



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
ELEKTRONINIŲ SISTEMŲ KATEDRA

Dmitrijus SKUNČIKAS

**OBJEKTŲ ATPAŽINIMO, NAUDOJANT SPALVOS IR FORMOS
POŽYMIŲ RINKINIUS, TYRIMAS**
**RESEARCH ON OBJECT RECOGNITION USING COLOR AND
SHAPE FEATURE DESCRIPTORS**

Magistro baigiamasis darbas

Informatikos inžinerijos studijų kryptis
Informacinių elektroninių sistemų studijų programa, valstybinis kodas 621E15003
Atvirojo kodo sistemų specializacija

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
ELEKTRONINIŲ SISTEMŲ KATEDRA

TVIRTINU
Katedros vedėjas

(parašas)
prof. habil. dr. R. Martavičius
2013 m. _____ mėn. ____ d.

Dmitrijus SKUNČIKAS

**OBJEKTŲ ATPAŽINIMO NAUDOJANT SPALVOS IR FORMOS
POŽYMIŲ RINKINIUS TYRIMAS**
**RESEARCH ON OBJECT RECOGNITION USING COLOR AND
SHAPE FEATURE DESCRIPTORS**

Magistro baigiamasis darbas

Informatikos inžinerijos studijų kryptis
Informacinių elektroninių sistemų studijų programa, valstybinis kodas 621E15003
Atvirojo kodo sistemų specializacija

Vadovas
Lietuvių kalbos konsultantė

dr. D. Matuzevičius
(mokslinis vardas ir laipsnis, vardas, pavardė)
lekt. V. Buivydienė
(mokslinis vardas ir laipsnis, vardas, pavardė)

(parašas)

(data)
2013 06-03

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
ELEKTRONINIŲ SISTEMŲ KATEDRA

Technologijos mokslų sritis

Elektros ir elektronikos inžinerijos mokslo kryptis

Informatikos inžinerijos studijų kryptis

Informacinių elektroninių sistemų studijų programa, valst. kodas 621E15003

Atvirojo kodo sistemų specializacija

TVIRTINU

Katedros vedėjas

prof. habil. dr. R. Martavičius

2013 m. 04 mėn. 4 d.

MAGISTRO BAIGIAMOJO DARBO
UŽDUOTIS

Studentui

Dmitrijus SKUNČIKAS

Baigiamojo darbo tema: Objektų atpažinimo, naudojant spalvos ir formos požymių rinkinius, tyrimas

Research On Object Recognition Using Color And Shape Feature Descriptors

Patvirtinta 2014 m. 10 mėn. 28 d. dekanų įsakymu Nr. 245el

Baigiamojo darbo užbaigimo terminas 2013 m. 06 mėn. 4 d.

Darbo tikslas – ištirti objektų atpažinimo metodus, grįstus spalvos ir formos požymių rinkiniais.

Darbo uždaviniai

1. Atlikti analitinę objektų atpažinimo metodų, grįstų spalvos ir formos požymių rinkiniais, apžvalgą.
2. Atrinkti perspektyvius objektų atpažinimo metodus, grįstus spalvos ir formos požymių rinkiniais.
3. Ištirti atrinktus objektų atpažinimo metodus, grįstus spalvos ir formos požymių rinkiniais, pateikiant efektyvumo rezultatus.

Aiškinamojo rašto turinys

1. Įvadas
 - 1.1. Darbo aktualumas ir tikslas
 - 1.2. Darbo uždaviniai
 - 1.3. Naudoti tyrimo ir analizės metodai
 - 1.4. Darbo naujumas ir praktinė nauda
 - 1.5. Darbo struktūra
2. Objektų atpažinimo naudojant požymių rinkinius analitinė apžvalga
3. Objektų atpažinimo metodų, grįstų spalvos ir formos požymių rinkiniais, analizė

4. Duomenų atranka objektų atpažinimo metodų tyrimui
5. Objektų atpažinimo metodų tyrimo rezultatai
6. Apibendrinimas. Išvados

Literatūra

Priedai

A priedas. Programos išeities tekstas

B priedas. Pranešimo 16-oje Lietuvos jaunųjų mokslininkų konferencijoje medžiaga

Baigiamojo darbo rengimo konsultantas:

—
(mokslinis vardas ir laipsnis, vardas, pavardė)

Vadovas

doc. dr. Dalius Matuzevičius

(mokslinis vardas ir laipsnis, vardas, pavardė)

Užduotį gavau


(parašas)

(parašas)

2013 m. 05 mėn. 02 d.

Vilniaus Gedimino technikos universitetas
Elektronikos fakultetas
Elektroninių sistemų katedra

ISBN ISSN
Egz. sk.
Data-.....-.....

Antrosios pakopos studijų **Informacinių elektroninių sistemų** programos baigiamasis darbas
Pavadinimas **Objektų atpažinimo, naudojant spalvos ir formos požymių rinkinius, tyrimas**
Autorius **Dmitrijus Skunčikas**
Vadovas **dr. Dalius Matuzevičius**

Kalba: lietuvių

Anotacija

Objektų atpažinimo naudojant spalvos ir formos požymių rinkinius tyrimas. Magistro baigiamasis darbas informatikos inžinerijos laipsniui. Vilniaus Gedimino technikos universitetas. Vilnius, 2013. 60 p. teksto be priedų, 36 iliustr., 8 lent., 43 bibl., 3 priedai. Magistro baigiamojo darbo tikslas – ištirti trimačių (3D) objektų atpažinimo metodus, grįstus trimačiais (3D) spalvos ir formos požymių rinkiniais. Atlikta analitinė objektų atpažinimo metodų, grįstų spalvos ir formos požymių rinkiniais, literatūros apžvalga. Atrinkti bei išanalizuoti perspektyvūs objektų atpažinimo metodai, grįsti spalvos ir formos požymių rinkiniais. Ištirti atrinktieji objektų atpažinimo metodai bei pateikti jų efektyvumo rezultatai. Ištirtas objektų atpažinimo efektyvumas keičiant objektų paviršiaus įvertinimo parametrus, būdingųjų taškų išrinkimo metodus ir jų parametrus, požymių skaičiavimo metodus (deskriptorius) ir jų parametrus, įvertintas klaidingų atitikimų filtravimo RANSAC metodo poveikis. Išrinktas efektyviausias objektų atpažinimo metodas, kuris atpažįsta objekto kategoriją – 100 %, objektą – 88 %.

Prasminiai žodžiai: 3D objektų atpažinimas, taškų debesis, būdingųjų taškų išskyrimas, SIFT, 3D požymių deskriptorius, FPFH, PFH, PFHRGB, SHOT, COLORSHOT, RANSAC, PCL, C++.

Vilnius Gediminas Technical University
Faculty of Electronics
Department of Electronic Systems

ISBN ISSN
Copies No.
Date-.....-.....

Master Degree Studies **Information Electronics Systems** study programme Final Work

Title **Research on Object Recognition Using Color and Shape Feature Descriptors**

Author **Dmitrijus Skunčikas**

Academic supervisor **Dr Dalius Matuzevičius**

Thesis language:
Lithuanian

Annotation

Research on Object Recognition Using Color and Shape Feature Descriptors. Master thesis for Informatics Engineering master degree. Vilnius Gediminas technical university. Vilnius, 2013. Thesis consists of: 60 p. text without appendixes, 36 figures, 8 tables, 43 bibliographical entries, 3 appendixes included. The aim of master's thesis is to investigate the 3D object recognition methods using 3D color and shape feature descriptors. The analytical review of object recognition methods is performed, reviewing the abilities of algorithms implementation based on color and shape feature descriptors. The promising elected recognition algorithms are analyzed. The recognition algorithms are investigated and effectiveness results are produced. The Efficiency of recognition methods is investigated by changing evaluation parameters of objects' surface, descriptors and their parameters, influence of reject's of correspondences method RANSAC. According to test results the most effective object recognition method recognizes correct object category 100%, objects 88%.

Keywords: Keywords: 3D object recognition, point cloud, point of interest extraction, SIFT, 3D features descriptor, FPFH, PFH, PFHRGB, SHOT, COLORSHOT, RANSAC, PCL, C++.

Vilniaus Gedimino technikos universiteto egzaminų,
sesijų ir baigiamųjų darbų rengimo bei gynimo
organizavimo tvarkos aprašo 2011-2012 m. m.
1 priedas

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Dmitrijus Skunčikas, 20075551

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Elektronikos fakultetas

(Fakultetas)

Informacinės elektroninės sistemos, AKSfm-11

(Studijų programa, akademinė grupė)

**BAIGIAMOJO DARBO (PROJEKTO)
SĄŽININGUMO DEKLARACIJA**

2013 m. birželio 4 d.

Patvirtinu, kad mano baigiamasis darbas tema „Objektų atpažinimo, naudojant spalvos ir formos požymių rinkinius, tyrimas“ patvirtintas 2011 m. spalio 28 d. dekanų potvarkiu Nr. 245el, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Mano darbo vadovas dr. Dalius Matuzevičius.

Kitų asmenų indėlio į parengtą baigiamąjį darbą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).

(Parašas)

Dmitrijus Skunčikas

(Vardas ir pavardė)

TURINYS

Žymenys ir santrumpos	8
1. Įvadas	11
1.1. Darbo aktualumas ir tikslas	11
1.2. Darbo uždaviniai	11
1.3. Naudoti tyrimo ir analizės metodai	11
1.4. Darbo naujumas ir praktinė nauda	12
1.5. Darbo struktūra	12
2. Metodų, grįstų spalvos ir formos požymių rinkiniais, analitinė apžvalga	13
2.1. Trimačio taško požymių reprezentacija	14
2.2. Normalės ir jų reikšmė	16
2.3. Objekto išskyrimas	18
2.4. Būdingųjų taškų išskyrimas	20
2.5. Požymių deskriptoriai	21
2.6. Požymių palyginimas	26
2.7. Trumpas skyriaus apibendrinimas	29
3. Perspektyvių objektų atpažinimo metodų analizė	30
3.1. PFH ir PFHRGB požymių įvertinimo metodas	30
3.2. FPFH požymių įvertinimo metodas	34
3.3. SHOT ir COLORSHOT požymių įvertinimo metodas	36
3.4. „UniformSampling“ būdingųjų taškų išskyrimo metodas	39
3.5. SIFT būdingųjų taškų išskyrimo metodas	41
3.6. Kd-tree požymių palyginimo metodas	44
3.7. RANSAC metodas	45
3.8. Trumpas skyriaus apibendrinimas	47
4. Tyrimas ir tyrimo rezultatai	49
4.1. Eksperimentinių duomenų atranka	49
4.2. Metodų parametrų nustatymas	51
4.3. 3D Objektų atpažinimo metodų tikslumas	55
4.4. RANSAC metodo įtaka objektų atpažinimo tikslumui	58
4.5. Metodų veikimo sparta	61
4.6. Trumpas skyriaus apibendrinimas	64
5. Apibendrinimas. Išvados	66
Literatūra	68
A priedas. Pranešimo 16-oje Lietuvos jaunųjų mokslininkų konferencijoje medžiaga	71
B priedas. Įgyvendintų algoritmų pirminis programos tekstas	77
C priedas. Kompaktinė plokštelė	83

Žymenys ir santrumpos

Žymenys

k	imties numeris
p_1, p_2	du taškai erdvėje
t_1, t_2	du laiko momentai
F	taško požymio vektorius
p_q	pasirinktas taškas
P^k	kaimynystėje esančių taškų vektorius
d	atstumo vertinimo metrika (Euklido atstumas)
Γ	panašumo matas
C	kovariacinė matrica
λ	kovariacinės matricos eigen vertės
$\vec{v}$	kovariacinės matricos eigen vektorius
$\vec{n}$	normalės vektorius
α, ϕ, θ	normalių kampinių požymių tripletas
u, v, w	geometrinių požymių tripletas
f	požymio vertė
b	požymio poskyriu skaičius
O	algoritmų laiko sudėtingumas
θ_i	kampai tarp normalių
$SH_{G,f}$	SHOT histograma
n_{vi}	izotropinio sferinio tinklelio skyrius
n_u	sferinio tinklelio normalė taškui u
$D(P)$	SHOT deskriptorius taškui P
Q	atraminių taškų sferinis plotas
G	SHOT taško požymių vektorius
$p()$	SHOT požymių vertinimo funkcija
R_P	taško P spalvos intensyvumų reikšmė
R_Q	sferinio ploto Q spalvos intensyvumų reikšmė
L	mastelio erdvės funkcijos reikšmė
G	Gauso branduolys
$*$	sąsūkos operacija
I	vaizdo funkcija
D	Gauso funkcijų skirtumas

σ	glotninimo spindulys
$\hat{x}$	x vektoriaus poslinkis
H	Heso matrica
Tr	matricos pėdsakas (įstrižainės reikšmės)
Det	matricos determinantas
α, β	didžiausia ir mažiausia eigen vertės
r	santykis tarp mažos ir didelės eigen vertės
l_V	vokselio kraštinės ilgis
R_D	kaimyninių taškų paieškos, apskritimo spindulys
S_R	RANSAC metodo, atstumas tarp artimiausių taškų

Santrumpos

2D	dviejų matmenų aplinka
3D	trijų matmenų aplinka
6D	šešių matmenų aplinka
3D PC (angl. <i>3D Point Cloud</i>)	trimatis taškų debesis
PCL (angl. <i>Point Cloud Library</i>)	taškų apdorojimo biblioteka
GMM (angl. <i>Gaussian Mixture Model</i>)	Gauso sumaišymo modelis
PFH (angl. <i>Point Feature Histograms</i>)	taško požymio histogramos
PFHRGB (angl. <i>Point Feature Histograms RGB</i>)	taško požymio histogramos su spalvos spalvų maišymo sistema, raudona (angl. <i>Red</i>), žalia (angl. <i>Green</i>) ir mėlyna (angl. <i>Blue</i>)
FPFH (angl. <i>Fast Point Feature Histograms</i> + RGB)	greitos taško požymio histogramos informacija
SPFH (angl. <i>Simplified Point Feature Histogram</i>).	supaprastinta taško požymio histograma
VFH (angl. <i>Viewpoint Feature Histogram</i>)	stebėjimo taško požymio histograma
6DoF (angl. <i>Six Degrees of Freedom</i>)	šeši laisvės laipsniai
CVFH (angl. <i>Clustered Viewpoint Feature Histogram</i>)	klasterizuota stebėjimo taško požymio histograma
SDC (angl. <i>Shape distribution Component</i>)	formos pasiskirstymo dedamoji
3DSC (angl. <i>3D shape context</i>)	trimatis formos kontekstas
NaN (angl. <i>Not a Number</i>)	neskaičius

USC (angl. <i>Unique Shape Context</i>)	unikalus formos kontekstas
EVD (angl. <i>EigenVector Decomposition</i>)	eigen vektorių dekompozicija
RIFT (angl. <i>Rotation Invariant Feature Transform</i>)	pastovaus sukimo požymio transformavimas
SIFT (angl. <i>Scale-Invariant Feature Transform</i>)	pastovios skalės požymio transformavimas
DoG (angl. <i>Difference of Gaussian</i>)	Gausianų skirtumai
PPF (angl. <i>Point Pair Feature</i>)	poros taškų požymis
ESF (angl. <i>Ensemble of Shape Functions</i>)	formos funkcijų visuma
PDF (angl. <i>Probability Density Function</i>)	tikimybinė tankio funkcija
PCE (angl. <i>Principal Curvatures Estimation</i>)	pagrindinio kreivumo įvertinimas
SHOT (angl. <i>Signature of Histograms of Orientations</i>)	orientacijos histogramų signatūra
CSHOT arba COLORSHOT (angl. <i>Color Signature of Histograms of Orientations</i>)	orientacijos histogramų signatūra su spalvos informacija
GPU (angl. <i>Graphics Processing Unit</i>)	grafinis procesorius
EGI (angl. <i>Extended Gaussian Image</i>)	Išplėstas Gauso vaizdas arba normalių sfera
PCA (angl. <i>Principal Component Analysis</i>)	pagrindinių komponentų analizė
LRF (angl. <i>Local Reference Frame</i>)	lokalį atskaitos sistema
TLS (angl. <i>Total Least Squares</i>)	mažiausių kvadratų suma
FLANN (angl. <i>Fast Library for Approximate Nearest Neighbors</i>)	greitoji, artimiausių kaimynų, aproksimavimo biblioteka
RANSAC (angl. <i>RANdom SAmple Consensus</i>)	atsitiktinės imties konsensusas

1. ĮVADAS

1.1. Darbo aktualumas ir tikslas

Daugelyje sričių nuolatos susiduriama su objektų atpažinimu. Objektų atpažinimo terminą galima apibrėžti kaip objektų identifikavimą panaudojant sukaupią objektų duomenų bazę. Viena svarbiausių objektų atpažinimo taikymo sričių – kompiuterinė rega. Svarbiausia kompiuterinės regos užduotis yra nustatyti 3D paviršių panašumus, šiuo gebėjimu sprendžiamos objektų atpažinimo, paviršių palyginimo užduotys. Objektų atpažinimo sistemoms pavedamos vis sudėtingesnės atpažinimo užduotys, to pasekoje sistemoms keliami vis didesni reikalavimai – norima, kad sistemose įgyvendinti, objektų atpažinimo, algoritmai veiktų sparčiau, tiksliau, būtų atsparūs įvairiems triukšmams, apšvietimo bei objekto pozos pokyčiams.

Magistro baigiamojo darbo tema aktuali dėl objektų atpažinimo sudėtingumo, kuris pasireiškia tuo, kad vis dar nepavyksta sukurti objektų atpažinimo sistemų, kurios būtų tokios efektyvios kaip žmogaus gebėjimas atpažinti objektus. Ieškoma intelektualių būdų bei metodų, kurių pagalba panaudojant skaitmenines priemones, galima būtų priartėti prie žmogaus gebėjimų atpažinti aplinkos objektus.

Darbo tikslas – ištirti trimačių (3D) objektų atpažinimo metodus, grįstus trimačiais (3D) spalvos ir formos požymių rinkiniais.

1.2. Darbo uždaviniai

Baigiamajam darbui atlikti iškelti trys pagrindiniai uždaviniai:

1. Atlikti analitinę 3D objektų atpažinimo metodų, grįstų 3D spalvos ir formos požymių rinkiniais, apžvalgą.
2. Atrinkti perspektyvius 3D objektų atpažinimo metodus, grįstus 3D spalvos ir formos požymių rinkiniais.
3. Ištirti atrinktus 3D objektų atpažinimo metodus, grįstus 3D spalvos ir formos požymių rinkiniais, pateikiant efektyvumo rezultatus.

1.3. Naudoti tyrimo ir analizės metodai

Objektų atpažinimo metodų tyrimui ir efektyvumui nustatyti taikytas eksperimentinis metodas. Eksperimentiniai tyrimai atlikti panaudojant programinės įrangos *Microsoft .NET* platformos modulį, kuriame C++ programavimo kalba įgyvendintas skaitmeninių vaizdų apdorojimas.

Panaudotas PCL bibliotekos modulis su skaitmeninių vaizdų apdorojimui skirtais metodais. Darbe taikyti taškų registravimo, filtravimo, aproksimavimo, taškų požymių klasifikavimo metodai.

1.4. Darbo naujumas ir praktinė nauda

Baigiamajame darbe nagrinėjama šiai dienai aktuali 3D objektų atpažinimo tema. Objektų atpažinimas tai kompiuterinės regos tyrimo objektas, kuris suteikia galimybę robotams, mašinoms bei intelektualioms sistemoms atlikti aukštesnio lygio užduotis. Kuo sudėtingesnės užduotys pavedamos atpažinimo sistemoms tuo geriau joms reikia perteikti mus supančią aplinką.

Darbe ištirti 3D objektų atpažinimo metodai, kurie pagrįsti spalvos ir formos požymių rinkiniais. Tyrimo medžiaga ir rezultatai gali būti panaudoti tolesniems objektų atpažinimo metodų tyrimams. Gauti metodų efektyvumo rezultatai suteikia galimybę, pagal spendžiamą užduotį, pasirinkti ir įgyvendinti labiausiai tinkamą – efektyviausią objektų atpažinimo metodą. Ištirti 3D objektų atpažinimo metodai gali būti sėkmingai taikomi daugybėje praktinio panaudojimo sričių bei įgyvendinti objektų atpažinimo sistemose, kurios praktikoje naudojamos objektų atpažinimo užduotims spręsti. Sudarytas 3D objektų atpažinimo metodas, tinkamas praktiškai įgyvendinti kompiuterinėje ar robotų regoje ir spręsti 3D objektų atpažinimo užduotis.

1.5. Darbo struktūra

Pirmajame skyriuje aptartas darbo aktualumas – siekis efektyviai atpažinti 3D objektus, išskeltas darbo tikslas, įvardinti pagrindiniai darbo uždaviniai, paminėti darbe naudoti analizės ir tyrimo metodai, įvardintas baigiamojo darbo naujumas bei praktinė nauda.

Antrajame skyriuje aptartas objektų atpažinimo procesas, metodika bei pagrindinės, objektų atpažinimo procesą sudarančios, dalys. Atlikta, objektų atpažinimo procese naudojamų, metodų, grįstų spalvos ir formos požymių rinkiniais, analitinė apžvalga.

Trečiajame skyriuje pateikiama atrinktų perspektyvių objektų atpažinimo metodų, kurie grįsti spalvos ir formos požymių rinkiniais, analizė.

Ketvirtajame skyriuje aptartas eksperimentinių duomenų atrankos procesas bei kriterijai, kuriais buvo remiamasi atliekant duomenų atranką. Pateikiamas objektų atpažinimo metodų parametrų nustatymas, įgyvendinimas, tyrimo rezultatai.

Paskutiniame skyriuje pateikiamas viso darbo apibendrinimas ir tyrimų išvados.

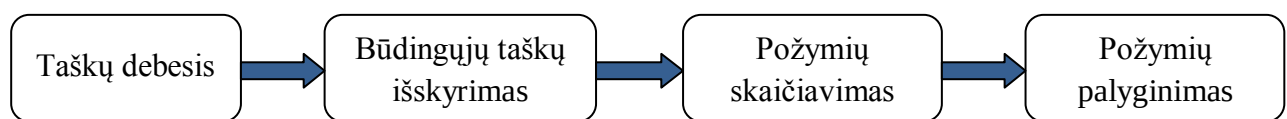
Magistro darbą sudaro 5 skyriai, literatūros sąrašas ir priedai. Darbo skyriai: Įvadas, Metodų, grįstų spalvos ir formos požymių rinkiniais, analitinė apžvalga, Perspektyvių objektų atpažinimo metodų analizė, Tyrimas ir tyrimo rezultatai, Apibendrinimas. Išvados.

Darbo apimtis: 60 puslapių, 36 paveikslai, 8 lentelės, 43 literatūros šaltiniai ir 3 priedai.

2. METODŲ, GRĮSTŲ SPALVOS IR FORMOS POŽYMIŲ RINKINIAIS, ANALITINĖ APŽVALGA

Objekto atpažinimas yra technologija, kuri gali lyginti įvairius vieno objekto požymius su kitų objektų požymiais. Vis didesnio tyrimų dėmesio sulaukia trimačių (3D) objektų atpažinimo technologijos, taigi atsirado daug norinčių tobulinti 3D aplinkos registravimo įrangą ir sistemas. Neseniai atsiradus nebrangiems, trimatės aplinkos registravimo įrenginiams – jutikliams, vienas iš tokių jutiklių yra gerai žinomas – „Kinect“, labai populiari tapo trimačių taškų debesų (angl. *3D point cloud*) technologija. 3D taškų debesį sudaro iš 3D objektų paviršių gauta 3D taškų visuma. 3D taškas lyginant su dvimačio (2D) vaizdo tašku turi papildomą matmenį – gylio informaciją. Gylio jutiklių dėka galima registruoti 3D objektus, aplinką, gaunant spalvos ir gylio duomenis.

Trimačių taškų debesims iš aplinkos paviršių surinkti ir apdoroti reikalingos tam skirtos priemonės ir metodai. 3D taškų debesims surinkti gali būti panaudotas tam skirtas jutiklis arba kamera, kuri gali pateikti gylio duomenis – atstumą iki objekto. Labiausiai išplėtotą, taškų debesims apdoroti, skirta biblioteka – PCL (angl. *Point cloud library*) [1] suteikia galimybę programiškai apdoroti gautus skaitmeninius duomenis bei kurti įvairias vaizdų apdorojimo sistemas, tarp jų ir objektų atpažinimo. Šiame darbe bus aptarinėjami PCL 1.6 versijos metodai ir jų parametrai.



1 pav. Pagrindiniai objekto atpažinimo žingsniai

1 paveiksle pateikiama preliminarinė 3D objekto atpažinimo veiksmų grandinė. Objektų atpažinimas susideda iš 4 pagrindinių žingsnių. Pirmame žingsnyje jutikliu skenuojamas aplinkos paviršius ir gaunamas trimačių taškų debesys. Kitame žingsnyje, iš gauto taškų debesies išskiriamas objektas, tai yra norimo objekto trimačių taškų išskyrimas iš gauto taškų debesies, ši procedūra vadinama segmentavimu. Gautas objekto taškų debesys filtruojamas, pašalinami nereikalingi triukšmai bei mažinamas duomenų tankumas. Po to surandami būdingieji taškai, kuriuose kitu žingsniu įvertinami objekto požymiai – skaičiuojami deskriptoriai [1]. Paskutiniame etape lyginami būdingųjų taškų suskaičiuoti požymiai – deskriptoriai su duomenų bazėje saugomais objektų požymių deskriptoriais. Objektas atpažįstamas pagal jam būdingus bruožus dar vadinamus požymiais. Pagrindiniai objekto požymiai yra forma, paviršiaus kreivumas ir spalva. Vienas iš svarbiausių, 3D taškų debesų objektų atpažinimo technologijos, žingsnių yra objekto trimačių taškų požymių įvertinimas – deskriptorių skaičiavimas, todėl deskriptorių skaičiavimui bus skiriamas didelis dėmesys.

2.1. Trimačio taško požymių reprezentacija

Jau minėta, kad trimatis taškas turi papildomą matmenį ir jis apibrėžiamas trimis koordinatėmis – Dekarto koordinatės sistemos pradžios atžvilgiu (X, Y, Z). Darant prielaidą kad koordinatės sistemos pradžia laikui bėgant nekinta, tai erdvėje turimi du taškai p_1 ir p_2 laiko momentais t_1 ir t_2 įgija tas pačias 3D koordinates. Taigi, čia susiduriama su taip vadinamos „blogos pozicijos“ problema, kadangi net jei šių taškų atstumai 3D erdvėje pagal Euklido atstumą yra lygūs, jie gali būti paimti iš visiškai skirtingų paviršių. Taigi, čia turima omenyje, kad aplinka ir paviršius laiko momentais t_1 ir t_2 gali pasikeisti ir šių trijų koordinatės jau nepakanka taškų skirtumams surasti. Reikalingi papildomi požymiai, kurių pagalba galima būtų 3D erdvėje atskirti taškus [2].

Objektų atpažinimas – viena iš sričių, kurioje taškų palyginimas bei objekto požymių įvertinimas yra svarbiausias žingsnis. Objektų atpažinimas reikalauja geresnių taško charakteristikų ir metrikų, tam kad galima būtų rasti skirtumą tarp geometrinių paviršių. Kai kurie aplinkos skenavimo įrenginiai gali suteikti dar ir papildomą informaciją, tokią kaip intensyvumas, paviršiaus mažėjimo vertė ar spalva, tačiau šios priemonės vis dėlto iki galo neišsprendžia minėtos problemos [2]. Taigi, trimačio taško sąvoka kaip visuma su Dekarto koordinatėmis išnyksta ir jos vietoje atsiranda nauja – lokalus deskriptorius. Literatūroje galima rasti labai daug aprašymų skirtingų tos pačios koncepcijos schemų, tokių kaip formos deskriptorius ar geometrinis požymis. Šiame darbe deskriptoriui bus priskiriamas toks pavadinimas kokius objekto požymius jis įvertina.

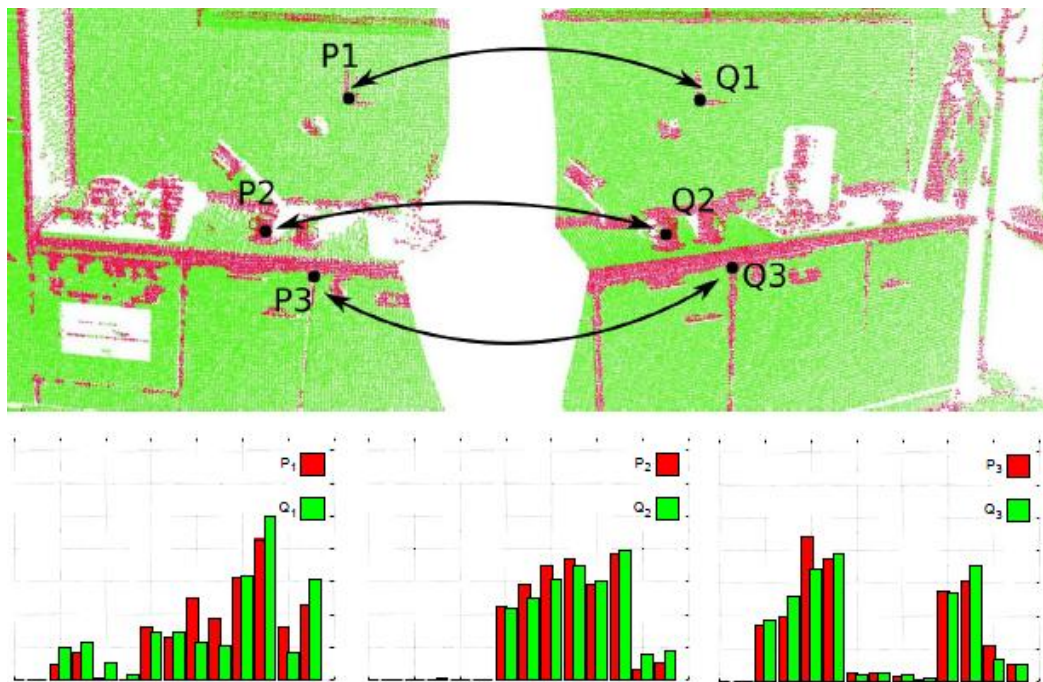
Kaip jau buvo minėta, pagrindiniai objekto požymiai yra forma, paviršiaus kreivumas ir spalva. Šiuos objekto požymius dalinai apibrėžia kiekvienas paviršiaus taškas, taigi turi būti vertinami ne vien tik pavienių taškų požymiai bet ir aplink juos esančių kaimyninių taškų požymiai. Čia atsiranda kaimyninių taškų panaudojimo koncepcija, iš kurios iškyla trimačio taško požymio apibrėžimas. Nagrinėjamo paviršiaus taškas gali būti aprašytas požymio vektoriumi F :

$$F(p_q, P^k) = \{x_1, x_2, x_3 \dots x_n\}, \quad (2.1)$$

čia p_q – pasirinktas taškas, P^k – artimiausių kaimynystėje esančių taškų vektorius, kur x_i , $i \in \{1, 2, 3 \dots n\}$ nurodo atstojamojo požymio vektoriaus dydį. Kai lyginami keli pasirinkti iš paviršiaus taškai p_1 ir p_2 palyginami yra jų atstojamieji vektoriai F_1 ir F_2 (2.1).

$$\Gamma = d(F_1, F_2), \quad (2.2)$$

čia Γ – matas apibūdinantis taškų p_1 ir p_2 panašumą, d – pasirinkta metrika atstumui įvertinti. 1 paveiksle pateikiamas pavyzdys, kaip PCL bibliotekoje, histogramos pavidalu pateikiami atstojamųjų taško požymių vektoriaus vertės.



