

Tartu Ülikool

Loodus- ja täppiseaduste valdkond

Tehnoloogiainstituut

Kaarel Mark

# **Liitreaalsus tootmisprotsessis asukohtade määramisel**

Bakalaureusetöö (12 EAP)

Arvutitehnika eriala

Juhendaja:

Robotika dotsent Karl Kruusamäe

Tartu 2019

# Resümee / Abstract

## **Liitreaalsuse kasutamine tootmisprotsessis asukohtade määramisel**

Liitreaalsus (ingl *augmented reality*) leiab nii igapäevaelus kui tööstuses üha enam rakendust. Tööstuses kasutatakse liitreaalsust peamiselt üldiste juhiste andmiseks. Seetõttu oleks järgmine samm luua süsteem, mille abil saavad töölised määrata täpseid asukohti detailidel liitreaalsusvisiiri abil. Käesoleva bakalaureusetöö eesmärgiks on luua lahendus, millega on võimalik LR-visiiri (ingl *augmented reality headset*) abil moodus näiteks puuravade asukohtade määramiseks tootmisdetailidel. Antud töö raames loodud lahendus on avatud lähtekoodiga ja ROSi ning simulatsioonirakenduse Gazebo-põhine. Selline platvorm ja avatud lähtekood võimaldavad edasiarendust ning muuotstarbelist kasutust. Töö raames valminud süsteemi täpsust hinnati mitmes erinevas asetusel, mediaannihkeks reaalsuse ja virtuaalsuse vahel registreeriti 25.8 mm.

**CERCS:** T125 Automatiseerimine, robotika, control engineering, T130 Tootmistehnoloogia

**Märksõnad:** liitreaalsus, avatud lähtekood, ROS, Gazebo

## **Augmented reality for location determination in manufacturing**

The use of augmented reality in both everyday life and industry is constantly growing. In industry augmented reality is commonly used for giving general guidelines. Therefore the next step would be to develop a system with which workers could determine accurate locations with the assistance of augmented reality headsets. The purpose of this bachelor's thesis is to create a system that could, using an augmented reality headset, be used to determine locations in manufacturing (e.g. drill holes). The system created in this thesis is open-source and based on ROS (Robot Operating System) and Gazebo simulator. Open-source platforms provide further development opportunities and applying augmented reality for other purposes. The accuracy of the system was tested in different setups, the median discrepancy between the reality and virtuality was recorded at 25.8 mm.

**CERCS:** T125 Automation, robotics, control engineering, T130 Production technology

**Keywords:** augmented reality, open-source, ROS, Gazebo

# Sisukord

<b>Resümee / Abstract</b>	<b>2</b>
<b>Sisukord</b>	<b>3</b>
<b>Jooniste loetelu</b>	<b>5</b>
<b>Tabelite loetelu</b>	<b>6</b>
<b>Lühendid, konsonandid, mõisted</b>	<b>7</b>
<b>1 Sissejuhatus</b>	<b>8</b>
<b>2 Kirjanduse ülevaade</b>	<b>9</b>
2.1 Liitreaalsuse kasutusviisid tööstuses	11
2.2 Turul saadaolevad LR töövahendid	13
2.2.1 Microsoft HoloLens 2	14
2.2.2 Google Glass Enterprise Edition	14
2.2.4 DAQRI Smart Helmet	15
2.3 Robot Operating System (ROS)	16
2.4 Gazebo	16
2.5 Olemasolevad liitreaalsuslahendused ROSis	17
<b>3 Nõuded</b>	<b>19</b>
3.1 Töö eesmärk	19
3.2 Funktsionaalsed nõuded	19
3.3 Riistvaralised nõuded	19
3.4 Tarkvaralised nõuded	19
<b>4. Disain</b>	<b>20</b>
4.1. Tööpõhimõte	20
4.2 Arhitektuur	21
4.3 Kasutatud kimbud ja konfiguratsioon	22
<b>5 Karakteriseerimine</b>	<b>25</b>
5.1 Metoodika	25
5.2 Süsteemi üldine täpsus	27
5.3 Täpsuse sõltuvus LR-märgiste arvust	28

5.4 Täpsuse sõltuvus vaatlemiskaugusest	29
5.5 Täpsuse sõltuvus vaatlemissuunast	30
5.6 Täpsuse sõltuvus märgiste asukohast kaamerapildis	31
5.7 Järeldused	32
<b>6. Kokkuvõte</b>	<b>33</b>
<b>Viited</b>	<b>34</b>
<b>Lisad</b>	<b>37</b>
Lisa 1. Mõõtmiste toorandmed	37
<b>Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks</b>	<b>38</b>

# Jooniste loetelu

<b>Joonis 1.</b> <i>Reaalsuse ja virtuaalsuse kontiinum [3].</i> .....	9
<b>Joonis 2.</b> <i>Pokemon GO [7].</i> .....	10
<b>Joonis 3.</b> <i>Projektoripõhine liitreaalsuslahendus Austraalia General Motorsi tehases [15].</i> ..	11
<b>Joonis 4.</b> <i>ROSi, riistvara ja simulaatori omavaheline suhtlus.</i> .....	17
<b>Joonis 5.</b> <i>LR-süsteemi ülevaade. Vaade tahvlile läbi LR-visiiri (vasakul) ning algne tahvel pärismaailmas (paremal).</i> .....	20
<b>Joonis 6.</b> <i>LR-süsteemi voodiagramm ja ülesannete jaotus kahe arvuti vahel.</i> .....	21
<b>Joonis 7.</b> <i>Teisenduspuu, roheline noolega märgitud teisendused on eeldefineeritud, punase joonega teisendus arvutatakse kaamerapildist.</i> .....	22
<b>Joonis 8.</b> <i>Testkeskkond koordinaatteisenduste vaatepunktist.</i> .....	22
<b>Joonis 9.</b> <i>Süsteemi ülesehitus ROSi rubriikide ja sõlmede kujul, rohelisega märgitud sõlmed asuvad serverarvutis, sinisega märgitud sõlmed LR-visiiris.</i> .....	23
<b>Joonis 10.</b> <i>Sõlme gazebo_ar töövoog.</i> .....	24
<b>Joonis 11.</b> <i>Mõõtmisprotsess. Virtuaalse kujutise nihe on <math>(x'-x, y'-y)</math>.</i> .....	25
<b>Joonis 12.</b> <i>Testkeskkonna ülesehitus, roheliselt on märgitud hiirevajutusasukohad virtuaalsel ruudustikul.</i> .....	26
<b>Joonis 13.</b> <i>Sooritatud mõõtmised terve katseala kontekstis</i> .....	28
<b>Joonis 14.</b> <i>Kõigi sooritatud mõõtmiste täpsus, sinise ringi sees on 50%, rohelistes ringis 70% ning punases ringis 90% mõõtmistulemustest.</i> .....	28
<b>Joonis 15.</b> <i>Süsteemi täpsus ühe (a), kahe (b) ja kolme märgisega (c).</i> .....	29
<b>Joonis 16.</b> <i>Süsteemi täpsus kahe (a), kolme (b) ja nelja meetri kauguselt (c).</i> .....	30
<b>Joonis 17.</b> <i>Süsteemi täpsus vaadeldes vasakult (a), keskelt (b) ja paremalt (c).</i> .....	31
<b>Joonis 18.</b> <i>Süsteemi täpsus kui märgised on kaamerapildi keskel (a) ja kaamerapildi servas (b).</i> .....	32

# Tabelite loetelu

<b>Tabel 1.</b> <i>LR töövahendite võrdlus.</i> .....	14
<b>Tabel 2.</b> <i>Mõõtmistulemused erineva märgiste arvuga.</i> .....	28
<b>Tabel 3.</b> <i>Mõõtmistulemused erinevatelt kaugustelt.</i> .....	29
<b>Tabel 4.</b> <i>Mõõtmistulemused erinevatest suundadest.</i> .....	30
<b>Tabel 5.</b> <i>Mõõtmistulemused erineva märgiste paigutusega kaamerapildis.</i> .....	31

# Lühendid, konstandid, mõisted

**API** – (*Application Programming Interface*) ehk rakendusliides võimaldab suhtlust erinevate tarkvarakomponentide vahel kasutades eeldefineeritud eeskirju.

**ARAUM** – (*Augmented Reality Authoring for Maintenance*) on hooldustoe pakkumiseks loodud liitreaalsussüsteem.

**IMS Lab** – (*Intelligent Materials and Systems Lab*) ehk Tartu Ülikooli Tehnoloogiainstituudi arukate materjalide ja seadmete labor.

**IMU** – (*Inertial Measurement Unit*) ehk inertsiaalandur mõõtemuundur, mille tajurid mõõdavad joon- ja pöörlemiskiirust.

**LR** – Liitreaalsus (*Augmented Reality*) on segareaalsuse vorm, kus pärismaailma lisatakse virtuaalseid elemente.

**LV** – Liitvirtuaalsus (*Augmented Virtuality*) on segareaalsuse vorm, kus virtuaalsesse maailma lisatakse pärismaailma elemente.

**OS** – (*Operation System*) on programmide kogum, mis käivitatakse arvutis algladimisprogrammi poolt ning mis juhib arvutisüsteemi tööd ja teenindab rakendusprogramme.

**ROS** – (*Robot Operating System*) on tarkvararaamistik robotite kasutamiseks, integreerimiseks ja arendamiseks

**SDK** – (*Software Development Kit*) ehk arendustarkvara on kogum erinevatest tööriistadest, mille abil luuakse tarkvara kindla platvormi jaoks.

**SR** – Segareaalsus (*Mixed Reality*) on virtuaalse ja pärismaailma kooslus.

**UCTN** – (*Technical University of Cluj-Napoca*) ehk Cluj-Napoca tehnikaülikool Rumeenias.

**UT** – (*University of Tartu*) ehk Tartu Ülikool.

**VR** – Virtuaalreaalsus (*Virtual Reality*) on täielikult virtuaalne tajutav keskkond.

# 1 Sissejuhatus

Alates tööstusrevolutsioonist on tootmisprotsesse üha enam optimeeritud. Tänapäeval leiab suures mahus aset automatiseerimine, kuid on töid, mida masinad inimestest efektiivsemalt teha ei suuda. Tootmismasinad on tihti loodud mingi kindla ülesande täitmiseks. Kui on vaja toota väikeses mahus erinevaid detaile, pole otstarbekas kulutada raha mitme masina peale. Inimesed on seevastu kohanemisvõimelised ning suudavad õppida valmistama erinevaid detaile [1]. Seega ei peaks piirduma vaid masintootmise arendamisega, inimeste tööprotsessi optimeerimine on samuti oluline.

Tootmisdetailidel täpsete asukohtade määramisel oleks abiks liitreaalsus (LR). Peaskantavate liitreaalsusprillide või muude peakomplektide ehk LR-visiiride hulk üha kasvab, tundumad lahendused on Google Glass, Microsoft HoloLens, Vuzix Blade ja Eyesight Raptor [2]. Lisaks kasutusele igapäevaelus rakenduste visualiseerimisliidesena, saab liitreaalsust rakendada ka tootmisprotsessides. Suurte või keeruliste detailide puhul on tööisel keeruline leida pidepunkti, millest täpselt mõõta näiteks puuritava ava asukoht. Liitreaalsust kasutades on tarvis vaid seda, et vaateväljas asuks märgis, mille järgi süsteem suudab tuvastada, kus detail asub ning seeläbi saab sinna lisada virtuaalseid tähiseid.

Käesoleva bakalaureusetöö eesmärgiks on luua süsteem, mis ROSi (*Robot Operating System*) ja simulatsioonirakenduse Gazebo abiga kuvab läbi LR-visiiri virtuaalseid kujutisi, mille abil saab määrata täpseid asukohti pärismaailmas. Töös kirjeldatakse olemasolevaid LR-põhiseid tootmislahendusi (peatükk 2.1), erinevaid LR-seadmeid (peatükk 2.2), kasutatud töövahendeid (peatükid 2.3-2.4) ja olemasolevaid ROSi-põhiseid LR-süsteeme (peatükk 2.5). Kolmandas peatükis täpsustatakse töö eesmärk ning seatakse tehnilised nõuded. Neljandas peatükis kirjeldatakse loodud süsteemi ülesehitus ning viiendas peatükis hinnatakse valminud süsteemi täpsust ehk nihet reaalsuse ja virtuaalsuse vahel.

## 2 Kirjanduse ülevaade

Virtuaalreaalsus (VR) on interaktiivne keskkond, milles vaatleja tajub ning ning kus vaatleja saab mõjutada täielikult sünteetilist maailma [3]. Segarealsus (SR) on viis virtuaal- ja pärismaailma ühendamiseks ühte keskkonda (joonis 1). SR sisaldab endas nii virtuaalse maailma kui pärismaailma elemente, seega saab segarealsust omakorda jaotada elementide osakaalu järgi. Liitreaalsus (LR) on süsteem, kus pärismaailma lisatakse virtuaalseid elemente, liitvirtuaalsuse (LV) puhul on enamust maailmast arvuti poolt loodud, lisatud on mõned pärismaailma elemendid.



