

**TALLINNA TEHNIKAÜLIKOOL**

Infotehnoloogia teaduskond

Arvutiteaduse instituut

**Mitme Kinecti kasutamine pööramisliigutuse  
registreerimiseks**

**Magistritöö**

Üliõpilane: Helena Lissenko

Üliõpilaskood: 122307IAPMM

Juhendaja: PhD Sven Nõmm  
PhD Juhan-Peep Ernits

Tallinn  
2015

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

.....

*(kuupäev)*

.....

*(allkiri)*

# Sisukord

|                                                                                   |    |
|-----------------------------------------------------------------------------------|----|
| Sissejuhatus.....                                                                 | 8  |
| 1. Teoreetiline taust.....                                                        | 10 |
| 1.1 Liikumise registreerimine.....                                                | 10 |
| 1.1.1 Optiline liikumise registreerimine.....                                     | 10 |
| 1.1.1.1 Markeritega liikumise registreerimine.....                                | 10 |
| 1.1.1.2 Markeriteta liikumise registreerimine.....                                | 11 |
| 1.1.2 Mehaaniline liikumise registreerimine.....                                  | 12 |
| 1.1.3 Magnetiline liikumise registreerimine.....                                  | 12 |
| 1.2 Microsoft Kinect.....                                                         | 12 |
| 1.2.1 Riistvara.....                                                              | 13 |
| 1.2.2 Tarkvara.....                                                               | 14 |
| 1.2.3 Piirangud kasutamisel.....                                                  | 15 |
| 2. Valdkonnas eelnevalt tehtud tööd.....                                          | 17 |
| 2.1 Välismaal tehtud tööd.....                                                    | 17 |
| 2.2 Tallinna Tehnikaülikoolis tehtud tööd.....                                    | 20 |
| 3. Ülesande kontekst.....                                                         | 24 |
| 3.1 Kvantitatiivse analüüsi raamistik inimese motoorsete funktsioonide jaoks..... | 24 |
| 3.2 Ülesande püstitus.....                                                        | 25 |
| 3.2.1 Võimalikud probleemid ning piirangud.....                                   | 26 |
| 4. Implementatsioon.....                                                          | 28 |
| 4.1 Meetodi ülevaade.....                                                         | 28 |
| 4.1.1 Kabschi algoritm.....                                                       | 29 |
| 4.2 Tarkvaraline lahendus.....                                                    | 31 |
| 4.2.1 Pööramise salvestamine.....                                                 | 35 |
| 4.3 Ilmnunud raskused ja nende lahendused.....                                    | 37 |
| 5. Tulemused.....                                                                 | 40 |
| 5.1 Keskkonna ettevalmistamine.....                                               | 40 |
| 5.2 Katsete kirjeldus.....                                                        | 41 |

|                                                   |    |
|---------------------------------------------------|----|
| 5.3 Andmete töötlemine.....                       | 42 |
| 5.4 Tulemuste analüüs.....                        | 44 |
| 5.4.1 Koordinaatide vahe.....                     | 45 |
| 5.4.1.1 Peamise Kinecti poole kõndimine.....      | 46 |
| 5.4.1.2 Pööramine.....                            | 47 |
| 5.4.1.3 Peamisest Kinectist eemale kõndimine..... | 48 |
| 5.4.2 Liigutuste liikumismass.....                | 48 |
| 5.4.2.1 Peamise Kinecti poole kõndimine.....      | 49 |
| 5.4.2.2 Pööramine.....                            | 50 |
| 5.4.2.3 Peamisest Kinectist eemale kõndimine..... | 51 |
| 5.4.3 Järeldused.....                             | 52 |
| 5.5 Arengusuunad tulevikus.....                   | 53 |
| Kokkuvõte.....                                    | 55 |
| Summary.....                                      | 57 |
| Kasutatud materjalid.....                         | 59 |
| Lisa 1.....                                       | 62 |

# Joonised

|                                                                  |    |
|------------------------------------------------------------------|----|
| Joonis 1: Kinecti struktuur [19].....                            | 13 |
| Joonis 2: Kinecti koordinaatsüsteem ja Skeleti punktid [17]..... | 14 |
| Joonis 3: Kinecti kaugussätted [27].....                         | 15 |
| Joonis 4: Skeleti moonumine.....                                 | 16 |
| Joonis 5: Kolme erineva skeleti kokkupanemine [15].....          | 18 |
| Joonis 6: TTÜ arvutiteaduse instituudis tehtud tööd.....         | 20 |
| Joonis 7: Sensorite paigutus süsteemis.....                      | 28 |
| Joonis 8: Punktipilve teisendamine [6].....                      | 30 |
| Joonis 9: Programmi töökäik.....                                 | 33 |
| Joonis 10: Meetodi tööpõhimõte.....                              | 34 |
| Joonis 11: Sensorite paigutus.....                               | 35 |
| Joonis 12: UniqueConnectionId määramine App.config failis.....   | 37 |
| Joonis 13: Sensorite ja kaamerate paigutus.....                  | 40 |
| Joonis 14: Peamise Kinecti peale kleebitud markerid.....         | 43 |
| Joonis 15: Pea z-koordinaadi muutus salvestuse ajal.....         | 44 |
| Joonis 16: Kinecti ja MotionAnalysise inimkeha mudelid.....      | 45 |
| Joonis 17: Koordinaatide vahe Kinecti poole kõndides.....        | 46 |
| Joonis 18: Koordinaatide vahe pöörates.....                      | 47 |
| Joonis 19: Koordinaatide vahe Kinectist eemale kõndides.....     | 48 |
| Joonis 20: Tõuse ja kõnni testi sooritamine [12].....            | 62 |

# **Mitme Kinecti kasutamine pööramisliigutuse registreerimiseks**

## **Annotatsioon**

Käesoleva töö põhieesmärgiks on välja töötada meetod, mis võimaldaks kuni nelja Kinecti kasutamist inimese liikumise ning pööramise salvestamiseks. Lisaks sellele luuakse väljatöötatud meetodi tarkvaraline lahendus, mida on võimalik kasutada teegina teistes projektides. Valminud meetodi täpsust valideeritakse kasutades MotionAnalysis kaamerate süsteemi.

Töö oluliseks tulemuseks on loodud süsteemi täpsuse hinnang võrreldes võimsa MotionAnalysise kaamera süsteemiga. Lisaks sellele omandati praktiline kogemus mitut Kinecti kasutatava süsteemi loomisel.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 63-l leheküljel, 5 peatükki, 20 joonist ja 14 tabelit.

# **Multi-Kinect system for acquisition of turning motion**

## **Abstract**

The main goal of this thesis is to develop a method which allows the usage of up to four Kinect sensors to capture turning motion. In addition to that, an implementation of the developed method is written which can be used as a library in other similar projects. The accuracy of the developed method is validated by using a powerful motion capturing system called MotionAnalysis.

The most important result of the work is the assessment of the developed system's accuracy in comparison to the precise MotionAnalysis system. In addition to that practical knowledge of creating multi-Kinect systems was acquired.

The thesis is written in Estonian and contains 63 pages of text, 5 chapters, 20 figures and 14 tables.

## Sissejuhatus

Microsoft Kinecti sensor on kasutusel eelkõige meelelahtusliku seadmena. Oma odavuse ja kasutuslihtsuse tõttu aga kasutatakse seda aina enam teadustöö eesmärkidel. Võrreldes kallimate ning keerulisemate liikumise registreerimise süsteemidele, on Kinect odav ja laialt kättesaadav alternatiiv.

Sensori odavusega kaasnevad ka oma puudused. Nimelt kasutades ühte Kinecti ei ole võimalik salvestada pööramist ega inimese liikumist selja tagant, kuna see pole Microsofti arendustarkvara poolt toetatud. Lisaks sellele varju jäävate kehaosade salvestatid koordinaadid ei ole usaldusväärsed ning neid ei saa analüüsimiseks kasutada.

Tallinna Tehnikaülikooli arvutiteaduse instituudis on enne seda tööd valminud teise lõputöö tulemusena ühte Kinecti kasutatav programm, mille abil salvestatakse patsiendi liikumist hilisemaks analüüsiks. Patsientidel lastakse sooritada tõuse ja kõnni test, mis on sisuliselt edasi-tagasi kõndimine. Kasutades ainult ühte Kinecti ei ole võimalik pööramisel ja selja tagant salvestatud andmeid kasutada, kuna neid ei saa pidada tõetruudeks.

Antud töö eesmärgiks on välja töötada meetod, mis võimaldaks kuni nelja Kinecti kasutamist liikumise ning pööramise salvestamiseks. Väljatöötatud meetodi jaoks luuakse tarkvaraline lahendus, mida on tulevikus võimalik kasutada teegina teistes projektides. Püstitatud ülesannet raskendab see asjaolu, et kuigi mitme Kinecti kasutamine liikumiste salvestamiseks on tarkvara poolt toetatud, siis informatsiooni raskuste ja probleemide kohta ei üleliia palju.

Valminud meetodi täpsust valideeritakse kasutades täpset MotionAnalysis kaamerate süsteemi. Väljatöötatud meetodi abil salvestatud andmed ning MotionAnalysise süsteemi poolt salvestatud andmed võrreldakse omavahel. Selle tulemusena on võimalik anda hinnang väljatöötatud süsteemi täpsusele.

Lisaks püstitatud ülesandele saab töö tulemusena teada, kuidas tagada sujuva programmi töö, mis tegeleb kuni neljalt Kinectilt saadavate andmete töötlemisega ning kuidas toime tulla ilmnenud raskuste ja probleemidega.



Töö on jaotatud viieks peatükiks. Esimeses peatükis tutvustatakse liikumise registreerimise teoreetilist tausta. Samuti tutvustatakse Microsoft Kinecti koos selle tugevate ja nõrkade külgedega. Teise peatükis antakse ülevaadet nii Tallinna Tehnikaülikooli arvutiteaduse instutuudis kui ka välismaal tehtud töödest, milleks kasutati ühte või mitut Kinecti. Kolmandas peatükis esitatakse ülesande püstitus. Neljandas peatükis kirjeldatakse teooriat meetodi taga ning kuidas arendatud meetod implementeeriti. Viimases peatükis kirjeldatakse süsteemi täpsuse hindamiseks tehtud katsed, analüüsitakse tulemusi ning tehakse järeldusi.

# 1. Teoreetiline taust

## 1.1 Liikumise registreerimine

Liikumise registreerimine (*i.k. Motion capture*) võimaldab salvestada inimkeha või muu objekti liikumist digitaalsel kujul, mida hiljem saab arvuti abil töödelda. Objektideks võivad olla elus- ning elutud kehad, näoilmed, kaamera või valguse asukohad ja muud elemendid [25].

Liikumise registreerimise tehniline eesmärk on saada huvipunktide liikumise andmeid, mille põhjal saab liikumise parameetreid arvutada (nt. kiirus, suund, kaugus jne) või neid andmeid saab kasutada millegi juhtimiseks või manipuleerimiseks [38]. Meelelahutusvaldkonnas kasutatakse liikumise registreerimist selleks, et panna animeeritud 3D tegelased loomulikult liikuma

Liikumise registreerimiseks ning andmete salvestamiseks on olemas mitmeid erinevaid meetodeid, mida saab nende implementatsioonide põhjal jaotada klassidesse. Kõikide süsteemide abil salvestatud andmed töödeldakse hiljem tarkvara abil, mis on kas süsteemiga kaasas või seda on võimalik hiljem omandada. Ideaalset liikumise registreerimise süsteemi pole olemas, neil kõigil on omad tugevad ning nõrgad küljed nagu näiteks maksumus, ülesseadmise aeg, täpsus jne. Järgnevalt tutvustatakse levinumaid viise liikumise registreerimiseks ning tuuakse välja nende tugevad ja nõrgad küljed.

### 1.1.1 Optiline liikumise registreerimine

#### 1.1.1.1 Markeritega liikumise registreerimine

Objektile, mille liikumist tahetakse registreerida lisatakse peegeldavad või kiirgavad markerid (*i.k. Marker*), mida jälgitakse kaameratega. Markerid lisatakse nendele objekti punktidele, mille liikumise registreerimine kõige rohkem huvi pakub ning nad kas peegeldavad valgust tagasi või kiirgavad infrapunavalgust, mida süsteem salvestab. Markerite abil on võimalik luua objekti või inimese mudel, mida seda meetodit kasutav süsteem automaatselt tuvastab. See võimaldab ka tuvastada korraga mitu objekti või inimesti.

Selle meetodi kasutamine on äärmiselt täpne ning paindlik, kuna kasutaja saab ise paika panna, millistele kohtadele markerid lisatakse. Lisaks sellele pole jälgitava objekti liikumine kuidagi takistatud, kuna markerid ei ole masinaga otseses ühenduses näiteks juhtmete kaudu. Üheks probleemiks võib küll olla see, kui marker on objekti külge halvasti ühendatud, mis võib põhjustada selle lahti tulemise või nihkumise. Sellisel juhul tuleks liikumise salvestamist alusata otsast peale vältimaks suuri ebatäpsusi andmetes.

Selle meetodi kasutamisel on suureks probleemiks valguse häiriv mõju, mis võib muuta mõõtmistulemused ebatäpsuks. Kui markerid pole kaamerale nähtavad, siis vastava punkti liikumise registreerimine muutub ebatäpsuks ning selle asemel kasutatakse hinnangulist asukohta, mis ei pruugi enam olla piisavalt täpne. Markeritega süsteemi ülesseadmine ning kalibreerimine võib olla keeruline ning kulukas nii raha kui ka aja suhtes [25].

### **1.1.1.2 Markeriteta liikumise registreerimine**

Erinevalt eelmisest meetodist toimub liikumise registreerimine sensorite ja kaamerate abil ning objekti eelnev ettevalmistus (nt. markerite asetamine) pole vajalik.

Selle meetodi eeliseks on markerite puudumine, mis teeb töö oluliselt lihtsamaks. Täpsuse suurendamiseks tuleb kasutatavate kaamerate ja/või sensorite arvu suurendada, et vähendada ohtu, kus mingid objekti osad ei ole näha. Kasutuses olevate kaamerate ja/või sensorite arvu suurendamine muudab antud meetodi peaaegu sama täpsuks nagu markeritega liikumise registreerimise.

Markerite puudumine piirab selle meetodi võimeid. Ainult need objektid saavad olla jälgitavad, mida antud süsteem on programmeeritud registreerima (kui süsteem on programmeeritud inimese liikumise registreerimiseks, siis näiteks loomade liikumist sellega jälgida ei saa). Seetõttu on autonoomne objektide jälgimine raskendatud. Süsteemi täpsuse suurendamine kaamera ja/või sensorite arvu suurendamise abil võib osutuda kalliks. Kõik kaamerad ja sensorid vajavad eelnevat kalibreerimist ning seadistamist. Nagu ka markeritega liikumise registreerimise puhul on liigeste nähtavuse kadumine kaamerate jaoks suureks probleemiks, mis vähendab oluliselt täpsust [9].

### **1.1.2 Mehaaniline liikumise registreerimine**

Mehaanilise liikumise registreerimise puhul peab objekt (tavaliselt inimene) kandma vastavat mehhanismi sisaldavat riietust, mida nimetatakse „ekso-skeletiks“. Selle meetodi kasutamine võimaldab jälgida liigeste liikumist ning jäsemete vahelisi nurki.

Selle meetodi eeliseks on selle suhteline odavus. Samuti ei mõjuta meetodi töövõimet ei valgus ega magnetväli. Lisaks sellele pole jälgitavate punktide nähtavus probleemiks, kuna nende liikumise registreerimine toimub mehaaniliselt, mitte kaamerate abil.

Suurimaks puuduseks on ekso-skeleti kandmisest tingitud liikumise piiramine (kui objektiks on inimene, siis kindlasti tekitab ka ebamugavust). Jälgitava objekti asukoht vahetus keskkonnas ei ole samuti automaatselt teada [26].

### **1.1.3 Magnetiline liikumise registreerimine**

Jälgitava objekti küljes on sensorid, mis mõõdavad madalsageduslikku magnetvälja, mida tekitab staatiline saatja. Sensorid ja saatja on ühendatud elektrilise juhtseadmega, mis arvutab nende asukohad magnetväljas. Sensorid annavad infot asukoha ja rotatsioonide kohta, seejärel kasutatakse pöörd-kinemaatikat (*i.k. Inverse kinematics*), et kätte saada liigeste vahelised nurgad [23].

Selle meetodi eeliseks on võimalus kindlaks teha jälgitava objekti asukoht. Lisaks pole jälgitavate punktide nähtavus oluline. Võrreldes markereid kasutatavate meetoditega on magnetilise liikumise registreerimise kasutamine oluliselt odavam.

Samas on magnetilisel liikumise registreerimisel ka mitu puudust. Antud meetodi kasutamist mõjutab tugevalt teiste magnetväljade poolt tekitatav interferents. Saadavad andmed võivad olla müra-rikkad ning sensorite ja saatja vahekauguse suurenemisel hakkab signaalis tekkima moonutus. Lisaks on objektide liikumine piiratud, kuna nad on juhtmetega ühendatud arvutiga andmete edastamiseks [25].

## **1.2 Microsoft Kinect**

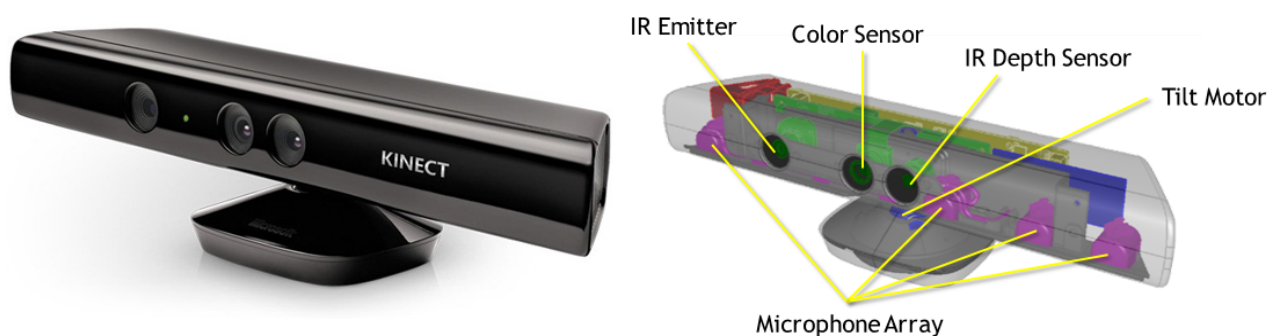
Microsoft Kinect Xboxi jaoks tuli turule 2010. aasta novembris ning siis muutis see täielikult videomängude konsooliga suhtlemise elamust. Kinect võimaldab mängijatel

suhelda konsooliga kasutades ainult oma keha või häälkäsklusi. Mängimise ajal võimaldab Kinect otseselt juhtida oma tegelast kasutades žeste [11].

