

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Tiina Sumeri 142379IAPB

KÕNNAKU ANALÜÜS ISIKU TUVASTAMISEKS

Bakalaureusetöö

Juhendaja: Sven Nõmm
Ph.D

Tallinn 2018

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: Tiina Sumeri

21.05.2018

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on luua meetod, mis tuvastab inimest tema kõnnaku järgi. Eesmärgi saavutamiseks kirjutati rakendus, mis registreerib Microsoft Kinect sensori abil inimese kõnnaku sammud, salvestab nende parameetrid ning seejärel tuvastab inimest kahel erineval meetodil. Esimene meetod on Bezier' kõverate koostamine kõnnaku parameetrite järgi ning nende võrdlemine dünaamilise ajadeformatsiooni abil, samuti kahe liikumise massi parameetri võrdlemine omavahel. Teine meetod on kõnnaku liikumise massi parameetrite arvutamine, Fisher'i skoori abil klassifitseerimiseks parimate parameetrite valimine ning nende klassifitseerimine otsustuspuu abil. Meetodite töökindluse testimisel osales kokku 13 inimest.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 23 leheküljel, 6 peatükki, 11 joonist, 3 tabelit.

Abstract

Gait analysis for human identification

The main goal of the current bachelor's thesis is to create a method, which recognizes the human by the way he/she walks. To accomplish this goal, an application was created, which registers human's gait footsteps with the help of Microsoft Kinect sensor, then saves the parameters of his/her gait and afterwards tries to identify who was walking in front of the sensor.

The first method uses Bezier' curves and dynamical time warping algorithm, supported by two motion mass parameters, which are saved for the second method in the current study. Then it compares the curves with dynamical time warping algorithm and also finds the most similar motion mass parameters. The second method uses the motion mass parameters normalized by the height of the human, finds best parameters with Fisher's score algorithm and classifies them with the help of decision trees. Cross-validation algorithm is used to get the accuracy of this method.

13 people were tested with these methods, 5 female and 8 male. In the end of this thesis the two methods are compared with one another and with other methods from earlier papers.

The thesis is in Estonian and contains 23 pages of text, 6 chapters, 11 figures, 3 tables.

Lühendite ja mõistete sõnastik

DTW	Dünaamiline ajadeformatsioon (<i>Dynamical Time Warping</i>)
HMM	Markovi peitmudel (<i>Hidden Markov model</i>)
LDA	Lineaarse diskriminandi analüüs
LDC	Lineaarse diskriminandi klassifikaator
NN	Lähima naabri algoritm
RGB	Punane, roheline ja sinine (<i>Red, green and blue</i>)
SDK	Tarkvaraarenduskomplekt (<i>Software Development Kit</i>)

Sisukord

1 Sissejuhatus	9
2 Kirjanduse ülevaade	10
2.1 Eelnevalt tehtud tööd	10
2.2 Tallinna Tehnikaülikoolis tehtud tööd.....	12
2.3 Microsoft Kinect.....	13
2.3.1 Riistvara.....	13
2.3.2 Tarkvara.....	15
2.4 Bezier' kõver	15
2.5 Dünaamiline ajadeformatsioon.....	16
2.6 Liikumise mass	17
2.7 Fisher'i skoor.....	18
2.8 Accord raamistik.....	18
2.9 Otsustuspuu	18
2.10 Ristvalideerimine.....	19
2.11 Juhuslik mets	19
3 Rakendus	20
3.1 Meetod 1 – Bezier' kõver ja dünaamiline ajadeformatsioon.....	22
3.1.1 Töö käik.....	22
3.2 Meetod 2 – Liikumise mass ja otsustuspuu	24
3.2.1 Töö käik.....	24
4 Tulemused	26
4.1 Meetod 1	26
4.2 Meetod 2.....	27
5 Analüüs.....	29
5.1 Võrdlused varasemate töödega.....	29
6 Kokkuvõte	31
Kasutatud kirjandus	32
Lisa 1	34

Jooniste loetelu

Joonis 1. Tallinna Tehnikaülikoolis eelnevalt tehtud tööd.	12
Joonis 2. Kinect sensori struktuur [21].	14
Joonis 3. 1 - parem käsi, 2 – parem põial, 3 – parema käe tipp, 4 - vasak põial, 5 – vasak käsi, 6 – vasak käe tipp, 7 – selgroo ülemine lüli, 8 – selgroo alumine lüli.	14
Joonis 4. Koordinaatsüsteem [23].	15
Joonis 5. Nelja punktiga Bezier' kõver.	16
Joonis 6. Rakenduse põhivaade. Ülemisel paremal pildil on inimese külgvaade, üleval vasakul eestvaade ning all vasakul pealtvaade.	20
Joonis 7. Graafikute vaade. Ülemine graafik näitab jalgade kõrgust maapinna suhtes, alumine graafik jalgade omavahelist kaugust kogu kõnnaku vältel. Punased täpid alumisel graafikul näitavad uue sammu alguspunkti.	21
Joonis 8. Bezier' kõverate vaade.	22
Joonis 9. Sammu faasid Bezier' kõveratena: faas 1 - vasak jalg maksimaalselt ees, faas 2 - parem jalg maksimaalsel kõrgusel, faas 3 - parem jalg maksimaalselt ees, faas 4 - vasak jalg maksimaalsel kõrgusel. Igal joonisel on kõver nii vasaku kui ka parema jala jaoks.	23
Joonis 10. Sinine kõver - ühe jala Bezier' kõver nelja punktiga, punased jooned – jala tegelik asend, b_0 tähistab puusa punkti, b_3 jalalaba punkti.	23
Joonis 11. Näide otsustuspuu reeglistikust.	25

Tabelite loetelu

Tabel 1. Tulemused Bezier' kõvera ja DTW meetodit kasutades.	26
Tabel 2. Tulemused liikumise massi ja otsustuspuud või juhuslikku metsa kasutades..	28
Tabel 3. Võrdlused teiste töödega.	30

1 Sissejuhatus

Inimese kõnnaku tuvastamine on tänapäeval mitmetes valdkondades aktuaalne teema. Meditsiinis kasutatakse kõnnaku tuvastamist ning selle parameetrite jälgimist haiguste diagnoosimiseks, kriminoloogias kurjategijate ja terroristide leidmiseks ning jälgimiseks.

Antud töö eesmärgiks on luua meetod isiku tuvastamiseks kõnnaku järgi. Eesmärgi saavutamiseks uuritakse varasemate meetodite sobivust kaasaegse riistvara jaoks, valitakse sobivaimad ning kohandatakse, kasutades kaasaegset inimbiomehhaanika modelleerimist. Saadud tulemused kirjutatakse tarkvara vormis, programm registreerib Kinect sensori abil kõnnaku 2-4 sammu, seejärel arvutab kõnnaku tunnused, võrdleb olemasolevate isikute andmebaasiga ning tagastab tulemuse, milleks on vastava inimese nimi, kes kaamera ees kõndis. Antud töös kasutatakse erinevaid masinõppe algoritme, mille tulemusi omavahel võrreldakse, näiteks kasutatakse dünaamilist ajadeformatsiooni (DTW). Töö tulemusena saab ka teada, millise antud töös kasutatud algoritmi abil saab kõige efektiivsemalt inimest kõnnaku järgi tuvastada. Tulemuste valideerimine toimub sarnasel viisil masinõppe algoritmide valideerimisega.

2 Kirjanduse ülevaade

2.1 Eelnevalt tehtud tööd

Kõnnakut on võimalik tuvastada erinevate kaamerate abil, näiteks soojuskaamera, stereokaamera, Kinect sensori või tavalise videokaamera abil. Kõnnaku tuvastamiseks kasutatavad meetodid jagatakse mudeli-põhisteks ja liikumise-põhisteks.

Meetoditest kasutatakse näiteks Fourier' teisendusi [1], erinevate vektorite abi, näiteks massivektori [2], Markovi peitmudelit (HMM) [3], [4], deterministlikku või Manifoldi [3] õpetust.

Erinevate kaamerate abil on võimalik leida kas inimese siluett [1], [2], [3], Kinect sensoriga saadud liigeste (*joint*) asukohad [5] või sügavuspildi järgi erinevad inimese omadused nagu pikkus või kõnnaku eripärad [4], [6].

[1] artiklis analüüsitakse ajas ja ruumis inimese karakteristikuid liikuval siluetil sageduse järgi. Leitakse, et *key* Fourier' deskriptorid tunnevad paremini inimese ära, kui teised Fourier' deskriptorid. *Key* Fourier' deskriptoriteks nimetatakse deskriptoreid, mille amplituudid jõuavad haripunkti.

