

ŠIAULI UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

Sandra Barkut

AKI JUDESI EKSPERIMENT TAIKINIO GENERAVIMO
METOD TYRIMAS

Magistro darbas

Vadovas
doc. dr. N. Ramanauskas

ŠIAULIAI, 2011

ŠIAULI UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

TVIRTINU
Katedros ved jās
doc. dr. G. Daunys

2011 06

Sandra Barkut

AKI JUDESI EKSPERIMENT TAIKINIO GENERAVIMO
METOD TYRIMAS

Magistro darbas

Vadovas
doc. dr. N. Ramanauskas

2011 06

Recenzentas
prof. habil. dr. V. Laurutis

2011 06

Atliko
RM-9 gr. stud.
S. Barkut

2011 06

ŠIAULIAI, 2011

SANTRAUKA

Barkut S., Aki judesi eksperiment taikinio generavimo metod tyrimas: Signal technologijos magistro darbas / mokslinis vadovas doc. dr. N. Ramanauskas; Šiauli universitetas, Technologijos fakultetas, Elektronikos katedra, Šiauliai, 2011. – 49p.

Nuotolin s video-okulografin s aki judesi matavimo sistemos turi mažiausi invazijos lyg , tod l yra labai paplitusios ir spar iai vystomos. Tokios sistemos pagalba Šiauli universitete, Technologijos fakultete sik r s Biomedicinin s inžinerijos mokslinis centras atlieka tyrimus, susijusius su aki judesiais. Aki judesi tyrimui mokslinio centro darbuotojai dažnai naudoja tolygia trajektorija judant taikin . Taikinio vaizdavimui naudojamas OpenGL metodas.

Šiame darbe tiriamos OpenGL metodo taikinio jud jimo trajektorijos koordina i generavimo galimybi ribos, metodo tikslumas, spartinimo galimyb s ir atliekamas lyginimas su kitu galimu taikinio jud jimo trajektorijos koordina i generavimo metodu.

SUMMARY

Barkut S., The Research of the Target Generation Method of Eye Movement: the master's Thesis of Signal Technologies / scientific advisor doc. dr. N. Ramanauskas; Šiauliai University, Technology Faculty, Electronics Department, Šiauliai, 2011. – 49 p.

The permanent systems of video-oculographic movement measuring have the lowest level of invasion, therefore are very common and expeditiously developed. The Scientific Centre of Biomedical Engineering situated with the help of such system in Šiauliai University, Technology Faculty performs researches in relation with the movements of eyes. The workers of the scientific centre usually use a target moving along an even trajectory for the research eye movements. The method OpenGL is used to depict the target.

In the present thesis, there are researched the limits of the possibilities of generating of the coordinates of the OpenGL target movement, the accuracy of the method, the possibilities of accelerating and there is performed the comparison with another possible method of generating of the coordinates of a target movement trajectory is performed.

TURINYS

PAVEIKSLAI.....	5
LENTELĖS.....	6
1. VADAS.....	7
2. TEORIJOS APŽVALGA.....	8
2.1. Video-okulografinis akių judesių matavimo sistemos.....	8
2.1.1. Realizacijos.....	8
2.1.2. Nuotolinis video-okulografinis AJMS.....	9
2.2. Taikinytis video-okulografinis AJMS.....	10
2.3. Generavimo metodai ir programavimo būdai.....	12
2.1.3. QPainter.....	12
2.1.4. OpenGL.....	12
2.1.5. QGLWidget.....	14
2.1.6. QThreads.....	14
2.1.7. Klasės.....	15
3. PROGRAMINIS RANGAS.....	16
3.1. Viena klasė.....	17
3.2. Dvi klasės.....	17
3.3. QGLWidget klasė.....	18
3.4. QThreads klasė.....	19
4. TYRIMO METODIKA.....	21
4.1. Taikinio judėjimo trajektorija.....	21
4.2. Dispersijos analizė.....	22
5. TYRIMO REZULTATAI.....	24
5.1. Programavimo būdų palyginimas prie kiekvieno generavimo metodo.....	24
5.1.1. QPainter metodas.....	24
5.1.2. OpenGL metodas.....	26
5.2. QPainter ir OpenGL metodų palyginimas prie kiekvieno programavimo būdo.....	30
5.2.1. Viena klasė.....	30
5.2.2. Dvi klasės.....	33
5.2.3. QGLWidget klasė.....	36
5.2.4. QThreads klasė.....	40
5.3. Standartinio nuokrypio kitimas.....	44
5.3.1. QPainter metodas.....	44
5.3.2. OpenGL metodas.....	45
6. IŠVADOS.....	47
LITERATŪROS ŠALTINIAI.....	48
PRIEDAI.....	49

PAVEIKSLAI

1 pav. Fiksuota video-okulografinis AJMS	8
2 pav. Šalminis video-okulografinis AJMS	8
3 pav. Nuotolinis video-okulografinis AJMS	9
4 pav. Nuotolinis video-okulografinis AJMS komponentai	10
5 pav. Qt piešimo sistema	12
6 pav. Gijimas veikimas	14
7 pav. Metodai ir programavimo būdai.....	16
8 pav. 1 algoritmas. Taikinio piešimas ir koordinatų generavimas vienoje klasėje.....	17
9 pav. 2 algoritmas. Taikinio piešimas ir koordinatų generavimas atskirose klasėse.....	18
10 pav. 3 algoritmas. Taikinio piešimas QGLWidget klasėje.....	19
11 pav. 4 algoritmas. Taikinio piešimas QThreads klasėje.....	20
12 pav. Taikinio judėjimo trajektorija.....	21
13 pav. Siektinas ir gautas taikinio generavimo laikas QPainter metodu	24
14 pav. Programavimo būdų palyginimas. Generavimo metodas: QPainter “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis.....	26
15 pav. Programavimo būdų palyginimas. Generavimo metodas: QPainter “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis.....	26
16 pav. Siektinas ir gautas taikinio generavimo laikas OpenGL metodu	27
17 pav. Programavimo būdų palyginimas. Generavimo metodas: OpenGL “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis.....	28
18 pav. Programavimo būdų palyginimas. Generavimo metodas: OpenGL “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis.....	29
19 pav. QPainter ir OpenGL gautas ir siektinas laikas. Programavimo būdas: viena klasė	30
20 pav. Generavimo metodų palyginimas. Programavimo būdas: viena klasė. “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis.....	32
21 pav. Generavimo metodų palyginimas. Programavimo būdas: viena klasė. “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis.....	32
22 pav. QPainter ir OpenGL gautas ir siektinas laikas. Programavimo būdas: dvi klasės	33
23 pav. Generavimo metodų palyginimas. Programavimo būdas: dvi klasės. “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis.....	35
24 pav. Generavimo metodų palyginimas. Programavimo būdas: dvi klasės. “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis.....	35
25 pav. QPainter ir OpenGL gautas ir siektinas laikas. Programavimo būdas: QGLWidget klasė	37
26 pav. Generavimo metodų palyginimas. Programavimo būdas: QGLWidget klasė. “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis	38
27 pav. Generavimo metodų palyginimas. Programavimo būdas: QGLWidget klasė. “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis	39
28 pav. QPainter ir OpenGL gautas ir siektinas laikas. Programavimo būdas: QThreads klasė	40
30 pav. Generavimo metodų palyginimas. Programavimo būdas: QThreads klasė. “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis	42
31 pav. Standartinio nuokrypio kitimas generuojant taikinį QPainter metodu	44
32 pav. Standartinio nuokrypio kitimas, generuojant taikinį OpenGL metodu	45

LENTELIS

1 lentel . QPainter gautas ir siektinas laikas prie vis programavimo b d	25
2 lentel . QPainter gautas ir siektinas laikas prie vis programavimo b d	25
3 lentel . OpenGL gautas ir siektinas laikas prie vis programavimo b d	27
4 lentel . OpenGL gautas ir siektinas laikas prie vis programavimo b d	28
5 lentel . Dispersin s analiz s rezultatai. Programavimo b das: viena klas	31
6 lentel . Dispersin s analiz s rezultatai. Programavimo b das: viena klas	31
7 lentel . Porinio Stjudento testo rezultatai (p=0,05). Programavimo b das: viena klas	33
8 lentel . Dispersin s analiz s rezultatai. Programavimo b das: dvi klas s.....	34
9 lentel . Dispersin s analiz s rezultatai. Programavimo b das: dvi klas s.....	34
10 lentel . Porinio Stjudento testo rezultatai (p=0,05). Programavimo b das: dvi klas s	36
11 lentel . Dispersin s analiz s rezultatai. Programavimo b das: QGLWidget klas	37
12 lentel . Dispersin s analiz s rezultatai. Programavimo b das: QGLWidget klas	38
13 lentel . Porinio Stjudento testo rezultatai (p=0,05). Programavimo b das: QGLWidget klas ...	39
14 lentel . Dispersin s analiz s rezultatai. Programavimo b das: QThreads klas	40
15 lentel . Dispersin s analiz s rezultatai. Programavimo b das: QThreads klas	41
16 lentel . Porinio Stjudento testo rezultatai (p=0,05). Programavimo b das: QThreads klas	42
17 lentel . Porinio Stjudento testo rezultatai (p=0,05). Programavimo b das: QThreads klas	44
18 lentel . Porinio Stjudento testo rezultatai (p=0,05). Programavimo b das: QThreads klas	46

1. VADAS

Aki judesi matavimo sistemos (AJMS) – renginiai, gebantys nustatyti žmogaus žiūros linijos kryptį. Daugelis AJMS yra invazinės (sistemos dalys turi fizinį kontaktą su tiriamuoju), todėl nėra tinkamos plačiam vartojimui.

Šiuo metu vienos populiariausių ir labiausiai vystomos yra nuotolinės video-okulografinės AJMS [13]. Tokios sistemos leidžia tiriamajam atlikti eksperimentus ar vartotojui dirbti beveik natūraliomis sąlygomis, todėl šios AJMS plačiai naudojamos ir aki judesių tyrimuose.

Aki judesių tyrimams dažnai naudojamas tolygia trajektorija judantis taikinys. Taikinio vaizdavimui ir jo judėjimo trajektorijos koordinatų generavimui gali būti pasirinkti vairūs metodai, tačiau ne visi metodai vienodai tiksūs ir ne visi atitinka tyrimo keliamus reikalavimus.

Šiame darbe siekiama išsiaiškinti pasirinktą metodą taikinio judėjimo trajektorijos koordinatų generavimo galimybių ribas, spartinimo galimybes ir metodo tikslumą.

Darbo tikslas ir uždaviniai

Darbo tikslas yra ištirti aki judesių eksperimentinį taikinio generavimo metodus, siekiant padidinti aki judesių tyrimų rezultatų tikslumą ir nustatyti generavimo metodų galimybių ribas. Šiam tikslui pasiekti išskelti uždaviniai:

- Sukurti pasirinktais metodais ir pasirinktais programavimo būdais taikinio generavimo programinį rangą;
- Eksperimentiškai nustatyti kokių metodų koks didžiausias galimas taikinio generavimo greitis.
- Eksperimentiškai nustatyti ar rezultatai priklauso nuo generavimo metodo.
- Eksperimentiškai nustatyti ar rezultatai priklauso nuo programavimo būdo.
- Eksperimentiškai nustatyti standartinio nuokrypio kitimą prieš skirtingą taikinio generavimo metodą ir programavimo būdą.
- Išanalizuoti taikinio generavimo metodų skirtumus su kiekvienu programavimo būdu.
- Išanalizuoti programavimo būdų skirtumus su kiekvienu taikinio generavimo metodu.
- Atlikti generavimo metodų lyginamąjį analizę.

2. TEORIJS APŽVALGA

2.1. Video-okulografin s aki judesi matavimo sistemos

2.1.1. Realizacijos

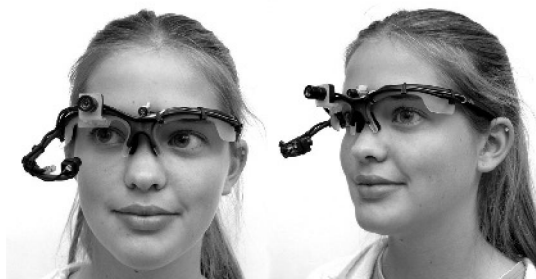
Video-okulografin s aki judesi matavimo sistemos (AJMS) gali b ti:

a) Fiksuotos sistemos – tiriamojo galva yra fiksuojama panaudojant galvos atram arba dant kandikl (1 pav.). Šios sistemos yra labiausiai invazin s (sistemos dalys turi fizin kontakt su tiriamuoju), tod l n ra tinkamos pla iam vartojimui. Ši sistem pagrindinis privalumas – nereikalingi galvos judesi eliminavimo algoritmai.



1 pav. Fiksuota video-okulografin AJMS

b) Šalmin s sistemos – ant tiriamojo galvos užd to šalmo arba akini tvirtinama viena arba kelios filmavimo kameros (2 pav.). Šalmin se aki judesi matavimo sistemose kamera, kuri fiksuoja akies vaizd yra stabili galvos atžvilgiu. Tai leidžia atlikti didel s amplitud s ir didelio grei io galvos judesius, kurie ne takoja tikslumo, nustatant žvilgsnio krypt galvos atžvilgiu. Taciau bendru atveju, norint nustatyti ži ros linijos krypt , turi b ti žinoma 3D (trimat) galvos pad ties erdv je (pozicija ir orientacija), t.y. reikalingi galvos pad ties nustatymo algoritmai [12]. Šios sistemos taip pat turi didel invazijos lyg .



2 pav. Šalmin video-okulografin AJMS

c) Nuotolinės sistemos – sistemos, kurios nereikalauja joki papildomais galvos stabilizacijai ar šalimais (3 pav.) Šios sistemos turi mažiausią invazijos lygį, todėl yra labai paplitusios ir sparčiai vystomos. Pagrindinė nuotolinės sistemos problema – žiros linijos nustatymo tikslumas ir galimas galvos judesio diapazonas. Kuo didesnis leistinas galvos judesio diapazonas, tuo mažesnis tikslumas. Šias sistemas paanalizuosime smulkiau.



