

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Arvutiteaduse instituut

ITI40LT

Siim Kirme 103734

**KÄEŽESTI TUVASTUS BEZIER' KÕVERAT
JA K-LÄHIMA NAABRI MEETODIT
KASUTADES**

Bakalaureusetöö

Juhendaja: PhD Sven Nõmm

Vanemteadur

Tallinn 2016

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: Siim Kirme

(kuupäev)

(allkiri)

Annotatsioon

Käesoleva töö põhieesmärk on käe liigutuste tuvastus e. žesti tuvastus. Käe asukohta ja liikumist jälgides peab tuvastama, millise varem ettenäidatud e. õpetatud žestiga on tegu. Selleks välja töötatud lahenduse uudsus seisneb Bezier' kõvera ja K-lähima naabri algoritmi koos kasutamises. Bezier' kõveraid on kasutatud inimeste identifitseerimiseks inimese kehaosade asukohti jälgides, käesolevas töös kasutatakse Bezier' kõverat žestide tuvastamiseks inimese kehaosade asukohti jälgides. Välja töötatud meetod viib sarnased žestid omavahel kokku, kasutades K-lähima naabri algoritmi. Töö tulemusena on valminud rakendus, mis implementeerib mainitud meetodit, rakendusele on võimalik õpetada žeste ja rakendus on võimeline žestituvastamiseks. Rakenduse žesti tuvastuse efektiivsuse kontrolliks tehtud katsete tulemused näitavad et lihtsate žestide tuvastamine toimub edukalt. Kuid kinnises kontuuris žestide tuvastamine (simuleeritud operatsiooni käigus kirurgi liigutuste tuvastamine) näitas, et süsteem vajab häälestamist tuvastatavale isikule.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 37 leheküljel, 7 peatükki, 8 joonist, 2 tabelit.

Abstract

The main goal of the thesis is gesture recognition. Hand location and hand movement data is used to recognise learned gestures. The method developed in this thesis uses Bezier' curves and K-nearest neighbours algorithm in unique combination with each other. Bezier' curves have been used to identify persons by observing the locations of body parts, in this thesis Bezier' curve is used similarly for gesture recognition. Developed method matches similar gestures using K-nearest neighbours algorithm. The result of this thesis is an application which implements mentioned method, it is possible to teach gestures to the application and the application is capable of recognising recorded gestures. The results of tests, that show gesture recognition efficiency, show that the application successfully recognises simple gestures. However gesture recognition in a closed cycle (recognising gestures during simulated surgery) showed that the system needs tuning to the person doing the gestures.

The thesis is in Estonian and contains 37 pages of text, 7 chapters, 8 figures, 2 tables.

Lühendite ja mõistete sõnastik

Bezier' parameetrid	<i>Bezier' parameters</i> Kasutatakse viitamisel Bezier' kõverat iseloomustavale kolmele parameetrile: x-dispersioon, y-dispersioon, keskmine.
X-dispersioon	<i>X-variance</i> Koordinaatide x dimensiooni väärtuste dispersioon.
Y-dispersioon	<i>Y-variance</i> Koordinaatide y dimensiooni väärtuste dispersioon.
Žesti parameetrid	<i>Gesture parameters</i> Bezier' parameetrite hulk mis defineerib ühe žesti andmebaasis.

Jooniste nimekiri

Joonis 1. Bezier' kõver.....	13
Joonis 2. Kinect for Xbox One.....	15
Joonis 3. Parema käe Bezier' kõver.....	17
Joonis 4. Dispersioonide võrdlus.....	18
Joonis 5. Andmetöötlusprotsess.....	19
Joonis 6. Käelehvituste K-keskmiste klasterdus.....	21
Joonis 7. Käeringi K-keskmiste klasterdus.....	21
Joonis 8. Parema käe jälgimine Kinect kaameraga.....	24

Tabelite nimekiri

Tabel 1. Ringi ja lehvitus tuvastus tulemused.	28
Tabel 2. Kirurgiliigutuste tuvastuse tulemused	29

Sisukord

1. Sissejuhatus.....	9
1. Kirjanduse ülevaade	10
2. Probleemi püstitus.....	12
3. Meetodid ja tehnoloogia.....	13
3.1 Meetodid ja definitsioonid	13
3.1.1 Bezier' kõver	13
3.1.2 Dispersioon	14
3.1.3 K-keskmiste klasterdamine	14
3.1.4 K-lähima naabri algoritm.....	14
3.2 Tehnoloogia	15
3.2.1 Riistvara	15
3.2.2 Tarkvara	15
4. Lahendus.....	16
4.1 Meetod ja algoritm	16
4.1.1 Žestide õpetamine.....	16
4.1.2 Žestide tuvastamine	22
4.2 Implementatsioon	23
4.3 Näide.....	25
4.3.1 Loodud rakendus Kinect Body Track.....	25
4.3.2 Rakendusele žestide õpetamine.....	26
5. Tulemused	28
6. Märkmeid lähtekoodist.....	30
7. Arutelu.....	32
Kokkuvõte	33
Summary	34
Kasutatud kirjandus	35
Lisa.....	37

1. Sissejuhatus

Žestide tuvastamine on tänapäeval teema millega tegeldakse paljudes erinevates valdkondades eri põhjustel: meditsiinis patsiendi jälgimisel, videomängudes füüsilise kontrolleri asemel, tööstuses robotkäe juhtimiseks.. selles töös uuritakse meetodit inimese käežesti tuvastamiseks Microsofti Kinect kaamera abil.

Žestide töötlemiseks on palju eri meetodeid: erinevad statistilised meetodid, masinõppe meetodid jne. Žestide tuvastamisega tihedalt seotud teema on muuhulgas inimese kõnnaku jälgimine inimeste tuvastamiseks. [1] jälgitakse inimese kõnnakut ning Bezier' kõverat kasutades eristatakse edukalt inimesi nende kõnnaku järgi. Käesolevas töös töötatakse välja meetod žestide tuvastamiseks, kasutades analoogselt Bezier' kõverat koos masinõppe meetodiga K-lähimat naabrit.

Töö tulemusena valmis rakendus, millele on võimalik žestide õpetamine ning mis on võimeline reaalajas nimetatud meetodi abil žeste tuvastama. Välja töötatud meetod Bezier' kõverat kasutades on võimeline tuvastama inimeste žeste, kes ise ei osalenud žestide õpetamisel. Töö käigus õpetati rakendus tuvastama kahte žesti: käelehvitus ja käega ringi tegemine. Nende žestide rakendusele õpetamisel osales seitse inimest, kes salvestasid kokku 419 žesti, kumbagi žesti üle 200 korra. Žesti tuvastamise katsetes suutis rakendus reaalajas tuvastada lehvitus žesti 76,6% tõenäosusega ja käega ringi tegemise žesti 86,6% tõenäosusega. Lisaks õpetati rakendust jälgima ühte protsessi laparoskoopilise operatsiooni käigus. Simuleeritud operatsiooni käigus õpetati rakendust tuvastama kirurgižeste: lauvalt operatsiooniriista üles võtmist ja enda juurde toomist, riista eemaldamist patsiendist pärast tööd ning riista lauale tagasi panemist. Žeste õpetas üks inimene, igat žesti ~40 korda rakendusele ette näidates. Žestide tuvastamise katsetes, kus osales kolm inimest, tuvastati lauvalt riista võtmise žesti 46,7% tõenäosusega, riista patsiendist eemaldamise žesti tuvastati 56,7% tõenäosusega ning riista lauale tagasi asetamist tuvastati 90,0% tõenäosusega.

1. Kirjanduse ülevaade

Kinect kaamerat on laialdaselt kasutatud žestituvastuse probleemide lahendamiseks kaamera odavuse ja suhtelise efektiivsuse tõttu. Kinect kaamerat kasutatakse tihti osana suuremast süsteemist, mis tegutseb Kinectiga tuvastatud inimeste käitumise järgi.

Kinect kaamera on võimeline tuvastama inimesi ja inimese kehaosade asukohta. [2] pakub välja Kinect kaamera põhise lahenduse vanadekodus „HomeMate“ robotile eakate inimeste abistamiseks. Robot peaks tooma inimestele jooke ja ravimeid, sealjuures oleks Kinecti tööülesandeks tuvastada inimesi ning mõnda käežesti (enda poole kutsuv käe viibe, kandikult joogi võtmise käeliigutus) vanadekodu kirjus keskkonnas. Katsete käigus selgus, et Kinectil on üks suuremaid probleeme istuva inimese tuvastamine (see osutuks probleemiks vanadekodus inimesetuvastamisel). [2] pakuti välja Kinecti väliseid inimesetuvastus meetodeid, mis töötavad koos Kinectiga. [3] ja [4] kasutati Kinecti robotkäe juhtimiseks, Kinect tuvastab käe liikumise ning robotkäsi liigub vastavalt. [4]-s liigub inimese juhtimisel terve robot, mille külge on ehitatud juhitav käsi.