**Joonis 1.** *Reaalsuse ja virtuaalsuse kontiinum* [3].

SR-süsteemide arendamisel on kõigi meelte hulgast enim kasutuses nägemine, kuulmine ja kompimine. Kompimismeelt kasutatakse virtuaalse keskkonnaga suhtlemiseks igapäevaselt arvutihiire näol, kuid sellisel juhul on suhtlus ühepoolne. Mis teeb seadme liitreaalsusseadme on mõlemapoolne suhtlus, seega on tänapäevaste mängukonsoolide puldid tihti liitreaalsusseadmed, sest puldi abil saab juhtida mängus toimuvat ning mängus toimuv võib panna puldi värisema [4]. Näide kuulmispõhisest liitreaalsusseadme on Nura kõrvaklapid, mis mõõdavad inimese kõrva poolt tuvastatud heli tugevust ning vastavalt sellele reguleerivad sissetuleva heli tugevust tervisliku piirini [5].

Levinum LR lahendus on visuaalne, kus kaamerapildile või prilliklaasidele lisatakse visuaalseid kujutisi. Pikalt App Store edetabelite tipus olnud Pokemon GO [6] on ilmekas näide sellest, kuidas liitreaalsust saab edukalt kasutada rakenduse visualiseerimisel. Mängu eesmärk on käia pärismaailmas ringi ning püüda kinni Pokemone, Jaapani konsoolimängudest pärinevaid olendeid. Kui mängija jõuab pärismaailmas õigesse asukohta, ilmub Pokemon

tema telefoni kaamerapildile (joonis 2), sealjuures näib, nagu olend seisaks maa peal inimese ees [7].



**Joonis 2.** *Pokemon GO* [7].

Visuaalsete objektide asetamiseks pärismaailma on esmalt vaja informatsiooni ümbruskonna kohta ja tuleb tuvastada vaatleja liikumist pärismaailmas kuue vabadusastme kujul: kolm asukohamuutujat ( $x, y, z$ ) ning kolm orientatsioonimuutujat ( $roll, pitch, yaw$ ).

Informatsiooni saamiseks ümbruskonna kohta kaamerapildi abil on vaja defineerida ankur – koht pärismaailmas, mille järgi tuvastada või defineerida vaatleja asukoht pärismaailmas. Selleks võib olla nii kindel koht kaamerapildis kui spetsiaalne märgis. Kui ankruks on märgis, siis selle mõõtmeid teades saab välja arvutada kaamera täpse asukoha süsteemis [8]. Märgisevaba liitreaalsuslahenduse puhul pole virtuaalse kujutise asukoht eelnevalt defineeritud, vaid see määratakse kas kasutaja poolt või keskkonna põhjal. Näiteks on loodud lahendus arheoloogilistel varemetel, kus keskkonda ei saa LR-märgistega rikkuda [9]. Andmebaasi on salvestatud mitu erinevat kaadrit varemetest ning kaamera asukoht leitakse võrreldes kaamerapildi salvestatud kaadritega. Lisaks saab vaatleja asukohta pärismaailmas määrata ka tema üldise asukoha põhjal näiteks GPSi abil. Näide sellest on mobiilirakendus

ARCity, mis manab kaamerapildile tänavanimetusi ning muud orienteerumiseks vajalikku informatsiooni leides kasutaja asukoha GPSi abil [10].

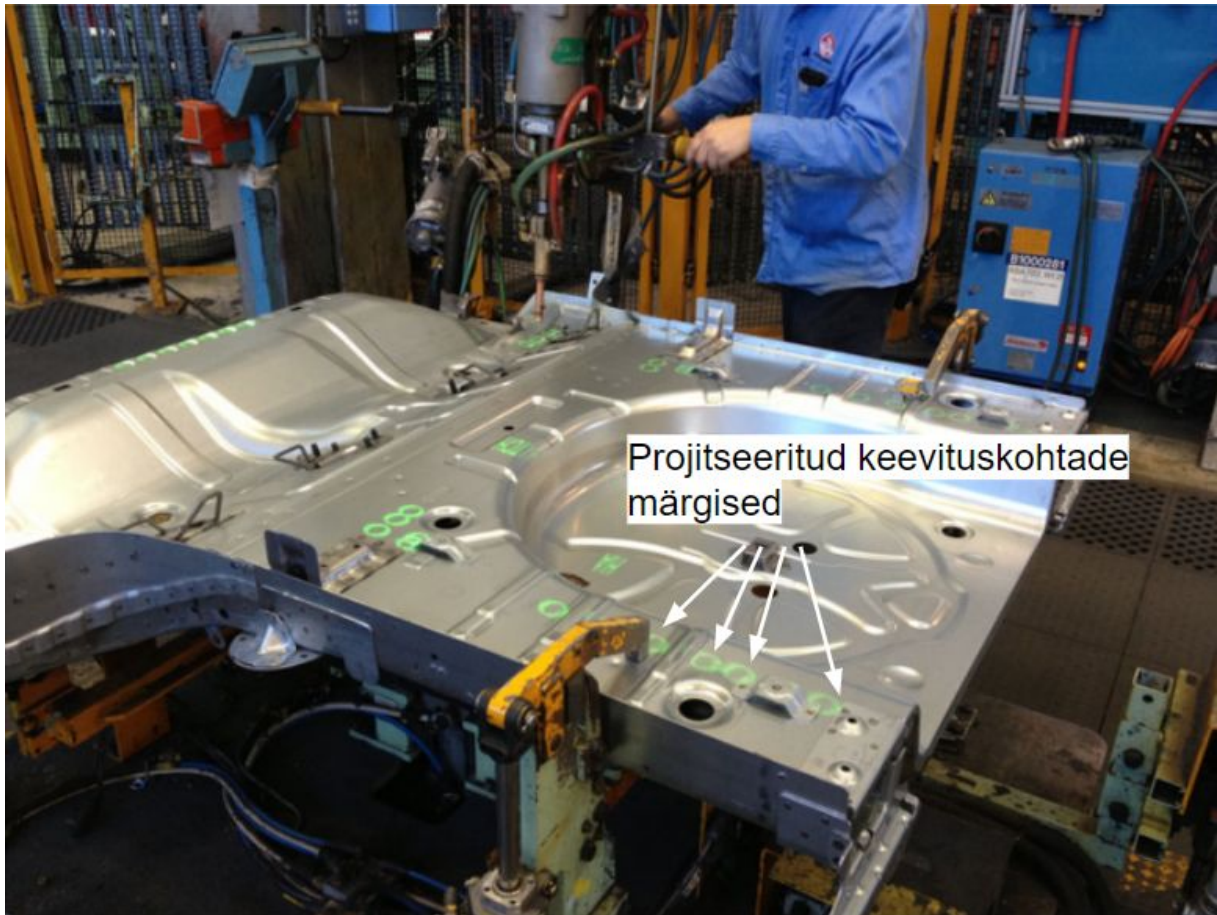
Vaatleja liikumisel pärismaailmas peab kaasa liikuma ka virtuaalne maailm. Liikumise tuvastamiseks kasutatakse peamiselt kaamerat ja inertsiaalاندurit [11]. Kaamerapildist saab tuvastada liikumist võrreldes kaadritevahelisi muutuseid. Inertsiaalاندur (IMU) koosneb tavaliselt kiirendusmõõturist, güroskoobist ning magnetomeetrist, igähte neist on kolm iga suuna kohta [12]. Kiirendusmõõtur mõõdab liikumist telje sihis, güroskoop ümber telje. Güroskoobi lugem triivib kui seade seisab paigal, sest integreeritakse müra. Maaga ristuva telje ümber tekkiva triivi filtreerimiseks on magnetomeeter, mis mõõdab sarnaselt güroskoobile orientatsiooni. Kahe tulemuse võrdlemisel saadakse täpsem tulemus kui ühe mõõteseadme kasutamisel.

## **2.1 Liitreaalsuse kasutusviisid tööstuses**

Liitreaalsus on tööstuses juba mitmel moel kasutuses. Peamiselt on liitreaalsus abiks komponentide paigutamisel õigesse kohta, manades töötaja silme ette kujutise sellest, kuidas toode välja peaks nägema [13]. Näiteks kasutatakse PTC poolt arendatud liitreaalsuslahendust akude koostamisel firmas BAE Systems [14]. Tööjuhised kuvatakse töötajale liitreaalsusprillide abil. Liitreaalsusprillide klaasid on läbipaistvad, kuid sinna saab manada graafilisi juhendeid, näiteks aku sammhaaval komplekteerimiseks.

Liitreaalsust ei pea alati rakendama LR-visiiri abil. Austraalias General Motorsi tehases on katsetatud projektoripõhist liitreaalsust punktkeevituste tegemisel [15]. Keevitatav komponent asetatakse lauale ja kahe projektori abil projitseeritakse vastava keevitustüübi tähis komponendile (joonis 3). Kahte projektorit on vaja juhaks, kui tööline ühe projitseeritava pildi oma kehaga ära varjab. Komponenti asend määratakse LR-märgise ja kaamera abil.

Projektoripõhine süsteem on lihtne ja seeläbi ka töökindel, kuid samas piiratud. Näiteks ei saa projitseerida keevituse asukohta komponendi kitsamale tahule. LR-visiiriga saaks liikuda ümber detaili ja keevituskohti saaks kujutada mistahes punkti osisel. Samas piirab peas kantav seade töölise tähelepanu muu ümbritseva suhtes. Tehases, kus on palju tööjaamu ja liiklust võib selline asjaolu osutada ohtlikuks.



**Joonis 3.** Projektoripõhine liitreaalsuslahendus Austraalia General Motorsi tehases [15].

Mitmekülgsemalt on liitreaalsust kasutatud näiteks auto telje monteerimisprotsessis. Auto teljesüsteem on kaalult raske, seega on inimesel raske seda käsitsi kokku panna. Selle jaoks on olemas robotkäsi, mis teljesüsteemi üleval hoiab. Peas kantava seadme ja liitreaalsuse abil luuakse liides töölise ning robotkäe vahel, kus tööline saab vaadata tootmisandmeid, robotkäe tööala ning komponentide 3D-mudeleid. Lisaks komponentide paigutusele kuvatakse LR-visiirile koostejuhend. Asendades mistahes trükitud juhendid, võimaldatakse töölisel kasutada töö tegemisel mõlemat kätt ning töölise keskendumisvõime on vähem häiritud, sest pilku pole vaja pöörata. Robotkäe tööala nägemine on oluline turvalisuse vaatepunktist. Töölisele kuvatakse värviliselt ala, kus robotkäsi tegutseb ja kust tööline peaks eemale hoidma [16].

Veel üks viis, kuidas liitreaalsust tööstuses kasutada, on kaasaskantav nutiseade. Suunates seadme tagaküljel oleva kaamera tööalale, saab seadme ekraanile kuvada nimetatud tööjaama infot. Iga tööjaama juures on erinev masinloetav märgis, mille abil identifitseeritakse tööjaam.

Seejärel saadab seade andmebaasi, kus asuvad andmed kõikide tööjaamade kohta, päringu. Nii saab tehase tööjaamadelt andmeid näiteks graafikute näol, mis võimaldavad kujutada andmeid mitmekülgsemalt kui harilikud näidikud [17].

Liitreaalsust saab kasutada ka hooldustööde teostamisel näidates, kuidas juba kokku pandud seade peaks välja nägema. Sellise eesmärgiga on näiteks loodud tarkvaralahendus ARAUM (*Augmented Reality Authoring for Maintenance*), mis aitab liitreaalsuse abil sammhaaval kontrollida seadme korrasolekut [18]. Kõnealusel süsteemis kasutatakse märgisevaba objektituvastust Vuforia SDK abil [19]. Sarnaselt teistele lahendustele saab kasutajale kuvada seadme detailide kujutisi. Lisaks sellele on võimalik koguda andmeid erinevatelt sensoritelt ning lisada neid kuvapildile. Sellisest abivahendist oleks lisaks professionaalidele kasu ka tavakasutajatele, kes oma seadet ise parandada soovivad.