[2] tuvastatakse inimest massivektori abil. Massivektor koosneb pikslite arvust, mille väärtus antud binaarkujul silueti real pole null. Esmalt saadakse pilt binaarkujul, seejärel normaliseeritakse silueti suurus, et oleks võrdsed pikkused pildil. Sarnaste massivektorite leidmiseks kasutatakse dünaamilist ajadeformatsiooni.

[3] esitleb meetodit, mis identifitseerib inimest kasutades selleks ainult siluetina kujutatud inimese sammude jada. Kõnnaku siluetid transformeeritakse madala-dimensiooniga manusteks ning genereeritakse HMM abil ajaseeriade pildid.

[6] kasutatakse Kinect sensorit tuvastamiseks inimest. Mõõdetakse erinevaid omadusi, nurki jäsemete osade vahel, kahe jäseme omavahelist kaugust, jäsemete kaugust maapinnast, neist andmetest luuakse andmebaas. Klassifikatsiooniks kasutatakse lähima naabri (NN) ja lineaarse diskriminandi klassifikaatorit (LDC).

[4] artiklis kasutatakse Kinect sensorit, millega leitakse sügavuspilt. Kõnnaku tsükkel leitakse Markovi peitmudeli (HMM) abil. Vaadatakse nii inimese pikkust kui ka

kõnnaku tsüklit ajal, mil ta on täielikult nähtav. Igale inimesele koostatakse temale vastav HMM, mille järgi oleks võimalik teda tuvastada. Meetod võimaldab identifitseerida ka inimesi, kes ei vasta ühelegi koostatud HMM-ile.

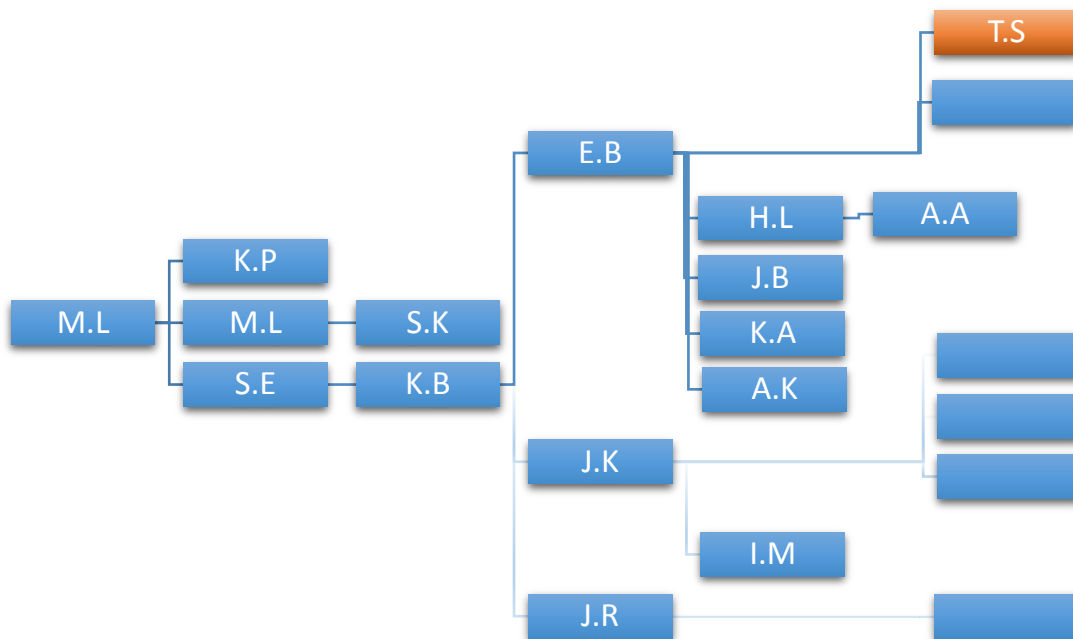
[7] vaadeldakse nelja parameetrit, mis iseloomustab inimese käe liikumise hulka ja sujuvust palli viskamisel. Käsitletakse inimese mootorikas õppimise käigus toimuvate muutuste mõõtmist ja kirjeldamist. Liigutuste massi kasutatakse mõõdetava näitajana, see kirjeldab kindla liigutuse tsükli käigus tehtud liigutuste hulka eukleidilises ruumis.

Artiklis [8] kasutatakse liikumise massi parameetreid, mis näitavad liikumise hulka ja sujuvust, leidmaks erinevusi Parkinsoni tõvega ning tervete isikute vahel. Artikli eesmärk on näidata *Up-and-Go* testi abil, et Parkinsoni tõvega isikute ja tervete isikute liikumine on üksteisest erinev.

[5] pakub välja meetodi, mis põhineb sensori andmetel, mis on saadud optilise süsteemi abil, töö eesmärk on inimese tuvastamine. Algoritmi on valitud liigutuste hulga alusel inimese puusa, põlve ja pahkluu liigesed. Andmetest tuuakse välja 3D staatilised ja dünaamilised liigete parameetrid. Meetod rakendab parameetrilise Bezier' kõvera tehnikat nende omaduste peal, et saada lülide liikumise vahel korrelatsioon.

[9] võrdleb omavahel suguluses olevate inimeste, nii kaksikute kui ka mitte kaksikute kõnnakuid. Parameetritena kasutatakse alajäsemete omavahelisi nurki. Töö pakub välja meetodi, mis normaliseerib kõnnaku tsüklit Bezier' polünoomi juure leidmisega ning arvestab kõnnaku kiiruse erinevustega. Klassifitseerimiseks kasutatakse lineaarse diskriminandi analüüsi (LDA) ja K-lähima naabri algoritmi. Töö tulemusena selgus, et kaksikute kõnnakutel on sarnasusi.

2.2 Tallinna Tehnikaülikoolis tehtud tööd



Joonis 1. Tallinna Tehnikaülikoolis eelnevalt tehtud tööd.

Joonis 1 esitab TTÜ-s eelnevalt tehtud töid, millest antud lõputöö teema välja kujunes.

Esimene riskülik initialsialidega M.L tähistab Mihhail Lapuškini tööd “*Application for Control of the „Pioneer” Robot with Manipulator*”, selle eesmärk oli luua süsteem, mille abil saaks kaardikeppi kasutades saata robotile mitteverbaalseid käsked [10]. Kenno Parm jätkas tööga “Hääletusautomaat liigutuste õppimiseks ja tuvastamiseks kirurgi robot-assistendis” [11] robotite juhtimise uurimisega.

M. Lapuškini magistritöö “Žestide peal baseeruv rakendus "Pioneer" roboti juhtimiseks” [12] oli jätk tema enda bakalaureusetööle, magistritöö eesmärk oli luua süsteem, mille abil juhtida robotit žestide abil. Sellest tööst arenes välja Siim Kirme bakalaureusetöö “Käežesti tuvastus Bezier’ kõverat ja K-lähima naabri meetodit kasutades” [13], mis jätkas liigutuste tuvastamise uurimist.

S.E tähistab Samet Erapi tööd “*Gesture Based PC Interface with Kinect Sensor*” [14], mille eesmärk oli luua žestidel põhinev kasutajaliides, mis võimaldaks inimesel suhelda arvutiga ilma füüsilise kontaktita, vaid Kinect sensori ees žeste tehes. Sellest tööst kasvas välja Kirill Buhhalko töö “*Monitoring of the human motor functions*” [15].

rehabilitation by neural networks based system with Kinect sensor”, milles loodi süsteem jälgimaks, kas patsient teeb taastusravi harjutusi õigesti.

Jevgeni Boruško “Alternatiivne meetod inimmotoorika modelleerimiseks” [15] töö eesmärk oli luua rakendus, mis võimaldab salvestada liikumise andmeid ja arvutada parameetrid, mis kirjeldavad liigutuste hulka ja sujuvust, samuti võimaldab see rakendus tuvastada mõningaid žeste. Tema tööst on arenenud välja mitmed nii magistri- kui ka bakalaureusetööd. Helena Lissenko töötas välja meetodi kuni neljalt Kinectilt saadud andmete töötlemiseks [16]. Tema tööst arenes välja Ahmed Abdelhady töö [17], mis kasutas kahte Kinecti liikumise salvestamiseks ning analüüsimiseks, jälgides inimese liikumist pöördemomendil.

Jan Joonas Bernstein arendas rakenduse, mis salvestab, registreerib, analüüsib ning säilitab inimliigutustel põhinevaid kontekstist sõltuvaid harjutusi [18]. Kris-Gerhard Aabrams lõi tarkvara, mille abil valideerida inimkeha skelettmudelit [19]. Anna Krajuškina arendas süsteemi, mis salvestab ja analüüsib Parkinsoni patsientide kõnnakut [20].

