

KLAIPĖDOS UNIVERSITETAS

Jūrų technikos ir gamtos mokslo fakultetas

Informatikos ir statistikos katedra

SKRAIDANČIO ROBOTO VAIZDO ATPAŽINIMU GRINDŽIAMOS OBJEKTO SEKIMO SISTEMOS TOBULINIMAS

*IMAGE RECOGNITION BASED OBJECT TRACKING
SYSTEM IMPROVEMENT FOR UAV*

Techninių informacinių sistemų inžinerijos studijų programos 621E15004
magistro baigiamasis darbas

Autorius TNMTS-14 gr. Vitalij Glazko

Vadovas Lekt. dr. Darius Drungilas

Klaipėda, 2016

BAKALAURO IR MAGISTRO BAIGIAMŲJŲ DARBŲ LYDRAŠČIO FORMA

Pildo bakalauro/magistro baigiamojo darbo autorius

..... **Vitalij Glazko**.....
(bakalauro/magistro baigiamojo darbo autoriaus vardas, pavardė)

Skraidančio roboto vaizdo atpažinimu grindžiamos objekto sekimo sistemos tobulinimas
(bakalauro/magistro baigiamojo darbo pavadinimas lietuvių kalba)

Patvirtinu, kad bakalauro/magistro baigiamasis darbas parašytas savarankiškai, nepažeidžiant kitiems asmenims priklausančių autorių teisių, visas baigiamasis bakalauro/magistro darbas ar jo dalis nebuvo panaudotas Klaipėdos universitete ir kitose aukštosiose mokyklose.

.....
(bakalauro/ magistro baigiamojo darbo autoriaus ir parašas)

Sutinku, kad bakalauro/magistro baigiamasis darbas būtų naudojamas neatlygintinai 5 m. Klaipėdos universiteto studijų procese.

.....
(bakalauro/ magistro baigiamojo darbo autoriaus ir parašas)

Pildo bakalauro/magistro baigiamojo darbo vadovas

Bakalauro/magistro baigiamąjį darbą ginti.....
.....(įrašyti - **leidžiu** arba **neleidžiu**)

...2016 Lekt. dr. Darius Drungilas
(data) (bakalauro/magistro baigiamojo darbo vadovo vardas, pavardė ir parašas)

Pildo katedros, kuriojančios studijų programa, administratorius (sekretorius)

Baigiamasis darbas įregistruotas katedroje

2016.....
(data)

Laima Brazdeikienė
(katedros sekretorės vardas, pavardė ir parašas)

Pildo katedros, kuriojančios studijų programą, vedėjas

Bakalauro/magistro baigiamąjį darbą ginti.....
.....(įrašyti - **leidžiu** arba **neleidžiu**)

2016..... prof. dr. Arūnas Andziulis.....
(data) (studijų programos vadovo vardas, pavardė ir parašas)

Recenzentu(-ais) skiriu

Prof. Olegas Ramašauskas.....

..... Lekt. dr. Gediminas Gricius.....
(įrašyti recenzento(ų) vardą, pavardę)

.....2016.....
(data)

prof. dr. Arūnas Andziulis.....
(studijų programos vadovo vardas, pavardė ir parašas)

ANOTACIJA

Darbe nagrinėjamas skraidančio roboto vaizdo atpažinimu grindžiamos objekto sekimo sistemos tobulinimas. Sukurtoji įterptinė sistema pagal vaizdus atpažįsta ir seka objekto žymeklį ir valdo skraidantį robotą. Literatūroje apžvelgiamos panašios sistemos, naudojami algoritmai ir priemonės, skirtos objekto atpažinimui, sekimui. Apžvelgiami esami žymekliai ir jų atsparumas sudėtingoms aplinkos sąlygoms. Darbe tiriami algoritmų ir priemonių pritaikomumas prie skraidančio roboto įterptinės sistemos. Darbe atsižvelgiamos vaizdo atpažinimu grindžiamos objekto sekimo kompiuterinės regos sistemos, nagrinėjama objekto atpažinimo ir sekimo algoritmų įvairovė, įterptinės sistemos komponentų pasirinkimas, įterptinės kompiuterinės regėjimo sistemos techninės ir programinės dalies projektavimas, bei vaizdo atpažinimu grindžiamos objekto sekimo kompiuterinės regos sistemos naudojamų metodų rezultatų ir žymeklio atpažinimo ir sekimo rezultatų įvertinimas. Eksperimentų metu nustatyta, kad patobulintas binarizacijos adaptyvaus slenksčio nustatymo metodas, labiausiai pagerina žymeklio atpažinimo, identifikavimo ir sekimo procesą. Naudojant patobulintą binarizacijos adaptyvaus slenksčio nustatymo metodą skraidančio roboto vaizdo atpažinimu grindžiamos objekto sekimo įterptinės sistemos žymeklio atpažinimo tikslumas padidėjo nuo 66,0% iki 87,3%, o žymeklio identifikavimo tikslumas padidėjo nuo 76,1% iki 88,3%.

PAGRINDINIAI ŽODŽIAI: objekto sekimas, vaizdo atpažinimas, OpenCV, skraidantis robotas, įterptinė sistema.

ABSTRACT

This essay covers image recognition based object tracking system improvement for UAV. Established system, using computer vision, detects and tracks the object and controls the flying robot. Similar systems were analyzed in the scientific articles, which are using the methods and algorithms for recognition and object tracking. Considered already used markers and their resistance to difficult environmental conditions. In this paper we verified the applicability of methods and algorithms to the flying robot system. In this paper we verified the applicability of methods and algorithms to the flying robot system. Also in this paper we consider image recognition based object tracking system, were studied a variety of methods for the detection and tracking, components necessary to build a system were selected, for designing engineering and software parts of the system, the results of the system and the algorithms used were estimated. The experiment found that the improved adaptive threshold is best to improve the process of

recognizing, identifying and tracking an object. Using an improved method of adaptive threshold, object recognition marker accuracy of flying robot system increased from 66.0% to 87.3%. The accuracy of identification of the object marker increased from 76.1% to 88.3%.

KEYWORDS: object tracking, computer vision, OpenCV, flying robot, embedded system.

TERMINŲ ŽODYNĖLIS

Terminų žodynėlio tikslas – pateikti šiame platų temų spektrą apimančiame darbe kai kurių dažniausiai vartojamų terminų apibrėžimus. Vartojant abipusiai suprantamus žodžius sukuriamas pagrindas keistis faktais ir idėjomis.

ABS (angl. Acrylonitrile Butadiene Styrene) – polimeras, kopolimero rūšis, sudaryta iš akrilnitrilo, 1,3-butadieno, ir stireno monomerų. ABS yra gana stiprus ir tvirtas, atsparus termoplastikas naudojamas gaminti įvairias plastiko dalis, kur reikia tokios savybės.

ACC (angl. ACCuracy) – tikslumas, dydis nusakantis kaip tiksliai sistema atpažįsta žymeklį, kai žymeklis yra scenos kadre ir kaip tiksliai sistema neatpažįsta žymeklio, kai žymeklio nėra scenos kadre.

APM (angl. ArduPilotMega) – skraidančio roboto valdiklio pavadinimas.

C++ – bendros paskirties programavimo kalba.

FN (angl. False Negative) – žymeklio neatpažinimų skaičius.

FP (angl. False Positive) – klaidingų žymeklio atpažinimų skaičius.

GPS (angl. Global Positioning System) – globalinė padėties nustatymo sistema. Tai globalioji sistema, iš Žemės palydovų siunčianti radijo signalus, pagal kuriuos padėties nustatymo imtuvai gali nustatyti savo buvimo vietas.

HSV (angl. Hue, Saturation, Value) – AGS spalvų modelis – kuriame spalva apibūdinama trimis parametrais: Atspalviu, Gryniu ir Skaisčiu.

NPV (angl. Negative Predictive Value) – neigiama žymeklio aptikimo prognozė vertė parodo, kokia dalis tyrimo metu teigiamų žymeklio neatpažinimų sudaro nuo visų žymeklio neatpažinimų.

PID (angl. Proportional–Integral–Derivative) – reguliatorius, turintis grįžtamąjį ryšį ir naudojamas pramoninėse kontrolės sistemose. Reguliatorius yra parametriškai optimizuojamas, kurį sudaro 3 grandys: proporcinė, integruojanti ir diferencijuojanti.

PPM (angl. Pulse Position Modulation) – impulso pareigos moduliacija.

PPV (angl. Positive Predictive Value) – teigiama žymeklio aptikimo prognozė vertė parodo, kokia dalis tyrimo metu teigiamų žymeklio atpažinimų sudaro nuo visų žymeklio atpažinimų.

PWM (angl. Pulse Width Modulation) – impulso pločio moduliacija.

RGB (angl. Red, Green, Blue) – RŽM spalvų maišymo modelis, kurioje naudojamos trys, žmogaus akių receptorių atitinkančios spalvos: Raudona, Žalia ir Mėlyna.

SPI (angl. Serial Peripheral Interface) – paprasčiausia sinchroninė nuoseklaus duomenų perdavimo sąsaja, skirta keistis duomenimis tarp dviejų įrenginių, kurių vienas yra pagrindinis (angl. master), kitas – valdomasis (angl. slave).

TN (angl. True Negative) – teisingų žymeklio neatpažinimų skaičius.

TNR (angl. True Negative Rate) – specifiškumas, rodiklis, apibūdinantis metodo tikslumą ir rodantis tikimybę, kad žymeklio atpažinimas bus neigiamas, kai žymeklio nėra scenos kadre .

TP (angl. True Positive) – teisingų žymeklio atpažinimų skaičius.

TPR (angl. True Positive Rate) – jautrumas, rodiklis, apibūdinantis metodo tikslumą ir rodantis tikimybę, kad žymeklio atpažinimas bus teigiamas, kai žymeklis yra scenos kadre.

UAV (angl. Unmanned Aerial Vehicle) – nepilotuojamas orlaivis.

TURINYS

ĮVADAS.....	11
1 LITERATŪROS APŽVALGA IR SUSIJĘ DARBAI	13
1.1 Kompiuterinis regėjimas ir vaizdo analizės metodai	13
1.2 Objekto atpažinimo ir sekimo principai	14
1.3 Išorinių faktorių įtaka vaizdo analizei.....	15
1.4 Žymekliai ir jų įvairovė	16
1.5 Vaizdo atpažinimu grindžiamos objekto sekimo įterptinės sistemos	17
1.6 Skyriaus apibendrinimas	21
2 OBJEKTO ATPAŽINIMO IR SEKIMO METODŲ IR PRIEMONIŲ PRITAIKOMUMAS SKRAIDANČIŲ ROBOTŲ SISTEMOSE.....	22
2.1 Skraidančio roboto vaizdo atpažinimu grindžiama objekto sekimo įterptinė sistema	22
2.2 Kompiuterinė rega skraidančiuose robotuose	23
2.3 Sekamo objekto žymeklio sudarymas.....	25
2.4 Vaizdo apdorojimo metodai, sekamo objekto atpažinimas	26
2.5 Objekto identifikavimas ir sekimas	29
2.6 Skyriaus apibendrinimas	30
3 SKRAIDANČIO ROBOTO OBJEKTO ATPAŽINIMO IR SEKIMO ĮTERPTINĖS SISTEMOS PROJEKTAVIMAS IR PROTOTIPO KŪRIMAS	31
3.1 Skraidančio roboto vaizdo atpažinimu grindžiamos objekto sekimo įterptinės sistemos reikalavimai	31
3.2 Valdiklio parinkimas vaizdo atpažinimu grindžiamai skraidančio roboto objekto sekimo įterptinei sistemai	32
3.3 Vaizdo kameros parinkimas skraidančio roboto objekto sekimo įterptinei sistemai	34
3.4 Papildomas apšvietimas skraidančio roboto objekto sekimo įterptinei sistemai	36
3.5 Techninės skraidančio roboto objekto sekimo įterptinės sistemos projektavimas	37

3.6	Vaizdo kameros stabilizacijos sistemos parinkimas	42
3.7	Konstrukcinių dalių projektavimas su SolidWorks	42
3.8	Skraidančio roboto įterptinės objekto sekimo sistemos programinio prototipo kūrimas	43
3.9	Skraidančio roboto objekto sekimo įterptinės sistemos prototipo kūrimas ir jos integravimas į skraidantį robotą	47
3.10	Skyriaus apibendrinimas	49
4	SKRAIDANČIO ROBOTO VAIZDO ATPAŽINIMU GRINDŽIAMOS OBJEKTO SEKIMO ĮTERPTINĖS SISTEMOS EKSPERIMENTINIAI TYRIMAI.....	50
4.1	Eksperimento tikslai	50
4.2	Eksperimento sąlygos	50
4.3	Vaizdo binarizacijos slenksčio nustatymo metodo eksperimentinis įvertinimas.	53
4.4	Skraidančio roboto įterptinės objekto sekimo sistemos atsparumo vibracijai eksperimentinis tyrimas.....	60
4.5	Skyriaus apibendrinimas	61
	IŠVADOS	63
	LITERATŪRA.....	64
	SANTRAUKA	68
	SUMMARY	68
	КРАТКОЕ ИЗЛОЖЕНИЕ.....	68
	PRIEDAI	1

PAVEIKSLĖLIŲ SĄRAŠAS

1 pav. Objektų atpažinimas: a) kai rutulio paviršius yra blizgantis; b) atpažinti cilindriniai objektai pažymėti mėlyna linija [23]	15
2 pav. Žymekliai: a) Barcode [18]; b) Cybercode [40]; c) VisualCode [41]; d) MaxiCode [41]; e) QR [18], [42], f) ReacTIVision [43].....	16
3 pav. Skraidančio roboto saugios nusileidimo vietos aptikimas : a) originalus vaizdas; b) kraštų aptikimas; c) potencialios nusileidimo vietos, pažymėtos balta spalva [47]	18

4 pav. Skraidančio roboto žymeklis, naudojamas kaip nusileidimo platforma [1].....	18
5 pav. Žymeklio atpažinimas: a) originalus vaizdas; b) aptiktas žymeklio kontūras [1].....	19
6 pav. Skraidantis robotas su akumulatoriaus bevielio parkavimo sistema [2].....	19
7 pav. Katamaranas su integruota vaizdo apdorojimo sistema „Raspberry Pi“ mikrovaldiklyje [48].....	20
8 pav. Sekamo objekto tekstūra [48].....	20
9 pav. Skraidančio roboto vaizdo atpažinimu grindžiamą objekto sekimo įterptinės sistemos modelis ir jo komunikacija su skaidančio roboto valdikliu.....	22
10 pav. Skraidantis robotas: a) <i>Hexacopter</i> ; b) <i>Quietcopter RC Multi</i> ; c) <i>Parallax ELEV-8</i>	23
11 pav. Nuotolinio vaizdo perdavimo palyginimas: a) skaitmeninis signalas; b) analoginis signalas.....	24
12 pav. Suprojektuotas žymeklis.....	26
13 pav. Paprastas ir adaptyvus binarizacijos slenksčio nustatymo metodai: a) Originalus vaizdas; b) Paprastas binarizacijos slenksčio nustatymas; c) Adaptyvus binarizacijos slenksčio nustatymas.....	27
14 pav. Rezultatai panaudojant skirtingus binarizacijos slenksčio nustatymo metodus: a). originalus vaizdas, b). paprastas binarizacijos slenksčio nustatymas, c) adaptyvus binarizacijos slenksčio nustatymas, d) binarizacijos slenksčio nustatymas naudojant vietinius blokus, e) pagerintas metodas naudojantis vietinių blokų binarizacijos slenksčio nustatymo rezultatus, d) patobulintas adaptyvus binarizacijos slenksčio nustatymas.....	27
15 pav. Valdikliai: a) <i>Raspberry Pi Model B</i> ; b) <i>Raspberry Pi 2 Model B</i> ; c) <i>ODROID-C1+</i> ; d) <i>ODROID-XU4</i>	33
16 pav. Kameros „Canyon Webcams“ vaizdo epipoliarinis iškraipymas prieš korekciją....	36
17 pav. Šviesos diodų juostos impulsinio maitinimo šaltinio principinė schema.....	37
18 pav. Skraidančio roboto valdymo pulto imtuvo siunčiamų signalų perdavimo principas.....	38
19 pav. Skraidančio roboto imtuvo modifikavimo schema: nuoseklus nelyginių kanalų sujungimas.....	39
20 pav. Įtampos lygio konverteris modulis su „TXS0104ED“ tarp „Raspberry Pi 2 Model B“ ir „Arduino Micro“ bedravimui per „SPI“ protokolą.....	40
21 pav. „MOP“ tranzistoriaus prijungimo prie vaizdo kameros modulio schema.....	41
22 pav. Įterptinės sistemos ir naudojamų komponentų prisijungimo schema.....	41
23 pav. Suprojektuotas laikiklis su „SolidWorks 2015“ programiniu paketu.....	43

24 pav. Patobulintas įterptinės sistemos modelis, rodantis ryšius tarp sistemos komponentų ir duomenų perdavimo trukmes: (°) be sinchronizavimo; (*) su sinchronizavimu	44
25 pav. Skraidančio roboto įterptinės sistemos įjungimo algoritmas	44
26 pav. Skraidančio roboto įterptinės sistemos branduolio algoritmas.....	45
27 pav. Skraidančio roboto įterptinės sistemos kompiuterinio regėjimo žymeklio atpažinimo ir sekimo algoritmas	46
28 pav. Originalus skraidantis robotas <i>Parallax ELEV-8</i>	47
29 pav. Modifikuotas skraidantis robotas su integruota įterptine sistema	48
30 pav. Modifikuota vaizdo kameros stabilizacijos sistema su atspausdintu laikikliu ir pritvirtintais komponentais	48
31 pav. Vaizdai gauti eksperimentu metu demonstruoja eksperimentu sąlygas.....	51
32 pav. Žymeklio filmavimo kadrai demonstruoja eksperimento sąlygas.....	53
33 pav. Skraidančio roboto objekto sekimo įterptinės sistemos telemetrija: vaizdo binarizacijos algoritmo apdorojimo trukmė naudojant skirtingus slenksčio nustatymo metodus.	54
34 pav. Skraidančio roboto objekto sekimo įterptinės sistemos telemetrija: žymeklio atpažinimo ir sekimo algoritmų apdorojimo trukmė naudojant skirtingus binarizacijos slenksčio nustatymo metodus	55
35 pav. Skraidančio roboto objekto sekimo įterptinės sistemos telemetrija: kontūrų aptikimo algoritmo apdorojimo trukmė, naudojant skirtingus binarizacijos slenksčio nustatymo metodus	55
36 pav. Skraidančio roboto objekto sekimo įterptinės sistemos telemetrija: aptiktų kontūrų kiekis naudojant skirtingus binarizacijos slenksčio nustatymo metodus	56
37 pav. Skraidančio roboto objekto sekimo įterptinės sistemos algoritmų apdorojimo vidutinės trukmės suma atskiruose etapuose: 1) binarizacijos slenksčio nustatymo algoritmas; 2) kontūrų aptikimo algoritmas; 3) žymeklio atpažinimo algoritmas; 4) žymeklio identifikavimo algoritmas; 5) žymeklio sekimo algoritmas.....	57
38 pav. Skraidančio roboto objekto sekimo įterptinės sistemos telemetrija: žymeklio atpažinimo rezultatas	58
39 pav. Skraidančio roboto objekto sekimo įterptinės sistemos telemetrija: žymeklio identifikavimo rezultatai.....	59
40 pav. Mokslinio darbo [17] eksperimento pakartojimo rezultatas naudojant patobulintą adaptyvų slenksčio nustatymo metodą	61

LENTELIŲ SĄRAŠAS

1 lentelė. Valdiklių parametrų palyginimas [53 – 56].....	33
---	----

2 lentelė. Kamerų techninių parametrų palyginimas [57 – 62].....	35
3 lentelė. OSM5D3Z2C1P diodo techniniai parametrai	37
4 lentelė. Įterptinei sistemai apnaudotų komponentų sąrašas	49
5 lentelė. Algoritimų apdorojimo trukmės tyrimo rezultatai naudojant skirtingus binarizacijos slenksčio nustatymo metodus	57
6 lentelė. Žymeklio atpažinimo rodikliai naudojant skirtingus binarizacijos slenksčio nustatymo metodus	58
7 lentelė. Žymeklio identifikavimo rodikliai naudojant skirtingus binarizacijos slenksčio nustatymo metodus	60

IVADAS

Skraidantys robotai – perspektyvi technologija su įvairiomis pritaikymo galimybėmis. daugelyje sričių: nuo priežiūros ir žvalgybos iki smulkių krovinių gabenimo. Šiuo metu svarbus uždavinys – skraidančių robotų automatizavimas [1]. Kuriamos sistemos, kurios jutiklių pagalba leidžia robotui orientuotis aplinkoje, atlikti žvalgybos funkcijas. Ši technologija vis dar yra aktyvioje vystymo stadijoje. Tačiau visų skraidančio roboto trūkumas – tai maža skraidymo trukmė [2], o akumulatoriaus talpos padidinimas neišsprendžia šio trūkumo. Siekiant užtikrinti skraidančio roboto saugų grįžimą į pradinę poziciją siūlomas automatinio nusileidimo ant akumulatoriaus įkrovimo platformos režimas, kuris automatiškai įsijungia, kai energijos kiekis akumuliatoriuje bus mažesnis kaip $\frac{1}{2}$ akumulatoriaus talpos. Skraidančio roboto automatinis nusileidimas ant akumulatoriaus įkrovimo platformos reikalauja tikslaus skraidančio roboto pozicionavimo nusileidimo metu. Vienas iš variantų – naudoti GPS modulį, tačiau naudojant tik GPS pozicionavimo duomenis, skraidančio roboto nusileidimo pozicijos neapibrėžtis seka iki 3 metrų [3], [4]. Naudojant antžeminių stočių GPS signalo patikslinimus, neapibrėžtį galima sumažinti iki 1 cm, tačiau priklauso nuo įvairių faktorių:

- ar vienodų palydovų signalus gauna GPS modulis ir antžeminė stotis;
- koks skraidančio roboto su antžemine stotimi mobiliojo interneto atsako greitis.

Naudojant tik GPS signalą neužtenka užtikrinti skraidančio roboto pozicionavimo tikslumą nusileidimo metu. Pozicionavimo tikslumas nusileidimo metu gali būti padidintas naudojant vaizdo atpažinimo grindžiamą objekto sekimo sistemą. Roboto kompiuterinės regos sistema aptinka specialius objektus – žymeklius, kurie naudojami kaip skraidančio roboto nusileidimo platformos. Pagerintas esamų sistemų nusileidimo tikslumas leistų automatizuoti skraidančio roboto akumulatoriaus įkrovimo procesą. Šiam sprendimui naudojamos naujos kartos kameros suteikia galimybę skraidančiam robotui suvokti aplinką, analizuoti ir rasti reikiamą sprendimą pagal nustatytą uždavinį. Tačiau tokiose sistemose taikomų algoritmų pagrindinis trūkumas – vaizdo apdorojimo greitis. Nuo kadrų apdorojimo greičio priklauso skraidančio roboto reakcijos trukmė, veiksmų tikslumas ir judėjimo greitis. Kitas aspektas – skraidančio roboto keliamoji galia yra apribota dėl techninių parametrų, todėl integruota įterptinė vaizdo apdorojimo sistema turi būti kompaktiška, pakankamai sparti ir vartoti mažai elektros energijos.

Darbo tikslas – patobulinti skraidančio roboto vaizdo atpažinimo grindžiamą objekto sekimo sistemą, siekiant padidinti atpažinimo tikslumą esant sudėtingomis aplinkos sąlygomis.

Darbo uždaviniai:

1. apžvelgti mokslinėje literatūroje pateikiamus populiariausius robotizuotų sistemų objekto atpažinimo ir sekimo algoritmus ir įvertinti jų privalumus ir trūkumus;

2. išanalizuoti objekto atpažinimo ir sekimo algoritmų pritaikomumą skraidančių robotų sistemose;
3. suprojektuoti įterptinę, vaizdo atpažinimu grindžiamą, objekto sekimo sistemą ir integruoti ją į skraidantį robotą;
4. eksperimentiškai įvertinti sukurtos įterptinės sistemos parametrus.

Šiame darbe atsižvelgiamos vaizdo atpažinimu grindžiamos objekto sekimo kompiuterinės regos sistemos, nagrinėjama objekto atpažinimo ir sekimo algoritmų įvairovė, įterptinės sistemos komponentų pasirinkimas, įterptinės kompiuterinės regėjimo sistemos techninės ir programinės dalies projektavimas, bei vaizdo atpažinimu grindžiamos objekto sekimo kompiuterinės regos sistemos naudojamų metodų rezultatų ir žymeklio atpažinimo ir sekimo rezultatų įvertinimas.

Dalis šio darbo rezultatų buvo pristatyti tarptautinėje mokslinėje konferencijoje „Knowledge in Telecommunication Technologies and Optics 2015 (KTTO-2015), Čekija, 2015 rugsėjo 9-11“ ir publikuota moksliniame darbe: „A. Andziulis, D. Drungilas, V. Glazko, E. Kiseliovas, „Resource Saving Approach of Visual Tracking Fiducial Marker Recognition“, Advances in Electrical and Electronic Engineering, 2015, pp. 359-366“.

1 LITERATŪROS APŽVALGA IR SUSIJĘ DARBAI

Kompiuterinis regėjimas (angl. *computer vision*) – mokslas apie vaizdų apdorojimą ir galiausiai erdvės suvokimą naudojant kompiuterius [5]. Kompiuterinis regėjimas vis labiau įsitraukia į naujas technologijas, inžinerinius sprendimus, suteikia galimybių spręsti įvairias užduotis, automatizuoti sistemas. Šiame skyriuje apžvelgiamos sistemos, kurios naudoja kompiuterinį regėjimą, su objekto atpažinimu ir sekimu susiję metodai bei komponentai.

1.1 Kompiuterinis regėjimas ir vaizdo analizės metodai

Realaus laiko kompiuterinis regėjimas vis plačiau taikomas robotikoje, automobilių sistemose ir kitur [3], [6]. Sistemos, naudojančios kompiuterinį regėjimą, gali būti realizuojamos taikant įvairius architektūrinius sprendimus [4], [6]–[9]. Priklausomai nuo probleminės srities tokiose sistemose gali būti naudojamos viena ar daugiau kamerų. Realaus laiko sistemos gali suvokti aplinką, atpažinti objektus, nustatyti jų poziciją.

Specialus kamerų išdėstymas suteikia galimybę gauti stereoskopinį vaizdą [10], [11]. Tokiu būdu, sistema naudojanti tik kompiuterinį regėjimą, gali nustatyti scenos gylį. Atstumo iki objektų tikslumas priklausys nuo vaizdų rezoliucijos ir pradinės kamerų kalibracijos [12], [13].

Šuo metu sukurta daug įvairių bibliotekų, kurios paspartina kompiuteriniu regėjimu grindžiamų programų ir aplikacijų kūrimą. Viena iš šių bibliotekų – *OpenCV*. Tai stabili biblioteka, turinti C++ sąsaja, ir padalinta į keletą atskirų modulių. Kiekvienas modulis yra vienakryptis ir turi savo sprendimų kompleksą. Šiandien biblioteka turi daugiau kaip 2500 optimizuotų vaizdų apdorojimo algoritmų. Biblioteka naudojama akademiniams ir komerciniams tikslais pagal *BSD* licenciją [5]. *OpenCV* biblioteka suteikia galimybę panaudojant sukurtus metodus ir algoritmus atpažinti, pavyzdžiui:

- tiesias linijas;
- rutulio formos objektus;
- žmogaus veidą;
- objektus pagal kontūrą;
- ir kitus.

Autonominių robotų valdymo sistemose taikomas vaizdo atpažinimas vystosi ypač greitai [3], [14], [15]. Jau sukurtas automobilio autopilotas, kuris naudodamas kompiuterinį regėjimą nustato eismo juostas, atstumą iki priešais važiojančios transporto priemonės ir pan. Tačiau šiose sistemose vis dar egzistuoja trūkumų, kuriuos reikia spręsti, pavyzdžiui, kaip sistema elgsis, kai

aplinkos sąlygos nebuvo numatytos programoje [3], kaip autonominis robotas elgsis, kai vaizdo scenos analizavimas užtrunka ilgiau nei numatytą dėl specifinės aplinkos [4].

1.2 Objekto atpažinimo ir siekimo principai

Vaizdų analize grindžiami objektų atpažinimo metodai parenkami priklausomai nuo objektų savybių: formos, spalvos, tekstūros, dydžio [1], [3]. Idealiausias atvejis, kai atpažįstamas ir sekamas objektas yra specifinis ir išsiskiriantis iš aplinkos [1], [16]. Dauguma objektų atpažinimo ir sekimo metodų yra pagrįsti pradinio vaizdo supaprastinimu – segmentavimu – iki vieno kanalo dvejetainio (binarinio) vaizdo – balto ir juodo pavidalo [1], [3], [14], [16]. Toliau objektai klasifikuojami pagal vaizdo fragmentų dydžius ir formas, jeigu tokios yra apibrėžtos [1], [14]. Jeigu objektas atpažįstamas naudojant žymeklį, tai pradinis vaizdas segmentuojamas be išankstinio apdorojimo ir perdangos (angl. *overlay*) filtrų [1], [3], [14].

