

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
MATEMATINĖS INFORMATIKOS KATEDRA

**Lukas Klusis**

Informatikos studijų programa  
Matematinės informatikos šaka

## **Bepiločių orlaivių vietas nustatymas remiantis vaizdine informacija**

**Vision-Based Localization for Unmanned Aerial Vehicles**

Bakalauro baigiamasis darbas

Vadovas:

lekt. **Mindaugas Eglinskas**

*parašas*

Vilnius 2015

# Turinys

<b>Įvadas</b>	<b>1</b>
<b>1 Apžvalga</b>	<b>4</b>
1.1 Inercinė navigacija	4
1.1.1 Vizuali odometrija	4
1.2 Žemėlapio sudarymas	5
1.3 Paveikslėlių registracija	6
1.3.1 Ypatybių atitikmens būdas	7
1.3.2 Kontūrų atitikmenų būdas	7
1.3.3 Šablono atitikmenų būdas	8
1.4 Lokalizacija	8
1.4.1 Bayes filtras	11
1.4.2 Dalelių filtras	12
1.4.3 Išplėstasis Kalman filtras	13
<b>2 Principiniai sprendimai</b>	<b>15</b>
2.1 Karkasas	15
2.2 Paveikslėlių registracija	16
2.2.1 Mastelio suvienodinimas	16
2.2.2 Klaidų mažinimas	17
2.2.3 Optimizavimas	17
2.3 Lokalizacija	18
<b>3 Rezultatai</b>	<b>20</b>
3.1 Naudojama įranga	20
3.2 Prielaidos ir apribojimai	20
3.3 Paveikslėlių registracija	21
3.4 Lokalizacija	23
<b>4 Išvados</b>	<b>24</b>
4.1 Apribojimai ir galimybės	25
<b>Terminai ir trumpiniai</b>	<b>26</b>
<b>Literatūros sąrašas</b>	<b>28</b>
<b>Santrauka</b>	<b>30</b>
<b>Summary</b>	<b>31</b>

# Ivadas

Bepiločiai orlaiviai (BPO) yra aktyviai naudojami įvairiose srityse ir tikimasi, kad jų panaudojimas stipriai plėsis. Ši technologija yra naudojama tokiose srityse kaip: kelių ar kartografinių žemėlapių sudarymas [11], žmonių paieškos ir gelbėjimo operacijos [8], karinė žvalgyba ar kovinės operacijos, ūkinės paskirties laukų apžiūra bei priežiūra ir daugelyje kitų sričių.

## Motyvacija

Praktiškai visu laikotarpiu, kada naudojama aviacija, egzistavo poreikis bepiločiam skrydžiui. Bepilotis orlaivis yra suprantamas kaip orlaivis, kuris neturi skraidinti piloto ir gali būti valdomas nuotoliniu būdu arba veikti autonomiškai. Dažniausiai bepiločius orlaivius yra siekiama panaudoti kai yra paliečiamas bent vienas iš šių kriterijų:

- Rizikingumas - tai užduotys, kuomet yra didelė orlaivio sudužimo ar numušimo tikimybė, kaip pavyzdžiui, karinės užduotys ar skrydžiai virš aktyvių ugnikalnių. Taip atvejais, kai yra pavojuς žmogaus sveikatai, pavyzdžiui virš užterštų, kaip pavyzdžiui, skrydžiai virš radioaktyvių ar chemiškai pavojingų zonų.
- Pasikartojančių žmogaus veiksmų eliminavimas - tai skrydžiai, kurių užduotys gali būti automatizuotos, pavyzdžiui, skrydis paskui objektą, paskirtos vietas fotografovimas.
- Ekonomiškumas - orlaivis gebantis gabenti žmogų iš princiopo negali būti mažas, dėl to ir pigus.
- Ilgos kelionės - ryškiausi pavyzdžiai būtų kosminiai orlaiviai ar orlaiviai, kurie kaip įmanoma ilgiau skleidžia žemės atmosferoje ir siunčia interneto signalą.

Anksčiausiai piloto nereikalaujantys skrydžiai buvo panaudoti kariniais tikslais. Išskirtiniu įvykiu galima išskirti Austrijos puolimą 1849 m., kuomet buvo puolamas Italijos miestas Venecija, balionais gabenančiais bombas. Visgi toks skrydis buvo menkai kontroliuojamas, todėl būtina buvo atrasti metodus, kaip valdyti ir aiškiau numatyti bepiločio orlaivio skrydžio rezultatus. Didžiulis proveržis buvo padarytas su radio bangų atradimu ir bepiločių orlaivių kūrimu prasidėjusi nuo Pirmojo Pasaulinio karo. Kitas svarbus etapas buvo pradėtas su kompiuterių technologine pažanga bei dirbtinio intelekto mokslų srities populiarėjimu. Vienas iš iškeltų tikslų yra įgalinti autonominius robotus, įskaitant ir bepiločius orlaivius, veikti kiek įmanoma labiau autonomiškai. Kartu su BPO autonomiškumo siekimu svarbus elementas tapo gebėjimas patiems orlaiviams nustatyti savo padėtį.

Egzistuoja dvi padėties nustatymo klasifikacijos. Pirmoji, kai orlaivis nustato savo vietą lokaliai, t. y. orlaivio pozicija yra skaičiuojama nuo lokalaus, dažniausiai tik šiam orlaiviu žinomo atskaitos taško, pavyzdžiui nuo skrydžio pradžios taško. Antrasis variantas yra, kuomet orlaivis savo vietą susieja su globaliu, standartiniu atskaitos tašku, pavyzdžiui, žemės centru.

## Problemos aprašymas

Daugelyje globalaus pozicionavimo reikalaujančiuose sprendimuose yra naudojama [globali pozicionavimo sistema \(GPS\)](#). Tai yra Jungtinių Amerikos Valstijų (JAV) Saugumo Departamento sukurta sistema. Sistema pradėta kurti 1973 m., o pilnai veikia nuo 1995 m. ir susideda iš dviejų pagrindinių dalių: 24 aplink žemę skriejančių dirbtinių palydovų ir žemės paviršiuje esančių GPS imtuvų, kurie iš gaunamų palydovų signalų geba apskaičiuoti savo poziciją. Nors egzistuoja GPS alternatyvų, kaip Rusijos Globali Navigacinių Palydovų Sistema (GLONASS) ar planuojama Europos Sajungos Galileo pozicionavimo sistema, tačiau šių sistemų veikimo principas yra analogiškas GPS todėl šio darbo metu, paprastumo dėlei yra vertinama tik GPS įtaka.

Būtinybę bepiločiams orlaiviams tureti pastovų kontaktą su GPS palydovais yra didelis trūkumas urbanistinėse vietovėse, kur, dėl pastatų blokuojamo GPS signalo, teikiama paklaida gali būti didesnė negu keliausdešimt metrų [14]. Ne retu atveju, tokio dydžio paklaida, gali būti lemiamas veiksny s apribojantis bepiločio orlaivio panaudojimą. Iš kitos pusės GPS sistemą negalima laikyti patikima [2], naudojant sąlyginai nebrangius prietaisus įmanoma „užteršti“ GPS signalo kanalą, taip niekados neleidžiant prietaisui nustatyti savo pozicijos ar net apgauti GPS imtuvus, taip kad teikiami rezultatai būtų parinkti kito asmens. Toks navigacinės sistemos pažeidžiamumas yra aktuali problema konfliktinėse zonose, stipriai apribojanti bepiločių orlaivių galimybes. Egzistuoja ir kitos priežastys, dėl kurių verta kurti nepriklausomą navigacinię sistemą. Po 2001 m. rugsėjo 11-tosios įvykių, kuomet teroristai įvykdė išpuolių prieš JAV, buvo paskatintos diskusijos, dėl visos GPS sistemos nenutrukstamo prieinamumo katastrofinių situacijų metu. Galima daryti tik vieną išvadą - negalima garantuoti, kad GPS palydovai, o kartu ir jų siunčiamas signalas visada bus pasiekimas.

Nagrinėjant toliau, GPS sistema šiuo metu yra galima tik žemės planetoje, dėl to reikia toliau ieškoti naujų navigacijos būdų, kuriais remiantis būtų galima naviguoti kituose kosminiuose objektuose - planetose, palydovuose, kometose.

## Vaizdo medžiaga pagrįsti sprendimai

Suvokiant visus šiuos GPS sistemos trūkumus yra ieškoma būdų, kaip nustatyti orlaivio vietą nesinaudojant jokiais papildomais išoriniais davikliais. Vienas iš tokų variantų yra vietas nustatymas, naudojant vaizdo kamerą. Tai yra tapusi viena iš aktualiausių tyrimų sferų bepiločių orlaivių ir visos robotikos srityje. Vaizdo kamera yra labai patraukus pasirinkimas dėl pigios kainos bei vis didėjančių tyrimų vaizdo paremtuose pozicionavimo sprendimuose, kurie įgijo didelį dėmesį, kuomet tapo įmanoma skaičiavimo galimybes perkelti į patį orlaivį. Galima išskirti daugiausiai dėmesio sulaukusius sprendimus. Pirmiausia, tai yra vizuali odometrija, kuomet roboto pozicija ir orientacija yra nustatoma analizuojant gretimus kameros paveikslėlius. Šis metodas yra pritaikytas tokiose sudėtingose operacijose, kaip Marsaeigiu vietos nustatymui. Kita didelio dėmesio sulaukusi sritis yra dviejų vaizdų paveikslėlių susiejimas arba kitaip tariant [paveikslėlių registraciją](#). Tai ypač aktualu vienos

nustatymui, turint mintyje, kad yra daug viešų servisų, kaip *Microsoft Bing Maps* ar *Google Maps*, kurie teikia pasaulio žemėlapius su žinomomis koordinatėmis.

## Tikslas ir uždaviniai

Šio darbo tikslas yra sukurti sistemą - grupę susijusių ir kartu dirbančių dalių - kuri, remiantis vaizdo kameros ir paveikslėlių, kurių žinoma geografinė padėtis, registracija, gebėtų nustatyti orlaivio poziciją. Šiam tikslui įgyvendinti yra apžvelgama literatūra, parenkami gali sprendimo būdai bei šie būdai yra eksperimentiškai patikrinami.

## Panašūs darbai

Yra atliktų kitų, panašių darbų. Vienas tokių yra [18]. Darbe yra siūloma navigacinė sistema skirta BPO, kuri gali sekti judantį objektą be GPS sistemos. Tai yra atliekama analizuojant vaizdo kadru seką ir nustatant sekamo objekto pikselių koordinates, kuriomis vadovaujantis yra toliau naviguojančios orlaivij paskui šį objektą. Kitų bandymų metu [5] yra siūloma navigacinė sistema, kuri apjungia vizualią odometriją ir žemėlapį, kurių žinoma geografinė padėtis, registraciją su gaunamu vaizdu. Atlikus registracijai gauti duomenys yra filtruojami naudojant taškų masės filtrą bei galiausiai duomenys sujungiami naudojant Kalman filtrą, tokiu būdu, kaip galutinis rezultatas, yra apskaičiuojamos koordinatės [WGS-84](#) koordinačių sistemoje.

