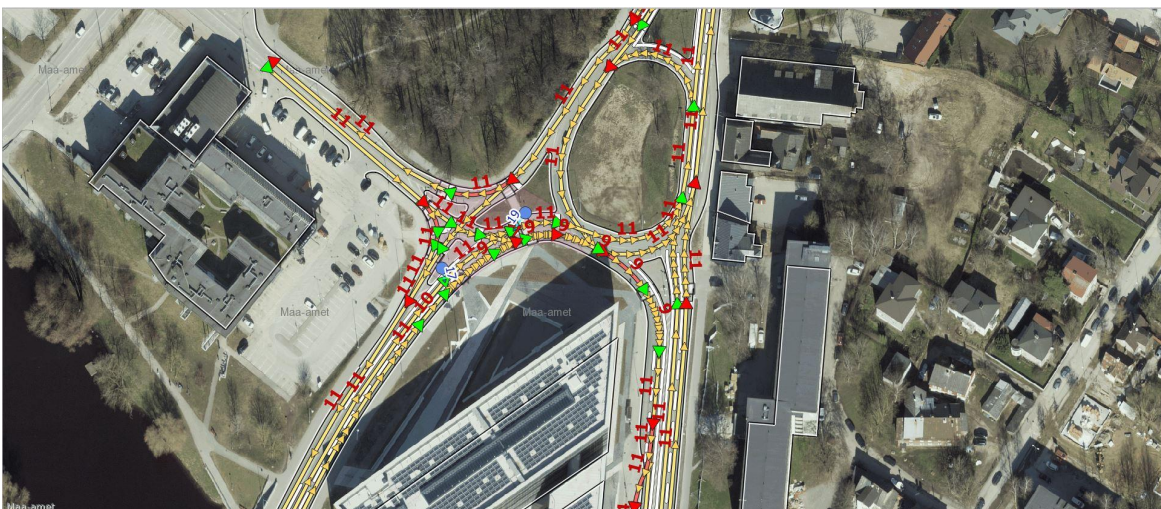
2.1.1 Isejuhtivate sõidukite labori vektorkaart

Tartu Ülikooli isejuhtivate sõidukite labori poolt on välja töötatud vektorkaart, millega on ära kaardistatud testimisrajad nii Tartus kui ka Tallinnas. Kuna isejuhtivate sõidukite labor kasutab autonoomseks sõiduks Autoware tarkvara, siis genereeritakse labori vektorkaardist Autoware kaardiformaadile vastavad CSV vormingus failid. Täpsemalt on labori vektorkaardist Autoware'i kaardi genereerimist kirjeldatud Edgar Sepa magistritöös [10]. Labori vektorkaart koosneb üheksast *shapefile* vormingus kaardikihist. Nendeks kihtideks on *signal*, *roadsign*, *wayarea*, *lane*, *stopline*, *whiteline*, *crosswalk*, *intersection* ja *curb*. Selle bakalaureusetöö raames valminud tööriist kasutab Lanelet2 kaardi genereerimiseks *lane*, *signal* ja *stopline* kihte.

Lane ehk sõiduraja kiht sisaldab andmeid sõiduraja kohta. Kaardil tähistab sõidurada keskjoon, millele on lisatud sildid LW ja RW, mis märgivad vastavalt sõiduraja vasaku ja parema ääre kaugust keskjoonest. Keskjoone küljes on ka *LaneType* silt, mille abil defineeritakse, kas tegemist on otse mineva, vasakule või paremale pöörduva sõidurajaga. Sõiduraja piirkiiruse defineerimiseks on keskjoonel veel kaks silti: *LimitVel* ja *RefVel*. Neist esimene tähistab sõiduraja piirkiirust ja teine soovitatavat kiirust sellel sõidurajal.

Signal ehk valgusfoori kihile märgitakse valgusfoorid. Valgusfoori tähistab kahe punktiga joon. Esimene punkt tähistab valgusfoori asukohta ja teine punkt on snäpitud stoppjoone ja sõiduraja keskjoone lõikumispunkti, et tähistada millisele sõidurajale antud foor kehtib. Snäppimine on nähtuste asukohaline seostamine, kus nende ruumiobjektide koordinaadid peavad ühtima vähemalt ühes punktis [11]. Kaks punkti on omavahel kokku snäpitud siis, kui neil on samad koordinaadid. Igal valgusfoori tähistaval joonel on oma unikaalne ID. Joonel olev *lights* silt defineerib valgusfoori tuled arvu ja järjestuse. *Heights* sildiga tähistatakse iga valgusfoori lambi kõrgust maapinnast. *Hang* ja *Vang* sildid märgivad vastavalt horisontaalset ja vertikaalset nurka.

Stopline ehk stoppjoone kihile kaardistatakse stoppjooned. Stoppjooneks on tavaliselt kolmest punktist koosnev murdjoon, mille keskmine punkt peab olema selle sõiduraja külge snäpitud, millele antud stoppjoon kehtib. Vajadusel on stoppjoone keskmisele punktile snäpitud ka valgusfoor või liiklusmärk. Igal stoppjoonel peaks sarnaselt valgusfooriga olema oma unikaalne ID.



Joonis 2. Tartu Ülikooli isejuhtivate sõidukite labori vektorkaardi näidis. Aluskaardiks on Maa-ameti ortofoto.

2.1.2 Autoware formaadi puudused

Nagu eelpool mainitud, siis genereeritakse isejuhtivate sõidukite labori vektorkaardist Autoware formaadis kaart. Edgar Sepa magistritöös [10] on võrreldud Autoware formaati teiste olemasolevate kaardiformaatidega, sealhulgas Lanelet2 formaadiga, ja toodud välja Autoware kaardi puuduseid. Sepa sõnul on Autoware formaat Jaapani firma Aisan Technology intellektuaalomandis oleva kaardi formaadi lihtsustatud kuju. Kuna Autoware formaat ei ole avatud lähtekoodiga, siis pole võimalik leida ka selle formaadi kohta käivat

dokumentatsiooni. Lisaks toob Sepp välja, et Aotware formaat on vähese paindlikusega, võimalusi omavahel teisendada Aotware formaati ja mõnda teist vektorkaardi formaati on vähe ning Aotware kaarti ei kasuta avalikult ükski teine ettevõte. Aotware formaadi puuduste tõttu on Aotware'i tarkvara Aotware.AI versioonile lisatud Lanelet2 laiendus ja Aotware'i tarkvara järgmise versiooni Aotware.Auto põhiline kaardiformaat on juba Lanelet2. Seetõttu soovib ka isejuhtivate sõidukite labor hakata testimata Lanelet2 formaadis vektorkaarte.

