

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

Erik Marcus Maks

# Navigatsioonirakenduse sisendi renderdamine autot juhtivale närvivõrgule

Bakalaureusetöö (9 EAP)

Juhendaja: Tambet Matiisen, MSc

# **Navigatsioonirakenduse sisendi renderdamine autot juhtivale närvivõrgule**

## **Lühikokkuvõte:**

Töö eesmärk on teha navigatsioonirakendus, mille ekraanipildi saaks ette anda närvivõrgule. Navigatsioonirakendus saab ette GPS-koordinaadid ja renderdab igal sammul kaardirakenduse ekraanipildi, kuhu on peale joonistatud kogu teekond alguspunktist lõpp-punkti. Närvivõrgu sisendiks oleks kaamerapilt ning navigatsioonirakenduse ekraanipilt, sarnaselt inimeselegi.

## **Võtmesõnad:**

Närvivõrk, Isejuhtiv auto, Navigatsioonirakendus

**CERCS:** P175 Informaatika

# **Navigation application input rendering for a self-driving car's neural network**

## **Abstract:**

This thesis describes the making of a navigation application that can be used as an input for a neural network. The application will receive GPS coordinates and will render step-by-step directions on screen. This rendering will be used together with camera images as an input for the neural network.

## **Keywords:**

Neural network, Self-driving car, Navigation application

**CERCS:** P175 Informatics

# Sisukord

1 Sissejuhatus	4
2 Taust	5
2.1 Modulaarne vs. närvivõrkudel põhinev lähenemine	5
2.2 Erinevad võimalused sihtkoha etteandmiseks närvivõrgule	6
2.2.1 Navigeerimiskäskluse edastamine vasak/parem/otse käsklusena	6
2.2.2 Navigeerimiskäskluse edastamine navigeerimisrakenduse ekraanipildina ja/või GPS-koordinaatidena	7
2.3 Navigatsioonirakendused ja nende programmeerimise võimalused	7
3 Metoodika	9
3.1 Loodud rakendus	9
3.1.1 Kohandatud offline meetod vs. valmis online meetod	9
3.1.2 Arendustööriistad	9
3.1.3 Kasutatud tehnoloogiad	10
3.1.4 Rakenduse struktuur	10
3.1.5 Rakenduse kasutusjuhend	12
3.2 Väljundid	13
3.2.1 Ekraanipilt	13
3.2.2 Tekstiline instruktsioon	14
3.2.3 Struktuurne instruktsioon	14
4 Tulemused	15
4.1 Töökiirus	15
4.2 Täpsus	16
4.3 Edasiarenduse võimalused	17
5 Kokkuvõte	19
Viidatud kirjandus	20
Lisad	21
I. Loodud navigatsioonirakenduse lähtekood	22
Litsents	23

# 1 Sissejuhatus

Isejuhtiv auto on sõiduk, mis suudab tajuda ümbritsevat keskkonda ning liigelda selles ohutult vähese inimsekkumisega. Autonoomsel autol on mitmeid eeliseid tavaautode ees, millest peamine on lootus, et isejuhtiv auto on ohutum kui inimjuht.

Praegu on isejuhtivate sõidukite valdkonnas kasutusel kaks peamist autonoomsuse saavutamise meetodit: modulaarne ja täielikult närvivõrgul põhinev. Mõlemal juhul on isejuhtivast sõidukist kasu vaid siis, kui saab määrata, kuhu sõiduk suundub. Modulaarsel lähenemisel kasutatakse selleks vektorkaarti, kuid närvivõrgule navigeerimiskäskluste edastamiseks on veel teisi võimalusi, näiteks ekraanipildina, tekstilise või struktuurse navigatsioonikäsklusena.

Antud bakalaureusetöö eesmärk on luua navigatsioonirakendus, mille ekraanipildi saaks ette anda närvivõrgule koos tekstiliste ja struktuursete navigatsioonikäsklustega. Navigatsioonirakenduse sisendiks on GPS-koordinaadid, mida töödeldakse, ning seejärel renderdab rakendus igal teekonna sammul kaardirakenduse ekraanipildi, kuhu on peale joonistatud kogu teekond alguspunktist lõpp-punktini. Tulevikus saab genereeritud andmeid kasutada närvivõrgu sisendiks, et seda sõitma õpetada. See navigatsioonirakendus aitaks kaasa isejuhtivate sõidukite valdkonnas eelnevalt nimetatud teisele lahendusmeetodile, täielikult närvivõrgul põhinevale iseõppivate sõidukite arendusele.

Töö on jaotatud neljaks peatükiks. Esimeses peatükis antakse lühiülevaade isejuhtivatest autodest, nende juhtimistarkvarast, navigeerimiskäskluse edastamisest isejuhtiva sõiduki närvivõrgule ja töö eesmärgist. Teises tuuakse välja toetav teoreetiline taust ning uuritakse olemasolevaid lahendusi. Kolmandas peatükis on täpsem rakenduse kirjeldus, millele järgneb tulemuste analüüs neljandas peatükis ning kokkuvõtte viiendas peatükis.

## 2 Taust

Järgneva peatükiga näidatakse antud bakalaureusetöö käigus loodava rakenduse vajadust, tuues näiteid isejuhtivate autode valdkonnas juba tehtud katsetustest, kus on närvivõrgule navigeerimiskäsklusi ette antud. Samuti kirjeldatakse lähemalt erinevaid viise isejuhtiva auto realiseerida ning uuritakse olemasolevaid navigatsioonirakendusi.

### 2.1 Modulaarne vs. närvivõrkudel põhinev lähenemine

Antud peatükk kirjeldab modulaarset ja närvivõrkudel põhinevat isejuhtiva auto tehnoloogiaid lähemalt ning toob välja nende plussid ja miinused.

Modulaarne lähenemine kasutab mooduleid, millel on kindlad ülesanded [1]:

- Tajumoodul – sisendiks on informatsioon erinevatelt sensoritelt, mis koguvad auto ümbrusest infot ja tajumoodul paneb selle õigele kujule teiste moodulite jaoks.
- Positsioneerimise moodul – teeb kindlaks auto asukoha sentimeetri täpsusega, tavaliselt kasutades selleks kõiki või mõnda kombinatsiooni järgnevatest süsteemidest:
  - Globaalne satelliitnavigatsioonisüsteem
  - Inertsiaalandur
  - Punktipilvekaart
  - LIDAR – laserskaneerimisseade.
- Planeerimise moodul – jaotatud kolmeks:
  - Teekonna planeerija – annab ette teekonna teedevõrgu tasemel, sarnaselt näiteks Google Mapsile.
  - Manöövrите planeerija – annab ette järgmise manöövri.
  - Trajektoori planeerija – planeerib auto liikumistee takistuste ja vajalike manöövrите põhjal.
- Juhtimismoodul – teisendab planeerimise mooduli väljundi auto juhtimiskäskudeks.

