



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
ELEKTRONINIŲ SISTEMŲ KATEDRA

Ignas Brašiškis

OBJEKTAMS SEKTI ERDVĖJE TINKAMŲ METODŲ TYRIMAS
RESEARCH ON METHODS FOR OBJECT SPATIAL TRACKING

Magistro baigiamasis darbas

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

Vilnius, 2015

Vilnius Gediminas Technical University Faculty of Electronics Department of Electronic Systems	ISBN _____ ISSN _____ Copies No. Date-.....-.....
Master Degree Studies Information Electronics Systems study programme Master Graduation Thesis	
Title	Research of Methods for Object Spatial Tracking
Author	Ignas Brašiškis
Academic supervisor	Assoc Prof Dr Dalius Matuzevičius
Thesis language: Lithuanian	
Annotation The Master Thesis covers video object tracking system problems, implementation and usage challenges. The Work covers basic parts and algorithms needed for implementation of system, and proposes SOM artificial neural network for use in tracking. Thesis provides a classification of objects, differences between possible implementations. More important algorithms for tracking were covered in depth . Data-set was formed for experiments. Experiment results were provided. Conclusions and proposals were made. Thesis consists of several parts: introduction, algorithm analysis, system creation, experiment results, conclusions.	
Keywords: features, object matching, object templates, object tracking, SOM network, tracking system	

Vilniaus Gedimino technikos universitetas
Elektronikos fakultetas
Elektroninių sistemų katedra

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

Antrosios pakopos studijų **Informacinių elektroninių sistemų** programos magistro baigiamasis darbas

Pavadinimas **Objektams sekti erdvėje tinkamų metodų tyrimas**

Autorius **Ignas Brašiškis**

Vadovas **doc. dr. Dalius Matuzevičius**

Kalba: lietuvių

Anotacija

Baigiamajame magistro darbe nagrinėjamos objektų sekimo sistemų vaizde problemos, perspektyvos ir įgyvendinimo būdai, egzistuojantys metodai šioje srityje. Išnagrinėti pagrindiniai metodai ir sudedamosios sistemų dalys, nuo kurių priklauso objektų sekimo efektyvumas, intelektualųjų algoritmų (SOM neuronų tinklo) taikymas optimizuojant sekimą. Pateiktas sekimo erdvėje metodų klasifikavimas, išanalizuoti įvairių jo rūšių skirtumai. Darbe išnagrinėti pagrindiniai algoritmai ir jų efektyvumas. Algoritmams tirti sudarytas duomenų rinkinys, sukurtas eksperimentinis planas. Su duomenų rinkiniu atlikti eksperimentai ir pateikti jų rezultatai. Išnagrinėjus algoritmus ir jų taikymo rezultatus, pateikiamos baigiamojo darbo išvados ir siūlymai.

Darbą sudaro 7 dalys: įvadas, analitinė apžvalga, atrinktų algoritmų analizė, sistemos kūrimas, duomenų atranka, eksperimentų rezultatai, išvados ir siūlymai.

Prasminiai žodžiai: būdingieji taškai, objektų sekimas, objektų šablonai, sekimo sistemos, SOM tinklas, vaizdo analizė

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

1 priedas

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Ignas Brašiškis, 20093062

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Elektronikos fakultetas

(Fakultetas)

Informacinės elektroninės sistemos, AKSfm-13

(Studijų programa, akademinė grupė)

**BAIGIAMOJO DARBO (PROJEKTO)
SĄŽINGUMO DEKLARACIJA**

2015 m. birželio 1 d.

Patvirtinu, kad mano baigiamasis darbas tema „Objektams sekti erdvėje tinkamų metodų tyrimas“ patvirtintas 2013 m. spalio 28 d. dekanų potvarkiu Nr. 392el, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Mano darbo vadovas doc. dr. Dalius Matuzevičius.

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

(Parašas)

Ignas Brašiškis

(Vardas ir pavardė)

TURINYS

TURINYS	7
PAVEIKSLŲ SĄRAŠAS	9
LENTELIŲ SĄRAŠAS	10
ŽYMENYS IR SANTRUMPOS	11
ĮVADAS	12
1. OBJEKTŲ SEKIMO METODŲ ANALITINĖ APŽVALGA	14
1.1. Objektų sekimo sistemos sudedamosios dalys	14
1.2. Požymių tinkamų sekimui klasifikacija	15
1.3. Objektų atpažinimo metodų klasifikacija	17
1.4. Objekto atvaizdavimas	18
1.5. Metodų analizės apibendrinimas	19
2. ATRINKTŲ ALGORITMŲ OBJEKTAMS SEKTI ERDVĖJE ANALIZĖ	20
2.1. Būdingųjų taškų algoritmai	20
2.1.1 SURF algoritmas	20
2.1.2 BRISK algoritmas	22
2.1.3 FREAK algoritmas	24
2.1.4 Šablono sutapdinimas būdingųjų taškų algoritmuose	25
2.2. Tekstūros išskyrimo algoritmai	26
2.2.1 HOG algoritmas	26
2.3. Trumpas skyriaus apibendrinimas	28
3. SEKIMO SISTEMOS GRĮSTOS SOM TINKLU KŪRIMAS	29
3.1. Objekto sekimo sistema	29
3.1.1 Sistemos įvestis	29
3.1.2 Sistemos struktūra	30
3.2. SOM algoritmo pritaikymas vaizdo sekimui	30
3.2.1 SOM algoritmas	30
3.2.2 SOM algoritmo pritaikymas vaizdo sekimui	32
3.3. Trumpas skyriaus apibendrinimas	35
4. DUOMENŲ ATRANKA OBJEKTŲ SEKIMO METODŲ TYRIMUI	36
4.1. Įvesties duomenys	36
4.1.1 Vaizdo sekos	36
4.1.2 Šablonai	37

4.2. Trumpas skyriaus apibendrinimas	38
5. OBJEKTŲ SEKIMO METODŲ TYRIMO REZULTATAI	39
5.1. Požymiams ir sutapdinimui naudojamų algoritmų tyrimas	39
5.2. SOM grįsto objektų sekimo algoritmo tyrimas	41
5.3. Trumpas skyriaus apibendrinimas	42
APIBENDRINIMAS. IŠVADOS	43
LITERATŪRA	45

PAVEIKSLŲ SĄRAŠAS

1.1 pav.	Sekimo sistemos apdorojimo seka	14
1.2 pav.	Fono pašalinimo algoritmo veikimo pavyzdys	17
1.3 pav.	Objektų formos atvaizdavimo metodai.	18
2.1 pav.	Integralinio vaizdo intensyvumo skaičiavimas	21
2.2 pav.	Intensyvumo antros eilės išvestinės	21
2.3 pav.	SURF Deskriptoriaus krypties parinkimas	22
2.4 pav.	BRISK Fast piramidė	23
2.5 pav.	BRISK požymio vektoriaus taškų pasiskirstymas.	24
2.6 pav.	FREAK deskriptoriaus langas	25
2.7 pav.	FREAK deskriptoriaus langas	25
2.8 pav.	HOG algoritmo eiliškumas.	27
2.9 pav.	HOG požymių vektoriaus (histogramų vektoriaus sudarymas).	27
3.1 pav.	Šablono ir kadro požymių sukūrimo, bei sistemos algoritmai	31
3.2 pav.	SOM tinklo pradinė neuronų padėtis	32
3.3 pav.	SOM tinklo poslinkis apmokius tinklą vienu tašku.	33
3.4 pav.	SOM tinklo taškų santalkos srities apskaičiavimas.	33
4.1 pav.	Šablono išskyrimas iš kadro	37
5.1 pav.	Taškų sutapdinimas, naudojant BRISK	39
5.2 pav.	Klaidingo sutapdinimo santykis palyginimas pagal atstumą tarp sutapdinimų.	39
5.3 pav.	Taškų sutapdinimo santykio palyginimas pagal atstumą tarp sutapdintų taškų.	40
5.4 pav.	Teisingo sutapdinimo santykio palyginimas pagal sėkmingai sutapdintų kad- rų skaičių	41

LENTELIŲ SĄRAŠAS

1.1 lentelė. Požymių grupės įgyvendinančių algoritmų pavyzdžiai	16
3.1 lentelė. Algoritmų kombinacijos	30
3.2 lentelė. SOM tinklo 6x6 svorių pradinis išdėstymas	32
4.1 lentelė. Įvesties duomenų šablonui generuoti formatas.	37
5.1 lentelė. Optimalus atstumas esant pasirinktai klaidos tolerancijai	40

ŽYMENYS IR SANTRUMPOS

BRISK (angl. Binary robust invariant scale keypoints) algoritmas naudojamas dvejetainiams būdingiesiems požymiams rasti. 30

FLANN (angl. Fast retina keypoint) algoritmas naudojamas būdingiesiems požymiams sutaptinti. 30

FREAK (angl. Fast retina keypoint) algoritmas naudojamas būdingiesiems požymiams rasti į žmogaus tinklainę panašiu langu. 30

HOG (angl. Histogram of gradients) algoritmas apibrėžiantis požymius gradientų histogramomis. 25

SURF (angl. Speed up robust features) algoritmas naudojamas būdingiesiems požymiams rasti. 19, 20, 30

taškų santalka (angl. points cluster) didelio tankio taškų sritis. 32

IVADAS

Darbo aktualumas ir tikslas

Objektų sekimas – vienas iš svarbiausių komponentų moderniose kompiuterinės vizualizacijos ir atpažinimo sistemose. Sekimas reikalingas į kompiuterines sistemas įvesti įvairius natūraliuosius vaizdus, tokius kaip fiziniai objektai arba vizualizuoti išvestį realiame vaizdų sraute. Šie uždaviniai reikalauja nemažų kompiuterio išteklių (Irvine *et al*, 2013) ir yra neatsiejami nuo vaizdo analizės, bei išmaniųjų algoritmų. Šių algoritmų aktualumą nulėmė didelės skaičiavimo galios ir našumo kompiuterių, bei mobiliųjų įrenginių paplitimas. Šios klasės algoritmai naudojami laivams (Teng, Liu, 2014), automobiliams (Telagarapu *et al*, 2012) ir kitoms transporto priemonėms sekti, bei pėsčiųjų (Führ, Jung, 2014) judėjimui apskaičiuoti. Taip pat ši algoritmų ir metodų klasė taikoma papildytajai realybei (Nee *et al*, 2012)(Benko *et al*, 2012). Be to, šie metodai gali būti panaudoti ir spręsti identifikacijos uždaviniams – pavyzdžiui, nustatyti, koks žmogus eina (Bazzani *et al*, 2013).

Pagrindinė sekimo algoritmų problema yra jų darbo savybių ir našumo priklausomybė nuo taikomos aplinkos ir objektų savybių (Fernando *et al*, 2014). Dalis algoritmų pritaikyti veikti geriausiai šviesioje aplinkoje ir sekti veidus (Zhang, van der Maaten, 2013), kiti gali sekti specifiniais objektus (žmogaus siluetą, vienalytės spalvos objektus) ir t. t. Tuo tarpu bendrųjų objektams sekti skirtų metodų našumą, lemia išoriniai aplinkos veiksniai, bei vidinis algoritmų sudėtingumas. Kadangi į anksčiau aprašytus aspektus reikia atsižvelgti, objektams sekti skirtos sistemos įgyvendinimui turi būti parinkta algoritmų grupė, kuri sujungiama į vieną bendrą sistemą.

Pagrindinis tikslas yra susietas su algoritmų ir metodų, skirtų sekti vaizdui, sistema. Aki-vaizdu, kad sistemai reikia įgyvendinti našumo tikrinimą, todėl pagrindinį tikslą galima suformuoti taip: atlikti metodų, skirtų objektų sekimui, tyrimą.

Objekto sekimo analizė yra sudėtingas uždavinys. Siekiant šio tikslo reikia sukurti papildomus uždavinius ir tyrimų metodiką.

Darbo uždaviniai

Tikslui buvo suformuoti uždaviniai:

1. Atrinkti objektų sekimo metodų analitinę apžvalgą.
2. Atrinkti perspektyviausius metodus, skirtus objektams erdvėje sekti.
3. Ištirti atrinktus metodus objektui sekti ir pateikti tyrimo rezultatus.

Objektų sekimo metodų analitinės apžvalgos uždavinys sukurtas norint nustatyti kriterijus, reikalavimus našumui, bei sekimo algoritmų ir metodų naudojimo atvejus. Kiti metodai yra svarbūs kuriant ir tiriant sekimo metodų visumą.

Algoritmų analizei iš metodų visumos naudingiausia atrinkti rinkinius ir kombinacijas tinkamas objektams sekti – ne visi algoritmų rinkiniai tinka tokio pobūdžio sistemoms, ne visi rinkiniai lengvai įgyvendinami. Svarbu atrinkti algoritmus, kurie gali būti panaudoti praktiškai taikomoms sistemoms konstruoti ir tolesniam tyrimui.

Šie metodai naudojami sukurti sistemas ir ištirti jų vidines savybes. Jei perspektyviausios sistemos pasižymės geriausiomis savybėmis arba paaiškės, kokių metodų reikia įgyvendinti panašiai sistemai, tyrimas galės būti laikomas sėkmingu.

Naudoti tyrimo ir analizės metodai

Tyrimui ir analizei naudojami eksperimentiniai metodai yra grįsti būdingųjų taškų surastu kiekiu, sekimo sutapimo tikslumu, bei panašiais kriterijais įprastais vaizdo analizėje. Pagrindinė analizė daroma lyginant atskirus algoritmus ir juos sujungiant į sistemą. Šie duomenys leidžia nustatyti, kokiomis aplinkybėmis algoritmai veikia optimaliausiai.

Kaip įvesties duomenys naudojami vaizdų, sekamų objektų šablonai, įvesties parametrai algoritmams. Išvesties duomenys yra skaitiniai rezultatai apibūdinantis konkretaus eksperimento rezultatus. Darbe naudojamos algoritmų grupės: būdingųjų požymių išrinkimo (SURF algoritmai, BRISK ir kt.), aprašymo (FLANN, SURF požymių išgryninimas (angl. *feature extraction*)) ir sekimo įvykdymą. Detalesnė šių etapų analizė pateikiama analizuojant sistemą.

Darbo naujumas ir praktinė nauda

Šiame darbe nagrinėjama sistema yra sudaryta iš daugelio pakopų, kuri veikia kaip vienas įrankis su keletą atskirai valdomų komponentų. Tokios sistemos kombinacija tinka praktiniam taikymui, leidžia efektyviai prisitaikyti prie naudojimo aplinkos (pakeičiant vieną dalį). Darbe apžvelgiama, ko reikia tokio tipo sistemai įgyvendinti ir kaip galima tokią sistemą optimizuoti. Įgyvendinamas intelektualusis SOM algoritmas taškų sutapdinimui, leidžiantis adaptyviai keisti sutapdinimo langą.

Darbo struktūra

Magistro baigiamąjį darbą sudaro 5.3 skyriai.

1 ir 2 skyriuose atliekama jau egzistuojančių algoritmų analizė, skirta išsiaiškinti tinkamus algoritmus objektams sekti.

3 skyriuje pateikiamas objektams sekti tinkamos intelektualiosios sistemos struktūra apdorojant sekimo rezultatus. Šios sistemos pagrindas yra dirbtinis neuronų tinklas, atmetantis išorinius taškus ir tikslinantis poziciją. Šios sistemos tyrimo rezultatai pateikiami 5 skyriuje.

1. OBJEKTŲ SEKIMO METODŲ ANALITINĖ APŽVALGA

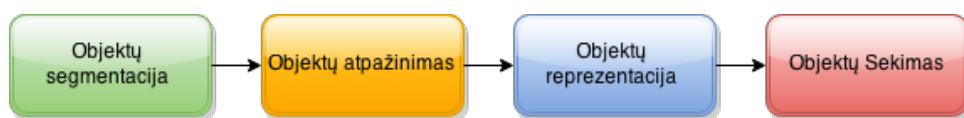
Šio skyriaus tikslas yra apžvelgti vaizdui sekti naudojamus algoritmus ir metodus, išnagrinėti sekimo sistemas. Sistemai reikia skirtingų algoritmų komponentų, kurie gali būti skirstomi pagal kategorijas, naudojimo paskirtį, spartą į grupes. Analizės metu apžvelgiamos kelios algoritmų grupės išskirtos pagal bendrą paskirtį ir eiliškumą objektų sekimo procese.

1.1. Objektų sekimo sistemos sudedamosios dalys

Paprasčiausiai sekimas gali būti apibūdintas, kaip objekto judėjimo trajektorijos ir padėties radimas scenoje. Sekimo metu kameros kadre išskiriama objekto sritis, judanti scenoje. Objekto zonoje uždedama „antraštė“ pačiam objektui atpažinti. Priklausomai nuo algoritmo, objektui aprašyti gali būti panaudota orientacija, spalvų rinkinys, forma kadre.

