



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMATIKOS STUDIJŲ PROGRAMA

TOMAS UKTVERIS

OBJEKTŲ POZICIJOS IR ORIENTACIJOS
NUSTATYMO METODŲ MOBILIAM ROBOTUI
EFEKTYVUMO TYRIMAS

Magistro darbas

Darbo vadovas

prof. dr. E. Bareiša

KAUNAS, 2014



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMATIKOS STUDIJŲ PROGRAMA

TOMAS UKTVERIS

OBJEKTŲ POZICIJOS IR ORIENTACIJOS
NUSTATYMO METODŲ MOBILIAM ROBOTUI
EFEKTYVUMO TYRIMAS

Magistro darbas

Vadovas:
prof. dr. E. Bareiša
2014-05-26

Recenzentas:
doc. dr. V. Pilkauskas
2014-05-26

Autorius:
IFM-2/1 gr. studentas
Tomas Uktveris
2014-05-26

KAUNAS, 2014

AUTORIŲ GARANTINIS RAŠTAS

DĖL PATEIKIAMO KŪRINIO

2014-05-26d.

Kaunas

Autorius, _____ Tomas Uktveris _____
(vardas, pavardė)

patvirtina, kad Kauno technologijos universitetui pateiktas baigiamasis bakalauro (magistro) darbas
(toliau vadinama – Kūrinys) _____

(kūrinio pavadinimas)

„Objektų pozicijos ir orientacijos nustatymo metodų mobiliam robotui efektyvumo tyrimas“

pagal Lietuvos Respublikos autorių ir gretutinių teisių įstatymą yra originalus ir užtikrina, kad

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

Autorius

_____ Tomas Uktveris _____

(vardas, pavardė)

_____ (parašas)

SANTRAUKA

Šiame darbe tiriami algoritminiai sprendimai mobiliam robotui, leidžiantys aptikti ieškomą objektą bei įvertinti jo poziciją ir orientaciją erdvėje. Atlikus šios srities technologijų analizę surasta įvairių realizacijai tinkamų metodų, tačiau bendro jų efektyvumo palyginimo trūko. Siekiant užpildyti šią spragą realizuota programinė ir techninė įranga, kuria atliktas labiausiai roboto sistemoms tinkamų metodų vertinimas. Algoritmų analizė susideda iš algoritmų tikslumo ir jų veikimo spartos vertinimo panaudojant tam paprastus bei efektyvius metodus. Darbe analizuojamas objektų orientacijos nustatymas iš Kinect kameros gylio duomenų pasitelkiant ICP algoritmą. Atliktas dviejų gylio sistemų spartos ir tikslumo tyrimas parodė, jog Kinect kamera spartos atžvilgiu yra efektyvesnis bei 2-5 kartus tikslesnis sprendimas nei įprastinė stereo kamerų sistema. Objektų aptikimo algoritmų efektyvumo eksperimentuose nustatytas maksimalus aptikimo tikslumas apie 90% bei pasiekta maksimali 15 kadrų/s veikimo sparta analizuojant standartinius VGA 640x480 raiškos vaizdus. Atliktas objektų pozicijos ir orientacijos nustatymo ICP metodo efektyvumo tyrimas parodė, jog vidutinė absoliutinė pozicijos ir orientacijos nustatymo paklaida yra atitinkamai apie 3.4cm bei apie 30°, o veikimo sparta apie 2 kadrai/s. Tolesnis optimizavimas arba duomenų kiekio minimizavimas yra būtinas norint pasiekti geresnius veikimo rezultatus mobilioje ribotų resursų roboto sistemoje. Darbe taip pat buvo sėkmingai realizuotas ir išbandytas techninis roboto sprendimas objektų pozicijos ir orientacijos nustatymui.

SUMMARY

This work presents a performance analysis of the state-of-the-art computer vision algorithms for object detection and pose estimation. Initial field study showed that many algorithms for the given problem exist but still their combined comparison was lacking. In order to fill in the existing gap a software and hardware solution was created and the comparison of the most suitable methods for a robot system were done. The analysis consists of detector accuracy and runtime performance evaluation using simple and robust techniques. Object pose estimation via ICP algorithm and stereo vision Kinect depth sensor method was used in this work. A conducted two different stereo system analysis showed that Kinect achieves best runtime performance and its accuracy is 2-5 times more superior than a regular stereo setup. Object detection experiments showcased a maximum object detection accuracy of nearly 90% and speed of 15 fps for standard size VGA 640x480 resolution images. Accomplished object position and orientation estimation experiment using ICP method showed, that average absolute position and orientation detection error is respectively 3.4cm and 30° while the runtime speed – 2 fps. Further optimization and data size minimization is necessary to achieve better efficiency on a resource limited mobile robot platform. The robot hardware system was also successfully implemented and tested in this work for object position and orientation detection.

TERMINŲ IR SANTRUMPŲ ŽODYNAS

- 2D/3D** (*angl. two/three dimensions*) – plokštumą arba trimatę erdvę nusakantis žymėjimas.
- CAD** (*angl. computer aided design*) – priemonės skirtos projektuoti ir modeliuoti kompiuteriuose.
- CPU** (*angl. central processing unit*) – centrinis skaičiavimus atliekantis procesorius.
- DB** (*angl. database*) – informacijos saugojimo duomenų bazė.
- FAST** (*angl. features from accelerated segment test*) – metodas bruožų nustatymui vaizde.
- GC** (*angl. graph cut*) – algoritmas gylio skaičiavimui iš stereo kadru.
- GCC** (*angl. GNU C compiler*) – atviro kodo C programavimo kalbos kodo kompiliatorius.
- GNU** (*angl. GNU*) – atviro kodo platinimo iniciatyva.
- GPIO** (*angl. general purpose input output*) – elektriniai išvadai skirti bendram naudojimui.
- GPU** (*angl. graphical processing unit*) – vaizdo apdorojimo procesorius.
- HARRIS** (*angl. harris*) – metodas bruožų nustatymui vaizde.
- HD** (*angl. high definition*) – aukštos raiškos (1920x1080) vaizdas.
- HDMI** (*angl. high definition multimedia interface*) – aukštos raiškos vaizdo perdavimo sąsaja.
- HOG** (*angl. histogram of oriented gradients*) – objektų aptikimo vaizduose metodas.
- HSV** (*angl. hue saturation value*) – spalvų erdvė skaidanti vaizdą į tris atspalvio, sodrumo ir reikšmės kanalus.
- ICP** (*angl. iterative closest point*) – iteracijomis paremtas artimiausio taško algoritmas trimatės erdvės taškų sulygiavimui (registracijai).
- IDE** (*angl. integrated development environment*) – aplinka ir priemonės programinės įrangos kūrimui.
- IR** (*angl. infrared*) – elektromagnetinio spektro infraraudonieji spinduliai.
- KLT** (*angl. Kanade-Lucas-Tomasi*) – vaizdo taškų sekimo tarp gretimų kadru algoritmas.
- LIDAR** (*angl. light detection and ranging*) – elektromagnetinių bangų naudojimas atstumo matavime.
- NAND** (*angl. Not AND*) – atmintinių tipas duomenų saugojimui.
- ORB** (*angl. Oriented FAST and rotated BRIEF*) – metodas bruožų nustatymui vaizde.
- RAM** (*angl. random access memory*) – atsitiktinės kreipties operatyvioji atmintis.
- RANSAC** (*angl. random sample consensus*) – atsitiktine paieška ir iteracijomis paremtas metodas matematiniam modeliui priklausančių elementų poaibio suradimui iš elementų aibės.
- RGB** (*angl. red green blue*) – trijų kanalų spalvų erdvė naudojama vaizdų kodavimui.
- ROI** (*angl. region of interest*) – konkreti analizuojama vaizdo sritis.

SAD (*angl. sum of absolute differences*) – algoritmas gylio skaičiavimui iš stereo kadru naudojantis absoliutinių reikšmių skirtumų sumą.

SD (*angl. secure digital*) – atminties kortelių tipas.

SGBM (*angl. semiglobal block matching*) – algoritmas gylio skaičiavimui iš stereo kadru.

SIFT (*angl. scale invariant feature transform*) – metodas bruožų nustatymui vaizde.

SLAM (*angl. simultaneous localization and mapping*) – roboto pozicijos nustatymas ir sekimas.

SSD (*angl. sum of squared differences*) – algoritmas gylio skaičiavimui iš stereo kadru naudojantis reikšmių skirtumo kvadratų sumą.

SURF (*angl. speeded up robust features*) – metodas bruožų nustatymui vaizde.

S-VIDEO (*angl. separate video*) – standartizuota signalų perdavimo sąsaja.

SVM (*angl. support vector machine*) – atraminių vektorių klasifikavimo modelis.

TLD (*angl. tracking-learning-detection*) – objektų aptikimo ir sekimo vaizduose metodas.

UML (*angl. unified modelling language*) – modeliavimo ir specifikacijų kūrimo kalba.

USART (*angl. universal synchronous-asynchronous receiver/transmitter*) – elektronikoje naudojamas nuoseklus duomenų perdavimo standartas.

USB (*angl. universal serial bus*) – standartizuota duomenų perdavimo sąsaja.

VGA (*angl. video graphics array*) – 640x480 pikselių raiškos vaizdo perdavimo ir sąsajos standartas.

WIFI (*angl. wifi*) – duomenų perdavimo bevieliais tinklais technologija.

YUV (*angl. yuv*) – spalvų erdvė susidedanti iš intensyvumo Y bei chromatiškumo U ir V kanalų.

Turinys

PAVEIKSLŲ SĄRAŠAS	10
LENTELIŲ SĄRAŠAS	11
1. ĮVADAS	12
1.1. Tyrimo problematika ir aktualumas	12
1.2. Tyrimo tikslas ir uždaviniai	13
1.3. Dokumento struktūra	13
2. ANALITINĖ DALIS	14
2.1. Egzistuojančių sprendimų apžvalga	14
2.1.1. Žymekliais paremti metodai	14
2.1.2. Modeliais paremti metodai	15
2.1.3. Spalvinio palyginimo ir atpažinimo metodai	15
2.1.4. Bruožais paremti metodai	17
2.2. Naudojamų technologijų analizė	21
2.2.1. Jutikliai aplinkos įvertinimui	21
2.2.2. Važiuoklės bazė ir mechanika	22
2.2.3. Realizacijos platformos	24
2.2.4. Valdymo metodai	26
2.3. Apibendrinimas ir išvados	26
3. TECHINĖS IR PROGRAMINĖS ĮRANGOS PROJEKTAVIMAS	28
3.1. Reikalavimai sprendimui	28
3.2. Funkciniai reikalavimai	28
3.3. Nefunkciniai reikalavimai	32
3.4. Kokybės kriterijai	32
3.5. Sprendimo projektavimas	33
3.5.1. Statinis sistemos vaizdas	33
3.5.2. Dinaminis sistemos vaizdas	34
3.5.2.1. Ieškomo objekto modelio sudarymas apmokant	35
3.5.2.2. Objekto paieška roboto veikimo metu	36
3.5.2.3. Robotų judėjimo planavimas	37
3.6. Sprendimo realizacija	38
3.6.1. Sistemos apmokymas	38
3.6.2. Ieškomo objekto aptikimas vaizde	39
3.6.3. Pozicijos nustatymas naudojant gylio informaciją	39

3.6.4. Pilnos objekto transformacijos nustatymas.....	40
3.6.5. Programinės realizacijos priemonės.....	41
3.6.6. Techninės realizacijos priemonės	41
3.6.7. Sprendimo testavimo apimtis.....	44
3.6.8. Testavimo planas.....	45
3.7. Apibendrinimas ir išvados	47
4. EKSPERIMENTINĖ DALIS	48
4.1. Eksperimentų planavimas	48
4.1.1. Objektų aptikimo metodų tyrimas.....	48
4.1.2. Objektų orientacijos nustatymo metodų tyrimas	49
4.2. Atlikti eksperimentai ir rezultatai	53
4.2.1. Objektų detektorių efektyvumo tyrimo rezultatai.....	53
4.2.2. Orientacijos ir pozicijos nustatymo tikslumo įvertinimas.....	54
4.3. Eksperimentų išvados	58
5. DARBO REZULTATAI IR IŠVADOS	59
6. LITERATŪRA	60

PAVEIKSLŲ SĄRAŠAS

1 pav. Žymeklių pavyzdžiai ir jų sekimas.....	14
2 pav. 3D modelio sutapdinimas su objektu vaizde.....	15
3 pav. Objekto pozicijos ir orientacijos nustatymas pagal briaunas	15
4 pav. Sistema spalvų atpažinimui.....	16
5 pav. Roboto sistema bei kamuolio sekimas skirtingame apšvietime.....	16
6 pav. Segmentacijos pradžia, vaizdas po kalibracijos ir rezultatas	17
7 pav. SIFT rezultatas – surastas ieškomas objektas	17
8 pav. Objekto vaizdas iš stereo kamerų sistemos	18
9 pav. Roboto ir objekto lokalizacija prieš ir po įverčių filtravimo.....	19
10 pav. Haro bruožai ir jų pritaikymas objekto sekimui.....	19
11 pav. HOG gradientų išskyrimas žmogaus siluetai.....	20
12 pav. Objekto kontūro sekimas tarp skirtingų kadru	20
13 pav. Vienos kameros, stereo kamerų ir IR jutiklių robotų sistemos	21
14 pav. Microsoft Kinect ir Asus Xtion gylio kameros	21
15 pav. Lazerinę skanavimo sistemą naudojantis robotas	22
16 pav. Fototranzistoriai ir IR jutiklis.....	22
17 pav. Ultragarso jutiklis, magnetometras ir akcelerometras	22
18 pav. Įvairios robotų važiuoklės	23
19 pav. Dviračių ir keturračių robotų kinematiniai modeliai.....	23
20 pav. Tiračio roboto modelis.....	23
21 pav. Enkoderių diskas ir sistema prie motoro	24
22 pav. Servo motoras.....	24
23 pav. Arduino, Raspberry Pi, BeagleBone, BeagleBoard-xM, VIA APC ir Odroid U3.....	24
24 pav. Pilna roboto sistemos panaudos atvejų diagrama.....	28
25 pav. Vaizdo analizės modulio klasių diagrama.....	33
26 pav. Ieškomo objekto apmokymo veiklos diagrama.....	35
27 pav. Objekto paieškos vykdymo veiklos diagrama.....	36
28 pav. Objekto paieškos vykdymo veiklos diagrama.....	37
29 pav. Sprendimo realizacijos schema	38
30 pav. Gylio skaičiavimas naudojant stereo kamerų sistemą.....	40
31 pav. 1/8 HSP Rock Crawler karkasas-šasi	41
32 pav. Roboto sistemos techninė realizacija	42
33 pav. Objektų detektorių apmokymo ir testavimo procedūra.....	48
34 pav. Stereo kamerų sistema.....	50
35 pav. Kamerų kalibravimas panaudojant šachmatinę plokštumą.....	50
36 pav. Objekto orientacijos ir pozicijos nustatymo tikslumo tikrinimas.....	51
37 pav. Modelio sudarymo ir orientacijos įvertimo procedūra	51
38 pav. Gylio vaizdas ir teisingai nustatyta objekto orientacija	52
39 pav. Kiekvienas taškas ant vienetinės sferos turi unikalias (φ , θ) koordinatas.....	52
40 pav. Eksperimente naudoti objektai ir keletas kadru iš apmokymo sekos.....	53
41 pav. Objektų aptikimo tikslumo ir vykdymo spartos rezultatai	54
42 pav. Pradiniai ir skirtingų gylio metodų sugeneruoti vaizdai	55
43 pav. Gylio algoritmų veikimo spartos palyginimo grafikas.....	56
44 pav. Blogų pikselių skaičiaus nustatymo rezultatų grafikai.....	56
45 pav. Stereo kamerų sistemos ir Kinect gylio vaizdų palyginimo rezultatai.....	57

LENTELIŲ SĄRAŠAS

1 lentelė. Kokybės kriterijai ir vertinimas	33
2 lentelė. Klasių diagramos komponentai.....	34
3 lentelė. Programinės realizacijos priemonės.....	41
4 lentelė. Sistemos komponentai	42
5 lentelė. Sąsajos ir protokolai	44
6 lentelė. Stereo gylio metodų palyginimo kriterijai.....	49
7 lentelė. Gylio metodų spartos ir tikslumo palyginimo rezultatai	55
8 lentelė. Apibendrintos gylio nustatymo įrenginių charakteristikos.....	57
9 lentelė. ICP orientacijos ir pozicijos tikslumo vertinimo rezultatai	58

1. ĮVADAS

Kasdien IT technologijos ir jomis paremta technika vis labiau veržiasi į mūsų gyvenimus. Žmogus, pasitelkdamas savo žinias, kuria sau geresnį gyvenimą ir jį palengvinančias priemones. Su kompiuterijos ir jų technologijų proveržio era vis dažniau minima *robotika*. Tai mokslų sritis – kurioje apsijungia daugelio kitų mokslo šakų – mechanikos, elektronikos, medžiagų, IT ir kt. žinios bei sprendimai. Tai ateities technologijos ir sprendimai, kuriuos kuriame šiandien.

Nauji laimėjimai ir labai aktyvus robotikos srities progresas jau šiandien duoda apčiuopiamą ir išbandomą naudą – robotai naudojami automobilių ir karo pramonėje, medicinoje, aviacijoje, gamybos sektoriuje ir t.t. Robotai pasižymi labai plačiu taikymo bei galimybių spektru, bet kartu ši sritis nenusileidžia ir uždavinių gausa. Pradedant efektyvesnių mechaninių sistemų ir valdymo modelių sudarymu, pažangesnių jutiklių tobulinimu ir kūrimu, visų sistemos elektroninių komponentų sujungimu į bendrą valdymo sistemą bei baigiant programinės įrangos projektavimu – tai tik kelios kryptys robotų kūrimo ir jų technologiniame vystyme. Jose gausu įvairių ir neištyrinėtų problemų bei sudėtingų uždavinių.

1.1. Tyrimo problematika ir aktualumas

Šiuolaikinės robotikos uždaviniai yra pakankamai sudėtingi ir ne taip lengvai išsprendžiami. Efektyviausi ir geriausi mechaniniai bei techniniai sprendimai dar nepakankamai ištobulinti, jog pasiektų žmogaus ar kitų gyvūnų motorikos galimybes. Robotui judant atsirandančios paklaidos jo pozicijos ir orientacijos nustatyme (lokalizacija žemėlapyje) – taip pat rimta, tačiau aktyviai sprendžiama problema. Valdymas ir kontrolė – nėra efektyvių ir universalių apmokymo bei valdymo metodų. Tuo labiau robotų intelektas vis dar tėra primityvus ir jo net negalima palyginti su žmogiškuoju [1].

Vis aktualesnė ir didelio dėmesio sulaukianti robotikos kryptis – *autonominiai ir mobilūs robotai*. Spartus pastarųjų metų progresas ir sujudimas joje rodo, jog tai tikrai perspektyvi ir daug žadanti kryptis. Tai yra pakankamai nauja robotų rūšis, kuriai ypač trūksta naujų metodų bei efektyvesnių esamų problemų sprendimų. Vienas iš dažnai sutinkamų autonominių robotų valdymo uždavinių – objekto paėmimo ir pervežimo iš vienos vietos į kitą. Į šį uždavinį įeina kelios pagrindinės problemos – objekto padėties ir orientacijos nustatymas, tikslus techninės įrangos valdymas ir koordinavimas (siekiant tinkamu kampu ir atstumu priartinti robotą prie objekto), pačio objekto paėmimo veiksmo valdymas bei objekto pargabenimas. Norint kuo geriau atlikti paskutinius paminėtus veiksmus, reikia pirmiausia kuo tiksliau nustatyti objektą ir jo orientaciją.

Šiame darbe yra analizuojami ieškomo objekto pozicijos bei orientacijos nustatymo metodai iš matomos RGB vaizdų sekos. Ieškoma sprendimo, padedančio kuo tiksliau nustatyti minėtus parametrus bei kuris būtų tinkamas naudoti mobilaus autonominio robotu valdyme.

1.2. Tyrimo tikslas ir uždaviniai

Ištirti ir palyginti algoritminius sprendimus mobiliam robotui, leidžiančius kuo tiksliau įvertinti ieškomo objekto poziciją bei orientaciją.

Iškelti uždaviniai:

- Išanalizuoti egzistuojančius objektų aptikimo ir orientacijos nustatymo metodus
- Ištirti metodų efektyvumą panaudojant sukurtą techninę ir programinę įrangą
- Įvertinti metodų tinkamumą uždaviniui spręsti pagal gautus eksperimentų rezultatus

1.3. Dokumento struktūra

Dokumentas suskirstytas į keturis pagrindinius skyrius. Pradžioje pateikiama robotikos srityje naudojamų objektų aptikimo ir orientacijos nustatymo metodų analizė, toliau skaitytojas supažindinamas su darbe projektuojamu techninės ir programinės įrangos sprendimu. Eksperimentinėje dalyje aprašoma tyrimų atlikimo eiga bei pateikiami metodų vertinimo kriterijai. Paskutiniame skyriuje apibendrinamas visas darbas, pateikiami esminiai tyrime gauti rezultatai bei išvados, gairės tolimesniems darbams.