# 1 Apžvalga

## 1.1 Inercinė navigacija

Vienas iš pagrindinių navigacinių sistemos komponentų yra inercinė navigacinė sistema (INS). INS yra rinkinys pasyvių jutiklių, kuriuos dažniausiai sudaro [akcelerometras](#) ir [giroskopas](#), kurie teikia tokius duomenis kaip greitis, posūkio kampus. INS duomenis skaičiuoja remdamasi savo paties pateiktais ankstesniais duomenimis, todėl, kad ir nedidelės paklaidos, laikui bėgant ji vis didėja, Tai yra pagrindinė priežastis, kodėl INS dažniausiai nėra naudojama kaip viena sistema, kartu naudojami papildomi jutikliai kaip barometras orlaivio aukščio patikslinimui, kompasas posūkio kampo patikslinimui, bei kiti jutikliai ar naudojama vizuali odometrija tiek greičio, tiek posūkio kampo nustatymui.

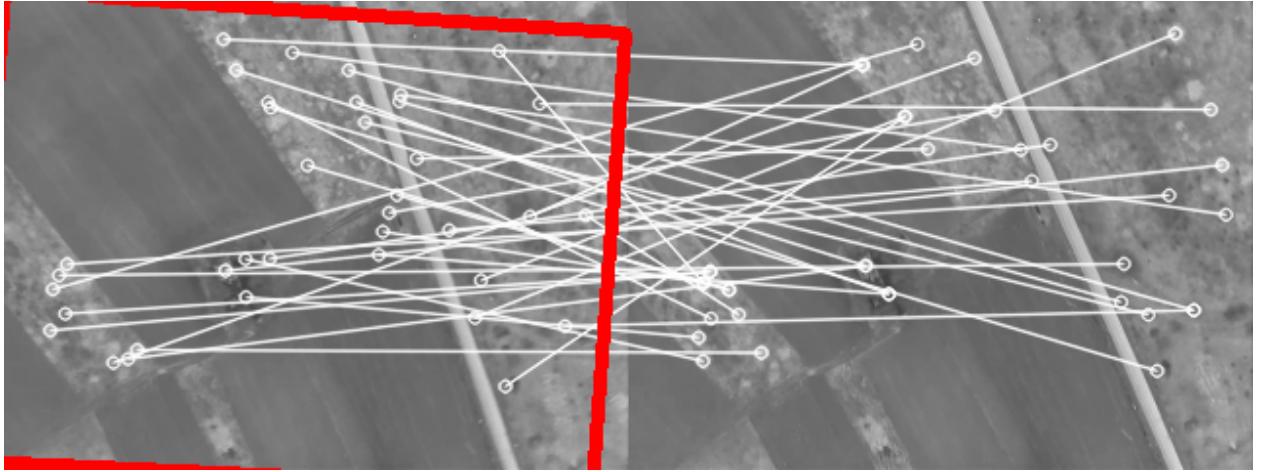
### 1.1.1 Vizuali odometrija

Vizuali odometrija yra tapusi populiaria tema, atlikti bandymai rodo, kaip ją praktiskai galima pritaikyti [BPO](#) vietos nustatymui [7]. Šio darbo metu pilnas vizualios odometrijos funkcionalumas nebus realizuotas, nors tai didelė ir perspektyvi sritis tobulinimui ateityje. Šio darbo vienas iš svarbiausių veiksnių, nuo kurio labai stipriai priklauso vaizdu paremtos vietos nustatymas yra žinojimas, kokiui kampu pasisuko orlaivis dvimatėje erdvėje per laiką  $t$ .

Šiam poreikiui įgyvendinti galima remtis tiek inercinės navigacijos teikiamais duomenimis, tiek išsiskaičiuoti kampą remiantis vizualios odometrijos principais. Būtent pastarasis variantas čia aprašomas. Turint paveikslėlj, kuris gautas esamu skrydžio metu ir paveikslėlj, kuris buvo užfiksuotas prieš laiką  $t$  galima nustatyti [Homografinę matricą](#), kuri sieja šiuos paveikslėlius, svarbu užtikrinti, kad laikas  $t$  nebūtų per didelis ir paveikslėliai turėtų pakankamai dideli kiekj bendrų taškų. Pavyzdys tokio sulyginimo yra pavaizduotas iliustracijoje 1. Tai galima padaryti atlikus šiuos žingsnius:

1. Aptinkamos paveikslėlių ypatybės abiejuose vaizduose. Tai gali būti įvairūs taškai, linijos, kampai ar kiti objektais. Dažniausiai naudojami [SURF](#), [SIFT](#) ar kiti algoritmai.
2. Ištraukti ir aprašyti paveikslėlių ypatybes, tai galima padaryti naudojant [SURF](#), [SIFT](#), [FREAK](#) ar kitus algoritmus.
3. Sugrupuoti ištrauktas paveikslėlių ypatybes, tai dažniausiai daroma brutalios jėgos arba [FLANN](#) algoritmu.
4. Sugrupuotų paveikslėlių ypatybės gali turėtų pakankamai daug neteisingai sugrupuotų taškų, todėl galiausiai yra pritaikomas [RANSAC](#) algoritmas, kuris išfiltruoja taškus – išskirtis ir apskaičiuojama [homografinė matrica](#).

Turint homografinę matricą galima ją išskaidyti į posūkių, postūmių matricas ir normalės vektorių [12].



Illiustracija 1: Homografinės matricos suradimas tarp kadrių, kuriuos skiria nedidelis laiko tarptas  $t$ .

## 1.2 Žemėlapio sudarymas

Konstruojant žemėlapį, yra naudojama [Mercator projekcija](#) projekcija, bei daroma, priešlaida, kad žemė yra tobulo rutulio formos. Taip yra supaprastinamas žemėlapį konstravimas bei jo manipuliavimas, tačiau naudojant ilgumos ir platumos koordinates yra daroma priešlaida, kad jos yra [WGS-84](#) standarto.

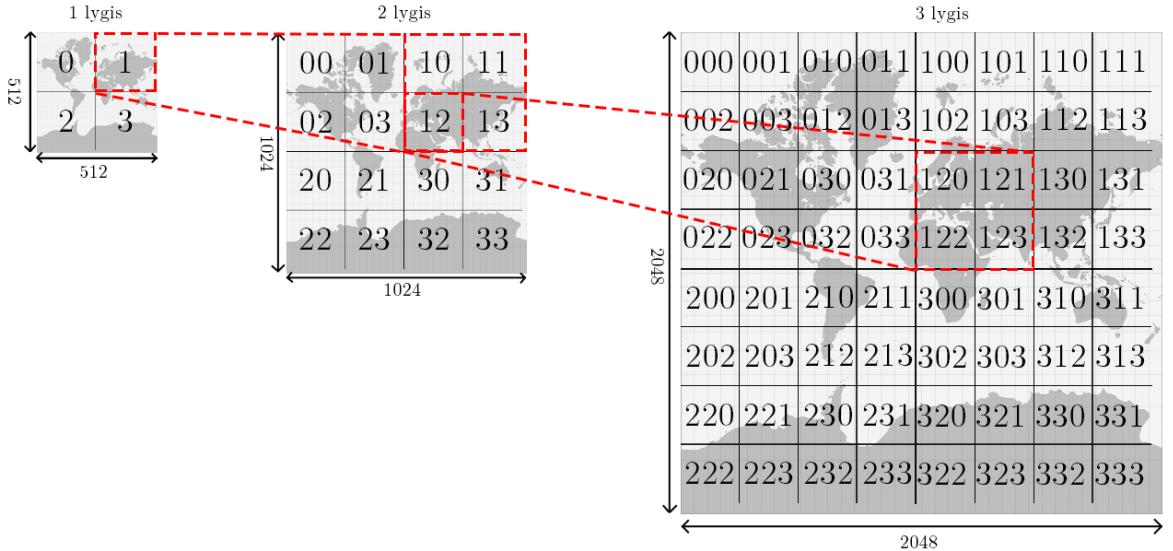
Visas pasaulio žemėlapis yra suskaidomas į  $256 \times 256$  pikselių kvadratelius, o jų kiekis priklauso nuo žemėlapio detalumo lygio. Detalumo lygis vienareikšmiškai apibrėžia kokio pločio ir aukščio (pikseliai) yra žemėlapis:

$$\text{žemėlapio dydis} = \text{žemėlapio aukštis} = \text{žemėlapio plotis} = 256 \times 2^{\text{detalumo lygis}} \text{ (pikselių)} \quad (1)$$

$$\text{žemėlapio rezoliucija} = \cos\left(\frac{\text{platuma} \times \Pi}{180}\right) \times \frac{\text{denžemės spindulys}}{\text{žemėlapio dydis}} \left( \frac{\text{metrai}}{\text{pikseliųje}} \right) \quad (2)$$

Žemėlapio pikseliai yra skaičiuojami nuo viršutinio, kairio kampo  $(0,0)$  iki apatinio dešinio kampo (*aukštis, plotis*)

Kiekvienam žemėlapio kvadrateliui yra suteikiamas kodas sudarytas iš 4 simbolių abécélės: *0, 1, 2, ir 3*. Kodas sudaromas tokia tvarka: pirmo detalumo lygio žemėlapis padalijamas į 4 dalis, joms paeiliui priskiriami simboliai iš abécélės, gaunami pirmo lygio kvadratelių raktų kodai. Kiekvieną iš gautų 4 kvadratelių vėl dalijame į 4 lygius dalis ir kiekvienai iš šių dalių paeiliui prie esamo kodo pridedame simbolį iš abécélės, taip gausime aukštesnio detalumo lygio kvadratelių kodus. Kartojant šiuos veiksmus galime sudaryti kaip norimai detalaus žemėlapio kvadratelių raktų kodus. Kodų sudarymas yra pavaizduotas iliustracijoje 2. Šis kodas pasižymi gera savybė - kodo ilgis yra lygus žemėlapio detalumo lygiui.



Illiustracija 2: Žemėlapio kvadratelių kodų sudarymas

Turimą ilgumą ir platumą galima konvertuoti žemėlapio pikselių koordinates:

$$\text{pikselio x koordinatė} = \frac{\text{ilguma} + 180}{360} \times \text{žemėlapio dydis} \quad (3)$$

$$\text{pikselio y koordinatė} = 0,5 - \log \left( \frac{1 + \sin(\text{platum}a \times \frac{\pi}{180})}{1 - \sin(\text{platum}a \times \frac{\pi}{180})} \right) \times \frac{1}{4\pi} \times \text{žemėlapio dydis} \quad (4)$$

Taigi žinant vieną iš šių variantų galima lengvai ir vienareikšmiškai apibrėžti žemėlapio poziciją bei apskaičiuoti likusius variantus:

- (a) kvadratėlio rakto kodą ir lokalias pikselio koordinates tame kvadratėlyje,
- (b) globalaus (viso pasaulio žemėlapio) pikselio koordinates ir detalumo lygi,
- (c) ilgumos ir platumos koordinates bei detalumo lygi.

### 1.3 Paveikslėlių registracija

Paveikslėlių registracija, tai procesas, kurios metu yra surandamas geometrinių transformacijų sąryšis tarp dviejų paveikslėlių duomenų poaibiu.

