



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**

**Adomas Ivanovas**

**KLIŪČIŲ IŠVENGIMO BEPILOČIAMS ORLAIVIAM  
ALGORITMŲ TYRIMAS**

Baigiamasis magistro projektas

**Vadovas**

Doc. dr. Armantas Ostreika

**KAUNAS, 2017**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**INFORMATIKOS FAKULTETAS**

**KLIŪČIŲ IŠVENGIMO BEPILOČIAMS ORLAIVIAMS**  
**ALGORITMŲ TYRIMAS**

Baigiamasis magistro projektas  
**Informatika (kodas 621I10003)**

**Vadovas**

(parašas) Doc. dr. Armantas Ostreika  
(data)

**Recenzentas**

(parašas) Lekt. dr. Kęstutis Jankauskas  
(data)

**Projektą atliko**

(parašas) Adomas Ivanovas  
(data)

**KAUNAS, 2017**



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Adomas Ivanovas

(Studento vardas, pavardė)

Informatika, 621I10003

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Kliūčių išvengimo bepiločiams orlaiviams algoritmų tyrimas“

### AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 \_\_\_\_\_ m. \_\_\_\_\_ d.  
Kaunas

Patvirtinu, kad mano, **Adomo Ivanovo**, baigiamasis projektas tema „Kliūčių išvengimo bepiločiams orlaiviams algoritmų tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

\_\_\_\_\_  
(vardą ir pavardę įrašyti ranka)

\_\_\_\_\_  
(parašas)

# TURINYS

1	ĮVADAS.....	9
2	KLIŪČIŲ IŠVENGIMO BEPILOČIAMS ORLAIVIAMS METODŲ APŽVALGA.....	10
2.1	Apžvalgos tikslas .....	10
2.2	Objektų aptikimo metodai.....	10
2.2.1	Radarai .....	10
2.2.2	Pasyvūs radarai .....	11
2.2.3	Lidar.....	12
2.2.4	Radijo atsakikliai .....	13
2.2.5	Optiniai objektų aptikimo metodai .....	15
2.2.5.1	„Block match“ algoritmas.....	15
2.2.5.2	„Semi global block match“ algoritmas .....	16
2.3	Apžvalgos apibendrinimas .....	16
3	BEPILOČIŲ ORLAIVIŲ ĮRANGOS APŽVALGA .....	17
3.1	Apžvalgos tikslas .....	17
3.2	Techninė įranga.....	17
3.2.1	Autonominio skrydžio valdikliai .....	17
3.2.2	Skrydžio kompiuteris.....	18
3.2.3	Erdvinio vaizdo kamera .....	19
3.3	Programinė įranga .....	20
3.4	Apžvalgos apibendrinimas .....	20
4	ERDVINIO VAIZDO ALGORITMŲ ANALIZĖ .....	21
4.1	Analizės tikslas .....	21
4.2	Erdvinio vaizdo kameros kalibravimas .....	21
4.3	„Block match“ algoritmas .....	25
4.4	„Semi global block match“ algoritmas .....	27
4.5	Erdvinio vaizdo algoritmų greitaveikos testavimas .....	28
4.6	Analizės išvados.....	32
5	ORLAIVIO KLIŪČIŲ IŠVENGIMO SISTEMOS PROJEKTAVIMAS .....	33
5.1	Projektavimo tikslas .....	33
5.2	Techninės įrangos projektavimas.....	33
5.3	Programinės įrangos projektavimas .....	36
5.4	Išvados .....	38
6	KLIŪČIŲ IŠVENGIMO EKSPERIMENTAS .....	39
6.1	Eksperimento planas .....	39
6.2	Eksperimento eiga.....	39
6.3	Eksperimento rezultatai.....	40
7	IŠVADOS.....	43

## PAVEIKSLŲ SĄRAŠAS

2.1 pav. Radaro veikimo schema.....	11
2.2 pav. Pasyvaus radaro veikimo schema .....	12
2.3 pav. „Lidar“ technologija sugeneruotas 3d žemėlapis [13].....	12
2.4 pav. TCAS II oro zona.....	14
2.5 pav. Erdvinio vaizdo generavimo diagrama .....	15
2.6 pav. „Block match“ algoritmu sugeneruotas erdvinis vaizdas .....	15
2.7 pav. „Semi global block match“ algoritmu sugeneruotas erdvinis vaizdas.....	16
3.1 pav. APM autonominio skrydžio valdiklis .....	17
3.2 pav. „Pixhawk“ autonominio skrydžio valdiklis .....	17
3.3 pav. Skrydžio kompiuterių testų rezultatai [9] .....	18
3.4 pav. „Odroid XU4“ kompiuteris.....	19
3.5 pav. „Playstation 4“ kamera .....	20
4.1 pav. Kameros vaizdo iškraipymas .....	21
4.2 pav. Kalibravimui naudojama šachmatų lenta.....	22
4.3 pav. „CornerSubPix“ funkcijos veikimo diagrama .....	22
4.4 pav. Šachmatų lentos aptikimas .....	23
4.5 pav. Epipolinės linijos .....	24
4.6 pav. Sukalibruota kamerų pora .....	24
4.7 pav. Makroblokų lyginimas.....	25
4.8 pav. SAD metodu realizuotas erdvinis vaizdas .....	26
4.9 pav. Paveikslų epipolinės linijos .....	27
4.10 pav. „Semi global block match“ algoritmu sugeneruotas erdvinis vaizdas.....	27
4.11 pav. Paveikslų pora – motociklas .....	28
4.12 pav. Testavimo rezultatai - motociklas.....	28
4.13 pav. Paveikslų pora - gėlės .....	29
4.14 pav. Testavimo rezultatai – gėlės .....	29
4.15 pav. Paveikslų pora - kuprinė .....	30
4.16 pav. Testavimo rezultatai - kuprinė .....	30
4.17 pav. Greitaveikos testavimo rezultatai .....	31
5.1 pav. „Playstation 4“ kameros jungtis [23].....	33
5.2 pav. USB TTL signalo keitiklis.....	33
5.3 pav. „Turnigy 9x“ siųstuvas .....	34
5.4 pav. Techninės įrangos diagrama .....	34
5.5 pav. Kliūčių išvengimo lėktuvo modelis .....	35
5.6 pav. BM parametrų testavimo grafinė sąsaja .....	37
5.7 pav. SGBM parametrų testavimo grafinė sąsaja .....	37
5.8 pav. Kliūties erdvinis vaizdas.....	38
5.9 pav. Kliūties nustatymas.....	38
6.1 pav. Autonominio skrydžio planas .....	39
6.2 pav. Vaizdo kameros medžiaga.....	40
6.3 pav. Skrydžio kelias.....	41
6.4 pav. BM algoritmo testavimo pirmoji iteracija .....	41
6.5 pav. BM algoritmo testavimo antroji iteracija.....	42
6.6 pav. BM algoritmo testavimo trečioji iteracija.....	42

## LENTELIŲ SĄRAŠAS

2.1 lentelė „Lidar“ jutikliai .....	13
2.2 lentelė TCAS signalai .....	14
3.1 lentelė Autonominio skrydžio valdiklių palyginimas .....	18
3.2 lentelė Skrydžio kompiuterių testų rezultatai [9] .....	19
3.3 lentelė Erdvinio vaizdo kamerų palyginimas .....	19
4.1 lentelė Testavimo rezultatai - motociklas .....	29
4.2 lentelė Testavimo rezultatai - gėlės .....	30
4.3 lentelė Testavimo rezultatai - kuprinė .....	31
4.4 lentelė Algoritmų vykdymo laiko vidurkiai .....	31
4.5 lentelė Erdvinio vaizdo kokybės vertinimas .....	32
5.1 lentelė Orlaivio pasirinkimo kriterijai .....	35
5.2 lentelė Komponentų svoriai ir kainos .....	36
6.1 lentelė Testavimo atvejai .....	39
6.2 lentelė BM algoritmo testavimo pirmos iteracijos duomenys .....	41
6.3 lentelė BM algoritmo testavimo antros iteracijos duomenys .....	42
6.4 lentelė BM algoritmo testavimo trečios iteracijos duomenys .....	42

Ivanovas, Adomas. Kliūčių išvengimo bepiločiams orlaiviams algoritmų tyrimas. Magistro baigiamasis projektas / vadovas doc. dr. Armantas Ostreika; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: Informatika

Reikšminiai žodžiai: *erdvinis vaizdas, kliūčių išvengimas, bepiločiai orlaiviai*

Kaunas, 2017. 45 p.

## SANTRAUKA

*Šiuo metu sparčiai didėja orlaivių naudojimas. Jų naudojimas įvairus: mėgėjiškas skraidymas, žvalgybos misijos, pasėlių priežiūra, trimačių žemėlapių kūrimas. Skraidant yra neapsieinama be atsitrengimų į kitus objektus, aparatūros apgadinimų ar galimų žmonių sužeidimų. Siekiant to išvengti yra projektuojama kliūčių išvengimo sistema.*

*Šiame darbe yra apžvelgiami metodai, kuriais galima išvengti kliūtis ir kurie metodai labiausiai tinka kuriamai sistemai. Yra testuojami erdvinio vaizdo algoritmai: „Block match“, „Semi global block match“. Tinkamiausi algoritmai taikomi kuriamai sistemai.*

*Eksperimento metu yra testuojami „block match“ ir „semi global block match“ algoritmai naudojant bepilotį orlaivį ir testuojamas kliūčių išvengimas taikant abu algoritmus.*

Ivanovas, Adomas. Research of Obstacle Avoidance Algorithms for Unmanned Aerial Vehicles. Master's thesis in Informatics / supervisor assoc. prof. Armantas Ostreika. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: Information technology

Key words: *stereo vision, obstacle avoidance, unmanned aerial vehicles*

Kaunas, 2017. 45 p.

## SUMMARY

*Unmanned aerial vehicles are becoming very popular these days. They have a variety of applications: search and rescue missions, crop inspection, 3D mapping, surveillance and other military applications. Problem is that most UAV's do not have object avoidance systems installed, and that can lead to broken equipment or people may get injured. To avoid that, it is needed to design a system capable of avoiding obstacles.*

*In this paper, popular methods of object detection are reviewed and ones best suited for project are selected for further analysis. „Block match“ and „Semi global match“ methods are analysed and selected for application in object avoidance.*

*Experiment is performed by building custom UAV platform and testing selected stereo algorithms using drone on a real life objects.*

## Terminų ir santrumpų žodynas

<b>Mavlink</b>	Mavlink yra biblioteka skirta bendravimui tarp autopiloto ir kompiuterio. Ji naudojama Pixhawk ir Apm platformose [3].
<b>APM</b>	Autopiloto techninė įranga sukurta naudojant Arduino Mega platformą. Turi galimybę valdyti įvairaus tipo skraidykles bei robotus [5].
<b>Pixhawk</b>	Pixhawk yra pažangi autopiloto sistema, suprojektuota kaip atviro kodo projektas PX4 ir pagamintas 3D Robotics kompanijos. Autopilotas pasižymi pažangiais sensoriais, procesoriumi ST Microelectronics® bei Nuttix realaus laiko operacine sistema. Dėka šių technologijų autopilotas yra plačiai pritaikomas, bet kokiam autonominiam transportui [4].
<b>Tcas</b>	( <i>angl. Terminal Access Controller Access-Control System</i> ) Oro eismo pavojaus ir susidūrimų išvengimo sistema [18].
<b>Lidar</b>	( <i>angl. Light Detection And Ranging</i> ) Elektromagnetinių bangų naudojimas atstumo matavime [7].
<b>BM</b>	( <i>angl. Block match</i> ) Algoritmas gylio skaičiavimui iš stereo kadrų [25].
<b>SGBM</b>	( <i>angl. Semi global block match</i> ) Algoritmas gylio skaičiavimui iš stereo kadrų [12].
<b>SAD</b>	( <i>angl. Sum of Absolute Differences</i> ) Skirtumų absoliutinės sumos.
<b>SSD</b>	( <i>angl. Sum of Squared Differences</i> ) Skirtumų kvadratinės sumos.



## 1 ĮVADAS

Radijo bangomis valdomi orlaiviai yra sparčiai populiarėjanti platforma. Bepiločiai orlaiviai, dar kitaip vadinami „dronai“, turi įvairių taikymo sričių: aeronuotraukų fiksavimas, pasėlių fotografavimas, žvalgybos bei paieškos misijos, trimačių žemėlapių kūrimas ir mėgėjiškas skraidymas. Dauguma šiuolaikinių orlaivių turi vidinius kompiuterius, kurie sensorių, gps imtuvų pagalba padeda saugiai ir stabiliai valdyti skraidyklę. Šiuolaikiniai orlaiviai nors ir turi įrangą, kuri padeda valdyti orlaivį, tačiau tai ne visada juos apsaugo nuo susidūrimų su aplinkiniais objektais. Norint išspręsti šią problemą, būtina sukurti kliūtis išvengiančią įrangą.

**Projekto objektas:** Kliūčių išvengimo metodai.

**Projekto tikslas:** Ištestuoti kliūčių išvengimo algoritmus ir juos pritaikyti bepiločiam orlaiviui.

**Projekto uždaviniai:**

- Atlikti BM ir SGBM algoritmų greitaveikos testavimą;
- Sukurti programinę ir techninę įrangą realizuojančią kliūčių išvengimą;
- Nustatyti algoritmą, kurį naudojant objektai orlaivio skrydžio kelyje yra nustatomi tiksliausiai ir kuriuo algoritmu objektai nustatomi greičiausiai.

**Projekto metodai:** Literatūros analizė, mokslinių straipsnių analizė, projektavimas.

**Projekto rezultatas:** Atlikta lyginamoji analizė ir eksperimentinis kliūčių išvengimo algoritmų tyrimas.

## 2 KLIŪČIŲ IŠVENGIMO BEPILOČIAMS ORLAIVIAMS METODŲ APŽVALGA

### 2.1 Apžvalgos tikslas

Robotikoje yra naudojami įvairūs lokacijos metodai. Šie metodai yra realizuojami naudojant įvairius jutiklius. Jutiklis - tai įtaisas, jaučiantis temperatūrą, slėgį, judesį, šviesą, magnetinį lauką arba kurį nors kitą poveikį ir apie tai pranešantis elektriniu signalu [1]. Jutikliai yra skirstomi į šiuos tipus:

- Deformacijų ir slėgio jutikliai;
- Judesio jutikliai;
- Šviesos ir radiacijos jutikliai;
- Vaizdo jutikliai;
- Temperatūros jutikliai;
- Garso, infragarso ir ultragarso jutikliai;
- Kietojo kūno, skysčių ir dujų jutikliai;
- Aplinkos ir atmosferos jutikliai.

Objektų aptikimui bei kliūčių išvengimui yra naudojami vaizdo, radijo bangų, ultragarso bei šviesos sensoriai. Naudojant šiuos jutiklius galima įgyvendinti lokacijų metodus:

- Garsinė lokacija:
  - Pasyvi akustinė lokacija - ieškomas objekto skleidžiamas garsas;
  - Aktyvi akustinė lokacija - siųstuvas skleidžia garsinius signalus, kurie atsispindi nuo aplinkos objektų ir grįžta atgal į imtuvą. Pagal grįžusių signalų charakteristikas nustatoma aplinkos objektų būvimo vieta.
- Radiolokacija:
  - Pasyvi radiolokacija - apdorojamos radijo bangos, kurias skleidžia objektai;
  - Aktyvi radiolokacija - siųstuvas siunčia radijo bangų signalus, imtuvas priima atsispindėjusias bangas.
- Šviesinė lokacija:
  - Pasyvi šviesinė lokacija - erdvinis vaizdas (*angl. stereo vision*), optinis srautas (*angl. optical flow*);
  - Aktyvi šviesinė lokacija:
    - Naudojami regimos arba infraraudonos šviesos žibintai;
    - lazerio šviesos pluoštas - Lidar (*angl. Light Detection And Ranging*).

Visi objektų nustatymo būdai turi savo privalumų ir trūkumų, tad jų taikymas yra įmanomas tik tam tikrose srityse. Apžvalgos tikslas yra išanalizuoti metodų pliusus ir minusus, nustatyti metodus, tinkamus kuriamai kliūčių išvengimo platformai.

### 2.2 Objektų aptikimo metodai

#### 2.2.1 Radarai

Radaras (*angl. Radio detecting and ranging*) yra viena iš objektų aptikimo sistemų. Radaras naudoja radijo bangas tam, kad nustatytų objektų atstumą, kampą bei greitį. Ši sistema yra naudojama nustatant lėktuvus, laivus, kosminius laivus, raketas, motorizuotas transporto priemones, orų bei aplinkos informaciją. Radaras siunčia radijo bangas arba mikrobangas, kurios atsispindi nuo bet kokio objekto esančio bangos kelyje. Radaro imtuvas, kurio veikimo principas panašus į siųstuvo, gauna ir apdoroja atsispindėjusias bangas tam, kad nustatytų objekto parametrus.

Galia  $P_r$  gaunama iš imtuvo anteną yra apskaičiuojama pagal formulę (2.1):

$$P_r = \frac{P_t G_t A_r \sigma F^4}{(4\pi)^2 R_t^2 R_r^2}; \quad (2.1)$$

čia

$P_t$  - siųstuvo galia;

$G_t$  - siųstuvo antenos stiprinimas;

$A_r$  - efektyvus antenos atstumas;

$\sigma$  - radaro skersinis pjūvis;  
 $F$  - modelio didėjimo faktorius;  
 $R_t$  - atstumas nuo siųstuvo iki gavėjo;  
 $R_r$  - atstumas nuo gavėjo iki imtuvo.

