

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Kris-Gerhard Aabrams 134666 IAPB

INIMKEHA MUDELI KORREKTSUSE VALIDEERIMINE

bakalaureusetöö

Juhendaja: Sven Nõmm
PhD

Tallinn 2017

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.

Autor: Kris-Gerhard Aabrams

22.05.2017

Annotatsioon

Antud lõputöös käsitletakse inimkeha mudeli registreerimisel esinevaid probleeme ning kirjeldatakse meetodeid, kuidas vigaselt registreeritud mudeleid algoritmiliselt tuvasta. Eesmärgiks oli sellise algoritmi väljatöötamine ja implementeerimine, mis lihtsustaks ja kiirendaks inimkeha mudeli kasutajate tööd. Töö tulemusena valmis tarkvara, mis vastab iga registreeritud kaadri kohta küsimusele, kas kaader on korrektne või mitte. See kiirendab ja lihtsustab oluliselt mudeli kasutajate töövoogu.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 36 leheküljel, 6 peatükki, 10 joonist, 9 tabelit.

Abstract

Correctness analysis of captured skeleton model

Capturing the skeleton model of a human body often results in an incorrect model. This happens due to many problems related with the method of capturing. Users have to validate these models frame-by-frame before they can be used. Depending on the length of the recording, the validation process can be slow and inefficient. The aim of this thesis is to develop and implement an algorithm to automate the validation process.

The author describes the method for detecting invalid recordings and how to distinguish incorrect frames from correct ones. Furthermore, the author describes how these theoretical methods are implemented in the created software and which kind of output files the software is capable of producing.

The product of this thesis is a software, which can capture the skeleton model using Kinect for Xbox One, play it back and perform a correctness analysis. This software offers correctness feedback for every recorded joint and a conclusive prediction for each frame. The conclusive prediction is based on a logistic regression analysis which can be trained by the user.

The created software greatly simplifies and speeds up the process every human skeleton model analyzer has to perform.

The thesis is in Estonian and contains 36 pages of text, 6 chapters, 10 figures, 9 tables.

Lühendite ja mõistete sõnastik

SDK	<i>Software Development Kit</i> , tarkvaraarenduspakett
API	<i>Application Programming Interface</i> , rakendusliides
Enum	<i>Enumeration</i> , numeratsioon
rad	radiaan
m	meeter
s	sekund

Sisukord

1 Sissejuhatus	9
2 Mudeli registreerimisel esinevad probleemid.....	10
3 Meetod.....	12
3.1 Loogiline valideerimine.....	13
3.2 Statistiline valideerimine	14
3.3 Valideerimise tulemus	15
4 Rakendus	17
4.1 Valideerimiskomponent.....	18
4.1.1 Mõõtepunktide ja luude valideerimine	18
4.1.2 Logistilise regressiooni abil kogu kaadri kohta ennustuse tegemine	19
4.2 Mudeli vaated	20
4.2.1 Taasesitusvaade	21
4.2.2 Reaalaja vaade	22
4.3 Väljundfailid.....	23
4.3.1 <i>Joint position and orientation</i> ehk punktide asukoht ja pööre	24
4.3.2 <i>Joint status</i> ehk punktide staatus	25
4.3.3 <i>Bone length</i> ehk luupikkused.....	26
4.3.4 <i>Joint angle</i> ehk nurgad	27
5 Rakenduse tulemuse valideerimine	29
6 Kokkuvõte	32
Kasutatud kirjandus	33
Lisa 1 – Mõõtepunktide eesti keelsed vasted	34
Lisa 2 – Enum <i>JointType</i>	35
Lisa 3 – Enum <i>BoneType</i>	36
Lisa 4 – Enum <i>AngleType</i>	36

Jooniste loetelu

Joonis 1 Näide ebakorrektselt skelettmudelilt	10
Joonis 2 Näide korrektsest skelettmudelilt.....	11
Joonis 3 Mõõtepunktid	12
Joonis 4 Menüü vaade	18
Joonis 5 Valideerimise värve illustreeriv eest vaade.....	20
Joonis 6 Taasesitlus vaade.....	22
Joonis 7 Reaalaja vaade	23
Joonis 8 Väljundfailide valiku vaade.....	24
Joonis 9 Järjestikused kaadrid esimesest eksperimentist: (a) korrektseks märgitud kaader, (b) eelmisele järgnev ning mittekorrektseks märgitud kaader	30
Joonis 10 Näide eksperiment 2 käigus registreeritud kaadrist ja selle valideerimise tulemusest	31

Tabelite loetelu

Tabel 1 Mõõdetavate nurkade aluseks olevad mõõtepunktid ja piirväärtused.....	13
Tabel 2 Mõõtepunktide vahele konstrueeritud luude standardhälbed.....	14
Tabel 3 Mudeli värvi ja valideerimise tulemuse vastavus.....	20
Tabel 4 Faili „ <i>jointPositionOrientation</i> “ formaat.....	24
Tabel 5 Faili „ <i>jointStatus</i> “ formaat.....	26
Tabel 6 Faili „ <i>boneLength</i> “ formaat.....	26
Tabel 7 Faili „ <i>jointAngle</i> “ formaat.....	27
Tabel 8 Faili „ <i>jointAngleWithPos</i> “ formaat.....	28
Tabel 9 Mõõtepunktid inglise ja eesti keeles.....	34

1 Sissejuhatus

Inimese skelettmudeli registreerimine on odavam ja lihtsam kui kunagi varem. Rakendused, mis sellist mudelit kasutavad, on väga erinevad – meelelahutuslikud mängud, meditsiinilised rakendused jne.

Kahjuks esinevad mudeli registreerimisel mitmesugused probleemid, mille tulemuseks on tihti salvestis, kus mõõtepunktide asukoht ei vasta tegelikkusele, liigeste nurk on ebainimlik või luude pikkus varieerub kümnetes sentimeetrites. Seetõttu vajavad rakendused kinnitust, et aluseks olev mudel on töötlemiseks kõlbulik. Veendumaks, et mudel on korrektne, tuleb töötlejal salvestis käsitsi kaader-kaadri haaval läbi vaadata. See on ebamugav ja ajakulukas tegevus.

Töö eesmärgiks on luua algoritm ja tarkvara, mis valideerib sisendina saadud keha mudeli lihtsalt ja kiirelt. Tarkvara annab kasutajale teada missuguste punktide asukohad on usaldusväärsed ja missuguste mitte. Rakendus annab kokkuvõtliku hinnangu ka iga kaadri kohta. Lisaks valideerimisele saab rakendusest ise mudeli salvestada (kasutades seadet Microsoft Kinect for Xbox One) ning salvestisi taasesitada. Töö on seotud TTÜ projektiga B37 „Raamistik inimmotorika kvantitatiivseks analüüsiks“.

Teises peatükis teeb autor ülevaate, mis on levinumad probleemid inimkeha skelettmudeli registreerimisel ja toob visuaalsed näited korrektsest ja mittekorrektsest mudelist.

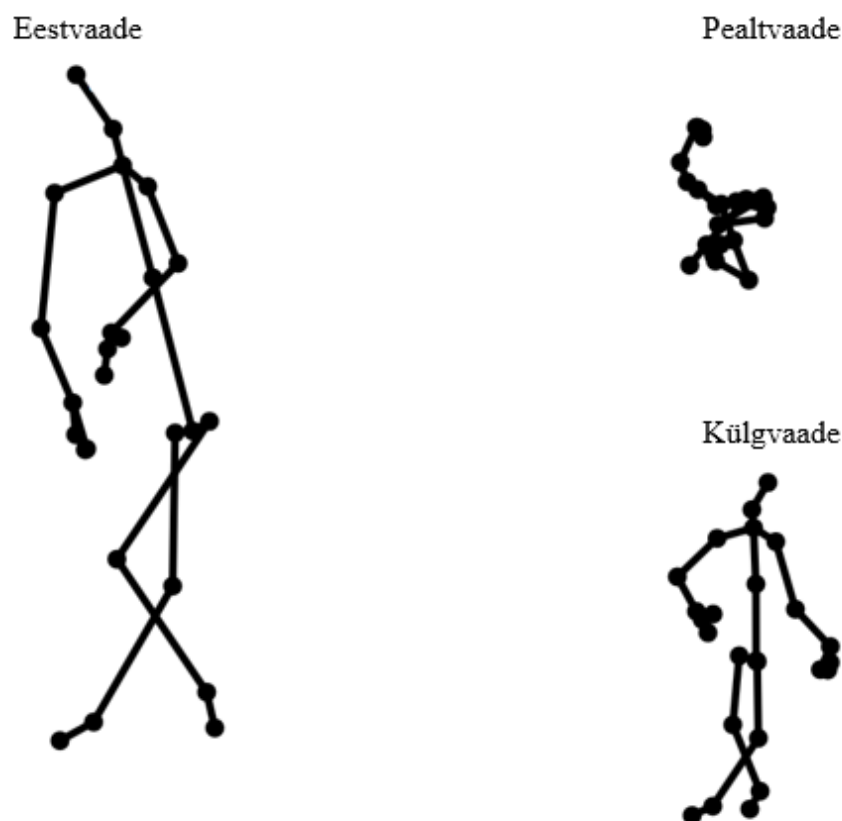
Kolmandas peatükis kirjeldatakse meetodeid, kuidas sisendiks saadud andmete põhjal skelettmudelit valideerida.

