

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS IR GAMTOS MOKSLŲ FAKULTETAS
ELEKTRONIKOS IR ELEKTROS INŽINERIJOS
KATEDRA

Almantas Zemblys

VIRTUALIOS REALYBĖS ŠALMO SEKIMO
TRIMATĖJE ERDVĖJE REALIU LAIKU
EFEKTYVUMO TYRIMAS

Magistro darbas

Vadovas
prof. dr. V. Lauruška

Šiauliai, 2015

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS IR GAMTOS MOKSLŲ FAKULTETAS
ELEKTRONIKOS IR ELEKTROS INŽINERIJOS KATEDRA

TVIRTINU

Katedros vedėjas

dr. D. Balbonas

2015 06

VIRTUALIOS REALYBĖS ŠALMO SEKIMO
TRIMATĖJE ERDVĖJE REALIU LAIKU
EFEKTYVUMO TYRIMAS

Magistro darbas

Recenzentas

prof. habil. dr. V. Laurutis

2013 05 28

Vadovas

prof. dr. V. Lauruška

2015 05 28

Atliko

RM-13 gr. Stud.

A.Zemblys

2015 05 28

Šiauliai, 2015

Summary

Zemblys A. Efficiency Analysis of Real Time Head Mounted Display Tracking. Master thesis of Signal processing/research advisor. Dr. V. Lauruška, Šiauliai University, Technological Faculty, Electronic and Electrical Engineering Department – Šiauliai, 2015 –45p.

The concept of Virtual Reality has captured imaginations of many scientists and visionaries during the recent decades. In the past few years it has become much more widely available for the public – recent announcement of Head Mounted Displays (HMDs) such as Oculus CV1 and HTC Vive has gathered a huge public interest. Creators of these devices claim that actual feeling of presence in another reality can be achieved through extremely well timed visual stimuli. This feeling is hugely impacted by accurate 6 degree of freedom positional tracking of said HMDs.

Such an accurate positional tracking is only implemented in very few devices in a cost effective manner – it usually requires extremely expensive state of the art positional tracking equipment. However, in systems of said HMDs only infrared cameras and infrared LEDs are used to achieve great accuracies. Furthermore, no company has expressed interest in sharing algorithms they use. It's closed source projects only. Which becomes a problem if one wants to develop additional functionality such as adding other Human-Computer interaction devices to the same camera.

The aim of this Master thesis is to evaluate state of the art algorithms available to the public to achieve greatest possible optical tracking accuracies. Oculus DK2 – a system that is known to work well is used for evaluating efficiency of perspective n point algorithms. Three non iterative algorithms are tested with real data in real time – DLS, EPnP and UPnP. EPnP is also combined with classic iterative Levenberg-Marquardt algorithm as stated in the original research paper.

The findings are that DLS was the slowest of the three – at around 1 ms. While EPnP and UPnP had superior computing time amounting to around 0.05 to 0.06 ms. While EPnP combined with Levenberg-Marquardt algorithm usually produced smallest reprojection error and smallest standard deviation in all 6 degrees of freedom and produced highest effectively measurable distance of 2,05 meters where standard deviations were over 3 degrees of orientation and 1 cm for position. It also outperformed all other algorithms in computational time until the distance of 1,35 meters where it became slower than EPnP and UPnP reaching around 25-35% increase in computing time.

Turinys

Turinys	4
Lentelių sąrašas	6
Paveikslų sąrašas	7
Įvadas	8
1 Virtualios realybės sąvoka ir jai įgyvendinti naudojamos technologijos	10
1.1 VR pateikimo metodai	10
1.1.1 Simuliacija	10
1.1.2 Projekcija	11
1.1.3 VR vaizdavimas ekrane – šalmas	11
1.2 Vartotojo galvos padėties nustatymas.....	14
1.2.1 Infraraudonųjų spindulių žymeklių ant monitoriaus sprendimas	16
1.2.2 Sekimas be žymeklių	17
1.2.3 Naudojant gylio žemėlapius	18
1.2.4 Žymekliai patalpoje	18
1.2.5 Pasyvūs žymekliai	19
1.2.6 Aktyvūs žymekliai	19
1.3 Tyrimai objekto padėties nustatymui iš žinomų jo taškų	20
1.3.1 Mažiausių kvadratų metodas	21
1.3.2 DLS.....	22
1.3.3 EPnP	23
1.3.4 UPnP.....	24
1.3.5 Levenberg–Marquardt metodas	25
2 Tyrimo metodika	26
2.1 Techninė įranga.....	26
2.1.1 Vaizdo jutiklis	26
2.1.2 Sinchronizacija su žymekliais	27
2.1.3 Kompiuteris	28
2.2 Programinė įranga.....	28
2.3 Pasirinkta vaizdų apdorojimo metodika	28
2.3.1 Kameros kalibravimas	28
2.3.2 Komporavimas	29
2.3.3 Žymeklių centrų gavimas	29
2.3.4 Ciklas VR šalmo padėties nustatymui	30
2.4 Tyrimo eiga.....	31
2.5 Algoritmų efektyvumo įvertinimo kriterijai	32
2.5.1 Reprojekcijos paklaida	32
2.5.2 Padėties standartinis nuokrypis	33

2.6	Duomenų apdorojimas	33
3	Tyrimo rezultatai	35
3.1	Algoritmų skaičiavimo laikai.....	35
3.2	Reprojekcijos paklaida.....	37
3.3	Standartiniai nuokrypiai X,Y ir Z ašyse	37
3.4	α , β ir γ pasisukimų standartiniai nuokrypiai.....	39
	Išvados ir siūlymai	42
	Literatūra.....	44

Lentelių sąrašas

2.1 lentelė	27
Kiekvieno aktyvaus IR žymeklio mirksimas 10 bitų kodas dešimtainėje sistemoje	27
3.1 lentelė	39
Didžiulė α pasisukimo kampo neapibrėžtis	39
3.2 lentelė	40
Ištaisyta α pasisukimo kampo neapibrėžtis	40

Paveikslų sąrašas

1.1 pav. Kairėje – simuliacija kariškių apmokymui. Dešinėje – simuliacija apmokant policijos pareigūną	10
1.2 pav. Projektijos pavyzdys	11
1.3 pav. Lęšių sistema <i>Samsung Galaxy Note 3</i> ekranui	12
1.4 pav. Mokymasis valdyti virtualią ranką VR šalmo pagalba	12
1.5 pav. Virtualios realybės šalmas. Kairėje – Development Kit 1, dešinėje – Development Kit 2	13
1.6 pav. Galvos padėčiai erdvėje nusakyti reikalingi parametrai.....	14
1.7 pav. Galvos padėties vaizduoklio atžvilgiu nustatymo sistemos schema	16
1.8 pav. Sudarytas virtualus keturkampis ir jo taškų žymėjimas	16
1.9 pav. Galvos aptikimas pagal veido geometriją	17
1.10 pav. Pakankamas kiekis žymeklių, kad būtų galima sekti vartotojo galvos judesius	18
1.11 pav. Kairėje – vaizdo kamera su IR šaltiniu. Dešinėje – virtualios realybės šalmas su šviesą atspindinčiais žymekliais	19
1.12 pav. Aktyvūs IR žymekliai po <i>Development Kit 2</i> įrenginio permatomu plastikumu	20
1.13 pav. Keturių kameros matomų taškų perspektyvos projekcija	20
1.14 Pav. minimalus kiekis matomų taškų mažiausių kvadratų metodu spresti PnP problemą	22
2.1 pav. USB kamera su infraraudonųjų spindulių filtru	26
2.2 pav. Į kamerą perduodamas sinchronizacijos signalas oscilografo ekrane	27
2.3 pav. Ciklas VR šalmo padėties nustatymui.....	30
2.4 pav. Eksperimento vieta ir kameros matomas vaizdas	31
2.5 pav. Ištaisomi IR spindulių atspindžiai sklandžiam eksperimentui	32
2.6 pav. 0.4 mm poslinkis Y ašyje 35 cm atstumu ties 9 matavimo sekunde	34
3.1 pav. Levenberg–Marquart metodo reprojekcijos paklaidų histograma	35
3.2 pav. pikselių telkinių centrų radimo ir atpažinimo algoritmo skaičiavimo laiko histograma	36
3.3 pav. Vidutinis algoritmų atlikimo laikas.....	36
3.4 pav. Kiekvieno algoritmo reprojekcijos paklaida	37
3.5 pav. Standartiniai nuokrypiai X ašyje	38
3.6 pav. Standartiniai nuokrypiai Y ašyje	38
3.7 pav. Standartiniai nuokrypiai Z ašyje.....	39
3.8 pav. α kampo pasisukimo standartiniai nuokrypiai.....	40
3.9 pav. β kampo pasisukimo standartiniai nuokrypiai.....	41
3.10 pav. γ kampo pasisukimo standartiniai nuokrypiai	41

Įvadas

Per paskutinį šimtmetį vis populiarėjo virtualios realybės idėja – pradedant nuo kino teatrų, perteikiančių daugiau pojūčių, nei vaizdinę informaciją iki pilnai kompiuterio simuliuojamų trimačių aplinkų, ar realių robotų valdymo per atstumą.

Šiandien virtuali realybė labai sparčiai plėtojama dėl 2012 m. išleisto *Oculus VR* kompanijos *Development Kit* virtualios realybės šalmo, kuris susilaukė didelio entuziazmo. Panašių sistemų anksčiau buvo sukurta daug, tačiau šiandien *Oculus VR* kompanija ir jos produktai aiškiai pirmauja dėl mažos kainos, didelio matymo lauko ir itin tikslaus judesio sekimo. Pirmą kartą už prieinamą kainą pasiektas buvimo virtualioje realybėje pojūtis.

Ant įrenginio ekrano projektuojami du stereoskopiniai vaizdai vartotojui suteikia vaizdo gylio pojūtį. O moderni akcelerometro, giroskopo ir magnetometro sistema leidžia itin tiksliai sekti vartotojo galvos pasisukimą visomis ašimis. Tačiau net ir naujausi minėti jutikliai turi pakankamai didelę paklaidą, ir nėra įmanoma sekti vartotojo galvos poslinkio – o tai yra būtina norint suteikti vartotojui pilną erdvinio buvimo virtualioje realybėje pojūtį. Šiai problemai išspręsti *Development Kit 2* sistemoje pasitelkta vaizdo kamera su infraraudonųjų spindulių filtru. Kamera seka infraraudonųjų spindulių šviesos diodus, esančius ant virtualios realybės šalmo. Visa sistema su personaliniu kompiuteriu apskaičiuoja vartotojo galvos padėtį kameros atžvilgiu ir ant šalmo esančiame ekrane atvaizduojamas kompiuterio sugeneruotas vaizdas, kuris atkartoja vartotojo galvos judesius.

Minėta sistema yra uždaro kodo projektas ir nėra išleistų mokslinių straipsnių šios, ar panašios sistemos padėčiai erdvėje nustatyti. Todėl, žinoma, jos potencialus panaudojimas yra stipriai apribotas – viena kamera galima sekti vieną vartotojo šalną, nors pasinaudojant tokia technologija būtų galima sekti keletą tokių arba skirtingų žmogaus – kompiuterio sąveikos įrenginių.

Šiame darbe nagrinėjamas infraraudonųjų spindulių žymeklių sekimo vaizdo kamera panaudojimo efektyvumas *Oculus VR* kompanijos *Development Kit 2* sistemoje. Lyginamas skirtingų algoritmų, skirtų šalmo padėčiai erdvėje nustatyti efektyvumas. Pasirinkti įverčiai algoritmų efektyvumui įvertinti –reprojekcijos paklaida, skaičiavimo laikasir padėties nustatymo neapibrėžtis 6 laisvės laipsniais. Lyginami keturi algoritmai – tiesioginis mažiausių kvadratų (DLS – 2011 m.), efektyvus perspektyvos n taškų (EPnP – 2007 m.), apjungtas perspektyvos n taškų (UPnP 2014 m.) bei EPnP su pridėtu klasikiniu Levenberg-Marquardt algoritmu reikšmėms tikslinti.

Tyrimo tikslas:

Įvertinti skirtingų algoritmų, skirtų objekto padėties nustatymui trimatėje erdvėje infraraudonųjų spindulių žymeklių pagalba atlikimo greitį ir tikslumą bei tinkamumą jutiklių apjungimui realiu laiku veikiančioje sistemoje.

Darbo uždaviniai:

- Apžvelgti literatūros šaltinius, nagrinėjančius virtualios realybės sąvoką ir jai įgyvendinti reikalingas technologijas bei algoritmus, skirtus objekto padėties nustatymui vaizdo kameros pagalba
- Eksperimentinių tyrimų atlikimui sudaryti laboratorinę ir programinę įrangą.
- Išbandyti minėtus algoritmus ir palyginti jais gaunamus parametrus.
- Išanalizuoti gautus eksperimentų rezultatus ir nustatyti dėsningumus.
- Sudaryti realiu laiku veikiančią virtualios realybės šalmo padėties nustatymo sistemą.

1 Virtualios realybės sąvoka ir jai įgyvendinti naudojamos technologijos

Šiame magistro darbe bus nagrinėjamas optinis objekto padėties nustatymas trimatėje erdvėje virtualios realybės kontekste. Tad prieš išsamiau nagrinėjant sekimo galimybes ir apribojimus, tikslinga susipažinti su pačia virtualios realybės sąvoka bei fizinio buvimo pojūčio joje sąvokomis. Taip pat išnagrinėti tam pasiekti naudojamas technologijas ir reikalingus parametrus.

Virtuali realybė (VR) – tai kompiuterio generuojama trimatė aplinka, simuliuojanti fizinį buvimą realiame arba įsivaizduojamame pasaulyje. Įvairios VR sistemos galia tkurti vieną, keletą arba visus penkis žmogaus pojūčius – skonį, uoslę, lytėjimą, regą bei klausą. Kadangi per regą žmogus gauna daugiausia informacijos, šis procesas turi būti labiausiai įtikinantis tam, kad vartotojas jaustų erdvinio buvimo VR pojūtį (angl. *presence*)[1].

1.1 VR pateikimo metodai

Vartotojui perteikti virtualią realybę galima įvairiais būdais. Šiame poskyryje aptariami keli dažniausiai naudojami ir labiausiai koncentruoti į regos stimulą VR pateikimo metodai.

1.1.1 Simuliacija

Simuliacijai reikalingas tam skirtas kambarys ir daug specialiai tam pritaikytos įrangos. Dažniausiai šis būdas naudojamas medicinoje, karo pramonėje, aviacijoje apmokant žmones tinkamai elgtis turima įranga, taip sumažinant galimą klaidų riziką realiose situacijose. Taip pat, Lietuvoje simuliacija naudojama apmokyti policininkus greitai ir atsakingai elgtis ekstremaliose situacijose. Ypač naudinga apmokant kada reikalinga panaudoti jėgą ar šaunamąjį ginklą.