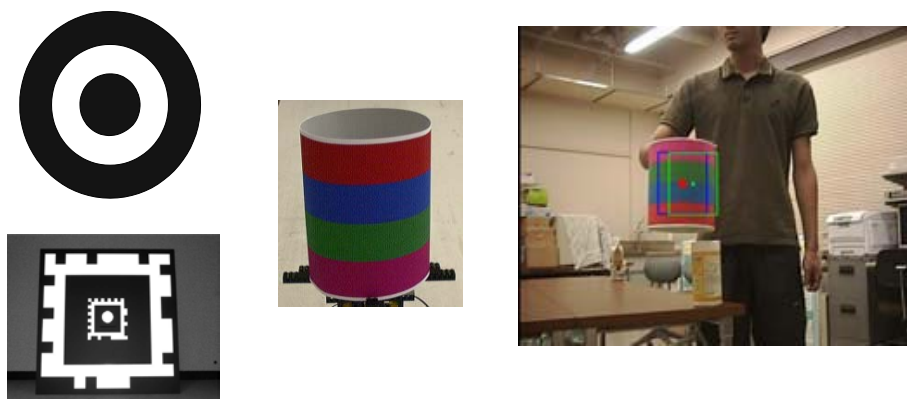
2. ANALITINĖ DALIS

2.1. Egzistuojančių sprendimų apžvalga

Analizuojant įvairius su darbo tema susijusius straipsnius ir mokslines publikacijas nerasta sprendimų įgyvendinančių visus darbo tikslu išskeltus uždavinius, tačiau verta apžvelgti sprendimus, kurie panašūs savo specifiškai ir turiniu į šį darbą. Išsamus objektų sekimo ir lokalizacijos metodikų aprašymas pateikiamas techninėje ataskaitoje [2] bei [3]. Joje esami objektų paieškos sprendimai skirstomi į tris pagrindines grupes: *naudojantys šablonus*, *objektų modelius* ir *objektų bruožus*. Šiuos sprendimus ir juose taikomus metodus apžvelgsime toliau.

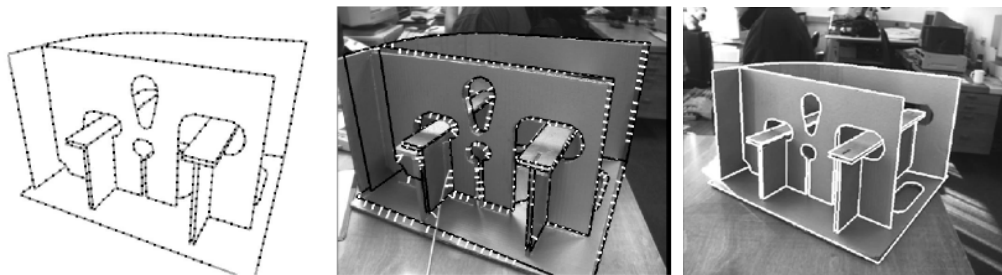
2.1.1. Žymekliais paremti metodai

Žymeklių metode ieškomi objektai pažymimi specialia ir žinomos geometrijos (1 pav.) figūra (markeriu) ar spalvų raštu. Tai pats paprasčiausias metodas. Vaizde aptikti ryškų ir žinomos formos markerį nesudėtinga, be to, tai atliekama pakankamai greitai. Tokio tipo sprendimas siūlomas [4] darbe. Jame aptikimui naudojama vienos kameros sistema. Metodo privalumas – jog iš markerio transformacijos vaizdo galima (*apskritimui* – įvertinus elipsės formos ištempimą, *raštui* – linijų pakreipimą) nustatyti objekto pasisukimą. Kitas analogiškas [5] sprendimas papildomai įvertina tikimybinę objekto poziciją vaizde. Bendru atveju, metodas nėra pakankamai tikslus, kai objektą užstoja kitos kliūtys arba kai žymeklio nėra matyti iš visų objekto pusių. Taip pat nėra labai patogu kaskart žymėti objektus bei ne visada yra galimybė tai padaryti.



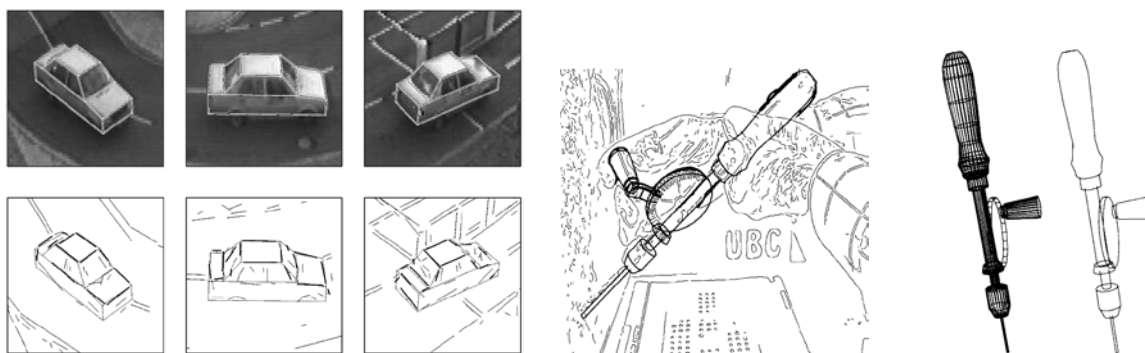
1 pav. Žymeklių pavyzdžiai ir jų sekimas

2.1.2. Modeliais paremti metodai



2 pav. 3D modelio sutapdinimas su objektu vaizde

Šio tipo algoritmuose iš anksto žinoma ieškomo objekto struktūra pagal turimą jo (2 pav.) modelį. Paieškos sistema gali pati susidaryti objekto 3D modelį prieš paiešką arba žinomas CAD modelis gali būti jau įrašytas duomenų bazėje. Modelis dideliu tikslumu [6] padeda nustatyti objekto orientaciją, tačiau reikalauja nemažai skaičiavimo resursų. Pradžioje turi būti nustatomos vaizde esančių objektų briaunos, kampai ar kontūras, tada atliekama modelio sutapdinimo (3 pav.) su išrinktais bruožais operacija. Pats metodas yra atsparus aplinkos apšvietimo pokyčiams. Priklausomai nuo to, kaip tiksliai nusakomi bruožai (briaunoms – sutapdinami tik modelio taškai ar visi segmentai), išlošiama skaičiavimų, tačiau didėja objekto ir modelio orientacijos neatitikimas [3]. Šie neatitikimai iššaukiami dėl kintančios aplinkos – modelio pasislėpimo už kitų objektų, tekstūros nevientisumo, šešėlių ir pan. Yra ir metodų [7], kurie objektų ir modelių sutapdinimą vykdo iteracijomis (optimizuoja), taip minimizuojant galimą paklaidą. Pagrindinis trūkumas – jog reikia turėti jau anksčiau paruoštą objekto modelį, o naudojant iteracinius algoritmus išauga skaičiavimų skaičius.

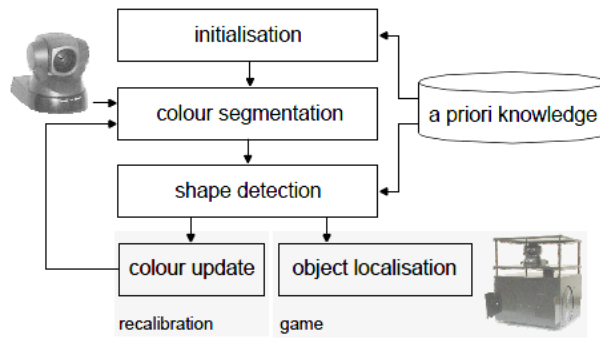


3 pav. Objekto pozicijos ir orientacijos nustatymas pagal briaunas

2.1.3. Spalvinio palyginimo ir atpažinimo metodai

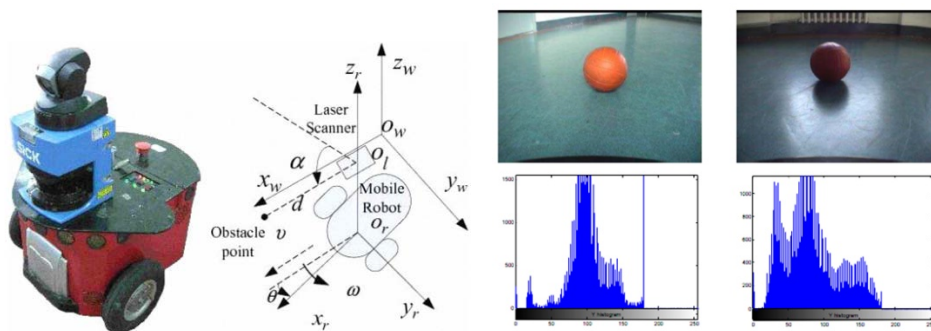
Spalvinio apdorojimo metodai paremti įvairių vaizdo spalvų erdvių – RGB, YUV, HSV ir t.t. analize. Visi algoritmai, naudojantys šį metodą, turi būti apmokomi arba jiems pradžioje turi būti iš anksto nustatomos spalvų diapazono reikšmės duomenų bazėje (4 pav.). Paieška vykdoma ieškant artimų spalvų regionų – atliekant vaizdo segmentaciją. Šie metodai yra neatspariausi aplinkos

apšvietimo pasikeitimui, tačiau gali būti sparčiai apdorojami, kas labai svarbu, esant robotams skaičiavimo resursams robote.



4 pav. Sistema spalvų atpažinimui

Objektų nustatymo vaizde problema, įvertinant kintantį apšvietimą, nagrinėjama [8, 9] darbuose. Vaizdas analizuojamas YUV spalvų erdvėje, kadangi tos pačios spalvos objektus prie skirtingo apšvietimo galima lengviau atskirti lyginant jų intensyvumus, o ne RGB reikšmes tiesiogiai. Metodas sudaro analizuojamo vaizdo histogramas (5 pav.) ir vykdo adaptyvų spalvų palyginimą naudodamas automatiškai nustatomus slenksčius (*angl. thresholds*). Algoritmas taip pat pakankamai tiksliai nustato judantį objektą. Papildomai – aptikimo tikslumui pagerinti ir objekto pozicijos judėjimo hipotezei sudaryti, naudojamas Kalmano filtras. Pagrindiniai trūkumai – nagrinėjamas per primityvus objektas (kamuolys), todėl apie objekto orientacijos nustatymą nekalbama. Patį objektą gali būti sudėtinga atpažinti už kliūčių ar atskirti nuo panašios spalvos kitų objektų. Neįvertinama ir tai, jog ieškomas objektas gali ir nebūti vienspalvis, todėl papildomai reikėtų atpažinti paviršiaus tekstūrą ar raštą.



5 pav. Roboto sistema bei kamuolio sekimas skirtingame apšvietime

Analogiški ir panašios spartos pavyzdžiai, taip pat pritaikyti roboto valdymui, pateikiami ir literatūroje [10, 11, 12, 13]. Atliekama spalvų segmentacija taip pat YUV erdvėje. Automatinė kalibracija pagal sudarytas histogramas leidžia pašalinti apšvietimo efektus prisitaikant iš naujo prie aplinkos. Suradus spalvos regioną naudojama atsitiktinė Hofo transformacija (*angl. randomized Hough transform*) apskritimų paieškai. Apskritimo centras randamas pagal Gauso pasiskirstymo modelį. Nauja kamuolio pozicija numatoma naudojant Bajeso tikimybinį modelį. Metodo privalumas – galimybė automatiškai išmokti naujų objekto spalvų vykdant pakartotinę kalibraciją (6

pav.), taip pat – kontūro informacijos panaudojimas. Autoriai pateikia trūkumus – jautrumas šešėliams, kamuolio apskritimas esant skirtingoms apšvietimo sąlygoms gali nesutapti su tikru. Vėlgi metodas pritaikytas tik kamuolio tipo objektų paieškai ir neįvertina sudėtingesnių objektų bei orientacijos.



6 pav. Segmentacijos pradžia, vaizdas po kalibracijos ir rezultatas

Visi spalvų palyginimo metodai yra jautrūs aplinkai – gali būti sudėtinga atlikti atpažinimą tokiose aplinkose, kurių fonas labai panašus į nagrinėjamo objekto, todėl reikalinga papildomai analizuoti kitą prieinamą informaciją – briaunas, kontūrą, tekstūrą ar kitus objektui būdingus bruožus.

2.1.4. Bruožais paremti metodai

Šiuose metoduose ieškoma vaizde esančių būdingų bei lengvai sekamų žymių (*angl. feature points*). Daugelis algoritmų analizuoja keletą kadrų iš eilės ir susieja kiekviename kadre esančius bruožus – ieško kitimo tendencijų. Analizuojami tokie bruožai kaip: segmentai, kampai, kontūras ir t.t. Segmentų ir tiesių išskyrimui naudojama **Hofo transformacija** (*angl. Hough transform*). Kampų ir taškų ant objekto sekimui gali būti panaudojamas **KLT** (santr. *Kanade-Lucas-Tomasi*) algoritmas [14].



7 pav. SIFT rezultatas – surastas ieškomas objektas

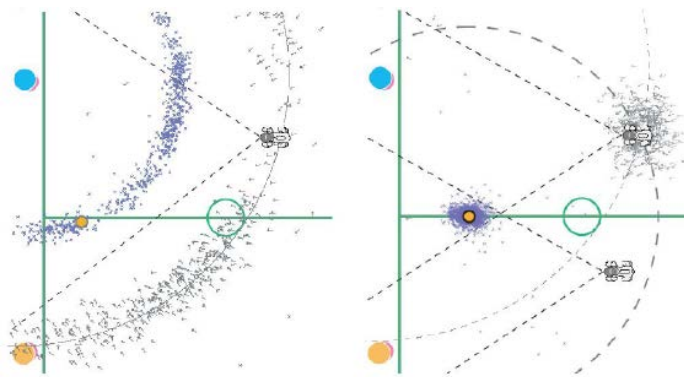
Turint ieškomo objekto vaizdą gali būti naudojamas **SIFT** (*angl. Scale Invariant Feature Transform*) algoritmas [15] ar kiti bruožų detektoriai (FAST, SURF, HARRIS, ORB ir pan.), įvertinantys vaizdo gradientus ar randantys kitus stabilius ir atsikartojančius vaizdo bruožus. Daugelis detektorių yra atsparūs vaizdo poslinkio, sukimo ir mastelio keitimo operacijoms. Leidžia aptikti skirtingo dydžio, dalinai pasislėpusius, pasuktus ar skirtingos orientacijos tuos pačius

objektus naujame vaizde (7 pav.). Vienas iš pagrindinių SIFT algoritmo trūkumų – gana imlus skaičiavimams, taip pat metodas geriausiai tinka tik analizuoti 2D erdvėje transformuotus vaizdus kaip plokštumas, o 3D atpažinimui nėra tiesiogiai pritaikomas. Šio metodo panaudojimas objekto orientacijai nustatyti aprašomas [16]. Nurodytame darbe naudojama dviejų kamerų stereo sistema bei SVM (*angl. support vector machines*) klasifikatorius. Su kamerų sistema (9 pav.) sukuriama 8 vaizdų duomenų bazė kiekvienam objektui, kai objektas pasukamas skirtingu kampu. SIFT metodu nustatoma objekto pozicija vaizde, o klasifikatorius toliau identifikuoja objekto tipą ir jo pasisukimo kampą su 90% tikslumu. Metodo trūkumai – fiksuota aplinka ir objekto pasisukimų kampų skaičius, apribotas objekto sukimasis Z ašyje, tinkama tik plokščių paviršių paieškai.



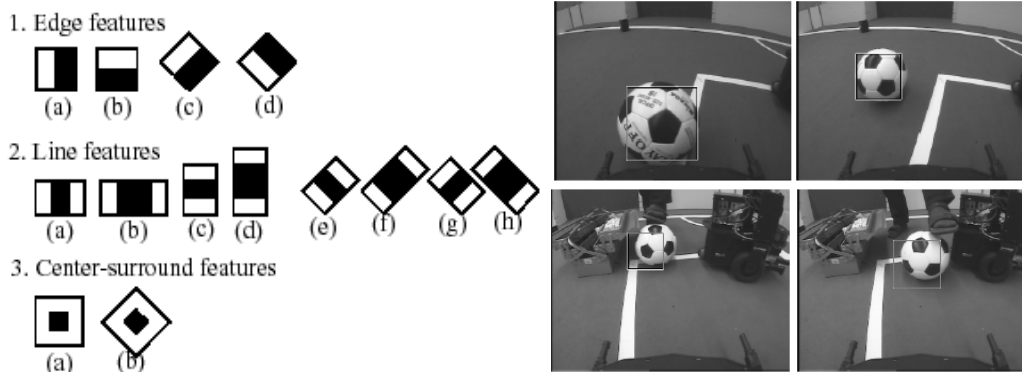
8 pav. Objekto vaizdas iš stereo kamerų sistemos

Įverčių filtrai (*angl. particle filters*) yra tikimybinis principas naudojamas Kalmano, Monte Carlo bei Markovo metoduose. Lokalizuodami objektą ir nustatant jo poziciją jie remiasi informacija apie daugelį tos pozicijos įverčių praeityje (t.y. jų tankio pasiskirstymą) bei tolesnį vykdomą/numatomą sistemos veiksmą. Pozicijos įverčiai lyginami su matomomis žymėmis (*angl. landmarks*) vaizde, kurios gali būti sudaromos automatiškai besimokant aplinką. Filtravimu atkreipiamas dėmesys į atsirandančias paklaidas dėl jutiklių signalų diskretizavimo, judėjimo ir kitų netikslumų bei leidžiama jas kompensuoti. Šie principai objektų nustatymui ir sekimui pritaikyti darbuose [5, 9, 17, 18, 19]. Kaip pristato minėti darbai, šie sprendimai puikiai (net su 95% tikslumu) veikia nežinomose aplinkose, be to nereikalauja daug skaičiavimų. Filtru įvertinus senesnius pozicijos įverčius ir aplinkines žymes galima pagal jų tankį lokalizuoti ieškomą objektą (9 pav.).



9 pav. Roboto ir objekto lokalizacija prieš ir po įverčių filtravimo

Haro bruožų (*angl. Haar features*) objektų paieškos vaizde metodas [20] vienas iš geriausiai žinomų algoritmų. Pirmąkart buvo pritaikytas veidų paieškai, bet gali būti naudojamas bet kokiems objektams surasti – reikia tik apmokyti klasifikatorių. Algoritmas nurodo vieną ar daugiau ROI (*angl. region of interest*) sričių vaizde, kuriuose yra didžiausia tikimybė surasti ieškomą objektą. Metodas konstruoja vieną stiprų klasifikatorių jungdamas daug paprastų (*angl. weak*) klasifikatorių į kaskadinę grandinę (filtrą). Paieška yra vykdoma monochrominiame vaizde. Skaičiuojamos vaizdo regionų integralinės sumos naudojant Haar kaukes (10 pav.). Algoritmas pritaikytas objektų atpažinimui realiu laiku [22], tačiau išlieka intensyvus skaičiavimų prasme. Jo privalumas – jį nereikia įvertinti spalvinės informacijos ir gali su 91% tikslumu aptikti dalinai matomą objektą net ir už kliūčių. Vis dėlto jis reikalauja didelio skaičiaus pradinių apmokymo duomenų (teigiamų ir neigiamų vaizdų) tam, kad būtų sudarytas minėtą tikslumą užtikrinantis klasifikatorius.



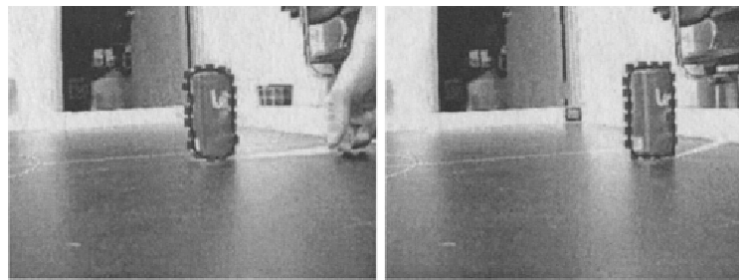
10 pav. Haro bruožai ir jų pritaikymas objekto sekimui

Labai panašus į Haar metodą yra **HOG** metodas [21], kuris naudoja SVM klasifikatorių apmokyta iš vaizdo gradientų informacijos. Fiksuoto dydžio paieškos lange, kuris suskirstytas į 8 atskirus blokus (celes), iš kiekvienoje celėje (11 pav.) esančių pikselių gradientų skaičiuojama 9 elementų histograma. Iš histogramų sudaromas požymių vektorius, kuris ir naudojamas tolesniam palyginimui. Algoritmas pirmąkart pritaikytas žmonių paieškai vaizduose, tačiau gali būti apmokytas bet kokių objektų paieškai. Šiam algoritmui taip pat reikia daug teigiamų ir neigiamų pradinių pavyzdžių, kad būtų galima apmokyti tikslų klasifikatorių.



11 pav. HOG gradientų išskyrimas žmogaus siluetai

Kontūro nustatymas ir sekimas (12 pav.) – kitas išskirtų bruožų panaudojimo pavyzdys [23]. Toks metodas pasiteisina esant prastoms apšvietimo sąlygoms ir yra tinkamas, kai objekto forma gali kisti. Paminėtame darbe objekto formos ribos yra dinaminės (kontūras) ir gali laisvai keistis (didėti/mažėti). Darbe panaudojamas energijos minimizavimo metodas kontūro pasikeitimams aptikti. Sekamų spalvų informacija atnaujinama kiekvieną kadrą, tai leidžia dinamiškiau įvertinti ir kintamo apšvietimo problemą. Uždavinys sprendžiamas dinaminio programavimo būdu. Naudoja ankstesnių kontūrų informaciją iš prieš tai buvusių vaizdo kadrų, kad būtų sumažinamos skaičiavimų apimtys, tačiau algoritmas vis tiek išlieka imlus skaičiavimams.



12 pav. Objekto kontūro sekimas tarp skirtingų kadrų

Objektų aptikimo ir sekimo sprendimai kaip Zdanek Kalal pristatytas **TLD** (*angl.* Tracking-Larning-Detection) ir Georg Nebehay realizuotas OpenTLD [24] leidžia apmokyti ieškomą objektą naudojant jo vaizdo šablonus. Pristatytas algoritmas ne tik naudoja ankstesnių vaizdo kadrų ir objekto pozicijos informaciją, taip užtikrindamas didesnę stabilumą, bet ir sugeba mokytis iš savo klaidų, kadangi jame realizuota automatinio apsimokymo funkcija. Metodas pritaikytas dirbti realiu laiku, skelbiamas algoritmo efektyvumas 81%.

