



KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

PAULIUS STAŠKŪNAS

**„KINECT“ SENSORIUMI PAREMTŲ VALDYMO
SĄSAJŲ KŪRIMAS IR TYRIMAS**

Magistro baigiamasis darbas

Darbo vadovas

doc. dr. T. Blažauskas

KAUNAS, 2013



KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

PAULIUS STAŠKŪNAS

**„KINECT“ SENSORIUMI PAREMTŲ VALDYMO
SĄSAJŲ KŪRIMAS IR TYRIMAS**

Magistro baigiamasis darbas

Recenzentas

prof. dr. R. Damaševičius

Darbo vadovas

doc. dr. T. Blažauskas

Atliko

IFM-1/2gr. studentas

P. Staškūnas

KAUNAS, 2013

PATVIRTINIMAS APIE ATLIKTO DARBO SAVARANKIŠKUMĄ

Patvirtinu, kad įteikiamas baigiamasis darbas „Gestų valdymo sąsajų kūrimas ir tyrimas“:

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

Paulius Staškūnas

2013-05-20

(studento vardas, pavardė)

(data)

(parašas)

SANTRAUKA

Šiame dokumente aprašytas darbas susideda iš keturių pagrindinių dalių. Pirmoji dalis – analizė. Šioje dalyje apžvelgiamos „Kinect“ jutiklio paremtos valdymo sąsajos kūrimo principai, nagrinėjama pati technologija, siekiant detaliau susipažinti su jos galimybėmis. Papildomai analizuojami egzistuojantys sprendimai.

Antrojoje (projektinėje) dalyje pagrindžiamas pats projekto kūrimas, aprašoma suformuluota projekto paskirtis ir panaudos atvejai. Šiame skyriuje aprašoma visa sistemos architektūra, reikalavimai sistemai, komponentai, statinis ir dinaminis vaizdai.

Trečioje (tyrimo ir eksperimentinėje) dalyje, detalizuojami mūsų atlikti tyrimai ir eksperimentai. Šių tyrimų metu siekiama ištirti pasiūlytą ir realizuotą automatinio gestų aptikimo algoritmą, atliekant eksperimentus ir identifikuojant problemas. Papildomai atlikta sistemos išeities kodo analizė, siekiant įvertinti sistemos kokybę ir nustatyti vietas, kurioms turėtų būti atliekami priežiūros darbai.

SUMMARY

„KINECT“ SENSOR BASED GESTURE RECOGNITION INTERFACE: DEVELOPEMENT AND RESEARCH

The work, described in this document, consists out of three main parts. The first part is analysis where we analyze the basic principles of a “Kinect” based user interface, get a better understanding of the technology itself, to get a better feel of its possibilities. An additional analysis is provided of existing products.

The second part is all about the implementation. This part describes various aspects of the project: we define the goal of the project, including various architectural means to achieve it. These means are described by using UML language models, including static, dynamic views.

The last part describes the research that was done during this project. Various researches and experiments were done in order to create an algorithm that can detect gestures with high precision. During the research phase we try to determine the problems that prevent a gesture to be recognized and propose solutions to make it better. Additionally a research was done with the created software itself, in order to determine its quality and find ways to increase it.

TURINYS

1. ĮVADAS.....	9
1.1. Dokumento paskirtis	9
2. ANALITINĖ DALIS.....	9
2.1. Projekto tikslas ir adresatas.....	9
2.2. Technologija	10
2.2.1. Įrenginys	10
2.2.2. Programinė įranga	11
2.3. Egzistuojantys sprendimai	12
2.3.1. Pagamintos sistemos, kurios gali būti nupirktos	12
2.4. Situacijos Lietuvoje įvertinimas	13
2.5. „Kinect“ sensoriumi paremtų valdymo sąsajų projektavimo principai	13
2.5.1. Sistemos nepastebimumas.....	13
2.5.2. Pritaikomumas.....	13
2.5.3. Naudojimosi paprastumas.....	14
2.6. Automatinio atpažinimo algoritmo analizė	14
2.7. Analitinės dalies rezultatai	15
3. PROJEKTINĖ DALIS	15
3.1. Projekto kūrimo pagrindimas	15
3.2. Sistemos paskirtis ir panaudos atvejai.....	16
3.3. Architektūros tikslai ir apribojimai	17
3.4. Reikalavimai sistemai	18
3.5. Sistemos architektūra.....	20
3.5.1. Sistemos statinis vaizdas	20
3.5.1.1. Komponentas „UserInterface“	20
3.5.1.2. Komponentas „KinectCon“	21
3.5.2. Komponentas „UserInterface “	22
3.5.3. Komponentas „KinectCon“	23
3.5.4. Sistemos dinaminis vaizdas	25
3.6. Projektavimo etapo rezultatai	26
4. TYRIMO IR EKSPERIMENTINĖ DALIS	27
4.1. Tyrimų eksperimentuose naudojamos sistemos parametrai.....	27
4.2. Automatinio atpažinimo algoritmas.....	28
4.3. Analizuojamų duomenų kiekio mažinimo įtaka gesto atpažinimo spartai.....	29
4.4. Matavimo paklaidų įvertinimas	31

4.4.1.	Atlikti matavimai.....	32
4.4.1.1.	Matavimai, esant geram apšviestumui	33
4.4.1.2.	Matavimai, esant prastam apšviestumui	36
4.5.	Neatpažintų gestų kiekio mažinimas panaudojant spartaus aptikimo algoritimą	39
4.6.	Sistemos išeities kodo analizė	43
4.6.1.	Kodo metrikų rezultatai.....	45
4.6.2.	Sistemos korektiškumo įvertinimas.....	47
5.	REZULTATŲ APIBENDRINIMAS IR IŠVADOS	47
6.	LITERATŪRA	49

LENTELIŲ SĄRAŠAS

Lentelė 1 1 m atstumas, geras apšviestumas.....	33
Lentelė 2 1.8 m atstumas, geras apšviestumas.....	33
Lentelė 3 2,4m. atstumas, geras apšviestumas.....	34
Lentelė 4 3 m atstumas, geras apšviestumas.....	35
Lentelė 5 1 m atstumas, tamsa.....	36
Lentelė 6 1.8 m atstumas, tamsa	36
Lentelė 7 2,4 m atstumas, tamsa.....	37
Lentelė 8 3m. atstumas, tamsa	37
Lentelė 9 Nenuskaitytų gestų eksperimento aplinkos sąlygos.....	39
Lentelė 10 Nenuskaitytų gestų eksperimento gesto koordinatės.....	39
Lentelė 11 Įvesto gesto, taško nr 2, koordinatinių minimalios ir maksimalios reikšmės.....	43
Lentelė 12 Ciklomatinių sudėtingumo reikšmės vertinimas:	44
Lentelė 13 Prižiūrimumo indekso reikšmės vertinimas:.....	44
Lentelė 14 Svarbiausi sistemos įvertinimo kriterijai.....	45
Lentelė 15 klasių metrikos.....	45
Lentelė 16 metodų, turinčių žemesnį nei 65 prižiūrimumo indeksą (MI) metrikos	46
Lentelė 17 Sistemos korektiškumo įvertinimas.....	47

PAVEIKSLŲ SĄRAŠAS

Paveikslas 1 Žmogaus judesių sekimas trimatėje erdvėje.....	10
Paveikslas 2 "Microsoft Kinect" sudedamosios dalys.....	10
Paveikslas 3 Sistemos panaudos atvejų vaizdas.....	17
Paveikslas 4 Sistemos suskaidymas į komponentus.....	20
Paveikslas 5 Komponento „UserInterface“ klasių diagrama.....	22
Paveikslas 6 Komponento „KinectCon“ klasių diagrama.....	23
Paveikslas 7 Gesto kūrimo ir funkcijos priskyrimo bendradarbiavimo diagrama.....	25
Paveikslas 8 Naujo gesto įvedimo sekų diagrama.....	26
Paveikslas 9 Nuskaitomų duomenų apdorojimo algoritmo sekų diagrama.....	28
Paveikslas 10 Reakcijos laiko priklausomybė nuo įvestų gestų kiekio.....	30
Paveikslas 11 Optimizuotos sistemos reakcijos laiko priklausomybė nuo įvestų gestų kiekio.....	31
Paveikslas 12 Matavimų schema.....	31
Paveikslas 13 Vidutinio koordinatės skirtumo kitimas, priklausomai nuo atstumo iki jutiklio (šviesoje).....	35
Paveikslas 14 Vidutinio koordinatės skirtumo kitimas, priklausomai nuo atstumo iki jutiklio (tamsoje).....	38
Paveikslas 15 Įvesto gesto koordinatės.....	40
Paveikslas 16 Sėkmingai ir nesėkmingai nuskaityto gesto X, Y koordinatės pasiskirstymo diagrama.....	41
Paveikslas 17 Spartaus judesio atpažinimo algoritmas.....	42

TERMINŲ IR SANTRUMPŲ ŽODYNAS

Gestas – naudotojo rankomis atliekama gestikuliacija.

Funkcija – veiksmas, atliekamas parodytus tam tikrą gestą (pvz. nurodytos programos paleidimas).

Naudotojas – žmogus, naudojantis sistemą, kuriantis individualius gestus, priskiriantis jiems funkcijas.

UML – Unified Modeling language (unifikuota modeliavimo kalba)

CMOS – Aktyvus pikselių jutiklis, panaudojantis papildomą metalo-oksido puslaidininkį (Complementary metal–oxide–semiconductor)

RGB - spalvų maišymo sistema, kurioje naudojamos trys, žmogaus akių receptorių atitinkančios spalvos: raudona (Red), žalia (Green) ir mėlyna (Blue).

SDK – Programinės įrangos kūrimo įrankiai (Software development kit)

VGA – 640 x 480 pikselių skiriamoji geba.

XBMC – media centro programinė įranga.

XML - Išplečiama žymų kalba (Extensible markup language).

TCP/IP - Tinklo protokolas, duomenų perdavimui.

WPF – programinė sąsaja (API), leidžianti valdyti XAML dokumentų objektus. (Windows Presentation Foundation)

1. ĮVADAS

1.1. Dokumento paskirtis

Šis dokumentas yra programų sistemų inžinerijos magistro baigiamasis darbas. Dokumente yra trumpai apžvelgiamos problemos su kuriomis yra susiduriama projektuojant programinę įrangą, skirtą naudotojo gestų nuskaitymui, taip pat analizuojami būdai, kaip pagerinti nuskaitymų gestų tikslumą.

Remiantis aptartais metodais, buvo suprojektuota ir realizuota programinė įranga. Reikalavimų rinkimo bei architektūros sudarymo etapai yra aprašomi šio darbo projektinėje dalyje. Joje detalizuojami projektavimo elementai, į kuriuos turėtų būti kreipiamas didelis dėmesys, siekiant kad programinė įranga būtų efektyvi. Tyrimo dalyje yra analizuojama sukurtos programinės įrangos kokybė, pateikiami ir realizuojami jos pakeitimai, siekiant pagerinti palaikomumą ir plečiamumą, taip pat analizuojamos „Kinect“ jutiklio paklaidos. Eksperimentinėje dalyje pateikiami pasiūlymai, sumažinti sistemos apkrovimą ir padidinti sėkmingai nuskaitytų gestų kiekį.

2. ANALITINĖ DALIS

2.1. Projekto tikslas ir adresatas

Suformuluota užduotis: sukurti programinę įrangą, išnaudojančią išorinio įrenginio suteikiamas galimybes asmeniniam kompiuteriui, šiuo atveju „Microsoft Kinect“ išorinį sensorių kuriant programinę įrangą, leisiančią atitinkamas funkcijas aktyvuoti gestų pagalba.

Projektas skirtas namų naudotojui, bet gali būti taikomas ir mokymo tikslais, supažindinant su šios technologijos galimybėmis, arba valdant kitus įrenginius, pvz., robotus.

Naudotojas galės naudotis tokiomis funkcijomis:

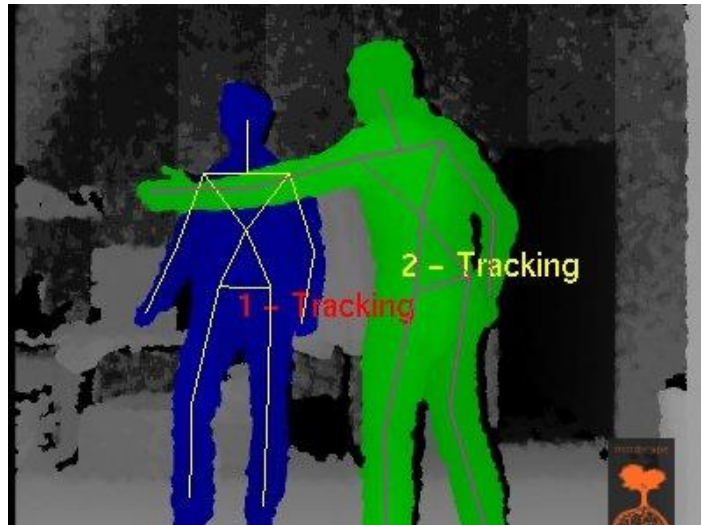
- Naujo gesto įvedimu į sistemą.
- Su gestu susietos funkcijos pridėjimu, redagavimu.
- Gesto parametrų redagavimu.
- Gesto šalinimu iš sistemos.
- Sistemos reagavimu į rodomą gestą ir atitinkamos funkcijos vykdymu.

Naudodamasis šia sistema naudotojas galės tam tikriam darbui priskirti gestus, siekiant sumažinti laiką, reikalingą atitinkamai funkcijai iškviešti. Taip pat sukurs interaktyvią valdymo

atmosferą, nereikalaujančią klaviatūros ar pelės, leidžiančią valdyti kitus išorinius įrenginius (robotas, televizorius t.t.).

2.2. Technologija

Vienas iš palyginti neseniai į rinką išleistų prietaisų, pilnai veikiančių pagal žmogaus judesius ir nereikalaujančių jokių priedų, išskyrus specialų išorinį įrenginį – kamerą, yra „Microsoft“ sukurtas įrenginys „Kinect“. Šis prietaisas užfiksuoja žmogaus judesius trimatėje erdvėje (Paveikslas nr. 1 „Žmogaus judesių sekimas trimatėje erdvėje“) [5], naudojantis „Light Coding“ technologija. Jis taip pat gali atpažinti balso komandas. Ši technologija yra sukurta

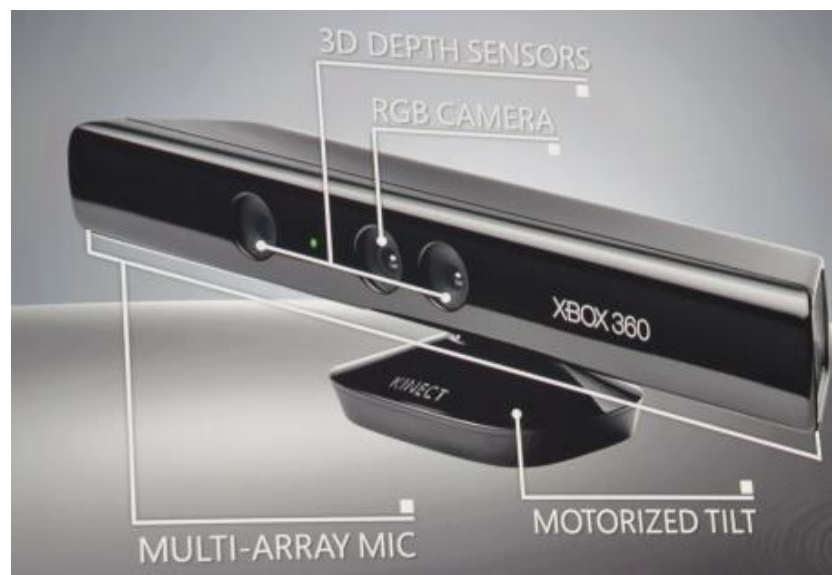


Paveikslas 1 Žmogaus judesių sekimas trimatėje erdvėje.