1.1 pav. Kairėje – simuliacija kariškių apmokymui. Dešinėje – simuliacija apmokant policijos pareigūną

1.1.2 Projektcija

VR projektcija arba urvas (angl. *CAVE*) tai kambarys, ant kurio 3-6 sienų projektuojami vaizdai (dažniausia naudojami projektoriai, nors populiarėja ir plokščiaekranių vaizduoklių panaudojimas). Norint gauti vaizdo gylį galima vartotojui uždėti poliarizuotus 3D akinius ir vaizdą perduoti vienai akiai viename kadre dideliu dažniu. Žinoma, galima panaudoti ir kūno padėties atpažinimo metodus.

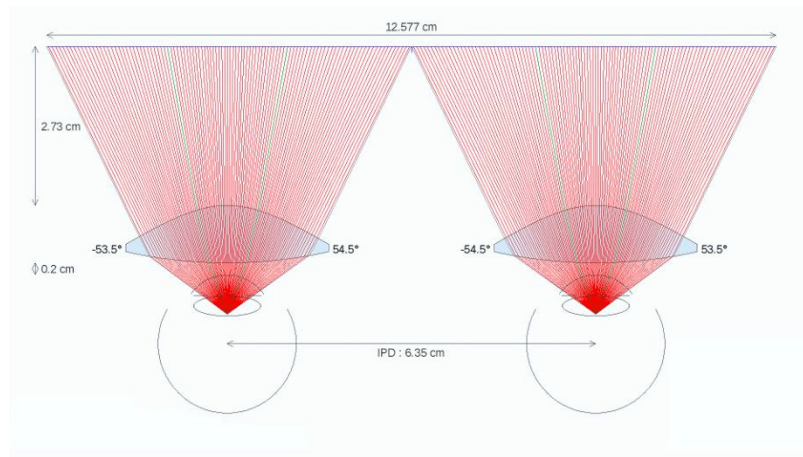


1.2 pav. Projektcijos pavyzdys

Didžiausias šio VR perteikimo metodo trūkumas – projektacijai perteikti reikalingas visas kambarys ir keli aukštos raiškos monitoriai arba projektoriai – tad metodas gali būti itin brangus. Privalumas – galima vaikščioti didelėje erdvėje.

1.1.3 VR vaizdavimas ekrane – šalmas

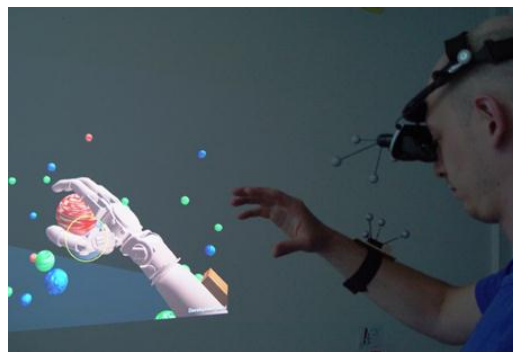
VR šalmas (angl. *Head-mounted display – HMD*) – tai vaizdavimo įrenginys, montuojamas ant vartotojo galvos, prieš vartotojo akis turintis lęšius ir ekraną, ant kurio projektuojamas vaizdas iš skirtingų perspektyvų kiekvienai akiai ir taip suteikiantis vaizdo gylio pojūtį. 1.3 pav. pateikta pavyzdinė sistema su 10 kartų artinančiais asferiniais lęšiais. Horizontalus matymo laukas – 107°.



1.3 pav. Lęšių sistema *Samsung Galaxy Note 3* ekranui

VR šalmi lyginant su projekcija ar simuliacija yra daug pigesnis variantas, nes tokius įrenginius galima gaminti masiškai ir nereikia užimti viso kambario. Virtualiai realybei perteikti gali būti panaudojami galvos sekimo metodai – matomas vaizdas pajuda tiek, kiek pajudėjo įrenginį dėvėtinčio žmogaus galva. Jei atsilikimas labai mažas – žmogus jaučia buvimo VR pojūtį.

VR šalmų panaudojimo sritis labai plati – žinoma, pramogoms (žiūrėti filmus ar žaisti kompiuterinius žaidimus), simuliacijoms pakeisti, potrauminio streso sutrikimams [2], socialinėms baimėms, fobijoms gydyti, architektūros planavimui, 3D modeliavimui, apmokymui, kaip elgtis gavus kūno dalies protezui ir t.t.



1.4 pav. Mokymasis valdyti virtualią ranką VR šalmo pagalba

Oculus ir *Valve* kompanijų inžinieriai ir mokslininkai dirbdami nuo 2012 m. empiriškai nustatė [3], įtikinamo buvimo virtualioje realybėje pojūčiui reikalingas sąlygas:

1. Patogumas – būtina VR šalmo savybė. Jis turi stipriai neapriboti vartotojo laisvės, daug nesverti, lęšiai turi nerastoti. Tik su patogia sistema vartotojas galės ilgą laiką dirbti.
2. Platus matymo kampas (80° ir daugiau).
3. Tinkama rezoliucija (1080p arba daugiau) kiekvienai akiai
4. Pikselių išlikimas ekrane (angl. *pixel persistence*) ne daugiau 3 ms
5. Aukštas atvaizdavimo dažnis ($>60\text{Hz}$, 95Hz yra pakankamai)

6. Globalus ekranas, kuriame visi taškai parodomi vienu metu
7. Optika (daugiausia du lęšiai vienai akiai. Ideali optinė sistema dar nėra praktiška)
8. Optinis kalibravimas kiekvienam vartotojui būtų didelis privalumas
9. Užtikrintas padėties sekimas – milimetro, arba mažesnės paklaidos, reakcija į $\frac{1}{4}^\circ$ pasisukimą, 1,5 m. arba didesnis sekimas į visas puses
10. Mažas vėlavimas (~20 – 25 ms)
11. Erdvinio buvimo pojūčiui būtini visi faktoriai

Nagrinėjamoje sistemoje nepilnai įgyvendinti visi faktoriai. Atvaizdavimo dažnis – 75 Hz, nėra optinio kalibravimo, o maksimalus padėties sekimas ~2–2,5 metro nuo kameros ne 360° diapazonu. Taigi galbūt yra galimybių šiam funkcionalumui pagerinti.

Šiuo metu jau išleisti keli VR šalmi, susilaukę didžiulio entuziazmo:

Oculus DK1 – 2012 m. išleistas įrenginys su 110° matymo lauku ir galvos orientacijos sekimu;

- *Razer OSVR* – atviro kodo produktas;
- *Samsung Gear VR* – įrenginys skirtas Galaxy Note 4 mobilijam telefonui;
- *Sensics dSight* – panoraminio vaizdo VR šalmas;
- *Oculus DK2* – įrenginys nagrinėjamas šiame darbe ir daug kitų.



1.5pav. Virtualios realybės šalmas. Kairėje – Development Kit 1, dešinėje – Development Kit 2

Iškylanti problema – tik *Oculus DK2* turi pilną integruotą 6 laisvės laipsnių sekimo sistemą. Taip pat 2015 metų pabaigoje pasirodys *HTC Vive*, o 2016 m. pradžioje *Oculus* versija skirta vartotojams – abu įrenginiai taip pat turės galvos padėties sekimo funkciją. Kitiems įrenginiams įmanoma nusipirkti tokią funkciją atliekančius priedus, tačiau tai kainuoja labai brangiai.

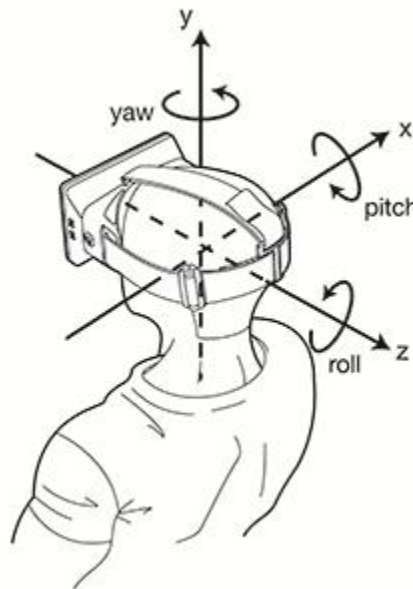
Atsižvelgiant į aukščiau minėtą reikalavimą nr.10–atsilikimas tarp judesio ir vaizdo ekrane turi būti nuo 20 iki 25 ms, reikia paminėti, kad stereoskopinių trimačių vaizdų sugeneravimas ir atvaizdavimas šiuolaikiniuose kompiuteriuose trunka apie 16 – 20 ms, tad virtualios realybės šalmo padėtis turi būti nustatoma daugiausia per 5 ms, tad norint sukurti padėties nustatymo algoritmą,

reikia labai greito ir tikslaus metodo. Dėl šio apribojimo ir dėl to, kad *Development Kit 1* sistemoje jau yra įvykdytas inercinis sekimas naudojant akselerometrą, giroskopą ir magnetometrą [4], toliau nagrinėjamos tik naujausios mašininės regos sistemos.

1.2 Vartotojo galvos padėties nustatymas

Bet kokia vartotojo galvos (arba VR šalmo, pritvirtinto ant vartotojo galvos) padėtis erdvėje gali būti išreiškiama 6 laisvės laipsniais, kurie gali būti skirstomi į 2 grupes (1.3.1 pav.):

- Galvos padėtis X, Y ir Z ašyse – nusakoma vienu tašku ir matuojama metrais,
- Galvos orientacija apie X, Y ir Z (dažniau žymima α , β ir γ) ašis, t.y. Eulerio kampai (angl.: *Euler angles: roll, pitch, yaw*) – matuojama laipsniais.



1.6 pav. Galvos padėčiai erdvėje nusakyti reikalingi parametrai

Galvos padėtis šiame darbe apskaičiuojama kameros koordinatinių sistemoje ir padėties parametrai X, Y ir Z ašyse gali būti išreiškiami perkėlimo transformacija, kuri gali būti užrašoma $x' = x + t$, arba:

$$x' = \begin{bmatrix} I & t \end{bmatrix} \bar{x} \quad (1.1),$$

kur x' – nauja taško trimatėje erdvėje koordinatė, I – 3x3 vienetinė matrica, t – perkėlimo atstumas visomis trimis ašimis, x – taškas, kurį perkeliame iš vienos koordinatinių sistemos į kitą.

Objektus pasukti trimatėje erdvėje šiek tiek sudėtingiau – galima skaidyti pasisukimus apie kiekvieną ašį (pvz.: x, y ir z arba x, y ir x) ir taip išreikšti galvos orientaciją Eulerio kampais, tačiau bendru atveju – tai bloga idėja, nes rezultatas priklauso nuo to, kokia seka buvo atlikti pasisukimai

[5, p. 41]. Taip pat ne visada įmanoma tolygiai judėti parametų erdvėje – kartais vienas ar daugiau Eulerio kampų labai stipriai pasikeičia nežymiai padidėjus objekto pasisukimui. Taip būna, kai antra pasisukimo ašis pasvyra 90° , sutampa su pirmąja ašimi ir taip prarandamas vienas laisvės laipsnis (angl. *gimbal lock*).

Norint išvengti šių problemų darbe visi pasisukimai buvo skaičiuojami vektoriumi $w = \theta \hat{n}$, kur θ – pasisukimo kampas radianais apie \hat{n} normalųjį vektorių. Tada pasisukimo matrica R skaičiuojama pagal formulę:

$$\mathbf{R} = \begin{bmatrix} u^2 + (1 - u^2) \cos \theta & uv(1 - \cos \theta) - w \sin \theta & uw(1 - \cos \theta) + v \sin \theta & 0 \\ uv(1 - \cos \theta) + w \sin \theta & v^2 + (1 - v^2) \cos \theta & vw(1 - \cos \theta) - u \sin \theta & 0 \\ uw(1 - \cos \theta) - v \sin \theta & vw(1 - \cos \theta) + u \sin \theta & u^2 + (1 - w^2) \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.2),$$

kur (u, v, w) – \hat{n} vektoriaus kryptis.

O transformacija, kuri perkelia tašką iš objekto koordinačių sistemos į kameros koordinačių sistemą (kitai vadinama kietoji kūnų transformacija [6, p. 5]) gali būti užrašoma:

$$\mathbf{x}' = \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \bar{\mathbf{x}} \quad (1.3).$$

Taip pat svarbu žinoti kaip atliekama perspektyvos transformacija – kurioje vietoje idealios kameros atveju atsiduria objekto taškas kameros matomame vaizde. Šiai transformacijai atlikti būtina vidinių kameros parametų matrica, kuria prašoma:

$$A = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1.4)$$

Kur

f_x, f_y – židinio nuotolis,

c_x ir c_y – kameros centro x ir y koordinatės,

$\gamma = \alpha_y \cdot \tan \varphi$ – pasisukimo koeficientas, kur φ – pikselių pasisukimo kampas tarp x ir y ašių

(šis parametras dažniausiai būna nereikšmingai mažas).

Tada taško padėtis kameros matomame vaizde:

$$\mathbf{s} \mathbf{m}' = \mathbf{A}[\mathbf{R}|\mathbf{t}]\mathbf{M}'$$

arba

$$\mathbf{s} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1.5),$$

Kur

u ir v – taško projekcija kameros vaizde (x ir y koordinatės),

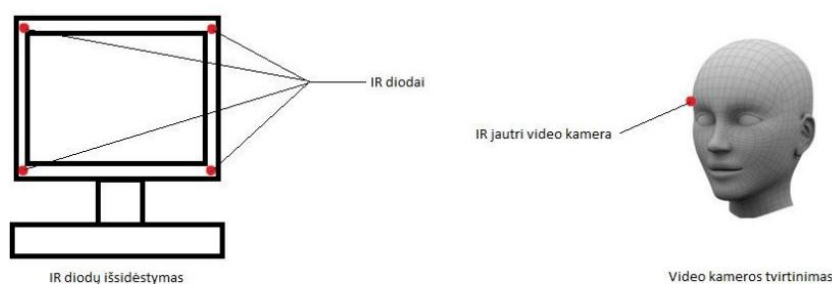
X , Y ir Z – taško padėtis objekto koordinatių sistemoje,

s – mastelio faktorius.

Taigi žinodami, kokiais parametrais kameros atžvilgiu galime aprašyti vartotojo galvos padėtį ir koks šių parametrų sąryšis su kameros koordinatių sistema bei kurioje vietoje atsiduria atitinkamų taškų projekcijos, formuluojame problemą – kokių taškų reiktų ieškoti kameros matomame vaizde. Toliau pateikiami keli tokie principai.

1.2.1 Infraraudonųjų spindulių žymeklių ant monitoriaus sprendimas

2012 m. Šiaulių universiteto studentas Evaldas Barcovas pateikė pilną galvos pozicijos nustatymo ekrano atžvilgiu sistemą, panaudojant ant galvos pritvirtintą kamerą [7]. Keturi infraraudonųjų spindulių šviesos diodai sekami vaizde kameros, pritvirtintos ant vartotojo galvos (1.7 pav.).