Metodai kurie išskiria konkrečią specifinę objekto spalvą, reikalauja išfiltruoti visas kitas spalvas iš pradinio vaizdo. Pagal spalvą objektas gali būti atpažįstamas ir klasifikuojamas tik esant specialiom sąlygom, objektas turi būti pasuktas reikiamų kampų kameros atžvilgiu [1]. Kitu atveju objektas gali atspindėti aplinkos spalvų įvairovę, ir taip pakeisti originalaus objekto spalvą [1], [16].

Metodai, naudojantys binarinį vaizdą, ieško uždarų kontūrų – uždarų „dėmių“ [14], pagal kurias objektai yra suklasifikuojami [3], [14], [16]. Klasifikavimas supaprastinamas panaudojant šablonus – žymeklius, kurie gali būti kelių rūšių [17], [18]:

- matematiškai (programiškai) pagrįsti,
- „chaotiški“ paveikslai.

Antros rūšies žymekliai pagrįsti paprastais vaizdais [17], [19], [20], pagal kuriuos kadre vyksta atitikmens paieška [21]. Pirmos rūšies žymekliai yra specialiai suprojektuoti ir yra griežtai nustatytos struktūros, be to, juose gali būti užkoduota informacija [18].

Objektu atpažinimas ir klasifikacija vyksta pritaikant vieną ar keletą algoritmų [22] – [24]. Jei objektas atpažįstamas pagal atspalvį, tai blizgantis paviršius sudaro atspindžius su įvairiomis spalvomis, todėl gali atsirasti spalvų mišinys. Pavyzdžiui, 1 pav. a dalyje pavaizduotas rutulys su blizgančiu paviršiumi, kurį sistema turi atpažinti ir klasifikuoti.



1 pav. Objektų atpažinimas: a) kai rutulio paviršius yra blizgantis; b) atpažinti cilindriniai objektai pažymėti mėlyna linija [23]

Vienas iš galimų sprendimų – tai rasti visus baltus regionus ir patikrinti: jei kontūras apsuptas raudonu rėmeliu – tai rutulys raudonas. Kitas metodas paima išorinį objekto kontūrą ir stebi visus vidinius regionus ignoruojant blizgius. Naudojant antrąjį metodą problema kyla su objektu paieška, nes blizgesys praeina per visą objektą ir algoritmas atpažįsta kaip du skirtingus regionus. Tokia pat problema gali atsirasti, kai rutuliai yra per arti sienos, arba šalia kito rutulio, tada algoritmas nebegali tiksliai nustatyti regionų skaičiaus. Šešėlius iš vaizdo gana lengvai galima panaikinti, vaizdą reikia konvertuoti į dvejetainį naudojant binarizacijos slenksčio nustatymą. Tačiau po normalizacijos gali atsirasti problemų su juodais objektais [23], [24]. Algoritmo apdorojimo rezultatas pavaizduotas 1 pav. b dalyje. Atpažinti ir sekti objektą galima panaudojant įvairius algoritmus:

- šablono atitikimas (angl. *template matching*) [25];
- žymeklio aptikimas (angl. *marker detection*) [26];
- krašto aptikimas (angl. *Canny edge detector*) [27];
- kampo aptikimas (angl. *corner detection*) [28].

Algoritmai skirti objekto sekimui:

- Lucas-Kanade metodas [29];
- judėjimo gradientas (angl. *motion gradient*) [30];
- Kalman filtras [31].

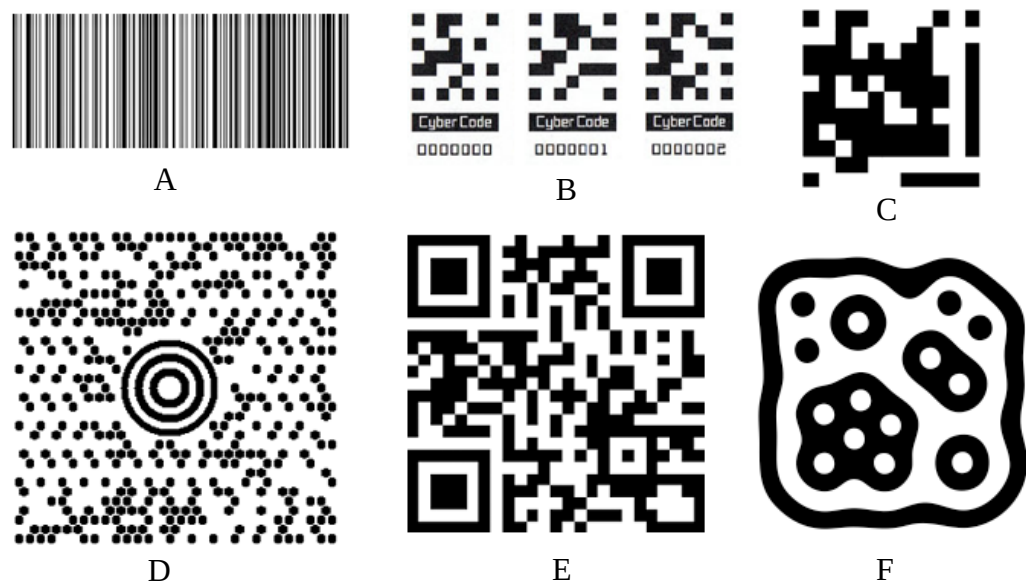
1.3 Išorinių faktorių įtaka vaizdo analizei

Objekto atpažinimo ir sekimo tikslumas priklauso nuo aplinkos sąlygų sudėtingumo [32], [33]. Sekamo objekto žymeklio pozicijos nustatymo metodai [34], [35] plačiai naudojami stacionariose stebėjimo sistemose. Šie metodai gali būti naudojami ir robotikoje – robotas gali sekti sekamo objekto žymeklį. Atpažinimo ir sekimo tikslumas sumažėja, kai sistema nėra

stacionari. Žymeklio sekimas reikalauja panašumu tarp esamo ir buvusio kadro vaizdų. Vaizdo kokybė roboto judėjimo metu sumažėja, vaizdas gali būti iškreiptas, triukšmingas, neryškus [33]. Vaizdo koregavimas reikalauja papildomų procesorinių ir laiko išteklių [36]. Aplinkos apšviestumo sąlygos taip pat turi didelę įtaką žymeklio atpažinimui [12], [37]. Žymeklių, kuriuose naudojamos spalvos, trūkumai pasireiškia esant skirtingoms apšvietimo sąlygoms. Atpažįstant šiuos žymeklius vaizdo spalvos formatai konvertuojami iš *RGB* į *HSV* spalvų erdvę [37]. Pagal *HSV* galima nustatyti spalvų intervalus. Juodi-balti, matematiškai pagrįsti, žymekliai mažiau priklauso nuo apšviestumo sąlygų [18]. Pats žymeklis išskiriamas konvertuojant vaizdą į dvejetainį pavidalą naudojant binarizacijos slenksčio nustatymo metodą.

1.4 Žymekliai ir jų įvairovė

Yra daug pavyzdžių, kurie demonstruoja žymeklio atpažinimo ir sekimo panaudojimą įrenginiuose [19], [20], [38], naudojant įvairius vaizdų atpažinimo algoritmus. Šiuo atveju naudojami vienos plokštumos, juodos ir baltos spalvos žymekliai, kuriuose užkoduojamas identifikavimo kodas sektoriuose arba žieduose [39].



2 pav. Žymekliai: a) Barcode [18]; b) Cybercode [40]; c) VisualCode [41]; d) MaxiCode [41]; e) QR [18], [42], f) ReacTIVision [43]

Barcode [18] žymekliai (2 pav.) sudaryti iš 1D-brūkšnių, kiekvieno brūkšninio plotis yra fiksuotas. Kiekvienas brūkšnelis – tai bitas: juodos spalvos – 0, baltos – 1. Šie identifikatoriai turi labai ribotą užkoduotos informacijos kiekį, kitaip jų plotis turi būti didesnis, be to, šis žymeklis turi būti ant lygaus paviršiaus.

Cybercode [40], VisualCode [41] ir MaxiCode žymekliai (2 pav.) sudaryti iš 2D-taškų, kur kiekvieno taško dydis yra fiksuotas. Šiuo atveju, lyginant su Barcode, užkoduotos informacijos

kiekis gali būti didesnis. ir Žymekliuose užkoduotos informacijos kiekis proporcingai priklauso žymeklio dydžiui. Šie žymekliai atpažįstami ir nuskaitomi naudojant atitinkamas kaukes, tačiau nelygus žymeklio paviršius gali sumaišyti taškų pozicijas kameros atžvilgiu ir informacija nebus nuskaitoma.

QR (greito atsako, angl. *Quick Response*) žymekliai [18], [42] (2 pav.) sudaryti iš „3D“ taškus. Kaip ir 2D žymekliai sudaryti iš taškų ar apskritimų, kurių dydis yra fiksuotas ar matematiškai (programiškai) apibrėžtas. QR žymeklio privalumas tai, jog galima nustatyti jo orientaciją, ir naudoti ant nelygių paviršių. QR žymeklis sudarytas iš 3 raktinius taškų – kvadratų, kurių viduje yra mažesni kvadratai. Mažesnio kvadrato dydis yra proporcingas didesniojo kvadrato ir yra pastovus. Papildomos juostos, sudarytos iš pasikartojančių „01“, papildomi maži kvadratėliai kurie yra išdėstyti griežtai nustatytais atstumais nuo raktinių taškų yra naudojami sukurti QR žymeklio nuskaitymo šabloną. „Laisva“ žymeklio zona naudojama informacijai užkoduoti. QR žymeklio atpažinimas plačiai naudojamas išmaniuosiuose telefonuose, nes šio žymeklio atpažinimas ir informacijos nuskaitymas yra greitas [42], [44].

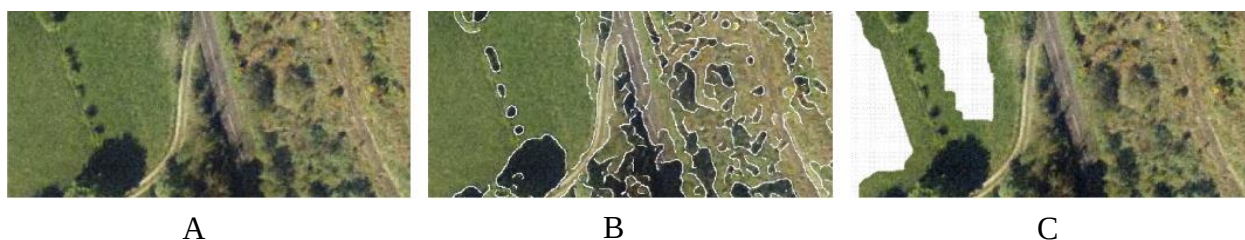
ReacTIVision žymekliuose (2 pav.) naudojami specifinės formos ir elementai, kurių konstrukcija optimizuojama naudojant genetinius algoritmus. ReacTIVision žymekliai buvo specialiai suprojektuoti taip, kad minimizuotų blogo aplinkos apšvietimo poveikį.

Kai žymeklis apšviestas netolygiai, paprasto binarizacijos slenksčio nustatymo metodo pritaikymas neduos gerų rezultatų, ypač tose vietose, kur žymeklio dalis apšviečiama mažiausiai [43]. Keletas autorių pasiūlė būdus, kaip pagerinti žymeklio aptikimą ir klasifikaciją esant blogoms aplinkos apšvietimo sąlygoms bei staigiai judant kamerali, kai kamerą nuskaito neryškų vaizdą [43], arba kai dalis žymeklio yra nematoma [45]. Viola ir Jones pasiūlė paprastą klasifikatorių greito objekto aptikimo vaizduose (FIPS) [21]. Jų pasiūlytas metodas naudoja spalvų filtrą, kaip binarizacijos slenksčio nustatymo parametą. Belussi ir Hirata pasiūlė metodą, kuris padeda nuspręsti, ar FIPS metodas yra tinkamas QR žymekliui [46].

1.5 Vaizdo atpažinimu grindžiamos objekto sekimo įterptinės sistemos

Kaip rašo Timothy Patterson ir kt. publikuotame straipsnyje „*Timely autonomous identification of UAV safe landing zones*“ [47] roboto sistema, naudodama vaizdo atpažinimo algoritmus, ieško tinkamos ir saugios nusileidimo vietos. Skraidantis robotas skraidymo metu gali susidurti su klaidomis, pavyzdžiui, dingsta ryšis su valdymo pultu arba GPS moduliu. Skraidančio roboto sistema laukia ryšio atkūrimo, jei ryšis nebuvo atkurtas per tam tikrą laiką, skraidantis robotas turi kuo greičiau ir saugiai nusileisti į atitinkamą vietą. Šiai problemai spręsti buvo sukurta

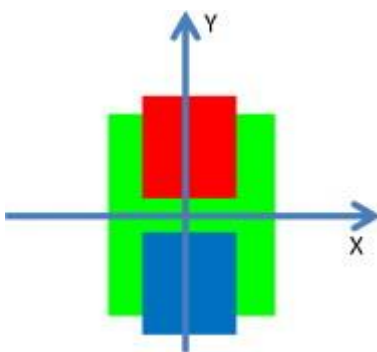
sistema [47], naudojanti kompiuterinį regėjimą. Sistemos nusileidimo vietos paieškos procesas pavaizduotas 3 pav.



3 pav. Skraidančio roboto saugios nusileidimo vietos aptikimas : a) originalus vaizdas; b) kraštų aptikimas; c) potencialios nusileidimo vietos, pažymėtos balta spalva [47]

Pirmiausiai, originalus vaizdas paruošiamas, konvertuojamas į vieno kanalo vaizdą ir daromas suliejimas. Pritaikomas kraštų aptikimo metodo algoritmas. Vaizdo paruošimas reikalingas siekiant sumažinti klaidingai aptiktų kraštų kiekį. Krašto aptikimo metodo esmė – aptikti staigius spalvos perėjimus. Aptikti kraštai tikrinami pagal didžius ir klasifikuojami. Pagal kraštus ir poslinkius nuo jų generuojama potencialios nusileidimo vietos kaukė, pavaizduota 3 pav. C dalyje. Kauke pritaikoma prie originalaus vaizdo. Papildomai klasifikuojamas potencialios nusileidimo vietos žemės paviršius pagal spalvą, formą ir kitus parametrus. Potencialios nusileidimo vietos aptikimo procesas kartojasi, kol skraidantis robotas atlieka nusileidimo procesą. Šios sistemos potencialios nusileidimo vietos aptikimas buvo vertinamas pagal dvejetainio klasifikavimo statistinius rodiklius [68]. Tikslumo rodiklis be žemės paviršiaus klasifikuojamo sudaro 86,3%, su 94,7% jautrumu.

Kaip rašo Yingcai Bi ir kt. publikuotame straipsnyje „*Implementation of autonomous visual tracking and landing for a low-cost quadrotor*“ [1] skraidančio roboto sistema naudodama vaizdo atpažinimo algoritmus ieško nusileidimo platformos. Įterptinė sistema skirta atvejams, kai dingsta GPS signalas. Skraidančio roboto valdymui panaudojamas PID reguliatorius. Sistema naudojanti vaizdus, kurių rezoliucija yra 320x240 px, aptinka iš dalies matematiškai pagrįstą žymeklį (5 pav.). Vaizdų apdorojimas atliekamas AR.Drone skraidančio roboto valdiklyje, kurio parametrai: procesorius ARM9 468 MHz, atminties kiekis 128MB.



4 pav. Skraidančio roboto žymeklis, naudojamas kaip nusileidimo platforma [1]

Žymeklis sudarytas iš kelių elementų: žalios „H“ raidės ir raudono ir mėlyno stačiakampių. Stačiakampiai skirti nustatyti žymeklio kryptį. Žymeklio atpažinimo procesas atliekamas konvertuojant RGB vaizdą į vieno kanalo vaizdą, naudojant binarizacijos slenksčio nustatymą ir kontūrų aptikimo metodus. Šios sistemos rezultatas pavaizduotas 5 pav.



5 pav. Žymeklio atpažinimas: a) originalus vaizdas; b) aptiktas žymeklio kontūras [1]

Eksperimento metu skraidymo aukštis buvo nedidesnis nei 1m. Eksperimentas buvo atliktas esant geroms apšvietimo sąlygoms. Taip pat buvo testuojamas žymeklio sekimas, kai žymeklis buvo įrengtas ant radijo bangomis valdomo automobilio modelio.

Skraidančio roboto trūkumas – tai maža skraidymo trukmė [2], o akumulatoriaus talpos padidinimas neišsprendžia šios problemos. Žymeklių atpažinimas gali būti pritaikytas skraidančių robotų sistemose, ieškant pažymėtų platformų, kuriose nusileidęs robotas gali pasikrauti akumuliatorių [2]. Mokslininkai siūlo įdiegti specialų nusileidimo ant pakrovimo platformos režimą, kuris automatiškai įsijungia, kai akumulatoriaus įkrovimas bus mažesnis kaip $\frac{1}{2}$ maksimalios akumulatoriaus talpos.

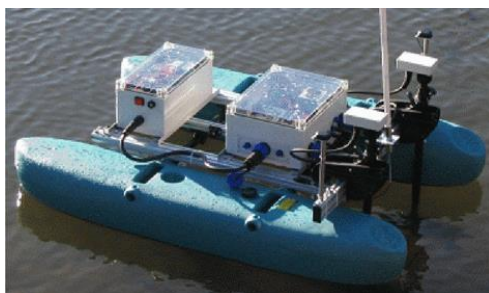


6 pav. Skraidantis robotas su akumulatoriaus bevielio parkovimo sistema [2]

Vien GPS pozicionavimo duomenų naudojimas negali užtikrinti skraidančio roboto tikslaus nusileidimo. Paprastai GPS neapibrėžtis seka iki 3 metrų. Naudojant GPS su antžeminėmis stotimis, neapibrėžtis sumažėja iki 1 cm, tačiau yra keletas svarbių faktorių: kokių palydovų signalus gauna GPS modulis ir antžeminė stotis, koks skraidančio roboto mobilusis interneto ryšis, koks skraidančio roboto mobiliojo interneto ryšio su antžemine stotimi atsako greitis. Todėl GPS pozicionavimo duomenų naudojimo neužtenka užtikrinti skraidančio roboto pozicionavimo

tikslumą nusileidimo metu. Skraidančio roboto nusileidimo tikslumas gali būti padidintas naudojant kompiuterinį regėjimą.

Ricardo Neves ir kt. publikuotame straipsnyje „*Raspberry Pi based stereo vision for small size ASVs*“ [48] pasiūlyta sistema sukurta panaudojant *OpenCV* biblioteka, bibliotekos plėtinį objekto sekimui „*cvBlobsLib*“ ir pritaikyta prie *Raspberry Pi*. Pasiūlyta sistema apdoroja vaizdus gautus iš kamerų ir gali komunikuoti per *WiFi* su pagrindiniu serveriu. Sistema įdegtą į „Gama“ katamaraną (7 pav.).



7 pav. Katamaranas su integruota vaizdo apdorojimo sistema „*Raspberry Pi*“ mikrovaldiklyje [48]

Raspberry Pi kompiuteris buvo specialiai pasirinktas dėl mažos savikainos, lyginant su kitais kompiuteriais, o jo operacijų apdorojimo greičio visiškai užtenka sistemos funkcionavimui. Objekto identifikavimo funkcija naudoja „*cvBlobsLib*“ plėtinį, kuriame yra nepageidaujamų objektų filtravimo iš scenos vaizdo funkcijos. Taip pat šis plėtinys leidžia įvertinti tokius parametrus kaip objekto orientaciją, šiurkštumą ir kt.

Sekamas objektas – tai plokštuma, kurios dydis 50x30 cm su specifine tekstūra. Tekstūra sudaryta iš dviejų stačiakampių vienspalvių regionų (8 pav).



8 pav. Sekamo objekto tekstūra [48]

Vaizdo rezoliucija sudaro 320x240 px. Kai vaizdas iš kameros matricos nuskaitytas, algoritmas konvertuoja spalvos formatą iš *RGB* į *HSV* spalvų erdvę. Tai standartinė procedūra, nes *HSV* spalvų erdvė mažiau priklauso nuo aplinkos apšvietimo [12], [37]. Algoritmas atpažįsta trijų tipų objektus:

- sekamas objektas;
- pagalbinis objektas dešinėje plūduri pusėje;
- pagalbinis objektas kairėje plūduri pusėje.

Dėl dinaminės aplinkos, duomenys apie objektus, kurie buvo apskaičiuoti iš scenos vaizdo, saugomi atskirai. Tai leidžia greičiau identifikuoti objektus, nustatyti poziciją, orientaciją ir atstumą. Objektas po atpažinimo sekamas pagal *Lucas-Kanade* metodą [49]. Metodo esmė –

palyginti dabartinį ir prieš tai buvusios scenos vaizdus. Pagal scenos skirtumą nustatoma sekamo objekto nauja pozicija. Kai atpažinimo algoritmas apskaičiuoja visus reikiamus duomenis, jie siunčiami į pagrindinį serverį.

1.6 Skyriaus apibendrinimas

GPS pritaikymo objekto sekimo sistemai nevisada pakanka tiksliam objekto pozicionavimui. GPS pozicionavimas su ir be antžeminių stočių koregavimo duomenimis turi šiuos trūkumus:

- pozicijos neapibrėžtis siekia iki 3 metrų.
- pozicijos neapibrėžtis sumažėja iki 1 cm, priklauso nuo:
 - ar vienodų palydovų signalai gauna GPS modulis ir antžemine stotis;
 - koks skraidančio roboto ryšio su antžemine stotimi mobiliuoju internetu atsako greitis.

Šiai problemai spręsti galima pritaikyti vaizdo atpažinimo metodus, kurie padėtų skraidančiam robotui tiksliai atpažinti ir pozicionuoti sekamą objektą. Apžvelgus mokslinėje literatūroje panašių sistemų, naudojamus komponentus, skraidančio roboto objekto atpažinimo ir sekimo įterptinei sistemai kurti pasirinkti binarizacijos slenksčio nustatymas, kontūro aptikimo algoritmai. Objektui pažymėti pasirinktas QR pagrindu žymeklio sudarymo principas. Objekto sekimui ir pozicijos prognozavimui reikalinga naudoti Kalman filtrą.

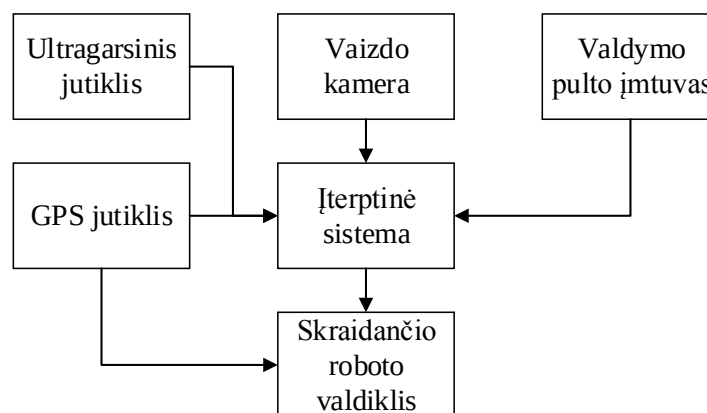
2 OBJEKTO ATPAŽINIMO IR SEKIMO METODŲ IR PRIEMONIŲ PRITAIKOMUMAS SKRAIDANČIŲ ROBOTŲ SISTEMOSE

Objekto atpažinimo ir sekimo metodų veiksmingumas gali skirtis skirtingose platformose, tai priklauso nuo platformos savybių. Skraidančio roboto platformos masė, procesorinių skaičiavimo, elektros energijos resursai yra labai apriboti, todėl, kuriant objekto atpažinimo ir sekimo įterptinę sistemą skraidančiam robotui, reikalingi specifiniai kompiuterinės regos, objekto pažymėjimo, identifikavimo ir sekimo metodai.

2.1 Skraidančio roboto vaizdo atpažinimu grindžiama objekto sekimo įterptinė sistema

Objekto sekimo įterptinės sistemos pagrindinė paskirtis atlikti aplinkos suvokimą – t.y. aplinkoje pažymėto objekto atpažinimą ir sekimą. Žymekliu pažymėtas sekamas objektas gali būti naudojamas kaip skraidančio roboto nusileidimo platforma. Skraidantis robotas pagal GPS signalą priartėjęs prie sekamo objekto, įsijungia kompiuterinę regėjimo sistemą ir pradeda žymeklio atpažinimo procesą. Kai žymeklis atpažintas skraidantis robotas pozicionuojamas taip, kad žymeklis būtų vaizdo centre. Pradedant skraidančio roboto nusileidimo procesą ant žymekliu pažymėtos platformos įterptinė sistema papildomai naudoja ultragarsinių jutiklių duomenis apdorojant vaizdo kadrus.

Įterptinė sistema naudoja keletą papildomų modulių. Svarbiausias – vaizdo kameros modulis, kurio pagalba įterptinė sistema, naudojant aplinkos vaizdus, galės aptikti ir sekti žymeklį. Prototipinė įterptinė sistema, naudojami moduliai ir komunikacija su skraidančiu robotu pavaizduota 9 pav.



9 pav. Skraidančio roboto vaizdo atpažinimu grindžiamą objekto sekimo įterptinės sistemos modelis ir jo komunikacija su skaidančio roboto valdikliu

Skraidančio roboto valdiklis, gavęs komandą iš įterptinės sistemos, atlieka reikiamą funkciją. Įterptinė sistema nuotoliniu būdu valdoma su pultu, maitinama skraidančio roboto

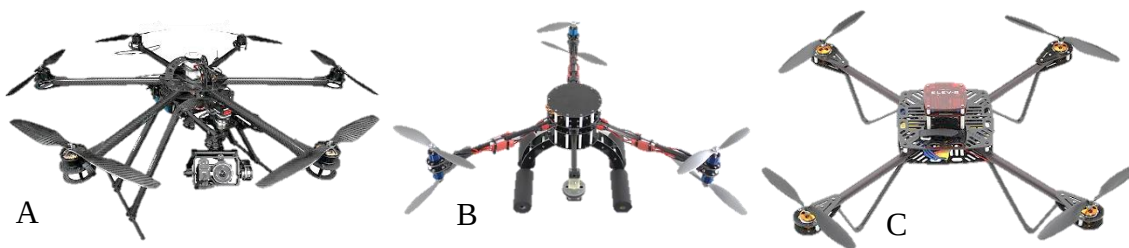
akumuliatoriumi. Sistema veikia skirtingais režimais, kuriuos galima pakeisti su valdymo pultu, keičiant permetamojo jungiklio padėtį:

- Įterptinė sistema naudojama kompiuterinį regėjimą aptinka reikiamą objektą ir jį seką siųsdama valdymo komandas į skraidančio robotą valdiklį. Skraidantis robotas veikia autonomiškai, todėl visos kitos valdymo komandos, kurios gaunamos iš pulto, yra ignoruojamos.
- Įterptinė sistema tiesiogiai perduoda gautus signalus iš valdymo pulto į skraidančio robotą valdiklį. Visi signalai perduodami be papildomų apdorojimų.

Vaizdo kamera suteikia galimybę robotui suvokti aplinką, analizuoti ir rasti reikiamą sprendimą pagal užduotą užduotį. Vaizdo apdorojimui reikalingiems matematiniams skaičiavimams atlikti naudojama *OpenCV* biblioteką, kurios pagalba galima realizuoti objekto atpažinimą ir sekimą panaudojant standartinius vaizdų apdorojimo metodus arba juos modifikuoti pagal poreikius. Visi vaizdo apdorojimo algoritmai atliekami skraidančiame robote integruotame valdiklyje, todėl objekto atpažinimo algoritmas turi būti kuo paprastesnis ir greičiau apdorojamas, tačiau kartu neprarandant vaizdo apdorojimo kokybės. Kameros rezoliucija ir vaizdo kokybė daro didelę įtaką objekto atpažinimui. Pavyzdžiui, mažesnės rezoliucijos vaizdas leidžia sumažinti triukšmo kiekį. Vaizdas su mažesniu triukšmo kiekiu duoda galimybę efektyviau analizuoti scenos vaizdą, be to, mažesnės rezoliucijos vaizdas greičiau apdorojamas.

2.2 Kompiuterinė rega skraidančiuose robotuose

Mokslinėje literatūroje buvo apžvelgtos kelios skirtingos skraidančio robotą kompiuterinio regėjimo koncepcijos. Prieš nustatant tinkamiausią sprendimą reikia atsižvelgti į skraidančių robotų savybes. Skraidantis robotas – tai platforma su lyginiu skaičių rotorių, kurie sukasi priešingomis įstrižainių kryptimis. Keli skraidantys robotai pavaizduoti 10 pav.



10 pav. Skraidantis robotas: a) *Hexacopter*; b) *Quietcopter RC Multi*; c) *Parallax ELEV-8*

Propeleriai sukami su rotoriais kurių sukimo greitis yra kontroliuojamas. Transporto priemonė valdoma keičiant vieno ar kelių propelerių sukimosi greitį. Skraidantys robotai skiriasi pagal rotorių skaičių, kurių gali būti 3-8 ir daugiau, pagal naudojamus valdiklius ir jų parametrus. Nuo jų skaičiaus ir parametrų priklauso skraidančio robotą stabilumas, keliamoji galia,

maksimalus greitis. Taip pat įtakos turi ir tai koku atstumu pritvirtinti rotoriai su propeleriais prie skraidančios roboto platformos. Skraidantis robotas būtinai turi būti komplektuojamas su akumuliatoriumi. Atsižvelgiant į skraidančio roboto, pagrindinius apribojimus (masę, akumuliatoriaus talpą ir jų savybes), kompiuterines regėjimo sistemai keliami reikalavimai:

- kompiuterinės regėjimo sistemos masė turi būti minimali,
- skraidančio roboto kompiuterinio regėjimo sistemos energijos sąnaudos turi būti minimalios,
- kompiuterinio regėjimo sistemos atsako reakcija skraidančio roboto valdymui turi būti minimali.