Neljas peatükk annab ülevaate realiseeritud rakendusest – kuidas on realiseeritud eelnevas kolmandas peatükis kirjeldatud meetodid, millised erinevad vaated on rakenduses ning missuguseid faile on võimalik koostada.

Viiendas peatükis valideeritakse eksperimentaalselt rakenduse tulemus, andes rakendusele sisendiks kontrollitud keskkonnas tehtud salvestisi.

2 Mudeli registreerimisel esinevad probleemid

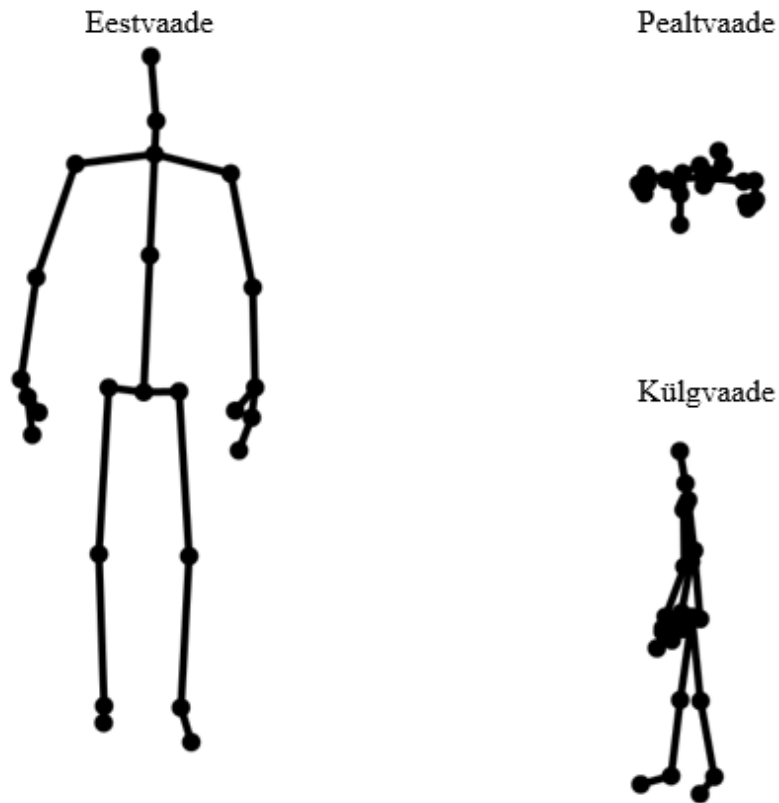
Varjutus (ingl k. *eclipse*) on üks levinumatest probleemidest liigutuste registreerimisel [1]. See probleem esineb olukorras, kus mõõtepunkt on kaamera vaates millegi taga peidus. Peitvaks objektiks võib olla näiteks mõni teine mõõtepunkt või ruumis olev objekt. Näiteks keerates kaamerale külje, varjame kaamera jaoks pooled mõõtepunktid, mistõttu on mudel tihti vale. Vaata näitena Joonist 1, kus modelleeritava keha üks käsi ja põlv on kaamera jaoks teise taga peidus. Tulemuseks on ebakorrektselt registreeritud mudel. Korrektse mudeli näide on Joonisel 2.



Joonis 1 Näide ebakorrektselt registreeritud skelettmodellist

Üheks laialt levinud võimaluseks inimese mudeli registreerimiseks on kasutada statistilist tuvastusalgoritmi. Antud töös kasutatud Kinect kasutab mudeli koostamiseks sama meetodit [2]. Sellisel juhul ei tuvasta kaamera mitte konkreetseid punkte, vaid näeb

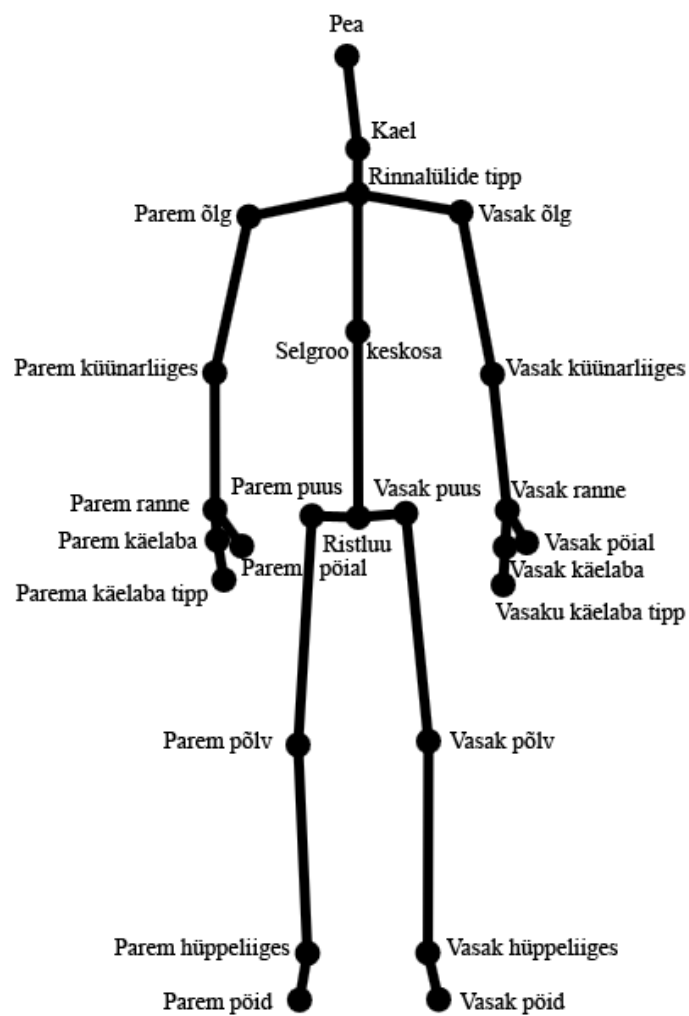
punktide pilve. Toetudes tõenäosusarvutustel pakub algoritm kus üks või teine punkt võiks asuda. Tavaolukorras on selliste algoritmide põhjal saadud mudelid väga täpsed, kuid olukorras, kus näiteks esineb varjutus ja algoritm ei ole oma tulemustes kindel, esineb vigu. Vigu mudeli registreerimisel on võimalik esile kutsuda tehes pöördeid ümber oma telje või lihtsalt kiirelt liigutades.



Joonis 2 Näide korrektsest skelettmodellist

3 Meetod

Mudeli valideerimise sisendiks on kahekümne viie mõõtepunkti (vt Joonis 3) positsioon ja pööre. Pööret on tarvis, kuna kolmemõõtmelises ruumis nurga arvutamiseks vahemikus 0-360° on tarvis teada suunda, mille suhtes nurk võetakse [3]. Mudeli valideerimiseks kasutan kahte peamist meetodikat – loogiline- ja statistiline valideerimine.



Joonis 3 Mõõtepunktid

3.1 Loogiline valideerimine

Inimkeha liigutustel on piirid. Näiteks põlved ega küünarnukid ei saa minna nõ tagurpidi ning me ei saa lehvitada kätt helikiirusel. Loogilise valideerimise käigus valideeritakse mõõtepunktide kiirendust, punktide vahele konstrueeritud nurkade suuruseid ja kiirendusi.

Mudelile konstrueeritakse kümme nurka ja mõõdetakse nende arvvaartusi ja kiirendusi. Nurkade piirvaartuste aluseks on võetud inimese joonistamise õpetus [4] ning täpsed piirvaartused (vt Tabel 1) on leitud mitmete katsetuste käigus, kus erinevatel isikutel on palutud sooritada mitmesuguseid harjutusi.

Nurkade ja mõõtepunktide kiirenduste piirvaartus on leitud samuti katsetuste käigus, kus modelleeritav keha tegi nii äkilisi liigutusi kui võimalik ning leidsin, et korrektse ja ebakorrekse mudeli kiirenduste piiriks on nurkade puhul 90 rad/s^2 (radiaan/sekund²) ning mõõtepunktide puhul 40 m/s^2 (meeter/sekund²). Järgnevas tabelis on loetletud kõik kontrollitavad nurgad ja nende lubatud piirvaartused. Nurk konstrueeritakse „Mõõtepunkt 2“ juurde.