Liitreaalsuse kasutamise eeliseid paberjuhendi ees on testitud Honda CBR 600 mootorratta mootori hooldamise näitel [20]. Fiorentino jt uuringus näidati katsealustele juhiseid mootori hooldamiseks suurel ekraanil. Liitreaalsuskujutis loodi statsionaarse kaamera ja markerite abil. Katse käigus mõõdeti kolme parameetrit: veaprotsenti, kulunud aega ja kasutaja rahulolu, kõigis kategooriates osutus liitreaalsus paremaks kui traditsiooniline paberjuhend. Samuti osutus liitreaalsus kasutajale sobivamaks töövahendiks, olles intuitiivsem, lihtsam kui ka üleüldiselt meeldivam abivahend. Suuremat kasu näitas liitreaalsus just komponentide äratundmise ja õige tööriista valimise juures, kuid ka protseduuri käigu juhendamisel oli liitreaalsusest rohkem kasu.

## **2.2 Turul saadaolevad LR töövahendid**

Kui LR-visiirid esmakordselt turule tulid, ei leitud neile kohe eristatavat kasutusvaldkonda. Aastate jooksul on aga neid tekkinud: meditsiiniline väljaõpe, toodete visualiseerimine ja palju muud [21]. Sellega seoses on kasvanud müügis olevate LR-visiiride hulk. Tuntumad neist on näiteks Google Glass ja Microsoft HoloLens (tabel 1).

**Tabel 1.** LR töövahendite võrdlus.

	<b>Microsoft HoloLens 2</b>	<b>Google Glass Enterprise Edition</b>	<b>Daqri Smart Helmet</b>	<b>Daqri Smart Glasses</b>
<b>Hind</b>	€3098 [22]	€1550 [23]	pole müügis, viimati €13337 [24]	€4454 [25]
<b>Operatsioonisüsteem</b>	Windows Holographic Operating System [26]	Glass OS Androidi baasil [27]	Ubuntu/Daqri OS [28]	Ubuntu/Daqri OS [29]
<b>Diagonaalne vaateväli</b>	>70 kraadi [26]	-	44 kraadi [30]	44 kraadi [25]

### 2.2.1 Microsoft HoloLens 2

Microsoft HoloLens on Microsofti poolt loodud LR-visiir. HoloLens 2 seade spetsifikatsioon avalikustati 24. veebruaril 2019 ning seda on võimalik ette tellida \$3500 eest [22].

Sensoreid on HoloLens 2 peal mitmeid: ümbruskonna tajumiseks 4 madala resolutsiooniga kaamerat, üks 8 MP kaamera ja sügavusandur. Lisaks on liikumise tuvastamiseks olemas IMU ja silmade liikumise tuvastamiseks 2 infrapunakaamerat. Seadme kasutamine käib läbi käe- ja silmaliigutuste. Protsessor on Microsofti enda poolt arendatud teise põlvkonna holograafiline protsessor. Muutmälu on seadmel 4 GB, andmekandjaks on 64-GB UFS 2.1 standardi pooljuhtketas. Seadmeid saab ühendada läbi USB-C liidese ja seadmel on nii WiFi kui Bluetooth võimekus. Operatsioonisüsteemiks on prillidel Windows Holographic Operating System, rakenduste haldamiseks ja allalaadimiseks on Dynamics 365 [26].

### 2.2.2 Google Glass Enterprise Edition

Google poolt arendatud nutiprillide Google Glass müük algas 2014. aasta mais [31]. Paraku oli Google Glass liitreaalsusprillide valdkonnas üks esimesi, seega polnud prillidele palju kasutusvaldkondi. Ajapikku hakkas aga seade leidma kasutust tööstuses, sellest tulenevalt lasti 2017. aastal välja Google Glass Enterprise Edition, mis on spetsifikatsioonide poolest suures osas sama, mis Google Glass [23]. Tänapäeval leiab seade kasutust paljudes firmades,

kaasaarvatud DHL, Samsung ja Volkswagen [32]. Ametlike teadete kohaselt on arenduses seadme uuendatud versioon Google Glass Enterprise Edition 2 [33].

Ümbruskonna tajumiseks on Google Glass Enterprise Edition seadmel valgussensor, IMU, GPS ja 8 MP kaamera. Muutmälu hulk on 2 GB ning andmekandja mahutab 32 GB andmeid [23]. Operatsioonisüsteemiks on Androidil põhinev Glass OS [27].

## **2.2.4 DAQRI Smart Helmet**

DAQRI Smart Helmet on liitreaalsusseade, mis on ehitatud kiivri baasile. Kandja silmade ette jääb prillide taoline visualiseerimisliides. Seadme aku on vahetatav ja paikneb kiivri sees, seadmega tuleb kaasa kaks 5800 mAh akut ning laadija. Kiivri kuklaosas paikneb arvuti, millele on paigaldatud Ubuntu operatsioonisüsteem. Arvuti juures paiknevad 2 USB-C pesa, mida saab kasutada seadmete ühendamiseks või aku laadimiseks. Kiivril on välismaailma tuvastamiseks kolm kaamerat: Intel® RealSense™ Camera LR200, lainurkkaamera liitreaalsusrakenduste jaoks ning soojuskaamera FLIR Lepton 3 4.1.1 [24].

Kiivri käivitamisel käivitub rakendus DAQRI OS, mis on kiivri kasutajaliideseks. Menüüs navigeerimine toimub *gaze-and-dwell* meetodil, mis näeb välja järgmiselt: liides on fikseeritud pärismaailma järgi ning ekraani keskel asub kursor, seega menüüs navigeerimiseks tuleb pead liigutades osutada kursor soovitud valikule ning seejärel umbes sekund oodata. Seadmega on ka kaasas bluetooth klaviatuur, mis on kiivriga eelnevalt paaristatud [24].

DAQRI rakendus Worksense võimaldab kasutajatel kuvada objekte ja teha videokõnesid. Objektid võivad olla nii 3D-mudelid kui ka sildid või andmed mõne pärismaailma objekti kohta näiteks nagu mootori voolutarve. Rakenduste loomiseks on olemas C++ API ning laiendus Unity jaoks. Mudelid ning pilte saab kuvamiseks üles laadida kasutades On-Device Portal rakendust, mis töötab kohaliku võrgu abil [34].

## **2.2.5. DAQRI Smart Glasses**

Edasiarendusena Smart Helmetist lasi DAQRI välja uue toote DAQRI Smart Glasses. Riistvaraliselt on toode oma kiivrist eelkäijaga peaaegu identne, kiivri asemel koosneb seade prillidest ning arvutist, mis on omavahel juhtmega ühendatud [35].

## 2.3 Robot Operating System (ROS)

ROS on töövahend robotlahenduste loomiseks, mille abil saab lihtsustada andmevahetust erinevate seadmete ning sensorite abil. Seades paika kindlad suhtlusstandardid, saavad täiesti erinevalt töötavad rakendused omavahel infot vahetada vähese vaevaga. Igal ROSi süsteemi osal on kindla infotüübiga sisendid ja väljundid, seega näiteks ühe sensori väljavahetamisel saab ülejäänud süsteemi jätta sisuliselt muutmata, tuleb vaid seada uus osa vastavusse projekti suhtlusstandarditega. See omadus muudab ROSis loodud sõlmed riistvarast sõltumatuks. ROS on hetkel implementeeritud programmeerimiskeeltes Python, C++ ja Lisp ning katsetused käivad ka Java ning Lua keeltes [36].

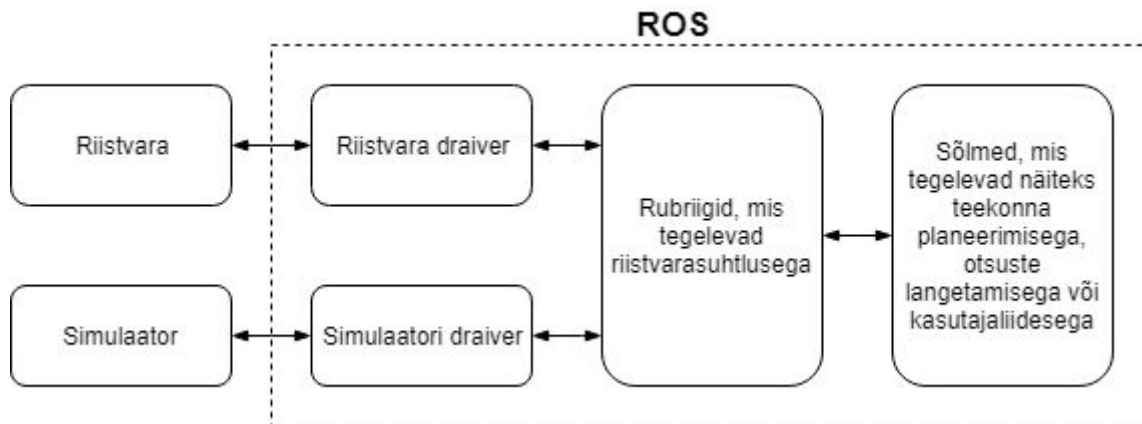
Põhjuseid, miks ROS sobib arendustegevuseks, on mitmeid. Esiteks kogub ROS kiiresti populaarsust. 2018. aastal kasvas unikaalsete külastajate poolt alla laaditud ROSiga seotud .deb failide arv 41% võrra võrreldes 2017. aastaga [37]. Mida enam on kasutajaid, seda rohkem luuakse erinevaid kimpe ning seda parem on kasutajatugi.

ROSi kasutamist süsteemide arendamisel soodustab ka asjaolu, et ROS on avatud lähtekoodi ja vähepiirava BSD-litsentsiga [38]. BSD-litsents taotleb sisuliselt vaid kahte asja: originaalkoodi osas peavad säilima autoriõigused ja originaalautor ei vastuta muudetud koodi õigesti toimimise eest.

## 2.4 Gazebo

Gazebo on simulatsioonikeskkond, mis on loodud peamiselt robotite disaini ja tööpõhimõtte katsetamiseks. Gazebo abil saab katsetada roboteid või robotite kogumeid päriselulistest keskkondades tänu paindlikule füüsikamootorile, kõrgkvaliteetsele graafikale ning lihtsasti kasutatavale kasutajaliidesele. Lisaks sellele on Gazebo ka avatud lähtekoodiga [39].

ROS ja Gazebo on omavahel tihedalt seotud alates 2009. aastast kui ROS integreeriti Gazebosse. Tänu sellele on saanud Gazebo-st üks enim kasutatud simuleerimiskeskondi ROS kogukonnas [39]. Gazebo ja ROSi lõimimiseks on loodud järgmised ROSi kimbud: gazebo\_ros\_pkgs koosneb alamjaotustest gazebo\_ros, gazebo\_msgs ja gazebo\_plugins, gazebo\_ros tegeleb kasutajaliidese tööga, gazebo\_msgs ROSi ja Gazebo vahelise suhtlusstandarditega ning gazebo\_plugins roboti tööga [40].



**Joonis 4.** ROSi, riistvara ja simulaatori omavaheline suhtlus.

Põhjus, miks Gazebo sobib robotite simuleerimiseks hästi on see, et ROSi jaoks pole oluline, kas töö toimub simuleeritud keskkonnas või pärismaailmas – infovahetus sõlmede vahel jääb samasuguseks (joonis 4). Robotil olevad andurid saavad tänu Gazebo füüsikamootorile pärismaailmale sarnaseid lugemeid, seega ROSi vaatepunktist on tegevus samasugune nii pärismaailma kui simulatsioonikeskkonna puhul. Tänu rubriikidele, mis on draiverite ning peaprogrammi vahel, ei tea kumbki osapool üksteise kohta midagi, vastu võetakse vaid vajalikku infot.

## 2.5 Olemasolevad liitreaalsuslahendused ROSis

ROS-Industrial on konsortsium, mis toetab ROSi-põhiseid tootmislahendusi. Nende hulka kuulub ka kolm liitreaalsuslahendust: ARViz, RViz2AR ja ROS2AR. Neist viimase kohta puudub täpsem info, kuid tundub, et UCTN (*Technical University of Cluj-Napoca*) Rumeenias proovib luua ROSi abil süsteemi, mis visualiseerib andmeid erinevate objektide kohta [10], [11].

ARViz on Microsoft Hololensile loodud liitreaalsusrakendus Rey Juan Carllose Ülikooli robotikagrupi poolt Hispaanias. Projekt on loodud eesmärgiga kombineerida TF2 ning UWP port ROS2 jaoks, et luua rakendus, mille abil saab luua liidese, mille abil jälgida roboti tööd ning sensori lugemeid. Projektiga on jõutud niikaugemale, et suudetakse tuvastada Hololensi asukoht maailma suhtes [12].

RViz2AR on nii liit- kui virtuaalreaalsusprojekt, mille eesmärgiks on luua liides töötaja ning tehase masinate vahel, mille abil saab visualiseerida andmeid masinate seisukorra kohta ja tuge masinate kasutamiseks. Valmis projekt peaks olema ühilduv kõigi praeguste

kättesaadavate liit- ja virtuaalreaalsusseadmetega. Andmete visualiseerimiseks kasutatakse Unityt, seadmete sensoritega suhtlemiseks ROSi. ROSi ja Unity omavaheliseks suhtluseks kasutatakse ROSBridge kimpu. Rakendust arendab Saksamaa firma Awesome Technologies Innovationslabor GmbH [13].

