

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

Skirmantas Sinkus

KINECT ĮRENGINIUI SKIRTŲ GESTŲ ATPAŽINIMO  
ALGORITMŲ TYRIMAS

Magistro darbas

Darbo vadovas:

dr. T. Blažauskas

Kaunas, 2014

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

Skirmantas Sinkus

KINECT ĮRENGINIUI SKIRTŲ GESTŲ ATPAŽINIMO  
ALGORITMŲ TYRIMAS

Magistro darbas

Darbo vadovas:

dr. T. Blažauskas

Recenzentas:

Kaunas, 2014

## AUTENTIŠKUMO PATVIRTINIMAS

Patvirtinu, kad įteikiamas baigiamasis darbas „Klaidų paieškos mobilių įrenginių programinėje įrangoje metodo, taikant programos vykdymo laikines charakteristikas, kūrimas ir tyrimas“:

1. Autoriaus atliktas savarankiškai, jame nėra pateikta kitų autorių medžiagos kaip savos, nenurodant tikrojo šaltinio.
2. Nebuvo to paties autoriaus pristatytas ir gintas kitoje mokymo įstaigoje Lietuvoje ar užsienyje.
3. Nepateikia nuorodų į kitus darbus, jeigu jų medžiaga nėra naudota darbe.
4. Pateikia visą naudotos literatūros sąrašą.

Skirmantas Sinkus

---

(studento vardas, pavardė)

(data)

(parašas)

# **RESEARCH OF GESTURE RECOGNITION ALGORITHMS DEDIDACATED FOR KINECT DEVICE**

## **SUMMARY**

Microsoft Kinect device was released in 2010. It was designed for Microsoft Xbox 360 gaming console, later on in 2012 was presented Kinect device for Windows personal computer. So this device is new and current.

Many games has been created for Microsoft Kinect device, but this device could be used not only in games, one of the areas where we can use it its sport, specific training, which can be performed at home.

At this moment in world are huge variety of games, software, training programs which allows user to control training course by following a person properly perform training provided movements. Since in Lithuania similar software is not available, so it is necessary to create software that would allow Lithuania coaches create training focused on the use of this device.

The main goal of this work is to perform research of the Kinect device gesture recognition algorithms to study exactly how they can recognize gestures or gesture. It will focus on this issue mainly, but does not address the criteria for recognition as the time and difficulty of realization.

In this paper, a program that recognizes movements and gestures are using the Golden section search algorithm. Algorhithm compares the two models or templates, and if it can not find a match, this is the first template slightly rotated and comparison process is started again, also a certain variable helping, we can modify the algorithm accuracy. Also for comparison we can use Hidden Markov models algorhithm received recognition results, which are obtained by other people. Or use system which have implemented Dynamic Time Warping algorithm.

# Turinys:

1.	ĮVADAS .....	9
1.1.	Dokumento paskirtis.....	9
1.2.	Santrauka.....	10
2.	ANALITINĖ DALIS .....	11
2.1.	Temos aktualumas bei kūrimo tikslas.....	11
2.2.	Pažintis su įrenginiu.....	11
2.3.	Natūrali sąveika .....	13
2.3.1.	Gestai .....	14
2.3.2.	Gestų atpažinimas .....	14
2.3.3.	Kūno sekimas.....	14
2.4.	Esami sprendimai pasaulyje.....	15
2.5.	Gestų atpažinimo algoritmų apžvalga.....	15
2.5.1.	Golden Section Search algoritmas .....	16
2.5.2.	Hidden Markov Model algoritmas.....	16
2.5.3.	Dynamic Time Warping algoritmas.....	16
3.	ŽMOGAUS SKELETO JUDESŲ ĮRAŠYMO IR ATKŪRIMO PROJEKTAS .....	17
3.1.	Pažintis su sistema .....	17
3.1.1.	Sistemos funkcijos .....	17
3.1.2.	Sistemos apribojimai.....	17
3.2.	Projekto paskirtis .....	17
3.3.	Veiklos kontekstas .....	18
3.4.	Sistemos funkciniai reikalavimai .....	18
3.5.	Sistemos nefunkciniai reikalavimai .....	18
3.6.	Sistemos architektūra .....	19
3.6.1.	Paskirtis.....	19
3.6.2.	Architektūros pateikimas .....	19
3.6.3.	Architektūros tikslai ir apribojimai .....	19
3.6.4.	Panaudojimo atvejų vaizdas.....	20
3.6.5.	Statinis sistemos vaizdas.....	21
3.6.5.1.	Detect Moves paketas .....	21
3.6.5.2.	Record Moves paketas .....	22
3.6.5.3.	Verify Moves paketas .....	22
3.6.6.	Klasių diagrama .....	23
3.6.7.	Būsenų diagramos.....	24
3.6.8.	Veiklos diagramos .....	25
3.6.9.	Išdėstymo vaizdas .....	27
3.6.10.	Testavimas .....	28
3.6.10.1.	Įvadas .....	28
3.6.10.2.	Testavimo tikslai ir objektai.....	28
3.6.10.3.	Vienetų testavimas .....	28
3.6.10.4.	Vienetų testavimo rezultatai .....	29
3.6.10.5.	Integracinis testavimas.....	29
3.6.10.6.	Priėmimo testavimas.....	29
4.	KINECT ĮRENGINIUI SKIRTŲ GESTŲ ATPAŽINIMO ALGORITMŲ TYRIMAS.....	29
4.1.	Įvadas .....	29
4.2.	Hidden Markov Model algoritmas.....	30
4.3.	Dynamic Time Warping algoritmas.....	33
4.4.	Golden Section Search algoritmas .....	34
5.	EKSPERIMENTINĖ DALIS.....	36
5.1.	Ižanga.....	36

5.2.	Tyrimo metodika.....	36
5.3.	Hidden Markov Models algoritmo eksperimentinis tyrimas .....	36
5.3.1.	Įvadas .....	36
5.3.2.	Eksperimentinė dalis .....	36
5.3.3.	Apibendrinimas.....	37
5.4.	Golden section Search algoritmo eksperimentinis tyrimas.....	38
5.4.1.	Įvadas .....	38
5.4.2.	Eksperimentinė dalis.....	39
5.4.3.	Apibendrinimas.....	41
5.5.	Dynamic Time warping algoritmo eksperimentinė dalis.....	41
5.5.1.	Įvadas .....	41
5.5.2.	Eksperimentinė dalis .....	42
5.5.3.	Apibendrinimas.....	44
6.	IŠVADOS .....	45
7.	LITERATŪRA .....	46
8.	TERMINŲ IR SUTRUMPINIMŲ ŽODYNAS .....	48

## Lentelių sąrašas:

1 lentelė. Kinect jutiklio specifikacijos.....	13
2 lentelė Apskritimo gesto atpažinimas .....	37
3. lentelė Apskritimo gesto atpažinimas kai ribinė reikšmė 0.6 .....	39
4. lentelė Apskritimo gesto atpažinimas kai ribinė reikšmė 0.7 .....	39
5. lentelė Apskritimo gesto atpažinimas kai ribinė reikšmė 0.8 .....	39
6. lentelė Apskritimo gesto atpažinimas kai ribinė reikšmė 0.9 .....	40
7. lentelė Apskritimo gesto atpažinimas kai ribinė reikšmė 0.99 .....	40
8. lentelė Dešinės rankos mostas į kairę. ....	42
9. lentelė Pamojavimas dešinė ranka. ....	43
10. lentelė Abiejų rankų gesto pritraukimas. ....	43
11. lentelė Kairės rankos nuleidimas į apačią. ....	43

## Paveikslų sąrašas:

1 pav. Judesių žaidimų valdiklių palyginimas. Nuotraukoje matomi Kinect, Wii Remote ir PlayStation Move.....	11
2 pav. Išardytas Kinect.....	12
3 pav. Kinect jutikliai. Šiame pav. pažymėti visus jutikliai, kurie yra Kinect, kaip 3D gylio kamera (IR projektorius and IR kamera), RGB kamera, mikrafonų masyvas ir motorizuotas tentas. ....	12
4 pav. Žmogaus ir kompiuterio sąveikos evoliucija schema. Šis pav. sudaro žmogaus ir kompiuterio sąveikos paradigmų raida bėgant metams.....	13
4 pav. Konteksto diagrama .....	18
5 pav. Panaudos atveju diagrama.....	20
6 pav. sistemos išskaidymas į paketus .....	21
7 pav. Trenerio klasių diagrama.....	23
8 pav. Vartotojo/atleto klasių diagrama.....	23
9 pav. Judesio įrašymo būsenų diagrama.....	24
11 pav. Meniu punkto pasirinkimo veiklos diagrama.....	25
12 pav. Treniruotės veiklos diagrama.....	26
13 pav. Judesių įrašymo veiklos diagrama .....	27
14 pav. Sistemos išdestymo vaizdas .....	28
15 pav. Testavimo rezultatų pavyzdys.....	29
16 pav. Idealus ir Realus apskritimo gestas laike. ....	30
17 pav. HMM modelio tipai: (a) kairės – dešinės modelis (b) – ergodiškas modelis.....	31
18 pav. Supaprastine struktūra ribinio modelio. Punktyrinės rodyklės yra nuliniai parėjimai.....	32
19 pav. Kinect sąnariai.....	34
20 pav. Golden section search su pradiniais režiais (1,2,3) tampa (4,2,3) (4,2,5).....	35
21 pav. Programos langas .....	38
22 pav. Atpažinimo procetas priklausantis nuo ribinių reikšmių .....	41
23 pav. Sistemos vaizdas su Dynamic Time Warping.....	42



# 1. ĮVADAS

2010 m. Lapkričio mėnesį Microsoft išleido, tai ką jie pavadino „gimimas naujos kartos pramogų namams“ [1]. Microsoft nežinojo išleidimo metu, kad Kinect yra ne tik „įprastas“ žaidimų prietaisas, tai yra kur kas daug daugiau. Tai nauja technologija ir nauja sąveikos tarp žmogaus ir kompiuterio paradigma, Kinect paveldėjo išorinių periferinių įrenginių, kurie šiuo metu naudojami, pavyzdžiui nuotolinio valdymo pulteliai, klaviatūros arba pelės, savybes. Tai kelia didelį susidomėjimą visame informacijų technologijų pasaulyje, iš tiesų internete greitai pasirodė daug universitetų projektų, kurie pabrėžė tokios rūšies technologijų potencialą. Dauguma šių projektų buvo paremti, karkaso kūrėjos, šios naujos paradigmos sąveikos pradininkės ir įkūrėjos Izraelio kompanijos pasivadinusios „PrimeSense“, kuri buvo pirma išleidusi stabilų karkasą sąveikauti ir kurti Kinect pagrindu paremtas programas. Gana keista, bet pats Microsoft atkreipė dėmesį į šį reiškinį ir išleido nekomercinį Kinect programinės įrangos kūrimo įrankį skirtą Windows platformai 2011 m. birželio 16 dieną.

Kinect įrenginys gali būti apibūdinamas kaip natūralios sąveikos įrenginys, kuris veikia natūralios vartotojo sąsajos kontekste, taip pat naujai atsirandančios metafizinės paradigmos žmogaus ir kompiuterio sąveika. Natūrali vartotoją sąveika remiasi koncepcija, kad ši sąveikauja su įrenginiais ir yra pagrįsta žmogaus pojūčiais [2]. Pagrindinis tikslas natūralios sąveikos paradigmos yra transformuoti paveldimas sąsajas ir įtaisus, kurie buvo naudojami pastaruosius metus technologijų raidoje į labiau intuityvias ir natūralias sąveikas be poreikio dėvėti tam tikrus prietaisus ar išmokti sudėtingus nurodymus kaip juos naudoti. Kinect gali puikiai atlikti šį vaidmenį pertvarkant šį procesą.

Kalbant apie natūralią vartotojo sąsaja, svarbu suprasti, kas tai dažniausiai Kinect įrenginys yra skirtas garso ar judesių atpažinimui. Norint sukurti garso ar judesių atpažinimą svarbu pasirinkti tinkamas priemones, kurios leistų tai padaryti lengviausiu keliu. Yra keletas algoritmų kurių pakalbam mes galime tai padaryti. Kadangi darbo tema yra apie judesių atpažinimą, taigi mes susikoncentruosime tik ties judesių atpažinimu, bei judesių atpažinimo algoritmais.

Darbo tikslai:

- Išanalizuoti literatūrą, kuri susijusi su Kinect įrenginiu, taip pat iš analizuoti jau esamus sprendimus, atlikti naudojamų algoritmų gestų atpažinimui analizę.
- Suprojektuoti ir sukurti sistemą kuri gebėtų įrašyti žmogaus skeleto judesius, gebėtų juos atpažinti ir duoti atgalinį ryšį.
- Atlikti eksperimentinius tyrimus siekiant išsiaiškinti tiriamų algoritmų gestų atpažinimo tikslumą.

Norint atlikti judesių atpažinimą susiduriama su keletą problemų tokių kaip:

- judesių atpažinimo tikslumas.
- judesių atpažinimo greitis.
- judesių atpažinimo kiekis.

Šiame darbe mes tirsime tik vieną charakteristiką – tai judesių atpažinimo tikslumą. Taipogi tirsime algoritmus skirtus atpažinti gestus.

Kadangi išspręsti šias visas problemas pareikalautų daug laiko, taigi šiame darbe mes spręsimė judesių atpažinimo tikslumą.

## 1.1. Dokumento paskirtis.

Dokumentu paskirtis yra pateikti sukurtų ir tyrinėjamų Kinect įrenginiui skirtų gestų atpažinimo algoritmų tyrimo rezultatus. Ištirta kaip tiksliai algoritmai atpažįsta gestus. Taip pat pateiktas sukurtas projektas. Jau egzistuojančių panašių sistemų analizė. Taip pat pateikta teorinis, bei praktinis atpažinimo algoritmų palyginimo rezultatai.

## 1.2. Santrauka

Microsoft Kinect įrenginys išleistas tik 2010 metais. Jis buvo skirtas Microsoft Xbox 360 vaizdo žaidimų konsolėi, vėliau 2012 metais buvo pristatytas Kinect ir Windows personaliniams kompiuteriams. Taigi tai palyginus naujas įrenginys ir aktualus šiai dienai.

Daugiausiai yra sukurta kompiuterinių žaidimų, kurie naudoja Microsoft Kinect įrenginį, bet šį įrenginį galima panaudoti daug plačiau ne tik žaidimuose, viena iš sričių tai sportas, konkrečiau treniruotės, kurias būtų galima atlikti namuose.

Šiuo metu pasaulyje yra programinės įrangos, žaidimų, sportavimo programų, kuri leidžia kontroliuoti treniruočių eigą sekdamas ar žmogus teisingai atlieka treniruotėms numatytus judesius. Kadangi Lietuvoje panašios programinės įrangos nėra, taigi reikia sukurti įrangą, kuri leistų Lietuvos treneriams kurti treniruotes orientuotas į šio įrenginio panaudojimą.

Šio darbo pagrindinis tikslas yra atlikti Kinect įrenginiui skirtų gestų atpažinimo algoritmų tyrimą, kaip tiksliai jie gali atpažinti gestus ar gestą. Pagrindinis dėmesys skiriamas šiai problemai, taip pat keliami, bet netyrinėjami kriterijai kaip atpažinimo laikas, bei realizacijos sunkumas.

Šiame darbe sukurta programa, judesius bei gestus atpažįsta naudojant Golden Section Search algoritmą. Algoritmas palygina du modelius ar šablonus, ir jei neranda atitikmens, tai pirmasis šablonas šiek tiek pasukamas ir lyginimo procesas paleidžiamas vėl, taipogi tam tikro kintamojo dėka galime keisti algoritmo tikslumą. Taipogi palyginimui galėsime panaudoti Hidden Markov Models algoritmo gautus atpažinimo rezultatus, kurie yra gauti kitų asmenų. Taip pat galima palyginti ir su sistema kurioje realizuotas Dynamic Time Warping algoritmas.

## 2. ANALITINĖ DALIS

### 2.1. Temos aktualumas bei kūrimo tikslas.

Kaip minėta anksčiau pats prietaisas ir kūrimo karkasas nėra labai seni, tai gana nauja sritis, kur dar vyksta tobulinimai, kadangi Kinect jau patapo neatsiejama Xbox žaidimų konsolės dalis, tai ši technologija gyvuos. Anksčiau norint valdyti programas, reikėdavo įvesties įrenginių kaip pelė ar klaviatūra, dabar tai įmanoma daryti judesiais ar balso komandomis.

Žvelgiant į pasaulinę rinką panašių produktų yra jau sukurta, mūsų darbo tikslas buvo patyrinti kaip vyksta žmogaus kūno judesių atpažinimas. Taipogi nebuvo girdėti, apie didelį kiekį tokių projektų Lietuvoje, galbūt tai kūrimas ir tyrinėjimas universitetų viduje. Atlikti gestų atpažinimą nėra lengva užduotis, nes yra nevienareikšmiška problema, arba, tiksliau, ne taip lengva sukurti veiksmingus euristinius ar mašinių mokymosi algoritmus, kurie gali suprasti, kas yra konkretus gestas. Taigi ir mūsų darbo tikslas yra patyrinti teoriškai ar praktiškai gestų atpažinimo algoritmus.

### 2.2. Pažintis su įrenginiu

Kinect judesių įvesties įrenginys, sukurtas ir parduodamas kompanijos "Microsoft" kontroliuoti ir bendrauti su "Xbox 360" žaidimų konsole be fizinio valdymo. Kinect anksčiau žinomas kaip „Projektas Natal“ [1] įėjo į tą patį rinkos sektorių jau užimtą Wii Remote parduodama Nintendo ir PlayStation Move kartu su PlayStation Eye judesių valdiklis pagamintas Sony sąveikauti su namų konsolėmis Wii ir PlayStation 3. Shane Kim, korporacijos viceprezidentas atsakingas už strategiją ir verslo plėtrą Microsoft žaidimų padalinyje apibrėžia Kinect savo interviu [1] kaip „gimimas naujos kartos pramogų namams“.



