

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS**

Karolis Ryselis

**ŽMOGAUS POZŲ ATPAŽINIMO ALGORITMO IR JŲ
REALIZUOJANČIOS PROGRAMINĖS ĮRANGOS
SUKŪRIMAS BEI ALGORITMO TIKSLUMO TYRIMAS**

Baigiamasis magistro projektas

Vadovas

Doc. dr. Tomas Blažauskas

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ INŽINERIJOS KATEDRA

TVIRTINU
Katedros vedėjas
Doc. dr. Tomas Blažauskas
2017-05-25

**ŽMOGAUS POZŲ ATPAŽINIMO ALGORITMO IR JĮ
REALIZUOJANČIOS PROGRAMINĖS ĮRANGOS
SUKŪRIMAS BEI ALGORITMO TIKSLUMO TYRIMAS**

Baigiamasis magistro projektas

Programų sistemų inžinerija (kodas 621E16001)

Vadovas

Doc. dr. Tomas Blažauskas
2017-05-25

Recenzentas

Doc. dr. Rytis Maskeliūnas
2017-05-25

Projektą atliko

Karolis Ryselis
2017-05-25

KAUNAS, 2017



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Karolis Ryselis

(Studento vardas, pavardė)

Programų sistemų inžinerija (621E16001)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Pavadinimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 17 m. gegužės 25 d.
Kaunas

Patvirtinu, kad mano, **Karolio Ryselio**, baigiamasis projektas tema „Žmogaus pozų atpažinimo algoritmo ir jį realizuojančios programinės įrangos sukūrimas, tikslumo tyrimas ir pritaikymas jogos žaidime“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

TURINYS

1.	Ižanga.....	13
1.1.	Dokumento paskirtis.....	13
1.2.	Darbo tikslas.....	13
1.3.	Mokslinis naujumas.....	13
1.4.	Uždaviniai	13
2.	Analitinė dalis.....	14
2.1.	Probleminė sritis.....	14
2.2.	Žmogaus judesių sekimo problematika.....	15
2.3.	Microsoft „Kinect“ trūkumai ir ribojimai	16
2.4.	Egzistuojantys sprendimai.....	20
2.5.	Judesių ir pozicijų palyginimo algoritmų analizė.....	22
2.5.1.	Hausdorfo atstumas	22
2.5.2.	Kendalio figūrų teorija ir apibendrinta Prokrusto analizė	24
2.5.3.	Figūrų sekų palyginimas.....	26
2.5.4.	Erdvėlaikinių šablonų lyginimo algoritmas ir nuožulnos atstumas.....	26
2.5.5.	Algoritmų apibendrinimas	28
2.6.	Esami Hausdorfo atstumo skaičiavimo metodai	29
2.6.1.	Yaniro Taflevo metodas	29
2.6.2.	„Elastic Search“ metodas.....	30
2.6.3.	Princetono universiteto metodas.....	30
2.7.	Analizės išvados	30
3.	Projektinė dalis	32
3.1.	Reikalavimai.....	32
3.1.1.	Veiklos kontekstas	32
3.1.2.	Sistemos panaudos atvejų modelis	33
3.1.3.	Reikalavimai sistemai	36
3.2.	Architektūra.....	38

3.2.1.	Architektūros tikslai ir apribojimai.....	38
3.2.2.	Sistemos statinis vaizdas.....	38
3.3.	Sistemos dinaminis vaizdas.....	42
3.4.	Sistemos išdėstymo vaizdas	46
4.	Tyrimo dalis.....	47
4.1.	Prokrusto analizės optimizacija.....	47
4.2.	Hausdorfo atstumo skaičiavimo optimizacija	48
5.	Ekspirimentinė dalis.....	50
5.1.	Ekspirimentinio tyrimo tikslas.....	50
5.2.	Ekspirimentinio tyrimo aprašymas	50
5.2.1.	Euristinės Prokrusto transformacijos įtakos Hausdorfo metrikos tikslumui įvertinimas	50
5.2.2.	Euristinės Prokrusto transformacijos koeficientų tikslumo įvertinimas.....	51
5.2.3.	Siūlomo Hausdorfo algoritmo veikimo greičio priklausomybės nuo tiriamo kadro dydžio tyrimas.....	51
5.2.4.	Hausdorfo atstumo skaičiavimo algoritmų greitaveikos palyginimas.....	51
5.3.	Ekspirimentinio tyrimo rezultatai	52
5.3.1.	Euristinės Prokrusto transformacijos įtakos Hausdorfo metrikos tikslumo įvertinimo rezultatai	52
5.3.2.	Euristinės Prokrusto transformacijos koeficientų nuokrypio įvertinimo rezultatai	57
5.3.3.	Siūlomo Hausdorfo algoritmo veikimo greičio priklausomybės nuo tiriamo kadro dydžio tyrimo rezultatai	58
5.3.4.	Hausdorfo atstumo skaičiavimo algoritmų greitaveikos palyginimo rezultatai	59
5.4.	Ekspirimentinio tyrimo išvados.....	60
6.	Išvados	61
7.	Literatūros sąrašas	62
8.	Terminų ir santrumpų žodynas	64
9.	Priedai	65
9.1.	Sistemos reikalavimai.....	65

9.1.1.	Apribojimai sistemai.....	65
9.1.2.	Funkciniai reikalavimai	66
9.1.3.	Nefunkciniai reikalavimai	72
9.2.	Detalios sistemos klasių diagramos.....	74
9.3.	Žmogaus sąnarių padėties nustatymo naudojant kelis „Kinect“ įrenginius algoritmo sudarymas ir tyrimas	78
•	Įžanga	78
•	Sąnarių koordinacių nustatymas naudojant kelis sensorius	78
•	Sąnarių koordinacių skaičiavimo algoritmas	79
•	Sąnarių koordinacių nustatymo tikslumo tyrimas	80
•	Išvados.....	81
•	Literatūra	81
9.4.	Efektyvus Hausdorfo atstumo metrikos skaičiavimo algoritmas žmogaus padėties palyginimui su šablonu ir jo greitaveikos ir tikslumo tyrimas.....	82
I.	Įžanga	82
II.	Hausdorfo atstumo skaičiavimo būdai.....	82
A.	Yanir Taflev siūlomas skaičiavimo būdas	82
B.	Elastic Search siūlomas skaičiavimo būdas.....	82
C.	Princetono universiteto siūlomas skaičiavimo būdas	83
III.	Siūlomas Hausdorfo atstumo skaičiavimo algoritmas.....	83
A.	Siūlomas optimalios transformacijos radimo algoritmas.....	83
B.	Siūlomas Hausdorfo atstumo skaičiavimo algoritmas.....	83
IV.	Tyrimo metodas	84
V.	Tyrimo rezultatai	84
VI.	Išvados	85
VII.	Literatūros sąrašas.....	85
9.5.	Nestandartinių žmogaus kūno pozicijų atpažinimo tikslumo naudojant „Kinect 2.0“ jutiklius tyrimas.....	86

Nestandartinių žmogaus kūno pozicijų atpažinimo tikslumo naudojant „Kinect 2.0“ jutiklius tyrimas	86
Įžanga	86
Tyrimo būdas	86
Tyrimo algoritmas	87
Tyrimo rezultatai	88
Išvados	90
Literatūros sąrašas	90
Nonstandard human body position tracking using „Kinect 2.0“ sensor accuracy research	90

LENTELIŲ SĄRAŠAS

Lentelė 2.1. Žaidimų, skirtų „Kinect“, palyginimas.....	20
Lentelė 2.2. Įrankių, skirtų „Kinect“ duomenų įrašymui, palyginimas.....	21
Lentelė 2.3. Nagrinėjamų algoritmų apibendrinimas.....	29

PAVEIKSLĖLIŲ SĄRAŠAS

2.1 pav. „Kinect 2“ įrenginys ir jo jutikliai	17
2.2. pav. "Kinect" atpažįstamo žmogaus skeleto struktūra.....	18
2.3. pav. "Kinect" jutiklių išdėstymo schema atliekant judesių atpažinimą keliais "Kinect" įrenginiais.....	19
2.4 pav. Pavyzdinis kamerų išdėstymas filmuojant pradinį siluetą erdvėlaikiniam figūrų palyginimui	28
3.1 pav. Sistemos konteksto diagrama	32
3.2 pav. Judesių sekimo komponento panaudos atvejų modelis	33
3.3 pav. Judesių konvertavimo ir redagavimo įrankio panaudos atvejų modelis.....	34
3.4 pav. Jogos žaidimo dalies, susijusios su judesių sekimu, panaudos atvejų modelis	35
3.5 pav. Judesių sekimo komponento ir jogos žaidimo panaudos atvejų modelių sąsajos diagrama	36
3.6 pav. Paketų diagrama.....	39
3.7 pav. Paketo „MotionTracker“ klasių diagrama	40
3.8 pav. Paketo „KinectController“ klasių diagrama	41
3.9 pav. Paketo „MotionConvertTool“ klasių diagrama	42
3.10 pav. Judesių konvertavimo veiklos diagrama.....	43
3.11 pav. Judesių tvarkymo veiklos diagrama.....	44
3.12 pav. Jogos žaidimo veiklos diagrama	45
3.13 pav. Sistemos išdėstymo diagrama	46
5.1 pav. Hausdorfo atstumo paklaidų pasiskirstymas	53
5.2 pav. Hausdorfo atstumo paklaidų pasiskirstymas, absoliutinės vertės.....	54
5.3 pav. Žmogaus siluetas standartinėje stovimoje padėtyje.....	54
5.4 pav. Šabloninis siluetas standartinėje stovimoje padėtyje.....	55
5.5 pav. Hausdorfo atstumo skaičiavimo paklaidų pasiskirstymas pagal dydį visame pratime ..	56
5.6 pav. Hausdorfo atstumo skaičiavimo paklaidų pasiskirstymas pagal dydį stovimose padėtyse	56
5.7 pav. Transformacijos koeficientų paklaidos viso pratimo metu.....	57
5.8 pav. Transformacijos koeficientų paklaidos stovimose padėtyse.....	58
5.9 pav. Hausdorfo atstumo skaičiavimo greitaveika.....	59
5.10 pav. Hausdorfo atstumo skaičiavimo algoritmų greitaveikos palyginimas	60
9.1 pav. Paketo „KinectController“ detali klasių diagrama	75
9.2 pav. Paketo „MotionConvertTool“ detali klasių diagrama	76
9.3 pav. Paketo „MotionTracker“ detali klasių diagrama	77

Pav. 9.4. Algoritmų greitaveikos palyginimas (tolimi siluetai).....	84
Pav. 9.5. Algoritmų greitaveikos palyginimas (artimi siluetai).....	85

SANTRAUKA

Pastaruoju metu populiarėja sveika gyvensena, ja užsiima vis daugiau žmonių [1]. Vienas iš pagrindinių užsiėmimų, prisidedančių prie šio gyvenimo būdo – joga. Ji padeda gerinti žmogaus fizinę ir mentalinę sveikatą. Šiuo metu joga gali užsiimti visi, tačiau norint tai daryti teisingai, reikia profesionalo pagalbos. Šiuo metu ji teikiama jogos centruose. Žmonės, kurie gyvena toli nuo jogos centrų arba dirba tuo pačiu laiku, kaip ir jogos centrai, negali dalyvauti jogos treniruotėse. Jie netenka galimybės gauti profesionalų jogos mokymą, o be jo jogos praktika yra tik mėgėjiška. Taip pat žmonės nori užsiimti joga ir savarankiškai namuose. Šiuo metu jie jogos konsultacijų namuose gauti negali. Šio darbo metu siekiama sukurti judesių sekimo komponentą virtualiam jogos treneriui.

Kuriamo judesių sekimo komponento pagrindinė dalis – judesių sekimo algoritmas. Jis remiasi Hausdorfo atstumo ir Prokrusto analizės principais ir efektyviai lygina dvi žmogaus pozų sekas. Darbo metu remiamasi prielaidomis, kylančiomis iš taikymo srities – jogos. Jų pagalba esami algoritmai optimizuojami, taikomi apytiksliai metodai, gerai aproksimuojantys matuojamas metrikas sekant žmogaus judesius. Programinė įranga, realizuojanti šiuos algoritmus, sukurta darbo metu ir įvertintas jos efektyvumas bei tikslumas, ji lyginama su kitais egzistuojančiais sprendimais.

Sukurtas judesių sekimo komponentas gali būti taikomas ir kitose srityse, kur reikalingas tikslus judesių atlikimas, pvz., reabilitacijai.

SUMMARY

Healthy life style is getting more and more popular. More and more people embrace it [1]. One of the main activities contributing towards this life style is yoga. It improves human's physical and mental health. Nowadays everyone can practice yoga, but assistance from a professional trainer is required to do it correctly. The assistance could be provided in a yoga center. People who live in remote areas or work at the same time as yoga centers cannot take part in those yoga trainings. They lose the ability to get professional yoga training. Yoga could be done incorrectly without a trainer and therefore it is only at amateur level. People want to practice yoga at home as well. A professional trainer is not available there. The goal of this work is to create a motion tracking component for a virtual yoga trainer.

The main piece of motion tracking component is motion tracking algorithm. It is based on principles of Hausdorff distance and Procrustes analysis and compares two sequences of human poses efficiently. Assumptions based on yoga applications are made in this work. They allow optimizing existing algorithms, applying approximate methods that fit well to measure metrics for human movements. Software that implements these algorithms is created in this work and its efficiency and precision are assessed, algorithm implementation is compared to other existing solutions.

The motion tracking component could be applied in other areas – health promotion, rehabilitation and other areas that require correct completion of a physical action.

1. ĮŽANGA

1.1. Dokumento paskirtis

Šio dokumento paskirtis – pristatyti žmogaus judesių sekimo problematiką naudojant stebinius jutiklius, egzistuojančius algoritmus ir programinę įrangą bei jos sąsają su jogos sportu. Taip pat šiame dokumente aprašomas magistro studijų metu sukurtas projektas, pristatomas dviejų metrikų, padedančių sekti žmogaus judesius, efektyvaus apskaičiavimo metodas.

1.2. Darbo tikslas

Šio darbo tikslas – sukurti algoritmą, leidžiantį sekti žmogaus judesius nepriklausomai nuo žmogaus padėties ar pozicijos ir ištirti jo efektyvumą bei tikslumą.

1.3. Mokslinis naujumas

Šiame darbe pateikiamas toks mokslinis naujumas:

- Sukurtas efektyvus algoritmas didelės raiškos žmogaus siluėtų palyginimui;
- Atlikta sukurtų Hausdorfo atstumo ir Prokrusto analizės skaičiavimo algoritmų veikimo greičio analizė.

1.4. Uždaviniai

Darbo metu keliami tokie uždaviniai:

- Ištirti esamus judesių sekimo algoritmus bei juos įvertinti;
- Sukurti algoritmą, kuris leistų palyginti dvi judesių sekas realiu laiku naudojant „Kinect“ jutiklį;
- Ištirti sukurto algoritmo veikimo savybes;
- Sukurtą algoritmą integruoti į jogos žaidimą bei sukurti pagalbines programas, kurias būtų galima paruošti judesių įrašus naudojimui jogos žaidime.

2. ANALITINĖ DALIS

2.1. Probleminė sritis

Pastaruoju metu populiarėja sveika gyvensena, ja užsiima vis daugiau žmonių. Vienas iš užsiėmimų, daugiausiai prisidedančių prie tokio gyvenimo būdo – joga. Ji padeda gerinti žmogaus fizinę ir mentalinę sveikatą. Šiuo metu joga gali užsiimti visi, tačiau norint tai daryti teisingai, reikia profesionalo pagalbos. Dažniausiai ji teikiama jogos centruose. Žmonės, kurie gyvena toli nuo jogos centrų arba dirba tuo pačiu laiku, kaip ir jogos centrai, negali dalyvauti jogos treniruotėse. Jie netenka galimybės gauti profesionalų jogos mokymą, o be jo jogos praktika yra tik mėgėjiška. Taip pat žmonės nori užsiimti joga ir savarankiškai namuose. Šiuo metu jogos konsultacijas namuose gauti yra sudėtinga.

Šio darbo metu siekiama sukurti virtualų jogos trenerį, kuris suteiktų profesionalaus jogos trenerio paslaugas namuose. Šiam treneriui yra reikalingas komponentas, kuris galėtų sekti vartotojo judesius ir užtikrinti, kad žmogus atlieka pratimą teisingai. Šioje srityje kompiuterizuotų sprendimų kol kas nėra daug, jie turi daug trūkumų, dėl to produktas aktualus. Sukurtas treneris galės būti tiek pramoga, t. y., žaidimas, tiek padės vartotojams tobulinti savo kūną ir atpalaiduoti bei išsivalyti mintis. Į žaidimą bus integruotas ir minčių aktyvumo sekimo komponentas, todėl produktas galės užtikrinti geriausią jogos patirtį. Joga ypač populiari Jungtinėse Amerikos Valstijose [2], todėl potencialių sistemos vartotojų yra daug.

Judesių sekimo komponentas galėtų būti panaudotas ir kitose srityse, kur reikalinga užtikrinti, kad vartotojas judesius atlieka teisingai, pvz., medicinoje, kai reikia atlikti pratimus po infarkto, insulto ir panašių ligų, taip pat šokių mokyme ir kitokio pobūdžio treniruotėse. Tai sudaro papildomą rinką kuriamam komponentui.

Populiarėjantys žmogaus kūno poziciją sekantys jutikliai daro judesiu pagrįstą vartotojo sąsają vis labiau prieinamą. Šiuo metu populiariausias tokio tipo įrenginys yra „Microsoft“ kompanijos sukurtas įrenginys „Kinect“. Jis žmogaus pozicijos atpažinimui naudoja infraraudonųjų spindulių ir standartinę vaizdo kamerą. Šiuo metu yra išleista antroji įrenginio versija. „Kinect“ yra suderinamas su „Windows“ operacine sistema ir žaidimų konsole „XBox One“. Šis įrenginys turi daug vartotojų, nes yra platinamas kartu su „XBox“ konsolėmis. Jo populiarumas padidina kuriamo komponento galimybes. Net jei vartotojas jo ir neturi, jis yra prieinamas daugumai žmonių, ypač Jungtinėse Amerikos Valstijose.

Kadangi sveika gyvensena vis populiarėja, o judesių jutikliai darosi vis labiau prieinami žmonėms, šis produktas turėtų likti perspektyvus ir ateityje. Jo perspektyvas didina ir didėjantis medicinos kompiuterizavimas.

Jogos sporte yra svarbu atlikti pratimus teisingai. Jogos pratimą (asaną) sudaro trys fazės:

1. Įėjimas į poziciją;
2. Pozicijos užfiksavimas;
3. Išėjimas iš pozicijos.

Kuriamas jogos žaidimas turės užtikrinti visų šių etapų atlikimo teisingumą. Tai bus pasiekta lyginant įrašytus trenerio judesius su žaidėjo daromais judesiais.

Problemos sprendimui keliami tokie reikalavimai ir apribojimai:

- Palyginti dvi judesių sekas;
- Nustatyti skirtumus tarp judesių;
- Vykdyti palyginimą realiu laiku (30 kadrų per sekundę);
- Apskaičiuoti įvertį judesio atitikimui;
- Palyginimą atlikti naudojant „Microsoft Kinect 2“ įrenginį;
- Palyginti įvairias pozas, ne tik stovimas.

Tokie reikalavimai keliami dėl taikymo jogos sričiai. Čia svarbu tiek tikslumas, tiek reakcijos laikas, tiek grįžtamasis ryšys vartotojui.

2.2. Žmogaus judesių sekimo problematika

Žmogaus judesių sekimas taikomas daugelyje sričių, pvz., virtualioje realybėje, žaidimuose, natūraliose vartotojo sąsajose, mokyme, sveikatinime, treniruotėse, robotikoje, gestų kalbos atpažinime [3]. Atsirandant naujiems judesių sekimo įrenginiams ir algoritmams reikėjo ir naujo, aukštesnio lygio, žmogaus kūno interpretavimo.

Viena iš tokių interpretavimo sričių – veiklos atpažinimas. Tokiam atpažinimui keliamas uždavinys: „iš duotos judesių sekos identifikuoti duomenyse atvaizduojamo subjekto atliekamus veiksmus“ [4]. Priklausomai nuo sudėtingumo veiklos skirstomos į kategorijas:

- Gestai (pasukti galvą dešinėn, pakelti koją);
- Veiksmai (eiti, mesti);
- Veiksmai su sąveika (žaisti su šuniu, žaisti futbolą).

Nors žmogui lengva identifikuoti visas šias veiklas, kol kas nėra technologijos, kuri tai galėtų padaryti patikimai ir efektyviai [4]. Išskiriami trys problemų aspektai:

- Žmogaus judesiai atvaizduojami į aukštą matmenų erdvę ir šablonų paieška joje tampa komplikauta;
- Konceptualiai panašių veiklų atlikimas subtiliai skiriasi priklausomai nuo žmogaus, atliekančio šią veiklą;

- Tradicinės vaizdo kameros gali analizuoti plokščią vaizdo informaciją ir yra jautrios apšvietimui.

Šioms problemoms spręsti naudojamos gylio kameros. Jos dažniausiai taikomos gestų ir paprastų veiksmų atpažinimui. Veiklų atpažinimas dažniausiai susideda iš trijų žingsnių:

- Esminių savybių radimas;
- Kvantavimas ir matmenų sumažinimas;
- Klasifikacija.

Atpažinimo algoritmai skirstomi į tokias kategorijas:

- Gylio žemėlapiais paremtų sekų metodai;
- Gylio žemėlapiais paremtų erdvėlaikinių tūrių metodai;
- Skeletu paremtų sekų metodai;
- Skeletu paremtų erdvėlaikinių tūrių metodai.

Gylio žemėlapiais paremtų algoritmų pagrindinis trūkumas – sunkumai randant reikiamas atitiktis laike. Tokie metodai bando nustatyti reikiamas žmogaus kūno savybes, iš kurių būtų galima patikimai spręsti apie atliekamą veiklą. Dažniausiai tokie algoritmai apdoroja žmogaus siluetą.

Gylio žemėlapiais paremtų erdvėlaikinių tūrių metodai bando gylio žemėlapyje rasti reikšminius trimačius taškus. Jų esmė – rasti nedidelį kiekį trimačių taškų, iš kurių būtų galima spręsti apie žmogaus veiklą. Tyrimais nustatyta, kad 90% atpažinimo informacijos neša tik 1% visų gylio žemėlapių taškų, dėl to nedidelio kiekio visiškai pakanka gana tiksliam atpažinimui ir pagreitina atpažinimo procesą. Šių metodų trūkumas – reikia, kad visi reikiami taškai būtų matomi; ne visose pozose esantį žmogų pavyksta sumodeliuoti į norimus trimačius taškus.

Skeletu paremtų sekų metodai bando rasti norimus žmogaus kūno taškus, dažniausiai sąnarius. Žmogaus kūnas reprezentuojamas sąnarių fazinės erdvės tašku, o judesys – kreive toje erdvėje. Veiksmai projektuojami kaip taškų seka ir atpažįstami patikrinant, ar jie yra ant reikiamo veiksmo kreivės. Veiksmų kreivės gali būti išskirstomos į keletą kreivių mažesnių matmenų fazinėse erdvėse modeliuojant atskirų kūno dalių judesius. Tokie metodai dažnai ribojami vieno žmogaus sekimu.

Skeletu paremtų erdvėlaikinių tūrių metodai panašūs į gylio žemėlapiais paremtus erdvėlaikinius metodus, tik apdoroja ne siluetą, o sąnarių taškų debesį.

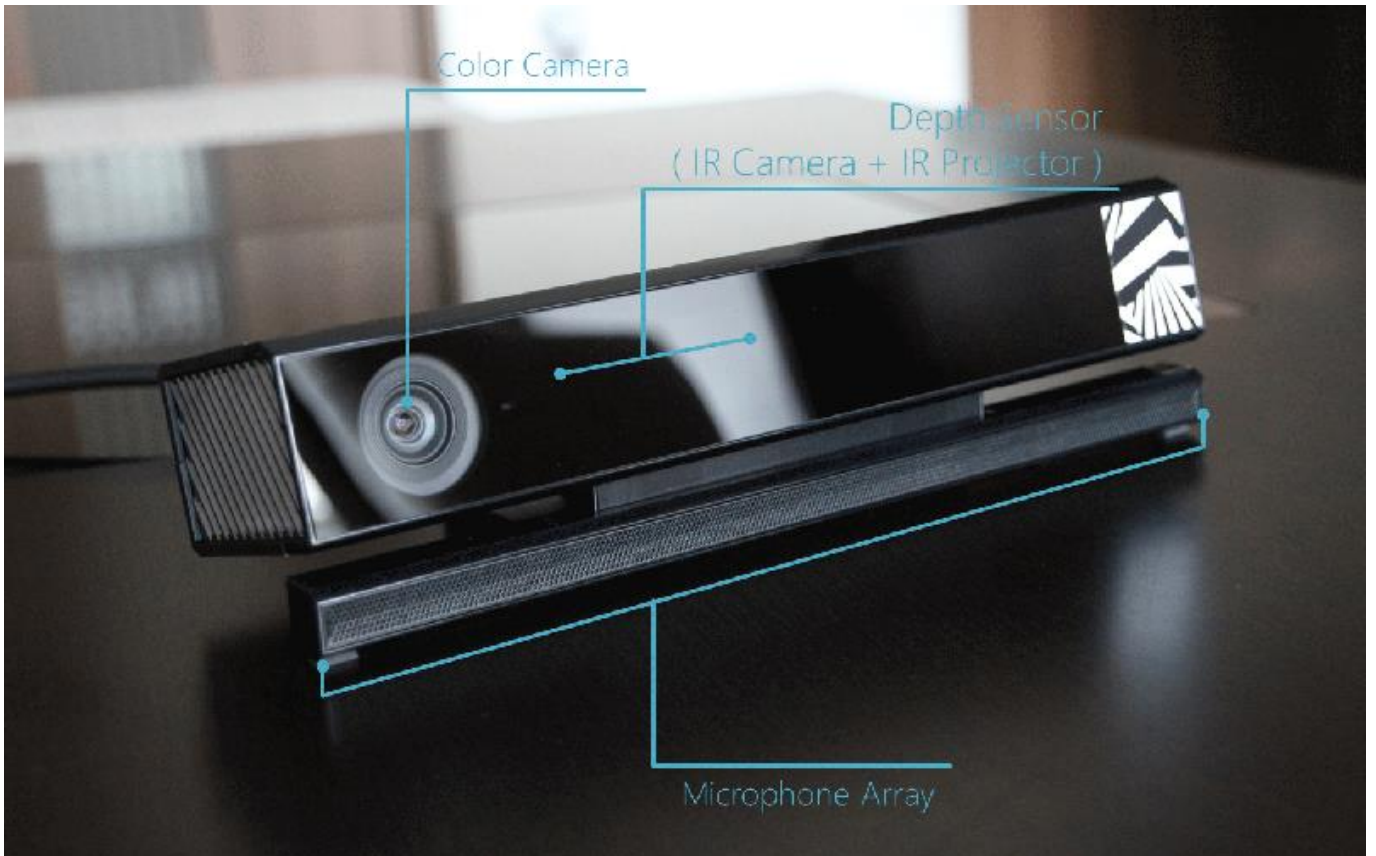
Skeletu grįstiems algoritmams reikia daugiau analizuoti gylio žemėlapi, kad galėtų išskirti žmogaus skeleto taškus, tačiau tai leidžia natūraliau, aukštesniu semantikos lygiu analizuoti jo veiklą.

2.3. Microsoft „Kinect“ trūkumai ir ribojimai

Microsoft „Kinect“ yra įrenginys, skirtas žmogaus kūno atpažinimui ir interpretavimui. „Kinect“ sudaro [5]:

- Infraraudonųjų spindulių jutiklis;
- Spalvų jutiklis;
- Infraraudonųjų spindulių gylio jutiklis;
- Mikrofonas.

Įrenginys ir jo jutikliai pavaizduoti 2.1 pav. [6]



2.1 pav. „Kinect 2“ įrenginys ir jo jutikliai

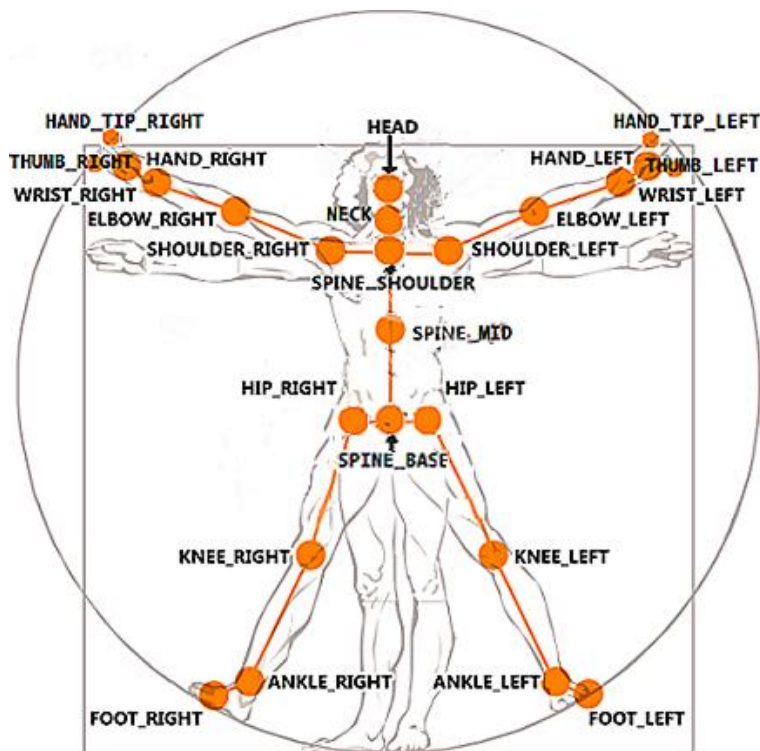
Remiantis šiais jutikliais „Kinect“ per savo programuojamą sąsają suteikia prieigą prie tokių duomenų srautų [7]:

- Garso;
- Kūno;
- Kūno indekso;
- Spalvų;
- Gylio;
- Infraraudonųjų spindulių;
- Ilgo išlaikymo infraraudonųjų spindulių.

Nagrinėjamai sričiai naudingi jau apdoroti duomenų srautai – kūno, kūno indekso ir gylio. Kūno srauto kadras atvaizduoja žmogaus skeletą 25 jo taškais [8]. Šie taškai pavaizduoti 2.2 pav. [9] Atlikus šio srauto tikslumo analizę paaiškėjo, kad „įrenginys ypač prastai atpažįsta žmogaus kūną

nestandartinėse būsenose - tupint, sėdint, gulint ir t.t.“ [10]. Analizės metu nustatyta, kad „Kinect“ daroma paklaida kai kuriose pozose siekia net 32%. Ypatingai iškraipomi trumpesni žmogaus kaulai, šiek tiek mažiau – ilgesni. Tyrimo detalės pateikiamos priede 9.5 „Nestandartinių žmogaus kūno pozicijų atpažinimo tikslumo naudojant „Kinect 2.0“ jutiklius tyrimas“.

Iš tyrimo rezultatų panašu, kad „Kinect“ skeleto atpažinimas buvo kurtas laikant, kad žmogus stovės prieš įrenginį ir skeletas visada gaunamas artimas stovimam. Dėl šios priežasties taikomajai sričiai sudėtinga taikyti jau „Kinect“ pilnai interpretuotą žmogaus skeletą. Norint tikslesnių rezultatų, reikės remtis „Kinect“ kūno indekso ir gylio srautais.

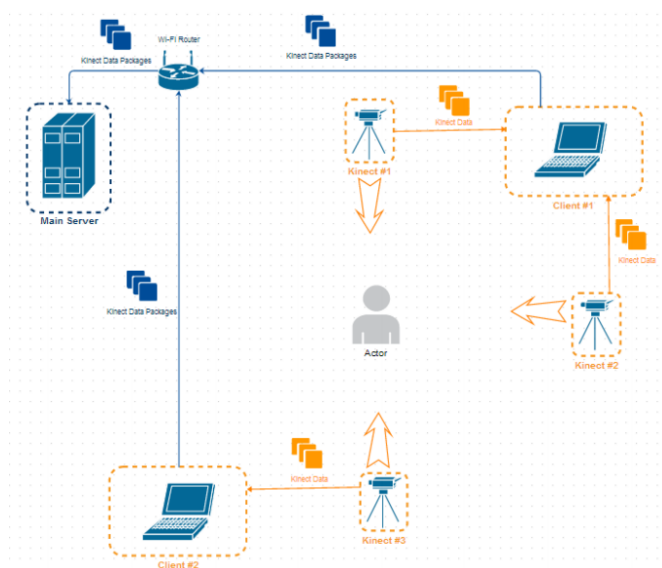


2.2. pav. "Kinect" atpažįstamo žmogaus skeleto struktūra

„Kinect“ kūno indekso ir gylio srautai yra susieti tarpusavyje – naudoja tą pačią koordinacijų sistemą [11]. Atlikus „Kinect“ gylio srauto tikslumo matavimus paaiškėjo, kad „atsitiktinė gylio matavimo paklaida didėja kvadratinio greičiu didėjant atstumui iki jutiklio ir pasiekia 4 cm ties maksimaliu 5 m atstumu“ [12]. Taigi blogiausiu atveju „Kinect“ daro maždaug 0,8% paklaidą vertindamas gylio srautą. Tai yra žymiai tiksliau, nei skeleto interpretacija, kurios net ir stovimose padėtyse paklaidos siekė iki 10%. Taigi šis srautas yra patikimesnis, nei skeleto srautas.

„Kinect“ jutiklio gylio horizontalus matymo kampas yra $84,1^\circ$, vertikalus – $53,8^\circ$ [13], maksimalus veikimo atstumas – 5 m. Tai techninės įrangos apribojimai, kurių pakeisti negalima. Nustatyta, kad geriausiai „Kinect“ veikia 1-3 m atstumu [14]. „Kinect 1“ gylio kadro raiška yra 320×240 taškų (76,8 kpix), „Kinect 2“ – 512×424 (217 kpix) [15]. „Kinect 2“ jutiklio gylio kadras yra maždaug 2,8 karto raiškesnis, nei „Kinect 1“.

„Kinect“ skeleto duomenų tikslinimui sukurta sistema iš keleto „Kinect“ įrenginių. Ji remiasi prielaida, kad „Kinect“ daro atsitiktinę paklaidą. Dėl šios priežasties keleto „Kinect“ panaudojimas leistų šią paklaidą sumažinti. Tyrimui buvo naudojama 3 „Kinect“ jutikliai. Tyrimo schema parodyta **2.3. pav.** [16]



2.3. pav. "Kinect" jutiklių išdėstymo schema atliekant judesių atpažinimą keliais "Kinect" įrenginiais

Sistemoje naudojami trys „Kinect“ sensoriai skeleto atpažinimui bei trys asmeniniai kompiuteriai sensorių prijungimui bei duomenų apdorojimui atlikti. Taip pat naudojama bevielio tinklo stotelė duomenų perdavimui tarp kompiuterių. Visa tai išdėstyta erdvioje aplinkoje, kurioje keturių metrų atstumu vienas nuo kito pastatyti trys „Kinect“ sensoriai, savo matymo kampu apimantys centre esantį žmogų.

Kiekvieno jutiklio gautos žmogaus sąnarių koordinatės siunčiamos tokioje koordinacių sistemoje, kurios centras yra pats jutiklis. Norint sujungti žmogaus sąnarių pozicijas, gaunamas iš kelių „Kinect“ sensorių, reikia gautus duomenis suvesti į bendrą koordinacių sistemą, o po to jiems pritaikyti algoritmą, kuris apibendrintų duomenis bendroje koordinacių sistemoje. Tikslumo gerinimui naudojamas sąnarių koordinacių skaičiavimo algoritmas.