2.2 Tehniline lahendus

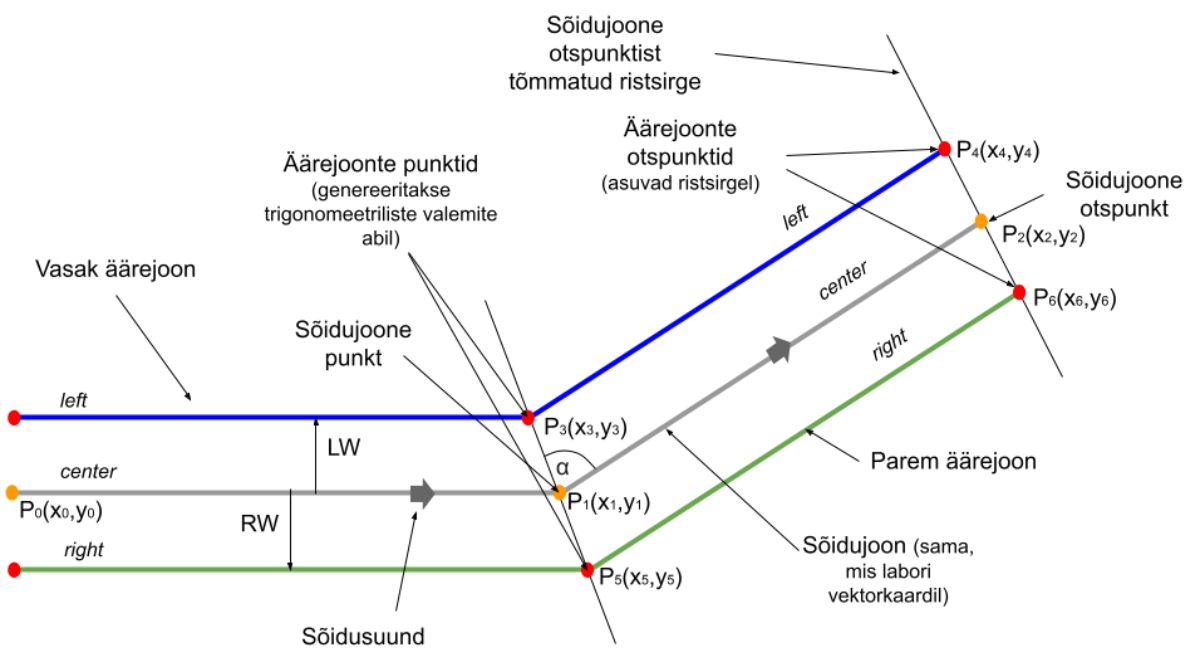
Bakalaureusetöö raames arendati programm, millega teisendada isejuhtivate sõidukite labori vektorkaart Lanelet2 formaati. Teisendaja kirjutati programmeerimiskeeles Python. *Shapefile*'de modifitseerimiseks kasutati Pythoni Geopandas ja Shapely teeke. Openstreetmap formaadis XML faili loomiseks kasutati Pythoni xml standardteeki.

Lanelet2 formaadis tuleb ära märkida sõiduraja ääred, kuid isejuhtivate sõidukite labori vektorkaart kasutas sõiduraja märkimiseks ainult sõiduraja keskjoont ehk sõidujoont. Andmeid sõiduradade äärte ega laiuse kohta vabalt saada ei ole. Maa-ametilt on küll võimalik saada sõiduteede maa-ala andmetega kaardikiht, kuid see ei ole Lanelet2 vajadusi silmas pidades piisavalt täpne. Kuna isejuhtivate sõidukite labori vektorkaardil on sõidujoonele märgitud umbkaudne sõiduraja ääre kaugus sõidujoonest (sildid LW ja RW), siis otsustati genereerida sõidujoonega paralleelsest vasakule ja paremale sõidutee äärt tähistavad jooned. Paralleelsete joonte kaugus sõidujoonest on sama, mis LW ja RW väärtus. Sellise lahenduse puuduseks on, et sõidukil ei ole võimalik sõiduradasid vahetada, sest sõidulõike külgi pidi ei ühendata.

Programmi töö võib üldiselt jagada neljaks etapiks. Kõigepealt toimub igale sõidujoonele paralleelsete äärejoonte genereerimine. Seejärel tuleb sõidulõigu segmentide otspunktid omavahel ühendada. Peale seda genereeritakse valgusfoorid ja defineeritakse vastavad regulatiivsed elemendid. Viimaks tuleb kaart viia shapefile vormingust Openstreetmap XML vormingusse.

2.2.1 Paralleelsete joonte genereerimine

Paralleelsete joonte genereerimisel käiakse esmalt tsükliga läbi iga sõiduraja sõidujoont tähistav murdjoon. Seejärel käiakse järjest läbi kõik sõidujoone punktid ja leitakse trigonomeetria valemite abil vasakule ja paremale genereeritava murdjoone punkt. Punkt peab asuma asukohas, kus vasak ja parem murdjoon oleksid paralleelsed sõidujoonega ja äärejoonte kaugus sõidujoonest oleks sama, mis LW ja RW väärtus. Arvestada tuleb nii sõidujoone sõidusuunaga kui ka sellega, kas murdjoon pöörab vasakule või paremale.



Joonis 3. Sõidujoonele paralleelsete äärejoonte genereerimine.

Järgnevalt on välja toodud valemid, mida kasutati äärejoone keskmiste punktide arvutamisel.

Väärtused x_i ja y_i on joonisel 3 on tähistatud punktide P_i koordinaadid ehk

$$P_i = (x_i, y_i), \text{ kus } i = 0, 1, 2, \dots$$

Kõigepealt leitakse väärtused dx , dy , dx_p ja dy_p

$$dx = x_2 - x_1 \quad dy = y_2 - y_1$$

$$dx_p = x_1 - x_0 \quad dy_p = y_1 - y_0$$

Edasi on võimalik leida väärtus $dist$, mis on vektori $\vec{P_1P_2}$ pikkus, ja väärtus $dist_p$, mis on vektori $\vec{P_0P_1}$ pikkus.

$$dist = \sqrt{dx^2 + dy^2}$$

$$dist_p = \sqrt{dx_p^2 + dy_p^2}$$

Seejärel leitakse, kas sõidulõik keerab vasakule või paremale järgneva valemi abil

$$t = dy_p(dx - dx_p) - dx_p(dy - dy_p)$$

Kui $t < 0$, siis sõidulõik keerab vasakule ja kui $t > 0$, siis sõidulõik keerab paremale.

Kui sõidulõik keerab vasakule

$$x_3 = x_1 + lw \cdot dy_p - \frac{lw \cdot dx}{dist \cdot \tan(\alpha)}$$

$$y_3 = y_1 - lw \cdot dx_p - \frac{lw \cdot dy}{dist \cdot \tan(\alpha)}$$

$$x_5 = x_1 - rw \cdot dy_p + \frac{rw \cdot dx}{dist \cdot \tan(\alpha)}$$

$$y_5 = y_1 + rw \cdot dx_p + \frac{rw \cdot dy}{dist \cdot \tan(\alpha)}$$

Kui sõidulõik keerab paremale

$$x_3 = x_1 + lw \cdot dy_p + \frac{lw \cdot dx}{dist \cdot \tan(\alpha)}$$

$$y_3 = y_1 - lw \cdot dx_p + \frac{lw \cdot dy}{dist \cdot \tan(\alpha)}$$

$$x_5 = x_1 - rw \cdot dy_p - \frac{rw \cdot dx}{dist \cdot \tan(\alpha)}$$

$$y_5 = y_1 + rw \cdot dx_p - \frac{rw \cdot dy}{dist \cdot \tan(\alpha)}$$

lw ja rw on väärtused LW ja RW. Nurk α leitakse järgmise valemi abil

$$\alpha = \frac{1}{2} \left(\pi - \arccos \left(\frac{dx \cdot dx_p + dy \cdot dy_p}{dist \cdot dist_p} \right) \right)$$

Paralleelsete murdjoonte esimesed ja viimased punktid asuvad vastavalt sõidujoone esimest ja viimast punkti läbival ristsirgel. Järgnevalt tuuakse välja valemid sõidulõigu otspunktide leidmiseks.

$$x_4 = x_2 + lw \cdot dy_p$$

$$y_4 = y_2 - lw \cdot dx_p$$

$$x_6 = x_2 - rw \cdot dy_p$$

$$y_6 = y_2 + rw \cdot dx_p$$

Kui kõik sõidujoone punktid on läbi käidud, siis moodustatakse leitud äärejoone punktide abil vasak ja parem murdjoon ja lisatakse need koos sõidujoonega väljundi hulka.