Kõik need moodulid jagavad isejuhtimise ülesande lihtsamateks alamülesanneteks ja lahendavad väiksemaid probleeme korraga, et saada jagu isejuhtimise probleemist [1].

Tehisnärvivõrkudel põhinev lähenemine õpetab autot käituma sarnaselt inimjuhile. Selleks söödetakse närvivõrgule väga palju andmeid sellest, kuidas inimesed sõidavad autoga, ja lastakse nende põhjal närvivõrgul õppida. Võrgu sisendiks on näiteks kaamerapilt ja

navigeerimiskäsklus ning väljundiks auto juhtimiskäsklused (või täpsemalt rooli ning gaasi- ja piduripedaalide asend) [2].

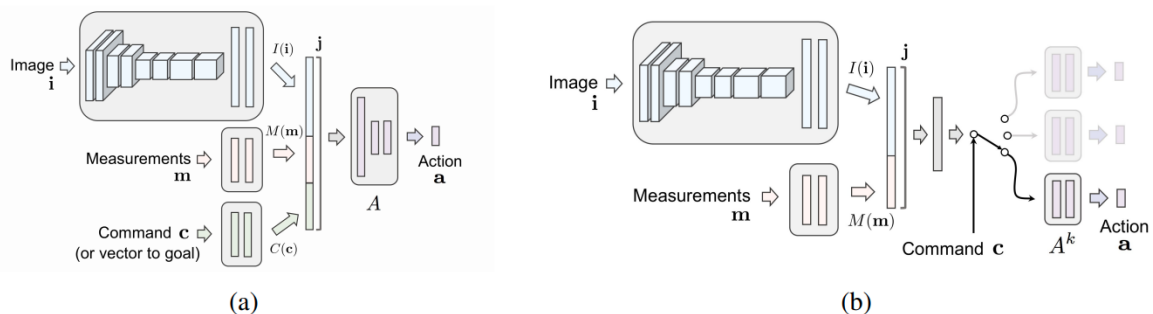
Hetkel on modulaarne lähenemine rohkem levinud ja arenenum, kui tehisnärvivõrkudel põhinev lähenemine. Modulaarne meetod põhineb kaardistatud vektorkaartidel ja etteantud reeglitel, kuidas mingis olukorras käituda. Siiski ei suuda see viis anda täielikku ehk 5. taseme autonoomsust, kuna kõiki olukordi ei ole võimalik autole ette anda ja võib juhtuda, et sõiduk satub olukorda, milles ta pole varem olnud ja seetõttu ei oska käituda nagu vaja. Siinkohal on tehisnärvivõrkudel eelis, sest need suudavad, võrreldes teiste masinõppe meetoditega ja modulaarse lähenemisega, üldistada paremini, et käitumine, mis sobib ühes olukorras, oleks kohane ka teises [2].

## 2.2 Erinevad võimalused sihtkoha etteandmiseks närvivõrgule

Suur osa isejuhtiva auto kasulikkusest on ta autonoomne navigeerimine ja teekonna läbimine. Praegu ei ole aga välja selgitatud parimat viisi selleks, kuidas närvivõrgule edastada navigeerimiskäsklusi. Järgnevalt näidatakse käesoleva bakalaureusetöoga seotud navigeerimiskäskluse edastamise viise.

### 2.2.1 Navigeerimiskäskluse edastamine vasak/parem/otse käsklusena

Antud navigeerimiskäskluse edastamise viis on võetud F. Codevilla jt uurimistööst [3].



Joonis 1. Erinevad viisid närvivõrgule navigeerimiskäsklust sisestada [3].

Üheks viisiks anda närvivõrgule käsklusi, et kuhu sõita, on edastada need navigeerimiskäsklusena ehk anda võrgule koos kaamerapildi ja muude sensorite andmetega lisaks juurde käsklusi, milleks võivad olla „järgi teed“, „paremale“, „vasakule“ või „otse“. Joonisel 1 on selleks variant a.

Antud uurimistöös toodi välja ka teine variant, mida autorid kutsusid *branched* viisiks, mis on joonisel 1 variant b. Lihtsustatult öeldes ei oleks navigeerimiskäsklus võrgu sisendiks otse, vaid toimiks lülitina, mis suunab võrgu väljundi õigesse närvivõrgu lõppharusse.

Selle uurimusega näidati vajadust navigeerimiskäskluse ette andmiseks närvivõrgule, kuid siiski on selle meetodi täiustamiseks veel palju ruumi, millest üks osa on sõiduki juhtimiseks loomuliku keele kasutamine, sest antud töös kasutati vaid fikseeritud käsklusi.

### **2.2.2 Navigeerimiskäskluse edastamine navigeerimisrakenduse ekraanipildina ja/või GPS-koordinaatidena**

S. Heckeri jt [4] poolt kasutatud närvivõrk ehitati selliselt, et planeeritud teekonna, sõiduki eelnevate olekute ja hetke teolude põhjal annaks võrk väljundiks soovitud juhtimiskäskluse. Täpsemalt antakse võrgule ette panoraamvaates video, planeeritud teekond, sõiduki asukoht, rooli pöördenurk ja sõiduki kiirus. Väljundiks on järgmise ajahetke rooli pöördenurk ja sõiduki kiirus.

Navigatsioonikäsklust edastati närvivõrgule kahel kujul: TomTom navigatsiooniseadme ekraanipildina ja OSM navigatsioonisüsteemi GPS-koordinaatidena. Parema tulemuse andis ekraanipildi kasutamine, kuna GPS-koordinaadid ilma töötlemiseta piisavat infot ei edasta.

Leiti, et kõige paremini toimib mudel, mis kasutab kaameraid panoraamvaate saavutamiseks ja navigatsioonirakendust käskluste edastamiseks ekraanipildina.

Samuti näitavad J. Hawke jt [5] oma uurimistöös, et ekraanipildi kasutamine on efektiivne viis närvivõrgule navigeerimiskäskluste andmiseks. Hawke jt. [5] uurimus on esimene eksperiment, kus õpetati mudelit, mis suudab juhtida täissuuruses autot reaalses liikluses, kasutades tingimuslikku imitatsiooniõpet.