Naudojamas tiesinėmis transformacijomis pagrįstas metodas. Algoritmo pradiniai duomenys - žmogaus sąnario koordinacių srautai, kuriuos siunčia „Kinect“ įrenginiai. Algoritmo grąžinamas rezultatas - atitinkamo sąnario koordinatės, apskaičiuotos įvertinus kiekvieno „Kinect“ įrenginio siunčiamą informaciją. Šio algoritmo tyrimo metu gautos paklaidos siekia 1,5%. Tai žymiai tiksliau, nei be jo gauti 10%. Toks tikslumas leidžia taikyti algoritmą praktikoje. „Kinect“ sensoriai yra pakankamai tikslūs, todėl gautų koordinacių buferizavimas nėra efektyvus. Be to, jiems sudaryti reikalingas laikas, dėl to sumažėja apdorojamų kadrų skaičius per laiko vienetą. Tikslumą būtų galima padidinti įvertinant „Kinect“ sensoriaus posvirį aukštyrį ar žemyn, tačiau šis tikslinimas išeina už šio

tyrimo ribų ir jį galima taikyti lygiai taip, kaip ir vienam skeletui. Detalus tyrimo ir algoritmo aprašymas pateikiamas priede 9.3 „Žmogaus sąnarių padėties nustatymo naudojant kelis „Kinect“ įrenginius algoritmo sudarymas ir tyrimas“.

2.4. Egzistuojantys sprendimai

Rinkoje egzistuoja keletas alternatyvių žaidimų kuriamam jogos žaidimui:

- *Your Shape Fitness Evolved* – žaidimas, siūlantis įvairias treniruotes: šokinėjimą per virvutę, boksavimąsi, jogą ir kt. Įtraukia veiklas, kurias reikia atlikti stovimoje padėtyje žiūrint į ekraną;
- *Dance Central* – šokių žaidimas, reikalaujantis teisingai atlikti šokio judesius. Remiasi rankų ir kojų judesiais, bet ne kūno pasisukimais – vartotojas visada stovi veidu į ekraną. Kitokių pratimų neapima;
- *Nike+ Kinect Training* – treniruočių žaidimas, kurio metu virtualus treneris praveda norimo tipo fizinius pratimus – norintiems atsikratyti riebalų, įgyti raumenų masės ar jėgos. Treniruotės apima sudėtingas žmogaus pozicijas – pritūpimai, atsispaudimai, gulimos, tupimos pozicijos. Pratimų jogai neturi.

Lentelėje **Lentelė 2.1** pateikiamas šių žaidimų palyginimas. Pagrindiniai kriterijai, kuriais remiamasi – jogos pratimai, virtualaus trenerio turėjimas, integracija su papildomais įrenginiais, judesių sekimas. Tai yra pagrindiniai kuriamo produkto bruožai.

Lentelė 2.1. Žaidimų, skirtų „Kinect“, palyginimas

Kriterijai	<i>Your Shape Fitness Evolved</i>	<i>Dance Central</i>	<i>Nike+ Kinect Training</i>
Turi jogos pratimus	Taip	Ne	Ne
Turi virtualų trenerį	Taip	Ne	Taip
Palaiko ne tik stovimus pratimus	Ne	Ne	Taip
Integruotas su keletu įrenginių	Ne	Ne	Ne
Seka ne tik pozicijas, bet ir judesius	Ne	Taip	Neįmanoma nustatyti

Kuriamam programiniam komponentui analogų rasti nepavyko. Panašus komponentas turėtų būti naudojamas žaidime „Dance Central“, tačiau apie tai jokios informacijos nėra. Viena iš komponento dalių bus judesių įrašymo įrankis. Žemiau pateikiamas kitų judesių įrašymo įrankių palyginimas.

- *Kinect Skeleton Recorder* – atviro kodo projektas, veikiantis „Windows“ aplinkoje ir galintis įrašyti „Kinect“ skeleto duomenis į failą. Tinkamas judesių įrašymui, tačiau gali įrašyti tik sąnarių pozicijų srautą, kitų srautų – ne. Nepalaiko „Kinect 2“ įrenginio;
- *Kinect Stream Saver* – atviro kodo projektas, veikiantis „Windows“ aplinkoje. Gali įrašyti „Kinect“ skeleto duomenis. Kiekvienam skeleto kadrui sukuria po failą, todėl ilgam judesiui įrašyti reikės labai daug failų. Nepalaiko „Kinect 2“ įrenginio;
- *Kinect Studio 2* – oficialus „Microsoft“ įrankis darbui su „Kinect“ įrenginiu. Gali įrašyti, atkurti visą „Kinect“ duomenų srautą, išsaugoti jį į failą, transliuoti įrašytus duomenis kaip originalų „Kinect“ duomenų srautą. Orientuotas į testavimą. Neturi programuojamos sąsajos (API).

Struktūrizuotas palyginimas pateikiamas **Lentelė 2.2** lentelėje.

Lentelė 2.2. Įrankių, skirtų „Kinect“ duomenų įrašymui, palyginimas

Kriterijai	<i>Kinect Skeleton Recorder</i>	<i>Kinect Stream Saver</i>	<i>Kinect Studio 2</i>
Gali įrašyti judesius	Taip	Taip	Taip
Gali atkurti judesius grafinėje sąsajoje	Taip	Ne	Taip
Palaikomas formatas	XML	Specialus tekstinis formatas	Uždaras XED formatas
Palaiko „Kinect 2“	Ne	Ne	Taip
Mokamas	Atviro kodo	Nemokamas, tik nekomerciniam naudojimui	Nemokamas
Gali palyginti pozicijas	Ne	Ne	Ne

Nė vienas iš nagrinėtų žaidimų nepalaiko visų norimų funkcijų, todėl yra poreikis kurti naują, joga skirtą žaidimą, kuris palaikytų ne tik stovimus jegos pratimus, bet ir turėtų virtualų trenerį, būtų integruotas su keletu įrenginių („Kinect“ ir „Muse“) bei sektų judesius, ne tik pavienes pozicijas. Nė vienas iš „Kinect“ skirtų įrašymo įrankių taip pat nepalaiko jiems keliamų reikalavimų, todėl reikia kurti naują įrankį, kuris galėtų tiek įrašyti, tiek atkurti judesių įrašus, palaikytų „Kinect 2“ ir būtų integruojamas kuriamame jegos žaidime.

2.5. Judesių ir pozicijų palyginimo algoritmų analizė

Jogos pozicijų atpažinimo algoritmuose svarbu jų veikimo greitis. Dažniausiai šis greitis priklauso nuo to, kokio lygio duomenimis operuoja algoritmas. Aparatūrinė įranga gali pateikti gylio žemėlapi. Idealu, jei algoritmas gali patikimai lyginti pozicijas vien tik iš jo. Daugeliui algoritmų reikia jau apdorotų duomenų. Kuo aukštesnės semantikos duomenys reikalingi, tuo ilgiau užtruks jų analizė. Kitas svarbus aspektas – algoritmo gebėjimas įvertinti, kad figūros viena kitos atžvilgiu yra pastumtos ir nevienodo dydžio. Taip pat algoritmas turi gebėti palyginti dvi judesių sekas ar bent jau turi būti galima jį modifikuoti taip, kad toks palyginimas būtų įmanomas. Žaidime bus reikalinga indikuoti skirtumus tarp figūrų, todėl naudinga, jei algoritmas veikdamas gali ne tik nustatyti kiekybinį įvertį, koks

Apibendrinus galime išskirti tokius pagrindinius algoritmų vertinimo kriterijus:

- Operando lygmuo, kurį nagrinėja algoritmas;
- Ar algoritmas įvertina tai, kad abejuose vaizduose pozicijos gali būti skirtingai išdėstytos;
- Ar reikia papildomai apdoroti gylio buferį;
- Ar pritaikomas sekų palyginimui;
- Ar analizės metu galima nustatyti neatitikimus tarp figūrų, ar tai reikia daryti jau atskiru žingsniu;

Keliama užduočiai galima naudoti du skirtingus modelius:

- Palyginti dvi figūras, jas laikant uždaromis kreivėmis;
- Palyginti dvi figūras, jas laikant taškų matricomis;

„Kinect“ programiniai įrankiai remiasi taškų matricų modeliu [17], todėl uždarų kreivių modeliams reiktų papildomo apdorojimo. Figūrų palyginimui taikomi algoritmai aprašyti šiame skyriuje.

2.5.1. Hausdorfo atstumas

Hausdorfo atstumas yra skirtumo tarp dviejų taškų aibių skaitinis įvertis. Tarkime, kad turime dvi baigtines taškų sekas $A = \{a_1, \dots, a_p\}$ ir $B = \{b_1, \dots, b_q\}$. Tuomet Hausdorfo atstumas šioms aibėms apibrėžiamas taip:

$$H(A, B) = \max(h(A, B), h(B, A)) \quad (9);$$

čia

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad (10).$$

Funkcija $h(A, B)$ yra vadinama kryptiniu Hausdorfo atstumu tarp aibių A ir B . Ji nurodo tašką $a \in A$, kuris yra tolimiausias bet kuriam taškui $b \in B$ ir nurodo atstumą tarp taško a ir jam artimiausio kaimyno iš aibės B . Funkcija $h(A, B)$ įvertina kiekvieną aibės A tašką priklausomai nuo jam artimiausio taško aibėje B ir tada naudoja didžiausią įvertį turintį tašką kaip atstumą (labiausiai nuklydusį aibės A tašką).

Hausdorfo atstumas yra didesnioji reikšmė iš $h(A, B)$ ir $h(B, A)$. Taigi jis parodo neatitikimo laipsnį tarp dviejų aibių, įvertinant mažiausią atstumą nuo taško iš aibės A , kuris yra toliausiai nuo bet kurio taško iš aibės B ir atvirkščiai. Visi aibės A taškai yra arčiau nors vieno aibės B taško nei Hausdorfo atstumas. Priešingai, nei kai kurie kiti palyginimo algoritmai, Hausdorfo atstumas nieško specialių taškų porų, o operuoja visais galimais taškais.

Hausdorfo atstumas gali būti apskaičiuotas per laiką $O(pq)$, bet gali būti optimizuotas iki laiko $O((p + q) \log(p + q))$, kur p ir q yra nagrinėjamų aibių dydžiai [18].

Hausdorfo atstumo pagrindinis trūkumas – jis matuoja skirtumus tarp dviejų vaizdų fiksuotoje pozicijoje. Jei vaizdai yra vienodi, bet pastumti vienas kito atžvilgiu, Hausdorfo atstumas bus nenulinis. D. Huttenlocheris, G. Klandermanas ir W. Rucklidge pasiūlė sprendimą šiai problemai spręsti [19]. Jie nagrinėja visas galimas vaizdų tarpusavio padėtis.

Kiekvienai galimai grupei G apibrėžiame minimalų Hausdorfo atstumą

$$M_G(A, B) = \min_{g_1, g_2 \in G} H(g_1 A, g_2 B) \quad (11).$$

Jei grupė G yra tokia, kad kiekvienam $g \in G$ ir kiekvienam taškui x_1, x_2 , $\|gx_1 - gx_2\| = \|x_1 - x_2\|$, tada reikia transformuoti tik vieną iš aibių. Tuomet minimalus Hausdorfo atstumas

$$M_G(A, B) = \min_{g \in G} H(A, gB) \quad (12).$$

Jei fiksuojame aibę A ir darome postūmį tik aibei, tada minimalų Hausdorfo atstumą galima užrašyti

$$M_T(A, B) = \min_t H(A, B \oplus t) \quad (13).$$

Iškyla optimalios funkcijos M_T radimo problema. Nors Hausdorfo atstumą galima suskaičiuoti gana greitai, reikia minimizuoti Hausdorfo atstumo apskaičiavimų kiekį pastumtiems vaizdams. Huttenlocheris, Klandermanas ir Rucklidge sprendžia šį uždavinį [19].

Sujungę (1) ir (2) formules galime Hausdorfo atstumą apibrėžti taip:

$$H(A, B) = \max(h(A, B), h(B, A)) = \max\left(\max_{a \in A} \min_{b \in B} \|a - b\|, \max_{b \in B} \min_{a \in A} \|b - a\|\right) \quad (14).$$

Apibrėžiame funkcijas $d(x) = \min_{b \in B} \|x - b\|$ ir $d'(x) = \min_{a \in A} \|x - a\|$, Hausdorfo atstumą galima performuluoti į:

$$H(A, B) = \max\left(\max_{a \in A} d(a), \max_{b \in B} d'(b)\right) \quad (15).$$

Tuomet minimalus Hausdorfo atstumas tarp dviejų aibių lygus

$$\begin{aligned} H(A, B \oplus t) &= \max\left(\max_{a \in A} \min_{b \in B} \|a - (b + t)\|, \max_{b \in B} \min_{a \in A} \|(a - b + t)\|\right) \\ &= \max\left(\max_{a \in A} \min_{b \in B} \|(a - t) - b\|, \max_{b \in B} \min_{a \in A} \|a - (b + t)\|\right) \\ &= \max\left(\max_{a \in A} d(a - t), \max_{b \in B} d'(b + t)\right) \end{aligned}$$

Autoriai siūlo nagrinėti tik tokias transformacijas t , kurios sutapdina abiejų nagrinėjamų aibių bent vieną tašką. Taip pat siūloma taikyti tokias optimizacijas:

- Užklojimo optimizacija. Nagrinėjamos tik tos transformacijos, kurias atlikus abi nagrinėjamos figūros persikloja bent vienoje vietoje, kitos atmetamos.
- Išankstinis tikrinimo nutraukimas. Jei figūrų palyginimo tikslas yra nustatyti faktą, ar figūrų skirtumas neviršija numatytos leidžiamos ribos, galima neskaičiuoti viso minimalaus Hausdorfo atstumo, jei nors vienam taškui jis viršija numatytą ribą.
- Tikrinimo praleidimas. Jei kuriai nors transformacijai atstumas didesnis, nei numatyta riba dydžiu Δx , tai galime praleisti visus taškus spinduliu Δx aplink tą tašką, nes Hausdorfo atstumas nebus mažesnis, nei numatyta riba ir juose.

Teigiama, jog 38x60 taškų dydžio objekto paieška 360x240 taškų paveiksle truko 1,4 sek.

2.5.2. Kendalio figūrų teorija ir apibendrinta Prokrusto analizė

Nagrinėjama figūrų palyginimo strategija remiasi Kendalio figūros apibrėžimu: „figūra yra visa geometrinė informacija, kuri išlieka pašalinus padėties, dydžio ir posūkio įtaką iš objekto“ [20]. Kendalio analizei naudojamas skaitmenizuotas siluetas – dvimatė matrica, reprezentuojanti figūros taškus, toliau vadinamus siluetu. Siluetas pertvarkomas į standartinį pavidalą – pašalinamos visos silueto savybės, kurias reikia atmesti pagal Kendalio apibrėžimą. Tai atliekama sucentruojant ir normalizuojant figūrą:

$$Z_C = \frac{CX}{\|CX\|}, \text{ kur } C = I_k - \frac{1}{k} \mathbf{1}_k \mathbf{1}_k^t \quad (16);$$

čia X – k matmenų kompleksinis vektorius, kurį sudaro visų nagrinėjamų taškų koordinatės, I_k – $k \times k$ dydžio identifikacinė matrica, $\mathbf{1}_k$ – k matmenų vienetinis vektorius.

Sakykime, kad turime du kompleksinius vektorius X ir Y , apibūdinančius normalizuotas figūras α ir β . Prokrusto atstumą apibrėšime kaip Euklidinį atstumą pilno Prokrusto atitiktens tarp α ir β [21]. Pilnas Prokrusto atitiktumo parenkamas taip, kad būtų minimizuota

$$d(Y, X) = \|\beta - \alpha s e^{j\theta} - (a + jb)1_k\| \quad (17);$$

čia s – dydžio pokytis, θ – posūkis ir $(a + jb)$ – postūmis. Taigi pilnas Prokrusto atstumas lygus:

$$d_F(Y, X) = \inf_{s, \theta, a, b} d(Y, X) \quad (18).$$

Norint surasti optimalius parametrus postūmiui, posūkiui ir dydžio pakeitimui atliekama apibendrinta Prokrusto analizė [22]. Ji atliekama trimis etapais:

- Postūmis;
- Izomorfinis mastelio pakeitimas;
- Posūkis;

Postūmio žingsnis perkelia nagrinėjamas figūras į bendrą centrą. Juo galima rinktis koordinatinių pradžių tašką, bet nebūtinai. Transformacija atliekama taip:

$$X_C = X - \frac{1}{k} \begin{pmatrix} \sum_{i=1}^k x_{i1} & \cdots & \sum_{i=1}^k x_{im} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^k x_{i1} & \cdots & \sum_{i=1}^k x_{im} \end{pmatrix} \quad (19);$$

čia X – pasirinktų taškų koordinatės m matmenyse, X_C – naujo centro koordinatės. Centroidas apskaičiuojamas iš matricos X stulpelių sumų, padalintų iš taškų kiekio (eilučių skaičiaus). Kai centroidas apskaičiuotas, jį atėmus iš kiekvieno matricos elemento gausime sucentruotą figūrą.

Izomorfinis mastelio pakeitimas transformuoja figūrą į didesnę ar mažesnę išlaikydamas figūros proporcijų santykį. Normalizavimas yra vienas iš izomorfinio mastelio pakeitimo variantų ir naudojamas suvienodinti figūrų dydžiui.

$$X_n = X \left(\frac{1}{\|X\|} \right) \quad (20);$$

čia: X – sucentruotos koordinatės, X_n – normalizuotos ir sucentruotos koordinatės.

Kai matricos yra sulygiuotos ir suvienodinto dydžio, reikia atlikti posūkio žingsnį. Posūkio metu visos figūros sutapdinamos su figūra, su kuria lyginame kitas. Posūkiui naudojama Euklido-Frobenijaus norma, kuri yra visų matricos elementų kvadratų suma. Tikslas yra minimizuoti skirtumą tarp pagrindinės figūros ir pasuktos figūros. Norint optimaliai pasukti figūrą, reikia jos matricą dauginti iš matricos Q , kuri lygi

$$Q = VU' \quad (21);$$

čia U ir V - matricos, gautos atlikus singuliarinę dekompoziciją nagrinėjamos ir pagrindinės matricų sandaugai.

Pagrindinis šio metodo apribojimas – taškų kiekis visose figūrose turi būti lygus.

2.5.3. Figūrų sekų palyginimas

A. Veeraraghvanas, A. K. Roy-Chowdhury ir R. Chellapa apibendrina figūrų palyginimo metodus [21]. Metodai skirstomi į parametrinius ir nparametrinius.

Nagrinėjamas nparametrinis metodas – dinaminė laike iškreipianti funkcija. Šio metodo tikslas – rasti skirtumą tarp dviejų figūrų sekų. Šis metodas turi tokius apribojimus:

- Abiejų sekų pirmas ir paskutinis taškai yra fiksuoti.
- Iškreipimo funkcija turi būti monotoniškai didėjanti, t. y., abiejose sekose įvykiai vyksta vienoda seka.
- Iškreipianti funkcija turi būti netrūki.

Šio metodo principas – nagrinėjamos figūros yra suvienodinamos, kaip Prokrusto algoritme, bei patalpinamos ant sferinio paviršiaus. Jei $A(t)$ ir $B(t)$ yra figūrų sekos, tai atstumas tarp jų apibrėžiamas

$$D(A(t), B(t)) = DTW(A(t), B(t)) + DTW(B(t), A(t)) \quad (22);$$

čia $DTW(A(t), B(t)) = \frac{1}{T \sum_{t=1}^T d(A(f(t)), B(g(t)))}$, f ir g – optimalios iškreipiančios funkcijos. Jos parenkamos taip, kad Prokrusto atstumas tarp A ir B figūrų būtų minimalus.

Kitas nagrinėjamas metodas – parametrinis ARMA metodas (angl. *AutoRegressive and Moving Average*). Jis remiasi figūrų transformacijų laike analize. Figūros apdorojamos taip pat, kaip prieš taikant dinaminę iškreipiančią funkciją, bet po to dar apskaičiuojamos tangentinės projekcijos figūroms $\alpha(t)$. Tuomet ARMA modelis apibrėžiamas

$$\alpha(t) = Cx(t) + w(t); w(t) \sim N(0, R) \quad (23);$$

$$x(t+1) = Ax(t) + v(t); v(t) \sim N(0, Q) \quad (24).$$

Šie algoritmai veikia naudodami ne visus, o tik pasirinktus žmogaus taškus, todėl jiems realizuoti dar reikia vykdyti žmogaus raktinių taškų radimą.

2.5.4. Erdvėlaikinių šablonų lyginimo algoritmas ir nuožulnos atstumas

Nuožulnos atstumu paremtas algoritmas remiasi žmogaus silueto kontūro palyginimu. Nuožulnos atstumas apibrėžiamas

$$d_{chamfer}(S, C) = \frac{1}{n} \sum_{s_1 \in S} \min_{c_j \in C} \|s_1 - c_j\| \quad (25);$$

čia S yra siluetas, sudarytas iš n taškų, C – kontūro taškų aibė [23]. M. Dimitrijevičius, V. Lepetitas ir P. Lua pasiūlė šio algoritmo optimizacijas, padedančias išvengti neteisingų palyginimų. Įvedama baudos funkcija silueto kraštų orientacijai

$$p(s_i, c_j) = K[\tan(\alpha_{s_i} - \beta_{c_j})]^2 \quad (26);$$

čia α_{s_i} ir β_{c_j} yra kraštinių orientacijos atitinkamai silueto taške ir kontūro taške, K – svoris, apibūdinantis baudos funkcijos posvyrį (krypties koeficientas). Algoritme naudojama atstumo transformacija. Ji modifikuojama taip, kad kiekvienas taškas atstumo transformacijos atvaizdyje turėtų informaciją apie artimiausio krašto orientaciją. Autoriai naudoja koeficiento K reikšmę, lygią 20. Jos pakanka pašalinti tų taškų įtaką, kurių kraštinės orientacija viršija 30° . Neaptiktų kraštinių įtakos sumažinimui naudojamas Tuki patikimas vertinimas [24]. Tuomet nuožulnos atstumas tampa

$$d_{chamfer}(S, C) = \frac{1}{n} \sum_{s_i \in S} \rho\left(\frac{1}{k} \|s_i - c(s_i)\| + p(s_i, c(s_i))\right) \quad (27).$$

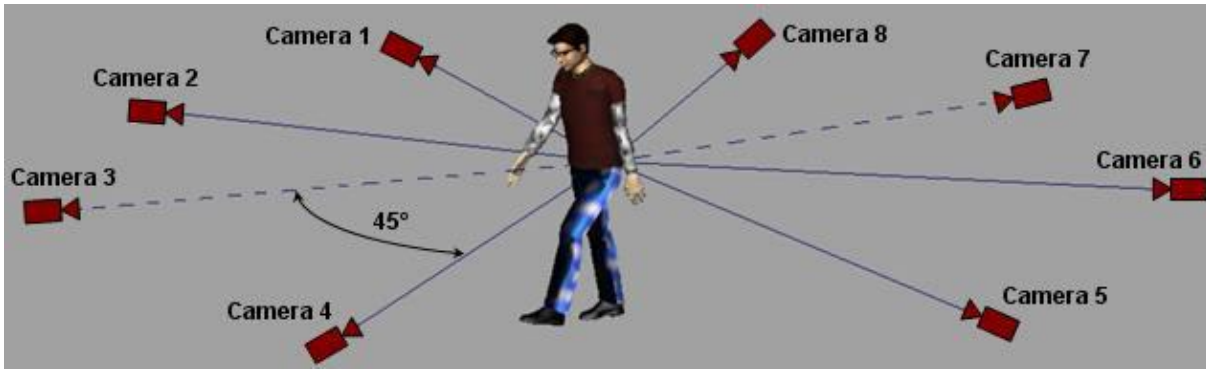
Šis metodas leidžia palyginti dvi statines pozas. Jis išplečiamas dinaminėms pozoms. Sakykime, kad turime žmogaus siluetus $I_1, I_2, \dots, I_t, \dots, I_{tmax}$, kur t – diskretus laikas, $tmax$ – laikas, kuriuo buvo užfiksuotas paskutinis kadras. Analogiškai kiekvienas šablono kadras sudarytas iš siluetai sekos $T = \{T_1, T_2, \dots, T_i, \dots, T_{N_S}\}$. Tegul $I_t^{(x,y)}$ – stačiakampė kadro I_t iškarpa, kurios kairysis viršutinis taškas yra (x, y) , o matmenys vienodi visiems kadrui. Tuomet atstumas D tarp siluetai sekos poaibio $I_{t+1}, \dots, I_{t+N_S}$ ir šablono T bus lygus

$$D(T, x, y, I_{t+1}, \dots, I_{t+N_S}) = \frac{1}{N_S} \sum_{i=1}^{N_S} d_i(x, y, T) \quad (28);$$

$$d_i(x, y, T) = \min_{x', y' \in Area(x, y)} d_{chamfer}(T_i, I_{t+1}^{(x', y')}) \quad (29).$$

Toks atstumo skaičiavimas leidžia nedideles variacijas tarp skirtingų siluetai kadrų.

Šis algoritmas leidžia lyginti sekas nepriklausomai nuo kameros padėties. Algoritmas taip pat leidžia išgauti trimatį objekto vaizdą, jei siluetai buvo sudaryti naudojant keletą kamerų. Pavyzdinis šių kamerų išdėstymas parodytas **2.4** pav.



2.4 pav. Pavyzdinis kamerų išdėstymas filmuojant pradinę siluetą erdvėlaikiniam figūrų palyginimui

2.5.5. Algoritmų apibendrinimas

Algoritmų taikomumui nagrinėjamose srityje svarbūs šie kriterijai:

- Operando lygmuo, kurį nagrinėja algoritmas
- Ar algoritmas įvertina tai, kad abejuose vaizduose pozicijos gali būti skirtingai išdėstytos
- Ar reikia papildomai apdoroti gylio buferį
- Ar pritaikomas sekų palyginimui
- Ar analizės metu galima nustatyti neatitikimus tarp figūrų, ar tai reikia daryti jau atskiru žingsniu

Hausdorfo algoritmas dirba iškart su dvejetainėmis matricomis, kurias labai nesunku gauti iš gylio nuskaitymo įrenginių. Tai leidžia sutaupyti laiko, nes nereikia papildomos analizės, kad būtų galima palyginti figūras. Hausdorfo atstumui rasti reikia apskaičiuoti atstumą visiems taškams, todėl iš karto galima gauti atstumų žemėlapi, kuris indikuotų neatitikimus tarp figūrų. Hausdorfo algoritmas gali veikti greitai, bet ilgai užtrunka duomenų analizė, kurios metu nustatoma, koks turi būti postūmis, posūkis ir mastelis. Hausdorfo algoritmas sekų palyginimo nenumato, bet jį galima emuliuoti iš individualių palyginimų.

Prokrusto analizei reikalinga esminių taškų aibė, kuriuos algoritmas turi galėti aptikti nagrinėjamose figūrose. Jų radimui reikalingas papildomas laikas. Sekų palyginimo pritaikymas analogiškas Hausdorfo algoritmui. Kadangi operuojama tik matricomis, atstumų žemėlapi teks skaičiuoti atskirai. Kita vertus, šis algoritmas puikiai vertina skirtingas orientacijas tarp figūrų.

Dinaminė laike iškreipianti funkcija naudoja tą patį mechanizmą pradinėms figūrų transformacijoms, kaip ir Prokrusto analizė, dėl to ši dalis yra stiprioji šio algoritmo pusė. Jis specialiai buvo kurtas judesių sekų palyginimui. Kita vertus, jis operuoja skeletu, kurio taškų radimas yra komplikovota užduotis. Taip pat algoritmas nepritaikytas skaičiuoti atstumų žemėlapiui.

Nuožulnos atstumas yra skirtas sekų palyginimui, jo metu konstruojami atstumų žemėlapiai, kurie duoda informaciją apie neatitikimus tarp figūrų. Šis algoritmas operuoja žmogaus silueto kontūru, kuriam reikia papildomo apdorojimo, bet mažiau, nei Prokrusto ar dinaminės laike iškreipiančios funkcijos atveju.

Apibendrintas algoritmų palyginimas pateiktas **Lentelė 2.3** lentelėje.

Lentelė 2.3. Nagrinėjamų algoritmų apibendrinimas.

Algoritmas	Hausdorfo	Prokrusto	Dinaminė iškreipianti funkcija	laike	Nuožulnos atstumas
Operandas	siluetas	esminiai taškai	Skeletas		silueto kontūras
Įvertina nevienodą pozicijų orientaciją	Taip, bet lėtas	Taip	Taip		Iš dalies
Reikia papildomai apdoroti gylio duomenis	Ne	Iš dalies	Taip		Iš dalies
Pritaikomas sekų palyginimui	Iš dalies	Iš dalies	Taip		Taip
Analizės metu galima nustatyti neatitikimus	Taip	Ne	Ne		Taip

Iš jos matome, kad Hausdorfo ir nuožulnos atstumai neblogai tinka taikomajai sričiai. Hausdorfo algoritmo atveju reiktų modifikacijų, kaip efektyviai transformuoti figūras ir pritaikyti jį sekų analizei. Nuožulnos algoritmui reikia optimizuoti gylio duomenų analizę.

2.6. Esami Hausdorfo atstumo skaičiavimo metodai

2.6.1. Yaniro Taflevo metodas

Yaniras Taflevas siūlo atviro kodo sprendimą Hausdorfo atstumų matricos skaičiavimui [25]. Šis algoritmas paties Hausdorfo atstumo neskaičiuoja, tačiau sudaro Hausdorfo atstumų matricas. Hausdorfo atstumų matrica yra matrica, kurios kiekvienas elementas parodo atstumą iki artimiausio aibės B taško kiekvienam aibės A taškui. Apskaičiavus šias matricas abiejų aibių atžvilgiu, Hausdorfo atstumas bus lygus abiejų matricų maksimumui.

Hausdorfo atstumų matrica skaičiuojama sudarant pradinę matricą, kurios visiems taškams, esantiems aibėje B , suteikiamos reikšmės 0, nesantiems - ∞ . Tuomet nuo kiekvieno aibės B taško didėjančios kraštinės dydžio kvadratais keliaujama per matricą ir surašomos reikšmės, lygios to

kvadrato kraštinės dydžiui, jei tame taške dar nėra mažesnės reikšmės. Dėl tokios matricos sudarymo specifikos gaunami atstumai pagal Čebyševio atstumo metriką. Realizacija būtų tinkama naudoti kuriamame projekte, tačiau veikia per lėtai.

Pateikiama algoritmo realizacija C# kalba. Ši realizacija naudojama C# karkase „Shape Matching Framework“.

2.6.2. „Elastic Search“ metodas

„Elastic Search“ klientas, sukurtas „Vivid Solutions“, pateikiamas kaip atvirojo kodo Java kalba sukurtas projektas [26]. Jame naudojama diskretizuoto Hausdorfo algoritmo realizacija. Algoritmas Hausdorfo atstumą skaičiuoja apytiksliai. Algoritmas nepateikiamas. Realizacija naudojama „Elastic Search“ paieškos variklio geometrijų palyginimui. Algoritmas skaičiuoja apytikslę Hausdorfo metrikos reikšmę, kuri remiasi įvesties diskretizacija. Algoritmas skaičiuoja Hausdorfo atstumą, apribotą iki diskrečių taškų [27]. Taškai gali būti tiek viršūnės, tiek sutankinti atkarpų segmentai. Algoritmas taip pat randa du taškus, tarp kurių yra gaunamas Hausdorfo atstumas. Aproximacija gali būti kiek norima tiksli sutankinant įvestis. Sutankinimo faktoriui artėjant į 0, skaičiuojama metrika artėja į Hausdorfo atstumą:

$$\lim_{D \rightarrow 0} DHD(A, B, D) = HD(A, B) \quad (30),$$

čia A ir B – analizuojamos geometrijos, HD – Hausdorfo atstumas, DHD – diskretizuotas Hausdorfo atstumas, D – sutankinimo koeficientas.

Taigi skaičiuojama metrika yra pritaikyta atstumams tarp linijų aibių skaičiuoti, todėl nėra lengvai adaptuojama panaudojimui su „Kinect“, kur duomenys yra dvimatė taškų matrica.

2.6.3. Princetono universiteto metodas

Princetono universiteto sukurtoje sprendimo bibliotekoje „RgResolve“ naudojama Hausdorfo algoritmo Java realizacija [28]. Algoritmas remiasi prielaida, kad taškai, kurie bus nagrinėjami, surikiuoti pagal laikrodžio rodyklę ir algoritmas analizuoja juos būtent tokia tvarka. Skirtingai, nei Yaniro Taflevo realizacijoje, duomenys analizuojami ne dvimatės matricos, bet taškų sąrašo pavidalu. Atstumas matuojamas pagal Euklido normos metriką. Biblioteka skaičiuoja Hausdorfo atstumą nuo taško iki daugiakampio, todėl sprendimą pritaikyti darbui su „Kinect“ yra sudėtinga.

2.7. Analizės išvados

1. Identifikuoti svarbiausi aspektai, kuriais remiasi jogos sportas ir jam taikomi žmogaus padėties palyginimo algoritmai, nustatyti tokių algoritmų vertinimo kriterijai.
2. Apžvelgta esama algoritmų klasifikacija, jų taikymo ypatumai ir veikimo principai. Nustatyta, kad esami rinkoje panašūs sprendimai neįgyvendina viso funkcionalumo, kurį norima pasiekti kuriamame produkte, jiems trūksta gero ne stovimų pozų palaikymo.

3. Identifikuoti esami techniniai ribojimai žmogaus pozicijos nustatymui, išryškintos „Kinect“ įrenginio galimybės ir trūkumai. Nustatyta, kad „Kinect“ skeleto atpažinimas atliekamas su nemaža paklaida (~10%) ir netinkamai veikia, kai žmogus yra ne stovimoje padėtyje. Žmogaus silueto išskyrimas iš gylio buferio veikia žymiai tiksliau (~1% paklaida).
4. Apibūdinti keturi žmogaus judesių sekimui skirti algoritmai, išaiškinti jų veikimo principai. Nustatyta, kad jogos srityje geriausiai pritaikomi algoritmai yra Hausdorfo ir nuožulnos, bet nė vienas algoritmas nepatenkina visų keliamų reikalavimų.
5. Nustatyta, kad nėra greitų ir „Kinect“ pritaiktų Hausdorfo atstumo skaičiavimo metodų.