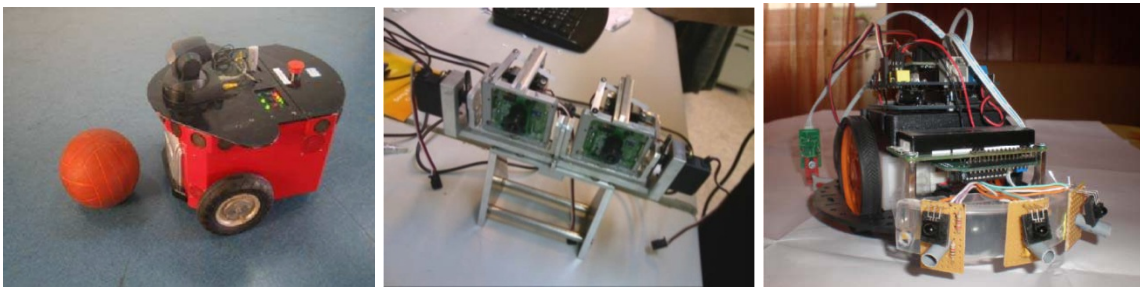
Efektyvūs sprendimai pristatyti [25] darbe, sujungiantys spalvinę ir gylio informaciją iš Microsoft Kinect kameros. Pristatytas **LINE-MOD** algoritmas buvo specialiai sukurtas vienspalvių ir objektų be tekstūros paieškai. Metodas leidžia sudaryti objekto paieškos duomenų bazę iš tūkstančių ieškomo vaizdo šablonų išlaikant artimą realiam laikui 10 kadrų/s apdorojimo spartą. Taip pat pagal aptinkamą objekto šabloną galima nustatyti ir pradinę apytikslę ieškomo objekto orientaciją, kai šablonas susietas su tokia informacija. Algoritmo pagrindinis trūkumas, jog reikia gylio kameros, norint juo pasinaudoti.

2.2. Naudojamų technologijų analizė

Panaudota techninė ir programinė įranga taip pat ne ką mažiau svarbi realizuojant robotikos srities sprendimus bei juos testuojant. Šiame skyriuje paanalizuosime esamas technines realizacijas ir jų komponentus.

2.2.1. Jutikliai aplinkos įvertinimui

Vaizdo kameros – vieni sudėtingiausių šiuo metu egzistuojančių jutiklių. Leidžia gauti milijonus signalo reikšmių-taškų daugiau nei 30 kadrų per sekundę greičiu. Labiausiai paplitusios žmogui matomo elektromagnetinio spektro RGB vaizdo kameros. Pagal sistemos komponentų skaičių (13 pav.) skiriamos vienos [26] (*angl. monocular*), dviejų (*angl. stereo*) [16] ir daugiau kamerų sistemos. Svarbiausi parametrai – skiriamoji geba (320x240, 640x480 ir pan.), fokusavimo atstumas (nuo 2cm iki begalybės). Robotų sistemose naudojama nedidelė raiška arba specialiai sumažinama ir tada analizuojama, kad apdorojimo sparta būtų didesnė. Stereo kamerų sistemos privalumas – leidžia papildomai išskirti vaizdo gylį naudojant trianguliacijos principą, tačiau tokia sistema reikalauja kiek galima identiškų kamerų, jų kalibracijos ir vaizdų sutaikymo tikslumui pasiekti.



13 pav. Vienos kameros, stereo kamerų ir IR jutiklių robotų sistemos

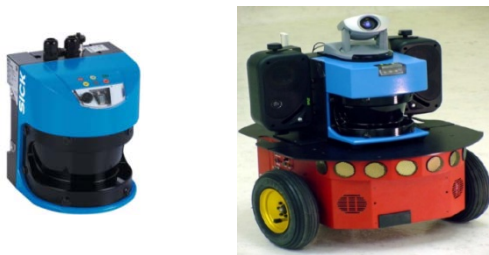
Populiarus sprendimas robotikos taikymuose naudoti vartotojams lengvai prieinamą Asus Xtion ar Microsoft Kinect kamerą (14 pav.), kuri geba 30 kadrų/s greičiu pateikti RGB ir gylio informaciją. Kamera naudoja struktūrinės šviesos (*angl. structured light*) metodą projektuodama žinomo rašto kaukę IR projektoriumi priešais save. Iš šio rašto pakitimų nustatomas matomo vaizdo gylis. Šių kamerų privalumas, jog jų vaizdo jutikliai yra gamykliškai sulygiuoti ir sukalibruoti.



14 pav. Microsoft Kinect ir Asus Xtion gylio kameros

Kitas sprendimas gyliui nustatyti yra naudoti specializuotą LIDAR (*angl. light detection and ranging*) lazerinę sistemą – skanerį. Prietaisas priešais save projektuoja lazerio šviesos liniją, o

robote analizuojama vieno skanavimo metu liniją sudarantys taškai. Yra nemažai autonominių robotų sistemų [26], naudojančių SICK® lazerinį skanerį (15 pav.). Lazerinės sistemos privalumai – didelis tikslumas, atsparumas triukšmams, trūkumai – labai brangi technologija.



15 pav. Lazerinę skanavimo sistemą naudojantis robotas

Taip pat naudojamos infraraudonųjų spindulių – IR kameros ir kiti jutikliai (fotorezistoriai, fototranzistoriai) [27], kai reikalinga analizuoti vaizdą prieblandoje/tamsoje (16 pav.). Tokie jutikliai taip pat taikomi artimo atstumo matavimams (iki 10 cm), o roboto sistemai pateikia ne RGB duomenis, o išmatuoto intensyvumo lygio reikšmę priklausomai nuo atstumo iki kliūtis.



16 pav. Fototranzistoriai ir IR jutiklis

Prie sudėtingesnių jutiklių galima priskirti ultragarso jutiklius (17 pav.), taip pat giroskopus, akselerometrus, magnetometrus, kurie plačiai naudojami kituose elektronikos prietaisuose ir yra labai mažų matmemų. Pasitelkiant keletą sensorių, gaunama patikimesnė judėjimo ir kliūčių informacija, o tai labai svarbu roboto valdymo tikslumui.



17 pav. Ultragarso jutiklis, magnetometras ir akselerometras

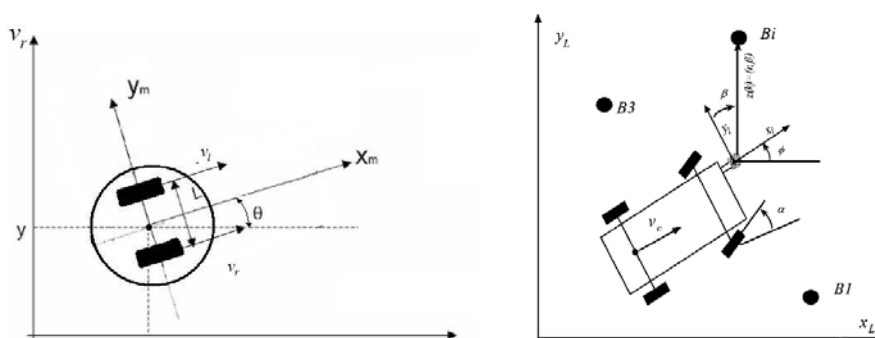
2.2.2. Važiuklės bazė ir mechanika

Priklausomai nuo roboto pravažumo, aplinkos – kurioje jie judės, pervežamų krovinių masės bei paties uždavinio reikalavimų, yra pasirenkamas roboto karkasas arba važiuklė (18 pav.). Tai gali būti nedidelis dviejų ratų [1] robotas arba artimas automobilio dydžiui keturratis, sveriantis net 680kg [28] ir daugiau.

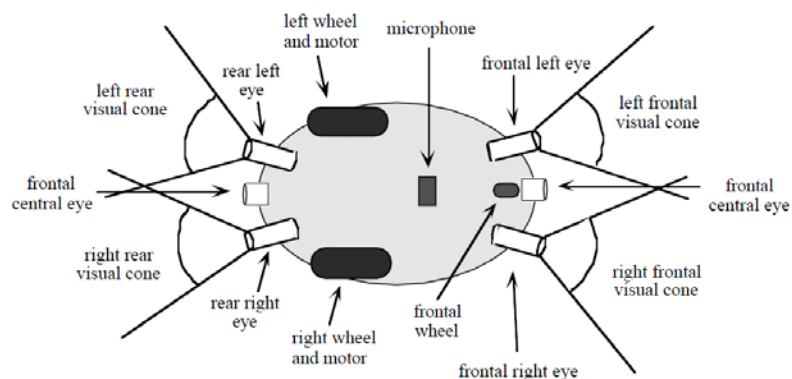


18 pav. Įvairios robotų važiuoklės

Roboto kinematiniai modeliai sudaromi priklausomai nuo roboto formos (19 pav.) ir galimybės judėti. Kuo daugiau laisvės laipsnių robotas turi, tuo sudėtingesnis yra jo valdymas ir judėjimas. Mažiau negu tris ratus (20 pav.) turinčių robotų ratai dažniausiai būna susieti su korpusu ir nesisukinėja į šonus.



19 pav. Dviračių ir keturračių robotų kinematiniai modeliai



20 pav. Triračio roboto modelis

Judančių robotų nuvažiuotas atstumas matuojamas prie ratų ar jų sukimosi ašies montuojama enkoderių (*angl. encoder*) sistema (21 pav.). Joje šviesos jutiklių pagalba pagal specialiai (paprasčiausiuose – vienodu raštu, sudėtingesniuose – dvejetainiu ar Grėjaus kodu) sužymėtą diską įmanoma nustatyti, kiek ir į kurią pusę pasisuko ratai, o žinant ratų spindulį ir nuvažiuotą atstumą. Dažnai ši sistema būna įtaisyta brangesniuose motoruose. Nuo disko žymių tankumo ir optinės sistemos priklausau pasisukimo kampo nustatymo tikslumas. Vieno koncentrinio takelio užtenka judėjimui nustatyti, su dviem – galima nustatyti sukimosi kryptį, o naudojant tris – skaičiuoti pilnus

apsisukimus. Bet koku atveju tokia sistema nėra be trūkumų, pagrindinė problema – per ilgą laiką besiakumuliuojanti paklaida arba atskaitos pradžios dreifas (*angl. zero drift*).



21 pav. Enkoderių diskas ir sistema prie motoro

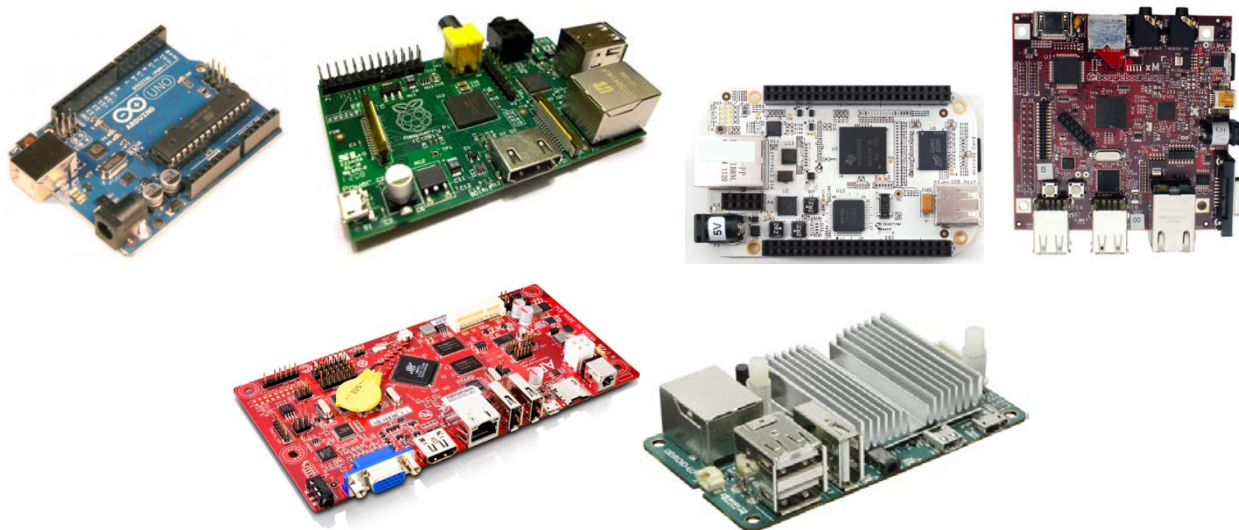
Kitų besisukinėjančių ir mechaninių dalių valdymas atliekamas naudojant (22 pav.) servo motorus. Šie motorai pasižymi dideliu kampiniu tikslumu ir yra valdomi impulsais. Kiekvienu impulsu servo motoras pasisuka fiksuotu kampu (nuo 1.8° ir daugiau), o priklausomai nuo tipo atlaiko 16-40 Kg•cm sukimo momentą prie 1.6 A.



22 pav. Servo motoras

2.2.3. Realizacijos platformos

Robotų logika ir įvairūs skaičiavimai atliekami naudojant mikroprocesorines sistemas ar mikrokompiuterius. Dabartiniu metu siūloma daug įvairių ir realizacijai tinkamų gaminių, o svarbiausi jų skirtumai – techninės specifikacijos ir kaina. Šiame skyriuje trumpai apžvelgsime egzistuojančias elektronikos priemones ir sprendimus.



23 pav. Arduino, Raspberry Pi, BeagleBone, BeagleBoard-xM, VIA APC ir Odroid U3

Vis dar populiarai ir tokia išliekanti – **Arduino** platforma (23 pav.). Tai Atmel firmos mikrovaldiklius naudojančios plokštės skirtos įvairiems elektronikos projektams. Skiriama net 15 skirtingų oficialių šios plokštės versijų – Arduino UNO (ATmega328, 16MHz, 16 GPIO, €20), Arduino MEGA (ATmega2560, 16MHz, 54 GPIO, USB, €39) ir t.t. bei naujausias sprendimas Arduino DUE naudojantis 32 bit Atmel SAM3X8E ARM Cortex-M3 procesorių dirbantį 84MHz dažniu ir turintį 54 GPIO išvadus bei 2x mikro-USB. Galima sakyti, kad tai labiausiai pamėgta ir paprasčiausia eksperimentuotojų platforma. Prie šių plokščių nemokamai siūloma programavimo ir kūrimo grafinė platforma paremta Processing IDE. Taip pat per daugelį metų sukurta labai daug įvairių bibliotekų ir modulių. Gaminys plačiai taikomas tiek robotikos tiek ir kitose srityse.

Kreditinės kortelės dydžio kompiuteris **Raspberry Pi** yra galingesnė platforma taip pat tinkanti ir taikoma įvairiose srityse. Tai Raspberry Pi fondo iniciatyva edukaciniais tikslais sukurtas pigus kompiuteris-platforma. Pristatytos versijos A - kainuojanti \$25 ir B – kainuojanti \$35. Svarbiausi versijų skirtumai – techniniai. B versijoje turime 700MHz ARMv11 Broadcom procesorių, 512MB RAM, 10/100 Ethernet, 2xUSB, 40xGPIO jungčių, RCA, HDMI, 3.5mm audio, SD kortelės ir mikroUSB jungtis. Taip pat yra GPU su OpenGL ES 2.0 Full HD (1080p) vaizdo spartinimu. A versijoje tėra 256MB RAM ir nėra Ethernet jungties. Sprendimas yra efektyvus energetiniu požiūriu, reikalinga 5V įtampa, o naudojama ir tik 3.5W - 700mA (A versijoje - 1.5W ir 300 mA) energijos. Platforma veikia su įvairiomis Linux distribucijomis Raspbian, Fedora, ArchLinux ir kt. Programuoti galima Python kalba, C ar C++, taip pat prieinami visi standartiniai Linux įrankiai.

Analogiškas variantas yra **APC VIA** kompanijos gaminys–plokštė. Joje galime pamatyti ARM Cortex-A9 procesorių su 800MHz taktiniu dažniu, 512MB DDR3 RAM, 4GB NAND Flash (arba 2GB pagal modelį) vidinę atmintį. Plokštės dydis standartizuoto Neo-ITX tipo. Jungiamas prie monitoriaus per HDMI arba VGA jungtį, taip pat turi nuo 2x iki 4x USB, 10/100 Ethernet, microUSB, audio-out/microphone-in ir microSD jungtis. Prireikus prijungti papildomą elektroniką tai galima padaryti per GPIO išvadus. Pateikiami trys plokštės variantai – standartinis 8750 (\$49), *rock* (\$79) patobulintas ir skirtas daugiau hakeriams, *paper* (\$99) - turi knygos tipo popierinį korpusą su aliuminio šonu. Priklausomai nuo versijos gali apdoroti HD arba Full HD vaizdą, valdomas per Android 4.0 ir Linux.

BeagleBoard bandymų plokštės, oficialiai prieinamos trijų versijų: *BeagleBone*, *BeagleBoard* ir *BeagleBoard-xM*. Galima rinktis kokio dydžio reikia ir pagal turimų lėšų kiekį. Jų kaina atitinkamai - \$89, \$137 bei \$202. Paprasčiausia ir mažiausia versija, t.y. *BeagleBone*, turi 720MHz ARMv7 Cortex-A8 procesorių, 256MB RAM, 1x USB, 10/100 Ethernet, 92 GPIO jungtis, 1x microSD, 4x valdomus LED ir veikia su Angstrom Linux, Android ar kita Linux distribucija. Didesnė *BeagleBoard* versija turi Texas Instruments OMAP3530 Cortex-A8 600MHz procesorių, 128MB DRAM, 256MB NAND Flash, TMS320C64x+ GPU veikiantį 430MHz dažniu. Turi abi –

kairės ir dešinės, 3.5mm audio jungtis, SD kortelių skaitytuvą, DVI-D jungtį monitoriui prijungti, S-VIDEO, 1x USB. BeagleBoard-xM yra pagerinta versija, kurioje procesoriaus dažnis padidintas iki 1 GHz, RAM iki 512MB bei atsirado 10/100 Ethernet bei 4x USB jungtys.

Jeigu reikia išties galingo sprendimo, o dydis irgi svarbus – yra **Odroid U2/U3** mikrokompiuteris \$65 (+\$30 už atsiuntimą). Šis mažas (59 x 57 x 60 mm) įrenginys yra ypač spartus, turi Exynos4412 Prime Cortex-A9 1.7GHz Quad Core procesorių ir 2GB DDR2 RAM. Taip pat yra 10/100Mbps Ethernet, HDMI, USB ir Audio išvestis, micro SD, trūkumas – nėra GPIO išvadų, papildomai elektronikai prijungti, tačiau visada tai galima realizuoti ir per USB ar USART išėjimą. Maitinamas 5V ir naudoja iki 2A srovės. Šilumai išsklaidyti reikalingas visą plokštę apimantis radiatorius.

2.2.4. Valdymo metodai

Tinkamas roboto valdymas yra svarbus faktorius bendro uždavinio sprendimui. Egzistuoja įvairūs valdymo metodai, jie skirstomi [1] taip:

- **Būsenų mašinos** (*angl. state machine*) – tinka paprastesnių uždavinių (statinių žemėlapių) sprendimui, griozdiškas modelis, sunku sugalvoti ir aprašyti visas būsenas, sudėtinga derinti ir projektuoti, tačiau nereikia apmokyti;
- **Braitenbergo modeliai** – roboto varikliai sujungti tiesiai su jutikliais, kiekvienai jungčiai suteikiamas svoris, veikia pagal grįžtamojo ryšio (*angl. feedback*) metodą, gali būti sudėtinga nurodyti tinkamus svorius, yra galimybė apmokyti (genetiniai metodai);
- **Elgsenomis grįšti modeliai** – užprogramuojama keletas elgsenų, kurios keičiamos pagal situaciją (pvz., klajoti, sekti ir t.t.), taikomas būsenų koordinavimas – konkuruojantis arba kooperacinis; problema – nežinoma, kiek ir kokių elgsenų reikės (sunku projektuoti), turi būti geras elgsenų koordinatorius;
- **Neuroniniai tinklai** – svorinis tinklas iš keleto paslėptų tinklo lygių, svorių optimizavimas (greitesni nei genetiniai), gali pakliūti į lokalius ekstremumus arba per daug evoliucionuoti (grėsmė permokyti);
- **Miglotoji logika** – geriausia nežinomai aplinkai analizuoti, pasitelkiama [26] ir [27] darbuose, aprašomos elgsenos bazinės taisyklės.

2.3. Apibendrinimas ir išvados

Taikomų kompiuterinės regos metodų ir algoritmų uždaviniui spręsti – išties gausu. Konkrečių realizuotų sprendimų robotikos srityje, nustatančių objekto orientaciją ir poziciją, įvertinant kintantį aplinkos apšvietimą, stinga. Vieną universalų atsakymą į iškeltą uždavinį surasti sunku, todėl atlikus pradinę srities analizę galima teigti, jog problemos sprendimas turės būti įvairių

algoritmų ir metodų kombinacija. Taip pat daugelis iš naujausių ir esamų algoritmų yra sunkiai panaudojami robotikoje, kadangi neatitinka kai kurių svarbių charakteristikų ir reikalavimų [32]:

1. Apdorojimas realiame laike – robotų algoritmai turi veikti greitai ir kuo sparčiau reaguoti į supančios aplinkos pasikeitimus
2. Minimali pradinė informacija – neturi reikalauti neautomatinio kalibravimo ar daug pradinių duomenų
3. Adaptacija naujose aplinkose – veikti nežinomoje ir kintančioje aplinkoje be žmogaus įsikišimo

Taigi, toliau realizuojant ir projektuojant sprendimą, reikės įvertinti šiuos bei kitus sritis, techninius ir funkcinius apribojimus. Technologine prasme, valdymo sistemų ir techninių sprendimų pasirinkimą nulemia turimi resursai ir patirtis dirbant su elektronikos komponentais. Planuojama sukurtą sprendimą išbandyti su paruošta technine įranga realiomis sąlygomis.