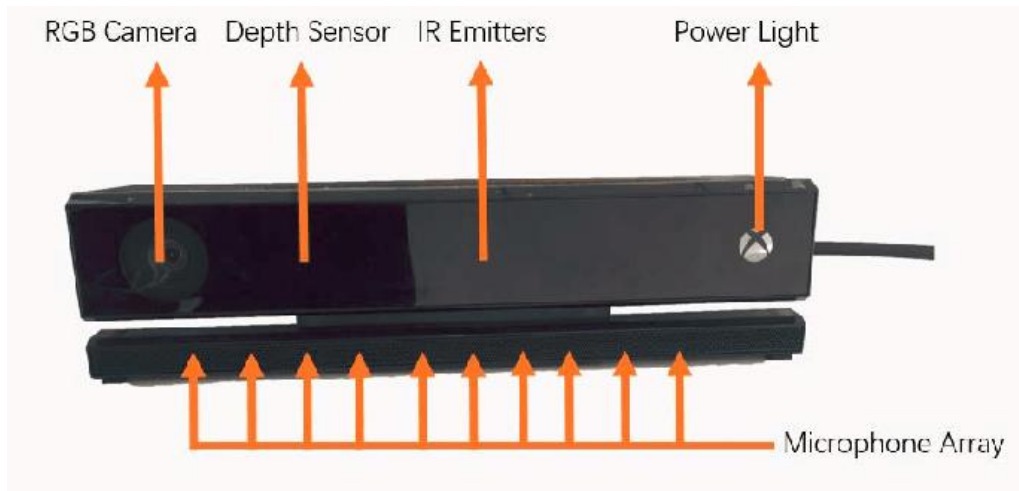
Samal ajal kirjutati paralleelselt ka töid, milles arendati rakendusi tahvelarvutile ning *Leap Motion* sensorile.

Tühjad riskülikud kujutavad töid, mis on hetkel tegemisel.

2.3 Microsoft Kinect

2.3.1 Riistvara

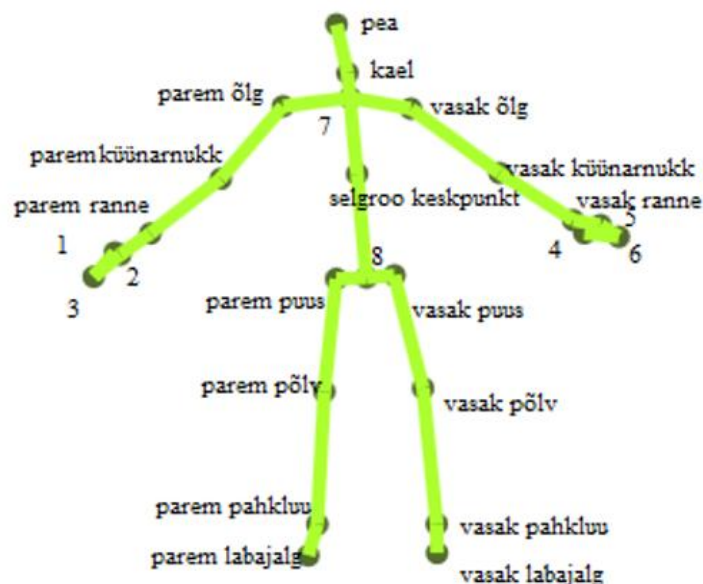
Kinect sensor on pikliku kujuga seade, mis seisab väikesel platvormil ja mida saab asetada lauale või statiivile. Seadmel on RGB kaamera, sügavussensor ja mitu mikrofoni (Joonis 2).



Joonis 2. Kinect sensori struktuur [21].

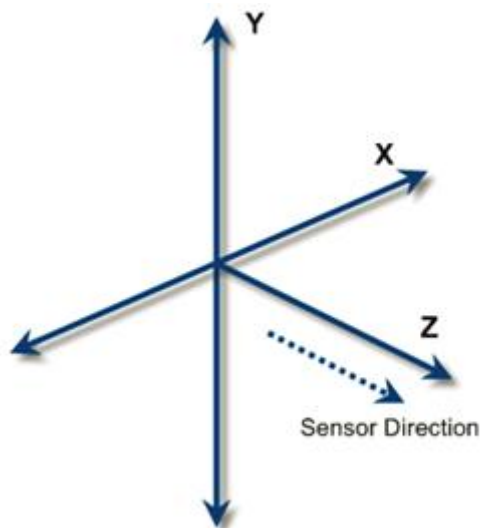
Nii video- kui ka sügavussensoril on 640x480-piksline resolutsioon ja need võimaldavad salvestada 30 kaadrit sekundis [22]. Kinecti sügavussensori ulatus on minimaalselt 800mm ja maksimaalselt 4000mm. Kinect kasutab infrapuna kaamerat, seega näeb ka läbi klaasi, kuid sensorit ei saa selle tõttu kasutada otsese päikese valguse käes.

Kinect sensori skeleti andmed koosnevad inimese kehal olevate punktide koordinaatidest (Joonis 3).



Joonis 3. 1 - parem käsi, 2 – parem põial, 3 – parema käe tipp, 4 - vasak põial, 5 – vasak käsi, 6 – vasak käe tipp, 7 – selgroo ülemine lüli, 8 – selgroo alumine lüli.

Skeleti andmete koordinaatsüsteem on 3D süsteem, mille väärtused on meetrites [23] (Joonis 4).



Joonis 4. Koordinaatsüsteem [23].

Selles töös kasutatakse teise generatsiooni Microsoft Kinect sensor v2.

2.3.2 Tarkvara

Kinect SDK on tarkvara, mis on loodud *Microsoft Visual Studio* jaoks. See võimaldab arendada rakendusi, mis kasutavad Kinect sensorilt kogutud RGB video, sügavuse, skeleti ja heli andmeid reaalajas [24].

Kinect SDK abil on võimalik ära tunda ning jälgida liikuvat inimest, leida kaugus objekti ning sensori vahel kasutades selleks sügavuskaameralt saadud andmeid, määrata kindla punkti värvi, salvestada heli või leida heli allikas [25].

2.4 Bezier' kõver

Igal polünoomiaalsel kõveral $b(u)$ kraadiga $\leq n$ on unikaalne n -kraadiga Bezier' kõver. Bezier' kõvera valem:

$$\mathbf{b}(u) = \sum_{i=0}^n c_i B_i^n(u).$$

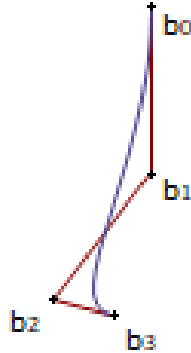
Afinne parameetrite teisendus

$$u = a(1 - t) + bt, \quad a \neq b,$$

ei muuda kurvi \mathbf{b} järku. Järelikult, $\mathbf{b}(u(t))$ -l on ka n -järguline Bezier' kuju,

$$b(u(t)) = \sum_{i=0}^n b_i B_i^n(t).$$

koefitsiente b_i kutsutakse Bezier' punktideks. Need on Bezier' polügooni $b(u)$ tipud intervalliga $[a,b]$. Parameetrit t kutsutakse lokaalseks ja parameetrit u globaalseks \mathbf{b} parameetriks Joonis 5[26] (Joonis 5).



Joonis 5. Nelja punktiga Bezier' kõver.

2.5 Dünaamiline ajadeformatsioon

Dünaamilist ajadeformatsiooni saab kasutada erinevat tüüpi andmete, näiteks video, audio või graafika võrdlemiseks. Paljud uurimused kasutavad dünaamilist ajadeformatsiooni kõne, allkirja või kõnnaku tuvastamiseks [24].

Dünaamilise ajadeformatsiooni eesmärk on võrrelda kahte (ajast sõltumatut) jada $X := (x_1, x_2, \dots, x_N)$ pikkusega $N \in \mathbb{N}$ ja $Y := (y_1, y_2, \dots, y_M)$ pikkusega $M \in \mathbb{N}$. Antud jaded võivad olla diskreetsed signaalid (aja seeriad) või, üldisemalt, tunnusjadad punktidest kindlal ajal. Järgmiselt määrame omadusruumi F . Seega $x_n, y_m \in F$, $n \in [1 : N]$ ja $m \in [1 : M]$. Selleks, et neid omadusi $x, y \in F$ võrrelda, vajatakse lokaalset kulufunktsiooni, vahel nimetatakse seda lokaalseks kauguse mõõduks, mis on defineeritud funktsiooniga

$$c : F \times F \rightarrow \mathbb{R}_{\geq 0}$$

Tüüpiliselt on $c(x,y)$ väike (madala kuluga), kui x ja y on sarnased üksteisele ning vastupidiselt on $c(x,y)$ suur (suure kuluga). Leides lokaalsed kulufunktsioonid igale elemendipaarile jadades X ja Y , saab kätte kulumatriksi $C \in \mathbb{R}^{N \times M}$, mis on defineeritud $C(n,m) := c(x_n, y_m)$. Seega on eesmärk leida X ja Y vaheline joondus minimaalse kuluga [27].