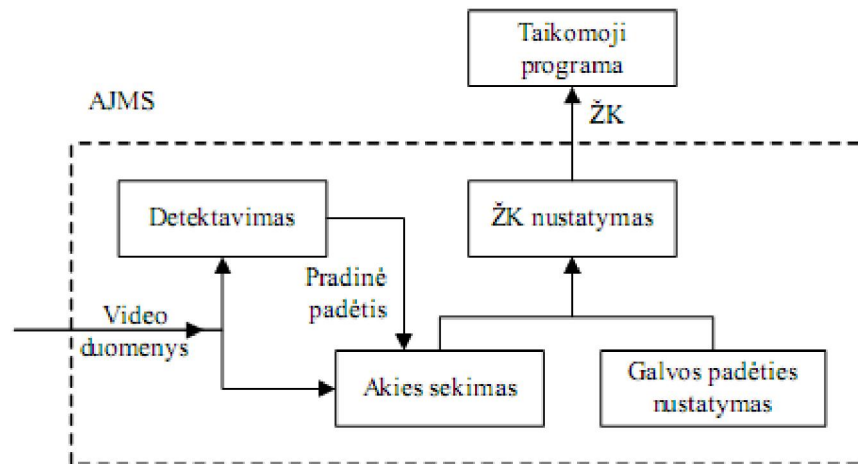
3 pav. Nuotolinė video-okulografinė AJMS

2.1.2. Nuotolinė video-okulografinė AJMS

Nuotolinė video-okulografinė AJMS skiriasi savo realizacija. Galima išskirti dvi nuotolinių video-okulografinių sistemų grupes: pagrįstas infraraudonųjų (IR) spindulių apšvietimu ir pagrįstas natūraliu apšvietimu. IR spinduliais pagrįstas sistemos sudaro IR diodai, apšvieiantys akis, ir IR arba matomos šviesos spektro kameros. Natūraliu apšvietimu pagrįstos sistemos pasikliauja tik aplinkos apšvietimu ir matomos šviesos spektro kameromis. Tokiose sistemose žvilgsnio krypties nustatymui reikalingos sudėtingos vaizdo apdorojimo technologijos. Algoritmai yra sudėtingesni, reikalaujantys didesnio apdorojimo laiko, todėl padidėja reikalavimai techniniam rangui. Kartais tai sprendžiama pasirenkant specializuotą vaizdo apdorojimo techninį rangą, tačiau labai dažnai naudojamos standartinės kameros ir personaliniai kompiuteriai. Šio tipo sistemose žiros linijos nustatymas dažniausiai yra realizuotas kartu su galvos padėties nustatymu. Esant natūraliam apšvietimui rainelės-vyždžio riba yra nekontrastinga, todėl dažniausiai sekamas limbas – riba tarp baltos skleros ir tamsesnės rainelės [12].

IR spindulių nuotolinė video-okulografinė AJMS dažniausiai turi vieną ar dvi kameras, pritvirtintas monitoriaus apačioje. Žmogaus žvilgsnio kalibravimo palengvinimui šios kameros gali būti reguliuojamos, t.y. gali būti pakeltos arba nuleistos, pasuktos vienu arba kitu puse. Kameros centre yra nedidelis, mažos galios infraraudonųjų spindulių diodas, kuris apšviečia akis. Kompiuteryje montuota vaizdo vedimo plokštė.

Pradin nuotolini video-okulografini AJMS informacija yra Video duomenys iš vienos ar kelių kamer (4 pav.). Gavusi šiuos duomenis AJMS nustato pradin akies padėtį (Detektavimo komponentas), kuri toliau naudojama akies padėties sekime kiekvienu laiko momentu (Akies sekimo komponentas). Remiantis iš šio komponento gauta informacija, bei informacija, gauta iš Galvos padėties nustatymo komponento, ŽK nustatymo komponentas nustato vartotojo žiūros linijos kryptį.



4 pav. Nuotolinis video-okulografinis AJMS komponentai

Žinant žiūros linijos kryptį galima nustatyti žiūros linijos tašką, kuris naudojamas vairiems akių judesių tyrimams atlikti ir sudėtingą grafinį aplinką (virtuali aplinką) valdymui.

2.2. Taikinyje video-okulografinis AJMS

Šiaulių universitete, Technologijos fakultete yra sukurtas Biomedicininis inžinerijos mokslinis centras. Centro darbuotojai atlieka tyrimus susijusius su žmogaus – kompiuterio sąveika. Pagrindinis Biomedicininis inžinerijos mokslinio centro tyrimo objektas yra akių judesiai [1]. Centre naudojama moderni LC Technologies sukurta IR spindulių veikimu pagrįsta nuotolinė video-okulografinė žiūros taško registracijos sistema EyeGaze System. Šios sistemos pagalba atlikta daugybė mokslinių tyrimų.

Vienas iš Biomedicininis inžinerijos mokslinio centro darbuotojų tyrimų, susijusių su akių judesiais, yra pasivejančio sakadų tyrimas (angl. catch-up saccade). Tyrimas atliekamas sekant tolygią trajektoriją judant taikinį. Išsiaiškinta, kad kai taikinio greitis pasidaro per didelis sekamiesiems akių judesiams, jie yra pertraukiami greitą žvilgsnio šuolį – pasivejančio sakadą, kurios neleidžia pamesti sekamo taikinio [6].

Kitas šio mokslinio centro tyrimas – žmogaus sensomotorinės sistemos informacinių charakteristikų tyrimas. Vykdamas sekimo procesą tiriamajam pateikiamas taikinys atsitiktine trajektorija judantis ekrane. Ištirta, kad esant tam tikram taikinio greičiui, žmogaus motorinė sistema nebespėja atkartoti taikinio trajektorijos ir sekimo paklaida išaugs [5].

Dar vienas Biomedicininių inžinerijos mokslinio centro darbuotojų tyrimas – ribinis taikinio sekimo greitis žmogaus žvilgsniu ir ranka. Vykdamas sekimo procesą, tiriamajam kompiuterio ekrane pateikiamas taikinys, dvimatis atsitiktinis trajektorija judantis ekrane, kur tiriama turi sekti žvilgsniu arba ranka. Atlikus eksperimentus nustatyta, kad sekant taikinį tiek žvilgsniu, tiek ranka, sekimo paklaidos standartiniai nuokrypiai, didinant taikinio greitį, nuolat didėja, tačiau sekimo žvilgsniu rezultatai išlieka truputį geresni negu sekant taikinį kompiuterio pele [7].

Visiems šiems tyrimams naudojamas judantis taikinys – tam tikro dydžio taškas, judantis nustatyta arba atsitiktine trajektorija. Taikinio vaizdavimui Biomedicininių inžinerijos mokslinio centro darbuotojai naudoja OpenGL metodą. Juo yra aprašoma taikinio išvaizda ir vykdomas taikinio perpiešimas. Taikinio judėjimo trajektorijos koordinatės aprašomos .txt formate dokumente, kuris keliamas programiniame lange ir yra nuskaitomas programai pradėjus darbą. Naudojant laikmačius (angl. timer) sukuriama taikinio judėjimo animacija.

Šiame darbe tiriamos OpenGL metodo taikinio judėjimo trajektorijos koordinatės generavimo galimybių ribos, metodo tikslumas, spartinimo galimybės ir atliekamas lyginimas su kitu galimu taikinio judėjimo trajektorijos koordinatės generavimo metodu.

2.3. Generavimo metodai ir programavimo būdai

2.1.3. QPainter

Qt piešimo sistema vykdo piešimą ir atvaizdavimą ekrane naudodama tą patį programavimo sąsają ir yra pirmiausiai pagrįsta QPainter, QPaintDevice ir QPaintEngine klasėmis (5 pav.). QPainter klasė naudojama piešimo operacijoms vykdyti, QPaintDevice klasė suteikia dviejų dimensijų erdvę, kurioje galima piešti QPainter klase, ir QPaintEngine klasė aprašo sąsają, kurios dėka piešimui gali būti naudojami skirtingi tipai renginių. QPaintEngine klasė yra naudojama viduje QPainter ir QPaintDevice klasių ir yra nematoma programuotojams.



5 pav. Qt piešimo sistema

QPainter klasė atlieka žemo lygio piešimą paprogramse (angl. widgets) ir kituose piešimo rankiuose. QPainter klasė suteikia optimizuotas funkcijas, kurių pagalba sukuriami patogūs grafini vartotojo sąsaja bei galimos didesnės piešimo galimybės. Jomis galima nupiešti viską, pradedant paprastomis linijomis ir baigiant sudėtingomis figūromis, skritulinėmis diagramomis. QPainter klase piešiama dviejų dimensijų erdvėje, taip pat ją galima atlikti vaizdo transformacijas, grafines operacijas.

Pagrindinis privalumas šio metodo yra tai, kad visas piešimas vyksta nuosekliai ir galima jį papildyti naujomis funkcijomis.

2.1.4. OpenGL

Šiuo metu viena populiariausių žemo (detalaus) lygio trimatnio grafinio programavimo taikomųjų vartotojo sąsajų (angl. API) yra OpenGL. OpenGL iš esmės nėra apribojimų. Dabartiniu metu ji pritaikyta visoms Windows versijoms (skaitant ir senąją Windows NT 4.0), Unix sistemoms, Mac OS, Playstation pultams (angl. console) ir t.t.

OpenGL yra funkcinio programavimo sąsaja, dar vadinama bsenos mašina (angl. state machine).

OpenGL realizuota C, Java, Delphi, Python ir daugeliu kitų populiariausių programavimo kalbų.

OpenGL s kmingai taikoma kompiuteriniuose žaidimuose, CAD sistemose, moksliniuose darbuose ir t.t.

OpenGL s saja pasižymi keletu išskirtini ir svarbi savybi :

1. B sena yra išsaugoma (angl. state machine);
2. Laisvai pasirenkama modeliavimo erdv ;
3. Bazin OpenGL dalis neturi joki nuo operacin s sistemos priklausanti funkcij (lang tvarkymo, darbo su vedimo/išvedimo renginiais ir kt.);
4. Formuojamiems daugiakampiems taikomi tam tikri apribojimai;
5. N ra elementari dvima i primitiv ;
6. Vaizdavimas yra tiesioginis (angl. immediate).

B senos išsaugojimas yra viena iš pagrindini OpenGL savybi . Bet ji b dinga ne visoms realizacijoms. Min tos savyb s esm tokia – tam tikri parametrai ar b senos taikomos vaizduojamiems objektams tol, kol nepakei iamos ar nepanaikinamos. Tai nuoseklus programos vykdymo b das, t.y. proced r kvietimo eiliškumas yra svarbus ir j pakeitus, dažniausiai pasikei ia programos funkcionalumas. Keletas elementari j b sen pavyzdži yra spalva, transformacija, normal s vektorius ir t.t.

OpenGL modeliavimo erdv s matmenys yra laisvai pasirenkami, santykiniai ir nepriklausomi nuo matavimo vienet . Svarbiausias dalykas – visi vaizduojami objektai turi išlikti pasirinkto ribojan io t rio viduje. Be to, reikia vertinti galimas trima i objekt transformacijas, pvz. animacijos metu.

Pradedant darb tiek steb tojas, tiek visi objektai yra koordinati pradžioje. Tod l turi b ti perkelta visa trimat scena arba steb tojas. Abiem atvejais rezultatas yra tas pats, bet jei neatliktas nei vieni iš ši veism , trimat scena bus per daug arti steb tojo, $z = 0$ plokštum kertantys objektai grei iausiai užstos vis vaizd .

OpenGL visi objektai yra trima iai, net jei jie sukuriamai dvimate koordinatės nusakan iu metodu.

Bendruoju atveju OpenGL modeliavimo erdv yra homogenin . J sudaro keturios koordinat s: x, y, z, w . Norint gauti atitinkamas koordinatės trimat je erdv je reikia atlikti tokius perskai iavimus: $x = x / w$; $y = y / w$, $z = z / w$.

Grafine sistemoje galimi du vaizdavimo b dai: tiesioginis (angl. immediate) ir buferizuotasis. Taikom sias programas OpenGL galima realizuoti tiek vienu, tiek kitu, tiek mišriuotu b dais. Kiekviena OpenGL funkcija vykdoma iš karto, nenaudojant speciali j strukt r . Jei tai vaizdavimo funkcija, vaizdas ekrane taip pat pakei iamas iš karto.

2.1.5. QGLWidget

QGLWidget klasė skirta atvaizduoti OpenGL grafiką integruotą Qt programoje. QGLWidget klasė naudojama kaip bet kuri kita paprogramė (angl. widget), tik suteikia galimybę rinktis QPainter arba OpenGL atvaizdavimo komandas.

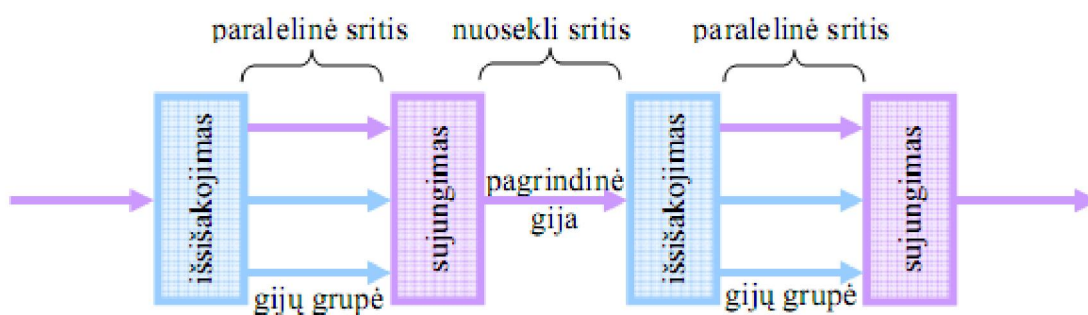
QGLWidget numatytos trys virtualios funkcijos, skirtos vykdyti OpenGL užduotis:

- paintGL() – piešimo funkcija. Kreipiamasi, kai paprogramė turi būti atnaujinta.
- resizeGL() – nustato OpenGL projekciją ir pan. Kreipiamasi, kai paprogramėje buvo pakeistas projekcijos dydis (taip pat kada rodoma pirmą kartą).
- initializeGL – GL buferių inicializavimas. Nustato OpenGL vaizdavimo kontekstą, nurodo atvaizdavimo sąrašą. Iššaukiama prieš paintGL() ir resizeGL().

QGLWidget klasėje galima naudoti laikmačius (angl. timer) ir jų pagalba kurti judančius objektus, scenas. Taip pat galima dalintis OpenGL vaizdavimo sąrašais tarp QGLWidget objektų.

2.1.6. QThreads

Įėjimo pagalba galima paskirstyti programos operacijas ir jas vykdyti vienu metu. Įėjimo taikymas prasmingas tada, kai darbai padalinami ir vykdomi vienu metu. Jei darbai priklauso vienas nuo kito, t.y. turi būti vykdomi iš eilės, įėjimo taikymas nėra efektyvus.



6 pav. Gijų veikimas

Programa prasideda kaip vienas procesas: pagrindinė gija vykdo kodą, kol sutinkama paralelinė sritis direktyva (6 pav.). Pagrindinė gija sukuria gijų grupę paralelinės srities paleidimo metu (išsišakojimas). Programos sakiniai, kurie priklauso paralelinės srities direktyvai, yra vykdomi gijų grupėje lygiagrečiai. Kai gijų grupė užbaigia vykdyti sakinius paralelinės srities direktyvoje, gijos yra sinchronizuojamos ir nutraukiamos, paliekant tik pagrindinę giją (sujungimas).

QThreads klasė skirta kurti gijas, nepriklausomas nuo platformos. QThreads klasė sukuria atskirą kontrolinę giją programos viduje. Ši gija dalinasi duomenimis su kitomis gijomis, tačiau tai daro nepriklausomai.