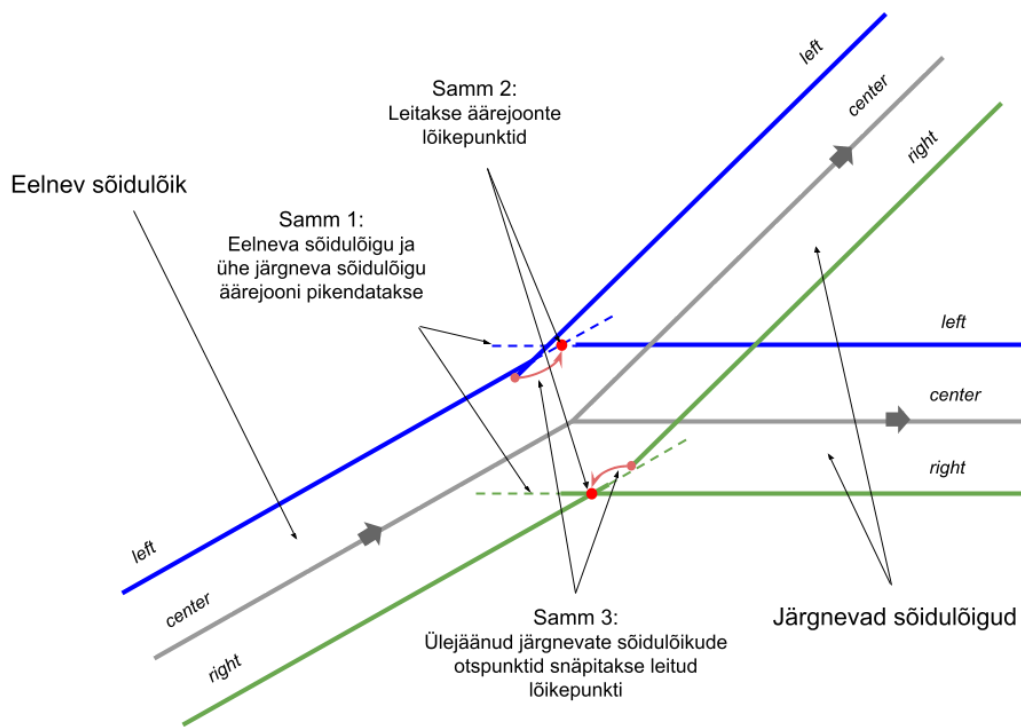
Kuigi Lanelet2 ei nõua sõiduraja sõidujoone olemasolu, siis dokumentatsioonis [12] on välja toodud, et seda on võimalik ise lisada. Sõidujoon otsustati Lanelet2 kaardile märkida pidades silmas olukordi, kui sõiduraja vasaku ja parema ääre kaugus sõidujoonest ei ole võrdne. Siis on sõidukil võimalik juhinduda juba kaardile märgitud sõidujoonest.

Enne väljundi hulka lisamist antakse murdjoontele *left*, *center* või *right* roll. Kõik kolm murdjoont saavad sama ID. Selle ID abil leitakse hiljem üles samasse sõidulõiku kuuluvad jooned. XML vormingus on antud ID kogu sõidulõigu relatsiooni ID. Joontele märgitakse eraldi sildiga ka sõiduraja tüüp (silt *turn_direction*) ehk kas sõidurada on otse sõiduks, vasakule pööramiseks või paremale pööramiseks. Sõiduraja tüüp on vajalik sõiduki suunatuudele lülitamiseks.

2.2.2 Sõidulõikude ühendamine

Sõidulõikude ühendamine osutus programmi kõige keerulisemaks etapiks. Kuigi enamasti tuleb omavahel ühendada vaid kaks sõidulõiku, siis arvestada tuleb ka kolme, nelja või veelgi suurema arvu sõidulõikude ühendamise. Lisaks on vaja pidada silmas sõidulõikude suunda, sest nii ühenduskoha poole kui ka sealt välja võib suunduda mitu sõidurada. Teisendaja snäpib ühte punkti kokku kõigi ühendatavate sõidulõikude vasaku äärejoone otspunktid. Sama tehakse ka parema äärejoone otspunktidega. Edaspidi on eelnevaks sõidulõiguks nimetatud sõidulõiku, mille külge ülejäänud ehk järgnevaid sõidulõigud ühendatakse. Vasaku ja parema otspunkti asukoha leidmiseks pikendatakse eelneva sõidulõigu ja suvaliselt valitud järgneva sõidulõiku äärejooni ning leitakse nende lõikepunkt kasutades Shapely funktsiooni `intersection()`. Lõikepunkti snäpatakse kokku vastavalt kõigi ühendatavate

sõidulõikude vasaku ja parema äärejoonte otspunktid. Kui lõikepunkti ei leitud või kui lõikepunkt sattus sõidulõigu sõidujoone otspunktist liiga kaugemale, siis prooviti teise suvalise sõidulõiguga. Kui rohkem järgnevaid sõidulõike ei olnud, siis eelneva sõidulõigu äärejoonte otspunktide asukohta ei muudetud ja samadesse punktidesse snäpiti kõik järgnevate sõidulõikude äärejoonte otspunktid. Sõidulõike ühendamist on kujutatud joonisel 4.



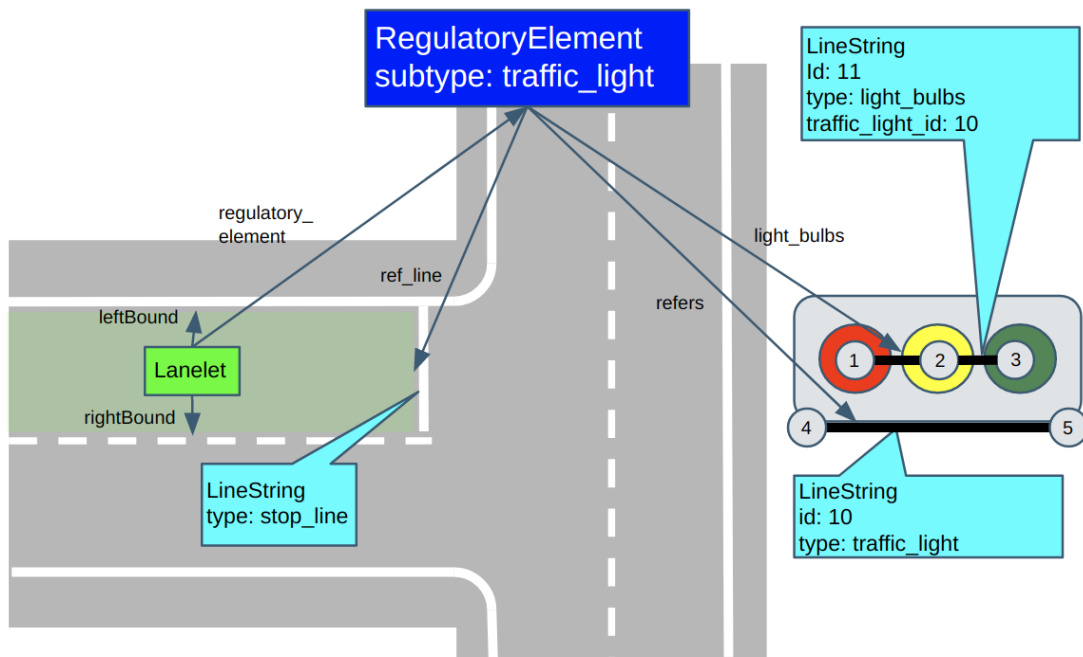
Joonis 4. Eelneva sõidulõigu ühendamine kahe järgneva sõidulõiguga.