2 pav. Taško požymių reprezentacija dvejose skirtingose duomenų bazėse [2]

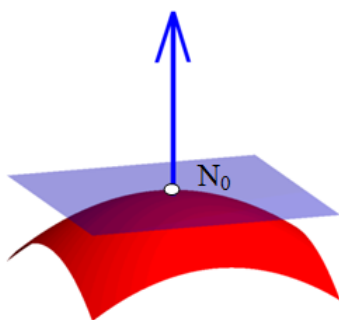
Tokiu būdu, galima išspręsti iškilusią taškų palyginimo problemą ir nesudėtingai atskirti geometrinius paviršius, taip pat ir objektus esančius tame paviršiuje. Geriausiu atveju atstojamieji požymiai bus labai panašūs tiems taškams, kurie bus tame pačiame arba panašiam paviršiuje ir skirtingi tiems taškams, kurie bus surasti skirtinguose paviršiuose (2 pav.). Taigi, reikia surasti „gerus“ – tinkamus atstojamuosius požymius, kurių diskriminantinė galia būtų kuo didesnė. Iškyla klausimas kuo skiriasi „geras“ taško požymis nuo „blogo“. Geras paviršiaus požymis turi pakankamai gerai fiksuoti pasirinkto paviršiaus charakteristikas [2]:

- griežta transformacija – 3D sukimasis bei duomenų konvertavimas, neturi turėti įtakos atraminio požymio vektoriaus skaičiavimui;
- įvairus pavyzdžių tankumas – didesnio ar mažesnio tankumo paviršiaus fragmentas, turi turėti tokia pačią požymio vektoriaus signatūrą;
- atsparumas triukšmams – taško požymis turi išlaikyti tokia pačią ar panašią vertę šiame požymio vektoriuje, esant nedideliui triukšmui duomenyse [2];

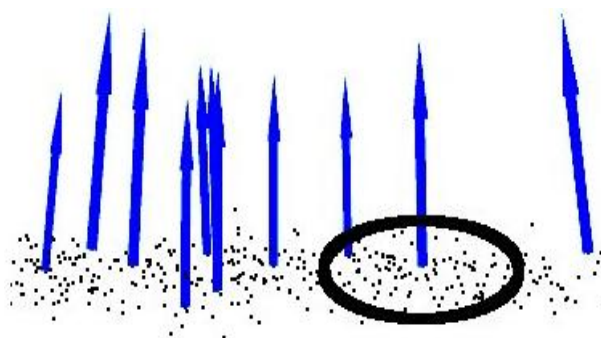
Taigi, šioje dalyje buvo aptartas 3D objektų atpažinimui svarbus trimačio taško požymio teorinis modelis bei pateiktas taškų palyginimo problemos sprendimo būdas. Trimačiai taškai gali suteikti tam tikros informacijos apie objekto paviršių ir jo formą. Norint įvertinti paviršiaus parametrus reikalingi tam skirti metodai, kurie galėtų įvertinti ne vieno taško požymius o kelių aplinkui esančių taškų požymius. Taigi visų pirma, 2.2 skyriuje, bus aptartas pagrindinis ir paprasčiausias, paviršiaus parametrų įvertinimo metodas – normalės ir kaip jų pagalba nustatoma paviršiaus geometrija. Verta pabrėžti, kad galimi ir kiti duomenys, kurie gali padėti apibrėžti paviršiaus parametrus. Apie juos bus kalbama deskriptorių aptarimo skyriuje.

2.2. Normalės ir jų reikšmė

Paviršiaus normalės yra svarbi paviršiaus savybė, jos dažnai naudojamos kompiuterinėje grafikoje norint tinkamai pritaikyti šviesos šaltinius, kurie generuoja šešėlius ir kitus vizualinius efektus. Normale paviršiaus taške N_0 yra vadinama tiesė, kuri eina per tašką N_0 statmenai liečiančios plokštumos paviršiaus šitame taške (3 pav., a). Šiuo atveju normalė o tiksliau jos kryptis ir kampas su plokštuma, naudojami paviršiaus parametrų nustatymui.



(a) Paviršiaus normalė [3]



(b) Paviršiaus, kaimyninių taškų, normalės [4]

3 pav. Paviršiaus normalių vaizdavimas

Kaip jau buvo minėta, atraminis taško požymis sudaromas atsižvelgiant į artimoje kaimynystėje esančius taškus (3 pav., b). Normalės nurodo geometrinių paviršių kryptis, taigi jų pagalba galima nustatyti ir geometrinius paviršiaus požymius. Kai skaičiuojamos paviršiaus normalės yra sprendžiama liestinės priderinimo prie paviršiaus problema. Kitaip tariant, kai skaičiuojama normalė, reikia tinkamai nustatyti liestinę, kuri liečia nagrinėjamą plokštumą. Yra daug įvairių normalių skaičiavimo metodų, aptarsime pati paprasčiausią ir dažniausiai naudojamą. Minėta problema sprendžiama mažiausių kvadratų plokštumos prie paviršiaus priderinimu.

$$x = \bar{p} = \frac{1}{k} \sum_{i=1}^k p_i, \quad (2.3)$$

čia x – plokštumos taškas, k – artimiausių taškų skaičius artimiausioje kaimynystėje p_i . Iš užklausių artimiausių kaimyninių taškų sudaromos kovariacinės matricos $\mathbf{C}$, bei analizuojami šių matricų eigen vektoriai ir eigen vertės.

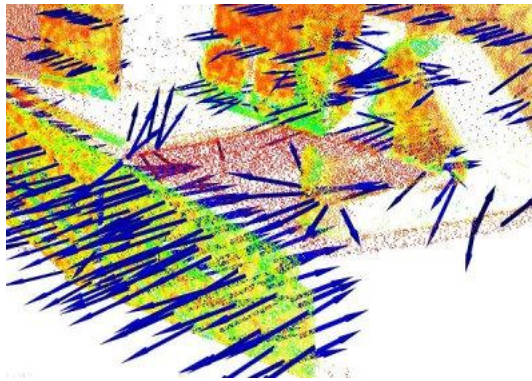
$$\mathbf{C} = \frac{1}{k} \sum_{i=1}^k (p_i - \bar{p}) \cdot (p_i - \bar{p})^T, \quad \mathbf{C} \cdot \vec{v}_j = \lambda_j \cdot \vec{v}_j, j \in \{1, 2, 3\}, \quad (2.4)$$

$\bar{p}$ pateikia 3D centroidą artimiausių kaimyninių taškų, λ_j – kovariacinės matricos j -oji realios eigen vertės, $\vec{v}_j$ – eigen vektorius. Siekiama, kad eigen vektorius $\vec{v}_0$ atitiktų mažiausią eigen vertę, tadą

aprokcimacijos rezultatas yra normalės aplinka $\vec{n} = \{n_x, n_y, n_z\}$. Tada, šią normalę $\vec{n}$ galima apibrėžti dviem parametrais (kampų pora) sferinėje normalių aplinkoje:

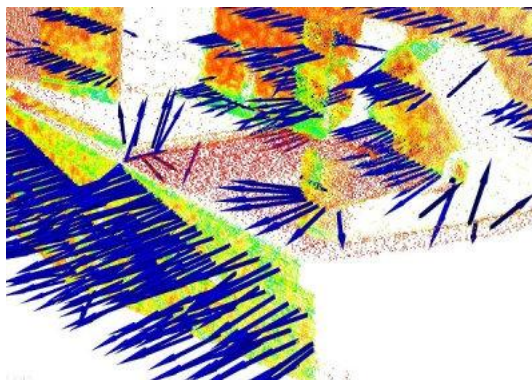
$$\phi = \arctan\left(\frac{n_z}{n_y}\right), \theta = \arctan\left(\frac{\sqrt{n_y^2 + n_z^2}}{n_x}\right), \quad (2.5)$$

Žinoma, kad įvertintos normalės ir jų sudaromi kampai su plokštumą, matematiškai nesudėtingai apskaičiuojami. Normalės taip pat turi ir kryptį – orientaciją, kurią matematiškai sunku aprašyti, todėl taikomi kiti metodai, tokie kaip pagrindinių komponentių analizė PCA (angl. *Principal Component Analysis*). Minėtus apskaičiavimus ir normalių orientaciją galima patrikti normalių sfera arba kitaip vadinamu Gauso vaizdu EGI (angl. *Extended Gaussian Image*) [5].



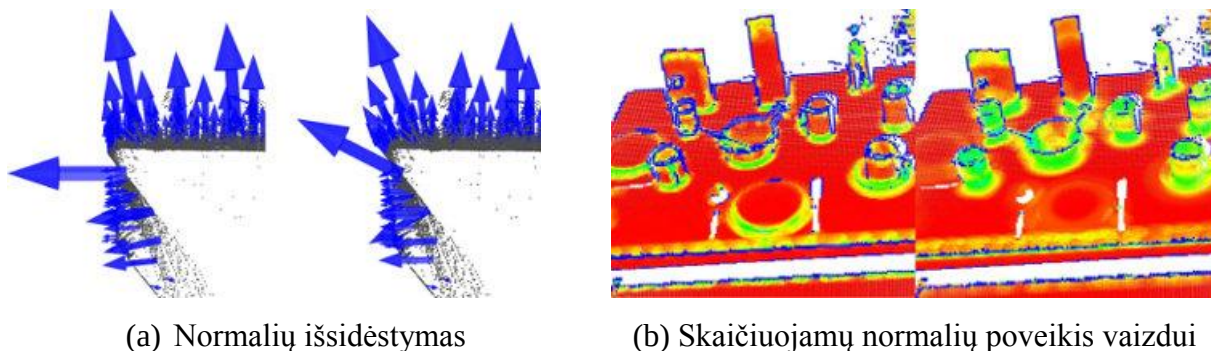
4 pav. Nesisteminga paviršiaus normalių orientacija [4]

4 paveiksle pateiktas pavyzdys, kuriame matyti nevienareikšmiškos normalių orientacijos ištisai per visą taškų debesies duomenų rinkinį (4 pav.). Jei normalių orientaciją nėra sistemingai orientuota, suprantama kad ne bus galima tinkamai įvertinti visų paviršiaus taškų. Taigi, išmėtytas keliomis kryptimis normales, reikia nukreipti stebėtojo (jutiklis) link. Šios problemos sprendimas yra nesudėtingas jei yra žinomas stebėjimo taškas. Žemiau pateikiamas rezultatas (5 pav.) po visų duomenų rinkinyje turimų normalių ištiso jų orientavimo link stebėjimo taško.



5 pav. Sisteminga paviršiaus normalių orientacija [4]

Atkreipiamas dėmesys į tai, kad reikia pasirinkti tinkamą įvertinamų aplinkinių taškų skaičių – mastelį. PCL bibliotekoje yra įgyvendinti du metodai mastelio pasirinkimui. Pirmasis metodas paremtas tuo, kad pasirenkamas artimiausių kaimynų skaičius k , kuris bus naudojamas normalių skaičiavimui. Antruoju nurodomas apskritimo spindulys r , kuriame visi esantys artimiausi taškai bus panaudoti normalių skaičiavimuose. Tinkamo mastelio pasirinkimas sukelia sunkumų sistemoms, kurios turi automatiškai vertinti požymių taškus. 6 paveiksle pateikiami rezultatai, kai pasirenkamas tinkamas ir netinkamas kaimyninių taškų paieškos mastelis [4].



6 pav. Pasirinkto normalių skaičiavimo mastelio rezultatas [4]

Kairėje paveikslo pusėje (6 pav., a) matyti teisingai ir neteisingai parinktus mastelius. Kai mastelis yra teisingai pasirinktas, apskaičiuotos normalės yra apytiksliai statmenos dviems plokštumos paviršiams ir smulkūs kampai gerai matomi (6 pav., b). Tačiau, jeigu pasirinktas mastelis bus per didelis, tai tokiu būdu vertinamų kaimyninių taškų rinkinys bus taip pat per didelis. Jei mastelis per didelis, tai normalių vertinimui paimami gretimos plokštumos taškai – įvyksta savotiškas geometrinių plokštumų persidengimas. Kai plokštumos bei jų taškai persidengia, gaunamas požymių iškraipymas. Dviejų plokštumų paviršių kampuose gaunamos pasuktos plokštumos normalės ko pasekoje gaunami nutrinti kampai ir pašalintos smulkios detalės. Taigi, priklausomai nuo to kokios ir kokio dydžio objekto detalės reikalingos tyrimams, yra pasirenkamas normalių skaičiavimo mastelio dydis bei mastelio įvertinimo metodas.

2.3. Objekto išskyrimas

Antro skyriaus pradžioje pateiktame paveiksle (1 pav.) pirmasis objekto atpažinimo etapas yra taškų debesies gavimas, šiame etape atliekamas objekto 3D taškų išrinkimas iš bendro 3D taškų debesies. Taigi, taškų debesis segmentuojamas o PCL bibliotekoje įgyvendinti segmentavimo metodai atlieka procedūras, kurių metu debesies taškai padalinami į atskirus klasterius. Po segmentavimo procedūros galima dirbti su atskirais taškų debesimis. Minėtoje PCL bibliotekoje yra keletas metodų skirtų segmentavimui. Jų sąrašas ir trumpas aprašymas pateikiami 2.1 lentelėje.

2.1 lentelė. Pagrindiniai segmentavimo metodai

Metodas	Požymiai ir charakteristikos
„ConditionalEuclideanClustering“	Skaičiuojamas Euklido atstumas. Naudotojai nustato klasterizavimo sąlygas: taškų kiekį klasteryje, atstumą tarp artimiausių kaimyninių taškų [6]
„ExtractPolygonalPrismData“	Naudojami taškai, kurie apibrėžia plokštumą, kartu su vartotojo nustatytu aukščiu. Sudaroma 3D daugiakampė prizmė, kurioje visi esantys taškai segmentuojami. Metodas taip pat tinkamas naudoti kai objektas kertasis su plokštuma [6]
„GrabCut“	Pagrindinis metodo elementas yra Gauso modelis GMM (angl. <i>Gaussian Mixture Model</i>) paremtas spalvos informacija. Segmentavimas atliekamas pagal vaizdo energijos mažėjimą [7, 8, 6]
„SACSegmentation“, „SACSegmentationFromNormals“	Atliekamas bazinių geometrinių formų priderinimas, taigi segmentavimas įgyvendinamas parametriniais modeliais (plokštuma, sfera, cilindras ir kiti). Gali būti panaudotos ir normalės [6]
„SeededHueSegmentation“	Leistinas klasterio nuokrypis nuo kito klasterio, Euklido erdvėje, paskaičiuojamas naudojant pilkumo, arba spalvinės skalės savybes. Kol kas naudojamos tik spalvos savybės [6]
„SegmentDifferences“	Naudojami ištiesinti taškų debesys. Metodas išmatuoja atstumą tarp dviejų ištiesintų debesų ir gražina skirtumą. Užsibrėžtą atstumą – slenkstį nustato pats vartotojas [6]

2.1 lentelėje pateikti segmentavimo metodai daugiausia yra tinkami dvimačiams taškų debesims segmentuoti. Šiuo metu vyksta šių metodų tobulinimai, mėginama juos sėkmingai pritaikyti trimačiams taškų debesims. Objektams atpažinti ilgai buvo taikomas dvimatis objekto modelis, todėl daugelis metodų dar tik pritaikomi pakankamai naujai 3D taškų apdorojimo technologijai. Dalis metodų matuoja atstumus tarp taškų, kiti naudoja paviršiaus spalvos parametrus, tačiau efektyviausi ir universaliausi metodai mėgina pritaikyti bazinių formų parametrus bei normalėmis įvertinti paviršiaus geometrinius parametrus. Pastebėta, kad metodai nėra autonominiai ir tobuli, kadangi daugelyje metodų jų parametrus nustatyti, ir su laiku pakeisti, reikia vartotojo dalyvavimo. Kiekvienas metodas dažniausiai taikomas pagal situaciją – keliamus reikalavimus ir rezultatus, tačiau taip pat gali būti naudojami ir šių segmentavimo metodų deriniai.

2.4. Būdingųjų taškų išskyrimas

Ne visi gautieji objekto taškų debesies, taškai yra svarbūs ir suteikia svarbią informaciją. Šiuo atveju yra galimybė panaudoti būdingųjų taškų išskyrimo algoritmus, kurių dėka išskiriami svarbesni taškai, sumažinamas naudojamų taškų kiekis ir tuo pačių apdorojamų duomenų kiekis. Galima atlikti tolimesnius skaičiavimus ir neatliekant būdingųjų taškų išskyrimo, tačiau iš vienos pusės tokiu būdu, duomenis apdorojančioms sistemoms, iškeliami dideli techniniai reikalavimai.

2.2 lentelė. Būdingųjų taškų išskyrimo metodai

Metodas	Požymiai ir charakteristikos
„HarrisKeypoint3D“	3D detektoriuje, būdingųjų taškų išskyrimo, naudojama ta pati idėja kaip ir 2D Harris (pagal vaizdo gradientą), tačiau vietoje gradiento panaudojamos paviršiaus normalės. Naudoja tik geometrijos parametrus [9, 10, 11]
„HarrisKeypoint6D“	Detektorius skirtas kampų aptikimui 3D (pagal X, Y, Z), 2D (pagal intensyvumą), taip pat galimas ir mišrus paminėtų veiksmų taikymas. Universalus pritaikymas. Ilgiausiai skaičiuojamas [9, 11]
„ISSKeypoint3D“	Detektorius nustato būdingus taškus pagal 3D vidinę formos signatūrą, vertinami 2D, 3D vaizdų parametrai, naudojami lokalūs bei pusiau lokalūs taškų debesies regionai, skirtingi stebėjimo taškai, įvertinamos normalės. Savo sparta nusileidžia „NarfKeypoint“, tačiau lenkia „HarrisKeypoint3D“ [9, 11, 12]
„NarfKeypoint“	Įvertinamos priekinio ir galinio plano taškų ribos – gaunamas vaizdo rėmelis iš kurio išskiriami būdingieji taškai. Įvertinamos bei surikiuojamos normalės, ieškoma labiausiai besiskiriančių taškų. Gali būti naudojamas kaip deskriptorius. Vienas iš sparčiausiai vykdomų [9, 11, 13]
„SIFTKeypoint“	Būdingiems taškams nustatyti naudojamas artimiausių taškų intensyvumo pokytis, Gauso-Laplaso filtras. Metodas tikslus ir stabilus, atsparus pasukimams bei mastelio keitimui, tačiau nėra įvertinama paviršiaus deformacija. Gali būti panaudojamas kaip deskriptorius [9, 11, 14]
„SUSANKeypoint“	Vertinami vaizdo taškų intensyvumo skirtumai, kurie sujungiami su normalių krypčių skirtumais. Metodas aptinka kraštus ir kampus, paremtas netiesiniu kraštų filtravimu. Skirtingai nuo „HarrisKeypoint3D“ tiesiogiai pasinaudoja normalių kryptimis taigi yra spartesnis [11]
„UniformSampling“	Metodas iš taškų debesies surenka nustatyto dydžio lokalų 3D tinklėlį ir taip atlieka duomenų retinimą bei filtravimą. Naudojamas 3D vokselinis tinklėlis, kiekvienas vokselis aproksimuojamas apie vekselio centrą. Metodas tiksliau apibrėžia esminį paviršių. Sparčiausias [9, 11, 15]

2.2 lentelėje pateikti metodai, kurie įgyvendinti PCL bibliotekoje ir sėkmingai gali būti taikomi būdingųjų taškų išskyrimui iš milžiniško 3D taškų debesies. Šie metodai mėgina taškų debesyse surasti tam tikrus požymius ir pagal juos atlikti, taškų išskyrimą. Visų aptartų metodų pagrindinis tikslas yra išskirti būdinguosius taškus, kuriuose vėliau sekančiuose žingsniuose bus skaičiuojami 3D deskriptoriai. Ne taip seniai 3D objektų atpažinimui pradėta naudoti spalvos informaciją todėl pastebėta, kad dauguma metodų yra panašūs – naudoja geometrinius paviršiaus parametrus. Iš visų metodų išskirti kurį nors vieną efektyviausią metodą būtų per sudėtinga, kadangi kiekvieno iš jų algoritmas nėra prastesnis už kitus. Sunku nustatyti, kokį poveikį objektų atpažinimui turi vienas ar kitas naudojamas, būdingųjų taškų išskyrimo metodas, kadangi objektų atpažinime vienas iš svarbiausių objektų atpažinimo metodo dalių yra požymių – deskriptorių skaičiavimas.

2.5. Požymių deskriptoriai

Deskriptorius kaip jau buvo minėta, tai yra tam tikros savybės ar bruožo apibūdinimas ir jis apibrėžia savybių apibūdinimo sintaksę ir semantiką. Ir atvirkščiai, galima keisti tam tikus objekto ar taško savybės parametrus, pakeitus deskriptoriaus reikšmę. Šiame skyriuje bus apžvelgti PCL bibliotekoje įgyvendinti 3D deskriptoriai, kurie įvertinta 3D objektų paviršiaus požymius.

2.3 lentelė. Deskriptorių pagrindiniai parametrai

Deskriptorius	Normalės	Histogramos ilgis	Lygiagretinimas
PFH	Yra	125	GPU
PFHRGB	Yra	250	GPU
FPFH	Yra	33	OpenMP ir GPU
VFH	Yra	308	GPU
CVFH	Yra	308	Nėra
3DSC	Yra	Kintamas	Nėra
USC	Nėra	Kintamas	Nėra
RIFT	Nėra	32	Nėra
PPF	Yra	5	OpenMP ir GPU
ESF	Nėra	640	Nėra
PCE	Yra	5	GPU
SHOT	Yra	352 + (9 LRF)	OpenMP
COLORSHOT	Yra	1344 + (9 LRF)	OpenMP

2.3 lentelėje pateikiamas sąrašas deskriptorių ir jų pagrindiniai parametrai. Sąraše pateikiami ne visi PCL 1.6 versijoje įgyvendinti deskriptoriai, kadangi kai kurie iš jų reikalauja papildomo duomenų apdorojimo bei papildomų algoritmų taikymo, taigi aptarsime tuos deskriptorius kurie gali būti tiesiogiai pritaikomi objektų atpažinime. Pateikiamų metodų skaičiavimus galima paspartinti naudojant *OpenMP* programavimo standartą, kuriuo kompiuteriuose įgyvendinami lygiagrečiai algoritmai arba naudojant grafinės plokštės GPU (angl. *Graphics Processing Unit*) skaičiavimo pajėgumus. Deskriptorių parametrai bus aptarti tokie kokie yra pagal nutylėjimą nustatyti PCL bibliotekoje.

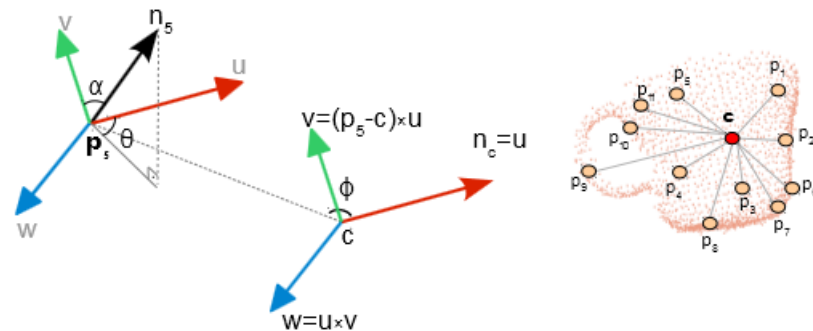
PFH (angl. *Point Feature Histograms*) deskriptorius įvertina paviršiaus kreivumą ir normales. Paviršiaus kreivumas ir normalės įvertinamos naudojant artimiausius aplinkinius kaimyninius taškus. Kaimyniniuose taškuose paskaičiuojamos normalės. Tolimesni skaičiavimai atliekami su kiekviena taškų pora panaudojant Darboux atskaitos sistemą [16], kurioje skaičiuojami trys vektoriai ($\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$). Taškų normalės naudojamos minėtiems lokalios sistemos vektoriams apskaičiuoti. Panaudojant $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$ vektorius paskaičiuojami kampai tarp taškų normalių, tada gaunami geometrijos požymiai α , ϕ , θ , d . Požymis d ne visada naudojamas, kadangi jis nurodo atstumą tarp taškų, jį pakeičia stebėjimo taškas. Šie požymiai naudojami deskriptoriaus histogramai sudaryti. Histogramos sudarymui kiekvienas požymis yra padalinamas į vadinamus „binus“ (angl. *bins*) talpyklas-skyrius, kurių skaičių gali pasirinkti pats naudotojas. Pagal nutylėjimą deskriptoriaus histogramą sudaro 125 skyriai [17].

PFHRGB (angl. *Point Feature Histograms + RGB*) deskriptorius sudarytas PFH pagrindu. Tai vienas iš dviejų PCL bibliotekoje įgyvendintų deskriptorių, kuris įvertina ne tik formos bet ir spalvos požymius. Prie pradinės deskriptoriaus formos histogramos pridėta tokio pačio ilgio (125 skyrių) histograma, kadangi naudojama trijų sumaišomų spalvų sistemą – RGB. Taigi, PFHRGB deskriptoriaus histograma dvigubai ilgesnė nei PFH ir susideda iš 250 skyrių (2.3 lentelė).

FPFH (angl. *Fast Point Feature Histograms*) tai taip pat PFH pagrindu sudarytas deskriptorius, kurio pagrindinis tikslas sumažinti PFH deskriptoriaus skaičiavimo sudėtingumą, bei sumažinti perteklinę informacijos kiekį [18]. Sudaryto deskriptoriaus histograma kelis kartus trumpesnė nei jo pirmtako. Histograma skaičiuojama ne visoms taškų poroms išarto, kaip PFH, o tik pasirinktam taškui ir jo kaimyniniams taškams. Gaunama supaprastinta taško požymio histograma SPFH (angl. *Simplified Point Feature Histogram*), kuri naudojama kaip svorio koeficientas kitų taškų porų histogramoms skaičiuoti. Nors metodo sudaroma histograma trumpesnė (2.3 lentelė), požymių įvertinimo kokybė, diskriminantinė galia, atsparumas triukšmams išlieka toks pats kaip PFH [17].

VFH (angl. *Viewpoint Feature Histogram*) deskriptorius užšifruoja geometrijos ir stebėjimo taško parametrus (7 pav.). Į FPFH deskriptorių pridedamas stebėjimo taškas, kuris suteikia deskriptoriui papildomų laisvės laipsnių 6DoF (angl. *Six Degrees of Freedom*). Papildomi laisvės

laipsniai reikalingi tam, kad galima būtų atpažinti objektus ir jų poziciją. Deskriptorius apskaičiuojamas keliais žingsniais. Pirmiausia surandamas paviršiaus masės centras c ir jo normalė n_c . Po to kiekvienam kaimyniniam taškui skaičiuojamos normalės ir taikoma lokali atskaitos sistema, toliau LRF (angl. *Local Reference Frame*) [19, 20], su parametrais (u, v, w) . Tada nustatomi kampiniai normalių parametrai α, ϕ, θ . Šiame deskriptoriuje pagal nutylėjimą naudojamas atstumas tarp taškų ir masės centro – d . Apskaičiuojamas papildomas parametras β , tai kampas tarp kiekvieno paviršiaus taško normalės ir stebėjimo taško.



7 pav. VFH deskriptorius [21]

Gauti parametrai, kurie aprašo paviršiaus ir stebėjimo taško požymius, padalinami į skyrius (2.3 lentelė) iš kurių sudaroma deskriptoriaus histograma. Tai yra globalus deskriptorius, kuris skaičiuojamas visam paviršiui – taškų debesiui o ne kiekvienam taškui [21].

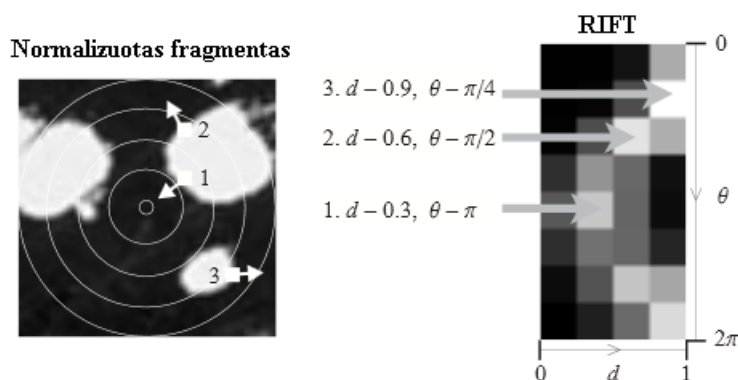
Kitas deskriptorius CVFH (angl. *Clustered Viewpoint Feature Histogram*) tai patobulintas VFH požymių skaičiavimo metodas taikomas XYZ taškų debesims su normalėmis [22]. Taigi, deskriptoriuje apskaičiuojami parametrai $(\alpha, \phi, \theta, \beta)$. Paskaičiuojama formos skirstinio dedamoji SDC (angl. *Shape distribution Component*) [23]. Kadangi VFH deskriptoriuje taikyta LRF yra jautri paviršiaus nelygumams ir paviršiaus skylėms (nėra duomenų), tai CVFH atliekamas paviršiaus glotninimas – didelių kreivumų pašalinimas.