Kiekviena QThreads klasė gali turėti savo vykių ciklą. vykių ciklą galima iškviešti arba sustabdyti. Turint vykių ciklą gijoje, ji galima traukti signalus iš kitų gijų naudojant eilės jungties mechanizmą. Tai taip pat leidžia naudoti klases, kurioms reikalingas vykių ciklas gijoje.

Ekstremaliose situacijose gali būti pageidaujama priverstinai atjungti veikiančią giją [10].

2.1.7. Klasė

Klasė – tam tikras objekto tipas. Klasės aprašyme yra jos būsenos duomenys ir protokolai rinkinys, kuriais yra suveikaujama su klasės objektais. Klasė yra suorganizuojamas hierarchinis struktūra ir poklasė (klasė einanti jos apimančiame klasių paveldi savybes jas sukursi klasių). Klasė susideda iš tokių elementų:

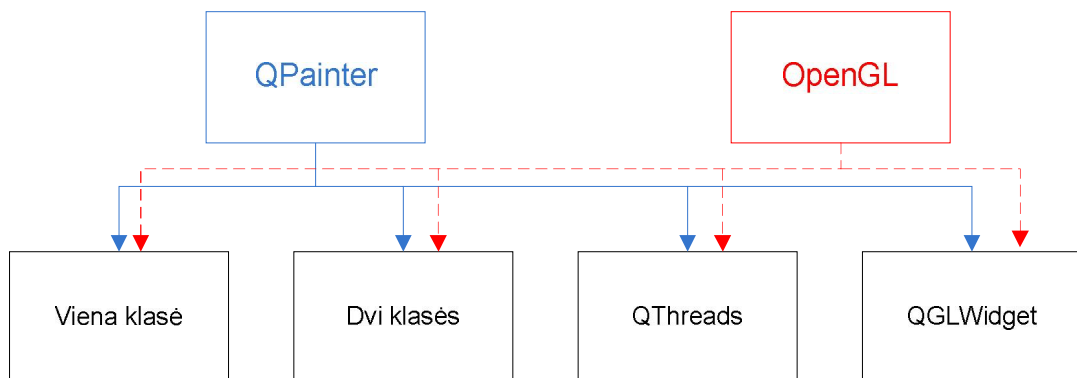
- Duomenų laukai. Kintamos būsenos, apibūdinančios klasės objektą, yra deklaruojamos klasės viduje. Visi tos pačios klasės objekto egzemplioriai turi tą pačią duomenų aibę, nors atskiri egzemplioriai būsenos skiriasi.
- Funkcijos – dalyviai (metodai). Kiekvienas klasės objekto apdorojamas pranešimas turi būti aprašytas funkcija – dalyviu. Funkcija turi būti paskelbta (deklaruota) klasės aprašyme. Detalus klasės aprašymas gali būti klasės aprašyme arba už jo (net ir kitame faile). Funkcijos deklaracijoje parametrai gali būti nurodomi tik tipais arba vardais. Funkcijos aprašyme parametrai turi būti nurodyti su tipais ir vardais.

Poklasė paveldi savybes iš jį apimančios klasės ir prie jų dažnai prideda savo savybių. Tokia paveldimumo savybė reiškia, kad žemesnio lygio objektai yra labiau specializuoti, o aukštesnio lygio objektai yra bendresni. Paveldimumo savybė leidžia sudaryti hierarchines klasių struktūras. Tam tikra klasė gali būti paskelbta kaip kitos (bazinės) klasės išvestinė klasė. Daugialypiškumo palaikymas reikalauja, kad būtų galimybė sukurti išvestinę klasę daugiau nei iš vienos bazinės klasės.

3. PROGRAMIN RANGA

Taikinio atvaizdavimui ir jo judimo trajektorijos koordinatų generavimui pasirinkti QPainter ir OpenGL metodai bei keturi programavimo būdai:

1. Viena klasė – taikinio piešimas ir jo judimo trajektorijos koordinatų generavimas vykdomas pagrindinėje programoje;
2. Dvi klasės – taikinio piešimas vykdomas atskiroje klasėje, o jo judimo trajektorijos koordinatų generavimas pagrindinėje programoje.
3. QThreads – taikinio piešimas vykdomas QThreads klasėje, o jo judimo trajektorijos koordinatų generavimas pagrindinėje programoje.
4. QGLWidget – taikinio piešimas vykdomas QGLWidget klasėje, o jo judimo trajektorijos koordinatų generavimas pagrindinėje programoje.



7 pav. Metodai ir programavimo būdai

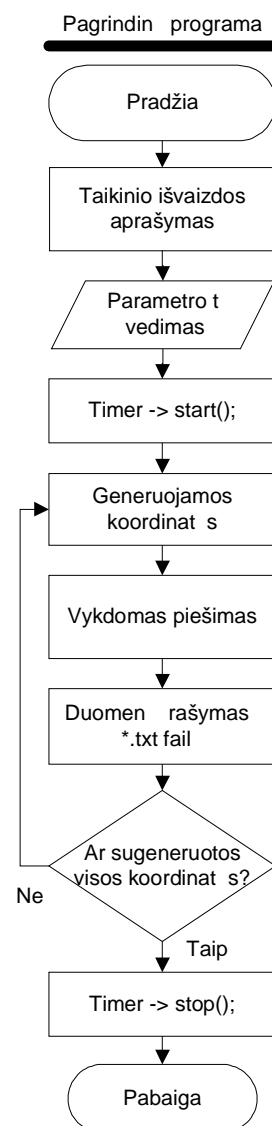
Šiais metodais ir programavimo būdais sukurta programinė ranga – viso 7-ios atskiros programos (7 pav.):

1. QPainter metodas + vienos klasės programavimo būdas;
2. QPainter metodas + dviejų klasių programavimo būdas ir OpenGL metodas + dviejų klasių programavimo būdas;
3. QPainter metodas + QGLWidget klasė ;
4. QPainter metodas + QThreads klasė ;
5. OpenGL metodas + vienos klasės programavimo būdas;
6. OpenGL metodas + QGLWidget klasė ;
7. OpenGL metodas + QThreads klasė ;

3.1. Viena klas

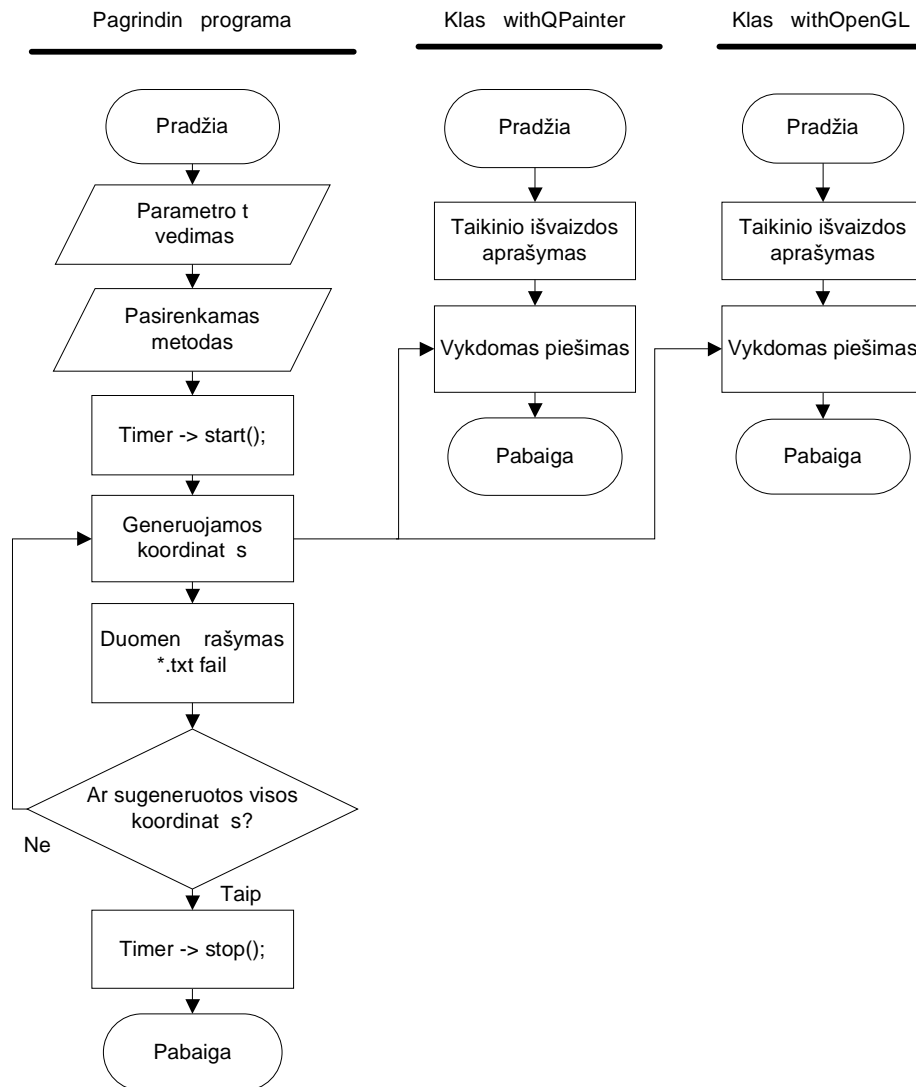
1-asis algoritmas (8 pav.) vaizduoja taikinio piešimą ir jo judimo trajektorijos koordinatų generavimą vienoje klasėje. Pirmiausia yra aprašyta judimo taikinio išvaizda. Programai pradėjus darbą vedamas parametras t , nurodantis kas kiek laiko bus sugeneruota nauja taikinio judimo trajektorijos koordinatė. Galimi variantai 1ms, 3ms, 5ms, 8ms, 10ms, 13ms, 15ms, 18ms arba 20ms. Tuomet startuoja laikmatis (angl. timer) ir kas nurodytą laiko tarpą yra generuojama nauja taikinio judimo trajektorijos koordinatė. Kiekvienoje koordinatėje piešiamas taikinyš. .txt formato dokumente rašomas naudojamas taikinio generavimo metodas, parametras, nurodantis kas kiek laiko generuojama nauja koordinatė, realus laikas, paverstas milisekundėmis, nurodantis kada buvo sugeneruota koordinatė, koordinatė pagal x ašį ir koordinatė pagal y ašį. Tuomet vyksta užklauskimas ar sugeneruotos visos koordinatės ir jei taip – stabdomas laikmatis (angl. timer) ir programa baigia darbą, o jei ne – toliau generuojamos koordinatės ir piešiamas taikinyš.

8 pav. 1 algoritmas. Taikinio piešimas ir koordinatų generavimas vienoje klasėje



3.2. Dvi klasės

2-asis algoritmas (9 pav.) vaizduoja taikinio piešimą ir jo judimo trajektorijos koordinatų generavimą atskirose klasėse. Kaip ir 1-ajame algoritme startavus programai vedamas parametras t , nurodantis kas kiek laiko bus generuojama nauja taikinio judimo trajektorijos koordinatė. Pasirenkamas taikinio generavimo metodas – QPainter arba OpenGL. Startuoja laikmatis (angl. timer) ir kas nurodytą laiko tarpą yra generuojama nauja taikinio judimo trajektorijos koordinatė.

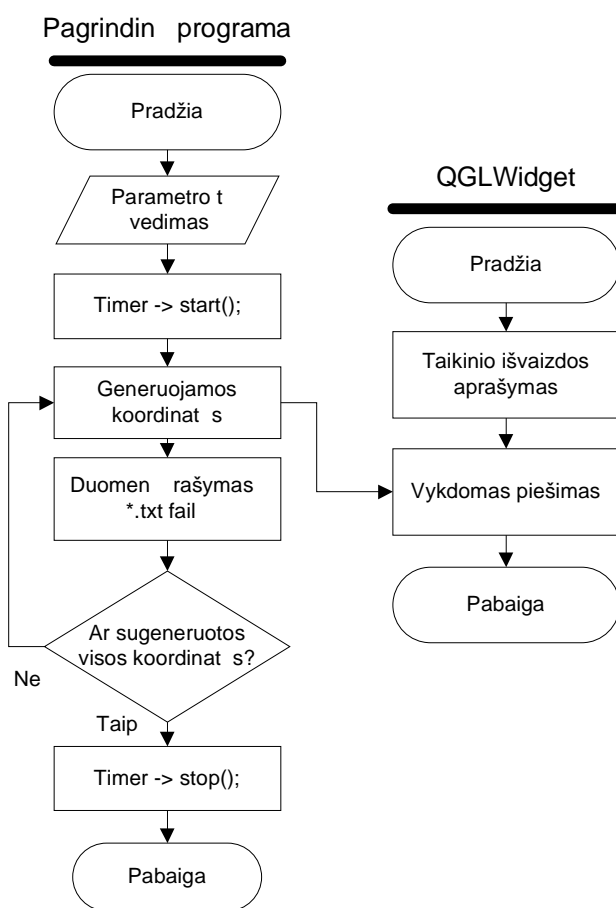


9 pav. 2 algoritmas. Taikinio piešimas ir koordinatų generavimas atskirose klasėse

Tuo pačiu metu kitoje klasėje vykdomas piešimas. .txt formato dokumentas rašomas naudojamas taikinio generavimo metodus, parametras, nurodantis kas kiek laiko generuojama nauja koordinatė, realus laikas, paverstas milisekundėmis, nurodantis kada buvo sugeneruota koordinatė, koordinatė pagal x ašį ir koordinatė pagal y ašį. Vykdomas užklauskimas ar sugeneruotos visos koordinatės. Jei atsakymas neigiamas – programa darbuotės, jei teigiamas – laikmatis (angl. timer) stabdomas ir programa darbu baigia.

3.3. QGLWidget klasė

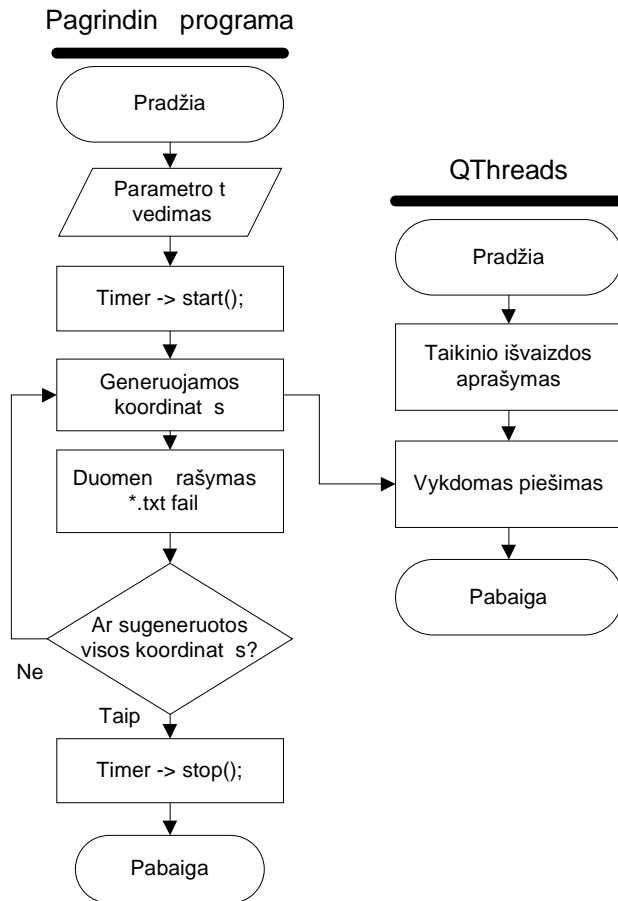
Programa, kurioje taikinio judėjimo koordinatės generuojamos pagrindinėje programoje, o piešimas vykdomas QGLWidget klasėje, vaizduoja 3-iasis algoritmas (10 pav.). Čia, kaip ir ankstesnėse programose, programai pradėjus darbą vedamas parametras t , nusakantis, kas kiek laiko bus generuojama nauja taikinio judėjimo trajektorijos koordinatė. Startuoja laikmatis (angl. timer), generuojamos koordinatės ir tuo pat metu QGLWidget klasėje vykdomas taikinio piešimas. .txt formato dokumente rašomi duomenys, analogiškai pirmoms programoms. Vykdomas užklausimas ar sugeneruotos visos koordinatės. Jei atsakymas neigiamas – programa darbuotės, jei teigiamas – laikmatis (angl. timer) stabdomas ir programa baigia darbą.



