

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMATIKOS STUDIJŲ PROGRAMA

RIMVYDAS KLEMENTAVIČIUS

RANKOS SKELETO JUDĖJIMO REKONSTRUKCIJA IŠ
MAŽO DETALUMO VIDEO VAIZDŲ METODAS

Magistro darbas

Vadovas
doc. dr. A. Ostreika

KAUNAS, 2014

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMATIKOS STUDIJŲ PROGRAMA

RIMVYDAS KLEMENTAVIČIUS

**RANKOS SKELETO JUDĖJIMO REKONSTRUKCIJA IŠ
MAŽO DETALUMO VIDEO VAIZDŲ METODAS**

Magistro darbas

Vadovas:
doc. dr. A. Ostreika
Data: 2014-05-
Parašas:

Recenzentas:

Data: 2014-05-
Parašas:

Atliko:
IFM-2/1 gr. studentas
R. Klementavičius
Data: 2014-05-
Parašas:

KAUNAS, 2014

AUTORIŲ GARANTINIS RAŠTAS DĖL PATEIKIAMO KŪRINIO

20.. - - d.

Kaunas

Autorius, _____
(vardas, pavardė)

patvirtina, kad Kauno technologijos universitetui pateiktas baigiamasis bakalauro (magistro) darbas (toliau vadinama – Kūriny) Rankos skeleto judėjimo rekonstrukcija iš mažo detalumo video vaizdų metodas pagal Lietuvos Respublikos autorių ir gretutinių teisių įstatymą yra originalus ir užtikrina, kad

- 1) jį sukūrė ir parašė Kūrinyje įvardyti autoriai;
- 2) Kūriny nėra ir nebus įteiktas kitoms institucijoms (universitetams) (tiek lietuvių, tiek užsienio kalba);
- 3) Kūrinyje nėra teiginių, neatitinkančių tikrovės, ar medžiagos, kuri galėtų pažeisti kito fizinio ar juridinio asmens intelektualinės nuosavybės teises, leidėjų bei finansuotojų reikalavimus ir sąlygas;
- 4) visi Kūrinyje naudojami šaltiniai yra cituojami (su nuoroda į pirminį šaltinį ir autorių);
- 5) neprieštarauja dėl Kūrinio platinimo visomis oficialiomis sklaidos priemonėmis.
- 6) atlygins Kauno technologijos universitetui ir tretiesiems asmenims žalą ir nuostolius, atsiradusius dėl pažeidimų, susijusių su aukščiau išvardintų Autorių garantijų nesilaikymu;
- 7) Autoriai už šiame rašte pateiktos informacijos teisingumą atsako Lietuvos Respublikos įstatymų nustatyta tvarka.

Autorius

(vardas, pavardė)

(parašas)

Rankos skeleto judėjimo rekonstrukcija iš mažo detalumo video vaizdų metodas

Santrauka

Skeleto judėjimo rekonstrukcijos iš video vaizdų metodas yra plačiai taikomas metodas. Toks metodas naudojamas kompiuteriniuose žaidimuose animuoti veikėjų judesiams, tiek interaktyviai žaidimo sąsajai kurti kai žaidėjas gali valdyti virtualų veikėją judėdamas pats. Taip pat skeleto judesio atkūrimas naudojamas ir kino pramonėje, kuriant specialiuosius efektus.

Šiame darbe sukurtas rankos skeleto judėjimo rekonstrukcijos iš mažos rezoliucijos vaizdų metodas nereikalauja didelių ekonominių išteklių lyginant su kitais skeleto judesio rekonstravimo metodais iš video vaizdų. Iškėlus tokio metodo veikimui būtinus reikalavimus ir tikslus, sukurta sistema, veikianti *OpenCV* vaizdo apdorojimo bibliotekos pagrindu, bei panaudoti kuo paprastesni komponentai tokios sistemos korektiškam funkcionavimui. Realizavus sistemą, ištirtas metodo tikslumas, greitumas, bei efektyvumas veikiant įvairiomis sistemoje naudojamų kamerų rezoliucijomis. Paskaičiavus ir palyginus gautus darbo rezultatus, suformuluojama metodo naudojimo patarimai. Šis darbas įrodė, jog naudojant internetinių kamerų porą veikiančias 480x320 rezoliucija, bei pasyvius žymeklius, galima rekonstruoti rankos skeleto judėjimą efektyviai ir kokybiškai su labai ribotais ištekliais.

Raktiniai žodžiai

Judėjimo rekonstravimas, judesio atkūrimo sistemos, *OpenCV* biblioteka, ekonominis sprendimas.

Hand skeletal movement reconstruction from low resolution video frames method

Summary

Skeleton movement reconstruction from low quality videos is a widely used method, there are many types of these methods. This kind of method is being used in video games to recreate characters movements or to create interactive gameplay when the player can control game character just by moving himself. Skeleton movement reconstruction is also being used in cinema industry, to create special effects.

The skeleton reconstruction method that requires low quality video created in this work, doesn't require expensive resources compared to the other motion capture methods. By setting up requirements and goals of how to create such method, a system was created that operates by OpenCV motion capture library basis and uses low cost components for its proper functioning. After calculating and measuring work results, in summary it is save to say, that this work have proven right, that it is indeed possible to reconstruct hand movement effectively and qualitatively by using a couple of low cost 480x320 resolution webcams.

Keywords

Motion capture, motion capture systems, OpenCV library, economic decision.

TURINYS

Terminų ir Santrumpų Žodynas	8
Lentelių sąrašas	9
Paveikslų sąrašas	10
Įvadas	11
1.1 Darbo tikslas ir uždaviniai	11
1.2 Tyrimo problema ir objektas	11
1.3 Tyrimo aktualumas ir naujumas	11
2 Skeletinio judėjimo rekonstrukcijos iš mažo detalumo video vaizdų metodo analizė	12
2.1 Analizės tikslas	12
2.2 Judesio sekimo sistemos	12
2.3 Judesio sekimo sistemų naudojimas	12
2.4 Žymekliai	13
2.5 Kameros	14
2.5.1 Kamrų kalibravimas	14
2.5.2 Kameros modelis ir projekcijos matrica	16
2.5.3 Populiariausi kamerų kalibravimo metodai	16
2.5.4 Vaizdo pradinis apdorojimas	17
2.6 Erdvinio taško rekonstrukcija iš 2-jų kameros vaizdų	19
2.7 Judesio sekimo metodologija	21
2.7.1 Žymeklių išskyrimas	21
2.7.2 Žymeklių rekonstrukcija erdvėje	23
2.8 Skeleto rekonstrukcija	23
2.8.1 Skeleto judėjimas	24
2.8.2 Virtualaus skeleto judėjimas ir skeleto medžio sudarymas	25
2.9 Objektų sekimo algoritmai	26
2.10 Bibliotekos	27
2.11 Esamų sprendimų analizė	27
2.12 Duomenų perdavimas tarp sistemų	28
2.13 Analizės išvados	29
3 Metodo sistemos reikalavimų specifikacija	30
3.1 Funkciniai reikalavimai	30
3.2 Nefunkciniai reikalavimai	31
3.3 Techniniai reikalavimai	31
4 Metodą realizuojančios sistemos kūrimas	32
4.1 Metodo pagrindimas	32
4.2 Sistemos architektūra	32
4.2.1 Reikalavimų analizė	32

4.2.2	Sistemos loginė architektūra	32
4.3	Sistemos detalus modelis.....	34
4.4	Sistemos realizacijos modelis ir diegimas	34
5	Metodo tyrimas ir rezultatų analizė	36
5.1	Sukurto metodo tyrimo metodologija.....	36
5.2	Metodo eksperimentinio tyrimo aprašas	36
5.3	Metodo tyrimui naudojamos techninės ir programinės įrangos aprašas	36
5.4	Tyrimo rezultatai	38
5.4.1	Metodo erdvinio taško rekonstrukcijos laiko priklausomybė nuo perduodamos skiriamosios gebos	38
5.4.2	Metodo erdvinio taško rekonstrukcijos klaidų kiekio priklausomybė nuo perduodamos skiriamosios gebos	39
5.4.3	Metodo toliausio taško nustatymo priklausomybė nuo perduodamo skiriamosios gebos 40	
5.4.4	Metodo rankos skeleto judesio rekonstrukcijos greitaveikos ir tikslumo priklausomybė nuo judesio atlikimo laiko.....	40
5.4.5	Metodo nuotolio nustatymas lyginant realų nuotolį su apskaičiuotu.....	41
5.5	Metodo taikymo rekomendacijos	41
5.5.1	Metodo sistemos kūrimo ekonominiai skaičiavimai.....	42
6	Išvados	43
7	Naudota literatūra.....	44

TERMINŲ IR SANTRUMPŲ ŽODYNAS

MOCAP	(angl. k <i>Motion Capture</i>) judesio pagavimo sistema
Off-line	Atsijungus nuo tinklo režimas
NBA 2K12	Trimatis kompiuterinis krepšinio žaidimas
OpenCV	Tai vaizdo apdorojimo biblioteka
3D	Trijų dimensijų kitaip trimatis
2D	Dvimatis
Monokulerinis	Vienas vaizdas
USB	Jungtis duomenims perduoti
FireWire	Jungtis duomenims perduoti
Stereo	Du vaizdai arba garsai
Pikselis	Kvadratinis langelis turintis savyje nustatyta reikšmę kuri reiškia spalvą
Gradientas	Difiriancialinis, skaliarinį lauką atvaizduojantis į vektorinį lauką taip, kad kiekvienas vektorinio lauko vektorius būtų nukreiptas skaliarinio lauko reikšmių didėjimo kryptimi, o jo modulis būtų lygus kryptinei išvestinei šiame taške
Blender	Trimatės grafikos ir animacijos kūrimo programa
Wii	Žaidimų konsolė
RGB	Raudona žalia mėlyna
CPU	Procesorius
GPU	Grafinės plokštės procesorius
HDD	Kietasis diskas
DDR3	Operatyviosios atminties rūšis
RAM	Operatyvioji atmintis
C++	Programavimo kalba kilus iš C kalbos
Gimnasto poza	Poza kai žmogus laiko rankas ištiesiąs į šalis
RGB	(angl. k. <i>red, green, blue</i>) slapvos kodavimo standartas
HSV	(angl. k. <i>hue, saturation, value</i>) tai spalvos kodavimo standartas
UML	(angl. k. <i>Unified Modeling Language</i>) kitaip tai vieninga modeliavimo kalba
TCP IP	Duomenų perdavimo protokolas
UDP IP	Duomenų perdavimo protokolas
Eclipse	Programavimo aplinka
Windows 7/XP	Operacinė sistema
Raspberry Pi	Pigus kompiuteris, kurio kaina siekia 50 \$

LENTELIŲ SĄRAŠAS

1 lentelė. Objektų sekimo algoritmai.....	26
2 lentelė. Bibliotekų palyginimas.....	27
3 lentelė. Esami sprendimai.....	27
4 lentelė. TCP IP ir UDP IP skirtumai.....	28
5 lentelė. Judesio sekimo sistemos funkcijų aprašas.....	30
6 lentelė. Klasų diagramos aprašymas.....	34
7 lentelė. Personalinio kompiuterio parametrai.....	37
8 lentelė. Internetinės kameros parametrai.....	37
9 lentelė. Trimačio taško priklausomybė nuo laiko.....	39
10 lentelė. Pamestų žymeklių kiekis per 100 iteracijų naudojant skirtingas skiriamąsias gebas.....	39
11 lentelė. Atkuriamo taško atstumo priklausomybė nuo skiriamosios gebos.....	40
12 lentelė. Metodo rankos skeleto judesio rekonstrukcijos greitaveikos ir tikslumo vertinimas.....	40
13 lentelė. Nuotolio nuo kameros matavimas.....	41
14 lentelė. Sistemos kaina.....	42
15 lentelė. Teorinė judesio sekimo sistemos kaina.....	42

PAVEIKSLŲ SĄRAŠAS

1 pav. Žymekliais grįstų sistemų tipai (a - realaus laiko, b - ne realaus laiko)	12
2 pav. Avatar filmo filmavimas.....	13
3 pav. Pasyvūs žymekliai.....	13
4 pav. Aktyvių žymeklių kostiumas.....	14
5 pav. 3D objektu paremto kalibravimo šablonas.....	15
6 pav. 2D kalibravimo šablonas.....	15
7 pav. Projekcijos matricos sudarymo principas.....	16
8 pav. Tsai kameros projekcijos modelis su perspektyvine projekcija.....	17
9 pav. 3x3 glotninimo filtrai.....	18
10 pav. Vaizdo ryškinimo kaukė.....	19
11 pav. 3D taško rekonstrukcija naudojant 2 kameras.....	19
12 pav. Polinė geometrija.....	20
13 pav. Intensyvumo segmentavimas.....	21
14 pav. Linijinis identifikavimas.....	22
15 pav. Identifikavimas pagal kampus.....	22
16 pav. Spalvinis žymeklių atpažinimas.....	22
17 pav. Gimnasto poza.....	23
18 pav. Skeleto modelio pavyzdys.....	23
19 pav. Spyruoklinis modelis.....	24
20 pav. Kaulų judėjimas naudojant atvirkštinės kinetikos metodą.....	24
21 pav. Tiesinė kinetika.....	25
22 pav. Skeleto kaulų medžio sudarymas.....	26
23 pav. Panaudos atvejų diagrama.....	30
24 pav. Sistemos loginė architektūra.....	32
25 pav. Vartotojo elgsenos diagrama.....	33
26 pav. Būsenų diagrama.....	33
27 pav. Supaprastinta klasių diagrama.....	34
28 pav. Metodo naudojimas: (a) aktorius užsidėjęs spalvinius žymeklius, (b) - dviejų internetinių kamerų sistema, (c) sistema apdorojanti vaizdo medžiagą.....	35
29 pav. Metodo realizacijos modelis.....	35
30 pav. Pasyvūs žymekliai.....	37
31 pav. Kamrų vaizdų suvienodinimas x ašyje.....	38
32 pav. Realizuoto metodo veikimo demonstravimas.....	38
33 pav. Trimačio taško rekonstrukcijos priklausomybė nuo laiko.....	39
34 pav. Pamestų žymeklių kiekis per 100 trimačio taško atkūrimo iteracijų.....	40
35 pav. Judesio rekonstrukcijos greitaveikos ir tikslumo vertinimas.....	41

IVADAS

Judesio rekonstravimas (angl. k. *motion capture*), tai procesas kurio metu įrašomi žmogaus ar objektų judesiai. Šis procesas naudojamas kariuomenėje, pramogose, sporte, robotikoje. Filmų ir video žaidimų kūrime, dažniausiai naudojamas žmonių judesių informacijos įrašymas tam, kad ši informacija būtų panaudota virtualių personažų animacijai kurti. Toks animacijos kūrimo būdas taupo animatorių laiką, nes nebereikia judesių kurti rankomis.

Tačiau judesio rekonstravimo sistemos, galinčios rekonstruoti skeleto judesį iš video vaizdų, reikalauja sparčių ją sudarančių komponentų kiekio, tuo tarpu spartūs komponentai reikalauja didelio biudžeto. Kuo sudėtingesnė ir spartesnė sistema tuo didesnė tikimybė rekonstruoti tikroviškesnį skeleto judesį.

Šiuo metu rinkoje yra nevienas sprendimas galintis rekonstruoti skeleto judesį iš video vaizdo, tačiau visi sprendimai reikalauja sudėtingos ir sparčios techninės įrangos. Todėl, šiuo magistriniu darbu bus bandoma sukurti metodą, kurį būtų galima plačiai taikyti kompiuterinėje regoje ir erdvinėse interaktyviosiose pramogose, naudojant kuo paprastesnius ir nesudėtingesnius komponentus metodo įgyvendinimui, tuo pačiu siekiant sukurti ir kuo ekonomiškesnį variantą.

1.1 Darbo tikslas ir uždaviniai

Darbo tikslas

Šio darbo tikslas – sukurti metodą galintį rekonstruoti rankos skeleto judėjimą iš mažos raiškos video vaizdų.

Darbo uždaviniai

1. Surinkti ir išanalizuoti informaciją apie judesio atpažinimo algoritmus ir skeleto rekonstrukcijos metodus siekiant išsiaiškinti algoritmų ir metodų trūkumus, bei privalumus.
2. Suformuluoti reikalavimus remiantis analizės išvadomis kuriamam metodui.
3. Sukurti metodą gebantį efektyviai sekti ir rekonstruoti rankos skeleto judėjimą iš video vaizdų.
4. Realizuoti metodą pasinaudojus judesio rekonstravimo sistemos komponentais.
5. Ištirti metodo greitaveiką ir pritaikymą.