Nende uurimustega näidati, et täielikult närvivõrgul põhinevad isejuhtivad autod suudavad kasutada efektiivselt ära navigatsioonirakenduse ekraanipilti ja panoraamvaadet.

## **2.3 Navigatsioonirakendused ja nende programmeerimise võimalused**

Navigatsioonirakendused on programmid, mis annavad teekonnajuhised etteantud asukoha ja sihtkoha põhjal.

Navigatsioonirakendused võib liigitada selle järgi, mille peal need töötavad. Järgnevalt on toodud enim kasutatavad:

- Veebirakendus – näiteks Bing Maps [6]

- Telefonirakendus – näiteks Google Maps [7]
- Töölauarakendus – näiteks NavigatorFREE [8]
- Eraldi navigatsiooniseadmel toimiv rakendus – näiteks TomTom Go Discover [9]

Kõik need rakendused võimaldavad uurida kaarti ja saada teekonnajuhised, kuid ei paku otseselt programmeerimise võimalusi ega viisi automatiseerida teekonnajuhiste küsimist.

Programmeerimiseks sobivad rohkem kohandatavad navigatsioonirakendused. Sellistel rakendustel eksisteerivad teegid, mida saab kasutada oma loodud programmis ja seeläbi automatiseerida paljud funktsionaalsused, nagu näiteks teekonnatrajektoorie kuvamine. Selleks sobivad näiteks QT Map Viewer [10] ja Mapbox GL JS [11].

Kolmas võimalus teekonnajuhiste küsimiseks on kasutada ainult liidest, andes neile GPS-koordinaate. Sellisel juhul tuleks liidestest saadud informatsiooni põhjal renderdada kaart ja teekond sellel. Liideseks sobivad näiteks Mapbox Directions API [12] ja Google Maps Directions API [13]. Kuigi antud liideseid on tasuta kuni kindla arvu päringuteni, siis käesoleva bakalaureusetöö seisukohalt on neil siiski negatiivne külg, need lubavad juhiseid küsida maksimaalselt 25 teekonnapunkti kohta. See võib jääda väheks käesolevas töös loodud rakenduse jaoks, kus isejuhtiva auto poolt sõidetud koordinaadid ja rakenduse loodud ekraanipilt peavad olema väga täpselt sünkroonis. Tulemuste peatükis võrreldakse, kas loodud rakenduse jaoks on piisav kasutada vaid 25 teekonnapunkti.

## 3 Metoodika

Järgnev peatükk kirjeldab loodud rakendust, selle põhimõtet ning struktuuri lähemalt ja tuuakse välja rakenduse väljundid.

### 3.1 Loodud rakendus

Antud bakalaureusetöö käigus valmis rakendus, mis väljastab erinevaid navigatsioonikäsklusi etteantud GPS-koordinaatide põhjal. Rakenduse baasiks on võetud Mapboxi guidance-sim [14] programm, mis toimis algselt vaid Linux operatsioonisüsteemis ja seda vigaselt. Rakendus tehti korda, lisati funktsionaalsusi ning pandi töötama ka Windows arvutis.

#### 3.1.1 Kohandatud *offline* meetod vs. valmis *online* meetod

Esimene võimalus treenimisandmetele navigatsioonijuhiste lisamiseks on tagantjärele ehk *offline* meetodiga andmete loomine. See põhimõte lubab ette anda GPS-koordinaadid ja simuleerida teekonna läbimist arvutis lokaalselt ilma andmete loomise hetkel teed reaalselt läbimata.

Teine võimalus on koguda andmeid reaalajas ehk *online* meetodiga. Kõige lihtsam viis *online* meetodiga andmeid koguda on kasutada Android nutitelefoni ja mõnda mobiilirakendust, näiteks Google Mapsi. Planeeritav teekond tuleks sisestada antud rakendusse ja seda sõidukiga läbima hakata. Samal ajal peab telefoni ekraanipilti salvestama mõne rakendusega, näiteks Androidi sisseehitatud ekraanisalvestajaga [15].

*Offline* meetodi eelised *online* meetodi ees on järgnevad:

- Saab kasutada varasemalt salvestatud sessioone.
- Võimalik korrigeerida ajalist sünkroniseerimist ekraanipiltide ja GPS-koordinaatide vahel.
- Andmeid saab luua kiiremini kui reaalajas antud teekonda läbi sõites.

Antud bakalaureusetöö käigus loodud rakendus kasutab *offline* meetodit eespool mainitud eeliste tõttu.

#### 3.1.2 Arendustööriistad

Rakenduse loomisel kasutati Visual Studio Code ja Google Chrome rakendusi. Järgnevalt kirjeldatakse neid lähemalt.

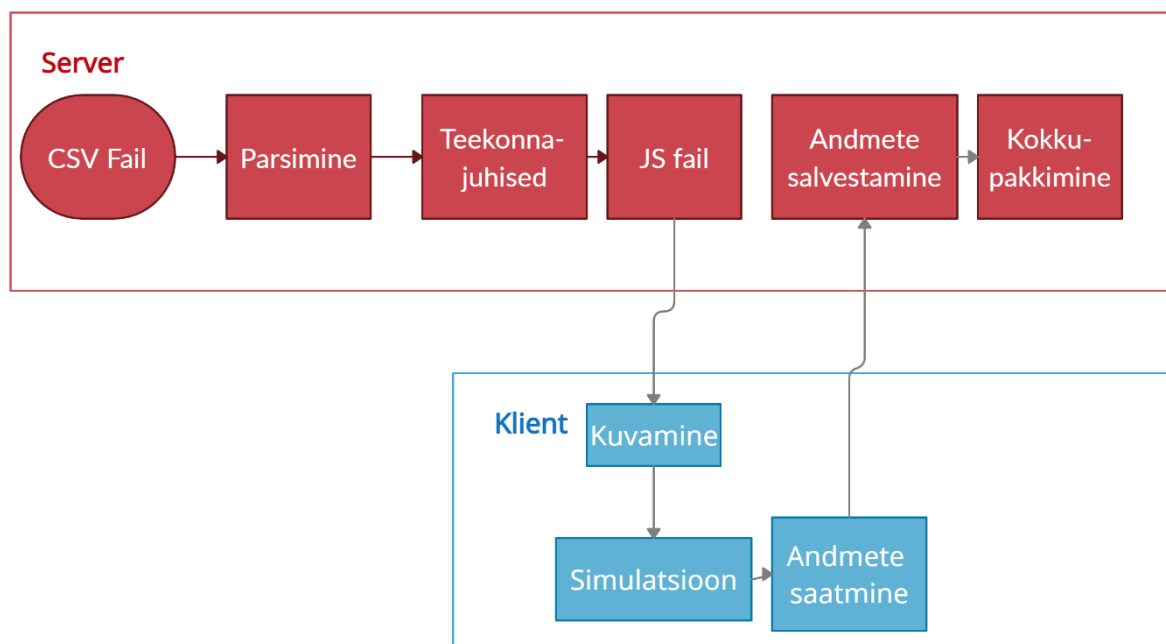
Visual Studio Code on Microsofti poolt loodud integreeritud programmeerimiskeskond [16], mis toetab paljusid erinevaid programmeerimiskeeli. VSC-l on väga ulatuslik pistikprogrammide tugi, mis teeb selle multifunktsionaalseks tööriistaks. Arendusel kasutati programmi 1.55.2 versiooni.

