

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

Evaldas Borcovas

**GALVOS POZICIJOS NUSTATYMAS EKRANO
ATŽVILGIU, PANAUDOJANT ANT GALVOS
PRITVIRTINTĄ KAMERĄ
Magistro darbas**

Vadovas

prof. dr. G. Daunys

ŠIAULIAI, 2012

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

TVIRTINU
Katedros vedėjas

prof. dr. G. Daunys

2012 06

**GALVOS POZICIJOS NUSTATYMAS EKRANO
ATŽVILGIU, PANAUDOJANT ANT GALVOS
PRITVIRTINTĄ KAMERĄ
Magistro darbas**

Recenzentas

ŠU technologijos fakulteto
Elektronikos katedros
doc. dr. D. Dervinis
2012 06

Vadovas

2012 06

prof. dr. G. Daunys

Atliko

2012 06 09

RMO stud.
Evaldas Borcovas

ŠIAULIAI, 2012

SANTRAUKA

Borcovas E., Galvos pozicijos nustatymas ekrano atžvilgiu, panaudojant ant galvos pritvirtintą kamerą: Signalų technologijos magistro darbas / mokslinis vadovas doc. dr. G. Daunys; Šiaulių universitetas, Technologijos fakultetas, Elektronikos katedra, -Šiauliai, 2012. – 63 p.

Tobulėjant kompiuteriams, ieškoma vis naujų galimybių informacijos perdavimui tarp vartotojo ir mašinos. Yra atliekama daug tyrimų *žmogaus ir kompiuterio sąveikos* – HCI (Human computer interaction) – srityje. Ieškoma naujų būdų bei metodų, kaip perduoti informaciją iš vartotojo kompiuteriui, ir atvirkščiai. Senesni metodai, tokie kaip įvestis pelės ir klaviatūros pagalba, išvestis vaizduoklio, projektoriaus bei spausdintuvo pagalba, yra optimizuojami. Siekiama surasti ergonomiškiausių įvesties, išvesties ir informacijos perdavimo metodą. Darbo tikslas – sumodeliuoti ir realizuoti sistemą galvos pozicijos nustatymui ekrano atžvilgiu kompiuterinės regos pagalba ir sukurti algoritmą, kurio pagalba nustatoma galvos padėtis ekrano atžvilgiu.

Sistemos kūrimui atliktas keturkampio geometrinių parametrų tyrimas. Pasirinkti geometriniai keturkampio požymiai, turintys daugiausiai informacijos apie galvos pozicijos ir orientacijos parametrus. Taip pat nustatyta parametrų tarpusavio priklausomybė, skaičiavimo eiliškumas. Pagal keturkampio geometrines savybes sukurtas taškų filtravimo ir rūšiavimo algoritmas.

Algoritmas realizuotas 4 skirtingose aplinkose: Matlab, C# windows form application, C# console application ir C++ console application. Algoritmui atlikti tyrimai: optimalus algoritmo parametrų parinkimas, algoritmo vykdymo sparta skirtingose aplinkose, algoritmo segmentų spartos tyrimas, algoritmo tikslumo įvertinimas. Nustatyti optimalūs parametrai sistemos veikimui ir pasiūlyti pradiniai jos parametrai. Ištirta algoritmo sparta ir nustatyta, kuriame algoritmo vykdymo etape yra ilgiausiai skaičiuojama. Lėtai veikiančios etapai optimizuoti. Pasirinkta patogiausia ir sparčiausia aplinka algoritmo realizavimui. Nustatytos algoritmo atsitiktinės ir matavimo paklaidos. Reikšmių patikslinimui taikomas Niutono optimizavimo metodas. Nustatomas optimizavimo metodo tinkamumas sistemai.

SUMMARY

Borcovas E., Tracking of Head Position Relative to the Screen using Head Mounted Camera: Master thesis of Signal Technology / research advisor Assoc. Dr. G. Daunys; Šiauliai University, Faculty of Technology, Electronics Department, – Šiauliai, 2012. – 63 p.

With the development of computers engineers are looking for new ways of commutation between user and machine. There are many researches done in human-computer interaction – HCI field. Engineers are always searching for new ways, methods of transmitting information from user to machine and vice versa. Older methods, mouse and keyboard input, monitor, projector and printer output, are optimized. The aim of these studies is to find the most ergonomic input, output and communication method. The main goal is to design and implement system for determining the position of the head against the screen with help of computer vision aid. Create an algorithm for determining the position of the head against the screen.

System development was conducted from rectangle geometry research. The geometric features of a rectangle with the most information about head position and orientation parameters were selected. Estimated parameters of interdependence, the calculation sequence. According to the geometrical properties of the rectangle an algorithm was developed for filtering and sorting points.

The algorithm is realized in four different environments: Matlab, C # Windows Form application in C # console application and C++ console application. The next algorithm research was performed: optimal algorithm parameter selection, algorithm execution speed in different environments, algorithm segments speed test, evaluation of the accuracy of the algorithm. Optimal parameters for the system operation were determined and their initial values were suggested. The speed of the algorithm in every segment was determined and the segment which needs longest calculation was found. Slow working segments were optimized. The most comfortable and fastest environment for algorithm implementation was selected. Random and measurement errors were determined for the algorithm. Value adjustment was made by Newton optimization method. The suitability of system optimization method was determined.

TURINYS

IŽANGA.....	8
GALVOS PADĖTIES ERDVĖJE NUSTATYMO SISTEMOS.....	10
1.1. Parametrai galvos padėčiai nustatyti.....	10
1.2. Galvos padėties erdvėje nustatymo metodai.....	10
1.2.1. Stereo matymo metodas.....	11
1.2.2. Jutiklių metodas.....	13
1.2.3. Šviesos atspindžio metodas.....	14
1.2.4. 3D Elipsoidės metodas.....	16
1.2.5. Išvaizdos algoritmo metodas.....	17
1.2.6. Paukščio skrydžio metodas.....	19
1.2.7. Cilindro tekstūros metodas.....	21
1.3. Pasirinkta vaizdų apdorojimo metodika.....	22
1.3.1. Komporavimas.....	22
1.3.2. Sobel operatorius.....	23
1.3.3. Kraštų aptikimas.....	24
1.3.4. Kontūrų aptikimo.....	25
1.3.5. Subpikselinių verčių nustatymo metodas.....	26
1.3.6. Kalmano filtras.....	27
1.3.7. Niutono optimizavimo metodas.....	28
2. GALVOS POZICIJOS NUSTATYMO SISTEMA.....	30
2.1. Techninė įranga.....	30
2.2. Programinė įranga.....	32
3. TYRIMAI.....	38
3.1. Keturkampio geometrinių savybių pokytis, kintant stebėjimo kampui.....	38
3.2. Keturkampio parametrų tarpusavio priklausomybių nustatymas.....	43
3.3. Taškų filtravimas ir rūšiavimas taikant geometrines taisykles.....	48
3.4. Algoritmo parametrų optimalus parinkimas.....	50
3.5. Algoritmo vykdymo laiko tyrimas įvairiose aplinkose.....	51
3.6. Algoritmo segmentų vykdymo laiko įvertinimas.....	54
3.7. Algoritmo tikslumo įvertinimas.....	56
3.8. Ieškomų reikšmių optimizavimas Niutono metodu.....	58
IŠVADOS IR SIŪLYMAI.....	60
LITERATŪRA.....	62
PRIEDAI.....	64

LENTELĖS

1 lentelė. Algoritmo atsitiktinės neapibrėžtys.....	57
2 lentelė. Algoritmo matavimo neapibrėžtys keičiant x koordinatę.....	57
3 lentelė. Algoritmo matavimo neapibrėžtys keičiant β kampą.....	58

PAVEIKSLAI

1.1 pav. Galvos padėties nustatymo parametrai.....	10
1.2 pav. Stereo matymo metodo techninės įrangos schema.....	11
1.3 pav. Stereo matymo metodo vaizdų apdorojimo algoritmo blokinė schema.....	12
1.4 pav. Stereo matymo metodo charakteringi taškai.....	12
1.5 pav. Jutiklių metodo techninės įrangos išdėstymas.....	13
1.6 pav. Šviesos atspindžio metodo veikimo principas.....	14
1.7 pav. Šviesos atspindžio metodo vaizdų apdorojimo algoritmo blokinė schema.....	15
1.8 pav. Šviesos atspindžio metodo vaizdo filtravimas.....	15
1.9 pav. Šviesos atspindžio metodo galvos padėties sekimas realiu laiku.....	15
1.10 pav. Vienos kameros sekimo metodo techninės įrangos išdėstymas.....	16
1.11 pav. 3D elipsoidės metodo vaizdo apdorojimo blokinė schema.....	17
1.12 pav. 3D elipsoidės metodo vaizdo apdorojimas.....	17
1.13 pav. Išvaizdos algoritmo vaizdų apdorojimo algoritmo blokinė schema.....	18
1.14 pav. Aktyvių išvaizdos modelių metodas.....	18
1.15 pav. Aktyvių išvaizdos modelių metodo per greito judesio apdorojimas.....	19
1.16 pav. Paukščio skrydžio metodo techninės įrangos išdėstymas.....	20
1.17 pav. Paukščio skrydžio metodo galvos pozicijos parametrų apskaičiavimo principas.....	20
1.18 pav. Cilindro tekstūros metodo algoritmo blokinė schema.....	21
1.19 pav. Cilindro tekstūros modelio gavimo pavyzdys.....	21
1.20 pav. Kompjavimo pavyzdys.....	23
1.21 pav. Sobel operatoriaus pavyzdys.....	23
1.22 pav. Ryškumo perėjimo algoritmų metodai.....	25
1.23 pav. Kontūrų radimo metodo pavyzdys.....	25
1.24 pav. Vertikalios pikselių ryškumo histogramos.....	26
1.25 Pav. Subpikselinių verčių nustatymas.....	26
1.26 pav. Kalmano filtro veikimo schema.....	27
2.1 pav. Galvos padėties vaizduoklio atžvilgiu nustatymo sistemos schema.....	30
2.2 pav. 4 diodų jungimo schema.....	31
2.3 pav. Techninė įranga.....	31
2.4 pav. Programos algoritmo blokinė schema.....	33
2.5 pav. Algoritmo suskirstymas į 4 blokus.....	34

2.6 pav. Matlab aplinkoje kurtos programos grafinė vartotojo sąsaja.....	35
2.7 pav. Visual C# aplinkoje kurtos programos grafinė vartotojo sąsaja.....	36
3.1 pav. Keturkampio žymėjimas.....	38
3.2 pav. Perimetrų santykis keičiant z atstumą.....	38
3.3 pav. Keturkampio kraštinių santykiai keičiant α kampą.....	39
3.4 pav. Keturkampio centro padėties kitimas priklausomai nuo α kampo.....	39
3.5 pav. Keturkampio kraštinių santykiai keičiant β kampą.....	40
3.6 pav. Keturkampio centro padėties kitimas priklausomai nuo β kampo.....	40
3.7 pav. Galvos padėties ekrano atžvilgiu parametrų skaičiavimo hierarchija.....	43
3.8 pav. Z parametro pokytis nuo 50 cm iki 500 cm ir priklausomybę atitinkanti lygtis.....	43
3.9 pav. α parametro pokyčiai lygiagrečiai keičiant z atstumą, aproksimuotų lygčių grafikai.....	44
3.10 pav. β parametro pokyčiai lygiagrečiai keičiant z atstumą, aproksimuotų lygčių grafikai.....	45
3.11 pav. x korekcijos pokyčiai lygiagrečiai keičiant z atstumą, aproksimuotų lygčių grafikai.....	46
3.12 pav. y korekcijos pokyčiai lygiagrečiai keičiant z atstumą, aproksimuotų lygčių grafikai.....	47
3.13 pav. Taškų filtravimas, keturkampio aptikimas.....	49
3.14 pav. Realus vaizdas atlikus filtravimą.....	49
3.15 pav. Filtravimo parametro įtaka taškų aptikimui.....	50
3.16 pav. Kontūrų aptikimo parametro įtaka taškų aptikimui.....	50
3.17 pav. Kontūrų jungumo parametro įtaka taškų aptikimui.....	51
3.18 pav. 4 taškų aptikimo paklaidos parametro įtaka keturkampio aptikimui.....	51
3.19 pav. Algoritmo vykdymo laikas Matlab aplinkoje.....	52
3.20 pav. Algoritmo vykdymo laikas C# windows form application aplinkoje.....	53
3.21 pav. Algoritmo vykdymo laikas C# console application aplinkoje.....	53
3.22 pav. Algoritmo vykdymo laikas C++ console application aplinkoje.....	53
3.23 pav. Algoritmų spartos palyginimas.....	54
3.24 pav. Algoritmo segmentų vykdymo laikas.....	55
3.25 pav. Algoritmo segmentų vykdymo laikas optimizavus vaizdo filtravimą.....	55
3.26 pav. Algoritmo atsitiktinės neapibrėžtys matuojant iš skirtingų pozicijų.....	56
3.27 pav. Algoritmo atsitiktinės neapibrėžtys.....	57
3.28 pav. Reikšmių optimizavimas Niutono metodu, α kampas.....	58
3.29 pav. Reikšmių optimizavimas Niutono metodu, β kampas.....	59
3.30 pav. Neapibrėžtys optimizavus sistemos reikšmes.....	59

ĮŽANGA

Šiame darbe yra modeliuojama ir tiriama galvos padėties ekrano atžvilgiu nustatymo sistema pasinaudojant kompiuterine rega.

Darbo tikslas – sumodeliuoti ir realizuoti galvos padėties erdvėje nustatymo ekrano atžvilgiu sistemą pasinaudojant kompiuterine rega.

Darbo uždaviniai:

- iširti galvos padėties nustatymo metodus;
- išskirti galvos padėčiai nustatyti naudojamus parametrus;
- iširti keturkampio pokyčius kintant stebėjimo kampui;
- nustatyti parametrų nustatymo hierarchiją;
- sukurti algoritmą galvos padėties ekrano atžvilgiu nustatymui;
- iširti algoritmo vykdymo laiką priklausomai nuo aplinkos, kurioje buvo realizuotas;
- įvertinti sistemos tikslumą.

Žmogaus ir kompiuterio sąveikos srityje yra atliekami tyrimai norint optimizuoti jau esamus įvesties, išvesties ir informacijos perdavimo metodus bei pasiūlyti naujus, inovatyvius sprendimus šioje srityje. *Žmogaus ir kompiuterio sąveika* – HCI (*angl. Human computer interaction*) – tai sistema, kai žmogus sąveikauja su mašina. Sąveika – tai informacijos perdavimas iš vieno objekto (žmogus) kitam objektui (mašina) [1]. Galvos padėtį ekrano atžvilgiu galima įvertinti kaip informaciją apie vartotoją. Siekiant perteikti šią informaciją mašinai, reikia pasinaudoti tam tikrais jutikliais, kompiuterine rega.

Galvos padėties pozicijos nustatymo sistemos panaudojimo sritys:

- 3D vaizduoklis. Vaizduoklis yra naudojamas kaip langas. Sekant informaciją apie vartotojo galvos padėtį erdvėje, yra transformuojamas vaizdas. Tokiu būdu vaizdas tampa realistiškesnis.
- Kompiuterio valdymas vartotojo galvos krypties pagalba. Sekant informaciją apie galvos padėtį, yra apskaičiuojama žiūrėjimo kryptis. Kintant šiai krypčiai, kinta ir kursoriaus padėtis. Tam tikri judesiai gali būti identifikuojami kaip vienos ar tam tikros funkcijos atlikimas.
- Kompiuterio valdymas regos pagalba, EyeWrite modifikacija. EyeWriter – tai sistema, kurios pagalba galima valdyti kompiuterio kursorių žvilgsniu. Tačiau šios sistemos trūkumas yra tas, jog galvos padėtis turi būti stabili. Apjungus galvos padėties nustatymo ir EyeWrite sistemas, rezultatas būtų įvesties/išvesties įrenginys, alternatyvus klaviatūrai ir pelei [2].

- Kompiuterinių žaidimų 3D vizualizacija. Siekiant išgauti realistiškesnį vaizdą, jis yra keičiamas priklausomai nuo vartotojo galvos padėties erdvėje. Ši sistema būtų kaip papildomas įvesties įrenginys.

Šis darbas yra suskirstytas į kelis skyrius, kuriuose pateikiami metodai galvos padėties nustatymui, parametrų galvos padėčiai nustatyti aprašymas, vaizdų apdorojimo metodika, sistemos modelis, detali algoritmo blokinė schema, sistemos spartumo ir tikslumo įvertinimo rezultatai, tyrimų rezultatai bei išvados.

Magistro darbo tema skaitytas straipsnis VGTU organizuojamoje 15-ojoje mokslinėje konferencijoje „Elektronika ir elektrotechnika 2012“. Taip pat pateiktas mokslinis straipsnis konferencijos leidiniui.

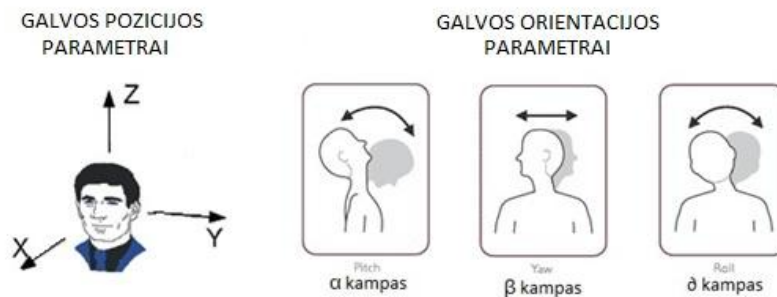
Magistro darbo tema skaitytas straipsnis ŠU organizuojamoje 7-ojoje mokslinėje konferencijoje „Studentų moksliniai darbai“.

GALVOS PADĖTIES ERDVĖJE NUSTATYMO SISTEMOS

1.1. Parametrai galvos padėčiai nustatyti

Galvos padėtis erdvėje yra nustatoma 6 parametrais. Visus šiuos parametrus galima suskirstyti į 2 grupes (žr. 1.1 pav.):

- galvos pozicijos parametrai (x , y , z koordinatės);
- galvos orientacijos parametrai (α , β , ϑ kampai).



1.1 pav. Galvos padėties nustatymo parametrai

Žmogaus galvos judesiai ir motorika turi ribotą amplitudę. Galvos orientacijos parametrai gali kisti tik iki tam tikros ribos. Ribos yra nustatytos tiriant žmogaus galvos judesius. Statistiškai yra nustatyta, jog:

- α kampo (*angl. pitch*) amplitudė yra $-100^\circ - 130^\circ$;
- β kampo (*angl. yaw*) amplitudė yra $-80^\circ - 80^\circ$;
- ϑ kampo (*angl. roll*) amplitudė yra $-90^\circ - 90^\circ$.

Galvos pozicijos parametrai neturi apribojimų [3].

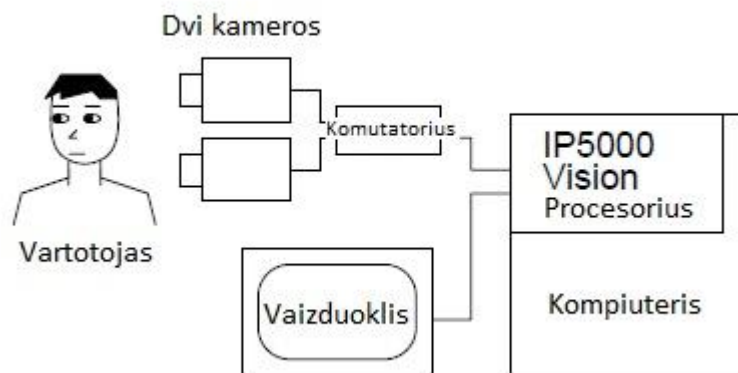
1.2. Galvos padėties erdvėje nustatymo metodai

Nagrinėti 7 skirtingi galvos padėties nustatymo metodai. Ne visi apskaičiuoja visus 6 parametrus. Metodai yra skirtingi tiek algoritmo vykdymo sparta, tiek tikslumu, tiek sistemos sandara. Taip pat ne visi nagrinėti metodai naudoja kompiuterinę regą. Vienas iš jų apskaičiuoja ieškomus parametrus mechaniškai. Pagal sandarą galima suklasifikuoti metodus į grupes:

- sekimas dvejomis kameromis:
 - stereo matymo (*angl. stereo vision*) metodas;
- mechaninis:
 - jutiklių metodas;
- sekimas viena kamera:
 - šviesos atspindžio (*angl. light source*) metodas;
 - 3D elipsoidės (*angl. 3D ellipse*) metodas;
 - išvaizdos algoritmo (*angl. appearance algorithm*) metodas;
 - paukščio skrydžio (*angl. birds eye view*) metodas;
 - cilindro tekstūros (*angl. cylinder texture map*) metodas.

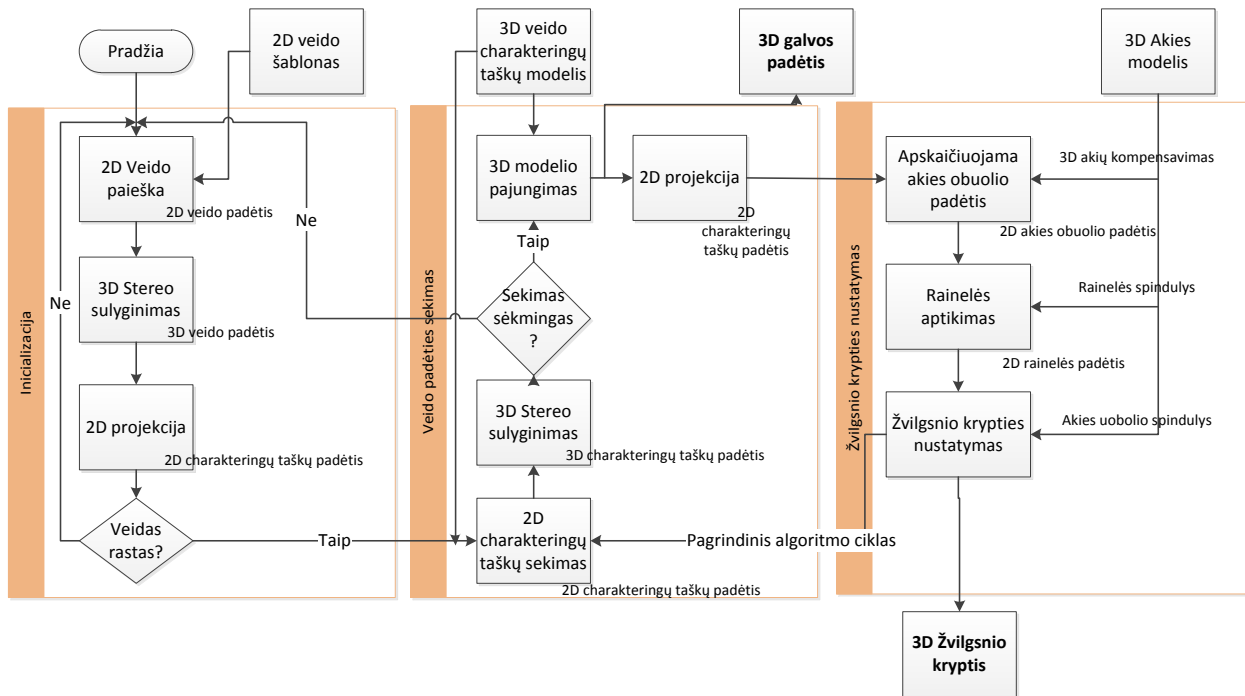
1.2.1. Stereo matymo metodas

Matsumotoy, Y. ir kt. (2000) pasiūlyto metodo realizavimui yra naudojamos dvi kameros. Sistema veikia realiu laiku. Kiekvieno ciklo vykdymo metu fiksuojami du kadrai. Gautieji vaizdai eina per komutatorių ir patenka į procesorių. Procesoriuje vyksta sulyginimo algoritmo vykdymas. Apdorotas vaizdas yra išvedamas į vaizduklį (žr. 1.2 pav.).



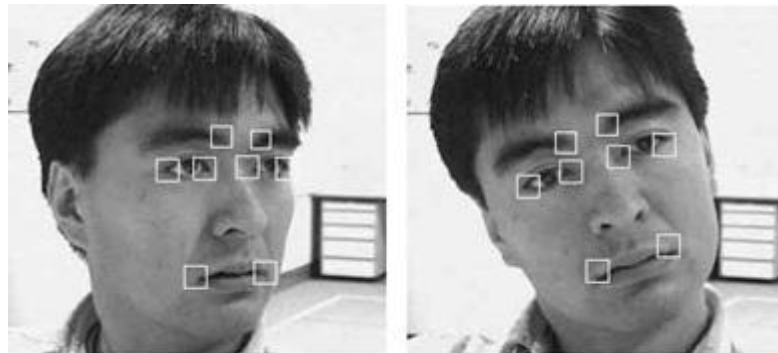
1.2 pav. Stereo matymo metodo techninės įrangos schema

Perdavus vaizdus į procesorių yra vykdomas vaizdų apdorojimas. Algoritmas yra suskirstytas į 3 dalis (žr. 1.3 pav.): inicializacija, veido padėties sekimas, žvilgsnio krypties nustatymas.



1.3 pav. Stereo matymo metodo vaizdų apdorojimo algoritmo blokinė schema

Inicijacijos dalyje 2D vaizde yra ieškomas veidas, pagal anksčiau jau sukauptus veido šablonus. Analizuojant surastus veidus, vaizduose yra vykdomas 3D stereo sulyginimas. Gaunama 3D veido padėtis erdvėje. Pagal gautus duomenis yra nustatomi charakteringi taškai 2D vaizde (žr. 1.4 pav.).



1.4 pav. Stereo matymo metodo charakteringi taškai

Veido padėties sekimo dalyje yra sekami gautieji charakteringi taškai. Jei nebėra sekami, grįžtama prie inicijacijos dalies. Iš dviejų vaizdų apskaičiuojami 3D charakteringi taškai. Iš šių taškų yra apskaičiuojama 3D galvos padėtis erdvėje.

Žvilgsnio krypties nustatymo dalyje nustatoma akies obuolio padėtis. Iš turimų duomenų apie galvos ir akies obuolio padėtį apskaičiuojama žvilgsnio kryptis. Ciklas yra vykdomas kiekvieną kadra.

Tikslumas: autorių pateiktas tikslumas galvos pozicijos parametrams yra ± 1 mm, orientacijos parametrams $\pm 1^\circ$.

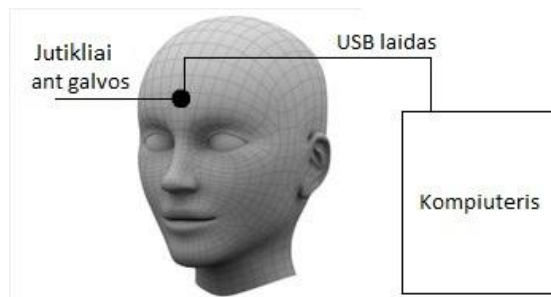
Algoritmo sparta: autorių pateikiamas algoritmo vykdymo laikas yra 30 ms/kadra [4,5].

