



**KAUNO TECHNOLOGIJOS UNIVERSITETAS
PANEVĖŽIO TECHNOLOGIJŲ IR VERSLO FAKULTETAS**

Matas Garbašauskas

**ORIENTACIJOS SISTEMOS, NAUDOJANČIOS
POLIARIZACINIUS FILTRUS, TYRIMAS**

Baigiamasis magistro projektas

Vadovas
Prof. Vytenis Sinkevičius

PANEVĖŽYS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS
PANEVĖŽIO TECHNOLOGIJŲ IR VERSLO FAKULTETAS
TECHNOLOGIJŲ KATEDRA

TVIRTINU

Katedros vedėjas doc. Arūnas Tautkus

(parašas)

(data)

ORIENTACIJOS SISTEMOS, NAUDOJANČIOS
POLIARIZACINIUS FILTRUS, TYRIMAS

Baigiamasis magistro projektas
Valdymo technologijos (kodas 621H66001)

Vadovas Prof. Vytenis Sinkevičius

(parašas)

(data)

Recenzentas

(parašas)

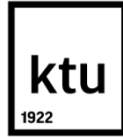
(data)

Projektą atliko Matas Garbašauskas

(parašas)

(data)

PANEVĖŽYS, 2015



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Panevėžio technologijų ir verslo fakultetas

(Fakultetas)

Matas Garbašauskas

(Studento vardas, pavardė)

Valdymo technologijos 621H66001

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Darbo pavadinimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 15 m. birželio 1 d.
_____ Panevėžys _____

Patvirtinu, kad mano **Mato Garbašausko** baigiamasis projektas tema „**Orientacijos sistemos, naudojančios poliarizacinius, filtrus tytimas**“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

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

(vardą ir pavardę įrašyti ranka)

(parašas)

BAIGIAMOJO PROJEKTO UŽDUOTIS

Išduota studentui: Matui Garbašauskui Grupė PME – 3

1. Darbo tema:
Lietuvių kalba: Orientacijos sistemos, naudojančios poliarizacinius filtrus, tyrimas

Anglų kalba: Research of orientation system using polarizing filters

Patvirtinta 2015m. vasario mėn. 13 d. dekanu potvarkiu Nr. ST17-F-13-11

2. Darbo tikslas: *Sudaryti ir iširti etaloniniu principu pagrįstą kelio dangos segmentavimo sistemą bei nustatyti poliarizacinių filtrų įtaką sistemos darbui.*

3. Reikalavimai ir sąlygos: *Darbui panaudoti „Microsoft Visual Studio“ programinį paketą ir „OpenCV“ funkcijų rinkinį.*

4. Projekto struktūra. Turinys konkretizuojamas kartu su vadovu, atsižvelgiant į BP pobūdį.

1. *Praktikoje taikomų kelio dangos atpažinimo metodų analizė.*
2. *Polarizacinių filtrų taikymo vaizdo atpažinimo sistemose analizė.*
3. *Etaloniniu pagrindu pagrįstos kelio dangos segmentavimo metodikos, kurios veikimas nesutrikę tolstant nuo etalono zonos, sudarymas.*
4. *Reikiamos programinės įrangos sudarymas.*
5. *Segmentavimo rezultatų tolstant nuo etaloniškos zonos, esant įvairioms kelio dangos tekstūros, šešėliams, prastam matomumui tyrimai.*
6. *Polarizacinių filtrų įtakos sudarytos sistemos segmentavimo rezultatams tyrimai.*

5. Ši užduotis yra neatskiriama baigiamojo projekto dalis.

6. Projekto pateikimo gynimui kvalifikacinėje komisijoje terminas	<u>2015 – 06 – 01</u>
	(data)
Užduotį gavau: <u>Matas Garbašauskas</u>	<u>2015 – 02 – 02</u>
(studento vardas, pavardė, parašas)	(data)
Vadovas: <u>Prof. Vytenis Sinkevičius</u>	<u>2015 – 02 – 02</u>
(pareigos, vardas, pavardė, parašas)	(data)

Garbašauskas, M. Orientacijos sistemos, naudojančios poliarizacinius filtrus, tyrimas. Magistro baigiamasis projektas / vadovas prof. Vytenis Sinkevičius; Kauno technologijos universitetas, Panevėžio technologijų ir verslo fakultetas, Technologijų katedra.

Panevėžys, 2015. 53 psl.

SANTRAUKA

Praktikoje kelio atpažinimas roboto orientacijai gali būti vykdomas apdorojant informaciją, gautą vaizdo kamera, kuri įtaisyta roboto priekyje. Kelio plotų atpažinimas dažnai vykdomas remiantis etalonine informacija, t. y. analizuojamas kelio dangos pavyzdys ir nustatius jo savybes atliekamas kitų kelio dangos fragmentų palyginimas. Šiuo principu atpažinimas dažnai vykdomas taikant Gauso maišos metodą, šablono sutapatinimo ir Haar savybes. Taikant šiuos metodus, nežymiai kintant aplinkai, dažnai atsiranda dideli neatitikimai.

Šiame darbe siekiama sudaryti ir ištirti objekto orientacijai skirtą etaloniniu pagrindu pagrįstą kelio dangos segmentavimo sistemą, kurios veikimas nesutrikėtų tolstant nuo etalono zonos bei poliarizacinių filtrų įtaką šios sistemos darbui. Sudarius sistemą atlikti bandymai, kurių metu patikrinta kaip keičiasi segmentavimo rezultatai:

1. keičiant etalono plotą;
2. atsiradus šešėliams ir kelio dangos defektams;
3. dėl sniego suprastėjus matomumui;
4. keičiantis kelio dangos tekstūrai;
5. sistemą papildant poliarizaciniais filtrais.

Raktiniai žodžiai: kelio dangos segmentavimas, kompiuterinė rega, poliarizaciniai filtrai, vaizdo apdorojimas.

Garbašauskas, M. Research of orientation system using polarizing filters. Masters final project / supervisor prof. Vytenis Sinkevičius; Kaunas University of Technology, Panevėžys faculty of technology and business, department of technologies.

Panevėžys, 2015. 53 psl.

SUMMARY

In practice robot navigation can be done by processing information acquired from front mounted video camera. Usually the sample information is used for road segmentation and recognition. To be more specific we can take a small video sample and extract important information that can be used to compare and recognize the road pattern. This road segmentation concept is utilized in the Gaussian mixture model, template matching and Haar features algorithms. Although these algorithms are effective in some cases even slight changes in road color may cause serious misclassifications.

In this work our goal is to research and develop a road segmentation system which work would not be effected by distance from sample zone and test if polarizing filters would improve performance of this system. In addition to developing this system we test its recognition result in various situations:

1. different sample area;
2. shadows and road texture imperfections;
3. poor visibility;
4. various road textures;
5. adding polarizing filters.

Keywords: road segmentation, computer vision, polarizing filters, image processing.

TURINYS

IVADAS	8
1. ANALITINĖ DALIS	9
1.1 Kelio dangos atpažinimo metodikos	9
1.1.1 Haar kaskadiniai klasifikatoriai	10
1.1.2 Šablono sutapatinimo metodas	12
1.1.3 Gaboro filtras	14
1.1.4 Gauso maišos modelis	17
1.1.5 Spalvų pasiskirstymo nagrinėjimas	19
1.2 Poliarizacinių savybių taikymas vaizdo atpažinimo sistemose	20
1.2.1 Poliarizacijos efekto stebėjimo priemonės.....	22
1.2.2 Aktyvus poliarizacinių filtrų taikymas	24
1.2.3 Pasyvus poliarizacinių filtrų taikymas.....	25
1.3 Skyriaus išvados	27
2. METODOLOGINĖ DALIS	28
2.1 Kelio dangos segmentavimo metodika	28
2.1.1 Objekto spalvos pasiskirstymo nagrinėjimas	28
2.1.2 Spalvos intensyvumo pasiskirstymo prognozavimas	30
2.2 Bandymams naudojama programinė įranga	31
2.2.1 Programinės įrangos struktūra	31
2.2.2 Duomenų struktūra	32
2.2.3 Duomenų išsaugojimas ir įkėlimas.....	33
2.2.4 Vaizdo šaltiniai	34
2.2.5 Mokymo procedūra.....	35
2.2.6 Vaizdo segmentavimas ir papildomos funkcijos	36
2.3 Sistemos papildymas poliarizaciniais filtrais	39
2.4 Skyriaus išvados	40

3. TIRIAMOJI DALIS.....	41
3.1 Etalninės zonos dydžio įtaka segmentavimo rezultatams	41
3.2 Segmentavimo rezultatai keičiant σ_x ir σ_y vertes.....	43
3.3 Segmentavimo rezultatai tolstant nuo pradinio etalninio fragmento zonos.....	46
3.4 Segmentavimo rezultatai esant įvairioms kelio dangos tekstūroms.....	49
3.5 Kelio dangos tekstūros neatitikimų bei šešėlių įtaka segmentavimo rezultatams ..	52
3.6 Kelio prasto matomumo įtaka segmentavimo rezultatams.....	53
3.7 Poliarizacinių filtrų taikymo įtaka segmentavimo rezultatams.....	54
3.8 Poliarizacinių filtrų taikymo įtaka segmentavimo rezultatams kamerą akinant saulei	56
3.9 Segmentavimo rezultatų praktinis taikymas.....	58
3.10 Skyriaus išvados	59
IŠVADOS	60
LITERATŪROS SĄRAŠAS.....	61
PRIEDAI	64

IVADAS

Kelio radimas mobiliųjų robotų judesiui buvo ir yra vienas svarbiausių prioritetų. Autonominių robotų navigacijai kuriamos kelio atpažinimo sistemos ilgą laiką buvo naudojamos tyrimams, mėgėjiškiems robotikos entuziastų projektams bei nišinėse pramonės srityse. Toks požiūris į kelio sistemų taikymą pradėjo keistis 2005 – 2007 metais, kai DARPA („Defense Advanced Research Projects Agency“) organizacija surengė pirmąsias autonominių automobilių varžybas bekelės ir miesto aplinkose.

Šių varžybų rezultatai įrodė autonominių automobilių perspektyvumą plačiajam vartotojų ratui. „Google“ kompanija, supratusi pritaikomumo galimybes, investavo į tokių automobilių vystymą ir pirmoji 2011 metais gavo leidimą bandyti šią technologiją Nevados valstijos gatvėse (automobilyje privalo sėdėti vairuotojas galintis perimti valdymą). Žvelgiant į autonominių automobilių ateitį matomas aiškus didžiųjų automobilių gamintojų bei vartotojų susidomėjimas šia technologija. Šį susidomėjimą galima pagrįsti „Google“, „VW“ bei „Toyota“ planais 2018 – 2020 m. pristatyti vartotojams skirtą šios technologijos versiją. Šios tendencijos rodo, kad patikimesnių, greitesnių ir mažiau resursų naudojančių kelio atpažinimo sistemų poreikis yra kaip niekada didelis.

Darbo metodai – mokslinės literatūros analizė, eksperimentai.

Darbo objektas – kelio dangos segmentavimo sistema.

Darbo tikslas: sudaryti ir iširti etaloniniu principu pagrįstą kelio dangos segmentavimo sistemą bei nustatyti poliarizacinių filtrų įtaką sistemos darbui.

Darbo uždaviniai:

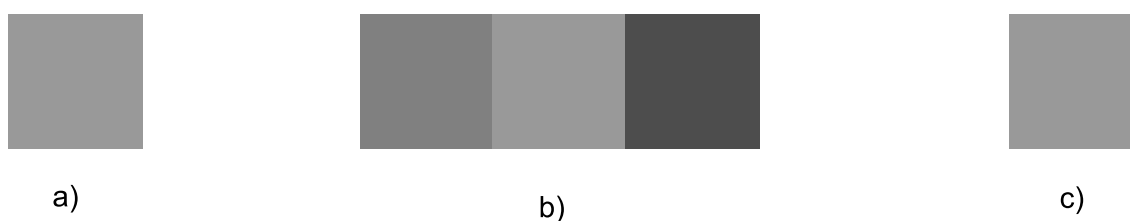
1. Išanalizuoti praktikoje taikomus kelio dangos atpažinimo metodus.
2. Išanalizuoti poliarizacinių filtrų taikymą vaizdo atpažinimo sistemose.
3. Sudaryti etaloniniu principu pagrįstą kelio dangos segmentavimo metodą, kurio veikimas nesutrikėtų tolstant nuo etaloninės zonos.
4. Sudaryti segmentavimui skirtą programinę įrangą.
5. Iširti segmentavimo metodo veikimą tolstant nuo etaloninės zonos, esant įvairioms kelio dangos tekstūroms, šešėliams, prastam matomumui.
6. Iširti poliarizacinių filtrų taikymo įtaką sudarytos sistemos segmentavimo rezultatams.

1. ANALITINĖ DALIS

1.1 Kelio dangos atpažinimo metodikos

Nagrinėjant orientacijos sistemas, kurių veikimas remiasi vaizdo kameromis gauto vaizdo analize, galima išvelgti etaloninės informacijos naudojimo tendenciją. Atpažinimui naudojama etaloninė informacija gali būti nuo paprasčiausios spalvos intensyvumo vertės iki kaskadinių schemų, gautų naudojant didžiulius informacijos kiekius. Nepriklausomai nuo etaloninės informacijos tipo atpažinimo procesas sudarytas iš tinkamo vaizdo fragmento parinkimo, jo etaloninės informacijos nustatymo bei jos lyginimo su kitų vaizdo kadru informacija.

Tokią informacijos apdorojimo principą lengva iliustruoti paprastu pavyzdžiu (1 pav.).



1 pav. Etaloninis principas: a – etalonas; b – tikrinami duomenys; c - rezultatas

Čia (1 pav.) pavaizduotas objekto išskyrimas naudojant etaloninę informaciją. Pirmiausiai pasirenkamas vaizdo fragmentas (1 pav. a) ir nustatoma jo intensyvumo vertė (šiam pavyzdyje fragmento intensyvumo vertė lygi 157). Tada žinant reikiamą vaizdo fragmento parametro vertę (šiam pavyzdyje intensyvumas) ją galima palyginti su kito vaizdo vertėmis. Lyginant nustatytą intensyvumo vertę su kito vaizdo (1 pav. b) intensyvumo vertėmis matyti, kad abi vertės sutampa tik viduriniame kvadrato (1 pav. b). Atlikus palyginimą taškai, kuriuose intensyvumo vertės nesutapo, pakeičiami baltais taškais – jų intensyvumo vertės prilyginamos 255. Pertvarkius taškų vertes išskiriamas reikiamas vaizdo plotas (1 pav. c).

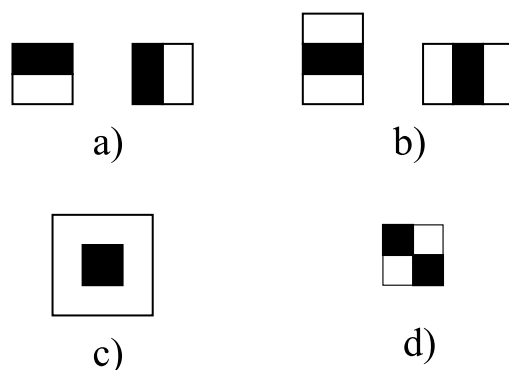
Pavyzdyje (1 pav.) apžvelgtas metodas praktikoje naudojamas labai retai dėl kelių didelių trūkumų. Realiose situacijose negalima daryti prielaidos, kad objektas, kurį bus norima atpažinti turės tik vieną intensyvumo vertę. Taip yra dėl nevienodo apšvietimo. Kintant apšvietimui keičiasi ir kameros fiksuojamo objekto intensyvumo vertė. Tokiu atveju bus gauti nukrypimai nuo etaloninės intensyvumo vertės bei neteisingai išskirtas norimas vaizdas. Taip pat atsiranda intensyvumo verčių svyravimai dėl paties objekto struktūros t. y. praktikoje objektas turi įvairių atspalvių dedamųjų, įtakojančių jo spalvą. Kartais papildomai reikia įvertinti ir objekto formą, todėl gali prireikti papildomų skaičiavimų.

Siekiant išvengti galimų neatitikimų atpažinimui taikomi sudėtingesni skaičiavimai, kuriais papildomai galima įvertinti objekto formą, intensyvumo verčių pasiskirstymą ir kitus parametrus, leidžiančius pagerinti objekto atpažinimą kintant aplinkai.

1.1.1 Haar kaskadiniai klasifikatoriai

Kaskadiniai klasifikatoriai labai dažnai naudojami šiuolaikinėse vaizdo atpažinimo sistemose. Šis metodas remiasi mašinų mokymo principais panaudojant teigiamus ir neigiamus vaizdo pavyzdžius [1]. Paprastai apmokymui naudojami tūkstančiai arba net milijonai („Microsoft Kinect“ sistema) teigiamų ir neigiamų vaizdo pavyzdžių. Teigiamais pavyzdžiais laikomi vaizdo kadrai, kuriuose yra tik norimas atpažinti objektas, o neigiamais laikomi kadrai, kuriuose gali būti vaizduojami bet kokie kiti objektai.

Norint išgauti naudingą informaciją iš teigiamų vaizdo pavyzdžių taikomos Haar savybės (2 pav. [2] [27]).



2 pav. Haar savybės: a – kraštų; b – atkarpos; c – centrinio taško; d – keturių kvadratų

Haar savybės iš esmės yra skaitinės vertės gaunamos apskaičiavus baltos ir juodos spalvos stačiakampiais (2 pav.) apribotų taškų intensyvumo sumų skirtumą [2].



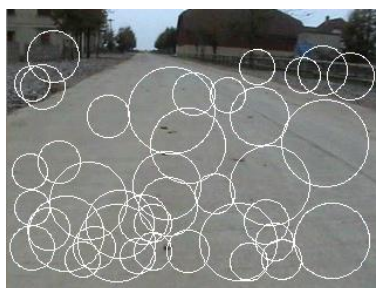
3 pav. Haar savybių panaudojimo pavyzdys: a – krašto savybė; b – atkarpos savybė [1]

Haar savybių panaudojimo atpažinimui principą lengviau suprasti pasinaudojant pavyzdžiu (3 pav.). Atpažįstant žmogaus akis (3 pav.) daroma prielaida, kad akis bus tamsesnė

nei akiduobė (3 pav. a) ir kartu abi akys bus tamsesnės nei nosis (3 pav. b). Tokiu būdu išfiltruojami regionai, kuriuose Haar savybių vertės atitinka sistemos padarytas prielaidas. Apytikslės etaloninės Haar savybių vertės parenkamos analizuojant teigiamus vaizdo pavyzdžius.

Apmokant klasifikatorių mažas naudojamų Haar savybių kiekis stipriai padidina neteisingo atpažinimo tikimybę. Norint sumažinti neteisingo atpažinimo tikimybę naudojami didžiuliai savybių kiekiai, kurie gali siekti nuo tūkstančių iki keleto šimtų tūkstančių. Dideli savybių kiekiai ženkliai sumažina sistemos spartą, todėl siekiama atrinkti savybes, turinčias mažiausią klaidos tikimybę. Savybių atrinkimui dažnai panaudojama „AdaBoost“ technologija [3]. Taikant šią technologiją įvertinama kiekvienos savybės klasifikatoriaus paklaidos tikimybe [27]. Tada visi klasifikatoriai, turintys didelę paklaidos tikimybę, pakeičiami vienu mažos paklaidos klasifikatoriumi. Šis mažos paklaidos klasifikatorius iš esmės yra visų didelės paklaidos klasifikatorių „balsavimo“ rezultatas. Balsavimo metu kiekvienas silpnas klasifikatoriaus balsas turi svorį proporcingą jo paklaidos tikimybei. Taip apjungiant silpnus klasifikatorius sumažinamas atpažinimui naudojamų savybių kiekis ir kartu padidinama sistemos sparta[3].

Nepaisant „AdaBoost“ technologija pasiekiamo spartos padidėjimo naudojamų savybių kiekis išlieka labai didelis. Todėl bandant visomis savybėmis patikrinti sistemos įėjime gautą kadrą pastebimas dramatiškas sistemos sulėtėjimas. Siekiant to išvengti taikomas klasifikatorių kaskadų principas [4]. Šiuo principu klasifikavimo procesas yra suskirstomas į atskiras stadijas. Kiekvienoje stadijoje naudojamas tam tikras klasifikatorių skaičius [27]. Paprastai pradeda nuo mažiausio klasifikatorių skaičiaus pradinėse stadijose ir baigiama didžiausiu paskutinėse stadijose. Šis principas ženkliai padidina sistemos spartą, nes galima greitai atmesti netinkamus vaizdo regionus. Jei pirmojoje stadijoje yra tik vienas klasifikatorius ir tikrinamame regione apskaičiuota vertė neatitinka to regiono, galima nebetikrinti kitų stadijų klasifikatoriais taip sutrumpinant kiekvieno regiono tikrinimo laiką [4].



4 pav. Haar kaskadinis klasifikatorius kelio atpažinimui

Haar kaskadiniai klasifikatoriai nėra pats geriausias pasirinkimas atpažįstant kelio dangą (4 pav.) (svarbu paminėti, kad rezultatai gali svyruoti, nes panaudotas sąlyginai nedidelis

pavyzdžių kiekis). Taip yra dėl keleto priežasčių susijusių su jų veikimo principu. Pirmiausiai pati kelio danga neturi vieningų išskirtinių bruožų lyginant su kitais objektais. Taip pat atsiranda nesklandumų norint apmokyti klasifikatorių atpažinti keleto tipų kelią. Norint iš naujo apmokyti klasifikatorių reikalingas didelis kiekis būtent to kelio duomenų esant įvairiam apšvietimui. Taip pat svarbus faktorius yra laikas, nes klasifikatoriaus apmokymas gali užtrukti nuo keleto valandų iki savaičių priklausomai nuo stadijų kiekio, savybių skaičiaus, naudojamų spartinimo technologijų bei kompiuterinės technikos pajėgumo.

Apibendrinus Harr kaskadinių klasifikatorių informaciją galima išskirti šio metodo privalumus ir trūkumus.

Privalumai:

- greitas atpažinimo procesas;
- universalus pritaikymas;
- didelis įvertinamų parametrų kiekis.

Trūkumai:

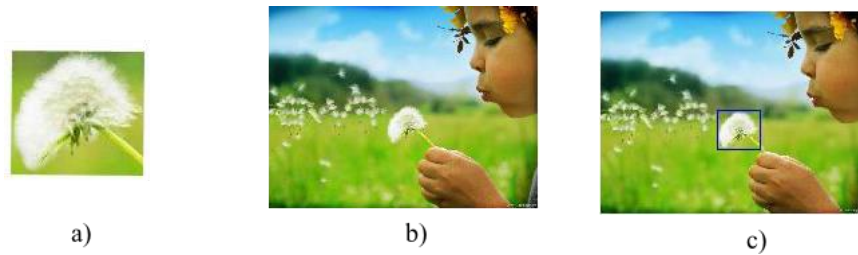
- ilgai trunkantis apmokymo procesas;
- reikalingas didžiulis kiekis etaloninių duomenų;
- ilgai trunkantis neatitikimų šalinimas (pakartotinis mokymo procesas).

1.1.2 Šablono sutapatavimo metodas

Šablono sutapatavimo metodas plačiai naudojamas objektų atpažinimui dvimatėje erdvėje [5]. Šis metodas naudoja etaloninio fragmento ir didesnio vaizdo kadro panašumą paiešką. Vaizdo kadro panašumas įvertinamas skaičiuojant koreliacijos koeficientą (1). Remiantis formule (1) etaloninis fragmentas T , kurio dydis lygus $(2h+1) \times (2w+1)$ lyginamas su vaizdu X gautu regione (u, v) .

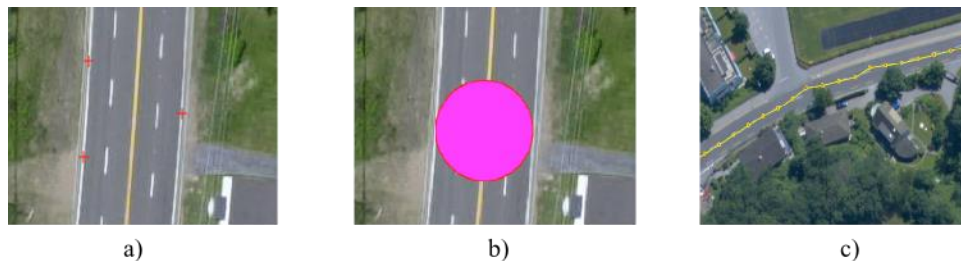
$$C(u, v) = \frac{\sum_{i=-h}^h \sum_{j=-w}^w X(i, j)T(i, j)}{\sqrt{\sum_{i=-h}^h \sum_{j=-w}^w X(i, j)^2} \sqrt{\sum_{i=-1}^h \sum_{j=-w}^w T(i, j)^2}} \quad (1)$$

Taip apskaičiavus lyginimų vaizdo fragmentų kryžminės koreliacijos koeficientą galima spręsti ar regione (u, v) yra ieškomas objektas. Tokiu būdu patikrinami visi įeinančio vaizdo regionai ir nustatomas regionas, kuriame galimai yra ieškomas objektas(5 pav.).