Algoritmai skirtingais aspektais sprendžia identiškąs problemas. Pagrindinė problema – kaip aprašyti objektą ir koks aprašas yra atspariausias kliūtims, kurios patiriamos įgyvendinant algoritmus. Galimos sistemos įgyvendinimui kliūtys (Babenko *et al*, 2011):

- realios trimatės erdvės transformacija dvimačiame kadre;
- kameros kadro kokybės suprastėjimas (trukdžiai, kameros artefaktai);
- sudėtinga objekto judėjimo trajektorija, kintanti, sudėtinė objekto forma (prigimtis);
- objektų persidengimai, išėjimas iš vaizdo;
- apšvietimo pokyčiai scenoje, dalinis apšvietimas.



1.1 pav. Sekimo sistemos apdorojimo seka

Problemos sekant objektą gali būti sumažinamos apibūrinant palengvinančių savybių rinkinį: nustatant, kad objekto orientacija, judėjimo kryptis ar forma pastovi. Kai kurie algoritmai išnaudoja šias savybes specializacijai ir našesniai arba tikslesniai veikimui. Sistema sukuriamą iš komponentų, kurie išrikiuojami į nuoseklią seką (angl. *pipeline*).

Seką sudaro (1.1 pav.) apdorojimo blokai (Mitzel, Leibe, 2012):

- objektų segmentacija,
- objektų atpažinimas,
- objektų atvaizdavimas ir modeliavimas,

- objektų sekimas.

Objekto segmentacijos metu išrenkamų objektai. Priklausomai nuo algoritmo, objektai gali būti aptinkami segmentacijos metu, t. y. atpažinimas sutampa su segmentacija. Kita vertus, segmentavimo procesas gali būti išskiriamas ir tik panaudojamas atlikti paruošiamąjį vaidmenį atpažinimui. Algoritmai dalyvaujantys segmentacijoje, apžvelgiami 1.2. Sugeneravus objektą, jį reikia modeliuoti. Ši proceso dalis aprašoma 1.1.

1.2. Požymių tinkamų sekimui klasifikacija

Objektų segmentacijos žingsnio metu aptinkami bendrieji vaizdo požymiai, kurie gali būti panaudoti sekimui. Šiame žingsnyje išrenkami požymiai, kurie yra tinkami objektų reprezentacijai, bet nebūtinai reprezentuoja objektą (pvz. anot (Jiang, Yilmaz, 2011) geri požymiai sekimui yra kampai, didelio kontrasto zonos).

Pagrindinis reikalavimas „geriems“ požymiams yra unikalumas. Šių ir panašių reikalavimų apibrėžimas yra svarbūs sėkmingam sekimui. Anot Smeulders *et al* (2014) galima išskirti šias požymių rūšis:

- **Spalva.** Ši požymio rūšis priklauso nuo fizinių faktorių: šviesos šaltinio spektro sudėties, paviršiaus atspindžio. Pagrindiniai šios požymių rūšies trūkumas – spalvinės sistemos vaizde ne visados atitinka realaus spalvinio suvokimo. Šiems požymiams naudojamos kelios skirtingos spalvinės sistemos, o tos kurios artimos žmogaus suvokimui, jautrios pašaliniam triukšmui.
- **Gradientų požymiai** Šie požymiai remiasi gradientų radimu ir jų kompozicija. Gradientų algoritmai ypač naudingi surasti specifiniam objektams, kaip žmogus, automobilis – tiems objektams kurie turi aiškią ir orientuotą tekstūrą išsiskiriančią iš aplinkos.
- **Kontūrai.** Objekto kontūro kraštai įprastai sudaro didelio intensyvumo sritis, tinkančias atskirti ir išskirti sritis. Šie požymiai atsparūs apšvietimo pokyčiui. Ši požymių klasė ypač tinkama realių objektų išskyrimui. Požymio kontūrai yra dažnai (artimu idealiam atvejui) stipriai susiję su fizinėmis objekto ribomis.
- **Optinis poslinkis.** Optinio poslinkio algoritmai naudoja srities taškų tankio poslinkio vektorius išskirti judančioms sritims. Pagrindinė šios grupės paskirtis yra būti naudojamam segmentavimui grįstam judesiu. Tokiu algoritmu įgyvendintos sistemos, kuriose reikia segmentuoti judančiam objektui statiniam fone.
- **Tekstūra.** Šie požymiai išskiria objektą pagal paviršiaus parametrus: intensyvumą, lygumą, reguliarumą. Kitaip nei spalviniai požymiai, tekstūros požymiams sugeneruoti reikia papildomo žingsnio perskaičiuojant spalvinius duomenis į tekstūros atitikmenį. Šie metodai primena kraštais grįstus požymių detektorius, kadangi priklauso nuo intensyvumo, nors įprastai tekstūrų pasiskirstymas tankesnis.
- **Erdviniai požymiai.** Šie požymiai naudojami stereo sistemose, taikant trimatį kontūrą kintantį laike. Įprastai šie požymiai sudaromi iš vienmačių taškų transformuojant ir

1.1 lentelė. Požymių grupės įgyvendinančių algoritmų pavyzdžiai

Algoritmo klasė	Pavyzdžiai
Spalviniai požymiai	RGB slenkstis SIFT lokalus deskriptorius (Lowe, 1999) Dalelių filtrų modeliai
Gradientų požymiai	SURF gradientų laukai (Bay <i>et al</i> , 2006)
Kontūrų požymiai	Canny kontūrų algoritmas (Ding, Goshtasby, 2001) Sobel Kontūro algoritmas (Vincent, Folorunso, 2009) Harris kampų detektorius
Tekstūrų požymiai	HOG algoritmo požymiai (Zhu <i>et al</i> , 2006) LBP algoritmo požymiai (Heikkilä <i>et al</i> , 2009)
Optinis poslinkis	Camshift algoritmas (Ning <i>et al</i> , 2012) Meanshift algoritmas
Erdviniai požymiai	Viso (Geiger <i>et al</i> , 2011)
Daugelio požymių junginiai	Įvairių požymių rinkiniai
Biologiniai požymiai	Dėmesio regionai (AR - angl. <i>Attention Region</i>) (Borji, Itti, 2013) EBIM (angl. Biologically Inspired Model)

nustatant jų poziciją trimatėje erdvėje. Ypač tikslūs požymiai erdvėje ir stereoskopiniame vaizde, tačiau reikalauja trimatės sąryšio transformacijos tarp kameros/kamerų vaizdo ir fizinės erdvės arba trimačio objekto modelio.

- **Daugelio požymių junginiai.** Ši klasė yra sudėtingų požymių klasė. Šie požymiai išnaudoja kitų požymių klasių sąryšių kombinacijas tiksliai išreikšti objektams.
- **Biologiniai požymiai.** Sudėtiniai požymiai skirti imituoti žmogaus suvokimą. Iš esmės šie požymiai atitinka žmogaus regą, modeliuojant sudėtingą daugelio pakopų modelį. Modeliai perspektyvūs, tačiau dažniausiai taikomi situacijose, kurioje žmogui artima reprezentacija svarbiau už algoritmo greitį. Įprastai vaizdui išskirti taikomi dirbtiniai neuronų tinklai, imituojantys regos centrą.

Pagal požymius į šias klases išskirti algoritmai gali būti naudojami tiek segmentavimui, tiek objekto vidiniam atpažinimui. Kiekvienas požymio algoritmus reikalauja detalesnės analizės. Iš šių požymių kelių atrinktų grupių algoritmus detalesnė analizė tęsiama 2 skyriuje.

Toliau pateikiama keletas implementacijų pavyzdžių 1.1 lentelėje.

Vienas iš svarbiausių aspektų į kurį reikia atkreipti dėmesį tai, kad skirtingų grupių algoritmai turi skirtingus atpažinimo taikymus, kurie skiriasi tiek našumu, tiek ir naudojimu. Vien atpažinimo metodai, objektų modeliai gali nulemti, kurie algoritmai bus naudojami, kurie ignoruojami.

1.3. Objektų atpažinimo metodų klasifikacija

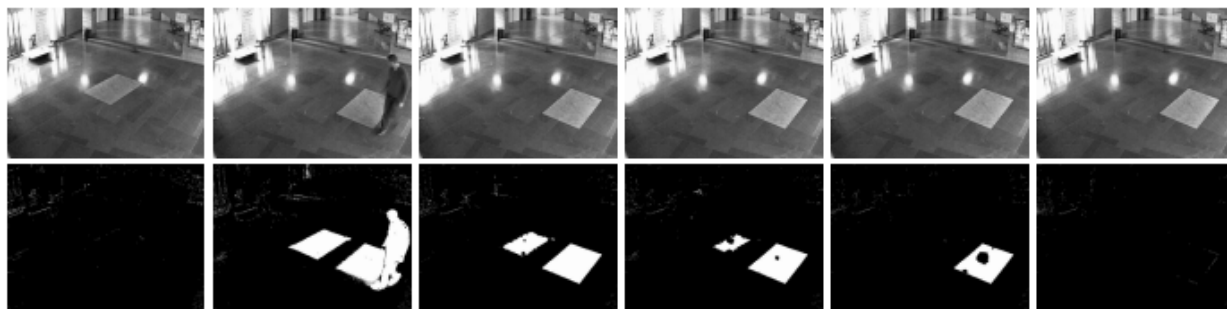
Objektų segmentacijos ir atpažinimo žingsnių metu išskiriami objektai kurie bus naudojami sekimui. Įprastai objekto modeliui atvaizduoti panaudojamas iš anksto atskirtas ir susegmentuotas objektas ar jo sritis. Objekto sritį aprašo 1.2 poskyryje aprašytų požymių rinkiniai. Kad šiuos požymius (nepriklausomai nuo klasės) pavyktų sugrupuoti į objektų modelius, reikalingi metodai, kurie nuspręstų ryšį tarp objektų.

Išskiriamos keturios atpažinimo metodų klasės: segmentavimu grįsti metodai, fono modeliavimo metodai, adaptyvūs apsimokantys fono atskyrimo metodai ir taškiniai metodai.

Segmentavimu grįsti metodai. Šie metodai aktualūs tada, kai objektai gali būti atskiriami dar segmentavimo metu. Šių metodų pagrindas yra objektų atskyrimas į vizualiai identišką sritį. Šios sritys ir apibrėžia objektą. Segmentavimas vykdomas taškų medžio principu: kur objektai suskirstomi į regionus pagal tai kurioje medžio dalyje atsидuria. Medis atsiradus poreikiui gali būti plečiamas, pridedant papildomas šakas, pašalinant neaktualias – medis transformuojamas dinamiškai.

Vis dėlto segmentavimo metodai įprastai pritaikomi optinio poslinkio (angl. *optical flow*) požymių klasei, kur judantis objektas išskiriamas kaip poslinkių regionas (pavyzdžiui, automobilis filmuotame vaizde matomas tarp kadru, kaip poslinkio vektorių telkinys judantis nuo kitų objektų). Segmentavimą ne visados galima pritaikyti.

Fono modeliavimo metodai Šie metodai grįsti fono pašalinimu iš vaizdo. Jie gerai veikia statiniams fonams, metodai grįsti statistiniu pasiskirstymu ir jo priklausomybe. Vienas iš metodų yra Gaus'o pasiskirstymas dinaminėms aplinkoms modeliuoti (Zivkovic, 2004), kur atkreipiamas dėmesys į tai, kad fono taškų pasiskirstymas stipriai skiriasi nuo pagrindinio vaizdo.



1.2 pav. Fono pašalinimo algoritmo veikimo pavyzdys. (Barnich, Van Droogenbroeck, 2011)

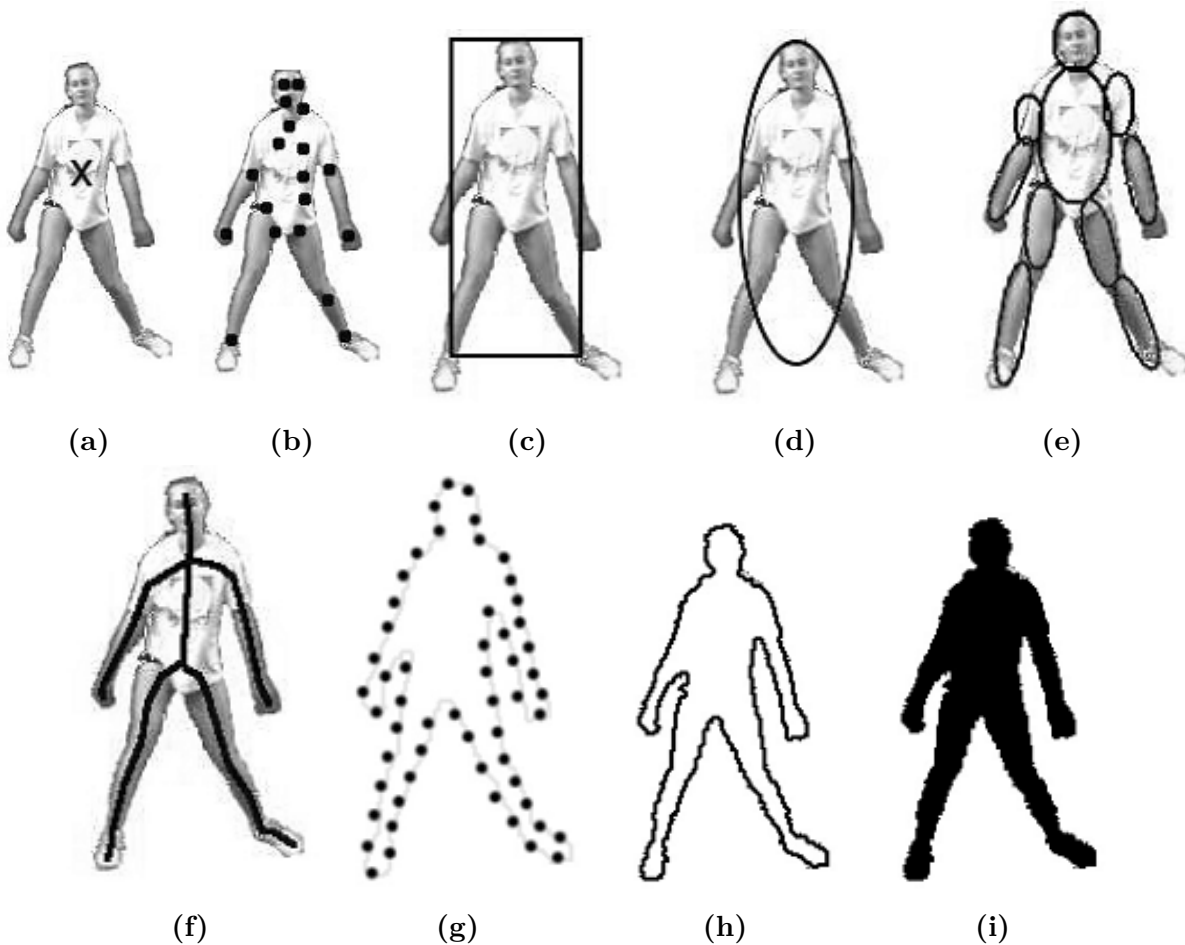
Gauso modeliavimo metu naudojamas spėjimo kitam kadru rasti. Pagal tikimybę, kad kadre objektas pajudės išskiriami objekto ir fono regionai, vaizdui pritaikomas aukšto dažnio filtras surasti nejudančiam fonui (Tokio algoritmo veikimo pavyzdys matomas 1.2 pav. – žmogaus ir judinamo stačiakampio formos objekto. Išskyrimas nėra idealus, kadangi po judesio gaunamas neegzistuojantis objekto atvaizdas išlikęs dėl apmokytos ankstesnės pozicijos.). Sudėtingesnis modeliavimas šios rūšies algoritmas įvykdomas taikant paslėptąjį Markovo modelį klasifikuoti į objektus ir foną, statistiškai apmokant algoritmą.

Adaptyvūs fono išskyrimo metodai Adaptyvūs fono išskyrimo metodai naudoja apmokymą objektams atpažinti. Šie metodai ypač efektyvūs iš anksto žinomai objektams ir fonams. Algoritmai apmokomi, specialiu objektų rinkiniu, ir atsiradus objektams apmokyti algoritmai juos suklasifikuoja į fono segmentus ir objektus. Pats klasifikavimas gali būti taikomas įvairiai, naudojant vektorių palaikymo mašinas (angl. *support vector machine*), tiesinius klasifikatorius ir kitas intelektualiujų algoritmų grupes.

Taškiniai detektoriai Šie metodai grupuoja objektus į požymių zonas, bei šablonus. Taškiniai detektoriai objektus aprašo požymiu rinkiniu, kuris papildomai gali būti praplėstas. Šiame darbe didžiausiais dėmesys skiriamas šiai objektų grupei. Šios grupės esmė yra grupių suskirstymas iš požymių siekiant surasti objektą apibrėžiantį rinkinį. Tam objektui konstruojamas šablonas - apibrėžiančių taškų rinkinys.