10 pav. 3 algoritmas. Taikinio piešimas QGLWidget klasėje

3.4. QThreads klasė

Programa su QThreads klase yra analogiška programai su QGLWidget klase, tik čia gij pagalba taikinio judėjimo koordinatė generavimas (pagrindinė programa) ir taikinio piešimas (QThreads klasė) yra lygiagrečiai vykdomi vienu metu (4 algoritmas) (11 pav.).

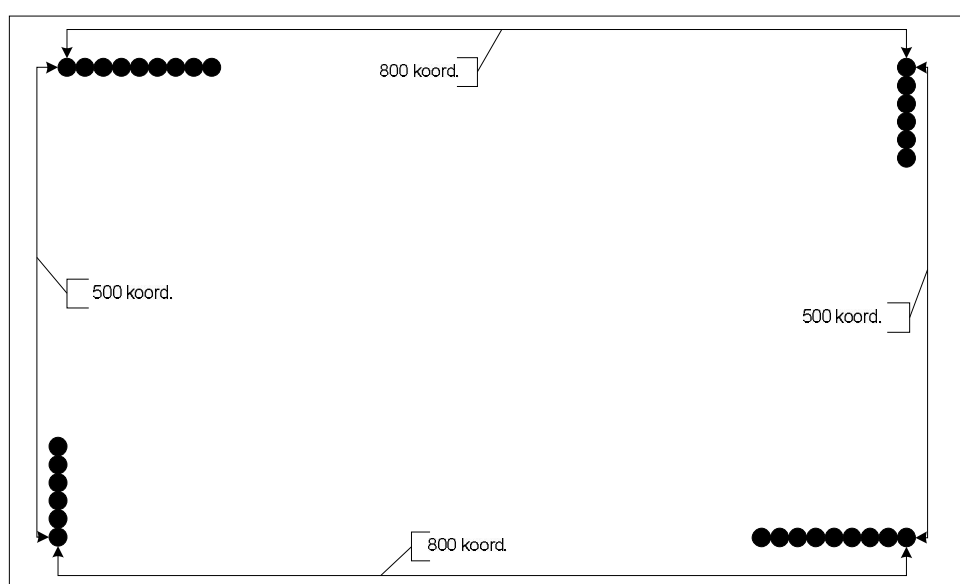


11 pav. 4 algoritmas. Taikinio piešimas QThreads klas je

4. TYRIMO METODIKA

4.1. Taikinio judėjimo trajektorija

Sukurta programine ranga kas 1ms, 3ms, 5ms, 8ms, 10ms, 13ms, 15ms, 18ms ir 20ms generuojamos taikinio judėjimo trajektorijos koordinatės. Tyrimui generuojama 2600 koordinatės: 1600 koordinatės x ašimi ir 1000 koordinatės y ašimi. Taikinyje juda stačiakampė trajektorija (12 pav).



12 pav. Taikinio judėjimo trajektorija

Programos vykdymo metu, kaip jau minėta anksčiau, .txt formate dokumente išsaugomi šie duomenys:

- Ø taikinio judėjimo trajektorijos koordinatės generavimo metodas;
- Ø parametras t , nurodantis kas kiek laiko bus generuojama nauja koordinatė;
- Ø realus laikas, paverstas milisekundėmis, kada buvo sugeneruota koordinatė;
- Ø koordinatė pagal x ašį;
- Ø koordinatė pagal y ašį.

Šie duomenys nuskaitomi ir apdorojami MATLAB programa. Skaičiuojama vieno faktoriaus (vienmatė) dispersinis analizė ir atliekamas Stjudento testas.

4.2. Dispersin analiz

Matuojamas dydis gali priklausyti nuo faktorių, kuri negalima vertinti skaičiumi. Toks faktorius gali būti, pavyzdžiui, mėnuo: sausis, vasaris ir t.t.; asmuo: Tomas, Rokas ir pan.

Vieno faktoriaus dispersin analiz — tai Studento t-kriterijaus apibendrinimas keletui nepriklausom imi. Vieno faktoriaus dispersin analiz (angl. One-Way ANOVA) taikoma tada, kai populiacijos viena nuo kitos skiriamos tik pagal vien požym. Kategorinis kintamasis (populiacijos požymis), pagal kur populiacijos skiriamos viena nuo kitos, vadinamas nepriklausomu kintamuoju arba faktoriumi. Priklausomo kintamojo s voka vartojama vardyti matuojamam parametrai, kuris priklauso nuo veikianio faktoriaus (nepriklausomo kintamojo). Dispersin s analiz s tikslas — nuspr sti, ar priklausomo kintamojo, išmatuoto skirtingose populiacijose, vidurkiai statistiškai reikšmingai skiriasi. Metodas leidžia atsakyti klausim , ar iš keli nepriklausom imi bent dviej vidurkiai statistiškai reikšmingai skiriasi.

Vieno faktoriaus dispersin analiz , kuriai žym ti vartojama santrumpa ANOVA (angl. – ANalysis Of VAriance) grindžiama dviej dispersijos ver i palyginimu. Visa kvadrat suma SST (angl. sum of squares total), kuri apima vis duomen ir bendrojo vidurkio skirtumus, yra suskaidoma dviej komponenti — vidin s kvadrat sumos SSW (angl. sum of squares within), vertinan ios kiekvienos imties duomen sklaid (apie imties vidurk) ir grupi kvadrat sumos SSB (angl. sum of squares between), vertinan ios imi vidurki ir bendrojo vidurkio skirtumus, sum

$$SST = SSW + SSB,$$

ia

$$SST = \sum_{i=1}^k \sum_{j=1}^{n_i} (X_{ij} - \bar{X})^2, \quad SSW = \sum_{i=1}^k \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2,$$

$$SSB = \sum_{i=1}^k n_i (\bar{X}_i - \bar{X})^2,$$

X_{ij} — i-osios imties j-asis steb jimas, \bar{X} — bendrasis vis imi vidurkis,

\bar{X}_i — i-osios imties vidurkis, k — nepriklausom populacij skaičius,

n_i — i-osios imties didumas;

ANOVA hipotezei tikrinti yra sudaroma statistika tarpgrupin s ir grupi vidin s dispersijos pagrindu.

Tikrinama statistin hipotez :

$$\begin{cases} H_0 : \mu_1 = \mu_2 = \dots = \mu_k, \\ H_1 : \text{bent du vidurkiai skiriasi,} \end{cases}$$

ia μ_j — nepriklausom kintam j vidurkiai.

Sprendimo pri mimo taisykl :

H_0 atmetama (ne visi vidurkiai lyg s), jeigu $p < \alpha$;

H_0 neatmetama (vidurki skirtumo nerasta), jeigu $p \geq \alpha$;

ia α — nustatytasis reikšmingumo lygmuo.

Nusta ius, kad yra statistiškai reikšmingai besiskirian i im i vidurki , galima nustatyti, kuri im i vidurkiai statistiškai reikšmingai skiriasi. Tam yra naudojami vadinamieji daugkartini lyginim kriterijai. Kad b t galima patikimai patvirtinti arba atmesti nulini hipotez apie priklausomo kintamojo, stebimo skirtingose populiacijose, vidurki lygyb taikant vieno faktoriaus dispersin s analiz s metod , turi b ti vykdytos šios s lygos:

- Kintamieji $X_1 \dots X_k$ populiacijose turi b ti pasiskirst pagal normal j d sn . Taciau, jeigu imtys yra pakankamai didel s arba j didumai mažai skiriasi, nedideli skirstini nuokrypiai nuo normaliojo n ra kritiški.
- Kintamieji $X_1 \dots X_k$ turi b ti nepriklausomi.
- Faktorinis (nepriklausomas) kintamasis turi b ti sveikas skai ius, o priklausomas kintamasis — intervalinis.
- Kintam j $X_1 \dots X_k$ dispersijos turi b ti lygios — turi b ti tenkinama dispersij homogeniškumo s lyga. Kuo mažiau skiriasi im i didumas, tuo mažiau ANOVA jautri populiacij dispersij skirtumams. Paprastai didžiausioji iš im i dispersij neturi b ti didesn už mažiausi j daugiau kaip tris kartus.

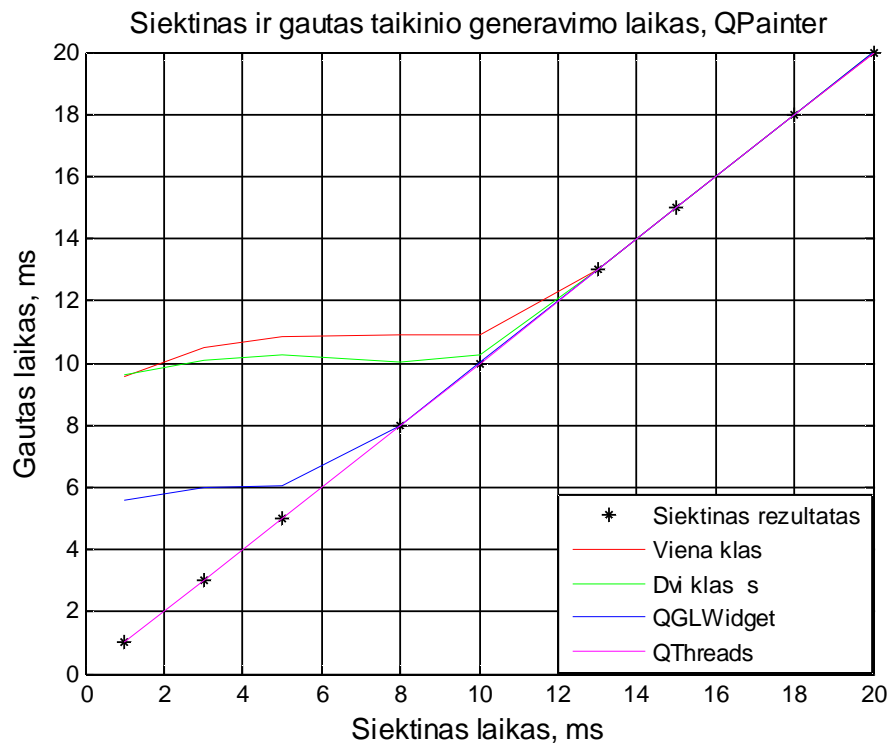
5. TYRIMO REZULTATAI

5.1. Programavimo būd palyginimas prie kiekvieno generavimo metodo

Šiame tyrime naudojami 4 programavimo būdai: viena klasė, dvi klasės, QGLWidget klasė ir QThreads klasė. Programavimo būdų tikslas, taikinio judėjimo trajektorijos koordinatų generavimo metodams QPainter ir OpenGL, tiriama šiame skyriuje.

5.1.1. QPainter metodas

13 paveiksle pavaizduotas siektinas ir gautas rezultatas kiekvieno programavimo būdo, generuojant taikinį QPainter metodu. Rezultatai rodo, jog programavimo būdas daro tikslų taikinio judėjimo trajektorijos koordinatų generavimui.



13 pav. Siektinas ir gautas taikinio generavimo laikas QPainter metodu

Programavimo būd palyginimui atlikta vienmatė dispersinė analizė (Anova). Analizės rezultatai pateikti 1 lentelėje. Lyginti visi 4 programavimo būdai prie kiekvieno taikinio judėjimo trajektorijos koordinatų generavimo laiko. Programavimo būdų numeracija atitinka metodinę

dalyje išvardint tvark , t.y. 1 programavimo b das – viena klas ; 2 programavimo b das – dvi klas s; 3 programavimo b das – QWidget klas ; 4 programavimo b das – QThreads klas . Gautas reikšmingas programavimo b d skirtumas kai generavimo laikas yra 10ms ir mažesnis. Kai generavimo laikas 13ms ir didesnis – programavimo b das nereikšmingas.

Taikinio generavimas QPainter metodu			
Programavimo b dai	Laikas	F	Tikimyb
1,2,3,4	20 ms	0	1
1,2,3,4	18 ms	0	1
1,2,3,4	15 ms	0	0,9999
1,2,3,4	13 ms	0,01	0,999
1,2,3,4	10 ms	142,18	0
1,2,3,4	8 ms	11,14	2,70295E-07
1,2,3,4	5 ms	220,45	0
1,2,3,4	3 ms	321,08	0
1,2,3,4	1 ms	162,76	0

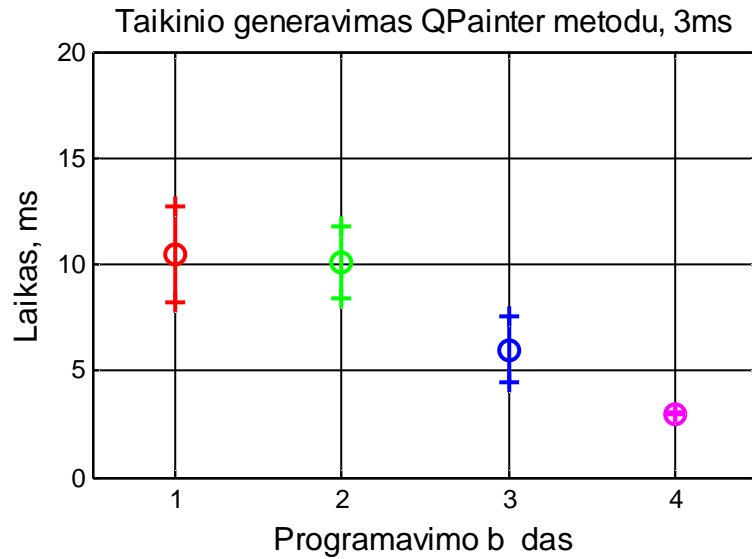
1 lentel . QPainter gautas ir siektinas laikas prie vis programavimo b d

Tikslesn s programavimo b d reikšmingumo ribos nustatymui papildomai pakartoti eksperimentai su 11ms ir 12ms generavimo laikais. Rezultatai pateikti 2 lentel je. Matome, kad visgi programavimo b das prie 11ms ir 12ms generavimo laiko yra nereikšmingas. Galima daryti išvad , jog programavimo b das, kai naudojamas QPainter generavimo metodas, yra reikšmingas tuomet, kai taikinio jud jimo trajektorijos koordinat s generavimo laikas yra 10ms ir mažesnis.