Selle töö üks kaugemaid eesmärke oleks kasutada Kinect kaamerat, et juhtida robot-assistenti operatsiooni käigus.

Bezier' kõverat on kasutatud žestituvastamisel.[5] sisendiks olevatest 3-D andmetest jägitakse käelaba liikumist. Käelaba liikumisest moodustatakse Bezier' kõver. [6] eeldatakse, et inimene on kaamera suunas avatud käelabaga. Nahavärvi järgi tuvastatakse käelaba ning seejärel arvutatakse käelaba keskpunkt. Keskpunkti liikumisest moodustatakse Bezier' kõver.

[5], [6] moodustatakse Bezier' kõver ühe kehaosa (käelaba) liikumise põhjal ning ühte žesti iseloomustab üks Bezier' kõver. Selles töös moodustatakse Bezier' kõver mitme kehaosa omavahelise asukoha põhjal ning ühte žesti iseloomustab üks Bezier' kõver iga kaadri kohta, millal žest toimus.

[7] arendati meetod Kinect kaamera abil sõrmede identifitseerimiseks ja käemärkide eristamiseks. Meetodi täpsuse testimiseks loodi mäng, kus mängija peab sõrmeotstega puuvilju löikuma. Mängijate tagasisides mängu kohta oli positiivseks küljeks süsteemi

intuitiivsus, aga frustratsioon tekkis kohati ebatäpse käe ja sõrmeotste kujutistega ning olukorras, kus käemärke ei tundud ära. Rakenduse arendajad väitsid, et probleemi tekitas Kinect v2 kaamera (sama kaamera, mida kasutatakse selles töös) sügavuskaamera müra.

[8] esitab meetodi käemärkide eristamiseks Kinecti salvestatud punktipilve abil 3D ruumis. Jälgides iga (käe)punkti ümbritseva punktipilve tihedust, valitakse skeletipunktid. Iga välja valitud punkti ümbritsev ruum jaotatakse sektoriteks, sektorites olevad skeletipunktid loetakse kokku ning moodustub histogramm. Moodustatud histogrammide kaugus üksteisest viitab punktide kaugusele üksteisest. Iga skeletipunkti kohta arvutati 72-dimensiooniline histogramm. Taoliste histogrammi massiivide omavaheline sobitamine (käemärgi tuvastamise eesmärgil) ei andnud täpsemaid tulemusi seni väljatöötatud meetoditest, kuid meetod ja tulemused näitasid potentsiaali edasisteks uuringuteks.

[9] võrreldakse Kinect-i ja Kinect 2-e inimese liigutuste tuvastamisel. Kaks musta vööga oyama-karate instruktorit teostasid kümmet eri liigutust (kaitsvad tehnikad, löögid, seisakud) kokku 400 korda. Kinect 2 oli enamiku liigutuste identifitseerimisel sama edukas või edukam Kinectist. Kinectil oli probleeme jalaliigete täpse asukoha ja orientatsiooni identifitseerimisega, erinevus on seletatav Kinect 2-e parema sügavussensoriga.

[10] identifitseeritakse inimesi nende kõnnaku järgi. Selleks kasutatakse Kinecti identifitseeritud inimese skeleti, arvutatakse kehapunktide vahelised kaugused, inimese pikkus ja mõõdetakse kui suure nurga all erinevad liigesed on. Analüüsiti, mis kehaosad ja mis mõõtmismeetodid annavad kõige paremaid tulemusi inimese identifitseerimisel. Leiti, et inimese keha alumine osa andis kvaliteetsemat informatsiooni kui keha ülaosa, arvati, et kui Kinect kõnnakut kõrvaltvaates salvestab, siis ülakeha on varjatud kui jalad. Lisaks selgus katsetes, et kehaosade vahelised kaugused sobisid paremini inimeste identifitseerimiseks kui liigete vahelised nurgad.

[11] tuvastatakse Kinecti abil India klassikalise tantsu Odissi tantsija tundeid. Odissi tantsus väljendavad keha hoiak ja liigutused viha, hirmu, rõõmu jne. Tunnete identifitseerimisel saavutatakse edu, kui jälgitakse kehaosade omavahelist paiknemist ja käte liikumise kiirust.

2. Probleemi püstitus

Käesoleva töö põhiülesanne on žestituvastus. Selleks on tuleb lahendada järgmised probleemid:

- Arendada rakendus mis on võimeline žestide salvestamiseks Microsoft Kinect kaamera abil.
- Välja töötada ja implementeerida meetod žestide töötlemiseks.
 - Žestide kujutamine Bezier' kõverana.
 - Bezier' kõvera parameetrite jälgimine žesti jooksul.
- Žesti tuvastamine.
 - Žestide andebaasi salvestamine Bezier' kõvera parameetritena.
 - Tuvastatava žestiga sarnaste žestide andmebaasist üles leidmine kasutades K-lähima naabri algoritmi.
- Süsteemi efektiivsuse katsetamine kinnises kontuuris.
 - Žestide tuvastamine laparoskoopilise operatsiooni simulatsiooni käigus.

3. Meetodid ja tehnoloogia

Selles töös kasutatakse Kinectist saadud info töödelduseks Bezier' kõverat ja dispersiooni. Arendatud rakenduse masinõppe osas kasutatakse k-keskmiste klasterdamist ja k-lähimate naabrite algoritmi. Riistvara osas kasutatakse Microsofti Kinect kaamerat. Järgnevalt kirjeldatakse nimetatud meetodeid ja kasutatud tehnoloogiaid.

3.1 Meetodid ja definitsioonid

3.1.1 Bezier' kõver

Bezier' kõver on defineeritud hulga kontrollpunktidega $P_0..P_n$. Kõverat saab esitada rekursiivse funktsiooniga:

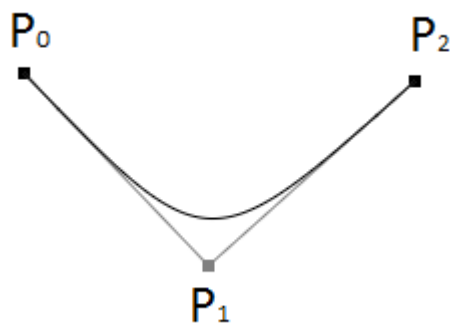
$$B_{P_0}(t) = P_0,$$

$$B(t) = B_{P_0P_1\dots P_n}(t) = (1 - t) \times B_{P_0P_1\dots P_{n-1}}(t) + t \times B_{P_1P_2\dots P_n}(t)$$

Kus $B_{P_0P_1\dots P_n}(t)$ on Bezier' kõver defineeritud punktidega $P_0P_1 \dots P_n$ ja $t \in [0,1]$.

Rekursiivse funktsiooni lihtsuse tõttu (seda on kerge implementeerida) kasutati mainitud meetodit arendatud rakenduses Bezier' kõvera arvutamiseks.

Joonisel (Joonis 1. Bezier' kõver.) on esitatud kolme punktiga defineeritud Bezier' kõver.



Joonis 1. Bezier' kõver.

Bezier' kõverat kasutatakse laialdaselt arvutigraafika disainis kõverate ja kurvide modelleerimisel (nt. Adobe Illustrator'i tööriist Pen Tool[11]).

3.1.2 Dispersioon

Dispersioon on juhusliku suuruse X hajuvuse mõõt[12]. Dispersiooni arvutamise valem:

$$\sigma_x^2 = \frac{1}{n} \sum_{i=0}^{n-1} [x_i - \bar{x}]^2$$

3.1.3 K-keskmiste klasterdamine

K-keskmiste klasterdamine (inlg. k. k-means clustering) on meetod klastrite ja klastrite keskpunktide leidmiseks andmehulgast. Valides sobiva klastrite arvu K, liigutab K-keskmiste protseduur klastrite keskpunkte et vähendada klastrisest dispersiooni nii palju kui võimalik. Valides esialgsed klastrite keskpunktid, täidab K-keskmiste algoritm vaheldumisi järgmised sammud:

- Iga keskpunkti jaoks leitakse alamhulk sisendandmehulgast, mille liikmed on vastavale keskpunktile lähimad.
- Iga alamhulga keskmised arvutatakse ning sellest saab vastava klastrite uus keskpunkt

Algoritm lõpetab töö, kui andmepunktide klastrissekuuluvus ja klastrite keskpunktide asukohad ei muutu. [13][14][15]

3.1.4 K-lähima naabri algoritm

K-lähima naabri (ingl. k. k-nearest neighbors - KNN) algoritmi kasutatakse klassifitseerimiseks. Võttes punkti x_0 , mida tahetakse klassifitseerida, leitakse klassifitseeritud andmehulgast k andmepunkti, mis on lähimad x_0 -le. Enamuse arvamus k-hulgas otsustab x_0 -i klassi kuuluvuse. Mitme klassi võrdse tulemuse korral valitakse neist suvaline: kui hääle kogumisel jäävad kaks või enam klassi viiki, siis x_0 saab endale suvalise viiki jääva klassi väärtuse.[13][14]

3.2 Tehnoloogia

3.2.1 Riistvara

Kinect on Microsofti poolt arendatud kaamera mille abil on võimalik suhelda Xbox'i mängukonsooliga žestide ja hääle abil. Kaamera sisaldab endas RGB kaamerat, sügavussensorit (infrapuna kaamera) ja mikrofone. Selles töös kasutati Kinect for Xbox One kaamerat, millel on võrreldes vanema generatsiooniga muuhulgas kõrgem kaamera resolutsioon ja laiem kaamera vaateväli. Uus kaamera suudab korraga tuvastada kuut inimest varasema kahe asemel.