Tabel 1 Mõõdetavate nurkade aluseks olevad mõõtepunktid ja piirvaartused

Mõõtepunkt 1	Mõõtepunkt 2	Mõõtepunkt 3	Min. väärtus (°)	Maks. väärtus (°)
Vasak puus	Vasak põlv	Vasak hüppeliiges	25	190
Parem puus	Parem põlv	Parem hüppeliiges	25	190
Vasak põlv	Ristluu	Pea	40	270
Parem põlv	Ristluu	Pea	40	270
Vasak põlv	Ristluu	Parem põlv	0	170
Selgroo keskosa	Rinnalülide tipp	Pea	130	180
Ristluu	Selgroo keskosa	Rinnalülide tipp	172	180
Vasak ranne	Vasak küünarliiges	Vasak õlg	30	180
Parem ranne	Parem küünarliiges	Parem õlg	30	180

3.2 Statistiline valideerimine

Statistiline valideerimine hõlmab endast statistika kasutamist korrektsete ja mittekorrektsete mõõtmiste eristamiseks. Mõõdetakse punktide vahelisi kaugusi ning kuna need kattuvad enamasti anatoomiliselt luudega, ongi neid edaspidi selliselt nimetatud.

Läbi on viidud katsed viie erineva kehatüübiga inimesega ja registreeritud nende keha mudelid, mis on käsitsi valideeritud ning korrektseks tunnistatud. Nendel mudelitel mõõdeti luude pikkuse keskmist erinevust keskmisest väärtusest ehk standardhälvet. Tulemusi analüüsid selgus, et puudub korrelatsioon inimese pikkuse ja luude pikkuse varieeruvuse vahel. Samuti selgus, et mõõtetulemuste jaotumine sarnaneb normaaljaotusele. Seetõttu on käesoleva töö autor otsustanud, et lubatud erinevus keskmisest on maksimaalselt kolm standardhälvet, mis katab normaaljaotuse korral 99.7% mõõtmistest. Iga luu standardhälve on fikseeritud suurus, milleks on läbiviidud katsete käigus leitud suurim väärtus (vt Tabel 2).

Tabel 2 Mõõtepunktide vahele konstrueeritud luude standardhälbed

Mõõtepunkt 1	Mõõtepunkt 2	Standardhälve (m)
Pea	Kael	0.011863631473322
Kael	Rinnalülide tipp	0.001796107888618
Rinnalülide tipp	Selgroo keskosa	0.005116526095936
Selgroo keskosa	Ristluu	0.00897811395399
Ristluu	Vasak puus	0.006964116845016
Vasak puus	Vasak põlv	0.028932328145078
Vasak põlv	Vasak hüppeliiges	0.031156351362592
Vasak hüppeliiges	Vasak põid	0.017642641730294
Ristluu	Parem puus	0.00465586051349
Parem puus	Parem põlv	0.033407904911811
Parem põlv	Parem hüppeliiges	0.035330459610187
Parem hüppeliiges	Parem põid	0.017681729592529

Mõõtepunkt 1	Mõõtepunkt 2	Standardhälve (m)
Rinnalülide tipp	Vasak õlg	0.006752473511202
Vasak õlg	Vasak küünarliiges	0.01836959465422
Vasak küünarliiges	Vasak ranne	0.014140684084087
Vasak ranne	Vasak käelaba	0.020174204362611
Vasak käelaba	Vasaku käelaba otspunkt	0.011079360245912
Vasak ranne	Vasak põial	0.028785442223362
Rinnalülide tipp	Parem õlg	0.009176872898877
Parem õlg	Parem küünarliiges	0.010803663285133
Parem küünarliiges	Parem ranne	0.014471289202014
Parem ranne	Parem käelaba	0.013027450947962
Parem käelaba	Parema käelaba otspunkt	0.007983560996173
Parem ranne	Parem põial	0.021260029999708

3.3 Valideerimise tulemus

Kahel eelnevalt nimetatud viisil on saadud väga täpne tulemus, mis on kõiksugustele varieeruvustele väga tundlik. Modelleerimisel esineb harva olukord, kus kõik valideeritavad parameetrid on korrektsed, mistõttu ei saa mudeli töötledjad endale nõnda suurt tundlikust lubada. Seega tuleb kogutud andmete põhjal luua vähem tundlik - kokkuvõtlik hinnang. Kokkuvõtva hinnangu tegemiseks kasutan masinõppe algoritmi. Kuna antud juhul on võimalikke tulemusi kaks – korrektne ja mittekorrektne, siis on kasutatud logistilise regressiooni meetodit [5].

Algoritmi õpetamiseks, peab logistilise regressiooni mudel teadma, millised kaadrid on korrektsed ja millised mitte. Selleks on käsitsi valideeritud 1500 kaadriga salvestis. Mudel saab sisendiks mõõtepunktide kiirendused, luude erinevused keskmisest, nurkade väärtused ning nurkade kiirendused, seega kokku $25+24+10+10 = 69$ sisendit, mida on liialt palju. Et sisendite arvu vähendada, eemaldatakse kõige väiksema usaldusväärsusega sisend ning arvutatakse uus mudel. Sisendi usaldusväärsuse mõõtmiseks on kasutatud

Wald testi p-väärtust [6]. Seda protsessi korratakse seni, kuni kõikide kasutatavate sisendite Wald testi p-väärtus on maksimaalselt 5%.

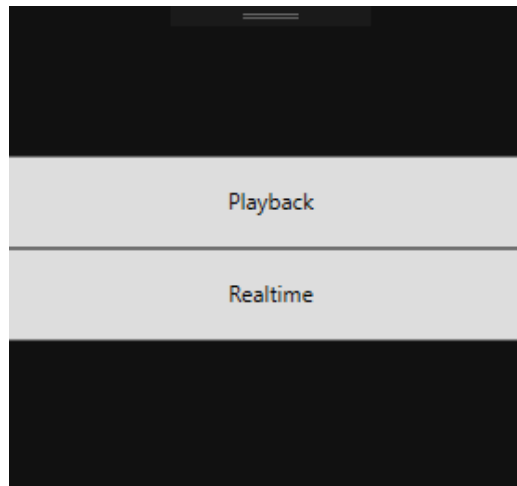
Rakendusega tuleb kaasa käesoleva töö autori loodud mudel, kuid kasutajal on võimalik ka ise valideeritud salvestise põhjal uus logistilise regressiooni mudel luua. Mudeli loomine on kirjeldatud leheküljel 21.

4 Rakendus

Töö raames on mudelite salvestamiseks kasutatud Microsoft Kinecti (täpsemalt Kinect for Xbox One) ning kuna rakendus peab valideerima ka reaalsajas registreeritud salvestisi on rakendus arendatud C# programmeerimiskeeles kasutades Visual Studio arenduskeskkonda. Valiku tegemine oli lihtne, sest Microsofti Kinect For Windows SDK (*Standard Development Kit*) 2.0 Windows Runtime API (*Application Programming Interface*) teeb Kinectiga suhtlemise lihtsaks ning ka API dokumentatsioon on põhjalik [7]. Töö autor otsustas rakenduse kasutajaliidese ja programmikoodi kirjutada inglise keeles, et seda saaks kasutada võimalikult paljud soovijad ning et tehtud tööd oleks võimalik ka teistel edasi arendada. Rakenduse käivitamiseks on vajalik Kinect for Windows Runtime 2.0.

Rakendus võimaldab valideerida eelnevalt registreeritud mudelit ning ka salvestada reaalsajas valideeritud mudeli. Valideerimise eest vastutab vastav komponent, mille sisu on kirjeldatud jaotises 4.1.

Rakenduses on kokku neli vaadet – avamenüü, taasesitusvaade, reaalsaja vaade ja väljundfailide valiku vaade. Menüüvaade avaneb rakendust avades ning sellel tuleb teha valik taasesitusvaate ja reaalsaja vaate avamise vahel. Taasesitus- ja reaalsaja vaade ehk mudelivaated on täpsemalt kirjeldatud jaotises 4.2. Väljundfailide valiku vaadet on võimalik avada mõlemast mudelivaatest ning sellelt on võimalik valida, milliseid faile ja kuhu salvestada soovitakse. Vaade ning valikus olevate failide sisu on kirjeldatud jaotises 4.3.



Joonis 4 Menüü vaade

4.1 Valideerimiskomponent

Valideerimiskomponent on ülejäänud programmist võimalikult sõltumatu. See on arendatud eraldi klassina, millel on kolm avalikku (public) meetodit – *ValidateBody*, *ClearHistory* ja *GetBodiesAndBoneData*.

Meetod *ValidateBody(MyBody)* võtab sisendiks *MyBody* objekti, valideerib selle ning täidab valideerimistulemuste muutujad. Komponenti mälu saab kustutada meetodiga *ClearHistory()*, mis käsib komponendil unustada kõik eelnevalt valideeritud kaadrid. Meetod *GetBodiesAndBoneData()* tagastab kõik valideeritud kaadrid koos vastava keha luude kohta kogutud statistilise informatsiooniga. Täpsemalt tagastatakse *Dictionary*, kus võtmeks on *List* valideeritud *MyBody* objektidest ning väärtuseks jällegi *Dictionary*, mille võtmeks *enum BoneType* ja väärtuseks *BoneLengthData* objekt, kus on muutujad *Mean* (keskmine), *StandardDeviation* (standardhälve), *CoefficientVariation* (variatsioonikoefitsient).