1.2.2. Jutiklių metodas

Foxlin, E. ir kt. (2000) pasiūlytame metode nenaudojamos kameros. Visi matavimai atliekami mechaniškai pasinaudojant 3 naudojamais jutikliais: giroskopas, magnetometras, akselerometras. Viso metodui realizuoti yra naudojami 3 giroskopai, 3 magnetometrai ir 2 akselerometrai. Kiekvienas įrenginys fiksuoja tam tikrą informaciją:

- giroskopas – fiksuoja galvos orientacijos parametrus;
- magnetometras – fiksuoja kintančius magnetinius laukus aplink galvą;
- akselerometras – fiksuoja galvos judesių pagreitį.

Apjungus šiuos įrenginius mes gauname InterTrax 2 galvos sekimo prietaisą (žr. 1.5 pav.). Kiekvienas vidinis įrenginys fiksuoja skirtingus parametrus (posūkio kampus, pagreičius, magnetinio lauko pokyčius). Šis įrenginys perduoda informaciją į kompiuterį USB sąsaja. Kompiuteryje informacija yra apdorojama. Gaunami galvos orientacijos parametrai erdvėje. Šio metodo trūkumas tas, jog jis neapskaičiuoja galvos pozicijos parametrų.



1.5 pav. Jutiklių metodo techninės įrangos išdėstymas

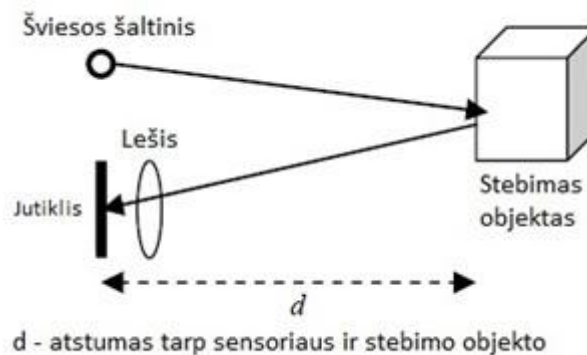
Tikslumas: autorių pateikiamos metodo neapibrėžtys, nustatant orientacijos parametrus, yra $\pm 0,02^\circ$.

Judesių amplitudės ribos: autorių pateikiamos ribos, kuriose fiksuojami galvos orientacijos parametrai: $\alpha \pm 80^\circ$, $\beta \pm 180^\circ$, $\vartheta \pm 90^\circ$ [6].

1.2.3. Šviesos atspindžio metodas

Göktürk, B. S. ir kt. (2004) pasiūlytame metode yra naudojama šviesai jautri kamera (*angl. Time-of-flight camera*). Metodas sukurtas remiantis veidų atpažinimu ir koreliacija paremta svorių interpoliacija.

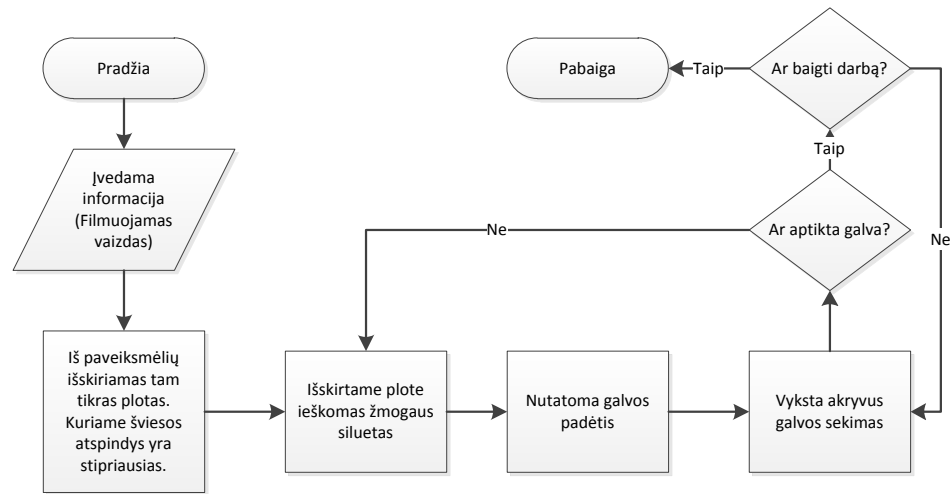
Į stebimą objektą yra šviečiama šviesos šaltiniu. Kiekvienas objektas dalį šviesos absorbuoja ir dalį atspindi. Šviesai jautri kamera fiksuoja šviesos atspindžius. Informacija perduodama į kompiuterį. Gauti vaizdai yra apdorojami (žr. 1.6 pav.).



1.6 pav. Šviesos atspindžio metodo veikimo principas

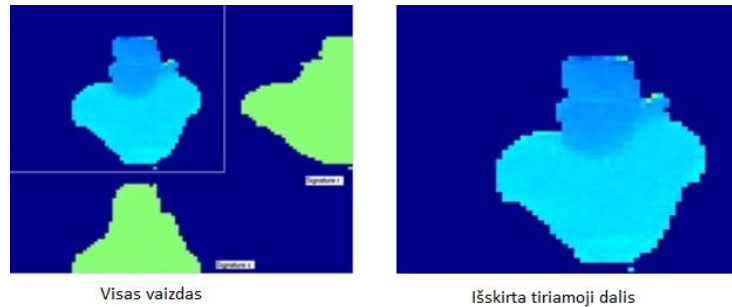
Šviesai jautrios kameros veikimo principas yra tas, jog gaunami vaizdai analizuojami tam, kad iš jų būtų galima gauti tikslią informaciją apie objektus ir atstumus nuo jų. Kamera fiksuoja šviesos atspindžius, jų stiprumą. Pagal jį yra apskaičiuojamas atstumas iki objekto. Metodas nustato galvos pozicijos parametrus, tačiau nenustato orientacijos parametrų. Išskaičiuojama gylio matrica.

Gautieji duomenys apdorojami kompiuterio pagalba. Vykdomas vaizdų apdorojimo algoritmas (žr. 1.7 pav.). Kuo atspindys intensyvesnis, tuo objektas yra arčiau kameros. Metodo tikslumas labai priklauso nuo aplinkos, kurioje jis vykdomas. Jei stebėjimo zonoje yra objektai, kurių atspindžio koeficientas didesnis nei žmogaus odos, tai rezultatai gali būti iškraipyti. Algoritmo tikslumui padidinti yra naudojamas šviesos atspindžių filtravimas. Iš vaizdo panaikinami itin stiprūs ir itin silpni atspindžiai (žr. 1.8 pav.).



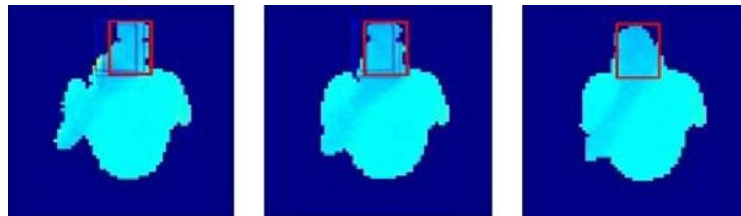
1.7 pav. Šviesos atspindžio metodo vaizdų apdorojimo algoritmo blokinė schema

Nufiltravus vaizdą, vykdomas galvos parametrų radimo algoritmas. Išskyrus interesų regioną (*angl. ROI – region of interest*), yra apskaičiuojama gylio matrica. Gylio matrica lyginama su ankstesnių eksperimentų duomenimis. Apskaičiuojama koreliacija. Randama maksimali koreliacija. Išskiriama vieta vaizde, kurioje yra žmogaus galva, ir apskaičiuojamas atstumas iki jos.



1.8 pav. Šviesos atspindžio metodo vaizdo filtravimas

Likę galvos pozicijos parametrai yra apskaičiuojami interpoliuojant gylio matricą, naudojant koreliacijos koeficientų matricą kaip svorius. Algoritmo vykdomo metu yra sekama galvos padėtis vaizde (žr. 1.9 pav.).



1.9 pav. Šviesos atspindžio metodo galvos padėties sekimas realiu laiku

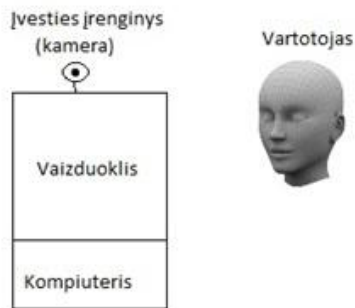
Autorių teigimu, bandymai atlikti galvą sukant 180 laipsnių kampu. Įtakos sekimui tai nedaro. 90% viso sekimo laiko algoritmas sugebėjo nustatyti galvos padėtį. Jei kažkuriuo momentu nutrūkdavo algoritmo tęstinumas dėl techninių priežasčių (jutiklis nebegavo signalo, šviesos šaltinis buvo uždengtas ir t. t.), ištaisius technines kliūtis sekimas buvo tęsiamas.

Tikslumas: Autorių teigimu, atliktų bandymų metu 70% laiko galvos padėtis buvo nusakoma 1 Pikselio tikslumu, skaičiuojant x ir y parametrus. Apie Z ašies tikslumą nėra minima.

Algoritmo sparta: autorių pateikiamas algoritmo vykdymo laikas yra 30 kadru/s [7].

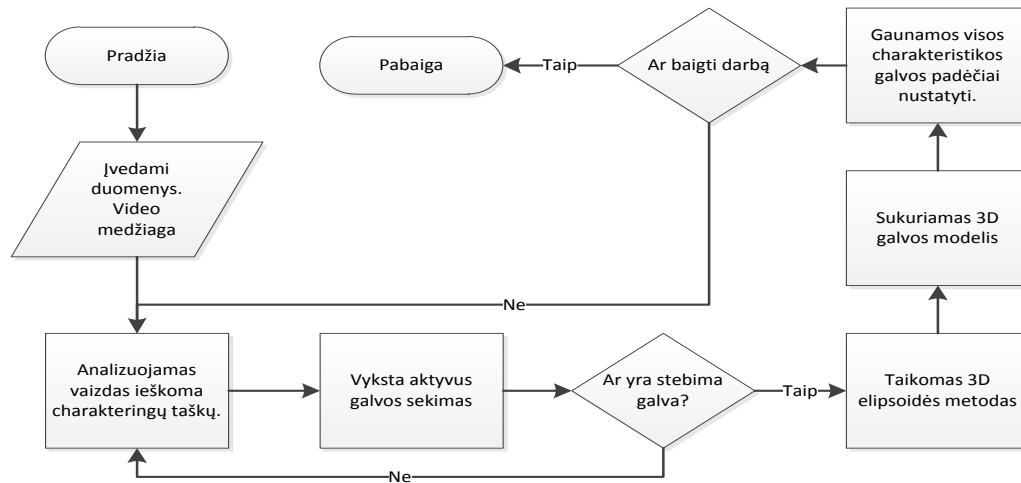
1.2.4. 3D Elipsoidės metodas

Basu, S. ir kt. (1996) pasiūlytame metode naudojama viena kamera. Galvos padėties parametrai yra nustatomi kompiuterinės regos pagalba. Techninė įranga naudojami metodui realizuoti: video kamera, kompiuteris, vaizduoklis (žr. 1.10 pav.).

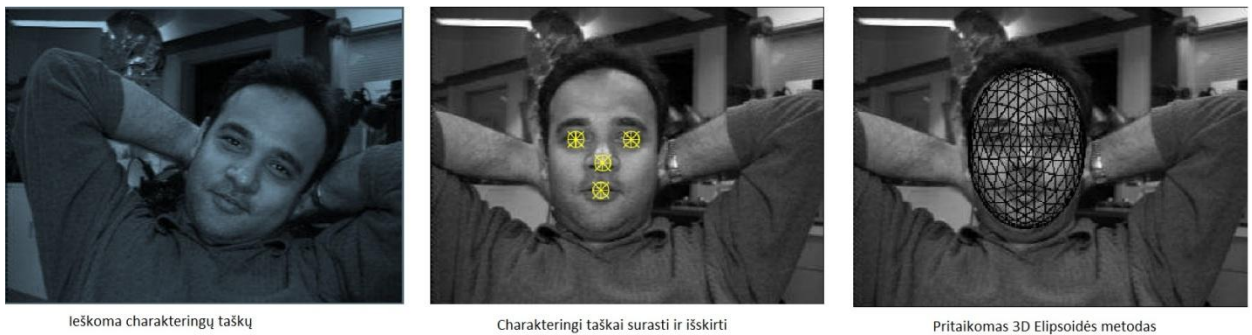


1.10 pav. Vienos kameros sekimo metodo techninės įrangos išdėstymas

Vykdomas vaizdo apdorojimo algoritmas (žr. 1.11 pav.). Analizuojamas 2D vaizdas. Vaizde yra ieškomi charakteringi taškai. Pagal charakteringus taškus nustatoma veido padėtis 2D vaizde. Radus veido padėtį vaizde, yra taikomas 3D elipsoidės metodas. Metodo pagalba sudaroma 3D veido reprodukcija. Iš reprodukcijos yra nustatomi galvos pozicijos ir orientacijos parametrai. Gautasis vaizdas išvedamas į ekraną (žr. 1.12 pav.).



1.11 pav. 3D elipsoidės metodo vaizdo apdorojimo blokinė schema



1.12 pav. 3D elipsoidės metodo vaizdo apdorojimas

Tikslumas: Autorių teigimu, orientacijos parametrų nustatymas vyksta su 10° neapibrėžtimi. Apie pozicijos parametrų tikslumą autoriai nemini.

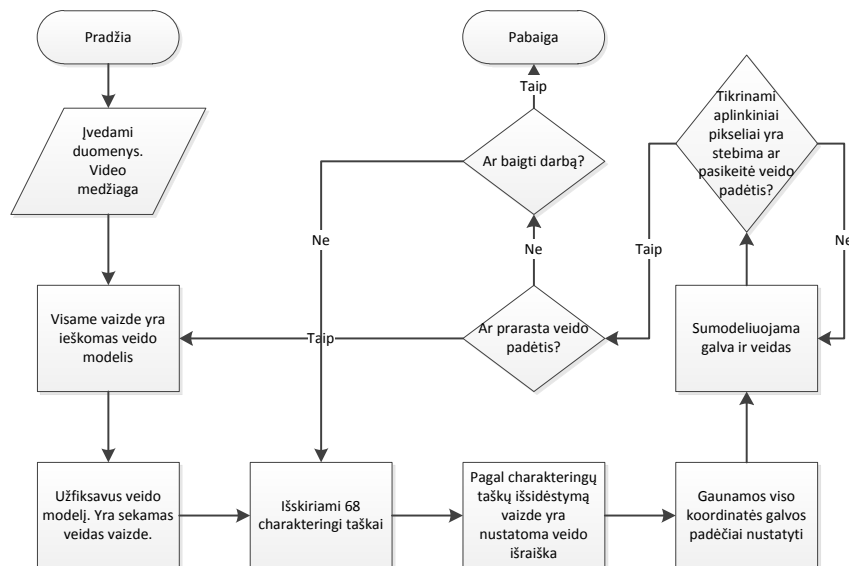
Algoritmo sparta: autorių pateikiamas algoritmo vykdymo laikas yra 30 kadru/s [8].

1.2.5. Išvaizdos algoritmo metodas

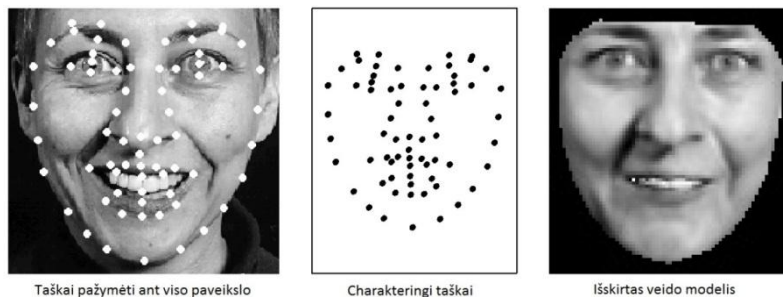
Cootes, F. T., Ahlberg J. ir kt. (2001) pasiūlytas metodas sukurtas remiantis išvaizdos algoritmo metodu. Metodui realizuoti reikalinga techninė įranga: kompiuteris, video kamera (žr. 1.10 pav.). Taip pat reikalinga ir programinė įranga: programa realizuojant išvaizdos algoritmą (žr. 1.13 pav.).

Veido paieška yra vykdoma aktyvių išvaizdos modelių (*angl. Active Appearance models*) principu, kuris buvo pasiūlytas 2001 Timothy F. Cootes. Veido paieška vykdoma tokiu principu: pirmiausia tiriamas visas vaizdas; vaizde ieškomas veido modelis; modelyje ieškomi charakteringi

taškai. Šio metodo pagalba fiksuojamos veido išraiškos, tad charakteringų taškų yra daugiau nei kituose nagrinėtuose metoduose. Pateiktame pavyzdyje užfiksuoti 68 charakteringi taškai (žr. 1.14 pav.).

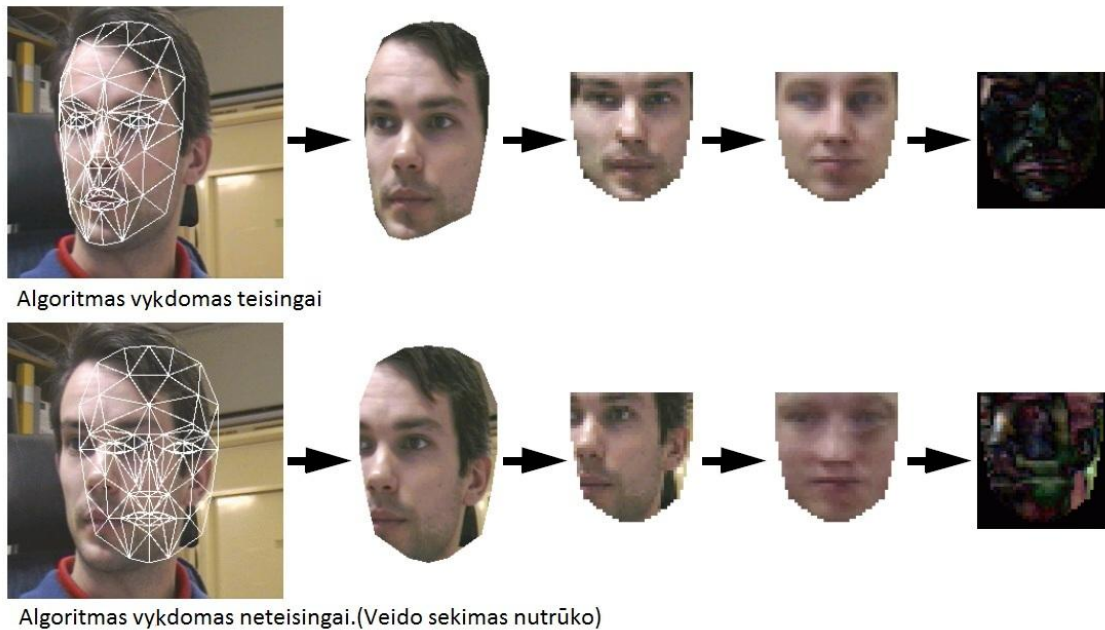


1.13 pav. Išvaizdos algoritmo vaizdų apdorojimo algoritmo blokinė schema



1.14 pav. Aktyvių išvaizdos modelių metodas

Suradus visus charakteringus taškus, jie yra sekami tikrinant jų postūmį erdvėje. Tikrinami artimi pikseliai, ieškoma kur pasislinko charakteringi taškai. Informacija apie galvos poziciją ir orientaciją yra saugoma dviejuose masyvuose. Viename saugoma x , y ir z koordinatės, o kitame – α , β , ϑ posūkio kampai. Turint šiuos duomenis, galima nustatyti galvos padėtį erdvėje. Pagrindinis šio metodo trūkumas – greitų judesių fiksavimas. Jei judesiai yra greitesni nei gali juos apdoroti algoritmas, sekimas yra nutraukiamas (žr. 1.15 pav.). Tuomet algoritmas yra vykdomas iš naujo.



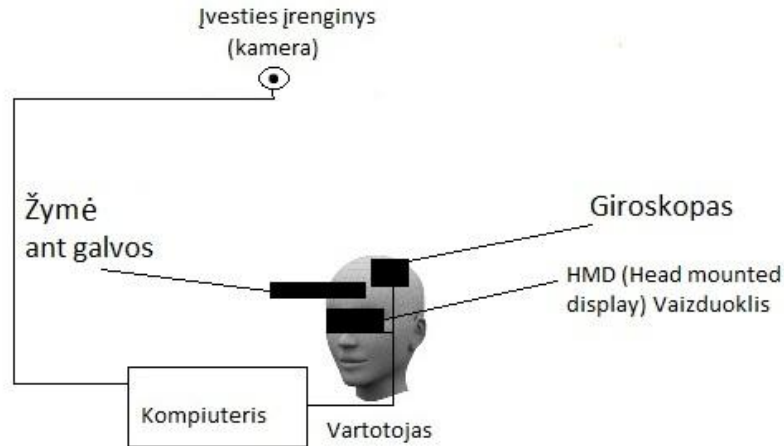
1.15 pav. Aktyvių išvaizdos modelių metodo per greito judesio apdorojimas

Tikslumas: autoriai apie metodo tikslumą informacijos nepateikia.

Algoritmo sparta: autorių pateikiamas algoritmo vykdymo laikas yra 14 ms/kadrą [9,10].

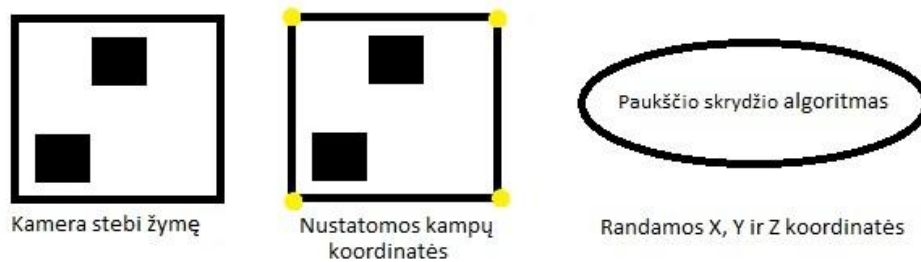
1.2.6. Paukščio skrydžio metodas

Satoh, K. ir kt. (2004) pasiūlytas metodas yra sudarytas pasinaudojant paukščio skrydžio principu. Metodo išskirtinumas tas, jog video kamera nėra prieš vartotoją, o stebi jį iš viršaus. Šiam metodui realizuoti reikalinga techninė įranga (žr. 1.16 pav.): kompiuteris, apdorojantis visą informaciją, video kamera, giroskopas, žymė ir vaizduoklis. Metodui realizuoti reikalinga programinė įranga: algoritmas, apskaičiuojantis galvos pozicijos parametrus, pasinaudojant paukščio skrydžio metodu, bei nuskaitantis kampus iš giroskopo.



1.16 pav. Paukščio skrydžio metodo techninės įrangos išdėstymas

Šio metodo pagalba nustatomi visi 6 parametrai, reikalingi galvos padėčiai nustatyti. Galvos orientacijos parametrai nustatomi giroskopo pagalba. Visa informacija iš giroskopo yra siunčiama į kompiuterį. Galvos pozicijos parametrai nustatomi kameros ir žymės ant galvos pagalba. Kamera stebi vartotoją iš viršaus. Yra fiksuojami vaizdai ir ieškoma žymės. Žymės fiziniai parametrai žinomi iš anksto. Pagal žymės dydį ir padėtį vaizde yra nustatomi galvos pozicijos parametrai erdvėje (žr. 1.17 pav.).



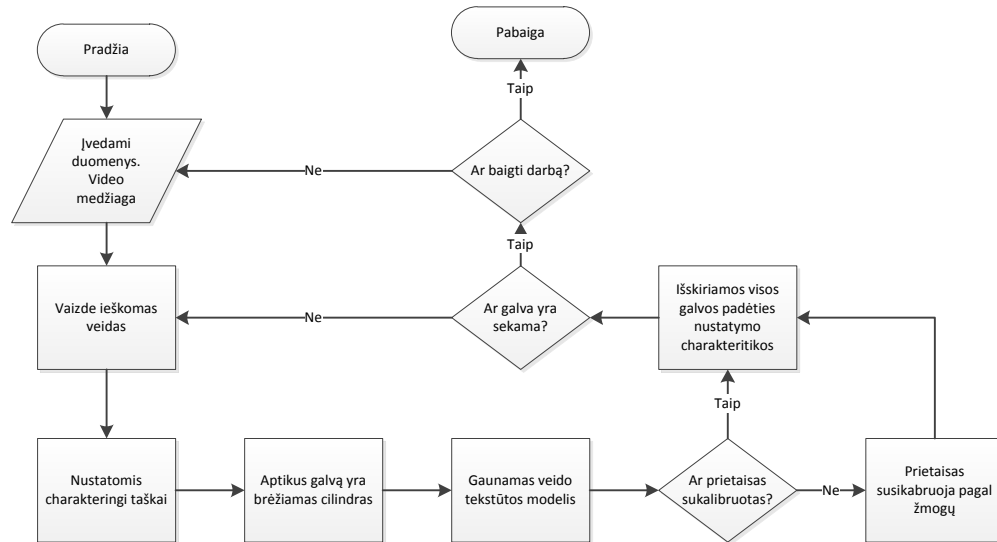
1.17 pav. Paukščio skrydžio metodo galvos pozicijos parametų apskaičiavimo principas

Algoritmas suranda žymę vaizde ir pagal kampų koordinates nustatoma jos pozicija erdvėje. Žymė yra ant galvos. Galvos padėtis tikslinama pagal ankstesnių tyrimų korekcijas.

Tikslumas: autorių teigimu, galvos orientacijos parametrai nustatomi giroskopu, kurio neapibrėžtis yra 1° . Galvos pozicijos parametrai nustatomi iš vaizdo kadro su 3–5 pikselių neapibrėžtimi. Filmuojama iš 1,5 m atstumo. Kameros raiška yra 640x480 [11].

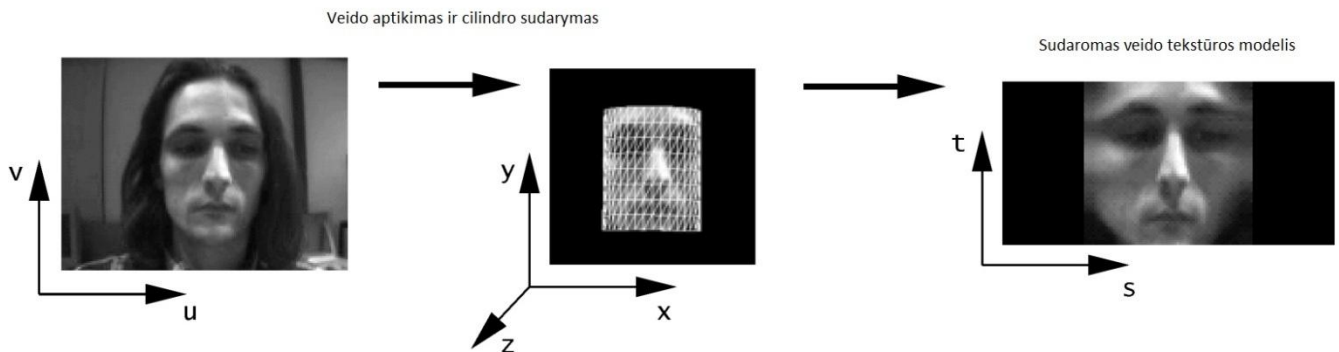
1.2.7. Cilindro tekstūros metodas

La Cascia, M. ir kt. (2000) metodas yra sudarytas naudojantis cilindro tekstūros metodu. Techninė įranga, reikalinga metodui realizuoti (žr. 1.10 pav.): kompiuteris, vaizdą fiksuojanti kamera, vaizduoklis. Programinė įranga: vaizdų apdorojimo algoritmas, realizuojantis cilindro tekstūros metodą (žr. 1.18 pav.).