Joonis 2. Kinect for Xbox One.

3.2.2 Tarkvara

Kinect kaamerat on võimalik ühendada lisaks mängukonsoolile ka Windowsi arvuti külge. Tarkvara poolelt on Microsoft pakkunud toeks Kinect for Windows SDK 2.0. Lisaks on selle töö raames arendatud Bezier' kõvera moodustamise meetod ja K-nearest neighbors algoritm. Rakendus on kirjutatud C# keeles Microsoft Visual Studio keskkonnas.

Bezier' kõvera illustatsiooniks on kasutatud avatud lähtekoodiga tarkvara veebisaidil <http://www.victoriakirst.com/beziertool/> [16].

4. Lahendus

4.1 Meetod ja algoritm

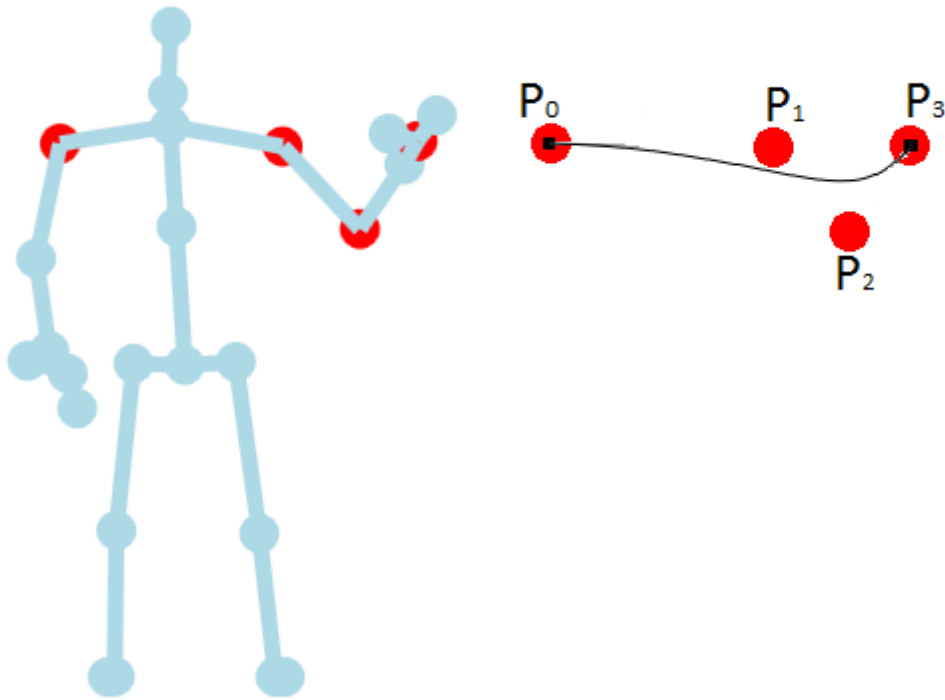
Inimese käežestide jälgimisel kasutatakse andmetöötlusprotsessi mis koosneb kolmest osast: inimese liigutuste registreerimine, registreeritud liigutuste põhjal Bezier' kõvera koostamine ja Bezier' kõvera parameetrite arvutamine. Inimese käežestide registreerimiseks kasutatakse Microsofti poolt arendatud Kinect kaamerat. Kinect võimaldab salvestada 25 punkti inimese kehal, mis kirjeldavad inimese keha asendit, nt. õlg, küünarnukk, käetipp. Registreeritud liigutuste põhjal koostatakse inimese kehal väljavalitud punktide (vasak õlg, parem õlg, parema käe küünarnukk, parema käe käelaba) abil Bezier' kõver. Seejärel arvutatakse Bezier' kõvera parameetrid: dispersioonid ja keskmised väärtused (kokku 3 parameetrit) – need parameetrid on andmetöötlusprotsessi väljunditeks.

Et arvuti tuvastaks inimese žeste tuleb kõigepealt moodustada žestide kogum mida arvuti hakkab tuvastama. Inimene teostab Kinecti kaamera ees žesti ning Kinect salvestab infot kuni 30 kaadrit sekundis. Iga kaader läbib mainitud andmetöötlusprotsessi ja tulemuseks saadud parameetrite hulk moodustab konkreetse žesti ühe eksemplari rakenduse poolt kasutatavas andmebaasis. Näiteks kaks sekundit kestev žest võib olla andmebaasis salvestatud 180 parameetriga ($2 \text{ sekundit} * 30 \text{ kaadrit sekundis} * 3 \text{ parameetrit kaadri kohta}$). Iga erineva käežesti kohta salvestatakse andmebaasi ~32 eksemplari. Kümne erineva žesti puhul oleks andmebaasis kokku ~320 žesti eksemplari.

Kui andmebaas on loodud ja on vaja tuvastada tundmatut žesti, siis moodustatakse tundmatu žesti põhjal analoogselt andmetöötlusprotsessi kasutades 1 eksemplar ning „K-nearest neighbors“ algoritmi abil otsustatakse, mis andmebaasi salvestatud žestiga tundmatu žest sarnaneb.

4.1.1 Žestide õpetamine

Selles töös valiti Bezier' kõvera esitamiseks neli kontrollpunkti inimese kehal: vasak õlg, parem õlg, parema käe küünarnukk, parema käe käelaba:

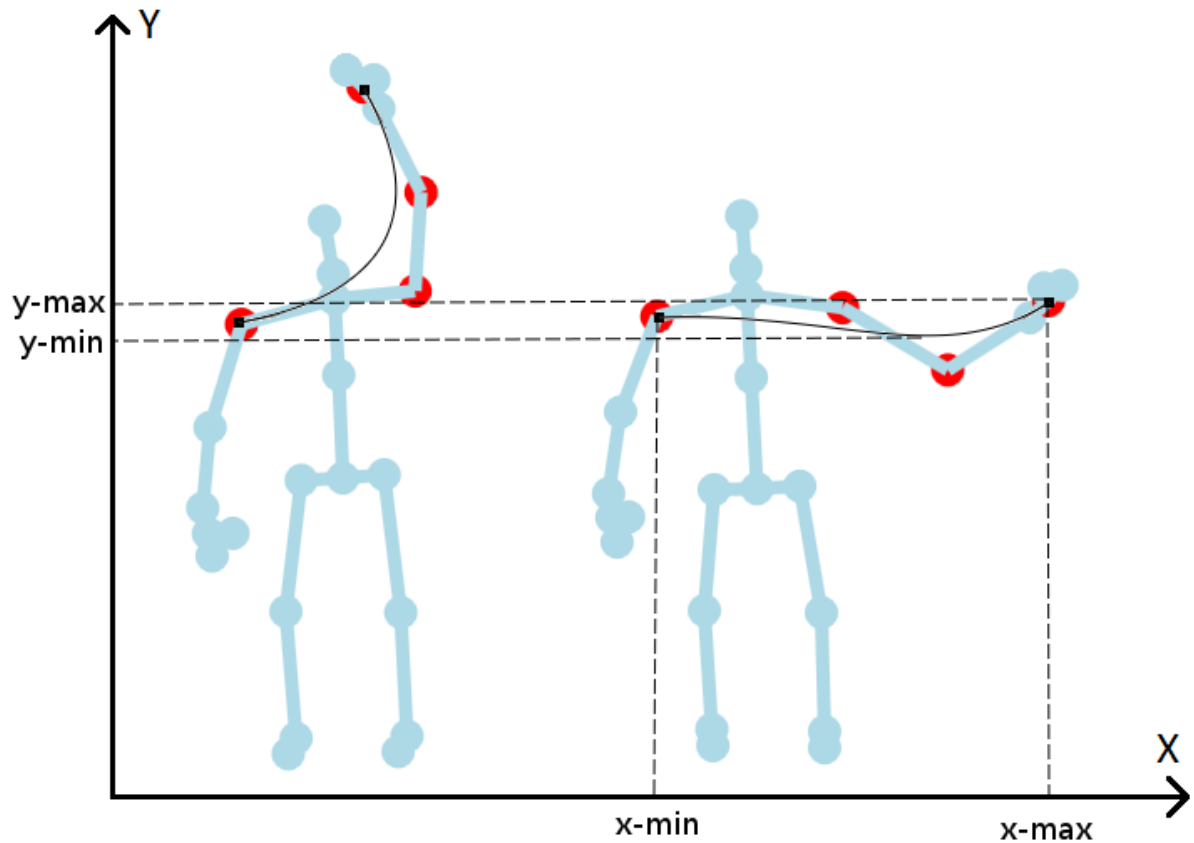


Joonis 3. Parema käe Bezier' kõver.