Deskriptorius 3DSC (angl. *3D shape context*) tai 2D formos konteksto pritaikymas trimačiai aplinkai [24]. Deskriptorius skaičiuojamas naudojant aplinkinius taškus artimoje kaimynystėje ir kiekvienam taškui naudoja sferinį tinklėlį (10 pav., a). Tinklėlis susideda iš kelių skyrių, kurias sudaro tinklelio sferos spindulio, azimuto ir aukščio matmenys. Šie matmenų dydžiai gaunami, kai sferinio tinklelio šiaurinis polius nustatomas paviršiaus normalės kryptimi. Naudotojas gali pats pasirinkti, kiek skyrių sudarys kiekvienas iš matmenų dėl to šio deskriptoriaus histogramos dydis yra kintamas (2.3 lentelė). Histogramos dydis visiems naudotiems taškams yra vienodas. Galutinė 3DSC deskriptoriaus histograma gaunama kai visi, naudotų taškų, histogramų skyriai atitinkamai sudedami. Svarbu paminėti, kad paviršiaus liestinė nustatoma atsitiktiniu algoritmu, todėl visus, įvairiomis azimuto kryptimis gautus taškų deskriptorius, reikia papildomai patikrinti, išrinkti

geriausius ir tik po to pritaikyti. Visi kaimyniniai taškai turi būti skaičiuojamieji – baigtinio dydžio, kitaip deskriptorius pateiks NaN (angl. *Not a Number*) reikšmes [22].

USC (angl. *Unique Shape Context*) tai patobulintas, aptarto 3DSC deskriptoriaus, variantas. Jau minėta, kad 3DSC deskriptorius skaičiuojamas įvertinant daugelį, skirtingomis azimuto kryptimis gautų, deskriptorius. Patobulintame metode nebereikia skaičiuoti daugkartinio deskriptoriaus pasukimo skirtingomis azimuto kryptimis, kadangi naudojama unikali, nedviprasmiška lokali atskaitos sistema LRF, kuri pateikia ne tik kartotines normalių ašis bet ir unikalias paviršiaus liestinių kryptų poras. Skaičiuojant, sudaroma kovariacinė svorių matrica $\mathbf{M}$, joje įvertinami taškai kurie yra prie skaičiuojamo taško sferinėje (9 pav.) kaimynystėje. Po to skaičiuojama kovariacinės matricos mažiausių kvadratų suma. Suskaičiuojama kovariacinės matricos eigen vektorių dekompozicija EVD (angl. *EigenVector Decomposition*) ir gaunami trys LRF vektoriai. Eigen vektorius su mažiausia reikšme nurodo normalės kryptį. Kiti deskriptoriaus skaičiavimo etapai tokie patys kaip ir 3DSC [25].

RIFT (angl. *Rotation Invariant Feature Transform*) tai patobulintas SIFT [26] deskriptoriaus variantas, kurį sudaro debesie XYZ taškai ir intensyvumas [22]. Pirmiausia kiekviename taške sudaromas normalizuotas paviršiaus fragmentas (8 pav.).



8 pav. RIFT deskriptoriaus histogramos skaičiavimas [27]

Apskritimais normalizuotas fragmentas padalinamas į keturis koncentrinus žiedus, kurių plotis yra lygus. Orientacijos gradiento histograma skaičiuojama kiekviename žiede padalinant kiekvieną žiedą į 8 skyrius. Histogramos orientacija skaičiuojama kiekviename taške atitinkamai pagal nurodomą išorės kryptį iš centro [27].

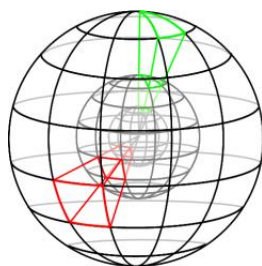
PPF (angl. *Point Pair Feature*) deskriptorius kombinuoja intensyvumo ir gylis duomenis (normalės) [22]. Šiuo metodu siekiama aprašyti objektų siluetus ir paviršiaus išvaizdą. Pirmiausia taškų porai p_1 ir p_2 skaičiuojami požymiai $F(p_1, p_2)$, kurie užšifruojami trimis kampais tarp normalių ir atstumu d tarp dviejų skaičiuojamų taškų: $\angle(n_1, d)$, $\angle(n_2, d)$, $\angle(n_1, n_2)$. Po to surandami visų taškų porų požymiai. Sudaromas daugiamatis modelis M . Tarkime M bus įkeltos scenos S objekto modelis. Globalus modelio deskriptorius gaunamas iš atskirų modelio erdvės elementų.

Tegu pasirinktas taškas $s \in S$ yra objekto modelyje M . Taigi objekto modelyje yra taškas $m \in M$, kuris atitinka scenos tašką s . Iš to seka, jei šių taškų normalės sulygintos, tai galima sulyginti objekto modelį ir sceną sukanč modelį kampu α aplink sulygintą normalę. Parametrų pora (m, α) vadinama lokaliomis koordinatėmis. Tam, kad surasti lokalias koordinatas ieškoma didžiausio skaičiaus scenos taškų, kurie priklauso objekto modeliui. Tai atliekama naudojant balsavimo schemą [28]. Lokalių koordinatų pagalba galima nustatyti objekto pozą.

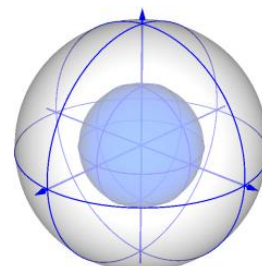
ESF (angl. *Ensemble of Shape Functions*) tai globalus deskriptorius, kuriame naudojamos tris formos skirstiniai – funkcijos: D2, D3, A3 [22]. D2 – skaičiuojamas atstumas tarp dviejų atsitiktinai pasirinktų paviršiaus taškų. D3 – skaičiuojama kvadratinė šaknis trikampės erdvės, kurią sudaro trys atsitiktinai pasirinkti paviršiaus taškai. A3 – skaičiuojamas kampas tarp trijų, atsitiktinai pasirinktų, 3D modelio paviršiaus taškų [29]. Kiekviena iš verčių dar padalinama į tris klases, kurios nurodo kur yra išsidėsčiusios linijos, kurios jungia atsitiktinai pasirinktus plokštumos taškus. Taigi, deskriptoriaus dedamąsias histogramas sudaro trys jungiamųjų linijų klasės: „ant paviršiaus“, „ne ant paviršiaus“ ir mišrus variantas. Papildomai paskaičiuojama dar viena histograma D2 funkcijos atstumams tarp linijų „ant paviršiaus“ ir „ne ant paviršiaus“ nustatyt. Prieš atstumų skaičiavimą, gautas taškų debesis aproksimuojamas vokseliniu tinkleliu. Minėtos, kiekvienos funkcijos klasės, grupuojamos, pagal L_1 metriką nustatomas atstumas PDF (angl. *Probability Density Function*) ir gaunama deskriptoriaus histograma [30].

PCE (angl. *Principal Curvatures Estimation*) deskriptorius apskaičiuoja pagrindinio paviršiaus kreivumo, kuris apibrėžiamas saugomais taškais ir normalėmis, kryptis (eigen vektoriai) ir reikšmes (eigen reikšmės) [22]. Tam kad gauti minėtas kryptis ir reikšmes, atliekama pagrindinių komponentų analizė PCA. Deskriptoriaus histograma susideda iš 5 skyrių. Pirmi trys skyriai užšifruoja pagrindinio kreivumo požymius (eigen vektoriai), kiti du saugo mažiausią ir didžiausią eigen reikšmes.

SHOT (angl. *Signature of Histograms of Orientations*) deskriptorius taikomas taškų debesims su normalėmis [22]. SHOT priklauso nuo kartotinės lokals atskaitos sistemos LRF, kuri paremta paviršiaus sklaidos matricos, eigen reikšmių dekompozicija (EVD) aplink skaičiuojamą tašką.



(a) 3DSC deskriptoriuje [24]



(b) SHOT deskriptoriuje [31]

9 pav. Deskriptoriuose naudojami 3D Sferinio tinklelio pavyzdžiai

Gautos eigen reikšmės apima eile kaimyninių taškų. 3D sferinis tinklelis uždedamas ant LRF sistemos skaičiuojamame taške. Lokali sistema geometriniais požymiais skaičiuoti naudoja taškų normales. Tinklelio kaimyninė erdvė padalinama tam, kad kiekviename tinklelio skyriuje būtų gautas taško histogramos svoris. Galutinėje deskriptoriaus histogramoje visos kaimyniniam taškams paskaičiuotos histogramos sujungiamos į vieną histogramą [31]. LRF apibrėžti panaudojamos 9 reikšmės o formos požymiams apibrėžti reikia 11 reikšmių, kurios padalinamos į 32 skyrius tam, kad būtų gauta 352 skyrių deskriptoriaus histograma (2.3 lentelė).

CSHOT arba COLORSHOT (angl. *Color Signature of Histograms of Orientations*) deskriptorius sudarytas SHOT deskriptoriaus pagrindu. Tai antras PCL bibliotekoje įgyvendintas deskriptorius, kuris įvertina ne tik formos bet ir spalvos požymius. Nuo savo pirmtako šis deskriptorius skiriasi tik papildoma histogramos dalimi, kuri skirta spalvos požymiams užšifruoti. Spalvos šifravimui naudojamos 31 reikšmės, kurios taip pat dalinamos į 32 skyrius. Taigi spalvos histograma kelis kartus ilgesnė nei formos požymių histograma [32]. Be įprastos RGB spalvų sistemos skaičiavimuose galima panaudoti ir kitas spalvų sistemas. Geriausi deskriptoriaus efektyvumo parametrai nustatyti taikant *CIElab* spalvų sistemą ir skaičiuojant L_1 atstumą (normą).

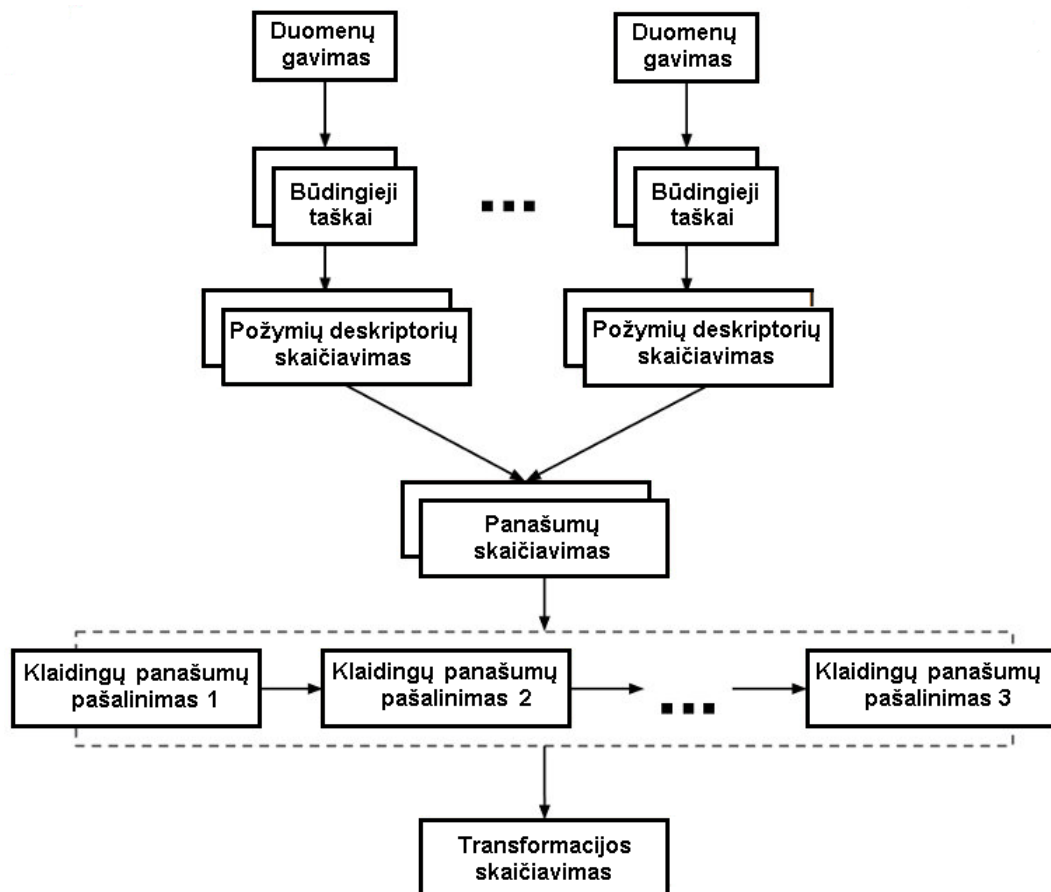
Taigi, 2.3 lentelėje pateikiami pagrindiniai deskriptorių parametrai – kokio ilgio deskriptoriaus histograma, ar yra galimybė atlikti lygiagrečius skaičiavimus, ar skaičiavimuose naudojamos normalės. Pateiktą sąrašą sudaro didžioji dalis deskriptorių, kurių skaičiavimus galima paspartinti. Visi deskriptoriai įvertina paviršiaus parametrus – formos požymius. Dauguma 3D deskriptorių, paviršiaus parametrams nustatyti, naudoja normales. Tik du deskriptoriai naudoja ir formos ir spalvos požymius. Aptarti metodai yra viena iš objektų atpažinimo metodo dalių. Kaip jau minėta ši dalis yra viena svarbiausių objektų atpažinime. Kitame žingsnyje siekiama palyginti jau aptartų metodų – 3D deskriptorių reikšmes (požymių palyginimas).

2.6. Požymių palyginimas

Požymių palyginimas tai beveik paskutinis objekto atpažinimo žingsnis. Šiame žingsnyje atliekamas 3D objekto požymių palyginimas su visais 3D objektų požymiais, esančiais duomenų bazėje bei išrenkamas labiausiai tinkantis objektas – atpažintas objektas. Tam kad atpažinti objektą reikia nustatyti skirtumus, panašumus ir atitikimus tarp objektų požymių vektorių.

Prieš tai aptartiems objektų atpažinimo metodų žingsniams, PCL bibliotekos 1.6 versijoje, yra numatyti ir įgyvendinti metodai, tačiau paskutiniojo atpažinimo žingsnio PCL bibliotekoje, bendrai priimto, nėra. Šio žingsnio įgyvendinimui skirti metodai vis dar tobulinami bibliotekos kūrėjų. 3D Objektų atpažinimo procesas yra labai panašus į 3D aplinkos duomenų registravimą, todėl objektų atpažinime sprendžiamos labai panašios problemos kaip ir aplinkos registravime. Registruojant

aplinkos duomenis dažnai naudojami keli jutikliai – skenavimo įrenginiai bei skirtingi skenavimo kampai o gauti duomenys – taškų debesys sujungiami į vieną taškų debesį (aplinkos modelis). Tam kad sujungti taškų debesis, iš pradžių atliekamas būdingųjų taškų palyginimas, kurio metu ieškoma taškų atitikimų. Labiausiai atitinkantys taškai parodo panašumą tarp taškų debesų. Taigi ši procesą galima taikyti objekto atpažinimo metodui sudaryti. 10 paveiksle pateikiami aplinkos duomenų registravimo žingsniai.



10 pav. Aplinkos duomenų registravimo schema [33]

Kadangi PCL bibliotekos atpažinimo metodai vis dar plėtojami, bus laisvai pasirinkti požymių palyginimo – objekto atpažinimo metodai bei jų parametrai. Pasirinkta sekanti veiksmų seka:

- atitinkančių – panašių taškų išrinkimas
- klaidingų atitikimų pašalinimas
- atstumų tarp panašių taškų požymių skaičiavimas ir tinkamiausio objekto pasirinkimas

Atitinkančiais taškais laikomi tie taškai, kurie būdami skirtinguose objektų paviršiuose yra panašūs arba vienodi. Šie atitikimai gali būti skaičiuojami pagal taškų koordinates (x, y, z) ir pagal taškų suskaičiuotus požymius (darbe naudojami taškų požymiai):

- „grubios jėgos“ (angl. *Brute force*) skaičiavimo metodas
- kd-tree artimiausių kaimynų paieška

Požymių palyginimui taikant „grubios jėgos“ požymių skaičiavimo metodą, nėra naudojamas koks nors specialus klasifikatorius, tiesiog reikia įgyvendinti funkciją į kurios įėjimą paduodami du rinkiniai požymių deskriptorių A ir B. Po to iš vieno požymių rinkinio A paimamas vienas požymis ir palyginimas su visais kitais požymiais kitame rinkinyje B. Požymių palyginimui naudojamas kvadratinis atstumas arba kitaip dar – Euklido atstumas. Jau minėta, kad „grubios jėgos“ metodas skaičiuoja atitikimus tarp visų taškų debesyje A ir taškų debesyje B. Taigi, kadangi nėra panaudojama jokių papildomų duomenų apdorojimų atitinkantiems taškams surasti, labai didelė tikimybė, kad bus gauta daug klaidingų atitikimų.

Kd-tree metodas atlieka erdvės dalinimą – dekompoziciją ir skaičiuoja Euklido atstumus tarp taškų arba taškų požymių tik tam tikrose daugiamatės erdvės ribose (kadangi erdvė dalinama). Taip pat svarbu paminėti, kad atitikimų paieškos metodai pagal nutylėjimą turi tam tikras nustatytas savybes. Kiekvienam, B taškų debesies, taškui ieškoma atitikimų su kiekvienu, A taškų debesies, tašku. Panaudojami tik tie atitikimai, kurie turi abipusį atitikimų susikirtimą: atitikimai iš B debesies į A debesį ir atitikimai iš A į B [33].

Praktika rodo, kad ne visi požymių atitikimai būna teisingi. Klaidingi atitikimai gali neigiamai įtakoti rezultatus, todėl juos reikia šalinti. PCL bibliotekoje įgyvendinti keli metodai, kurie gali pašalinti netinkamus atitikimus. Tai galima atlikti pritaikius RANSAC metodą arba gali būti procentais mažinamas surastų atitikimų kiekis. RANSAC (angl. *RANdom SAMple Consensus*) metodas nustato matematinį modelį ir jo parametrus remiantis atsitiktiniu duomenų išrinkimu. Šiuo atveju matematinis modelis tai objekto taškų debesis A o skaičiuojami parametrai tai kito objekto taškų debesis B. Metodas atskiria du taškų tipus – patenkantys (angl. *inlier*) į modelį ir nepatenkantys (angl. *outlier*) į modelį. Metodas pasižymi atsparumu triukšmams [34]. Procentais mažinti naudojamų atitikimų skaičių nėra tikslinga, kadangi tokiu būdu, nėra tiksliai žinoma kokie požymių atitikimai yra klaidingi. Taigi, yra didelė tikimybė, kad bus pašalinti teisingi požymių atitikimai arba iš kitos pusės, gali būti per mažai pašalinta klaidingų atitikimų.

10 paveiksle pateiktas transformacijos skaičiavimas, jis aplinkos registravimo procese reikalingas tam, kad galima būtų nustatyti objekto atitinkančių taškų padėtį arba kitaip – objekto pozą. Tačiau, kadangi šiame darbe nagrinėjami atpažinimo metodai, kurie pagrįsti spalvos ir formos požymių rinkiniais, tai objekto pozos atpažinimo nėra siekiama ir šiuo atveju paskutinis atpažinimo etapas bus atstumų tarp atrinktų (teisingų) atitikimų įvertinimas – sumavimas.

Taigi, minėtas taškų požymių panašumų palyginimas atliekamas kiekvienam objektui esančiam duomenų bazėje. Suminiai atstumai įvertinami tarp kiekvieno taškų debesies ir geriausias rezultatas (mažiausias atstumas) nurodo į daugiausiai panašumų turintį taškų debesį – objektą. Šio objekto pavadinimas priskiriamas daugiausiai panašumų turinčiam objektui – objektas atpažintas. Taip pat galima paminėti, kad objektų atpažinimo metodo sudėtinės dalys – atskirų dalių metodai gali būti

labai įvairūs. Įmanoma sugeneruoti įvairių atpažinimo metodų ir išbandyti juos su įvairiais atskirų dalių parametrais. Šiame darbe tai pat bus pasirinktas jau aptartas būdas – atpažinimo metodų surinkimas iš atskirų metodų dalių. Iš aptartų atpažinimui skirtų metodų bus sudaryti keli objektų atpažinimo metodai.

2.7. Trumpas skyriaus apibendrinimas

Šiame skyriuje buvo aptarta 3D objektų atpažinimo technologija, kuri paremta trimačių taškų debesimis. Apžvelgta preliminarinė objekto atpažinimo veiksmų grandinė, kurioje aptarti pagrindiniai 3D objekto atpažinimo žingsniai. Išsiaiškinta kaip 3D erdvėje reprezentuojamas trimatis taškas. Aptarta kodėl būtina keisti Dekarto koordinačių sistemos sąvoką į kitą koncepciją – lokalųjį deskriptorių. Analitinėje apžvalgoje buvo aptartos atskiros, 3D objektų atpažinimo metodų, sudarančios dalys.

Išsiaiškinti pagrindiniai paviršiaus parametrai, bei šių parametų įvertinimo būdai. Pateiktas paviršiaus požymių palyginimo problemos sprendimas. Nustatyta normalių reikšmė ir jų pritaikymas. Išanalizuota kaip skaičiuojami paviršiaus taškų požymiai – požymių vektoriai, bei paviršių palyginimo galimybės. Apibrėžtos „gero“ požymio savybės bei charakteristikos.

Pagal pateiktą, objekto atpažinimo, veiksmų grandinę (1 pav.) apžvelgti kiekvienos grandinės dalies metodai bei jų pagrindiniai parametrai. Aptarti objektų segmentavimo būdai. Nustatyta, kad dauguma 3D metodų kilę iš dvimačių vaizdų apdorojimo metodų. Šiuos metodus vis dar bandoma pritaikyti trimačiams taškų debesims apdoroti.

Trumpai apžvelgti būdingųjų taškų išrinkimo metodai. Nustatyta, kad dauguma metodų yra panašūs ir naudoja geometrinius paviršiaus parametrus. Kai kurie metodai pasižymi filtravimo savybėmis, todėl gali būti taikomi kaip filtravimo metodai.

Nustatyta viena iš svarbiausių, 3D objektų atpažinimo metodo, dalių – 3D požymių deskriptorių skaičiavimas. Dauguma PCL bibliotekoje įgyvendintų 3D deskriptorių skaičiavimo metodų įvertina tik paviršiaus geometrijos – formos požymius. Iš aptartų 13-os deskriptorių, tik du iš jų įvertina spalvos ir formos paviršiaus požymius. Dauguma deskriptorių, paviršiaus parametrus vertinti, naudoja normales.

Objektų atpažinimo metodo paskutinis žingsnis – objektų paviršių požymių palyginimas ir panašiausio pavyzdžio išrinkimas. Požymių palyginimo metodai daugiamačiuose požymių erdvėje skaičiuoja atstumus tarp taškų požymių – ieškoma požymių atitikimų. Mažiausius atstumus turintys atitinkantys taškai laikomi teisingais atitikimais. Klaidingų atitikimų pašalinimui numatyti tam skirti metodai. Šiame skyriuje susipažinta su, PCL 1.6 versijos, bibliotekoje įgyvendintais 3D taškų debesų apdorojimo metodais.

3. PERSPEKTYVIŲ OBJEKTŲ ATPAŽINIMO METODŲ ANALIZĖ

Vienas iš svarbiausių objektų atpažinimo metodų žingsnių, kaip jau buvo minėta 2 skyriuje, yra 3D objekto požymių įvertinimas – 3D deskriptorių skaičiavimas, todėl sudarant objektų atpažinimo metodus, daugiausia bus remiamasi deskriptoriais. Šiame skyriuje bus pamėginta apžvelgti kelis perspektyvius požymių įvertinimo metodus PFH, PFHRGB, FPFH, SHOT, COLORSHOT. Taip pat bus aptarti minėtų, PCL bibliotekoje įgyvendintų metodų, parametrai nuo kurių priklauso objektų atpažinimo kokybė. Šiame skyriuje taip pat bus išanalizuoti, pasirinktieji būdingųjų taškų išskyrimo metodai „UniformSampling“, SIFT, požymių palyginimo metodas kd-tree ir klaidingų požymių atitikimų pašalinimo metodas RANSAC.

Visų pirma reikėtų paminėti, kad nagrinėjama 3D objektų atpažinimo trimačių taškų technologija, ją sudarantys elementai ir metodai nėra pripažinti kaip geriausi ir nepakeičiami – technologija vis dar yra tyrimų bei eksperimentavimo stadijose. Dar pakankamai nauja technologija leidžia mums savo nuožiūra rinktis objektų atpažinimo priemones ir metodus. FPFH metodas buvo pasirinktas dėl jo populiarumo naudotojų tarpe ir gaunamų patenkinamų rezultatų. Iš deskriptorių pavadinimų PFH, PFHRGB ir SHOT, COLORSHOT matyti, kad tai yra išvestiniai metodai. PFHRGB, COLORSHOT yra vieninteliai metodai, PCL bibliotekos 1.6 versijoje, kurie naudoja ir spalvos ir formos požymius. Pastarieji metodai buvo pasirinkti kaip perspektyvūs objektų atpažinime remiantis tuo, kad visi šie metodai PCL bibliotekos kūrėjų, tyrėjų ir naudotojų duomenimis yra populiariausi, dažniausiai sutinkami literatūros šaltiniuose ir yra tinkami objektų atpažinimo užduotims spręsti. Visi minėti metodai, tam kad įvertinti objekto požymius, naudoja normales. Šių metodų panašumas suteikia galimybę lygiomis galimybėmis ištirti ir išrinkti efektyviausią požymių įvertinimo metodą.

3.1. PFH ir PFHRGB požymių įvertinimo metodas

Taško požymio histograma, toliau PFH (angl. *Point Feature Histograms*) deskriptorius, kurį pateikia ir aprašo B. Radas [2], skirtas įvertinti objekto geometrinius parametrus. Labai nesudėtingai galima nustatyti geometrinius parametrus tik skaičiuojant normales ir paviršiaus kreivumą, ką ir atlieka daugelis metodų, tačiau šie metodai įvertina tik keletą parametrų. Siūlomo PFH metodo naujumas grindžiamas tuo, kad naudojamos ne tik normalės, paviršiaus kreivumas bet ir aplinkinių kaimyninių taškų ryšiai tarp normalių. Taigi, siūlomo metodo pagalba gaunama daugiau paviršiaus geometrinių parametrų.

Metodo įgyvendinimui formuluojama visiškai nauja požymių erdvė, kuri vadinasi dvigubu kaimynystės žiedu. Sakykime, kad P yra trimačio taško geometrinės koordinatės $\{x_i, y_i, z_i\}$. Tada sakoma, kad paimtas taškas $p_i \in P$ turi dvigubą kaimynystės žiedą tada, kai tenkinama sąlyga:

$$(\exists) r_1, r_2 \in R, r_1 < r_2, \text{ tai yra } \begin{cases} r_1 \Rightarrow P^{k_1} \\ r_2 \Rightarrow P^{k_2} \end{cases} \text{ kur } 0 < k_1 < k_2, \quad (3.1)$$

čia r_1 ir r_2 dviejų žiedų spinduliai, jie nustato du skirtingus žiedų sluoksnius, kurie apibrėžia taško p_i požymius. Normalių apibrėžime jau buvo minėta, kad kaimyniniai taškai, kuriems skaičiuojamos normalės gali būti pasirenkami nurodant apskritimo spindulį r , taigi jau žinoma, kad pirmasis žiedas nurodo paviršiaus normalių fragmentą – plotą P^{k_1} . Normalių skaičiavimui pritaikomas pagrindinių komponentų metodas (angl. *Principal Component Analysis*) [35]. Antrajame žiede P^{k_2} turi būti užšifruota aplinkinių taškų paviršiaus geometrija, čia aplink pasirinktą tašką p_i įvertinamas vidutinis paviršiaus kreivumas. Paviršiaus kreivumui nustatyti naudojama daugiamatė reikšmių histograma. Daugiamatė histograma suteikia labai informatyvią taško požymio signatūrą. Taip pat ši signatūra labai gerai susitvarko su skirtingo tankumo duomenimis bei triukšmo lygiu esančiu taškų kaimynystėje. Minėti du žiedai tai taškų, artimoje kaimynystėje P^{k_2} , ir normalių esančių fragmente P^{k_1} ryšys, kuris sudaro PFH deskriptoriaus pagrindą. Šis ryšys parodo tai, kad stengiamasi paimti kuo geresnes padalinto paviršiaus, sąveikų tarp paskaičiuotų normalių krypčių, variacijas. Gaunama hiper-erdvė tai yra daugelio matmenų erdvė (6D), kurios kokybė priklauso nuo paskaičiuotų normalių pasirinktame taške [2].

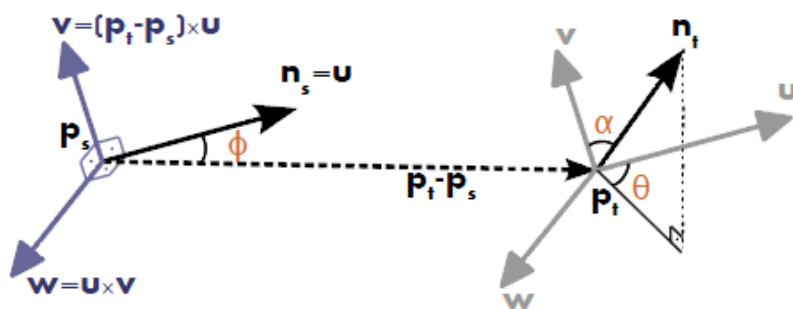
Visų pirmą reikia paskaičiuoti normales, tai galima atlikti daugeliu būdu. Matavimai atliekami kiekvienai kaimynystėje esančiai taškų p_i ir p_j porai, kurias apriboja žiedas P^{k_2} . Pirmasis PFH deskriptoriaus skaičiavimo žingsnis taškui p_i yra suskaičiuoti visas normales $\vec{n}$ visiems taškams, kurie yra žiede P^{k_2} . Įvertinamas santykinis skirtumas tarp dviejų taškų p_i ir p_j , ir juos jungiančiomis normalėmis $\vec{n}_i$ ir $\vec{n}_j$. Paviršiuje sukonstruojama Darboux atskaitos sistema [2, 16], tai yra savotiška lokali atskaitos sistema [16], kuri juda sukonstruotu paviršiumi. Darboux sistemos koordinatės apibrėžiamos vieno iš taško, kuriame skaičiuojami parametrai, koordinatėmis. Unikali Darboux atskaitos sistemos apibrėžiamas:

$$\begin{aligned} \text{jei:} \quad & \arccos(\vec{n}_i \cdot \vec{p}_{ji}) \leq \arccos(\vec{n}_i, \vec{p}_{ji}), \quad p_{ji} = p_j - p_i, \quad p_{ij} = p_i - p_j, \\ \text{tada:} \quad & \begin{cases} p_s = p_i, & n_s = n_i \\ p_t = p_j, & n_t = n_j \end{cases} \\ \text{priešingu atveju:} \quad & \begin{cases} p_s = p_j, & n_s = n_j \\ p_t = p_i, & n_t = n_i \end{cases} \end{aligned} \quad (3.2)$$

čia p_s yra šaltinis, p_t yra adresatas. Šaltinio taškas pasirenkamas taip, kad kampas tarp taškus jungiančios linijos ir normalės būtų minimalus. Tada Darboux sistemos taške p_s gali būti apibrėžtas kaip:

$$\begin{cases} u = n_s \\ v = u \times \frac{p_t - p_s}{\|p_t - p_s\|_2} \\ w = u \times v \end{cases} \quad (3.3)$$

11 paveiksle pateikiamas Darboux sistemos grafinis vaizdas, kuris kaip matyti parodo ryšį tarp dviejų kaimynystėje esančių taškų ir jų normalių. Gaunami tris nauji paviršiaus geometriniai požymiai $\langle u, v, w \rangle$.