1.2 Tyrimo problema ir objektas

Tyrimo problema, jog judesio sekimo ir skeleto judėjimo rekonstravime iš video vaizdų naudojamų metodų tikslumas priklauso nuo komponentų brangumo.

Tyrimo objektas yra rankos skeleto judėjimo rekonstrukcijos iš mažos rezoliucijos video vaizdų metodas. Darbe planuojama atlikti judesio sekimo algoritmų ir skeleto atkūrimo iš video vaizdų metodų analizę, bei sukurti efektyvų metodą galintį atkurti rankos skeleto judėjimą iš video vaizdų.

1.3 Tyrimo aktualumas ir naujumas

Rankos skeleto judėjimo rekonstrukcijos iš mažos raiškos video vaizdų metodą bus galima pritaikyti praktikoje kaip nesudėtingą ir ekonomišką alternatyvą sudėtingoms judesio atkūrimo sistemoms. Taip pat metodas gali būti plačiai taikomas, ne tik rekonstruojant rankos skeleto judesį, bet ir atkuriant kitų kaulinių, tiek robotų sistemų judėjimą.

Tyrimo naujumas, jog bandoma pasiekti geriausią įmanomą tikslumą naudojant kiek įmanoma ekonomiškesnę įrangą kuriant rankos skeleto judėjimo rekonstrukcijos iš mažos raiškos video vaizdų metodą.

2 SKELETINIO JUDĖJIMO REKONSTRUKCIJOS IŠ MAŽO DETALUMO VIDEO VAIZDŲ METODO ANALIZĖ

2.1 Analizės tikslas

Analizės metu bus bandoma nustatyti efektyvius skeleto judėjimo rekonstrukcijos metodus bei algoritmus, patogiausias bibliotekas vaizdo apdorojimui. Išsiaiškinti skeleto judėjimo principus. Taip pat nustatyti judesio atkūrimo sistemų aparatinės įrangos naudojimo aspektus, specifikacijas, rūšis ir panaudojimą. Pagal atliktą analizę bus kuriami algoritmai ir metodai rankos skeleto judesio atkūrimui pasitelkiant vaizdo apdorojimo sistemos elementus.

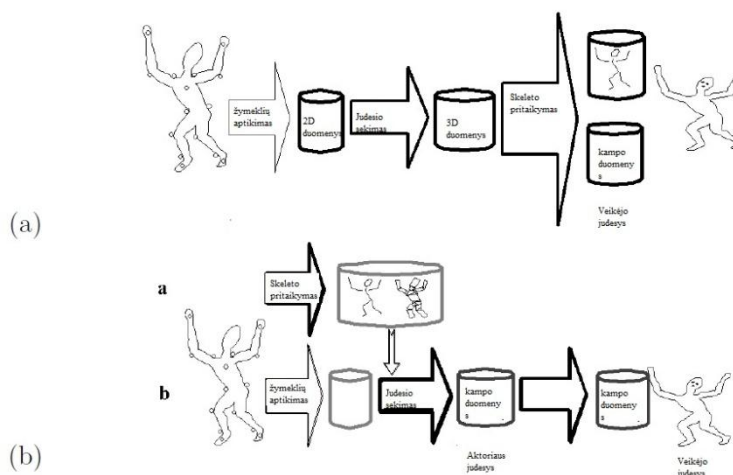
2.2 Judesio sekimo sistemos

Judesio sekimo sistemos gebančios atkurti skeleto judėjimą būna: magnetinės, mechaninės ir optinės. Magnetinės sistemos naudoja elektromagnetinius sensorius, kurie realiu laiku perduoda savo padėties erdvėje duomenis registruojančiai ir analizuojančiai sistemai. Tačiau tokios sistemos varžo judesius. Panašiai veikia ir mechaninės sistemos, bet vietoje magnetinių sensorių, naudojami kostiumai su integruotais mechaniniais sensoriais, kurie registruoja judesius taip pat realiu laiku perduoda sensorių informaciją analizuojančiai sistemai. Optinės sistemos paremtos vaizdų analizavimo metodais. Tokios sistemos dažniausiai būna labai tikslios ir nevaržančios judėjimo laisvės, galinčios apdoroti ne vieną, o kelis sistemos naudotojus, tačiau reikalingi dideli skaičiavimo ir vaizdo apdorojimo resursai.

Optinės sistemos dar skirstomos pagal naudojamą vaizdų skaičių vienos vaizdo (angl. k. *monocular*) arba daugelio vaizdų, bei pagal žymeklių naudojimą (žymekliais grįstos ir be žymeklių). Vieno vaizdo sistemos naudojama tik viena kamera, o daugelio vaizdų sistemos naudojamos kelios kameros. Žymekliais grįstos sistemos matuoja trajektorijas tarp atžymėtų ant kūno taškų, o be žymeklių – matuoja judesio parametrus iš gauto silueto[1].

Taip pat žymekliais grįstos sistemos gali būti dviejų klasių (1 pav.):

- Realaus laiko judesio atkūrimo sistemos 1 pav. (a) - tokios sistemos realiu laiku gali valdyti virtualų personažą ir atkartoti aktorius judesius. Šios sistemos dažniausiai būna mechaninės, tačiau labai greitos.
- Ne realaus laiko (angl. k. *off-line*) judesio atkūrimo sistemos 1 pav. (b) - tokios sistemos labiau pritaikytos kai reikia atlikti sudėtingas animacijas, remiasi kelių kamerų vaizdais.



1 pav. Žymekliais grįstų sistemų tipai (a - realaus laiko, b - ne realaus laiko)

2.3 Judesio sekimo sistemų naudojimas

Dažniausiai judesio sekimo technologijos yra naudojamos animuojant virtualius personažus arba žaidžiant interaktyvius žaidimus. Tokių technologijų naudojimas animuojant virtualius

personažus žymiai pagreitina animacijos kūrėjų darbą, suteikia personažams tikroviškesnius judesius. Žaidimuose padidėja interaktyvumas tarp žaidėjo ir žaidimo.

Kino pramonėje, judesio sekimo sistemos naudojamas gana seniai. Geriausi to pavyzdžiai, tai filmai *Avatar* (2 pav.), *Tron:Legacy*, *The Lord of the Rings* ir kt.



2 pav. Avatar filmo filmavimas

Taip pat tokie žaidimai kaip *NBA 2K14*, *World of Warcraft*, *Madden NFL* ir žinoma *FIFA 2014* neatsiejami be tikroviškos veikėjų animacijos[2].

2.4 Žymekliai

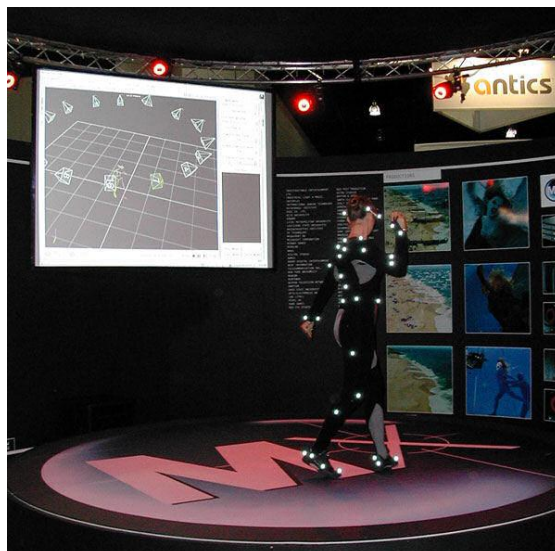
Optinėse judesio sekimo sistemose dažniausia naudojami pasyvūs ir aktyvūs žymekliai [3]:

Pasyvūs žymekliai - tai atspindintys šviesą. Paprasčiausias pavyzdys yra atšvaitas (3 pav.), šie žymekliai atspindi šviesą į kamerą. Tokių žymeklių pranašumas lyginant su kitais rinkoje esančiais žymekliais toks, kad jie neriboja judėjimo laisvės, bei yra pigūs ir lengvai pagaminami [4].



3 pav. Pasyvūs žymekliai

Aktyvūs žymekliai naudoja LED diodus, kuriuos aptinka kamera jos matymo lauke. Kiekvienas aktyvus žymeklis sugeba šviesti unikaliu dažniu, tokiu būdu galima lengviau identifikuoti žymeklius pagal jų švietimo dažnį. Tačiau tokių žymeklių (4 pav.) vieno vieneto kaina yra žymiai didesnė, nei pasyvių žymeklių.



4 pav. Aktyvių žymeklių kostiumas

Norint, kad judesio sekimo sistema veiktų patikimai reikia taisyklingai išdėstyti žymeklius. Žymekliai turėtų padengti [5] pagrindinius aktoriaus sąnarius - kaulų sujungimų vietas. Rekomenduojama, kad stuburas turėtų keturis žymeklius. Kaklą, pečius, rankas, delnus, šlaunis, blauzdas ir pėdas geriausiai pažymėti naudojant po tris žymeklius kiekvienai kūno elementui.

2.5 Kameros

Aukštos klasės judesio sekimui reikalingos brangios ir aukštos kokybės kameros galinčios fiksuoti aukštos rezoliucijos vaizdus didesne, nei 60 kadrų per sekundę sparta. Kuo aukštesnės raiškos kamera tuo tiksliau galima nustatyti net pačius menkiausius judesius. Kuo greitesnė kamera tuo geriau yra pašalinamas suliejimas (angl. k. *blur*) tarp kadrų.

Judesio sekimo kameros dažniausiai turi įmontuotą blyksinčią infraraudonąją šviesą. Taip lengviau išskirti žymeklius iš aplinkos, nes pašalinami tam tikri atspindžiai[4].

Kameros su aparatine įranga turi bendrauti per *USB* arba *FireWire* jungtis, užtikrinant patikimą ir greitą duomenų perdavimą [6].

Taip pat kameras geriausiai yra išdėstyti ratu aplink sceną, bei pastatyti ant stovų [1].

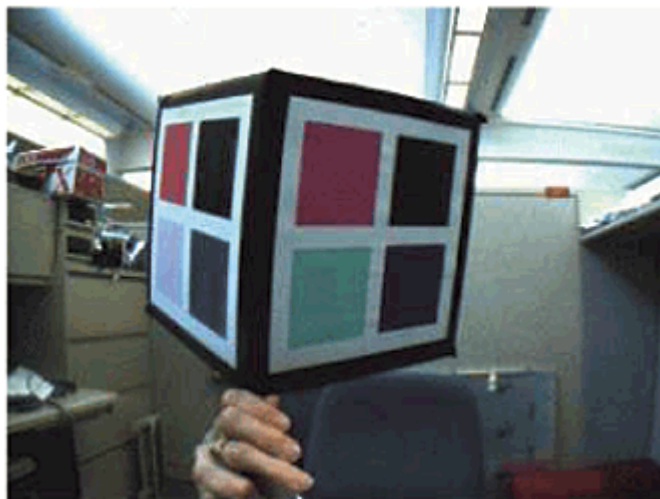
2.5.1 Kamerų kalibravimas

Kamerų kalibravimas nustato vaizdo taškų (toliau pikselių) ir tikro pasaulio dimensių sąveiką [5], taip pat ištaiso kameros lęšio vaizdo iškraipymus. Dažniausiai kalibruojant kameras gaunami tokie parametrai kaip kameros židinio ilgis, vaizdo formato dydis, lęšio iškraipymas, centrinis taškas. Šie parametrai dar skirstomi į :

- Vidinius (angl. k. *intrinsic*):
 - ✓ židinio nuotolis;
 - ✓ didinimo faktorius;
 - ✓ lęšio iškraipimai.
- Išorinius (angl. k. *extrinsic*):
 - ✓ pozicija (transliacija);
 - ✓ orientacija (pasisukimas).

Kamerų kalibravimą galima suskirstyti į šias kategorijas [7]:

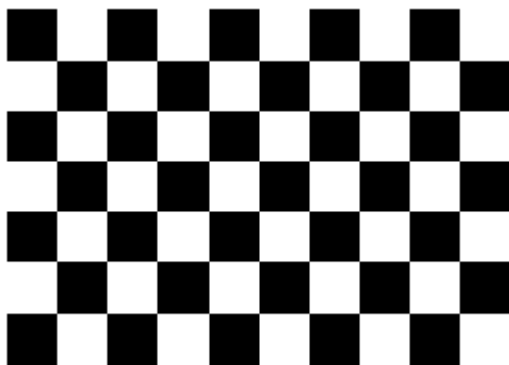
- a) 3D objektais paremta kalibracija, atliekama naudojant dvi arba tris ortogonaliai sukliajuotas plokštes (5 pav.). Tokia kalibracija labai efektyvi, tačiau reikalauja brangios aparatinės įrangos.



5 pav. 3D objektu paremta kalibravimo šablonas

Ši kalibracijos būdą galima suskirstyti į keturis žingsnius:

1. Surandami šablono (5 pav.) kampai.
 2. Gaunama kameros projekcijos matrica.
 3. Gaunami vidiniai ir išoriniai parametrai.
 4. Gauti parametrai optimizuojami.
- b) 2D plokštuma paremta kamerų kalibracija. Ši kalibracija nesudėtinga, tereikia dvimatę plokštumą pasukinėti įvairiais kampais. Taip pat nereikalauja specialaus pasirengimo, o kalibracijai naudojamą šabloną lengva pasigaminti (6 pav.).



6 pav. 2D kalibravimo šablonas

2D vaizdu paremtai kalibracijai rekomenduojami tokie žingsniai:

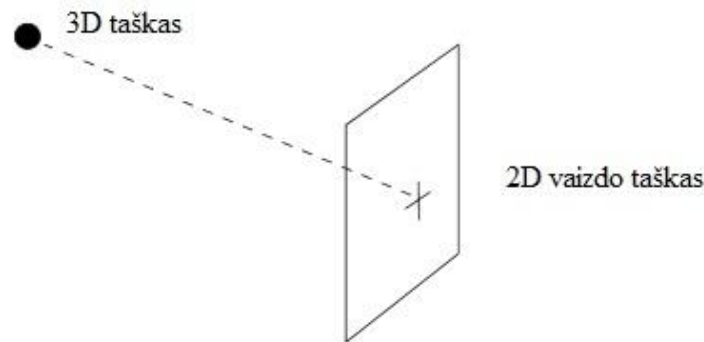
1. Priklijuoti atspausdintą (6 pav.) šabloną prie kieto paviršiaus.
 2. Padaryti kelis kadrus su šablonu, judinant šabloną arba kamerą.
 3. Atrasti pagrindinius (angl. k. *feature points*) vaizdo taškus.
 4. Surasti visus vidinius ir išorinius parametrus.
 5. Apskaičiuoti iškraipymo koeficientus.
 6. Minimizuoti visus parametrus.
- c) 1D linija paremta kalibracija naudojama lazdelė su keliais taškais. Sukant lazdelę aplink vieną iš taškų gaunama padėties erdvėje pozicija.
- d) Automatinei kalibracijai nereikia jokių objektų, visi reikalingi duomenys išgaunami iš trijų kameros vaizdo kadrų.

2.5.2 Kameros modelis ir projekcijos matrica

Kameros vaizdo atvaizdavimo procesą galima nusakyti trimis projekcijos matricomis. Apskritai koordinačių perkėlimas iš vaizdo į pasaulio koordinatės i-tajam vaizdui gali būti nusakytas matricų seka [8]:

$$P_i = KH_iH_{ii} = K(R_i t_i), K = \begin{pmatrix} \alpha & s & u_0 \\ 0 & k\alpha & v_0 \\ 0 & 0 & 1 \end{pmatrix}, H_i = \begin{pmatrix} R_i & t_i \\ 0^T & 1 \end{pmatrix}, H_{ii} = \begin{pmatrix} I & 0 \end{pmatrix} \quad (1)$$

(1) formulė nusako tiesinę projekciją $P^3 \rightarrow P^2$ (7 pav.), tai matrica kuria perkeliamos pasaulio trimatės (angl. k. *the world*) koordinatės į dvimates (angl. k. *the image*), čia taško koordinatės nusakomos homogenine matrica. R_i yra rotacijos matrica, t_i - perdavimo vektorius, I - identifikavimo 3×3 matrica, 0^T yra perkėlimo nulinis vektorius P^3 matricoje. K - tai 3×3 matrica aprašanti kameros optines savybes, α - židinio ilgis, (u_0, v_0) yra pagrindinio taško pikselio koordinatės, s - asimetrija.



7 pav. Projekcijos matricos sudarymo principas

Projekcijų matricos sudarymo algoritmas [9]:

- pozicijuoti kamerą į norimą vietą.
- kiekvienam taškui $p(x, y, z)$ 3D scenoje:
 - pridėti elementą $w=1$, homogeninio vektoriaus suformavimui $[xyz \ 1]^T$,
 - apskaičiuoti $[abw]^T = P[xyz1]^T$ (a, b - 2D vaizdo taško koordinatės),
 - padalinti iš w , jog rasti 2D vaizdo tašką: $x=a/w$ ir $y=b/w$,
 - atvaizduoti tašką kuris yra vaizde.