1.7 pav. Galvos padėties vaizduoklio atžvilgiu nustatymo sistemos schema

Autoriaus sukurtas algoritmas nustato vartotojo galvos pozicijos ir orientacijos parametrus (1.8 pav. a) analizuojant sudaryto virtualaus keturkampio (1.8 pav.b) parametrus ir keičiant kameros padėties parametrus paeiliui.



1.8 pav. Sudarytas virtualus keturkampis ir jo taškų žymėjimas

Išanalizavęs priklausomybę tarp kameros padėties parametrų pokyčio ir virtualaus keturkampio geometrinių savybių pokyčio, autorius pateikia priklausomybes, pagal kurias būtų galima gauti 6 laisvės laipsnių parametrus:

- Z ašies atstumas skaičiuojamas pagal gauto keturkampio perimetrų santykius pagal

$$Z = \frac{782.439}{P_{pixel}/P_{cm}}; \text{ formulę,}$$

- α (ALFA) kampas gaunamas pagal AB ir CD kraštinių santykį –

$$Alfa = \frac{Alfa_{param} + (0.0000016 * Z - 0.9984)}{0.000048 * Z + 0.0094}; \text{ formulę,}$$

- β (BETA) kampas gaunamas pagal BC ir AD kraštinių santykį –

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

- δ (GAMA) kampas gaunamas pagal $GAMA = \text{Atan} \left(\frac{D_y - C_y}{C_x - D_x} \right) * \left(\frac{180}{3.14} \right);$ formulę,

- X ašies atstumas gaunamas pagal $X = \frac{Beta + (0.0002 * Z - 0.6414)}{0.0173 * Z - 0.011};$ formulę,

- Y ašies atstumas gaunamas pagal $Y = \frac{Alfa + (0.0016 * Z - 0.2261)}{-0.0174 * Z + 0.0231};$ formulę.

Apskaičiavus parametrus pagal šias formules galima gauti kameros padėtį monitoriaus atžvilgiu. Nurodomas gauto algoritmo tikslumas pozicijos $\pm 0,5$ cm ir orientacijos $\pm 0,5^\circ$ parametrus. Vykdyto laikas, priklausomai nuo algoritmo vykdymo aplinkos – 2 – 10,74 ms. Tačiau, kadangi [7] darbe nurodyti parametrai skaičiuojami nenurodant kameros židinio nuotolio ir centrinio taško bei nenurodyta, ar buvo ištaisyti kameros iškraipymai, šios formulės tinka tik autoriaus sukurtoje sistemoje su jo naudojama vaizdo kamera.

1.2.2 Sekimas be žymeklių

Jeigu objekto geometrija žinoma (pvz.: turint CAD modelį), galima jį atpažinti ir rasti jo padėtį erdvėje naudojant algoritmą, kuris lygina kameros vaizdą su trimačiu objekto modeliu [8]. Šiuo atveju lengviausia kameros vaizde aptikti kraštus arba spalvų pakitimus ir lyginti juos su žinomu objekto modeliu. Toks sekimas gali būti panaudojamas ir žmogaus veidui aptikti – išrenkant charakteringus taškus ir sudarant 3D veido reprodukciją [9].



1.9 pav. Galvos aptikimas pagal veido geometriją

Šis metodas pigus jutiklio prasme – reikalinga tik viena kamera. Tačiau realiu laiku apskaičiuojant visus kraštus ir pagal juos taikant trimatį objekto modelį galima stipriai apkrauti duomenis apdorojantį procesorių – reikalingas adaptyvus algoritmas skirtingoms apšviestumo sąlygoms, o nežinant pradinės objekto padėties, reikalingas palyginti ilgas skaičiavimas. Be to – šis sekimo būdas nėra labai tikslus[9] – galima paklaida vienetų laipsnių eilės.

1.2.3 Naudojant gylio žemėlapius

Vaizdo aptikimui galima panauoti jutiklius aptinkančius gylį vaizde, kaip *Kinect* arba *AsusXtion Pro*. Patogu tuo, kad šios technologijos pakankamai išvystytos lengvai aptikti žmogų ir jį išskirti iš aplinkos ir galima gauti jau iš dalies apdorotus duomenis – pvz.: žmogaus skeleto padėtį. Tačiau šios neprofesionalios sistemos vis dar palyginti brangios ir kol kas negali išlaikyti pastovios mažiau, nei 1 mm. paklaidos 1m atstumu [10].

1.2.4 Žymekliai patalpoje

Matomi žymekliai ne ant šalmo gali būti panašūs į QR kodus. Šie žymekliai paprastai turi būti lengvai aptinkami mašininės regos sistemos ir, kadangi jų padėtis žinoma, galima apskaičiuoti kameros padėtį. Tačiau mažesnei, nei 1 mm paklaidai pasiekti reikia labai daug žymeklių (1.10 pav.), ant virtualios realybės šalmo turi būti montuojama kamera – išauga jo svoris. Be to, vartotojas tikrai nenorės savo kambarį apklijuoti žymekliais.



1.10 pav. Pakankamas kiekis žymeklių, kad būtų galima sekti vartotojo galvos judesius

1.2.5 Pasyvūs žymekliai

Pasyvūs žymekliai būna išdėstomi ant sekamo objekto ir yra pagaminti iš šviesą atspindinčios medžiagos. Pati vaizdo kamera turi turėti infraraudonųjų spindulių šaltinį (1.11 pav.). Pirmas objekto padėties apskaičiavimas procesoriui užima daug laiko, nes neaišku, kuris žymeklis kameroje atitinka kokį realybėje. Tad reikia spręsti perspektyvos n taškų problemą įvertinant daug galimų pradinių variantų [11, p. 27]. Apskaičiavus pirmą padėtį tolimesni skaičiavimai paprastesni, tačiau vis tiek apkrauna procesorių daugiau, nei aktyvių žymeklių atveju. Taip pat, tolstant nuo kameros, objekto padėties paklaida didėja daugiau, nei aktyvių žymeklių atveju.



1.11 pav. Kairėje – vaizdo kamera su IR šaltiniu. Dešinėje – virtualios realybės šalmas su šviesą atspindinčiais žymekliais

1.2.6 Aktyvūs žymekliai

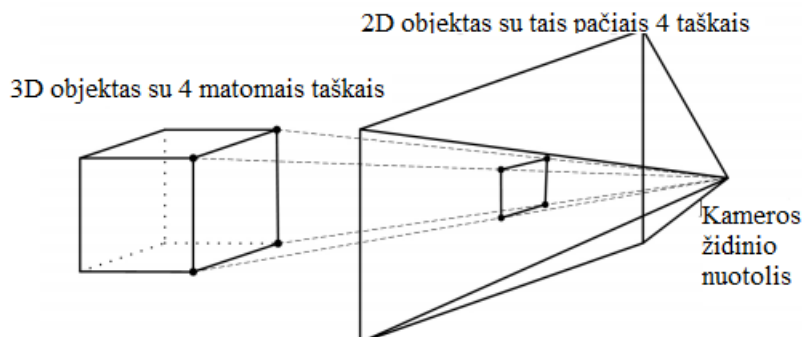
Aktyvūs žymekliai sekami vienos kameros – turbūt efektyviausias būdas nustatyti objekto padėtį erdvėje. Šį būdą galima naudoti prie įvairių apšvietumo sąlygų, žymekliai aiškiai matomi kameros iš pakankamai didelio atstumo lyginant su pasyviais žymekliais, yra pigus lyginant su gylio žemėlapių atveju – kainuoja tik IR šviesos diodai ir vaizdo kamera. Kadangi parametrai geri, o problema daug kartų tirta, galima pakankamai tiksliai ir labai daug neapkraunant procesoriaus (lyginant su sekimu kitais atvejais) apskaičiuoti objekto padėtį trimatėje erdvėje. Dėl šių priežasčių, aktyvių žymeklių būdas ir pasirinktas Oculus *Development Kit 2* sistemoje (1.12 pav).



1.12 pav. Aktyvūs IR žymekliai po *Development Kit 2* įrenginio permatomu plastikumu

1.3 Tyrimai objekto padėties nustatymui iš žinomų jo taškų

Žinodami taškų padėtį kamera gaunamame vaizde bei atitinkamų taškų padėtį objekto trimačiame modelyje (1.13 pav.), galime sužinoti, kokioje padėtyje šis objektas yra kameros atžvilgiu, t.y. gauti poslinkio vektorių ir pasisukimo matricą.



1.13 pav. Keturių kameros matomų taškų perspektyvos projekcija

Algoritmai, sprendžiantys šią problemą vadinami 2D–3D padėties radimo (angl. *2D–3D pose estimation*) arba perspektyvos n taškų (angl. *perspective n -point – PnP*) ir yra skirstomi į dvi dideles grupes:

- Iteraciniai – nutraukiantys objekto padėties paiešką po nustatyto kiekio žingsnių arba pasiekus tam tikrą tikslumą – tad galima sumažinti algoritmo laiką, jei nereikalaujamas didelis tikslumas. Tokio tipo algoritmų tinkamai neinicializavus galima rasti labai netikslią objekto padėtį, arba išvis jos nerasti.

- Ne iteraciniai algoritmai, kurie dažniausiai naudojami tiesinės algebros metodais ir objekto padėtį erdvėje randa per vieną, sudėtingesnę ir daugiau skaičiavimo laiko užimančią žingsnį.

Šiame poskyryje apžvelgiami algoritmai, kurie naudojami VR šarmo perspektyvos n taškų problemai spręsti ir kurių efektyvumas lyginamas šiame darbe – trys ne iteraciniai metodai ir vienas iteracinis.

1.3.1 Mažiausių kvadratų metodas

Tam, kad būtų lengviau suprasti 2D-3D padėties radimo problemą mažiausių kvadratų metodu, pateikiamas bendriausias šios problemos variantas. Iš pradžių randamas apytikslis atstumas, kuriame yra objektas (tai galima padaryti gylį matuojančia kamera arba ortografinė projekcija [12, p. 12]). Tada ieškomas 3D-3D padėties tikslinimas.

Žinome modelio taškus $\{x_1, x_2, \dots, x_n\}$

Atitinkami taškai realioje erdvėje $\{y_1, y_2, \dots, y_n\}$

$$\text{tada } y_n = \mathbf{R}x_n + \mathbf{T} \quad (1.6)$$

$$\text{Tikslo funkcija } \sum_{n=1}^N w_n \|y_n - \mathbf{R}x_n + \mathbf{T}\|^2 \quad (1.7)$$

Supaprastiname problemai suskaičiuojame visų taškų rinkinių vidurkius (centrus) ir paslenkame erdvėje taip, kad jie būtų koordinatinių sistemos pradžioje:

$$\sum_{n=1}^N w_n \|y_n - \mathbf{R}x_n\|^2 \quad (1.8)$$

$$\begin{aligned} \text{Išskleidžiame } \sum_{n=1}^N w_n (\|y_n\|^2 - 2(y_n, \mathbf{R}x_n) + \|\mathbf{R}x_n\|^2) = \\ = \sum_{n=1}^N w_n (\|y_n\|^2 - 2\text{trace}(\mathbf{R}^T \sum_{n=1}^N w_n y_n x_n^T) + \|\mathbf{R}x_n\|^2) \end{aligned} \quad (1.9)$$

Čia trace – pagrindinės matricos įstrižainės elementų sumos operatorius.

Išanalizavus (1.9) formulę, randame, kad tik vidurinis narys kinta, tad reikia rasti maksimumą funkcijos: $2\text{trace}(\mathbf{R}^T \sum_{n=1}^N w_n y_n x_n^T)$

Pažymime koreliacijos matricą \mathbf{K} :

$$\mathbf{K} = \sum_{n=1}^N w_n y_n x_n^T \quad (1.10)$$

Tada reikia apskaičiuoti pasisukimo matricą \mathbf{R} taip, kad:

$$\text{trace}(\mathbf{R}^T \mathbf{K}) \rightarrow \max \quad (1.11)$$

Sprendiniui gauti galime atlikti matricos ypatingųjų reikšmių dekompoziciją (SVD [13]):

$$\mathbf{K} = \mathbf{V} \mathbf{A} \mathbf{U}^T \quad (1.12)$$

$$\text{, kur } \mathbf{A} = \begin{pmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{pmatrix}$$

σ_1, σ_2 ir σ_3 – ypatingosios reikšmės.

tada

$$\mathbf{R} = \mathbf{V} \begin{pmatrix} 1 & & \\ & 1 & \\ & & \det(\mathbf{V}\mathbf{U}^T) \end{pmatrix} \mathbf{U}^T \quad (1.13)$$

Galiausiai poslinkio vektorius $\mathbf{T} = \bar{\mathbf{y}} - \mathbf{R}\bar{\mathbf{x}}$

Šis metodas gali būti lėtas dėl funkcijų kiekio – jei skaičiuojama padėtis dideliame kiekiui taškų, eksponentiškai didėja algoritmo atlikimo laikas, tad išrasti metodai, kurie padėtų įveikti šią problemą.

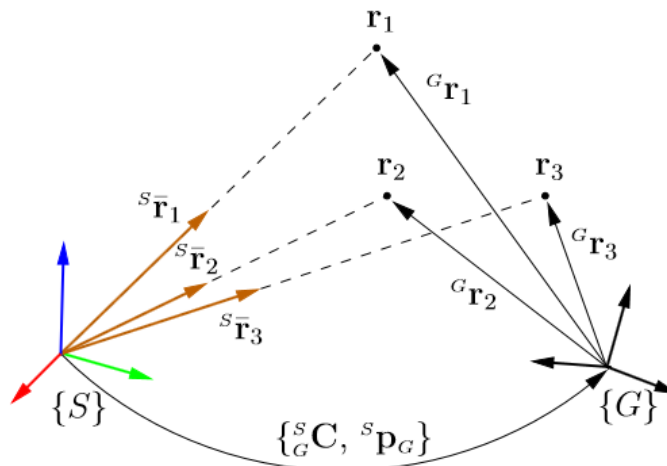
1.3.2 DLS

DLS (angl. *Direct Least Squares*[14]) – tiesioginis mažiausių kvadratų algoritmas, pristatytas 2011m. Joel A. Hesch ir Stergios I. Roumeliotis, kuris gali rasti keletą sprendimų iš turimų $n \geq 3$ taškų. Algoritme pagal kameros matavimo lygtis sudaroma netiesinė mažiausių kvadratų tikslo funkcija, kurios optimalumo sąlyga sudaryta iš trijų trečiojo laipsnio polinomų.