**1 pav.** Judesių žaidimų valdiklių palyginimas. Nuotraukoje matomi Kinect, Wii Remote ir PlayStation Move.

Kaip jūs galite pamatyti 2 pav. Kinect yra ne vienos dalies įrenginys, jį sudaro daug skirtingų komponentų ir technologijų pasiūlyti vartotojui naują pramogų potyrį, kuri apima įvairius pojūčius.



**2 pav.** Išardytas Kinect.

Kinect yra pritaikytas pastatymui po vaizdo ekranu, ir iš esmės yra suformuotas iš horizontalios juostos jutiklių, prijungtų prie mažos bazės su motorizuota ašimi. Motorizuota bazė leidžia keisti vertikalią diapazoną į viršų arba apačią  $\pm 28^\circ$ , kaip pranešė [3]. Paveikslėlis rodo tris pagrindinius jutiklius Kinect kitaip tariant „Kinect akys“: dvi kameros ir infraraudonųjų spindulių (IR) projektorius. Pirmasis komponentas kairėje yra projektorius, kaip parodyta paveiksle, centriniai komponentai yra spalvos, papildomo metalo oksido-puslaidininkio (CMOS), paprasta raudona, žalia ir mėlyna (RGB), kamera su rezoliucija 640x480 32-bit, 30 kadru / sek. ir galiausiai dešinėje turime infraraudonųjų spindulių, papildomo metalo oksido-puslaidininkio arba infraraudonųjų spindulių imtuvas Su 320x240 16-bitų gylio skiriamoji geba, 30 kadru / sek. Prietaise taip pat yra įrengtas mikrofonas, kuris leidžia Kinect priimti ir atpažinti garso komandas. Mikrofonas leidžia lokalizuoti akustinį šaltinį ir slopinti aplinkos triukšmą. Keturi mikrofonai yra išdėstyti linija, trys iš jų kairėje pusėje, o kitas dešinėje, visi jie įdėti po įrenginiu.

Žemiau mes apibūdinsim specifikaciją [3]: jutiklis turi žiūrėjimo kampo lauką  $57^\circ$  horizontaliai ir  $43^\circ$  vertikalčiai, o mechaninės pavaros į bazę, kaip minėta anksčiau, gali judinant jutiklį  $27^\circ$  aukštyn arba žemyn.



**3 pav.** Kinect jutikliai. Šiame pav. pažymėti visus jutikliai, kurie yra Kinect, kaip 3D gylio kamera (IR projektorius ir IR kamera), RGB kamera, mikrofonų masyvas ir motorizuotas tentas.

**1 lentelė.** Kinect jutiklio specifikacijos

Jutiklio daiktas	Specifikacijos intervalas
Matymo kampas	43 ° vertikaliai 57 ° horizontaliai matymo laukas.
Mechanizuoto tento intervalas (vertikalus)	±28°
Kadrų dažnis (gylis ir spalvų srautas)	30 kadrų per sekundę (FPS)
Rezoliucija, gylis srautas	QVGA(320×240)
Rezoliucija, spalva srautas	VGA(640×480)
Audio formatas	16 kHz, 16-bitų mono pulsas kodo moduliacija
Garso įvesties charakteristikos	Keturių mikrofonų matrica su 24-bitų analoginio signalo konverteriu (ADC) ir Kinect signalų apdorojimas, pvz. akustinio aido panaikinimo ir triukšmo slopinimas

Jeigu norime suteikti Kinect įrenginiui tikslų apibrėžimą, mes galime pasakyti, kad tai infraraudonųjų spindulių struktūrizuotas šviesos 3D skeneris. PrimeSense schema parodyta 4 pav. paaiškina, kaip veikia jų standartinė platforma. IR siųstuvas siunčia struktūrizuotus infraraudonųjų šviesos spindulius nešančius koduotą informaciją įvairiuose šviesos formų modeliuose. Kai IR jutiklis aptinka gražintą signalą, jis ieško deformacijų užkoduotame modelyje ir sąryšio prieš nurodant modelį žinomo gylio. Iš šių pokyčių įrenginys atlieka skaičiavimus nustatyti gylio informaciją ir įvairias charakteristikas objektų ir žmonių kurie yra Kinect matymo lauke [4][5].

### 2.3. Natūrali sąveika

Natūrali sąveika remiasi koncepcija, pagal kurią sąveika su prietaisu paremta žmogiškaisiais pojūčiais [2]. Šios naujos paradigmos suteikiamos išoriniams periferiniams įrenginiams, kurie šiuo metu naudojami, pavyzdžiui, nuotolinio valdymo pulteliai, klaviatūros arba pelės. Kinect gali būti apibrėžtas kaip natūralios sąveikos įtaisas, kuris veikia iš natūralioje vartotojo sąsajoje, naujai atsirandančiai metafizinei paradigmai žmogaus ir kompiuterio sąveikai. 2006 metais krikščionių Moore įkūrė mokslinių tyrimų bendruomenę su mokymosi ir dalijimosi naujų žmogaus ir kompiuterio sąveikos metodais ir sąvokomis tikslas apibrėžti žmogaus ir kompiuterio sąveikos technologijas. Natūralios sąveikos paradigma gali būti nustatyta, konkrečiu atveju kaip žmogaus ir kompiuterio sąveika, nes šioje mokslinių tyrimų srityje apima daug kitų įvairių paradigimų, modelių ir technologijų. Kaip apibūdinama [6], įprastinė vartotojo sąsaja yra tolimesnis žingsnis sąsajų raidos etapui su tikslu sukurti labiau organines sąsajas, kurios yra pagrįstos tokiais reiškiniais kaip lietimasis, matymas, kalbos ir daugiau tradicinių žmogaus sąveikos paradigimų ir svarbiausia kūrybiškumas. natūralios sąveikos paradigmos pagrindinis tikslas yra transformuoti senąsias sąsajas ir įtaisyti, naudojamus per pastaruosius metus vykdamas technologijų raida kurios dėka būtų paprastesnė ir natūralesnė sąveika be prietaisų dėvėjimo arba mokintis sudėtingus nurodymus kaip juos naudoti.



**4 pav.** Žmogaus ir kompiuterio sąveikos evoliucija schema. Šis pav. sudaro žmogaus ir kompiuterio sąveikos paradigimų raida bėgant metams

Kaip mes minėjome anksčiau daugiau dėmesio skirsime natūrali sąveikos tam tikro tipo rinkiniams ir natūralios vartotojo sąveikos naujiems metodams:

**Kūno judėjimo stebėjimą** 3D duomenų modelių analizavimą ir interpretaciją leidžiantį viso kūno judesių sekimą ir skeleto atpažinimą rasti ir atpažinti atskirus sąnarius ir įvairias kūno dalis;

**Gestų atpažinimas**, kurie iš anksto nustatyti gestų modeliai yra atpažįstami ir naudojami tam, kaip naujas būdas kontroliuoti prietaisus;

### 2.3.1. Gestai

Šioje disertacijoje [7] atžvelgiant į natūralią sąveiką apibrėžiame gestai cituodamas Turk kaip išraiškingas, prasmingas kūno judėjimas su ketinimu perduoti informaciją ar sąveikauti su jį supančia aplinka. [8] apibrėžta taksonomija atpažinti įvairių galimų gestų tipus: gestikuliacija, manipuliacijos, ženklų kalbą. Gestikuliavimas yra viena iš natūraliausių gestų formų ir tai apibūdina gestų pobūdį, kuris yra lydimas šnekamosios kalbos. Šios rūšies gestai nereikalauja, kad vartotojas atliktų tam tikras pozas arba mokytis kokių nors gestų, kaip pavyzdžiui nutinka su semaforais, kai iš anksto būna apibrėžti ir įrašyti gestai ir jie yra atpažįstami. Manipuliavimo gestai yra kita forma sąveikos su kuria susiduriame, nes šie gestai dažniausiai naudojama sąveikauti su kompiuteriu, iš tikrųjų tai susiję su pele ar plunksna judinti žymeklį ar manipuluoti langą. Trimatėje aplinkoje manipuliavimo gestai yra taikomi fiziniams objektams, kurie atvaizduoja skaitmeninius objektus ekrane. Gestai rodo veiksmus, kurie nurodo į taikymo sritį, pavyzdžiui, vartotojas gali būti rodyti į didelį ekraną, siekiant nustatyti konkretų objektą, rasti arba gal perkelti objektą Semaforo modelis apibrėžia stilizuotus statinius ir dinامينius gestus, pavyzdžiui, gali būti daroma prielaidą, kaip pavyzdžiui rankos mojavimas.

### 2.3.2. Gestų atpažinimas

Atlikti gestų atpažinimą nėra lengva užduotis, nes yra nevienareikšmiška problema, arba, tiksliau, ne taip lengva sukurti veiksmingus euristinius ar mašinų mokymosi algoritmus, kurie gali suprasti, kas yra konkretus gestas. Gestas analizė turėtų izoliuoti gestų charakteristikas ir pradėti kurti modelį, atpažįstantį tam tikrus gestus. Kai kuriais atvejais yra reikalingi psichologiniai tyrimai, norint suprasti kontekstą, kuriame atliekami gestai. Gestai atrinkti bendrauti ir įgauti sistemos kontrolę neturėtų būti per daug panašūs, kitaip jie reikalaus labai tikslaus judesio, kitu atveju sunku pakeisti su visišku praradimu intuityvumo ir natūralių komponentų, kuriuos gestas gali turėti. Kaip [7] siūlo tai, kad gestai turi būti paprasti ir intuityvūs, lengvai pakartojami. Duoti realaus laiko grįžtamąjį ryšį iš vartotojo ekrane ir nustatyti vizualių valdymo ciklą sukurti tinkamą grįžtamąjį ryšį.

### 2.3.3. Kūno sekimas

Kūno judėjimo, vaidmuo kaip aprašyta [9] yra pamiršta, nes prietaisai ir davikliai, kuriuos vartotojas turi dėvėti stipriai sumažina malonumą judėti aplinkoje. Skeleto stebėjimas leidžia daugiau intuityvias ir natūralias sąveikas, be poreikio dėvėti tam tikrus prietaisus ar išmokti sudėtingus nurodymus juos naudoti. Pagalvokite apie visas taikymo galimybes medicinos srityje, reabilitacijos laikotarpiu siekiant padėti žmonėms, arba švietimo sektoriuje, kaip pavyzdžiui aprašyta [10] kur Kinect natūralios sąveikos sąsaja, kuri suteikia viso kūno sekimą yra apibrėžiama kaip interaktyvi technologija, kuri gali padidinti studentų ar mokinių dalyvavimą ir aktyvumą pamokose ar paskaitose, sudaryti tinkamas pedagogines galimybes studentams sąveikauti su šiomis technologijomis. Ši Kinect bruožas siūlo teikti fizinius ir intuityvius sąveikos modelius ir pedagoginės strategijos, kaip siūloma, ji gali skatinti studentų dalyvavimą.

## 2.4. Esami sprendimai pasaulyje

Programinės įrangos, kuri būtų specializuota tik į sporto treniruočių rengimą nėra daug palyginus su kitomis programomis. Žinoma yra sukurta žaidimų, kurie suteikia panašias galimybes, bet minusas, kad treniruotės jau būna paruoštos iš anksto, ir jų nebegalima redaguoti, tik atkartoti jas. O šio kuriamo produkto pagrindinė idėja ir yra kad treniruotes būtų galima sukurti, bei redaguoti pagal pageidavimus. Rimčiausias produktas mano nuomone yra „Nike + Kinect Training“, išleidimo data 2012 metų spalio mėnuo, taigi vėl matome kad tai tikrai nauja rinka. Paanalizuokime, programinę įrangą ar žaidimus kurie jau yra rinkoje:

- **Your Shape Fitness Evolved**

[11] Vienas iš pirmųjų treniruočių Kinect žaidimų, žaidimas, kuris neseniai buvo išleistas visame pasaulyje garsaus žaidimo kūrėjo "Ubisoft". Žaisdami šį žaidimą, jūs fiziškai sąveikaujate su juo, naudodamiesi savo kūnu. Žaidimas prasideda kai įrenginys nuskaitys jūsų kūną, bei sudėjamą ir pritaiko visa žaidimą įtraukiant ir meniu pasirinkimus. Šis žaidimas suteikia galimybes turėti asmeninį trenerį ar imtis įprastinių klasių kaip jogos ar kovos menų.

- **Zumba® Fitness**

[12] Kitas labai puikus treniruočių žaidimas grindžiamas gerinant vartotojų fitnessą su intensyvia šokių mokymosi pagalba. Žaidimas gali būti nustatytas įvairių šokių stilių, pavyzdžiui, Reggaeton, Merengue, salsa, Cumbia, hip-hop, Mambo, rumba, flamenko ir "Calypso". Judesiai iš visų šių Lotinų Amerikos šokių stilių yra įtraukti į vieną vientisą fitnesso programą.

- **EA Sports Active 2.0**

[13] Šio žaidimo išskirtinumas yra, kad leidžia vartotojui tiksliai stebėti, kad pratimai turi įtakos jo kūnui. Šis žaidimas taip pat turi papildomų priedų kaip, pavyzdžiui, prietaisas skirtas stebėti širdies ritmą, pateikiama informacija realiu laiku Sporto Active 2.0 pagrindinis bruožas yra tai, kad siūlo atsiliepimus tiek viso kūno padėti, ir tai, kas vyksta jos viduje, darant pratimus taip, kad jums sudarytų kuo geresnį įspūdį.

Bet rimčiausias šios kuriamos programinės įrangos konkurentas rinkoje manau yra kaip minėjau anksčiau „NIKE + Kinect Training“ [14] manau Nike kompanija aprašymo nereikalauja, tai visiems puikiai žinoma kompanija. Nepriklausomai nuo jūsų fitnesso tikslų, „Nike + Kinect Training“ duos jums įrankius. Asmeniai treneriai su savo patirtimi, bei kartu su Kinect magija, gali pamatyti, kaip jūsų kūnas juda, įvertinti fizines stiprybes ir nustatyti tobulintinas sritis, ir sukurti asmeninį treniruočių planą, siekiant padėti jums pasiekti savo asmeninę geriausią formą. Su realaus laiko atsaku gauti informaciją ar pozicijos ir judesiai yra teisingi, „Nike + Kinect Training“ pagal norus programa vystosi, kaip ir jūs, ir padės jums tapti stipresniais, greitesniais bei grakštesniais.

## 2.5. Gestų atpažinimo algoritmų apžvalga

Kadangi nėra konkretaus literatūros šaltinio kur būtų apžvelgiami visi gestų atpažinimo algoritmai. Taigi teko pasinaudoti interneto pagalba ir tiesiog atrinkti tuos, kurie buvo daugiausiai minimi užklausoje ar forumuose:

- Golden Section Search algoritmas
- Hidden Markov Model algoritmas
- Dynamic Time Warping algoritmas

Programoje buvo pasirinktas Golden Section Search algoritmas, kadangi nebuvo aišku, kuris algoritmas yra pats optimaliausias ar geriausiai tinkantis gestų atpažinimui, taigi pasirinkimas buvo daromas atsižvelgiant į algoritmų įgyvendinimo sunkumą.

### **2.5.1. Golden Section Search algoritmas**

Golden Section Search, metodas yra naudojamas surasti ekstremumo taškus maksimumą arba minimumą vienaarūšės funkcijos (Vienarūšė funkcija turi tik viena maksimumą arba minimumą intervale  $[a,b]$ ). Mūsų atveju algoritmas veikimo principas toks pat. Algoritmas lygina du gestų šablonus, esamą kuris yra atliekamas vartotojo, ir tarp to kuris jau yra paruoštas. paieškos principas gana paprastas, jei iš pirmo karto neranda sutapimų, pirmasis šablonas šiek tiek yra pasukamas ir lyginimo procesas paleidžiamas iš naujo. Ribinė reikšmė yra naudojama nustatyto algoritmo tikslumui. Kaip rezultata, metodas grąžina atitikimo patikimumo rezultata kuris yra tarp 0 ir 1. Mūsų atveju a ir b apibrėžti paieškos intervale  $[a, b]$ , kur a ir b yra kampai[15].

### **2.5.2. Hidden Markov Model algoritmas**

Hidden Markov Model metodas, yra kolekcija baigtinių būsenų sujungtų perėjimais. Kiekviena būsena yra charakterizuojama dviejų rinkinių tikimybėmis: perėjimo tikimybe, ir arba diskrečia išėjimo pasiskirstymo tikimybe arba pastovia išėjimo tikimybės tankio funkcija, kuri atsižvelgiant į būsenas, kurios apibrėžia tikimybės sąlygas, išskleisti kiekvieną išvesties simbolį nuo baigtinės alfabeto arba nenutrūkstamo atsitiktinio vektoriaus. Bet šis algoritmas pasižymi sunku įgyvendinimu, todėl jis nebuvo pasirinktas. Eksperimentinėje dalyje patyrinėsime jį plačiau, kadangi yra pateikti apskritimo gesto atpažinimo rezultatai[16].