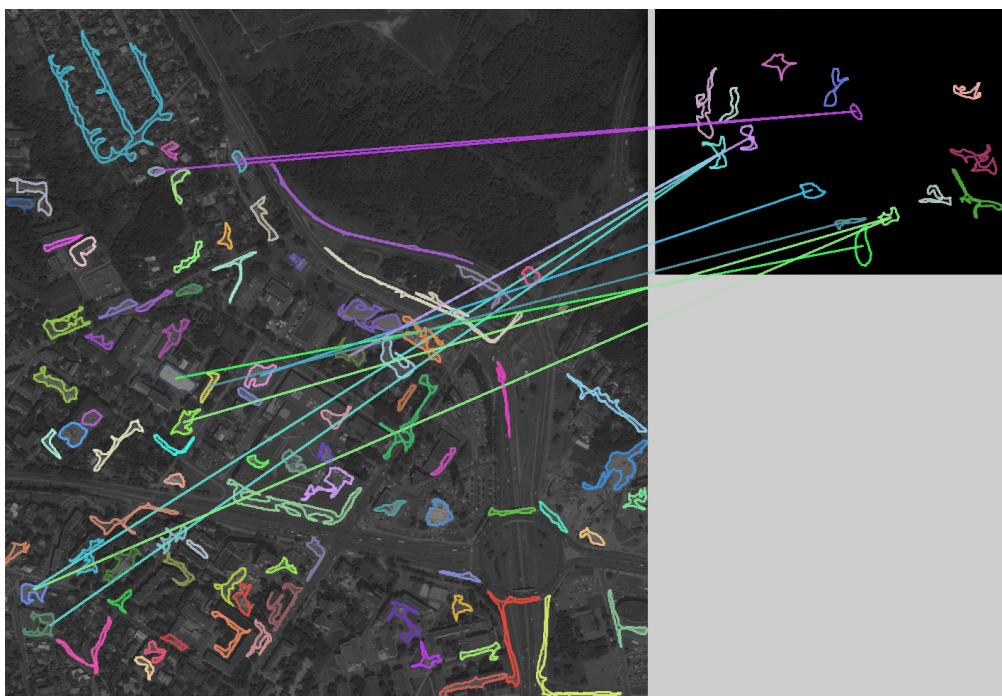
Tariama, kad **Scena** ir **Modelis** yra du skirtingi paveikslėliai, užfiksuoti skirtingu laiku, kameros pozicija, posūkio kampu bei pačia kamera, tačiau fiksuojama vieta fiziškai arti, iš tiesų, scenos geografinė apréptis yra didesnė negu modelio.

Egzistuoja daug sprendimo variantų, kaip scenoje atrasti modelį. Visi jie skirtingai reaguoja į scenos ir modelio mastelio, posūkio kampų, spalvų, jutiklių (kameros) skirtumus, taip pat jų skaičiavimo trukmė yra labai skirtinga.

### 1.3.1 Ypatybių atitinkmens būdas

Vienas iš paprasčiausių variantų yra pritaikyti SIFT ar kitus paveikslėlio ypatybių išskyrimo ir sugrupavimo algoritmus, kurie jau buvo aprašyti skyriuje [Vizuali odometrija](#). Šis algoritmas nėra priklausomas nuo paveikslėlių mastelių, pasukimo kampų, yra sąlyginai greitas, tačiau šis algoritmas yra ganétinai jautrus apšvietimo pasikeitimui, skirtingų jutiklių naudojimui.

### 1.3.2 Kontūrų atitinkmenų būdas



Iliustracija 3: Scenos (kairėje) ir modelio (dešinėje) kontūrų grupavimas

Kitas būdas yra ryškiausiu kontūru išskyrimas ir jų lyginimas. Kontūru vadiname aibę taškų, kurie paeiliui apibréžia kreivę. Ši kreivė dažniausia yra padaroma iš objekto kampų ir linijų, tai dažniausiai yra objekto paveikslėlyje ribos. Paieškos globaliamame žemėlapyje atveju yra tariama, kad tiek scenoje, tiek modelyje egzistuoja diskrečių, pakankamai didelių objektų, pavyzdžiui: namas, ežeras, sala, aikštė ar panašus objektais, kurie gali būti apibréžti kontūru. Ištraukus apibréztus kontūrus tiek apie scenos, tiek apie modelio ryškiausius objektus, reikia juos sugrupuoti, grupavimo pavyzdys pavaizduotas iliustracijoje 3. Grupavimo būdai yra aprašyti darbuose [1, 6, 17]. Atlikus kontūrų grupavimą, reikia nustatyti modelio padėti scenoje, tam naudojamas [RANSAC](#) algoritmas. Sulyginti scenos ir modelio mastelius, galima papildomai vertinti kontūrus pagal plotą ar perimetram.

### 1.3.3 Šablono atitikmenų būdas

Šablono atitikmenų paieška (arba šablono poravimas) yra technika, kuria ieškoma dalies paveikslėlio (modelo) didesniame paveikslėlyje (scenoje). Šablono poravimas yra atliekamas modelio paveikslėlių paeiliui lyginant su scenos paveikslėlio dalimi. Pradedant nuo viršutinio kairio kampo yra paskaičiuojama įvertis, kiek scenos dalis, prasidedanti nuo šio taško sutampa su modeliu. Toks įvertis yra paskaičiuojamas kiekvienam scenos taške „slenkant“ po vieną pikselį į šoną ar apačią, kol tik įmanoma sudaryti modelio dydžio paveikslėlio poaibį. Atitikimo įvertinimą galima skaičiuoti keletą metodų, šio darbo metu yra naudojamas normalizuotos kryžminės koreliacijos metodas, kurį galima aprašyti lygtimi:

$$C(x, y) = \frac{\sum_{(x_1, y_1) \in M} (M(x_1, y_1) \cdot S(x + x_1, y + y_1))}{\sqrt{\sum_{(x_1, y_1) \in M} M(x_1, y_1)^2 \cdot \sum_{(x_1, y_1) \in M} S(x + x_1, y + y_1)^2}} \quad (5)$$

čia  $M$  yra modelio, o  $S$  scenos paveikslėliai. Suskaičiavus, kiekvienam galimam taškui jo atitikimo įvertį, gausime  $(w_s - w_m + 1) \times (h_s - h_m + 1)$  (čia  $w_s$  ir  $h_s$  bei  $w_m$  ir  $h_m$  yra atitinkamai scenos ir modelio pločiai bei aukščiai) taškų, kurių vertės yra nuo 0 iki 1, šių taškų aibę žymime

$$C = \{ \langle x, y, C(x, y) \rangle \mid 0 \leq x \leq w_s - w_m + 1, 0 \leq y \leq h_s - h_m + 1 \} \quad (6)$$

Svarbiausias yra taškas, kurio vertė didžiausia, pažymėkime jį  $C_{max}$  tačiau dėl per laiką pakitusio žemės reljefo, apšvietimo sąlygų ar kitų veiksnių, dažnai negalima remtis tik didžiausio taško suradimu, todėl yra įvedamas parametras  $c^- \in (0; 1]$ , kuris atlieka slenksčio funkciją, t.y. iš aibės  $C$  išrenkame taškus

$$F = \{ \langle x, y, c \rangle \mid \langle x, y, c \rangle \in C \wedge c \geq c^- \cdot C_{max} \wedge (x, y) \text{ yra lokaus maksimumo taškas} \} \quad (7)$$

Kitaip tariant, aibės  $F$  elementus vadinsime šablono poravimo kandidatais, t. y. taškai scenoje ir jų įvertis  $c \in [0; 1]$  kiek tiksliai šablonas atitinka scenos vietą. Kadangi tikrasis atitikmuo gali būti ne daugiau negu 1 kartą, vadinasi aibėje  $F$  yra mažiausiai  $|F| - 1$  klaidingų sutapatinimų.

Aibės  $C$  taškus galima interpretuoti kaip atskirą paveikslėlį, iliustracijoje 9 pateiktas pavyzdys, šviesesni taškai reiškia stipresnį sutapimo įvertį, šį paveikslėlį vadinsime Atitikmenų paveikslėliu. Toks duomenų interpretavimas suteikia papildomų galimybių, pavyzdžiui klaidų mažinimui. Galima pritaikyti plačiai žinomus vaizdo apdorojimo algoritmus.

## 1.4 Lokalizacija

Šiame skyriuje yra apjungiamos visos sistemos dalys į vieną bendrą algoritmą. Apie vietas nustatymo principus ir algoritmus kalbama knygose [15, 16]. Šio darbo metu yra

naudojami dviejų tipų Bayes filtrai: Išplėstasis Kalman filtras (detaliau aprašomas darbe [13]) ir Dalelių filtras (detaliau aprašomas darbe [3]).

Lokalizacija stipriai remiasi tikimybinių skirstinių sudarymu bei aplinkos sąveikomis. Aplinkos parametrai yra apibrėžiami kaip  $b\bar{u}sena$ . Būsena galima suprasti kaip  $n - dimensijų$  kintamajį, kuris kinta laiko  $t$  atžvilgi ir kuriuo apibrėžiami orlaivio fizikiniai parametrai, pavyzdžiui vieta, pagreitis, kitaip tariant, būsena yra elementas iš aibės

$$\mathbb{S} = \{\langle s_1, s_2, \dots, s_n \rangle \mid i = \overline{1..n}, s_i \in S_i\} \quad (8)$$

kur kiekviena aibė  $S_i$  simbolizuoją būsenos dedamąją dalį, kurios dažnai būna:

- pozicija - tai dažniausiai viena, dvejom arba trimis dimensijomis apibrėžia orlaivio vietą sutartoje koordinačių sistemoje.
- greitis - tai viena, dvejom arba trimis dimensijomis apibrėžia kokiui greičiu kinta orlaivio pozicija sutartoje koordinačių sistemoje.
- orlaivio kampus - taip pat vienam dejom arba trimis dimensijomis apibrėžia kokiui kampu pasisukęs orlaivis.

Galima būsena apibrėžti ir daugeliu kitų būsenos parametrų.

Pagrindine idėja yra nustatyti orlaivio būseną remiantis jutiklių duomenimis. Jutiklių duomenys gali būti tik daliniai bei taip pat jų matavimai gali būti atlikti su tam tikra paklaida - triukšmu. Užuot stengiantis išmatuoti tikslią vienintelę būseną, ji yra skaičiuojama kaip tikimybinis būsenų skirstinys, kuris yra vadinas įsitikinimu, kuris aprašytas toliau (13).

Svarbūs standartiniai žymėjimai  $x_{t_1:t_2}$ ,  $z_{t_1:t_2}$ ,  $u_{t_1:t_2}$ , detalesnius jų aprašymus galima rasti knygoje [16](p. 16 - 32). Tai yra daugia-dimensiniai kintamieji, kurie šio darbo metu yra apibrėžti aibėje  $\mathbb{S}$ .

- būsenų aibė, kurioje yra orlaivis per laiką  $(t_1; t_2]$ ,  $(t_1 \leq t_2)$ :

$$x_{t_1:t_2} = x_{t_1}, x_{t_1+1}, \dots, x_{t_2} \quad (9)$$

- aibė matavimų duomenys, kurie sako, kokioje būsenoje yra orlaivis per laiką  $(t_1; t_2]$ ,  $(t_1 \leq t_2)$ :

$$z_{t_1:t_2} = z_{t_1}, z_{t_1+1}, \dots, z_{t_2} \quad (10)$$

- $u_{t_1:t_2}$  - kontroliniai, judesio duomenys. Tai yra informacija kaip pasikeitė būsena per laiką  $(t_1; t_2]$ ,  $(t_1 \leq t_2)$ :

$$u_{t_1:t_2} = u_{t_1}, u_{t_1+1}, \dots, u_{t_2} \quad (11)$$

Svarbios dvi tikimybinės lygtys:

- $p(z_t|x_t)$  - matavimo tikimybė. Dažnai gali nepriklausyti nuo laiko parametru  $t$ , t.y  $p(z_t|x_t) = p(z|x), \forall t$ . Kitaip tariant, tai orlaivio pozicija nustatyta, remiantis jutiklių informacija su triukšmu.
- $p(x_t|x_{t-1}, u_t)$  - būsenos pakitimo tikimybė. Ši tikimybė rodo, kur tikėtiniausia yra orlaivis laiko momentu  $t$ , žinant kad orlaivis atliko  $u_t$  veiksmus.