1.4. Objekto atvaizdavimas

Objekto atvaizdavimas svarbus žingsnis sekant objektą atvaizduoti ir modeliuoti atmintyje sekimui. Nuo pasirinktos formos priklauso ar algoritmas sugebės tiksliai apibūdinti objektą ir įvykdyti sekimo uždavinį. Sekant objektų forma, judesį dažnai reikia kitokios reprezentacijos, nei tiesiogiai sekant padėtį aplinkoje. Todėl formai išskirta grupė atvaizdavimo metodų:



1.3 pav. Objektų formos atvaizdavimo metodai.

- **Taškinis atvaizdavimas.** Šis metodas objektą modeliuoja kaip vieną tašką centre (centroidą 1.3a pav.) arba taškų debesį(1.3b pav.). Ši reprezentacija naudinga surasti objekto trajektoriją labai mažame regione.
- **Primityvios geometrinės figūros.** Objekto regionas apibrėžiamas stačiakampiu (1.3c pav.) ar elipse (1.3d pav.) ir objektas modeliuojamas tuo regionu. Tokiems objektams judėjimo trajektorija perskaičiuojama poslinkio, afiniosiomis arba projekcinėmis transformacijomis. Šios transformacijos gali būti taikomos tiek pastoviams, tiek kintančios formos objektams modeliuoti.
- **Objekto siluetas, kontūras.** Kontūras(1.3g ir 1.3h pav.) apibrėžia fizinę objekto ribą. Siluetas yra kontūro modeliavimo variantas, kai į modelį patenka ir vidinės objekto detalės. Ši modelio reprezentacija naudojama, kai reikia detalumo objektui apibrėžti.
- **Sudėtiniai objektų modeliai** sudedami iš primityvesnių objektų. Šie modeliai reikalingi norint aprašyti kintančių objektų būsenas ir sekti kitimo procesą. Paprastoms formoms sudarančioms objektą gali būti taikomas atskiras sekimas.
- **Skeleto modelis** (1.3f pav.) išskirtinai taikomas žmonių sekimui, žmogaus siluetą transformuojant į linijas atitinkančias skeleto linijas (išskiriant rankas). Šis modelis naudojamas nustatyti žmogaus, eisenai, pozai ir panašioms judėjimo ypatumams.

1.5. Metodų analizės apibendrinimas

Požymiai ir jų grupavimas svarbus suprasti pačią algoritmų prigimtį ir būdingųjų požymių trūkumus, bei tikėtinus rezultatus. Vadovaujantis bazinėmis savybėmis akivaizdu, kad algoritmų klases sėkmingai implementacijai reikia apžvelgti ir išsirinkti efektyviausiais metodus. Šie metodai yra labai įvairūs ir jų implementacijos skirtingos. Todėl tikslinga aptarti mažesnę grupę algoritmų, kurie naudojami šiame darbe, ši analizė pateikiama 2 skyriuje.

2. ATRINKTŲ ALGORITMŲ OBJEKTAMS SEKTI ERDVĖJE ANALIZĖ

Analizei atrinkta specifinė grupė algoritmų, vadinamų šablono sutapdinimo (angl. *template matching*) algoritmais. Šie algoritmai veikia naudodami šablono iš būdingųjų taškų ir požymių. Bendroji požymių klasifikacija buvo aptarta 1.1 poskyryje. Šiame skyriuje gilinamasi į konkrečius šablono sutapdinimo algoritmus naudojamus sekimui ir jų veikimą.

Išskirtos dvi perspektyviausios tyrimui klasės:

1. Būdingųjų taškų algoritmai (2.1 poskyris),
2. Tekstūros išskyrimo algoritmai (2.2 poskyris).

Šių algoritmų įgyvendinimas įmanomas naudojant vieną kamerą (nereikalinga stereo kamerų įrangos) ir iš vaizdo sukuriama šablono. Fono modeliavimo metodai nenagrinėjami, nes jie neleidžia sukurti objekto šablono – objektas pakeičiamas fono modeliu, t. y. objektas tiesiogiai nemodeliuojamas. Taip pat netiriami metodai reikalaujantis trimačio išankstinio objekto modelio. Šie metodai taip pat tinkami modeliavimui, tačiau reikalauja stereo kamerų sistemos sujungti vaizdams arba išankstinio kameros pozicijos nustatymo ir įvairių parametrų (kaip kameros aukštis, atstumas iki objekto) apskaičiavimo – šablonas tampa priklausomas ne tik nuo objekto, bet ir kameros pozicijos, lęšių savybių.

2.1. Būdingųjų taškų algoritmai

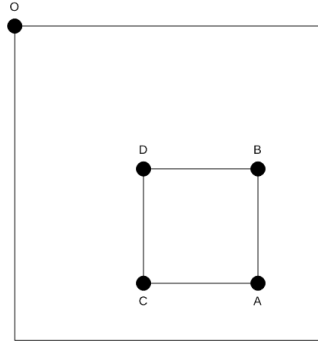
Būdingųjų taškų algoritmai veikia išrenkant vaizde požymius (būdinguosius taškus (angl. *features*)) ir iš jų suformuojant šablono. Toks algoritmas gali naudoti orientuotuosius (algoritmo sugeneruoti požymiai turi posūkio kampą) ir neorientuotuosius (požymio kampo pozicija nesaugoma) požymius. Panašūs struktūra algoritmai gali būti naudojant kraštus (kraštas viena iš paprasčiausių požymių rūšių), tačiau čia aptariami sudėtingesni algoritmai, pasižymintys geresnėmis savybėmis. Kaip būdingųjų taškų radimo algoritmo pavyzdžiai aptarti SURF (Bay *et al*, 2006), BRISK (Leutenegger *et al*, 2011) algoritmai. Šie algoritmai plačiai taikomi praktikoje ir turi kelis programinius įgyvendinimus. Papildomai poskyryje aptariamas nestandartinis požymių išskyrimo FREAK (Alahi *et al*, 2012) algoritmas, modeliuojantis akies tinklainės modeliu.

2.1.1. SURF algoritmas

SURF Algoritmo atranka pradedama išdalinant vaizdą į mažus regionus ir suskaičiuojant intensyvumus regione. Algoritmo pagrindas yra integralinis vaizdas $I_{\Sigma}(x)$ sudarytas iš vaizdo sumos aplink tašką $x = (x, y)^T$, sukuriant filtrą:

$$I_{\Sigma}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j). \quad (2.1)$$

Apdorojus vaizdą taškais sukuriamos vertės vadinamas integralinio vaizdu (angl. *Integral Images*). Integralinio vaizdo esminė savybė tai, kad rezultata lengva panaudoti apskaičiuoti objekto zonos intensyvumui. Intensyvumas apskaičiuojamas nesudėtinga operacija (2.1 pav.).



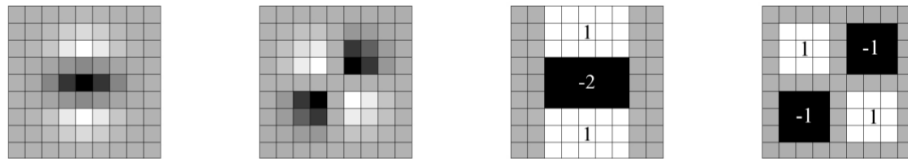
2.1 pav. Integralinio vaizdo intensyvumo skaičiavimas, intensyvumas lygus $\sum = A - B - C + D$ O atžvilgiu .

Šis principas leidžia sukurti detalų vaizdo intensyvumo žemėlapi, kurio regionų intensyvumo vertes apskaičiuoti nėra sudėtinga (regionas apskaičiuojamas kaip jo kraštų verčių skirtumų suma). Intensyvumas susietas tiesiogiai su gradientu.

SURF algoritmas išnaudoja intensyvumus gautus iš integralinio vaizdo. Ne visi intensyvumo regionai yra aktualūs – tik didelio intensyvumo skirtumo sritys, iš kurių SURF algoritmas formuoja požymius. Šios sritys daug atsparesnės apšvietimo pokyčiui, netolygumui nei įprasti spalva grįsti požymiai. Taipogi šie požymiai gali būti orientuoti kryptimi ir šios savybės lemia atsparumą priartinimui (skalės pokyčiams). SURF algoritmas sukuria intensyvumo zoną taške x skale σ (2.2 pav.) :

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (2.2)$$

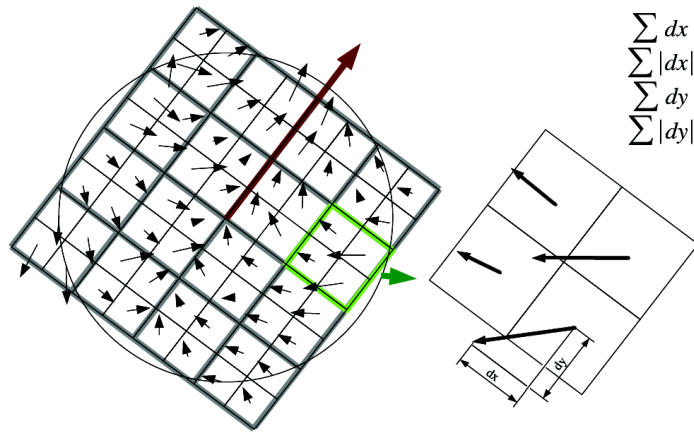
L_{xx} yra dalinė išvestinė x ašimi. Atitinkamai kitos išvestinės kitomis kryptimis. Šios išvestinės



2.2 pav. Intensyvumo antros eilės išvestinės: y (L_{yy}) ir xy (L_{xy}) kryptimi (Bay *et al*, 2006). Šalia pateiktos atitinkamos dalinės išvestinės naudojamos SURF suskaičiuotos iš intensyvumo vaizdo.

sudedamos į būdinguosius taškus (angl. *points of interest*). Šios išvestinės SURF algoritme supaprastinamos į diskrečias vertes (2.2 pav.) - sukonstruojami filtrai, kurių reakcija atitinka intensyvumo kryptį.

Šie filtrai gali būti keičiami dydžiu SURF aptinkant regionus, sukonstruojami iš 9×9 ar 15×15 taškų blokų ir t. t. Regionai su didžiausiu gradientu gali sudedami į didesnę požymių vektorių, kuris naudojamas sudėti orientaciją atrastam taškui 2.3 pav.



2.3 pav. SURF Deskriptoriaus krypties parinkimas (Bay *et al*, 2006)

Vektorius turi kryptį. Požymių vektoriaus centrinis taškas tinka sekti elementus ir turi aiškią orientaciją. SURF požymius vektorių patogiu naudoti sekimui, kadangi jie yra atsparūs skalės ir orientacijos pokyčiams. Šie požymiai taipogi pasižymi svarbia savybe, lokalumu – dažniausiai jie yra unikalūs tik tai sričiai, dėl ko objekto šablonas tinka tik tam konkrečiam objektui. Algoritmas kitaip nei dauguma neorientuotųjų alternatyvų turi daug geresnės patikimumo savybes, taškai nekeičia be reikalo padėties.

2.1.2. BRISK algoritmas

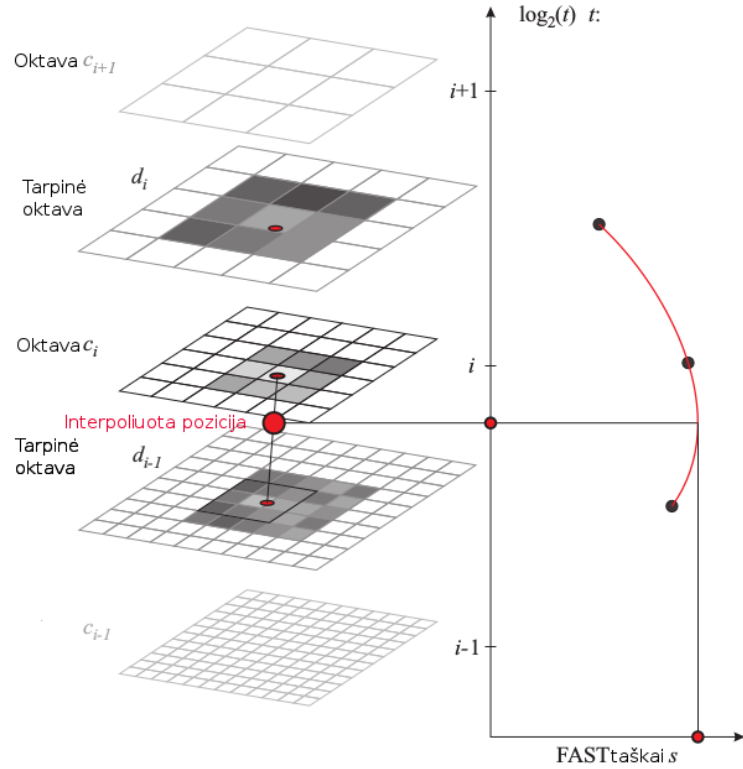
BRISK algoritmas yra panašus SURF taškų lokalumu ir greičiu algoritmas, naudojantis kitokius metodus aprašyti požymiams. Algoritmas turi du požymių išskyrimo žingsnius (Leutenegger *et al*, 2011):

- Budingųjų taškų radimą (angl. Scale-space keypoint detection),
- Taškų apibūdinimą (angl. Keypoint description).

Šis algoritmas labiau optimizuotas greičiui, nei požymių patikimumui (angl. *feature reliability*), nors šiomis savybėmis artimas SURF. Kitaip nei SURF, kuris naudoja kryptingumo sekimui gradientų kompoziciją, šis metodas konstruoja požymių piramidę, su požymi atitinkančia kauke. Piramidės kūrimo metu naudojami neorientuoti FAST požymiai.

Kiekvienam požymių piramidės sluoksnius surandamas primityvesniu FAST detektoriumi. Požymių apibūdinančios piramidės sluoksniams skaičiuojama funkcija surasti optimaliai skalei ir oktavai (požymio priartinimo sluoksniui 2.4 pav.). Jei tarp sluoksnių optimali vertė, specialiai apskaičiuojamas tarpinis žemėlapis interpoliacijos metodu. Papildomai suradus požymius jie aprašomi į mažesnius blokus – tai yra požymių radimas vyksta ne bazinių elementų (filtrų) kompozicijos principu, o vaizdą skaidant į smulkesnes zonas – vykdant dekompoziciją.

Taškų apibūdinimo etapo metu taško kaimyninėms vertėms vykdomas išrinkimas (angl. *sampling*) išdėstant taškus ir individualiai skaičiuojant gradientus. Kitaip nei SURF požymių vektoriuje forma ir veikimo principas skiriasi – naudojamas langas (atskaitai išrinkti dažniausiai parenkamas Gauso langas) gradientams skaičiuoti apie centrinį tašką. Šiame lange skaičiuojami gradientai kiekvienai taškų porai, o bendras kiekvieno taško gradientas lygus individualių



2.4 pav. Fast piramidė naudojama užtikrinti skalės interpoliacijai BRISK algoritme. (Leutenegger *et al*, 2011) Pavaizduota interpoliacija naudojama rasti tarpiniams taškui požymio piramidėje.

gradientų sumai:

$$g = (g_x g_y) = \frac{1}{L} \sum_{(p_i, p_j)} g(p_i, p_j) \quad (2.3)$$

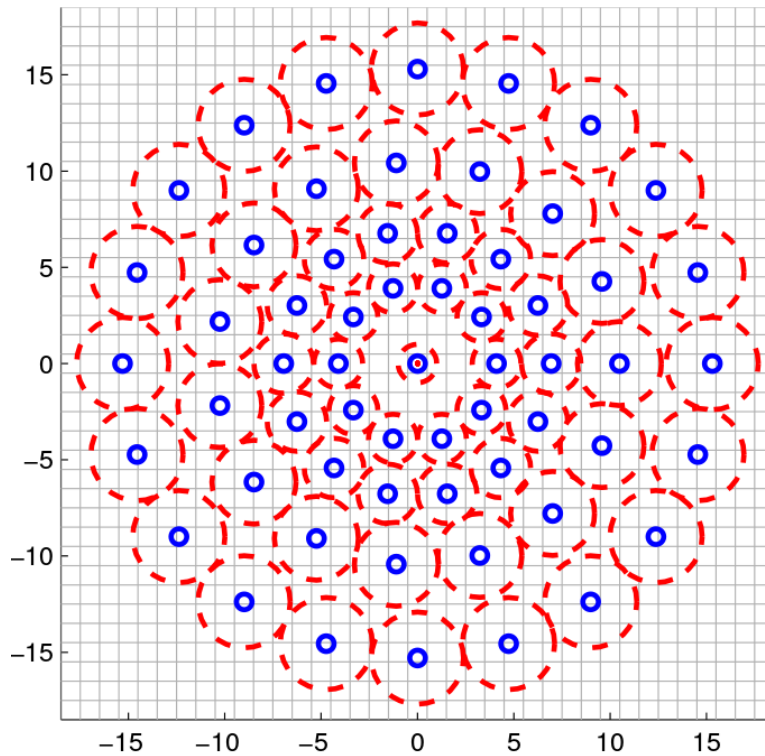
Čia L tolimų taškų porų skaičius. Svarbu pabrėžti, kad lyginamos ne visos poros, tik taškai viršijantys atstumo slenkstį (t. y. tik taškai kurie yra pakankamai dideliu atstumu naudojami gradiento skaičiavime, artimi taškai ignoruojami, nes jų įtaka bendrai gradiento vertei maža). Taipogi taškai išskirstomi ir į grupę kurioje tarpusavio atstumas įvertinamas kaip artimas. Šios poros bus naudojamos vėlesniame etape.