„Rare“ padalinio, priklausančio „Microsoft Game Studios“ įmonei. Technologija leidžia tiksliai nustatyti naudotojo poziciją erdvėje ir tiksliai reaguoti į atliekamus judesius, taip sukuriant valdymo aplinką, kuri yra natūrali žmogui, t.y. judesiai ir balso komandos.

2.2.1. Įrenginys

„Microsoft Kinect“ įrenginys susideda iš infraraudonųjų spindulių projektoriaus, RGB kameros, gylio sensoriaus, motorizuoto kameros pavertimo, mikrofonų ir specialios mikroschemos [6], leidžiančios sekti objektų ir atskirų individų judesius trimatėje erdvėje ir atpažinti balso komandas (Paveikslas nr. 2 „Microsoft Kinect sudedamosios dalys“). Papildomai „Kinect“ jutiklio mikrofoni leidžia tiksliai nustatyti iš kur sklinda garsas, jį lokalizuoti, taip sumažinant pašalinių garsų įtaką atpažinimui.



Paveikslas 2 "Microsoft Kinect" sudedamosios dalys

Gylio jutiklis susideda iš infraraudonųjų spindulių projektorius, sujungto su monochrominės šviesos CMOS sensorium, kas leidžia fiksuoti vaizdus esant menkam apšvietimui. Gylio jutiklio reagavimo atstumas yra reguliuojamas, ši funkcija yra automatiškai realizuojama „Kinect“ programinės įrangos. Tai leidžia jutikliui prisitaikyti prie naudotojo aplinkos. [7]

Pati technologija vienu metu leidžia sekti iki šešių asmenų judesius vienu metu, o realiai reaguoti į dviejų asmenų judesius (judesių analizė iki 20 sąnarių, asmeniui). Realiose situacijose ši technologija yra labiau apribota ne naudotojų kiekiu, o kiek jų gali tilpti į kameros matomą plotą. [8]

„Kinect“ jutiklio perduodamas vaizdo srautas yra 30 kadru per sekundę. RGB vaizdo srautas naudoja 8 bitų VGA raišką (640x480 pikseliai) su „Bayer“ spalvų filtru. Monochrominio gylio jutiklio vaizdo srautas yra taip pat VGA raiškos (640x480 pikseliai), bet naudoja 11 bitų gylį, kas suteikia 2048 jautrumo lygius. „Kinect“ jutiklio aktyvus veikimo nuotolis svyruoja tarp 1-3,5 m. Sumontuoti lęšiai suteikia jutikliui 57° horizontalaus ir 43° vertikalaus matymo plotą, o motorizuotas pavertimo mechanizmas papildomai leidžia jutiklį paversti iki 27° į viršų ar apačią.

2.2.2. Programinė įranga

Specialiai „Xbox 360“ sukurta programinė įranga reikalauja 190 MB laisvos vietos. Ši programinė įranga leidžia „Xbox 360“ sistemos naudotojams valdyti konsolės naudotojo sąsają balsu ar rankų judesiais. Taip pat „Microsoft“ sukurta programinė įranga „Video Kinect“ leidžia sistemai atpažinti naudotojus pagal jų balsus ar veidus, užmegzti vaizdo pokalbius su kitais „Xbox 360“ arba „Windows Live Messenger“ naudotojais, motorizuoto sukimosi taško funkcija naudotoją visada laiko kadre, net jam judant.

2011 m. vasario 21 d. „Microsoft“ paskelbė, kad išleis nekomercinę „Kinect“ SDK „Windows“ operacinei sistemai. Šis programinės įrangos kūrimo paketas buvo išleistas 2011 m. birželio 16 d. Vėlesniuose planuose ruošiamasi išleisti komercinei veiklai skirtą paketą. Ne komercinis paketas susideda iš „Windows 7“ operacinei sistemai pritaikytų tvarkyklių, leidžiančių kurti programinę įrangą naudojančią C++, C# ar „Visual Basic“ programavimo kalbomis su „Microsoft Visual Studio 2010“ programa. [9] Ne komerciniame pakete prieinamos funkcijos:

- Jutiklių srautai: gylio jutiklio, kameros, mikrofonų srautas.
- Skeleto sekimas: iki dviejų asmenų skeletų sekimas, skirtas gestų atpažinimui.
- Pažangios balso atpažinimo galimybės: akustinio garso, aido slopinimas, garso šaltinio atpažinimas, „Windows“ balso atpažinimo programinės įrangos integracija.
- Pavyzdžiai ir dokumentacija.

2.3. Egzistuojantys sprendimai

2.3.1. Pagamintos sistemos, kurios gali būti nupirktos

Yra keletas komercinių sprendimų, įgyvendinančių įrenginių valdymo gestais idėją.

„GestureTek“ sukurta sistema leidžia valdyti įvairius įrenginius gestais. Sistema yra suderinama su dauguma rinkoje esančių erdvinio vaizdo kamerų. Kūrėjų teigimu jų sistema yra tiksli ir patikima.

Programinės įrangos teikiamos galimybės:

- Įrenginių valdymas rankų ar viso kūno judesiais.
- Suderinamumas su dauguma kompiuterinių sistemų.
- Tikslus naudotojo kūno dalių padėties ir trajektorijos nustatymas.
- Platus gestų pasirinkimas (vienos rankos, dviejų rankų gestai).
- Nereikalauja kalibravimo atskiriems naudotojams.
- Nereikalingos specialios pirštinės ar kiti, naudotojo dėvimi, priedai.

Didelė sprendimo kaina.

„Mgestyk“ sukurtas sprendimas taip pat leidžia naudotojui žaisti žaidimus ar naudotis kompiuterio funkcijom gestų pagalba. Išskirtinumas yra tas, kad ši sistema palaiko ir rankos pirštų judesių sekimą.

Programinės įrangos teikiamos galimybės:

- Geba atpažinti ne tik kūno dalių judesius, bet ir smulkius rankos judesius.
- Nėra apribojimų aplinkai (šviesumo lygis patalpoje).
- Nuskaitymų tikslumui neturi įtakos papildomi asmenys, esantys už naudotojo.
- Nereikalingos specialios pirštinės ar kiti, naudotojo dėvimi, priedai.

Vienetinis sprendimas, galima nupirkti tik specifikaciją.

2.4. Situacijos Lietuvoje įvertinimas

2011 metų duomenimis 63,5 % namų ūkių, esančių miestuose turi kompiuterius. Orientuojamasi į miestų gyventojus, nes reikalingo priedo kaina yra apie 400 lt ir jis būtų priskiriamas prabangos prekei. Duomenų, apie įrenginio pardavimus Lietuvoje nepavyko gauti, tad nėra tikslių skaičių, kiek potencialių klientų būtų, kurie jau turi judesius fiksuojančius įrenginius ir jiems reikalinga tik programinė įranga.

2.5. „Kinect“ sensoriumi paremtų valdymo sąsajų projektavimo principai

2.5.1. Sistemos nepastebimumas

Naudotojas turi beveik nejausti, kad jis naudoja papildomą sąsają. Neturėtų būti naudojamos specialios pirštinės, specialūs jutikliai, tvirtinami prie rūbų, ar kiti priedai, siejami su naudotoju, o ne su kompiuteriu. Sistemos naudojimuisi turėtų užtekti kasdienės aprangos. Papildomai bus siekiama išvengti reikalavimo paspausti tam tikrus mygtukus, apibrėžiančius gesto pradžių ir pabaigą. Šiuo tikslu bus bandoma panaudoti F. Hofmann sukurtą algoritmą [2], kurio pagalba bus galima išvengti šių reikalavimų mygtukų paspaudimams. Gestas bus automatiškai analizuojamas realiu laiku iš nuolatos gaunamo išorinių duomenų srauto. Vienintelis pastebėtas apribojimas yra trumpos pauzės tarp gestų.

Dar vienas svarbus faktorius yra reakcijos laikas, per kurį sistema reaguoja į naudotojo judesį. Remiantis literatūra, šis laikas neturėtų viršyti 100 ms., kad naudotojas **visai** nejaustų papildomos sąsajos. Jei ši reakcija sieks 1s., naudotojo minčių seka liks nepakitus, bet jis jaus uždelsimą. [4]. Kuriant sistemą bus siekiama išlaikyti reakcijos laikus tarp 0,1-1 s., remiantis šiais nurodymais.

2.5.2. Pritaikomumas

Kuriant gestų valdymo sistemą, turi būti priimtas sprendimas ar bus naudojami nustatyti gestai, ar naudotojas pats galės juos sukurti. Pirmu atveju kuriant sistemą, yra gana sudėtinga apmastyti visus atvejus, ypač dėl kultūrų, pomėgių skirtumų, taip pat ar naudotojas yra kairiarankis ar dešiniarankis. Taigi sistema turėtų turėti naudotojo nustatomus gestus, kuriuos jis pats susikurs pagal savo pomėgius.

Galimas variantas yra naudoti [3] siūlomą projektavimą, kuris paremtas optimalių gestų nustatymu tam tikriems veiksmams, bet tai yra per ilgas kūrimo ir projektavimo procesas, todėl mūsų sistema naudos prisitaikymo principus ir leis pačiam naudotojui nustatyti savo gestus ir jiems priskirti funkcijas.

2.5.3. Naudojimosi paprastumas

Sistemos funkcijos turėtų būti aiškios ir lengvai suprantamos naudotojui. Pati sistema yra orientuojama į įprastą namų naudotoją, tad nereikėtų kelti aukštų reikalavimų jo kompiuteriniams gabumams. Bet taip pat nereiktų pamiršti pažengusių naudotojų. Šiems naudotojams planuojama leisti patiems kurti funkcijas, kurios bus priskiriamos gestams, o ne rinktis iš atitinkamo sąrašo. Taip pat naudotojui turėtų būti rodomi pranešimai apie jo rodomą gestą, kadangi gestų atpažinimas niekada 100% nesutaps su įvestu gestu. Šie pateikiami pranešimai leis naudotojui aiškiai suprasti ar nereiks pakartoti gesto, ar viskas atlikta teisingai.

2.6. Automatinio atpažinimo algoritmo analizė

Šio skyriaus tikslas yra aprašyti bendrus principus, kuriais turėtų vadovautis automatinio gestų atpažinimo algoritmas. Algoritmas turi realizuoti šiuos veiksmus:

- Spartus nuskaitomų iš išorinio judesių jutiklio duomenų apdorojimas.
- Paklaidos (tolerancijos) realizacija: algoritmas turi įgyvendinti paklaidų sistemą, kuri pagal individualius gesto parametrus, reaguotų į gestus pagal nustatytą toleruotiną koordinačių svyravimo ribą.
- Turi palaikyti daugiau nei vieno gesto paiešką vienu metu. Turi būti realizuotas situacijos, kai sistemoje yra įvesti keli panašūs gestai, apdorojimas.

Algoritmas, sistemos darbo metu, analizuos didelį kiekį duomenų. Turės atlikti pradinius skaičiavimus prieš paleidimą, kurie suteiks kiekvienam įvestam gestui tolerancijos ribas (kiek kiekvieno gesto momento koordinatės gali neatitikti nuskaitytiems duomenims). Atlikus inicializacijos veiksmus, algoritmas turės apdoroti nuolatinį srautą duomenų ir atitinkamai tvarkyti kintamuosius, kad būtų vykdomas aptiktų sutampančių momentų sekimas. Jeigu pasiekiamas ir aptinkamas gesto paskutinis fiksuotas momentas (gestą sudaro neribotas kiekis momentų. Kiekvienas momentas turi ribotą kiekį koordinačių, kurios jį apibrėžia) – algoritmas turi vykdyti veiksmų valdiklio suaktyvinimą, kurio tikslas – perduoti informaciją apie tai, koks gestas buvo aptiktas.

2.7. Analitinės dalies rezultatai

1. Vykdamas analizės etapą, buvo aiškinamasi su kokiomis problemomis susiduriama, projektuojant interaktyvią programinę įrangą. Buvo suformuluotos gairės, kuriomis vadovaujantis reikėtų kurti interaktyvią gestų sąsają. Papildomai buvo gilinamasi į algoritmą, reikalingą automatiniam gesto atpažinimui įgyvendinti.
2. Šio etapo metu, didelis dėmesys buvo skiriamas egzistuojančių sprendimų analizei. Analizės metu, buvo surinkta informacija, apie realizuotų sprendimų teikiamas funkcijas.
3. Analizės metu nustatyta, kad įgyvendintų sprendimų, realizuojančių interaktyvų gestų įvedimą, yra vos keli. Įrankio, leidžiančio naudotojui pačiam kontroliuoti, kokius gestus jis nori įvesti į sistemą, kūrimas atveria dalinai neužpildytą nišą rinkoje.
4. Atlikta automatinio atpažinimo algoritmo analizė. Suformuluotos gairės, kuriomis remiantis gali būti realizuojamas atpažinimo algoritmas, veikiantis visiškai automatinio principu, t.y. nereikalinga fiksuoti rodomo gesto pradžios ir pabaigos.
5. Analitinės dalies metu buvo atliekama galimų išorinių jutiklių analizė. Buvo nustatyta, kad didžiausią paplitimą rinkoje turi „Microsoft Kinect“ įrenginys. Šio įrenginio veikimo principai išnagrinėti detaliau, siekiant geriau susipažinti su technologija.

3. PROJEK TINĖ DALIS

3.1. Projekto kūrimo pagrindimas

Nemažas kiekis parduotų „Microsoft Kinect“ įrenginių (2011 m. kovo mėn. duomenimis virš 10 mln. parduotų vienetų) sukuria didelį klientų kiekį, norinčių šį įrenginį naudoti ne vien žaidimams, o ir kasdieniniams darbams ar pomėgiams. Taip pat „Microsoft“ išleistos specialios bibliotekos, naudojančios „.Net“ programavimo technologijas, atvėrė duris naujoms valdymo sistemoms, skirtoms asmeninių kompiuterių naudotojams.

Ši kuriama gestų valdymo sąsaja papuola į didelės rizikos – didelio pelno sritį. Pagrindinės priežastys:

- nėra panašių projektų patirties;
- pačios technologijos yra naujos ir neįsisavintos;
- kaštų prognozavimas yra sudėtingas;

Pati sąsaja yra orientuota į namų naudotoją, asmeninių kompiuterių rinką, bet neatmetama galimybė pritaikyti programinę įrangą namų kino ar kitų buitinių prietaisų valdymui.