Kameros matomų žinomų taškų projekcija gali būti nusakoma sferiniu kameros modeliu:

$$\begin{aligned} \mathbf{z}_i &= {}^S\bar{\mathbf{r}}_i + \boldsymbol{\eta}_i \\ {}^S\mathbf{r}_i &= {}^S\mathbf{C}^G \mathbf{r}_i + {}^S\mathbf{p}_G \end{aligned} \quad (1.14)$$

Kur z_i yra krypties kameros matavimo dydis vienetinio vektoriaus ${}^S\bar{\mathbf{r}}_i = \frac{{}^S\mathbf{r}_i}{\|{}^S\mathbf{r}_i\|}$, iš kameros koordinatinių sistemos $\{S\}$ link taško i , kurio įvertis sugadintas triukšmo $\boldsymbol{\eta}_i$. Taško koordinatės kameros koordinatinių sistemoje yra funkcija žinomų koordinatinių ${}^G\mathbf{r}_i$, trimačio modelio koordinatinių sistemoje $\{G\}$, ir nežinomų modelio – kameros transformacijos, aprašomos pasisukimo matrica ${}^S\mathbf{C}^G$ ir poslinkio vektoriumi ${}^S\mathbf{p}_G$.



1.14 Pav. minimalus kiekis matomų taškų mažiausių kvadratų metodu spresti PnP problemą

1.14 pav. pavaizduoti trys matomi taškai r_i , $i=1,2,3$. Taškai matomi iš kameros koordinatinių sistemos $\{S\}$ vienetinių vektorių kryptimi ${}^S\bar{r}_i$. Atstumas nuo $\{S\}$ iki kiekvieno taško yra $a_i = \|{}^S\bar{r}_i\|$. Vektorius ${}^S\mathbf{p}_G$ yra $\{G\}$ koordinatinių sistemos pradžia $\{S\}$ atžvilgiu. Pasisukimo matrica iš $\{G\}$ į $\{S\}$ yra ${}^S_C^G$, o ${}^G\mathbf{r}_i$ yra kiekvieno taško padėtis $\{G\}$ koordinatinių sistemoje.

Suformuojama perspektyvos n taškų netiesinė mažiausių kvadratų tikslo funkcija J ir ieškomas jos minimumas:

$$J = \sum_{i=1}^n \|z_i - {}^S\bar{r}_i\|^2 = \sum_{i=1}^n \left\| z_i - \frac{1}{\alpha_i} ({}^S_C^G \mathbf{r}_i + {}^S\mathbf{p}_G) \right\|^2 \quad (1.15)$$

Su apribojimais:

$${}^S_C^G {}^S_C^G {}^S_C^G = \mathbf{I}_3, \quad \det({}^S_C^G) = 1$$

$$\alpha_i = \|{}^S_C^G \mathbf{r}_i + {}^S\mathbf{p}_G\|$$

Imami betriukšmiai geometriniai apribojimai:

$$\alpha_i {}^S\bar{r}_i = {}^S_C^G \mathbf{r}_i + {}^S\mathbf{p}_G, \quad i = 1, \dots, n. \quad (1.16)$$

Ir sudaromas matricos formos vektorius:

$$\underbrace{\begin{bmatrix} {}^S\bar{r}_1 & & -\mathbf{I} \\ & \ddots & \vdots \\ & & {}^S\bar{r}_n & -\mathbf{I} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \\ {}^S\mathbf{p}_G \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} {}^S_C^G & & \\ & \ddots & \\ & & {}^S_C^G \end{bmatrix}}_{\mathbf{W}} \underbrace{\begin{bmatrix} {}^G\mathbf{r}_1 \\ \vdots \\ {}^G\mathbf{r}_n \end{bmatrix}}_{\mathbf{b}}$$

$$\Leftrightarrow \mathbf{Ax} = \mathbf{Wb} \quad (1.17)$$

Čia \mathbf{A} ir \mathbf{b} – žinomi arba išmatuojami dydžiai, \mathbf{x} – nežinomųjų vektorius, kuris turės būti panaikintas, o \mathbf{W} – nežinomų pasisukimo blokinė matrica.

Tolesnis sprendimas, kaip parodyta [14].

1.3.3 EPnP

2007 m. pasiūlytas ne iteracinis metodas perspektyvos n taškų problemai spręsti pavadintas Effective Perspective n point– EPnP[15]. Pagrindiniai šio metodo išskirtinimai – tiesinė skaičiavimo laiko priklausomybė nuo objekto taškų kiekio ($n \geq 4$) ir įvesta keturių virtualių kontrolės taškų sąvoką, kuriais galima išreikšti visus žinomus objekto taškus kameros koordinatinių sistemoje.

Šie keturi taškai pažymimi:

$$\mathbf{c}_j, \quad j=1, \dots, 4,$$

o objekto taškai, kurių koordinatės objekto koordinatinių sistemoje žinomos:

$$\mathbf{p}_i, \quad i=1, \dots, n$$

ir kiekvienas kameros matomas taškas gali būti išreikštas:

$$\mathbf{p}_i^c = \sum_{j=1}^4 \alpha_{ij} \mathbf{c}_j^c \quad (1.18),$$

$$\text{kur } \sum_{j=1}^4 \alpha_{ij} = 1$$

α_{ij} – tai homogeninės svorio centrų koordinatės, kurios apskaičiuojamos tolimesnėse lygtyse.

Po kelių žingsnių gaunamos tiesinės lygtys:

$$\sum_{j=1}^4 \alpha_{ij} f_u x_j^c + \alpha_{ij} (u_c - u_i) z_j^c = 0 \quad (1.19)$$

$$\sum_{j=1}^4 \alpha_{ij} f_v y_j^c + \alpha_{ij} (v_c - v_i) z_j^c = 0 \quad (1.20)$$

Kur f_u, f_v, u_c, v_c – dydžiai iš vidinių kameros parametrų matricos, o u_i ir v_i – \mathbf{p}_i taškų projekcijos kameros vaizde.

Tiesinės lygtys sukoncentruojamos į formą

$$\mathbf{M}\mathbf{x}=\mathbf{0}, \quad (1.21)$$

Kur $\mathbf{x} = [c_1^{cT}, c_2^{cT}, c_3^{cT}, c_4^{cT}]^T$ nežinomųjų kontrolės taškų 12 dimensijų vektorius, o \mathbf{M} – $2n \times 12$ matrica gaunama iš kiekvieno (1.19 ir 1.20) lygčių taško. Lygties sprendinys priklauso \mathbf{M} matricos branduoliui ir gali būti išreikštas:

$$\mathbf{x} = \sum_{i=1}^N \beta_i \mathbf{v}_i \quad (1.22)$$

Keturi galimi sprendimo būdai nurodyti [15]. Išsprendžiami visi keturi ir randamas sprendinys, duodantis mažiausią reprojekcijos paklaidą.

1.3.4 UPnP

2014 m. Laurent Kneip, Hongdong Li ir Yongduek Seo pasiūlytas vieningas perspektyvos n taškų (angl. *Unified PnP*[16]) sprendimas gali būti kelių kamerų atveju. Autorių teigimu, šis metodas garantuotai gražina globalų reprojekcijos minimumą, turi tiesinę priklausomybę nuo ieškomo objekto taškų ir gali gražinti keletą sistemos sprendimų. Algoritmas veikia geriau, nei EPnP, kai matomų taškų yra nedaug ir greičiau, nei DLS, kai matomų taškų labai daug.

Algoritme pirmiausia išvedama tikslo funkcija, kuri yra kvaternionų erdvėje ir atspindi geometrinę paklaidą kaip absoliučios orientacijos funkciją. Visi lokalūs tikslo funkcijos minimumai randami visų stacionarių taškų uždaros formos skaičiavimu. Tai pasiekama skaičiuojant Gröbner

bazę su pirmos eilės optimalumo sąlygomis ir kvaternionų vienetinės normos apribojimu, taip išvengiant Lagranžo daugiklių metodo apribojimams skaičiuoti.

1.3.5 Levenberg–Marquardt metodas

Levenberg–Marquardt metodas – populiari alternatyva originaliam Gauso–Niutono metodui, interpoliuojanti tarp Gauso–Niutono ir gradiento nusileidimo metodo. Atliekamas ieškant $F(x)$ funkcijos minimumą, kuri yra netiesinių funkcijų mažiausių kvadratų suma:

$$F(x) = \frac{1}{2} \sum_{i=1}^m [f_i(x)]^2. \quad (1.23)$$

Levenberg–Marquardt metodas ieško sprendinio kryptimi, kuri gaunama išsprendžiant dydį p_k iš lygties:

$$(\mathbf{J}_k^T \mathbf{J}_k + \lambda_k \mathbf{I}) p_k = -\mathbf{J}_k^T f_k, \quad (1.24)$$

Kur

Indeksas k nurodo iteracijos numerį,

\mathbf{J}_i yra $f_i(x)$ dalinių išvestinių matrica

λ_k – neneigiami skaliariniai dydžiai (gradiento slopinimo koeficientas)

o \mathbf{I} – vienetinė matrica.

Šiame metode gradiento slopinimo koeficientas stipriai padidina tikimybę rasti globalų minimumą, net jei pradedama toli nuo jo.

Naudojant šį algoritmą perspektyvos n taškų problemai spręsti geriausia pradinius netiesinių funkcijų koeficientus perduoti iš kito, apytikslio greito metodo.

2 Tyrimo metodika

2.1 Techninė įranga

Oculus Development Kit 2 sistema sudaryta iš vaizdo kameros ir virtualios šalmo, kuriame integruotas ekranas iš *Galaxy Note 3* išmaniojo telefono. Į ekraną išvedami vaizdai per HDMI sąsają 75 Hz dažniu, o žmogus į jį žiūri per 2 lęšius. Šalme taip pat įmontuoti inerciniai jutikliai – akcelerometras, giroskopas ir magnetometras. Po plastiku priekinėje įrenginio pusėje yra 40 aktyvių infraraudonųjų spindulių žymeklių, kurių pagalba sekama šalmo padėtis erdvėje. Tiek daug žymeklių reikia sistemos patikimumui užtikrinti net ir uždengus vieną ar keletą jų. Šiame poskyryje plačiau nagrinėjamas šios sistemos optinio sekimo principas.

2.1.1 Vaizdo jutiklis

Sistemoje panaudota kamera – USB vaizdo klasės (angl. USB video device class) su infraraudonųjų spindulių filtru (2.1 pav). Kamera vaizdą fiksuoja 60 kadru per sekundę dažniu 752×480 taškų raiška ir naudoja Y8 – 8 bitų kodavimo nespalvotą formatą. Kameroje panaudotas jutiklis – Aptina MT9V034 [17].

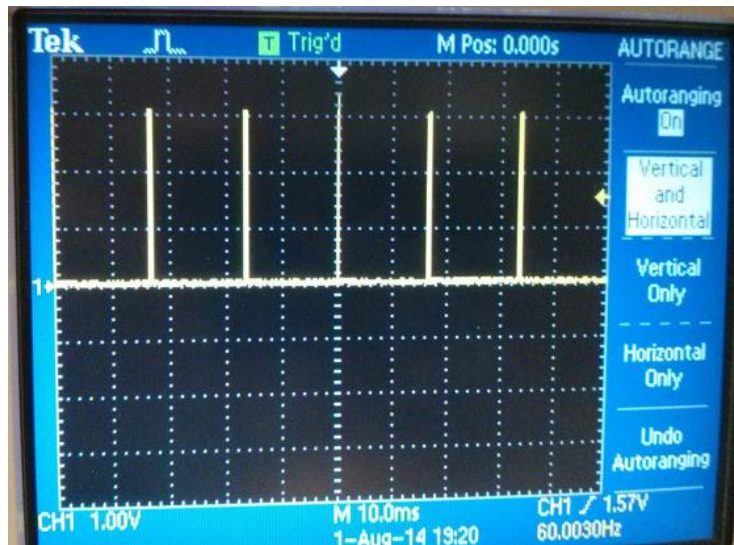


2.1 pav. USB kamera su infraraudonųjų spindulių filtru

Šios kameros jutiklis vienam kadru išsiųsti užtrunka apie 15.4 ms. Kadangi dar užtrunka ir vaizdo siuntimas per USB, o padėtis turi būti apskaičiuojama daugiausia per 5 ms.

Jutiklis gali veikti įprastu režimu, arba momentiniu – kameros užraktas (angl. shutter) atidaromas ir uždaromas pagal išorinį – sinchronizacijos signalą. Tai reikalinga tam, kad būtų aiškus tikslus laikas, kada buvo užfiksuotas vaizdas.

Taigi prie šios kameros jungiami du laidai – USB vaizdo perdavimui ir 3 gyslų laidas, per kurį kamerai perduodamas sinchronizacijos signalas (2.2 pav.).



2.2 pav. Į kamerą perduodamas sinchronizacijos signalas oscilografo ekrane

2.1.2 Sinchronizacija su žymekliais

Ant 1.4 pav. pavaizduoto virtualios realybės šalmo esantys žymekliai šviečia skirtingu ryškumu skirtinguose kadruose. Tai – viena iš įdomiausių ir gudriausių *Development Kit 2* savybių.

Bet kuriuo laiko momentu bet kuris žymeklis šviečia viena iš dviejų būsenų – 100 % stiprumu, arba ~90 % – taip kamerai vaizdu perduodamas dvejetainis kodas. Kiekvieno žymeklio kodas yra unikalus ir kartojasi kas 10 kadro. Taigi perduodamas 10 bitų kodas, skirtas identifikuoti kiekvieną žymeklį. 2.1 lentelėje pateiktas kiekvieno žymeklio kodas paverstas dešimtainiu skaičiumi.

2.1 lentelė

Kiekvieno aktyvaus IR žymeklio mirksimas 10 bitų kodas dešimtainėje sistemoje

2	513	385	898	320	835	195	291	800	675
97	610	482	993	144	592	648	170	27	792
410	345	730	56	827	697	379	251	1016	196
165	21	534	407	916	853	727	308	182	119

Atidžiau išstudijavus šias 10 bitų sekas matoma, kad kiekviena kaimyninė seka skiriasi bent 3 bitais – pagal informacijos teoriją – hemingo atstumas (angl. hamming distance) yra 3. Taigi jei dėl kokių nors priežasčių (triukšmo, ar papildomų IR spindulių šaltinių) aptikto žymeklio sekoje pasikeičia vienas bitas, informacija gali būti lengvai atkuriamą klaidas taisančiu kodu.

Taigi kiekvienas kameros kadras sinchronizuojamas su žymeklių ryškumo pasikeitimu. Taip po dešimties kadrų galima tiksliai sužinoti kuris žymeklis atitinka žymeklį šalmo trimačiame modelyje, kurį galima gauti per USB sąsają. Šis būdas labai palengvina pradinės objekto padėties apskaičiavimo uždavinį.

2.1.3 Kompiuteris

Skaičiavimai atlikti nešiojamu kompiuteriu TOSHIBA SATELLITE L50-B-169. Jo techniniai parametrai:

- Procesorius: Intel Core i7-4500U (1,8 GHz),
- Operatyvioji atmintis: 4 GB DDR3,
- Vaizdo plokštė: AMD Radeon R7 M265 – 2 GB,
- Kietasis diskas: 500 GB (HDD – 5400 RPM).