Parema käe žestide jälgimisel on kontrollpunktide valimiseks loomulik võtta kolm punkti, mis liiguvad kõige rohkem: parem õlg, parem küünarnukk ja parem käelaba. Aga et näha, kuidas käe liigutamisel liigub ka õlg võrreldes ülejäänud kehaga, võeti kontrollpunktiks ka teine õlg. Täpsema tulemuse saaks rohkemate kontrollpunktidega, kuid siis kannataks rakenduse jõudlus.

Žestide võrdlemiseks jälgitakse taolise kõvera kuju muutumist žesti jooksul. Selleks on välja valitud kõvera parameetrid: kõveral asuvate punktide x ja y dimensiooni dispersioonid ning nende keskmine väärtus. Selles töös kasutatakse kolmele arvutatud parameetritele viitamisel „Bezier' parameetrid“ või lihtsalt „parameetrid“.

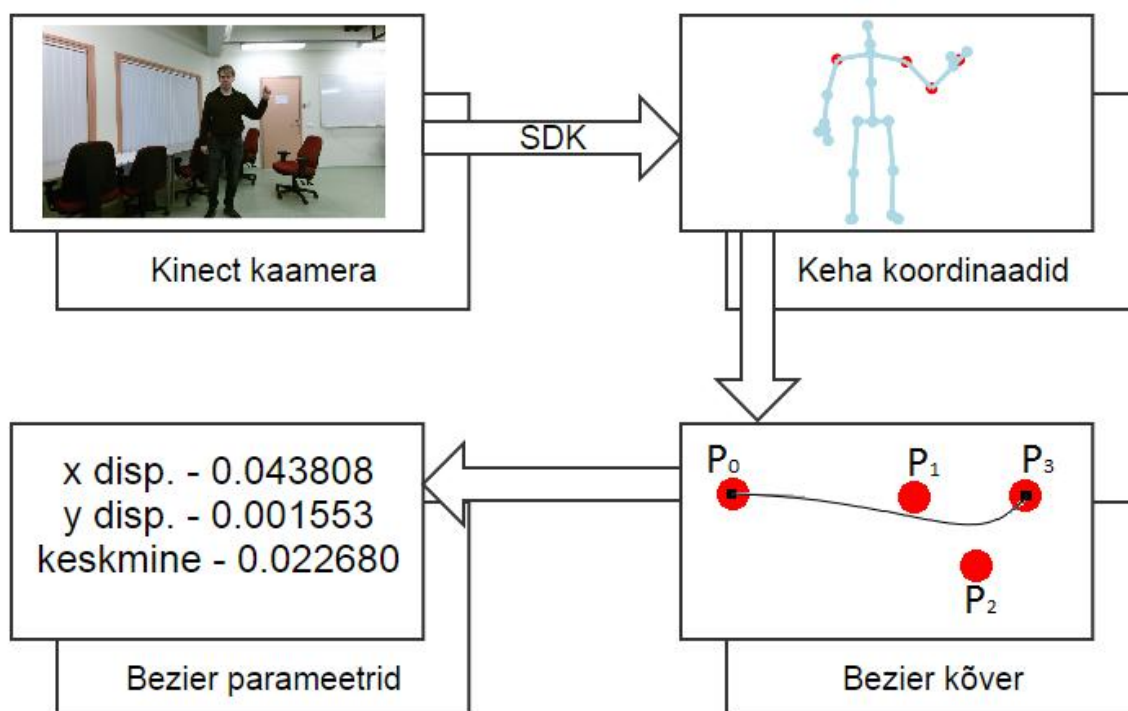
Dispersioon näitab võrreldavate väärtuste varieeruvust (kui palju nad üksteisest erinevad). Näiteks järgneval joonisel (Joonis 4. Dispersioonide võrdlus.) on ühel inimesel käsi üles sirutatud ning nii tema x-dispersioon kui y-dispersioon on võrdlemisi suured. Teisel inimesel: x-dispersioon on suur ja y-dispersioon on väga väike (Bezier' kõveral asuvate punktide x-dimensiooni väärtused varieeruvad rohkem, y-dimensiooni väärtused varieeruvad vähe):



Joonis 4. Dispersioonide võrdlus.

Jälgides eelneval joonisel (Joonis 4. Dispersioonide võrdlus.) olevat parempoolset inimest arvutatakse X-dispersioon punktide põhjal, mille väärtused jäävad vahemikku x-min ja x-max, y-dispersioon arvutatakse punktide põhjal, mille väärtused jäävad vahemikku y-min ja y-max.

Kogu senine andmetöötlusprotsess:



Joonis 5. Andmetöötlusprotsess.

Iga žesti jooksul läbib kaamerast saabunud kaader mainitud protsessi ja nende kohta arvutatakse Bezier' parameetrid. Tulemuseks saadud Bezier' parameetrite hulk defineerib ühe žesti. Taolisele parameetrite hulgale viidatakse selles töös: „žesti parameetrid“. Žesti parameetrid on hulk mille element on „Bezier' parameetrid“. Olenevalt sellest, mitu kaadrit pikk žest on, nii mitu elementi žesti parameetrites on. Sealjuures on Bezier' parameetrid omakorda hulk mis koosneb alati kolmest elemendist, nendeks on arvud: x-dispersioon, y-dispersioon ja keskmine. Näiteks ühe sekundi pikkune žest salvestati 30 kaadriga, iga kaadri kohta arvutati Bezier' parameetrid (st. kolm arvu). Kogu žest on defineeritud (30 * (x-dispersioon, y-dispersioon, keskmine)) 90 arvuga – need on žesti parameetrid.

4.1.1.1 Žesti parameetrid vektoritena

Žesti parameetreid (30 * (x-dispersioon, y-dispersioon, keskmine)) võib vaadelda ka iga parameetri tüübi järgi eraldi: (30 * x-dispersioon) + (30 * y-dispersioon) + (30 * keskmine). Iga vaadeldav osa moodustab vektori. Nende vektoreid võrreldakse žestide tuvastamiseks.

4.1.1.2 K-keskmiste klasterdamine

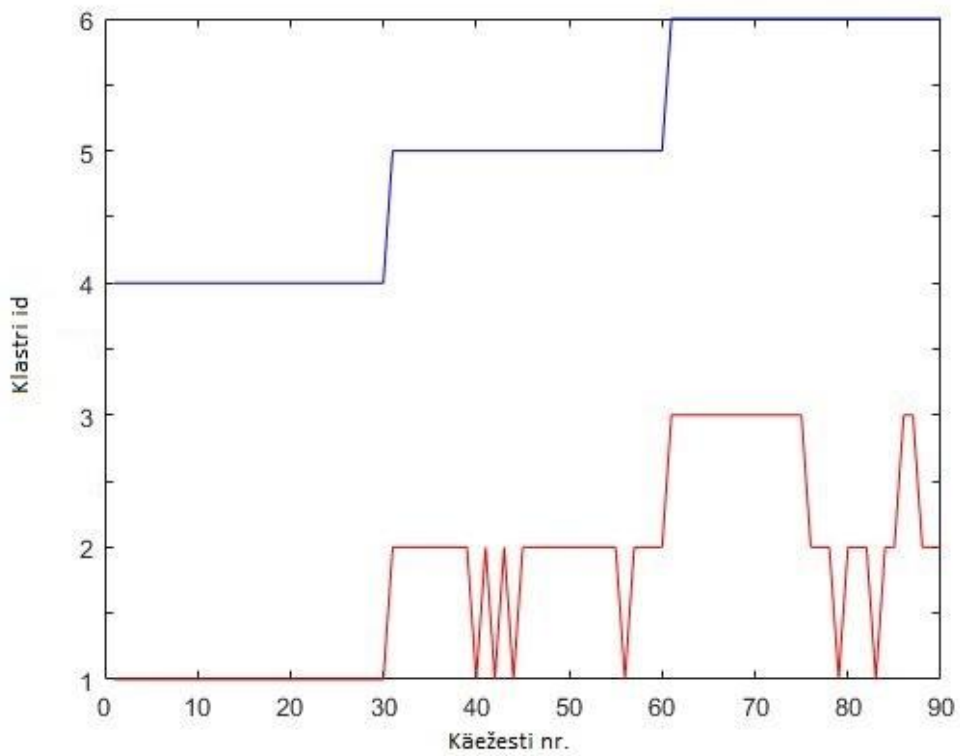
Bezier' parameetreid kasutati uuringus, kus tuvastati inimest tema kõnnaku järgi[1]. St. et inimese kõnnakut jälgides on kõnnaku järgi arvutatavad Bezier' parameetrid omavahel niivõrd eristuvad, et nende järgi on võimalik eristada (ja seega identifitseerida) inimesi. Et proovida žesti parameetrite klastritesse jaotumist, kasutati K-keskmiste klasterdamist (ingl. k. *k-means clustering*). Klasterdamiseks kasutati MATLAB'i.