Atstumo matavimui yra naudojamas įžambusis nuotolis [8]. Įžambusis nuotolis yra atstumas tarp radaro ir atsispindėjusio objekto. Plokštumoje atstumą tarp radaro ir objekto galima apskaičiuoti pagal formulę (2.2):

$$R = \frac{C_0 t}{2}; \quad (2.2)$$

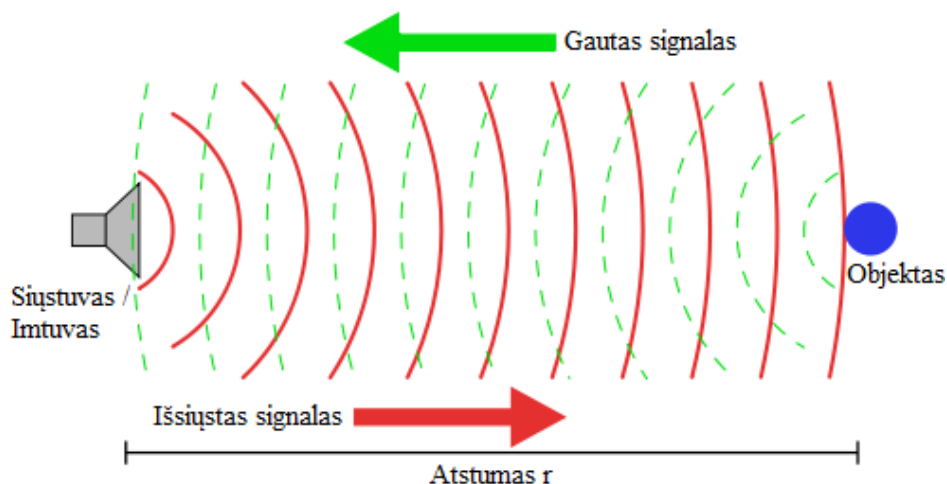
Čia  $C_0$  yra šviesos greitis,  $t$  - signalo keliavimo laikas,  $R$  – atstumas. Formulė  $R$  yra išreiškiama šiomis formulėmis (2.3), (2.4):

$$v = \frac{s}{t}; \quad (2.3)$$

Čia  $v$  - greitis,  $s$  - atstumas,  $t$  - laikas. Į formulę į greičio kintamąjį įstatome šviesos greitį, atstumą pakeičiami dvigubu atstumu tarp siųstuvo ir objekto.

$$c_0 = \frac{2R}{t}; \quad (2.4)$$

Čia  $C_0$  – šviesos greitis,  $R$  - atstumas iki objekto,  $t$  – signalo keliavimo laikas. Veikimo schema pavaizduota schemoje (2.1 pav.).

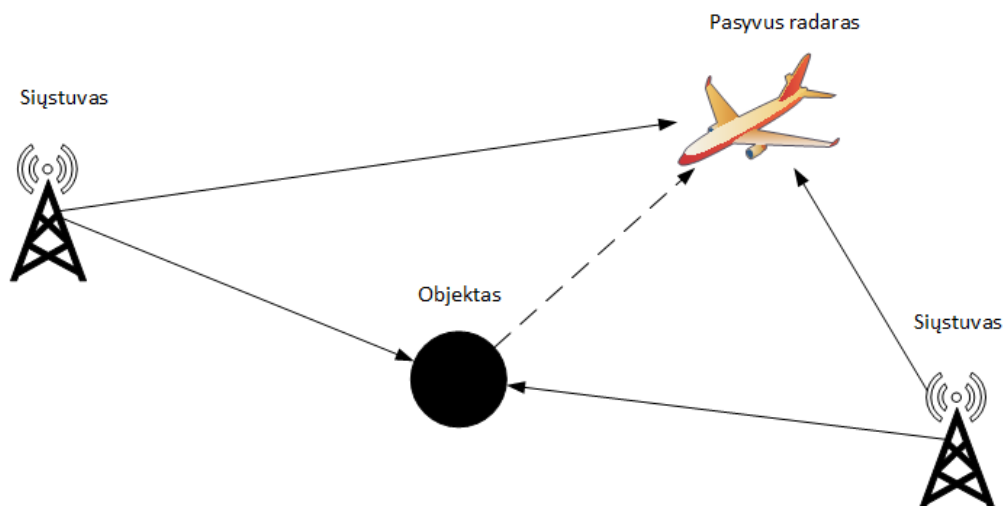


**2.1 pav.** Radaro veikimo schema

Radarai yra plačiai naudojami lėktuvuose ir kitose transporto priemonėse, tačiau nėra pritaikomi lengvuose bepiločiuose orlaiviuose. Radarai yra sunkūs, brangūs bei naudoja daug energijos, todėl kliūčių išvengimui reikia naudoti alternatyvius metodus.

### 2.2.2 Pasyvūs radarai

Pasyvaus radaro sistemos apjungia kelias radaro sistemas, kurios aptinka ir stebi objektus naudojant aplinkos spinduliavimą. Aplinkos spinduliavimas gali būti komercinės transliacijos ir komunikavimo signalai. Pasyvus radaras tai yra bistatinis radaras, kuris naudojasi kooperatyviais ir ne kooperatyviais radaro siųstuvais (2.2 pav.).



**2.2 pav.** Pasyvaus radaro veikimo schema

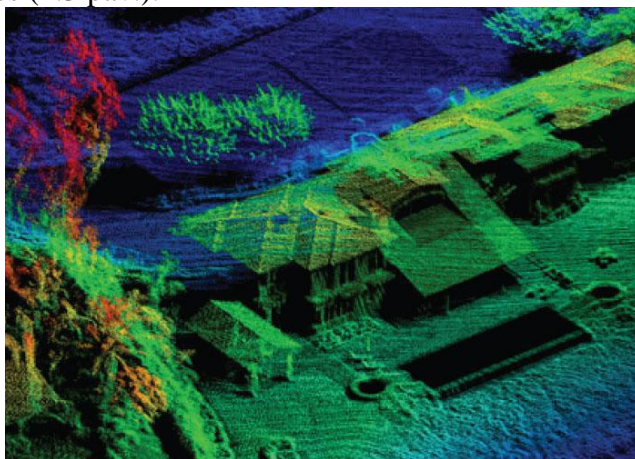
Paprastos radarų sistemos susideda iš siųstuvo ir imtuvo, kurios naudoja antenas išsiųsti ir gauti signalui. Laiko įvertinimas tarp išsiųsto ir nuo objekto atsispindėjusio signalo leidžia nustatyti atstumą iki objekto.

Pasyvaus radaro sistemose nėra siųstuvo. Vietoj to, imtuvas naudoja trečių šalių siųstuvus iš aplinkos ir matuoja laiko skirtumą tarp signalo, kuris atsispindėjo nuo objekto, ir signalo iš siųstuvo. Pasyvus radaras taip matuoja bistatinį „doppler shift“ aidą, pagal kurį yra nustatoma signalo kryptis. Tai leidžia nustatyti objekto būseną, kryptį bei greitį. Naudojant kelis siųstuvus ar imtuvus galima patikslinti sekimo tikslumą [19].

Pasyvūs radarai reikalauja daug skaičiuojamosios galios, todėl, kol kas, nėra pritaikomi bepiločiams orlaiviams.

### 2.2.3 Lidar

„Lidar“ yra atstumo matavimo metodas, kuriuo matuojamas atstumas nuo siuntėjo iki taikinio naudojant lazerio šviesą. „Lidar“ technologija naudojama generuoti aukštos raiškos žemėlapius, kurie taikomi geodezijos, archeologijos, geografijos, geologijos reikšmėms. Taip pat šia technologija galima atkurti trimates erdves (2.3 pav.).



**2.3 pav.** „Lidar“ technologija sugeneruotas 3d žemėlapis [13]

Objektai gali būti skenuojami aktyviuoju būdu, kai iš jutiklio yra siunčiami lazerio spinduliai, arba pasyviuoju, kai yra registruojamos nuo objekto atsispindėjusios saulės spindulių ar terminės radiacijos elektromagnetinės bangos. Skenavimas vyksta naudojant lazerinę sistemą, kuri generuoja lazerio spindulio impulsus. Impulso trukmė yra 7-15 ns.

Lazerinę skenavimo sistemą sudaro šie pagrindiniai komponentai [14]:

- Lazerio spindulio impulsų siųstuvas;
- Siųstuvo optinė sistema;
- Imtuvo optinė sistema;




- Fotodiodinis detektorius;
- Elektroninis duomenų apdorojimo ir išsaugojimo blokas.

Atstumo matavimo formulė yra pateikiama (2.5) formulėje:

$$D = \frac{ct}{2}; \quad (2.5)$$

čia D yra atstumas iki objekto, c – šviesos greitis, t – signalo keliavimo laikas. Atstumo matavimas šiuo metodu yra greitas ir tikslus, tačiau turi ir trūkumų. Pavyzdžiui, naudojant vienakrypčius lazerius, nėra įmanoma nustatyti objektų aplink siunčiamą spindulį. „Lidar“ jutikliai, jų privalumai ir trūkumai apžvelgiami lentelėje (2.1).

**2.1 lentelė** „Lidar“ jutikliai

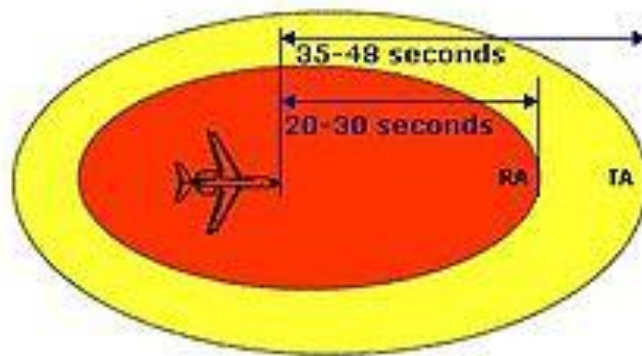
			
„Lidar“ tipas	Vienakryptis skenavimas	Skenavimas 360 laipsnių kampu	Erdvės atkūrimas
Pavadinimas	„LIDAR-Lite 3 Laser Rangefinder“ [16]	„RPLIDAR A2 360“ [15]	„Puck VLP-16“ [17]
Matavimo atstumas, m	40	6	100
Svoris, g	16	340	830
Kaina, Eur	149	449	7999
Trūkumai	Kryptinis skenavimas	Nedidelis skenavimo atstumas, ganėtinai brangus	Brangus, sunkus
Privalumai	Lengvas, pigus, didelis skenavimo atstumas	Lengvas, dvimatis skenavimas	Didelis skenavimo atstumas, trimatės erdvės atkūrimas

Pagal lentelėje pateiktus duomenis galima išskirti 3 sensorių tipus: vienakrypčius, dvimatės erdvės ir trimatės erdvės jutikliai. Vienakrypčiai sensoriai siunčia signalą viena kryptimi ir dažniausiai yra naudojami atstumui iki objekto nustatymui. Juos naudoja greičio matuokliai, žiūronai. Orlaiviai juos naudoja atstumo iki žemės nustatymui. Įmanomas matavimas ir horizontalia kryptimi, norint išvengti kliūties, tačiau tikimybė lazeriu pataikyti į objektą yra maža. Dvimatės erdvės skeneriai generuoja dvimatį aplinkos žemėlapi, sukdami lazerio prožektorių 360 laipsnių kampu. Juos naudoja automobiliai turintys autopiloto funkcija, aukščio matavimo prietaisai. Šis sensorius tinka kliūčių išvengimo orlaiviui, tačiau trūkumas, kad vaizdas projektuojamas tik dvimatėje erdvėje. Trimatės erdvės sensorius suteikia galimybę projektuoti trimatį erdvės žemėlapi, tačiau yra sunkus ir brangus, todėl reikia ieškoti alternatyvų.

„Lidar“ objektų aptikimo metodas yra priimtinas kliūčių išvengimo realizacijai naudojant dvimatės erdvės skenavimo „lidar“ sensorių, tačiau jo svoris ir kaina yra per didelė siekiamam orlaivio sprendimui.

#### 2.2.4 Radijo atsakikliai

TCAS (*angl. Traffic collision avoidance system*) yra elektroninis įrenginys, kuris generuoja atsakymą, kai gauna radijo signalo užklausą. Orlaiviai naudoja atsakiklius, tam, kad juos galėtų identifikuoti oro eismo kontrolė. Taip pat atsakiklių pagalba veikia susidūrimų išvengimo sistema, kuri leistų išvengti susidūrimų su kitais orlaiviais. Radijo atsakikliai leidžia išvengti susidūrimų, tačiau juos naudoja ne visi orlaiviai. Oro erdvė skirstoma į zonas, kuriose yra stebimos kitų TCAS siųstuvų lokacijos (2.4 pav.). Orlaiviui įskridus į pavojaus zoną yra generuojamas TCAS II signalas.



2.4 pav. TCAS II oro zona

TCAS II signalai skirstomi į [18]:

- TA – patariamieji signalai;
- RA – veiksmų, kurių reikia imtis signalai;
- CC – įspėjimų pabaigos signalai.

Gavus TA signalą, pilotas privalo apžiūrėti aplinką ir įsitikinti kas sukelia TA signalą. Maži kurso pakeitimai yra galimi esant TA pranešimui. Pasirodžius RA signalui, pilotas privalo nedelsiant imtis instrukcijoje nustatytų veiksmų. Pavyzdžiui, „Climb; Climb“ signalo metu pilotas privalo kilti į viršų. Visos signalų reikšmės pateikiamos lentelėje (2.2 lent.).

2.2 lentelė TCAS signalai

Tipas	Tekstas	Reikšmė	Veiksmai
TA	Traffic; traffic.	Kliūtis yra šalia	Būti pasirengusiam, jeigu atsiras RA pranešimas
RA	Climb; climb.	Kliūtis praskris žemiau	Pradėti kilti
RA	Descend. Descend.	Kliūtis praskris virš	Pradėti leistis
RA	Increase climb.	Kliūtis praskris po	Pradėti kilti
RA	Increase descent.	Kliūtis praskris virš	Pradėti leistis
RA	Reduce climb.	Kliūtis tikriausiai yra gerokai žemiau	Kilti lėčiau
RA	Reduce descent.	Kliūtis tikriausiai yra gerokai virš	Leistis lėčiau
RA	Climb; climb now.	Kliūtis, kuri buvo virš, praskris po	Iš leidimosi pereiti į kylimą
RA	Descend; descend now.	Kliūtis, kuri buvo po, praskris virš	Iš kylimo pereiti į leidimąsi
RA	Maintain vertical speed; maintain.	Kliūtis bus išvengta jeigu bus laikomasi vertikalios greičio	Laikytis vertikalios greičio
RA	Level off, level off.	Kliūtis tikriausiai jau praskrido	Išlyginti poziciją
RA	Monitor vertical speed.	Kliūtis virš arba po	Likti lygioje pozicijoje
RA	Crossing.	Praskrendama pro kliūtis aukštį	Tęsti atsižvelgiant į RA pranešimą
CC	Clear of conflict.	Kliūtis grėsmės nebekelia	Grįžti į pradinę būseną

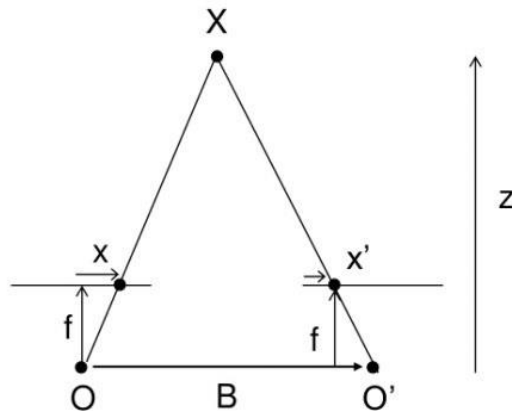
Nors ši sistema ir padeda išvengti susidūrimų su kitais objektais, turinčiais TCAS siųstuvą-imtuvą, tačiau šiuo metu naudojamos sistemos yra sunkios ir brangios. Taip pat jos naudoja daug energijos bei aptinka tik objektus, turinčius šią sistemą, todėl technologijos pritaikymas kuriamai platformai yra neįmanomas.



## 2.2.5 Optiniai objektų aptikimo metodai

### 2.2.5.1 „Block match“ algoritmas

Blokų lyginimo algoritmas yra būdas surasti atitinkančius makroblokus skaitmeniniuose paveiksluose. Šiuo būdu galima sukurti erdvinio vaizdo ir taškų debesis. Erdvinio vaizdo kūrimui yra naudojamos dvi kameros nukreiptos viena kryptimi paraleliai viena kitos (2.5 pav.).



2.5 pav. Erdvinio vaizdo generavimo diagrama

Erdvinis vaizdas gaunamas pagal formulę (2.6):

$$D = x - x' = \frac{Bf}{Z}; \quad (2.6)$$

čia

$D$  yra erdvinis vaizdas;

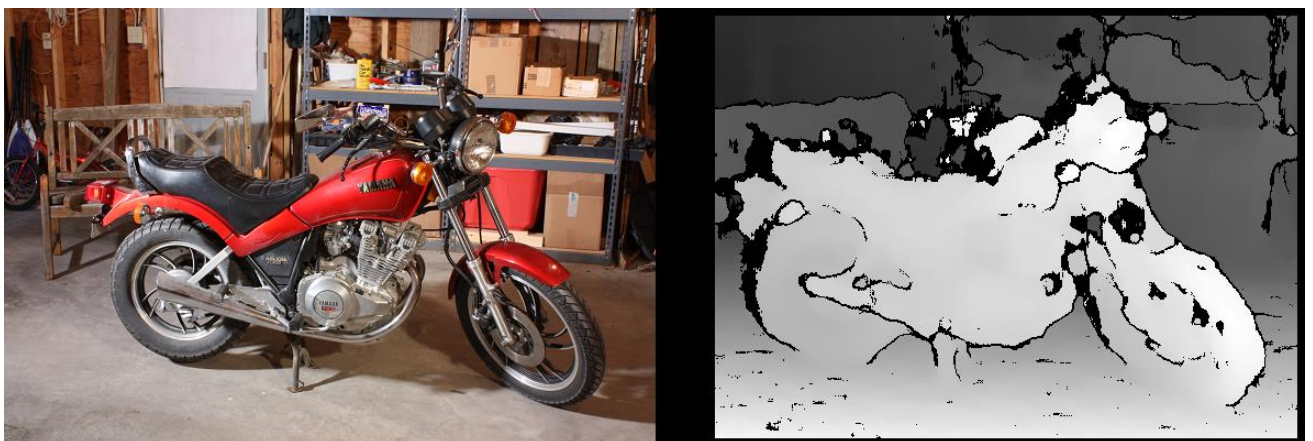
$x, x'$  - skirtumas tarp taškų paveiksluose;

$B$  – skirtumas tarp 2 vaizdo kameros;

$Z$  – gylis;

$f$  – židinio nuotolis.