2.2 Programinė įranga

Programinė įranga kurta Linux operacinės sistemos Ubuntu 14.04 distribucijos aplinkoje C++ programavimo kalba, o kompiliuota su GCC 4.9.2 kompiliatoriumi. Panaudotos funkcijos iš OpenCV v2.4.11, TheiaSfM v0.3, OpenGV ir Vrui–3.2 bibliotekų. Minėtos bibliotekos skirtos vaizdų apdorojimams ir turi integruotus perspektyvos n taškų sprendimo algoritmus. OpenCV galima naudoti Levenbergo–Markvarto metodą; Vrui taip pat turi Levenbergo–Markvarto metodą bei EPnP, iš OpenGV panaudotas UPnP metodas, o iš TheiaSfM – DLS.

2.3 Pasirinkta vaizdų apdorojimo metodika

2.3.1 Kameros kalibravimas

Pirmas žingsnis sprendžiant bet kokią vaizdų apdorojimo užduotį – tai kameros kalibravimas – vidinių kameros parametrų radimas.

Po šių parametrų gavimo, gaunami ir netiesinių iškreipymų – tangentiniai ir radialiniai koeficientai. Kalibravimas atliktas *Matlab* programiniame pakete esančiu *Image Acquisition Tool* įrankiu.

Gautuose iš kameros vaizduose iškreipymai ištaisomi pagal Brown-Conrady vaizdų iškreipymo modelį [18]:

$$x_d = x_u(1 + K_1r^2 + K_2r^4 + \dots) + (P_2(r^2 + 2x_u^2) + 2P_1x_u y_u)(1 + P_3r^2 + P_4r^4 \dots) \quad (2.1)$$

$$y_d = y_u(1 + K_1r^2 + K_2r^4 + \dots) + (P_1(r^2 + 2y_u^2) + 2P_2x_u y_u)(1 + P_3r^2 + P_4r^4 \dots) \quad (2.2)$$

Kur:

(x_d, y_d) – iškraipyto vaizdo taškai,

(x_u, y_u) – ištaisyti vaizdo taškai,

(x_c, y_c) – iškraipymų centras

K_n – n-tasis radialinis iškraipymo koeficientas

P_n – n-tasis tangentinis iškraipymo koeficientas

$$r = \sqrt{(x_u - x_c)^2 + (y_u - y_c)^2} \quad (2.3)$$

2.3.2 Komporavimas

Komporavimas (angl. *binary threshold*) atliekamas norint išskirti, didžiausio ryškumo pikselius vaizde. Kiekvienas pikselis lyginamas su pasirinkta verte; jei vertė didesnė, nei pasirinkta – pikselis turėtų priklausyti kažkuriam žymekliui ir yra išrenkamas į atskirą masyvą.

$$\theta(f, t) = \begin{cases} 1 & \text{if } f \geq t, \\ 0 & \text{else,} \end{cases} \quad (2.4)$$

Šių pikselių vertės toliau perduodamos į taškų centroidų gavimo algoritmą.

2.3.3 Žymeklių centrų gavimas

Po komporavimo tikrinama, kiek pikselių iš eilės yra kiekvienoje eilutėje. Kiekvienam pikselių telkiniui (angl. *blob*) suteikiamas unikalus numeris. Išrinkus pikselių telkinį iš kitos eilutės, tikrinama, kuris iš jų liečiasi su praeitos eilutės; jei liečiasi – telkiniai sujungiami į vieną. Kiekvieno pikselių telkinio centras gaunamas formulėmis:

$$X = (X_{\max} - X_{\min}) / 2 + X_{\min} \quad (2.5)$$

$$Y = (Y_{\max} - Y_{\min}) / 2 + Y_{\min} \quad (2.6)$$

Kur:

(X, Y) – gaunamas pikselių telkinio centras.

X_{\max} – didžiausia pikselių telkinio reikšmė x ašyje,

X_{\min} – mažiausia pikselių telkinio reikšmė x ašyje,

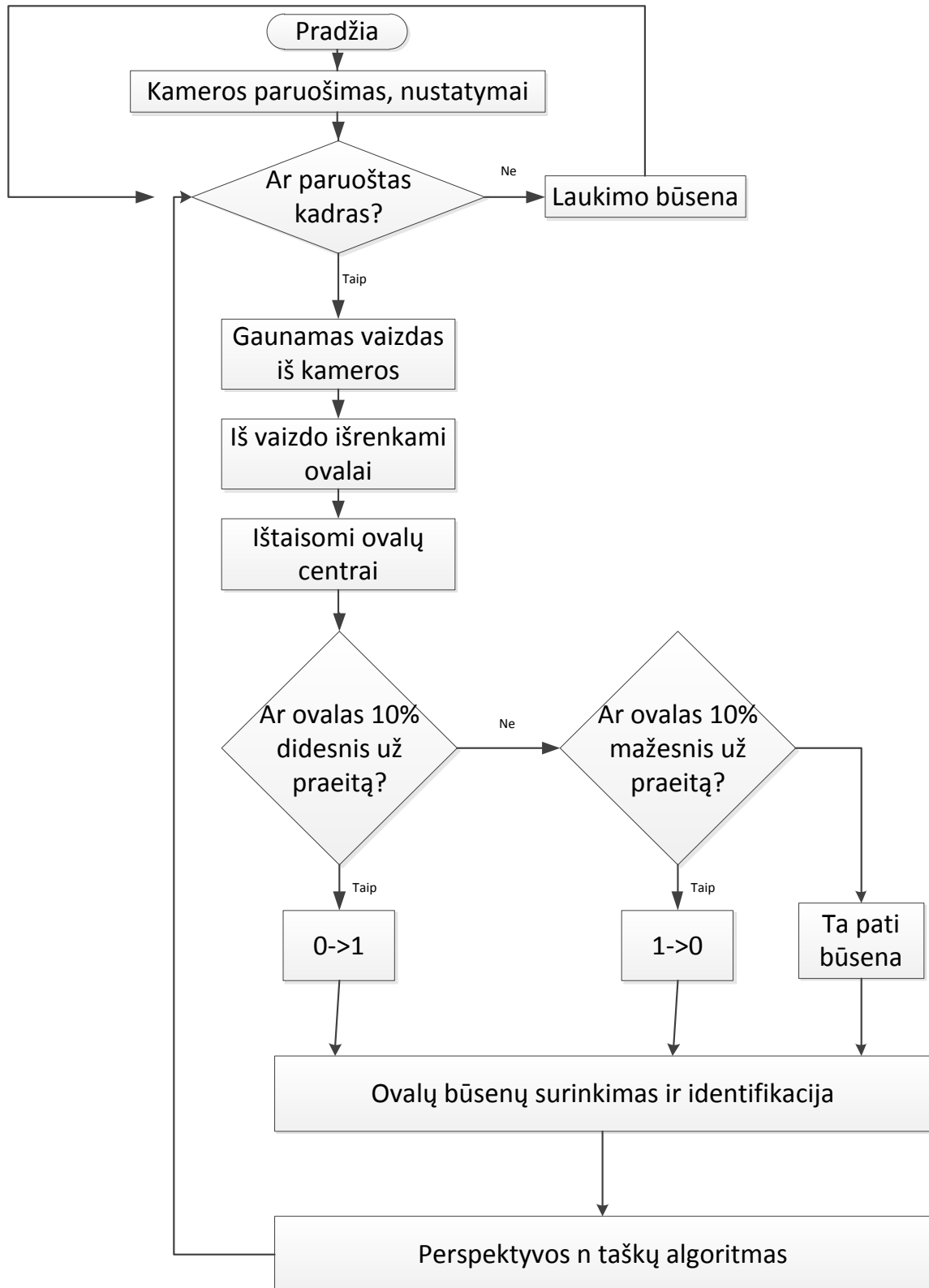
Y_{\max} – didžiausia pikselių telkinio reikšmė y ašyje,

Y_{\min} – mažiausia pikselių telkinio reikšmė y ašyje.

analogiškai skaičiuojamos vertės Y ašyje.

2.3.4 Ciklas VR šalmo padėties nustatymui

Sudarytas ciklas, skirtas rasti VR šalmo padėtį iš gautos vaizdinės medžiagos:



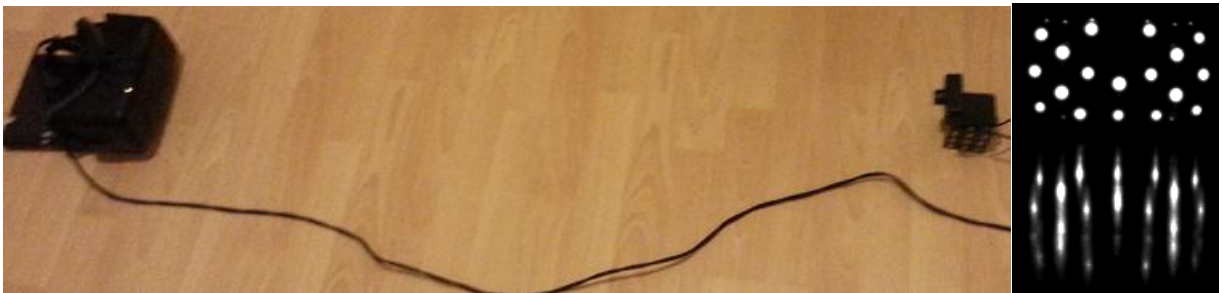
2.3 pav. Ciklas VR šalmo padėties nustatymui

Tam, kad būtų aiškesnis algoritmas, tiksliau apibrėžiamas eiliškumas:

- 1) Išsiunčiamos kameros ir infraraudonųjų žymeklių aktyvavimo sekos per USB sąsają.
- 2) Gaunamas vienas kadras ir komporavimo metodu išrenkami ryškiausi vaizde matomi taškai.
- 3) Iš ryškiausių matomų taškų išrenkami visi ovalai. Jų dydžiai ir centrai išsaugomi masyve.
- 4) Gaunamas kitas kadras, analogiškai atliekama komporavimo procedūra ir išrenkami ovalai.
- 5) Šio kadro ovalų dydžiai lyginami su artimiausiais praeito kadro ovalų dydžiais: Jei naujame kadre ovalas 10 % didesnis, atmintyje išsaugomas pakitimas $0 \rightarrow 1$, jei 10 % mažesnis $1 \rightarrow 0$. Šis veiksmas kartojamas 10 kadrų ir taip identifikuojamas kiekvienas vaizde matomas žymeklis.
- 6) Jei identifikuoti bent 4 žymekliai, paleisti vieną iš padėties nustatymo metodų.
- 7) Jei algoritmas aptiko šalmo padėtį su tenkinančia paklaida, atvaizduoti VR šalmo modelį ekrane.

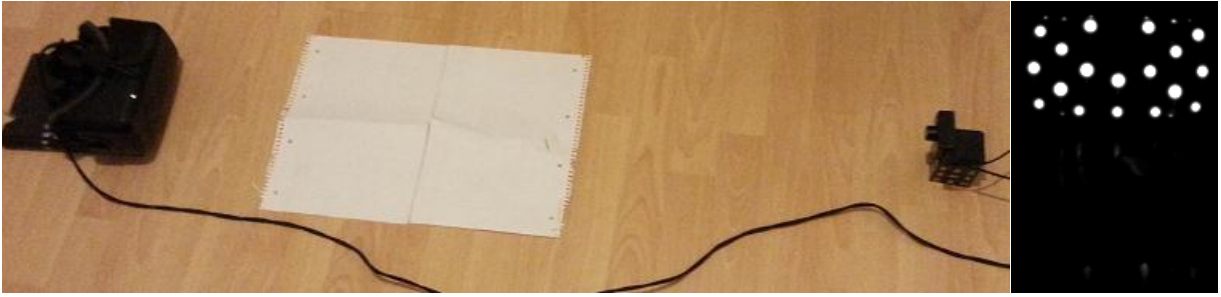
2.4 Tyrimo eiga

Šiame darbe algoritmų efektyvumo kriterijai vertinami dažniausiai pasitaikančiu atveju, koks būtų realioje sistemoje – VR šalmas atsuktas tiesiai prieš vaizdo kamerą. Nagrinėtas tik statinis atvejis – nejudinant įrenginio, nes eksperimentas kartojamas kiekvienam algoritmui ir nėra galimybės atkartoti lygiai tokį patį veiksmą su kitu algoritmu. Be to, judėjimas galėtų daryti labai didelę įtaką matavimo rezultatams iteracinio algoritmo atveju.



2.4 pav. Eksperimento vieta ir kameros matomas vaizdas

Tačiau pastebėta, kad tokiu atveju susidaro atspindys (2.4 pav.), kuris gali būti atpažintas vaizdo kameros, kaip žymeklis. Šiam atspindžiui eliminuoti panaudoti du A4 formato popieriaus lapai, kurie sugertų IR spindulius:



2.5 pav. Ištaisomi IR spindulių atspindžiai sklandžiam eksperimentui

Iš pradžių VR šalmas padedamas 25 cm atstumu nuo kameros ir paleidžiama sukurta programa su vienu PnP algoritmu ir laukiama, kol praeis viena minutė. Tada paleidžiama programa su kitu algoritmu ir vėl laukiama 1 minutę.

VR šalmas pastumiamas 10 cm ir vėl paleidžiamos visos programos paeiliui.

Maksimalus atstumas, iki kurio pastumiamas VR šalmas 10 cm žingsniu – 2,05 m.

2.5 Algoritmų efektyvumo įvertinimo kriterijai

Kadangi nėra bendro standarto palyginti objekto padėties aptikimo algoritmams [12], išskirti pagrindiniai kriterijai, nusakantys aptikimo tikslumą, pagal kuriuos lyginsime šiuos algoritmus. Kiekvieno algoritmo kriterijų imtis – paskutinių 3600 kadrų (1 minutės) gauti įverčiai.

2.5.1 Reprojekcijos paklaida

Reprojekcijos paklaida yra bendra forma funkcijos, kurios minimumas ieškomas perspektyvos n taškų algoritmais ir apibrėžiama pagal formulę:

$$\sum_{i=1}^n d(x_i, \hat{x}_i)^2 \quad (2.7)$$

Kur x_i – matomo žymeklio centras

\hat{x}_i – algoritmo spėjamas žymeklio centras (apskaičiuota koordinatė padauginta iš pasisukimo matricos ir poslinkio vektoriaus),

d – euklidinio atstumo skaičiavimo operatorius,

n – kameros matomų žymeklių kiekis.

Ši paklaida matuojama pikseliais ir dažniausiai yra gražinama paties algoritmo, kuris ieško funkcijos minimumo. Pagal reprojekcijos paklaidą galima spręsti, ar pakankamai tiksliai buvo apskaičiuota objekto padėtis erdvėje ir tai yra pirmas įvertis, kuris parodytų kokią nors problemą. Iteracinio algoritmo atveju šis įvertis yra algoritmo stabdymo sąlyga.