Pagal taškus uždedama orientacija (sektorius kuriame orientuoti požymiai), gaunamas dvejetainė krypties išraiška:

$$b = \begin{cases} 1, & \text{jei } I(p_j^\alpha, \sigma_j) > I(p_i^\alpha, \sigma_i) \\ 0 \end{cases} \quad (2.4)$$

Orientacijai paskaičiuoti lyginamas intensyvumas $I(p, \sigma)$. Kryptingumui skaičiuoti naudojamos tik artimos poros (kurių dalis persidengia).

BRISK kryptių skaičius yra fiksuotas, paprastas įgyvendinimas (kryptingumas yra bitas taškų porai 2.4 formulė) leidžia sumažinti požymio išraiška. Išraiškos dydis, kai ilgo atstumo slenkstis yra $9,75t$, o trumpo atstumo riba yra $13,67t$ (t – minimalus atstumas), tada kryptims užtenka atvaizduoti 512 bitų, kas yra 64 baitai atminties, tai lemia, kad požymių kryptiškumo duomenys labai kompaktiškai išnaudoja atmintį. Dėl to šis požymių vektorius ir vadinamas dvejetainiu požymių vektorius. Kadangi lango elementams naudojama Gauso funkcija galima



2.5 pav. BRISK požymio vektoriaus taškų pasiskirstymas (Leutenegger *et al*, 2011). Punktyrine linija žymimas lango plotis aplink taškus.

laikyti, su vieno bito komponentu, laikoma, kad skaičiuojama Gauso lango slenksčio loginė funkcija išreiškianti orientaciją (kryptingumą).

Apibendrinant, BRISK gana atsparus algoritmas šviesumo pokyčiams, kadangi jam reikia tik taško aplinkos orientacijai rasti, negana to šio algoritmo požymių rinkimo langas yra statinio pavidalo, kas leidžia optimizuoti patį algoritmą ir užtikrina, kad lango operacija vyks pastovia eiga ir išdėstymu (aplink tašką kryptingumas vertinamas vienodai). Kita-vertus piramidės lygmenys ir slenksčiai, bei kintantis taškų skaičius naudojamas požymių vektoriui sukurti, leidžia pritaikyti algoritmą įvairioms situacijoms.

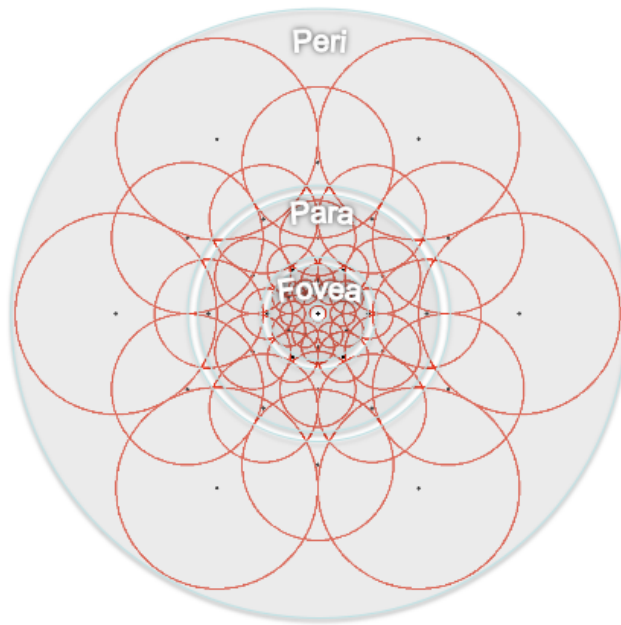
2.1.3. FREAK algoritmas

FREAK algoritmas yra panašus dvejetainiam požymių vektoriui BRISK, bet kitaip nei dvejetainiai požymių algoritmai požymius išsiskiria pagal tinklainės modelį: akies analogo intensyvumo reakciją į vaizdą – taškai kryptingumui nustatyti išdėstomi ne koncentrinio žiedo, bet išrinkimu artimu akies tinklainės neuronams.

Išskiriamos trys sritys atitinkančios biologinį vaizdą (lango pavidalas pateiktas 2.6 pav.):

- centrinė zona atitinkanti objekto centrą, identiška regos centrui – tinklainės duobutei.
- gretima sritis su didžiausiu požymiu tankiu. Dauguma atskaitų šioje srityje persidengia.
- išorinis žiedas, šioje srityje tik dalis žiedo ploto uždengiama atskaitomis.

Požymio vektoriaus langas panašus į BRISK, požymių dvejetainė orientacijos funkcija identiška BRISK(2.4 formulė). Išdėstymas lemia labai išsiskiriančias savybes, algoritmas lokalesnis

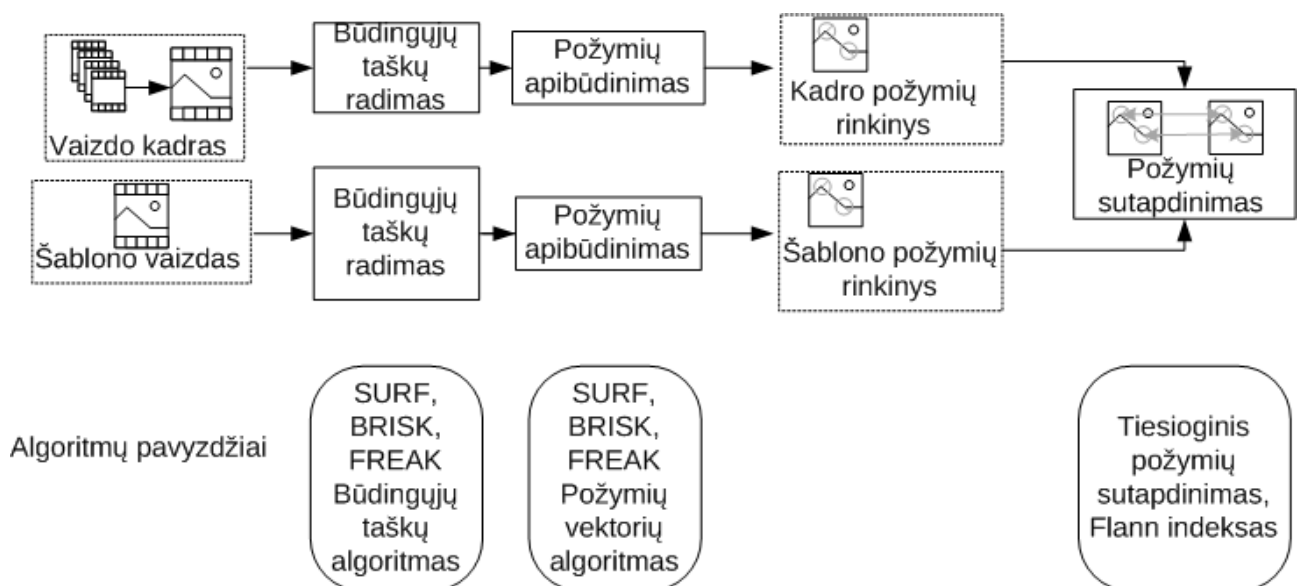


2.6 pav. FREAK deskriptoriaus krypties skaičiavimo taškai, su pažymėtais lango dydžiais ir ekvivalenčiomis biologinei tinklainei sritimis. (Alahi *et al*, 2012)

ir požymiai patikimesni. Šis algoritmas gali būti naudojamas kaip alternatyva BRISK požymių vektoriui, bet taškų išrinkimo žingsnio jis neturi.

2.1.4. Šablono sutapdinimas būdingųjų taškų algoritmuose

Sukonstruoti šablono yra atskiras uždavinys. Šis uždavinys aprašomas paprasta problema: šablone požymius reikia sutapdinti su požymiais ieškomame vaizde. Įvestis susideda iš viso kadro būdingųjų taškų ir apskaičiuotų šablono būdingųjų taškų. Rezultatams gauti reikia nustatyti ar šablono požymiai atitinka požymių grupę iš viso kadro požymių rinkinio.



2.7 pav. Būdingųjų požymių sutapdinimo sistemos veikimas

Vienas iš paprasčiausių metodų, kaip sutapdinti yra lyginti šablono taškus tiesiogiai, naudo-

jant „primityvios jėgos“ sutapdinimą (angl. *Brute Force Matcher*). Šis metodas remiasi taškų rinkimusi ir skaičiavimu, kurie požymiai sutampa. Kadangi realiai taškai yra panašūs savo parametrais, panaudojamas filtras, kuris veikia kaip skirtumų ir smulkių neatitikimų kompensavimo mechanizmas. Filtro rūšių gali būti įvairių. SURF labiausiai tinkamas yra L^2 normos langas, dvejetainių požymių algoritmams labiausiai tinka Hemingo langas (Schaeffer, 2013). Šie metodai yra idealūs dvejetainiams požymiams ir kai kuriais atvejais požymiams be orientacijos - dvejetainiams požymiams lyginti užtenka paprastos XOR operacijos.

SURF požymiams sutapdinti, dėl orientacijos duomenų jau pačiuose taškuose (BRISK požymiai orientaciją įgauna požymių vektorių skaičiuojant, todėl jų sutapdinimas paprastesnis nei SURF požymių, kurie nuolatos turi kryptingumą). Apytiksliai kaimynų paieškai naudojamas FLANN algoritmas pagreitinantis paiešką suskaidydamas taškus į vidurkių medį ir į blokus aplink kiekvieną požymį. Paieškos medis naudoja panašumą tarp šakų. Radus atitinkantį tašką ieškomos gretimos šakos, kadangi tikėtina, kad objektas bus tarp šakų. Taip randami artimiausi požymiai.

2.2. Tekstūros išskyrimo algoritmai

Tekstūros išskyrimo metodai remiasi principu, kad iš gradientų sukurta tekstūra turi aki-vaizdžius kraštus ir orientaciją. Šie metodai labai efektyvūs žmonių ir ryškios tekstūros objektams rasti. Šie metodai įvertina visą paviršių, ir iš viso paviršiaus tekstūros randa svarbiausius regionus, kartais vadinami tankiųjų požymių algoritmais (angl. *dense features matching*).

Šiame poskyryje bus aptartas vienas iš svarbiausių šios klasės algoritmų HOG algoritmas. Šis algoritmas pasirinktas dėl galimybės sekti žmonės ir greito įgyvendinimo vaizdo plokštei.

2.2.1. HOG algoritmas

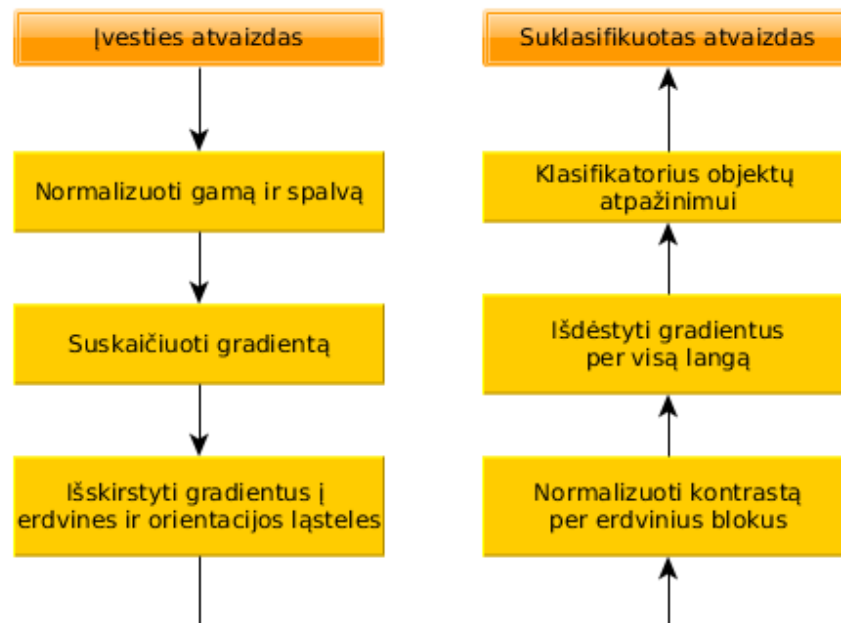
HOG algoritmas susideda kaip ir būdingųjų požymių algoritmų, gradientų aprašymo paveiksle, tačiau neformuojami konkretūs požymiai, užuot paliekamas vektorių regionų rinkinys, kuris vertinamas statistiniais metodais, arba suklasifikuojamas neuronų tinklais (2.8 pav.).

HOG algoritmas vaizdą supjausto į erdvinius blokus. Kiekvienam blokui sudaromi blokai (2.9 pav.). C12 yra erdvinio bloko pavyzdys. Šiame bloke sudaromi mažesni regionai, kuriuose lyginamos anksčiau apskaičiuotos gradientų vertės, sudaromi požymių kryptis. Rezultatai gali būti bloke panaudojami įvairūs langai. 2.9 paveiksle naudojamas $H_{8,1}$ operatoriaus langas iš kurio elementų sudaroma histograma. Jei bloko kryptis atitinka histogramos krypties sritis (angl. *orientationbins*).

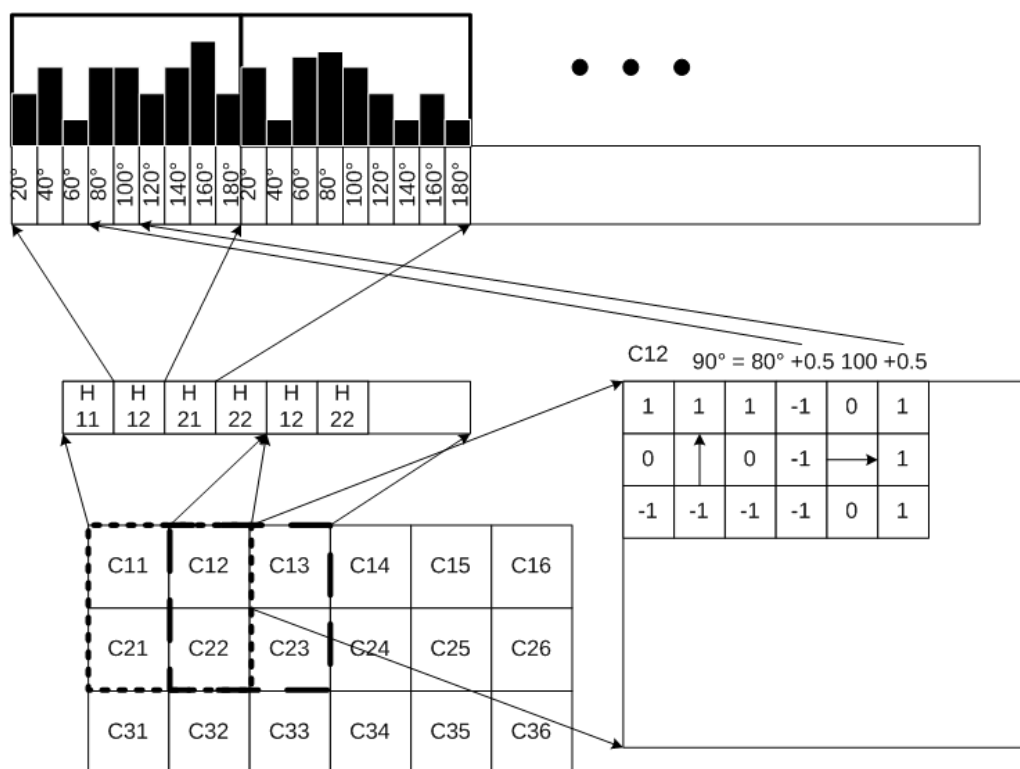
Jei kryptis neatitinka kampo laipsnių, krypties dalis išdalinama koeficientais tarp stulpelių. Kai kurios algoritmo variacijos histogramą sudaroma slenksčio principu: histogramos blokai sudedami diskrečiomis vertėmis, jei kampas patenka į intervalą.

Histogramų vektorius gali būti papildomas ir didesniais langais, kai kuriuose algoritmo variacijose (Wang *et al.*, 2009). Blokai sudedami į didesnius blokus (H_{11} , H_{12} , H_{21} , H_{22} histogramų rinkinys yra tokio bloko pavyzdys).