2.5.3 Populiariausi kamerų kalibravimo metodai

Kamerų kalibravimo esmė yra susieti vaizdo ir pasaulio žinomus taškus. Populiariausi kamerų kalibravimo metodai aprašomi *H. Zollner and R. Sablatnig* [8], tokie metodai yra Zang, Roger Tsai [10] ir DLT.

Zang kamerų kalibravimo algoritmą naudoja *Intel OpenCV* biblioteka ir *Matlab* kamerų kalibravimo įrankis. Šio algoritmo esmė 6×6 šachmatų lentos naudojimas (6 pav.) ieškant jos kampų ir kampų susiejimas su aplinkos taškais, laikant pagrindiniu tašku viršutinį kairįjį kampą. Iš aplinkos taškų išskaičiuojama projekcijos matrica (H) 3×3 dydžio susiejanti aplinkos taškus ir vaizdo taškus. Taip pat tokia pat matrica (H) gali būti perduota į kameros savybinius parametrus (A), pasukimo, perkėlimo matricas.

$$sm = HM \quad (2)$$

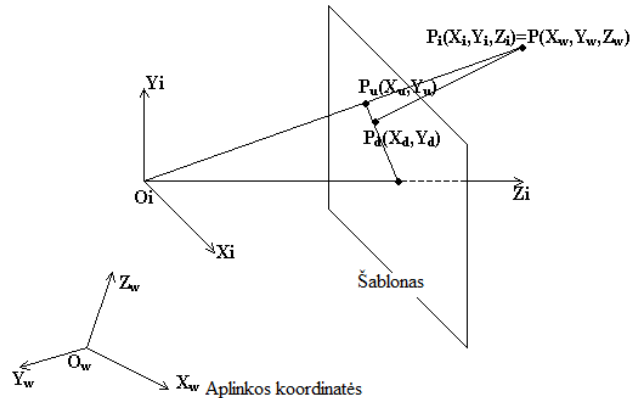
formulėje (2) s - yra didinimo faktorius, $m=[u,v,1]^T$ - paveikslo koordinatės, $M=[x,y,1]^T$ - kameros modelio koordinatės.

Tsai algoritmas yra dviejų etapų procesas, reikalaujantis 3D objekto kalibravimo šablono (6 pav.). Algoritmą galima lengvai pritaikyti tiek vienai kamerai tiek kelioms. Šis kalibravimo metodas iš kameros modelio išskaičiuoja R - posūkio kampą, T - perkėlimo komponentus iš transformacijos matricos tarp pasaulio ir kameros koordinačių, f - kameros židinio nuotolį, k - kameros lęšio iškraipymo koeficientą, S_x - didinimo koeficientą.

Pirmu etapu surandami S_x ir R , bei T perkėlimo vektoriai, taip pat surandamas f . Visa tai galima aprašyti formule:

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \quad (3)$$

formulę (3) lengva susieti su 8 pav., kur prie ašių koordinatų prirašyta w reiškia aplinkos (angl. k. *world*) koordinatės, o kur i vaizdo (angl. k. *image*) koordinatės. Antru etapu randamas k , kameros iškreipimo koeficientas.



8 pav. Tsaio kameros projekcijos modelis su perspektyvine projekcija

Tsai kalibravimo procedūra prasideda išsprendžiant tiesinės transformacijos matricą perėjimo iš pasaulio koordinatų (x_j, y_j, z_j) į dvimačio vaizdo koordinatas (u_{ij}, v_{ij}) . Tokią matricą nusako 3×4 matrica P_i i-tajai projekcijai ir N matomų taškų vaizde. DLT matricai naudojamus parametrus $p_{11} \dots p_{34}$ galima rasti iš homogeninės matricos lygties:

$$Lp_i = 0 \quad (4)$$

čia L yra $N \times 12$ matrica atitinkanti pasaulio ir dvimačio vaizdo koordinatas, o $p_i = (p_{11} \dots p_{34})$.

2.5.4 Vaizdo pradinis apdorojimas

Norint gauti patikimus duomenis iš vaizdo, vaizdą reikia suderinti. tam yra naudojama: šviesumo ir kontrasto gerinimas, gama korekcija, triukšmo šalinimas, vaizdo ryškinimas (angl. k. *sharpening*) ir dar daugelis kitų vaizdo gerinimo metodų.

Šviesumo ir kontrasto gerinimas

Šviesumas reguliuojamas[11] kiekvieno pikselio RGB paletės spalvą dauginant iš nustatytos konstantos α , jei ji didesnė už vienetą tada didėja šviesumas, jei mažesnė, paveikslas tamsėja, tai galima aprašyti tokia formule:

$$A = \alpha * B \quad (5)$$

čia B yra spalvos reikšmė, o A naujai gauta spalva, spalvos reikšmė gali būti nuo 0 iki 255, jei naudojame aštuonių bitų kodavimą.

Kontrasto gerinimas[12] taip pat vadinamas normalizavimu, tai paprastas metodas praplečiantis intensyvumo reikšmes vaizdo pikseliui, (6) formulė.

Pirmiausiai reikia nustatyti ribas, kuriose bus plečiamos reikšmės. Žemesnioji riba a , o aukštesnioji b , jei turime 8 bitų juodai baltą vaizdą tai proporcingai reikšmės bus 0 ir 255.

Antra, sudaroma originalaus vaizdo histograma ir nustatoma žemiausia ir aukščiausia reikšmės c ir d . Tada kiekvieną vaizdo pikselį galima normalizuoti pagal tokią formulę, kurioje r yra originali pikselio reikšmė:

$$s = (r - c) \left(\frac{b - a}{d - c} \right) + a \quad (6)$$

Gama korekcija

Norint pašviesinti tik tamsias vietas, reikia atlikti gama korekciją [11], matematiškai tai galime užrašyti tokia formule:

$$\text{Nauja reikšmė} = 255 * (\text{Sena reikšmė} / 255)^Y \quad (7)$$

šioje lygtyje (7) *Sena reikšmė* yra spalvos reikšmė, o *Nauja reikšmė* naujai gauta spalva, 255 yra absoliuti spalvos reikšmė, Y - konstanta, jei ji $Y > 1$, juodi atspalviai šviesėja, o jei $Y < 1$ balti atspalviai tamsėja.

Triukšmo šalinimas

Vaizdo glotninimas yra populiariausias metodas triukšmui šalinimui vaizde, tam naudojami filtrai [13]. Vaizdo filtravimo esmė paprasta - paimamas vaizdo pikselis, išgaunama pikselio spalvos reikšmė, reikšmė padauginama iš filtro vidurinės reikšmės, toliau analogiškai dauginami pikselio aplinkiniai kaimynai iš filtro reikšmių. Taip keičiama RGB paletės kiekviena spalva atskirai, arba jei turim tik pilką 8 bitų paveikslą, kuriame koduojama tik juodai balta spalva, sudarome 8 bitų kaukę.

	1	1	1	
$\frac{1}{9} \times$	1	1	1	
	1	1	1	
	1	1	1	

	1	2	1	
$\frac{1}{16} \times$	2	4	2	
	1	2	1	

9 pav. 3x3 glotninimo filtrai

9 pav. yra 3x3 glotninimo filtrai. Pirmas filtras apskaičiuoja pikselių vidurkį, po tokia kaukę galima užrašyti tokia lygtimi:

$$R = \frac{1}{9} \sum_{i=1}^9 z_i \quad (8)$$

šioje lygtyje apskaičiuojamas vidutinis pilkumas kiekvienam pikseliui 3x3 kaukėje. Pirmoje matricoje visos kaukės dedamosios yra 1 dėl didesnio efektyvumo skaičiuojant, tada visa matricą dauginama iš $1/9$. Šis filtravimo procesas suskirsto vaizdą į 9 kvadratus ir sudaroma $m \times n$ kaukė normalizuojanti konstantą lygia $1/mn$. Toks filtras vadinamas dėžiniu filtru.

Antra kaukė 9 pav. naudoja svorius, tokio svorinio suskirstymo esmė, jog kiekvienas pikselis dauginamas iš skirtingo koeficiento. Centre yra suteikiama didžiausias svoris, o tolstant nuo centro pikselių svoris mažėja (mažėjimo faktorius $\sqrt{2}$).

Vaizdo ryškinimas

Vienas iš populiariesnių ir paprastesnių metodų vaizdo aštrinimui yra Laplaso filtrų naudojimas [13]. Filtro esmė - kontūrų ryškinimas. Kiekvienam vaizdo pikseliui galime pritaikyti tokias pačias kaukes kaip vaizdo glotninimo metode - tačiau kaukės naudojamos skirtingos, pavyzdys:

-1	-1	-1
-1	9	-1
-1	-1	-1

10 pav. Vaizdo ryškinimo kaukė

tokios kaukės (10 pav.) turi ir trūkumų - jos sukelia triukšmą vaizde jeigu naudojamos didelės filtro reikšmės.

Statinio vaizdo pašalinimas

Žymeklių indentifikavimas vyksta pirmiausiai iš turimo vaizdo pašalinant statinį vaizdą [4]:

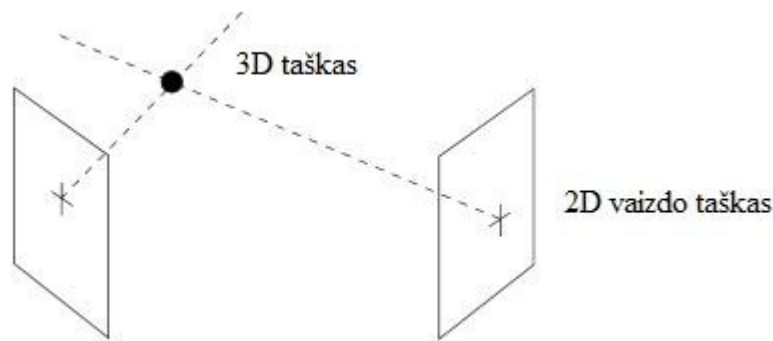
$$|RGB(kadras_{bg}) - RGB(kadras_i)| > 0 \quad (9)$$

formulės (9) dedamosios - $kadras_{bg}$ - statinis vaizdas, $kadras_i$ - sekantis kadras kuriame yra sistemos naudotojas, RGB - spalvų dedamosios. Tačiau ši formulė (9) gali sugeneruoti triukšmą, kad to būtų išvengta iš paskutinių kadrų išmetami paskutiniai kadrai, panaudojus tokią formulę:

$$\frac{1}{n} \sum_{j=1}^n |RGB(kadras_{bg}) - RGB(kadras_i)| > 0 \quad (10)$$

2.6 Erdvinio taško rekonstrukcija iš 2-jų kameros vaizdų

Sukalibravus kameras galima atlikti taško erdvėje rekonstrukciją naudojant 2-jų ar daugiau kamerų vaizdus, tai atvirkštinis metodas kameros vaizdavimui (11 pav.). Tas pats taškas stebimas iš dviejų kamerų, to rezultatas dvi linijos kurios kertasi 3D taške. Tokios linijos vadinamos projekcijos linijomis, jos dažniausiai būna skirtingos, tačiau projektuojamos į vieną tašką. Visa tai galima užrašyti lygtimis[9].



11 pav. 3D taško rekonstrukcija naudojant 2 kameras

3D taško procedūrai reikalingos projekcijos matricos P_1 ir P_2 gaunamos iš kamerų.

$$\begin{bmatrix} a_1 \\ b_1 \\ w_1 \end{bmatrix} = P_1 \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} a_2 \\ b_2 \\ w_2 \end{bmatrix} = P_2 \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (12)$$

čia a_n/w_n ir b_n/w_n yra koordinatės kameros vaizde, w homogeninė koordinatė, bet ji nėra lygi 1. Taigi norėdami rasti X , Y ir Z reikia išspręsti matricą $[XYZ]^T$. Sudauginus lygtis (11) ir (12), galima gauti tokią išraišką:

$$\begin{aligned}\alpha_i &= X_{p_{11}}^i + Y_{p_{12}}^i + Z_{p_{18}}^i + p_{14}^i \\ b_i &= X_{p_{21}}^i + Y_{p_{22}}^i + Z_{p_{28}}^i + p_{24}^i \\ w_i &= X_{p_{81}}^i + Y_{p_{82}}^i + Z_{p_{88}}^i + p_{84}^i\end{aligned}\quad (13)$$

čia p_{11}^i reiškia elementą 1,1 projekcijos matricoje P_i , kur i yra 1 arba 2. Pakeitus $x_i w_i = a_i$ ir $y_i w_i = b_i$ gauname:

$$\begin{aligned}X(x_i p_{81}^i - p_{11}^i) + Y(x_i p_{82}^i - p_{12}^i) + Z(x_i p_{88}^i - p_{18}^i) &= p_{14}^i - x_i p_{84}^i \\ X(y_i p_{81}^i - p_{21}^i) + Y(y_i p_{82}^i - p_{22}^i) + Z(y_i p_{88}^i - p_{28}^i) &= p_{24}^i - y_i p_{84}^i\end{aligned}\quad (14)$$

pakeitus $i=1$ ir $i=2$, gausime 4 lygtis, ir galima bus užrašyti matricų daugybą:

$$\begin{bmatrix} x_1 p_{81}^1 - p_{11}^1 & x_1 p_{82}^1 - p_{12}^1 & x_1 p_{88}^1 - p_{18}^1 \\ y_1 p_{81}^1 - p_{21}^1 & y_1 p_{82}^1 - p_{22}^1 & y_1 p_{88}^1 - p_{28}^1 \\ x_2 p_{81}^2 - p_{11}^2 & x_2 p_{82}^2 - p_{12}^2 & x_2 p_{88}^2 - p_{18}^2 \\ y_2 p_{81}^2 - p_{21}^2 & y_2 p_{82}^2 - p_{22}^2 & y_2 p_{88}^2 - p_{28}^2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p_{14}^1 - x_1 p_{84}^1 \\ p_{24}^1 - y_1 p_{84}^1 \\ p_{14}^2 - x_2 p_{84}^2 \\ p_{24}^2 - y_2 p_{84}^2 \end{bmatrix}\quad (15)$$

tokia lygtį galima būtų toliau išreikšti per formulę:

$$AX = B\quad (16)$$

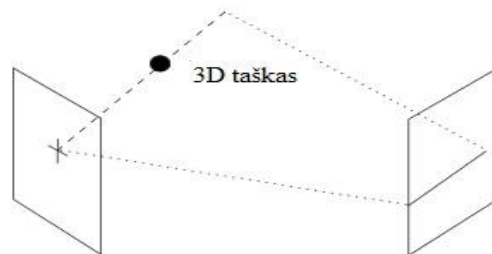
ir galiausiai gauti 3D koordinatas per lygtį:

$$X = (A^T A)^T A^T B\quad (17)$$

Algoritmas 1

- Rasti P_1 ir P_2 projekcijos matricas kameroms.
- 3D taško radimui:
 - Rasti taško poziciją kiekviename kameros vaizde (x_1, y_1) ir (x_2, y_2) .
 - Suformuoti matricas A ir B iš (x_1, y_1) ir (x_2, y_2) , P_1 ir P_2 naudojant formulę (16) :
 - Suformuoti ir invertuoti matricas $(A^T A)$.
 - Išspręsti lygtį (17) kuri duoda 3D poziciją

Kitoks 3D taško rekonstrukcijos metodas naudojant polinę geometriją (12 pav.). Tokios rekonstrukcijos principas, jog kairinėje kameroje randamas taškas, kuris yra ant projekcijos linijos, jo tiksli vieta nežinoma. Paėmus ir suprojektavus kairės kameros projekcijos liniją dešinėje kameroje, galima išskaičiuoti taško erdvės koordinatas. Tam yra apjungiamos abiejų kamerų projekcijos matricos ir suformuojama fundamentali matrica F .



12 pav. Polinė geometrija

Algoritmas 2

- Rasti P_1 ir P_2 projekcijos matricas apjungti ir suformuoti F matricą.
- 3D taško radimas:

- rasti kairėje kameroje poziciją $p_1=[x_1y_11]^T$;
- apskaičiuoti polinę liniją $e=Fx$, e forma $[abw]^T$ kur taškai ant linijos tenkina tokia lygtį $ax+by+w=0$;
- skanuoti pagal e liniją dešinėje kameroje, jog rasti $p_2=[x_2y_21]^T$;
- panaudoti (17) lygtį 3D taško radimui iš p_1 ir p_2 .