2.5.2 Padėties standartinis nuokrypis

Kadangi skirtingų algoritmų efektyvumas vertinamas statinėje sistemoje – t.y. nejudant sekamam objektui, daug informacijos apie rastos objekto padėties netikslumą suteikia duomenų išsibarstymas apie vidurkį – standartinis nuokrypis.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad (2.8)$$

Kur σ – standartinis nuokrypis,

N – matuotų reikšmių kiekis,

x_i – gautas dydis, o μ – reikšmių vidurkis.

Standartiniai nuokrypiai atskirai apskaičiuoti kiekvienai nustatytai VR šalmo padėčiai – poslinkio vektoriaus nariams.

Standartiniai nuokrypiai taip pat apskaičiuoti kiekvienam Eulerio kampui. Vertimas iš pasisukimo matricių atliktas pagal [19] dokumentą.

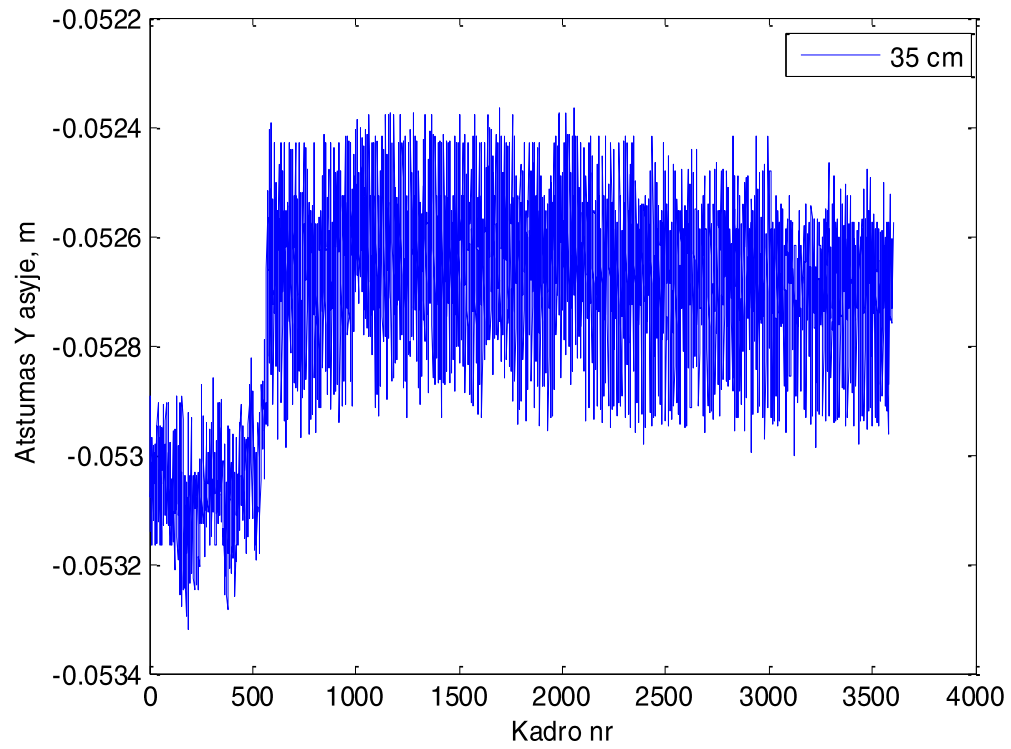
Algoritmo skaičiavimo laikas

Taip pat matuotas kiekvieno įgyvendinto algoritmo laikas, paverstas milisekundėmis (naudotos C++ funkcijos laikui skaičiuoti).

2.6 Duomenų apdorojimas

VR šalmo padėties skaičiavimo metu į atskirus failus įrašomi algoritmo efektyvumo kriterijai kiekvienam kadru.

Šie failai vėliau apdorojami MATLAB programine įranga. Iš pradžių nubraižomi visi išmatuoti dydžiai (MATLAB funkcija *plot*) ir tikrinama, ar duomenyse nėra akivaizdaus dreifo arba kitų netikslumų (2.6 pav.) – jei jis yra, matavimas toje padėtyje pakartojamas.

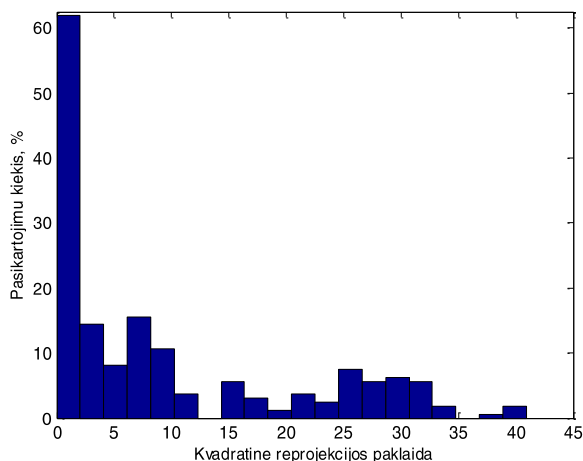


2.6 pav. 0.4 mm poslinkis Y ašyje 35 cm atstumu ties 9 matavimo sekunde

Vidurkiams skaičiuoti naudota funkcija *mean*, standartiniam nuokrypiui – *std*, histogramoms – *hist*.

3 Tyrimo rezultatai

Darant eksperimentą buvo išbandytas Levenber-Marquardt (LM) algoritmas be jokios pradinės VR šalmo modelio padėties – algoritmas kiekvienoje iteracijoje ieškojo galimos padėties nuo kameros koordinatų sistemos pradžios. Pasirinktas maksimalus iteracijų kiekis – 200 (dvigubai daugiau, nei standartinė reikšmė). Kaip pavyzdys, 3.1 pav. pateikiama reprojekcijos paklaidos histograma.



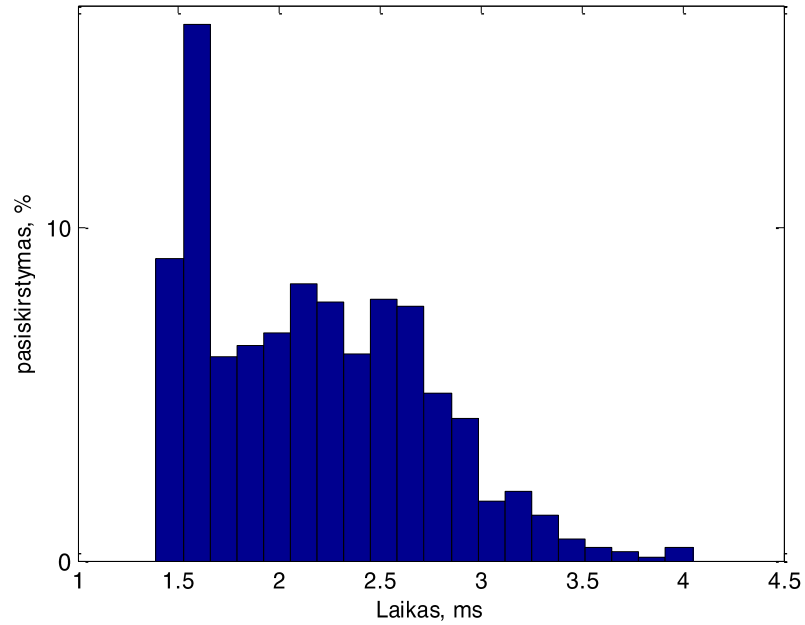
3.1 pav. Levenberg–Marquart metodo reprojekcijos paklaidų histograma

Taigi iškart aišku, kad šiam algoritmui efektyviai veikti reikalinga apytikslė objekto padėtis erdvėje. Pasirinktas algoritmas, kurio gautos transformacijos duomenys perduodami į LM – EPnP, nes pagal [14] aprašą, šiam algoritmui patobulinti siūloma Gauso-Niutono tipo optimizacijos schema. Šis modifikuotas algoritmas toliau žymimas EPnP+LM ir EPnP algoritmą paleidžia tik pirmame kadre, arba jei reprojekcijos paklaida praeitame kadre buvo didesnė už 2 pikselius. Pasirinktas maksimalus iteracijų kiekis – 50.

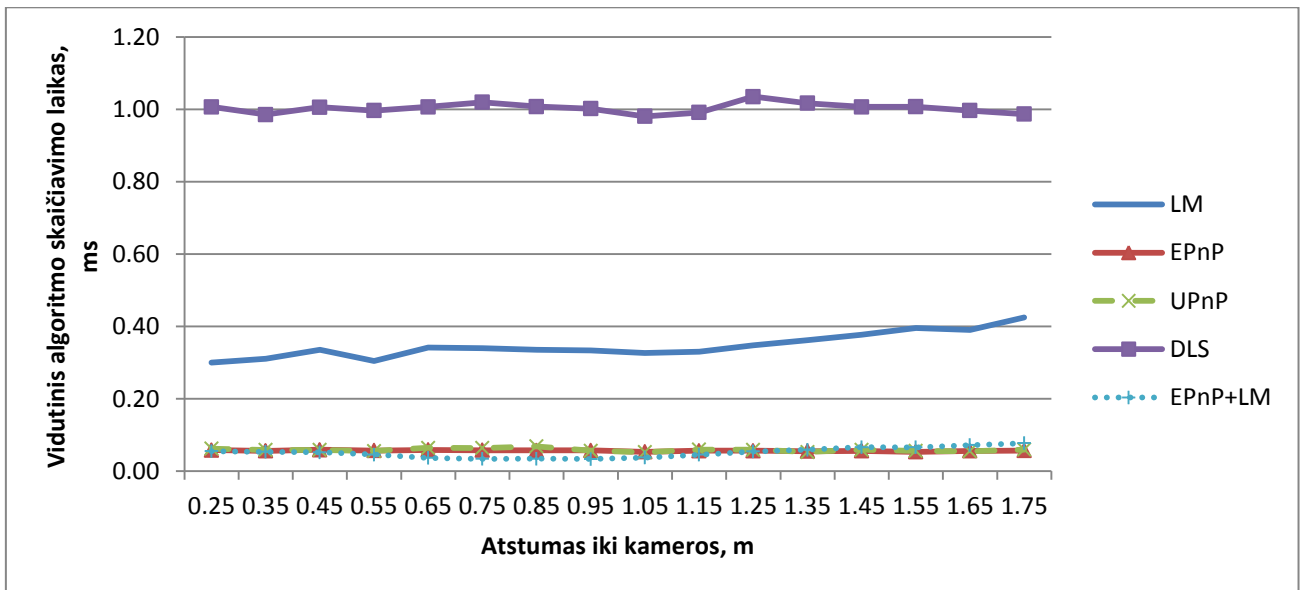
Taip pat, paprasto LM algoritmo reprojekcijos paklaidos ir standartiniai nuokrypiai toliau nepateikiami – šie rezultatai dešimtimis arba šimtais kartų viršija pageidaujamos sistemos įverčius.

3.1 Algoritmų skaičiavimo laikai

Po duomenų apdorojimo pirmas gautas parametras kiekvienam algoritmui – skaičiavimo laikas kiekvienoje padėtyje. Kadangi pikselių telkinių aptikimo efektyvumas nėra lyginamas su kitais algoritmais, kaip pavyzdys pateikiama šios algoritmo dalies histograma:



3.2 pav. pikselių telkinių centrų radimo ir atpažinimo algoritmo skaičiavimo laiko histograma
Apskaičiuojamas vidurkis kiekvieno algoritmo atliekamam laikui visose padėtyse:



3.3 pav. Vidutinis algoritmų atlikimo laikas

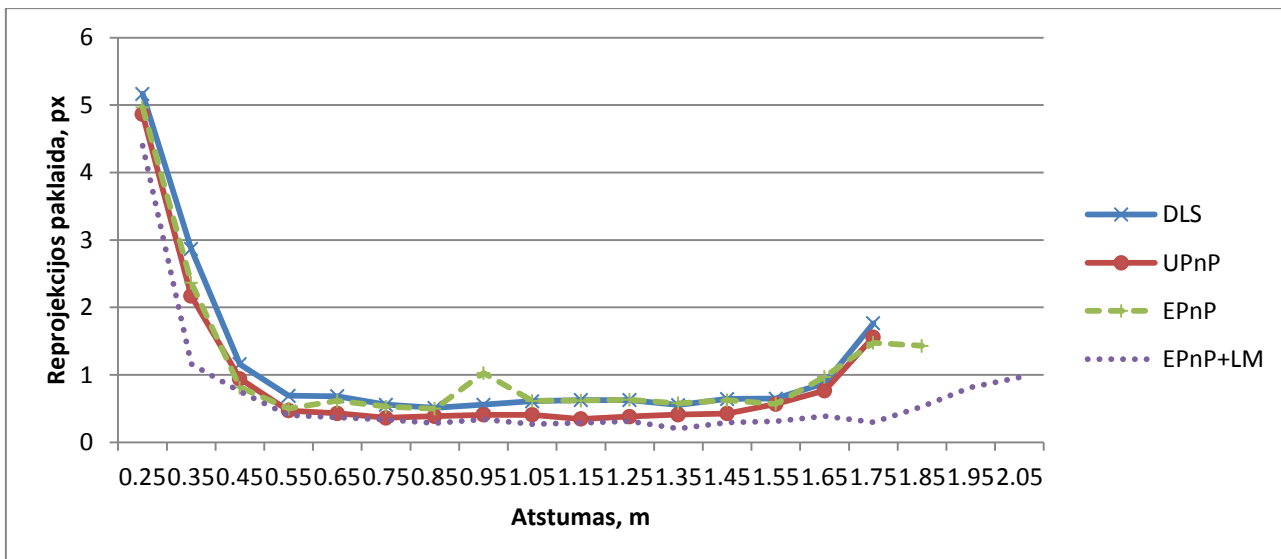
Iš šio grafiko akivaizdžiai matomas UPnP ir EPnP algoritmų skaičiavimo spartumo pranašumas (0,05–0,06 ms visuose atstumuose). Nuo jų atsiliko LM algoritmas, kuriam parinkti parametrai –200 iteracijų ir 0,25 px reprojekcijos paklaida. Šiuo atveju LM algoritmo vykdymo sparta priklauso nuo atstumo, nes 0,25 px paklaidą algoritmui gauti nėra lengva, tad kuo toliau objekto taškai – tuo didesnis triukšmas ir tuo daugiau iteracijų trunka gauti tokią mažą reprojekcijos paklaidą, arba dėl triukšmo – tokios paklaidos gauti neįmanoma.

Jungtinis EPnP ir LM algoritmas 0,45 – 1,15 m intervale buvo sparčiausias ir lenkė EPnP ir UPnP algoritmus 0,01 – 0,02 ms. Didesniu nuotoliu šis algoritmas tapo tokiu pačiu dydžiu lėtesnis.