1.18 pav. Cilindro tekstūros metodo algoritmo blokinė schema

Algoritmo pradinė dalis yra kalibracija. Ieškomas veidas 2D vaizde. Veidas aptinkamas radus charakteringus taškus. Susiejus visus charakteringus taškus į visumą, yra nustatoma veido padėtis vaizde. Aptikus veidą, ši vaizdo dalis iškerpama. Sudaromas 3D cilindras, kuris padengiamas iškirptu vaizdu kaip tekstūra. Nuskaičius vaizdą nuo cilindro paviršiaus, yra gaunamas veido tekstūros modelis (*angl. Cylinder texture map*) (žr. 1.19 pav.).



1.19 pav. Cilindro tekstūros modelio gavimo pavyzdys

Kiekvieno ciklo metu yra sudaromas naujas cilindro tekstūros modelis. Naujasis modelis lyginamas su prieš tai buvusi. Iš gautų pasikeitimų apskaičiuojami galvos padėties pokyčiai. Būtina sąlyga ta, kad gali dirbti tik vienas vartotojas. Jei darbą pradeda kitas žmogus, reikia išvalyti visus sugeneruotus modelius. Apšvietimas neturi įtakos sistemos veikimui.

Autorių hipotezė – gautasis modelis turės tam tikras paklaidas, kadangi galvos tapatinimas su cilindru neatitinka teisybės. Tačiau algoritmas turėtų būti vykdomas galvai sukantis net 360° kampu bei judant visomis trimis kryptimis greičiau nei bet kuris kitas algoritmas.

Tikslumas: autorių teigimu, nustatant pozicijos parametrus neapibrėžtis buvo $\pm 7,5$ cm, o skaičiuojant orientacijos parametrus – $\pm 5^\circ$ [12].

1.3. Pasirinkta vaizdų apdorojimo metodika

Kuriant vaizdo apdorojimo algoritmą buvo išanalizuoti metodai vaizdo apdorojimui. Iš nagrinėtų metodų buvo pasirinkti geriausiai tinkantys, sparčiausiai veikiantys ir labiausiai pritaikyti šio algoritmo realizavimui:

- Komporavimas (*angl. Binary threshold*);
- Sobel operatorius;
- Kraštų aptikimas (*angl. Edge detection*);
- Kontūrų aptikimas (*angl. Find contours*);
- Subpikselinių verčių nustatymo metodas;
- Kalmano filtras (*angl. Kalman filter*);
- Niutono optimizavimo metodas.

Pasirinkti metodai naudojami informacijos kiekiui vaizde sumažinimui komporavimas ir kraštų nustatymas, kontūrų aptikimui ir atskyrimui, centrinių verčių pikselių lauke suradimui, sekamų objektų judėjimo trajektorijos prognozavimui bei tikslinimui Kalmano filtras, gautųjų apskaičiuotų verčių patikslinimui Niutono optimizavimo metodas[13,14].

1.3.1. Komporavimas

Komporavimas metodas yra naudojamas vaizdo filtravimui. Jis naudojamas reikiamo ryškumo segmentų išskyrimui, fono panaikinimui. Metode yra naudojami du parametrai: apatinė ir viršutinė riba. Apatinė riba nustato kokia minimali pikselio ryškumo vertė neturi būti filtruojama. Viršutinė riba

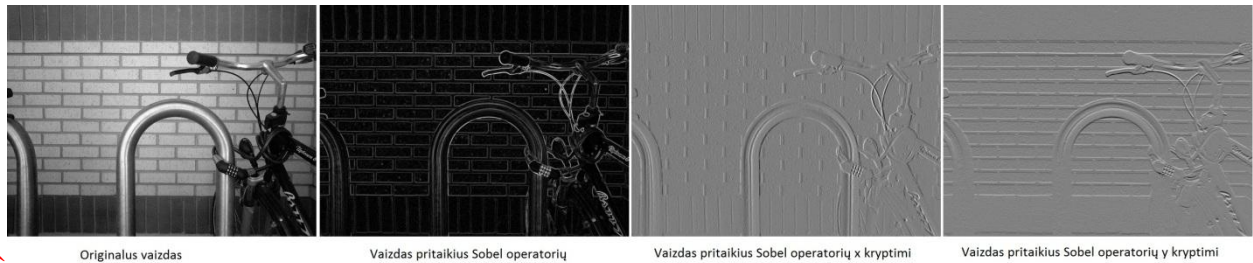
nusako kokia maksimali. Pikseliai patenkantys į nustatytas ribas yra pažymimi 1 ir nudažomi balta spalva, o kiti pikseliai – 0 ir nudažomi juoda spalva (žr. 1.20 pav.) [15].



1.20 pav. Komporavimo pavyzdys

1.3.2. Sobel operatorius

Sobel operatorius yra naudojamas vaizdų apdorojimo srityje. Tai vienas iš metodų, nustatančių vaizdo ryškumo pokyčius (*angl. Gradient*). Šio operatoriaus tikslas – tikrinti perėjimo iš tamsaus pikselio į šviesų (ir atvirkščiai) intensyvumą x ir y kryptimi (žr. 1.21 pav.).



1.21 pav. Sobel operatoriaus pavyzdys

Šio metodo matematiniam modelyje naudojama 3x3 sąsūkų branduolių pora, kuri naudojama išvestinių horizontalia ar vertikalia kryptimi apskaičiuoti. Išvestinių apskaičiavimo formulė (1.1). Ryškumo vertės vaizde apskaičiuojamos pagal formulę (1.2). Šis operatorius yra naudojamas kontūrų aptikimo algoritme kaip priemonė nustatyti ryškumo pokytį [13,16].

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A; \quad (1.1)$$

čia A – pradinis vaizdas;
 G_x – išvestinės x kryptimi;
 G_y – išvestinės y kryptimi.

$$G = \sqrt{G_x^2 + G_y^2}; \quad (1.2)$$

čia G – ryškumo vertės vaizde.

1.3.3. Kraštų aptikimas

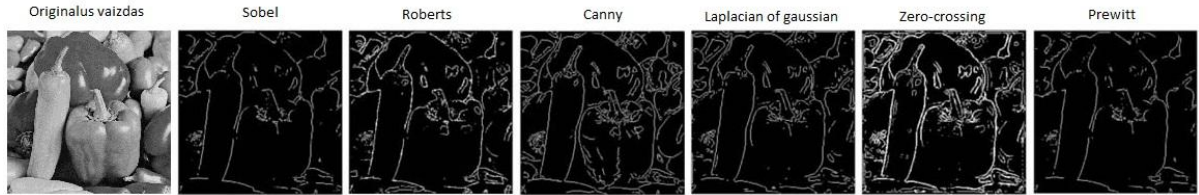
Kraštų aptikimas (*angl. Edge detection*) yra vienas pagrindinių metodų vaizdų apdorojimo srityje. Metodo esminis principas – aptikti vietas vaizde, kuriose yra kontūrai. Kontūras – tai tokia vieta vaizde, kurioje pikselio vertės perėjimo pokytis iš vieno pikselio į kitą yra aštrus. Idealiu atveju pritaikius šį metodą, pateikiami visi vaizde esantys objektų kraštai, tačiau metodas yra įtakojamas nuo apšvietimo. Jei apšvietimas yra per stiprus ar per silpnas, gali būti gaunama netiksli informacija. Kontūrų aptikimas gali būti naudojamas nustatyti geometrines vaizdo savybes.

Kontūrų paieškos metodai gali būti suskirstyti į grupes pagal kraštinių nustatymo metodo principą:

- paieškos pagrindo (*angl. search-based*);
- nulio kirtimo (*angl. zero-crossing based*).

Kontūrų nustatymo metodai skiriasi pagal kontūrų aptikimo metodą, t.y. jo griežtumą. Vieni metodai aptinka daugiau kontūrų, kiti – mažesnę kiekį. Visa tai priklauso nuo to, koks yra naudojamas ryškumo perėjimo algoritmas. Ryškumo perėjimo algoritmų metodai (žr. 1.22 pav.):

- Sobel;
- Prewitt;
- Roberts;
- Laplacian of gaussian;
- Zero – cross;
- Canny.



1.22 pav. Ryškumo perėjimo algoritmų metodai

Suradus kontūrus vaizde, panaikinami nereikalingi bei netinkantys pagal prasmę. Tai atliekama ne maksimumų pašalinimu (*angl. non-maximum suppression*) – ieškoma, ar pokyčio reikšmė yra lokalusis maksimumas.

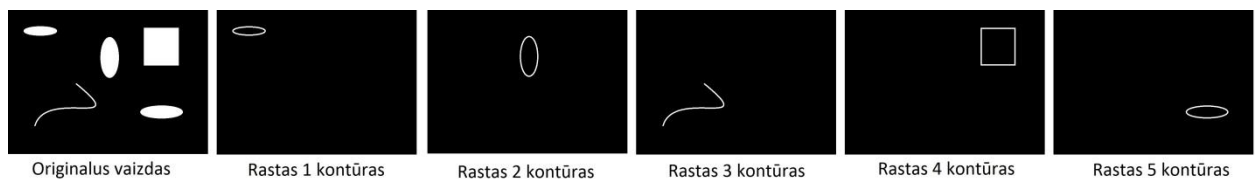
Nustatant vaizdo ir histezinės slenkstinius lygius, galimas kontūrų ryškumo filtravimas. Pritaikomi du slenksčiai. Vienas apriboja galimą maksimalų kontūro intensyvumą, o kitas – minimalų [14,17].

1.3.4. Kontūrų aptikimo

Kontūrų aptikimo (*angl. Find contour*) metodas (žr. 1.23 pav.) yra naudojamas atskirti skirtingas linijas vaizde. Šis metodas plačiai taikomas vaizdų analizėje. Kontūrų, linijų aptikimas yra vykdomas ieškant vaizde atskirų linijų. Kiekviena rasta jungi linija saugoma masyve kaip atskiras elementas. Kiekvienas kontūras yra uždara, su kitais kontūrais nejungi linija. Analizuojant kontūrus, galima rasti ieškomus objektus vaizde, o nereikalingus atmesti.

Metodo gaunama informacija apie kontūrą:

- užimamas plotas vaizde;
- minimalus keturkampis, kuriame galima įpiešti kontūrą;
- kontūro perimetras.

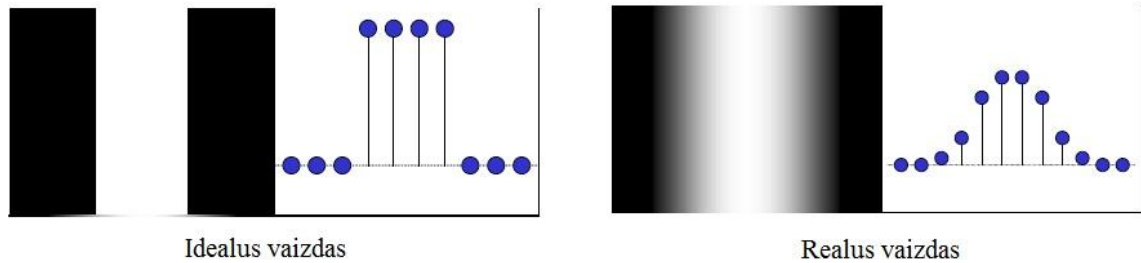


1.23 pav. Kontūrų radimo metodo pavyzdys

Kontūrų radimas yra naudojamas rūšiuoti, atskirti, išmatuoti objektus, esančius vaizde. Taip pat gali būti taikomas įvertinant objekto geometrines savybes [14].

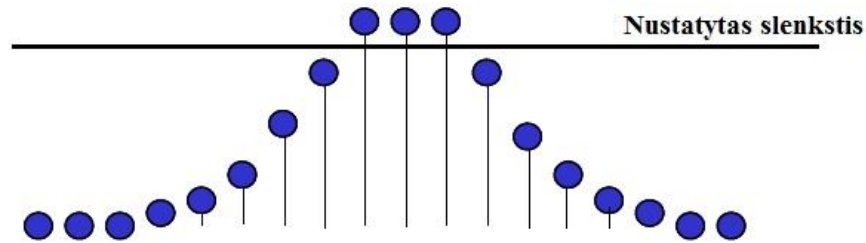
1.3.5. Subpikselineių verčių nustatymo metodas

Siekiant patikslinti rezultatus, galima skaičiuoti subpikselineis (*angl. subpixel value*) vertėmis. Standartinė pikselio sąvoka yra tokia: pikselis – tai mažiausia vaizdo sudedamoji dalis. Tačiau siekiant patikslinti skaičiavimus, galima naudoti subpikselineis vertes. Tarkime, turime kontūrą, kuris sudarytas iš 2 taškų. Taškų koordinatės yra $[1;1]$ ir $[2;1]$. Kaip įvertinti vidurinę kontūro tašką, jei matuojama pikseliais? Centrinis taškas bus vienas iš taškų, tačiau tai nėra teisinga. Nustatant subpikselineis vertes, centrinis taškas yra $[1,5;1]$. Subpikselineis vertės yra dažnai naudojamos kai vaizde yra daugiau ne dviejų skirtingų ryškumų pikseliai. T.y. jei vaizde ryškumo tono pokyčiai yra netolygūs. Idealiu atveju vaizde yra tik 2-viejų tonų pikseliai. Idealiu atveju subpikselineis vertės nėra skaičiuojamos (žr. 1.24 pav.).



1.24 pav. Vertikalios pikselių ryškumo histogramos

Darbe naudojamo diodo centro aptikimo metodas. Randama diodo centrinė vertė pikselio tikslumu. Nustatomas interesų regionas, kurio ribos yra $n \times n$. Interesų regione ieškoma aštrių ryškumo pokyčių pagal nustatytą slenkstį (žr. 1.25 pav.). Slenkstis yra nustatomas vartotojo pagal jo poreikius. Tikrinant x kryptimi, randamos šviesos diodo spindulio ribos iš kairės X_1 ir dešinės X_2 . Analogiškai yra randamos y kryptimi – ribos iš viršaus Y_1 ir iš apačios Y_2 . Atikus ribų taškus ir centrinę tašką esantį tarp jų, apskaičiuojamas visų taškų vidurkis x ir y koordinatėms atskirai. Vertės yra gaunamos subpikselineiu tikslumu.



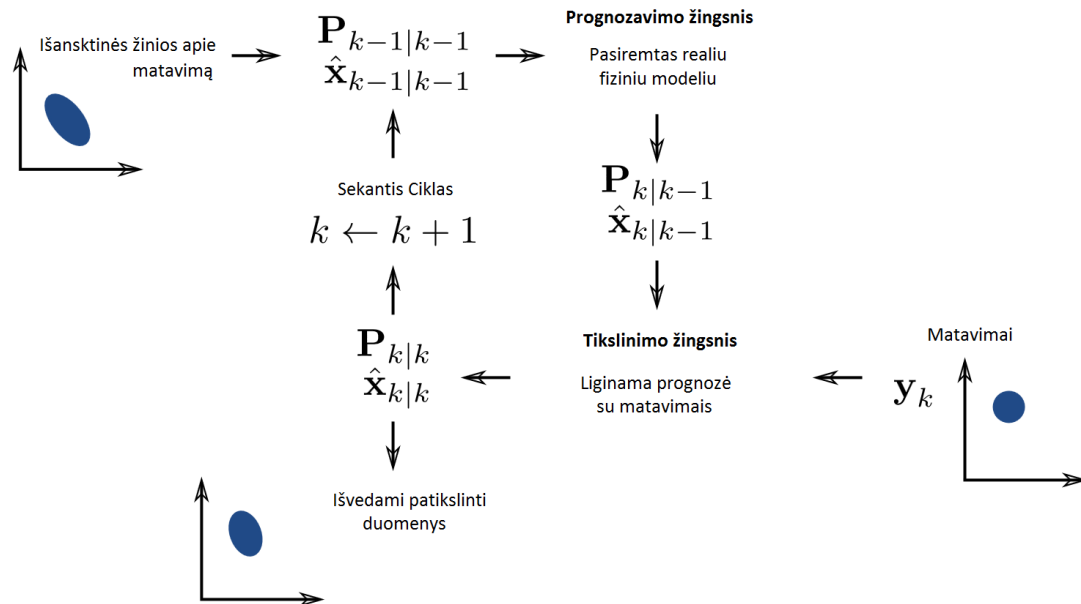
1.25 pav. Subpikselineių verčių nustatymas

Šio metodo panaudojimas, atliekant tikslius skaičiavimus, sumažina neapibrėžtis. Tačiau algoritmo vykdymo laikas nėra sumažinamas [18].

1.3.6. Kalmano filtras

Kalmano filtras kitaip vadinamas linijiniu kvadratinu vertinimu – LQE (*angl. linear quadratic estimation*). Šio algoritmo pritaikymų yra labai daug technologijos srityje. Pirmą kartą šį metodą pasiūlė Rudolf E. Kalman. Tai yra algoritmas, kurio pagalba įvairūs matavimai yra patikslinami, bei prognozuojami tolimesni verčių pokyčiai. Šio algoritmo pagalba galima pašalinti įvairius mechaninius triukšmus.

Kalmano filtro veikimą galima suskirstyti į dvi dalis (žr. 1.26 pav.): prognozavimas ir tikslinimas. Jis analizuoja prieš tai buvusią informaciją bei tikslina naujus duomenis, remiantis jau ankščiau buvusiais matavimais. Jei tam tikras triukšmas yra pastovus, kalmano filtras padeda to išvengti. Jis yra plačiai naudojamas realaus laiko vaizdo apdorojimo algoritmų kūryboje. Taip yra dėl to, jog kalmano filtras pasižymi spartumu ir nereikalauja daug kompiuterio atminties. Duomenys reikalingi darbui: realaus matavimo duomenys, prieš tai buvusio matavimo duomenys, apskaičiuotos konstantos, stebint objektą per tam tikrą laiką.



1.26 pav. Kalmano filtro veikimo schema

Tai vienas dažniausiai naudojamų metodų objektų sekimui vaizde. Neigiamas jo aspektas yra tas, kad, jei sekamas objektas nebejudą, kalmano filtras veikia pagal anksčiau veikusį modelį, ir tada atliekamas judesys, kurio realiai nėra [14,19].

1.3.7. Niutono optimizavimo metodas

Niutono optimizavimo metodas padeda patikslinti jau turimas reikšmes. Šis metodas kiekvienos iteracijos metu tikslina reikšmes, kurios yra priklausomos nuo tam tikrų lygčių. Tarkime yra apskaičiuotos pradinės tiesinių poslinkių reikšmės: x , y , z ir kampų reikšmės: α , β , ϑ . Šias reikšmes įstatę į 8 funkcijas $f_i(x, y, z, \alpha, \beta, \vartheta)$ ($i=1, 2, \dots, 8$), gauname keturkampio kampų koordinatas vaizde: $x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4$. Turime 8 lygčių sistemą:

$$\begin{cases} f_1(x, y, z, \alpha, \beta, \vartheta) - x_1 = 0 \\ f_2(x, y, z, \alpha, \beta, \vartheta) - x_2 = 0 \\ f_3(x, y, z, \alpha, \beta, \vartheta) - x_3 = 0 \\ f_4(x, y, z, \alpha, \beta, \vartheta) - x_4 = 0 \\ f_5(x, y, z, \alpha, \beta, \vartheta) - y_1 = 0 \\ f_6(x, y, z, \alpha, \beta, \vartheta) - y_2 = 0 \\ f_7(x, y, z, \alpha, \beta, \vartheta) - y_3 = 0 \\ f_8(x, y, z, \alpha, \beta, \vartheta) - y_4 = 0 \end{cases} \quad (1.3)$$

Tačiau dažniausiai 1.3 formulė nėra teisinga. Siekiant patikslinti šias reikšmes yra sudaroma 1.4 lygčių sistema. Sistema yra sprendžiama mažiausių kvadratų metodu. Apskaičiuojamos visos pokyčių Δ reikšmės.

$$\begin{cases} (f_1(x, y, z, \alpha, \beta, \vartheta) + \frac{df_1}{dx} \Delta x + \frac{df_1}{dy} \Delta y + \frac{df_1}{dz} \Delta z + \frac{df_1}{d\alpha} \Delta \alpha + \frac{df_1}{d\beta} \Delta \beta + \frac{df_1}{d\vartheta} \Delta \vartheta - x_1)^2 = 0 \\ (f_2(x, y, z, \alpha, \beta, \vartheta) + \frac{df_2}{dx} \Delta x + \frac{df_2}{dy} \Delta y + \frac{df_2}{dz} \Delta z + \frac{df_2}{d\alpha} \Delta \alpha + \frac{df_2}{d\beta} \Delta \beta + \frac{df_2}{d\vartheta} \Delta \vartheta - x_2)^2 = 0 \\ (f_3(x, y, z, \alpha, \beta, \vartheta) + \frac{df_3}{dx} \Delta x + \frac{df_3}{dy} \Delta y + \frac{df_3}{dz} \Delta z + \frac{df_3}{d\alpha} \Delta \alpha + \frac{df_3}{d\beta} \Delta \beta + \frac{df_3}{d\vartheta} \Delta \vartheta - x_3)^2 = 0 \\ (f_4(x, y, z, \alpha, \beta, \vartheta) + \frac{df_4}{dx} \Delta x + \frac{df_4}{dy} \Delta y + \frac{df_4}{dz} \Delta z + \frac{df_4}{d\alpha} \Delta \alpha + \frac{df_4}{d\beta} \Delta \beta + \frac{df_4}{d\vartheta} \Delta \vartheta - x_4)^2 = 0 \\ (f_5(x, y, z, \alpha, \beta, \vartheta) + \frac{df_5}{dx} \Delta x + \frac{df_5}{dy} \Delta y + \frac{df_5}{dz} \Delta z + \frac{df_5}{d\alpha} \Delta \alpha + \frac{df_5}{d\beta} \Delta \beta + \frac{df_5}{d\vartheta} \Delta \vartheta - y_1)^2 = 0 \\ (f_6(x, y, z, \alpha, \beta, \vartheta) + \frac{df_6}{dx} \Delta x + \frac{df_6}{dy} \Delta y + \frac{df_6}{dz} \Delta z + \frac{df_6}{d\alpha} \Delta \alpha + \frac{df_6}{d\beta} \Delta \beta + \frac{df_6}{d\vartheta} \Delta \vartheta - y_2)^2 = 0 \\ (f_7(x, y, z, \alpha, \beta, \vartheta) + \frac{df_7}{dx} \Delta x + \frac{df_7}{dy} \Delta y + \frac{df_7}{dz} \Delta z + \frac{df_7}{d\alpha} \Delta \alpha + \frac{df_7}{d\beta} \Delta \beta + \frac{df_7}{d\vartheta} \Delta \vartheta - y_3)^2 = 0 \\ (f_8(x, y, z, \alpha, \beta, \vartheta) + \frac{df_8}{dx} \Delta x + \frac{df_8}{dy} \Delta y + \frac{df_8}{dz} \Delta z + \frac{df_8}{d\alpha} \Delta \alpha + \frac{df_8}{d\beta} \Delta \beta + \frac{df_8}{d\vartheta} \Delta \vartheta - y_4)^2 = 0 \end{cases} \quad (1.4)$$

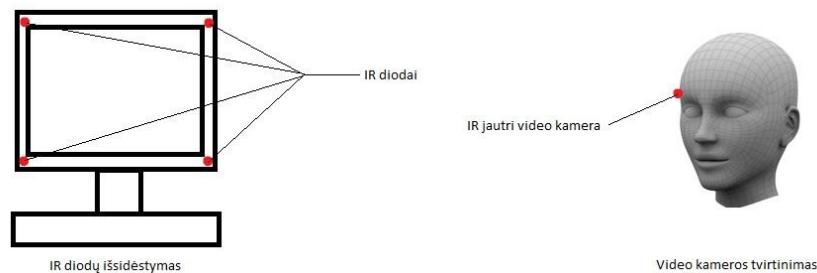
Pagal 1.5 formules yra apskaičiuojamos patikslintos ieškomų parametrų reikšmės.

$$\begin{aligned}x &= x + \Delta x \\y &= y + \Delta y \\z &= z + \Delta z \\ \alpha &= \alpha + \Delta \alpha \\ \beta &= \beta + \Delta \beta \\ \vartheta &= \vartheta + \Delta \vartheta\end{aligned}\tag{1.5}$$

Optimizacija yra baigiama, kai kita iteracija apskaičiuotos reikšmės labai mažai skiriasi nuo ankstesnės iteracijos rezultatų. Optimizacija yra teisinga jei yra rastas globalus minimumas. Optimizacija yra klaidinga jei yra randamas vienas iš lokaliųjų minimumų[20].

2. GALVOS POZICIJOS NUSTATYMO SISTEMA

Pasirinkta galvos pozicijos nustatymo sistema, kai kamera yra nukreipta į vaizduoklį (žr. 2.1 pav.). Kamera pritvirtinta vartotojui ant galvos ir nukreipta žiūrėjimo kryptimi. Prie kameros yra pritvirtintas IR filtras, kuris filtruoja vaizdą. Kameros pagalba yra fiksuojami tik IR spinduliai. Filtras naudojamas siekiant sumažinti informacijos kiekį iš kameros. Vaizduose yra paliekama tik reikalinga informacija ar nereikšmingi trukdžiai. Vaizduoklio kampuose įmontuoti IR diodai. Stebint diodų padėtį, analizuojami sudaryto keturkampio pokyčiai. Pagal tyrimų nustatytas priklausomybes yra apskaičiuojami galvos padėties parametrai vaizduoklio atžvilgiu.



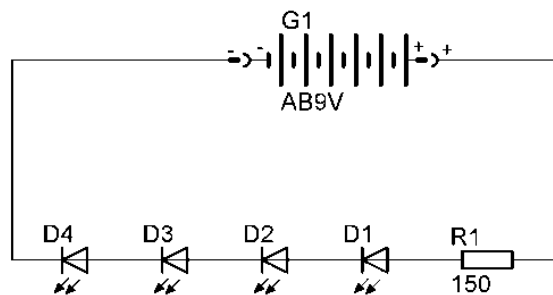
2.1 pav. Galvos padėties vaizduoklio atžvilgiu nustatymo sistemos schema

Fiksuotas vaizdas apdorojamas vaidų apdorojimo algoritmu. Yra vykdomas papildomas filtravimas, aptinkami IR spindulių telkiniai, nereikšmingų taškų išvalymas, keturkampio kampų koordinatėms išskyrimas, koordinatėms tikslinimas bei galvos pozicijos parametrų apskaičiavimas. Įvykdžius algoritmą, išvedamas vaizdas į ekraną, taip pat išvedamos keturkampio kampų koordinatės, galvos padėties parametrai, algoritmo vykdymo laikas. Sistema yra realizuota Matlab, C# ir C++ aplinkose.