Žesti parameetrite klasterdamine MATLAB'i abil näitas, et žestid jaotuvad klastritesse suures osas vastavalt nende žestitüübi kuuluvusele. Näiteks käelehvitused jagunesid ühte ja käega ringi tegemise žestid teise klastrisse. See tähendab, et žestid on eristuvad nendega vastavusse pandud Bezier' parameetrite järgi.

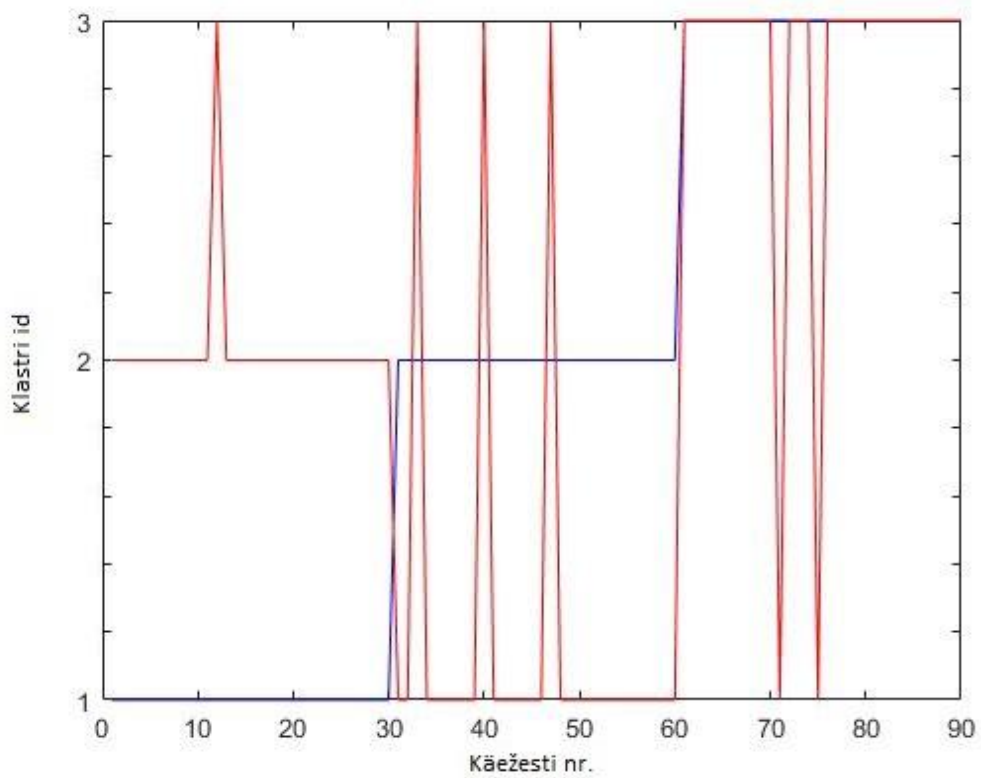
Kuna inimese kõnnaku jälgimise uuringus identifitseeriti inimesi, siis selles töös prooviti kas inimestel oleks võimalik vahet teha ka nende käežestide järgi. Selgus, et K-keskmiste klasterdamise algoritm on võimeline Bezier' parameetrite järgi inimesi eristama.

Katses osales kolm inimest, igaüks tegi 30 käelehvitust. Joonisel (Joonis 6. Käelehvituste K-keskmiste klasterdus.) tähistab sinine joon sisendandmeid, inimese 30 žesti on identifitseeritud vertikaalsel skaalal väärtustega 4, 5 või 6. K-keskmiste algoritm üritas iga žesti jaotada kolme klastrisse: 1, 2 ja 3.

Joonisel (Joonis 7. Käeringi K-keskmiste klasterdus) on kolme inimest analoogselt eristatud teise käežesti, käega ringi tegemine, järgi.



Joonis 6. Käehevõtuste K-keskmiste klasterdus.



Joonis 7. Käeringi K-keskmiste klasterdus.

4.1.2 Žestide tuvastamine

Et rakendus tunneks ära inimese žeste, näidati talle ette, millised need žestid välja näevad ja loodi ettenäidatud žestidest andmebaas. Igat unikaalset žesti salvestati andmebaasi mitmeid kordi. Käega ringi tegemise ja käelehvituse tuvastamise õpetamises salvestati seitsme inimese kohta kokku: 214 ringi ja 205 käelehvitust). Iga žest läbis mainitud andmetöötlusprotsessi (Joonis 5. Andmetöötlusprotsess.).

Kui inimene teostab Kinect kaamera ees žesti, mida rakendus peab ära tundma, siis moodustatakse tehtava (tundmatu) žesti põhjal analoogselt andmetöötlusprotsessi kasutades 1 eksemplar ning K-lähima naabri k-NN (ingl. k. *K-nearest neighbors*) algoritmi abil otsustatakse mis andmebaasi salvestatud žestiga tundmatu žest sarnaneb.

Kuna žesti iseloomustavad kolm parameetrit mida omavahel võrrelda ei saa, siis rakendatakse k-NN algoritmi iga parameetri kohta eraldi (vaata 4.1.1.1 Žesti parameetrid vektoritena), ehk siis algoritmi rakendatakse kokku kolm korda üksteisest sõltumatult. Kui igale parameetrile rakendatud k-NN algoritm annab sama lähima žesti, siis otsustatakse, et tegu on tõepoolest selle žestiga. Näiteks kui x-dispersiooni ja y-dispersiooni järgi on lähim žest käelehvitus, aga kolmanda parameetri „dispersioonide keskmise“ järgi on lähim žest midagi muud, siis otsustatakse, et tegu ei ole teadaoleva žestiga. Katsetati ka varianti, et kui 2/3 parameetritest on kooskõlas, siis on žest identifitseeritud: tulemuseks saadi tunduvalt rohkem *false-positive* tulemusi: tekkisid vead klassifitseerimisel (inimene tegi rakendusele teadaoleva käežesti, aga rakendus tundis selles ära hoopis mingi muu žesti) ja ka suvalisi käevõpatusi hakati identifitseerima legitiimsete žestidena.

3/3 parameetrist meetodi puhul aga ei tundud ära mõnda žesti, mida 2/3 tundis ära. 3/3 töötas katsetel usaldusväärsemalt. Kui võtta kasutusele uusi viise ja meetodeid, kuidas eemaldada 2/3-l tekkinud *false-positive* tulemusi, võib osutada paremaks 2/3 meetod.

Reaalajas žeste tuvastades otsustati katsete käigus võtta k-NN algoritmis $k=15$. See tähendab, et vaadeldi lähimat viitteist žesti. Lisaks kontrolliti, et vaadeldavate žestide kaugus tuvastatavast tundmatust žestist oleks väiksem staatilisest arvust *GestureMaxDistance*. Kui 15 vaadeldavast žestist kaugeima žesti kaugus oli suurem kui *GestureMaxDistance*, siis otsustati et tegu ei ole teadaoleva žestiga. Nii vähenes *false-positive* tulemuste arv.

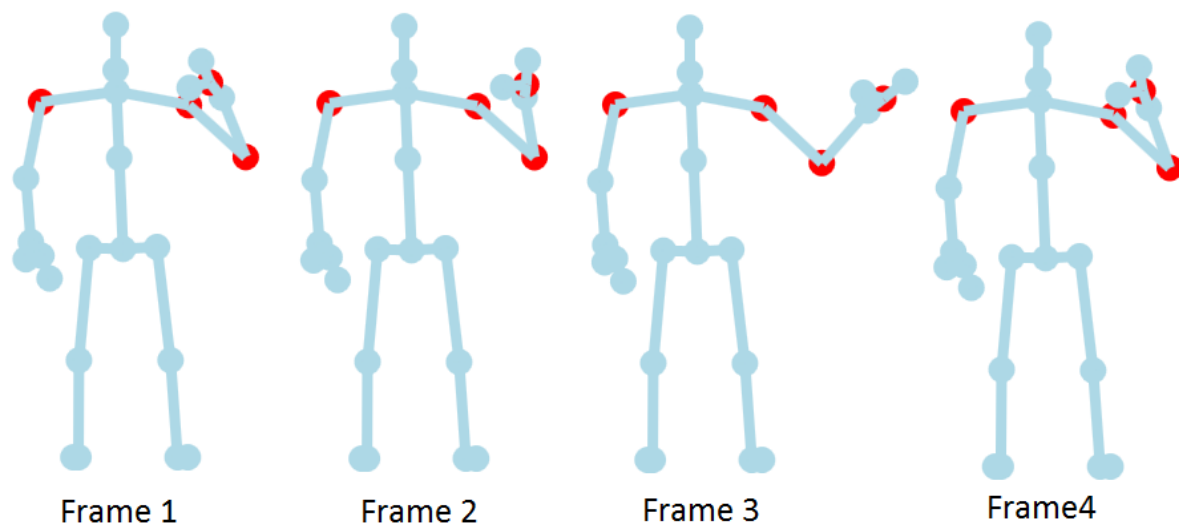
Veel optimeerimisest: k ja $\text{GestureMaxDistance}$ optimaalne väärtus sõltub ka sellest, kas vahet tuleb teha eri žestidel, žesti järgi eri inimesel või mõlemal korraga. Kahe lehvitus vaheline kaugus (isegi kui lehvivad kaks inimest) on tunduvalt väiksem kui kaugus kahe žesti (nt. lehvitus ja käega ringi tegemine) vahel.