5 pav. Šablono sutapatinimas: a – etaloninis šablonas; b – vaizdo kadras; c – regionas, kuriame galimai yra ieškomas objektas [5]

Šis metodas taip pat gali būti pritaikytas kelio atpažinimui [6]. Skirtingai nei tradiciniame šablono sutapatinimo metode (5 pav.) čia naudojamas apskritimo formos šablonas (6 pav. b). Tokios formos šablonas taikomas dėl autoriaus [6] pasirinkto vaizdo skenavimo metodo. Tradiciškai skenuojamas vaizdo kadras sudalinamas į šablono dydžio regionus, kuriuose ieškomi objektai.



6 pav. Kelio atpažinimas taikant šablono sutapatinimą: a – pasirenkamas etaloninis regionas; b – sudarytas etaloninis šablonas; c – tikrinamų regionų masyvas [6]

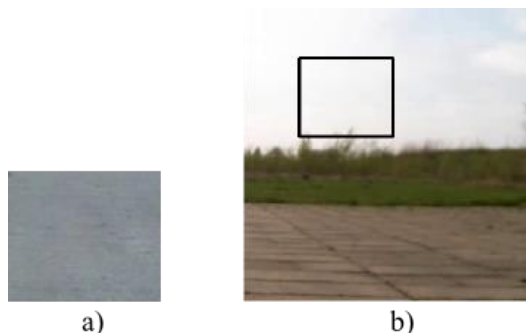
Skirtingai nei tradiciškai kitas tikrinamas vaizdo regionas pasirenkamas taikant mažiausių kvadratų metodą. Remiantis šiuo metodu prognozuojamos kito tikrinamo taško koordinatės (6 pav. c), taip sumažinant tikrinamų regionų kiekį. Taikant šablono sutapatinimą, net tolstant nuo pradinio taško (7 pav. a) pasiekiami pakankamai neblogi rezultatai.



7 pav. Šablono sutapatinimo rezultatai kintant atstumui: a – pradinis taškas; b – vidutinis nuotolis; c – didelis nuotolis; d – labai didelis nuotolis

Pagrindinis šio etalonu pagrįsto vaizdo atpažinimo metodo trūkumas yra susijęs su šablono panaudojimo principu. Vaizdą sudalinus į šablono dydžio regionus stipriai „apkarpomą“ atpažinta kelio danga (7 pav. d), t. y. jei pusę regiono užima kelio danga, o kitą pusę žolė, tada

toks regionas bus laikomas netinkamu. Taip pat pasirenkant regioną su didžiausiu koreliacijos koeficientu neužtikrinama, kad jame yra ieškomas objektas. Šį teiginį lengva paaiškinti pasinaudojant pavyzdžiu (8 pav.).



8 pav. Šablono sutapatinimo trūkumas: a – šablonas; b – tikrinamas vaizdo kadras

Pasinaudojant šablonu (8 pav. a) taikytu ankstesniame pavyzdyje (7 pav.) patikrintas kitas vaizdo kadras. Šiame vaizdo kadre (8 pav. b) kelio intensyvumo vertės gerokai skiriasi lyginant su šablono vertėmis, tad teoriškai šiame kadre koreliacijos koeficientas turėtų būti pakankamai mažas. Patikrinus šį vaizdo kadrą viename fragmente gautas pakankamai didelis koreliacijos koeficientas (8 pav. b). Nepaisant didelio koreliacijos koeficiento atpažintas vaizdo fragmentas esantis danguje (8 pav. b) ir neturi nieko bendro su tikrintu šablonu (8 pav. a). Dėl šios priežasties taikant šablono sutapatinimo metodą galima gauti nepatikimus rezultatus.

Apibendrinus šablono sutapatinimo metodo savybes galima išskirti šio metodo privalumus ir trūkumus.

Privalumai:

- greitas atpažinimo procesas;
- greitas mokymo procesas;
- universalus taikymas.

Trūkumai:

- mažas išskirtinių savybių neturinčių objektų atpažinimo patikimumas;
- atpažįstamas regionas, o ne ieškomas objektas.

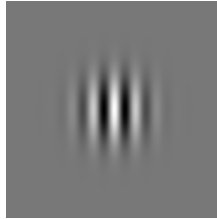
1.1.3 Gaboro filtras

Gaboro filtrą galima priskirti prie pseudo etaloningą informaciją naudojančių metodų t. y. filtro parametrai parenkami remiantis ieškomų objektų savybėmis. Šio filtro veikimas naudoja Gauso funkcijos moduliavimą taikant sudėtingą sinusoidinį signalą [7]. Šis filtras gali būti išreiškiamas koordinačių bei dažnio erdvėje. Koordinačių erdvėje Gaboro filtras išreiškiamas tokia formule [7]:

$$g(x, y, \theta, \lambda, \sigma, \gamma, \varphi) = \exp\left\{-\left(\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\right\} \exp\left\{j\left(2\pi \frac{x'}{\lambda} + \varphi\right)\right\}, \quad (2)$$

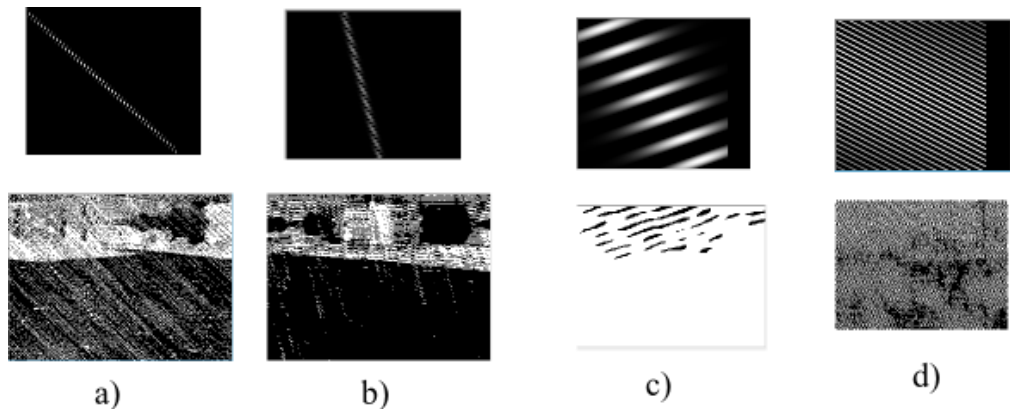
čia $x' = x \cos \theta + y \sin \theta$; $y' = -x \sin \theta + y \cos \theta$.

Panaudojus formulę (2) apskaičiuojamas dvimačio Gaboro filtro branduolys (8 pav.), kuris ir yra naudojamas vaizdo segmentavimui.



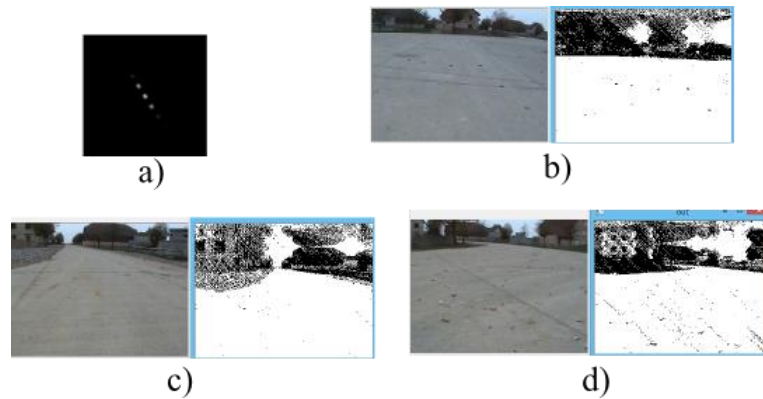
9 pav. Dvimačio Gaboro filtro pavyzdys [8]

Gaboro filtro veikimas (10 pav.) valdomas keičiant matricos dydį x_{\max} ir y_{\max} , standartinį Gauso pasiskirstymo nuokrypį σ , Gaboro funkcijos dryžių kampą θ (10 pav. b), sinusoidės bangos ilgį λ (10 pav. c), erdvinį santykį γ (10 pav. c, d) bei fazę φ .



10 pav. Gaboro filtras (viršuje) ir rezultatai (apačioje): a – pradiniai parametrai; b – keičiama θ ; c – keičiama γ ir λ ; d – keičiama γ

Keičiant šiuos parametrus Gaboro filtras gali būti pritaikomas kelio dangos segmentavimui (10 pav.). Tam, panaudojant formulę (1) ir parinkus parametrus, sudaromas Gaboro filtro branduolys (11 pav. a). Tada kiekvienam apdorojamam kadru (11 pav. b, c, d kairėje) skaičiuojama sąsuka su Gaboro filtro branduoliu (11 pav. a).



11 pav. Gaboro filtras kelio segmentavimui: a – sudarytas filtras; b, c, d – rezultatai

Atlikus šiuos skaičiavimus išėjime gaunamas Gaboro filtru apdorotas vaizdo kadras (11 pav. b, c, d dešinėje). Iš rezultatų matyti, kad taikant Gaboro filtrą gaunami pakankamai neblogi rezultatai, tačiau kintant aplinkai filtro perderinimo procesas gali užtrukti daug laiko.

Norint to išvengti dažnai taikoma sudėtinga automatinė sistema [9]. Tokios sistemos veikimas remiasi etaloninės kelio zonos sukūrimu. Etaloninei zonai nustatyti reikalingos nykstančio taško koordinatės. Šioms koordinatėms nustatyti taikomi keletu orientacijų Gaboro filtrai ir išrenkama dominuojanti orientacija θ [9]. Žinant vaizdo kadro nykstantį tašką suformuojama trapezoidinė etaloninė zona (12 pav.) ir taikant spalvinių savybių identifikavimo metodus nustatomos etaloninės savybės. Galiausiai remiantis nustatytomis savybėmis rūšiuojami įeinančių vaizdo kadrų taškai (12 pav.).



12 pav. Automatinės kelio segmentavimo sistemos veikimas keičiantis kelio formai [9]

Apibendrinus Gaboro filtro taikymą galima išskirti šio metodo privalumus ir trūkumus.

Privalumai:

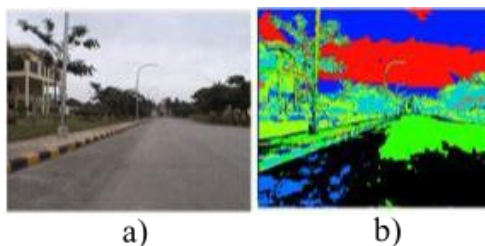
- greitas atpažinimo procesas;
- greitas derinimo procesas.

Trūkumai:

- sudėtingas filtro derinimas;
- taikant tik Gaboro filtrą sudėtinga parinkti parametrus tinkamus ilgam darbui.

1.1.4 Gauso maišos modelis

Kelio atpažinimui be anksčiau apžvelgtų metodų gali būti pritaikytas Gauso maišos modelis [10]. Taikant šį modelį bandoma nustatyti kiekvieno vaizdo taško priklausymo keliui tikimybę. Šios tikimybės apskaičiavimui reikalingi pavyzdiniai duomenys pagal kuriuos bus nustatomas skirtumas tarp kelio ir kitų vaizdo taškų [10]. Pavyzdiniais duomenimis paprastai tampa keletas vaizdo kadru, kuriuose kelio dangą turi didelę spalvos variaciją (13 pav. a). Tokie vaizdo kadrai sudalinami į segmentus, turinčius skirtingas spalvos variacijas (13 pav. b).



13 pav. Pavyzdiniai duomenys: a – vaizdo kadras; b – sudalinti segmentai [10]

Atskyrus vaizdo segmentus apskaičiuojami jiems būdingi Gauso maišos parametrai: vidurkis (3), kovariacijos matrica (4), svorio koeficientai (5) [10]:

$$\mu = \frac{\sum_{i=1}^N p(c/x_i)x_i}{\sum_{i=1}^N p(c/x_i)}, \quad (3)$$

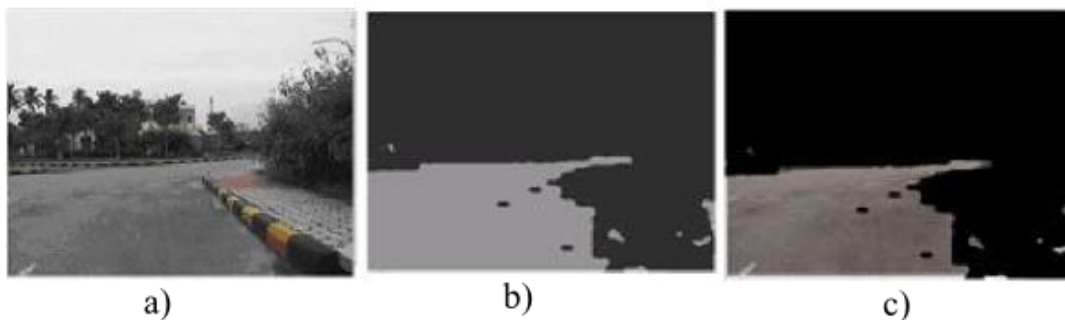
$$CO_c = \frac{\sum_{i=1}^N p(c/x_i)(x_i - \mu)(x_i - \mu)^T}{\sum_{i=1}^N p(c/x_i)}, \quad (4)$$

$$P(c) = 1/N \sum_{i=1}^N p(c/x_i). \quad (5)$$

Žinant šiuos parametrus galima apskaičiuoti taško priklausymo keliui tikimybę tokiu būdu:

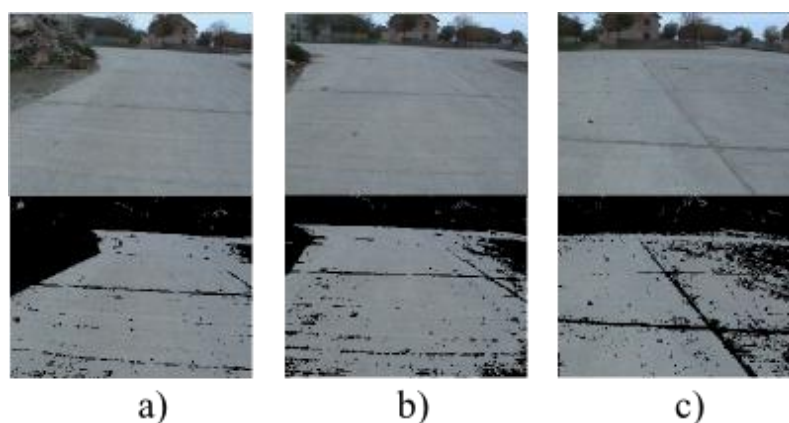
$$p(c/x_i) = \frac{p(c/x_i)p(c)}{\sum_{c=1}^K p(x_i/c)p(c)}, \quad \text{čia } p(c) = 1/K; \quad p(x/c) = \frac{\exp(-5(x - \mu)^T CO_c^{-1}(x - \mu))}{\sqrt{[CO_v]2\pi^{d/2}}} \quad (6)$$

Suradus kiekvieno taško priklausymo keliui tikimybę galima teigti, kad taškai su didžiausia tikimybės verte priklausys keliui. Remiantis šiuo principu visi įeinančio kadro (14 pav. a) taškai gali būti suskirstyti į kelio ir nedominančių objektų taškus (14 pav. b). Norint vizualizuoti tokiu būdu atpažintą kelią, suskirstyti taškai gali būti pakeičiami įeinančio vaizdo kadro taškais (14 pav. c).



14 pav. Segmentavimas taikant Gauso maišos modelį: a – įeinantis vaizdo kadras; b – suskirstyti taškai; c – perkelti pradinio vaizdo taškai [10]

Atpažinimo sistemos, naudojant Gauso maišos modelį, mokymo laikas yra sąlyginai trumpas (lyginant su Harr kaskadiniu klasifikatoriumi), tačiau realaus laiko sistemose norint automatiškai (be žmogaus įsikišimo bei sistemos darbo sutrikimų) išmokyti atpažinti naują kelią gali iškilti nesklandumų, susijusių su sistemos apmokymo laiku. Kitas svarbus faktorius yra atpažinimo kokybė. Pakartojus literatūroje [10] aprašomą metodą pastebėta, kad tolstant nuo pradinės mokymo zonos (15 pav. a) atpažinimo rezultatai prastėja (15 pav. b, c). Taip pat pastebėta, kad net mokymo zonoje atsirado taškų neatitikimų.



15 pav. Gauso maišos rezultatai tolstant nuo mokymo zonos: a – mokymo zona; b – tolstant; c – nutolus

Norint išvengti neatitikimų susijusių su atstumu nuo pradinės mokymo zonos teoriškai reikėtų vykdyti nuolatinį sistemos mokymą, tačiau šis procesas užima nemažai laiko. Kitas nemažiau svarbus faktorius yra apdorojimo laikas. Taikant Gauso maišą pagrįstą sistemą vieno kadro

apdorojimas užtrunka apie 0,11 sekundės (bandymuose bei literatūroje [10]). Šis laikas gerokai išauga papildomai atliekant pakartotinį sistemos apmokymą ir užtrunka apie 3,9 sekundės.

Apibendrinus Gauso maišos metodo taikymą galima išskirti šio metodo privalumus ir trūkumus.

Privalumai:

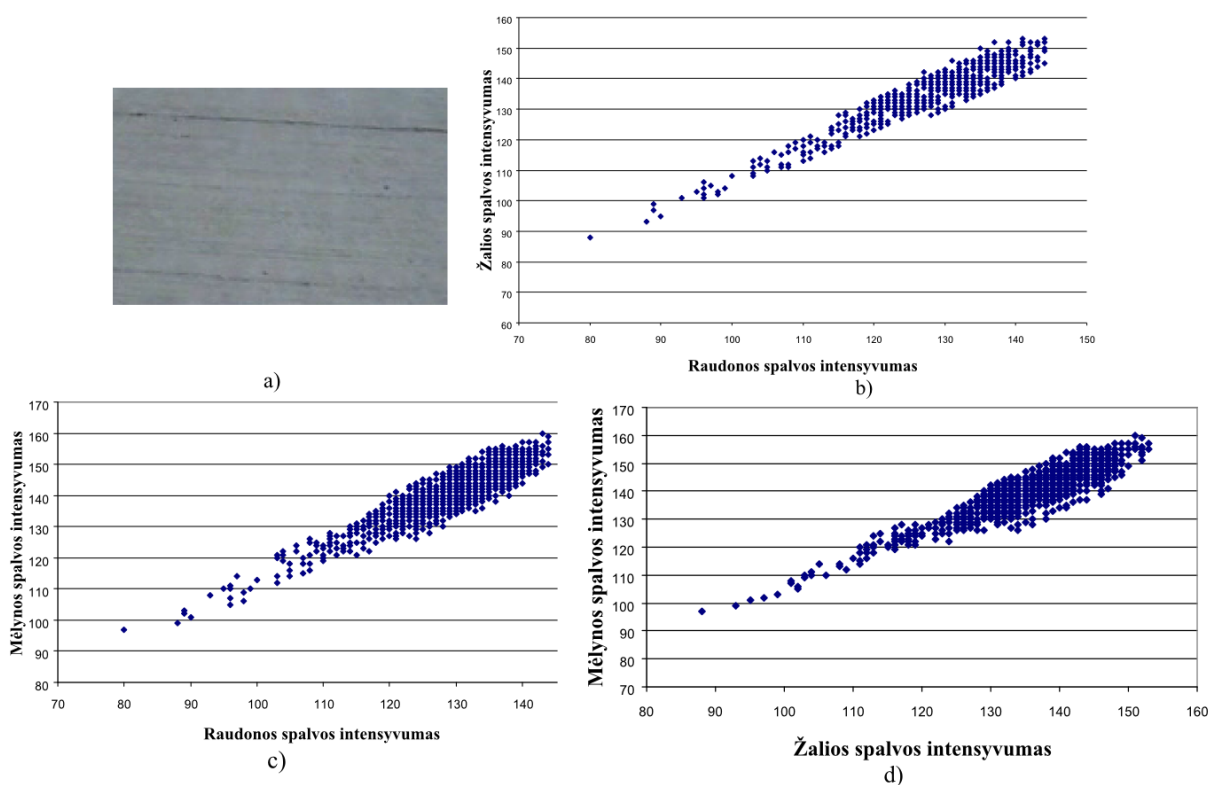
- pakankamai patikimas atpažinimas;
- reikalingas mažas etaloninių duomenų kiekis.

Trūkumai:

- sąlyginai lėtas atpažinimo ir mokymo procesas;
- nežymiai kintant aplinkai atsiranda neatitikimai.

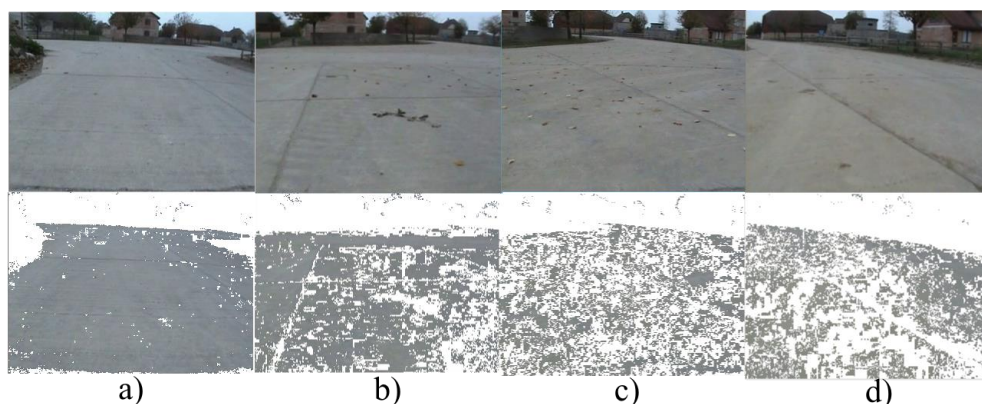
1.1.5 Spalvų pasiskirstymo nagrinėjimas

Vaizdo kadro spalvų pasiskirstymo nagrinėjamas [11] nėra pats naujausias, tačiau aktualus ir teigiamų savybių turintis vaizdo atpažinimo metodas. Spalvų pasiskirstymo nagrinėjimui reikalingas etaloninis vaizdo pavyzdys (16 pav. a).



16 pav. Spalvų pasiskirstymo nagrinėjimas: a – vaizdo pavyzdys; b – raudonos ir žalios spalvų pasiskirstymas; c – raudonos ir mėlynos spalvų pasiskirstymas; d – žalios ir mėlynos spalvų pasiskirstymas

Turint etaloninį vaizdo fragmentą sudaromos dvimatės matricos (16 pav. b, c, d), kuriose išsaugomos spalvų kombinacijos: raudona – žalia (16 pav. b), raudona – mėlyna (16 pav. c) ir žalia – mėlyna (16 pav. c). Remiantis sudarytose matricose įrašyta informacija, kiekvieno įeinančio kadro informacija sulyginama su matricų informacija.



17 pav. Atpažinimo rezultatai: a – pradinėje mokymo zonoje; b – tolstant nuo mokymo zonos; c – nutolus nuo mokymo zonos; d – už mokymo zonos ribų

Palyginus duomenis neatitikimai pakeičiami baltais taškais taip išskiriant kelio dangą. Iš rezultatų (17 pav.) nesunku pastebėti, kad tolstant nuo etaloninės mokymo zonos (17 pav. a) didėja neatitikimų kiekis ir dėl šio reiškinio prastėja kelio dangos išskyrimo kokybė (17 pav. b, c, d). Apibendrinus spalvų pasiskirstymo taikymą galima išskirti šio metodo privalumus ir trūkumus.

Privalumai:

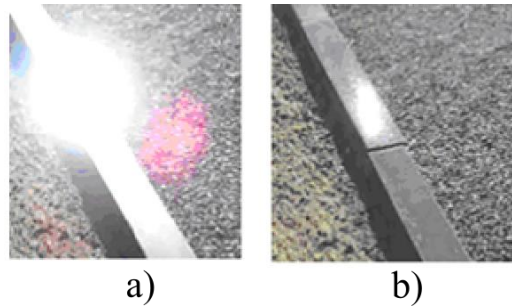
- itin greitas atpažinimo procesas;
- itin greitas mokymo procesas;
- reikalingas mažas kiekis etaloninių duomenų.

Trūkumai:

- nežymiai kintant aplinkai atsiranda dideli neatitikimai.

1.2 Poliarizacinių savybių taikymas vaizdo atpažinimo sistemose

Poliarizacinės objektų savybės vis dažniau taikomos vaizdo atpažinimo sistemose. Šios savybės suteikia galimybę neapsiriboti objektų spalva ir intensyvumu atliekant jų analizę. Gamtoje sklindanti šviesa paprastai laikoma nepoliarizuota. Kai tokia šviesa sklinda per poliarizuojančią medžiagą, atspindima nuo bet kokių paviršių ar išsklaidoma dalelių, pvz. Rėlėjaus sklaidos efektas, kai sklaidanti medžiaga – oro molekulės, ji tampa dalinai poliarizuota. Dalinai poliarizuota šviesa susideda iš dviejų komponentių: dominuojančios pilnai poliarizuotos komponentės ir nepoliarizuotos komponentės pėdsakų (18 pav.).