2.1. Techninė įranga

1. USB kamera – fiksuoja vaizdą realiu laiku. Naudojamos kameros raiška yra 640*480. Informacija perduodama USB sąsaja į kompiuterį.
2. IR filtras – optinis filtras, praleidžiantis tik IR spindulius. Naudojamas filtras yra 25 mm skersmens. Jo dydis parinktas pagal kameros parametrus. Šio filtro laidumas – 1000 nm. Yra nufiltruojamas visas žmogaus regimos šviesos diapazonas. Į vaizdą praleidžiami tik IR spinduliai.

3. Kompiuteris: naudojamas nešiojamas kompiuteris Asus K50I. Jo techniniai parametrai:
- procesorius: Intel Pentium dual core T4500 – 2,3 GHz;
 - operatyvioji atmintis (RAM): DDR3 4,00 GB;
 - kietasis diskas: 500 GB;
 - vaizdo plokštė: Nvidia Geforce GT 320M – 1 GB.
4. Vaizduoklio maketas, kurio kampuose įmontuoti IR diodai. 4 diodai yra sujungti nuosekliai būdu (žr. 2.2 pav.). Jie maitinami 9 voltų maitinimo šaltiniu – krona.



2.2 pav. 4 diodų jungimo schema

5. IR diodai: vaizduoklyje yra naudojami L-53F3C IR diodai.



2.3 pav. Techninė įranga

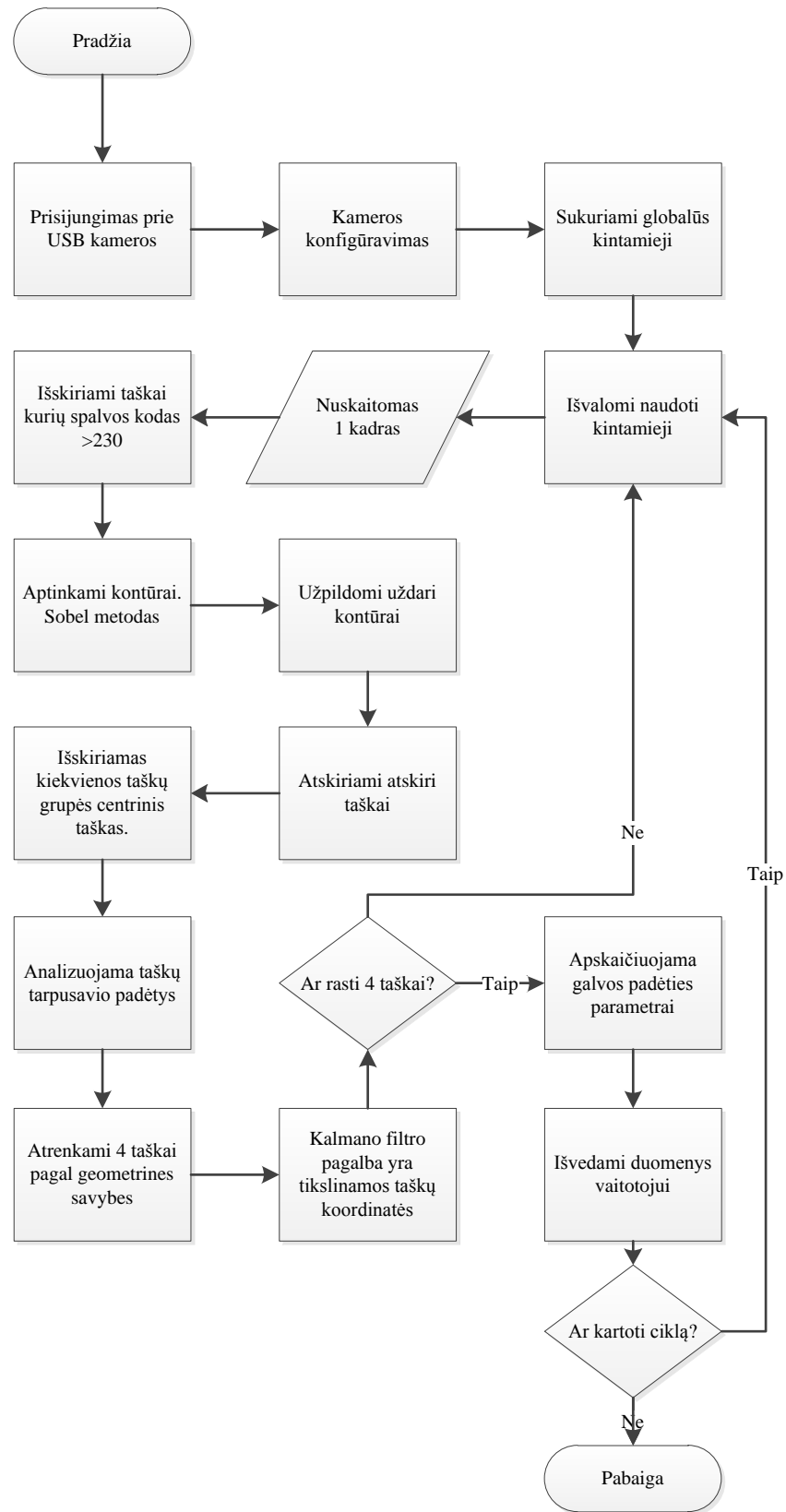
2.2. Programinė įranga

Programinė įranga buvo kurta naudojant Matlab, Microsoft Visual C# 2010 ir Microsoft Visual C++ 2003 aplinkas. Vienas iš tyrimų buvo ištirti, kurioje aplinkoje kurta programa apdoros vaizdo medžiagą ir vykdys algoritmą sparčiausiai.

Matlab aplinka plačiai naudojama įvairiems tyrimams, skaičiavimams bei vaizdo apdorojimui. Ši programavimo aplinka yra greita ir paprasta atliekant nedidelius skaičiavimus. Programa veikia tiesiogiai iš kodo, prieš tai jo nekompiliuodama. Kiekvieno programos vykdymo metu yra saugomi visi globalūs kintamieji, naudoti programoje, ir po jos įvykdymo galime atlikti atskirus skaičiavimus su jais. Programos, scenarijai gali būti kuriami kaip konsolinės ar vartotojo sąsają turinčios programos. Vaizdams apdoroti pasitelkta *Image Processing Toolbox* biblioteka. Naudojama Matlab versija R2010b [21].

C# aplinkta plačiai naudojama kuriant .NET struktūra veikiančias programas. Tai yra viena iš C kalbų šeimos narių. Vaizdų apdorojimui naudojama *Emgu.CV* biblioteka. *Emgu.CV* – tai biblioteka, leidžianti naudotis *Open.CV* funkcijomis .NET struktūros programose. Programos, scenarijai gali būti kuriami kaip konsolinės ar vartotojo sąsają turinčios programos. Naudojama Visual C# 2010 programa [22,23].

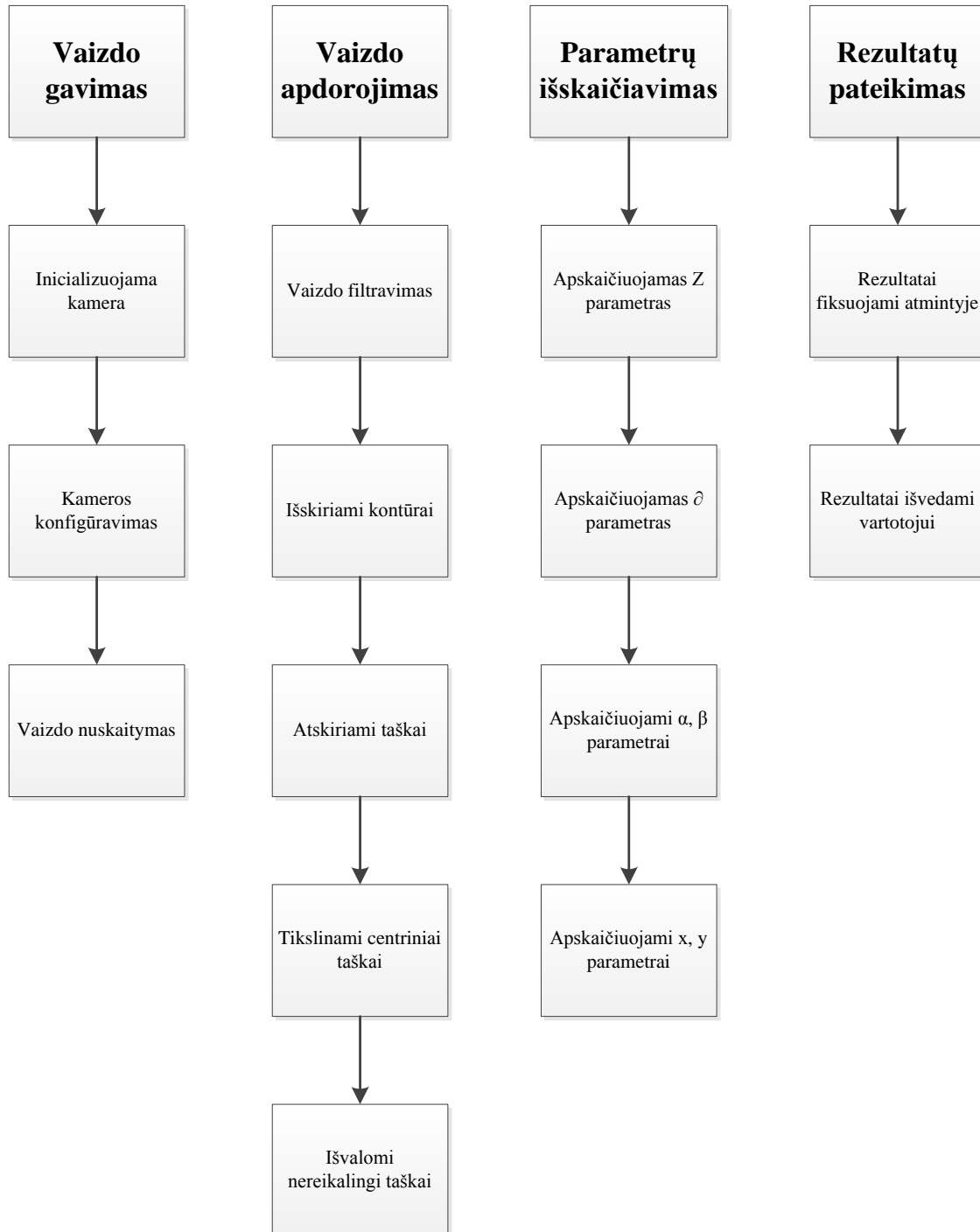
C++ aplinka plačiai naudojama realizuojant įvairius algoritmus, kuriant programas. Tai viena iš C kalbų šeimos narių. Vaizdų apdorojimui yra naudojama *Open.CV* biblioteka. *Open.CV* biblioteka yra pritaikyta darbui su vaizdais, filmuota medžiaga. Programos, scenarijai gali būti kuriami kaip konsolinės ar vartotojo sąsają turinčios programos. Naudojama Visual C++ 2003 programa [24,25].



2.4 pav. Programos algoritmo blokinė schema

Vaizdų apdorojimo algoritmas (žr. 2.4 pav.) fiksuoja vaizdą. Vaizde surandami IR šviesos šaltiniai. Yra atskiriami 4 taškai, kurie atitinka vaizduoklio kampuose šviečiančius diodus. Pagal gautas priklausomybes yra nustatomi galvos padėties parametrai. Duomenys išvedami į ekraną.

Realizuojamas algoritmas yra suskirstytas į 4 blokus (žr. 2.5 pav.):



2.5 pav. Algoritmo suskirstymas į 4 blokus

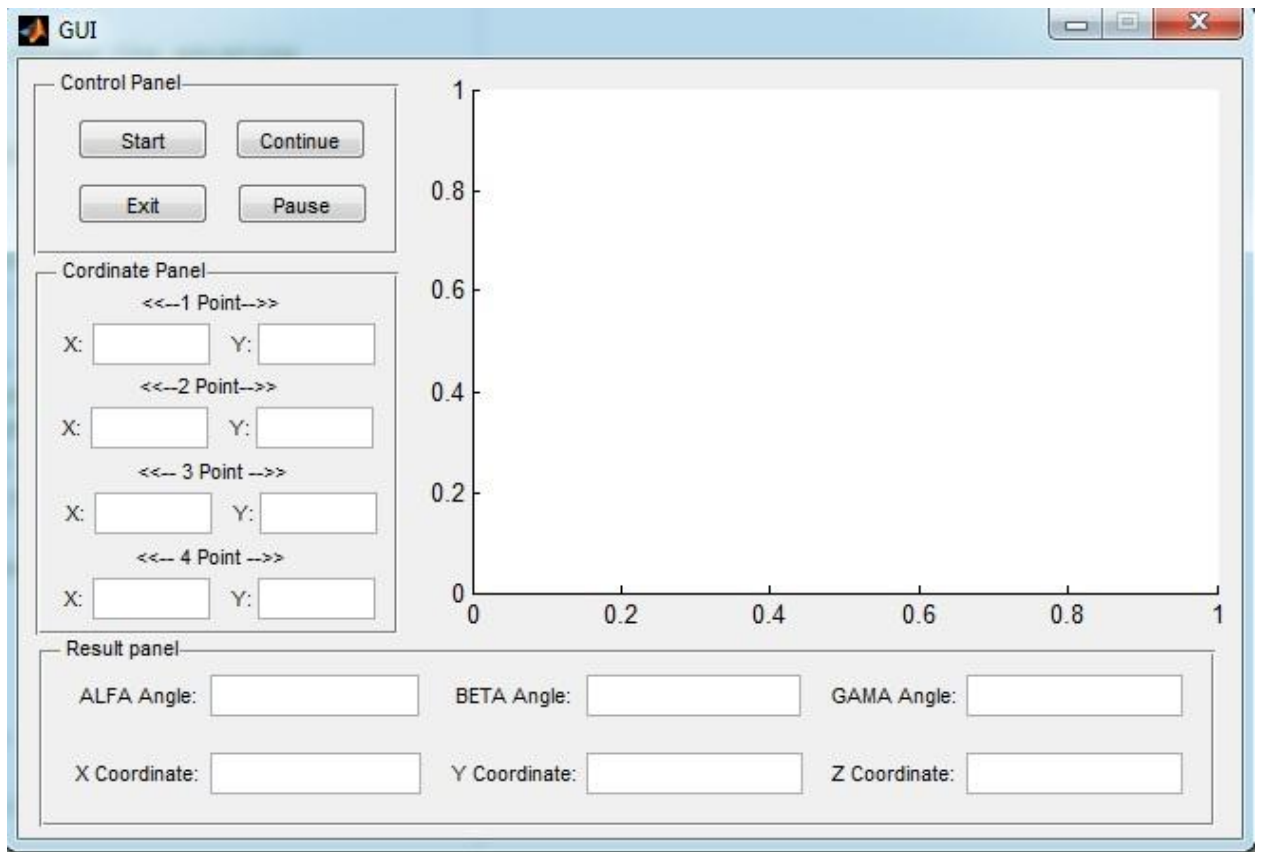
Matlab aplinkoje kurtos programos grafinė vartotojo sąsaja (žr. 2.6 pav.) susideda iš trijų dalių:

- valdymo mygtukai;
- video išvestis;
- duomenų išvestis.

Valdymo mygtukais galima pradėti darbą paspaudus *Start*. Mygtuku *Pause* galima sustabdyti algoritmo vykdymą. Mygtuku *Continue* pratęsimas sustabdytas darbas. Programos darbas baigiamas paspaudus mygtuką *Exit*.

Video išvesties lauke yra išvedamas algoritmo apdorotas vaizdas.

Duomenų išvesties laukas padalintas į dvi dalis: Cordinate panel ir Result panel. Cordinate panel yra išvedamos x ir y koordinatės, o Result panel – apskaičiuoti galvos padėties parametrai.



2.6 pav. Matlab aplinkoje kurtos programos grafinė vartotojo sąsaja

Visual C# aplinkoje kurtos programos grafinė vartotojo sąsaja (žr. 2.7 pav.) susideda iš trijų dalių:

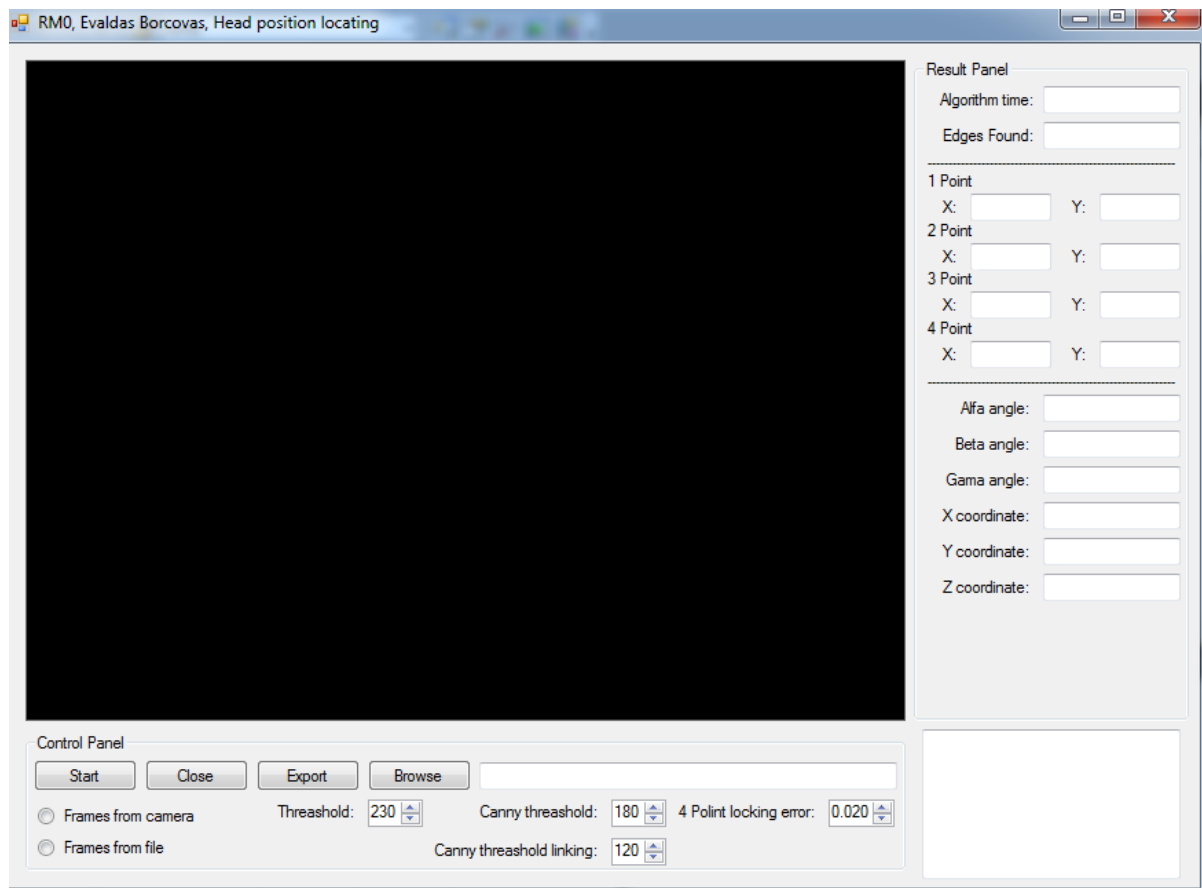
- valdymo mygtukai;
- video išvestis;

- duomenų išvestis.

Valdymo mygtukais galima pradėti darbą paspaudus *Start*. Pradėjus programos vykdymą, vietoje mygtuko *Start* atsiranda naujas mygtukas *Pause*, kurio pagalba galima sustabdyti algoritmo vykdymą. Tai atlikus, vietoje mygtuko *Pause* atsiranda mygtukas *Continue*, kurio pagalba pratęsimas sustabdytas darbas. Paspaudus *Export*, išsaugomi 100 paskutinių skaičiavimo rezultatų. *Browse* iškviečia aplankalų paieškos langą. Pasirinkus reikiamą aplankalą jis fiksuojama lauke greta jo. Pasirinkus *Frames from camera* duomenys imami iš kameros, o *Frames from file* duomenys imami iš nurodyto aplankalo iš .jpg failų. Programos darbas baigiamas paspaudus mygtuką *Close*. *Threshold* nustatoma filtravimo riba. *Canny threshold* ir *Canny threshold linking* – kontūrų radimo funkcijos parametrai. *4 Point locking error* – tai maksimali paklaida, tikrinant keturkampio geometrinės savybes.

Video išvesties lauke išvedamas algoritmo apdorotas vaizdas.

Duomenų išvesties lauke *Result panel* yra išvedami visi duomenys: algoritmo vykdymo laikas, kontūrų skaičius vaizde, taškų x ir y koordinatės, parametrai galvos padėčiai nustatyti, papildoma informacija, jei to reikia.



2.7 pav. Visual C# aplinkoje kurtos programos grafinė vartotojo sąsaja

Visual C++ aplinkoje kurta programa yra konsolinė. Ji neturi grafinės vartotojo sąsajos. Testavimo metu programa skaito vaizdus iš .avi failo ir vykdo algoritmą. Algoritmo vykdymo rezultatai yra išvedami tik pabaigus darbą. Jie išvedami į atskirus failus:

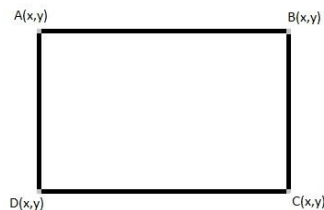
- trukmes.txt – segmentų vykdymo laikas.
- tikslinimas.txt – patikslintos reikšmės po kiekvienos iteracijos.
- tikslinimas_delta.txt – reikšmių pokytis po kiekvienos iteracijos.
- tikslinimas_rezultatai.txt – optimizuotos ieškomų parametrų reikšmės.
- koordinates.txt – aptiktų IR telkinių centrinių taškų koordinatės.

3. TYRIMAI

3.1. Keturkampio geometrinių savybių pokytis, kintant stebėjimo kampui

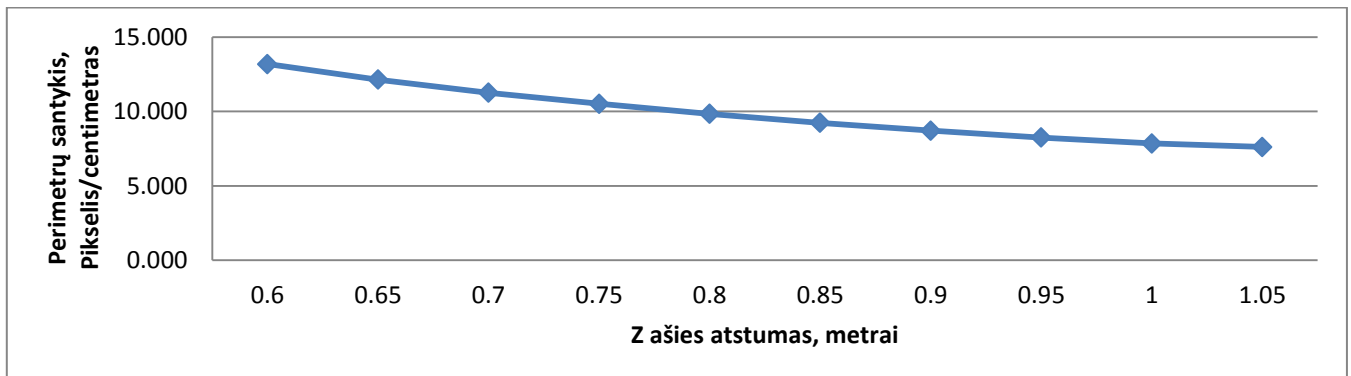
Tyrimo tikslas: nustatyti, kokie keturkampio parametrai kinta keičiant stebėjimo poziciją; kaip jie kinta; taip pat nustatyti, kokie keturkampio geometriniai parametrai suteikia daugiausiai informacijos apie galvos pozicijos ir orientacijos padėtį.

Tyrimo eiga: sugeneruojamas virtualus keturkampis (žr. 3.1 pav.). Keičiant kameros stebėjimo padėtį, yra fiksuojami vaizdai. Kiekviename vaizde stebimi ir analizuojami keturkampio geometriniai parametrai: kraštinių ilgi, kraštinių santykiai, kampai tarp kraštinių. Tyrimas atliekamas keičiant visus parametrus paeiliui, t. y. analizuojamas keturkampis atskirai kiekvienam parametru. Parenkami parametrai, suteikiantys daugiausiai informacijos apie stebėjimo padėtį.



3.1 pav. Keturkampio žymėjimas

Pirmasis tiriamas parametras yra z ašies atstumas. Visi matavimo rezultatai pateikti lentelėje (žr. 1 priedas). Stebint keturkampio geometrines savybes (žr. 2 priedas) ir keičiant z ašies atstumą, pastebėta, jog visi kraštinių ilgi kinta tolygiai.

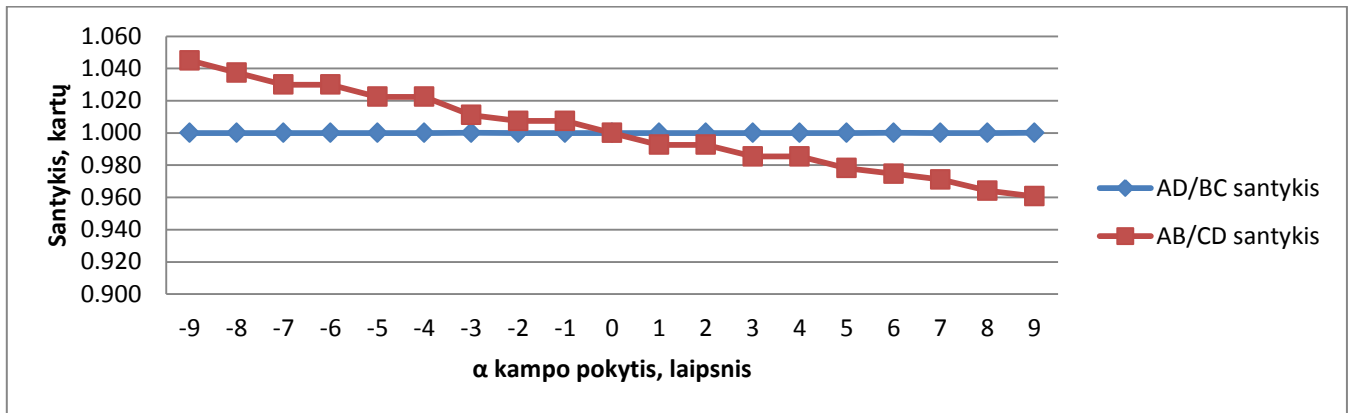


3.2 pav. Perimetrų santykis keičiant z atstumą

Daugiausia informacijos apie keičiamą atstumą perteikia perimetrų santykis, kuris mažėja didinant z atstumą, ir, atvirkščiai, – didėja mažinant atstumą (žr. 3.2 pav.).