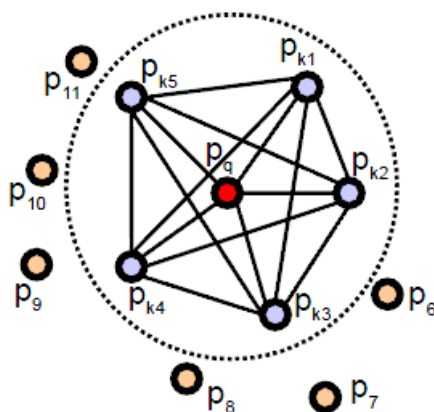


11 pav. Darboux atskaitos sistemos grafinis vaizdas ir PFH deskriptoriaus parametrai [2]

Naudojant Darboux sistemą, su u, v, w parametrais, įvertinamas skirtumas tarp dviejų normalių n_s ir n_t ir gaunami sekantys normalių kampiniai požymiai:

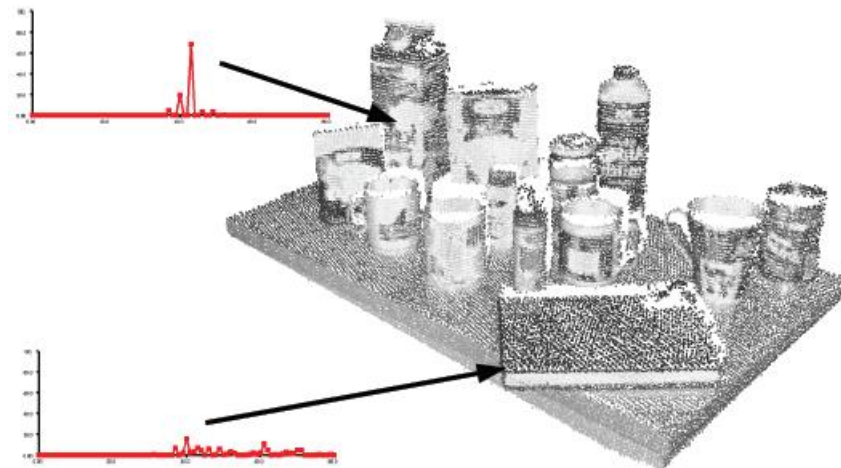
$$\begin{aligned} \alpha &= v \cdot n_t, \\ \phi &= u \cdot \frac{(p_t - p_s)}{d}, \\ \theta &= \arccos(w \cdot n_t, u \cdot n_t), \end{aligned} \quad (3.4)$$

čia d – Euklido atstumas tarp taškų p_s ir p_t , $d = \|p_t - p_s\|_2$. Gautas ketvertas $\langle \alpha, \phi, \theta, d \rangle$ skaičiuojamas kiekvienai taškų porai esančiai P^{k_2} kaimynystėje. Visi išvedimai parodo, kad dviejų taškų 12 verčių (x, y, z, n_x, n_y, n_z) ir jų normalių kiekis sumažinamas iki 4 požymių. Žemiau pateikiama PFH deskriptoriaus, skaičiavimus įtakojančių taškų, diagrama (12 pav.) [2].



12 pav. PFH deskriptoriaus, skaičiavimus įtakojančių taškų, diagrama [2]

Diagramoje pateikiamas centrinis taškas p_q kuriame skaičiuojamas deskriptorius, apskritimu apibrėžtoje kaimynystėje esantys taškai panaudojami skaičiavimams, už apskritimo ribų esantys taškai neįtakoja skaičiavimų. Matyti, kad kiekvienas taškas turi ryši su kiekvienu tašku kaimynystėje, esančiu tašku. Tam, kad taško PFH deskriptorius būtų užbaigtas, visų p_i taškų požymių ketvertai – $\langle \alpha, \phi, \theta, d \rangle$ o tiksliau tai tų požymių vertės sudedamos į histogramą. Histogramos formavimas prasideda nuo kiekvieno požymio vertės padalinimo į b poskyrius, tada nustatomas atitinkančių šį poskyrį skaičius (tiesiog sudedamos reikšmės).



13 pav. Skirtingų paviršių, taškų PFH deskriptorių, histogramos [2]

13 paveiksle pateikiamas minėtos histogramos pavyzdys. Matyti, kad iš skirtingų paviršių paimtų taškų histogramos yra skirtingos – unikalios, tai suteikia galimybę vertinti paviršiaus formos požymius. Svarbu paminėti, kad ketvirtas požymis d gali būti nenaudojamas dėl to, kad atstumas tarp kaimyninių taškų didėja priklausomai nuo stebėjimo taško ir jis gali būti gautas tiesiogiai iš jutiklio, jei jutiklis gali pateikti. PCL bibliotekoje pagal nutylėjimą yra naudojami trys požymiai ($f = 3$), kurie dalinami į 5 poskyrius ($b = 5$). Gaunama, kad PFH deskriptoriaus histograma bus sudaryta iš 125 skyrių (5^3). Šis deskriptorius įvertina, skaičiavimus įtakančių taškų, normales ir kampus tarp jų, taigi įvertinami tik geometriniai paviršiaus parametrai. Deskriptorius vadinamas 3D formos arba 3D geometriniu deskriptoriumi [2].

Tam tikri jutikliai gali pateikti ne tik gylį informaciją bet ir spalvos. Sekantis deskriptorius PFHRGB (angl. *Point Feature Histograms + RGB*) paremtas jau prieš tai aptarto deskriptoriaus PFH algoritmu, tačiau šis 3D deskriptorius dar įvertina ir spalvos požymius. Visa histogramos sudarymo schema geometriniais požymiais lieka nepakitusi, prie jos dar pridėjama RGB spalvų sistemos histograma. RGB spalvų sistemos požymiai kaip žinome yra trys – R (raudona), G (žalia) ir B (mėlyna) spalvų santykis. Šie požymiai taip pat pagal nutylėjimą PCL bibliotekoje, šiam deskriptoriui, dalinami į 5 poskyrius. Taigi visą PFHRGB deskriptoriaus histogramą sudaro 250 skyrių.

Pateikti tyrimų rezultatai [2] parodo, kad geriau pasirinkti nelyginį poskyrių skaičių, tokiu būdu požymių histogramos, tuo pačiu ir patys deskriptoriai, yra atsparesni triukšmams. Pateikiamas bendras teorinis deskriptoriaus algoritmo išskaičiavimo sudėtingumas $O(nk^2)$. Debesies taškų rinkiniui P , kur n – taškų skaičius, k – taškai kaimynystėje. Deskriptoriaus PFH skaičiavimo sudėtingumą ir skaičiavimo laiką galima sumažinti net 75 %, tačiau tam reikia įgyvendinti papildomus algoritmus. Iš skaičiavimų pastebima, kad kai kurioms taškų poroms skaičiavimai atliekami po kelis kartus, taigi jei atlikti pakartotinai naudojamų apskaičiuotų verčių saugojimą, bei papildomą taškų indeksavimą, kuris sumažina artimiausių kaimynų paieškos laiką, galima deskriptorių optimizuoti. PFH ir PFHRGB deskriptoriai pasižymi didele diskriminantine galia bei atsparūs triukšmams.

3.2. FPFH požymių įvertinimo metodas

FPFH (angl. *Fast Point Feature Histograms*) deskriptorių taip pat aprašo B. Radas [1, 18]. Kaip jau buvo paminėta, prieš tai aptarto deskriptoriaus PFH išskaičiavimo sudėtingumas, gali būti sumažintas naudojant papildomą apskaičiuotų duomenų saugojimą ir taškų indeksavimą. Trumpas duomenų skaičiavimo laikas, kaip žinoma yra labai svarbus realaus laiko sistemose o per didelis duomenų tankumas yra viena iš problemų. Taigi, sekantis deskriptorius – FPFH yra greitoji PFH deskriptoriaus skaičiavimo versija būtent skirta greitam formos požymių įvertinimui.

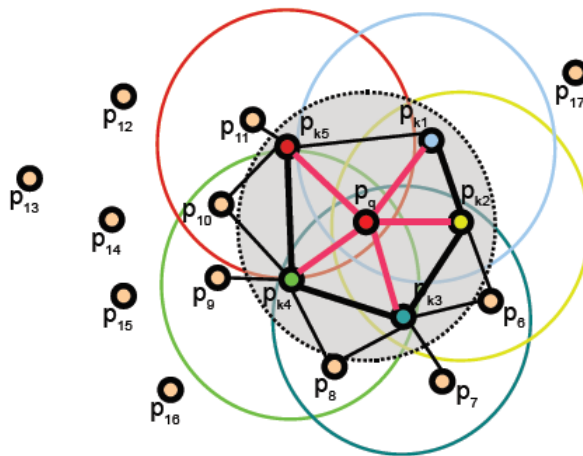
Įdomu yra tai, kad būtent sumažinus išskaičiavimo sudėtingumą nuo $O(nk^2)$ iki $O(nk)$ FPFH deskriptoriaus diskriminantinė galia išlieka ne prastesnė nei jo pirmtako PFH deskriptoriaus. Visų pirma bus apžvelgti pagrindiniai FPFH deskriptoriaus skaičiavimo žingsniai [2]:

1. Visų pirma kaip ir PFH deskriptoriuje, skaičiuojamas jau žinomas geometrijos požymių tripletas – $\langle \alpha, \phi, \theta \rangle$. Šis tripletas pavadinamas supaprastinta taško požymio histograma SPFH (angl. *Simplified Point Feature Histogram*).

2. Po to kiekvienas taškas k esantis kaimynystėje yra perskaičiuojamas pagal žemiau pateiktą formulę (3.5).

$$FPFH(p_q) = SPFH(p_q) + \frac{1}{k} \sum_{i=1}^k \frac{1}{w_k} \cdot SPFH(p_k), \quad (3.5)$$

čia w_k – atstumas tarp taško p_q ir kaimyninio taško p_k . Žemiau pateikiama FPFH deskriptoriaus svorių pasiskirstymo diagrama, skaičiavimus įtakojantys taškai ir ryšiai tarp jų (14 pav.). Skaičiavimams panaudojami ir tie duomenys, kurie buvo paskaičiuoti anksčiau – SPFH. Šiuo požiūriu FPFH deskriptorius yra optimalesnis nei PFH – nėra skaičiuojamos vėliau nepritaikomos reikšmės.



14 pav. FPFH deskriptoriaus svorių pasiskirstymo diagrama [2]

Pateiktoje diagramoje (14 pav.) centrinis taškas p_q , kuriam skaičiuojamas FPFH deskriptorius, turi tiesioginį ryšį su kiekvienu tašku esančiu k taškų kaimynystėje. Čia taško p_q kaimynystės ribos apibrėžiamos apskritimu. Kiekvienas k kaimynystėje taškas, turi ryši su kiekvienu savo kaimynystėje esančiu tašku. Ryškiau pažymėti ryšiai parodo, kad tam tikri ryšiai buvo perskaičiuoti ne viena karta. Taigi, šį metodą taip pat dar galima bandyti dar labiau optimizuoti.

Šių deskriptorių geometriniai požymiai sudedami į histogramą, kurios skyrių skaičius buvo 125. Kadangi kiekvienas geometrinis požymis pagal nutylėjimą dalinamas į 5 dalis, tai ši dalinimą galima įsivaizduoti kaip trijų matmenų $5 \times 5 \times 5$ kubą. Šio kubo vienas iš poskyrių atitinka tašką, kuris turi tam tikras šių požymių reikšmes. Taigi, jei kažkuris iš taškų šių požymių neturi, yra galimybė kad histograma vistiek bus pildoma to taško atžvilgiu. Gaunama pilnai koreliuota požymių erdvė. PFH deskriptorius galėtų būti toliau optimizuojamas jei galima būtų išspręsti šią koreliacijos problemą, kuri pastebima požymio histogramos erdvėje. FPFH deskriptoriuje siūloma minėtas koreliuotas histogramos reikšmes iškoreliuoti. Galima kiekvienam taško požymio matmeniui atlikti histogramos padalinimą į d poskyrius, tada šiuos poskyrius sukongcentruoti.

Apibendrinant galima paminėti kuo skiriasi PFH ir FPFH deskriptoriai [2]:

1. Kaip iš pateikiamų deskriptorių diagramų matyti, kad FPFH deskriptoriaus taškai kaimynystėje nėra jungiami kiekvienas su kiekvienu, dėl to mažiau jungčių ir mažiau atliekama skaičiavimų.
2. PFH įvertina paviršių aplink tašką p_q , kurio ribas tiksliai nustato apskritimas su spinduliu r . FPFH įvertina ir išorinius taškus – didesnės ribos.
3. Kadangi FPFH deskriptoriuje vertinami svorių koeficientai SPFH, tai FPFH deskriptorius panaudoja ir prieš tai skaičiuotų koeficientų vertes – mažiau skaičiavimų, kurie lieka nepanaudoti.

3.3. SHOT ir COLORSHOT požymių įvertinimo metodas

SHOT (angl. *Signature of Histograms of Orientations*) deskriptorius taikomas taškų debesims su normalėmis, geometriniams paviršiaus požymiams vertinti [22]. Šio deskriptoriaus sudarytojai F. Tombaris, S. Saltis, L. Stefanas [31] ypatingą dėmesį skyrė lokaliai 3D atskaitos sistemai LRF, kuri taikoma 3D deskriptoriuose. Didžiausias dėmesys skirtas LRF sistemos kartotinumui ir tvirtumui, kurie turi įtakos 3D deskriptoriaus savybėms.

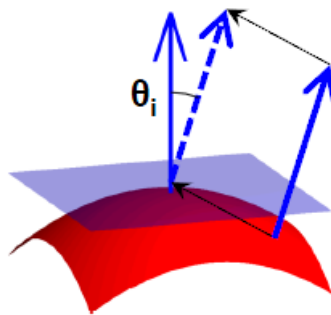
SHOT požymių skaičiavimo metodas įkvėptas SIFT [26] deskriptoriaus, kuris yra vienas patikimiausių deskriptorių skirtų 2D vaizdams apdoroti. SIFT skaičiuojamas naudojant histogramas, kurios sudaromos taikant specifinio tinklelio taškus bei gradiento intensyvumą. Globaliai histogramai sudaryti naudojama daug lokalių histogramų – skaičiuojami geometriniai lokalių histogramų rinkiniai. SIFT taip pat pasižymi didele diskriminantine galia.

Nors beveik kiekvienas naujas deskriptoriaus pasiūlymas lokaliai formos aprašymui yra sulyginamas su LRF, eksperimentai visada būna sutelkti į rezultatus gautus sujungus LRF ir deskriptorių, kai atskirai LRF poveikis deskriptoriaus savybėms nėra tyrinėjamas. SHOT deskriptoriuje bandoma sumažinti LRF dviprasmiškumą. Tyrimų trūkumas sąlygoja tai, kad LRF naudojimo galimybės nėra aiškiai aprašytos, taip pat stokojama eksperimentinių tyrimų, kurie aprašytų atsparumą triukšmams, kartotinumą, patikimumą, unikalumą. Ištyrus LRF yra įrodyta, kad norint gauti efektyvų lokalaus paviršiaus aprašymą, LRF kartotinum savybė yra būtina. LRF yra nekintama perskaičiavimams, pasukimams ir atspari triukšmui, tai suteikia galimybę 3D deskriptoriui išlikti pastoviam esant šaltinio duomenų permainoms, taip pat kaip dvimačiai deskriptoriai išlieka pastovūs pasukimams ir mastelio pakeitimui. Taip pat svarbios ir kitos savybės: LRF unikalumas ir nedviprasmiškumas. 3D formos deskriptoriaus efektyvumo savybių pagerinimui siūloma naudoti histogramų ir signatūrų kombinacijas [36].

Histogramomis paremti metodai surenka geometrinius ar topologinius išmatavimus (taškų skaičius, trikampių tinklelio sritys) į histogramas pagal specifines kvantuotas sritis (taškų koordinatės, kreivumas), kurioms reikalinga yra geometrinės ašies informacijos arba lokalsios atskaitos sistemos. Signatūros yra gerai aprašomos kadangi jose naudojama lokalizuota informacija. Histogramos tuo tarpu aprašomąją galią iškeitė į diskriminantinę galią – tvirtumą ir patikimumą, supausdamos geometrinę struktūrą į saugyklas. Histogramomis paremti metodai priklausantys nuo LRF yra pagrįsti požymio taško normalėmis. Signatūrų pagrindų sukurti metodai remiasi arba 3D kreive arba poza. Šiuo atveju užšifruojami daugiakampės aproksimacijos kampai tarp nuosekliai einančių segmentų paviršiaus kraštų. Tada, kaip ir 3D kreivė, užšifruojamas lokalus išsidėstymas ant paviršiaus, naudojant geodezinį apskritimą (16 pav.), kuris yra centruotas pagal skaičiuojamą tašką. Signatūra gaunama pagal 3D kreivės aukščio žymę, kuri įgyta iš suskirstytos į dalis,

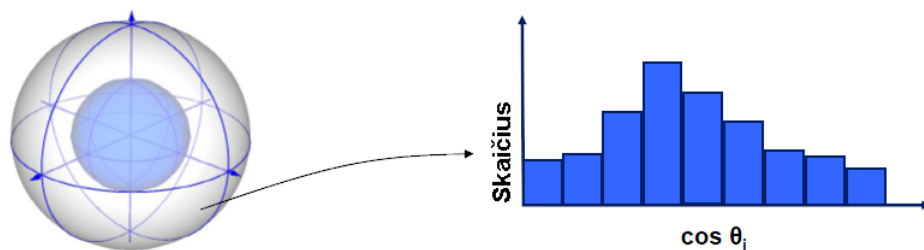
centruotos pagal skaičiuojamą tašką, sferos (16 pav.). 3D taško signatūra gaunama užšifruojant normalių kampų variacijas ir horizontalės radialines variacijas, tai atliekama išilgai skirtingų geodezinių apskritimų, kurie suprojektuoti ant liestinės plokštumos.

SHOT 3D deskriptorius užšifruoja histogramas, kuriomis apibrėžiamos atraminių taškų normalės. Atraminių taškų normalės naudojamos kadangi jos geriau reprezentuoja lokalią struktūrą nei paprastos 3D koordinatės. Histogramų naudojimas suteikia filtravimo efektą, kuris yra būtinas norint gauti didesnę atsparumą triukšmams. Turint unikalią ir patikimą 3D LRF galima padidinti deskriptoriaus diskriminantinę galią, tam panaudojant atraminių taškų geometrinę informaciją. Tokiu būdu imituojama signatūra. Visų pirma skaičiuojamos lokalsios histogramos, jos sudaromos iš 3D reikšmių, kurios gautos uždedant ant atraminių taškų 3D tinklelį (16 pav.). Gautos lokalsios histogramos atitinkamai grupuojamos tam kad sudaryti galutinį deskriptorių. Taigi aptartas deskriptorius yra signatūros ir histogramos susikirtime, iš to kyla jo pavadinimas – orientacijos histogramų signatūra SHOT (angl. *Signature of Histograms of Orientations*).



15 pav. SHOT deskriptoriaus, kampų skaičiavimas [37]

Kiekvienai histogramai sudaryti, pagal kosinuso kampo funkciją $\cos\theta_i$, suskaičiuojamos į histogramos skyrius patekusios vertės (16 pav., „Skaičius“). Minėti kampai θ_i skaičiuojami tarp normalių, šios normalės skaičiuojamos kiekvienam taškui, kuris atitinka tinklelio skyrių n_{v1} ir taškui n_u kuriam skaičiuojamas požymis (15 pav.). Kosinuso funkciją gali būti suskaičiuota naudojant normalių sandaugą $\cos\theta_i = n_{v1} \cdot n_u$. Struktūros signatūrai naudojamas izotropinis sferinis tinklelis, kurio dalys apima radialinę, azimutinę ir aukščio ašis (16 pav.).



16 pav. SHOT deskriptoriaus histogramos verčių sudarymas [37]

Kadangi kiekviena pateikto sferinio tinklelio reikšmė aiškiai užšifruoja parametrus, kuriuos reprezentuoja lokali histograma, galima naudoti ir grubų sferinio tinklelio dalinimą ir taip neapsunkinti deskriptoriaus. Nustatyta, kad tinkamiausias skyrių skaičius yra 32, kurį sudaro 8 azimuto, 2 aukščio ir 2 radialiniai padalinimai. 16 paveiksle pateikiamas 4 azimuto padalinių sferinis tinklelis.

Kiekvienas, histogramos skyriaus, skaičius kiekvienam matmeniui yra dauginamas iš svorio koeficiento $(1 - d)$. Lokaliai histogramai d tai einamojo įrašo atstumas nuo centrinės reikšmės tame skyriuje. Tinklelio aukščiui ir azimutui d yra kampinis atstumo įrašas nuo centrinės apimties reikšmės. Išilgai radialinio matmens, d tai įrašo Euklido atstumas nuo centrinės apimties reikšmės. Išilgai kiekvieno matmens d koeficientas normalizuojamas pagal atstumą tarp dviejų artimiausių skyrių ar apimčių. Kad deskriptorius būtų vienodai efektyvus skirtingiems duomenų tankumams, atliekama viso deskriptoriaus normalizavimas ir jo suma sulyginama su 1 [31].

CSHOT arba COLORSHOT (angl. *Signature of Histograms of Orientations*) deskriptorius, kuris kartu naudoja formos ir spalvos požymius. CSHOT deskriptoriaus formulavimui naudojamas SHOT deskriptoriaus pagrindas: atraminiai taškai, spindulys, histogramos skyrių skaičius. CSHOT tai SHOT deskriptorius natūraliai išplėstas – apjungtas su tekstūra. Norint išsiaiškinti tekstūrų ir geometrijos parametrų apjungimą, reikia matematiškai apibrėžti SHOT deskriptorių. SHOT histograma – $SH_{G,f}(P)$ apibrėžiama kaip visų lokalių histogramų signatūra. Tai lokalsios histogramos, kurios suskaičiuotos po visą atraminių taškų sferinį plotą Q , aplink požymio tašką P . Minėta histogramos signatūra priklauso nuo dviejų parametrų G – skaičiuojamo taško požymių vektorius ir f – metrika, pagal kurią lyginamos taškų savybės. Šiuo atveju G vektorius tai normalių N skaičiavimo rezultatas. Skaičiuojant signatūros histogramą, kiekvienai taškų porai skaičiuojami požymiai, kurie atitinka normalių N_P, N_Q sandaugą:

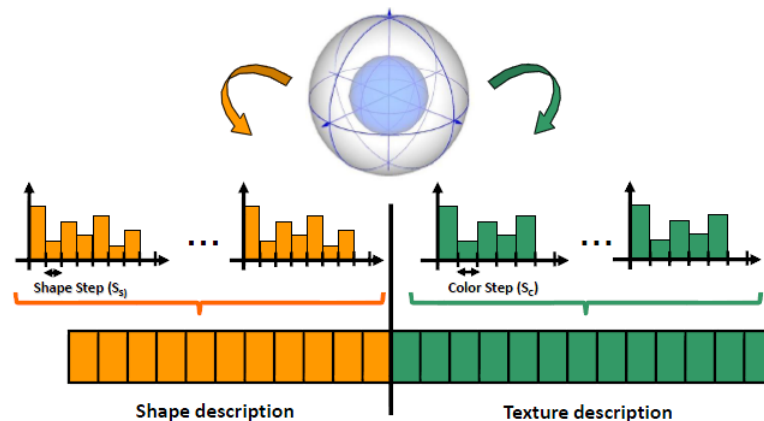
$$f(G_P, G_Q) = p(N_P, N_Q) = N_P \cdot N_Q, \quad (3.6)$$

Histogramą sudaro m signatūrų. Signatūros yra susijusios skirtingomis poromis, kurios yra skaičiuojamos sferinėje atraminių taškų erdvėje. Galiausiai gautos signatūros tarpusavyje sujungiamos tam kad sudaryti požymio taškui P deskriptorių $D(P)$:

$$D(P) = \bigcup_{i=1}^m SH_{(G,f)}^i(P), \quad (3.7)$$

Tam, kad įterpti į SHOT deskriptorių tekstūros informaciją, reikia surasti tinkamą vektorių, kuris apibrėžia taško atžvilgiu tekstūros savybę. Viso to reikia tam kad galima būtų palyginti dviejų deskriptorių formos ir tekstūros savybes vienodose metrikose. 3.6 formulėje vietoje geometrinių

požymių naudojami RGB spalvų sistemos spalvų intensyvumai, kurie pažymėti trumpiniu – R : $p(R_P, R_Q)$. Taip pat yra galimybė CSHOT deskriptoriuje naudoti ir kitas spalvų sistemas.



17 pav. CSHOT deskriptoriaus histogramos sudėtis [32]

Deskriptoriaus tekstūros informacija pateikiama *Lab* erdvėje. *Lab* erdvės porų tripletai (centrinio taško ir jo kaimynų) gali būti skaičiuojami naudojant specifines metrikas, tokias kaip *CIE94*, *CIE2000*. Galutinis CSHOT deskriptorius pagrįstas dvejomis histogramų signatūromis (17 pav.).

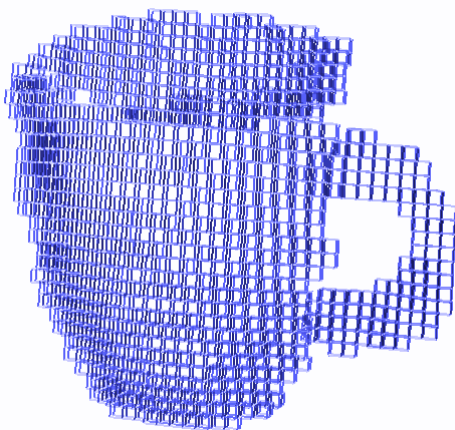
Taigi, šiame 3D deskriptoriuje panaudojamos histogramos ir signatūros savybės. Deskriptorius skirtas lyginti taškų požymių atitikmenis taškas į tašką. Taikomas metodas pateikia patikimą LRF, kuri skirtingai nuo kitų aprašomų, yra unikali ir vienareikšmė. Lokalūs pagalbiniai taškai suskirstomi pagal sferinį tinklėlį. Kiekvienai sferinio tinklelio reikšmei, histograma sudaroma iš kosinuso kampų θ_i . Šie kampai skaičiuojami tarp normalių kiekvieno artimiausio taško ir skaičiuojamo požymio taško normalės. Kvantavimo iškraipymų glotninimui panaudojama keturių linijų interpoliacija. SHOT deskriptorius priklauso nuo kartotinės LRF, kuri pagrįsta kaimyninių taškų sklaidos matricos eigen reikšmių dekompozicija. Atsižvelgiant į LRF, izotropinis sferinis tinklelis yra nustatomas tam, kad užšifruoti erdvinei lokalizuotai informacijai, kuri apibūdina struktūros signatūrą. Aprašomas kiekvienas, normalių histogramos, sferinio tinklelio sektorius. Iš gautų ir sugretintų lokalių histogramų sudaromas galutinis deskriptorius. Svarbu paminėti, kad COLORSHOT arba kitaip CSHOT, spalvos informacijai – tekstūrai koduoti, naudoja tris kartus daugiau duomenų nei pirmtakas SHOT.

3.4. „UniformSampling“ būdingųjų taškų išskyrimo metodas

Praktika rodo, kad ne visi taškai taškų debesyse yra informatyvūs – informacijos perteklius. Taip pat, dėl įvairių aplinkybių ir triukšmų, ne visada taškų požymiai – charakteristikos yra geros, taigi galima filtruoti charakteringuosius taškus o dar geriau išskirti ir naudoti tik charakteringiausius

taškus. Ar tai efektyvu priklauso ir nuo naudojamų duomenų. Šių metodų tikslas yra išskirti charakteringuosius nekintamus požymius. Pasirinktas perspektyvus būdingųjų taškų išskyrimo metodas – „UniformSampling“ turi dvejopą veikimą, kuris pagrįstas 3D vokseliniu tinkleliu. Jei šis metodas naudojamas kaip duomenų tankumo keitimo metodas, tai į funkcijos įėjimą paduodamas taškų debesis o išėjime gaunamas naujas, pakitusio (sumažinto arba padidinto) tankumo, taškų debesis. Jei metodas naudojamas būdingųjų taškų išskyrimui, tai į funkcijos įėjimą paduodamas taškų debesis o išėjime gaunami indeksai, kurie gali būti panaudoti taškų debesyje tam kad nustatyti būdingųjų taškų vietas. Minėti du būdai skiriasi tik tuo kad jie įgyvendinti skirtingose PCL bibliotekos dalyse ir indeksų gavimo būdas yra spartesnis kadangi nereikia kurti papildomų taškų debesų. Atliekant tyrimus su nedideliais taškų debesimis spartos veikimo skirtumas neįjuntamas.

Vokselis tai dviejų anglišų žodžių derinys (angl. *volumetric*) tūrio ir (angl. *pixel*) vaizdo elementas [38, 39], kuris naudojamas trimačiams (3D) duomenims apibrėžti. Kiekvienas vokselis, apibrėžiamas kaip trimatis (3D) kubelis, kuris turi kraštinės parametrus nuo kurių priklauso vokselinio tinklelio tankumas o tuo pačių ir duomenų tankumas. 3D vokselinis tinklelis tai tinklelio pavidalo funkciją, kuri pritaikoma taškų debsiui.



18 pav. 3D vokselinio tinklelio pritaikymas

18 paveiksle pateikiamas 3D vokselinis tinklelis, kuris pritaikytas taškų debsiui. Visi taškai, kurie patenka į kubelio vidų yra aproksimuojami pagal savo masių centrus. Aproksimavimas pagal taškų masių centrus atliekamas ilgiau nei pagal kubelio centrą, tačiau tokiu būdu tiksliau pateikiamas esminis paviršius [2]. Pagrindinis 3D vokselinio tinklelio parametras yra vokselio kraštinės ilgis – l_v , kurį pasirenka naudotojas.

Apibendrinant galima paminėti tai, kad šis būdingųjų taškų išrinkimo metodas ir kartu duomenų tankumo keitimo metodas yra paprasčiausias. Metodas vienodai įvertina visus taškų požymius, todėl gali būti taikomas kai nėra tiksliai žinoma kokius duomenis reikės naudoti objektų atpažinimo įgyvendinimui. Šis metodas yra sparčiausias iš visų, PCL bibliotekos 1.6 versijoje, pateiktų būdingųjų taškų išrinkimo metodų.