18 pav. Atspindys: a – dalinai poliarizuota šviesa; b – dalinai pašalinta dominuojanti komponentė

Manipuliuojant poliarizatoriaus padėtimi galima nustatyti gaunamo vaizdo poliarizacijos savybes. Naudojant tiesinius poliarizatorius jie gali būti apibūdinami perdavimo našumu. Perdavimo našumas parodo, kokia dalis įeinančios šviesos praleidžiama per tiesinį poliarizatorių ir išreiškiamas intensyvumu (7) [12].

$$I = I_0 \cos^2 \theta_i, \quad (7)$$

čia I_0 – įeinantis intensyvumas; θ_i – kampas tarp šviesos poliarizacijos krypties ir poliarizatoriaus perdavimo ašies.

Dalinai poliarizuotos šviesos veikiamų objektų savybėms charakterizuoti plačiai taikomas poliarizacijos laipsnis (DOP – „Degree of Polarization“). Šis dydis parodo santykį tarp vidutinio poliarizuoto ir bendro intensyvumo (8) [12] ir priklauso nuo lūžio koeficiento ir šviesos kritimo kampo.

$$DOP = \frac{2 \sin^2 \theta \sqrt{n^2 - \sin^2 \theta - n^2 \sin^2 \theta + 2 \sin^4 \theta}}{n^2 - \sin^2 \theta - n^2 \sin^2 \theta + 2 \sin^4 \theta}, \quad (8)$$

čia n – lūžio koeficientas; θ – šviesos kritimo kampas.

Taip pat poliarizacijos laipsnį galima išreikšti intensyvumo vertėmis (9).

$$DOP = \frac{\|I_{max} - I_{min}\|}{I_{max} + I_{min}}, \quad (9)$$

čia I_{max} ir I_{min} šviesos intensyvumo vertės jai perėjus per poliarizatorių.

Poliarizacijos laipsnio vertės yra tarp 0 ir 1, laikant kad 0 – šviesa nepoliarizuota, o 1 – šviesa pilnai tiesiškai poliarizuota. Poliarizuotai šviesai charakterizuoti taip pat gali būti panaudoti Stoko DOP ir Freznelio poliarizacinio santykio parametrai. Stoko DOP (10)

parametras lengviau taikomas lyginant su *DOP* (8), nes visus reikiamus parametrus galima gauti iš eksperimento duomenų.

$$p = \frac{\sqrt{S_1^2 + S_2^2}}{S_0}, \quad (10)$$

čia S_1 ir S_2 – parametrai, parodantys tiesinės poliarizacijos būseną; S_0 – parametras, parodantis bendrą intensyvumą.

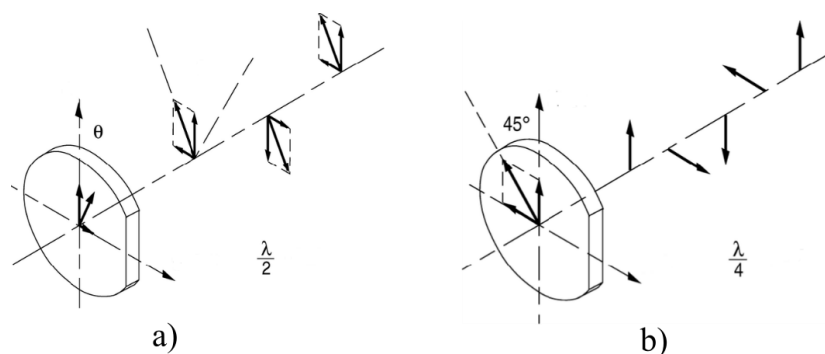
Freznelio poliarizacinis santykis (11) [12] gali būti naudingas atskiriant tiesinės poliarizacijos būsenas (lygiagrečią ir statmeną) [12].

$$pfs = \frac{F_{\perp}}{F_{\parallel}}, \quad (11)$$

čia F_{\perp} ir F_{\parallel} Freznelio lygiagretaus ir statmeno atspindžio koeficientai.

1.2.1 Poliarizacijos efekto stebėjimo priemonės

Vaizdo atpažinimo sistemose poliarizacijos efektui stebėti plačiai taikomos poliarizacinės kameros bei filtrai. Poliarizaciniai filtrai dažniausiai gaminami iš dviejų rūšių optinių bangų plokščių, galinčių keisti jas veriančių šviesos bangų poliarizacijos būsenas. Optinės bangų plokštės gali būti ketvirčio arba pusės bangos ilgio. Naudojant šias plokštes gaunami tiesiniai ir apskritiminiai (toliau CPL) poliarizaciniai filtrai.

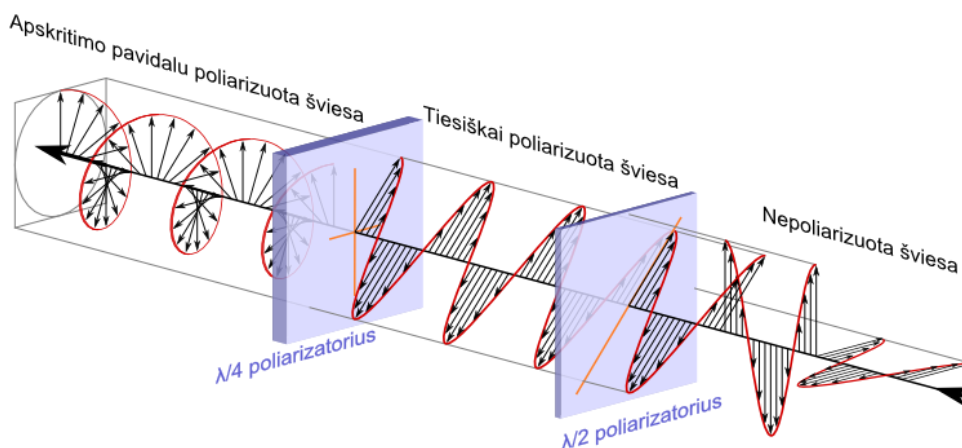


19 pav. Optinės bangų plokštės: a – pusės bangos ilgio; b – ketvirčio bangos ilgio [13]

Šie filtrai konstrukciniu atžvilgiu labai panašūs. Tiesinis poliarizacinis filtras naudoja tik pusės bangos ilgio optinę bangų plokštelę (19 pav. a). Tuo tarpu CPL filtre naudojamos abi: ketvirčio (19 pav. b) ir pusės bangos ilgio plokštelės. Tokių filtrų naudojimo skirtumas išryškėja taikant sudėtingesnę vaizdo fiksavimo įrangą. Tokia įranga dažnai turi atskiras fokusavimo ar baltos šviesos balanso sistemas į kurias šviesa patenka per papildomus tiesinius poliarizacinius

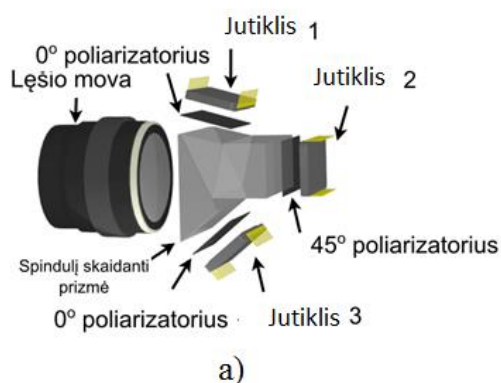
filtrus. Tokiu atveju papildomai naudojant tiesinį poliarizacinį filtrą, sistemose gali būti blokuojama įeinanti šviesa ir dėl šios priežasties sistema gali netinkamai atlikti savo funkcijas.

CPL filtre įeinanti šviesa pirmiausiai poliarizuojama taikant pusės bangos ilgio plokštelę (20 pav.) taip gaunant tiesinės poliarizacijos efektą. Šioje stadijoje taip poliarizuota šviesa gali kliudyti kai kurioms vaizdo fiksavimo sistemoms, todėl šviesa perleidžiama per ketvirčio bangos filtrą. Tokiu būdu tiesiškai poliarizuota šviesa pakeičiama į poliarizuotą apskritimo pavidalu. Taip pakeistai šviesai sklindant per dar vieną tiesinį poliarizatorių ji nebus blokuojama.



20 pav. Apskritiminio poliarizacinio filtro veikimo principas [14]

Skaičiuojant vaizdo poliarizacines savybes paprastai neužtenka vieno kadro. Šioms savybėms gali reikėti dviejų, trijų ar net keturių vaizdo kadro. Tam tikslui naudojamos specializuotos kameros, turinčios integruotas poliarizacines sistemas.

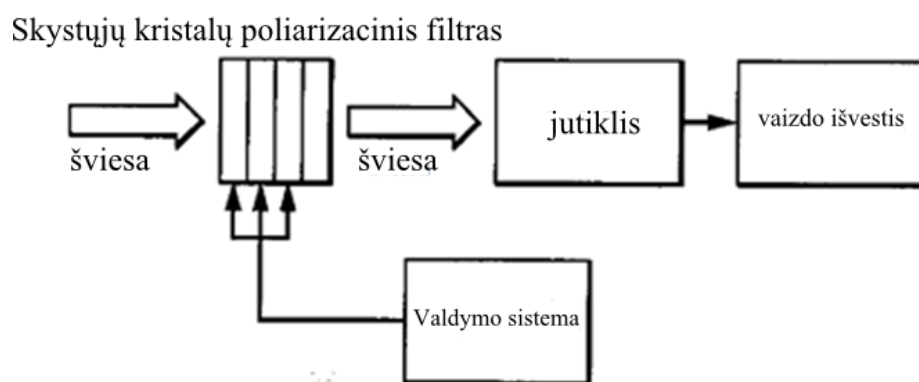


21 pav. Trijų kadro poliarizacinė kamera: a – struktūra; b – konstrukcija [15] [16]

Viena tokių sistemų taikoma trijų ir dviejų kadro poliarizacinėse kameros. Ši sistema lygiagrečiai vienu metu fiksuoja keletą vaizdo kadro. Tam naudojama speciali prizmė (21 pav.) galinti išskaidyti įeinančią šviesą trimis kryptimis [17]. Taip išskaidyta šviesa sklinda per įvairiais kampais pasuktus tiesinius poliarizacinius filtrus ir yra fiksuojama trimis jutikliais.

Tokiu būdu beveik be uždelsimo gaunama keletas vaizdo kadru, turinčių skirtingas poliarizacines charakteristikas [17].

Taip pat gali būti naudojama sistema su vienu jutikliu. Tokiu atveju naudojamas specialus skystųjų kristalų filtras (22 pav.). Šie filtrai gali keisti savo poliarizacijos ašį valdant kristalų orientaciją. Tokiu būdu galima gauti įvairias filtro orientacijas. Kameros, naudojančios skystųjų kristalų poliarizacinius filtrus, veikia skirtingai nei sistemos taikančios spindulių skaidymą. Šios kameros veikia nuosekliai t. y. norint gauti skirtingų poliarizacinių savybių kadru rinkinį reikia atlikti du nuskaitymus. Pirmojo nuskaitymo metu kristalų kryptis nekeičiama, o antruoju jie yra persukami ir gaunama kita poliarizacinė būseną. Toks kadru rinkinio gavimas šiek tiek lėtesnis nei naudojant prizmes [18].



22 pav. Sistemos su skystųjų kristalų filtru struktūra [18]

1.2.2 Aktyvus poliarizacinių filtrų taikymas

Aktyviu poliarizacinių filtrų taikymu galima vadinti atvejus, kai atpažinimas vykdomas remiantis nustatytais poliarizaciniais parametrais (pvz. DOP). Šis principas gali būti pritaikytas kelio neatitikimų nustatymui. Kelio neatitikimų klasifikavimas remiasi tuo, kad per filtrą perėjusio šviesos spindulio vertė kiti sinusoidės 180° periodu ir ta sinusoidė gali būti charakterizuojama trimis taškais. Tad naudojamos trys orientacijos (0°, 45°, 90°) ir pasinaudojant formulėmis (12 – 14) [19] apskaičiuojamas tiesinės poliarizacijos laipsnis (DOLP).

$$\theta = \frac{1}{2} \tan^{-1} \left[\frac{I_0 - 2I_{45} + I_{90}}{I_{90} - I_0} \right] + 90, \quad (12)$$

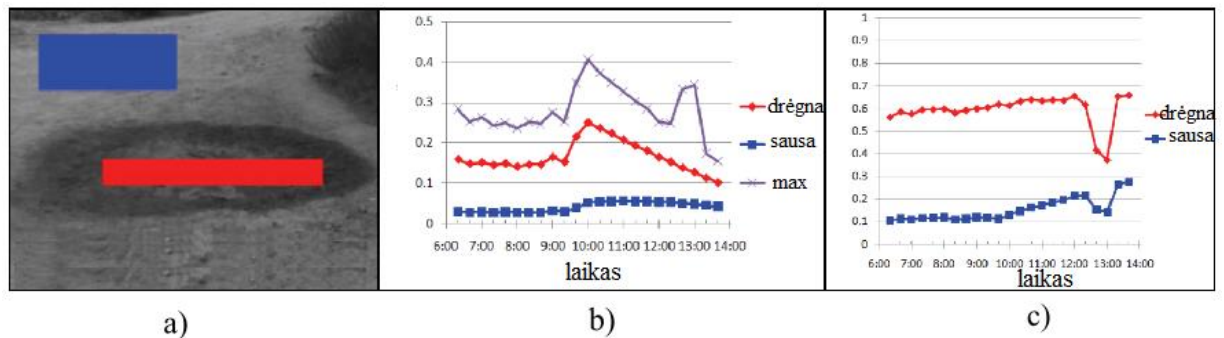
čia I_0, I_{45} ir I_{90} taškų intensyvumo vertė esant įvairioms filtro orientacijoms.

$$\theta = \begin{cases} \theta + 90, I_{90} < I_0, I_{45} < I_0 \\ \theta - 90, I_{90} > I_0 \end{cases}, \text{ čia } \theta - \text{ fazė.} \quad (13)$$

$$DOLP = \frac{I_{90} - I_0}{(I_{90} + I_0) \cos(2\theta)} \quad (14)$$

Jei viena iš filtro orientacijų sutampa su įeinančios šviesos orientacija t. y. fazė $\theta = 0$ tada poliarizacinis kontrastas bus lygus DOLP. Darant prielaidą, kad neatitikimai susiję su vandens įtaka (balos ar klampus purvas) pakanka naudoti tik dvi filtrų orientacijas: 0° ir 90° laipsnių. Tokiu atveju DOLP parametras gali būti apskaičiuojamas panaudojant poliarizacinio kontrasto formulę (9).

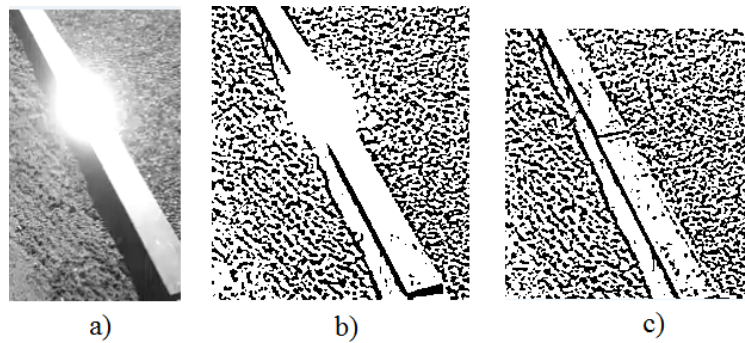
Taip apskaičiuotas DOLP vertes galima palyginti. Jei apskaičiuota DOLP vertė žymiai didesnė nei aplinkinių taškų galima spėti, kad tame taške yra struktūriniai neatitikimai. Tačiau DOLP vertę įtakoja ant objektų krintančios šviesos poliarizacija, kuri priklauso nuo oro sąlygų, saulės pozicijos (6 pav.) ir stebėjimo geometrijos. Taip pat egzistuoja sąlygos, kai struktūrinių neatitikimų DOLP nedaug arba visai nesiskiria pvz. šalia esantys augalai gali turėti beveik tokią pat DOLP vertę kaip ir aplinkiniai objektai. Iš atliktų tyrimų [19] rezultatų (23 pav.) matyti, kad DOLP vertė kinta keičiantis saulės padėčiai. Skaičiuojant poliarizacinį kontrastą struktūrinių neatitikimų DOLP vertės svyruoja gana dideliame diapazone. Tuo tarpu taikant normalizuoto poliarizacinio kontrasto skaičiavimą šios vertės svyruoja mažesniame diapazone.



23 pav. DOLP kitimas laike: a – vaizdo pavyzdys; b – poliarizacinis kontrastas; c – normalizuotas poliarizacinis kontrastas [19]

1.2.3 Pasyvus poliarizacinių filtrų taikymas

Pasyviu poliarizacinių filtrų taikymu galima vadinti šių filtrų pritaikymą vaizdo charakteristikų gerinimui t. y. atpažinimui netaikomos poliarizacinės savybės (pvz. DOP).

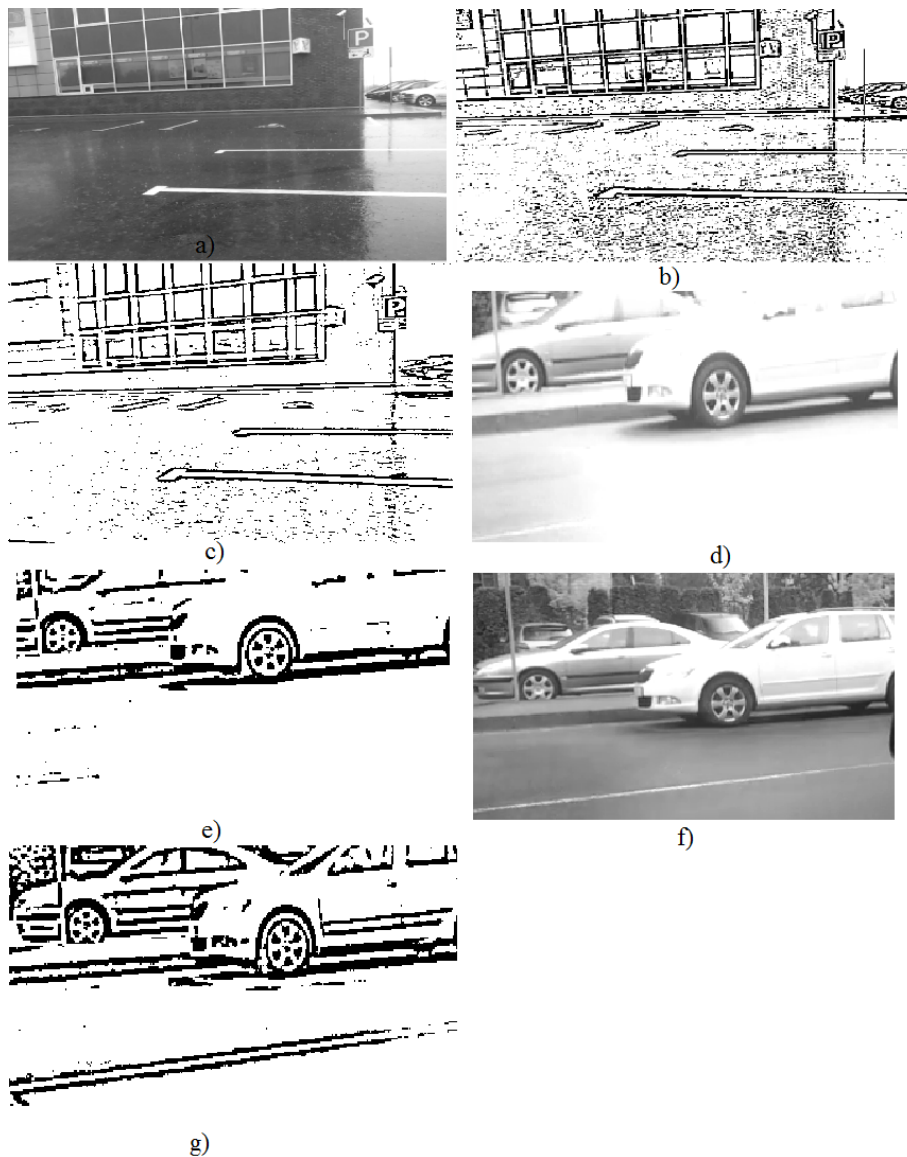


24 pav. Stipriai atspindintis paviršius: a – pradinis vaizdas; b – kraštų išskyrimas; c – kraštų išskyrimas pritaikius poliarizacinį filtrą

Pasyviai naudojami poliarizaciniai filtrai gali pagelbėti sprendžiant problemas, atsirandančias dėl vandens bei saulės poveikio [20]. Stebint atspindinčius paviršius (24 pav. a) dėl šviesos poveikio dažnai iškraipoma išskiriamo objekto forma (24 pav. b), tačiau šio efekto įtaka gali būti sumažinta sistemą papildžius poliarizaciniu filtru (24 pav. c). Nepaisant pakankamai gerų išskyrimo rezultatų (24 pav. c) svarbu pabrėžti, kad pašalinama tik dalis atspindžio, tad reikia įvertinti likusius objekto spalvos svyravimus.

Dėl lietaus poveikio (pvz. balos ant kelio dangos) gali atsirasti veidrodinis efektas (25 pav. a) sutrikdysiantis atpažinimo sistemos darbą [20]. Konkrečiame pavyzdyje (25 pav. a, b) ant kelio dangos galima pastebėti kelio ženklų stulpų atspindžius (25 pav. a). Nepašalinus šių atspindžių (25 pav. b) objektų klasifikavimo metu bus iškraipomas kelio dangos kontūras. Šią atpažinimo sistemą papildžius poliarizaciniu filtru iš objekto išskyrimo rezultatų (25 pav. c) matyti, kad vieno stulpo atspindys visiškai pašalintas, o kito sumažintas iki galimai nebeužfiksuojamos ribos. Taip pat gerokai sumažinamas triukšmingų skaičiavimo spartą įtakančių taškų regionų skaičius.

Nagrinėjant stipriai apšviestą kelio dangą (25 pav. d) bei kraštų išskyrimo rezultatus (25 pav. e) nesunku pastebėti keletą neigiamų faktorių. Pirmiausiai stipriai apšviesto kelio ploto (25 pav. d) intensyvumo verčių skirtumas yra pakankamai ženklaus, kad atpažinimui taikant spalvomis paremtą metodą būtų gautos paklaidos. Taip pat stipriai apšviestame plote galima pastebėti informacijos sumažėjimą (25 pav. d, e) t. y. kelio skiriamosios linijos spalvos intensyvumo vertės susilygina su kelio dangos vertėmis ir to pasėkoje neatpažįstama skiriamoji kelio juosta (25 pav. e). Sistemą papildžius poliarizaciniu filtru (25 pav. f, g) dalinai išsprendžiamos šios problemos. Kelio dangos spalvos intensyvumo vertės šiek tiek skiriasi (25 pav. f), tačiau šis skirtumas yra pakankamai mažas, kad atpažinimo metu būtų išvengta didelių paklaidų. Dėl intensyvumo verčių supanašėjimo išlaikomas pakankamai didelis kontrastas tarp kelio skiriamosios juostos ir kelio dangos (25 pav. f) taip neprarandant galimai svarbios informacijos (25 pav. g).



25 pav. Pasyvus poliarizacinių filtrų taikymas: a – šlapia kelio danga; b – išskirti kraštai (be filtro); c – išskirti kraštai (su filtru); d – stipriai apšviesta kelio danga; e - išskirti kraštai (be filtro); f – stipriai apšviesta kelio danga (su filtru); g – išskirti kraštai (su filtru);

1.3 Skyriaus išvados

1. Tolstant nuo etaloninės zonos prastėja kelio dangos segmentavimo rezultatai.
2. Aktyviai taikant poliarizacinius filtrus rezultatai įtakojami nepoliarizuotos natūralios šviesos dedamosios, stebėjimo geometrijos ir saulės padėties.
3. Pasyviai taikant poliarizacinius filtrus sumažinamas triukšmingų regionų kiekis.