Šio algoritmo veikimas sukuria aiškų šabloną. HOG algoritmas labiausiai naudojamas klasifikuoti ir surasti tokius objektus kaip automobiliai ir žmogus. Algoritmas labai efektyvus, kai



2.8 pav. HOG algoritmo eiliškumas.



2.9 pav. HOG požymių vektoriaus (histogramų vektoriaus sudarymas).

reikalingas požymių vektorius apibūdinantis labai smulkias detales, defektus, tekstūros pokyčius. HOG algoritmas gali būti panaudotas kaip galutinis požymių vektorius SURF ar kitam taškų išrinkimo algoritmui (Wang *et al*, 2009). Algoritmo pritaikymas paralelizmui labai paprastas, kadangi skaičiavimas vyksta blokais, gradientų vertės skaičiuojamos lokaliai, o gautos vertės sudedamos į histogramą kaip rezultatas. Šis algoritmas įgyvendintas FPGA įrenginiuose, taipogi, kaip vaizdo plokštėse veikianti programa, todėl tai yra šiuolaikiniame taikyme dažnai

sutinkamas algoritmas.

2.3. Trumpas skyriaus apibendrinimas

Šiame skyriuje aptarti orientuoti ir skalės pokyčiui atsparūs požymių algoritmai(2.1 poskyris), bei tekstūros algoritmai(2.2). Šios dvi klasės yra tinkamos sukurti šablonus ir juos sekti.

Nagrinėti (2.1.1., 2.1.2., 2.1.3 poskyriuose) būdingųjų taškų algoritmai sukuria orientuotuosius požymius, kurie yra lokalūs, stabilūs ir apibūdinantys. Šie algoritmai dažnai naudojami tyrimuose, tačiau jie vieni netinka sekimo sistemoms, reikia papildomų algoritmų sujungti juos į sistemą, išgryninti tinkamus sekimui požymius. Skyriuje nagrinėti algoritmai savyje turi integruotus požymių vektorius išgryninti požymiams, kaip alternatyva patikrintas tekstūrų gradientų požymių algoritmas(2.2.1 poskyris), taipogi aptarti sutapdinimo metodai(2.1.4 poskyris).

Visų aptartų algoritmų pagrindas vienodas: gradientai, tačiau skiriasi gradientų kompozicijos metodai, kompozicijos eiliškumas, algoritmų savybės(būdingųjų taškų skaičius, sutapdintųjų požymių skaičius). Išvestys taip pat vienodos: orientacijos vektoriai ir taško centras. Tai leidžia algoritmus naudoti pakopomis, vieną pakeičiant kitu, tačiau dėl vidinių skirtumų sistemos veikimas ir objektų gavimo našumo charakteristikos skiriasi.

3. SEKIMO SISTEMOS GRĮSTOS SOM TINKLU KŪRIMAS

Skyriaus tikslas – aptarti sekimo sistemą objektams. Šios sistemos aktuali savybė yra dirbtinio neuronų tinklo panaudojimas pagerinti sekimo rezultatus. Skyriuje apžvelgtos sekimo sistemos sudedamosios dalys, SOM algoritmo pritaikymas sistemoje. Eksperimentams įgyvendintos sistemos struktūra pateikta parodyti veikimą ir algoritmų įtaką. Šioje struktūroje pateiktas mano siūlomas algoritmo praplėtimas, naudojant dirbtinį neuronų tinklą. Dirbtinio neuronų tinklo formavimo metodika ir bei jo integracija sistemoje aptartos atskirai, kitame poskyryje. Šis poskyris išskirtas, todėl kad neuronų tinklas yra aktualiausia sistemos dalis, išskirianti iš tradicinių sistemų ir yra mano siūlymas kaip įgyvendinti sistemą.

3.1. Objekto sekimo sistema

Poskyryje aptariama objektų sekimo sistema, atskiri komponentai aptarti 2 skyriuje. Taip pat klasikinio sekimo sistemos struktūra buvo pateikta 2.7 pav. Šiame poskyryje sistema nagrinėjama detaliau, pabrėžiami skirtumai nuo įprastos sistemos, apibrėžiama įvestis ir išvestis, sudaromas algoritmas. Taip pat paaiškinamas atskirų komponentų tikslas sistemoje.

3.1.1. Sistemos įvestis

Objektų sekimo sistemos įvestis susideda iš dviejų komponentų: objekto kuris sekamas, ir vaizdo sekos, kurioje vykdomas sekimas. Žinoma, kadangi kalbama apie sistemą, kuri naudoja skaitmeninius algoritmus sekimui, įvesties duomenys yra skaitmeninio pavidalo.

Įvesties seka yra vaizdo įrašas, arba atskirų vaizdų rinkinys, tiesiog atskiri vaizdai – sekos pasirinkimas svarbus kiek laiko objektas sekamas. Sekti objektą galima reguliariai, apdorojant kiekvieną kadrą. Taipogi įmanomas sekimas periodiškai atnaujinant poziciją, kai objekto pozicija kadre svarbi tik esamą akimirką, arba naudoti tik pavienį tapdinimą objekto identifikacijai. Šioje sistemoje tiriamas pastovus sekimas, kadangi objekto identifikacijos uždavinys neaktualus (tokiu atveju sutapdinimas vykdomas su šablonu iš šablonų duomenų bazės, sekimo periodas reguliaraus taškų pozicijos atnaujinimui dėmesio neskiriama – visiškai priešingai šiam tyrimui, kur sekimo, atnaujinimo operacijoms skiriamas didžiausias dėmesys). Turint laiko atžvilgiu tolydų vaizdo įrašą mažesnės aplinkos sąlygų pokyčių tikimybė (mažesnė šviesumo pokyčių galimybė, tolydesnis objekto poslinkis tarp kadru). Kita vertus numatoma, kad įgyvendinti pakeitimai veiks ir su kitokio pobūdžio įvesties duomenimis.

Šablono įvesties šaltinis parenkamas įvairiai. Parinkti galima iš vaizdo sekos, pažymint vieną kadrą, kuriame objekto regionas svarbus. Šis metodas pasirinktas šiai sistemai. Naudojant šabloną iš vaizdo sekos, pašalinami skirtumai tarp šviesumo (jei tokie skirtumai kliūtys). Šis šablono kūrimo metodas tinka realaus laiko sekimui: pažymėjus objektą viename kadre, objektas sekamas tolimesniuose kadruose. Alternatyva yra iš anksto paruoštas šablonas – ankščiau gauta objekto nuotrauka. Šis metodas irgi tinkamas šablono kūrimui, bet siekiant optimalaus sutapdinimo rezultato reikia įvertinti skirtumą tarp kameros vaizdo ir šablono nuotraukos

3.1 lentelė. Algoritmų kombinacijos

Būdingųjų taškų radimo algoritmas	Būdingųjų taškų išgryninimo algoritmas	Sutapdinimo algoritmo langas
SURF	SURF	FLANN indeksas
	SURF	L2 langas
	BRISK	Hamming langas
	FREAK	Hamming langas
BRISK	SURF	L2 langas
	BRISK	Hamming langas
	FREAK	Hamming langas

(priartinimas, identiškos šviesumo skirtumo problemos).

Būdingieji taškai remiasi intensyvumu ir gradientais (aprašyta 1.1 ir 2.1 poskyriuose). Kad-
rų spalva algoritmams didelės įtakos neturi, nors egzistuoja variantai išnaudojantys spalvą –
spalvinėje srityje intensyvumo pokyčiai nėra aktualūs, nebent objektas turi aiškiai išreikštą
spalvą. Įvesties duomenys privalo būti paversti nespaltvotais – spalvų skalė (RGB, BGR) turi
būti pakeista į *grayscale*.

3.1.2. Sistemos struktūra

Sistemos susideda iš komponentų šablonui ir vaizdo įrašui apdoroti. Šie komponentai nau-
dojami apdoroti įvesčiai (3.1.1 poskyris). Sistemos šablonas paruošiamas vieną kartą kaip
požymių vektorius.

Šablono ir vaizdo paruošime yra šie etapai: vaizdo kadro vertimas į *grayscale* (spalvos nėra
svarbios šiems požymiams), būdingųjų taškų radimas, būdingųjų taškų aprašymas požymių
vektoriumi. Vaizdo apdorojimas tęsiamas tol kol vaizdo seka baigiasi (3.1 pav.).

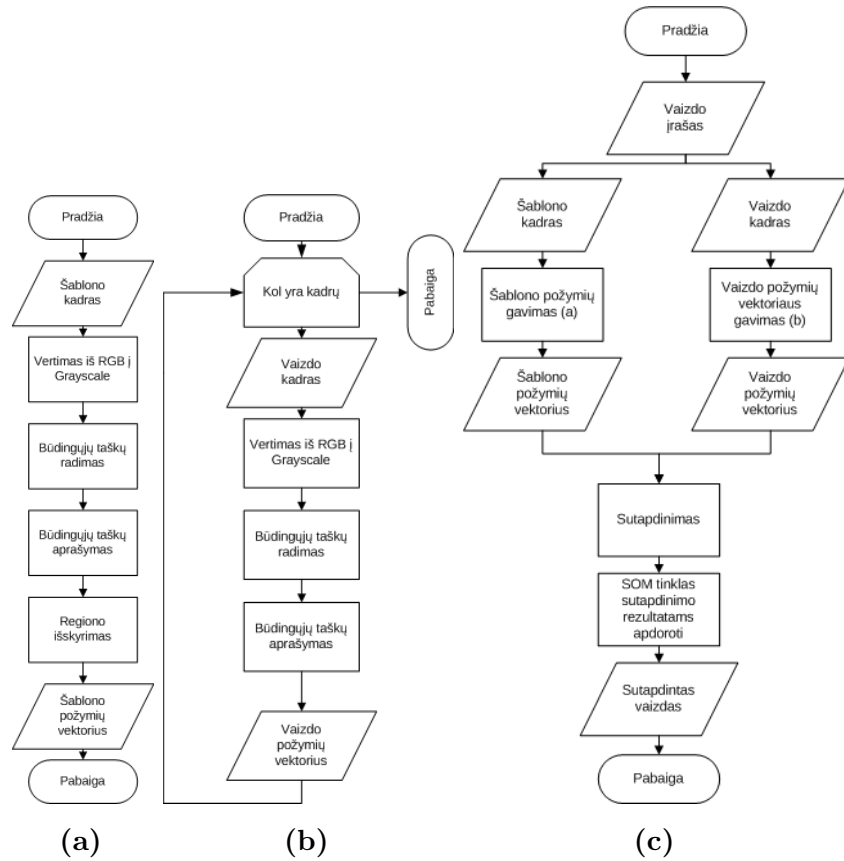
Sistemoje svarbiausias išvestyje sutapdinimo blokas. Šabloną ir vaizdą suformavus požy-
mių vektoriumi, vykdomas tapdinimas, kurio metu šablonas uždedamas ant vaizdo. Vaizdo
sutapdinimo uždavinys nėra paprastas, kadangi vaizdas sutapdinamas iš anksto nežinant kur
ieškomas objektas: sutapdinimas vykdomas visam vaizdai. Šiam algoritmui bus skiriama dau-
giau dėmesio kitame poskyryje (3.2.1).

Sistemos algoritmo kombinacijos sudaromos iš būdingųjų taškų radimo, būdingųjų taškų
aprašymo ir sutapdinimo algoritmų. Taškų radimui naudojami SURF ir BRISK algoritmai (3.1
lentelė). Trečias tirtas komponentas (FREAK) neturi savo taškų išrinkimo algoritmo. Sutap-
dinimo atstumui skaičiuoti SURF algoritmas nenaudoja Hamming normavimo, užuot taikomas
kvadratinio polinomo L2 langas. SURF alternatyviai naudoja FLANN medžio indeksą.

3.2. SOM algoritmo pritaikymas vaizdo sekimui

3.2.1. SOM algoritmas

SOM yra viena iš dirbtinių neuronų tinklų rūšių plačiai taikomų įvairiose srityse. Šis tinklas
kitai žinomas Kohonen'o tinklu (Kohonen, 2001). Algoritmas sprendžia įvairius uždavinius,



3.1 pav. Šablono (a) ir kadro (b) požymių sukūrimo, bei sistemos (c) algoritmai

nuo segmentavimo (Jiang *et al*, 2003), iki objektų klasifikacijos pagal poziciją uždavinių (Somervuo, Kohonen, 1999). Algoritmas turi dvi veikimo fazes: mokymąsi ir klasifikavimą. Šis algoritmas veikia išdėstydamas atsitiktinius svorius į kelių dimensijų žemėlapi, ir mokydamasis su įvestimi sudaro naują žemėlapi, pagal duomenų įvestį. Išmokytas žemėlapis gali būti panaudotas klasifikacijai: Neuronų tinkle egzistuoja laimėtojas neuronas, kuris reaguoja į įvestį ir priskiria prie neuronų apmokymo rezultatų.

Apmokymo algoritmas išreiškiamas keliais žingsniais:

1. Parinkti tinklui svorius. (Tradiciiniame algoritme svoriai parenkami atsitiktinai).
2. Gauti įvesties duomenų iš vektoriaus $D(t)$
3. Pereiti per kiekvieną mazgą tinkle:
 - (a) Kiekvienam mazgui paskaičiuoti atstumą iki taško įvesties duomens.
 - (b) Pasirinkti neuroną kuris yra laimėtojas.
4. Atnaujinti visus neuronus, įskaitant laimėtoją arčiau įvesties vektoriaus vertės:

$$W_v(s+1) = W_v(s) + \Theta(u, v, s)\alpha(s)(D(t) - W_v(s)) \quad (3.1)$$

5. Adaptuoti žingsnio plotį nuo iteracijos.

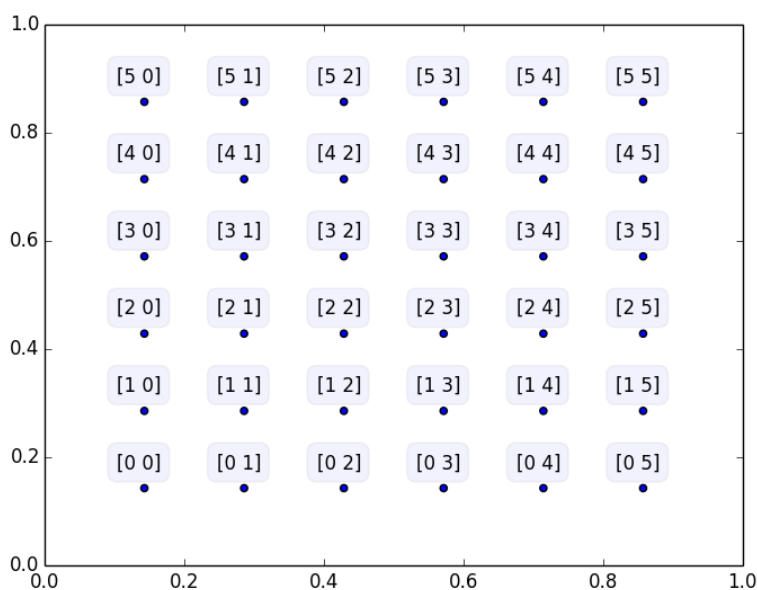
3.2 lentelė. SOM tinklo 6x6 svorių pradinis išdėstymas

Mazgo indeksas	Mazgo pozicija	
	x	y
(0,0)	1/7	1/7
(0,1)	1/7	2/7
:	:	:
(2,2)	3/7	3/7
:	:	:
(5,5)	6/7	6/7

Čia s – iteracijos skaičius, W_v – esamas svoris iteracijos metu, Θ – kaimyninio mazgų aktyvacijos funkcija (dažniausiai lango funkcija), u – laimėtojo neurono indeksas, v – kito (tikrinamo) neurono indeksas, $\alpha(s)$ – stabdymo koeficientas priklausomas nuo iteracijų skaičiaus.

Rezultate gaunamas tinklas gali būti naudojamas patikrinti, kuris neuronas laimės: tai yra klasifikacijai. Tinklo neuronai gali būti panaudojami kvantavimui, kitiems uždaviniais. Aktyvuoto neurono koordinatės galima panaudoti ir aproksimacijai.