Pagal formulę galima teigti, kad pikselių gylis priklauso nuo atstumo tarp tų pačių pikselių skirtingų kamerų nuotraukose. Šiuo metodu galima gauti erdvinį vaizdą (2.6 pav.).



2.6 pav. „Block match“ algoritmu sugeneruotas erdvinis vaizdas

Sugeneruotas erdvinis vaizdas išskaido sceną į skirtingo gylio zonas, kuriose šviesi spalva atspindi zonas, kurios yra arti, o tamsi – zonas, kurios toli. Blokų palyginimo metodą realizuojant pasitelkus aukšto dažnio erdvinio vaizdo kameras ir spartų kompiuterį galima sukurti aukšto tikslumo kliūčių išvengimo sistemą.

### 2.2.5.2 „Semi global block match“ algoritmas

SGBM metodas yra aprašytas autoriaus Hirschmüller [12]. Šis metodas apjungia globalius ir lokalius erdvinio vaizdo metodus. Algoritmas skaičiavimams naudoja paveikslų poras su žinomomis vidinėmis ir išorinėmis objektų orientacijomis. Algoritmo skaičiavimui yra naudojama  $E(D)$  formulė (2.7):

$$E(D) = \sum_p (C(p, D_p) + \sum_{q \in N_p} P_1 I[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 I[|D_p - D_q| > 1]); \quad (2.7)$$

čia

$E(D)$  - erdvinio vaizdo energija;

$q$  ir  $p$  nurodo paveikslo pikselių indeksus;

$N_p$  yra artimiausias pikselio  $p$  kaimynas;

$C(p, D_p)$  nurodo pikselio tapatumo reikšmę su gylio reikšme  $D_p$ ;

$P_1$  yra koeficientas nurodantis pikselių pasikeitimus tarp kaimynų, kurie lygūs vienas;

$P_2$  yra koeficientas nurodantis pikselių pasikeitimus tarp kaimynų, kurie yra daugiau nei vienas;

$I$  yra funkcija kuri grąžina vienetą jeigu argumentas yra teisingas arba nulį, jeigu yra priešingai.

Minimizuota funkcija generuoja erdvinį vaizdą (2.7 pav.)



**2.7 pav.** „Semi global block match“ algoritmu sugeneruotas erdvinis vaizdas

Gautas erdvinis vaizdas yra tikslesnis nei naudojant BM metodą, tačiau algoritmo skaičiavimo laikas yra ilgesnis. Šį algoritmą galima pritaikyti kuriamai kliūčių išvengimo sistemai.

### 2.3 Apžvalgos apibendrinimas

- Išanalizuoti pagrindiniai objektų atpažinimo metodai: radarai, pasyvūs radarai, lidar, TCAS transponderiai, optiniai atpažinimo algoritmai – „block match“ ir „semi global block match“.
- Eksperimentui pasirinkta naudoti optinius objektų atpažinimo algoritmus: „block match“ ir „semi global block match“, nes naudojant metodus yra įmanomas greitas ir tikslus objektų atpažinimas.



### 3 BEPILOČIŲ ORLAIVIŲ ĮRANGOS APŽVALGA

#### 3.1 Apžvalgos tikslas

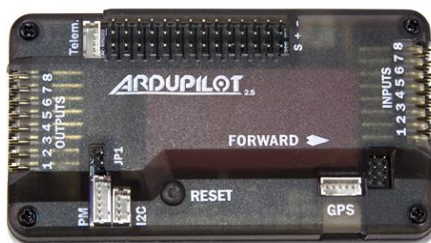
Rinkoje egzistuoja daug bepiločių orlaivių techninės bei programinės įrangos. Bepiločiai orlaiviai naudoja autonominio skrydžio valdiklius, kurie apjungia orlaivio sistemas į vieną bendrą sistemą ir valdo viską iš valdiklio. Autopilotų pasirinkimas yra gana didelis, tad būtina pasirinkti tinkamiausią variantą.

Apžvalgos tikslas yra apžvelgti orlaiviuose naudojamąsias sistemas ir pasirinkti tinkamą įrangą eksperimento realizavimui.

#### 3.2 Techninė įranga

##### 3.2.1 Autonominio skrydžio valdikliai

„Ardupilot Mega“ arba APM yra profesionali autopiloto sistema, paremta „Arduino Mega“ platforma [5]. Šis autopilotas gali valdyti lėktuvų modelius, sraigtasparnius, multirotorius. Tai pilnas autopilotas turintis autonominio stabilizavimo galimybes, gps navigaciją, telemetriją. Sistema palaiko 8 RC kanalus su keturiomis serial jungtimis. APM susideda iš pagrindinio procesoriaus ir IMU skydo (3.1 pav.).



3.1 pav. APM autonominio skrydžio valdiklis

„Pixhawk“ yra pažangi autopiloto sistema, suprojektuota kaip atviro kodo projektas PX4 ir pagamintas 3D Robotics kompanijos. Autopilotas pasižymi pažangiais sensoriais ir procesoriumi „ST Microelectronics“, bei „Nuttix“ realaus laiko operacine sistema. Dėka šių technologijų autopilotas yra plačiai pritaikomas, bet kokiam autonominiam transportui.

„Pixhawk“ sistemos privalumai susideda iš integruotos sąsajos, programavimo aplinka panaši į Unix/Linux. Visiškai naujos autopiloto funkcijos, tokios kaip sudėtingi misijų scenarijai, skrydžio režimai, PX4 vadovo pagalba visi procesai vyksta sklandžiai ir greitai. Prie Pixhawk modulio galima prijungti daug naujų periferinių modulių, tokių kaip: skaitmeninis greičio sensorius, spalvotas LED indikatorius, išorinis magnetometras (3.2 pav.). „Pixhawk“ palaiko PPM jungtį, o tai reiškia, kad RC imtuvas su autopilotu sujungiami vienu laidu per imtuvo PPM jungtį. Tokia jungtis yra „Futaba S.Bus“, bei „Spektrum“ pajungimo jungtys. Jeigu RC imtuvas neturi tokios jungties pajungimas vykdomas per PPM daviklį [4].



3.2 pav. „Pixhawk“ autonominio skrydžio valdiklis

APM ir „Pixhawk“ valdikliai yra plačiai naudojami orlaiviuose. Abu autopilotai turi savo pliusų ir minusų. Charakteristikos apžvelgiamos lentelėje (3.1 lent.).

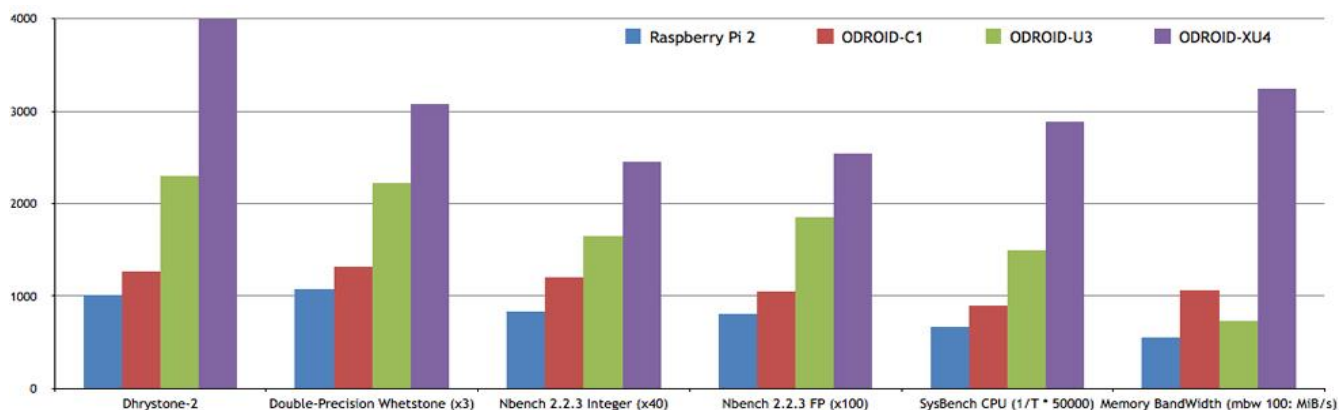
### 3.1 lentelė Autonominio skrydžio valdiklių palyginimas

Charakteristikos	APM	Pixhawk	Komentaras
Dydis	Mažas	Didelis	APM - 35x35x5 mm. Pixhawk - 81x50x16 mm
Kaina	Maža	Vidutinė	APM ~50 eur. Pixhawk ~100 eur.
Kokybė	Kintanti	Gera	APM kontaktų prasta kokybė, gali tekti persiluituoti kontaktus.
Palaikomumas	Geras	Geras	
Skrydžio stabilumas	Geras	Geras	
Tikslumas	Geras	Geresnis	Pixhawk turi galingesnę procesorių ir daugiau atminties.
Indikatoriai	Mažai	Daug	Pixhawk turi daug LED indikatorių, APM keletą.
Diegimo sunkumas	Vidutinis	Vidutinis	
Galimybė skristi autonominiu būdu	Yra	Yra	
Mission planner PĮ stabilumas	Geras	Geras	
Skrydžio registrų kūrimas	Geras	Geresnis	Pixhawk autopilote log'us galima įrašyti į SD kortelę.
Atnaujinamumas	Blogas	Geras	APM sistema nebeatnaujinama, Pixhawk valdikliui dažnai išleidžiamos naujos versijos.
Derinimo sunkumas	Geras	vidutinis	

Apžvelgus APM bei „Pixhawk“ valdiklius nuspręsta eksperimentui naudoti Pixhawk autopilotą. Palyginus su APM, šis autopilotas turi greitesnę procesorių, tad bus įmanoma greitai apdoroti informaciją ir komunikuoti su skrydžio kompiuteriu. Taip pat, „Pixhawk“ turi galimybę registrus kaupti atminties kortelėje ir po to juos atskirai analizuoti kompiuteryje.

### 3.2.2 Skrydžio kompiuteris

Skrydžio kompiuteris reikalingas vaizdo medžiagos apdorojimui. Kompiuterių yra įvairių rūšių, tačiau reikia išsirinkti tinkamiausią erdvinio vaizdo apdorojimui. Palyginimui naudojami 4 kompiuteriai „Raspberry PI 2“, „Odroid C1“, „Odroid U3“, „Odroid XU4“. Kompiuteriai testuoti naudojant šias testavimo sistemas: „Dhrystone-2“, „Double-Precision Whetstone“, „Nbench 2.2.3 Integer“, „Nbench 2.2.3 FP“, „SysBench CPU“, „Memory BandWidth“. Testų rezultatai pateikiami grafiškai (3.3 pav.)[9].



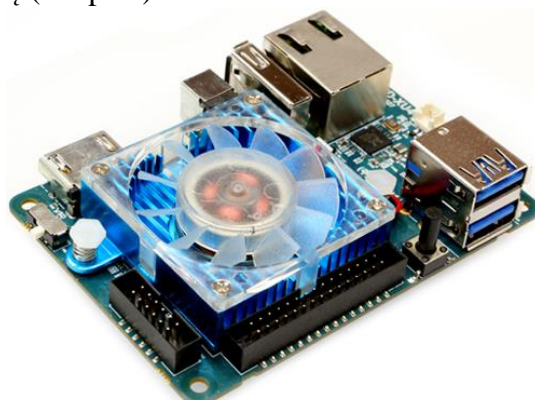
3.3 pav. Skrydžio kompiuterių testų rezultatai [9]

Pagal testavimo rezultatus (3.2 lent.) „Odroid XU4“ kompiuteris yra 4 kartais greitesnis nei „Raspberry Pi 2“ kompiuteris, tris kartus greitesnis nei „Odroid C1“ ir du kartus greitesnis nei „Odroid U3“ kompiuteris.

**3.2 lentelė** Skrydžio kompiuterių testų rezultatai [9]

	<b>Raspberry Pi 2</b>	<b>ODROID C1</b>	<b>ODROID U3</b>	<b>ODROID XU4</b>
Dhrystone 2	1006.6	1262.8	2300.7	3994.1
Double-Precision Whetstone	1076.1	1318.8	2217.9	3074.7
Nbench 2.2.3 Integer	840.0	1208.0	1648.0	2451.1
Nbench 2.2.3 FP	809.0	1050.0	1860.0	2536.9
SysBench CPU	669.3	902.2	1497.0	2891.8
Memory BandWidth	557.1	1069.6	736.6	3243.8

XU4 įgyja pranašumą, nes naudoja du keturių branduolių ARM procesorius (A15 ir A7). Taip pat turi 2GB DDR3 RAM atmintį (3.4 pav.).



**3.4 pav.** „Odroid XU4“ kompiuteris

Taip pat didelis privalumas yra USB 3.0 jungtis, kuri turi 5 kartus didesnę duomenų pralaidumą nei USB 2.0 jungtis. Pagal testų rezultatus nuspręsta pasirinkti „Odroid XU4“ skrydžio kompiuterį.

### 3.2.3 Erdvinio vaizdo kamera

Kliūčių išvengimui reikalinga erdvinio vaizdo kamera, kuri sugebėtų atnaujinti vaizdą 120 Hz dažniu, nes kliūčių išvengimui reikalingas greitas kadrų atnaujinimas. Taip pat svarbu, kad vaizdo kameros būtų visada paraleliai viena kitos, tam kad erdvinis vaizdas būtų kuo mažiau iškreiptas.

Erdvinio vaizdo kamerą galima imituoti dvejomis web kameromis, prijungtomis paraleliai viena kitos. Šis sprendimas tiktų, jeigu kameros visą laiką būtų stacionarioje pozicijoje. Tačiau montuojant kameras bepiločiame lėktuve, sunku jas būtų išlaikyti horizontalioje pozicijoje. Šiam atvejui pasirinkta naudoti erdvinio vaizdo kamerą, kurioje kameros yra įmontuotos viename rėme. Šioje dalyje lyginamos dvi erdvinės vaizdo kameros: „Playstation 4 eye“ ir „ZED“ kameros. Jų palyginimas pateikiamas lentelėje (3.3 lent.).

**3.3 lentelė** Erdvinio vaizdo kamerų palyginimas

	<b>Playstation 4 eye</b>	<b>ZED [20]</b>
Lango dydis	1280×800 pikseliai, 60 Hz 640×400 pikseliai, 120 Hz 320×192 pikseliai, 240 Hz	4416x1242 pikseliai, 15 Hz 3840x1080 pikseliai, 30 Hz 2560x720 pikseliai, 60 Hz 1344x376 pikseliai, 100 Hz
Matymo laukas	85°	110°
Atstumas tarp lęšių, mm	80	120
Jungtis	AUX	USB 3.0
Svoris, g	183	159
Kaina, Eur	50	409

Pagal lentelėje pateiktus duomenis, „ZED“ kamera turi galimybę filmuoti aukštos raiškos vaizdus. Tačiau aukštos raiškos vaizdai apdorojami reikalingas galingas kompiuteris, tad raiška tinka ir mažesnė. „ZED“ kameros matomumo laukas yra didesnis, tačiau nežymiau. Atstumas tarp kamerų ilgesnis, ko pasekoje įmanoma išgauti gilesnį erdvinį vaizdą. „Playstation 4“ kameros svoris didesnis, nes ji nėra pritaikyta skraidyklėms (3.5 pav.). Jungtis tinkama tik Playstation 4 kompiuteriui.



**3.5 pav.** „Playstation 4“ kamera

Vienintelis „ZED“ kameros minusas tai kaina, kuri yra ženkliai didesnė, ko pasekoje pasirinkta naudoti playstation 4 kamerą. Ji atitinka erdvinio vaizdo išgavimo lūkesčius bei yra pritaikoma kuriamai kliūčių išvengimo platformai.

### **3.3 Programinė įranga**

Programinę įrangą, kaip ir techninę, būtina teisingai pasirinkti, kad visi techniniai įrenginiai veiktų tinkamai bei optimaliai. Tinkamas pasirinkimas leidžia tiksliau atlikti eksperimentinę projekto dalį.

„Odroid XU4“ kompiuteryje pasirinkta diegti „Ubuntu server 14.04“ operacinę sistemą, nes naudojant šią operacinę sistemą erdvinio vaizdo kamera veikia optimaliausiai. Vaizdų apdorojimui naudojama „opencv“ biblioteka, kuri susideda iš kompiuterinei regai orientuotų skaičiavimo funkcijų.

Kompiuteris bei autopilotas apjungiamas „Dronekit“ programine įranga. „Dronekit“ suteikia galimybę kurti aplikacijas, kurios apjungia skrydžio valdymo valdiklius ir skrydžio kompiuterius. Programinė įranga komunikuoja naudodajama „MAVLink“ protokolą, kuris suteikia galimybę gauti skraidyklės telemetrijos duomenis, parametrus bei suteikia galimybę perimti orlaivio valdymą. „Dronekit“ įrangą galima pritaikyti kompiuterinės regos, 3d modeliavimo, skrydžio planavimo aplikacijoms.