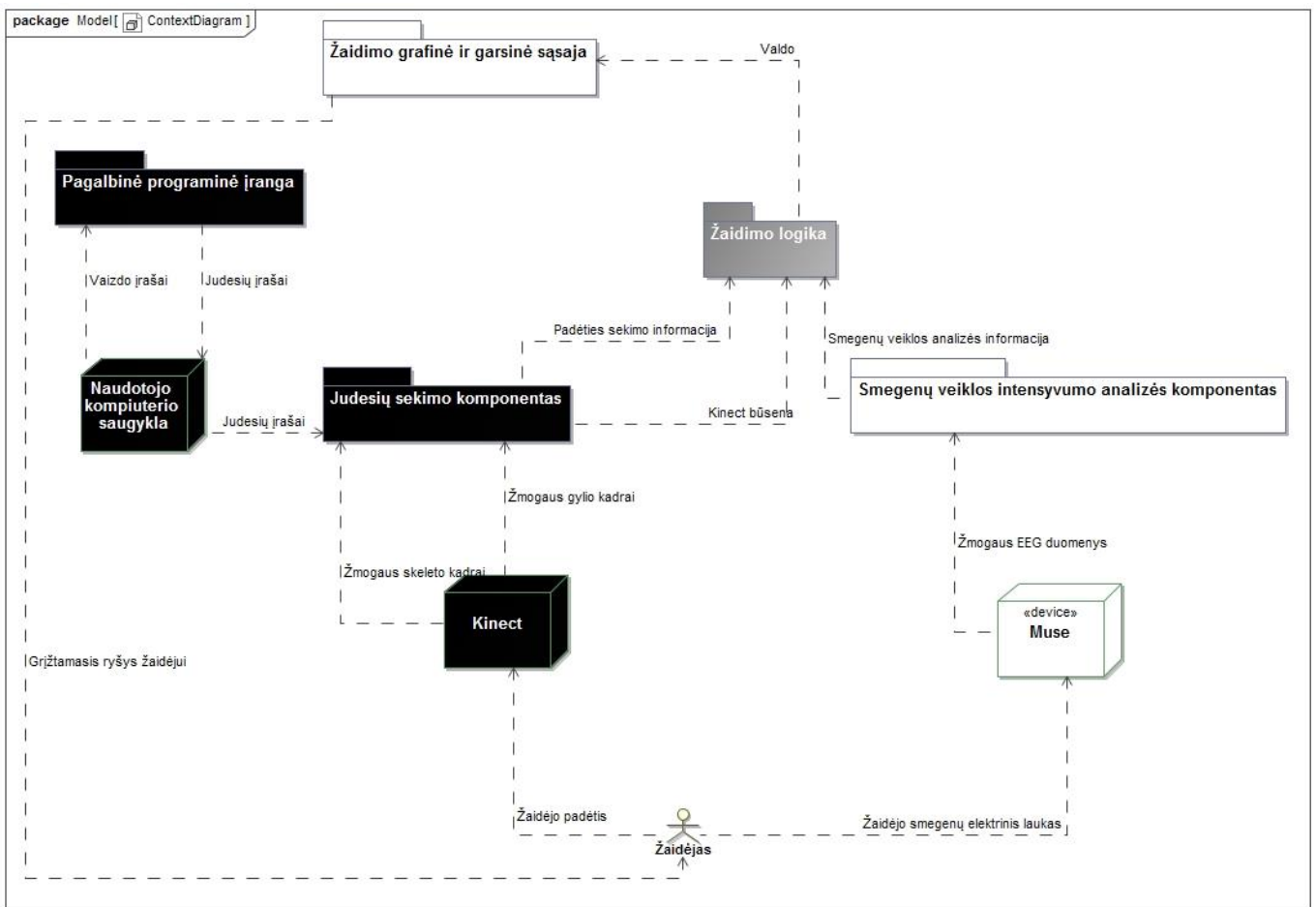
3. PROJEK TINĖ DALIS

3.1. Reikalavimai

3.1.1. Veiklos kontekstas

Judesių sekimo programinis komponentas naudojamas jogos žaidime. Šio komponento paskirtis – nustatyti pratimo atlikimo teisingumą. Savo ruožtu komponentas naudos „Kinect“ programinę įrangą, kurios pagalba pasieks duomenis, kuriuos pateikia „Kinect“. Sistemos kūrimo darbai apima ir komponento integraciją į žaidimą. Visa komponentų bendradarbiavimo schema pateikiama 3.1 pav. Juoda spalva pavaizduoti elementai (judesių sekimo komponentas, Kinect, naudotojo kompiuterio saugykla, pagalbinė programinė įranga) yra sistemos dalis, pilnai įgyvendinta šio darbo metu, pilka (žaidimo logika) – dalinai įgyvendinta darbo metu. „Kinect“ ir „Muse“ jutikliai surenka duomenis apie žmogaus dabartinę padėtį ir smegenų veiklą, atitinkami komponentai tuos duomenis apdoroja ir pateikia žaidimo logikos valdymui. Ten jais remiantis valdoma žaidimo grafinė ir garsinė sąsaja, per kurią pateikiamas grįžtamasis ryšys žaidėjui.

Padėties sekimo informacija susideda iš delno padėties, žmogaus padėties sekimo informacijos, įvertčio vartotojui. Judesių įrašus sudaro siluetaų sekos ir jų stabdos taškų informacija.



3.1 pav. Sistemos konteksto diagrama

3.1.2. Sistemos panaudos atvejų modelis

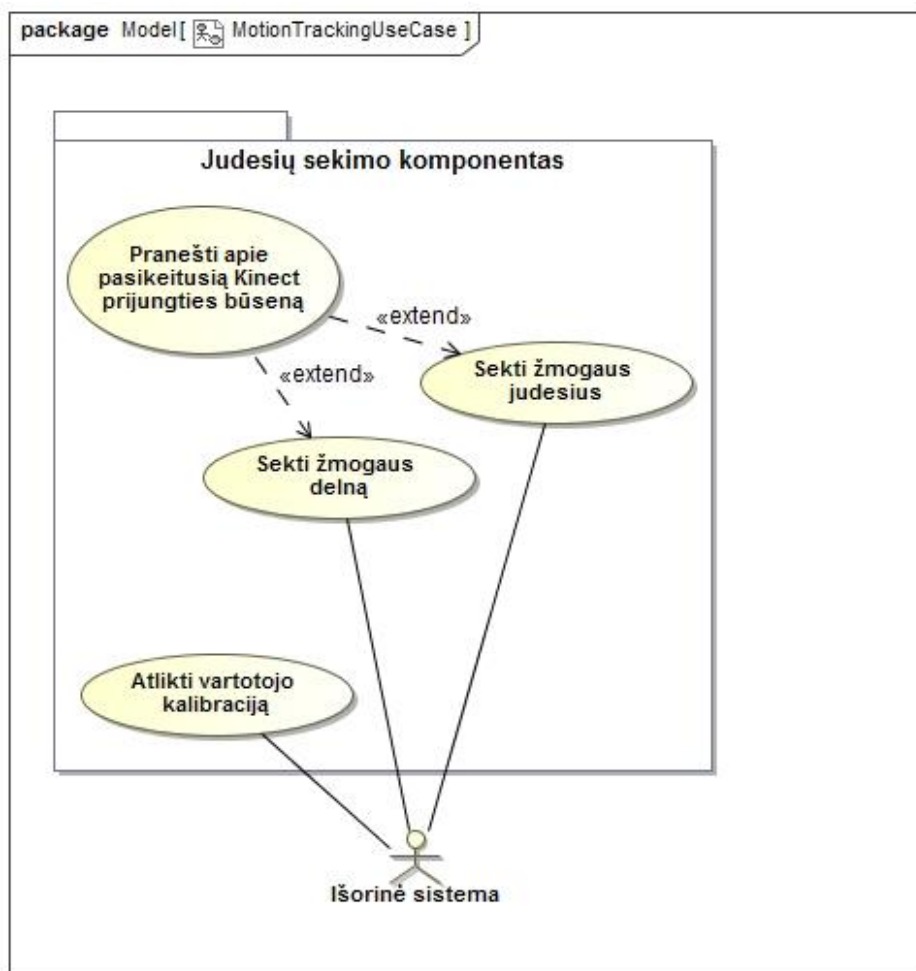
Sistema susideda iš trijų dalių – judesių sekimo komponento, jogos žaidimo, kuriame judesių sekimo komponentas naudojamas, ir pagalbinio įrankio, skirto paruošti įrašų failus. Kiekviena dalis turi atskirus naudotojus, todėl galima panaudos atvejus sugrupuoti pagal išskirtas dalis.

Judesių sekimo komponento naudotojas yra išorinė sistema, šiuo atveju – jogos žaidimas. Komponentas per savo sąsają leidžia išorinei sistemai naudotis trimis pagrindinėmis funkcijomis:

- Atlikti vartotojo kalibraciją, kurios metu nustatomas žmogaus lankstumo tipas;
- Sekti žmogaus delno padėtį;
- Sekti žmogaus judesius.

Sekimo metu taip pat galima gauti informaciją apie „Kinect“ jutiklio prijungimą ar atjungimą.

Judesių sekimo komponento panaudos atvejų diagrama pavaizduota 3.2 pav.



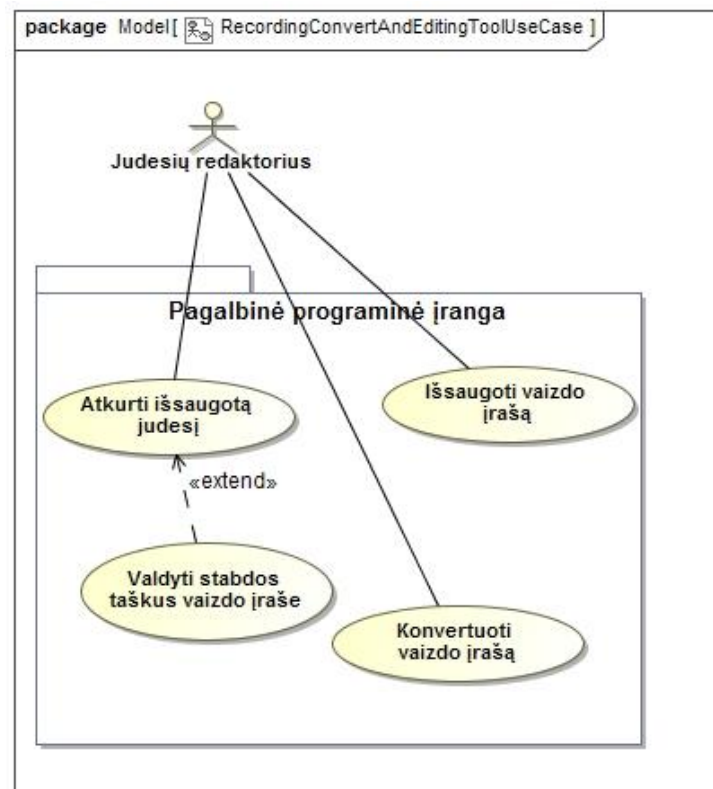
3.2 pav. Judesių sekimo komponento panaudos atvejų modelis

Judesių konvertavimo ir redagavimo įrankis yra skirtas paruošti judesių įrašus judesių sekimo komponentui priimtinu formatu. Šio įrankio atliekamos funkcijos:

- Konvertuoti specialiai paruoštą vaizdo įrašą į judesių sekimui naudojamą formatą .hdm;

- Atkurti išsaugotą judesį;
- Valdyti stabdos taškus įrašė;
- Išsaugoti įrašą.

Įrankį naudos judesių redaktorius. Šio įrankio panaudos atvejų diagrama pavaizduota 3.3 pav.



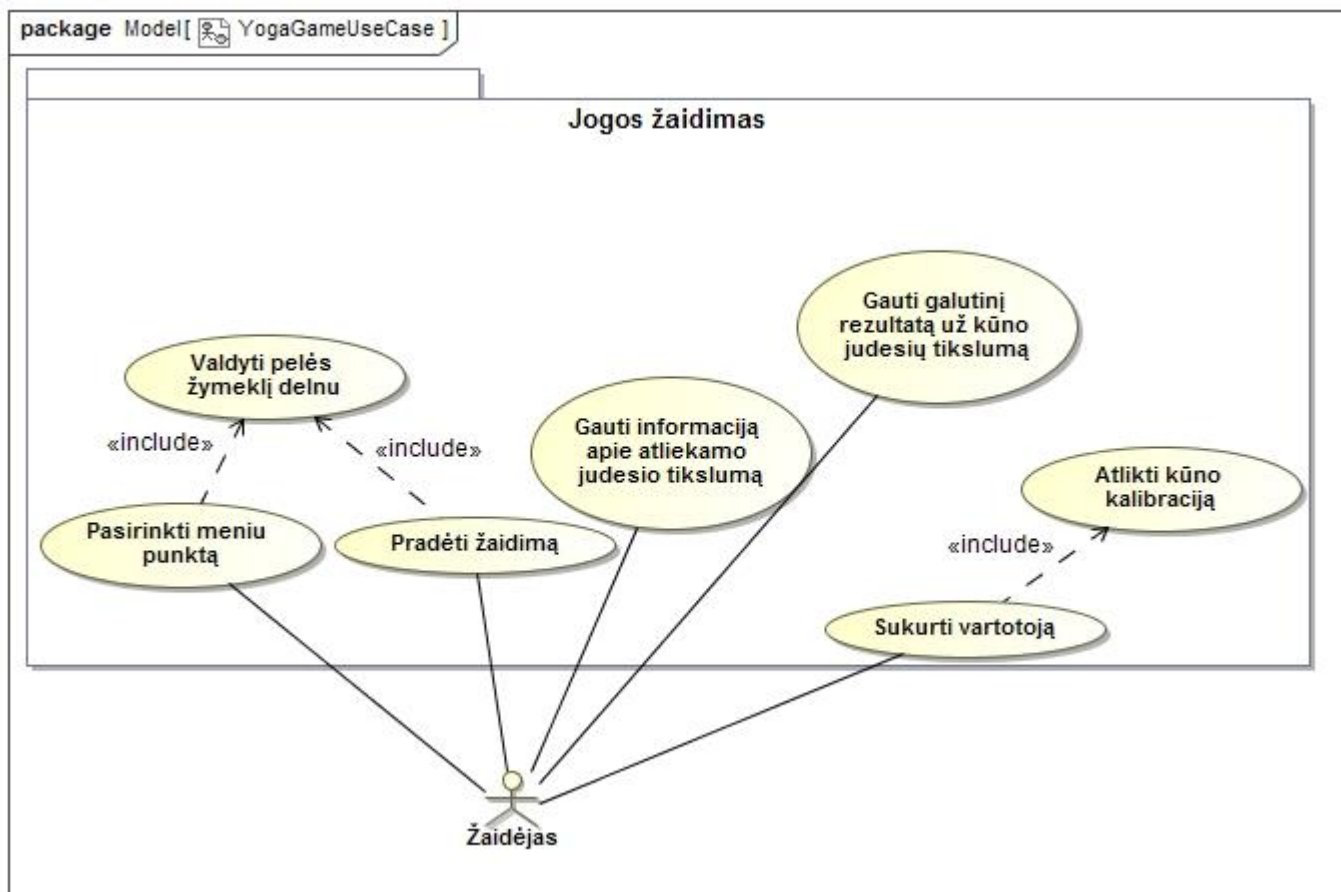
3.3 pav. Judesių konvertavimo ir redagavimo įrankio panaudos atvejų modelis

Darbo metu numatyta tik integruoti judesių sekimo komponentą į jogos žaidimą, viso jogos žaidimo kūrimas nėra šio darbo dalis. Dėl šios priežasties pateikiami tik tie panaudos atvejai, kurie tiesiogiai siejasi su komponento integracija žaidime.

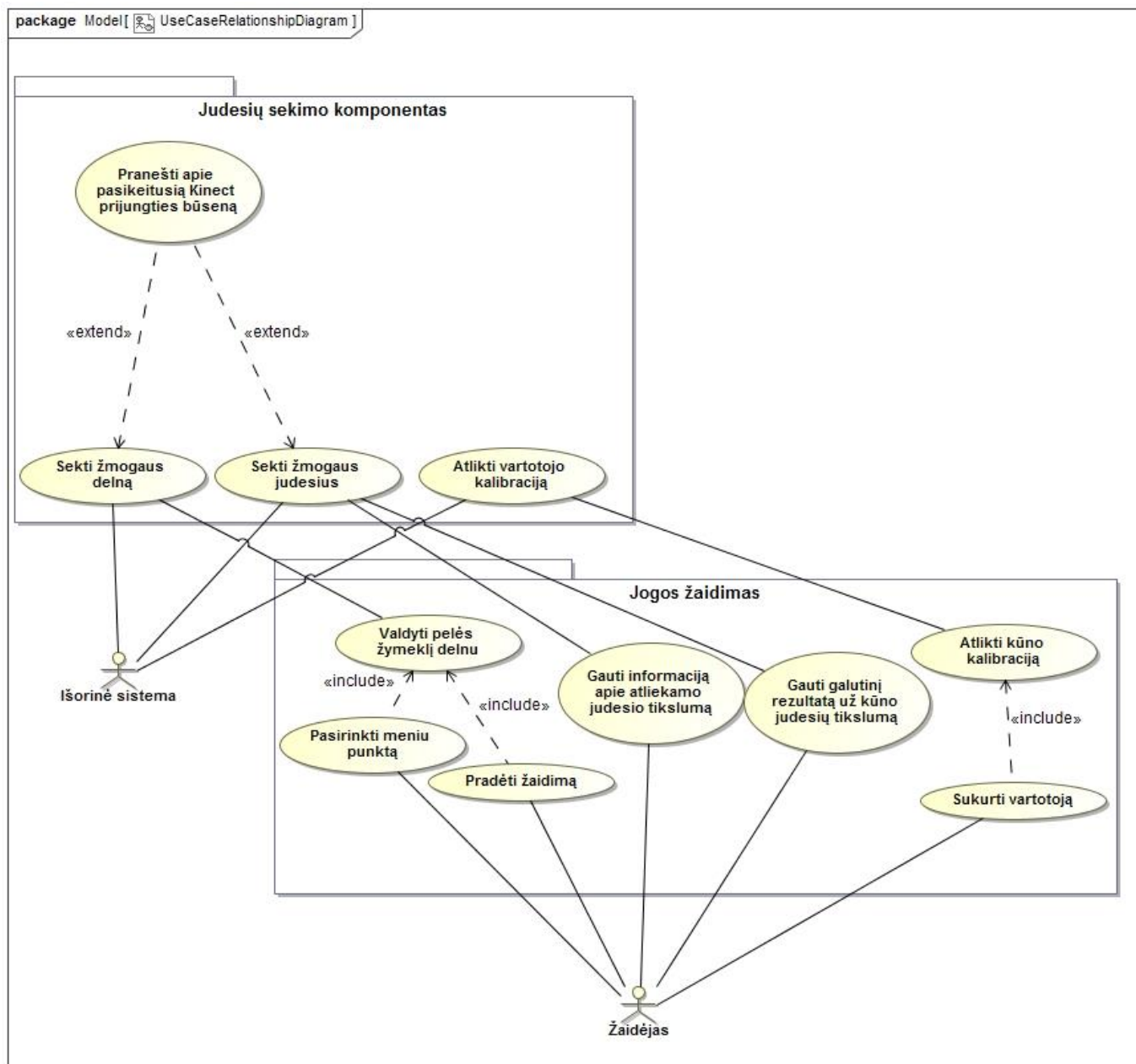
Žaidime numatytos tokios integravimo funkcijos:

- Valdyti pelės žymeklį delnu:
 - Pasirenkant meniu punktą;
 - Pradedant žaidimą.
- Sužinoti atliekamo judesio tikslumą;
- Sužinoti galutinį rezultatą;
- Atlikti kalibraciją sukuriant vartotoją.

Šie panaudos atvejai siejasi su judesių sekimo komponento panaudos atvejais, tik skiriasi naudotojas: judesių sekimo komponentą naudoja žaidimas, o žaidimą – žaidėjas. Jogos žaidimo dalinė panaudos atvejų diagrama parodyta 3.4 pav., judesių sekimo komponento ir jogos žaidimo panaudos atvejų sąsajos diagrama - 3.5 pav.



3.4 pav. Jogos žaidimo dalies, susijusios su judesių sekimu, panaudos atvejų modelis



3.5 pav. Judėsių sekimo komponento ir jogos žaidimo panaudos atvejų modelių sąsajos diagrama

3.1.3. Reikalavimai sistemai

Kuriamai sistemai keliami tokie reikalavimai:

1. Judėsių sekimo programinė įranga turi būti suderinama su „Unity“ žaidimų varikliu;
2. Judėsių sekimo programinė įranga turi veikti „Windows 8“, „Windows 8.1“ ir „Windows 10“ sistemose;
3. Judėsių sekimo komponentas turi palaikyti „Kinect“ ir „Kinect 2“ įrenginius;
4. Kalibracija turi pateikti vieną iš šių žmogaus lankstumo tipų: lankstus, vidutinis, nelankstus;
5. Kalibracijai įgyvendinti naudojamos trys jogos pozos, apimančios persilenkimus per skirtingas vietas;

6. Palyginimo progreso informacija turi leisti žaidimui žinoti, kuri dalis pratimo jau yra atlikta;
7. Sistema turi žaidimui pateikti, kokie yra skirtumai tarp žmogaus silueto ir šablono;
8. Sistema turi semantiškai nustatyti, kuri žmogaus kūno dalis yra pozicijuota neteisingai, jei tokių yra;
9. Galutinis vartotojo rezultatas turi būti pateikiamas režiuose nuo 0 iki 1, kur 0 – prasčiausias galimas įvertis, 1 – geriausias;
10. Tarpinis vartotojo rezultatas turi būti pateikiamas režiuose nuo 0 iki 1, kur 0 – prasčiausias galimas įvertis, 1 – geriausias;
11. Sistema žaidimui delno koordinatas turi pateikti žaidimo lango koordinatinių sistemoje;
12. Žaidimo pradžioje paleidžiamas judesių sekimas;
13. Turi būti galimybė inicijuoti judesių sekimą pačiam žaidimui;
14. Sistema turi pati įkelti judesių įrašą, pagal kurį bus vykdomas judesių palyginimas;
15. Turi būti galimybė nutraukti judesių sekimą;
16. Sistema turi pateikti judesių sekimo tarpinius rezultatus žaidimui;
17. Sistema turi suskaičiuoti galutinį vartotojo rezultatą;
18. Turi būti galimybė inicijuoti žmogaus delno sekimą;
19. Turi būti galimybė sustabdyti žmogaus delno sekimą;
20. Judesių redagavimo įrankis ir judesių sekimo komponentas turi naudoti bendrą duomenų formatą;
21. Konvertuojant judesio įrašą vartotojui turi būti rodomas konvertavimo progresas;
22. Judesys vizualizuojamas grafinėje sąsajoje tam skirtoje lango dalyje;
23. Atkuriant judesio įrašą turi būti galimybė prasukti įrašą laiko skalėje iki norimo momento;
24. Stabdos taškų valdymas vykdomas laiko skalėje;
25. Failas turi būti išsaugomas programos aplanke;
26. Judesių kadrų palyginimas turi vykti bent 25 kadrų per sekundę greičiu (40 ms vienam kadru);
27. Delno pozicijos atnaujinimas turi vykti bent 25 kartus per sekundę;
28. Sistema turi neversti žaidėjo persitempti;
29. Sistema turi veikti namų sąlygomis;
30. Sistema ateityje gali būti pritaikoma „XBox“ žaidimų konsolei.

Reikalavimai detaliam aprašyti priede 9.1 „Sistemos reikalavimai“.

3.2. Architektūra

3.2.1. Architektūros tikslai ir apribojimai

Architektūros tikslas – pateikti programinės įrangos modelį, tenkinantį keliamus apribojimus bei atitinkantį panaudojimo atvejų modelį bei reikalavimus.

Architektūrai keliami tokie reikalavimai:

- Programinė įranga turi veikti „Windows 8“, „Windows 8.1“ ir „Windows 10“ operacinėse sistemose;
- Judesių sekimo komponentas turi veikti „Unity“ aplinkoje;
- Judesių sekimo programinę įrangą turi būti nesunku perkelti į kitas operacines sistemas, pvz., „XBox“ žaidimų konsolę;
- Judesių sekimo komponentas turi veikti su „Kinect“ įrenginiais;
- Judesių sekimo komponentą turi būti paprasta pritaikyti darbui su nauju judesių sekimo įrenginiu;
- Judesių sekimo komponentą turi būti galima pakartotinai panaudoti kitoje programinėje įrangoje;
- Programinės įrangos kūrimui bus naudojama „Kinect SDK“, pritaikyta atitinkamai „Kinect“ versijai.

Architektūros kūrimui taikoma strategija: sukuriamas panaudos atvejų modelis, jo pagrindu sukuriamas analitinis modelis, iš jo – projektinis sistemos modelis, kurį sudaro statinis sistemos vaizdas. Šių etapų metu identifikuojamos esybės taikomosios srities lygiu (analitinis modelis) ir realizavimo lygiu (projektinis modelis). Toliau kuriamas dinaminis sistemos vaizdas. Esant reikalui papildomas statinis modelis.

Projektavimui naudojama „MagicDraw“ programinė įranga, „Libre Writer“ tekstų doroklė. Judesių sekimo komponentą projektuoja jo kūrėjas, jogos žaidimą - judesių sekimo komponento kūrėjas, minčių intensyvumo stebėsenos komponento kūrėjas, grafikos dizaineris, užsakovai.

3.2.2. Sistemos statinis vaizdas

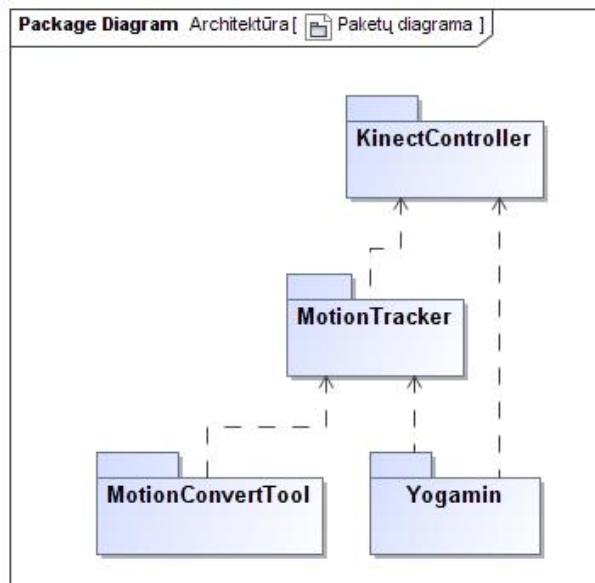
3.2.2.1. Sistemos paketai

Sistemą sudarys tokie paketai:

- „Kinect“ valdymo paketas. Jis skirtas komunikuoti su judesių sekimo įranga bei pateikti jo duomenis nuo tos įrangos nepriklausomu pavidalu;
- Judesių sekimo komponentas. Skirtas sekti žmogaus daromus judesius ir lyginti juos su šablono judesiu remiantis gylio žemėlapiu ar kūno indekso žemėlapiu;
- Jogos žaidimas. Programinės įrangos taikymo sritis;

- Judesių konvertavimo įrankis. Pagalbinis įrankis parengti judesių įrašus, naudojamus judesių sekimo komponente.

Paketų diagrama pateikiama 3.6 pav.



3.6 pav. Paketų diagrama

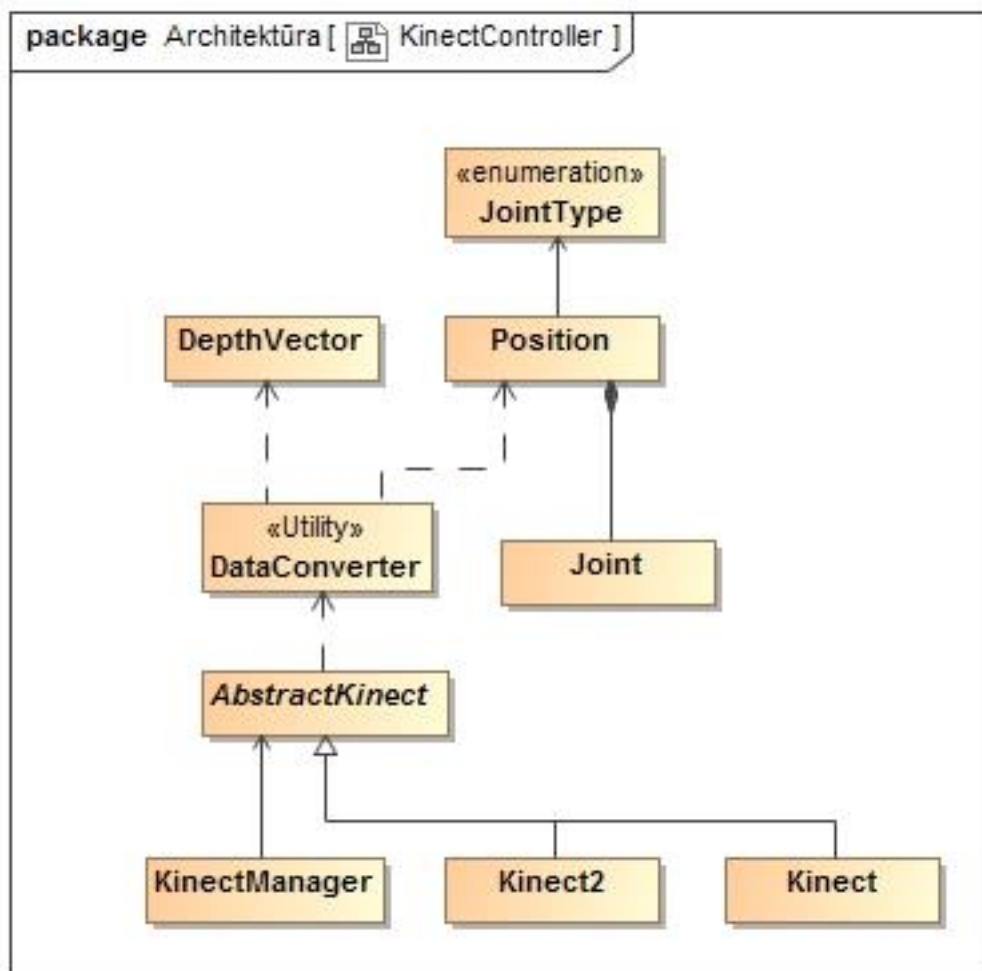
3.2.2.2. Sistemos paketų detalizacija

Šiame skyriuje pateikiama sistemos paketų detalizacija. Čia aprašomas kiekvienas paketas, pateikiamos svarbiausios klasės bei parodyta, kokiais ryšiais klasės yra susijusios. Detalios klasių diagramos pateikiamos priede 9.2 „Detalios sistemos klasių diagramos“.

Paketas „MotionTracker“ yra sudėtingiausias ir svarbiausias. Jis atsakingas už žmogaus judesių sekimą. Svarbiausios paketo klasės:

- Tracker – sąsajos klasė, per kurią išorinė sistema vykdo judesių sekimą;
- Calibrator – sąsajos klasė, per kurią išorinė sistema vykdo vartotojo kalibraciją;
- Movement ir agreguojamos klasės – aprašo judesių įrašo struktūrą;
- MotionTrackingSession – kaupia judesių sekimo duomenis;
- ProcrustesAnalyser, HausdorffMatchingAlgorithm – naudojamų sekimo algoritmų realizacijos;
- MotionTracker – vykdo dviejų kadrų palyginimą;
- HandTracker – sąsajos klasė, skirta rankos sekimui.

Paketas realizuoja sekimui naudojamus algoritmus, vykdo judesių sekimą. Jo klasių diagrama pateikta 3.7 pav.

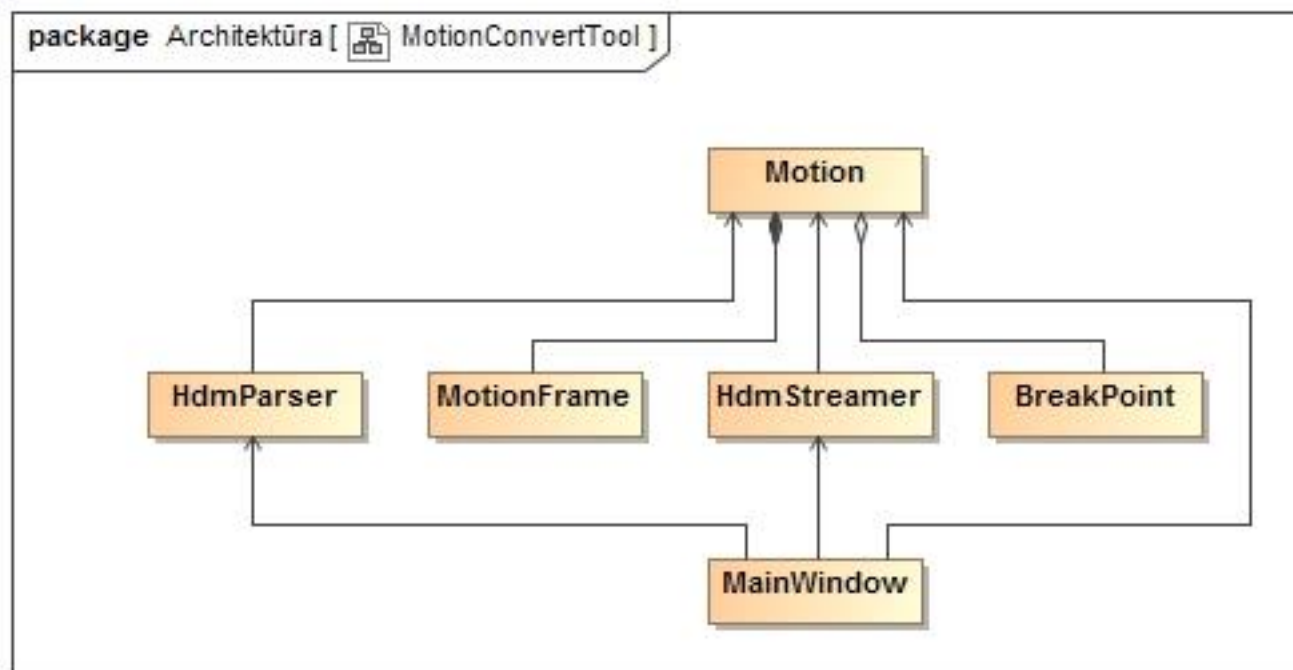


3.8 pav. Paketo „KinectController“ klasių diagrama

Paketo „MotionConvertTool“ architektūroje išsiskiria trys sluoksniai:

- Atvaizdavimo;
- Duomenų apdorojimo;
- Duomenų.

Atvaizdavimo sluoksnį sudaro klasė `MainWindow`. Ji atsakinga už vartotojo sąsają ir duomenų vaizdavimą joje. Duomenų apdorojimo sluoksnį sudaro klasės `HdmParser` ir `HdmStreamer`. Šios klasės skirtos įrašo konvertavimui ir atkūrimui. Duomenų klasės yra `Motion`, `MotionFrame` ir `BreakPoint`. Jos aprašo duomenis, kuriais manipuliuoja sistema. Paketo klasių diagrama pavaizduota 3.9 pav.