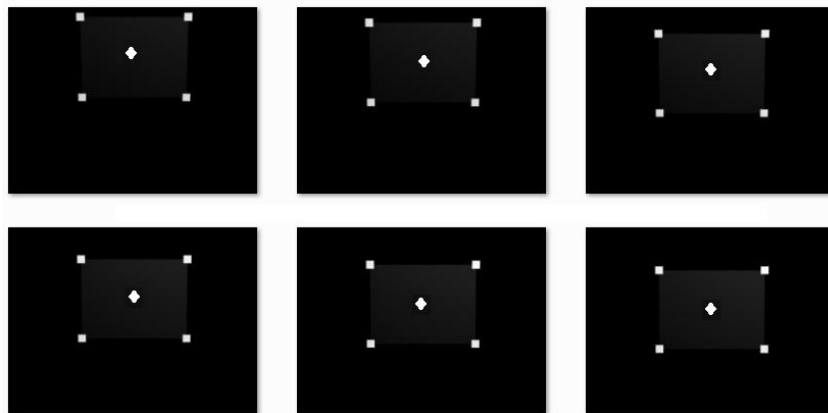
Antrasis parametras yra α kampas. Vaizdai fiksuojami virtualioje aplinkoje. Atstumas nuo kameros iki stebimo objekto – 1 metras. Keičiamas parametras yra α kampas. Visi matavimo rezultatai pateikti lentelėje (žr. 3 priede).

Išanalizavus keturkampio geometrinius parametrus (žr. 4 priede), pastebėta, jog ne visų kraštinių ilgai kinta tolygiai. Horizontalios kraštinės ilgėja ar trumpėja priklausomai nuo kampo pokyčio. Jei α kampas daugiau už 0, tuomet AB kraštinė yra trumpesnė už CD, o jei kampas yra mažiau už 0, tai CD trumpesnė už AB. Pasirinktas parametras, teikiantis didžiausią kiekį informacijos apie α kampo pokytį, – AB ir CD kraštinių santykis (žr. 3.3 pav.).



3.3 pav. Keturkampio kraštinių santykiai keičiant α kampą

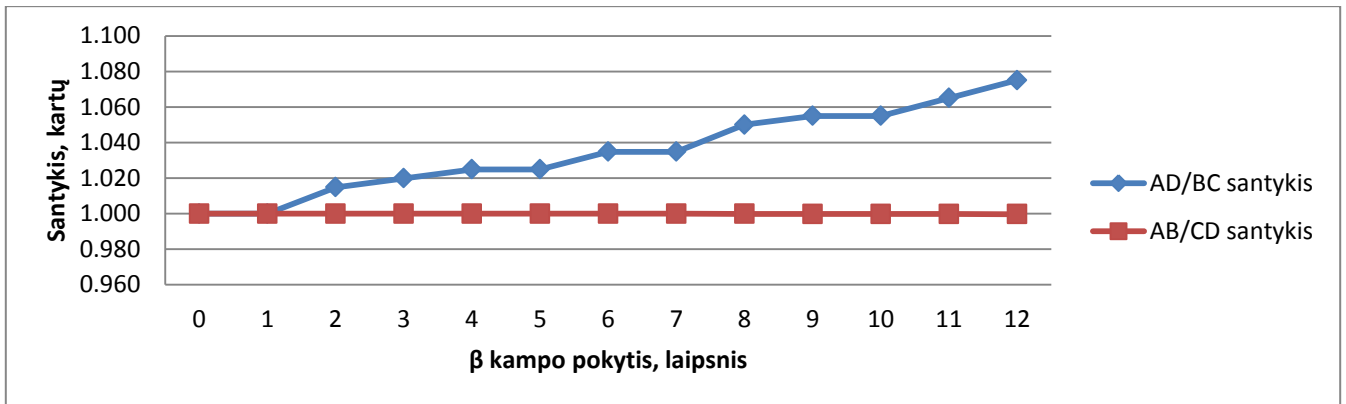
Pastebėta, jog kintant kampui α , kinta ir keturkampio centro padėtis (žr. 3.4 pav.). Centro padėties matavimas yra numatytas x ir y koordinačių matavimui. Todėl reikalinga korekcija, kurios pagalba bus siekiama išvengti nuokrypio.



3.4 pav. Keturkampio centro padėties kitimas priklausomai nuo α kampo

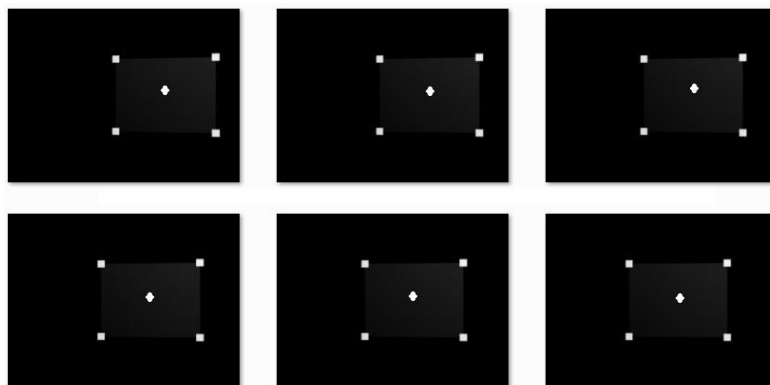
Trečiasis parametras yra β kampas. Vaizdai fiksuojami virtualioje aplinkoje. Atstumas nuo kameros iki stebimo objekto yra 1 metras. Keičiamas parametras – β kampas. Visi matavimo rezultatai pateikti lentelėje (žr. 5 priede).

Išanalizavus keturkampio geometrinius parametrus (žr. 6 priede), pastebėta, jog ne visų kraštinių ilgių kinta tolygiai. Vertikalios kraštinės ilgėja arba trumpėja priklausomai nuo kampo pokyčio. Jei β kampas didėja, tuomet BC kraštinė trumpesnė už AD, o jei kampas mažėja – BC yra ilgėja už AD. Pasirinktas parametras, teikiantis didžiausią kiekį informacijos apie β kampo pokytį, – BC ir AD kraštinių santykis (žr. 3.5 pav.).



3.5 pav. Keturkampio kraštinių santykiai keičiant β kampą

Pastebėta, jog kintant kampui α , kinta ir keturkampio centro padėtis (žr. 3.6 pav.). Centro padėties matavimas yra numatytas x ir y koordinatų matavimui. Todėl reikalinga korekcija, kurios pagalba bus siekiama išvengi nuokrypio.



3.6 pav. Keturkampio centro padėties kitimas priklausomai nuo β kampo

Ketvirtasis parametras yra δ kampas. Vaizdai fiksuojami virtualioje aplinkoje. Atstumas nuo kameros iki stebimo objekto yra 1 metras. Keičiamas parametras – δ kampas. Visi gauti duomenys pateikti lentelėje (žr. 7 priede).

Išanalizavus keturkampio geometrinius parametrus (žr. 8 priede), pastebėta, kad kraštinių ilgiai kinta nereikšmingai. Stebimi keturkampio kampų pokyčiai taip pat nereikšmingi. Pastebėta, jog matuojant δ kampą kiti parametrai įtakos neturi. Kampo pokytis taip pat neįtakojo keturkampio centro koordinatė. δ kampas apskaičiuojamas pagal formulę (3.1).

$$GAMA = Atan \frac{(D_y - C_y)}{(C_x - D_x)} * \left(\frac{180}{3.14} \right); \quad (3.1)$$

čia $GAMA$ – δ kampas;
 C_x – C taško x koordinatė;
 C_y – C taško y koordinatė;
 D_x – D taško x koordinatė;
 D_y – D taško y koordinatė.

Kintant stebėjimo padėčiai vertikaliai ar horizontaliai, keturkampio parametrai nekinta, tačiau keičiasi keturkampio padėtis vaizde. Įvertinus keturkampio postūmį vaizde, galime išmatuoti, koks yra stebėjimo taško postūmis, x ir y kryptimis. Siekiant rezultatą pervesti į metrus, yra naudojamas pikselio ir centimetro santykis, kuris matuojamas nustatant atstumą z ašies atžvilgiu. Gautiems rezultatams taikomos x ir y korekcijos, gautos iš α ir β matavimų.

Išskirti parametrai, kurie suteikia daugiausiai informacijos apie galvos padėtį vaizduoklio atžvilgiu. Gauti duomenys kintant galvos pozicijos ir orientacijos parametrus. Pritaikius tiesinę aproksimaciją, gautos parametru apskaičiavimo formulės:

- z atstumas (3.2);
- α kampas (3.3);
- β kampas (3.4);
- δ kampas (3.5);
- x korekcija (3.6);
- y korekcija (3.7).

$$Z = \frac{(S-20,0812)}{-12,4205}; \quad (3.2)$$

čia Z – atstumas iki keturkampio metrais;
 S – perimetrų santykis (pikselis/centimetras).

$$Alfa = \frac{AB/CD-1,0009}{-0,0045}; \quad (3.3)$$

čia Alfa – α kampas laipsniais;
 AB – AB kraštinės ilgis pikseliais;
 CD – CD kraštinės ilgis pikseliais.

$$Beta = \frac{AD/BC-0,9964}{0,0065}; \quad (3.4)$$

čia Beta – β kampas laipsniais;
 AD – AD kraštinės ilgis;
 BC – BC kraštinės ilgis.

$$X = 1,6777 * BETA - 0,7307; \quad (3.5)$$

čia BETA – β kampas laipsniais;
 X – x koordinatės korekcija centimetrais.

$$Y = -1.7092 * ALFA - 0.9126; \quad (3.6)$$

čia ALFA – α kampas laipsniais;
 Y – y koordinatės korekcija centimetrais.

Gautos lygtys apskaičiuoja x , y , z koordinates bei α , β , ϑ kampus. Atliktų bandymų metu tikslumas yra:

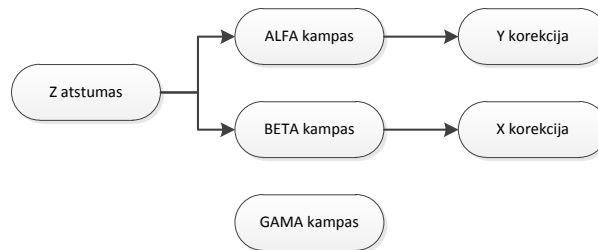
- galvos pozicijos parametrams – 2 centimetrai;
- galvos orientacijos parametrams – 2 laipsniai.

Toks metodo tikslumas yra tik atliekant matavimus iš 1 metro atstumo. Galime daryti išvadą, jog z ašies atstumas turi tiesioginę įtaką kitų parametų skaičiavimui. Siekiant, jog sistema veiktų nepriklausomai nuo z ašies atstumo, reikia atlikti papildomus tyrimus ir nustatyti parametų tarpusavio priklausomybes.

3.2. Keturkampio parametrų tarpusavio priklausomybių nustatymas

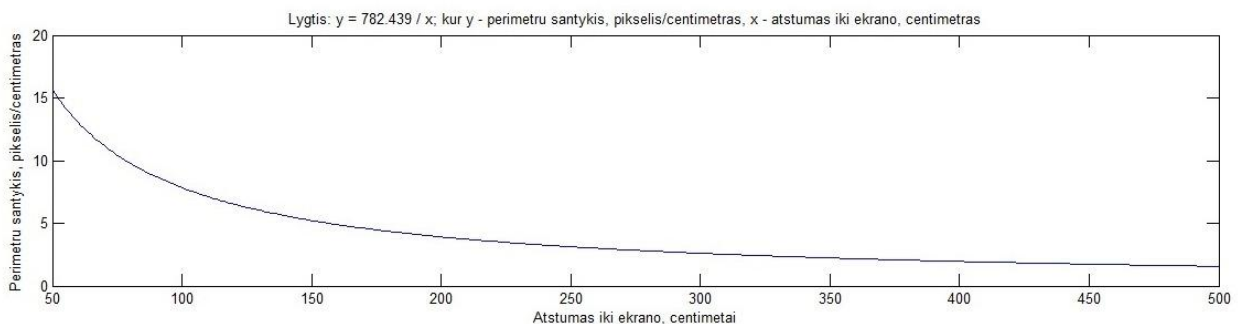
Tyrimo tikslas: nustatyti pasirinktų keturkampio geometrinių parametrų tarpusavio sąryšius ir pagal juos išvesti tikslesnes formules galvos pozicijos ir orientacijos parametrų skaičiavimui.

Tyrimo eiga: sugeneruojamas virtualus keturkampis (žr. 3.1 pav.). Stebimi pasirinktų parametrų pokyčiai keičiant parametrus. Siekiant išvesti tikslesnes formules, keičiamas atstumas iki ekrano, ir kiekvienos iteracijos metu yra keičiami orientacijos parametrai. Gaunama z ašies atstumo priklausomybė skaičiuojant α ir β kampus. 3. 1. skyriuje pastebėtoms x ir y korekcijoms taip pat yra apskaičiuojamos tikslesnės formulės keičiant z ašies atstumą. Pasirenkama parametrų skaičiavimo hierarchija (žr. 3.7 pav.).

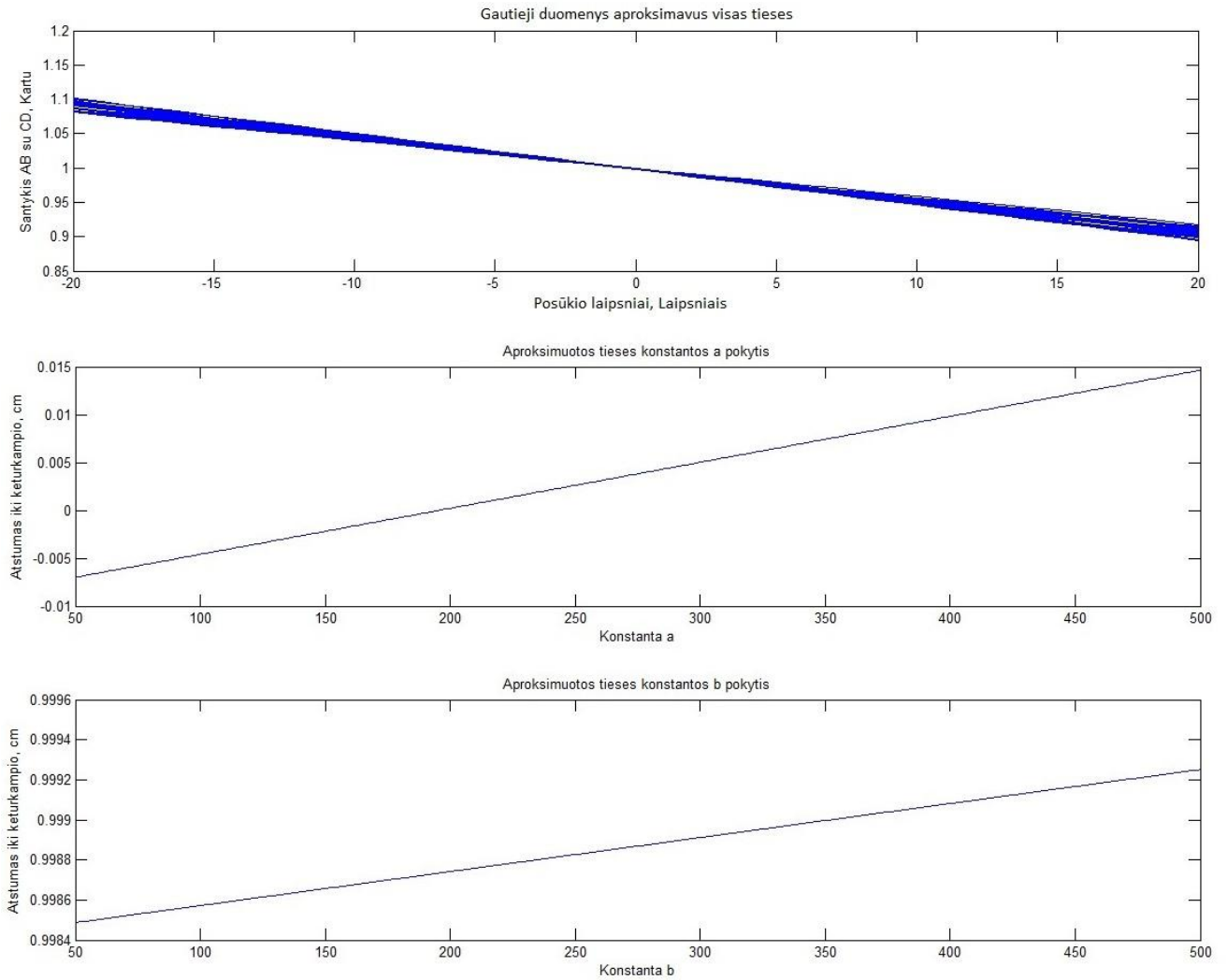


3.7 pav. Galvos padėties ekrano atžvilgiu parametrų skaičiavimo hierarchija

Analizuojant gautus duomenis, pastebėta, jog x (žr. 3.11 pav.) ir y (žr. 3.12 pav.) korekcija yra tiesiškai priklausoma nuo α ir β kampų, o α (žr. 3.9 pav.) ir β (žr. 3.10 pav.) kampai priklausomi nuo z atstumo. Pagal pastebėtus tarpusavio sąryšius yra atliekami tyrimai. Tikslinama z parametro apskaičiavimo formulė tiriant daugiau vaizdų. Iteracijos žingsnis yra 1 cm, minimali reikšmė – 50 cm, maksimali reikšmė – 500 cm (žr. 3.8 pav.).

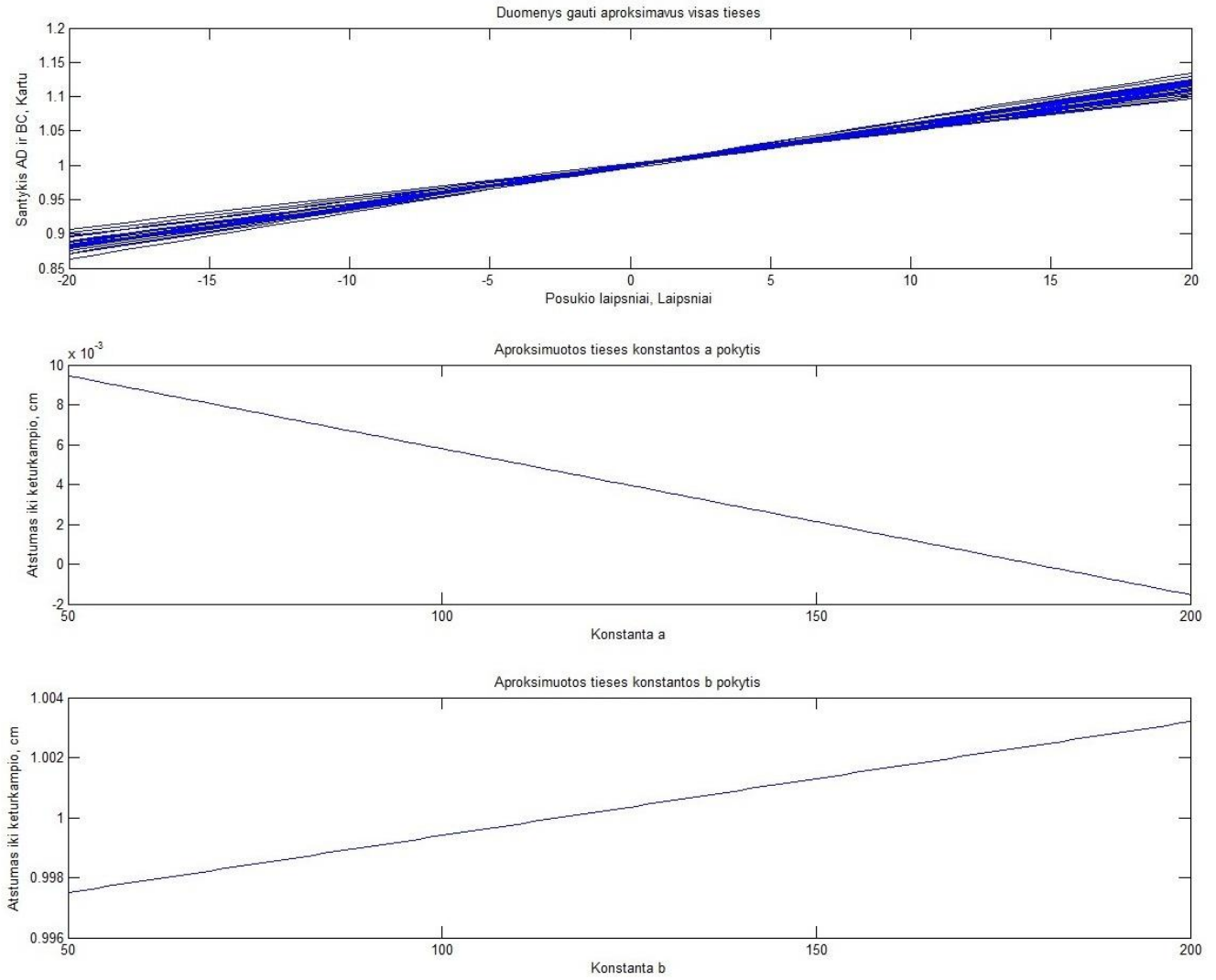


3.8 pav. z parametro pokytis nuo 50 cm iki 500 cm ir priklausomybę atitinkanti lygtis



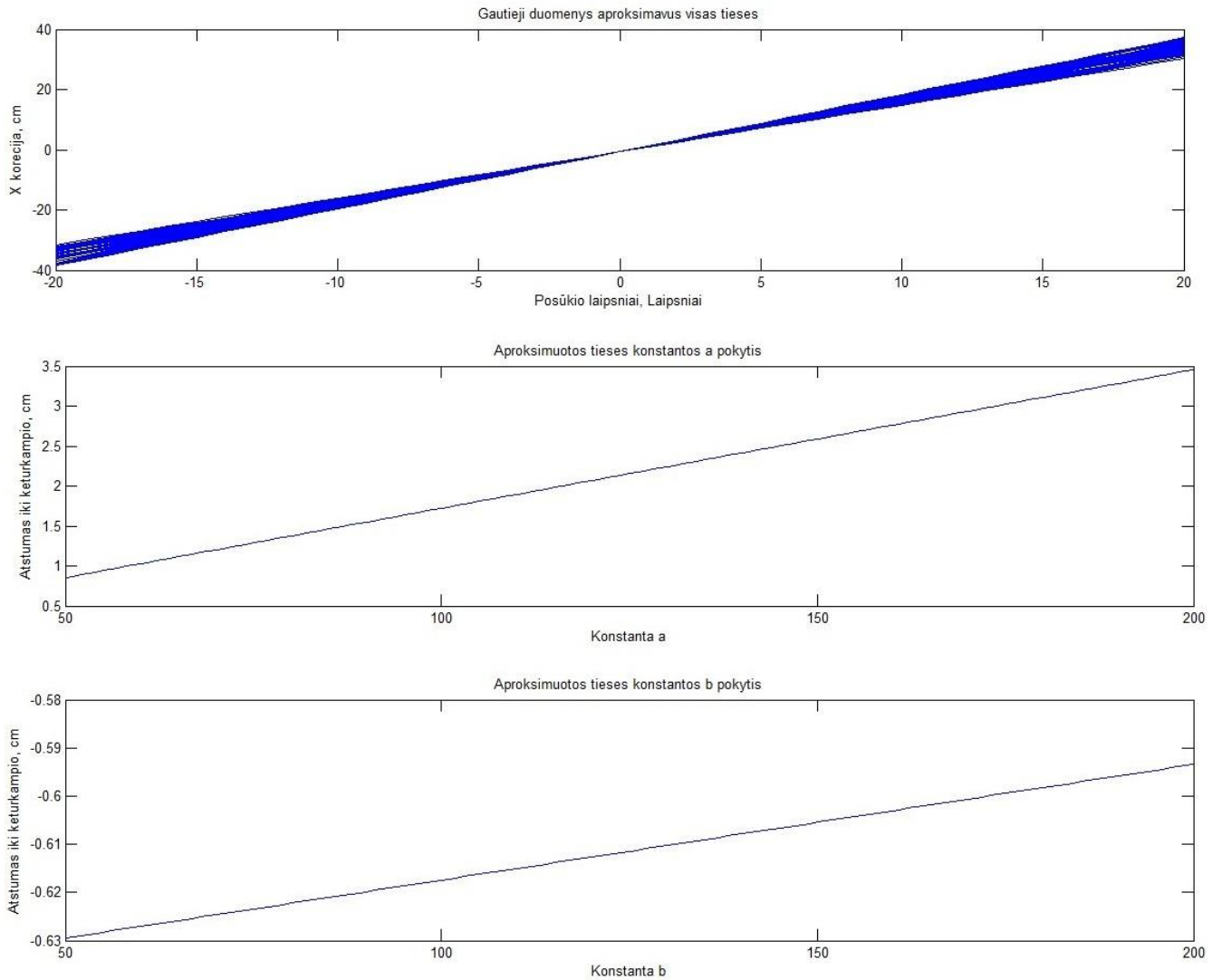
3.9 pav. α parametro pokyčiai lygiagrečiai keičiant z atstumą, aproksimuotų lygčių grafikai

Kiekvienai gautai lygčiai atlikta tiesinė regresija, sudaromas formulių rinkinys. Formulių forma: $y = ax + b$. Gautųjų lygčių a ir b įverčiamas yra atliekama tiesinė regresija (žr. 3.9 pav.) ir išvedama universali formulė, kuri apskaičiuoja α parametą nepriklausomai nuo atstumo.