Pagal šiuos reikalavimus tinkamos dvi kompiuterinio regėjimo sistemos koncepcijos. Pirmos koncepcijos pagrindinis principas tai, kad visi apskaičiavimai atliekami nuotoliniu būdu. Kameros nuskaitytas vaizdas atpažinimo sistemai perduodamas bevieliu ryšiu, sistema apdoroja vaizdą, priima sprendimą ir valdymo signalus perduoda bevieliu ryšiu atgal į skraidantį robotą. Pagrindinis šios sistemos trūkumas tai, kad gaunamas vaizdas analoginių pavidalų yra triukšmingas. Didinant atstumą vaizdo kokybė dar labiau sumažėja. Toks efektas buvo pastebėtas apžvelgiant mokslinį literatūrą [50], 11 pav. „a“ dalyje parodytas automobilio važiavimo metų gaunamas vaizdas.



11 pav. Nuotolinio vaizdo perdavimo palyginimas: a) skaitmeninis signalas; b) analoginis signalas

Perduodant vaizdus skaitmeninių pavidalų vaizdas matomas be triukšmų, tačiau vaizdas gaunamas su vėlavimu. Vėlavimą galima pastebėti lyginant 11 pav. „a“ ir „b“ vaizdus, filmuojamo automobilio greitis yra didesnis negu filmuojančiojo. Šis trūkumas susijęs su atsako reakcija, kada atsiranda vėlinimas apdorojant ir perduodant vaizdo ir valdymo signalus. Mūsų atveju skraidantis robotas turi reaguoti į aplinkos pakeitimus iškart, todėl ši vaizdo apdorojimo koncepcija nėra tinkama.

Antro sprendimo esmė tai, kad visus vaizdo apdorojimui reikalingus skaičiavimus atlieka robotas. Šiuo atveju nereikalingas vaizdo perdavimas bevieliu ryšiu, todėl kompiuterinio regėjimo sistema gauna geros kokybės vaizdus, be papildomai atsirandančių triukšmų nuskaitymo metu. Šis sprendimas taip pat turi trūkumą – kompiuterinė regėjimo sistema padidina skraidančio roboto elektros energijos sąnaudas. Tačiau siekiant sumažinti skraidančio roboto elektros energijos

sąnaudas ankstesniuose tyrimuose [32] buvo nustatyta kad naudojant skraidančio roboto pagreičių duomenis, galima prognozuoti vaizdo kameros nuskaitymo kokybę, ir perjungti sistemos būseną į budėjimo režimą laukiant geresnės kokybės vaizdų.

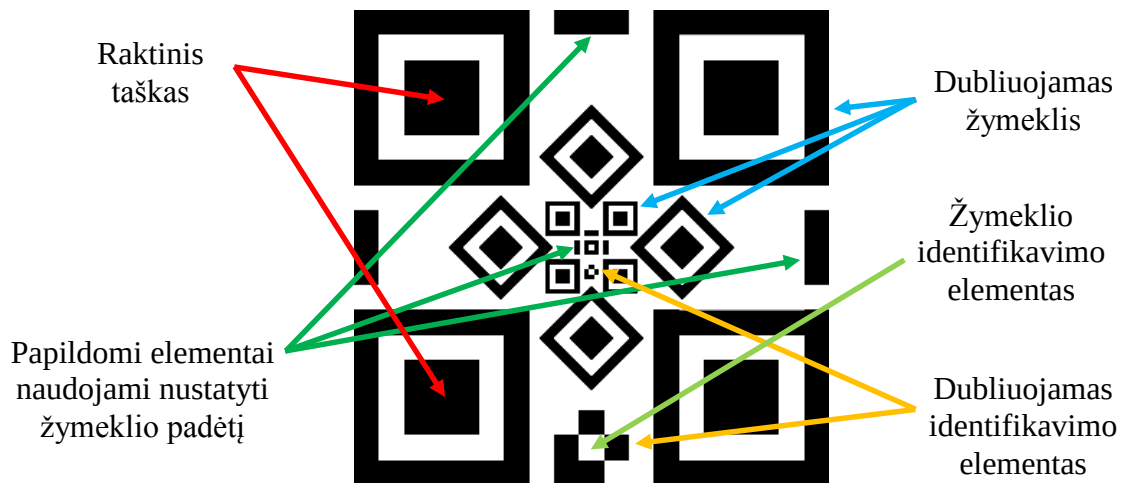
2.3 Sekamo objekto žymeklio sudarymas

Kadangi, įterptinė objekto atpažinimo ir sekimo sistema kartu valdys ir skraidantį robotą, tai bendra vaizdo nuskaitymo, apdorojimo, žymeklio atpažinimo ir valdymo veiksmų apskaičiavimo trukmė turi būti minimali. Atpažinimo ir sekimo sistema turi identifikuoti žymeklį ir priskirti jam unikalų numerį. Skraidantis robotas turi turėti galimybę atpažinti objektus iš kuo didesnio atstumo. Kaip buvo minėta anksčiau, žymeklis bus panaudotas kaip skraidančio roboto nusileidimo platformą.

Remiantis nustatytomis sąlygomis žymeklis turi turėti tokios savybės:

- greitai atpažįstamas ir sekamas,
- turėti priskirtą identifikacinį numerį,
- būti atsparus vaizdo nuskaitymo triukšmams,
- atpažįstamas tiek iš didesnių, tiek ir iš mažesnių atstumų.

Pagal šiuos nustatytos kriterijus žymeklis buvo sudaromas QR kodo pagrindu. QR privalumas – tai matematiškai pagrįstas raktinio kodo atpažinimas. Taip pat QR kodas pagal savo sudarymą panašus į ReacTIVision žymeklius – turi staigius kraštus. Nuskaitant kameros vaizdą, kai yra didelis roboto pagreitis arba vibracijos, staigūs kraštai tampa neraiškūs, tačiau po slenkstinės binarizacijos metodo panaudojimo žymeklio kraštai atgauna beveik originalų pavidalą. Žymeklio numerio identifikavimui panaudotas VisualCode žymeklio sudarymo principas. Siekiant atpažinti žymeklį tiek didesniais atstumais tiek mažesniais (kada vykdomas nusileidimas ant žymeklio platformos), žymeklis yra dubliuojamas keletą kartų. Nusileidimo metu pats didžiausias žymeklis išeina už kameros matomumo ribų, o mažesnis žymeklis tampa aiškiai matomas. Mažesnis žymeklis iš didesnio atstumo sudarys bendro vaizdo mažą dalį, todėl kraštai susilies ir žymeklis bus neatpažįstamas. Suprojektuotas žymeklis pavaizduotas 12 pav.



12 pav. Suprojektuotas žymeklis

Žymeklis sudarytas iš 4 raktinių taškų ir identifikavimo numerio. Žymeklis yra dubliuojamas du kartus. Žymeklio identifikavimo elementas dubliuojamas vieną kartą. Dubliuojamas žymeklio identifikavimo elementas skirtas atpažinti nusileidimo platformos numerį, kai skraidantis robotas dar yra ant pačios platformos. Skraidymo metų robotas atpažįsta nusileidimo platformos numerį, ir pagal žymeklio padėtį pradeda jo sekimą. Nusileidimo metų žymeklio identifikavimo elementas išeina iš kameros matymo ribos, tačiau sekant objektą sistema žino žymeklio identifikavimo numerį. Papildomi elementai (12 pav.) naudojami nustatyti žymeklio padėtį, jie pagrįsti 2D žymeklio grotelių taškų išdėstymo principu.

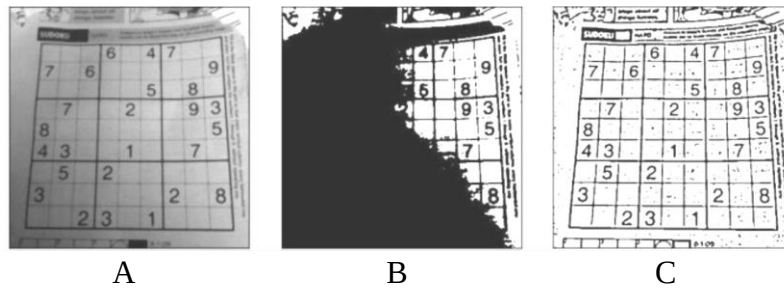
2.4 Vaizdo apdorojimo metodai, sekamo objekto atpažinimas

Pagrindiniai žymeklio atpažinimo principai buvo numatyti projektuojant žymeklį. Pirmasis žingsnis žymeklio atpažinimui – tai vaizdo konvertavimas į dvejetainį vaizdą (binarizacija). Vaizdo konvertavimas atliekamas naudojant skirtingus binarizacijos slenksčio nustatymo metodus. Formulėje (1) parodomas paprastas vieno kanalio vaizdo konvertavimas į dvejetainį vaizdą.

$$dst(x,y) = \begin{cases} maxval & \text{if } src(x,y) > thresh \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

kur $src(x,y)$ – vieno kanalio vaizdo matricos elementas, $maxval$ – maksimali pikselio vertė, $thresh$ – slenksčio ribos parametras, $dst(x,y)$ – dvejetainio vaizdo matricos elementas.

Paprastas binarizacijos slenksčio nustatymas nuo adaptyvaus skiriasi slenksčio ribos parametro koregavimu įvertinant šalia esančių pikselių spalvas. Adaptyvų metodą geriau naudoti, kai atpažįstamas objektas yra apšviestas netolygiai. Metodų palyginimas pavaizduotas 13 pav.



13 pav. Paprastas ir adaptyvus binarizacijos slenksčio nustatymo metodai: a) Originalus vaizdas; b) Paprastas binarizacijos slenksčio nustatymas; c) Adaptyvus binarizacijos slenksčio nustatymas

Apžvelgiant mokslinę literatūrą buvo pastebėti esamų metodų trūkumai. Tačiau efektyviausias metodas – binarizacijos slenksčio nustatymas naudojant vietinius blokus. Kiekvienam vietiniam blokui pritaikomas paprastas binarizacijos slenksčio nustatymo metodas. Slenksčio ribos parametras apskaičiuojamas įvertinant tamsiausią ir šviesiausią pikselius. Ankstesniame tyrime [32] buvo naudojamas pagerintas binarizacijos slenksčio nustatymo metodas naudojant vietinius blokus. Pasiūlytas sprendimas leidžia apjungti keletą rezultatų gautų vietinių blokų binarizacijos slenksčio nustatymo metu naudojant skirtingus parametrus. Algoritmas sudarytas iš trijų etapų. Pirmas etapas, tai binarizacijos slenksčio nustatymas naudojant vietinius blokus. Antras etapas, pirmo etapo pakartojimas su vietinių blokų pasislinkimu į pirmo etapo vietinių blokų centrus. Trečias etapas – pirmo ir antro etapų dvejetainio vaizdų loginis apjungimas. Toks sprendimas žymiai pagerina konvertavimo kokybę į dvejetainį vaizdą ir padidina metodo atsparumą vibracijos triukšmams [32]. Kitas būdas padidinti sistemos efektyvumą – patobulinti binarizacijos adaptyvaus slenksčio nustatymo metodą, kuris būtų yra atsparesnis vibracijoms. Patobulintas metodas apjungia keletą binarizacijos su adaptyviu slenksčio nustatymu rezultatų. Originalus vaizdas ir po binarizacijos slenksčio metodo pritaikymo rezultatas parodytas 14 pav.



14 pav. Rezultatai panaudojant skirtingus binarizacijos slenksčio nustatymo metodus: a). originalus vaizdas, b). paprastas binarizacijos slenksčio nustatymas, c) adaptyvus binarizacijos slenksčio nustatymas, d) binarizacijos slenksčio nustatymas naudojant vietinius blokus, e) pagerintas metodas

naudojantis vietinių blokų binarizacijos slenksčio nustatymo rezultatus, d) patobulintas adaptyvus binarizacijos slenksčio nustatymas

Antrasis žingsnis žymeklio atpažinimui – tai kontūrų paiešką iš dvejetainio vaizdo. Kontūrų paieška panaši į balto objekto išrinkimą iš juodo fono. Kontūrų paieškai naudojamas Satoshi Suzuki sukurtas algoritmas [51]. Kiekvienas aptiktas kontūras turi savo parametrus – perimetrą, plotį, masės centrą. Raktinių taškų suradimui atliekamas kiekvieno kontūrų palyginimas su kitais kontūrais:

$$P_{check} \in \left(P \pm (P \cdot K_{P_{error}}) \right) \cdot R_P, \quad (2)$$

$$S_{check} \in \left(S \pm (S \cdot K_{S_{error}}) \right) \cdot R_S, \quad (3)$$

kur R_P – didesnio ir mažesnio kvadratų perimetrų proporcija, P – didesnio ir mažesnio kvadrato perimetras, $K_{P_{error}}$ – perimetro paklaidos koeficientas, P_{check} – mažesnio ir didesnio kvadrato palyginimo perimetras, R_S – didesnio ir mažesnio kvadratų plotų proporcija, S – didesnio ir mažesnio kvadrato plotas, $K_{S_{error}}$ – ploto paklaidos koeficientas, S_{check} – mažesnio ir didesnio kvadrato palyginimo plotas.

Jei kvadratus perkelti į tinklėlį, tai mažesnis kvadratas sudarytas iš 3x3, o didesnis kvadratas iš 7x7. Galima apskaičiuoti jų perimetrus ir plotus. Mažesnio kvadrato perimetras 12, plotas 9. Didesnio – 28 ir 49 atitinkamai. Dalinant gautus skaičius, gauname R_P ir R_S proporcijas. Kvadratų parametrai palyginami tarpusavyje. Išskaičiuojami kiekvienam kontūrai erdviniai momentai pagal formulę 4:

$$m_{ji} = \sum_{x,y} (array(x,y) \cdot x^j \cdot y^i), \quad (4)$$

kur (x,y) – dvejetainio vaizdo pikselio koordinatės, m – erdviniai momentai. Kontūro masės centras $(\bar{x}, \bar{y})$ skaičiuojamas kaip:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}. \quad (5)$$

Jei gauti parametrai atitinka pagal numatytą paklaidą su lyginamo kontūro parametrais ir jų masės centrai taip pat pagal numatytą paklaidą sutampa, tai mažesnis ir didesnis kvadratai sudaro vieną raktinį tašką. Apsikaičiojami visi įmanomi raktiniai taškai.

Trečias žingsnis – kontūrų rūšiavimas ir grupavimas. Nusileidimo metu kamerą su vaizdo kameros stabilizacijos sistemos pagalba visada nukreipta į žymeklio plokštumą. Tai kelių raktinių taškų plotis bus apytiksliai vienodas. Raktinio taško plotas tai didesnio kvadrato plotas. Rūšiavimas atliekamas lyginant raktinių taškų plotus:

$$A_{check} \in A \pm (A \cdot K_{A_{error}}), \quad (6)$$

kur A – raktinio taško plotas, $K_{A_{error}}$ – raktinio taško ploto palyginimo paklaida, A_{check} – tikrinamo raktinio taško plotas.

Surūšiuoti pagal dydį raktiniai taškai turi būti sugrupuoti pagal šalio esančios kaimynus. Raktinis taškas gali būti kito raktinio taško kaimynu, jei atstumą, tarp jų pozicijų, galima matematiškai aprašyti naudojant raktinio taško dydį ir Euklido atstumą. Kaimynas priklauso tai pačiai grupei, kai atstumas tarp raktinių taškų pozicijų atitinka pagal numatyta paklaidą. Galimi atstumai iki kaimynų apskaičiuojami:

$$D = \sqrt{(x_0 - x)^2 + (y_0 - y)^2}, \quad (7)$$

$$D_{cathetus_{check}} \in D \cdot \sqrt{A} \pm (D \cdot \sqrt{A} \cdot K_{D_{error}}) \cdot R_{cathetus}, \quad (8)$$

$$D_{hypotenuse_{check}} \in D \cdot \sqrt{A} \pm (D \cdot \sqrt{A} \cdot K_{D_{error}}) \cdot R_{hypotenuse}, \quad (9)$$

kur D – atstumas tarp raktinių taškų masės centrų, x_0, y_0 – raktinio taško masės centras, x, y – tikrinamo raktinio taško masės centras, A – raktinio taško plotis, $K_{D_{error}}$ – atstumo paklaidos koeficientas, $R_{cathetus}$ – atstumo ir raktinio taško pločio šaknies proporcija, $R_{hypotenuse}$ – diagonalinio atstumo ir raktinio taško pločio šaknies proporcija, $D_{cathetus_{check}}$ – palyginimo atstumas iki tikrinamo kaimyno, $D_{hypotenuse_{check}}$ – diagonalinis palyginimo atstumas iki tikrinamo kaimyno.

Surūšiuoti ir sugrupuoti raktiniai taškai dabar sudaro žymeklio dalis. Žymeklis sudarytas iš trijų skirtingo mastelio dubliuojančių dalių. Apskaičiuoti žymekliai papildomai palyginami pagal savo centrus.

$$(x_{check}, y_{check}) \in (x, y) \pm ((x, y) \cdot K_{A_{error}}), \quad (10)$$

kur (x, y) – žymeklio centras, (x_{check}, y_{check}) – palyginimo žymeklio centro ribos, $K_{A_{error}}$ – žymeklio centro paklaidos.

Kiekvienas žymeklis su kitu žymekliu tikrinamas po vieną kartą, jei žymeklių centrai atitinka (su apibrėžta paklaida), tai reiškia, kad kiekvienas toks žymeklis yra dalis ieškomo žymeklio.

2.5 Objekto identifikavimas ir sekimas

Atpažinti žymekliai turi būti identifikuoti. Žymeklio identifikavimas sekimo procese leidžia nesupainioti žymeklius, kai jie būna labai arti vienas kito, be to padeda nustatyti į kurią platformą turės nusileisti skraidantis robotas. Kiekvienam kadru žymeklio atpažinimo ir identifikavimo procesas kartojamas iš naujo. Identifikatorius panašus į VisualCode žymeklį, kuris užkoduojamas kauke, sudaryta iš 3x3 matricos. Žymeklio sekimas atliekamas vertinant žymeklio pozicijas dabartinio ir buvusio vaizdo kadro atžvilgiu. Tačiau žymeklio atpažinimas gali būti nesėkmingas vieną ar keletą kadrų dėl blogos vaizdo kokybės, ar kitų išorinių priežasčių. Todėl būtinas žymeklio pozicijos prognozavimas, kuris ypač reikalingas kai žymeklis sekamas nusileidimo metu

– kai robotas autonomiškai priima sprendimą pagal gautą vaizdą. Vienas iš būdų šiai problemai spręsti – Kalmano filtro taikymas.

Kalmano filtras tai rekursinis filtras, kuris prognozuoja tiesinio dinaminio proceso būseną pagal triukšmingus ir nepilnus matavimus. Naudojant Kalmano filtrą žymeklio pozicijos prognozavimui kiekviename kadre atpažinti žymekliai susiejami su praeitais kadrais. Tokiu būdu Kalmano filtras prognozuoja dabartinę žymeklio poziciją pagal buvusias pozicijas [31]. Kalmano filtras aprašomas formulėmis 11-13. Kadro apdorojimo trukmė apskaičiuojama:

$$\Delta t = t_c - t_p, \quad (11)$$

kur t_c – dabartinio kadro apdorojimo pradžios laikas, t_p – buvusio kadro apdorojimo pradžios laikas, Δt – buvusio kadro apdorojimo trukmė. Žymeklio judėjimo greitis vaizdo atžvilgiu skaičiuojamas:

$$(v_x, v_y) = \frac{(x_c, y_c) - (x_p, y_p)}{\Delta t}, \quad (12)$$

kur (x_c, y_c) – dabartinė žymeklio pozicija, (x_p, y_p) – buvusi žymeklio pozicija, (v_x, v_y) – žymeklio judėjimo greitis. Žymeklio pozicijos prognozavimas atliekamas naudojant praeito kadro apdorojimo trukmę ir žymeklio judėjimo greitį. Žymeklio pozicijos prognozavimas, apskaičiuojamas:

$$(x_k, y_k) = (x_p, y_p) + (v_x, v_y) \cdot \Delta t, \quad (13)$$

kur (x_p, y_p) – buvusi žymeklio pozicija, (v_x, v_y) – žymeklio judėjimo greitis, Δt – buvusio kadro apdorojimo trukmė, (x_k, y_k) – žymeklio prognozuojama pozicija.

Sistema praradus žymeklį pasilieka prognozavimui tik 5 kadrus. Jei nusileidimo platforma pamesta, kompiuterinio regėjimo sistema perjungia skraidantį robotą į autonominio skraidymo režimą pagal GPS signalą.

2.6 Skyriaus apibendrinimas

Siekiant pritaikyti vaizdo atpažinimu grindžiamus objekto atpažinimo ir sekimo metodus skraidantiems robotams turi būti parenkami sprendimai reikalaujantys mažiausiai sistemos resursų, tačiau pakankamai efektyvūs kokybiškai greitai apdorojantys vaizdo informaciją. Norint kuo mažiau prarasti kamera gaunamos informacijos, siūloma vaizdo apdorojimą atlikti skraidančio roboto platformoje. Taikant patobulintą vaizdo binarizacijos adaptivaus slenksčio nustatymo algoritmą bei žymeklio pozicijos prognozavimą galima užtikrinti efektyvų (sistemos resursų suvartojimo atžvilgiu) ir atsparų vaizdo nuskaitymo triukšmams objekto atpažinimą ir jo sekimą. Suprojektuotas matematiškai apibrėžtas QR žymeklis, atsižvelgiant į siūlomus sprendimus, skirtas užtikrinti matomumą kadre iš didelio ir mažo atstūmų iki skraidančio roboto.

3 SKRAIDANČIO ROBOTO OBJEKTO ATPAŽINIMO IR SEKIMO ĮTERPTINĖS SISTEMOS PROJEKTAVIMAS IR PROTOTIPO KŪRIMAS

Žymeklio atpažinimo ir sekimo sistemos ir įterptinės sistemos projektavimas susideda iš kelių žingsnių:

- funkcinių ir nefunkcinių reikalavimų įterptinei sistemai iškėlimas;
- valdiklio, vaizdo kameros ir kitų komponentų pasirinkimas įterptinei sistemai;
- techninės įterptinės sistemos projektavimas ir surinkimas;
- programinės įterptinės sistemos projektavimas, programavimas;
- įterptinės sistemos verifikavimas pagal iškeltus reikalavimus;
- įterptinės sistemos integravimas į skraidančio robotą platformą.

3.1 Skraidančio roboto vaizdo atpažinimu grindžiamos objekto sekimo įterptinės sistemos reikalavimai

Kuriamos sistemos tikslas – žymeklio atpažinimas ir jo sekimas esant sudėtingomis aplinkos sąlygomis. Žymeklis gali būti įvairiose padėtyse, skirtingose scenos vaizdo vietose. Sistema privalo atpažinti objektą esant bet kokiam aplinkos apšvietimui. Kuriant įterptinę objekto atpažinimo ir sekimo sistemą skraidančiam robotui, buvo išskirti funkciniai ir nefunkciniai reikalavimai.

Funkciniai reikalavimai:

1. sistema turi būti autonominė;
2. sistema turi atpažinti objektą bet kokioje scenos vaizdo dalyje;
3. sistema turi atpažinti objektą esant kintančiam aplinkos apšvietimui;
4. sistema turi sekti atpažintą objektą;
5. sistema turi komunikuoti su skraidančio roboto valdikliu ir siųsti valdymo komandas;
6. skraidančiam robotui sekant objektą sistema turi priimti sprendimus ir sugeneruoti robotą valdymo komandas, pagal įvertintus scenos duomenis;
7. maksimalus atstumas iki kurio sistema turi aptikti objektą - 15 m;
8. tamsiu metu markeris turi būti apšviestas;
9. jei kompiuterinio regėjimo sistema neaptinka sekamo objekto, tai įjungiamas skraidančio robotą skraidymo režimas pagal GPS signalą;
10. sistema privalo turėti galimybę perjungti savo veikimo režimus:
 - a. autonominis skraidymas pagal GPS signalą;

- b. pritaikytas prie vaizdo atpažinimo sistemos, maksimalus skraidymo nuokrypis sudaro nedaugiau kaip 2 laipsnius;
- c. rankinis valdymas.

Nefunkciniai reikalavimai:

1. surinktos sistemos masė turi būti nedidesnė negu 600 g;
2. sistema turi vartoti kuo mažiau elektros energijos, nedaugiau kaip 15 W;
3. sistema turi apdoroti ne mažiau 10 vaizdo kadrų per sekundę;
4. sistemos savikaina turi būti nedidesnė nei 250 €;
5. skraidantis robotas turi būti papildomai sukomplektuotas su šviesos šaltiniu, kuris padės apšviesti scenos vaizdą;
6. scenos vaizdo matymo laukas neturi būti užstotas skraidančio roboto korpuso ar kitais jo elementais.

3.2 Valdiklio parinkimas vaizdo atpažinimu grindžiamai skraidančio roboto objekto sekimo įterptinei sistemai

Dažniausiai valdiklis valdo konkrečios paskirties įrenginį arba yra pagalbinė įprasto kompiuterio dalis, atsakinga už tam tikrus veiksmus. Šiuo metų valdiklius galima skirstyti į dvi grupes – galingesnius ir lėtesnius [1], [52]. Lėtesni valdikliai yra pigūs. Jie turi gana primityvius bendros paskirties procesorius, kurių dažnis nedidesnis negu 32 MHz, bei nedidelį kiekį atminties – kelis kilobaitus. Galingesni valdikliai yra brangesni. Dažniausiai tai būna ARM architektūros procesoriai. Procesorių dažnis gali viršyti 900 MHz. Tokius valdiklius galima pavadinti kompiuteriu, nes turi visus reikiamus išėjimus ir įėjimus, galima prijungti ekraną, klaviatūrą ir pelę.

Parenkant valdiklį iškeliami funkciniai ir nefunkciniai reikalavimai.

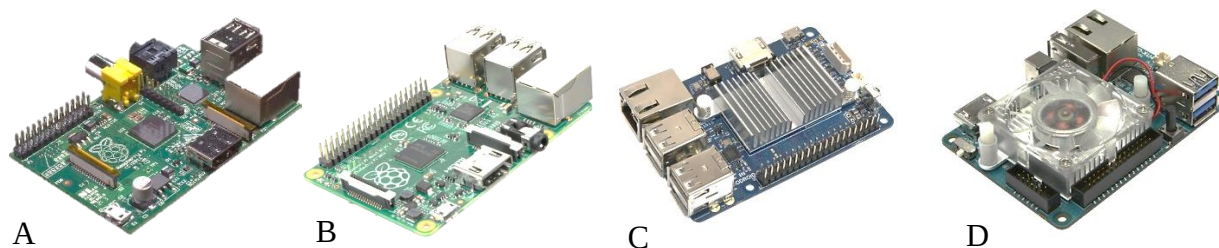
Funkciniai reikalavimai:

1. turi palaikyti „Linux“ operacinę sistemą

Nefunkciniai reikalavimai:

2. elektros sunaudojimas nedidesnis nei 7 W;
3. masė nedidesnė nei 40 g;
4. dydis ne didesnis nei 90x65x25 mm.

Įterptinei sistemai sukurti, buvo palyginti 4 skirtingai valdikliai: „Raspberry Pi Model B“, „Raspberry Pi 2 Model B“, „ODROID-C1+“, „ODROID-XU4“.



15 pav. Valdikliai: a) *Raspberry Pi Model B*; b) *Raspberry Pi 2 Model B*; c) *ODROID-C1+*; d) *ODROID-XU4*.

Paliginami valdikliai pavaizduoti 15 pav. (Valdiklių palyginimas buvo atliktas gegužės mėnesį 2015 metais). Pasirinkimas pagrįstas jau įsigytu „*Raspberry Pi 2 Model B*“ ir įgūdžiais, įgytais dirbant su juo. Be to, skraidantis robotas buvo paruoštas prie „*Raspberry Pi 2 Model B*“ valdiklio panaudojimo. Palyginami valdikliai yra panašūs pagal savo dydžius. Jų techniniai parametrai pateikti 1 lentelėje.

