

ŠIAULIU UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

Andrius Tamašauskis

EKRANO VALDYMAS RANKŲ GESTAIS

Magistro darbas

Vadovas

prof. dr. G. Daunys

ŠIAULIAI, 2012

ŠIAULIU UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

TVIRTINU

Katedros vedėjas
prof. dr. G. Daunys

2012 06

Andrius Tamašauskis

EKRANO VALDYMAS RANKŲ GESTAIS

Magistro darbas

Vadovas

prof. dr. G. Daunys

2012 06

Recenzentas

doc. dr. D. Dervinis

2012 06

Atliko

RM-10 gr. stud.

A. Tamašauskis

2012 06

ŠIAULIAI, 2012

TURINYS

ĮVADAS	6
1. LITERATŪROS ANALIZĖ	7
1.1 Taikomų problemos sprendimų apžvalga.....	7
1.1.1 Pirštu rodoma kryptimi valdomas ekranas	7
1.1.2 Rankų gestais valdomas CAVE įrenginyje	9
1.1.3 „Kamera – pelė“ HCI sistema	11
1.2 Taikomų metodų apžvalga.....	12
1.2.1 Pradinis vaizdo apdorojimas.....	12
1.2.2 Objektų išskyrimui iš aplinkos	13
1.2.3 Objektų sekimas	17
2. METODINĖ DALIS	19
2.1 Kuriamos sistemos panaudojimo vizija	19
2.2 Apibendrintas sistemos modelis	20
2.3 Gijos ir duomenų perdavimas tarp jų.....	22
2.4 Vaizdo kadro gavimas, bei pradinis apdorojimas	23
2.5 Odos spalvos objektų išskyrimas	24
2.6 Rankos ir veido sekimas	26
2.7 Rankos gestų interpretavimas sistemos valdymui	27
3. PRAKTINIS TYRIMAS	29
3.1 Priemonių pasirinkimas	29
3.2 Eksperimentų atlikimo tvarka, bei numatytų tyrimų tikslas	29
3.3 Veido paieškos laiko tyrimas	30
3.4 Atskirų sistemos dalių kadro apdorojimo laikas.....	31
3.5 Sistemos darbo skirtinguose kompiuteriuose tyrimas	32
3.6 Sekimo nuokrypių priklausomybė nuo sekamos rankos judėjimo greičio ir pagreičio	33
3.7 Sekimo nuokrypių priklausomybė nuo sekamos rankos judėjimo trajektorijos	34
3.8 Pastebėti sistemos trūkumai	36
IŠVADOS	37
LITERATŪROS SĄRAŠAS	38
PRIEDAI	38

SANTRAUKA

Tamašauskis A., Ekranų valdymas rankų gestais. Signalų technologijos magistro darbas / mokslinis vadovas prof. dr. G. Daunys; Šiaulių universitetas; Technologijos fakultetas; Elektronikos katedra, - Šiauliai, 2012 – 40 p.

Ekranų valdymui iki šiol naudojama populiariausia valdymo priemonė - kompiuterinė pelytė. Ji populiariausia ne dėl to, jog patogiau, o dėl to, jog pigiau. Šiuolaikinės kompiuterio ir žmogaus sąveikos sąsajos tobulinamos siekiant sukurti kuo natūralesnes valdymo sąsajas. Sukurtų rankų gestais valdomų sistemų įranga brangi, todėl šiame darbe buvo siekiama sukurti sistemą, kuri rankų gestus sugebėtų išskirti iš paprasta internetine kamera gaunamo vaizdo. Darbe siekiama sukurti ekranų valdymo rankų gestais sistemos modelį, kurio pagalba būtų galima pakeisti pelytės valdymą į valdymą rankų gestais. Kuriant sistemą buvo panaudota gijų technologija, kurios pagalba optimaliau išnaudojami kompiuterio resursai. Odos spalvos objektai išskiriami HSV spalvų erdvėje panaudojant aptikto veido odos spalvos histogramą. Veidas aptinkamas Haar tipo požymių pagalba. Sistemoje panaudoti Kalmano filtrai, kurių pagalba nuspėjamos sekamos rankos padėties, tai sumažino judančios rankos „pametimo“ tikimybę. Realizacijoje buvo panaudota nemokama ir pakankamai sparti kompiuterinės regos algoritmų „OpenCV“ biblioteka. Sistema geba valdyti pelės žymeklio pozicionavimą bei kairiojo klavišo paspaudimą. Vidutinė sistemos kadro apdorojimo sparta ~ 52 kadrai (640 x 400 dydžio) per sekundę, o tai rodo jog sistema tinkama valdymui realiu laiku. Ši sistema nesudėtingai gali būti panaudojama ir įvairiems kompiuterinės regos algoritmų testavimams atlikti.

Sukurta sistema pateikiama blokinėmis schemomis, C++ programavimo kalba parašytu programos kodu, bei sukompiliuota programine įranga su pavyzdiniu vaizdo įrašu, kuris buvo naudojamas tiriant sistemą.

SUMMARY

Tamašauskis A., Screen Control by Hand Gestures. The master's Thesis of Signal Technologies / scientific advisor prof. dr. G. Daunys; Šiauliai University, Technology Faculty, Electronics Department, Šiauliai, 2012. – 40 p.

The most popular screen control device is still computer mouse. This device is popular not because it is comfortable, but because it costs a little. The interest of modern researches in human computer interaction interfaces is concentrated to natural human computer interaction. Hand control interaction devices cost a lot of money, so we decided to develop a system that could pick out hand gestures from low cost web camera frames. In this work screen control by hand gestures system model was developed, and it was made for changing screen control by mouse device to screen control by hand gestures. Thread technology was used to increase system performance. Skin color objects are extracted in HSV color space by detected face skin color histogram. Harr like features were used for Face detection. The Kalman filters were used to increase system tracking stability by predicting moving hand positions. OpenCV (Open Source Computer Vision Library) computer vision library was used for this system realization, because it has a lot of fast optimized real time image processing algorithms and it is free open source library. Our system can control mouse cursor position and left button event. The average processing speed of the frame is about 52 frames (640 x 400 size) per second. It means that this system is appropriate to work in real time. The system easily can be used for a various computer vision algorithms testing purposes.

Developed system is presented by flowcharts, C++ program code files, compiled software and input video record sample which was used for system testing.

PAVEIKSLŲ SĄRAŠAS

- 1.1 pav. Virtualaus jutiminio ekrano panaudojimas
- 1.2 pav. Rodomos krypties nustatymas
- 1.3 pav. Procentinė pataikymų kiekio priklausomybė nuo objekto dydžio
- 1.4 pav. Sistema apdorotas vaizdo kadras
- 1.5 pav. Gestų atpažinimu pagrįsta valdymo sąsaja testuojama su CAVE įrenginiu
- 1.6 Programos stadijos – (a) eliminuotas fonas (b) išskirta odos spalva (c) galvos ir rankų sritys
- 1.7 pav. HSV spalvų erdvės modelis
- 1.8 pav. Keturi Haar tipo požymiai
- 1.9 pav. Kaupiamasis vaizdas taške (x,y) naudojamas greitai apskaičiuoti Haar tipo požymį
- 2.1 pav. Bendrinė sistemos struktūra
- 2.2 Sistemos funkcinė diagrama
- 2.3 Sistemos struktūrinė diagrama
- 2.4 pav. Pirmosios gijos algoritmo blokinė schema
- 2.5 pav. Kalibravimo lenta
- 2.6 pav. „Šeimininko“ inicializavimo procedūra
- 2.7 pav. Odos spalvos objektų išskyrimas
- 2.8 pav. Kadro apdorojimo laiko vykdant veido paieškas palyginimas
- 2.9 pav. Rankos identifikavimo algoritmo blokinė schema
- 2.10 pav. Rankos ir veido sekimo algoritmo blokinė schema
- 2.11 pav. Rankos gestų interpretavimo algoritmo blokinė schema
- 2.12 pav. Galimos kumščio padėtys
- 3.1 pav. Kadro apdorojimo laiko vykdant veido paieškas palyginimas
- 3.2 pav. Atskirų sistemos dalių kadro apdorojimo laiko grafikas
- 3.3 pav. Kompiuterių kadro apdorojimo laiko grafikas
- 3.4 pav. Nuokrypių priklausomybės nuo pagreičio grafikas
- 3.5 Nuokrypių priklausomybės nuo greičio grafikas
- 3.6 pav. Kumščio centro koordinatžių kitimo laike grafikas
- 3.7 pav. Nuokrypių kitimo laike grafikas
- 3.8 pav. Gautų nuokrypių tikimybinis pasiskirstymas
- 3.9 pav. Kumščio centro y koordinatės kitimo dvigubai pagreitintame vaizdo įrašė grafikas
- 4.1 pav. Sistemos praktinio panaudojimo vizija

IVADAS

Šiuo metu žmogaus ir kompiuterio sąveikos sąsajos sparčiai vystomos natūralaus kompiuterio ir žmogaus „bendravimo“ linkme. Šiuolaikiniai kompiuteriai kaip ir žmogus turi tuos pačius pagrindinius supančio pasaulio informacijos kanalus, t.y. vaizdo ir garso kanalus. Atliekama nemažai tyrimų siekiant „išmokyti“ kompiuterį suvokti pasaulį taip, kaip jį suvokia žmogus. Tam, kad neapkrauti kompiuterio sudėtingais vaizdų apdorojimo algoritmais naudojama speciali techninė įranga, kaip pavyzdžiu „Kinect“ naudoja infraraudoniesiems spinduliams jautrią kamerą ir apšviečia erdvę infraraudonųjų spindulių pluoštų tinkleliu, kurio pagalba nesudėtingai gaunamas erdvinis vaizdas. Tokia įranga pakankamai brangi ir ne kiekvienam prieinama, be to, žmogus mato paprastą spalvotą vaizdą, kaip ir kompiuteris vaizdo kameros pagalba, o vaizdo kameras turi beveik kiekvienas dabartinis kompiuteris. Šiame darbe siekiama sukurti sistemą, kurios pagalba žmogus rankų gestais galėtų valdyti kompiuterio pelytę panaudojant paprastą internetinę kamerą. Tai pakankamai sudėtingas vaizdo apdorojimo procesas kuris mano manymu bus naudojamas ateities žmogaus ir kompiuterio sąveikose. Siekiant šį procesą tobulinti reikalingas tikslingas sistemos pagrindas, t.y. sistemos modelis, todėl šio darbo **tikslas** - ekrano valdymo rankų gestais sistemos modelio sudarymas, o šiam tikslui pasiekti išskelti šie **uždaviniai**:

1. Apžvelgti naudojamus problemos sprendimo metodus;
2. Sudaryti sistemos modelį taikant rankų gestų atpažinimui skirtų sistemų analizę;
3. Realizuoti sukurta sistemos modelį ir patobulinti atliekant eksperimentus;
4. Išstirti galutinės sistemos spartumą, bei atsparumą trikdžiams.

Šio darbo rezultatus puikiausiai galima bus panaudoti tolimesniems panašių sistemų tyrimams, eksperimentams bei tobulinimui. Jaunieji Lietuvos mokslininkai pasinaudodami šios sistemos modeliu bei programiniu kodu nesudėtingai galės kurti ir tobulinti įvairius kompiuterinės regos algoritmus, nes nereikės visko kurti nuo pat pradžių. Sistema bus sukurta taip, kad būtų patogų redaguoti atskiras sistemos dalis.

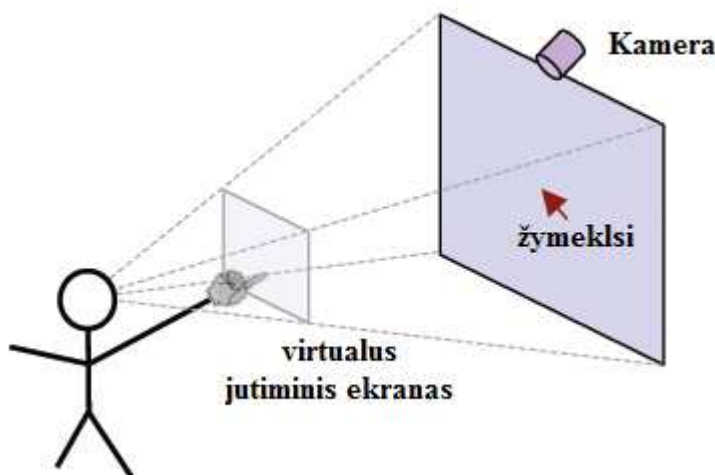
Tokios sistemos panaudojimas gali būti labai platus. Ją galima panaudoti pristatymų valdymui mokyklose, universitetuose įvairiuose konferencijose. Šiek tiek patobulinus sistemos gestų atpažinimą, sistemą būtų galima pritaikyti nebylių kalbos gestų mokymams ir t.t. Gestais valdomos sistemos mano manymu netolima ateitis, todėl su nedideliais pakeitimais šia sistemą būtų galima pritaikyti daugelyje sričių.

1. LITERATŪROS ANALIZĖ

1.1 Taikomų problemos sprendimų apžvalga

1.1.1 Pirštu rodoma kryptimi valdomas ekranas

Kelvinas Čengas (Kelvin Cheng) ir Masahiro Takasuka (Masahiro Takasuka) [1] iš Sidnėjaus (Australija) ITC nacionalinio universiteto pateikė piršto sąveikos su dideliu ekranu idėją.



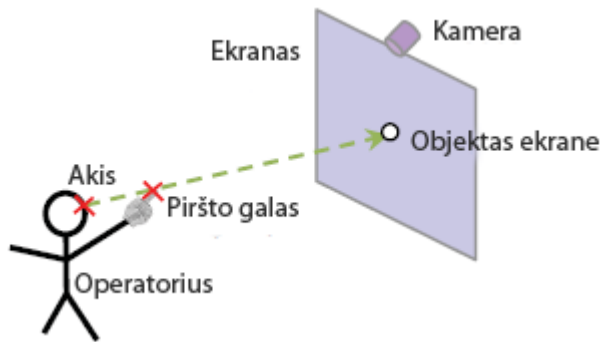
1.1 pav. Virtualaus jutiminio ekrano panaudojimas

Idėja gana paprasta, žmogus stovi priešais didelį ekraną (žr.1.1 pav.) (apie 1 – 2 m. atstumu nuo ekrano) ir ištiesęs ranką pirštu paliečia virtualų (nematomą) jutimini ekraną. Virtualus jutiminsis ekranas juda kartu su žmogumi, t. y. nuo žmogaus pagrindinės akies padėties projektuojama piramidė, kurioje įterpiamas virtualus jutiminsis ekranas rankos ilgio atstumu nuo žmogaus. Kamera pritvirtinta ekrano viršuje ties viduriu, tam, kad sekama ranka neužstotų veido.