### **3.4 Apžvalgos apibendrinimas**

Apžvelgus naudojamą techninę įrangą ir atlikus analizę, nuspręsta eksperimento realizaciją atlikti naudojant „Odroid XU4“ kompiuterį, „Pixhawk“ skrydžio valdiklį, „Playstation 4“ kamerą. „Odroid“ kompiuteris pasirinktas, pagal testų rezultatus nustatčius, kad jis yra galingesnis už tos pačios klasės kompiuterius. „Pixhawk“ skrydžio valdiklis, palyginus su APM, turi greitesnį procesorių, tad greitam informacijos apdorojimui pasirinkta jį naudoti kliūčių išvengime. „Playstation 4“ kamera pasirinkta, nes ji palaiko aukšto dažnio vaizdų atnaujinimą bei jos kaina yra maža, palyginus su kitomis erdvinio vaizdo kameromis. Pasirinkta naudoti „Ubuntu server 14.04“, „opencv“, „dronekit“ programinę įrangą.

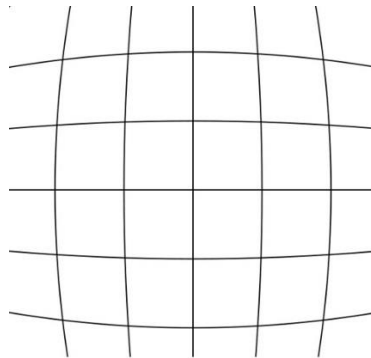
## 4 ERDVINIO VAIZDO ALGORITMŲ ANALIZĖ

### 4.1 Analizės tikslas

Erdvinio vaizdo kūrimui egzistuoja daug algoritmų. Šiame skyriuje analizuojamas „block match“ bei „semi global block match“ algoritmų veikimas. Apžvelgiami visi žingsniai reikalingi teisingo erdvinio vaizdo sudarymui bei ištestuojama algoritmų greitaveika taikant paveikslų poras. Analizės tikslas yra surasti efektyviausią algoritmą kliūčių išvengimo taikymui.

### 4.2 Erdvinio vaizdo kameros kalibravimas

Norint sugeneruoti neiškreiptą erdvinį vaizdą, būtina atlikti kameros kalibravimą [22]. Šiuolaikinės „pinhole“ kameros, dėl lęšiuko formos iškraipo fiksuojama vaizdą. Filmuojant tiesios linijos atrodo išlenktos. Išlenkimas daugiausiai vyksta toliau nuo vaizdo centro. Paveiksle pavaizduotas linijų spindulinis iškraipymas (pav. 4.1)



4.1 pav. Kameros vaizdo iškraipymas

Šis iškraipymas yra pašalinamas taikant formulę (4.1):

$$x_c = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6); \quad (4.1)$$

čia  $x_c$  koordinatė  $x$  be iškraipymo,  $x$  – iškraipyto vaizdo koordinatė,  $k_1 k_2 k_3$  yra spindulinio iškraipymo koeficientai. Taip pat ši formulė taikoma ir koordinatėms  $y$  (4.2):

$$y_c = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (4.2)$$

Taip pat egzistuoja ir tangentinis iškraipymas, kuris atsiranda kai lęšis nėra sulygiuotas paraleliai su vaizdavimo plokštuma. Taigi kai kurios vietos gali atrodyti arčiau nei yra iš tikrųjų. Ši problema yra sprendžiama taikant formulę (4.3):

$$x_c = x + [2p_1 xy + p_2(r^2 + 2x^2)]; \quad (4.3)$$

čia  $x_c$  koordinatė  $x$  be iškraipymo,  $x, y$  – iškraipyto vaizdo koordinatė,  $p_1 p_2$  yra tangentinio iškraipymo koeficientai. Toks pat metodas taikomas ir  $y$  koordinatėms (4.4):

$$y_c = y + [p_1(r^2 + 2y^2) + 2p_2 xy] \quad (4.4)$$

Taigi pagal apžvelgtas formules, tam, kad panaikinti vaizdo iškraipymą yra reikalingi penki iškraipymo koeficientai (4.5)

$$K = (k_1 \ k_2 \ p_1 \ p_2 \ k_3) \quad (4.5)$$

Tam, kad pilnai atlikti iškraipymo panaikinimą reikalingi dar keli parametrai. Kalibravimui yra reikalingi išoriniai ir vidiniai kameros parametrai. Vidiniai parametrai kiekvienai kamerai yra skirtingi. Vidinius parametrus sudaro židinio nuotolio bei optinio centro informacija. Kitaip šie parametrai vadinami kameros matrica. Šie parametrai yra unikalūs kiekvienai kamerai, tad apskaičiavus šią matricą, ją galima naudoti ir kitoms formulėms apskaičiuoti. Kameros matrica yra užrašoma pagal formulę (4.6):

$$\text{kameros matrica} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}; \quad (4.6)$$

Čia  $f_x, f_y$  yra židinio nuotolis, o  $c_x, c_y$  yra optinis centras.

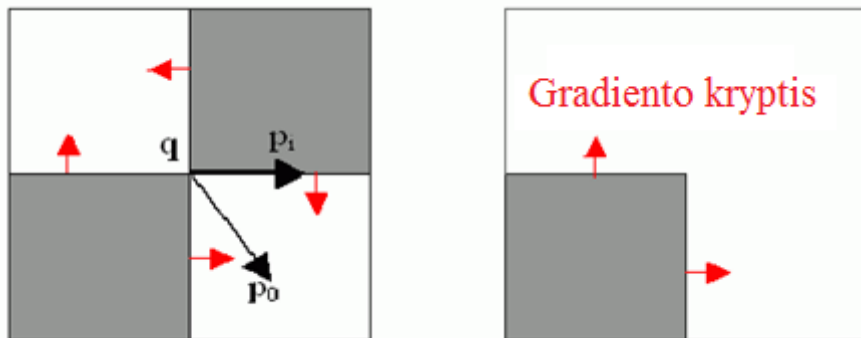
Parametrų gavimui yra naudojama šachmatų lenta (4.2 pav.). Naudojant paveikslą su šachmatų lenta galima nustatyti kameros iškraipymą. Paveiksle, lentos kraštai bei kvadratų susijungimo taškai yra atvaizduojami koordinatėmis, pagal kurias galima nustatyti iškraipymo lygį.



**4.2 pav.** Kalibravimui naudojama šachmatų lenta

Kalibravimui yra naudojama OpenCV biblioteka. Šachmatų lenta yra naudojama su devyniais kvadratų stulpeliais ir septyniomis kvadratų eilėmis. Kvadrato dydis yra 11 milimetrų, o pačios lentos dydis yra 100 x 77 centimetrai. Kameros kalibravimui reikalinga išgauti trimačius lentos taškus. Tai yra atliekama fiksuojant lentą įvairiais kampais ir pasukimais.

Šachmatų lentos taškų radimui yra naudojama `cv2.findChessboardCorners()` funkcija. Šiai funkcijai reikia nurodyti, kokio šablono šachmatų lenta yra fiksuojama (8 x 8, 7 x 6, 8 x 6 tinklas). Šie skaičiai nurodo šachmatų lentos taškų skaičių tarp baltų ir juodų kvadratų. Šiuo atveju yra naudojamas 8 x 6 tinklas. Funkcija grąžina *retval* reikšmę, kurioje nurodyta ar yra aptiktas šachmatų lentos šablonas. Kai yra surandamas šablonas, galima patikslinti taškų tikslumą naudojant `cv2.cornerSubPix()` funkciją (4.3 pav.).



**4.3 pav.** „CornerSubPix“ funkcijos veikimo diagrama

`cv2.cornerSubPix()` funkcija veikia tokiu principu, kad kiekvienas vektorius nuo centro  $q$  iki taško  $p$ ,  $q$  paieškos lango ribose, yra statmenas paveikslo gradientui (4.7).

$$M_i = DI_{pi}^T \cdot (q - p_i); \quad (4.7)$$

čia  $DI_{pi}$  yra paveikslo gradientas viename iš taškų  $p_i$  esančiamie šalia  $q$  taško.  $q$  reikšmė yra randama, tam, kad  $M_i$  funkcija galėtų būti minimizuota. Funkcijų sistema atvaizduojama pagal formulę (4.8):

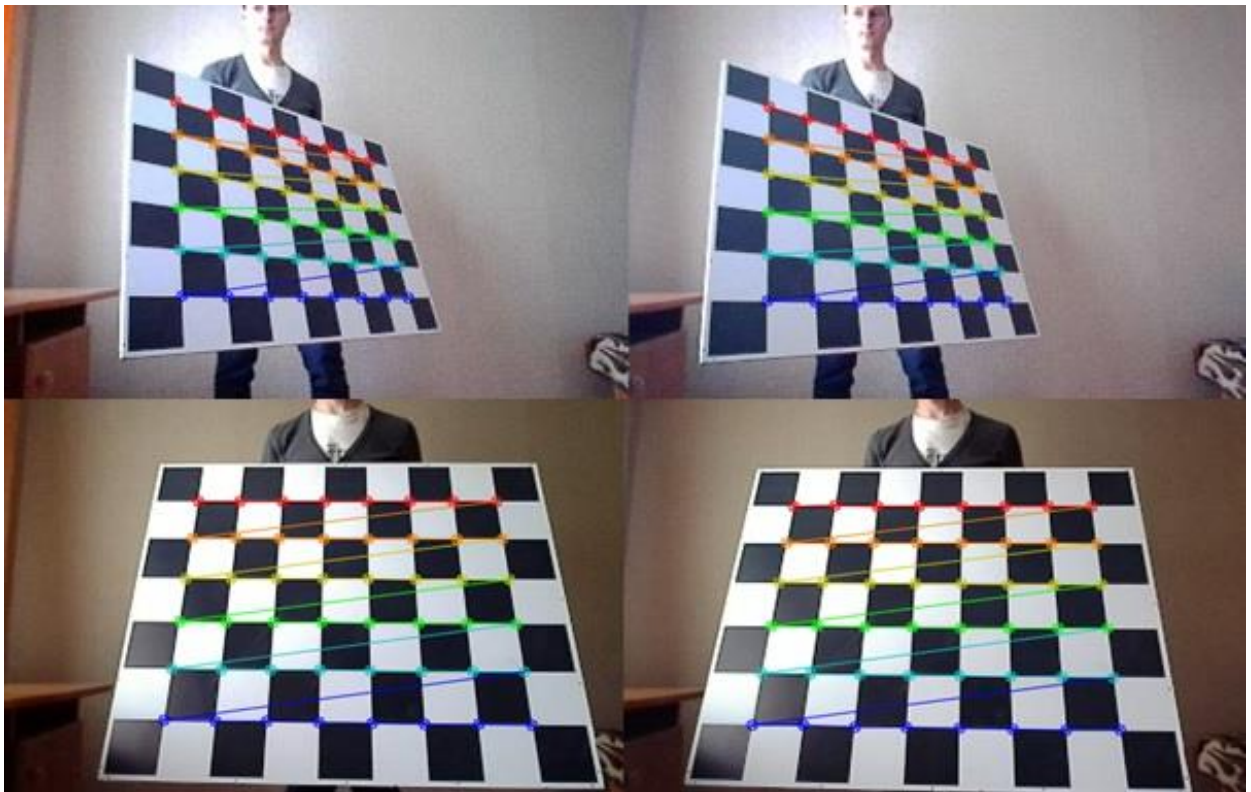


$$\sum_i (DI_{pi} \cdot DI_{pi}^T) - \sum_i (DI_{pi} \cdot DI_{pi}^T \cdot p_i); \quad (4.8)$$

Čia gradientai sumuojami  $q$  taško skenavimo lange. Kviečiant pirmąją gradiento reikšmę  $G$  ir antrąją –  $b$  yra gaunama pastaroji formulė (4.9):

$$q = G^{-1} \cdot b; \quad (4.9)$$

Algoritmas nustato naują skenavimo centrą taške  $q$  ir yra vykdomos iteracijos kol centras yra pasiekiamas pagal nustatytą slenksčio dydį. Naudojant `cv2.drawChessboardCorners()` funkciją yra atvaizduojami aptikti lentos taškai (4.3 pav.).



**4.4 pav.** Šachmatų lentos aptikimas

Gavus trimačius bei dvimačius erdvės taškus galima atlikti kameros kalibravimą. Kalibravimui naudojama `cv2.calibrateCamera()` funkcija. Šiai funkcijai yra paduodami trimačiai bei dvimačiai taškai. Funkcija grąžina kameros matricą, iškraipymo koeficientus, sukimo bei judesio vektorius.

Funkcija kairei kamrai apskaičiuo kameros matricą (4.10) ir iškraipymo koeficientus (4.11):

$$kamos\ matrica = \begin{bmatrix} 210.09259171 & 0. & 170.33259654 \\ 0. & 209.99393091 & 102.59767386 \\ 0. & 0. & 1. \end{bmatrix} \quad (4.10)$$

$$koeficientai = [-0.03122971 \ 0.05584016 \ -0.00173277 \ 0.0078608 \ -0.04493766] \quad (4.11)$$

Dešinei kamrai apskaičiuo kameros matricą (4.12) ir iškraipymo koeficientus (4.13):

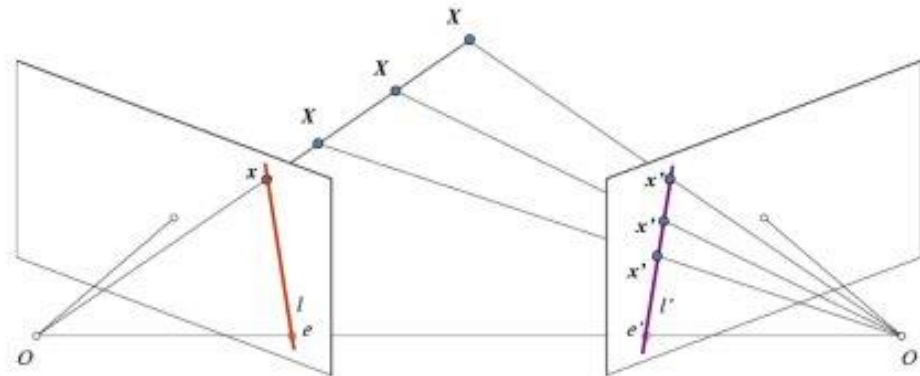
$$kamos\ matrica = \begin{bmatrix} 210.70240033 & 0. & 169.33413534 \\ 0. & 210.61564403 & 103.76542068 \\ 0. & 0. & 1. \end{bmatrix} \quad (4.12)$$

$$koeficientai = [-0.01592789 \ 0.01387636 \ -0.00057269 \ 0.006048 \ -0.00785658] \quad (4.13)$$

Pagal gautus koeficientus galima atlikti erdvinio vaizdo kameros porų kalibravimą. Šiuo metodu yra nustatoma transformacija tarp dviejų kamerų. Šis kalibravimas atliekamas naudojant `cv2.stereoCalibrate()` funkciją. Šiai funkcijai siunčiamos karės ir dešinės kameros trimatės ir

dvimatės šachmatų lentos koordinatės. Abiejų kamerų matricos bei iškraipymo koeficientai. Ši funkcija grąžina naujus kairės ir dešinės kameros koeficientus ir matricas.

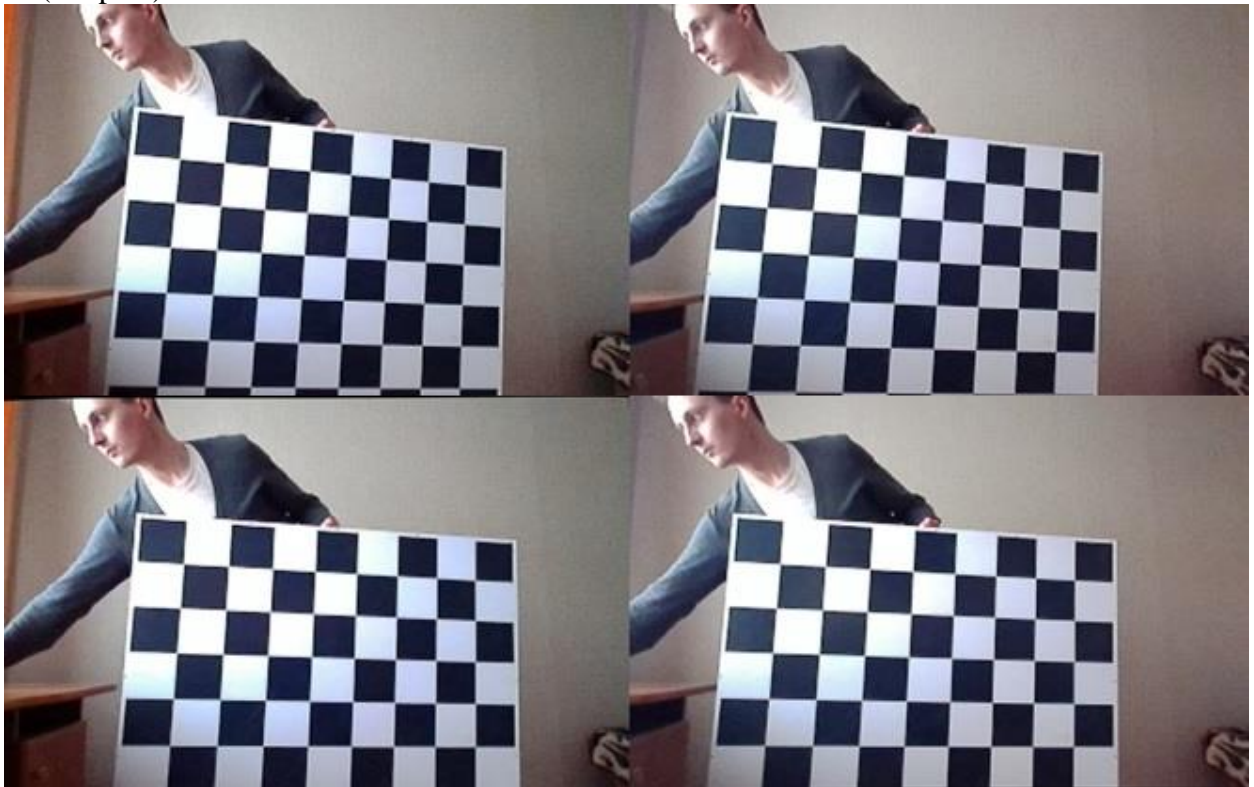
Naudojant funkciją `cv2.stereoRectify()` abiejų kamerų vaizdas yra deformuojamas taip, kad kamerų epipolinės linijos atitiktų viena kitą (4.3 pav.).