1 lentelė. Valdiklių parametų palyginimas [53 – 56]

Parametrai	Raspberry Pi Model B	Raspberry Pi 2 Model B	ODROID-C1+	ODROID-XU4
Kaina	40 €	40 €	40 €	120 €
Procesorius	ARMv6l BCM2835 700 MHz	900MHz quad-core ARM Cortex-A7 CPU from Broadcom	1.5Ghz quad core ARM Cortex-A5 CPU from Amlogic	Samsung Exynos5422 ARM Cortex-A15 Quad 2.0GHz / Cortex-A7 Quad 1.4GHz
Srovė	0,7 A	0.6-2 A	2 A	4 A
Įtampa	5 V	5 V	5 V	5 V
Galia	~3,5 W	~3-10 W	~10 W	~20 W
Atminties kiekis	512 MB	1 GB	1 GB	2 GB
USB jungtis	2 vnt.	4 vnt.	4 vnt.	3 vnt.
Operacinė sistema	Linux distribucija	Linux distribucija, Windows 10 IoT, Android	Linux distribucija, Android	Linux distribucija, Android
Kameros sąsaja	MIPI CSI 1080p (Raspberry Pi Camera), USB 720p	MIPI CSI 1080p (Raspberry Pi Camera), USB 720p	USB 720p	USB 720p
Sąsajos	GPIO, SPI, I2C	GPIO, SPI, I2C, JTAG	GPIO, SPI, I2C, RTC	GPIO, SPI, I2C, IRQ, RTC
Vaizdo išvestis	HDMI	HDMI	HDMI	HDMI
Valdiklio dydis	85x56x17 mm	85x56x17 mm	85x56x18 mm	82x58x22 mm
Mase	~40 g	~40 g	~40 g (55 g su radiatoriu)	~38 g (60 g su aušintuvu)

Įterptinė sistema, kuri bus integruota į skraidantį robotą turi būti kompaktiška, kiek įmanoma mažesnės masės, sunaudoti mažai elektros energijos. Įterptinei sistemai sukurti buvo parinktas „*Raspberry Pi 2 Model B*“ valdiklis, kadangi jo elektros energijos suvartojimas yra tarp mažiausių. Lyginant su kitais valdykliais, „*Raspberry Pi 2 Model B*“ turi sąsają kameros su maksimalia vaizdo rezoliucija 1080p. Į valdiklį įdegtą oficiali operacinė sistema „*Raspbian*“.

3.3 Vaizdo kameros parinkimas skraidančio roboto objekto sekimo įterptinei sistemai

Nuo kameros kokybės priklauso kompiuterio galimybė suvokti sceną, erdvę. Tikslesniam žymeklio atpažinimui vaizdas turi būti didesnės rezoliucijos, su mažesniu triukšmo kiekiu. Vaizdo nuskaitymo trukmė iš kameros matricos turi būti ko mažesnė, nes nuo jo priklauso vaizdo kokybė. Tai ypač aktualu, kai kamera nėra statinė, t.y. ji bus pritvirtinta prie skraidančio roboto. Skraidantis robotas skraidymo metu dėl išorinių veiksnių, tokių kaip vėjas, vibracija, negalės visą laiką būti vienoje pozicijoje. Dėl šių veiksnių, gautas vaizdas judėjimo metu bus netolygiai deformuotas ir pasislinkęs. Vaizdas turi būti kiek įmanoma geresnės kokybės, be to, greitas vaizdo nuskaitymas duoda didesnę kadrų skaičių per sekundę.

Kamerai parinkti keliami funkciniai ir nefunkciniai reikalavimai.

Funkciniai reikalavimai:

1. rezoliucija nemažesnė nei 1280x960 px;
2. kadrų kiekis per sekundę nemažesnis nei 10 kadrai;
3. rankinis fokusavimas, su automatinio fokusavimo išjungimo galimybe;
4. ekspozicijos nustatymai.

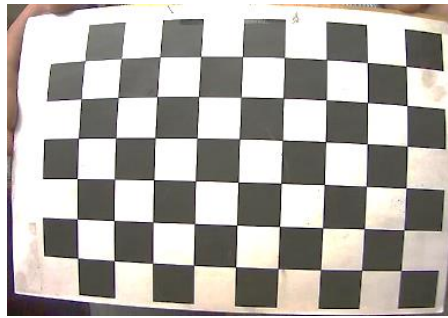
Nefunkciniai reikalavimai:

1. elektros energijos sunaudojimas nedidesnis nei 2 W;
2. masė nedidesnė nei 30 g;
3. nepriklausant nuo šviesos intensyvumo aplinkoje išgauti kuo geresnę vaizdų kokybę ir didesnę kadrų kiekį per sekundę;
4. kameros inicializavimas užtrunka neilgiau nei 3 s.
5. vaizdas turi būti ryškus, kiekvienas scenos objektas gerai matomas;

Buvo išbandytos trys kameros, kurios skiriasi pagal vaizdo rezoliuciją, kokybę ir prijungimo sąsają: „*Canyon Webcams – CNF-WCAM01B*“, „*Logitech® Webcam C905*“, „*Raspberry Pi Camera*“. 2 lentelėje pateiktas kamerų techninių parametrų, kurie turi reikšmingą įtaką vaizdo kokybei, palyginimas.

Kamera	Canyon Webcams CNF-WCAM01B	Logitech® Webcam C905	Raspberry Pi Camera
Megapikselių kiekis	1.3	2.0	5.0
Automatinis fokusavimas	nėra	yra	yra
Židinio nuotolis	3.85 mm	3.7 mm	3.6 mm
Objektyvo diafragma	f/6.0	f/2.0	f/2.9
Matrica	High Quality CMOS Sensor	Native 2-MP HD	OmniVision OV5647
Matricos rezoliucija	1280x1024 px	~1630x1227 px	2592x1944 px
Matricos dydis	8.30x6.65 mm	4.65x3.50 mm	3.76x2.74 mm
Pikselių dydis	6.4x6.4 μm	2.8x2.8 μm	1.4x1.4 μm
Optinis dydis	1/6"	1/45"	1/4"
Vaizdo gavimo trukmė	45-559 ms	7-69 ms	3-65 ms
Masė (be korpuso)	25 g	21 g	3 g
Dydis	58x38x35 mm	58x25x28 mm	25x24x9 mm
Galimybė nustatyti fokusavimą	rankinis	per programinę įrangą	per programinę įrangą
Fokusavimas	nuo 50 mm iki begalybės	nuo 50 mm iki begalybės	nuo 1 m iki begalybės
Ekspozicijos nustatymas	tik automatinis	yra, per programinę įrangą	yra, per programinę įrangą
Skaitmeninis triukšmo mažinimas	nėra	yra	yra
Palaikoma rezoliucija pikseliais su 30 kadrų per sekundę	iki 800x600 px	iki 960x720 px	iki 1920x1080 px
Palaikoma rezoliucija pikseliais su 25 kadrų per sekundę	-	1280x800 px	-
Palaikoma rezoliucija pikseliais su 15 kadrų per sekundę	-	1280x720 px	-
Palaikoma rezoliucija pikseliais su 10 kadrų per sekundę	1280x960 px	1280x1024 px 1600x900 px 1600x1200 px	-
Palaikoma rezoliucija pikseliais su 5 kadrų per sekundę	1600x1200 px	-	-
Prisijungimo sąsaja	USB	USB	MIPI CSI

Kameros buvo išbandytos su pasirinktu „Raspberry Pi 2 Model B“ valdikliu naudojant „OpenCV“ bibliotekoje realizuotus atpažinimo algoritmus. Naudojant ankstesnę patirtį pirmiausiai buvo išbandyta „Logitech Webcam C905“ kamera. Visų kameros funkcijų panaudojimui reikalingos papildomos palaikymo bibliotekos. Ankstesnė vaizdo atpažinimo sistema [66] buvo sukurta „Windows“ operacinei sistemai, papildomos bibliotekos buvo parsisiųstos iš oficialios svetainės ir sėkmingai įdegti. Tačiau šiai kamerai nėra oficialių bibliotekų „Raspbian“ operacinei sistemai. Derinant įvairius sprendimus pavyko gauti iš kameros vaizdus su pačia minimaliausia palaikoma rezoliucija 800x600 px. Nustatyti kameros papildomus parametrus programiškai taip pat nebuvo galimybės. Kita išbandyta kamera – „Canyon Webcams“. Po papildomų bibliotekų įdėgimo kamera iškart po prisijungimo veikdavo. Bandymo metu buvo pastebėtas kameros epipoliarinis iškraipymas, pavaizduotas 16 pav.



16 pav. Kameros „Canyon Webcams“ vaizdo epipoliarinis iškraipymas prieš korekciją

Šios kameros naudojimas reikalauja papildomo vaizdo apdorojimo – korekcijos. Su didesnio vaizdo rezoliucija mažėja vaizdo nuskaitymo trukmė. Lyginant su „Logitech Webcam C905“ kamera „Canyon Webcams“ kameros vaizdo kokybė žymai blogesnė: triukšmo faktorius svarbiausiose vaizdo vietose sudaro didelę spalvos įvairovę arba keičia objekto kontūro formą, tamsiu metu triukšmo kiekis ypač didesnis, o kameros vaizdo nuskaitymo trukmė padidėja 2-5 kartus. Įtaka daro ir matricos taškų kiekis – 2.0 Mp prieš 1.3 Mp.

Lyginant visus reikšmingus parametrus, buvo išrinkta „Raspberry Pi Camera“ kamera. Kameros privalumas – specialios sąsajos prijungimas prie valdiklio, o tai reiškia kad kamera oficialiai yra palaikoma. Vaizdo kameros nuskaitymo trukmė ypač minimali palyginant su kitomis išbandytomis kameromis su maksimalia rezoliucija 1920x1080 px, gaunamas vaizdas yra be epipoliarinio iškraipymo, todėl nereikia atlikti vaizdo korekcijos. Modulio masė sudaro tik tai 3 g. Vienintelis pastebėtas trūkumas – „OpenCV“ biblioteka nepalaiko šios kameros, tačiau buvo rastas atskiras sprendimas „RaspiCam“ [63], kurį panaudojant galima naudoti kamerą su „OpenCV“ biblioteka.

3.4 Papildomas apšvietimas skraidančio roboto objekto sekimo įterptinei sistemai

Kameros tamsoje negali suvokti aplinkos. Scenos vaizdas gautas tamsiu metu sudarytas iš juodų regionų ir matomų triukšmų. Eksperimentiškai buvo nustatytas teigiamas šviesos faktorius, kai scena buvo apšviesta – objektas buvo greičiau atpažintas ir sekamas, tiksliai nustatomas atstumas iki jo. Kai scenos apšvietimas buvo išjungtas, apskaičiuojamas atstumas buvo netikslus, o sekamas objektas pamestas dėl juodo triukšmo. Reikalingas papildomas apšvietimas, kuris atitiks funkcinius ir nefunkcinius reikalavimus.

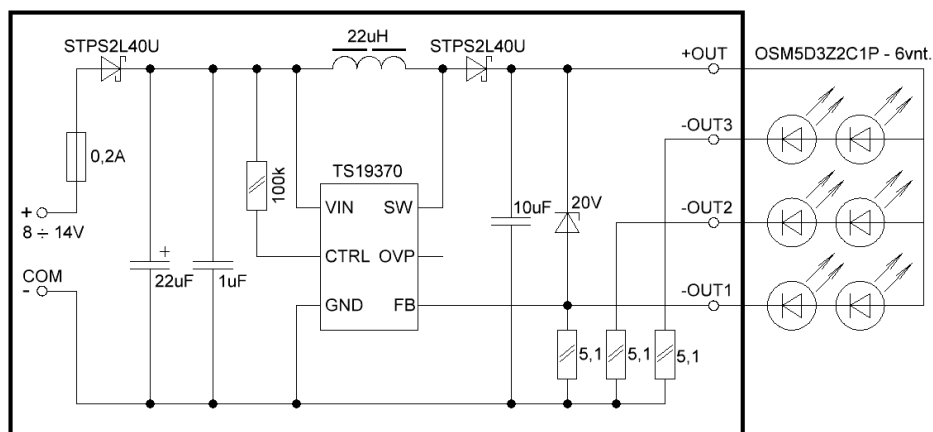
Funkciniai reikalavimai:

1. žymeklis tamsiu metu turi būti apšviestas.

Nefunkciniai reikalavimai:

1. papildomas apšvietimas turi būti pritvirtintas kuo arčiau kameros;
2. papildomas apšvietimas neturi užstoti ar kitaip trukdyti vaizdo scenai.

Pagal reikalavimus suprojektuotas šviesos diodų juostos impulsinis maitinimo šaltinis, prie kurios bus prijungti 6 baltos šviesos diodai (17 pav.).



17 pav. Šviesos diodų juostos impulsinio maitinimo šaltinio principinė schema

Buvo panaudoti „OSM5D3Z2C1P“ diodai dėl skleidžiamos baltos spalvos ir stipraus ryškumo šviesos, be to jiems nereikalingas papildomas aušinimas. Diodo parametrai pateikti 3 lentelėje.

3 lentelė. OSM5D3Z2C1P diodo techniniai parametrai

Diodas	OSM5D3Z2C1P
Šviesos intensyvumas	11000 mcd
Srovė	20 mA
Šviesos kampas	120°
Tiesioginis įtampos kritimas	9,3 V

Skraidančio roboto elektros akumuliatoriaus įtampa yra nuo 9V iki 12,6 V. Diodams maitinti reikia 20 mA srovės šaltinio, kurio išėjimo įtampa 18,6V, įtampos kritimas ant diodo turi būti 9,3 V. Tiesiogiai maitinti diodą per tiesinius stabilizatorius ar srovę apribojančius rezistorius negalima. Todėl buvo panaudotas įtampos pakėlimo mikroschema *TS19371* su srovės stabilizavimu apkrovos metu (angl. *Boost (Step-up) WLED Driver*). Mikroschemos pajungimo schema paimta iš specifikacijos [64], diodai po du sujungiami nuosekliai. Visi papildomi elementai išrinkti pagal skaičiavimus naudojant specifikaciją. Diodų srovė priklauso nuo balastinių rezistorių. Pagal pasirinktų balastinių rezistorių apskaičiuotą vertę nominalūs parametrai dydžiai yra 5,1 Ω – srovė bus apie 18,6 mA.

3.5 Techninės skraidančio roboto objekto sekimo įterptinės sistemos projektavimas

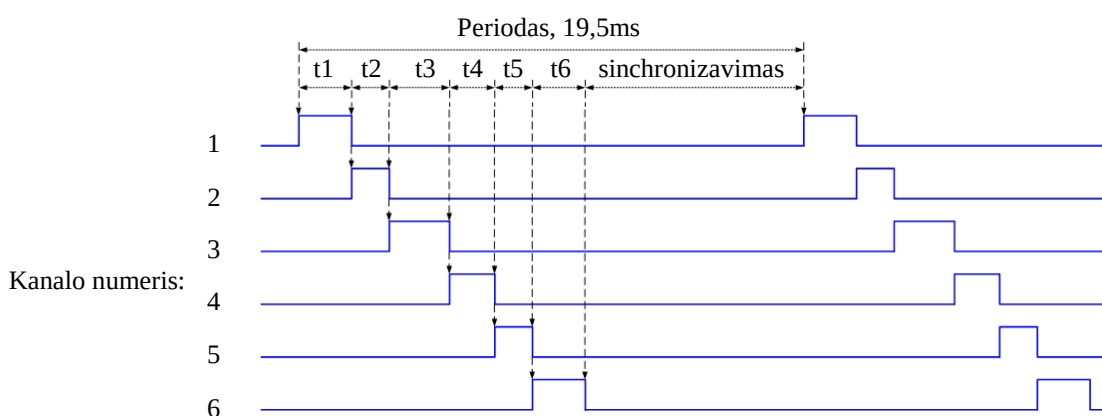
Kompiuterinio regėjimo sistema turi būti integruota į skraidantį robotą. Skraidančio roboto valdiklį „APM 2.6“ galima perkonfigūruoti ir perprogramuoti. Oficialios programinės įrangos

konfigūracijos galimybių pakanka, kad ją būtų galima naudoti kartų su įterptine sistema pagal poreikius. Valdiklis palaiko kelis skraidymo režimus, kuriuos galima perjungti per valdymo pultą. Įterptinė sistema pajungiama tarp skraidančio roboto valdiklio ir valdymo pulto imtuvo. Pirmiausia reikėjo išaiškinti kaip veikia skraidančio roboto komponentai kurie bus panaudojami įterptinėje sistemoje.

Originaliai imtuvas jungiamas tiesiogiai į skraidančio roboto valdiklį „APM 2.6“ per 6 kanalus:

- pirmi 4 – skraidančio roboto valdymas;
- 5 – pasirinktas skraidančio roboto skraidymo režimas;
- 6 – vaizdo kameros stabilizacijos sistemos valdymas.

Įterptinė sistema naudoja skraidančio roboto valdymo ir skraidymo režimo signalus. Pirmiausia reikėjo išaiškinti kokių pavidalų signalai perduodami į skraidančio roboto valdiklį. Naudojant oscilografą, buvo nustatyta signalo perdavimo seka, signalo periodas, kiekvieno atskiro kanalo signalo minimali ir maksimali trukmė. Signalų impulsas užtrunka nuo ~1ms iki ~2ms, periodas sudaro ~19,5ms. Signalų perdavimo principas parodytas 18 pav.



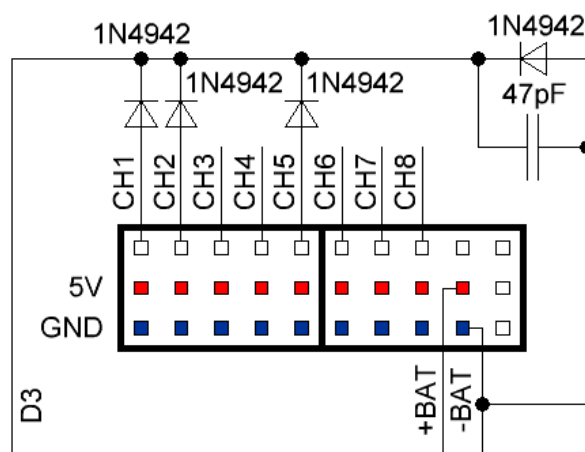
18 pav. Skraidančio roboto valdymo pulto imtuvo siunčiamų signalų perdavimo principas

Kiekvienas kanalo signalas perduodamas impulsu, keli impulsai vienu metu neperduodami, kai vieno impulso perdavimas baigėsi, pradedamas signalo perdavimas kitu kanalu. Kai visi imtuvo signalai perduoti, imtuvas laukia sekančio perdavimo pradžios, procesas kartojasi su 19,5ms periodu.

Atkartojant nustatytą siųstuvo ir imtuvo komunikavimo signalų seką su „Raspberry Pi 2 Model B“ skraidančio roboto valdiklis nuskaitydavo signalus, tačiau tik tada kai įterptinis valdiklis nebuvo apkrautas. Įterptinės sistemos valdiklio procesorius apkrovimo metu nebespėdavo su didesniu tikslumu nustatyti signalo impulso pločio. Priežastis – „Raspberry Pi 2 Model B“ valdiklis palaiko tik programinį įėjimo nuskaitymą, t.y., signalo būseną yra tikrinama programiškai, o ne su pertraukimiais, kurios fiksuoja signalo pasikeitimus. Galiausiai, po įvairių testavimo variantų, buvo nuspręsta komunikavimo signalų apdorojimui naudoti atskirą

mikrovaldiklį, kuris specialiai pritaikytas tokioms operacijoms, t.y., turi aparatinį signalo nuskaitymą – išorines pertrauktis. Buvo pasirinktas „AVR“ mikrovaldiklis, nes yra jų panaudojimo ir programavimo patirtis. Pagal įterptinės sistemos poreikius, konkrečiai buvo pasirinktas „ATMega32u4“ mikrovaldiklis. Programavimui ir komponentų prijungimui buvo naudota „Arduino Micro“ platforma su „ATMega32u4“ mikrovaldikliu.

Skraidančio roboto valdiklis palaiko du imtuvo signalų skaitymo režimus – per kelis kanalus arba per vieną, naudojant „PPM“ (impulso pozicijos moduliacija, angl. *Pulse Position Modulation*) signalą. Naudojant „PPM“ signalą pavyko sutaupyti mikrovaldiklio išėjimų skaičių, nes visų kanalų signalai perduodami per vieną išėjimą. Skraidančio roboto imtuvas nepalaiko „PPM“ signalo perdavimo režimo, todėl nuskaityti visus signalus per vieną įėjimą be papildomų sprendimų nepavyks. Kiekvieno kanalo nuskaitymas po kelių bandymų buvo nesėkmingas, nes nuskaitymo periodas sudarė apie 60 ms, o turėjo būti 19,5 ms. Kaip paaiškėjo, imtuvas perduoda signalus sumaišant kanalų išėjimus: pirmas signalas perduodamas per pirmą kanalą, antras – per trečią, trečias – per antrą ir t.t. Išsiaiškinus signalo perdavimo seką skirtingais kanalais, nuskaitymo periodo trukmė sudarė 19,5 ms, tačiau nuskaitytų signalo impulso trukmė buvo netiksli. Buvo sugalvotas sprendimas –prijungiant nelyginius imtuvo kanalus per vieną bendrą mikrovaldiklio įėjimą (19 pav). Nuskaityti signalus galima naudojant mikrovaldiklio išorines pertrauktis, apskaičiuojant nelyginių ir lyginių kanalų signalo impulso trukmes, su sąlyga, kad paskutinis kanalas bus nelyginis. Mūsų atveju yra 5 kanalai – tai sąlyga tenkinama.



19 pav. Skraidančio roboto imtuvo modifikavimo schema: nuoseklus nelyginių kanalų sujungimas

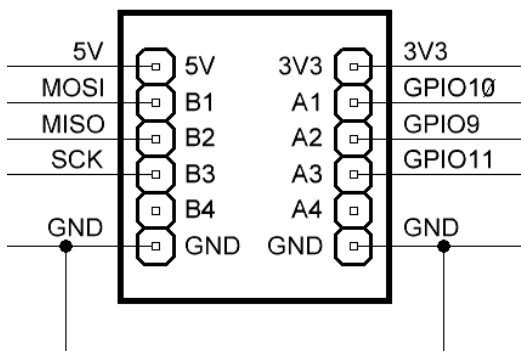
Atbulinės srovės apsaugai į nelyginius imtuvo išėjimus jungiami Šotkio diodai ir kondensatoriai. Mikrovaldiklis po šios imtuvo modifikacijos nuskaityti signalus per vieną įėjimą ir atkartoja imtuvo signalus su 0,5 μ s tikslumu.

Skraidančio roboto nusileidimo metu sistema turi turėti duomenis apie skridimo aukštį. Skraidančio roboto valdiklio aukščio jutiklis nebetinka, nes nustato aukštį iki jūros lygio. Skraidančio roboto valdiklis įjungimo metu nustato pradinį aukštį, kuris laikomas kaip nulinis.

Skraidančiam robotui leidžiantis ant platformos, kurios aukščio padėtis nežinoma, turi būti naudojamas ultragarsinis jutiklis. Buvo pasirinktas „*XL-MaxSonar - EZ Series*“. Ultragarsinis jutiklis, kurio tikslumo pakanka šiai užduočiai.

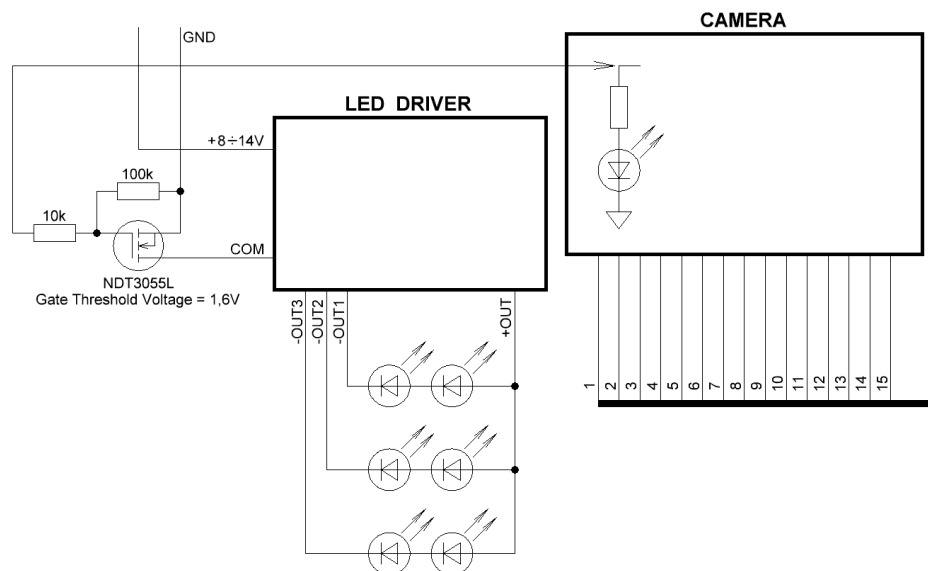
Bendravimas tarp „*Raspberry Pi 2 Model B*“ ir „*Arduino Micro*“ turi būti patikimas. Bendravimo metu perduodami imtuvo ir jutiklio duomenys. Bendravimas realizuotas naudojant „*SPI*“ protokolą. Įterptinės sistemos valdiklis pajungtas kaip pagrindinis, o papildomas mikrovaldiklis kaip valdomasis. Bendravimo metų perduodamas vienas baitas informacijos ir laukiama atsakymo. Per vieną bendravimą valdiklis perduoda duomenis apie skraidančio roboto valdymą, o tuo pačiu metu mikrovaldiklis jutiklio ir imtuvo duomenis.

Įterptinė sistema maitinimą nuo skraidančio roboto akumuliatoriaus (įtampos diapazonas 9V iki 12,6 V) per konverterį iš 12 V į 5 V. „*Raspberry Pi 2 Model B*“ valdiklis reikalauja 5V maitinimo, tačiau jo išėjimų įtampos lygis (maksimaliai 3,3V) nesutampa su „*Arduino Micro*“ išėjimų įtampos lygiu (maksimaliai 5V). Komunikacijai tarp „*Raspberry Pi 2 Model B*“ ir „*Arduino Micro*“ papildomai naudojamas modulis, su lygio konverteriu „*TXS0104ED*“ [65], pavaizduotas 20 pav.



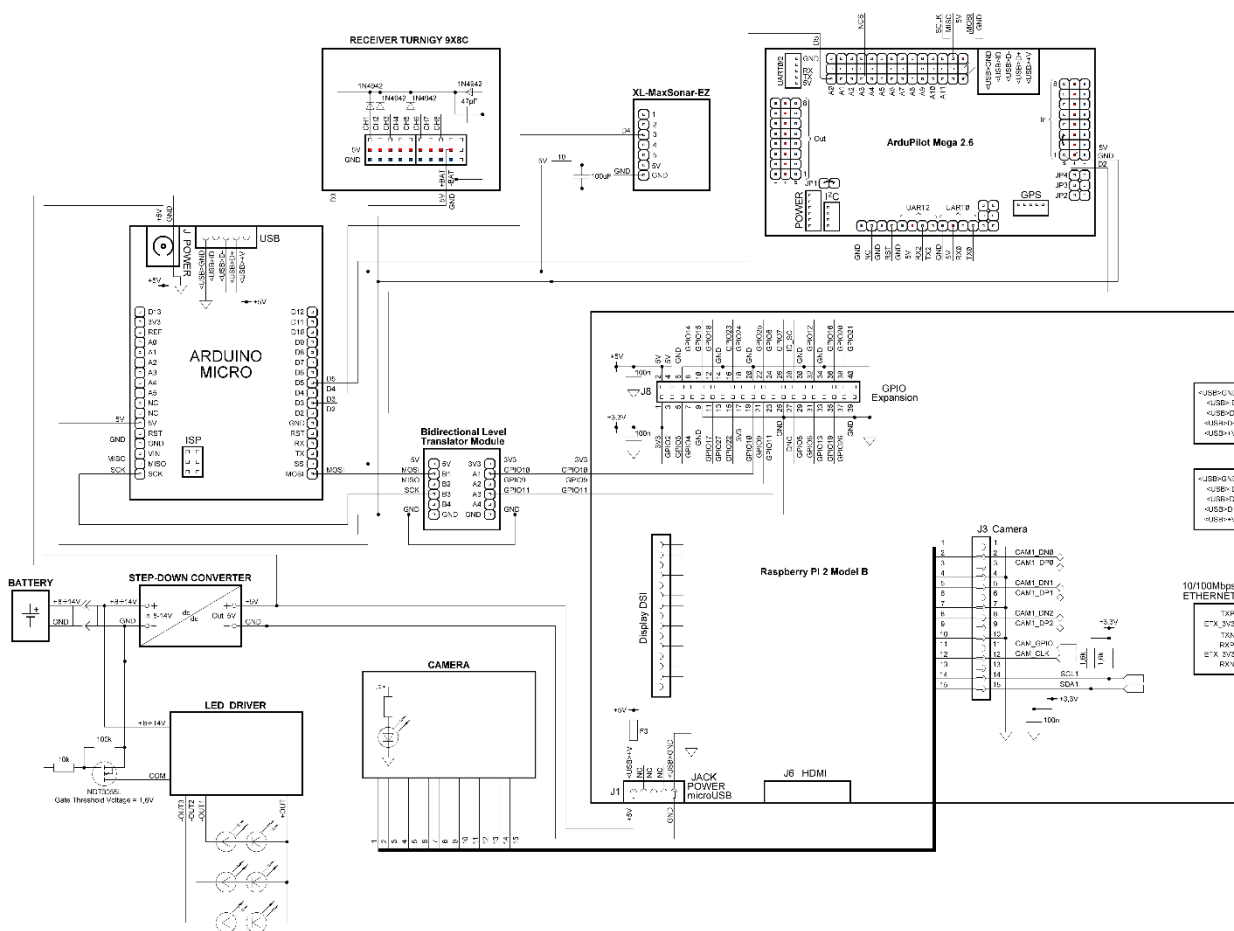
20 pav. Įtampos lygio konverteris modulis su „*TXS0104ED*“ tarp „*Raspberry Pi 2 Model B*“ ir „*Arduino Micro*“ bedravimui per „*SPI*“ protokolą

Parinktos vaizdo kameros modulis paprastai prijungimas prie „*Raspberry Pi 2 Model B*“ per „*MIPI CSI*“ sąsają. Kameros prijungimui buvo panaudotas 4 laidais platesnis plokščias kablelis: du laidai panaudoti ultragarsinio jutiklio prijungimui, trečias papildomo apšvietimo maitinimui, ketvirtas įžeminimui. Papildomas apšvietimas įjungiamas vaizdo kameros įjungimo metu. Panaudojant „*MOP*“ tranzistorių kaip raktą, atidaromas arba uždaromas papildomo apšvietimo maitinimo šaltinis. „*MOP*“ tranzistorius prijungtas prie vaizdo kameros modulio, prieš diodą ir rezistorių, kaip pavaizduota schemoje (21 pav).