Naudojant abiejus algoritmus gaunamas tas pats rezultatas - rekonstruojamas trimatis taškas. Pirmo algoritmo privalumas, jog jis yra lengvai, realizuojamas lyginant su antru algoritmu, tačiau veikia žymiai lėčiau, nei antrasis. Tačiau antrojo trūkumas - tikslumas, jei neteisingai suprojektuojamas taškas vienoje iš naudojamų kamerų, gaunami labai netikslūs rezultatai.

2.7 Judesio sekimo metodologija

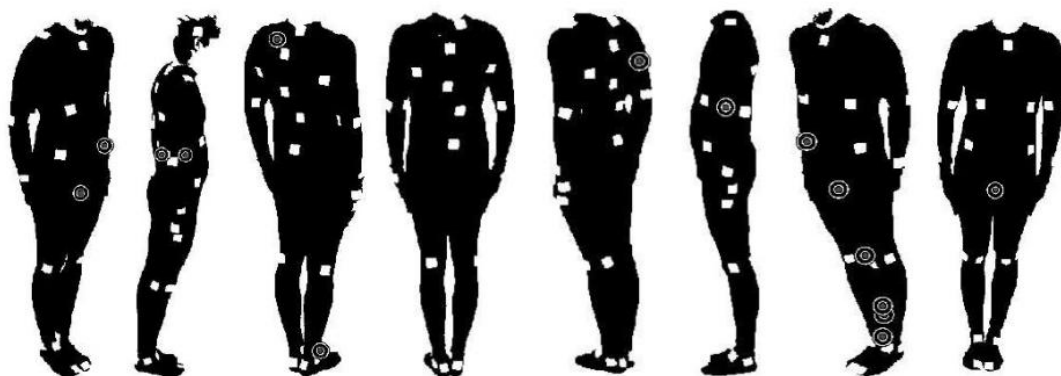
2.7.1 Žymeklių išskyrimas

Komercinėse sistemose žymekliai sekami naudojant infraraudonąsias kameras. Tačiau neturint tokių kamerų, sekti žymeklius galima ir naudojant paprastas internetines kameras. Tada žymeklius galim sekti[5] tokiais būdais:

- intensyvumo segmentavimu;
- pagal linijas;
- pagal kampus;
- pagal spalvas

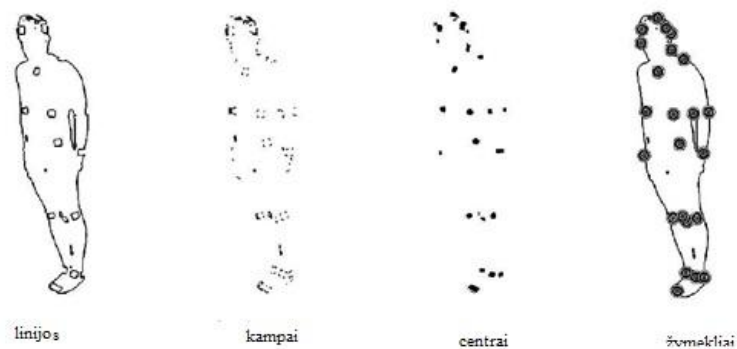
Idealiu atveju sistemos kameros gražina 2D žymeklių pozicijos vaizdą.

Naudojant intensyvumo segmentavimo metodą pašalinimas foninis vaizdas. Po to naudojamas juodai baltas vaizdas, tokiu būdu išryškėja žymekliai įgaudami baltą spalvą (13 pav.).



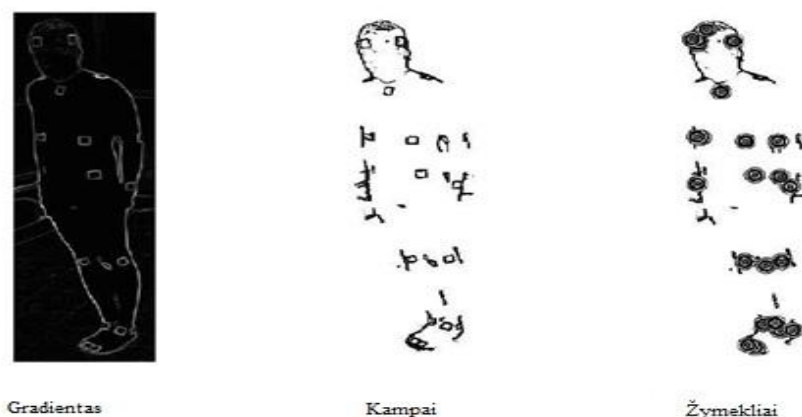
13 pav. Intensyvumo segmentavimas

Kitas būdas žymekliams išskirti yra pagal linijas (14 pav.), kai vaizde ieškoma tamsių linijų, kampų porų kurie identifikuojami kaip linijos ir jei toks segmentas savyje turi baltą regioną, jam nustatomas centras ir tada toks regionas traktuojamas kaip žymeklis.



14 pav. Linijinis identifikavimas

Identifikavimo pagal kampus metodu (15 pav.) vaizdui pritaikomas gradientinis filtras. Tokiu būdu ieškomi kraštai ir kampai su nustatytu kampų ar kraštų kiekiu. O apsijungę kampai identifikuojami kaip žymekliai.



15 pav. Identifikavimas pagal kampus

Identifikavimas pagal spalvas [14] yra daug paprastesnis, nei prieš tai išvardinti metodai, bei ekonomiškai patrauklesnis piniginių resursų atžvilgiu, nes nebereikia pritaikyti jokių filtrų, tereikia naudoti spalvos išskyrimo filtrą arba spalvos pikselių indentifikavimą vaizde (16 pav.). Filtrus arba histogramas galima sudarinėti naudojantis HSV arba RGB spalvų paletes, taip pat galima identifikuoti spalvas paveiksle naudojant klasifikatorius.

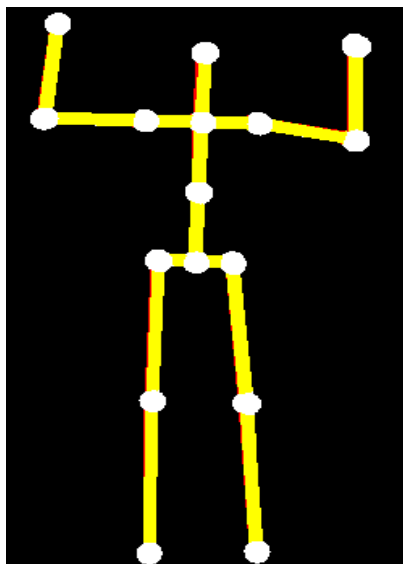


16 pav. Spalvinis žymeklių atpažinimas

2.7.2 Žymeklių rekonstrukcija erdvėje

Žymeklio rekonstrukcijos metu reiki išskirti žymeklius, identifikuoti ir nustatyti jų poziciją erdvėje [1]. Pozicija erdvėje nustatoma naudojantis algoritmais aprašytais 2.6 skyriuje.

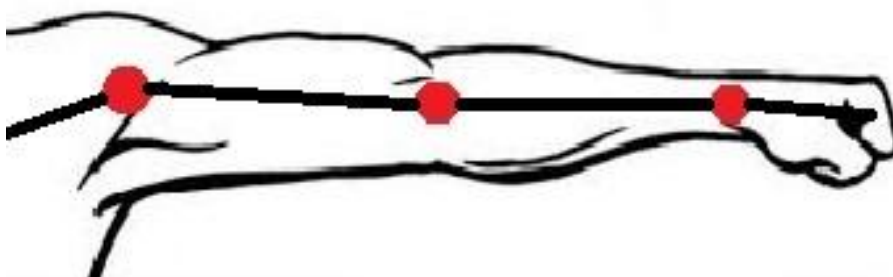
Identifikavimas vyksta atsistojus į gimnasto pozą (17 pav.). Tada rankiniu būdu priskiriama reikšmė žymekliui, kokia tai kūno dalis. Jei naudojami spalviniai žymekliai, galima juos identifikuoti pagal spalvas, spalviniai žymekliai pranašesni tuo jog, galima iškart priskirti normą spalvą tam tikram skeleto sąnariui. Tokiu būdu išvengiama rankinio žymeklių priskyrimo.



17 pav. Gimnasto poza

Žymeklių trūkumas, jog žymekliai gali persidengti, dėl to ir juos gali būti sudėtinga identifikuoti, bet naudojant spalvinius žymeklius, tai nutinka retai. Sistemose su aktyviais žymekliais to nenutinka, nes kiekvienas žymeklis gražina savo poziciją erdvėje. Ekonominės klasės sistemose naudojančiose pasyviuos žymeklius - didinamas kamerų kiekis. O dingus žymekliui taikomi prognozavimo metodai.

2.8 Skeleto rekonstrukcija



18 pav. Skeleto modelio pavyzdys

Norint atlikti skeleto rekonstrukciją [15] reikia turėti jau sumodeliuotą skeleto modelį - sužymėtus žymeklius (skyrius 2.7.2). Identifikuoti žymekliai yra priderinami prie sumodeliuoto modelio, o skeletas pagal žymeklių išsidėstymą ant aktorius yra sumažinamas arba padidinamas pagal jo anatomiją. Skeletas yra sudaromas iš trimačių primitivų. Dažniausiai yra naudojami kūginiai ir cilindriniai.

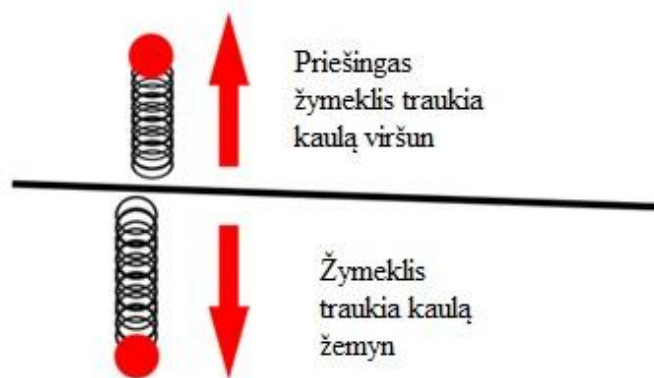
18 pav. pateiktas sudaryto skeleto modelio pavyzdys, kur raudoni apskritimai yra sąnariai, tarp jų yra kaulai, kartu šie tampa ir valdikliais. Toks skeleto modelis naudojamas atvirkštinės kinetikos (angl. k. *inverse kinematics*) skeleto judėjimui.

2.8.1 Skeleto judėjimas

Skeletas gali judėti naudojant spyruoklinį modelį[4], atvirkštinės kinetikos[15] arba tiesinės kinetikos (angl. k. *forward kinematics*) būdą.

Spyruoklinis modelis

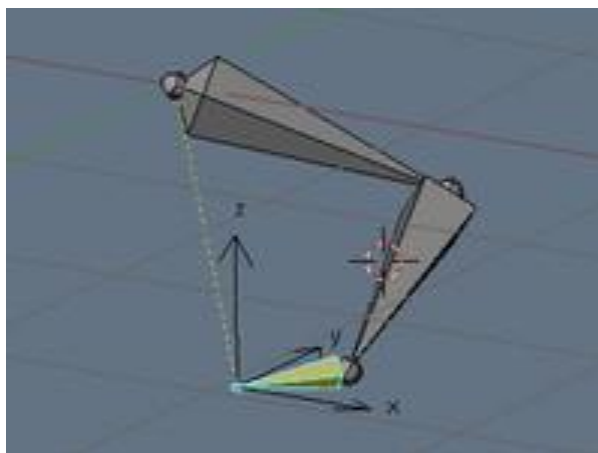
Toks modelis pasižymi, jog prie kiekvieno kaulo prisegamas žymeklis, žymeklį su kaulu sieja spyruoklė. Tokiu būdu kiekvienas žymeklis tempia virtualaus skeleto kaulus pagal savo judėjimo kryptį (19 pav.). Tačiau tokiam judėjimui reikia žymiai daugiau žymeklių, nei atvirkštinės kinetikos (angl. k. *inverse Kinematics*) metodui ir taip pat reikia daugiau kamerų.



19 pav. Spyruoklinis modelis

Atvirkštinės kinetikos metodas

Naudojant atvirkštinės kinetikos metodą skeleto judesys apskaičiuojamas pagal mechanikos taisykles[4]. Tokią animacijos rūšį galima sutikti, pavyzdžiui trimatės animacijos kūrimo programose, kai valdantieji kaulai apskaičiuoja kitų grandine sujungtų kaulų judėjimą pagal savo judėjimą ir pasisukimą (20 pav.).



20 pav. Kaulų judėjimas naudojant atvirkštinės kinetikos metodą

Atvirkštinės kinetikos judėjimui reikia daug mažiau žymeklių, nei spyruokliniam metodui.

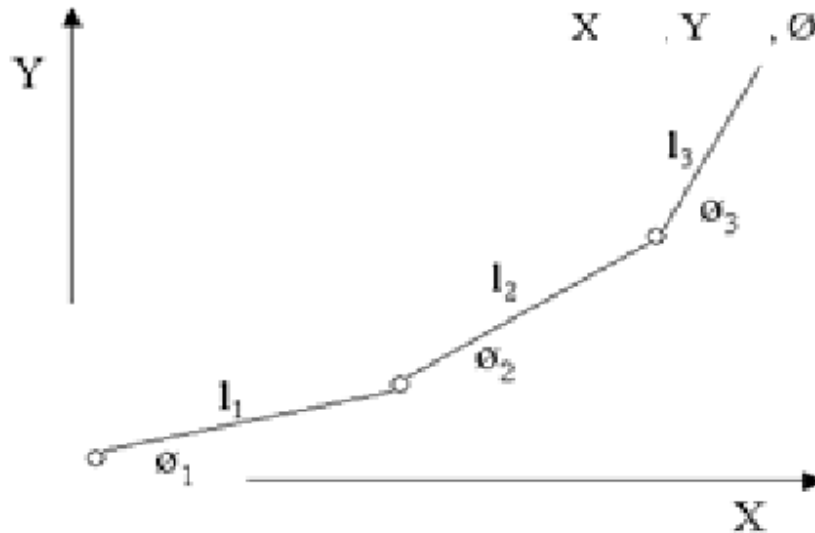
Tokiame modelyje [15] kiekvieno skeleto sąnario orientacija gali būti nusakyta kaip vektorius $\theta = (\theta_1, \dots, \theta_n)^T$, θ_i nusako rotaciją aplink nustatytą ašį. O atvirkštinės kinetikos skeleto judėjimą galima nusakyti formule:

$$\Delta \theta = J^+ \Delta s + (I - J^+ J) \Delta \varphi \quad (18)$$

šioje formulėje J yra Jacobian'o matrica, kurioje nusakoma skeleto valdiklių pozicija ir posūkio kampas, J^+ pseudo J inversija. Vektorius $\Delta s \in \mathbb{R}^{3k}$ apibūdina valdiklio kampo pasikeitimus, kitas vektorius $\Delta \varphi \in \mathbb{R}^n$ apibūdina skeleto pozą, o operatorius $I - J^+ J$ garantuoja, kad nebus pažeista valdiklio pozicija.

Tiesinės kinetikos kaulinis judėjimas

Tiesinės kinetikos kaulinis judėjimas dažniausiai naudojamas robotų judėjime [16]. Naudojant tokią techniką ir valdant roboto ranką, kiekvienam sąnariui reikia nurodyti jo Eulerio kampą, pagal tai paskaičiuojama kiekvieno kaulo kuris yra rankoje pozicija ir orientacija. 21 pav. matome 2D atvejį, kai turim l_i - kaulo ilgį, X, Y - koordinačių sistemą, ir kampą θ_i . Tada kiekvieną elemento pasisukimą perskaičiuojame priklausomai nuo grandinės ilgio.

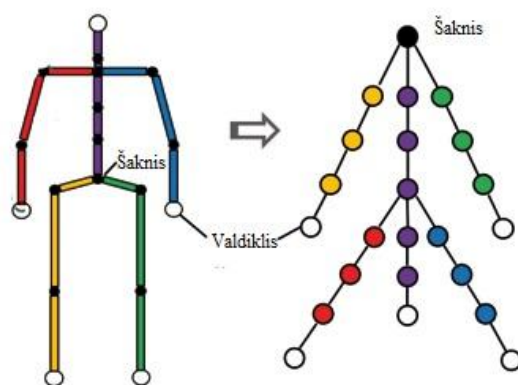


21 pav. Tiesinė kinetika

2.8.2 Virtualaus skeleto judėjimas ir skeleto medžio sudarymas

Dažniausiai skeletinės [5] sistemos kaulo, segmento vietai lokaliojoje koordinačių sistemai nustatyti reikia trijų žymeklių. Segmento poziciją ir orientaciją erdvėje galima apskaičiuoti per transformaciją tarp lokalios ir pasaulio koordinačių sistemų.