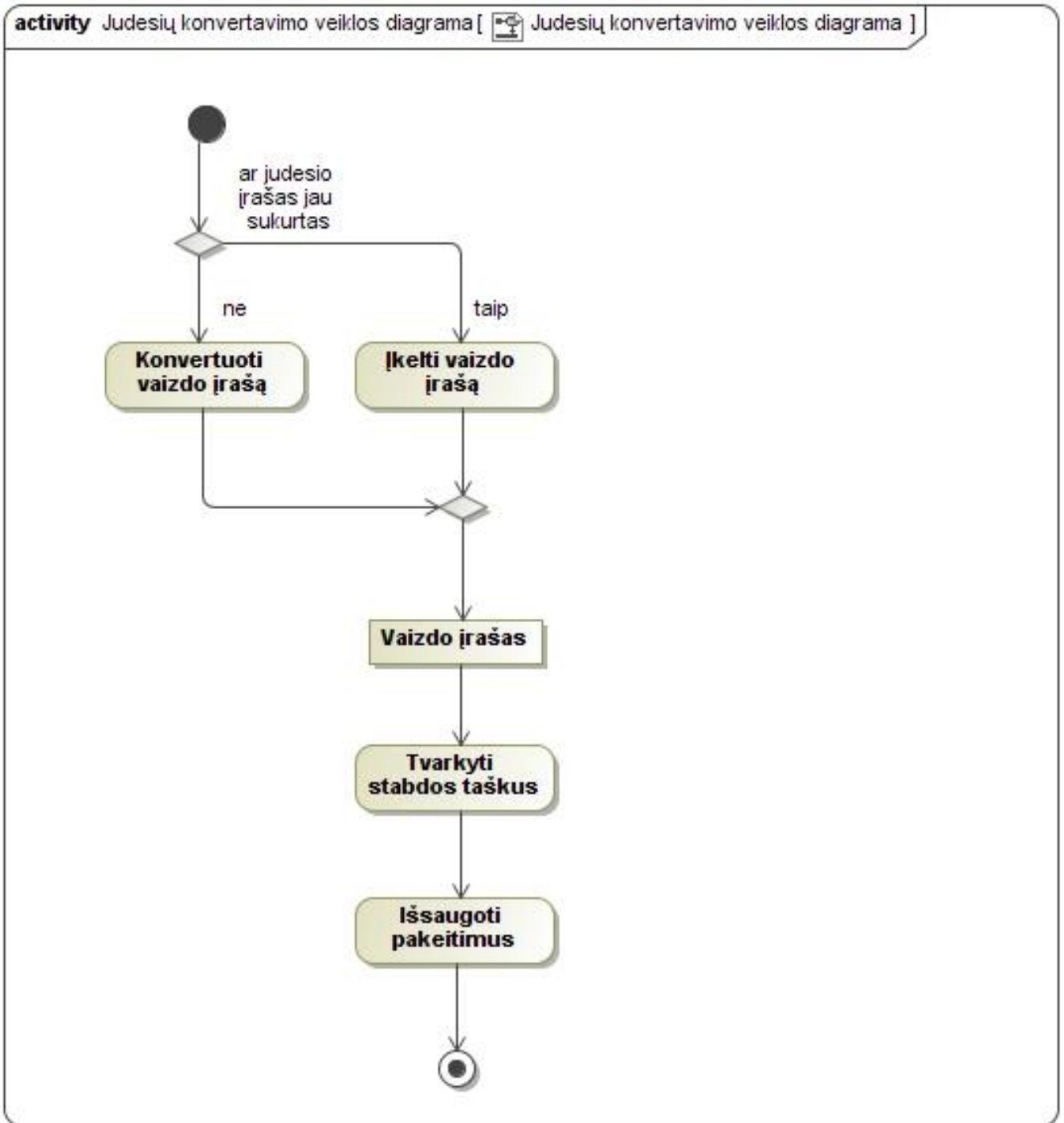
3.9 pav. Paketo „MotionConvertTool“ klasių diagrama

3.3. Sistemos dinaminis vaizdas

Šiame skyriuje apžvelgiami pagrindiniai sistemos naudojimo scenarijai:

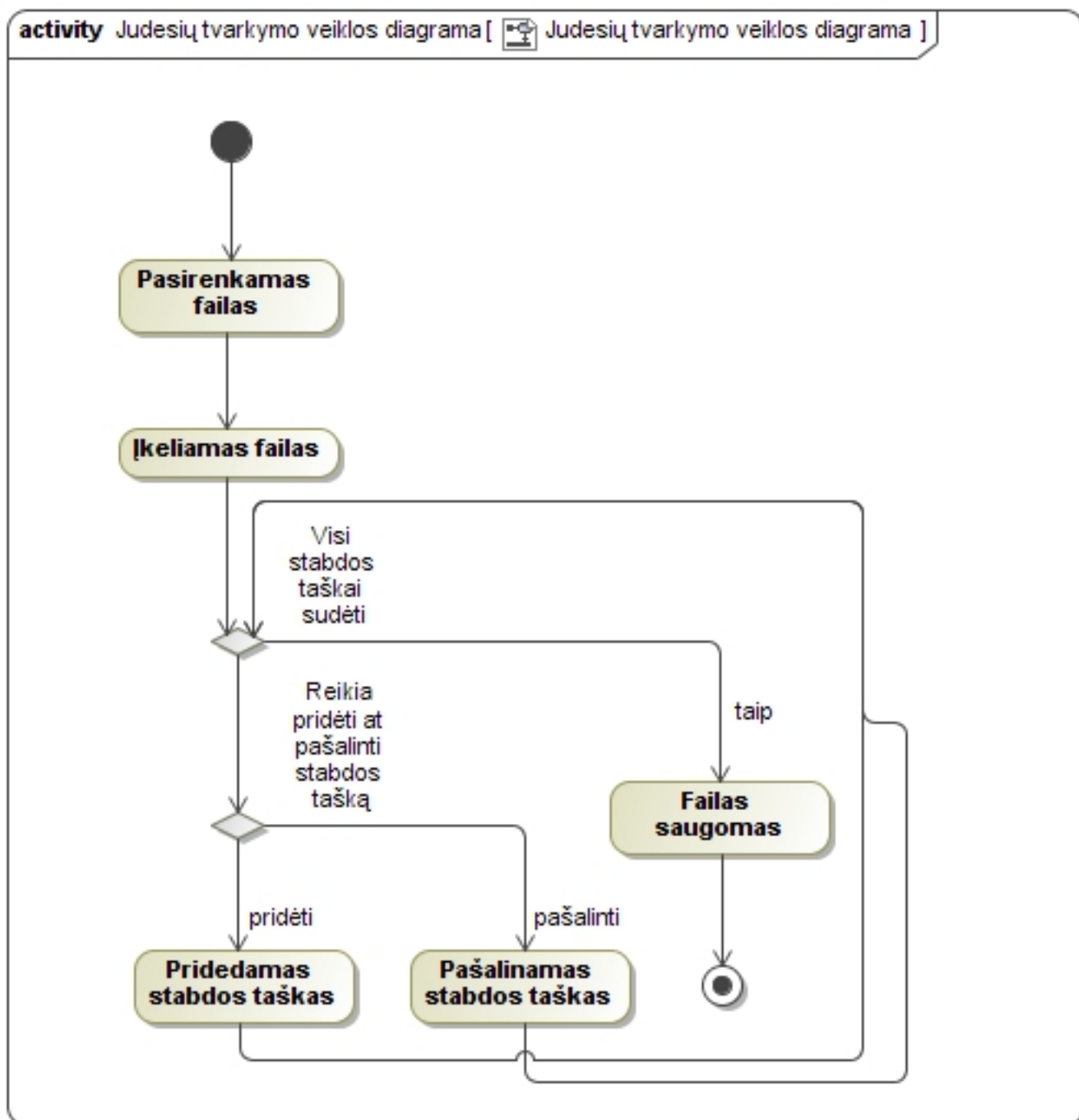
- Judesių konvertavimo ir tvarkymo įrankiui:
 - Judesių konvertavimas;
 - Judesių tvarkymas.
- Jogos žaidimui:
 - Pagrindinis veiklos scenarijus.

Judesių konvertavimas vykdomas pasirenkant vaizdo įrašą arba judesių įrašą. Jeigu pasirinktas vaizdo įrašas, jis pirmiausia turi būti konvertuotas į judesių įrašą. Atlikus šiuos veiksmus sistemoje yra įkeltas judesių įrašas. Tuomet jame galima tvarkyti stabdos taškus ir įrašyti pakeitimus. Judesių konvertavimo veiklos diagrama pateikiama 3.10 pav.



3.10 pav. Judesių konvertavimo veiklos diagrama

Judesių tvarkymui pirmiausia reikia pasirinkti ir įkelti failą. Tuomet pridedami ar pašalinami stabdos taškai. Kai jie visi yra sudėti, įrašas su pakeitimais išsaugomas. Judesių įrašo tvarkymo veiklos diagrama pateikiama 3.11 pav.

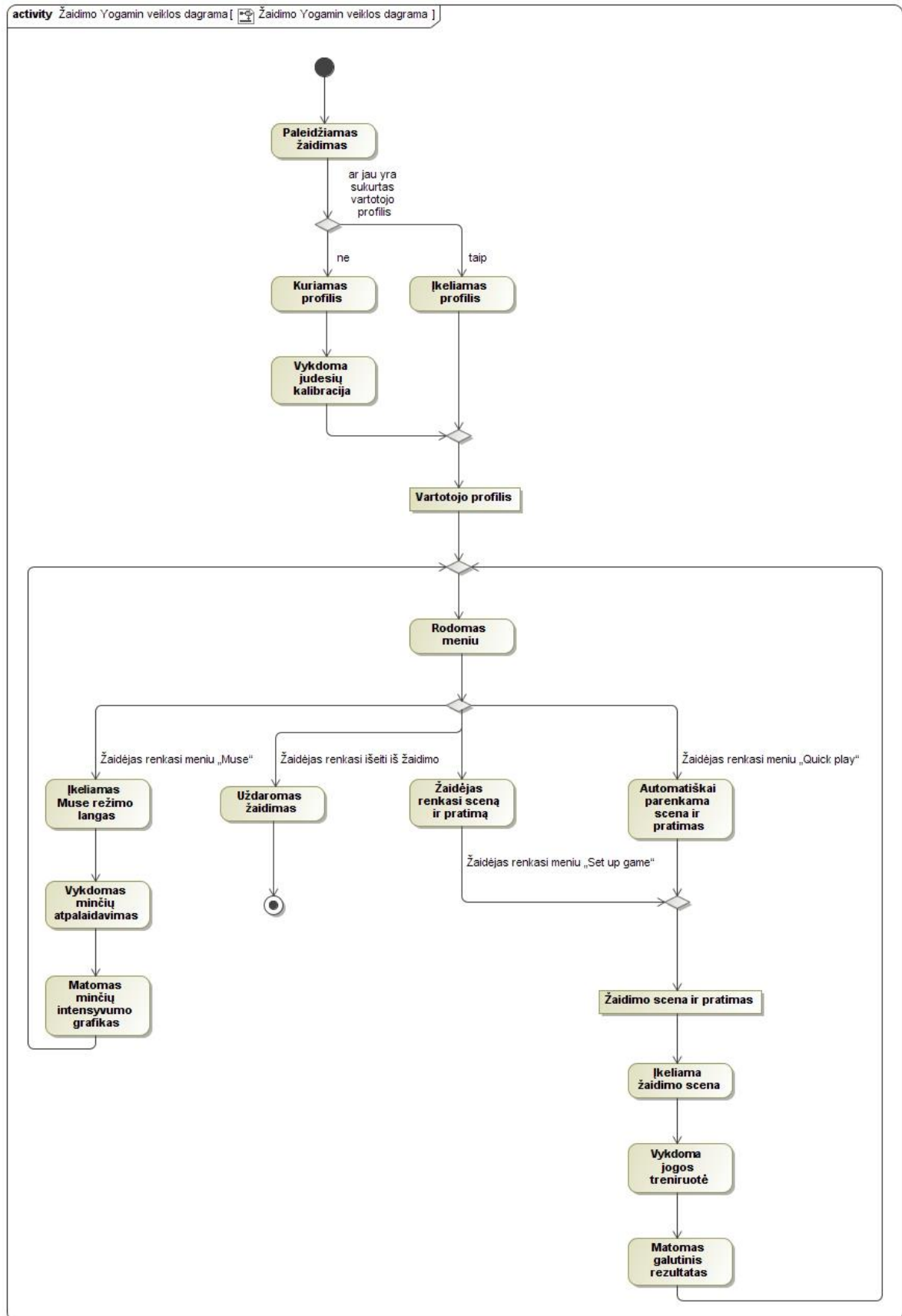


3.11 pav. Judesių tvarkymo veiklos diagrama

Jogos žaidimas prasideda nuo jo paleidimo. Jeigu vartotojas dar nėra susikūręs profilio, tai jis sukuriamas. Tuomet vartotojas patenka į meniu langą. Meniu lange jis renka viena iš meniu punktų:

- „Muse“ – skirtas minčių atpalaidavimui, į šio darbo apimtį neįeina;
- „Greitas žaidimas“ – automatiškai parenkama scena ir pratimas ir pradeda treniruotė;
- „Pasirinktinis žaidimas“ – vartotojas pasirenka sceną ir pratimą, tuomet pradeda treniruotė;
- „Išeiti“ – uždaromas žaidimas.

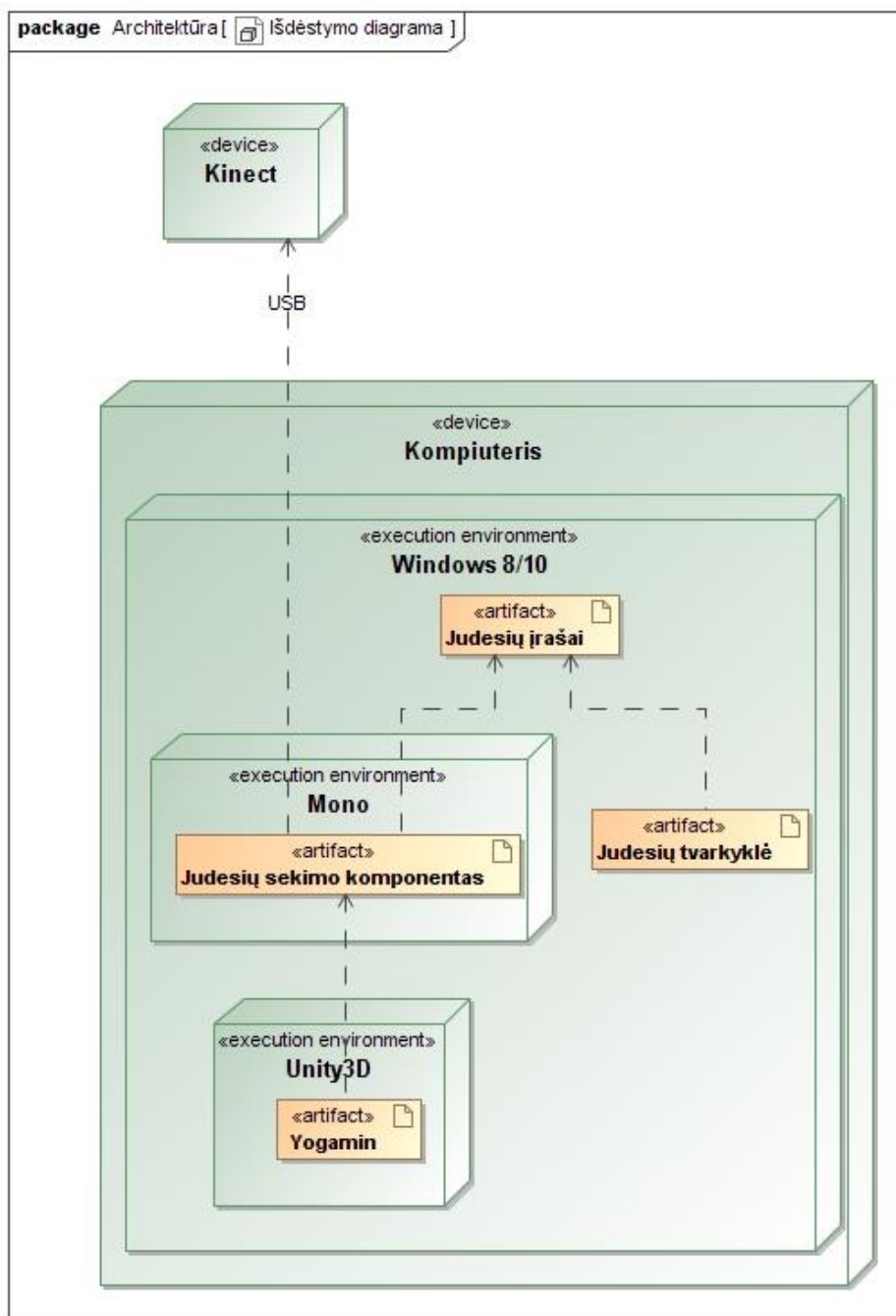
Po treniruotės rodomas galutinis rezultatas. Žaidimo veiklos diagrama parodyta 3.12 pav.



3.12 pav. Jogos žaidimo veiklos diagrama

3.4. Sistemos išdėstymo vaizdas

Sistema veikia kompiuteryje su „Windows“ operacine sistema. Joje veikia „Unity3D“ žaidimų variklis, kuris vykdo jogos žaidimą, bei „Mono“ aplinka, kurioje veikia judesių sekimo komponentas. Žaidimas naudoja judesių sekimo komponentą. Jam reikalingi vaizdo įrašai. Judesių tvarkyklė taip pat naudoja vaizdo įrašus, bet veikia kaip atskira programa „Windows“ aplinkoje. Judesių sekimo komponentas per USB sąsają bendrauja su „Kinect“. Išdėstymo vaizdas parodytas 3.13 pav.



3.13 pav. Sistemos išdėstymo diagrama

4. TYRIMO DALIS

Skyriuje 2.5 „Judėsių ir pozicijų palyginimo algoritmų analizė“ buvo nustatyta, kad keliamam judėsių sekimo uždaviniui geriausiai tinka Hausdorfo atstumo metrika. Ją patogu taikyti, kai analizuojami duomenys yra dvejetainės matricos pavidalo, t. y., matricos, kurios kiekvienas elementas lygus 0 arba 1. Metrikos skaičiavimui nereikia išskirti reikšminių taškų, galima lyginti skirtingus taškų kiekius turinčias matricas. Skyriuje 2.6 „Esami Hausdorfo atstumo skaičiavimo metodai“ buvo nustatyta, kad nėra Hausdorfo atstumo skaičiavimo algoritmo, kuris tenkintų visus taikomosios srities keliamus reikalavimus. Dėl šių priežasčių reikia algoritmo, galinčio efektyviai dirbti su „Kinect“ duomenimis.

Taikant Hausdorfo atstumo skaičiavimus žmogaus ir įrašo padėčių sulyginimui kyla dar viena problema – kadruose žmogaus ir šablono aukštis ir padėtis gali skirtis. Tai priklauso nuo žmogaus padėties „Kinect“ atžvilgiu, bet negali turėti įtakos metrikų apskaičiavimui. Dėl šios priežasties bus reikalinga ir Prokrusto analizė. Abu algoritmai yra didelio algoritminio sudėtingumo, todėl reikalingos optimizacijos, padidinančios jų skaičiavimo efektyvumą.

4.1. Prokrusto analizės optimizacija

Tarkime, kad turime dvi baigtines taškų aibes $A = \{a_1, a_2, \dots, a_n\}$ ir $B = \{b_1, b_2, \dots, b_m\}$. Darome prielaidą, kad žmogus kadre stovės orientuotas taip, kad pagrindo, ant kurio yra žmogus, plokštumos vektorius yra orientuotas vertikalia kryptimi. Ši prielaida eliminuoja pasukimo transformaciją. Tuomet reikia atlikti dvi tiesines transformacijas: mastelio keitimo ir postūmio. Tarkime, kad aibės B centrinis taškas, kurio atžvilgiu bus keičiamas mastelis, yra b_m , optimalus mastelio keitimo koeficientas – s , o optimalios postūmio transformacijos vektorius \vec{t} . Tuomet transformuota taškų aibė

$$B_t = \{b_{t1}, b_{t2}, \dots, b_{tn}\} = b_m + \frac{(b_1 - b_m + \vec{t})}{s}, b_m + \frac{(b_2 - b_m + \vec{t})}{s}, \dots, b_m + \frac{(b_n - b_m + \vec{t})}{s} \quad (31).$$

Po optimalios transformacijos aibė B_t turi maksimizuoti kardinalumo funkciją i :

$$i(A, B_t) = \frac{2|A \cap B_t|}{|A| + |B_t|} \quad (32).$$

Šios funkcijos maksimali galima reikšmė yra 1, kuri pasiekama, kai visi aibių taškai sutampa, o minimali reikšmė – 0, kai nei vienas aibių taškas nesutampa. Tokios funkcijos maksimalios reikšmės tikslus radimas yra imlus skaičiavimams, todėl tikslinga taikyti euristinius algoritmus.

Greitai apskaičiuojama ir pakankamai tiksli euristika mastelio keitimo transformacijai yra mastelio keitimo koeficientu laikyti abiejų objektų aukščių santykį. Užtikrinus, kad šablonas ir žmogaus siluetas bus panašiose padėtyse, galima suvienodinti abiejų siluetų aukščius. Tuomet

$$s = \frac{h_A}{h_B} \quad (33),$$

kur

$$h_A = \max_{a \in A} a_y - \min_{a \in A} a_y \quad (34),$$

kur a_y - vertikaliosios ašies koordinatė.

Greitai apskaičiuojamas postūmio transformacijos algoritmas yra perstumti vieną iš siluėtų per vektorių, lygų skirtumui tarp siluėtų centroidų. Centroidas apibrėžiamas kaip

$$A_C = \left(\frac{\sum_{a \in A} a_x}{n}; \frac{\sum_{a \in A} a_y}{n} \right) \quad (35).$$

Tuomet

$$b_i = C_B + \frac{(b_i - C_B + C_B - C_A)}{s} \quad (36).$$

Jeigu šablono ir žmogaus siluetai yra iš tos pačios judesių sekos, tai galima laikyti, kad transformacijos parametrai nesikeis viso palyginimo eigoje ir juos galima apskaičiuoti judesių palyginimo pradžioje ir vėliau naudoti, taip pagreitinant skaičiavimus. Tuomet

$$t = C_{B1} - C_{A1} \quad (37),$$

$$s = \frac{h_2}{h_1} \quad (38),$$

$$b_i = C_B + \frac{(b_i - C_B + t)}{s} \quad (39).$$

4.2. Hausdorfo atstumo skaičiavimo optimizacija

Siūlomas algoritmas remiasi Yaniro Taflevo idėjomis, tačiau yra žymiai geriau optimizuotas greičiui. Naudojant šį algoritmą nereikia duomenų konversijos iš dvimatės matricos į kitus formatus, pvz., taškų sąrašą. Naudojama atstumo metrika – Čebyševio atstumas. Pirmiausia sudaroma dvimatė matrica, kurios visi taškai su tomis koordinatėmis, kuriose siluete yra taškas, užpildomos 0, kiti - ∞ . Sudaroma taškų eilė, kurioje iš pradžių surašomi visi taškai, esantys siluete. Iš eilės imamas taškas ir jam skaičiuojami 8 aplinkiniai taškai, kurių viena ar abi koordinatės skiriasi 1 nuo nagrinėjamo taško. Visiems 8 taškams Čebyševio atstumas lygus centrinio taško Čebyševio atstumui, padidintam 1, jei neviršija tam taškui jau įrašytos reikšmės. Jei gautas mažesnis atstumas, nei jau buvo įrašytas, toks taškas taip pat įrašomas į eilę. Tokiu būdu apskaičiuojamas atstumas visiems taškams. Tuomet pašalinami visi taškai, kurie neegzistuoja antroje aibėje, nes jie neaktualūs. Tokios matricos elementai parodo skirtumus tarp siluėtų. Reikia surasti dvi tokias matricas, bet jų skaičiavimai nepriklausomi, todėl juos galima vykdyti lygiagrečiai.

Kad būtų mažiau skaičiavimų, nukerpami matricos kraštai, kuriuose nėra silueto taškų. Gauta apkirpta matrica yra minimali stačiakampė matrica, kurioje yra visi taškai, priklausantys siluetai.

Abiejų matricų bendras maksimumas yra Hausdorfo atstumas. Algoritmo pseudokodas:

```
matrices ← template matrix and user matrix
```

```
find common left upper corner of matrices
```

```
find common right lower corner of matrices
```

```
∀ matrix ∈ matrixes
```

```
do initialize queue and ret
```

```
while queue ≠ ∅
```

```
do p ← take from queue
```

```
val ← get p value from ret
```

```
val ← val + 1
```

```
ret ← min(r, val) ∀ adjacent point r to p
```

```
set points of res that do not belong to opposing matrix to ∅
```

```
hausdorff distance = max(max(res[1]), max(res2))
```

5. EKSPERIMENTINĖ DALIS

5.1. Eksperimentinio tyrimo tikslas

Eksperimentinio tyrimo tikslas – nustatyti, ar siūlomi algoritmai atitinka jiems keliamus tikslumo bei greitaveikos reikalavimus, taip pat palyginti jų greitaveiką su kitais egzistuojančiais sprendimais.

5.2. Eksperimentinio tyrimo aprašymas

Eksperimentinio tyrimo metu nustatomos tokios algoritmų savybės:

- Euristinės Prokrusto transformacijos įtaka Hausdorfo atstumo metrikos tikslumui lyginant su persiklojimą maksimizuojančia transformacija;
- Euristinės Prokrusto transformacijos koeficientų (mastelio keitimo ir tiesinio postūmio) nuokrypis nuo persiklojimą maksimizuojančios transformacijos koeficientų;
- Siūlomo Hausdorfo algoritmo skaičiavimo metodo veikimo greičio priklausomybės nuo kadro dydžio nustatymas taikant dvi algoritmo variacijas – su Prokrusto analize ir su iš anksto apskaičiuotais Prokrusto analizės koeficientais;
- Siūlomo Hausdorfo algoritmo skaičiavimo metodo veikimo greičio palyginimas su kitais egzistuojančiais sprendimais.

Eksperimentams naudojamas kompiuteris su tokia programine ir aparatūrine įranga:

- „Windows 10“ operacinė sistema;
- „Intel i7-3770K“ procesorius (4 fiziniai branduoliai, 8 virtualūs branduoliai, 3,5 GHz taktinis dažnis);
- 16 GB DDR3 tipo operatyviosios atmintinės, taktinis dažnis 1600 MHz;
- Kuriamam algoritmui – „.NET Framework 4.5.2“.

5.2.1. Euristinės Prokrusto transformacijos įtakos Hausdorfo metrikos tikslumui įvertinimas

Šio eksperimento metu bus įvertinama, kokia paklaida gaunama taikant apytikslį siūlomą Prokrusto analizės koeficientų radimo algoritmą. Paklaidoms nustatyti atliekami tokie veiksmai:

1. Parenkamos figūros palyginimui;
2. Surandama optimali transformacija kiekvienai figūrai;
3. Surandamas Hausdorfo atstumas taikant surastą transformaciją;
4. Surandama euristinė transformacija;
5. Surandamas Hausdorfo atstumas taikant euristinę transformaciją;
6. Apskaičiuojama gauta Hausdorfo atstumo metrikos paklaida.

Naudojami žmogaus, darančio jogos pratimą, siluetai. Pasirinkta jogos poza – pasilenkimas į šoną. Pratimo įrašas sudarytas iš dviejų siluetų sekų – šablono ir tą patį pratimą darančio kito žmogaus.

Fiksuojama po 270 kadrų 100 milisekundžių intervalu ir atliekamas palyginimas tarp gautos Hausdorfo atstumo metrikos reikšmės abiem atvejais. Paklaida įvertinama procentais nuo optimalios reikšmės.

Optimalios reikšmės radimas nėra trivialus. Kadangi nėra žinoma priklausomybė tarp transformacijos koeficientų ir aibių persidengimo lygio, tai nežinoma ir analitinė formulė, pagal kurią būtų galima maksimizuoti šią funkciją. Dėl šios priežasties randamas pseudo-optimalus sprendinys – pasirenkamas pradinis sprendinys ir taikomas atkaitinimo modeliavimo algoritmas (angl. *simulated annealing*) [29]. Taikomi tokie parametrai:

- Koeficientas β temperatūros mažinimui – 0,1;
- Iteracijų be pagerinimo kiekis – 25 000;
- Objektinė funkcija – kardinalumo funkcija pagal (32) formulę, tik taikomas priešingas ženklas, padalinama iš 10, nes didelė erdvė kadre yra tuščia, ir pridedama 1, kad 0 atitiktų optimalų rezultatą, 1 – visiškai netinkantį.

5.2.2. Euristinės Prokrusto transformacijos koeficientų tikslumo įvertinimas

Šio eksperimento metu tiriama, koks gaunamas nuokrypis nuo optimaliu būdu suskaičiuotų koeficientų. Paklaidoms nustatyti atliekami tie patys veiksmai, kaip ir euristinės Prokrusto transformacijos įtakos Hausdorfo metrikos tikslumui įvertinimui, tik skaičiuojami ne gaunami Hausdorfo atstumai, o tiesinio postūmio vektoriai ir dydžio keitimo koeficientai.

5.2.3. Siūlomo Hausdorfo algoritmo veikimo greičio priklausomybės nuo tiriamo kadro dydžio tyrimas

Šio tyrimo metu naudojami du kadrai, parinkti iš pirmajame ir antrajame eksperimentuose naudotų siluetų sekų. Sukuriami pakeisto dydžio kadrai nuo 0,1 iki 5 originalaus kadro dydžių 0,1 intervalu. Gaunama 50 kadrų porų. Kiekvienai kadrų porai skaičiuojamas Hausdorfo atstumas. Laikas matuojamas dviem būdais – įvertinant euristinės Prokrusto analizės vykdymo laiką ir jo nevertinant. Pastarasis atvejis modeliuoja atvejį, kai transformacijos parametrai jau yra apskaičiuoti ir vėliau tik naudojami palyginimo metu. Gauti vykdymo laikai palyginami, nustatoma, su kokio dydžio kadrais gali dirbti algoritmas, nustatoma, kokią įtaką skaičiavimo laikui turi euristinė Prokrusto analizė.

5.2.4. Hausdorfo atstumo skaičiavimo algoritmų greitaveikos palyginimas

Tyrimas yra toks pat, kaip ir 5.2.3 skyriuje aprašytas eksperimentas, tik Hausdorfo atstumas skaičiuojamas taikant šiuos algoritmus:

- Siūlomą Hausdorfo atstumo skaičiavimo algoritmą nevertinant Prokrusto analizės trukmės;
- Yaniro Taflevo algoritmą;
- „RgResolve“ algoritmą;

- „Elastic Search“ algoritmą.

Gauti vykdymo laikai palyginami, nustatoma, su kokio dydžio kadrais gali dirbti algoritmai.

5.3. Eksperimentinio tyrimo rezultatai

5.3.1. Euristinės Prokrusto transformacijos įtakos Hausdorfo metrikos tikslumo įvertinimo rezultatai

Eksperimento metu nustatyta, kad gaunamos paklaidos priklauso nuo pozų, kuriose yra šablonas ir žmogaus siluetas. Gautus rezultatus galima suskirstyti į dvi kategorijas:

- Abu siluetai yra paprastoje stovimoje padėtyje;
- Siluetai yra perėję į kitas padėtis.

Kai siluetai yra standartinėse padėtyse, euristinė Prokrusto analizė veikia žymiai geriau, nei kitais atvejais. Viso pratimo paklaidų statistika:

- Vidutinis nuokrypis – 21%;
- Didžiausia paklaida – 54%;
- Vidurkis – 19%;
- Mediana – 6%;
- Tiksliai įvertintų kadrų dalis – 44%;
- Duomenų imtis – 270 siluetų porų.

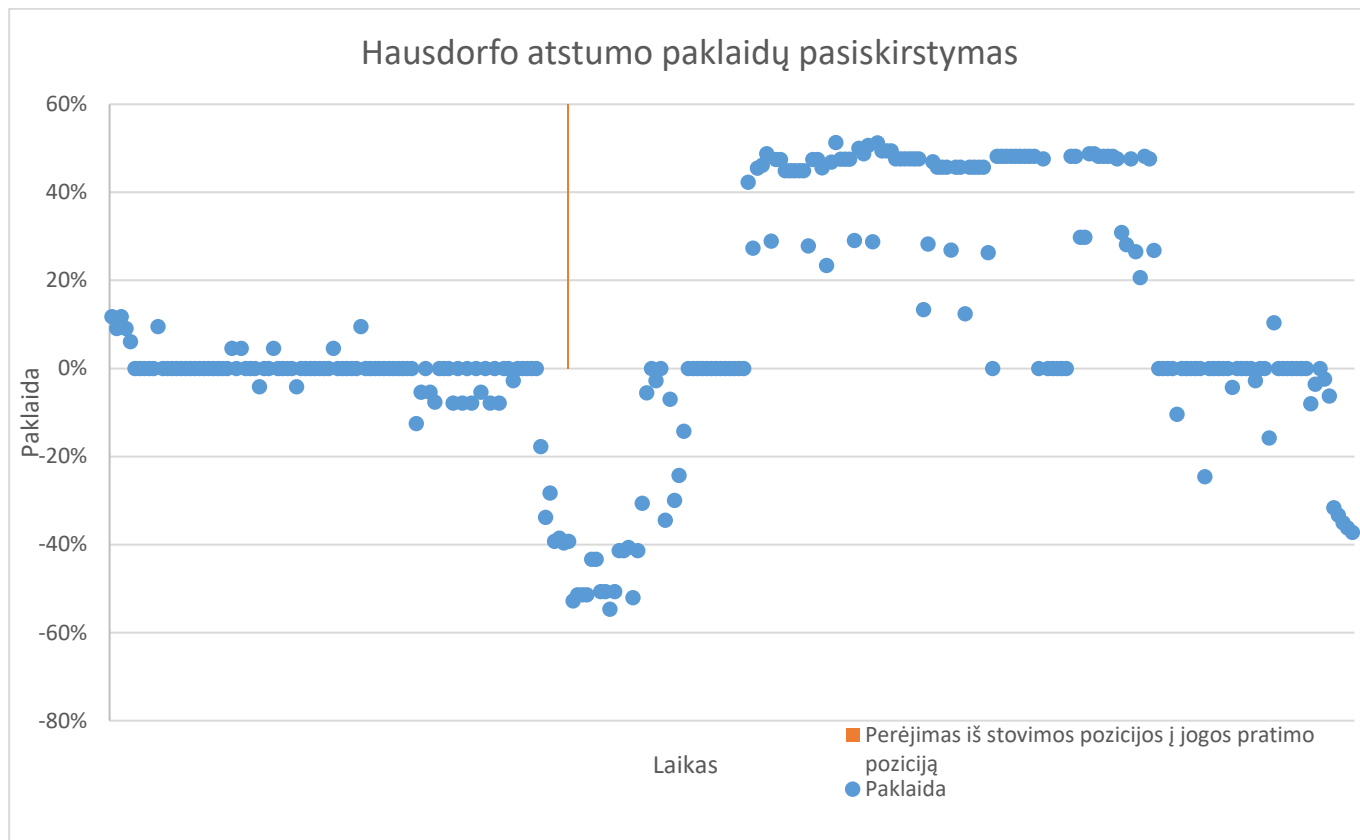
Gautų paklaidų vidurkis gerokai viršija 5% ribą, mediana taip pat yra didesnė nei 5%. Didelis vidutinis nuokrypis rodo, kad rezultatai gaunami labai skirtingi – tiek labai tikslūs, tiek labai netikslūs. Tokie rezultatai rodo, kad šios euristikos taikymas viso pratimo metu yra netinkamas, nes galima gauti labai netikslų rezultatą.