21 pav. „MOP“ tranzistoriaus prijungimo prie vaizdo kameros modulio schema

Ultragarsinis jutiklis ir papildomas apšvietimas naudoja bendrą įžeminimą. Vaizdo kameros modulio įžeminimas su „Raspberry Pi 2 Model B“ nenaudojamas. Visos įterptinės sistemos schema pavaizduota 22 pav.



22 pav. Įterptinės sistemos ir naudojamų komponentų prisijungimo schema

Papildomai skraidančio roboto valdiklis prijungiamas prie „Arduino Micro“ (D5), šis kanalas skirtas įterptinės sistemos informavimui apie skraidančio roboto atskridimą į numatytą

GPS poziciją. Suprojektuota įterptinė sistema apjungia skraidančio roboto valdiklį, skraidančio roboto imtuvą, papildomą „Arduino Micro“ mikrovaldiklį ir „Raspberry Pi 2 Model B“ sistemos valdiklį.

3.6 Vaizdo kameros stabilizacijos sistemos parinkimas

Valdant skraidantį robotą jo horizontali padėtis nuolat keičiasi, todėl tiesiogiai pritvirtinus kamerą prie roboto korpuso sekamas žymeklis nuolat išeis iš kameros matymo zonos ribų. Kamera visada turi būti nukreipta į apačią, statmenai žymekliui. Šiai problemai spręsti naudojamas dviejų arba ar trijų ašių vaizdo kameros stabilizacijos sistema, dar žinoma kaip „gimbal“. Vaizdo kameros stabilizacijos sistemai buvo iškelti funkciniai ir nefunkciniai reikalavimai.

Funkciniai reikalavimai:

1. galimybė valdyti.

Nefunkciniai reikalavimai:

1. dviejų ašių stabilizacijos sistema;
2. elektros energijos vartojimas nedidesnis nei 7 W;
3. masė nedidesnė nei 300 g;
4. dydis ne didesnis nei 90x65x25 mm.

Pasirinkimas galėjo būti didesnis, tačiau buvo panaudota jau įsigyta stabilizacijos sistema „CNC FPV BGC 2 Axis Brushless Gimbal“ (masė 212 g), kuris atitinka iškeltus reikalavimus.

3.7 Konstrukcinių dalių projektavimas su SolidWorks

Vaizdo kameros modulis, ultragarsinis jutiklis ir papildomo apšvietimo diodai turi būti įrengti ant pasirinktos vaizdo kameros stabilizacijos sistemos. Originalus stabilizacijos sistemos kameros laikiklis nebetinka šiems komponentams, todėl reikalingas specialus laikiklis. Laikiklis turi būti pagamintas labai tiksliai, pagal visus reikiamus parametrus. Laikikliui pagaminti buvo pasirinkta 3D spausdinimo technologija. Detalė suprojektuota kompiuteryje su „SolidWorks 2015“ programine įranga. Prieš projektuojant laikiklį buvo iškelti funkciniai ir nefunkciniai reikalavimai.

Funkciniai reikalavimai:

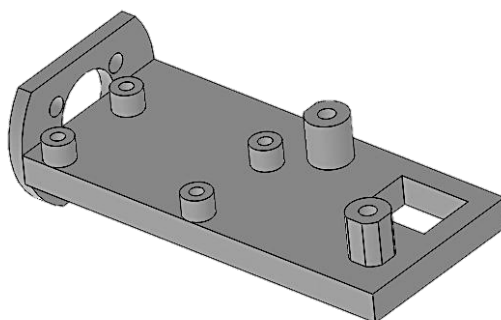
1. prie laikiklio bus galima pritvirtinti vaizdo kameros modulį;
2. prie laikiklio bus galima pritvirtinti ultragarsinį jutiklį;
3. prie laikiklio bus galima pritvirtinti papildomą apšvietimą.

Nefunkciniai reikalavimai:

1. laikiklis turi būti kiek įmanomą lengvesnis;

2. laikiklis turi būti stiprus ir išlaikyti smūgius;
3. laikiklis turi būti tikslus pagal dydžių parametrus;
4. papildomas apšvietimas neturi užstoti ar kitaip trukdyti vaizdo scenos daliai;
5. ultragarsinis jutiklis neturi užstoti ar kitaip trukdyti vaizdo scenos daliai;
6. duomenų plokščių kabelis turi daryti minimalią įtaką vaizdo kameros stabilizacijos sistemos veikimui.

Pagal šiuos reikalavimus su „SolidWorks 2015“ programiniu paketu buvo suprojektuotas laikiklis, pavaizduotas 23 pav. Laikiklis suprojektuotas taip, kad ultragarsinis jutiklis ir papildomo apšvietimo diodai neužstotų scenos vaizdo.



23 pav. Suprojektuotas laikiklis su „SolidWorks 2015“ programiniu paketu

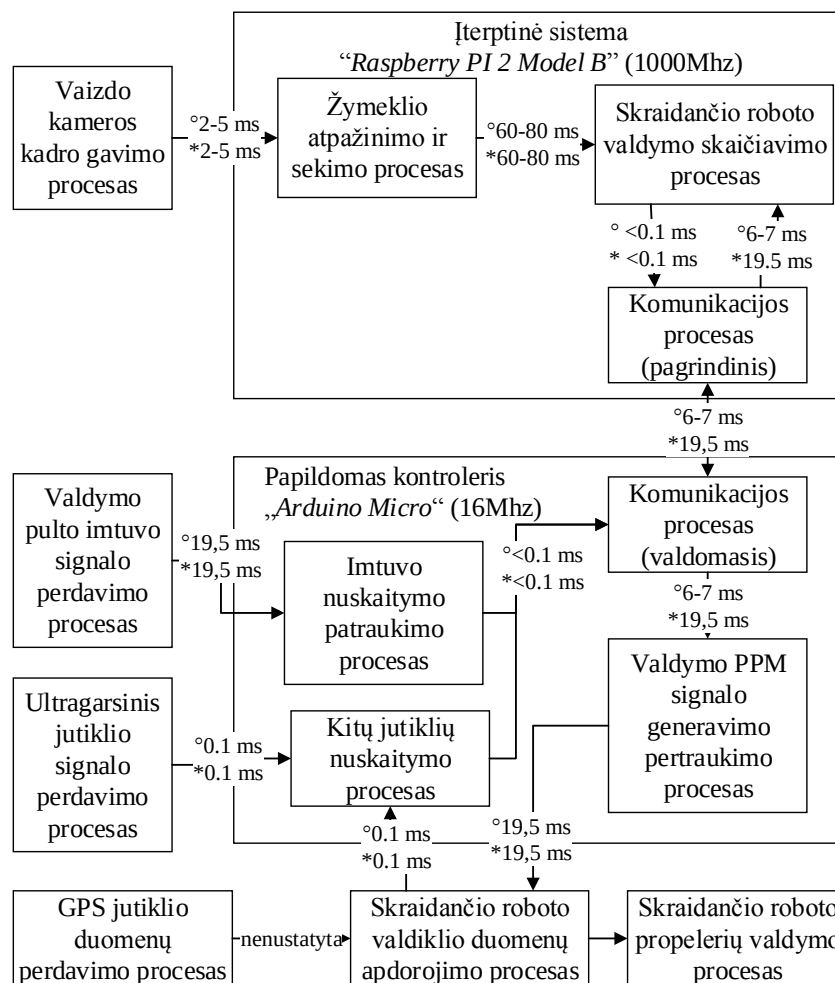
Pagrindas atspausdintas 3D spausdintuvu „Replicator“, iš „ABS“ plastiko, kurio masė – 7g., spausdinimo trukmė – 28 minučių. Projektuojant laikiklį, buvo numatytos vietos prie kurių bus tvirtinami komponentai ir pats laikiklis prie vaizdo kameros stabilizacijos sistemos.

3.8 Skraidančio roboto įterptinės objekto sekimo sistemos programinio prototipo kūrimas

Kiekvienas įterptinės sistemos modulis buvo programiškai testuojamas. Apjungiant komponentus į bendrą įterptinę sistemą, kelių komponentų duomenis buvo prarasti arba pakartotinai nuskaitomi, pavyzdžiui, įterptinės sistemos komunikacijos srautas testavimo metų nuskaitydavo imtuvo duomenis per „SPI“ protokolą kas 6-7 ms, o imtuvas atiduoda duomenis kas 19,5 ms. Kiekvieno modulio duomenų srautas turi savo veikimo dažnį, todėl įterptinės sistemos signalų sinchronizacija buvo atlikta „Arduino Micro“ mikrovaldiklyje.

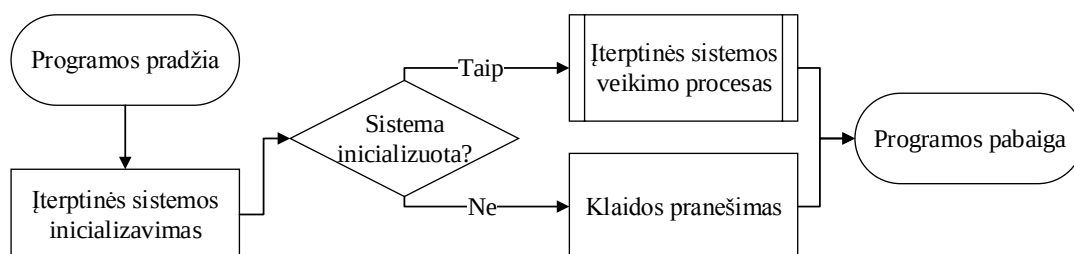
24 pav. pavaizduotas įterptinės sistemos modelis rodo atskiruose komponentuose vykdomų procesų trukmę. Be įterptinės sistemos skraidančio roboto valdiklis priimdavo pulto valdymo signalus kas 19,5 ms. Įterptinės sistemos sinchronizavimas su imtuvu buvo atliktas pagal imtuvo siunčiamų signalų periodą. Kompiuterinio regėjimo srauto vaizdo atpažinimo procesas užtrunka

60-80 ms, pagreitinėti iki 19,5 ms nėra galimybės. Skraidančio roboto valdymo srautas ištesiamas iki sekančio vaizdo atpažinimo, t.y. sinchronizuojamas su vaizdo atpažinimo procesu.



24 pav. Patobulintas įterptinės sistemos modelis, rodantis ryšius tarp sistemos komponentų ir duomenų perdavimo trukmes: ($\circ$) be sinchronizavimo; ($*$) su sinchronizavimu

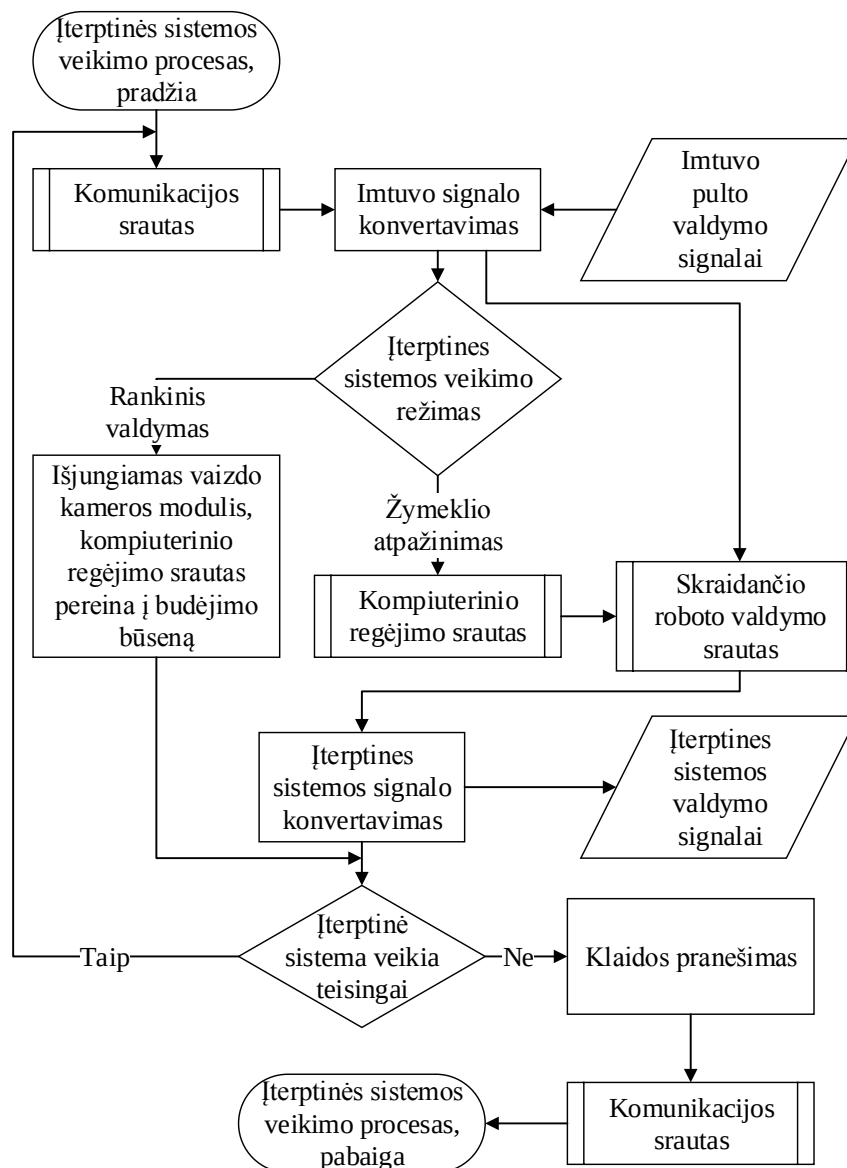
Toliau bus pateikti pagrindiniai įterptinės sistemos algoritmų blokinės schemos. Įjungiant įterptinę sistemą atliekamas sistemos inicializavimas. Įterptinės sistemos įjungimo algoritmas parodytas 25 pav.



25 pav. Skraidančio roboto įterptinės sistemos įjungimo algoritmas

Pirmasis žingsnis – tai įterptinės sistemos inicializavimas, kurio metu pasikrauna įterptinės sistemos operacinė sistema, sukuriama srautai, kurie vykdo atskirus sistemos algoritmus. Jei inicializacija buvo nesėkminga, sistema praneša apie klaidą. Jei inicializavimas buvo sėkmingas, įterptinė sistema pradeda veikimą. Įterptinė sistema, kaip parodyta 24 pav., sudaryta iš 3 srautų.

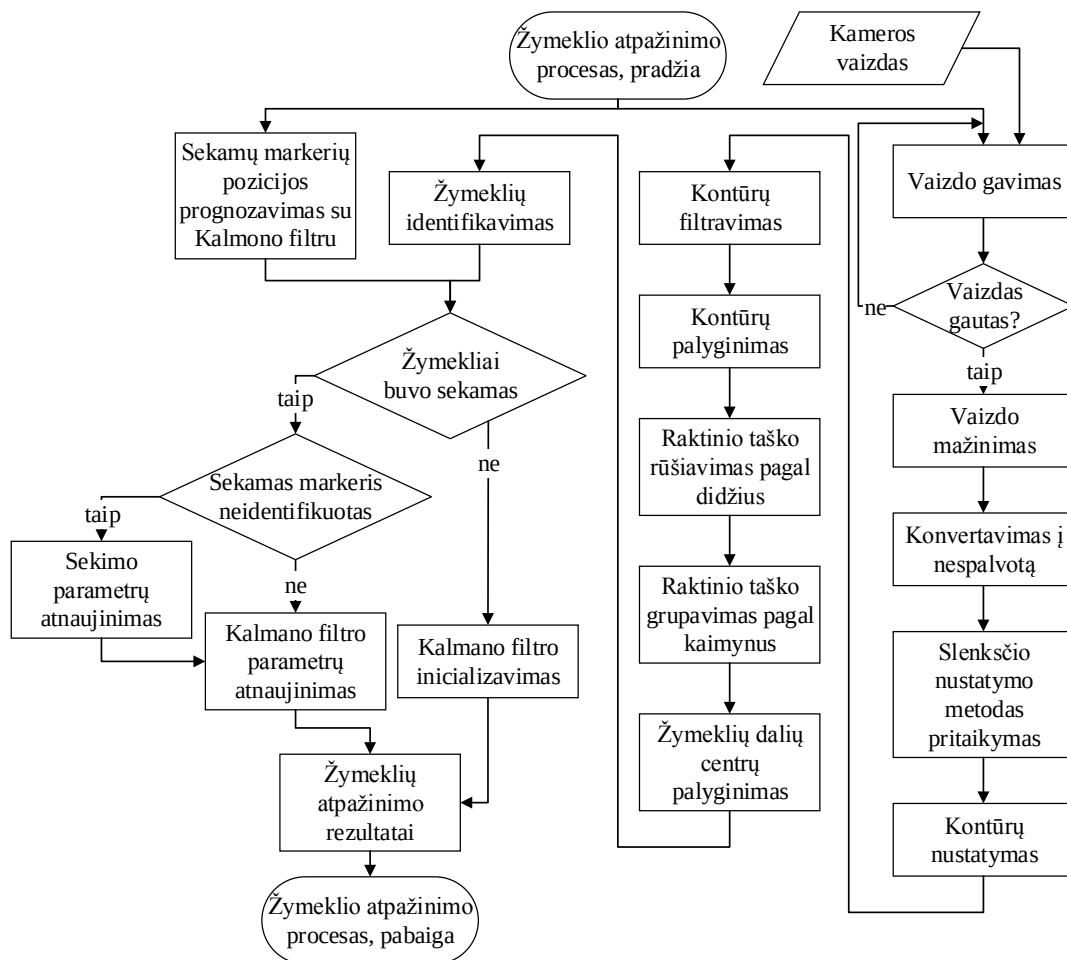
Nuo įterptinės sistemos veikimo režimo priklauso kompiuterinio regėjimo srauto veikimas, jeigu įjungtas rankinis valdymas, tai vaizdo kameros modulis išjungiamas ir srautas pereina į budėjimo būseną.



26 pav. Skraidančio roboto įterptinės sistemos branduolio algoritmas

Skraidančio roboto įterptinės sistemos branduolys, pavaizduotas 26 pav., pirmiausiai atlieka komunikavimą, nuskaitymo imtuvo signalus ir jutiklių duomenis. Imtuvo signalai konvertuojami į skaičius su kuriais programiškai patogiau dirbti. Priklausomai nuo pasirinkto sistemos režimo vyksta arba nevyksta kompiuterinio regėjimo srautas. Skraidančio roboto valdymo srautas, įvertinant jutiklio duomenis ir vaizdo atpažinimo rezultatą apskaičiuoja valdymo komandas. Valdymo komandos konvertuojamos atgal į skraidančio roboto valdiklio formatą. Jei valdymo komandos apskaičiuotos be klaidų, jos perduodamos į komunikacijos srautą. Su sekančiu signalų nuskaitymu bus perduotos valdymo komandas.

Įterptinės sistemos svarbiausią dalį sudaro kompiuterinio regėjimo žymeklio atpažinimas ir sekimas. Žymeklio atpažinimo ir sekimo algoritmas realizuotas keliais etapais (27 pav).



27 pav. Skraidančio roboto įterptinės sistemos kompiuterinio regėjimo žymeklio atpažinimo ir sekimo algoritmas

Pradinis etapas – vaizdo gavimo procesas. Vaizdo nuskaitymo rezoliucija 1280x960px, yra dvigubai didesnė, nei reikalinga tolesniam vaizdo apdorojimui. Vaizdo sumažinimas panaikina dalį triukšmo, vaizdas tampa aiškesnis, todėl atliekamas vaizdų sumažinimo procesas. Binarizacijos slenksčio nustatymo metodas reikalauja vieno kanalo vaizdo, todėl atliekamas konvertavimas į nespaltvotą. Nespaltvotam vaizdui pritaikomas patobulintas adaptyvus binarizacijos slenksčio nustatymo metodas. Gautas dvejetainis vaizdas naudojamas kontūrų atpažinimui.

Sekantis etapas – kontūrų rūšiavimas ir grupavimas. Maži kontūrai, kurių plotis mažesnis už nustatytą ribą (konstantą) iškart naikinamas. Naudojant formules (2) – (5) atliekamas kontūrų palyginimas. Gauti raktiniai taškai rūšiuojami pagal dydžius ((6) formulė). Surūšiuoti raktiniai taškai grupuojami pagal kaimynines pozicijas ((7) – (9) formulės). Sugrupuoti raktiniai taškai sudaro žymeklio dalį, jie palyginami pagal formulę (10). Atpažinti žymekliai identifikuojami, jei identifikavimas buvo nesėkmingas, žymeklis žymimas kaip neidentifikuotas. Visi žymekliai sekami. Priklausant nuo žymeklio sekimo būsenos, atliekamas Kalmano filtro inicializavimo

procesas arba Kalmano filtro parametrų atnaujinimo procesas. Skraidančiam robotui užduodamas nusileidimo platformos numeris, kuris įrašomas į įterptinę sistemą. Jei kompiuterinio regėjimo sistema aptiko daugiau nei vieną žymeklį, sekimo prioritetą skiriamas užduotam skraidančio roboto žymekliui.

3.9 Skraidančio roboto objekto sekimo įterptinės sistemos prototipo kūrimas ir jos integravimas į skraidantį robotą

Skraidančio roboto platformai naudojamas skraidantis robotas *Parallax ELEV-8* (28 pav.), kuris turi 4 propelerius, tačiau galima patobulinti iki 6 propelerių. Esant 4 propeleriams, jo masė sudaro 1,2 kg be akumulatoriaus, keliamoji galia yra apie 0,9 kg. Skraidantis robotas valdomas su *HoverFly Open* valdikliu. Valdiklis palaiko ne tik oficialią programinę įrangą. Su specialiai pritaikyta programavimo kalba galima parašyti savo valdymo algoritmą ir įkelti į valdiklį. Skraidantis robotas šio metų sukomplektuotas su 5Ah akumulatoriumi, kurio masė 350g. Skraidantis robotas gali skraidyti 20-30 minučių. Skraidantis robotas valdomas su skaitmeniniu valdymo pultu.

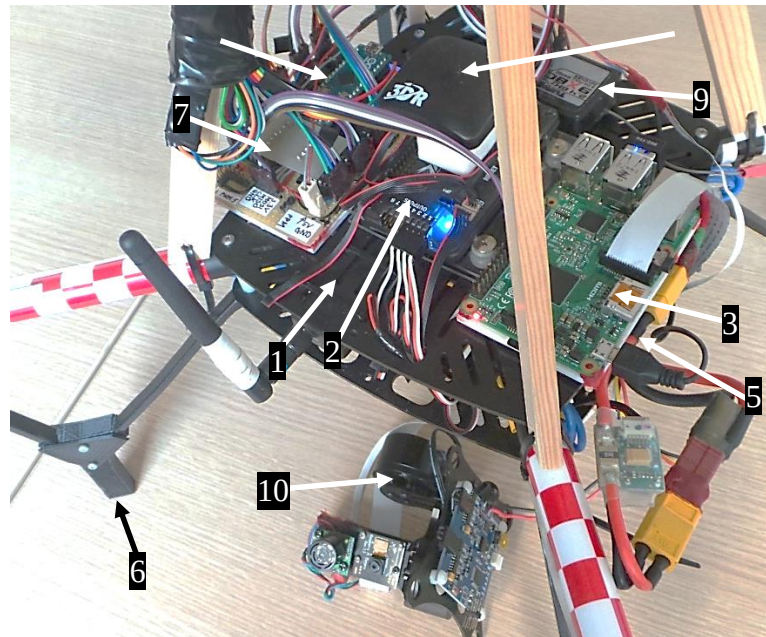


28 pav. Originalus skraidantis robotas *Parallax ELEV-8*

Siekiant sukurti skraidantį robotą, kurį valdo skraidančio roboto vaizdo atpažinimu grindžiama objekto sekimo įterptinė sistema, reikia modifikuoti originalų „*Parallax ELEV-8*“ skraidantį robotą. Originalus skraidančio roboto valdiklis turi mažai funkcijų: nėra GPS navigacijos, skraidantis robotas valdomas tik per 4 kanalus. Pirmiausiai buvo pakeistas valdiklis į „*APM 2.6*“. Valdiklis lengvai konfigūruojamas, palaiko kelis skraidymo režimus, kuris gali būti išrinktas vieno valdymo pulto mygtuko paspaudimu, taip pat yra telemetrija. Valdiklis buvo sukonfigūruotas ir paruoštas įterptinės sistemos integracijai pagal iškeltus reikalavimus.

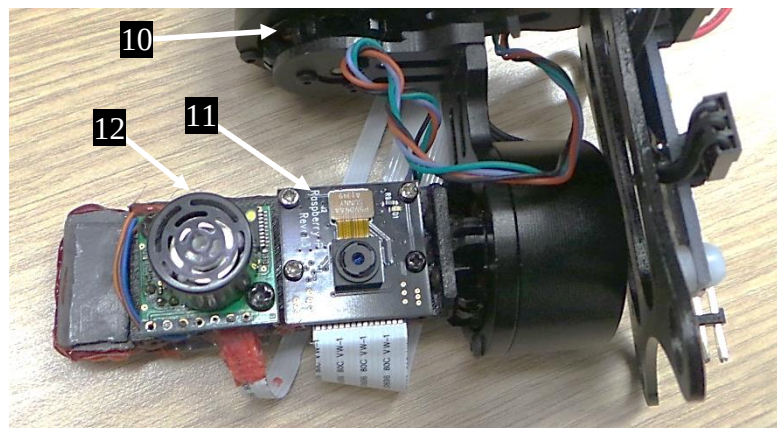
Skraidančio roboto masės centras buvo perneštas į apačią. Skraidančio roboto akumulatorius perneštas į apatinę platformos poziciją, o valdiklis perkeltas į akumulatoriaus vietą, varikliai buvo pakelti iki viršutinės platformos. Panaudojant 3D spausdintuvo spausdinimo technologiją buvo prailgintos skraidančio roboto nusileidimo kojos, taip pat pakeltas GPS imtuvo laikiklis. Įterptinė sistema buvo integruota į skraidantį robotą pagal 22 pav. pavaizduotą schemą,

gautas rezultatas - modifikuotas skraidantis robotas su pilnai integruota įterptine sistema pavaizduotas 29 pav.



29 pav. Modifikuotas skraidantis robotas su integruota įterptine sistema

Modifikuota vaizdo kameros stabilizacijos sistema pavaizduota 30 pav. Vaizdo kameros modulis, ultragarsinis jutiklis pritvirtinti prie atspausdinto laikiklio. Papildomi apšvietimo diodai nebuvo naudojami, nes įterptinės sistemos eksperimentai buvo atliekami dienos metu. Panaudoti komponentai pateikti 4 lentelėje



30 pav. Modifikuota vaizdo kameros stabilizacijos sistema su atspausdintu laikikliu ir pritvirtintais komponentais

Skraidančio roboto masė dabar sudaro 1423g. su akumuliatoriumi. Visi laidai buvo saugiai pritvirtinti prie skraidančio roboto korpuso. Panaudoti komponentai pateikti 4 lentelėje.

4 lentelė. Įterptinei sistemai apnaudotų komponentų sąrašas

Nr.	Komponento aprašymas	Nr.	Komponento aprašymas
1.	modifikuota skraidančio roboto platforma	7.	įtampos lygio konverteris
2.	APM 2.6 valdiklis	8.	3DR UBlox LEA-6S GPS modulis
3.	Raspberry PI 2 Model B valdiklis	9.	skaitmeninis 8-kanalinis imtuvas
4.	Arduino Micro mikrovaldiklis	10.	vaizdo kameros stabilizacijos sistema
5.	akumuliatorius	11.	Raspberry Pi Camera modulis
6.	prailgintos kojos	12.	XL-MaxSonar - EZ Series jutiklis

Skraidančio roboto platformą su integruota į skraidančio roboto vaizdo atpažinimu grindžiamą objekto sekimo įterptine sistema paruošta atlikti sekamo objekto žymeklio atpažinimo ir sekimo eksperimentinius tyrimus .

3.10 Skyriaus apibendrinimas

Binarizacijos slenksčio nustatymo metodo naudojimas su sekamo objekto QR žymekliu suteikia galimybę atpažinti ir sekti objekto žymeklį sudėtingoje aplinkoje, kai atstumas iki sekamo objekto žymeklio nedidesnis nei 15 m. Suprojektuota, sukurta ir verifikuotą pagal iškeltus funkcinius ir nefunkcinius reikalavimus vaizdo atpažinimu grindžiama objekto sekimo įterptinė sistema buvo integruota į skraidančio roboto platformą. Skraidančio roboto objekto sekimo sistemos procesų sinchronizavimas užtikrina duomenų perdavimo efektyvumą. Sinchronizavimas atliktas pagal lėčiausią įterptinės sistemos naudojamą komponentą – imtuvą, kurio vieno ciklo duomenų perdavimo trukmė – 19,5ms.