Alates 2011. aastast on saadaval Microsofti ametlik arendustarkvara, mis võimaldab kasutaja jälgimist ja tema liigutuste tuvastamist kolmemõõtmelises ruumis. Kinecti odav hind, võrreldes teiste sarnaseid võimalusi pakkuvate seadmetega, on teinud ta populaarseks teaduse valdkonnas, eriti nendes suundades, kus tuleb reaajas liikumist registreerida ning hiljem analüüsida [31].

### 1.2.1 Riistvara

Kinecti peamine funktsionaalsus tuleneb värvikaamera, infrapuna valguse projektori, infrapuna valguse vastuvõtja ning nelja ritta seatud mikrofoni kombinatsioonist (vt. Joonis 1) [11].



Joonis 1: Kinecti struktuur [19]

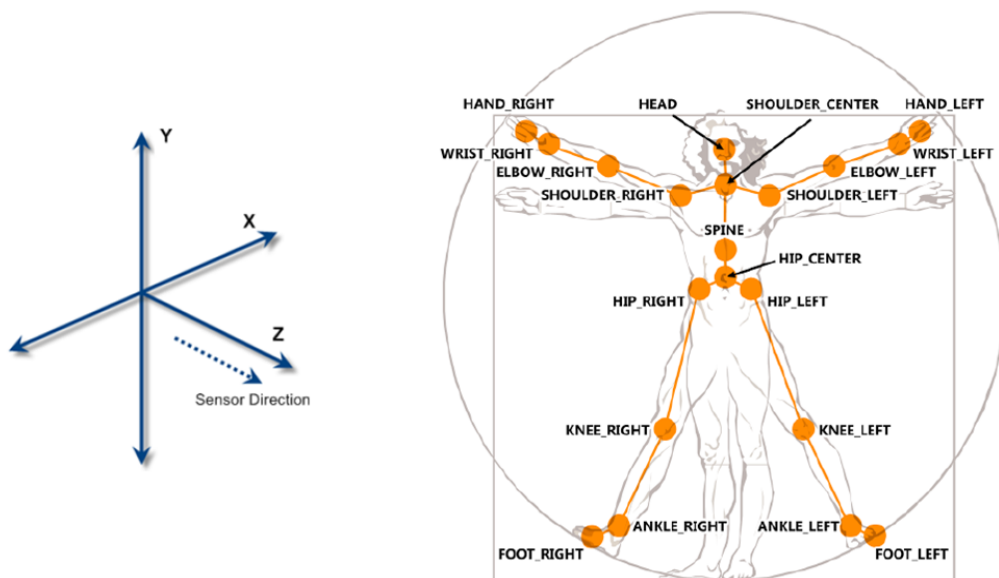
RGB kaamera (vt. Joonis 1 - *Color sensor*) võimaldab salvestada värvilist pilti eraldusvõimega kuni 1280x960, nii videona (sagedusega 30 kaadrit sekundis) kui ka fotodena. Samuti saab videostriimi pruukida näotuvastamisel. Fikseeritud mustrit projitseeriv infrapuna saatja (vt. Joonis 1 - *IR emitter*) kiirgab infrapunavalgust ning infrapuna kaamera (vt. Joonis 1 - *IR depth sensor*) võtab vastu tagasipeegeldunud valgust. Projitseeritud mustrit tagasipeegeldunud valguse põhjal suudab Kinect tuvastada kui kaugel asuvad objektid. Kuna infrapunakaamera eraldusvõime on vaid kuni 640x480, siis ka RGB kaamera puhul on kasutusel sama eraldusvõime ning mitte kaamera maksimaalne eraldusvõime.

Lisaks nägemise aspektile on Kinect võimeline salvestama ka häält tänu neljale mikrofonile, mis on seatud ritta (vt. Joonis 1 - *Microphone array*). Tänu sellele on Kinect

võimeline tuvastama, kus asub hääle allikas ning eraldama taustamüra [19]. Antud lõputöö käigus ei kasutata Kinecti poolt pakutava heli-komponendi võimalusi.

### 1.2.2 Tarkvara

Kasutades Kinect for Windows SDK'd on arendajatel võimalik saada ligipääs kõikidele andmetele, mida sensor registreerib. Juurdepääs töötlemata sügavusandmetele ning värvivideole võimaldab arendajatel kindlaks määrata kindla punkti värvi või kaugust sensorist. Tarkvaral on samuti olemas sisseehitatud algeline skeleti jälgimise võimalus. On olemas 20 eelnevalt defineeritud punkti ehk liigest (*i.k. Joint*) inimkehal, mille ühendamisel tekib jälgitava inimese kujutis kriipsujuku näol (vt. Joonis 2). Kinect for Windows SDK abil on võimalik saada informatsiooni iga selle kujutise 20 punkti asukoha, rotatsioonist ning liigi kohta.



Joonis 2: Kinecti koordinaatsüsteem ja Skeleti punktid [17]

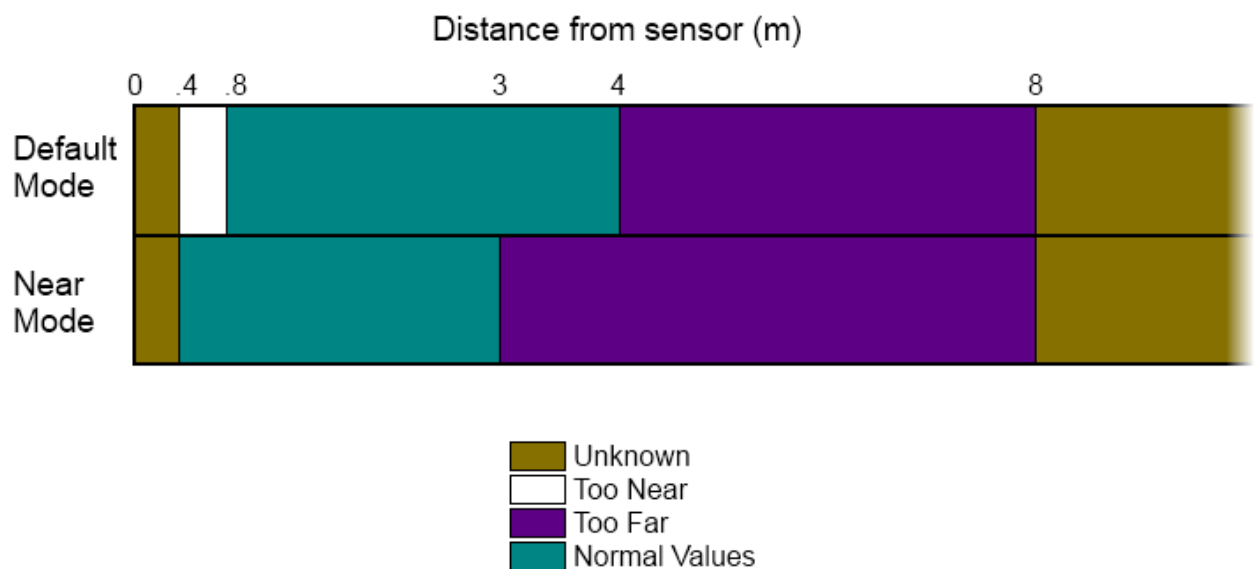
Antud SDK'l on olnud mitu versiooni, milles on järjest võimaldatud uute võimaluste kasutamist. Antud lõputöö kirjutamisel kasutatakse versiooni 1.8. Selle versiooniga kaasnenud uuendus on tausta eemaldamine, mis võimaldab välja valida peamist jälgitavat objekti ning ülejäänud värvipikslid eemaldada [18].

2014. aasta keskel tuli välja Kinect versioon 2. Koos sellega ilmus ka uus arendustarkvara – Kinect for Windows SDK 2.0. Uus arendustarkvara annab rohkem

võimalusi inimekeha liikumise jälgimiseks, kuid kahjuks see ei tööta vanemate Kinectidega [20].

### 1.2.3 Piirangud kasutamisel

Kinecti üks peamisi nõrkusi on sensorite vaatevälja piiratus. Alates 2012.-st aastast on Kinectil olemas kaks režiimi – tavarežiim (*i.k. Default mode*) ja lähirežiim (*i.k. Near mode*). Lähirežiim võimaldab vähendada kaugust, millelt Kinect saab usaldusväärseid andmeid (vt. Joonis 3). Enne seda peeti kõige sobivamaks andmete kogumise kauguseks 0.8 – 4 meetrit, lähivaates on selleks kauguseks 0.4 – 3 meetrit. Lähivaate funktsionaalsuse lisamine aga ei eemaldanud pimedat kohta, mis asub sensorile lähemal kui 40cm, samuti mõlemas vaates ei ole Kinect võimeline koguma andmeid kaugemalt kui 8 meetrit. Need asjaolud piiravad Kinecti kasutusalasid [27].

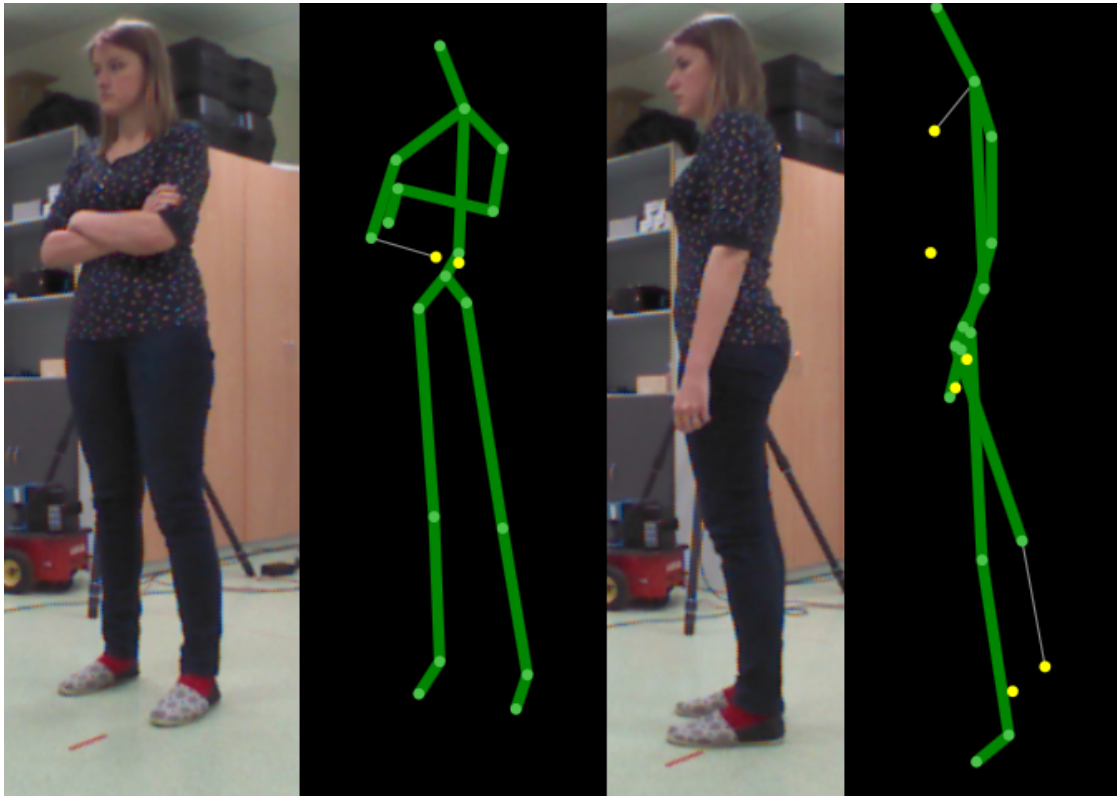


Joonis 3: Kinecti kaugussätted [27]

Kinecti sügavussensor töötab struktureeritud infrapuna valguse põhjal, seetõttu igasugune infrapunavalguse poolt tekitatud interferents muudab salvestatud andmed ebatäpseks. Seeõttu pole Kinecti võimalik kasutada päikesevalguse käes [30].

Kinectil esineb probleeme ka objektide jälgimisega, mis ei mahu nende vaatevälja, näiteks liiga madalal või liiga kõrgel asuvad objektid. Samuti skeleti jälgimisel on probleemiks sensorite vaateväljast jäävate punktide jälgimine. Näiteks, kui jälgida

inimest sensorite abil ning ta paneb käed ees risti või seisab küljega sensor poole, siis skelett moondub (vt. Joonis 4).



**Joonis 4: Skeleti moondumine**

Kinect for Windows SDK 1.8 ei võimalda inimese selja tagant jälgimist. Juhul kui inimene on ennast sensori poole seljaga pööranud, käsitletakse seda olukorda kui sensori poole näoga seismist [13]. See ning erinevate vaatlusepunktide varju jäämine põhjustab suurt ebatäpsust, kui proovida salvestada inimese pööramist.



## 2. Valdkonnas eelnevalt tehtud tööd

Antud peatükis antakse ülevaade Tallinna Tehnikaülikooli arvutiteaduse instituudis tehtud töödest, milles kasutati Kinecti ja/või MotionAnalysis süsteemi. Samuti tehakse lühiülevaadet välismaal tehtud töödest, milles kasutati objekti jälgimiseks ning liikumise registreerimiseks mitut Kinecti. Mitme Kinecti kasutamine võimaldab vähendada vigu, mis tekivad kui kehajäsemed ei ole sensori vaateväljas.

### 2.1 Välismaal tehtud tööd

[1] artiklis oli autorite eesmärgiks töötada välja süsteem, mis võimaldaks rekonstrueerida inimkehasid 3D-s, liites kokku mitmelt Kinecti sügavuskaameralt saadud andmed. Valminud süsteemi saaks kasutada näiteks telekonverentsi programmis, mis võimaldaks eri paikades asuvatel inimestel omavahel paremini suhelda. On olemas mitmeid kirjeldusi täpsetest 3D kehade rekonstrueerimissüsteemidest, kuid neid ei saa pika töötlusaja tõttu reaalajas kasutada. Antud artikli töö tulemuseks on süsteem, mis koosneb mitmest Kinectist ja suudab produtseerida reaalajas 3D kujutist inimese kehist.

Süsteem koosneb viiest Kinectist, mis on ühenduses ühe arvutiga. Üks Kinect on asetatud horisontaalselt kõrgusele 1.3 meetrit, et jälgida ülakeha eespoolt. Ülejäänud Kinectid on asetatud vertikaalselt kõrgusele 1.8 meetrit ning nad on kallutatud pööranda poole 25-kraadise nurga all. Kõik Kinectid asetsevad ringil diameetriga 3.6 meetrit ning on suunatud ringi keskpunkti.

Üksiku Kinecti kalibreerimiseks kasutatakse meetodit, mille algoritmi on välja töötanud [10]. See algoritm hindab samaaegselt sügavus- ja RGB kaamera sisemisi parameetreid ning nende vahelist suhtelist asukohta. Lisaks hinnatakse sügavussensori mõõtmiste moonutust. Artiklis kirjeldatud töö tulemusena valmis süsteem, mille abil saab reaalajas rekonstrueerida võrdlemisi realistlikke inimkeha 3D mudeleid.

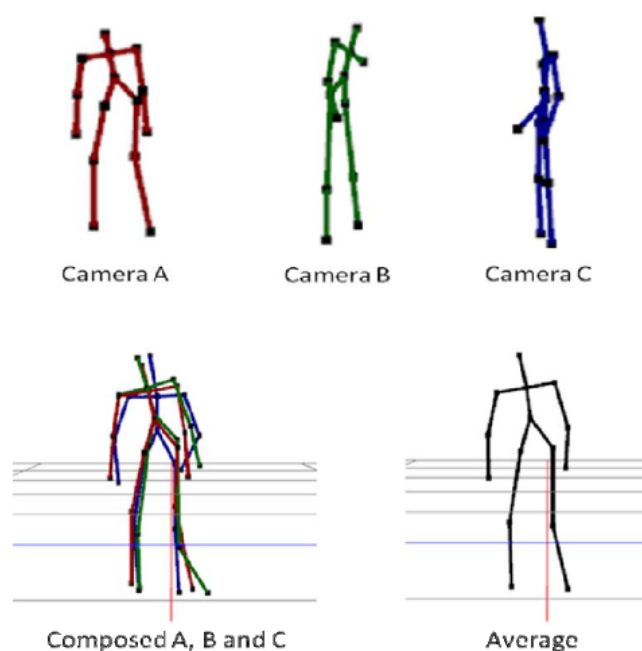
[15] võtsid kokku tööd, milles liideti mitmelt Kinectilt korraga saadud skeleti kujutised kehahoiakute analüüsimise eesmärgiga. Selleks, et kokku liita erikaameratelt saadud andmed tuli teisendada kaamerate kohalikud koordinaatsüsteemid ühte globaalsesse koordinaatsüsteemi. Lõplik skelett saadakse kõikide teisendatud skelettide keskmise arvutamise teel.

Koordinaatsüsteemi transformatsiooni kasutatakse selleks, et teisendada ühte koordinaatsüsteemi teise soovitud süsteemi. Selline koordinaatsüsteemide teisendamine globaalsesse süsteemi lihtsustab objektidega ümberkäimist. Transformatsiooni kirjeldab järgmine valem,

$$Q' = M \cdot Q$$

milles Q on koordinaatide maatriks enne transformatsiooni- maatriksiga M korrutamist, tulemuseks on uute koordinaatide maatriks Q'.

Ühe kindla Kinect sensori lokaalne koordinaatsüsteem valitakse globaalseks ning kõik andmed, mida saadakse teistelt sensoritelt teisendakse sellesse globaalsesse süsteemi. Pärast skelettide liigete koordinaatpunktide teisendust globaalsesse süsteemi on olemas kolm erinevat liigete komplekti. Need kolm koordinaatide komplekti koondatakse kesmise põhjal ühte skeletti, mida hilisemas analüüsis kasutatakse (vt. Joonis 5). Selline skeleti jälgimise lähenemine välistab olukorra, kus mõned kehaosad ei asu sensori vaateväljas.