Kui vaadeldavad kehaosad ei liigu, siis žesti ei tuvastata. Katsete käigus selgus, et lehvitus tuvastamisel tuvastati käe paigalhoidmist lehvitus positsioonis tihtilugu lehvitus žestina. Kui paigal olevat kätt ignoreerida, vähenes *false-positive* žesti tuvastuste arv. Tundmatu žesti jooksul jälgiti vaadeldavate kehaosade liikumist. Kui kehaosade liikumise trajektooride summeeritud pikkus on väiksem kui määratud vahemaa $\text{MinDistanceTraveled}$, siis žesti ei tuvastata. $\text{MinDistanceTraveled}$ väärtuse valikust: kui inimene on Kinect kaamera ees täiesti paigal, siis sellegipoolest tuvastab Kinect müra tõttu väikest liikumist. $\text{MinDistanceTraveled}$ on kahekordne distants Kinecti müra tõttu arvatud liikumise distantsist paigal seisva inimese korral.

Viimaseks määrati žestile ka minimaalne võimalik kestvus: 18 kaadrit. Kinect salvestab kuni 30 kaadrit sekundis ja pool sekundit kehtvat käevõpatust otsustati mitte jälgida. Žesti minimaalse kestvuse deklareerimine vähendab *false-positive* tulemusi ja parandab rakenduse jõudlust. Kui võtta kasutusele teisi meetodeid *false-positive* tulemuste eemaldamiseks, saaks jälgida ka kiiremaid lühema kestvusega liigutusi.

4.2 Implementatsioon

Kinect suudab jälgida 25 liigest (ingl.k. „Joint“, selles töös kasutatakse ka „punkt kehal“ ja „kehaosa“) inimese kehal. Selles töös tuvastatakse inimese parema käe žeste. Selleks otsustati jälgida nelja punkti inimese kehal: vasakut õlga, paremat õlga, parema käe küünarnukki, parema käe käelaba.



Joonis 8. Parema käe jälgimine Kinect kaameraga.

Iga punkti kohta inimese kehal pakub Kinect selle asukohta kolmemõõtmelises ruumis ja ka selle punkti (kehaosa) orientatsiooni. Selles töös kasutatud Bezier' kõvera meetod kasutab ainult keha asukoha muutumist ruumis, et tuvastada žeste. St. kehaosade orientatsiooni ei võeta arvesse žestituvastamisel, kuigi orientatsiooni saaks teoreetiliselt kasutada liigutuste jälgimisel.

Teine lihtsustus: kehaosa liikumist jälgitakse ainult X ja Y koordinaatide järgi. Uuringus millest see töö on inspireeritud, jälgiti inimese kõnnakut külje pealt, et välja selgitada, kellega oli tegu[1]. St. järeltõlge tehti ainult keha liikumisest kahemõõtmelises ruumis. Selle töö eesmärgiks oli proovida, kas sarnase Bezier' meetodiga saab vahet teha käežestidel. Kuigi Bezier' kõverat saab moodustada ka punktidest kolmemõõtmelises ruumis, jäädigi truuks originaalmeetodile: Bezier' kõverat jälgitakse ainult kahemõõtmelises ruumis. Keerukamat lahendust otsustati enne mitte implementeerida, kui ei ole selgunud, kas meetod suudab inimese tuvastuse asemel tuvastada käežeste.

Žesti tuvastamisel on eeldatud, et inimene seisab näoga kaamera suunas. See annab Kinect kaamerale võimalikult hea ülevaate olukorrast (kaamera ei näe inimese taha).

Tuvastatud inimese kehaosade koordinaadid salvestatakse .csv faili. Iga kaadrit esitab .csv failis üks rida, mis koosneb kaadri id-st, ajatemplist, kehaosade koordinaatidest ja võimalikest markeritest.

Nimetatud .csv failist genereeritakse Bezier' kõver ja Bezier' kõvera parameetrid. Bezier' kõverat ennast kuskile maha ei salvestata. Bezier' kõverale arvatud Bezier' kõvera parameetrid salvestatakse jällegi uude .csv faili sarnasel kujul: iga kaader on esitatud ühe reaga, mis koosneb id-st, ajatemplist, ühest žesti klassifikatsioonist (kes žesti esitas) ja kolmest arvatud parameetrist.

Nimetatud žesti klassifikatsiooni kasutatakse selles töös märkimaks, kes žesti esitas. Rakendus on võimeline tuvastama nii žeste kui ka inimesi, kes žesti tegi. Kui tekib soov õpetada rakendust tuvastama vaid žeste, siis võib kõik õpetatavad žestid klassifitseerida ühte klassi: st. kasutada kogu aeg sama klassifikaatorit, näiteks null string või string „inimene“. Klassifikaatori välja .csv failis võib loomulikult kasutada mis iganes klassifikatsiooniks, mitte ainult inimese tuvastamiseks. Näiteks võib parema käe käelehvitusi klassifitseerida ka selle järgi, kas žesti teostas vasakukäeline või paremakäeline inimene (mitte selle järgi, kes žesti teostas). Selle töö raames ei ole kontrollitud, kas vasaku- ja paremakäelised on eristatavad nende käežestide järgi.

4.3 Näide

Selle töö raames on loodud rakendus Kinect Body Track. Rakenduse põhiülesanne on Kinect kaamerat kasutades tuvastada inimese poolt tehtud käežeste. Rakendusele saab ka õpetada žeste, mida see peab ära tundma: on võimalik salvestada inimese tehtud žeste ja nendest žestidest luua andmebaas, mida rakendus kasutab žestide tuvastamiseks.

4.3.1 Loodud rakendus Kinect Body Track

Rakenduse käivitades ja kaamera sisse lülitades: kui Kinect tuvastab inimese, kuvab rakendus inimese skeleti. Rakendus võib esitada ka Kinecti RGB kaamera pilti, kuid sellega kaasnevad jõudlusprobleemid ning vaikeväärtuslikult on RGB kaamera välja lülitatud.

Rakendus logib kaamera poolt tuvastatud inimese kohta infot: kehaosade asukoht. Sellest logist saab hiljem välja võtta uued žestid, mida soovitakse rakendusele õpetada. Teine põhjus logi pidamiseks on rakenduse debugimine arenduse käigus.

Logi pidamine ei ole vajalik žestituvastamisel ja kuna logipidamine on suur ressursikulu siis võiks žestide tuvastamisel reaajas logi pidamisest loobuda. Kuid kuna siia maani arendatud

rakenduses ei ole jõudlus suureks probleemiks osutunud, siis peetakse logi kogu rakenduse töö käigus, isegi siis kui see ei ole ilmtingimata vajalik.

Rakendus on võimeline uuesti esitama salvestatud (logitud) inimese liikumist ja žeste (menüüst: File -> Open, valida tuleb .csv kujul salvestatud koordinaatide fail). Kuna salvestatakse ainult info inimese asukoha kohta, siis taasmängitakse samuti ainult inimese keha liikumine tühjal taustal, st. RGB kaamera videot ei saa taasmängida, kuna seda lihtsalt ei salvestata.

Rakendus on võimeline numbriliselt markeerima salvestatud kaadreid 0-st valitud arvuni. Valitud arv valitakse Menüüst Data Processing -> Set Marker count. Nupuvajutusega „b“-kiirklahv suurendatakse markeri väärtust.

4.3.2 Rakendusele žestide õpetamine

Rakendus on võimeline õppima ettenäidatud žeste. Selleks tuleb rakendusele ette näidata konkreetset žesti mitu korda (nt. ~30 korda) ja salvestada need žestid rakenduse poolt kasutatavasse andmebaasi. Hiljem on rakendus võimeline tuvastama ettenäidatud žesti.

Tavalises režiimis töötades otsib rakendus pidevalt Kinecti poolt tuvastatud inimeste tehtud žeste. See pole vajalik žestide õpetamise ajal ning seda on võimalik ja soovitatav välja lülitada: Camera -> Toggle Gesture Recognition.

Kui inimene teeb kaamera ees mingeid käeliigutusi mida ta soovib rakendusele õpetada, siis on vaja ära märkida, millal käeliigutus algab ja lõppeb. Rakendus ei ole võimeline ise aru saama, et inimene tahab õpetada arvutile näiteks justnimelt lehvitus ja rakendus ei teagi, mis asi see lehvitus üldse on. Selleks on kasutajal olemas võimalus (nupuvajutusega „F5“) märku anda, millal žest algab ja lõppeb. Märku andes lisatakse marker viimasele salvestatud kaadriks. Hulga žeste ette näidates ja neid markeerides on tulemuseks üks markeeritud kaadritega fail. Rakendus saab lõigata faili tükkideks markeeringute järgi (menüüst: Data Processing -> Extract marked gestures, valida tuleb .csv kujul salvestatud koordinaatide fail, mida soovitakse tükeldada).