Vietos nustatymas nėra vienareikšmis procesas, kurio metu būtų galima garantuotai nusakyti orlaivio vietą. Orlaivis remiantis jutiklių duomenis gali įvertinti tik savo pozicijos tikimybę, kuri yra vadina *jsitikinimu*. Įsitikinimas dažniausiai yra reprezentuojamas tikimybinių skirstiniu. Šio darbo metu tokius įsitikinimus galima apibrėžti funkcija iš aibės:

$$\mathbb{B} = \left\{ bel \mid bel : \mathbb{S} \rightarrow [0; 1]^n, \sum_{x \in \mathbb{S}} bel(x) = 1 \right\} \quad (12)$$

čia funkcijos *bel* pavadinimas yra kilęs iš anglų žodžio *belief* - *jsitikinimas*.

Įsitikinimą, galima išreikšti tikimybine priklausomybe:

$$bel(x_t) = p(x_t|z_{1:t}, u_{1:t}) \quad (13)$$

Kas kitaip žodžiais reikštų, kokia tikimybė, kad robotas vietoje  $x_t$ , kai yra žinomi matavimų  $z_{1:t}$  ir atlirkę judesių  $u_{1:t}$  duomenys

Pagal šią lygtį (13) įsitikinimą galima apskaičiuoti tik žinant to laiko matavimų įvertinimą  $z_t$ . Dažnai yra naudinga apskaičiuoti įsitikinimą prieš atliekant matavimus ir ką tik atlikus judesius.

$$\overline{bel}(x_t) = p(x_t|z_{1:t-1}, u_{1:t}) \quad (14)$$

Tikimybinis skirstinys  $\overline{bel}$  dažnai yra vadinamas *prognozė*. O apskaičiavimas *bel* žinant  $\overline{bel}$  yra vadinamas *korekcija* arba *matavimo atnaujinimas*.

Kadangi dažniausiai įsitikimas būna pasiskirstęs pagal Gauso skirstinį, paprastumo dėlei darome prielaidą, kad įsitikinimas, negali turėti dviejų taškų, kurie yra maksimumo taškai. Tada tašką, kuriame įsitikinimas įgyja maksimalią reikšmę vadiname įsitikinimo pozicija:

$$pos(bel) = x \in \mathbb{S}, \text{ kur } bel(x) = \max_{x' \in \mathbb{S}} bel(x) \quad (15)$$

Yra keli variantai, kaip galima reprezentuoti roboto įsitikinimą - vienos hipotezės arba daugelio hipotezių įsitikinimu. Vienos hipotezės įsitikinimo atveju yra skaičiuojamas vienas unikalus taškas, ar tiksliau - įsitikinimas, žemėlapje. Šis principas remiasi sąlyga, kad iš gaunamų duomenų apskaičiuota pozicija duoda vienintelį rezultatą, t. y. nesukelia pozicijos neaiškumą. Toks būdas yra įmanomas turint jutiklius, kurių duomenys su paklaida yra pasiskirsčiusi pagal Gauso skirstinį, tokius jutiklio duomenis galima suvesti į vieną tašką, pavyzdžiui, inercinės navigacinės sistemos, ar vizualios odometrijos atvejais. Visgi vienos hipotezės atveju dėl jutiklių triukšmo ar judesių kontrolės netikslumą tokis duomenų sutrauki-

mas į vieną tašką tampa sudėtingu, turintis didelį neapibrėžtumą, kuris laikui bégant padaro galutinį rezultatą praktiskai nepanaudojama. Tuo labiau, tais atvejais, kai duomenys yra „šuoliujantys“, t. y. jutiklis gali pateikti rezultatus, kurie turi daugiau negu vieną lokalaus maksimumo tašką, kitaip tariant yra pasiskirstę ne pagal Gauso skirstinį, vienos hipotezės įsitikinimas tampa išvis neįmanomas. Pastaroji situacija yra kuriama ir paveikslėlių registracijos metu, todėl šio darbo metu, būtina remtis daugelio hipotezių įsitikinimu. Daugelio hipotezių atveju, be pilotis orlaivis seka, ne tik vieną, bet taip pat ir galimai begalinę aibę įsitikinimų:

$$Bel = \{bel_i \mid i = \overline{1...n}, n \in \mathbb{N} \wedge bel_i \in \mathbb{B}\} \quad (16)$$

Turint daug galimų hipotezių, labai svarbu nustatyti tam tikrą įsitikinimų rikiavimą. Tam tikslui konstruojama svorinė pozicijų aibė:

$$Bel_w = \{\langle bel_i, w_i \rangle \mid w_i \in [0, 1], bel_i \in Bel \forall i\} \quad (17)$$

T. y. įvertinant tam tikrus faktus galimoms pozicijoms yra suteikiama tikimybė. Kaip šios tikimybės suteikiamos bus aiškinamasi vėliau. Toks įsitikinimų suskirstymas remiasi dvejais teiginiais:

1. gali egzistuoti ne daugiau kaip vienas tikrasis įsitikinimas. Įsitikinimas vadinamas tikruoju (teisingu) jeigu jo pozicija yra nutolusi nuo fizinės vietas, kurioje egzistuoja orlaivis per sąlyginai nereikšmingą atstumą. Bandymų metu tokiu atstumu buvo laikyti 2 metrai. Visi kiti įsitikimai vadinami netikrais (neteisingais).
2. rekursyviai atjauninant įsitikinimą, tikimasi, kad gauti duomenys patvirtins tikrojo įsitikinimo poziciją (padidins įsitikinimo tikimybę), o netikrų įsitikinimų tikimybė bus mažinama.

#### 1.4.1 Bayes filtras

Bayes filtras, dar vadinamas rekursyviu Bayes filtru yra algoritmas, kuris yra tapęs, kaip karkasas daugeliui navigacijos praktikoje naudojamų algoritmų. Tai yra rekursyviai veikiantis algoritmas, kuris susideda iš dviejų esminių dalių:

1. Prognozė - Šiuo žingsniu yra apskaičiuojamas įsitikinimas  $\overline{bel}(x_t)$  laiko momentų  $t$ , kuomet yra žinomas įsitikinimas  $bel(x_{t-1})$  laiko momentu  $t-1$  bei atlikti kontroliniai (judesio) veiksmai  $u_t$ .
2. Taisymas - tai yra prognozės korekcija atsižvelgiant į matavimų tikimybę.

Dažniausiai Bayes filtras praktikoje nėra naudojamas, tačiau dvi šio filtro implementacijos - Kalman [15, 16] ir Dalelių filtrais [9] yra apžvelgiami toliau.

Šie algoritmai skiriasi, su kokiais tikimybiniais skirtiniai gali dirbti.

Kalman filtras yra ne kartą patvirtinės, kad yra patikimas algoritmas jutiklių duomenų filtravimui. Tačiau Kalman filtras gali būti pritaikomas, tik tada, kai matavimų duomenys yra pasiskirstę pagal Gauso skirstinį. Yra pasiūlyta [10] Kalman filtro variacijų duomenims pasiskirsčiusiems ne pagal Gauso skirstinį, tačiau reikalaujama, kad skirstinys būtų **vienamodalinis**. Priešingai šiai situacijai yra dalelių filtras, kuris gali dirbti su **multi-modaliniai** skirtiniais.

#### 1.4.2 Dalelių filtras

Visa dalelių filtro idėja yra reprezentuoti orlaivio poziciją aibe pavyzdžiu, pasiskirsčiusiu po turimą žemėlapį. Tada kiekvieną iš šių dalelių atnaujinti kartu su iš jutiklių gaunamais duomenimis ir įvertinus šių dalelių tikėtinumą, palikti tik tikėtinas daleles, taip apytiksliai nustatoma orlaivio pozicija žemėlapyje. Dalelių filtrą galima suskirstyti į 3 dalis: 1. Dalelių pozicijos „paslinkimą“, pagal jutiklių duomenis; 2. Dalelių pasvérimą; 3. Dalelių išrinkimą. Algoritmas detaliai pavaizduotas (1).

Dalelių filtro algoritme yra naudojama aibė mėginių - dalelių, kur kiekviena dalelė yra konkreti hipotezė teigianti, kad orlaivis yra nurodytoje vietoje:

$$\chi_t := \left\{ x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[M]} \right\} \subset \mathbb{S} \quad (18)$$

Daleles pirmai iteracijai sugeneruojame pasinaudojė šablonų poravime gautu aibės  $F$  (7) rezultatu, bei aplink šiuos taškus paskirstomos dalelės pagal Gauso skirstinį.

Eilutėje 6 yra nurodoma, kad turint dalelę  $x_t$  reikia sugeneruoti  $m$  naujų dalelių pagal būsenos pakeitimo tikimybę  $p(x_t | x_{t-1}, u_t)$ , kitaip tariant žinant ankstesnę būseną bei kokie veiksmai buvo atlikti, kuriamas prognozė kur orlaivis yra dabar. Tai dažnai yra duomenys iš tokių dalių kaip inercinė navigacinė sistema.

Kuo didesnis dalelių skaičius, tuo tiksliesnius rezultatus galima gauti, tačiau su didėjančiu dalelių sk. kartu auga ir skaičiavimo trukmė. Todėl kiek dalelių bus generuojama yra konfigūruojama, eilutėje 5 yra nurodomas maksimalus dalelių skaičius  $N_{max}$  ir parametras  $inc$ .  $N_{max}$  parametras nusako kiek maksimaliai dalelių gali būti sugeneruotų viso algoritmo vykdymo metu. Skaičius  $inc$  nurodo kiek maksimaliai iš kiekvienos dalelės gali būti suprojektuotą naujų dalelių, šis skaičius dažniausiai būna pakankamai mažas.

Kiekvienai naujai suprojektuotai dalelei yra suteikiamas svoris, tai atliekama eilutėje 8. Šie svoriai dažniausiai yra nurodomi jutiklių. Pasakius kitaip, šiuo žingsniu yra įvertinama, kokia yra tikimybė, kad orlaivis yra dalelės būsenoje, kai žinome, kokius duomenis pateikė jutikliai.

Suteikus dalelėms svorius reikia juos normalizuoti, t.y  $\sum_{(x_t, w_t) \in \chi_t} x_t$  turi būti lygi 1. Šiam tikslui eilutėje 14 kiekvienas dalelės svoris yra padalijamas ir normalizacijos koeficiente  $c$ .