Grupuojant žymeklius prie segmentų, galima apskaičiuoti - vidinių sujungimų pasisukimus. Sugrupavus žymeklius reikia formuoti hierarchinį skeleto segmentų medį (22 pav.). Žymeklis arčiausiai gravitacijos centro tampa šakniniu medžio elementu. Toks žymeklis perduoda globalius kūno poslinkio ir orientacijos duomenis erdvėje. Šaknies atšakų žymekliai talpina tik lokalius duomenis apie savo padėtį erdvėje.



22 pav. Skeleto kaulų medžio sudarymas

Medžio segmentų pasisukimas ir poslinkis apskaičiuojamas kaip transformacija kuri siejasi su lokaliąja koordinatų sistema kartu su tėvine koordinatų sistema. Kampai panašūs į transformacijas tarp koordinatų sistemų kurios jungia skeleto segmentus vadinami pasisukimo duomenimis.

Pasisukimai nusakomi homogeninėmis transformacijomis kuriose prisideda ašių kampai, kvarterionai, pasisukimo matricos ir Eulerio kampas. Pasisukimo matricą galime apskaičiuoti pagal formulę kai ašys nusakomos tokia tvarka: Z-Y-Z ir kampai tarp jų φ , θ , ψ . Tada pasisukimo matricą galime aprašyti taip[5]:

$$R(\varphi, \theta, \psi) = R_Z(\varphi) \times R_Y(\theta) \times R_Z(\psi) \quad (19)$$

kur R yra posūkio matrica, Eulerio kampai posūkio matricoje pagal tam tikrą ašį apsiskaičiuoja:

$$\varphi = \text{atan2}(R_{2,3}, R_{1,3}) \quad (20)$$

$$\theta = \text{atan2}(\cos\varphi R_{1,3} + \sin\varphi R_{2,3}, R_{3,3}) \quad (21)$$

$$\psi = \text{atan2}(-\sin\varphi R_{1,1} + \cos\varphi R_{2,1}, -\sin\varphi R_{1,2} + \cos\varphi R_{2,2}) \quad (22)$$

2.9 Objektų sekimo algoritmai

Vaizdo sekimo algoritmų yra labai daug, tačiau dažniausiai naudojami:

- artimiausio kaimyno[17];
- Kalmano filtras[18];
- Kondensacijos metodas[19];
- Lucas-Kanade metodas[20].

Pirmoje lentelėje (1 lentelė.) pateikiami visi aukščiau pateiktus algoritmai, bei jų privalumai ir trūkumai.

1 lentelė. Objektų sekimo algoritmai

Algoritmo pavadinimas	Privalumai	Trūkumai
Artimiausio kaimyno	-lengvai aprašomas; -galima sekti kelis žymeklius; -gali sekti besikeičiantį žymeklių skaičių; -galimas apsimokymas;	-įvykus persidengimui, nenaudingas; -nėra judesio nuspėjimo; -gana lėtas;
Kalmano filtras	-sekamo objekto trajektorijos nuspėjimas; -algoritmas apsimoko; - trumpas atsako laikas; -lengvai seka objektus tarp daugybės detalių; -naudoja statistinius metodus;	-sudėtingas; -reikalinga duomenų bazė kurioje būtų saugomi istorija objektų kurios sekė; -reikia atlikti matavimus prieš naudojimą; -limituotas kiekis tikimybių; -negali nuspėti persidengusio žymeklio judėjimo;

Algoritmo pavadinimas	Privalumai	Trūkumai
Kondensacijos metodas	-paprastesnis algoritmas, nei Kalmano filtras; - lengvai seka objektus tarp daugybės detalių; - bendresnis objektų judėjimo nuspėjimas; -naudoja statistinius metodus; -galima pasirinkti hipotezių skaičių. -galima sekti kelis žymeklius;	-negali sekti besikeičiančio kiekio žymeklių; - reikia atlikti matavimus, bandymus;
Lucas-Kanade metodas	-atsparus vaizdo triukšmams; -gali pagauti didelius judesius naudojant piramides;	-jei padaromas staigus judesys, toks judesys nesusekamas;

2.10 Bibliotekos

Populiariausios vaizdo apdorojimo bibliotekos yra OpenCV, SimpleCV, VXL ir LTI. Bibliotekos (2 lentelė.) yra lyginamos pagal programavimo kalbų suderinamumą, greitį (kaip greitai atlieka algoritmus) [23,24], kainą, paprastumą naudojantis tokia biblioteka ir ar tai funkcinė kalba, ar objektinė grysta biblioteka [23,24].

2 lentelė. Bibliotekų palyginimas

Biblioteka	Programavimo kalbos	Greitis	Platforma	Kaina	Paprastumas naudojantis	Kalba
OpenCV	C++ Python Java C#	9/10	Linux Window Mac	Nemokama	Paprasta	Funkcijomis grįsta
SimpleCV	Python	5/10	Linux Window Mac	Nemokama	Labai paprasta	Funkcijomis grįsta
VXL	C++	4/10	Linux Window Mac	Nemokama	Paprasta	Objektinė
LTI	C++	4/10	Linux Window Mac	Nemokama	Paprasta	Objektinė

2.11 Esamų sprendimų analizė

Esami sprendimai analizuojami pagal tai kokias rinkoje jau egzistuojančios sistemos turi teigiamas ir neigiamas savybes (3 lentelė.).

3 lentelė. Esami sprendimai

Pavadinimas	Pliusai	Minusai
Kinect[25]	- nereikia žymeklių - automatinis apsimokymas - skirtingų žmonių identifikavimas	- kaina - reikia daug vietos

Pavadinimas	Pliusai	Minusai
	- galima pritaikyti judesio sekimo sistemoms. - Palaiko OpenCV, OpenNI bibliotekas	
PlayStation Move[26]	- nereikia žymeklių - didelės spartos kamera	- kaina - reikalingas specialus valdiklis
FreeTrack[27]	- nereikia žymeklių - suderinama su betkokia kamera - galima naudoti kelias kameras	reikalingas Wii valdiklis
Qualysis[28]	- judesio atkūrimas po vandeniu - didelė judesio atkūrimo sparta - platus taikymas tiek medicinoje tiek multimedijoje	Reikalingos sudėtingos kameros

2.12 Duomenų perdavimas tarp sistemų

Dažniausiai duomenų perdavimui tarp sistemų naudojami du pagrindiniai duomenų perdavimo protokolai TCP IP ir UDP IP, kurių skirtumus apžvelgiame (4 lentelė.) pagal panaudojimą, pavyzdžius, duomenų paketus, persiuntimo greitį, patikimumą, bei antraštės dydį.

4 lentelė. TCP IP ir UDP IP skirtumai

	TCP	UDP
Naudojimas	Tinkamas aplikacijoms kurios reikalauja aukšto patikimumo ir perdavimo laikas nėra aktualus.	Naudojamas kai reikia greito duomenų perdavimo.
Pavyzdžiai	HTTP, HTTPS, FTP, SMTP, Telnet	DNS, DHCP, TFTP, SNMP, RIP, VOIP
Duomenų paketai	Duomenys surikiuojami ir išsiunčiami pagal nustatytą tvarką.	Duomenys nerikiuojami, duomenų paketai paleidžiami be tvarkos.
Persiuntimo greitis	Lėtesnis, nei UDP.	Greitesnis nei TCP, nes nėra klaidų tikrinimo.
Patikimumas	Yra užtikrinimas, kad duomenys bus išsiųsti ir pasieks siuntimo vietą ta pačia tvarka kaip ir buvo išsiųsti.	Nėra garantijų, kad duomenų paketai pasieks siuntimo vietą.
Antraštės dydis	20 bitų.	8 bitai.

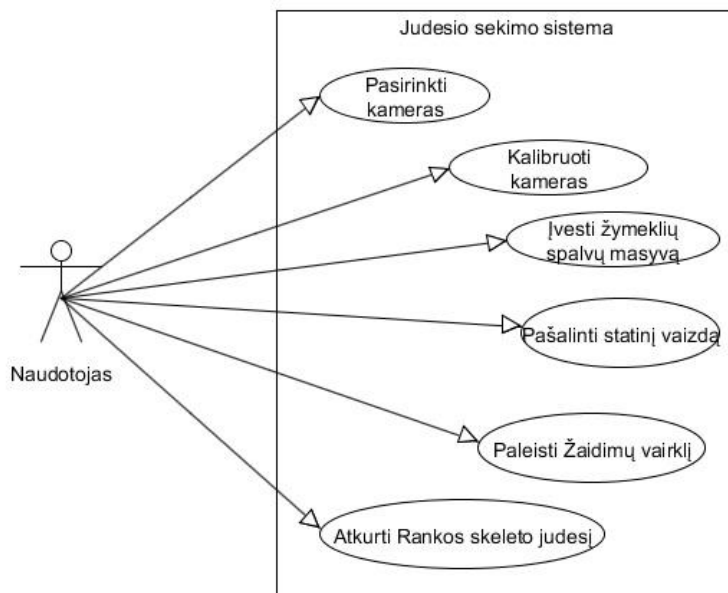
2.13 Analizės išvados

1. Atlikus judesio sekimo metodų ir elementų, bei skeleto iš vaizdo atkūrimo analizę, paaiškėjo, kad siekiant sukurti neaukštos kainos, tikslių, reikalaujantį nedidelės skaičiuojamosios galios metodą, būtina, jog sistema naudotų žymeklius ir 2-3 kameras judesio tikslumui didinti.
2. Atlikus judesių atkūrimo sistemų žymeklių analizę, paaiškėjo, kad siekiamam sprendimui idealiai tiks pasyvūs žymekliai, dėl savo, paprastumo lengvo pritaikymo, bei nedidelės kainos.
3. Ištyrus naudojamų kamerų judesio atkūrimo sistemose specifikacijas, priimtas sprendimas naudoti internetines kameras, dėl jų didesnės nei 30 kadrų per sekundę greitaveikos, bei priimtinos raiškos (640x480) vaizdui tirti.
4. Susipažinus su kalibravimo metodais, paaiškėjo, jog lengviausiai pritaikomas yra Zang 2D šablono kalibracijos algoritmas, dėl lengvo šablono pagaminimo ir nesudėtingo algoritmo pritaikymo, bei dėl tokio algoritmo naudojimo *OpenCV* ir *Matlab* vaizdo tyrimo bibliotekose.
5. Atliekant 3D taško rekonstrukciją iš dvimačio vaizdo, naudojant dvi ar daugiau kamerų, paaiškėjo, jog naudojant projekcijos linijas galima paprasčiau pasiekti norimą rezultatą, nei naudojant polines projekcijas dėl *OpenCV* jau esančių metodų.
6. Norint sutaupyti skaičiuojamosios galios, buvo išsiaiškinta jog identifikavimui žymeklių geriausiai tinka identifikavimas pagal spalvas, dėl paprasto išskyrimo vaizde ir greito veikimo.
7. Atlikus kaulų judesio valdymo analize nustatyta, jog geriausiai tiks tiesinės kinetikos metodas, dėl nesudėtingo skaičiavimo ir savo paprastumo.
8. Objektų sekimui vaizde geriausiai tinka statistiniai metodai, dėl išankstinio metodų apsimokymo ir geresnių rezultatų sekant objektus vaizde ir nuspėjant jų buvimo vietą.
9. Atlikus programavimo kalbų, vaizdo tyrimo bibliotekų ir esamų sprendimų analizę, nustatyta, jog metodas turėtų greičiausiai veikti naudojant C++ programavimo kalbos ir *OpenCV* biblioteką. Norint greičiau parašyti metodą realizuojančią sistemą ir lengviau pritaikyti *OpenCV* biblioteką nuspręsta naudoti *Python* programavimo kalbą, nes kodo programavimo laikas sutrumpėja per pusę, nei programuojant C++ programavimo kalba.
10. Atlikus TCP ir UDP duomenų perdavimo protokolų palyginimą, nuspręsta naudoti TCP dėl to, kad UDP gali prarasti duomenis perdavimo metu, bei duomenys negaunami išsiuntimo tvarka.

3 METODO SISTEMOS REIKALAVIMŲ SPECIFIKACIJA

3.1 Funkciniai reikalavimai

Sistemos, kuri bus panaudota metodo kūrimui funkcijos aprašytos panaudos atvejų diagrama (23 pav.). Diagramoje vaizduojama kokias funkcijas gali atlikti naudotojas besinaudojantis judesio atkūrimo sistema..



23 pav. Panaudos atvejų diagrama

Judėsio sekimo sistemos funkcijos detalizuotos žemiau pateiktoje lentelėje (5 lentelė.).

5 lentelė. Judėsio sekimo sistemos funkcijų aprašas

Funkcija	Pasirinkti kameras
Aprašymas	Naudotojo užklausiama „Kokias kameras naudosite? “. I tekstinę eilutę įrašoma norimų naudoti kamerų numeriai (0, 1, ... n)
Aktoriai	Naudotojas
Sąlyga	Ijungtos kameros į kompiuterį
Rezultatas	Pasirinktos kameros, programa kameras naudos tolimesnėse funkcijose.
Funkcija	Kalibruoti kameras
Aprašymas	Naudotojo užklausiama: „Ar naudosite kamerų kalibraciją?“. Naudotojas įrašo: „Taip“. Naudotojas įrašo: "Ne" .
Aktoriai	Naudotojas
Sąlyga	Pasirinktos kameros
Rezultatas	Paleidžiama kamerų kalibravimo posistemė. Gaunami erdvės duomenys, suderinamos kameros tarpusavyje.
Funkcija	Įvesti žymeklių spalvų masyvą
Aprašymas	Naudotojo prašoma įvesti žymeklių masyvo reikšmes „Įvesti žymeklių spalvų reikšmes: “. Suvedamas žymeklių spalvų matrica.
Aktoriai	Naudotojas
Sąlyga	Išpildytos pasirinkti kameras, kalibruoti kameras funkcijos
Rezultatas	Žymekliams priskiriamos spalvos.

Funkcija	Pašalinti statinį vaizdą
Aprašymas	Naudotojo užklausiama „Ar naudosite statinio vaizdo šalinimą?“. Naudotojas įrašo: „Taip“ arba „Ne“.
Aktoriai	Naudotojas
Sąlyga	Sukalibruotos kameros
Rezultatas	Iš kamerų gaunamo vaizdo pašalinimas statinis vaizdas.
Funkcija	Paleisti žaidimų variklį
Aprašymas	Naudotojas paleidžia žaidimų Panda3D žaidimų variklį naudojam NetBeans aplinkoje.
Aktoriai	Naudotojas
Sąlyga	
Rezultatas	Pasileidžia langas kuriame yra animuojamas modelis.
Funkcija	Atkurti rankos skeleto judesį
Aprašymas	Naudotojo užklausiama „Ar ar pradėti judesio atkūrimo procesą“. Naudotojas įrašo: „Taip“ arba „Ne“.
Aktoriai	Naudotojas
Sąlyga	Turi būti atrinkti žymekliai, suderintos skeleto ir žymeklių pozicijos.
Rezultatas	Pagal sistemos naudotojo judesius atvaizduojamas virtualaus skeleto judėjimas ir įrašoma į kietąjį diską jo judesių animacija, tam tikru judesio atkūrimo failo formatu.

3.2 Nefunkciniai reikalavimai

Sistemos nefunkciniai reikalavimai:

- Sistema turi veikti *Python 2.7* pagrindu.
- Sistema turi veikti *Windows 7* operacinėje sistemoje.
- Sistema turi naudoti *OpenCV 2.4* judesių aptikimui.
- Sistema turi naudoti *Panda3D 1.8.1* variklį judesių atvaizdavimui.
- Sistemos veikimui turi pakakti 2Ghz procesoriaus (angl. k. *CPU*), Integruoto grafinio procesoriaus (angl. k. *GPU*) su 128mb DDR3 atminties, bei 2GB operatyviosios atminties.
- Sistemos veikimui turi pakakti 1GB HDD atminties.
- Sistemos skeleto judėjimo rekonstravimo posistemės greیتaveika turi būti didesnė nei 20 kadrų per sekundę.

3.3 Techniniai reikalavimai

Sistema turi tenkinti tokius techninius reikalavimus:

- Dviejų branduolių 2Ghz *CPU*, integruota vaizdo plokštė palaikanti *OpenGL 2.0* biblioteką.
- Ne daugiau 2GB RAM.
- Ne daugiau nei 1GB HDD.
- Sistemai reikalingas kompiuteris su dviem laisvais *USB* prievadais.
- Spalviniai žymekliai.
- Naudojamas vienspalvis fonas (rekomenduotinas žalios spalvos).
- Aktoriaus apranga vienspalvė, sugerianti šviesą (tinkamiausia juoda).
- Apšvietimas turi būti metantis kuo mažiau šešėlių.
- Reikalingos 2 *USB* internetinės kameros palaikančios 30 kadrų per sekundę filmavimo dažnį ir 640x480 raišką.