3.5. SIFT būdingųjų taškų išskyrimo metodas

Kitas pasirinktas perspektyvus būdingųjų taškų išskyrimo metodas – SIFT (angl. *Scale-Invariant Feature Transform*). Kaip jau buvo minėta antrame skyriuje, SIFT metodas gali būti naudojamas kaip būdingųjų taškų detektorius ir kaip deskriptorius. Visą SIFT metodą sudaro detektoriaus ir deskriptoriaus dalys [26]:

1. Mastelio-erdvės ekstremumų paieška
2. Būdingųjų taškų lokalizacija (būdingųjų taškų tobulinimas ir netinkamų taškų atmetimas)
3. Orientacijos priskyrimas būdingiesiems taškams (pašalinami mastelio ir pasukimo efektai)
4. Sudaromas deskriptorius (naudojamos orientacijų histogramos)

Pirma ir antra metodo dalys skirtos SIFT detektoriumi aprašyti, trečia ir ketvirta – SIFT deskriptoriumi. Taigi bus aptartos tik dvi pagrindinės – detektoriaus SIFT metodo dalys. Mastelio erdvės sukūrimui ir ekstremumų paieškai naudojamas, tolydžios Gauso funkcijos, Gauso branduolys.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \quad G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}, \quad (3.8)$$

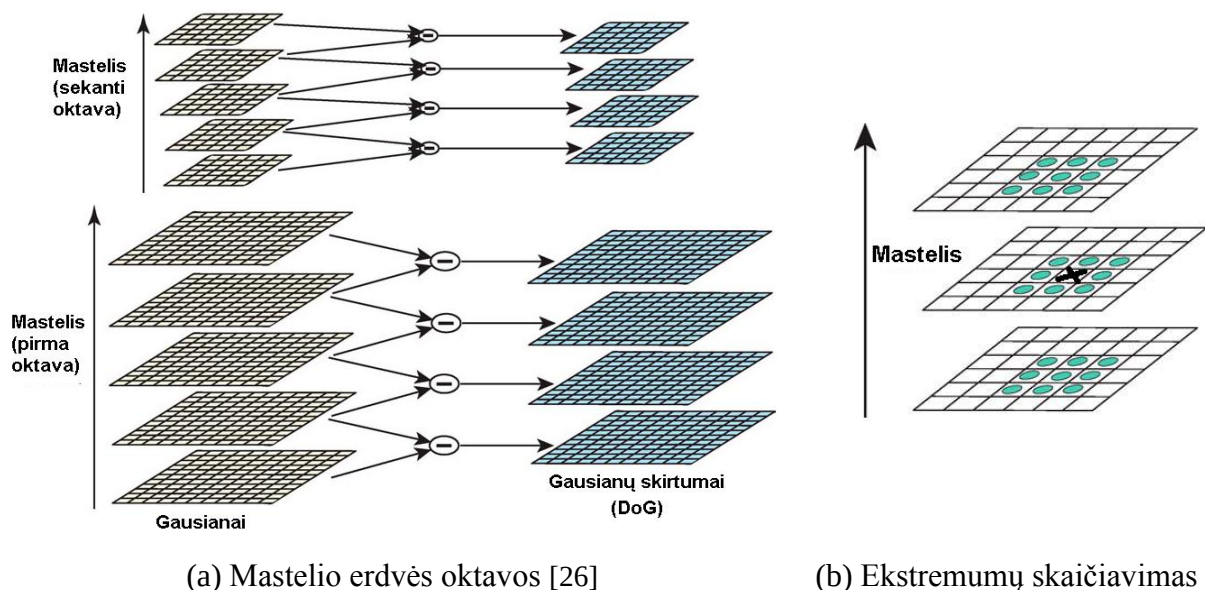
čia L – mastelio erdvės funkcijos reikšmė pasirinktame taške, (x, y) – pasirinkto taško koordinatės, σ – glotninimo spindulys, G – Gauso funkcijos branduolys, I – vaizdo funkcija, $*$ – sąsūkos operacija. Tam, kad padidinti būdingųjų taškų paieškos efektyvumą, naudojamas Gauso funkcijų (Gaussianų) skirtumas DoG (angl. *Difference of Gaussian*) su skirtingais glotninimo funkcijos spinduliais:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma), \quad (3.9)$$

čia k – daugiklis ($\sqrt{2}$), kuris keičia glotninimo spindulį. Panaudojant Gaussianų skirtumų funkciją – D , atliekama Gauso mastelio normalizaciją panaudojant Laplaso funkciją su koeficientu σ^2 . Tokiu būdu gaunami skirtingo glotnumo vaizdai. Taigi, naudojant skirtingus glotninimo koeficientus, randant Gauso funkcijų skirtumus iš tiesu gaunamos Laplaso-Gauso funkcijos.

Gauso kintamo dydžio erdvė yra tiesinė, atspari postūmiams, pasukimams, mastelio pokyčiui. Atsparumas mastelio pokyčiams atsiranda dėl būdingųjų taškų suradimo, kurie paimti iš skirtingo mastelio vaizdų. Tam, kad sudaryti skirtingų mastelių vaizdus, sudaroma Gaussianų piramidė – visa kintamo dydžio erdvė padalinama į ruožus – oktavas. Kintamo dydžio erdvės dalis, kuri užima sekančią oktavą yra du kartus didesnė nei prieš tai užimančios. Kiekvieną kartą pereinant vis prie kitos oktavos atliekamas duomenų pergrupavimas ko pasekoje duomenų kiekis sumažėja du kartus (19 pav., a).

Kadangi Gausianų kiekis N piramidėje labai didelis, išrenkami tik kai kurie iš jų, atitinkamai pagal glotninimo žingsnį σ . Kiekvienoje DoG piramidėje ieškomi lokalių minimumų taškai. Kiekvienas, to pačio DoG vaizdo, taškas palyginamas su, tame pačiame DoG vaizde, kaimynystėje esančiais, aštuoniais taškais. Taip pat minėtas taškas lyginamas su kitais devyniais taškais DoG vaizde, kuris piramidėje yra vienu lygiu žemiau (19 pav., a). Iš viso atliekami 26 palyginimai. Jei 19 paveiksle (19 pav., b) pažymėtas taškas (X) yra DoG funkcijos lokalus ekstremumas, tai šis taškas laikomas būdinguoju.



19 pav. SIFT metodo skaičiavimo schemas

Būdingųjų taškų lokalizacijos žingsnyje atliekama atrinktų būdingųjų taškų – ekstremumų patikra. Pirmiausia, taško tikslumu, nustatomos atrinktų taškų koordinatės. Tai atliekama panaudojant antros eilės Teiloro daugianarį, kuris atlieka DoG funkcijos aproksimaciją:

$$D(x) = D + \frac{\partial D^T}{\partial x} + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x, \quad (3.10)$$

čia D – DoG funkcija ir jos išvestinė apskaičiuota pasirinktame taške, $x = (x, y, \sigma)^T$ – taško poslinkio vektorius, pirmoji DoG išvestinė tai gradientas, antroji – Heso matrica. Teiloro daugianario ekstremumas randamas skaičiuojant DoG funkcijos išvestinę, kuri po to prilyginama nuliui. Gaunamas paskaičiuoto ekstremumo taško poslinkis:

$$\hat{x} = D + \frac{\partial D^{-1}}{\partial x} \frac{\partial D}{\partial x}, \quad (3.11)$$

Jei, bet kuriame mastelyje, vektoriaus $\hat{x}$ poslinkis yra didesnis nei pusė (0,5), tai reiškia kad buvo netinkamai išrinktas ekstremumo taškas – ekstremumas yra kaimynystėje. Tokiu atveju pereinama

link kaimyninio taško, pagal nurodytas vektorių komponentes. Taip tikrinami visi kaimyniniai taškai. Jei po šių patikrinimų, paliekamos oktavos ribos, tai pasirinkto ekstremumo taško siūloma nenagrinėti. Funkcijos ekstremumo vertė $D(\hat{x})$ naudojama žemo kontrasto taškams atmesti:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x}, \quad (3.12)$$

Metodo stabilumui užtikrinti nepakanka tik eliminuoti taškus su žemu kontrastu. DoG funkcija įsivaizduojama kaip paviršius, tada DoG funkcijos reikšmės labai keisis (didelis kreivumas), netgi tada, kai taškų padėtis išilgai objekto kraštų bus prastai įvertinta, todėl metodas tampa nestabilus net dėl mažo triukšmo. Taigi jei taškas yra ant objekto ribos ir blogai apšviestas, tai šį tašką reikia eliminuoti. Pagrindiniam kreivumui vertinti naudojama 2×2 Heso matrica $\mathbf{H}$, kuri skaičiuojama vertinant būdingojo taško mastelį ir padėtį [26].

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}, \quad (3.13)$$

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta = \lambda_1 + \lambda_2, \quad (3.14)$$

$$Det(H) = D_{xx} D_{yy} - (D_{xy})^2 = \alpha\beta = \lambda_1 \lambda_2, \quad (3.15)$$

čia $Tr(H)$ – matricos $\mathbf{H}$ pėdsakas (įstrižainės elementai), $Det(H)$ – matricos $\mathbf{H}$ determinantas, α – didžiausia eigen reikšmė, β – mažiausia eigen reikšmė, λ_1, λ_2 – pirmoji ir antroji eigen reikšmė. Įdomu yra tai, kad minėtus kreivumo parametrus galima apskaičiuoti naudojant eigen reikšmes. Tačiau, kurių determinantai neigiami yra atmetami. Po to taikomas slenkstis:

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r}, \quad \alpha = r\beta, \quad (3.16)$$

čia r – dydžio santykis tarp mažos ir didelės eigen vertės. Būdingieji taškai kurių santykis tarp eigen reikšmių yra didesnis nei r – atmetami.

Išnagrinėtas būdingųjų taškų išskyrimo metodas atsparus:

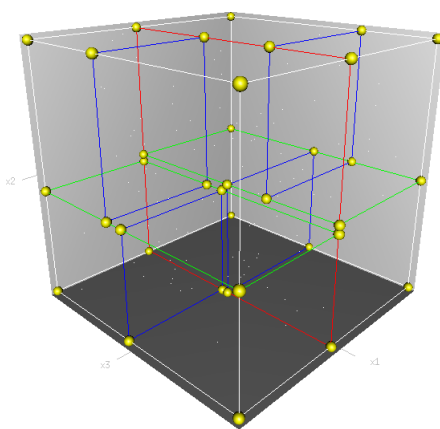
- vaizdo mastelio pokyčiams;
- vaizdo pasukimams;
- apšvietimui;
- stebėjimo taško kampo pokyčiui;

Aptartas SIFT metodas – tai lokalus būdingųjų taškų išskyrimo metodas. Būdingiesiems taškams nustatyti naudojamas artimiausių taškų intensyvumo pokytis. Naudojami skirtingo glotnumo vaizdai, kurie gaunami taikant Gauso-Laplaso filtrą. Skirtinguose vaizdo masteliuose ieškoma būdingųjų taškų, kuriuose Gauso-Laplaso funkcijos reikšmės – ekstremumai. Būdingųjų

taškų efektyvumui gerinti, atliekamas netinkamų būdingųjų taškų filtravimas naudojant kontrasto ir objekto kraštų slenksčius. Pagrindiniai, PCL bibliotekoje įgyvendinto SIFT metodo, parametrai yra keturi: minimalus kontrastas (*min_contrast*), mažiausias mastelis (*min_scale*), oktavų skaičius (*nr_octaves*) ir mastelių skaičius oktavoje (*nr_scales_per_octave*).

3.6. Kd-tree požymių palyginimo metodas

Kd-tree tai tokia duomenų struktūra, kurią sudaro k matmenų medis, kuris saugoja k matmenų taškus ir atstumą tarp jų. 20 paveiksle pateikiamas trijų matmenų struktūros pavyzdys. Struktūra gali būti naudojama, efektyviai artimiausių kaimyninių taškų, paieškai. Taip pat struktūra sėkmingai naudojama, kai norima nustatyti atitikimus tarp taškų ar požymių deskriptorių. PCL bibliotekoje yra įgyvendinta pagalbinė biblioteka – FLANN (angl. *Fast Library for Approximate Nearest Neighbors*), kuri paspartina artimiausių kaimynų paiešką [40]. FLANN atlieka matmenų mažinimą – aproksimaciją. FLANN turi keletą artimiausių taškų paieškos metodų, vienas iš jų automatiškai parenkamas pagal įėjimo duomenis. Tam kad pritaikyti FLANN reikia papildomai įrašinėti FLANN biblioteką. Kadangi šiame darbe apdorojami nedideli taškų debesys, tai net naudojant kd-tree metodą, tai skaičiavimo laikas nėra juntamas.



20 pav. Trimatė kd-tree (3d-tree) duomenų struktūra [41]

Taškų požymiams lyginti PCL bibliotekoje yra įgyvendintas kd-tree metodas. Kd-tree yra pagrindinis ir dažniausiai, PCL bibliotekoje, taikomas taškų panašumų palyginimo metodas, kadangi nereikia papildomai rašyti palyginimo funkcijų – patogus naudojimas. Šis metodas kaip jau minėta gali būti naudojamas požymiams tarp taškų lyginti pagal taškų koordinates ir pagal taškų suskaičiuotus požymius – deskriptorius. Šiame darbe bus atliekamas taškų požymių palyginimas remiantis spalvos ir formos požymiais.

Visų pirma, prieš atliekant panašumų paiešką, reikia sudaryti kd-tree duomenų struktūrą. Kd-tree struktūra sudaroma kai į įėjimą paduodami taškų požymiai. Tada daugiamatėje erdvėje taškų

požymiai klasifikuojami, taip pat tarp jų nustatomi atstumai. Atstumai, PCL bibliotekoje, įgyvendintame kd-tree metode skaičiuojami pagal Euklido atstumą arba kitaip L^2 metrika [42]:

$$L^2 = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{k=1}^n (p_k - q_k)^2} \quad (3.17)$$

čia $p = (p_1, p_2, \dots, p_n)$ – pirmas taškas n -matėje erdvėje, $q = (q_1, q_2, \dots, q_n)$ – antras taškas n -matėje erdvėje, k – kintamasis. Kai atliekamas taškų požymių palyginimas, taikant PCL bibliotekoje įgyvendintą kd-tree metodą, į metodo įėjimą paduodami, dviejų objektų, taškų požymiai – deskriptorių reikšmės o išėjime gaunami, dviejų objektų, taškų požymių atitikimai (atitikimų pora) ir atstumas tarp atitikusių taškų. Atitinkančiais taškais laikomi tie taškai, kurie būdami skirtinguose objektų paviršiuose turi panašius arba vienodus požymių vektorius – deskriptorius. 21 paveiksle pateikiamas dviejų objektų (bananų) taškų porų atitikimo pavyzdys.



21 pav. Taškų porų atitikimai

Gautoms atitikimų poroms o tiksliau jų požymių vektoriams skaičiuojami atstumai, jei jie jau suskaičiuoti tai gali būti randamas suskaičiuotų taškų porų atstumų vidurkis arba suma. Lyginant skirtingų pavyzdžių gautus suminius atstumus galima rasti pavyzdį kurio atstumas mažiausias – tai daugiausia panašumų turintis pavyzdys, tokių atveju laikoma kad pavyzdys atpažintas.

Apibendrinant reikia paminėti, kad kd-tree metodas atlieka erdvės dalinimą – dekompoziciją ir skaičiuoja Euklido atstumus tarp taškų arba taškų požymių tik tam tikrose daugiamatės erdvės ribose. 20 paveiksle pateiktas trimatės erdvės dalinimas, todėl taškų požymiai lyginami ne su kiekvienu požymiu o tik su tam tikros erdvės dalies požymiais. Tokiu būdu tikimybė, kad metodas pateiks klaidingus atitikimus sumažėja. Taigi šiuo metodu dviejuose taškų debesyse surandamos atitinkančių taškų poros. Tačiau, dėl duomenyse esančių triukšmų, kurie gal būt iki iki galo nenufiltruoti, taip pat gali atsirasti klaidingai įvertinti taškų požymių panašumai. 3.7 skyriuje bus aptartas metodas kuris pašalina klaidingai įvertintus taškų porų atitikimus.

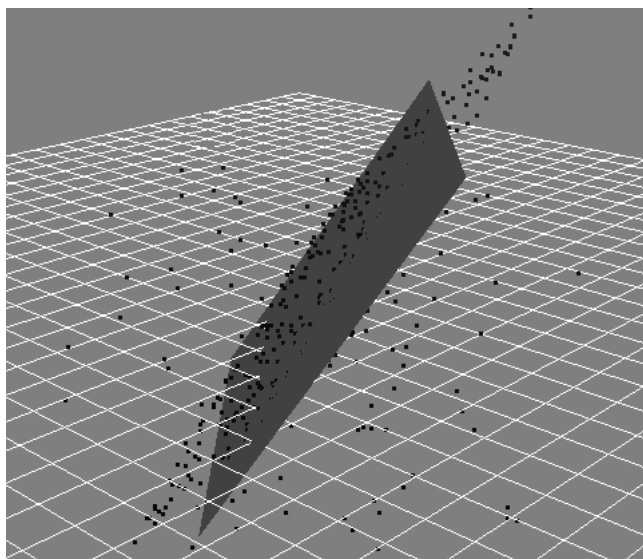
3.7. RANSAC metodas

RANSAC – tai iteracinis metodas paremtas atsitiktinių duomenų išrinkimu, kuriuo galima įvertinti modelio parametrus. Šis metodas stabilus ir yra atsparus triukšmams. RANSAC taikomas, kai reikia apdoroti duomenis, kuriuose reikia įvertinti parametrus modelio. Modelis turi tenkinti

įėjimo duomenis. Minėtus įeinančius duomenis galima padalinti į dvi dalis – patenkančius (angl. *inlier*) į modelį ir nepatenkančius (angl. *outlier*) į modelį. Nepatenkantys į modelį laikomi laikomi triukšmais. RANSAC algoritmą sudaro [34]:

1. Įėjimo duomenys X
2. Funkcija M , kuri paskaičiuoja modelio P parametrus θ , pagal duomenų rinkinį, kurį sudaro n taškų.
3. Įvertinimo funkcija E , kuri įvertina modelį sudarančių taškų atitikimą
4. Funkcijai E nustatomas slenkstis t
5. Iteracijų kiekis k

Taigi, visą metodas sudarytas iš vieno ciklo. Kiekvieną metodo iteraciją galima padalinti į du etapus. Pirmame etape išrenkami taškai ir suskaičiuojamas modelis. Iš daugelio įėjimo taškų X , atsitiktine tvarka išrenkama n įvairių taškų. Pagal išrinktus taškus n nustatomi parametrai θ , kurie sudaro modelį P . Tai atliekama panaudojus funkcija M . Gauta modelį susitarta vadinti hipoteze. Antrame etape patikrinama gauta hipotezė. Kiekvienam taškui atliekamas atitikimo hipotezei patikrinimas. Tam panaudojama funkcija E ir slenkstis t . Kiekvienas taškas pažymimas patenkančiu arba nepatenkančiu. Po visų taškų patikrinimo, tikrinama ar gauta hipotezė yra geriausia šiam momentui. Jei hipotezė geriausia tai ji pakeičia prieš tai buvusią geriausią hipotezę. Pabaigoje paliekama geriausia hipotezė. Rezultate gaunamas modelis P , jo parametrai θ bei įėjimo duomenys (taškai) pažymėti kaip patenkančius arba nepatenkančius į modelį [34]. 22 paveiksle 3D aplinkoje pateikiamas RANSAC taikomas modelis – plokštuma.



22 pav. RANSAC modelis

RANSAC privalumas toks, kad jis sugeba pateikti patikimą modelio parametrų įvertinimą – modelio parametrai gali būti pateikti dideliu tikslumu net tada kai įėjimo duomenyse daug į modelį

nepatenkančių taškų. Metodo trūkumas pasireiškia dėl metodo laiko ribos, kuri nustatoma iteracijomis. Taigi gali būti nustatytas nepakankamas iteracijų skaičius, to pasekoje ne vienas modelis nebus surastas ir nebus atliekamas atitikimų filtravimas. Iteracijų skaičius gali būti teoriškai paskaičiuotas tačiau dažniausiai jis nustatomas eksperimentiniu būdu. Tam kad įgyvendinti metodą reikia žinoti slenksčio reikšmę – nustatoma eksperimentiniu būdu. Metodo pagalba galima nustatyti tik vieną modelį vienam įėjimo duomenų rinkiniui.

PCL bibliotekoje pagrindiniai RANSAC parametrai yra iteracijų skaičius (*max_iterations*) ir slenkstis (*threshold*) taikomas patenkantiems į modelį taškams. Nepatekę į modelį taškai nenagrinėjami – atmetami. Patekę į modelį taškai toliau vertinami – atliekamas atstumų tarp jų požymių įvertinimas. Mažiausią požymių atstumą, su duomenų bazėje esančių objektu, sudaręs objektas yra laikomas panašiausiu. Šiam objektui priskiriamas atitinkamas pavadinimas iš duomenų bazės – tai laikoma objekto atpažinimu.

3.8. Trumpas skyriaus apibendrinimas

Šiame skyriuje buvo pasirinkti perspektyvūs metodai, kurie bus taikomi 3D objektų atpažinimo metodams sudaryti ir praktiškai įgyvendinti. Kadangi nagrinėjamoje PCL bibliotekoje nėra įgyvendinto ir bendrai priimtino 3D objektų atpažinimo metodo, tai metodai iš kurių bus sudaromi objektų atpažinimo metodai, pasirinkti savo nuožiūra, pagal populiarumą, tyrėjų, kūrėjų pasiūlymus, bei pagal dažną minėjamą literatūros šaltiniuose.

Išanalizavus 3D požymių deskriptorių FPFH, PRH ir PFHRGB veikimą nustatyta, kad šie deskriptoriai veikia labai panašiai (tai matyti ir iš pavadinimų). PFH yra pirmasis ir pagrindinis deskriptorius iš kurio atsirado kiti du FPFH ir PFHRGB. Kadangi PFH deskriptorius įvertina tik geometrijos parametrus, sukurtas papildomas deskriptorius – PFHRGB, kuris įvertina ir spalvos ir formos požymius. PFH ir PFHRGB deskriptoriai ilgai skaičiuojami todėl buvo sudarytas dar vienas deskriptorius FPFH, kuris yra paspartintas – optimizuotas PFH deskriptoriaus variantas. Taip pat išnagrinėti SHOT ir CSHOT deskriptoriai. Kaip ir PPH atveju, CSHOT buvo sudarytas norint pagerinti SHOT deskriptoriaus veikimo galimybes – pridėti spalvos įvertinimą. Išsiaiškinta, kad aptarti FPFH, PFH, PFHRGB deskriptoriai paremti histogramomis o SHOT ir COLORSHOT naudoja ir histogramas ir signatūras. Visi išrinkti deskriptorių skaičiavimo metodai gali būti įgyvendinti panaudojus lygiagrečius skaičiavimus, tai sąlygoja greitesnį duomenų apdorojimą.

Taip pat išnagrinėti du būdingųjų taškų išskyrimo metodai „UniformSampling“ ir SIFT. Šie metodai palyginti su kitais, PCL bibliotekos metodais yra išskirtiniai. „UniformSampling“ metodas taiko vokselinį tinklą ir jis gali būti naudojamas kaip duomenų tankumo keitimo ir kaip būdingųjų taškų išrinkimo metodas. SIFT metodas gali būti naudojamas kaip būdingųjų taškų išrinkimo

metodas ir kaip deskriptorius. „UniformSampling“ turi vieną aiškų ir svarbiausią parametą – vokselio kraštinės ilgį – l_v . SIFT turi keturis parametrus, kurie praktiškai visi nustatomi eksperimentiniu būdu.

Paskaičiuotų požymių deskriptorių palyginimui pasirinktas ir išanalizuotas kd-tree metodas. Šis metodas naudojamas artimiausių kaimyninių taškų paieškai, atstumo tarp taškų, pagal X, Y, Z koordinates, paieškai, taškų požymių palyginimui daugiamatėje erdvėje. Aptarta pagal nutylėjimą, atstumui tarp taškų požymių, skaičiuoti naudojama Euklido metrika (Euklido atstumas). Išnagrinėta, kad kd-tree metodas veikia kaip klasifikatorius, kuris pagal panašius požymius daugiamatėje erdvėje atlieka taškų požymių klasifikavimą. Taip pat nustatyta, kad atliekant požymių palyginimą, palyginimas vykdomas pagal padalintas daugiamatės erdvės dalis – dalinai atsiribojama nuo klaidingų atitikimų.

Galiausiai išanalizuotas klaidingų atitikimų pašalinimo metodas – RANSAC. Šio metodo parametų pasirinkimas panašus į SIFT būdingųjų taškų išskyrimo metodo – eksperimentiniu būdu. Visi šiame skyriuje aptarti perspektyvūs metodai bus panaudoti 3D objektų atpažinimo metodų sudarymui, tyrimo rezultatų aptarimo skyriuje.

4. TYRIMAS IR TYRIMO REZULTATAI

Šiame skyriuje mėginama įsigilinti į atrinktų perspektyvių 3D objektų atpažinimo metodų veikimą ir jų įgyvendinimą. Metodai įgyvendinti C++ programavimo kalba, naudojant *Microsoft Visual C++ 2010 Express* kompiliatorių ir 1.6 versijos PCL biblioteką. Įgyvendinus objektų atpažinimo metodus bus ištirtas jų efektyvumas – objektų atpažinimo tikslumas ir metodų – atskirų dalių vykdymo trukmė. Taip pat bus ištirta, kokia įtaką atpažinimo efektyvumui turi spalvos ir formos požymių skaičiavimo metodai – 3D deskriptoriai, būdingųjų taškų išskyrimo metodai bei jų parametrai. Galiausiai visi gauti tyrimo rezultatai bus apibendrinti aptarus įgyvendintų metodų parametrų vertes bei efektyvumo charakteristikas. Pagal gautus 3D objektų atpažinimo efektyvumo rezultatus bus stengiamasi išrinkti efektyviausią 3D objektų atpažinimo metodą.

Taip pat norima pabrėžti, kad baigiamojo darbo tikslas nėra sudaryti patį efektyviausią atpažinimo metodą, šiame darbe siekiama iš atrinktų perspektyvių metodų sudaryti, įgyvendinti ir išrinkti patį efektyviausią objektų atpažinimo metodą. Eksperimentams pasirinkta jau iš anksto, kitų tyrėjų, paruošta eksperimentinių objektų duomenų bazė. Šiame darbe labiau bus susitelkiama į objektų atpažinimo metodų tyrimą ir todėl atpažinimo metodų tyrime didelę reikšmę turės gera naudojamų duomenų kokybė, kuriai pasiekti reikia papildomų duomenų apdorojimo.

4.1. Eksperimentinių duomenų atranka

Tam, kad atlikti objektų atpažinimo metodų, paremtų 3D spalvos ir formos požymiais tyrimą, reikia atlikti spalvos ir gylio duomenų parengimą. Eksperimentams bus naudojami duomenys iš jau egzistuojančios didelės 3D objektų duomenų bazės, kuri buvo sudaryta kitų naudotojų [43]. Duomenų bazę sudaro 300 objektų, jie suskirstyti į kategorijas, kurių yra 51. Kiekvienoje kategorijoje papildomai yra dar keli objektų pavyzdžių tipai. Duomenų bazė sukurta naudojant jau pirmame skyriuje minėtą jutiklį – „Kinect“. „Kinect“ jutikliu gauti spalvos (RGB) ir gylio vaizdai duomenų bazėje yra sinchronizuoti ir išlyginti. Taigi, objektus sudaro 3D taškų debesys su spalvos ir gylio duomenimis. Kiekvienas objektas, minėtu jutikliu buvo skenuojamas trimis skirtingais aukščiais/pozicijomis (24 pav.) ir skirtingais posūkio kampais. Kiekvieną duomenų bazės objektą sudaro daugiau nei tūkstantis taškų debesų, taigi yra taškų persidengimas bei informacijos perteklius. Visi objektų taškų debesys pateikiami *.pcd formatu. Kiekvieno objekto pavadinimas koduojamas, pavyzdžiui: objekto kategorijos pavadinimas, objekto kategorijos pavyzdys, pasirinktas skenavimo aukštis, vaizdo eilė (*orange_1_1_1*). Kiekvieną objekto vaizdą sudaro didelis kiekis taškų, tarp kurių atstumas – 1 mm. Kadangi duomenų bazėje iš tiesų yra labai daug duomenų, kurie tarpusavyje labai koreliuoti, nuspręsta iš duomenų bazės išsirinkti tik tam tikus objektų vaizdus – trimačius taškų debesis.



Apelsinas



Bananas



Sauso maisto pakuotė



Puodelis



Lėkštė

23 pav. Eksperimentiniai duomenys – objektai ir juos sudarantys taškai

Šiam darbui skirta eksperimentinių duomenų bazė bus sudaryta iš dviejų dalių. Pirmoji skirta metodams ir eksperimentams įgyvendinti, antroji – testavimo duomenys. Kaip minėta objektai skenuoti trimis skirtingomis pozicijomis ir duomenys yra pertekliniai, taigi nutarta pirmajai duomenų bazės daliai panaudoti dviejų skirtingų pozicijų duomenis (a ir c). Testavimo duomenims pasirinkta likusi pozicija (b).



(a) Pirmą poziciją



(b) Antra pozicija



(c) Trečia pozicija

24 pav. Skirtingos jutiklio pozicijos

Objektų duomenys iš duomenų bazės išrinkti taip, kad parengtas duomenų rinkinys būtų skirtas universaliam panaudojimui. Iš didelės duomenų bazės atrinkti objektai, kurie skiriasi savo forma ir spalva bei taip pat yra panašūs pagal minėtus požymius. Kadangi duomenų bazėje yra duomenų perteklius o įrenginių skaičiavimo spartos stinga, bus pasirinkti tik keli pagrindiniai objekto vaizdai. Iš kasdieninės žmogaus gyvenimo aplinkos, savo nuožiūra, pasirinktos 5 objektų kategorijos: apelsinas, bananas, lėkštė, sauso maisto pakuotė, puodelis. 23 paveiksle pateikiami pasirinktų objektų kategorijų vaizdai.