**4.5 pav.** Epipolinės linijos

Funkcija grąžina deformavimo bei projekcijos matricas abejoms kameroms.

Kitas žingsnis yra apskaičiuoti kamerų transformavimo reikšmes naudojant funkciją `initUndistortRectifyMap()`. Ši funkcija sukuria matricą, su kuria yra gaunamas galutinis kameros vaizdas (4.4 pav.).



**4.6 pav.** Sukalibruota kamerų pora

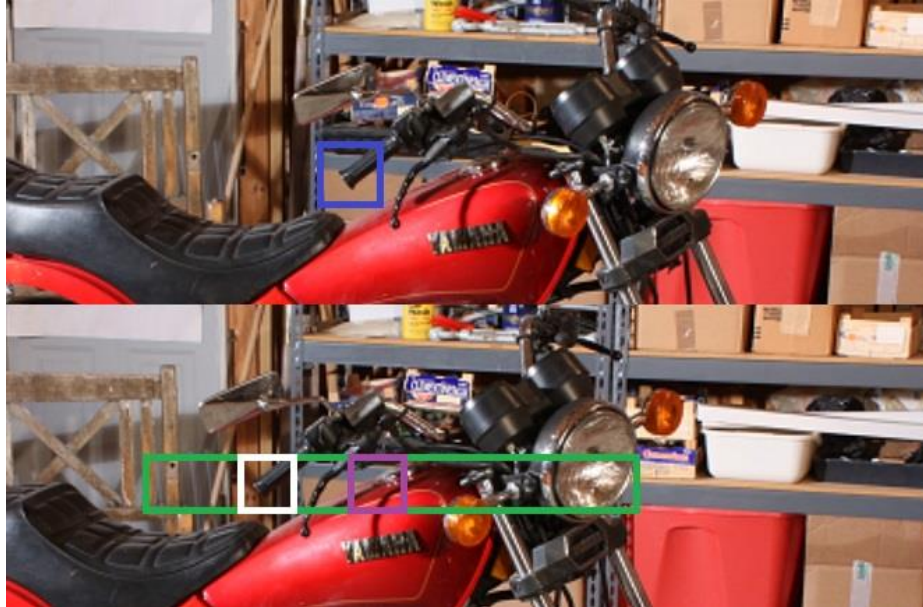
Dvi viršutinės nuotraukos atvaizduoja kamerų vaizdą prieš kalibravimą. Yra matomi epipolinių linijų neatitikimai, šachmatų lenta nėra vienodame aukštyje. Apatinės dvi nuotraukos atvaizduoja vaizdų porą po kalibravimo. Šachmatų lentos kvadratai yra viename aukštyje ir aplinkos objektai turi vienodas epipolines linijas.

Atlikus kamerų kalibravimą galima realizuoti erdvinio vaizdo algoritmus.



### 4.3 „Block match“ algoritmas

„Block match“ - tai algoritmas skirtas erdvinio vaizdo generavimui. Šiam algoritmui yra naudojamos dvi erdvinio vaizdo poros, kuriose yra lyginimai blokai ir skaičiuojami blokų skirtumų sumos [25]. Blokas su mažiausia suma nurodo atitinkantį kitos nuotraukos makrobloką. Makrobloko poslinkio dydis nurodo erdvinio vaizdo gylį tame makrobloke. Atlikus visų makroblokų patikrinimą yra gaunamas erdvinis vaizdas (4.5 pav.).



4.7 pav. Makroblokų lyginimas

BM algoritmas dažniausiai yra skaičiuojamas keliais būdais: skaičiuojant makroblokų sumų skirtumo modulius (SAD) arba skaičiuojant makroblokų sumų skirtumus pakeltus kvadratu (SSD).

SAD metodas yra išreiškiamas formule (4.12):

$$\sum_{(i,j) \in W} |I_1(i,j) - I_2(x+i, y+j)|; \quad (4.14)$$

Čia  $i$  yra paveikslo  $x$  koordinatė,  $j$  – paveikslo  $y$  koordinatė,  $I_1$  – pirmas paveikslas,  $I_2$  – antras paveikslas,  $x$  – makrobloko  $x$  koordinatė,  $y$  – makrobloko  $y$  koordinatė. Panašią metodą galima apskaičiuoti ir SSD reikšmes (4.13):

$$\sum_{(i,j) \in W} (I_1(i,j) - I_2(x+i, y+j))^2; \quad (4.15)$$

Čia  $i$  yra paveikslo  $x$  koordinatė,  $j$  – paveikslo  $y$  koordinatė,  $I_1$  – pirmas paveikslas,  $I_2$  – antras paveikslas,  $x$  – makrobloko  $x$  koordinatė,  $y$  – makrobloko  $y$  koordinatė.

Algoritmas veikia tokiu principu:

*For ciklu pasirenkami paveikslo  $x$  pikseliai*

*For ciklu pasirenkami paveikslo  $y$  pikseliai*

*Nustatoma mažiausia makroblokų skirtumo reikšmė*

*Nustatoma aukščiausia SAD reikšmė*

*For ciklu einama per nustatytą makroblokų skirtumo reikšmę*

*For ciklu einame per nustatytą makrobloko dydžio  $x$  reikšmes*

*For ciklu einame per nustatytą makrobloko dydžio  $y$  reikšmes*

*Sumuojame visas paveikslų skirtumo modulio reikšmes*

*Tikriname ar apskaičiuota makrobloko reikšmė yra mažesnė už prieš tai skaičiuotą*

*Jeigu reikšmė mažesnė nurodome, kad apskaičiuota reikšmė atitinka kitą makrobloką*

*Apskaičiavus pikselio makroblokus, gylį reikšmę įvedama į erdvinio vaizdo matricą*

„Block match“ SAD metodas realizuotas šiuo algoritmu:

```
for y in xrange(window, height - window):
    for x in xrange(window, width - window):
        min_dif = 0
        lastSAD = 9999999999
        for dif in xrange(numdisp):
            SAD = 0
            for v in xrange(-window, window):
                for u in xrange(-window, window):
                    SAD = SAD + abs(int(l[y+v, x+u]) - int(r[y+v, (x+u) - dif]))
            if SAD < lastSAD:
                lastSAD = SAD
                min_dif = dif
        disparity[y, x] = min_dif
```

„Block match“ SSD algoritmo principas yra panašus į SAD, tik skirtumo reikšmės yra ne modulinės, o pakeltos kvadratu. Taip yra gaunamas tikslesnis erdvinis vaizdas. SSD reikšmės yra apskaičiuojamos šiuo metodu:

$$SSD = SSD + (int(l[y+v, x+u]) - int(r[y+v, (x+u) - dif]))**2$$

Skaičiavimo funkcijai yra pateikiami kairysis bei dešinysis erdvinio vaizdo kameros paveikslai. Paveikslų spalvos pakeičiamos į nespaltvotus, tam kad pikselių reikšmės būtų nuo 0 iki 255. Funkcijai taip pat nurodomas pikselio tinklėlio dydis, bei pikselio poslinkio reikšmė. Parinkus teisingus parametrus yra gaunamas erdvinis vaizdas (4.6 pav.).



**4.8 pav.** SAD metodu realizuotas erdvinis vaizdas

Sugeneruotame erdvinio vaizdo paveiksle skirtingas intensyvumo lygis nurodo skirtingą paveiksle esančių objektų gylį. Kuo didesnis intensyvumas, tuo objektas yra arčiau kameros. Mažesnis intensyvumas reiškia, kad objektas yra toliau. Erdvinis vaizdas šiuo algoritmu nėra tobulas, nes paveiksle yra daug vietų, kur sunku nustatyti disparatiškumą.

#### 4.4 „Semi global block match“ algoritmas

SGBM metodas yra aprašytas autoriaus Hirschmüller [12]. Šis metodas apjungia globalius ir lokalius erdvinio vaizdo metodus. Algoritmo realizavimui būtina naudoti sukalilbruotos kameros vaizdus su atitinkančiomis epipolinėmis linijomis (4.9 pav.)



4.9 pav. Paveikslų epipolinės linijos

Algoritmas skaičiavimams naudoja paveikslų poras su žinomomis vidinėmis ir išorinėmis objektų orientacijomis. Algoritmo skaičiavimui yra naudojama  $E(D)$  formulė (2.7):

$$E(D) = \sum_p (C(p, D_p) + \sum_{q \in N_p} P_1 I[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 I[|D_p - D_q| > 1]); \quad (4.16)$$

čia

$E(D)$  erdvinio vaizdo energija;

$q$  ir  $p$  nurodo paveikslo pikselių indeksus;

$N_p$  yra artimiausias pikselio  $p$  kaimynas;

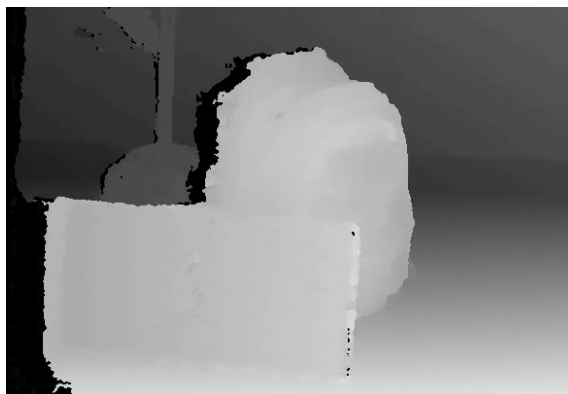
$C(p, D_p)$  nurodo pikselio tapatumo reikšmę su gylio reikšme  $D_p$ ;

$P_1$  yra koeficientas nurodantis pikselių pasikeitimus tarp kaimynų, kurie lygūs vienas;

$P_2$  yra koeficientas nurodantis pikselių pasikeitimus tarp kaimynų, kurie yra daugiau nei vienas;

$I$  yra funkcija kuri grąžina vienetą jeigu argumentas yra teisingas arba nulį, jeigu yra priešingai.

Minimizuota funkcija generuoja erdvinį vaizdą (4.10 pav.)



4.10 pav. „Semi global block match“ algoritmu sugeneruotas erdvinis vaizdas

Gautas erdvinis vaizdas yra tikslesnis nei naudojant BM metodą, tačiau algoritmo skaičiavimo laikas yra ilgesnis. Ši algoritmą galima pritaikyti kuriamai kliūčių išvengimo sistemai.



#### 4.5 Erdvinio vaizdo algoritmų greitaveikos testavimas

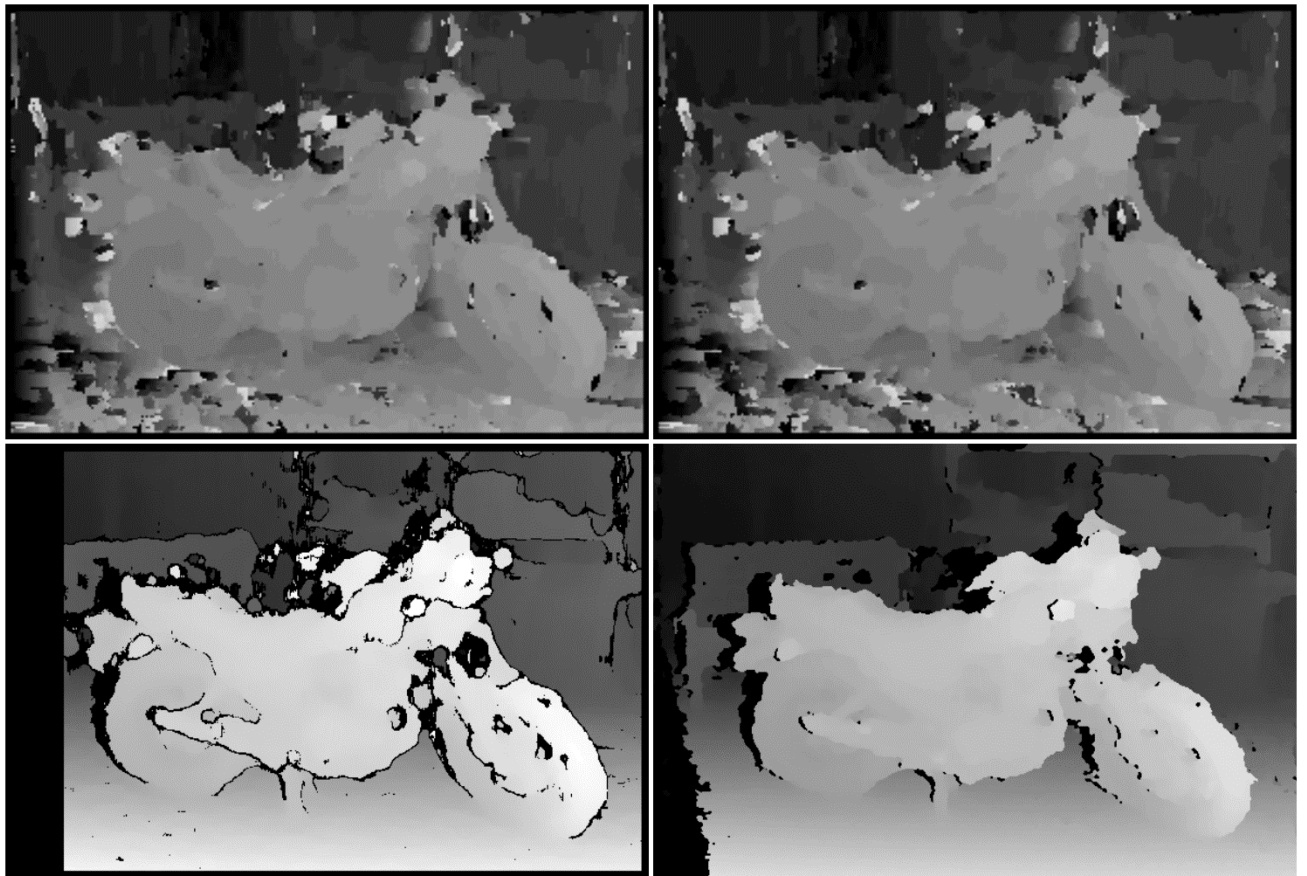
Greitaveikos testavimo metu yra fiksuojamas keturių erdvinio vaizdo algoritmų skaičiavimo laikas. Testavimui naudojamos dvi realizuotos „block match“ funkcijos bei dvi „opencv“ disparatiškumo skaičiavimo funkcijos – „block match opencv“ ir „semi global block match opencv“. Pastarieji algoritmai yra įvardinami trumpiniais. Algoritmų realizacija įvykdyta naudojant „Odroid XU4“ kompiuterį. Kompiuterio procesorius – „Exynos5422 Cortex-A15“ ir „Cortex-A7“, operatyvinė atmintis - 2 GB LPDDR3 RAM. Skaičiavimui naudojamas vienas procesoriaus branduolys iš aštuonių. Testuojamos trys paveikslų poros, kuriose objektai yra skirtingame gylyje [21].

Pirma testuojama paveikslų pora yra motociklas (4.8 pav.)



4.11 pav. Paveikslų pora – motociklas

Sugeneruoti erdviniai vaizdai pateikiami tokia tvarka: viršuj kairėje – BM SAD, viršuje dešinėje – BM SSD, apačioje kairėje – BM CV, apačioje dešinėje – SGBM CV (4.9 pav.).



4.12 pav. Testavimo rezultatai - motociklas

Algoritmų parametrai:

- BM SAD: lango dydis – 3, disparatiškumas – 32;
- BM SSD: lango dydis – 3, disparatiškumas – 32;
- BM OpenCV: lango dydis – 3, disparatiškumas – 32;
- SGBM OpenCV: lango dydis – 3, mažiausia disparatiškumo reikšmė – 4, disparatiškumas – 48; unikalumas – 2, taško lango dydis – 1000, taško atstumas – 5, didžiausias galimas skirtumas – 1, P1 – 606, P2 – 1399.

Taikant šiuos parametrus nustatyti šie algoritmų vykdymo laikai (4.1 lent.).