4 METODĄ REALIZUOJANČIOS SISTEMOS KŪRIMAS

Šiame skyriuje realizuojamas metodo tyrimui reikalingas architektūrinis sprendimas.

4.1 Metodo pagrindimas

Metodas yra kuriamas naudoti nekomerciniais tikslais. Juo galės naudotis vartotojai siekiantys išgauti žmogaus rankos skeleto judesio atkūrimą iš video vaizdų, naudojantis lengvai prieinamomis priemonėmis, tokiomis kaip: internetinės kameros, kompiuteriai ir spalviniai žymekliai.

4.2 Sistemos architektūra

Sistemos architektūrinis sprendimas parodo ryšį tarp sistemos elementų, naudojamų judesio atkūrimo metodui išgauti iš video vaizdų.

4.2.1 Reikalavimų analizė

Išanalizavus funkcinis ir nefunkcinis, bei techninius reikalavimus (3 skyrius) nuspręsta, jog metodą turinti realizuoti sistema bus analizuojama sudarant UML diagramas:

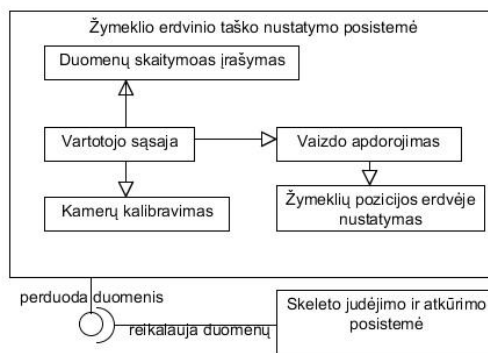
1. loginę sistemos diagramą;
2. vartotojo elgsenos diagramą;
3. sistemos veiklos diagramą;
4. klasių diagramą;
5. sukuriant sistemos realizacijos modelio diagramą;

4.2.2 Sistemos loginė architektūra

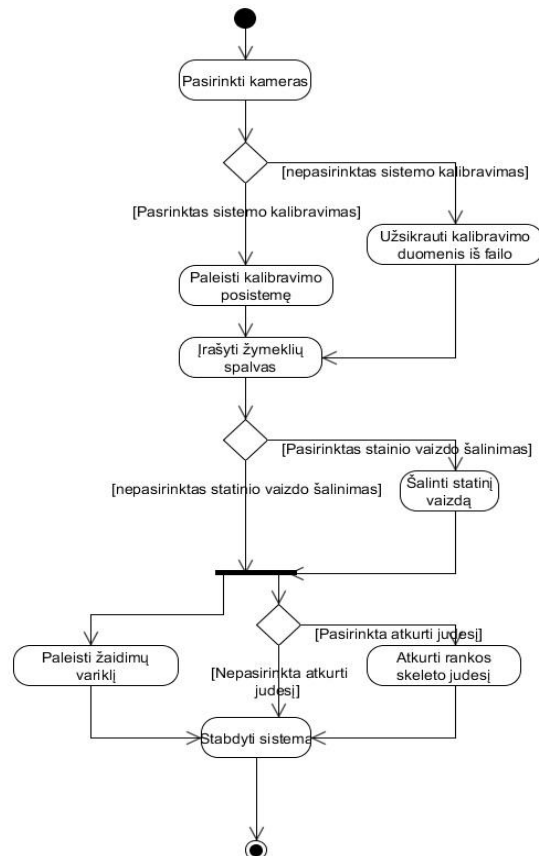
Sistemos loginė architektūra susideda iš 2-jų posistemių (24 pav.) kurios tarpusavyje veikia priklausomos tik nuo duomenų, žymeklio erdvinio taško nustatymo posistemė siunčia duomenis į skeleto judėjimo ir atkūrimo posistemę.

Remiantis sistemos logine architektūra (24 pav.) galima sudaryti vartotojo elgsenos diagramą (25 pav.). Elgsenos diagramoje yra nurodytas scenarijus kaip vartotojas gali elgtis, bei kokius veiksmus gali atlikti naudodamasis sistema. Paleidus sistemą vartotojas gali pasirinkti: ar užkrauti kamerų kalibravimo duomenis, ar naujai kalibruoti. Vartotojas turės įrašyti norimų žymeklių spalvas, kiekvieną kartą jungiant sistemą, dėl to, kad skirtingu dienos metu gali būti skirtingas apšvietimas. Taip pat naudotojas turi galimybę pasirinkti ar šalinti statinį vaizdą ar ne. Rekomenduojama statinį vaizdą šalinti kai apšvietimas nemeta šešėlių. Toliau reikalinga pasileisti skeleto atkūrimo posistemę, kuri veikia žaidimo variklio pagrindu, bei pasirinkti ar bus atkuriamas judesys kitoje posistemėje ar ne.

Pabaigus darbą spaudžiamas klavišas „Išeiti“ ir posistemės baigia darbą.



24 pav. Sistemos loginė architektūra



25 pav. Vartotojo elgsenos diagrama

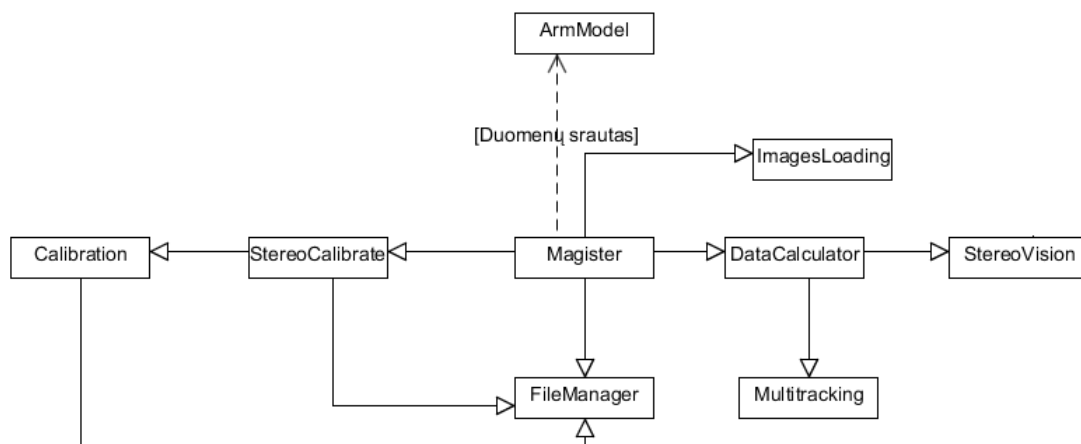
Sistemos būsenos diagrama (26 pav.) vaizduoja kokias būsenas gali įgauti sistema savo veikimo metu. Diagramoje atvaizduota, kaip vartotojui pasileidus abi posistemės automatiškai pradeda veikti tokiose būsenose.



26 pav. Būsenų diagrama

4.3 Sistemos detalus modelis

Metodui realizuoti reikalinga sistema pavaizduota UML klasių diagrama (27 pav.). Klasių diagramoje nurodyta ryšiai tarp klasių, bei jas sudarantys metodai.



27 pav. Supaprastinta klasių diagrama

Detalesnis klasių diagramoje sudarančių klasių metodų aprašymas yra pateikiamas lentelėje (6 lentelė.).

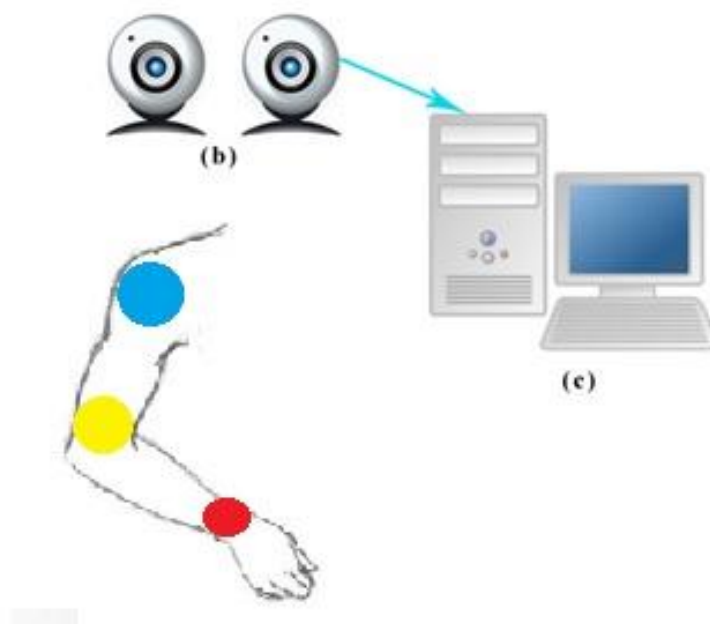
6 lentelė. Klasių diagramos aprašymas

Klasė	Apibūdinimas
Magister	Pagrindinė klasė apjungianti sistemą į vieną.
FileManager	Klasė skirta kamerų kalibracijos duomenų įrašymui ir skaitymui.
StereoCalibrate	Atlieka dviejų kamerų kalibravimo logiką.
Calibration	Atlieka vienos kameros kalibravimui reikalingą logiką.
ImagesLoading	Klasė skirta vaizdo iš kamerų filtravimui, žymeklių pozicijos vaizde nustatymui.
DataCalculator	Klasė iš gautų duomenų išskaičiuoja trimatį tašką.
Multitracking	Klasė skirta sekti keliems objektams vienu metu.
StereoVision	Klasė skaičiuojanti 3D tašką iš gautų 2D taškų.
ArmModel	Klasė esanti Panda 3D žaidimų variklyje atvaizduojanti rankos skeleto judėjimą pagal gautus duomenis iš Magister klasės.

4.4 Sistemos realizacijos modelis ir diegimas

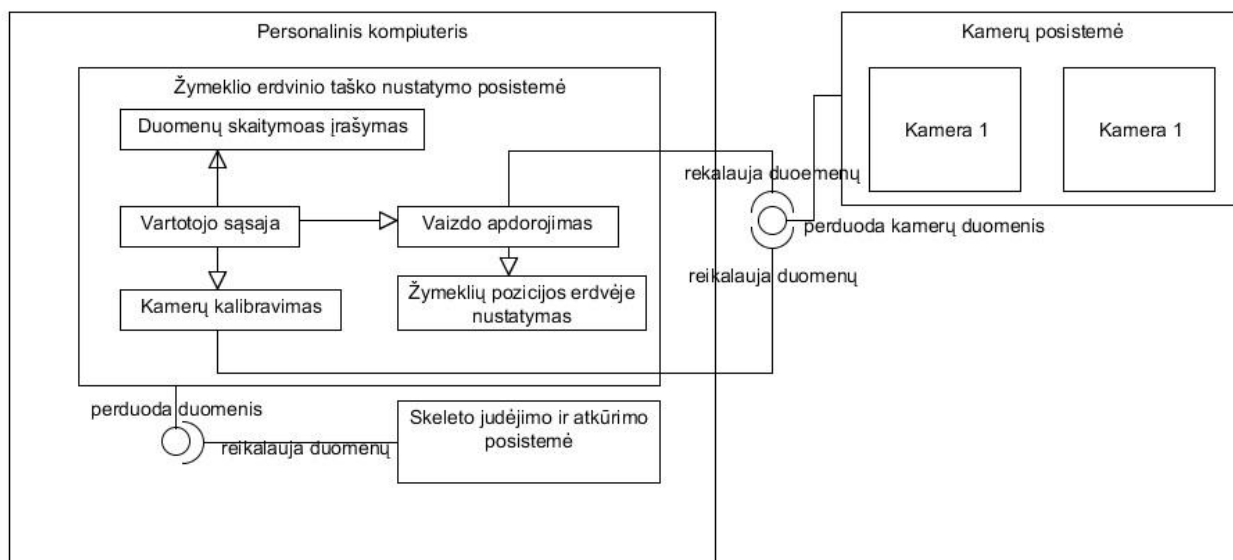
Metodui realizuoti reikalingą sistemą (28 pav.) sudaro trys techniniai komponentai: aktorius užsidėjęs spalvinius žymeklius (a), dviejų internetinių kamerų sistema (b), bei sistema apdorojanti gautą informaciją iš internetinių kamerų (c). Aktoriaus spalvinis žymeklio forma nėra svarbi, svarbiausia yra jo dydis ir spalvos ryškumas.

Detalesnė sistemos schema (29 pav.) atvaizduoja, kaip sistema bendrauja su posistemėmis ir kitais išoriniais sistemos komponentais. Galime pamatyti, kad visa reikalinga informacija bus gaunama iš kamerų.



28 pav. Metodo naudojimas: (a) aktorius užsidėjęs spalvinius žymeklius, (b) - dviejų internetinių kamerų sistema, (c) sistema apdorojanti vaizdo medžiagą

Metodui realizuoti ir tirti nereikalingas specifinis sistemos diegimas. Ši sistemą realizuojanti programa nukopijuojama į personalinio kompiuterio laikmeną iš duomenų laikmenos. Sistemos paleidimui reikalinga, kad personaliniame kompiuteryje būtų įdiegta *Python 2.7* versija, *Panda 3D 1.81* žaidimų variklis, bei *Eclipse* programavimo aplinka, o operacinė sistema gali būti tiek *Windows Xp/Vista/7*, tiek *Linux* operacinė sistema.



29 pav. Metodo realizacijos modelis

5 METODO TYRIMAS IR REZULTATŲ ANALIZĖ

5.1 Sukurto metodo tyrimo metodologija

Darbo metu sukurtas rankos skeleto judėjimo rekonstrukcija iš mažo detalumo video vaizdų metodas. Sekančiuose skyriuose atliekamas eksperimentinis metodo tyrimas siekiant išsiaiškinti sukurto metodo privalumus ir trūkumus, bei sudaryti rekomendacijas jo naudojimo rekomendacijas. Metodo tyrimo metu bus naudojama ta pati techninė įranga, tačiau bus keičiami programinės įrangos nustatymai.

5.2 Metodo eksperimentinio tyrimo aprašas

Sukurto metodas tyrimų sąrašas:

1. Metodo erdvinio taško rekonstrukcijos laiko priklausomybė nuo perduodamos skiriamosios gebos.
2. Metodo erdvinio taško rekonstrukcijos klaidų kiekio priklausomybė nuo perduodamos skiriamosios gebos.
3. Metodo toliausio taško nustatymo priklausomybė nuo perduodamos skiriamosios gebos.
4. Metodo rankos skeleto judesio rekonstrukcijos greitaveikos priklausomybė nuo judesio atlikimo laiko.
5. Metodo nuotolio nustatymas lyginant realų nuotolį su apskaičiuotu.

Metodo tyrimų aprašymas:

1. Patikrinti erdvinio taško rekonstrukcijos laiko priklausomybę naudojant skirtingas internetinių kamerų skiriamąsias gebas : 800x600, 640x480, 480x320, 320x240.
2. Išsiaiškinti pametamų žymeklių kiekį po 100 ištirtų kadrų naudojant skirtingas internetinių kamerų skiriamąsias gebas 800x600, 640x480, 480x320, 320x240.
3. Patikrinti didžiausią atstumą metrais nuo internetinių kamerų kuriuo yra fiksuojamas ir rekonstruojamas erdvinis taškas, taip pat naudojant skirtingas internetinių kamerų skiriamąsias gebas 800x600, 640x480, 480x320, 320x240.
4. Patikrinti kaip judesio atlikimo laikas lenkiant ranką per alkūnę įtakoja rankos skeleto judesio rekonstrukcija per 4, 3, 2, 1, 0,5 sekundes naudojant skirtingas internetinių kamerų skiriamąsias gebas 800x600, 640x480, 480x320, 320x240, įsivedant vertinimo parametrus 5-balėje sistemoje:
 - 5 - judesys rekonstruojamas tolygiai;
 - 4 - judesys rekonstruojamas tolygiai tačiau su vėlinimu;
 - 3 - judesys rekonstruojamas tolygiai, tačiau trūkinėja;
 - 2 - judesys rekonstruojamas su dideliais trūkumais, didelis vėlinimas, judesio trūkinėjimas;
 - 1 - judesys nerekonstruojamas..
5. Palyginti gautus gylio rezultatus skirtingose rezoliucijos esant skirtingam nuotoliui nuo kameros, kai žymeklis padedamas 30, 60, 90 cm atstumo nuo internetinių kamerų.

5.3 Metodo tyrimui naudojamos techninės ir programinės įrangos aprašas

Metodui tirti pasirinkta techninė įranga :

- Personalinis kompiuteris (7 lentelė.).
- Pasyvūs žymekliai, kurių skersmuo 5 cm (30 pav.).
- 2 x internetinės kameros STARTEC (8 lentelė.).