Joonis 5: Kolme erineva skeleti kokkupanemine [15]

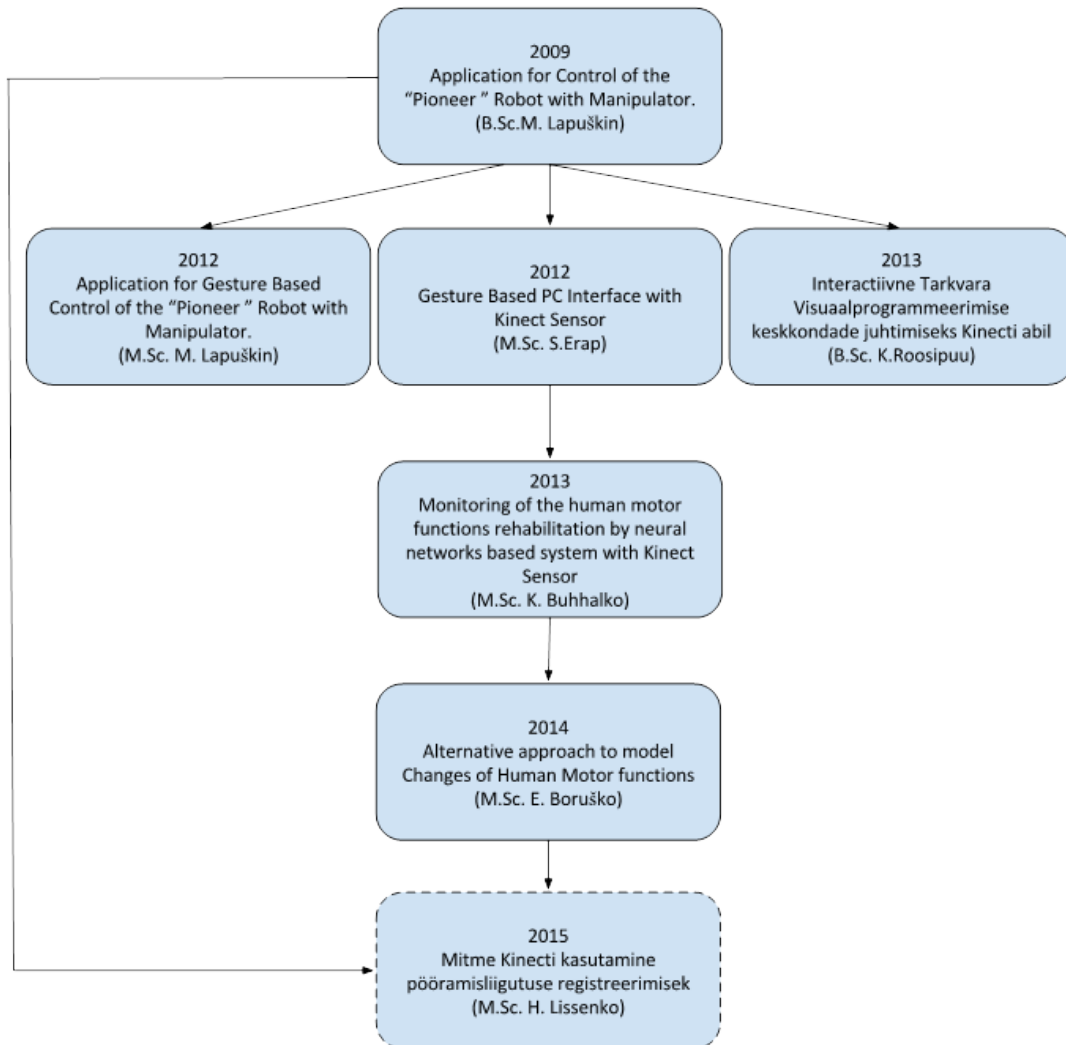
[32] eesmärgiks oli analüüsida mitme Kinecti kasutamist kasutajate jälgimiseks virtuaalses reaalsuses. Täpsemalt analüüsitakse Kinecti võimet jälgida infrapunavalguse punkte, mida kasutatakse virtuaalse reaalsuse valdkonnas. Artikkel kirjeldab Kinecti võimetele jälgida infrapunapunkte 7 x 9 meetrises ruumis, keskendutakse asukoha täpsusele, mida saadakse ning kas mitme Kinecti kasutamine põhjustab suurt interferentsi. Mitme Kinecti kasutamine objektide jälgimiseks võib potentsiaalselt suurendada nii jälgimisala kui ka täpsust. Antud artiklis keskenduti spetsiifiliselt sellele, kui tundlik on Kinect teiste Kinectide suhtes, st. kas mitme Kinecti kasutamisel tekivad interferentsid tekitab probleeme või mitte.

Eksperimendi käigus kasutati viite Kinecti, mida asetati sama kõrgusega toolidele. Ühte nendest loeti primaarseks Kinectiks ning ülejäänud tekitasid interferentsi. Tööalas oli 30 punkti, mis olid meetri kaugusel teineteisest. Igaühe kohta koguti andmeid neid asukoha kohta. Töö tulemusena järeldati, et Kinectiga objektide jälgimisel pole interferents suureks probleemiks ning mitme Kinecti kasutamine võimaldab luua odavamaid süsteeme kui need, mis on hetkel kasutusel.

[37] eesmärgiks oli valmis teha süsteem, mis vähendaks skeleti liigeste asukoha ebatäpsusi võrreldes ainult ühte Kinecti kasutamisega. Selleks kasutatakse kahte sensorit, mis näevad jälgitavat inimest erinevate nurkade alt ning seetõttu on võimalik konstrueerida täielik skelett.

Töö tulemusena valmis raamistik, mis võimaldab paigutada kahte Kinecti sensorit sellisel viisil, et salvestatavad andmed on palju täpsemad kui ühe sensori puhul. Kasutuses olevad sensorid paigutatakse teineteise suhtes ortogonaalselt. Microsoft Kinect arendustarkvara abil saadakse mõlemalt sensorilt skeleti andmed ning pärast arendatud algoritmi rakendamist saadakse kahest skeletist kokku pandud täpsem skelett. Samuti valmis meetod, mis võimaldab hinnata skeleti asukoha usaldusväärsust.

## 2.2 Tallinna Tehnikaülikoolis tehtud tööd



Joonis 6: TTÜ arvutiteaduse instituudis tehtud tööd

Joonisel 6 on näha TTÜ arvutiteaduse instituudis eelnevalt tehtud tööd, millest antud lõputöö teema välja kasvas. Kõik eelnevad tööd on kuidagi seotud inimkeha liikumise registreerimisega.

[22] on osa suuremast projektist, mille eesmärgiks oli luua süsteem, mis võimaldaks inimestel saata robotile mitteverbaalseid käske kasutades kaardikeppi. Töö tegemisel kasutati võimsat MotionAnalysis kaamerate süsteemi, mis võimaldas jälgida objekte

nende külge kinnitatud markerite abil – antud juhul kinnitati markerid kaardikepi külge. Samuti kasutati robotit Pioneer 3 DX, millele käske edastati.

Bakalaureusetöö eesmärgiks oli luua rakendus, mille ülesandeks on analüüsida objekti liikumist, kasutades andmeid, mis kirjeldavad reaajas tema asukohta. Analüüs seisneb teatud liikumise mustrite avastamises. Iga rakendusse sisse programmeeritud liikumise mustrile vastab oma signaal, mida rakendus tõlgendab käsuks ja saadab robotile täitmiseks. Töö tulemusena valmis rakendus, mille abil saab juhtida robotit kasutades kaardikeppi.

[21] oli sama autori poolt kirjutatud magistritöö, mis oli [22] loogiliseks jätkuks. Töö eesmärgiks oli luua süsteem, mis võimaldaks inimesel juhtida robotit žestide abil. Töö tegemisel kasutati võimsat MotionAnalysis kaamerate süsteemis, mis võimaldas jälgida objekte nende külge kinnitatud markerite abil. Antud juhul kinnitati markerid tavalise kinda peale, et ta oleks jälgitav kaamerate süsteemi poolt, mis edastab reaajas markerite asukoha koordinaadid žestide tuvastamiseks. Samuti kasutati robotit Pioneer 3DX, millele käske edastati.

Magistritöö eesmärgiks oli välja töötada ja realiseerida žestide tuvastamise algoritm, mille abil saab robotile käske edastada. Töö tulemusena realiseeriti kaks žestide tuvastamise algoritmi.

[31] keskendus juba Kinect for Windows sensorile. Bakalaureusetöö eesmärgiks oli projekteerida ja realiseerida rakendus, mille abil on võimalik kasutades Kinect sensorit ning Kinect for Windows arendustarkvara käeliigutuste ja viibete tuvastamiseks, interaktiivselt juhtida SciLab visuaalprogrammeerimise keskkonda Xcos. Kasutaja paikneb sensori poole seljaga (ruumi asetuse tõttu), mis tekitas probleeme, kuna Kinect on eelkõige mõeldud kasutamiseks kui inimene on selle poole näoga. Töö tulemusena valmis Kinect sensorit kasutav rakendus Windows operatsioonisüsteemile, mis võimaldab žestide abil kasutada SciLab tarkvara visuaalprogrammeerimise keskkonda Xcos. Lisaks sellele on võimalik rakendust kasutada ka muude Windowsi töölaua rakendustega.

[5] töö kuulub inimese ja masina vahelise koostoime valdkonda. Magistritöö eesmärgiks oli luua puutevaba kasutajaliides kasutades Kinect sensorit, mis kasutas žestidel põhinevat kasutajaliidest. Sellisel juhul saab kasutaja suhelda arvutiga, mille küljes on Kinect sensor, ilma füüsilise kontaktita. Töö tulemusena valmis kolm eraldi rakendust. Esimene nendest võimaldas vahetada slide Powerpoint ettekande ajal kasutades žeste. Teine rakendus juhtis sama robotit, mida kasutati [22] ja [21], kasutades Kinect sensorit. Viimane programm oli lihtne rakendus, mis salvestab Kinecti arendustarkvara poolt eelnevalt defineeritud liigete koordinaadid faili. Neid salvestatud andmeid saab kasutada edasises teadustöös, mis hõlmab endas inimese liikumise analüüsimist ning žestide tuvastamist.

[4] eesmärgiks oli arendada süsteem rehabilitatsiooni jaoks, mis kasutab ühte Kinect sensorit. Süsteem jälgib, kuidas inimene sooritab taastusravi harjutusi ja hindab nende õigsust. Selline lähenemine vähendab vajadust spetsialisti vahetu järelevalve järele ning võimaldab inimestel teha taastusravi harjutusi kodus.

Magistritöö tulemusena valmis raamistik taastusravi harjutuste sooritamise jälgimiseks, mis kasutab mudelite loomiseks närvivõrke (*i.k. neural network*). Inimese jäsemete liikumise andmeid salvestatakse Kinecti abiga. Eelnevalt luuakse närvivõrgu abil mudel, mis kirjeldab õigest harjutuse liikumist. Seda mudelit kasutatakse jäsemete asukoha ennustamiseks harjutuse sooritamise ajal. Kui tegelik ja ennustatud asukoha vahe ületab kindla lävendi, palutakse inimesel harjutus korrata.

[3] on pühendatud inimese motoorsete funktsioonide olekute mõõtmisega ja modelleerimisega. Psühholoogia või meditsiini valdkondade uuringud vajavad teatud kognitiivsete tegevustega või raviprotseduuridega kaasneva motoorika muutuste mõõtmist. Antud töö pakub välja lahenduse, mis annab laialdase võimluse pakutud meetodi kasutamiseks nii teadusuuringute kui ka ravi eesmärgil.

Magistritöö tulemusena valmis rakendus, mis kasutab Kinect sensorit ning seda saab jooksutada keskklassi sülearvutitel. Rakendus võimaldab salvestada liikumise andmeid, välja arvutada parameetrid, mis kirjeldavad liigutuste hulka ja sujuvust, mis on seotud teatud liiki liigutustega. Samuti on selle abiga võimalik tuvastada mõningaid žeste

ning salvestada andmeid hilisemaks analüüsiks. Valminud rakendust kasutatakse Tallinna Ülikooli ja Tartu Ülikooli uuringute läbi viimisel.

### **3. Ülesande kontekst**

Selles peatükis kirjeldatakse projekti, mille raames antud lõputööd kirjutati. Antud töö oli toetaud TTÜ poolt basfinatseerimise teema B37 kaudu. Kirjeldatakse, milliseid eesmärke töö tulemusena täidetakse ning räägitakse lühidalt teistest alamprojektidest.

#### ***3.1 Kvantitatiivse analüüsi raamistik inimese motoorsete funktsioonide jaoks***

Antud lõputöö tehti Tallinna Tehnikaülikooli arvutiteaduse instituudis. Arvutiteaduse instituudis tehtava teadustöö üks suundadest keskendub inimese motoorika funktsioonide analüüsimise meetodite parandamisele. Instituudis tehtava töö lähenemisviis erineb teistest samas valdkonnas tehtud töödest. Peamine erinevus seisneb selles, et diagnostikat ja jälgimist peetakse laiema raamistiku osaks, milles inimese ja masina vaheline koostoime toimub nii patsiendi ja seadme vahel kui ka seadme ja arsti vahel.

Projekti eesmärgiks on arendada uus matemaatiline raamistik, mille abil oleks võimalik mitmetasandiline inimese motoorsete funktsioonide analüüs. Analüüs toimub kahel tasemel: madalamal tasemel toimub kindlate liikumiste ja žestide analüüs ning kõrgemal tasemel analüüsitakse motoorsete funktsioonide seisundit, mida on modelleeritud pikema aja jooksul. Mõlemalt tasemelt saadud tulemused ei käsitleta eraldi. Kõrgemalt tasemelt saadud tulemused kasutatakse modelleerimismeetodite parametriseerimiseks madalamal tasemel ning saadud tulemuste käsitlemiseks.

Valmivat raamistikku saab kasutada neurofüsioloogia ning psühholoogia valdkondades. Madalama taseme jäsemete modelleerimine ning žestide tuvastamine on vajalik terapeutiliste ülesannete ning diagnostika järelvalveks. Inimese motoorsete funktsioonide seisundite muutus on oma olemuselt sarnane, olenemata sellest kas tegemist on rehabilitatsiooniga pärast traumat või insulti, süveneva Parkinsoni haigusega või uue motoorse tegevuse ära õppimisega. Toimuvate muutuste protsess on aga tavaliselt mittelineaarne ning mõningatel juhtudel ka mitte-monotoonne.

Mitmed akadeemilised ja kaubanduslikud organisatsioonid on teatanud keerulistest tulemustest diagnostika ning taastusravi valdkonnas. Vaatamata sellele,



praegu kasutusel oleval lähenemisviisil on mitu puudust, mida TTÜ arvutiteaduse instituudis loodav raamistik loodab parandada:

- Suurem osa lähenemisviise keskendub ainult patsiendi ja masina vahelisele koostoimele, mis isoleerib täielikult arsti järelvalvet taastusprotsessist.
- Turul olemasolevatest toodetest nagu näiteks Virtual Reality Kinect Rehabilitation või Nike+ Kinect training, ei sobi ükski kliinilisse kasutusse või akadeemilise teadustöö jaoks, kuna nad on mõeldud vanemate inimeste või spordi tegemise jaoks.
- Lahendused, mis on Kinecti-põhised, kasutavad ainult ühte sensorit. Seetõttu on need süsteemid ebatäpsed ning ei võimalda keerilisemate liikumiste analüüsi.

Arendatav raamistik peaks nendest puudustest üle saama. Raamistiku valideerimine toimub kasutades suure täpsusega liikumise registreerimise süsteemi "Motion Analysis". See süsteem koosneb seitsmest kaamerast, lõplikud lahendused peaksid võimaldama odavamate sensorite kasutamist nagu näiteks Kinect.

Enne antud lõputööd valmis TTÜ arvutiteaduse instituudis töö [3], milles kasutati patsientide liikumise salvestamiseks ja analüüsimiseks ühte Kinecti sensorit. Ühte Kinecti kasutaval süsteemil on suurimaks puuduseks ebatäpsus. Liikumise salvestamisel lähevad kaotsi andmed nende jäsemete kohta, mis on sensori vaateväljast väljas. Samuti tekib probleeme ning palju müra, kui patsient on sensori poole seljaga. Üheks süsteemi kasutamise eesmärgiks on tõuse ja kõnni testi läbi viimine. Kahjuks süsteem, mis kasutab ainult ühte Kinecti ei ole võimeline salvestama inimese pööramist ning selja tagant saadud andmed on äärmiselt ebatäpsed.

### ***3.2 Ülesande püstitus***

Ühe Kinecti kasutamisest tulenevat ebatäpsust saab lahendada, kui võtta kasutusele mitu sensorit ning paigutada nad ümber testimise ala. Sellisel juhul vähemalt üks Kinect näeb salvestatavat skeletti täielikult ja eestpoolt.

Antud lõputöö eesmärkideks on:

1. Välja töötada meetod, mis võimaldaks kuni nelja Kinecti kasutamist liikumise ning pööramise salvestamiseks.

2. Luua meetodi takrvaraline lahendus, mis võimaldaks kasutada seda teegina teistes projektides.
3. Valideerida meetodi täpsust kasutades MotionAnalysis kaamerate süsteemi.

### 3.2.1 Võimalikud probleemid ning piirangud

Püstitatud ülesande lahendamisel võib ette tulla mitmeid probleeme ning piiranguid, mis teevad selle lahendamise keeruliseks. Antud alapeatükis räägitakse nendest lähemalt.

Üheks peamiseks probleemiks on asjaolu, et rakendusi, mis kasutavad mitut Kinecti pole väga palju kirjutatud, ning kui on, siis nende kohta on vähe informatsiooni. Microsofti ametlikus dokumentatsioonis on kirjutatud, et skeleti jälgimine on toetatud kuni 4 Kinecti puhul, kuid interferentsi tõttu muutub see ebatäpseks. Samuti skelettide ID-sid ei sünkroniseerita üle kõikide ühendatud Kinectide, mis tähendab seda, et mitme inimese jälgimise puhul ei saa olla kindel, et sensoritelt saadud andmed käivad samade skelettide kohta [34].

Kinect kasutab skelettide jälgimiseks sisseehitatud infrapunavalguse projektorit ning infrapuna valguse vastuvõtjat. Kui kasutatakse mitut sensorit, siis ühte ja samasse alasse kiiratakse infrapuna valgust mitmest allikast. See võib põhjustada interferentsi ning vähendada süsteemi täpsust. Loodava süsteemi täpsusest ning tulemustest räägitakse lähemalt 5.-s peatükis.