Kõigi ühendavate sõidulõikude äärejoonte otspunktide ühte punkti kokku snäppimine muudab nende sõidulõikude laiusi, mille äärejooni ei pikendatud. Samas on sõidulõigu laiuse muutumine enamustel juhtudel piisavalt väike, kuna äärejoone nihkumine toimub ainult otspunkti ja sellele järgneva punkti vahel ning pöörangud sõidulõikude vahel on väga laused. Kuna ka isejuhtivate sõidukite labori vektorkaardil on sõiduradade laius defineeritud 10 cm täpsusega, siis selline väike äärejoonte nihkumine kaardi täpsusele suurt mõju ei avalda. Samuti ei teki probleemi siis, kui ühendada tuleb ainult kaks sõidulõiku. Kui oleks soov nihkumisi vältida, siis peaks sõidulõike tükeldama ja omakorda neid tükke õigesti kokku liitma. Kuna pöörded on laused, siis oleksid uued sõidulõigud vaid mõne sentimeetri pikkused. Programmi töö oleks siis tunduvalt keerulisem ja teisendamine võtaks rohkem aega. Samuti oleks vaja tegeleda rohkemate erijuhtudega. Täpsemalt on Lanelet2 kaardi genereerimisel tekkinud probleeme kirjeldatud peatükis 2.5.

2.2.3 Valgusfoorid

Vastavalt Lanelet2 dokumentatsioonile [13] kasutatakse valgusfooride kaardile märkimisel regulatiivset elementi. Valgusfoori regulatiivne element peab olema antud foorile vastava stoppjoonele eelnenud sõidulõigu relatsiooni liige. Valgusfoori regulatiivse elemendi üheks liikmeks peab kindlasti olema joon, mis märgib foori alumist serva. Seega peab antud joon asuma täpselt foori asukohas. Foori alumist serva märkival joonele on võimalik lisada ka *height* silt, mis määrab ära foori kõrguse. Valgusfoori regulatiivse elemendi teiseks kohustuslikuks liikmeks on stoppjoon, mille taha sõiduk punase tule korral seisma jääb. Autoware dokumentatsiooni [14] kohaselt on võimalik foori regulatiivsele elemendile lisada liikmeks ka valgusfoori lampe tähistav murdjoon. Iga punkt sellisel murdjoonel tähistab ühte valgusfoori lampi. Seetõttu peab igal foori lampe tähistava joone punktil olema *ele* silt, mis määrab lambi kõrguse, ja *color* silt, millega määratakse lambi tule värv. Võimalik on veel lisada *arrow* silt, et tähistada nooltuledega lampe.

Valgusfooride alumist serva tähistava joone genereerimisel kasutatakse isejuhtivate sõidukite labori vektorkaardi valgusfooridel olevat silti *Hang*, mille abil arvutatakse välja nurk joone ja põhjasuuna vahel. Seejärel leitakse joone otspunktid nii, et labori vektorkaardil valgusfoori asukohta tähistav punkt jääks täpselt genereeritava joone keskele. Kuna labori vektorkaardil ei ole foori mõõtmeid märgitud, siis antakse foori alumist serva tähistava joone pikkuseks 0,36 meetrit, kuna keskmiselt on Tartus kasutatavad Swarco foorid just nii laiad. Sarnaselt on foori tuledekomplekti kõrguseks (*height* silt) valitud 1,185 meetrit. Valgusfoori lampe tähistava joone kõik kolm punkti on samade koordinaatidega, kuna Eestis asuvad fooride tuled üksteise all. Lampide punktid asuvad isejuhtivate sõidukite labori vektorkaardi fooride asukohta tähistavate punktide asukohas ehk foori alumist serva tähistava joone keskpunktis. Igale lampi tähistavale punktile antakse määratakse sama kõrgus (*ele* silt), mis oli labori vektorkaardil. Viimaks genereeritakse stoppjooned, mille keskpunkt on samas asukohas nagu labori vektorkaardil. Stoppjoone otspunktid snäpitakse sõidulõigu keskpunktile vastavate äärejoonte punktide külge. Lisaks tuleb arvestada ka sellega, et mõnedele stoppjoontele võib kehtida kaks või enam valgusfoori.



Joonis 5. Valgusfooride märkimine Lanelet2 kaardile vastavalt Autoware nõuetele [14].

2.2.4 OpenStreetMap formaat

Viimaseks tööriista töö etapiks on Lanelet2 kaardi teisendamine OpenStreetMap (OSM) vormingusse. OSM-i kaart on XML kujul ja koosneb kolmest primitiivist: sõlmest (ingl *node*), teest (ingl *way*) ja relatsioonist (ingl *relation*). Lanelet2 punkt vastab OSM-is sõlmele, murdjoon teele ning sõidulõiku, ala ja regulatiivset elementi tähistatakse OSM-is relatsiooniga. Seetõttu saab OSM vormingusse teisendamise jagada kolmeks etapiks. Esimeses etapis lisatakse XML faili kõik punktid sõlmedena, teises etapis lisatakse faili murdjooned teedena ja viimasena lisatakse XML faili sõidulõigud ja valgusfooride regulatiivsed elemendid. Kõige keerulisem osa OSM vormingus XML faili genereerimisel on elementidele ID andmine, sest OSM vormingus kaardil peab igal elemendil olema oma unikaalne ID. Isejuhtivate sõidukite labori vektorkaardil tihti elementidel ID-sid ei olnud või kui oli, siis võisid elementide ID-d üle kihtide korduda. Näiteks võis valgusfooril ja stoppjoonel olla sama ID. Foori elementide ID-d genereeritakse labori vektorkaardi fooride ID-de põhjal. Samas tuleb tähele panna, et mõnikord on ühe regulatiivse elemendi juures kaks või enam foori.

Kuna sõlmed on ainukesed elemendid, millele lisatakse asukohaandmed, siis tuleb arvestada ka erinevate koordinaatsüsteemidega. Isejuhtivate sõidukite labori vektorkaart on L-EST97 (EPSG:3301) koordinaatsüsteemis, kuid OSM vormingus kaardile on vaja sõlmed märkida