# 3 Nõuded

## 3.1 Töö eesmärk

Luaa ROSi-põhine liitreaalsuslahendus, mille abil saab LR-visiiri kasutades kujutada virtuaalseid objekte võimalikult täpses asukohas. Mõõta süsteemi täpsust erinevates olukordades.

## 3.2 Funktsionaalsed nõuded

- Vaadates läbi LR-visiiri kuvatakse kasutajale virtuaalne kujutis õiges asukohas võrdlusobjekti suhtes.
- Kujutis peab olema piisavalt stabiilne, et kasutajal oleks võimalik määrata selle järgi asukohti pärismaailmas.

## 3.3 Riistvaralised nõuded

- Kasutataval LR-visiiril peab töötama Ubuntu Linux operatsioonisüsteem.
- Visiiri esiosas peab paiknema kaamera.
- Arvutusmahu vähendamiseks LR-visiiris võimaldada arvutuste jaotumine mitme arvuti vahel.

## 3.4 Tarkvaralised nõuded

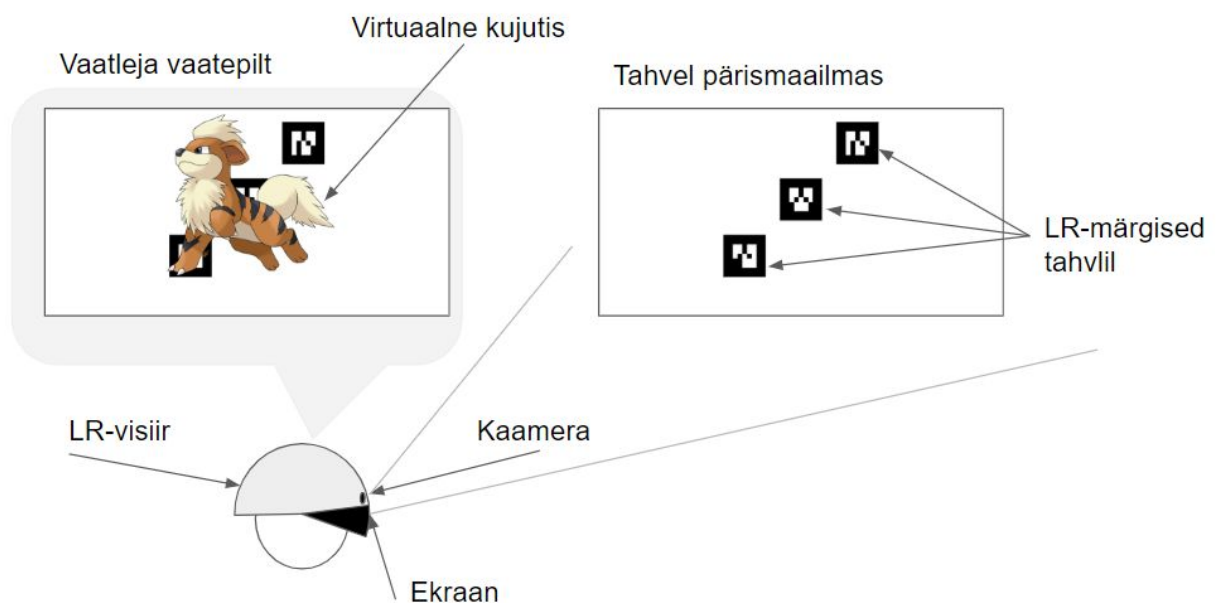
- Virtuaalsete kujutiste visualiseerimiseks kasutada simulatsioonirakendust Gazebo.
- Töö raames loodud ROSi kimpude kood kirjutada programmeerimiskeeles C++ ning kood kommenteerida ning vormistada vastavalt ROS C++ stiilinõuetele [41].

# 4. Disain

## 4.1. Tööpõhimõte

Käesoleva projekti raames on loodud süsteem, mille abil saab kasutada liitreaalsust objektide lisamisel pärismaailma võimalikult täpselt. Kasutaja saab läbi LR-visiiri näha virtuaalseid objekte, mis paiknevad pärismaailma suhtes kindlaksmääratud kohal.

Liitreaalsus koosneb kahest maailmast, mis kattuvad omavahel. Selleks, et virtuaalsed objektid asuks vaatleja vaatepunktist õiges kohas, on vaja virtuaalne maailm pärismaailmaga joondada. Selle jaoks on käesolevas töös kasutusel LR-märgised (joonis 5), mille asukohad on nii reaalsuses ka virtuaalsuses kindlalt paika pandud. Kasutades mitut LR-märki, saab andmeid kombineerides laiendada tööpiirkonda, suurendada täpsust ning tõsta veakindlust. Kui luua virtuaalses maailmas pärismaailma objekti koopiat (*digital twin*) ning asetada see virtuaalses maailmas märgise suhtes samasse asukohta, kattuvad mõlemad objektid kasutaja vaateväljas omavahel. Kui nüüd virtuaalsele objektile pärismaailmas puuduvaid täiendusi lisada, tekib kasutajala liitreaalsukuva (joonis 5).



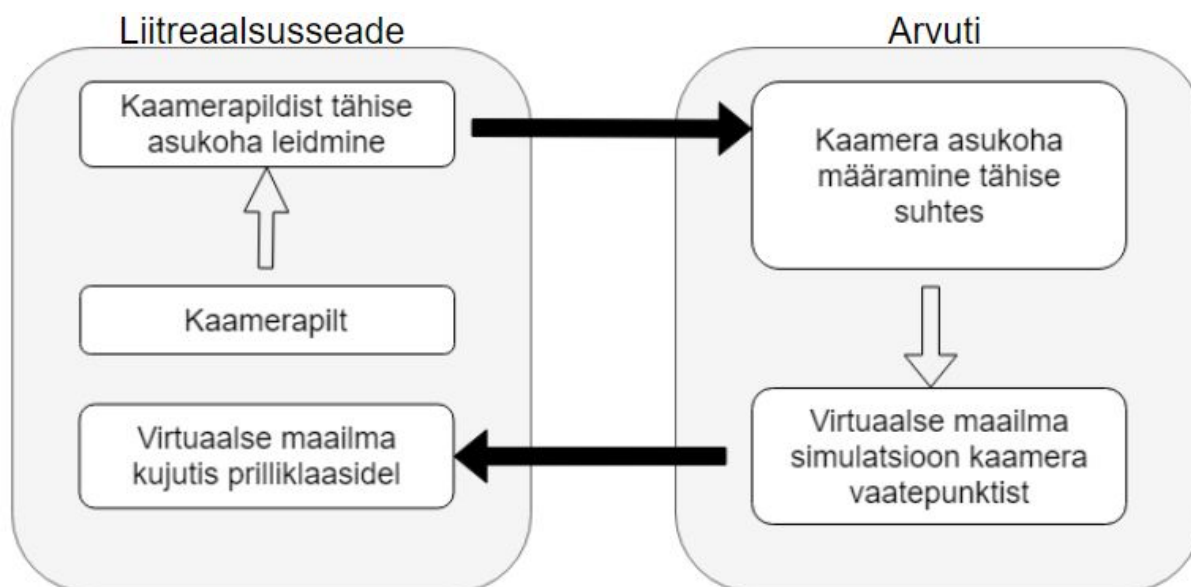
**Joonis 5.** LR-süsteemi ülevaade. Vaade tahvlile läbi LR-visiiri (vasakul) ning algne tahvel pärismaailmas (paremal).

## 4.2 Arhitektuur

Kirjeldatud funktsionaalsuse saavutamiseks on esmalt vaja määratleda vaatleja asukoht pärismaailmas, selleks on vaja kaamerapildist LR-märgised ja nende asendeid kaamera suhtes. Märkiste asendite põhjal saab kindlaks teha vaatleja asukoha töökeskkonnas. Teades, kus asub vaatleja ning kus asuvad märgised mõlemas reaalsuses, saab virtuaalset maailma kuvada samast vaatepunktist pärismaailmaga.

Süsteemi katsetamiseks kasutati LR-visiiri DAQRI Smart Helmet. Visiir osutus valituks seetõttu, et ta oli parasjagu kättesaadav ning vastas antud projekti nõuetele. Vastavalt nõuetele on süsteem loodud ROSi ja Gazebo tarkvara abil.

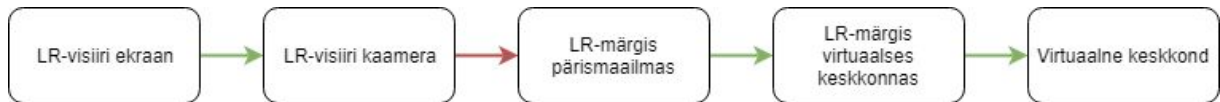
Pärismaailma simuleerimine nõuab palju jõudlust, seega on otstarbekas sellega LR-visiiri pardaarvutit mitte koormata. Seetõttu on valminud lahenduse arvutusjõudlus jagatud eraldi arvuti ning visiiri vahel (joonis 6). Kuna ROSi sõnumid liiguvad niikuinii üle võrguliidese [42], on hajussüsteemi loomine juba arhitektuurselt toetatud.



**Joonis 6.** LR-süsteemi voodiagramm ja ülesannete jaotus kahe arvuti vahel.

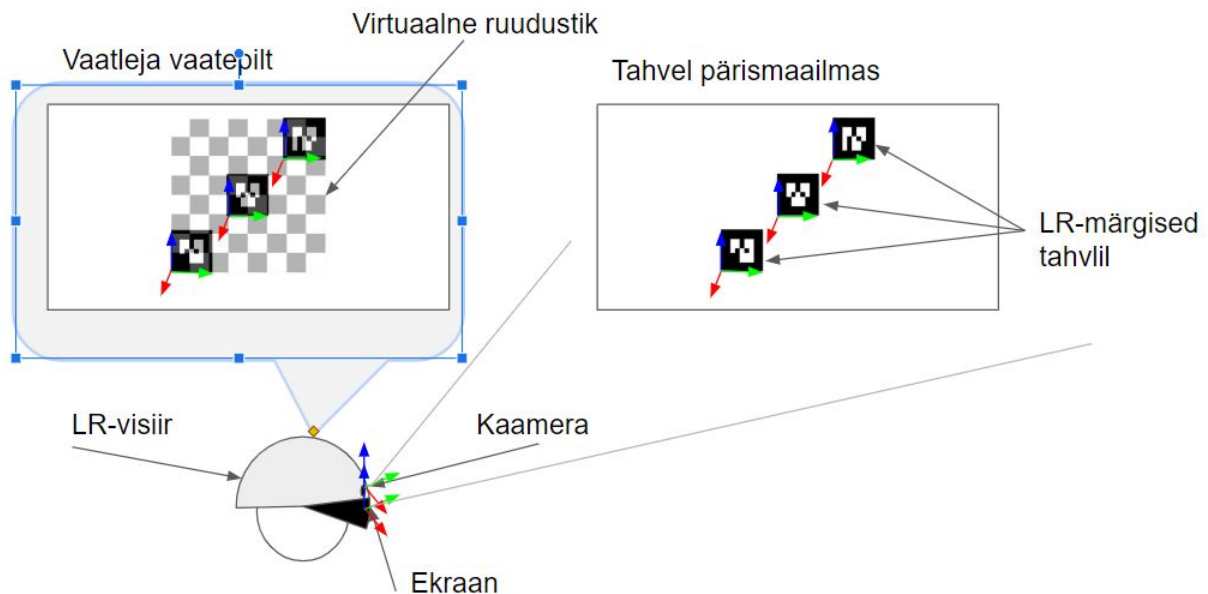
Virtuaalse maailma joondamist pärismaailmaga saab kujutada teisenduspuuna (joonis 7). Igal selle liikmel (koordinaatteljestikul) on defineeritud asend (positsioon ning orientatsioon) (joonis 8). Teisenduspuus (joonis 7) iga kahe noolega ühendatud liikme vahel on asukohaline seos (teisendus). Kui liikmed on omavahel andmepuu ülesehitusega seotud, saab tuletada iga kahe puu liikme vahelise teisenduse. Seega käesoleva projekti ülesehituse korral saab

teisenduspuus teha päringu, kuidas paikneb LR-visiiri ekraan virtuaalse keskkonna suhtes. Selle teisenduse abil paigutataksegi virtuaalne objekt pärismaailma suhtes õigesse kohta. Mitme märgise kasutamise puhul keskmistatakse kõigi leitud märgiste teisendused liikuva keskmise abil.



**Joonis 7.** Teisenduspuu, roheline noolega märgitud teisendused on eeldefineeritud, punase joonega teisendus arvutatakse kaamerapildist.