**4.1 lentelė** Testavimo rezultatai - motociklas

	<b>BM SAD</b>	<b>BM SSD</b>	<b>BM OpenCV</b>	<b>SGBM OpenCV</b>
Laikas, s	25.11	27.35	0.09	0.76

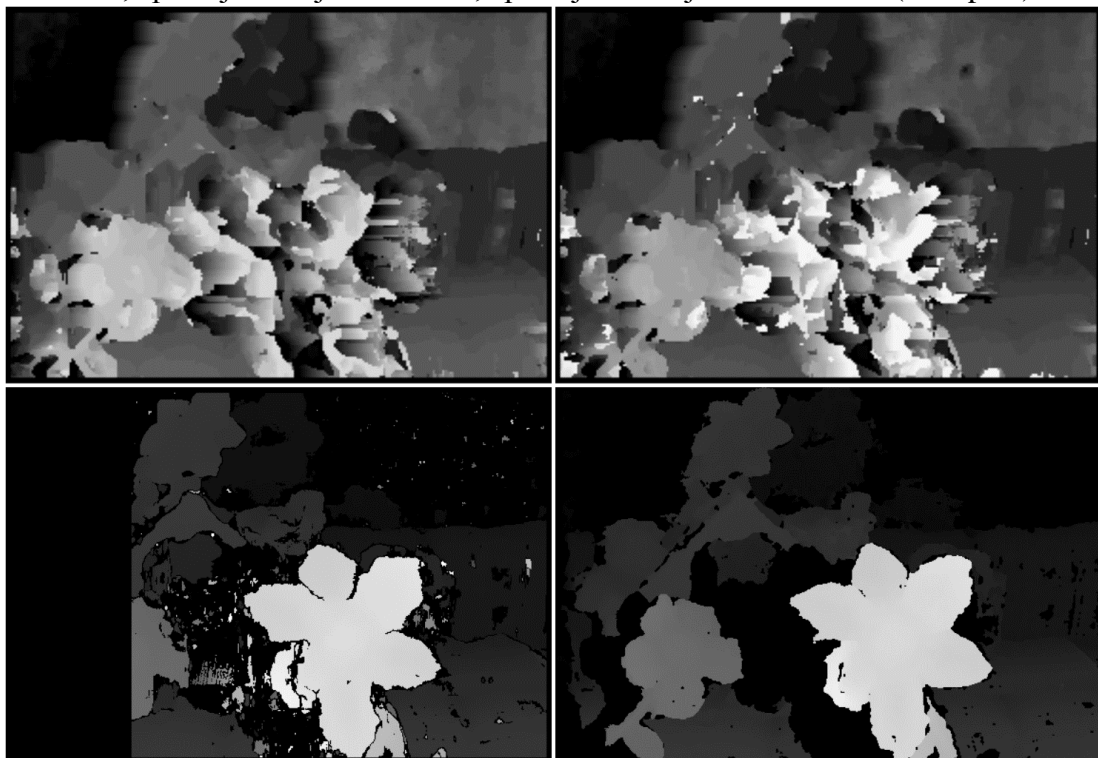
Apžvelgus gautus rezultatus galima teigti, kad greičiausiai veikia BM OpenCV algoritmas, kurio skaičiavimo greitis 0.09 s, toliau seka SGBM OpenCV su 0.76s, toliau BM SAD su 25.11s ir lėčiausiai BM SSD, kurio vykdymo laikas yra 27.35 sekundės.

Antra testuojama paveikslų pora yra gėlės (4.10 pav.)



**4.13 pav.** Paveikslų pora - gėlės

Sugeneruoti erdviniai vaizdai pateikiami tokia tvarka: viršuje kairėje – BM SAD, viršuje dešinėje – BM SSD, apačioje kairėje – BM CV, apačioje dešinėje – SGBM CV (4.11 pav.).



**4.14 pav.** Testavimo rezultatai – gėlės



Algoritmų parametrai:

- BM SAD: lango dydis – 3, disparatiškumas – 42;
- BM SSD: lango dydis – 3, disparatiškumas – 42;
- BM OpenCV: lango dydis – 13, disparatiškumas – 128;
- SGBM OpenCV: lango dydis – 6, mažiausia disparatiškumo reikšmė – 1, disparatiškumas – 128; unikalumas – 0, taško lango dydis – 1000, taško atstumas – 1, didžiausias galimas skirtumas – 5, P1 – 0, P2 – 0.

Taikant šiuos parametrus nustatyti šie algoritmų vykdymo laikai (4.2 lent.).

**4.2 lentelė** Testavimo rezultatai - gėlės

	<b>BM SAD</b>	<b>BM SSD</b>	<b>BM OpenCV</b>	<b>SGBM OpenCV</b>
Laikas, s	34.62	35.59	0.11	2.03

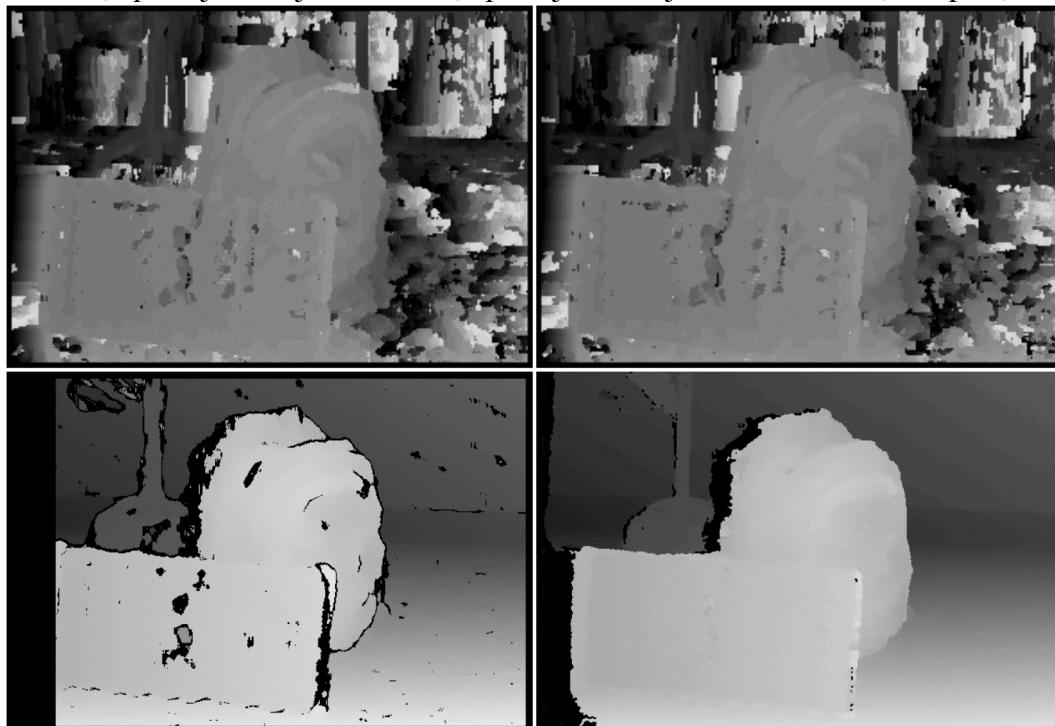
Apžvelgus gautus rezultatus galima teigti, kad greičiausiai veikia BM OpenCV algoritmas, kurio skaičiavimo greitis 0.11s, toliau seka SGBM OpenCV su 2.03s, toliau BM SAD su 34.62s ir lėčiausiai BM SSD, kurio vykdymo laikas yra 35.59 sekundės.

Trečia testuojama paveikslų pora yra gėlės (4.12pav.)



**4.15 pav.** Paveikslų pora - kuprinė

Sugeneruoti erdviniai vaizdai pateikiami tokia tvarka: viršuje kairėje – BM SAD, viršuje dešinėje – BM SSD, apačioje kairėje – BM CV, apačioje dešinėje – SGBM CV (4.13 pav.).



**4.16 pav.** Testavimo rezultatai - kuprinė

Algoritmų parametrai:

- BM SAD: lango dydis – 3, disparatiškumas – 32;
- BM SSD: lango dydis – 3, disparatiškumas – 32;
- BM OpenCV: lango dydis – 17, disparatiškumas – 48;
- SGBM OpenCV: lango dydis – 3, mažiausia disparatiškumo reikšmė – 1, disparatiškumas – 48; unikalumas – 0, taško lango dydis – 1000, taško atstumas – 11, didžiausias galimas skirtumas – 1, P1 – 212, P2 – 1415.

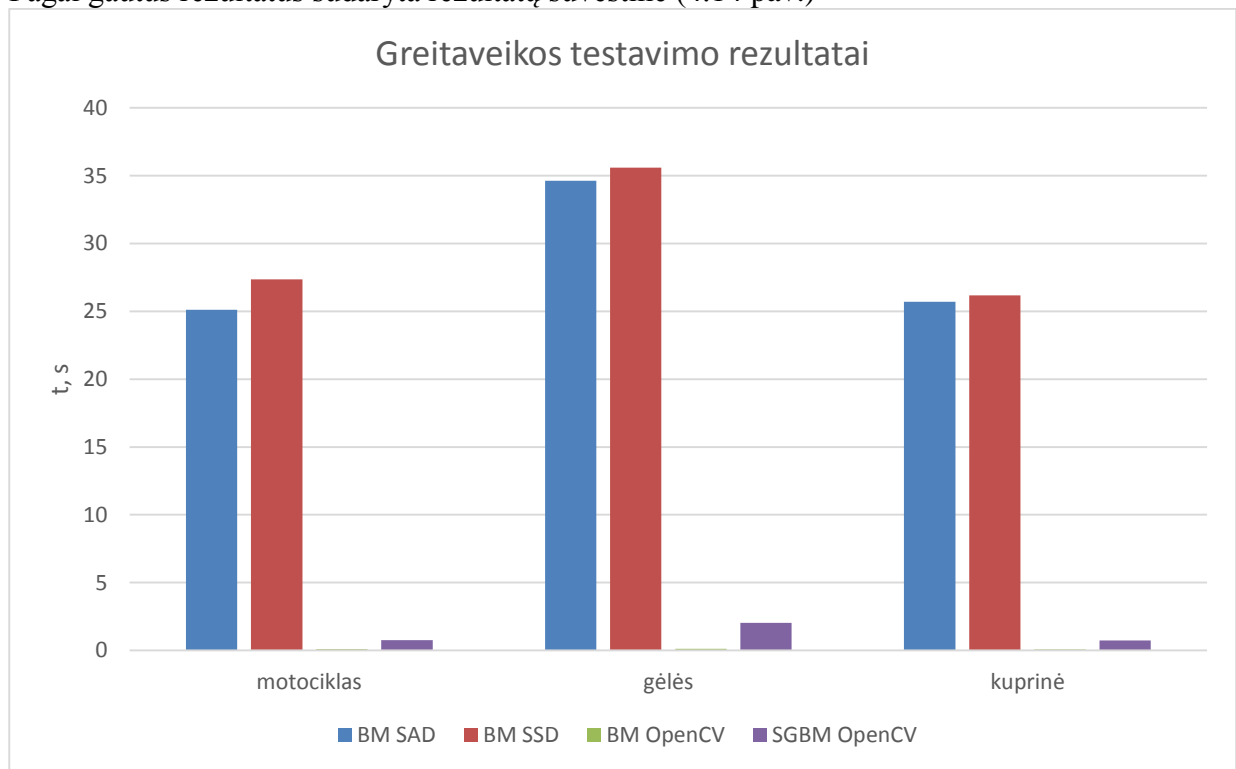
Taikant šiuos parametrus nustatyti šie algoritmų vykdymo laikai (4.3 lent.).

**4.3 lentelė** Testavimo rezultatai - kuprinė

	<b>BM SAD</b>	<b>BM SSD</b>	<b>BM OpenCV</b>	<b>SGBM OpenCV</b>
Laikas	25.67	26.17	0.06	0.74

Apžvelgus gautus rezultatus galima teigti, kad greičiausiai veikia BM OpenCV algoritmas, kurio skaičiavimo greitis 0.06 sekundės, toliau seka SGBM OpenCV su 0.74s, toliau BM SAD su 25.67s ir lėčiausiai BM SSD, kurio vykdymo laikas yra 26.17 sekundės.

Pagal gautus rezultatus sudaryta rezultatų suvestinė (4.14 pav.)



**4.17 pav.** Greitaveikos testavimo rezultatai

Pagal gautus duomenis galima daryti išvadą, kad greičiausiai veikia OpenCV „block match“ ir „semi global block match“ algoritmai. Realizuoti „block match“ SAD ir SSD algoritmai veikia lėtai, nes jie realizuoti „Python“ programavimo kalba. Pagal vidurkius pateiktus lentelėje (4.4 lent.) yra pastebima, kad BM OpenCV algoritmas vykdomas 1.09 sekunde greičiau nei SGBM, 29.61 sekunde greičiau nei BM SSD, 28.39 sekunde greičiau nei BM SAD algoritmu.

**4.4 lentelė** Algoritmų vykdymo laiko vidurkiai

	<b>BM SAD</b>	<b>BM SSD</b>	<b>BM OpenCV</b>	<b>SGBM OpenCV</b>
motociklas	25.10	27.34	0.09	0.76
gėlės	34.62	35.60	0.11	2.03
kuprinė	25.70	26.17	0.06	0.74
vidurkis	28.47	29.70	0.09	1.18

Realizuojant kliūčių išvengimo sistemą būtina atsižvelgti ne tik į algoritmų vykdymo laiką, bet ir į erdvinio vaizdo kokybę. Kokybė vertinama parametrais: žemas, vidutinis, aukštas. Paveikslų porų vertinimas pateiktas lentelėje (4.5 lent.).

**4.5 lentelė** Erdvinio vaizdo kokybės vertinimas

	<b>BM SAD</b>	<b>BM SSD</b>	<b>BM OpenCV</b>	<b>SGBM OpenCV</b>
motociklas	Žemas	Žemas	Vidutinis	Aukštas
gėlės	Žemas	Žemas	Vidutinis	Aukštas
kuprinė	Žemas	Žemas	Vidutinis	Aukštas

Pagal vertinimo lentelę, yra pastebima, kad SGBM OpenCV algoritmas generuoja aukščiausios kokybės erdvinį vaizdą, jame mažai triukšmo bei yra aiškiai matomas objektų gylis. BM OpenCV algoritmas generuoja prastesnės kokybės vaizdus, tačiau gylis yra aiškiai matomas bei triukšmo nėra daug. BM SAD ir BM SSD generuoja panašios kokybės erdvinį vaizdą, tačiau triukšmas yra didelis ir disparatiškumas sunkiai matomas.

**4.6 Analizės išvados**

- Atlikus analizę nustatyta, kad greičiausiai vykdomas yra „block match“ OpenCV algoritmas
- Nustatyta, kad geriausios kokybės erdvinį vaizdą generuoja „semi global block match“ OpenCV algoritmas
- Nuspręsta kliūčių išvengimo realizacijai naudoti abu OpenCV algoritmus



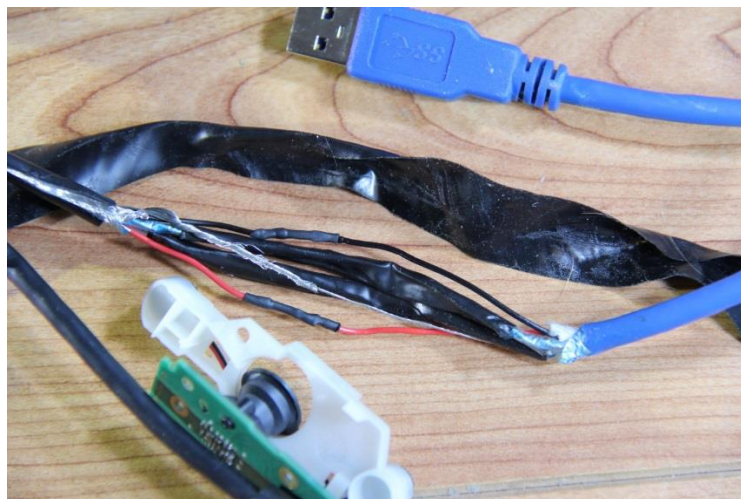
## 5 ORLAIVIO KLIŪČIŲ IŠVENGIMO SISTEMOS PROJEKTAVIMAS

### 5.1 Projektavimo tikslas

Eksperimento atlikimui būtina suprojektuoti techninę ir programinę įrangą. Techninė įranga turi būti suprojektuota taip, kad eksperimentas įvyktų sklandžiai. Programinė įranga turi būti parengta taip, kad būtų įvykdytas kliūčių išvengimas pasirinktais erdvinio vaizdo algoritmais.

### 5.2 Techninės įrangos projektavimas

Prieš pradėdant programinės įrangos projektavimą, reikia tinkamai paruošti techninę įrangą. „Playstation 4“ kamera naudoja „playstation“ AUX jungtį, tad norint kamerą prijungti prie „Odroid XU4“ kompiuterio reikia AUX jungtį pakeisti USB 3.0 jungtimi (5.1 pav.).



5.1 pav. „Playstation 4“ kameros jungtis [23]

Kamera jungiama į kompiuterio USB 3.0 lizdą. Tačiau prieš jungimą yra būtina alkoholyje išmirkytu krapštuku išvalyti USB 3.0 lizdą, tam, kad būtų pašalintas oksidavęsis metalo sluoksnis.

Kompiuterio prie autopiloto prijungimui yra naudojamas USB TTL signalo keitiklis, kuris leidžia tarpusavyje keistis MAVlink duomenimis (5.2 pav.). Keitiklis yra jungiamas į skrydžio valdiklio TELEM2 jungtį [24].



5.2 pav. USB TTL signalo keitiklis

Prie autopiloto yra prijungiami gps, telemetrijos bei radijo imtuvai, kurie leidžia bendrauti su išoriniais įrenginiais. Skrydžio valdiklis valdo keturis servo motorus bei elektroninį greičio reguliatorių. Reguliatorius pagal atitinkamą signalą valdo šešepetėlinį AC variklį. Komponentams energija teikiama naudojant 2200 mAh 7.4 V LiPo bateriją. Energija išskaidoma naudojant energijos paskirstymo modulį. Po modifikacijos, prie energijos paskirstymo modulio prijungtas 7A 5V įtampos keitiklis. Įtampos keitiklis tiekia energiją „odroid“ kompiuteriui. Rankiniam skrydžio valdymui naudojamas aštuonių kanalų „Turnigy 9x“ radijo siųstuvas (5.3 pav.). Kiekvienas siųstuvo kanalas gali valdyti bet kokią autopiloto funkciją.

