



VILNIAUS GEDIMINO
TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
ELEKTRONINIŲ SISTEMŲ KATEDRA

Gediminas STANKEVIČIUS

REGA GRĮSTŲ JUDESIO ANALIZĖS ALGORITMŲ TYRIMAS
RESEARCH ON VISION-BASED MOTION ANALYSIS ALGORITHMS

Magistro baigiamasis darbas

Elektronikos studijų programa, valstybinis kodas 621H61003

Kompiuterizuotų elektroninių sistemų specializacija

Elektronikos inžinerijos studijų kryptis

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
ELEKTRONINIŲ SISTEMŲ KATEDRA

TVIRTINU
Katedros vedėjas

(parašas)
Prof. dr. D. Navakauskas
2019 m. _____ mėn. ____ d.

Gediminas STANKEVIČIUS

REGA GRĮSTŲ JUDESIO ANALIZĖS ALGORITMŲ TYRIMAS
RESEARCH ON VISION-BASED MOTION ANALYSIS ALGORITHMS

Magistro baigiamasis darbas

Elektronikos studijų programa, valstybinis kodas 621H61003

Kompiuterizuotų elektroninių sistemų specializacija

Elektronikos inžinerijos studijų kryptis

Vadovas

doc. dr. Dalius Matuzevičius

(pedag. vardas, moksl. laipsnis, vardas, pavardė) _____ (parašas) (data)

Lietuvių kalbos konsultantas

lekt. dr. Anželika Gaidienė

(pedag. vardas, moksl. laipsnis, vardas, pavardė) _____ (parašas) (data)

Vilniaus Gedimino technikos universitetas
Elektronikos fakultetas
Elektroninių sistemų katedra

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

Antrosios pakopos studijų **Elektronikos inžinerijos** programos magistro baigiamasis darbas

Pavadinimas **Rega grįstų judesio analizės algoritmų tyrimas**

Autorius **Gediminas Stankevičius**

Vadovas **Dalius Matuzevičius**

Kalba: lietuvių

Anotacija

Rega grįstų judesio analizės algoritmų tyrimas. Magistro baigiamasis darbas elektronikos inžinerijos laipsniui. Vilniaus Gedimino technikos universitetas. Vilnius, 2019, 70 p., 28 iliustr., 12 lent., 41 bibl., 2 priedai.

Naujausi vaizdų atpažinimo algoritmai leido pasiekti puikių rezultatų sprendžiant įvairius uždavinius. Realiojo laiko uždaviniuose dažnai apdorojamas vaizdų srautas, tačiau, daugelis vaizdų atpažinimo algoritmų rezultatams gauti vis dar naudoja tik pavienius vaizdus ir neišnaudoja galimybės atlikti vaizdų analizę laiko srityje. Baigiamojo darbo tikslas – ištirti rega grįstus judesio analizės algoritmus atskleidžiant galimybes juos taikyti realiojo laiko uždaviniuose. Darbo tikslui pasiekti atlikta judesio analizės algoritmų analitinė apžvalga, atrinkti, įgyvendinti ir eksperimentiškai ištirti perspektyvūs judesio aptikimo ir optinio srauto algoritmai, pateikti tyrimų rezultatai. Darbo metu pasiūlytas daugelio objektų aptikimo ir sekimo vaizde algoritmas, kuriame algoritmui spartinti pritaikytas eksperimentiškai atrinktas optinio srauto algoritmas. Pasiūlytu algoritmu pavyko iki penkių kartų padidinti objektų aptikimo ir sekimo algoritmo spartą, išsaugant didžiąją dalį tikslumo.

Prasminiai žodžiai: judesio analizė, optinis srautas, judesio aptikimas, intelektualiosios sistemos, objektų aptikimas, objektų sekimas.

Vilnius Gediminas Technical University
Faculty of Electronics
Department of Electronic Systems

ISBN ISSN
Copies No.
Date-.....-.....

Master Degree Studies **Electronics Engineering** study programme Master Graduation Thesis

Title **Research on Vision-Based Motion Analysis Algorithms**

Author **Gediminas Stankevičius**

Academic supervisor **Dalius Matuzevičius**

Thesis language: Lithuanian

Annotation

Latest image recognition algorithms allowed to achieve great results in various computer vision tasks. Typically, continuous video stream is available in real-time tasks, however, many image recognition algorithms use only separate frames and do not use the option of video analysis in time domain. The aim of the thesis was to investigate vision-based motion analysis algorithms that could be used in real-time. The aim of the research was achieved by analytical review and experimental study of motion detection and optical flow algorithms. A novel multiple object detection and tracking algorithm was proposed. The aforementioned algorithm showed to have increased the speed of object detection and tracking by up to five times maintaining most of the accuracy.

Master Thesis consists of 70 pages, 28 figures, 12 tables, 41 bibliographical entries, 2 annexes.

Keywords: Motion analysis, optical flow, motion detection, intelligent systems, object detection, object tracking.

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Gediminas Stankevičius, 20133978

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Elektronikos fakultetas

(Fakultetas)

Elektronikos inžinerija, EKSfm-17

(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO (PROJEKTO)

SĄŽININGUMO DEKLARACIJA

2019 m. gegužės 28 d.

Patvirtinu, kad mano baigiamasis darbas tema „Rega grįstų judesio analizės algoritmų tyrimas“ patvirtintas 2017 m. lapkričio 03 d. dekanų potvarkiu Nr. 216el, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Mano darbo vadovas docentas daktaras Dalius Matuzevičius.

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

(Parašas)

Gediminas Stankevičius

(Vardas ir pavardė)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
ELEKTRONINIŲ SISTEMŲ KATEDRA

Technologijos mokslų sritis

Elektros ir elektronikos inžinerijos mokslo kryptis

Elektronikos inžinerijos studijų kryptis

Elektronikos inžinerijos studijų programa, valst. kodas 621H61003

Kompiuterizuotų elektroninių sistemų specializacija

TVIRTINU

Katedros vedėjas

prof. dr. D. Navakas

201_ m. _____ mėn. ____ d.

MAGISTRO BAIGIAMOJO DARBO
UŽDUOTIS

Studentui

Gediminas STANKEVIČIUS

Baigiamojo darbo tema: Rega grįstų judesio analizės algoritmų tyrimas

Research on Vision-Based Motion Analysis Algorithms

Patvirtinta 201_ m. _____ mėn. ____ d. dekanų potvarkiu Nr. _____.

Baigiamojo darbo užbaigimo terminas 201_ m. _____ mėn. ____ d.

Darbo tikslas – ištirti rega grįstus judesio analizės algoritmus atskleidžiant galimybes juos taikyti realaus laiko vaizdų atpažinimo uždaviniuose.

Darbo uždaviniai

1. Atlikti rega grįstų judesio analizės algoritmų analitinę apžvalgą.
2. Atrinkti judesio analizės algoritmus, perspektyvius taikyti vaizdų atpažinimo realiu laiku uždaviniuose.
3. Įgyvendinti ir eksperimentiškai ištirti atrinktus judesio analizės algoritmus specializuotoje elektroninėje aparatūroje, ir pateikti tyrimo rezultatus.

Aiškinamojo rašto turinys

Įvadas

Darbo aktualumas ir tikslas

Darbo uždaviniai

Naudoti tyrimo ir analizės metodai

Darbo naujumas ir praktinė nauda

Darbo struktūra

1. Rega grįstų judesio analizės algoritmų analitinė apžvalga.
2. Atrinkti judesio analizės algoritmai ir jų taikymas vaizdų atpažinimo realiu laiku uždaviniuose.
3. Duomenų paruošimas judesio analizės algoritmų eksperimentiniam tyrimui.