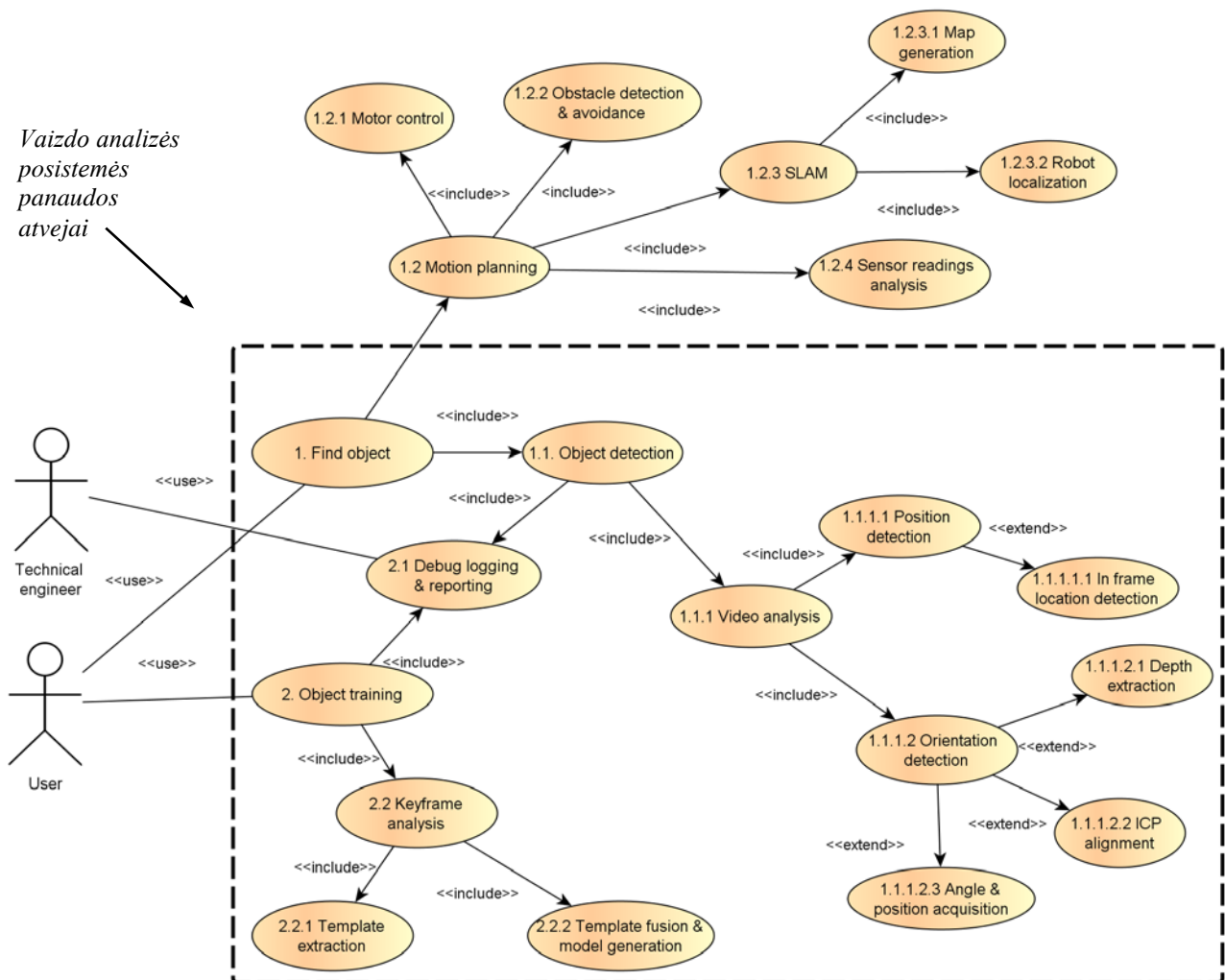
3. TECHINĖS IR PROGRAMINĖS ĮRANGOS PROJEKTAVIMAS

3.1. Reikalavimai sprendimui

Siekiant sudaryti metodus, gebančius kuo tiksliau nustatyti ieškomo objekto poziciją ir orientaciją iš RGB vaizdo sekos, būtina apibrėžti reikalavimus sistemos realizacijai, siekiamus kokybės ir jos įvertinimo kriterijus bei realizacijos testavimo procedūras. Sudarant šiuos reikalavimus taip pat atsižvelgiama į technines realizacijos charakteristikas bei aplinkos, kurioje veiks autonominė roboto sistema, sąlygas.

3.2. Funkciniai reikalavimai

Pilną autonominio roboto sistemą sudaro aibė įvairių funkcijų – pradedant navigacija žemėlapyje/aplinkoje, jutiklių duomenų nuskaitymu, motorų ir kitos techninės įrangos valdymu, vaizdo analize ir pan. Autonominio roboto sistemos funkcijos pavaizduotos toliau panaudos atvejų (24 pav.) diagramoje.



24 pav. Pilna roboto sistemos panaudos atvejų diagrama

Trumpai apžvelgiamos pagrindinės sistemos funkcijos:

1. Find object – objekto paieškos aplinkoje funkcija, kuria naudosis vartotojas eksploatuodamas robotą.

1.1 *Object detection* – čia atliekami skaičiavimai susiję su ieškomo objekto paieška, nustatymu bei lokalizacija žemėlapyje.

1.1.1 *Video analysis* – RGB vaizdo srautas yra apdorojamas analizuojant atskirus kadrus siekiant išgauti reikiamą informaciją.

1.1.1.1 *Position detection* – viena iš objekto nustatymo užduočių yra jo pozicijos nustatymas vaizdo kadre ir žemėlapyje.

1.1.1.1.1 *In frame location detection* – sprendimas turi sugebėti aptikti objektą vaizdo kadre ir nurodyti konkrečią kadro vietą.

1.1.1.2 *Orietation detection* – objekto pozicijos ir pasisukimo erdvėje nustatymo funkcionalumas.

1.1.1.2.1 *Depth extraction* – funkcija, kuri iš stereo kamerų sistemos išgautą gylį panaudoja trimatei objekto pozicijai nustatyti.

1.1.1.2.2 *ICP alignment* – orientacijos nustatymui sistema turi atlikti objekto modelio taškų ir vaizde esančių taškų suluginimą ICP metodu.

1.1.1.2.3 *Angle & position acquisition* – iš sutapdinto modelio sistema turi leisti nusakyti objekto transformaciją erdvėje (pasukimo kampus ir poziciją).

1.2 *Motion planning* – bet kurio judančio roboto viena iš pagrindinių dalių yra judėjimo planavimas, čia atliekami skaičiavimai ir tolesni roboto judėjimo koordinavimo veiksmai norint pasiekti užduotą tikslą.

1.2.1 *Motor control* – tinkamas motorų valdymas yra labai svarbi roboto judėjimo funkcija bet kurioje aplinkoje.

1.2.2 *Obstacle detection & avoidance* – aplinka, kurioje veikia robotas, dažnai yra dinaminė, pasitaiko įvairios kliūtys. Funkcija realizuoja kliūčių nustatymą ir jų išvengimo metodiką.

1.2.3 *SLAM* – (angl. *simultaneous localization and mapping*) judantis robotas turi žinoti savo aplinką ir suvokti, kurioje aplinkos vietoje jis yra. Pozicijos lokalizacija ir žemėlapijo sudarymo funkcija yra būtina autonominiam robotui.

1.2.4 *Sensor readings analysis* – jutiklių duomenų gavimas ir analizė yra neatsiejama bet kurios mechaninės bei techninės sistemos dalis norint nagrinėti aplink esančią aplinką.

2. Object training – ieškomo objekto apmokymo funkcija leidžianti sudaryti jo modelį iš surastų ir vaizde esančių bruožų. Atliekama prieš pradedant objekto paiešką aplinkoje.

2.1 *Debug logging & reporting* – pagalbinė sisteminė funkcija leidžianti registruoti žurnale visą esminę sistemos veikimo informaciją bei būseną, informuoti apie klaidas.

2.2 *Keyframe analysis* – sistema turi sudaryti modelį analizuodama matomą vaizdą, kai atsiranda papildoma nauja informacija – turi būti išsaugomas šis (raktinis) kadras

2.2.1 *Template extraction* – modelio sudarymui sistema turi mokėti išskirti ieškomo objekto būdingus bruožus ar vaizdus.

2.2.2 *Template fusion & model generation* – surinkti modelio bruožai apmokymo metu sistemoje vėliau turi būti paverčiami į objektą nusakantį modelį (apmokymas).

Sistemos aktoriai:

- *virtotojas* – naudoja valdymo ir apmokymo funkcijas,
- *techninis personalas* – atlieka priežiūros bei sistemos derinimo funkcijas

Visi reikalavimai sistemai toliau pateikiami pagal Volere šabloną.

Reikalavimo nr.	1	Panaudojimo atvejo nr.	Panaudos atvejis 1.1
Reikalavimas	<i>Ieškomas objektas turi būti atpažįstamas vaizde.</i>		
Aprašymas	Paieškos funkcija leidžianti robotui surasti reikiamą objektą.		
Pagrindimas	–		
Atitikimo kriterijus	Sistema turi objektų atpažinimo funkciją, ji yra veikianti – leidžia ieškoti nurodytų objektų.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	2	Panaudojimo atvejo nr.	Panaudos atvejis 1.1.1.1.1
Reikalavimas	<i>Objektas turi būti identifikuojamas kadro erdvėje.</i>		
Aprašymas	Nustatyti kadro regioną, kuriame yra matomas objektas (jeigu yra).		
Pagrindimas	–		
Atitikimo kriterijus	Sistema kadre randa ieškomą objektą ir jį pažymi (nurodo regioną).		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	3	Panaudojimo atvejo nr.	Panaudos atvejis 1.1.1.1
Reikalavimas	<i>Turi būti nustatoma matomo objekto pozicija aplinkoje.</i>		
Aprašymas	Objekto lokalizacijai svarbu nustatyti jo poziciją aplinkoje globalios/santykinės ar roboto koordinatų sistemos atžvilgiu.		
Pagrindimas	–		
Atitikimo kriterijus	Sistema pateikia ir nustato objekto poziciją aplinkoje atžvilgiu roboto.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	4	Panaudojimo atvejo nr.	Panaudos atvejis 1.1.1.2
Reikalavimas	<i>Turi būti nustatoma objekto transformacija erdvėje.</i>		
Aprašymas	Siekiant paimti bet kokį objektą reikia, jog būtų žinoma jo erdvinė orientacija.		
Pagrindimas	–		
Atitikimo kriterijus	Sistema sugeba nustatyti objekto 3D transformaciją.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	5	Panaudojimo atvejo nr.	Panaudos atvejis 1.1.1.2.1
Reikalavimas	<i>Sistema turi įvertinti vaizdo gylį informaciją.</i>		
Aprašymas	<i>Pilnos 3D pozicijos įvertinimui sistema turi sugebėti išskirti gylį iš vaizdo.</i>		
Pagrindimas	–		
Atitikimo kriterijus	Sistema pateikia objekto poziciją kaip trimatį vektorių.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	6	Panaudojimo atvejo nr.	Panaudos atvejis 2
Reikalavimas	<i>Turi būti realizuotas ieškomo objekto apmokymas.</i>		
Aprašymas	Sprendimai nenaudojantys apmokymo nėra konkurencingi.		
Pagrindimas	–		
Atitikimo kriterijus	Realizuotas automatinis apmokymas naujo objekto paieškai.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	7	Panaudojimo atvejo nr.	Panaudos atvejis 2.1
Reikalavimas	<i>Sistemos veikimo rezultatai turi būti fiksuojami žurnale.</i>		
Aprašymas	Analizei ir stebėjimui sistema turi išvesti informaciją į žurnalinį failą.		
Pagrindimas	–		
Atitikimo kriterijus	Sistema rašo informacinius pranešimus į žurnalinį failą.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	8	Panaudojimo atvejo nr.	Panaudos atvejis 2.2.1
Reikalavimas	<i>Apmokymo metu turi būti išskiriami objekto bruožai.</i>		
Aprašymas	Naudoti tik esminę, objektą nusakančią, informaciją apmokymui.		
Pagrindimas	–		
Atitikimo kriterijus	Apmokymo metu išskiriami vaizdai ar bruožai modelio sudarymui.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	9	Panaudojimo atvejo nr.	Panaudos atvejis 2.2.2
Reikalavimas	<i>Apmokymo metu išskirti bruožai turi būti naudojami modelio sudarymui.</i>		
Aprašymas	Iš bruožų reikia sudaryti objektą atitinkantį modelį.		
Pagrindimas	–		
Atitikimo kriterijus	Apmokymo metu yra sudaromas ieškomo objekto modelis.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	10	Panaudojimo atvejo nr.	Panaudos atvejis 1.1.1.2.2
Reikalavimas	<i>Gylis turi būti panaudojama objekto orientacijos nustatymui.</i>		
Aprašymas	Panaudojant ICP metodą nustatyti objekto transformaciją erdvėje.		
Pagrindimas	–		
Atitikimo kriterijus	ICP metodas naudojamas objekto transformacijos nustatymui.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	11	Panaudojimo atvejo nr.	Panaudos atvejis 2.2
Reikalavimas	<i>Apmokymo metu turi būti naudojami kamera matomo vaizdo kadrai.</i>		
Aprašymas	Raktiniai kadrai sudaro neperteklinę reikalingą bruožų informaciją modeliui.		
Pagrindimas	–		
Atitikimo kriterijus	Sistema sudaro modelį naudodama neperteklinius kadrus iš visų objekto pusių.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	12	Panaudojimo atvejo nr.	Panaudos atvejis 1.1.1
Reikalavimas	<i>Roboto sistemoje turi būti analizuojama vaizdo informacija.</i>		
Aprašymas	Objektų nustatymo aplinkoje pagrindinis jutiklis yra vaizdo kamera.		
Pagrindimas	–		
Atitikimo kriterijus	Sistema ieško objektų naudodama vaizdo kameros pateikiamą informaciją.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	13	Panaudojimo atvejo nr.	Panaudos atvejis 1.1.1.2.3
Reikalavimas	<i>Nustatoma objekto transformacija turi nusakyti poziciją ir orientaciją.</i>		
Aprašymas	Nustatyta objekto transformacija erdvėje turi būti pilna.		
Pagrindimas	–		
Atitikimo kriterijus	Grąžinama objekto transformacija turi posūkio ir poslinkio informaciją.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

3.3. Nefunkciniai reikalavimai

Sprendimui keliami nefunkciniai reikalavimai apibrėžia, kaip posistemė turi atlikti savo funkcijas. Nustatomi spartos, tikslumo bei kiti reikalavimai.

Reikalavimo nr.	14	Panaudojimo atvejo nr.	Nefunkcinis
Reikalavimas	<i>Objekto aptikimas ir orientacijos nustatymas atliekamas per 1 sekundę ar greičiau.</i>		
Aprašymas	Mobilūs robotai yra realaus laiko sistemos, todėl taikomų metodų ir algoritmų sparta turi būti kiek galima didesnė ir kuo artimesnė kameros vaizdo įrašymo greičiui su atitinkamu rezervu.		
Pagrindimas	–		
Atitikimo kriterijus	Objekto nustatymas vaizde neužtrunka ilgiau nei numatyta.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	15	Panaudojimo atvejo nr.	Nefunkcinis
Reikalavimas	<i>Sprendimas turi veikti dinaminėse ir nežinomose aplinkose.</i>		
Aprašymas	Realizacija neturi priklausyti nuo apšvietimo sąlygų ar konkrečios roboto veikimo aplinkos.		
Pagrindimas	–		
Atitikimo kriterijus	Sistemos metodai veikia vienodai gerai esant skirtingam apšvietimui bei aplinkoms.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

Reikalavimo nr.	16	Panaudojimo atvejo nr.	Nefunkcinis
Reikalavimas	<i>Turi būti aptinkami bet kokios formos, spalvos ir tekstūros objektai.</i>		
Aprašymas	Sprendimas neturi priklausyti nuo objekto savybių.		
Pagrindimas	–		
Atitikimo kriterijus	Sprendimas vienodai gerai veikia su bet kokiais objektais.		
Data	Versija	Pakeitimo aprašymas	Autorius
2013-04-01	1.0	Pradinė versija	T.Uktveris

3.4. Kokybės kriterijai

Tam, kad realizuojamas sprendimas būtų gerų kokybinių charakteristikų, jam iškeliami kokybės reikalavimai. Sprendimo realizacijos kokybės užtikrinimui pasirinkti programinės įrangos

bendrieji ir su vaizdo analize susiję kokybės įvertinimo metodai. Kriterijai toliau apibrėžti 1 lentelėje.

1 lentelė. Kokybės kriterijai ir vertinimas

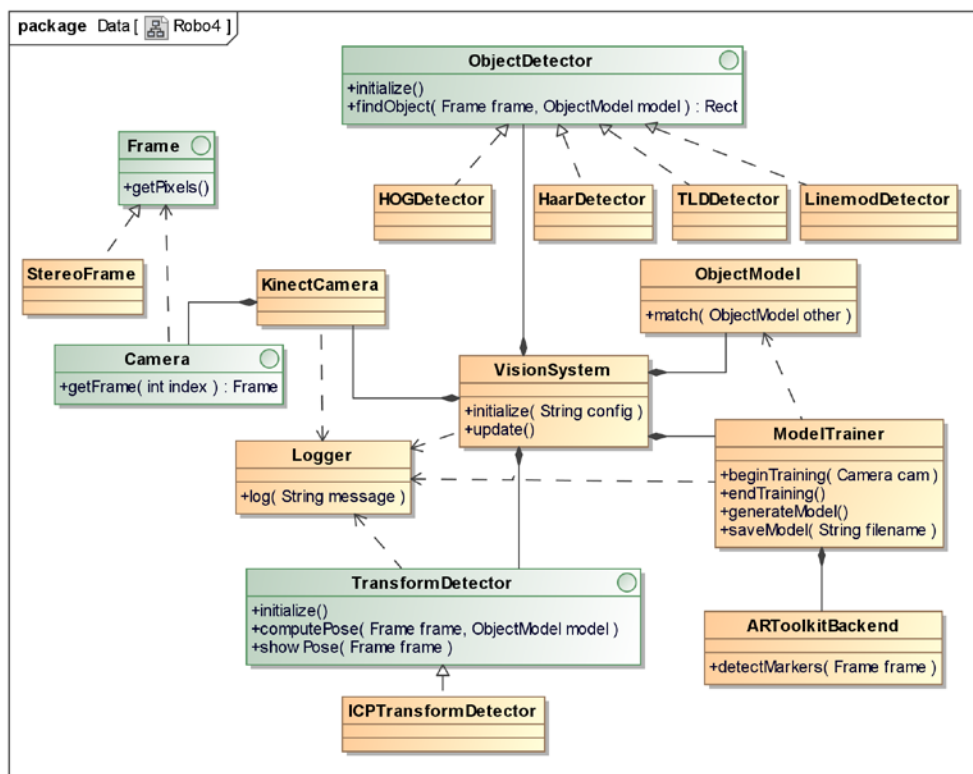
Kriterijus	Vertinimo metodika
1. Funkcinių ir nefuncinių reikalavimų atitikimas	Patikrinama, ar visi funkciniai ir nefunkciniai reikalavimai yra realizuoti ir išpildyti.
2. Objekto nustatymo tikslumas	Vertinama, ar ieškomas objektas yra teisingai nustatomas vaizde ir ar nenustatomas, jeigu jo vaizde nėra.
3. Pozicijos nustatymo tikslumas	Palyginama nustatyta pozicijos reikšmė su realia išmatuota testavimo aplinkoje.
4. Orientacijos nustatymo tikslumas	Palyginama, ar nustatyta orientacija sutampa su objekto orientacija ir kokia yra kampinė paklaida.
5. Apdorojimo sparta	Matuojamas algoritmų veikimo laikas.

Detalesnė sprendimo vaizdo analizės dalies kokybės vertinimo metodika aprašoma 4.1.2.3 skyriuje.

3.5. Sprendimo projektavimas

3.5.1. Statinis sistemos vaizdas

Įvertinus reikalavimus vaizdo analizės sistemai ir siekiant geriau įvertinti sprendimo architektūrą, sudarytas statinis sistemos vaizdas – klasių diagrama (25 pav.). Diagramos elementų klasių aprašymas pateikiamas toliau (2 lentelė).