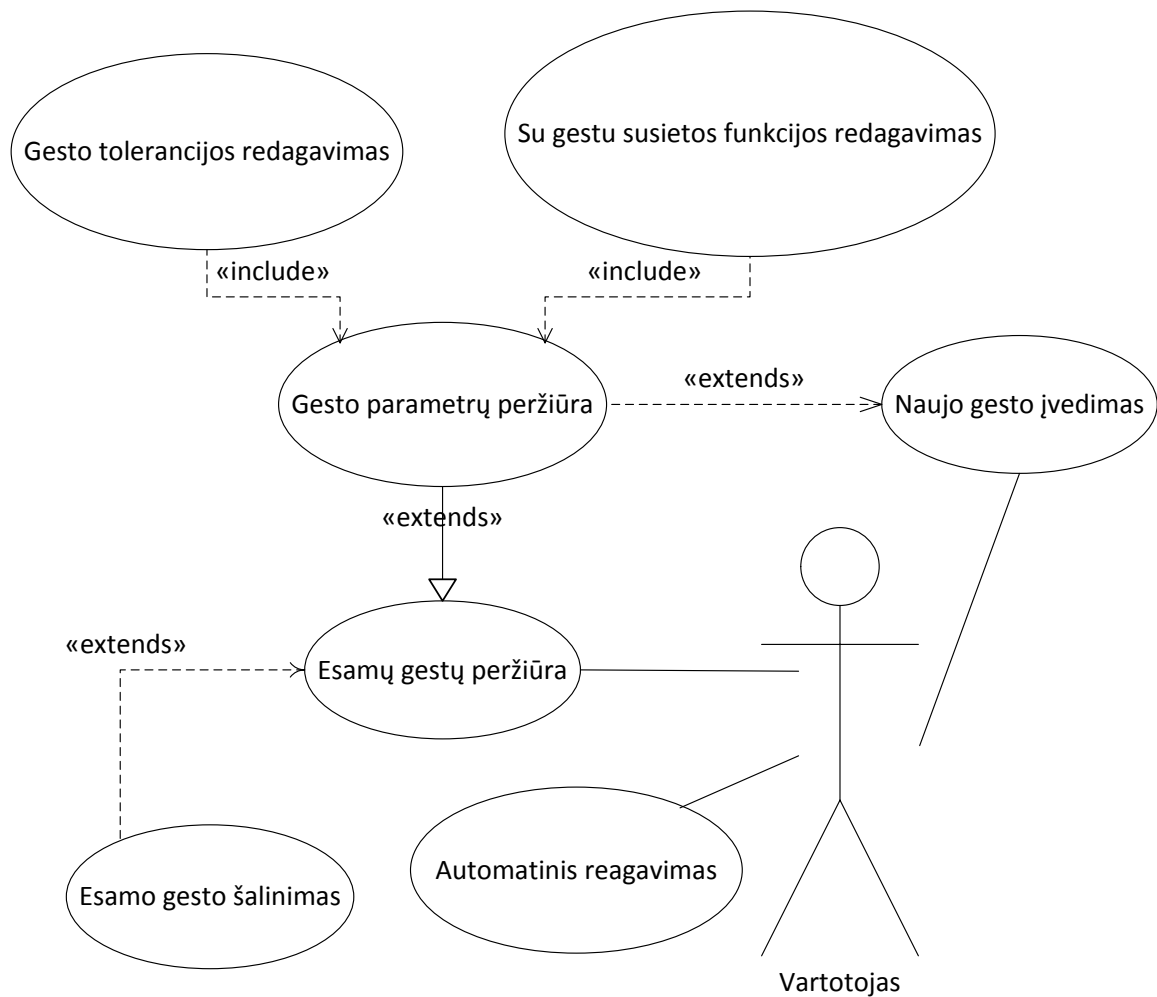
3.2. Sistemos paskirtis ir panaudos atvejai

Projekto tikslas yra sukurti gestais valdomą sąsają, leidžiančią naudotojui valdyti įvairias kompiuterio funkcijas gestais ar balso komandomis. Naudodamasis šia programine įranga naudotojas galės paspartinti kasdienių darbų atlikimą.

Sistema leis:

- Sukurti kiekvienam naudotojui individualius gestus, pagal kuriuos bus atliekamos atitinkamos operacijos.
- Keisti atliekamas operacijas, susietas su gestu.
- Sumažinti laiką, reikalingą iškviešti tam tikras funkcijas.

Paveiksle nr. 3 „Sistemos panaudos atvejų vaizdas“ pateikiama panaudos atvejų diagrama.



Paveikslas 3 Sistemos panaudos atvejų vaizdas

Darbo su sistema metu, naudotojas gali atlikti visus standartinius duomenų tvarkymo veiksmus, kaip: naujų duomenų įvedimas, redagavimas, šalinimas. Projektuojant sistemą, buvo priimtas sprendimas leisti naudotojui gestus kurti pačiam, o ne naudotis iš anksto sukurtu gestų rinkiniu, tad naujų gestų įvedimas buvo sukurtas kaip atskira sistemos dalis, leidžianti nuskaityti naudotojo judesius ir iš jų suformuoti gestą. Papildomai, naudotojas gali aktyvuoti automatinį reagavimą į gestus.

3.3. Architektūros tikslai ir apribojimai

Yra keletas reikalavimų ir apribojimų, kurie turi įtaką sistemos architektūrai. Jie yra:

- Naudotojas turi beveik nejausti, kad jis naudoja gestų sąsają.
- Neturėtų būti naudojamos specialios pirštinės, specialūs jutikliai, tvirtinami prie rūbų, ar kiti priedai, siejami su naudotoju, o ne su kompiuteriu.
- Sistema turi būti galima naudotis dėvint įprastinę aprangą.

- Turi būti siekiama išvengti reikalavimo paspausti tam tikrus mygtukus, apibrėžiančius gesto pradžią ir pabaigą.
- Reakcijos laikas neturėtų viršyti 100 ms., kad naudotojas visai nejaustų gestų sąsajos. [1]
- Kuriant sistemą bus siekiama išlaikyti reakcijos laikus tarp 0,1-1 s.
- Sistemos funkcijos turėtų būti aiškios ir lengvai suprantamos naudotojui.
- Pažengusiems naudotojams leisti patiems kurti funkcijas.
- Naudotojui turi būti rodomi pranešimai apie jo rodomą gestą.
- Sistema turi veikti „Windows“ aplinkoje.

3.4. Reikalavimai sistemai

Detalizuojant sistemos panaudos atvejus, buvo apibrėžta 14 funkcinių ir 11 nefunkcinių reikalavimų. Dauguma funkcinių reikalavimų susiję su duomenų nuskaitymu ir darbo su sistema funkcionalumo apibrėžimu. Nefunkciniai reikalavimai apibrėžia reikalavimus sistemos išvaizdai, panaudojamumui, vykdymo charakteristikoms. Svarbiausi reikalavimai yra šie:

Reikalavimas #:	4	Reikalavimo tipas:	10	Įvykis/panaudojimo atvejis #:	1
Aprašymas:	Sistema turi leisti naudotojui įvesti naują gestą į sistemą.				
Pagrindimas:	Leis sistemai registruoti įvestus gestus, pagal kuriuos bus vykdomos atitinkamos funkcijos.				
Šaltinis:	Naudotojai				
Tikimo kriterijus:	Galima bus peržiūrėti įvestus gestus.				
Užsakovo tenkinimas:	3	Užsakovo netenkinimas:	5		
Priklausomybės	1-3	Konfliktai:	Nėra		
Papildoma medžiaga:	„Microsoft Kinect SDK“ dokumentacija.				
Istorija:	Užregistruotas 2012 Vasario 24d.				

Reikalavimas #:	8	Reikalavimo tipas:	11	Įvykis/panaudojimo atvejis #:	4
Aprašymas:	Sistema turi leisti naudotojui peržiūrėti įvestų gestų parametrus.				
Pagrindimas:	Leis naudotojui matyti įvestų gestų tolerancijos lygius (paklaidas), susietas funkcijas su gestu.				
Šaltinis:	Naudotojai				
Tikimo kriterijus:	Galima bus peržiūrėti įvesto gesto parametrus.				
Užsakovo tenkinimas:	3				Užsakovo netenkinimas: 4
Priklausomybės	4-5				Konfliktai: Nėra
Papildoma medžiaga:	„Microsoft .Net“ dokumentacija.				
	„Microsoft Kinect SDK“ dokumentacija.				
Istorija:	Užregistruotas 2012 Vasario 24d.				

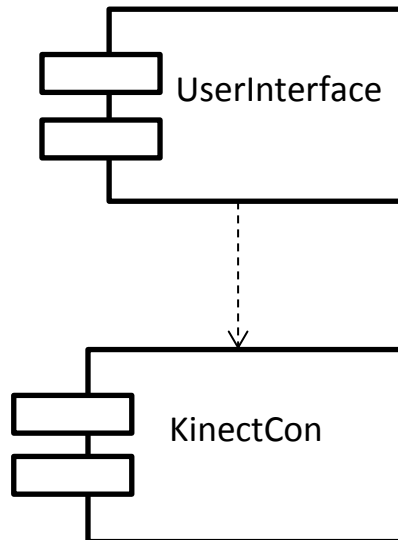
Reikalavimas #:	10	Reikalavimo tipas:	11	Įvykis/panaudojimo atvejis #:	5
Aprašymas:	Sistema turi leisti naudotojui priskirti pageidaujamas funkcijas gestui.				
Pagrindimas:	Leis naudotojui priskirti pageidaujamas funkcijas gestui.				
Šaltinis:	Naudotojai				
Tikimo kriterijus:	Galima bus peržiūrėti įvesto gesto funkcijas.				
Užsakovo tenkinimas:	4				Užsakovo netenkinimas: 4
Priklausomybės	4,5,8				Konfliktai: Nėra
Papildoma medžiaga:	„Microsoft .Net“ dokumentacija.				
Istorija:	Užregistruotas 2012 Kovo 2d.				

3.5. Sistemos architektūra

Šio skyriaus tikslas surinkti ir pateikti svarbius architektūrinius sprendimus, kurie buvo atlikti, projektuojant sistemą.

3.5.1. Sistemos statinis vaizdas

Šiame poskyryje detalizuojami esminiai sistemos loginės struktūros aspektai.



Paveikslas 4 Sistemos suskaidymas į komponentus

Sistema suskaidyta į du pagrindinius komponentus – *UserInterface* ir *KinectCon*:

3.5.1.1. Komponentas „UserInterface“

Apibrėžimas

Komponente pateikiamos klasės, realizuojančios visą sistemos naudotojo sąsają. Šis paketas realizuoja duomenų atvaizdavimą, kitų paketų funkcijų iškvietimą. Paketas naudoja „KinectCon“ komponentą, tam kad gauti išsaugotus naudotojo gestus, suaktyvinti funkciją reaguoti į gestus automatiškai, taip pat apdoroja vaizdiniam pateikimui gaunamus duomenis iš išorinio judesių jutiklio.

Atsakomybės

Komponentas realizuoja gestų kūrimo, redagavimo, šalinimo funkcijas, parametrų redagavimą, susietų funkcijų kūrimą, redagavimą, šalinimą, automatinio reagavimo aktyvinimą, gaunamų duomenų atvaizdavimą.

Struktūra

Komponentą sudaro klasės pateiktos paveiksle nr. 6. „Komponento „UserInterface“ klasių diagrama.

Sąveikavimas

Komponentas naudoja „KinectCon“ komponentą.

Resursai

Naudojamos Microsoft .NET bibliotekos.

Sąsaja/eksportas

Komponentas bendrauja su naudotoju per WPF formas.

3.5.1.2. Komponentas „KinectCon“

Apibrėžimas

Komponente pateikiamos klasės, realizuojančios sistemos duomenų struktūrą ir jos kintamųjų keitimą. Šis komponentas realizuoja duomenų saugojimą, įkėlimą, redagavimą, automatinį reagavimą.

Atsakomybės

Komponentas realizuoja duomenų tvarkymo funkcijas, automatinį reagavimą.

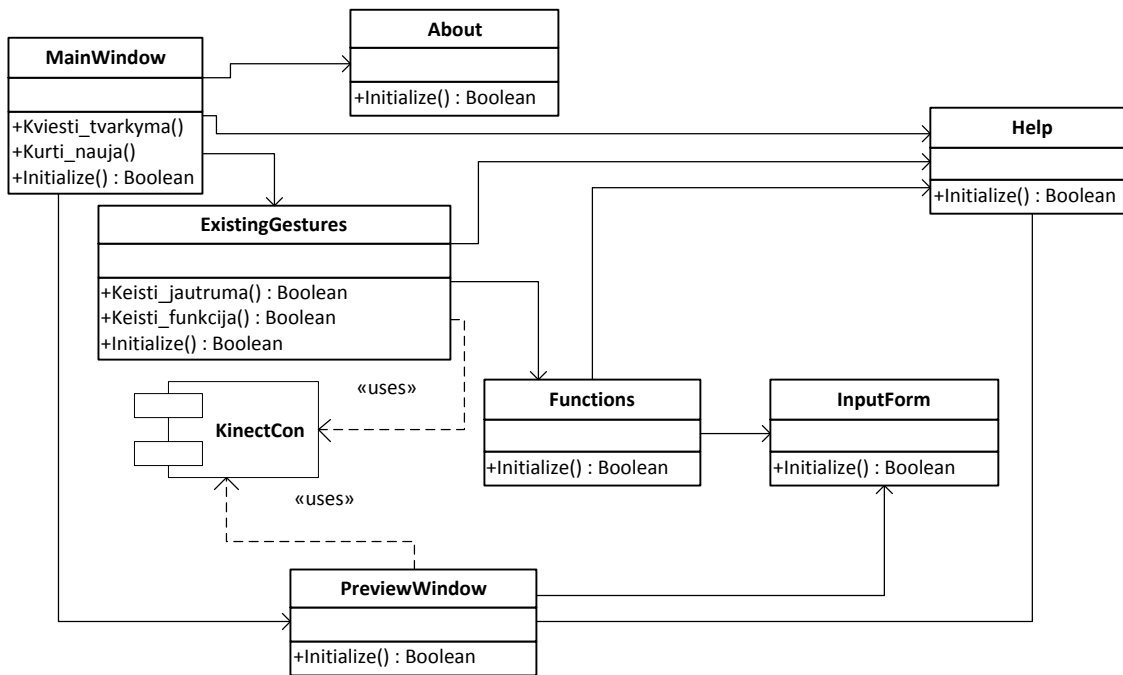
Struktūra

Komponentą sudaro klasės pateiktos paveiksle nr. 7 „Komponento „KinectCon“ klasių diagrama“

Resursai

Naudojamos Microsoft .NET bibliotekos.

3.5.2. Komponentas „UserInterface“



Paveikslas 5 Komponento „UserInterface“ klasių diagrama

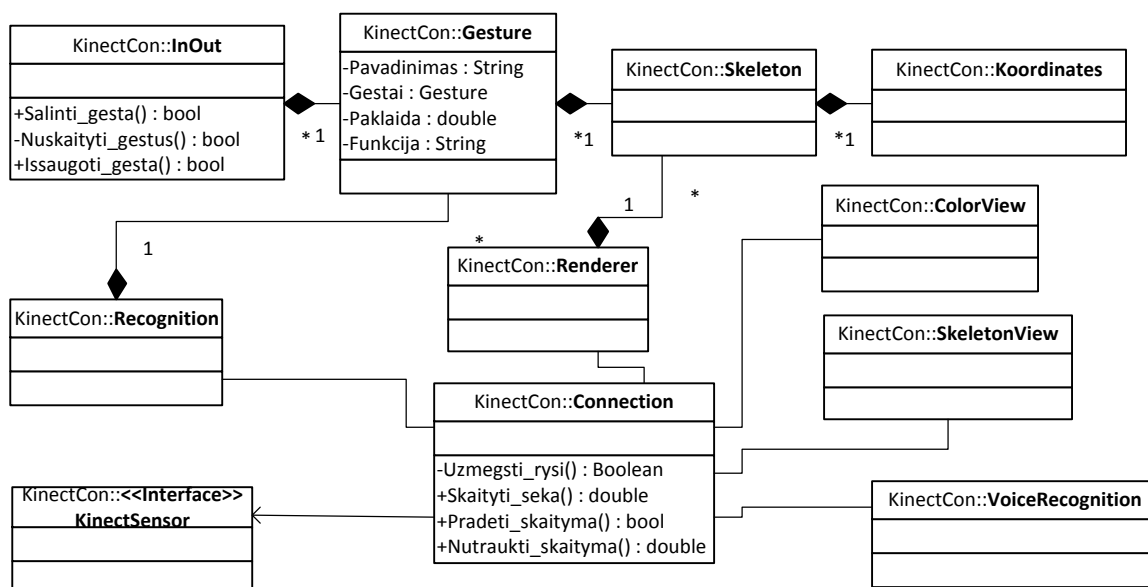
Paveiksle nr. 5 „Komponento „UserInterface“ klasių diagrama“ yra pateikta „UserInterface“ komponento klasių diagrama. Šiame komponente yra realizuotos šios klasės:

- *MainWindow* – klasė, atsakinga už bendrą sistemos naudojimąsi. Kitų naudotojo sąsajos dialogų iškvietimą. Aktyvavus automatinį reagavimą, ši klasė inicijuoja veiksmų klausytojus iš „KinectCon“ komponento ir suaktyvina „KinectCon“ komponento „Recognition“ klasės metodus, skirtus automatiniam gesto aptikimui. Kai aptinkamas gestas, ši klasė gauna parametrus, koks gestas yra aptiktas ir pati atlieka atitinkamus veiksmus (konkrečiu realizacijos atveju yra iškviečiamos nustatytos funkcijos).
- *ExistingGestures* – dialogo, leidžiančio tvarkyti įvestus gestus, klasė. Ši klasė realizuoja duomenų mainus tarp „UserInterface“ komponento ir „KinectCon“ komponento. Ši klasė taip pat gali iškviešti pagalbos „Help“ ir funkcijos keitimo „Functions“ dialogus.
- *PreviewWindow* – klasė, realizuojanti naudotojo sąsają, leidžiančią įvesti naujus gestus į sistemą. Klasės pagrindinis tikslas – atvaizduoti gaunamą duomenų srautą iš „KinectCon“ komponento, leisti naudotojui suformuoti pageidaujama gestą ir jį perduoti „KinectCon“ komponentui, kad būtų išsaugotas. Ši klasė taip pat naudoja pagalbos „Help“ ir tekstinės reikšmės įvedimo „InputForm“ dialogus.
- *Functions* – dialogas, leidžiantis įvesti ar redaguoti su gestu susiejama funkcija. Pagrindinis tikslas – atvaizduoti galimus funkcijų variantus ir jų parametrus.