Pärast virtuaalse keskkonna kaamera joondamist LR-visiiri ekraaniga on vaja joondada vaatleja silma vaateväli liitreaalsusseadme vaateväljaga, sest liitreaalsusseadmed tavaliselt ei kata tervet inimese vaatevälja (tabel 1). Seda probleemi saab lahendada näiteks rakenduse CompizConfig Settings Manager [43] abil, suurendades kogu nähtavat liidest vaatleja vaatevälja suuruseni.

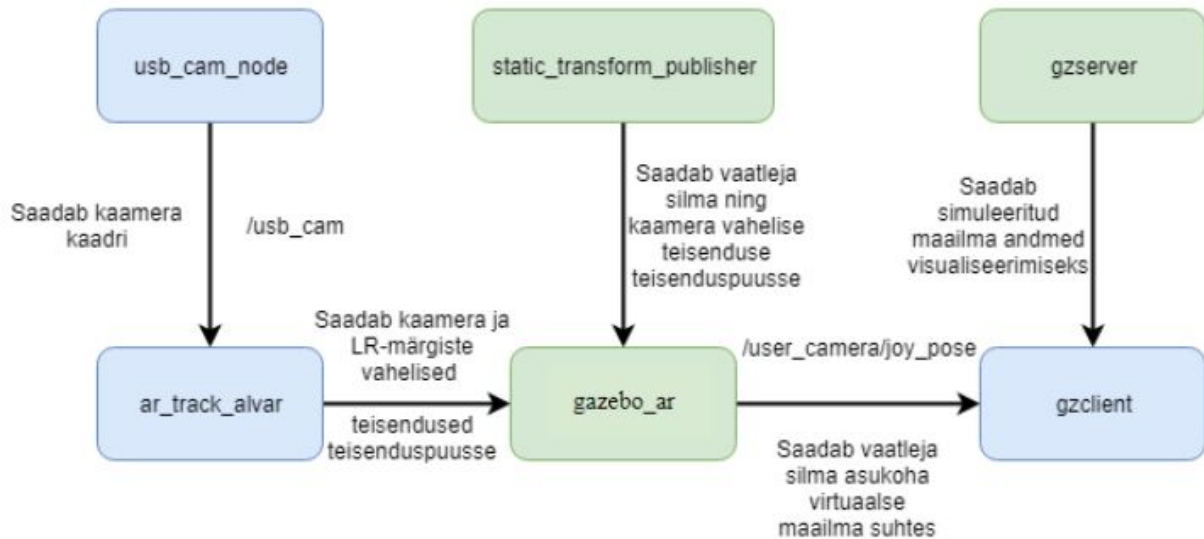


**Joonis 8.** Testkeskkond koordinaatteisenduste vaatepunktist.

### 4.3 Kasutatud kimbud ja konfiguratsioon

Töö raames kasutati olemasolevaid ROSi kimpe `usb_cam` [44], `ar_track_alvar` [45], `tf` [46] ja `gazebo_ros` [47]. Kogu valminud lahenduse saab jaotada kahele seadmele (joonis 6), et vähendada koormust LR-visiiril, sest nende seadmete jõudlus on tihti piiratud ning koormus

mõjutab aku eluiga. Tööjaotus, mille kohaselt visiiril jookseb minimaalne arv sõlmi, on järgmine: LR-visiiri peal on käitatud sõlmed `usb_cam_node`, `ar_track_alvar` ja `gzclient`, serveris `gazebo_ar`, `static_transform_publisher` ja `gzserver` (joonis 9).

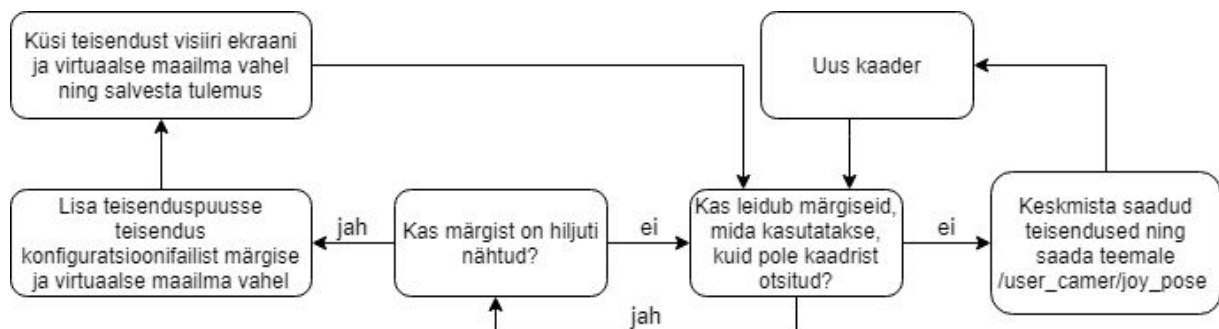


**Joonis 9.** *Süsteemi ülesehitus ROSi rubriikide ja sõlmede kujul, rohelisega märgitud sõlmed asuvad serverarvutis, sinisega märgitud sõlmed LR-visiiris.*

Sõlm `usb_cam_node` on vajalik kaamerapildi hõivamiseks ning ROS-sõnumina kuulutamiseks. Antud rakenduse kontekstis piisab kimbuga kaasasolevast `usb_cam_test.launch` käivitusfailist. Katse raames kasutati lahutusvõimet  $1920 \times 1080$ , kuid seda saab muuta vastavalt vajadusele käivitusfailis. Sõlm `ar_track_alvar` tegeleb kaamerapildist LR-märgiste asukohtade tuletamisega. Antud sõlm on käitatud visiiris, mitte serverarvutis seetõttu, et kaamera kaadrite saatmisele kulub umbes kümme korda rohkem aega kui märgiste leidmine visiiris ning nende asukohtade saatmine serverarvutisse. Käivitusfailis `usb_cam_test.launch` tuleb defineerida rubriik, kus kaamera kaadrisõnumeid kuulutatakse ning LR-märgise küljepikkus. Sõlm `ar_track_alvar` kuulutab kõigi kaadris nähtavate märgiste asukohad rubriigis `ar_pose_marker` ning lisab kaamera ja iga märgise vahelise teisenduse teisenduspuusse. Sõlm `gzclient` tegeleb Gazebo simulatsioonikeskkonna visualiseerimisega. Selleks, et joondada virtuaalne keskkond pärismaailmaga, tuleb Gazebo graafiline liides kuvada täisekraanivaates ning peita tööriistariba.

Sõlm gzserver tegeleb simulatsioonikeskkonna jaoks vajalike arvutustega, static\_transform\_publisher edastab teisendust liitreaalsusseadme kaamera ning LR-visiiri ekraani vahel.

Sõlm gazebo\_ar on käesoleva lahenduse jaoks spetsiaalselt loodud. Antud sõlmes tuletatakse märgiste asukohtadest LR-visiiri kaamera suhtes välja kaamera asukoht Gazebo simulatsioonikeskkonnas, seda asukohta loeb sõlm gzclient (joonis 10). Tuletatud kaamera asukohast ning asendist virtuaalset maailma kuvades kattub virtuaalne kujutis objektiga pärismaailmas. Iga kasutatava märgise puhul vaadatakse programmi peatsükliks, kas seda märgist on hiljuti kaamerapildis nähtud. Kui on, siis lisatakse teisenduspuusse teisendus antud märgise ning virtuaalse maailma vahel (joonis 7) ning küsitakse teisendust visiiri ekraani ning virtuaalse maailma vahel. Kõik saadud väärtused keskmistatakse ning tulemuseks on simulaatori kaamera asukoht, mille vaatepunktist kujutatuna LR-visiiril kattub virtuaalne maailm pärismaailmaga. See asukoht saadetakse teemale /user\_camera/joy\_pose.



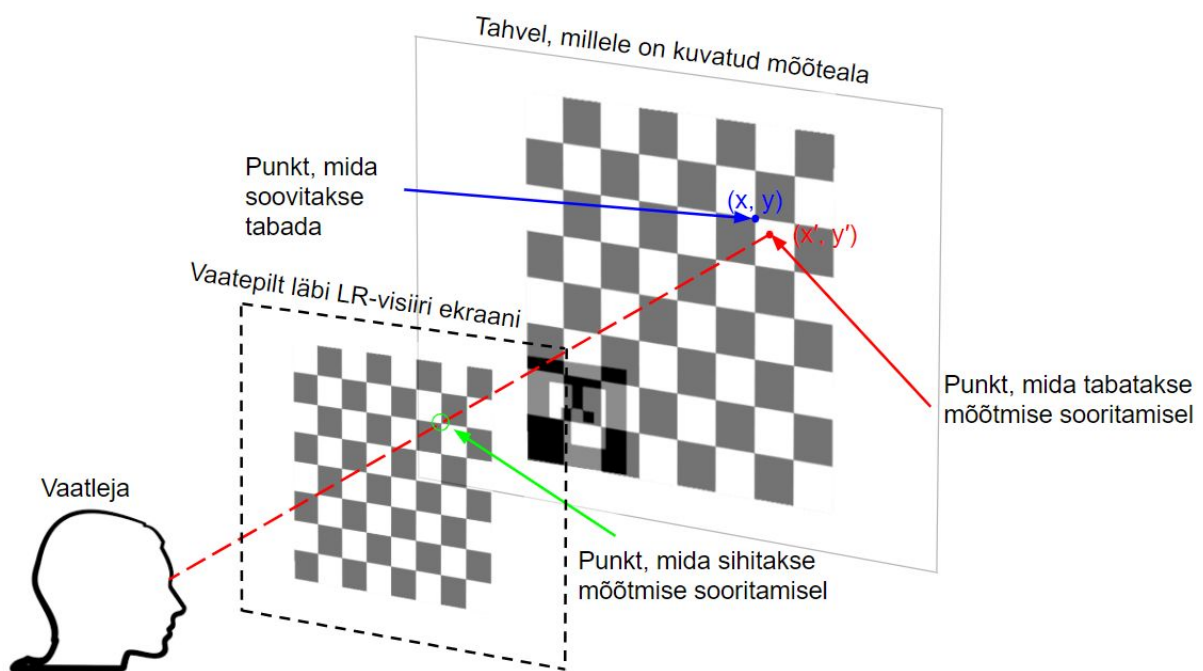
**Joonis 10.** Sõlme gazebo\_ar töövoog.

Kogu antud süsteem töötab maailmakella järgi. LR-visiir DAQRI Smart Helmet on akutoitega ning süsteemikell muutub aku eemaldamisel valeks. Lisaks sellele ei arvestanud LR-visiiri arvuti aega õigesti, süsteemikell läbis ühe minuti umbes 50 sekundiga. Seega oli töö raames vaja kasutada sünkroniseerimisrakendust chrony [48], kus serveriks oli kasutatav arvuti ning kliendiks LR-visiir. Seadistamine toimus chrony kodulehel oleva juhise järgi [49].

# 5 Karakteriseerimine

Käesoleva bakalaureusetöö raames loodud süsteemi kasutatavuse määramiseks töötati välja meetodika ning viidi läbi täpsuse mõõtmised erinevates olukordades. Mõõdeti nihet soovitud asukoha pärismaailmas ja LR-viisiris osutatud asukoha vahel. Link mõõtmistulemuste tabelile asub lisa 1.