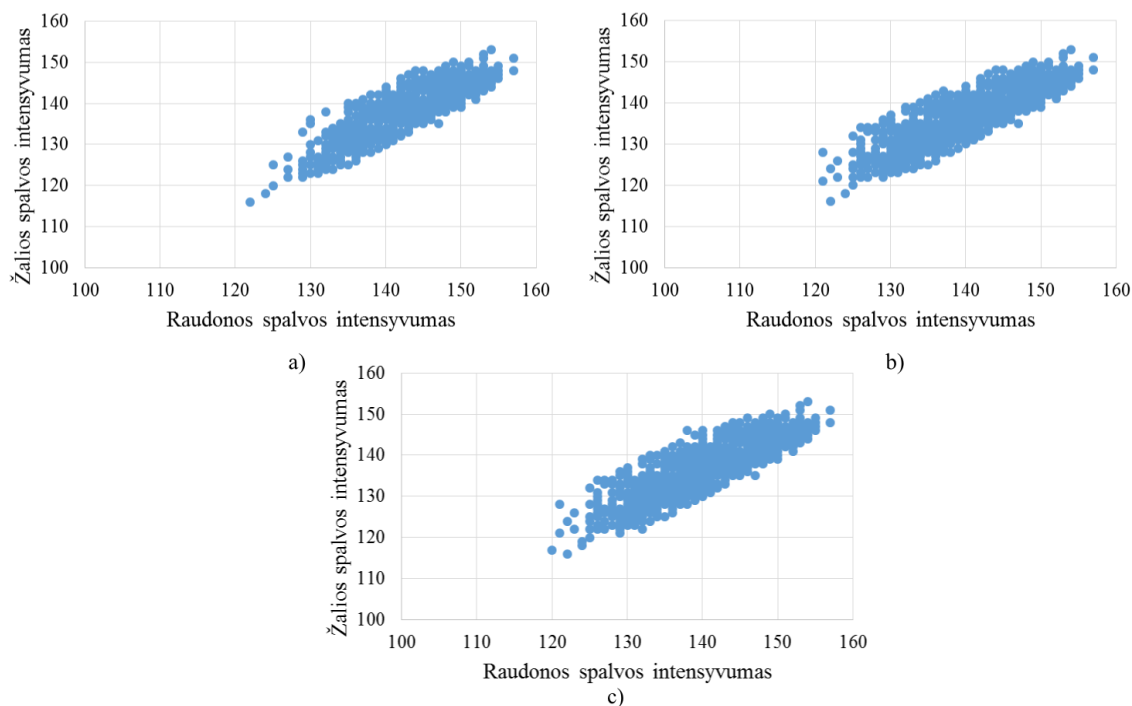
2. METODOLOGINĖ DALIS

2.1 Kelio dangos segmentavimo metodika

2.1.1 Objekto spalvos pasiskirstymo nagrinėjimas

Nagrinėjant objekto spalvos pasiskirstymą (16 pav.) susiduriama su netikslumais, atsirandančiais tolstant nuo pradinės mokymo zonos. Tad norint pašalinti šiuos netikslumus svarbu nustatyti, kaip kinta objekto spalvos pasiskirstymas tolstant nuo pradinės mokymo zonos. Siekiant išsiaiškinti, kaip kinta objekto spalvų pasiskirstymas, atliekamas palyginimas (26 pav.) keičiant atstumą tarp mokymo ir duomenų nuskaitymo zonų.

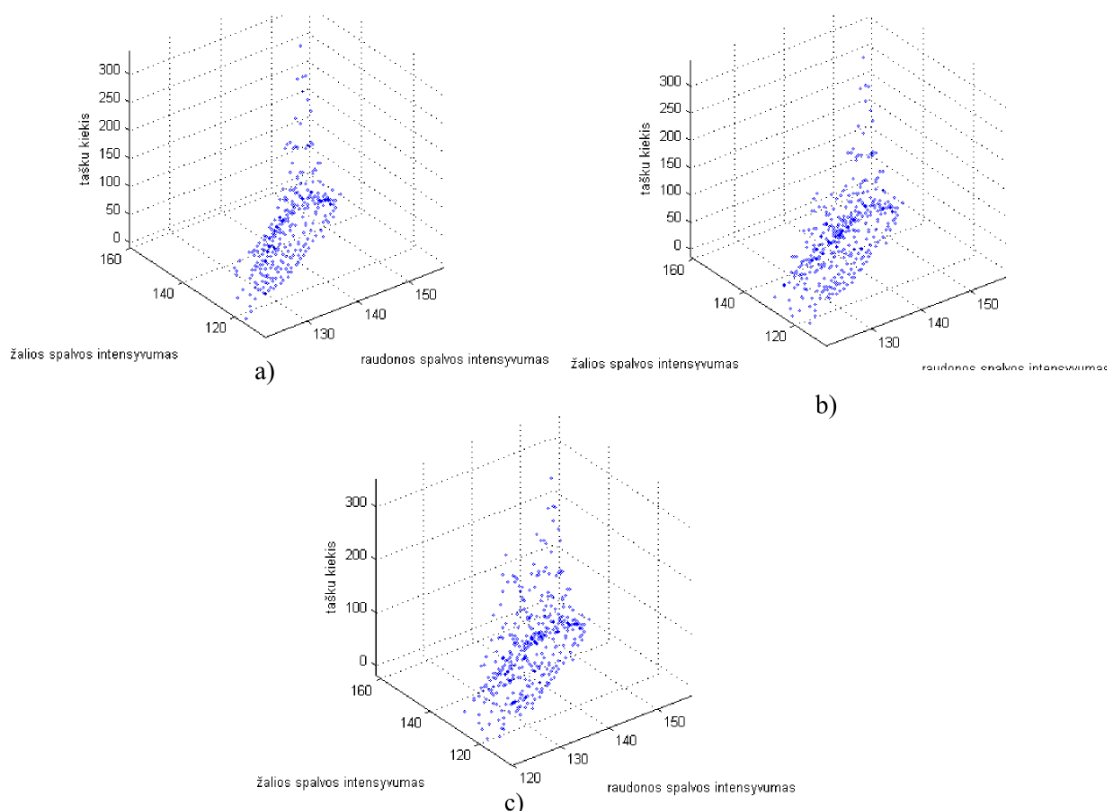
Šiam palyginimui pasirinktos trys to paties ploto duomenų nuskaitymo zonos: mokymo, tarpinė (apie 7 metrai nuo mokymo zonos), nutolusi (apie 15 metrų nuo mokymo zonos). Palyginus mokymo (26 pav. a) ir tarpinės (26 pav. b) zonų spalvų pasiskirstymo rezultatus matomas naujų spalvos intensyvumo kombinacijų atsiradimas, tačiau šios kombinacijos yra artimos pradinėms (26 pav. a) ir jų kiekis yra pakankamai mažas. Tuo tarpu palyginus tarpinės ir nutolusios zonų rezultatus spalvų pasiskirstymo neatitikimai beveik nepastebimi. Apibendrinus spalvų pasiskirstymo palyginimo rezultatus sunku paaiškinti žymų segmentavimo rezultatų kitimą (17 pav.), tad siekiant išsiaiškinti tokių neatitikimų priežastį reikia pakartoti pasiskirstymų palyginimą įvertinant papildomus veiksnius, galinčius įtakoti šių neatitikimų atsiradimą.



26 pav. Objekto spalvos intensyvumo pasiskirstymas: a – pradinė mokymo zona; b – tolstant nuo pradinės mokymo zonos; c – nutolus nuo pradinės mokymo zonos

Siekiant išsiaiškinti šių neatitikimų priežastį papildomai nagrinėjamas kiekis taškų, kuriuose spalvos intensyvumo verčių kombinacijos yra vienodos (27 pav.). Nagrinėjant spalvos intensyvumo kombinacijų pasiskirstymą (27 pav.) matomas žymus taškų kiekio (z ašyje) kitimas. Palyginus mokymo (27 pav. a) ir tarpinės zonų rezultatus iš dalies galima paaiškinti neatitikimų (17 pav.) priežastis. Stebint spalvos intensyvumo kombinacijų pasiskirstymą (26 pav. b) buvo sunku susieti mažus grafikų (26 pav. a ir b) ir didelius fragmentavimo neatitikimus (17 pav. c). Įvertinus neatitikusių taškų kiekį nesunku pastebėti, kad tolstant nuo pradinės mokymo zonos užfiksuojami dideli kiekiai taškų, kurių intensyvumo kombinacijos neatitinka mokymo duomenų, tačiau yra jiems artimos.

Panašus reiškinys pastebimas palyginus tarpinės (27 pav. b) ir nutolusios (27 pav. c) zonų duomenis. Palygus šių zonų duomenis pastebimas dar mažesnis naujų intensyvumo kombinacijų atsiradimas, tačiau matomas gretutinių kombinacijų taškų kiekio didėjimas. Iš šių rezultatų galima daryti prielaidą, kad tolstant nuo pradinės mokymo zonos atsiranda naujų spalvos intensyvumo kombinacijų stipriai įtakojančių objekto segmentavimo kokybę. Todėl siekiant išlaikyti pakankamą kelio dangos segmentavimo kokybę reikia prognozuoti naujų objekto spalvos intensyvumo kombinacijų atsiradimą.



27 pav. Spalvos pasiskirstymo kitimas įvertinant taškų kiekį: a – pradinė mokymo zona; b – tolstant nuo pradinės mokymo zonos; c – nutolus nuo pradinės mokymo zonos

2.1.2 Spalvos intensyvumo pasiskirstymo prognozavimas

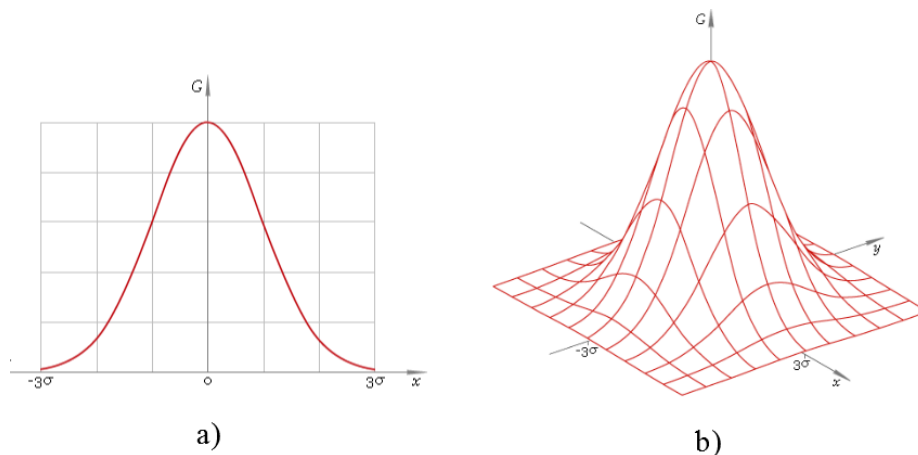
Išnagrinėjus duomenis gautus analizuojant kelio dangos spalvos intensyvumo pasiskirstymą, pastebėtos aiškios maksimalios taškų kiekio vertės bei nuokrypiai nuo mokymo metu gautų spalvos intensyvumo verčių kombinacijų. Tokiu atveju taikant tiesioginio informacijos nuskaitymo metu gautą informaciją, atpažinimo patikimumas neatitiks minimalių reikalavimų t. y. palankiausi atveju bus atpažinta tik dalis kelio. Norint išlaikyti pakankamai gerą segmentavimo kokybę tiesioginio informacijos nuskaitymo metu gautą informaciją reikia papildyti kombinacijomis gautomis prognozuojant, kaip keisis spalvos intensyvumo kombinacijų pasiskirstymas.

Spalvos intensyvumo kombinacijų pasiskirstymo kitimo įvertinimui gali būti panaudotas Gauso skirstinys (15).

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad (15)$$

čia: σ – standartinis vertės nuokrypis; μ – vertės vidurkis; x – vertė.

Prognozuojant kelio dangos spalvos intensyvumo kombinacijų pasiskirstymą trimatėje erdvėje (įvertinant taškų kiekį) taikomas ne vienmatis (28 pav. a), o dvimatis (28 pav. b) Gauso skirstinys.



28 pav. Gauso skirstinys: a – vienmatis; b – dvimatis [21]

Taikant dvimatį Gauso skirstinį gauti rezultatai priklauso nuo amplitudės, vidurkio bei standartinio nuokrypio (16).

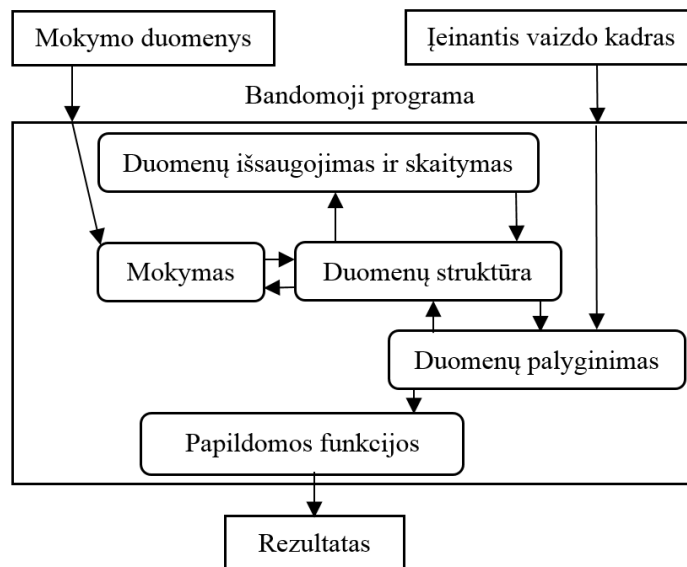
$$p(x) = A \exp\left(\frac{-(x - \mu_x)^2}{2\sigma_x^2} + \frac{-(y - \mu_y)^2}{2\sigma_y^2}\right), \quad (16)$$

čia: x, y – kombinacijos taško koordinatės; σ – standartinis intensyvumo vertės nuokrypis; μ – intensyvumo verčių vidurkis; A – intensyvumo amplitudė.

2.2 Bandyams naudojama programinė įranga

2.2.1 Programinės įrangos struktūra

Metodo tyrimui reikalinga programinė įranga, galinti nuskaityti įvairių vaizdo šaltinių informaciją, ją apdoroti bei patogiai pateikti rezultatus. Šių uždavinių įgyvendinimui panaudojamas „Microsoft Visual Studio“ programiniame pakete esantis programinio kodo redaktorius bei kompiliatorius ir „Opencv“ bibliotekos pagrindinių funkcijų rinkinys. Siekiant suprastinti programinės įrangos modifikavimą pasirenkamas OOP („Object Oriented Programming“) programavimo stilius. Taikant OOP programavimo stilių programos kodas suskirstomas į funkcinius objektus, atliekančius tam tikras funkcijas (29 pav.).



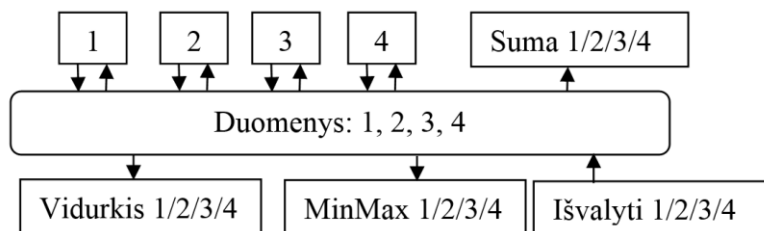
29 pav. Programinės įrangos struktūra

Neapdorotų duomenų nuskaitymo funkcinis objektas atlieka įvairių vaizdo laikmenų nuskaitymo bei pradinio duomenų apdorojimo funkciją. Turint pradinis duomenis mokymo objekte atliekamas pradinių duomenų interpretavimas, o atlikus mokymo procesą rezultatai išsaugomi pasinaudojant duomenų išsaugojimo bei skaitymo objektu. Galiausiai vadovaujantis mokymo duomenimis, duomenų palyginimo funkciniam bloke atliekamas nuskaityto vaizdo kadro segmentavimas ir atlikus papildomus skaičiavimus rezultatai pateikiami vartotojui.

2.2.2 Duomenų struktūra

Vienas pagrindinių sudarytos programinės įrangos elementų yra duomenų struktūra. Ši duomenų struktūra naudojama tarpiniam duomenų saugojimui programos darbo metu. Kadangi taikomam vaizdo segmentavimo metodui netinka nei viena standartinė „OpenCV“ duomenų struktūra sudaroma nestandartinė, reikiamas funkcijas atliekanti duomenų struktūra. Šioje duomenų struktūroje spalvų kombinacijos turėtų būti išsaugomos atskirose matricose bei jų vertės efektyviai nuskaitomos nepriklausomai nuo saugomų duomenų kiekio. Taipogi duomenys turėtų būti efektyviai išsaugomi, ištrinami bei lengvai atliekami pagrindiniai matematiniai skaičiavimai (30 pav.).

Šių principų įgyvenimui sudarytas duomenų struktūros funkcinis objektas. Šiame objekte kiekvienai spalvos kombinacijai sukuriama atskiri dvimačiai sveikųjų skaičių masyvai.



30 pav. Duomenų struktūra

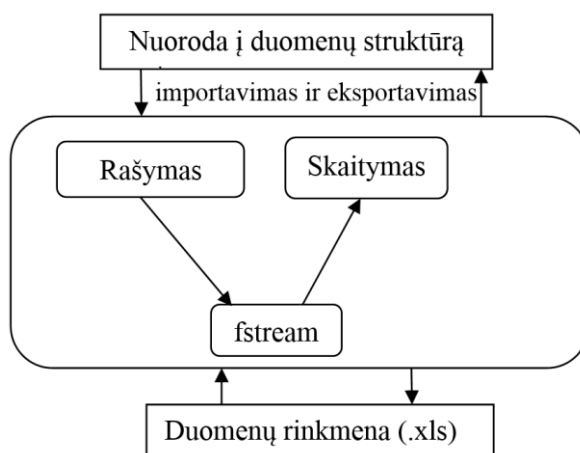
Taip pat kiekvienai spalvų kombinacijai sukuriama atskiros duomenų nuskaitymo bei įrašymo funkcijos. Taikant išskaidytą duomenų saugojimo metodiką padidinama duomenų apdorojimo sparta bei pasiekiamas lankstesnis apdorojimo procesas. Išskaidžius duomenų apdorojimą galima lengvai perrašyti ar ištrinti tam tikros spalvų kombinacijos vertes neįtakojant kitų kombinacijų verčių bei nepriklausomai nuo kitų kombinacijų intensyvumo verčių kiekio.

Be duomenų išsaugojimo bei nuskaitymo funkcijų reikalingos papildomos pagalbinės funkcijos, pateikiančios naudingą statistinę intensyvumo verčių kombinacijų pasiskirstymo informaciją ir efektyviai išvalančios nereikalingus duomenis. Šioje situacijoje nagrinėjamo kelio dangos segmentavimo metodo veikimui reikalingos funkcijos, pateikiančios statistinę nagrinėjamo pasiskirstymo vidurkio ir minimalios bei maksimalios verčių informaciją. Kadangi šios funkcijos gali būti iškviestos pagrindiniame apdorojimo cikle, reikalingos papildomos priemonės siekiant sutrumpinti jų vykdymo trukmę. Siekiant sumažinti vidurkio bei „Min Max“ funkcijų vykdymo laiką taikomas nuorodų principas. Taikant šį principą funkcijos vidaus bei gražinami duomenys perduodami nuorodų pavidalu taip sutrumpinant skaičiavimo laiką, nes sumažinamas bereikalingas duomenų kopijavimas ir dalinai išvengiami galimi atminties nuotėkiai. Duomenų išvalymo trukmei sumažinti taikoma greičiausia duomenų kopijavimo

metodika, ignoruojanti galimai nesančias („null“) bei persidengiančias vertes taip sutaupant skaičiavimo laiką.

2.2.3 Duomenų išsaugojimas ir įkėlimas

Programos darbo metu susidariusių duomenų analizei kitose terpėse (pavyzdžiui „Microsoft Excel“ arba „Matlab“) reikalingos duomenų eksportavimo funkcijos, išsaugančios reikiamus apdorojimo duomenis įvairiais formatais. Taip pat patogu išsaugoti tam tikros kelio dangos duomenis ir esant reikalui lengvai juos nuskaityti į programos duomenų struktūrą. Šiems tikslams įgyvendinti sudaromas duomenų eksportavimo bei importavimo objektas (31 pav.).

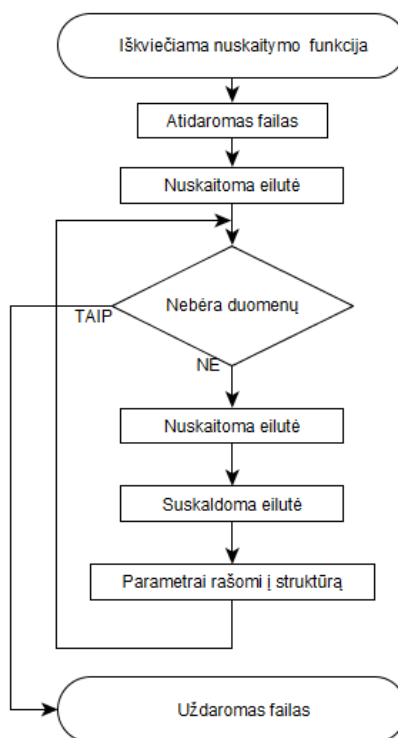


31 pav. Duomenų importavimo ir eksportavimo objekto struktūra

Duomenų eksportavimo metu siekiama sudaryti duomenų rinkmeną (šiuo atveju .xls), kurioje būtų išsaugoma pasirinktos spalvų kombinacijos pasiskirstymo vertės. Duomenų išsaugojimas laikmenoje atliekamas pasinaudojant išvesties duomenų srauto „ofstream“ funkcija. Taikant šią funkciją sukuriama „Microsoft Excel“ formato išvesties failas, kuriame kiekvienoje eilutėje išsaugoma duomenų seka: pirmos spalvos intensyvumo vertė – antros spalvos intensyvumo vertė – taškų kiekis. Tokiu būdu duomenų eksportavimo funkcijai nurodžius nuorodą į sudarytą duomenų struktūrą, visi jos duomenys perkeliama į „Microsoft Excel“ rinkmeną, kuri išsaugoma programos šakniniame („root“) kataloge. Šio proceso metu gauta duomenų laikmena panaudojama duomenų pasiskirstymo vizualizavimui dvimatėje arba trimatėje erdvėje.

Duomenų importavimu siekiama pakartotinai panaudoti anksčiau išsaugotą kelio dangos spalvos intensyvumo verčių kombinacijų pasiskirstymo informaciją. Šiam tikslui sudaroma duomenų importavimo funkcija paremta įvesties duomenų srauto („ifstream“) funkcijomis. Taikant šias funkcijas atidaromas failas su anksčiau išsaugota informacija ir kiekvienai eilutei atliekamas duomenų apdorojimas, kurios metu išskiriamos duomenų vertės. Eilutės nuskaitymo

metu gražinama duomenų seka, kuri papildomo apdorojimo metu suskaidoma į tris smulkesnes skaičių sekas. Šios sekos konvertuojamos į sveikuosius skaičius, kurie surašomi į atitinkamas pateiktos duomenų struktūros nuorodos skiltis (32 pav.).



32 pav. Duomenų importavimo procedūra

Taip apdorojus importuojamą rinkmeną gaunama pilnai apdorojimui ar atpažinimui (jei importuojami mokymo duomenys) paruošta duomenų struktūra.

2.2.4 Vaizdo šaltiniai

Analizuojant segmentavimo metodo veikimą pravartu naudoti kuo įvairesnius vaizdo pavyzdžius. Šiam tikslui sudaromas vaizdo nuskaitymo objektas, atliekantis pirminio vaizdo nuskaitymo procedūrą. Vaizdas gali būti gaunamas iš trijų vaizdo šaltinių: pavienių paveikslėlių, vaizdo failų, bei nuskaitant „Google Street View“. Pirmais dviem atvejais vaizdo pavyzdžiai gali būti gauti renkant savo vaizdo pavyzdžių kolekciją arba ieškant vaizdo pavyzdžių internete. Pavyzdžių paieškai pasitelkus internetą galima sudaryti įvairių kelio dangų rinkinį, tačiau paveikslėlių atveju negalima įvertinti rezultatų tolstant nuo pradinės zonos, vaizdo failų atveju dėl kintančios vaizdo kadro padėties erdvėje ar įrašo trukmės apribojamas tinkamo kelio ilgis. Sudarinėjant savo vaizdo pavyzdžių rinkinį galima pasirinkti patogią vaizdo kadro padėtį erdvėje ir vaizdo įrašo trukmę, tačiau iš dalies apribojama vaizdo pavyzdžių įvairovė.

Pasinaudojant „Google Street View“ galimybėmis galima gauti įvairių neapriboto ilgio kelio dangos pavyzdžių. Norint panaudoti „Google Street View“ informaciją reikia ją

konvertuoti į „OpenCv“ matricų („Mat“) formatą ir galiausiai ši matrica („Mat“) gali būti konvertuojama į programos naudojamą duomenų formatą. „Google Street View“ duomenys nuskaitomi tiesiai iš naršyklės (šiuo atveju naudojama „Google Chrome“ naršyklė). Prieiga prie naršyklės pasiekama pasinaudojant („Windows“) programavimo sąsaja [24] t. y. prieiga pasiekama išnaudojant abstrakčią nuorodą į sistemos išteklius (šiuo atveju „Google Chrome“ programos lango valdymą). Turint nuorodą į naršyklės išteklių galima gauti įtaiso kontekstą bei sukurti suderinamą įtaiso kontekstą.