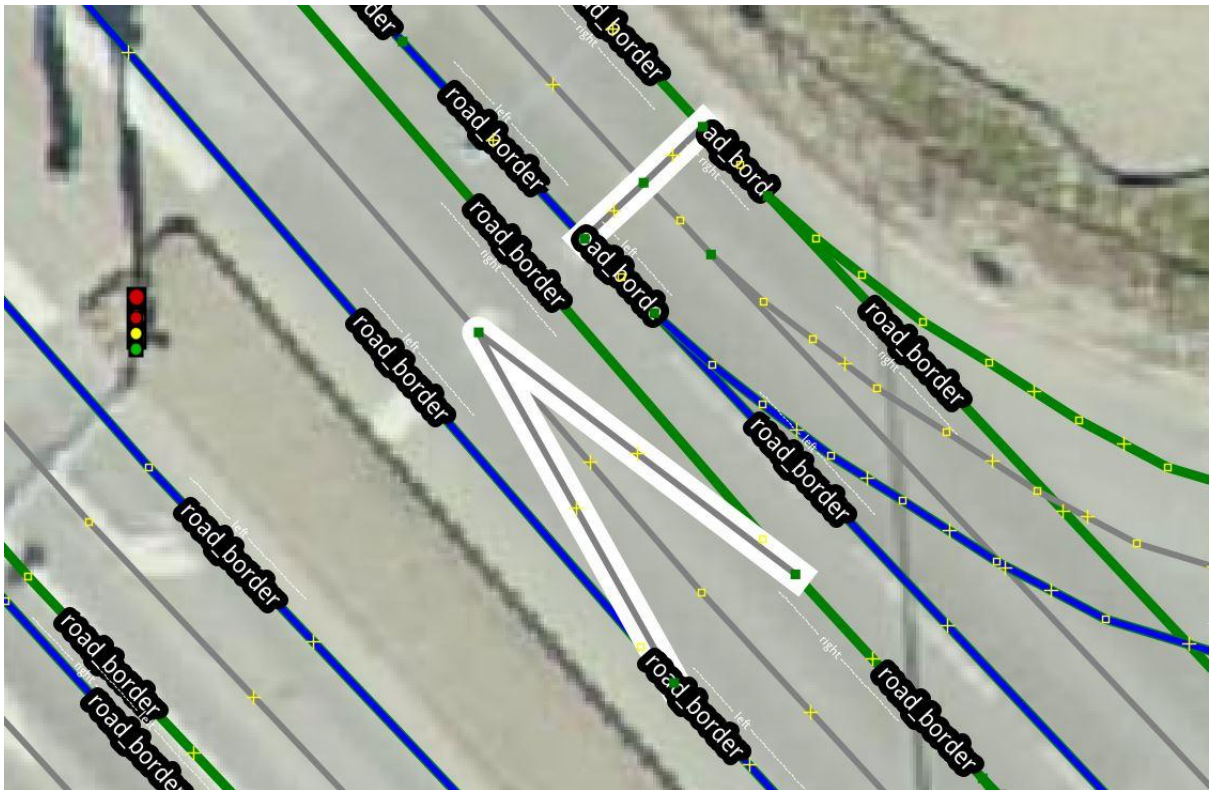
WGS84 koordinaatsüsteemis (ESPG:4326). Samas vastavalt dokumentatsioonile [14] on võimalik Autoware jaoks lisada punktidele ka kohalikud koordinaadid (siltidega *local_x* ja *local_y*). Teisendaja viimase sisendargumendina on võimalik määrata, kas Lanelet2 kaardile lisatakse kohalikud koordinaadid. Kohalikke koordinaate arvutatakse L-EST97 koordinaatide põhjal.

2.3 Lahenduse loomisel ületatud takistused

Igas tööriista peatükis 2.2 kirjeldatud etapil ilmnes ka mitmeid probleeme, mida selles peatükis lähemalt kirjeldatakse. Kuigi tööriista esimene etapp oli kõige lihtsam, tekkis ka selles üksjagu probleeme. Kõige rohkem probleeme oli sõidulõigu äärejoonte punktide õigete asukohtade leidmisega. Esimestel katsetustel ei tahtnud tööriist töötada teravnurkade korral. Kuigi kaardil, mille järgi isejuhtiv sõiduk sõidab, teravnurki tõenäoliselt ei ole, siis oli soov, et teisendaja töötaks õigesti ükskõik milliste sõidujoonte korral. Teised esinenud probleemid esimese etapi juures olid seotud sõidusuuna määramise ja sõidulõigu sõidujoone ja äärejoone vahelise kaugusega.

Kuna sõidulõikude ühendamise on kõige keerulisem tööriista tööetapp, siis tekkis selles etapis ka kõige rohkem probleeme. Peatükis 2.2.2 on kirjeldatud, et ühendatavate sõidulõikude äärejooned on snäpitud kõik ühte punkti kokku, mille tõttu sõidulõikude laius natukene muutus. Tegelikult katsetati esialgu ka lahendust, kus äärejooned pikendati otse eelneva sõidulõigu äärejoone külge. Sellise lahenduse korral sõidulõikude laius ei muutunud, kuid lahenduse implementeerimine oli palju keerukam ja tekitas hulgaliselt probleeme. Kuna sellise lahenduse korral eelnev sõidulõigu äärejoon ei lõppenud punktis, kus see järgneva sõidulõigu äärejoonega lõikus, siis oleks pidanud eelneva sõidulõigu pooleks tegema, kuna sõidulõigud peavad alati hargnema otspunktides. Sõidulõigu poolitamisel oleks olnud vaja leida ka punktid, kust sõidujoon ja teine äärejoon poolitada. Sellist poolitamist oleks olnud võimalik teha kohas, kus hargnevad sõidulõigud kolmes või neljas suunas, kuid suuremate hargnemistega kohtades osutuks see liiga keeruliseks. Samuti tekitas selline lahendus probleeme kohtades, kus mitu hargnemist olid lähestiku. Lisaks oli erijuhte, kui mõni ühendatav sõidulõik ei lõikunud ühegi teise sõidulõiguga. Selline olukord tekkis tihti ringristmikel, kus on palju pööravaid sõidulõike. Suurem probleem tekkis sõidulõikude ühendamisel ka siis, kui järgnev sõidulõik jätkus peaaegu, kuid mitte täpselt samas suunas kui eelnev. Sellisel juhul oli sõidulõikude äärejoonte lõikepunktid keskpunktist väga kaugel ja sõidulõigu ots muutus V-kujuliseks (vt. joonis 6). Selle tõttu muutusid V-kujuliseks ka mõned

stoppjooned. Lahendusena ei pikendatud sõidulõike enam nii palju ja kui lõikepunkti ei leitud, siis ühendati lihtsalt järgneva sõidulõigu äärejoonte otspunktid eelneva sõidulõigu äärejoonte otspunktide külge. Sõidulõikude pikendamisel tekkis ka väiksemaid probleeme. Näiteks olukorras, kui äärejoon ristub teise sõidulõigu äärejoonega rohkem kui ühe korra, siis tuleb leida sõidulõikude ühenduskohale lähim lõikepunkt.



Joonis 6. Lanelet2 kaardi genereerimisel tekkinud vea näide, kus valge stoppjoon vasakpoolsel sõidurajal on V-kujuline. Aluskaardiks on Maa-ameti ortofoto.

Hulgaliselt oli probleeme ka valgusfooride genereerimisel. Esiteks tuli valgusfoorid kaardile märkida Autoware dokumentatsiooni järgi, mis erines mõningal määral Lanelet2 enda dokumentatsioonist ja see tekitas segadust. Teiseks mõjutasid fooride genereerimist isejuhtivate sõidukite labori vektorkaardil olnud vead. Näiteks ei olnud mõni foor korralikult stoppjoone ja sõidurada tähistava joone külge snäpitud. Kuna foorid, stoppjooned ja sõidurajad asusid labori vektorkaardil erinevatel kihtidel, siis eelnev kontrollimine, kas nad on omavahel snäpitud, oli mõnevõrra keerulisem. Sellepärast väljastab tööriist veateated punktidest, kus fooride snäppimine on vigane. Kolmas probleem oli foori lampe tähistavate punktidega. Kuna lambid on üksteise all, siis kaardil peaks nende koordinaadid olema samad.