Google Chrome [17] on veebibrauser, mis on viimase 12 kuu jooksul enim kasutatav maailmas [18]. Seetõttu valiti antud brauser rakenduse loomisel tööriistaks, et testida ja käivitada valminud rakendust. Arenduse käigus kasutati versiooni 89.0.4389.114.

### 3.1.3 Kasutatud tehnoloogiad

Valminud rakendus põhineb Mapboxi poolt loodud guidance-sim rakendusel, on kirjutatud Javascript programmeerimiskeeles ja kasutab serveri poolel NodeJS käitussüsteemi. Rakendus kasutab Mapbox GL JS teeki, et luua kaart ning kuvada sellel teekonnatrajektoor ja auto asukoht. Samuti on rakendatud arenduses versioonihaldussüsteemi GIT ning teekidehaldussüsteemi NPM.

### 3.1.4 Rakenduse struktuur



Joonis 2. Rakenduse struktuur

Rakendus koosneb kahest poolest: server ja klient. Serveri pool vastutab etteantud GPS-koordinaatide põhjal teekonna simulatsiooni loomise eest ning kliendi pool läbib antud

simulatsiooni ja saadab genereeritud ekraanipildi koos teekonna juhistega serverile, kus need salvestatakse.

Rakenduse töökäigu (Joonis 2) võib jagada järgnevateks osadeks:

1. Rakenduse käivitamine järgides kaasa antud kasutusjuhendit.
2. CSV andmefaili saatmine serverile.
3. CSV faili parsimine, et saada vajalikud andmed ehk GPS-koordinaadid sealt kätte.
4. Saadud andmetega Mapbox Directions API-st teekonnajuhiste pärimine.
5. Saadud teekonnajuhistega simulatsiooni Javascript faili loomine.
6. Loodud Javascript faili käivitamine kliendi brauseris.
7. Javascript faili poolt loodud simulatsiooni läbimine brauseris.
8. Simulatsiooni käigus loodud ekraanipiltide sünkroonimine GPS-koordinaatide ja navigeerimiskäsklustega.
9. Loodud ekraanipiltide ja teiste andmete saatmine serverile.
10. Saadud andmete salvestamine serverisse.
11. Salvestatud andmete kokkupakkimine ning kliendile nende allalaadimise võimaldamine.

```
--  
  "maneuver": {  
    "type": "depart",  
    "instruction": "Drive south on 3/Narva mnt.",  
    "bearing_after": 186,  
    "bearing_before": 0,  
    "location": [  
      26.726307,  
      58.385264  
    ]  
  },  
  "name": "Narva mnt",  
  "duration": 24.847,
```

Joonis 3. Näide Mapbox Directions liidese vastusest.

Rakendus võimaldas algselt vaid kaardipildi kuvamist koos teekonnaga ning toimis ainult Linux'i peal. Rakendusele lisati järgnevad funktsionaalsused autori poolt:

- Windowsi peal rakenduse toimimise panek - algne rakendus toimis vaid Linux operatsioonisüsteemil, kuna toetus .sh failidele. Rakenduselt eemaldati antud failide vajadus.

- Isejuhtiva sõiduki CSV-faili lugemine ning töötlemine - algselt ei olnud võimalik anda rakendusele ette töötlemata GPS-koordinaate. Antud funktsionaalsus lisati ning programm/rakendus töötab kõikide CSV-failidega, kus eksisteerivad lon (longitude) ja lat (latitude) lahtrid GPS-koordinaatide jaoks.
- Teekonnatrajektoori küsimine veebiliidesest - koos eelnevalt mainitud CSV-faili töötlemise funktsionaalsusega lisati veel automaatne navigatsioonitrajektoori pärimine Mapbox Directions liideselt. Antud liides toetab korraga vaid 25 GPS-koordinaadi kohta päringu tegemist, kuid isejuhtival sõidukil tekib neid tuhandeid juba paariminutilise sõiduga. Seetõttu küsib lisatud funktsionaalsus liideselt 25 koordinaadi kaupa trajektoore ning liidab need lõpus kokku, et saada võimalikult täpne teekond, mis kattuks isejuhtiva auto andmetega. Kuna kasutatav liides kohtleb igat koordinaadipaari kui eraldi teekonda, siis tuleb tagastavate andmetega teha veel järeltöötlust. Esmalt liidetakse päringu vastuses olevad teekonnad kokku üheks pikaks teekonnaks. Teiseks eemaldatakse teekonna viimase lõigu andmed ja liidetakse kokku järgneva päringu esimeste instruksioonidega. Selle tagajärjel võib tekkida rakenduse läbitud teekonnas imelike kohti, näiteks edasi-tagasi liikumist seal, kus seda algandmetes polnud. Sellele probleemile pakutakse lahendus tulemuste peatükis.
- Ekraanipiltide salvestamine koos navigatsioonikäsklustega ning GPS-koordinaatidega sünkronimine - antud funktsionaalsuse lisamise tegi raskeks asjaolu, et ei ole ühtegi head viisi saata brauseris kuvatud ekraanipilti veebiserverile. Ainukene Javascripti sisseehitatud võimalus selleks salvestab kogu ekraani või rakenduse akent, [19] kuid sellist funktsionaalsust loodud rakendusel pole vaja. Probleemi lahendamiseks kasutati teeki `html2canvas` [20]. Antud teeki ei tee ekraanipilti, vaid ehitab Javascript `canvas` objekti ning muudab selle pildiks, mis saadetakse koos GPS-koordinaatide ning navigatsioonikäsklustega serverile.
- Salvestatud andmete kokkupakkimine - rakenduse töö lõpus pakitakse saadud pildid, algandmed ning genereeritud andmed kokku ZIP-failiks.