Dünaamilise ajadeformatsiooni eelis eukleidilise kauguse algoritmi ees on selle suutlikkus leida kahe erineva pikkusega jada omavaheline kaugus [28].

2.6 Liikumise mass

Mõiste liikumise mass iseloomustab, kui palju ning kui sujuvalt liigesed liiguvad. Liikumise massi parameetrid on trajektoori mass, kiirenduse mass, kombineeritud eukleidiline kaugus ning liikumise aeg. Liikumise massi elemendid iseloomustavad, kui palju ning kui sujuvalt kindel liigutus toimub.

Kombineeritud eukleidilise kauguse valem:

$$E_J = \sum_{i=1}^n E_{J_i},$$

kus E_J tähistab eukleidilist kaugust algus- ja lõpp-punkti vahel konkreetsel liigesel.

Trajektoori massi valem:

$$T_J = \sum_{i=1}^n T_{J_i}$$

Kiirenduse massi valem:

$$A_J = \sum_{i=1}^n A_{J_i}$$

Trajektoori mass iseloomustab liigutuste koguhulka ning kiirenduse mass liigutuse sujuvust. Neid koos liigutuse ajalise pikkusega kombineerides on võimalik võrrelda liigutusi erinevates õppimisprotsessi faasides. Trajektoori mass ja kiirenduse mass olenevad inimese kehaehitusest, seega inimestel, kellel on pikemad jäsemed, on suuremad trajektoori ja kiirenduse massi väärtused. Võimaldamaks võrrelda liikumise masse erinevate indiviidide vahel, on välja pakutud kaks järgmist parameetrit: kombineeritud eukleidilise kauguse ja trajektoori massi suhe

$$R_d = \frac{E_J}{T_J},$$

ning kombineeritud eukleidilise kauguse ja kiirenduse massi suhe

$$R_a = \frac{E_J}{A_J}.$$

Liikumise massi vektor koosneb neist kolmest parameetrist ning liikumiseks kulunud ajast [7]:

$$M_J = \{T_J, A_J, E_J, t\}$$

2.7 Fisher'i skoor

Fisher'i skoori abil mõõdetakse erinevust ühe klassi atribuudi väärtuse ning kõigi klasside keskmise atribuudi väärtuse vahel. Mida suurem Fisher'i skoor, seda suurem atribuudi eristusvõime. μ_j on keskmine, σ_j standardhälve j klassi kuuluvatel punktidel. p_j on murdosa andmepunktidest, mis kuuluvad klassi j . μ on globaalne keskmine kogu andmepunktidest

$$F = \frac{\sum_{j=1}^k p_j (\mu_j - \mu)^2}{\sum_{j=1}^k p_j \sigma_j^2}$$

Murru lugeja näitab klassidevahelist erinevust, nimetaja klassidesisest erinevust. Suurima Fisher'i skooriga atribuute kasutatakse klassifitseerimiseks [29].

2.8 Accord raamistik

Accord raamistik on .NET masinõppe raamistik kombineeritud audio- ja pilditööstustekidega, mis on kirjutatud programmeerimiskeeles C#. See raamistik on mõeldud tootmaks rakendusi arvutinägemise, arvutikuulmise, signaali töötlemise ning statistilise analüüsi jaoks, ka kommertsiaalseks kasutuseks [30]. Käesoleva töö raames on kasutatud Accord raamistiku otsustuspuud ning ristvalideerimise meetodit.

2.9 Otsustuspuu

Otsustuspuu on õppimismeetod, mida kasutatakse klassifitseerimise ja regressiooni jaoks. Otsustuspuu eesmärk on koostada mudel, mis ennustab etteantud muutuja väärtuse õppides eelnevalt lihtsaid otsustusreegleid vastavalt andmete omadustele [31].

Üheks otsustuspuude populaarsuse põhjuseks ning eeliseks teiste klassifitseerimismeetodite ees on nende intuitiivne arusaadavus ning kasutuslihtsus [32].

Töös kasutatakse C45 õpetamise algoritmi, mis genereerib otsustuspuu ja puu reeglid etteantud andmete järgi. Esimene faas otsustuspuud kasutades on õpetamisfaas, kus valitakse teatud hulk andmeid, mis antakse algoritmile ette ning mille järgi genereeritakse otsustuspuu koos reeglitega. Järgmine faas on valideerimisfaas, mille puhul antakse ette ülejäänud hulk andmeid ning kontrollitakse otsustuspuu õigsust [33].

2.10 Ristvalideerimine

Ristvalideerimise meetodit kasutatakse teiste klassifitseerimistehnikate täpsuse testimiseks. Väikest osa õpiandmetest (näiteks 10%) kasutatakse testandmetena – neid puu konstrueerimise faasis ei kasutata. Valmis puud testitakse seejärel varem kõrvale pandud testandmete peal, et näha, kas saadud puu suudab klassifitseerida vaid õpiobjekte või on see piisavalt korrektne ka tundmatute objektide korral.

Tihti jagataksegi õpihulk juhuslikult kümneks ning konstrueeritakse kümme erinevat otsustuspuud, valides iga kord erineva test-kümnendiku. Edasi hinnatakse iga puu klassifitseerimisviga oma testandmetel [32].

2.11 Juhuslik mets

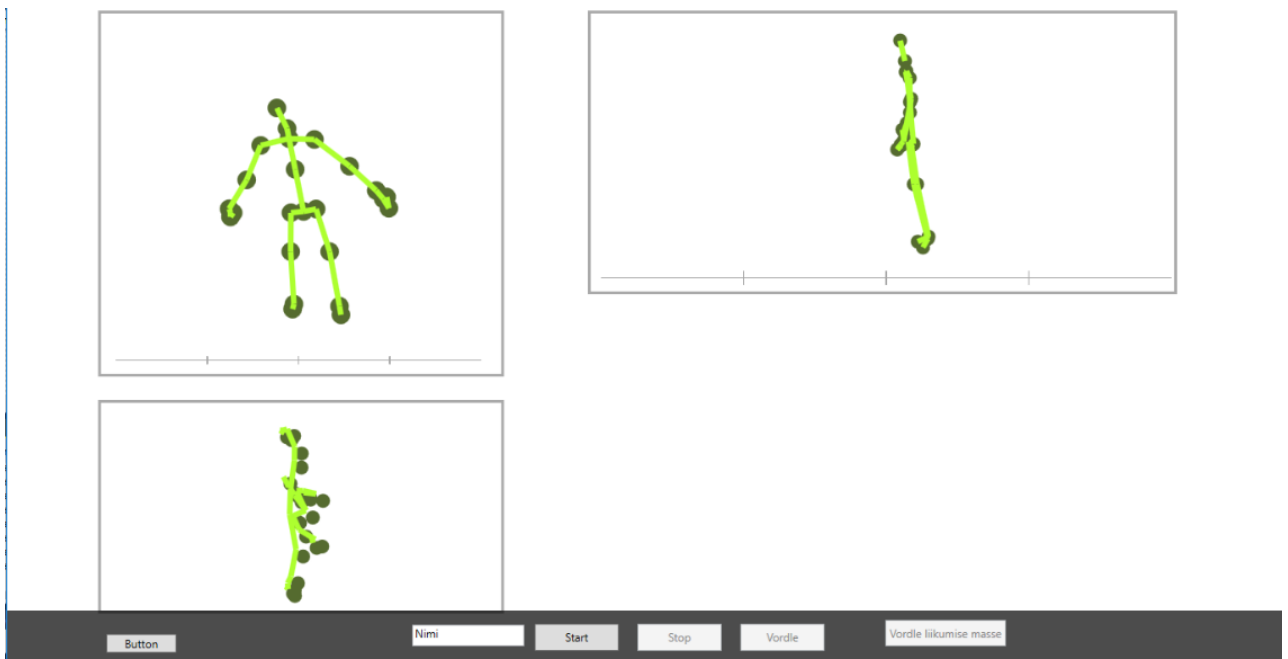
Juhuslik mets on mitmekülgne masinõppe meetod paljude rakendustega nii turunduses, tervishoius kui ka kindlustusvaldkonnas. Seda saab kasutada modelleerimaks turunduse mõju inimesele või ennustamaks haigestumisriski patsientidel. Juhuslik mets on võimekas regressiooni- ja klassifitseerimismeetod, mis suudab hallata palju erinevaid tunnuseid ning aitab anda hinnangut, millised väärtused on tähtsad modelleerimisel.