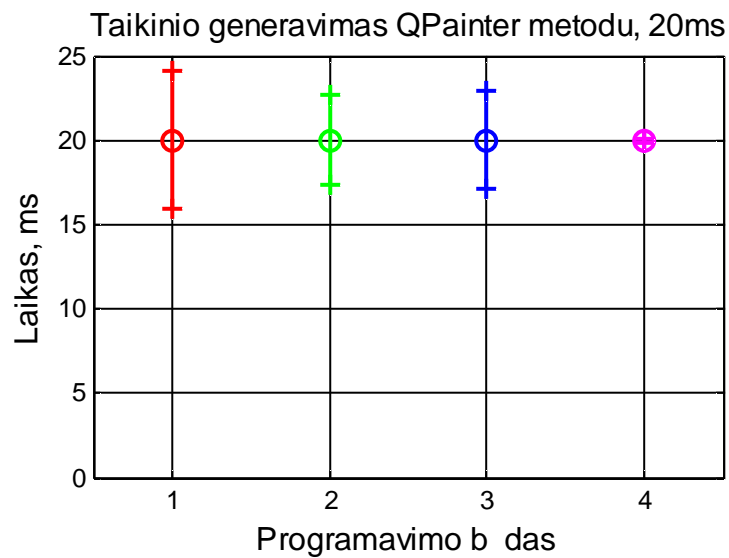
Taikinio generavimas QPainter metodu			
Programavimo b dai	Laikas	F	Tikimyb
1,2,3,4	12 ms	0,62	0,6018
1,2,3,4	11 ms	0,53	0,6613

2 lentel . QPainter gautas ir siektinas laikas prie vis programavimo b d

14-ame ir 15-tame paveiksluose pateikti programavimo b d rezultatai. Programavimo b d numeracija atitinka 1 lentel s numeracij . Paveikslai vaizdžiai parodo programavimo b d skirtumus kai programavimo b das yra reikšmingas (14 pav.) ir kai nereikšmingas (15 pav). Didžiausi tak rezultatams programavimo b das daro tuomet, kai generavimo laikas yra 3ms (14 pav.), o takos nedaro, kai generavimo laikas yra, pavyzdžiui, 20ms (15 pav.).



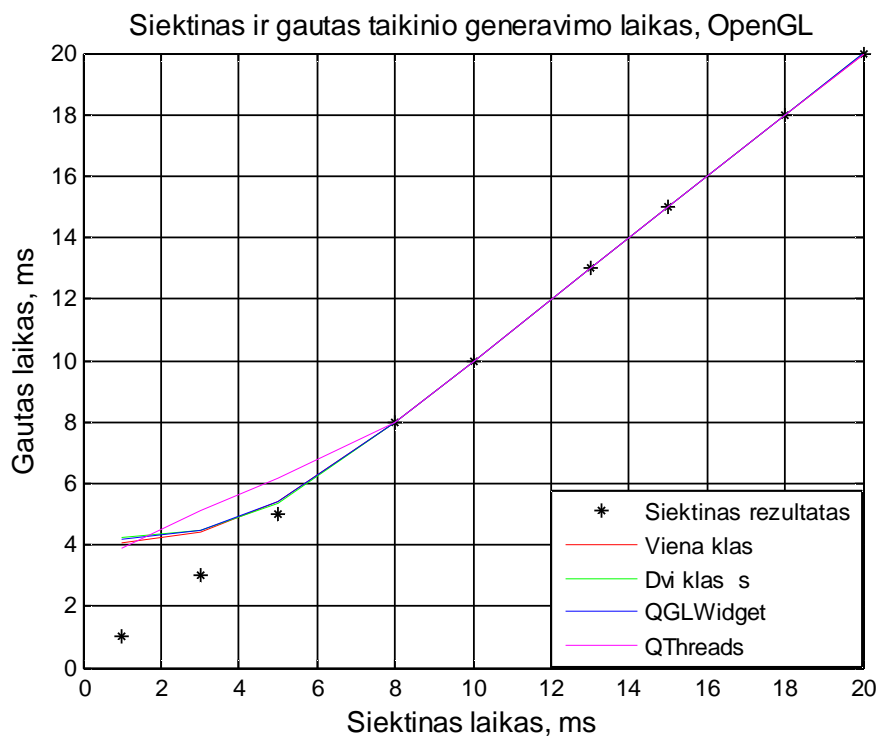
14 pav. Programavimo b d palyginimas. Generavimo metodas: QPainter "o" – vidurkis, "+" – vidurkis \pm standartinis nuokrypis



15 pav. Programavimo b d palyginimas. Generavimo metodas: QPainter "o" – vidurkis, "+" – vidurkis \pm standartinis nuokrypis

5.1.2. OpenGL metodas

16 paveiksle pavaizduotas siektinas ir gautas rezultatas kiekvieno programavimo b do, generuojant taikinio jud jimo trajektorijos koordinates OpenGL metodu. Iš paveikslo matyti, jog gauti norim rezultat OpenGL metodu nepavyko.



16 pav. Siektinai ir gautas taikinio generavimo laikas OpenGL metodu

Iš pirmo žvilgsnio atrodo, jog programavimo b d as taikinio jud jimo trajektorijos koordinat i generavimui, kai naudojamas OpenGL metodas, didel s takos neturi. Taciau tolimesni tyrimai rodo, jog programavimo b d as rezultatus takoja.

Atlikus vienmat dispersin analiz su visais 4 programavimo b dais (programavimo b d numeracija atitinka 1 lentel s numeracij) paaišk jo, jog programavimo b d skirtumas yra reikšmingas kai generavimo laikas yra 5ms ir mažesnis. Kai generavimo laikas 8ms ir didesnis – programavimo b d as nereikšmingas.

Taikinio generavimas OpenGL metodu			
Programavimo b dai	Laikas	F	Tikimyb
1,2,3,4	20 ms	0	1
1,2,3,4	18 ms	0,95	0,4169
1,2,3,4	15 ms	0,98	0,4008
1,2,3,4	13 ms	0	0,9996
1,2,3,4	10 ms	0,99	0,3981
1,2,3,4	8 ms	0,08	0,9695
1,2,3,4	5 ms	8,43	1,35466E-05
1,2,3,4	3 ms	7,59	4,54764E-05
1,2,3,4	1 ms	1,07	0,3593

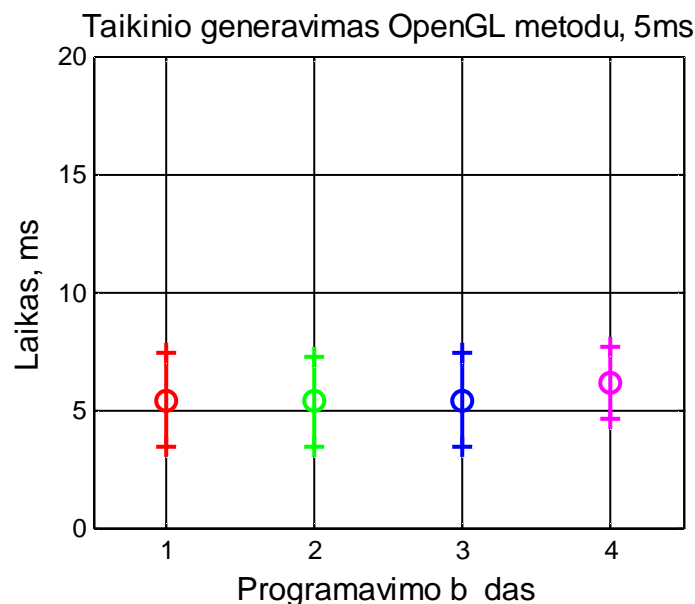
3 lentel . OpenGL gautas ir siektinai laikas prie vis programavimo b d

Tikslesnis programavimo būdų reikšmingumo ribos nustatymui papildomai pakartoti eksperimentai su 6ms ir 7ms generavimo laikais. Rezultatai pateikti 4 lentelėje. Matome, kad programavimo būdas, kai generuojamos koordinatės kas 6ms ir 7ms, yra reikšmingas. Galima daryti išvadą, jog programavimo būdas, kai naudojamas OpenGL generavimo metodas, yra reikšmingas tuomet, kai taikinio judėjimo trajektorijos koordinatės generavimo laikas yra 7ms ir mažesnis.

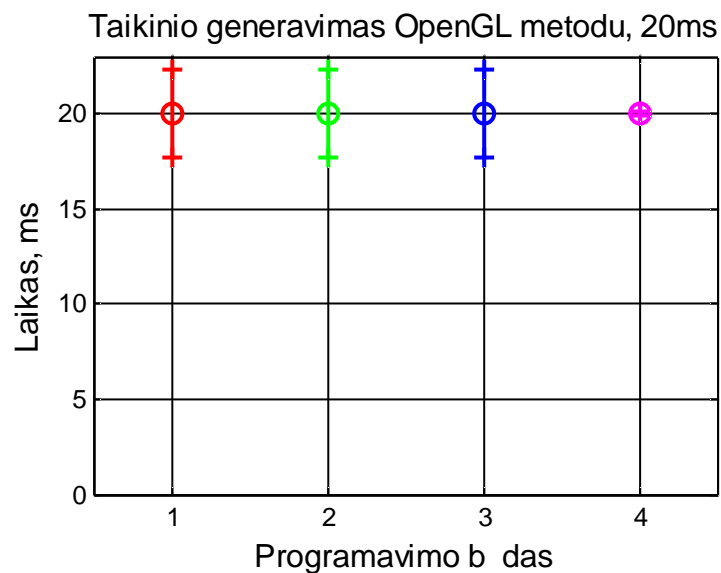
Taikinio generavimas OpenGL metodu			
Programavimo būdai	Laikas	F	Tikimybė
1,2,3,4	7 ms	7,05	0
1,2,3,4	6 ms	5,26	0

4 lentelė. OpenGL gautas ir siektinas laikas prie visų programavimo būdų

17 ir 18 paveikslai vaizdžiai parodo programavimo būdų skirtumus kai programavimo būdas yra reikšmingas (17 pav.) ir kai nereikšmingas (18 pav.). Didžiausi tokie rezultatai programavimo būdas daro tuomet, kai generavimo laikas yra 5ms (17 pav.), o tokos nedaro kai generavimo laikas yra, pavyzdžiui, 20ms (18 pav.).



17 pav. Programavimo būdų palyginimas. Generavimo metodas: OpenGL "o" – vidurkis, "+" – vidurkis \pm standartinis nuokrypis



18 pav. Programavimo b d palyginimas. Generavimo metodas: OpenGL “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis

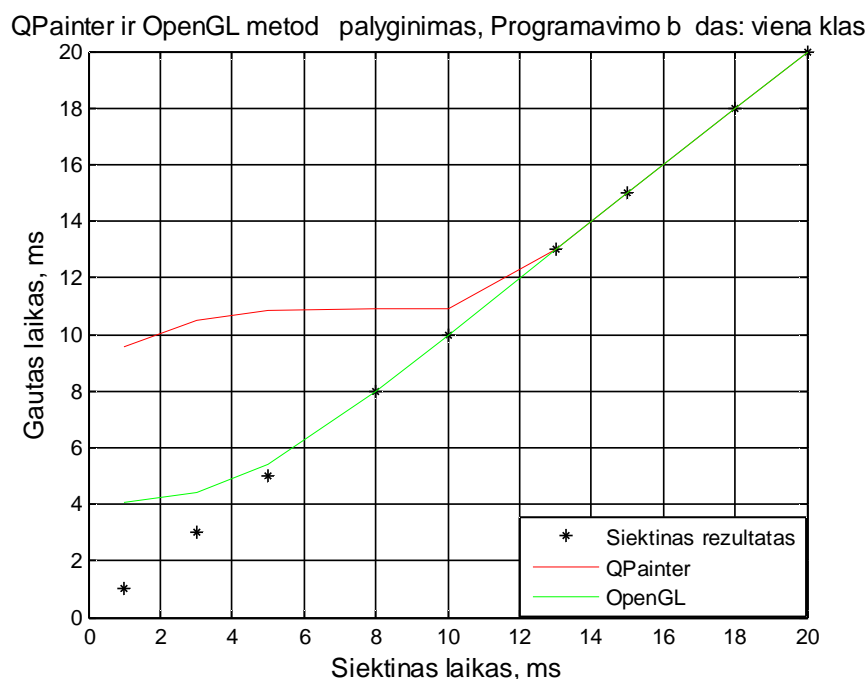
Remiantis šio tyrimo duomenimis galima teigti, jog programavimo b d as taikinio jud jimo trajektorijos generavimui yra reikšmingas, kadangi visais programavimo b dais gauti rezultatai yra skirtingi. Artimiausi rezultat siektinam pavyko gauti QPainter metodu, naudojant QThreads klas s programavimo b d .

5.2. QPainter ir OpenGL metodų palyginimas prie kiekvieno programavimo budo

Taikinio judėjimo trajektorijos koordinatų generavimas šiame tyrime atliekamas dviem metodais: QPainter ir OpenGL. Generavimo metodų skirtumai prie kiekvieno programavimo budo tiriami šiame skyriuje.

5.2.1. Viena klasė

Lyginant QPainter ir OpenGL metodus pagal siektiną ir gautą rezultatą, kai naudojamas vienos klasės programavimas, iš 19 paveikslo matyti, kad OpenGL metodu gauti rezultatai yra artimesni siektinam, nei gauti QPainter metodu.



19 pav. QPainter ir OpenGL gautas ir siektinas laikas. Programavimo budo: viena klasė

Metodų palyginimui atlikta vienmatė dispersinė analizė (Anova). Analizės rezultatai pateikti 5 lentelėje. Lyginti abu metodai prie kiekvieno taikinio judėjimo trajektorijos koordinatų generavimo laiko. Gautas reikšmingas skirtumas kai generavimo laikas yra 10ms ir mažesnis. Kai generavimo laikas 13ms generavimo metodas tampa nebereikšmingu.