Šios sistemos nukreipimo krypties (žr.1.2 pav.) nustatymą sudaro trys žingsniai:

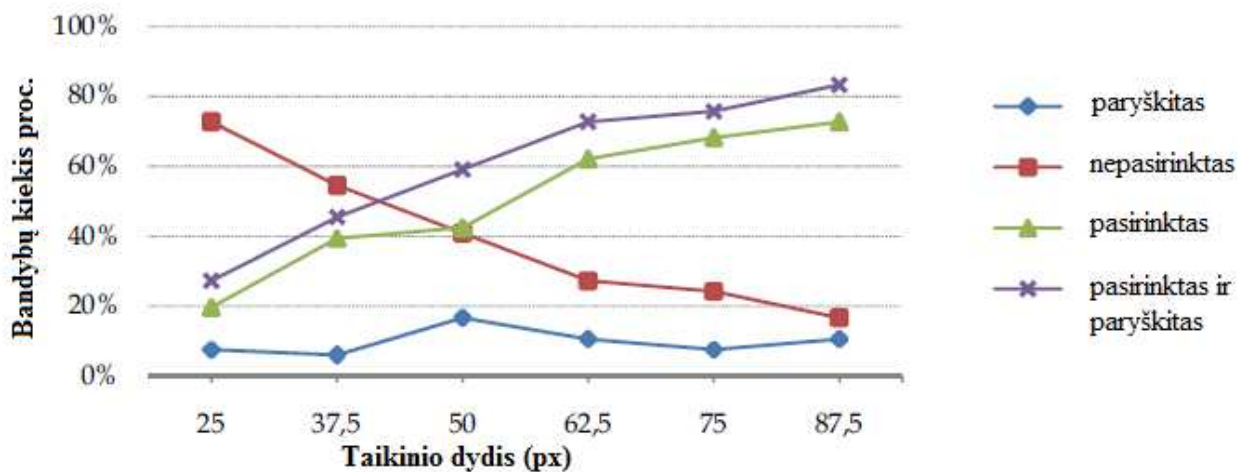
1. Akies pozicijos nustatymas
2. Piršto galo pozicijos nustatymas
3. Atstojamojo taško nustatymas

Veido aptikimas naudojamas pagrindinės akies radimui, be to, pagal veido plotį nustatomas atstumas iki kameros. Piršto galas randamas kaip tamsiausios odos spalvos taškas matomas iš kameros pozicijos. Piršto atstumas nuo žmogaus apskaičiuojamas pagal sferinę ištiesios rankos savybę, kur sferos centras yra žmogaus petys, o spindulys – žmogaus ranka. Akies sekimo stabilumas buvo padidintas Kalmano filtro pagalba, kuris šiuo metu plačiai naudojamas objektų sekimo algoritmuose.



1.2 pav. Rodomos krypties nustatymas

Šios sistemos tikslumą apriboja dauguma faktorių, todėl labai sunku nustatyti tikslias rodomo piršto koordinatas. Žmogaus kūno formos, bei judesių paklaidos. 1.3 paveikslėlyje pateikiama procentinio pataikymų, bei nepataikymų kiekio diagrama. Žmogus nukreipęs pirštą į objektą turi sustingti vienai sekunde. Jeigu žmogus rodo į tą patį objektą jau pasibaigus sustingimo laikui, objektas paryškinamas. Kaip matome, į mažus objektus gana sunku pataikyti dėl sistemos daromų aptikimo paklaidų. Kuo objektas didesnis tuo didesnė tikimybė, jog žmogus į jį pataikys.



1.3 pav. Procentinė pataikymų kiekio priklausomybė nuo objekto dydžio

Vienas iš pagrindinių šios sistemos privalumų – žmogus gali valdyti objektus per atstumą be jokių prietaisų ant rankų ar galvos. Manau tai labai perspektyvi sąveikos idėja, tačiau turi gana nemažai trūkumų, dėl kurių atsiranda nemažos paklaidos.



1.4 pav. Sistema apdorotas vaizdo kadras

Kaip matome (žr.1.4 pav.) šios sistemos apdorotame vaizdo kadre yra aptikta žmogaus galva, akis, bei rodomasis pirštas. Pasviręs mėlynas keturkampis atvaizduoja virtualaus jutiminio ekrano kontūrą.

Šiame straipsnyje aprašytame prototipe naudojama brangi kamera, kurios pagalba autoriams pavyko pakankamai tiksliai išskirti akies ir piršto galo padėtis. Su paprasta internetine kamera tai padaryti yra beveik neįmanoma.

1.1.2 Rankų gestais valdomas CAVE įrenginyje

Kompiuterių ir komunikacijų inžinerijos tarptautiniame žurnale 2012 metais paskelbtas straipsnis glaudžiai susijęs su šio darbo tema [2][5][7]. Autoriai sukūrė kompiuterinės regos sąsajos karkasinę biblioteką (framework) skirtą supančių projekcijų sistemai, kuri realizuota CAVE įrenginyje (žr.1.5 pav.).



1.5 pav. Gestų atpažinimu pagrįsta valdymo sąsaja testuojama su CAVE įrenginiu

CAVE įrenginys buvo sukurtas virtualios realybės tyrimams, bei eksperimentams atlikti. Autorių teigimu tai vienas iš populiariausių virtualios realybės tyrimams naudojamų įrenginių. Šio įrenginio konstrukcija panaši į kubą, kuriame priekinė ir šoninės sienos bei grindys rodo erdvinio vaizdo projekcijas. Žmogus stovintis kubo centre jaučiasi lyg būtų patekęs į virtualią realybę. Iki šiol ši sistema buvo valdoma įvairiausiais valdymo prietaisais kurie autorių manymu jau visiems įgriso, todėl jie nusprendė panaudodami kompiuterinės regos technikas sukurti gestais valdomą vartotojo sąsają. Gestų atpažinimui buvo panaudotas nešiojamas kompiuteris su integruota internetine kamera, mano manymu tai labai originalus sprendimas, nes nešiojamą kompiuterį dabar turi beveik kiekvienas.

Nešiojamo kompiuterio internetine kamera gautą vaizdą sistema apdoroja autorių sukurtu algoritmu ir interpretuotas komandas siunčia per bevielį kompiuterinį tinklą į projekcijas valdanti

kompiuterį. Sukurtos sistemos pagalba žmogus rankų gestais gali pasukti, patolinti arba priartinti vaizdą rodomą projekcijoje. Autoriai sukūrė šią sistemą siedami pakeisti jau įgriusius žaidimams valdyti skirtus įrenginius (pirštines, kepures ir pan.), kurie varžo žmogaus judesius ir sukelia diskomfortą, be to, tokie įrenginiai gana nemažai kainuoja ir naudojami dėvisi.

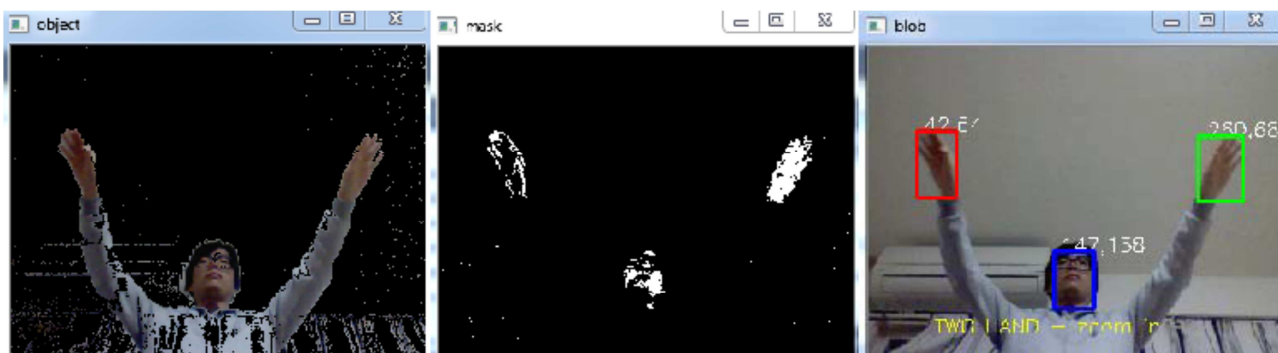
Šioje sistemoje gestams atpažinti panaudotas judesių šablonų metodas, kuris yra pakankamai nejautrus apšvietimo kitimams, nes interpretuojami rankos judesiai o ne rankos plaštakos formos. Šios sistemos vykdymo etapai:

- Gaunamas fono vaizdas (pilkas)
- Gaunamas dabartinis kadras (sumažinamas ir apverčiamas)
- Išskiriamas objektas iš fono
- BGR spalvų erdvės transformavimas į HSV spalvų erdvę
- Odos spalvos išskyrimas
- Triukšmų eliminavimas
- Sričių atpažinimas (pagal dydį)
- Rankų gestų atpažinimas (vienos rankos ir dviejų rankų)

Pradinėje sistemos stadijoje gaunamas fono vaizdas, kuris padeda eliminuoti foną iš kitų kadrų. Po to gaunami kadrai sumažinami per pus sistemos darbo greičio padidimui. Kadras kuriame atskirtas žmogaus vaizdas (žr.1.6.a pav.) nuo fono konvertuojamas į HSV spalvų erdvę. Šios spalvų erdvės pagalba išskiriamos odos spalvos spalva (žr.1.6.b pav.). Odos spalvos išskyrimui naudojamos šios ribos:

- H kinta nuo 0 iki 35
- S kinta nuo 52 iki 173
- V kinta nuo 90 iki 255

Gautas vaizdas triukšmingas, todėl triukšmai eliminuojami morfologiniais filtrais. Gautos sritys sunumeruojamos ir išrikiuojamos pagal dydį. Per mažos ir per didelės sritys automatiškai eliminuojamos, taip atrenkamos galvos ir rankų sritys (žr. 1.6.c pav.).



1.6 Programos stadijos – (a) eliminuotas fonas (b) išskirta odos spalva (c) galvos ir rankų sritys

Šioje sistemoje gestai suskirstyti į: vienos rankos ir dviejų rankų gestų kategorijas. Vienos rankos kategorijai priklauso dešinė ir kairė komandos. Dviejų rankų kategorijai priklauso artyn ir tolyn komandos. Komandos interpretuojamos sekant rankų centrų padėčių kitimą tarp kadru.

Jeigu rankos viena nuo kitos tolėja vaizdas artinamas ir atvirkščiai, jeigu rankos viena prie kitos artėja vaizdas tolinamas. Jeigu aptikta tik viena ranka, tuomet vaizdas sukamas į tą pusę į kurią juda ranka.

Šiame straipsnyje aprašytas puikus rankų gestais valdomos sistemos pavyzdys. Sistema realizuota panaudojant nemokamą atviro kodo OpenCV biblioteką, be to, ji dirba realiu laiku. Naudojama techninė įranga (nešiojamas kompiuteris) yra beveik kiekvienam prieinama. Straipsnyje aiškiai išskiriamos trys sistemos sritys: pradinis vaizdo apdorojimas, rankų ir veido išskyrimas, rankos gestų interpretavimas.

1.1.3 „Kamera – pelė“ HCI sistema

„Kamera – pelė“ HCI (*ang.* human computer interaction) sistema - tai sistema, kuri kameros pagalba seka vartotojo judesius ir paverčia juos į kompiuterio pelytės žymeklio judesius. Šioje sistemoje bet kokia žmogaus kūno dalis (lūpos, nosis, pirštas ir pan.) gali būti sekama pelytės valdymui [9].

Bostono universitete buvo atliktas šios sistemos tyrimas, kuriame buvo siekiama išsiaiškinti su kuria kūno dalimi žmogus greičiausiai parašys žodžius („RAINING“, „MINIMAL“, „POOR“) „SpeechStaggered“ programa, nupieš liniją tarp dviejų rodomų taškų „EaglePaint“ programa, bei sužais žaidimą „KiteGame“, kuriame reikia pasirinkti du vienodus aitvarus. Valdymo objektas buvo nurodomas pelės paspaudimu.

Eksperimentui buvo pasirinkti šie valdymo taškai:

- Nosies galas
- Apatinė lūpos vidurys
- Kairės akies vidinis kraštas
- Nykščio vidurys

Atlikti eksperimentai parodė, jog greičiausiai sužaistas žaidimas „KiteGame“, bei nupiešta linija akies krašto pagalba. Žodžiai greičiausiai parašyti nosie galo pagalba, tačiau lėčiausiai ir tolydžiausiai visi veiksmai buvo atlikti nykščio pagalba.

Iš šio tyrimo matyti, jog nykščiu (ranka) atliekami veiksmai yra stabiliausi, nosimi – taikliausi, o akies kraštu – greičiausi. Mano manymu kuriama rankų gestais valdoma sistema turėtų pasižymėti stabilumu.

1.2 Taikomų metodų apžvalga

Šiame darbe siekiama sukurti demonstracijų, rodomų dideliuose ekranuose, valdymo modelį kuris pakeistų jau įgriusį ir nepatogų valdymą kompiuterio pelyte. Panaudojant kompiuterinės regos technikas bus siekiama sukurti realaus laiko valdymo sistemą kurią būtų galima pritaikyti dideliuose ekranuose rodomų prezentacijų valdymui nenaudojant prietaisų pritvirtinamų prie žmogaus.

Bendrą sistemos modelį galime suskirstyti į šiuos funkcinis blokus:

- Pradinis vaizdo kadru apdorojimas
- Rankų išskyrimas iš aplinkos bei sekimas
- Rankos formos ir trajektorijų pavertimas į operacinės sistemos valdymo komandas

Kiekvienam iš šių blokų išskiriamos savitos problemos.

Pradinis vaizdo apdorojimas labai priklauso nuo naudojamos kameros. Jeigu iš kameros gaunami kadrai turi radialinius iškraipymus, juos būtinai reikia eliminuoti, tačiau yra tokių kamerų kurios tokių iškraipymų nesukuria arba sukuria labai mažus, kurie turi labai mažą įtaką sistemos darbui. Tokiu atveju radialinių iškraipymų eliminavimo galime ir atsisakyti, nes kiekvienas vykdomas procesas sistemoje išnaudoja atitinkama kiekį kompiuterio resursų. Siekiant sukurti realaus laiko valdymo sistemą labai svarbu sukurti sistemą, kuri kompiuterio resursus naudotų kuo optimaliau.

Rankų išskyrimui iš aplinkos labai svarbu pasirinkti tinkamą objektų išskyrimo iš aplinkos metodą. Metodas turi naudoti kuo mažiau kompiuterio resursų, tačiau kuo greitesnė sistema, tuo ji paprastesnė ir nestabilėsnė. Todėl reikia pasirinkti tokį metodą, kuris pakankamai gerai išskirtų rankos objektus ir naudotų kuo mažiau kompiuterio resursų.

Kitas labai svarbus reikalavimas šiai sistemai – kaina. Siekiama sukurti produktą kurį galėtų naudoti ir tobulinti kiekvienas nešiojamą kompiuterį turintis studentas, todėl parenkant techninę ir programinę įrangą į tai būtina atsižvelgti.