Rakenduse väljund on toodud välja eraldi alapeatükis allpool.

### 3.1.5 Rakenduse kasutusjuhend

Rakendus tuleb alla laadida GitHub repositooriumist (Lisa 1).

Eeldatakse, et NodeJS ja NPM on juba arvutis, kus antud programmi üritatakse käivitada, olemas. Samuti rakenduse kliendi poolt soovitatakse jooksutada Google Chrome

veebibrauseriga, millel arendamine ning testimine toimus. Veebibrauseri aken peaks rakenduse töötamise käigus olema aktiivne.

Etapid, et rakendust lokaalselt jooksutada on järgnevad:

1. Linuxi puhul terminalis, Windowsi puhul *command prompt*-i kasutades tuleb jooksutada käsklus `„npm install“`.
2. Jooksutada käsklus `„node server.js“` rakenduse juurkaustas.
3. Brauseris avada leht `„localhost:3000“`.
4. Üleslaadida CSV fail, kus on „lat“ ja „lon“ nimelised veerud, määrata rakenduse kiirus ning käivitada simulatsioon vajutades „Submit“ nuppu.
5. Antud etapis käivitub simulatsioon ning peale selle lõppemist suunatakse järgmisele lehele.
6. Sellel lehel saab simulatsiooni loodud failid alla laadida ning samuti antakse võimalus minna pealehele tagasi, et uuesti simulatsiooni käivitada.

## 3.2 Väljundid

Rakenduse loomisel on valitud, et programmi väljundis oleks järgnevad navigeerimiskäskluse edastamise meetodid: navigatsioonirakenduse ekraanipilt, tekstilised navigatsioonikäsklused, struktuursed navigatsioonikäsklused.

### 3.2.1 Ekraanipilt



Joonis 4. Rakenduse loodud ekraanipilt.

Ekraanipildi kasutamine närvivõrgu sisendis suurendab isejuhtiva auto võimet jõuda õigesse sihtpunkti [4.5]. Seetõttu salvestab ka loodud rakendus ekraanipilte (Joonis 4) valitud intervalliga, mis peavad olema sünkroonis auto GPS-koordinaatidega. Ekraanipilt näitab visuaalselt eesolevat teekonda ning auto asukohta kindlal ajahetkel. Ekraanipilt saadakse kasutades Javascript teeki Mapbox GL JS, mis omakorda saab kaardi info OpenStreetMap liidesest [21].

### 3.2.2 Tekstiline instruktsioon

Instruktsiooni sisu tuleneb Mapbox Directions liidesest, kust saab etteantud GPS-koordinaatide järgi pärida teekonnajuhised. Antud liides tagastab JSON formaadis (Joonis 4) samm-sammu haaval juhised sihtpunkti jõudmiseks koos tekstiliste instruktsioonidega. Loodud rakendus seob need ekraanipildi ja GPS-koordinaatidega igal ajahetkel.

	long	lat	text	type	modifier	driving_side
0	26.726328	58.385364	Drive south on 3/Narva mnt.	depart	undefined	right
1	26.726328	58.385364	Drive south on 3/Narva mnt.	depart	undefined	right
2	26.726328	58.385364	Drive south on 3/Narva mnt.	depart	undefined	right
3	26.72630726	58.38526538	Drive south on 3/Narva mnt.	depart	undefined	right
4	26.72629173	58.38518683	Drive south on 3/Narva mnt.	depart	undefined	right
5	26.7262695	58.38507156	Drive south on 3/Narva mnt.	depart	undefined	right

Joonis 5. Rakenduse väljund.

Käesolevas töös loodud rakenduse väljund sisaldab detailset tekstilist instruktsiooni iga GPS-koordinaadi kohta (Joonis 5). Instruktsioon ütleb täpselt, millise manöövri järgmisena peab tegema.

### 3.2.3 Struktuurne instruktsioon

Rakendusele lisati veel struktuurse instruktsiooni väljund (Joonis 5), et tulevikus katsetada selle efektiivsust närvivõrgul. Struktuurne instruktsioon annab koos navigatsioonikäsklusega juurde veel järgneva informatsiooni:

- *type* - manöövri tüüp, mis võib olla *turn, new name, depart, arrive, merge, on ramp, off ramp, fork, end of road, continue, roundabout, rotary, roundabout turn, notification, exit roundabout, exit rotary*.
- *modifier* - juhul kui manööver muudab suunda, siis antakse see ka kaasa. Võimalikud väärtused on: *uturn, sharp right, right, slight right, straight, slight left, left, sharp left*.
- *driving\_side* - näitab kummal pool teed liigeldakse antud GPS-koordinaatidel.

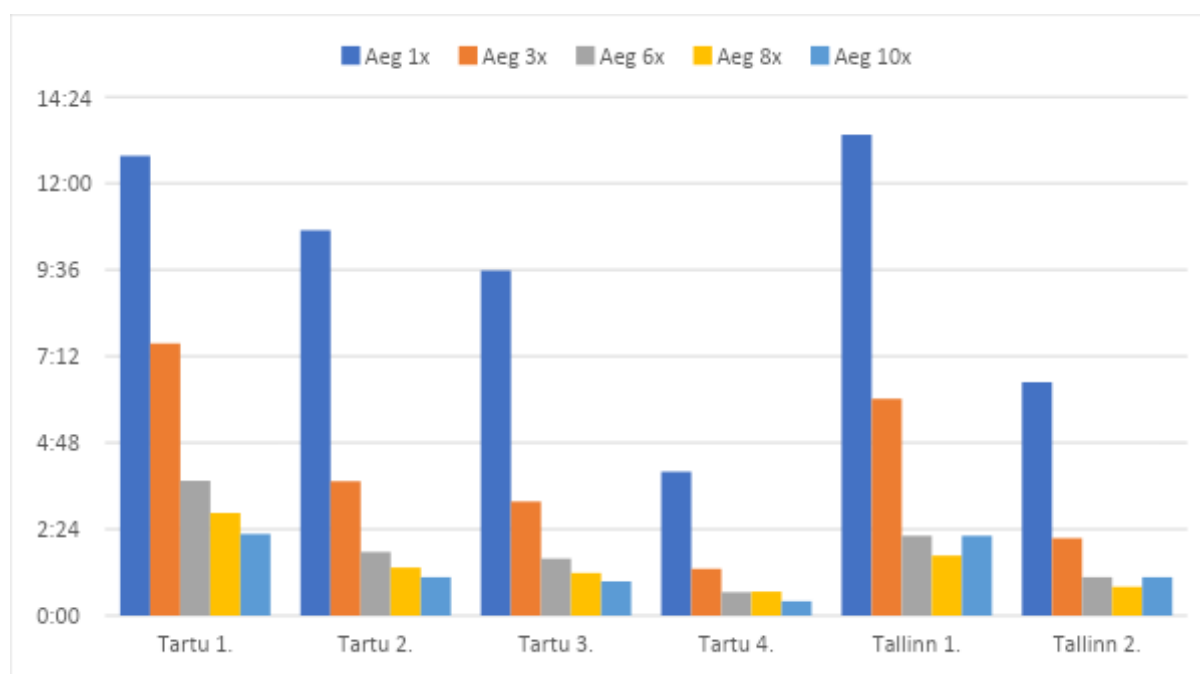
Käskluste sisu saadakse Mapbox Directions liidesest nagu ka tekstilise instruktsiooni puhul.