4 SKRAIDANČIO ROBOTO VAIZDO ATPAŽINIMU GRINDŽIAMOS OBJEKTO SEKIMO ĮTERPTINĖS SISTEMOS EKSPERIMENTINIAI TYRIMAI

Siekiant įvertinti sukurtos įterptinės sistemos parametrus bei patobulinto vaizdo binarizacijos algoritmo su adaptyviu slenksčio nustatymu efektyvumą bus atliekami eksperimentiniai tyrimai. Patobulintas vaizdo binarizacijos adaptyvaus slenksčio nustatymo metodas lyginimas su ankstesniuose tyrimuose pagerintu metodu, kuris naudoja vietinių blokų binarizacijos slenksčio nustatymą bei kitais metodais.

4.1 Eksperimento tikslai

Skraidančio roboto įterptinės vaizdo atpažinimu grindžiamos objekto sekimo sistemos prototipas buvo suprojektuotas ir sukurtas remiantis kitų kūrėjų panašiomis sistemomis ir patirtimi. Ankstesniuose tyrimuose, kurie, kartu su bendraautoriais, buvo paskelbti mokslinėje publikacijoje „*Resource Saving Approach Of Visual Tracking Fiducial Marker Recognition For Unmanned Aerial Vehicle*“ [32] buvo nustatytas vibracijos poveikis objekto atpažinimui. Mokslinio darbe buvo pasiūlytas pagerintas objektų atpažinimo metodas, naudojant vietinių blokų binarizacijos slenksčio nustatymą. Kai vibracija viršija 0,2 g sėkmingų atpažinimo rezultatų kiekis sudaro mažiau nei 20% nuo visų žymeklio atpažinimo bandymų. Mokslinio darbo tikslas buvo sumažinti sistemos elektros energijos sąnaudas bei padidinti sistemos efektyvumą patobulinant naudojamus metodus.

Šio darbo pagrindiniai eksperimento tikslai:

- palyginti patobulintą vaizdo binarizacijos adaptyvaus slenksčio nustatymo metodą su kitais binarizacijos slenksčio nustatymo metodais, įvertinti patobulinto metodo įtaką objekto atpažinimo ir sekimo procesui;
- Palyginti anksčiau atliktų tyrimų [32] rezultatus su šiame darbe patobulintų metodų rezultatais vertinant vaizdo binarizacijos adaptyvaus slenksčio nustatymo metodo atsparumą vibracijai.

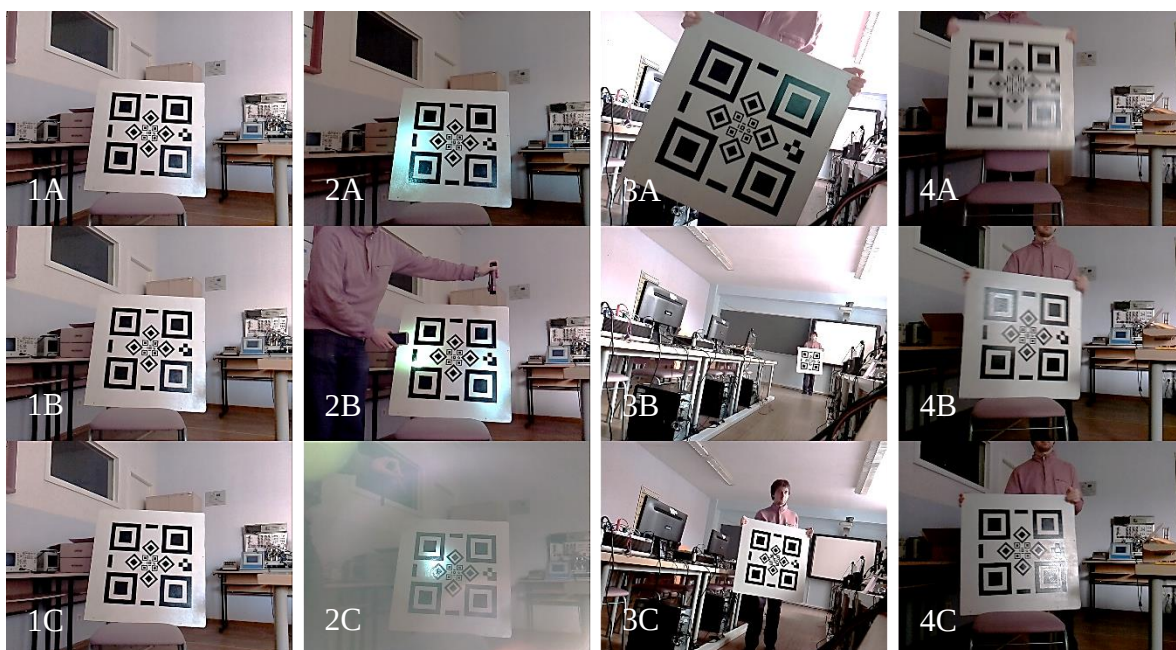
4.2 Eksperimento sąlygos

Pirmoji eksperimento dalis buvo atliktas laboratorijoje dienos metu. Robotas eksperimento metu buvo su išjungtais varikliais. Skraidančio roboto vaizdo atpažinimu grindžiama objekto sekimo įterptinė sistema buvo specialiai sukonfigūruota – įjungtas modifikuotas režimas su įjungta

kompiuterinio regėjimo sistema ir jo telemetrija. Visi vaizdo binarizacijos slenksčio nustatymo metodai naudoja vienodus vaizdus. Eksperimentas buvo padalintas į 4 etapus:

1. Žymeklio atpažinimas ir sekimas vyksta žymeklio ramybės būsenoje. Atstumas iki žymeklio 2 m. Žymeklis apšviestas natūralia dienos šviesa per kambario langus.
2. Žymeklio atpažinimas ir sekimas vyksta su dinaminiais papildomais apšvietimais. Atstumas iki žymeklio 2 m. Papildomų apšvietimų atstumas iki žymeklio nuo 0,1 iki 3 m.
3. Žymeklio atpažinimas ir sekimas vyksta keičiant žymeklio atstumą. Atstumas iki žymeklio nuo 0,5 m iki 10 m. Žymeklis apšviestas natūralia dienos šviesa per kambario langus.
4. Žymeklio atpažinimas ir sekimas vyksta žymeklį judinant staigiais judesiais. Atstumas iki žymeklio 1,9 – 2,1 m. Žymeklis apšviestas natūralia dienos šviesa per kambario langus.

Eksperimento etapai vyko apdorojant po 1500 kadrų. Duomenys buvo gauti naudojant skraidančio roboto objekto sekimo įterptinės sistemos telemetriją. 31 pav. pavaizduoti pirmojo eksperimento 3 atsitiktiniai kadrai (A, B, C) kiekvienam etapui



31 pav. Vaizdai gauti eksperimentu metu demonstruoja eksperimentu sąlygas

1 etape aplinkos sąlygų pakeitimas (keičiasi kambario apšvietumas dėl besikeičiančio debesuotumo) beveik nematomas. Kambario apšvietumo sąlygos keičiasi ir 2, 3 ir 4 etapuose. 2 etapo sąlygos buvo sukurtos naudojant rankinį žibintuvėlį „2015 Light FlashLight Plus“ ir mobiliojo telefono „Samsung Galaxy S3“ žibintuvėlį. 31 pav., 2C šviesos šaltinis buvo nukreiptas į kameros pusę. 3 etapo sąlygos buvo sukurtos artinant arba atitolinant žymeklį nuo kameros. Šiame etape dienos šviesa pasiskirsto netolygiai, 31 pav., 3A matome, jog žymeklis yra

neapšviestas tiesioginių dienos šviesos spindulių. 4 etapo sąlygos buvo sukurtos staigiai judinant žymeklį. Tam tikru kampu pasukus žymeklį, vaizdo kameros atžvilgiu, žymeklio paviršius atspindi dienos šviesą (31 pav., 4C).

Vidutinė vaizdo apdorojimo algoritmo trukmė ir absoliutinė paklaida su pasiklivimo lygmeniu 0,95 apskaičiuojama atitinkamai pagal (14) ir (15) formules [67]. Algoritmo apdorojimo vidutinė trukmė:

$$\bar{t} = \frac{\sum_{i=1}^n t_i}{n}, \quad (14)$$

kur n – kadrų skaičius, t_i – kadro apdorojimo trukmė, $\bar{t}$ – kadro apdorojimo vidutinė trukmė. Algoritmo apdorojimo vidutinės trukmės absoliutinė paklaida su pasiklivimo lygmeniu 0,95:

$$\Delta \bar{t} = t_{p,n-1} \sqrt{\frac{\sum_{i=1}^n (t_i - \bar{t})^2}{n(n-1)}} \quad (15)$$

kur $t_{p,n-1}$ – Stjudento koeficientas ($p = 0,95$), $\Delta \bar{t}$ – kadro apdorojimo trukmės absoliutinė paklaida.

Vaizdo binarizacijos slenksčio nustatymo metodai buvo vertinami pagal dvejetainio klasifikavimo metodo statistinius rodiklius [68]. Eksperimento metu gauti vaizdai buvo papildomai analizuojami ekspertu. Eksperimento rezultatai lyginami su ekspertų vertinimu, sudaroma „teigiama – neigiama“ matrica su rodikliais:

- TP (*true positive*) – teisingų žymeklio atpažinimų skaičius;
- FP (*false positive*) – klaidingų žymeklio atpažinimų skaičius;
- FN (*false negative*) – žymeklio neatpažinimų skaičius;
- TN (*true negative*) – teisingų žymeklio neatpažinimų skaičius.

Pagal „teigiama – neigiama“ matricą apskaičiuojami binarizacijos slenksčio nustatymo metodų rodikliai:

- ACC (angl. *accuracy*) – tikslumas, dydis nusakantis kaip tiksliai sistema atpažįsta žymeklį, kai žymeklis yra scenos kadre ir kaip tiksliai sistema neatpažįsta žymeklio, kai žymeklio nėra scenos kadre, skaičiuojamas:

$$ACC = \frac{\sum TP + \sum TN}{\sum TP + \sum TN + \sum FP + \sum FN} \quad (16)$$

- TPR (angl. *true positive rate*) – jautrumas, rodiklis, apibūdinantis metodo tikslumą ir rodantis tikimybę, kad žymeklio atpažinimas bus teigiamas, kai žymeklis yra scenos kadre, skaičiuojamas:

$$TPR = \frac{\sum TP}{\sum TP + \sum FN} \quad (17)$$

- TNR (angl. *true negative rate*) – specifiškumas, rodiklis, apibūdinantis metodo tikslumą ir rodantis tikimybę, kad žymeklio atpažinimas bus neigiamas, kai žymeklio nėra scenos kadre, skaičiuojamas:

$$TNR = \frac{\Sigma TN}{\Sigma TN + \Sigma FP} \quad (18)$$

- PPV (angl. *positive predictive value*) – teigiama žymeklio aptikimo prognozė parodo, kokia dalis tyrimo metu teigiamų žymeklio atpažinimų sudaro nuo visų žymeklio atpažinimų, skaičiuojamas:

$$PPV = \frac{\Sigma TP}{\Sigma TP + \Sigma FP} \quad (19)$$

- NPV (angl. *negative predictive value*) – neigiama žymeklio aptikimo prognozė parodo, kokia dalis tyrimo metu teigiamų žymeklio neatpažinimų sudaro nuo visų žymeklio neatpažinimų, skaičiuojamas:

$$NPV = \frac{\Sigma TN}{\Sigma TN + \Sigma FN} \quad (20)$$

Antrojoje eksperimento dalyje naudojama ankstesnio mokslinio darbo [32] roboto skraidymo metu filmuota medžiaga, kurioje buvo naudotas tas pats žymeklis kaip ir pirmojoje eksperimento dalyje. Atstumas iki žymeklio apie 10 m [32], eksperimento sąlygos (32 pav.) panašios į pirmojo eksperimento.



32 pav. Žymeklio filmavimo kadrai demonstruoja eksperimento sąlygas

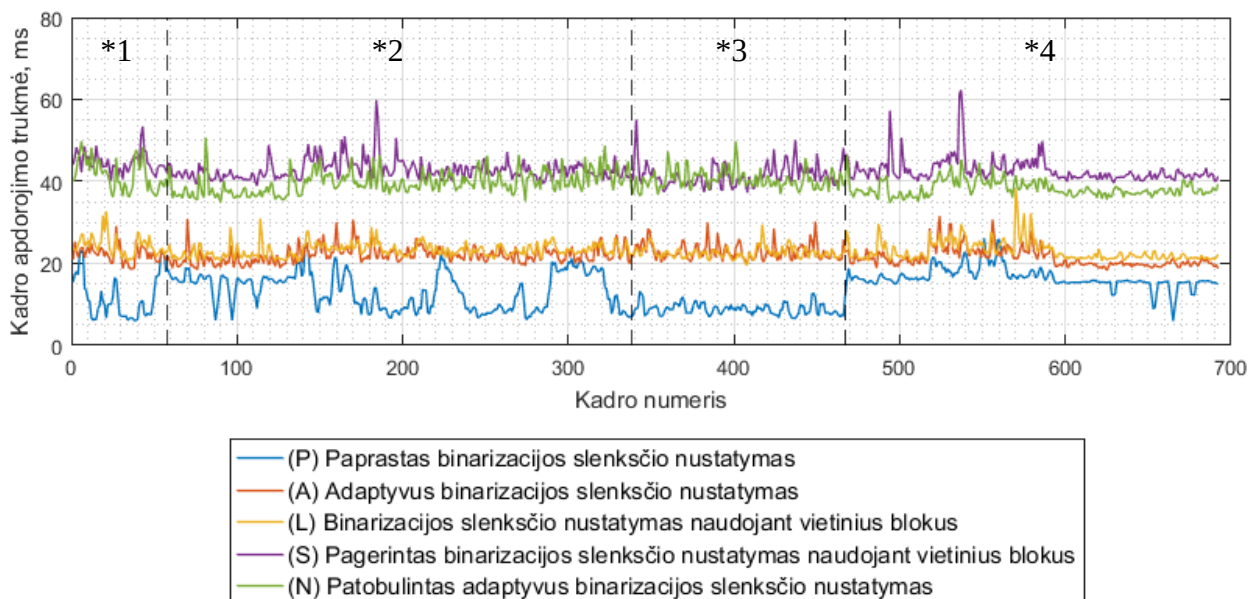
Mokslinio darbo eksperimento skaičiavimai buvo atlikti su „ASUS K55A“ nešiojamu kompiuteriu, kuris yra galingesnis už naudojama „Raspberry PI 2 Model B“ valdiklį, todėl žymeklio atpažinimo ir sekimo kadrų apdorojimo trukmė nebuvo palyginta – tiriamas tik algoritmo atsparumas vibracijai.

4.3 Vaizdo binarizacijos slenksčio nustatymo metodo eksperimentinis įvertinimas

Eksperimento metu buvo ištirti penki vaizdo binarizacijos slenksčio nustatymo metodai:

- paprastas binarizacijos slenksčio nustatymas (P);
- adaptyvus binarizacijos slenksčio nustatymas (A);
- binarizacijos slenksčio nustatymas naudojant vietinius blokus (L);
- pagerintas binarizacijos slenksčio nustatymas naudojant vietinius blokus (S);
- patobulintas adaptyvus binarizacijos slenksčio nustatymas (N).

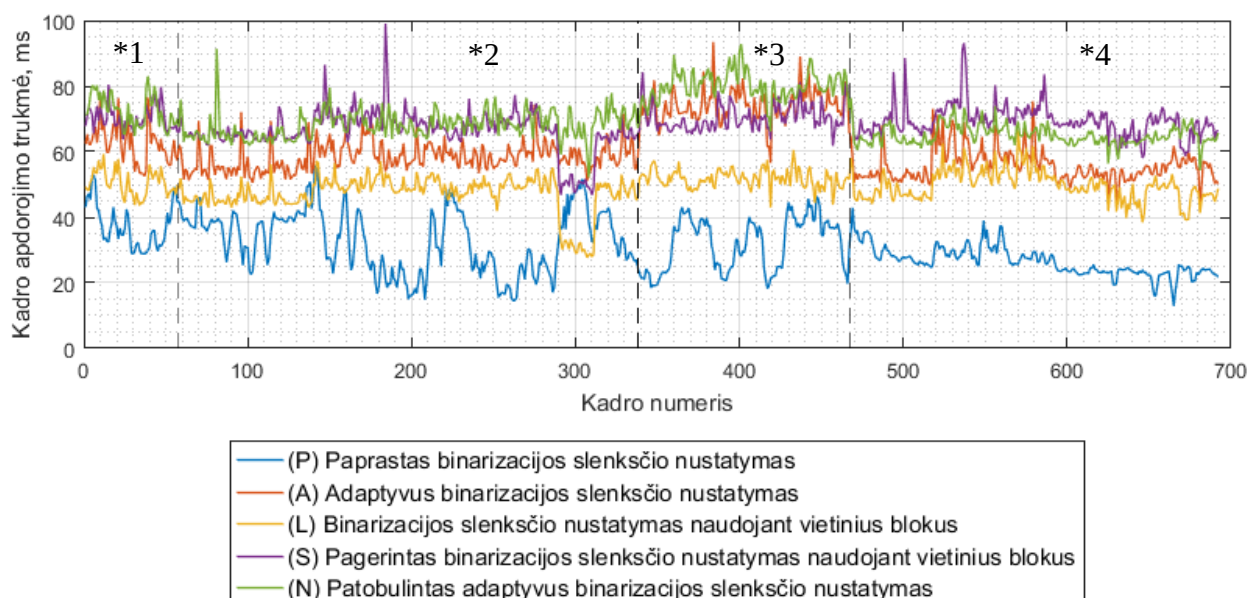
Skraidančio roboto objekto sekimo įterptinės sistemos žymeklio atpažinimo ir sekimo programos telemetrijos duomenys pavaizduoti 33-39 pav. Žvaigždute * pažymėti pirmojo eksperimento atskiri etapai atskirti vertikaliomis punktyrinėmis linijomis.



33 pav. Skraidančio roboto objekto sekimo įterptinės sistemos telemetrija: vaizdo binarizacijos algoritmo apdorojimo trukmė naudojant skirtingus slenksčio nustatymo metodus

Grafike (33 pav) pavaizduoti, pagal telemetrijos duomenis, binarizacijos slenksčio nustatymo metodų apdorojimo trukmės kiekvienam kadrai. Matome, jog paprastas binarizacijos slenksčio nustatymo metodas labiau priklauso nuo aplinkos sudėtingumo, kiti metodai mažiau. Algoritmų apdorojimo trukmė nepriklauso nuo išplėstos telemetrijos, duomenų ir kadro išsaugojimų į vidinę „Raspberry PI 2 Model B“ valdiklio atmintį.

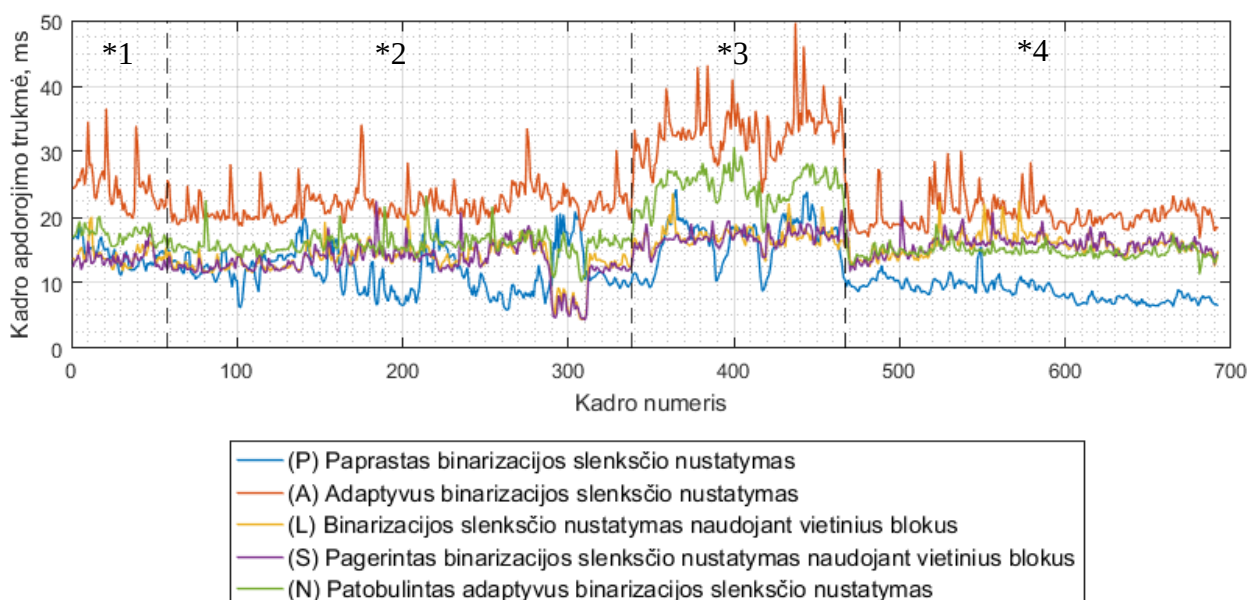
Lyginant binarizacijos slenksčio nustatymo metodų apdorojimo vidutinę trukmę (5 lentelė) patobulintas adaptyvus binarizacijos slenksčio nustatymas yra greitesnis negu pagerintas metodas, naudojantis vietinių blokų binarizacijos slenksčio nustatymo rezultatus.



34 pav. Skraidančio roboto objekto sekimo įterptinės sistemos telemetrija: žymeklio atpažinimo ir sekimo algoritmų apdorojimo trukmė naudojant skirtingus binarizacijos slenksčio nustatymo metodus

Grafike (34 pav.) pavaizduotos žymeklio atpažinimo ir sekimo, naudojant skirtingus binarizacijos slenksčio nustatymo metodus, algoritmų apdorojimo trukmės.

Lyginant žymeklio atpažinimo ir sekimo algoritmų vidutines apdorojimo trukmes (5 lentelė) matome, jog patobulintas adaptyvaus binarizacijos slenksčio nustatymas, lyginant su kitais, yra pats lėčiausias. Naudojant adaptyvų binarizacijos slenksčio nustatymą, vidutinė apdorojimo trukmė, lyginant su paprastu slenksčio nustatymu, padidėjo 3 kartus.

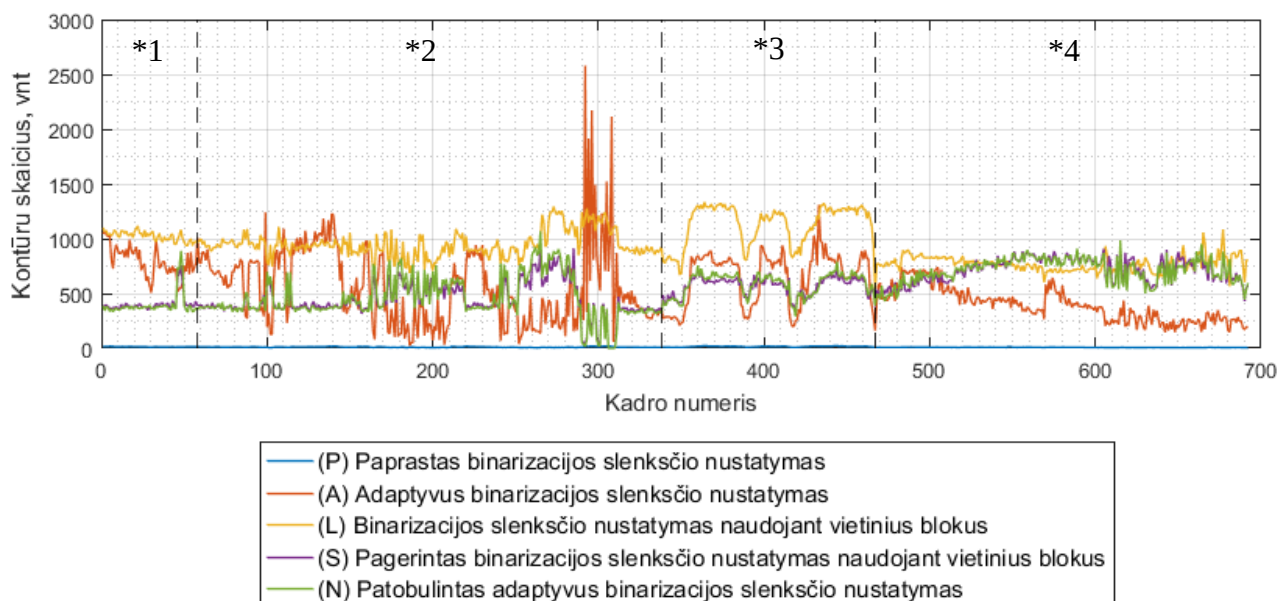


35 pav. Skraidančio roboto objekto sekimo įterptinės sistemos telemetrija: kontūrų aptikimo algoritmo apdorojimo trukmė, naudojant skirtingus binarizacijos slenksčio nustatymo metodus

Analizuojant surinktus duomenis buvo nustatyta staigus sistemos produktyvumo kritimo priežastis – kontūrų aptikimo metodas. Grafike (35 pav.) pavaizduoti, kontūrų aptikimo algoritmo apdorojimo trukmės kiekvienam kadrai. 3 etape matosi kontūrų aptikimo algoritmo apdorojimo

trukmės padidėjimas, kuris atsiranda dėl besikeičiančio atstumo nuo kameros iki žymeklio – atstumui iki žymeklio didėjant nuo 0,5 m iki 10 m kontūrų aptikimo algoritmo apdorojimo trukmė taip pat didėjo.

Vertinant kontūrų aptikimo algoritmo apdorojimo trukmę (5 lentelė), naudojant skirtingus binarizacijos slenksčio nustatymo metodus, buvo pastebėta šių metodų tarpusavio priklausomybė. Jei kadras po binarizacijos slenksčio nustatymo būna labai „triukšmingas“, tai kontūrų aptikimo algoritmo apdorojimo trukmė taip pat padidėja. Toks efektas buvo aiškiai matomas naudojant adaptyvų binarizacijos slenksčio nustatymo metodą.



36 pav. Skraidančio roboto objekto sekimo įterptinės sistemos telemetrija: aptiktų kontūrų kiekis naudojant skirtingus binarizacijos slenksčio nustatymo metodus

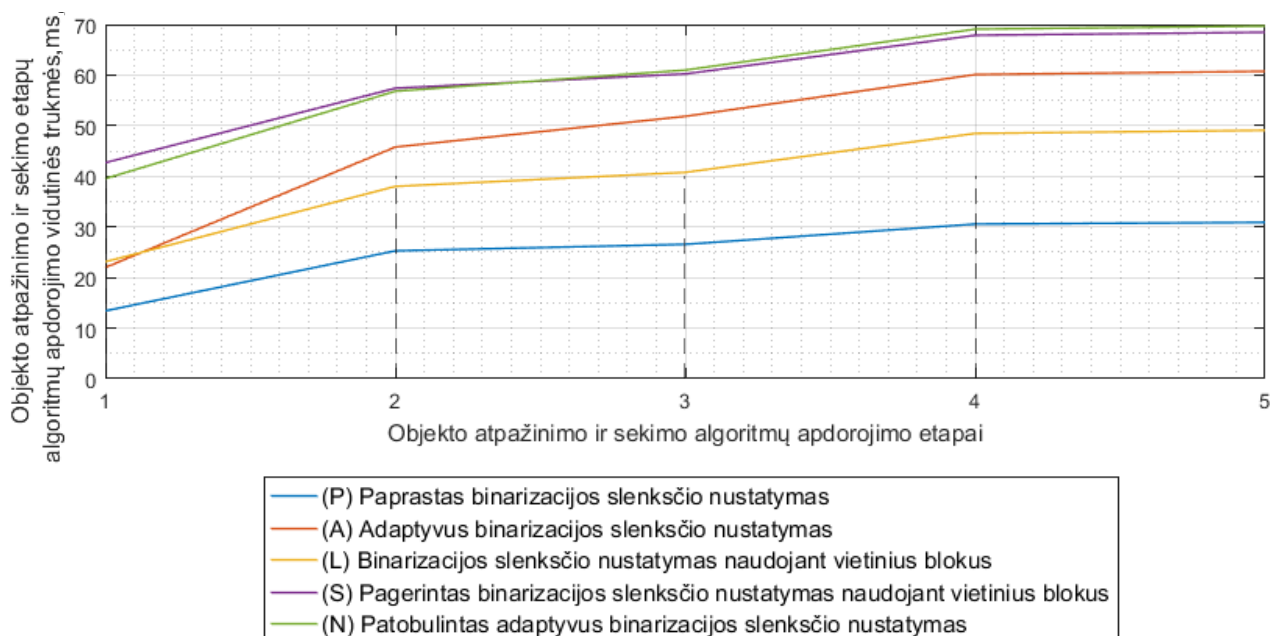
Grafike (36 pav.) pavaizduoti kontūrų aptikimo apdorojimo rezultatai – aptiktų kontūrų kiekis. Apdorojant kadrus nuo 280 iki 320, aptiktų kontūrų kiekis, naudojant adaptyvų binarizacijos slenksčio nustatymo metodą, viršija 1450 vienetų, nes dvejetainis vaizdas gaunasi labai triukšmingas dėl originalaus vaizdo mažo kontrasto. Naudojant kitus binarizacijos slenksčio nustatymo metodus aptiktų kontūrų kiekis neviršijo 1450 vienetų, nes dvejetainis vaizdas gaunasi mažiau triukšmingas.