Jei vertintume tik įžanginę pratimo dalį, kai abu siluetai yra standartinėse padėtyse, rezultatai žymiai tikslesni. Įžanginės pratimo dalies paklaidų statistika:

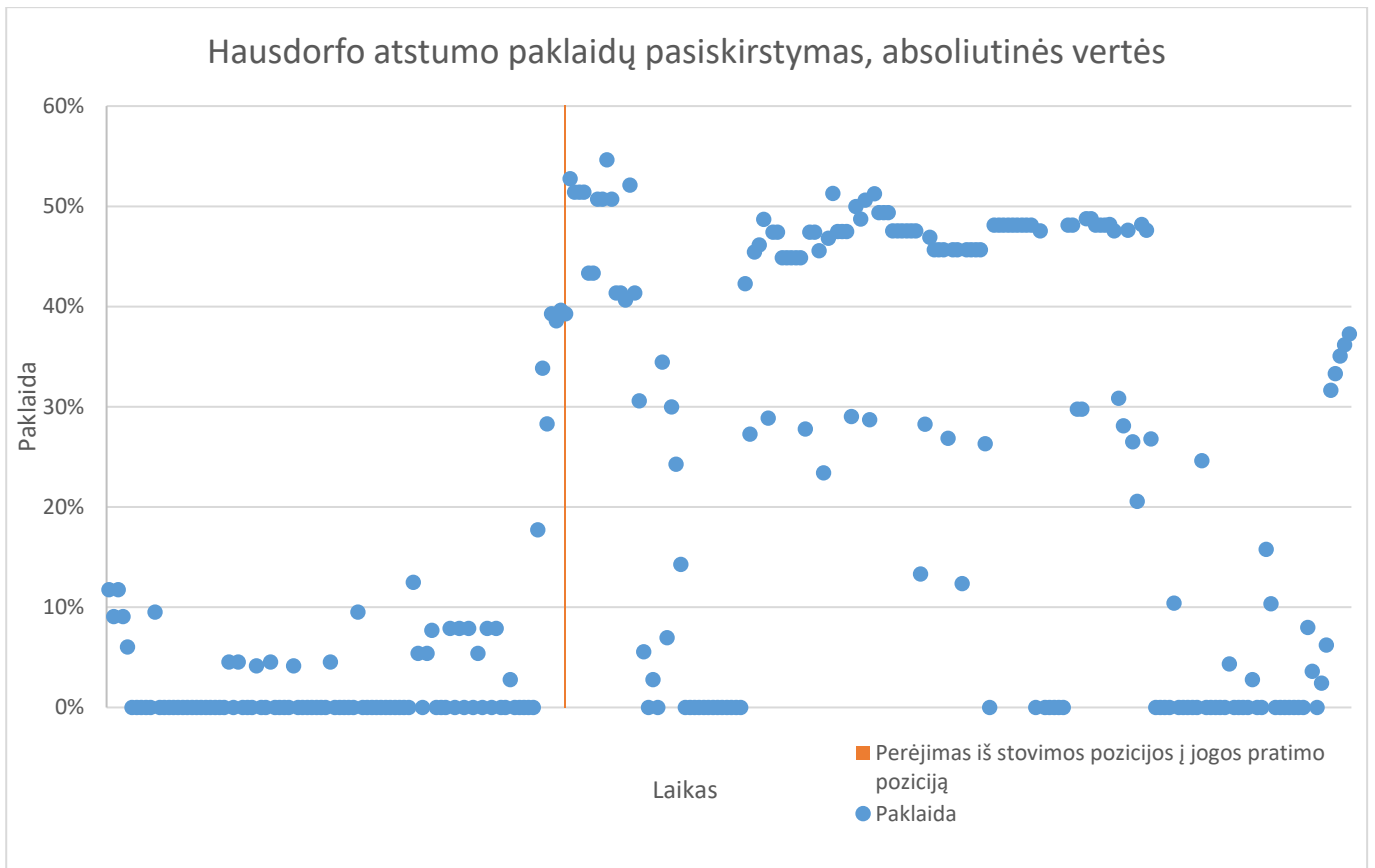
- Vidutinis nuokrypis – 3,4%;
- Didžiausia paklaida – 18%;
- Vidurkis – 2,0%;
- Mediana – 0%.
- Tiksliai įvertintų kadrų dalis – 91%.

Gautų paklaidų vidurkis – tik 2%, t. y., gerokai mažiau, nei 5% ribą, o mediana lygi 0, nes net 91% visų kadrų įvertinti tiksliai. Vidutinis nuokrypis taip pat nedidelis, todėl gaunamos paklaidos svyruoja nedaug. Tokie rezultatai rodo, kad Prokrusto analizės euristikos taikymas gerai tinka, kai jis vykdomas pratimo pradžioje, kai abu siluetai yra standartinėse stovimose padėtyse.

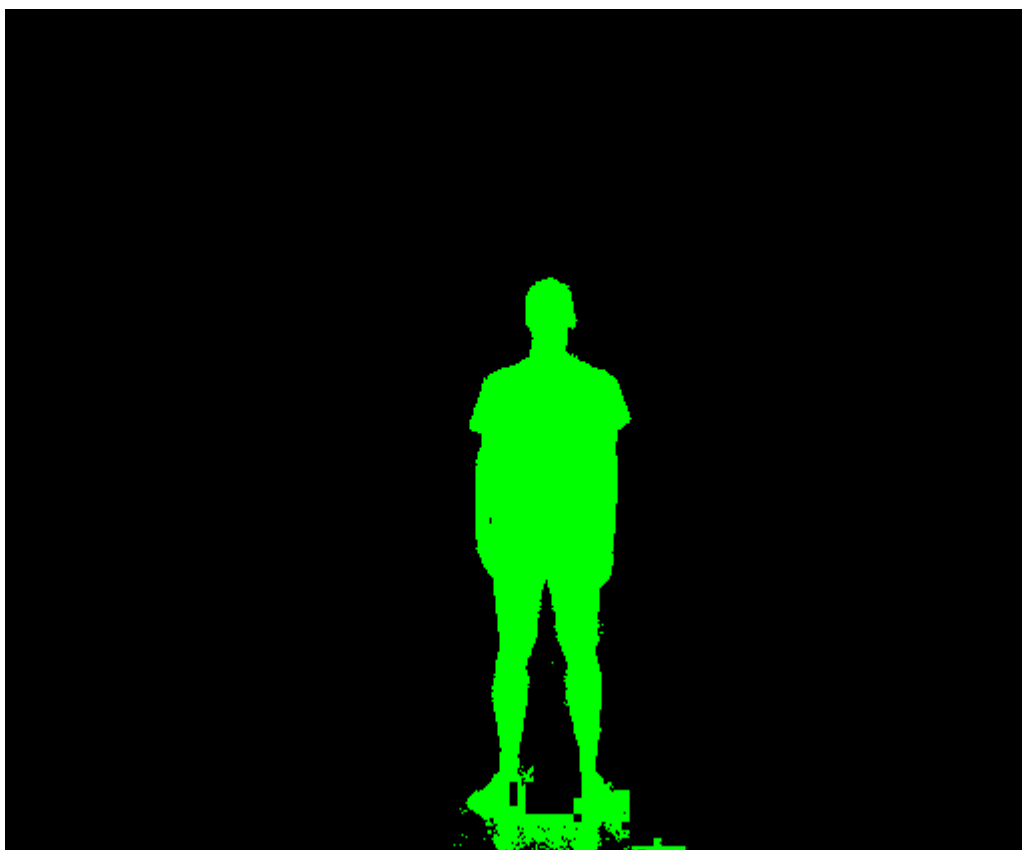
5.1 ir 5.2 pav. parodytos gautos paklaidos viso pratimo metu, o vertikali linija rodo, iki kurio laiko momento abu siluetai buvo stovimose padėtyse. 5.3 ir 5.4 pav. pavaizduoti žmogaus ir šabloninis siluetai standartinėse stovimose padėtyse.



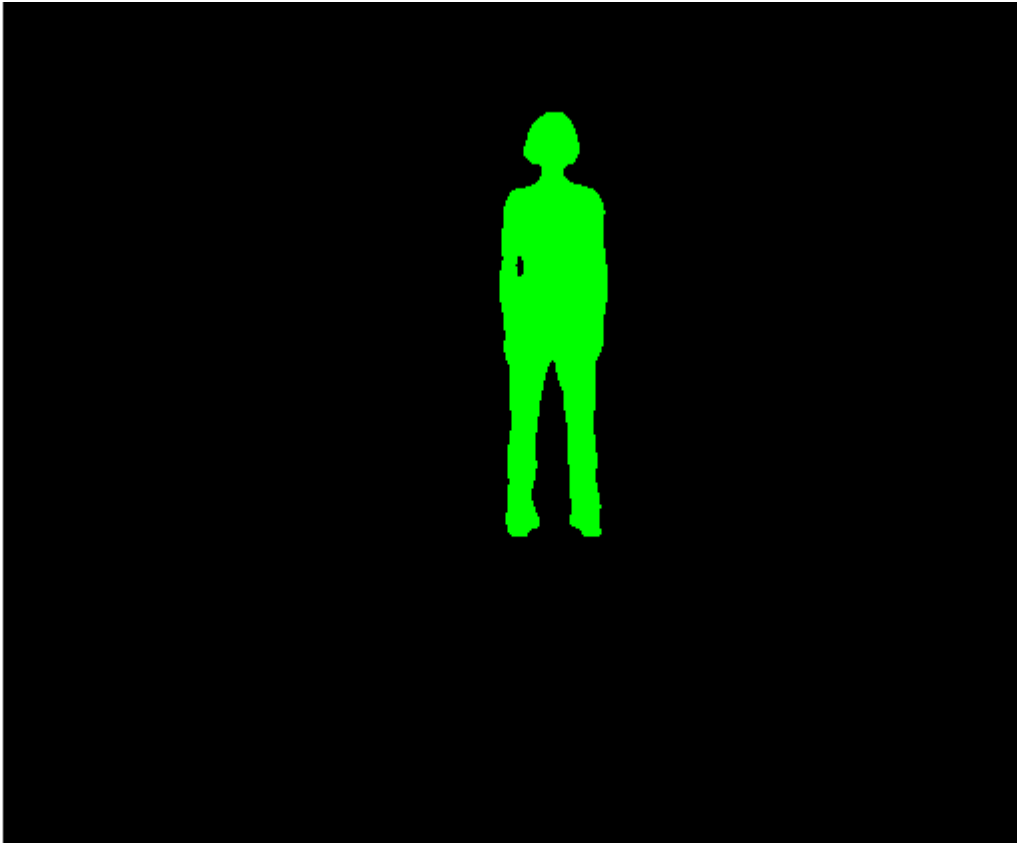
5.1 pav. Hausdorfo atstumo paklaidų pasiskirstymas



5.2 pav. Hausdorfo atstumo paklaidų pasiskirstymas, absoliutinės vertės

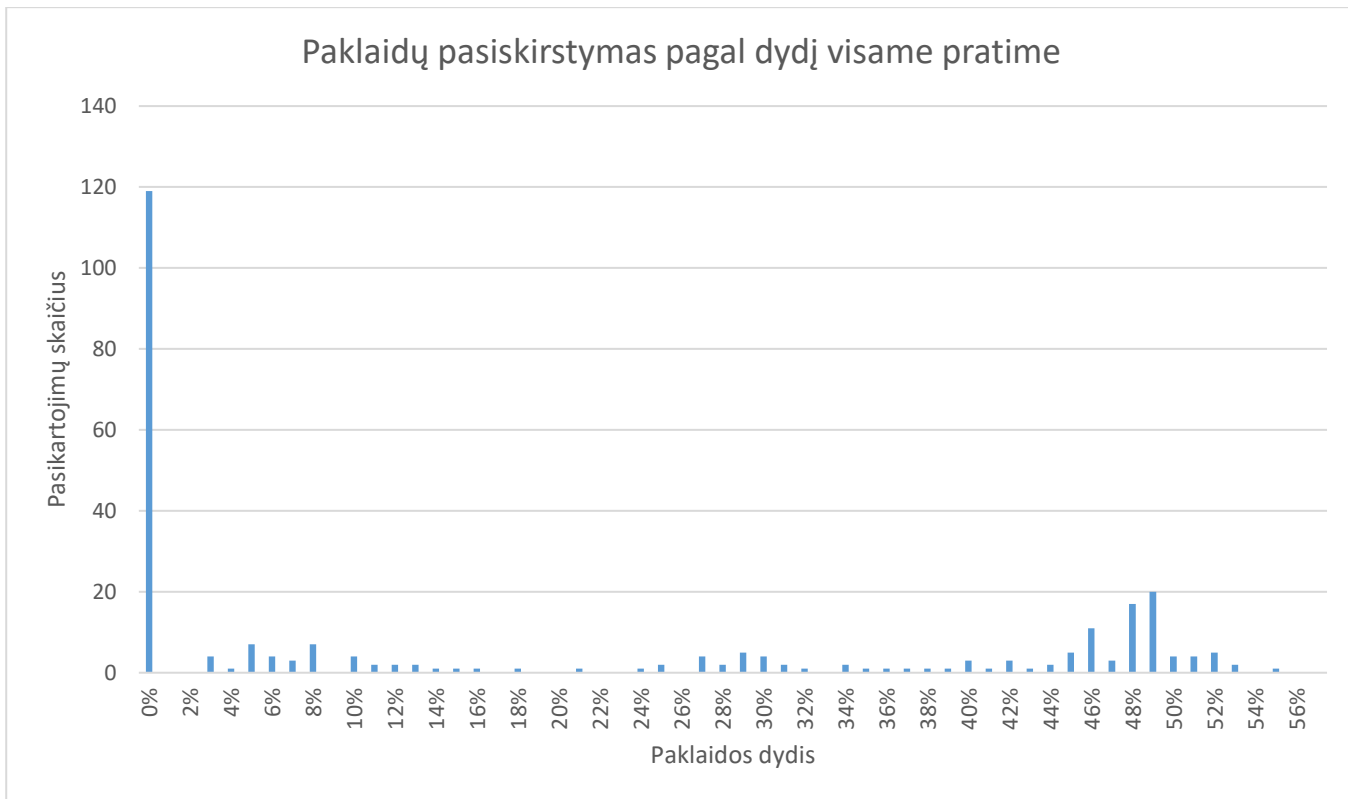


5.3 pav. Žmogaus siluetas standartinėje stovimoje padėtyje



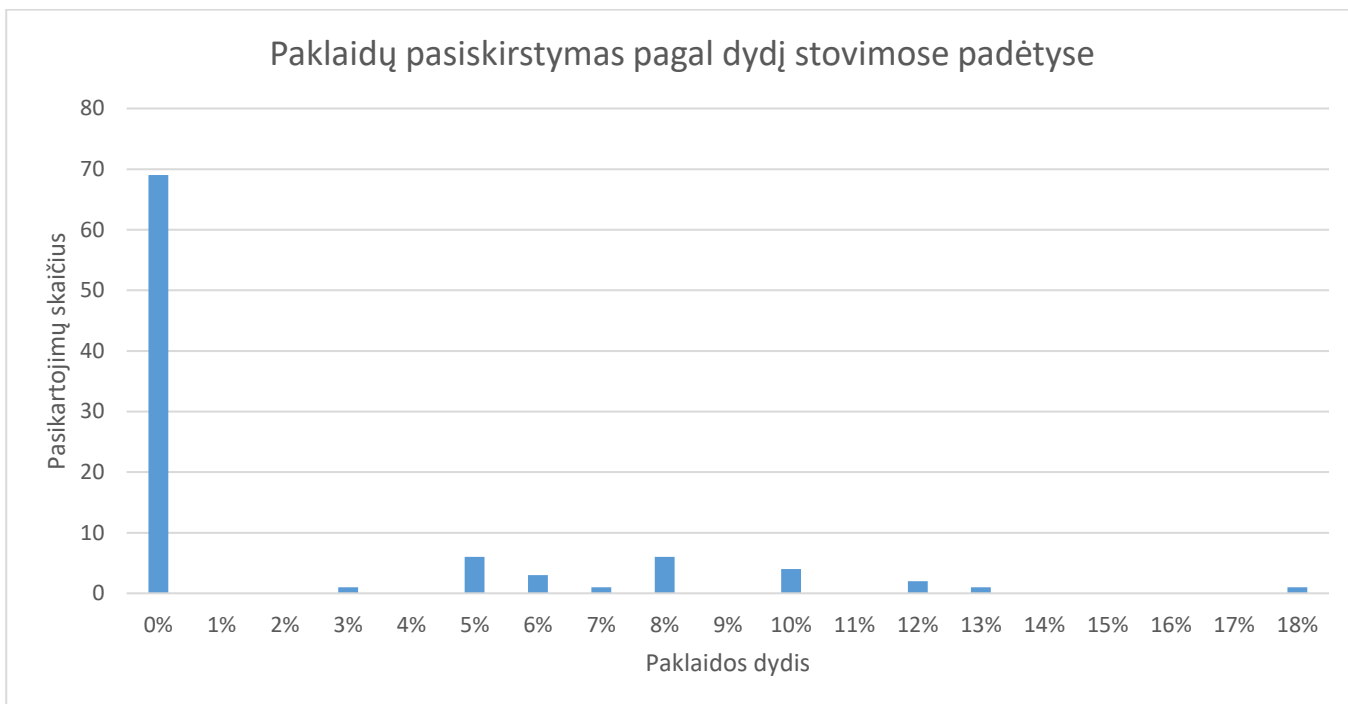
5.4 pav. Šabloninis siluetas standartinėje stovimoje padėtyje

5.5 pav. parodytas paklaidų pasiskirstymas pagal jų dydį visame pratime. Aukščiausi spygliai grafike matomi ties 8%, 28% ir 50% paklaidomis. Aiškios tendencijos iš grafiko nematyti, tačiau paklaidos gaunamos labai įvairaus dydžio.



5.5 pav. Hausdorfo atstumo skaičiavimo paklaidų pasiskirstymas pagal dydį visame pratime

5.6 pav. parodytas paklaidų pasiskirstymas pagal jų dydį visame stovimose padėtyse. Gautas paklaidos daugiausia pasiskirsčiusios intervale nuo 5% iki 10%, kitų paklaidų yra labai nedaug.



5.6 pav. Hausdorfo atstumo skaičiavimo paklaidų pasiskirstymas pagal dydį stovimose padėtyse

5.3.2. Euristinės Prokrusto transformacijos koeficientų nuokrypio įvertinimo rezultatai

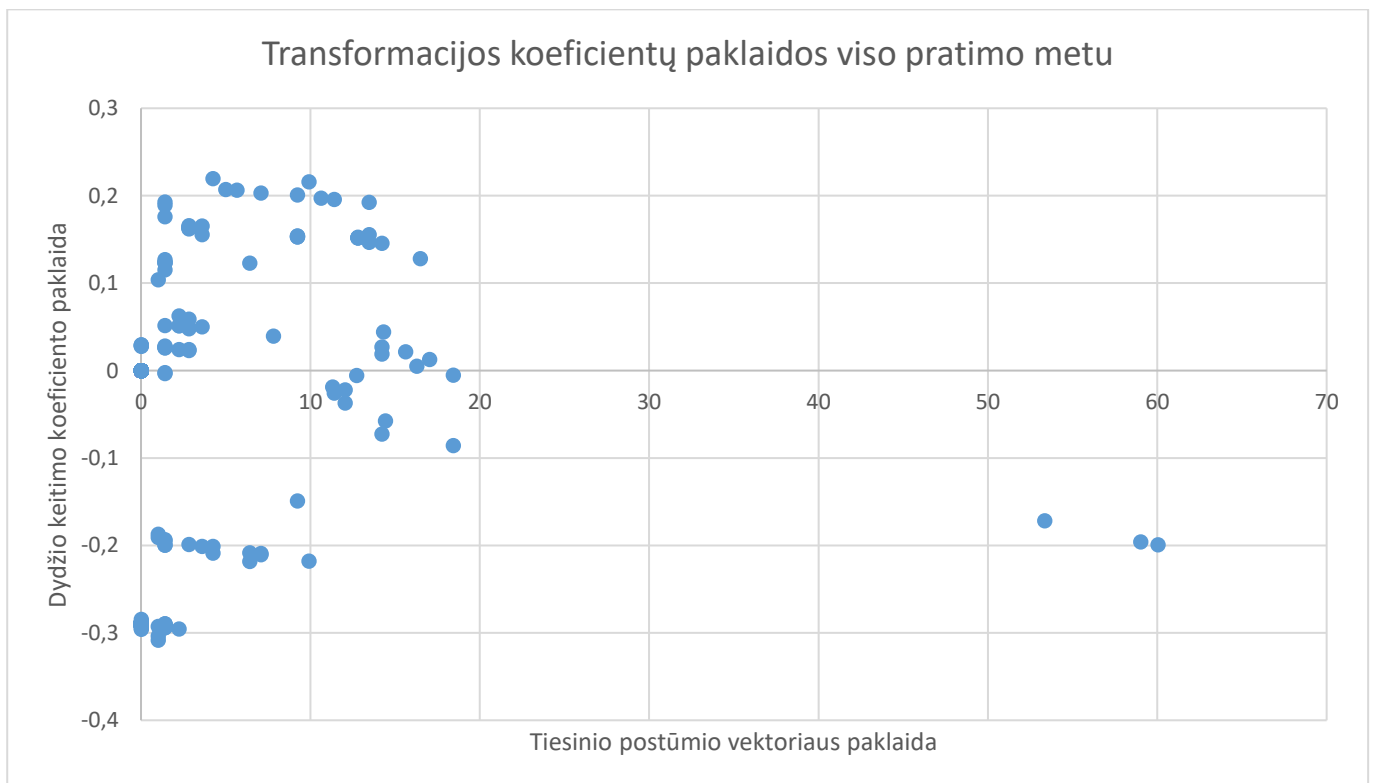
Ištyrus transformacijos koeficientų nuokrypius nustatyta, kad dydžio keitimo transformacijos paklaida svyruoja iki 30% nuo tikslios reikšmės (5.7 pav.) Diagramoje dydžio keitimo transformacijai įvertinti naudojamas paklaidos koeficientas:

$$e_s = 1 - \frac{S_a}{S_p} \quad (40),$$

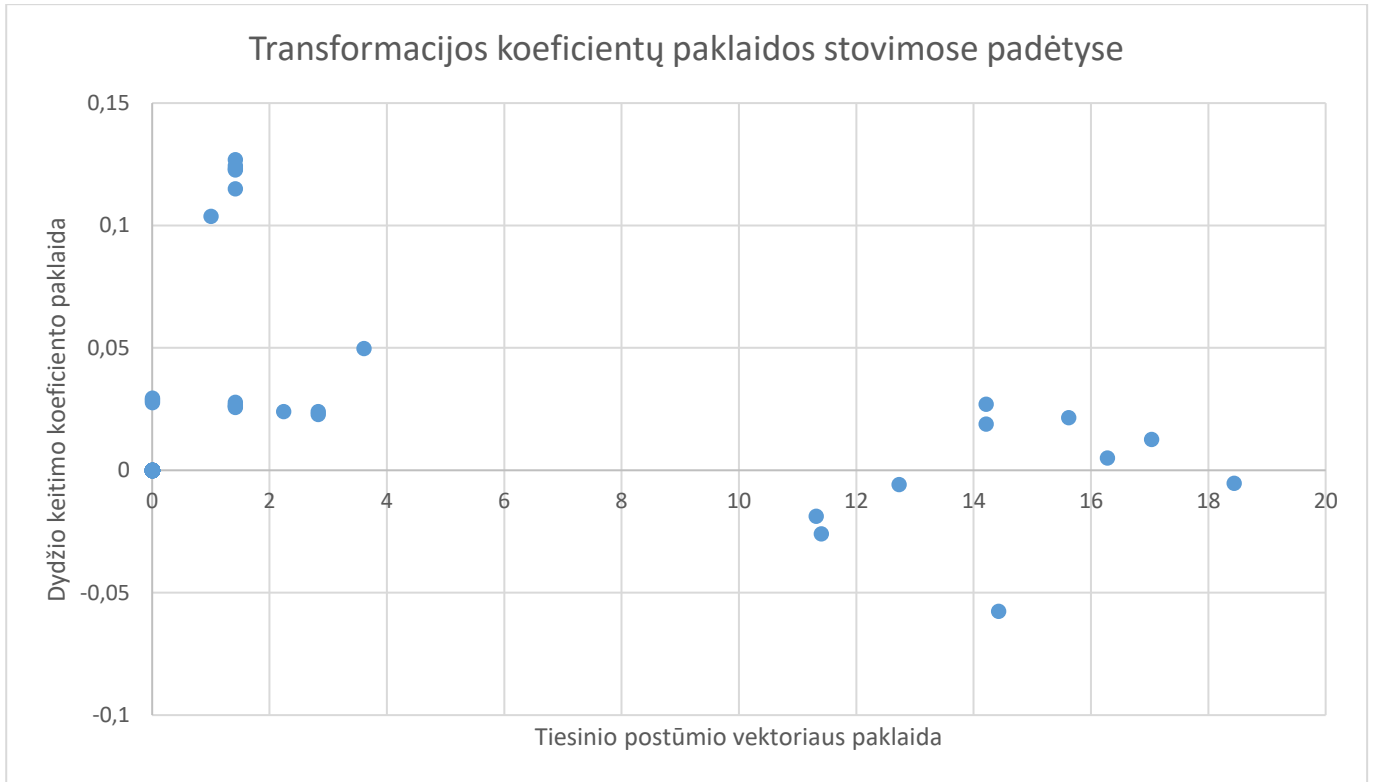
čia e_s – paklaidos koeficientas, S_a – gauta parametro reikšmė, S_p – patikslinta parametro reikšmė. Jei paklaidos koeficientas lygus 0, gauta tiksli reikšmė, jei neigiamas – per maža, jei teigiamas – per didelė.

Kitoje ašyje atidėta tiesinio postūmio vektoriaus paklaida. Ji išreikšta absoliutine skirtumu tarp gauto ir patikslinto transformacijos vektorių ilgio reikšme taškais. Paklaidos išsidėsčiusios daugiausia tarp 1 ir 20 taškų. Eksperimento metu žmogaus silueto aukštis buvo 200-300 taškų. Koreliacijos koeficientas tarp šių paklaidų lygus 0,12, todėl galima teigti, jog sąryšio tarp šių paklaidų dydžių nėra, jos yra beveik nepriklausomos viena nuo kitos.

5.8 pav. parodyta ta pati paklaidų statistika, bet vertinamos tik stovimos pozicijos. Šiuo atveju mastelio keitimo paklaida žymiai mažesnė ir dažniausiai neviršija 5% ribos. Tiesinio postūmio transformacijos paklaidos mažai skiriasi nuo viso pratimo metu daromų paklaidų, tik nėra didžiausių ekstremalių reikšmių.



5.7 pav. Transformacijos koeficientų paklaidos viso pratimo metu



5.8 pav. Transformacijos koeficientų paklaidos stovimose padėtyse

5.3.3. Siūlomo Hausdorfo algoritmo veikimo greičio priklausomybės nuo tiriamo kadro dydžio tyrimo rezultatai

Hausdorfo atstumas „Kinect 2“ naudojamam kadro dydžiui (512x424 taškų) su naudota aparatūrine įranga apskaičiuojamas per maždaug 8 milisekundes (5.9 pav.). 30 ms riba pasiekama ties 1,7 „Kinect 2“ kadro dydžio (870x720 taškų). Kadangi kadro dydžiui padidėjus nuo 1 iki 2 „Kinect 2“ kadro dydžių skaičiavimo laikas išaugo 4,7 karto, o nuo 2 iki 4 „Kinect 2“ kadro dydžio – 4,3 karto, nuo 2,5 iki 5 – 4,5 karto, tai algoritmo veikimo laiko priklausomybė nuo kadro dydžio yra artima tiesinei, kai analizuojami dideli kadrai.

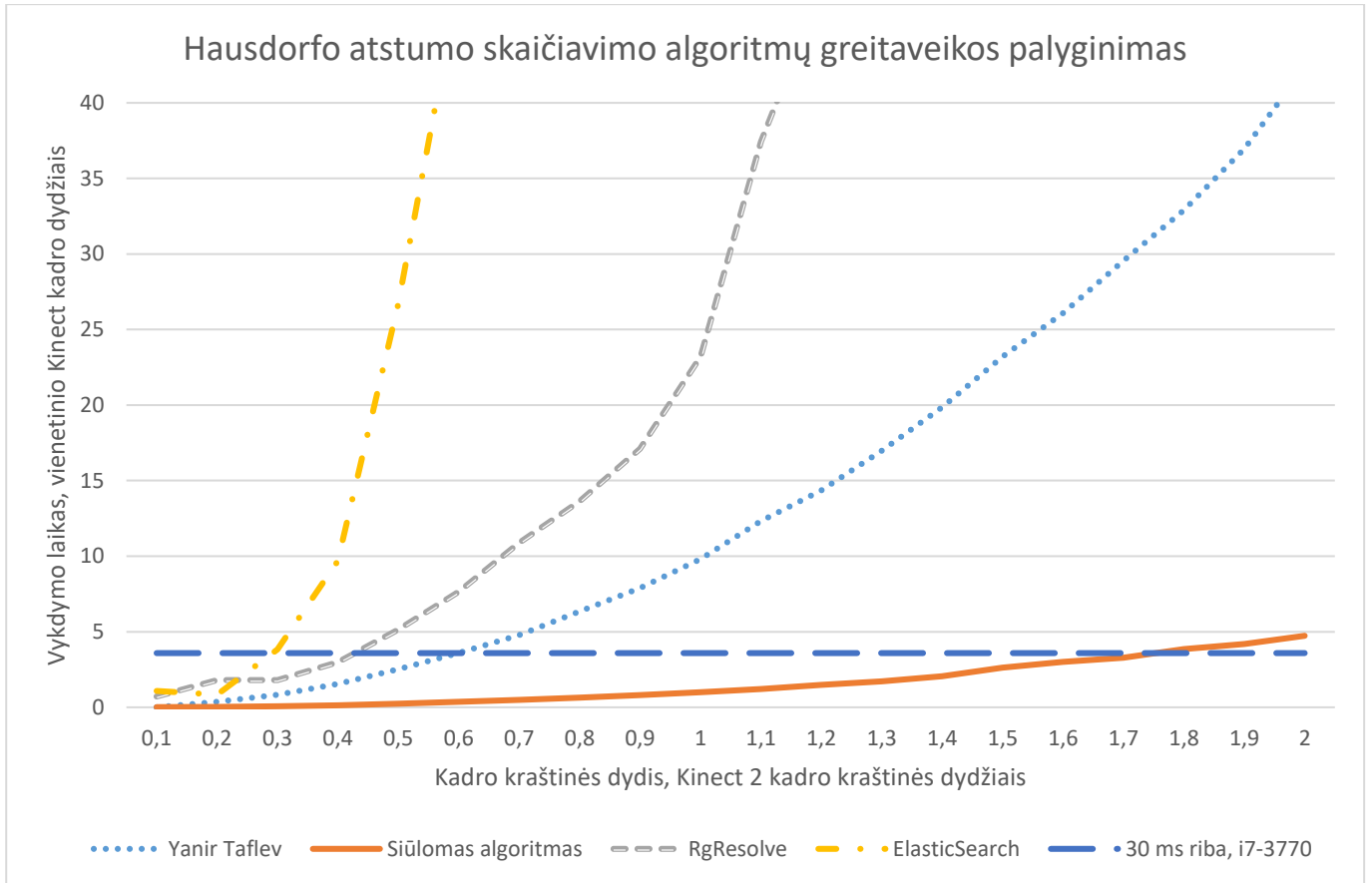
Grafike abi kreivės praktiškai sutampa, tad euristicinių Prokrusto transformacijos koeficientų skaičiavimas bendram apdorojimo greičiui įtakos praktiškai neturi. Grafike pažymėta riba, ties kuria pasiekama 30 ms apdorojimo laiko riba su aprašyta aparatūrine įranga.



5.9 pav. Hausdorfo atstumo skaičiavimo greitaveika

5.3.4. Hausdorfo atstumo skaičiavimo algoritmų greitaveikos palyginimo rezultatai

Šioje dalyje siūlomas algoritmas lyginamas su kitais Hausdorfo atstumo skaičiavimo algoritmais. 5.10 pav. parodytas toks pat grafikas, kaip ir ankstesniame skyriuje, tik palyginimui pridėtas kitų algoritmų veikimo laikas su tokiais pat kadrais. Iš grafiko matome, jog siūlomas algoritmas veikia žymiai greičiau, nei kiti nagrinėjami algoritmai. Siūlomas algoritmas punktyrinę 30 ms liniją kerta ties 1,7 „Kinect 2“ kadro dydžio (870x720 taškų), Yaniro Taflevo algoritmas – ties 0,6 „Kinect 2“ kadro (307x254 taškų). 320x240 taškų yra „Kinect 1“ jutiklio kadro dydis, todėl šis algoritmas galėtų būti taikomas darbui su senesniu „Kinect“ įrenginiu. „RgResolve“ Hausdorfo algoritmo realizacija šią ribą pasiekia ties 0,4 „Kinect 2“ kadro dydžio (205x170 taškų), „Elastic Search“ – ties 0,3 „Kinect 2“ kadro dydžio (154x127 taškų). Pastarųjų dviejų algoritmų veikimo laikas auga žymiai greičiau, nei dviejų pirmųjų, didinant kadro dydį. Siūlomas algoritmas yra žymiai greitesnis net ir už Yaniro Taflevo algoritmą. Tyrime naudotiems duomenims šis skirtumas yra ~10 kartų.



5.10 pav. Hausdorfo atstumo skaičiavimo algoritmų greitaveikos palyginimas

5.4. Eksperimentinio tyrimo išvados

Atlikus eksperimentinį tyrimą padarytos tokios išvados:

1. Siūlomo algoritmo daroma paklaida Hausdorfo atstumo skaičiavimui vidutiniškai neviršija 3%, jei ji taikoma tik stovimose standartinėse padėtyse;
2. Euristicinis metodas Prokrusto analizės transformacijos koeficientų radimui nėra tikslus, nes tiesinės transformacijos vektoriaus paklaida gali siekti 7-10% , tačiau įtaka surastam Hausdorfo atstumui yra mažesnė;
3. Sukurtas algoritmas tyrimui naudotiems „Kinect 2“ jutiklio duomenims Hausdorfo atstumo metriką gali apskaičiuoti greičiau, nei duomenys yra pateikiami (8 ms apdorojimo laikas, 33 ms intervalai tarp duomenų pateikimo);
4. Sukurtas algoritmas su tyrimui naudotais „Kinect 2“ duomenimis veikia bent 10 kartų greičiau, nei kiti egzistuojantys sprendimai.
5. Geriausia naudoti euristinę Prokrusto analizę judesio pradžioje, kai žmogus ir šablonas yra standartinėse stovimose padėtyse, o gautus transformacijos koeficientus taikyti viso pratimo metu.

6. IŠVADOS

1. Skyriuje 2.5 „Judėsių ir pozicijų palyginimo algoritmų analizė“ atlikta žinomų judėsių sekimo algoritmų analizė ir nustatyta, kad nei vienas iš jų netenkina visų keliamų kokybinių kriterijų, kuriuos turėtų atitikti algoritmas, naudojamas jogos žaidime, tačiau geriausi yra Hausdorfo atstumo ir nuožulnos algoritmai.
2. Sukurtas algoritmas, kuris su tyrime naudotais duomenimis Hausdorfo atstumo metriką gali apskaičiuoti greičiau, nei duomenys yra pateikiami „Kinect“ (8 ms apdorojimo laikas, 33 ms intervalai tarp duomenų pateikimo su tyrimui naudota aparatūrine įranga). Toks veikimo greitis tinka taikyti algoritmą judėsių sekų palyginimui realiu laiku;

Skyriuje 0 „

3. Eksperimentinė dalis“ nustatyta, kad:

- Siūlomo algoritmo daroma paklaida vidutiniškai neviršija 5%;
- Euristinis metodas Prokrusto analizės transformacijos koeficientų radimui nėra tikslus, nes tiesinės transformacijos vektoriaus paklaida gali siekti 7-10% , tačiau įtaka surastam Hausdorfo atstumui yra mažesnė;
- Sukurtas algoritmas su tyrime naudotais duomenimis Hausdorfo atstumo metriką gali apskaičiuoti greičiau, nei duomenys yra pateikiami „Kinect“ (8 ms apdorojimo laikas, 33 ms intervalai tarp duomenų pateikimo);
- Sukurtas algoritmas veikia bent 10 kartų greičiau, nei kiti egzistuojantys sprendimai su tyrimui naudotais duomenimis.