25 pav. Vaizdo analizės modulio klasių diagrama

2 lentelė. Klasių diagramos komponentai

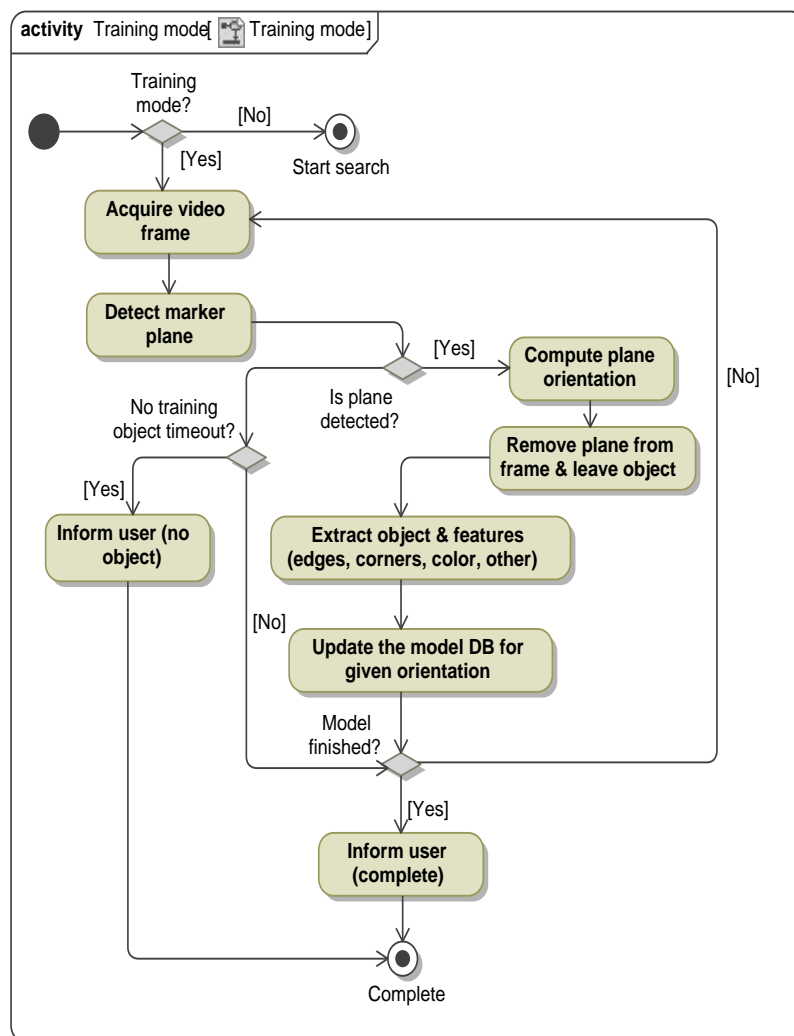
Klasė	Aprašymas
<i>VisionSystem</i>	Pagrindinė vaizdo analizės sistemos klasė. Metodai: <i>initialize(...)</i> – veiksmai sistemos inicializavimui ir darbui <i>update()</i> – periodiškai kviečiama funkcija vaizdo analizei
<i>ARToolkitBackend</i>	Klasė realizuojanti AR žymeklių funkcijas. Metodai: <i>detectMarkers(...)</i> – kviečiama siekiant aptikti žymeklius kadre
<i>ModelTrainer</i>	Pradinio apmokymo proceso veiksmus realizuojanti klasė. Iš nustatytų bruožų DB sukuria objekto modelį. Metodai: <i>beginTraining(...)</i> – leidžia pradėti ieškomo objekto apmokymą <i>endTraining()</i> – ieškomo objekto apmokymo pabaiga <i>generateModel()</i> – sukuriamas ieškomo objekto modelis iš sukauptos bruožų informacijos DB <i>saveModel(...)</i> – leidžia išsaugoti apmokytą modelį diske
<i>ObjectModel</i>	Iš vaizde esančių bruožų sudarytas objekto modelio aprašymas ir jo duomenys. Metodai: <i>match(...)</i> – leidžia palyginti modelį su kitu
<i>ICPTransformDetector</i>	Klasė realizuojanti ICP taškų suluginimo ir transformacijos išgavimo metodus.
<i>HOGDetector</i> , <i>HaarDetector</i> , <i>TLDDetector</i> , <i>LinemodDetector</i>	Objektų atpažinimo iš vaizdo kadro metodų realizacijų klasės.
<i>KinectCamera</i>	Stereo kamerų sistemą apibrėžianti klasė su metodais.
<i>StereoFrame</i>	Stereo kamerų kadra apibendrinančioji klasė.
<i>Logger</i>	Sisteminių-žurnalinių pranešimų išvedimą realizuojanti klasė. Metodai: <i>log(...)</i> – leidžia išvesti pranešimą į žurnalinį failą
Sąsajos (interfeisai)	
<i>TransformDetector</i>	Objekto transformacijos nustatymo iš vaizdo kadro realizacijos metodų klasė. Metodai: <i>initialize()</i> – pradinei modulio inicializacijai <i>computePose(...)</i> – įvertinama objekto orientacija vaizde <i>showPose(...)</i> – pavaizduojama orientacija kadre
<i>ObjectDetector</i>	Klasifikatoriaus sąsaja, specifikuoja bendrą klasifikatorių struktūrą ir realizuojamus metodus. Metodai: <i>initialize()</i> – pradinei klasifikatoriaus inicializacijai <i>findObject(...)</i> – nustato, ar vaizdo kadre yra ieškomas objektas
<i>Frame</i>	Vaizdo kadro sąsaja apibrėžianti apdorojimo funkcijas. Metodai: <i>getPixels(...)</i> – grąžina nurodyto kanalo kadro informaciją
<i>Camera</i>	Kameros sąsaja apibrėžianti metodus vaizdo įvedimo funkcijas. Metodai: <i>getFrame(...)</i> – grąžina nuskaitytą vaizdo kadra

3.5.2. Dinaminis sistemos vaizdas

Roboto sistemos veikimą apibendrintame lygyje parodo toliau pateiktos veiklos diagramos. Jos nusako tris pagrindines roboto sistemos funkcijas: *apmokymą*, *objekto paiešką* ir *judėjimą aplinkoje*.

3.5.2.1. Ieškomo objekto modelio sudarymas apmokant

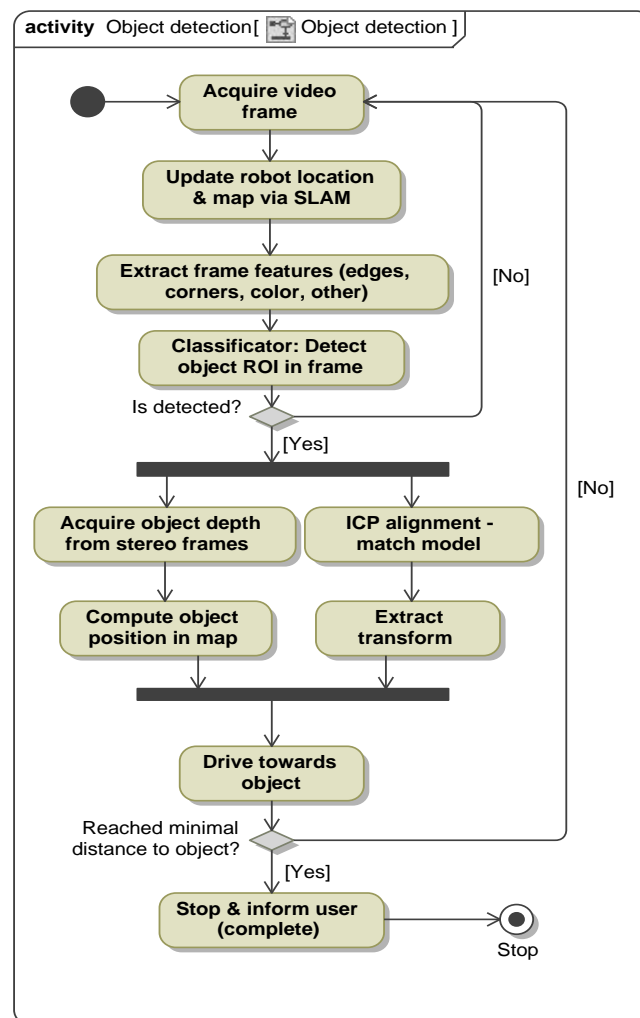
Apmokymas, tai režimas, kuris gali būti įjungtas robotui, siekiant sukurti naują ieškomo objekto modelį sistemos duomenų bazėje. Apmokymas vyksta (26 pav.) rodant naują objektą tiesiai prieš kameras įvairiais kampais. Tam, kad būtų geriau įvertinama ir objekto orientacija – papildomai objektas patalpinamas ant žymekliais pažymėtos plokštumos, kuri sukiojama kartu su objektu. Išskiriant objekto bruožus, kurie bus naudojami ieškomo objekto modeliui sudaryti, plokštumos dalis iš vaizdo yra pašalinama. Modelis yra atnaujinamas kiekvienu raktiniu kadru (kuris suteikia daugiau informacijos apie objektą). Kai daugiau naujos informacijos apie modelį nebegalima surinkti, modelis traktuojamas kaip užbaigtas ir vartotojas informuojamas apie sėkmingą modelio sudarymą garsiniu signalu ar kitu būdu.



26 pav. Ieškomo objekto apmokymo veiklos diagrama

3.5.2.2. Objekto paieška roboto veikimo metu

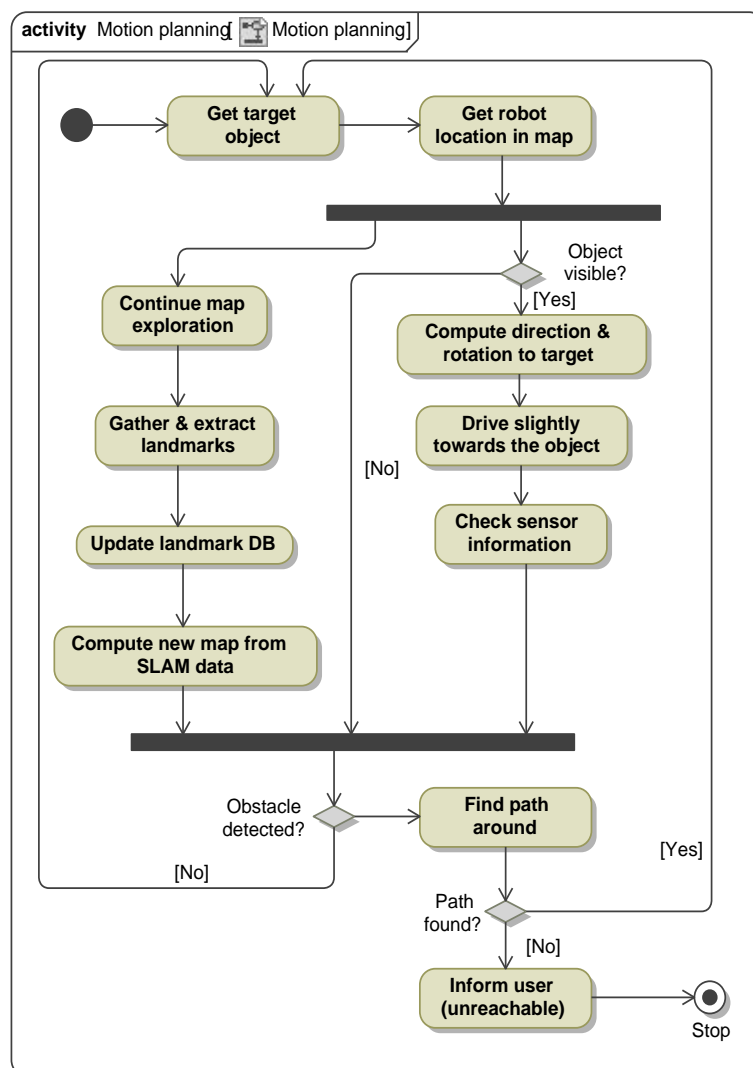
Objekto paieška yra antrasis roboto veikimo režimas, kuriame vykdoma jau išmokto objekto paieška nežinomoje aplinkoje. Veikimo metu (27 pav.) yra gaunama vaizdo kadro informacija iš kameros jutiklio. Kiekvienas kadras analizuojamas išskiriant esminius vaizdo bruožus ir pateikiamas klasifikatoriui, kuris nustato, ar pateiktame vaizde yra ieškomas objektas ar ne. Sėkmingos paieškos metu klasifikatorius grąžina srities koordinates vaizde, kuriame yra objektas. Tam, kad būtų galima nustatyti objekto trečiąją pozicijos vektoriaus komponentę bei poziciją aplinkos žemėlapyje yra įvertinamas vaizdo gylis iš stereo kamerų sistemos. Taip pat lygiagrečiai atliekami modelio ir matomo vaizdo palyginimo veiksmai. Jų metu naudojant ICP algoritmą turimas objekto modelis yra sukiojamas įvairiais kampais ir bandoma optimizavimo būdu sutapatinti gautą vaizdą su matomu objekto vaizdu. Sėkmingai sutapatinus galima įvertinti ir objekto orientaciją vaizde. Žinant šią informaciją robotas toliau gali važiuoti link objekto, kol jį pasieks bei atlikti kitus veiksmus.



27 pav. Objekto paieškos vykdymo veiklos diagrama

3.5.2.3. Roboto judėjimo planavimas

Nuolat vykdoma roboto sistemos veikla yra judėjimo planavimas (28 pav.). Turint ieškomo objekto vaizdą ir žinant savo poziciją žemėlapyje, robotas gali judėti autonomiškai tyrinėdamas aplinką, kol suras tą objektą. Roboto lokalizacija žemėlapyje ir paties žemėlapio sudarymas yra nuolat vykstantis procesas SLAM (*angl. simultaneous localization and mapping*), kurio metu iš kameros jutiklio gražinamo vaizdo nustatomos matytos ir nematytos aplinkos detalės (*angl. landmarks*). Pagal jas robotas orientuojasi aplinkoje, t.y. leidžia jam pačiam nustatyti, kurioje vietoje jis yra žemėlapyje. Jeigu roboto sistema nustato vaizde ieškomą objektą, tai didesnis prioritetas teikiamas judėjimui link to objekto. Tuo tikslu yra paskaičiuojama kryptis, atstumas iki objekto ir judama lėtai iki jo įvertinant aplinkos kliūtis. Tam naudojami kiti aplinkos jutikliai. Nustačius kliūtį kelyje, roboto sistema bando ją apvažiuoti (rasti kitą kelią). Įvertintas ir tas atvejis, jeigu kelio iki objekto nėra – tuomet robotas negali pasiekti ieškomo objekto ir apie tai informuoja vartotoją garsiniu signalu ar kitu būdu. Paieška ir SLAM yra vykdomi nuolat, kol sistema nepasiekia savo tikslo – neranda ir nenuvažiuoja iki objekto.

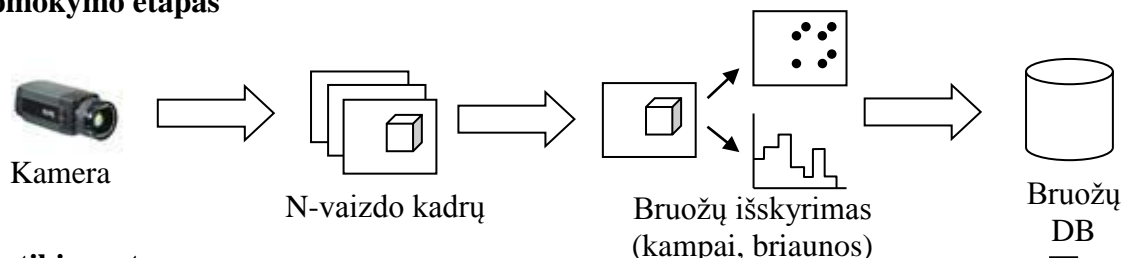


28 pav. Objekto paieškos vykdymo veiklos diagrama

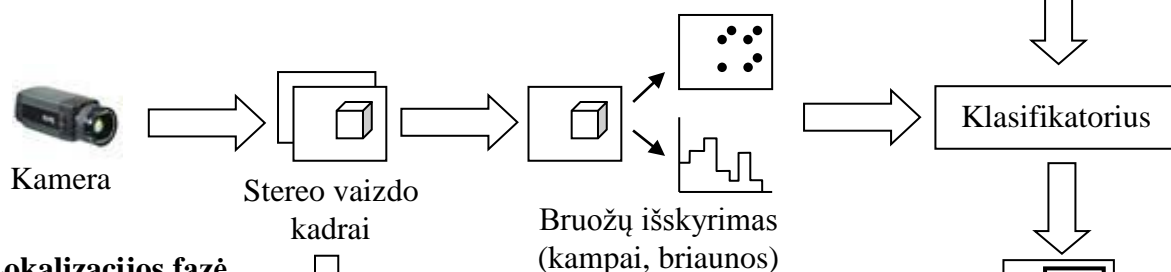
3.6. Sprendimo realizacija

Išnagrinėjus esamus ir dažniausiai naudojamus metodus objektų aptikimui bei orientacijos nustatymui buvo pasirinktas 3-jų etapų roboto sprendimas kaip pavaizduota (29 pav.). Kiekvienas iš jų aprašytas toliau.

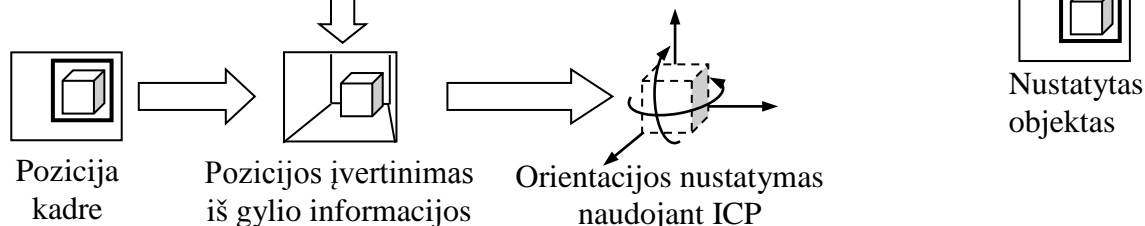
I. Apmokymo etapas



II. Aptikimo etapas



III. Lokalizacijos fazė



29 pav. Sprendimo realizacijos schema

3.6.1. Sistemos apmokymas

Objektas patalpinamas ant žymekliais pažymėtos plokštumos ir rodomas stereo kamerų sistemai iš visų pusių sukinėjant plokštumą kartu su objektu. Apmokymo režime analizuojami vaizdo kadrai ir iš jų išskiriamas objektas bei jo taškai (*angl. point cloud*). Gylgio informacija panaudota kartu su žymeklių plokštuma leidžia tiksliau nustatyti objekto orientaciją jau apmokymo fazėje bei padeda greitai išskirti objektą iš aplinkos, taip pašalinant nereikalingą informaciją. Plokštumos nustatymui naudojamos ARTToolkit+ bibliotekos funkcijos, o objekto segmentacijai – PointCloud bibliotekos plokštumos išskyrimo ir grupavimo (*angl. clustering*) funkcijos, o vaizdų apdorojimui – OpenCV biblioteka.

Kiekvieno kadro metu surasti ir išskirti taškai yra sujungiami ir sudaromas ieškomo objekto taškų modelis. Taip pat iš gautų objekto RGB vaizdų bus apmokomas objekto paieškos klasifikatorius. Bendru atveju, apmokymo fazės rezultatas yra sudarytas ieškomo objekto modelis ir

apmokytas klasifikatorius. Šie duomenys toliau bus naudojami vykdant objekto aptikimą roboto darbo metu.

3.6.2. Ieškomo objekto aptikimas vaizde

Šiam etapui pasirinkta naudoti ir išbandyti keturis objektų aptikimo metodus Haar, HOG, TLD ir LINE-MOD. Paieškos metu dirba vienas iš nurodytų metodų. Sistemos veikimo metu iš kameros gaunami vaizdo kadrai yra paduodami naudojamam metodui (klasifikatoriui), kuris nustato ar objektas yra kadre ir kurioje jo vietoje. Klasifikatorius grąžina labiausiai tikėtiną stačiakampį regioną vaizde, kuriame gali būti ieškomas objektas. Gavus šią informaciją galima atlikti jos patikslinimą įvertinant gylio informaciją.

3.6.3. Pozicijos nustatymas naudojant gylio informaciją

Aptikto objekto pozicija gali būti nustatoma įvertinant gylio informaciją iš stereo vaizdo kamerų. Turint įprastines RGB vaizdo kameras jų sistema prieš naudojimą turi būti sukalibruojama. Tai galima atlikti naudojant standartines *OpenCV* bibliotekos funkcijas ir šachmatinės tekstūros kalibravimo lentą. Bet kurio vaizdo taško (u, v) koordinatės su erdvės koordinatėmis sieja (1) lygtis. Bendru atveju po kalibracijos yra sužinomi kamerų vidiniai (*angl. intrinsic*) parametrai (2) formulė ir išoriniai (*angl. extrinsic*) parametrai (3) formulė (t.y. kameros transformacijos matrica M).

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = AM \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T \quad (1)$$

$$A = \begin{bmatrix} mf_x & \gamma & c_x \\ 0 & mf_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

čia m – daugiklis, f_x ir f_y – židinio nuotolis x bei y ašyse, c_x ir c_y – vaizdo centro koordinatės bei γ – šlyties koeficientas tarp x ir y ašių.

$$M = [R | t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \quad (3)$$

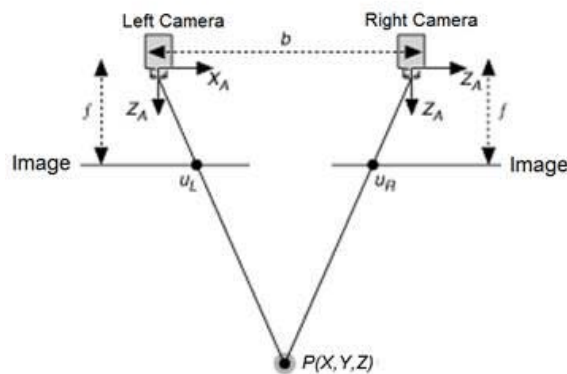
Pasinaudojus stereo kamerų savybėmis galima išgauti bet kurio kadro pikselio gylio informaciją pagal trianguliacijos principą (30 pav.). Žinant atstumą tarp kamerų – b , fokusavimo atstumą – f ir vaizdo centro koordinatės – $(c_1$ ir $c_2)$ ant projektavimo plokštumos, pagal trianguliacijos principą galima apskaičiuoti gylį Z pagal (4) formulę.

$$Z = \frac{bf}{d} = \frac{bf}{(U_L - U_R)} \quad (4)$$

čia d (*angl. disparity*) – yra horizontalus atstumas tarp sutampančių pikselių kamerų vaizduose. Didžiausia problema yra būtent sutampančių pikselių radimas. Nuo šio rezultato tikslumo labai priklauso ir gaunama gylio reikšmė, kadangi b ir f yra tik kamerai būdingi koeficientai. Siekiant palengvinti užduotį, naudojamas kalibravimas, taip sumažinant vertikalias paklaidas ir toliau dirbama su kalibruotais (*angl. rectified*) vaizdais. Taikomi įvairūs lokalūs ir globalūs sutapdinimo (*angl. matching*) metodai d reikšmių skaičiavimui, kaip SAD (5) formulė, SSD, SGBM, GC ir pan., kurie realizuoja optimizuojančius algoritmus.

$$d(x, y) = \arg \min_{0 \leq d < d_{max}} \sum_{(i, j) \in W} |L(x+i, y+j) - R(x+i-d, y+j)| \quad (5)$$

čia W – paieškos langas, L ir R – kairės ir dešinės pusės kamerų vaizdai, d – *disparity* reikšmė, d_{max} – maksimali d riba.