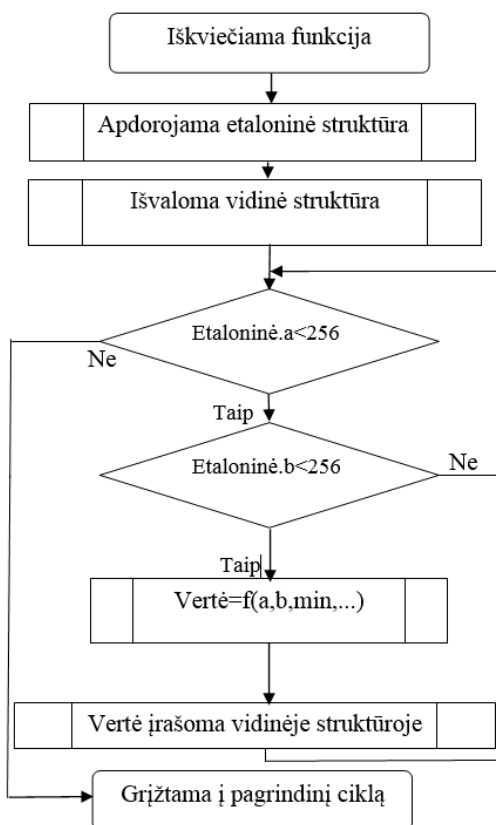
Pasinaudojus nuoroda į naršyklės išteklių gaunami lange esančio vaizdo dydis ir remiantis gautais duomenimis inicializuojamas „OpenCV“ duomenų formato objektas, o pasitelkus įtaiso kontekstą sukuriamas suderinamas taškinės grafikos („bitmap“) objektas nurodant jo dydį, bitų kiekį bei duomenų suspausdinimo tipą (šiuo atveju RGB). Galiausiai remiantis taškinės grafikos objekto parametrais bitų vertės nukopijuojamos iš suderinamo įtaiso konteksto į „OpenCV“ matricą. Taip kopijuojant naršyklės lango duomenis atsiranda atminties nutekėjimas susijęs su nuorodomis į sistemos išteklius. Siekiant išvengti atminties nutekėjimo baigus nuskaitymo procesą sunaikinami visi nereikalingi objektai. Nuskaitymo metu gauta matrica apdorojama lygiai taip pat, kaip ir tiesioginio paveikslėlių ar vaizdo įrašų kadru nuskaitymo metu gauta matrica.

2.2.5 Mokymo procedūra

Mokymo etapo metu siekiama sudaryti naują vidinę duomenų struktūrą, kurioje bus išsaugomos naujai perskaičiuotos kelio dangos spalvos intensyvumo kombinacijų pasiskirstymo vertės. Šiai procedūrai sukuriamas duomenų apdorojamo objektas. Iškviečiant šio objekto duomenų apdorojimo funkciją pateikiama nuoroda į nuskaitymo metu gautos etaloninės informacijos duomenų struktūrą. Turint etaloninę informaciją ir pritaikius pagrindinės duomenų struktūros funkcijas atliekamas pradinis duomenų apdorojimas. Šio apdorojimo metu apskaičiuojamos mažiausiai bei daugiausiai taškų turinčios spalvos intensyvumo verčių kombinacijos, vidutinis taškų kiekis bei kiti statistiniai parametrai.

Apdorojus etaloninę duomenų struktūrą išvaloma objekto vidinė duomenų struktūra ir pradedamas naujų verčių skaičiavimas. Naujų verčių skaičiavimo metu skenuojama etaloninė duomenų struktūra ir taikant formulę (16) perskaičiuojama kiekviena jos vertė, o skaičiavimų rezultatai įrašomi vidinėje objekto duomenų struktūroje (33 pav.). Taikant vidinę duomenų struktūrą programoje vienu metu galima turėti keletą etaloninės informacijos apdorojimo objektų. Vidinės duomenų struktūros panaudojimui išorėje sudaroma objekto funkcija, gražinanti šią duomenų struktūrą kitoms išorinėms funkcijoms ar objektams. Taikant šią informacijos

gražinimo būdą išvengiama nenaudingo naujų kintamųjų kūrimo, nes gražinamas objektas tiesiogiai panaudojamas tolesniame apdorojimo procese.



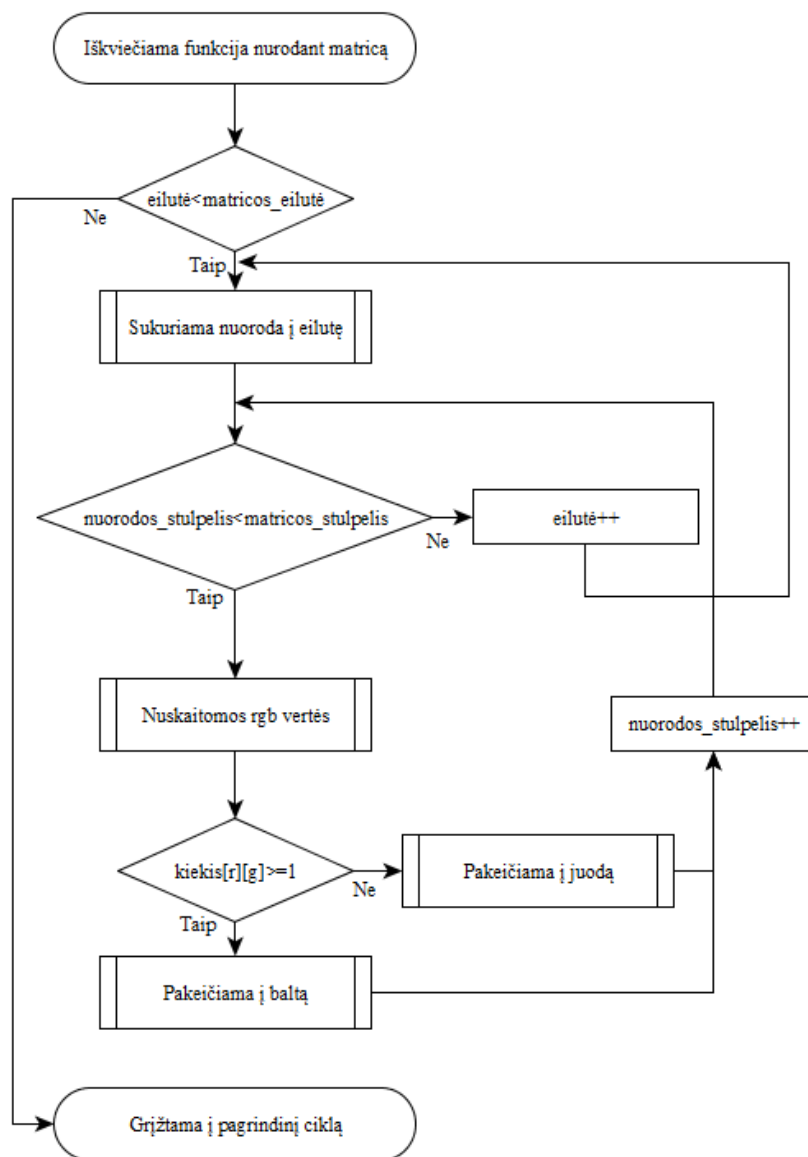
33 pav. Mokymo procedūra

2.2.6 Vaizdo segmentavimas ir papildomos funkcijos

Baigus mokymo procesą belieka atlikti kelio dangos segmentavimą. Segmentavimo metu visi įeinančio kadro taškai išrūšiuojami į dvi grupes: kelio dangą ir kiti nedominantys taškai. Rūšiavimas atliekamas pakeičiant kiekvieno taško spalvą, šiuo atveju kelio dangos taškai yra baltos spalvos, o visi kiti taškai juodos. Taip pat vizualizavimo tikslams gali būti pakeisti tik netinkami arba kelio dangos taškai, taip pradiniam kadre pažymint kelio dangą arba išskiriant kelio dangos tekstūrą. Taškų išrūšiavimo metu reikia nuskaityti ir patikrinti kiekvieno įeinančio vaizdo taško vertę.

Įeinančio kadro verčių nuskaitymo greitis vienas pagrindinių faktorių įtakančių visos sistemos apdorojimo spartą. Taikant įprastą įeinančio vaizdo verčių nuskaitymą bendra apdorojimo sparta gali būti pakankamai žema. Siekiant išlaikyti pakankamai didelę apdorojimo spartą taikomas nuorodomis pagrįstas įeinančio vaizdo kadro spalvos intensyvumo verčių nuskaitymo metodas. Taikant šį verčių nuskaitymo būdą įeinantis vaizdo kadras suskirstomas į eilutes ir apdorojamos nuorodos į jas. Nuoroda į eilutės informaciją yra vektorių masyvas, kuriame išsaugota visų trijų spalvų kanalų intensyvumo verčių informacija. Gautas vektorių

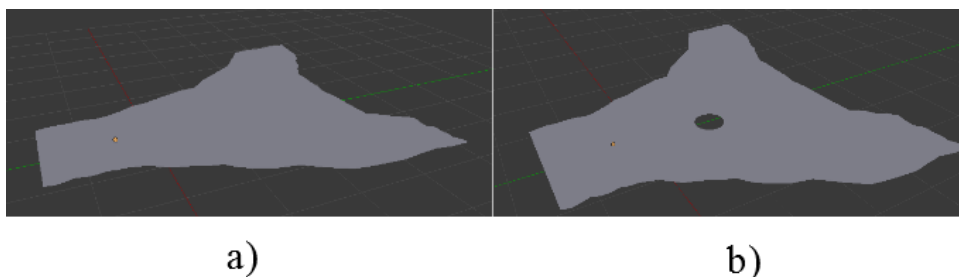
masyvas apdorojamas pasinaudojant įeinančio kadro stulpelių numeriais. Nuskaitant kiekvieno masyvo elemento raudonos, žalios ir mėlynos spalvų kanalų intensyvumo vertes, jų kombinacijos palyginamos su mokymo proceso metu apskaičiuotomis kombinacijomis. Kombinacijų palyginimo metu tikrinama ar apskaičiuotoje duomenų struktūroje tikrinamos spalvos intensyvumo kombinacijos taškų kiekis viršija vienetą (34 pav.). Jei taškų kiekis viršija vienetą taškas laikomas kelio dangos tašku ir jo spalvų kanalų vertės pakeičiamos į 255 (balta spalva), priešingu atveju kanalų vertės pakeičiamos į 0 (juoda spalva).



34 pav. Segmentavimo procedūra

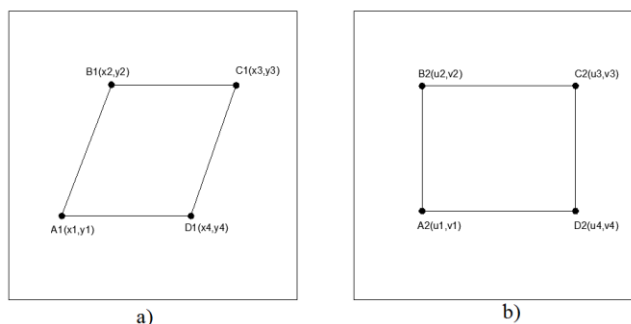
Tokiu būdu atlikus įeinančio vaizdo kadro segmentavimą galima spręsti apie kokybę, tačiau programa papildyta funkcijomis padedančiomis įvertinti skaičiavimo spartą bei išbandyti segmentavimo išėjime gautos informacijos pritaikomumą orientacijai. Kadro apdorojimo spartai įvertinti sudaryta laikmačio principu veikianti laiko skaičiavimo funkcija. Ši funkcija remiasi sistemos taktinių impulsų („clock tick“) skaičiavimu. Skaičiavimas pradedamas nuskaičius

įeinantį vaizdo kadra ir sustabdomas baigus segmentavimo procesą. Baigus matavimą laikas apskaičiuojamas matavimo rezultatą padalinus iš impulsų per sekundę (CLOCKS_PER_SEC). Segmentavimo rezultatams išbandyti sudaroma navigacinė plokštuma (35 pav. a). Sudaryta navigacinė plokštuma [25] gali būti panaudojama objekto orientacijai bei kliūčių (kliūtimis galima klasifikuoti skyles atsiradusias navigacinėje plokštumoje) išvengimui (35 pav. b).



35 pav. Navigacinė plokštuma: a – tuščia; b – su „kliūtimi“

Norint sudaryti navigacinę plokštumą reikia apskaičiuoti segmentuotos kelio dangos kraštų koordinates. Šiam procesui sudaroma papildoma funkcija atliekanti papildomą segmentavimo metu gautos kelio dangos valymą, bei jos kontūro koordinatų apskaičiavimą. Šioje funkcijoje papildomas vaizdo valymas atliekamas pasinaudojant standartinėmis „OpenCV“ paketo funkcijomis [22]. Valymo pagrindinis tikslas yra sumažinti segmentavimo metu gautos kelio dangos duomenų kiekį, sumažinant kanalų kiekį, spalvų skalę bei pašalinant triukšmingus regionus. Praleidus valymo procesą atsiranda pakankamai didelė paklaidų tikimybė bei sumažėja apdorojimo sparta. Baigus valymo procesą ir pasinaudojus standartine „OpenCV“ kontūrų apskaičiavimo funkcija gaunamas visos išvalyto vaizdo matricos kontūrų vektorinis masyvas. Idealiu atveju šiame masyve bus vienintelis viso kelio kontūras, tačiau siekiant išvengti klaidų atsirandančių dėl neefektyvaus valymo pasirenkamas didžiausias kontūras (jei apskaičiuotame masyve yra keli kontūrai). Išrinkto kontūro koordinatės perskaičiuojamos pritaikius perspektyvos transformaciją [23]. Šios transformacijos metu parenkami 8 taškai iš kurių sudaromas transformacijos koeficientų masyvas „ d “. Pirmi keturi taškai „ $A1$ “, „ $B1$ “, „ $C1$ “, „ $D1$ “ nurodo netransformuoto pradinio objekto koordinates (36 pav. a), o likę keturi taškai „ $A2$ “, „ $B2$ “, „ $C2$ “, „ $D2$ “ – transformuoto objekto koordinates (36 pav. b).



36 pav. Perspektivos transformacija: a) pradinis objektas; b) transformuotas objektas

Remiantis šiomis koordinatėmis pagal lygtį (17) apskaičiuojami transformacijos koeficientai „c00“ – „c21“ („c22“ visada lygus 1).

$$\begin{bmatrix} x1 & y1 & 1 & 0 & 0 & 0 & -x1 \cdot u1 & -y1 \cdot u1 \\ x2 & y2 & 1 & 0 & 0 & 0 & -x2 \cdot u2 & -y2 \cdot u2 \\ x3 & y3 & 1 & 0 & 0 & 0 & -x3 \cdot u3 & -y3 \cdot u3 \\ x4 & y4 & 1 & 0 & 0 & 0 & -x4 \cdot u4 & -y4 \cdot u4 \\ 0 & 0 & 0 & x1 & y1 & 1 & -x1 \cdot v1 & -y1 \cdot v1 \\ 0 & 0 & 0 & x2 & y2 & 1 & -x2 \cdot v2 & -y2 \cdot v2 \\ 0 & 0 & 0 & x3 & y3 & 1 & -x3 \cdot v3 & -y3 \cdot v3 \\ 0 & 0 & 0 & x4 & y4 & 1 & -x4 \cdot v4 & -y4 \cdot v4 \end{bmatrix} * \begin{bmatrix} c00 \\ c01 \\ c02 \\ c10 \\ c11 \\ c12 \\ c20 \\ c21 \end{bmatrix} = \begin{bmatrix} u1 \\ u2 \\ u3 \\ u4 \\ v1 \\ v2 \\ v3 \\ v4 \end{bmatrix}, \quad (17)$$

čia $x1, x2, x3, x4, y1, y2, y3, y4$ – netransformuotos koordinatės;
 $u1, u2, u3, u4, v1, v2, v3, v4$ – transformuotos koordinatės;
 $c00, c01, c02, c10, c11, c12, c20, c21$ – transformacijos koeficientai.

Sudarius transformacijos koeficientų masyvą ir taikant formules (18 ir 19) galima perskaičiuoti kiekvieno kontūro taško koordinatžių reikšmes.

$$xt = \frac{c00 \cdot x + c01 \cdot y + c02}{c20 \cdot x + c21 \cdot y + c22} \quad (18)$$

$$yt = \frac{c10 \cdot x + c11 \cdot y + c12}{c20 \cdot x + c21 \cdot y + c22} \quad (19)$$

Perskaičiuotas koordinatžių masyvas gali būti išsaugotas rinkmenoje apdorojimui kitose programose arba tiesiogiai panaudotas navigacinės plokštumos sudarymui. Tokiu būdu sudarytoje plokštumoje navigacija bei kliūčių išvengimas gali būti atliekamas taikant „A žvaigždės“, „Dijkstro“ [26] ar kitą trumpiausio kelio išvengiant kliūčių apskaičiavimo algoritmą gražinantį trumpiausio kelio koordinatžių seką.

2.3 Sistemos papildymas poliarizaciniais filtrais

Be sudaryto kelio dangos segmentavimo metodo bandymų šiame darbe taip pat yra tiriama poliarizacinių filtrų naudojimo įtaka segmentavimo rezultatams. Iš analitinėje dalyje įgytų žinių matyti, kad realiose kelio dangos atpažinimo situacijose aktyviai taikant poliarizacinius filtrus dėl dalinės poliarizacijos atsirastų didelės paklaidos dėl kurių sistema būtų

visiškai neefektyvi. Dėl šios priežastis bus tiriama pasyvaus poliarizacinių filtrų taikymo įtaka sudarytos segmentavimo metodikos rezultatams. Tiriant pasyvinių filtrų įtaką segmentavimo rezultatus, vaizdo kamera bus papildoma apskritiminiu poliarizaciniu filtru (37 pav.) pasuktu statmenai atspindėtai šviesai.



37 pav. CPL poliarizacinis filtras

Siekiant palyginti rezultatus nekeičiant kameros padėties nuskaitomi du vaizdo kadrai pritaikant poliarizacinį filtrą bei be filtro. Turint šiuos vaizdo kadrus ir taikant sudarytą metodą atliekamas kelio dangos segmentavimas. Segmentavimo rezultatams palyginti sudaroma funkcija, suskaičiuojanti teisingai bei klaidingai atpažintų taškų kiekį. Šie skaičiavimai atliekami remiantis prielaida, kad segmentavimo metu gautoje matricoje visi elementai yra kelio dangos taškai. Norint užtikrinti, kad segmentavimo rezultatų matricoje bus tik kelio dangos taškai, galima kamerą pastatyti taip, kad visą kadrą užimtų vien tik kelio danga arba prieš segmentavimą rankiniu būdu iškirpti kelio dangą iš bendro sistemos įėjime gauto vaizdo kadro. Taip pat siekiant įvertinti neatitikimų įtaką reikia apskaičiuoti visos kelio dangos taškų skaičių. Galiausiai žinant šiuos duomenis galima įvertinti kiek procentų taškų buvo atpažinta netinkamai atliekant įprastinį apdorojimą ir sistemą papildžius poliarizaciniais filtrais.

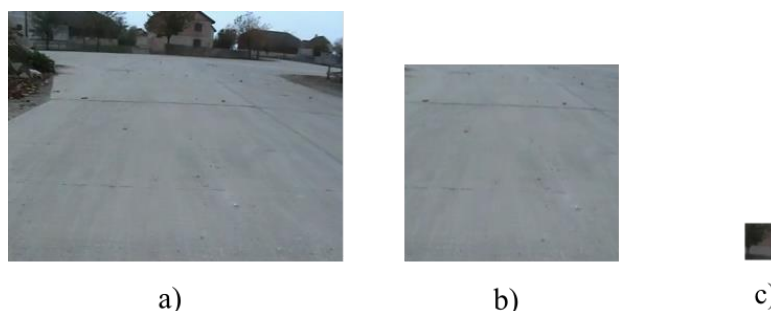
2.4 Skyriaus išvados

1. Tolstant nuo etaloninės zonos atsiranda dideli kiekiai taškų, kurių spalvos intensyvumo verčių kombinacijos yra artimos etaloninės zonos kombinacijoms.
2. Taikant OOP („Object Oriented Programming“) stilių galima lengvai modifikuoti programinės įrangos kodą.
3. Taikant programinės taškų masyvo nuorodas ženkliai sumažinamas kadro apdorojimo laikas.

3. TIRIAMOJI DALIS

3.1 Etaloninės zonos dydžio įtaka segmentavimo rezultatams

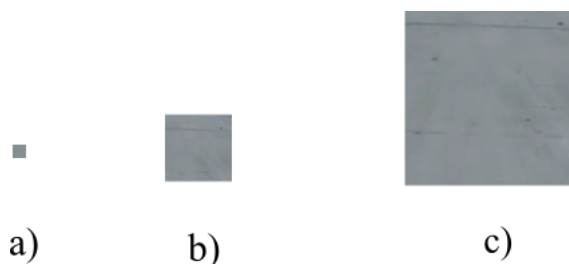
Siekiant išsiaiškinti kokią įtaką segmentavimo rezultatams turi etaloninės zonos dydis atliekamas bandymas, kurio metu apdorojamas pasirinktas 321 taškų pločio ir 239 taškų ilgio vaizdo kadras (38 pav. a). Įvertinant etaloninės zonos dydžio įtaką tikslinga skaičiuoti bandymo metu atpažintų taškų kiekį. Šiam tikslui iš vaizdo kadro (38 pav. a) iškerpamas 205 taškų ilgio bei 188 taškų pločio kelio dangos fragmentas (38 pav. b), kurį apdorojus bus skaičiuojamas atpažintų taškų kiekis. Pritaikius šią etalono dydžio įtakos įvertinimo metodiką ir etalonu pasirinkus visą vaizdo kadrą (38 pav. a) bus gauti klaidingi rezultatai (sistema atpažins 100% fragmento taškų). Siekiant išvengti klaidingų rezultatų įvedamas papildomas 35 taškų ilgio bei pločio pradinio kadro (38 pav. a) viršuje esančio namo fragmentas (38 pav. c). Apdorojus šiuos vaizdo fragmentus (38 pav. b ir c) bus gauti teigiamų bei neigiamų vaizdo fragmentų taškų masyvai.



38 pav. Bandyme naudojama informacija: a – kadras; b – teigiamas; c – neigiamas

Suskaičiavus teigiamo vaizdo fragmento taškus bus įvertintas teisingai atpažintų taškų kiekis, o neigiamus – klaidingai atpažintų taškų kiekis. Idealiu atveju teigiamo vaizdo fragmento taškų kiekis turėtų siekti 100%, o neigiamo – 0%. Tai reikštų, kad buvo atpažinti visi kelio dangos taškai ir visiškai išvengta netikrų taškų atpažinimo.

Šiam bandymui iš pradinio vaizdo kadro parenkami 7 skirtingo dydžio etaloniniai fragmentai: 10x10, 30x30, 50x50, 70x70, 90x90, 110x110, 130x130, 150x150.

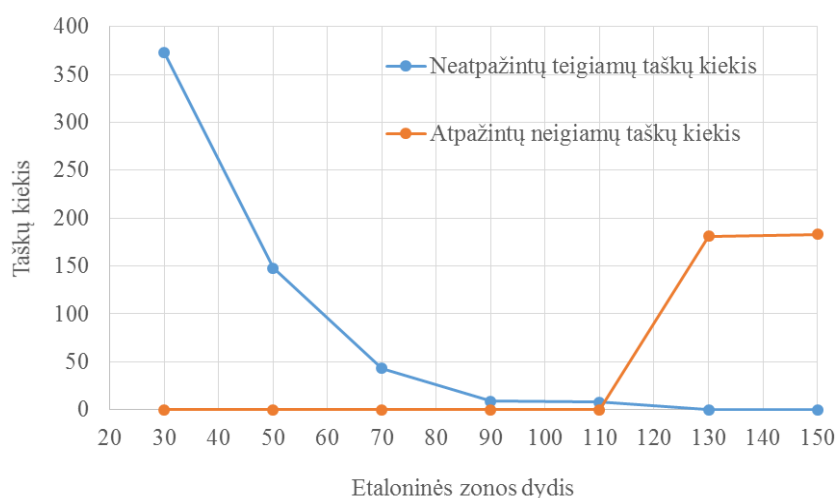


39 pav. Etaloniniai fragmentai: a – 10x10; b – 50x50; c – 130x30

Etaloninės zonos dydžio įtakos segmentavimo rezultatams lentelė

Etaloninės zonos dydis	Atpažinti teigiami taškai	Atpažinti neigiami taškai	Teigiamų taškų kiekis	Neigiamų taškų kiekis	Atpažinta teigiamų, %	Atpažinta neigiamų, %
10x10	35873	0	38540	1225	93,08	0
30x30	38167	0			99,03	0
50x50	38392	0			99,62	0
70x70	38497	0			99,89	0
90x90	38531	0			99,98	0
110x110	38532	0			99,98	0
130x130	38540	181			100	14,78
150x150	38540	183			100	14,94

Bandymo metu patikrinus 38540 teigiamų ir 1225 neigiamų taškų pastebimas pakankamai geras taškų atpažinimas (1 lentelė). Apmokymui pasirinkus 10x10 taškų dydžio etaloninę zoną sudarančią 0,13% pradinio vaizdo kadro ploto atpažintų taškų kiekis siekia 93,08%. Etaloninę zoną padidinus iki 1,17% pradinio vaizdo kadro plote atpažįstamas teigiamų taškų kiekis padidėja iki 99,03%. Keičiant etaloninės zonos plotą nuo 1,17% iki 15,77% nepastebimas ryškus atpažintų teigiamų taškų kiekio pokytis. Šį plotą didinant nuo 22,03% iki 29,33% pradinio vaizdo ploto pasireiškia neigiamas etaloninės zonos dydžio poveikis. Nors pasiektas idealus atpažintų teigiamų taškų kiekis (100%), tačiau atsirado neteisingai klasifikuoti neigiami taškai, kurių kiekis siekia 14,78% – 14,94% neigiamo fragmento taškų kiekio.