1.2.1 Pradinis vaizdo apdorojimas

Pradinis vaizdo apdorojimas būtinas tuomet, kuomet iš vaizdo šaltinio gaunami kadrai su iškraipymais. Vaizdo kameros, ypač internetinės, savyje turi daug integruotų automatinių funkcijų, kurios didina sistemos nestabilumą. Šias funkcijas galime valdyti kameros tvarkyklių pagalba.

Kitas iškraipymų šaltinis – kameros lęšių sistema. Kameros lęšių sistema sukuria radialinius iškraipymus, kuriuos galime eliminuoti pasinaudodami kalibravimo procedūra. Procedūra gan nesudėtinga. Priešais kamerą kilnojama žinomo dydžio „šachmatų lenta“. Sistema suranda kiekviename kadre šachmatų lentos kvadratėlių kampus tarp kurių būtinai turi būti tiesės.

Gavusi pakankamą kiekį informacijos sistema deformuoja vaizdą taip, kad „šachmatų lentų“ gauti kampai sudarytų tieses. Taip gaunama kiekvieno kadro taško poslinkio vektorių matrica, kitaip dar vadinama iškraipymų matrica. Ši matrica išsaugoma kompiuterio atmintyje tolimesniam naudojimui.

Kuomet turima iškraipymų matricą gauti nauji kadrai transformuojami prie kiekvieno kadro taško pridėdant poslinkio vektorių.

1.2.2 Objektų išskyrimas iš aplinkos

Šiame poskyryje aptarsime populiariausius objektų išskyrimo iš aplinkos kompiuterinės regos metodus. Objektų išskyrimas atlieka labai svarbų vaidmenį kuriamoje sistemoje. Jeigu objektai bus išskiriami netinkamai, sistema dirbs nestabiliai arba išvis nesusidoros su užduotimi. Pasirinktas metodas privalo būti spartus ir pakankamai tikslus, tam kad sistema sugebėtų dirbti realaus laiko režimu. Štai keletas pagrindinių objektų iš aplinkos išskyrimo metodikų:

1. **Fono eliminavimas.** Jei objektas fotografuotas monochromatiniame fone, pakanka paprasčiausiai eliminuoti monochromatinį foną. Objektus taip pat nesunku išskirti, jei fotografuota statiniame (nekintame) žinomame fone. Foną dar galime išskirti iš nejudamos kameros filmuojamo vaizdo, žinoma jei tik fonas nesikeičia. Internetinės kameros vaizdas labai triukšmingas, todėl fonui išskirti naudojami papildomi filtravimai.
2. **Paieška pagal spalvą.** Pastebėta, kad žmogaus odos spalvos išsidėsčiusios kompaktiškai. Tiksliau trimačiame spalvų kube yra kompaktiškos spalvų zonos, kurioms priklauso daugeliu atvejų veidą atitinkantys vaizdo pikseliai. Deja ši metodika jautri apšvietimo sąlygoms priklausomai nuo pasirinktos spalvų erdvės modelio ir visiškai netinka pilkiems vaizdams.
3. **Paieška pagal judesį.** Daugeliu atvejų vaizdo įrašuose žmones atitinkantys objektai juda. Šiuo atveju pakanka taikyti bendras judesio sekimo technikas. Ši metodikai nelabai efektyvi tuo atveju kai scenos yra sudėtingos ir juda daug objektų, arba priešingai - žmogus ilsisi ar miega.
4. **Hibridiniai metodai.** Kadangi išvardinti metodai gana nepriklausomi, juos galima efektyviai kombinuoti. Visuomet galima juos apjungti paprasčiausiai taikant kiekvieną metodą atskirai, kaupti rezultatų statistiką ir juos derinti.
5. **Nespalvotų nejudančių objektų išskyrimas.** Šis metodas yra sudėtingiausias, tačiau universalus, nes tinka ir kitais atvejais (spalvotiems, judantiems vaizdams). Šiam tikslui pasiekti naudojami šie metodai:
 - Dirbtiniais neuroniniais tinklais pagrįsti metodai
 - Statistiniai klasterizavimo metodai

- Veido modeliu pagrįsti metodai
- Palyginimo su etalonais metodai

Nespalvotų nejudančių objektų išskyrimo metoduose naudojama sukaupta informacija apie objekto tam tikras savybes. Dažniausiai ši informacija gaunama mokymo metodais. Tai pakankamai tiksli ir sparti metodika, tačiau netinkama bendriniais atvejais. Kadangi šiame darbe bus siekiama išskirti rankas ir veidą iš realiu laiku filmuojamo spalvoto vaizdo, pasirinktas paieškos pagal spalvą metodas, kuris yra pakankamai spartus, bei tinkamas bendriniais atvejais.

1.2.2.1 Odos spalvos išskyrimui naudojamų spalvos erdvių palyginimas

Iliojaus universitete buvo atliktas odos spalvos išskyrimo priklausomybės nuo spalvos erdvės tyrimas, kuriame buvo tiriamos penkios spalvos erdvės:

- CIELAB
- Fleck HS
- HSV
- Normalizuota RGB
- $Y_C C_b$

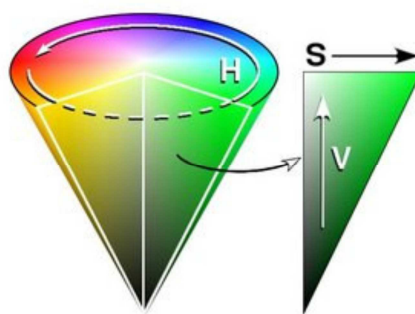
Algoritmuose naudojami Bajeso klasifikatoriai buvo apmokomi 48 – iais paveikslėliais, o bandymai atliekami su 64 – iais paveikslėliais, kurie buvo paimti iš filmų, televizijos laidų bei žurnalų. Buvo parenkami paveikslėliai visiškai atsitiktinai, todėl į sistemą paduodamų kadru vaizdai buvo su skirtingais apšvietimais ir plačiu odos spalvos diapazonu.

Tyrimas parodė, jog ne odos spalvos taškų aptikimo paklaidos mažėja tuomet, kuomet odos spalvos aptikimo paklaidos didėja ir odos spalvos taškų kiekis vaizde mažėja. Paieškos lentelėje metodas geriausiai susitvarkė su odos spalvos išskyrimo užduotimi Fleck HS ir HSV spalvų erdvėse, todėl kuriamos sistemos odos spalvos išskyrimui pasirenkame HSV spalvų erdvę.

1.2.2.2 Odos spalvos išskyrimas HSV spalvų erdvėje

HSV spalvų erdvė (žr. 1.7 pav.) susideda iš trijų komponenčių: H – atspalvis; S – įsotinimas; V – kiekis arba intensyvumas. Atspalvis apibūdina raudoną, žalią, purpurinę spalvą, įsotinimas – kiek spalva nutolusi nuo to paties intensyvumo pilkos spalvos, o kiekis apibūdina šviesumą. Šis spalvos erdvės modelis naudingas odos spalvos išskyrimui tuo, jog dinamiškai filtruojamas tik spalvos atspalvis, dėl ko odos spalvos išskyrimo procesas tampa pakankamai spartus. Spalvų reikšmių kitimas labai mažas kintant apšvietimui, todėl prie mažų apšvietimo pokyčių spalvų reikšmės kinta labai mažai. Kaip tuo tarpu RGB spalvų erdvėje kintant apšvietimui

spalvų reikšmės kinta labai stipriai ir nepriklausomai, todėl RGB spalvų erdvėje labai sunku išskirti norimos spalvos objektus kintant apšvietimui.



1.7 pav. HSV spalvų erdvės modelis

Odos spalvos objektų išskyrimo procese S ir V reikšmės atrenkamos pasirinktomis statinėmis slenkstinėmis vertėmis:

- S kinta nuo 52 iki 173
- V kinta nuo 90 iki 255

O tuo tarpu atspalviui išskirti naudojamos histogramos. Sistemos odos atspalvį išskiria odos atspalvio reikšmių tikimybinio pasiskirstymo, kitaip dar vadinamo histograma, pagalba. Turėdami odos atspalvio histogramą galime nustatyti kiekvieno kadro taško odos spalvos tikimybę. Šią histogramą galima gauti keliais būdais. Vienas iš paprasčiausių – rankinis odos spalvos objekto pasirinkimas. Pelės žymeklio pagalba kadre pažymimas odos spalvos plotas, tuomet suskaičiuojamos kiekvienos spalvos suminės reikšmės. Taip gaunamas odos spalvos histograma pažymėtam plotui. Šis metodas nereikalauja išankstinio apmokymo, kaip pavyzdiumi naudojant Bajeso klasifikatorių. Po histogramos apskaičiavimo kiekvieno taško atspalvio reikšmė sudedama su histogramoje gauta atitinkama reikšme. Nustatant slenkstinę ribą atrenkami taškai kurie „tikrai“ yra odos spalvos ir kurie nėra. Taip gaunamas binarinis kadras.

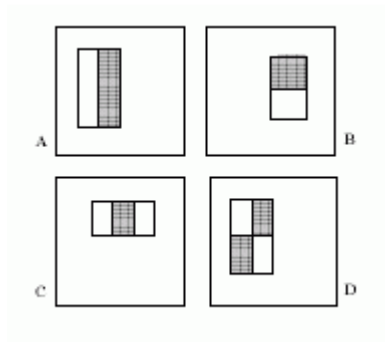
Gauti odos spalvos taškai filtruojami, t.y. išmetami pavieniai taškai, bei užpildomos mažos skylutės. Po filtravimo taškai sugrupuojami į sritis kurios taip pat filtruojamos paliekant tik didžiausias sritis.

Vienas iš esminių šio metodo trūkumų – rankinis odos spalvos srities pasirinkimas. Kadangi didžiausias odos spalvos objektas esantis kadre turėtų būti veidas, jį galime aptikti Haar tipo požymių klasifikatoriaus pagalba. Šis metodas pakankamai tikslus ir greitas, be to, veido aptikimas vykdomas nespaltoto vaizdo kadruose. Aptikus veidą minėtu metodu galima iš veido srities gauti odos spalvos histogramą, kurios pagalba išskirsime odos spalvos objektus.

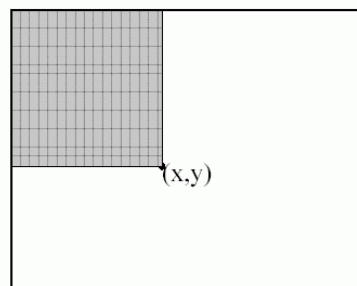
1.2.2.3 Veido atpažinimas Haar tipo požymiais

Trumpai aprašysime gerai pasiteisinusią veidų ir rankų išskyrimo metodiką, kuri yra viena iš mažiausiai kompiuterio resursų reikalaujančių metodikų, t. y. objektai aptinkami greit ir gana

tiksliai. Ši metodika vadinama jos autorių P. Viola ir M. J. Jones vardu [13]. Viola ir Jones pasiūlė skaičiuoti labai paprastus ir greitai apskaičiuojamus parametrus. 1.8 paveikslėlyje iliustruojami šių požymių keturi pagrindiniai tipai. Paveikslėlio šablonai balta spalva žymi pozicijas, kuriose esantys vaizdo pikseliai sumuojami su pliuso ženklu, o pilka spalva žymi pozicijas, kuriose esantys pikseliai sumuojami ir paimami su minuso ženklu. Požymių sudaro srityse esančių pikselių sumos.



1.8 pav. Keturi Haar tipo požymiai



1.9 pav. Kaupiamasis vaizdas taške (x,y) naudojamas greitai apskaičiuoti Haar tipo požymį

Greitam požymių apskaičiavimui naudojamas "kaupiamasis vaizdas". 1.9 paveikslėlyje iliustruojamos jo reikšmės taške (x,y) , kuris gaunamas sumuojant vaizdo reikšmes pilkame stačiakampyje. Kaupiamąją sumą galima apskaičiuoti ir išsaugoti masyve, kurio matmenys bus tokie patys kaip ir pradinio vaizdo. Tam, kad būtų išskirti tinkamiausi požymiai klasifikacijai, autoriai naudojo Adaboost apmokymo algoritmą.

Šie algoritmai realizuoti OpenCV bibliotekoje, be to, bibliotekos kūrėjai pateikia jau apmokytų pavyzdinių klasifikatorių naudojamų veidams bei rankoms išskirti.. Pastebėta, jog šis metodas gan tiksliai ir greitai randa veidą kadre, tačiau veidas turi būti pasuktas tiesiai į kamerą. Kad ir koks šis algoritmas būtų greitas, judantis veidas keičia padėtį ir dauguma atvejų gali būti neaptinkamas. Kaip žinome pati ilgiausia paieška trunka tuomet, kuomet nieko nerandama, todėl kiekviename kadre ieškoti veido būtų netikslinga. Idėja būtų gana paprasta. Veidą galime aptikti seanso pradžioje ir identifikuoti odos spalvos objektą kaip veidą. Kuomet veidas aptiktas, sekame tik odos spalvos objektą. Galime įvesti tam tikras sąlygas, kurios informuotų sistemą apie veido

objekto praradimą. Praradus veido objektą sistema vėl pradėtų ieškoti veido šiuo metodu. Tam, kad paieška užtruktų dar trumpiau, galime veido ieškoti odos spalvos srityse pradedant nuo didžiausios srities.

1.2.3 Objektų sekimas

Aptiktų objektų sekimui gali būti panaudota keletas skirtingų algoritmų [15]. Visi jie charakterizuojami sekančių veiksmų realizavimo būdu: nuoseklus sričių išskyrimas, gautos informacijos minimizavimas ir pasikeitimu sekimo aprašymas. Du paskutiniai etapai liečia duomenų pateikimo klausimą ir yra skirtingi beveik visose realizacijose. Pirmam etapui egzistuoja keletas technologijų. Visų pirma būtina išskirti analizuojamo objekto vaizdą nuo fono vaizdo. Sprendžiantys šį uždavinį algoritmai skirstomi į laikinius ir erdvinius. Laikiniai algoritmai remiasi kelių kadrų palyginimu. Vienas iš greičiausių šiuo metu naudojamų laikinų metodų – kameros poslinkio (CAMSHIFT) metodas.

1.2.3.1 Kameros poslinkio metodas

Kameros poslinkio (CAMSHIFT) metodas - tai patobulintas populiarus vidurkio poslinkio (Mean-Shift) metodas. Vidurkio poslinkio metodas modifikuotas taip, kad tikrų dinamiškai kintančiam laike spalvos tikimybiniam pasiskirstymui sekti [6]. Štai keletas šio metodo charakteristikų:

- 1) Sugeba sekti objektą realiu laiku
- 2) Gali būti naudojamas be kameros lęšio kalibravimo proceso, nes sekamas objektas ne pagal formą, o pagal objekto spalvų pasiskirstymą.