30 pav. Gylio skaičiavimas naudojant stereo kamerų sistemą

Naudojant Microsoft Kinect, kitą gylio kamerą ar lazerinę sistemą šių skaičiavimų nebereikia atlikti – gauname iškart naudojimui tinkamus gylio duomenis. Turint ribotus skaičiavimo resursus tai ypač svarbu, todėl tai buvo viena iš priežasčių, kodėl buvo pasirinkta eksperimentuose naudoti Kinect kamerą.

3.6.4. Pilnos objekto transformacijos nustatymas

Pasinaudojant gylio informacija kiekvienam vaizdo kadro taškui yra nustatoma jo 3D pozicija, t.y. gaunamas vaizdo taškų masyvas/debesis (*angl. point cloud*). PointCloud bibliotekos segmentavimo funkcijomis iš taškų masyvo yra išskiriamas objektas:

1. Nustatoma vaizde dominuojanti plokštuma, ant kurios yra padėtas objektas RANSAC metodu
2. Nesantys arti plokštumos taškai yra pašalinami (filtruojami) ir toliau nagrinėjami tik aptikto objekto regionui priklausantys taškai

3. Pritaikomas Euklidinio grupavimo (*angl. clustering*) metodas taškų grupei ant plokštumos surasti
4. Naudojamas ICP algoritmas sulygiuoti turimą taškų modelį su taškų grupe, gauta transformacijos matrica ir yra laukiamas rezultatas (esant kelioms taškų grupėms pasirenkama ta, kurios ICP konvergavimo rezultatas/paklaida yra mažiausia).

3.6.5. Programinės realizacijos priemonės

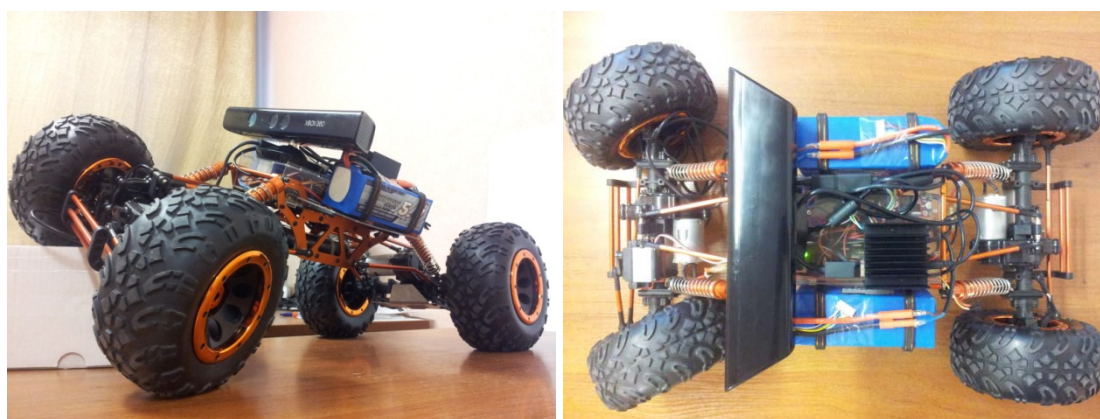
Sprendimo realizacijai pasirinktos realizavimo priemonės aprašomos toliau 3 lentelėje.

3 lentelė. Programinės realizacijos priemonės

Priemonės	Apibūdinimas
<i>OpenCV, Arduino, ARToolkit+, PointCloud bibliotekos</i>	Atviro kodo vaizdo analizės bibliotekos, realizuojančios dažniausiai naudojamus vaizdo padorojimo algoritmus ir metodus.
<i>C++, GNU GCC 4.2 kompiliatorius</i>	Objektinė programavimo kalba plačiai paplitusi spartos reikalaujantiems sprendimams realizuoti.
<i>Debian Wheezy Linux OS</i>	Kūrimo ir realizavimo operacinės sistemos platforma.

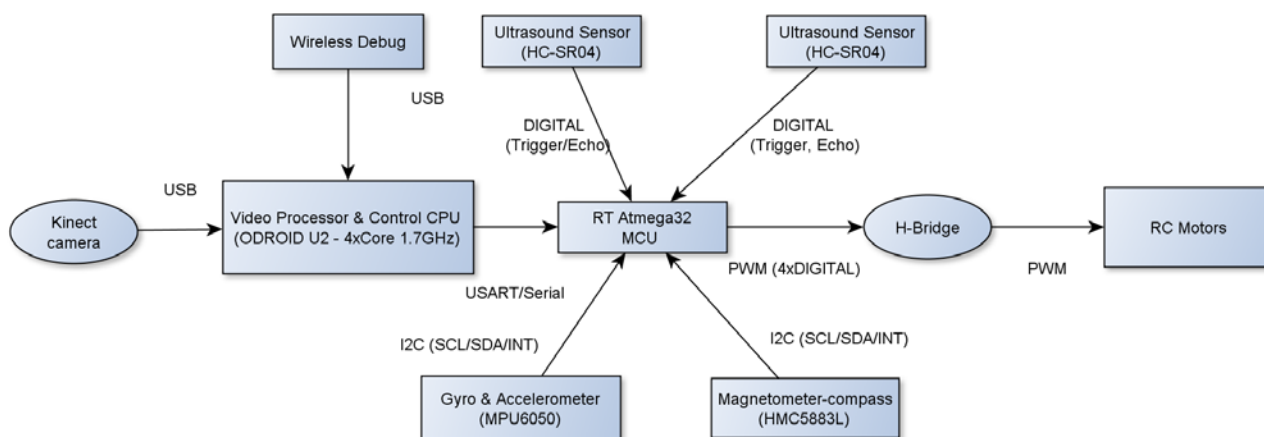
3.6.6. Techninės realizacijos priemonės

Ekspirimentinė dalis buvo atliekama su tikra roboto sistema. Pasirinktas 1:8 mastelio roboto karkasas–šasi: *HSP-Rock Crawler* (645x460x225 mm) ant kurio patalpinta visa elektroninė įranga ir jutikliai (31 pav.).



31 pav. 1/8 HSP Rock Crawler karkasas-šasi




Sistemą sudaro techninė įranga kaip pavaizduota žemiau esančiame 32 paveiksle. Darbo realizacijos testavimui ir įvertinimui pasirinkta hibridinė minimali *Arduino* platforma, kuri plačiai paplitusi tarp elektronikos mėgėjų bei turi dažnai naudojamų jutiklių valdymo bibliotekas. Arduino bibliotekos buvo naudojamos programuojant AVR Atmega32 mikrovaldiklį. Sistemos testavimui buvo naudojamas WiFi ryšys, kuriuo persiūsta valdymo ir derinimo informacija iš kompiuterio ir atgal.



32 pav. Roboto sistemos techninė realizacija

Sistemos techniniai komponentai detalizuoti ir aprašyti 4 lentelėje.

4 lentelė. Sistemos komponentai

Komponentas	Aprašymas	Pasirinkimo kriterijai
 <p><i>Odroid-U2 QuadCore 1.7GHz, 2GB RAM, 32GB microSD</i></p>	<p><i>Pagrindinis vaizdo apdorojimo ir analizės procesorius veikiantis su Debian Wheezy Linux OS.</i></p>	<p><i>Sparta, kaina, maži matmenys, nedidelės energijos sąnaudos – 10W.</i></p>
 <p><i>Microsoft Kinect</i></p>	<p><i>Vaizdo ir gylio kamera.</i></p>	<p><i>Pakankama vaizdo raiška - 0.3Mpx 640x480, kaina, sukalibruota gamykloje, geri tvirtinimo elementai.</i></p>
 <p><i>Wireless USB adapter TP-Link TL-WN725N</i></p>	<p><i>Modulis sistemos komunikacijai su kompiuteriu ir kitais įrenginiais.</i></p>	<p><i>Kaina, maži matmenys, WiFi IEEE 802.11b/g/n standarto palaikymas.</i></p>
<p><i>Atmega32 AVR</i></p>	<p><i>Mikroprocesorius bendravimui su išoriniais jutikliais.</i></p>	<p><i>Veikia Arduino platformos bibliotekos, atskiria techninės įrangos valdymą į aukštesnį sluoksnį.</i></p>

		
<p><i>Du HC-SR04 jutikliai</i></p> 	<p><i>Ultragarso jutikliai aplinkos kliūtims aptikti.</i></p>	<p><i>Kaina, paprastumas naudotis, yra Arduino bibliotekos.</i></p>
<p><i>Akselerometras ir giroskopas (MPU6050 / MMA7361)</i></p> 	<p><i>Leidžia nustatyti ir sekti roboto judėjimo greitį bei orientaciją aplinkoje.</i></p>	<p><i>Kaina, paprastumas naudotis, yra Arduino bibliotekos.</i></p>
<p><i>Magnetometras (HMC5883L)</i></p> 	<p><i>Leidžia nustatyti ir sekti orientaciją aplinkoje įvertinant Žemės magnetinį lauką.</i></p>	<p><i>Kaina, paprastumas naudotis, yra Arduino bibliotekos.</i></p>
<p><i>H-bridge</i></p> 	<p><i>Pagamintas H tiltelis – motorų valdymo mikrovaldiklio signalais schema.</i></p>	<p><i>Netiko tradicinės roboto motorų valdymo radijo bangomis priemonės.</i></p>
<p><i>Valdymo plokštė</i></p> 	<p><i>Suprojektuota ir pagaminta valdymo plokštė.</i></p>	<p><i>Reikėjo optimalaus varianto jutikliams suintegruoti vienoje vietoje dėl ribotos vietos.</i></p>

Sąsajos su komponentais ir komunikavimui naudojami protokolai aprašomi 5 lentelėje.

5 lentelė. Sąsajos ir protokolai

Sąsaja/protokolas	Tarp komponentų	Paskirtis
USB	Vaizdo apdorojimo procesoriaus, kamerų bei WiFi sąsajos.	Video informacijos gavimui-perdavimui. Derinant bevielį ryšį naudojamas duomenims ir komandoms perduoti.
USART	Vaizdo procesoriaus ir Atmega mikrokontrolerio.	Motorų valdymo komandoms perduoti, jutiklių informacijai paimti.
I ² C	Akselerometro-giroskopo, magnetometro ir Atmega.	Jutiklių valdymas ir jų informacijos paėmimas.
PWM	H-tiltelio, Atmega ir RC motorų.	Motorų sukimosi krypties ir paduodamos galios valdymui.
Kita skaitmeninė	Ultragarso modulių ir Atmega	Ultragarso modulių valdymui ir informacijos paėmimui.

3.6.7. Sprendimo testavimo apimtis

Sprendimo programinės įrangos kokybės užtikrinimo etapai:

1. **Funkcinis testavimas** – apima visus funkcinius ir nefunkcinius atvejus, testuojamos pagrindinės funkcijos ir ar sistemos veikimas atitinka jai išskeltus reikalavimus.
2. **Integracinis testavimas** – atliekamas kūrimo bei derinimo metu sujungiant keletą modulių tarpusavyje bei po visų komponentų apjungimo. Tikrinama kaip veikia daugelis funkcijų ir komponentų vienu metu.
3. **Spartos testavimas** – reikalinga, jog sistema veiktų kiek galima greičiau ir atitiktų išskeltus nefunkcinius spartos reikalavimus, todėl įvairiems testiniams atvejams bus skaičiuojamas algoritmų veikimo laikas ir renkama statistika analizei. Surinkta informacija bus naudojama algoritmų veikimo tobulinimui ir optimizacijai.
4. **Stabilumo ir atstatymo testavimas** – siekiama patikrinti kaip sprendimas veikia dirbant ilgą laiko tarpą bei ar praradus ryšį su darbu reikalingais resursais sistema išlieka stabili, o jiems tapus prieinamiems – sugeba atsistatyti ir tęsti darbą toliau, tai ypač svarbu autonominėse sistemose.

Kur įmanoma, bus naudojamas automatinis testavimas ir rezultatų validavimas. Pagal sudarytą testavimo planą bus atliekami jame nurodyti testai. Kiekvienam testui aprašytas testavimo atvejis su pradinių sąlygų, veiksmų eigos bei laukiamų rezultatų informacija.

3.6.8. Testavimo planas

Sprendimo programinės įrangos kokybės užtikrinimui bus atliekami toliau išvardinti testai.

TEST ID: T01 – OBJEKTO MODELIO APMOKYMAS (5, 6, 7, 8, 9, 11)

Aprašymas

Patikrinti ar objekto apmokymo sistema yra veikianti.

Pradinės sąlygos

- Sistema neapmokyta naujam tiriamam objektui

Veiksmai

- Sistema paleidžiama apmokymo režime
- Prieš kameras rodomas naujas nematytas objektas kol sudaromas modelis

Laukiami rezultatai

- Nėra klaidų sisteminiame žurnale
- Sistema pradeda ir baigia darbą sėkmingai be gedimų
- Apmokymo režime naudojami kamerų jutikliai informacijai įvesti
- Sistema informuoja apie apmokymo pabaigą
- Sistema po apmokymo išsaugo sudaryto objekto modelį

TEST ID: T02 – OBJEKTO PAIEŠKA (1, 2, 3, 7, 12)

Aprašymas

Patikrinti ar objekto paieška yra veikianti.

Pradinės sąlygos

- Sukomplektuotas, techniškai tvarkingas ir važiuojantis robotas
- Ieškomas objekto modelis sukurtas naudojant apmokymo funkciją

Veiksmai

- Padėti ieškomą objektą robotui prieinamoje vietoje darbo aplinkoje
- Paleisti robotą ieškoti objekto pagal modelį ir stebėti jo veiksmus
- Aptikimo vaizdą stebėti derinimo priemonėmis

Laukiami rezultatai

- Nėra klaidų sisteminiame žurnale ar roboto veikime
- Robotas sėkmingai aptinka objektą
- Roboto veiksmai nuoseklūs, objektas nepametamas
- Sistema pradeda ir baigia darbą sėkmingai be gedimų
- Objektas identifikuojamas ir pažymimas vaizdo kadre

TEST ID: T03 – OBJEKTO POZICIJOS NUSTATYMAS (3, 4, 5, 7, 12)

Aprašymas

Patikrinti, ar sistema sugeba nustatyti objekto poziciją aplinkoje.

Pradinės sąlygos

- Sukomplektuotas, techniškai tvarkingas ir važiuojantis robotas
- Ieškomas objekto modelis sukurtas naudojant apmokymo funkciją
- Ieškomas objektas padėtas tiksliose ir žinomose koordinatėse

Veiksmai

- Paleisti robotą ieškoti ir nustatyti objekto poziciją
- Patikrinti ar pateikiama pozicija atitinka realias koordinates

Laukiami rezultatai

- Nėra klaidų sisteminiame žurnale ar roboto veikime

- Robotas sėkmingai pamato, randa objektą ir nustato jo koordinates
- Sistema pradeda ir baigia darbą sėkmingai be gedimų

TEST ID: T04 – OBJEKTO ORIENTACIJOS NUSTATYMAS (4, 5, 7, 10, 12, 13)

Aprašymas

Patikrinti, ar sistema sugeba nustatyti objekto orientaciją aplinkoje.

Pradinės sąlygos

- Sukomplektuotas, techniškai tvarkingas ir važiuojantis robotas
- Ieškomas objekto modelis sukurtas naudojant apmokymo funkciją
- Ieškomas objektas padėtas tiksliose ir žinomose koordinatėse bei pasuktas žinomu kampu

Veiksmai

- Paleisti robotą ieškoti ir nustatyti objekto poziciją
- Patikrinti ar objektas aptinkamas ir ar pateikiama orientacija atitinka realias kampu posūkių reikšmes

Laukiami rezultatai

- Nėra klaidų sisteminiame žurnale ar roboto veikime
- Robotas sėkmingai aptinka objektą ir nustato jo koordinates bei orientaciją
- Sistema pradeda ir baigia darbą sėkmingai be gedimų
- Nustatyti posūkio kampai neviršija 5 laipsnių kampą

TEST ID: T05 – GREITAVEIKOS TIKRINIMAS (14)

Aprašymas

Patikrinti, ar sistemos algoritmų veikimo laikas neviršija numatytos leistinos ribos.

Pradinės sąlygos

- Sukomplektuotas, techniškai tvarkingas ir važiuojantis robotas
- Ieškomas objekto modelis sukurtas naudojant apmokymo funkciją
- Objektas padėtas robotui matomoje vietoje
- Laiko matavimo taškai sudėti kode

Veiksmai

- Paleisti robotą ieškoti ir nustatyti objekto poziciją
- Stebėti derinimo režimo pateikiamą informaciją

Laukiami rezultatai

- Nėra klaidų sisteminiame žurnale ar roboto veikime
- Robotas sėkmingai aptinka objektą ir nustato jo koordinates bei orientaciją
- Sistema pradeda ir baigia darbą sėkmingai be gedimų
- Greitaveikos statistika spausdinama derinimo žurnale
- Algoritmų veikimo laikas neviršija numatytojo

TEST ID: T06 – NEŽINOMOS APLINKOS (15)

Aprašymas

Patikrinti, ar robotas vienodai gerai veikia skirtingose aplinkose.

Pradinės sąlygos

- Sukomplektuotas, techniškai tvarkingas ir važiuojantis robotas
- Ieškomas objekto modelis sukurtas naudojant apmokymo funkciją
- Paruoštos kelios bandymo aplinkos robotui

- Objektas bandymų aplinkoje padėtas robotui matomoje vietoje

Veiksmai

- Paleisti robotą ieškoti ir nustatyti objekto poziciją pirmoje aplinkoje
- Fiksuoti veikimo laiką iki suradimo
- Pabaigus, robotą pernešti į kitą aplinką ir pakartoti eksperimentą

Laukiami rezultatai

- Nėra klaidų sisteminiame žurnale ar roboto veikime
- Robotas sėkmingai aptinka objektą ir nustato jo koordinates bei orientaciją
- Sistema pradeda ir baigia darbą sėkmingai be gedimų
- Veikimo laikai apytiksliai vienodi visose aplinkose

TEST ID: T07 – BET KOKIE OBJEKTAI (16)

Aprašymas

Patikrinti, ar robotas vienodai gerai aptinka ir suranda skirtingus objektus.

Pradinės sąlygos

- Sukomplektuotas, techniškai tvarkingas ir važiuojantis robotas
- Paruošti keli bandymų objektai (vienspalviai, raštuoti ir kt.)
- Ieškomas objekto modelis sukurtas naudojant apmokymo funkciją
- Objektas bandymų aplinkoje padėtas robotui matomoje vietoje

Veiksmai

- Paleisti robotą ieškoti objekto
- Laukti, kol objektas bus surastas
- Pabaigus, robotą apmokyti su kitu objektu ir pakartoti eksperimentą

Laukiami rezultatai

- Nėra klaidų sisteminiame žurnale ar roboto veikime
- Robotas sėkmingai aptinka objektą ir nustato jo koordinates bei orientaciją
- Sistema pradeda ir baigia darbą sėkmingai be gedimų
- Visi ieškomi objektai randami

3.7. Apibendrinimas ir išvados

Sudarytas darbo projektas leido atidžiau pažvelgti į realizacijai reikalingas priemones ir metodus. Geriau suplanuoti realizacijos etapus, atrasti trūkstamus elementus. Išsikelti kokybės kriterijai ir reikalavimai leido pasiekti norimą ir užsibrėžtą darbo kokybę, o pasirinktos realizacijos priemonės padėjo efektyviau išnaudoti darbui skirtą laiką. Sprendimo statinio ir dinaminio lygių detalizavimas UML diagramomis leido susidaryti konkretesnį sistemos modelį realizacijai, o testavimo planas – vėliau patikrinti plano ir realizacijos teisingumą.

Apibrėžus sistemos ribas, reikalavimus ir projektavimo gaires buvo numatyta toliau kitame etape plėsti ir patobulinti projektą, detalizuoti mažiausiai aptartas sritis, realizuoti sistemą, tobulinti techninę įrangą eksperimentui bei ruošti paties eksperimento vykdymui. Sprendimo tobulinimas yra viena iš prioritetinių kryptių tolimesnio darbo plėtojimui.

4. EKSPERIMENTINĖ DALIS

4.1. Eksperimentų planavimas

Eksperimentų tyrimo eiga buvo suskaidyta į tokius etapus:

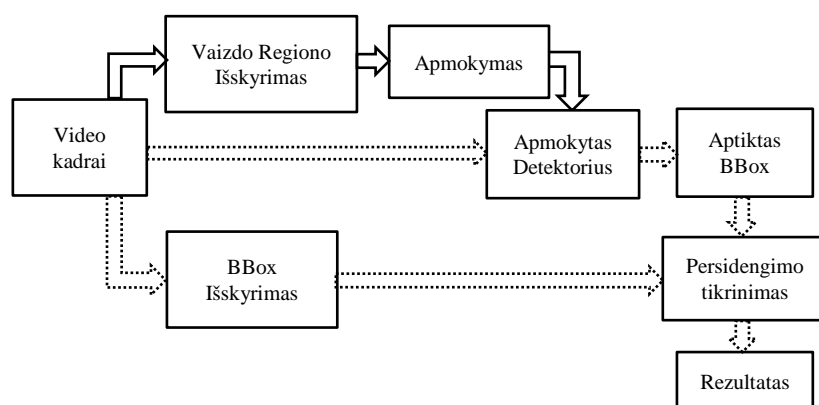
1. Objektų aptikimo metodų tyrimas
2. Objektų orientacijos nustatymo metodų tyrimas

Kiekvienas iš etapų toliau apžvelgiamas atskirai pateikiant jų vertinimo metodiką.