### **2.5.3. Dynamic Time Warping algoritmas**

Tradicinis Dynamic Time Warping algoritmas iš esmės yra dinaminio programavimo algoritmas, kuris naudojami rekursyvų Dynamic Time Warping atnaujinimo sąnaudas pridėdamas atstumą tarp dviejų sujungtų elementų iš dviejų sekų kiekvienu rekursijos žingsniu. Atstumas tarp dviejų elementų yra dažnai vadinamas Euklido atstumu, kuris suteikia lygius svorius visiems sekos matmenims. Pavyzdžiui tipinė gestų atpažinimo problema, kūno sąnarių jungtys naudojamos geste gali kisti nuo gestų klasės iki gestų klasės. Taigi ne visas sąnarių jungtis yra vienodai svarbu atpažinti geste. Dynamic Time Warping yra šablonų atitikimo algoritmas surasti geriausią atitikmenį duotam šablonui, iš jau esamų paruoštų šablonų, kur šablonai gali būti laiko sekos matavimai ar savybės gautos iš matavimų. Gestų atpažinimo atveju, tikriname žmogaus atliekamą gestą su tuo kuris yra išsaugotas programos[18].



## **3. ŽMOGAUS SKELETO JUDESIŲ ĮRAŠYMO IR ATKŪRIMO PROJEKTAS**

### **3.1. Pažintis su sistema**

#### **3.1.1. Sistemos funkcijos**

Žemiau pateikiamos sistemos funkcijos su trumpais aprašymais:

- Judesių įrašymas. Funkcija skirta atpažinti ir įrašyti trenerio judesius, bei juos išsaugoti
- Judesių atpažinimas. Taip pat galima atskirti kaip atskira funkciją, kuri yra svarbiausia mūsų programoje, kad vyktų tikslus atpažinimas.
- Atgalinis ryšys. Taip pat svarbi funkcija mūsų programoje, kadangi vartotojas galės gauti atgalinį ryšį ir sužinoti ar jis atlieka teisingai judesius.

#### **3.1.2. Sistemos apribojimai**

Norint naudotis sistema reikės:

- Personalinio kompiuterio su su įdiegta Windows 7 ar aukštesne operacine sistema.
- Kinect įrenginys
- Turi būti įdiegtas Microsoft Kinect SDK

Kompiuteris turi atitikti tokius reikalavimus:

- 32 bit (x86) arba 64 bit (x64) procesorius
- Dviejų branduolių 2.66-GHz arba greitesnis procesorius
- USB 2.0 jungtį
- 2 GB operatyviosios atminties.

### **3.2. Projekto paskirtis**

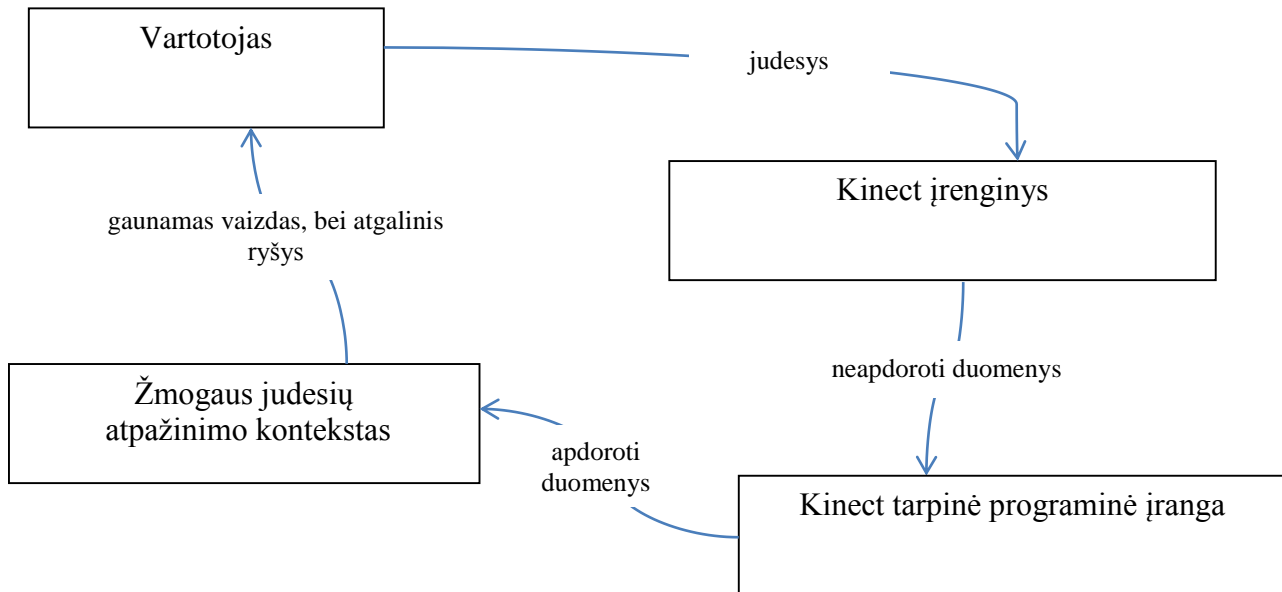
Sukurta programa kuri sugeba įrašyti ir atpažinti žmogaus kūno judesius, bei priklausomai nuo rezultatų duoti atgalinį ryšį. Pagrindinis tikslas, kurio buvo siekiama projekto metu suprasti ir sužinoti kaip vyksta žmogaus kūno judesių atpažinimas. Sistema yra paprasta vartojimui, ir nereikia jokių papildomų pasirošimų norint ja naudotis. Programa skirta naudotis Windows operacinių sistemų aplinkoje.

Sistemos teikiamos funkcijos:

- Judesių įrašymas.
- Judesių atpažinimas.
- Atgalinis ryšys.

### 3.3. Veiklos kontekstas

Žemiau pateikiama 4 pav. veiklos konteksto diagrama. Svarbiausia veiksmi vyksta Kinect įrenginio tarpinėje programinėje įrangoje, kadangi joje apdorojami duomenys.



4 pav. Konteksto diagrama

### 3.4. Sistemos funkciniai reikalavimai

Pateikiami išskelti funkciniai reikalavimai sistemai:

- Palaikius ilgiau, ant pasirinkto meniu punkto turi įvykti paspaudimas.
- Įrašoma judesių seka.
- Atlikus norimus judesius, jie yra išsaugomi
- Kai vartotojas pradeda judėti prieš Kinect įrenginį judesiai yra atpažįstami.
- Keisti ar modifikuoti kaip vartotojas turi bendrauti su sistema.
- Juda pelės cursorius, kai yra judinama ranka.
- Įjungus Kinect įrenginį bei paleidus sistemą užkrauna jau paruoštus judesius.
- Įjungus Kinect įrenginį, bei pasileidus sistemą įvykdomas žmogaus viso kūno bendradarbiavimas su sistema.
- Tikrinama ar vartotojas su sistema bendrauja sėkmingai.
- Nesėkmingo bendravimo atveju tarp vartotojo ir sistemos, grąžina rezultatą.
- Sistema pašaliną nereikalingus garsus bei daiktus iš fono, sutelkdama dėmesį tik į vartotoją.
- Vartotojas iš sistemos gauna atgalinį ryšį kai jis atlieka gestus.

### 3.5. Sistemos nefunkciniai reikalavimai

Pateikiami išskelti funkciniai reikalavimai sistemai:

- Vartotojo sąsaja neturėtų būti sudėtinga.
- Meniu turėtų būti taip pat lengvai naviguojamas, su aiškiais pavadinimais, arba net paaiškinimais, kad vartotojui nekiltų klausimų.
- Vartotojo sąsaja turėtų būti įdomi, bei gražiai atrodanti kad vartotojas jaustų malonumą ja naudodamasis.
- Nereikalingas papildomas apmokymas naudotis sistema.

- Kadangi mūsų sistema tai kuriama treniruočių sistema, kuri atpažįstą žmogaus judesius, taigi reikėtų kad vartotojo abi rankos bei kojos būtų laisvos.
- Svarbu kad mūsų sistema greitai atpažintų žmogaus judesius.
- Svarbu kad mūsų sistema greitai duotų atgalinį atsaką.
- Sistema privalo tiksliai sekti vartotojo judesius.
- Svarbu kad patalpa kurioje atliekami judesiai būtų apšviesta.
- Kinect įrenginys taip pat turi būti saugomas, jei jis bus pažeistas fiziškai, gali atsitikti taip kad mūsų sistema tiesiog nefunkcionuos.
- Kadangi yra išleistas galutinis produktas, taigi palaikymo nėra numatyta, nebent jei norima išplėsti dar labiau funkcionalumą galima kurti naują sistemos versija.
- Kadangi mūsų sistema atpažįstą kūno judesius, jei nėra taikomi politiniai ar kultūriniai apribojimai.
- Sistema kuriama konkrečiai organizacijai, taigi jei kitos organizacijos norės taip pat ja pasinaudoti teks tartis su užsakovų organizacija.

## **3.6. Sistemos architektūra**

### **3.6.1. Paskirtis**

Sistemos architektūros skyrius apžvelgia pagrindinius sistemos architektūrinius aspektus, naudojant daug įvairių architektūrinių vaizdų, kurie skirti pavaizduoti skirtingus sistemos aspektus. Taip pat dokumentas skirtas fiksuoti ir perteikti svarbius architektūrinius sprendimus, kurių yra imtasi kuriant sistemą. Dokumentas reikalingas komunikacijai tarp projekto kūrėjų ir kitų žmonių, dalyvaujančių projekto kūrime: projekto vadovo, užsakovo bei atlikto darbo vertintojo.

### **3.6.2. Architektūros pateikimas**

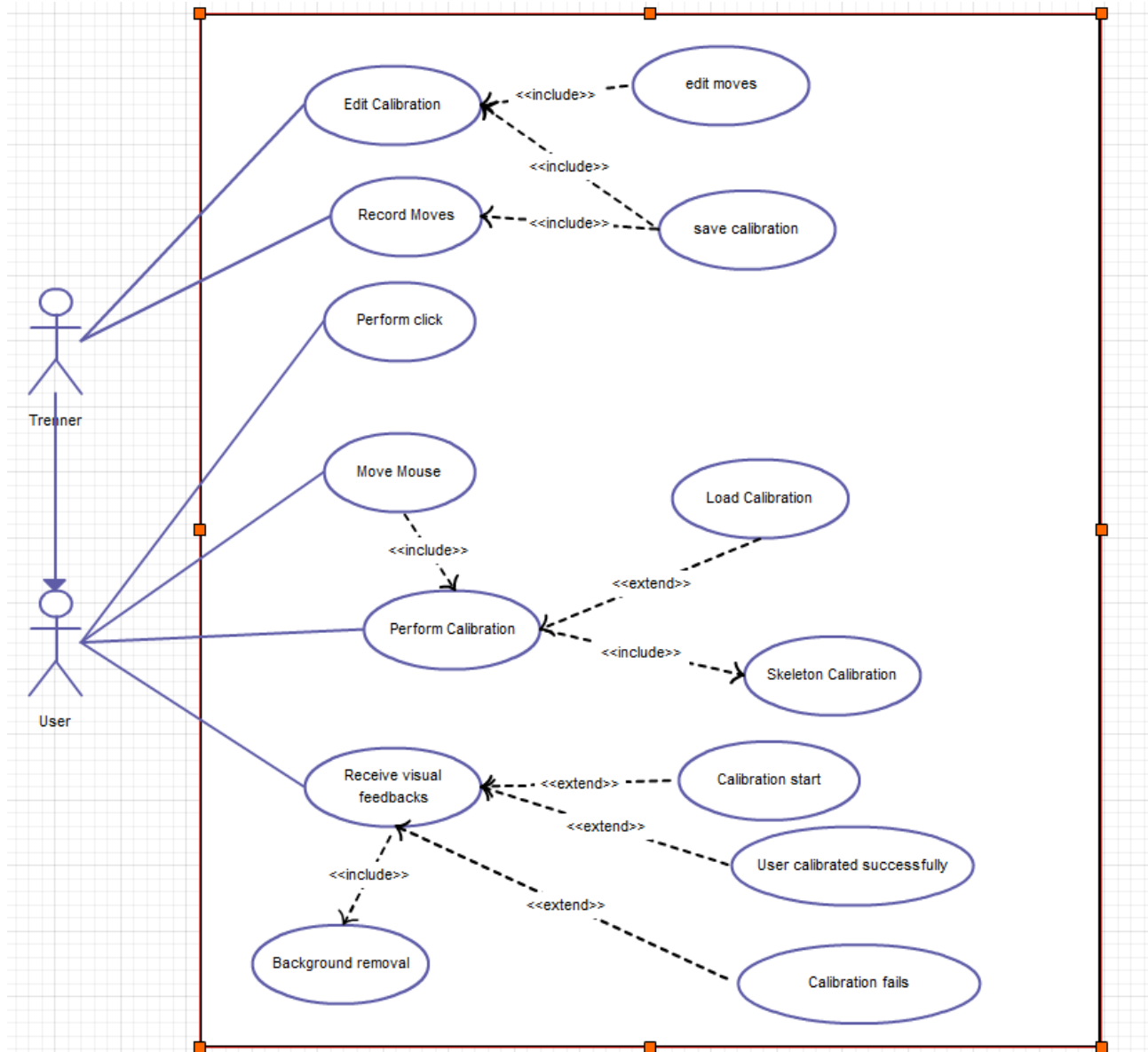
Šiame dokumente pateikiami šie vaizdai:

- Panaudojimo atvejų vaizdas – vaizduojamas naudojant panaudos atvejų diagramą.
- Sistemos statinis vaizdas – vaizduojamas naudojant paketų ir klasių diagramas.
- Sistemos dinaminis vaizdas – pateikiamas naudojant sąveikos, būsenų, veiklos diagramomis, bei viena bendradarbiavimo diagrama.

### **3.6.3. Architektūros tikslai ir apribojimai**

Ši sistema sukurta naudojant, C# programavimo kalbą, bei Microsoft Kinect SDK, Microsoft Visual Studio 2010 aplinkoje, taigi mūsų sistema veiks tik Windows operacinėje sistemoje. Operacinės sistema versija turėtų būti Windows 7 ar aukštesnė. Svarbu atsižvelgiant, kad sistema buvo kuriama esant 1.5 Microsoft Kinect SDK versijai, taigi jei yra įdiegta naujesnė versija yra galimybė, kad programa gali veikti nekorektiškai, dėl nepalaikomų funkcijų ar kitų veiksmų. Projektavimo procese naudojami internetiniai įrankiai, kuriuos galima rasti adresu <http://creately.com/>, bei Microsoft Visio. Architektūros tikslas tinkamai ir atsakingai pasiruošti sistemos programavimo darbams, bei sutaupyti laiko.

### 3.6.4. Panaudojimo atvejų vaizdas

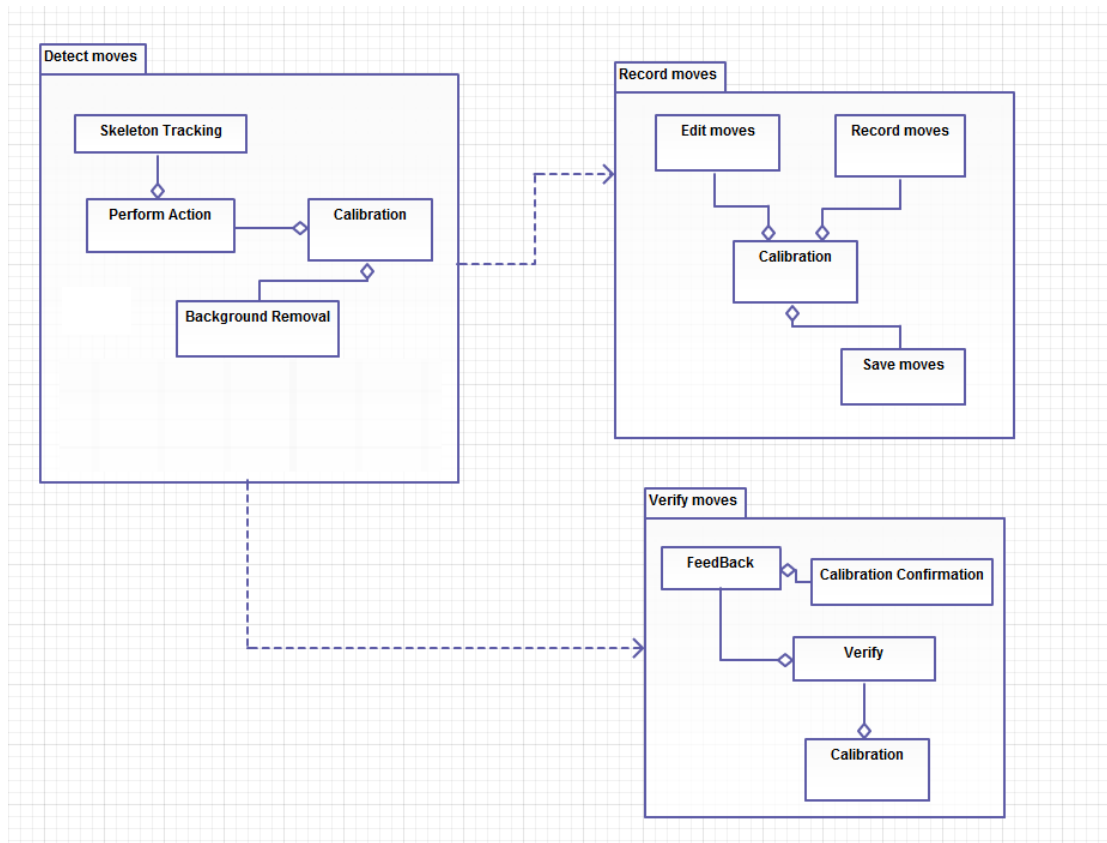


5 pav. Panaudos atvejų diagrama.