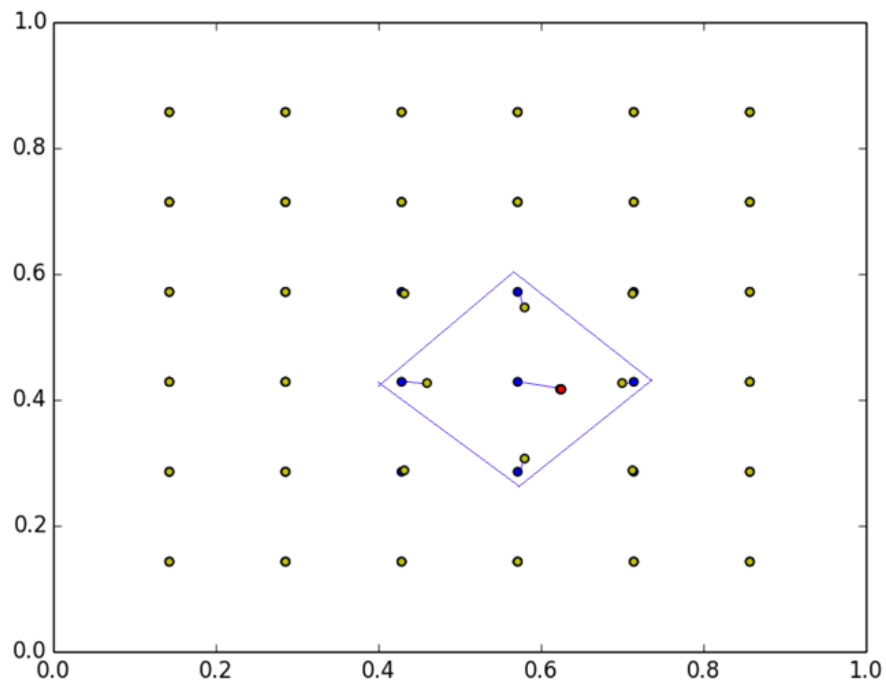
3.2.2. SOM algoritmo pritaikymas vaizdo sekimui



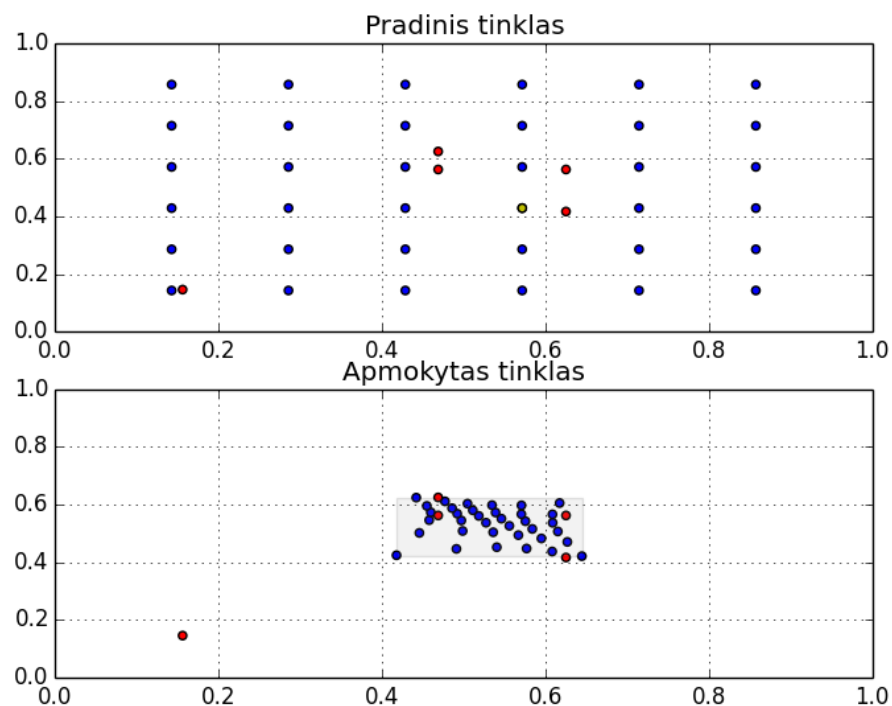
3.2 pav. SOM tinklo pradinė neuronų padėtis

Kad pritaikyti SOM algoritmą vaizdo sekimui, atsižvelgta į tai, kad SOM tinklo svoriai gali būti išdėstyti plokštumoje (3.2 lentelė, 3.2 pav.). SOM tinklo neuronų skaičius yra indeksai, kurie tiesiog nurodo neurono eilę apmokymo metu. Įprastame SOM algoritme pradiniai koeficientai pasirenkami atsitiktinai, arba iš pradinio duomenų rinkinio. Šioje algoritmo variacijoje vaizdo sekimui buvo panaudota kita pradinė apmokymo schema – pradinė svorio koeficientų vertės išdėstomos plokštumoje.

Įvesties neuronų tinklas, jei duomenys sunormuoti tam pačiam intervalė, leidžia tinklui padengti visą duomenų intervalą. Pavyzdžiui, jei yra kadras kurio rezoliucija yra 640 taškų x ašimi, 480 taškų y ašimi, įvesties taškus tinklui apmokyti realios koordinatės reikia su normuoti, pagal tos krypties koordinates. Tinklo apmokymo metu svoriai ir duomenys sunormuojamas nuo 0 iki 1, 0.



3.3 pav. SOM tinklo poslinkis apmokius tinklą vienu tašku. Pažymėtas didžiausios dirbtinių neuronų tinklo aktyvacijos regionas



3.4 pav. SOM tinklo taškų santalkos srities apskaičiavimas.

Vaizdo koordinatėms panaudojus šablona ir išvesties koordinatės rezultate gaunamas poslinkis. Neuronų tinklas susiorganizuoja į grupę. Tinklas deformuojasi pagal įvesties duomenis. Įvedus vieną objekto poziciją (jei objektas modeliuojamas centroidu) tinklo dirbtiniai neuronų mazgai keičia poziciją link objekto surastos pozicijos. Mokymo metu į įvesties argumentai parenkami vaizdo šablono rasti taškai.

Apmokymo metu neuronų tinklas patiria poslinkį link taškų sankaupos. Naudojant centroidą tinklas deformuojasi link vienos centrinės pozicijos. Jei sekamas objektas susideda iš kelių taškų tinklas dreifuoja į tų taškų srautą. Vaizdo sekimui šios savybės naudingos. Pirmą savybę, kad vaizdo tinklas leidžia nustatyti objekto poziciją jei, objektas randamas keliais skirtingais metodais. Ši savybė yra atitinka pozicijos vidurio radimą. Parinkus pakankamai didelį langą tinklui tinklas neteks išorinių taškų (angl. *outliers*). Bandymo metu rasta, kad esant dideliui atnaujinimo plotui $\Theta = 1, 5 - 2, 2$, iteracijų skaičiui 100, mokymo greičiui 0, 1 pasiekiamas tinklo kontūro taškų santalkos sričiai. Alternatyviai lokali santalka gaunama naudojant langą su parametru $\Theta = 1, 0 - 1, 2$, iteracijų skaičiui 100, mokymo greičiui 0, 1.

Išorinių taškų pašalinimas leidžia adaptyvų šablono dydžio valdymą ir išskyrimą. Šablonas užuot turėjęs pastovų kriterijų, kuriuo vykdomas šalinimas, klaidingus taškus gali nustatyti pagal taškų santalkos sritį. Po apmokymo taškai susigrupuoja į sritį, kurios dydis nustatomas pagal taškų, koordinatės. Paprasčiausiam skaičiavimui panaudojus sukuriama stačiakampė paieškos sritis (3.4 pav.). Šios srities koordinatės išreiškiamos:

$$x_{0w} = \min(x(W_v(s))) \quad (3.2)$$

$$y_{0w} = \min(y(W_v(s))) \quad (3.3)$$

$$x_{1w} = \max(x(W_v(s))) \quad (3.4)$$

$$y_{1w} = \max(y(W_v(s))) \quad (3.5)$$

Būdingiems taškams ($p(x, y)$) rasti ar jie regione ($r(x, y)$ – loginė matrica) pritaikomos konjunkcija:

$$r(x, y) = (x_{0w} \leq p(x) \leq x_{1w}) \wedge (y_{0w} \leq p(y) \leq y_{1w}) \quad (3.6)$$

Jei $r(x, y) = 1$ taškas $p(x, y)$ įeina į sritį. Šis tikrinimas paprasčiausias neuronų tinklo panaudojimas. Kita neuronų tinklo implementacija pašalinti išnaudoja išorinį neuronų tinklo sluoksnį. Sąlyga pakeičiama, kad laimėjęs tinklo neuronas $W_{vw}(p)$ su indeksu v patenka į vidinį neuronų sluoksnį.

$$r(x, y) = \begin{cases} 1, & \text{jei } W_{vw}(p(x, y)) \mapsto 0 < v_x < v_{x\max}, 0 < v_y < v_{y\max}; \\ 0 & \end{cases} \quad (3.7)$$

v_x ir v_y yra neuronų tinklo sluoksnių indekso x ir y pozicijos dedamoji.

Be apribojimo galimybių sistema sugeba ir SOM tinklą panaudoti sekimui, dėl tinklo santalkos. Lokali santalka randama apskaičiavus atstumą tarp artimiausių neuronui taškų. Sunormavus pagal didžiausią atstumą ir įvedus slenkstį atmetami tolimesni taškai esantys toliau nuo santalkos.

3.3. Trumpas skyriaus apibendrinimas

Sistamai reikia trijų dalių: požymių radimo, požymių išskyrimo ir sutapdinimo. Šios dalys gal būti sukuriamos ir iš skirtingų algoritmų: SURF ir BRISK, BRISK ir FREAK, bei kitų kombinacijų (3.1). Sutapdinimo žingsniui reikia išorinių taškų atmetimo metodų. Tam įgyvendinti naudojamas SOM dirbtinis neuronų tinklas. Šio algoritmo veikimo principas grįstas dirbtinio neuronų tinklo išdėstymu plokštumoje ir tinklo svorių pozicijos judėjimu mokantis. Esant dideliame atnaujinimo pločiui $\Theta = 1,0 - 1,2$, iteracijų skaičiui 100, mokymo greičiui 0,1 pasiekiamas tinklo kontūro atitikimas santalkos sričiai. SOM tinklo apmokymas organizuojamas sunormavus koordinates plokštumoje $0,0 - 1,0$ intervale. SOM tinklo klasifikavimo naudojimui pasiūlytos trys strategijos: dirbtinį neuronų tinklą atitinkančios srities gaubtinės išskyrimas ir išorinio neuronų tinklo sluoksnio aktyvacijos srities atmetimas, lokalios santalkos srities išskyrimas.

4. DUOMENŲ ATRANKA OBJEKTŲ SEKIMO METODŲ TYRIMUI

Duomenys objektams sekti parenkami neatsitiktinai. Sekimo algoritmams tikrinti pateikiamos video įvestys, su papildomai duomenimis išreiškiančiais pavyzdžio išrinkimo srityje (angl. *regions of interest*). Duomenys parinkti, kad atitiktų aplinkas ir scenarijus sutinkamus objektų sekimo algoritmų taikyme.

4.1. Įvesties duomenys

Sistema veikia sukurdamą šablonus iš vaizdo kadro. Su šiais šablonais objektas yra nuolat sekamas, kiekviename kadre panaudojant šablono duomenis sutapdinti šabloną su kadru: įvykdyti kadro radimą. Taigi sistemos veikimui reikia dviejų komponentų:

- vaizdo sekos (4.1.1 poskyris),
- šablono (4.1.2 poskyris).

Skirstymo pagrindumas ir motyvai aprašomi 3.1.1 poskyryje. Šis skyrius panaudotas patiems duomenims apibūdinti. Vaizdo seka yra video failai naudojami sekimui, parinkti pagal įvairius scenarijus. Šablonas – tai ieškomo objekto atvaizdas pagal kurį vyks paieška.

4.1.1. Vaizdo sekos

Vaizdo sekų sukūrimui panaudoti atvejai įprasti objektų sekimo taikyme. Sekime svarbūs trys komponentai: objekto padėtis, fono statiskumas, kameros padėtis. Remiantis komponentų klasifikacija sukurtos sekos pagal tokius scenarijus:

1. statinė kamera filmuoja statinį foną, objektas juda scenoje. Šios vaizdo sekos rezultatas yra artimas idealiam scenarijui. Kameros pozicija nekinta, todėl kameros įtaka minimali objekto sekime.
2. statinė kamera filmuoja dinaminį foną, objektas juda scenoje. Šioje sekoje algoritmai fone nuolatos pagauti vis naujus požymius, kas apsunkina algoritmo veikimą.
3. statinė kamera filmuoja statinį foną, objektas netik juda scenoje, bet ir keičia formą, bei posūkio kampą. Šablono sutapdinimas šioje sekoje sunkus.
4. statinė kamera filmuoja statinį foną, objektas juda scenoje. Aplinkoje galimi atspindžiai (vanduo, veidrodys).
5. statinė kamera filmuoja statinį (ar beveik statinį) foną, objektas pastovus. Kinta apšvietimo sąlygos. Ši scenarijaus seka skirta nustatyti ar vaizdas priklauso nuo apšvietimo.

6. judanti kamera filmuoja statinį objektą. Šis scenarijus sunkesnis nei pirmasis, kadangi vaizdo seka priklauso nuo kameros savybių, jei kamera be stabilizatoriaus, vaizdas išgautas video daug sudėtingesnis dėl įvairių veiksmų (virpėjimo, fokusavimo ir t.t.).
7. judanti kamera filmuoja objektą su sudėtinga tekstūra. Šis scenarijus padeda išsiaiškinti taškų radimo savybes ir kaip jos įtakoja vaizdo radimą.

Paruošti vaizdo įrašai turi 1920×1080 taškų dydį. Vaizdo įrašai spalvoti, suspausti H264 vaizdo kodeku. Sistemos įvesčiai panaudojami išskleisti kadrai, spalvų keitimas numatytas sistemoje (3.1a).

4.1.2. Šablonai

Šablonų formavimas aktualus išrenkant objektą iš vaizdo sekų. Šablono formavimui buvo pasirinktas metodas išrinkti objektą iš kadro ir jį sekti. Objektas iš sekos parinktas todėl, kad netinkamas iš anksto paruoštas šablonas gali turėti įtakos vaizdo sekimo rezultatams (dėl šablonų kūrimo aplinkos skirtumo, dėl šablono skalės neatitikimo ir daugelio kitų veiksmų).

4.1 lentelė. Įvesties duomenų šablonui generuoti formatas.

video	template	roi_x	roi_y	roi_w	roi_h
rt_3.mkv	rt_3.png	533	270	462	240



4.1 pav. Šablono išskyrimas iš kadro, pagal formato lentelės duomenis, vaizdo įrašas atitinka pirmą scenarijų.

Vaizdo įrašas šablono kadrai išgauti ranka parenkamas kadras ir langas jame. Kadras išskirtas (4.1 pav.) iš vaizdo šablonui saugomas atskirai. Toks sprendimas parinktas todėl, kad tyrimai daromi naudojant iš anksto paruoštus vaizdo įrašus, o vaizdo įrašą prasukti iki reikiamo kadro

sudėtingiau programiškai ir kompiuterio skaičiavimo išteklių atžvilgiu, nei išsaugoti kadrą atmintyje/diske ir jį užkrauti. Kadro aktuali sritis išsaugoma kaip duomenys. Įprastai realaus laiko sistemose, kuriose kadrą sužymimas šablonas iš karto saugomas atmintyje.

Šablonui nuspręsta iš sekos išrinkti vieną kadrą ir jam pažymėti regioną kuris bus sekamas. Eksperimentų duomenims parinktas regiono sritį atitinkančios koordinatės surašytos į lentelę(formato laukų pavyzdys pateiktas 4.1 lentelėje). Išskiriama vaizdo seka(*video* laukas), šablono kadras(*template* laukas), paieškos sritis (*roi* – angl. *regions of interest*) apibrėžiama koordinatinių laukais (*roi_x*, *roi_y*, *roi_w*, *roi_h* – srities viršutinio kampo x, y koordinatės, plotis, aukštis).

4.2. Trumpas skyriaus apibendrinimas

Objektų sekimui reikia dviejų rūšių duomenų: tiriamos vaizdo sekos ir šablono naudojamo sekimui. Objektų sekimui vaizdo sekų scenarijai klasifikuojami pagal: kameros poziciją, objekto statiškumą ir aplinkos statiškumą.

Šiems scenarijams parengtos vaizdo sekos. Šablonai parengti atskirais kadrais ir su svarbos sričių aprašu. Toks metodas pasirinktas siekiant atskirti šablono gavimą nuo bendro vaizdo apdorojimo.

5. OBJEKTŲ SEKIMO METODŲ TYRIMO REZULTATAI

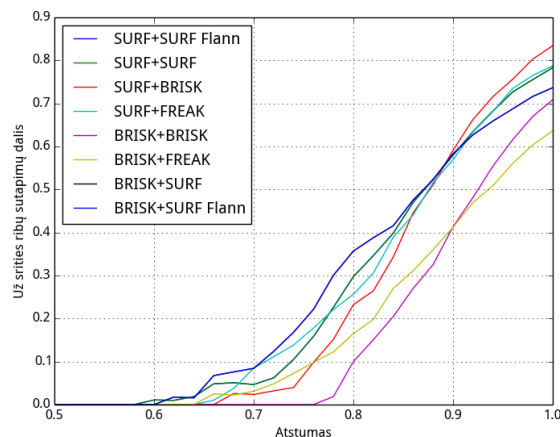
Objekto sekimo sistemai patikrinti reikalingi eksperimentai. Eksperimentai daromi naudojant 4 skyriuje aprašytus įvesties duomenis. Šiuo skyriaus tikslas pateikti tyrimo duomenis ir rezultatus, aptarti eksperimentų metodiką. Išskirtos dvi tyrimų grupės: bendras algoritmų tyrimas ir SOM tinklo sutapdinimo tyrimas.