4.1.1. Objektų aptikimo metodų tyrimas

Šiame darbo etape buvo tiriami metodai skirti objektų atpažinimui iš RGB vaizdo. Siekta išsiaiškinti efektyviausią algoritmą šiai užduočiai atlikti. Tyrimo metu buvo analizuojami keturi objektų aptikimo algoritmai – HOG, Haar, TLD ir LINE-MOD. Naudota tyrimo metodika aprašoma toliau.

Pirmiausia, siekiant realizuoti kuo įvairesnius ir kiek įmanoma pilnus testus – detektorių apmokymui ir testavimui buvo pasirinkti įvairūs objektai, pagal formos, dydžio ir tekstūros kriterijus – su tekstūra ir be, vienspalviai ir kelių spalvų, nesimetriniai. Darbe įgyvendinta detektorių efektyvumo nustatymo eiga pavaizduota 33 pav. Apmokymo eiga nurodyta vientisomis rodyklėmis, o testavimo punktyrinėmis. Apmokymo etape kiekvienam pasirinktam objektui buvo sugeneruoti detektorių apmokymo duomenys – naudojant Kinect kamerą sukurtas vienodos trukmės spalvos ir gylio duomenų vaizdo įrašas. Įrašai atlikti įvairiai keičiant objekto mastelį, kameros poslinkį, pasisukimo ir matymo kampą. Iš kiekvieno vaizdo įrašo kadro rankiniu būdu buvo iškirpti detektorių apmokymo vaizdai – objektą gaubiantys stačiakampiai (*angl. bounding box*).



33 pav. Objektų detektorių apmokymo ir testavimo procedūra

Testavimo procedūra atlikta analogišku principu. Kiti vaizdo įrašai skirtingose aplinkose buvo atlikti visiems tiriamiems objektams, įvairiai keičiant kameros pasisukimą, poslinkį, matymo kampą ir pan. Kiekviename naujo vaizdo įrašo kadre rankiniu būdu buvo pažymėtas objektą gaubiantis stačiakampis. Toliau tie patys vaizdo kadrai buvo paduodami apmokytam objektų detektoriui.

Vertinant aptikimo tikslumą detektoriaus nustatytas gaubiantis stačiakampis ir tikrasis gaubiantis stačiakampis buvo palyginami naudojant persidengimo santykį kaip pateikta (6) formulėje.

$$\omega(B_1, B_2) = \frac{B_1 \cap B_2}{B_1 \cup B_2} > T \quad (6)$$

čia T – teisingo nustatymo slenksčio reikšmė (testuose - 0.5), B_n – kiekvieno stačiakampio plotas (kur $n = 1..2$). Taip pat testavimo metu buvo matuojamas detektorių vieno kadro apdorojimo laikas.

4.1.2. Objektų orientacijos nustatymo metodų tyrimas

Šiame darbo etape buvo tiriami metodai skirti objektų orientacijos nustatymui panaudojant stereo kamerų gylio informaciją. Palygintos dvi technologijas tarpusavyje – įprastinė stereo kamerų sistema ir Microsoft Kinect įrenginys bei išsiaiškintas efektyviausią metodą tolesniam objektų orientacijos nustatymo tyrimui.

Tyrimo eiga buvo suskirstyta taip:

1. Stereo kamerų gylio nustatymo algoritmų spartos ir tikslumo tyrimas
2. Stereo kamerų sistemos ir Kinect įrenginio tikslumo tyrimas
3. Objekto orientacijos nustatymo iš taškų debesies (*angl. point cloud*) panaudojant ICP algoritmą spartos ir tikslumo tyrimas

4.1.2.1. Stereo kamerų gylio nustatymo algoritmų tyrimo metodika

Atliekant gylio nustatymo metodų iš stereo kamerų vaizdų tyrimą buvo analizuoti keturi – du OpenCV bibliotekos (1 ir 2) bei du savarankiškai realizuoti algoritmai (3 ir 4):

1. BM/SGBM (*angl. block matching / semiglobal block matching*)
2. GC (*angl. graph cut*)
3. SAD (*angl. Sum of Absolute Differences*)
4. SSD (*angl. Sum of Squared Differences*)

Pateikti metodai iš RGB vaizdų poros (kairės ir dešinės kamerų) leidžia išskirti užfiksuoto vaizdo gylio reikšmes kiekviename jo taške. Šių algoritmų vertinimo kriterijai pateikiami 6 lentelėje.

6 lentelė. Stereo gylio metodų palyginimo kriterijai

Kriterijus	Aprašymas
Sparta	Matuojamas algoritmo veikimo laikas apdorojant vieną stereo kadrų porą.
Tikslumas	Procentinis klaidingų pikselių skaičius lyginant su tikru etaloniniu gylio vaizdu.

Gylio vaizdų klaidingų pikselių procentas buvo skaičiuojamas lyginant pikselių gylių skirtumą pagal formulę:

$$D_{err} = \frac{100}{wh} \sum_{\substack{0 \leq x < w \\ 0 \leq y < h}} f(x,y), \text{ kur } f(x,y) = \begin{cases} 0, & |d(x,y) - D(x,y)| \leq \varepsilon \\ 1, & \text{priešingu atveju} \end{cases}$$

čia D_{err} – klaidingų pikselių skaičius procentais, w ir h – paveikslo plotis ir aukštis, $d(x,y)$ – pikselio gylio reikšmė taške $(x; y)$, $D(x,y)$ – etaloninio paveikslo gylio reikšmė taške $(x; y)$, ε – maksimali leistina gylio paklaida.

4.1.2.2. Stereo kamerų sistemos ir Kinect įrenginio tikslumo tyrimas

Gavus gylio algoritmų tikslumo ir spartos rezultatus, toliau buvo siekiama palyginti dviejų technologijų – stereo kamerų sistemos ir Microsoft Kinect efektyvumą tarpusavyje, todėl buvo atliktas gylio nustatymo tikslumo tyrimas.



34 pav. Stereo kamerų sistema

Eksperimentams naudota viename lygyje fiksuota dviejų VGA 640x480 raiškos Prestigio PWC 2 kamerų sistema ir Kinect įrenginys kaip pavaizduota 34 paveiksle. Stereo kamerų optinės ašys buvo sulygiuotos lygiagrečiai, o tarp optinių centrų paliktas 3.5 cm atstumas. Taip pat kamerų suderinimui naudotas vertikalus vaizdų sulygiavimas. Siekiant išvengti netikslumų įnešamų dėl optinės sistemos lęšio tangentinių ir radialinių iškraipymų, buvo atliekamas kalibravimas panaudojant 20 vaizdo porų (35 pav.) bei OpenCV kalibravimo algoritmus. Nustatinėjant gylį iš paveikslų ir juos lyginant, tai itin svarbu. Gauta vidutinė kalibravimo paklaida – 0.506 px.

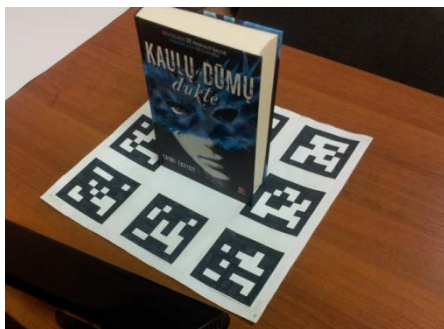


35 pav. Kamerų kalibravimas panaudojant šachmatinę plokštumą

Testų metu žinomu atstumu prieš kiekvieną gylio kamerą buvo pastatomas 60x45cm objektas lygiu plokščiu paviršiumi. Panaudojant SGBM bei GC metodus iš stereo sistemos gautų vaizdų buvo apskaičiuojamas gylio vaizdas. Iš gauto rezultato buvo nustatomas vidutinis atstumas iki

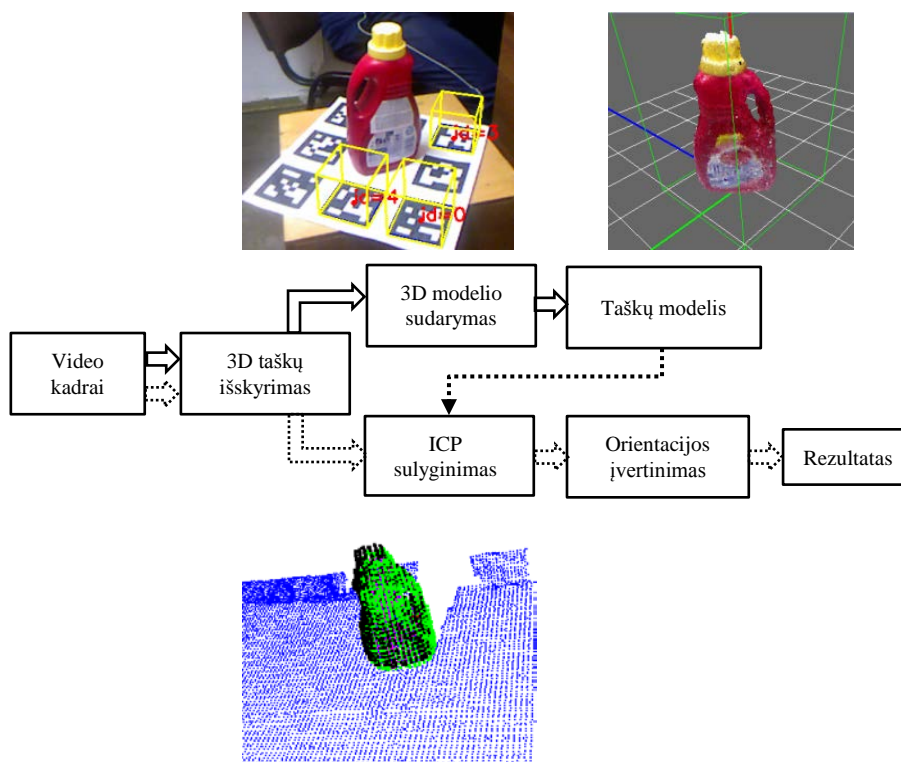
plokštumos. Tai atliekama pasirenkant stabilų regioną gylio vaizde sutampantį su to objekto plokštuma ir paskaičiuojant to regiono gylio vidurkį.

4.1.2.3. Objekto orientacijos nustatymo iš gylio informacijos tyrimas



36 pav. Objekto orientacijos ir pozicijos nustatymo tikslumo tikrinimas

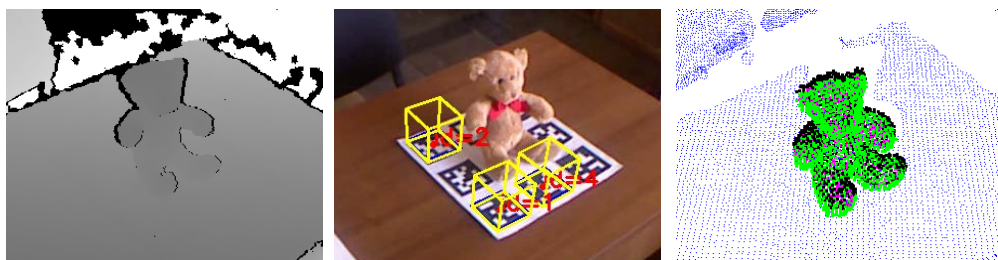
Pozicijai ir orientacijai įvertinti buvo naudojama specialiai paruošta testavimo aplinka, kurioje skirtingi objektai pastatomi reikiamu kampu bei išlygiuojami ant žymekliais pažymėtos plokštumos (36 pav.). Buvo vertinama, ar objektas ir jo modelis yra teisingai sulygiuoti – skaičiuojama orientacijos nustatymo kampinė paklaida bei atstumo nuo nustatytos pozicijos iki tikrosios pozicijos Euklidinis skirtumas.



37 pav. Modelio sudarymo ir orientacijos įvertimo procedūra

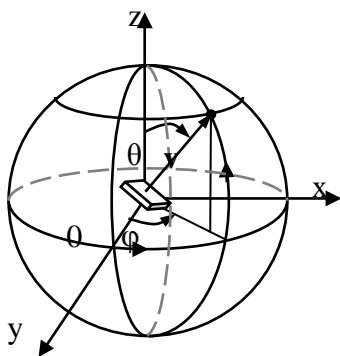
Pagal procedūrą pateiktą 37 pav. buvo vertinama ICP algoritmo sparta ir tikslumas. Ji sudaryta iš dviejų etapų – apmokymo ir orientacijos tikslumo įvertinimo. Prieš atliekant eksperimentus kiekvienam objektui buvo sukurtas jo taškų (*angl. point cloud*) modelis apmokymo

etape (vientisos rodyklės). Tai buvo atlikta naudojant Kinect kamerą, ARToolkit+ bei PointCloud bibliotekas ir žymekliais paremtą techniką kaip aprašyta [29] darbe. Testavimo etape (punktyrinės rodyklės) kiekvieno objekto, pastatyto ant žymeklių plokštumos, 1 minutės trukmės vaizdo ir gylio įrašas buvo padarytas su Kinect kamera iš skirtingų padėčių ir atstumų erdvėje. Toliau analizuojant kiekvieną vaizdo ir gylio įrašo kadrą buvo išskiriami to vaizdo 3D taškai. Naudojant PointCloud bibliotekos plokštumos nustatymo ir Euklidinio grupavimo (*angl. clustering*) algoritmus iš taškų aibės buvo išskiriami objekto taškai.



38 pav. Gylio vaizdas ir teisingai nustatyta objekto orientacija

Pasitelkus ICP algoritmą anksčiau sukurtas objekto taškų modelis ir vaizdo objekto taškai buvo sulygiuojami kaip pavaizduota 38 pav., o ICP nustatyta transformacija toliau panaudota orientacijos įvertinimui. Testavimo metu žymeklių plokštuma buvo naudojama tik tikrajai objekto orientacijai nustatyti. Po sulygiavimo tikroji ir nustatyta transformacijos buvo palygintos tarpusavyje pagal žemiau pateiktą metodiką.



39 pav. Kiekvienas taškas ant vienietinės sferos turi unikalias (φ, θ) koordinatas

$$\varphi(\mathbf{p}) = \arctan\left(\frac{p_y}{p_x}\right) \quad (7)$$

$$\theta(\mathbf{p}) = \arccos(p_z) \quad (8)$$

$$\delta(\mathbf{p}_1, \mathbf{p}_2) = \{\varphi_{p1} - \varphi_{p2}, \theta_{p1} - \theta_{p2}\} \quad (9)$$

Turint tikrąją modelio transformacijos matricą $\mathbf{M}_1=[\mathbf{R}_1|\mathbf{t}_1]$ ir sulygiuoto modelio transformacijos matricą $\mathbf{M}_2=[\mathbf{R}_2|\mathbf{t}_2]$ (kur \mathbf{R}_i – yra 3x3 posūkio matrica, o \mathbf{t}_i – 3x1 poslinkio vektorius) norime sužinoti sulygiavimo teisingumą. Išskiriant Oilerio kampus iš posūkio matricų \mathbf{R}_1 bei \mathbf{R}_2 ir juos lyginant gausime nevienareikšmiškus rezultatus, kadangi yra daug būdų kaip sukompnuoti tas pačias posūkio matricas naudojant skirtingus Oilerio kampus [30], todėl sferinių

koordinaciu metodas buvo panaudotas šiame darbe posūkiu matricų palyginimui. Kiekviena posūkiu matrica yra unikaliam identifikuojama sferiniu koordinaciu pora (ϕ , θ) ant vienetinio spindulio sferos kaip parodyta 39 pav. Taigi, galutinis orientacijos įvertinimas buvo atliktas taip:

1. Transformuotas vienetinis vektorius $\mathbf{p} = \{1,0,0\}$ pagal \mathbf{R}_1 ir \mathbf{R}_2 matricas
2. Gauti taškai \mathbf{p}_1 ir \mathbf{p}_2 transformuoti į sferines koordinatas panaudojant (7) ir (8)
3. Gautos koordinatės atimtos pagal (9) formulę
4. Transformacijos matricų poslinkio dalys palygintos imant \mathbf{t}_1 ir \mathbf{t}_2 skirtumą

Orientacijos palyginimo teisingumo tikrinimui pasirinkta papildomai naudoti vertinimo metriką (10), kuri pateikiama [31].

$$\phi(\mathbf{R}_1, \mathbf{R}_2) = \|\mathbf{I} - \mathbf{R}_1 \mathbf{R}_2^T\|_F, \quad \phi \in [0, 2\sqrt{2}] \quad (10)$$

čia \mathbf{I} – vienetinė matrica, o $\|\cdot\|_F$ yra matricos Frobenius'o norma.

4.2. Atlikti eksperimentai ir rezultatai

Pagal anksčiau apibrėžtą tyrimo metodiką buvo atlikti eksperimentai. Jų rezultatai bei iš to kylančios išvalgos pateikiamos tokiu pačiu eiliškumu toliau.

4.2.1. Objektų detektorių efektyvumo tyrimo rezultatai

Objektų aptikimo algoritmų efektyvumo vertinimo eksperimentui buvo pasirinkti 8 skirtingi objektai kaip pavaizduota 40 paveiksle.

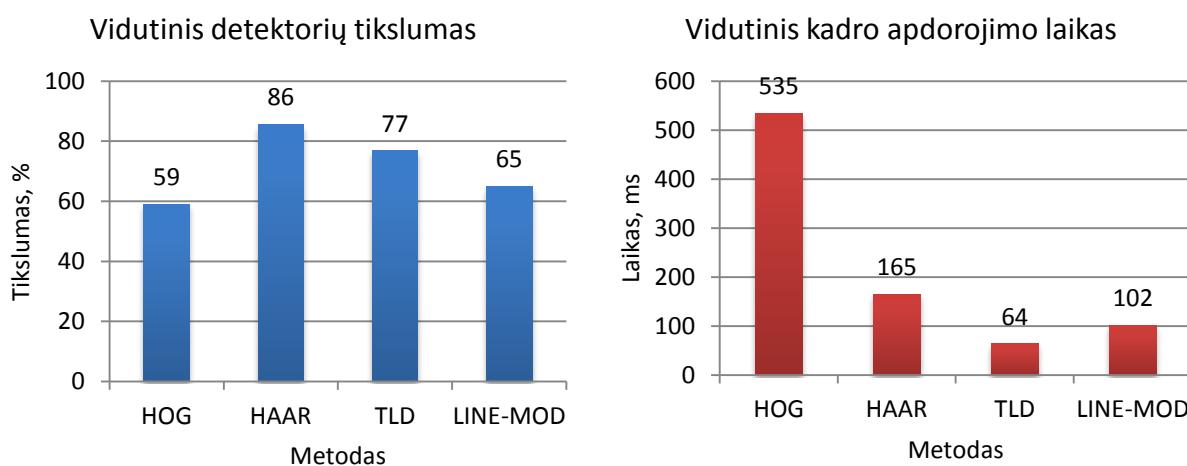


40 pav. Eksperimente naudoti objektai ir keletas kadru iš apmokymo sekos

Vienos minutės 640x480 raiškos ir 30 kadru/s RGB ir gylio vaizdo įrašas buvo padarytas naudojant Kinect kamerą kiekvienam objektui apmokyti. Iš kiekvieno įrašo rankiniu būdu buvo išskirta 1800 spalvos ir gylio vaizdų gaubiančiu stačiakampiu (*angl. bounding box*) apimančių apmokomą objektą. Tik dalis šių vaizdų buvo panaudoti automatiniam apmokymui dėl didelių apmokymo proceso atminties reikalavimų. 1300 teigiamų vaizdų ir 3200 neigiamų vaizdų buvo skirta apmokyti HOG algoritmu. Eksperimentams naudoti standartiniai OpenCV apmokymo nustatymai bei 96x96 px lango dydis. Haar metodas buvo apmokytas su 500 teigiamų ir 1500

neigiamų vaizdų lango dydį imant 24x24 px. Apmokyta 15 etapų Haar kaskada su minimaliu pataikymo dažniu – 0.999, klaidingų aptikimų dažniu – 0.5 ir 20 silpnų klasifikatorių per etapą.

Testavimas buvo atliekamas automatiškai tikrinant gaubiančių stačiakampių susikirtimus kaip aprašyta anksčiau naudojant naujus 1 minutės trukmės vaizdo įrašus. Visi testai buvo vykdomi Intel Core 2 Duo P8400 2xCore 2.26GHz CPU, 4GB RAM sistemoje. Algoritmų tikslumo bei vykdymo spartos rezultatai pateikiami 41 paveiksle.



41 pav. Objektų aptikimo tikslumo ir vykdymo spartos rezultatai

Iš testų rezultatų galime matyti, jog detektoriai gali pasiekti 60-90% aptikimo tikslumą. Haar algoritmo tikslumas artimas pateiktam 91% originaliame darbe. Taip pat matyti, jog naudojant gylio ir spalvinę informaciją (LINE-MOD) ar sekimą tarp kadro (kaip tą daro TLD) aptikimo laikas yra mažesnis. TLD rezultatas taip pat artimas paskelbtam 81% originaliame darbe, tačiau LINE-MOD tikslumas yra gerokai mažesnis nei oficialiame jo autoriaus darbe – 95%. Mažiausiai tikslus HOG algoritmas taip pat parodė ir blogiausią vykdymo spartą, tuo tarpu TLD metodas pats greičiausias iš visų. Maksimali pasiekta sparta – 15 kadro/s standartiniam VGA raiškos vaizdai.

4.2.2. Orientacijos ir pozicijos nustatymo tikslumo įvertinimas

Pagal anksčiau apibrėžtą orientacijos nustatymo tyrimo metodiką atliktų eksperimentų rezultatai pateikiami tolesniuose skyriuose.