Žestide õpetamisel ei teosta inimene tihtilugu perfektselt vajalikke žeste järjest, vahetevahel tekib mingisugune hääming, paus või mingi segav asi võtab tähelepanu endale. Seetõttu ei pruugi lõigatud failid koosneda ainult soovitud žestidest, tegeletud on teiste samuti huvitavate tegevustega. Rakendusele ei ole sisse ehitatud mingit meetodit taoliste anomaaliatega

tegelemiseks. Kasutaja saab lasta kahtluse korral lõigatud failid ükshaaval rakendusel ette mängida, et kontrollida nende korrektsust ja käsitsi kustutada mitesobilikud failid, või kustutada manuaalselt konkreetsed read (kaadrid) .csv failis.

Kui hulk žeste (faile) on saadaval mida soovitakse rakendusele õpetada, siis saab need failid anda rakendusele ette, öeldes mis žestiga on tegu ja kes žesti teostas (menüüst: Data Processing -> Calculate bezier curve parameters., valida tuleb .csv kujul salvestatud koordinaatide fail või failid). Rakendus koostab žestidest Bezier' kõverad ja arvutab kõverate parameetrid – tegu on varem mainitud andmetöötlusprotsessiga. Arvutatud parameetrid salvestatakse uutesse csv failidesse, mida kasutaja saab kopeerida kasutatavasse andmebaasi.

Kui žestid on lisatud andmebaasi siis on rakendus võimeline neid ka reaajas inimese liigutustest otsima.

Bezier' kõvera parameetrite arvutamisel küsitakse kasutajalt, et kes žesti teostas, on eeldatud, et rakendust kasutatakse inimeste identifitseerimiseks. Inimeste klassifikatsioon toimub sisestatavate stringide võrdlemisel. Kui soovitakse inimeste klassifitseerimist vältida, võib välja tühjaks jätta. Võib ka vastupidiselt küsitule (nõutakse inimese nime) välja mõelda omi klasse, kuidas žeste identifitseerida. Näiteks saab jaotada parema- ja vasakukäelised kahte klassi, andes ühtedele nimeks „parem“ ja teistele „vasak“.

5. Tulemused

Uurimistöös, millest antud töö oli inspireeritud, kasutati analoogset Bezier' meetodit inimeste tuvastamiseks kõnnaku järgi. Käesolevas töös tehti katseid käežestide eristamiseks.

Esmalt tehti katsetusi kahe elementaarse käežesti tuvastamiseks: käega ringi tegemine ja käega lehvitamine. Nende žestide rakendusele õpetamises osales 7 inimest, ning kokku salvestati 214 käeringi ja 205 käelehvitust, mille põhjal hakkas rakendus tuvastama käežeste. Reaalajas osales tuvastuse kvaliteedi kontrollis 3 inimest, igaüks teostas 20 käežesti: 10 ringi ja 10 lehvitust.

	Ringi tuvastus	Ringi tuvastus %	Lehvituse tuvastus	Lehvituse tuvastus %
Inimene 1*	9/10	90%	9/10	90%
Inimene 2	8/10	80%	9/10	90%
Inimene 3*	9/10	90%	5/10	50%
Kokku	26/30	86,6%	23/30	76,6%

Tabel 1. Ringi ja lehvituse tuvastus tulemused.

Tärniga märgitud katsealused õpetasid rakendusele žeste. On näha et võrdlemisi suure andmebaasi järgi žestide tuvastamine toimub reaalajas hea tuvastustõenäosusega.

Teiseks simuleeriti operatsiooni, mille käigus kontrolliti kirurgiliigutuste tuvastuse kvaliteeti. Rakendusele õpetati 4 žesti ühe inimese poolt: kirurgi tööriista enda juurde toomine (*bring*), riista eemaldamine patsiendist (*pullout*) ja tööriista kõrvale panek (*putback*). Rakenduse õpetamisel salvestati žeste: *bring* – 41 žesti, *pullout* – 39, *putback* – 37. Tuvastuse kvaliteedi kontrollis osales 3 inimest, igaüks teostas 10 käežesti: *bring* - 10, *pullout* - 10, *putback* - 10.

	<i>Bring</i>	<i>Bring %</i>	<i>Pullout</i>	<i>Pullout %</i>	<i>Putback</i>	<i>Putback %</i>
Inimene 1*	5/10	50%	9/10	90%	9/10	90%
Inimene 2	5/10	50%	6/10	60%	9/10	90%
Inimene 3	4/10	40%	1/10	10%	9/10	90%
Kokku	14/30	46,7%	16/30	53,3%	27/30	90,0%

Tabel 2. Kirurgiliigutuste tuvastuse tulemused

Tärniga märgitud inimene osales žestide rakendusele õpetamisel. On näha et kahe žesti tuvastamine ebaõnnestus ja järjepidevalt õnnestus tuvastada vaid putback žesti. Nõrgemad tulemused žestide tuvastamisel on osaliselt põhjendatud vähese arvu žestide õpetamisega rakendusele: igat žesti ~40 korda (eelmises katses igat žesti ~200 korda). Teiseks osales rakenduse õpetamises ainult üks inimene, eelpool mainitud katses õpetas rakendusele žeste 7 inimest. See tähendab väiksemat mitmekesisust ühe žesti „tundmises“ rakenduse poolt. On näha (Tabel 2. Kirurgiliigutuste tuvastuse tulemused) et tärniga märgitud inimese pullout žesti tuvastati tunduvalt suurema tõenäosusega – rakendus oli ainult tema peal välja õpetatud.

6. Märkmeid lähtekoodist

GlobalVariables.cs failis on muutujaid mille väärtust võib muuta žestituvastuse peenhäälestamiseks. Alamgrupi „Buffer“ muutujate väärtuste abil saab parandada/halvendada žesti tuvastuse täpsust, samas vähendades/parandades rakenduse jõudlust. Muutuja MinGestureFramesCount määrab minimaalse žesti kestvuse, liiga väike kestvus vähendab rakenduse jõudlust ja võib tekitada rohkem *false-positive* žesti tuvastamist.

Muutujad GestureMaxDistance ja NrOfPointsOnBezierCurveCompared on omavahel seotud. Ühe muutmisel peab tõenäoliselt muutma ka teist. GestureMaxDistance kasutatakse žestide vahelise maksimaalse kauguse määramiseks, mille korral neid ikkagi loetakse sarnasteks žestideks (5.1.2 Žestide tuvastamine). NrOfPointsOnBezierCurveCompared kasutatakse kahe žesti vahelise kauguse arvutamisel (4.1.4.1 Bezier' kõverate vahelise kauguse arvutamise implementatsioon). Mida suuremadimensioonilisem on eukleidiline ruum, seda kaugemal asuvad žestid, ning muutuma peab ka GestureMaxDistance väärtus.

Muutuja BezierCurveOutputSegmentCount määrab, mitme punktiga on defineeritud Bezier' kõver. On võimalik suurendada Bezier' kõvera arvutamise täpsust jõudluse arvelt. BodypartsForBezierCurve on hulk kehaosi, mille põhjal moodustatakse Bezier' kõver. Erinevate kehaosade jälgimiseks tuleb muuta seda muutujat.

nrOfClosestGesturesComparedKNN määrab K-lähima naabri algoritmis K väärtuse.

MinDistanceTraveled määrab, kui väikest keha liikumist loetakse žestiks. Paigalseisvat ja väikest liikumist otsustati žestina mitte tuvastada.

Rakenduses püütakse salvestada võimalikult palju informatsiooni. Näiteks: kuigi Bezier' kõver ei kasuta arvutamiseks Z-koordinaati (4.2 Implementatsioon) salvestatakse Z-koordinaat .csv failidesse koos X ja Y koordinaadiga. Teiseks näiteks salvestatakse kehaosade orientatsiooni, kuigi seda ei kasutata hetkel mitte kuskil.

Lähtekoodis on kasutatud koodilõike, meetodeid ja klasse mis ei ole kirjutatud töö autori poolt, vaid on vabalt saadaval internetis. Nimetatud programmikoodi autor on lähtekoodis

kommentaari „author:“ järel, lisaks on Kasutatud kirjanduse all loetletud kasutatud koodi leiukohad: [18], [19], [20], [21], [22].

7. Arutelu

Selle töö käigus arendatud rakendus on võimeline jälgima mis tahes Kinecti poolt tuvastatud kehaosa. Rakendust saab kerge ümberkonfigureerimisega kasutada inimese mistahes liigutuste tuvastamiseks (ja inimese identifitseerimiseks mis tahes kehaosade liikumise järgi). Kuid uusi kehaosi jälgides, tuleb terve rakendus ümber õpetada, mis on üsna ajamahukas.