Juhuslik mets kuulub suuremasse masinõppe algoritmide klassi, mida nimetatakse komiteeõppe meetoditeks.

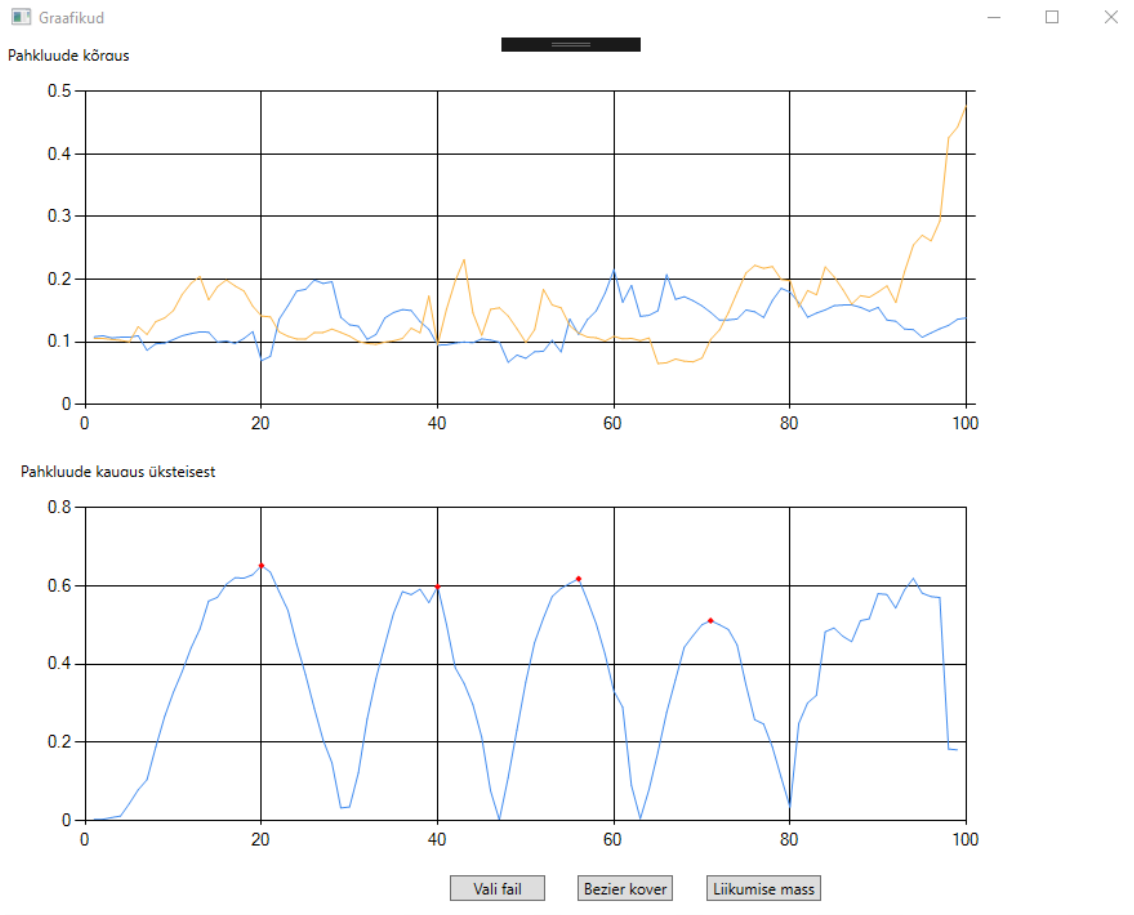
Juhusliku metsa algoritm genereerib hulga juhuslikke otsustuspuid. Puud genereeritakse juhuslikult, seega paljud neist pole klassifitseerimiseks/regressiooniks kasulikud, 99.9% puudest teevad ebaolulisi otsuseid ning taandavad seega üksteist välja. Samas genereerib algoritm ka mõned väga head puud, mis teevad häid otsuseid ning tänu neile on see meetod hea klassifitseerija [34].

3 Rakendus

Inimese tuvastamiseks kasutatav rakendus on kirjutatud C# programmeerimiskeeles. Rakenduse kasutajaliides koosneb mitmest erinevast vaatest. Esimene ehk üldvaade näitab kaamera ees oleva inimese skeletti nii otse-, kül- kui ka pealtvaates (Joonis 6). Selles vaates on võimalik salvestada inimese kõnnaku parameetrid ning võrrelda varasemate vastavate andmetega, mille tulemusena kuvatakse ekraanil Kinect sensori ees kõndinud inimese nimi. Antud töös saadi tulemused kahe erineva meetodi abil. Rakenduse teine vaade näitab varasemalt salvestatud failide infot ja tulemusi. Täpsemalt on selles vaates näha pahkluu koordinaatide järgi koostatud graafid jalgade asetusest maapinna suhtes ning nende omavahelisest kaugusest (Joonis 7). Nende graafide järgi on võimalik tuvastada, kas kõnnaku salvestisel on puuduseid või vigu. Jooniselt 7 on ka näha, et kõnnaku salvestuse algust ja lõppu ei saa kõnnaku tuvastamisel arvestada, salvestuse alguses on inimene veel nii kaugel, et Kinect sensor ei suuda inimest tuvastada ning salvestuse lõpul on inimene sensorile nii lähedal, et sensor ei näe enam inimese jalgu. Samuti on rakendusel Bezier' kõverate vaade (Joonis 8).

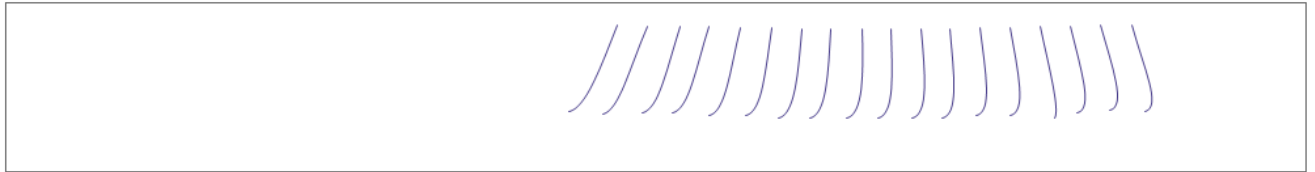


Joonis 6. Rakenduse põhivaade. Ülemisel paremal pildil on inimese külgsuurt, üleval vasakul eestvaade ning all vasakul pealtvaade.

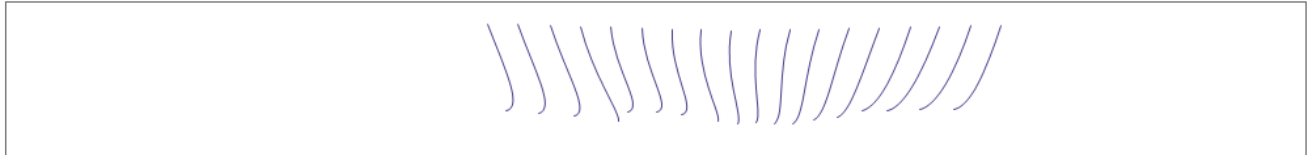


Joonis 7. Graafikute vaade. Ülemine graafik näitab jalgade kõrgust maapinna suhtes, alumine graafik jalgade omavahelist kaugust kogu kõnnaku vältel. Punased täpid alumisel graafikul näitavad uue sammu alguspunkti.

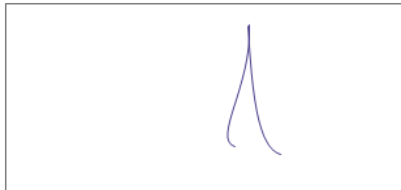
Vasaku jala Bezier kover



Parema jala Bezier kover



Vasak jalg korgemal Bezier kover



Parem jalg korgemal Bezier kover



Tuvastatud inimene: tarmo



Vasak jalg ees Bezier kover

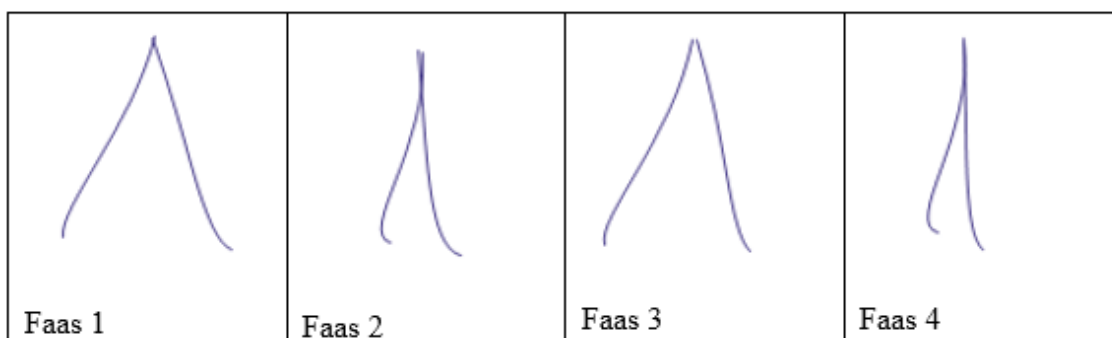
Parem jalg ees Bezier kover

Joonis 8. Bezier' kõverate vaade.