4. Judesio analizės algoritmų tyrimo rezultatai.

Apibendrinimas. Išvados

Literatūra

Priedai

A priedas. Judesio analizės algoritmų eksperimentio tyrimo rezultatai.

B priedas. Pranešimo XXII-oje Lietuvos jaunųjų mokslininkų konferencijoje medžiaga.

Baigiamojo darbo rengimo konsultantas:

–

(pedag. vardas, moksl. laipsnis, vardas, pavardė)

Vadovas

(parašas)

doc. dr. Dalius Matuzevičius

(pedag. vardas, moksl. laipsnis, vardas, pavardė)

Užduotį gavau

(parašas)

201_ m. _____ mėn. ____ d.

TURINYS

Ivadas	7
Darbo aktualumas ir tikslas	7
Darbo uždaviniai	7
Naudoti tyrimo ir analizės metodai	7
Darbo naujumas ir praktinė nauda	8
Darbo struktūra	8
1. Rega grįstų judesio analizės algoritmų analitinė apžvalga	9
1.1. Judesio aptikimas	9
1.1.1. Kadryų skirtumo metodai	9
1.1.2. Fono atimties metodai	10
1.1.3. Judesio aptikimo algoritmų apžvalgos apibendrinimas	13
1.2. Optinis srautas	13
1.2.1. Optinio srauto algoritmų tikslas	14
1.2.2. Optinio srauto vaizdinimas	14
1.2.3. <i>Lucas-Kanade</i> algoritmas	15
1.2.4. <i>Farneback</i> algoritmas	16
1.2.5. <i>DIS</i> algoritmas	16
1.2.6. Optinio srauto skaičiavimas dirbtiniais neuronų tinklais	17
1.2.7. <i>DeepFlow</i> algoritmas	17
1.2.8. <i>FlowNet</i> algoritmas	17
1.2.9. <i>FlowNet2</i> algoritmas	19
1.2.10. <i>LiteFlowNet</i> algoritmas	20
1.3. Judesio požymių taikymas vaizdų atpažinimo uždaviniuose	21
1.3.1. Judesio požymių taikymas žmogaus veiksams atpažinti	22
1.3.2. Judesio požymių taikymas judantiems objektams aptikti ir sekti	24
1.3.3. Judesio požymių taikymas vaizdams segmentuoti	26
1.4. Trumpas skyriaus apibendrinimas	28
2. Atrinkti judesio analizės algoritmai ir jų taikymas vaizdų atpažinimo realiuoju laiku uždaviniuose	29
2.1. Atrinkti judesio analizės algoritmai	29
2.1.1. Pradiniam palyginimui atrinkti optinio srauto algoritmai	29
2.1.2. Eksperimentiniam spartos tyrimui atrinkti judesio aptikimo algoritmai	30
2.1.3. Eksperimentiniam spartos ir tikslumo tyrimui atrinkti optinio srauto algoritmai	30
2.2. Pasiūlytas naujas objektų aptikimo ir sekimo, taikant optinį srautą, algoritmas	31
2.2.1. Pasiūlytas algoritmas	32
2.2.2. Objektų aptikimas vaizde	34
2.2.3. Objektų sekimas vaizde	35
2.3. Trumpas skyriaus apibendrinimas	36
3. Duomenų paruošimas judesio analizės algoritmų eksperimentiniam tyrimui	37
3.1. Duomenų paruošimas pradiniam optinio srauto algoritmų palyginimui	37
3.2. Duomenų paruošimas atrinktų optinio srauto ir judesio aptikimo algoritmų spartai ir optinio srauto algoritmų tikslumui tirti	38
3.3. Duomenų paruošimas pasiūlyto objektų aptikimo ir sekimo, taikant optinį srautą, algoritmui tirti	39
3.4. Objektų aptikimo algoritmo modelio pasirinkimas	42
3.5. Trumpas skyriaus apibendrinimas	42
4. Judesio analizės algoritmų tyrimo rezultatai	43
4.1. Pradinio optinio srauto algoritmų palyginimo rezultatai	43

4.1.1. <i>Farneback</i> algoritmu apskaičiuotas optinis srautas	44
4.1.2. <i>DIS</i> algoritmu apskaičiuotas optinis srautas	45
4.1.3. <i>FlowNet2</i> algoritmu apskaičiuotas optinis srautas.....	46
4.1.4. Skirtingais algoritmais apskaičiuoto optinio srauto palyginimas	47
4.2. Judesio aptikimo ir optinio srauto algoritmų spartos tyrimo rezultatai.....	48
4.3. Optinio srauto algoritmų tikslumo tyrimo rezultatai.....	51
4.4. Pasiūlyto objektų aptikimo ir sekimo algoritmo tyrimo rezultatai.....	52
4.4.1. Objektų sekimo tikslumo priklausomybė nuo objektų aptikimo dažnio	52
4.4.2. Objektų sekimo spartos priklausomybė nuo objektų aptikimo dažnio	58
4.5. Trumpas skyriaus apibendrinimas.....	59
Apibendrinimas. Išvados	60
Literatūra	62
PRIEDAI	65
A priedas. Judesio analizės algoritmų tyrimo rezultatai.....	65
B priedas. Pranešimo XXII-oje Lietuvos jaunųjų mokslininkų konferencijoje medžiaga ...	66

ĮVADAS

Kompiuterizuotos elektroninės sistemos taikomos daugelio šiuolaikinių žmonių kasdienybėje. Augant šių sistemų galimybėms, jos imtos taikyti ne tik kaip žmogaus naudojami įrankiai, bet ir kaip žmogaus asistentai, o kai kur net pakaitas žmogui. Tokiam taikymui kuriamos sistemos, pasižyminčios žmogaus intelektui būdingomis savybėmis, viena jų – vaizdinės informacijos suvokimas ir panaudojimas savo tikslams siekti. Šios sistemos vadinamos intelektualiosiomis, o sritis, tirianti būtent vaizdinės informacijos apdorojimą ir analizę kompiuterizuotose sistemose, – kompiuterine rega (angl. *computer vision*).

Kompiuterinės regos sritis dėl savo perspektyvumo susilaukė daug tyrėjų dėmesio ir jau šiuo metu sukauptos šios srities žinios sėkmingai taikomos komerciškai, o tai leidžia ją toliau plėsti. Šiame darbe tiriami rega grįsti algoritmai, skirti iš vaizdo išgauti regimąjį judesį apibūdinančią informaciją, ir jų taikymo realiojo laiko vaizdų atpažinimo uždaviniuose galimybės.

Darbo aktualumas ir tikslas

Naujausiais vaizdų atpažinimo algoritmais pavyko pasiekti žmogui prilygstantį tikslumą ir tai atvėrė galimybę vaizdų atpažinimo algoritmus taikyti pramonės automatizavimo, autonominio transporto, išmaniojo miesto ir kitose išmaniosiose sistemose. Daugelyje minėtų vaizdų atpažinimo algoritmų taikymo sričių naudojamas realiuoju laiku gaunamas vaizdų srautas, kuriame galima analizuoti ne tik pavienius kadrus, tačiau ir jų kitimo laike savybes. Tik nedidelė dalis šiuolaikinių vaizdų atpažinimo algoritmų išnaudoja šią galimybę.

Šio darbo tikslas – ištirti rega grįstus judesio analizės algoritmus atskleidžiant galimybes juos taikyti realiojo laiko vaizdų atpažinimo uždaviniuose.