Sensori kõige sobivamaks andmete kogumise kauguseks on 0.8 – 4 meetrit. Seetõttu ei saa jälgitav objekt olla sensoritele liiga lähedal ega liiga kaugel nendest. Antud töö eesmärgiks on võimaldada Up-and-Go testi salvestamist, mis eeldab seda, et inimene kõnnib, mitte ei seisa kohapeal. See tähendab, et testimise ala muutub võrdlemisi suureks ning piklikuks. On oht, et võivad tekkida kohad, kus ükski sensor ei näe jälgitavat inimest piisavalt hästi.

Microsoft Kinect on võimeline jälgima objekte kiirusega 30 kaadrit sekundis. Sellest tingituna vajab üks sensor vähemalt 50% USB ribalaiusest ning seetõttu soovitatakse, et ühe USB kontrolleri külge ei ole midagi muud ühendatud peale selle ühe Kinecti [7]. Antud projektis aga kasutatakse kuni nelja sensorit, mis tähendab seda, et läheb vaja veidi võimsamat arvutit mitme USB kontrolleriaga. Vastasel juhul võib juhtuda nii, et arvuti koormatakse üle ning programm ei lähe üldse käima.

Sensoriga kaasas olev USB kaabel on antud ülesande jaoks liiga lühike. Nimelt on see 2,7 meetrit pikk [36], mis tähendab seda, et testimise ala sensoritega ümbritsemiseks läheb vaja USB kaabli pikendust. Pikkade pikenduste kasutamisel võib olla probleemiks andmetekadu või programmi üldine aeglasem käitumine.

Kinect ei ole võimeline jälgima inimest selja tagant, seetõttu on selja tagant saadud andmed ebatäpsed, kuna sensor eeldab, et inimene on tema poole näoga [13]. Ühe Kinecti puhul soovitatakse kasutada näotuvastamist või jälgida inimese liikumist, et aru saada, millal inimene ei ole sensori poole näoga. See on ka üks põhjustest, miks seni kasutusel olev ühe Kinectiga süsteem on ebatäpne. Liikumine analüüsimiseks kasutatakse ainult neid andmeid, mida saadi siis kui inimene on reaalselt sensori poole näoga.

Up-and-Go test üheks oluliseks osaks on tool, millelt inimene püsti tõuseb ning mille peale hiljem maha istub. On oht, et võivad tekkida olukorrad, kus tool varjab mõned jäsemeid sensorite eest. Seda probleemi saab ilmselt lahendada Kinectide paigutamise abil sellisel viisil, et tool jääb minimaalselt ette.

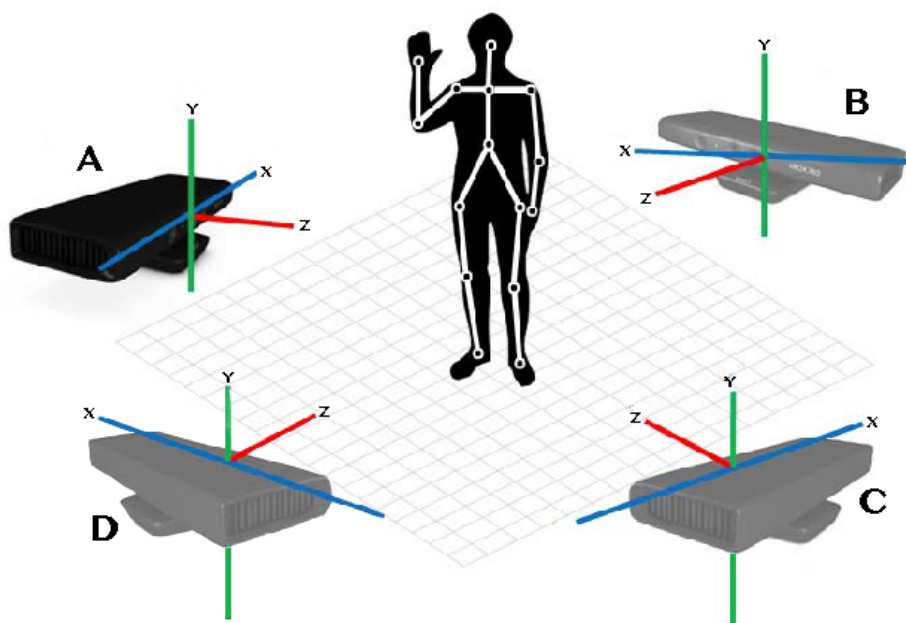
## 4. Implementatsioon

Antud lõputöö käigus loodavat süsteemi arendati kasutades C# .NET programmeerimiskeelt ning programmeerimiskeskkonnaks oli Microsoft Visual Studio 2013. Arendamiseks kasutati Kinect for Windows 1.8 SDK-d ning maatriksitega seotud arvutused teostati kasutades MathNet Numerics teeki. Samuti olid nii arendamisel kui ka testimisel kasutusel neli Kinect for Windows sensorit. Valminud koodi hoiti BitBucketi privaatses koodihoidlas.

Süsteemi arendamine ning testimine toimus HP EliteBook 2570p sülearvutis, milles oli Windows 8.1. Sülearvuti protsessoriks on Intel Core i5 (3rd Gen) 3210M / 2.5 GHz ning arvutil on olemas kolm USB hub'i.

### 4.1 Meetodi ülevaade

Neli Kinecti paigutatakse testimisala ümber. Üks nendest, antud juhul Kinect A, määratakse n-õ peamiseks Kinectiks. Kõikidelt teistelt sensoritelt (B, C ja D) saadud andmed teisendatakse peamise Kinecti koordinaatsüsteemi (vt. Joonis 7).



Joonis 7: Sensorite paigutus süsteemis

Selleks, et teisendada andmed ühest koordinaatsüsteemist teise on vaja teada rotatsioonimaatriksit ning nihkevektorit, mille abil on võimalik teisendada esialgsed koordinaadid soovitud süsteemi.

Rotatsioonimaatriks on 3x3 maatriks, mis kirjeldab objekti rotatsiooni 3D ruumis. Rotatsioonimaatriksit defineeritakse sellisel viisil, nagu on näha valemis (1).  $(x', x)$  tähistab nurka kahe erineva koordinaatsüsteemi x-telgede vahel,  $(x', y)$  tähistab nurka esimese süsteemi x-telje ning teise süsteemi y-telje vahel jne [29].

$$(1) \quad R = \begin{bmatrix} \cos(x', x) & \cos(y', x) & \cos(z', x) \\ \cos(x', y) & \cos(y', y) & \cos(z', y) \\ \cos(x', z) & \cos(y', z) & \cos(z', z) \end{bmatrix}$$

Rotatsioonimaatriksi abil on võimalik pöörata objekti ühes koordinaatsüsteemis sellisel viisil, et see oleks sama nurga all võrreldes soovitud koordinaatsüsteemiga.

Pööratud objekt teisest koordinaatsüsteemis ei pruugi asetseda samas kohas nagu seesama objekt vaadeldavas koordinaatsüsteemis. Selleks tuleb arvutada välja objekti nihkevektor ning selle abil nihutada objekt soovitud kohta. Nihkevektori abil saab iga keha punkti koordinaate nihutada kindlas suunas [35]. Teisendamise põhimõtet saab näha valemis (2), kus  $x, y, z$  on punkti esialgsed koordinaadid,  $R$  on rotatsiooni maatriks,  $t_x, t_y, t_z$  on nihkevektori koordinaadid ning  $x', y', z'$  on teisendatud punkti koordinaadid.

$$(2) \quad \begin{bmatrix} x \\ y \\ z \end{bmatrix} * R + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Valemis (2) kirjeldatud tegevust peab rakendama iga mitte peamiselt Kinectilt saadud skeletti punktile. Selle tulemusena saadakse neli skeletti, mis on kõik samas koordinaatsüsteemis – peamise Kinecti skelett ning kolm teisendatud skeletti. Nende neljas skeleti põhjal arvutatakse keskmine skelett. Kuna sensoreid ei liigutata, siis rotatsiooni maatriksit ja nihkevektorit ei pea pidevalt üle arvutama, neid seotakse vastava sensoriga tulevaste teisenduste jaoks.

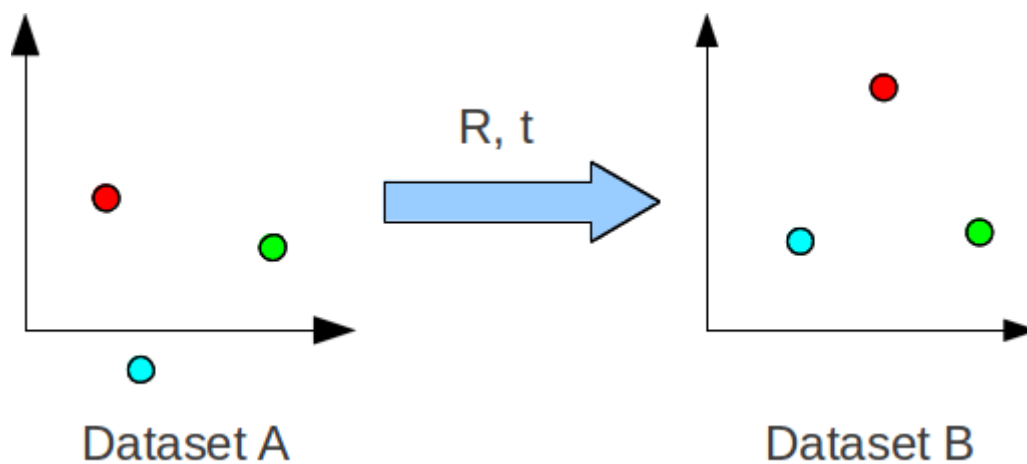
#### 4.1.1 Kabschi algoritm

Kinectidelt saadud skelettide andmed on punktipilved. Selleks, et arvutada B, C ja D Kinectilt saadud punktipilvede optimaalsed rotatsioonimaatriksid ning nihkevektorid

Kinect A koordinaatsüsteemi teisendamiseks, kasutatakse Kabschi algoritmi [14]. Kabschi algoritmi kasutamine minimeerib ruutkeskmist vigat [14].

Selleks, et leida optimaalset rotatsioonimaatriksit ja nihkevektorit punktipilvest A punktipilve B teisendamiseks (seejuures on mõlemas pilves vähemalt kolm teineteisele vastavat punkti) (vt. Joonis 8) tuleb läbi kolm sammu:

1. Leida mõlema punktipilve keskkohad
2. Nihutada mõlemad keskkohad koordinaatsüsteemide alguspunkti ning leida rotatsioonimaatriks  $R$
3. Leida nihkevektor  $t$



Joonis 8: Punktipilve teisendamine [6]

Keskpunktide leidmine on lihtne. Igal punktipilvedesse kuuluvad punktil on kolm koordinaati  $x, y, z$ . Keskpunktide arvutamise valemid on järgmised:

$$K_A = \frac{1}{N} \sum_i^N P_A^i \qquad K_B = \frac{1}{N} \sum_i^N P_B^i$$

Järgmiseks sammuks on rotatsioonimaatriksi leidmine. Selleks kasutatakse singulaarlahutuse meetodit (i.k. *Singular Value Decomposition*) [33], mille abil faktoriseeritakse reaalarvulist maatriksit (valemis (3) tähistatud  $E$ ). Faktoriseerimise tulemusena saadakse kolm maatriksit.

$$(3) \qquad [U, S, V] = SVD(E)$$

Seos faktoriseeritud maatriksi ning saadud kolme maatriksi vahel on (4).

$$(4) \quad E = USV^T$$

Selleks, et lihtsustada probleemi ning elimineerida nihke komponent, nihutame mõlemad punktipilved koordinaatetelgede alguspunkti. Pärast seda saab välja arvutada kovariatsiooni maatriksi  $H$  (5), millele saab hiljem rakendada SVD meetodit.

Kovariatsioonimaatriksi arvutamisel tuleb olla tähelepanelik, tulemuseks peab olema maatriks suurusega  $3 \times 3$ . Lisaks sellele on korrutamise järjekord oluline, kui see ära

$$H = \sum_i^N (P_A^i - K_A)(P_B^i - K_B)^T \quad (5)$$

vahtetada, siis arvutatakse rotatsioon pilvest B pilvesse A (mis antud näite puhul on vale). Pärast SVD meetodi rakendamist (6) on võimalik leida rotatsioonimaatriks  $R$  (7).

$$[U, S, V] = SVD(H) \quad (6)$$

$$R = VU^T \quad (7)$$

On erijuhtum, kui rotatsioonimaatriksi arvutus tagastab peegeldusmaatriksi, mis on küll arvuliselt õige, kuid reaalsuses väärtusetu. Selle vältimiseks tuleb kontrollida, kas maatriksi  $V$  determinant on negatiivne. Kui see on negatiivne, siis maatriksi  $V$  kolmandat veergu tuleb korrutada  $-1$ ga.

Viimaseks sammuks on nihkevektori  $t$  leidmine. Nihkevektori saab arvutada kasutades valemit (8), milles  $K_A$  ja  $K_B$  on veeruvektorid suurusega  $3 \times 1$ .

$$(8) \quad T = -R \times K_A + K_B$$

## 4.2 Tarkvaraline lahendus

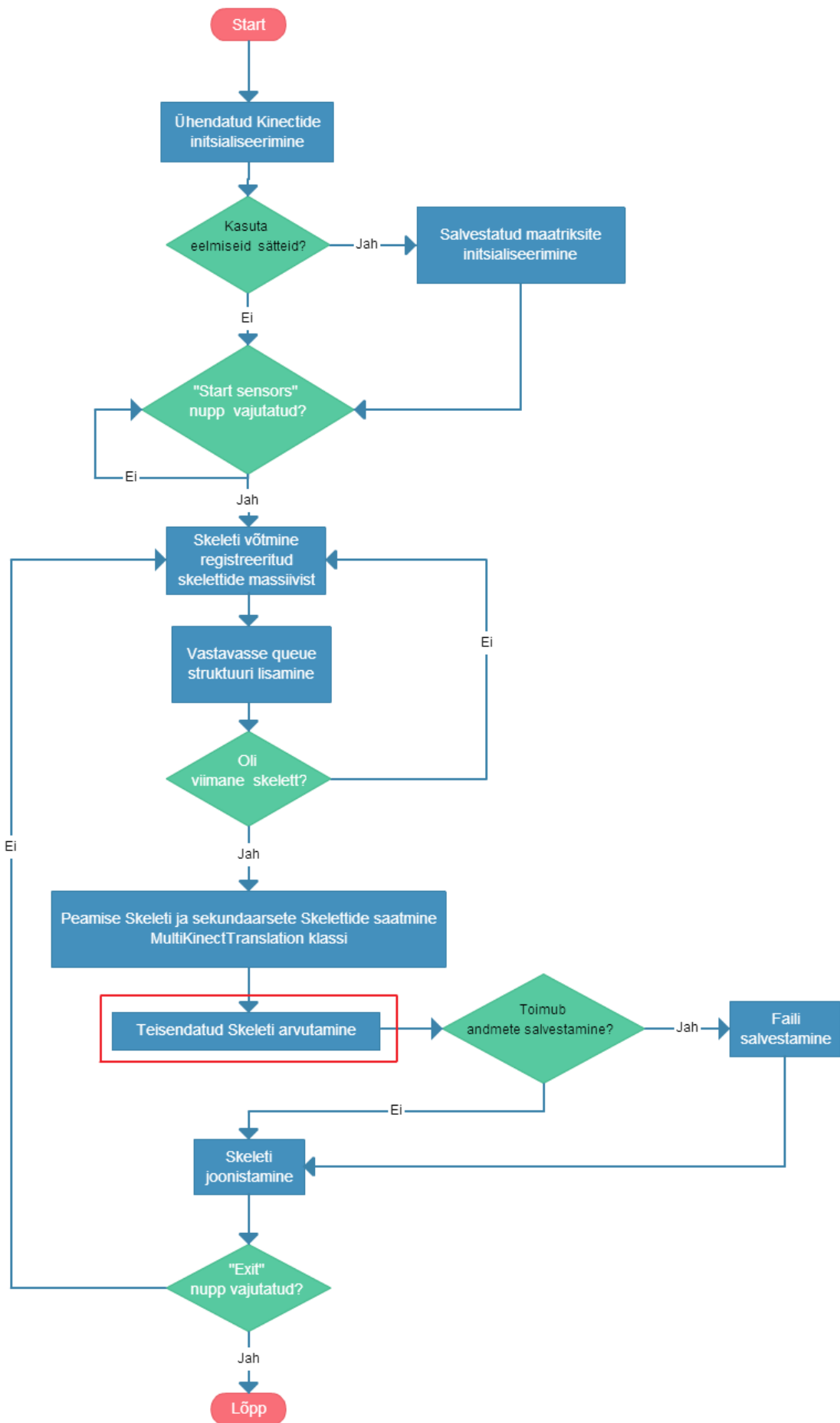
Tarkvaralise lahenduse loomisel lähtuti eesmärgist luua teek, mida saab kasutada teistes projektides. Loodud meetodi katsetamiseks loodi rakendus, mille abil saab neljalt Kinectilt teisendatud andmed salvestada hilisemaks analüüsiks. Valmis ka kaks lisarakendust, mille abil saab identifitseerida Kinecte ning teisendada salvestatud koordinaadid teise koordinaatsüsteemi.

Kinecti eraldusvõime on 30 kaadrit sekundis, kuid kuna ühe arvuti külge ühendatakse kuni neli sensorit, siis ülekoormamise vältimiseks võetakse sensori iga kuues kaader.