3.1 Meetod 1 – Bezier' kõver ja dünaamiline ajadeformatsioon

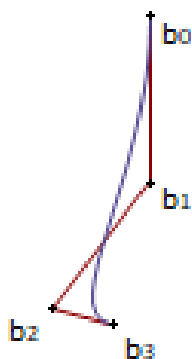
3.1.1 Töö käik

Inimese tuvastamisel salvestatakse kõnnaku 2-4 sammu parameetrid .csv laiendiga failina, nendeks on liigeste x-, y- ja z koordinaadid sammude vältel, mis salvestatakse ~30 korda sekundis. Inimene kõnnib otse Kinect sensori poole, salvestust alustatakse, kui ta on jõudnud sensori vaatevälja ning lõpetatakse, kui ta on umbes poole meetri kaugusel sensorist. Seejärel leitakse neli kaadrit, mis iseloomustavad inimese kõnnakut. Esimesel kaadril on vasak jalg ees ning maksimaalsel kaugusel paremast jalast, teisel parem jalg ees ning maksimaalsel kaugusel vasakust, kolmandal vasak jalg (pahklud arvestades) maksimaalsel kõrgusel põrandast ning neljandal parem jalg (pahklud arvestades) maksimaalsel kõrgusel põrandast. Need andmed salvestatakse eraldi ning vastavate kaadrite koordinaatide järgi koostatakse Bezier' kõverad eelneval neljal juhul mõlema jala jaoks (Joonis 9).



Joonis 9. Sammu faasid Bezier' kõveratena: faas 1 - vasak jalg maksimaalselt ees, faas 2 - parem jalg maksimaalsel kõrgusel, faas 3 - parem jalg maksimaalselt ees, faas 4 - vasak jalg maksimaalsel kõrgusel. Igal joonisel on kõver nii vasaku kui ka parema jala jaoks.

Bezier' kõvera joonistamisel kasutatakse iga kaadri nelja punkti y- ja z-koordinaate, antud juhul on vastavateks punktideks puus, põlv, pahkluu ning jalalaba, vastava nelja punkti abil arvutatakse ülejäänud kontrollpunktid, kokku on kõveral 101 punkti, millest esimene puusa punkti järgi võetud punkt ning viimane jalalaba punkt (Joonis 10). Bezier' kõvera kõikide punktide andmed salvestatakse omakorda eraldi faili.



Joonis 10. Sinine kõver - ühe jala Bezier' kõver nelja punktiga, punased jooned – jala tegelik asend, b_0 tähistab puusa punkti, b_3 jalalaba punkti.

Järgmisena võrreldakse testitava inimese Bezier' kõveraid, mida on kokku 8, teiste inimeste Bezier' kõveratega, kasutades selleks dünaamilise ajadeformatsiooni algoritmi. Algoritm võrdleb iga Bezier' kõvera koordinaatide jada kõikide salvestatud inimeste kõverate koordinaatide jadadega. Algoritm tagastab kahe koordinaatide jada omavahelise kauguse, seega mida väiksem on tulemus, seda sarnasemad on jadad omavahel. Lisaks Bezier' kõveratele võrreldakse liikumise massi parameetreid, nimelt eukleidilise kauguse ja trajektoori kauguse suhet, mis on normaliseeritud inimese pikkusega ning kiiruse massi, mis on samuti normaliseeritud inimese pikkusega. Kokku saadakse 10 väärtust, millest igale leitakse inimese nimi, kelle Bezier' kõver või

liikumise massi parameeter sarnaneb enim testitava isiku kõnnaku parameetritega. Tulemuseks on inimese nimi, keda esineb kümnest korrast kõige rohkem. Liikumise massi kaks parameetrit on lisatud meetodile, sest nende lisamisel andis meetod veidi paremaid tulemusi juhtudel, kui meetod tagastas kaks nime, millest üks oli õige. Nende parameetrite lisamise tulemusena tagastas meetod rohkematel kordadel vaid õige nime.

3.2 Meetod 2 – Liikumise mass ja otsustuspuu

3.2.1 Töö käik

Inimese tuvastamisel salvestatakse kõnnaku 2-4 sammu parameetrid, nendeks on liigete x-, y- ja z koordinaadid sammude vältel, mis salvestatakse 30 korda sekundis. Inimene kõnnib otse Kinect sensori poole, salvestust alustatakse, kui ta on jõudnud sensori vaatevälja ning lõpetatakse, kui ta on umbes poole meetri kaugusel sensorist. Seejärel arvutatakse iga sammu nelja liigese jaoks liikumise massi parameetrid, paremate tulemuste saamiseks normaliseeritakse neid inimese pikkusega. Inimese pikkus arvutatakse varem salvestatud liigete koordinaatide järgi, võttes aluseks pea, selgroo ülemise lüli, puusa, põlve, pahkluu ning jalalaba punktide omavaheliste kauguste summa. Korrekse pikkuse leidmiseks vaadeldakse esimest 40 kaadrit, mõlema jala pikkuseid eraldi ning iga inimese pikkuseks arvestatakse maksimaalset saadud pikkust. See tähendab, et kaadril, mille järgi pikkus leitakse, peaks inimene olema sirges asendis, jalg kõverdamata. Kokkuvõttes salvestatakse iga inimese kohta 48 parameetrit, mõlema jala puusa, põlve, pahkluu ja labajala kiiruse mass, kiirenduse mass, eukleidilise kauguse mass, trajektoori kauguse mass, eukleidilise kauguse massi ja trajektoori kauguse massi suhe ning eukleidilise kauguse massi ja kiirenduse massi suhe. Edasi arvutatakse iga sammu parameetrite jaoks Fisher'i skoorid, mille järgi leitakse parimad parameetrid, antud töö puhul prooviti läbi erinevad variatsioonid parameetritest, mille Fisher'i skoor oli üle 0,7. Nende parameetrite järgi toimub klassifitseerimine otsustuspuu abil, otsustuspuu genereerib reeglid, mille alusel klassifitseeritakse (Joonis 11). Esmalt toimub õppimisfaas, milles antakse otsustuspuule ette 70% kogu inimeste andmetest, seejärel valideerimisfaas, milles kontrollitakse, kas otsustuspuu oskab leida ülejäänud 30% inimeste andmetest õiged vastused, ehk kui anda ette üks fail, mis kuulub kindlale inimesele, siis peaks otsustuspuu vastuse olema selle inimese nimeline klass. Fisher'i skoori arvestades said valituteks parameetriteks parema

puusa kiiruse mass, mis on normaliseeritud inimese pikkusega, vasaku põlve eukleidilise kauguse massi ja trajektoori kauguse massi suhe, mis on normaliseeritud pikkusega, vasaku pahkluu kiiruse mass, mis on normaliseeritud inimese pikkusega ning vasaku puusa eukleidilise kauguse massi ja trajektoori kauguse massi suhe, mis on normaliseeritud inimese pikkusega. Töös prooviti ka juhusliku metsa algoritmi abil tuvastada inimest.

```
0 =: (2 <= 0.615112346997996) && (0 <= 12.3490081031513) && (1 > 11.9584364215624)
1 =: (2 <= 0.615112346997996) && (0 > 12.3490081031513) && (1 <= 16.4653869117071) && (3 > 24.0689381659333)
2 =: (2 > 0.615112346997996) && (0 <= 15.110418486799) && (1 <= 8.45672909600362)
3 =: (2 <= 0.615112346997996) && (0 > 12.3490081031513) && (1 <= 16.4653869117071) && (3 <= 24.0689381659333)
3 =: (2 <= 0.615112346997996) && (0 > 12.3490081031513) && (1 > 16.4653869117071) && (3 <= 25.2324548063145)
4 =: (2 > 0.615112346997996) && (0 <= 15.110418486799) && (1 > 8.45672909600362) && (3 <= 30.320615469339)
5 =: (2 > 0.615112346997996) && (0 > 15.110418486799) && (3 <= 36.5089933612542) && (1 <= 18.0466584837948)
6 =: (2 <= 0.615112346997996) && (0 > 12.3490081031513) && (1 > 16.4653869117071) && (3 > 25.2324548063145)
7 =: (2 <= 0.615112346997996) && (0 <= 12.3490081031513) && (1 <= 11.9584364215624)
8 =: (2 > 0.615112346997996) && (0 > 15.110418486799) && (3 <= 36.5089933612542) && (1 > 18.0466584837948)
9 =: (2 > 0.615112346997996) && (0 > 15.110418486799) && (3 > 36.5089933612542)
9 =: (2 > 0.615112346997996) && (0 <= 15.110418486799) && (1 > 8.45672909600362) && (3 > 30.320615469339)
```

Joonis 11. Näide otsustuspuu reeglistikust.