Suformavus pageidaujamą funkciją, duomenys gražinami atgal „ExistingGestures“ klasei, kuri, vartotojui pageidaujant, įvykdys duomenų saugojimo kvietimą.

- *Input form* – standartinis tekstinės reikšmės įvedimo dialogas. Leidžia naudotojui įvesti pageidaujamą tekstą, kuris bus priskirtas arba gesto, arba gestui priskirtos funkcijos vardui atvaizduoti.
- *Help* – naudotojo pagalbinis dialogas. Priklausomai nuo jį iškvietusios klasės, atvaizduoja duomenis, kuriuose pateikiama atitinkama naudojimosi instrukcija.
- *About* – duomenų apie programą (kūrėjas, kontaktinė informacija) atvaizdavimo dialogas.

3.5.3. Komponentas „KinectCon“



Paveikslas 6 Komponento „KinectCon“ klasių diagrama

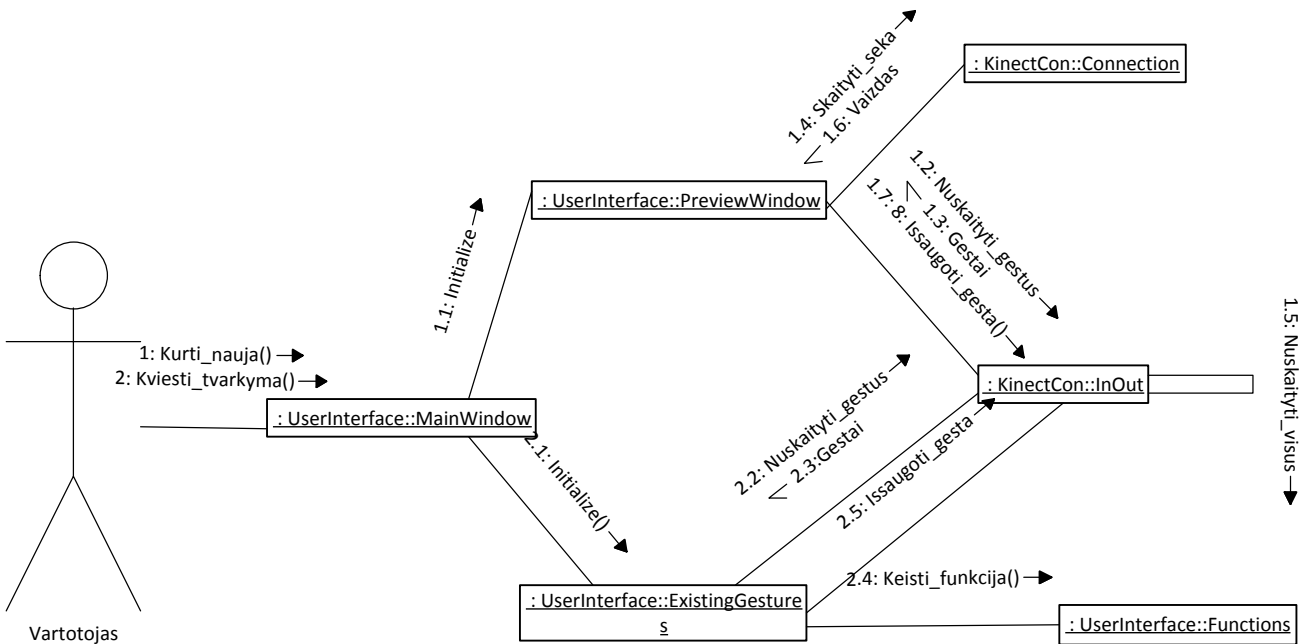
Paveiksle nr. 6 „Komponento „KinectCon“ klasių diagrama“ yra pateikta „UserInterface“ komponento klasių diagrama. Šiame komponente yra realizuotos šios klasės:

- *Connection* – klasė, atliekanti ryšio su išoriniu jutikliu per sąsają „KinectSensor“ užmezgimą. Taip pat ši klasė veikia kaip valdiklis, apdorojanti pagrindines iškviečiamas komandas.
- *SkeletonView* – klasė, inicijuojanti skeleto duomenų nuskaitymą iš išorinio judesių jutiklio.
- *ColorView* – klasė, inicijuojanti spalvinių duomenų nuskaitymą iš išorinio judesių jutiklio. Spalviniai duomenys naudojami naudotojo sąsajoje ir skirti naudotojo aplinkai atvaizduoti. Pagrindinis šios klasės ir jos duomenų tikslas yra, kad klasės „Renderer“ veikimo metu generuojamas vaizdas būtų interaktyvesnis.

- *VoiceRecognition* – klasė, realizuojanti balso komandų nuskaitymą. Balso komandos yra naudojamos gesto įvedimo metu, su tikslu užfiksuoti rodomą momentą. Pagrindinis panaudojimo atvejis – rodomas dviejų rankų gestas ir naudotojui turi būti galimybė užfiksuoti gestą, tad buvo pasirinktas variantas, kad bus ištariama nustatyta balso komanda fiksacijai patvirtinti.
- *Renderer* – klasė, vykdanči gaunamų iš išorinio jutiklio duomenų atvaizdavimui. Ši klasė atvaizduoja pasirinktus duomenų srautus: „SkeletonView“ ar „ColorView“, arba abu.
- *InOut* – klasė, skirta struktūrizuotam duomenų nuskaitymui ir saugojimui. Duomenis saugomi XML struktūra.
- *Recognition* – klasė, kuri yra inicijuojama, kai naudotojas pageidauja aktyvuoti automatinį reagavimą į gestus. Jos pagrindinis tikslas, yra tikrinimas, ar rodomi naudotojo gestai nėra tarp išsaugotų, jei randamas gestas – aktyvuojamas veiksmų klausytojas, kurio klausosi „UserInterface“ komponentas.
- *Gesture* – gesto duomenų struktūros klasė. Ši klasė savyje turi neribotą kiekį „Skeleton“ tipo duomenų rinkinių, kurie apibrėžia gestų seką. Papildomai šioje klasėje saugomi įvairūs gesto parametrai, kaip pavadinimas, susieta funkcija, tolerancija, t.t.
- *Skeleton* – Klasė, sauganti užfiksuoto ar nuskaityto momento duomenis. Joje saugomos užfiksuoto ar nuskaityto naudotojo skeleto galūnių koordinatės (delno, alkūnės t.t.).
- *Coordinates* – Duomenų struktūros klasė, skirta koordinačių reikšmėms saugoti.

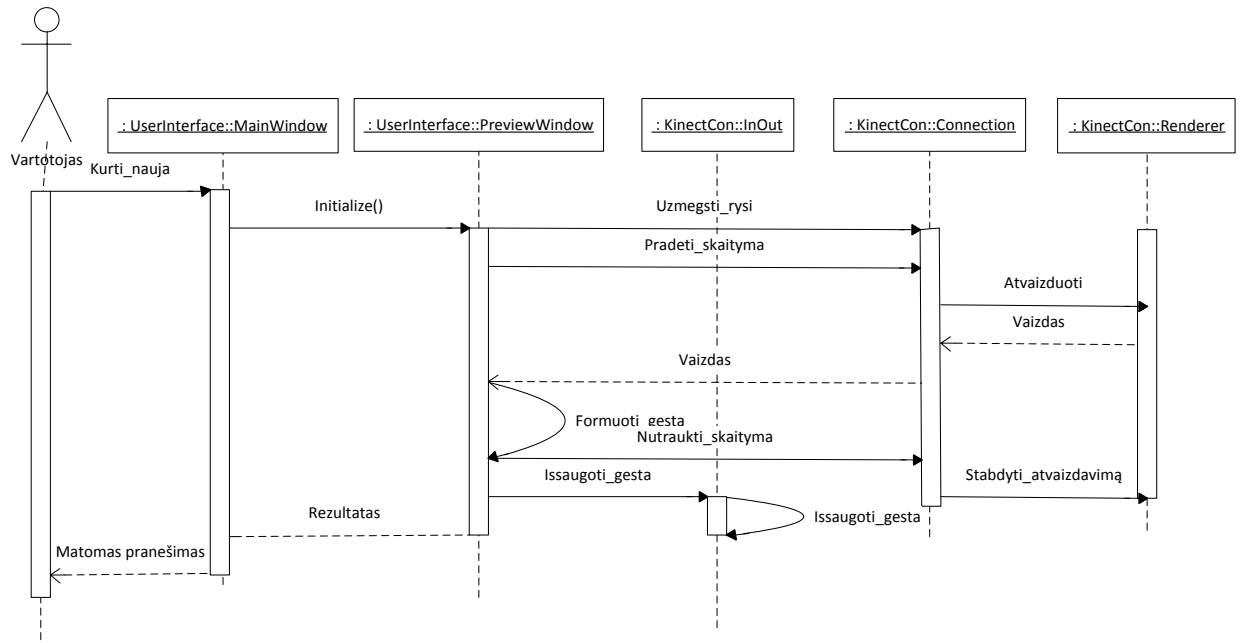
3.5.4. Sistemos dinaminis vaizdas

Šiame skyriuje pateikiamas sistemos dinaminis vaizdas.



Paveikslas 7 Gesto kūrimo ir funkcijos priskyrimo bendradarbiavimo diagrama

Paveiksle 7 „Gesto kūrimo ir funkcijos priskyrimo bendradarbiavimo diagrama“ demonstruojama, kaip sąveikauja sistemos elementai. Diagramoje matomas kviečiamų klasių, ir veiksmų sekos kelias, naudojantis viena pagrindinių funkcijų – gesto įvedimas ir funkcijos priskyrimas. Iš diagramos matoma, kad naudojantis funkcionalumu, leidžiančiu įvesti naują gestą, komponentas „UserInterface“ iškviečia komponento „KinectCon“ dvi klases: „InOut“ ir „Connection“. Pirmoji klasė panaudojama duomenų saugojimui, antroji – ryšio su išoriniu jutikliu užmezgimui ir vaizdo generavimui.



Paveikslas 8 Naujo gesto įvedimo sekų diagrama

Paveikslas nr. 8 „Naujo gesto įvedimo sekų diagrama“ demonstruoja naujo gesto įvedimo sekų diagramą. Iš diagramos matoma, kad naudojantis funkcionalumu, leidžiančiu įvesti naują gestą, komponentas „UserInterface“ iškviečia komponento „KinectCon“ dvi klases: „InOut“ ir „Connection“. Pirmoji klasė panaudojama duomenų saugojimui, antroji – ryšio su išoriniu jutikliu užmezgimui ir vaizdo generavimui.

3.6. Projektavimo etapo rezultatai

1. Vykdam projektavimo etapą, buvo stengiamasi kuo labiau įsigilinti į gestų sąsajų realizacijos problemas. Padiniuose etapuose buvo siekiama pagrįsti projekto realizaciją ir sukurti panaudos atvejus. Vėliau sukurtiems panaudos atvejams buvo sukurti funkciniai ir nefunkciniai reikalavimai.
2. Surinkus reikalavimus ir žinant pageidaujamus rezultatus, pradėtos galimo sprendimo paieškos, kuris leistų naudotojui valdyti gestus ir naudotis automatiniu jų atpažinimu. Sukurtas sprendimo projektas, detalizuojant jį klasių, sąveikos, veiklų diagramomis – statinis ir dinaminis vaizdas.
3. Projektavimo metu, lygiagrečiai buvo kuriama sistema, remiantis prototipų principu (projektavimo, programavimo, testavimo darbai atliekami iteratyviai). Tai sutrumpino programinės įrangos kūrimo laiką ir leido lengviau projektuoti, kadangi gestų atpažinimo sistemos projektavimo patirtis buvo minimali.

4. Prototipinis realizavimo principas leido išbandyti priimamus sprendimus sparčiau ir išrinkti geriausią variantą.

4. TYRIMO IR EKSPERIMENTINĖ DALIS

Šiame skyriuje pristatysime:

1. Automatinio atpažinimo algoritmą: formuluojamas algoritmo, realizuojančio automatinį atpažinimą siūlymas, detalizuojant jo veikimo specifiką.
2. Analizuojamų duomenų kiekio mažinimo įtakos gesto atpažinimo tikimybei tyrimą: eksperimentų atlikimas su realizuotu automatinio atpažinimo algoritmu siekiant nustatyti, ar tikrinamų duomenų kiekio mažinimas yra tinkamas sprendimas siekiant sumažinti sisteminių resursų sunaudojimą..
3. Matavimo paklaidų įvertinimo tyrimą: siekiama nustatyti išorinio jutiklio matavimo paklaidų kitimą, priklausomai nuo aplinkos sąlygų. Tyrimo metu atliekami eksperimentai esant kintančiom sąlygom, siekiant nustatyti optimalią aplinką, kurioje gaunamos mažiausios paklaidos.
4. Neatpažintų gestų kiekio mažinimo, panaudojant spartaus aptikimo algoritmą, tyrimą. Remiantis matavimo paklaidų įvertinimo tyrimu, parenkama aplinkos konfigūracija, sukurianti mažiausios paklaidos duomenis ir normalias naudojimosi sąlygas. Žinant, kad paklaidos pakankamai mažos, tiriama spartaus atpažinimo algoritmo įtaka neaptiktų gestų kiekiui.
5. Sistemos kokybės tyrimą: atliekamas sukurtos sistemos tyrimas, analizuojant įvairias kodo metrikas ir sistemos projekto teisingumą.