Võib tekkida tahe püüda sama žesti ära tunda jälgides erinevaid kehaosi. Selles töös kasutatakse parema käe žestide äratundmisel nelja punkti inimese kehal. Jälgides rohkemaid punkte (näiteks kuidas inimene oma tervet keha žesti ajal hoiab), võivad žestituvastamise katsete tulemused paraneda (või halveneda). Ent, uusi punkte valides, muutub loomulikult ka moodustatav Bezier' kõver ja sellele arvutatavad parameetrid (meeldetuletuseks: nimetatud parameetrid moodustavad kogu masina teadvuse/andmebaasi). Seega tuleks terve rakendus jällegi ümber õpetada. Samade žestide uuesti õppimisel (uute punktide järgi) saab õnneks kasutada juba varem teostatud žestide salvestusi.

Uurimistöös ei kasutatud ei masinõppel ega žestituvastusel reaalsel operatsioonikeskkonda. See tähendab, et žeste, mida kirurg teeb operatsiooni käigus simuleeriti. Lisaksei teostanud žeste professionaalne kirurg, vaid inimene, kes teadis, kuidas opereerimisliigutused välja näevad. Et saada tõesemaid tulemusi, peaks masinõppe käigus žeste esitama tegevkirurg reaalse operatsiooni käigus. Soovitavalt peaks masinõppest osa võtma mitu kirurgi.

Kokkuvõte

Töö eesmärgiks oli uurida ja välja töötada meetod kirurgi käe žestide tuvastamiseks laparoskoopilise operatsiooni käigus. Välja töötatud lahenduse efektiivsust kontrolliti katsetes, kus tuvastati käežeste reaalselt: ühes katses üritati identifitseerida kahte elementaarset käežesti (käega ringi tegemine ja käega lehvitamine), teises katses tuvastati kirurgi poolt tehtavaid käeliigutusi simuleeritud operatsiooni käigus.

Katsete tulemused näitavad, et valitud meetod (Bezier' kõvera ja K-NN masinõppe algoritmi kombinatsioon) on võimeline käežestide tuvastamiseks. Sellele lisaks valmis töö käigus rakendus, mis on võimeline reaalselt žestituvastuseks, rakendusele on võimalik õpetada žeste, mida see on võimeline hiljem tuvastama. Rakendus kirjutati C# programmeerimiskeeles. Rakendus sobib edasiseks arendustööks, või osaks teistest rakendustest, et edukalt tuvastada kirurgi käeliigutusi laparoskoopilise operatsiooni käigus.

Lisaks käežestide tuvastamisele selgus, et Bezier' kõvera abil on võimalik tuvastada inimest, jälgides inimese käežeste.

Olulisemad järeldused:

- Käežestile arvatud Bezier' kõvera parameetrid (x-dispersioon, y-dispersioon ja keskmine) omavad piisavat informatsiooni et teha vahet eri žestidel.
- Välja töötatud meetod kasutades Bezier' kõverat ja K-NN algoritmi sobib käežestide reaalselt tuvastamiseks.

Summary

The goal of this thesis was to research and develop a method to recognise hand gestures of a surgeon during laparoscopic surgery. The effectiveness of developed solution was confirmed in tests where hand gestures were recognised in real time: in one test two simple gestures were identified (circular motion with hand and hand wave), in second test hand motions of a surgeon were identified in a simulated surgery.

The results of the experiments show that the chosen method (combination of Bezier' curve and K-NN algorithm) is capable of recognising gestures. In addition an application was developed which is capable of real time gesture recognition, it is possible to teach gestures to the application, and it will later try to recognise these gestures. Application was developed in C# programming language. The application is fitting for further development to successfully recognise hand motions of a surgeon during laparoscopic surgery.

In addition to recognising gestures when using Bezier' curve it is possible to identify persons by their hand gestures.

Key conclusions:

- Bezier' curve parameters (x-variance, y-variance, average) that are calculated based on gestures carry enough information to differentiate gestures.
- Developed method using Bezier' curve and K-NN algorithm is capable of real time gesture recognition.

Kasutatud kirjandus

- [1] P. Mishra and S. Ezra, "Human Gait Recognition Using Bezier Curves," *International Journal on Computer Science and Engineering*, 2011.
- [2] X. Zhao, A. M. Naguib and S. Lee, "Kinect Based Calling Gesture Recognition for Taking Order Service of Elderly Care Robot," in *IEEE*, Edinburgh, 2014.
- [3] G. Baron, P. Czekalski, D. Malicki and K. Tokarz, "Remote Control of the Artificial Arm Model Using 3D Hand Tracking," in *IEEE*, Opole-Zawiercie, 2013.
- [4] M. M. F. M. Fareed, Q. I. Akram, S. B. A. Anees and A. H. Fakih, "Gesture based wireless single-armed Robot in Cartesian 3D space using Kinect," in *IEEE*, Gwalior, 2015.
- [5] M. C. Shin, L. V. Tsap and D. B. Goldgof, "Towards Perceptual Interface for Visualization Navigation of Large Data Sets," in *IEEE*, Madison, Wisconsin, USA, 2003.
- [6] O. Rashid and A. Al-Hamadi, "Utilizing the Bezier descriptors for hand gesture recognition," in *IEEE*, Quebec City, 2015.
- [7] H. Cook, Q. V. Nguyen, S. Simoff, T. Trescak and D. Preston, "A Close-Range Gesture Interaction with Kinect," in *IEEE*, Hobart, TAS, 2015.
- [8] S. Wu, F. Jiang and D. Zhao, "Hand gesture recognition based on skeleton of point clouds," in *IEEE*, Nanjing, 2012.
- [9] T. Hachaj, M. R. Ogiela and K. Koptyra, "Effectiveness Comparison of Kinect and Kinect 2 for Recognition of Oyama Karate Techniques," in *IEEE*, Taipei, 2015.
- [10] B. Dikovski, G. Madjarov and D. Gjorgjevikj, "Evaluation of different feature sets for gait recognition using skeletal data from Kinect," in *IEEE*, Opatija, 2014.
- [11] S. Saha, S. Ghosh, A. Konar and A. K. Nagar, "Gesture Recognition from Indian Classical Dance Using Kinect Sensor," in *IEEE*, Madrid, 2013.
- [12] "adobe.com," [Online]. Available: <https://helpx.adobe.com/photoshop/using/drawing-pen-tools.html>. [Accessed 1 4 2016].
- [13] Ü. Kaasik, *Matemaatika leksikon*, Tallinn: Eesti Entsüklopeediakirjastus, 1992.
- [14] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Second Edition, USA: Springer, 2009.
- [15] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, London: The MIT Press, 2012.
- [16] C. C. Aggarwal, *Data Mining: The Textbook*, New York: Springer, 2015.
- [17] "www.victoriakirst.com," [Online]. Available: <http://www.victoriakirst.com/beziertool/>. [Accessed 1 4 2016].
- [18] "pterneas.com," [Online]. Available: <http://pterneas.com/2014/03/13/kinect-for-windows-version-2-body-tracking/>. [Accessed 20 5 2016].
- [19] "outcoldman.com," [Online]. Available: <https://www.outcoldman.com/en/archive/2010/09/11/register-hotkey-in-system-for-wpf->

application/. [Accessed 20 5 2016].

- [20] "sinairv.wordpress.com," [Online]. Available: <https://sinairv.wordpress.com/2011/05/15/code-snippet-mean-variance-and-standard-deviation-methods/>. [Accessed 20 5 2016].
- [21] "stackoverflow.com," [Online]. Available: <http://stackoverflow.com/a/8914725>. [Accessed 20 5 2016].
- [22] "stackoverflow.com," [Online]. Available: <http://stackoverflow.com/a/13948059>. [Accessed 20 5 2016].

Lisa

Tööle on kaasa lisatud:

- Rakendus
- Rakenduse lähtekood
- Näiteeksemplar rakenduse andmebaasist
- Hulk salvestatud žeste

Rakenduse ühte põhiülesannet, žestide tuvastamist, saab täita vaid andmebaasi olemasoluga. Rakenduse teise kasutusviisi (žestide õpetamine) abil saab ka tekitada vajamineva andmebaasi, kuid see on võrdlemisi ajamahukas ning rakenduse esmase töö kontrollimiseks on hea kasutada juba olemasolevat andmebaasi.

Meeldetuletuseks: rakenduse andmebaas koosneb žeste iseloomustavatest parameetritest, nende parameetrite järgi ei ole võimalik tuletada keha liikumist žesti jooksul. Lisatud on ka andmebaasi moodustavate žestide salvestused (kehaosade täpsed koordinaadid), kui peaks tekkima huvi teada missuguseid žeste on rakendusele täpselt õpetatud. Rakendus on võimeline salvestatud žeste taasesitama, kui need on salvestatud koordinaatidena (mitte parameetritena).

Salvestatud žeste saab ka kasutada andmebaasi genereerimiseks (rakenduse õpetamiseks). Tööle kaasa lisatud andmebaas on genereeritud kaasa lisatud žestidest.