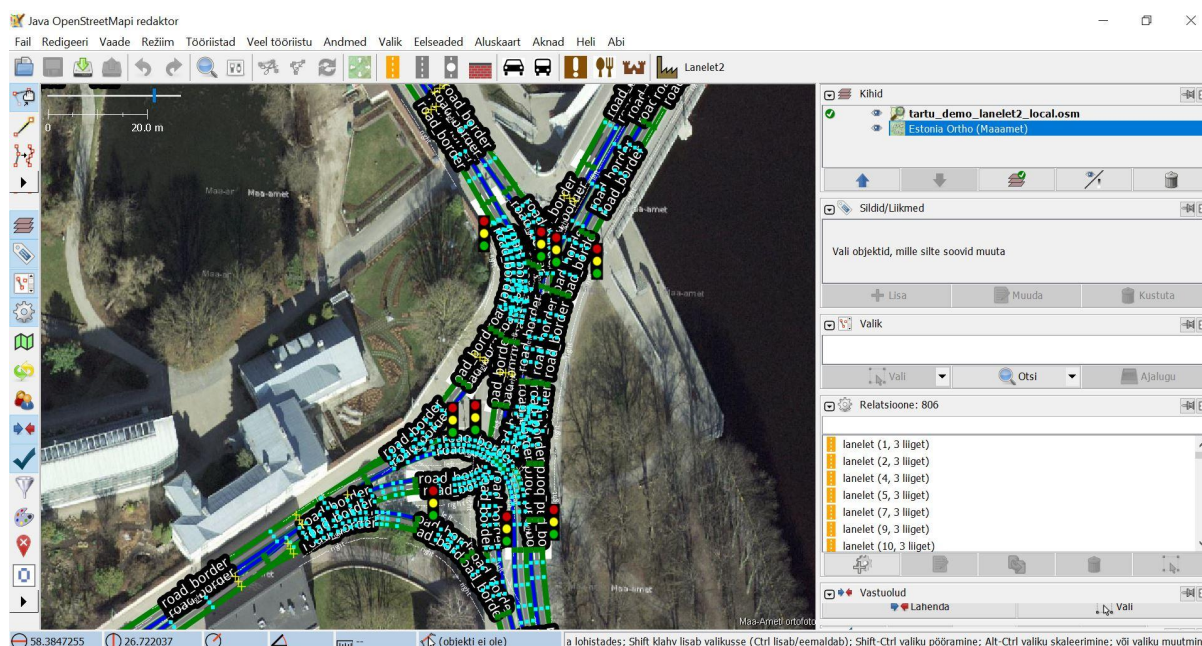
Samade koordinaatide korral snäpitakse aga punktid üksteise külge ja seetõttu võib mõni punkt kaarti OSM vormingusse pannes kaotsi minna. Lahendusena on lampide LEST97 koordinaate muudetud 0,000000005 meetri võrra, et need oleks teineteisest piisavalt erinevad.

Kaardi teisendamisel OSM vormingusse tekkisid probleemid peamiselt kaardi elementidel ID-de genereerimisega. Ükski kaardi element ei tohi viidata ID-le, mida ei eksisteeri, samuti ei tohi ühelgi kaardi elemendil olla ID-ks 0. OSM vormingus ei tohi kahe kaardi elemendi ID olla sama ja ID ei tohi sisaldada tähti. Lisaks ei ole soovitatav anda elementidele negatiivne ID. Kõik eelnevalt nimetatud vead tulid tööriista arenduse käigus ka ette.

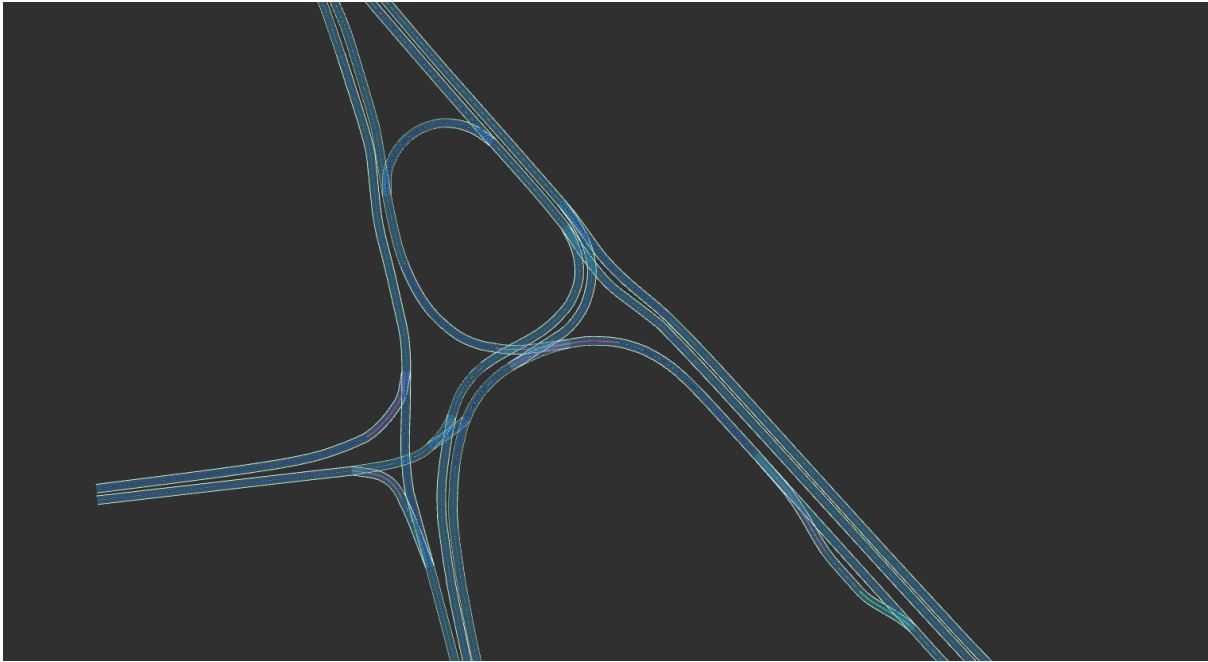
3. Testimine ja edasine arendustöö

Järgmine samm peale tööriistaga Lanele2 kaardi genereerimist oli kaardi testimine. Testimist alustati kaardi avamisega JOSM-i redaktoris (joonis 7), et saada esialgne ülevaade, kas kõik geometriad on genereeritud õigesti. Esinenud vead parandati ja prooviti uuesti. Selline iteratiivne arendustöö toimus iga peatükis 2.2 kirjeldatud etapi juures eraldi enne tööriista esimese versiooni valmimist. Kui tööriista esimene versioon oli valmis ja JOSM-s probleeme ei olnud, siis alustati tööriistaga genereeritud kaardi valideerimist. Kaart valideeriti nii Lanelet2 paketiga kui ka Autoware'iga kaasas olnud validaatoritega. Valideerimise protsessi ja tulemusi on täpsemalt kirjeldatud peatükis 3.1. Peale valideerimist hakati kaardiga katsetama Autoware'is. Kaardi visualiseerimiseks kasutati rakendust Rviz (joonis 8). Sarnaselt varasemale lahendati jooksvalt probleeme, mis kaardi testimisega Autoware'is tekkisid. Viimane samm testimisel oli autonoomne sõit tänaval tööriistaga genereeritud Lanelet2 kaardi järgi. Kuna kõik põhilised funktsionaalsused testsõidu ajal toimisid, siis võis lugeda tänaval testimise õnnestunuks.

Peatükis 3.3 kirjeldatakse võimalusi tulevikus kaardile uute funktsionaalsuste lisamiseks. Kuna paljud Lanelet2 formaadi funktsionaalsused ei ole kujutatud ka isejuhtivate sõidukite labori praegusel vektorikaardil või need on puudulikud, siis tuuakse peatükis välja muudatused, mida peaks labori vektorikaardil tegema, et tööriistaga saaks veel puuduvad funktsionaalsused Lanelet2 kaardile lisada.



Joonis 7. Lanelet2 kaart avatud JOSM-i redaktoris. Aluskaardiks on Maa-ameti ortofoto.



Joonis 8. Lanelet2 kaart avatud Rviz visualiseerijas.