4. Sukurtas algoritmas integruotas jogos žaidime ir sukurta pagalbinė programinė įranga judesių šablonų failams ruošti. Programinė įranga aprašyta skyriuje 3 „Projektinė dalis“

Aprobavimas:

- Virtualus jogos treneris buvo pristatytas šiuose renginiuose:
 - „GameOn 2015“;
 - „Technorama 2015“ (III vietos prizas).
- Sukurtas algoritmas buvo pristatytas renginyje „Technorama 2016“ („Intermedix“ geriausio IT srities darbo prizas);
- Sukurtas algoritmas buvo pristatytas konferencijoje „IVUS 2017“;
- Susiję darbai publikuoti konferencijose „IVUS 2014“ ir „IVUS 2015“.

7. LITERATŪROS SĄRAŠAS

1. Use of Complementary Health Approaches in the U.S. [Tinkle] National Center for Complementary and Integrative Health, 2012 m. [Cituota: 2015 m. 11 29 d.] <https://nccih.nih.gov/research/statistics/NHIS/2012/mind-body/yoga>.

2. 9.5% of U.S. adults (21 million) used. [Tinkle] Health National Center for Complementary and Integrative, 2015 m. 02 12 d. [Cituota: 2015 m. 11 04 d.] <https://nccih.nih.gov/research/statistics/NHIS/2012/mind-body/yoga>.

3. *A Survey of Applications and Human Motion Recognition with Microsoft Kinect*. Lun, Roanna ir Zhao, Wenbing. 2015 m., International Journal of Pattern Recognition and Artificial Intelligence.

4. Ye, Mao, et al. A Survey in Human Motion Analysis from Depth Data. *Time-of-Flight and Depth Imaging. Sensors, Algorithms and Applications*. s.l. : Springer, 2013, p. 148-186.

5. Network, Microsoft Developer. Kinect for Windows Sensor Components and Specifications. [Tinkle] Microsoft. [Cituota: 2015 m. 11 04 d.] <https://msdn.microsoft.com/en-us/library/jj131033.aspx?f=255&MSPPErr=-2147217396>.

6. Fernandez, Dan. Installing and Using the Kinect Sensor (Beta 2 SDK). [Tinkle] 2012 m. 06 16 d. [Cituota: 2015 m. 11 12 d.] <https://channel9.msdn.com/Series/KinectSDKQuickstarts/Understanding-Kinect-Hardware>.

7. FrameSourceTypes Enumeration. [Tinkle] Microsoft. [Cituota: 2015 m. 11 04 d.] <https://msdn.microsoft.com/en-us/library/windowspreview.kinect.framesourcetypes.aspx>.

8. V., Pterneas. Kinect for Windows version 2: Body tracking. [Tinkle] 2014 m. 04 11 d. [Cituota: 2015 m. 11 04 d.] <http://www.codeproject.com/Articles/743862/Kinect-for-Windows-version-Body-tracking>.

9. JointType Enumeration. [Tinkle] Microsoft. [Cituota: 2015 m. 11 12 d.] <https://msdn.microsoft.com/en-us/library/windowspreview.kinect.jointtype.aspx>.

10. Petkus, Tautvydas ir Ryselis, Karolis. Nestandartinių žmogaus kūno pozicijų atpažinimo tikslumo naudojant "Kinect 2.0" jutiklius tyrimas. *XX tarpuniversitetinės magistrantų ir doktorantų konferencijos „Informacinė visuomenė ir universitetinės studijos“ (IVUS 2015) pranešimų medžiaga "Informacinės technologijos" (ISSN 2029-249X)*. 2015 m.

11. J., Voissele. Kinect Augmented Reality Organ Visualization. [Tinkle] 2013 m. [Cituota: 2015 m. 11 04 d.] http://doc.rero.ch/record/235894/files/Voisselle_Joel_2014.pdf.

12. Khoshelham K., Elberink S. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. [Tinkle] 2012 m. 02 01 d. [Cituota: 2015 m. 11 04 d.] <http://www.mdpi.com/1424-8220/12/2/1437/htm>.
13. Smeenk, Roland. Kinect V1 and Kinect V2 fields of view compared. [Tinkle] 2014 m. 03 11 d. [Cituota: 2017 m. 05 01 d.] <http://smeenk.com/kinect-field-of-view-comparison/>.
14. *Accuracy Analysis of Kinect Depth Data*. Khoshelham, K. Calgary : s.n., 2011. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-5/W1.
15. V., Pterneas. Kinect for Windows version 2: overview. [Tinkle] 2014 m. 02 08 d. [Cituota: 2015 m. 11 04 d.] <http://pterneas.com/2014/02/08/kinect-for-windows-version-2-overview/>.
16. *Žmogaus sąnarių padėties nustatymo naudojant kelis Kinect įrenginius algoritmo sudarymas ir tyrimas*. Petkus, Tautvydas ir Ryselis, Karolis. 2014. IXI tarpuniversitetinės magistrantų ir doktorantų konferencijos „Informacinė visuomenė ir universitetinės studijos“ (IVUS 2014) pranešimų medžiaga "Informacinės technologijos".
17. BodyIndexFrame.CopyFrameDataToArray Method. [Tinkle] Microsoft. [Cituota: 2015 m. 11 07 d.] <https://msdn.microsoft.com/en-us/library/windowspreview.kinect.bodyindexframe.copyframedatatoarray.aspx>.
18. *Measuring the resemblance of polygonal curves*. Alt, Helmut ir Godau, Michael. 1992 m., SCG '92 Proceedings of the eighth annual symposium on Computational geometry, p. 102-109.
19. Huttenlocher, Daniel, Klanderman, Gregory ir Rucklidge, William. Comparing Images Using Hausdorff Distance. *IEEE Transactions On Pattern Analysis And Machine Intelligence*. 1993 m. 09, p. 850-863.
20. Dryden, Ian ir Mardia, Kanti. *Statistical Shape Analysis*. Chichester : John Wiley, 1998.
21. Veeraraghavan, Ashok, Roy-Chowdhury, Amit K ir Chellappa, Rama. Matching Shape Sequences in Video with Applications in Human Movement Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2005 m. 10 31 d., p. 1896-1909.
22. Ross, Amy. *Procrustes Analysis*. Columbia : Department of Computer Science and Engineering, University of South Carolina, 20014.
23. *Human Body Pose Recognition Using Spatio-Temporal Templates*. Dimitrijevic, M, Lepetit, V ir Fua, P. 2005. ICCV workshop on Modeling People and Human Interaction.

24. Huber, Peter ir Ronchetti, Elvezio. *Robust Statistics*. New Jersey : Josh Wiley & Sons Inc., 2009.

25. Taflev, Yanir. Using the Hausdorff distance algorithm to point out differences between two drawings. [Tinkle] 2009 m. 09 27 d. [Cituota: 2017 m. 03 10 d.] <https://www.codeproject.com/articles/42669/using-the-hausdorff-distance-algorithm-to-point-ou>.

26. Solutions, Vivid. `elasticsearch-client/DiscreteHausdorffDistance`. [Tinkle] [Cituota: 2017 m. 03 10 d.] <https://github.com/jprante/elasticsearch-client/blob/master/elasticsearch-client-jts-jdk5/src/main/java/com/vividsolutions/jts/algorithm/distance/DiscreteHausdorffDistance.java>.

27. jprante. `elasticsearch-client/DiscreteHausdorffDistance.java`. [Tinkle] 2012 m. 11 9 d. [Cituota: 2017 m. 05 11 d.] <https://github.com/jprante/elasticsearch-client/commits/master/elasticsearch-client-jts-jdk5/src/main/java/com/vividsolutions/jts/algorithm/distance/DiscreteHausdorffDistance.java>.

28. Hausdorff Distance. [Tinkle] Princeton University. [Cituota: 2017 m. 03 12 d.] <http://www.princeton.edu/~rkatzwer/rgsolve/doc/edu/princeton/polygon/HausdorffDistance.html>.

29. Buseti, Franco. *Simulated annealing overview*.

30. Kinect API Overview. [Tinkle] [Cituota: 2017 m. 03 10 d.] <https://msdn.microsoft.com/en-us/library/dn782033.aspx>.

31. *Nestandartinių žmogaus kūno pozicijų atpažinimo tikslumo naudojant „Kinect 2.0“ jutiklius tyrimas*. Ryselis, Karolis ir Petkus, Tautvydas. Kaunas : s.n., 2015.

32. *Comparing Images using the Hausdorff Distance*. Huttenlocher, Daniel P., Klanderman, Gregory A. ir Rucklidge, William J. 15, 1993 m., IEEE Transactions on Pattern Analysis and Machine Intelligence, p. 850-863.

8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

Kinect – „Microsoft“ sukurtas įrenginys, skirtas žmogaus padėties nustatymui.

Joga – veikla, kurią apima protinė, fizinė ir dvasinė treniruotė.

Siluetas – vienspalvis žmogaus atvaizdas, pateikiamas kaip uždara užpildyta figūra, kurios kraštai sutampa su išorine vaizduojamo subjekto kraštine.

Gylio žemėlapis – trimatėje kompiuterinėje grafikoje naudojamas objekto taškų padėčių atvaizdis į jų atstumus nuo stebėjimo taško.

Skeletas – iš anksto numatytų taškų žmogaus taškų, per kuriuos galimi lenkimai, koordinacių rinkinys.

Figūrų seka – žmogaus pozicijos atvaizdų rinkinys, išdėstytas chronologine tvarka ir reprezentuojantis vieną veiksmą.

Žmogaus judesių sekimas – įrašo ir esamos žmogaus veiklos sulginimas indikuojant skirtumus tarp jų ir nustatant, ar veikla atitinka šabloną.

9. PRIEDAI

9.1. Sistemos reikalavimai

9.1.1. Apribojimai sistemai

Reikalavimas	1	Reikalavimo tipas	Reikalavimas sprendimui	Panaudos atvejis	-
Aprašymas	Judesių sekimo programinė įranga turi būti suderinama su „Unity“ žaidimų varikliu.				
Pagrindimas	Žaidimas „Yogamin“ kuriamas naudojant „Unity“ žaidimų variklį.				
Atitikimo kriterijus	Žaidime „Yogamin“ pavyksta paleisti sukurtą judesių sekimo komponentą.				
Kliento pasitenkinimas	1	Kliento nepasitenkinimas		5	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	2	Reikalavimo tipas	Reikalavimas sprendimui	Panaudos atvejis	-
Aprašymas	Judesių sekimo programinė įranga turi veikti „Windows 8“, „Windows 8.1“ ir „Windows 10“ sistemose.				
Pagrindimas	Žaidimas „Yogamin“ bus palaikomas šiose „Windows“ sistemos versijose.				
Atitikimo kriterijus	Sukurtą programinę įrangą pavyksta paleisti su nurodytomis operacinės sistemos versijomis.				
Kliento pasitenkinimas	1	Kliento nepasitenkinimas		5	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	3	Reikalavimo tipas	Reikalavimas sprendimui	Panaudos atvejis	-
Aprašymas	Judesių sekimo komponentas turi palaikyti „Kinect“ ir „Kinect 2“ įrenginius.				
Pagrindimas	„Kinect“ šiuo metu yra plačiausiai paplitęs įrenginys, tinkamas judesių sekimui. Nors naujausia jo versija yra „Kinect 2“, daug vartotojų iki dabar turi senąjį „Kinect“ variantą. Norint, kad jie išliktų potencialiais sistemos vartotojais, reikia palaikyti jų turimą įrangą.				
Atitikimo kriterijus	Judesių sekimo komponentas analogiškai veikia su „Kinect“ ir „Kinect 2“.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		4	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

9.1.2. Funkciniai reikalavimai

Reikalavimas	4	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA1
Aprašymas	Kalibracija turi pateikti vieną iš šių žmogaus lankstumo tipų: lankstus, vidutinis, nelankstus.				
Pagrindimas	Sistema turi būti pritaikyta įvairaus lankstumo žmonėms.				
Atitikimo kriterijus	Sistema išskiria visus tris žmogaus lankstumo tipus.				
Kliento pasitenkinimas	4	Kliento nepasitenkinimas		3	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	5	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA1
Aprašymas	Kalibracijai įgyvendinti naudojamos trys jėgos pozos, apimančios persilenkimus per skirtingas vietas.				
Pagrindimas	Žmogaus lankstumas per skirtingus taškus gali būti skirtingas, reikia įvertinti kelis taškus.				
Atitikimo kriterijus	Kalibruojant naudojamos trys pozos, rezultatą sudaro visų trijų pozų įverčiai				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		2	
Prioritetas	Vidutinis			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	6	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA2
Aprašymas	Palyginimo progreso informacija turi leisti žaidimui žinoti, kuri dalis pratimo jau yra atlikta.				
Pagrindimas	Žaidimas turės informuoti vartotoją apie tai, kiek pamokos jau įveikta.				
Atitikimo kriterijus	Sistema kiekvienam kadru pateikia esamą progresą.				
Kliento pasitenkinimas	3	Kliento nepasitenkinimas		2	
Prioritetas	Vidutinis			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	7	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA2
Aprašymas	Sistema turi žaidimui pateikti, kokie yra skirtumai tarp žmogaus silueto ir šablono.				
Pagrindimas	Žaidimas turės informuoti vartotoją apie tai, ką jis daro neteisingai.				
Atitikimo kriterijus	Sistema kiekvienam kadru pateikia skirtumus tarp žmogaus silueto ir įrašo.				
Kliento pasitenkinimas	3	Kliento nepasitenkinimas		3	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	8	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA2
Aprašymas	Sistema turi semantiškai nustatyti, kuri žmogaus kūno dalis yra pozicionuota neteisingai, jei tokių yra.				
Pagrindimas	Žaidime bus naudojama garsinė informacija klaidoms pranešti, todėl žaidimas turės pasakyti, kad tam tikrai kūno daliai reikia korekcijų.				
Atitikimo kriterijus	Sistema nustato žmogaus kūno dalį, kur daroma klaida.				
Kliento pasitenkinimas	3	Kliento nepasitenkinimas		3	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	9	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA2
Aprašymas	Galutinis vartotojo rezultatas turi būti pateikiamas režiuose nuo 0 iki 1, kur 0 – prasčiausias galimas įvertis, 1 – geriausias.				
Pagrindimas	Rezultatui turi būti apibrėžta vertinimo skalė.				
Atitikimo kriterijus	Sistemos pateikiami rezultatai niekada nepatenka už nustatytų rėžių.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		2	
Prioritetas	Žemas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	10	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA2
Aprašymas	Tarpinis vartotojo rezultatas turi būti pateikiamas režiuose nuo 0 iki 1, kur 0 – prasčiausias galimas įvertis, 1 – geriausias.				
Pagrindimas	Rezultatui turi būti apibrėžta vertinimo skalė.				
Atitikimo kriterijus	Sistemos pateikiami rezultatai niekada nepatenka už nustatytų režių.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		2	
Prioritetas	Žemas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	11	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA4
Aprašymas	Sistema žaidimui delno koordinatas turi pateikti žaidimo lango koordinatinių sistemoje.				
Pagrindimas	Sistema turi pateikti koordinatas taip, kad nereikėtų papildomų perskaiciavimų.				
Atitikimo kriterijus	Valdymui delnu nereikia transformuoti duomenų.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		1	
Prioritetas	Žemas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	12	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA10
Aprašymas	Žaidimo pradžioje paleidžiamas judesių sekimas.				
Pagrindimas	Žaidimo metu turi būti sekami vartotojo judesiai.				
Atitikimo kriterijus	Paleidus žaidimą veikia judesių sekimas.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		4	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	13	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA2
Aprašymas	Turi būti galimybė inicijuoti judesių sekimą pačiam žaidimui.				
Pagrindimas	Gali būti atjungtas ir vėl prijungtas judesių sekimo įrenginys.				
Atitikimo kriterijus	Yra būdas paleisti judesių palyginimą pačiam žaidimui ir juo pasinaudojus informacija apie judesių sekimą tampa prieinama.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		4	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	14	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA2
Aprašymas	Sistema turi pati įkelti judesių įrašą, pagal kurį bus vykdomas judesių palyginimas.				
Pagrindimas	Judesių palyginimo failai yra duomenys, kuriais manipuliuoja sistema, todėl jų valdymas paliekamas jai pačiai.				
Atitikimo kriterijus	Sistemai nurodžius judesių įrašo failą jis yra įkeliamas.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		3	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	15	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA2
Aprašymas	Turi būti galimybė nutraukti judesių sekimą.				
Pagrindimas	Toks veikimas reikalingas, kai žaidėjas nutraukia žaidimą arba kai pabaigia visą treniruotę.				
Atitikimo kriterijus	Yra būdas sustabdyti judesių palyginimą pačiam žaidimui ir juo pasinaudojus atlaisvinami <i>Kinect</i> naudojami resursai.				
Kliento pasitenkinimas	1	Kliento nepasitenkinimas		3	
Prioritetas	Vidutinis			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	16	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA2
Aprašymas	Sistema turi pateikti judesių sekimo tarpinius rezultatus žaidimui.				
Pagrindimas	Ši informacija reikalinga, kad parodyti vartotojui jo daromas klaidas ir informuoti apie progresą.				
Atitikimo kriterijus	Įjungus judesių sekimą tarpiniai rezultatai yra prieinami žaidimui.				
Kliento pasitenkinimas	3	Kliento nepasitenkinimas		3	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	17	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA2
Aprašymas	Sistema turi suskaičiuoti galutinį vartotojo rezultatą.				
Pagrindimas	Ši informacija reikalinga, kad parodyti vartotojui, kaip sėkmingai jam pavyko atlikti treniruotę.				
Atitikimo kriterijus	Išjungus judesių sekimą galutiniai rezultatai yra prieinami žaidimui.				
Kliento pasitenkinimas	3	Kliento nepasitenkinimas		3	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	18	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA4
Aprašymas	Turi būti galimybė inicijuoti žmogaus delno sekimą				
Pagrindimas	Žmogaus delno valdymą kontroliuoja žaidimas.				
Atitikimo kriterijus	Yra būdas paleisti delno sekimą pačiam žaidimui ir juo pasinaudojus delno koordinatės tampa prieinamos.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		2	
Prioritetas	Žemas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	19	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA4
Aprašymas	Turi būti galimybė sustabdyti žmogaus delno sekimą				
Pagrindimas	Žmogaus delno valdymą kontroliuoja žaidimas.				
Atitikimo kriterijus	Yra būdas sustabdyti delno sekimą pačiam žaidimui ir juo pasinaudojus atlaisvinami <i>Kinect</i> resursai.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		2	
Prioritetas	Žemas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	20	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA2, PA5
Aprašymas	Judesių redagavimo įrankis ir judesių sekimo komponentas turi naudoti bendrą duomenų formatą.				
Pagrindimas	Redagavimo įrankio paskirtis yra pateikti judesių sekimo komponentui duomenis.				
Atitikimo kriterijus	Konvertavus judesį ir įkėlus į žaidimą veikia judesių palyginimas pagal tą judesį.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		4	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	21	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA5
Aprašymas	Konvertuojant judesio įrašą vartotojui turi būti rodomas konvertavimo progresas.				
Pagrindimas	Vartotojas turi matyti konvertavimo progresą.				
Atitikimo kriterijus	Paleidus judesio įrašo konvertavimą rodomas konvertavimo progresas.				
Kliento pasitenkinimas	3	Kliento nepasitenkinimas		1	
Prioritetas	Vidutinis			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	22	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA6
Aprašymas	Judesys vizualizuojamas grafinėje sąsajoje tam skirtoje lango dalyje.				
Pagrindimas	Vartotojas turi vizualiai matyti, koks judesys atliekamas.				
Atitikimo kriterijus	Paleidus judesio įrašo atkūrimą grafinėje sąsajoje matoma jo vizualizacija.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		4	
Prioritetas	Vidutinis			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	23	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA6
Aprašymas	Atkuriant judesio įrašą turi būti galimybė prasukti įrašą laiko skalėje iki norimo momento.				
Pagrindimas	Tai reikalinga, kad nereikėtų kiekvieną kartą žiūrėti viso įrašo iš naujo.				
Atitikimo kriterijus	Grafinėje sąsajoje rodoma laiko skalė, kurioje galima pasirinkti laiko momentą, nuo kurio rodyti įrašą.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		4	
Prioritetas	Vidutinis			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	23	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA7
Aprašymas	Esami stabdos taškai turi būti matomi atskiroje laiko skalėje.				
Pagrindimas	Judesių redaktorius turi aiškiai matyti, kokie stabdos taškai yra naudojami.				
Atitikimo kriterijus	Įkėlus failą, turintį stabdos taškų, jie matomi laiko skalėje.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		4	
Prioritetas	Vidutinis			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	24	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA7
Aprašymas	Stabdos taškų valdymas vykdomas laiko skalėje.				
Pagrindimas	Stabdos taškų valdymas ir vizualizacija turi būti integruoti.				
Atitikimo kriterijus	Paspaudus stabdos taškų laiko skalėje pridamas / pašalinamas stabdos taškas.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		4	
Prioritetas	Vidutinis			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	25	Reikalavimo tipas	Funkcinis reikalavimas	Panaudos atvejis	PA7
Aprašymas	Failas turi būti išsaugomas programos aplanke.				
Pagrindimas	Įrankis turi būti kiek galima paprastas.				
Atitikimo kriterijus	Išsaugotas failas atsiranda programos aplanke.				
Kliento pasitenkinimas	1	Kliento nepasitenkinimas		2	
Prioritetas	Žemas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

9.1.3. Nefunkciniai reikalavimai

9.1.3.1. Reikalavimai vykdymo charakteristikoms

Reikalavimas	26	Reikalavimo tipas	Reikalavimas vykdymo savybėms	Panaudos atvejis	PA2
Aprašymas	Judesių kadrų palyginimas turi vykti bent 25 kadrų per sekundę greičiu (40 ms vienam kadru).				
Pagrindimas	Judesių palyginimas turi veikti realiu laiku.				
Atitikimo kriterijus	Kadrai vidutiniškai palyginami per 40 ms ar mažiau.				
Kliento pasitenkinimas	3	Kliento nepasitenkinimas		4	
Prioritetas	Aukštas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	27	Reikalavimo tipas	Reikalavimas vykdymo savybėms	Panaudos atvejis	PA4
Aprašymas	Delno pozicijos atnaujinimas turi vykti bent 25 kartus per sekundę.				
Pagrindimas	Delno pozicijos sekimas turi vykti realiu laiku.				
Atitikimo kriterijus	Delno pozicija atnaujinama bent 25 kartus per sekundę.				
Kliento pasitenkinimas	2	Kliento nepasitenkinimas		3	
Prioritetas	Vidutinis			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

Reikalavimas	28	Reikalavimo tipas	Reikalavimas vykdymo savybėms	Panaudos atvejis	PA1, PA2
Aprašymas	Sistema turi neversti žaidėjo persitempti.				
Pagrindimas	Jei žaidėjas bandys persilenkti daugiau, nei gali, jis gali patirti traumą.				
Atitikimo kriterijus	Sistema toleruoja ne iki galo atliktą persilenkimą.				
Kliento pasitenkinimas	1	Kliento nepasitenkinimas		4	
Prioritetas	Vidutinis			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

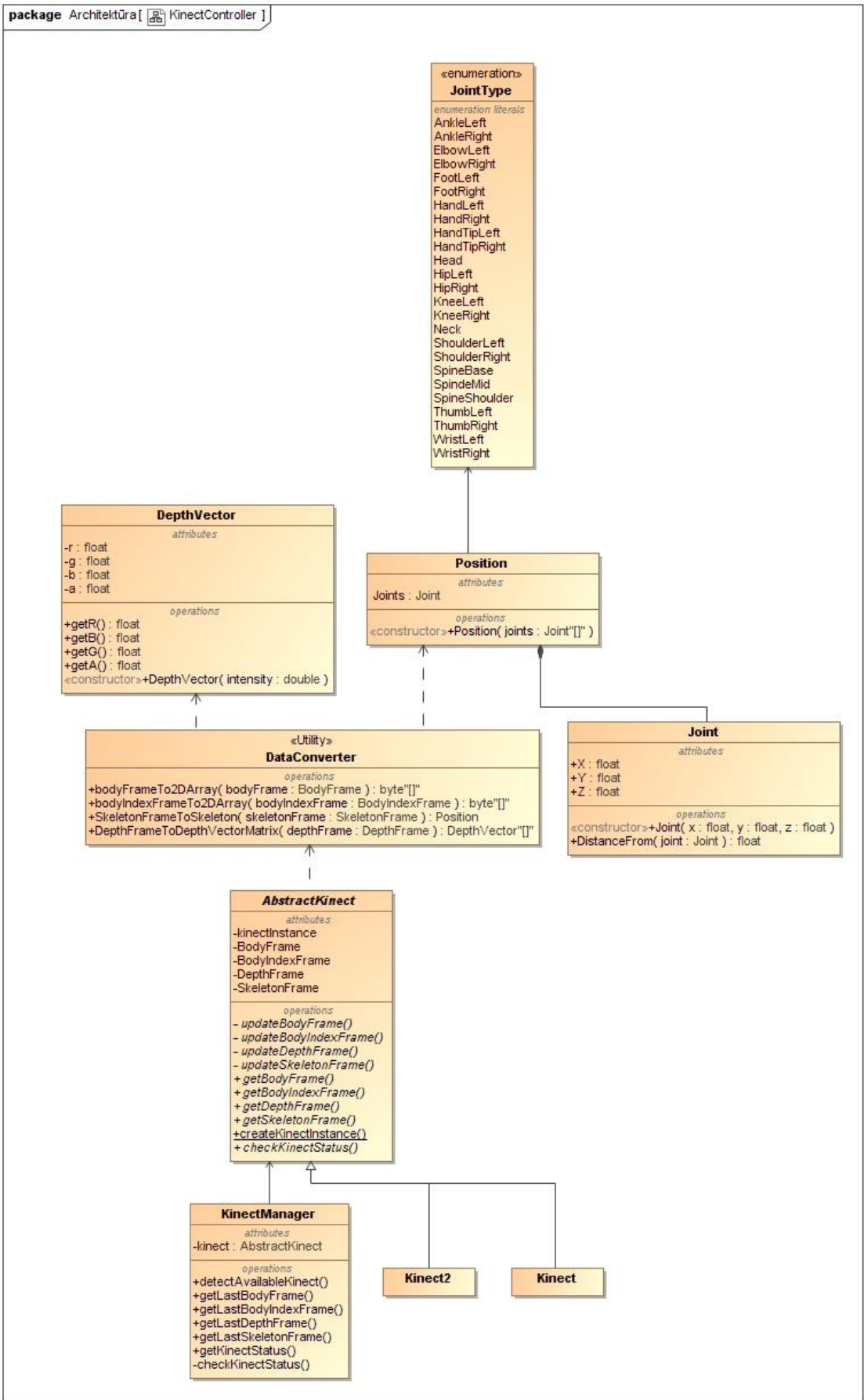
9.1.3.2. Reikalavimai veikimo sąlygoms

Reikalavimas	29	Reikalavimo tipas	Reikalavimas veikimo sąlygoms	Panaudos atvejis	-
Aprašymas	Sistema turi veikti namų sąlygomis.				
Pagrindimas	Sistemą naudos namų vartotojai.				
Atitikimo kriterijus	Sistema veikia numatytoje testinėse aplinkose, panašiose į tipinių namų.				
Kliento pasitenkinimas	1	Kliento nepasitenkinimas		4	
Prioritetas	Vidutinis			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

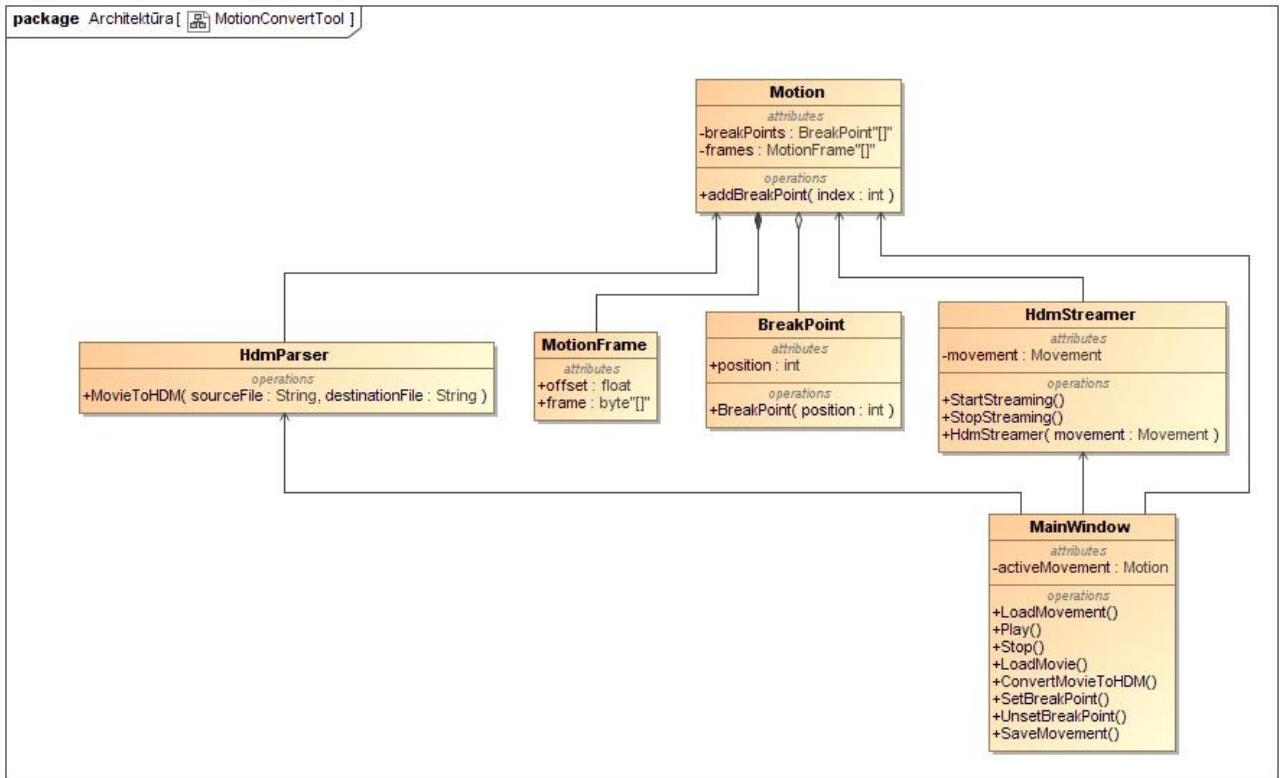
9.1.3.3. Reikalavimai sistemos priežiūrai

Reikalavimas	30	Reikalavimo tipas	Reikalavimas sistemos priežiūrai	Panaudos atvejis	-
Aprašymas	Sistema ateityje gali būti pritaikoma XBox žaidimų konsolėi.				
Pagrindimas	Žaidimas naudoja Kinect jutiklius, kurie platinami kartu su XBox konsolė.				
Atitikimo kriterijus	Sistema turi būti perkeliama XBox platformai per 10% produkto kūrimo laiko.				
Kliento pasitenkinimas	1	Kliento nepasitenkinimas		1	
Prioritetas	Žemas			Konfliktai	-
Papildanti medžiaga	-				
Istorija					

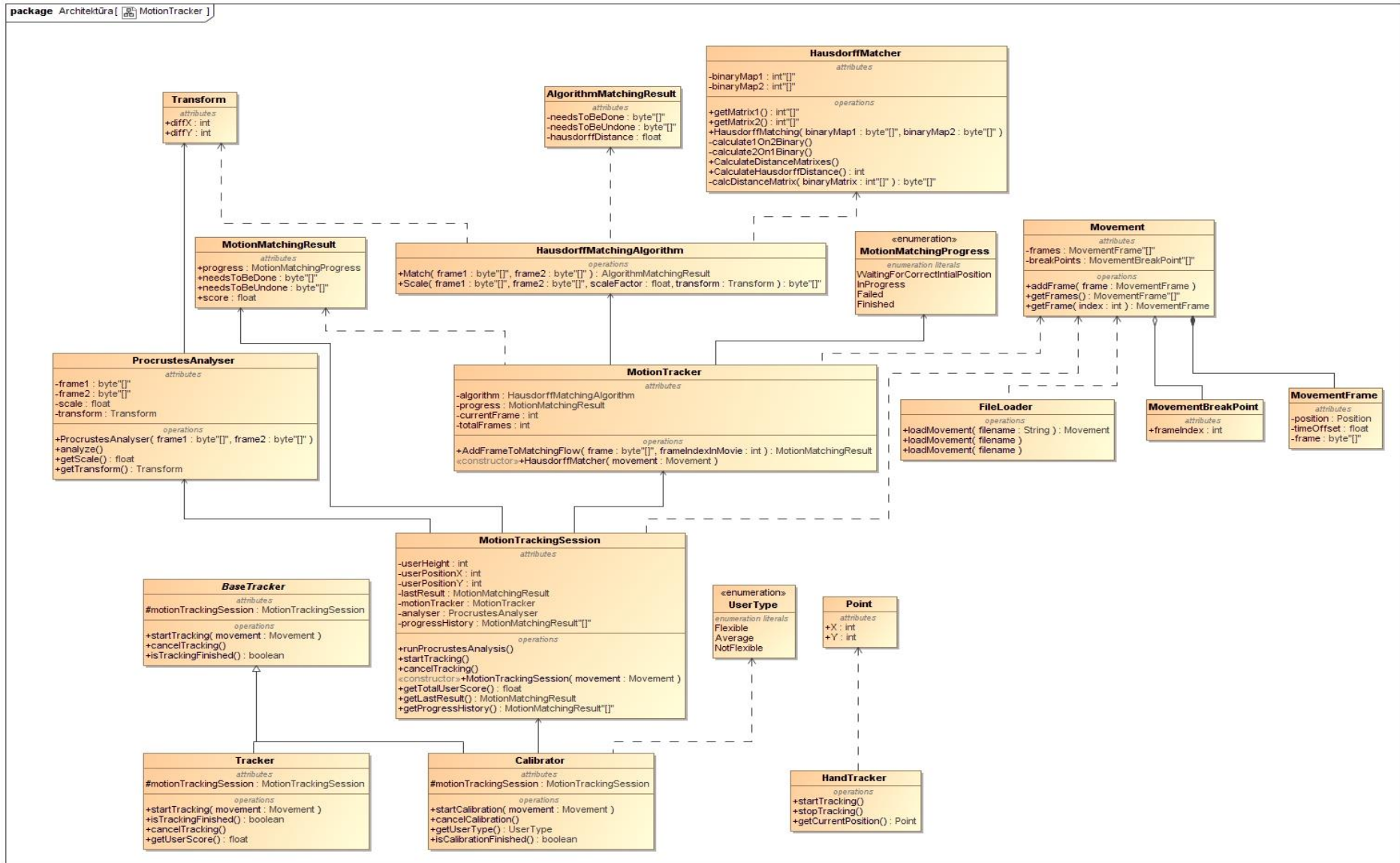
9.2. Detalios sistemos klasių diagramos



9.1 pav. Paketo „KinectController“ detali klasių diagrama



9.2 pav. Paketo „MotionConvertTool“ detali klasių diagrama



9.3 pav. Paketo „MotionTracker“ detali klasių diagrama

9.3. Žmogaus sąnarių padėties nustatymo naudojant kelis „Kinect“ įrenginius algoritmo sudarymas ir tyrimas

Žmogaus sąnarių padėties nustatymo naudojant kelis Kinect įrenginius algoritmo sudarymas ir tyrimas

Tautvydas Petkus, Karolis Ryselis

KTU Programų inžinerijos katedra

Kauno technologijos universitetas

Kaunas, Lietuva

Tautvydas.Petkus@ktu.edu, Karolis.Ryselis@ktu.edu

Šiame straipsnyje apžvelgiama Žmogaus sąnarių atpažinimo sensorių problema atskirti visus žmogaus taškus priklausomai nuo asmens stovėjimo, pasisukimo padėties. Pateikiamas būdas, kaip išvengti sensorių trūkumų – fiksuoti žmogų su keliais sensoriais, sujungiant jų užfiksuotus duomenis į vieną žmogaus skeletą. Tyrime įvertiname mūsų gautų rezultatų tikslumą pagal duotą algoritmą ~1,5% tikslumu nustatant žmogaus sąnarių koordinates.