5 lentelėje pateiktos algoritmų apdorojimų vidutinės trukmės ir jų absoliutinės paklaidos (apskaičiuotos pagal (14) ir (15) formules) naudojant skirtingus binarizacijos slenksčio nustatymo metodus.

5 lentelė. Algoritmų apdorojimo trukmės tyrimo rezultatai naudojant skirtingus binarizacijos slenksčio nustatymo metodus

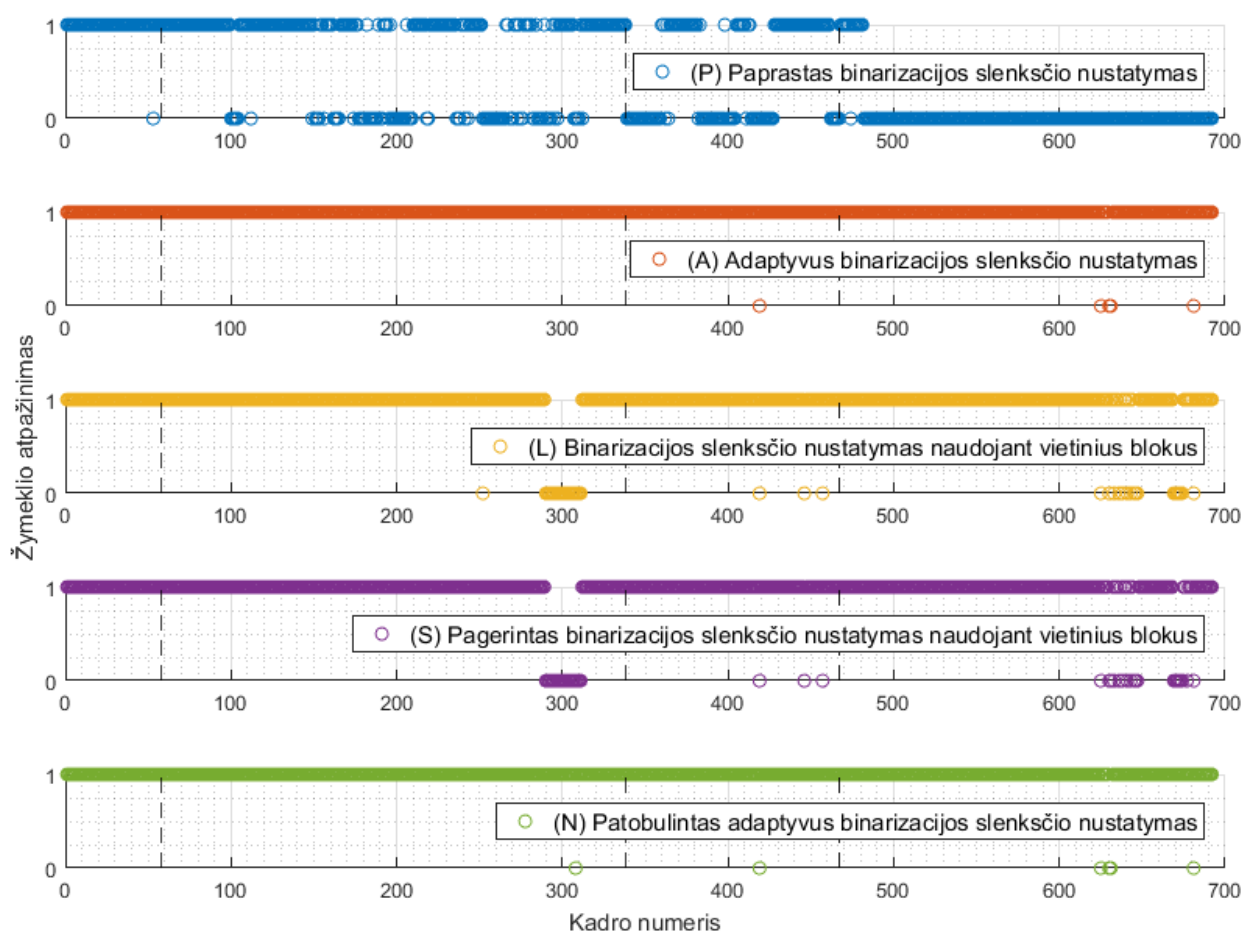
Binarizacijos slenksčio nustatymo metodas	Objekto atpažinimo ir sekimo etapų algoritmų apdorojimo vidutinės trukmės (ms)					Algoritmų apdorojimo vidutinių trukmių suma (ms)
	Slenksčio nustatymo algoritmas	Kontūru aptikimo algoritmas	Žymeklio atpažinimo algoritmas	Žymeklio identifikavimo algoritmas	Žymeklio sekimo algoritmas	
(P)	13,436 ± 0,324	11,859 ± 0,285	1,288 ± 0,051	3,999 ± 0,267	0,343 ± 0,022	30,924 ± 0,603
(A)	22,009 ± 0,169	23,792 ± 0,358	6,068 ± 0,118	8,255 ± 0,069	0,644 ± 0,010	60,009 ± 0,557
(L)	23,139 ± 0,157	14,889 ± 0,169	2,767 ± 0,049	7,695 ± 0,130	0,618 ± 0,010	49,109 ± 0,351
(S)	42,716 ± 0,213	14,711 ± 0,172	2,824 ± 0,050	7,632 ± 0,137	0,596 ± 0,010	68,479 ± 0,399
(N)	39,560 ± 0,206	17,269 ± 0,253	4,180 ± 0,083	8,087 ± 0,062	0,616 ± 0,013	69,712 ± 0,451

Žymeklio atpažinimo algoritmo apdorojimo trukmė taip pat priklauso nuo taikomo binarizacijos slenksčio nustatymo metodo ir aptiktų kontūrų kokybės. Vertinant kiekviename etape apskaičiuotas algoritmų apdorojimo vidutinės vertės galima teigti, kad daugiausiai laiko užtrunka vaizdo binarizacijos slenksčio nustatymas ir šiek tiek mažiau kontūrų aptikimas. 37 pav. pateiktas demonstruojamas skraidančio roboto objekto sekimo įterptinės sistemos algoritmų apdorojimo vidutinės trukmės suma atskiruose etapuose.



37 pav. Skraidančio roboto objekto sekimo įterptinės sistemos algoritmų apdorojimo vidutinės trukmės suma atskiruose etapuose: 1) binarizacijos slenksčio nustatymo algoritmas; 2) kontūrų aptikimo algoritmas; 3) žymeklio atpažinimo algoritmas; 4) žymeklio identifikavimo algoritmas; 5) žymeklio sekimo algoritmas

Žymeklio atpažinimo rezultatai, naudojant skirtingus binarizacijos slenksčio nustatymo metodus pavaizduoti 38 pav.



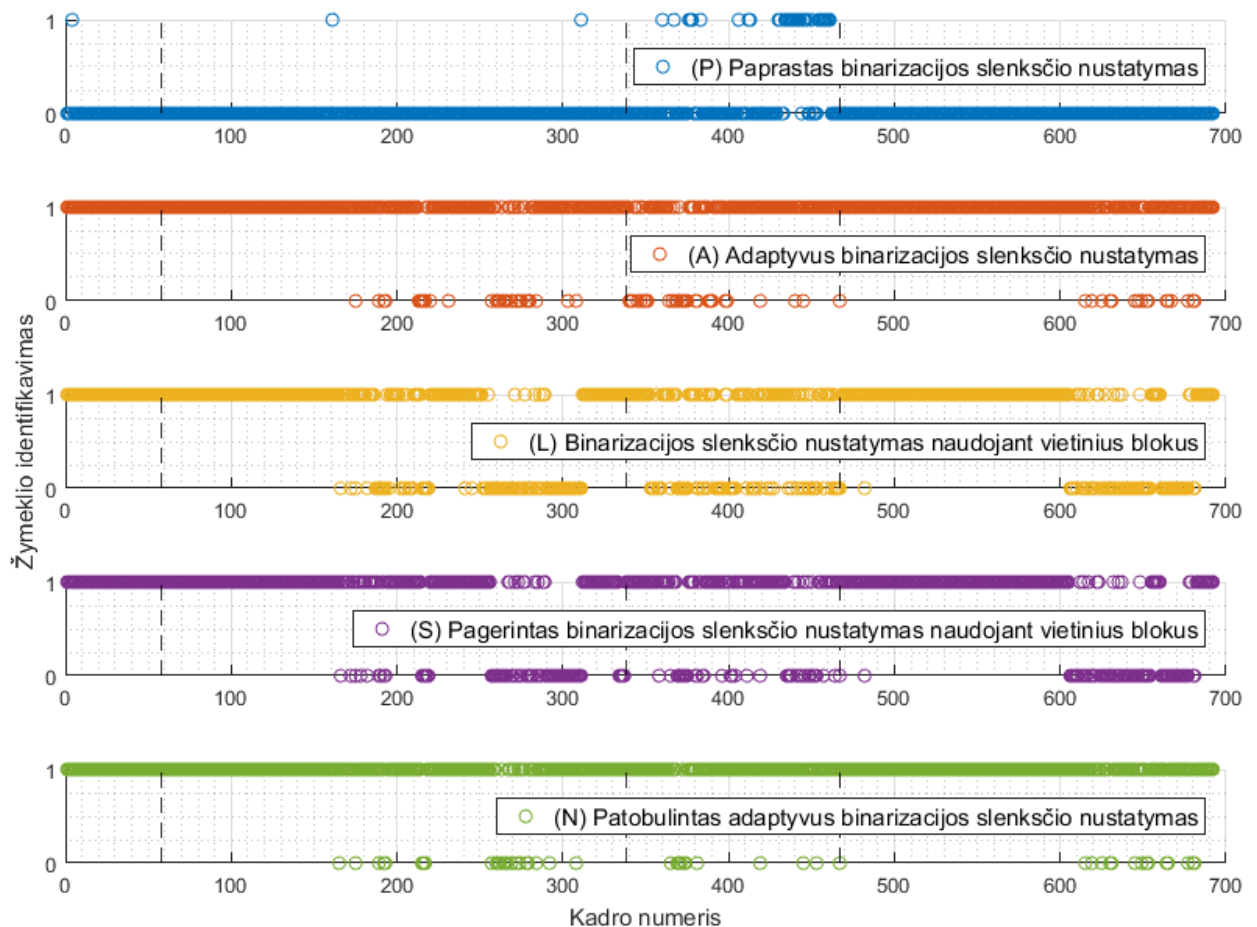
38 pav. Skraidančio roboto objekto sekimo įterptinės sistemos telemetrija: žymeklio atpažinimo rezultatas

Pagal skraidančio roboto objekto sekimo įterptinės sistemos telemetriją matosi, jog žymeklio atpažinimas yra blogiausias (daugiausiai kadrų kuriuose nebuvo atpažintas žymeklis) naudojant paprastą vaizdo binarizacijos slenksčio nustatymą. Paprasto binarizacijos slenksčio nustatymo metodas ypač neefektyvus, kai kinta žymeklio apšvietimas. Naudojant binarizacijos slenksčio nustatymą su vietiniais blokais ir pagerintą vietinių blokų binarizacijos slenksčio nustatymą žymeklio atpažinimo rezultatas taip pat neigiamas, esant sudėtingoms aplinkos apšvietimo sąlygoms (31 pav. 2C) ir staigiai judinant žymeklį.

6 lentelė. Žymeklio atpažinimo rodikliai naudojant skirtingus binarizacijos slenksčio nustatymo metodus

Binarizacijos slenksčio nustatymo metodas	TPR (%)	TNR (%)	PPV (%)	NPV (%)	ACC (%)
(P)	44,9	97,6	99,2	20,9	51,7
(A)	97,9	99,2	99,9	87,3	98,0
(L)	92,6	98,4	99,7	66,5	93,4
(S)	92,4	99,2	99,9	66,0	93,3
(N)	97,9	99,2	99,9	87,3	98,0

Naudojant dvejetainio klasifikavimo metodus, pagal (16) – (20) formules, buvo apskaičiuoti žymeklio atpažinimo statistiniai rodikliai naudojant skirtingus binarizacijos slenksčio nustatymo metodus. Lyginant statistinius rodiklius, kurie pateikti 6 lentelėje, aukščiausi tikslumo rodikliai (98%) pasiekiami naudojant patobulintą binarizacijos adaptyvaus slenksčio nustatymą ir adaptyvaus binarizacijos slenksčio nustatymą. Kiekvienas metodas tam tikromis sąlygomis turi savo pranašumų ir trūkumų. Žymeklio atpažinimo tikslumo rodiklis naudojant paprastą binarizacijos slenksčio nustatymo metodą pats mažiausias – 51,7%. Naudojant kitus binarizacijos slenksčio nustatymo metodus tikslumo rodiklis sudaro apie 93%. Neigiama žymeklio aptikimo prognozinė vertė (NPV) charakterizuoja, kokia dalis skraidančio roboto objekto sekimo įterptinės sistemos atpažinimų sudaro teisingų žymeklio neatpažinimų nuo visų žymeklio neatpažinimų. Naudojant paprastą binarizacijos slenksčio nustatymo metodą šis rodiklis sudaro 20,9%, tai pats blogiausias atpažinimo rezultatas. Naudojant patobulintą adaptyvų binarizacijos slenksčio nustatymą ir adaptyvų binarizacijos slenksčio nustatymą šis rodiklis sudaro 87,3%. Naudojant kitus binarizacijos slenksčio nustatymo metodus rodiklis sudaro apie 66%.



39 pav. Skraidančio roboto objekto sekimo įterptinės sistemos telemetrija: žymeklio indentifikavimo rezultatai

Skraidančio roboto objekto sekimo įterptinė sistema turi identifikuoti žymeklius. Eksperimento metu naudojamas žymeklio identifikavimo metodas labai jautrus kampui tarp kameros matricos plokštumos ir žymeklio plokštumos. Ateityje šis žymeklio identifikavimo metodas galėtų būti patobulintas. Atsižvelgiant į žymeklio identifikavimo metodo trūkumus žymeklio identifikavimas ypač priklauso nuo naudojamo binarizacijos slenksčio nustatymo metodo ir žymeklio atpažinimo rezultato. Žymeklio identifikavimas reikalauja atpažinti visus 4 žymeklio raktinius taškus, o žymeklio atpažinimui užtenka atpažinti 3 raktinius taškus.

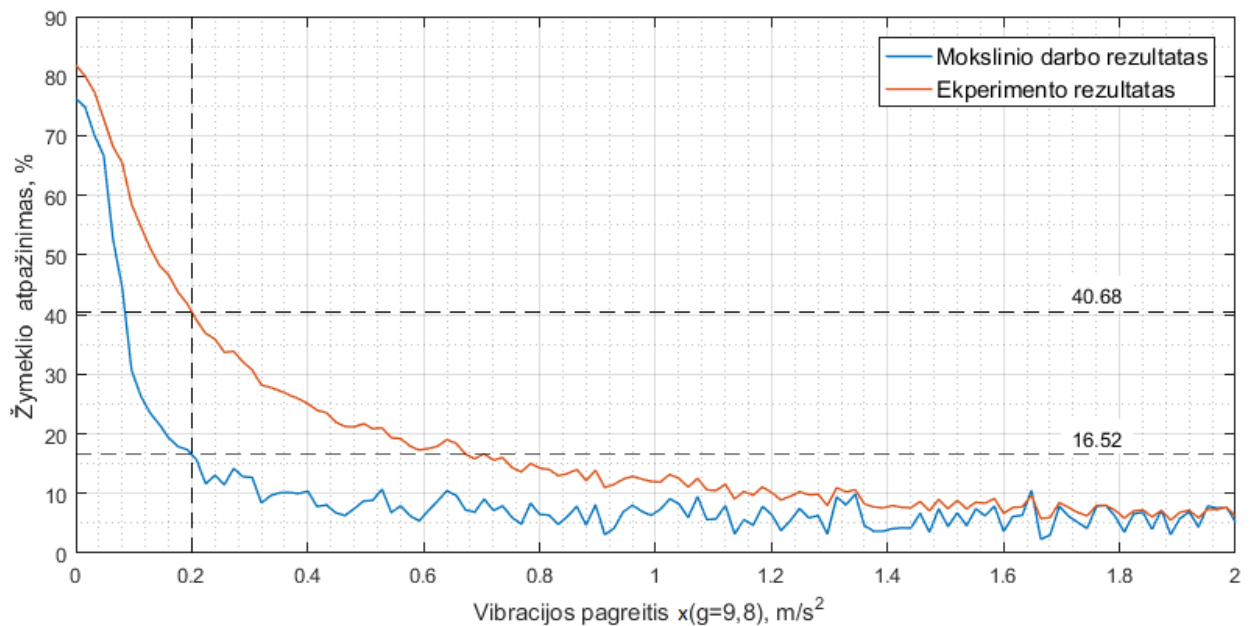
7 lentelė. Žymeklio identifikavimo rodikliai naudojant skirtingus binarizacijos slenksčio nustatymo metodus

Binarizacijos slenksčio nustatymo metodas	TPR (%)	TNR (%)	PPV (%)	NPV (%)	ACC (%)
(P)	5,1	100,0	100,0	13,6	17,5
(A)	83,3	100,0	100,0	47,2	85,5
(L)	69,1	100,0	100,0	32,6	73,1
(S)	72,6	100,0	100,0	35,2	76,1
(N)	86,5	100,0	100,0	52,5	88,3

Naudojant dvejetainio klasifikavimo metodus, pagal (16)–(20) formules, buvo paskaičiuoti žymeklio identifikavimo statistiniai rodikliai naudojant skirtingus binarizacijos slenksčio nustatymo metodus. Įvertinant statistinius rodiklius, kurie pateikti 7 lentelėje, iš karto atkreipiamas dėmesys į TNR ir PPV rodiklius, kurie sudaro 100%. Šie rodikliai maksimalūs, nes skraidančio roboto objekto sekimo įterptinė sistema visada naudoja atpažinto žymeklio rezultatus. Lyginant NPV rodiklius (7 lentelė) aukščiausias yra naudojant patobulintą adaptyvų binarizacijos slenksčio nustatymo metodą – 52,5%, šiek tiek blogesnis rezultatas yra naudojant adaptyvaus binarizacijos slenksčio nustatymo metodą – 47,2%. Aukščiausias skraidančio roboto objekto sekimo įterptinės sistemos tikslumo rodiklis (ACC) pasiektas naudojant patobulintą adaptyvaus binarizacijos slenksčio nustatymo metodą – 88,3%.

4.4 Skraidančio roboto įterptinės objekto sekimo sistemos atsparumo vibracijai eksperimentinis tyrimas

Anksčiau atliktame moksliniame tyrime [32] nustatytas vibracijos poveikis objekto atpažinimui. Eksperimentas buvo pakartotas naudojant šiame darbe pirmo eksperimento metu nustatytą geriausią binarizacijos slenksčio nustatymo metodą (patobulintą adaptyvų binarizacijos slenksčio nustatymo metodą) ir vietinių blokų binarizacijos slenksčio nustatymą.



40 pav. Mokslinio darbo [17] eksperimento pakartojimo rezultatas naudojant patobulintą adaptyvų slenksčio nustatymo metodą

Grafike (40 pav.) pavaizduoti eksperimentų rezultatai, su vertikalia punktyrine juoda linija pažymėta pasirinkta vibracijos pagreičio riba – 0,2g, kai žymeklis turėtų būti nebeatpažįstamas. Naudojant ankstesnį binarizacijos slenksčio nustatymo metodą, kai vibracijos pagreitis viršija 0,2g sėkmingų atpažinimo rezultatų kiekis sudaro mažiau nei ~17% nuo visų žymeklio atpažinimo bandymų. Naudojant patobulintą adaptyvų slenksčio nustatymo metodą skraidančio roboto vaizdo atpažinimu grindžiamą objekto sekimo įterptinės sistemos atsparumas vibracijai padidėjo dvigubai. Kai vibracijos pagreitis viršija 0,2g, sėkmingų atpažinimo rezultatų kiekis sudaro mažiau nei ~41% nuo visų žymeklio atpažinimo bandymų, todėl galima teigti, jog patobulintas binarizacijos adaptyvaus slenksčio nustatymo metodas yra žymiai atsparesnis vibracijai.

4.5 Skyriaus apibendrinimas

Eksperimentų metu nustatyta, kad patobulintas binarizacijos adaptyvaus slenksčio nustatymo metodas, labiausiai pagerina žymeklio atpažinimo, identifikavimo ir sekimo procesą. Naudojant patobulintą binarizacijos adaptyvaus slenksčio nustatymo metodą skraidančio roboto vaizdo atpažinimu grindžiamos objekto sekimo įterptinės sistemos žymeklio atpažinimo tikslumas padidėjo nuo 66,0% iki 87,3%, o žymeklio identifikavimo tikslumas padidėjo nuo 76,1% iki 88,3%. Patobulinto binarizacijos adaptyvaus slenksčio nustatymo vidutinė trukmė sumažėjo iki $(39,560 \pm 0,206)$ ms, lyginant su pagerintu metodu, naudojantis vietinių blokų binarizacijos slenksčio nustatymo rezultatus, vidutinė apdorojimo trukmė $(42,716 \pm 0,213)$ ms. Tačiau skraidančio roboto objekto sekimo įterptinės sistemos algoritmo apdorojimo trukmė, naudojant patobulintą binarizaciją su adaptyviu slenksčio nustatymu, vidutiniškai padidėjo nuo $(68,479 \pm$

0,399) ms iki $(69,712 \pm 0,451)$ ms. Tačiau toks nežymus trukmės padidėjimas yra neesminis, turint omenyje, kad šis algoritmas ženkliai padidina žymeklio atpažinimo tikslumą. Žymeklio identifikavimo algoritmas ateityje galėtų būti tobulinimas, nes šiame darbe pasiūlytas sprendimas leidžia identifikuoti iki 86,5%, visų atpažintų žymeklių.

IŠVADOS

1. Apžvelgus mokslinėje literatūroje panašiose objektų atpažinimo ir sekimo sistemose, naudojamus metodus ir algoritmus, skraidančio roboto įterptinei sistemai kurti objekto atpažinimui buvo pasirinkti binarizacijos slenksčio nustatymas ir kontūro aptikimo metodai, objekto sekimui ir pozicijos prognozavimui – Kalman filtras, objekto žymekliui – QR žymeklio sudarymo principas.
2. Siekiant pritaikyti vaizdo atpažinimu grindžiamus objekto atpažinimo ir sekimo metodus skraidantiems robotams turi būti parenkami sprendimai reikalaujantys mažiausiai sistemos resursų, tačiau pakankamai efektyvūs kokybiškai greitai apdorojantys vaizdo informaciją. Norint kuo mažiau prarasti kamera gaunamos informacijos, siūloma vaizdo apdorojimą atlikti skraidančio roboto platformoje. Taikant patobulintą vaizdo binarizacijos adaptivaus slenksčio nustatymo algoritmą bei žymeklio pozicijos prognozavimą galima užtikrinti efektyvų (sistemos resursų suvartojimo atžvilgiu) ir vaizdo nuskaitymo triukšmams atsparų objekto atpažinimą ir jo sekimą.
3. Pagal iškeltus funkcinius ir nefunkcinius reikalavimus suprojektuota įterptinė, vaizdo atpažinimu grindžiama, objekto sekimo sistema, sukurtas jos prototipas ir integruotas į skraidantį robotą. Sistemos verifikaciją parodė, jog parinkta techninė ir programinė įranga sistemai suteikia galimybę atpažinti ir sekti objekto QR žymeklį sudėtingoje aplinkoje, kai atstumas iki sekamo objekto žymeklio nedidesnis nei 15 m.
4. Eksperimentų metu nustatyta, kad patobulintas binarizacijos adaptivaus slenksčio nustatymo metodas, labiausiai pagerina žymeklio atpažinimo, identifikavimo ir sekimo procesą. Naudojant patobulintą binarizacijos adaptivaus slenksčio nustatymo metodą skraidančio roboto vaizdo atpažinimu grindžiamos objekto sekimo įterptinės sistemos žymeklio atpažinimo tikslumas padidėjo nuo 66,0% iki 87,3%, o žymeklio identifikavimo tikslumas padidėjo nuo 76,1% iki 88,3%. Patobulinto binarizacijos adaptivaus slenksčio nustatymo vidutinė trukmė sumažėjo nuo $(42,716 \pm 0,213)$ ms iki $(39,560 \pm 0,206)$ ms, lyginant su pagerintu metodu, naudojančiu vietinių blokų binarizacijos slenksčio nustatymą. Tačiau skraidančio roboto objekto sekimo įterptinės sistemos algoritmų apdorojimo trukmė, naudojant patobulintą binarizaciją su adaptyviu slenksčio nustatymu, vidutiniškai padidėjo nuo $68,479 \pm 0,399$ ms iki $69,712 \pm 0,451$ ms. Tačiau toks nežymus trukmės padidėjimas yra neesminis, turint omenyje, kad šis algoritmas ženkliai padidina žymeklio atpažinimo tikslumą. Žymeklio identifikavimo algoritmas ateityje galėtų būti tobulinimas, nes šiame darbe pasiūlytas sprendimas leidžia identifikuoti iki 86,5%, visų atpažintų žymeklių.

LITERATŪRA

- [1]. Y. Bi, H. Duan. „Implementation of autonomous visual tracking and landing for a low-cost quadrotor“, *Optik – International Journal for Light and Electron Optics*. 2013, vol. 124, iss. 18, pp. 3296–3300.
- [2]. A. B. Junaid, Y. Lee, Y. Kim „Design and implementation of autonomous wireless charging station for rotary-wing UAVs“ *Aerospace Science and Technology*, July 2016, vol 54, pp 253–266.
- [3]. G. Xu , Y. Zhang, S. Ji, Y. Cheng, Y. Tian „Research on computer vision-based for UAV autonomous landing on a ship“, *Pattern Recognition Letters*, 15 April 2009, vol 30, iss. 6, pp 600–605.
- [4]. B. Ludington, E. Johnson, G. Vachtsevanos, „Augmenting UAV autonomy“, *IEEE Robotics & Automation Magazine*, 2006, vol. 13, iss. 3, pp. 63–71.
- [5]. I. Culjak, D. Abram, T. Pribanic, H. Dzapo, M. Cifrek, „A brief introduction to OpenCV“, *MIPRO, 2012 Proceedings of the 35th International Convention*, Opatija, 21-25 May 2012, pp. 1725-1730.
- [6]. D. Maravalla, J. de Lopeb, J. P. Fuentes, „Vision-based anticipatory controller for the autonomous navigation of an UAV using artificial neural networks“, *Neurocomputing*, 3 March 2015, vol. 151, part 1, pp 101–107
- [7]. V. Lepetit, P. Fua. „Monocular Model- Based 3D Tracking of Rigid Objects“, *Foundations and Trends in Computer Graphics and Vision*, 2005, vol. 1, iss. 1, pp. 1–89.
- [8]. B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid And J. Tardos, „A comparison of loop closing techniques in monocular SLAM“, *Robotics and Autonomous Systems*, 2009, vol. 57, iss. 12, pp. 1188–1197.
- [9]. M. E. Campbell, W. W. Whitacre, „Cooperative Tracking Using Vision Measurements on SeaScan UAVs“, *IEEE Transactions on Control Systems Technology*, 2007, vol. 15, iss. 4, pp. 613–626.
- [10]. M. Hariti, Y. Ruichek, and A. Koukam, „A Fast Stereo Matching Method For Real Time Vehicle Front Perception With Linear Cameras“, *IEEE Intelligent Vehicles Symposium*, 9-11 June 2003, pp. 247-252.
- [11]. E. Andreas, K. Schindler, L. Bastian, L.V. Gool „Object detection and tracking for autonomous navigation in dynamic environments“ [tinkle][žiūrėta 2016.04.23] Prieiga per internetą https://www.researchgate.net/publication/220121880_Object_Detection_and_Tracking_for_Autonomous_Navigation_in_Dynamic_Environments
- [12]. H. Endresen „Tracking objects in 3D using Stereo Vision“ [tinkle][žiūrėta 2016.02.19] Prieiga per internetą <http://www.diva-portal.org/smash/get/diva2:347993/FULLTEXT01.pdf>
- [13]. R. I. Hartley, A. Zisserman „Multiple View Geometry in Computer Vision“, *Cambridge University Press*, 2004, pp 670.
- [14]. G. Xu, X. Qi, Q. Zeng, Y. Tian, Ruipeng Guo, Biao Wang, „Use of land’s cooperative object to estimate UAV’s pose for autonomous landing“, *Chinese Journal of Aeronautics*, Dec. 2013, vol. 26, iss. 6, pp 1498–1505.
- [15]. P. Ben-Tzvi, X. Xu, „An embedded feature-based stereo vision system for autonomous mobile robots“, *Robotic and Sensors Environments (ROSE), 2010 IEEE International Workshop on*, Phoenix, AZ, 15-16 Oct. 2010, pp. 1-6.
- [16]. X. Wanga, , X. Konga, J. Zhia, Y. Chena, X. Dong, „Real-time drogue recognition and 3D locating for UAV autonomous aerial refueling based on monocular machine vision“, *Chinese Journal of Aeronautics*, December 2015, vol. 28, iss. 6, pp 1667–1675.
- [17]. P Bhattacharya, M. Gavrilova, „Intelligent Computer Graphics 2012“. *Berlin: Springer*, 2013.
- [18]. C. Chena, A. C. Kota, H. Yangb, „A two-stage quality measure for mobile phone captured 2D barcode images“, *Pattern Recognition*, September 2013, vol. 46, iss. 9, pp 2588–2598.