5 pav. pavaizduota esamos sistemos panaudos atvejų diagrama. Kaip matyti ši sistema turi du vartotojus, tai trenerį ar žmogų kuris parengia gestus, ir vartotoją kuris juos atlieka. Taipogi skiriasi šių vartotojų galimybės, treneris gali įrašyti judesius, bei juos redaguoti, vartotojas gali juos tik atlikti, peržiūrėti atlikti, bei gauti atgalinį ryšį iš sistemos.

### 3.6.5. Statinis sistemos vaizdas

6 pav. pateikiamas statinis sistemos vaizdas, kurio detalesni paaiškinimai yra pateikiami žemiau.



6 pav. Sistemos išskaidymas į paketus

#### 3.6.5.1. Detect Moves paketas

*Apribojimai.* Šį paketą sudaro klasės skirtos žmogaus skeleto judesių atpažinimui, taigi šis paketas kartu su klasėmis yra sistemos branduolys, taipogi jį naudos ir kiti komponentai. Išoriniai vartotojai su sistema gali bendrauti tik per numatytas klases bei metodus.

*Struktūra.* Paketą sudaro klasės:

- *Calibration* – klasė koordinuojanti sąveiką tarp vartotojo ir sistemos.
- *PerformAction* – klasė tikriną kokius veiksmus atlieka vartotojas.
- *BackgroundRemoval* – klasė koordinuoja pašalinių trikdžių pašalinimą.
- *SkeletonTracking* – klasė koordinuoja žmogaus skeleto judesių sekimą bei atpažinimą.

*Sąveikavimas.* Paketo klasės bendrauja su visais kitais sistemos paketais per numatytas sąsajas. Šis komponentas nenaudoja kitų komponentų, bet jį naudoja Verify Moves ir Record moves komponentai. Jei šis komponentas nebus įgyvendintas, kiti komponentai negalės veikti, kaip ir visa sistema.

*Resursai.* Paketas atlieka veiksmus su išoriniu objektu tai Kinect įrenginiu, paketas bendrauja su Kinect įrenginiu per NUI, natūralią vartotojo sąsaja.

*Skaičiavimai.* Paketas atlieka žmogaus skeleto judesių atpažinimą bei trukdžių šalinimą.

*Sąsaja.* Paketą sudarančios klasėmis naudojasi kitų paketų objektai. Calibration – klasės skirtos abipusiam bendravimui tarp sistemos ir išorinio vartotojo.

### 3.6.5.2. Record Moves paketas

*Apribojimai.* Paketą sudarančios klasės atlieka žmogaus skeleto įrašymą bei išsaugojimą, būtina išsaugoti tinkamu formatu, kad kitas komponentas galėtų tikrinti.

*Struktūra.* Išskirtinos tokios paketą sudarančios klasės:

- *RecordMoves* – Klasė skirta įrašyti judesius.
- *SaveMoves* – Klasė skirta išsaugoti judesius.
- *EditMoves* – klasė skirta redaguoti esamus judesius.
- *Calibration* – klasė koordinuojanti sąveiką tarp vartotojo ir sistemos. Šiame pakete ji įdėta dėl aiškumo kadangi, veiksmai vyksta tik tada kai vartotojas sąveikauja su sistema, jos detalesnis aprašymas 3.1 skyriuje.

*Sąveikavimas.* Paketo klases naudoja Detect Moves paketo klases bei objektai. Patys paketo klasių objektai į kitus nesikreipia.

*Resursai.* Išoriniais resursais nesinaudoja.

*Skaičiavimai.* Šiame komponentai jokie sudėtingi skaičiavimai neatliekami.

*Sąsaja.* Šis paketas sąveikauja kartu su DetectMoves paketu, kadangi pastarasis paketas atpažįsta judesius, o RecordMoves paketas juos įrašo bei išsaugo.

### 3.6.5.3. Verify Moves paketas

*Apribojimai.* Paketas skirtas, atlikti veiksmų tikrinimą, ar vartotojas atlieka gerai pratimus, taipogi priklausomai nuo rezultatų ar jis tai atlieka teisingai ar ne, duoti atgalinį ryšį.

*Struktūra.* Paketą sudaro klasės:

- *Calibration* – klasė koordinuojanti sąveiką tarp vartotojo ir sistemos. Šiame pakete ji įdėta dėl aiškumo kadangi, veiksmai vyksta tik tada kai vartotojas sąveikauja su sistema, jos detalesnis aprašymas 3.1 skyriuje
- *Verify()* – klasė atliekanti judesių tikrinimą.
- *FeedBack()* – Klasė atsakinga už vartotojo atgalinio ryšio davimą.
- *Calibration Confirmation* – klasė gražinanti rezultatą apie tarpininkavimą tarp sistemos ir vartotojo.

*Sąveikavimas.* Paketo klases naudoja DetectMoves paketas.

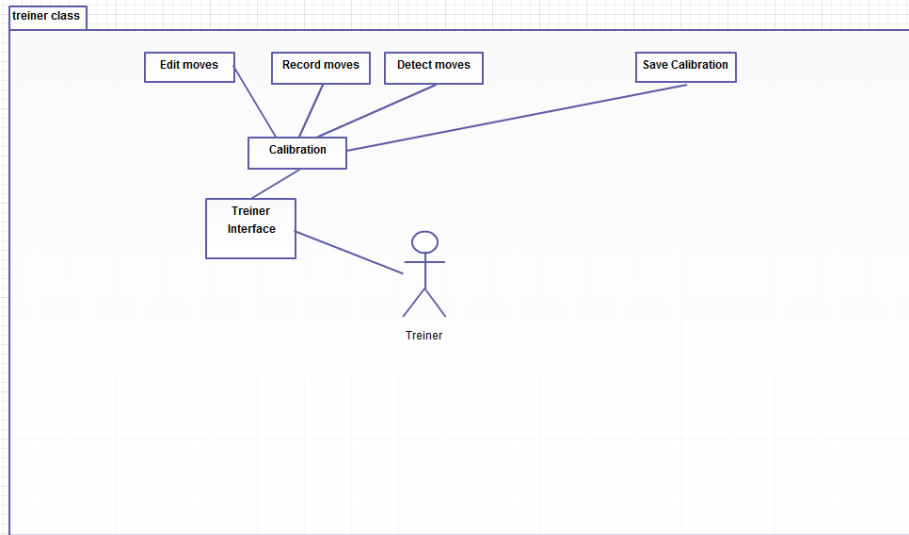
*Resursai.* Komponentas neturi priėjimo su išoriniais resursais.

*Skaičiavimai.* Pakete esančios klasės atlieka veiksmų atpažinimo tikrinimą.

*Sąsaja.* Šis paketas naudojamas tada, kai vartotojas nori pradėti treniruočių programą, pradedama sąveikauti su sistema, vartotojas ekrane mato pratimus, o vartotojas turi juos atlikti komponentas turi tikrinti ar jie atliekami teisingai, viskas vyksta realiame laike.

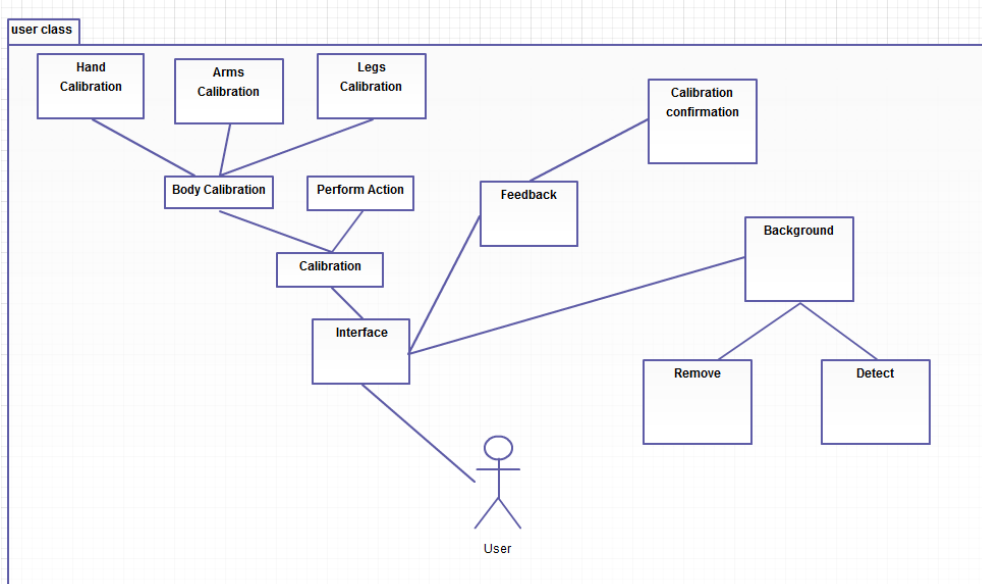
### 3.6.6. Klasių diagrama

Kaip minėta 3.6.4 skyriuje sistema turi du vartotojus taigi 7 pav. pateikiamas sistemos klasių vaizdas, kurios sudaro trenerio klases. Klasių pavadinimai gali asocijuotis su metodų pavadinimais, bet šiuo atveju tai klasės, pavadinimai buvo rašomi, kad būtų lengva atskirti kam kiekviena klasė skirta. Calibration klasė apjungia, su judesiais susijusias klases, taipogi išsaugojimą.



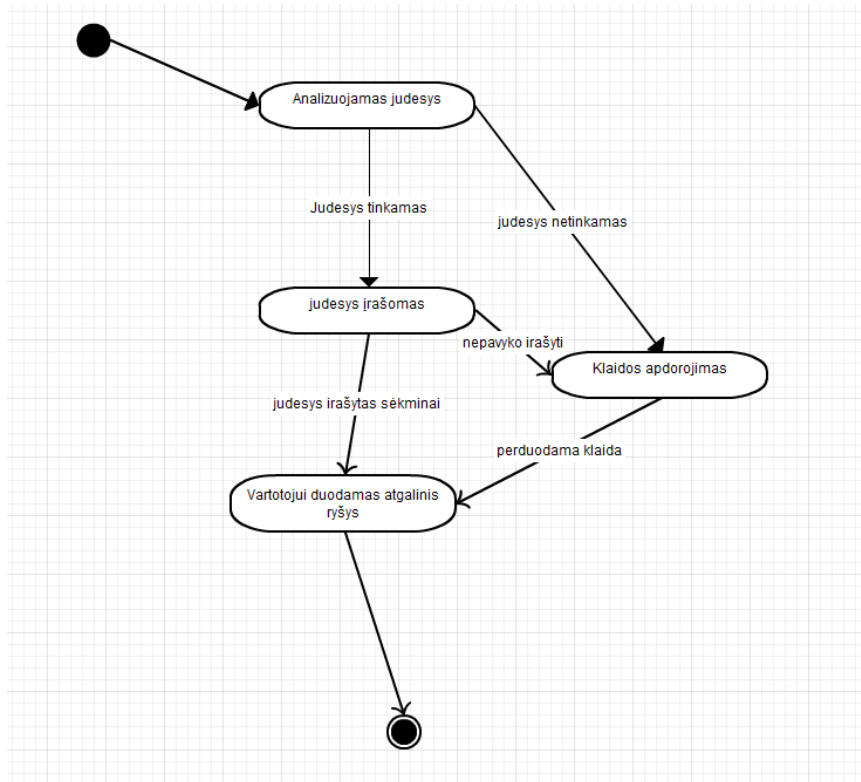
7 pav. Trenerio klasių diagrama.

Kaip minėta 3.6.4 skyriuje sistema turi du vartotojus taigi 8 pav. pateikiamas sistemos klasių vaizdas, kurios sudaro vartotojo klases, taipogi reikia atkreipti dėmesį kad treneris paveldi visas vartotojo klases. Taip kaip ir trenerio klasių diagramoje pavadinimai gali asocijuotis su metodų pavadinimais, bet šiuo atveju tai klasės, pavadinimai buvo rašomi, kad būtų lengva atskirti kam kiekviena klasė skirta. Pagrindinė klasė pavadinimu interface, skirta įgyvendinti sąveika tarp vartotojo ir sistemos, bei duoti atgalinį ryšį.



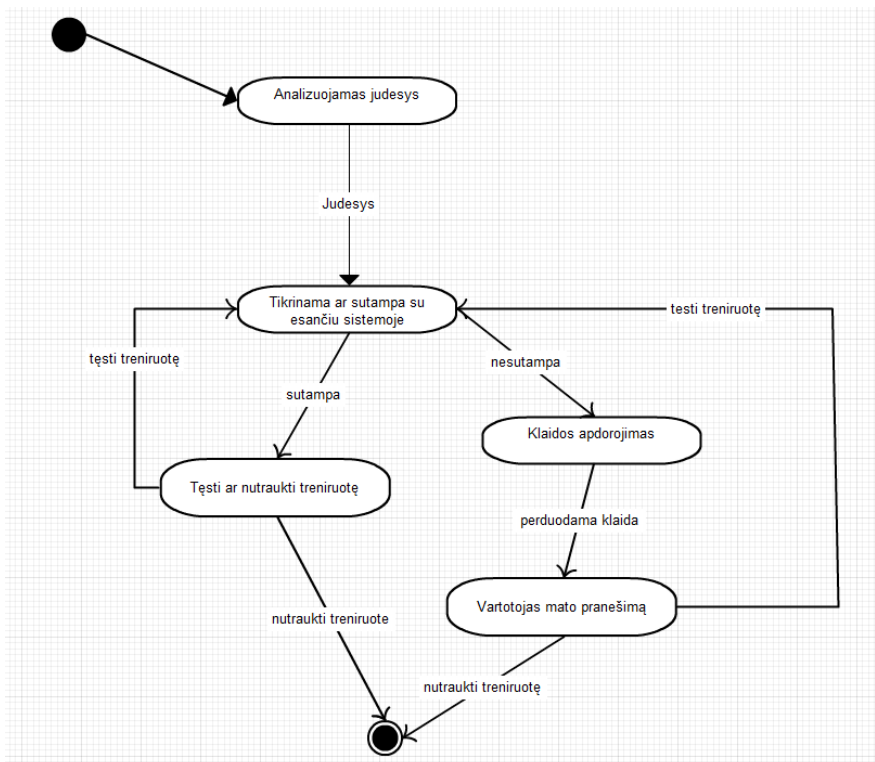
8 pav. Vartotojo/atleto klasių diagrama.

### 3.6.7. Būsenų diagramos



9 pav. Judesio įrašymo būsenų diagrama

9 pav. Pateikiama judesių įrašymo būsenos diagrama, būseną yra sužadinama ir pradedamas judesių analizavimas, yra galimi du variantai judesys tinkamas arba ne, tikimo atveju judesys yra įrašomas į sistemą, jei ne vyksta klaidos apdorojimas, abejais atvejais vartotojas gauna atgalinį ryšį.