Paskutinis žingsnis, kurį reikia padaryti tai atsižvelgiant į dalelių svorius iš jų

```

1: Duomenys: Aibė dalelių laiko momentu  $t - 1$ :  $\chi_{t-1}$ , judesio duomenys:  $u_t$ ,  

   matavimų duomenys:  $z_t$ 
2: Rezultatas: Aibė dalelių laiko momentu  $t$ :  $\chi_t$ 
3:  $\overline{\chi}_t = \chi_t = \emptyset$ ;
4: foreach dalelė  $x_{t-1} \in \chi_{t-1}$  do
   | // 1. Dalelių paslinkimas:
5:   |  $m = \min \left\{ \frac{|\chi_{t-1}|}{N_{max}}, \text{Random}(1, inc) \right\}$  ;
6:   | parinkti pavyzdžius  $\overline{\chi}_t' = \{x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[m]}\} \sim p(x_t | x_{t-1}, u_t)$  ;
   | // 2. Nustatomi dalelių svoriai:
7:   | foreach dalelė  $x_t \in \overline{\chi}_t'$  do
8:     | |  $w_t = p(z_t | x_t)$  ;
9:     | |  $\overline{\chi}_t = \overline{\chi}_t + \{x_t, w_t\}$ 
10:    | end
11:   | end
12:   |  $c = \sum_{(x_t, w_t) \in \overline{\chi}_t} x_t$  ;
13:   | foreach elementui  $(x_t, w_t) \in \overline{\chi}_t$  do
14:     | |  $w_t = \frac{w_t}{c}$  ;
15:   | end
   | // 3. Dalelių parinkimas:
16:   |  $\hat{N}_{eff} = \frac{1}{\sum_{(x_t, w_t) \in \overline{\chi}_t} w_t^2}$  ;
17: Iš aibės  $\overline{\chi}_t$  atsitiktinai išrenkama  $\hat{N}_{eff}$  skaičius elementų  $x_t \in \overline{\chi}_t$  į aibę  $\chi_t$ , kur  

   kiekvieno elemento  $x_t$  tikimybė patekti į aibę  $\chi_t$  yra  $w_t$  ;
18: Gražiname aibę  $\chi_t$ ;

```

### Algoritmas 1: Dalelių filtras

išrinkti apytiksliai  $N_t$  dalelių, tai atliekama eilutėse 16, 17. Tai vadinama *méginių émimu*. Darbe [9] yra apskaičiuotas efektyviausias méginių skaičius  $N_t = \hat{N}_{eff} = \frac{1}{\sum_{(x_t, w_t) \in \overline{\chi}_t} w_t^2}$ . Šiam koeficientui galioja sąlyga  $1 \leq \hat{N}_{eff} \leq |\overline{\chi}_t|$  naudojant šį koeficientą bus imama daugiau méginių, kai dalelių svoriai yra pasiskirstę tolygiai ir atvirkščiai - mažiau dalelių, kai egzistuoja išskiriančiu savo svoriu dalelių.

Gautas rezultatas yra reprezentuoojamas dalelių spiečiais, t. y. grupėmis dalelių, kurios reiškia įsitikinimą *bel*.

#### 1.4.3 Išplėstasis Kalman filtras

Kalman filtras yra taip pat vienas iš Bayes filtro variacijų. Kalman filtras reprezentuoja įsitikinimą tam tikru momentu. Įsitikinimas yra reprezentuoojamas parametrais: vidurkiu  $\mu_t$  ir kovariacija  $\sum_t$ . Šio darbo metu yra naudojamas EKF, bei kadangi EKF yra Bayes filtro implementacija, EKF taip pat galima suskirstyti į dvejas pagrindines dalis: prognozė ir taisymas.

EKF tikslas yra išspręsti pagrindinį standartinio Kalman filtro trūkumą - priklausomybę nuo tiesinių judėjimo ir matavimų pakitimų. EKF išsprendžia šią problemą įvesdamas dvi netiesines funkcijas  $g$  ir  $h$ .

- funkcija  $g$  apibrėžia kiek judėjimo pakitimą.

$x_t = g(u_t, x_{t-1}) + \epsilon_t$ , čia  $\epsilon_t$  yra atsitiktinis Gauso vektorius, kuris dar vadinamas Gauso triukšmu, šio vektoriau vidurkis yra lygus 0, o kovariacijos yra žymima  $R_t$ .

- funkcija  $H$  apibrėžia matavimų pakitimą.

$z_t = h(x_t) + \delta_t$ , čia  $\delta_t$  yra analogija  $\epsilon_t$  - tai atsitiktinis vektorius generuojantis matavimo triukšmą,  $\delta_t$  vidurkis yra lygus 0, o jo kovariacijos yra žymima  $Q_t$ .

EKF „ištiesina“ netiesine funkcijas, tam tikslui naudodamas pirmos eilės *Taylor eilučių* išplėtimu, kuriame yra naudojama dalinė išvestinė:

$$g'(u_t, x_{t-1}) := \frac{\partial g(u_t, x_{t-1})}{\partial x_{t-1}} \quad (19)$$

funkcija  $g$  yra aproksimuojama jos reikšme taške  $\mu_{t-1}$

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + \underbrace{g'(u_t, \mu_{t-1})(x_{t-1} - \mu_{t-1})}_{=:G_t} \quad (20)$$

Čia  $G_t$  yra *Jakobiano*  $n \times n$  matrica ir yra priklausoma nuo  $u_t$  ir  $u_{t-1}$  dėl to ji skirtingai, skirtiniems laiko momentams. Analogiskai funkcija  $h$  aproksimuojama taške  $\bar{\mu}_t$ :

$$h(x_t) \approx h(\bar{\mu}_t) + \underbrace{h'(\bar{\mu}_t)(x_t - \bar{\mu}_t)}_{=:H_t} \quad (21)$$

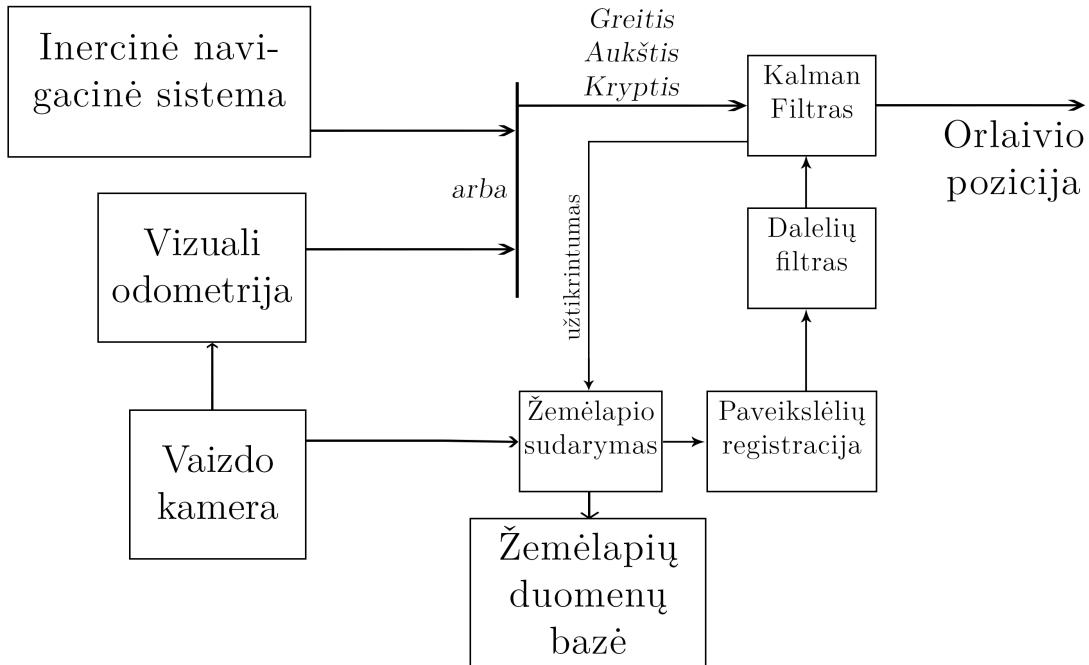
Galiausiai EKF galima aprašyti algoritmu:

- 1: **Duomenys:**  $\mu_{t-1}, \sum_{t-1}, u_t, z_t$
- 2: **Rezultatas:**  $\mu_t, \sum_t$   
// Spėjimas žingniai:  
// Suprojektuojame (prognozuojame) būse laiku  $t$
- 3:  $\bar{\mu}_t = g(u_t, \mu_{t-1});$   
// Suprojektuojame (prognozuojame) klaidos kovariaciją laiku  $t$
- 4:  $\sum_t = G_t \sum_{t-1} G_t^T + R_t;$   
// Taisymo žingniai:  
// Suskaičiuojame (Kalman) prieaugį - kiek norima pakeisti paskaičiuotą būseną
- 5:  $K_t = \sum_t H_t^T (H_t \sum_t H_t^T + Q_t)^{-1};$   
// Atnaujinti paskaičiuotą būseną tam tikru pamatavimu
- 6:  $\mu = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t));$   
// Atnaujinti klaidos kovariaciją
- 7:  $\sum_t = (I - K_t H_t) \sum_t;$

**Algoritmas 2:** išplėstasis Kalman filtras

## 2 Principiniai sprendimai

### 2.1 Karkasas



Iliustracija 4: Vaizdine medžiaga pagrįstos navigacinės sistemos dalys

Šio darbo metu nagrinėjama navigacinė sistema, yra sudaryta iš pagrindinių dalių pavaizduotų iliustracijoje 4. Kiekvienam skrydžiui būtina pasiruošti iš anksto ir paruošti žemėlapių duomenų bazę, apie kurią detaliau kalbama skyriuje [Žemėlapio sudarymas](#). Žemėlapių duomenų bazę būtų galima konstruoti skrydžio metu, internetu prisijungus prie išorinės duomenų bazės, kur tai yra įmanoma, tačiau siekiama sukurti bepilotį orlaivį kiek galima labiau autonomišką ir nepriklausomą, todėl apie tai plačiau nebus kalbama šio darbo metu. Duomenys gaunami iš vaizdo kameros gali būti panaudojami dvejopai, pirmiausia duomenys apdorojami vizualios odometrijos pasirinktinai gali pakeisti tam tikras arba visas reikalingas inercinės navigacinės sistemos dalis, apie tai buvo kalbama skyriuje [Inercinė navigacija](#). Tačiau tiek vizuali odometrija, tiek inercinė navigacija pateikia rezultatus su tam tikra paklaidą, todėl parinkus skirtingus vaizdo kadrus gaunamus iš vaizdo kameros yra taikomas [Paveikslėlių registracija](#) ir atliekama paieška žemėlapyje, kurio koordinatės yra žinomos. Gaunama informacija yra naudojama konstruoti galimas vietų sekas - hipotezes, kur gali būti orlaivis. Šios hipotezės yra filtruojamos dalelių filtro. Iteratyviai atlikus vis daugiau palyginimų su geografiniu žemėlapiu iš šių hipotezių ima ryškėti viena hipotezė, kuri galiausiai yra pripažystama kaip tikroji ir šie rezultatai parodo orlaivio poziciją.

## 2.2 Paveikslėlių registracija

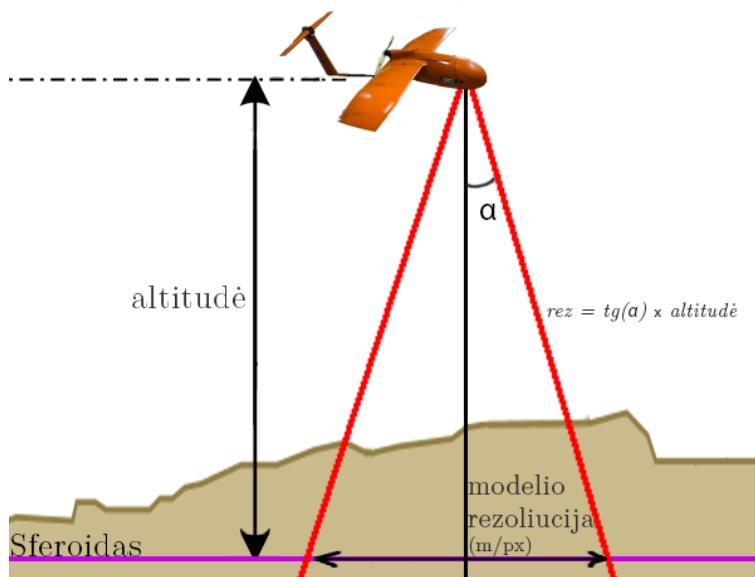
Paveikslėlių registracijai yra pasirinktas šablonų poravimo būdas. Yra siekiama suporuoti du paveikslėlius, kurių vienas reprezentuotų orlaivio būseną, o kitas geografinę padėtį. Šiam tikslui yra konstruojamas globalaus pasaulio žemėlapio poaibis ([Scena](#)), kuriame gali būti bepilotis orlaivis. Scenos dydis yra nusakomas aukščio ir pločio kvadratelių skaičiumi aprašytu skyriuje [Žemėlapio sudarymas](#) ir visada yra konstruojamas aplink centra, kuriame labiausiai tikėtina yra bepilotis orlaivis. Scenoje ieškomas atitinkmuo vaizdo kadro, kurį fiksuoja orlaivis([Modelis](#)).