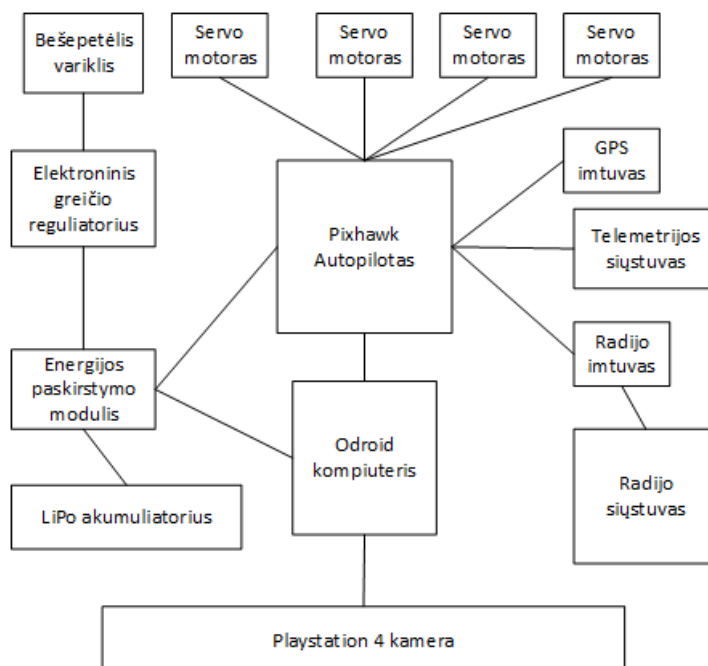


**5.3 pav.** „Turnigy 9x“ siųstuvas

Autopilotui priskirtos šios kanalų funkcijos:

- 1 kanalas – posūkio valdymas (*angl. roll*);
- 2 kanalas – aukščio valdymas (*angl. height*);
- 3 kanalas – greičio (motoro) valdymas (*angl. speed*);
- 4 kanalas – vingiravimo valdymas (*angl. yaw*);
- 5 kanalas – skrydžio režimo valdymas (rankinis valdymas, stabilizuotas režimas, autonominis skrydis, skraidymas ratais, grįžimas namo ir t.t.);
- 6 kanalas – skrydžio režimo valdymas (rankinis valdymas, stabilizuotas režimas, autonominis skrydis, skraidymas ratais, grįžimas namo ir t.t.);
- 7 kanalas – algoritmo inicijavimas ir stabdymas;
- 8 kanalas – algoritmo pakeitimas.

Skrydžio filmavimui naudojama „Mobius“ vaizdo kamera, filmuojama naudojant 1080P H.264 vaizdo kokybės parametrus. Visų komponentų sąajos yra pavaizduotos pateiktoje diagramoje (5.4 pav.).



**5.4 pav.** Techninės įrangos diagrama

Kliūčių išvengimo realizacijai pasirinkta naudoti lėktuvo modelį. Pagrindiniai kriterijai, kurie nulėmė šios platformos pasirinkimą yra nurodyti pateiktoje lentelėje (5.1 lent.).

#### 5.1 lentelė Orlaivio pasirinkimo kriterijai

	Lėktuvas	Multirotorius
Kaina	Vidutinė	Didelė
Žala po nukritimo	Maža	Didelė
Modifikavimo galimybė	Didelė	Maža
Stabilumas	Vidutinis	Didelis
Skrydžio laikas	Ilgas	Mažas

Pasirinkti kriterijai nulėmė lėktuvo modelio pasirinkimą. Naudojamas lėktuvo modelis yra „LRP F-1400“. Lėktuvas pasirinktas „trainer“ kategorijos, nes jis stabiliai skrenda, lėktuvo forma suteikia galimybę ilgai skristi. Taip pat propeleris yra viršuje, o ne priekyje, ko pasekoje, padarius klaidą ir atsitrenkus į žemę, propeleris nebus pažeistas. Visa įranga telpa lėktuvo liemenyje (*angl. fuselage*) ir yra apsaugota nuo smūgių. Erdvinio vaizdo kamera sumontuota leimens priekyje, virš sparno, tam, kad būtų apsaugota nuo smūgių bei nepakenktų lėktuvo balanso centrui. Originalūs sparnai pakeisti į ilgesnius bei platesnius sparnus, tam, kad būtų generuojama didesnė kėlimo jėga. Lėktuvo sparnų mojis yra 180 cm, liemens ilgis – 95 cm. Paruoštas skrydžiui modelis pateiktas paveiksle (5.5 pav.).



5.5 pav. Kliūčių išvengimo lėktuvo modelis

Bendras lėktuvo svoris su visą įranga yra 1304g. Atlikus bandomąjį skrydį nustatyta, kad lėktuvo svoris netrukdo eksperimento vykdymui, yra greitas bei manevringas. Nustatyta, kad minimalus lėktuvo greitis 8 m/s, maksimalus 30 m/s. Šis greitis yra pakankamas erdvinio vaizdo algoritmų testavimui. Bendra projekto kaina 332 eurai. Komponentų kainos ir svoriai pateikiami lentelėje (5.2 lent.).



## 5.2 lentelė Komponentų svoriai ir kainos

Komponentas	Svoris, g	Kaina, €
LiPo baterija 2200 mah 7.4 V	131	10
„Playstation 4“ kamera	112	40
„Odroid XU4“ kompiuteris	91	100
„Pixhawk“ autopilotas	49	70
Įtampos keitiklis 5.5 V 7 A	18	5
Įtampos paskirstymo modulis	27	7
Radio imtuvas	25	10
Telemetrijos siųstuvas	24	10
Mobius kamera	44	30
GPS imtuvas	28	10
„LRP F-1400“ Lėktuvas	755	40
Iš viso	1304	332

## 5.3 Programinės įrangos projektavimas

Programinės įrangos projektavimo dalyje yra kuriama programinė įranga reikalinga eksperimento įvykdymui.

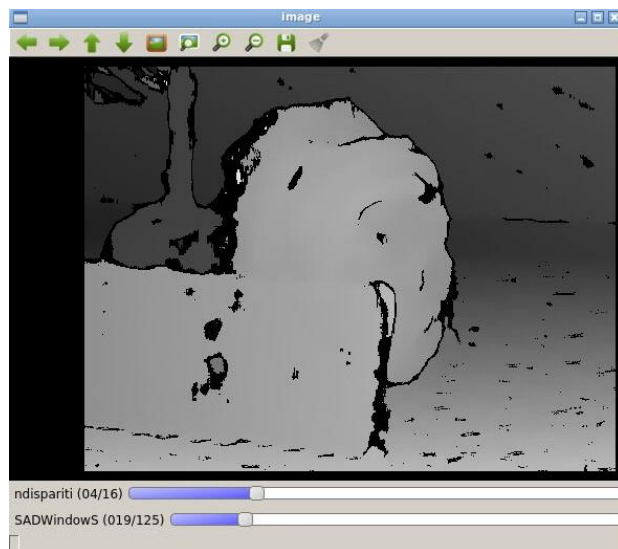
Programinės įrangos projektavimas susideda iš šių dalių:

- Kameros kalibravimas;
- Erdvinio vaizdo generavimas pasirinktais algoritmais;
- Kliūčių nustatymas iš erdvinio vaizdo;
- Kompiuterio ir autopiloto komunikavimas.

Sistemos kūrimui pasirinkta „python“ programavimo kalba, nes ją naudojant yra įmanomas sklandus kameros, kompiuterio ir autopiloto tarpusavio ryšys. Algoritmai yra kuriami naudojant „opencv 2.4.12.1“ kompiuterinės regos biblioteką. Ši biblioteka turi įrankius, kuriais galima sukurti programinę įrangą skirtą eksperimento atlikimui.

Programinės įrangos projektavimas prasideda nuo kameros kalibravimo. Kadangi kamera gali transliuoti keliomis raiškomis, kalibravimui yra pasirinktos dvi raiškos - 319x192 ir 637x400. Kameros transliuojamas vaizdas yra sujungtas į vieną kadrą, tad visų pirma yra atskiriamas kairiojo lęšio vaizdas nuo dešinio. Tai atliekama naudojant *img[0:192, 48:367]* komandą. Kalibravimas pradedamas skaičiuojant šachmatų lentos dvimačius bei trimačius erdvės taškus. Apdorojus bent 30 lentos vaizdų skirtingomis lentos pozicijomis yra atliekamas kairės ir dešinės kameros kalibravimas. Naudojant *cv2.calibrateCamera()* funkciją yra gaunamos kamerų matricos bei vaizdo iškraipymo koeficientai. Šie koeficientai yra siunčiami į *cv2.stereoCalibrate()* funkciją, su kuria yra skaičiuojamos abiejų kamerų transformacijos. Grąžinti nauji koeficientai yra siunčiami į *cv2.stereoRectify()* funkciją, kurios pagalba yra surandamos paveikslų epipolinės linijos. Su funkcija *cv2.initUndistortRectifyMap()* yra sugeneruojama transformacijos matrica, kurią galima taikyti naujoms kamerų vaizdų poroms. Šios matricos yra sugeneruojamos kairiam ir dešiniam vaizdui skirtingomis raiškomis ir yra išsaugomos faile. Kamera yra galutinai sukalibruota, kiekvienai vaizdų porai taikant *remap()* funkciją.

Kliūčių išvengimo testavimui pasirinkti „opencv“ bibliotekos „block match“ ir „semi global block match“ erdvinio vaizdo algoritmai. BM algoritmas yra realizuojamas naudojant funkciją *cv2.StereoBM(cv2.STEREO\_BM\_BASIC\_PRESET,ndisparities=16, SADWindowSize=25)*, kur *ndisparities* nurodo erdvinio vaizdo skirtumo dydį, *SADWindowSize* nurodo makrobloko dydį. Erdvinio vaizdo generavimui sukurta grafinė naudotojo sąsaja, kuria galima nustatyti tinkamus disparatiškumo parametrus (5.6 pav.).

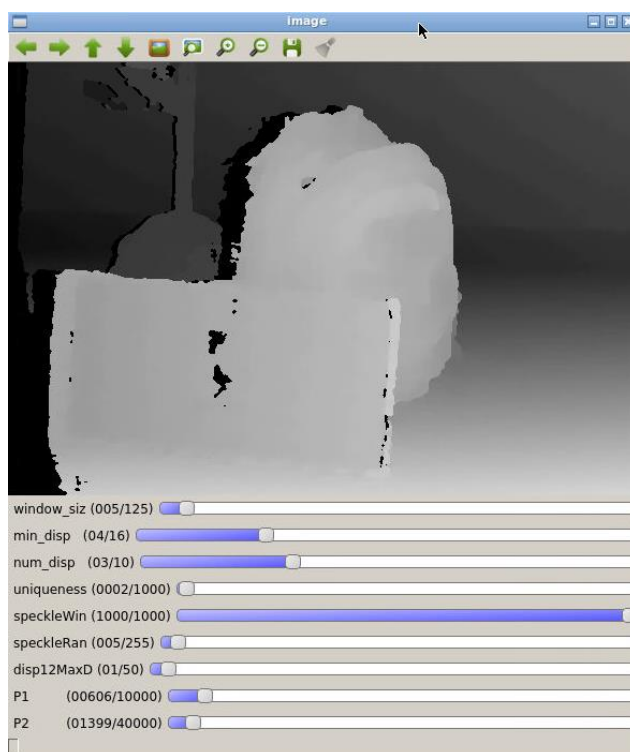


**5.6 pav.** BM parametrų testavimo grafinė sąsaja

„Semi global block match“ algoritmas yra iškviečiamas su funkcija *cv2.StereoSGBM()*. Šiai funkcijai yra nurodomi parametrai:

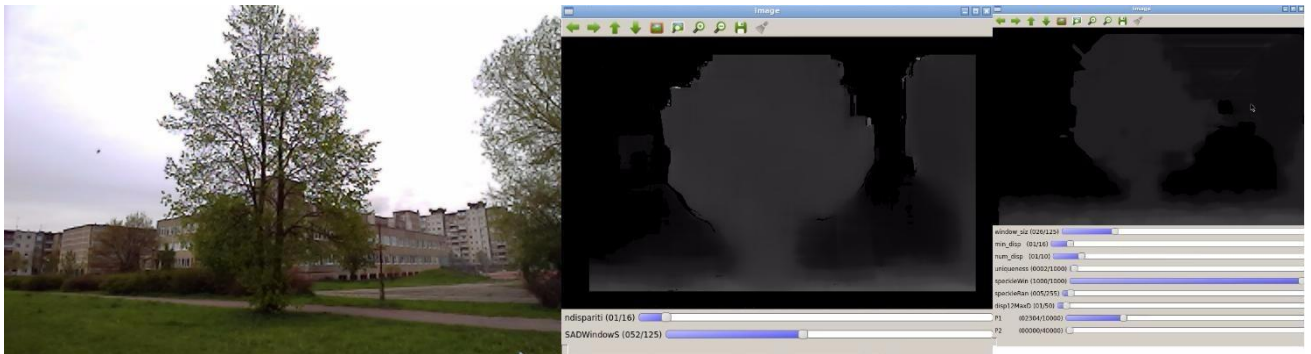
- minDisparity – minimalus disparatiškumo dydis;
- numDisparities – disparatiškumo dydis;
- SADWindowSize – lango dydis;
- uniquenessRatio – unikalumo santykis;
- speckleWindowSize – taško dydis;
- speckleRange – taško gylis;
- disp12MaxDiff – didžiausias galimas skirtumas;
- P1 - koeficientas nurodantis pikselių pasikeitimus tarp kaimynų, kurie skaičius lygūs vienam
- P2 – koeficientas nurodantis pikselių pasikeitimus tarp kaimynų, kurių skaičius yra didesnis nei vienas;

Šiam algoritmui taip pat pritaikyta grafinė sąsaja (5.7 pav.).



**5.7 pav.** SGBM parametrų testavimo grafinė sąsaja

Projektuojant kliūčių išvengimo sistemą būtina parinkti tikslius erdvinio vaizdo parametrus. Erdviniui vaizdui sudaryti yra pasirinktas vaizdas su objektu esnčių už dešimties metrų atstumo. Pritaikius tinkamus parametrus BM ir SGBM algoritmas yra gauti teisingi erdviniai vaizdai (5.8 pav.).



5.8 pav. Kliūtis erdvinis vaizdas

Kliūčių išskyrimas iš erdvinio vaizdo gali būti realizuotas erdvini vaizdą pavertus į taškų debesį. Taškų debesio kūrimui yra naudojama *cv2.reprojectImageTo3D()* funkcija. Tačiau ši funkcija reikalauja daug procesoriaus resursų, tad kliūčių išskyrimui yra naudojamas kitas metodas.

Erdvinio vaizdo paveiksle objektai yra atvaizduojami tam tikru intensyvumu. Siekiant išvengti kliūti tam tikru atstumu, galima išskirti tos kliūtis intensyvumą erdvinio vaizdo paveiksle. Sugeneruoto vaizdo pikselių reikšmės yra nuo 0 iki 1. Išskyrus intensyvumą nuo 0.1 iki 0.2 yra gaunamas kliūtis atvaizdas ties tam tikru atstumu (5.9 pav.).



5.9 pav. Kliūtis nustatymas

Paveiksle raudonos vietos pažymi aptiktą kliūtį kelyje.

Nustačius kliūtis skrydžio kelyje, būtina jas išvengti. Tai yra atliekama siunčiant servo motorus valdančias komandas į skrydžio valdiklį. Skrydžio valdiklis yra prijungiamas prie skrydžio kompiuterio naudojant „dronekit“ biblioteką. Šia biblioteka yra inicijuojama „mavlink“ jungtis tarp prietaisų. Kompiuteris gali gauti įvairius skrydžio duomenis realiuoju laiku. Pavyzdžiui, funkcija *vehicle.channels['I']*, nurodo siųstuvo pirmo kanalo pwm reikšmę. Naudojant šią funkciją galima sužinoti ar yra inicijuotas siųstuvo mygtukas. Taip pat galima perimti lėktuvo servo motorų valdymą. Funkcija *vehicle.channels.overrides['I'] = 1400* inicijuoja lėktuvo sukimąsi į kaire pusę tam tikru kampu. Naudojant šias funkcijas yra vykdomas kliūčių išvengimas, kai skrydžio kelyje yra nustatoma kliūtis.

## 5.4 Išvados

- Kliūčių išvengimui suprojektuota techninė įranga. Skrydžio platformai pasirinktas lėktuvo modelis. Lėktuvo modelis pasirinktas, nes eksperimento atlikimas naudojant šią platformą yra saugesnis nei naudojant „multirotorių“.
- Suprojektuota kliūčių išvengimo programinė įranga. Atliktas kameros kalibravimas, erdvinio vaizdo generavimas, kliūčių nustatymas iš erdvinio vaizdo, inicijuotas autopiloto komunikavimas su kompiuteriu.

## 6 KLIŪČIŲ IŠVENGIMO EKSPERIMENTAS

### 6.1 Eksperimento planas

Eksperimentas pradedamas gerai ištestavus techninę bei programinę įrangą. Eksperimentas atliekamas atviroje erdvėje, esant kliūtims skrydžio kelyje. „Mission planner“ programinės įrangos pagalba nurodoma skrydžio trajektorija, kuria lėktuvas skris kol užduotis bus įvykdyta. Įvykdžius testavimo skrydžius, surenkami eksperimento rezultatai.

### 6.2 Eksperimento eiga

Kliūčių išvengimo eksperimentui pasirinkta testuoti kelis tarpusavio parametrus:

- Lango dydis - 319x192 arba 637x400;
- Algoritmas – „block match“ arba „semi global block match“;
- Lygiagretus skaičiavimas – įjungtas arba išjungtas.

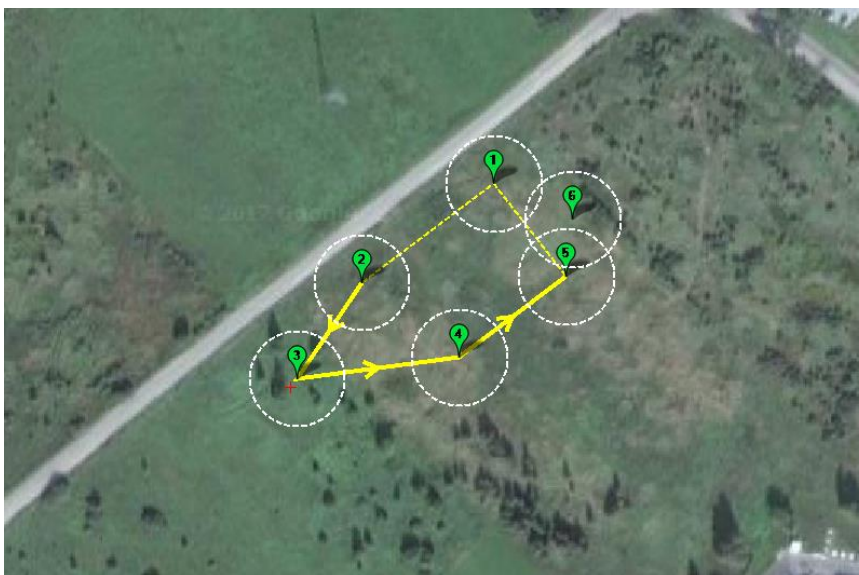
Pagal šiuos parametrus yra sukurti aštuoni testavimo atvejai.