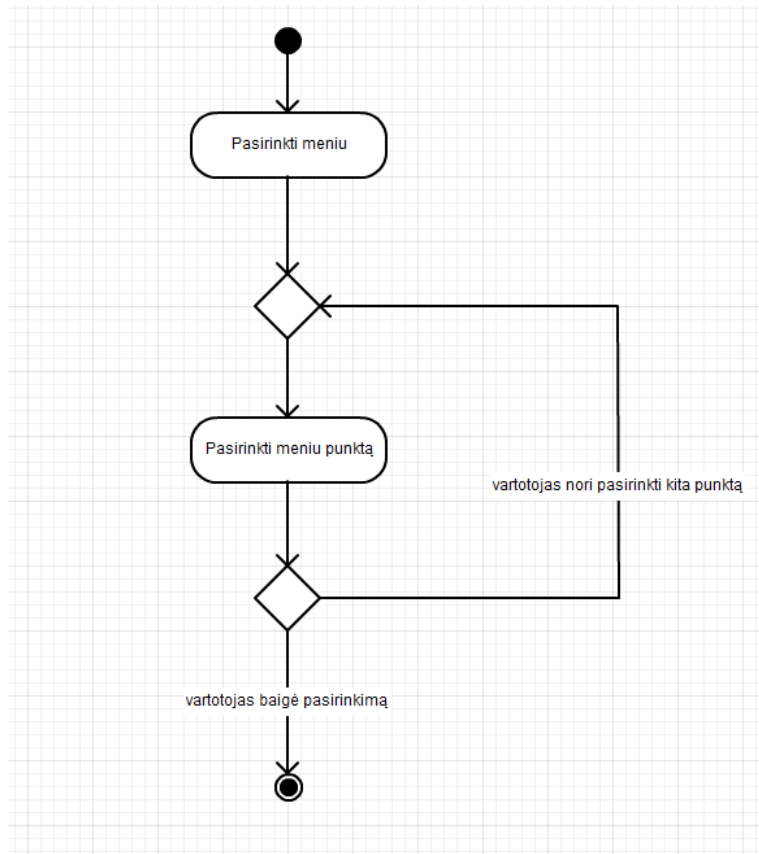


10 pav. Judesio atpažinimo būsenų diagrama



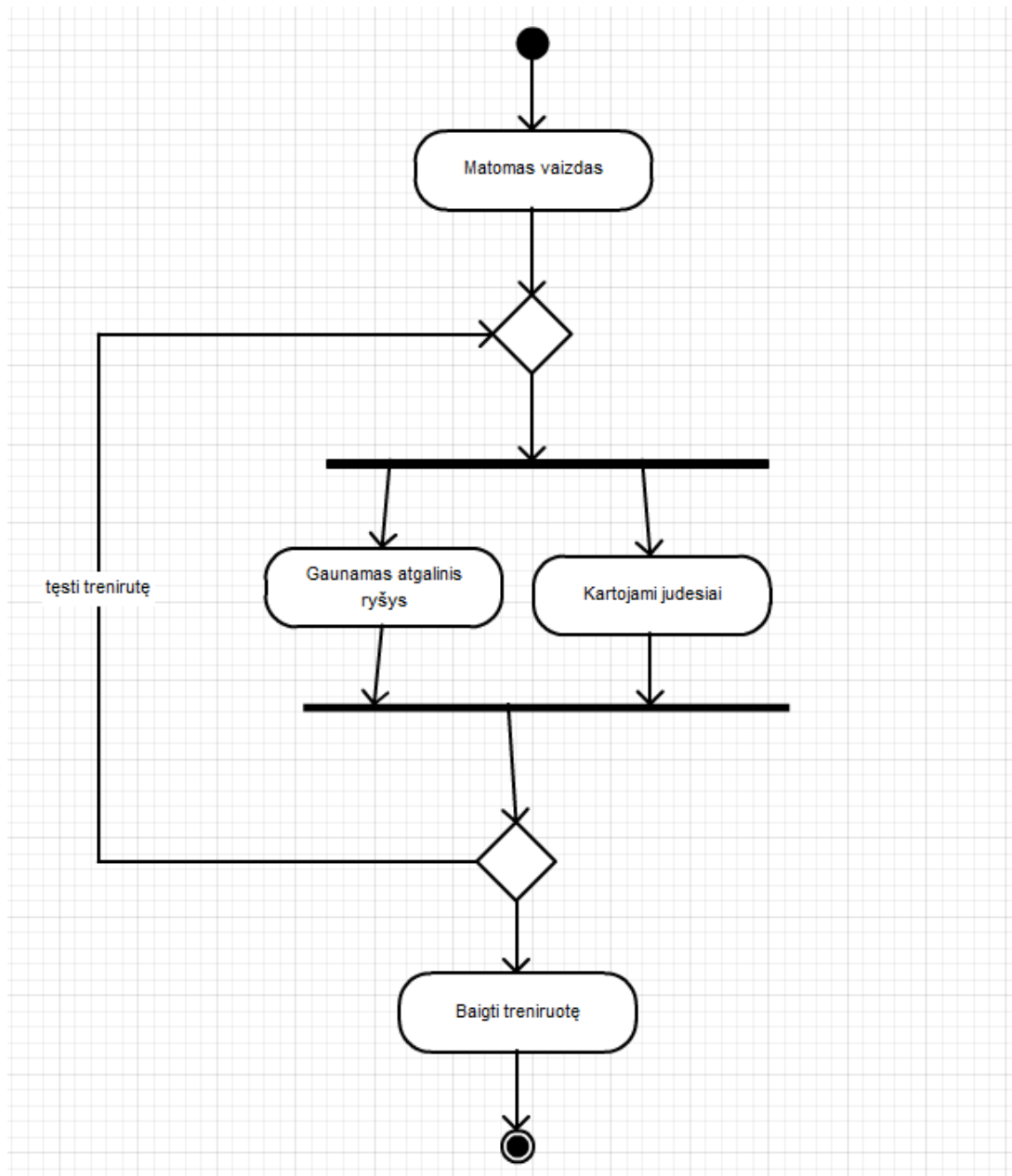
**10 pav.** Pateikiama judesių atpažinimo būsenos diagrama, būseną yra sužadinama ir pradedamas judesių analizavimas, sistema tikrina ar rodomas judesys sutampa su sistemoje jau paruoštu judesio šablonu, nuo rezultatų priklauso atgalinis ryšys, jei rezultatas teigiamas tęsiamas judesių atpažinimas, kai rezultatas neigiamas vartotojas ekrane mato, kad nėra atpažintas joks gestas.

### 3.6.8. Veiklos diagramos



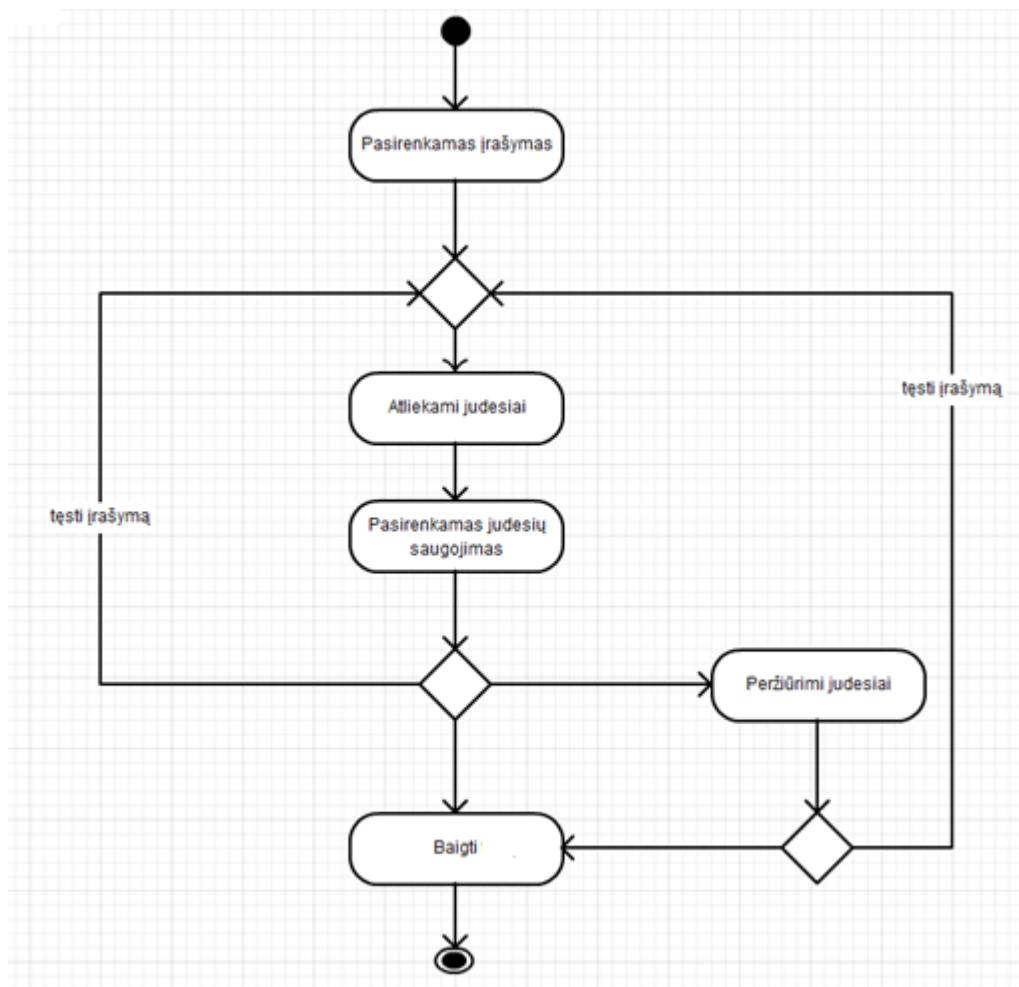
**11 pav.** Meniu punkto pasirinkimo veiklos diagrama

**11 pav.** Pateikiama meniu veiklos diagrama, veikla prasideda vaikščiojimas po meniu, ir vartotojų veiksmų sekimas, sistema reaguoja į vartotojo paspaudimus.



**12 pav.** Treniruotės veiklos diagrama

**12 pav.** Pateikiama treniruotės veiklos diagrama, veikla prasideda kai vartotojas ekrane prieš save mato vaizdą arba iš anksto žino kokius judesius jam reikia atlikti jau pagal paruoštus šablonus, tada pradamas atpažinimas, sistema duoda atgalinį ryšį, kartojami judesiai kol galiausiai vartotojas nusprendžia baigti treniruotę.



13 pav. Judesių įrašymo veiklos diagrama

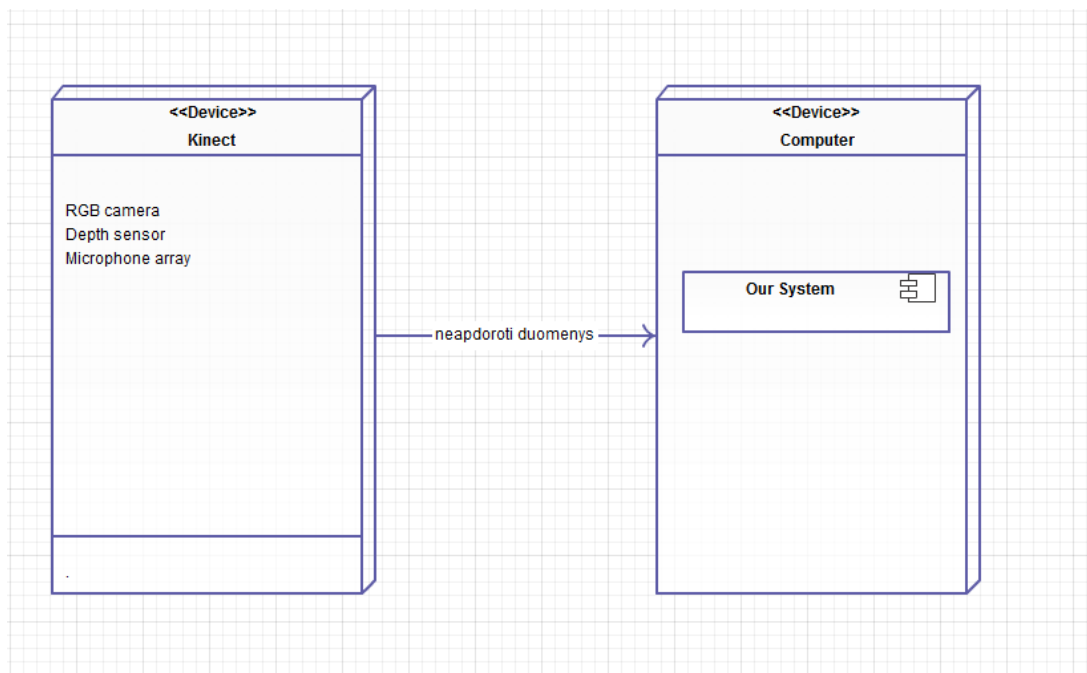
13 pav. Pateikiama judesių įrašymo veiklos diagrama, norint tai pradėti daryti reikia pasirinkti judesių įrašymą, sekantis žingsnis atlikti norimus judesius, juos išsaugoti, jei yra noras juos peržiūrėti ir tęsti toliau įrašymą arba baigti įrašymą.

**Pastaba:**

Sistema neturi didelių funkcionalumo vartotojo atžvilgiu, kadangi pagrindinis sistemos tikslas yra teisingai ir tiksliai atpažinti žmogaus judesius, taigi veiklos ir būsenų diagramų skaičius nėra didelis, nes sistema buvo sukurta koncentruojantis į gestų atpažinimo algoritmus.

**3.6.9. Išdėstymo vaizdas**

14 pav. pateikiamas išdėstymo vaizdas. Tarpusavyje sąveikauja Kinect įrenginys, kuris siunčia neapdorotus duomenis į mūsų sistemą, kuri yra įdiegta vartotojo kompiuteryje. Tada mūsų sistema apdoroja gautus duomenis iš įrenginio.



14 pav. Sistemos išdėstymo vaizdas

### 3.6.10. Testavimas

#### 3.6.10.1. Įvadas

Šiame skyriuje pateikiamas sukurtos programinės įrangos testavimo apžvalga. Bei tolimesniuose skyriuose yra pateiktas programinės įrangos testavimo planas.

#### 3.6.10.2. Testavimo tikslai ir objektai

Produkto testavimo tikslas yra pašalinti visus programinės įrangos defektus bei kitokius trūkumus. Tačiau šis tikslas yra teorinis, nes reali programinė įranga visuomet turės bent nedidelį kiekį klaidų. Šio tikslo įgyvendinimą riboja įvairūs veiksniai, kaip ribotas laikas, biudžetas ir pan. Todėl realiai įgyvendinamas tikslas yra sukurti kiek įmanoma geriau ištestuotą ir išstobulintą programinę įrangą.

Pagrindinis siekis yra sukurti pakankamai stabilų programinės įrangos produktą, turintį toleruotiną kiekį loginių klaidų. Taip pat svarbu užtikrinti pakankamai gerą tinkamumą naudoti bei patogumą. Tokie veiksniai, kaip programinės įrangos greیتaveika, yra tik antraeilis tikslas, nes kuriamas produktas yra tik prototipas.

#### 3.6.10.3. Vienetų testavimas

Vienetų testavimas atliktas laikantis baltosios dėžės testavimo strategijos. Pagrindinis dėmesys buvo skiriamas žmogaus kūno judesių komponento testavimui, nes nuo jo priklauso visi kiti moduliai.

Įprastose situacijose vienetų testavimas laikysis testavimais paremtos kūrimo (angl. „test driven development“) principų ir atliktas tokiais žingsniais:







- 1) Parašomas testas naujam funkcionalumui;
- 2) Įsitikinama, jog naujas testas atskleidžia naujo funkcionalumo nebūvimą (kyla testo klaida);
- 3) Realizuojamas naujas funkcionalumas;
- 4) Įsitikinama, jog naujas testas patvirtina, jog naujas funkcionalumas išties veikia;

Tiek kodas, tiek testai pertvarkomi, kad taptų lengviau skaitomi. Pertvarkymai atliekami mažai žingsniais, po kiekvieno tokio žingsnio iš naujo paleidžiant atitinkamos srities testus.

Tam tikrais atvejais yra leistinas programinio kodo rašymas prieš pačius vienetų testus, bet tik jei tai padės sutaupyti laiko ir naujai kuriamas funkcionalumas būtų nesudėtingas. Vienetų testavimas atliekamas programinės įrangos kūrėjo.

#### 3.6.10.4. Vienetų testavimo rezultatai

Vienetų testavimas atliktas naudojant standartinį Visual Studio 2010 UnitTesting įrankį. Kadangi laiko testavimo darbams skirta nebuvo daug, taigi testuota klasė, kuri atsakinga už dvimačių vektorių matematinius veiksmus, sudėtį, atimtį, daugybą, dalybą iš skaičiaus, ilgio apskaičiavimo funkcijas. 15 pav. pateikiami testų rezultatai.

	Result	Test Name	Project	Error Message	
<input type="checkbox"/>	 Passed	TestVector2Zero	MathTests		
<input type="checkbox"/>	 Passed	TestVector2Length	MathTests		
<input type="checkbox"/>	 Passed	TestVector2MinusOperator	MathTests		
<input type="checkbox"/>	 Passed	TestVector2PlusOperator	MathTests		
<input type="checkbox"/>	 Passed	TestVector2MultiplyOperator	MathTests		
<input type="checkbox"/>	 Passed	TestVector2DivisionOperator	MathTests		

15 pav. Testavimo rezultatų pavyzdys

#### 3.6.10.5. Integracinis testavimas

Integracinis testavimas atliktas laikantis juodosios dėžės testavimo strategijos. Kadangi mano programinė įranga sudaryta iš trijų pagrindinių komponentų, kurie yra paminėti aukščiau. Iš pradžių ištestuota kiekvienas komponentas atskirai, po to jie apjungti į bendrą visumą ir tada ištestuota bendra jų sąveika.

#### 3.6.10.6. Priėmimo testavimas

Užsakovas ir projekto vadovas buvo supažindinami su testavimo rezultatais, bei jų akivaizdoje buvo atliekamas numatytas testų rinkinys. Kadangi mano testiniai atvejai negali būti sugeneruojami, kad patikrinti ar teisingai veikia programinė įranga, teko man stovėti prieš įrenginį ir bandyti įrašinėti bei po to bandyti atkartoti jau įrašytus judesius. Taipogi sistema buvo rodoma komisijai, kurią jie įvertinimo teigiamai.

## 4. KINECT ĮRENGINIUI SKIRTŲ GESTŲ ATPAŽINIMO ALGORITMŲ TYRIMAS

### 4.1. Įvadas

Šiame skyriuje apžvelgiami Kinect įrenginiui skirtų gestų atpažinimo algoritmai. Kiekvienas algoritmas turi savitą veikimo principą, bei pasižymi skirtingomis savybėmis. Visi šie algoritmai skirti spręsti mūsų problemą – žmogaus kūno judesių, gestų atpažinimą. Norint palyginti šiuos algoritmus yra žiūrima į taip kaip tiksliai jie sugeba atpažinti vienokius ar kitokius žmogaus judesius.

Kaip paminėta 2.5 skyriuje mūsų nagrinėjami algoritmai bus:

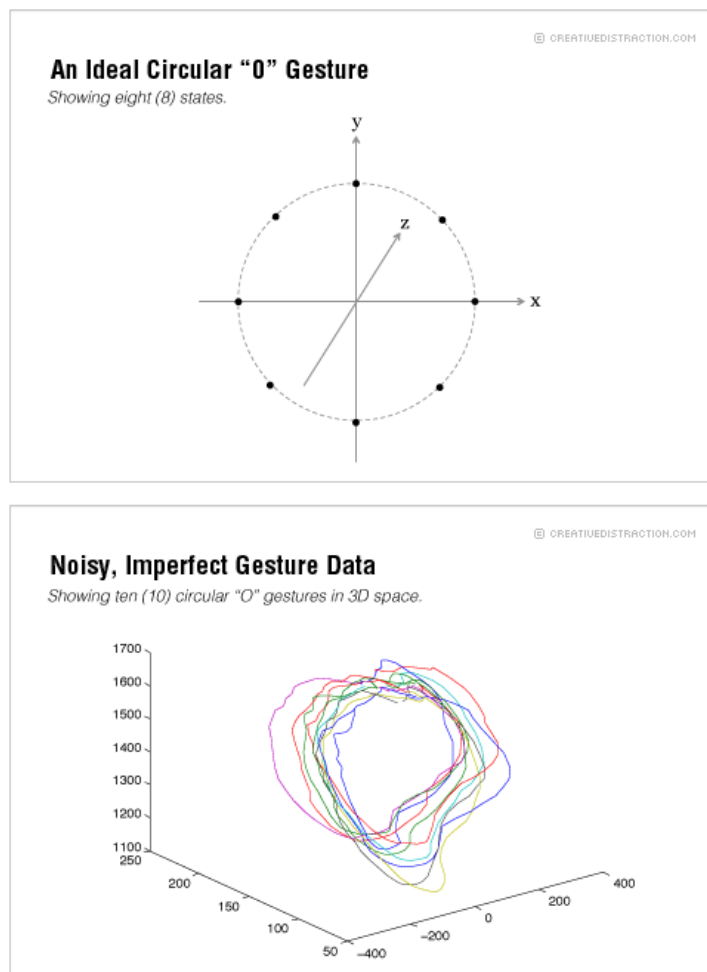
- Golden Section Search algoritmas
- Hidden Markov Model algoritmas
- Dynamic Time Warping algoritmas

Skyriuje apžvelgiama algoritmų matematiniai sprendimai, bei jų savybės.