### 2.2.1 Mastelio suvienodinimas

Šablonų poravimas priklauso nuo vienodos scenos ir modelio paveikslėlių rezoliucijos, t.y. kad tas pats vaizduojamas objekto skirtiniuose paveikslėliuose užimtu panašų kieki pikselių, kitaip tariant jo plotas ar perimetras būtų labai panašus. Šiam tikslui atliekamas scenos ir modelio mastelio suvienodinimas. Žemėlapio rezoliucija vadinsime dydį, kuris nusako, kiek metrų atitinka vienas pikselis. Scenos rezoliuciją galima apskaičiuoti pagal formulę (2), tuo tarpu modelio rezoliuciją reikia apskaičiuoti skrydžio metu (22), pavaizduota iliustracijoje 5.

$$r_m = \tan(\alpha) \times \text{altitudė} \quad (22)$$

čia  $r_m$  yra modelio rezoliucija, amplitudę galima sužinoti iš barometro duomenų, o kampus  $\alpha$  yra vaizdo kameros aprėpties kampus. Jeigu aprėpties kampus nėra žinomas, tai jį galima nustatyti bandomojo skrydžio metu. Skaičiuojant rezoliuciją yra daroma prielaida, kad žemės paviršius yra lygus [WGS-84](#) standarto numatytam paviršiui, dėl to realus rezoliucijos dydis gali šiek tiek skirtis, tačiau tai neturi jokios prasminės įtakos atliekamiems skaičiavimams.



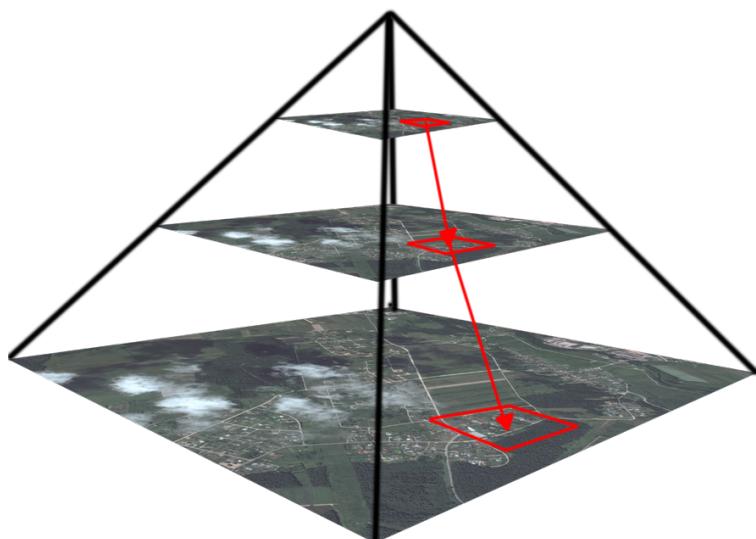
Galiausiai turint modelio rezoliucija ( $r_m$ ) ir scenos rezoliucija ( $r_s$ ), galima pakeisti modelio mastelį  $\frac{r_s}{r_m}$  kartų.

### 2.2.2 Klaidų mažinimas

Klaidos gali būti dviejų rūšių, 1 rūšies klaida - kai atmetame teisingą sugrupavimą ir 2 rūšies klaida, kaip priimame neteisingą sugrupavimą. Siekiant sumažinti 2 rūšies klaidų tikimybę aibės  $F$  (7) taškus galima konstruoti kiek kitaip. Atitikmenų paveikslėliui pritaikius slenksčio su parametru  $c^-$  operaciją, gausime įvairaus dydžio ir formos baltų laukų grupes paveikslėlyje. Grupuojant kryžminės koreliacijos metodu bei slenkant modelį per sceną, neteisingose vietose gali atsirasti aukštų atitikmenų įverčių, tačiau šie taškai dažniausiai bus išsibarstę ir nesusikoncentravę į vieną didelį taškų spiečių, priešingai, taškai poruojami aplink tikrają vietą įgis vis didesnį ir didesnį įvertį, taip lemdamas baltų taškų susigrupavimą į spiečių. Atskirti šias situacijas galima pasinaudojus paveikslėlio erozijos operaciją. Atlikus šią operaciją, pavieniai, nedideli taškai bus panaikinti ir bus palikti tik dalis taškų iš buvusių didelių grupių. Iš šių taškų išrinkus centrinius taškus galima sudaryti aibę  $F$ , taip yra sumažinama 2 rūšies klaida.

### 2.2.3 Optimizavimas

Kryžminės koreliacijos apskaičiavimas reikalauja daug skaičiavimo pajėgumų, tai yra  $O(w_m^2(w_s-w_m+1)^2)$  operacijų. Yra sukurtų variantų, kaip šiuos skaičiavimus optimizuoti [4]. Šio darbo metu yra naudojama piramidžių konstravimo optimizacija. Paveikslėlio piramidės - tai rinkinys mažesnės rezoliucijos paveikslėlių padarytų iš pagrindinio paveikslėlio. Piramidės rezoliuciją apibrėžia jos lygis, kiekvieno lygio paveikslėlis yra perpus mažesnis negu žemesnio lygio paveikslėlis, t. y. pirmajame lygyje yra paveikslėlis sumažintas du kartus, o antrajame - 4 kartus ir t. t., pavyzdys pateiktas iliustracijoje 6.



Illiustracija 6: Šablonų poravimo optimizacija piramidžių būdu

Sukonstravus piramides galima šablonų poravimą atlikti aukščiausiam piramidės lygyje, ir tada kiekviename žemesniame piramidės lygyje patikslinti surasto taško vietą ir jo įverti, tai galima aprašyti algoritmu:

```

1: Duomenys: Scena -  $S(w_s, h_s)$ , Modelis -  $M(w_m, h_m)$ , piramidės lygių skaičius - l
2: Rezultatas: poravimų kandidatų aibė  $F$ 
3: Sukonstruojamos l lygių piramides  $P_M$  modeliui ir  $P_S$  scenai;
4:  $i = l$ ;
5: while  $i > 0$  do
6:   if  $i = l$  then
7:     Atliekame šablonų poravimą su scena  $P_S[i]$  ir moduliu  $P_M[i]$  bei gauname
      atitinkmenų aibę  $F_i$ ;
8:     Atliekame klaidų mažinimą (2.2.2);
9:   else
10:     $F_i = \{\}$  // Sukuriame tuščią aibę
11:    foreach elementui  $(x, y, c) \in F_{i-1}$  do
12:      suprojektuojame tašką  $(x, y)$  į scenos  $P_S[i]$  paveikslėlį ;
13:      ištraukiame scenos regioną  $S_{i(x,y)}$ , kuris yra šablono  $P_M[i]$  + pakraščių,
        kurios išskaičiuojamos c dydžio ;
14:      atliekame šablonų poravimą su scena  $S_{i(x,y)}$  ir šablonu  $P_M[i]$ , gautą
        maksimalų tašką  $(x', y', c')$  įdedame į aibę  $F_i$ ;
15:    end
16:  end
17:   $l = l - 1$ 
18: end
19: Gražiname aibę  $F = F_0$ 
```

**Algoritmas 3:** Piramidžių optimizavimas šablonų poravimui

## 2.3 Lokalizacija

Šiame darbe vietas nustatymas vyksta geografinėje sistemoje, t. y. aibėje

$$\mathbb{G} = \{\langle lat, long \rangle \mid lat \in [-90^\circ; 90^\circ], long \in [-180^\circ; 180^\circ]\} \quad (23)$$

Ir kartu yra fiksuojamas pasiskrimimo kampus dvimatėje erdvėje:

$$\mathbb{O} = [0^\circ; 360^\circ] \quad (24)$$

Taigi aibė  $\mathbb{S}$  (8) šiame darbe yra apibrėžiama:

$$\mathbb{S} = \{\langle lat, long, \phi \rangle \mid \langle lat, long \rangle \in \mathbb{G}, \phi \in \mathbb{O}\} \quad (25)$$

Šablonų poravimas (5), kuris yra naudojamas paveikslėlių registracijoje, yra pasiskirstęs pagal **multimodalinių** skirstinį, tai akivaizdžiai matosi iliustracijoje 9, todėl lokalizacijai yra pritaikomas anksčiau aprašytas dalelių filtras (1.4.2) ir konkrečiai algoritmas 1. Eilutėje 6 nurodoma  $p(x_t | x_{t-1}, u_t)$  yra ne kas kitas kaip duomenys iš inercinės navigacinės sistemos ar vizualios odometrijos kartu su paklaida, kuria šie duomenys yra teikiami. Pavyzdžiu jeigu buvo turima dalelė  $(54.73^\circ, 25.263^\circ, 33^\circ) \in \mathbb{S}$  ir yra žinoma, kad orlaivis pasislinko  $(0.01^\circ \pm 0.005^\circ, -0.01^\circ \pm 0.005^\circ) \subset \mathbb{G}$ , bei pasisuko  $(1^\circ \pm 0.5^\circ) \subset \mathbb{O}$  tai naujos dalelės  $\{x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[m]}\} \in \mathbb{S}$  bus generuojamos pagal Gauso skirstinį režiuose:  $lat_t^{[i]} \in [54.7315^\circ; 54.7305^\circ]$ ,  $long_t^{[i]} \in [25.2615^\circ; 25.2625^\circ]$ ,  $\phi_t^{[i]} \in [33.5^\circ, 34.5^\circ]$ , čia  $x_t^{[i]} = \langle lat_t^{[i]}, long_t^{[i]}, \phi_t^{[i]} \rangle, i = \overline{1..m}$ .

$N_{max}$  parametras nusakantis kiek maksimaliai dalelių gali būti sugeneruotų yra išskaičiuojamas iš kitų turimų duomenų. Turint **scenos** dydį nusakyta kvadratėliais (skyrius: 1.2), galima parinkti  $N_{max} = 200 \times \text{kvadratelių sk.}$ . Skaičius *inc* nurodantis kiek maksimaliai iš kiekvienos dalelės gali būti suprojektuotą naujų dalelių, bandymų metu šis skaičius būdavo ribose [3; 10].

Kadangi yra taikomas **Šablono atitikmenų būdas**, tai eilutėje 8 suteikiami svoriai yra lygties (5) sprendiniai.

Gautas rezultatas yra reprezentuoojamas dalelių spiečiais, t. y. grupėmis dalelių, kurios reiškia įsitikinimą *bel*. Kadangi vaizdu paremta navigacija yra skirta tik inercinės navigacinės sistemos arba vizualios odometrijos duomenų patikslinimui, dalelių filtras yra atliekamas tol, kol išryškėja viena grupė dalelių. Bandymų metu tokia grupė buvo laikoma išssiskirianti kai dalelių skaičius būdavo du kartus didesnis negu antros didžiausios grupės. Gavus vienintelį įsitikinimą, galima ji aproksimuoti į Gauso skirstinį, todėl šiuos duomenis galima pateikti Kalman Filtrui.