- [19]. D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, D. Schmalstieg. „Real-Time Detection and Tracking for Augmented Reality on Mobile Phones“, *IEEE Transactions on Visualization and Computer Graphics*. 2010, vol. 16, iss. 3, pp. 355–368.
- [20]. G. Klein, D. Murray. „Parallel Tracking and Mapping for Small AR Workspaces“. In: *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. Nara: IEEE, 2007, pp. 225–234.
- [21]. VIOLA, P. and M. JONES. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*. Hawaii: IEEE, 2001, pp. 511.
- [22]. N. Sawasaki, M. Nakao, Y. Yamamoto, K. Okabayashi, „Embedded Vision System for Mobile Robot Navigation“, *Proc of the 2006 IEEE Int Conf on Robotics and Automation*, Orlando, Florida. 15-19 May 2006, pp: 2693-2698.
- [23]. H. Endresen „Tracking objects in 3D using Stereo Vision“ [tinkle][žiūrėta 2016.01.17] Prieiga per internetą <http://www.diva-portal.org/smash/get/diva2:347993/FULLTEXT01.pdf>
- [24]. E. Andreas, K. Schindler, L. Bastian, L.V. Gool „Object detection and tracking for autonomous navigation in dynamic environments“ [tinkle][žiūrėta 2015.12.17] Prieiga per internetą http://www.igp.ethz.ch/photogrammetry/publications/pdf_folder/ess10ijrr.pdf
- [25]. T. Ye, L. Haitao, G. Haiyan, L. Xiangguo, „Road tracking by circular template matching from high resolution remotely sensed imagery“, *Urban Remote Sensing Event, 2009 Joint*, Shanghai, 20-22 May 2009, pp. 1-5.
- [26]. Y. Mochizuki, Ch. Univ, A. Imiya, A. Torii, „Circle-Marker Detection Method for Omnidirectional Images and its Application to Robot Positioning“, *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference*, Rio de Janeiro, 14-21 Oct. 2007, pp 1-8.
- [27]. X. Qian, C. Chakrabarti, L. Karam, „A distributed Canny edge detector and its implementation on FPGA“, *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011 IEEE*, Sedona, AZ, 4-7 Jan. 2011, pp. 500-505.
- [28]. Q. Yujing, T. Yanchao, L. Junshi, „Improved Harris sub-pixel corner detection algorithm for chessboard image“, *Measurement, Information and Control (ICMIC), 2013*, Harbin, 16-18 Aug. 2013, pp. 1408-1411
- [29]. F. Abdat, C. Maaoui, A. Pruski, „Real time facial feature points tracking with Pyramidal Lucas-Kanade algorithm“, *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium*, Munich, 1-3 Aug. 2008, pp. 71-76.
- [30]. N. Ray, S. Acton, „Tracking rolling leukocytes with motion gradient vector flow“, *Conference Record of the Thirty-Seventh Asilomar*, 9-12 Nov. 2003, pp. 1948-1952.
- [31]. A. Kaehler, G. Bradski, „Learning opencv: computer vision with the opencv library“, *O'Reilly Media*, September 2008, pp 571.
- [32]. A. Andziulis, D. Drungilas, V. Glazko, E. Kiseliovas, „Resource Saving Approach of Visual Tracking Fiducial Marker Recognition“, *Information and communication technologies and services*, 2015, vol. 13, num. 4, iss. special, pp. 359-366.
- [33]. S. Ali, V. Reilly, M. Shah, „Motion and Appearance Contexts for Tracking and Re-Acquiring Targets in Aerial Videos“, In: *Conference on Computer Vision and Pattern Recognition*. Minneapolis: IEEE, 2007, pp. 1–6.
- [34]. J. Lou, T. Tan, W. Hu, H. Yang, S. J. Maybank, „3-D model-based vehicle tracking“, *IEEE Transactions on Image Processing*. 2005, vol. 14, iss. 10, pp. 1561–1569.
- [35]. H. Zhou, Y. Yuan, C. Shi, „Object tracking using SIFT features and mean shift“, *Computer Vision and Image Understanding*, 2009, vol. 113, iss. 3, pp. 345–352.

- [36]. S. Lee, H. Park, S.. Hwang, „Motion deblurring using edge map with blurred/noisy image pairs“, *Optics Communications*, 2012, vol. 285, iss. 7, pp. 1777–1786.
- [37]. O. K. Hamilton, T. P. Breckon, X. Bai, S. Kamata, „A foreground object based quantitative assessment of dense stereo approaches for use in automotive environments“, *Image Processing (ICIP), 2013 20th IEEE International Conference on*, Melbourne, VIC, 15-18 Sept. 2013, pp. 418-422.
- [38]. H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, K. Tachibana. „Virtual object manipulation on a table-top AR environment“. In: *Proceedings IEEE and ACM International Symposium on Augmented Reality. Munich: IEEE*, 2000, pp. 111–119.
- [39]. L. Naimark, E. Foxlin, „Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker“, In: *Proceedings. International Symposium on Mixed and Augmented Reality*. Darmstadt: IEEE, 2002, pp. 27–36.
- [40]. J. Rekimoto, Y. Ayatsuka, „CyberCode: designing augmented reality environments with visual tags“, In: *Proceedings of DARE 2000 on Designing augmented reality environments*. Elsinore: ACM, 2000, pp. 1–10.
- [41]. M. Rohs, B. Gfeller, „Using camera equipped mobile phones for interacting with realworld objects“, In: *Advances in Pervasive Computing*. Linz: OCG, 2004, pp. 1–7.
- [42]. H. S. Al-Khalifa, „Utilizing QR Code and Mobile Phones for Blinds and Visually Impaired People“, In: *Proceedings of the 11th International Conference on Computers Helping People with Special Needs*. Linz: Springer-Verlag, 2008, pp. 1065–1069.
- [43]. D. Claus, A. Fitzgibbon, „Reliable automatic calibration of a marker-based position tracking system“, In: *Workshop on the Applications of Computer Vision*, Breckenridge: IEEE, 2005, pp. 300–305.
- [44]. O. Parvu, A. G. Balan, „A method for fast detection and decoding of specific 2D barcodes“, In: *17th Telecommunications Forum*, Belgrade: IEEE, 2009, pp.
- [45]. S. Garrido-Jurado, R. Munoz-Salinas, F. J. Madrid-Cuevas, M. J. Marinjimenez, „Automatic generation and detection of highly reliable fiducial markers under occlusion“, *Pattern Recognition*, 2014, vol. 47, iss. 6, pp. 2280–2292.
- [46]. L. F. F. Belussi, N. S. T. Hirata, „Fast component-based QR code detection in arbitrarily acquired images“, *Journal of Mathematical Imaging and Vision*, 2013, vol. 45, iss. 3, pp. 277–292.
- [47]. T. Pattersona, S. McCleana, P. Morrowa, G. Parra, C. Luo, „Timely autonomous identification of UAV safe landing zones“, *Image and Vision Computing*, September 2014, vol. 32, iss. 9, pp. 568–578.
- [48]. R. Neves, A.C. Matos, „Raspberry PI based stereo vision for small size ASVs“, *Oceans - San Diego*, 2013, San Diego, CA, 23-27 Sept. 2013, pp. 1-6.
- [49]. J.-Y. Bouguet, „Pyramidal Implementation of the AÆne Lucas Kanade Feature Tracker Description of the algorithm“ [tinkle][žiūrëta 2016.04.05] Prieiga per internetą http://robots.stanford.edu/cs223b04/algos/affine_tracking.pdf
- [50]. Digital vs. Analog Wireless Video Transfer (DESi WDV Test Records) [tinkle][žiūrëta 2016.04.05] Prieiga per internetą https://www.youtube.com/watch?v=ImW_z3QBMw0
- [51]. S. Suzuki, B. KeiichiA, „Topological structural analysis of digitized binary images by border following“, *Computer Vision, Graphics, and Image Processing*, April 1985, vol. 30, iss. 1, pp 34-46.
- [52]. D. Shaha, V. Bharadib, „IoT based Biometrics Implementation on Raspberry Pi“, *Procedia Computer Science*, 2016, vol. 79, pp 328-336.
- [53]. RPi Hardware - eLinux.org [tinkle][žiūrëta 2016.03.27] Prieiga per internetą http://elinux.org/RPi_Hardware

- [54]. ODROID-XU4 Development Board is a Smaller and Cheaper Version of ODROID-XU3 Board [tinkle][žiūrėta 2016.03.27] Prieiga per internetą <http://www.cnx-software.com/2015/07/14/odroid-xu4-board-is-a-smaller-and-cheaper-version-of-odroid-xu3/>
- [55]. ODROID | Hardkernel [tinkle][žiūrėta 2016.03.27] Prieiga per internetą http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143703355573
- [56]. Showdown: Raspberry Pi 2 vs ODROID C1 vs HummingBoard vs MIPS Creator CI20 (updated) [tinkle][žiūrėta 2016.03.28] Prieiga per internetą <http://www.androidauthority.com/raspberry-pi-2-vs-odroid-c1-vs-hummingboard-vs-mips-creator-ci20-599418/>
- [57]. Camera Module - Raspberry Pi Documentation [tinkle][žiūrėta 2016.03.27] Prieiga per internetą <https://www.raspberrypi.org/documentation/hardware/camera>
- [58]. Raspberry Pi [tinkle][žiūrėta 2016.04.29] Prieiga per internetą <http://www.truetex.com/raspberrypi>
- [59]. Logitech C905 720p Webcam Specifications [tinkle][žiūrėta 2016.04.28] Prieiga per internetą <http://www.cnet.com/products/logitech-c905-720p-webcam/specs/>
- [60]. Ask about focal length of webcam C905 - Logitech Forums [tinkle][žiūrėta 2016.04.29] Prieiga per internetą <http://forums.logitech.com/t5/Webcams/Ask-about-focal-length-of-webcam-C905/td-p/509454>
- [61]. How can I know the focal length and sensor size of my webcam? - Photography Stack Exchange [tinkle][žiūrėta 2016.04.30] Prieiga per internetą <http://photo.stackexchange.com/questions/61724/how-can-i-know-the-focal-length-and-sensor-size-of-my-webcam>
- [62]. J. Lenchner, C. Isc, J. O. Kephart, C. Mansley, J. Connell, S. McIntosh, „Towards Data Center Self-Diagnosis Using a Mobile Robot“ [tinkle][žiūrėta 2016.04.05] Prieiga per internetą <http://www.chrismansley.com/doc/icac11.pdf>
- [63]. RaspiCam: C++ API for using Raspberry camera with/without OpenCv | Aplicaciones de la Visión Artificial [tinkle][žiūrėta 2016.04.15] Prieiga per internetą <http://www.uco.es/investiga/grupos/ava/node/40>
- [64]. OSM5D3Z2C1P Datasheet [tinkle][žiūrėta 2016.04.15] Prieiga per internetą <http://www.tme.eu/en/Document/599172d795be5ae448905759b6730838/osm5d3z2c1p.pdf>
- [65]. TXS0104ED Datasheet [tinkle][žiūrėta 2016.04.15] Prieiga per internetą <http://www.ti.com/lit/ds/symlink/txs0104e.pdf>
- [66]. V. Glazko, „3D vaizdų panaudojimo sistema erdviniam objekto sekimui skraidančiu robotu“, bakalauro baigiamasis darbas, 2014.
- [67]. Konspektai [tinkle][žiūrėta 2016.04.28] Prieiga per internetą http://inga.vgtu.lt/~akrl/Medziaga/Studiju_medziaga/2008P/MRU/6/6.htm
- [68]. A.Krilavičiūtė, G.Smilytė, „Diagnosticiniai testai. Jautrumo ir specifiškumo vertinimas. Teigiama ir neigiama prognozinės vertės“ [tinkle][žiūrėta 2016.04.28] Prieiga per internetą http://old.nvi.lt/loc/m/m_files/wfiles/file699.pdf

SUMMARY

Today the popularity of flying robots are growing, from tracking area to transporting small loads. It's uptime is limited by battery charge. Automatic charging system is needed for stand-alone flying robots. One of the variants is to use special platforms landed on which it can charge its battery for subsequent flight. The given method requires a precise landing. GPS provides the accuracy of the landing up to 3 meters, using of additional adjustments from the terrestrial stations decreases uncertainty to 1 centimeter, however it depends on the following factors: do the flying robot receives the same GPS signal as the terrestrial station, and the response speed of the terrestrial station to flying robot. Solution - use image recognition system.

This paper presents a system for tracking an object using image recognition flying robot. Using techniques that have been presented in similar systems and are presented in the scientific literature, they have been studied and selected the most suitable for use on a flying robot. Platform marker was designed based on QR marker. To implement the system (recognition of objects in images) OpenCV computer vision library was used. To prevent loss of information received from the camera, it was decided to make all the calculations on the flying robot platform.

The practical part of the work includes the establishment of requirements for object tracking system and component selection and design of the technical and programmatic parts. The system created on the basis of the Comptroller "Raspberry PI two Model B" and built on the platform of a flying robot "Parallax Elev-8" made it possible to conduct an experiment and identify the most appropriate method for the establishment of a binary image threshold. Three different methods were examined, also two of their improved version.

Control tests of the system were conducted using different methods for the establishment of a binary image threshold, as a result of which revealed the advantage of using an improved adaptive method. Lead time is $39,560 \pm 0,206$ ms as compared to other best method of $42,716 \pm 0,213$ ms.

КРАТКОЕ ИЗЛОЖЕНИЕ

В настоящее время популярность использования летающих роботов увеличивается, способы их применения различные, от слежения за территорией до транспортировки небольших грузов. Время работы ограничено зарядом аккумулятора летающего робота. Для автономности летающих роботов, необходима система автоматической зарядки. Один из

вариантов использовать специальные платформы, приземлившись на которые летающий робот сможет зарядить аккумулятор для последующего полета. Предложенный метод требует точной посадки. GPS системы позиционирования обеспечивают точность во время посадки до 3 метров, дополнительное использование корректировок наземных станций уменьшает погрешность до 1 см, однако зависит от следующих факторов: одинаковые ли сигналы спутников ловят GPS летающего робота и наземная станция, а также от скорости отклика наземной станции к летающему роботу. Решение – использовать систему распознавания изображений.

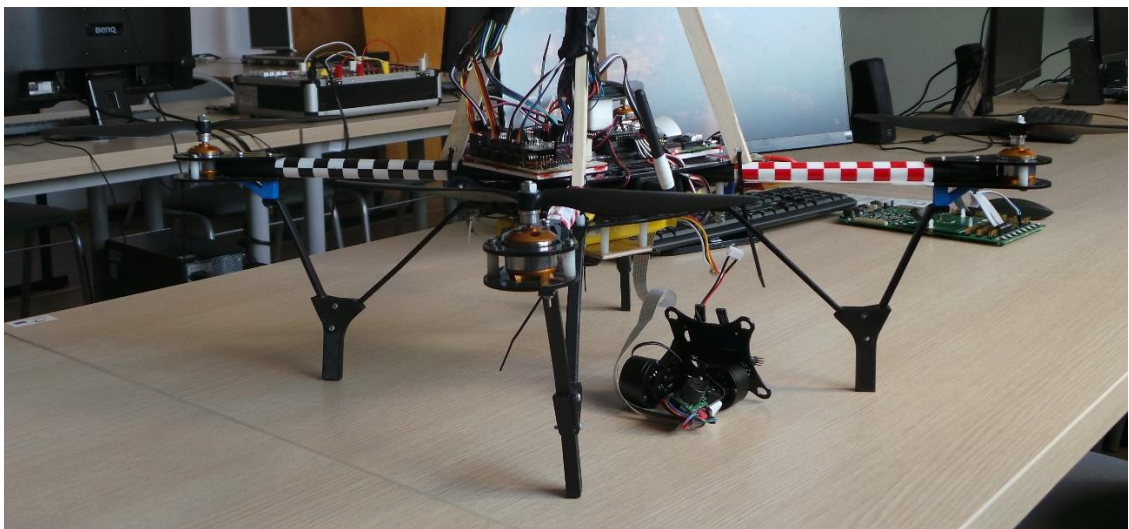
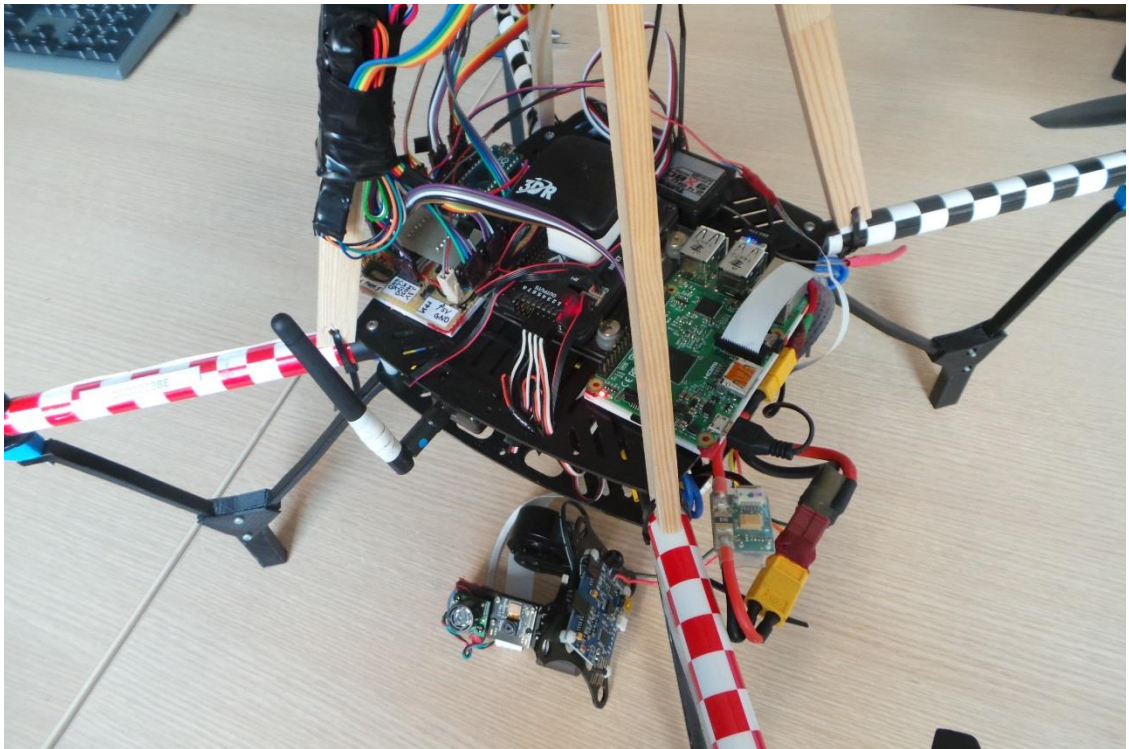
В этой работе представлена система слежения за объектом используя распознавание изображений летающим роботом. Используя методы, которые были представлены в схожих системах и представлены в научной литературе были изучены и выбраны наиболее подходящие для использования на летающем роботе. Был спроектирован маркер платформы на основе QR маркера. Для реализации системы, а в точности распознавания объектов на изображениях, была использована библиотека компьютерного зрения *OpenCV*. Во избежание потери информации, получаемой с камеры, решено производить все вычисления на платформе летающего робота.

Практическая часть работы включает постановление требований к системе слежения за объектом, выбора компонентов, а также проектирование технической и программной частей. Созданная система на базе контролера “*Raspberry PI 2 Model B*” и встроенная на платформу летающего робота “*Parallax Elev-8*” дала возможность провести эксперимент и выявить наиболее лучший метод установлении двоичного порога изображения. Были изучены 3 разных методов и 2 два улучшенных их варианта.

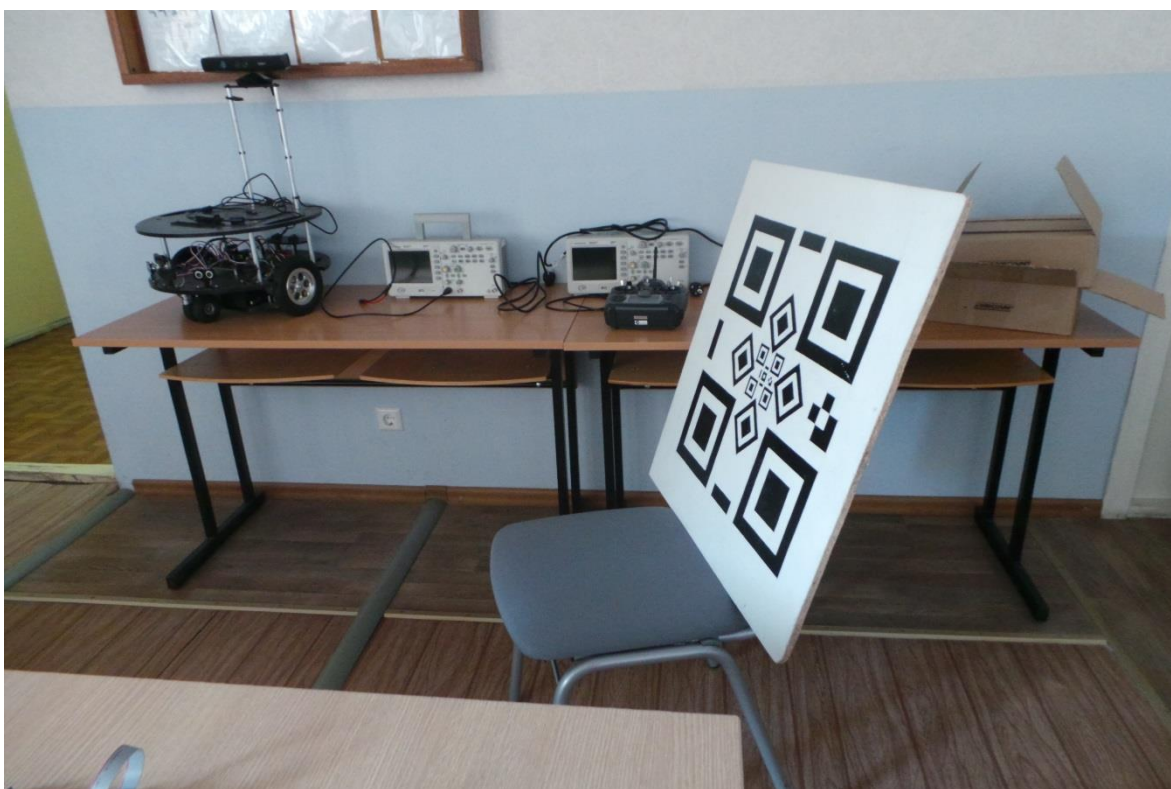
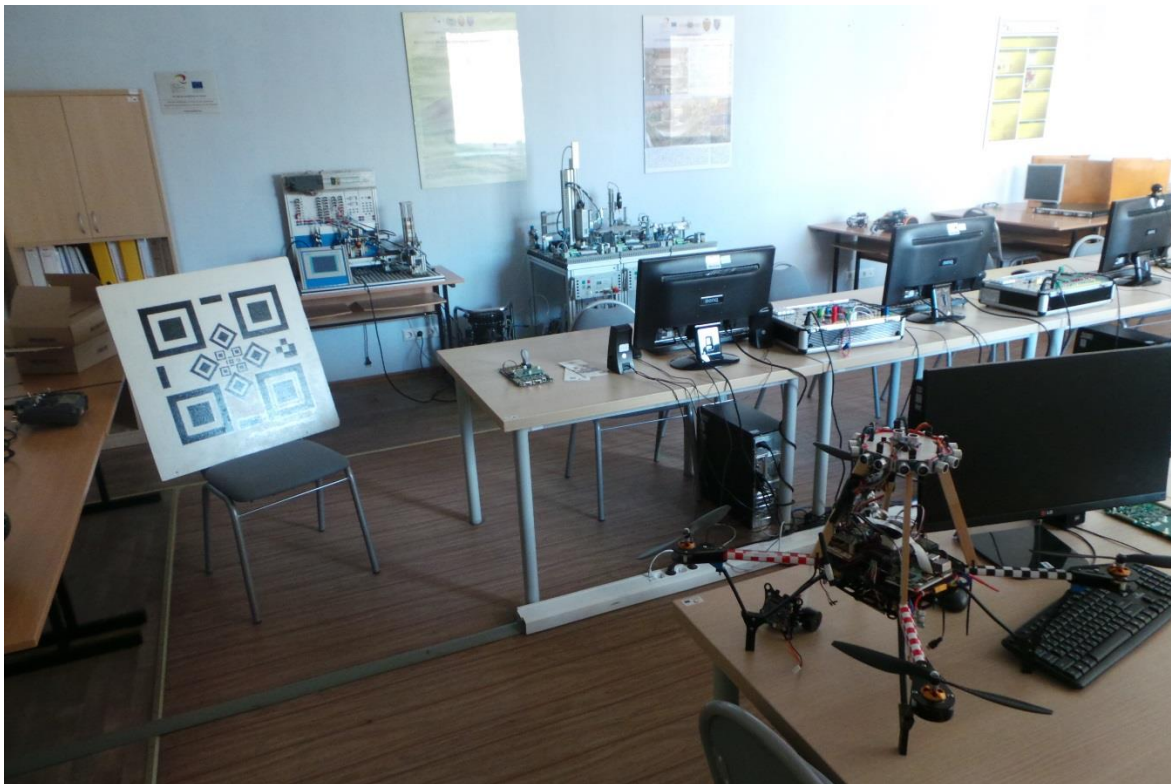
Была проведены контрольные испытания системы используя разные методы установлении двоичного порога изображения, в результате которых выявлено преимущество в использовании улучшенного адаптивного метода. Время выполнения составляет $39,560 \pm 0,206$ мс, по сравнению с другим наиболее лучшим методом $42,716 \pm 0,213$ мс.

PRIEDAI

SURINKTOS SISTEMOS NUOTRAUKOS



EKSPERIMENTO SĄLYGOS



EKSPERIMENTO DUOMENYS

PAPRASTAS BINARIZACIJOS SLENKSČIO NUSTATYMAS

Žymeklio atpažinimas

	Žymeklis yra kadre	Žymeklio nėra kadre	
Žymeklis atpažintas	4136	33	4169
Žymeklis neatpažintas	5082	1342	6424
	9218	1375	21186

Žymeklio identifikavimas

	Žymeklis yra kadre	Žymeklio nėra kadre	
Žymeklis identifikuotas	473	0	473
Žymeklis ne identifikuotas	8745	1375	10120
	9218	1375	21186

ADAPTYVUS BINARIZACIJOS SLENKSČIO NUSTATYMAS

Žymeklio atpažinimas

	Žymeklis yra kadre	Žymeklio nėra kadre	
Žymeklis atpažintas	9020	11	9031
Žymeklis neatpažintas	198	1364	1562
	9218	1375	21186

Žymeklio identifikavimas

	Žymeklis yra kadre	Žymeklio nėra kadre	
Žymeklis identifikuotas	7678	0	7678
Žymeklis ne identifikuotas	1540	1375	2915
	9218	1375	21186

**BINARIZACIJOS SLENKSČIO NUSTATYMAS NAUDOJANT VIETINIUS
BLOKUS**

Žymeklio atpažinimas

	Žymeklis yra kadre	Žymeklio nėra kadre	
Žymeklis atpažintas	8536	22	8558
Žymeklis neatpažintas	682	1353	2035
	9218	1375	21186

Žymeklio identifikavimas

	Žymeklis yra kadre	Žymeklio nėra kadre	
Žymeklis identifikuotas	6369	0	6369
Žymeklis ne identifikuotas	2849	1375	4224
	9218	1375	21186

**PAGERINTAS BINARIZACIJOS SLENKSČIO NUSTATYMAS NAUDOJANT
VIETINIUS BLOKUS**

Žymeklio atpažinimas

	Žymeklis yra kadre	Žymeklio nėra kadre	
Žymeklis atpažintas	8514	11	8525
Žymeklis neatpažintas	704	1364	2068
	9218	1375	21186

Žymeklio identifikavimas

	Žymeklis yra kadre	Žymeklio nėra kadre	
Žymeklis identifikuotas	6688	0	6688
Žymeklis ne identifikuotas	2530	1375	3905
	9218	1375	21186

PATOBULINTAS ADAPTYVUS BINARIZACIJOS SLENKSČIO NUSTATYMAS

Žymeklio atpažinimas

	Žymeklis yra kadre	Žymeklio nėra kadre	
Žymeklis atpažintas	9020	11	9031
Žymeklis neatpažintas	198	1364	1562
	9218	1375	21186

Žymeklio identifikavimas

	Žymeklis yra kadre	Žymeklio nėra kadre	
Žymeklis identifikuotas	7975	0	7975
Žymeklis ne identifikuotas	1243	1375	2618
	9218	1375	21186

KOMPAK TINĖ PLOKŠTELĖ

Kompaktinis diskas, kuriame pateikiami:

1. Darbo katalogas: „Vitalij_Glazko/..“.
2. Darbo aprašymas: „Vitalij_Glazko/Vitalij_Glazko_magistrinis_darbas.pdf“.
3. Suprojektuoti su „SolidWorks 2015“ programiniu paketu komponentai:
„Vitalij_Glazko/solidworks/..“.
4. Darbo paveiksliukai: „Vitalij_Glazko/paveiksliukai/..“.
5. Nuotraukos: „Vitalij_Glazko/nuotraukos/..“.
6. Sistemos algoritmai: „Vitalij_Glazko/algoritmai/..“.

MOKSLINĖS PUBLIKACIJOS