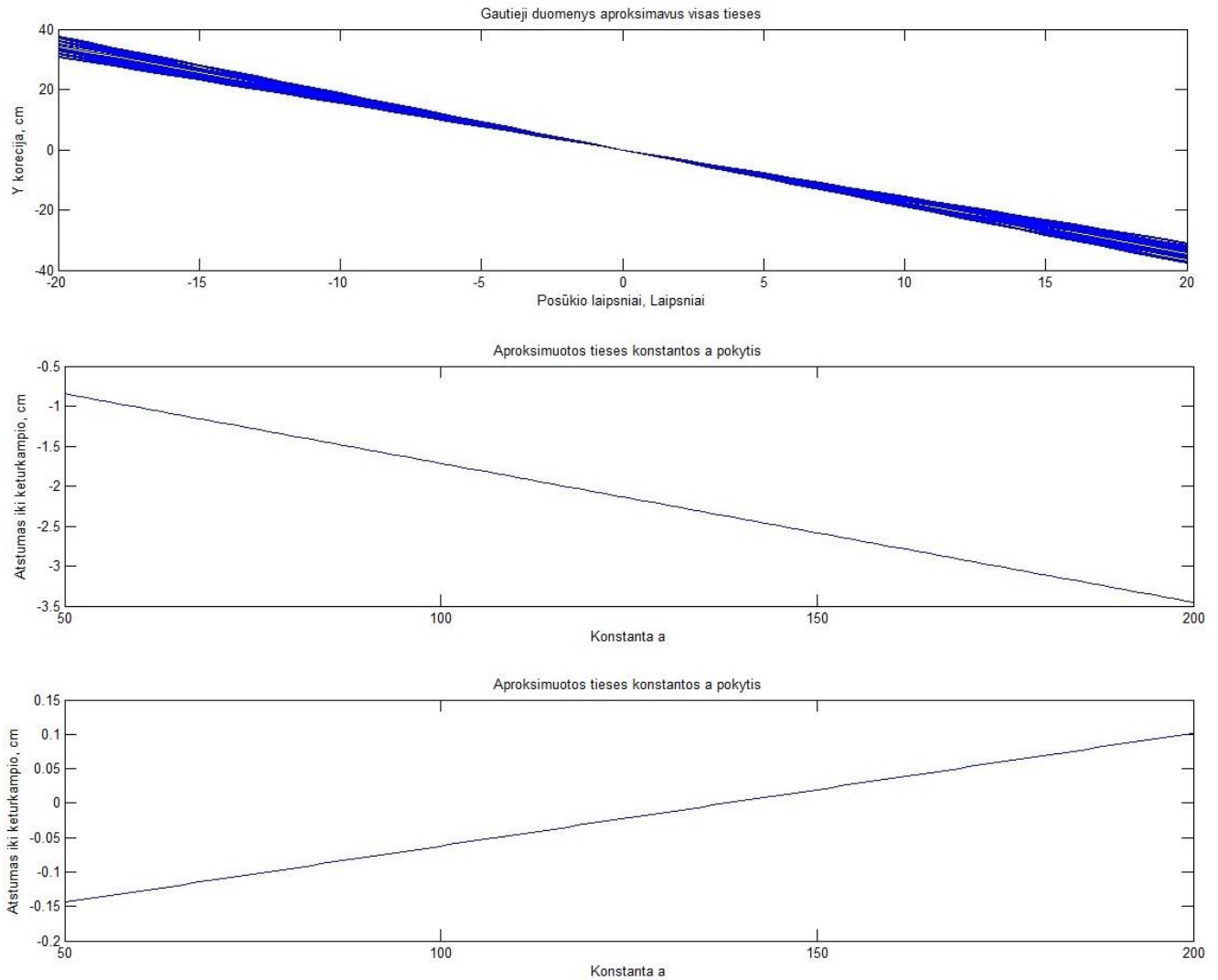
3.10 pav. β parametro pokyčiai lygiagrečiai keičiant z atstumą, aproksimuotų lygčių grafikai

Kiekvienai gautai lygčiai atlikta tiesinė regresija, yra sudaromas formulių rinkinys. Formulių forma: $y = ax + b$. Gautųjų lygčių a ir b įverčiamas yra atliekama tiesinė regresija (žr. 3.10 pav.) ir išvedama universalinė formulė, kuri apskaičiuoja β parametą nepriklausomai nuo atstumo.



3.11 pav. x korekcijos pokyčiai lygiagrečiai keičiant z atstumą, aproksimuotų lygčių grafikai

Kiekvienai gautai lygčiai atlikta tiesinė regresija, yra sudaromas formulų rinkinys. Formulų forma: $y = ax + b$. Gautųjų lygčių a ir b įverčiamas yra atliekama tiesinė regresija (žr. 3.11 pav.) ir išvedama universalinė formulė, kuri apskaičiuoja x korekciją nepriklausomai nuo atstumo.



3.12 pav. y korekcijos pokyčiai lygiagrečiai keičiant z atstumą, aproksimuotų lygčių grafikai

Kiekvienai gautai lygčiai atlikta tiesinė regresija, yra sudaromas formulių rinkinys. Formulių forma: $y = ax + b$. Gautųjų lygčių a ir b įverčiamas yra atliekama tiesinė regresija (žr. 3.12 pav.) ir išvedama universali formulė, kuri apskaičiuoja y korekciją nepriklausomai nuo atstumo.

Išanalizavus gautuosius duomenis, yra išvestos lygtys, apskaičiuojančios galvos pozicijos ir orientacijos parametrus nepriklausomai nuo atstumo. Apskaičiuojami parametrai:

- z atstumas (3.8);
- α kampas (3.9);
- β kampas (3.10);
- x korekcija (3.11);
- y korekcija (3.12).

$$Z = \frac{782.439}{P_{pixel}/P_{cm}}; \quad (3.8)$$

čia Z – z ašies atstumas;

P_{pixel} – keturkampio perimetras pikseliais;

P_{cm} – keturkampio perimetras centimetrais.

$$Alfa = \frac{Alfa_{param} + (0.000016 * Z - 0.9984)}{0.000048 * Z + 0.0094}; \quad (3.9)$$

čia Alfa – α kampas;

$Alfa_{param}$ – AB ir CD santykis.

$$Beta = \frac{Beta_{param} + (0.000037 * Z - 0.9956)}{-0.0001 * Z + 0.0131}; \quad (3.10)$$

čia Beta – β kampas;

$Beta_{param}$ – DA ir BC kraštinių santykis.

$$X = \frac{Beta + (0.0002 * Z - 0.6414)}{0.0173 * Z - 0.011}; \quad (3.11)$$

čia X – x korekcija.

$$Y = \frac{Alfa + (0.0016 * Z - 0.2261)}{-0.0174 * Z + 0.0231}; \quad (3.12)$$

čia Y – y korekcija.

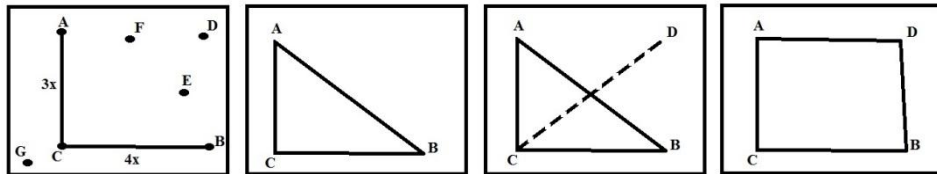
3.3. Taškų filtravimas ir rūšiavimas taikant geometrines taisykles

Tyrimo tikslas: sukurti metodiką taškų filtravimui.

Tyrimo eiga: programos vykdymo metu fiksuojami IR spindulių telkiniai. Jei yra daugiau nei 4 telkiniai, reikia atlikti taškų filtravimą ir nereikalingų taškų panaikinimą. Taškų filtravimui taikomos trikampio geometrinės savybės. Išrenkamos savybės, kurių pagalba filtravimas yra optimalus. Aptikus 3 reikiamus taškus, pagal trikampio įstrižainę nustatomas 4 keturkampio taškas. Taškai paruošiami darbui atsižvelgiant į padėtį vaizde.

Vaizde ieškoma taškų. Aptikus 4 taškus, yra tikrinama, ar 3 taškai sudaro trikampį. Žinant vaizduoklio kraštinių santykį, tikrinama sąlyga, ar trikampio kraštinių santykis yra toks pat – 3:4. Santykis parinktas pagal kraštinių ilgių duomenis iš 3. 1. tyrimo. Atmetami taškai D, E, F, G. Aptiktas

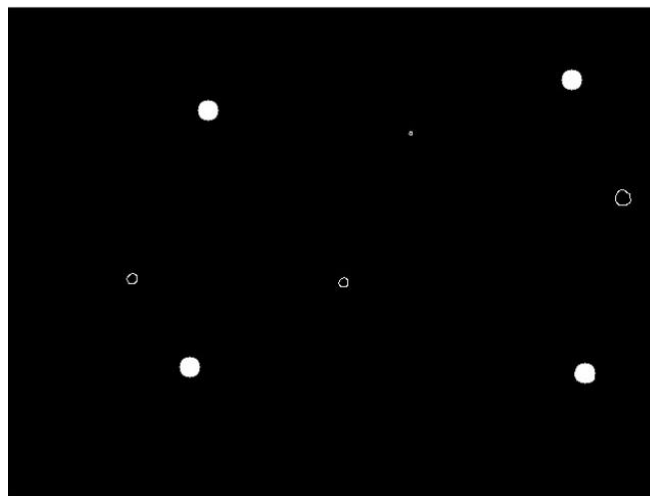
trikampis ABC. Aptikus atkarpas, kurios tenkina pirmąją sąlygą, ieškoma ilgiausia trikampo kraštinė AB. Rastoji kraštinė yra ieškomo keturkampio įstrižainė. Ieškomas ketvirtas keturkampio taškas. Ieškomas taškas, kurio atstumas nuo taško C yra artimiausias įstrižainės AB atstumui. Aptikus 4 tašką, tikrinama sąlyga, ar keturkampis yra tinkamas. Dėl to tikrinama, ar visi keturkampio kampai neviršija 100° . 100° parinktas išanalizavus duomenis, gautus 3. 1. poskyryje. Jei sąlyga tenkinama, tuomet keturkampis yra tinkamas, ir galima tęsti skaičiavimus (žr. 3.13 pav.). Jei sąlyga netenkinama, ieškomi kiti taškai, sudarantys trikampį. Jei tinkami taškai nėra rasti, pereinama prie sekančio kadro.



3.13 pav. Taškų filtravimas, keturkampio aptikimas

Aptiktus taškus reikia juos surūšiuoti pagal padėtį vaizde. Taškų rūšiavimas vyksta tokiu principu:

- 1 taško x ir y koordinatė yra mažiausia;
- 4 taško x ir y koordinatė yra didžiausia;
- 2 taško x koordinatė yra artimiausia 4 taško x koordinatei;
- 3 taško x koordinatė yra artimiausia 1 taško x koordinatei.



3.14 pav. Realus vaizdas atlikus filtravimą

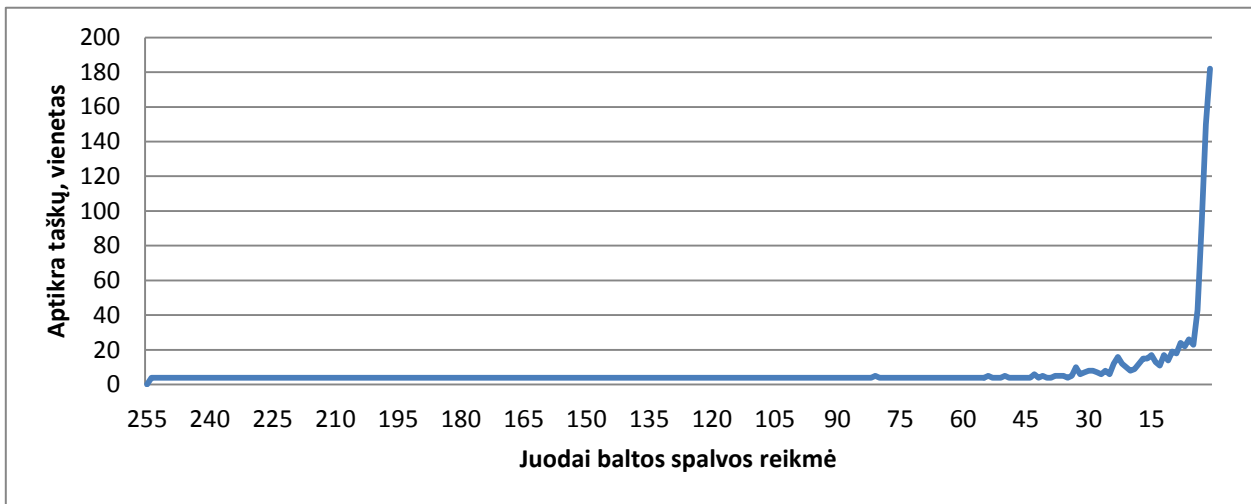
3.4. Algoritmo parametrų optimalus parinkimas

Tyrimo tikslas: įvertinti algoritmo parametrų įtaką vaizdo filtravimui, nereikalingų taškų atmetimui, teisingų taškų pasirinkimui. Išrinkti optimalius parametrus algoritmo vykdymui.

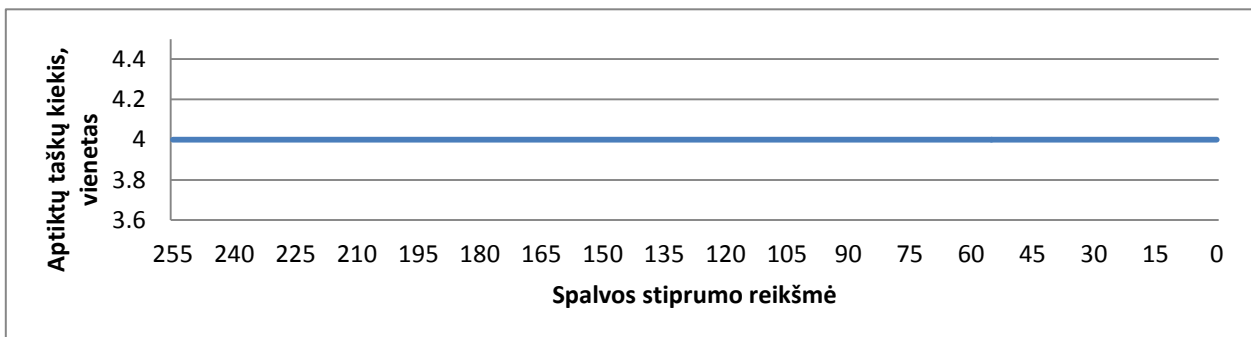
Tyrimo eiga: paleidžiama programa. Keičiamos parametrų vertės nuo minimalios iki maksimalios galimos vertės arba atvirkščiai. Stebima, kiek taškų yra aptinkama vaizde, ar aptinkami ieškomi keturi taškai. Nustatomos vertės, su kuriomis algoritmas veikia optimaliai. Parenkamos pradinės algoritmo vykdymo reikšmės.

Parametrai, naudojami algoritmo vykdyme:

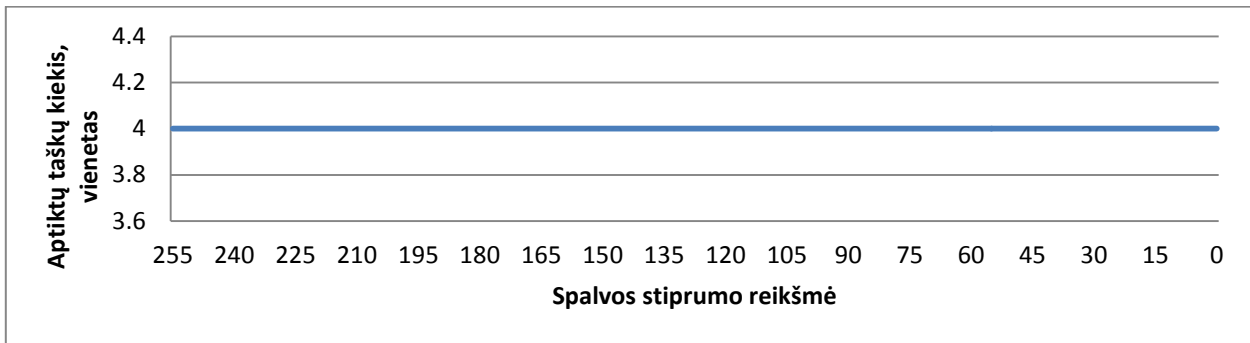
- filtravimo (*angl. threshold*);
- kontūrų aptikimo (*angl. Canny threshold*);
- kontūrų jungumo (*angl. Canny threshold linking*);
- 4 taškų aptikimo paklaidos (*angl. 4 point locking error*).



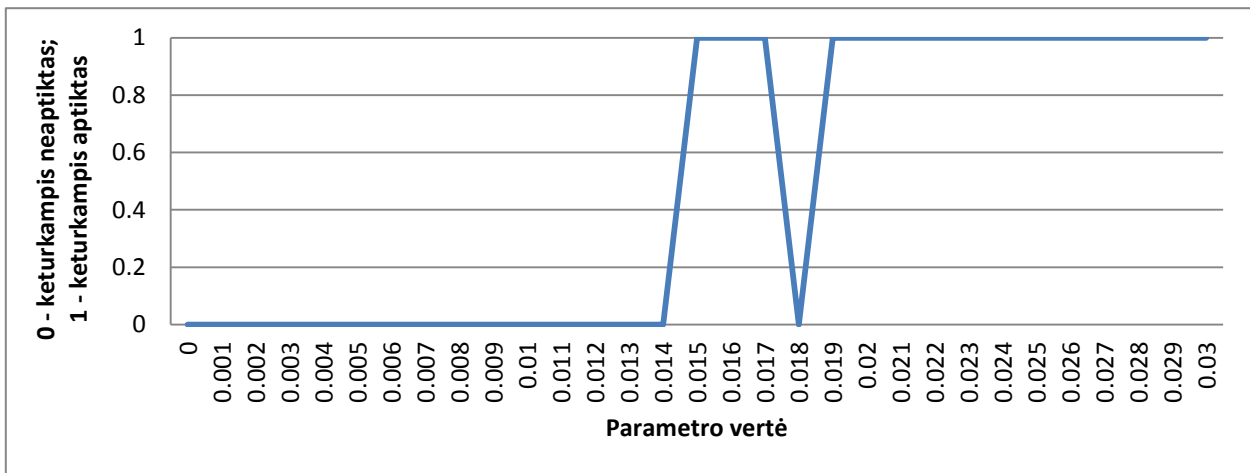
3.15 pav. Filtravimo parametro įtaka taškų aptikimui



3.16 pav. Kontūrų aptikimo parametro įtaka taškų aptikimui



3.17 pav. Kontūrų jungumo parametro įtaka taškų aptikimui



3.18 pav. 4 taškų aptikimo paklaidos parametro įtaka keturkampio aptikimui

Išanalizavus gautus duomenis yra nustatyta jog optimalūs parametrai, vykdant algoritmą su kamera, kurios skiriamoji geba yra 640x480:

- $50 < \text{Filtravimo parametro vertė} < 250$; pasirinkta pradinė vertė – 230 (žr. 3.15 pav.);
- kontūrų aptikimo vertė neturi įtakos algoritmo vykdymui; pasirinkta pradinė vertė – 180 (žr. 3.16 pav.);
- kontūrų jungumo vertė neturi įtakos algoritmo vykdymui; pasirinkta pradinė vertė – 120 (žr. 3.17 pav.);
- 4 taškų aptikimo paklaidos $> 0,019$; pasirinkta pradinė vertė – 0,02 (žr. 3.18 pav.).

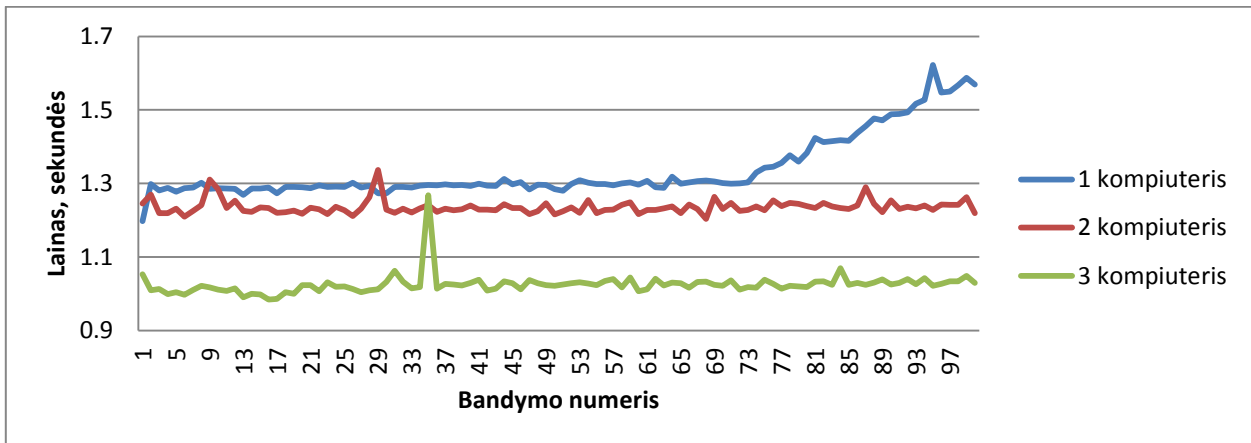
3.5. Algoritmo vykdymo laiko tyrimas įvairiose aplinkose

Tyrimo tikslas: įvertinti algoritmo vykdymo laiką. Palyginti su kitų autorių siūlytų metodų algoritmų spartumu.

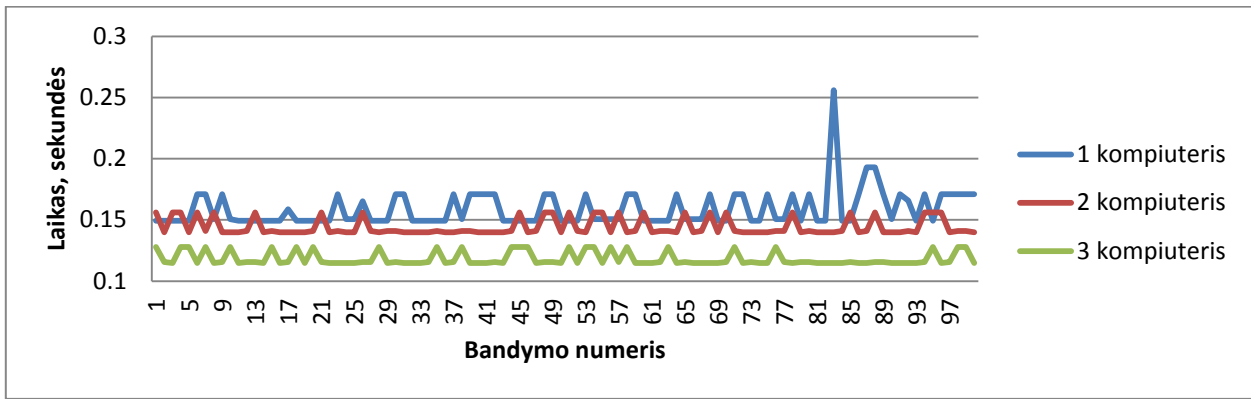
Tyrimo eiga: paleidžiama programa. Algoritmo vykdymo laikas yra fiksuojamas 100 kartų. Rezultatai pateikiami grafiškai. Apskaičiuojamas algoritmo vykdymo laiko vidurkis. Algoritmai yra tiriami su 3 skirtingais kompiuteriais. Nustatomas greičiausias iš suprogramuotų algoritmų įvairiose aplinkose. Rezultatai lyginami su kitų autorių pateiktais rezultatais.

Naudotų kompiuterių techniniai parametrai:

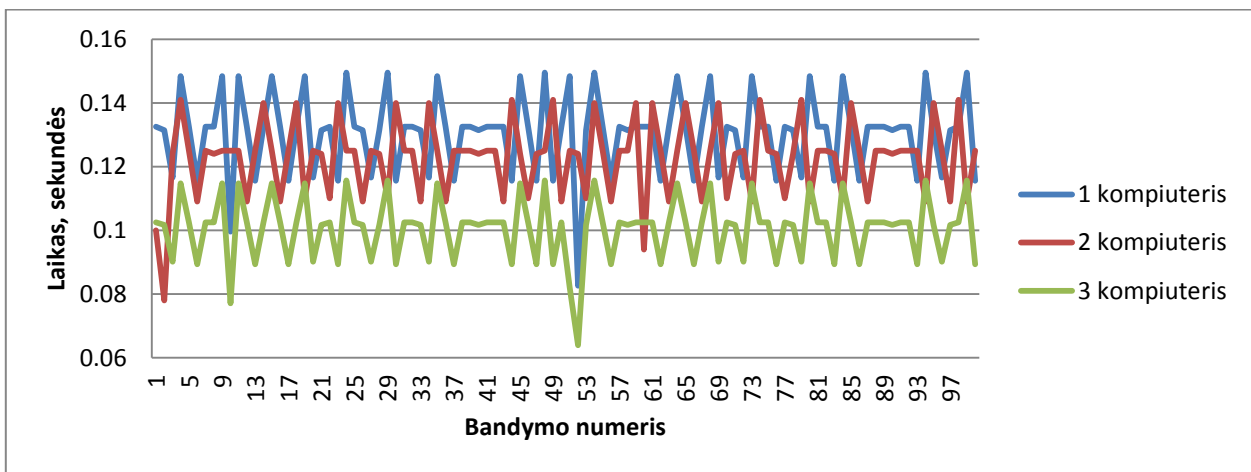
1. nešiojamas kompiuteris Acer extensa 5220:
 - procesorius: Intel celeron CPU 530 – 1,73 GHz;
 - operatyvioji atmintis (RAM): DDR2 2,00 Gb;
 - kietasis diskas: 120 Gb;
 - vaizdo plokštė: Nvidia Geforce 8400M – 128 Mb;
2. nešiojamas kompiuteris Asus K50I:
 - procesorius: Intel Pentium dual core T4500 – 2,3 GHz;
 - operatyvioji atmintis (RAM): DDR3 4,00 GB;
 - kietasis diskas: 500 GB;
 - vaizdo plokštė: Nvidia Geforce GT 320M – 1,00 GB;
3. stacionarus kompiuteris:
 - procesorius: Intel Core I3 540 – 3,06 GHz; 3,07 GHz;
 - operatyvioji atmintis (RAM): DDR3 4,00 GB;
 - kietasis diskas: 500 GB;
 - vaizdo plokštė: ATI Radeon HD 5670 – 1,00 GB.