## 4.2. Hidden Markov Model algoritmas

Gestų atpažinimo užduotis dalinasi tam tikrai panašumais su kitomis atpažinimo užduotimis, tokiomis kaip balso atpažinimas ar biometrinių duomenų: duotas signalas, kuris kinta laike, kaip galima nustatyti variacijas signale, kurios atitinka žinomus modelius. Nors alternatyvos buvo bandoms pasiekti dinaminio programavimo ar dinaminio programavimo atitikimo algoritmais, bet patys sėkmingiausi sprendimai, vis dar linkę įtraukti funkcijomis pagrįstą iš apačios į viršų statistinį mokymą, dažniausiai Hidden Markov Models.

Apsvarstykime rankų gestą, pavyzdžiui kaip apskritimą, padarytą kaip nupieštą apskritimą ant lentos be kreidos arba lentos. Izoliuotas laike apskritimo gestas turi idealią kanoninę formą ir triukšmingą, netobulą momentinę realybę kaip pavaizduota 16 pav. žemiau[16][19].



16 pav. Idealus ir Realus apskritimo gestas laike.

Hidden Markov Models yra turtingos matematinės struktūros; jos tarnauja kaip teorinis pagrindas įvairiausiuose taikymuose. Jis gali modeliuoti erdvės – laiko informaciją natūraliu būdu. Jis taip pat turi elegantišką ir veiksmingą algoritmą apmokymui ir atpažinimui, kaip Baum – Welch algoritmas ir Viterbi paieškos algoritmas.

Hidden Markov Models yra būsenų rinkinys sujungtų parėjimais. Kiekvienas parėjimas turi porą tikimybių: perėjimo tikimybę (kuri suteikia, tikimybę rasti perėjimą) ir išvesties tikimybę (kuri apibrėžia sąlyginę tikimybę sudarant išėjimo simbolių nuo baigtinio alfabeto duotos būsenos). Formalus HMM apibūdinamas taip:

- $\{s_1, s_2, s_3, \dots, s_n\}$  – Rinkinys  $N$  būsenų. Būsena laiko momentu  $t$ , yra žymima atsitiktiniu kintamuoju  $q_t$ .
- $\{v_1, v_2, v_3, \dots, v_m\}$  –  $M$  rinkinys atskirų stebėjimų simbolių ar atskirų alfabetų. Stebėjimai laiko momentu  $t$ , yra žymimi atsitiktiniu kintamuoju  $O_t$ . Stebėjimo simboliai atitinkantys fizinės sistemos išvestis yra modeliuojami.

- $A = \{a_{i,j}\}$  –  $N \times N$  matrica skirta būsenų tikimybių skirstinių perėjimui kur  $a_{i,j}$  yra tikimybė padaryti perėjimą iš būsenos  $s_i$  į  $s_j$ :

$$a_{i,j} = P(q_{t+1} = s_j | q_t = s_i).$$

- $B = \{b_j(k)\}$  –  $N \times M$  matrica skirta stebėjimo simbolių skirstinių tikimybėms, kur  $b_j(k)$  yra tikimybė išvesti  $v_k$  laiko momentu  $t$ , būsenoje  $s_j$ :

$$b_j(k) = P(O_t = v_k | q_t = s_j).$$

- $\pi = \{\pi_i\}$  – Pradinis būsenos pasiskirstymas kur  $i$  yra tikimybė, kad būseną yra  $s_i$  yra pradinėje būsenoje.

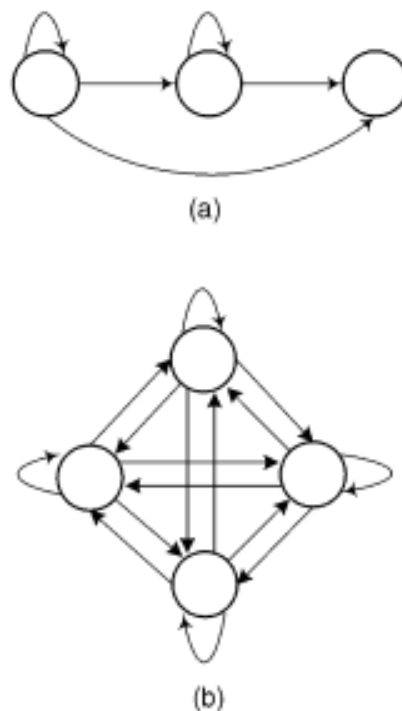
$$\pi_i = P(q_1 = s_i).$$

Kadangi  $A$ ,  $B$  ir  $\pi$  yra tikimybinės, turi paisyti šių apribojimų:

- $\sum_j a_{i,j} = 1 \forall i, \text{ ir } a_{i,j} \geq 0.$
- $\sum_j b_j(k) = 1 \forall j, \text{ ir } b_j(k) \geq 0.$
- $\sum_i \pi_i = 1 \text{ ir } \pi_i \geq 0.$

Remiantis konvencija, kompaktinis žymėjimas  $\lambda = (A, B, \pi)$  yra naudojamas, kuris įtraukia tik tikimybinis parametrus.

Keletas HMM topologinių pavyzdžių yra pavaizduoti 17 pav. Kairės – dešinės modelis yra pavaizduotas 17a pav. yra geras modeliuoti apribotom laiko eilutėms, kurių savybės nuosekliai kinta laikui bėgant [15]. Kadangi kairės – dešinės modelis neturi atgalinio kelio, būsenos indeksas taipogi didėja arba išlieka toks pat, kai laikas didėja. Kitais žodžiais, būseną vykdoma iš kairės į dešinę arba lieka kur ji buvo. Kita vertus kiekviena būseną ergodiškame arba pilnai sujungtume modelyje gali pasiekti kiekvieną kitą būseną vienu perėjimu, kaip parodyta 17b pav.



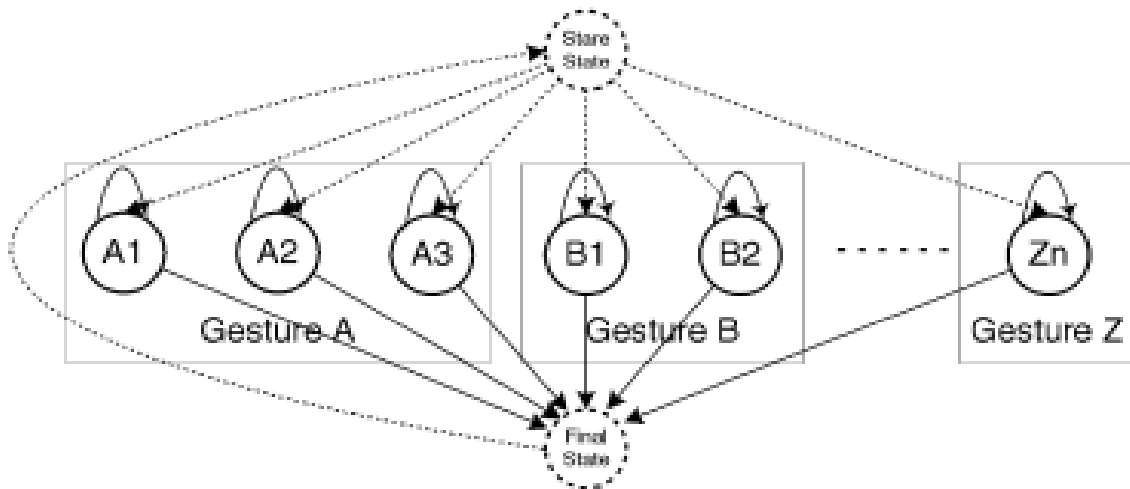
**17 pav.** HMM modelio tipai: (a) kairės – dešinės modelis (b) – ergodiškas modelis

Teisingam gestų aptikimui, gestų modelio tikimybė, duotajam šablonui turėtų būti pakankamai skirtinga. Deja nors ir HMM atpažinimas pasirenka modelį su geriausia tikimybe, mes negalime garantuoti, kad šablonas yra tikrai panašus į rodomąjį gestą, nebent tikimybės vertė yra pakankamai didelė. Paprasta riba dėl tikimybės dažnai neveikia. Todėl yra siūloma nauja koncepcija, vadinamas ribos modelis, kuris duoda tikimybės vertę, kuri yra naudojama kaip ribinis dydis. Gestas yra atpažįstamas tik tuomet jei tikimybė geriausio gesto modelio yra aukštesnė nei ribinio modelio tikimybė.

HMM vidaus segmentavimo savybė reiškia, kad kiekviena būseną su perėjimu į save sudaro ieškomo gesto segmentinį šabloną ir išeinantis perėjimą sudaro nuoseklią segmentų progresiją geste. Su šia savybe mes galime sudaryti ergodišką modelį su būsenomis nukopijuotomis iš gestų modelio sistemoje ir tada pilnai sujungti būsenas 17b pav. Pagal šį modelį, kiekviena būseną gali būti pasiekta visų kitų būsenų vienu perėjimu. Išėjimo stebėjimo tikimybės ir perėjimai į save tikimybės šiame modelyje yra laikomos kaip gestų modelyje, bet visos išeinančių perėjimų tikimybės yra lygiai priskirtos:

$$a_{i,j} = \frac{1 - a_{i,i}}{N - 1}, \text{ visiems } j, i \neq j,$$

kur  $a_{i,j}$  yra perėjimo tikimybė iš būsenos  $s_i$  į  $s_j$  ir  $N$  yra būsenų skaičius (suma visų būsenų neįtraukiant pradinės ir galinės būsenos). Pradinė ir galinė būseną nesuteikia jokių stebėjimų. 17b pav. vaizduoja ribinį modelį kaip supaprastintą versiją ergodiško modelio 18 pav. [18].



18 pav. Supaprastinta struktūra ribinio modelio. Punktyrinės rodyklės yra nuliniai parėjimai.



### 4.3. Dynamic Time Warping algoritmas

Pradinis DTW algoritmas buvo skirtas atitikmenų paieškai tarp laikinai iškraipytų modelių, randant suderinamą/deformuotą kelią tarp dviejų laiko eilučių  $Q = \{q_1, \dots, q_n\}$  ir  $C = \{c_1, \dots, c_m\}$ . Siekiant suderinti šias dvi sekas,  $M_{m \times n}$  matricia sukuriama, kur matricos pozicija  $(i, j)$  talpina sulyginimo sąnaudas tarp  $c_i$  ir  $q_j$ . Tada deformavimo kelio ilgis  $\tau$  yra apibrėžiamas kaip rinkinys gretimų matricių elementų, apibrėžiantys sujungimą tarp  $C$  ir  $Q$ :  $W = \{w_1, \dots, w_t\}$  kur  $w_i$  pozicijos indeksai sąnaudų matricioje. Deformavimo keliui paprastai taikomi keli apribojimai:

- Rėžių sąlygos:  $w_1 = (1,1)$  ir  $w_\tau = (m,n)$ .
- Tęstinumas ir monotoniškumas: Duotas  $w_{\tau'-1} = (a',b')$ , tada  $w_{\tau'} = (a,b)$ ,  $a - a' \leq 1$  ir  $b - b' \leq 1$  ši sąlyga verčia taškus  $W$  būti išsidėstę monotoniškai laike.

Visumoje mus domina galutinis deformavimo kelias kuris tenkina šias sąlygas, sumažinti deformavimo sąnaudas:

$$DTW(M) = \min_w \{M(w_\tau)\},$$

kur  $\tau$  kompensuoja skirtingus deformavimo kelio ilgius. Šis kelias gali būti rastas labai efektyviai naudojant dinaminį programavimą. Sąnaudos tam tikroje pozicijoje  $M(i, j)$  gali būti randos kaip kompozicija Euklido distancijos  $d(i, j)$  tarp sekos savybių vektorių  $c_i$  ir  $q_j$ , ir matricos gretimų elementų, mažiausios sąnaudos iki to momento, pavyzdžiui:

$$M(i, j) = d(i, j) + \min\{M(i-1, j-1), M(i-1, j), M(i, j-1)\}.$$

Pateiktas mūsų problemos srauto pobūdis, įėjimo vektorius  $Q$  neturi apibrėžto konkretaus ilgio ir gali būti sudarytas iš kelių pasikartojančių gestų šablonų  $C$ . Tuo metu sistema mano, kad tenais yra atitikimas tarp esamo bloko  $k$  esančio  $Q$  ir gesto, jei tenkinama ši sąlyga,  $M(m, k) < \mu$ ,  $k \in [1, \dots, \infty]$  duotai sąnaudų ribai  $\mu$ .

Ši ribinė vertė yra apskaičiuojama iš anksto naudojant palikti vieną iš kryžminio tikrinimo strategiją. Tai apima naudojimą vieno stebėjimo nuo pradinio pavyzdžio kaip tikrinamų duomenų ir likusių stebėjimų kaip apmokymų duomenys. Tai kartojama tol kol, kiekvienas stebėjimas pavyzdyje yra panaudojamas kartą kaip tikrinimo duomenys. Kiekvienoje iteracijoje mes įvertiname panašumą tarp kandidato ir likusių apmokytų rinkinių. Galiausiai mes pasirenkame ribinę reikšmę, kuri susijusi su didžiausių skaičiumi pataikymų.

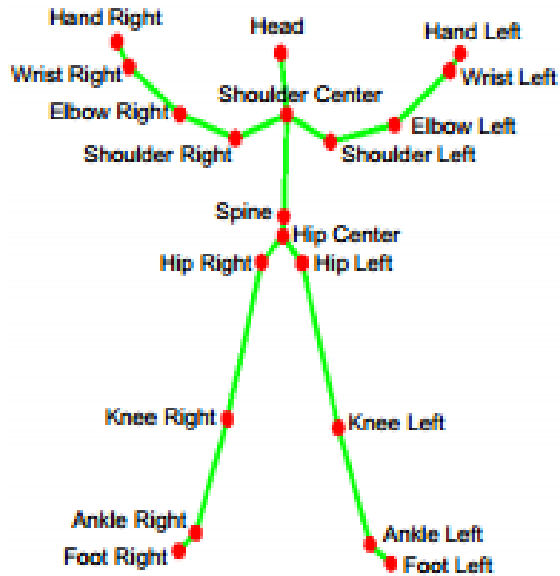
Kai aptinkamas galimas gestų šablonų galas, darbo kelias  $W$  gali būti rastas per atgalinio sekimo minimalių kelių nuo  $M(m, k)$  iki  $M(0, z)$  kai  $z$  laiko akimirka esanti  $Q$ , kur gestas prasideda. [20]

Kadangi yra naudojamas Microsoft Kinect sensorius gestų atpažinimui, Kinect SDK seka 3D koordinatas iš 20 kūno jungčių-sąnarių kaip pavaizduota 19 pav. realiu laiku 30 kadrų per sekundę. Kadangi mašinos mokymosi algoritmas naudoja gylio vaizdus numatyti sąnarių poziciją, skeleto modelis yra gana atsparus spalvoms, tekstūroms ir fonui. Rankų gestams atlikti reikia 6 iš 20 sąnarių:

- Kairė ranka
- Dešinė ranka
- Kairysis riešas
- Dešinysis riešas
- Kairė alkūnė
- Dešinė alkūnė

Savybės vektorius susideda iš 3D vektorių koordinačių iš šių 6 sąnarių ir tai yra 18 matmenų kaip parodyta:

$$f_n = [X_1, Y_1, Z_1, X_2, Y_2, Z_2, \dots, X_6, Y_6, Z_6],$$



19 pav. Kinect sąnariai

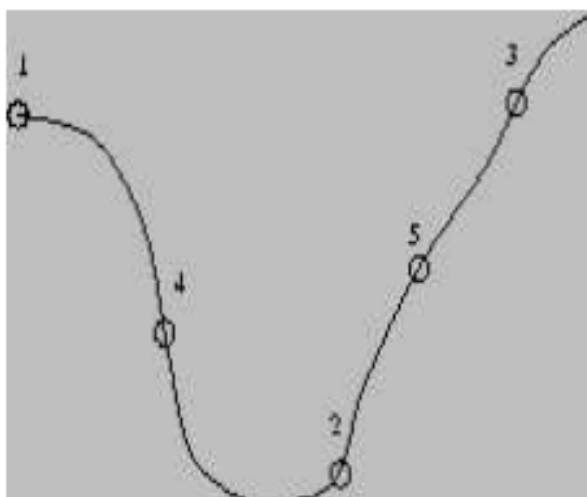
kur  $n$  yra skeleto indeksas kadro metu  $t_n$ . Gestų seka sujungta iš  $N$  kaip savybių vektoriai.