Darbo uždaviniai

Baigiamojo darbo metu iškelti trys uždaviniai:

1. Atlikti rega grįstų judesio analizės algoritmų analitinę apžvalgą.
2. Atrinkti judesio analizės algoritmus, perspektyvius taikyti vaizdų atpažinimo realiuoju laiku uždaviniuose.
3. Įgyvendinti ir eksperimentiškai ištirti atrinktus judesio analizės algoritmus specializuotoje elektroninėje aparatūroje, pateikti tyrimo rezultatus.

Naudoti tyrimo ir analizės metodai

Baigiamojo darbo tyrimo metu atlikta su regimojo judesio analize susijusios mokslinės literatūros analizė. Tiriami algoritmai įgyvendinti ir eksperimentiškai ištirti. Tiriamiems algoritmams

įgyvendinti naudotos įterptinės sistemos, kuriose naudojami ARM A57, ARM A53 architektūrų centriniai procesoriniai įrenginiai. Intelektualiaisiais metodais grįstiems algoritmams įgyvendinti naudotas aparatinis skaičiavimų spartinimas ir lygiagretus atlikimas CUDA architektūrą palaikančiuose įrenginiuose.

Eksperimentų programinis kodas rašytas *Python*, C ir C++ kalbomis, naudotas *Matlab* programų paketas. Programiniame kode panaudotos *OpenCV*, *Numpy*, *Scipy* bibliotekos.

Darbo naujumas ir praktinė nauda

Darbo metu tirtos rega grįstų judesio analizės algoritmų įgyvendinimo ir taikymo galimybės sistemose, kurios vykdo vaizdų atpažinimo uždavinius realiuoju laiku. Tyrimo metu įvertintos galimybės išplėsti naujausius vaizdų atpažinimo uždavinius taip, kad būtų išnaudota judesį apibūdinanti informacija. Baigiamojo darbo metu pasiūlytas naujas objektų aptikimo ir sekimo vaizde algoritmas, leidžiantis kelis kartus paspartinti algoritmo veikimą. Pasiiekta algoritmo sparta įgalina pasiūlytą algoritmą taikyti realiuoju laiku. Galimybė vaizdų atpažinimo uždavinius vykdyti realiuoju laiku toje pačioje sistemoje, kurioje yra ir vaizdo signalo šaltinis, leidžia išvengti nereikalingo duomenų perdavimo, padidinti sistemos saugumą ir autonomiškumą. Šios savybės naudingos vaizdų atpažinimo uždavinius taikant praktiškai.

Darbo struktūra

Darbo rezultatai padalinti į keturis etapus, etapų rezultatai pateikti atskiruose skyriuose.

Pirmajame skyriuje pateikti judesio analizės algoritmų analitinės apžvalgos rezultatai. Analitinės apžvalgos metu išanalizuotas judesio aptikimo ir optinio srauto algoritmų veikimas, apžvelgti klasikiniai ir modernūs judesio aptikimo ir optinio srauto algoritmai, pateikti judesio analizės algoritmų taikymo pavyzdžiai judančių objektų aptikimo ir sekimo, veiksmų atpažinimo, vaizdų segmentavimo uždaviniuose.

Antrajame skyriuje pateikti eksperimentiniams tyrimams atrinkti judesio analizės algoritmai, pasiūlytas naujas objektų aptikimo ir sekimo vaizde algoritmas ir paaiškintas jo veikimas.

Trečiajame skyriuje pateikti eksperimentiniams tyrimams pasirinktų duomenų paruošimo rezultatai. Pateikti ir paaiškinti duomenų rinkiniams keliama reikalavimai, algoritmams vertinti naudotos metrikos.

Ketvirtajame skyriuje pateikti ir paaiškinti atliktų eksperimentinių tyrimų rezultatai.

1. REGA GRĮSTŲ JUDESIO ANALIZĖS ALGORITMŲ ANALITINĖ APŽVALGA

Šiame darbe regimuoju judesiu vadinamas vaizdų kitimas laike, pavyzdžiui, vaizdų sraute stebint kameros atžvilgiu judantį objektą, jo padėtis vaizdo ribų atžvilgiu kinta ir analizuojant šį kitimą galima atlikti objekto judesio analizę. Šiame skyriuje aptarti svarbiausi judesio skaitmeniniame vaizdų sraute analizės algoritmų tipai, apžvelgti populiarūs kiekvienos grupės algoritmai ir jų veikimo principai. Taip pat apžvelgti judesio analizės algoritmų taikymo vaizdų atpažinimo uždaviniuose pavyzdžiai.

1.1. Judesio aptikimas

Vienas iš dažniausiai kompiuterinės regos sistemose naudojamų algoritmų tipų – judesio aptikimo algoritmai. Šių algoritmų tikslas – aptikti vaizde judančias dalis ir jas pažymėti. Tipinis judesio aptikimo algoritmo rezultatas – dvejetainis apdorojamo kadro formato vaizdas. Šiame vaizde vienetu pažymėti judantys taškai, o nulių – nejudantys.

Judesio aptikimas atliekamas daugumos vaizdų analizės algoritmų pradžioje, siekiant atmesti informacijos neteikiančią vaizdo dalį, pavyzdžiui, nekintantį foną. Judesio aptikimo uždavinys kartais tapatinamas su fono atimties uždaviniu, nes dažnai naudojamos stacionarios vaizdo kameros, todėl fonas nejuda, o jame esantys objektai juda. Šiame skyriuje aptariant judesio aptikimo algoritmus daroma prielaida, kad vaizdą sudaro stacionarus arba palyginti lėtai kintantis fonas ir judrūs objektai. Šiuo atveju objektų atskyrimo nuo fono ir judesio aptikimo uždaviniai tapatūs ir gali būti apžvelgti kartu.

Šiuo metu vaizdų analizės sritis yra intensyviai vystoma ir nuolatos atsiranda naujų judesio aptikimo algoritmų, todėl šiame skyriuje aptariamais klasikiniai, gerai žinomi judesio aptikimo metodai, kuriais remiasi ir daugelis naujų algoritmų.

1.1.1. Kadro skirtumo metodai

Vienas paprasčiausių būdų aptikti pokytį (judesį) tarp dviejų dvimačio vaizdo kadro ($f(x,y,t_1)$ ir $f(x,y,t_2)$), užfiksuotų atitinkamais laiko momentais t_1 ir t_2 – suskaičiuoti šių kadro taškų matricų skirtumą [1]. Šis metodas įgyvendinamas 1.1 formule.

$$d_{ij}(x,y) = f(x,y,t_i) - f(x,y,t_j), \quad (1.1)$$

čia ir toliau šiame skyriuje $f(x,y,t)$ yra analizuojamo skaitmeninio vaizdo x eilutės, y stulpelio taško intensyvumas kadre, kuris užfiksuotas laiko momentu t .

1.1 formulė atitinka matricų atimtį. Siekiant aiškesnio rezultato galima taikyti skirtumo modulio palyginimą su slenkstine verte θ , o skirtumui viršijus slenkstinę vertę kadrų skirtumas prilyginamas vienetui, kitu atveju – nuliui. Atlikus minėtą operaciją gaunamas kadras, kuriame vienetais pažymėti taškai, kurie lyginamuose kadruose skiriasi daugiau nei nustatyta slenkstinė vertė. Tai leidžia pasirinkti norimą judesio aptikimo algoritmo jautrumą keičiant θ vertę. Taip išplėstą kadrų skirtumo metodą atitinka 1.2 formulė.

$$d_{ij}(x, y) = \begin{cases} 1, & \text{kai } |f(x, y, t_i) - f(x, y, t_j)| > \theta \\ 0, & \text{kai } |f(x, y, t_i) - f(x, y, t_j)| \leq \theta \end{cases} \quad (1.2)$$

Kadrų skirtumo matricioje vienetui lygūs taškai laikomi taškais, kuriuose aptiktas judesys. Šis metodas yra gana paprastas, reikalaujantis santykinai mažai resursų, tačiau jautrus triukšmui. Šio metodo taikymo rezultato pavyzdys pateiktas 1.1 pav.