4 Tulemused

4.1 Meetod 1

Kõnnaku tuvastamise meetodi testimisel osales 13 inimest: 5 naist ning 8 meest, kes kõndisid Kinect sensori ees minimaalselt kolm korda ning kelle kõnnaku andmed salvestati. Kolmandat korda kõndides prooviti ka rakenduse töökindlust ning iga kõndija sai näha, kas rakendus tunneb ta eelneva kahe kõnnaku järgi viimasel korral ära.

Tulemuse saamiseks valiti iga inimese kohta juhuslikult 3 salvestatud faili. Edasi valiti juhuslikult iga inimese kohta üks fail, mis jäeti testimiseks. Seega oli testfailide osakaal 33%. Seejärel anti rakendusele ette testfailid ning rakendus tagastas inimese nime, kelle kõnnaku parameetrid sarnanesid enim testfaili parameetritele. Esimest kahte sammu korrati 10 korda, et tekiks sarnane algoritm ristvalideerimise algoritmile. Tabel 1 toob välja meetodi tulemused, „+“ tähistab inimese õigesti tuvastamist programmi poolt, „-“ valesti tuvastamist. Juhtudel, mil rakendus tagastas tulemuseks kaks nime, mida pidas võrdsel määral vastuseks, arvestati tulemus täpsuse arvutamisel õigeks, kui üks neist nimedest oli õige. Tabelis 1 on märgitud sellised juhud tärniga.

13 inimese failidest tuvastas meetod kokku 112 katsel 130-st inimese ära, seega inimese tuvastuse tõenäosus selle meetodi tulemusena on 86%.

Tabel 1. Tulemused Bezier' kõvera ja DTW meetodit kasutades.

Katse nr. Isiku nr.	1	2	3	4	5	6	7	8	9	10
1	+	+	+	+	+	+	+	+	+	+
2	+	+	+	+	+	+	+	+	+	+
3	+	+	+	+	+	+	+	+	+	+
4	+	+	+	+	+	+	+	+	+	+
5	-	-	-	+*	+	-	-	-	+	-

6	-	+	+	+	+	+	+	+	+	+
7	+	+	+	+	+	+	+	+	+	+
8	+	+	+	+	+	-	+	+	+	+
9	+	+	+	+	+	+	+	+	+	+
10	-	-	+	+	+	+	-	+	+	+
11	+	+	+	+	+	+	+	-	+	+
12	+	+	+	+	+	+	+	+	+	+
13	+	+	-	-	+	+	+	-	-	-

4.2 Meetod 2

Teise meetodi testimisel osales 10 inimest: 5 meest ning 5 naist, iga inimese kohta salvestati vähemalt kolme kõnnaku parameetrid. Tulemus saadi kahel meetodil, tavalise otsustuspuu meetodiga ning juhusliku metsa meetodiga. Tulemust valideeriti, proovides läbi erinevate kõnnakufailide kombinatsioonid. Iga inimese kohta oli valitud 3 faili, need olid jagatud õppimiseks ning testimiseks, vastavalt 67% ning 33%. Tabel 2 näitab tulemusi iga inimese kõnnaku kohta, näiteks h1, h2, h3 tähistab ühe inimese kolme erineva kõnnaku salvestist, h1 kõnnaku puhul tuvastas juhusliku metsa algoritm inimese õigesti, kuid otsustuspuu mitte. Kõnnaku h2 puhul ei tuvastanud kumbki meetod inimest, samas kõnnaku h3 puhul tuvastasid mõlemad meetodid inimese õigesti. Tabelis tähistab „+“ inimese õigesti tuvastamist programmi poolt ning „-“ valesti tuvastamist.

Meetodi täpsus saadi kasutades ristvalideerimise algoritmi, mis oli kättesaadav Accord raamistikus nagu ka otsustuspuu ise. Ristvalideerimise puhul jagas algoritm puu kümneks osaks, mida kordamööda testiti. Rakendus tagastas peale igit testimist ristvalideerimise täpsuse.

Testides läbi kõik failid, tuvastas otsustuspuu 30-st korrast 12 korda inimese ära, juhusliku metsa algoritm vaid 7 korda 30-st. Otsustuspuu tuvastuse tõenäosus on seega 40% ning juhusliku metsa tuvastuse tõenäosus 23%

Juhusliku metsa algoritmi kasutades on tulemus halvem, seega arvestame inimese tuvastuse tõenäosuse selle meetodi tulemusena 40%, mis on inimese tuvastuse tõenäosus kasutades vaid otsustuspuud.

Tabel 2. Tulemused liikumise massi ja otsustuspuud või juhuslikku metsa kasutades.

	Otsustuspuu	Juhuslik mets
a1	+	-
a2	+	-
a3	-	-
b1	-	-
b2	-	-
b3	+	-
c1	+	+
c2	+	+
c3	+	+
d1	-	-
d2	-	-
d3	-	-
e1	-	-
e2	+	+
e3	-	-
f1	-	-
f2	-	-
f3	-	-
g1	-	-
g2	-	-
g3	+	+
h1	-	+
h2	-	-
h3	+	+
i1	+	-
i2	+	-
i3	+	-
j1	-	-
j2	-	-
j3	-	-

5 Analüüs

Töös kasutatud Bezier' kõvera ning dünaamilise ajadeformatsiooni meetodi abil on võimalik inimest tuvastada suure tõenäosusega. Liikumise massi parameetrite ning otsustuspuu meetodi abil antud töös sama häid tulemusi ei saadud. Esimese meetodi inimese tuvastuse tõenäosus on 86%, teisel meetodil 40%.

Esimese meetodi puhul on märgata, kuidas mõne inimese tuvastamisega on eksimusi rohkem, kui teistel, samas mõne inimese tuvastas meetod igal korral ära.

Teise meetodi juures on märgatav asjaolu, et kahte inimest tuvastab meetod kõigil kordadel otsustuspuu abil ära. Nende kahe inimesega samades tingimustes on salvestatud ka mõningate teiste inimeste kõnnakud, seega ei saanud olla põhjuseks vaid välised tegurid, näiteks teistsugune valgus või Kinect sensori vale asend.

Teise meetodi puhul oli ka näha, kuidas parameetrite Fisher'i skoorid on arvestatavalt suuremad juhul, kui vaid ühe inimese liikumise massi parameetrid on teiste omadest erinevad, seega oli keeruline valida, milliste parameetrite järgi oleks mõistlik klassifitseerida.

Kindlasti oleksid töö tulemused paremad ka siis, kui oleks kasutatud mõnda varem salvestatud andmebaasi inimeste kõnnakute parameetritest. Praegusel juhul on kõnnakud salvestatud eri aegadel ning erinevates ruumides, seega pole välistatud, et välised tegurid mõjutasid salvestuste korrektsust. Liikumise massi parameetrite ning otsustuspuu meetodi puhul jäi väheseks Kinect sensori ees tehtavaid samme, rohkemate sammude korral oleks saanud võtta sammude keskmiseid liikumise massi parameetreid, praegusel juhul oli võimalik arvestada vaid ühte parema jalaga tehtavat sammu ning ühte vasaku jalaga tehtavat sammu.

5.1 Võrdlused varasemate töödega

Tabel 3 illustreerib käesoleva töö ning mõningate varasemalt tehtud tööde tulemusi. Tabelist on näha, et käesolevast tööst parem tulemus saadi vaid lähima naabri algoritmi ja lineaarse diskriminandi klassifikaatorit kasutades. Käesoleva tööga võrdne tulemus on saadud Manifoldi õppimist ning HMM meetodit kasutades. Veidi madalamad

tulemused on saadud kolme teise meetodiga: massivektori ja dünaamilise ajadeformatsiooni, HMM-i ning Fourier' deskriptorite meetoditega.

Tabel 3. Võrdlused teiste töödega.