4.1 lentelė. Eksperimentinių duomenų bazė

Parametras	Eksperimentavimo duomenys	Testavimo duomenys
Kategorijų skaičius	5	5
Kategorijos pavyzdžių skaičius	2	2
Skenavimo pozicijų skaičius	2	1
Objekto vaizdų skaičius	5	5
Iš viso vaizdų	100	50

Visą eksperimentinių duomenų bazę sudarys 150 objektų vaizdų. Objektų atpažinimo metodų patikros metu, kiekvienas testavimo duomenų bazės vaizdas bus lyginamas su kiekvienu eksperimentams skirtu vaizdu. Taigi, iš viso bus atlikta 5000 vaizdų palyginimų. 4.1 lentelėje pateikiami pagrindiniai sudarytos duomenų bazės parametrai ir ją sudaromų elementų kiekis. Tam, kad objektų atpažinimas būtų efektyvesnis ir artimesnis realių, aukštesnio lygio užduočių sprendimui, eksperimentams atrinkti duomenys – vaizdai (taškų debesys) sudaryti taip, kad testavimui ir bandymams nebūtų panaudojami tie patys objektų pavyzdžiai.

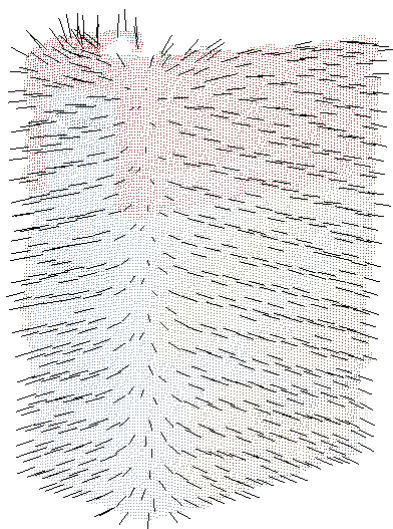
4.2. Metodų parametrų nustatymas

Kiekvienos atpažinimo etapo dalies metodai turi savo parametrus, nuo kurių priklauso 3D objektų atpažinimo efektyvumas. Taigi, šiame darbe bus bandoma keisti įvairius atpažinimo metoduose taikomų metodų parametrus tam, kad nustatyti efektyviausią atpažinimo metodą ir jo parametrus. Visų pirma reikia paminėti parametrus, kuriuos galima iš anksto nustatyti pagal sprendžiamą užduotį ir turimus duomenis. Taip pat, tam kad nešališkai įvertinti pasirinktų metodų efektyvumą, pasirinkti metodų parametrai turi vienodai sąlygoti kitų atpažinime naudojamų metodų veikimą.

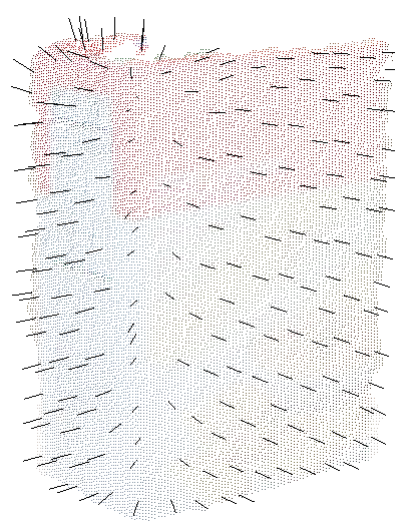
Efektyvus objektų atpažinimas priklauso ne tik nuo pasirinktų metodų bet ir nuo tinkamai pasirinktų ir įvertintų eksperimentavimo duomenų. Taigi pirmame objektų atpažinimo etape, reikia

įvertinti duomenų parametrus. Kadangi duomenų tankumas sąlygoja atpažinimo rezultatus ir skaičiavimo laiką, reikia savo nuožiūra pasirinkti, objektų atpažinime, naudojamų duomenų tankumą. Duomenų bazėje saugomi 3D objektai, kurių taškų debesis yra vienodo tankumo, tačiau skirtingo dydžio. Objektų atpažinimo metodų tyrimui bus panaudoti du skirtingi duomenų tankumai. Duomenų tankumas bus keičiamas naudojant vokselinį tinklą, taip pat tai bus vienas iš įvardintų būdingųjų taškų išrinkimo metodų. Duomenų – taškų tankumas mažinamas vokseliniu tinklu, kurio kiekvieno vokselio kraštinės ilgis – 1 cm, kitu atveju – 2 cm. Taškų tankumas visų palyginamų debesų turi būti panašus, nes atpažinime taikomi paviršiaus vertinimo metodai reikalauja duomenų – taškų aplink skaičiuojamą tašką. Jei taškų nerandama arba jų trūksta, tai metodas neveikia. Pagal turimus duomenis reikia nustatyti naudojamų duomenų tankumą. Jei naudojami būdingųjų taškų išskyrimo metodai, tai būdingųjų taškų paieškos metodų parametrus reikia parinkti taip, kad tolimesniuose žingsniuose naudojamiems metodams netrūktų duomenų. Taip pat reikia atsižvelgti į visus naudojamus duomenis, jų tankumą ir kiekį kadangi duomenų trūkumas gali atsirasti apdorojant mažesnę objektą, kai duomenų apdorojimo parametrai nustatyti pagal daugiau duomenų turintį objektą. Tai svarbu todėl, kad automatiškai naudojami metodai negali įvertinti duomenų kiekio, tankumo ir patys pakeisti parametrus į tinkamesnius.

Kaip jau minėta normalės yra svarbi paviršiaus savybė, tad jų tinkamas įvertinimas yra būtinas. Pagal turimų duomenų tankumą galima eksperimentiniu būdu nustatyti normalių parametrus – plotelio, kuriame ieškomi artimiausių kaimynų taškai, apskritimo spindulį. Kadangi naudojamų duomenų tankumas visoje duomenų bazėje yra vienodas, normalių skaičiavimo parametrų derinimas atliekamas pagal pasirinktą taškų debesį. Svarbu paminėti, kad požymių skaičiavimo metodai – deskriptoriai efektyviai veikia tik tada, kai tinkamai paskaičiuojamos normalės.



(a) Apskritimo spindulys 2 cm

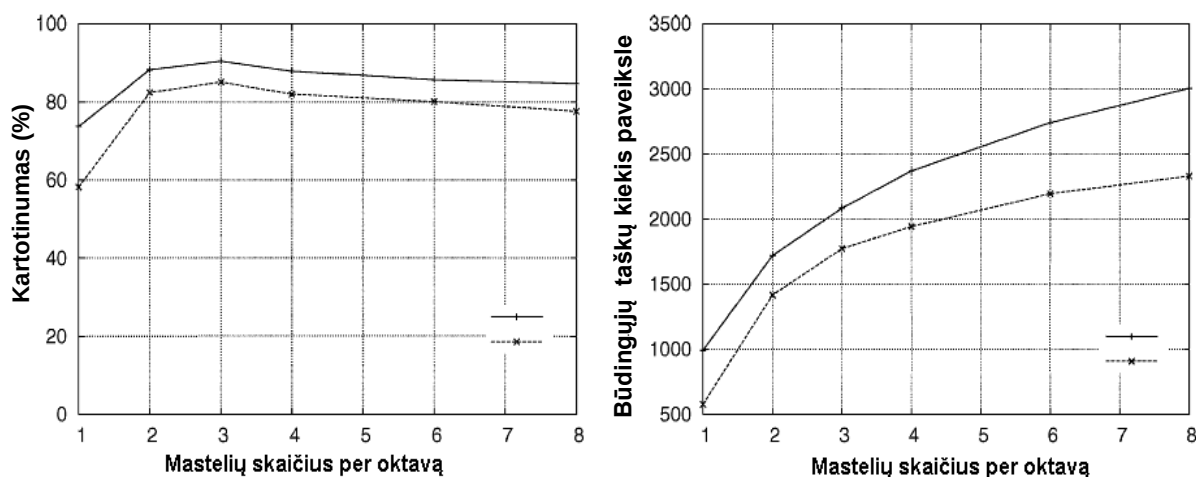


(b) Apskritimo spindulys 3 cm

25 pav. Normalių skaičiavimo parametrų nustatymas (objektas – sauso maisto pakuotė)

Normalių skaičiavimas visada turi būti paremtas turimų duomenų tankumo įvertinimu, todėl po kiekvieno duomenų tankumo pakitimo taip pat reikia keisti ir normalių skaičiavimo parametrus. Šiuo atveju normalės skaičiuojamos panaudojant kaimyninius taškus, kurie patenka į aplink tašką, apibrėžtą apskritimą. Svarbu pabrėžti, kad skaičiuojant normales apskritimo spindulio dydis priklauso nuo turimų duomenų tankumo. Normalių skaičiavimo metu, apskritimo spindulys turi būti parinktas taip, kad būtų surasti ir įvertinti kaimyniniai taškai. Atitinkamai, pagal pasirinktus du skirtingus duomenų tankumus, nustatomi tinkami normalių skaičiavimo apskritimo spinduliai – 2 cm ir 3 cm (25 pav.). Normalių skaičiavimo laikas priklauso nuo taškų kiekio, kuriuose skaičiuojamos normales ir nuo to kokio dydžio yra aplinkinių taškų paieškos laukas – apskritimo spindulys. Kuo spindulys didesnis tuo paieškos laikas ilgesnis, tačiau šiame darbe taikomų duomenų kiekis nėra didelis, todėl normalių skaičiavimo laikas bus neįvertinamas o atpažinimo metodų vykdymo trukmės skaičiavimuose į tai nebus atsižvelgta.

Prieš deskriptorių skaičiavimą reikia išskirti būdinguosius taškus, kuriuose bus skaičiuojami deskriptoriai. Darbe panaudojami du būdingųjų taškų išskyrimo metodai: vokselinis tinklelis ir SIFT metodas. Reikia priminti, kad vokselinis tinklelis gali būti naudojamas kaip duomenų tankumo mažinimo metodas ir kaip būdingųjų taškų išskyrimo metodas. Pastarojo metodo parametrai jau buvo aptarti (1 cm ir 2 cm). Tam tikri SIFT metodo pagrindiniai parametrai pasirenkami pagal 26 paveiksle pateikiamas metodo charakteristikas. SIFT būdingųjų taškų išskyrimo metodas taip pat pasižymi tuo, kad jo parametrai dažnai nustatomi eksperimentiniu būdu.



26 pav. Mastelių skaičiaus oktavoje pasirinkimas [26]

Pagal 26 paveiksle pateiktą grafiką pasirenkama koks mastelių skaičius bus oktavoje. Kartotinumai tai savybė, kai metodas gali surasti požymį kitoje scenoje, kuri nedaug yra pasikeitusi (pasisukusi). Pasirenkamas mastelių skaičius, kuris geriausiai apibrėžia kartotinumą – 3. Iš 24 paveikslo matyti, kad norint padidinti išskiriamų būdingųjų taškų kiekį, reikia didinti mastelių skaičių, kurio kartotinumai tokių būdų nežymiai mažėja. Oktavų skaičius, pasirinktas pagal PCL

bibliotekoje pateiktą pavyzdį – 10. Mažiausia kontrasto reikšmė taip pat pasirinkta tokia, kokia pateikiama PCL pavyzdyje – 0,5. Minimalus mastelis paliktas kaip PCL pavyzdyje – 0,002.

Kitame žingsnyje nustatomi 3D deskriptorių parametrai. Viena iš svarbiausių deskriptorių skaičiavimo dalių – teisingai paskaičiuotos normalės. Kiekvienas deskriptorius PCL bibliotekoje turi tuos pačius pagrindinius parametrus, tai taško kuriame skaičiuojamas deskriptorius aplinka – R_D . Iš šios aplinkos, deskriptorių skaičiavimui, naudojamos tašką supančios normalės. Kaip ir normalių skaičiavimo atveju, čia taip pat pasirinktas apskritimo apibrėžimas apie skaičiuojamą tašką. Apskritimo spindulys – R_D taip pat turi būti parinktas toks, kad metodas galėtų surasti ir įvertinti aplinkui esančias normales. Deskriptorių aprašymuose minima sąlyga kurios reikia laikytis – deskriptoriaus apskritimo skaičiavimo spindulys R_D turi būti didesnis už normalių apskritimo spindulį. Tačiau kiek didesnis turi būti spindulys nėra minima, todėl bus atlikti keli bandymai su skirtingais apskritimų spinduliais. Pirmam duomenų tankumui apskritimo spinduliai – 4 cm ir 5 cm, antram – 5 cm ir 6 cm. Nuo šio apskritimo spindulio dydžio taip pat priklauso deskriptorių skaičiavimo laikas – kuo didesnis spindulys tuo ilgiau atliekami skaičiavimai (4.5 skyrius).

4.2 lentelė. Pasirinktų metodų parametrų sąrašas

Parametras	Reikšmė
Taškų tankumas / „UniformSampling“ metodo vokselio kraštinės ilgis, l_V	1 cm 2 cm
SIFT metodo minimalus mastelis	0,002
SIFT metodo oktavų skaičius	10
SIFT metodo mastelių skaičius oktavoje	3
Artimiausių kaimyninių taškų paieškos, apskritimo spindulys, R_D	Kai $l_V = 1$ cm, $R_D = 4$ cm; $R_D = 6$ cm; Kai $l_V = 2$ cm; $R_D = 5$ cm; $R_D = 6$ cm;
RANSAC metodo slenkstis – atstumas tarp atitikimų, S_R	Kai $l_V = 1$ cm; $S_R = 0,5$ cm; $S_R = 2$ cm; $S_R = 5$ cm; Kai $l_V = 2$ cm; $S_R = 0,3$ cm; $S_R = 1$ cm; $S_R = 3$ cm;
RANSAC iteracijų skaičius	1000

Taigi, 4.2 lentelėje pateikiami visi parametrai, kurie bus panaudoti objektų atpažinimo metodų efektyvumo tyrimams. Atlikus deskriptorių skaičiavimus, sekančiame žingsnyje atliekami požymių palyginimai. Atliekant požymių palyginimą surandami panašūs – atitinkantys požymių taškai ir atstumai tarp jų. Tačiau ne visada nustatyti požymių panašumai būna teisingi. Norint pagerinti atpažinimo rezultatus bus pritaikytas klaidingų atitikimų pašalinimo metodas – RANSAC. Šio metodo parametrai pasirinkti atlikus eksperimentinius bandymus, kurie aprašyti 4.4 skyriuje.

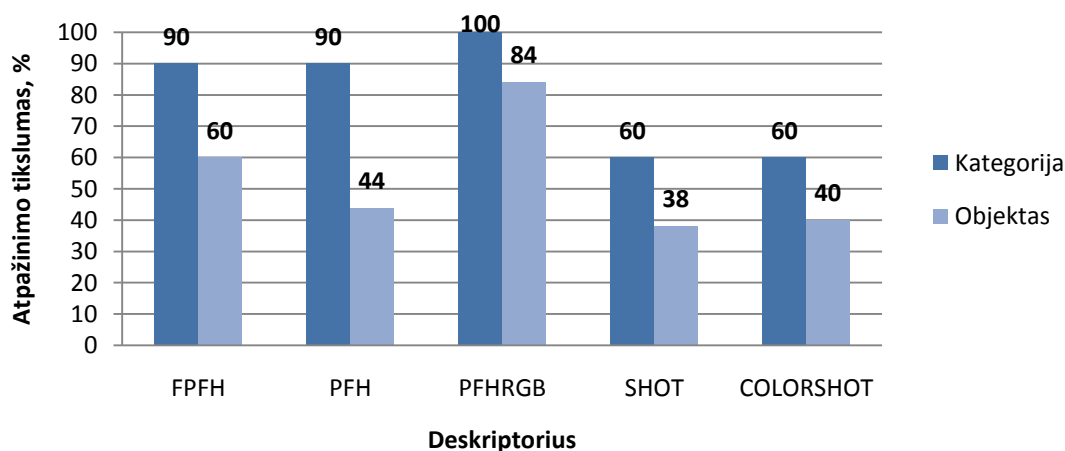
4.3. 3D Objektų atpažinimo metodų tikslumas

Atpažinimo metodų tikslumas dažniausiai yra pats svarbiausias objektų atpažinimo kriterijus. Kadangi darbe nagrinėjami spalvos ir formos požymiais grįsti atpažinimo metodai, tai požymių skaičiavimo metodai – deskriptoriai bus laikomi atraminiais metodais. Šiame skyriuje bus ištirti metoduose naudojami parametrai, atpažinimo metodų deriniai, pateiktos objektų atpažinimo tikslumo priklausomybės nuo pasirinktų metodų ir metodų parametrų reikšmių.

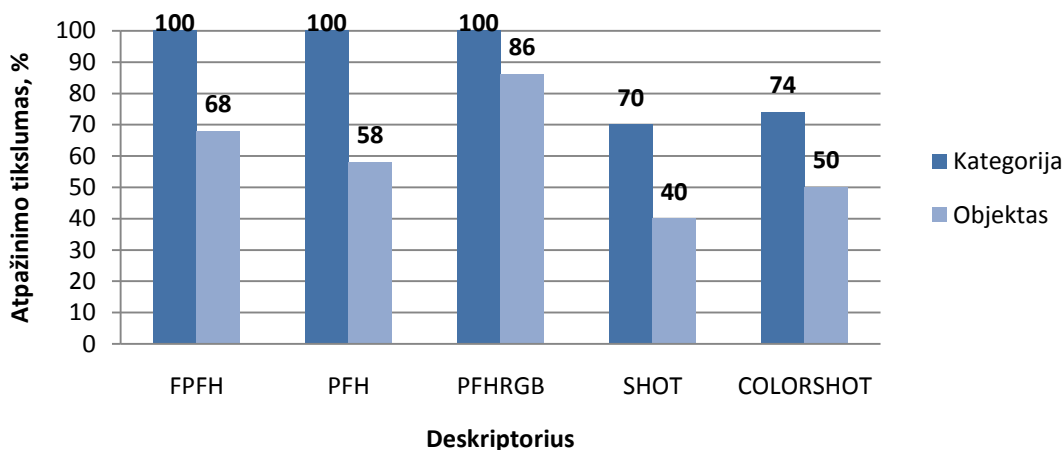
Tyrimas būs atliktas keičiant deskriptoriaus parametą – apskritimo spindulio dydį. Šiame apskritime atliekama artimiausių kaimyninių taškų paieška. Šio skyriaus grafikuose pateikiami dviejų skirtingų R_D parametro, verčių taikymo rezultatai, naudojant skirtingus būdingųjų taškų išskyrimo metodus bei jų parametrus.

Programiškai įgyvendinant FPFH, PFH, PFHRGB, SHOT bei COLORSHOT deskriptorius nustatyta, kad SHOT ir COLORSHOT deskriptoriai yra jautresni artimiausių taškų paieškos parametrai R_D . Nustačius per mažą R_D parametro reikšmę SHOT ir COLORSHOT deskriptorių metodai pateikė pranešimus dėl LRF klaidos ir neįgyvendino požymių skaičiavimo. Kaip jau minėta norint, kad visi skaičiavimai būtų atliekami lygiomis metodams sąlygomis, pasirinktas ir 4.2 lentelėje pateiktas toks minimalus apskritimo spindulys, kurio dydis tenkintų visus objektų atpažinime taikomus metodus.

Visų pirma atlikti objektų atpažinimo bandymai, kuriuose nepanaudojamas klaidingų atitikimų pašalinimo metodas – RANSAC. Tokiu būdu bandoma nustatyti kaip tiksliai veikia požymių palyginimo metodas – kd-tree bei norima nustatyti R_D parametrų reikšmes, su kuriomis bus gauti stabiliausi taškų požymiai – stabiliausi deskriptoriai. 27 ir 28 paveiksluose pateikti atpažinimo tikslumo rezultatai taikant būdingųjų taškų išrinkimo metodą „UniformSampling“, kurio vokselio kraštinė $l_V = 1$ cm.



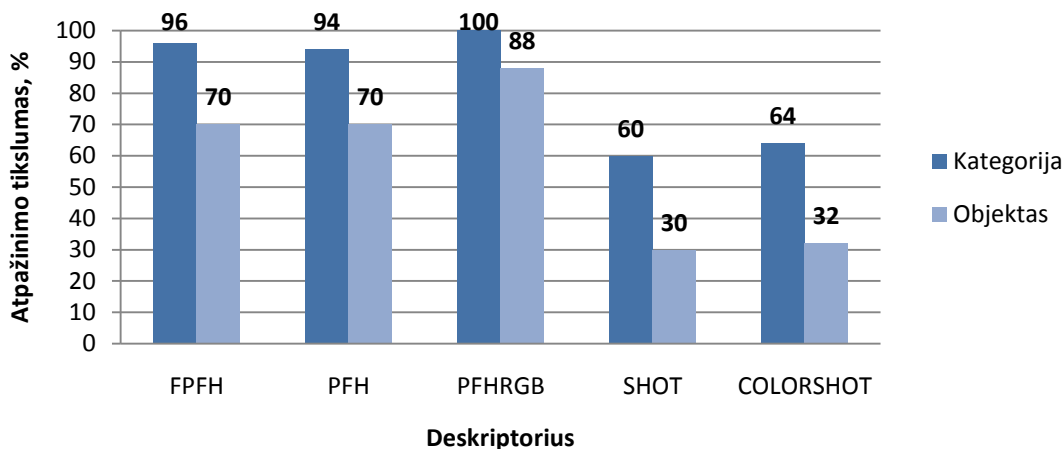
27 pav. Metodų atpažinimo tikslumas, kai $l_V = 1$ cm, $R_D = 4$ cm



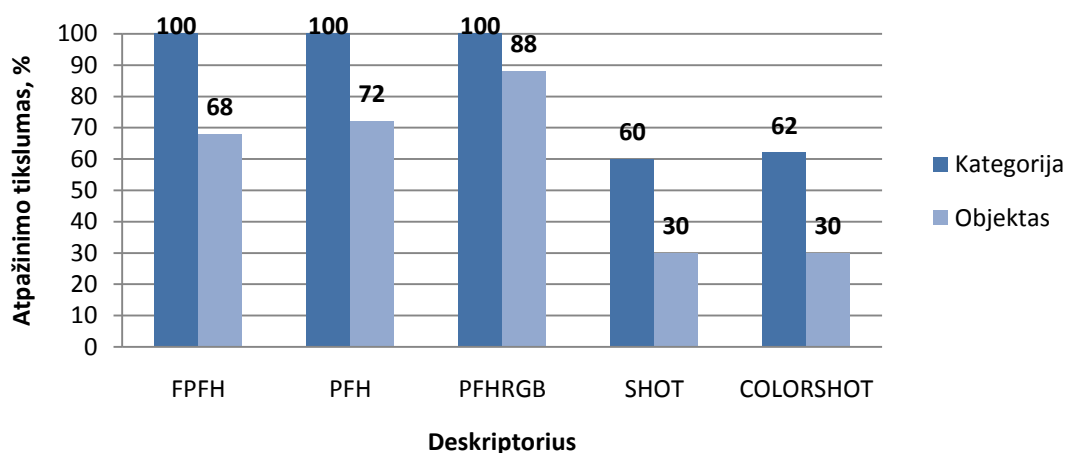
28 pav. Metodų atpažinimo tikslumas, $l_V = 1$ cm, $R_D = 6$ cm

Gauti rezultatai rodo, kad padidinus požymio skaičiavimo spindulį R_D , atpažinimo tikslumas padidėjo taikant bet kurį iš deskriptorių. Reikia išskirti COLORSHOT deskriptorių, kadangi taikant šį deskriptorių kategorijos atpažinimo tikslumas, priešingai nei kitų metodų, sumažėjo tačiau padidėjo objekto atpažinimo tikslumas. Iš 28 paveikslo matyti kad didžiausias atpažinimo tikslumo pokytis į gerąją pusę įvyko taikant PFH formos deskriptorių. 27 paveiksle matyti, kad taikant spalvos ir formos požymių deskriptorių – PFHRGB pavyko pasiekti 100 % objekto kategorijos atpažinimą. 27 ir 28 paveikslai parodo, kad padidinus R_D parametą nuo 4 cm iki 6 cm, pavyko pasiekti 100 % kategorijos atpažinimą taikant ir kitus požymių deskriptorius – FPFH, PFH.

29 ir 30 paveiksluose pateikti atpažinimo tikslumo rezultatai taikant būdingųjų taškų išrinkimo metodą „UniformSampling“, kurio vokselio kraštinė $l_V = 2$ cm. Tokiu būdu sumažinamas taškų tankumas ir kiekis. Kaip ir tuo atveju, kai vokselio kraštinė buvo prilyginta 1 cm, taip pat ir šiuo atveju tyrinėjamas metodų objektų atpažinimo tikslumas, dviem skirtingais apskritimo spindulio – R_D parametrais.



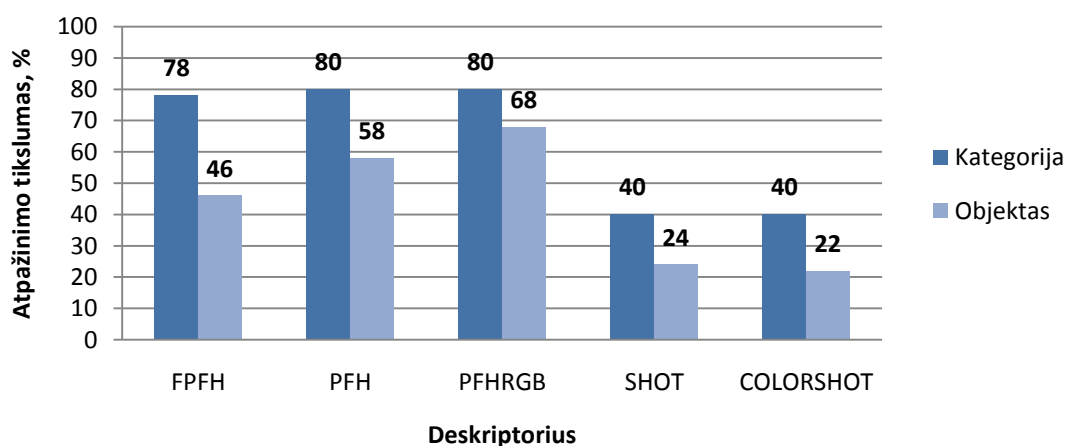
29 pav. Metodų atpažinimo tikslumas, kai $l_V = 2$ cm, $R_D = 5$ cm



30 pav. Metodų atpažinimo tikslumas, $l_V = 2$ cm, $R_D = 6$ cm

Būdingųjų taškų išrinkimo metodu, sumažinus taškų tankumą, iš 29 ir 30 paveiksluose pateiktų rezultatų matyti, kad šiuo atveju taip pat padidinus apskritimo spindulį R_D atpažinimo tikslumas padidėjo, tačiau taikant tik kai kuriuos deskriptorius. Objekto kategorijos atpažinimo tikslumas padidėjo taikant FPFH ir PFH deskriptorius. Šiuo atveju spalvos informacija nesuteikė papildomo atpažinimo tikslumo. Objekto atpažinimas padidėjo taikant tik PFH metodą. Taikant PFHRGB ir SHOT metodus atpažinimo tikslumas nepakito. COLORSHOT metodo taikymas, abiem atvejais – kategorijos ir objekto atpažinimas, tik sumažėjo. Taip pat kaip ir didesniai taškų tankumui esant ($l_V = 1$), šiuo atveju ($l_V = 2$) padidinus parametą R_D , pasiektas 100 % kategorijos atpažinimo tikslumas taikant ir kitus – FPFH, PFH deskriptorius. Dėl per didelio apskritimo spindulio nukenčia atpažinimo tikslumas, kadangi įvertinami per dideli kitų taškų požymių indeliai t.y. gaunama per maža požymių – deskriptorių diskriminantinė galia. Taip pat dėl padidinto apskritimo spindulio padidėja deskriptoriaus skaičiavimo o taip pat ir atpažinimo metodo vykdymo laikas.

Sekantis būdingųjų taškų išskyrimo metodas – SIFT buvo taikomas sumažinus duomenų tankumą, panaudojant vokselinį tinklą, kurio vokselio kraštinės ilgis – 1 cm.



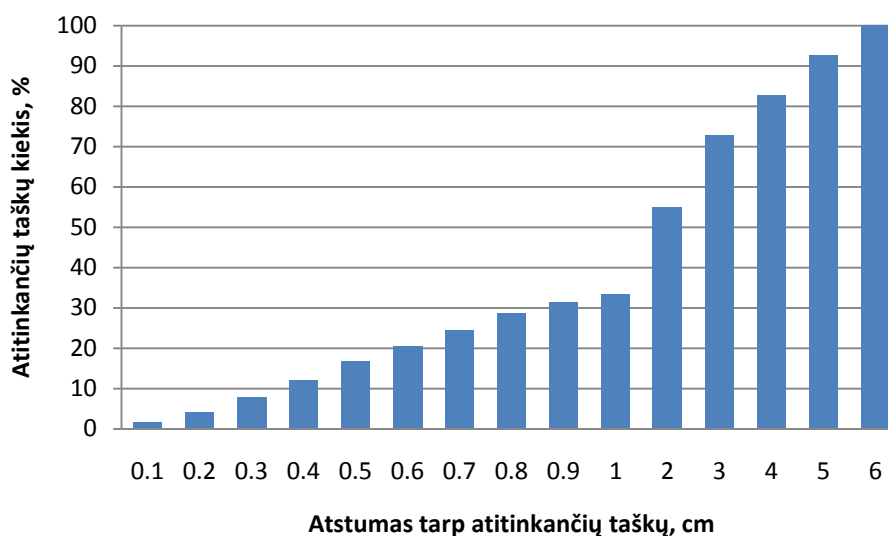
31 pav. Metodų atpažinimo tikslumas, kai $l_V = 1$ cm, $R_D = 4$ cm

Iš 31 paveiksle pateiktų rezultatų matome, kad naudojant SIFT metodą geriausias atpažinimo tikslumas gaunamas naudojant spalvos ir formos požymių deskriptorių PFHRGB. Priešingai nei taikant „UniformSampling“ būdingųjų taškų išrinkimo metodą, nėra gautas maksimalus – 100 % objekto kategorijos atpažinimas. Taikant SIFT būdingųjų taškų išrinkimo metodą gauti rezultatai yra prastesni nei naudojant „UniformSampling“ metodą, taikant bet kurį iš požymių deskriptorių. Taip pat reikia pabrėžti, kad taikant SIFT metodą gauti prasčiausi iš visų bandymų rezultatai.