Lėčiausiai atliktas algoritmas – DLS (truko apie 1 ms visiems atstumams).

3.2 Reprojekcijos paklaida

Toliau analizuojama reprojekcijos paklaida kiekvienam algoritmui:



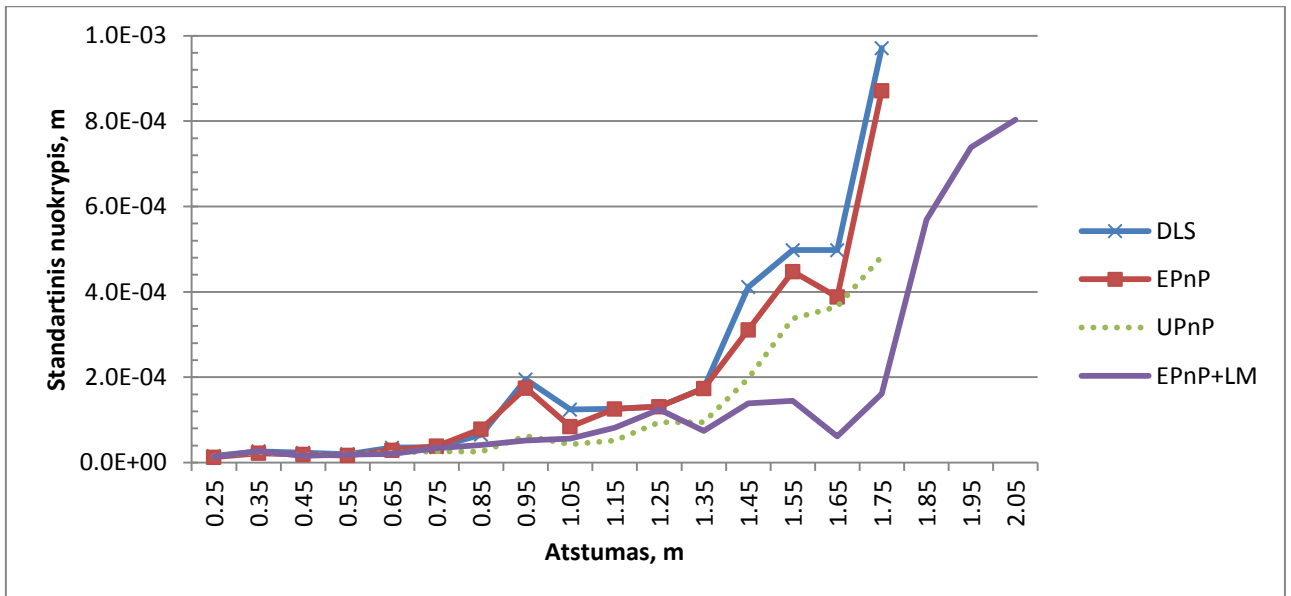
3.4 pav. Kiekvieno algoritmo reprojekcijos paklaida

Matoma, kad didžiausia reprojekcijos paklaida visiems algoritmams yra 25-45 cm atstumu. Taip yra todėl, kad mirksint šviesos diodams labai arti kameros gaunami ne visiškai tikslūs pikselių centrų įverčiai, o pritaikant objekto modelį prie šių įverčių gaunamas didelis euklidinis atstumas, tačiau tai turi labai mažai įtakos galutiniam poslinkio vektoriui ir pasisukimo matricai (žr. 3.3 ir 3.4 poskyrius).

Stabiliausia reprojekcijos paklaida DLS, EPnP ir UPnP algoritmams gaunama 0,55-1,65 atstumu nuo kameros – iki 1 pikselio. Jungtinis EPnP ir LM metodas šį atstumą prailgina iki 205 cm.

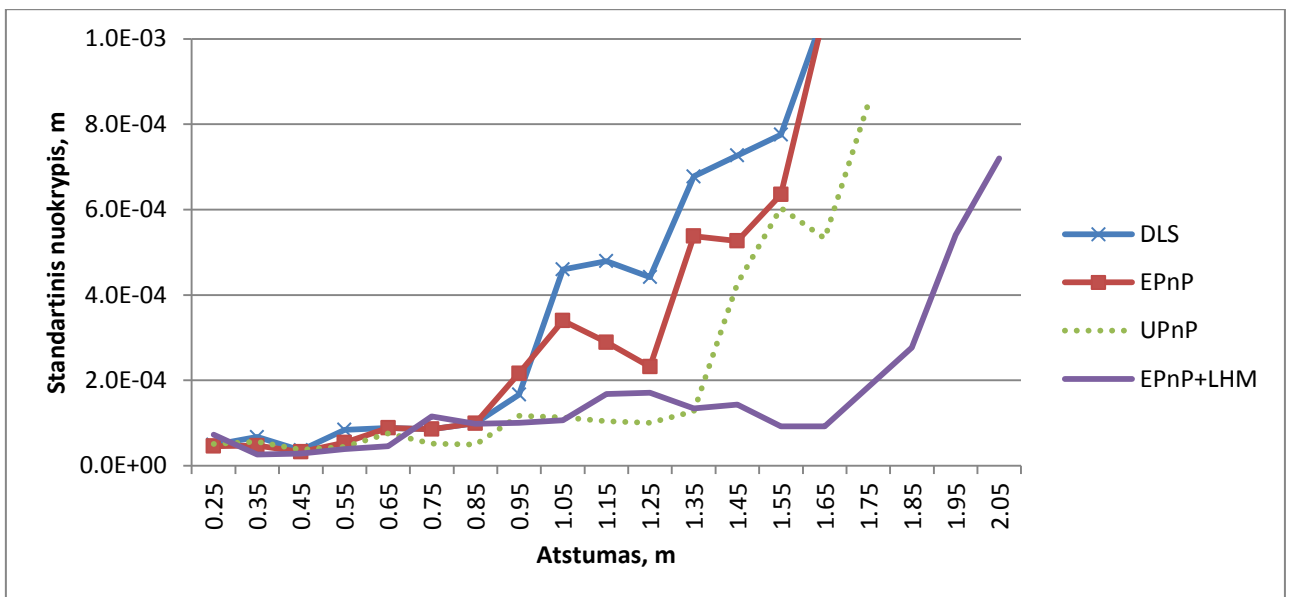
3.3 Standartiniai nuokrypiai X,Y ir Z ašyse

Toliau analizuojami neapibrėžties įverčiai – standartiniai nuokrypiai kiekvienam algoritmui, kiekvienu gautu atstumu.



3.5 pav. Standartiniai nuokrypiai X ašyje

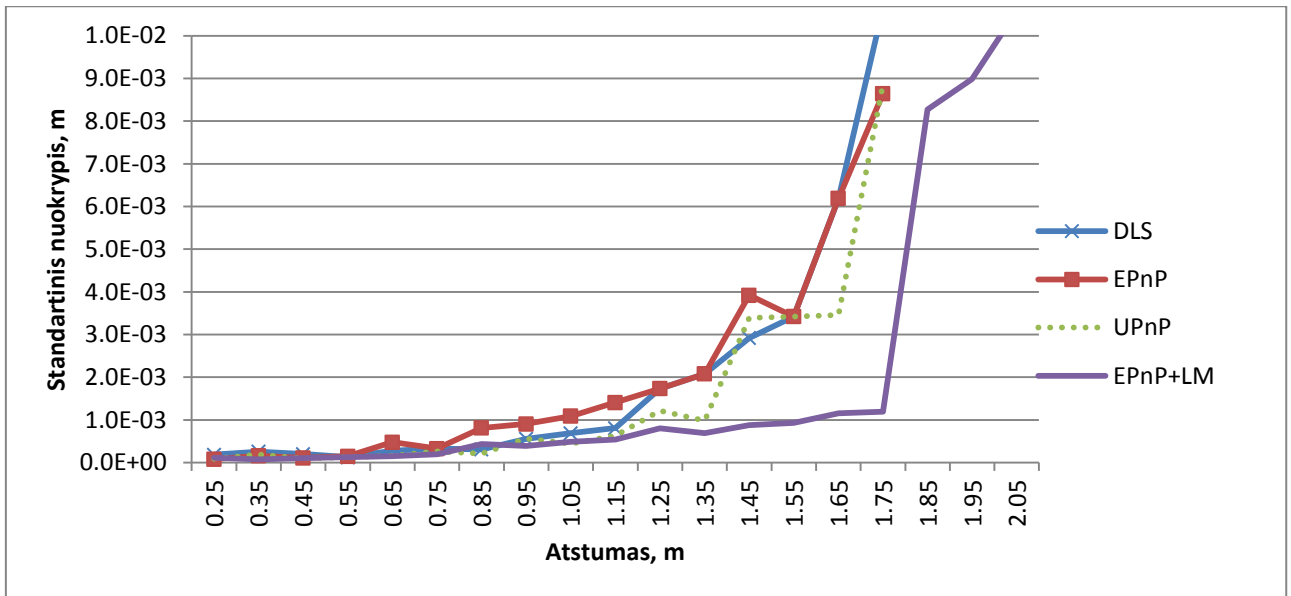
X ašyje neapibrėžtis mažiausia – visiems atstumams šis įvertis lieka mažesnis, nei 1 mm. EPnP ir DLS algoritmų standartiniai nuokrypiai priartėja prie 1 mm, kai atstumas iki kameros 1,65 m. UPnP standartinis nuokrypis šiame atstume 0,49 mm, o EPnP+LM – 0,18 mm. Pastarasis algoritmas X ašyje triukšmą išlaiko mažesnę, nei 1 mm dydžio net 2 metrų atstumu.



3.6 pav. Standartiniai nuokrypiai Y ašyje

Padėties nustatymo Y ašyje neapibrėžtis šiek tiek didesnė, nei X ašyje. Tik DLS ir EPnP 1,55 m. atstumu viršija 1 mm. standartinį nuokrypį. UPnP standartinis nuokrypis šiame atstume 0,6 mm, o EPnP+LM 0,1 mm. Pastarasis algoritmas didžiausiu atstumu – 2,05 m. išlaiko 0,73 mm. standartinį nuokrypį.

Toliau analizuojamas standartinis nuokrypis Z ašyje. Šiame grafike pasikeičia mastelis – maksimalus įvertis – 1 cm.



3.7 pav. Standartiniai nuokrypiai Z ašyje

Z ašies – atstumo iki kameros aptikimas algoritmams pasirodė sudėtingiausia užduotis. Visi algoritmai iki 1,05 m. turėjo mažesnę, nei 1 mm standartinę nuokrypį. EPnP, DLS ir UPnP neapibrėžtis stabiliai didėja iki 4 mm atstume 1,05-1,45 m., o po to staigiai išauga iki 1 cm. Tuo tarpu jungtinis EPnP ir LM algoritmo neapibrėžtis stabiliai didėja iki 1 mm iki 1,75 m. atstumo. Tada atstumo iki kameros neapibrėžtis staigiai išauga iki beveik 1 cm.

3.4 α , β ir γ pasisukimų standartiniai nuokrypiai

Skaičiuojant α , β ir γ pasisukimų neapibrėžtis pastebėta, kad kartais šios reikšmės labai stipriai išauga, nors, atrodo, taip neturėtų būti:

3.1 lentelė

Didžiulė α pasisukimo kampo neapibrėžtis

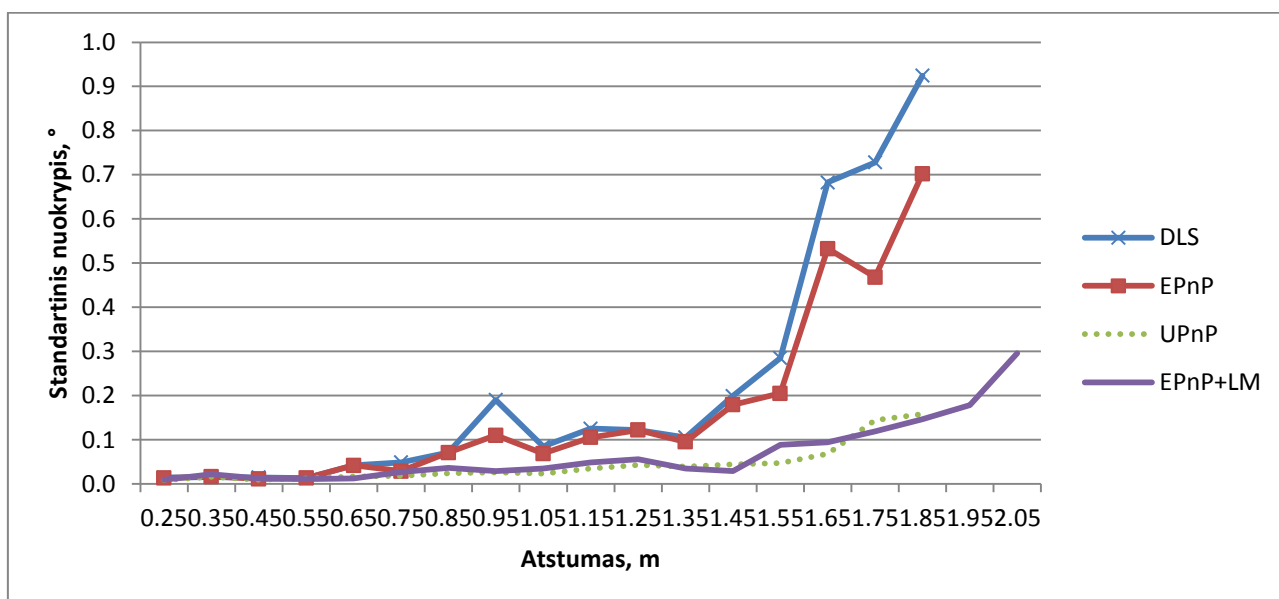
α							
Atstumas, m	0,75	0,85	0,95	1,05	1,15	1,25	1,35
Standartinis nuokrypis, °	0,028667	110,2128	140,5257	48,28731	173,4745	0,122158	117,8816
Vidurkis, °	-179,875	142,2191	112,278	173,2969	47,75297	179,3688	135,8939

Šiuo atveju tai buvo tik vieno kampo 360° persisukimas, nes jo įvertis buvo apie 180° . Pastebėjus tokius netikslumus rankiniu būdu ištaisomi duomenys. Šiuo atveju reikėjo visus α kampo pasisukimo įverčius, mažesnius už 0, padidinti 360° ir gauti teisingi duomenys:

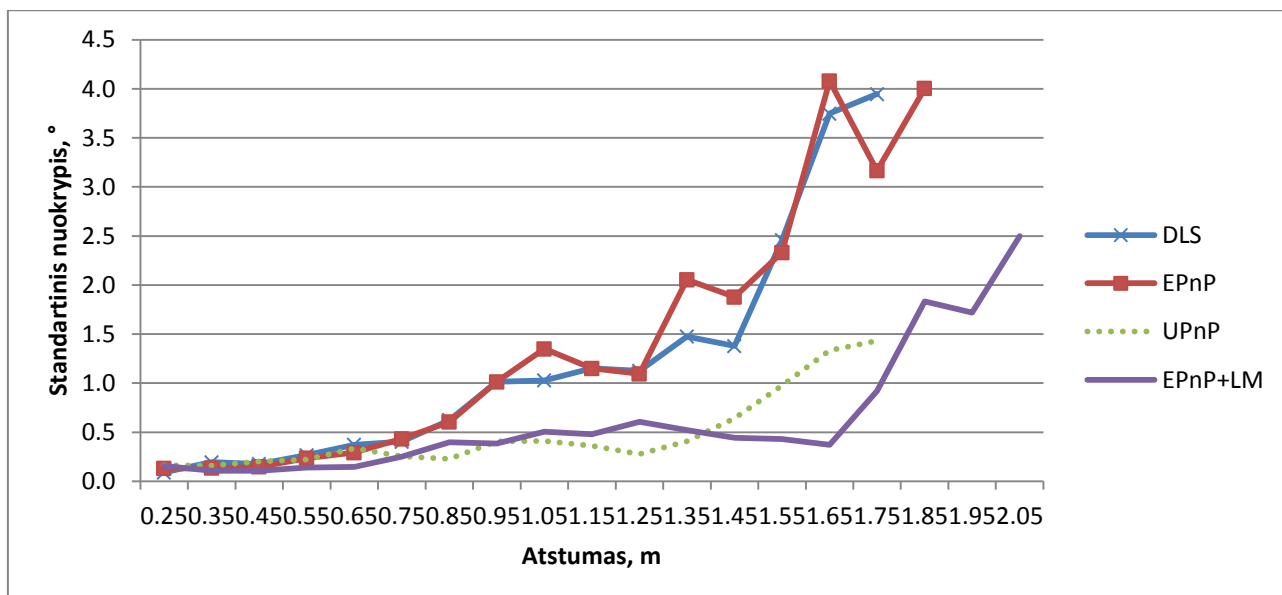
Ištaisyta α pasisukimo kampo neapibrėžtis

α							
Atstumas, m	0,75	0,85	0,95	1,05	1,15	1,25	1,35
Standartinis nuokrypis, °	0,028667	0,070847	0,110064	0,068731	0,105413	0,122158	0,095327
Vidurkis, °	180,1254	179,9191	179,878	179,8969	179,953	179,3688	179,8939

Tačiau kartais įvyksta visų Eulerio kampų pasisukimas. Taip įvyksta tada, kai antrojo pasukimo kampo reikšmė priartėja prie 90° arba -90° ir yra neišvengiama Eulerio kampų savybė ir eksperimentą tenka pakartoti.

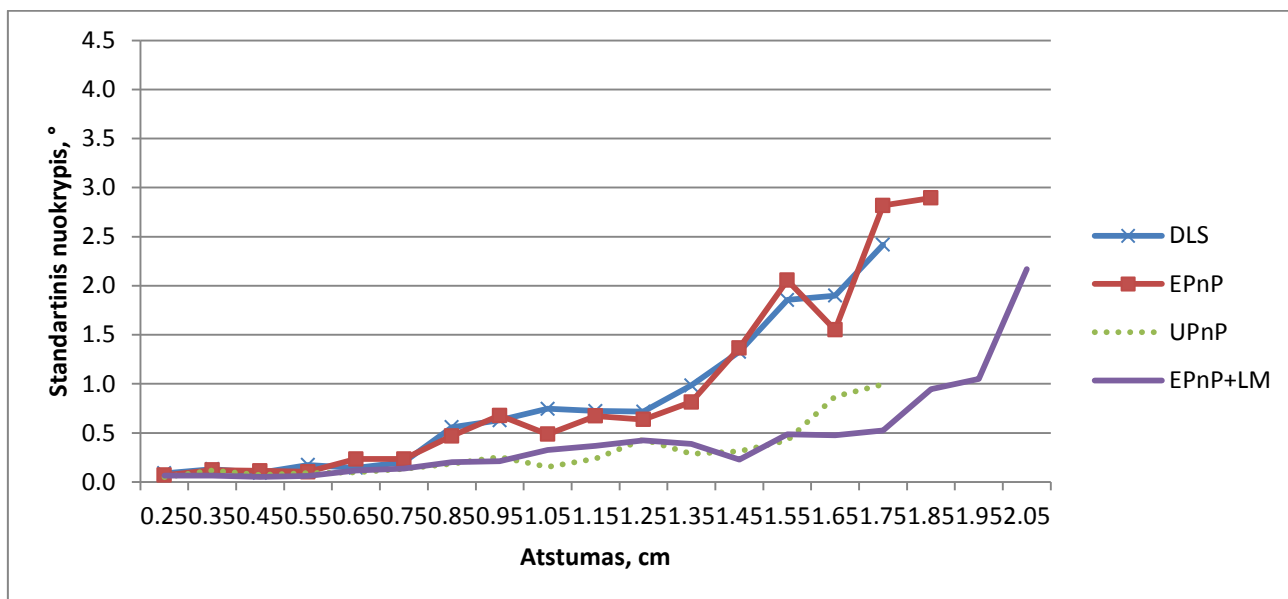
3.8 pav. α kampo pasisukimo standartiniai nuokrypiai

α kampo pasisukimo neapibrėžtis mažiausia – visiems algoritams iki 1,35 m. atstumu standartinis nuokrypis išlaikomas iki $0,1^\circ$ (išskyrus DLS 0,95 m. atstumu). UPnP šis įvertis pasiekiamas tik prie 1,65 m., o EPnP ir LM – 1,55 m. atstumu. DLS pasiekia $0,92^\circ$ paklaidą prie 1,85 atstumu, EPnP – $0,7^\circ$. Stabiliausiai veikiantys UPnP pasiekia $0,16^\circ$ 1,85 m., o EPnP – $0,3^\circ$ 2,05 m. atstumu.



3.9 pav. β kampo pasisukimo standartiniai nuokrypiai

β kampo pasisukimo neapibrėžtis didžiausia – EPnP ir DLS algoritmai 1° standartinį nuokrypį pasiekia vos 0,95 m. atstumu nuo kameros ir toliau auga iki 4° . VR šalmo padėti nustatant su UPnP algoritmu 1° neapibrėžtis pasiekama 1,55 m. atstumu, o jungtiniu EPnP ir LM metodu – 1,75 m.



3.10 pav. γ kampo pasisukimo standartiniai nuokrypiai

VR šalmo pasisukimo neapibrėžtis γ kampu taip pat didelė. 1° standartinis nuokrypis gaunamas EPnP ir DLS algoritmų 1,45 m. atstumu. Toks nuokrypis skaičiuojant UPnP algoritmu gaunamas tik 1,75 m., o EPnP ir LM – 1,85 m. atstumu. EPnP ir DLS atitinkamai kyla iki $2,8^\circ$ ir $2,4^\circ$ neapibrėžčių matuojant 1,75 m padėtyje. EPnP ir LM algoritmas 2,05 m. atstumu gauna $2,2^\circ$ neapibrėžtį.

Išvados ir siūlymai

- Virtualios realybės idėjos šiuo metu labai sparčiai plečiasi dėl aukštos rezoliucijos ekranų ir greitų bei patikimų jutiklių. VR šalmo sekimą erdvėje labai patogiu atlikti aktyvių žymeklių ir vaizdo kameros pagalba.
- Pasirinkti įverčiai algoritmų efektyvumui įvertinti – skaičiavimo laikas, reprojekcijos paklaida ir padėties neapibrėžtis 6 laisvės laipsniais.
- Sudarytas algoritmas gauti aktyvių žymeklių padėti kameros vaizde trunka apie 1,7–2,6 ms.
- Levenberg-Marquardt algoritmui greitai ir patikimai rasti globalų funkcijos minimumą reikalingi duomenys iš kito objekto padėties ieškojimo algoritmo.
- EPnP ir UPnP dėl trumpo skaičiavimo laiko puikiai tinka realaus laiko sistemoms – skaičiavimai vidutiniškai trunka apie 0,06 ms.
- Jei pradiniai padėčiai aptikti naudojamas EPnP metodas, o po to patikslinamas LM metodu – galima gauti spartesnę ir tikslesnę algoritmą iki 1,15 m atstumo. Didesniu atstumu LM reikia daugiau iteracijų ir jis tampa 0,01 – 0,02 ms (15-35%) lėtesnis, nei EPnP ir UPnP
- DLS algoritmo atlikimo laikas vidutiniškai truko apie 1 ms.
- Stabiliausia reprojekcijos paklaida DLS, EPnP ir UPnP algoritmams gaunama 0,55–1,65 atstumu nuo kameros – iki 1 pikselio. Jungtinis EPnP ir LM metodas šį atstumą prailgina iki 205 cm.
- Poslinkio vektorius neapibrėžtis X ašyje – mažiausia ir išlieka mažesnė, nei 1 mm visuose matavimuose visiems algoritmams.
- VR šalmo atstumą iki kameros tiksliai išmatuoti sunkiausia – Z ašyje neapibrėžtis didžiausia – stabiliai mažesnė, nei 1 mm išlieka iki 1,15 m visiems algoritmams išskyrus EPnP ir LM jungtinį – iki 1,75 m. Nuo šių atstumų atitinkamiems algoritmams neapibrėžtis labai sparčiai pradeda augti – iki 1 cm maksimaliu išmatuojamu atstumu.
- α kampo pasisukimo neapibrėžtis mažiausia – visiems algoritmams iki 1,35 m. atstumu standartinis nuokrypis išlaikomas iki $0,1^\circ$ (išskyrus DLS 0,95 m. atstumu). UPnP šis įvertis pasiekiamas tik prie 1,65 m., o EPnP ir LM – 1,55 m.
- β ir γ kampo pasisukimo neapibrėžtis didžiausia. EPnP ir DLS algoritmai 1° standartinį nuokrypį pasiekia 0,95 m atstumu. Didinant atstumą sparčiai didėja šie nuokrypiai.
- Tyrimo metu pastebėti kartais pasitaikantys neteisingų taškų atpažinimai dažniausia susidarantys dėl atspindžio nuo šalia esančių objektų. Reikėtų gaunamų taškų padėčių patikimumą statistiškai tikrinti. Siūloma schema – RANSAC[20].

- Sistemos triukšmui pašalinti ir reakcijos į judesį parametrams pagerinti reikėtų jutiklių apjungimo – panaudoti inercinius sensorius, integruotus VR šalme ir darniai sujungti su matomų kamera vaizdų duomenimis.
- Sukurtas algoritmas galėtų būti naudojamas projektuojant analogišką įrenginį, arba plečiant šio galimybes – aptinkant keletą VR šalmų arba papildomų žmogaus–kompiuterio sąsajos įrenginių.

Literatūra

1. Schuemie J. M., Straaten P. Iš Cyber Psychology & Behaviour [interaktyvus]. 2001, vasaris [žiūrėta 2015 02 25]. Prieiga per internetą: <http://graphics.tudelft.nl/~vrphobia/surveypub.pdf>
2. Rizzo. S. (2015). Virtual Reality Exposure Therapy 2005-Present. Technologies [žiūrėta 2015 02 25]. Prieiga per internetą: http://ict.usc.edu/wp-content/uploads/overviews/Post%20Traumatic%20Stress%20Disorder_Overview.pdf
3. Abrash M. (2014). What VR could, should, and almost certainly will be within two years [Žiūrėta 2015-01-11]. Prieiga per internetą: <http://media.steampowered.com/apps/abrashblog/Abrash%20Dev%20Days%202014.pdf>
4. LaValle M. S. Head Tracking for the Oculus Rift/S. M. LaValle, A. Yershova., M. Katsev, M. Antonov// 2014 IEEE International Conference on Robotics & Automation (ICRA).
5. Szeliski R. Computer Vision: Algorithms and applications [interaktyvus]. September 3, 2010 draft. [žiūrėta 2015-05-15] Prieiga per internetą: <http://szeliski.org/Book/>
6. Staugaitis Ž. Geometrinių transformacijų sistema//Magistro darbas. 2006, Kaunas
7. Borcovas E. Galvos pozicijos nustatymas ekrano atžvilgiu, panaudojant ant galvos pritvirtintą kamerą//Magistro darbas. 2012, Šiauliai
8. 3D Tracking - Javier Barandiaran Martirena [Žiūrėta 2015-01-11]. Prieiga per internetą: <https://sites.google.com/site/jbarandiaran/3dtracking>
9. 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.
10. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications/ Khoshelham K., Elberink S.// *Sensors*. 2012; 12(2), p.1437-1454.
11. Michael M. Implementation of a Low Cost Marker Based Infrared Optical Tracking System [Žiūrėta 2015-05-11]. Prieiga per internetą: http://publik.tuwien.ac.at/files/PubDat_210294.pdf
12. Petersen T. A Comparison of 2D-3D Pose Estimation Methods.//Master thesis. 2008, Aalborg University
13. Baker K. Singular Value Decomposition Tutorial [interaktyvus]. March 29, 2005 (Revised January 14, 2013). [žiūrėta 2015-05-15] https://www.ling.ohio-state.edu/~kbaker/pubs/Singular_Value_Decomposition_Tutorial.pdf
14. A Direct Least-Squares (DLS) Method for PnP/Hesch A. J., Roumeliotis I. S.// ISSN. 1550-5499. 2011, Nr. 6-13 Nov, p.383 - 390

15. EPnP: An Accurate $O(n)$ Solution to the PnP Problem/ Vincent L, Francesc M., Pascal F.// Int J Comput Vis. 2008. [Žiūrėta 2015-01-11]. Prieiga per internetą:
<http://cvlabwww.epfl.ch/~lepetit/papers/lepetit_ijcv08.pdf>
16. UPnP: An Optimal $O(n)$ Solution to the Absolute Pose Problem with Universal Applicability/ Kneip L., Li H., Seo Y.// 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I. 2014, nr. 6-12 p.127-142
17. 1/3-Inch Wide-VGA CMOS Digital Image Sensor [Žiūrėta 2015-01-13]. Prieiga per internetą:
<<http://www.aplina.com/assets/downloadDocument.do?id=406>>
18. Decentering Distortion of Lenses/ Brown C. D.// Annual Convention of the American Society of Photometry, 1965.
19. Diebel J. Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors [interaktyvus]. 20 October 2006. [Žiūrėta 2015-05-05]. Prieiga per internetą:
<<http://www.swarthmore.edu/NatSci/mzucker1/e27/diebel2006attitude.pdf>>
20. Derpanis G. K. Overview of the RANSAC Algorithm [interaktyvus]. 13 May 2010. [Žiūrėta 2015-05-25]. Prieiga per internetą:
<http://www.cse.yorku.ca/~kosta/CompVis_Notes/ransac.pdf>