4.1.1 Mõõtepunktide ja luude valideerimine

Valideeritakse keha mõõtepunktid ja nende vahele konstrueeritud luud. Iga mõõtepunkti asukoha järgi arvutatakse kiirus ja kiirendus. Nendeks arvutusteks on tarvis eelmise asukoha olemasolu. Kui komponendil puudub info sama keha eelmise kaadri kohta, siis on nende arvutuste tulemuseks 0. Suurim lubatud kiirendus on 40 m/s^2 , suurema absoluutväärtuse korral märgitakse mõõtepunkt mittekorrektseks. Piirväärtus on leitud

katsetamise käigus ning selle leidmine on pikemalt kirjeldatud jaotises 3.1. Valideerimise tulemus salvestatakse muutujas *JointData*.

Keha mõõtepunktide vahel kontrollitakse kokku kümmet erinevat nurka. Valideeritakse nii nende arvvaartust kui ka kiirendust. Iga nurka iseloomustab kolm mõõtepunkti ning minimaalne ja maksimaalne väärtus. Piirväärtuste leidmine on kirjeldatud jaotises 3.1. Nurga arvvaartuse leidmiseks kasutatakse algoritmi, mille sisendiks on kolm mõõtepunkti ja keskmise mõttepunkti pööre. Pöörde vektor saadakse kasutades John A. Berry kvaterniooni pööramise algoritmi [8]. Nurga valideerimise tulemus salvestatakse muutujas *JointAngleData*.

Veel valideeritakse mõõtepunktide vahele konstrueeritud luude pikkuse muutumist. Iga luu kohta arvutatakse välja tema pikkus ning võrreldakse seda teadaolevate väärtuste keskmisega. Kui luu pikkus erineb keskmisest rohkem kui kolme standardhälbe jagu (vt Tabel 2), siis märgitakse luu mittekorrektseks. Standardhälbe leidmise ja kolme kordse erinevuse reegel on pikemalt seletatud jaotises 3.2. Esimese kaadri korral on valideerimise tulemus alati tõese väärtusega, kuna puudub võimalus võrrelda keskmisega.

4.1.2 Logistilise regressiooni abil kogu kaadri kohta ennustuse tegemine

Et vastata küsimusele, kas terve mudel on korrektne või mitte, kasutatakse logistilise regressiooni mudelit (vt jaotist 3.3), mis asub programmi juurkaustas asuvas „*logistic regression*“ nimelises kaustas. Samas kaustas asuvad ka sisendite nimekirju sisaldavad failid, mida mudelisse tuleks vastuse saamiseks anda. Algsed failid on autori poolt loodud, kuid jaotises 4.2.1 on kirjeldatud, kuidas luua uus mudel ja sisendid. Valideeritud *MyBody* objektilt võetakse regressiooni mudeli sisendiks vajalikud parameetrid ja antakse need mudelile ette, mille peale mudel vastab tõeväärtustüüpi muutujaga. Tegemist on ennustusega, kas kogu kaader on korrektne või mitte. Vastus seatakse *MyBody* objekti *RegressionValidationResult* muutuja väärtuseks.

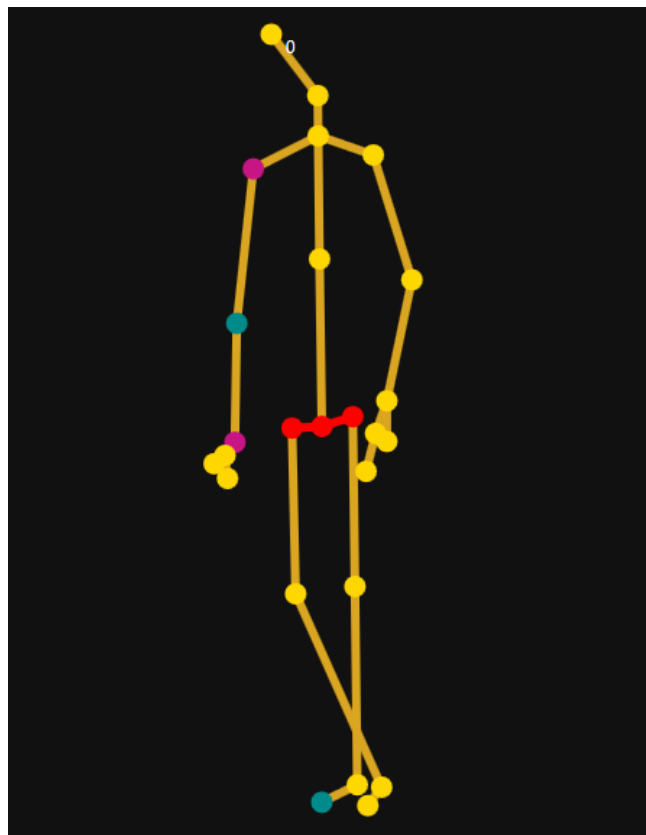
Lõpetuseks väärtustatakse keha objekti muutuja *Validated* tõese väärtusega, mis viitab, et keha mudel on valideeritud ning lisatakse keha valideerimiskomponendi mällu, et järgmise kaadri saabumisel saaks teha eelmisest kaadrist lähtuvaid arvutusi.

4.2 Mudeli vaated

Rakenduses on kaks mudeli vaadet – taasesitus- ehk *playback* vaade ja reaalaaja- ehk *realtime* vaade. Mõlemas vaates kuvatakse lisaks vaate spetsiifilistele elementidele valideeritud mudel kolmelt erinevalt suunalt – eest, pealt ja vasakult. Kuvatav mudel on värvitud vastavalt valideerimise tulemusele. Tabel 3 esitab värvide ja valideerimise tulemuse vastavuse ning Joonisel 5 on värve illustreeriv ekraaniväljavõte rakendusest.

Tabel 3 Mudeli värvi ja valideerimise tulemuse vastavus

	Värv	Tingimus
Mõõtepunkt	Helekollane	Mõõtepunkt on korrektne.
	Tsüaansinine	Punkti kiirendus on lubatust suurem.
	Violetne	Punktiga seotud nurk ei ole korrektne.
	Punane	Punktiga seotud luu ei ole korrektne.
Luu	Kuldne	Luu on korrektne.
	Punane	Luu ei ole korrektne.



Joonis 5 Valideerimise värve illustreeriv eest vaade

4.2.1 Taasesitusvaade

Taasesitusvaade on mõeldud eelnevalt registreeritud salvestiste taasesitamiseks ning valideerimiseks. Sellest vaatest on võimalik valideerimise ka tulemus faili salvestada.

Vaadet avades kuvatakse koheselt faili avamise dialoogaken, kuid seda on võimalik ka ise avada vajutades nupule „*Open file*“ (vt Joonis 6). Avatav fail peab olema korrektses vormis, mis on sama väljundfailiga „*jointPositionOrientation*“ (vt jaotisest 4.3.1). Faili avamisel luuakse iga korrektselt teisendatud rea kohta üks *MyBody* objekt. See on analoog Kinect API *Body* objektile (vt *Body* objekti seletust jaotisest 4.2.2). Selle järgnevalt valideeritakse koheselt kõik koostatud *MyBody* objektid.

Valideerimise lõppedes kuvatakse kasutajale esimene valideeritud kaader. Kasutajal on võimalus salvestis mängima ja seisma panna vajutades „*Start/Stop*“ nuppu (vt Joonis 6). Kaadrid taasesitatakse sama sagedusega, millega nad ka salvestati, milleks on keskmiselt 30 kaadrit sekundis. Vaate allservas oleva liuguriga on võimalik valida ka konkreetset kaadrit ning peale liuguril klikkimist on „*Page up*“ ja „*Page down*“ klahve vajutades võimalik liikuda kaader haaval vastavalt edasi või tagasi.

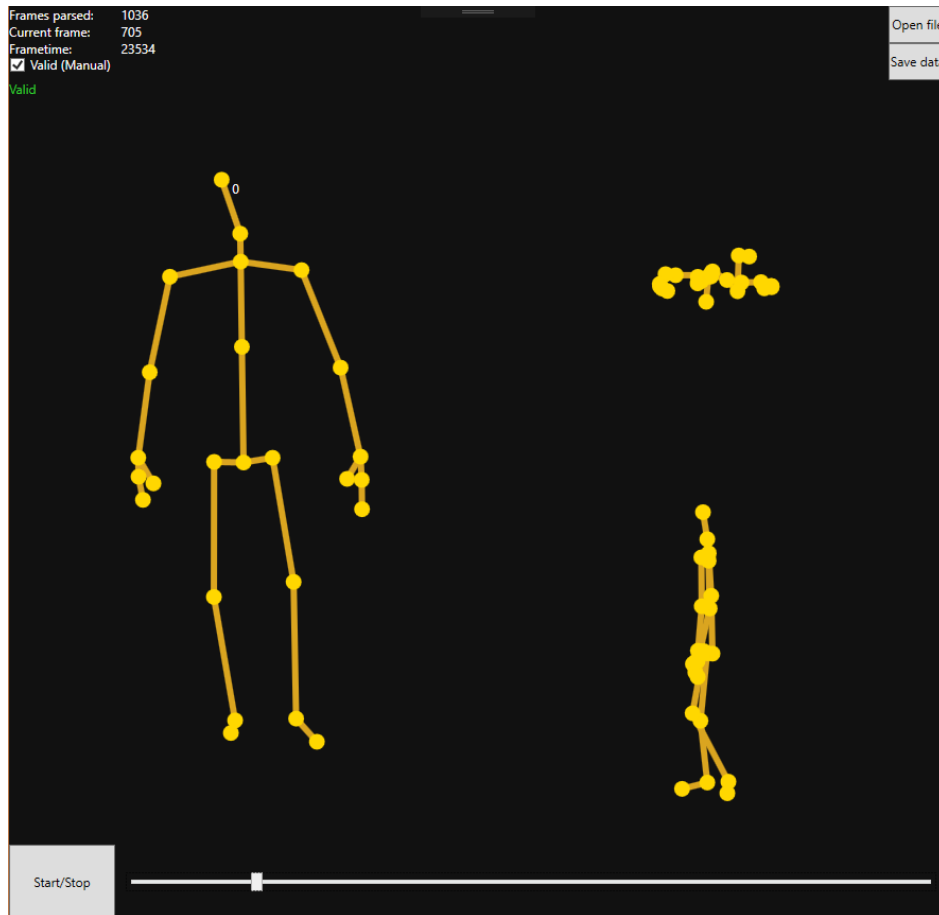
Kasutajal on võimalus valideerimise tulemus salvestada faili vajutades „*Save data*“ nupule. Selle peale avaneb väljundite valiku vaade, mille sisu on kirjeldatud jaotises 4.3.