4.2.2.1. Gylio nustatymo metodų palyginimo rezultatai

Panaudojant OpenCV biblioteką realizuota programinė įranga, kuri pagal anksčiau apibrėžtą gylio metodų tikslumo bei spartos matavimo metodiką pateikia tyrimo rezultatus. Testavimo duomenys pasirinkti iš Middlebury duomenų bazės¹. Keletas gautų rezultatų vaizdų pateikiama paveikslais (42 pav.) žemiau: pirmas stulpelis – originalus RGB vaizdas, antras – Semi Global Block Matching algoritmo (SGBM) rezultatas, trečias – SAD algoritmo, ketvirtas – SSD, penktas –

¹ <http://vision.middlebury.edu/stereo/data/>

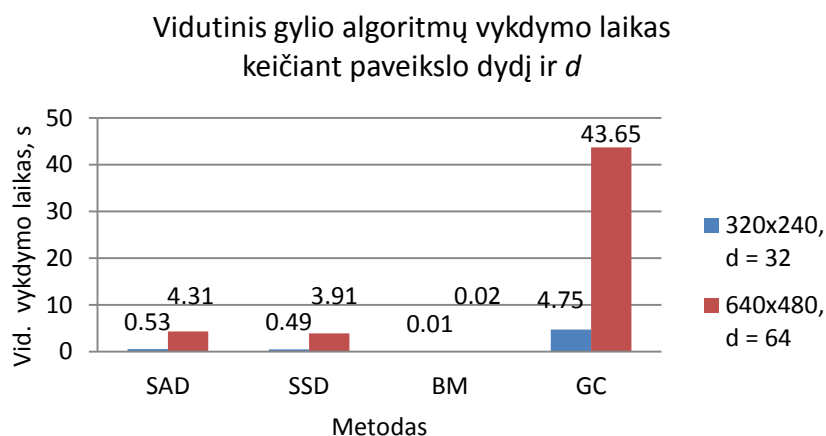
Graph Cut (GC) algoritmo ir paskutinis – etalonišės reikšmės (*angl. ground truth*) gautos lazeriniu arba struktūrinės šviesos (*angl. structured light*) matavimo būdu. Eilučių didėjimo tvarka testai išdėstyti taip: Aloe, Cones, Tsukuba, Teddy bei Art.



42 pav. Pradiniai ir skirtingų gylio metodų sugeneruoti vaizdai

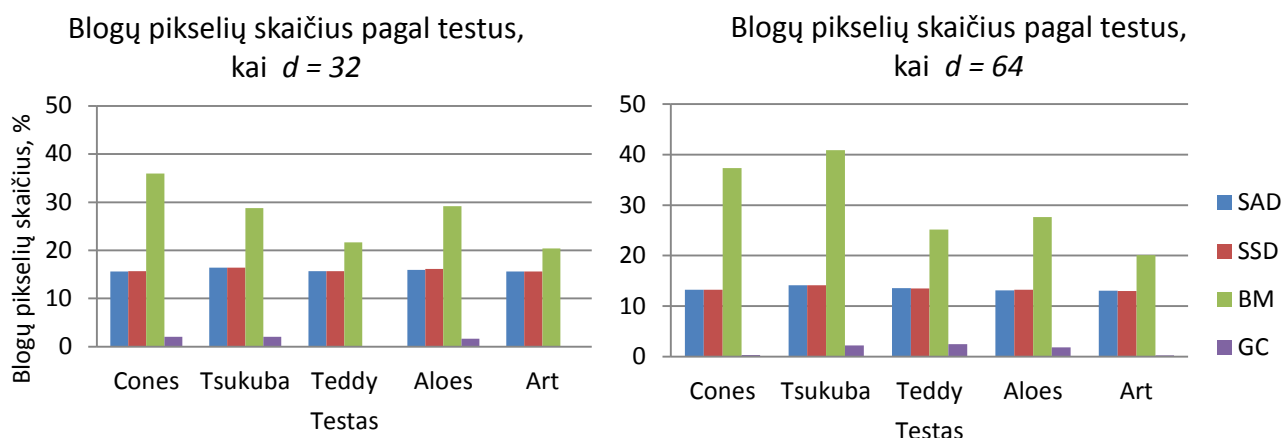
7 lentelė. Gylio metodų spartos ir tikslumo palyginimo rezultatai

Metodas	d	Vaizdas	Cones		Tsukuba		Teddy		Aloe		Art		Vid. t, s	Vid. err, %
			t, s	D_{err} , %	t, s	D_{err} , %	t, s	D_{err} , %	t, s	D_{err} , %	t, s	D_{err} , %		
SAD	32	320 x 240	0.53	16	0.53	16	0.53	16	0.53	16	0.53	16	0.53	16
SSD			0.49	16	0.48	16	0.49	16	0.49	16	0.48	16	0.49	16
SGBM			0.01	36	0.01	29	0.01	22	0.01	29	0.01	20	0.01	27
GC			4.66	2	4.21	2	5.68	0	5.26	2	3.95	0	4.75	1
SAD	64	640 x 480	4.30	13	4.32	14	4.30	14	4.33	13	4.30	13	4.31	13
SSD			3.91	13	3.92	14	3.91	13	3.92	13	3.90	13	3.91	13
SGBM			0.02	37	0.02	41	0.02	25	0.02	28	0.02	20	0.02	30
GC			41.99	0	37.68	2	52.70	2	52.58	2	33.31	0	43.65	1



43 pav. Gylio algoritmų veikimo spartos palyginimo grafikas

Iš eksperimento rezultatų (43 pav.) galima matyti, jog SGBM gylio nustatymo metodas yra sparčiausias iš analizuotų (0.006-0.024 s), tačiau tuo pačiu ir netiksliausias – blogų pikselių skaičius siekia net iki 37 % viso paveikslo dydžio. GC metodas atvirkščiai – yra labai lėtas skaičiuoti realiu laiku (3.954-52.701 s), tačiau jo kokybė yra labai gera, blogų pikselių skaičius vaizde siekia tik 2 %.



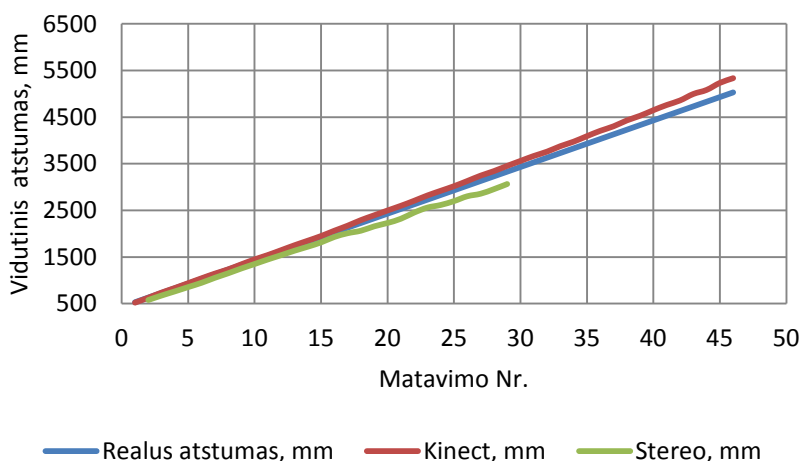
44 pav. Blogų pikselių skaičiaus nustatymo rezultatų grafikai

SAD ir SSD metodai labai panašūs savo skaičiavimų ir realizacijos prasme (44 pav.), todėl jų laiko ir tikslumo charakteristikos taip pat panašios.

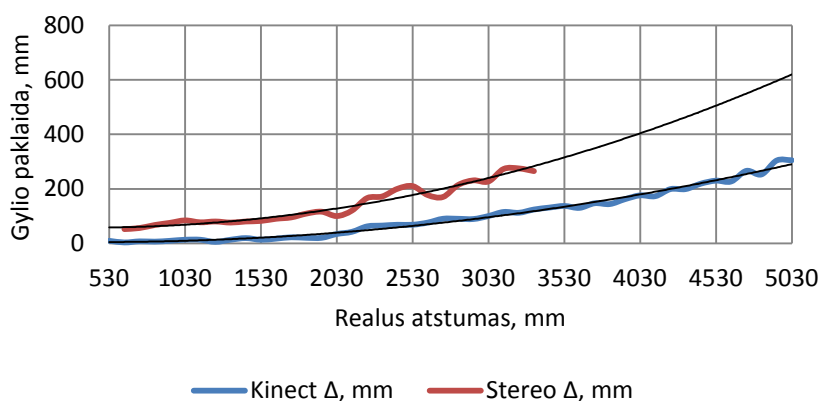
4.2.2.2. Stereo kamerų sistemos ir Kinect įrenginio tikslumo rezultatai

Tikslumo palyginimo eksperimentams skirta programinė įranga buvo realizuota panaudojant OpenCV biblioteką. Gauti rezultatai pateikiami 45 pav. Iš grafikų matyti, jog Kinect įrenginio pateikiama gylio informacija yra vidutiniškai 2-5 kartus tikslesnė ir paklaida auga lėčiau didėjant atstumui nuo kameros. Abi kreivės pavaizduotos grafike yra antros eilės polinomai.

Realaus ir išmatuoto gylio palyginimo grafikas



Gylio matavimo paklaidos priklausomybė nuo atstumo



45 pav. Stereo kamerų sistemos ir Kinect gylio vaizdų palyginimo rezultatai

Išanalizavus anksčiau atliktų tyrimų rezultatus apibendrinta gylio metodų informacija atitinkamiems įrenginiams pateikiama 8 lentelėje. Iš lentelės matyti, jog universalios sprendimo gylio informacijos duomenims išgauti nėra. Įvertinus šio darbo taikymo pobūdį ir turimus resursus tolesniuose tyrimuose buvo nuspręsta naudoti Microsoft Kinect įrenginį dėl savo tikslumo ir minimalios CPU apkrovos (leidžiančios užtikrinti maksimalią apdorojimo spartą).

8 lentelė. Apibendrintos gylio nustatymo įrenginių charakteristikos

Įrenginys	Apdorojimo sparta, ms	Kadrai, Hz	Naudojama galia, W	Efektyvus atstumas, m	Santykinis tikslumas	Reikia CPU gylio skaičiavimams	Lauko sąlygoms
Stereo kameros	20–4365	0.23–50	5 W (2 USB: 5V, 0.5A)	3.5	1	Taip	Taip
Microsoft Kinect	33	30	12W (12V, 1A) + 2.5W (1 USB)	5	2–5	Ne	Ne

4.2.2.3. Objekto orientacijos nustatymo iš gylio informacijos tyrimo rezultatai

Anksčiau pristatyti 8 objektai buvo naudojami šiame eksperimente. Visi analizuoti 640x480 vaizdo taškai (*angl. point cloud*) buvo filtruojami panaudojant 1cm vokselio filtrą prieš atliekant ICP sulygiavimą. Visuose testuose ICP algoritmui buvo nustatyta 100 iteracijų maksimali riba, minimalus $1e-3$ orientacijos konvergavimo tikslumas ir atitinkančių (*angl. correspondence*) taškų paieškos atstumo slenkstis – 4cm. Rezultatai pateikiami 9 lentelėje toliau. Φ metrikos tikslumas buvo paskaičiuotas pagal (11) formulę.

$$Acc(\phi) = \left(1 - \frac{\phi}{2\sqrt{2}}\right) \cdot 100\% \quad (11)$$

9 lentelė. ICP orientacijos ir pozicijos tikslumo vertinimo rezultatai

Testas	Pozicija (abs. vid. klaida, mm)			Posūkis (abs. vid. klaida)		Φ metrika		Laikas, ms
	X	Y	Z	$\Phi, ^\circ$	$\Theta, ^\circ$	Error	Acc. %	
Book	37.50	31.58	54.38	5.02	7.28	0.30	89.44	621.15
Bottle	23.82	49.82	43.75	12.96	28.00	0.75	73.32	496.66
Box	31.78	18.07	52.33	6.36	5.02	0.24	91.48	456.01
Car	23.77	16.94	11.05	15.87	31.12	0.95	66.35	374.93
Cup	7.24	6.50	2.20	17.96	78.11	1.82	35.52	238.21
Sheep	30.55	23.20	41.71	25.65	39.28	1.22	56.87	340.56
Shoe	18.30	22.90	7.04	11.48	17.76	0.60	78.88	404.70
Teddy	36.22	26.34	65.76	38.67	29.76	1.25	55.89	613.95
Vidurkis	26.15	24.42	34.78	16.75	29.54	0.89	68.47	443.27

Testų rezultatai rodo, jog taškų sulygiavimo ICP metodas vidutiniškai pateikia rezultatus su 3.4 cm pozicijos paklaida ir apie 30° vidutine orientacijos kampine paklaida. ICP sulyginimo metodas yra skaičiavimams imli operacija ir jo veikimo laikas nėra artimas realaus laiko apdorojimo greičiui – vidutinė kadro apdorojimo sparta 443.27ms (arba ~ 2 kadrai/s). Galima pastebėti, jog paprastesnės formos objektai (book, box) parodo geresnį tikslumą nei sudėtingesni objektai (kaip sheep, teddy).

4.3. Eksperimentų išvados

Visi suplanuoti ir aprašyti eksperimentai buvo atlikti sėkmingai, o pasirinkta vertinimo metodika pasiteisino – leisdama įvertinti skirtingus algoritmus naudojant tuos pačius tyrimo duomenis. Gauti objektų nustatymo tikslumo ir spartos rezultatai padėjo įvertinti detektorių efektyvumą bei pasirinkti optimalius algoritmus roboto sistemos realizacijai. Gylio metodų tyrimo rezultatai parodė, jog tinkamiausias sprendimas spartos ir tikslumo atžvilgiu yra gaunamas pasitelkiant Kinect kamerą. Objektų orientacijos nustatymo iš vaizdo gylio informacijos eksperimentas leido suprasti, jog ICP metodas yra tinkamas objekto transformacijos nustatymui, tačiau reikalauja daug skaičiavimo resursų ir papildomos optimizacijos.

5. DARBO REZULTATAI IR IŠVADOS

1. Šiame darbe buvo analizuojami ir tiriami algoritminiai sprendimai mobiliam robotui, leidžiantys kuo tiksliau aptikti ieškomą objektą bei įvertinti jo poziciją ir orientaciją erdvėje. Atlikus šios srities technologijų analizę surasta įvairių realizacijai tinkamų metodų, tačiau bendro jų efektyvumo palyginimo trūko. Siekiant užpildyti šią spragą realizuota programinė įranga, kuri leido atlikti labiausiai roboto sistemai tinkamų metodų palyginimą.
2. Objektų aptikimo algoritmų efektyvumo tyrimo metu pasiekta maksimali 15 kadru/s veikimo sparta (TLD algoritmas) analizuojant standartinius VGA 640x480 raiškos vaizdus. Eksperimentuose nustatyta objektų aptikimo tikslumo viršutinė riba apie 90% (Haar algoritmas) bei apatinė riba apie 60% (HOG algoritmas), kitų algoritmų – TLD bei LINE-MOD vidutinis tikslumas išsidėstė nurodytame diapazone. Siekiant maksimalaus tikslumo būtina naudoti Haar metodą, o spartos – TLD metodą.
3. Papildomas gylio metodų tikslumo tyrimas padėjo nustatyti, jog efektyviausias sprendimas roboto stereo gylio sistemai spartos ir tikslumo atžvilgiu yra gaunamas naudojant Microsoft Kinect kamerą. Ištirta, kad Kinect tikslumas yra 2-5 kartus geresnis nei įprastinės stereo kamerų sistemos, o gylio paklaida auga lėčiau didėjant atstumui. Gylio skaičiavimų perkėlimas ant specializuoto procesoriaus leidžia pasiekti maksimalią vaizdo apdorojimo sistemos spartą bei gylio duomenis iš kameros gauti 30 kadru/s sparta.
4. Atliktas objektų pozicijos ir orientacijos nustatymo ICP metodo efektyvumo tyrimas parodė, jog vidutinė absoliutinė pozicijos nustatymo paklaida yra apie 3.4cm, o vidutinė absoliutinė kampinė paklaida – apie 30°. Kadangi algoritmas yra intensyvus skaičiavimų prasme ir pasiekia apie 2 kadru/s veikimo spartą, tolesnis optimizavimas arba duomenų kiekio minimizavimas yra būtinas norint pasiekti geresnius veikimo rezultatus.
5. Įvertinus gautus tyrimų rezultatus buvo sėkmingai sukurtas ir išbandytas techninis roboto sprendimas objektų pozicijos ir orientacijos nustatymui. Atliktuose testuose apdorojimo greitis roboto sistemoje buvo apie 2 kartus mažesnis nei nešiojamo kompiuterio. Ne visi robotikos sprendimai reikalauja apdoroti didelius informacijos kiekius realiu laiku. Galimybė sumažinti vaizdo kadro raišką iki 320x240 (analizuojamų duomenų kiekį 4 kartus) visada išlieka. Išlaikyti tokį patį ar pasiekti geresnį algoritmų tikslumą su tokia raiška būtų dar viena užduotis ateičiai.

6. LITERATŪRA

1. G. Narvydas, R. Simutis, *Autonominių Mobilių Robotų Judėjimo Valdymas*, 2006
2. J. Giesbrecht, *Development of a Vision-Based Robotic Follower Vehicle*, Technical report, Canada, 2009
3. V. Lepetit, P. Fua, *Monocular model-based 3D tracking of rigid objects*, 2005
4. G. Xiaogang, W. Changhong, Q. Zhenshen, *Object Tracking for Autonomous Mobile Robot based on Feedback of Monocular-vision*, I21E Conference on Industrial Electronics and Applications, 2007
5. H. Ukida, M. Kawanami, Y. Terama, *3D Object Tracking by Pan-Tilt Moving Cameras and Robot Using Sparse Template Matching and Particle Filter*, SICE Conference, 2011
6. T. Drummond, R. Cipolla, *Real-Time Visual Tracking of Complex Structures*, Transactions On Pattern Analysis And Machine Intelligence, 2002
7. D. G. Lowe, *Fitting Parameterized Three-Dimensional Models to Images*, Transactions on Pattern Analysis and Machine Intelligence, 1991
8. L. Rui, D. Zhijiang, H. Fujun, K. Minxiu, S. Lining, *Tracking a Moving Object with Mobile Robot Based on Vision*, I21E International Joint Conference on Neural Networks, 2008
9. S. Lining, L. Rui, W. Weidong, D. Zhijiang, *Mobile Robot Real-time Path Planning Based on Virtual Targets Method*, Third International Conference on Measuring Technology and Mechatronics Automation, 2011
10. C. Gönner, M. Rous, K. F. Kraiss, *Real-Time Adaptive Colour Segmentation for the RoboCup Middle Size League*, 2005
11. B. Browning, M. Veloso, *Real-Time, Adaptive Color-based Robot Vision*, 2005
12. L. Iocchi, *Robust Color Segmentation through Adaptive Color Distribution Transformation*, 2006
13. C. Stanton, M. A. Williams, *A Novel and Practical Approach towards Color Constancy for Mobile Robots using Overlapping Color Space Signatures*, 2006
14. C. Tomasi, T. Kanade, *Detection and Tracking of Point Features*, Carnegie Mellon University Technical Report CMU-CS-91-132, 1991
15. D. Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*, International Journal of Computer Vision, 2004
16. C. Lin, E. Setiawan, *Object Orientation Recognition Based on SI6 and SVM by Using Stereo Camera*, International Conference on Robotics and Biomimetics, 2009
17. M. Wu, J. Y. Sun, *Moving Object Detecting and Tracking with Mobile Robot Based on Extended Kalman Filter in Unknown Environment*, International Conference on Machine Vision and Human-machine Interface, 2010

18. D. Gohring and H. D. Burkhard, Multi Robot Object Tracking and Self Localization Using Visual Percept Relations, International Conference on Intelligent Robots and Systems, 2006
19. B. Guo, W. Sun, Multiple object detection and tracking with pseudo-particle filter, Frontiers of Electrical and Electronic Engineering, 2009
20. P. Viola, M. Jones, Rapid Object Detection Using a Boosted Cascade of Simple Features, Computer Vision and Pattern Recognition, vol.1, pp. I-511,I-518, 2001
21. N. Dalal, B. Triggs, Histograms of Oriented Gradients for Human Detection, Computer Vision and Pattern Recognition, vol. 1, pp. 886,893, 2005
22. A. Treptow, A. Masselli, A. Zell, Real-Time Object Tracking for Soccer-Robots without Color Information, Robotics and Autonomous Systems, 2004
23. K. Cheol-taek, L. Ju-Jang, An active contour model for object tracking using the previous contour, Artificial Life and Robotics, 2002
24. G. Nebehay, Robust Object Tracking Based on Tracking-Learning-Detection, Faculty of Informatics, TU Vienna, 2012
25. S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, V. Lepetit, Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes, IEEE International Conference on Computer Vision, pp. 858,865, 2011
26. M. S. Guzel, Mobile Robot Navigation using a Vision Based Approach, 2009
27. C. G. Rusu, I. T. Birou, E. Szöke, Fuzzy Based Obstacle Avoidance System for Autonomous Mobile Robot, 2010
28. S. Thrun et al., Autonomous Exploration and Mapping of Abandoned Mines, I21E Robotics & Automation Magazine, 2004
29. R.G. Mihalyi, K. Pathak, N. Vaškevičius, A. Birk, Uncertainty estimation of AR-marker poses for graph-SLAM optimization in 3D object model generation with RGBD data, IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1807,1813, 3-7, 2013
30. S. Winkler, Model-based pose estimation of 3-D objects from camera images using neural networks. Technical Report IB 515-96-12, Diplomarbeit, Institut für Nachrichten-und Hochfrequenztechnik, Technische Universität Wien, pp. 28-31, 1996
31. D. Q. Huynh, Metrics for 3D Rotations: Comparison and Analysis, J. Math. Imaging, vol. 35, no. 2, pp. 155- 164, 2009
32. M. Sridharan, P. Stone, Color Learning and Illumination Invariance on Mobile Robots: A Survey, Robotics and Autonomous Systems, 2009