Trumpai aprašysime šio metodo algoritmą:

1. Pasirenkamas objektas kurį norime sekti
2. Sukuriama objekto spalvų pasiskirstymo histograma, kuri apibūdina objekto modelį
3. Apskaičiuojamos kadro pikselių tikimybės panaudojant gautą spalvų histogramą
4. Apskaičiuotų tikimybių kadre randamas paieškos lango masės centras panaudojant vidurkio poslinkio metodą
5. Centruojamas paieškos langas ties 4 – am žingsnyje rastu tašku iteracijų metodu
6. Apdorojamas sekantis kadras su 5 – am žingsnyje gauta paieškos lango pozicija

Šis metodas skirtas vieno objekto sekimui. Jeigu panašios spalvos objektas priartėja prie sekamo objekto paieškos langas padidėja. Sistema tai supranta kaip sekamo objekto padidėjimą, todėl veido ir rankų sekimui viename kadre teks patobulinti sekimo algoritmą.

1.2.3.2 Kalmano filtras

Kalmano filtras – tai rekursinis filtras, kuris nuspėja tiesinio dinaminio proceso būseną pagal matavimus, kurie gaunami iš triukšmingų matavimų. Jis pavadintas Rudolfo E. Kalmano garbei, kuris 1960 m. paskelbė straipsnį apie šį filtrą. Kalmano filtro darbą kiekviename žingsnyje galima suskirstyti į dvi fazes: prognozavimo ir korekcijos [14].

Prognozavimo fazėje prognozuojama būsena nustatoma kaip ankstesnės būsenų kaitą nusakančios matricos ir ankstesnės pakoreguotos prognozuojamos būsenos sandauga, pridėdant reakciją į įėjimus nusakančios matricos ir įėjimo sandaugą:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_{k-1} \mathbf{u}_{k-1}.$$

Prognozuojamos būsenos kovariacija nustatoma iš ankstesnės pakoreguotos prognozuojamos kovariacijos, ankstesnės būsenų kaitų matricos ir triukšmo kovariacijos:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_{k-1}.$$

Paklaidos kovariacija gaunama iš matavimo modelį nusakančios matricos, prognozuojamos būsenos kovariacijos ir triukšmo kovariacijos:

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k.$$

Optimalus stiprinimo koeficientas nustatomas iš prognozuojamos būsenos, matavimo modelio ir paklaidos kovariacijos:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}.$$

Pakoreguota prognozuojama būsena apskaičiuojama pridėdant prie prognozuojamos būsenos optimaliai sustiprintą paklaidą:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k.$$

Pakoreguota prognozuojama kovariacija gaunama iš atitinkamų matmenų vienetinės matricos, optimalaus stiprinimo, matavimo modelio ir prognozuojamos kovariacijos:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}.$$

Šio filtro pagalba galėsime sumažinti triukšmo įtaką sekimo fazėje prognozuodami natūralaus rankos judesio trajektorijas.

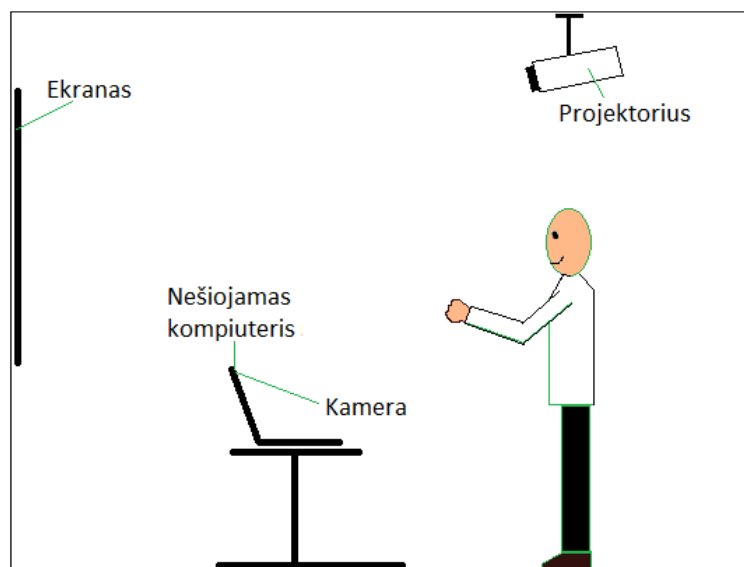
2. METODINĖ DALIS

2.1 Kuriamos sistemos panaudojimo vizija

Tam, kad būtų aiškesnė šio darbo idėja šiame skyrelyje trumpai apžvelgsime kuriamos sistemos panaudojimo vizijas. Ši sistema gali būti panaudojama ir kaip ekrano valdymo priemonė ir kaip priemonė kompiuterinės regos algoritmų tobulinimas bei tyrimas atlikti.

Kuriamos sistemos panaudojimo vizija iliustruojama 4.1 paveikslėlyje. Kaip matome šioje iliustracijoje pavaizduoti prietaisai dažniausiai jau yra naudojami, tačiau ekrano valdymui naudojama pelytė, o ne rankų gestai, kuriuos galime išgauti iš nešiojamo kompiuterio kameros vaizdo.

Viena iš esminių priežasčių kodėl to dar nėra, tai – tokia kamara gautą vaizdą apdoroti yra pakankamai sudėtinga, dėl ko naudojant standartinius kompiuterinės regos algoritmus kadro apdorojimo laikas trunka daugiau nei sekundę. Kompiuterio procesorius tokiu atveju dirba pilnu pajėgimu ir stabdo kitas programas. Norint to išvengti reikia optimizuoti vykdomus procesus, tačiau reikia optimizuoti tiek, kad sistema dirbtų ir pakankamai tiksliai ir pakankamai greitai. Kaip žinome, kuo labiau optimizuoti bus procesai, tuo mažesnis bus sistemos stabilumas bei tikslumas.



4.1 pav. Sistemos praktinio panaudojimo vizija

Šiame darbe bus siekiama sukurti sistemą, kurią būtų galima panaudoti šios vizijos realizacijai. Ar tai yra įmanoma parodys sukurtos sistemos tyrimai.

2.2 Apibendrintas sistemos modelis

Apžvelgiant rankų gestų valdymo modelį skirtą CAVE įrenginiui buvo išskirtos panašių sistemų pagrindinės trys sistemos sritys: pradinis vaizdo apdorojimas, rankų ir veido išskyrimas, rankos gestų interpretavimas. Kadangi sistemos realizavimui buvo pasirinkta C++ programavimo kalba, optimaliam sistemos resursų išnaudojimui buvo panaudotos gijos (Threads). Gijose veikiantys procesai vykdomi lygiagrečiai.

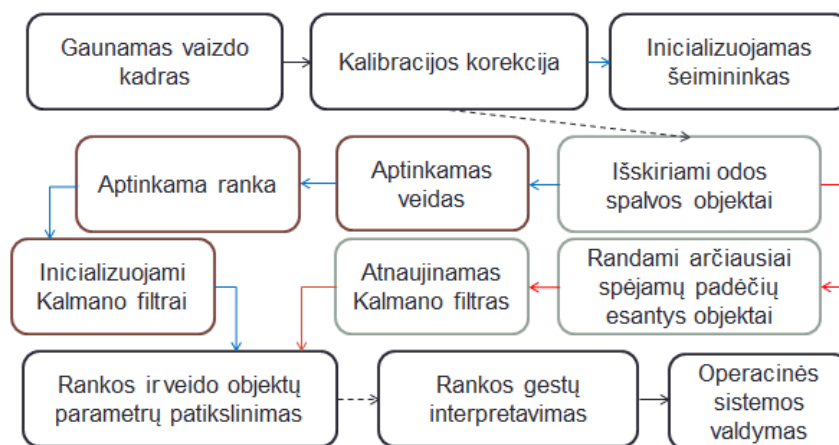


2.1 pav. Bendrinė sistemos struktūra

2.1 paveikslėlyje pavaizduota bendrinė sistemos struktūra. Sistemą sudaro trys gijos kurių inicializacijos tvarka pavaizduota rodyklėmis. Kuomet yra gautas vaizdo kadras pirmoje gijoje, antra gija gali pradėti dirbą, o kuomet yra gaunamos rankos objekto koordinatės antroje gijoje, trečioji gali pradėti dirbą.

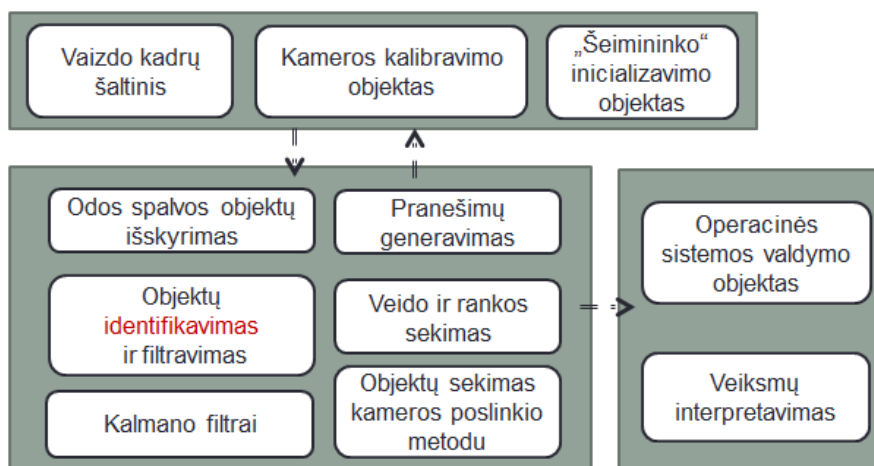
Funkcinėje sistemos diagramoje (žr. 2.2 pav.) iliustruojamas detalesnis sistemos darbas. Mėlynos spalvos rodyklė žymi inicializacijų kryptis. Odos spalvos išskyrimui reikalinga atlikti „šeimininko“ inicializacija, kurios pagalba aptinkamas kadre matomas didžiausias veidas ir gaunama to veido odos spalvos histograma, kuri naudojama odos spalvos objektams išskirti. Antroje gijoje sistema dirba pilnu pajėgumu tik tuomet kai yra aptiktas veidas ir ranka. Tik tuomet kai veidas atsiranda aptiktų objektų sąrašė galima ieškoti rankos. Na žinoma, kuomet aptikta ranka ir veidas sukuriami atitinkami Kalmano filtrai ir sistema pereina į objektų sekimo stadiją. Dingus objektui kartu su juo sunaikinami ir jam priklausantys Kalmano filtrai.

Sistema sekimo stadijoje išskiria odos spalvos objektus sekančiame kadre ir ieško odos spalvos objektų esančių arčiausiai Kalmano filtrais spėjamų padėčių. Ieškant artimiausių objektų įvertinamas ir objekto dydžio pokytis, t.y. objektas negali labai greit padidėti arba sumažėti. RADIUS rankos ir veido atitikmenis atnaujinami objektų, bei Kalmano filtrų parametrai naujomis koordinatėmis, aukščiu, pločiu, bei pasvirimo kampu.



2.2 Sistemos funkcinė diagrama

Kuomet aptikta ranka ir veidas apskaičiuojama aktyvios zonos padėtis bei dydis veido atžvilgiu. Aktyvios zonos ir rankos objektai siunčiami į rankos gestų interpretavimo bloką, kuriame identifikuojami tam tikri rankų getai vykdomi aktyvioje zonoje. Identifikuotos komandos siunčiamos Windows operacinei sistemai. Na šias komandas taip pat galime siųsti kompiuteriniu tinklu į kitą kompiuterį.



2.3 Sistemos struktūrinė diagrama

Informacijos pasikeitimas tarp gijų vykdomas signalais. 2.3 paveiksle punktyrinėmis rodyklėmis iliustruojamos signalų kryptys. Pirmoji gija siunčia sukalibruotus kadrus bei „šeimininko“ odos spalvos histogramą į antrąją giją tuomet, kuomet antroji gija šių duomenų paprašo atitinkamais signalais. Žinoma jeigu nėra ko siųsti (nėra duomenų) siunčiami klaidų signalai. Tam, kad sistema neapdorotų kadro kelis kartus, o taip gali nutikti tuomet, kuomet sistema kadra apdoroja per trumpesnį laiką negu gaunamas naujas kadras, įvedamas kadro identifikatorius.

Taip pat 2.3 paveiksle pavaizduota supaprastinta sistemos struktūra. Išskirti tik pagrindiniai sistemoje esantys objektai, bei procedūros.

2.3 Gijos ir duomenų perdavimas tarp jų

Paprasčiausiai programos gijos galima būtų įsivaizduoti kaip atskirus programos „darbininkus“, kurie atlieka darbus nepriklausomai vienas nuo kito. Pagrindinė programa sukuria darbininkus ir duoda jiems nurodymus signalais. Signalus gali siųsti ir darbininkai ir pagrindinė programa. Darbininkų signalai keliauja į pagrindinę programą, kurioje aprašyti signalų sujungimai su atitinkamomis operacijomis.

```
MainWindow wi //Pagrindinio lango objektas
OdosSpalva obj;//Sukuriamas odos spalvos išskyrimo objektas
QThread gija;//Sukuriamas gija
QThread::connect(&obj, SIGNAL(Init()), &w, SLOT(InitOdosSpalva()));
obj.moveToThread(&gija);
gija.start();
```

Kaip matome pateiktoje sutrumpintoje programos kodo ištraukoje programos pradžioje sukuriama pagrindinio lango (to kurį matome ir valdome) objektas „w“, po to sukuriama odos spalvos apdorojimo objektas „obj“, bei jam skirta gija „gija“. Kaip minėjau pagrindinėje programoje aprašomi signalų sujungimai su operacijomis. Šioje kodo ištraukoje sujungiamas „obj“ objekto signalas „Init()“ su pagrindinės programos operacija „InitOdosSpalva()“, tai reiškia, jog, kuomet „obj“ objektas išleis signalą „Init()“ pagrindinė programa įvykdys „InitOdosSpalva()“ operaciją. Po sujungimų aprašymo objektas įkeliamas į giją ir gija aktyvuojama („start()“). Tam, kad tarp gijų būtų galima perduoti nestandartinius duomenis reikia užregistruoti jų duomenų tipus, kaip pavyzdžiui:

„IplImage*“ - tai OpenCV paveikslėlio duomenų tipas. Norint jį išsiųsti signalo pagalba ši tipą reikia užregistruoti (`qRegisterMetaType<IplImage*>("IplImage*")`). Šiame darbe naudojamas asinchroninis duomenų apsikeitimas, todėl gija norėdama gauti vaizdo kadrą privalo išsiųsti prašymo signalą. Išsiųstas prašymo signalas aktyvuoja operaciją, kuri siunčia kadrą signalą išsiuntusiai gijai. Tai patogiu, nes nereikia nurodyti iš anksto kur siųsti kadrą. Kadras siunčiamas tam, kas jo paprašo. Duomenų siuntimo operacija:

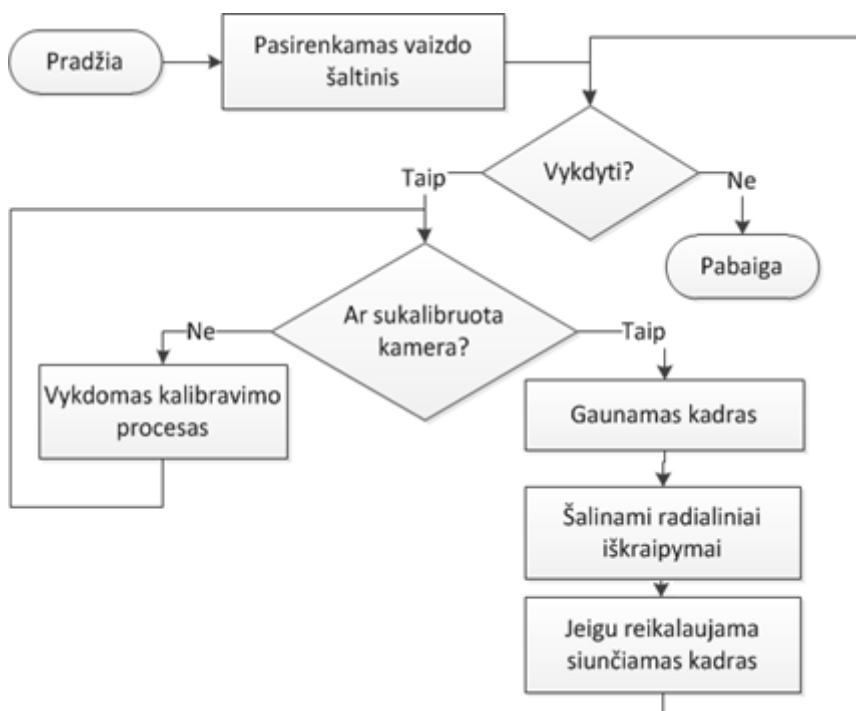
```
void MainWindow::SendKadras() {
    QMetaObject::invokeMethod(this->sender(), "GetKalibruotasKadras",
    Qt::QueuedConnection,
                                Q_ARG(IplImage*, KalibruotasKadras),
                                Q_ARG(int, vaizdas->indeks),
                                Q_ARG(bool, showskin)
    );
}
```

Kaip matome ši operacija siunčia duomenis prašymo signalo siuntėjui „sender()“ į jo priėmimo operaciją „GetKalibruotasKadras“. Siunčiamas kalibruotas vaizdo kadras, kadro numeris ir nurodymas rodyti arba nerodyti išskirtų odos spalvos sričių langą.

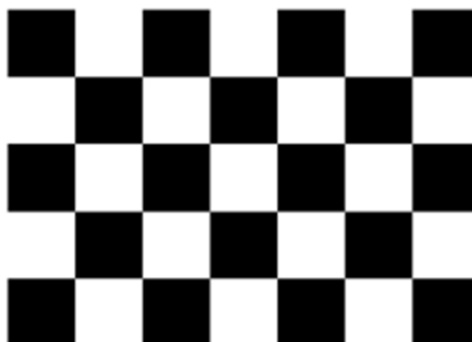
Panaudojant gijas galime optimaliau paskirstyti procesų darbą, nurodyti kada kam dirbi ir kada ne pagal situaciją. Panaudojant gijas buvo optimizuotas kompiuterio procesoriaus resurso panaudojimas, be to, išnaudojamas daugiabranduolių procesorių darbas.

2.4 Vaizdo kadru gavimas, bei pradinis apdorojimas

2.4 iliustracijoje pavaizduotas supaprastintas pirmosios gijos algoritmas. Šioje gijoje gaunami vaizdo kadrai iš pasirinkto šaltinio (vaizdo įrašo arba kameros). Nuo pradinės gijos priklauso visos sistemos darbas, todėl jeigu pirmoji gija baigia darbą, atitinkamai sustabdomos ir kitos gijos. Gaunamas kadras negali būti paduodamas į sistemą nesukalibruotas, todėl pirmą kartą paleidus sistemą būtina sukalibruoti kamerą vykdant kalibravimo procesą.



2.4 pav. Pirmosios gijos algoritmo blokinė schema



2.5 pav. Kalibravimo lenta

Paleidus kameros kalibravimo procesą prieš kamerą turi būti rodoma „šachmatų“ lenta (žr. 2.5 pav.). Šios lentos vieno kvadratėlio plotis ir aukštis lygiai 10 cm. Kas 30-ą kadrą ieškoma

kalibravimo lenta. Aptikus šimtą kalibravimo lentų baigiama kalibravimo lentų paieška ir skaičiuojama iškreipymų matrica. Suskaičiuota matrica išsaugoma kietajame diske. Kuomet egzistuoja iškreipymų matricos byla, sistema leidžia paduoti kadrus kameros radialinių iškreipymų eliminavimo procesui. Kalibravimo proceso realūs vaizdai apteikiami priede 1.

Kadras su eliminuotais radialiniais iškreipymais gali būti siunčiamas į antrąją giją, tik antroji gija turi siųsti kadro prašymo signalą. Jeigu prašymo signalas gautas, kadras siunčiamas į antrąją giją. Sekantis kadras imamas tuomet, kuomet gaunamas signalas iš tam tikro laikmačio, kurio laiko intervalai nustatomi pagal vaizdo šaltinio parametrus. Pavyzdžiui, jeigu vaizdo šaltinis paduoda 25 kadrus per sekundę, tuomet laikmačio intervalai bus: $1000/25=40$ ms. Tai reiškia, jog laikmatis kas 40 ms išsiųs sistemai signalą „paimk naują kadrą“.

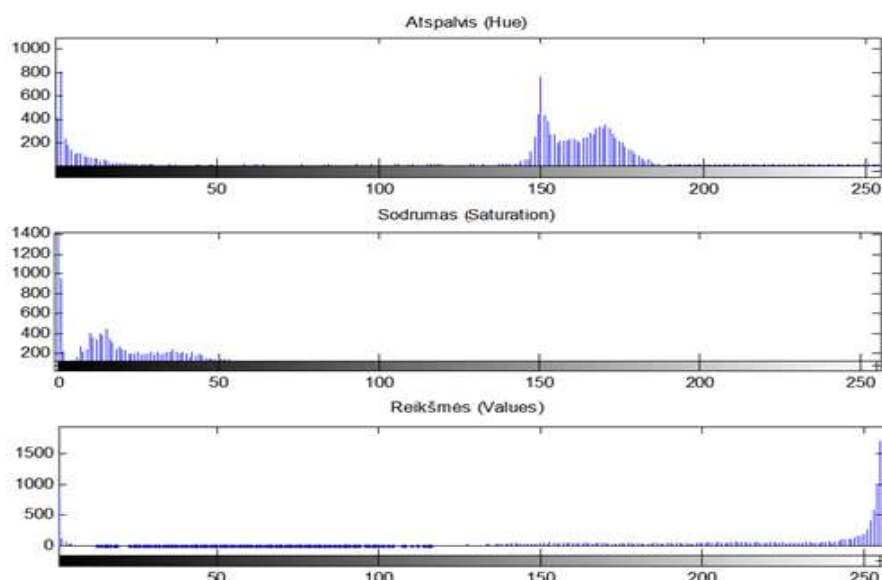
2.5 Odos spalvos objektų išskyrimas

Antrosios gijos pirmasis žingsnis – odos spalvos objektų išskyrimas. Odos spalvos objektų išskyrimui būtina operatoriaus odos spalvos histograma. Ši histograma gaunama „šeimininko“ inicializavimo procedūra.



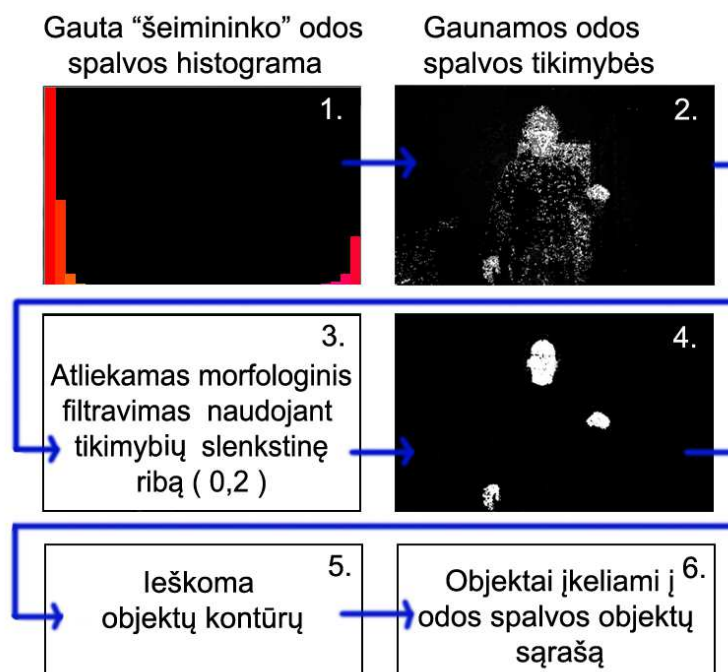
2.6 pav. „Šeimininko“ inicializavimo procedūra

„Šeimininko“ inicializavimo procedūra iliustruojama 2.6 pav.. Ši procedūra paleidžiama atitinkamo mygtuko paspaudimu. Paspaudus mygtuką ekrane rodomas pranešimas apie būsimą veido aptikimą. Pasiruošimui skiriamos penkios sekundės. Kas sekundę rodomas likęs laikas iki veido aptikimo pradžios. Kuomet laikas baigiasi sistema pradeda ieškoti didžiausio veido Haar tipo požymių klasifikatoriaus pagalba per visą kadrą. Kuomet randamas veidas iš aptiktos zonos išskaičiuojama odos spalvos atspalvio HSV spalvų erdvėje histograma. Po histogramos gavimo pradedamas antros gijos darbas.



2.7 pav. Odos spalvos histogramos HSV spalvų erdvėje

Odos spalvos išskyrimui buvo pasirinkta HSV spalvų erdvė. Statinės slenkstinės S ir V reikšmės buvo gautos odos spalvą išskyrus rankiniu būdu. Buvo atrinkta 15 kadrų, kuriuose piešimo programos pagalba buvo užtušuojami ne odos spalvos objektai. Matlab programos pagalba gauta bendros visų kadrų H S ir V histogramos (žr. 2.7 pav.). Iš šių histogramų ir buvo pasirinktos S ir V slenkstinės vertės. S kinta nuo 5 iki 50, o V kinta nuo 240 iki 255.



2.8 pav. Odos spalvos objektų išskyrimas

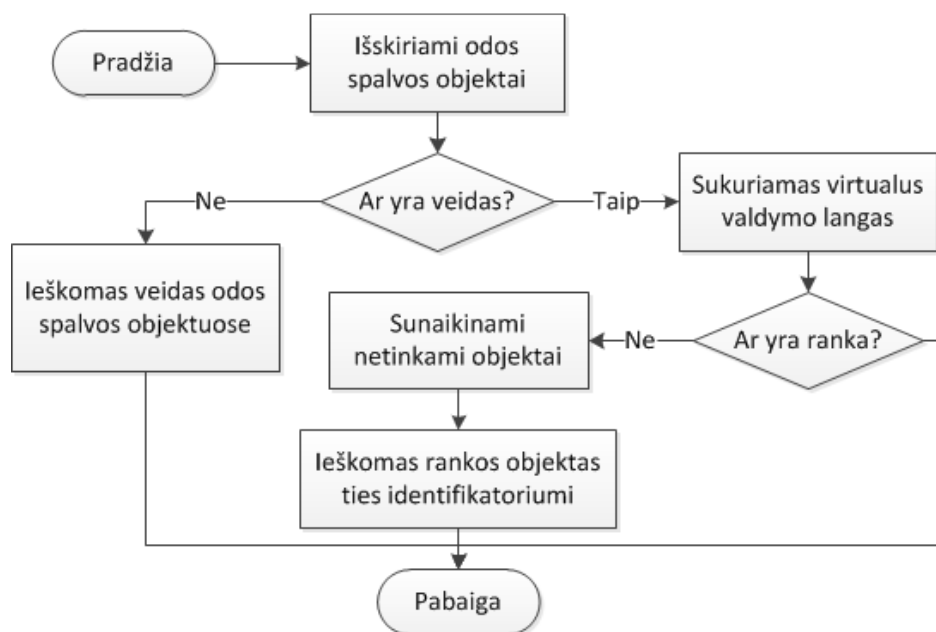
2.8 pav. Iliustruojama odos spalvos objektų išskyrimo eiga. Kaip jau minėjau, pirmame žingsnyje gaunama operatoriaus odos spalvos histograma. Toliau paskaičiuojamos kiekvieno vaizdo taško tikimybės būti odos spalvos tašku. Kaip matome, galimų odos spalvos taškų yra daugiau negu

reikia, todėl įvedamas tikimybės slenkstis, kuris nusako kokia minimali tikimybė yra leistina. Atmetus mažiau tikėtinus taškus vaizdas filtruojamas. Naikinami pavieniai taškai ir užpildomos didelių plotų „skylutės“. Po filtravimo gaunamos ryškios veido ir rankų sritys. Vykdoma kontūrų paieška, kurios pagalba išskiriami objektai. Aplink kontūrus apibrėžiamos elipsės, kurios nusako objekto dydį, pasvirimo kampą bei centrą.

2.6 Rankos ir veido sekimas

Po veido aptikimo sukuriama virtualus valdymo langas (aktyvi zona). Kadangi rankos atpažinti pagal naudojamas kompiuterinės regos technologijas internetinės kameros gaunamame kadre yra beveik neįmanoma, buvo sugalvota sukurti identifikavimo zoną, ant kurios patekęs odos spalvos objektas patenkinantis rankos dydžio sąlygas yra identifikuojamas kaip rankos objektas. Identifikatorius sukuriama aktyvios zonos dešiniajame apatiniame kampe (iliustracija pateikiama priede 2). Ši pozicija pasirinkta todėl, jog ranka patekdama į aktyvią zoną dažniausiai kerta šią identifikatoriaus zoną.

Rankos identifikavimo algoritmas pateikiamas 2.9 paveikslėlyje.