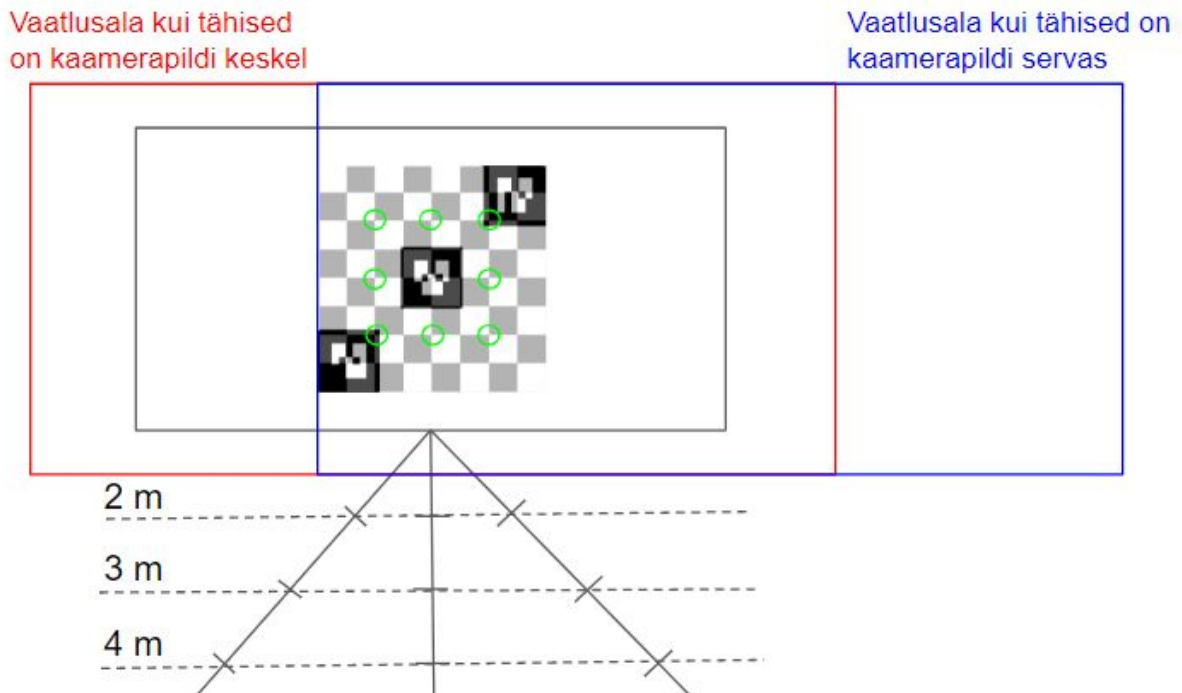
## 5.1 Meetodika



**Joonis 11.** Mõõtmisprotsess. Virtuaalse kujutise nihe on  $(x'-x, y'-y)$ .

Katsekeskkonnas vaatas inimene läbi DAQRI Smart Helmeti tahvlile istudes tahvlist kindlal kaugusel ning kindla nurga all. Tahvlile projitseeriti projektoriga üks või mitu LR-märgist, mis paiknesid mõõteala kindlates asukohtades. LR-viisiri ekraanile kuvati mõõteala suurune ruudustik, mis oli fikseeritud mõõtealale (joonis 11). Ruudustik osutus valituks seetõttu, et nelja ruudu kokkusaamispunkt on väga täpselt määratav ning selliseid punkte on ruudustikul küllalt, et varieerida vajutuskohti. Et mõõteala suurus oli teada, sai tuletada asukohad, kus ruudustik LR-viisiril peaks asetsema. Seejärel sooritati hiireklikk LR-viisiril kuvatud ruudustiku ruutude kohtumispunktis ning salvestati kliki asukoht mõõtealal. Arvutatud ruutude kohtumispunkti ning hiirekliki asukoha vahe on nihe virtuaalse ja reaalse maailma

vahel (joonisel 11 ( $x'-x$ ,  $y'-y$ )). Katse olemuse paremini selgitamiseks on joonisel 10 tahvlile lisatud ruudustik, katse läbiviimisel paiknesid seal vaid LR-märgised.



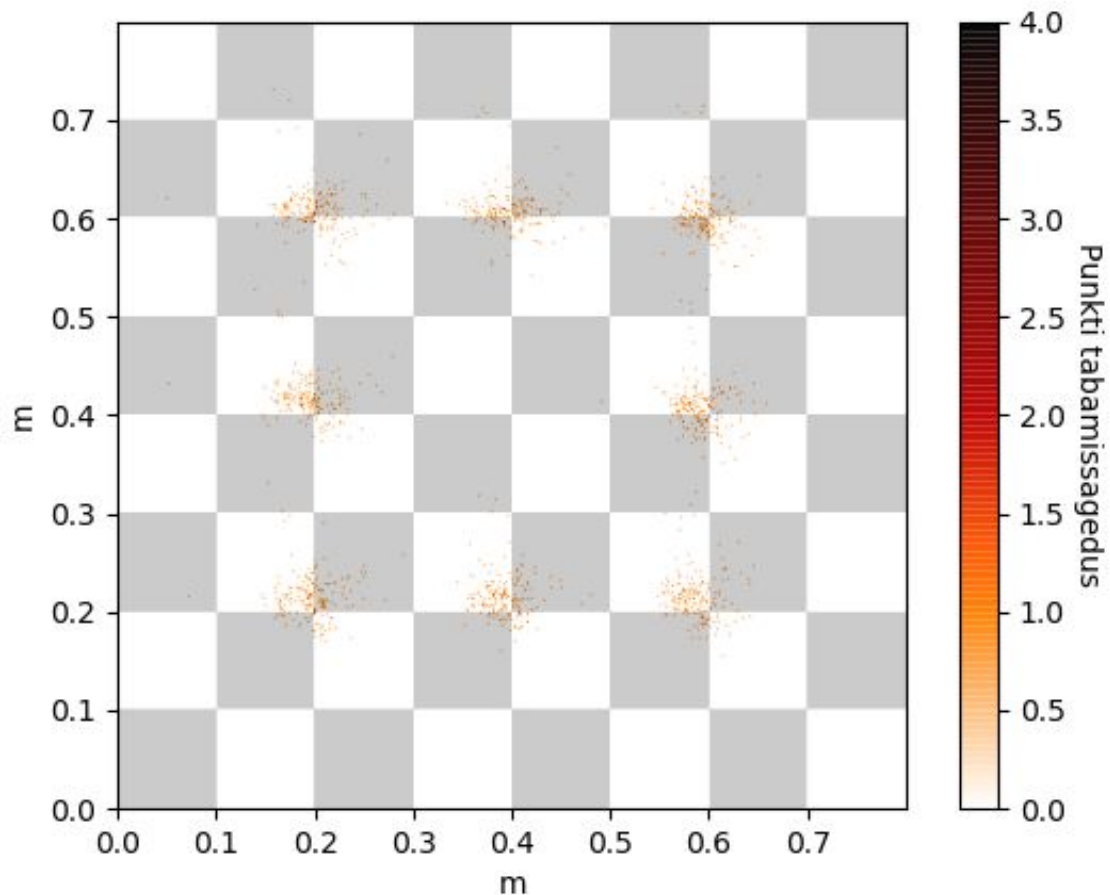
**Joonis 12.** Testkeskkonna ülesehitus, roheliselt on märgitud hiirevajutusasukohad virtuaalsel ruudustikul.

Mõõtmisi sooritati kolme nurga alt ( $30^\circ$  tahvlit vasakult, otse eest ning  $30^\circ$  tahvlit paremalt), iga nurga alt kolmelt kauguselt (2 m, 3 m ning 4 m) (joonis 12).

Süsteemi testiti kolmes märgisekonfiguratsioonis: üks märgis all vasakul, märgised all vasakul nurgas ning keskel ja märgised all vasakul nurgas, keskel ning üleval paremal nurgas ja kahel viisil: märgised paiknesid kas vaatevälja keskel või ääres (joonis 12).

Mõõtmised sooritati 8 kaupa, sooritades hiirekliki virtuaalse ruudustiku kaheksas sõlmpunktis ringjooneliselt päripäeva liikumissuunas (joonis 13). Esimeseks sõlmpunktiks loeti ülemist vasakpoolset.

Teostati kõikvõimalikud mõõtmisvariatsioonid ja igas asendis sooritati 10 mõõtmist, seega mõõtmistulemuste arv oli  $8 \cdot 10 \cdot 3 \cdot 3 \cdot 2 = 4320$ .



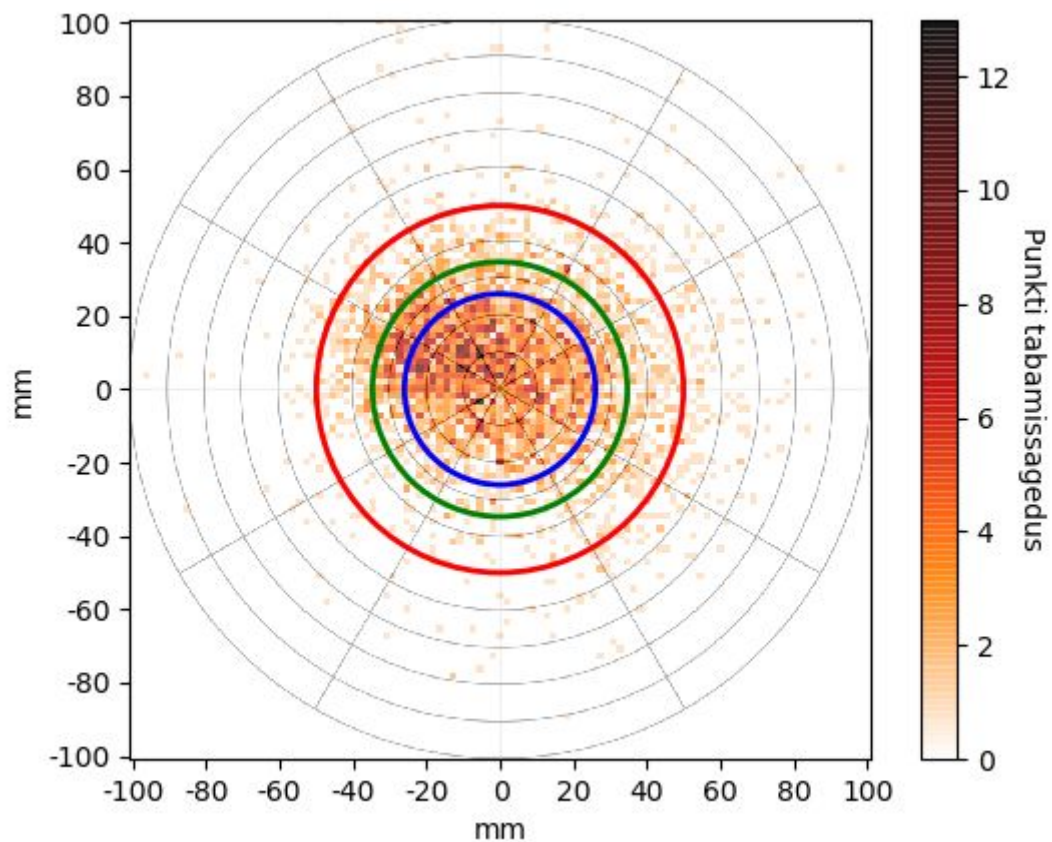
Joonis 13. Sooritatud mõõtmised terve katseala kontekstis.

## 5.2 Süsteemi üleüldine täpsus

Süsteemi iseloomustamiseks kogutud andmepunktid on kujutatud joonisel 14. Kõigi mõõtmistulemuste mediaannihe oli 25.8 mm, keskmine nihe 28.6 mm ning standardhälve 17.5 mm. Ligi kahe ja poole sentimeetrine nihe on tööstuses lubamatu, samas juhendite kuvamiseks või disaini visualiseerimiseks kuvatakse kujutis piisavalt õiges asukohas. Samas on 90% mõõtmistulemustest 5 cm raadiuses, seega süsteem on madala õigsuse, kuid hea täpsusega.

Kõigi mõõtmiste täpsust kujutavast graafikust (joonis 14) jäeti välja 32 andmepunkti, mille horisontaalne või vertikaalne nihe oli suurem kui 10 cm. Antud katse kontekstis on 32 tulemust 0.007% kogu tulemustest, seega graafiku üldpilt palju ei muutu. Välja jäänud väärtuseid kasutatakse mõõtmist iseloomustavate väärtuste arvutamisel. Tasub märkida, et

nende 32 mõõtmise sees olid mitmed vead järjestikusel mõõtmisel, mis viitab sellele, et antud vaatenurga all oli kaamerapildis midagi, mis põhjustas süsteemse mõõtevea.



**Joonis 14.** Kõigi sooritatud mõõtmiste täpsus, sinise ringi sees on 50%, rohelises ringis 70% ning punases ringis 90% mõõtmistulemustest.

### 5.3 Täpsuse sõltuvus LR-märgiste arvust

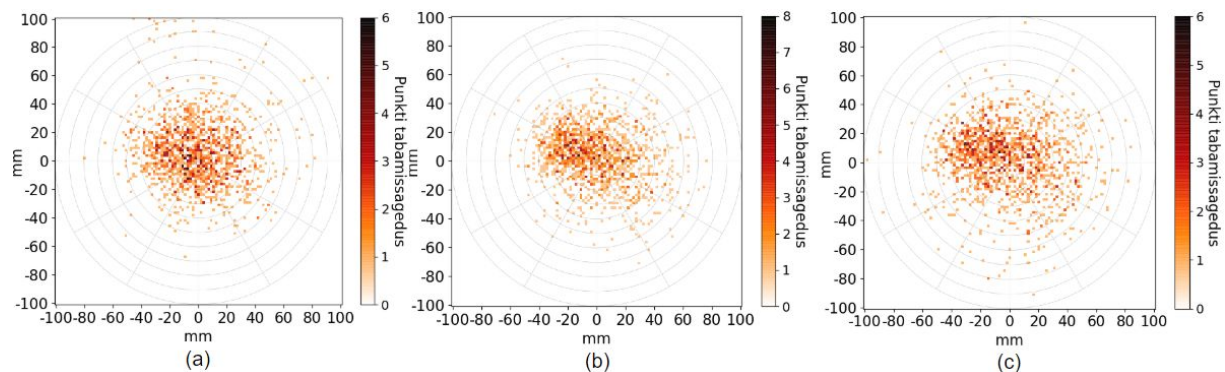
**Tabel 2.** Mõõtmistulemused erineva märgiste arvuga.