## 4 Tulemused

Järgnev peatükk analüüsib rakenduse tulemusi ning efektiivsust. Andmed on kogutud käivitades rakendust erinevate parameetritega ning nende võrdlemiseks on kasutatud loodud Pythoni skripte.

### 4.1 Töökiirus

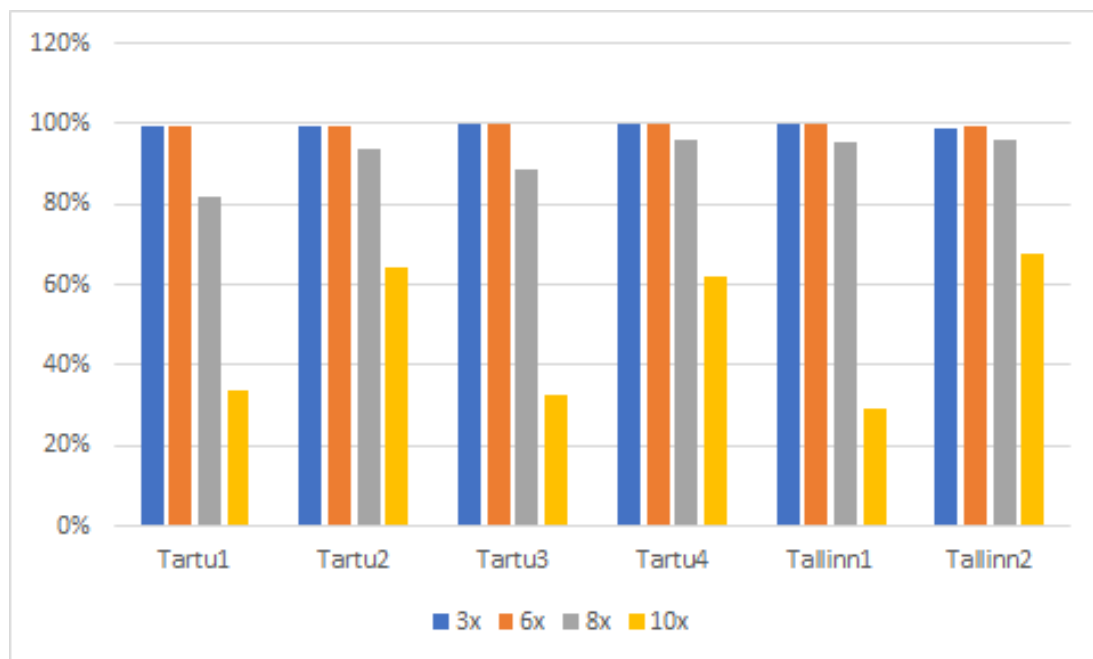
Antud rakenduse kasutamisel on võimalus andmeid genereerida kiiremini kui reaalajas. Kiiruse määramisel on võimalik valida mitu korda kiiremini reaalajast antud simulatsioon töötab. Järgnevalt võrdleb autor rakenduse efektiivsust kiirustel 1x, 3x, 6x, 8x ja 10x, kus 1x tähendab rakenduse töötamist reaalajaga võrdsel kiirusel, 3x reaalajast 3 korda kiiremini jne.



Joonis 6. CSV Faili töötlemise aeg

Joonisel 6 on näha, kuidas muutes rakenduse kiirust muutub ka aeg, mis kulub rakendusel andmete genereerimiseks. Joonise X-teljel on kasutatud CSV andmefail ja Y-teljel kulunud aeg formaadis minutid:sekundid. Erinevad värvid näitavad etteantud kiiruse parameetrit rakendusele. Kuigi 3x aeg peaks olema 3 korda kiirem kui 1x aeg, siis kõigi CSV failide peal see nii ei ole. Autor pakub välja, et see juhtub, kuna kiiruse kordaja kiirendab küll simulatsiooni töökiirust, kuid kasutatava Mapboxi API vastuses olevad hinnangulised kiirused erinevad teelõikudes. Rakendus kasutab antud hinnangulisi kiiruseid, et teada kui kiirelt läbida simulatsioonis mingi teelõik ning see mõjutab ka rakenduse töökiirust. Teisisõnu kõik

simulatsiooni etapid pole kiirendatud täpselt vastava kordaja võrra. Lahendus sellele probleemile oleks luua uus või kasutada mõnda muud moodulit, mis vastutab teekonnatrajektoori läbimise eest loodud rakenduses.