7 lentelė. Personalinio kompiuterio parametrai

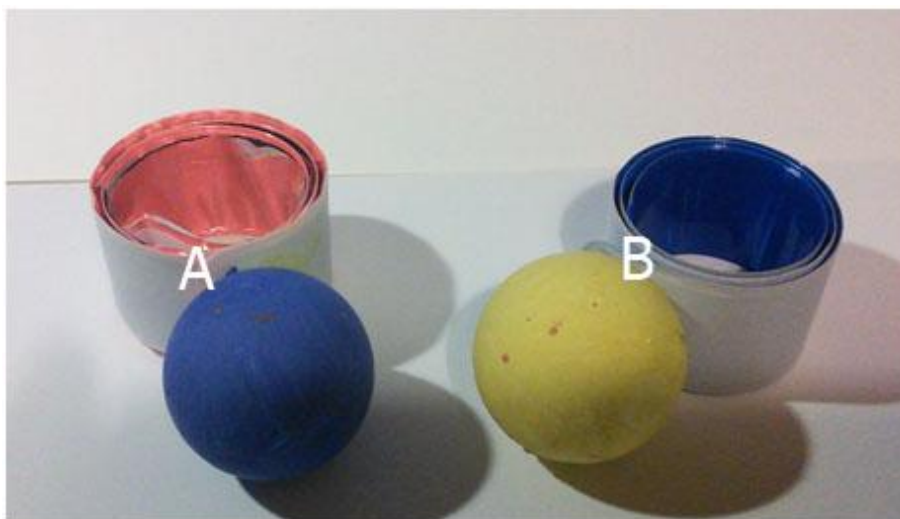
Komponentas	Specifikacija
Procesorius	INTEL CORE2DOU E8400 3 GHz
Procesoriaus branduolių skaičius	2
Atmintis (RAM)	4,00 GB
Grafinis procesorius	AMD HD4850
Video atmintis	512 MB GDDR3
Universalioji jungtis	USB 2.0
Operacinė sistema	Windows 7 64 bit

8 lentelė. Internetinės kameros parametrai

Komponentas	Specifikacija
Skiriamoji geba	1280 x 1024
Greitaveika	30 fps
Universalioji jungtis	USB 2.0

Metodui tirti panaudota programinė įranga ir bibliotekos:

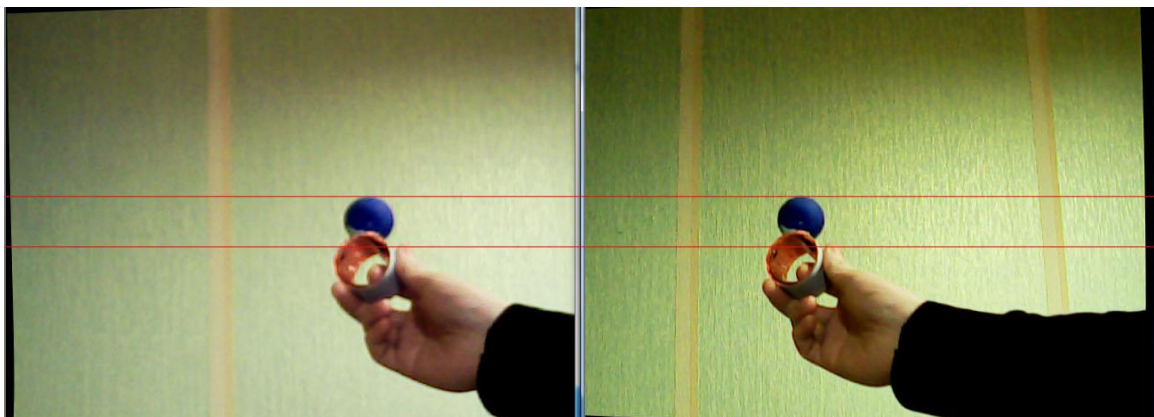
- OPEN CV 2.49 vaizdo apdorojimo biblioteka.
- Panda 3D 1.81 žaidimų variklis.
- Blender 2.7



30 pav. Pasyvūs žymekliai, A - mėlynos spalvos ir B - geltonos spalvos žymekliai

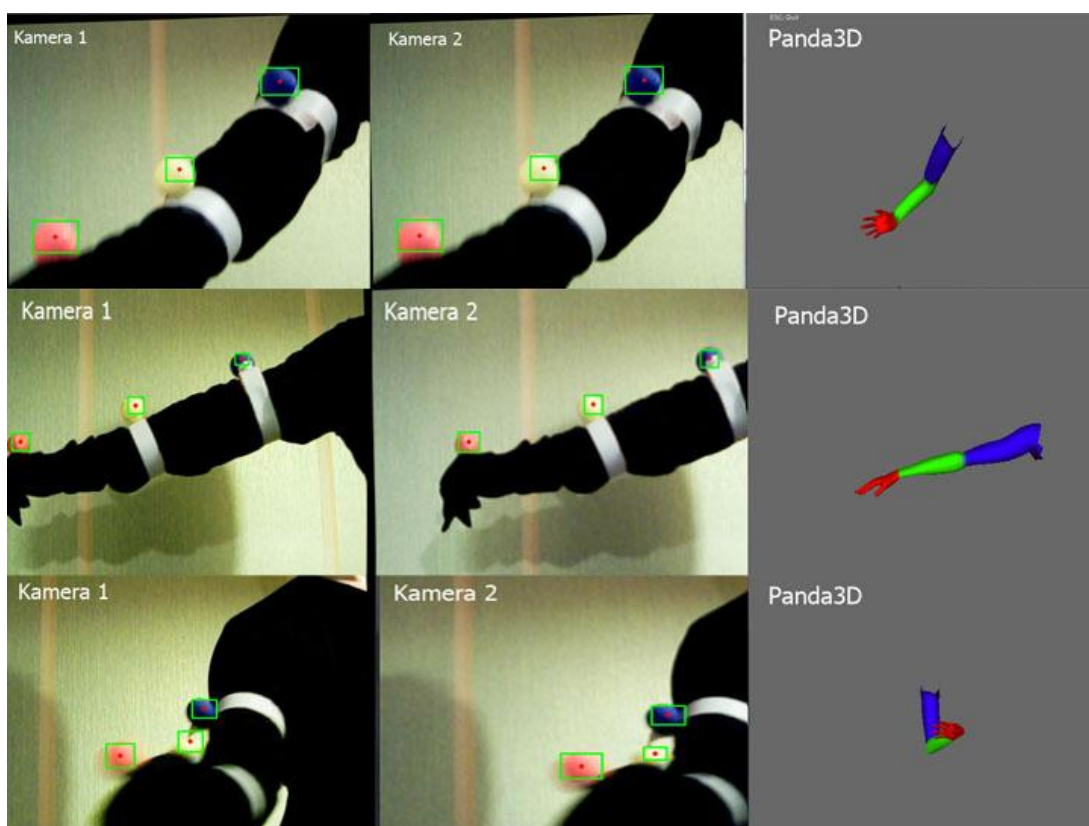
Metodas eksperimentiškai ištestuotas su pateikta programine ir technine įranga (7 lentelė. ir 8 lentelė.), todėl naudojant kitokius metodo realizacijos elementus, nei pateikti šiame darbe, gauti rezultatai gali skirtis.

Norint ištestuoti metodą, reikėjo sukurti sistemą gebančią rekonstruoti rankos skeleto judėjimą iš video vaizdų, naudojant pasyviuos žymeklius, dvi internetines kameras, bei šiuos komponentus jungiančią sistemą. Tam reikėjo atlikti kamerų kalibraciją naudojamų kamerų suvienodinimui x ašyje (31 pav.).



31 pav. Kamerų vaizdų suvienodinimas x ašyje

Atlikus kamerų suvienodinimą buvo galima naudoti sistemą realizuojančią rankos skeleto rekonstrukciją iš video vaizdų (32 pav.) ir taip eksperimentiškai ištirti darbe sukurtą metodą.



32 pav. Realizuoto metodo veikimo demonstravimas

5.4 Tyrimo rezultatai

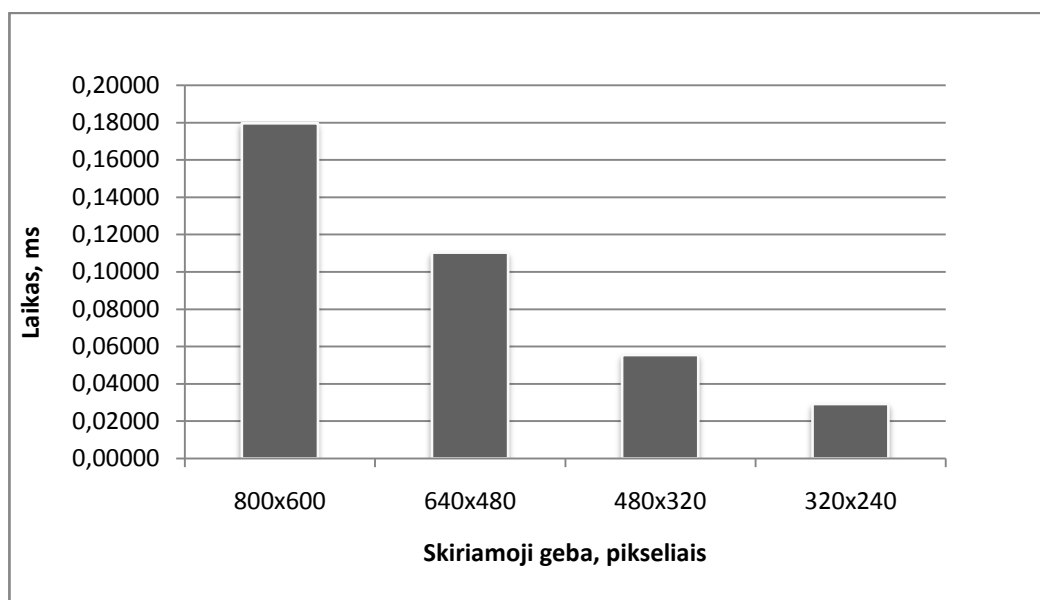
5.4.1 Metodo erdvinio taško rekonstrukcijos laiko priklausomybė nuo perduodamos skiriamosios gebos

Šiame eksperimente reikėjo išsiaiškinti erdvinio taško rekonstrukcijos laiko priklausomybę naudojant skirtingas internetinių kamerų skiriamąsias gebas : 800x600, 640x480, 480x320, 320x240. Naudojant kiekvieną skiriamąją gebą buvo atlikta po 10 skaičiavimų kuriais nustatyta laiko priklausomybė milisekundėmis ir išvestas vidurkis (9 lentelė.).

9 lentelė. Trimačio taško priklausomybė nuo laiko

Eil. Nr.	Skiriamoji geba			
	800x600	640x480	480x320	320x240
1	0,18000	0,11500	0,05500	0,02900
2	0,18000	0,11000	0,05500	0,02900
3	0,17900	0,11000	0,05500	0,02900
4	0,17800	0,10900	0,05500	0,02800
5	0,17800	0,11100	0,05500	0,02900
6	0,17800	0,10800	0,05600	0,02900
7	0,17800	0,11100	0,05600	0,03000
8	0,18300	0,10900	0,05700	0,03000
9	0,18200	0,11000	0,05400	0,02900
10	0,17900	0,10900	0,05600	0,02900
Vidurkis	0,17950	0,11020	0,05540	0,02910

Rezultatus iliustruojantis grafikas (33 pav.) parodo, jog mažėjant skiriamajai gebai greitėja ir taško rekonstrukcija erdvėje - palyginimui 800x600 rekonstrukcijos laikas nuo 320x240 skiriamosios gebos skiriasi net ~83%.



33 pav. Trimačio taško rekonstrukcijos priklausomybė nuo laiko

Galima teigti, jog mažėjant skiriamajai gebai proporcingai mažėja ir trimačio taško atkūrimo laikas, visa tai yra todėl, kad naudojant mažesnę skiriamąją gebą sumažėja ir duomenų kiekis kuriuos reikia apdrikti sistemai, dėl to skaičiavimai tampa greitesni.

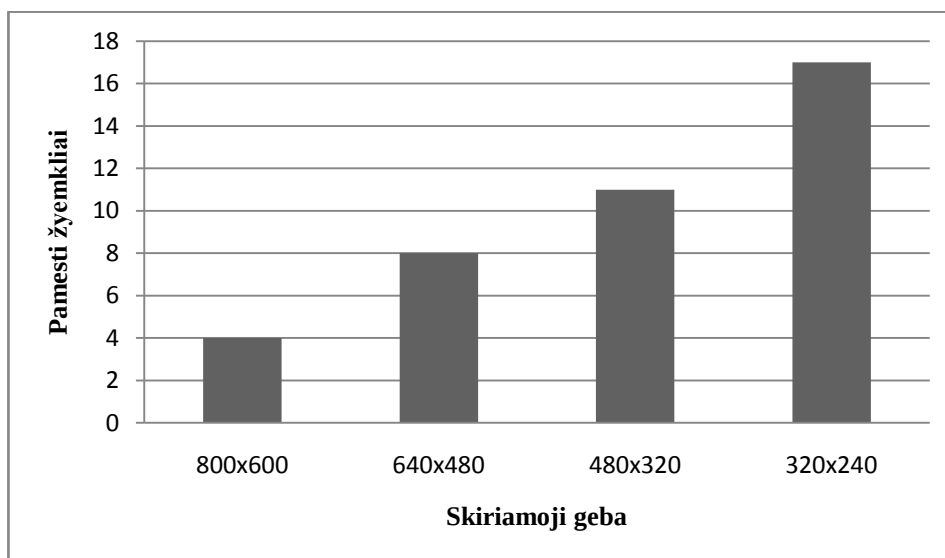
5.4.2 Metodo erdvinio taško rekonstrukcijos klaidų kiekio priklausomybė nuo perduodamos skiriamosios gebos

Šio eksperimento tikslas - nustatyti kaip skiriamoji geba įtakoja pasyvių žymeklių praradimo kiekį per 100 trimačio taško atkūrimo iteracijų. Šiam eksperimentui taip pat buvo naudota tokios pat skiriamosios gebos kaip ir 5.4.1 skyriuje, bei sudaryta lentelė (10 lentelė.) kurioje galime puikiai matyti, jog mažėjant skiriamajai gebai didėja ir klaidų kiekis.

10 lentelė. Pamestų žymeklių kiekis per 100 iteracijų naudojant skirtingas skiriamąsias gebas

Skiriamoji geba	800x600	640x480	480x320	320x240
Pamesti žymekliai	4	8	11	17

Žiūrint į grafiką (34 pav.) ir lentelę (10 lentelė.) galime teigti, jog metodas mažėjant skiriamajai gebai pametamų žymeklių kiekis svyruoja nuo 27% iki 50 %.



34 pav. Pamestų žymeklių kiekis per 100 trimačio taško atkūrimo iteracijų

5.4.3 Metodo toliausio taško nustatymo priklausomybė nuo perduodamo skiriamosios gebos

Ištyrus metodą pagal tai kaip įtakuoja jo erdvinio taško atkūrimo atstumas nuo skiriamosios gebos (11 lentelė.) gauti netikėti rezultatai. Rezultatai rodo, jog didžiausias atstumas nuo kamerų, naudojant 5 cm pasyviuos žymeklius, yra 3,6 metro. Tačiau mažinant rezoliuciją atstumas nesikeitė ir išliko 3,2 metro.

Gavus tokius rezultatus galima teigti, jog metodas gali rekonstruoti rankos skeleto judėjimą nuo 3,2 iki 3,6 metrų atstumu nuo kameros, neprarandant žymeklių.