1.1 pav. Judesio aptikimo kadrų skirtumo metodu pavyzdys [2]

1.1 pav. viršuje pateikti kadrai naudoti skirtumui skaičiuoti, apačioje kairėje pusėje – rezultatas gautas pagal 1.1 formulę, apačioje dešinėje pusėje – rezultatas gautas pagal 1.2 formulę. Siekiant papildomai apdoroti gautą rezultatą galimos įvairios morfologinės operacijos, kurias tinkamai parinkus gaunamos vientisos sritys, žyminčios aptiktą judesį.

Kadrų skirtumo metodas atitinka fono atimties algoritmą, kai fonu dabartiniam kadrai laikomas prieš tai buvęs kadras. Detaliau fono atimties algoritmai aptarti 1.1.2 skirsnyje.

1.1.2. Fono atimties metodai

Fono atimties (angl. *background subtraction*) metodai pagrįsti prielaida, kad stebimą vaizdą sudaro stacionarus arba palyginti lėtai kintantis fonas ir įvairūs objektai, kurie vaizde gali atsirasti ir

judėti. Siekiant vaizde aptikti judančius objektus reikėtų lyginti dabartinį kadrą su sudarytu fono modeliu. Vaizdo sritys, kuriose dabartinis kadras skiriasi nuo fono modelio, laikomos sritimis, kuriose matomi judantys objektai, taigi – aptiktas judesys. Tokį palyginimą galima įgyvendinti skaičiuojant fono modelio $b(x,y,t_i)$ ir dabartinio kadro $f(x,y,t_i)$ atitinkamų taškų skirtumą:

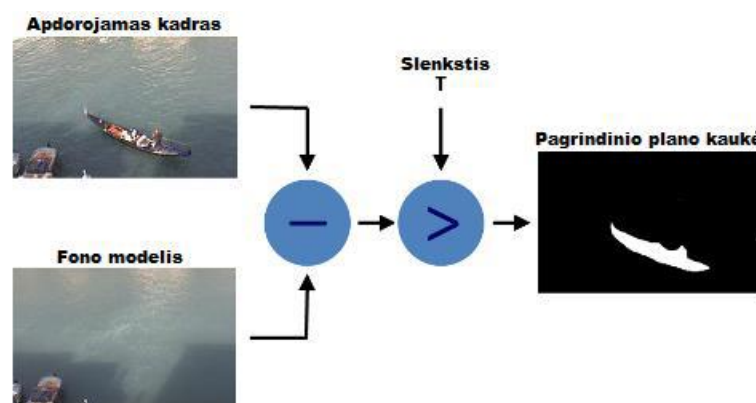
$$d_i(x, y) = f(x, y, t_i) - b(x, y, t_i). \quad (1.3)$$

Kaip ir taikant kadrų skirtumo metodą, šiam skirtumui galime taikyti slenksčio θ funkciją, tai leidžia pasirinkti algoritmo jautrumą ir pritaikyti konkrečiam uždaviniui. Taip pertvarkę 1.3 išraišką gauname 1.4 išraišką.

$$d_i(x, y) = \begin{cases} 1, & \text{kai } |f(x, y, t_i) - b(x, y, t_i)| > \theta \\ 0, & \text{kai } |f(x, y, t_i) - b(x, y, t_i)| \leq \theta \end{cases} \quad (1.4)$$

1.4 išraiška labai panaši į 1.2 išraišką ir gali visiškai sutapti, jeigu fonu laikomas prieš tai buvęs kadras, tačiau šio algoritmo esmė yra fono modelio $b(x,y,t)$ sudarymas. Fono modelį galima sudaryti įvairiais būdais. Paprasčiausias būdas – laikyti pirmąjį kadrą fonu ir visus ateinančius kadrus lyginti su pirmuoju. Toks sprendimas paprastas, tačiau negali prisitaikyti prie kintančio fono, taigi pakitus stebimos aplinkos apšvietimui visas kadras gali būti priskirtas aptikto judesio sričiai. Siekiant sumažinti algoritmo jautrumą nežymiems pokyčiams, pavyzdžiui, triukšmui, galimas tam tikro skaičiaus pirmų kadrų vidurkio skaičiavimas. Kitas būdas sudaryti fono modelį – skaičiuoti tam tikro skaičiaus paskutinių kadrų vidurkį. Toks fono modelis palaipsniui prisitaiko prie pakitusio fono, tačiau reikalauja atlikti daugiau skaičiavimų, nes fono modelis turi būti nuolatos perskaičiuojamas.

Galimi įvairūs būdai sudaryti fono modelį, tačiau tai nekeičia šio metodo esmės ir šį metodą galima apibūdinti 1.2 pav. pateikta schema.



1.2 pav. Fono atimties algoritmo pagal 1.4 išraišką veikimo schema

Kitas galimas fono modelio ir kadro palyginimas grįstas kiekvieno taško verčių statistine analize. Šiame metode kiekvienam vaizdą sudarančiam taškui apskaičiuojama Gauso funkcija, atitinkanti to taško verčių pasirinktame kiekyje buvusių kadrų pasiskirstymą. Kiekvieną Gauso funkciją apibūdina

vidurkis $\mu(x,y)$ ir dispersija $\sigma^2(x,y)$, algoritmo vykdymo pradžioje parametrų priskiriamos tokios vertės:

$$\mu(x,y,t_0) = f(x,y,t_0), \quad (1.5)$$

$$\sigma^2(x,y,t_0) = \alpha, \quad (1.6)$$

čia α – tam tikra pasirenkama vertė (priklauso nuo konkretaus atvejo). Gavus naują kadrą i -tuoju laiko momentu, šie parametrai atnaujinami pagal 1.7 ir 1.8 formules:

$$\mu(x,y,t_i) = \rho f(x,y,t_i) + (1 - \rho) \cdot \mu(x,y,t_{i-1}), \quad (1.7)$$

$$\sigma^2(x,y,t_i) = d^2 \rho + (1 - \rho) \cdot \sigma^2(x,y,t_{i-1}), \quad (1.8)$$

čia ρ – pasirenkamas parametras, apibrėžiantis kadrų, iš kurių sudaromas modelis, kiekį. Šis parametras gali būti interpretuojamas kaip algoritmo prisitaikymo prie naujo fono greitis.

Apdorojamo kadro ir esamo fono modelio skirtumo modulis gaunamas skaičiuojant kadro ir sudaryto fono modelio vidurkių skirtumo modulį pagal:

$$d(x,y,t_i) = |f(x,y,t_i) - \mu(x,y,t_i)|, \quad (1.9)$$

čia $\mu(x,y,t_i)$ – apskaičiuotas fono modelio x eilutės y stulpelio taško vidurkis laiko momentu t_i .

Palyginus šį skirtumą su apskaičiuotu fono modelio standartiniu nuokrypiu galime kiekvienam taškui nustatyti, ar jis priklauso fono modeliui ar ne (aptiktas judesys) pagal:

$$fg(x,y,t_i) = \begin{cases} 1, & \text{kai } \frac{|f(x,y,t_i) - \mu(x,y,t_i)|}{\sigma(x,y,t_i)} > \theta \\ 0, & \text{kai } \frac{|f(x,y,t_i) - \mu(x,y,t_i)|}{\sigma(x,y,t_i)} \leq \theta \end{cases}, \quad (1.10)$$

čia $fg(x,y,t_i)$ – apskaičiuota aptikto judesio kaukė, žyminti, ar apdorojamo kadro x eilutės, y stulpelio taške aptiktas judesys ($fg = 1$) ar ne ($fg = 0$), θ – laisvai parenkama slenkstinė vertė. Parinkus didesnę slenkstinę vertę algoritmas labiau tinka dinamiškam ar triukšmingam fonui, o mažesnė slenkstinė vertė didina algoritmo jautrumą.