## 3 Rezultatai

### 3.1 Naudojama įranga

Darbo metu buvo naudojama tiek programinė, tiek techninė įranga. Bendradarbiaujant su Kosmoso mokslo ir technologijų institutu, buvo naudojama medžiaga, renkama iš kuriamo bepiločio orlaivio, pavaizduoto iliustracijoje 7.



Illiustracija 7: Kosmoso mokslų ir technologijų instituto kuriamas bepilotis orlaivius, kuris buvo naudotas šio darbo metu

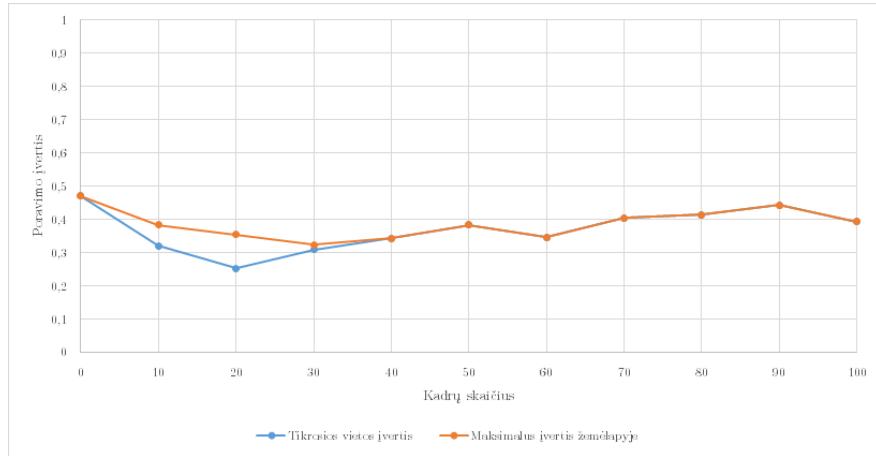
Iš programinės pusės buvo naudojama [C++](#) programavimo kalba kartu su [OpenCV](#) biblioteka vaizdų apdorojimui, tačiau testavimui ir prototipų kūrimui naudojama [C#](#) kalba ir [Emgu](#) biblioteka vaizdų apdorojimui. Geografiniai žemėlapiai buvo sukonstruoti naudojant [Bing Maps API](#).

### 3.2 Prielaidos ir apribojimai

Siekiant naudotis vaizdine medžiaga paremta navigacija svarbu turėti prieigą prie geografinių žemėlapių duomenų bazių. Šio darbo rašymo metu, tokios sistemos kaip *Microsoft Bing Maps* ar *Google Maps* yra viešai ir nesudėtingai prieinamos, todėl tai nelaikoma dideliu sunkumu. Taip pat sistema néra pritaikyta veikti tai atvejais, kai egzistuoja didelis vaizdinis skirtumas tarp dabartinės situacijos ir turimų geografinių žemėlapių, pavyzdžiu: skirtinges metų laikų sezonas, didelio gaisro ar potvynio atvejais, taip pat navigacija néra pritaikyta veikti tuo atveju, kai žemės paviršius yra vizualiai beveik vienalytis, pavyzdžiu: jūra, vandenynas, didelė giria be proskynų. Taip pat norint įgalinti sistemą pradėti vykdyti navigaciją būtina nurodyti bent abstraktų pradžios tašką.

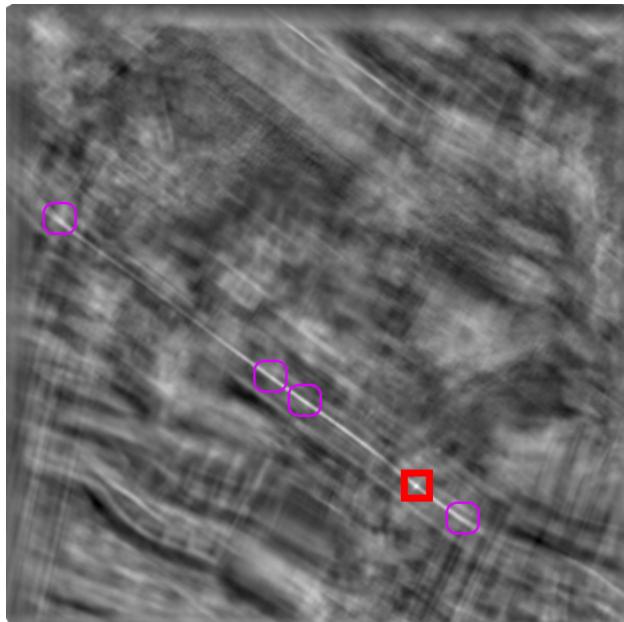
### 3.3 Paveikslėlių registracija

Eksperimentų metu poruojant sceną su modeliu ne visada galima remtis tik aukščiausią lygties 5 apskaičiuota reikšme. Grafike 8 yra pateiktas tikrosios vienos, kuri buvo pažymėta rankiniu būdu, ir maksimalaus poravimo įvertinimo. Atlikus poravimą su 100 kadrių, matyti, kad nors dažniausiai tikroji vieta yra surandama tinkamai, tai nėra visada vienareikšmiškas rezultatas.



Iliustracija 8: Modelių (vaizo kadrių) ir scenos poravimas ne visada aukščiausią įvertį paskaičiuoja tikrajam taškui.

Iš kitos pusės egzistuoja daug įvertinimų, kurių skirtumas lyginat su maksimalia reikšme nėra didelis. Paveikslėlyje 9 yra pažymėtos vienos (apibrėžtų violetine spalva), kurios nuo maksimalaus taško (apibrėžto raudona spalva) skiriasi ne mažiau kaip 70%.



Iliustracija 9: Aibės  $C$  taškai pavaizduoti kaip paveikslėlis, lokalių maksimumo taškų aibės  $F$  elementai yra apibrėžti stačiakampiais, globalaus maksimumo taškas raudono, kiti violetinės spalvos.

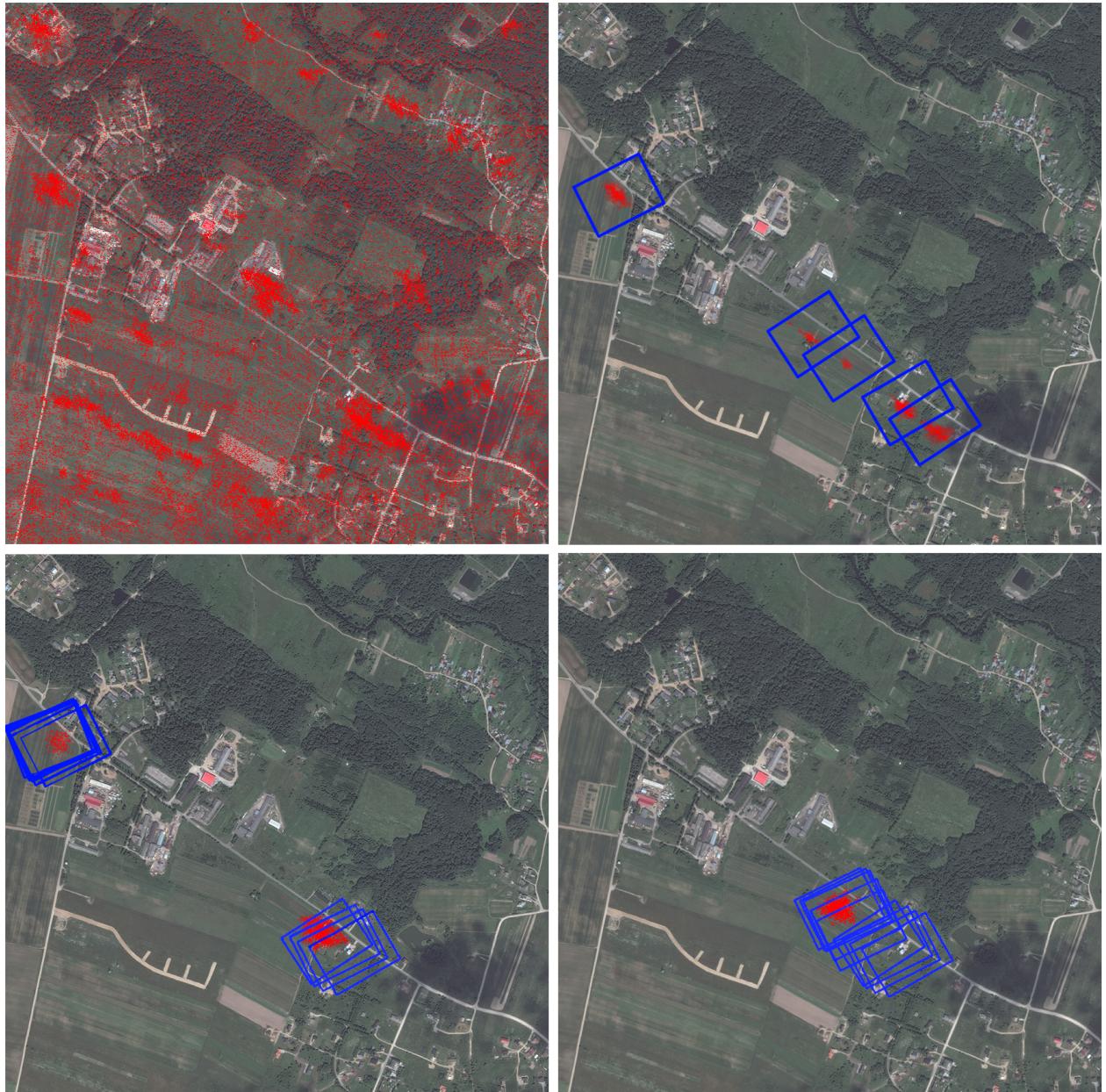
Iliustracijoje 10 yra pavaizduoti šie suporuoti taškai.



Illiustracija 10: Vaizdo kadro (dešinėje) paieška sukonstruotame  $20 \times 20$  kvadratelių arba daugiau negu  $9km^2$  ploto žemėlapyje (kairėje). Raudonais stačiakampiais pavaizduoti surasti pradiniai kandidatai.

### 3.4 Lokalizacija

Atlikus pirmąjį paveikslėlių poravimą yra sukuriamos pradinės dalelių filtro algoritmui. Dalelių filtras rekursyviai perskaičiuodamas daleles išrenka tik tas, kurios yra labiausiai tikėtinės, taip apytiksliai apskaičiuojant orlaivio poziciją. Šis procesas yra pavaizduotas paveikslėlyje 11



Iliustracija 11: Dalelių filtro algoritmu yra atliekama lokalizacija žemėlapyje ir iš daugelio įsitikinimų yra išrenkamas vienas. Raudonai yra pažymėtos dalelės, o mėlynais stačiakampiais yra pažymėti suporuoti su žemėlapiu kadrai. Nuo viršaus iš dešinės paveikslėliai žymi navigacinės sistemos duomenis: 1) pradinėje stadioje 2) po 5 kadru 3) po 50 kadru 4) po 150 kadru

Galiausiai yra apskaičiuojama orlaivio pozicija WGS 84 koordinačių sistemoje. GPS ir vaizdu pagrįstos lokalizacijos rezultatų palyginimas yra pavaizduotas iliustracijoje

**12.** Vidutinis skirtumas tarp GPS ir kuriamos navigacinės sistemos duomenų yra 40 metrų. Tačiau galima pastebėti, kad lyginant su GPS visi vaizdu pagrįstos navigacinės sistemos duomenys yra apytiksliai 36 metrais pasislinkę į rytus. Darbo metu šio poslinkio priežastys tiksliai nebuvvo nustatytyos.