Vasakus ülemises nurgas kuvatakse numbrilist informatsiooni salvestise kohta. „*Frames parsed*“ väärtus näitab mitu edukalt teisendatud *MyBody* objekti loodi, „*Current frame*“ viitab hetkel pildis oleva kaadri järjekorranumbrile ja „*Frame time*“ on antud kaadri suhteline aeg võrreldes salvestamise algusega millisekundites. Lisaks kuvatakse linnukesega kaadri manuaalse valideerimise staatust ning logistilise regressiooni mudeli tulemust kas roheliselt „*Valid*“ või punaselt „*Invalid*“ sõltuvalt, kas valideerimise tulemus on korrektne või mitte.

Taasesitusvaates on võimalik mudelit ka manuaalselt valideerida. Vajutades klahvi „*K*“ muutub kaadri manuaalse valideerimise väärtus vastupidiseks. Antud väärtused salvestatakse „*jointPositionOrientation*“ faili ning leiavad kasutust logistilise regressiooni mudeli loomisel.

Vajutades klahvi „*F3*“ loob programm logistilise regressiooni mudeli, mida kasutatakse mudeli valideerimisel kogu kaadri kohta ennustuse tegemiseks. Loodud mudel ja

kasutatud sisendid salvestatakse kausta nimega „*logistic regression*“, mis asub rakenduse juurkaustas. Mudeli saamise meetod on kirjeldatud jaotises 3.3. Antud töös on mudeli loomiseks kasutatud Accord.NET Framework-i *LogisticRegressionAnalysis* klassi [9] ning mudeli ja sisendite faili salvestamiseks ja failist lugemiseks sama teegi *Serializer* klassi [10].



Joonis 6 Taasesitlus vaade

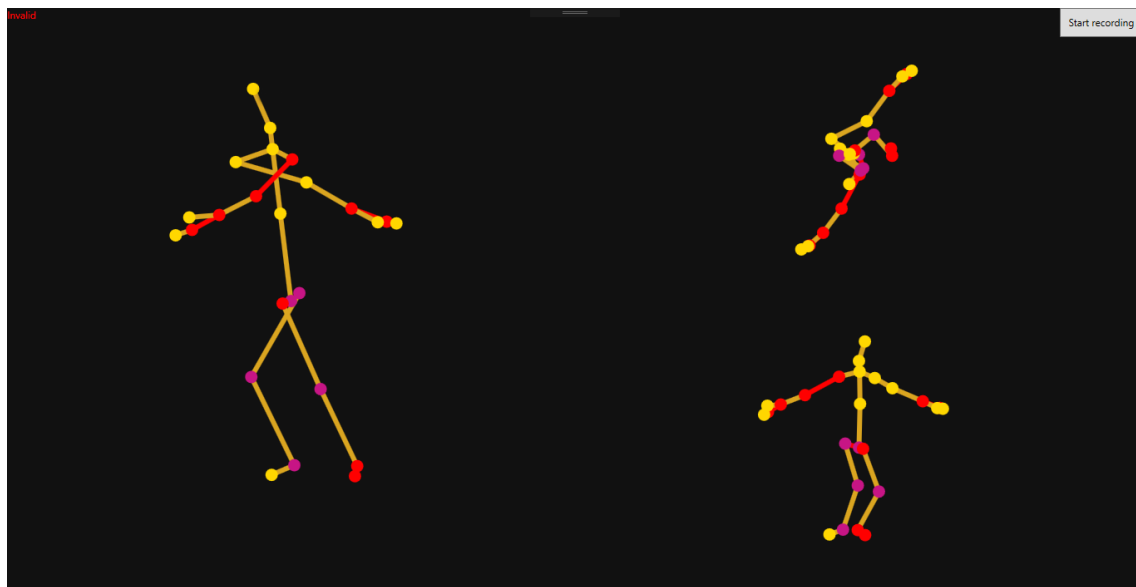
4.2.2 Reaalaja vaade

Reaalaja vaates kuvatakse reaalajas registreeritud ja valideeritud mudeli kujutis (vt Joonis 7). Arvutiga peab olema ühendatud Kinect for Xbox One ning samuti peavad olema täidetud minimaalsed nõuded Kinecti kasutamiseks [11]. Kui need tingimused on täidetud, saab rakendus Kinect API käest kaadris olevate inimeste põhjal loodud *Body* objektid, keskmiselt 30 korda sekundis. Nendelt objektidelt võetakse valideerimiseks vajalik informatsioon – mõõtepunktide (vt Lisa 1) asukohad, keha identifitseeriv number, kaadri järjekorra number, suhteline salvestusaeg salvestamise alguse suhtes millisekundites ning info mõõtepunktide pöörde kohta. Nende andmete põhjal luuakse

MyBody objekt, mis on võimeline hoidma endas lisaks eelnevale ka valideerimise andmeid. Seejärel valideeritakse loodud *MyBody* objekt ning kuvatakse see kasutajale.

Vaates kuvatakse ka logistilise regressiooni mudeli põhjal ennustatud tulemust kogu kaadri kohta. Vasakpoolses ülemises nurgas on kas roheliselt „Valid“ või punaselt „Invalid“ sõltuvalt, kas valideerimise tulemus on ennustuse kohaselt korrektne või mitte.

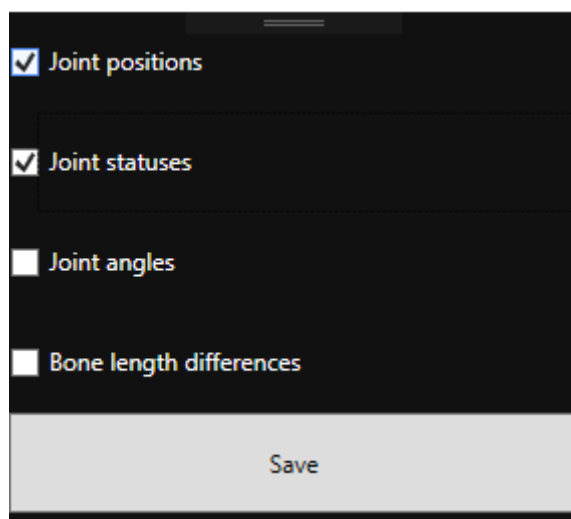
Kasutajal on võimalik ka registreeritud mudel faili salvestada vajutades nupule „Start recording“. Selle peale muudetakse nupu tekstiks „Stop recording“ ning nupule uuesti vajutades lõpetatakse salvestamine ja avaneb väljundfailide valiku vaade (vt jaotist 4.3). Failidesse kirjutatakse kahe nupu vajutuse vahel registreeritud kaadrite informatsioon.



Joonis 7 Reaalaja vaade

4.3 Väljundfailid

Kasutajal on mõlemast vaatest võimalus valideeritud kaadrite andmed faili kirjutada, lisaks sellele on valikus ka erinevad failid, mis sisaldavad valideerimise aluseks olnud andmeid, nagu näiteks mõõdetud nurgad ja luude pikkuste erinevus keskmisest. Vajutades kummalgi vaatel selleks vastavat nuppu, avaneb kasutajale uus vaade, millel on võimalik märkida, milliseid andmeid salvestada soovitakse (vt Joonis 8).



Joonis 8 Väljundfailide valiku vaade

Kõik väljundfailid on CSV formaadis. Iga kaadri info on eraldi real ning iga rida algab kaadri järjekorranumbri ja suhtelise registreerimisajaga võrreldes salvestamise algusajaga millisekundites. Kõik väärtused on failides üksteisest eraldatud komaga (','), mistõttu on numbrite kümnendkoha eraldajaks punkt ('.'). Kuna registreerida saab mitut keha korraga, on iga faili nime lõpus keha identifitseeriv number. Mõnedel failidel on faili alguses päis, et faili tabeli kujule üle viies anda parema ülevaade, mida konkreetne väärtus tähistab.