Šis algoritmas gana aiškus ir efektyvus, saugomą fono modelį sudaro du parametrai vienam fono modelio taškui, modelio sudarymas ir atnaujinimas ganėtinai paprastas. Pastebėta, kad fono modeliavimas Gauso funkcijomis dažnai nepakankamas naudoti praktiškai (tikrasis taško intensyvumo pasiskirstymas neatitinka Gauso funkcijos). Tai lemia dažnai generuojamą klaidingą judesio aptikimą ir gaunama triukšminga dvejetainė judesio aptikimo (daug pavienių taškų, kurių

vertė lygi 1). Tą galima spręsti mažinant algoritmo jautrumą (mažinant slenkstinę vertę θ), tačiau toks sprendimas nėra optimalus, nes taip didėja klaidingo neigiamo rezultato tikimybė. Šios problemos sprendimus savo tyrimuose nagrinėjo Z. Zivkovič'ius [3], pasiūlęs kiekvieno taško vertės modeliavimą kelių Gauso funkcijų pasvertąja suma, vėliau su bendraautoriais patobulinęs šį algoritmą [4]. Taip pat fonui modeliuoti galima naudoti k artimiausių kaimynų algoritmą [5]. Daugelis šių algoritmų įgyvendinami naudojant kompiuterinės regos uždaviniams spręsti skirtą programinę įrangą ir yra sėkmingai įgyvendinti ir ištirti atliekant šio darbo eksperimentus.

1.1.3. Judesio aptikimo algoritmų apžvalgos apibendrinimas

Šiame poskyryje aptarti vaizdų analize grįstų judesio aptikimo algoritmų bendrieji veikimo principai, išnagrinėti klasikiniai judesio aptikimo algoritmai. Pirmojo tipo algoritmas skaičiuoja gretimų kadrų skirtumą ir šiam skirtumui viršijus slenkstinę vertę taškas laikomas kintančiu (aptiktas judesys). Antrojo tipo algoritmuose kuriamas fono modelis, kuris vėliau naudojamas fono atimčiai iš naujai gautų kadrų. Pastarojo tipo algoritmais pasiekiami geresni rezultatai, tačiau reikia atlikti daugiau skaičiavimų.

Šių algoritmų rezultatas – dvejetainis vaizdas, kuriame vieneto vertė priskiriama taškams, kuriuose aptiktas judesys, o nulis – stacionariems taškams. Toks vaizdas vadinamas pagrindinio plano kauke (angl. *foreground mask*). Šį rezultatą galima papildomai apdoroti siekiant išgryninti konkretų rezultatą, pavyzdžiui, atmetant pavienius taškus, kuriems priskirta vieneto reikšmė, tačiau nesiriboja su kitais tokiais taškais ir nesudaro vientisų judesio sričių.

Šio algoritmo taikymo pavyzdys – automatizuota vaizdo stebėjimo sistema, pradedanti vaizdo įrašymą tik tuomet, kai vaizde aptiktas judesys. Ši sistema taupyti vietą įrašams saugoti naudojamame kaupiklyje, be to, vėlesnė įrašų analizė reikalautų mažiau laiko. Taip pat rega grįsti judesio aptikimo algoritmai naudojami objektų atpažinimo ir kituose kompiuterinės regos uždaviniuose, siekiant atrinkti dominančias sritis ir išvengti nereikalingų skaičiavimų.

1.2. Optinis srautas

Judesio aptikimas leidžia vaizde atskirti kintančias sritis nuo nekintančių, tačiau judesio aptikimo algoritmų rezultatai neteikia detalesnės informacijos apie judesį, pavyzdžiui, judėjimo greitį ir judėjimo kryptį. Uždaviniams, kuriuose reikia nustatyti ne tik judančias sritis, bet ir jose esančių objektų judėjimo greitį ir kryptį ar trajektoriją, naudojami optinio srauto (angl. *optical flow*) algoritmai. Šiame skyriuje aptariami bendrieji optinio srauto algoritmų principai ir apžvelgti keli nauji ir perspektyvūs optinio srauto algoritmai.

1.2.1. Optinio srauto algoritmų tikslas

Optinio srauto algoritmo tikslas – analizuojant nuoseklią vaizdų srauto kadrų seką surasti vaizdo taškų padėties pokytį tarp gretimų kadrų. Šie metodai remiasi prielaida, kad judančio taško ryškumas nekinta tarp kadrų (pastovus laike). Taigi tikslas – rasti ir δ_y vertes, kurios minimizuotų tokią išraišką:

$$\varepsilon(\delta_x, \delta_y, t_i) = f(x, y, t_i) - f(x + \delta_x, y + \delta_y, t_{i+1}), \quad (1.11)$$

čia $\varepsilon(\delta_x, \delta_y, t_i)$ – skirtumas tarp pirmojo kadro taško (x, y, t_i) ir šio taško naujame kadre, pajudėjusio per judesio vektorius (δ_x, δ_y) .

Tuomet apskaičiuotas vektorius (δ_x, δ_y) yra taško, kurio koordinatės yra x ir y , judesio vektorius laiko momentu t_i . Remiantis prielaida, kad taško intensyvumas nekinta tarp kadrų, 1.11 išraiška turi būti lygi nuliui, arba galime užrašyti taip:

$$f(x, y, t_i) = f(x + dx, y + dy, t + dt), \quad (1.12)$$

Išskleidus 1.12 išraiškos dešinę pusę Teiloro eilutės aproksimacija, suprastinus ir padalinus iš dt gauname:

$$f_x u + f_y v + f_t = 0, \quad (1.13)$$

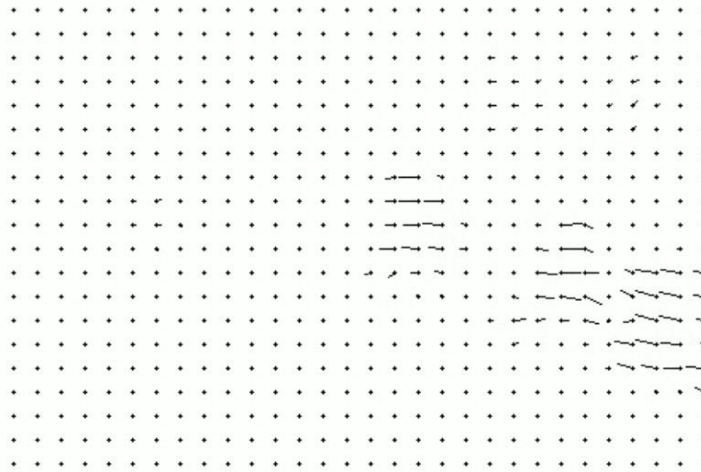
čia $f_x = \frac{df}{dx}$; $f_y = \frac{df}{dy}$; $u = \frac{dx}{dt}$; $v = \frac{dy}{dt}$.

Ši išraiška vadinama optinio srauto lygtimi. Joje iš turimų vaizdų galime nustatyti f_x , f_y ir f_t , tačiau u ir v yra nežinomi. Taigi turime vieną lygtį ir du nežinomuosius, todėl negalime šios lygties išspręsti. Yra pasiūlytas ne vienas šios problemos sprendimo būdas, vienas jų – Lucas-Kanade [6], jis aptartas 1.2.3 šio darbo skirsnyje.