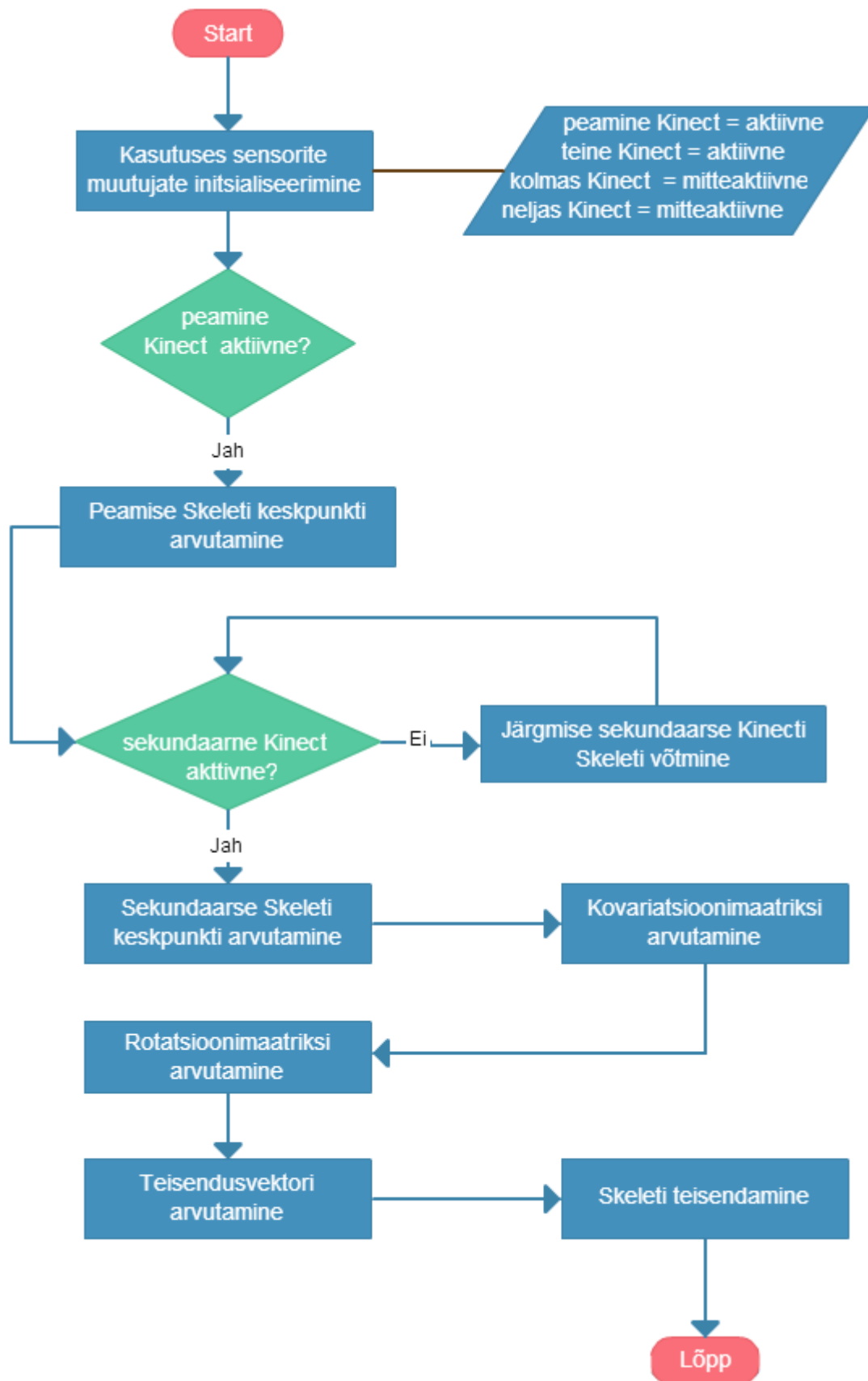
Sellisel juhul ei koormata arvutit üle ning mälus töötlemiseks salvestatud Skeleti objekt ei kirjutata enne arvutuste lõppu üle.

Loodud rakenduse töö käigu kirjeldus on Joonisel 9, kusjuures arendatud meetodiga seotud protsess on märgitud punase kastiga. Meetod ise on üpris keerulise loogikaga. Seetõttu on Joonisel 10 välja toodud üldine arvutusloogika, kui on aktiivne peamine Kinect ja üks sekundaarsetest Kinectidest. Olukordades, kus peamine Kinect pole aktiivne, on arvutuskäik sama, kuid teisenduste rakendamise järjekord on teistsugune.





Joonis 9: Programmi töökäik



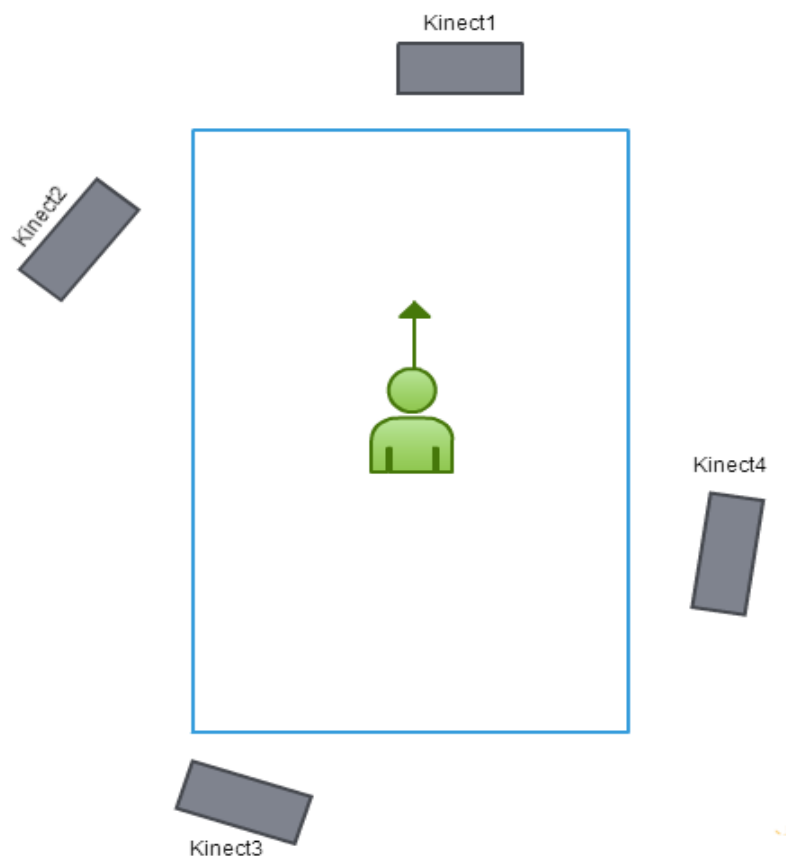
Joonis 10: Meetodi tööpõhimõte

### 4.2.1 Pööramise salvestamine

Kinecti arendustarkvara ei võimalda salvestada usaldusväärseid Skeleti liigete koordinaate, kui inimene on sensori poole seljaga. See tähendab seda, et kui inimene pöörab ennast sensori poole seljaga, siis saadud andmed käsitletakse tarkvara poolt nagu inimene oleks Kinecti poole näoga. Loogika järgi peaks neljalt Kinectilt, mis on ümber inimese paigutatud, saadud Skelett olema palju täpsem.

Antud lõputöö üheks oluliseks eesmärgiks on inimese pööramise salvestamine. Seetõttu tuleb meetodi tarkvaralises lahenduses Kinecti arendustarkvara puudusest üle saama.

Väljapakutud lahenduses eeldatakse, et inimene on peamise sensori poole näoga (vt. Joonis 11 – *Kinect1*) ning andmeid loetakse ainult sellelt sensorilt. Paralleelselt jälgitakse inimese orientatsiooni [28] Kinect1 suhtes.



Joonis 11: Sensorite paigutus

Inimese orientatsiooni arvutamine Kinecti suhtes toimub radiaanides. Kui inimene pöörab vastupäeva, siis nurk väheneb, vastasel juhul see suureneb. Arendatavas meetodis jälgitakse inimese orientatsiooni nurga muutumist aktiivsete sensorite suhtes. Kui inimese orientatsioon saavutab kindlat nurka, siis vastavalt nurgale, kas lülitatakse sisse järjekorras järgmine sensor või lülitatakse välja eelmine sensor.

Proovimise teel määrati kindlaks, mis nurkade all tuleks aktiivne sensor välja lülitada, kuna siis inimene kipub asetsema peaaegu küljega sensori poole. Selleks nurgaks on vastupäeva pöörates  $-0.7$  radiaani ( $-40^\circ$ ) ning päripäeva pöörates  $0.7$  radiaani ( $40^\circ$ ). Jägmise sensori sisselülitamine toimub, siis kui vastupäeva pöörates on nurk  $-0.6$  radiaani ( $-34^\circ$ ) ning päripäeva pöörates  $0.6$  radiaani ( $34^\circ$ ).

Sensorite ümberlülitamisprotsessi seletatakse Joonisel 11 toodud asetuse põhjal. Inimene seisab Kinect1 poole näoga ning alguses on ainult see sensor aktiivne. Iga saadud kaadri abil arvutatakse orientatsiooni. Inimene hakkab ennast pöörama vastupäeva (Kinect2 suunas) ning kui nurk on väiksem kui  $-0.4$  radiaani, siis Kinect2 muudetakse aktiivseks lisaks Kinect1-le. Kinect2-lt saadud Skeleti koordinaadid teisendatakse Kinect1 koordinaatsüsteemi ning selleks kasutatud rotatsioonimaatriks  $R_2$  ning nihkevektor  $T_2$  jäetakse meelde. Kui pööramisel muutub nurk väiksemaks kui  $-0.7$  radiaani, siis Kinect1 tehakse mitte aktiivseks. Kinect2-lt saadud koordinaadid teisendatakse jätkuvalt Kinect1 koordinaatsüsteemi kasutades salvestatud maatriksit  $R_2$  ning vektorit  $T_2$ . Kui nurk Kinect2 suhtes on väiksem kui  $-0.4$  radiaani, siis aktiveeritakse Kinect3. Rotatsioonimaatriksi ning nihkevektori arvutamiseks kasutatakse teist lähenemist, kuna Kinect1-lt saadud andmeid ei arvestata. Alguses arvutatakse rotatsioonimaatriks ning nihkevektor, et teisendada Kinect3 andmed Kinect2 koordinaatsüsteemi. Olgu nende maatriksi ning vektori tähistused  $R$  ja  $T$ . Edasi saab arvutada rotatsioonimaatriks  $R_3$  Kinect1 koordinaatsüsteemi teisendamiseks. Selleks tuleb sooritada maatriksite korrutamine  $R_3 = R * R_2$ . Nihkevektori leidmiseks tuleb vektorid omavahel liita  $T_3 = T + T_2$ . Kinect3-ga seotud rotatsioonimaatriks ja nihkevektor on olemas. Edasisel pööramisel jätkub sensorite ümberlülitamise loogika nii nagu oli enne kirjeldatud. Päripäeva pöörates on loogika sama, kuid nurgad, millega võrreldakse on  $0.7$  ja  $0.6$  radiaani.

Eelnevalt kirjeldatud sensorite ümberlülitamise protsess nõuab seda, et Kinectid oleksid paigutatud kindlas järjekorras ümber testimise ala ning sensoritelt saadud andmed töödeldakse kindlas järjekorras. Selle realiseerimisel tekkis mitu probleemi. See, mis järjekorras Kinectid programmi käigus initsialiseeritakse sõltub, mis järjekorras need arvuti külge ühendati. Sensorite kindlas järjekorras arvuti külge ühendamisel ei saa tagada seda, et neid initsialiseeritakse selles samas järjekorras.

Kinect SDK võimaldab kätte saada sensori unikaalse ID nimega UniqueKinectId. Selle ID kasutamine tundus paljulubav, kui jällegi tekkis probleem. Nimelt sensor, mida esimesena ühendatakse arvuti külge saab endale ID „/0000000000000000” ning ülejäänud saavad juba unikaalsed ID-d. UniqueKinectId kasutamine ei saanud jällegi tagada fikseeritud sensorite järjekorda.

Kinect SDK võimaldab samuti kätte saada iga sensori UniqueConnectionId, mis sõltub otseselt sellest, millisesse USB pessa sensor on arvuti külge ühendatud. Sensorite ühendamise järjekord ei mängi siin rolli. Arendatava meetodi abiks kirjutati ka väike programm, mille abil saab iga ühendatud sensori UniqueConnectionId teada. Saadud ID-d saab paika määrata loodud rakenduse App.config failis (vt. Joonis 12). Juhul kui sensorid jäävad ühendatud samadesse pesadesse, siis ID-sid ei pea muutma. Kui aga Kinectid ühendatakse teistesse pesadesse, siis UniqueConnectionId tuleb uuesti kindlaks määrata ning App.config failis muuta.

```
<appSettings>
  <add key="kinect1" value="6&A6EB709&1&3"/>
  <add key="kinect2" value="6&A6EB709&1&1"/>
  <add key="kinect3" value="6&5BFCF34&0&4"/>
  <add key="kinect4" value="6&5BFCF34&0&2"/>
</appSettings>
```

Joonis 12: UniqueConnectionId määramine App.config failis

### ***4.3 Ilmnenud raskused ja nende lahendused***

Tarkvaralise lahenduse loomisel ilmnis mitu probleemi, mis raskendasid püstitatud ülesande lahendamist. Pääaegu kõikidele ilmnenud probleemidele leiti viisid, kuidas neid lahendada.

Esimeseks probleemiks osutus arvuti ülekoormamine. Selgus, et ühe USB hub'i külge ei saa ühendada lõputu arvu Kinecte. Sellest probleemist räägiti ka alampeatükis 3.2.1.

Katsetamise tulemusena selgus, et loodud programmi sujuva töö tagamiseks saab ühe hub'i külge ühendada maksimaalselt kaks Kinecti. Kui ühe hub'i külge ühendada kolm sensorit, siis programm küll käivitub, kuid see on väga aeglane ning arvuti hakkab üle kuumenema.

Järgmisele probleemile samuti viidati alampeatükis 3.2.1. Nimelt selleks, et sensorid paigutada ümber testimise ala peab kasutama USB pikendusi, kuna Kinectiga komplektis olev juhe ei ole piisavalt pikk. Maksimaalne USB pikenduse pikkus, mille puhul sensorid käivituvad on viis meetrit. Kui üritada kasutada pikemaid pikendusi, siis arvuti ei suuda ühendatud Kinecti tuvastada. Ühe arvuti külge mitme Kinecti ühendamine piirab ka salvestamisala suurust, ala suurus hakkab sõltuma juhtmete pikkusest.

Programmi edukas töötamine sõltub ka inimese pikkusest. Selgus, et kui inimene on pikem kui kaks meetrit, siis ta ei mahu enam täielikult kaadrisse ning Kinecti tarkvara ei suuda tuvastada Skeletti. Lahendusena võib paigutada sensorid testimise alast kaugemale, et nende vaateväli suureneks või siis valida katsete jaoks inimesi, kes ei ole väga pika kasvuga.

Inimese pööramise salvestamisel toimuv Kinectide ümberlülitamine töötab kõige paremini kui testimise ala on kandiline. Sellisel juhul on alati olemas kaks Kinecti, mis on aktiivsed. Püstitatud ülesande üks eesmärkidest on aga Up-and-Go testi pööramise salvestamine. See eeldab seda, et inimene kõnnib pööramiste vahel ning see omakorda eeldab seda, et ala on hoopis piklik. Mida piklikum on tööala, seda suurema tõenäosusega hakkavad tekkima pimekohad, kus ükski sensor ei ole aktiivne inimese orientatsiooni nurga tõttu. Antud töö sooritamisel sõltus tööala otseselt ruumi suurusest. Selles tingituna maksimaalne tööala oli 152cmx66cm ning pimekohtade tekkimist ei märgatud.

Pööramise salvestamisel ilmnisid suuremad jälgitava skeleti asukoha muutused, kuid lülituti pikematel külgedel asuvate Kinectide vahel (Joonisel 11 Kinect2 ja Kinect3 vahel või Kinect1 ja Kinect4 vahel). Sellise järsu asukoha muutumise tõttu muutuvad salvestatud andmed ebatäpseks. Salvestatud andmete ebatäpsuse vältimiseks loodi lisameetod, mis arvutab Skeleti asukoha nihet ümberlülitumise hetkel võrreldes viimase

kaadriga, kus ei esinenud ümberlülitumist. Seejärel Skelett nihutatakse selle võrra ning sellisel viisil vähendatakse suurt asukoha muutumist sensorite ümberlülitamisel.

Igal programmi käivitamisel tuleb süsteemi kalibreerida, ehk sisuliselt tööalas sooritada üks pööre, mille jooksul arvutatakse iga sensoriga seotud rotatsioonimaatriksid ning nihkevektorid. See on suhteliselt tüütu, eriti kui sensoreid ei nihutata vahepeal ning maatriksid ja vektorid jäävad samadeks. Selle tõttu lisati arendatud süsteemile ka võimalus kasutada eelneval korral arvutatud rotatsioonimaatriksid ning nihkevektorid. Ainsaks piiranguks on see, et sensoreid ei tohi salvestuste vahel liigutada.

## 5. Tulemused

Loodud meetodi valideerimiseks kasutati võimsat MotionAnalysis kaamerate süsteemi. TTÜ arvutiteaduse instituudi valduses on seitsmest kaamerast koosnev liikumise registreerimise süsteem MotionAnalysis. Antud süsteem on markeritega liikumise registreerimise süsteem, mis võimaldab jälgida objekte sagedusega kuni 200 kaadrit sekundis. Valideerimiseks eeldatakse, et MotionAnalysisi poolt salvestatud kehapunktide koordinaadid on väga täpsed.

### 5.1 Keskkonna ettevalmistamine

MotionAnalysisi ning Kinectide süsteem paigutatakse ümber testimise ala, antud töö puhul on selle suuruseks 152cmx66cm (vt. Joonis 13). Mõlemad süsteemid tuleb enne salvestuste tegemist ruumis ringi nihutada, et saavutada selline pigutus, mille puhul kogu ala on kaamerate poolt kaetud. Lisaks sellele tuleb veel MotionAnalysisi süsteemi kalibreerida. Pärast seda ei tohi kaameraid liigutada, vastasel juhul peab neid uuesti kalibreerida. Peale seda saab inimese külge kinnitada markereid. Markereid kinnitatakse vastavalt [24] kirjeldatud viisile.

Kinectide süsteemi ettevalmistamine ei nõua otsest kalibreerimist. Teisenduste jaoks vajalikke rotatsioonimaatrikseid ja nihkevektoreid arvutatakse jooksvalt, kui inimene sooritab oma esimese 360-kraadise pöörde. Küll aga tuleb identifitseerida Kinectide UniqueConnectionId-d, mille abil fikseeritakse nende järjekord testimisala ümber.



Joonis 13: Sensorite ja kaamerate paigutus



## **5.2 Katsete kirjeldus**

Eialgu oli plaanis kleepida markerid teste sooritava inimese külge vastavalt [24] kirjeldatud viisile, mis on standartne viis, kuidas markereid liikumise registreerimiseks inimese külge pannakse. Mitme paralleelse salvestuse järel selgus aga, et ühendatud markerid tekitavad Kinectide jaoks tugeva interferentsi, mille tagajärel välja joonistatav skelett võbeleb ja vibreerib. Tekitatud interferentsi tõttu ei suutnud Kinectide süsteem õigesti aktiivsete sensorite vahel lülituda ning salvestada analüüsiks sobilikke andmeid.