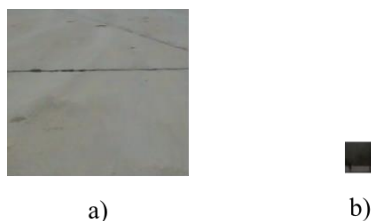


40 pav. Neatpažintų ir neteisingai atpažintų taškų kiekio priklausomybė nuo etaloninės zonos dydžio

Nubrėžus neatpažintų teigiamų taškų bei atpažintų neigiamų taškų priklausomybės nuo etaloninės zonos dydžio grafiką (40 pav.) pastebima, kad didinant etaloninę zoną matomas eksponentinis etaloninės zonos dydžio įtakos mažėjimas atpažinimo kokybei. Apibendrinant bandymo rezultatus galima teigti, kad etaloninės zonos plotui viršijus 22% pradinio vaizdo kadro ploto apdorojimo išėjime atpažįstami pašaliniai objektai, kurių spalva yra artima kelio dangos spalvai. Taip pat konkrečiu atveju siekiant pagerinti teigiamų taškų atpažinimą nėra efektyvu didinti etaloninės zonos plotą daugiau nei 1,2% pradinio vaizdo ploto.

3.2 Segmentavimo rezultatai keičiant σ_x ir σ_y vertes

Sistemos mokymo proceso metu σ_x ir σ_y vertės apskaičiuojamos automatiškai, tačiau pravartu išbandyti kaip pasikeis segmentavimo rezultatai pakeitus šias vertes. Šiame bandyme parenkant įvairias σ_x ir σ_y vertes sistema apmokoma panaudojant 110x110 etaloninę zoną iškirptą iš pradinio vaizdo kadro (39 pav. a). Apmokius sistemą segmentavimas atliekamas fragmentams iškirptiems iš pradinio kadro (38 pav. b ir c) bei iškirptiems iš kito kadro užfiksuoto toli nuo pradinės zonos (41 pav. a ir b).



41 pav. Vaizdo fragmentai: a – teigiamas; b – neigiamas

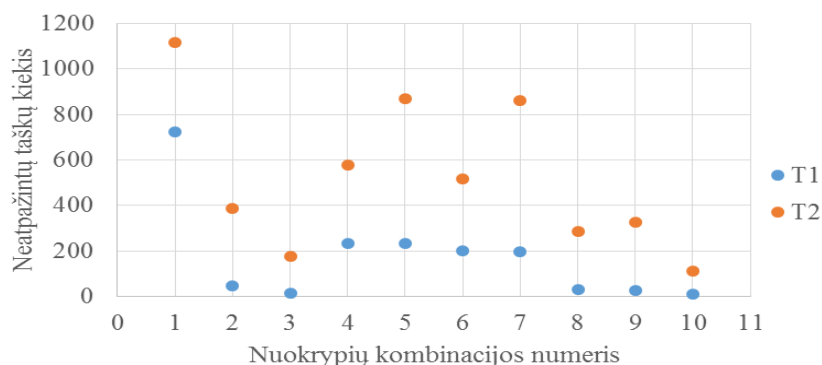
Panašiai kaip ir ankstesniame bandyme rezultatų palyginimui skaičiuojamas dviejuose skirtinguose atstumuose užfiksuotų teigiamų ir neigiamų vaizdo fragmentų atpažintų taškų kiekis pasirinkus įvairias σ_x ir σ_y verčių kombinacijas. Taip pat gautų rezultatų palyginimui apskaičiuojamas taškų kiekis, kai mokymo metu σ_x ir σ_y vertės apskaičiuojamos automatiškai. Remiantis teigiamuose bei neigiamuose vaizdo fragmentuose atpažintų taškų kiekiais siekiama palyginti kaip kinta atpažintų taškų kiekis nežymiai keičiantis aplinkai taikant iš anksto parinktas bei sistemos mokymo metu apskaičiuotas standartinio nuokrypio σ_x ir σ_y vertes. Taip pat svarbu paminėti, kad gauti rezultatai neatspindi drastiško kelio dangos struktūros pokyčio. Tai reiškia, kad bandyme parinktos standartinio nuokrypio vertės nėra universalios visoms įmanomos kelio dangos tekstūroms.

Segmentavimo rezultatų keičiant standartinio nuokrypio vertes lentelė

σ_x	σ_y	T1	N1	T2	N2	Teigiamo fragmento taškų kiekis	Neigiamo fragmento taškų kiekis
5	5	37818	0	37423	0	38540	1125
10	10	38493	0	38152	0		
15	15	38528	0	38363	0		
5	10	38308	0	37964	0		
10	5	38308	0	37671	0		
5	15	38341	0	38024	0		
15	5	38343	0	37680	0		
10	15	38511	0	38254	0		
15	10	38512	0	38215	0		
Automatiškai apskaičiuota	Automatiškai apskaičiuota	38532	0	38429	0		

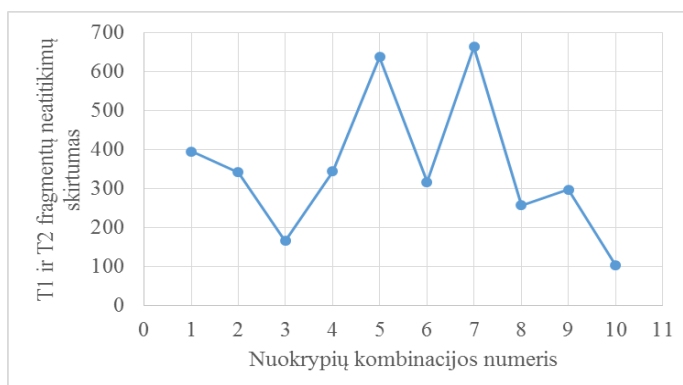
Atlikus bandymą sudaryta rezultatų lentelė (2 lentelė), kurioje: „T1“ – pradinio vaizdo kadro teigiamame fragmente atpažintų taškų kiekis, „N1“ – pradinio vaizdo kadro neigiamame fragmente atpažintų taškų kiekis, „T2“ – toli nuo mokymo zonos užfiksuoto vaizdo kadro teigiamame fragmente atpažintų taškų kiekis, „N2“ – toli nuo mokymo zonos užfiksuoto vaizdo kadro neigiamame fragmente atpažintų taškų kiekis. Analizuojant bandymo metu gautus duomenis (2 lentelė) pirmiausiai pastebimas neigiamuose vaizdo fragmentuose atpažintų taškų kiekis. Visais atvejais nepriklausomai nuo pasirinktų standartinio nuokrypio verčių išvengta neigiamų taškų atpažinimo.

Nagrinėjant teigiamų fragmentų apdorojimo metu gautus duomenis („T1“ ir „T2“) nesunku pastebėti, kad visais atvejais atpažinta bent 97% visų fragmente esančių taškų. Daugiausiai taškų neatpažinta (1117 – 2,9% visų fragmento taškų) fragmente „T2“ kai standartiniai nuokrypiai σ_x ir σ_y buvo lygūs 5, o mažiausiai – (8 – 0,02% visų fragmento taškų) fragmente „T1“ kai standartiniai nuokrypiai σ_x ir σ_y buvo automatiškai apskaičiuoti sistemos mokymo metu. Siekiant palyginti kaip keičiasi neatpažintų teigiamų fragmentų taškų kiekiai, nubrėžiamas grafikas (42 pav.), kurio x ašyje pažymimas lentelėje (2 lentelė) esančios standartinių nuokrypių kombinacijos numeris (5 ir 5 – 1, o automatiškai apskaičiuotas – 10), o y ašyje pažymimas neatpažintų taškų kiekis.



42 pav. Atskirų fragmentų neatpažintų taškų kiekis kiekvieno matavimo metu

Iš grafike (42 pav.) pavaizduotų duomenų pastebimas ryškus fragmente „T2“ neatpažintų taškų kiekio išsibarstymas. Tuo tarpu fragmento „T1“ neatpažintų taškų kiekiai išsibarstę pakankamai mažame diapazone išskyrus pirmojo matavimo metu, kai standartiniai nuokrypiai buvo lygūs 5, o neatpažintų taškų kiekis – 772. Įvertinant bendrą taškų kiekio neatitikimą tiek fragmento „T1“, tiek fragmento „T2“ atvejais nubrėžiamas dar vienas grafikas apjungiantis abiejų fragmentų duomenis.



43 pav. T1 ir T2 fragmentų neatitikimų skirtumas

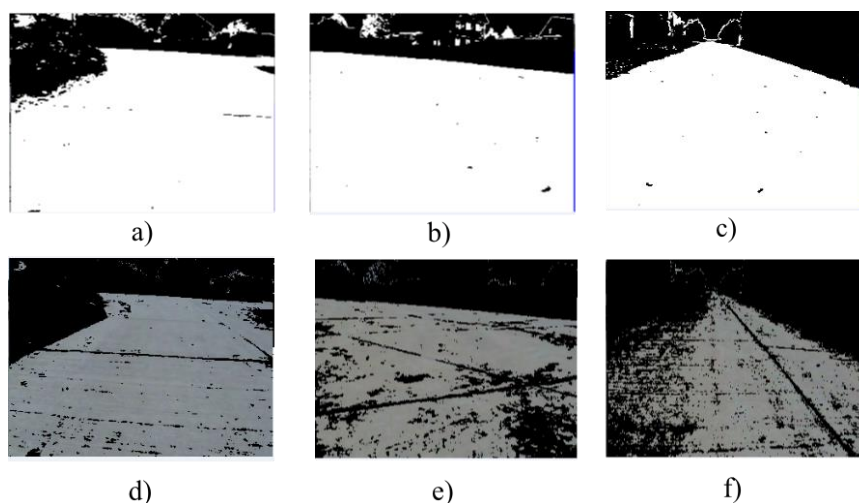
Šio grafiko (43 pav.) y ašyje pažymimas bandyme naudotuose fragmentuose („T1“ ir „T2“) neatitikusių taškų kiekio skirtumas, taip pavaizduojant neatpažintų taškų kiekio kitimą naudojant kiekvieną standartinių nuokrypių kombinaciją ir keičiant apdorojamą fragmentą. Iš grafike pavaizduotų duomenų matyti dvi aiškios taškų grupės. Pirmoji taškų grupė, kai standartinių nuokrypių kombinacijų numeriai lygūs 5 ir 7. Esant šioms kombinacijoms pastebimas didžiausias neatpažintų taškų kiekio pokytis leidžiantis teigti, kad keičiant apdorojamą fragmentą rezultatai kis dideliame diapazone ir galimos didelės paklaidos. Antrajai taškų grupei galima priskirti 3 ir 10 nuokrypio verčių kombinacijas. Esant šioms taškų kombinacijoms pastebimas mažiausias taškų kiekio pokytis (165 ir 103 taškai), tačiau kaip minėta anksčiau iš anksto parinktos standartinio nuokrypio vertės (3 kombinacija 43 pav.) gali

netikti stipriai keičiantis kelio dangos tekstūrai. Dėl šios priežasties geriausi rezultatai (mažiausias taškų kiekio pokytis bei mažiausias neatpažintų taškų kiekis) pasiekiami automatiškai kiekvieno mokymo metu apskaičiuojant standartinio nuokrypio vertes (10 kombinacija 43 pav.).

3.3 Segmentavimo rezultatai tolstant nuo pradinio etaloninio fragmento zonos

Iš ankstesnių bandymų rezultatų matyti, kad mokymo zonosose pasiekiami pakankamai geri segmentavimo rezultatai, tačiau neįvertinamas šių rezultatų kitimas nutolstant nuo mokymo zonos. Siekiant įvertinti šį pokytį atliekamas bandymas, kurio metu apdorojami skirtinguose atstumuose užfiksuotos kelio dangos vaizdo kadrai. Šiam bandymui parinkti trys 320 taškų ilgio ir 240 taškų pločio skirtingais atstumais užfiksuoti vaizdo kadrai. Sistemos apmokymui visais atvejais panaudojamas iš pirmojo kelio dangos vaizdo kadro iškirptas 100 taškų pločio ir 100 taškų ilgio vaizdo fragmentas. Galiausiai atlikus bandyme naudojamų vaizdo kadro segmentavimą (44 pav. a, b, c) iškerpami 21000 taškų dydžio fragmentai ir suskaičiuavus šiuose fragmentuose atpažintų taškų kiekį atliekamas rezultatų palyginimas.

Papildomam sudaryto metodo rezultatų įvertinimui pravartu juos palyginti su literatūroje aprašomais metodais. Šiam tikslui panašus bandymas atliekamas taikant literatūroje aprašomą Gauso maišos metodu pagrįstą kelio dangos segmentavimo būdą. Panašiai kaip ir sudarytame metode apmokymas atliekamas pasinaudojant tuo pačiu pirmame kelio dangos kadre iškirptu vaizdo fragmentu. Atlikus sistemos apmokymą kaip ir sudarytame metode baigus segmentavimo procesą tose pačiose pradinių vaizdo kadro (44 pav. d, e, f) vietose iškerpami tokio paties ploto vaizdo fragmentai. Be metodais atpažįstamų taškų kiekio palyginimo taip pat svarbu įvertinti kiek laiko trunka vieno kadro apdorojimas. Abiem atvejais šis laikas pradedamas skaičiuoti baigus sistemos mokymą ir užbaigiamas atlikus segmentavimo procesą.

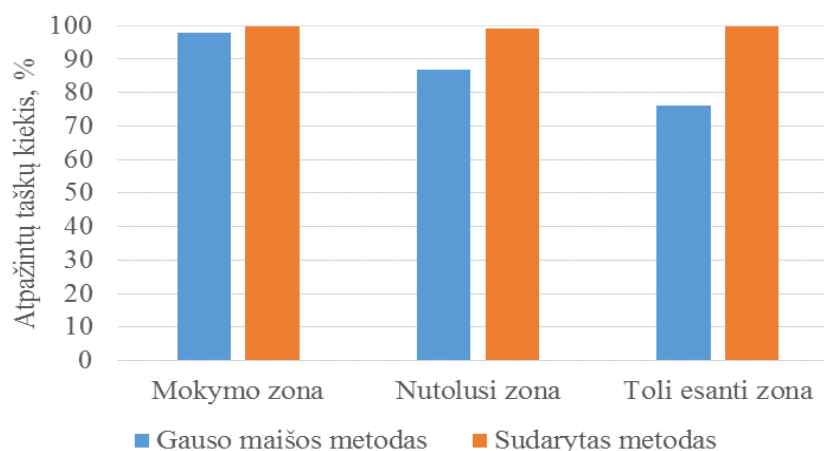


44 pav. Segmentavimo rezultatai: a, b, c – sudarytas metodas(mokymo zona, nutolus, toli); d, e, f – Gauso maišos metodas (mokymo zona, nutolus, toli)

Gauso maišos ir sudaryto metodo segmentavimo rezultatų lentelė

	Gauso maišos metodas	Sudarytas metodas
Vaizdo kadro dydis	320x240	
Apdorojamo fragmento taškų skaičius	21000	
Mokymo zonoje atpažintų taškų skaičius	20569	20929
Nutolusioje zonoje atpažintų taškų skaičius	18219	20815
Toli esančioje zonoje atpažintų taškų skaičius	15991	20933
Vieno vaizdo kadro apdorojimo trukmė, ms	110	20

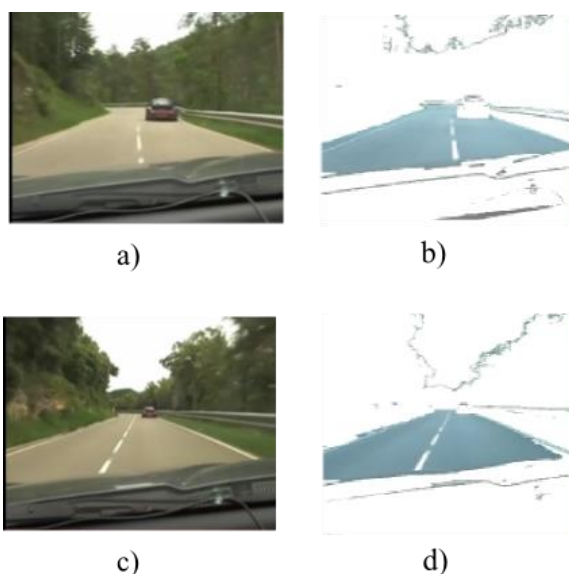
Nagrinėjant bandymo metu sudarytą lentelę (3 lentelė) pastebimas ryškus toli esančioje zonoje (44 pav. f) atpažintų taškų kiekio sumažėjimas (45 pav.). Šioje zonoje atpažintų taškų kiekis siekė 76% visų iškirpto fragmento taškų kiekio (21000). Geriausi Gauso maišos metodo rezultatai pasireiškia pradinėje mokymo zonoje ir siekia 98% visų fragmento taškų, tačiau kintant nuotoliui kiekvienoje tolesnėje zonoje atpažintų taškų kiekis sumažėjo 10% – 11% (45 pav.). Nors trečioje zonoje išlieka gana didelis atpažintų taškų kiekis (76% visų iškirpto fragmento taškų), tačiau segmentavimo išėjime (44 pav. f) gautoje matricoje kairė kelio dangos pusė praktiškai neatpažįstama. Iš gautų rezultatų galima daryti prielaidą, kad kelio dangos segmentavimui taikant Gauso maišos metodą ir toliau tolstant nuo pradinės mokymo zonos kelio danga bus visiškai neatpažįstama.



45 pav. Atpažintų taškų kiekio palyginimas

Tuo tarpu taikant sudarytą metodą matomas panašus atpažintų taškų kiekis visose nutolusiose zonose (45 pav.). Visais trimis atvejais atpažintų taškų kiekis siekia bent 99% visų iškirpto kelio dangos fragmento taškų, o mažiausias atpažintų taškų kiekis užfiksuotas nutolusioje zonoje ir siekė 99,12%. Iš šių rezultatų galima daryti prielaidą, kad taikant sudarytą metodą skirtingai nei Gauso maišos (GMM) atveju tolstant nuo pradinės mokymo zonos segmentavimo rezultatai turėtų kisti nežymiai arba kur kas mažesniu tempu nei GMM atveju. Be segmentavimo metu atpažintų taškų kiekio skirtumų pastebimas ženklus apdorojimo spartos skirtumas. Įeinančio vaizdo kadro segmentavimui pritaikius sudarytą metodą, vieno vaizdo kadro apdorojimas trunka 5,5 karto trumpiau (20 ms) nei taikant literatūroje aprašomą Gauso maišos metodą (110 ms).

Kadangi pritaikius sudarytą metodą pasiektas pakankamai geras atpažįstamų taškų kiekis pravartu išbandyti segmentavimo rezultatus žymiai didesniame nuotolyje. Šiam tikslui pasirinktas devynių minučių ilgo kelionės automobiliu vaizdo įrašas. Šio vaizdo įrašo pirmojo kadro (46 pav. a) informacija panaudojama sistemos apmokymui, o 16200 – tojo kadro informacija (46 pav. c) panaudojama kelio dangos segmentavimo rezultatų dideliame nuotolyje palyginimui.



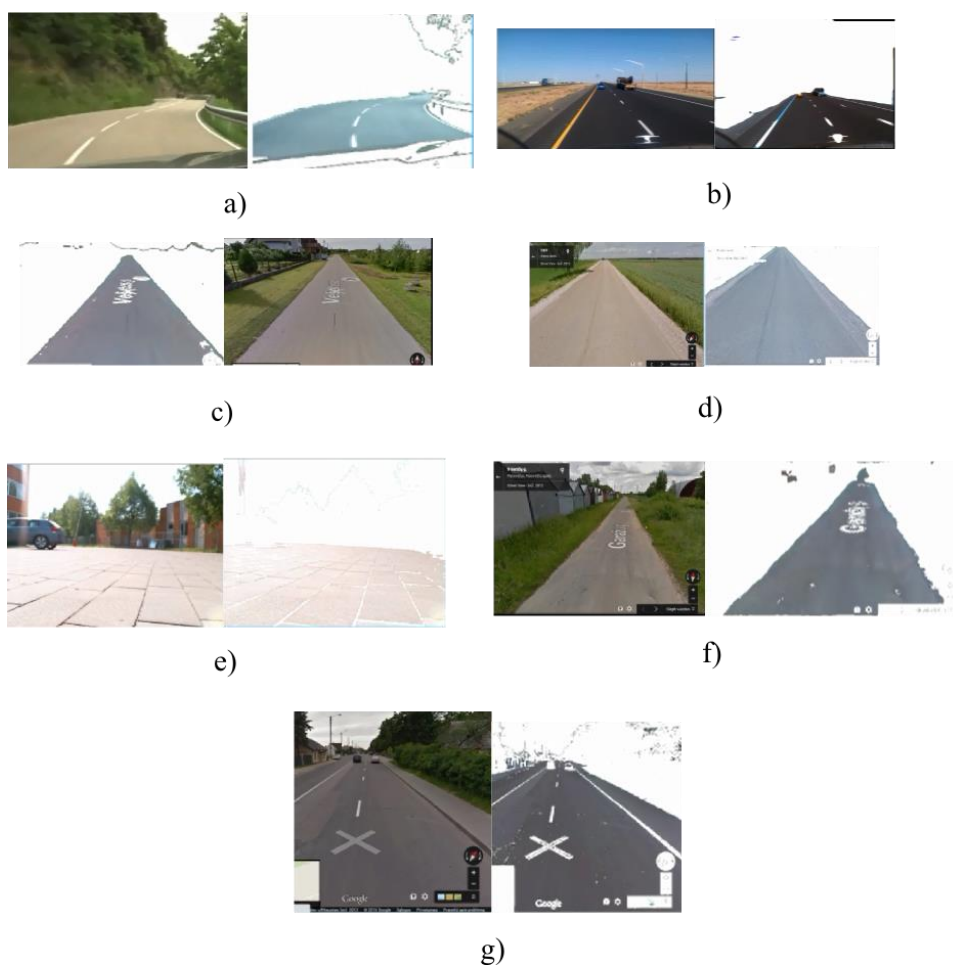
46 pav. Segmentavimo rezultatai dideliame nuotolyje: a – pradinis mokymo zonos kadras; b – pradinio kadro segmentavimo rezultatai; c – 16200 vaizdo kadras; d – 16200 vaizdo kadro segmentavimo rezultatai

Lyginant šio vaizdo įrašo kadrų segmentavimo rezultatus pradinėje mokymo zonoje (46 pav. b) ir dideliame nuotolyje (46 pav. d) pastebimi minimalūs neatitikimai parodantys, kad pritaikius sudarytą metodą segmentavimo metu atpažintų taškų kiekis nežymiai kinta tolstant nuo pradinės mokymo zonos ir yra artimas pirmojo bandymo rezultatams (3 lentelė).

3.4 Segmentavimo rezultatai esant įvairioms kelio dangos tekstūroms

Be atpažinimo nuotolio taip pat svarbu patikrinti ar tokie pat segmentavimo rezultatai gaunami keičiantis kelio dangos tekstūrai. Šiam tikslui atliekamas bandymas, kurio metu apdorojami ir palyginami skirtingose kelio dangos tekstūrose atpažintų taškų kiekiai. Bandyme naudojami žvyrkelio ir asfalto kelio dangų pavyzdžiai. Šiuos pavyzdžius sudaro „Google Street View“ pagalba gauti bei realiose situacijose užfiksuoti kelio dangos vaizdo fragmentai.

Pirmiausiai palyginami skirtingų tekstūrų asfalto dangų pavyzdžiai (47 pav. a, b, c, d, e, f, g kairėje). Kaip ir ankstesniuose bandymuose kiekvienam vaizdo pavyzdžiui atliekamas taškų segmentavimo procesas (47 pav. a, b, c, d, e, f, g dešinėje) ir iškerpamas apdorotos kelio dangos matricos fragmentas.