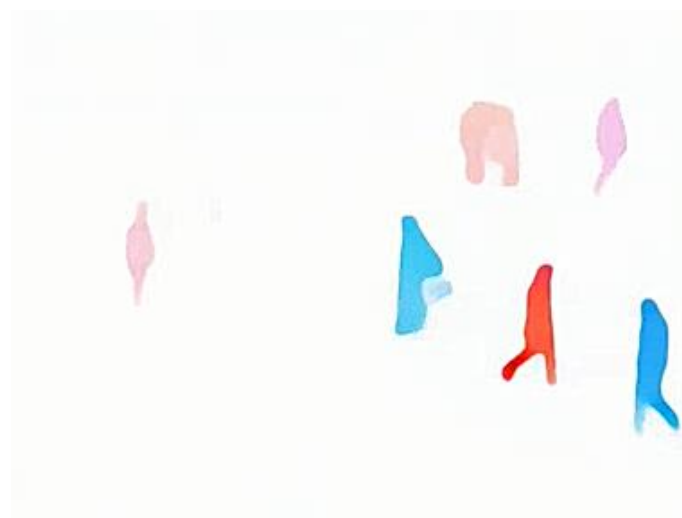
1.2.2. Optinio srauto vaizdinimas

Atlikus optinio srauto skaičiavimus tarp dviejų gretimų vaizdų srauto kadrų, gaunami apskaičiuoti dvimačiai kiekvieno analizuojamo taško judesio tarp šių kadrų vektoriai. Šie vektoriai gali būti apibūdinti judesio x ir y kryptimis dedamosiomis arba ilgiu ir krypties kampų. Abiem atvejais kiekvieną judesio vektorius apibūdina du skaičiai, tačiau, norint vaizdžiai pateikti apskaičiuotą optinį srautą, dažnai naudojami šie du rezultatų vaizdinimo būdai:

- Judesio vektorius vaizdinimas rodyklėmis, kur rodyklių ilgiai proporcingi vektorius ilgiams, tokio vaizdinimo būdo pavyzdys pateiktas 1.3 pav.
- Judesio vektorius vaizdinimas spalvomis taikant HSV spalvų erdvę, kur spalva apibūdinama spalviniu tonu, sodrumu ir intensyvumu, tokio vaizdinimo būdo pavyzdys pateiktas 1.4 pav.



1.3 pav. Optinio srauto vektorių vaizdinimas rodyklėmis



1.4 pav. Optinio srauto vektorių vaizdinimas spalvomis

Šiuose pavyzdžiuose vaizdinamas tos pačios kadrų poros optinis srautas (nejudančiame fone matomi skirtingomis kryptimis einantys žmonės). Iš rezultatų matyti, kad optinį srautą vaizdinant antruoju būdu geriau matomi judantys objektai, kurių judėjimo greitis nedidelis ir galimas aukštesnės raiškos vaizdinimas, todėl šio darbo metu naudotas antrasis optinio srauto vaizdinimo būdas.

1.2.3. *Lucas-Kanade* algoritmas

Lucas-Kanade metodas [6] remiasi prielaida, kad gretimi vaizdo taškai juda panašiai. *Lucas-Kanade* metode imamas, pavyzdžiui, 3×3 taškų vaizdo segmentas ir tariama, kad visi segmento taškai juda kartu. Tokiu atveju galime visiems taškams apskaičiuoti f_x , f_y ir f_t , tuomet gaunamos devynios lygtys su dviem nežinomaisiais. Tokią lygčių sistemą galime išspręsti. Optimizuota lygčių sistemos sprendimo išraiška pateikta (1.14).

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i f_{x_i}^2 & \sum_i f_{x_i} f_{y_i} \\ \sum_i f_{x_i} f_{y_i} & \sum_i f_{y_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i f_{x_i} f_{t_i} \\ -\sum_i f_{y_i} f_{t_i} \end{bmatrix}, \quad (1.14)$$

čia simbolių reikšmės tokios pat, kaip ankstesnėse išraiškose.

Šia išraiška įgyvendintas sudarytos lygčių sistemos sprendimas pagal mažiausių kvadratų kriterijų. Lyginant su kitais algoritmais, šis yra greitas, tačiau gaunamos paklaidos ties judančio objekto ribomis. Dar vienas gerai žinomas šios lygties sprendimo metodas – *Horn-Schunck*, aprašytas autoriaus straipsnyje [7].

1.2.4. *Farneback* algoritmas

Farneback optinio srauto algoritmas [8] paremtas autorių pasiūlyta polinominio išplėtimo transformacija (angl. *polynomial expansion transform*), kurios rezultatas – kiekviena pasirinkto dydžio vaizdą sudarančių taškų kaiminystė aproksimuojama kvadratinio polinomu. Lyginant gretimų kadro aproksimacijas kvadratiniais polinomaais galima mažiausių kvadratų ar kitais metodais rasti tokį analizuojamos vaizdo taškų kaiminystės poslinkį, kuris minimizuotų skirtumą tarp pirmojo kadro polinominio išplėtimo su pritaikytu poslinkiu ir antrojo kadro polinominio išplėtimo. Detaliai šis algoritmas ir polinominio išplėtimo transformacija aprašyta autorių straipsnyje [8].

Šis algoritmas sėkmingai taikomas praktikoje, dėl savo spartos jau gali būti skaičiuojamas realiuoju laiku asmeniniu kompiuteriu, įterptine sistema ar šiuolaikiniu išmaniuoju telefonu. Šis algoritmas įgyvendintas populiariuose kompiuterinei regai naudojamuose programiniuose paketuose, pavyzdžiui, *MATLAB* ar *OpenCV*.

1.2.5. *DIS* algoritmas

DIS (angl. *Dense Inverse Search*) algoritmas [9] taip pat paremtas vaizdo taškų kaimynystės poslinkio, kuris tiksliausiai apibūdintų erdvinį pokytį tarp pirmojo ir antrojo kadro, paieška. Ieškoma tokio poslinkio analizuojamam pirmojo kadro segmentui, kuriuo iškraipius pirmąjį kadrą būtų minimizuota jo ir antrojo kadro, atitinkamo segmento taškų verčių skirtumo kvadratų suma. Šiam minimizavimo uždaviniui autoriai naudoja gradientinio nusileidimo (angl. *gradient descent*) metodą. *DIS* optinio srauto algoritmo autoriai ypatingą dėmesį skyrė algoritmo vykdymo spartai, kad algoritmas galėtų būti taikomas net įterptinėse sistemose. Šis algoritmas yra vienas moderniausių šiame darbe apžvelgtų optinio srauto algoritmų. Detaliai šis algoritmas aprašytas autorių straipsnyje [9].

Šio darbo metu *DIS* algoritmas įgyvendintas įterptinėse sistemose, ištirtas ir palygintas su kitais apžvelgtais algoritmais.

1.2.6. Optinio srauto skaičiavimas dirbtiniais neuronų tinklais

Pastaruoju metu stebima intelektualųjų metodų srities sėkmė lėmė gausius šios srities tyrimus. Išskirtinius rezultatus intelektualiaisiais metodais pavyko pasiekti ir kompiuterinės regos srityje. Dėl šios priežasties šiuo metu jau pasiūlytas ne vienas optinio srauto algoritmas, kuriam pritaikyti daugiasluoksniai dirbtinių neuronų tinklai.

Šiame darbe apžvelgti trys didesnio dėmesio sulaukę ir gerus rezultatus parodę optinio srauto algoritmai, kuriuose naudojami daugiasluoksniai dirbtinių neuronų tinklai. Visuose apžvelgtuose algoritmuose naudojami gilieji sąsūkos dirbtinių neuronų tinklai, į kurių įėjimą siunčiama apdorojama vaizdo kadrų pora, o išėjime apskaičiuojamas optinis srautas tarp šių dviejų kadrų. Šie algoritmai aptarti 1.2.7–1.2.10 skirsniuose.