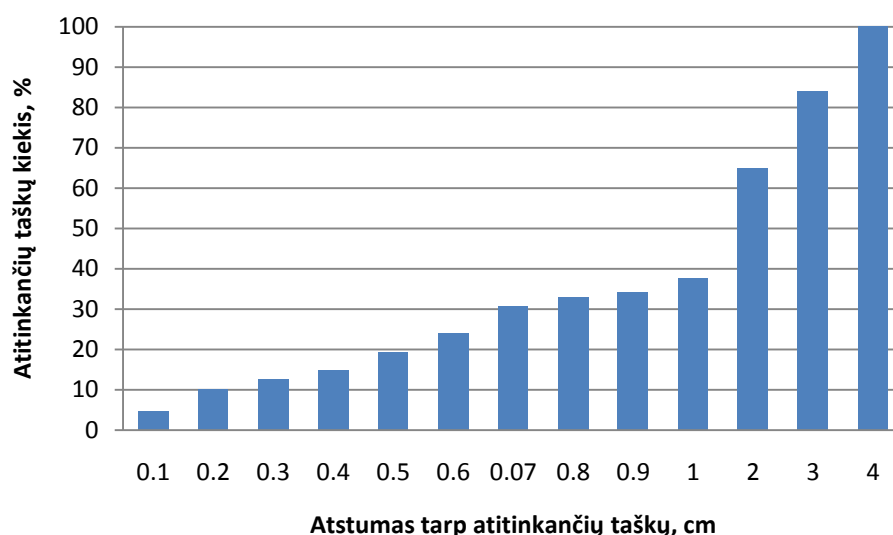
Apibendrinant rezultatus galima teigti, kad lyginant 3D objektų atpažinimą, naudojant skirtingo tankumo duomenis, tiksliau atpažinti 3D objektus pavyksta, kai naudojamas mažesnis duomenų tankumas. Lyginant būdingųjų taškų išskyrimo metodų poveikį atpažinimo rezultatams matyti, kad taikant „UniformSampling“ metodą gauti geresni atpažinimo rezultatai nei taikant SIFT metodą. Padidinus, kaimyninių taškų paieškos, apskritimo spindulį gaunami geresni objekto kategorijos atpažinimo rezultatai. Taikant „UniformSampling“ metodą buvo gautas 100 % objektų kategorijos atpažinimo tikslumas nepriklausomai nuo duomenų tankumo. Reikia pabrėžti, kad geriausių atpažinimo rezultatus visais atvejais parodė metodas, kuriame naudojamas spalvos ir formos požymių deskriptorius. Iš gautų rezultatų matyti, kad objektų atpažinimo tikslumas gali būti dar gerinamas, tam bus pritaikytas RANSAC klaidingų atitikimų pašalinimo metodas.

4.4. RANSAC metodo įtaka objektų atpažinimo tikslumui

Visų pirma prieš taikant RANSAC metodą buvo pasirinkti pagrindiniai RANSAC metodo parametrai. RANSAC slenkstis – S_R , tai didžiausias atstumas tarp atitinkančių taškų nustatytas pagal turimus duomenis. Eksperimentavimo būdu, pagal turimų apdorojamų duomenų parametrus nustatyta RANSAC metodo atitinkančių taškų kiekio priklausomybė nuo atstumo tarp atitinkančių taškų.



32 pav. RANSAC metodo atitinkančių taškų priklausomybė, $V_K = 1$ cm



33 pav. RANSAC metodo atitinkančių taškų priklausomybė, $V_K = 2$ cm

Taikant „UniformSampling“ būdingųjų taškų išskyrimo metodą gautos priklausomybės dviems taškų tankumams, kai vokselio kraštinės ilgis $l_V = 1$ cm ir $l_V = 2$ cm. Visų pirma nustatytas atitinkančių taškų diapazonas. Eksperimentavimo būdu buvo stebėta kokiam atstumui esant tarp atitinkančių taškų, kokią dalį atitinkančių taškų RANSAC metodas pašalina. 32 paveiksle pateiktas nustatytas atitinkančių taškų diapazonas nuo 0,1 cm iki 6 cm. Šis diapazonas mažesniau duomenų tankumui yra mažesnis (0,1–4 cm). Pastebėta, kad nustatytas RANSAC metodo atitinkančių taškų atstumas gali tiesiškai – pagal procentus filtruoti atitinkančius taškus. Ši priklausomybė bus pritaikyta tolesniam atpažinimo tikslumo tyrimui. Taip pat eksperimentiniu būdu nustatyta, kad metodo įgyvendinimui užtenka 1000 iteracijų. Pagal nutylėjiną PCL bibliotekoje taip pat yra nustatytos 1000 iteracijos.

Kadangi nėra iš anksto žinoma kaip RANSAC metodo pašalinami atitinkantys taškai įtakoja atpažinimo tikslumą, be eksperimentinių bandymų nėra galimybės nustatyti kokia dalį taškų pašalinus bus pagerintas atpažinimo tikslumas. Jei pašalinama daug atitinkančių taškų, gali būti kad pašalinami teisingi atitikimai ir jei pašalinama mažai atitikimų, gali būti kad nėra pašalinami klaidingi atitikimai. Taigi, tolesniems atpažinimo efektyvumo tyrimams bus pasirinkti keli atstumai tarp atitinkančių taškų. Pagal gautas priklausomybes (32 pav., 33 pav.) parenkami atstumai tarp atitinkančių taškų, skirtingiems duomenų tankumams: kai $l_V = 1$ cm; $S_R = 0,5$ cm; $S_R = 2$ cm; $S_R = 5$ cm; ir kai $l_V = 2$ cm; $S_R = 0,3$ cm; $S_R = 1$ cm; $S_R = 3$ cm.

4.3 lentelėje pateikiami rezultatai, kai RANSAC metodas pritaikytas atpažinimo metodams, kurie tiksliausiai atpažįsta objektus, kadangi norima pagerinti gautus, tinkamus objektų atpažinimui, atpažinimo rezultatus. Pritaikius „UniformSampling“ metodą su mažiausiu vokselio kraštinės ilgiu – l_V , iš lentelėje pateiktų rezultatų matyti, kad taikant FPFH, PFH ir SHOT t.y. tik formos deskriptorius, pavyko pagerinti objektų atpažinimo metodų tikslumą keliais procentais.

4.3 lentelė. Atpažinimo metodų tikslumas, taikomas „UniformSampling“, $l_V = 2$ cm, $R_D = 6$ cm

Deskriptorius	$S_R = 0,3$ cm		$S_R = 1$ cm		$S_R = 3$ cm	
	Kategorija	Objektas	Kategorija	Objektas	Kategorija	Objektas
FPFH	100 %	70 %	88 %	48 %	88 %	58 %
PFH	100 %	78 %	100 %	58 %	90 %	68 %
PFHRGB	90 %	80 %	100 %	70 %	80 %	60 %
SHOT	62 %	34 %	62 %	28 %	28 %	22 %
COLORSHOT	42 %	28 %	82 %	58 %	22 %	14 %

Tačiau taikant PFHRGB ir COLORSHOT objektų kategorijos ir objekto atpažinimas pablogėjo. Kadangi tai spalvos ir formos deskriptoriai tai reiškia, kad spalvos informacija buvo netinkamai interpretuota ir netgi trukdė. Geriausi gauti rezultatai, netaikant RANSAC, buvo gauti naudojant PFGRBG deskriptorių, šiuo atvejų naudojant šį deskriptorių ir taikant RANSAC, gauti prastesni objekto kategorijos ir objekto atpažinimo rezultatai. Pastebėta, kad naudojamiems duomenų – objektų rinkiniams, geriausi atpažinimo rezultatai gauti, kai pasirenkamas mažiausias atstumas tarp atitikimų – S_R . Tai reiškia, kad kuo daugiau taškų buvo pašalinta tuo tikslesni gauti atpažinimo rezultatai. Tačiau koks be būtų mažas atitinkančių taškų skaičius, vis tiek nepavyko pagerinti geriausių gautų rezultatų kai nebuvo nenaudojamas RANSAC.

4.4 lentelė. Atpažinimo metodų tikslumas, taikomas SIFT, $l_V = 1$ cm, $R_D = 4$ cm

Deskriptorius	$S_R = 0,5$ cm		$S_R = 2$ cm		$S_R = 5$ cm	
	Kategorija	Objektas	Kategorija	Objektas	Kategorija	Objektas
FPFH	88 %	78 %	88 %	58 %	78 %	78 %
PFH	88 %	48 %	88 %	78 %	88 %	48 %
PFHRGB	78 %	64 %	92 %	72 %	64 %	48 %
SHOT	58 %	38 %	58 %	48 %	58 %	48 %
COLORSHOT	68 %	38 %	38 %	28 %	68 %	38 %

Sekantis metodas buvo tirtas kuriame taikomas kitas būdingųjų taškų išrinkimo metodas – SIFT. Iš 4.4 lentelėje pateiktų rezultatų matyti, kad taikant bet kurį deskriptorių gaunami geresni kategorijos ir objekto atpažinimo rezultatai negu, kai nėra taikomas RANSAC metodas. Priešingai nei taikant „UniformSampling“ metodą, spalvos ir formos požymių deskriptorių taikymas yra naudingas – teisingai įvertinta spalvos informacija ir padidina atpažinimo tikslumą. Geriausi atpažinimo rezultatai gaunami taikant FPFH, PFH ir PFHRGB deskriptorius. Nors daugeliu atveju

RANSAC metodas pagerino atpažinimo tikslumą, naudojant SIFT, tačiau nė vienu atveju nėra gautas maksimalus objekto kategorijos ar objekto atpažinimo tikslumas. Taip pat nėra pagerintas, prieš tai užfiksuotas geriausias, objekto kategorijos ir objekto atpažinimo tikslumas.

Apibendrinant reikia paminėti, kad siekiant pagerinti atpažinimo tikslumo rezultatus buvo ištirtas klaidingų sutapimų pašalinimo metodas RANSAC. Pritaikius RANSAC metodą, gauti rezultatai rodo, kad RANSAC metodas yra tinkamas geresniems rezultatams pasiekti, tada kai taikomas SIFT būdingųjų taškų išrinkimo metodas. Kai taikomas „UniformSampling“ metodas, tada RANSAC nėra tinkamas gerinti atpažinimo rezultatus, kadangi tuo atveju klaidingai įvertinama spalvos informacija. Taigi, kadangi tiksliausiu išrinktas atpažinimo metodas, kuris naudoja „UniformSampling“ metodą, tai šiuo atveju RANSAC metodas nėra geriausias sprendimas gerinti objektų atpažinimo tikslumą, metodas ypatingai nėra naudingas kai objektų atpažinimui naudojami spalvos ir formos požymiai. Ištyrus RANSAC metodo įtaką objektų atpažinimui paaiškėjo, kad nenaudojant šio metodo gaunami geresni atpažinimo rezultatai o tai reiškia, kad metodas nepataiso klaidingų atitikimų. Požymių palyginimui naudojamas kd-tree metodas taip pat turi minimalų klaidingų atitikimų pašalinimą. RANSAC taikymas neparodė geresniu tikslumo rezultatu, todėl norint padidinti atpažinimo tikslumą reikia keisti kitus atpažinimo metodo vykdymo etapus (dalis) ar dalių parametrus, kadangi ne bepavyksta atlikti klaidingų atitikimų pašalinimo – gaunama išvada, kad jų nėra. Todėl sprendimo reikia ieškoti ankstesniuose, atpažinimo metodų, vykdymo žingsniuose.

4.5. Metodų veikimo sparta

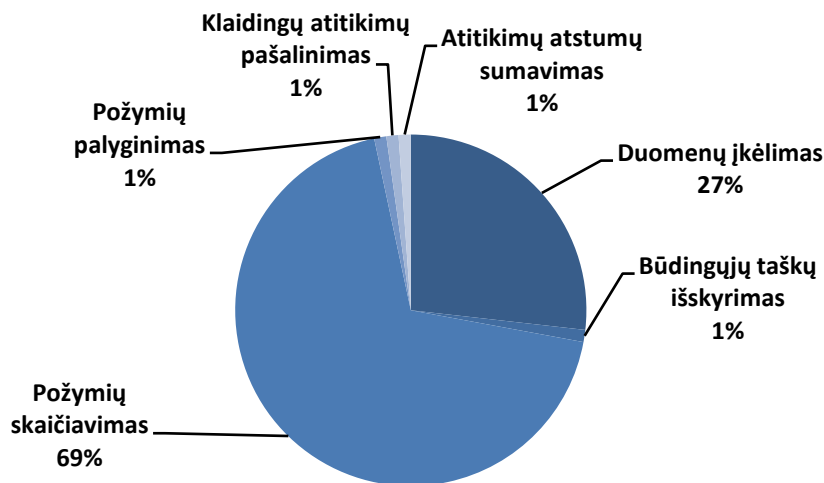
Objektų atpažinimo sparta ir atpažinimo metodų vykdymo sparta dažnai svarbi tada kai norima įgyvendinti realaus laiko sistemą arba kai atpažinimo tikslumas turi mažesnę vertę nei atpažinimo sparta. 4.2 skyriuje buvo minėta, kad dėl per didelio R_D parametro padidėja 3D deskriptoriaus skaičiavimo laikas o taip pat ir atpažinimo metodo vykdymo laikas.

4.5 lentelė. Deskriptorių skaičiavimo laikas

Deskriptorius	Laikas, s; $l_V = 1$ cm		Laikas, s; $l_V = 2$ cm	
	$R_P = 4$ cm	$R_P = 6$ cm	$R_P = 5$ cm	$R_P = 6$ cm
FPFH	1	1	1	1
PFH	12	46	2	3
PFHRGB	22	87	1	1
SHOT	2	3	1	1
COLORSHOT	4	5	1	1

4.5 lentelje pateiktas deskriptorių skaičiavimo laikas (sekundėmis), kai naudojamos dvi skirtingos, būdingųjų taškų išskyrimo metodo „UniformSampling“, vokselio kraštinės reikšmės – l_v . Padidinus apskritimo spindulį R_D , padidėja požymio paieškos plotas – padaugėja skaičiavimų, kadangi didesniame paieškos plote yra daugiau taškų, kurie panaudojami požymio skaičiavimui. Kai apdorojami tankesni taškų debesys, spindulio padidinimas gali kelis kartus pailginti kai kurių deskriptorių skaičiavimo laiką. PFH ir PFHRGH deskriptorių skaičiavimo laikas padidėja net keturis kartus (5.4 lentelė.). Atitinkamai, pagal R_D parametro dydį, palyginus 4.5 lentelėje pateiktus deskriptorių skaičiavimo laikus ir objektų atpažinimo tikslumą (4.3 skyrius) matyti, kad požymių paieškos spindulio padidinimas nėra efektyvus laiko atžvilgiu, kadangi padidinus spindulį, objektų atpažinimo tikslumas kinta ir pagerėja nežymiai.

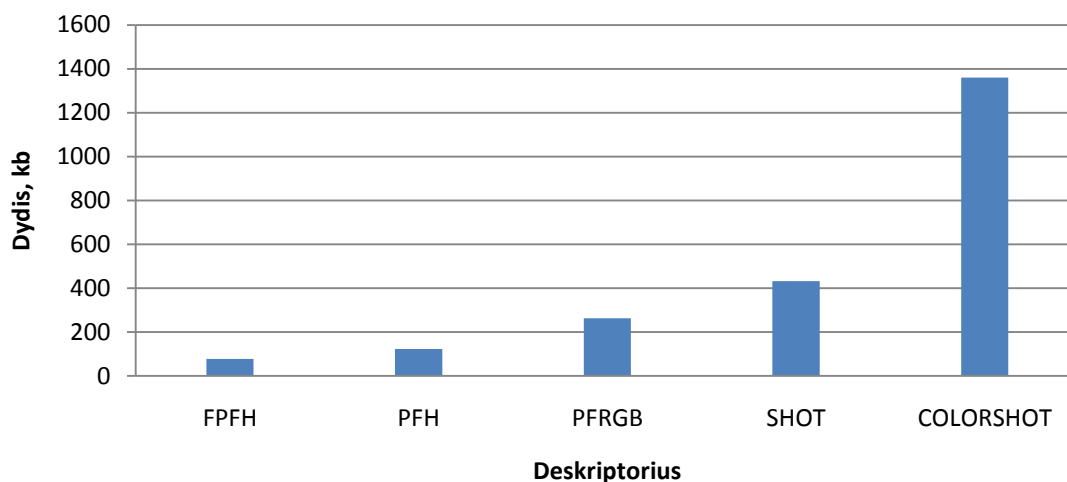
Įvertinus objektų atpažinimo metodų spartą bei metodus sudarančių dalių spartą, nustatyta kad daugiausia laiko užima požymių skaičiavimas. Požymių skaičiavimas yra viena iš svarbiausių objektų atpažinimo dalių. Kitos metodų dalys vykdomos kur kas trumpiau, tai parodo 34 paveiksle pateikta diagrama. Diagramoje pateikiama objektų atpažinimo metodų vykdymo trukmė, čia pateikiamas penkių požymių deskriptorių skaičiavimo vidurkis o kitų dalių, nepriklausomai nuo deskriptoriaus tipo, vykdomo laikas nekinta. Vykdomo trukmei apskaičiuoti, savo nuožiūra, buvo pasirinkti du objektai. Objektų (dviejų duomenų failų) įkėlimas truko 6 sekundes. Duomenų failų dydis 434 kb ir 432 kb, atitinkamai taškų skaičius 7799 ir 7774. Būdingųjų taškų išskyrimo metodas išskyrė 300 taškų, kuriuose paskaičiuoti deskriptoriai.



34 pav. Objektų atpažinimo, atskirų etapų, vykdymo trukmė

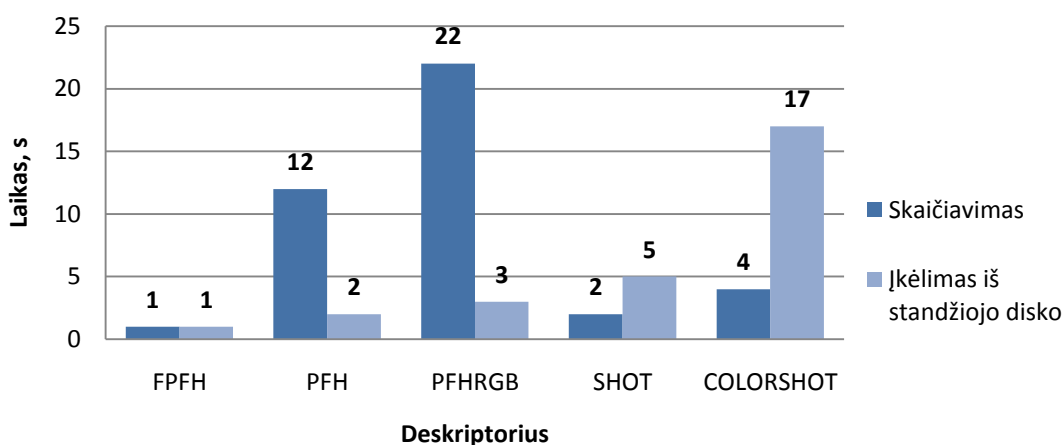
Jeigu, kiekvieną kartą paleidžiant sistemą, reikėtų skaičiuoti požymius, tai tokia sistema būtų neefektyvi veikimo trukmės požiūriu. Šią problemą galima išspręsti išsaugojant suskaičiuotus objekto požymius į kompiuterio standųjį diską. Išsaugotų požymių įkėlimas iš standžiojo disko yra spartesnis nei požymių skaičiavimas (36 pav.).

Kiekvienas deskriptorius turi savo unikalią signatūrą, kuri yra ilgesnė ar trumpesnė priklausomai nuo to koks skaičiavimo metodas naudojamas ir kiek skaičiavimo metodas naudoja saugyklų histogramai sudaryti. Pastebėta kad, kuo ilgesnė deskriptoriaus signatūra tuo daugiau vietos standžiajame diske užima išsaugotas požymių deskriptorius (35 pav.), tačiau deskriptoriaus signatūros ilgis nenusprendžia požymio skaičiavimo trukmės (36 pav.).



35 pav. Išsaugoto į standųjį diską deskriptoriaus dydis

Normalių ir deskriptorių skaičiavimo laikas tiesiogiai priklauso nuo duomenų kiekio – taškų skaičiaus, kuriuose skaičiuojami atitinkami parametrai, bei nuo to, kokio dydžio yra aplinkinių taškų/normalių paieškos laukas – apskritimo spindulys R_D . Atlikus tyrimus pastebėta, kad paskaičiuotų požymių įkėlimas iš kompiuterio standžiojo disko yra daug kartų greitesnis nei jų skaičiavimas. Taigi, suskaičiuoti taškų požymiai gali būti išsaugoti kompiuterio standžiajame diske ir vėliau atliekant daug pakartotinių bandymų, požymių skaičiavimas gali būti pakeistas jų įkėlimu.



36 pav. Požymių skaičiavimo ir požymių įkėlimo iš standžiojo disko laikas

Gauti rezultatai (36 pav.) parodo, kad požymių įkėlimas iš standžiojo disko, gali net kelis kartus paspartinti kai kurių požymių panaudojimo laiką o spartus požymių panaudojimas yra labai svarbus realaus laiko sistemose. PFH ir PFHRGB deskriptoriai skaičiuojami žymiai ilgiau nei jų, prieš tai paskaičiuotos, reikšmės būtų įkeliamos iš standžiojo disko. Greitis PFH deskriptoriaus skaičiavimo variantas – FPFH iš tiesu skaičiuojamas žymiai greičiau nei jo pirmtakas ir šis deskriptorius skaičiuojamas greičiausiai iš visų nagrinėtų šiame darbe. SHOT ir COLORSHOT deskriptoriai priešingai nei PFH ir PFRGB, skaičiuojami žymiai greičiau negu jų paskaičiuotos reikšmės būtų įkeliamos. Taip yra todėl, kad SHOT ir COLORSHOT deskriptorių histogramos yra ilgesnės nei FPFH, PFH ir PFHRGB deskriptorių (2.3 lentelė) ir jos užima daugiau vietos standžiajame diske. Taigi SHOT ir SHOTCOLOR deskriptorius, dėl jų skaičiavimo spartos, efektyviau yra kiekvieną kartą skaičiuoti o PFH ir PFHRGB deskriptorius efektyviau yra įkelti iš standžiojo disko.

4.6. Trumpas skyriaus apibendrinimas

Šiame skyriuje buvo įgyvendinti bei ištirti 3D objektų atpažinimo metodai, kurie pagrįsti spalvos ir formos požymių rinkiniais. Buvo ištirta, kokią įtaką, objektų kategorijos ir objekto atpažinimui, turi būdingųjų taškų išskyrimo, požymių skaičiavimo, bei klaidingų požymių atitikimų pašalinimo metodai bei jų parametrai. Nustatyti įgyvendintų atpažinimo metodų efektyvumo parametrai – atpažinimo tikslumas ir vykdymo trukmė.

Nustatyta, kad taikant „UniformSampling“ būdingųjų taškų išskyrimo metodą, RANSAC – klaidingų atitikimų pašalinimo metodas nėra veiksmingas. Metodas netgi yra nuostolingas kadangi jo taikymas nėra veiksmingas, net nuostolingas, naudojant 3D spalvos ir formos požymių deskriptorius. Gautas atpažinimo tikslumas gali būti sėkmingai gerinamas naudojant SIFT būdingųjų taškų išskyrimo metodą. Priešingai nei „UniformSampling“, SIFT metodas pagerina atpažinimo rezultatus taikant visus, net 3D spalvos ir formos deskriptorius. Nepaisant SIFT metodo galimybių, sėkmingai taikyti metodą kartu su visais deskriptoriais, objektų atpažinimo metodas su SIFT nusileidžia atpažinimo tikslumu metodui, kuriame taikomas „UniformSampling“ metodas. Naudojant „UniformSampling“ metodą su mažesniu vokselio kraštinės ilgiu – l_v ir naudojant PFHRGB metodą su didesniu aplinkinių taškų paieškos lauku R_D , gaunamas 3D objektų atpažinimo metodas, kuris tiksliausiai sugeba atpažinti objekto kategoriją – 100 % ir objektą – 88 %. Šis metodas labiausiai tinkamas atpažinti objektus pagal spalvos ir formos požymius. Verta pabrėžti, kad RANSAC metodas, tiksliausio metodo įgyvendinime, nedalyvauja. Tiksliausiais metodas dažnai nebūna sparčiausias, šiuo atveju taip ir yra. Įgyvendintas, tiksliausias atpažinimo metodas be to yra ir pats lėčiausias. Šis metodas gali būti taikomas sistemose kur reikalingas didelis tikslumas ir

visai nesvarbus metodo vykdymo laikas. Verta paminėti metodą, kuriame vietoje PFHRGB panaudojamas FPFH deskriptorius, sudarytas objektų atpažinimo metodas gali taip pat pateikti aukštą atpažinimo tikslumą: kategorijos atpažinimas – 100 %, objekto atpažinimas – 70 %. Šio metodo laikinės ir atpažinimo tikslumo charakteristikos pagal santyki yra geriausios. Toks metodas universalus kadangi gali būti taikomas ne tik realaus laiko sistemose bet ir sistemose, kur reikalingas tikslus atpažinimas. Tačiau, kadangi įvertinami tik formos požymiai, šio metodo taikymo sritis vėl susiaurėja. Kiti metodai, kuriuose taikomi SHOT ir COLORSHOT deskriptoriai, pagal laikines charakteristikas, galėtų būti taikomi realaus laiko sistemose, tačiau jų gautas atpažinimo tikslumas iš tiesu yra mažas. Praktiškai įgyvendinti 3D objektų atpažinimo metodai sugeba atpažinti 3D objektus ir gali būti taikomi 3D objektų atpažinimo užduotims spręsti.

5. APIBENDRINIMAS. IŠVADOS

Šiame magistro darbe buvo atlikta trimačių (3D) taškų debesų apdorojimui skirtų programinių priemonių – metodų apžvalga. Apžvelgtos gylgio duomenų – 3D taškų debesims, apdoroti skirtos ir labiausiai šiuo metu programuotojų plėtojamos, PCL (angl. *Point Cloud Library*) bibliotekos galimybės. Apžvelgti ir išanalizuoti bibliotekoje, 3D objektams atpažinti, taikomi metodai, kurie pagrįsti 3D spalvos ir formos požymių rinkiniais. Nustatyta, kad daugelis metodų kilo iš dvimačių vaizdų apdorojimo srities. Visi požymių deskriptoriai įvertina paviršiaus geometrijos – formos požymius ir tik du iš 13-os deskriptorių įvertina geometrijos ir tekstūros – spalvos požymius.

Pasirinkti perspektyvūs metodai praktiškai įgyvendinti ir yra tinkami naudoti 3D objektų atpažinimui. Iš praktiškai įgyvendintų metodų sudaryti objektų atpažinimo metodai, grįsti 3D spalvos ir formos požymių rinkiniais, kurie yra tinkami 3D objektų atpažinimo užduotims spręsti. Atlikus eksperimentinius tyrimus, pateikti objektų atpažinimo metodų, efektyvumo rezultatai. Taip pat aptartos atskiros objektų atpažinimo metodų dalys, parametrų reikšmės ir jų įtaka objektų atpažinimo efektyvumui. Išrinktas objektų atpažinimo metodas – metodų rinkinys ir jo metodų parametrų vertės, kurioms esant gaunami geriausi objektų atpažinimo rezultatai.

1. Atlikta analitinė objektų atpažinimo metodų, grįstų spalvos ir formos požymių rinkiniais, apžvalga. Objektų atpažinimo metodai, grįsti spalvos ir formos požymių rinkiniais, yra tinkami atpažinti 3D objektus ir yra tinkami 3D objektų atpažinimo užduotims spręsti.
2. Objektų atpažinimas praktiškai pritaikytas ir naudojamas įvairiose gyvenimo srityse, pvz.: robotų rega, kompiuterinė rega, apsaugos sistemos, slaugos sistemos.
3. Iš viešai prieinamos objektų duomenų bazės atrinkti eksperimentiniai duomenys ir sudaryta objektų atpažinimo eksperimentams skirta duomenų bazė. Atrinkti duomenys tinkami objektų atpažinimo efektyvumui tirti, naudojant metodus, grįstus 3D spalvos ir formos požymių rinkiniais.
4. Ištirti atrinktieji objektų atpažinimo metodai, kurie grįsti spalvos ir formos požymių rinkiniais. Pateikti objektų atpažinimo efektyvumo rezultatai – objektų atpažinimo tikslumas ir pagrindinių metodų vykdymo trukmė. Atlikti eksperimentai, kurių metu nustatytas efektyviausias objektų atpažinimo metodas (sudarančios dalys): „UniformSampling“ būdingųjų taškų išskyrimo metodas, PFHRGB (angl. *Point Feature Histograms + RGB*) – 3D spalvos ir formos požymių deskriptorius. Pasiiektas objekto kategorijos atpažinimo tikslumas – 100 % ir objekto atpažinimo tikslumas – 88 %.
5. Objektų atpažinimo metode ilgiausiai atliekami deskriptorių skaičiavimai.
6. Geriausias 3D objektų atpažinimo tikslumas gautas naudojant 3D spalvos ir formos požymių rinkinius.

7. Tiksliausias objektų atpažinimo metodas vykdomas ilgiausiai, todėl nėra tinkamas naudoti realaus laiko sistemose.
8. Realaus laiko sistemose labiausiai tinkamas naudoti sparčiausias, formos požymių, deskriptorius – FPFH (angl. *Fast Point Feature Histograms*).
9. Atlikti eksperimentai įrodo pasirinkto, geriausių rezultatus pateikusių („UniformSampling“ būdingųjų taškų išskyrimo metodas, PFHRGB (angl. *Point Feature Histograms + RGB*) – spalvos ir formos požymių deskriptorius), objektų atpažinimo metodo ir jo parametrų tinkamumą 3D objektams atpažinti.
10. Klaidingų atitikimų pašalinimo metodo – RANSAC (angl. *RANdom SAmple Consensus*) taikymas kartu su geriausių sudarytu objektų atpažinimo metodu, kuris naudoja spalvos ir formos požymių rinkinius ir „UniformSampling“ – būdingųjų taškų išrinkimo metodą, nepasiteisino – RANSAC metodas nepagerina atpažinimo tikslumo.
11. Norint pagerinti 3D objektų atpažinimo metodų tikslumą, viena iš galimybių yra ieškoti geresnių, 3D spalvos ir formos požymių, įvertinimo metodų.