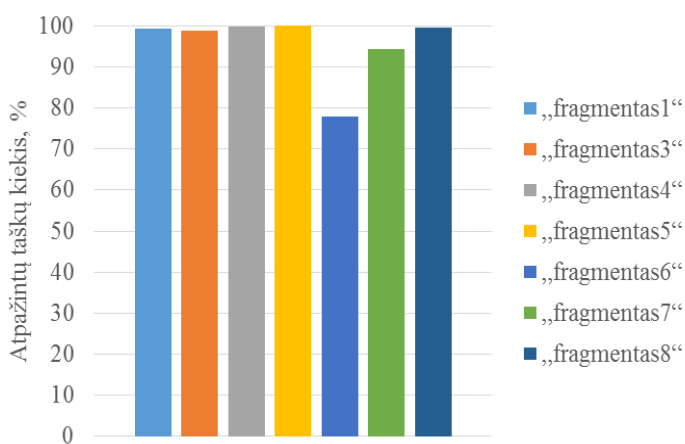
47 pav. Asfalto segmentavimo rezultatai (pradinis vaizdo kadras kairėje, segmentavimo rezultatai dešinėje): a – „fragmentas1“; b – „fragmentas3“; c – „fragmentas4“; d – „fragmentas5“; e – „fragmentas6“; f – „fragmentas7“; g – „fragmentas8“

Galiausiai iškirptame fragmente suskaičiuojamas atpažintų taškų kiekis, kuris bus palygintas su visų iškirpto fragmento taškų skaičiumi, tačiau svarbu paminėti, kad visi šie vaizdo pavyzdžiai yra skirtingo dydžio todėl skirsis ir kiekvieno fragmento bendras taškų skaičius.

Asfalto dangos segmentavimo rezultatų lentelė

Fragmentas	Fragmente atpažintų taškų kiekis	Bendras fragmento taškų kiekis	Atpažinta taškų, %
„fragmentas1“	5159	5200	99,2
„fragmentas3“	1999	2025	98,7
„fragmentas4“	10586	10600	99,9
„fragmentas5“	24497	24497	100
„fragmentas6“	12152	15576	78
„fragmentas7“	4000	4240	94,3
„fragmentas8“	1812	1820	99,6

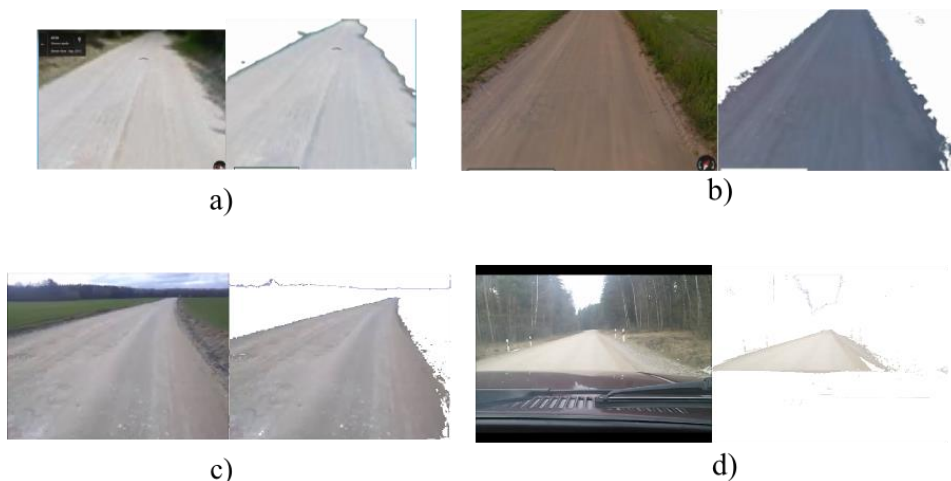
Atlikus bandymą sudaryta rezultatų lentelė (4 lentelė). Analizuojant atpažintų taškų kiekį (48 pav.) pastebimas ženklus „fragmentas6“ atpažintų taškų kiekio sumažėjimas siekiantis 16% (lyginant su kitu mažiausiu kiekiu „fragmentas7“). Siekiant paaiškinti šį taškų kiekio sumažėjimą pravartu palyginti bandymui naudotus pradinis vaizdo kadrus (47 pav. a, b, c, d, e, f, g kairėje). Palyginus šiuos vaizdo kadrus galima pastebėti, kad „fragmentas6“ atveju (47 pav. e kairysis viršutinis pradinio kadro kampas) kamera buvo akinama saulės. Kitas galimas šių neatitikimų paaiškinimas gali būti ženklus tam tikrų analizuojamo vaizdo kadro elementų spalvų nuokrypis, tačiau panašus vaizdo kadro elementų nuokrypis pastebimas ir kituose vaizdo kadruose (47 pav. d, f).



48 pav. Atpažintų taškų kiekio palyginimas

Analogiška bandymų seka atliekama ir žvyro kelio dangos tekstūros rezultatų palyginimui. Žvyro dangos segmentavimo bandymui parinkti 2 „Google Street View“ vaizdo kadrai (49 pav. a, b) ir 2 realiomis sąlygomis užfiksuoti vaizdo kadrai (49 pav. c, d). Galiausiai taip pat kaip ir asfalto tekstūros palyginimui iš kiekvieno apdoroto vaizdo kadro palyginimui

iškerpamas skirtingo ploto (priklausomai nuo pradinio vaizdo kadro dydžio) bandomasis fragmentas.



49 pav. Žvyrkelio segmentavimo rezultatai (pradinis vaizdo kadras kairėje, segmentavimo rezultatai dešinėje): a – „fragmentas2“; b – „fragmentas9“; c – „fragmentas10“; d – „fragmentas11“

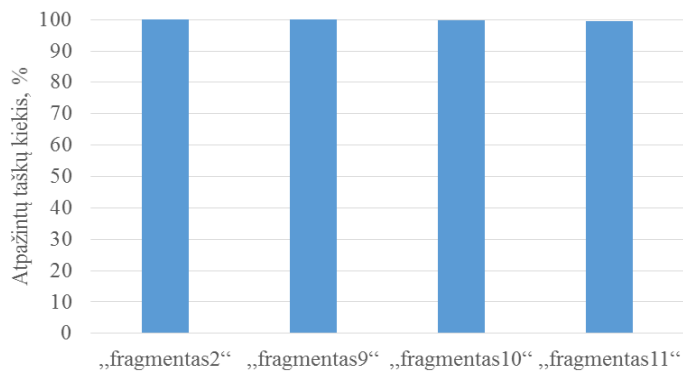
Atlikus bandymą sudaryta žvyro kelio dangos segmentavimo rezultatų lentelė (5 lentelė).

5 lentelė

Žvyro kelio dangos tekstūros segmentavimo rezultatų lentelė

Fragmentas	Fragmente atpažintų taškų kiekis	Bendras fragmento taškų kiekis	Atpažinta taškų, %
„fragmentas2“	10900	10900	100
„fragmentas9“	10776	10780	99,96
„fragmentas10“	34282	34314	99,9
„fragmentas11“	12714	12771	99,6

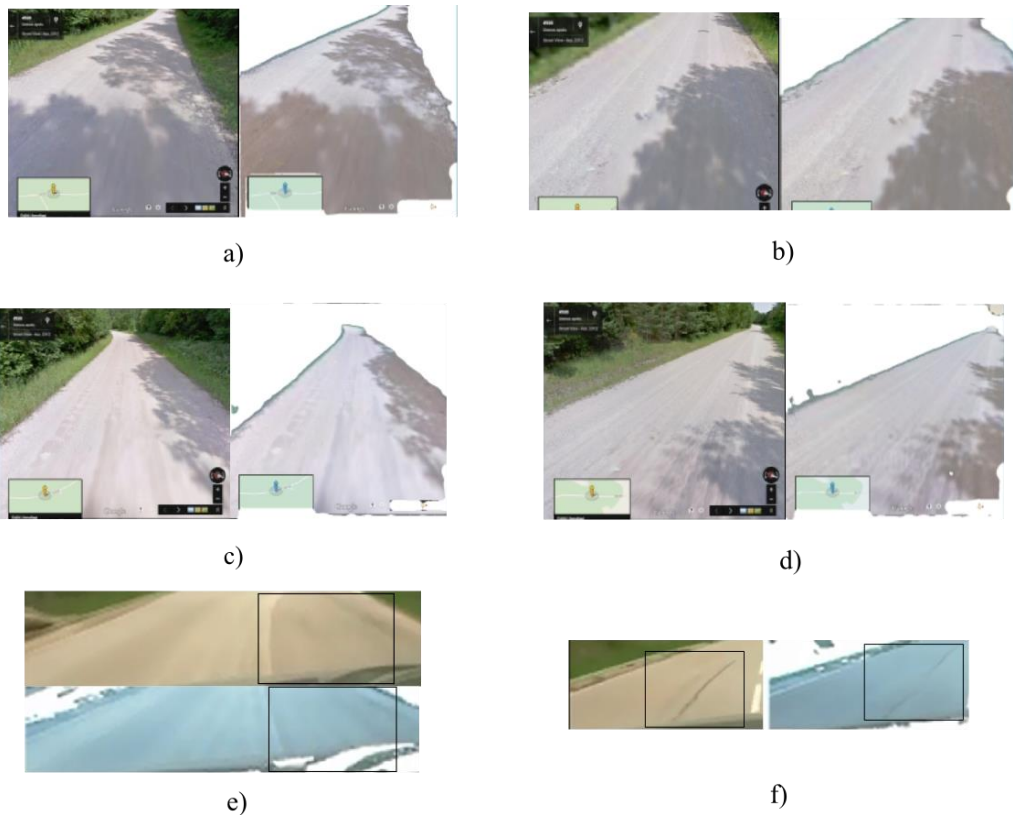
Palyginus bandymo rezultatus (50 pav.) visais keturiais atvejais (49 pav.) pasiekiami itin geri kelio dangos segmentavimo rezultatai siekiantys 99%. Palyginus abiejų kelio rūšių kelio dangos bandymų rezultatus matomi beveik identiški rezultatai. Remiantis šiais duomenimis galima daryti prielaidą, kad panašūs rezultatai bus gauti atpažįstant įvairių tekstūrų kelio dangas. Viso bandymo metu vienintelis žymus segmentavimo rezultatų neatitikimas pastebėtas bandymuose naudotame vaizdo fragmente „fragmentas6“, kurį dalinai galima paaiškinti vaizdo kameros akinimu. Jei šis neatitikimas susijęs su saulės įtaka, vaizdo kamera fiksuojamam vaizdui, tada pravartu išbandyti poliarizacinio filtro įtaką segmentavimo rezultatams realioje situacijoje palyginant sistemos su filtru bei sistemos be filtro segmentavimo rezultatus.



50 pav. Atpažintų taškų kiekio palyginimas

3.5 Kelio dangos tekstūros neatitikimų bei šešėlių įtaka segmentavimo rezultatams

Praktikoje taikant etalonine informacija pagrįstą sistemą gali atsirasti sutrikimų dėl kelio dangos tekstūros neatitikimų ar ant jos krentančio šešėlio įtakos. Šių veiksnių įtakai įvertinti atliekamas paprastas bandymas, kurio metu pasirenkami vaizdo kadrai, turintys tekstūros neatitikimų arba kuriuose ant kelio dangos krinta šešėlis. Turint tokių vaizdo pavyzdžių rinkinį atliekamas kiekvieno vaizdo kadro segmentavimas ir stebima ar teisingai atpažįstamas plotas, turintis anksčiau minėtų tekstūros neatitikimų (51 pav.).



51 pav. Segmentavimo rezultatai esant įvairiems neatitikimams: a – šešėlis 1; b – šešėlis 2; c – šešėlis 3; d – šešėlis 4; e – tekstūros neatitikimas 1; f – tekstūros neatitikimas 2

Atlikus bandymą iš rezultatų (51 pav. a, b, c, d) matyti, kad sudaryto metodo segmentavimo rezultatai mažai priklauso nuo šešėlių, krentančių ant kelio dangos. Panašūs rezultatai gauti patikrinus kelio dangos fragmentą, turintį tekstūros neatitikimų (51 pav. e, f pažymėta kvadratu). Apibendrinant bandymo rezultatus galima teigti, kad sudarytas metodas pakankamai gerai susidoroja su šešėlių bei kelio dangos tekstūros neatitikimų įtaka segmentavimo rezultatams.

3.6 Kelio prasto matomumo įtaka segmentavimo rezultatams

Kitas praktikoje dažnai pasitaikantis trikdys yra suprastėjęs matomumas. Pagrindiniai suprastėjusio matomumo šaltiniai yra rūkas, pūga ir kelio danga padengta sniegu. Šių trikdžių įvertinimui parenkami keturi vaizdo kadrai (52 pav.), kuriuose kelio danga yra padengta dideliu (52 pav. a) arba mažu (52 pav. b) sniego sluoksniu arba dėl pūgos poveikio suprastėjęs kelio dangos matomumas (52 pav. c, d).



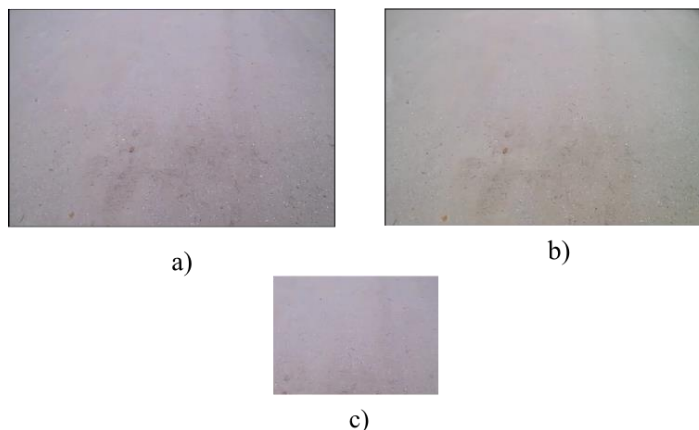
52 pav. Kelio dangos segmentavimo rezultatai esant prastam matomumui (pradinis kadras kairėje, rezultatai dešinėje): a – visiškai sniegu padengta kelio danga; b – mažu sniego kiekiu padengta kelio danga; c – mažu sniego kiekiu padengta kelio danga esant pūgai; d – kelio danga esant pūgai

Šių vaizdo pavydžių apdorojimui iš kiekvieno kadro iškerpamas 100 taškų ilgio bei 100 taškų pločio etaloninis fragmentas, kuriuo apmokius sistemą atliekamas segmentavimo procesas. Iš gautų rezultatų (52 pav. a, b, c, d dešinėje) matyti, kad visais keturiais atvejais gauti pakankamai geri rezultatai ir galima teigti, kad šie rezultatai išliks pakankamai geri tol, kol kelio dangos intensyvumo vertės bent nežymiai skirsis nuo aplinkinių objektų spalvos intensyvumo verčių. Taip pat iš dalinai sniegu padengtos kelio dangos (52 pav. b) segmentavimo rezultatų galima daryti išvadą, kad jei etaloniniame fragmente yra nors keli tekstūros neatitikimų taškai, visa kelio danga bus atpažinta tinkamai t. y. jei kelio danga apipilta dažais arba šiuo atveju maži

sniego plotai ir pažymėtame etaloniniame fragmente užfiksuotas bent vienas šio neatitikimo pavyzdys, tada tokie plotai bus laikomi kelio dangos dalimi.

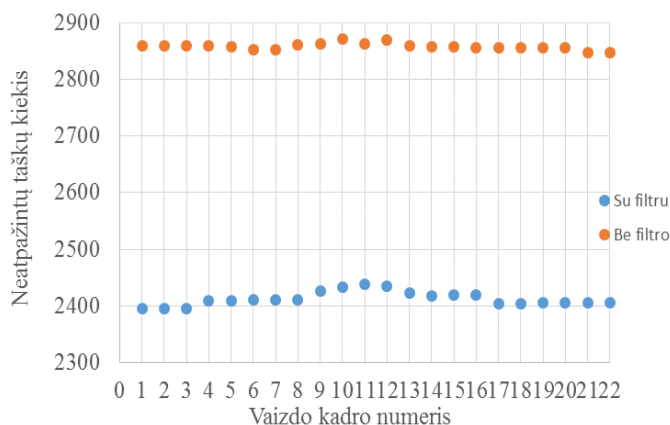
3.7 Poliarizacinių filtrų taikymo įtaka segmentavimo rezultatams

Poliarizacinių filtrų taikymo įtakai nustatyti sudaromas dviejų toje pačioje vietoje užfiksuotų vaizdo įrašų segmentavimo rezultatų palyginimo bandymas. Vaizdo pavyzdžių sudarymui kamera nukreipiama į kelio dangą ir nekeičiant jos pozicijos įrašomi du vaizdo įrašai, kai priešais kamerą pastatomas poliarizacinis filtras (53 pav. b) ir be filtro (53 pav. a).



53 pav. Bandyme naudojami vaizdo pavyzdžiai: a – įprasto vaizdo įrašo kadras; b – vaizdo įrašo kadras pritaikius poliarizacinį filtrą; c – etaloninis vaizdo fragmentas

Sudarius vaizdo informacijos rinkinį iš vaizdo įrašo, kurio sudaryme nenaudotas poliarizacinis filtras, pirmajame kadre iškerpamas etaloninis fragmentas, kuris panaudojamas sistemos apmokymo procese. Apdorojimui iš kiekvieno vaizdo įrašo paaimami 22 vaizdo kadrai ir atlikus segmentavimą sudaroma bandymo rezultatų lentelė (6 lentelė).



54 pav. Neatpažintų taškų kiekis

Remiantis sudarytos bandymų rezultatų lentelės (6 lentelė) duomenimis nubrėžus neatpažintų taškų kiekio grafiką pastebima, kad sistemą papildžius poliarizaciniu filtru ženkliai

sumažinamas neteisingai atpažintų taškų kiekis. Pritaikius poliarizacinį filtrą neteisingai atpažintų taškų kiekis sumažėjo 445 taškais arba 15,5%. Nepaisant nemažo neatpažintų taškų kiekio sumažėjimo reikia šiuos rezultatus įvertinti bendrame kontekste. Šiame bandyme buvo tikrinama 275200 vaizdo taškų. Netaikant poliarizacinio filtro iš šių taškų vidutiniškai neatpažinti apie 2858 taškai, kurie sudarė apie 1,04% visų bandyme tikrintų taškų. Tad remiantis apibendrintais duomenimis konkrečiu atveju pritaikius poliarizacinį filtrą neatpažintų taškų kiekis sumažintas tik 0,16% visų taškų kiekio.

6 lentelė

Segmentavimo rezultatų palyginimo lentelė

Vaizdo kadro numeris	Atpažintų taškų kiekis (poliarizacinis)	Atpažintų taškų kiekis (įprastas)	Bendras taškų kiekis	Atpažinta taškų (poliarizacinis), %	Atpažinta taškų (įprastas), %
1	272804	272340	275200	99,13	98,96
2	272804	272340		99,13	98,96
3	272804	272340		99,13	98,96
4	272791	272340		99,12	98,96
5	272791	272343		99,12	98,96
6	272790	272348		99,12	98,96
7	272790	272348		99,12	98,96
8	272790	272339		99,12	98,96
9	272773	272338		99,12	98,96
10	272766	272329		99,12	98,96
11	272762	272337		99,11	98,96
12	272765	272330		99,12	98,96
13	272778	272340		99,12	98,96
14	272783	272343		99,12	98,96
15	272781	272343		99,12	98,96
16	272781	272344		99,12	98,96
17	272796	272344		99,13	98,96
18	272796	272344		99,13	98,96
19	272795	272344		99,13	98,96
20	272795	272344		99,13	98,96
21	272795	272352		99,13	98,95
22	272795	272353		99,13	98,95

Apibendrinant bandymo rezultatus galima teigti, kad esant normaliam apšvietimui ir sistemą papildžius poliarizaciniu filtru sumažinamas neatpažintų taškų kiekis, tačiau bent jau konkrečiu atveju šis pagerėjimas nėra žymus lyginant su bendru tikrintų taškų skaičiumi.

3.8 Poliarizacinių filtrų taikymo įtaka segmentavimo rezultatams kamerą akinant saulei

Atliekant įvairių tekstūrų kelio dangos segmentavimo bandymą pastebėta, kad saulei akinant kamerą (47 pav. e) sumažėjo atpažintų kelio dangos taškų kiekis, tad siekiama išsiaiškinti, kaip šioje situacijoje pasikeis kelio dangos segmentavimo rezultatai sistemą papildžius poliarizaciniu filtru. Kaip ir prieš tai atliktame bandyme sudaromi 2 vaizdo įrašai. Abiem atvejais kamera pastatoma taip, kad būtų gautas akinimo efektas (55 pav. a) ir nekeičiant kameros padėties įrašomi vaizdo įrašai prieš kamerą pastačius poliarizacinį filtrą (55 pav. b) ir be jo (55 pav. a).



55 pav. Bandyme naudojami vaizdo pavyzdžiai: a – akina saulė (įprastas); b – akina saulė (poliarizacinis); c – etaloninis pavyzdys

Iš gautų vaizdo pavyzdžių matyti, kad šiam bandymui sudarytuose vaizdo pavyzdžiuose (55 pav. a) dėl kameros akinimo pastebimas žymiai didesnis kelio dangos spalvos intensyvumo verčių pokytis nei ankstesniame bandyme (47 pav. e kairėje). Teoriškai kameros akinimas gali atsirasti staiga, todėl pradiniam sistemos apmokymui pasirenkamas kelio dangos pavyzdys (55 pav. c), kuriame jos apšvietimas nėra pakitęs. Tokiu būdu bus patikrinta, kaip keisis segmentavimo rezultatai atsiradus staigiam apšvietimo pokyčiui.

Sudarius bandymui reikalingų duomenų rinkinį ir atlikus pradinį sistemos apmokymo procesą, atliekamas abiejų vaizdo įrašų segmentavimo procesas. Kadangi apdorojamuose vaizdo kadruose (55 pav. a, b) yra pašalinių objektų, nuskaitant kiekvieną kadrą programiškai iškerpamas kiek įmanoma didesnis kelio dangos plotas. Galiausiai atliekamas iškirpto kelio

dangos ploto segmentavimo procesas, kurio metu visi fragmento taškai išrūšiuojami ir suskaičiuojamas atpažintų taškų skaičius.

7 lentelė

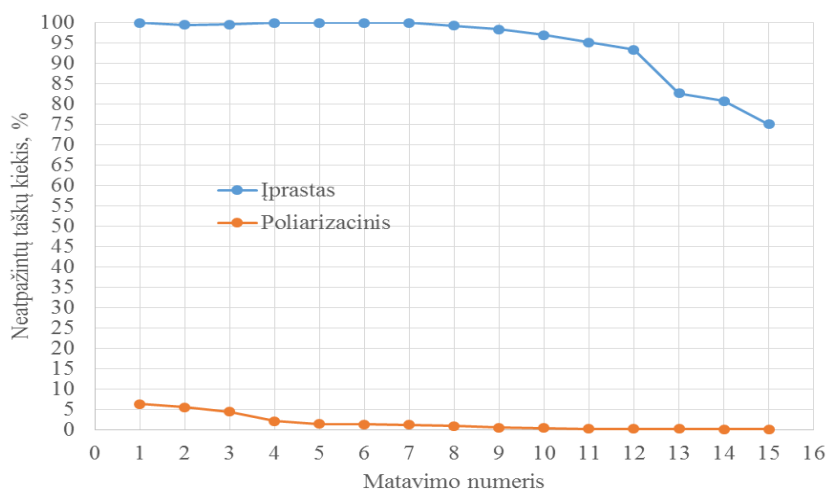
Segmentavimo rezultatų kamerą akinant saulei lentelė

Atpažintų taškų skaičius (poliarizacinis)	Atpažintų taškų skaičius (įprastas)	Bendras taškų skaičius	Atpažintų taškų skaičius (poliarizacinis), %	Atpažintų taškų skaičius (įprastas), %
19280	3	20574	93,7	0,01
19450	107		94,5	0,52
19677	83		95,6	0,4
20149	0		97,9	0
20293	0		98,6	0
20318	0		98,8	0
20340	15		98,9	0,07
20390	141		99,1	0,69
20470	333		99,5	1,62
20503	624		99,7	3,03
20530	1004		99,8	4,88
20547	1370		99,9	6,66
20546	3562		99,9	17,31
20560	3966		99,9	19,28
20557	5127		99,9	24,92

Analizuojant sudarytos lentelės duomenis (7 lentelė) pastebimas didžiulis atpažintų taškų skaičiaus verčių svyravimas netaikant poliarizacinio filtro. Ieškant šių svyravimų dar kartą patikrinami vaizdo pavyzdžiai. Patikrinus kiekvieną vaizdo įrašo kadrą nustatyta, kad kiekviename šiame bandyme naudojamame vaizdo kadre keičiasi kelio dangos intensyvumo vertės, nes palaipsniui mažėja akinantis saulės poveikis. Kitas ryškus rezultatų skirtumas pastebimas nagrinėjant abiem atvejais atpažintų taškų kiekį (56 pav.). Sistemai veikiant be poliarizacinio filtro visų matavimų metu gauti itin prasti segmentavimo rezultatai. Šiame bandyme geriausiu užfiksuotu atveju atpažįstama tik 24,92% viso kelio dangos fragmento taškų (56 pav.) ir praktikoje būtų sunku atpažinti pakankamą kelio dangos plotą.