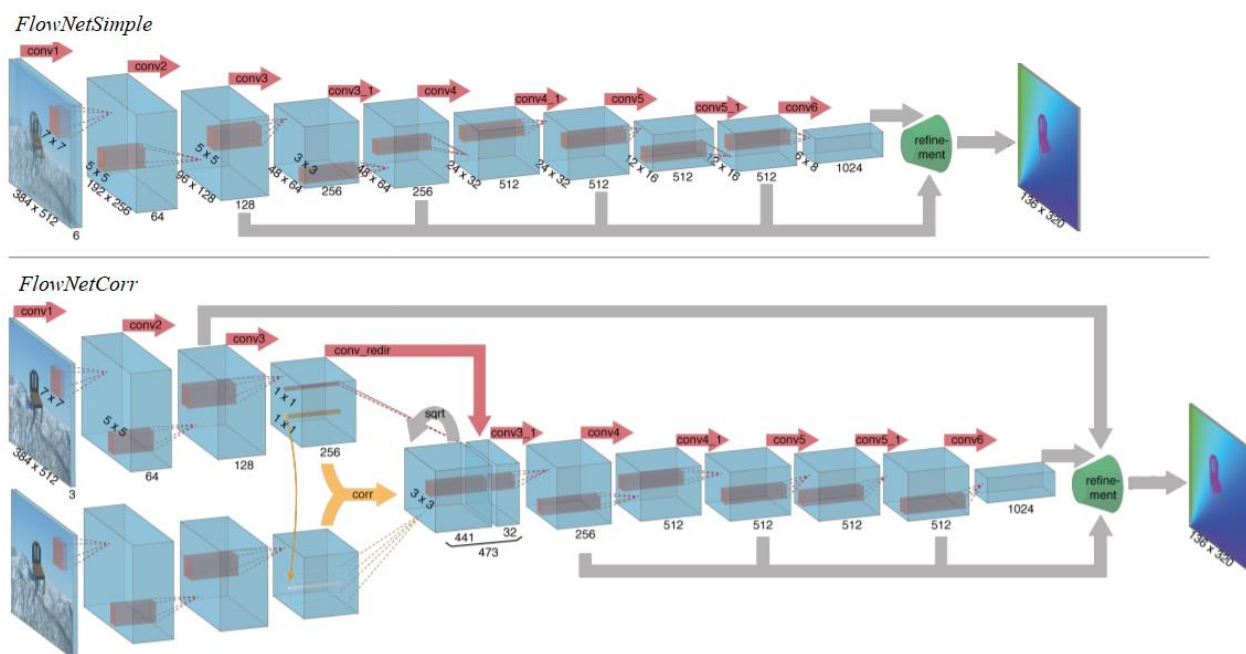
1.2.7. *DeepFlow* algoritmas

DeepFlow optinio srauto algoritmas [10] yra vienas iš ankstyvų bandymų taikyti sąsūkos dirbtinių neuronų tinklus optiniam srautui skaičiuoti. Šį algoritmą daugelis vėliau išleistų algoritmų autorių naudoja kaip atskaitos tašką savo rezultatams įvertinti. Šio algoritmo idėja – taikyti sąsūkos neuronų tinklus, į kurių įėjimą paduodami du gretimi vaizdų srauto kadrai, tarp kurių norimas apskaičiuoti optinis srautas. Šie kadrai apdorojami dvimačiais sąsūkos filtrais, o šių filtrų atsako signalai – kito sluoksnio sąsūkos filtrais. Tokiu principu konstruojama daugiasluoksnė tinklo struktūra. Tarp sluoksnių gali būti naudojamas papildomas signalų apdorojimas. Tokio daugiasluoksnio sąsūkos neuronų tinklo parametrams (sąsūkos filtrų koeficientams) skaičiuoti naudotas tinklo mokymas iš teisingų pavyzdžių (angl. *supervised learning*).

Šio algoritmo autoriams pavyko apmokyti sudarytą neuronų tinklą ir su keliais bandymo duomenų rinkiniais pademonstruoti stabilius rezultatus bei ypač aukštą tikslumą esant dideliems poslinkiams tarp kadrų. Tačiau, apmokytas tinklas sudėtingas ir optinio srauto skaičiavimas užtrunka gana ilgai, autorių bandymų metu dviejų kadrų apdorojimas, naudojant aparatinį spartinimą grafiniais procesoriais, truko apie 20 sekundžių. Tai yra viena iš problemų, kurias mėgino spręsti kitų apžvelgtų algoritmų autoriai.

1.2.8. *FlowNet* algoritmas

FlowNet [11] autoriai taip pat optiniam srautui skaičiuoti pasiūlė taikyti sąsūkos dirbtinių neuronų tinklus ir juos mokyti naudojant turimą optinio srauto duomenų bazę (šiuos duomenų rinkinius dažniausiai sudaro susimuliuoti 3D grafikos vaizdų srautai ir jų optiniai srautai). Šio algoritmo autoriai eksperimentavo su dvejomis skirtingomis tinklo architektūromis, jos pateiktos 1.5 pav.

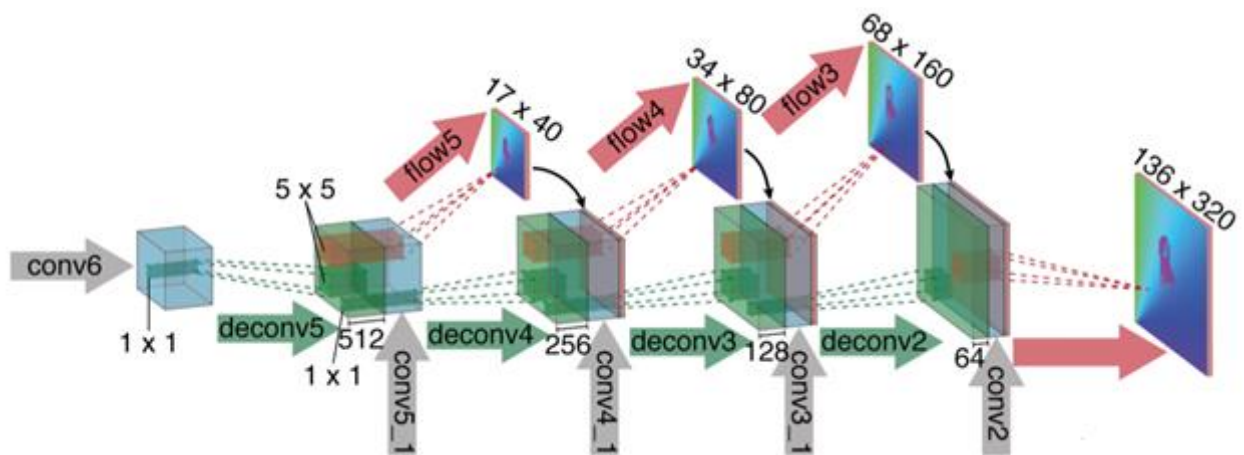


1.5 pav. *FlowNetSimple* ir *FlowNetCorr* struktūros [11]

Iš pateiktų diagramų matyti, kad *FlowNetSimple* tinkle įėjime du trijų kanalų įėjimo vaizdai sudėti į šešių kanalų įėjimą, kuris filtruojamas 7×7 sąsūkos filtrais, gautas rezultatas filtruojamas kito sluoksnio filtrais ir taip tęsiame iki paskutiniojo sluoksnio, kurio išėjimas plečiamas atstatant artimesnį pradiniam vaizdui formatą. *FlowNetCorr* architektūra kiekvieną iš dviejų nagrinėjamų kadrų atskirai paduoda į vienodus sąsūkos tinklus, šiais tinklais išgaunami požymiai, kuriuos vertina papildomas sluoksnis. Suminiai rezultatai paduodami tolimesnei tinklo daliai apdoroti.