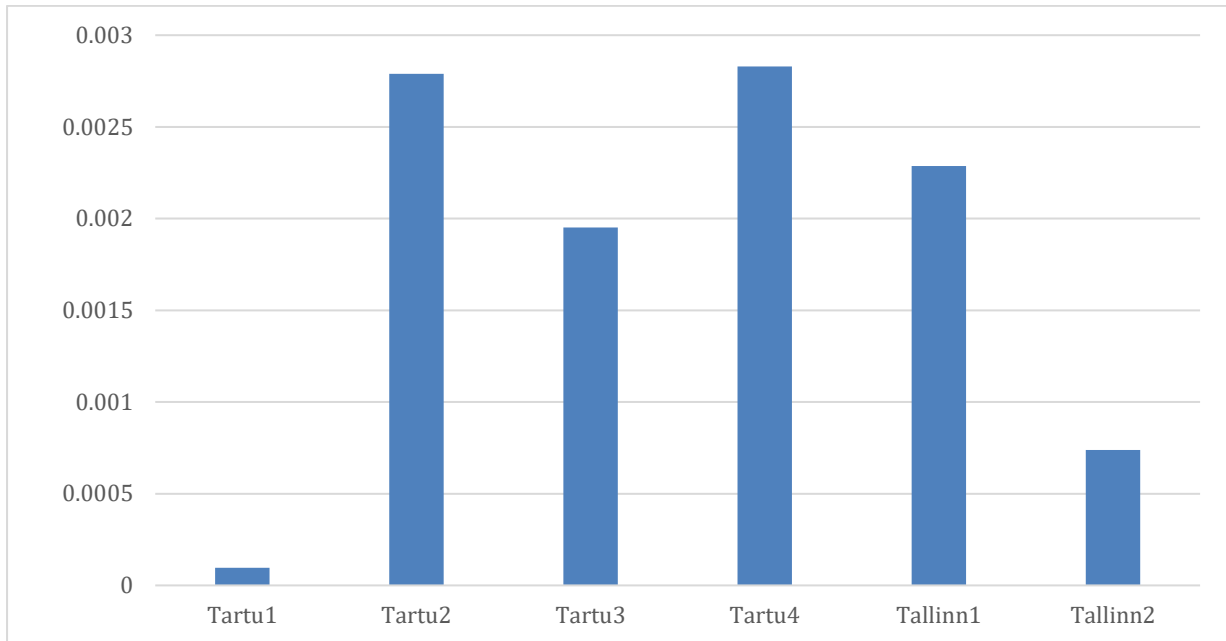
Joonis 7. Kattuvad andmed.

Samuti leidis testimise käigus autor, et kasutades kiiruseid, mis on suuremad kui 6x tekivad ekraanipiltidel defektid ning need ei jõua serverisse õiges järjekorras. Testimisel võrdles autor 1x kiiruse peal rakenduse genereeritud andmeid teiste kiiruste peal loodud andmetega. Selleks vaatas autor visuaalselt läbi kõik 1x andmed ning kasutas võrdlemiseks Pythoni teeki ImageHash [22]. Autor pakub välja, et see juhtub, kuna ekraanipiltide tegemiseks kasutatakse html2canvas teeki, mis ei suuda luua pilte konstantse ajaga. Teisisõnu erinevatel simulatsiooni hetkedel kulub erinev aeg, et salvestada kasutatavast Mapbox GL JS teegi kaardi lõuendist pilt ning rakenduse kiirust suurendades saavad mõned varasemad pildid hiljem valmis kui hilisemad. Esimene võimalik lahendus sellele probleemile oleks luua tõhusam viis brauserist ekraanipiltide salvestamiseks. Praegu paremat võimalust, kui käesolevas töös kasutatav viis, selleks pole [23]. Teine variant oleks ehitada rakendus mitte veebirakendusena, vaid töölaarakendusena.

## 4.2 Täpsus

Kuna antud rakendus saab GPS-koordinaadid Mapboxi liidesest, siis peab kontrollima kattuvust etteantud testandmetega. Eelnevalt on välja toodud, et kõige mõistlikum on rakendust

kasutada kiirusel 6x. Seetõttu on antud täpsus arvutatud vaid rakenduse 6x kiirusel genereeritud andmetel.



Joonis 8. Algandmete ja rakenduse väljundi kokkulangevus.

Joonisel 8 on näha algandmete ja rakenduse väljund GPS-koordinaatide kokkulangevus. Joonise X-teljel on kasutatud CSV andmefail ning Y-teljel keskmine erinevus meetrites. Täpsuse hindamiseks leitakse igale rakenduse poolt väljastatud punktile lähim punkt algandmetest. Punktide omavahelise kauguse arvutamiseks on kasutatud Haversine kauguse arvutamise valemit [24]. Keskmine kaugus leitakse liites kõik punktide omavahelised kaugused ja jagades nende arvuga. Kõige paremad tulemused saadi algandmetel nimega Tartu1, kus keskmiselt erines algandmete ja rakenduse väljundi punktide kaugus vaid 0.0000962 meetrit. Kõikide testitud algandmete peale on keskmiselt täpsus parem kui 0.003 meetrit. Kuna rakenduse loodud andmeid kasutatakse tulevikus isejuhtiva auto navigeerima õpetamisel, siis saadud täpsus võiks anda häid tulemusi kuid enne närvivõrgul treenimata seda teada ei saa.

### 4.3 Edasiarenduse võimalused

Loodud rakendus täidab oma eesmärgi, kuid siiski on sellel mitmeid puudusi ning edasiarenduse võimalusi on mitmeid. Järgnevalt tuuakse välja mõned:

- Uuendada aluseks olevat kaardi versiooni – baasiks võetud guidance-sim kasutab 5 aastat vana versiooni kaardi renderdamise mootorit. Uuel versioonil on lisatud

kasulikke võimalusi, näiteks saaks kaameranurka muuta rohkem kui praegusel versioonil.

- Luua oma liides teekonnatrajektooride küsimiseks – Mapbox Directions API kasutab aluseks OSRM [25] liidest, kuid piirab päringud 25 teekonnapunktini. Võimalus on rajada lokaalne OSRM server, et luua teekonnatrajektoore, niiviisi oleks võimalik kasutada rohkem kui 25 teekonnapunkti. Teine variant oleks algandmetes määrata, kus sõiduk teeb manöövri ja kasutada ainult nende asukohtade koordinaate. Sellel variandil saaks kasutada vaid 25 manöövriga algandmeid kui ei tehta samasugust töötlemist päringute vastustega nagu hetkel rakendus teeb.
- Luua täiesti nullist uus rakendus – loodud programm kasutab aluseks Mapboxi loodud guidance-sim rakendust, mille kood on 5 aastat vana ning väga keeruline ja segane. Uue rakenduse loomine teeks ilmselt edasiarendamise võimalusi ka lihtsamaks. Lisaks moodustab praegune rakendus 1.4 MB suuruse Javascript faili iga kord kui seda käivitada. Luues uue rakenduse, on kindlasti võimalik see ülesehitada nii, et pole vaja moodustada Javascript faili igal käivituskorral.
- Luua töölauarakendus – praegune programm koosneb kliendi ja serveri poolest ning on seetõttu vähem tõhus kui töölauarakendus oleks. Rakendus, mis töötaks lokaalselt arvutil, võiks saavutada kiirema töökiiruse ehk läbida simulatsioone kiiremini, kui hetkene veebirakendus seda võimaldab.

Kindlasti üks kasulikumaid edasiarenduse suundi oleks luua oma liides teekonnatrajektooride küsimiseks. Nii saaks olla kindel, et simulatsiooni loodud ekraanipildid lähevad kokku isejuhtiva auto andmetega ning sel juhul võib tulevikus närvivõrgu treenimisel saavutada paremaid tulemusi.