4.3.1 *Joint position and orientation* ehk punktide asukoht ja pööre

See fail on kõige olulisem, kuna sisaldab endast valideerimise tulemust. Fail erineb sisendfailist vaid iga rea lõpus oleva lisaväärtuse võrra. Igal real on lisaks kaadri infole ka iga mõõtepunkti asukoha X, Y ja Z koordinaadid ning mõõtepunktide pöörde X, Y, Z ja W väärtused. Mõõtepunktid on järjestatud vastavalt enumi *JointType* dikteeritud järjekorrale (vt Lisa 2 – Enum *JointType*). Otspunktiks oleva mõõtepunkti pöördevektorid on nullvektorid. Rea lõppu lisatud väärtus on kas „*True*“ või „*False*“ vastavalt, kas valideerimise käigus märgiti kaader on korrektseks või mitte.

Tabel 4 Faili „*jointPositionOrientation*“ formaat

Faili nime vorm	jointPositionOrientation_[KEHA_ID].csv
Näide faili nimest	jointPositionOrientation_72057594037928801.csv
Päise ridade arv	0

Rea sisu vorm	[KAADRI_NUMBER],[SUHTELINE_AEG], [MÕÕTEPUNKT1_X],[MÕÕTEPUNKT1_Y], [MÕÕTEPUNKT1_Z],[MÕÕTEPUNKT1_PÖÖRE_X], [MÕÕTEPUNKT1_PÖÖRE_Y], [MÕÕTEPUNKT1_PÖÖRE_Z], [MÕÕTEPUNKT1_PÖÖRE_W], ..., [MÕÕTEPUNKT25_X],[MÕÕTEPUNKT25_Y], [MÕÕTEPUNKT25_Z],[MÕÕTEPUNKT25_PÖÖRE_X], [MÕÕTEPUNKT25_PÖÖRE_Y], [MÕÕTEPUNKT25_PÖÖRE_Z], [MÕÕTEPUNKT25_PÖÖRE_W], [MANUAALSE_VALIDEERIMISE_VÄÄRTUS], [AUTOMAATSE_VALIDEERIMISE_VÄÄRTUS]
Näide rea sisust	548,18300, -0.08537506,-0.4445771,3.466888, -0.00146458,0.9998944,-0.0054303,-0.01340157, ..., 0.1881481,-0.4997416,3.318888, 0, 0, 0, 0, True, True

Lisaks salvestatakse seda valikut tehes ka ilma pöördevektoriteta, kuid muus osas identne fail. See fail kannab nime „*jointPosition*“.

4.3.2 *Joint status* ehk punktide staatus

Selle valiku puhul salvestatakse samuti kaks faili. Esimene neist sisaldab iga mõõtepunkti kohta teadaolevat valideerimise tulemust. See tähendab, et iga punkti kohta on vastatud kolmele küsimusele:

1. Kas punkti kiirendus jääb lubatud piiridesse?
2. Kas punktiga seotud luu pikkus jäi lubatud piiridesse? Punkt on seotud luuga, kui ta on luu otspunktiks.
3. Kas punktiga seotud nurk on korrektne? Punkt on seotud nurgaga, kui ta on üks kolmest punktist, mille põhjal nurk arvutati.

Kui vastus on tõene, kuvatakse „*True*“ ning kui vastus on väär, kuvatakse „*False*“. Kui mõõtepunkti kohta vastavat otsust tehtud ei olegi, ei kuvata midagi. Mõõtepunkti kohta puudub otsus, kui punkt pole seotud ühegi mõõdetava nurgaga. Mõõtepunktid on järjestatud vastavalt enumi *JointType* dikteeritud järjekorrale (vt Lisa 2 – Enum *JointType*).

Tabel 5 Faili „*jointStatus*“ formaat

Faili nime vorm	jointStatus_[KEHA_ID].csv
Näide faili nimest	jointStatus_72057594037928801.csv
Päise ridade arv	2
Rea sisu vorm	[KAADRI_NUMBER],[SUHTELINE_AEG], [MÕÕTEPUNKT1_KÜSIMUS1_VASTUS], [MÕÕTEPUNKT1_KÜSIMUS2_VASTUS], [MÕÕTEPUNKT3_KÜSIMUS1_VASTUS], ..., [MÕÕTEPUNKT25_KÜSIMUS1_VASTUS], [MÕÕTEPUNKT25_KÜSIMUS2_VASTUS], [MÕÕTEPUNKT25_KÜSIMUS1_VASTUS]
Näide rea sisust	732,24434, True, False, False, True, False, True, True, ..., True, , True, True, , True, True,

Teine loodav fail on erinev kõigist teistest väljundfailidest, kuna ei ole kaadri põhine, vaid kirjeldab kogu salvestist. Selles failis kuvatakse mõõtepunktide mittekorrektse staatuse korreleerumine. Tegemist on maatriksiga, kus kuvatakse rea väärtuse mittekorrektse esinemissagedus tulba väärtuse mittekorrektse väärtuse korral. Faili esimesed kaks rida ja iga rea esimesed kaks veergu kirjeldavad vastava rea või veeru väärtusi, et faili tabeli kujule viies oleks seda lihtsam mõista. Fail kannab nime „*jointStatusCor*“.

4.3.3 *Bone length* ehk luupikkused

Siia faili kirjutatakse iga arvutatud luupikkuse erinevus antud luu senisest keskmisest, kus ühikuks on standardhälve. Luud on järjestatud vastavalt enumi *BoneType* dikteeritud järjekorrale (vt Lisa 3 – Enum *BoneType*). Failil on 9-realine päis, kus lisaks luude nimetustele kuvatakse informatsiooni selle kohta, milline oli arvutuste aluseks olnud luu standardhälve (vt Tabel 2) ning statistiline informatsioon konkreetse salvestise kohta – iga luu keskmine pikkus, standardhälve ja variatsioonikordaja.

Tabel 6 Faili „*boneLength*“ formaat

Faili nime vorm	boneLength_[KEHA_ID].csv
Näide faili nimest	boneLength_72057594037928801.csv
Päise ridade arv	9

Rea sisu vorm	[KAADRI_NUMBER],[SUHTELINE_AEG], [LUU1_ERINEVUS], [LUU2_ERINEVUS], [LUU3_ERINEVUS], ... [LUU22_ERINEVUS], [LUU23_ERINEVUS], [LUU24_ERINEVUS]
Näide rea sisust	857,28600, 0.13235556416836,0.4852288374437,0.606226086341457, 0.460664405551202,0.197168051042281, ... 0.892948205223941,0.0773673770654556,0.546613369301835

4.3.4 Joint angle ehk nurgad

Seda valikut tehes luuakse kaks faili. Mõlemas failis on nurgad järjestatud vastavalt enumi *AngleType* dikteeritud järjekorrale (vt Lisa 4 – Enum *AngleType*).

Esimene fail sisaldab iga mõõdetud nurga väärtust kraadides ning selle kiirendust rad/s².

Tabel 7 Faili „jointAngle“ formaat

Faili nime vorm	jointAngle_[KEHA_ID].csv
Näide faili nimest	jointAngle_72057594037928801.csv
Päise ridade arv	2
Rea sisu vorm	[KAADRI_NUMBER],[SUHTELINE_AEG], [NURK1_VÄÄRTUS],[NURK1_KIIRENDUS], [NURK2_VÄÄRTUS],[NURK2_KIIRENDUS], ... [NURK10_VÄÄRTUS],[NURK10_KIIRENDUS]
Näide rea sisust	575,19200, 173.847540434329,-0.389693285039556, 174.578357676521,1.88169248180002, ... 163.125586569623,0.720859638168901, 168.097509074365,2.87694871264119

Teine loodav fail on sisukam ja sisaldab lisaks nurkade arvväertustele ka iga nurga aluseks olnud mõõtepunkti koordinaate.