4.1. Tyrimų eksperimentuose naudojamos sistemos parametrai

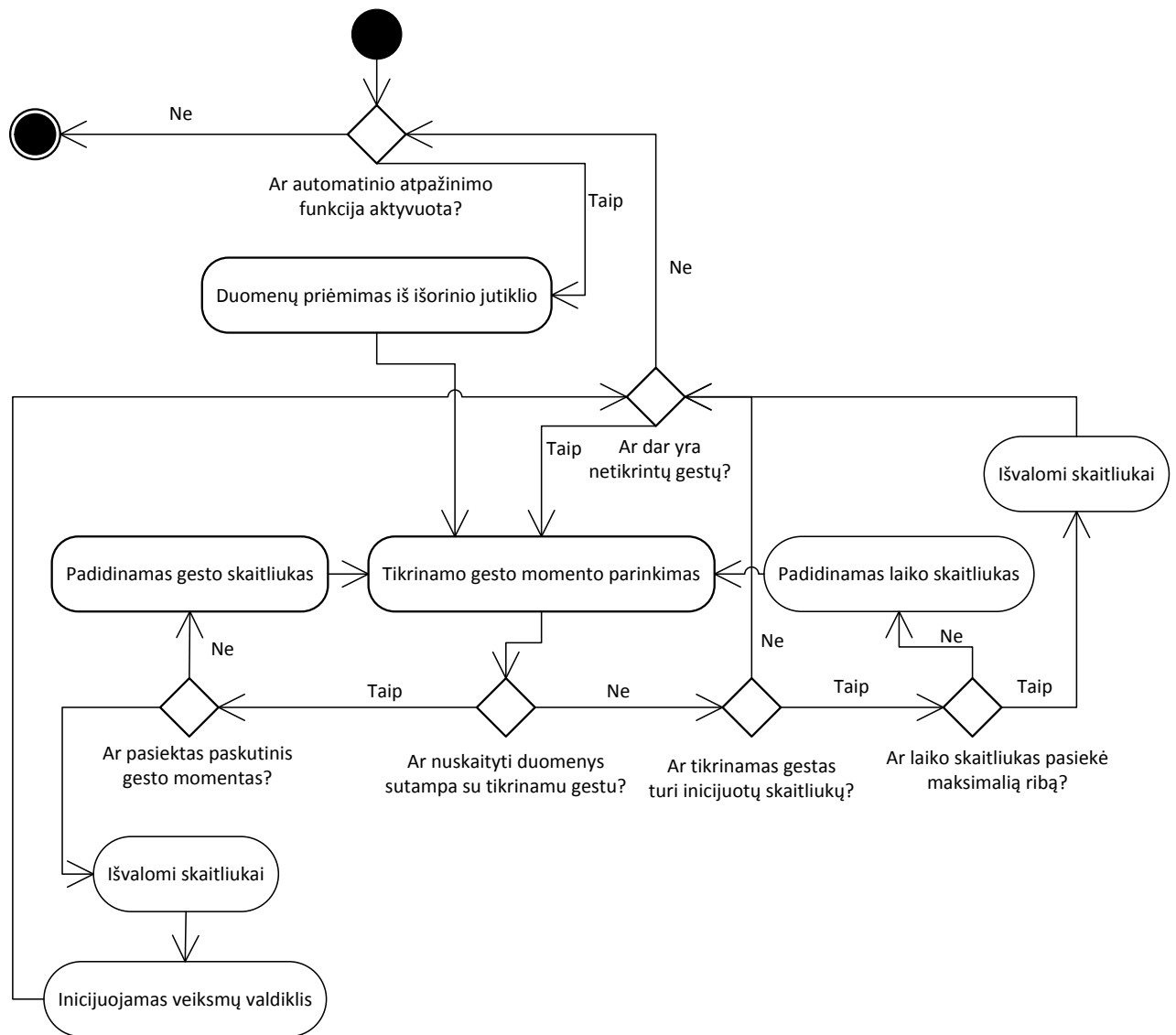
Tyrimui yra naudojama tokia aparatūrinė bei programinė įranga:

- Centrinis procesorius: Intel i5-2410M 2.30Ghz.
- Operatyvioji atmintis: DDR3 6 GB @1334 MHz.
- Grafinis procesorius: GeForce GT 555M..
- Operacinė sistema: Microsoft Windows 7 64 bitų.

Visų tyrimų rezultatai yra tiesiogiai priklausomi nuo aukščiau aprašytos įrangos. Tai reiškia, jog kitos konfigūracijos atveju gali būti gauti šiek tiek kitokie arba net visai priešingi rezultatai.

4.2. Automatinio atpažinimo algoritmas

Šiame skyriuje yra aprašomas siūlomas automatinio atpažinimo algoritmas. Šis algoritmas leidžia nepertraukiant sistemos darbo vykdyti reagavimą į rodomus gestus. Pagrindinis jo privalumas – nereikalinga rodomo gesto pradžios ar pabaigos fiksacija. Algoritmas realizuotas remiantis principais, aprašytais analizės dalyje. Nuskaitytųjų duomenų apdorojimo algoritmo veikimą atvaizduosime sekų diagrama:



Paveikslas 9 Nuskaitytųjų duomenų apdorojimo algoritmo sekų diagrama

Šio algoritmo tikslas – vykdyti duomenų perdavimą tikrinimo algoritmui ir sekti gestus kurie buvo aptikti. Tikrinimo algoritmui perduodami nuskaityti iš išorinio jutiklio duomenys ir konkretus gestas kurio tikimasi.

Jei tikrinimo algoritmas grąžina, kad nuskaityti duomenys atitinka nagrinėjamą gesto momentą, algoritmas pastumia to gesto skaitliuką, leidžiantį sekti, su kuriais gesto einamaisiais duomenimis iš jutiklio gaunamus duomenis reikėtų lyginti. Jeigu per apibrėžtą laiką nėra pasiekama gesto duomenų sekos pabaiga, gesto skaitliukai išvalomi ir gestas turi būti rodomas nuo pradžių. Šis sekimas reikalingas tam, kad būtų ribotas laikas, per kurį gestas turėtų būti atvaizduotas, siekiant sumažinti tikimybę, kad bus aktyvuota funkcija, kurios naudotojas nesitikėjo.

Tikrinimo algoritmo veikimas paremtas sąlygų tikrinimu. Šio algoritmo tikslas – patikrinti ar gaunamos iš išorinio jutiklio koordinatės atitinka tikrinamą gestą įvertinant paklaidas. Tai įgyvendinama tikrinant ar kiekviena koordinatė telpa į tolerancijos ribas. Tolerancija skaičiuojama remiantis formulėmis:

$$Tol[1] = koordinatė - |1 * paklaida|$$

$$Tol[2] = koordinatė + |1 * paklaida|$$

, kur koordinatė – koordinatė, kuriai skaičiuojama paklaida;

paklaida – paklaidos procentinė išraiška.

Šios formulės pagalba yra sukuriama viršutinė ir apatinė ribos, į kurias turi tilpti atitinkama koordinatė, kuriai ši tolerancija paskaičiuota. Kadangi visos koordinatės tenkina sąlygą $0 \leq koordinatė \leq 1$, tad formulės principas yra prie koordinatės pridėti arba atimti 1 padaugintą iš paklaidos procentinės išraiškos. Šios tolerancijos yra sugeneruojamos kiekvienai gesto koordinatei algoritmo inicializacijos metu visiems įvestiems gestams. Šio veiksmo rezultate tolerancijų pakartotinis generavimas nereikalingas ir sutaupomas skaičiavimo laikas.

Jei bent viena iš gautų koordinačių nepatenka į atitinkamas ribas tai koordinatei, grąžinama, kad neaptiktas gestas, ir, jei yra daugiau gestų, tikrinamas sekantis, su tais pačiais išorinio jutiklio duomenimis. Jei daugiau gestų neaptinkama, laukiama, kol išorinis jutiklis atsiųs naują duomenų objektą, su kuriuo bus kartojamas visas procesas.

4.3. Analizuojamų duomenų kiekio mažinimo įtaka gesto atpažinimo spartai

Naudojantis programine įranga, buvo pastebėta, kad sistema naudoja didesnę sisteminių resursų kiekį nei buvo tikėtasi, ko pasekoje neįgyvendinamas vienas iš sistemos nepastebimumo tikslų: sistemos reakcijos laikas. Esant didesniam įvestų gestų kiekiui, buvo pastebėta kad kartais gesto nuskaitymas trunka ilgiau nei 1s. Diagrama, atvaizduojanti reakcijos laiko augimą, priklausomai nuo įvestų gestų kiekio, pateikta paveiksle nr. 13 „Reakcijos laiko priklausomybė nuo įvestų gestų kiekio“.



Paveikslas 10 Reakcijos laiko priklausomybė nuo įvestų gestų kiekio

Eksperimento metu nustatyta, kad ilgą gesto nuskaitymą sukelia du veiksniai:

1. Esant dideliui kiekiui įvestų gestų, automatinio reagavimo algoritmas praranda efektyvumą dėl didelio gaunamų duomenų kiekio. (30 duomenų rinkinių per sekundę).
2. Įvestų gestų duomenys nėra rikiuojami, nėra suskirstomi į kategorijas ar kaip kitaip apdorojami, tad paieška tarp jų yra sudėtinga.

Siekiant sumažinti naudojamų resursų kiekį, buvo pakoreguotas automatinio atpažinimo algoritmas. Realizuota nauja sąlyga: jei nėra saistomoje įvestų gestų, turinčių ypatingai žemą toleranciją ($tol < 10\%$), analizės metu, kas antras duomenų rinkinys, nuskaitytas iš išorinio jutiklio, yra praleidžiamas. Šis sprendimas efektyvią nuskaitymų spartą sumažino iki 15 kadrų per sekundę, bet testavimo metu problemų nesukėlė. Papildomai įvestiems gestams buvo pritaikytas rikiavimas ir algoritmas papildytas sąlyga, leidžiančia jam nustatyti, ar reikia toliau iteruoti įvestų gestų sąrašą, jei dar yra tikimybė, kad rodomas gestas bus surastas tarp įvestų.

Dėl atliktų aptikimo algoritmo korekcijų, pavyko pasiekti, kad tik esant 24 gestams, yra pasiekama 1s. reakcijos laiko riba. Optimizuotos programinės įrangos reakcijos spartos priklausomybės nuo įvestų gestų kiekio diagrama pateikta paveiksle nr. 14 „Optimizuotos sistemos reakcijos laiko priklausomybė nuo įvestų gestų kiekio“.



Paveikslas 11 Optimizuotos sistemos reakcijos laiko priklausomybė nuo įvestų gestų kiekio

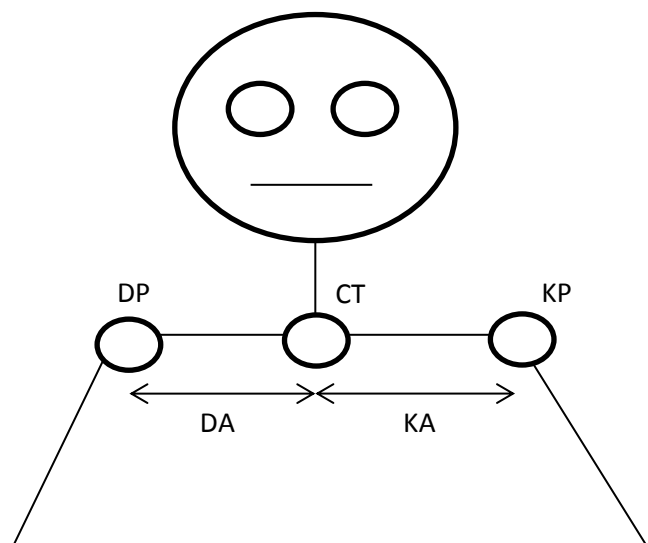
Reakcijos laikas sėkmingai sumažintas 5 kartus. Pasiektas našumas tenkina užsakovo reikalavimus.

4.4. Matavimo paklaidų įvertinimas

Tyrimo metu bus analizuojami duomenys, gaunami iš išorinio jutiklio. Jutiklis realiu metu fiksuoja apibrėžtą rinkinį naudotojo koordinatų (galvos, pečių, alkūnių, plaštakų, t.t.). Konkrečiai analizuojami centrinio taško (CT), kairio (KP) ir dešinio (DP) peties abscisių ašies koordinatų duomenys.

Šių taškų išdėstymas pateiktas paveiksle nr. 12 „Matavimų schema“.

Matavimai atliekami keturiais atstumais tarp naudotojo ir jutiklio: 1m, 1,8m, 2,4m ir 3m. Šie matavimai atlikti esant geram apšviestumui (šviesiu paros metu, nesant tiesioginiams saulės spinduliams, krentantiems į jutiklį ar sukuriantiems atspindžius) ir tamsoje (tamsiu paros metu, jokių šviesos šaltinių). Tamsios aplinkos matavimai paremti



Paveikslas 12 Matavimų schema

jutiklio turimos infraraudonųjų spindulių kameros duomenimis. Kiekvienos konfigūracijos matavimai kartojami 5 kartus, siekiant sumažinti paklaidos tikimybę.

Tikslumo tikrinimas atliekamas remiantis prielaida, kad kairio peties koordinatės (KP) esant idealioms sąlygoms yra vienodai nutolusios nuo centrinio taško (CT), kaip ir dešinio peties koordinatės (DP), t.y. atstumas DA yra vienodas atstumui KA, ko pasėkoje atėmus vieną reikšmę iš kitos, turėtumėm gauti 0. Koordinačių skirtumo matavimo formulė:

$$DIF = (DP - CT) - (CT - KP)$$

Kadangi dešinio peties koordinatės abscisių ašies reikšmė visada bus didesnė už centrinės koordinatės, iš šio skirtumo atėmus centrinės koordinatės ir kairio peties abscisių reikšmių skirtumą (centrinės koordinatės abscisių reikšmės visada didesnės už kairio peties), gauname bendrą koordinačių skirtumą.

Vidutiniam skirtumui matuoti sumuojami kiekvieno matavimo skirtumų moduliai ir padalinama iš matavimų kiekio. Papildomai, bus skaičiuojama koordinačių skirtumo paklaida. Paklaidos skaičiavimo formulė remiasi prielaida, kad taško koordinatės svyruoja tarp -1 ir 1. Paklaidos skaičiavimo formulė:

$$PAKL = \frac{DIF * 100}{2}$$

,kur *DIF* – skirtumas tarp koordinačių

4.4.1. Atlikti matavimai

Lentelėse pateikiami apibendrinti matavimų duomenys. Stulpeliuose „Duomenų kiekis“ pateikiamas nuskaitytų koordinačių kiekis konkrečiam matavimui. Vidutiniškai matavimai truko 7-10 s (duomenų kiekis padalintas iš 30 kadrų per sekundę). Per šį laiką užfiksuotoms koordinatėms paskaičiuojamas vidurkis.

4.4.1.1. Matavimai, esant geram apšviestumui

Lentelė 1 1 m atstumas, geras apšviestumas

Matavimo nr.	Duomenų kiekis	Koordinacių vidurkis			Skirtumas
		Kairio peties	Dešinio peties	Centrinės koordinatės	
1	295	-0,0831	0,2802	0,0957	0,0056
2	249	-0,0736	0,2844	0,1055	-0,0001
3	231	-0,0136	0,3567	0,1694	0,0042
4	266	-0,0482	0,2920	0,1250	-0,0062
5	224	-0,0739	0,3340	0,1220	0,0161
Vidutinis skirtumas:					0,0065

Lentelėje nr. 1 „1 m atstumas, geras apšviestumas“ matomi matavimų, atliktų esant geram apšviestumui ir 1 m. atstumui tarp naudotojo ir jutiklio, apibendrinti duomenys. Realiomis sąlygomis tai yra mažiausias atstumas, kokiu gali būti naudotojas nutolęs nuo jutiklio, kad tilptų į jutiklio matomumo zoną.

Iš šių matavimų matome, kad koordinacių skirtumo modulis svyruoja tarp 0,0001 ir 0,0161, o vidutinis skirtumas yra 0,0065. Šis skirtumas, esant 1 m. atstumui ir šviesiai aplinkai, gaunama 0,325 % paklaida. Žinant, kad atstumas iki jutiklio yra 1 m., remiantis šia paklaida, galime daryti prielaidą, kad jei šios koordinatės būtų panaudotos matavimams, vidutiniškai matmenys turėtų 3,25 mm paklaidą.

Lentelė 2 1.8 m atstumas, geras apšviestumas

Matavimo nr.	Duomenų kiekis	Koordinacių vidurkis			Skirtumas
		Kairio peties	Dešinio peties	Centrinės koordinatės	
1	268	-0,1617	0,1564	-0,0096	0,0139
2	213	-0,1966	0,1795	-0,0127	0,0083
3	240	-0,1580	0,1427	-0,0085	0,0017
4	243	-0,1613	0,1510	-0,0090	0,0077
5	251	-0,1520	0,1470	-0,0070	0,0090
Vidutinis skirtumas:					0,0081

Lentelėje nr. 2 „1,8 m atstumas, geras apšviestumas“ matomi matavimų, atliktų esant geram apšviestumui ir 1,8 m atstumui tarp naudotojo ir jutiklio, apibendrinti duomenys. Nagrinėjant jutiklio specifikaciją, šis atstumas buvo pateiktas, kaip rekomenduojamas.