Iliustracija 12: GPS ir vaizdu pagrįstos navigacinės sistemos rezultatų palyginimas. Mėlynai pažymėtas takas žymi GPS, o raudonai vaizdu pagrįsto navigacinės sistemos duomenis.

## 4 Išvados

Šio darbo metu buvo sukurta vaizdu paremta vietas nustatymo sistema, kuri dirba kartu su inercine navigacine sistema arba vizualia odometrija. Lokalizacija buvo atlikta tikrų skrydžių simuliacijų metu - t.y. atliekamas skrydis ir vėliau jis yra atkartojama jo simuliacija. Bepilotis orlaivis gali sėkmingai nustatyti savo vietą žemėlapyje, net kai egzistuoja daugiau kaip  $9km^2$  neužtikrintumas. Rekursyviai atnaujinant savo poziciją, vaizdu paremta navigacinė sistema geba nustatyti orlaivio koordinates WGS-84 koordinacių sistemoje. Nors vaizdu paremtos navigacinės sistemos duomenys yra nukrypę nuo GPS apskaičiuotų koordinacių vidutiniškai 40 metrų, apytiksliai 36 metrai yra pastovus ir nekontinantis nuokrypis, kurį būtų galima pašalinti ateityje. Visgi šiai sistemai reikalinga, kur kas didesni skaičiavimo pajėgumai negu remiantis [GPS](#) lokalizacija, taip pat būtina orlaivio atmintyje saugoti geografinius žemėlapius. Nepaisant šių trūkumų, vietas nustatymas remiantis vaizdine informacija gali būti viena iš pagrindinių alternatyvų, kaip lokalizuoti bepilotį

orlaivij urbanistinėse vietovėse, kur GPS paklaida būna didžiulė ar konfliktinėse zonose.

## 4.1 Apribojimai ir galimybės

Darbo metu naudojamas dalelių filtras yra apibrėžtas 3 dimensijų, kurias sudaro platumos, ilgumos ir posūkio kampai. Esant nedideliam dimensijų skaičiui dalelių filtras rodo puikius rezultatus, tačiau jeigu būtų siekiama išmatuoti orlaivio būseną matuojant daugiau dimensijų, pavyzdžiui, aukštis ar posūkio kampas trijose dimensijos, šis algoritmas reikalautų didelių skaičiavimo pajėgumų.

Daugiausiai skaičiavimo pajėgumų reikalauja paveikslėlių registravimas, tačiau šis algoritmas gali būti skaičiuojamas naudojant vaizdo plokštę.

# Terminai ir trumpiniai

**Akcelerometras** prietaisas, kuris išmatuoja pagreitį. [4](#)

**BPO** bepilotis orlaivis. [1](#), [4](#)

**C++** Objektiškai orientuota programavimo kalba. [20](#)

**C#** Objektiškai orientuota programavimo kalba sukurta Microsoft. [20](#)

**EKF** abreviatūra, angl. *Extended Kalman Filter*- algoritmas išplėstas Kalman filtro algoritmas, kuris yra tapęs *de facto* standartu navigacinėse sistemoje. [13](#)

**Emgu** biblioteka, kuri įgalina naudoti [OpenCV](#) kartu su .NET karkasu. [20](#)

**FLANN** abreviatūra, angl. *Fast Library for Approximate Nearest Neighbors* - tai yra biblioteka, kuri turi kolekciją algoritmų optimizuotų greitai gretimų ypatybių paieškai dideliuose duomenų kiekiuose, pasiūlytas M. Muja, D. Lowe. [4](#)

**FREAK** abreviatūra, angl. *Fast Retina Keypoint* algoritmas apibūdinantis paveikslėlio ypatybes. [4](#)

**Giroskopas** prietaisas, skirtas sukimuisi surasti, jį išmatuoti. [4](#)

**GPS** globali pozicionavimo sistema. [2](#), [23](#)

**Homografinė matrica** Matrica, kuri naudojama susieti du paviršius. [4](#)

**Mercator projekcija** Flamandų geografo ir kartografo Gerardus Mercator sukurta cilindrinė žemėlapio projekcija. [5](#)

**Modelis** Žemės paviršiaus nuotrauka nufotografuota orlaivio kamera. [6](#), [16](#)

**Multimodalinis skirstinys** Skirstinys, kuris turi daugiau negu vieną lokalų maksimumą. [12](#), [19](#)

**OpenCV** abreviatūra *Open Source Computer Vision*, programavimo biblioteka pagrinde skirta darbui su vaizdo apdorojimu. [20](#)

**Paveikslėlių registracija** Tai procesas, kurio metu dvi skirtingos aibės duomenų yra susijamos į vieną koordinačių sistemą. [2](#)

**RANSAC** abreviatūra, angl. *Random sample consensus* - algoritmas, kuris iš aibės išmeta išskirtis. [4](#), [7](#)

**Scena** Žemės paviršiaus žemėlapis, kurio geografinė padėtis yra žinoma. [6](#), [16](#), [19](#)

**SIFT** abreviatūra, angl. *Scale Invariant Feature Transform algorithm*- algoritmas, kuris suranda ir apibūdina paveikslėlio ypatybes, yra nepriklausomas nuo ypatybių mastelio transformavimo, pasiūlytas D. Lowe. [4](#)

**SURF** abreviatūra, angl. *Speeded Up Robust Features* - algoritmas kuris suranda ir apibūdina paveikslėlio ypatybes. [4](#)

**Vienmodalinis skirstinys** Skirstinys, kuris turi vieną lokalų maksimumą. [12](#)

**WGS-84** abreviatūra, angl. *World Geodetic System* – Pasaulio geodezinė sistema, tai standartas apibrėžiantis koordinačių sistemą žemės planetai, naudojamas kartografijoje, geodezijoje, navigacijoje, kurios paskutinė versija yra 1984 metų. [3](#), [5](#), [16](#)

## Literatūros sarašas

- [1] S. Abbasi and F. Mokhtarian. Shape similarity retrieval under affine transform: application to multi-view object representation and recognition. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, tomas 1, puslapiai 450–455 vol.1, 1999.
- [2] John Nielsen Ali Jafarnia-Jahromi, Ali Broumandan and Gérard Lachapelle. Gps vulnerability to spoofing threats and a review of antispoofting techniques. *International Journal of Navigation and Observation*, 12:16, 2012.
- [3] L. Bazzani, D. Bloisi, and V. Murino. A comparison of multi hypothesis kalman filter and particle filter for multi-target tracking. *Performance Evaluation of Tracking and Surveillance workshop at CVPR*, puslapiai 47–54, Miami, Florida, 2009.
- [4] Roberto Brunelli. *Template matching techniques in computer vision*, skyrius Computational aspects of template matching, puslapiai 201–220. John Wiley & Sons Ltd, 2009.
- [5] Gianpaolo Conte. *Vision-Based Localization and Guidance for Unmanned Aerial Vehicles*. PhD thesis, Linköpings universitet, 2009.
- [6] Ming Cui, John Femiani, Jiuxiang Hu, Peter Wonka, and Anshuman Razdan. Curve matching for open 2d curves. *Pattern Recognition Letters*, 30(1):1–10, 2009.
- [7] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, puslapiai 15–22, Gegužė 2014.
- [8] Michael A. Goodrich, Bryan S. Morse, Damon Gerhardt, Joseph L. Cooper, Morgan Quigley, Julie A. Adams, and Curtis Humphrey. Supporting wilderness search and rescue using a camera-equipped mini uav. *Journal of Field Robotics*, 25(1-2):89–110, 2008.
- [9] F. Gustafsson. Particle filter theory and practice with positioning applications. *Aerospace and Electronic Systems Magazine, IEEE*, 25(7):53–82, Liepa 2010.
- [10] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, Kovas 2004.
- [11] Yucong Lin and S. Saripalli. Road detection from aerial imagery. *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, puslapiai 3588–3593, Gegužė 2012.
- [12] Ezio Malis and Manuel Vargas. Deeper understanding of the homography decomposition for vision-based control. Research Report RR-6303, 2007.

- [13] Guoqiang Mao, S. Drake, and B.D.O. Anderson. Design of an extended kalman filter for uav localization. *Information, Decision and Control, 2007. IDC '07*, puslapiai 224–229, Vasaris 2007.
- [14] Marko Modsching, Ronny Kramer, and Klaus ten Hagen. Field trial on gps accuracy in a medium size city: The influence of built-up. *3rd Workshop on Positioning, Navigation and Communication*, puslapiai 209–218, 2006.
- [15] R. Siegwart, I.R. Nourbakhsh, and D. Scaramuzza. *Introduction to Autonomous Mobile Robots*. Intelligent robotics and autonomous agents. MIT Press, 2011.
- [16] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Intelligent robotics and autonomous agents. MIT Press, 2005.
- [17] Yue Wang and Eam Khwang Teoh. 2d affine-invariant contour matching using b-spline model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(10):1853–1858, 2007.
- [18] Y. Watanabe, P. Fabiani, and G. Le Besnerais. Simultaneous visual target tracking and navigation in a gps-denied environment. *Advanced Robotics, 2009. ICAR 2009. International Conference on*, puslapiai 1–6, Birželis 2009.

## Santrauka

Globali pozicionavimo sistema (GPS) néra visada pasiekama ar saugi naudojimui. Dažniausiai tokia problema iškyla urbanistinėse ar konfliktinėse zonose. Bepiločių orlaiviu (BPO) panaudojimo tikslinės sritys dažnai apima vėtos nustatymą šiose zonose, todėl šio darbo metu yra apžvelgiamos galimos GPS sistemos alternatyvos, kurios remtūsi vaizdo kamera kaip pagrindiniu jutikliu bei pateikiami pasiūlymai kaip įgyvendinti šią navigacinę sistemą. Vaizdo medžiaga iš kameros naudojama susiejant (registruojant) su geografiniu žemėlapiu, taip siekiant nustatyti globalias orlaivio koordinates WGS 84 koordinačių sistemoje. Šios koordinatės yra sujungiamos su inercinės navigacinės sistemos duomenis, taip įgyvendinant pilnavertę navigacinę sistemą. Eksperimentiniai tyrimai, kurių metu buvo simuliacijoje atkuriami tikrų skrydžių duomenys, parodė, kad sistema yra pakankamai tikslia, o paklaida laikui bėgant nesikaupia, todėl ši sistema gali pakeisti GPS ir taip užtikrinti pilną orlaivio autonomiškumą.

# Summary

The Global Positioning System (GPS) is not always available and safe to use. Mostly this problem occurs in urban areas or conflict zones. Unmanned aerial vehicles (UAV) tasks often include these areas where UAV possibilities are strongly limited. This work is an overview of possible alternatives to the GPS system, which would use video camera as a primary sensor and proposes how to implement vision-based navigation system. Video footage from the camera are registered to the geographical map in order to assess UAV coordinate in WGS-84 coordinate system. These coordinates are combined with data from inertial navigation system. These decisions implement a fully-fledged navigation system. Experimental studies, conducted simulation played real flight data, showed that the system is accurate enough and the error does not accumulate over time, so the system can change the GPS to ensure complete autonomy of the aircraft.