Reikšminiai žodžiai: įvesties įrenginiai, Kinect sensorius, skeletų sujungimas.

- IŽANGA

2010 m. Pristatytas Microsoft Kinect įrenginys, suteikiantis galimybę sensoriaus pagalba atpažinti žmogaus skeletą ir judesius. Sensoriaus apdorotą informaciją galima panaudoti įvairių aplikacijų kūrimo – žaidimuose, interaktyviose simuliacijose. Iš viso galima atpažinti 20 skirtingų žmogaus kūno dalių. Šio tyrimo tikslas yra kompensuoti įrenginio trūkumus pateikiant struktūrinį įrenginio panaudojimą bei algoritmo pritaikymą. Mūsų naudojamas Kinect įrenginys pavaizduotas (pav. 1) nuotraukoje.



1 pav. Kinect sensorius

Kinect sensorius turi apribotas veikimo galimybes: jis gali atpažinti asmenis, stovinčius prieš įrenginį nuo 0,8 iki 2,5 metrų. Kinect matymo kampas yra 47° (horizontaliai) ir 57° (vertikaliai). Kinect įrenginys nėra pritaikytas atpažinti žmogų, stovintį profiliu, nes ne visi kūno sąnariai yra matomi priešais kamerą [1]. Tokie sensoriaus techniniai apribojimai neleidžia panaudoti įrenginio platesniems, sudėtingesniems projektams ar užduotims, kaip pilnam žmogaus judėjimo atpažinimui erdvėje.

Susiję projektai analizuoja panašias problemas. Osakos Universiteto darbuotojų moksliniame tyrime analizuojamas kelių Kinect sensorių gylių sujungimas [3]. Tyrime pagal pateiktą H. Nakajima problemos sprendimo būdą sulieti kelis gylio taškų debesis į vieną sistemą. Mūsų atveju, mums aktualu yra sujungti kelis žmogaus sąnarių taškus.

Rochesterio technologijų instituto Ijano Vitenbergo pateikiamas tyrimas sprendžia skeleto taškų sujungimo problemą. Straipsnyje pateikiame sprendimo būdą sukuriama virtualūs žmogaus skeleto taškai, kurie leidžia išlaikyti žmogaus proporcijas [4].

- SĄNARIŲ KOORDINAČIŲ NUSTATYMAS NAUDOJANT KELIS SENSORIUS

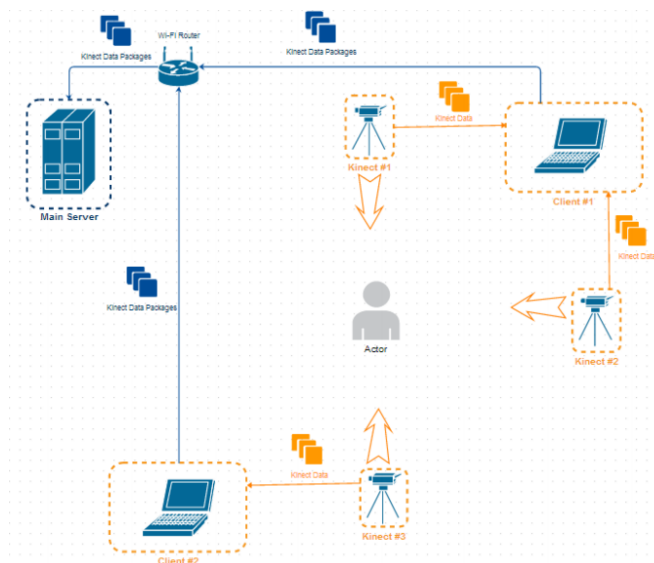
Norint išspręsti šias technines problemas, galime panaudoti kelis sensorius atpažįstant asmenį pasirinktoje erdvėje. Tinkamai išplanavus Kinect įrenginius norimoje patalpoje, galime išspręsti riboto sensoriaus apimties kampo ir atstumo problemą. Taip pat, kelių kamerų pagalba galime atpažinti žmogaus sąnarius nepriklausomai nuo jo pasisukimo kampo – jeigu vienas sensorius negali korektiškai atpažinti žmogaus skeleto, kitas sensorius, esantis kambaryje, gali jį atpažinti ir suteikti tikslią informaciją. Panaudojus visų sensorių, esančių kambaryje duomenis, galime sujungti Kinect įrenginių užfiksuotus žmonių skeletus ir sujungti į vieną bendrą modulį.

Šis problemos sprendimo būdas turi kelis apribojimus:

1. Infraraudonųjų spindulių interferencija. Pagal Microsoft oficialius duomenis, kelių Kinect įrenginių palaikymas vienu metu gali sukelti interferencijos problemą dėl to, jog sensorius skleidžia infraraudonuosius spindulius. To pasekoje galimas įrenginių tikslumo praradimas [2].
2. Papildoma programinė įranga duomenų persiuntimui. Kinect sensorių palaikymo kiekis skiriasi priklausomai nuo USB valdiklio tipo, įrengto asmeniniame kompiuteryje. Norint palaikyti 3 ar daugiau Kinect įrenginių vienu metu, reikia juos prijungti prie skirtingų kompiuterių, kurie savo duomenis siųs bevieliu ryšiu į pagrindinį serverį, kuris gautus duomenis apdoros. Visai šiai duomenų judėjimo realizacijai reikalinga programinė įranga, užtikrinanti duomenų judėjimą.

- Sensorių Sinchronizacija ir Duomenų apjungimas. Norint sužinoti tikslią žmogaus padėtį kelių Kinect įrenginių pagalba, turime kiekvieno sensoriaus skeletą sujungti su kitų sensorių skeletais, užfiksuotais tuo pačiu fiksavimo metu. Kad sujungtume kelių Kinect įrenginių atpažintus skeletus, siūlome algoritmą, aprašytą 4 skyriuje.

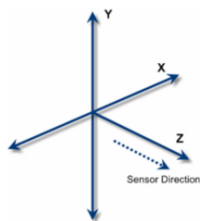
Žemiau (2 pav.) pateikiame sistemos išdėstymo schemą, kai panaudojami keli Kinect įrenginiai žmogaus sąnarių padėties nustatymui.



2 pav. Sistemos išdėstymo schema

Sistemoje naudojami trys Kinect sensoriai skeleto atpažinimui bei trys asmeniniai kompiuteriai sensorių prijungimui bei duomenų apdorojimui atlikti. Taip pat naudojama bevielio tinklo stotelė duomenų perdavimui tarp kompiuterių. Visa tai išdėstyta erdvioje aplinkoje, kurioje keturių metrų atstumu vienas nuo kito pastatyti trys Kinect sensoriai, savo matymo kampą apimantys centre esantį žmogų.

Kiekvieno sensoriaus gautos žmogaus sąnarių koordinatės siunčiamos tokioje koordinacių sistemoje, kurios centras yra pats sensorius (pav. 3). Norint sujungti žmogaus sąnarių pozicijas, gaunamas iš kelių Kinect sensorių, reikia gautus duomenis suvesti į bendrą koordinacių sistemą, o po to jiems pritaikyti algoritmą, kuris apibendrintų duomenis bendroje koordinacių sistemoje.



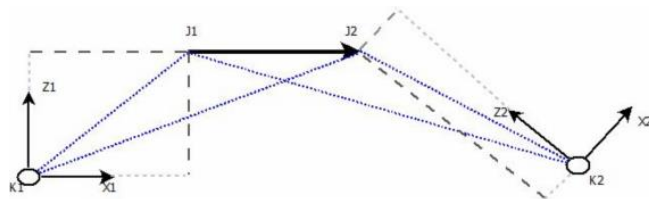
3 pav. Sensoriaus koordinacių sistema

• SĄNARIŲ KOORDINACIŲ SKAIČIAVIMO ALGORITMAS

Naudosime tiesinėmis transformacijomis pagrįstą metodą. Algoritmo pradiniai duomenys - žmogaus sąnario koordinacių šaltai, kuriuos siunčia Kinect įrenginiai. Algoritmo gražinamas

rezultatas - atitinkamo sąnario koordinatės, apskaičiuotos įvertinus kiekvieno Kinect įrenginio siunčiamą informaciją.

Pirmoji algoritmo dalis - suvesti koordinacių šaltus į bendrą koordinacių sistemą.



4 pav. Dviejų sensorių panaudojimo scenarijus

Nagrinėjame dvimatį atvejį, kai yra du Kinect sensoriai K_1 ir K_2 bei du stebimi sąnariai J_1 ir J_2 (pav. 4). Sudarome vektorių $\overrightarrow{J_1J_2}$. Abu Kinect sensoriai turi savo koordinacių sistemas. Bendrąja koordinacių sistema pasirenkame Kinect sensorius K_1 . Tuomet reikia pasukti ir pastumti erdvėje koordinacių sistemą K_2 , kad ji sutaptų su K_1 .

Randame kampą tarp koordinacių sistemų, tai yra, kampą tarp vektorių $\overrightarrow{K_1Z_1}$ ir $\overrightarrow{K_2Z_2}$. Pasirenkame vieną iš sąnarių, tarkime J_1 . Nagrinėjame vektorius $\overrightarrow{K_1J_1}$ ir $\overrightarrow{K_2J_1}$. Randame kampus tarp šių vektorių ir atitinkamai vektorių $\overrightarrow{K_1Z_1}$ ir $\overrightarrow{K_2Z_2}$. Nagrinėjame vektorių $\overrightarrow{K_1J_1}$ Kinect sensoriaus K_1 koordinacių sistemoje. Sakykime, jo koordinatės $[x_1; z_1]$. Vektoriaus $\overrightarrow{K_1Z_1}$ koordinatės $[1; 0]$. Tuomet kampas tarp šių vektorių bus lygus:

$$\cos(\varphi_1) = \frac{x_1 \cdot 1 + z_1 \cdot 0}{\sqrt{x_1^2 + z_1^2} \cdot \sqrt{1^2 + 0^2}} = \frac{x_1}{\sqrt{x_1^2 + z_1^2}} \quad (1)$$

$$\varphi_1 = \arccos\left(\frac{x_1}{\sqrt{x_1^2 + z_1^2}}\right). \quad (2)$$

Žinant, kad $\cos(\alpha) = \cos(2\pi - \alpha)$, negalima pasakyti, koks yra tikrasis kampas: α ar $2\pi - \alpha$. Kadangi $-\frac{\pi}{2} \leq \arccos(\alpha) \leq \frac{\pi}{2}$, tai tuo atveju, kai koordinatė $x_1 \geq 0$, laikome, kad gautas kampas yra teisingas. Priešingu atveju pritaikome korekciją atimdami kampą iš 2π . Galime užrašyti:

$$\varphi_1 = \begin{cases} \arccos\left(\frac{x_1}{\sqrt{x_1^2 + z_1^2}}\right), & \text{jei } x_1 \geq 0 \\ 2\pi - \arccos\left(\frac{x_1}{\sqrt{x_1^2 + z_1^2}}\right), & \text{jei } x_1 < 0 \end{cases} \quad (5)$$

Analogiškai apskaičiuojame kampą φ_2 tarp vektorių $\overrightarrow{K_2J_1}$ ir $\overrightarrow{K_2Z_2}$.

Priimant, kad bendroji koordinacių pradžia sutampa su pirmojo Kinect sensoriaus koordinacių pradžia, skaičiavimus vykdome atsižvelgdami į jo koordinacių sistemos orientaciją erdvėje. Pasirenkame bet kurį tašką J , kurio koordinatės norime perskaičiuoti iš koordinacių sistemos K_2 į sistemą K_1 . Kadangi koordinacių centrai turi sutapti, kampai tarp nagrinėjamų vektorių taip pat turi sutapti, todėl vektoriui $\overrightarrow{K_1J}$ reikia pritaikyti pasukimą kampu $\varphi_1 - \varphi_2$, tai yra, tokiu kampu, kuris būtų skirtumas tarp kampų, kuriais yra orientuotas atskaitinis vektorius abiejų koordinacių sistemų erdvėse. Toks pasukimas yra ekvivalentus koordinacių sistemos pasukimui kampu $\varphi_2 -$

φ_1 . Pirmiausia pervedame taško koordinatas į polinę koordinačių sistemą. Tuomet taško koordinatės, remiantis kosinusų teorema, tampa

$$[R; \varphi] = [\sqrt{x_j^2 + z_j^2}; \text{acos}(\frac{x_j}{\sqrt{x_j^2 + z_j^2}}) + C], \text{ kur}$$

$$C = \begin{cases} 0, & \text{jei } x_j \geq 0 \\ \pi, & \text{jei } x_j < 0 \end{cases}$$

φ reikšmę padidiname dydžiu $\varphi_1 - \varphi_2$ ir koordinatas $[R; \varphi + \varphi_1 - \varphi_2]$ vėl pervedame į stačiakampę koordinačių sistemą. Gauname koordinatas:

$$[x_j'; z_j'] = [R \cos(\varphi + \varphi_1 - \varphi_2); R \sin(\varphi + \varphi_1 - \varphi_2)]. \quad (6)$$

Kadangi dabar abi koordinačių sistemos yra vienodai orientuotos erdvėje, galime perskaičiuoti koordinatas tarp abiejų sistemų naudodami tiesinį postūmį. Sakykime, turime dvi vienodai orientuotas koordinačių sistemas su centrais Q_1 ir Q_2 bei tašką P . Akivaizdu, jog $\overrightarrow{Q_1 Q_2} = \overrightarrow{Q_1 P} - \overrightarrow{Q_2 P}$. Vektoriaus $\overrightarrow{Q_1 P}$ koordinatės yra taško P koordinatės sistemoje Q_1 , vektoriaus $\overrightarrow{Q_2 P}$ koordinatės yra taško P koordinatės sistemoje Q_2 . Belieka atimti šiuos vektorius. Mūsų atveju, priimant, kad bendrosios koordinačių sistemos atžvilgiu taško koordinatės yra $[x_j; z_j]$, galutinės perskaičiuotos koordinatės iš nagrinėjamos pasuktos koordinačių sistemos yra:

$$[\tilde{x}_j; \tilde{z}_j] = [x_j' - x_j; z_j' - z_j]. \quad (7)$$

Išplečiame algoritmą trimačiam atvejui. Kinect sensorius pateikia aukštį, kuriame jo atžvilgiu yra taškas, nepriklausomai nuo to, kaip jis orientuotas, todėl vienintelis faktorius, nuo kurio priklauso aukščio reikšmė Kinect atžvilgiu yra skirtumas tarp taško ir Kinect sensoriaus absoliutinių aukščių, su sąlyga, kad Kinect sensoriai yra orientuoti taip, kad vektorius \overrightarrow{KZ} yra lygiagretus plokštumai, ant kurios stovi žmogus. Atskaitos tašku pasirinkę bet kurį tašką ir pamatavę jo aukštį keleto Kinect sensorių atžvilgiu, nesunkiai gausime aukščio skirtumus tarp Kinect sensorių. Išlaikomas ankstesnis žymėjimas: $y_j' = y_j$, kadangi nereikia atlikti persukimų. Galutinė koordinatės reikšmė bus $\tilde{y}_j = y_j' - y_j$.

Taigi, perskaičiavimai, reikalingi pervesti taško koordinatas tarp skirtingų Kinect sensorių koordinačių sistemų, yra tokie:

$$[\tilde{x}; \tilde{y}; \tilde{z}] = [x' - x; y' - y; z' - z], \text{ kur} \quad (8)$$

$$x' = R \cos(\varphi + \varphi_1 - \varphi_2)$$

$$y' = y_1, \text{ kur}$$

$$z' = R \sin(\varphi + \varphi_1 - \varphi_2) \quad (9)$$

$$R = \sqrt{x_j^2 + z_j^2}$$

$$\varphi = \text{acos}(\frac{x_j}{\sqrt{x_j^2 + z_j^2}}) + C$$

$$C = \begin{cases} 0, & \text{jei } x_j \geq 0 \\ \pi, & \text{jei } x_j < 0 \end{cases}, \text{ kur } (10)$$

$$\varphi_n = \begin{cases} \text{acos}(\frac{x_n}{\sqrt{x_n^2 + z_n^2}}), & \text{jei } x_n \geq 0 \\ 2\pi - \text{acos}(\frac{x_n}{\sqrt{x_n^2 + z_n^2}}), & \text{jei } x_n < 0 \end{cases}, n = 1, 2$$

x_n, y_n, z_n - atskaitinio vektoriaus koordinatės n -tojo Kinect sensoriaus koordinačių sistemoje.

x_j, y_j, z_j - norimo perskaičiuoti taško koordinatės nebazinio Kinect sensoriaus sistemoje.

Matome, jog transformacijos, pritaikomos vektoriui, parametrai priklauso nuo vektoriaus $\overrightarrow{J_1 J_2}$, o juos kartą apskaičiavus galima taikyti visiems taškams. Kad būtų užtikrinamas didesnis tikslumas, tikslinga šiuos parametrus rasti ne vieną kartą, o kelis. Tokiu būdu eliminuojame atsitiktinių koordinačių fliuktuacijų tikimybę. Iš visų sukauptų parametrų variantų reikia išrinkti vieną. Tai darysime skaičiuodami jų vidurkį arba medianą.

Turint galutinius transformacijos parametrus galima juos taikyti visiems taškams. Siekiant padidinti tikslumą, taip pat galima sukaupti reikšmes į buferį, ir naudoti arba reikšmių vidurkį, arba medianą. Didelis buferis užtikrins gerą tikslumą, tačiau duos lėtesnį reakcijos laiką, nes bus įvertinami duomenys iš ankstesnių laiko momentų.

Norint padidinti tikslumą galima neįskaičiuoti tų reikšmių, kurių Kinect sensorius tiesiogiai nemato, o tik bando nuspėti jų reikšmes. Taikant tokį filtravimą ir išdėliojus Kinect sensorius skirtinguose aukščiuose galima pasiekti didelį tikslumą, nes visas aukščių spektras bus tiesiogiai matomas.

• SAŅARIŲ KOORDINAČIŲ NUSTATYMO TIKSLUMO TYRIMAS

Tyrimo aplinkai pasirinktas erdvus kabinetas, kuriame keturių metrų atstumu vienas nuo kito pastatyti trys Kinect sensoriai, savo matymo kampu apimantys visą centrinę erdvę. Eksperimentai buvo atlikti vakare, erdvę apšviečiant dienos šviesos lempomis.

Šiame tyrime išsiaiškinsime, kurią algoritmo variaciją geriausia taikyti, kad būtų gaunamas tikslesnis rezultatas. Tirsime aprašyto algoritmo veikimą pritaikius duomenų buferizavimą, buferio dydžio įtaką algoritmo veikimui bei algoritmo veikimą, kai atmetami nematomi taškai.

Tirsime taip: įjungiamo aprašyta sistema su trimis Kinect sensoriais ir jų siunčiami duomenys pervedami į vieno iš sensorių koordinačių sistemą. Teoriškai kiekvieno taško koordinatės, gautos iš skirtingų sensorių, po perskaičiavimų turėtų būti vienodos. Realiu atveju pastebime reikšmių svyravimus. Metodas bus tuo tikslesnis, kuo mažiau bus išsibarstę reikšmės po perskaičiavimų. Kaip kiekybinį matą naudojame gautų reikšmių dispersiją. Skaičiuosime dispersijas

atskirai kiekvieno taško kiekvienai koordinatei ir suskaičiuosime tų dispersijų vidurkius iš duomenų imties. Kiekviename tyrime panaudota bent 800 skeleto pozicijų.

Sukaupto vektorių buferio bendros reikšmės išvedimo algoritmo tyrimas. Bazinio vektoriaus koordinatės galime tikslinti sukauptą jų buferį ir suskaičiuosime sukauptų vektorių koordinatė vidurkius arba medianą. Žemiau esančioje lentelėje (Lentelė 1) pateikti tyrimų rezultatai:

LENTELĖ I. BUFERIZUOTO ALGORITMO TYRIMAS

	Vidurkis	Mediana
D(x)	0,042	0,032
D(y)	0,010	0,015
D(z)	0,027	0,050
MD	0,026	0,032

Iš lentelės matome, kad skaičiuojant reikšmių vidurkį dispersija gaunama šiek tiek mažesnė, vidutinis kvadratinis nuokrypis siekia ~0,16, tai yra, ~2% nuokrypis.

Kalibracijos buferio dydžio parinkimo tyrimas. Paaiškėjus, jog buferizuojant duomenis galima gauti mažesnę nuokrypį, reikia iširti, kaip tikslumas priklauso nuo buferio dydžio. Žemiau esančioje lentelėje (Lentelė 2) pateikti tyrimų rezultatai gauti atlikus eksperimentus su 1, 50 ir 100 sąnarių pozicijų buferiais:

LENTELĖ II. BUFERIO DYDŽIO ĮTAKOS TYRIMAS

	1	50	100
D(x)	0,013	0,056	0,032
D(y)	0,014	0,012	0,009
D(z)	0,022	0,024	0,021
MD	0,016	0,031	0,021

Iš lentelės matome, jog neverta didinti buferio gylį, nes tikslumo nelaimime, o buferio kaupimas veda į lėtesnį algoritmo veikimą, nes buferio sukaupimas yra išėstas laike. Netaikant buferio gaunamas vidutinis nuokrypis ~0,13, tai yra, gaunamas nuokrypis ~1,5%.

Nematomų taškų atmetimo įtakos tyrimas. Norint gauti tikslesnius duomenis, galima pamėginti neįtraukti į skaičiavimus tų taškų, kurių Kinect sensorius tiesiogiai nemato. Žemiau esančioje lentelėje (Lentelė 3) pateikti tyrimų rezultatai gauti atmetus nematomus ir įskaičiuojant matomus taškus:

LENTELĖ III. NEMATOMŲ TAŠKŲ ATMETIMO TYRIMAS

	Įskaičiuoti	Atmesti
D(x)	0,056	0,412
D(y)	0,012	0,051
D(z)	0,023	0,588
MD	0,031	0,350

Kaip matome iš lentelės, toks būdas ne visada efektyvus, nes atsiranda situacijų, kai kuris nors taškas tampa visiškai nematomu, ir tai veda į dideles paklaidas.

• IŠVADOS

Šiame straipsnyje nagrinėjamas tiesinėmis transformacijomis grįstas algoritmas, skirtas pasiekti kuo didesnę tikslumą atliekant žmogaus padėties matavimus keletu Kinect sensorių. Algoritmas remiasi pasirinkto vektoriaus, sudaryto iš dviejų Kinect sensoriais matomų taškų, reliatyvia padėtimi kiekvieno Kinect sensoriaus atžvilgiu. Tikslumui įvertinti buvo atliktas eksperimentas, kurio metu buvo skaičiuojami gautų reikšmių nuokrypiai. Jo metu gautos paklaidos siekia 1,5%. Toks tikslumas leidžia taikyti algoritmą praktikoje. Kinect sensoriai yra pakankamai tikslūs, todėl gautų koordinatė buferizavimas nėra efektyvus. Be to, jiems sudaryti reikalingas laikas, dėl to sumažėja apdorojamų kadrų skaičius per laiko vienetą. Tikslumą būtų galima padidinti įvertinant Kinect sensoriaus posvirį aukštyr ar žemyn, tačiau šis tikslinimas išėina už šio tyrimo ribų ir jį galima taikyti lygiai taip, kaip ir vienam skeletui.

• LITERATŪRA

D. Catuhe, „Programming with the Kinect for Windows Software Development Kit: Microsoft Press“, 2012.
 Microsoft Developer Network - Skeletal Tracking <http://msdn.microsoft.com/en-us/library/hh973074.aspx#ID4ENB>.
 M. Nakazawa, I. Mitsugami, Y. Makihara, H. Nakajima, H. Habe, H. Yamazoe, Y. Yagi, „Dynamic Scene Reconstruction using Asynchronous Multiple Kinects“, 2012.
 I. Wittenberg, „Performance and Motion Capture Using Multiple Xbox Kinects“, 2013.

9.4. Efektyvus Hausdorfo atstumo metrikos skaičiavimo algoritmas žmogaus padėties palyginimui su šablonu ir jo greitaveikos ir tikslumo tyrimas

Efektyvus Hausdorfo atstumo metrikos skaičiavimo algoritmas žmogaus padėties palyginimui su šablonu ir jo greitaveikos ir tikslumo tyrimas

Karolis Ryselis, Tautvydas Petkus
KTU Programų inžinerijos katedra
Kauno technologijos universitetas
Kaunas, Lietuva

Karolis.Ryselis@ktu.edu, Tautvydas.Petkus@ktu.edu

Santrauka – Šiame straipsnyje pristatomas efektyvus Hausdorfo atstumo metrikos skaičiavimo algoritmas bei pateikiamas euristinis algoritmas, leidžiantis maksimizuoti dviejų siluėtų persidengiančių taškų skaičių per labai trumpą laiką. Taip pat pristatomi šių algoritmų greitaveikos bei tikslumo tyrimai. Parodoma, jog algoritmas tinkamas taikyti žmogaus padėties sekimui realiu laiku su „Kinect“ įrenginiu.

Raktiniai žodžiai: Kinect, Hausdorfo atstumas, lyginimas su šablonu

I. ĮŽANGA

Microsoft pristatytas žmogaus judesių sekimo jutiklis „Kinect“ yra vienas iš populiariausių namuose naudojamų judesių sekimo prietaisų. „Kinect“ programuojama sąsaja pateikia keletą duomenų srautų su duomenimis apie dabartinę žmogaus padėtį [30]: skeleto srautą, kuriame pateikiama apdorota informacija apie žmogaus sąnarių padėtis; gylio srautas, kurį sudaro dvimatė matrica su atstumo nuo sensoriaus iki objekto duomenimis; kūno indekso matrica, kuri kiekvieną vaizdo tašką priskiria sekamam žaidėjui. Skeleto srautas pateikia informaciją apie atskiras žmogaus kūno dalis, todėl taikymo srityse, kur reikia identifikuoti kūno dalis, yra patogiausias. Tačiau tyrimai [31] rodo, kad skeleto srauto informacija tiksliai yra ne visada, o tik tuomet, kai žmogus yra gerai matomas „Kinect“ jutikliui ir stovimoje padėtyje. Kitose padėtyse duomenys iškraipomi.

Kitų duomenų srautų analizei reikalingi algoritmai, kurie leistų palyginti žmogaus padėtį su šablonu. Visi kiti duomenų srautai grąžina didelį duomenų kiekį lyginant su interpretuota skeleto informacija, todėl reikia taikyti efektyvius skaičiavimo metodus, jei keliamas reikalavimas lyginti žmogaus padėtį su šablonu realiu laiku ar greičiu, artimu realaus laiko palyginimui.

Paprastiausias ir lengviausiai apdorojamas duomenų srautas yra kūno indekso srautas. Jeigu sekamas tik vienas žmogus, šis srautas nesunkiai suvedamas į dvejetainę matricą, kurioje 1 reiškia, kad tame taške žmogaus kūno taškas yra, 0 – kad nėra. Tokiems duomenims lyginti tinkama metrika yra Hausdorfo atstumas [32]. Tarkime, kad $A = \{a_1, a_2, \dots, a_n\}$ ir

$B = \{b_1, b_2, \dots, b_m\}$ yra baigtinio dydžio taškų aibės. Tuomet Hausdorfo atstumas apibrėžiamas taip:

$$H(A, B) = \max(h(A, B), h(B, A)), \text{ kur} \quad (1)$$

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \quad (2)$$

Čia $\|a - b\|$ gali būti bet kokia atstumo metrika tarp taškų a ir b , pvz., Euklido norma ar L_2 norma. Žvelgiant semantiškai, ši metrika parodo labiausiai nuo kitos aibės nutolusio taško atstumą iki tos aibės pagal pasirinktą atstumo metriką.

Iš (2) formulės matome, kad tiesioginis metrikos skaičiavimas yra $O(nm)$ sudėtingumo, nes reikia patikrinti abiejų aibių visus taškus tarpusavyje. Jei $m \approx n$, tai sudėtingumas artėja į $O(n^2)$ ir realaus laiko taikymams nėra tinkamas.

II. HAUSDORFO ATSTUMO SKAIČIAVIMO BŪDAI

A. Yanir Taflev siūlomas skaičiavimo būdas

Yanir Taflev siūlo atviro kodo sprendimą Hausdorfo atstumų matricos skaičiavimui [25]. Šis algoritmas paties Hausdorfo atstumo neskaičiuoja, tačiau sudaro Hausdorfo atstumų matricas. Hausdorfo atstumų matrica yra matrica, kurios kiekvienas elementas parodo atstumą iki artimiausio aibės B taško kiekvienam aibės A taškui. Apskaičiavus šias matricas abiejų aibių atžvilgiu, Hausdorfo atstumas bus lygus abiejų matricų maksimumui.

Hausdorfo atstumų matrica skaičiuojama sudarant pradinę matricą, kurios visiems taškams, esantiems aibėje B , suteikiamos reikšmės 0, nesantiems - ∞ . Tuomet nuo kiekvieno aibės B taško didėjančios kraštinės dydžio kvadratais keliaujama per matricą ir surašomos reikšmės, lygios to kvadrato kraštinės dydžiui, jei tame taške dar nėra mažesnės reikšmės. Dėl tokios matricos sudarymo specifikos gaunami atstumai pagal Čebyševio atstumo metriką.

Pateikiama algoritmo realizacija C# kalba. Ši realizacija naudojama C# karkase „Shape Matching Framework“.

B. Elastic Search siūlomas skaičiavimo būdas

Elastic Search klientas, sukurtas Vivid Solutions, pateikiamas kaip atvirojo kodo Java kalba sukurtas projektas

[26]. Jame naudojama diskretizuoto Hausdorfo algoritmo realizacija. Algoritmas Hausdorfo atstumą skaičiuoja apytiksliai. Algoritmas nepateikiamas. Realizacija naudojama Elastic Search paieškos variklio geometrijų palyginimui.

C. Princetono universiteto siūlomas skaičiavimo būdas

Princetono universiteto sukurtoje sprendimo bibliotekoje „RgResolve“ naudojama Hausdorfo algoritmo Java realizacija [28]. Algoritmas remiasi prielaida, kad taškai, kurie bus nagrinėjami, surikiuoti kryptimi pagal laikrodžio rodyklę ir algoritmas analizuoja juos būtent tokia tvarka. Skirtingai, nei Yanir Taflev realizacijoje, duomenys analizuojami ne dvimatės matricos, bet taškų sąrašo pavidalu. Atstumas matuojamas pagal Euklido normos metriką.

III. SIŪLOMAS HAUSDORFO ATSTUMO SKAIČIAVIMO ALGORITMAS

„Kinect“ gražiname duomenų kadre žmogaus siluetas gali būti bet kurioje vietoje. Gaunamas įvertinimas neturi priklausyti nuo to, kurioje vietoje atsistojo žmogus, todėl prieš atliekant atstumo skaičiavimą reikia atlikti transformaciją, kuri perkeltų siluetą taip, kad jis maksimaliai persidengtų su šablonu.

Efektyviam Hausdorfo atstumo skaičiavimo algoritmui keliami pagrindiniai reikalavimai:

1. Veikimo laikas artimas 30 ms su „Kinect“ kūno indekso srauto duomenimis
2. Algoritmas turi surasti optimalią ar artimą optimaliai tiesinę transformaciją, kurią atlikus šablonas ir vartotojo siluetas būtų taip erdvėje orientuoti, kad būtų maksimizuotas persidengiančių taškų kiekis

A. Siūlomas optimalios transformacijos radimo algoritmas

Tarkime, kad turime dvi baigtines taškų aibes $A = \{a_1, a_2, \dots, a_n\}$ ir $B = \{b_1, b_2, \dots, b_m\}$. Darome prielaidą, kad žmogus kadre stovės orientuotas taip, kad pagrindo, ant kurio yra žmogus, plokštumos vektorius yra orientuotas vertikalia kryptimi. Ši prielaida eliminuoja pasukimo transformaciją. Tuomet reikia atlikti dvi tiesines transformacijas: mastelio keitimo ir postūmio. Tarkime, kad aibės B centrinis taškas, kurio atžvilgiu bus keičiamas mastelis, yra b_m , optimalus mastelio keitimo koeficientas – s , o optimalios postūmio transformacijos vektorius \vec{t} . Tuomet transformuota taškų aibė

$$B_t = \{b_{t1}, b_{t2}, \dots, b_{tn}\} = b_m + \frac{(b_1 - b_m + \vec{t})}{s}, b_m + \frac{(b_2 - b_m + \vec{t})}{s}, \dots, b_m + \frac{(b_n - b_m + \vec{t})}{s} \quad (3)$$

Po optimalios transformacijos aibė B_t turi maksimizuoti kardinalumo funkciją

$$i(A, B_t) = |A \cap B_t| \quad (4)$$

Tokios funkcijos tikslus apskaičiavimas yra imlus skaičiavimams, todėl tikslinga taikyti euristinius algoritmus.

Greitai apskaičiuojama ir pakankamai tiksli euristika mastelio keitimo transformacijai yra mastelio keitimo koeficientu laikyti abiejų objektų aukščių santykį. Užtikrinus,

kad šablonas ir žmogaus siluetas bus panašiose padėtyse, galima suvienodinti abiejų siluetų aukščius. Tuomet

$$s = \frac{h_A}{h_B}, \text{ kur} \quad (5)$$

$$h_A = \max_{a \in A} a_y - \min_{a \in A} a_y, \quad (6)$$

kur a_y - vertikaliuosios ašies koordinatė.

Greitai apskaičiuojamas postūmio transformacijos algoritmas yra perstumti vieną iš siluetų per vektorių, lygų skirtumui tarp siluetų centroidų. Centroidas apibrėžiamas kaip

$$A_C = \left(\frac{\sum_{a \in A} a_x}{n}; \frac{\sum_{a \in A} a_y}{n} \right) \quad (7)$$

Tuomet

$$b_i = C_B + \frac{(b_i - C_B + C_B - C_A)}{s} \quad (8)$$

Jeigu siluetai iš šablono ir žmogaus yra iš tos pačios judesių sekos, tai galima laikyti, kad transformacijos parametrai nesikeis viso palyginimo eigoje ir juos galima apskaičiuoti judesių palyginimo pradžioje ir vėliau naudoti, taip pagreitinant skaičiavimus. Tuomet

$$t = C_{B1} - C_{A1} \quad (9)$$

$$s = \frac{h_2}{h_1} \quad (10)$$

$$b_i = C_B + \frac{(b_i - C_B + t)}{s} \quad (11)$$

B. Siūlomas Hausdorfo atstumo skaičiavimo algoritmas

Siūlomas algoritmas remiasi Yanir Taflev idėjomis, tačiau yra žymiai geriau optimizuotas greičiui. Naudojant šį algoritmą nereikia duomenų konversijos iš dvimatės matricos į kitus formatus, pvz., taškų sąrašą. Naudojama atstumo metrika – Čebyševio atstumas. Pirmiausia sudaroma dvimatė matrica, kurios visi taškai su tomis koordinatėmis, kuriose siluete yra taškas, užpildomos 0, kiti – ∞ . Sudaroma taškų eilė, kurioje iš pradžių surašomi visi taškai, esantys siluete. Iš eilės imamas taškas ir jam skaičiuojami 8 aplinkiniai taškai, kurių viena ar abi koordinatės skiriasi 1 nuo nagrinėjamo taško. Visiems 8 taškams Čebyševio atstumas lygus centrinio taško Čebyševio atstumui, padidintam 1, jei neviršija tam taškui jau įrašytos reikšmės. Jei gautas mažesnis atstumas, nei jau buvo įrašytas, toks taškas taip pat įrašomas į eilę. Tokiu būdu apskaičiuojamas atstumas visiems taškams. Tuomet pašalinami visi taškai, kurie neegzistuoja antroje aibėje, nes jie neaktualūs. Tokios matricos elementai parodo skirtumus tarp siluetų. Reikia surasti dvi tokias matricas, bet jų skaičiavimai nepriklausomi, todėl juos galima vykdyti lygiagrečiai.