Meetod	Tulemus
Bezier' kõver ja DTW	86%
Liikumise massi parameetrid ja otsustuspuid	40%
Fourier' deskriptorid [1]	85.2%
Massivektor ja DTW[2]	85%
Manifoldi õppimine ja HMM [3]	86%
NN, LDC [4]	94.37%
HMM [5]	85%

6 Kokkuvõte

Töö tulemusena valmis rakendus, mis realiseerib kaks inimese tuvastuse meetodit. Esimese meetodi puhul genereeriti inimese kõnnaku parameetrite järgi Bezier' kõverad nelja erineva sammufaasi kohta ning võrreldi neid dünaamilise ajadeformatsiooni algoritmiga, lisaks võrreldi kahte liikumise massi parameetrit, mis lisas meetodile efektiivsust. Teise meetodina prooviti kasutada liikumise massi parameetreid, mida klassifitseeriti otsustuspuu abil.

Esimene meetod on võimeline inimest tuvastama piisavalt hea täpsusega, kuid teine meetod sobib ilmselt paremini näiteks erinevate haigusseisunditega inimeste ning tervete inimeste kõnnakute võrdlemiseks. Praegusel juhul olid teadaolevalt kõik katses osalenud inimesed terved, seega nende kõnnakud olid üksteisega selle meetodi järgi sarnased.

Kasutatud kirjandus

- [1] S. Yu, L. Wang, W. Hu ja T. Tan, „Gait analysis for human identification in frequency domain,“ IEEE, Hong Kong, 2004.
- [2] S. Hong, H. Lee, I. F. Nizami, S.-J. An ja E. Kim, „Human identification based on gait analysis,“ IEEE, Seoul, 2007.
- [3] M. H. Cheng, M. F. Ho ja C. L. Huang, „Gait Analysis for Human Identification through Manifold Learning and HMM,“ IEEE, New Orleans, 2007.
- [4] A. Dubois ja J.-P. Bresciani, „Person identification from gait analysis with a depth camera at home,“ IEEE, Milan, 2015.
- [5] S. Ali, Z. Wu, M. Zhou, G. Du, X. Li ja F. Pengcheng, „Human Identification Using Sensors Data Based on 3D Gait Area,“ IEEE, Santander, 2014.
- [6] A. Sabir, M. Ahmed, A. Al-Talabani ja H. Maghdid, „Human gait identification using Kinect sensor,“ Kurdistan Journal of Applied Research, 2017.
- [7] S. Nõmm ja A. Toomela, „An alternative approach to measure quantity and smoothness of the human limb motions,“ *Estonian Journal of Engineering*, p. 298–308, 2013.
- [8] S. Nõmm, A. Toomela, M. Vaske, D. Uvarov ja P. Taba, „An Alternative Approach to Distinguish Movements of Parkinson Disease Patients,“ *IFAC-PapersOnLine*, kd. 49, nr 19, pp. 272-276, 2016.
- [9] W.-N. Mohd-Isa, J. Abdullah ja C. Eswaran, „Gait classification of twins and non-twins siblings,“ IEEE, Singapore, 2013.
- [10] M. Lapuškin, „Application for Control of the „Pioneer” Robot with Manipulator,“ 2009.
- [11] K. Parm, „Häälletusautomaat liigutuste õppimiseks ja tuvastamiseks kirurgi robot-assistendis,“ 2010.
- [12] M. Lapuškin, „Žestide peal baseeruv rakendus "Pioneer" roboti juhtimiseks,“ 2012.
- [13] S. Kirme, „Käežesti tuvastus Bezier’ kõverat ja K-lähima naabri meetodit kasutades,“ 2016.
- [14] S. Erap, „Gesture Based PC Interface with Kinect Sensor,“ 2012.
- [15] J. Boruško, „Alternatiivne meetod inimtoorika modelleerimiseks,“ 2014.
- [16] H. Lissenko, „Mitme Kinecti kasutamine pööramisliigutuse registreerimiseks,“ 2015.
- [17] A. Abdelhady, „Odav meetod kõnnaku registreerimiseks pööramisliigutuse jooksul,“ 2017.
- [18] J.-J. Bernstein, „Kontekstipõhine inimliigutuste registreerimine ja analüüs,“ 2017.
- [19] K.-G. Abrams, „Inimkeha mudeli korrektsuse valideerimine,“ 2017.
- [20] A. Krajuškina, „Alternatiivne meetod Parkinsoni patsientide kõnnaku analüüsimiseks,“ 2017.
- [21] S. Liu, M. Sun, W. Wang ja F. Wang, „researchgate,“ märts 2017. [Võrgumaterjal]. Available: https://www.researchgate.net/figure/The-structure-of-Kinect-V2_fig1_315655916. [Kasutatud 17 mai 2018].

- [22] S. Crawford, „How Microsoft Kinect Works“.
- [23] „msdn.microsoft.com,“ [Võrgumaterjal]. Available: <https://msdn.microsoft.com/en-us/library/hh438998.aspx>. [Kasutatud 17 mai 2018].
- [24] N. Jianwattanapaisarn, A. Cheewakidakarn, N. Khamsemanan ja C. Nattee, „Human Identification using Skeletal Gait and Silhouette data extracted by Microsoft Kinect,“ IEEE, Kitakyushu, 2014.
- [25] „docs.microsoft.com,“ [Võrgumaterjal]. Available: [https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn782037\(v%3dieb.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn782037(v%3dieb.10)). [Kasutatud 25 aprill 2018].
- [26] H. Prautzsch, W. Boehm ja M. Paluszny, Bézier and B-Spline Techniques, Berlin: Springer-Verlag Berlin Heidelberg, 2002.
- [27] M. Müller, „Dynamic Time Warping,“ %1 *Information Retrieval for Music and Motion*, Berlin, Heidelberg, Springer, 2007, pp. 69-84.
- [28] „mirlab.org,“ [Võrgumaterjal]. Available: <http://mirlab.org/jang/books/dcpr/dpDtw.asp?title=8-4%20Dynamic%20Time%20Warping&language=english>. [Kasutatud 15 mai 2018].
- [29] C. C. Aggarwal, Data Mining, New York: Springer International Publishing, 2015.
- [30] „accord-framework.net,“ [Võrgumaterjal]. Available: <http://accord-framework.net/>. [Kasutatud 9 mai 2018].
- [31] „scikit-learn.org,“ [Võrgumaterjal]. Available: <http://scikit-learn.org/stable/modules/tree.html>. [Kasutatud 2 mai 2018].
- [32] K. Käärman. [Võrgumaterjal]. Available: http://www.quaretec.com/u/vilo/edu/2003-04/DM_seminar_2003_II/Raport/P02/main.pdf. [Kasutatud mai 2018].
- [33] „crsouza.com,“ [Võrgumaterjal]. Available: <http://crsouza.com/2012/01/04/decision-trees-in-c/>. [Kasutatud 2 mai 2018].
- [34] „blog.yhat.com,“ [Võrgumaterjal]. Available: <http://blog.yhat.com/posts/random-forests-in-python.html>. [Kasutatud 2 mai 2018].
- [35] „stackoverflow.com,“ [Võrgumaterjal]. Available: <https://stackoverflow.com/questions/5550394/how-to-produce-an-interpolated-function-from-a-given-x-and-fx-points-using-new>. [Kasutatud detsember 2017].
- [36] „github.com,“ [Võrgumaterjal]. Available: <https://gist.github.com/socrateslee/1966342>. [Kasutatud jaanuar 2018].
- [37] „accord-framework.net,“ [Võrgumaterjal]. Available: http://accord-framework.net/docs/html/T_Accord_MachineLearning_CrossValidation.htm. [Kasutatud aprill 2018].
- [38] „accord-framework.net,“ [Võrgumaterjal]. Available: http://accord-framework.net/docs/html/T_Accord_MachineLearning_DecisionTrees_DecisionTree.htm. [Kasutatud aprill 2018].
- [39] „csharp-helper.com,“ [Võrgumaterjal]. Available: <http://csharp-helper.com/blog/2014/09/use-transformations-draw-graph-wpf-c/>. [Kasutatud jaanuar 2018].
- [40] „docs.microsoft.com,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/dotnet/framework/winforms/advanced/how-to-draw-a-sequence-of-bezier-splines#compiling-the-code>. [Kasutatud detsember 2017].
- [41] „pterneas.com,“ [Võrgumaterjal]. Available: <https://pterneas.com/2014/03/13/kinect-for-windows-version-2-body-tracking/>. [Kasutatud november 2018].

Lisa 1

Tööle on lisatud kaasa käesoleva töö raames loodud rakendus, milles kasutatud teiste autorite poolt kirjutatud programmikoodid on kasutatud kirjanduse all viidatud [35] [36] [37] [38] [39] [40] [41]. Samuti on lisatud .csv laiendiga failid, mille abil saab rakendust testida. Rakenduses on olemas õpetus, kuidas rakendust kasutada, selle leiab menüüribalt nupu „Abi“ alt.