Antud valdkonna üks olulisi edasiarenduse suundi oleks ka *online* versiooni loomine valminud rakendusest, mida kasutatakse siis, kui närvivõrk on juba treenitud *offline* versiooni poolt genereeritud andmetel. *Online* versioon annaks võimaluse sõitmisel isejuhtivale autole navigatsioonikäsklusi edastada ning võimaldaks kasutada treenitud närvivõrku. Sõitmisel on kindlasti tähtis, et navigatsioonikäsklusi edastataks närvivõrgule samal viisil nagu neid edastati treenimise puhul ning seetõttu oleks mõistlik põhjaks kasutada antud bakalaureusetöö raames loodud rakendust.

## 5 Kokkuvõte

Bakalaureusetöö eesmärgiks oli luua navigatsioonirakendus, mille ekraanipilti saaks ette anda närvivõrgule koos tekstiliste ja struktuursete navigatsioonikäsklustega. Töö raames kasutati vigast olemasolevat lahendust, mis tehti korda ja millele lisati külge mitmeid funktsionaalsusi. Lisatud funktsionaalsused on järgnevad:

- Windowsi peal rakenduse toimima panek
- Isejuhtiva sõiduki CSV-faili lugemine ning töötlemine
- Teekonnatrajektoori küsimine veebiliidesest
- Ekraanipiltide salvestamine koos navigatsioonikäsklustega ning GPS-koordinaatidega sünkroonimine
- Salvestatud andmete kokkupakkimine

Valminud rakendus täidab bakalaureusetöö töö eesmärgi ning võimaldab luua andmeid isejuhtiva auto närvivõrgu treenimiseks. Tulemuste analüüsil näidati, et loodud lahendus toimib kiiremini, kui reaajas andmete loomine. Samuti näidati loodud rakenduse täpsust kasutatavate testandmete peal, mis oli keskmiselt parem kui 0.003 m. Siiski on rakendusel mitu edasiarendamise võimalust, millest peamine on loodud andmete täpsemaks muutmise, luues parema liidese teekonnatrajektooride genereerimiseks.

## Viidatud kirjandus

- [1] Matiisen T., Modulaarne lähenemine, Tehisintellekti Algkursus, 2020, [https://courses.cs.ut.ee/2020/Tehisintellekti\\_algkursus/spring/Main/PARTIVmodulaarne](https://courses.cs.ut.ee/2020/Tehisintellekti_algkursus/spring/Main/PARTIVmodulaarne) (15.01.2021)
- [2] Matiisen T., Täielikult närvivõrkudel põhinev lähenemine, Tehisintellekti Algkursus, 2020, [https://courses.cs.ut.ee/2020/Tehisintellekti\\_algkursus/spring/Main/PARTIVtehis](https://courses.cs.ut.ee/2020/Tehisintellekti_algkursus/spring/Main/PARTIVtehis) (15.01.2021)
- [3] Codevilla F., Müller M., López A., Koltun V., Dosovitskiy A. End-to-end Driving via Conditional Imitation Learning, *arXiv*, 2018, <https://arxiv.org/abs/1710.02410> (15.02.2021)
- [4] Hecker S., Dai D, Van Gool L., End-to-End Learning of Driving Models with Surround-View Cameras and Route Planners, *arXiv*, 2018, <https://arxiv.org/abs/1803.10158> (17.12.2020)
- [5] Hawke J. *et al*, Urban Driving with Conditional Imitation Learning, *arXiv*, 2019, <https://arxiv.org/abs/1912.00177> (16.12.2020)
- [6] Bing Maps. <https://www.bing.com/maps> (02.05.2021)
- [7] Google Maps. <https://www.google.com/maps> (21.04.2021)
- [8] NavigatorFREE. <https://navigatorfree.mapfactor.com/en/>
- [9] TomTom Go Discover. [https://www.tomtom.com/en\\_ee/navigation/promo/go-discover/](https://www.tomtom.com/en_ee/navigation/promo/go-discover/) (02.05.2021)
- [10] QT Map Viewer. <https://doc.qt.io/qt-5/qtlocation-mapviewer-example.html> (05.05.2021)
- [11] Mapbox GL JS. <https://docs.mapbox.com/mapbox-gl-js/api/> (18.04.2021)
- [12] Mapbox Directions API. <https://docs.mapbox.com/api/navigation/directions/> (18.04.2021)
- [13] Google Maps Directions API. <https://developers.google.com/maps/documentation/directions/overview> (02.05.2021)
- [14] Mapbox guidance-sim. <https://github.com/mapbox/guidance-sim> (02.05.2021)
- [15] Android Screen Recorder. <https://support.google.com/android/answer/9075928?hl=en> (21.04.2021)

- [16] Visual Studio Code. <https://code.visualstudio.com/> (20.04.2021)
- [17] Google Chrome. <https://www.google.com/intl/et/chrome/> (20.04.2021)
- [18] GlobalStats Statcounter. <https://gs.statcounter.com/browser-market-share> (20.04.2021)
- [19] Mozilla. [https://developer.mozilla.org/en-US/docs/Web/API/Screen\\_Capture\\_API/Using\\_Screen\\_Capture](https://developer.mozilla.org/en-US/docs/Web/API/Screen_Capture_API/Using_Screen_Capture) (06.05.2021)
- [20] html2canvas. <https://html2canvas.hertzen.com/> (06.05.2021)
- [21] OpenStreetMap. <https://www.openstreetmap.org/about/> (18.04.2021)
- [22] ImageHash. <https://pypi.org/project/ImageHash/> (07.05.2021)
- [23] Hackernoon. <https://hackernoon.com/how-to-take-screenshots-in-the-browser-using-javascript-192k3xq7> (05.05.2021)
- [24] Haversine formula. <https://www.movable-type.co.uk/scripts/latlong.html> (05.05.2021)
- [25] OSRM. <http://project-osrm.org/docs/v5.24.0/api/#> (05.05.2021)

## **Lisad**

### **I. Loodud navigatsioonirakenduse lähtekood**

Lähtekoodi saamiseks tuleb pöörduda autori poole. Email: [makserik@hotmail.com](mailto:makserik@hotmail.com)

# Litsents

## **Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks**

Mina, Erik Marcus Maks,

1. Annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose, Navigatsioonirakenduse sisendi renderdamine autot juhtivale närvivõrgule, mille juhendaja on Tambet Matiisen, 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.

*Erik Marcus Maks*

**07.05.2021**