Iš šių matavimų matome, kad koordinačių skirtumas svyruoja tarp 0,0017 ir 0,0139, o vidutinis skirtumas yra 0,0081. Žinant, kad jutiklio skaitomos koordinatės svyruoja tarp -1 ir 1, šis skirtumas, esant 1,8 m atstumui ir šviesiai aplinkai, gaunama 0,4% paklaida. Žinant, kad atstumas iki jutiklio yra 1,8 m, remiantis šia paklaida, galime daryti prielaidą, kad jei šios koordinatės būtų panaudotos matavimams, vidutiniškai matmenys turėtų 7,3 mm paklaidą.

Lentelė 3 2,4m. atstumas, geras apšviestumas

Matavimo nr.	Duomenų kiekis	Koordinačių vidurkis			Skirtumas
		Kairio peties	Dešinio peties	Centrinės koordinatės	
1	237	-0,0646	0,0183	-0,0173	-0,0118
2	255	-0,0618	0,0013	-0,0213	-0,0180
3	227	-0,0641	0,0153	-0,0203	-0,0083
4	289	-0,0681	0,0180	-0,0171	-0,0159
5	254	-0,0680	0,0140	-0,0220	-0,0100
				Vidutinis skirtumas:	0,0128

Lentelėje nr, 3 „2,4 m atstumas, geras apšviestumas“ matomi matavimų, atliktų esant geram apšviestumui ir 2,4 m atstumui tarp naudotojo ir jutiklio, apibendrinti duomenys. Šis atstumas buvo pasirinktas, kaip tarpinis tarp rekomenduojamo 1,8 m ir maksimalaus pasirinkto matavimams – 3 m.

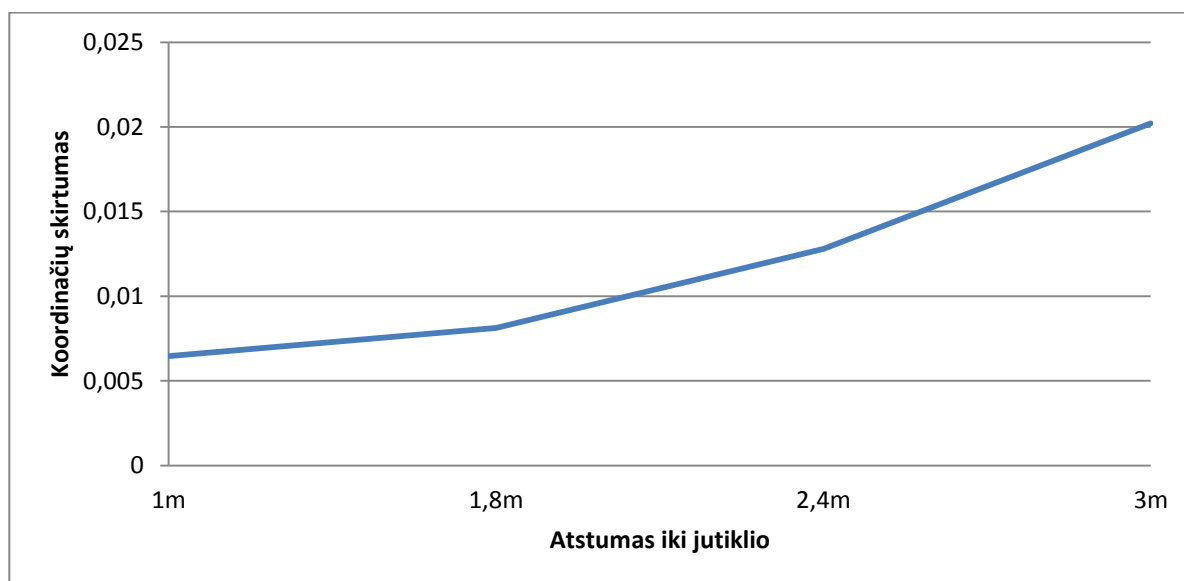
Iš šių matavimų matome, kad koordinačių skirtumo modulis svyruoja tarp 0,018 ir 0,0180, o vidutinis skirtumas yra 0,0128. Šis skirtumas, esant 2,4 m. atstumui ir šviesiai aplinkai, gaunama 0,639% paklaida. Žinant, kad atstumas iki jutiklio yra 1 m, remiantis šia paklaida, galime daryti prielaidą, kad jei šios koordinatės būtų panaudotos matavimams, vidutiniškai matmenys turėtų 1,28 cm paklaidą.

Lentelė 4 3 m atstumas, geras apšviestumas

Matavimo nr.	Duomenų kiekis	Koordinačių vidurkis			Skirtumas
		Kairio peties	Dešinio peties	Centrinės koordinatės	
1	264	-0,1915	0,1566	-0,0293	0,0237
2	239	-0,1823	0,1384	-0,0329	0,0218
3	254	-0,1903	0,1457	-0,0325	0,0204
4	255	-0,1890	0,1420	-0,0320	0,0170
5	235	-0,2090	0,1572	-0,0350	0,0182
Vidutinis skirtumas:					0,0202

Lentelėje nr. 4 „3 m atstumas, geras apšviestumas“ matomi matavimų, atliktų esant geram apšviestumui ir 3 m atstumui tarp naudotojo ir jutiklio, apibendrinti duomenys.

Iš šių matavimų matome, kad koordinačių skirtumas svyruoja tarp 0,0170 ir 0,0237, o vidutinis skirtumas yra 0,0202. Šis skirtumas, esant 3 m atstumui ir šviesiai aplinkai, gaunama 1% paklaida. Žinant, kad atstumas iki jutiklio yra 1 m., remiantis šia paklaida, galime daryti prielaidą, kad jei šios koordinatės būtų panaudotos matavimams, vidutiniškai matmenys turėtų 3,03 cm paklaidą.



Paveikslas 13 Vidutinio koordinačių skirtumo kitimas, priklausomai nuo atstumo iki jutiklio (šviesoje)

Paveiksle nr. 13 „Vidutinio koordinačių skirtumo kitimas, priklausomai nuo atstumo iki jutiklio (šviesoje)“ matomas priklausomybės tarp atstumo ir koordinačių skirtumo grafikas, matavimus atliekant šviesiu paros metu. Šiame grafike aiškiai matosi, kad didinant atstumą tarp jutiklio ir naudotojo, esant geram apšviestumui, paklaida eksponentiškai didėja. Galima priimti išvadą, kad jei būtų atliekami objektų matmenų fiksavimai 1-3 m atstumu, jų matmenų dydžio

paklaida svyruotų tarp 0,325 – 1 %. Remiantis koordinačių skirtumais, papildomai galima priėti išvadą, kad tiksliausius matavimus esant šviesiai aplinkai galime gauti esant 1 m. atstumui, nes vienas iš šio atstumo matavimų buvo beveik idealus (Lentelė nr. 7 „1 m atstumas, geras apšviestumas“, matavimas nr. 2).

4.4.1.2. Matavimai, esant prastam apšviestumui

Lentelė 5 1 m atstumas, tamsa

Matavimo nr.	Duomenų kiekis	Koordinačių vidurkis			Skirtumas
		Kairio peties	Dešinio peties	Centrinės koordinatės	
1	295	-0,1951	0,1258	-0,0479	0,0265
2	292	-0,1748	0,1573	-0,0176	0,0175
3	338	-0,1730	0,1526	-0,0223	0,0204
4	303	-0,1651	0,1336	-0,0238	0,0160
5	351	-0,0071	0,2534	0,1118	0,0227
Vidutinis skirtumas:					0,0206

Lentelėje nr. 5 „1 m atstumas, tamsa“ matomi matavimų, atliktų tamsoje ir 1 m atstumui tarp naudotojo ir jutiklio, apibendrinti duomenys.

Iš šių matavimų matome, kad koordinačių skirtumas svyruoja tarp 0,0175 ir 0,0237, o vidutinis skirtumas yra 0,0202. Šis skirtumas, esant 1 m atstumui tamsoje, gaunama 1 % paklaida. Lyginant su matavimais šviesoje, ši paklaida yra daugiau nei 3 kartus didesnė. Žinant, kad atstumas tarp naudotojo ir jutiklio yra 1 m, ši paklaida sudarytų 1 cm atstumo matavimo paklaidą.

Lentelė 6 1.8 m atstumas, tamsa

Matavimo nr.	Duomenų kiekis	Koordinačių vidurkis			Skirtumas
		Kairio peties	Dešinio peties	Centrinės koordinatės	
1	331	0,2100	0,5558	0,3764	0,0131
2	311	0,2110	0,5505	0,3742	0,0130
3	341	0,2151	0,5571	0,3763	0,0197
4	352	0,2202	0,5612	0,3778	0,0258
5	357	0,2157	0,5568	0,3903	-0,0080
Vidutinis skirtumas:					0,0159

Lentelėje nr. 6 „1,8 m atstumas, tamsa“ matomi matavimų, atliktų tamsoje ir 1,8 m atstumui tarp naudotojo ir jutiklio, apibendrinti duomenys.

Iš šių matavimų matome, kad koordinačių skirtumo modulis svyruoja tarp 0,008 ir 0,0258, o vidutinis skirtumas yra 0,0159. Šis skirtumas, esant 1,8 m atstumui tamsoje, gaunama 0,8% paklaidą.

Lyginant su matavimais šviesoje, ši paklaida yra 2 kartus didesnė. Žinant, kad atstumas tarp naudotojo ir jutiklio yra 1,8 m, ši paklaida sudarytų 1,43 cm atstumo matavimo paklaidą.

Lentelė 7 2,4 m atstumas, tamsa

Matavimo nr.	Duomenų kiekis	Koordinačių vidurkis			Skirtumas
		Kairio peties	Dešinio peties	Centrinės koordinatės	
1	298	-0,0826	0,2683	0,0773	0,0312
2	275	-0,0891	0,2118	0,0689	-0,0151
3	314	-0,0791	0,2018	0,0693	-0,0159
4	288	-0,0891	0,1918	0,0605	-0,0183
5	301	-0,0758	0,2514	0,0713	0,0332
Vidutinis skirtumas:					0,0227

Lentelėje nr. 7 „2,4 m atstumas, tamsa“ matomi matavimų, atliktų tamsoje ir 2,4 m atstumui tarp naudotojo ir jutiklio, apibendrinti duomenys.

Iš šių matavimų matome, kad koordinačių skirtumo modulis svyruoja tarp 0,0151 ir 0,0332, o vidutinis skirtumas yra 0,0227. Šis skirtumas, esant 2,4 m atstumui tamsoje, gaunama 1,13 % paklaida. Lyginant su matavimais šviesoje, ši paklaida yra apytiksliai 1,7 karto didesnė. Žinant, kad atstumas tarp naudotojo ir jutiklio yra 2,4 m, ši paklaida sudarytų 2,73 cm atstumo matavimo paklaidą.

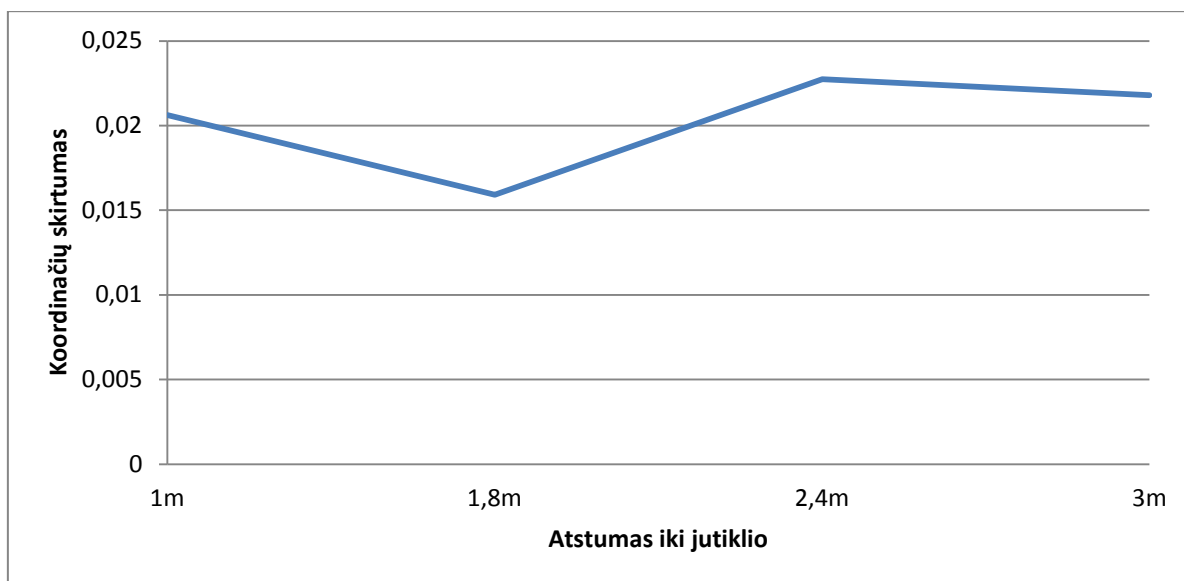
Lentelė 8 3m. atstumas, tamsa

Matavimo nr.	Duomenų kiekis	Koordinačių vidurkis			Skirtumas
		Kairio peties	Dešinio peties	Centrinės koordinatės	
1	360	-0,7170	-0,3702	-0,5546	0,0219
2	381	-0,7314	-0,3884	-0,5740	0,0283
3	362	-0,7436	-0,3959	-0,5776	0,0158
4	344	-0,7868	-0,3534	-0,5812	0,0223
5	328	-0,7980	-0,3509	-0,5848	0,0208
Vidutinis skirtumas:					0,0218

Lentelėje nr. 8 „3 m atstumas, tamsa“ matomi matavimų, atliktų tamsoje ir 3 m atstumui tarp naudotojo ir jutiklio, apibendrinti duomenys.

Iš šių matavimų matome, kad koordinačių skirtumo modulis svyruoja tarp 0,0158 ir 0,0283, o vidutinis skirtumas yra 0,0218. Šis skirtumas, esant 3m. atstumui tamsoje, gaunama 1,09 % paklaida. Lyginant su matavimais šviesoje, ši paklaida yra beveik vienoda (0,0016 didesnė, >8 % skirtumas).

Žinant, kad atstumas tarp naudotojo ir jutiklio yra 3 m, ši paklaida sudarytų 3,27 cm atstumo matavimo paklaidą.



Paveikslas 14 Vidutinio koordinačių skirtumo kitimas, priklausomai nuo atstumo iki jutiklio (tamsoje)

Paveiksle nr. 14 „Vidutinio koordinačių skirtumo kitimas, priklausomai nuo atstumo iki jutiklio (tamsoje)“ matomas priklausomybės tarp atstumo ir koordinačių skirtumo grafikas, matavimus atliekant tamsoje. Iš šio grafiko matome, kad jutikliui naudojant tik infraraudonųjų spindulių kamerą objekto koordinatėms nustatyti, paklaida visuose atstumuose yra panaši (rezultatai 1 m atstumu panašūs į rezultatus 3 m atstumu). Matavimų tikslumas yra iki 3 kartų prastesnis, esant mažam (1m) atstumui. Tik didinant atstumą tarp jutiklio ir naudotojo, paklaidų skirtumas tarp tamsios ir šviesios aplinkos mažėja, ir esant 3 m atstumui beveik susilygina.