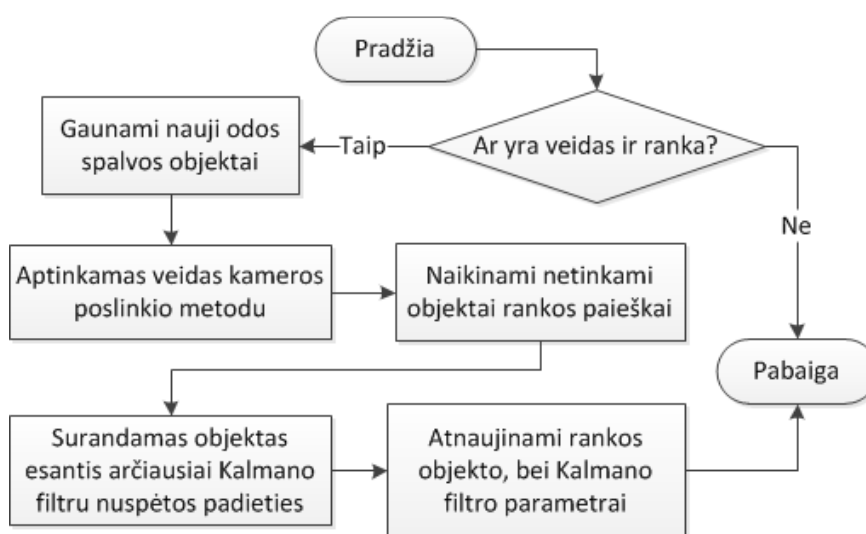
2.9 pav. Rankos identifikavimo algoritmo blokinė schema

Kuomet yra aptiktas veidas ir ranka pradedama objektų sekimo procedūra. Šios procedūros supaprastintas algoritmas iliustruojamas 2.10 paveikslėlyje. Veidas šioje sistemoje sekamas kameros poslinkio metodu. Šis metodas tinkamas būtent veidui todėl, kad veidas palyginus lėtai juda ir yra didelis. Suradus veidą naujame kadre, iš odos spalvos objektų sąrašo panaikinamas veido objektas, bei per maži ir per dideli objektai veido dydžio atžvilgiu. Be to, apibrėžiama zona kurioje gali būti dešinioji ranka.

Zonos viršutinė riba – galvos viršus, kairioji – per du galvos plokščius į kairę nuo galvos vidurio, dešinioji per septynis galvos plokščius į dešinę ir apatinė, per septynis galvos plokščius į apačią. Į septynis galvos plokščius telpa ištiesta ranka į apačią. Šios ribos sukurtos kaip apsauga, jeigu kartais rankos požymis atitektų ne dešinės rankos objektui.

Atrinktame odos spalvos objektų sąrašė ieškoma ranka. Jeigu ranka buvo ankstesniame kadre, imamas Kalmano filtrais nuspėtas taškas ir skaičiuojami kiekvieno odos spalvos objekto atstumai iki to taško. Rastas artimiausias objektas identifikuojamas kaip ranka. Atnaujinami Kalmano filtrai, bei rankos objekto parametrai. Jeigu ankstesniame kadre rankos nebuvo, ieškoma rankos objekto ties identifikatoriumi.

Šis sugalvotas rankos sekimo algoritmais žymiai geriau susidoroja su greitais rankos judesiais negu kameros poslinkio metodas.



2.10 pav. Rankos ir veido sekimo algoritmo blokinė schema

Kalmano filtras šiame algoritme puikiai nuspėja tolygiai greitėjančios, arba lėtėjančios rankos padėtis, tačiau kuomet daromas staigus nenuspėjamas judesys atgal, Kalmano filtras padidina nuokrypius ir netgi gali atsitikti taip, kad arčiausiai Kalmano filtru nuspėto taško gali atsidurti kitas odos spalvos objektas, kuris tuomet būtų identifikuojamas kaip ranka.

2.7 Rankos gestų interpretavimas sistemos valdymui

Trečioji gija atsakinga už rankų gestų interpretavimą ir operacinės sistemos valdymą. Ši gija signalus gauna tiesiogiai iš antros gijos. Signalas neša aktyvios zonos ir rankos objektų kopijas. Šios gijos algoritmo blokinė schema pavaizduota 2.11 paveikslėlyje.

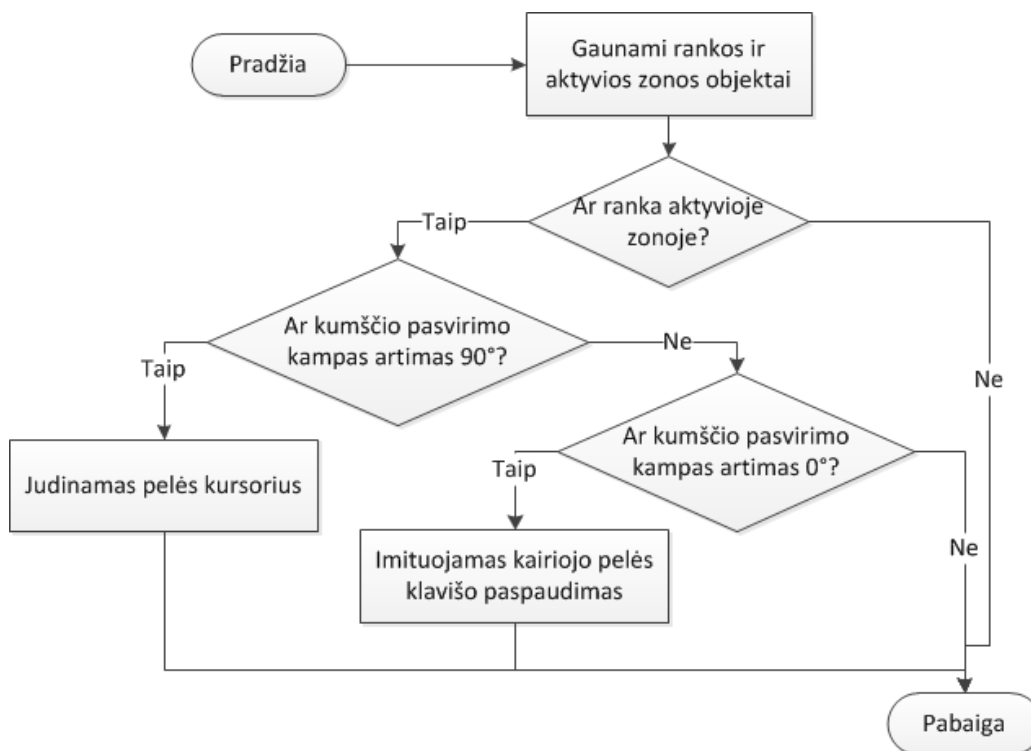
Gavus signalą vykdoma gestų interpretavimo procedūra. Jeigu ranka aktyvioje zonoje stebimas kumščio pasvirimo kampas.

Jeigu kumščio pasvirimo kampas artimas (leidžiamas 5 laipsnių nuokrypis į abi puses) 90 laipsnių, fiksuojami poslinkiai tarp rankos centrų ir transformuojami į pelės žymeklio poslinkius.

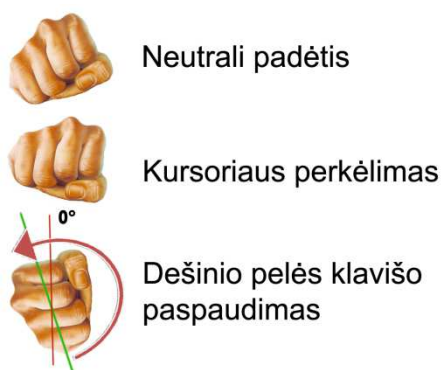
Jeigu poslinkiai labai maži konstatuojama, jog ranka nejuda. Taip eliminuojami virpesiai atsirandantys odos spalvos išskyrimo fazėje. Siekiant palengvinti tolimų objektų pasiekimą buvo įvestas daugiklis, kuris didėja didėjant poslinkiams. Kuo staigesni rankos judesiai, tuo didesni žymeklio perkėlimo poslinkiai.

Jeigu kumščio pasvirimo kampas artimas (leidžiamas 5 laipsnių nuokrypis į abi puses) 0 laipsnių, vykdoma pelės kairiojo klavišo paspaudimo komanda.

Kitais atvejais konstatuojama jog ranka neutralioje padėtyje. Visos galimos kumščio padėtys iliustruojamos 2.12 paveikslėlyje.



2.11 pav. Rankos gestų interpretavimo algoritmo blokinė schema



2.12 pav. Galimos kumščio padėtys

Ši sistemos dalis gali būti papildyta įvairiais rankų gestų atpažinimo algoritmais. Galima būtų sekti rankos trajektorijas ir identifikuoti tam tikrus ženklus. Pasinaudojant HOG (histogramų orientuotų gradientų) metodu atpažinti rankos pirštais rodomus ženklus ir t.t.

3. PRAKTINIS TYRIMAS

3.1 Priemonių pasirinkimas

Siekdamas ištirti savo sugalvotą metodą privalu jį realizuoti programinėje įrangoje. Tam tikslui pasiekti pasirinktos šios priemonės:

- OpenCV 2.1 biblioteka. Ši biblioteka nemokama ir puikiai panaudoja kompiuterio resursus, nes bibliotekos funkcijos parašytos labai žemame programavimo lygmenyje.
- Microsoft windows biblioteka, nes siekiama sukurti sistemą kuri valdytų Windows operacinę sistemą.
- Qt 2.4.0 Creator programavimo aplinka, nes ji taipogi nemokama ir pakankamai funkcionali
- C++ programavimo kalba, nes šią programavimo kalbą moku geriausiai.

Techninė įranga taip pat nieko nekainavo, nes buvo pasirinktas asmeninis nešiojamasis kompiuteris su integruota internetine kamera, kurios parametrai štai tokie:

- 640x400 kadro dydis
- 25 kadrai per sekundę
- Kadro formatas – JPEG

Nešiojamo asmeninio kompiuterio pagrindiniai parametrai:

- Intel dviejų branduolių U4100 1.3 GHz procesorius
- 4GB DDR3 operatyviosios atminties

Rankų išskyrimui pasirinktas HSV spalvų erdvei taikomas odos spalvos objektų išskyrimo metodas, nes gaunamuose kadruose internetine kamera atpažinti ranką pagal formą yra beveik neįmanoma. HSV spalvų erdvės privalumas – padidina atsparumą apšvietimo pokyčiams.

Objektų sekimui pasirinktas kameros poslinkio metodas, nes šis metodas pakankamai tikslus ir labai spartus.

3.2 Eksperimentų atlikimo tvarka, bei numatytų tyrimų tikslai

Šio darbo pagrindinis tikslas – sukurti ekrano valdymo rankų gestais sistemos modelį, todėl pagrindinis planuojamų atlikti tyrimų tikslas - ištirti atskirų sistemos dalių darbą, bei daromą įtaką bendram sistemos darbui. Planuojama atlikti šiuos tyrimus:

- 1) Veido atpažinimo ir sekimo metodo tyrimas
- 2) Odos spalvos objektų išskyrimo metodo tyrimas
- 3) Objektų sekimo metodo tyrimas
- 4) Kalmano filtro įtakos rankų sekimui tyrimas
- 5) Sistemos darbo kituose kompiuteriuose tyrimas

- 6) Sekimo priklausomybės nuo judančio objekto greičio tyrimas
- 7) Sekimo priklausomybės nuo judančio objekto pagreičio tyrimas

Šių tyrimų pagalba bus siekiama:

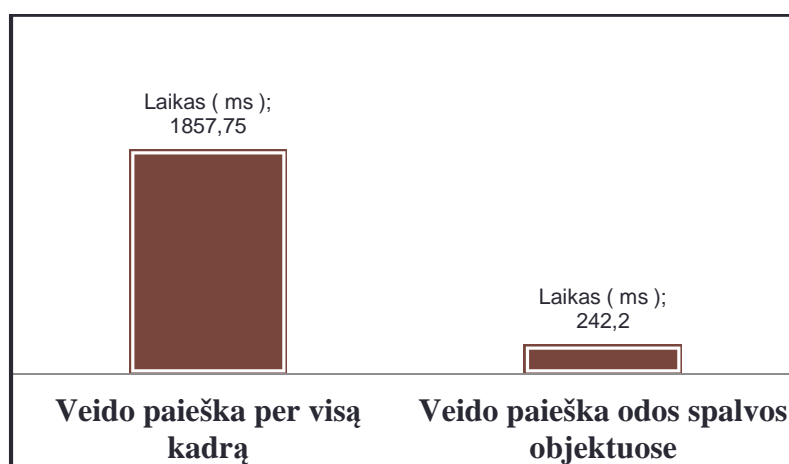
- Išsiaiškinti sistemos trūkumus bei privalumus
- Atlikti atskirų sistemos dalių patobulinimus pašalinant arba minimizuojant trūkumus
- Iširti sistemos tinkamumą Windows operacinės sistemos valdymui realiu laiku

3.3 Veido paieškos sparta

Vykdamas „šeimininko“ inicializavimo procedūrą veido ieškoma po visą kadrą, kai tuo tarpu sistemos darbinėje stadijoje pradingus veidui veidas ieškomas odos spalvos objektuose.

Buvo atliktas veido paieškos pradinėje ir darbinėje stadijose palyginimas. Tyrimo tikslas – išsiaiškinti kadrų apdorojimo laiko skirtumą vykdant paiešką pradinėje ir darbinėje stadijose.

Pradinėje stadijoje vykdoma veido paieška Haar tipo požymių klasifikatoriaus pagalba po visą kadrą. Paprastai pradinėje stadijoje pakanka apdoroti vieną kadrą, nes operatorius jau būna pasiruošęs veido aptikimo procedūrai. Ši veido paieška užtrukdavo apie 300 ms. Tačiau atliekant šį tyrimą veidas buvo uždengiamas ranka tam tikrą laiko tarpą. Pastebėta, jog kuo ilgiau sistema nerasdavo veido tuo labiau ilgėdavo kadro apdorojimo laikas. Laikant uždengtą veidą 20 sekundžių kadro apdorojimo laikas išaugo iki 1,857 ms. Jeigu šį metodą naudotumėme ir darbinėje stadijoje sistema nebesuspėtų dirbti realiu laiku, todėl buvo sugalvotas veido paieškos paspartinimas.