Tuo tarpu sistemą papildant poliarizaciniu filtru, viso bandymo metai išlaikomas itin aukštas atpažintų taškų kiekis. Prasčiausias bandyme užfiksuotas rezultatas siekė 93,7% viso kelio dangos taškų skaičiaus ir praktikoje būtų nesunku atpažinti pakankamą kelio plotą. Taip pat

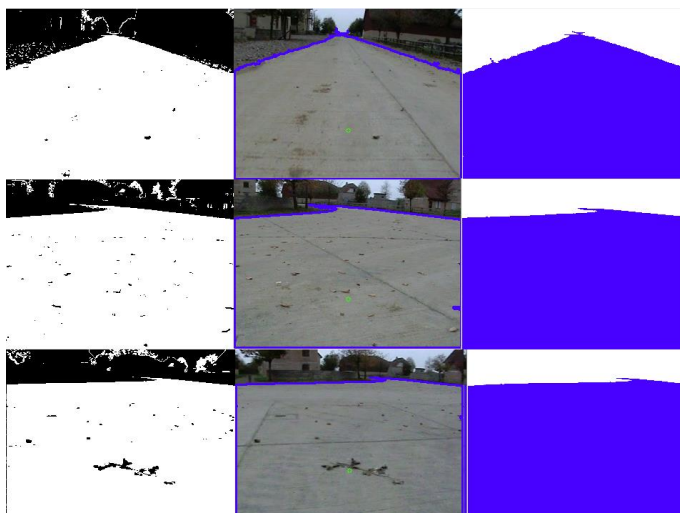
geriausiu atveju atpažintų taškų kiekis priartėja prie 100% ir siekia 99,9% viso bandyme tikrinto kelio dangos fragmento taškų. Galiausiai palyginus abiem atvejais atpažintų taškų kiekius galima teigti, kad kamerą akinant saulei ir sistemą papildžius poliarizaciniais filtrais atpažįstamas bent 4 kartais didesnis kelio dangos plotas.



56 pav. Neatpažintų taškų kiekis procentais

3.9 Segmentavimo rezultatų praktinis taikymas

Taikant sudarytą metodą ir atlikus kelio dangos segmentavimo procesą, išėjime gaunamas kelio dangos ir pašalinių objektų taškų verčių masyvas. Nors vizualiai šie duomenys yra gana aiškūs (52 pav.), tačiau programiškai vis dar neaiški kelio padėtis. Šiam tikslui sudarytas papildomų funkcijų rinkinys, padedantis gauti objekto navigacijai reikalingą informaciją.



57 pav. Segmentavimo rezultatų taikymas: kairėje – dangos segmentavimo rezultatai; viduryje – kelio dangos kraštų koordinatinių vizualizavimas; dešinėje – navigacinė plokštuma

Pasinaudojant sudarytomis papildomomis funkcijomis atliekamas papildomas segmentavimo metu gautos matricos (57 pav. kairėje) apdorojimas. Paprasčiausias navigacijai naudingos informacijos gavimo būdas yra kelio kraštų nuskaitymas (57 pav. viduryje). Pritaikius papildomai sudarytą kelio kraštų koordinacių nustatymo funkciją, gražinamas kelio dangos koordinacių masyvas bei atstumai iki kelio dangos krašto objekto priekyje, kairėje ir dešinėje (nuo taško 47 pav. viduryje). Kitas kiek pažangesnis navigacinės informacijos gavimo metodas yra navigacinės plokštumos sudarymas (57 pav. dešinėje). Sudarius navigacinę plokštumą galima gauti objekto kelio žemėlapi bei apskaičiuoti trumpiausią kelią iki tam tikros žemėlapio vietos išvengiant žemėlapio sudarymo metu užfiksuotų kliūčių ir darant prielaidą, kad kliūtys yra navigacinės plokštumos skylės.

Išbandžius šiuos navigacinės informacijos gavimo būdus (57 pav.) galima teigti, kad kelio dangos segmentavimui pritaikius sudarytą metodą, segmentavimo rezultatus galima nesunkiai pritaikyti kelio dangos kraštų koordinacių nustatymui (57 pav. viduryje) bei navigacinės plokštumos sudarymui (57 pav. dešinėje).

3.10 Skyriaus išvados

1. Daugiausiai taškų atpažįstama, kai etaloninė zona siekia 15,7% nagrinėjamo vaizdo kadro ploto.
2. Segmentavimui taikant sudarytą metodą vaizdo kadro apdorojimas trunka 5,5 karto trumpiau nei taikant literatūroje aprašomą GMM metodą.
3. Tolstant nuo mokymo zonos išlaikoma pakankamai aukšta segmentavimo kokybė (99% atpažintų taškų)
4. Nepastebėta ženkli šešėlių ar prastesnio matomumo įtaka segmentavimo rezultatams.
5. Saulei akinant kamerą sutrinka sistemos veikimas (atpažįstama 0 – 24% taškų).
6. Sistemą papildžius poliarizaciniu filtru ir kamerą akinant saulei išlaikomas pakankamai aukštas atpažįstamų taškų kiekis (atpažįstama 93 – 99 % taškų).
7. Sistemą papildžius poliarizaciniu filtru ir esant įprastam apšvietimui nepastebėti ženkli poliarizacinio filtro įtaka segmentavimo rezultatams (0,16%).
8. Remiantis segmentavimo metu gautais duomenimis galima sudaryti navigacinę plokštumą bei gauti kelio dangos kraštų koordinates.

IŠVADOS

1. Tolstant nuo etaloninės zonos atsiranda dideli kiekiai taškų, kurių spalvos intensyvumo verčių kombinacijos yra artimos etaloninės zonos kombinacijoms.
2. Nustatyta, kad pakankamai geri segmentavimo rezultatai gaunami, kai etaloninės zonos dydis siekia 1,17 – 15,77% apdorojamo vaizdo kadro ploto.
3. Taikant sudarytą metodą nepastebėta ženkli šešėlių bei prastesnio kelio dangos matomumo įtaka segmentavimo rezultatams.
4. Saulei akinant kamerą sutrinka sistemos darbas ir atpažįstama tik 0 – 24% kelio dangos taškų.
5. Esant įprastam apšvietimui sistemą papildžius poliarizaciniais filtrais neteisingai atpažintų taškų kiekis sumažėja 0,16% viso taškų skaičiaus.
6. Saulei akinant kamerą ir sistemą papildžius poliarizaciniais filtrais išvengiama sistemos sutrikimų ir atpažįstama 93 – 99% kelio dangos taškų.
7. Segmentavimui taikant sudarytą metodą vaizdo kadro apdorojimas trunka 5,5 karto trumpiau nei taikant literatūroje aprašomą GMM metodą.
8. Tolstant nuo etaloninės zonos ir segmentavimui taikant sudarytą metodą išlaikomas pakankamas atpažįstamų taškų kiekis, o taikant literatūroje aprašomą GMM metodą kiekviename sekančiame plote šis kiekis sumažėja apie 10%.

LITERATŪROS SĄRAŠAS

1. **Nishimura, J.** „Vehicle Recognition Using Haar – Like Feature and Cascaded Classifier“. IEEE Sensor Journal Vol. 10, 2010, p. 455.
2. **Jiang N., Lu Y., Tang S., Goto S.** „Rapid Face Detection Using a Multi – Mode Cascade and Separate Haar Feature“. International Symposium on Intelligent Signal Processing and Communication Systems. 2010, p. 1 – 5.
3. **Peng Y., Xu M., Jin J., Luo S., Zhao G.** „Cascade – based License Plate Localization with Line Segment Features and Haar – like Features“. International Conference on Image and Graphics, 2011, p. 1024 – 1025.
4. **Sathyadevan S., Balakrishnan A., Raghunath A.** „Identifying Moving Bodies from CCTV Videos Using Machine Learning Techniques“. International Conference on Networks and Soft Computing, 2014, p. 151 – 154.
5. **Ahuja K., Tuli P.** „Object Recognition by Template Matching Using Correlations and Phase Angle Method“. International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue3. 2013, p. 1370.
6. **Tian Y., Li H.** „Road Tracking by Circular Template Matching from High Resolution Remotely Sensed Imagery“. Urban Remote Sensing Joint Event. 2009, p 1 – 5.
7. **Jin H., Miska M., Chung E., Li M., Feng Y.** „Road Feature Extraction from High Resolution Aerial Images Upon Rural Regions Based on Multi – Resolution Image Analysis and Gabor Filter“. Remote Sensing – Advanced Techniques and Platforms. ISBN.978-953-51-0652-4, 2012, p. 404 – 407.
8. Gaboro filtro iliustracija [Interaktyvus]. [Žiūrėta 2015 m. vasaris 15 d.]. Prieiga per internetą: <<http://aurellem.org/cortex/images/gabor-50-50.png>>
9. **Chang H., Wang J., Li J.** „A Self – supervised Road Detection Method Based on Gabor Filter“. Advanced Engineering Forum Vols. 6 –7, Trans Tech Publications, Switzerland. 2012, p. 764.
10. **Satvitha D. K., Rakshit S.** „Gaussian Mixture Model Based Road Signature Classification for Robot Navigation“. Emerging Trends in Robotics and Communication Technologies. 2010, p. 232.
11. **Fan C., Li Z., Ye X., GU W.** „An adaptive method of color road segmentation“. Computer analysis on images and patterns. “ Zhejiang University, Hangzhou, China. ISBN 978-3-540-69556-1. 2005, p. 663.

12. **Mahendru A., Sarkar M.** „Bio – inspired Object Classification using Polarization Imaging“. Šeštoji tarptautinė konferencija ICST. Kolkata, 2012, p. 1 – 6 .
13. Optinių banginių plokščių iliustracija [Interaktyvus]. [Žiūrėta 2014 balandžio 27 d.]. Prieiga per internetą: <http://www.newport.com/Polarization/144921/1033/content.aspx>
14. CPL filtro iliustracija [Interaktyvus]. [Žiūrėta 2014 balandžio 27 d.] Prieiga per internetą: <http://en.wikipedia.org/wiki/File:Circular.Polarization.Circularly.Polarized.Light.Circular.Polarizer.Creating.Left.Handed.Helix.View.svg>
15. Poliarizacinės kameros sandaros iliustracija [Interaktyvus]. [Žiūrėta 2014 balandžio 27 d.]. Prieiga per internetą: <http://www.fluxdata.com/files/FD-1665P%20Diagram.png>
16. Poliarizacinės kameros konstrukcijos iliustracija [Interaktyvus]. [Žiūrėta 2014 balandžio 27 d.]. Prieiga per internetą: http://upload.wikimedia.org/wikipedia/commons/2/28/A_3CCD_imaging_block.jpg
17. Whole beam image splitting system (US Patent US8619368 B2) [Interaktyvus]. [Žiūrėta 2014 balandžio 29 d.]. Prieiga per internetą: <http://www.google.com/patents/US8619368>
18. Liquid crystal based broadband filter for fast polarization imaging (US Patent US20100315567 A1) [Interaktyvus]. [Žiūrėta 2014 balandžio 29 d.]. Prieiga per internetą: <http://www.google.com/patents/US20100315567>
19. **Rankin A., Matthies L.** „Passive sensor evaluation for unmanned ground vehicle mud detection. California Institute of Technology“. USA, 2010, p. 11 – 15
20. **Manimurugan S., Prakash A. S.** „A simple method for removing reflection and distortion from a single image“. International Journal of Engineering and Innovative Technology Volume 2, Issue 3. 2012, p. 287 – 289.
21. Gauso pasiskirstymo iliustracija [Interaktyvus]. [Žiūrėta 2014 balandžio 27 d.]. Prieiga per internetą: <http://www.librow.com/articles/article-9>
22. **Howe K.** „A practical introduction to computer vision with OpenCV“. Willey, Edition 1. ASIN – B00J9C1FKE. 2014, p. 215.
23. **Suarez O.** „OpenCV Essentials“. Packt Publishing. ISBN – 978 – 1783984244. 2014, p. 67.
24. **Hart J.** „Windows System Programming“. Addison – Wesley Publishing, 4 Edition. 2010, p 124.

25. **Lo D.** „Finite element mesh generation“. CRC Press. ISBN – 978 – 0415690485. 2014, p. 25 – 28.
26. **Millington I.** „Artificial Intelligence for games“. CRC Press. 2012, p. 280.
27. **Harvey A.** „Face detection using Haar Cascades“ [Interaktyvus]. [Žiūrėta 2014 balandžio 28 d.]. Prieiga per internetą:<
http://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html>

PRIEDAI

1 PRIEDAS

Bandymuose naudotos segmentavimo programos kodas

```
#include <opencv/highgui.h>
#include <opencv2/opencv.hpp>
#include <opencv/cv.h>
#include "taskai.h"
#include "lsq.h"
#include <fstream>
#include "plotter.h"
#include "dataProc.h"
#include "dataWriter.h"
#include "SCapture.h"
using namespace cv;
using namespace std;
using namespace kompr;
bool rs,bbb,a,ready=false,ps=false,type=false;
Point p1, p2;
Rect roi,roi2;
Mat base, frame,org,temp, save;
taskai ts,tp,tl,tr,trr;
plotter pl;
dataProc proc;
dataWriter dw;
SCapture sc;
lsq ls;
int mode,r,b,g, gr,gn,gx;
char c;
char cc;
ofstream RB,RG,GB,pk,met;
char file[10];
void pokytis(int event, int x, int y, int flags, void* userdata){
switch (event){
case CV_EVENT_LBUTTONDOWN:
if (!ps){
p1.x = x;
p1.y = y;
ps = true;}
break;
case CV_EVENT_LBUTTONUP:
if (ps){
ps = false;
cout << "aaaa" << endl;
p2.x = x;
p2.y = y;
cout << "isskiriamas vaizdas..." << endl;
roi2 = Rect(p1, p2);
temp = Mat(org, roi);}
break;
case CV_EVENT_RBUTTONDOWN:
for (int y = 0; y < temp.rows; y++){
for (int x = 0; x < temp.cols; x++){
Vec3b vals = base.at<Vec3b>(y, x);
trr.setGB(vals.val[1], vals.val[2]);
trr.setRG(vals.val[0], vals.val[1]);
```

```

        trr.setRB(vals.val[0], vals.val[2]);}}
    dw.savePoints("realus", trr);
    break;}}
void ms(int event,int x,int y,int flags, void* userdata){
switch (event){
    case CV_EVENT_LBUTTONDOWN:
        if (!rs &&!ready){
            pl.x = x;
            pl.y = y;
            ready = true;}
        break;
    case CV_EVENT_LBUTTONUP:
        if (!rs && ready){
            p2.x = x;
            p2.y = y;
            cout << "isskiriamas vaizdas22..." << endl;
            roi = Rect(pl, p2);
            base = Mat(frame,roi);
            //namedWindow("iskirpta", CV_WINDOW_AUTOSIZE);
            //imshow("iskirpta", base);
            //waitKey(33);
            cout << "sudaromas masyvas..." << endl;
            for (int y = 0; y < base.rows;y++){
                for (int x = 0; x < base.cols;x++){
                    Vec3b vals = base.at<Vec3b>(y, x);
                    ts.setGB(vals.val[1], vals.val[0]);
                    ts.setRG(vals.val[2], vals.val[1]);
                    ts.setRB(vals.val[2], vals.val[0]);}}
            cout << "Skaiciuojamas lsq.."<<endl;
            ls.calculate2(ts);
            cout <<"Lygtis: " <<"y=" <<ls.pRG[0]<<"*x+" <<ls.pRG[1] << endl;
            cout << "atviz.." << endl;
            //pl.plotRG(ts);
            //pl.plotLine(ls);
            pl.plotS("Ismatuota", true, false, false, false, ts, ls);
            cout << "ispletimas" << endl;
            //proc.expandAuto(ts, ls);
            proc.expandGaussian(ts);
            proc.getData(&tl);
            //tl = ts;
            pl.plotS("RG", true, false, false, false, tl , ls);
            //pl.plotS("GB", false, true, false, false, tl, ls);
            //pl.plotS("RB", false, false, true, false, tl, ls);
            dw.savePoints("duomenys",ts);
            cout << "Tikrinti pagal: " << "a /rb" << " " << "b /rg" <<
            " " << "c /gb" << " " <<"d -all " <<"e test lsq " <<" f -test is failo" <<
            endl;

            cin.clear();
            cin.get(cc);
            if (cc == 'a'){
                mode = 1;}
            if (cc == 'b'){
                mode = 2;}
            if (cc == 'c'){
                mode = 3;}
            if (cc == 'd'){
                mode = 4;}
            if (cc == 'e'){
                mode = 5;}
            if (cc == 'f'){
                mode = 6;
                dw.loadPoints("realus", &tr);}
            mode = 5;
            cout << mode << endl;

```

```

        cout << "pradedamas atvaizdavimas..." << endl;
        rs = true;}

    break;}}
int main(){
cout << "Capture window ? y/n" << endl;
cin.clear();
cin.get(cc);
    if (cc == 'y'){
        type=true;}
    if (cc == 'n'){
        type = false;}
    cin.clear();
CvCapture* capture =
cvCaptureFromFile("C:/Users/MT/Desktop/samples/new/lyginimas/5/polar.mp4");
cvSetCaptureProperty(capture,CV_CAP_PROP_FRAME_WIDTH,800);
cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_HEIGHT, 600);
    if (!capture){
        cout << "nera vaizdo" << endl;
        cin.get();}
namedWindow("Isejimas", CV_WINDOW_AUTOSIZE);
//namedWindow("Iejimas", CV_WINDOW_AUTOSIZE);
//cvSetMouseCallback("Iejimas",pokytis,NULL);
cvSetMouseCallback("Isejimas", ms, NULL);
    if(type){
        frame = sc.getScreen(sc.getWindow());
        cvtColor(frame, frame, CV_RGB2BGR);}
    if(!type){
        for (int i = 0; i < 19; i++){
            frame = cvQueryFrame(capture);}}
//frame = imread("1.jpeg",CV_LOAD_IMAGE_COLOR);
    if (frame.empty()){
        cout << "nera vaizdo" << endl;
        cin.get();}
while (1){
    if (rs){
        if (type){
            frame = sc.getScreen(sc.getWindow());
            cvtColor(frame, frame, CV_RGB2BGR);
            frame.copyTo(org);}
        if(!type){
            frame = cvQueryFrame(capture);
            frame.copyTo(org);}
        for (int y = 0; y < frame.rows;y++){
            cv::Vec3b* pixel = frame.ptr<cv::Vec3b>(y);
            for (int x = 0; x < frame.cols; ++x){
                r = pixel[x][2];
                g = pixel[x][1];
                b = pixel[x][0];
                if (mode == 1){
                    if (ts.getRB(r, b) == 0){
                        frame.at<Vec3b>(y, x)[0] = 255;
                        frame.at<Vec3b>(y, x)[1] = 255;
                        frame.at<Vec3b>(y, x)[2] = 255;}}
                if (mode == 2){
                    if (ts.getRG(r, g) == 0){
                        frame.at<Vec3b>(y, x)[0] = 255;
                        frame.at<Vec3b>(y, x)[1] = 255;
                        frame.at<Vec3b>(y, x)[2] = 255;}}
                if (mode == 3){
                    if (ts.getGB(g, b) == 0){
                        frame.at<Vec3b>(y, x)[0] = 255;
                        frame.at<Vec3b>(y, x)[1] = 255;
                        frame.at<Vec3b>(y, x)[2] = 255;}}
                if (mode == 4){

```

```

        if ((ts.getGB(g, b)>0) && (ts.getRG(r, g)>0) &&
(ts.getRB(r, b)>0))){
        else{
            frame.at<Vec3b>(y, x)[0] = 255;
            frame.at<Vec3b>(y, x)[1] = 255;
            frame.at<Vec3b>(y, x)[2] = 255;}}
        if (mode == 5){
            if (tl.getRG(r, g) < 1 /*|| tl.getRB(r, b)<1 ||
tl.getGB(g, b)<1*/){
                //frame.at<Vec3b>(y, x)[0] = 255;
                //frame.at<Vec3b>(y, x)[1] = 255;
                //frame.at<Vec3b>(y, x)[2] = 255;
                pixel[x][2]=255;
                pixel[x][1]=255;
                pixel[x][0]=255;
                org.at<Vec3b>(y, x)[0] =255;
                org.at<Vec3b>(y, x)[1] =255;
                org.at<Vec3b>(y, x)[2] =255;}}
            if (mode == 6){
                if (tr.getRG(r, g) == 0){
                    frame.at<Vec3b>(y, x)[0] = 255;
                    frame.at<Vec3b>(y, x)[1] = 255;
                    frame.at<Vec3b>(y, x)[2] = 255;}}}}
        // tr.clearRG();
        for (int y = 0; y < temp.rows; y++){
            for (int x = 0; x < temp.cols; x++){
                Vec3b vals = base.at<Vec3b>(y, x);
                tr.setGB(vals.val[1], vals.val[2]);
                tr.setRG(vals.val[0], vals.val[1]);
                tr.setRB(vals.val[0], vals.val[2]);}}
        //pl.plotS("Nuolat atnaujinama", true, false, false, false, tr, ls);
        //rectangle(org, roi2, Scalar(255, 255, 255));
        //hconcat(org, frame, save);
        //sprintf(file, "C:/Users/MT/Desktop/results/testas/p %d.png", sk);
        //imwrite(file, save);
        //sk++;
        imshow("Iejimas", org);
        imshow("Isejimas", frame);
        waitKey(33);}
    else{
        imshow("Isejimas", frame);
        waitKey(33);}}
return 0;}

```

Bandymuose naudotos navigacinės plokštumos programos kodas

```

#include <opencv/highgui.h>
#include <opencv2/opencv.hpp>
#include <opencv/cv.h>
#include "dataPoints.h"
#include "dataProc.h"
#include <fstream>
using namespace cv;
using namespace manoLib;
//variables begin
CvCapture* capture;
Mat frame,base,out,sav,save,saves,savess,org;
dataPoints ts;
dataProc proc ,proc2;
bool rs, bbb, a, ready = false, ps = false, type = false;
Point p1, p2;
Rect roi,roi2;
int sk = 0;
char file[10];
//variables end
void ms(int event, int x, int y, int flags, void* userdata){
    switch (event){
        case CV_EVENT_LBUTTONDOWN:
            if (!rs &&!ready){
                p1.x = x;
                p1.y = y;
                ready = true;}
            break;
        case CV_EVENT_MOUSEMOVE:
            cout << "X: " << x << " y: " << y << endl;
            break;
        case CV_EVENT_LBUTTONUP:
            if (!rs && ready){
                p2.x = x;
                p2.y = y;
                std::cout << "isskiriamas vaizdas22..." << std::endl;
                roi = Rect(p1, p2);
                base = Mat(frame, roi);
                std::cout << "sudaromas masyvas..." << std::endl;
                for (int y = 0; y < base.rows; y++){
                    for (int x = 0; x < base.cols; x++){
                        Vec3b vals = base.at<Vec3b>(y, x);
                        ts.setGB(vals.val[1], vals.val[0]);
                        ts.setRG(vals.val[2], vals.val[1]);
                        ts.setRB(vals.val[2], vals.val[0]);}}
                std::cout << "ispletimas" << std::endl;
                proc.expandGaussian(ts);
                rs = true;}
            break;}}
int main(int argc, char **argv){
capture = cvCaptureFromFile("15.avi");
cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_WIDTH, 800);
cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_HEIGHT, 600);
namedWindow("org", CV_WINDOW_AUTOSIZE);

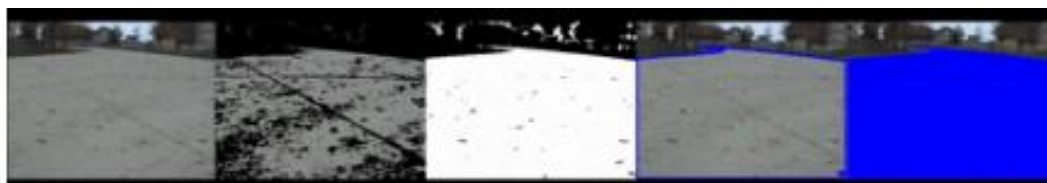
```

```

cvSetMouseCallback("Isejimas", ms, NULL);
rs = true;
frame = cvQueryFrame(capture);
cv::Mat sample = cv::imread("sample.png");
    for (int y = 0; y < sample.rows; y++){
        for (int x = 0; x < sample.cols; x++){
            Vec3b vals = sample.at<Vec3b>(y, x);
            ts.setGB(vals.val[1], vals.val[0]);
            ts.setRG(vals.val[2], vals.val[1]);
            ts.setRB(vals.val[2], vals.val[0]);}}
proc.expandGaussian(ts);
while (1){
    if (rs){
        frame = cvQueryFrame(capture);
        frame.copyTo(org);
        proc.filter(frame);
        imshow("Isejimas", proc.getRaw());
        imshow("org", proc.getMesh());
        waitKey(33);}
    else{
        imshow("Isejimas", frame);
        waitKey(33);}}
return 0;}

```

Pristatyme naudotas animuotas segmentavimo paveikslėlis



Iš kairės į dešinę: pradinis vaizdo kadras, segmentavimo rezultatai taikant GMM, segmentavimo rezultatai taikant sudarytą sistemą, atpažinto kelio kraštų koordinatinių vizualizavimas, remiantis kraštų koordinatėmis sudaryta navigacinė plokštuma.