Interferentsi probleemi lahendamiseks on kaks võimalust, kuidas salvestada analüüsiks sobilikke andmeid:

1. Saab salvestada ühe ja sama inimese poolt sooritatud liikumist, aga ainult, et alguses tehakse seda ainult Kinectidega ning hiljem kleebitakse markerid külge ning salvestus toimub juba MotionAnalysis süsteemi abil. Hiljem andmeid võrreldakse.

Antud lähenemise puuduseks on see, et salvestus ei toimu mõlema süsteemiga samaaegselt. Seetõttu ei saa garanteerida seda, et sooritatud liikumised tulevad täpselt samasugused ning liikumise kestvusaeg tuleb sama. Selle pärast on andmete analüüs keerukam.

2. Saab kleepida markerid ainult inimese tagumisele küljele, kus nad ei hakka segama aktiivseid Kinecte. Sellisel juhul on võimalik salvestada liikumist samaaegselt mõlema süsteemiga.

Antud lähenemise juures võib osutuda probleemseks andmete sünkroniseerimine, kuna MotionAnalysis salvestab andmeid palju kõrgema kaadrisagedusega kui Kinectide süsteem.

Antud meetodi valideerimisel otsustati teise lahenduse kasuks, kuna paralleelselt tehtud salvestuste analüüs annab paremat tagasisidet arendatud süsteemi täpsuse ja töökindluse kohta.

Markerid kleebiti inimese seljale, tulemusena nägi MotionAnalysis poolt salvestatav skelett väga sarnane Kinectide süsteemi skelettidele. Ainus vahe oli selles, et Kinectidelt saadud skeletil oli nelja võrra rohkem jälgitavaid liigese punkte.

Selja peale markerite kleepimine tekitas ka probleemi. Nimelt vähendas see testimisala pikkust oluliselt, kuna vastaselt juhul ei tuvastanud MotionAnalysise süsteem seljal asuvaid markereid. Selle tulemusena vähenes liikumisala pikkuses peaaegu poole võrra (86cmx66cm).

Lisaks markerite poolt tekitatud interferentsile tuli katsetest välja, et Kinectide süsteem on tundlik riide materjali suhtes. Näiteks tekitasid interferentsi püksid, mille materjal oli kergelt sädelev. Väljajoonistatava inimese Skeleti jalad võbelesid ja vibreerisid ning süsteem üldiselt hakkas sensoreid valesti ümber lülitama.

Katsed ise olid triviaalsed. Inimesel tuli testimisala sees kõndida edasi-tagasi 3-4 korda. Selleks, et oleks võimalik tuvastada katse algus ning lõpp, tuli inimesel enne kõndimise algust sooritada hüpe koha peal ning uuesti pärast kõndimise lõpetamist. Sellisel juhul on hiljem andmete töötlemises võimalik identifitseerida hüppekohad ning analüüsida nende vahele jäävad andmed. Ala väikese suuruse tõttu ei saanud sellesse paigutada tool, et sooritada tõuse ja kõnni test algusest lõpuni.

Katsete ajal tuli välja, et Kinectid hakkasid liiga kuumaks minema ning see mõjutad süsteemi tööd. Seetõttu tuli umbes iga 10-12 salvestuse järel sensorid paariks minutiks välja lülitada.

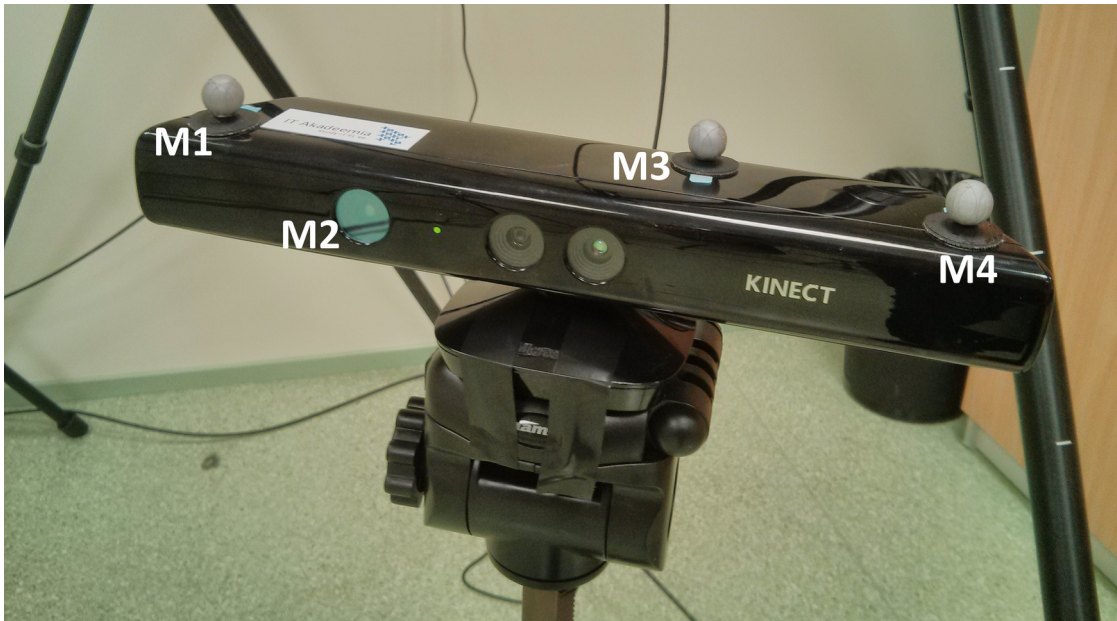
### ***5.3 Andmete töötlemine***

Paralleelselt salvestatud andmeid ei saanud koheselt analüüsida, kuna nad ei olnud samas koordinaatsüsteemis. Kinectidelt saadud andmed olid teisendatud Kinect1 (vt. Joonis 11) koordinaatsüsteemi ning MotionAnalysise andmed olid süsteemi kalibreerimise käigus paika pandud koordinaatsüsteemis. Selleks, et neid andmeid saaks omavahel võrrelda tuli Kinectidelt saadud andmed teisendada MotionAnalysise koordinaatsüsteemi.

Kasutades MotionAnalysise süsteemi saadi teada punktide M1, M2, M3, M4 koordinaadid (vt. Joonis 14). Teisendamisel nihutatakse peamine Kinect MotionAnalysise koordinaatsüsteemi alguspunkti ning siis rakendatakse rotatsioonimaatriks, mille tulemusena mõle süsteemi teljed kattuvad omavahel.

Selleks teisenduseks on vaja välja arvutada rotatsioonimaatriks ning nihkevektor. Peamise Kinecti peale kleebiti kolm markerit (vt. Joonis 14), mille abil oli võimalik kätte

saada selle sensori asukoht MotionAnalysise koordinaatsüsteemis. Lisaks sellele ilmnes, et Kinecti infrapuna saatjat tõlgendas MotionAnalysis samuti kui markerit. Antud teisenduse jaoks eeldati, et just infrapuna saatjast algabki Kinecti koordinaatsüsteem.



**Joonis 14: Peamise Kinecti peale kleebitud markerid**

Nihkevektoriks on antud juhul punkti M2 koordinaadid. Kasutades nihkevektorit nihutati kõik punktid selle võrra. Rotatsioon hakkab toimuma ümber y-telje (telg, millel asuvad koordinaadid, mis kirjeldavad inimese pikkust, vt. Joonis 2). Seetõttu saab arvutada lihtsama rotatsioonimaatriksi (9) ümber y-telje pööramiseks.

$$(9) \quad R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

Kasutades punkte M1 ja M4 arvutati vektori M1M4 normaal. Selleks, et välja arvutada nurka  $\beta$  tuleb välja arvutada nurk normaali ja x-telje vahel. Seda saab teha kasutades valemit (10), kus  $x$  on normaali x-koordinaat ja  $\|N\|$  on normaali pikkus.

$$(10) \quad \beta = \arccos\left(\frac{x}{\|N\|}\right)$$

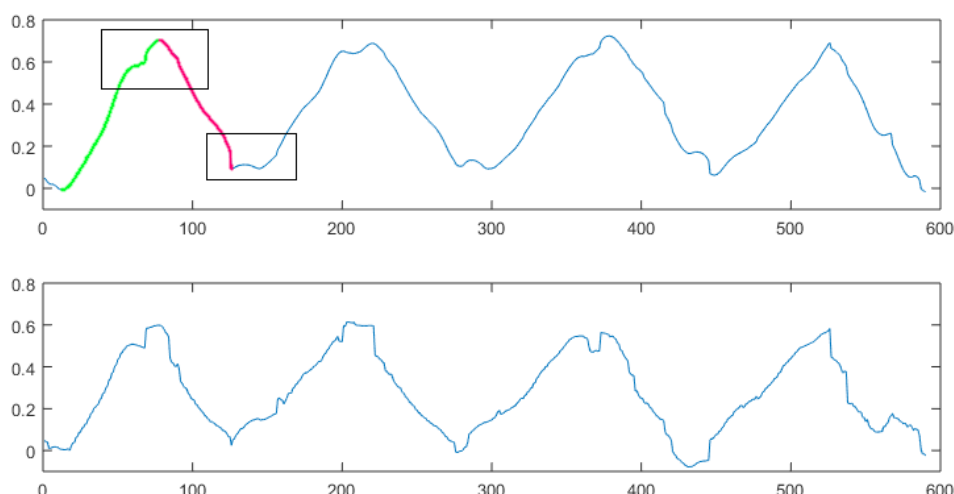
Kui rotatsioonimaatriks ning nihkevektor on olemas, siis on võimalik teisendada Kinectidelt saadud andmed MotionAnalysise koordinaatsüsteemi.

## 5.4 Tulemuste analüüs

Analüüsiks vajaminemvaid anmeid koguti tehes salvestusi seitsme inimesega, igaühega sooritati viis salvestust ning iga salvestuse jooksul kõnniti edasi-tagasi 3-5 korda. Kinectide süsteemi täpsuse hindamiseks jaotatakse üks liikumine kolmeks osaks:

1. Peamise Kinecti poole kõndimine
2. Pööramine
3. Peamisest Kinectist eemale kõndimine

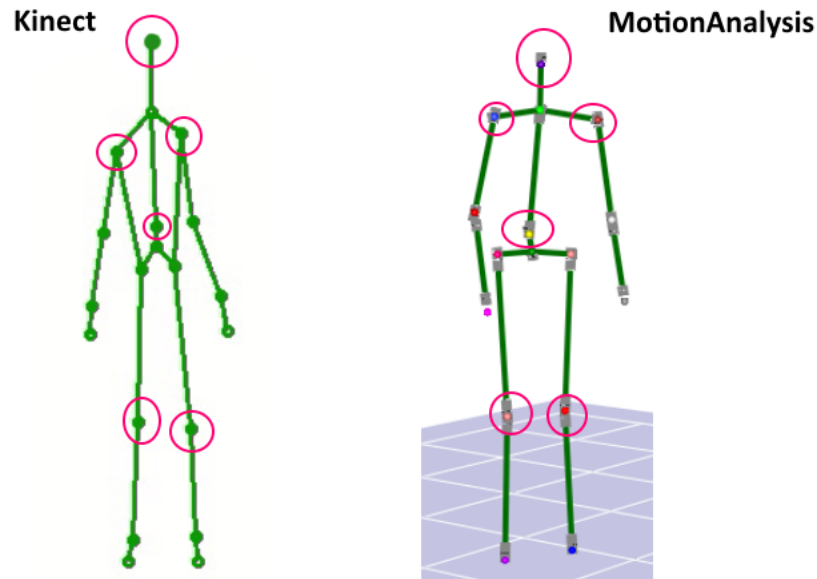
Joonisel 15 on näha graafikut, mis illustreerib pea z-koordinaadi muutust salvestuse ajal. Ülemine graafik on MotionAnalysise süsteemiga salvestatud liikumine, alumine on Kinectidega salvestatud. Rohelisega on märgitud peamise Kinecti poole kõndimisele vastav liikumine, punasega on peamisest Kinectist eemale kõndimine ning kastidega on märgitud pööramistele vastavad kohad.



**Joonis 15: Pea z-koordinaadi muutus salvestuse ajal**

Iga Kinectide süsteemi poolt salvestatud liikumist võrreldakse vastava MotionAnalysise poolt salvestatud liikumisega. Selle tulemusena on võimalik anda hinnang arendatud süsteemi täpsusele.

Võrdluspunktideks mõlemast süsteemis võetakse kindlad liigespunktid mudelitest. Nendeks on pea, õlad, selg ja põlved, Joonisel 16 on need punktid märgitud punase ringiga.



**Joonis 16: Kinecti ja MotionAnalüüsi inimkeha mudelid**

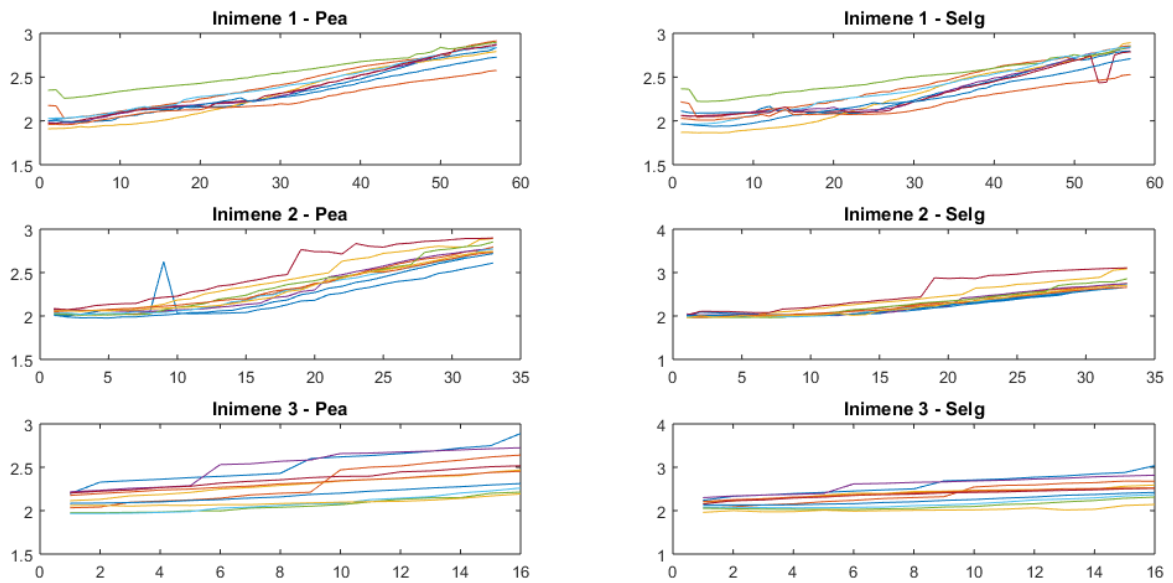
Kinectide ja MotionAnalüüsi poolt salvestatud andmete maht ei ole sama, kuna süsteemid salvestasid erineva sagedusega. Seetõttu tuli Kinectidelt saadud andmed teisendada MotionAnalüüsi sagedusele. Seejärel eemaldati mõlemast andmete hulgast kaadrid, mis vastasid Kinectide ümberlülitumisele (nad olid salvestatud andmetes ära märgitud). Ümberlülitumisele vastavate kaadrite eemaldamine eemaldas suuremate hüpete kohad, kuid mitte kõik. Admetest valiti välja kolme inimese salvestused, millest oli võimalik eraldada vähemalt üheksa kolmele liikumisele vastavad andmed. Neid saadud liigutusi hakati analüüsima.

#### **5.4.1 Koordinaatide vahe**

Nagu on näha Jooniselt 16, ei ole Kinecti ja MotionAnalüüsi poolt salvestatud punktid samade kohtade peal. Seetõttu Kinectilt saadud andmed ei teisendu täpselt samasse kohta nagu MotionAnalüüsi andmed. Võib eeldada, et kui arvutada iga vastava punktipaari vahelise kauguse ruumis, siis see võrdub mingi konstandiga või on mingi konstandi ümbruses.

Järgnevalt arvutatakse iga liigutuse kohta Kinecti ja MotionAnalüüsi andmete vahe pea ning selja kohta.

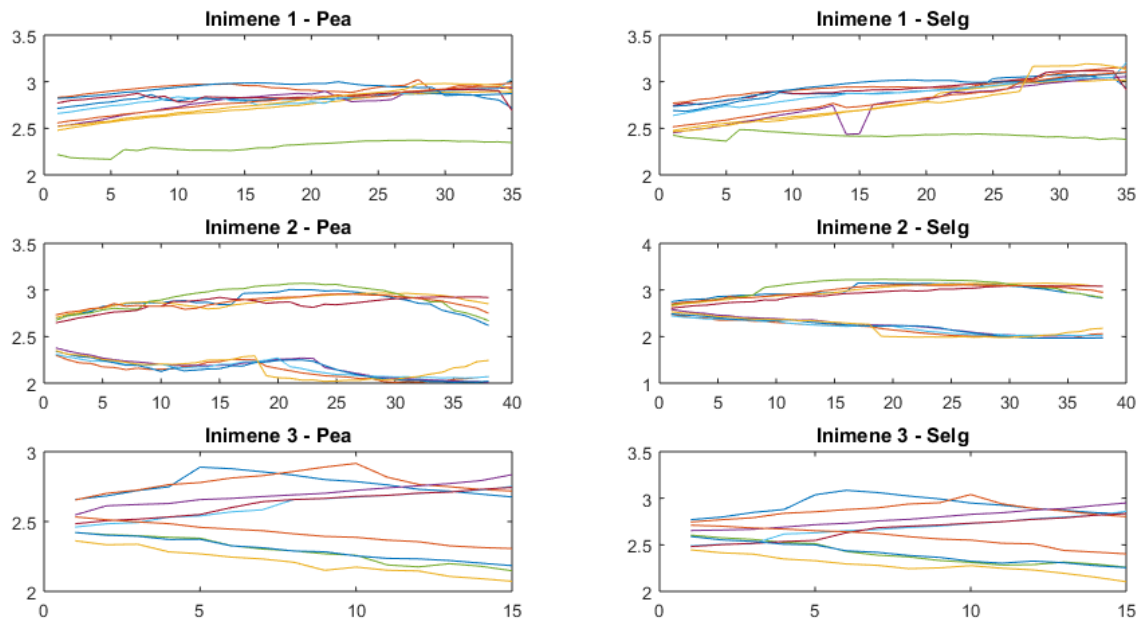
### 5.4.1.1 Peamise Kinecti poole kõndimine



**Joonis 17: Koordinaatide vahe Kinecti poole kõndides**

Jooniselt 17 on näha koordinaatide vahet pea ja selja kohta peamise Kinecti poole kõndimisel. Graafikutest on näha, et alguses on vahe üpris konstantne ning hiljem hakkab muutuma. Seda põhjustab liikumisala väike suurus. Sooritades katset ei olnud võimalik väga pikalt sirgelt kõndida, sisuliselt mahtus üks samm ning pärast seda algas pööramine. Kui vaadata ainult seda väikest osa, milles pööramist veel ei toimu, siis võib väita et koordinaatide vahe on peaaegu konstantne.