Tabel 8 Faili „*jointAngleWithPos*“ formaat

Faili nime vorm	jointAngleWithPos_[KEHA_ID].csv
Näide faili nimest	jointAngleWithPos_72057594037928801.csv
Päise ridade arv	1
Rea sisu vorm	[KAADRI_NUMBER],[SUHTELINE_AEG], [NURK1_PUNKT1_X],[NURK1_PUNKT1_Y], [NURK1_PUNKT1_Z],[NURK1_PUNKT2_X], [NURK1_PUNKT2_Y],[NURK1_PUNKT2_Z], [NURK1_PUNKT3_X],[NURK1_PUNKT3_Y], [NURK1_PUNKT3_Z],[NURK1_VÄÄRTUS], ... [NURK10_PUNKT1_X],[NURK10_PUNKT1_Y], [NURK10_PUNKT1_Z],[NURK10_PUNKT2_X], [NURK10_PUNKT2_Y],[NURK10_PUNKT2_Z], [NURK10_PUNKT3_X],[NURK10_PUNKT3_Y], [NURK10_PUNKT3_Z],[NURK10_VÄÄRTUS]
Näide rea sisust	555,18534, -0.160358,-0.4421717,3.433621, -0.1826317,-0.8170503,3.47894, -0.1684649,-1.158752,3.499409,173.291280908128, ... 0.1603538,-0.4402362,3.357194,0.1360054, -0.2170189,3.411049,0.08411073,0.05493017,3.423301, 168.148240974834

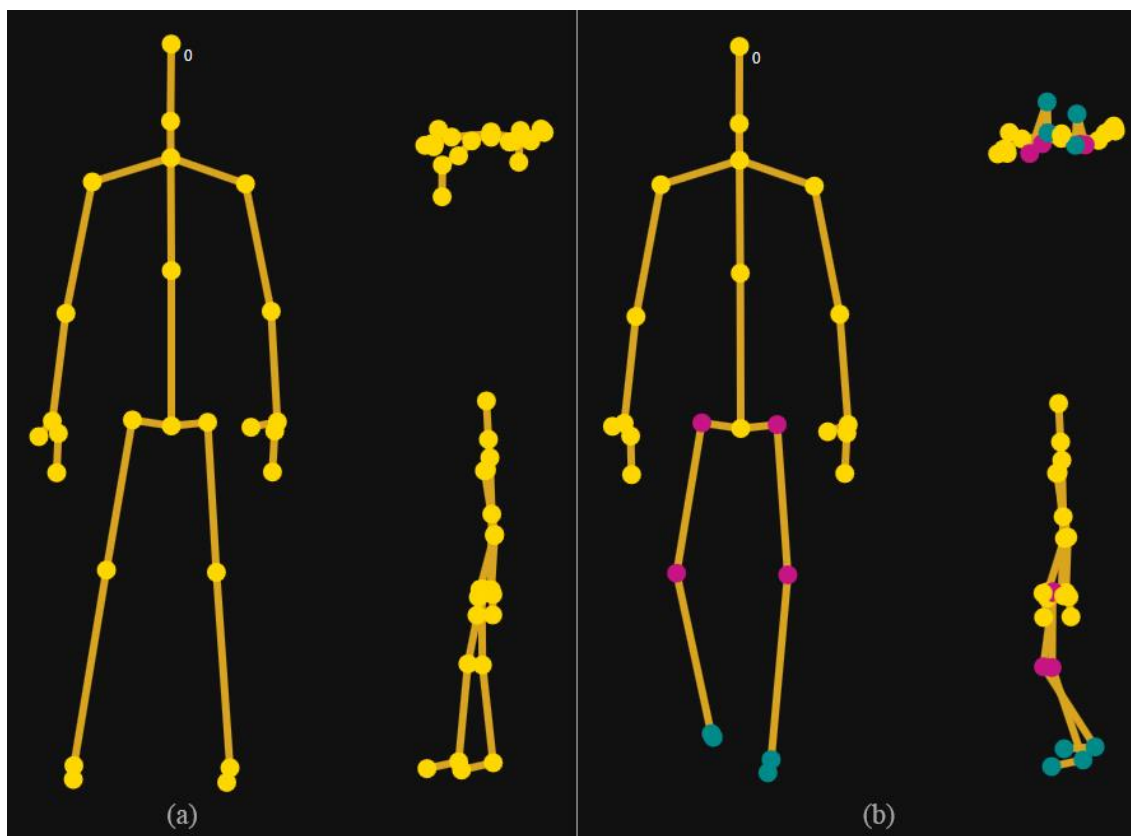
5 Rakenduse tulemuse valideerimine

Et veenduda rakenduse tulemuse korrektsuses, viidi läbi eksperimentaalsed mudeli registreerimised. Modelleerimisel kasutati isikut, keda pole antud töö raames veel modelleeritud. See välistas võimaluse, et rakendus on selgeks õppinud kindlate isikute karakteristika. Läbi viidud eksperimentideks on:

1. kõndimine edasi-tagasi
2. kohapeal seistes keha pööramine 90° kaamera suhtes

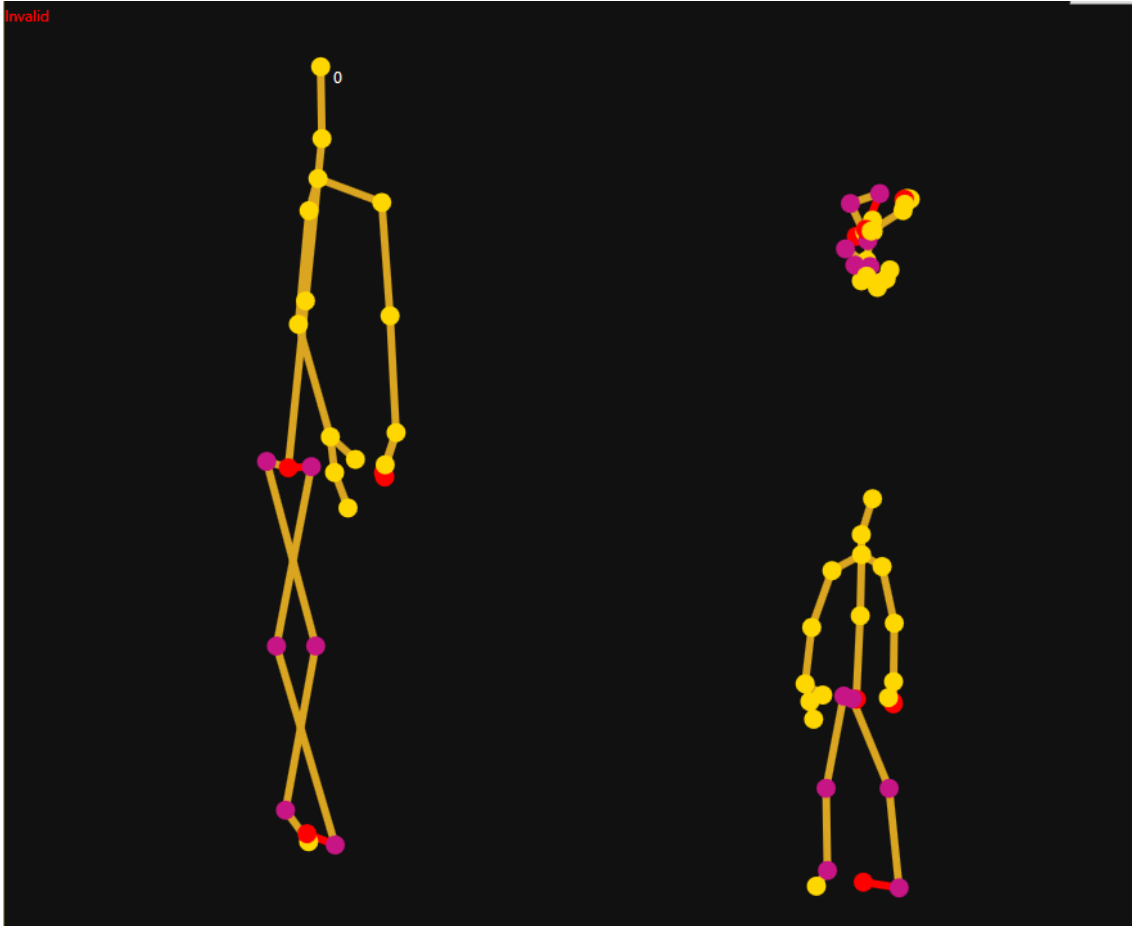
Salvestamise ajal jälgis autor reaalaja vaates ilmutavat tagasisidet valideerimise tulemuse kohta ning salvestamise järel võrdles autor taasesitusvaates automaatse valideerimise tulemust enda arvamusel.

Esimese eksperimendi käigus palus autor modelleeritaval kõndida kaamera ees edasi-tagasi, olles näoga kaamera poole. Sellise mudeli koostamisega saab Kinect hästi hakkama ning vigu esineb üpris vähe, nõnda paistis ka reaalaja vaates mudeli registreerimise ajal. Kogu salvestise pikkuseks on 876 kaadrit ning rakendus märkis mittekorrektseks neist vaid 35 kaadrit ehk vähem kui 4%. Kaadreid üksikult analüüsid leidis autor, et need tõepoolest olid ebaloomulikud. Vaata Joonist 9, kus on kujutatud järjestikused kaadrid ning valideerimiskomponendi reaktsioon jalgade asukoha äkilisele muutusele. Vasakpoolne kaader on märgitud korrektseks ning parempoolne kaader on märgitud mittekorrektseks. Värvide Tabel 3 järgi tõlgendades on näha, et mittekorrektseks on märgitud jalgade mõõtepunktid ning põlvedele konstrueeritud nurgad.