Taikinio generavimas QPainter ir OpenGL metodais			
Programavimo b das	Laikas	F	Tikimyb
Viena klas	20 ms	6,95215E-05	0,9933
	18 ms	0	0,9851
	15 ms	0	0,9835
	13 ms	0,02	0,89
	10 ms	315,62	0
	8 ms	2737,67	0
	5 ms	10947,51	0
	3 ms	13734,75	0
	1 ms	5750,59	0

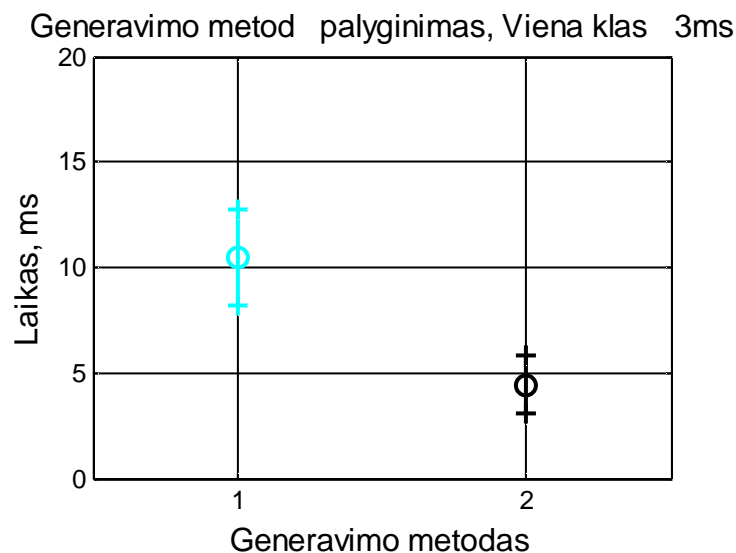
5 lentel . Dispersin s analiz s rezultatai. Programavimo b das: viena klas

Tikslesn s ribos nustatymui, kada generavimo metodas dar yra reikšmingas, papildomai pakartoti eksperimentai su 11ms ir 12ms generavimo laikais. Rezultatai pateikti 6 lentel je. Matome, kad kai generavimo laikas yra 11ms generavimo metodas vis dar reikšmingas, o kai generavimo laikas 12ms – nebereikšmingas. Galima daryti išvad , jog generavimo metodas, kai naudojamas vienos klas s programavimo b das, yra reikšmingas tuomet, kai taikinio jud jimo trajektorijos koordinat s generavimo laikas yra 11ms ir mažesnis.

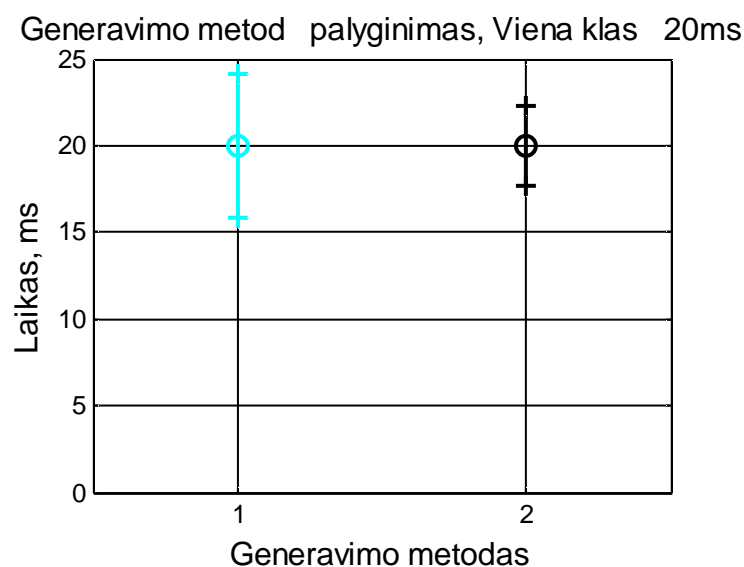
Taikinio generavimas QPainter ir OpenGL metodais			
Programavimo b das	Laikas	F	Tikimyb
Viena klas	12 ms	0,04	0,8504
	11 ms	1,8	0,18

6 lentel . Dispersin s analiz s rezultatai. Programavimo b das: viena klas

20 ir 21 paveiksluose pateikti QPainter ir OpenGL generavimo metod rezultatai – žydra spalva žymi QPainter metod (1-mas generavimo metodas), juoda – OpenGL metod (2-as generavimo metodas). Paveikslai vaizdžiai parodo generavimo metod skirtumus kai generavimo metodas yra reikšmingas (20 pav.) ir kai nereikšmingas (21 pav). Didžiausi tak rezultatams generavimo metodas daro tuomet, kai generavimo laikas yra 3ms (20 pav.), o takos nedaro kai, pavyzdžiui, generavimo laikas yra 20ms (21 pav.).



20 pav. Generavimo metodų palyginimas. Programavimo būdas: viena klasė. „o” – vidurkis, „+” – vidurkis ± standartinis nuokrypis



21 pav. Generavimo metodų palyginimas. Programavimo būdas: viena klasė. „o” – vidurkis, „+” – vidurkis ± standartinis nuokrypis

Metodų palyginimui taip pat atliktas porinis Stjudento testas. Testuota alternatyvioji hipotezė $H=1 \mu_1 > \mu_2$ (QPainter metodo rezultato vidurkis didesnis už OpenGL metodo rezultato vidurkį), esant reikšmingumo lygmeniui 0,05. Rezultatai pateikti 7 lentelėje.

Taikinio generavimas QPainter ir OpenGL metodais Programavimo būdas: viena klasė			
Tikrinama hipotezė	Laikas	H	Tikimybė
$\mu_1 > \mu_2$	20 ms	0	0,5033
$\mu_1 > \mu_2$	18 ms	0	0,5074
$\mu_1 > \mu_2$	15 ms	0	0,5082

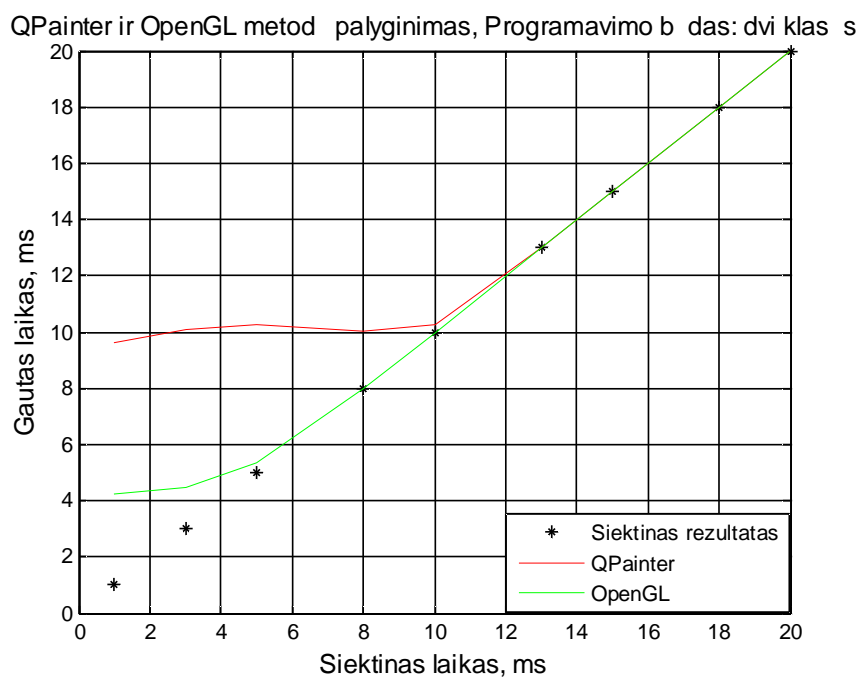
$\mu_1 > \mu_2$	13 ms	0	0,4450
$\mu_1 > \mu_2$	10 ms	1	0
$\mu_1 > \mu_2$	8 ms	1	0
$\mu_1 > \mu_2$	5 ms	1	0
$\mu_1 > \mu_2$	3 ms	1	0
$\mu_1 > \mu_2$	1 ms	1	0

7 lentel . Porinio Stjudento testo rezultatai (p=0,05). Programavimo b das: viena klas s

Iš 7 lentel s matyti, kad QPainter metodu gaut rezultat vidurkis yra didesnis už rezultat , gaut OpenGL metodu, tod l galima teigti, kad OpenGL metodu gautas rezultatas yra artimesnis siektinam.

5.2.2. Dvi klas s

22 paveiksle pateiktas siektino ir gauto rezultato QPainter ir OpenGL metodais palyginimas, kai naudojamas dviej klasi programavimo b das. Kaip ir naudojant vienos klas s programavimo b d , iš paveikslo matyti, kad OpenGL metodu gauti rezultatai artimesni siektinam, nei gauti QPainter metodu.



22 pav. QPainter ir OpenGL gautas ir siektinas laikas. Programavimo b das: dvi klas s

Generavimo metodo reikšmingumo nustatymui atlikta vienmat dispersin analiz . Analiz s rezultatai pateikti 8 lentel je. Abu metodai lyginti prie kiekvieno taikinio jud jimo trajektorijos koordina i generavimo laiko. Nustatyta, kad taikinio jud jimo trajektorijos koordina i generavimo metodas yra reikšmingas tuomet, kai generavimo laikas yra 10ms ir mažesnis.

Taikinio generavimas QPainter ir OpenGL metodais			
Programavimo b das	Laikas	F	Tikimyb
Dvi klas s	20 ms	3,07583E-05	0,9956
	18 ms	0	1
	15 ms	0	0,9685
	13 ms	0	0,9874
	10 ms	37,83	8,30568E-10
	8 ms	2029,27	0
	5 ms	12106,26	0
	3 ms	18876,94	0
	1 ms	8715,27	0

8 lentel . Dispersin s analiz s rezultatai. Programavimo b das: dvi klas s

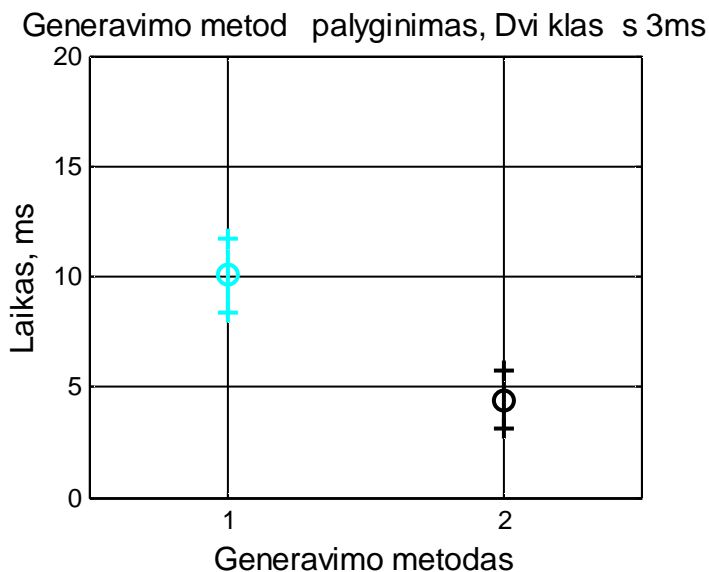
Tikslesn s ribos nustatymui pakartotinai atlikti eksperimentai su 11ms ir 12ms laikais (9 lentel). Matyti, kad kai generavimo laikas 11ms generavimo metodas vis dar reikšmingas, o kai generavimo laikas 12ms – nebereikšmingas. Išvada: kaip ir naudojant vienos klas s programavimo b d , generavimo metodas reikšmingas tuomet, kai taikinio jud jimo trajektorijos generavimo laikas yra 11ms ir mažesnis.

Taikinio generavimas QPainter ir OpenGL metodais			
Programavimo b das	Laikas	F	Tikimyb
Dvi klas s	12 ms	0	0,986
	11 ms	1,9	0,1684

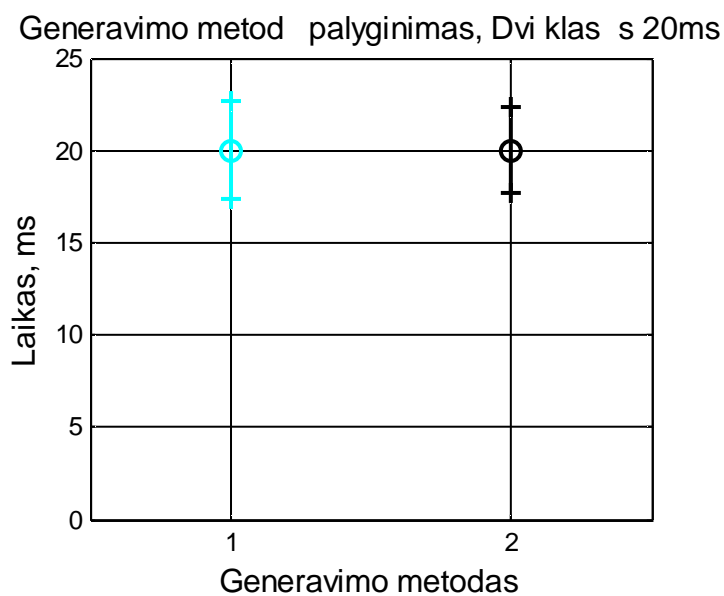
9 lentel . Dispersin s analiz s rezultatai. Programavimo b das: dvi klas s

23 ir 24 paveikslai vaizdžiai parodo generavimo metod skirtumus, kai generavimo metodas yra reikšmingas (23 pav.) ir kai nereikšmingas (24 pav). Žydra spalva žymi QPainter metod (1-mas generavimo metodas), juoda – OpenGL metod (2-as generavimo metodas). Didžiausi tak

rezultatams generavimo metodas daro tuomet, kai generavimo laikas yra 3ms (23 pav.), o tokos nedaro kai, pavyzdžiui, generavimo laikas yra 20ms (24 pav.).



23 pav. Generavimo metodų palyginimas. Programavimo būdas: dvi klasės. “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis



24 pav. Generavimo metodų palyginimas. Programavimo būdas: dvi klasės. “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis

Generavimo metodų rezultatų vidurkių palyginimui atliktas Stjudento testas. Testuota alternatyvioji hipotezė $H=1 \mu_1 > \mu_2$ (QPainter metodo rezultatų vidurkis didesnis už OpenGL metodo rezultatų vidurkį), esant reikšmingumo lygmeniui 0,05. Rezultatai pateikti 10 lentelėje.