	1 märgis	2 märgist	3 märgist
<b>Keskmine nihe</b>	29.0	26.3	30.6
<b>Mediaannihe</b>	25.1	24.5	28.2
<b>Standardhälve</b>	20.8	14.0	16.9

Mõõtmistel kogutud andmepunktid on jaotatud märgiste arvu järgi joonisel 15. Kõige täpsemaks osutus kahe märgise kasutamine, seejärel ühe märgise kasutamine. Süsteem oli kõige ebatäpsem kasutades kolme märgist (tabel 2).

Mitme märgise kasutamine annab teoorias rohkem andmepunkte, mille abil leida vaatleja asukoht objekti suhtes. Samas suureneb võimalus ühest kaamera kaadrist tuletada vale asukohaga märgis, sest märgiseid, mida võib ühest kaadrist valesti tuvastada, on rohkem.

Antud lahenduses ei filtreeritud väärtusi üldse, seega tulemus oluliselt ei paranenud. Tasub märkida, et ühe LR-märgisega süsteemi testimisel esinesid 27 32st suuremast mõõtevast, mis võis põhjustada tulemuse, et ühe märgise kasutamine on täpsem kui kolme märgise, kuid ebatäpsem kui kahe märgise kasutamine (joonis 15). Nende tulemuste põhjus võib seisneda ka selles, et katse läbiviija sihtis valet punkti.



Joonis 15. Süsteemi täpsus ühe (a), kahe (b) ja kolme märgisega (c).

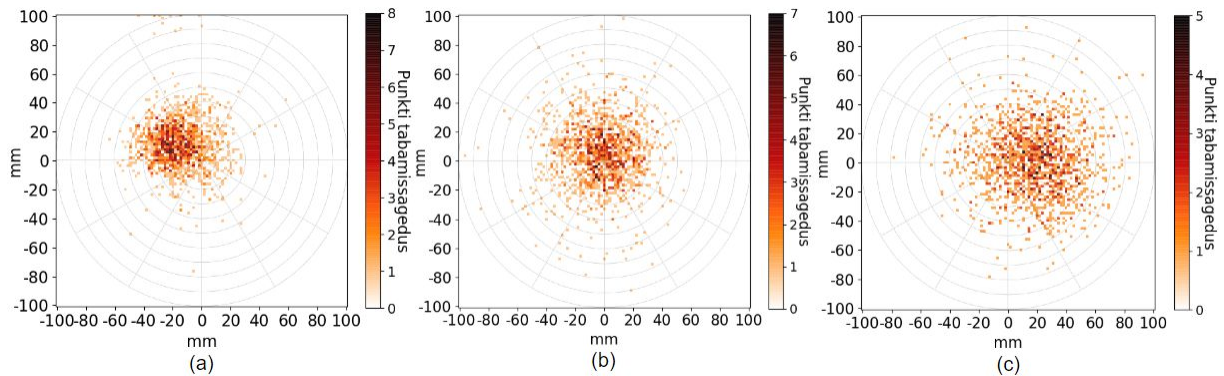
## 5.4 Täpsuse sõltuvus vaatlemiskaugusest

Tabel 3. Mõõtmistulemused erinevatelt kaugustelt.

	2 meetrit	3 meetrit	4 meetrit
<b>Keskmine nihe</b>	28.7	24.8	32.4
<b>Mediaannihe</b>	26.2	21.8	30.2
<b>Standardhälve</b>	18.5	16.2	17.0

Mõõtmistulemused on kujutatud tabelis 3. Jooniselt 16 ilmneb, et kaugusega täpsus väheneb ning kahe meetri kauguselt on virtuaalne kujutis umbes 2 cm nihkes vasakule (joonis 16-a) ning 4 m kauguselt sama palju paremale (joonis 16-c). Tänu nihkele keskpunktist teistel mõõtmiskaugustel on 3 m kauguselt sooritatud mõõtmised kõige täpsemad. Täpsuse kahanemine kauguse suurenemisega oli oodatud, sest kaamera lahutusvõime piirab LR-märgise kujutise täpsust ning suurus kaamerapildis kahaneb. Mida suurem on kaamera lahutusvõime, seda täpsemini suudab märgisetuvastussõlm servi tuvastada ja seeläbi ka

täpsemini märgise asukohta määrata. Tabamuste keskpunkti nihke põhjuseks võib olla kaamera ja LR-visiiri ekraani vahelise teisenduse ebatäpsus.



Joonis 16. Süsteemi täpsus kahe (a), kolme (b) ja nelja meetri kauguselt (c).

## 5.5 Täpsuse sõltuvus vaatlemissuunast

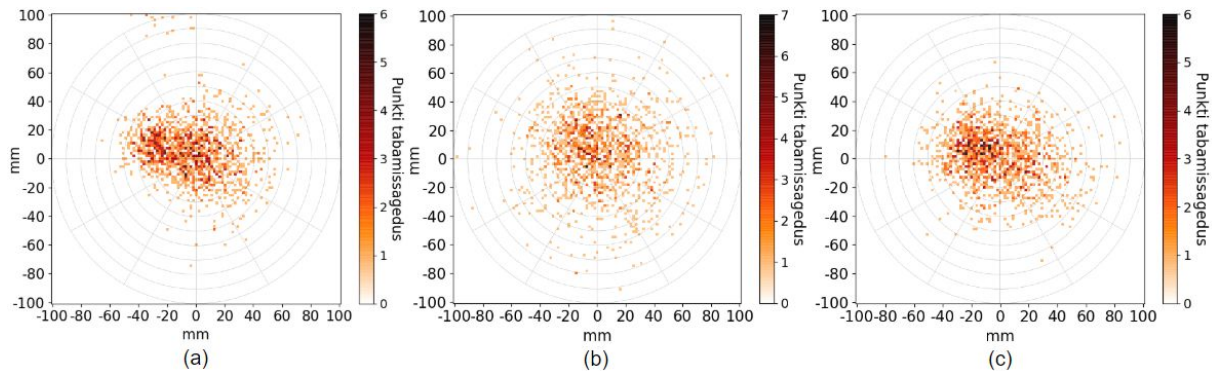
Tabel 4. Mõõtmistulemused erinevatest suundadest.

	30° vasakult	otse eest	30° paremalt
<b>Keskmine nihe</b>	28.0	31.2	26.7
<b>Mediaannihe</b>	25.1	28.0	24.5
<b>Standardhälve</b>	19.5	18.5	13.8

Mõõtmistulemused on kujutatud graafikutel jaotatuna vaatlemissuuna järgi joonisel 17. Mediaannihe on sarnane kui tahvlit vaadati paremalt ja vasakult (tabel 4). Otse eest vaadates aga oli süsteem tunduvalt ebatäpsem. Vaadates graafikut on näha, et hajuvuse poolest on vasakult ning paremalt vaadates täpsus sarnane, kuid vasakult vaadates on kogum mõõtmisi, mis asuvad skaala piiril. Lisaks peatükis 5.2 mainitud väga suure nihkega mõõtmistulemused asuvad just vasakult vaatamise mõõtmistulemuste seas. See võib põhjustada suure erinevuse standardhälbes, kuid väikese erinevuse mediaannihkes paremalt ning vasakult vaatamise vahel (tabel 4).

Erinevast suunast vaadates kuvatakse kujutist erinevas orientatsioonis, seega mõõtmistes võib ilmnedas teisendusvigu. Üllatuslikult ilmneb, et keskelt mõõtes on tulemused väga hajusad võrreldes äärtest sooritatud mõõtmistega (joonis 17). See võib viidata probleemile

nullteisendusega, sest orientatsioonimuutjad on siis nullilähedaseimad. Samas sooritati kõik teisendused kvaternioonide abil, mis on just selle probleemi lahendamiseks loodud.



Joonis 17. Süsteemi täpsus vaadeldes vasakult (a), keskelt (b) ja paremalt (c).

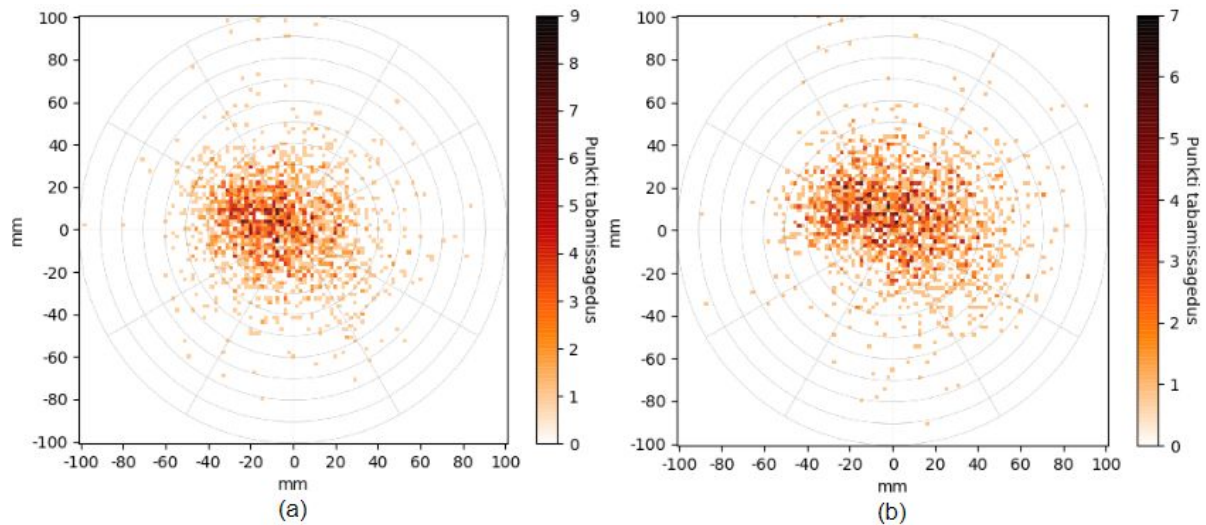
## 5.6 Täpsuse sõltuvus märgiste asukohast kaamerapildis

Tabel 5. Mõõtmistulemused erineva märgiste paigutusega kaamerapildis.

	Märgised vaatevälja ääres	Märgised vaatevälja keskel
<b>Keskmine nihe</b>	30.3	27.0
<b>Mediaannihe</b>	27.2	24.2
<b>Standardhälve</b>	18.6	16.2

Mõõtmistulemused on kujutatud joonisel 18 ja tabelis 10. Mõõtmistest ilmneb, et vaadates mõõteala nii, et märgised on vaatepildi keskel, on süsteem oluliselt täpsem kui jättes märgised vaatepildi serva.

Mõõtmised teostati nii, et LR-märgised asuvad kas kaamerapildi ääres või keskel sel eesmärgil, et näha, kas märgisetuvastusega seotud teisendused töötavad, sest kui märgis paikneb kaamerapildi ääres, on tema asukohta kirjeldavad asukohamuutjad suuremate väärtustega ning teisendusel tekkiv nihe on suurem. Katsest selgus, et mediaannihe on märgise paigutusel kaamerapildi servas umbes 3 mm suurem kui märgise paigutusel kaamerapildi keskel (joonis 18). Selline nihe võib tuleneda näiteks kimbu `ar_track_alvar` käivitusfailis defineeritud märgise suuruse ebatäpsusest.



**Joonis 18.** Süsteemi täpsus kui märgised on kaamerapildi keskel (a) ja kaamerapildi servas (b).

## 5.7 Järeldused

Antud mõõtemetoodika toomis, saadi mitmeid erinevaid tulemusi, mille abil saab iseloomustada süsteemi tööd. Töö kontekstis oleks olnud sobilikum teostada mõõtmisi sarnaselt kauguselt, nagu kasutatakse süsteemi tootmiskeskkonnas. Samas märgise suuruse kahanedes muutub kaugusvahemik, kus süsteem on antud riistvaraga stabiilne. Viies mõõtmised suuremale skaalale, saadi täpsuse arvelt tulemused, mis iseloomustavad hästi süsteemi tööd. Selline iseloomustus on oluline edasiarenduse vaatepunktist.

Osa mõõdetud nihkest on kindlasti tingitud katse ülesehitusest. Hiirekursoriga kaugelt sihtmärgi tabamine on juba isenesest ebatäpne, kui vaateväljas on lisaks virtuaalne kujutis, on õige koha tabamine veelgi keerulisem. Seda kinnitab ka asjaolu, et 90% mõõtmistest asusid 5 cm raadiuses keskpunktist ning tulemused on selles alas ühtlaselt jaotunud. Seega on autori arvates süsteem tegelikkuses täpsem kui 25.8 mm. Tulevikus võiks sooritada katseid lähtuvalt rakendusele tööstuses, tuues katseala vaatlejale käeulatusse.