3.1 Valideerimine

Lanelet2 paketiga on kaasas validaator, mille käivitamiseks on vajalik robotite operatsioonisüsteem (ROS). Vastavalt validaatori dokumentatsioonile [15] väljastab validaator kõik kaardilt leitud vead ja hoiatused. Validaatori käivitamiseks on vaja anda käsk `lanelet2_validate <mymap> --lat <lat> --lon <lon>`, kus `<mymap>` on Lanelet2 kaardi faili nimi, `<lat>` ja `<lon>` on vastavalt kaardi asukoha laius- ja pikkuskoordinaadid. Tartu kaardi korral on `<lat>` ja `<lon>` väärtused vastavalt 58 ja 26. Tööriista genereeritud Lanelet2 kaardil ei tohiks olla ühtegi viga ja ka hoiatused tuleks võimalusel eemaldada. Samas ei pruugi kaardile vead tekkida teisendaja tõttu, vaid need võivad olla olemas juba algsel vektorkaardil ja kanduda üle Lanelet2 kaardile. Sama kehtib ka mõnda tüüpi hoiatuste kohta, näiteks annab validaator 35 hoiatust liiga järskudele pööretele. Peamiselt puudutavad hoiatused üleliigseid silte, kuid neid pole eemaldada võimalik, sest Autoware nõuab teatud Lanelet2 kaardi elementidele selliste siltide lisamist, mida Lanelet2 formaadis algselt ette nähtud ei ole. Kuna teisendaja jätab kaardile ka sõidujoone, siis antakse antakse hoiatus ka selle siltide kohta. Sõidujoone siltide 722 hoiatust on loetud kui teisendaja põhjustatud hoiatusi. Ülejäänud 38 teisendaja põhjustatud hoiatused on liiga lähestikku asuvate punktide kohta. Sellised hoiatused tekivad, kui kaks valgusfoori on kõrvuti või sõidulõigu kurvi sisemisel äärejoonel satuvad kaks punkti liiga lähestikku. Seega põhjustab teisendaja kokku

760 hoiatust. Vastavalt validaatori dokumentatsioonile on võimalik ise validaatorit täiendada, kuid selle tegemine jäi antud töö mahust välja.

Autoware'i Lanelet2 laiendusega on samuti kaasas Lanelet2 kaardi validaator. See validaator kontrollib peamiselt Autoware'i nõutud elementide ja siltide olemasolu. Kuna antud bakalaureusetöö raames valminud tööriista eesmärgiks oli genereerida Lanelet2 formaadis kaart, mida kasutatakse koos Autoware'i tarkvaraga, siis peaks genereeritud kaart vastama täielikult Autoware'i nõutele. See tähendab, et Autoware validaator ei tohiks anda ühtegi viga ega hoiatust. Tööriistaga genereeritud kaartidele ühtegi viga ega hoiatust Autoware'i validaator ei andnud. Ülevaade validaatorite väljunditest on välja toodud tabelis 1.

Tabel 1. Ülevaade Lanelet2 kaartide valideerimisest

		tartu_demo_lanelet2.osm (ilma lokaalsete koordinaatidega)		tartu_demo_lanelet2_local.osm (lokaalsete koordinaatidega)	
		Vead	Hoiatused	Vead	Hoiatused
Lanelet2 validaator	Autoware'ile vajalike siltide tõttu	0	1207	0	30209
	Labori vektorkaardilt pärinevad	0	35	0	35
	Teisendaja põhjustatud	0	760	0	760
	Kokku	0	2002	0	31004
Autoware'i validaator	Kokku	0	0	0	0

3.2 Töökiirus

Tulevikus tuleb tõenäoliselt tööriista tööd optimeerida, kui on soov lisada sellele funktsionaalsusi või kasutada seda palju suuremate vektorkaartide teisendamisel. Hetkel teisendatakse Tartu testraja kaart Lanelet2 formaati umbes 30 sekundiga. Mahult väiksemate Cleveroni ja Tallinna testraja kaartide teisendamisele kulub 2–4 sekundit. Tabelis 2 on välja

toodud täpne ajakulu erinevate kaartide teisendamisel ja kaardi suuruse illustreerimiseks on tabelisse lisatud igale kaardile genereeritud sõidulõikude ja valgusfooride arv.

Tabel 2. Tööriista ajakulu erinevate kaartide teisendamisel

		Genereeritud sõidulõikude arv	Genereeritud valgusfooride arv	Ajakulu sekundites
Cleveroni testraja kaart	Ilma lokaalsete koordinaatideta	149	1	2.02
	Lokaalsete koordinaatidega			2.51
Tallinna testraja kaart	Ilma lokaalsete koordinaatideta	197	13	3.19
	Lokaalsete koordinaatidega			4.19
Tartu testraja kaart	Ilma lokaalsete koordinaatideta	722	97	29.12
	Lokaalsete koordinaatidega			31.82

3.3 Edasine arendustöö

Bakalaureusetöö raames valminud teisendaja peamine eesmärk oli genereerida Lanelet2 formaadis kaart, millel on säilinud kõik isejuhtivate sõidukite labori vektorkaardi funktsionaalsused. Seetõttu ei saa genereeritud kaardiga kasutada paljusid funktsionaalsusi, mida Lanelet2 formaat võimaldab. Samas oleks vaja paljude Lanelet2 funktsionaalsuste lisamiseks muuta ka labori vektorkaarti. Järgnevalt on toodud välja mõned funktsionaalsused, mida saaks Lanelet2 kaardile lisada.

Kõige lihtsamalt oleks võimalik kaardile lisada ülekäigurajad. Labori vektorkaardil on ülekäigurajad tähistatud nelinurkse alana, kuid Lanelet2 kaardile tuleb need märkida sõidulõiguna, millel on kaks paralleelset äärejoont. Äärejoonteks saaks võtta labori vektorkaardi nelinurkse ala kaks külge. Lanelet2 kaardil tuleks snäppida sõidulõigu otspunktid ülekäiguraja äärejoonte külge.

Kindlasti tuleks Lanelet2 kaardile lisada eesõigusemärgid. Märk “peatu ja anna teed” ehk STOP-märk lisatakse Lanelet2 kaardile sarnaselt valgusfoorile. Märki juures peab kindlasti olema stoppjoon. Sellele stoppjoonele eelnevale sõidulõigule lisatakse regulatiivne element, mis omakorda viitab stoppjoonele ja STOP-märki tähistava joone ID-le. Selleks, et oleks võimalik STOP-märgid Lanelet2 kaardile lisada, tuleb STOP-märgid panna ka labori vektorkaardile. Labori vektorkaardil tuleks STOP-märke tähistada sarnaselt valgusfooridele kahe punktiga joonega, kus üks punkt tähistab märgi asukohata ja teine stoppjooe asukohta. “Anna teed” märk lisatakse Lanelet2 kaardile sarnaselt STOP-märgiga, kuid “anna teed” märgi puhul tuleb regulatiivsele elemendile ära märkida ka need sõidulõigud, millele auto peab teed andma. Seega tuleks ka labori vektorkaardil märkida “anna teed” märgi juurde sõiduradade ID-d, millele peab teed andma.

Üks peamisi funktsionaalsusi, mida Lanelet2 kaardiformaat võimaldab on sõiduradade vahetus. Selleks, et sõiduradasid saaks vahetada, peavad sõidulõigud omama ühist küljejoont ja see küljejoon peab olema märgitud kui katkendjoon. Teekattemärgistuse ja äärekivide kiht on labori vektorkaardil olemas, kuid andmed nendel kihtidel ei ole täielikud. Kui oleks soov kasutada teekatte märgistuse ja äärekivide kihte Lanelet2 kaardi genereerimiseks, siis tuleks märkida iga sõiduraja paremale ja vasakule äärde teekattemärgistus või äärekivi. Kõige keerulisem olukord on ristmikel, kus sõiduradasid eraldav teekattemärgistus tavaliselt puudub. Üheks võimaluseks on labori vektorkaardile märkida niinimetatud virtuaalsed äärejooned, mis tähistasid ligikaudset sõiduki liikumiskoridori. Teine võimalus on virtuaalsed jooned genereerida automaatselt teatud kaugusele sõidujoonest.