#### 6.1 lentelė Testavimo atvejai

	Lango dydis	Algoritmas	Lygiagretus skaičiavimas
1 testas	319x192	BM	Ne
2 testas	319x192	BM	Taip
3 testas	319x192	SGBM	Ne
4 testas	319x192	SGBM	Taip
5 testas	637x400	BM	Ne
6 testas	637x400	BM	Taip
7 testas	637x400	SGBM	Ne
8 testas	637x400	SGBM	Taip

Algoritmų keitimas vykdomas radijo siųstuvo pagalba, naudojant jungiklį, kuris valdo aštuntą radijo kanalą. Kadangi kliūties išvengimo testavimas vykdomas tik prieš pat kliūtį, įdiegtas algoritmo įjungimo – išjungimo jungiklis, kuris priskirtas septintam radijo kanalui. Kiekvienas testas yra išbandomas tris kartus.

Autopiloto programavimui naudojama „Mission Planner“ programinė įranga. Jos pagalba yra atliktas lėktuvo sensorių kalibravimas bei reikalingų parametrų nustatymas. Lėktuvui yra nurodomas autonominio skrydžio kelias. Kelias parinktas toks, kad lėktuvas skristų į medžius, esančius penkių metrų aukštyje (6.1 pav.).



6.1 pav. Autonominio skrydžio planas



Lėktuvui užfiksavus kliūtį yra išsaugoma aptiktos kliūties nuotrauka. Taip pat yra išsaugoma kompiuterio ir autopiloto duomenų informacija:

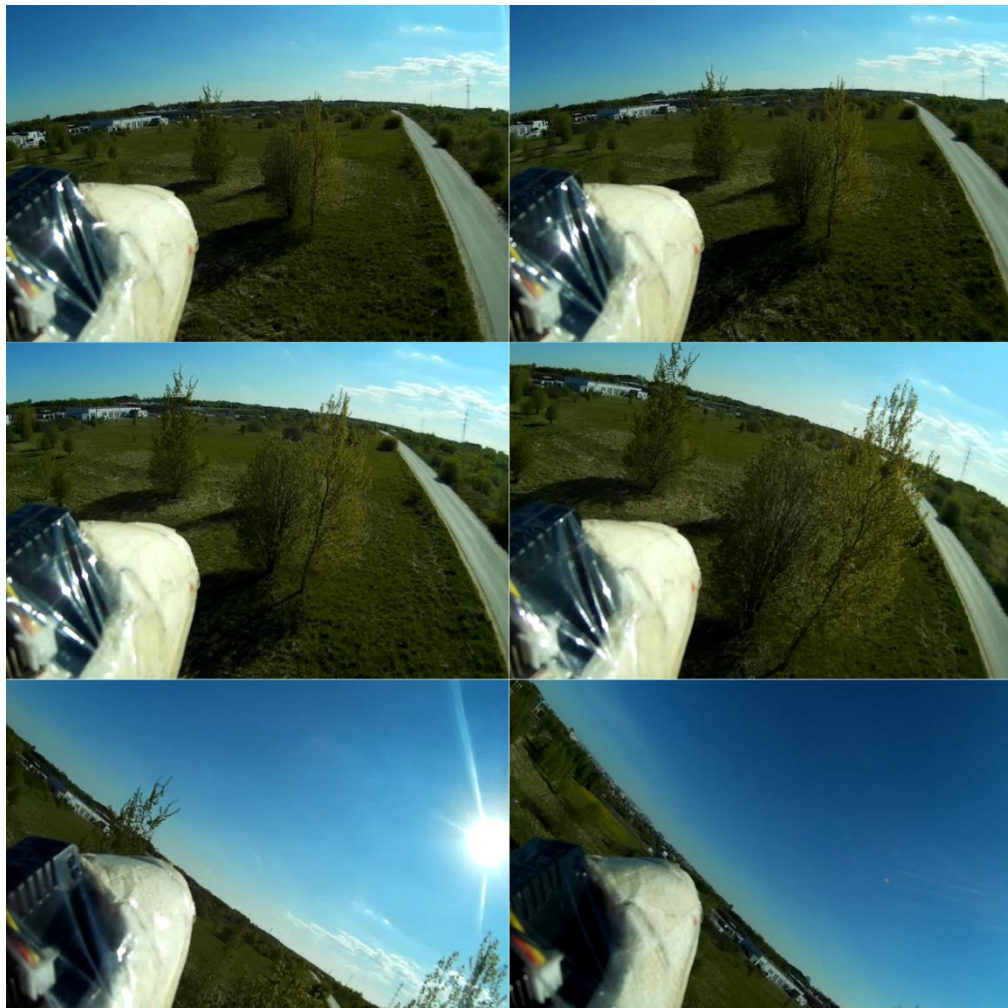
- Algoritmas;
- Lango dydis;
- Paralelinis skaičiavimas;
- Kadrai per sekundę;
- Kadro intervalas, ms;
- Vėlinimas, ms;
- Greitis, m/s;
- Aukštis, m;
- GPS koordinatės.

Oro sąlygos eksperimento metu:

- Oro temperatūra – 20;
- Vėjo greitis - 2 m/s;
- Matomumas – geras;
- Krituliai - 0%;
- Atmosferos reiškiniai – giedra.

### 6.3 Eksperimento rezultatai

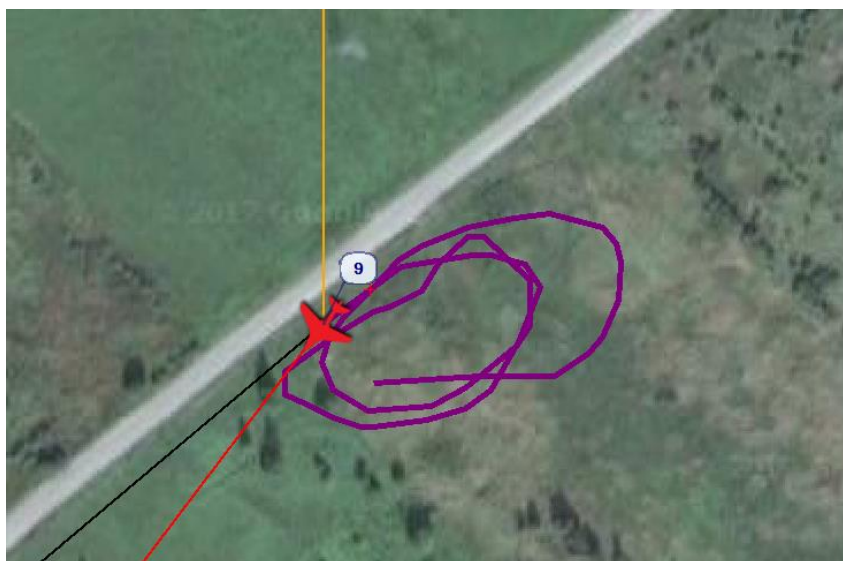
Eksperimentas atliktas sklandžiai. Orlaivio skrydžio trajektorija yra pastovi be didesnių nukrypimų. Skrydžio aukštis tenkina eksperimento reikalavimus. Naudojant „Mobius“ vaizdo kamerą yra gauta skrydžio vaizdo medžiaga (6.2 pav.).



6.2 pav. Vaizdo kameros medžiaga



„Mission Planner“ programinės įrangos pagalba, pateikiama skrydžio kelio vizualizacija (6.3 pav.).



**6.3 pav.** Skrydžio kelias

Eksperimento metu kliūčių nustatymas įvykdytas vienu iš aštuonių testavimo atvejų. Kliūtys visose trijose iteracijose nustatytos BM algoritmu naudojant 319x192 dydžio langą, su išjungtu lygiagrečiu skaičiavimu. Pirmosios iteracijos metu aptiktų kliūčių vaizdas pateiktas paveiksle (6.4 pav.). Paveiksle, raudona spalva, matomos nustatytos dvi kliūtys.



**6.4 pav.** BM algoritmo testavimo pirmoji iteracija

Pirmos iteracijos metu, aptikus kliūtį, gauti lentelėje pateikti duomenys (6.2 lent.).

**6.2 lentelė** BM algoritmo testavimo pirmos iteracijos duomenys

Algoritmas	Opencv Block match
Lango dydis	319x192
Lygiagretus skaičiavimas	Ne
Kadrai per sekundę	30
Kadro intervalas, ms	32.9299130951
Vėlinimas, ms	24.8615395201
Greitis, m/s	9.41805171967
Aukštis, m	4.91
GPS koordinatės	lat=54.9369282,lon=23.8674157

Antros iteracijos metu nustatytos 4 kliūtys (6.5 pav.)



**6.5 pav.** BM algoritmo testavimo antroji iteracija

Antros iteracijos duomenys pateikiamos lentelėje (6.3 lent.).

**6.3 lentelė** BM algoritmo testavimo antros iteracijos duomenys

Algoritmas	Opencv Block match
Lango dydis	319x192
Lygiagretus skaičiavimas	Ne
Kadrai per sekundę	30
Kadro intervalas, ms	32.9505384595
Vėlinimas, ms	16.8171964568
Greitis, m/s	17.4672393799
Aukštis, m	6.36
GPS koordinatės	lat=54.9362868,lon=23.8678017

Trečios iteracijos metu yra aptikta viena kliūtis (6.6 pav.).



**6.6 pav.** BM algoritmo testavimo trečioji iteracija

Trečios iteracijos duomenys pateikiamos lentelėje (6.4 lent.).

**6.4 lentelė** BM algoritmo testavimo trečios iteracijos duomenys

Algoritmas	Opencv Block match
Lango dydis	319x192
Lygiagretus skaičiavimas	Ne
Kadrai per sekundę	30
Kadro intervalas, ms	33.5015775311
Vėlinimas, ms	16.3193857014
Greitis, m/s	8.5785788564
Aukštis, m	4.31
GPS koordinatės	lat=54.9302033,lon=23.8679772

Eksperimento metu nustatyta, kad kliūčių išvengimas atliktas taikant „block match“ algoritmą. Nustatytas vidutinis greitis kliūties išvengimo metu – 11.8 m/s, Vidutinis aukštis – 5.19 m, kadro intervalas 33.12 ms, kadrų vėlinimo laikas – 19.3 ms., lango dydis - 319x192 pikseliai.

## 7 IŠVADOS

- Atlikus „block match SAD“, „block match SSD“, „block match opencv“ bei „semi global block match opencv“ algoritmų greitaveikos testavimą nustatyta, kad BM OpenCV algoritmas įvykdomas 1.09 s greičiau nei SGBM algoitmas, 29.61 s greičiau nei BM SSD algoritmas ir 28.39 s greičiau nei BM SAD algoritmas.
- Sukurta techninė ir programinė kliūčių išvengimo sistema.
- Atlikus eksperimentą, nustatyta, kad kliūtys yra tiksliausiai nustatomos naudojant opencv "block match" erdvinio vaizdo algoritmą. Trejomis iteracijomis iš trijų buvo užfiksuotos kliūtys skrydžio kelyje. Vidutinis greitis kliūties išvengimo metu – 11.8 m/s, aukštis – 5.19 m, kadro intervalas 33.12 ms, lango dydis - 319x192 pikseliai.
- Atlikus eksperimentą, nustatyta, kad kliūtys yra greičiausiai nustatomos naudojant opencv "block match" erdvinio vaizdo algoritmą. Trejomis iteracijomis iš trijų buvo užfiksuoto kliūtis skrydžio kelyje. Vidutinis greitis kliūties išvengimo metu – 11.8 m/s, aukštis – 5.19 m, kadro intervalas 33.12 ms, lango dydis - 319x192 pikseliai.
- Tolimesniam kliūčių išvengimo sistemos vystymui rekomenduojama išanalizuoti kitas objektų aptikimo metodų pritaikymo galimybes. Ultragarso jutiklių susiejimas su erdvinio vaizdo išvengimo metodais, galėtų padidinti kliūčių aptikimo tikimybę.

## Literatūra

1. Valentina, D., Kompiuterijos terminai. *Enciklopedinis kompiuterijos žodynas*. [interaktyvus] [žiūrėta 2016-03-30]. Prieiga per internetą: <http://ims.mii.lt/EKŽ/>
2. Mironaitė, M., Orlaivių techniniai terminai. *Mokomasis anglų-lietuvių kalbų orlaivių techninės priežiūros terminų žodynas*. [interaktyvus] [žiūrėta 2016-03-30]. Prieiga per internetą: <http://www.lka.lt/download/8049/orlaiviu>
3. *Mavlink apibrėžimas*. [interaktyvus] [žiūrėta 2016-03-30]. Prieiga per internetą: <http://qgroundcontrol.org/mavlink/start>
4. *Pixhawk apibrėžimas*. [interaktyvus] [žiūrėta 2016-03-30]. Prieiga per internetą: <https://pixhawk.org/modules/pixhawk>
5. *Apm apibrėžimas*. [interaktyvus] [žiūrėta 2016-03-30]. Prieiga per internetą: <http://www.ardupilot.co.uk/>
6. *Radar apibrėžimas*. [interaktyvus] [žiūrėta 2016-03-30]. Prieiga per internetą: <https://en.wikipedia.org/wiki/Radar>
7. *Lidar apibrėžimas*. [interaktyvus] [žiūrėta 2016-03-30]. Prieiga per internetą: <http://waset.org/publications/9997995/lidar-obstacle-warning-and-avoidance-system-for-unmanned-aircraft>
8. *Atstumo matavimas naudojant radarą*. [interaktyvus] [žiūrėta 2016-04-12]. Prieiga per internetą: <http://www.radartutorial.eu/01.basics/Distance-determination.en.html>
9. *Odroid kompiuterio techniniai parametrai*. [interaktyvus] [žiūrėta 2016-04-12]. Prieiga per internetą: [http://www.hardkernel.com/main/products/prdt\\_info.php?g\\_code=G143452239825](http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825)
10. *Optic flow apibrėžimas*. [interaktyvus] [žiūrėta 2016-04-12]. Prieiga per internetą: [http://docs.opencv.org/3.1.0/d7/d8b/tutorial\\_py\\_lucas\\_kanade.html#gsc.tab=0](http://docs.opencv.org/3.1.0/d7/d8b/tutorial_py_lucas_kanade.html#gsc.tab=0)
11. *Dronekit apibrėžimas*. [interaktyvus] [žiūrėta 2016-04-20]. Prieiga per internetą: <http://python.dronekit.io/>
12. Hirschmüller H., R. Stereo SGBM. *Stereo Processing by Semi-Global Matching and Mutual Information*, Vol. 2, No. 5, p. 8. [interaktyvus] [žiūrėta 2016-11-15]. Prieiga per internetą: <https://core.ac.uk/download/pdf/11134866.pdf>
13. *Lidar 3d žemėlapis*. [interaktyvus] [žiūrėta 2016-11-15]. Prieiga per internetą: <http://velodynelidar.com/vlp-16-lite.html>
14. Ruzgienė, B. *Fotogrametrija: vadovėlis*. Vilnius: Technika, 2008, 203 p.
15. *Lidar 360 parametrai*. [interaktyvus] [žiūrėta 2016-11-15]. Prieiga per internetą: <http://www.robotshop.com/en/rplidar-a2-360-laser-scanner.html>
16. *Lidar rangefinder parametrai*. [interaktyvus] [žiūrėta 2016-11-15]. Prieiga per internetą: <http://www.robotshop.com/en/lidar-lite-3-laser-rangefinder.html>
17. *Lidar vlp16 parametrai*. [interaktyvus] [žiūrėta 2016-11-15]. Prieiga per internetą: <http://velodynelidar.com/vlp-16.html>
18. *TCAS pranešimų reikšmės*. [interaktyvus] [žiūrėta 2017-02-12]. Prieiga per internetą: [https://www.faa.gov/documentLibrary/media/Advisory\\_Circular/TCAS%20II%20V7.1%20Intro%20booklet.pdf](https://www.faa.gov/documentLibrary/media/Advisory_Circular/TCAS%20II%20V7.1%20Intro%20booklet.pdf)
19. Hugh D. Griffiths, Christopher J. Baker, *An Introduction to passive radar*, 2017
20. *ZED kameros parametrai*. [interaktyvus] [žiūrėta 2017-02-12]. Prieiga per internetą: <https://www.stereolabs.com>
21. *Stereo vaizdų poros*. [interaktyvus] [žiūrėta 2017-02-12]. Prieiga per internetą: <http://vision.middlebury.edu/stereo/data/scenes2014/>
22. *Kameros kalibravimas OpenCV biblioteka*. [interaktyvus] [žiūrėta 2017-02-12]. Prieiga per internetą: [http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_calib3d/py\\_calibration/py\\_calibration.html](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html)

23. *Playstation 4 jungties keitimas*. [interaktyvus] [žiūrėta 2016-05-12]. Prieiga per internetą: <http://ps4eye.tumblr.com/>
24. *Autopiloto prijungimas prie kompiuterio*. [interaktyvus] [žiūrėta 2016-05-12]. Prieiga per internetą: <http://ardupilot.org/dev/docs/odroid-via-mavlink.html>
25. C. Je and H.-M. Park. Registration. Signal Processing: Image Communication, 2013