5.1. Požymiams ir sutapdinimui naudojamų algoritmų tyrimas

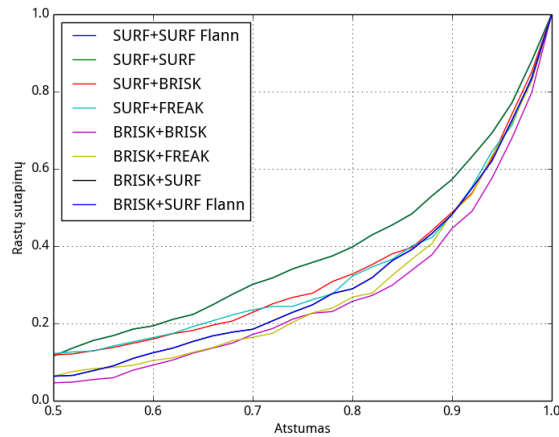
Sutapdinimui naudojamų algoritmų tyrimas skirtas atrinkti efektyviausias algoritmų poras sekimui. Algoritmui tyrimui parinkti aprašyti 2 skyriuje algoritmai. Algoritmų rinkinių yra labai įvairių (3.1 lentelė), sistemos tyrimo metu pasirinktos šios kombinacijos, kurios buvo naudojamos lentelėje.



5.1 pav. Taškų sutapdinimas, naudojant BRISK ir Hamming lango indeksą, kai nustatyta 50 artimiausių sutapimų.



5.2 pav. Klaidingo sutapdinimo santykis palyginimas pagal atstumą tarp sutapdinimų.



5.3 pav. Taškų sutapdinimo santykio palyginimas pagal atstumą tarp sutapdintų taškų.

Sistemos šablono sutapdinimui paprasčiausias eksperimentas patikrinti ar sutapdinimas įvykdomas. Požymiams nustatomas pastovus slenkstis, pagal atstumą kurie požymiai atvaizduojami. Paprasčiausias metodas patikrinimui, yra nustatyti pastovų požymių kiekį, požymiai rikiuojami, pagal mažiausią atstumą. Riboto sutapdinimo pavyzdys pateikiamas 5.1 pav.

Taškų skaičius sutapdinimo kokybei tirti buvo sutapdinti vaizdai paeiliui su skirtingais algoritmais ir tikrinta koku atstumu sutapdinimo rezultatų poros buvo viena nuo kitos (artimiausio kaimyninio sutapimo kriterijus). Kuo sutapdintų taškų atstumas mažesnis, tuo geriau įvykdomas sutapdinimas. Atstumui tirti parinkti kadrai su ranka išskirtai sutapdinimo sritimis. Sritis yra stačiakampis regionas turintis koordinatas, matomas objekto patekimas į šią sritį. Sritis parinkta vizualiai aplink sutapdinamą objektą, objekto srities plotis už kraštų gali išeina iki 10 pikselių, kas sudaro mažiau nei 10 procentų srities vaizdo.

5.1 lentelė. Optimalus atstumas esant pasirinktai klaidos tolerancijai

Algoritmų rinkinys	atstumas esant 5%	atstumas esant 10%	atstumas esant 15%
SURF-SURF-L2	0,70	0,74	0,76
SURF-SURF FLANN	0,66	0,72	0,74
SURF-BRISK	0,66	0,72	0,74
SURF-FREAK	0,69	0,71	0,74
BRISK-BRISK	0,79	0,8	0,82
BRISK-SURF	0,70	0,74	0,76
BRISK-FREAK	0,72	0,76	0,79

Objektui tikrinti kaimyninio sutapimo taškas parinktas intervale atstumo santykio intervale $0,5 - 1,0$, $0,01$ žingsniu kaimyninio taško sutapimai skaičiuojami poromis: pagrindiniam taškui m tikrinamas tik artimiausias kaimyninis taškas (n). Pagrindinis taškas laikomas surastu, kai aktyvuojama funkcija $m < a \cdot n$, kur a yra atstumo santykis. Ši sąlyga leidžia rasti neišsibarsčiusius taškus esančius vienoje santalkoje su minimaliu atstumu. Atstumo koeficientui rasti buvo panaudotos dešimtys kadru, požymių duomenims skaičius siekia virš 5 tūkstančių.

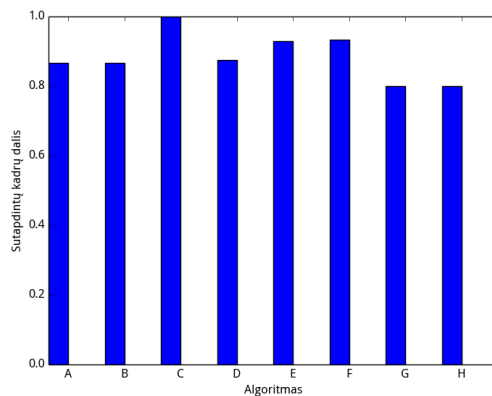
5.2 pav. palyginti skirtingi algoritmai, klaidingų sutapdinimų atžvilgiu. Klaidoms skai-

čiuoti naudojamas santykinis įvertis nuo visų sutapdintųjų taškų. Augant atstumui, klaidingų sutapdinimų (esančių už pažymėtos srities ribų) skaičius didėja. Tačiau taip pat didėja ir sutapdintų taškų skaičius (5.3). Geriausiu atveju leistiną klaidos ribą galima nustatyti kuo didesnę, siekiant padidinti požymių skaičių.

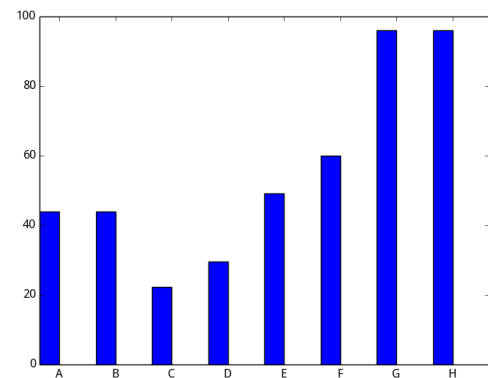
Stebint algoritmus (5.1) pastebėta, kad Visų algoritmų atstumas panašus, ir patenkinamą klaidą galima gauti ties 0,7 atstumas. Šis pastebėjimas negalioja BRISK algoritmui, kuris optimalus atstumas artimas 0,8.

5.2. SOM grįsto objektų sekimo algoritmo tyrimas

SOM neuronų sistema naudojama sutapdinimo rezultatų. SOM tinklo tyrimas skirtas nustatyti kaip ši sistema sugeba surasti taškus bandomojoje vaizdo sekoje. Neuronų tinklo santalka buvo pasirinkta nustatyti rasto objekto poziciją kadre. Santalkos nustatymui buvo panaudotas atmetimas naudojant atstumą, jei kaimyninis neuronas siekia daugiau maksimalaus atstumo santykinio koeficiento (k_r , kuris sudaro kaimyninio neurono paieškos sąlygą $m_{SOM} < k_r \cdot n_{SOM}$) neuronas atmetamas. Bandymo metu pastebėta, kad geriausias atmetimas neuronų tinkle gavo si nustačius 0,3 atmetimo ribą. Jei maksimalus atstumas tarp neuronų lygus 1, tai visi neuronų tinklo mazgai kurių tarpusavio atstumas mažesnis nei 0,3 bus santalkoje.



(a)



(b)

Sistema	Žymuo	Sutapdinimų santykis	Nuokrypis nuo centro
SURF-SURF FLANN	A	0,87	43,86
SURF-SURF-L2	B	0,87	43,86
SURF-BRISK	C	1,0	22,22
SURF-FREAK	D	0,875	29,50
BRISK-BRISK	E	0,93	49,21
BRISK-FREAK	F	0,93	60,05
BRISK-SURF	G	0,8	96,04
BRISK-SURF FLANN	H	0,8	96,04

5.4 pav. Teisingo sutapdinimo santykio palyginimas pagal sėkmingai sutapdintų kadrų skaičių (a) ir vidutinį atstumą tarp kadrų (b).

Su sistema atlikti bandymai patikrinti kurios rūšies sistemos su neuronų tinklu veikia ge-

riausiai (5.4 pav.). Visos sistemos veikė panašiu sutapdinimo atžvilgiu skaičiumi (apie 0,9 (90%) teisingų sutapdinimų, atsitiktinai išrinktuose kadruose). Sutapdinimo nepilnas atradimas paaiškinamas taškų trūkumu dėl skalės pokyčio (objekto vaizdai parinkti su objektu, tačiau skirtingo dydžio sritimis). Pačiam sutapdinimo tikslumui reikalingas įvertis. Buvo pasirinktas srities centras ir vertinta koku Euklido atstumu pikseliais sutapdinto objekto centras nutolęs taško padėties. Geriausiai pasirodė sistema grįsta SURF taškų išrinkimu ir BRISK požymių išskyrimu. BRISK išskyrimas pasižymėjo didžiausiu tikslumu. Šio algoritmu tirtas SOM sutapdinimas turėjo mažiausią nuokrypį. SURF požymio radimo sistemos pasižymėjo mažesniu klaidų skaičiumi, bet šios sistemų požymių vektoriaus (būdingųjų taškų išgryninimo etape) buvo prasčiausios.

5.3. Trumpas skyriaus apibendrinimas

Ištirtos sistemos požymių sutapdinimui ir sekimui. Sistema sukurta darbe naudojo SOM neuronų tinklą rasti objekto poziciją, pagal sutapdintus požymius. Rekomenduojama, kad taškams sutapdinti būtų panaudota 0,7 maksimalaus artimiausio kaimyno atstumo kriterijus (atstumas mažesnis, nei iki kaimyninio neurono). Tokiam matavimų kiekiu gaunamas maksimalus sutapdinimų kiekis turint mažą klaidų kiekį. Taip pat rekomenduojama, kad SOM tinklo santalką apmokius būtų parinktas kriterijus 0,3 maksimalaus neuronų tinklo elementų. SOM tinklo apmokymo metu naudoti parametrai – lango dydis $\Theta = 1.0$, apmokymo greitis 0,1, esant 100 apmokymo iteracijų.

Ištyrus sistemas paaiškėjo, kad geriausiai veikia kombinuota SURF požymių radimo ir BRISK požymių išskyrimo sistema. BRISK požymių išskyrimą naudojusios sistemos pasiekė geresnių rezultatų. Šio algoritmo tyrimo metu naudotos požymių aptikimo slenkstis – 15, algoritmo nuokrypis nuo centro buvo daug mažesnis nei alternatyvų. SURF požymio radimo sistemos pasižymėjo mažesniu klaidų skaičiumi, bet šios sistemų požymių vektoriaus (būdingųjų taškų išgryninimo) etape buvo prasčiausios.

APIBENDRINIMAS. IŠVADOS

Baigiamajame magistro darbe iškeltas tikslas pasiektas ištyrus objektų sekimo sistemą, pildytą SOM dirbtinio neuronų tinklo moduliu. Analitinės literatūros apžvalgos metu, ištyrus naudojamus algoritmus, sukūrus sistemą ir ištyrus jos veikimą suformuluotos darbo išvados:

1. Atlikus analitinę literatūros apžvalgą galima padaryti išvadas:

- (a) Egzistuoja nemažai algoritmų naudojamų skirtingoms paskirtims ir turinčių skirtingų veikimo metodų. Klasifikacija ir atranka būtina norint sukurti objektų sekimo sistemą.
- (b) Tyrimui pasirinkti būdingųjų taškų požymių algoritmai, kurie plačiai taikomi ir yra atsparūs skalės, bei šviesumo pokyčiams. Šie algoritmai naudoja šablono formavimo metodą, kuriame ieškomo objekto šablonas naudojamas objektams aprašyti.
- (c) Objektai gali turėti įvairius objekto atvaizdavimo ir modeliavimo metodus. Kai kurie modeliai tinka žmogui sekti, kai kurie deformuotiems objektams, tretį atitinka paprastą geometrinio primityvo modelį. Modelio objektas gali būti išreikštas tiek dvimačiu atvaizdu, tiek trimačiu, tačiau, kad trimatę išraišką panaudoti reikia turėti trimatės erdvės ar taškų modeliavimo galimybes.

2. Atlikus pasirinktų algoritmų analizę pastebėta:

- (a) Šablonams sekti algoritmai taikomi trimis etapais: požymių radimo, požymių apibūdinimo, požymių sutapdinimo.
- (b) Sėkmingi algoritmai remiasi gradientais ir intensyvumu, bei turi kryptingumą. Kryptingumo duomenų formavimas esminis skirtumas tarp nagrinėtų algoritmų. SURF algoritmas naudoja gradientų kompoziciją kryptingumui rasti. BRISK ir FREAK išrinkimo langus aprašomus dvejetainiais Gauso funkcijų langais. Tarp pastarųjų algoritmų langų išdėstymas yra visiškai skirtingas.

3. Įgyvendinus sistemą galima daromos prielaidos:

- (a) Sistemai reikia trijų dalių: požymių radimo, požymių išskyrimo ir sutapdinimo. Šios dalys gal būti sukuriamos ir iš skirtingų algoritmų: SURF ir BRISK, BRISK ir FREAK ir kitų kombinacijų.
- (b) Sutapdinimo žingsniui reikia išorinių taškų atmetimo metodų. Šiam tikslui įgyvendintas SOM dirbtinis neuronų tinklas. Šio algoritmo veikimo principas grįstas dirbtinio neuronų tinklo išdėstymu plokštumoje ir tinklo svorių pozicijos judėjimu mokantis. Esant dideliui atnaujinimo pločiui $\Theta = 1,5 - 2,2$, iteracijų skaičiui 100, mokymo greičiui 0,1 pasiekiamas tinklo kontūro atitikimas santalkos sričiai. Lokali santalkos metodu atmetami dideliu kaimyninio atstumo dirbtiniai neuronai,

kurių tarpusavio atstumas didesnis, nei 0,3 didžiausio labiausiai nutolusio neurono atstumo vertės. Lokalių santalkų formavimui aplink tašką naudojamas mažesnis lango plotis $\Theta = 1,0 - 1,2$. SOM tinklo apmokymas organizuojamas sunormavus koordinates plokštumoje $0,0 - 1,0$ intervale.

- (c) SOM tinklo klasifikavimo naudojimui pasiūlytos dvi strategijos: dirbtinį neuronų tinklą atitinkančios srities gaubtinės išskyrimas ir išorinio neuronų tinklo sluoksnio aktyvacijos srities atmetimas.

4. Atlikus duomenų atranką galima teigti:

- (a) Vaizdo sekimui scenarijai klasifikuojami pagal: kameros poziciją, objekto statiškumą ir aplinkos statiškumą. Šiems scenarijams parengtos vaizdo sekos.
- (b) Šablonai parengti atskirais kadrais ir su svarbos sričių aprašu. Toks metodas pasirinktas siekiant atskirti šablono gavimą nuo bendro vaizdo apdorojimo.

5. Atlikus eksperimentus, gauti rezultatai:

- (a) Visoms tirtoms sistemoms taikomi atstumo kriterijus rekomenduojamas naudoti yra kaimyninis taškas turi būti arčiau, nei 0,7 to taško sutapdinimo atstumo. Sutapdinant algoritmai klaidų skaičius nelabai skyrėsi, kai atstumas tarp taškų buvo artimas 0,7 kaimyninio taško atstumo.
- (b) Įgyvendintas SOM neuronų tinklo lokaliąją santalką sutapdinimas, kurio metu randama dirbtinio neuronų tinklo santalka, atmetus nutolusius neuronų tinklo taškus (kurių atstumas didesnis nei 0,3 maksimalaus nuotolio), gaunamas neuronų tinklo branduolys iš kurio galima skaičiuoti taško poziciją.
- (c) Ištyrus sistemą su SOM tinklo dirbtiniais neuronais paaiškėjo, kad geriausiais rezultatais pasižymėjo BRISK požymių vektorius ir jo kombinacija su SURF būdingųjų taškų radimo algoritmu. Šių algoritmų klaidų atstumai buvo minimalūs. Kiti algoritmų klaidos atstumas buvo 1,3-4,32 kartus didesnis. BRISK požymių vektoriaus sudarymo algoritmo sistemos turėjo mažesnę klaidą, nei alternatyvios sistemos. SURF požymio radimo sistemos pasižymėjo mažesniu klaidų skaičiumi, bet šios sistemų požymių vektoriaus (būdingųjų taškų išgryninimo) etape buvo prasčiausios.