3.1 pav. Kadro apdorojimo laiko vykdant veido paieškas palyginimas

Trumpai apžvelgsime sugalvotą veido paieškos algoritimą. Kol veido objektas nesusiduria su kritinėmis sąlygomis (dingsta iš matymo lauko, staiga padidėja ar sumažėja ir pan.) veido atpažinimo algoritmas nėra vykdomas. Tačiau įvykus kritinei sąlygai pradedama veido paieška per odos spalvos objektų sąrašą. Sąrašas išrikiuojamas mažėjimo tvarka ir pradedama ieškoti nuo

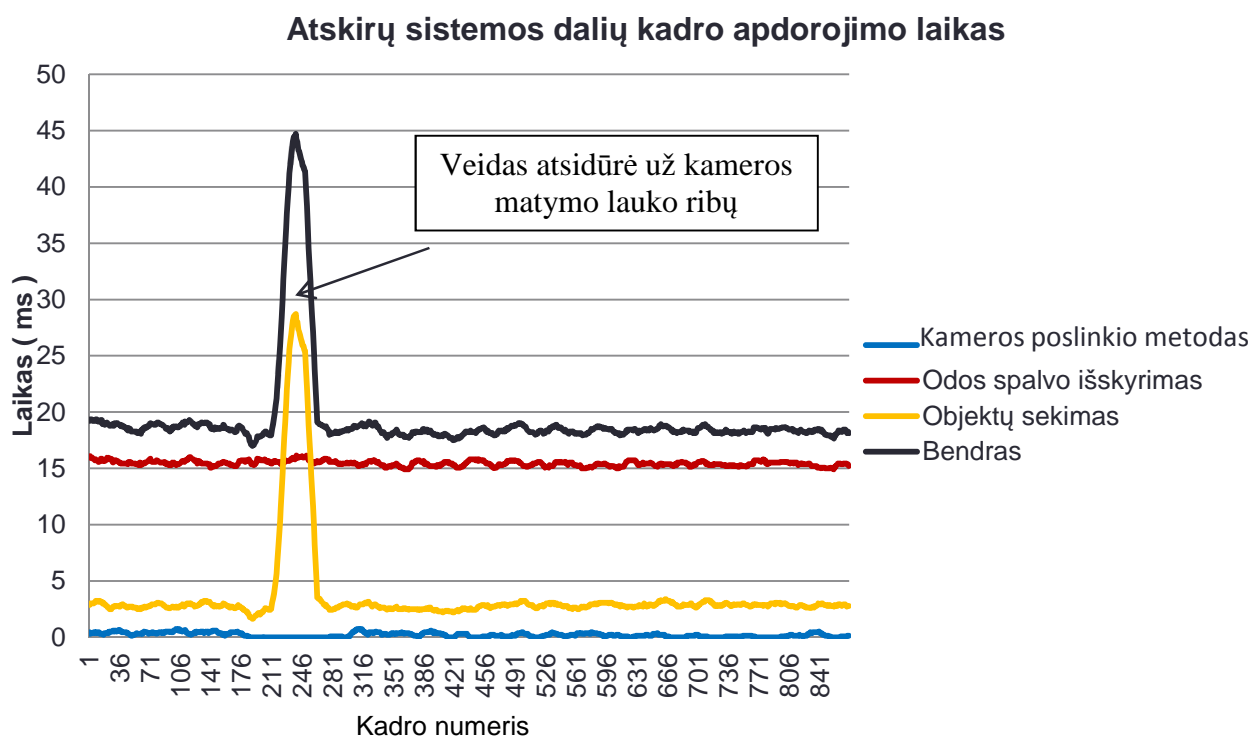
didžiausio odos spalvos objekto. Veidas ieškomas taip pat Haar tipo požymių klasifikatoriaus pagalba. Aptikus veidą, odos spalvos objektas pažymimas kaip veidas ir perkeliamas į atpažintų objektų sąrašą.

Pastebėta, jog šio metodo pagalba kadro apdorojimo laikas vykdant veido paiešką ženkliai sutrumpėjo. Kadro apdorojimo vidutinis laikas – 242 ms.

Rezultatai pateikiami 3.1 diagramoje.

3.4 Atskirų sistemos dalių įtaka sistemos spartai

Tyrimo tikslas – išsiaiškinti kadro apdorojimo laiko sudedamąsias.



3.2 pav. Atskirų sistemos dalių kadro apdorojimo laiko grafikas

Atskiruose kadro apdorojime dalyvaujančiuose sistemos dalyse buvo įrašyti laiko matuokliai. Buvo leidžiamas vaizdo įrašas ir matuojami atskirų sistemos dalių kadro apdorojimo laikai. Rezultatai pateikiami 3.2 paveikslėlyje.

Grafike aiškiai matome jog daugiausiai laiko užtrunka odos spalvos objektų išskyrimas. Šis sistemos dalis naudoja daugiausiai kompiuterio resursų, be to, dėl odos spalvos išskyrimo atsiranda dideli sistemos matavimų nuokrypiai. Šią sistemos dalį reikia tobulinti, arba panaudoti kitokį objektų išskyrimo iš aplinkos metodą.

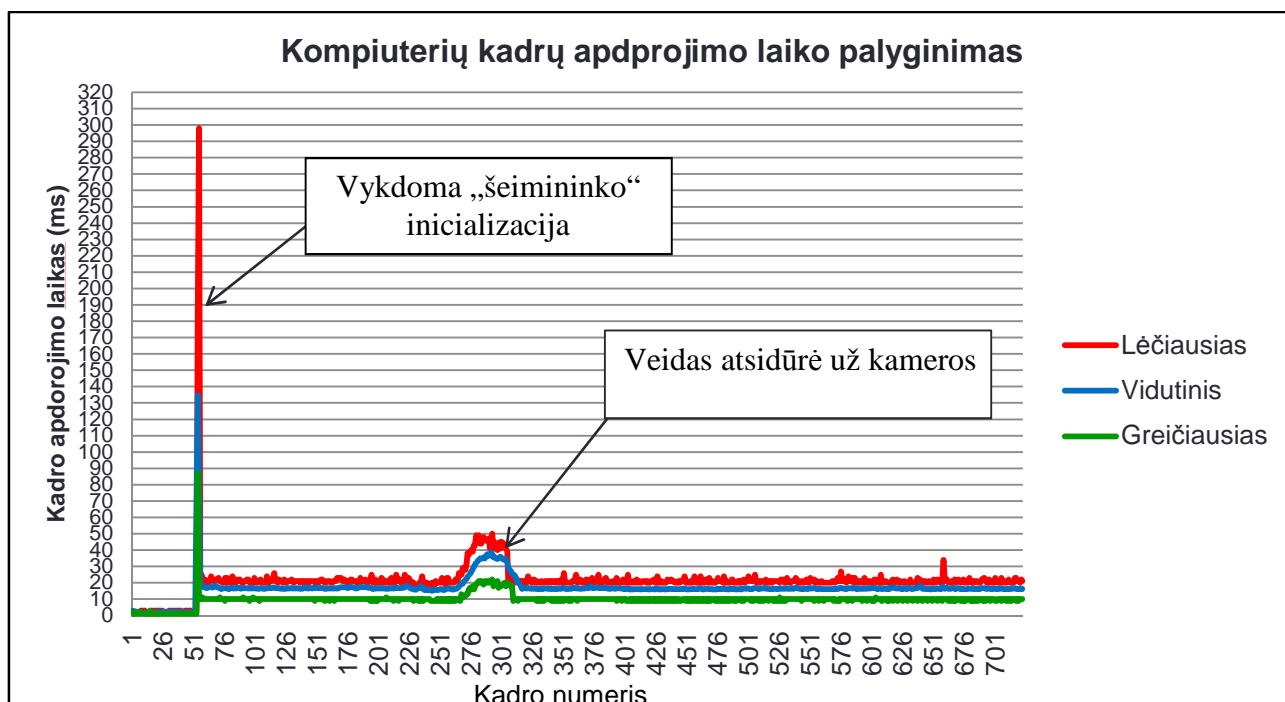
3.5 Sistemos darbo skirtinguose kompiuteriuose

Tyrimo tikslas – patikrinti sistemos tinkamumą realaus laiko valdymui.

Šiam tikslui pasiekti buvo pasirinkti trys kompiuteriai:

- Nešiojamas kompiuteris:
 - Intel dviejų branduolių U4100 1.3 GHz procesorius
 - 4GB DDR3 operatyviosios atminties
- Nešiojamas kompiuteris:
 - Intel Pentium Dual T2390 1.86GHz procesorius
 - 3GB DDR2 operatyviosios atminties
- Stacionarus kompiuteris:
 - Pentium Dual Core E6500 2.93GHz procesorius
 - 2GB DDR2 operatyviosios atminties

Visuose įdiegta Windows 7 32-bit operacinė sistema.



3.3 pav. Kompiuterių kadro apdorojimo laiko grafikas

Kiekviename kompiuteryje buvo paleidžiamas tas pats vaizdo įrašas po penkis kartus. Gauti duomenys išsaugomi „csv“ lentelių bylose. Atrinkti duomenys, kuriuose matoma mažiausia kitų programų (antivirusinių ir pan.) įtaka. Gauti atskirų kompiuterių duomenys pateikti 3.3 paveikslėlyje.

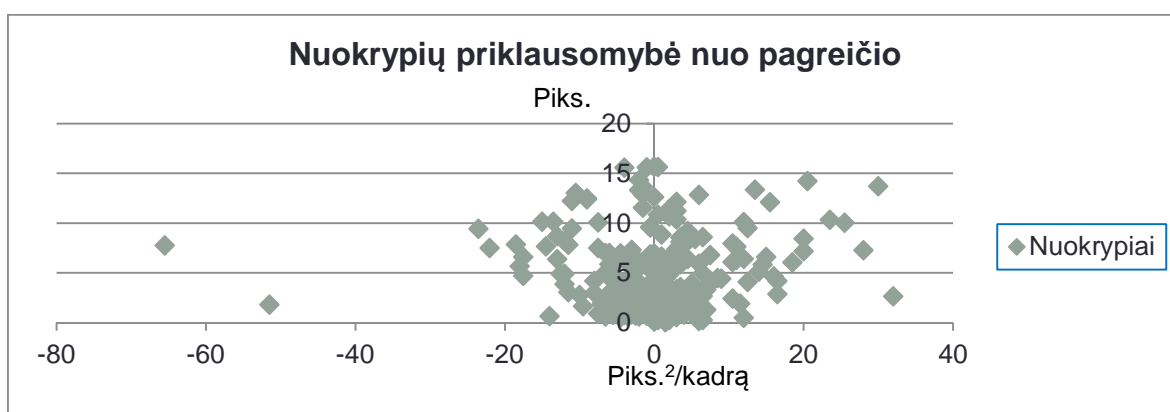
Iš grafiko matyti, jog vidutinis kadro apdorojimo laikas valdymo rankų gestais metu apie 19 ms, o tai reiškia, jog sistema gali apdoroti iki 52 kadro per sekundę srautą. Kadangi internetinės

kameros dažniausiai siunčia 25 kadro per sekundę srautą, galime teigti, jog ši sistema tinkama ekrano valdymui rankų gestais realiu laiku.

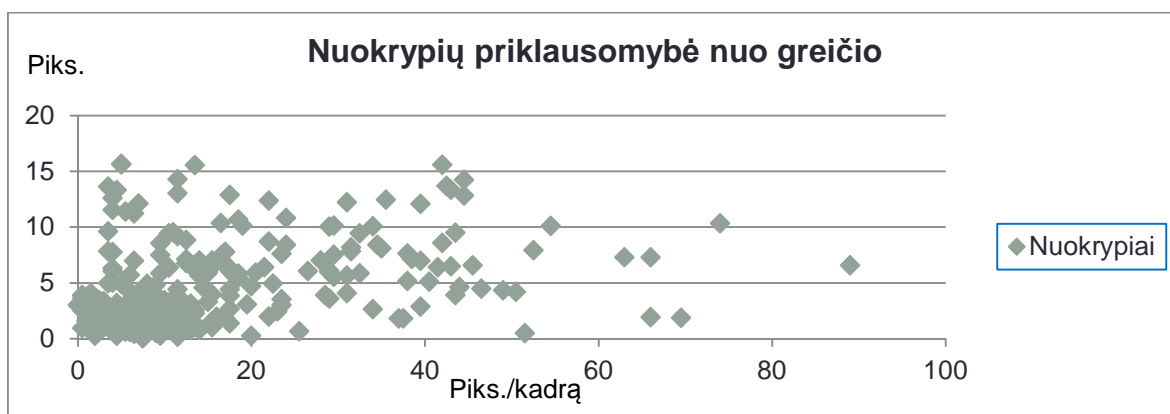
3.6 Sekimo nuokrypių priklausomybė nuo sekamos rankos judėjimo greičio ir pagreičio

Tyrimo tikslas – ištirti rankos sekimo nuokrypių priklausomybes nuo rankos judėjimo greičio ir pagreičio.

Šiam tyrimui atlikti buvo sukurta atskira programa, kurios pagalba rankiniu būdu buvo sužymimi rankos centrai judriausioje vaizdo įrašo atkarpoje (330 kadro). Po to, toje pačioje atkarpoje buvo gaunami sekamos rankos centrai. Rezultatai pateikiami 3.4 ir 3.5 paveiksluose.



3.4 pav. Nuokrypių priklausomybės nuo pagreičio grafikas



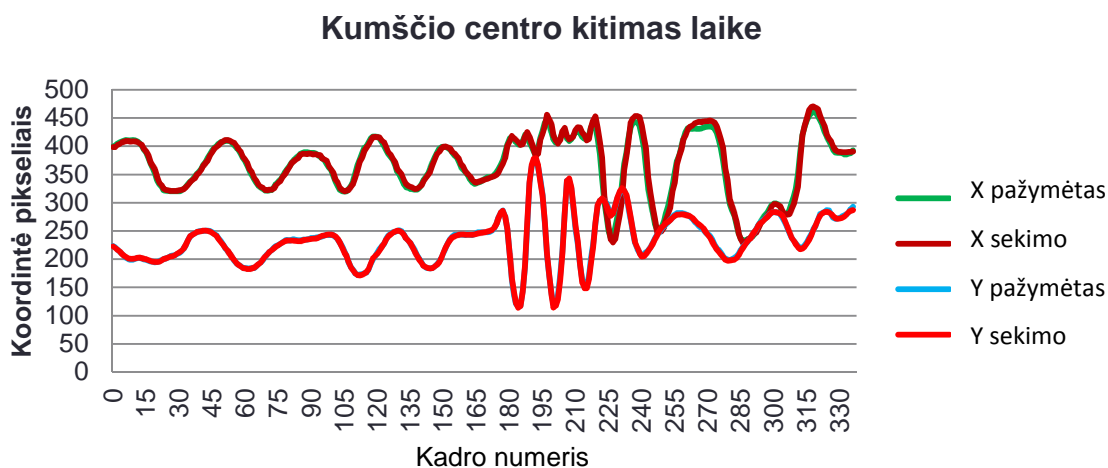
3.5 Nuokrypių priklausomybės nuo greičio grafikas

Kaip matome ryškių priklausomybių nėra, o tai yra todėl, kad kuomet ranka juda labai greitai atsiranda nuokrypiai kuriuos slopina Kalmano filtrai, tačiau staigiai kintant trajektorijai greitis ir pagreitis trumpam sumažėja, o tuo tarpu Kalmano filtrai nuokrypius tik tai didina. Šio tyrimo rezultatai aiškiai parodo Kalmano filtrų privalumus ir trūkumus šios sistemos darbui.