Po  $N$  savybių vektoriai sujungiami sukurti gestų seką jie apdorojami prieš DTW sąnaudų apskaičiavimą. Pirminis apdorojimo etapas pašalina variacijas savybių vektoriuose dėl asmens dydžio ar jo pozicijos kameros matymo lauke. Pirmia visi savybių vektoriai yra normalizuojami pagal atstumą tarp kairiojo ir dešiniojo pečių, kad būtų atsižvelgta variacijų dėl asmens dydžio. Antra normalizacija apima atimant pečių centrą iš visų elementų  $f_n$ , kuris sudarytas atvejams kur vartotojas nėra centre gylis vaizde [17].

#### 4.4. Golden Section Search algoritmas

Golden Section Search algoritmas yra panašus į perpjovimo metodą, kuris naudojamas siekiant aptikti funkcijų šaknis vienoje dimensijoje. Šaknis turėtų būti apibrėžta intervale  $(a,b)$  ir kai bandoma surasti tarpinį tašką  $x$  tarp  $a$  ir  $b$ , intervalas yra sumažinamas  $(a,x)$  ir  $(x,b)$ . Šis procesas kartojamas tol, kol intervalo režiiai yra priimtina maži. Golden Section Search algoritme mes bandomė surasti minimumą vietoj šaknies. Šaknis yra sujungta tarp  $a$  poros taškų  $(a,b)$  kai funkcijos  $f$  reikšmės šiuose taškuose  $a$  ir  $b$  turi priešingus ženklus. Tuo tarpu mažiausia susieta yra trijų taškų  $a,b,c$  tokiu būdu, kad  $a < b < c$  arba  $c < b < a$  ir  $f(b)$  yra mažiau negu abu  $f(a)$  ir  $f(b)$ . Dabar žinome kad mažiausia reikšmė tarp  $a$  ir  $c$ .

Analogas pjūviui yra pasirinkti tašką tarp  $a$  ir  $b$  arba tarp  $b$  ir  $c$ . Tarkime pasirinkome pastarąjį  $b$  ir  $c$ . Tada mes įvertiname  $f(x)$ . Jei  $f(b) < f(x)$ , tada naujas sugretinimas iš trijų taškų yra  $(a,b,x)$  kitu atveju naujas sugretinimo intervalas yra iš trijų taškų  $(b,x,c)$ . Mes tęsiame sugretinimą tol kol atstumas tarp dviejų išorinių taškų iš trejeto yra toleruotina mažas. 20 pav. vaizduoja kaip mes ieškome funkcijos minimumą. Šis metodas neturi jokios informacijos apie funkcijos išvestines. Jei tokia informaciją mes turime, mes galime ją naudoti nustatant naujus taškus  $x$  mūsų algoritme [21].



**20 pav.** Golden section search su pradiniais rėžiais (1,2,3) tampa (4,2,3) (4,2,5)....

## 5. EKSPERIMENTINĖ DALIS

### 5.1. Įžanga

Šiame skyriuje yra apžvelgiama Kinect įrenginiui skirtų gestų atpažinimo algoritmų tyrimo eksperimentų analizė. Pagrindinė charakteristika, pagal kurią yra vertinami 4 skyriuje apžvelgti algoritmai tai gestų atpažinimo tikslumas. Kadangi sistema buvo kurta pritaikant tik vieną iš trijų algoritmų, taigi eksperimentų rezultatai palyginimui yra naudojami, kurie yra gauti kitų asmenų. Norint kad tyrimas būtų tikslesnis, tikslinga yra bandyti atpažinti tą patį gestą keletą kartų, šiuo atveju po 10 kartų ta patį gestą ir tada procentaliai įvertinsime gestų atpažinimo procentą.

### 5.2. Tyrimo metodika

Norint atlikti eksperimentinį tyrimą tikslinga sudaryti tyrimo metodiką, eksperimento metu yra atsižvelgta į tai ar galima gauti prieigą prie sistemų, kuriuose būtų realizuoti mūsų pasirinkti algoritmai, jei tai nebus galima pasiekti bus remtasi kitų žmonių gautais eksperimentiniais rezultatais. Sistemų kuriuos yra realizuotos su mūsų algoritmais tyrimas vyks, tokiais žingsniais, norint įsitikinti ar sistema teisingai atpažįsta rodomus gestus, kiekvienas gestas yra kartojamas po 10 kartų ir tiriami rezultatai kiekvienu atveju ir gautų eksperimentinių rezultatų apibendrinimas padaromas apibendrinimas.

### 5.3. Hidden Markov Models algoritmo eksperimentinis tyrimas

#### 5.3.1. Įvadas

Kadangi kaip minėta 5.1. skyriuje, kad sistema buvo pritaikyta naudoti tik vieną algoritmą, taigi mums teks remtis kitų žmonių atliktais eksperimentiniais tyrimais apie Kinect įrenginiui skirtų gestų atpažinimą. Hidden Markov Models gestų atpažinimo algoritmas šiuo atveju buvo įgyvendintas matlab aplinkoje. Norint atpažinti gestą pirma reikia algoritmą apmokyti tai padaryti, sekančiame žingsnyje mes jau galime tirti gestų atpažinimą. Šiuo atveju apmokymų ir testavimo duomenys yra CSV faile x,y ir z koordinatės iš daugybės rankų gestų, bet šie duomenys nėra identiški. Taipogi jie buvo kuriami naudojant OpenNI SDK. Autorius pabrėžia kad jis duomenų rinkinius sukūrė du kart sėdėdamas prieš Kinect įrenginį ir gestikuliuodamas. Šiuo konkrečiu atveju yra vykdomas apskritimo gesto atpažinimas

#### 5.3.2. Eksperimentinė dalis

Algoritmas yra apmokomas 10 sekų, kurių maksimalus ilgis 60 iš alfabeto kurio dydis yra 8. Taipogi naudojamas Hidden Markov Models su 12 paslėptų būsenų.

```
*****  
Training 10 sequences of maximum length 60 from an alphabet of size 8  
HMM with 12 hidden states  
*****
```

```
cycle 1 log likelihood = -1248.538144  
cycle 2 log likelihood = -982.629625  
cycle 3 log likelihood = -732.009407  
cycle 4 log likelihood = -696.890070  
cycle 5 log likelihood = -689.663383  
cycle 6 log likelihood = -685.441817
```

```

cycle 7 log likelihood = -674.623979
cycle 8 log likelihood = -598.479502
cycle 9 log likelihood = -530.171032
cycle 10 log likelihood = -522.078143
cycle 11 log likelihood = -521.170559
cycle 12 log likelihood = -520.974759
cycle 13 log likelihood = -520.894473
cycle 14 log likelihood = -520.851012
cycle 15 log likelihood = -520.825580
cycle 16 log likelihood = -520.810292
cycle 17 log likelihood = -520.800933
cycle 18 log likelihood = -520.795098
cycle 19 log likelihood = -520.791392
end

```

Žemiau pateikiami eksperimento rezultatai, kaip minėta 4.1.1 naudojamas HMM su ribine reikšme, iš rezultatų matyti kad atpažinimas yra 100 % :

```

*****
Testing 10 sequences for a log likelihood greater than -104.157798
*****

Log likelihood: -48.890827 > -104.157798 (threshold) -- FOUND circle GESTURE!
Log likelihood: -49.721725 > -104.157798 (threshold) -- FOUND circle GESTURE!
Log likelihood: -49.795951 > -104.157798 (threshold) -- FOUND circle GESTURE!
Log likelihood: -51.037861 > -104.157798 (threshold) -- FOUND circle GESTURE!
Log likelihood: -53.615149 > -104.157798 (threshold) -- FOUND circle GESTURE!
Log likelihood: -49.848664 > -104.157798 (threshold) -- FOUND circle GESTURE!
Log likelihood: -46.062106 > -104.157798 (threshold) -- FOUND circle GESTURE!
Log likelihood: -47.221862 > -104.157798 (threshold) -- FOUND circle GESTURE!
Log likelihood: -47.344446 > -104.157798 (threshold) -- FOUND circle GESTURE!
Log likelihood: -44.769018 > -104.157798 (threshold) -- FOUND circle GESTURE!
Recognition success rate: 100.000000 percent

```

### 5.3.3. Apibendrinimas

Rezultatus pateiksime lentelės pavidalu, kurioje yra nurodomas sekos numeris ir ar pavyko atpažinti gestą ar ne. Iš eksperimentų rezultatų matome, kad gestų atpažinimas pavyko visu 100 % buvo rasti visi 10 ieškoti apskritimo gestai. Kadangi mes jau naudojames pateiktais eksperimento rezultatais, negali visu 100 % garantuoti, kad jie yra tikslūs ir atitinkantys realybę.

**2 lentelė** Apskritimo gesto atpažinimas

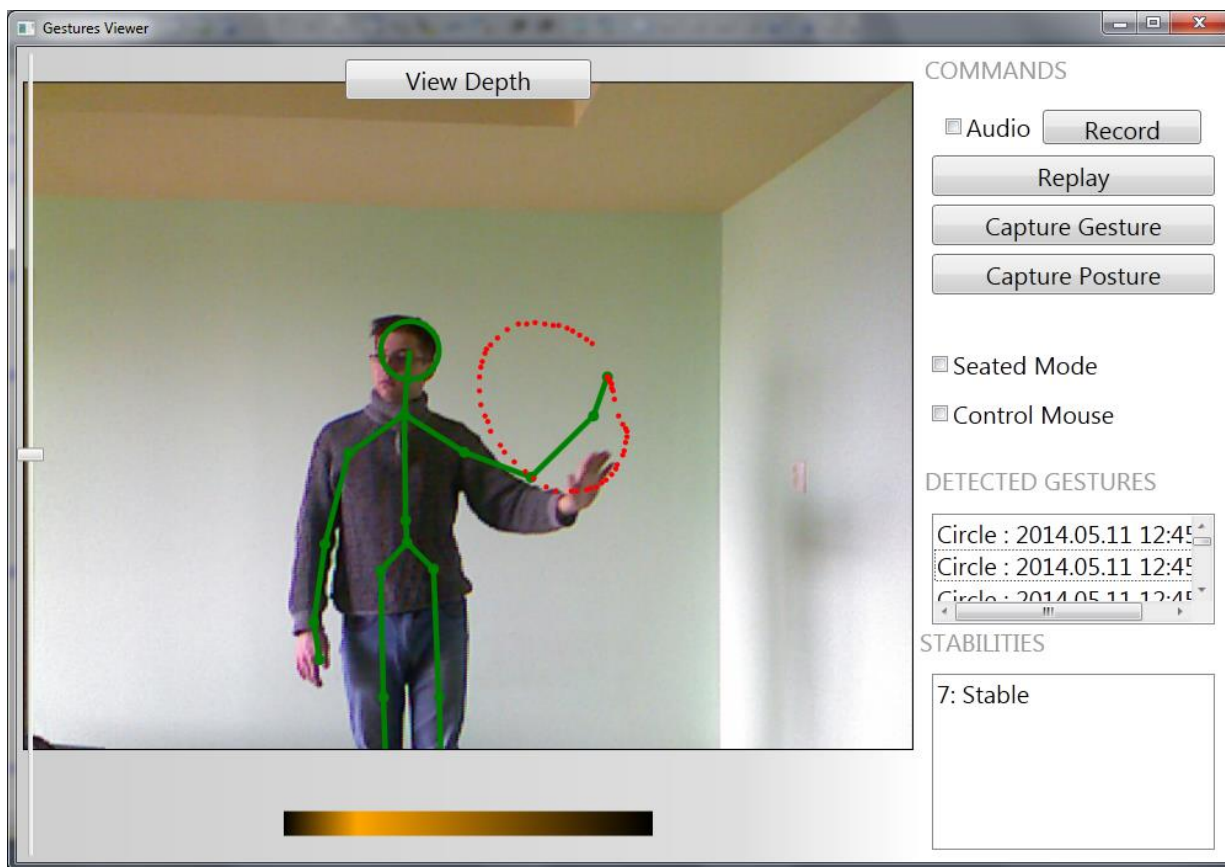
Sekos skaičius	Ar gestas (apskritimas) buvo atpažintas
1	Taip
2	Taip
3	Taip
4	Taip
5	Taip
6	Taip
7	Taip
8	Taip
9	Taip
10	Taip

## 5.4. Golden section Search algoritmo eksperimentinis tyrimas

### 5.4.1. Įvadas

Šis algoritmas buvo įgyvendintas mūsų sistemoje, 21 pav. priežastis kodėl jis buvo pasirinktas buvo remtasi kitų žmonių nuomone apie jo įgyvendinimo lengvumą, patikimumą ir greitą veikimą. Taipogi algoritmo realizavimas mums leidžia patiems nustatyti kokių tikslumu yra ieškomas norimo gesto šablonas, naudojant panašumo koeficiento ribinę reikšmę. Kadangi kaip minėta anksčiau norint atpažinti gestą pirmas žingsnis, kurį reikia atlikti tai apmokyti algoritmą atpažinti tą gestą šiuo atveju kaip ir Hidden Markov Models algoritmo pavyzdžiu algoritmas yra apmokytas atlikti apskritimo atpažinimą. Mes netikrinsime rankų mostų į kairę ar dešinę aptikimo, nes šiais atvejais algoritmas nieko neskaičiuoja, tiesiog yra tikrinamos rankų sąnarių pozicijos ir tada sistema nusprendžia ar tai mostas ar ne.

Kadangi algoritmas grąžina atpažinimo koeficientą tarp 0 ir 1, ir lyginama ar ji yra aukštesnė už ribinę reikšmę kurią mes esame nustatę. Kadangi labai mažą panašumo koeficiento ribinę reikšmę imti netikslinga, nes būtų didelė paklaida gestų atpažinimą pradėsime, kai jo pradinė reikšmė yra 0.6



21 pav. Programos langas

### 5.4.2. Eksperimentinė dalis

3. lentelė Apskritimo gesto atpažinimas kai panašumo koeficiento ribinė reikšmė 0.6

Sekos skaičius	Ar gestas (apskritimas) buvo atpažintas
1	Taip
2	Taip
3	Taip
4	Taip
5	Taip
6	Taip
7	Taip
8	Taip
9	Taip
10	Taip

4. lentelė Apskritimo gesto atpažinimas kai panašumo koeficiento ribinė reikšmė 0.7

Sekos skaičius	Ar gestas (apskritimas) buvo atpažintas
1	Taip
2	Taip
3	Taip
4	Taip
5	Taip
6	Taip
7	Taip
8	Taip
9	Taip
10	Taip

5. lentelė Apskritimo gesto atpažinimas kai panašumo koeficiento ribinė reikšmė 0.8

Sekos skaičius	Ar gestas (apskritimas) buvo atpažintas
1	Taip
2	Taip
3	Taip
4	Taip
5	Taip
6	Taip
7	Taip
8	Taip
9	Ne
10	Taip

**6. lentelė** Apskritimo gesto atpažinimas kai panašumo koeficiento ribinė reikšmė 0.9

Sekos skaičius	Ar gestas (apskritimas) buvo atpažintas
1	Ne
2	Taip
3	Taip
4	Taip
5	Taip
6	Taip
7	Taip
8	Ne
9	Taip
10	Taip

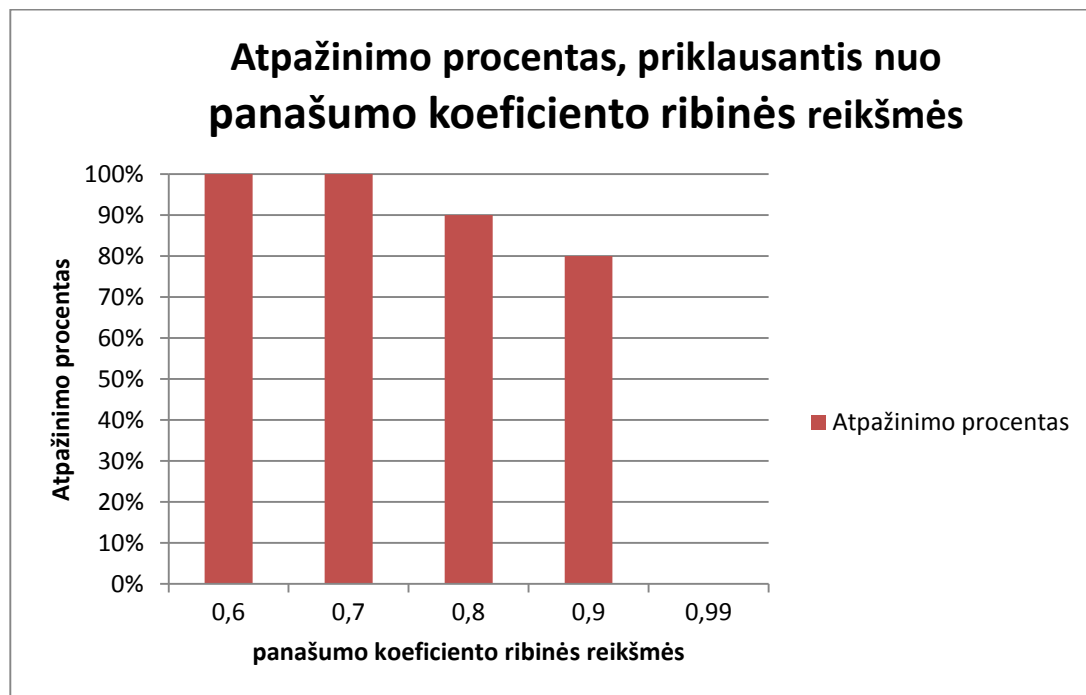
**7. lentelė** Apskritimo gesto atpažinimas kai panašumo koeficiento ribinė reikšmė 0.99