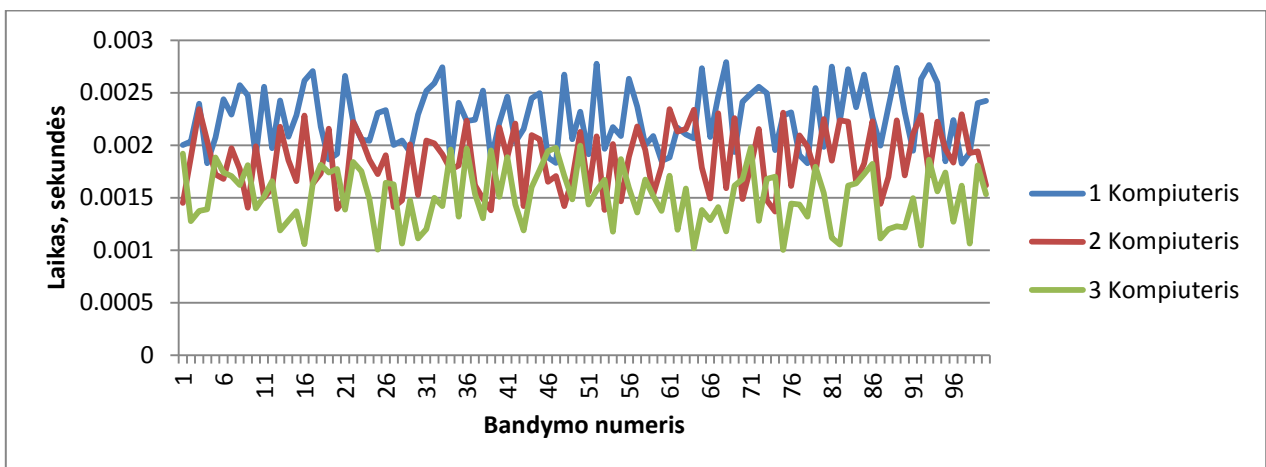
3.19 pav. Algoritmo vykdymo laikas Matlab aplinkoje



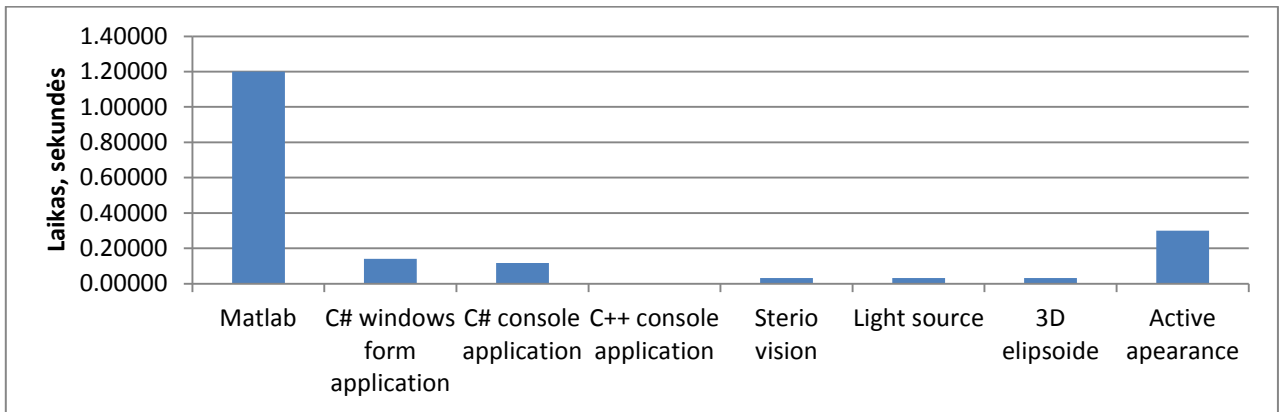
3.20 pav. Algoritmo vykdymo laikas C# windows form application aplinkoje



3.21 pav. Algoritmo vykdymo laikas C# console application aplinkoje



3.22 pav. Algoritmo vykdymo laikas C++ console application aplinkoje



3.23 pav. Algoritmų spartos palyginimas

Nustatyta jog lėčiausiai yra vykdomas algoritmas Matlab aplinkoje (žr. 3.19 pav.). Sparčiausias iš realizuotų algoritmų yra C++ console application (žr. 3.22 pav.). Iš C# kalbos programų sparčiausia buvo console application (žr. 3.21 pav.), tačiau ši aplinka neturi grafinės vartotojo sąsajos. Pasirinkta aplinka yra C# Windows form application (žr. 3.20 pav.). Jos vykdymo laikas yra tinkamas tolimesniems tyrimams. Ji taip pat turi ir vartotojo grafinę sąsają. Rezultatai yra palyginti su kitų autorių siūlytais metodais. Kitų autorių siūlyti metodai yra spartesni už Matlab ir C#. Metodų spartumą įtakoja programavimo kalba. C# programavimo kalba yra lėtesnė nei C++. Visi autorių siūlyti metodai yra realizuoti C++ kalboje. Iš palyginimo histogramos (žr. 3.23 pav.) galime pastebėti jog realizuotas algoritmas yra spartesnis, nei kitų autorių siūlyti algoritmai. Jei yra programuojama ta pačia programavimo kalba.

3.6. Algoritmo segmentų vykdymo laiko įvertinimas

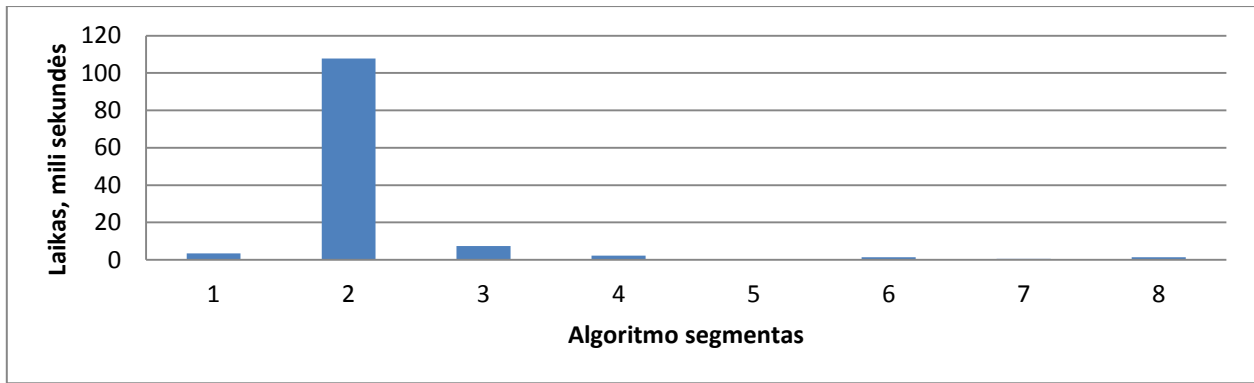
Tyrimo tikslas: iširti algoritmo segmentų vykdymo spartą ir įvertinti, kuris algoritmo segmentas labiausia lėtina algoritmą.

Tyrimo eiga: Tyrimas atliekamas C# windows form application programa, ji yra suskirstoma į segmentus. Algoritmas yra vykdomas 100 kartų. Fiksuojamas kiekvieno segmento vykdymo laikas ir įvertinamas vidurkis. Išrenkamas ilgiausiai vykdomas segmentas.

Algoritmas yra suskirstomas į segmentus:

1. vaizdo gavimas;
2. vaizdo filtravimas;
3. IR telkinių aptikimas ir centrinės koordinatės nustatymas;
4. taškų filtravimas;

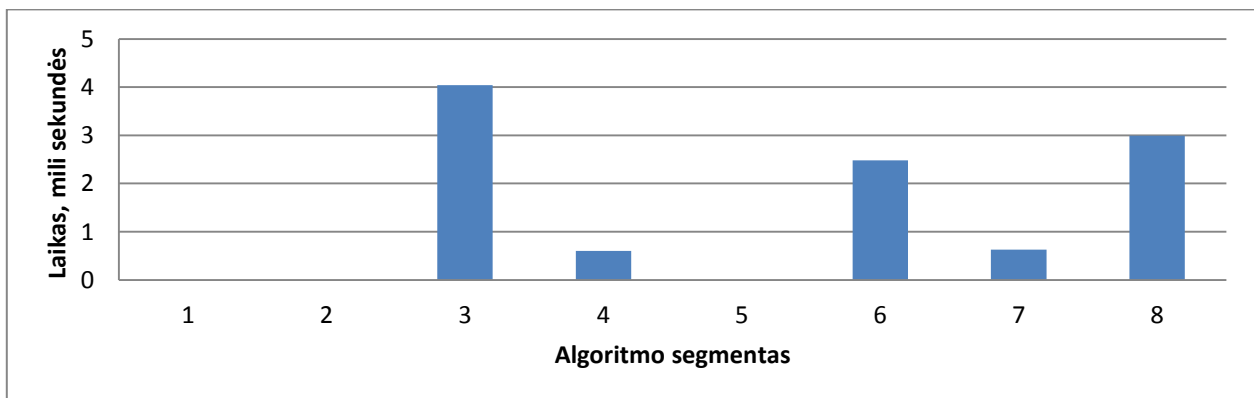
5. taškų rūšiavimas;
6. taškų koordinatų tikslinimas;
7. parametrų apskaičiavimas;
8. duomenų išvedimas.



3.24 pav. Algoritmo segmentų vykdymo laikas

Iš gautų duomenų (žr. 3.24 pav.) pastebėta, jog ilgiausiai yra vykdomas 2 segmentas – vaizdo filtravimas. Visi kiti segmentai vykdomi greičiau nei 10 ms. Antras algoritmo segmentas vykdomas ilgai, nes yra tikrinami visi taškai ir keičiamos reikšmės vaizde pagal vaizdo filtravimo parametro reikšmę.

Siekiant pagerinti algoritmo vykdymo laiką ieškomas alternatyvus metodas vaizdo filtravimui. Pasirinktas standartinis metodas siūlomas algoritmo filtravimui, fono ir ieškomo ryškumo objektų atskyrimui – komporavimas (*angl. Binary threshold*). Pritaikius šį metodą filtravimo vykdymo laikas ženkliai sumažėjo (žr. 3.25 pav.). Sukurtas algoritmas yra tinkamas darbui realiu laiku. Jo greitis vidutiniškai yra 10.74 ms jei vaizdas yra skaitomas iš failo. Jei vaizdas yra gaunamas iš kameros, sistemos veikimo greitis yra apribotas kameros sparta.

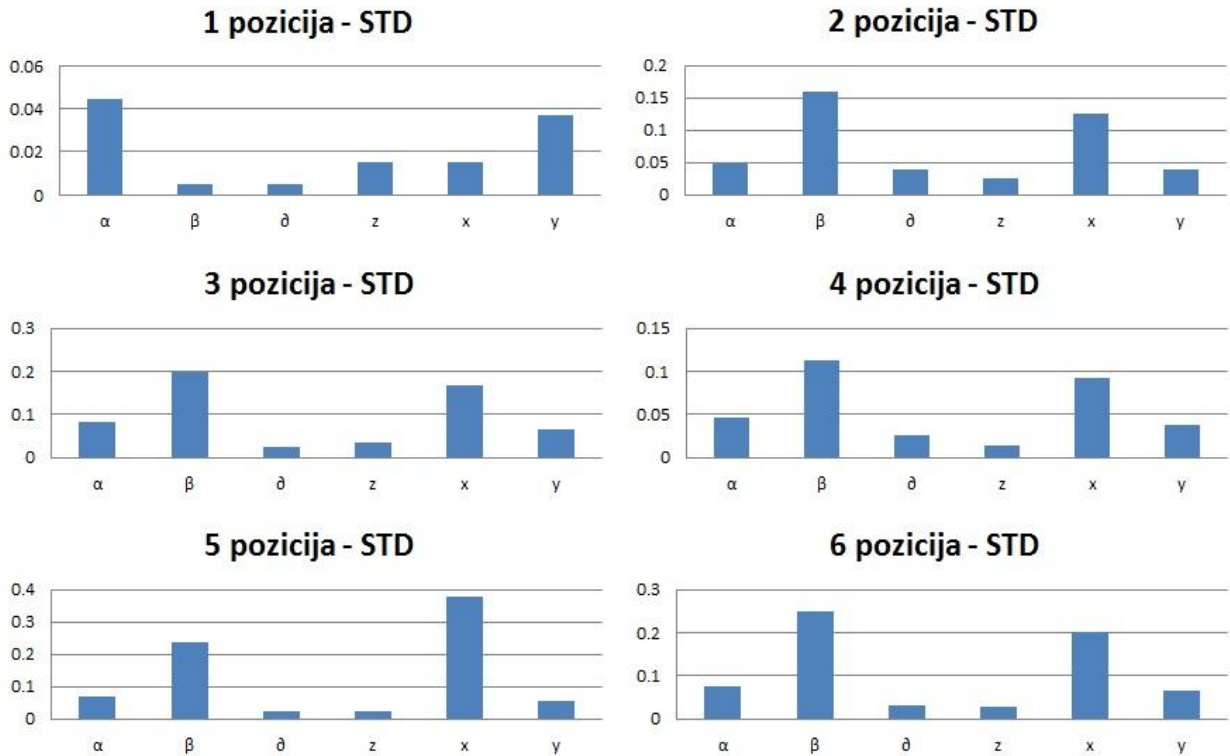


3.25 pav. Algoritmo segmentų vykdymo laikas optimizavus vaizdo filtravimą

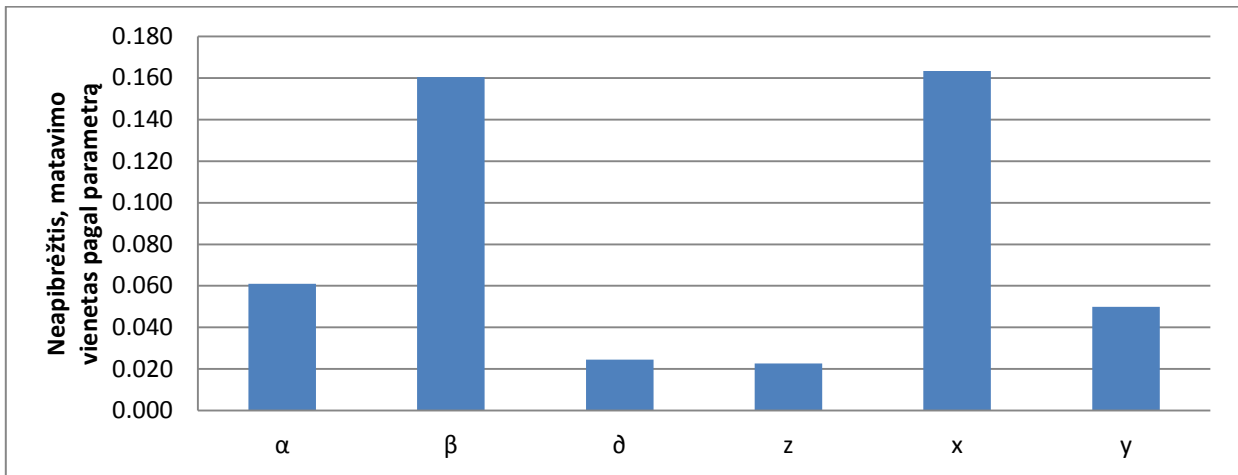
3.7. Algoritmo tikslumo įvertinimas

Tyrimo tikslas: įvertinti sistemos neapibrėžtis.

Tyrimo eiga: fiksuojama kamera stabilioje padėtyje. Yra imama 100 kadrų stabilioje padėtyje. Gautuose parametruose stebimas jų pokytis nekeičiant kameros padėties. Neapibrėžtis įvertinama apskaičiuojant standartinį nuokrypį iš gautų reikšmių. Eksperimentas yra kartojamas keičiant kameros padėtį. Įvertinamos atsitiktinės neapibrėžtys. Pasinaudojant sukonstruotu stendu universitete, yra fiksuojami parametrai. Fiksuojamos reikšmės keičiant atstumą x ir kampą β . Lyginamas realus ir programos apskaičiuotas pokytis. Nustatoma matavimo neapibrėžtis. Iš gautų rezultatų išvedama sistemos neapibrėžtis.



3.26 pav. Algoritmo atsitiktinės neapibrėžtys matuojant iš skirtingų pozicijų



3.27 pav. Algoritmo atsitiktinės neapibrėžtys

1 lentelė. Algoritmo atsitiktinės neapibrėžtys

Atsitiktinės neapibrėžtys					
α	β	δ	z	x	y
0.061	0.160	0.025	0.023	0.163	0.050

Išanalizavus duomenis (žr. 3.26 pav.) matuojant iš statinės padėties, pastebėta, jog matuojami parametrai yra iškraipomi trikdžių. Apskaičiuojami parametrai kinta tik tam tikrose ribose. Pagal gautus duomenis apskaičiuotos atsitiktinės neapibrėžtys (žr. 3.27 pav.), kurios yra pateiktos 1 lentelėje. Maksimali atsitiktinė neapibrėžtis matuojant galvos pozicijos padėtį – 0,163 cm ir orientacijos parametrus – 0,16°.

2 lentelė. Algoritmo matavimo neapibrėžtys keičiant x koordinatę

Parametrų reikšmės				Parametrų pokytis			Matavimo neapibrėžtis	
	$x = 0$	$x = +2.5$	$x = +5$		$\Delta x = 2.5$	$\Delta x = 5$		
α	4.8078	4.7592	4.7269	α	-0.0486	-0.0809	α	0.0648
β	-4.2558	-4.3677	-4.3158	β	-0.1119	-0.0600	β	0.0860
δ	-3.6158	-3.6667	-3.7937	δ	-0.0509	-0.1779	δ	0.1144
z	72.3574	72.5324	72.7775	z	0.1750	0.4201	z	0.2976
x	5.9138	8.3462	11.3012	x	2.4325	5.3874	x	0.1599
y	5.0955	5.0623	5.0078	y	-0.0332	-0.0877	y	0.0604

3 lentelė. Algoritmo matavimo neapibrėžtys keičiant β kampą

Parametrų reikšmės			Parametrų pokytis			Matavimo neapibrėžtis		
	$\beta = -5$	$\beta = 0$	$\beta = +5$		$\Delta\beta=-5$	$\Delta\beta=5$		
α	4.7247	4.7925	4.8245	α	-0.0678	0.0320	α	0.0179
β	-3.3486	1.4116	5.9387	β	-4.7602	4.5271	β	0.1166
ϑ	-3.3605	-3.4444	-3.4890	ϑ	0.0839	-0.0446	ϑ	0.0196
z	71.4193	71.9976	71.7695	z	-0.5783	-0.2281	z	0.4032
x	-6.3604	-6.3451	-6.2865	x	-0.0153	0.0586	x	0.0216
y	5.1546	5.0058	5.2735	y	0.1488	0.2678	y	0.2083

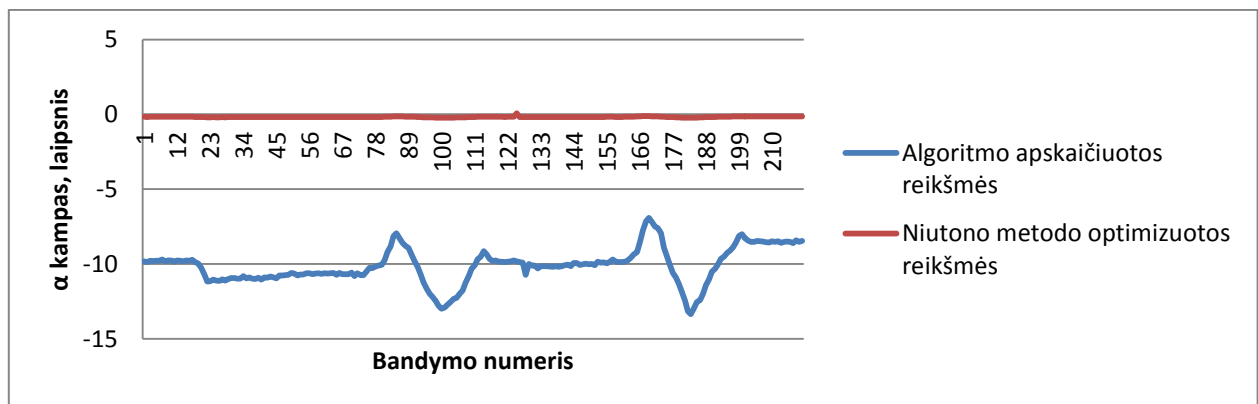
Išanalizavus gautus duomenis apie kameros padėties ar kampo keitimą, pastebima, jog maksimali matavimo neapibrėžtis matuojant galvos padėties parametrus yra 0,40 cm, o matuojant orientacijos parametrus – 0,11°.

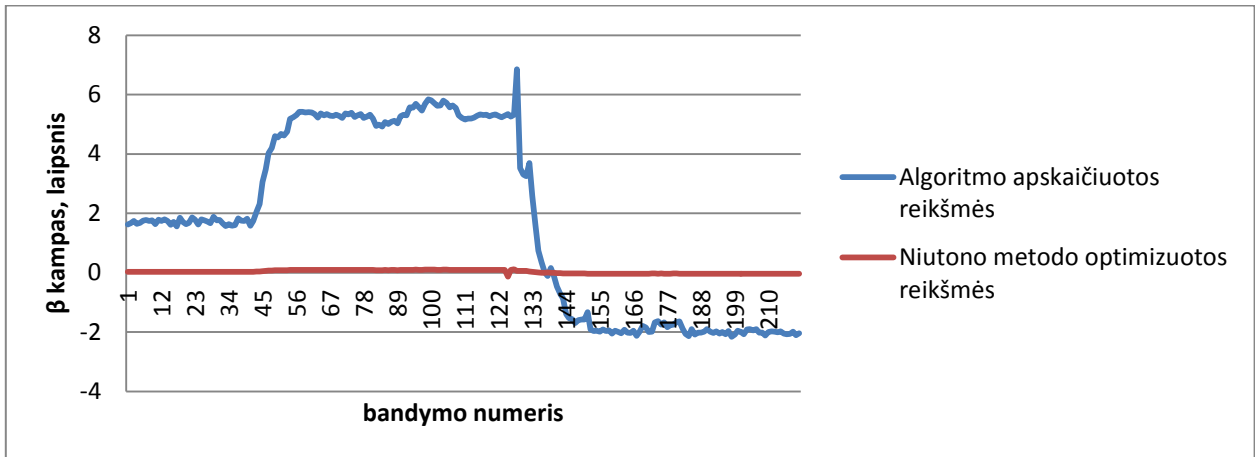
Sistemos tikslumas pozicijos $\pm 0,5$ cm ir orientacijos $\pm 0,5^\circ$ parametrums.

3.8. Ieškomų reikšmių optimizavimas Niutono metodu

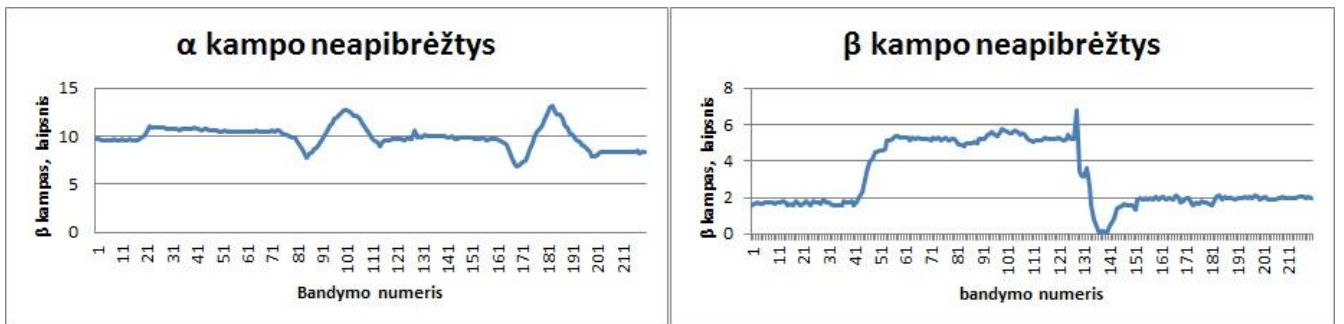
Tyrimo tikslas: įvertinti Niutono optimizavimo metodo pritaikymą sistemoje ieškomų verčių patikslinimui.

Tyrimo eiga: fiksuojama kamera stabilioje padėtyje. Tyrimo eigoje keičiami parametrai α ir β . Apskaičiuojamos reikšmės sukurto algoritmo pagalba. Reikšmės tikslinamos pritaikant Niutono optimizavimo metodą. Lyginamos optimizuotos ir algoritmo rastos reikšmės (žr. 3.28 ir 3.29 pav.). Įvertinamas optimizavimo metodo pritaikomumas sistemoje.

3.28 pav. Reikšmių optimizavimas Niutono metodu, α kampas



3.29 pav. Reikšmių optimizavimas Niutono metodu, β kampas



3.30 pav. Neapibrėžtus optimizavus sistemos reikšmes

Išanalizavus gautus duomenis pastebėta jog Niutono optimizavimo metodas netinka kuriamai sistemai. Ieškant globalaus minimumo yra aptinkamas lokalusis ir reikšmės yra iškraipomos. Gautieji rezultatai po optimizavimo yra mažiau tikslūs nei prieš jį.

IŠVADOS IR SIŪLYMAI

- Išanalizavus galvos padėties nustatymo metodus ir sistemas, pasirinkta sistema, kuomet kamera yra fiksuojama vartotojui ant galvos.
- Parinkti keturkampio geometriniai parametrai suteikiantys daugiausia informacijos apie kiekvieną iš galvos pozicijos ir orientacijos parametrų. Nustatyta tarpusavio priklausomybė ir skaičiavimo eiliškumas galvos padėties ir orientacijos parametrams.
- Sukonstruotas vaizduoklio maketas su 4 IR diodais jo kampuose. Vaizduoklio maketas yra naudojamas sistemos testavimui realiu laiku.
- Sukurtas vaizdo apdorojimo algoritmas 4 skirtingose aplinkose: Matlab, C# windows form application, C# console application ir C++ console application.
- Ištirta algoritmo parametrų įtaka taškų aptikimui ir filtravimui. Rekomenduojamos pradinės algoritmo parametrų reikšmės:
 - filtravimo (*angl. Threshold*) vertė – 230;
 - kontūrų aptikimo (*angl. Canny Threshold*) vertė – 180;
 - kontūrų jungumo (*angl. Canny Threshold Linking*) vertė – 120;
 - 4 taškų aptikimo paklaidos (*angl. 4 point locking error*) vertė – 0,02.
- Ištirtas algoritmų vykdymo laikas realizuotose aplinkose:
 - Matlab aplinkoje – vidutiniškai 1,2 sekundės.
 - C# windows form application aplinkoje – vidutiniškai 10.74 ms
 - C# console application aplinkoje – vidutiniškai 9.56 ms.
 - C++ console application aplinkoje – vidutiniškai 2 ms.
- C++ programoje lėčiausiai vykdomas segmentas yra atvaizdų koordinačių nustatymas. Siekiant dar paspartinti algoritmo veikimą reikia optimizuoti šį segmentą.
- Pastebėta, jog algoritmo tikslumo įvertinimą galima suskirstyti į atsitiktines ir matavimo neapibrėžtis.
- Sistemos tikslumas pozicijos $\pm 0,5$ cm ir orientacijos $\pm 0,5^\circ$ parametrms.
- Pritaikytas Niutono optimizavimo metodas ir nustatyta jog jis netinkamas kuriamai sistemai. Optimizavimo metu ieškant globaliojo minimumo yra aptinkamas lokalis.
- C# ir C++ aplinkose kurtų programų paleidimui yra sugeneruoti instaliaciniai failai. Papildomos programinės įrangos nereikia.
- Matlab kurtos programos paleidimui reikalinga papildoma programa Matlab.

- Siekiant pagerinti sistemos darbą vaizduoklio maketui naudoti tik pilnai įkrautus elementus
- Siekiant sumažinti pašalinių trikdžių kiekį, reikia vengti tiesioginių IR šviesos šaltinių (pvz.: saulės spindulių).
- Siekiant algoritmą vykdyti nurodytu spartumu informaciją reikia skaityti iš failo. Jei informacija yra imama iš kameros atsiranda vėlinimas dėl techninės įrangos.