Taikinio generavimas QPainter ir OpenGL metodais Programavimo b dais: dvi klas s			
Tikrinama hipotez	Laikas	H	Tikimyb
$\mu_1 > \mu_2$	20 ms	0	0,4978
$\mu_1 > \mu_2$	18 ms	0	0,5
$\mu_1 > \mu_2$	15 ms	0	0,4842
$\mu_1 > \mu_2$	13 ms	0	0,5063
$\mu_1 > \mu_2$	10 ms	1	4,1528E-06
$\mu_1 > \mu_2$	8 ms	1	0
$\mu_1 > \mu_2$	5 ms	1	0
$\mu_1 > \mu_2$	3 ms	1	0
$\mu_1 > \mu_2$	1 ms	1	0

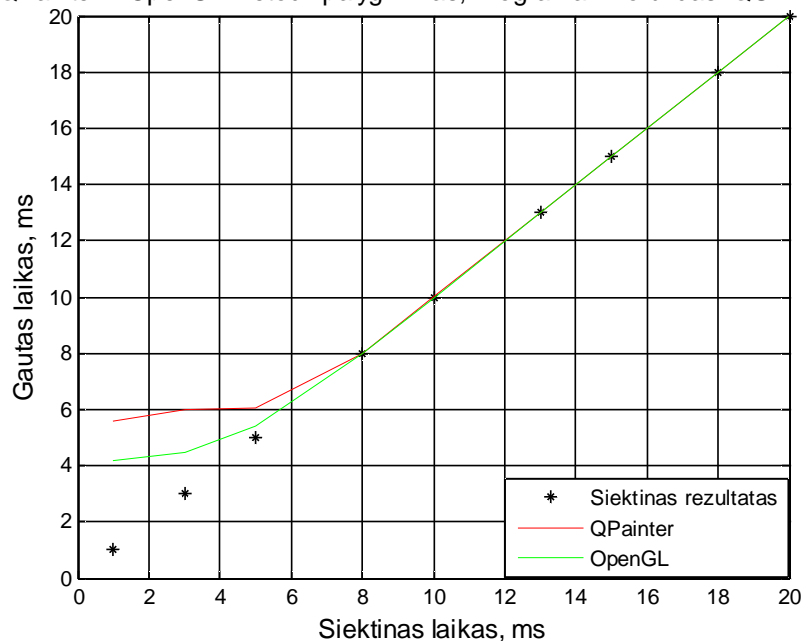
10 lentel . Porinio Stjudento testo rezultatai (p=0,05). Programavimo b dais: dvi klas s

Matome, kad QPainter metodu gaut rezultat vidurkis didesnis už gaut OpenGL metodu, tod l galima teigti, jog OpenGL metodu gautas rezultatas yra artimesnis siektinam.

5.2.3. QGLWidget klas

Kai naudojamas QGLWidget klas s programavimo b dais rezultatai abiem generavimo metodais gauti artimesni siektinam, nei anks iau dviem nagrin tais programavimo b dais. Generavimo metod QPainter ir OpenGL rezultatai ir siektinas rezultatas pavaizduotas 25 paveiksle. Iš šio paveikslo matyti, kad kaip ir ankstesniuose programavimo b duose OpenGL metodu gauti rezultatai artimesni siektinam, nei gauti QPainter metodu.

QPainter ir OpenGL metodų palyginimas, Programavimo būdas: QWidget



25 pav. QPainter ir OpenGL gautas ir siektinas laikas. Programavimo būdas: QWidget klas

11 lentelėje pateikti QPainter ir OpenGL generavimo metodų vienmatės dispersinės analizės rezultatai, kai naudojamas QWidget klasės programavimo būdas. Matome, kad generavimo metodas yra reikšmingas, kai generavimo laikas yra 5ms ir mažesnis.

Taikinio generavimas QPainter ir OpenGL metodais			
Programavimo būdas	Laikas	F	Tikimybė
QWidget klas	20 ms	2,78903E-05	0,9958
	18 ms	0	0,9813
	15 ms	0	0,9917
	13 ms	0	0,9686
	10 ms	0	0,9621
	8 ms	0	1
	5 ms	173,75	0
	3 ms	1541,02	0
1 ms	846,5	0	

11 lentelė. Dispersinės analizės rezultatai. Programavimo būdas: QWidget klas

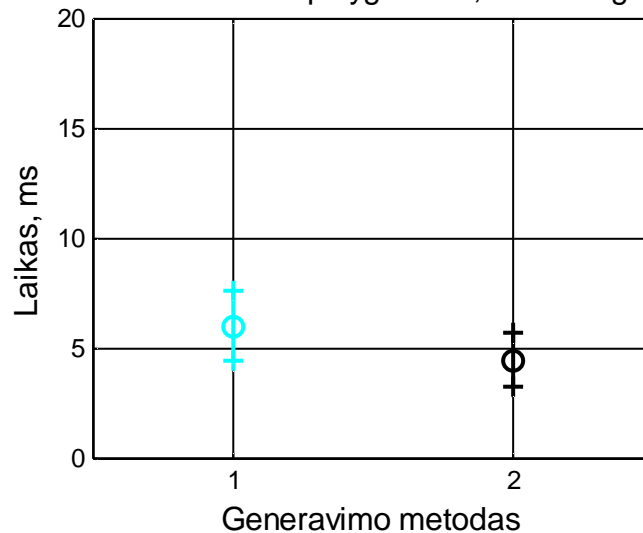
Atlikus pakartotinius tyrimus su 6ms ir 7ms generavimo laikais paaiškėjo, kad generavimo metodas yra reikšmingas ir prie šių laikų (lentelė). Išvada: generavimo metodas reikšmingas tuomet, kai taikinio judėjimo trajektorijos koordinatų generavimo laikas yra 7ms ir mažesnis.

Taikinio generavimas QPainter ir OpenGL metodais			
Programavimo b das	Laikas	F	Tikimyb
QGLWidget klas	7 ms	13,3	0,0003
	6 ms	83,02	0

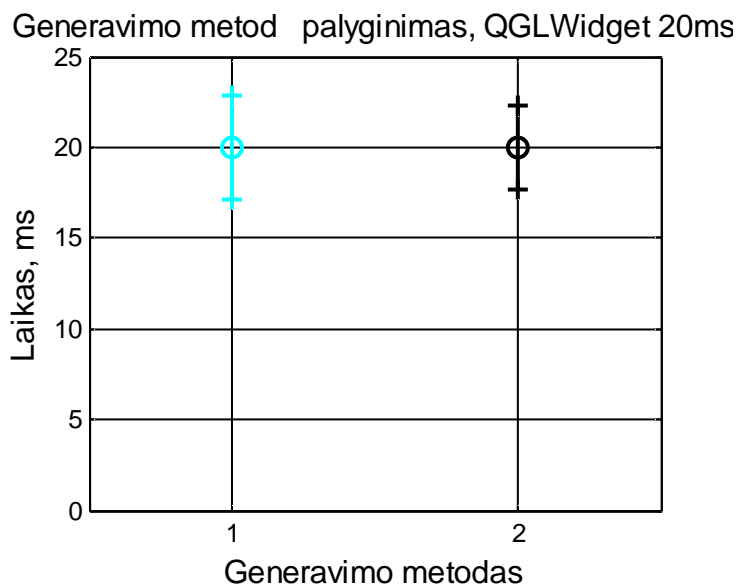
12 lentel . Dispersin s analiz s rezultatai. Programavimo b das: QGLWidget klas

26 ir 27 paveiksiai vaizdžiai parodo generavimo metod skirtumus, kai generavimo metodas yra reikšmingas (26 pav.) ir kai nereikšmingas (27 pav). Žydra spalva žymi QPainter metod (1-mas generavimo metodas), juoda – OpenGL metod (2-as generavimo metodas). Didžiausi tak rezultatams generavimo metodas daro tuomet, kai generavimo laikas yra 3ms (26 pav.), o takos nedaro kai, pavyzdžiui, generavimo laikas yra 20ms (27 pav.).

Generavimo metod palyginimas, QGLWidget 3ms



26 pav. Generavimo metod palyginimas. Programavimo b das: QGLWidget klas . “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis



27 pav. Generavimo metod palyginimas. Programavimo b das: QGLWidget klas . “o” – vidurkis, “+” – vidurkis ± standartinis nuokrypis

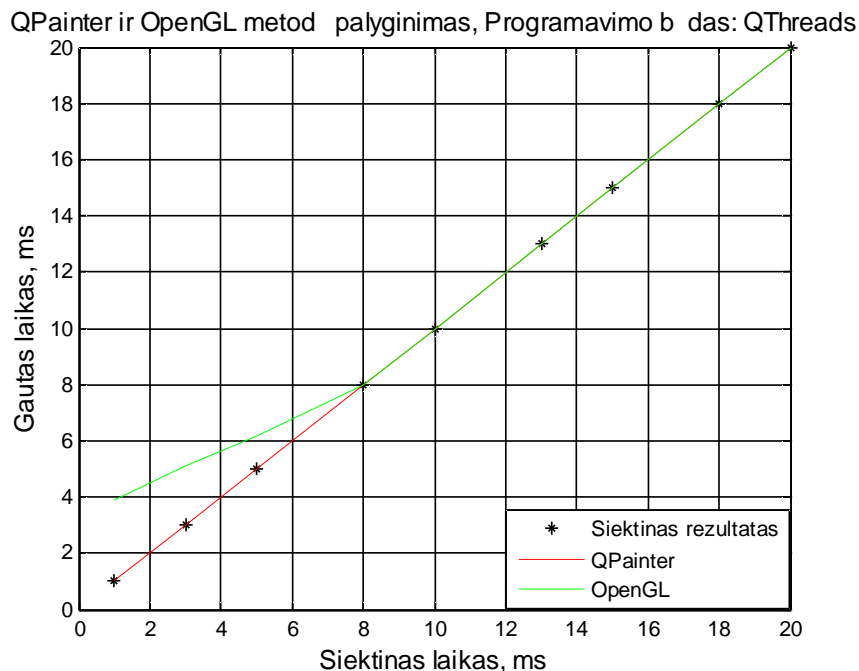
Atlikus porin Stjudento test (13 lentel) (testuota alternatyvioji hipotez $H=1 \mu_1 > \mu_2$ (QPainter metodo rezultat vidurkis didesnis už OpenGL metodo rezultat vidurk), esant reikšmingumo lygmeniui 0,05) paaišk jo, jog, kaip ir ankstesniuose tyrimuose, QPainter metodu gauti rezultatai yra didesnio vidurkio, tod l galima teigti, kad OpenGL metodu gauti rezultatai yra artimesni siektinam.

Taikinio generavimas QPainter ir OpenGL metodais Programavimo b das: QGLWidget			
Tikrinama hipotez	Laikas	H	Tikimyb
$\mu_1 > \mu_2$	20 ms	0	0,4895
$\mu_1 > \mu_2$	18 ms	0	0,5311
$\mu_1 > \mu_2$	15 ms	0	0,4855
$\mu_1 > \mu_2$	13 ms	0	0,4791
$\mu_1 > \mu_2$	10 ms	0	0,4831
$\mu_1 > \mu_2$	8 ms	0	0,5
$\mu_1 > \mu_2$	5 ms	1	0
$\mu_1 > \mu_2$	3 ms	1	0
$\mu_1 > \mu_2$	1 ms	1	0

13 lentel . Porinio Stjudento testo rezultatai (p=0,05). Programavimo b das: QGLWidget klas

5.2.4. QThreads klas

28 paveiksle pateiktas QPainter ir OpenGL metod bei siektino rezultato palyginimas, kai programavime naudojama QThreads klas . Matome, jog skirtingai nei ankstesniuose tyrimuose, pasiektas norimas rezultatas. Tai padaryta QPainter metodu.



28 pav. QPainter ir OpenGL gautas ir siektinas laikas. Programavimo b das: QThreads klas

Atlikus vienmat dispersin analiz paaišk jo, jog generavimo metodas reikšmingas prie vairi laik ir nustatyti slenkstin rib yra sunku (14 lentel).

Taikinio generavimas QPainter ir OpenGL metodais			
Programavimo b das	Laikas	F	Tikimyb
QThreads klas	20 ms	0	1
	18 ms	0,96	0,326
	15 ms	1,02	0,3135
	13 ms	0,31	0,5792
	10 ms	1,01	0,3141
	8 ms	0,96	0,3267
	5 ms	4,62	0,0317
	3 ms	15,59	7,97508E-05
	1 ms	14,17	0,0002

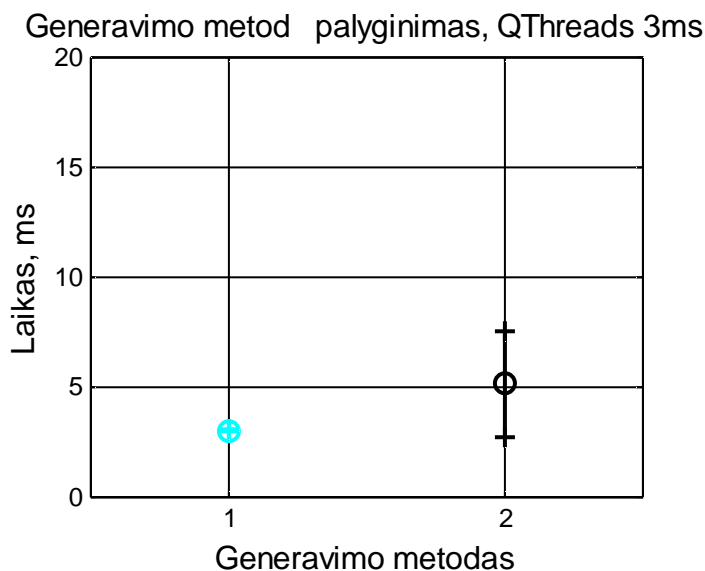
14 lentel . Dispersin s analiz s rezultatai. Programavimo b das: QThreads klas

Papildomai atlikus eksperimentus su kitais generavimo laikais pastebta, kad generavimo metodas reikšmingas prie 15ms, 11ms, 10ms, 6ms ir mažesni laik (14 lentelė ir 15 lentelė).

Taikinio generavimas QPainter ir OpenGL metodais			
Programavimo būdas	Laikas	F	Tikimybė
QThreads klasė	19 ms	0,38	0,5371
	17 ms	0,23	0,628
	16 ms	0	0,9777
	14 ms	0	0,9872
	12 ms	0	0,9867
	11 ms	2,57	0,996
	9 ms	0,84	0,9977
	7 ms	0,11	0,7372
	6 ms	1,42	0,2343

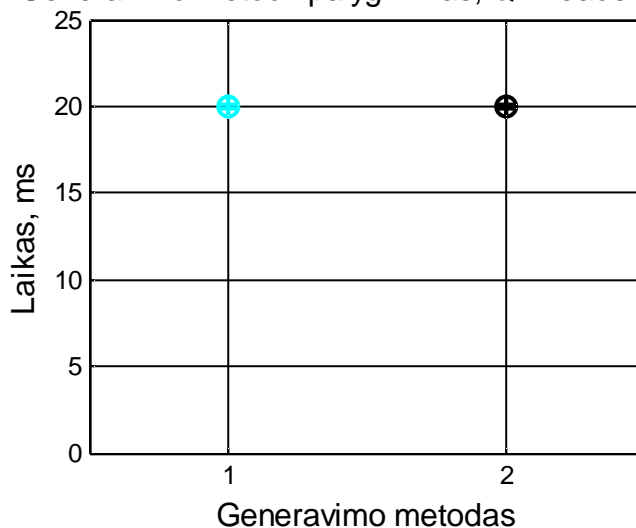
15 lentelė. Dispersijos analizės rezultatai. Programavimo būdas: QThreads klasė

29 ir 30 paveiksluose pavaizduoti generavimo metodų skirtumai, kai generavimo metodas yra reikšmingas (29 pav.) ir kai nereikšmingas (30 pav.). Žydra spalva žymi QPainter metodą (1-asis generavimo metodas), juoda – OpenGL metodą (2-asis generavimo metodas). Didžiausi tokie rezultatai generavimo metodams turi tuomet, kai generavimo laikas yra 3ms (29 pav.), o tokos neturi, pavyzdžiui, kai generavimo laikas yra 20ms (30 pav.).