3.7 Sekimo nuokrypių priklausomybė nuo sekamos rankos judėjimo trajektorijos

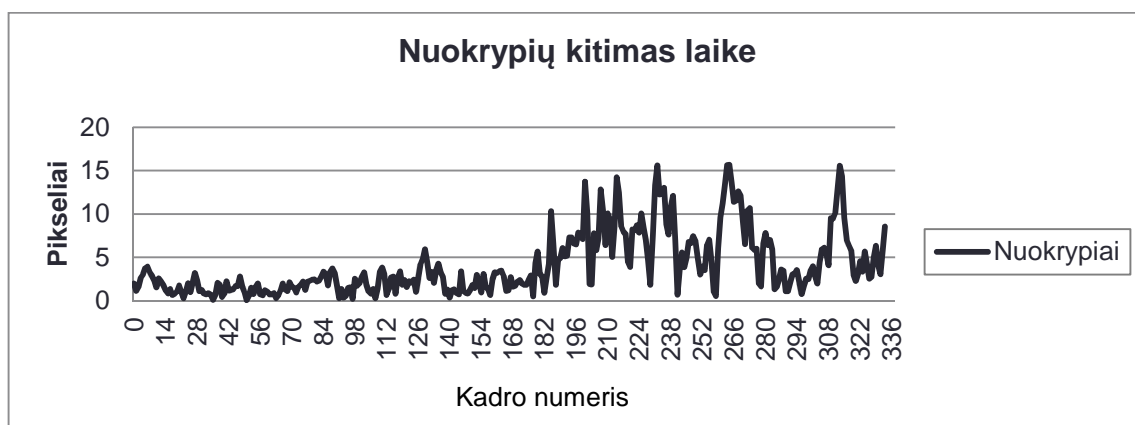
Tyrimo tikslas – iširti nuokrypių priklausomybę nuo trajektorijų.

Duomenys buvo pasirinkti tokie pat kaip ir nuokrypių priklausomybių nuo greičio ir pagreičio tyrime. Rankiniu būdu ir sistemos sekimo gautos kumščio centrų koordinatės atvaizduotos 3.6 paveikslėlyje, gauti nuokrypiai 3.7 paveikslėlyje.

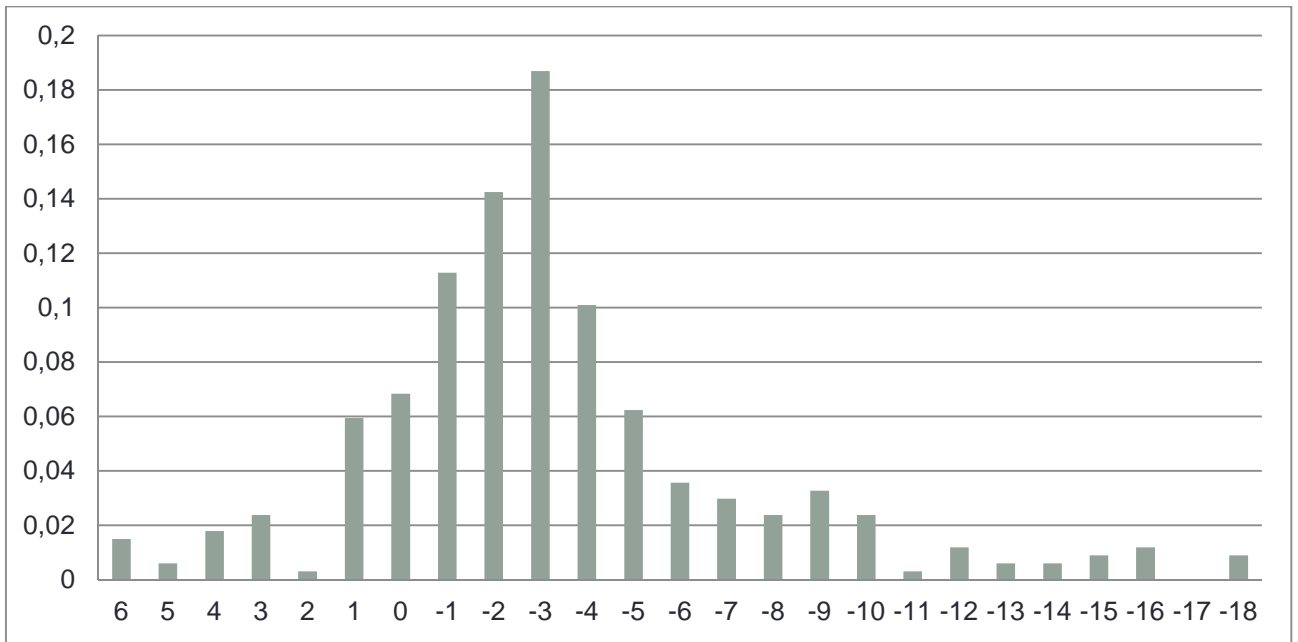


3.6 pav. Kumščio centro koordinatžių kitimo laike grafikas

Kaip matome vizualiai kumščio centro trajektorijos beveik identiškos. Didžiausi nuokrypiai atsiranda tuomet, kuomet ranka pradeda judėti staigiais, didelės amplitudės judesiais aukštyn ir žemyn. Tai tik dar kartą patvirtina, jog Kalmano filtrai ties staigiai trajektorijos pakitimais dėl netikėtumo (negalima nuspėti) padidina nuokrypius nuo tikrosios kumščio padėties.



3.7 pav. Nuokrypių kitimo laike grafikas

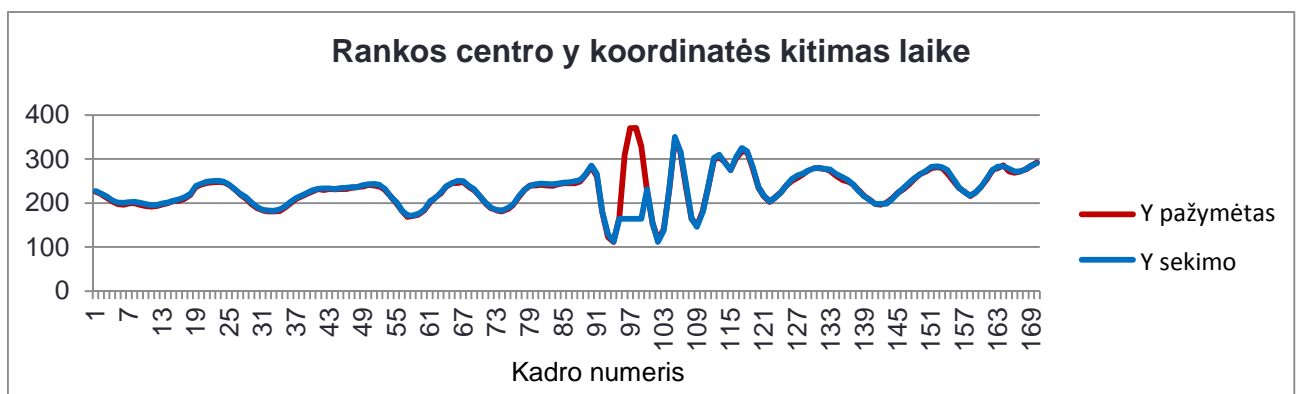


3.8 pav. Gautų nuokrypių tikimybinis pasiskirstymas

3.8 paveikslėlyje pavaizduotas gautų nuokrypių tikimybinis pasiskirstymas. Gauti parametrai:

- **Matematinė viltis:** -3,3 piks.
- **Dispersija:** 28,97 piks.²
- **Standartinis nuokrypis:** 5,39 piks.

Gautas pasiskirstymas smailus, todėl galime teigti jog dauguma reikšmių yra pakankamai tikslios.



3.9 pav. Kumščio centro y koordinatės kitimo dvigubai pagreitiname vaizdo įrašė grafikas

Tas pats eksperimentas buvo atliktas pagreitinus vaizdą du kartus. Žinoma ranka judėdama tokiu greičiu paprastos internetinės kameros kadruose būtų kabai išsiliejusi arba visai neregima, tačiau eksperimentas buvo atliktas siekiant išsiaiškinti maksimalų sistemos sekimo greitį ir pagreitį. Rezultatai pateikiami 3.9 paveikslėlyje, kuriame aiškiai matome trajektorijų nesutapimus. Maksimalus gautas pagreitis – 80 piks.²/kadra. Maksimalus greitis – 148 piks./kadra.

3.8 Pastebėti sistemos trūkumai

Testuojant sistemą buvo pastebėti šie trūkumai:

- Raudonos spalvos rūbai aptinkami kaip odos spalva, nes baltaodžio žmogaus odos spalva turi daug raudonos spalvos atspalvio
- Būtina vilkėti rūbus su ilgomis rankovėmis. Šį trūkumą būtų galima pašalinti panaudojant apyranę.
- Automatinis baltos spalvos balansas sunkiai susidoroja su silpnu, bei mažai infraraudonųjų spindulių spinduliuojančiu apšvietimu. Spalvos gaunamos „šaltos“ (dominuoja mėlynas atspalvis), todėl prastai išskiriami odos spalvos objektai. Šią sistemą patartina naudoti prie kaitrinių lempų arba saulės šviesos apšvietimo.

IŠVADOS

1. Atlikus egzistuojančių analogiškų sistemų analizę buvo išskirtos trys pagrindinės sistemos dalys: vaizdo gavimas, objektų išskyrimas ir gestų interpretavimas.
2. Veido atpažinimui pasirinktas vienas iš sparčiausių ir tiksliausių Viola ir Jones Harr tipo požymiais pagrįstas veido atpažinimo metodas, kuris panaudojamas tik veido objekto pradiniam atpažinimui, bei odos spalvos zonos parinkimui.
3. Rankos identifikavimas vykdomas užvedant ranką ant identifikatoriaus, nes gaunamas vaizdas internetine vaizdo kamera yra per prastas rankos atpažinimui kompiuterinės regos metodais.
4. Atlikus bandymus pasirinktas optimaliausias odos spalvos išskyrimo metodas, kuris HSV spalvų erdvėje išskiria odos spalvą pagal odos spalvos histogramą.
5. Ištyrus sistemos spartumą paaiškėjo, jog sistema tinkama valdymui realiu laiku, nes vidutiniškai ji gali apdoroti iki 52 kadro per sekundę gestų valdymo studijoje. Maksimalus rankos sekimo pagreitis – 80 piks.²/kadra, o greitis 148 piks./kadra.
6. Atlikus sistemos sekimo nuokrypių analizę paaiškėjo, jog dėl Kalmano filtrų nuokrypiai išauga rankai staiga pakeitus trajektorijos kryptį, o judant tolygiai nuokrypiai dėl Kalmano filtrų prognozavimo mažėja.
7. Atliekant sukurtos sistemos bandymus buvo pastebėtas didžiausias jos trūkumas – odos spalvos objektų išskyrimas. Jis užtrunka ilgiausiai, bei esant prastam apšvietimui nebesugeba tinkamai išskirti odos spalvos objektų, tačiau esant geram, infraraudonosios spindulius spinduliuojančiam apšvietimui, sistema dirba stabiliai.

LITERATŪROS SĄRAŠAS

1. Cheng K. Interaction Paradigms for Bare-Hand Interaction with Large Displays at a Distance/K. Cheng, M. Takatsuka. Sidnėjaus universitetas, Australija, 2012. 27p.
2. Hasup L. Hand Gesture Recognition using Blob Detection for Immersive Projection Display System/Hasup Lee, Yoshisuke Tateyama, Tetsuro Ogi/ International Journal of Computer and Communication Engineering. 2012, nr. 6, 4p.
3. Ki-Won B. Skin Region Detection Using a Mean Shift Algorithm Based on the Histogram Approximation/Ki-Won Byun* and Ki-Gon Nam, Soo-Young Ye/Transaction on electrical and electronic materials, 2012, nr 13, 10-15p.
4. Vezhnevets V. A Survey on Pixel-Based Skin Color Detection Techniques / Vezhnevets V., Sazonov V., Andreeva A., Graphics and Media Laboratory, Faculty of Computational Mathematics and Cybernetics, Maskvos valstybinis universitetas, 2010, 8p.
5. Hasup L. Realistic Visual Environment for Immersive Projection Display System/Hasup Lee, Yoshisuke Tateyama, Tetsuro Ogi/International Conference on Virtual Systems and Multimedia, 2010, nr. 16, 128-132p.
6. Hidayatullah P. Object Tracking: State of The Art and CAMSHIFT Improvement Using Multi-dominant Colors Tracking/ Master Thesis Report. Prancūzija, 2010, 62p.
7. Hasup L. Panoramic Stereo Representation for Immersive Projection Display System/Hasup Lee, Yoshisuke Tateyama, Tetsuro Ogi/International Conference on VRCAI (VR Continuum and Its Applications n Industry). 2010, nr. 9, 79-382p.
8. Kakumanu S. A survey of skin-color modeling and detection methods/Kakumanu, S. Makrogiannis, N. Bourbakis/Pattern Recognition,, 2007, nr. 40, 1106-1122p.
9. Robyn C. Experiments with a Camera-Based Human-Computer Interface System/Robyn Cloud, Margrit Betke, James Gips, Bostono universitetas, 2005, 8p.
10. Piccardi M. Background subtraction techniques: a review. Sidnėjus, Australija, 2004, 30p.
11. Sezgin M. Survey over image thresholding techniques and quantitative performance evaluation/ M. Sezgin, B. Sankur/Electronic Imaging. 2004, nr. 13, 146–165p.
12. Y. Wang and B. Yuan, A novel approach for human face detection from color images under complex background/Pattern Recognition, 2001, nr. 34, 1983–1992p.
13. Viola P. Robust Real-time Object Detection/P. Viola, M. Jones. Vankuveris, Kanada, 2001, 25p.
14. Welch G. An Introduction to the Kalman Filter/G. Welch, G. Bishop. Šiaurės Kalifornijos universitetas, 2001, 81p.

15. Stauffer C. *Adaptive background mixture models for real-time tracking*/C. Stauffer, W. E. L. Grimson/Computer Vision and Pattern Recognition, Colorado. 1999, 7p.
16. Benjamin D. Comparison of Five Color Models in Skin Pixel Classification/Benjamin D. Zait, Boaz J. Super, Francis K. H. Quek/International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-time Systems, 1999, 58-63p.

PRIEDAI

PRIEDAS 1

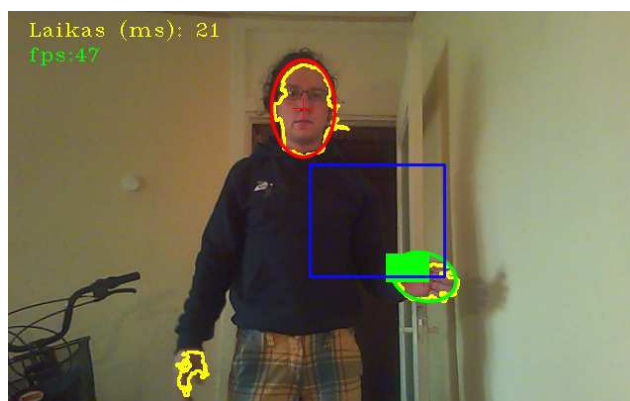


Kameros kalibravimo procesas



Prieš kalibravimą ir po

PRIEDAS 2



Rankos identifikavimas