Sekos skaičius	Ar gestas (apskritimas) buvo atpažintas
1	Ne
2	Ne
3	Ne
4	Ne
5	Ne
6	Ne
7	Ne
8	Ne
9	Ne
10	Ne



### 5.4.3. Apibendrinimas

Galime pavaizduoti grafiškai, 22 pav. koks atpažinimo procentas yra prie tam tikrų ribinių reikšmių:



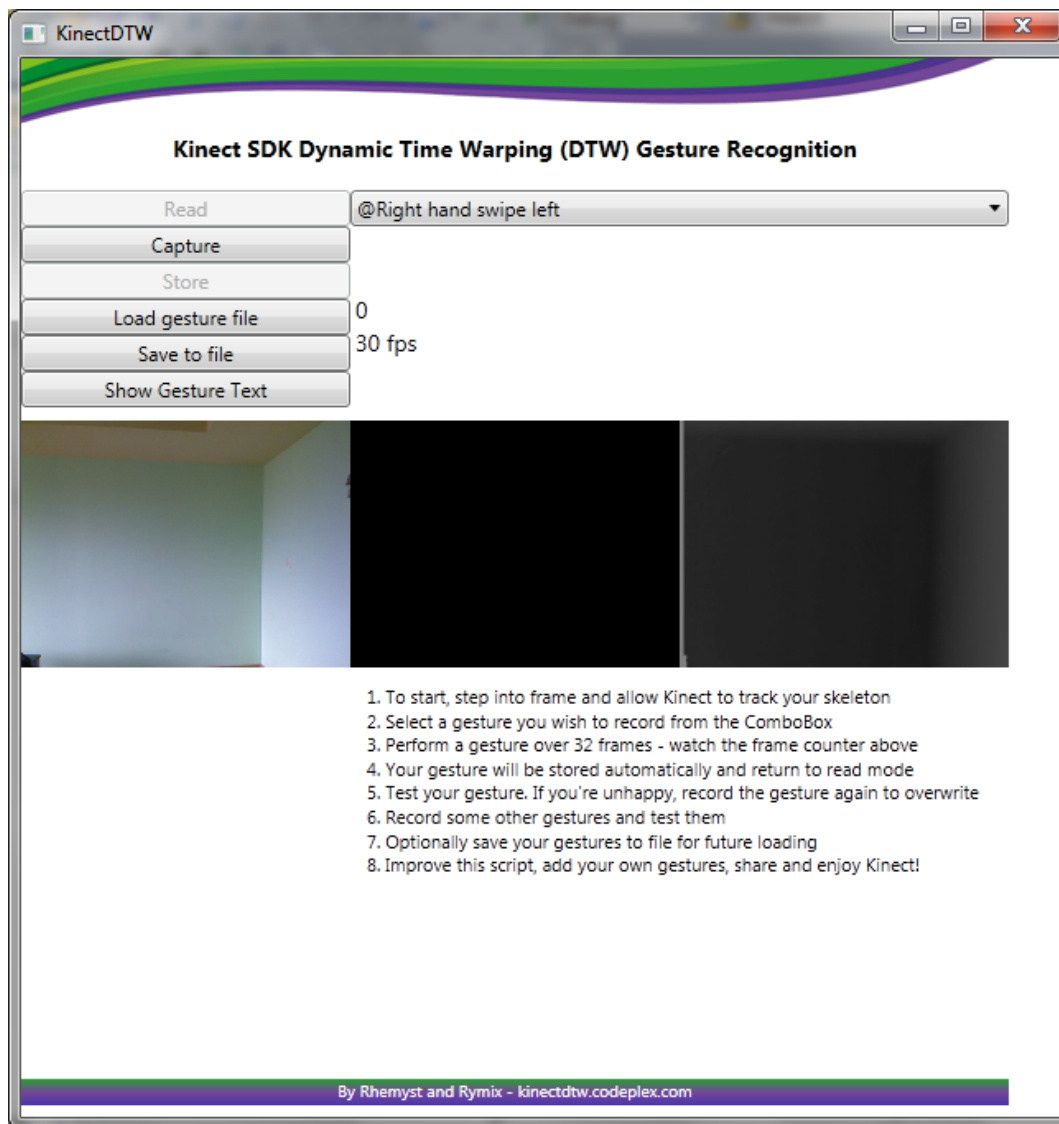
22 pav. Atpažinimo procentas priklausantis nuo panašumo koeficiento ribinių reikšmių

Paskutiniu atveju, kai ribinė reikšmė yra 0,99 mūsų algoritmas neatpažįsta apskritimo gesto, nes jis turi būti atliktas tobulai, tai buvo bandyta daryti daugiau nei 10 kartų, bet pasiekti kad apskritimo gestas būtų atpažintas nepavyko. Sistema padedama mūsų tiriamo algoritmo sėkmingai atpažindavo apskritimo gestą, taipogi lyginant su sekančiu algoritmu apskritimo dydį rankomis buvo galima keisti darant su mažu skersmeniu ar su labai dideliu, rezultatai buvo gaunami tie patys.

## 5.5. Dynamic Time warping algoritmo eksperimentinė dalis

### 5.5.1. Įvadas

Šis algoritmas taipogi nebuvo realizuotas mūsų sistemoje, kaip ir Hidden Markov Models algoritmas, bet buvo rasta programa kuri jau yra realizuota ir joje panaudotas šis algoritmas, ši sistema yra parašyta naudojant dar 2011 metais išleistą Kinect SDK Beta 2.0 versiją. taigi lyginant su realizuota sistema dalis karkaso funkcijų yra pakeistos naujomis. Sistemos išskirtinė savybė, kad žmogus turi apmokytį algoritmą atpažinti tam tikrus gestus, kuriuos galima pasirinkti iš išsiskleidžiančio meniu, taipogi būtina sąlyga, kad judesys būtų atliktas per 32 kadrus per sekundę, taigi gestus reikia atlikti šiek tiek lėčiau nei įprastai. Taipogi, kai žmogus pats nurodo judesį išskyla netikslumas, kitam žmogui gali nesisekti Algoritmą tyrinėjime tokiu pačiu principu kaip ir ankstesnius, tam tikrus gestus atlikinėjime po 10 kartų ir žiūrėsime kaip sistema padedama algoritmo juos atpažįsta 23 pav. matomas pagrindinis sistemos langas.



**23 pav.** Sistemos vaizdas su Dynamic Time Warping

### 5.5.2. Eksperimentinė dalis

**8. lentelė** Dešinnės rankos mostas į kairę.

Sekos skaičius	Ar gestas (apskritimas) buvo atpažintas
1	Taip
2	Taip
3	Taip
4	Taip
5	Taip
6	Taip
7	Taip
8	Taip
9	Taip
10	Taip

**9. lentelė** Pamojavimas dešinė ranka.

Sekos skaičius	Ar gestas (apskritimas) buvo atpažintas
1	Taip
2	Taip
3	Taip
4	Taip
5	Taip
6	Taip
7	Taip
8	Taip
9	Taip
10	Taip

**10. lentelė** Abiejų rankų gesto pritraukimas.

Sekos skaičius	Ar gestas (apskritimas) buvo atpažintas
1	Taip
2	Taip
3	Taip
4	Taip
5	Taip
6	Taip
7	Taip
8	Taip
9	Taip
10	Taip

**11. lentelė** Kairės rankos nuleidimas į apačią.

Sekos skaičius	Ar gestas (apskritimas) buvo atpažintas
1	Taip
2	Taip
3	Taip
4	Taip
5	Taip
6	Taip
7	Taip
8	Taip
9	Taip
10	Taip

### **5.5.3. Apibendrinimas**

Apibendrinant šio algoritmo eksperimentinį tyrimą galima pasakyti, kad jis tiksliai atpažįsta gestus, kuriuos sistema yra apmokyta atpažinti. Tačiau, visas atpažinimas vyksta labai identiškai, gestai turi būti rodomi identiškai tai kaip buvo apmokyti, jei padidiname tarpus tarp rankų, ar darome ilgesnius tokius pačius gestus sistema jų neatpažįsta. Bet šiuo atveju mes vertiname ne sistemą, o algoritmo veikimą taigi galima teigti, kad algoritmas atpažįsta gestus tinkamai.

## 6. IŠVADOS

1. Atlikus analizę išanalizavus literatūrą, algoritmus bei jau esamus sprendimus pasaulyje, daroma išvada, kad gestų atpažinimas nėra lengva užduotis, nes yra nevienareikšmiška problema, arba, tiksliau, ne taip lengva sukurti veiksmingus euristinius ar mašinių mokymosi algoritmus, kurie gali suprasti, koks gestas panaudotas.
2. Kuriant sistemą, pasiūlyta gestų atpažinimui naudoti Golden Section Search algoritmą, kuris pasižymi šiomis savybėmis:
  - greitai veikiantis.
  - lengvai realizuojamas.
  - galima koreguoti panašumo koeficiento ribinę reikšmę nuo 0 iki 1, kuri nusako kaip tiksliai atpažįstami gestai.
3. Suprojektuota ir realizuota sistema, įgyvendinanti pasiūlytą algoritmą ir leidžianti su Kinect įrenginiu atpažinti žmogaus kūno gestus, ir duoti atgalinį ryšį.
4. Atliktas Kinect įrenginiui skirtų gestų atpažinimo algoritmo tyrimas, buvo apžvelgta kaip šie algoritmai veikia, taipogi išnagrinėti matematiniai aspektai. Buvo išnagrinėti šie algoritmai:
  - Hidden Markov Models
  - Dynamic Time Warping
  - Golden Section Search
5. Atlikus pasirinktų algoritmų eksperimentinį tyrimą nustatyta, kad algoritmai gestus atpažino sėkmingai, esminis skirtumas kad šie trys algoritmai buvo įgyvendinti trijose skirtingose sistemose.
  - Mūsų sistemos algoritmas Golden Section Search, parodė tuos pačius rezultatus kaip kiti algoritmai, tuo atveju kai panašumo koeficiento ribinė reikšmė buvo 0,7 ir mažesnė. Priežastis kodėl prie aukštesnių ribinių reikšmių, atpažinimo procentas buvo mažesnis, ta kad gestas turėjo būti atliktas labai kruopščiai.
  - Sistema kurioje buvo realizuotas Dynamic Time Warping algoritmas judesius atpažinimo sėkmingai, tačiau judesius reikėjo atlikti identiška, tokius kokie buvo įrašyti, buvo bandoma didinti tarpus tarp rankų ar daryti platesnius judesius, tuo atveju sistema neatpažindavo judesio, nors jis buvo daromas toks pat tik su didesniais tarpais tarp rankų.
  - Gauti realizuotos sistemos su Hidden Markov Models, nepavyko taigi teko panaudoti kito asmens padarytais eksperimentiniais rezultatais, bet taipogi algoritmas atpažino gestus sėkmingai.
6. Norint detalesnio algoritmų eksperimentinio tyrimo, būtų galima tirti jų atpažinimo greitį, judesių kiekį per tam tikrą laiką ar panašias charakteristikas, bet šie darbai yra tolimesnių tyrimų objektas.

## 7. LITERATŪRA

1. Takahashi, D. (2009), «Microsoft games exec details how Project Natal was born», adresas: <http://venturebeat.com/2009/06/02/microsoft-games-executive-describes-origins-of-project-natal-game-controls/>, [Internetinis; žiūrėta 2012-11-25]
2. OpenNI (2011), OpenNI User Guide, OpenNI, version 3 ed., adresas: [http://www.openni.org/images/stories/pdf/OpenNI\\_UserGuide\\_v3.pdf](http://www.openni.org/images/stories/pdf/OpenNI_UserGuide_v3.pdf), [Internetinis; žiūrėta 2012-11-26].
3. Research, M. (2011), Programming Guide - Getting Started with the Kinect for Windows SDK Beta from Microsoft Research, Microsoft, beta 1 draft version 1.1 ed., adresas: [http://202.120.53.119/wordpress/wp-content/uploads/2011/06/ProgrammingGuide\\_KinectSDK.pdf](http://202.120.53.119/wordpress/wp-content/uploads/2011/06/ProgrammingGuide_KinectSDK.pdf), [Internetinis; žiūrėta 2012-11-30].
4. Ten, S. (2010), «How Kinect depth sensor works – stereo triangulation?», adresas: <http://mirror2image.wordpress.com/2010/11/30/how-kinect-works-stereo-triangulation/>, [Internetinis; žiūrėta 2012-11-28].
5. Khoshelham, K. (2011), «ACCURACY ANALYSIS OF KINECT DEPTH DATA», Tech. rep., ITC Faculty of Geo-information Science and Earth Observation, University of Twente., adresas: [http://www.isprs.org/proceedings/XXXVIII/5-W12/Papers/ls2011\\_submission\\_40.pdf](http://www.isprs.org/proceedings/XXXVIII/5-W12/Papers/ls2011_submission_40.pdf), [Internetinis; žiūrėta 2012-11-26].
6. Wikipedia(2011), «Natural user interface — Wikipedia, The Free Encyclopedia», adresas: [http://en.wikipedia.org/wiki/Natural\\_user\\_interface](http://en.wikipedia.org/wiki/Natural_user_interface), [Internetinis; žiūrėta 2012-12-02].
7. Valli, A. (2008), «Notes on Natural Interaction», Tech. rep., adresas: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.7674&rep=rep1&type=pdf>, [Internetinis; žiūrėta 2012-11-27].
8. Karam, M. and Schraefel, M. (2005), «A taxonomy of Gestures in Human Computer Interaction», Tech. rep., University of Southampton, adresas: <http://eprints.ecs.soton.ac.uk/11149/1/GestureTaxonomyJuly21.pdf>, [Internetinis; žiūrėta 2012-12-02].
9. Valli, A. (2008), «Notes on Natural Interaction», Tech. rep., adresas: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.7674&rep=rep1&type=pdf>, [Internetinis; žiūrėta 2012-12-02].
10. Hsu, H. J. (2011), «The Potential of Kinect as Interactive Educational Technology», in «2nd International Conference on Education and Management Technology», vol.13, pp. 334–338, IACSIT Press, Singapore, adresas: <http://www.ipedr.com/vol13/64-T10050.pdf>, [Internetinis; žiūrėta 2012-12-01].
11. Your Shape: Fitness Evolved 2012 – review, adresas: <http://www.guardian.co.uk/technology/gamesblog/2012/mar/05/your-shape-fitness-evolved-2012-review> [Internetinis; žiūrėta 2012-12-12].
12. «zumba-fitness-kinect-review» [interaktyvus]. [žiūrėta 2012-12-13]. Prieiga per internetą: <http://www.gamingtruth.com/2011/01/31/zumba-fitness-kinect-review/>
13. EA SPORTS Active 2». adresas: <http://www.ea.com/ea-sports-active-2> [Internetinis; žiūrėta 2012-12-14].
14. «Nike plus kinect training». adresas: [http://www.nike.com/us/en\\_us/c/training/nike-plus-kinect-training](http://www.nike.com/us/en_us/c/training/nike-plus-kinect-training) [Internetinis; žiūrėta 2012-11-15].
15. «Golden Section Search Method» adresas: [http://mathforcollege.com/nm/mws/gen/09opt/mws\\_gen\\_opt\\_txt\\_goldensearch.pdf](http://mathforcollege.com/nm/mws/gen/09opt/mws_gen_opt_txt_goldensearch.pdf), [Internetinis; žiūrėta 2014-05-01].
16. Jie Yang, Yangsheng Xu (1994) «Hidden Markov Model for Gesture Recognition», Tech. rep., Carnegie Mellon University adresas: [http://www.ri.cmu.edu/pub\\_files/pub3/yang\\_jie\\_1994\\_1/yang\\_jie\\_1994\\_1.pdf](http://www.ri.cmu.edu/pub_files/pub3/yang_jie_1994_1/yang_jie_1994_1.pdf), [Internetinis; žiūrėta 2014-05-01].

17. Sait Celebi, Ali S. Aydin, Talha T. Temiz and Tarik Arici, «Gesture Recognition Using Skeleton Data with Weighted Dynamic Time Warping» adresas: <http://users.utcluj.ro/~tmarita/IOC/C9-10/Doc/VISAPP2013.pdf>, [Internetinis; žiūrėta 2014-05-03].
18. Hyeon-Kyu Lee and Jin H. Kim (1999), «An HMM-Based Threshold Model Approach for Gesture Recognition» adresas: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.41.2014&rep=rep1&type=pdf>, [Internetinis; žiūrėta 2014-05-02].
19. Jonathan C. Hall (2011) «How to Do Gesture Recognition With Kinect Using Hidden Markov Models (HMMs)», adresas: <http://www.creativedistracted.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/> [Internetinis; žiūrėta 2014-05-05]
20. Miguel Angel Bautista, Antonio Hernandez-Vela, Victor Ponce, Xavier Perez-Sala, Xavier Baro, Oriol Pujol, Cecilio Angulo and Sergio Escalera «Probability-based Dynamic Time Warping for Gesture Recognition on RGB-D data» adresas: <http://www.maia.ub.es/~sergio/linked/icprdepthdtwgm.pdf>, [Internetinis; žiūrėta 2014-05-05].
21. Muthukumar B. Puranam (2005) University of Kentucky «Towards Full-Body Gesture Analysis and Recognition » adresas: [http://uknowledge.uky.edu/cgi/viewcontent.cgi?article=1229&context=gradschool\\_theses](http://uknowledge.uky.edu/cgi/viewcontent.cgi?article=1229&context=gradschool_theses), [Internetinis; žiūrėta 2014-05-02].

## **8. TERMINŲ IR SUTRUMPINIMŲ ŽODYNAS**

SDK – programinės įrangos kūrimo karkšas

HMM – Hidden Markov Models algoritmas