Bet reiktų atkreipti dėmesį į tai, kad matavimų paklaidos yra pernelyg didelės bet kokiu atstumu iki jutiklio, norint duomenis panaudoti matavimams, bet naudojantis šiais duomenimis vis tiek galima atlikti naudotojo taškų nustatymą ir judėjimo krypties ar kitus skaičiavimus.

4.5. Neatpažintų gestų kiekio mažinimas panaudojant spartaus aptikimo algoritmą

Šio eksperimento metu, buvo atliekami įvesto gesto atkartojimai, siekiant nustatyti, gesto neaptikimo priežastis.

Lentelė 9 Nenuskaitytų gestų eksperimento aplinkos sąlygos

Veiksny	Reikšmė
Aplinkos apšvietumas	Geras apšvietumas
Atstumas iki jutiklio	1,8 m
Įvesto gesto paklaida	1 %

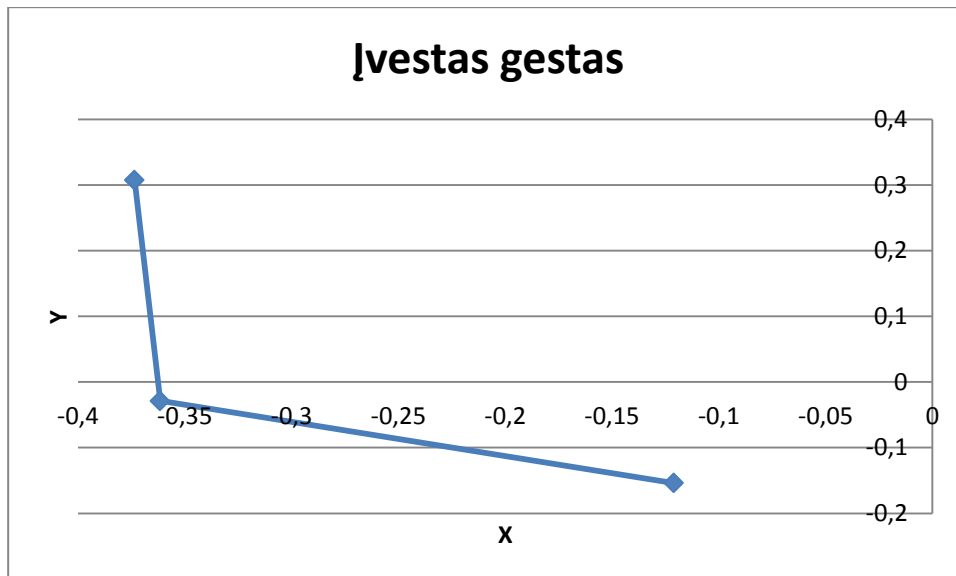
Lentelėje nr. 9 suformuotos eksperimento sąlygos remiasi tyrimo dalies, 4.2 skyriaus „Matavimo paklaidų įvertinimas“ rezultatais.

Eksperimento metu įvestas paprastas trijų kairės rankos taškų gestas, kurio pagrindą sudaro plaštakos judesiai (plaštaka turi kirsti tris taškus, kad būtų užfiksuotas gestas).

Lentelė 10 Nenuskaitytų gestų eksperimento gesto koordinatės

Taško nr.	X	Y	Z
1	-0,37354	0,307355	0,002524
2	-0,36157	-0,02871	0,002125
3	-0,12107	-0,15382	0,002378

Lentelėje nr. 10 „Nenuskaitytų gestų eksperimento gesto koordinatės“ pateiktos eksperimento metu naudoto gesto X, Y, Z koordinatės.

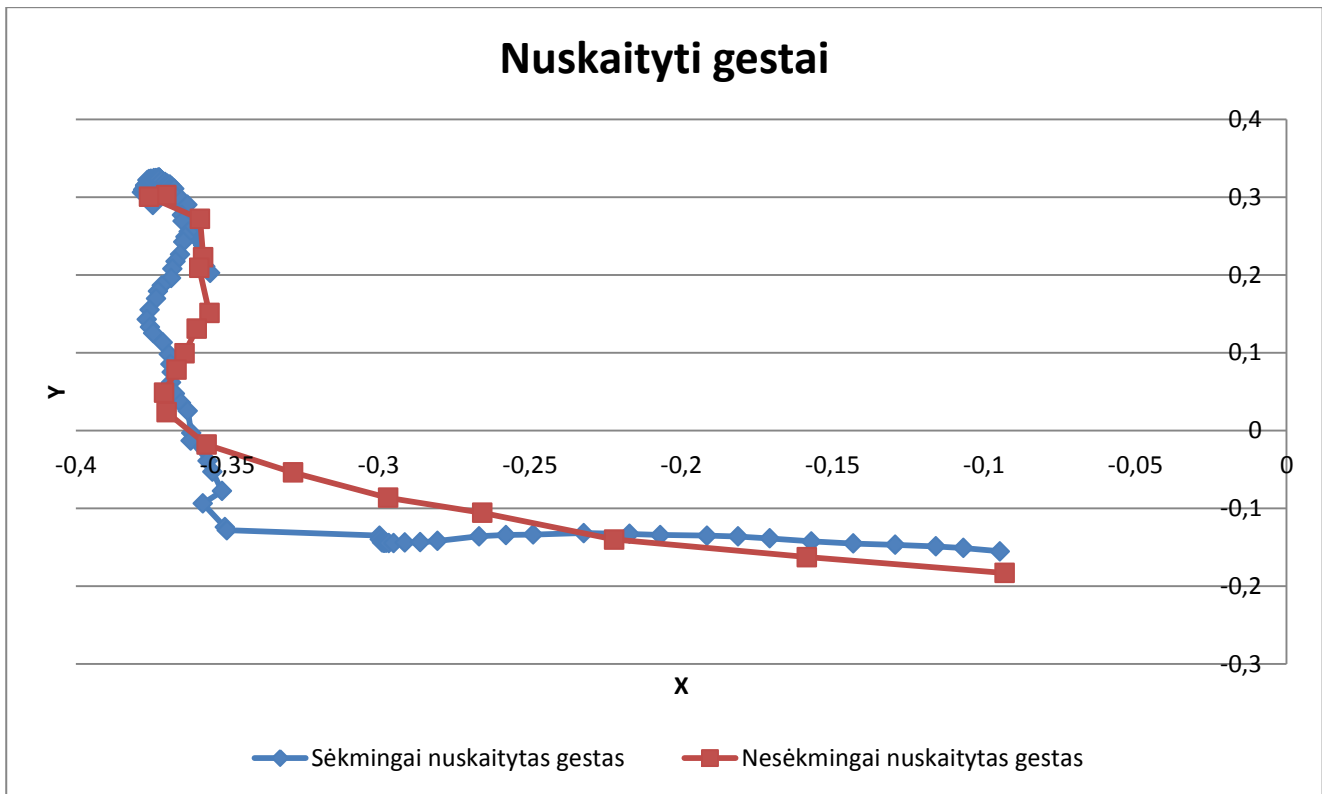


Paveikslas 15 Įvesto gesto koordinatės

Paveiksle nr. 15 „Įvesto gesto koordinatės“ matome teorinį kelią, kuriuo turėtų keliauti kairė plaštaka, siekiant kad gestas būtų užfiksuotas. Šio kelio tarpinės reikšmės tarp dviejų fiksuotų taškų, nėra svarbios, svarbu būtų pasiekiami gesto pradžios, tarpiniai ir galutinis taškai. Z ašies koordinatės nėra atvaizduojamos, kadangi jos sąlyginai nekinta.

Ekspermineto metu, buvo atliekama serija matavimų, kurių metu pastebėta, kad pagrindinė priežastis kodėl gestas yra nuskaitomas yra spartūs plaštakos judesiai, t.y gestas rodomas greitai. Siekiant išanalizuoti priežastį, suformuoti sėkmingai ir nesėkmingai nuskaityto gesto duomenų rinkiniai.

Remiantis šiais rinkiniais, sudaryta X ir Y koordinačių pasiskirstymo diagrama:



Paveikslas 16 Sėkmingai ir nesėkmingai nuskaityto gesto X, Y koordinatinių pasiskirstymo diagrama

Paveiksle nr. 16 „Sėkmingai ir nesėkmingai nuskaityto gesto X, Y koordinatinių pasiskirstymo diagrama“ matome dviejų duomenų rinkinių rezultatus. Mėlynai pažymėtas gestas, kuris buvo nuskaitytas sėkmingai, t.y. šio gesto nuskaitymo metu buvo kirsti visi 3 įvesto gesto taškai. Raudonai pažymėtas gestas yra nesėkmingai nuskaitytas. Analizuojant šio gesto duomenis, nustatyta, kad Z ašies reikšmės nesukėlė nenuskaitymo problemos. Gestas buvo neaptiktas, todėl kad jis buvo rodomas per sparčiai ir arba nebuvo užfiksuotas būtent tą sekundės dalį, kai jis buvo taške 2 ($x = -0,36157$, $y = -0,02871$), arba naudotojas nepakankamai privedė kairę plaštaką iki šio taško. Būtent ši problema aiškiai matoma šiame grafike, kur nenuskaityto gesto judėjimo fiksavimas staigiai pradeda didėti X ašies atžvilgiu, dar nepasiekęs antrojo tako koordinatės.

Nustačius konkrečią problemą, sudarytas problemos sprendimo pasiūlymas: naudotojui pageidaujant, turėtų būti aktyvuojama spartaus gestų rodymo galimybė, leisianti jam gestus rodyti greitais judesiais ir nesukuriant šio eksperimento metu aptiktos situacijos.

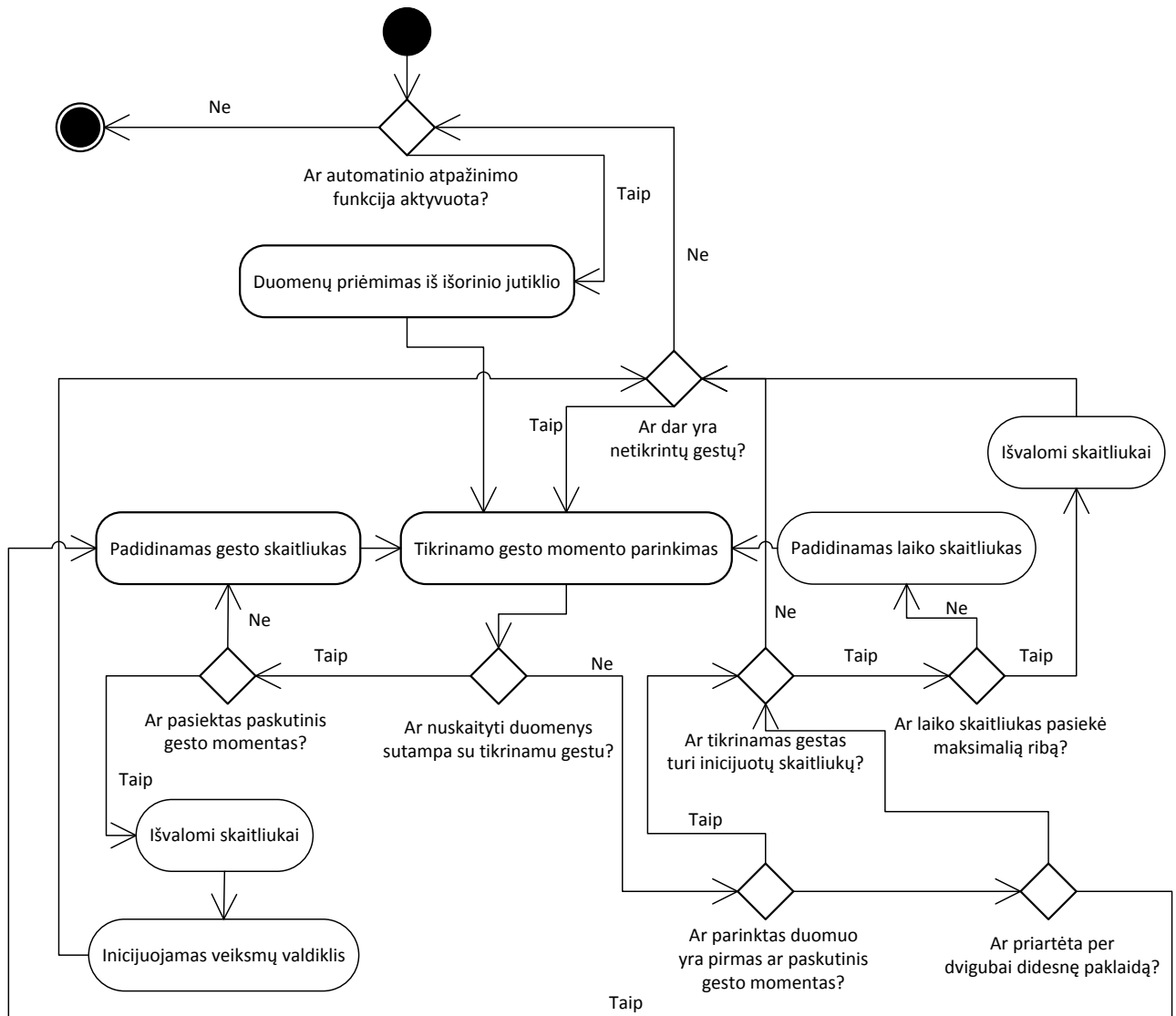
Spartaus rodymo nuskaitymas remiasi šiais principais:

- Naudotojas turi pilnai fiksuoti gesto pradžią ir pabaigą, t.y. gesto pradžia ir pabaiga turi sutapti su įvestu gestu (įvertinus nustatytą paklaidą).

- Užfiksavus atitinkamo gesto pradžią ir analizuojant toliau gaunamą seką, gesto tarpinis taškas yra fiksuojamas, jei prie jo priartėjama per dvigubai didesnę tolerancijos reikšmę, nei yra nustatyta gesto parametruose.

Šie principai apibrėžia idėją, kad naudojantis sparčiu atpažinimu, tarpiniams gesto taškams yra leistina dvigubai didesnė paklaida, nei pradiniam ir galutiniam taškui.

Remiantis šiais principais, siūlome atlikti tokią modifikaciją 4.2 skyriuje pateiktam automatiniam aptikimo algoritmui:



Paveikslas 17 Spartaus judesio atpažinimo algoritmas

Kaip matome paveiksle nr. 17, jeigu gesto momentas yra fiksuojamas kaip nesutampantis, atliekamas papildomas tikrinimas, analizuojantis ar nebuvo priartėta pakankamai arti fiksuoto taško, kad jį būtų galima laikyti fiksuotinu.

Naudodami šią modifikaciją, įrodysime algoritmo veikimo teisingumą:

Lentelė 11 Įvesto gesto, taško nr 2, koordinacių minimalios ir maksimalios reikšmės

	X	Y
Minimali reikšmė	-0,38157	-0,04871
Maksimali reikšmė	-0,34157	-0,00871

Lentelėje nr. 11 „Įvesto gesto, taško nr. 2, koordinacių minimalios ir maksimalios reikšmės“ pateikiamos įvesto gesto, antro taško, minimalios ir maksimalios reikšmės, atsižvelgiant į tai, kad tai yra tarpinis taškas, todėl jam taikoma dviguba (2 %) įvesto gesto tolerancijos reikšmė (tolerancijos skaičiavimo metodika pateikta skyriuje 3.5 „Automatinio atpažinimo algoritmas“).

Remiantis šiomis ribomis, tarp neaptikto gesto duomenų, identifikuotas matmuo, kurio koordinatės: $X = -0,3568317$, $Y = -0,01795644$. Šios koordinatės būtų fiksuojamos kaip gesto tarpinis taškas ir šis gestas būtų atpažintas, nors ir tarpinis taškas nebuvo pilnai pasiektas, bet prie jo buvo pakankamai priartėta.