LITERATŪRA

1. Carroll J. M. Human – computer interaction in the new millennium, New York, 2002, ISBN 978-0201704471.
2. EyeWriter sistemos aprašymas, [žiūrėta 2011-03-25] prieiga per internetą: <http://www.eyewriter.org/diy/>
3. Bortolami S. Localization of the subjective vertical during roll, pitch, and recumbent yaw body tilt/ S. Bortolami, A. Pierobon, P. DiZio, J. R. Lackner// Expérimentation cérébrale, 2006, 173(3):364-73.
4. Matsumoto Y. Behavior Recognition Based on Head Pose and Gaze Direction Measurement/ Y. Matsumoto T. Ogasawara, A. Zelinsky// Proceedings of 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2000), 2000, pp.2127-2132.
5. Matsumoto Y. An Algorithm for Real-time Stereo Vision Implementation of Head Pose and Gaze Direction Measurement/ Y. Matsumoto, A. Zelinsky// Proceedings of IEEE Fourth International Conference on Face and Gesture Recognition (FG'2000), 2000, pp.499-505.
6. Foxlin E. Wear Track: A Self - Referenced Head and Hand Tracker for Wearable Computers and Portable VR/ E. Foxlin, M. Harrington// Proceedings of International Symposium on Wearable Computers (ISWC 2000), 2000, pp.155-162.
7. Göktürk B. S. 3D Head Tracking Based on Recognition and Interpolation Using a Time-Of-Flight Depth Sensor/ B. S. Göktürk, C. Tomasi// IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04) - Volume 2, 2004, pp.211-217.
8. Basu S. Motion Regularization for Model-Based Head Tracking/ S. Basu, I. Essa A. Pentland// Proceedings of the 13th IEEE Int'l Conf. on Pattern Recognition (ICPR '96), 1996, pp. 611-616.
9. Ahlberg J. Using the active appearance algorithm for face and facial feature tracking// Swedish Symposium on Image Analysis, 2001, pp.1–4.
10. Cootes T. F. Active Appearance Models/ T. F. Cootes, J. G. Edwards, J. C. Taylor// IEEE Trans. Pattern Anal. Mach (Intell.23), 2001, pp.681-685.
11. Satoh K. A Head Tracking Method Using Bird's-Eye View Camera and Gyroscope/ K. Satoh, S. Uchiyama, H. Yamamoto// Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04), 2004, pp.202-21.
12. La Cascia M. Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models/ M. La Cascia, S. Sclaroff, V. Athitsos// IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 4, 2000, pp.322-336.

13. Gonzalez R. C. Digital image processing/ R. C. Gonzalez, R. E. Woods// Addison-Wesley Longman Publishing Co. Inc. Boston, JAV, 2001, ISBN 0-13-094650-8.
14. Jahne B. Computer vision and applications, a guide for students and practitioners/ B. Jahne, H. Haubecker// Academic press, San Diego, JAV, 2000, ISBN 0-12-379777-2.
15. Gonzalez R. C. Thresholding. In Digital Image Processing/ R. C. Gonzalez, R. E. Woods// Pearson Education, 2002, ISBN 81-7808-629-8, pp. 595–61.
16. Engel K. Real-time volume graphics/ K. Engel, M. Hadwiger, J. Kniss, C. Rezk-Salama, D. Weiskopf// CRC Press 2006, ISBN 1-56881-266-3, pp. 112–114.
17. Lindeberg T. "Edge detection", in Hazewinkel, Michiel, Encyclopedia of Mathematics// Springer, 2001, ISBN 978-1-55608-010-4.
18. Pelgrims P. Sub-pixel edge detection/ P. Pelgrims, G. Van de Velde, B. Van de Vondel// De Nayer Instituut, 2004, OPL v 1.0.
19. Chui C. K. Kalman Filtering with Real-Time Applications/ C. K. Chui, G. Chen// Springer Series in Information Sciences, New York, 2009, ISBN 978-3-540-87848-3, pp. 229.
20. Bonnans J. F. Numerical Optimization/ J. F. Bonnans, J. C. Gilbert, C. Lemarechal, C. A. Sagastizábal// Springer, 2006, ISBN 978-3-540-35445-1
21. Gonzalez R. C. Digital image processing using Matlab/ R. C. Gonzalez, R. E. Woods, S. L. Eddins//Addison-Wesley Longman Publishing Co. Inc. Boston, JAV, 2004, ISBN 0-13-008519-7.
22. Programavimo kalbos C# žinynas, [žiūrėta 2012-02-12] prieiga per internetą: <http://msdn.microsoft.com/en-us/library/618ayhy6.aspx>.
23. C# vaizdų apdorojimo bibliotekos Emgu.CV žinynas, [žiūrėta 2012-02-12] prieiga per internetą: http://www.emgu.com/wiki/index.php/Main_Page.
24. Programavimo kalbos C++ žinynas, [žiūrėta 2012-05-25] prieiga per internetą: <http://www.cplusplus.com/reference/>
25. C++ vaizdų apdorojimo bibliotekos Open.CV žinynas, [žiūrėta 2012-05-25] prieiga per internetą: <http://opencv.willowgarage.com/documentation/cpp/>

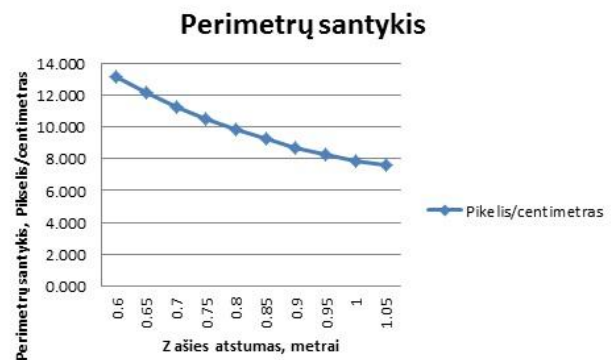
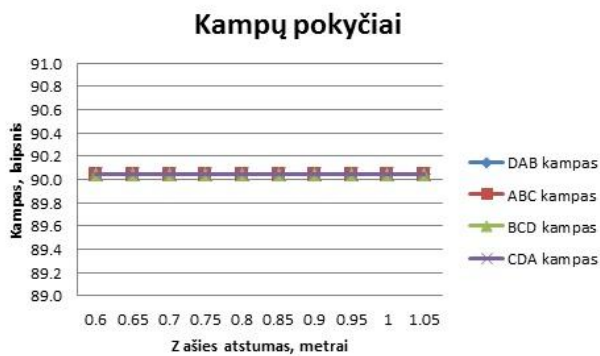
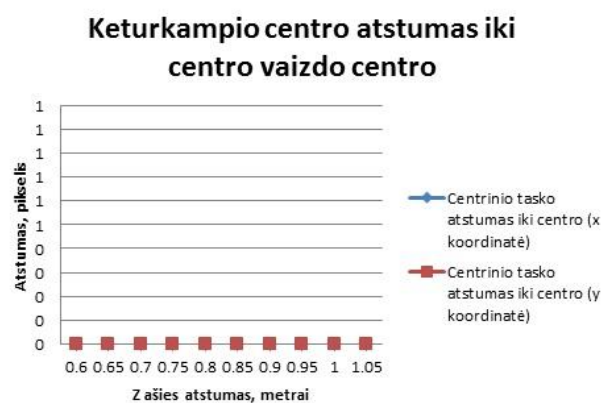
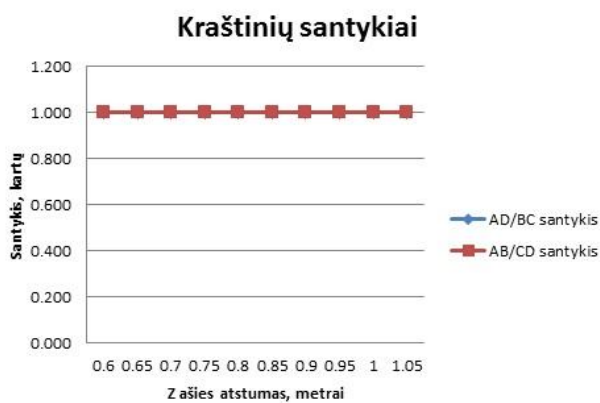
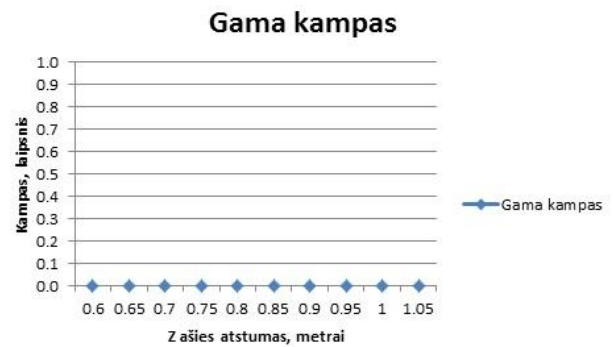
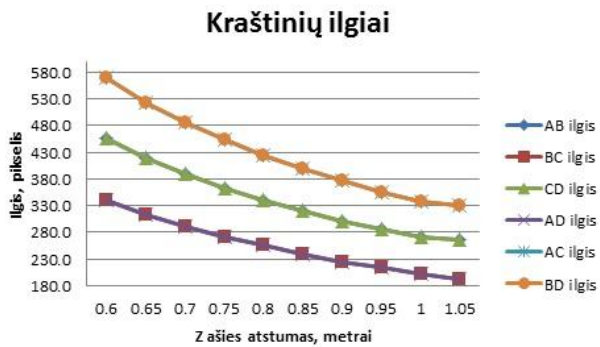
PRIEDAI

1 PRIEDAS

GEOMETRINIAI KETURKAMPIO PARAMETRAI KEIČIANT Z AŠIES ATSTUMĄ (1)

Z atstumo pokytis metrais	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1	1.05
Ilgiai pikseliais										
AB ilgis	455.0	419.0	389.0	363.0	339.0	319.0	301.0	285.0	271.0	267.0
BC ilgis	341.0	314.0	291.0	272.0	255.0	239.0	225.0	213.0	203.0	193.0
CD ilgis	455.0	419.0	389.0	363.0	339.0	319.0	301.0	285.0	271.0	267.0
AD ilgis	341.0	314.0	291.0	272.0	255.0	239.0	225.0	213.0	203.0	193.0
AC ilgis	568.6	523.6	485.8	453.6	424.2	398.6	375.8	355.8	338.6	329.5
BD ilgis	568.6	523.6	485.8	453.6	424.2	398.6	375.8	355.8	338.6	329.5
Santykiai										
AD/BC santykis	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
AB/CD santykis	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Kampai laipsniais										
DAB kampas	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0
ABC kampas	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0
BCD kampas	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0
CDA kampas	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0	90.0
Gama kampas	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Atstumai pikseliais										
Centrinio tasko atstumas iki centro (x koordinatė)	0	0	0	0	0	0	0	0	0	0
Centrinio tasko atstumas iki centro (y koordinatė)	0	0	0	0	0	0	0	0	0	0
Perimetrų santykis										
Pikelis/centimetras	13.179	12.136	11.258	10.513	9.834	9.238	8.709	8.245	7.848	7.616

GEOMETRINIŲ KETURKAMPIO PARAMETRŲ POKYČIAI KEIČIANT Z ATSTUMĄ (2)

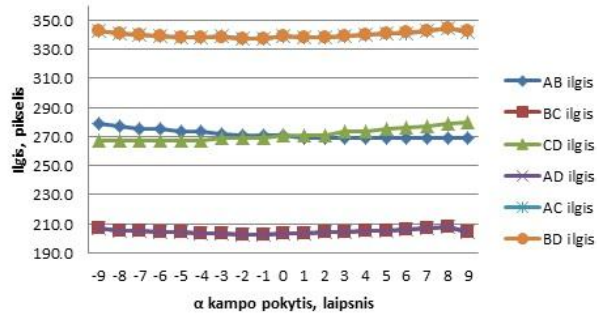


GEOMETRINIAI KETURKAMPIO PARAMETRAI KEIČIANT α KAMPĄ (1)

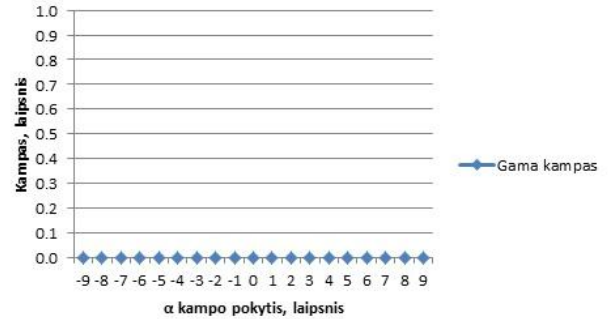
Alfa kampo pokytis laipsniais	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9
Ilgiai pikseliais																			
AB ilgis	279.0	277.0	275.0	275.0	273.0	273.0	272.0	271.0	271.0	271.0	269.0	269.0	269.0	269.0	269.0	269.0	269.0	269.0	269.0
BC ilgis	207.1	205.1	205.0	204.0	204.0	203.0	203.0	202.0	202.0	203.0	203.0	204.0	204.0	205.0	205.0	206.0	207.0	208.1	204.1
CD ilgis	267.0	267.0	267.0	267.0	267.0	267.0	269.0	269.0	269.0	271.0	271.0	271.0	273.0	273.0	275.0	276.0	277.0	279.0	280.0
AD ilgis	207.1	205.1	205.0	204.0	204.0	203.0	203.0	202.0	202.0	203.0	203.0	204.0	204.0	205.0	205.0	206.0	207.0	208.1	204.1
AC ilgis	342.6	340.6	339.8	339.2	338.4	337.8	338.6	337.2	337.2	338.6	337.8	338.4	339.2	339.8	340.6	341.2	342.6	344.0	341.6
BD ilgis	342.6	340.6	339.8	339.2	338.4	337.8	337.8	337.2	337.2	338.6	337.8	338.4	339.2	339.8	340.6	342.0	342.6	344.0	342.4
Santykiai																			
AD/BC santykis	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
AB/CD santykis	1.045	1.037	1.030	1.030	1.022	1.022	1.011	1.007	1.007	1.000	0.993	0.993	0.985	0.985	0.978	0.975	0.971	0.964	0.961
Kampai laipsniais																			
DAB kampas	88.4	88.6	88.9	88.9	89.2	89.2	89.5	89.8	89.8	90.0	90.3	90.3	90.6	90.6	90.9	91.2	91.2	91.4	91.7
ABC kampas	88.4	88.6	88.9	88.9	89.2	89.2	89.8	89.8	89.8	90.0	90.3	90.3	90.6	90.6	90.9	90.9	91.2	91.4	91.5
BCD kampas	91.7	91.4	91.2	91.2	90.9	90.9	90.3	90.3	90.3	90.0	89.8	89.8	89.5	89.5	89.2	89.2	88.9	88.7	88.6
CDA kampas	91.7	91.4	91.2	91.2	90.9	90.9	90.6	90.3	90.3	90.0	89.8	89.8	89.5	89.5	89.2	88.9	88.9	88.7	88.4
Gama kampas	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Atstumai pikseliais																			
Centrinio tasko atstumas iki centro (x koordinatė)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Centrinio tasko atstumas iki centro (y koordinatė)	109	95	81	67	53	40	26	12	-2	-16	-30	-43	-57	-71	-85	-98	-113	-126	-138
Perimetrų santykis																			
Pikelis/centimetras	7.880	7.898	7.881	7.865	7.848	7.831	7.840	7.815	7.815	7.848	7.831	7.848	7.864	7.881	7.850	7.830	7.850	7.890	7.900

GEOMETRINIŲ KETURKAMPIO PARAMETRŲ POKYČIAI KEIČIANT α KAMPĄ (2)

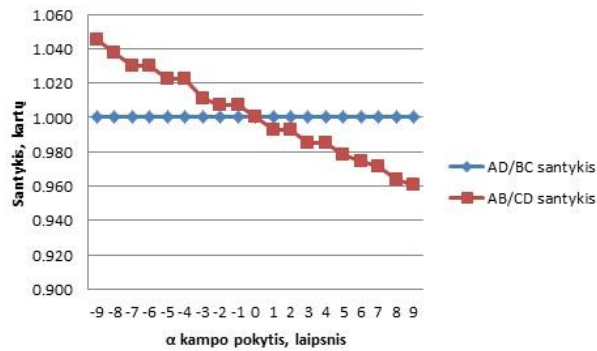
Kraščių ilgiai



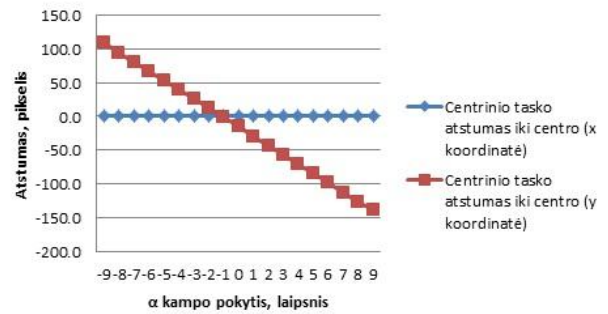
Gama kampas



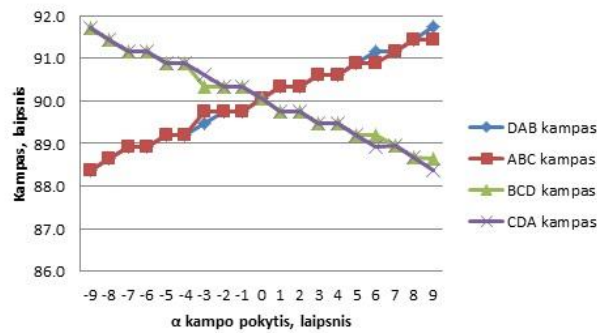
Kraščių santykiai



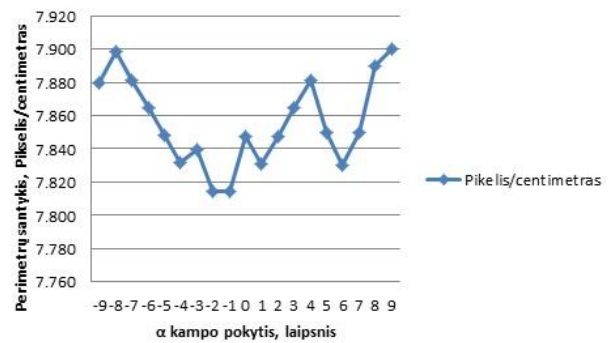
Keturkampio centro atstumas iki centro vaizdo centro



Kampų pokyčiai

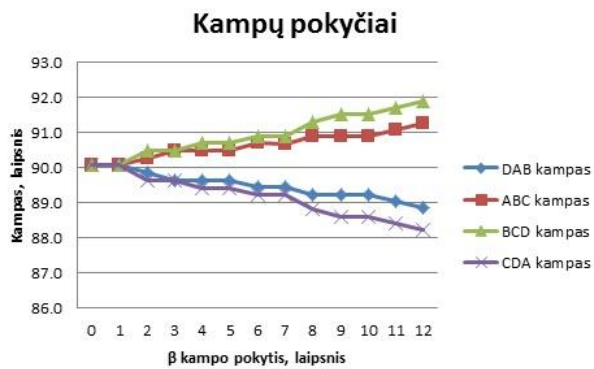
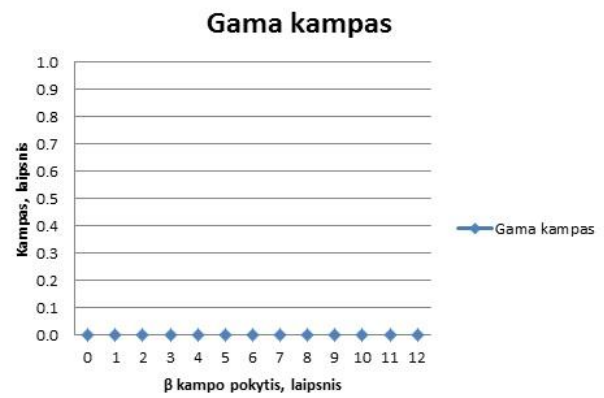
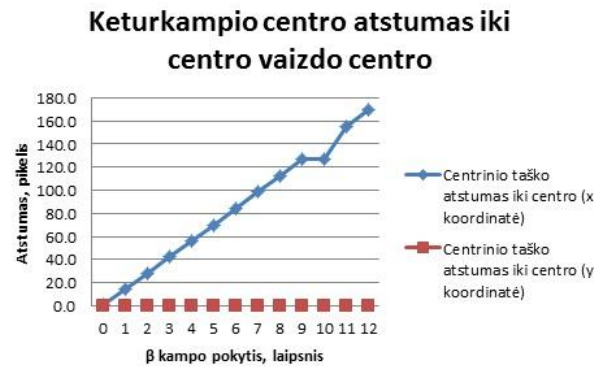
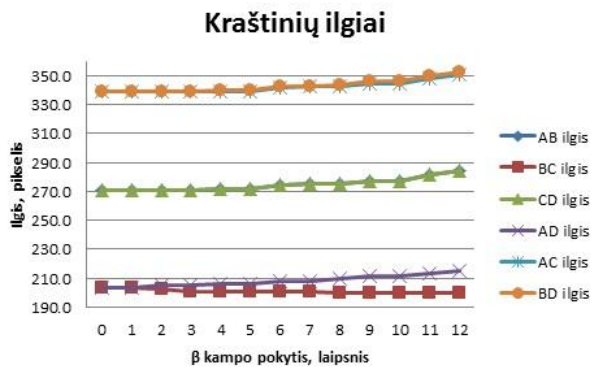


Perimetrų santykis



GEOMETRINIAI KETURKAMPIO PARAMETRAI KEIČIANT β KAMPĄ (1)

Beta kampo pokytis laispniais	0	1	2	3	4	5	6	7	8	9	10	11	12
Ilgiai pikseliais													
AB ilgis	271.0	271.0	271.0	271.0	272.0	272.0	274.0	275.0	275.0	277.0	277.0	281.0	284.1
BC ilgis	203.0	203.0	202.0	201.0	201.0	201.0	201.0	201.0	200.0	200.0	200.0	200.0	200.0
CD ilgis	271.0	271.0	271.0	271.0	272.0	272.0	274.0	275.0	275.1	277.1	277.1	281.1	284.1
AD ilgis	203.0	203.0	205.0	205.0	206.0	206.0	208.0	208.0	210.0	211.0	211.0	213.0	215.0
AC ilgis	338.6	338.6	338.6	338.6	339.4	339.4	341.6	342.4	342.4	344.0	344.0	347.8	350.8
BD ilgis	338.6	338.6	339.2	338.6	340.0	340.0	342.2	343.0	343.6	345.8	345.8	349.6	352.6
Santykiai													
AD/BC santykis	1.000	1.000	1.015	1.020	1.025	1.025	1.035	1.035	1.050	1.055	1.055	1.065	1.075
AB/CD santykis	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Kampai laispniais													
DAB kampas	90.0	90.0	89.8	89.6	89.6	89.6	89.4	89.4	89.2	89.2	89.2	89.0	88.8
ABC kampas	90.0	90.0	90.3	90.5	90.5	90.5	90.7	90.7	90.9	90.9	90.9	91.1	91.3
BCD kampas	90.0	90.0	90.5	90.5	90.7	90.7	90.9	90.9	91.3	91.5	91.5	91.7	91.9
CDA kampas	90.0	90.0	89.6	89.6	89.4	89.4	89.2	89.2	88.8	88.6	88.6	88.4	88.2
Gama kampas	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Atstumai pikseliais													
Centrinio taško atstumas iki centro (x koordinatė)	0.5	14.5	28.5	42.5	56.0	70.0	84.0	98.5	112.5	126.5	126.5	155.5	170.0
Centrinio taško atstumas iki centro (y koordinatė)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Perimetrų santykis													
Pikelis/centimetras	7.848	7.848	7.856	7.848	7.873	7.873	7.880	7.860	7.880	7.830	7.850	7.810	7.830

GEOMETRINIŲ KETURKAMPIO PARAMETRŲ POKYČIAI KEIČIANT β KAMPĄ (2)

GEOMETRINIAI KETURKAMPIO PARAMETRAI KEIČIANT δ KAMPĄ (1)

Gama kampo pokytis laipsniais	-50	-45	-40	-35	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30	35	40	45	50	
Ilgiai pikseliais																						
AB ilgis	270.4	271.5	270.4	269.9	270.1	272.9	270.5	270.2	270.1	270.0	271.0	270.0	270.1	268.3	270.5	270.2	270.1	269.9	270.4	269.4	269.8	
BC ilgis	203.7	202.9	202.3	202.5	202.9	203.1	202.5	203.0	203.0	203.1	203.0	202.7	203.0	202.8	203.1	203.1	202.6	202.5	203.1	203.6	202.9	
CD ilgis	270.4	270.1	269.8	270.8	270.7	270.2	270.1	269.3	270.1	265.0	271.0	271.1	270.1	269.3	269.2	270.2	271.5	269.9	269.8	270.1	270.4	
AD ilgis	203.7	202.9	203.1	203.1	202.1	201.9	203.4	202.8	203.0	202.6	203.0	203.6	203.0	202.5	202.5	203.1	202.9	202.5	202.3	202.9	202.2	
AC ilgis	338.5	338.4	337.9	337.7	338.2	339.2	337.6	337.7	338.2	334.9	338.6	337.4	337.7	335.6	337.5	338.3	338.4	337.2	337.5	337.3	336.7	
BD ilgis	338.6	338.4	337.5	338.2	337.4	338.3	338.5	337.4	337.7	336.6	338.6	339.2	338.2	337.7	337.6	337.8	338.2	337.7	337.9	338.3	338.5	
Santykiai																						
AD/BC santykis	1.000	1.000	1.004	1.003	0.996	0.994	1.005	0.999	1.000	0.998	1.000	1.005	1.000	0.999	0.997	1.000	1.002	1.000	0.996	0.997	0.996	
AB/CD santykis	1.000	1.005	1.002	0.997	0.998	1.010	1.001	1.004	1.000	1.019	1.000	0.996	1.000	0.996	1.005	1.000	0.995	1.000	1.002	0.997	0.998	
Kampai laipsniais																						
DAB kampas	90.1	89.8	89.8	90.2	90.1	89.6	90.1	89.9	90.0	89.7	90.0	90.4	90.1	90.6	89.9	89.9	90.2	90.1	90.1	90.4	90.5	
ABC kampas	90.0	89.8	90.1	90.1	90.2	89.7	89.9	89.9	90.1	89.0	90.0	90.0	90.0	89.8	89.8	90.1	90.3	89.9	89.8	89.9	89.7	
BCD kampas	90.1	90.2	90.1	90.1	89.7	90.1	90.4	90.1	90.0	91.0	90.0	90.3	90.1	90.3	90.2	89.9	89.9	90.1	90.1	90.0	90.2	
CDA kampas	90.0	90.2	90.1	89.8	90.2	90.7	89.8	90.3	90.1	90.5	90.0	89.5	90.0	89.6	90.3	90.1	89.8	89.9	90.1	89.8	89.7	
Gama kampas	-50.0	-45.0	-39.9	-34.9	-30.2	-25.0	-19.9	-15.1	-10.0	-5.0	0.0	5.1	10.0	15.1	20.0	25.0	30.1	35.1	39.9	45.0	50.0	
Atstumai pikseliais																						
Centrinio tasko atstumas iki centro (x koordinatė)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Centrinio tasko atstumas iki centro (y koordinatė)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Perimetrų santykis																						
Pikelis/centimetras	7.850	7.844	7.827	7.834	7.829	7.849	7.836	7.825	7.834	7.830	7.848	7.843	7.834	7.840	7.825	7.837	7.841	7.822	7.827	7.832	7.825	

GEOMETRINIŲ KETURKAMPIO PARAMETRŲ POKYČIAI KEIČIANT θ KAMPĄ (2)