29 pav. Generavimo metodų palyginimas. Programavimo būdas: QThreads klasė. „o“ – vidurkis, „+“ – vidurkis ± standartinis nuokrypis

Generavimo metodų palyginimas, QThreads 20ms



30 pav. Generavimo metodų palyginimas. Programavimo būdas: QThreads klasė. „o” – vidurkis, „+” – vidurkis ± standartinis nuokrypis

Taip pat metodų palyginimui buvo atliktas porinis Stjudento testas. Kaip ir anksčiau, testuota alternatyvioji hipotezė $H=1 \mu_1 > \mu_2$ (QPainter metodo rezultato vidurkis didesnis už OpenGL metodo rezultato vidurkį), esant reikšmingumo lygmeniui 0,05. Rezultatai pateikti 16 lentelėje. Prie visų taikinio judėjimo trajektorijos koordinatų laikų hipotezė yra neteisinga, vadinasi, QPainter metodu gauti rezultato vidurkis yra mažesnis už OpenGL metodo. Išvada: QPainter metodu gauti rezultatai yra artimesni siektinam, nei gauti OpenGL metodu.

Taikinio generavimas QPainter ir OpenGL metodais Programavimo būdas: QThreads klasė			
Tikrinama hipotezė	Laikas	H	Tikimybė
$\mu_1 > \mu_2$	20 ms	0	0,3505
$\mu_1 > \mu_2$	18 ms	0	0,1713
$\mu_1 > \mu_2$	15 ms	0	0,3113
$\mu_1 > \mu_2$	13 ms	0	0,2006
$\mu_1 > \mu_2$	10 ms	0	0,3150
$\mu_1 > \mu_2$	8 ms	0	0,6373
$\mu_1 > \mu_2$	5 ms	0	1
$\mu_1 > \mu_2$	3 ms	0	1
$\mu_1 > \mu_2$	1 ms	0	1

16 lentelė. Porinio Stjudento testo rezultatai ($p=0,05$). Programavimo būdas: QThreads klasė

Apibendrinant generavimo metodus galima teigti, jog kiekvienas generavimo metodas turi tokios taikinio judėjimo trajektorijos koordinatų generavimą. Pagal tai, kiek dažnai generavimo metodu buvo gautas rezultatas artimesnis siektinam rezultatui, generavimo metodus galima suskirstyti sekančiai:

1. OpenGL metodas – Viena klasė; Dvi klasės; QGLWidget klasė.
2. QPainter metodas – QThreads klasė.

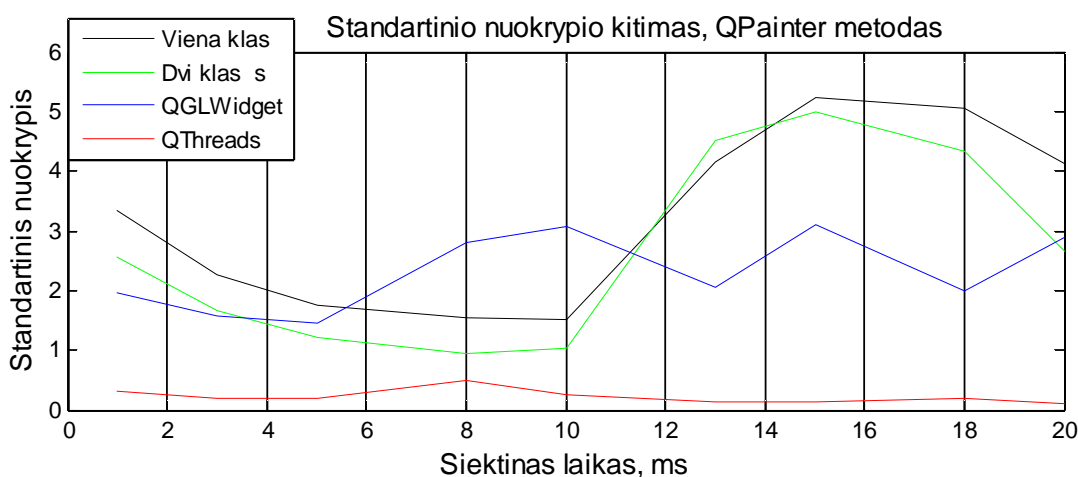
Kur 1 – metodu gauti rezultatai artimesni siektinam.

5.3. Standartinio nuokrypio kitimas

Šiame skyriuje tiriamas standartinio nuokrypio kitimas prie kiekvieno taikinio judimo trajektorijos koordinatų generavimo metodo.

5.3.1. QPainter metodas

31 paveiksle pavaizduotas standartinio nuokrypio kitimas generuojant taikinio judimo trajektorijos koordinates QPainter metodu. Matome, jog standartinis nuokrypis svyruoja nuo ~0,2 (programavimo blokas su QThreads klase) iki ~5,3 (vienos klasės programavimo blokas).



31 pav. Standartinio nuokrypio kitimas generuojant taikinio QPainter metodu

Programavimo bloko standartiniai nuokrypiai palyginimui atliktas porinis Stjudento testas. Testuota hipotezė $H=1 \mu_x > \mu_y$ (programavimo bloko x standartiniai nuokrypiai vidurkis didesnis už programavimo bloko y standartiniai nuokrypiai vidurkis), esant reikšmingumo lygmeniui 0,05. Rezultatai pateikti 17 lentelėje.

Testuota hipotezė	H	Tikimybė
$\mu_4 > \mu_1$	0	1
$\mu_4 > \mu_2$	0	0.9998
$\mu_4 > \mu_3$	0	1
$\mu_1 > \mu_2$	0	0.2268
$\mu_1 > \mu_3$	0	0.0598
$\mu_2 > \mu_3$	0	0.2842

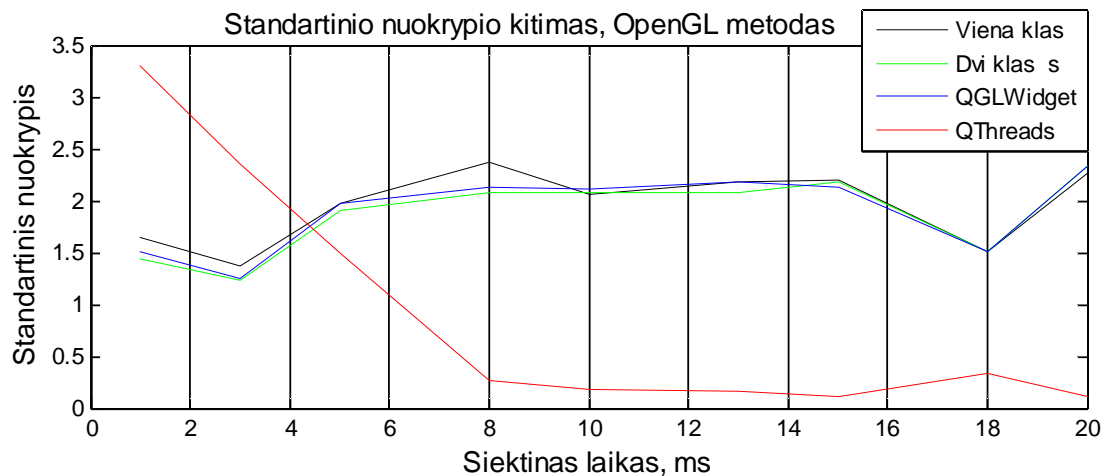
17 lentelė. Porinio Stjudento testo rezultatai ($p=0,05$). Programavimo blokas: QThreads klasė

Remiantis 17 lentelę programavimo būdus pagal standartinį nuokrypį vidurkį galima išrikiuoti sekanti, kur 1 – programavimo būdas su mažiausiu standartinio nuokrypį vidurkiu, 4 – programavimo būdas su didžiausiu standartinio nuokrypį vidurkiu:

- 1) QThreads;
- 2) Viena klasė ;
- 3) Dvi klasės;
- 4) QGLWidget.

5.3.2. OpenGL metodas

32 paveiksle pavaizduotas standartinio nuokrypio kitimas generuojant taikinį OpenGL metodu. Šis standartinis nuokrypis svyruoja nuo ~0,2 (programavimo būdas su QThreads klase) iki ~3,3 (programavimo būdas su QThreads klase).



32 pav. Standartinio nuokrypio kitimas, generuojant taikinį OpenGL metodu

Atlikus porinį Studento testą su visais 4-iais programavimo būdais (18 lentelė) (kaip ir ankstesniame tyrime testuota alternatyvioji hipotezė $H=1 \mu_x > \mu_y$ (programavimo būdas x standartinio nuokrypį vidurkis didesnis už programavimo būdo y standartinio nuokrypį vidurkį), esant reikšmingumo lygmeniui 0,05) paaiškėjo, jog programavimo būdai pagal standartinio nuokrypį vidurkį išrikiuojami tokia pat tvarka, kaip ir su QPainter generavimo metodu:

- 1) QThreads;
- 2) Viena klasė ;

- 3) Dvi klasės;
- 4) QGLWidget.

(1 – programavimo būdas su mažiausiu standartiniu nuokrypi vidurkiu, 4 – programavimo būdas su didžiausiu standartiniu nuokrypi vidurkiu).

Testuota hipotezė	H	Tikimybė
$\mu_4 > \mu_1$	0	0.9879
$\mu_4 > \mu_2$	0	0.9813
$\mu_4 > \mu_3$	0	0.9841
$\mu_1 > \mu_2$	0	0.3173
$\mu_1 > \mu_3$	0	0.3878
$\mu_2 > \mu_3$	0	0.5744

18 lentelė. Porinio Stjudento testo rezultatai (p=0,05). Programavimo būdas: QThreads klasė

Remiantis tyrimo rezultatais galima teigti, jog mažiausias standartinis nuokrypis gaunamas naudojant QThreads klasės programavimo būdą. Iš paveikslėly matyti, jog naudojant OpenGL generavimo metodą kitais 3-mis programavimo būdais gaunamas mažesnis standartinis nuokrypis, nei naudojant QPainter generavimo metodą.

6. IŠVADOS

1. Artimiausias rezultatas siektinam gautas QPainter metodu naudojant QThreads klasės programavimo būdą, tačiau kitais 3-mis programavimo būdais gauti rezultatai artimesni siektinam, naudojant OpenGL metodą.
2. Mažiausias standartinis nuokrypis gautas naudojant QThreads klasės programavimo būdą. Kitais 3-mis programavimo būdais OpenGL metodu gautas mažesnis standartinis nuokrypis nei QPainter metodu.
3. Naudojant QPainter metodą programavimo būdas yra reikšmingas tuomet, kai taikinio judėjimo trajektorijos koordinatų generavimo laikas yra 10ms ir mažesnis. Visais kitais atvejais programavimo būdas nereikšmingas.
4. Naudojant OpenGL metodą programavimo būdas yra reikšmingas tuomet, kai taikinio judėjimo trajektorijos koordinatų generavimo laikas yra 7ms ir mažesnis. Visais kitais atvejais programavimo būdas nereikšmingas.
5. Generavimo metodas, kai naudojamas vienos klasės programavimo būdas, yra reikšmingas tuomet, kai taikinio judėjimo trajektorijos koordinatų generavimo laikas yra 11ms ir mažesnis. Visais kitais atvejais generavimo metodas nereikšmingas.
6. Generavimo metodas, kai naudojamas dviejų klasių programavimo būdas, yra reikšmingas tuomet, kai taikinio judėjimo trajektorijos koordinatų generavimo laikas yra 11ms ir mažesnis. Visais kitais atvejais generavimo metodas nereikšmingas.
7. Generavimo metodas, kai programavime naudojama QGLWidget klasė, yra reikšmingas tuomet, kai taikinio judėjimo trajektorijos koordinatų generavimo laikas yra 7ms ir mažesnis. Visais kitais atvejais generavimo metodas nereikšmingas.
8. Generavimo metodas, kai programavime naudojama QThreads klasė, yra reikšmingas tuomet, kai taikinio judėjimo trajektorijos koordinatų generavimo laikas yra 6ms ir mažesnis. Visais kitais atvejais generavimo metodas nereikšmingas.

LITERATŪROS RAŠAS

1. Biomedicininis inžinerijos mokslinis centras, prieiga per internet : <http://bimc.su.lt/>, (žiūrta 2011 05 21);
2. Rayner K. (Ed), 1992. Eye Movements and Visual Cognition: Scene Perception and reading. New York, NY: Springer-Verlag. (Springer Series in Neuropsychology).
3. Laurutis V., Daunys G., 2006. COGAIN – ES projektas ne galiesiems. Biomedicininė inžinerija. Tarptautinė konferencijos pranešimų medžiaga. Kaunas: Technologija. P. 13-16.
4. D. Molkentin, The Book of Qt 4: The Art of Building Qt Applications. No Starch Press, 2007.
5. Žmogaus sensomotorinės sistemos informacinės charakteristikos, prieiga per internet : <http://bimc.su.lt/tyrimai/62-research/103-informational-characteristics-of-human-sensomotor-system> (žiūrta 2011 05 22)
6. V. Laurutis, R. Zemblys, S. Niauronis, Relationship between pursuit gain and parameters of catch-up saccades // Biomedical Engineering - 2010: Proceedings of International Conference ISSN 2029-3380 p.74-77.
7. R. Zemblys, V. Laurutis, Ribinis taikinio sekimo greitis žmogaus žvilgsniu ir ranka // Jaunųjų mokslininkų darbai. 2009, nr.3(24). ISSN 1648-8776 P. 154-158.
8. G. Liutkus, A. Lenkevičius, Kompiuterinė grafika su OpenGL, Technologija, Kaunas, 2009.
9. QGLWidget Class Reference, prieiga per internet : <http://doc.qt.nokia.com/4.7-snapshot/qglwidget.html#details> (žiūrta 2011 01 12)
10. QThread Class Reference, prieiga per internet : <http://doc.qt.nokia.com/latest/qthread.html> (žiūrta 2011 01 23)
11. Analysis of Variance (Statistics Toolbox(tm)), prieiga per internet : <http://www.mathworks.com/help/toolbox/stats/bqttcvf.html> (žiūrta 2011 05 19);
12. N. Ramanauskas, Aki judesi registravimo nuotoline video – okulografija algoritmai // Daktaro disertacija. Kaunas, 2007;
13. LC Technologies The EyeGaze systems // Prieiga per internet : <http://www.eyegaze.com/PRODUCTS.htm> (žiūrta 2011 04 21);
14. Hennessey C., Nouredin B., Lawrence P., A single camera eye-gaze tracking system with free head motion. // In Proceedings of Eye Tracking Research & Applications (ETRA). – 2006. – P.87 – 94;
15. Šimaitis A., Žmogaus sensomotorinių sistemų informacinis vertinimas // Magistro darbas. Šiauliai, 2009;

PRIEDAI

1. Programin ranga (CD);
2. Tiriomojo darbo aprašymas .pdf formato dokumente.