4.6. Sistemos išėities kodo analizė

Kodo analizė atliekama naudojant „Microsoft Visual Studio Code Metrics“ ir „FxCop“ [10] automatinius įrankius. „Microsoft Visual Studio Code Metrics“ produktas atlieka gali išmatuoti tokias metrikas, kaip:

Dydžio metrikos, pvz., sistemos klasių (klasių metodų, funkcijų, modulių, kintamųjų) skaičius; programos dydis kodo eilutėmis (LOC), komentarų kiekis (COM);

Programos modulio (klasės, komponento) sudėtingumas yra žemas, jei jo dydis yra mažesnis nei 100 LOC.

Objektinės sudėtingumo metrikos:

- **Jungumas tarp klasių** (*coupling between object classes*, CBO). Kuo didesnis jungumas, tuo sudėtingesnė klasės priežiūra. Jei $CBO > 14$, klasę reikia labai gerai ištestuoti.
- **Paveldėjimo medžio gylis** (*depth of inheritance tree*, DIT) – ilgiausio kelio iki šakninės klasės ilgis. Aukštesnė DIT reikšmė reiškia sudėtingesnę sistemą, kurioje gali būti daugiau klaidų. Rekomenduojama, kad DIT reikšmė būtų ≤ 5 .
- **McCabe Ciklomatinis sudėtingumas** (*Cyclomatic Complexity*, CC) – nepriklausomų kelių programos kodo grafe skaičius;

Lentelė 12 Ciklomatinio sudėtingumo reikšmės vertinimas:

CC reikšmė	Programos įvertinimas
1 - 10	paprasta, klaidų tikimybė maža
11 - 20	vidutinio sudėtingumo, klaidos tikimybė vidutinė
21 - 50	sudėtinga, didelė klaidos tikimybė
> 50	netestuojama

- **Prižiūrimumo indeksas** (Maintainability Index, MI) – programų sistemos priežiūrai reikalingų pastangų matas.

Prižiūrimumo indeksas gali būti skaičiuojamas tiek visai sistemai (tuomet imamos vidutinės parametrų reikšmės), tiek atskiriems jos moduliams.

Lentelė 13 Prižiūrimumo indekso reikšmės vertinimas:

MI reikšmė	Programos įvertinimas
> 85	geras prižiūrimumas
65 – 85	vidutinis prižiūrimumas
0 - 65	blogas prižiūrimumas
< 0	labai blogas programos kodas (nestructūrizuotas, nekomentuotas)

Programų keitimas atliekamas atsižvelgiant į programų sudėtingumo tyrimo rezultatus: laikoma, kad reikia atkreipti dėmesį į blogiausius 5 % programos modulius (klases).

Jeigu sistemoje yra daugiau nei 5 % modulių, kurie viršija ribinę prižiūrimumo vertę, tuomet visa sistema reikalauja priežiūros.

Jeigu sistema vertinama pagal kelias metrikas, tuomet reikia atkreipti dėmesį į tuos modulius, kurie 5 % taisyklę tenkina bent pagal 2 metrikas.

Tyrimo metu bus analizuojamos šios metrikos:

- Chidamber ir Kemerer pasiūlytos metrikos [11]: Metodų skaičius klasėje (WMC), jungiamumas tarp klasių (CBO).
- McCabe ciklomatinis sudėtingumas (CC).
- Prižiūrimumo indeksas (Maintainability Index, MI).

Papildoma kodo analizė bus atlikta panaudojant „FxCop“ įrankį. Šis įrankis automatiškai analizuoja kodą ir pateikia galimas problemas šiose srityse: projekto projektavimo logika, globalizacija, sąveika su kitomis sistemomis, palaikomumas, mobilumas, kintamųjų vardų validavimas, našumas, patikimumas, saugumas, naudojamumas.

Šio projekto atveju, didžiausias dėmesys kreipiamas į šias metrikas:

Lentelė 14 Svarbiausi sistemos įvertinimo kriterijai

Eil. Nr.	Sritis	Aprašymas
1.	Išplėčiamumas	Galimybė papildyti sistemą naujomis funkcijomis.
2.	Palaikomumas	Sistemos klaidų taisymo sparta.
3.	Funkcionalumas	Sistemos funkcijų gausa.
4.	Naudojamumas	Ar sistema greitai perprantama naudotojui?

4.6.1. Kodo metrikų rezultatai

Analizuojant klases, buvo paimitos sukurto universalus komponento klasės, turinčios daugiausiai metodų. Šios klasės atlieka esmines sistemos funkcijas ir turi didžiausią tikimybę būti keičiamos ateityje. Rezultatai pateikti lentelėje nr. 15 „klasių metrikos“.

Lentelė 15 klasių metrikos

Klasės pavadinimas	WMC	MI	CC	CBO
Connection	6	71	17	8
SkeletonView	6	79	10	2
ColorView	8	79	12	5
Handlers	6	89	8	6
Koordinates	8	84	11	0
Gesture	17	83	17	1
SkeletonData	23	71	33	5
Recognition	15	63	62	17
SkeletonRenderer	16	63	42	29
InOut	7	51	45	16

Išanalizavus sugeneruotas metrikas buvo priimtas sprendimas peržiūrėti klases, kurių prižiūrimumo indeksas yra mažesnis nei 65, kas rodo blogą prižiūrimumą (lentelė nr. 2 „Prižiūrimumo indekso reikšmės vertinimas“). Siekiant detaliau nustatyti galimas problemines sritis, identifikuotų klasių metodams buvo sugeneruotos prižiūrimumo indekso (MI), ciklomatinio sudėtingumo (CC), jungiamumo tarp klasių (CBO) metrikos.

Lentelė 16 metodų, turinčių žemesnį nei 65 prižiūrimumo indeksą (MI) metrikos

Klasės pavadinimas	Metodo pavadinimas	MI	CC	CBO
InOut	LoadAll() : Gesture[]	34	29	10
InOut	Save(Gesture) : bool	42	3	5
Recognition	Checking(float, SkeletonData, SkeletonData) : bool	50	21	2
SkeletonRenderer	SensorSkeletonFrameReady(object, SkeletonFrameReadyEventArgs) : void	52	8	11
SkeletonRenderer	DrawBonesAndJoints(Skeleton, DrawingContext) : void	53	6	7
Recognition	Processing(SkeletonData) : void	53	8	4
InOut	FormKoordinates(XmlReader) : koordinates	57	8	2
Recognition	Recognition()	58	3	4
InOut	WriteBatch(XmlTextWriter, string, koordinates) : void	59	1	3
SkeletonRenderer	DrawBone(Skeleton, DrawingContext, JointType, JointType) : void	60	7	6
Recognition	SensorSkeletonFrameReady(object, SkeletonFrameReadyEventArgs) : void	60	6	4
SkeletonRenderer	DrawBonesAndJoints(SkeletonData, DrawingContext) : void	61	1	2
SkeletonRenderer	SkeletonRenderer(DrawingGroup, Connection)	62	1	7
SkeletonRenderer	SaveMoment(bool, bool) : int	62	3	3
SkeletonRenderer	SkeletonRenderer(DrawingGroup)	63	1	6

Įvertinus lentelėje nr. 16 „metodų, turinčių žemesnį nei 65 prižiūrimumo indeksą (MI) metrikos „ pateiktas metrikas pradinis dėmesys buvo sutelktas į metodus, turinčius didesnį nei 20 ciklomatinį sudėtingumą.

Klasės „Recognition“ metodui „Checking“ pritaikytas veiksmų išskirstymas, t.y. tam tikrų sąlygų tikrinimas perkeltas į naujus metodus, taip sumažinant ciklomatinį sudėtingumą. Šio veiksmo pasekoje klasėje „Recognition“ atsirado du nauji metodai, kurių ciklomatinis sudėtingumas yra 7 ir 12. Metodo „Checking“ ciklomatinis sudėtingumas nukrito iki 17.

Klasės „InOut“ metodas „LoadAll“ vykdė struktūrizuotų XML duomenų nuskaitymą iš rinkmenos, ko pasekoje jame buvo realizuota daug sąlygų. Šios problemos sprendimui pritaikyta XML struktūros elementų deserilizacijos metodologija, leidžianti XML rinkmenų duomenis tiesiogiai priskirti objektų kintamiesiems, kai tie objektai yra atitinkamai sugeneruoti remiantis tos rinkmenos struktūra. Šio sprendimo realizavimas pakėlė metodo palaikomumo indeksą iki 73 ir sumažino ciklomatinį sudėtingumą iki 13. Papildomai buvo sukurta nauja klasė „XmlGesture“.

4.6.2. Sistemos korektiškumo įvertinimas.

Atlikus sistemos korektiškumo įvertinimą, nebuvo aptikta kritinių klaidų. Įrankis „FxCop“ sugeneravo kelis įspėjimus, kurie buvo sutvarkyti. Bendras sugeneruotų rezultatų įvertinimas pateiktas lentelėje nr. 17 „Sistemos korektiškumo įvertinimas“

Lentelė 17 Sistemos korektiškumo įvertinimas

Eil. Nr.	Sritis	Įvertinimas
1.	Išplėčiamumas	Puikus
2.	Palaikomumas	Geras
3.	Funkcionalumas	Patenkinamas
4.	Naudojamumas	Geras

Ateityje plečiant sistemos galimybes yra tikimasi, kad funkcionalumo įvertinimas kils. Svarbiausi įvertinimai, kaip išplėčiamumas ir palaikomumas yra aukštame lygyje, kas leis nesudėtingai papildyti sistemą nauju funkcionalumu, išlaikant kitas metrikas aukštame lygyje.

5. REZULTATŲ APIBENDRINIMAS IR IŠVADOS

- 5.1. Išanalizavus užsakovo reikalavimus sistemai, nustatyti pagrindiniai sistemos kūrimo uždaviniai. Remiantis šiais uždaviniais sudarytas galimų problemų sprendimo sąrašas ir sukurta gestų valdymo sąsaja, naudojančių „Kinect“ įrenginį specifikacija. Specifikacijos kūrimo metu nustatyti pagrindiniai tikslai, kurių reikėtų siekti, kuriant tokio tipo sąsajas. Remiantis šiais tikslais, sukurta programinės įrangos sistema.
- 5.2. Siekiant ištirti automatinio atpažinimo algoritmo principus ir sukurti veikiantį algoritmą, atliktas automatinio atpažinimo algoritmo tyrimas. Šio tyrimo rezultate suformuluotas ir realizuotas automatinio gestų atpažinimo algoritmas. Šis algoritmas sėkmingai vykdo gestų atpažinimą nereikalaujant gesto pradžios ir pabaigos fiksavimo.
- 5.3. Atlikus sukurto algoritmo našumo eksperimentinius tyrimus, nustatyta algoritmo naudojamų resursų problema. Esant dideliame įvestų gestų kiekiui, algoritmas per 1 s. spėja patikrinti ar nuskaityti iš išorinio jutiklio duomenys neatitinka 9 išsaugotiems gestams. Tolimesnė atitikimo paieška viršija 1 s reakcijos ribą. Pasiūlyta dvigubai sumažinti nuskaitomų duomenų kiekį didinant koordinačių fiksavimo intervalus. Realizavus šiuos pasiūlymus ir atlikus eksperimentinius tyrimus nustatyta, kad tiriamu atveju greitaveika pagerėjo apie 5 kartus (9 gestai patikrinami per 200 ms.), 1 s reakcijos riba pasiekama esant 23 įvestiems gestams. Tikrinamų duomenų kiekio sumažinimas neįtakojė gestų atpažinimo.

- 5.4. Trečiojo tyrimo metu buvo siekiama nustatyti koordinačių nuskaitymo tikslumą kintant aplinkos veiksniams ir atstumui tarp naudotojo ir jutiklio. Buvo nustatyta, kad esant mažam atstumui ir geriems aplinkos veiksniams (geras apšvietumas, neatspindinti apranga, nėra tiesioginių saulės spindulių kritimo į jutiklį), matavimų paklaida yra ganėtinai maža (0,325 %). Tikėtina, kad atlikus jutiklio kalibravimą specifinėms situacijoms, sukūrus dar geresnes aplinkos sąlygas (didelis naudotojo ir fono kontrastas ar kitos priemonės), šią paklaidą galima būtų dar labiau sumažinti. Tiriant jutiklio galimybes tamsoje, nustatyta, kad nepriklausomai nuo atstumo iki jutiklio, matavimų paklaida žymiai nekinta ir svyruoja 0,8-1,13 % ribose.
- 5.5. Remiantis trečiuoju tyrimu, atliktas neatpažintų gestų priežasties nustatymo tyrimas. Šio tyrimo metu eksperimentuota fiksuotomis aplinkos sąlygomis, siekiant nustatyti automatinio atpažinimo algoritmo spragas, dėl kurių tam tikrais atvejais gestai buvo nenuskaitomi. Šių eksperimentų metu nustatyta, kad tam tikri gestai buvo nenuskaitomi dėl sparčių naudotojo judesių. Suformuluota ir patikrinta spartaus atpažinimo algoritmo idėja. Papildomais eksperimentais įrodyta, kad sukurtas algoritmo papildymas išsprendžia nustatytą problemą.
- 5.6. Paskutinio tyrimo metu nustatyti neatitikimai aukšto lygio programinės įrangos standartams – kelių klasių prižiūrimumo indeksai pateko į blogo prižiūrimumo sritį. Tiriant programinės įrangos kokybę, rasti nesklandumai sėkmingai pašalinti ir priimti sprendimai dokumentuoti. Gautas rezultatas rodo, kad sukurta programinė įranga atitinka aukšto lygio sistemos reikalavimus, yra lengvai plečiama ir prižiūrima.

6. LITERATŪRA

- [1] C. Vogler, D. Metaxas. „A framework for recognizing the simultaneous aspects of american sign language. Computer Vision and Image Understanding“, 358–384 psl., Springer-Verlag, 2010.
- [2] F. Hofmann, P. Heyer, and G. Hommel. „Velocity profile based recognition of dynamic gestures with discrete hidden Markov models. Gesture and Sign Language in Human-Computer Interaction“, 81–95psl., Springer-Verlag, 1998.
- [3] Zoltán Prekopcsák, Péter Halácsy, Csaba Gáspár-Papanek. Design and development of an everyday hand gesture interface. [Žiūrēta 2011 10 16], prieiga internete:
< <http://dl.acm.org/citation.cfm?id=1409318>>
- [4] J. Nielsen. „Usability Engineering“, Academic Press, 1993.
- [5] Iason Oikonomidis, Nikolaos Kyriazis, Antonis A. Argyros. „Efficient Model-based 3D Tracking of Hand Articulations using Kinect.“ [Žiūrēta 2011 10 16], prieiga internete: http://www.ics.forth.gr/~argyros/mypapers/2011_09_bmvc_kinect_hand_tracking.pdf
- [6] Totilo, Stephen. "Natal Recognizes 31 Body Parts, Uses Tenth of Xbox 360 computing Resources". 2010, [Žiūrēta 2011 10 16], prieiga internete:
<<http://kotaku.com/5442775/natal-recognizes-31-body-parts-uses-tenth-of-xbox-360-computing-resources>>
- [7] Wilson, Mark; Buchanan, Matt. "Testing Project Natal: We Touched the Intangible", [Žiūrēta 2011 10 17], prieiga internete: < <http://gizmodo.com/5277954/testing-project-natal-we-touched-the-intangible>>
- [8] „Kinect: The company behind the tech explains how it works". [Žiūrēta 2011 10 17], prieiga internete: <<http://www.joystiq.com/2010/06/19/kinect-how-it-works-from-the-company-behind-the-tech/>>
- [9] Rob Knies. "Academics, Enthusiasts to Get Kinect SDK" [Žiūrēta 2011 10 19], prieiga internete: <<http://research.microsoft.com/en-us/news/features/kinectforwindowssdk-022111.aspx>>
- [10] „FxCop“ [Žiūrēta 2012 12 07], prieiga internete <http://www.microsoft.com/en-us/download/details.aspx?id=6544>

[11] Shyam R. Chidamber. „A metrics suite for object oriented design“ [Žiūrėta 2013 03 02],
<prieiga internete:
http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=295895&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D295895>