Joonis 9 Järjestikused kaadrid esimesest eksperimentist: (a) korrektseks märgitud kaader, (b) eelmisele järgnev ning mittekortseks märgitud kaader

Teises eksperimentis keeras modelleeritav end kaamera suhtes 90° korduvalt vasakule ja paremale. Selle eksperimenti eesmärgiks oli veenduda, et rakendus märgib kaadrid mittekortseks ka salvestise puhul, mis on valdavas osas kehv kvaliteediga. Küljega kaamera poole seistes ei näe kaamera varjutuse tõttu mitmeid mõõtepunkte ning see kajastub kehv registreeritud mudeli näol. Salvestises on 926 kaadrit, millest rakendus märkis ebakortseks tervelt 857 ehk umbes 92% (vt näidist Joonisel 10). Selline tulemus on oodatav, kuna mittekortseid kaadreid oli palju. Üksikute kortsete kaadrite kohta ei saa võtta vastu adekvaatset otsust, kuna luude pikkuse arvutused sõltuvad keskmistest väärtustest.



Joonis 10 Näide eksperiment 2 käigus registreeritud kaadrist ja selle valideerimise tulemusest

Eksperimentide käigus sai kinnitust, et rakenduse mõõtepunktide ja luude valideerimistulemused on väga tundlikud. Esimeses eksperimentis, mis logistilisel regressioonil põhineva ennustuse järgi oli valdavalt korrektne, on tegelikult väga palju väikeseid vigu. Kõige rohkem on neid seoses põidade ja käelaba tippude mõõtepunktidega. Ennustus kogu mudeli kohta on aga stabiilne, mida oligi tarvis saavutada.

6 Kokkuvõte

Käesolevas töö eesmärgiks oli luua tarkvara inimkeha skelettmudeli valideerimiseks, mis annaks skelettmudeli töötlejatele võimaluse kiirelt ja lihtsalt veenduda töötlemise aluseks oleva mudeli kvaliteedis.

Töös tehti ülevaade põhjustest, miks registreeritud skelettmudel võib olla mittekorrektne ning pakuti välja meetodid, kuidas algoritmiliselt hinnata mudeli kvaliteeti. Suurt rõhku pöörati ka korrektseid ja mittekorrektseid andmeid eristavate piirväärtuste leidmisele, et kasutajale kuvataks võimalikult adekvaatne pilt tõenäolistest probleemidest. Piirväärtuste leidmiseks viidi läbi mitmed skelettmudeli registreerimise katsed erinevate kehatüüpidega inimestega. Töös tehti ülevaade, kuidas teoreetilised meetodid rakenduses praktiliselt lahendati ning missugused vaated ja väljundfailid iseloomustavad loodud tarkvara.

Töö koostamisel õppis autor palju uut. Tegemist oli autori esimese Kinecti rakendusega ning samuti esimese tööga C# programmeerimiskeeles. Töö koostamise vältel sai autor pidevat tagasisidet rakenduse potentsiaalse kasutaja ja käeoleva töö juhendaja Sven Nõmmi käest. See suhe õpetas autorile, kuidas matemaatilist teooriat tarkvaras praktiliselt rakendada.

Töö tulemusena loodi tarkvara, millega on võimalik inimese skelettmudelit registreerida, taasesitada ning valideerida. Valideeritud mudel ja selle aluseks olnud andmed on võimalik salvestada faili. Tulemuste kontrollimiseks viidi läbi eksperimentaalsed katsed ning kinnitati, et tarkvara annab mudeli kvaliteedi kohta õiget informatsiooni. Loodud tarkvara lihtsustab ja kiirendab inimkeha skelettmudeli töötlejate töövoogu.

Kasutatud kirjandus

- [1] F. L. P. D. Mathias Fontmarty, „Likelihood tuning for particle filter in visual tracking,“ [Võrgumaterjal]. Available: <http://ieeexplore.ieee.org/document/5413473>. [Kasutatud 10 märts 2017].
- [2] A. F. A. B. A. K. M. F. B. M. T. S. Jamie Shotton, „Real-Time Human Pose Recognition in Parts from Single Depth Images,“ [Võrgumaterjal]. Available: <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/BodyPartRecognition.pdf>. [Kasutatud 27 veebruar 2017].
- [3] H. v. Eitzen, „Mathematics Stack Exchange,“ [Võrgumaterjal]. Available: <https://math.stackexchange.com/a/473647>. [Kasutatud 05 aprill 2017].
- [4] J. Medlej, „Human Anatomy Fundamentals: Flexibility and Joint Limitations,“ [Võrgumaterjal]. Available: <https://design.tutsplus.com/articles/human-anatomy-fundamentals-flexibility-and-joint-limitations--vector-25401>. [Kasutatud 03 märts 2017].
- [5] J. Brownlee, „Logistic Regression for Machine Learning,“ [Võrgumaterjal]. Available: <http://machinelearningmastery.com/logistic-regression-for-machine-learning/>. [Kasutatud 01 mai 2017].
- [6] G. Rodríguez, „Wald test,“ [Võrgumaterjal]. Available: <http://data.princeton.edu/wws509/notes/c2s3.html>. [Kasutatud 03 mai 2017].
- [7] Microsoft, „Kinect for Windows v2 Windows Runtime API Reference,“ [Võrgumaterjal]. Available: <https://msdn.microsoft.com/en-us/library/dn758675.aspx>. [Kasutatud 28 veebruar 2017].
- [8] J. A. Berry, „KinectMathHelpers,“ [Võrgumaterjal]. Available: <https://github.com/IntStarFoo/KinectJointVisualizer/blob/master/FiveTwoFiveTwo/KinectMathHelpers.cs>. [Kasutatud 20 aprill 2017].
- [9] A. Framework, „Accord.NET Framework LogisticRegressionAnalysis,“ [Võrgumaterjal]. Available: http://accord-framework.net/docs/html/T_Accord_Statistics_Analysis_LogisticRegressionAnalysis.htm. [Kasutatud 10 mai 2017].
- [10] A. Framework, „Accord.NET Framework Serializer,“ [Võrgumaterjal]. Available: http://accord-framework.net/docs/html/T_Accord_IO_Serializer.htm. [Kasutatud 10 mai 2017].
- [11] Microsoft, „Kinect hardware requirements and sensor setup,“ [Võrgumaterjal]. Available: <https://developer.microsoft.com/en-us/windows/kinect/hardware-setup>. [Kasutatud 28 veebruar 2017].

Lisa 1 – Mõõtepunktide eesti keelsed vasted

Tabel 9 Mõõtepunktid inglise ja eesti keeles

Inglise keeles	Eesti keeles
SpineBase	Ristluu
SpineMid	Selgroo keskosa
Neck	Kael
Head	Pea
ShoulderLeft	Vasak õlg
ElbowLeft	Vasak küünarliiges
WristLeft	Vasak ranne
HandLeft	Vasak käelaba
ShoulderRight	Parem õlg
ElbowRight	Parem küünarliiges
WristRight	Parem ranne
HandRight	Parem käelaba
HipLeft	Vasak puus
KneeLeft	Vasak põlv
AnkleLeft	Vasak hüppeliiges
FootLeft	Vasak pöid
HipRight	Parem puus
KneeRight	Parem põlv
AnkleRight	Parem hüppeliiges
FootRight	Parem pöid
SpineShoulder	Rinnalülide tipp
HandTipLeft	Vasaku käelaba otspunkt
ThumbLeft	Vasak põial
HandTipRight	Parema käelaba otspunkt
ThumbRight	Parem põial

Lisa 2 – Enum *JointType*

```
enum JointType
{
    SpineBase,
    SpineMid,
    Neck,
    Head,
    ShoulderLeft,
    ElbowLeft,
    WristLeft,
    HandLeft,
    ShoulderRight,
    ElbowRight,
    WristRight,
    HandRight,
    HipLeft,
    KneeLeft,
    AnkleLeft,
    FootLeft,
    HipRight,
    KneeRight,
    AnkleRight,
    FootRight,
    SpineShoulder,
    HandTipLeft,
    ThumbLeft,
    HandTipRight,
    ThumbRight
}
```

Lisa 3 – Enum *BoneType*

```
enum BoneType
{
    HeadToNeck,
    NeckToSpineShoulder,
    SpineShoulderToSpineMid,
    SpineMidToSpineBase,
    SpineBaseToHipLeft,
    HipLeftToKneeLeft,
    KneeLeftToAnkleLeft,
    AnkleLeftToFootLeft,
    SpineBaseToHipRight,
    HipRightToKneeRight,
    KneeRightToAnkleRight,
    AnkleRightToFootRight,
    SpineShoulderToShoulderLeft,
    ShoulderLeftToElbowLeft,
    ElbowLeftToWristLeft,
    WristLeftToHandLeft,
    HandLeftToHandTipLeft,
    WristLeftToThumbLeft,
    SpineShoulderToShoulderRight,
    ShoulderRightToElbowRight,
    ElbowRightToWristRight,
    WristRightToHandRight,
    HandRightToHandTipRight,
    WristRightToThumbRight
}
```

Lisa 4 – Enum *AngleType*

```
enum AngleType
{
    HipLeftKneeLeftAnkleLeft,
    AnkleRightKneeRightHipRight,
    KneeLeftSpineBaseHead,
    HeadSpineBaseKneeRight,
    KneeRightSpineBaseKneeLeft,
    HeadSpineShoulderSpineMid,
    HeadSpineShoulderSpineBase,
    SpineShoulderSpineMidSpineBase,
    WristLeftElbowLeftShoulderLeft,
    WristRightElbowRightShoulderRight,
}
```