Kad būtų mažiau skaičiavimų, nukerpami matricos kraštai, kuriuose nėra silueto taškų. Gauta apkirpta matrica yra minimali stačiakampė matrica, kurioje yra visi taškai, priklausantys siluetai.

Abiejų matricų bendras maksimumas yra Hausdorfo atstumas. Algoritmo pseudokodas:

```

matrices ← template matrix and user matrix
find common left upper corner of matrices
find common right lower corner of matrices
∀ matrix ∈ matrices
do initialize queue and ret
while queue ≠ ∅
do p ← take from queue

```

```

val ← get p value from ret
val ← val + 1
ret ← min(r, val) ∀ adjacent point
r to p
set points of res that do not belong to
opposing matrix to 0
hausdorff distance = max(max(res[1]),
max(res2))

```

IV. TYRIMO METODAS

Tyrimo metu tiriamos dvi sukurto algoritmo savybės: greitis ir tikslumas.

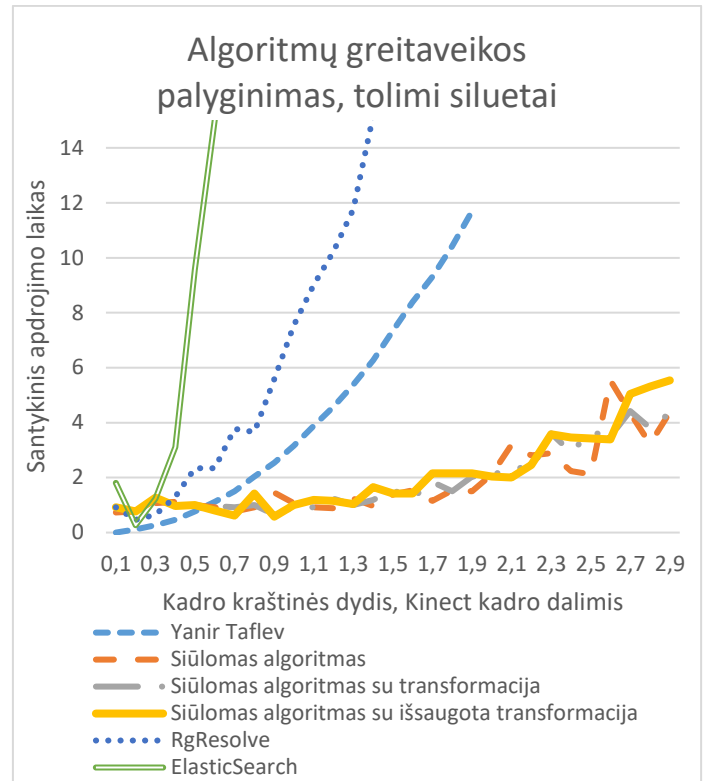
Greičio tyrimo metu sudaromi skirtingų tipų kadrai. Kadrai skiriasi dydžiu ir atstumu tarp siluetai. Naudojami kadro dydžiai yra nuo 10% iki 300% „Kinect“ kadro dydžio, 10% intervalu. Kiekvienam kadro dydžiui tiriama realus žmogaus siluetas. Matuojamas greitis kai siluetai yra toje pačioje ir priešingoje kadro pusėje. Greitis lyginamas su kitų esamų algoritmų greičiu. Kiekvienas matavimas vykdomas 10 kartų ir randamas vykdymo laiko vidurkis. Vykdomi palyginimai su trimis algoritmo versijomis – tik Hausdorfo atstumo skaičiavimo algoritmas, Hausdorfo atstumo skaičiavimas prieš tai pritaikius euristines transformacijas suvienodinti siluetai padėtims ir tas pats algoritmas, tik nevertinant transformacijoms apskaičiuoti reikalingo laiko – laikoma, kad transformacija buvo iš anksto suskaičiuota ir išsaugota. Kad algoritmas būtų pritaikomas, reikia, kad kadro apdorojimo trukmė neviršytų intervalo tarpo iš „Kinect“ siunčiamų kadru, t. y., 30 ms.

Tikslumo tyrimo metu matuojama su realiais skirtingais žmogaus siluetais. Surandama geriausia galima transformacija perrinkimo metodu, apskaičiuojamas Hausdorfo atstumas pritaikius tokią transformaciją, tuomet surandamas Hausdorfo atstumas pritaikius siūlomą algoritmą ir suskaičiuojama paklaida.

V. TYRIMO REZULTATAI

1 pav. parodyta algoritmų visų nagrinėtų algoritmų veikimo trukmės su skirtingais siluetai dydžiais. Visais atvejais buvo naudojamas tie patys siluetai, tik pakeisti jų dydžiai. Kadruose siluetai buvo nutolę vienas nuo kito per maždaug trečdalį kadro.

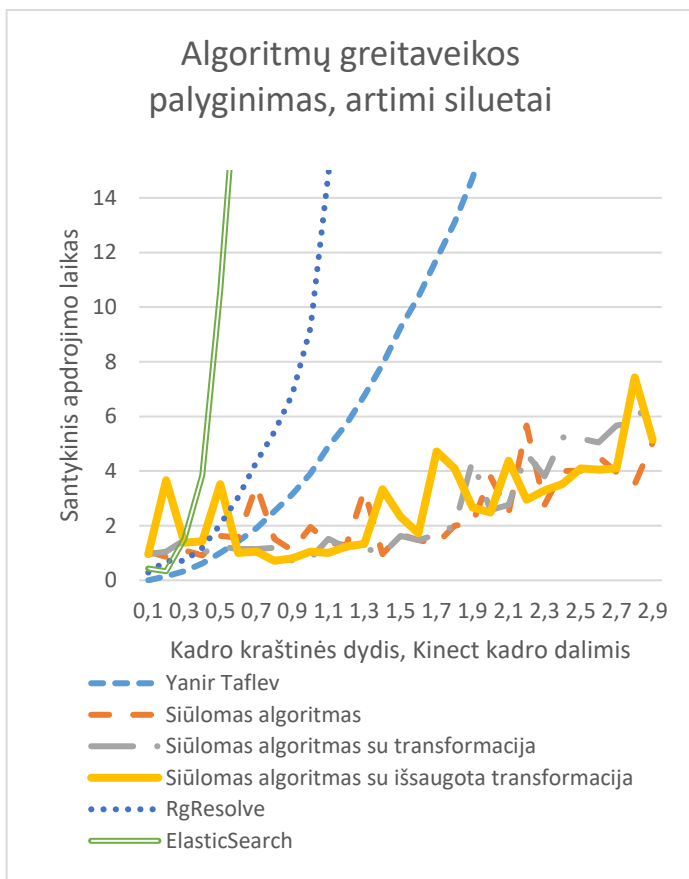
Diagramose vienetinę reikšmę atitinka siūlomo algoritmo su išsaugota transformacija veikimo laikas, kai apdorojamo kadro dydis lygus „Kinect“ kadro dydžiui. Naudojant Intel i7-3770K procesorių (3,5 GHz), tolimiems siluetais šis laikas lygus 26 ms, artimiems – 21 ms. Naudojant šią aparatūrinę įrangą pirmuoju atveju santykinis palyginimo laikas turėtų neviršyti 1,15 reikšmės, antroje diagramoje – 1,43. Šios reikšmės priklauso nuo aparatūros ir skirtos apytiksliam įvertinimui. Šis procesorius truputį viršija „Kinect 2“ sistemos reikalavimus (i7 serijos procesorius, 3,1 GHz dažnis).



Pav. 9.4. Algoritmų greitaikos palyginimas (tolimi siluetai)

Iš gautos diagramos matome, kad ElasticSearch algoritmas yra pats lėčiausias, RgResolve – kiek geresnis, Y. Taflev dar būtų pritaikomas su kadrais, kurių dydis yra apie 0,6 „Kinect 2“ kadro dydžio. Maždaug toks yra „Kinect 1“ jutiklio kadro dydis. Siūlomo algoritmo veikimo laikas smarkiai svyruoja ir aiškios priklausomybės nesimato, tačiau galima įžvelgti apytiksle lėtai kylantią priklausomybę. Tam įtakos turi tai, kad algoritmas yra išlygiagretintas ir sunku prognozuoti tikslų jo veikimo laiką, tačiau grafike pastebima, kad iki maždaug 1,3 kartų už „Kinect“ kadrus didesnių vaizdų apdorojimui dažniausiai užtrunkama iki 1,15 santykinų vienetų, iki 1,7 karto – iki 2 santykinų vienetų.

2 pav. parodyta analogiškų matavimų rezultatai, tik atstumas tarp siluetai buvo minimizuotas. Visiems algoritams, išskyrus siūlomą algoritmą, įtakos tokia transformacija praktiškai neturi, tačiau siūlomas algoritmas veikia pastebimai greičiau: 1,5 santykinio vieneto riba pasiekama maždaug ties 1,4 – 1,5 „Kinect“ kadro dydžio vaizdais, 2,5 – ties 1,8 – 2 kartus didesniais už „Kinect“ kadrus su „Kinect 2“ reikalavimus atitinkančiu procesoriumi. Dydžiui artėjant link 3 „Kinect“ kadru dydžių jaučiamas apie 1,5 karto pagreitis lyginant su tolimų siluetai atveju.



Pav. 9.5. Algoritmų greitimeikos palyginimas (artimi siluetai)

Algoritmo tikslumas buvo išbandytas su skirtingais žmogaus siluetais. Gauta vidutinė paklaida – 3,8%, mažiausia – 0% (tikslus rezultatas), didžiausia – 15,5%, mediana – 2,7%, duomenų imtis – 21 palyginimas.

VI. IŠVADOS

Sukurtas algoritmas veikia greičiau už esamus atvirojo kodo sprendimus, skirtus Hausdorfo atstumui skaičiuoti, tačiau dėl lygiagrečios algoritmo prigimties gaunamas gana didelis vykdymo laiko neapibrėžtumas. Nepaisant to, algoritmas su „Kinect“ kadrais blogiausiu atveju veikia greičiau, nei per 30 ms su „Kinect 2“ reikalavimus atitinkančiu procesoriumi, todėl yra tinkamas sekti žmogaus judesius naudojant „Kinect“ jutiklius. Sukurto algoritmo tikslumas jo taikymo srityje yra gana aukštas ir verta taikyti siūlomą euristiką, kadangi transformacijos koeficientai apskaičiuojami labai greitai, o tikslumo prarandama nedaug.

Sukurtas algoritmas daug greičiau dirba su toje pačioje kadro vietoje esančiais siluetais, nei su esančiais toliau vienas nuo kito. Taikant šį algoritmą judesių sekimo srityje ir prieš matuojant atstumą atlikus rekomenduojamas transformacijas siluetai visada bus toje pačioje kadro vietoje ir palyginimas veiks greitai.

Algoritmas gerai tinka žmogaus padėties sekimui, kai atliekama veikla, kurios metu žmogus nekeičia savo padėties jutiklio atžvilgiu, tik pozą, arba šabloninis siluetas juda kartu su juo. Tokiu atveju gerai suveikia numatytos

euristikos ir pakankamai greitai surandamas gana tikslus Hausdorfo atstumo įvertis Čebyševio atstumo atžvilgiu.

VII. LITERATŪROS ŠARAŠAS

- [1] „Kinect API Overview,“ [Tinkle]. Available: <https://msdn.microsoft.com/en-us/library/dn782033.aspx>. [Kreiptasi 10 03 2017].
- [2] K. Ryselis ir T. Petkus, „Nestandartinių žmogaus kūno pozicijų atpažinimo tikslumo naudojant „Kinect 2.0“ jutiklius tyrimas,“ Kaunas, 2015.
- [3] D. P. Huttenlocher, G. A. Klanderman ir W. J. Rucklidge, „Comparing Images using the Hausdorff Distance,“ IEEE Transactions on Pattern Analysis and Machine Intelligence, nr. 15, pp. 850-863, 1993.
- [4] Y. Taflev, „Using the Hausdorff distance algorithm to point out differences between two drawings,“ 27 09 2009. [Tinkle]. Available: <https://www.codeproject.com/articles/42669/using-the-hausdorff-distance-algorithm-to-point-out>. [Kreiptasi 10 03 2017].
- [5] V. Solutions, „elasticsearch-client/DiscreteHausdorffDistance,“ [Tinkle]. Available: <https://github.com/jprante/elasticsearch-client/blob/master/elasticsearch-client-jts-jdk5/src/main/java/com/vividsolutions/jts/algorithm/distance/DiscreteHausdorffDistance.java>. [Kreiptasi 10 03 2017].
- [6] „Hausdorff Distance,“ Princeton University, [Tinkle]. Available: <http://www.princeton.edu/~rkatzwer/rgsolve/doc/edu/princeton/polygon/HausdorffDistance.html>. [Kreiptasi 12 03 2017].

9.5. Nestandartinių žmogaus kūno pozicijų atpažinimo tikslumo naudojant „Kinect 2.0“ jutiklius tyrimas

NESTANDARTINIŲ ŽMOGAUS KŪNO POZICIJŲ ATPAŽINIMO TIKSLUMO NAUDOJANT „KINECT 2.0“ JUTIKLIUS TYRIMAS *

Tautvydas Petkus, Karolis Ryselis

¹ Kauno technologijos universitetas, Programų inžinerijos katedra, Studentų g. 50, Kaunas, Lietuva, tautvydas.petkus@ktu.edu

² Kauno technologijos universitetas, Programų inžinerijos katedra, Studentų g. 50, Kaunas, Lietuva, karolis.ryselis@ktu.edu

Santrauka (abstract). Šiame straipsnyje apžvelgiama naujos „Kinect“ judesių aptikimo įrangos (2014m.) galimybės atpažįstant kompleksines žmogaus pozas (atsigulimas, pasilenkimas ir t.t.). Įvertinamos senesnės ir naujesnės versijos galimybių skirtumai, apskaičiuojamas jutiklio interpretatoriaus tikslumas pasitelkiant atstumo tarp sąnarių apskaičiavimą pagal kalibracijos duomenis. Tyrime siekiama sužinoti, kokio tipo pozicijas jutiklis lengviausiai atpažįsta, o kokių interpretavime klysta labiausiai.

Raktiniai žodžiai: Kinect 2.0 jutiklis, žmogaus skeleto interpretavimas.

Ižanga

2010 m. Microsoft vartotojų rinkai pristatė Microsoft „Kinect“ jutiklį, skirtą žmogaus kūno, gestų ir judesių atpažinimui. Įrenginys buvo plačiai taikomas žaidimams, inžineriniams sprendimams bei inovacijoms reklamos, paslaugų sektoriuose. Deja, „Kinect“ jutiklis nepelnė tokio didelio populiarumo dėl jo ribotų galimybių atpažįstant žmogaus kūną esant dinamiškai aplinkai bei interpretatoriaus nepakankamo pajėgumo tiksliai atpažįstant nematomus kamerasi žmogaus sąnarius.

2013m., kartu su Microsoft Xbox One žaidimų konsole, buvo išleistas naujos versijos „Kinect“ jutiklis pavadinimu „Kinect 2.0“, praplečiantis senosios versijos galimybes. Netrukus 2014 m. buvo išleista įrenginio versija, skirta asmeniniams kompiuteriams. Palyginant su „Kinect 1“, naujasis jutiklis yra pranašesnis šiais aspektais [1]: didesnė kameros raiška (1920x1080 spalvinė ir 512x424 gylis), platesnis matymo kampas (70° horizontaleje ir 60° vertikaleje), 26 skirtingų žmogaus sąnarių atpažinimas, iki 6 žmonių atpažinimas vienu metu.

Atsižvelgiant į įrenginio technines galimybes, „Kinect 2“ turi geriau, tiksliau atpažinti žmogaus kūną nei pirmoji jo versija. Jutiklinės sistemos reikalauja galingų interpretatorių, sugebančių atpažinti sąnarius su kuo mažesne klaidos tikimybe, norint įrenginį pritaikyti platesnėse didesnio tikslumo reikalaujančiose sferose kaip sportas, mokslas ir kt. Tyrime bus remiamasi Jogos bei bendros fizinės kultūros užsiėmimais, kurie reikalauja ypač įvairaus tipo žmogaus judesių ir pozų. Pagal šiuos reikalavimus bus iširta, koks yra „Kinect 2“ jutiklinės sistemos taikomumas šiai sričiai, reikalaujančiai ypač didelio kūno atpažinimo tikslumo.

Susiję moksliniai tyrimai analizuoja naujojo „Kinect 2“ jutiklio galimybes panaudojant įrenginį inovatyviais būdais. Hermann Fürntratt, Helmut Neuschmied iš Informacinių ir komunikacijos technologijų instituto (Gracas, Austrija) ištyrė, kokios yra galimybės panaudoti naująjį jutiklį kaip valdymo pultelį televizoriui ar kitam elektroniniam prietaisui [2]. Tyrime apibendrinta, jog „Kinect 2“ įrenginys daug geriau atpažįsta žmogaus judesius, ypač rankų gestus, nei ankstesnioji versija. Taip pat, geriausiems rezultatams pasiekti, rekomenduojama jutiklį laikyti apytiksliai grindų lygmenyje.

Université libre de Bruxelles (ULB) tyrėjai atliko „Kinect 1“ jutiklio analizę siekiant sudaryti tikroviškus trimačius žmogaus modelius pagal suteiktą „Kinect“ įrenginio informaciją. Aprašytame tyrime [3] buvo sukurtas įrankis, leidžiantis apytiksliai sudaryti trimatį taškinį žmogaus modelį pagal atpažintus žmogaus skeleto taškus bei realų žmogaus skeleto modelį, interpretuojant likusių žmogaus kaulų vietas ir būseną pagal biologinę struktūrą. Tyrimui buvo panaudotas MBA [4] algoritmas, sukurtas chirurginiais tikslais tiksliai sumodeliuoti žmogaus kūno dalis.

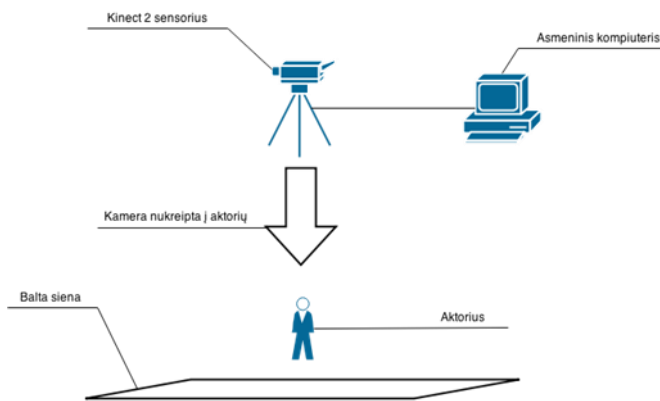
Tyrimo būdas

Šiame tyrime buvo siekta nustatyti įvairių sudėtingų pozicijų atpažinimo tikslumą naudojant „Kinect 2.0“ jutiklį. Tikslumo kriterijumi pasirinkome atstumą tarp sąnarių atitikimą, nes atstumas tarp tam tikrų žmogaus taškų, pavyzdžiui, alkūnės ir riešo, nesikeičia, kadangi tarp jų yra fiksuoto ilgio kaulas. Tyrimui buvo pasirinktos tokios sąnarių poros[5]: galva - kaklas, kaklas - stuburas tarp pečių, stuburas tarp pečių - dešinysis petys, dešinysis petys - dešinioji alkūnė, dešinioji alkūnė - dešinysis riešas, dešinysis riešas - dešinioji plaštaka, stuburas tarp pečių

* Išnaša

- kairysis petys, kairysis petys - kairioji alkūnė, kairioji alkūnė - kairysis riešas, kairysis riešas - kairioji plaštaka, stuburas tarp pečių - stuburas pilvo lygmenyje, stuburas pilvo lygmenyje - klubo centras, klubo centras - klubo dešinė, klubo centras - klubo kairė, klubo dešinė - dešinysis kelis, dešinysis kelis - dešinioji čiurna, dešinioji čiurna - dešinioji pėda, Klubo kairė - kairysis kelis, Kairysis kelis - kairioji čiurna, kairioji čiurna - kairioji pėda

Tyrimui nebuvo naudojamos tokios sąnarių poros, tarp kurių yra kitų sąnarių, nes tarp jų skirtingose pozicijose bus skirtingi atstumai. Taip pat nebuvo naudojamas pirštų atpažinimas, nes atstumai tarp pirštų gali keistis. Tariant padėtis buvo užtikrinta, kad kiekvienoje pozicijoje visas aktoriaus kūnas būtų „Kinect“ įrenginio matomumo lauke, tačiau dėl pozicijos sudėtingumo dalis aktoriaus kūno yra nematoma įrenginiui tiesiogiai, todėl buvo tiriama ir tai, kuri dalis sąnarių „Kinect“ jutikliui buvo matoma tiesiogiai kiekvienoje tiriamoje pozicijoje. Principinė tyrimo schema parodyta 1 pav.



1 pav. Paveikslėlio pavidimianis

„Kinect“ jutiklis prijungtas prie asmeninio kompiuterio ir pozicionuotas taip, kad jo matomame lauke būtų balta siena ir prie jos stovintis aktorius. Buvo pasistengta pašalinti visus nereikalingus objektus iš „Kinect“ jutiklio matymo lauko.

Tyrimui buvo naudojama tokia aparatinė įranga:

- „Kinect 2.0“ jutiklis
- Asmeninis kompiuteris su USB 3.0 valdikliu ir USB 3.0 prievadu

Taip pat buvo naudojama ši programinė įranga:

- Kinect Studio 2.0 programa
- Tyrimui skirta Windows RT tipo programėlė

Tyrimui skirta programėlė atlieka tokias pagrindines funkcijas:

- Išsaugo žmogaus, stovinčio priešais „Kinect“ jutiklį, sąnarių padėtis
- Leidžia įkelti žmogaus sąnarių padėtis iš išsaugoto failo
- Leidžia palyginti dvi padėtis
- Sugeneruoja palyginimų suvestinę laiko intervale

Programėlė realiu laiku analizuoja iš „Kinect“ jutiklio ateinančius duomenis ir leidžia juos iškart interpretuoti bei matuoti nuokrypius nuo bazinės padėties.

Tyrimo rezultatų tikslumui įtaką daro šie faktoriai[6]:

- Jutiklio netobulumas
- Matavimo aplinka
- Tiriamo objekto paviršius
- Atstumas iki matuojamo objekto [7]

Kadangi tyrime analizuojamas jutiklio netobulumas, kiti faktoriai buvo maksimaliai eliminuoti: parinktas tolygus apšvietimas, paskirstantis šviesos srautą ant aktoriaus; tyrime fonui buvo naudojama balta siena, aktorius dėvėjo tik apatinės kūno dalies aprangą. Taip pat viso tyrimo metu buvo stengtasi išlaikyti tą patį atstumą iki „Kinect“ sensoriaus, kad tarp pozicijų būtų mažesnis paklaidų išsibarstymas, tačiau kai kurių pozicijų prigimtis neleidžia to užtikrinti, nes aktorius turi būti išsitiesęs erdveje visu savo ilgiu. Šių paklaidų įvertinimui buvo pridėtos ir dvi paprastos pozicijos, kurios turėtų būti atpažintos gana tiksliai ir parodytų tik tas paklaidas, kurias siekta eliminuoti.

Tyrimo algoritmas

Tyrimui buvo naudojamas vektorių ilgiais grįstas algoritmas. Tarkime, kad turime du R3 erdvės taškus $A(x_A, y_A, z_A)$ ir $B(x_B, y_B, z_B)$. Sudarykime vektorių (\overrightarrow{AB}) . Tarkime, kad atstumas tarp taškų yra fiksuotas ir lygus

$$R = |\overrightarrow{AB}| = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2} \quad (1)$$

Tarkime, kad galime matuoti taškų koordinates su tam tikra paklaida. Tuomet tokių apytiksliai išmatuotų taškų A ir B koordinatės $A'(x_A + \Delta x_A, y_A + \Delta y_A, z_A + \Delta z_A)$ ir $B'(x_B + \Delta x_B, y_B + \Delta y_B, z_B + \Delta z_B)$. Tuomet apytikslis atstumas tarp taškų

$$R' = |\overline{A'B'}| = \sqrt{\left((x_A + \Delta x_A) - (x_B + \Delta x_B)\right)^2 + \left((y_A + \Delta y_A) - (y_B + \Delta y_B)\right)^2 + \left((z_A + \Delta z_A) - (z_B + \Delta z_B)\right)^2} \quad (2)$$

Santykinę paklaidą galime rasti:






$$\delta = \frac{|R - R'|}{R} = \frac{\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2} - \sqrt{\left((x_A + \Delta x_A) - (x_B + \Delta x_B)\right)^2 + \left((y_A + \Delta y_A) - (y_B + \Delta y_B)\right)^2 + \left((z_A + \Delta z_A) - (z_B + \Delta z_B)\right)^2}}{\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}} = 1 - \frac{\sqrt{\left((x_A + \Delta x_A) - (x_B + \Delta x_B)\right)^2 + \left((y_A + \Delta y_A) - (y_B + \Delta y_B)\right)^2 + \left((z_A + \Delta z_A) - (z_B + \Delta z_B)\right)^2}}{\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}} \quad (3)$$






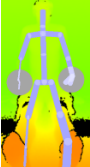


Idealiu atveju, kai visos absoliutinės paklaidos lygios 0, pošaknio skaitiklio reikšmė tampa lygi 0 ir santykinė paklaida taip pat tampa lygi nuliui. Realiu atveju visų trijų paklaidų dydžiai bus nenuliniai ir santykinės paklaidos dydis priklausys nuo visų paklaidų. Tyrime eksperimentiškai rasime paklaidas taikydami (3) formulę. Siekiant sumažinti reikšmių fluktuacijų įtaką skaičiavimams buvo naudojama 30 kadru suvidurkinti duomenys. Lyginimui buvo naudojamas su „Kinect“ jutikliu užfiksuotas gerai matomas pavyzdinis skeletas.

Tyrimo rezultatai

1 lentelėje parodytos visos tirtos padėtyys. Pateikta infraraudonųjų spindulių nuotrauka, išskirta geriausiai ir blogiausiai nustatytos atkarpos, vidutinė paklaida nuo pavyzdinio skeleto bei žmogaus skeleto skaitinis įvertinimas ir komentarai, kas buvo atpažinta netiksliai. Pažymėtina, kad vienos pozicijos tyrimas buvo vykdytas apie 20 sekundžių ir per tą laiką aktorius nejudėjo, tačiau daugeliu atvejų „Kinect“ jutiklio atpažįstamas skeletas šokinėdavo, todėl nuotrauka nusako tik vieno laiko momento skeleto atpažinimo tikslumą, o skaitiniai įverčiai taikomi visam matavimo intervalui. Palyginimui įtrauktos ir dvi standartinės pozicijos – stovėjimas priekiu ir stovėjimas šonu.

Lentelė Nr.1 Žmogaus pozų tyrimo rezultatai

Pozicijos nuotrauka	Pozicijos pavadinimas / Vidutinė paklaida	Geriausiai atpažinta atkarpa ir paklaida	Blogiausiai atpažinta atkarpa ir paklaida	Žmogaus skeleto matomumas
	Stovėjimas priekiu 10%	Dešinysis riešas - dešinioji plaštaka (0,4%)	Kairysis riešas - kairioji plaštaka (36%)	100% Klaidingai atpažinta žmogaus kairioji ranka, nepilnai atpažinta dešinioji koja
	Stovėjimas šonu 12%	Stuburas tarp pečių - dešinysis petys (0,1%)	Klubų centras - kairysis klubas (54%)	72% Klaidingai atpažintos žmogaus plaštakos, neproporciškos. Neteisingai atpažintas dešinysis petys.
	Atsitūpimas apglėbiant kojas 15%	Dešinioji čiurna - dešinioji pėda (1,5%)	Dešinysis riešas - dešinioji plaštaka (44%)	68% Klaidingai atpažintos kojos, iškraipytos aktoriaus pirštų proporcijos
	Kojos pirštų siekimas stovint 35%	Dešinysis klubas - dešinysis kelis (10,4%)	Stuburas tarp pečių - kairysis petys (102%)	76% Aktoriaus rankos nebuvo atpažintos reikiamoje padėtyje (jos buvo ties pėdomis). Aktorius buvo pasilenkęs - klaidingai buvo atpažinta galva ir stuburas
	Gulėjimas pilvu į žemę 17%	Dešinysis klubas - dešinysis kelis (1,4%)	Kairysis riešas - kairioji plaštaka (70%)	68% Kūno dalių proporcijos išlaikytos, tačiau jų pozicionavimas iškreiptas.

Pozicijos nuotrauka	Pozicijos pavadinimas / Vidutinė paklaida	Geriausiai atpažinta atkarpa ir paklaida	Blogiausiai atpažinta atkarpa ir paklaida	Žmogaus skeleto matomumas
	Stovėjimas aukštyn kojomis 24%	Kairysis klubas - kairysis kelis (1,0%)	Kaklas - stuburas tarp pečių (71%)	Žmogaus kūnas atpažintas atvirkščiai. Interpretuota, jog aktorius stovi ant kojų
	Kregždutės poza 21%	Kaklas - stuburas tarp pečių (0,4%)	Kairysis riešas - kairioji plaštaka (133%)	88% Klaidingai atpažintos plaštakos, neproporciškos
	Kojų iškėlimas į viršų 31%	Dešinysis kelis - dešinioji čiurna (0,7%)	Stuburas tarp pečių - dešinysis petys (106%)	64% Skeletas visiškai iškreiptas - rankos atpažintos aktorius kojose, klaidingai interpretuotos likusios kūno dalys
	Galūnių tempimas stovint išlenkiant krūtinę 32%	Stuburas tarp pečių - dešinysis petys (1,4%)	Dešinysis riešas - dešinioji plaštaka (62%)	88% Aktorius, būdamas nestandartinių, didelių proporcijų, nebuvo iki galo atpažintas. Kojos atpažintos tik iki pusės, pečių lygmuo atpažintas pernelyg aukštai.
	Pasiruošimo bėgimui poza 25%	Kaklas - stuburas tarp pečių (2,9%)	Dešinysis riešas - dešinioji plaštaka (63%)	80% Klaidingai atpažintos žmogaus kojos, netikslumai stuburo ir kaklo formoje
	Atsisėdimas ant žemės, ištiesus kojas į priekį 18%	Stuburas tarp pečių - dešinysis petys (1,4%)	Kairysis riešas - kairioji plaštaka (53%)	96% Klaidingai atpažintos žemiau pečių esančios kūno dalys (taškai neteisingose pozicijose)
	Kūno šono tempimas 20%	Dešinysis klubas - dešinysis kelis (0,1%)	Stuburas tarp pečių - kairysis petys (113%)	96% Didžioji kūno taškų dalis atpažinta neteisingai, neteisingose pozicijose.
	Nugaros tempimas atsigulus 21%	Stuburas tarp pečių - kairysis petys (1,1%)	Kairysis kelis - kairioji čiurna (51%)	80%

2 lentelėje parodyti per visas pozicijas agreguoti duomenys, parodantys, kiek kiekviena atkarpa vidutiniškai netiksliai buvo atpažinta. Pastebime, kad ilgesnių kūno dalių atpažinimas buvo tikslesnis, o su trumpesnėmis „Kinect“ jutiklis susidorojo sunkiau.

Lentelė Nr.2 Atkarpų tarp kūno dalių atpažinimo tikslumo rezultatai

Atkarpa	Vidutinė paklaida
Dešinysis klubas - dešinysis kelis	10,4%
Stuburas tarp pečių - dešinysis petys	13,7%
Dešinysis petys - dešinioji alkūnė	15,2%
Kairysis klubas - kairysis kelis	15,3%
Dešinioji alkūnė - dešinysis riešas	16,0%
Kaklas - stuburas tarp pečių	17,2%
Dešinioji čiurna - dešinioji pėda	18,2%

Atkarpa	Vidutinė paklaida
Kairysis petys - kairioji alkūnė	18,6%
Stuburas tarp pečių - stuburas pilvo lygmenyje	18,8%
Galva - kaklas	20,1%
Stuburas pilvo lygmenyje - klubų centras	20,4%
Kairysis kelis - kairioji čiurna	21,4%
Klubų centras - kairysis klubas	22,0%

Išvados

Atlikus sudėtingų žmogaus pozicijų analizę naudojant „Kinect 2“ jutiklį nustatyta, jog įrenginys ypač prastai atpažįsta žmogaus kūną nestandartinėse būsenose - tupint, sėdint, gulint ir t.t. Žmogaus atkarpos tarp sąnarių, kurių fiziškai įrenginys nematė, buvo interpretuojamos netiksliai, dydžiai pateikiami neproporcingi, gautos paklaidos gerokai viršija standartinėse padėtyse užfiksuotas paklaidas. Iš tyrimo matome, jog „Kinect“ jutiklis geriau ir tiksliau atpažįsta ilgas žmogaus kūno dalis, pvz., kojas, ilguosius rankų kaulus, tačiau ne taip tiksliai atpažįsta trumpas kūno dalis, pvz., delnus.

Įvertinus visus ištirtus atvejus galima teigti, jog „Kinect 2“, nepaisant techninių pranašumų nuo pirmosios įrenginio versijos, vis tiek dar nėra tinkamas įrenginys didelio tikslumo reikalaujančiose nišose - jogoje, sporte, medicinoje. Jutiklis ypač prastai atpažįsta žmogų, jam esant kitokioje pozicijoje nei stovėjimas. Norint jutiklines sistemas pritaikyti vartotojų rinkoje, reikia geresnės technologijos nei šiuo metu esantis įrenginys Microsoft „Kinect“.

Literatūros sąrašas

- [1] **Ashley J.** Quick Reference: Kinect 1 vs Kinect 2. *Microsoft*, 2014.
- [2] **Fürntratt H., Neuschmied H.** Evaluating Pointing Accuracy on Kinect V2 Sensor. *Digital – Institute for Information and Communication Technologies*, 2014.
- [3] **Bonnechère B., Sholukha V., Moiseev F., Rooze M., Van Sint Jan S.** From Kinect™ to anatomically-correct motion modelling: Preliminary results for human application. *Laboratory of Anatomy, Biomechanics and Organogenesis (LABO), Université Libre de Bruxelles*, 2013.
- [4] **Siddhartha K. Shakya.** Probabilistic model building Genetic Algorithm (PMBGA): A survey. *School of computing, The Robert Gordon University, Aberdeen*, 2003.
- [5] JointType Enumeration of Microsoft Developer Network. Official documentation. *Microsoft*, 2015.
- [6] **Khoshelham K., Oude Elberink S.** Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Faculty of Geo-Information Science and Earth Observation, University of Twente*, 2012.
- [7] **A. Livingston M., Sebastian J., Ai Z., W. Decker J.** Performance Measurements for the Microsoft Kinect Skeleton. *Naval Research Laboratory*, 2012.

Nonstandard human body position tracking using „Kinect 2.0“ sensor accuracy research

This publication is an overview of new „Kinect“ motion sensor (2014) capabilities of recognising difficult human postures (laying, bending over etc.). A brief comparison of limits between older and newer version of hardware, evaluation of sensor interpreter accuracy using calculation of distance between joints, determined by calibration data. Research aims to determine, what type of positions does the sensor recognise with ease and which ones are error-prone.