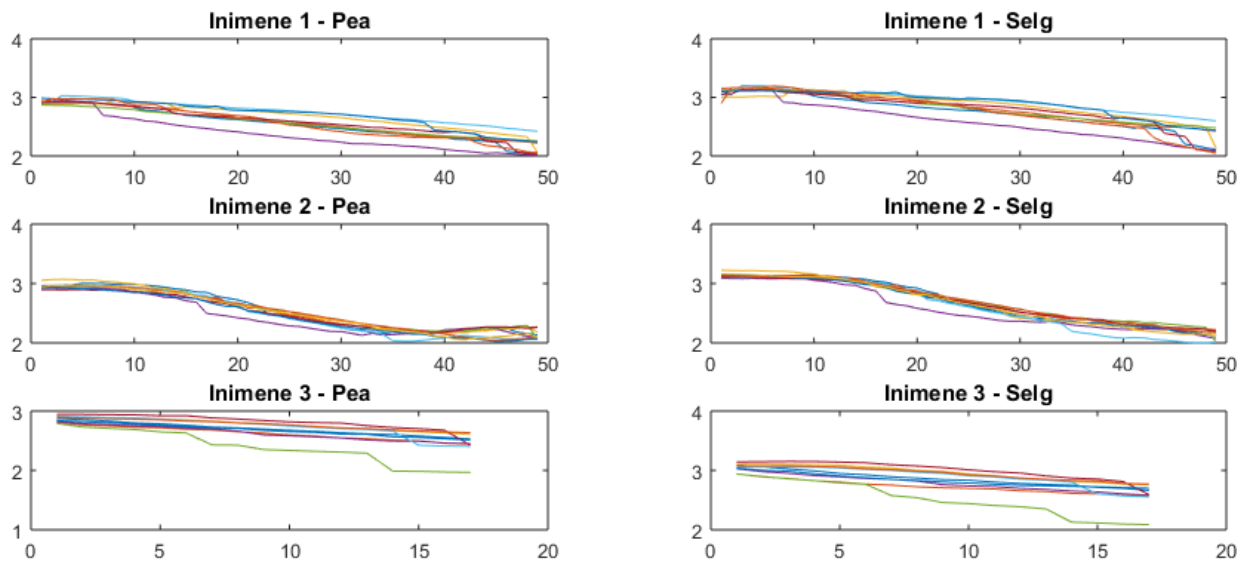
### 5.4.1.2 Pööramine



**Joonis 18: Koordinaatide vahe pöörates**

Jooniselt 18 on näha koordinaatide vahet pea ja selja kohta pöörates. Pöörates toimuvad ümberlülitumised ühelt sensorilt teisele, mis põhjustavad hüppeid andmetes. Neid hüppeid on ka graafikutel näha. Eriti hästi on koordinaatide vahet näha teise inimese graafikutelt. Pöörded sooritati eripidi, seetõttu on näha kaks erinevat joonte hulka. Kõigi graafikute puhul on samuti märgat, et koordinaatide vahe püsib kindla konstandi ümber.

### 5.4.1.3 Peamisest Kinectist eemale kõndimine



Joonis 19: Koordinaatide vahe Kinectist eemale kõndides

Jooniselt 19 on näha koordinaatide vahet pea ja selja kohta peamisest Kinectist eemale kõndimisel. On märgata sarnasust peamisest Kinecti poole kõndimisega. Siin toimub sirgelt kõndimine graafiku viimases osas ning enne seda toimub pöörde lõpetamine. Kui vaadata ainult graafikute viimast osa, milles pööramine on juba lõppenud, siis võib väita et koordinaatide vahe püsib konstandi ümbruses.

### 5.4.2 Liigutuste liikumismass

Olemasolevate liigutuste andmete jaoks põhjal [2] defineeritud liikumismassi (i.k. *MotionMass*). Liikumismass kirjeldab mingi liigutuse sujuvust. Eriti pakkusid huvi kolm parameetrit, milleks on trajektoori mass (i.k. *Trajectory mass*), kiiruse mass (i.k. *Velocity mass*) ja kiirenduse mass (i.k. *Acceleration mass*).

Kõigi kolme inimese andmete puhul jaotati arvutatud liikumismassid kaheks - Kinecti omaks ja MotionAnalyse omaks. Need andmed moodustasid kaks valimit, mida sai omavahel võrrelda. Võrdluse aluseks on nullhüpotees, et erinevate valimite mõõdetud tunnuste keskvaartused ei erine oluliselt [16]. MotionAnalyse andmetes olid kohad, kus süsteem ei näinud markereid, need andmed eemaldati nii Kinecti kui MotionAnalyse andmetest.



Andmete võrdlemiseks rakendati valimitele keskväärtuste testi ehk t-testi (*i.k two-sample t-test*) olulisuse nivooga 0.05. Kasutati MATLABi funktsiooni `ttest2`, mis tagastab kaks väärtuse  $h$ , mille põhjal saab nullhüpoteesi vastu võtta või tagasi lükata.. Kui  $h$  võrdub 0-ga, siis tegemist on valimitega, mille keskväärtused ei erine, kui 1-ga siis ei ole. Väärtus  $p$  näitab tõenäosust, et nullhüpotees on tõene.

Järgnevalt on esitatud arvutused kolme inimese kohta kõigi valitud liigete puhul.

Tabelites on kasutatud lühendid:

- $t_m$  – trajektoori mass
- $k_m$  – kiiruse mass
- $k_{im}$  – kiirenduse mass

### 5.4.2.1 Peamise Kinecti poole kõndimine

#### Pea

|          | Inimene 1                 | Inimene 2               | Inimene 3                 |
|----------|---------------------------|-------------------------|---------------------------|
| $t_m$    | $h = 1$<br>$p = 7.34e-04$ | $h = 0$<br>$p = 0.4965$ | $h = 0$<br>$p = 0.6089$   |
| $k_m$    | $h = 1$<br>$p = 6e-04$    | $h = 0$<br>$p = 0.4999$ | $h = 0$<br>$p = 0.9902$   |
| $k_{im}$ | $h = 1$<br>$p = 1.92e-08$ | $h = 0$<br>$p = 0.3675$ | $h = 1$<br>$p = 5.29e-08$ |

#### Selg

|          | Inimene 1               | Inimene 2                   | Inimene 3                 |
|----------|-------------------------|-----------------------------|---------------------------|
| $t_m$    | $h = 0$<br>$p = 0.7419$ | $h = 0$<br>$p = 0.9415$     | $h = 0$<br>$p = 0.5171$   |
| $k_m$    | $h = 0$<br>$p = 0.6495$ | $h = 0$<br>$p = 0.9890$     | $h = 0$<br>$p = 0.4364$   |
| $k_{im}$ | $h = 0$<br>$p = 0.4427$ | $h = 1$<br>$p = 8.2999e-05$ | $h = 1$<br>$p = 3.35e-06$ |

## Õlad

|     | Inimene 1             |                       | Inimene 2             |                     | Inimene 3             |                     |
|-----|-----------------------|-----------------------|-----------------------|---------------------|-----------------------|---------------------|
|     | Parem                 | Vasak                 | Parem                 | Vasak               | Parem                 | Vasak               |
| tm  | h = 1<br>p = 0.0063   | h = 0<br>p = 0.2663   | h = 1<br>p = 0.0319   | h = 1<br>p = 0.0013 | h = 0<br>p = 0.6058   | h = 0<br>p = 0.1238 |
| km  | h = 1<br>p = 0.0169   | h = 0<br>p = 0.5868   | h = 0<br>p = 0.3420   | h = 1<br>p = 0.0026 | h = 0<br>p = 0.6074   | h = 0<br>p = 0.0779 |
| kim | h = 1<br>p = 1.88e-05 | h = 1<br>p = 4.54e-11 | h = 1<br>p = 1.19e-04 | h = 1<br>p = 0.0256 | h = 1<br>p = 5.61e-05 | h = 0<br>p = 0.5879 |

## Põlved

|     | Inimene 1             |                       | Inimene 2           |                       | Inimene 3             |                       |
|-----|-----------------------|-----------------------|---------------------|-----------------------|-----------------------|-----------------------|
|     | Parem                 | Vasak                 | Parem               | Vasak                 | Parem                 | Vasak                 |
| tm  | h = 1<br>p = 9.23e-08 | h = 1<br>p = 0.0108   | h = 0<br>p = 0.7975 | h = 1<br>p = 0.0217   | h = 0<br>p = 0.5044   | h = 1<br>p = 0.001    |
| km  | h = 1<br>p = 0.0085   | h = 1<br>p = 0.0213   | h = 0<br>p = 0.8211 | h = 1<br>p = 0.0322   | h = 0<br>p = 0.6556   | h = 1<br>p = 0.0025   |
| kim | h = 1<br>p = 6.39e-11 | h = 1<br>p = 2.17e-08 | h = 0<br>p = 0.5491 | h = 1<br>p = 1.34e-04 | h = 1<br>p = 1.33e-06 | h = 1<br>p = 4.24e-05 |

## 5.4.2.2 Pööramine

### Pea

|     | Inimene 1             |  | Inimene 2             |  | Inimene 3             |  |
|-----|-----------------------|--|-----------------------|--|-----------------------|--|
| tm  | h = 1<br>p = 4.84e-05 |  | h = 1<br>p = 0.0384   |  | h = 0<br>p = 0.0877   |  |
| km  | h = 1<br>p = 0.0072   |  | h = 1<br>p = 0.0221   |  | h = 0<br>p = 0.2528   |  |
| kim | h = 1<br>p = 8.41e-08 |  | h = 1<br>p = 1.42e-09 |  | h = 1<br>p = 3.46e-04 |  |

### Selg

|     | Inimene 1             |  | Inimene 2             |  | Inimene 3           |  |
|-----|-----------------------|--|-----------------------|--|---------------------|--|
| tm  | h = 0<br>p = 0.0914   |  | h = 0<br>p = 0.3203   |  | h = 0<br>p = 0.8336 |  |
| km  | h = 0<br>p = 0.5976   |  | h = 0<br>p = 0.4679   |  | h = 0<br>p = 0.5404 |  |
| kim | h = 1<br>p = 1.01e-07 |  | h = 1<br>p = 3.68e-11 |  | h = 0<br>p = 0.3776 |  |

### Õlad

|     | Inimene 1             |                       | Inimene 2             |                       | Inimene 3             |                     |
|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|
|     | Parem                 | Vasak                 | Parem                 | Vasak                 | Parem                 | Vasak               |
| tm  | h = 1<br>p = 0.0046   | h = 0<br>p = 0.0883   | h = 0<br>p = 0.0670   | h = 1<br>p = 5.03e-05 | h = 0<br>p = 0.3409   | h = 0<br>p = 0.6182 |
| km  | h = 0<br>p = 0.0576   | h = 0<br>p = 0.3396   | h = 0<br>p = 0.2115   | h = 1<br>p = 5.25e-05 | h = 0<br>p = 0.3462   | h = 0<br>p = 0.5218 |
| kim | h = 1<br>p = 5.51e-05 | h = 1<br>p = 4.54e-11 | h = 1<br>p = 3.59e-14 | h = 1<br>p = 0.0021   | h = 1<br>p = 1.04e-04 | h = 0<br>p = 0.2757 |

### Põlved

|     | Inimene 1             |                       | Inimene 2            |                       | Inimene 3             |                     |
|-----|-----------------------|-----------------------|----------------------|-----------------------|-----------------------|---------------------|
|     | Parem                 | Vasak                 | Parem                | Vasak                 | Parem                 | Vasak               |
| tm  | h = 1<br>p = 0.0238   | h = 1<br>p = 0.0023   | h = 1<br>p = 0.0155  | h = 1<br>p = 0.0037   | h = 0<br>p = 0.5608   | h = 1<br>p = 0.0128 |
| km  | h = 0<br>p = 0.2420   | h = 1<br>p = 0.0025   | h = 0<br>p = 0.0752  | h = 1<br>p = 0.0048   | h = 0<br>p = 0.8738   | h = 1<br>p = 0.0393 |
| kim | h = 1<br>p = 1.62e-07 | h = 1<br>p = 1.68e-06 | h = 1<br>p = 2.3e-11 | h = 1<br>p = 3.68e-06 | h = 1<br>p = 1.94e-04 | h = 1<br>p = 0.0034 |

### 5.4.2.3 Peamisest Kinectist eemale kõndimine

#### Pea

|     | Inimene 1             |  | Inimene 2             |  | Inimene 3             |  |
|-----|-----------------------|--|-----------------------|--|-----------------------|--|
| tm  | h = 1<br>p = 0.0234   |  | h = 1<br>p = 0.0057   |  | h = 1<br>p = 0.0122   |  |
| km  | h = 1<br>p = 0.0089   |  | h = 1<br>p = 0.0049   |  | h = 0<br>p = 0.1131   |  |
| kim | h = 1<br>p = 1.49e-09 |  | h = 1<br>p = 4.01e-11 |  | h = 1<br>p = 7.29e-06 |  |

## Selg

|     | Inimene 1           |  | Inimene 2             |  | Inimene 3             |  |
|-----|---------------------|--|-----------------------|--|-----------------------|--|
| tm  | h = 0<br>p = 0.8732 |  | h = 0<br>p = 0.8194   |  | h = 0<br>p = 0.5169   |  |
| km  | h = 0<br>p = 0.6706 |  | h = 0<br>p = 0.8851   |  | h = 0<br>p = 0.6838   |  |
| kim | h = 0<br>p = 0.0558 |  | h = 1<br>p = 3.55e-09 |  | h = 1<br>p = 1.17e-05 |  |

## Õlad

|     | Inimene 1             |                       | Inimene 2             |                     | Inimene 3             |                     |
|-----|-----------------------|-----------------------|-----------------------|---------------------|-----------------------|---------------------|
|     | Parem                 | Vasak                 | Parem                 | Vasak               | Parem                 | Vasak               |
| tm  | h = 1<br>p = 0.0183   | h = 0<br>p = 0.0728   | h = 0<br>p = 0.1577   | h = 1<br>p = 0.0049 | h = 0<br>p = 0.0864   | h = 0<br>p = 0.1463 |
| km  | h = 0<br>p = 0.0720   | h = 0<br>p = 0.1368   | h = 0<br>p = 0.2313   | h = 1<br>p = 0.0075 | h = 0<br>p = 0.3072   | h = 0<br>p = 0.1081 |
| kim | h = 1<br>p = 6.03e-06 | h = 1<br>p = 1.35e-11 | h = 1<br>p = 3.91e-11 | h = 0<br>p = 0.3461 | h = 1<br>p = 2.14e-05 | h = 0<br>p = 0.7883 |

## Põlved

|     | Inimene 1             |                       | Inimene 2           |                      | Inimene 3             |                       |
|-----|-----------------------|-----------------------|---------------------|----------------------|-----------------------|-----------------------|
|     | Parem                 | Vasak                 | Parem               | Vasak                | Parem                 | Vasak                 |
| tm  | h = 1<br>p = 0.0415   | h = 1<br>p = 0.0041   | h = 0<br>p = 0.2350 | h = 1<br>p = 0.0063  | h = 0<br>p = 0.6980   | h = 1<br>p = 0.0014   |
| km  | h = 0<br>p = 0.1250   | h = 1<br>p = 0.0021   | h = 0<br>p = 0.2440 | h = 1<br>p = 8.6e-04 | h = 1<br>p = 0.8200   | h = 1<br>p = 0.0053   |
| kim | h = 1<br>p = 2.56e-08 | h = 1<br>p = 4.49e-05 | h = 1<br>p = 0.0022 | h = 1<br>p = 2.2e-09 | h = 1<br>p = 5.52e-06 | h = 1<br>p = 7.16e-06 |

### 5.4.3 Järeldused

- Nagu on näha siis kõige tihedamini kehtib nullhüpotees selja puhul. Teiste punktide kokkulangevust mõjutavad tugevalt ümberlülitumisest tulenevad hüpped ning ka see kas punkt mahub realselt Kinecti vaatevälja.

- Kõige usaldusväärsemaid andmeid liikumise sujuvuse kohta saab kasutades selja punkti.
- Pea ning põlved ei pruugi olla usaldusväärsed, kuna väikese liikumisala tõttu ei mahtunud nad alati kaadrisse.
- Õlgade liikumise sujuvus on hüplikust mõjutavad ümberlülitumisest tulevad hüpped.
- Kinectide poolt salvestatud punktid ei vasta üks-ühele MotionAnalysise punktidele. Seetõttu on salvestatud koordinaatides olemas nihe. Analüüsist on näha, et kuigi see nihe on olemas, siis see püsib mingi teatud konstandi ümbruses.

### ***5.5 Arengusuunad tulevikus***

Arendatud pööramisliigutuse registreerimise meetod ei ole täiuslik. Tegemist on esimese TTÜ arvutiteaduse instituudis valminud tarkvaralise lahendusega, mis kasutab liikumise registreerimiseks rohkem kui kahte Kinecti. Seetõttu saab nii meetodit ennast kui ka tarkvara tulevaste tööde käigus täiustada ning paremaks muuta.

Järgmiseks loogiliseks edasiarenduseks oleks rohkem kui nelja Kinecti kasutuselevõtmine. Rohkem kui nelja Kinectide teenindamisega ei saa üks arvuti enam hakkama. Seetõttu tuleb mõelda lahendusele, mis kasutaks mitut arvutit, mille külge on sensorid ühendatud. Ühes arvutis jookseks serverirakendus, mis saaks üle võrgu Kinectidelt tulevad andmed. Andmete sünkroniseerimine ning sensorite kindla järjekorra paika panemine muutub keerulisemaks, kuna Kinecte ei saa enam UniqueConnectionId põhjal eristada. Teisenduskoodi ennast peab muutma minimaalselt, et see saaks töötada ükskõik, mis arvu sensoritega. Rohkem kui nelja Kinecti kasutuselevõtmine parandab süsteemi täpsust, kui liikumisala on piklik. Samuti lahendab see liikumisala suuruse piiratuse probleemi, kuna Kinectid ei ole kõik ühe arvuti küljes ning sensoreid saaks paigutada ringi palju vabamalt. Praegusel hetkel sõltub Kinectide paigutus otseselt sellest, kui kaugele ulatub USB pikendusjuhe.