Lanelet2 võimaldab kaardile lisada veel parkimisalasid, jalgrattateid ja rööpaide. Labori vektorkaardil peaks tulevikus kindlasti olema parkimiskohtade kiht. Parkimiskohad võiks olla märgitud aladena, sest siis oleks neid lihtne Lanelet2 kaardile tõsta. Jalgrattateede märkimine labori vektorkaardile eraldi kihina oleks samuti mõistlik, kuna nii Tartus kui ka Tallinnas on tänavaid, mille sõidutee ääres on rattatee. Jalgrattateede märkimine labori kaardile võiks olla sarnane sõiduradadele – kaardile pannakse jalgrattatee sõidujoon. Rööbaste kiht tuleks kaardile märkida siis, kui oleks vaja kaardistada samatasandiline raudteeülesõit või trammitee. Rööbastee märkimine labori kaardil sõidujoonega oleks väga mugav ka Lanelet2 formaati silmas pidades, sest rööpmelaius on rööbasteel rangelt määratud. Lanelet2 kaardile on võimalik märkida ka ohutussaari, piirdeid, hooneid ja tee äärde jäävaid rohealasid, kuid nende lisamine pole esmatähtis. Pikemas perspektiivis tuleks need kuidugi lisada, sest hädaolukorras oskab isejuhtiv sõiduk paremaid otsuseid teha, kui sellel on ümbruskonnast rohkem informatsiooni.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli luua tööriist, mis teisendaks Tartu Ülikooli isejuhtivate sõidukite labori loodud vektorkaardi Lanelet2 formaadis täppiskaardiks. Täppiskaarte kasutavad isejuhtivad sõidukid teekonna planeerimiseks ja ümbruskonnas olevate objektide tuvastamiseks. Tööriista arendamisel lähtuti eelkõige sellest, et kõik funktsionaalsused, mis olid olemas labori kaardi põhjal genereeritud Autoware formaadis täppiskaardil, oleks olemas ka Lanelet2 kaardil.

Tööriist genereerib sõiduradasid tähistavad sõidulõigud, mis on Lanelet2 kaardiformaadi peamised elemendid, ja ühendab need omavahel, et isejuhtival sõidukil oleks võimalik liikuda ühelt sõidulõigult teisele. Seejärel lisab tööriist Lanelet2 kaardile ka valgusfoorid. Viimaks teisendab tööriist kaardi OpenStreetMap (OSM) vormingusse.

Peale kaardi genereerimist valideeriti seda kahe erineva validaatoriga. Esimene oli Lanelet2 paketi kaasas olnud validaator ja teine oli isejuhtivate sõidukite tarkvara Autoware'i validaator. Arendustöö käigus kõrvaldati kõik vead, mis validaatorid leidsid. Eesmärgiks oli vähendada võimalikult palju ka validaatorite hoiatusi. Autoware'i validaator tööriistaga genereeritud kaardi viimasele versioonile hoiatusi ei andnud, kuid Lanelet2 validaator siiski tagastas hoiatusi. Lahendamata jäänud Lanelet2 validaator hoiatused olid enamasti põhjustatud Autoware'i tarkvara jaoks vajalikest siltidest, sõidujoone siltidest, liiga lähestiku asuvatest punktidest või probleemidest isejuhtivate sõidukite labori kaardil.

Valideerimise järel testiti kaarti nii simulatsioonis kui ka realselt tänaval. Kuna isejuhtival sõidukil oli võimalik tänaval tööriistaga genereeritud Lanelet2 kaardi järgi sõita, siis sellega täideti ka bakalaureusetöö peamine eesmärk.

Viimaks toodi välja võimalused tööriista tulevikus täiendada. Käesoleva töö eesmärgiks ei olnud uute funktsionaalsuste lisamine, samas pakub Lanelet2 formaat mitmeid funktsionaalsusi, mida olemasolev Autoware kaart ei võimalda. Seetõttu soovib isejuhtivate sõidukite labor tööriista kindlasti edasi arendada.

Viidatud kirjandus

- [1] Topograafiliste andmete kaardistusjuhend. Maa-amet. 2016.
https://geoportaal.maaamet.ee/docs/ETAK/ETAK_juhend2016.pdf?t=20160708150835
(04.05.2021)
- [2] Terminite loetelu. <http://www.geo.ut.ee/gis2000/terminid.html> (04.05.2021)
- [3] K. Chellapilla, Rethinking Maps for Self-Driving,
<https://medium.com/lyftself-driving/https-medium-com-lyftlevel5-rethinking-maps-for-self-driving-a147c24758d6> (04.05.2021)
- [4] K. Efland, H. Rapp, Semantic Maps for Autonomous Vehicles,
<https://medium.com/lyftself-driving/semantic-maps-for-autonomous-vehicles-470830ee28b6> (04.05.2021)
- [5] F. Poggenhans *et al.*, "Lanelet2: A high-definition map framework for the future of automated driving," 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, 2018, pp. 1672-1679, doi: 10.1109/ITSC.2018.8569929.
- [6] P. Bender, J. Ziegler, C. Stiller, "Lanelets: Efficient map representation for autonomous driving," 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, 2014, pp. 420-425, doi: 10.1109/IVS.2014.6856487.
- [7] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda and T. Hamada, "An Open Approach to Autonomous Vehicles," in IEEE Micro, vol. 35, no. 6, pp. 60-68, Nov.-Dec. 2015, doi: 10.1109/MM.2015.133.
- [8] W. N. Tun, S. Kim, J. Lee H. Darweesh, "Open-Source Tool of Vector Map for Path Planning in Autoware Autonomous Driving Software," 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), Kyoto, Japan, 2019, pp. 1-3, doi: 10.1109/BIGCOMP.2019.8679340.
- [9] Autoware.AI repositoorium. GitLab.
https://gitlab.com/autowarefoundation/autoware.ai/common/tree/master/lanelet2_extension (04.05.2021)
- [10] E. Sepp, Creating High-Definition Vector Maps for Autonomous Driving. TÜ arvutiteaduse instituudi magistritöö. 2021.
https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=71505&year=2021
- [11] Topograafiliste andmete hõive kord ja üldist tähtsust omavad topograafilised nähtused, *Riigi Teataja*. <https://www.riigiteataja.ee/akt/129122013013?leiaKehtiv> (04.05.2021)

[12] Lanelet2 repository, GitHub.

https://github.com/fzi-forschungszentrum-informatik/Lanelet2/blob/master/lanelet2_core/doc/LaneletPrimitives.md (04.05.2021)

[13] Lanelet2 repository, GitHub.

https://github.com/fzi-forschungszentrum-informatik/Lanelet2/blob/master/lanelet2_core/doc/RegulatoryElementTagging.md (04.05.2021)

[14] Autoware.AI repository. GitLab.

https://gitlab.com/autowarefoundation/autoware.ai/common/-/blob/master/lanelet2_extension/docs/lanelet2_format_extension.md (04.05.2021)

[15] Lanelet2 repository, GitHub.

https://github.com/fzi-forschungszentrum-informatik/Lanelet2/tree/master/lanelet2_validation (04.05.2021)

Lisad

I. Tööriista lähtekood

Tööriista lähtekood ei ole avalikult kättesaadav. Lähtekoodi on soovi korral võimalik saada, kui võtta ühendust töö autoriga (e-post: karljohan30@gmail.com).

II. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Karl-Johan Pilve**,

(autori nimi)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose,
Tööriist Lanelet2 täppiskaartide loomiseks

(lõputöö pealkiri)

mille juhendajad on

Tambet Matiisen ja Edgar Sepp,

(juhendaja nimi)

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Karl-Johan Pilve

07.05.2021