LITERATŪRA

- Alahi, A.; Ortiz, R.; Vandergheynst, P. 2012. Freak: Fast retina keypoint, *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. Ieee, 510–517.
- Babenko, B.; Yang, M.-H.; Belongie, S. 2011. Robust object tracking with online multiple instance learning, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33(8): 1619–1632.
- Barnich, O.; Van Droogenbroeck, M. 2011. Vibe: A universal background subtraction algorithm for video sequences, *Image Processing, IEEE Transactions on* 20(6): 1709–1724.
- Bay, H.; Tuytelaars, T.; Van Gool, L. 2006. Surf: Speeded up robust features. *Computer vision–ECCV 2006*, Springer, 404–417.
- Bazzani, L.; Cristani, M.; Murino, V. 2013. Symmetry-driven accumulation of local features for human characterization and re-identification, *Computer Vision and Image Understanding* 117(2): 130–144.
- Benko, H.; Jota, R.; Wilson, A. 2012. Miragetable: freehand interaction on a projected augmented reality tabletop, *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 199–208.
- Borji, A.; Itti, L. 2013. State-of-the-art in visual attention modeling, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35(1): 185–207.
- Ding, L.; Goshtasby, A. 2001. On the canny edge detector, *Pattern Recognition* 34(3): 721–725.
- Fernando, W.; Herath, H.; Perera, P.; Ekanayake, M.; Godaliyadda, G.; Wijayakulasooriya, J. 2014. Object identification, enhancement and tracking under dynamic background conditions, *Information and Automation for Sustainability (ICIAfS), 2014 7th International Conference on*. IEEE, 1–6.
- Führ, G.; Jung, C. R. 2014. Combining patch matching and detection for robust pedestrian tracking in monocular calibrated cameras, *Pattern Recognition Letters* 39: 11–20.
- Geiger, A.; Ziegler, J.; Stiller, C. 2011. Stereoscan: Dense 3d reconstruction in real-time, *Intelligent Vehicles Symposium (IV)*.
- Heikkilä, M.; Pietikäinen, M.; Schmid, C. 2009. Description of interest regions with local binary patterns, *Pattern recognition* 42(3): 425–436.
- Irvine, J. M.; Wood, R. J.; Reed, D.; Lepanto, J. 2013. Video image quality analysis for enhancing tracker performance, *Applied Imagery Pattern Recognition Workshop: Sensing for Control and Augmentation, 2013 IEEE (AIPR)*. IEEE, 1–9.

- Jiang, J.; Yilmaz, A. 2011. Good features to track: A view geometric approach, *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 72–79.
- Jiang, Y.; Chen, K.-J.; Zhou, Z.-H. 2003. Som based image segmentation. *Rough sets, fuzzy sets, data mining, and granular computing*, Springer, 640–643.
- Kohonen, T. 2001. *Self-organizing maps*. Springer Science & Business Media.
- Leutenegger, S.; Chli, M.; Siegwart, R. Y. 2011. Brisk: Binary robust invariant scalable keypoints, *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2548–2555.
- Lowe, D. G. 1999. Object recognition from local scale-invariant features, *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, 2. Ieee, 1150–1157.
- Mitzel, D.; Leibe, B. 2012. *Taking mobile multi-object tracking to the next level: People, unknown objects, and carried items*. Springer.
- Nee, A.; Ong, S.; Chryssolouris, G.; Mourtzis, D. 2012. Augmented reality applications in design and manufacturing, *CIRP Annals-Manufacturing Technology* 61(2): 657–679.
- Ning, J.; Zhang, L.; Zhang, D.; *et al* 2012. Scale and orientation adaptive mean shift tracking, *Computer Vision, IET* 6(1): 52–61.
- Schaeffer, C. 2013. A comparison of keypoint descriptors in the context of pedestrian detection: Freak vs. surf vs. brisk.
- Smeulders, A. W.; Chu, D. M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; Shah, M. 2014. Visual tracking: An experimental survey, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36(7): 1442–1468.
- Somervuo, P.; Kohonen, T. 1999. Self-organizing maps and learning vector quantization for feature sequences, *Neural Processing Letters* 10(2): 151–159.
- Telagarapu, P.; Rao, M. N.; Suresh, G. 2012. A novel traffic-tracking system using morphological and blob analysis, *Computing, Communication and Applications (ICCCA), 2012 International Conference on*. IEEE, 1–4.
- Teng, F.; Liu, Q. 2014. Robust multiple ship tracking in inland waterway cctv system, *TEL-KOMNIKA Indonesian Journal of Electrical Engineering* 12(11): 7772–7777.
- Vincent, O.; Folorunso, O. 2009. A descriptive algorithm for sobel image edge detection, *Proceedings of Informing Science & IT Education Conference (InSITE)*, 40. 97–107.
- Wang, X.; Han, T. X.; Yan, S. 2009. An hog-lbp human detector with partial occlusion handling, *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 32–39.
- Zhang, L.; van der Maaten, L. 2013. Structure preserving object tracking, *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 1838–1845.

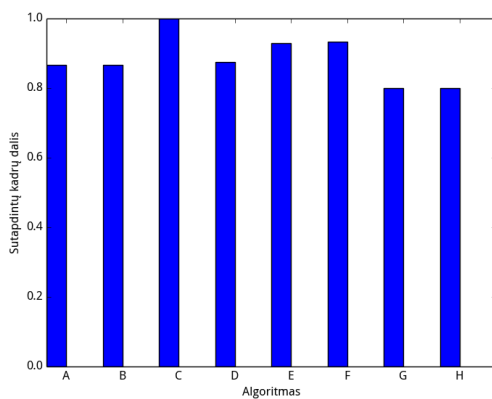
- Zhu, Q.; Yeh, M.-C.; Cheng, K.-T.; Avidan, S. 2006. Fast human detection using a cascade of histograms of oriented gradients, *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2. IEEE, 1491–1498.
- Zivkovic, Z. 2004. Improved adaptive gaussian mixture model for background subtraction, *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2. IEEE, 28–31.

PRIEDAI

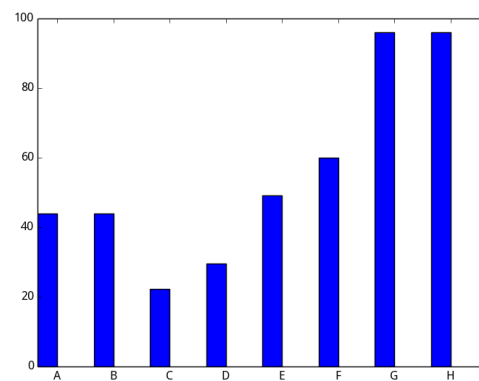
A priedas. Objektų sekimo metodų analizės rezultatai

Optimalus atstumas esant pasirinktai klaidos tolerancijai

Algoritmų rinkinys	atstumas esant 5%	atstumas esant 10%	atstumas esant 15%
SURF-SURF-L2	0,70	0,74	0,76
SURF-SURF FLANN	0,66	0,72	0,74
SURF-BRISK	0,66	0,72	0,74
SURF-FREAK	0,69	0,71	0,74
BRISK-BRISK	0,79	0,8	0,82
BRISK-SURF	0,70	0,74	0,76
BRISK-FREAK	0,72	0,76	0,79



(a)

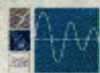


(b)

Sistema	Žymuo	Sutapdinimų santykis	Nuokrypis nuo centro
SURF-SURF FLANN	A	0,87	43,86
SURF-SURF-L2	B	0,87	43,86
SURF-BRISK	C	1,0	22,22
SURF-FREAK	D	0,875	29,50
BRISK-BRISK	E	0,93	49,21
BRISK-FREAK	F	0,93	60,05
BRISK-SURF	G	0,8	96,04
BRISK-SURF FLANN	H	0,8	96,04

Teisingo sutapdinimo santykio palyginimas pagal sėkmingai sutapdintų kadrų skaičių (a) ir vidutinį atstumą tarp kadrų (b).

B priedas. Pranešimo 18-oje Lietuvos jaunųjų mokslininkų konferencijoje medžia-
ga



VILNIAUS GEDIMINO
TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS

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

PAŽYMA

Ignas Brašiškis

Vaizdų technologijų (T111) sekcijoje perskaitė pranešimą

OBJEKTŲ SEKIMUI ERDVĖJE TINKAMŲ METODŲ TYRIMAS

Mokslo komiteto pirmininkas

prof. habil. dr. Romanas Martavičius

Sekcijos pirmininkas

dr. Dalius Matuzevičius

2015 m. kovo 20 d.
Vilnius

Ignas Brašiškis

Darbo vadovas doc. dr. Dalius Matuzevičius

Vilnius, 2015-03-20

ignas.brasiskis@vgtu.lt

Problema, tikslas ir uždaviniai

Problema

- Objekto sekimo sistemų valdymas

Darbo tikslas

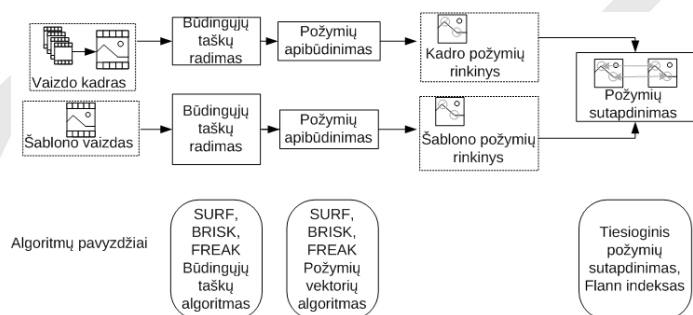
- Atlikti metodų, skirtų objektų sekimui tyrimą.

Uždaviniai

1. Atlikti objekto sekimo metodų analitinę apžvalgą
2. Atrinkti perspektyviausius metodus, skirtus objektams sekti erdvėje.
3. Ištirti atrinktus metodus objekto sekimui ir pateikti tyrimo rezultatus

ignas.brasiskis@vgtu.lt

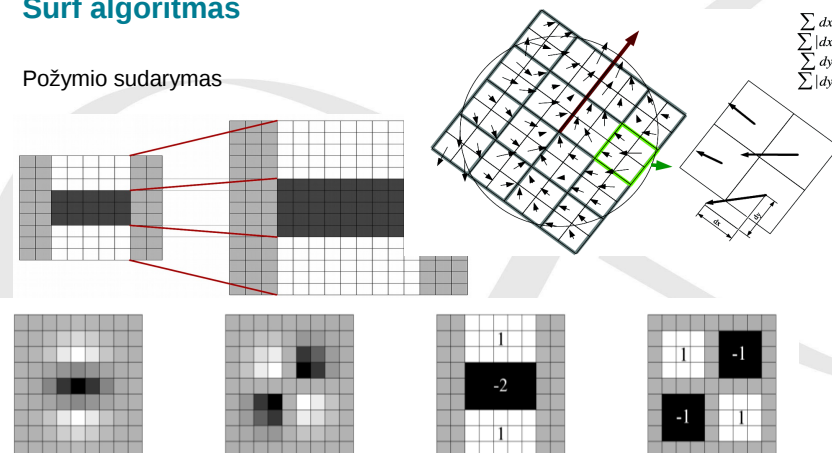
Tradicinės sistemos sudedamosios dalys



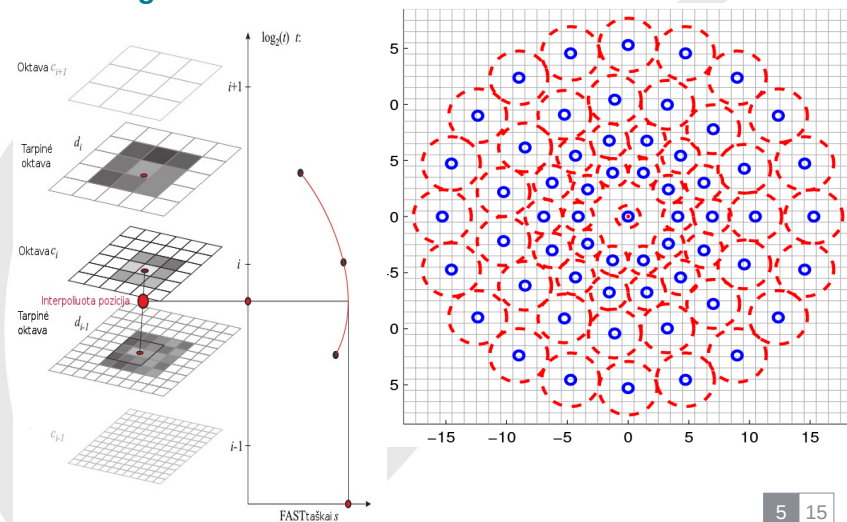
ignas.brasiskis@vgtu.lt

Surf algoritmas

Požymio sudarymas



BRISK algoritmas



5 15

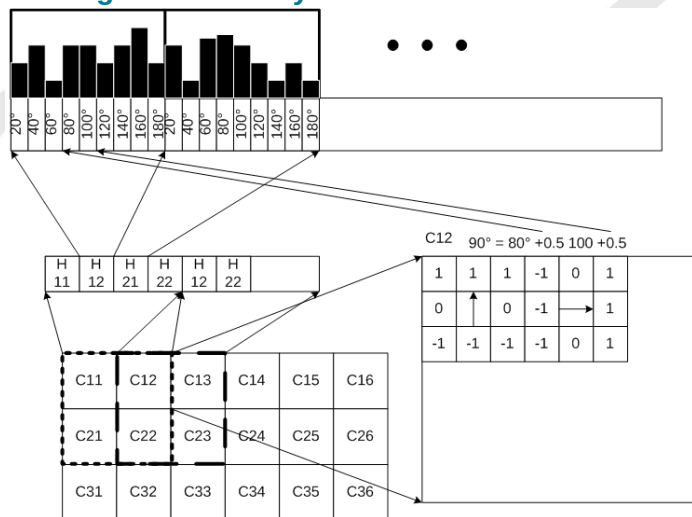
Tekstūros analizės metodai

Vaizdo analizė pagal tekstūrą grindžiama vaizdo skaisčio verčių, esančių tam tikrose apibrėžtose srityse, kodavimu. Objektui pagal jo tekstūrą aptikti dažniausiai taikomi šie metodai:

- Orientuotų gradientų histograma (angl. *histogram of oriented gradients, HOG*)

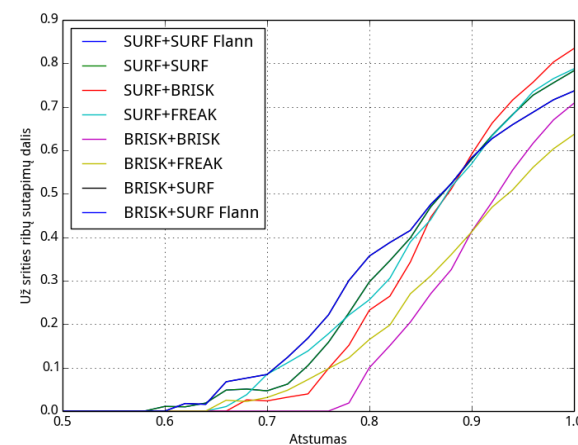
6 15

HOG histogramos sudarymas



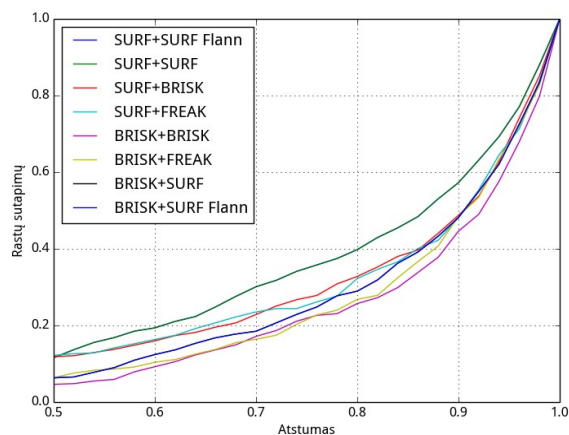
7 15

Bandymų rezultatai



8 15

Bandymų rezultatai(2)



9 15

Rezultatai

- Atlikto darbo metu buvo sukurti ir įgyvendinta sistema vaizdo sekimui.
- Sutapdinimą sistema geriausiai vykdo, kai sutapdinimo žingsnis yra artimas 0,7 kaimyno atstumo
- Atsrastų taškų skaičius auga kartu su klaidomis, klaidingų sutapdinimų skaičius mažiausias, kai atstumas tarp taškų yra iki 0,8

10 15

Išvados

- Sistemos sutapdinimas priklauso nuo atstumo atmetimo parametrų.
- Algoritmas geba automatiškai sutapdinti, jei parametrai parinkti tinkamai.
- Algoritmų veikimas panašus.

11 15

Ačiū už dėmesį!

Ignas Brašiškis
ignas.brasiskis@vgtu.lt