Hetkeseisuga ei ole arendatud süsteem kõige töökindlam ja täpsem. Ümberlülitudes ühelt sensorilt teisele esinevad hüpped, millest tulenevad ebatäpsed

andmed. Samuti esinevad olukorrad, kus süsteem satub segadusse ning hakkab sensoreid valesti ümber lülitama. Täpsuse probleemi saaks lahendada andmete töötlemisalgoritmiga, mis eemaldaks hüpped kas reaalajas või juba pärast salvestuse lõppu. Süsteemi töökindlust saaks muuta kas kasutades rohkem kui nelja Kinecti või luuga rangem sensorite ümberlülitumise algoritm, mis elimineeriks kaootilise ümberlülitumise võimalust.

Kahjuks Kinect for Windows sensoreid rohkem ei toodeta, selle asemele saab kasutada 2014ndal aastal ilmunud uut Kinecti mudelit – Kinect V2. Koos sellega ilmus ka uus arendustarkvara Kinect SDK 2.0, mis kahjuks vana sensorit enam ei toeta. Uus arendustarkvara on rohkem suunatud teadustöö tegemise võimaldamisele, see peaks olema ka täpsem ning jälgitavaid Skeleti punkte on 20 asemel 25. Valminud meetod koos selle tarkvaralise implementatsiooniga tuleb ümber kirjutada ning täiustada, et see töötaks kasutades uut Kinect V2 ja Kinect SDK 2.0.

## Kokkuvõte

Käesoleva töö põhieesmärgiks oli välja töötada meetod kuni neljalt Kinectilt saadud andmete töötlemiseks. Arendatud meetod implementeeriti C# programmeerimiskeeles sellisel kujul, et seda on võimalik kasutada teegina teistes projektides. Lisaks sellele valmis lisaprogramm, millega on võimalik identifitseerida ühe arvuti külge ühendatud Kinectide UniqueConnectionId. UniqueConnectionId teadmine võimaldab Kinectidega, mis on ühe arvuti külge ühendatud suhelda fikseeritud järjekorras.

Üheks selle töö olulisemaks tulemuseks on arusaamine, kuidas tagada programmi, mis tegeleb kuni nelja Kinecti teenindamisega, sujuv töö. Mitme Kinecti kasutamine on küll Microsofti poolt toetatud, kuid selle kohta pole piisavalt palju informatsiooni ning praktilisi näpunäiteid. Töö sooritamisel puututi kokku mitme probleemi ja raskusega, millele suudeti leida alternatiivne lahendus.

Arendatud meetodi täpsust valideeriti kasutades MotionAnalysise kaamerate süsteemi. Kuna Kinectide poolt salvestatud punktid ei ühtinud üks-ühele MotionAnalysise punktidele, siis eeldati algusest peale, et punktide vahel on olemas nihe. Andmete analüüsi käigus arutati valitud punktide vahel Kinecti ja MotionAnalysise andmete vahel. See arvutus andis ettekujutuse sellest, kuidas mõjutavad ümberlülitumisest tulenevad hüpped andmete täpsust võrreldes MotionAnalysise süsteemiga. Lisaks sellele rakendati Kinecti ja MotionAnalysise andmetele keskväärtuste võrdlust, millel põhjal sai otsustada kas nende valimite keskväärtused erinesid või mitte kasutades olulisuse nivood 5%.

Järgnevalt on välja toodud antud töö olulisemad järeldused.

- Mitme Kinecti kasutamisel ühe ala salvestamiseks ei esine nende vahelist interferentsi. See-eest tekitasid interferentsi MotionAnalysise markerid ja spetsiifiline riie materjal.
- Süsteem saab hakkama Kinecti poole, sellest eemale kõndimise ja pööramise salvestamisega. Usaldusväärseid andmeid ei saa kõikide punktide kohta ning kõige usaldusväärseim punkt on selg.
- Pööramisel kasvab koordinaatide vahe.
- Süsteemi täpsust mõjutavad ümberlülitumisest tulenevad hüpped.

- Kõige usaldusväärsem punkt analüüsimiseks on selg, ülejäänute täpsust mõjutavad ümberlülitumiste poolt põhjustatud hüpped ja sensori vaatevälja mitte mahtumine.

Praeguse seisuga saab valminud tarkvara edasi arendada, et muuta süsteem täpsemaks ning töökindlamaks. Arendatud süsteem võimaldab täpsemini salvestada pööramist ning liikumist selja tagant võrreldes ainult ühe Kinecti kasutamisega. Lisaks sellele on see hea baas edasiste mitut Kinecti kasutavate projektide tegemiseks.



## Summary

The main purpose of this thesis was to develop a method which allows the usage of up to four Kinect sensors to capture motion. The developed method was implemented in the C# programming language, the code can be imported for usage in other Kinect related projects. In addition to that, a small program was written which helps to identify the UniqueConnectionId of Kinect sensors which are connected to one computer. Knowing the UniqueConnectionId allows to communicate with several Kinects in a fixed order.

One of the results of this work is the practical knowledge of creating a system which supports the usage of multiple Kinect sensors. Using multiple Kinects to track people is supported by Microsoft, however the information available on this topic is scarce. During the developments quite a few obstacles and problems were encountered. For all the problems and obstacles solutions or workarounds have been found.

The accuracy of the developed system was validated using a precise camera system called MotionAnalysis. From the start it was assumed that points which are tracked by Kinects have a shift in them compared to MotionAnalysis points and thusly do not match exactly. While analysing the data the distance between coordinates of corresponding points was calculated. Based on this calculation the impact of Kinect sensor switching on data accuracy compared to MotionAnalysis system can be seen. Furthermore, two-sample t-test was applied to the two data sets to determine whether the means of the two sets differ or not with the significance level of 5%.

The following are the key conclusions of this thesis.

- No interference was noticed when using multiple Kinects to track humans the same area. However interference was caused by the MotionAnalysis system's markers and some specific material of clothes.
- The system can handle tracking of walking towards the main Kinect, walking away from the main Kinect and turning. The most reliable data can be acquired about the Spine joint.
- When turning the coordinate error of corresponding points increases.

- The accuracy of the system is affected by the data jumps cause by the Kinect sensor switching.
- The most reliable point for further analysis is the spine, the rest are influenced by the jumps caused by sensor switching and not fitting in to the frame of the sensor.

As of now the finished implementation of the developed method can be further bettered and thus improving the accuracy and the reliability of the system. Compared to using only one Kinect the developed system allows to track more precisely the turning motion and walking from the back. Furthermore, this work is a solid base for future multi-Kinect related systems' and projects' development.

## Kasutatud materjalid

1. Alexiadis, D. S., Zarpalas, D., Daras, P. (2013). Real-time, realistic full-body 3D reconstruction and texture mapping from multiple Kinects. - *IVMSP Workshop: 10.06 – 12.06.2013, Seoul, Lõuna-Korea, 1-4*
2. An alternative approach to measure quantity and smoothness of the human limb motions [WWW] [http://www.kirj.ee/public/Engineering/2013/issue\\_4/Eng-2013-4-298-308.pdf](http://www.kirj.ee/public/Engineering/2013/issue_4/Eng-2013-4-298-308.pdf) (24.05.2015)
3. Boruško, J. (2014). Alternative Approach to Model Changes of Human Motor Functions – *Magistritöö, Tallinn, Eesti*
4. Buhhalko, K. (2013). Monitoring of the human motor functions rehabilitation by neural networks based system with Kinect sensor – *Magistritöö, Tallinn, Eesti*
5. Erap, S. (2012). Gesture Based PC Interface with Kinect Sensor – *Magistritöö, Tallinn, Eesti*
6. Finding optimal rotation and translation between corresponding 3D points [WWW] [http://ngghiaho.com/?page\\_id=671](http://ngghiaho.com/?page_id=671) (05.11.2014)
7. Frequently asked questions [WWW] <https://msdn.microsoft.com/en-us/library/jj131032.aspx> (11.03.2015)
8. Get-up-and-go test [WWW] <http://www.albertahealthservices.ca/ps-1051701-fpp-gug-instructions.pdf> (23.04.2015)
9. Hasler, N. et al. (2009). Markerless Motion Capture with unsynchronized moving cameras. - *Computer Vision and Pattern Recognition: 20.06 – 25.06.2009, Miami, Ameerika Ühendriigid, 224 - 231*
10. Herrera, D., Kannala, J., Heikkilä, J. (2012). Joint Depth and Color Camera Calibration with Distortion Correction. - *IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE, 2058 - 2064*
11. How Microsoft Kinect works [WWW] <http://electronics.howstuffworks.com/microsoft-kinect2.htm> (27.02.2014)
12. In Depth Review of TUG [WWW] [http://strokengine.ca/assess/module\\_tug\\_indepth-en.html](http://strokengine.ca/assess/module_tug_indepth-en.html) (08.01.2015)
13. Is it possible to track user's direction when user turns his/her back to Kinect? [WWW] <https://social.msdn.microsoft.com/Forums/en-US/218f058f-776d-4182-a477-b65a0a71b54b/is-it-possible-to-track-the-users-direction-when-user-turns-hisher-back-to-kinect?forum=kinectsdk> (31.03.2015)

14. Kabsch algorithm [WWW] [http://en.wikipedia.org/wiki/Kabsch\\_algorithm](http://en.wikipedia.org/wiki/Kabsch_algorithm) (05.11.2015)
15. Kaenchan, S., Mongkolnam, P., Watanapa, B., Sathienpong, S. (2013). Automatic multiple Kinect cameras setting for simple walking posture analysis - *Computer Science and Engineering Conference: 04.09 – 06.09.2013, Nakorn Pathom, Thailand*, 245 – 249
16. Keskmiste võrdlemine [WWW] <http://rgraafikud.weebly.com/keskmiste-votilderdlemine.html> (24.05.2015)
17. Kinect face recognition and emotion analysis [WWW] <http://slides.com/gosha/execom-ted-kinect/fullscreen#/> (22.04.2015)
18. Kinect for Windows SDK [WWW] <http://msdn.microsoft.com/en-us/library/hh855347.aspx> (03.03.2014)
19. Kinect for Windows Sensor Components and Specifications <http://msdn.microsoft.com/en-us/library/jj131033.aspx> (27.02.2014)
20. Kinect SDK 2 Sumps Kinect V1 [WWW] <http://www.i-programmer.info/news/194-kinect/7542-kinect-sdk-2-dumps-kinect-v1.html> (22.04.2015)
21. Lapuškin, M. (2012). Application for Gesture Based Control of the „Pioneer” Robot with Manipulator – Magistritöö, *Tallinn, Eesti*
22. Lapuškin, M. (2009). Application for Control of the „Pioneer” Robot with Manipulator – Bakalaureusetöö, *Tallinn, Eesti*
23. Magnetic motion capture [WWW] <http://www.metamotion.com/motion-capture/magnetic-motion-capture-1.htm> (27.02.2014)
24. Marker placement guide [WWW] <http://mocap.cs.cmu.edu/markerPlacementGuide.pdf> (15.05.2015)
25. Motion Capture [WWW] <http://web.mit.edu/comm-forum/papers/furniss.html> (26.02.2014)
26. Motion Capture (Mocap) [WWW] <http://caig.cs.nctu.edu.tw/course/CA/Lecture/mocap.pdf> (27.02.2014)
27. Near Mode: What it is (and isn't) [WWW] <http://blogs.msdn.com/b/kinectforwindows/archive/2012/01/20/near-mode-what-it-is-and-isn-t.aspx> (03.03.2014)
28. Retrieving the player orientation [WWW] <http://3carrotsonastick.wordpress.com/2012/10/21/retrieving-the-player-orientation/> (01.12.2014)

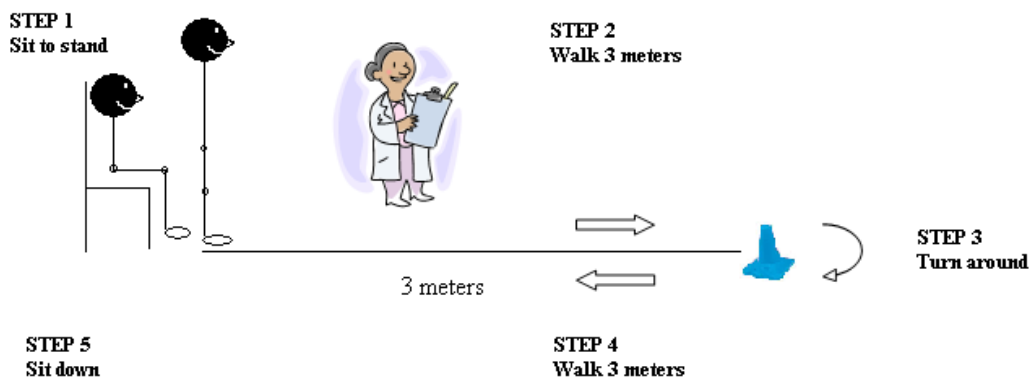
29. Rotation Matrices [WWW] <http://www.continuummechanics.org/cm/rotationmatrix.html> (10.04.2015)
30. Room lighting conditions for Kinect [WWW] <http://support.xbox.com/en-US/xbox-360/kinect/lighting> (03.03.2014)
31. Roosipuu, K. (2013). Interaktiivne Tarkvara Visuaalprogrammeerimise keskkondade juhtimiseks Kinecti abil – *Bakalaureusetöö, Tallinn, Eesti*
32. Satyavolu, S., Bruder, G., Willemsen, P., Steinicke, F. (2012). Analysis of IR-based virtual reality tracking using multiple Kinects. - *Virtual Reality Conference - Short Papers and Posters: 04.03-08.03.2012, Costa Mesa, Ameerika Ühendriigid, 149 – 150*
33. Singular Value Decomposition [WWW] [http://en.wikipedia.org/wiki/Singular\\_value\\_decomposition](http://en.wikipedia.org/wiki/Singular_value_decomposition) (05.11.2014)
34. Skeleton Tracking With Multiple Kinect Sensors [WWW] <https://msdn.microsoft.com/en-us/library/dn188677.aspx> (11.03.2015)
35. Translation (geometry) [WWW] [http://en.wikipedia.org/wiki/Translation\\_%28geometry%29](http://en.wikipedia.org/wiki/Translation_%28geometry%29) (24.04.2015)
36. Use your Kinect for Xbox 360 with front-projection TVs [WWW] <http://support.xbox.com/en-US/xbox-360/kinect/front-projection-tv> (31.03.2015)
37. Wang, C., Yeung, K-Y., Kwok, T-H. (2013). Improved Skeleton Tracking by Duplex Kinects: A Practical Approach for Real-Time Applications. - *Journal of Computing and Information Science in Engineering, 13 (4)*
38. What is motion capture [WWW] <http://www.ptiphoenix.com/what-is-motion-capture/> (26.02.2014)

## Lisa 1

### Up-and-Go test

Get-up-and-go test ehk tähendab inglise keeles “tõuse ja kõnni” test. See on kehaliste toimingute test, mida kasutatakse liikuvuse, tasakaalu ning lokomotoorsete tegevuste hindamiseks eakatel inimestel, kellel esineb tasakaalu häireid eesmärgiga neid käsitleda ning vältida traumasid [8].

Test ning selle läbiviimine on lihtne. Patsiendi ülesandeks on püsti tõusta toolilt (mis ei tohi olla asetatud vastu seina), kõndida 3 meetrit, ümber pöörata, kõndida tagasi tooli juurde ning maha istuda (vt. Joonis 20).



Joonis 20: Tõuse ja kõnni testi sooritamine [12]

Testi tuleks sooritada nii kiiresti ja ohutult kui seda patsiendi seisund võimaldab. Patsient peaks kandma oma tavapäraseid jalanõusid ning tohib kasutada kõndimiseks mõeldud abivahendit (kui ta kasutab seda igapäevaselt). Testi sooritamisel kõrvalist abi ei anta.

On olemas mitmeid variatsioone testist ning see, millist neist kasutatakse sõltub ravi või teadustöö eesmärgist. Näiteks üheks variatsiooniks on see, kui sooritajale antakse ette kindel kestvusaeg, ning ta peab testi sooritama nii mitu korda järjest kui ta suudab. Veel üks liik on ajastatud tõuse ja kõnni test, mille sooritamise ajal võtab hindaja aega, et teada saada palju sooritamiseks kulus aega. Selle põhjal, kui palju kulus sooritajal aega, saab tõlgendada testi tulemusi.

Võimalikke tõlgendusi saab näha Tabelis 1.

| Aeg  | Tõlgendus                                                                                                                                                                                                                                                  |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <10s | <b>Täielikult iseseisev</b><br>Patsient ei nõua kõrvalist abi tegevuste sooritamiseks, võib kuid ei pruugi kasutada kõndimiskeppi või muud kõndimiseks mõeldud abivahendit.                                                                                |
| <20s | <b>Saab peamiste tegevustega ise hakkama</b><br>Patsient võib, kuid ei pruugi kasutada kõndimiskeppi või muud kõndimiseks mõeldud abivahendit. Ta saab hakkama peamiste tegevustega ise, suudab kasutada peaagu kõiki treppe ning minna välja iseseisvalt. |
| >30s | <b>Nõuab abi</b><br>Patsient nõuab kõrvalist abi paljude tegevuste sooritamiseks.                                                                                                                                                                          |

**Tabel 1: TUG testi tõlgendamine [12]**

Kui tegemist ei ole ajastatud testiga, siis hindamine käib 5 palli süsteemis. Hinnangut annab testi vaatleja. Võimalikke hindeid ning nende lahtiseletusi saab näha Tabelis 2 [8].

| Hinne | Tõlgendus                                                                                                                                           |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 1     | Normaalne liikumine, pole mingit kukkumise ohtu                                                                                                     |
| 2     | Üldiselt ohutu ning ettevaatlik liikumine, veidi normist kõrvalekalduv                                                                              |
| 3     | Nähtavalt normist kõrvalekalduv – liikumine võib olla kiirustav ning võib märgata mitteregulaarseid ja mittekoordineeritud liigutusi nagu tuikumine |
| 4     | Nõuab kõrvalist abi, on märgata raskusi püsti tõusmisega ja maha istumisega                                                                         |
| 5     | Suur kukkumise oht, vajab kindlasti kõrvalist abi                                                                                                   |

**Tabel 2: Get-up-and testi tõlgendamine [8]**