Literatūra

1. RUSU R. B.; and COUSINS S. 3D is here: Point Cloud Library (PCL), *ICRA Communications*, Willow Garage, 68 Willow Rd., Menlo Park, CA 94025, USA, 2011.
2. RUSU R. B. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments, *Dissertation*, Institut für Informatik der Technischen Universität München, 2009.
3. *Normal (geometry)* [interaktyvus]. Wikipedia [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Normal_%28geometry%29>
4. *Estimating Surface Normals in a PointCloud* [interaktyvus]. PCL [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: <http://pointclouds.org/documentation/tutorials/normal_estimation.php>
5. BERTHOLD K. P. HOM, Extended Gaussian Images, *PROCEEDINGS OF THE IEEE*, VOL. 72, NO. 12, DECEMBER 1984.
6. *Module segmentation* [interaktyvus]. PCL [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: <http://docs.pointclouds.org/trunk/group_keypoints.html>
7. WANG P., „GrabCut“ – Interactive Foreground Extraction, Project report for CS766, Department of Biomedical Engineering, University of Wisconsin – Madison.
8. ROTHER C., KOLMOGOROV V., BLAKE A., „GrabCut“ – Interactive Foreground Extraction using Iterated graph cuts, *ACM Transactions on Graphics (TOG)*, Microsoft Research Cambridge, UK, 2004.
9. *Detectors evaluation: repeatability and time performances* [interaktyvus]. 2012. Point Google Summer Of Code [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: <<http://www.pointclouds.org/blog/gsoc12/gballin/all.php>>
10. SIPIRAN I., BUSTOS B., Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes, PRISMA Research Group, Department of Computer Science, University of Chile, Santiago, Chile, 2011.
11. *Module keypoints* [interaktyvus]. PCL [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: <http://docs.pointclouds.org/trunk/group_keypoints.html>
12. ZHONG Y., Intrinsic shape signatures: A shape descriptor for 3D object recognition, *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, vol., no., pp.689-696, Sept. 27 2009-Oct. 4 2009.
13. STEDER B., RUSU R. B., KONOLIGE K., and BURGARD W. Point Feature Extraction on 3D Range Scans Taking into Account Object Boundaries In *Proc. of the IEEE Int. Conf. on Robotics &Automation (ICRA)*. 2011.
14. LOWE D. G., Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, 60, 2 (2004), pp. 91-110.
15. *Downsampling a PointCloud using a VoxelGrid filter* [interaktyvus]. Point Cloud Library (PCL) Documentation [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: <http://www.pointclouds.org/documentation/tutorials/voxel_grid.php#voxelgrid>
16. *Darboux frame* [interaktyvus]. Wikipedia [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą:

- < http://en.wikipedia.org/wiki/Darboux_frame >
17. RUSU R., BLODOW N., MARTON Z., and BEETZ M., Aligning point cloud views using persistent feature histograms, in *International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, September 22-26 2008.
 18. RUSU R. B., BLODOW N., and BEETZ M., Fast Point Feature Histograms (FPFH) for 3D Registration, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17 2009.
 19. *Local reference frame* [interaktyvus]. Wikipedia [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Local_reference_frame>
 20. *Inertial frame of reference* [interaktyvus]. Wikipedia [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Local_reference_frame>
 21. RUSU R. B., BRADSKI G., R. THIBAU, and HSU J., Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram, in *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 18-22 2010.
 22. *Module features* [interaktyvus]. PCL [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: < http://docs.pointclouds.org/trunk/group_features.html >
 23. TOMBARI F., SALTI S., 3D descriptors [interaktyvus]. 2012. 3D Computer Vision, Computer Vision Laboratory [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: <<http://www.t3lab.it/wp-content/uploads/download/20120530/20120530-Lesson2A.pdf>>
 24. FROME A., HUBER D., KOLLURI R., BULOW T., and MALIK J., Recognizing objects in range data using regional point descriptors, in *8th European Conference on Computer Vision*, 2004, pp. 224–237.
 25. TOMBARI F., SALTI S., STEFANO L. D., Unique Shape Context for 3D data description, International Workshop on 3D Object Retrieval (3DOR 10) - in conjunction with ACM Multimedia 2010.
 26. LOWE D. G., Distinctive image features from scaleinvariant keypoints, *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, November 2004.
 27. LAZEBNIK S., SCHMID C., and PONCE J., A sparse texture representation using local affine regions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1265–1278, 2005.
 28. DROST B., ULRICH M., NAVAB N., and S. ILIC, Model globally, match locally: Efficient and robust 3d object recognition, in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2010, San Francisco, CA, USA, 2010, pp. 998–1005.
 29. OSADA R., T. FUNKHOUSER A., CHAZELLE B., and DOBKIN D. P., Matching 3D models with shape distributions, in *International Conference on Shape Modeling and Applications*, Princeton University, Princeton, NJ 08540, USA, Genoa, Italy, 2001, pp. 154–166.
 30. IP C. Y., LAPADAT D., SIEGER L., and REGLI W. C., Using shape distributions to compare solid models, in *Symposium on Solid Modeling and Applications*, 2002, pp. 273–280.

31. F. TOMBARI, S. SALT, STEFANO L. DI, Unique Signatures of Histograms for Local Surface Description, *11th European Conference on Computer Vision (ECCV)*, September 5-11, Hersonissos, Greece, 2010.
32. TOMBARI F., SALT S., STEFANO L. DI, A combined texture-shape descriptor for enhanced 3D feature matching, *IEEE International Conference on Image Processing (ICIP)*, September 11-14, Brussels, Belgium, 2011.
33. *The PCL Registration API* [interaktyvus]. PCL [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: < http://www.pointclouds.org/documentation/tutorials/registration_api.php#registration-api >
34. RANSAC [interaktyvus]. Wikipedia [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: < <http://ru.wikipedia.org/wiki/RANSAC> >
35. SMITH L. I. A tutorial on principal components analysis. Technical report, Cornell University, USA, February 26 2002.
36. PETRELLI and STEFANO L. DI, On the repeatability of the local reference frame for partial shape matching, CVLab – DEIS, University of Bologna Viale Risorgimento, 2 - 40135 Bologna, Italy. 2011.
37. TOMBARI F., How does a good feature look like? [interaktyvus]. 2013 PCL TUTORIAL @ICRA'13 [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: <http://www.pointclouds.org/assets/icra2013/pcl_features_icra13.pdf>
38. *Volumetric Picture Element* [interaktyvus]. Wikipedia [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: <<http://en.wikipedia.org/wiki/Voxel>>
39. LAINE, SAMULI; Tero Karras (February 2010). Efficient Sparse Voxel Octrees – Analysis, Extensions, and Implementation. NVIDIA Corporation. Retrieved June 11, 2010.
40. FLANN – *Fast Library for Approximate Nearest Neighbors* [interaktyvus]. FLANN [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: <<http://people.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN>>
41. *k-d tree* [interaktyvus]. Wikipedia [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: < https://en.wikipedia.org/wiki/K-d_tree >
42. *Norm (mathematics)* [interaktyvus]. Wikipedia [žiūrėta 2013 m. gegužės 7 d.]. Prieiga per internetą: < http://en.wikipedia.org/wiki/Norm_%28mathematics%29#Euclidean_norm >
43. LAI K., BO L., REN X., and FOX D., A Large-Scale Hierarchical Multi-View RGB-D Object Dataset, In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

A priedas. Pranešimo 16-oje Lietuvos jaunųjų mokslininkų konferencijoje medžiaga

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
ELEKTRONINIŲ SISTEMŲ KATEDRA

Objektų atpažinimo, naudojant spalvos ir formos požymių rinkinius, tyrimas

Parengė: Dmitrijus Skunčikas
Grupė: AKSfm-11

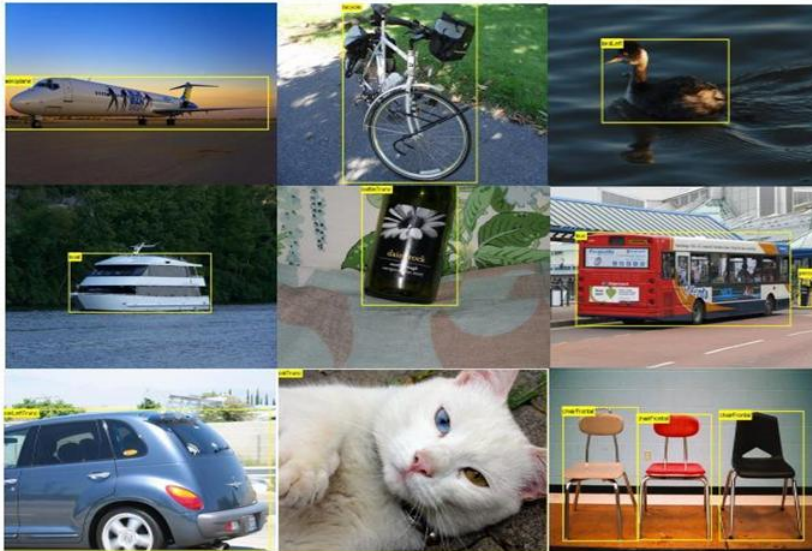
Jaunųjų Mokslininkų Konferencija "Elektronika ir elektrotechnika 2013"

Turinys

- Objektų atpažinimo reikalingumas
- Trimačių taškų debesų technologija
- Objektų atpažinimo problemos
- Požymiai/deskriptoriai objektų atpažinime
- Problemų sprendimo būdų pasiūlymai

2

Objektų atpažinimo reikalingumas



3

Trimačių taškų debesų technologija



4

Objektų atpažinimo problemos

Reikalingi “geri” požymiai/deskriptoriai:

- Aukšta diskriminantinė galia
- Griežta transformacija
- Kintamas pavyzdžių atrinkimo tankumas
- Atsparumas triukšmams

5

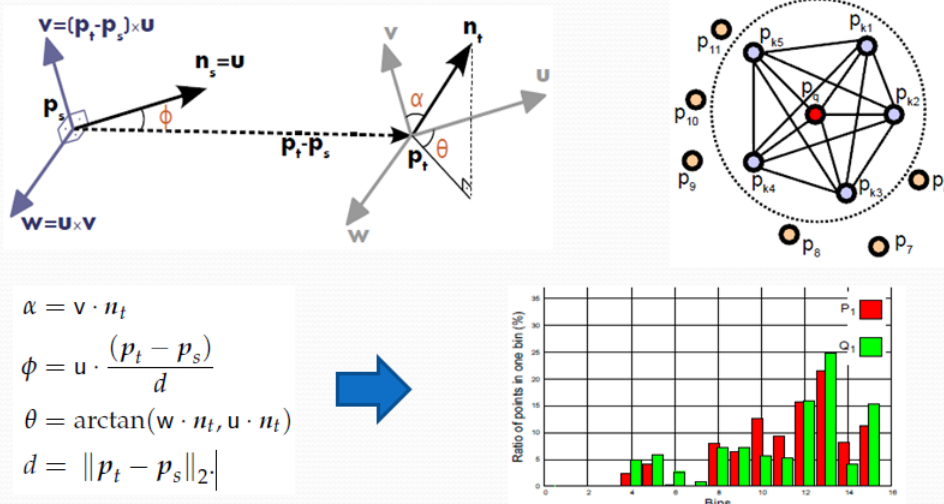
Požymiai/deskriptoriai objektų atpažinime

PCL deskriptorius	
3DSC	PPF
CVFH	RIFT
ESF	SHOT
FPFH	SHOTCOLOR
PFH	USC
PFHRGB	VHF
PCE	

6

PFHRGB deskriptorius

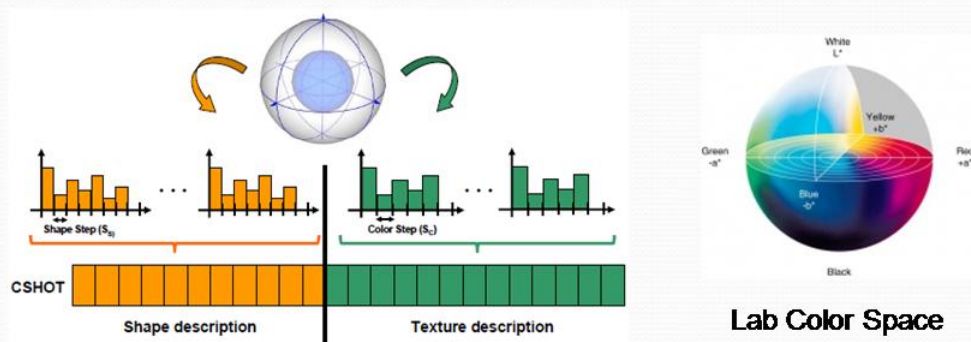
- PFHRGB (Point Feature Histograms + RGB)



7

SHOTCOLOR deskriptorius

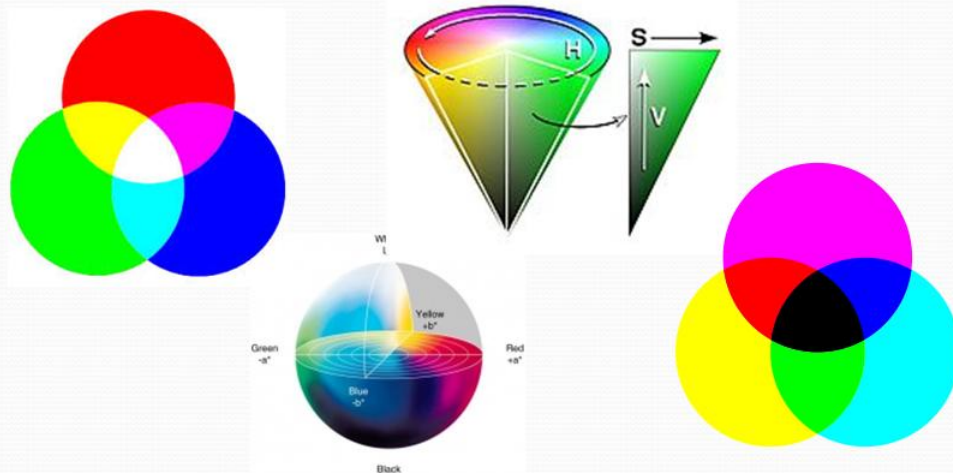
- SHOTCOLOR (Signature of Histograms of Orientations + COLOR)



8

Problemų sprendimo būdų pasiūlymai

- Kitų spalvinių sistemų panaudojimas



9

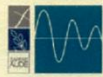
Problemų sprendimo būdų pasiūlymai

- Kitų būdingųjų taškų išskyrimo metodų panaudojimas

PCL būdingųjų taškų išskyrimo metodai	
AbstractAgastDetector	NarfKeypoint
AgastDetector	SIFTKeypoint
OastDetector	SUSANKeypoint
HarrisKeypoint3D	UniformSampling
ISSKeypoint3D	

10

DALYVIO PAŽYMOS KOPIJA



VILNIAUS GEDIMINO
TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS

XVI-toji Lietuvos jaunųjų mokslininkų konferencija
„MOKSLAS – LIETUVOS ATEITIS“
ELEKTRONIKA IR ELEKTROTECHNIKA

PAŽYMA

Dmitrijus Skunčikas

Vaizdų technologijos sekcijoje perskaitė pranešimą
***Objektų atpažinimo naudojant spalvos ir formos
požymių rinkinius tyrimas***

Mokslo komiteto pirmininkas

prof. habil. dr. Romanas Martavičius

Sekcijos pirmininkas

doc. dr. Darius Miniotas

2013 m. kovo 15 d.
Vilnius

B priedas. Įgyvendintų algoritmų pirminis programos tekstas

object_recognition.cpp

```

#include <limits>
#include <fstream>
#include <vector>
#include <iostream>
#include <Eigen/Core>
#include <pcl/io/pcd_io.h>
#include <pcl/point_types.h>
#include <pcl/point_cloud.h>
#include <pcl/filters/voxel_grid.h>
#include <pcl/keypoints/sift_keypoint.h>
#include <pcl/features/normal_3d.h>
#include <pcl/features/fpfh.h>
#include <pcl/features/pfh.h>
#include <pcl/features/pfhrng.h>
#include <pcl/features/shot.h>
#include <pcl/kdtree/kdtree.h>
#include <pcl/kdtree/kdtree_flann.h>
#include <pcl/kdtree/impl/kdtree_flann.hpp>
#include <pcl/registration/correspondence_estimation.h>
#include <pcl/registration/correspondence_rejection_sample_consensus.h>

// Būdingųjų taškų išskyrimo metodo - SIFT panaudojimas
// (true - naudojamas / false - nenaudojamas)
bool keypoints_ = false;

// Duomenų - taškų tankumo keitimo / būdingųjų taškų išskyrimo funkcija
void downsample (pcl::PointCloud<pcl::PointXYZRGBA>::Ptr &points, float
leaf_size,
                pcl::PointCloud<pcl::PointXYZRGBA>::Ptr &downsampled_out)
{
    pcl::VoxelGrid<pcl::PointXYZRGBA> vox_grid;

    // Vokselio kraštinės ilgio nustatymas
    vox_grid.setLeafSize (leaf_size, leaf_size, leaf_size);
    vox_grid.setInputCloud (points);
    vox_grid.filter (*downsampled_out);
}

// Paviršiaus normalių skaičiavimo funkcija
void compute_surface_normals (pcl::PointCloud<pcl::PointXYZRGBA>::Ptr
&points, float normal_radius,
                             pcl::PointCloud<pcl::Normal>::Ptr &normals_out)
{
    pcl::NormalEstimation<pcl::PointXYZRGBA, pcl::Normal> norm_est;

    // Artimiausių kaimyninių taškų paieškos metodo - kd-tree nustatymas
    norm_est.setSearchMethod (pcl::search::KdTree<pcl::PointXYZRGBA>::Ptr (new
pcl::search::KdTree<pcl::PointXYZRGBA>));

    // Artimiausių kaimyninių taškų paieškos apskritimo spindulio RD
nustatymas
    norm_est.setRadiusSearch (normal_radius);

    // Objekto taškų debesies nustatymas
    norm_est.setInputCloud (points);
    norm_est.compute (*normals_out);
}

```

```

// Požymių skaičiavimo funkcija
void compute_features (pcl::PointCloud<pcl::PointXYZRGBA>::Ptr &points,
                      pcl::PointCloud<pcl::Normal>::Ptr &normals,
                      float feature_radius,
                      pcl::PointCloud<pcl::FPFHSignature33>::Ptr
&descriptors_out)
{
    // Deskriptoriaus objekto sukūrimas
    pcl::FPFHEstimation<pcl::PointXYZRGBA, pcl::Normal, pcl::FPFHSignature33>
est;

    // Artimiausių kaimyninių taškų paieškos metodo - kd-tree nustatymas
    est.setSearchMethod (pcl::search::KdTree<pcl::PointXYZRGBA>::Ptr (new
pcl::search::KdTree<pcl::PointXYZRGBA>));

    // Artimiausių kaimyninių taškų paieškos apskritimo spindulio RD
nustatymas
    est.setRadiusSearch (feature_radius);

    // Objekto taškų debesies nustatymas
    est.setInputCloud (points);
    est.setInputNormals (normals);
    est.compute (*descriptors_out);
}

// Būdingųjų taškų išskyrimo funkcija
void detect_keypoints (pcl::PointCloud<pcl::PointXYZRGBA>::Ptr &points,
                      float min_scale, int nr_octaves, int nr_scales_per_octave,
float min_contrast,
                      pcl::PointCloud<pcl::PointWithScale>::Ptr &keypoints_out)
{
    // SIFT metodo objekto sukūrimas
    pcl::SIFTKeypoint<pcl::PointXYZRGBA, pcl::PointWithScale> sift_detect;

    // Artimiausių kaimyninių taškų paieškos metodo - kd-tree nustatymas
    sift_detect.setSearchMethod (pcl::search::KdTree<pcl::PointXYZRGBA>::Ptr
(new pcl::search::KdTree<pcl::PointXYZRGBA>));

    // Pagrindinių SIFT meotodo parametrų nustatymas
    sift_detect.setScales (min_scale, nr_octaves, nr_scales_per_octave);
    sift_detect.setMinimumContrast (min_contrast);

    // Objekto taškų debesies nustatymas
    sift_detect.setInputCloud (points);
    sift_detect.compute (*keypoints_out);
    pcl::PointCloud<pcl::PointXYZRGBA>::Ptr keypoints_xyzrgb (new
pcl::PointCloud<pcl::PointXYZRGBA>);
    pcl::copyPointCloud (*keypoints_out, *keypoints_xyzrgb);
}

// Požymių skaičiavimo būdinguose taškuose funkcija
void compute_features_at_keypoints (pcl::PointCloud<pcl::PointXYZRGBA>::Ptr
&points,
                                pcl::PointCloud<pcl::Normal>::Ptr
&normals,
                                pcl::PointCloud<pcl::PointWithScale>::Ptr
&keypoints, float feature_radius,
                                pcl::PointCloud<pcl::FPFHSignature33>::Ptr &descriptors_out)
{
    // Deskriptoriaus objekto sukūrimas
    pcl::FPFHEstimation<pcl::PointXYZRGBA, pcl::Normal, pcl::FPFHSignature33>
est;

```

```

// Artimiausių kaimyninių taškų paieškos metodo - kd-tree nustatymas
est.setSearchMethod (pcl::search::KdTree<pcl::PointXYZRGBA>::Ptr (new
pcl::search::KdTree<pcl::PointXYZRGBA>));

// Artimiausių kaimyninių taškų paieškos apskritimo spindulio RD
nustatymas
est.setRadiusSearch (feature_radius);

// Būdingųjų taškų objekto sukūrimas
pcl::PointCloud<pcl::PointXYZRGBA>::Ptr keypoints_xyzrgb (new
pcl::PointCloud<pcl::PointXYZRGBA>);
pcl::copyPointCloud (*keypoints, *keypoints_xyzrgb);

// Paieškos paviršiaus nustatymas
est.setSearchSurface (points);

est.setInputNormals (normals);

// Objekto taškų debesies nustatymas
est.setInputCloud (keypoints_xyzrgb);

est.compute (*descriptors_out);
}

// Taškų požymių panašumų nustatymo (atitikimų nustatymo) funkcija
void find_feature_correspondences
(pcl::PointCloud<pcl::FPFHSignature33>::Ptr &source_descriptors,
pcl::PointCloud<pcl::FPFHSignature33>::Ptr
&target_descriptors,
std::vector<int> &correspondences_out,
std::vector<float> &correspondence_scores_out)
{
    correspondences_out.resize (source_descriptors->size ());
    correspondence_scores_out.resize (source_descriptors->size ());

    // Artimiausių kaimyninių taškų paieškos ir požymių palyginimo metodo
    nustatymas
    pcl::search::KdTree<pcl::FPFHSignature33> descriptor_kdtree;

    // Taškų debesies, taškų požymių nustatymas
    descriptor_kdtree.setInputCloud (target_descriptors);

    // Geriausių atitikimų išsaugojimas
    const int k = 1;
    std::vector<int> k_indices (k);
    std::vector<float> k_squared_distances (k);
    float suma = 0;

    // Geriausių atitikimų atstumų suma
    for (size_t i = 0; i < source_descriptors->size (); ++i)
    {
        descriptor_kdtree.nearestKSearch (*source_descriptors, i, k, k_indices,
k_squared_distances);
        correspondences_out[i] = k_indices[0];
        correspondence_scores_out[i] = k_squared_distances[0];
        suma += correspondence_scores_out[i];
    }
}

// Objektų taškų debesų palyginimo funkcija
void compare (pcl::PointCloud<pcl::PointXYZRGBA>::Ptr points2,
std::string target_cloud_name_)
{

```

```

// Taškų debesų, deskriptorių, normalių objektų sukūrimas pirmam objektui
pcl::PointCloud<pcl::PointXYZRGBA>::Ptr points1 (new
pcl::PointCloud<pcl::PointXYZRGBA>);
pcl::PointCloud<pcl::PointXYZRGBA>::Ptr downsampled1 (new
pcl::PointCloud<pcl::PointXYZRGBA>);
pcl::PointCloud<pcl::Normal>::Ptr normals1 (new
pcl::PointCloud<pcl::Normal>);
pcl::PointCloud<pcl::PointWithScale>::Ptr keypoints1 (new
pcl::PointCloud<pcl::PointWithScale>);
pcl::PointCloud<pcl::FPFHSignature33>::Ptr descriptors1 (new
pcl::PointCloud<pcl::FPFHSignature33>);

// Taškų debesų, deskriptorių, normalių objektų sukūrimas antram objektui
pcl::PointCloud<pcl::PointXYZRGBA>::Ptr downsampled2 (new
pcl::PointCloud<pcl::PointXYZRGBA>);
pcl::PointCloud<pcl::Normal>::Ptr normals2 (new
pcl::PointCloud<pcl::Normal>);
pcl::PointCloud<pcl::PointWithScale>::Ptr keypoints2 (new
pcl::PointCloud<pcl::PointWithScale>);
pcl::PointCloud<pcl::FPFHSignature33>::Ptr descriptors2 (new
pcl::PointCloud<pcl::FPFHSignature33>);

// Objektų, su kuriais bus lyginami neatpažinti objektai, sarašo įkėlimas
std::ifstream template_input_stream ("clouds.txt");
std::string template_pcd_filename;
std::vector<string> template_ourNames;

float sum_rej_min = 100;
std::string min_template_cloud_name_ = "";

// Objektų įkėlimas
int template_cloud_number = 1;
while (template_input_stream.good ())
{
    std::getline (template_input_stream, template_pcd_filename);
    if (template_pcd_filename.empty () || template_pcd_filename.at (0) ==
'#')
        continue;

// Objekto įkėlimas
pcl::io::loadPCDFile (template_pcd_filename, *points1);

// Duomenų - taškų tankumo keitimas
const float voxel_grid_leaf_size = 0.01;
downsample (points1, voxel_grid_leaf_size, downsampled1);
downsample (points2, voxel_grid_leaf_size, downsampled2);

// Normalių skaičavimas dviems objektams
const float normal_radius = 0.02;
compute_surface_normals (downsampled1, normal_radius, normals1);
compute_surface_normals (downsampled2, normal_radius, normals2);

// Jei skaičiuojami būdingieji taškai
if (keypoints_ == true)
{
// Būdingųjų taškų skaičiavimo metodo - SIFT parametrai
const float min_scale = 0.002;
const int nr_octaves = 10;
const int nr_octaves_per_scale = 3;
const float min_contrast = 0.5;
// Būdingųjų taškų skaičiavimas dviems objektams
detect_keypoints (points1, min_scale, nr_octaves, nr_octaves_per_scale,
min_contrast, keypoints1);

```

```

    detect_keypoints (points2, min_scale, nr_octaves, nr_octaves_per_scale,
min_contrast, keypoints2);

    // Požymių skaičiavimas dviems objektams
    const float feature_radius = 0.04;
    compute_features_at_keypoints (downsampled1, normals1, keypoints1,
feature_radius, descriptors1);
    compute_features_at_keypoints (downsampled2, normals2, keypoints2,
feature_radius, descriptors2);
}
else
{
    // Požymių skaičiavimas dviems objektams
    const float feature_radius = 0.04;
    compute_features (downsampled1, normals1, feature_radius, descriptors1);
    compute_features (downsampled2, normals2, feature_radius, descriptors2);
}

// Taškų požymių palyginimo objekto sukūrimas
pcl::registration::CorrespondenceEstimation<pcl::FPFHSignature33,
pcl::FPFHSignature33> corEst;
corEst.setInputCloud (descriptors1);
corEst.setInputTarget (descriptors2);
boost::shared_ptr<pcl::Correspondences> cor_all_ptr (new
pcl::Correspondences);
corEst.determineCorrespondences (*cor_all_ptr);

// RANSAC metodas ir jo parametrų nustatymas
double epsilon_ransac = 0.1;
int iter_ransac = 1000;

pcl::registration::CorrespondenceRejectorSampleConsensus<pcl::PointXYZRGBA>
ransac;
// Dviejų objektų taškų debesų nustatymas
ransac.setInputCloud (points1);
ransac.setTargetCloud (points2);
ransac.setInlierThreshold (epsilon_ransac);
ransac.setMaxIterations (iter_ransac);
ransac.setInputCorrespondences (cor_all_ptr);
boost::shared_ptr<pcl::Correspondences> corr_inliers_ptr (new
pcl::Correspondences);
ransac.getCorrespondences (*corr_inliers_ptr);

float sum_rej = 0;
for (size_t c = 0; c < corr_inliers_ptr->size (); ++c) {
sum_rej += (*corr_inliers_ptr)[c].distance;
}

// Panašiausio objekto paieška
if ((sum_rej / 100000) < sum_rej_min)
{
sum_rej_min = (sum_rej / 100000);
min_template_cloud_name_ = template_pcd_filename;
}
template_cloud_number++;
}
template_input_stream.close ();
std::cout << "Objektas: " << target_cloud_name_ << " yra panasiausias i: "
<< min_template_cloud_name_ << std::endl;
}

// Pagrindinė programa
int main (int argc, char** argv)
{
    if (argc < 3)

```

```

{
    printf ("No target PCD file given!\n");
    return (-1);
}

// Objekto taškų debesų, objekto sukūrimas
pcl::PointCloud<pcl::PointXYZRGBA>::Ptr points1 (new
pcl::PointCloud<pcl::PointXYZRGBA>);
pcl::PointCloud<pcl::PointXYZRGBA>::Ptr points2 (new
pcl::PointCloud<pcl::PointXYZRGBA>);

// Objektų sąrašą saugančio failo įkėlimas
std::ifstream target_input_stream (argv[2]);
std::string target_pcd_filename;
std::vector<string> target_ourNames; //This is our variable for names

// Pradinių parametrų nustatymas
int target_cloud_number = 1;
int min_target_cloud_number = 1;
int min_template_cloud_number = 1;
std::string min_template_cloud_name = "";

// Objektų, kurie bus lyginami su atpažintais objektas, sąrašo įkėlimas
while (target_input_stream.good ())
{
    std::getline (target_input_stream, target_pcd_filename);
    if (target_pcd_filename.empty () || target_pcd_filename.at (0) == '#')
        continue;

    // Objekto įkėlimas
    pcl::io::loadPCDFile (target_pcd_filename, *points2);

    // Objektų palyginimas
    compare(points2, target_pcd_filename);
    target_cloud_number++;
}
target_input_stream.close ();

return (0);
}

```

Norint pritaikyti deskriptorių, programos tekste keičiami du deskriptoriaus parametrai (DESKR_ID ir DESKR_OBJ). Pavyzdžiui norint pritaikyti FPFH deskriptorių reikia **DESKR_ID** reikšmę pakeisti į **FPFHSignature33** o **DESKR_OBJ** reikšmę į **FPFHEstimation**.

FPFH deskriptorius

DESKR_ID:FPFHSignature33

DESKR_OBJ:FPFHEstimation

PFH deskriptorius

DESKR_ID:PFHSignature125

DESKR_OBJ:PFHEstimation

PFHRGB deskriptorius

DESKR_ID:PFHRGBSignature250

DESKR_OBJ:PFHRGBEstimation

SHOT deskriptorius

DESKR_ID:SHOT352

DESKR_OBJ:SHOTEstimation

COLORSHOT deskriptorius

DESKR_ID:SHOT1344

DESKR_OBJ:SHOTColorEstimation

C priedas. Kompaktinė plokštelė

- **Elektroninė ataskaitos versija**
- **Įgyvendintų algoritmų pirminis programos tekstas**
- **Objektų atpažinimo eksperimentiniai duomenys**
- **Jaunųjų mokslininkų konferencijos dalyvio pažymos kopija**
- **Pranešimo 16-oje Lietuvos jaunųjų mokslininkų konferencijoje medžiaga**