11 lentelė. Atkuriamo taško atstumo priklausomybė nuo skiriamosios gebos

Skiriamoji geba	800x600	640x480	480x320	320x 240
Pamesti žymekliai	3,6	3,2	3,2	3,2

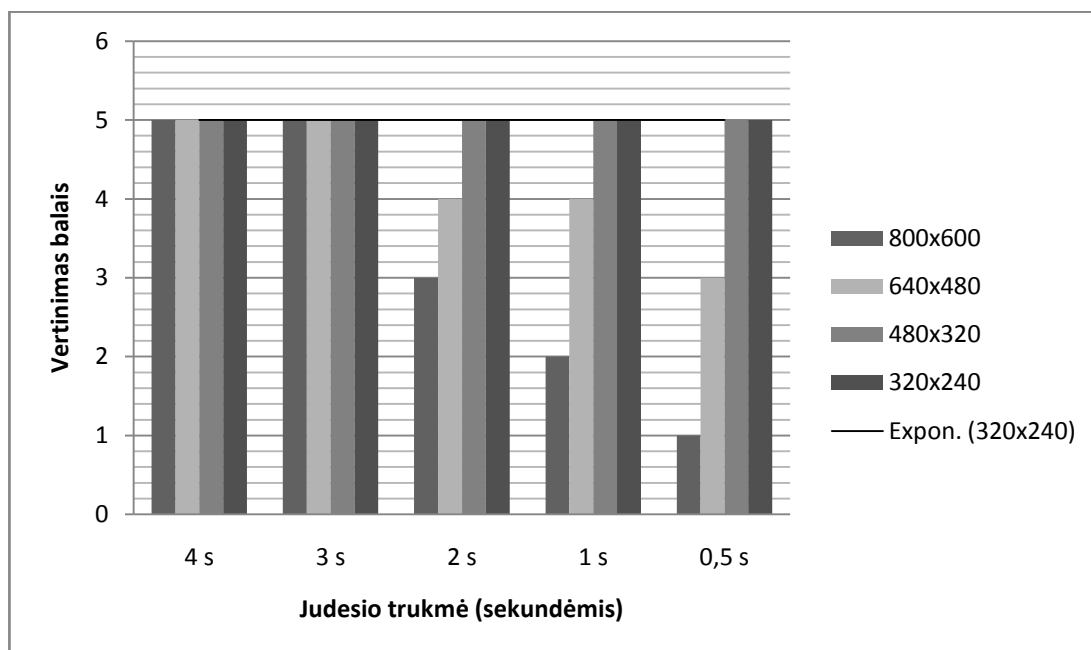
5.4.4 Metodo rankos skeleto judesio rekonstrukcijos greitaveikos ir tikslumo priklausomybė nuo judesio atlikimo laiko

Atlikus metodo rankos skeleto judesio rekonstrukcijos greitaveikos priklausomybės nuo judesio greičio tyrimą ir rezultatą įvertinus 5-kiabalėja sistemoje. Gauti rezultatai aprašomi lentelėje (12 lentelė.) kur surašyti tyrimo vertinimai, kai judesiui atkurti yra naudojami trys žymekliai. Matome (35 pav.), jog kuo didesnė raiška tuo judesio greitaveikos įvertis yra mažesnis, kai judesys greitėja arba tampa staigus. Naudojant skiriamąsias gebas 480x320, bei 320x240, matome, kad galima lengvai apdoroti staigius judesius kurių greitumas iki 0,5 sekundės.

12 lentelė. Metodo rankos skeleto judesio rekonstrukcijos greitaveikos ir tikslumo vertinimas

Eil. Nr.	Judėsio trukė	Skiriamoji geba			
		800x600	640x480	480x320	320x240
1	4 s	5	5	5	5
2	3 s	5	5	5	5
3	2 s	3	4	5	5
4	1 s	2	4	5	5
5	0,5 s	1	3	5	5

Iš rezultatų galime teigti, kad metodas gali veikti didesne judesio atkūrimo greitime naudojant mažesnes skiriamąsias gebas dėl mažesnio duomenų kiekio kurį reikia apdoroti sistemai.



35 pav. Judesio rekonstrukcijos greitimeikos ir tikslumo vertinimas

5.4.5 Metodo nuotolio nustatymas lyginant realų nuotolį su apskaičiuotu

Metodas buvo tiriamas prieš abiejų kamerų centrą 30, 60 ir 90 cm atstumu padedant žymeklį, tada tikrinama kokių nuotolį nuo kameros iki žymeklio gražina sistema esant skirtingai skiriamajai gebai, bei išskaičiuojant santykį kuris yra tarp gražintos ir realios reikšmės.

13 lentelė. Nuotolio nuo kameros matavimas

Atstumas	Skiriamoji geba							
	800x600	santykis	640x480	santykis	480x320	santykis	320x240	santykis
30	15	2	32	0,9375	29	1,034483	42	0,714286
60	45	1,333333	32	1,875	67	0,895522	83	0,722892
90	35	2,571429	52	1,730769	100	0,9	127	0,708661

Rezultatuose (13 lentelė.) matome, jog tiksliausiai kai santykis yra apie 1 metodas gražina esant 480x320 skiriamajai gebai, o blogiausias rezultatas esant 800x600.

5.5 Metodo taikymo rekomendacijos

Atlikus metodo eksperimentinį tyrimą galima sudaryti metodo taikymo rekomendacijas.

Metodui įgyvendinti rekomenduojama turėti kuo naujesnį ir geresnį personalinį kompiuterį, taip pat galima atskirti žymeklių sekimo posistemę nuo žymeklių atvaizdavimo posistemės, jas naudojant skirtinguose įrenginiuose.

Metodo korektiškam veikimui rekomenduojama 9 m² patalpa. Patalpos apšvietimas turi būti nuolatinis - nekintantis, taip pat reikalingas vienspalvis fonas.

Metodui realizuoti ir rankos sąnariams perduoti rekomenduojama vienam sąnariui turėti vieną ar daugiau žymeklių, tai priklauso nuo turimų kompiuterinių resursų.

5.5.1 Metodo sistemos kūrimo ekonominiai skaičiavimai

Metodo sistemai sukurti pakako nusipirkti techninės įrangos už 34 litus (14 lentelė.) neįskaičiavus jau turimo kompiuterio ir programinės įrangos sukūrimo kaštų.

14 lentelė. Sistemos kaina

	Kaina	Kiekis	Suma (Lt)
Internetinė kamera	12	2	24
Teniso kamuoliukai	6	1	6
Pirstiniu komplektas	4	1	4
		Viso:	34

Tačiau biudžetą pakėlus keliais šimtais litų ir susipirkus geresnius elementus galima būtų turėti judesio sekimo sistemą, kuri galėtų atkurti puikiai visą žmogaus skeletą, kai kiekvienai porai kamerų būtų naudojama po vieną kompiuterį, kad ir po *Raspberry Pi*. Tada kiekvienos kamerų poros duomenis apdorotų vienas kompiuteris ir gražintų į motininį kompiuterį, kuris apjungtų duomenis ir atvaizduotų skeleto judėjimą. Tokios sistemos kaina galėtų siekti vos 1576 litus (15 lentelė.).

15 lentelė. Teorinė judesio sekimo sistemos kaina

	Kaina	Kiekis	Suma (Lt)
Raspberry Pi kompiuteris	130	3	390
Kameros	30	6	180
Teniso kamuoliukai	6	1	6
Nešiojamas kompiuteris	1000	1	1000
		Viso:	1576

Remiantis *MetaMotion.com* [30] duomenimis *iPiSoft Desktop Motion Capture* sistema su keturiomis kameromis kainuoja 2080 lt (be kompiuterio), todėl galima suprasti, jog daug tobulesnė sistemos versija labiau apsimokėtų, nei pigiausia šiuo metu esanti judesio atkūrimo sistema.

6 IŠVADOS

1. Nustatyta, kad greičiausias ir paprasčiausias skeleto judėjimo rekonstravimo metodas iš video vaizdų realizuojamas naudojant dvi internetines kameras su spalviniais žymekliais, nes naudojant daugiau kamerų ir bežymeklinį skeleto rekonstravimo metodą, reikia daugiau kamerų ir skaičiuojamosios galios apdoroti gautai iš kamerų vaizdinei informacijai.
2. Suformavus funkcinis, nefunkcinis ir techninius reikalavimus, pastebėta, jog sistemos darbo našumas labiausiai priklausys nuo aparatinės įrangos greಿತaveikos.
3. Apibrėžus rankos skeleto judesio rekonstrukcijos metodo architektūrinį sprendimą, pastebėta, kad metodą techniškai sukomplektuoti yra paprasta, tačiau programiškai realizuoti yra sudėtinga.
4. Tyrimas parodė, kad mažėjant skiriamajai gebai, greitėja taško rekonstrukcija erdvėje, todėl 800x600 rekonstrukcijos laikas nuo 320x240 skiriamosios gebos padidėja net ~83%.
5. Ištyrus erdvinio taško rekonstrukcijos klaidų kiekio priklausomybę nuo perduodamos skiriamosios gebos, nustatyta, kad mažėjant skiriamajai gebai pametamų žymeklių kiekis svyruoja nuo 27% iki 50 %.
6. Pagal tyrimo rezultatus nustatyta, kad metodas gali rekonstruoti rankos skeleto judėjimą nuo 3,2 iki 3,6 metrų atstumu nuo kameros, neprarandant žymeklių pozicijos erdvėje, todėl metodo realizavimui pakanka 9 m² ploto patalpos.
7. Tyrimas parodė, kad realizuojant tiriamą metodą, jis tiksliausiai ir greičiausiai rekonstravo rankos skeleto judėjimą naudojant 480x320 skiriamąją gebą, nors ir 1,75 karto dažniau pametė žymeklį nei naudojant 800x600 skiriamąją gebą.
8. Ištyrus metodo nuotolio nustatymą, pastebėta, kad metodas veikia tiksliausiai naudojant 480x320 skiriamąją gebą, todėl šią skiriamąją gebą yra tikslingiausia taikyti skeleto judesio rekonstravimo metode.

7 NAUDOTA LITERATŪRA

1. L. Herda, L. Herda, P. Fua, R. Plankers, R. Boulic and D. Thalmann. *Skeleton-Based motion capture for robust reconstruction of human motion*. 2000. 2-5p. [Tinkle] Prieiga per internetą: <http://cvlab.epfl.ch/files/content/sites/cvlab2/files/publications/publications/2000/HerdaFPBT00.pdf> [kreiptasi 2012-10-12]
2. Ben Kuchera, *A look at animation in video games: why motion capture is the start, not the end, of the work* [Tinkle]. Prieiga per internetą: <http://www.penny-arcade.com/report/article/animation> [kreiptasi 2012-12-12]
3. Leonid Sigal. *Lecture 3, (Marker-based) Motion Capture* [Tinkle]. Prieiga per internetą: <http://www.cs.cmu.edu/~yaser/Lecture-3-MarkerBasedMocap.pdf>. [kreiptasi 2012 12 16]
4. Richard Roesler. *Optical Motion Capture guide*. [Tinkle] Prieiga per internetą: http://physbam.stanford.edu/cs448x/old/Optical_Motion_Capture_Guide.html [kreiptasi 2012 12 15]
5. Gutemberg B. Guerra-Filho. *Optical Motion Capture: Theory and Implementation*. 2005. 6-24p. [Tinkle] Prieiga per internetą: <http://smile.uta.edu/Guerra-FilhoRITA05optical.pdf> [kreiptasi 2012 12 05]
6. Frederic Besse, Nicolas Heynen, Bertrand Jamet, Patrice Lacour. *Cheap Motion Capture*. [Tinkle] Prieiga per internetą: <http://motion.capture.free.fr/project.html>. [kreiptasi 2013 03 05]
7. Zhengyou Zhang. *Camera calibration*. 2004. 2-3p. [Tinkle] <http://research.microsoft.com/en-us/um/people/zhang/Papers/Camera%20Calibration%20-%20book%20chapter.pdf> [kreiptasi 2013 05 24]
8. H. Zollner and R. Sablatnig. Comparison of methods for geometric camera calibration using planar calibration targets. p 1-5. [Tinkle] Prieiga per internetą: <http://www.caa.tuwien.ac.at/cvl/people/sab/papers/oagm04b.pdf> [kreiptasi 2012 12 05]
9. Peter Hillman. *White Paper: Camera Calibration and Stereo Vision*. 2005. 1-7p. [Tinkle] Prieiga per internetą: <http://www.peterhillman.org.uk/downloads/whitepapers/calibration.pdf> [kreiptasi 2013 04 05]
10. Tsai *Camera Calibration*. [Tinkle] Prieiga per internetą: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/DIAS1/ [kreiptasi 2013 03 17]
11. Shree Nayar. *Computer Vision Laboratory: Big Shot Camera. Brightness Control*. 2009. [Tinkle] Prieiga per internetą: <http://www.bigshotcamera.com/sections/learn/proc/brightness.html> [kreiptasi 2013 04 06]
12. Fr. Don Matthys, S.J. *Digital Image Processing. Contrast Stretching*. 2001 [Tinkle] Prieiga per internetą: <http://academic.mu.edu/phys/matthysd/web226/Lab01.htm> [kreiptasi 2013 04 06]
13. Rafel C. Gonzalez, Richard E. Woods. *Digital Image Processing. Second Edition*. 119-123p. [Tinkle] Prieiga per internetą: http://sgar91.files.wordpress.com/2011/04/dip_gonzalez_2e_full.pdf [kreiptasi 2013 05 26]
14. Robert Y. Wang, Jovan Popovic. *Real-Time Hand Tracking*. 2-3p. [Tinkle] Prieiga per internetą: <http://people.csail.mit.edu/rywang/handtracking/s09-hand-tracking.pdf> [kreiptasi 2013 06 01]
15. Liang Zhang, Guido Brunneth, Stephan Rusdorf. *Real-time Human Motion Capture with Simple Marker Sets and Monocular Video*. 2011. [Tinkle] Prieiga per internetą: <http://www.jvrb.org/past-issues/8.2011/2819> [kreiptasi 2013 06 03]
16. Duncan Fyfe Giles. *Graphics Lecture 18* 3-5p. [Tinkle] Prieiga per internetą: <http://www.doc.ic.ac.uk/~dfg/graphics/GraphicsLecture18.pdf> [kreiptasi 2013 06 03]
17. Steve Gu, Ying Zheng, Carlo Tomasi. *Efficient Visual Object Tracking with Online Nearest Neighbor Classification*. [Tinkle] Prieiga per internetą: <http://www.cs.duke.edu/~tomasi/papers/gu/guAccv10.pdf> [kreiptasi 2013 02 03]
18. Erik Cuevas, Daniel Zaldivar, Raul Rojas. *Kalman filter for vision tracking*. 2005. [Tinkle] Prieiga per internetą: <http://www.inf.fu-berlin.de/users/rojas/2005/tr-b-05-12.pdf> [kreiptasi 2013 04 02]
19. Changjiang Yang, Ramani Duraiswami, Larry Davis. *Fast Multiple Object Tracking via a Hierarchical Particle Filter*. [Tinkle] Prieiga per internetą: <http://webcache.googleusercontent.com/search?q=cache:IjG9Dh0LNX8J:citeseerx.ist.psu.edu/viewdoc/download%3Fdoi%3D10.1.1.79.5947%26rep%3Drep1%26type%3Dpdf&cd=2&hl=lt&ct=clnk&gl=lt&client=firefox-a> [kreiptasi 2013 05 07]
20. Tomas Svoboda. *Kanade-Lucas-Tomasi Tracker*. 2007. [Tinkle] Prieiga per internetą: <http://cmp.felk.cvut.cz/cmp/courses/XE33PVR/WS20072008/Lectures/Motion/klt-print.pdf> [kreiptasi 2012 11 07]
21. Nitesh Patel. *Compare The Pros And Cons Of C, Java And C++*. 2010. [Tinkle] Prieiga per internetą: <http://goarticles.com/article/Compare-The-Pros-And-Cons-Of-C-Java-And-C/3669527/> [kreiptasi 2012 06 07]
22. Dakusan's Domanai. *Python Pros and Cons*. 2009. [Tinkle] Prieiga per internetą: http://www.castledragmire.com/Posts/Python_Pros_and_Cons/ [kreiptasi 2012 06 07]
23. SimpleCV Blog. *Opencv vs matlab vs simplecv* [Tinkle] Prieiga per internetą: <http://simplecv.tumblr.com/post/19307835766/opencv-vs-matlab-vs-simplecv> [kreiptasi 2012 12 15]
24. Utkarsh. *Opencv vs vxl vs lti performance test*. 2010. [Tinkle] Prieiga per internetą: <http://www.aishack.in/2010/07/opencv-vs-vxl-vs-lti-performance-test/> [kreiptasi 2012 12 15]
25. Robert Cong, Ryan Winters. *How it Works: Xbox Kinect*. [Tinkle] Prieiga per internetą: <http://www.jameco.com/Jameco/workshop/howitworks/xboxkinect.html> [kreiptasi 2012 12 15]
26. Wesley Fenlon. *How Playstation Move Works*. [Tinkle] Prieiga per internetą: <http://electronics.howstuffworks.com/playstation-move2.htm> [kreiptasi 2012 12 15]

27. FreeTrack. *How it Work*. [Tinkle] Prieiga per internetą: <http://www.free-track.net/english/freetrack/comment-ca-marche.php?PHPSESSID=5e78617a1d147c114edb8bd3a3e6bf42> [kreiptasi 2012 12 15]
28. Qualysis. Prieiga per internetą: <http://www.qualisys.com/products/hardware/> [kreiptasi 2014 05 16]
29. TCP vs UDP. Prieiga per internetą: http://www.diffen.com/difference/TCP_vs_UDP [kreiptasi 2014 05 16]
30. Motion Capture Prices. Prieiga per internetą: <http://www.metamotion.com/FAQ/prices.html> [kreiptasi 2014 05 16]