## 6. Kokkuvõte

Käesoleva bakalaureusetöö raames loodi ROSi ja Gazebo-põhine avatud lähtekoodiga liitreaalsusrakendus. Loodud süsteemi saab küll kasutada simulatsioonide visualiseerimiseks või virtuaalsete objektide kuvamiseks, sest 90% mõõtmistest olid 5 cm raadiuses keskpunktist. Samas pole mõõdetud 25.8 mm mediaannihe piisavalt väike, et lahendust tööstuses rakendada. Kuna loodud platvorm on nii vabavaraline kui ka riistvaraülene, siis on võimalik seda tulevikus täiendada. Tänu ROSi modulaarsusele saab tulevikus lisada kimbule juurde näiteks IMU kasutamise võimekus ning rakendada algoritme valeandmete filtreerimiseks. Teostatud katsete tulemustest saab välja lugeda, mis osi loodud süsteemis on vaja parandada. Näiteks mõõtmiste nihkumine horisontaalsihis sõltuvalt kaugusest ning suur nihe vaadates süsteemi otse eest võrreldes vaatamisega nurga alt viitavad teisendusveale süsteemis.

Töö käigus loodud gazebo\_ros pakett on hetkel kättesaadav UT IMS Robotics organisatoorses GitHub koodihoidlas aadressil: [https://github.com/ut-ims-robotics/gazebo\\_ar](https://github.com/ut-ims-robotics/gazebo_ar).

# Viited

- [1] M. Wiktorsson, A. Granlund, M. Lundin, ja B. Södergren, „Automation and Flexibility: Exploring Contradictions in Manufacturing Operations“, 2016.
- [2] „The best augmented reality glasses 2019: Snap, Vuzix, Microsoft, North & more“, *Wareable*, 05-märts-2019. [Online]. Available at: <https://www.wareable.com/ar/the-best-smartglasses-google-glass-and-the-rest>. [Vaadatud: 10-mai-2019].
- [3] P. Milgram ja F. Kishino, „A Taxonomy of Mixed Reality Visual Displays“, *IEICE Trans Inf. Syst.*, kd E77-D, nr 12, lk 1321–1329, dets 1994.
- [4] V. Hayward, O. R. Astley, M. Cruz-Hernandez, D. Grant, ja G. Robles-De-La-Torre, „Haptic interfaces and devices“, *Sens. Rev.*, kd 24, nr 1, lk 16–29, märts 2004.
- [5] R. Manthorpe, „Listen up: augmented reality is coming to your ears“, *Wired UK*, 24-dets-2016.
- [6] „What Is Pokémon Go and Why Is Everyone Talking About It?“ [Online]. Available at: <https://lifehacker.com/what-is-pokemon-go-and-why-is-everyone-talking-about-it-1783420761>. [Vaadatud: 17-mai-2019].
- [7] „Pokémon GO Revenue and Usage Statistics (2017) - Business of Apps“. [Online]. Available at: <http://www.businessofapps.com/data/pokemon-go-statistics/>. [Vaadatud: 07-apr-2019].
- [8] A. Syberfeldt, O. Danielsson, M. Holm, ja L. Wang, „Dynamic Operator Instructions Based on Augmented Reality and Rule-based Expert Systems“, *Procedia CIRP*, kd 41, lk 346–351, 2016.
- [9] D. Stricker, „Tracking with reference images: a real-time and markerless tracking solution for out-door augmented reality applications“, *Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage - VAST '01*, Glyfada, Greece, 2001, lk 77.
- [10] „Welcome to AR City: Augmented Reality Maps and Navigation“. [Online]. Available at: <https://www.blippar.com/blog/2017/11/06/welcome-ar-city-future-maps-and-navigation>. [Vaadatud: 28-apr-2019].
- [11] D. W. F. van Krevelen ja R. Poelman, „A Survey of Augmented Reality Technologies , Applications and Limitations“.
- [12] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, ja V. Kasi, „Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications“, *Int. J. Signal Process. Syst.*, lk 256–262, 2013.
- [13] J. Novak-Marcincin, J. Barna, M. Janak, ja L. Novakova-Marcincinova, „Augmented Reality Aided Manufacturing“, *Procedia Comput. Sci.*, kd 25, lk 23–31, 2013.
- [14] „Industrial Augmented Reality Solutions for Manufacturing | PTC“. [Online]. Available at: <https://www.ptc.com/en/products/augmented-reality/solutions-for-manufacturing>. [Vaadatud: 19-apr-2019].
- [15] A. Doshi, R. T. Smith, B. H. Thomas, ja C. Bouras, „Use of projector based augmented reality to improve manual spot-welding precision and accuracy for automotive manufacturing“, *Int. J. Adv. Manuf. Technol.*, kd 89, nr 5–8, lk 1279–1293, märts 2017.
- [16] G. Michalos, P. Karagiannis, S. Makris, Ö. Tokçalar, ja G. Chryssolouris, „Augmented Reality (AR) Applications for Supporting Human-robot Interactive Cooperation“, *Procedia CIRP*, kd 41, lk 370–375, 2016.

- [17] C. Kollatsch, M. Schumann, P. Klimant, V. Wittstock, ja M. Putz, „Mobile Augmented Reality Based Monitoring of Assembly Lines“, *Procedia CIRP*, kd 23, lk 246–251, 2014.
- [18] J. A. Erkoyuncu, I. F. del Amo, M. Dalle Mura, R. Roy, ja G. Dini, „Improving efficiency of industrial maintenance with context aware adaptive authoring in augmented reality“, *CIRP Ann.*, kd 66, nr 1, lk 465–468, 2017.
- [19] „Overview“. [Online]. Available at: <https://library.vuforia.com/getting-started/overview.html>. [Vaadatud: 30-apr-2019].
- [20] M. Fiorentino, A. E. Uva, M. Gattullo, S. Debernardis, ja G. Monno, „Augmented reality on large screen for interactive maintenance instructions“, *Comput. Ind.*, kd 65, nr 2, lk 270–278, veebr 2014.
- [21] „10 Real Use Cases for Augmented Reality | Inc.com“. [Online]. Available at: <https://www.inc.com/james-paine/10-real-use-cases-for-augmented-reality.html>. [Vaadatud: 15-mai-2019].
- [22] „Microsoft HoloLens 2 launches at Mobile World Congress“. [Online]. Available at: <https://www.cnn.com/2019/02/24/microsoft-hololens-2-launches-at-mobile-world-congress.html>. [Vaadatud: 19-apr-2019].
- [23] „Google Glass Enterprise Edition review - augmented reality smartglasses“. [Online]. Available at: <https://www.aniwaa.com/product/vr-ar/google-glass-enterprise-edition/>. [Vaadatud: 19-apr-2019].
- [24] „The Daqri Smart Helmet – A closer look | Digital Reality Consulting“. [Online]. Available at: <http://digitalreality.guru/2017/04/the-daqri-smart-helmet-first-thoughts/>. [Vaadatud: 31-märts-2019].
- [25] „DAQRI Smart Glasses review - augmented reality display for professionals“, *Aniwaa*.
- [26] „HoloLens 2—Overview, Features, and Specs | Microsoft HoloLens“. [Online]. Available at: <https://www.microsoft.com/et-ee/hololens/hardware>. [Vaadatud: 04-apr-2019].
- [27] „glass-omap-xrr02 - kernel/omap - Git at Google“. [Online]. Available at: <https://android.googlesource.com/kernel/omap/+/-/glass-omap-xrr02>. [Vaadatud: 19-apr-2019].
- [28] „DAQRI showcases Ubuntu powered augmented reality helmet at MWC | Ubuntu blog“. [Online]. Available at: <https://blog.ubuntu.com/2017/02/21/daqri-showcases-ubuntu-powered-augmented-reality-helmet-at-mwc>. [Vaadatud: 08-mai-2019].
- [29] „10 Linux or Android Based Smart Eyewear Devices“, *Linux.com | The source for Linux information*, 28-juuli-2017. [Online]. Available at: <https://www.linux.com/news/10-linux-or-android-based-smart-eyewear-devices>. [Vaadatud: 09-mai-2019].
- [30] „DAQRI Smart Helmet Developer Edition“. [Online]. Available at: <http://donar.messe.de/exhibitor/hannovermesse/2017/U208979/dsh-data-sheet-eng-495327.pdf>. [Vaadatud: 08-mai-2019].
- [31] „Google Glass: \$1,500 to buy, \$80 to make?“ [Online]. Available at: <https://www.cbsnews.com/news/google-glass-1500-to-buy-80-to-make/>. [Vaadatud: 15-mai-2019].
- [32] „Glass“. [Online]. Available at: <https://www.x.company/glass/>. [Vaadatud: 05-apr-2019].
- [33] „Google Glass’s second-gen enterprise model gets updated specs and USB-C - The Verge“. [Online]. Available at: <https://www.theverge.com/circuitbreaker/2019/3/5/18251800/google-glass-second-gen-enterprise-model-leaks-usb-c>. [Vaadatud: 05-apr-2019].
- [34] „Professional Grade AR - DAQRI“. [Online]. Available at: <https://daqri.com/worksense/>.

- [Vaadatud: 31-märts-2019].
- [35] „Smart Glasses™ - DAQRI“. [Online]. Available at: <https://daqri.com/products/smart-glasses/>. [Vaadatud: 19-apr-2019].
- [36] „ROS/Introduction - ROS Wiki“. [Online]. Available at: <http://wiki.ros.org/ROS/Introduction>. [Vaadatud: 31-märts-2019].
- [37] „Metrics - ROS Wiki“. [Online]. Available at: <http://wiki.ros.org/Metrics>. [Vaadatud: 19-apr-2019].
- [38] „ROS.org | Powering the world's robots“. [Online]. Available at: <https://www.ros.org/>. [Vaadatud: 19-apr-2019].
- [39] „Gazebo“. [Online]. Available at: <http://gazebosim.org/>. [Vaadatud: 31-märts-2019].
- [40] „Gazebo : Tutorial : ROS overview“. [Online]. Available at: [http://gazebosim.org/tutorials?tut=ros\\_overview](http://gazebosim.org/tutorials?tut=ros_overview). [Vaadatud: 31-märts-2019].
- [41] „CppStyleGuide - ROS Wiki“. [Online]. Available at: <http://wiki.ros.org/CppStyleGuide>. [Vaadatud: 05-mai-2019].
- [42] „ROS/Concepts - ROS Wiki“. [Online]. Available at: <http://wiki.ros.org/ROS/Concepts>. [Vaadatud: 31-märts-2019].
- [43] M. Helmke, E. Joseph, ja J. A. Rey, „Customizing Ubuntu for Performance, Accessibility, and Fun“, 2016.
- [44] „usb\_cam - ROS Wiki“. [Online]. Available at: [http://wiki.ros.org/usb\\_cam](http://wiki.ros.org/usb_cam). [Vaadatud: 19-mai-2019].
- [45] „ar\_track\_alvar - ROS Wiki“. [Online]. Available at: [http://wiki.ros.org/ar\\_track\\_alvar](http://wiki.ros.org/ar_track_alvar). [Vaadatud: 19-mai-2019].
- [46] „tf - ROS Wiki“. [Online]. Available at: <http://wiki.ros.org/tf>. [Vaadatud: 19-mai-2019].
- [47] „gazebo\_ros - ROS Wiki“. [Online]. Available at: [http://wiki.ros.org/gazebo\\_ros](http://wiki.ros.org/gazebo_ros). [Vaadatud: 19-mai-2019].
- [48] „chrony – Introduction“. [Online]. Available at: <https://chrony.tuxfamily.org/index.html>. [Vaadatud: 18-mai-2019].
- [49] „chrony – Manual for version 2.3“. [Online]. Available at: <https://chrony.tuxfamily.org/manual.html#Typical-scenarios>. [Vaadatud: 18-mai-2019].

# Lisad

## Lisa 1. Mõõtmiste toorandmed

[https://docs.google.com/spreadsheets/d/14Xi-bkfnY32Cjwj6M\\_Y6p\\_CT1wioyaPCoem6fQ5qcBE/edit?usp=sharing](https://docs.google.com/spreadsheets/d/14Xi-bkfnY32Cjwj6M_Y6p_CT1wioyaPCoem6fQ5qcBE/edit?usp=sharing)

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

Mina, Kaarel Mark

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

## **"Liitreaalsus tootmisprotsessis asukohtade määramisel"**

mille juhendaja on Karl Kruusamäe

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'i 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.

Kaarel Mark

**20.05.2019**