Pagal šio straipsnio rezultatus nustatyta, kad sudėtingesnė *FlowNetCorr* struktūra nepasiteisino ir nepagerino tinklo rezultatų, tačiau prideda papildomų skaičiavimų. Ši išvada paneigta vėlesniuose autorių tyrimuose, o tai patvirtina intelektualųjų metodų taikymo kompleksiskumą.

Abiejų architektūrų išėjime naudojami *refinement* sluoksniai, kurių tikslas performatuoti požymių erdvėje esantį tinklo išėjimą į artimesnį pradiniam vaizdui, šiam procesui naudojama struktūra pateikta 1.6 pav.



1.6 pav. FlowNet tinklo refinement etapo struktūra [11]

Iš 1.6 pav. matyti, kad tinkle rezultatui grąžinti artimesnį pradiniam formatui naudojami sluoksniai, kurių išėjimo raiška didesnė nei įėjimo, tokiu būdu optinis srautas įgauna apdorojamų vaizdų matmenis.

1.2.9. FlowNet2 algoritmas

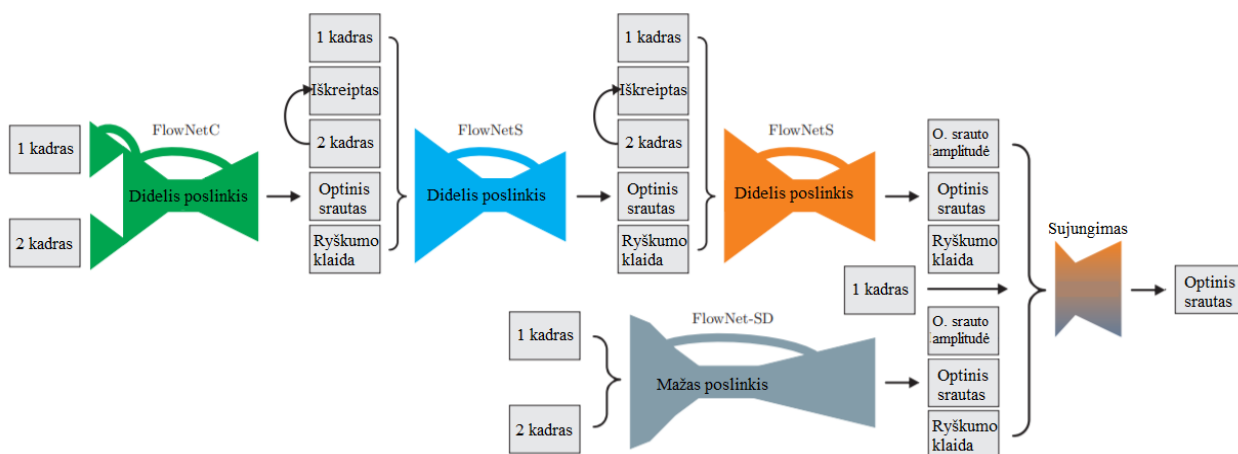
FlowNet2 algoritmas [12] remiasi anksčiau aptarto FlowNet algoritmo [11] rezultatais, tačiau pritaikant naujas idėjas pavyko žymiai pagerinti anksčiau pasiektus rezultatus. Tęsdami tyrimus, pradėtus kuriant FlowNet, autoriai atliko tokius pastebėjimus:

- Tinklo rezultatams labai svarbus mokymo duomenų rinkinys.
- Dinamiškai keičiamas mokymosi žingsnio parametras gali padėti pasiekti geresnių rezultatų,
- anksčiau nepasiteisinusi FlowNetCorr architektūra rodo geresnius rezultatus nei FlowNetSimple (priešingai nei teigta [11]).
- Naudojant tam tikrą FlowNetSimple ir FlowNetCorr kombinaciją galima pasiekti geresnių rezultatų.

Svarbiausias pakeitimas nuo FlowNet yra sudėtinė struktūra, kurioje taikoma anksčiau minėtų architektūrų tinklų kombinacija, tokios architektūros pavyzdys pateiktas 1.7 pav.

Šioje struktūroje be nuosekliai sujungtų FlowNetS ir FlowNetC pridėtas dar ir specializuotas, mažiems poslinkiams skirtas tinklas. Šių tinklų rezultatai apdorojami ir gaunamas bendras optinio srauto rezultatas.

Minėti pakeitimai lėmė daug geresnius optinio srauto skaičiavimo tikslumo rezultatus atlikus bandymus su populiariausiais optinio srauto tikslumui matuoti naudojamais duomenų rinkiniais. FlowNet2 rezultatai įtikino šios srities tyrėjus, kad sąsūkos dirbtinių neuronų tinklų taikymas optiniam srautui skaičiuoti yra perspektyvus, nes pasiektas išskirtinis tikslumas ir tai pritraukė platesnio tyrėjų rato dėmesį, to pavyzdys yra 1.2.10 skirsnyje aptartas LiteFlowNet algoritmas.

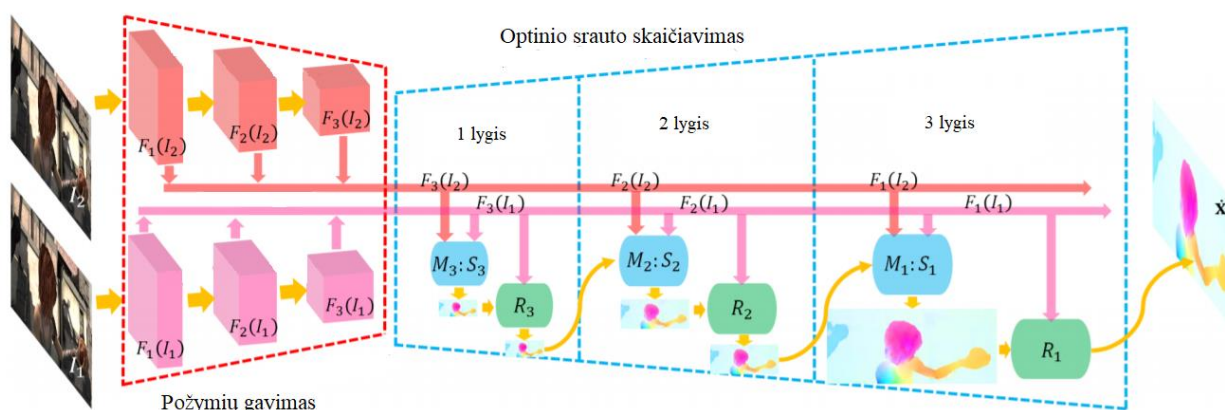


1.7 pav. *FlowNet2* tinklo struktūra [12]

1.2.10. *LiteFlowNet* algoritmas

LiteFlowNet [13] yra vienas iš naujausių ir sėkmingiausių savo kategorijoje optinio srauto algoritmų, šis algoritmas paremtas *FlowNet* ir *FlowNet2* sėkme ir sukaupta patirtimi. Lyginant *FlowNet* ir *FlowNet2* algoritmus pristatančiuose straipsniuose pateiktus rezultatus, matomas žymus tikslumo padidėjimas, tačiau tai pareikalavo kur kas didesnės tinklo struktūros (apie 160 milijonų svorių), o tai mažina šio algoritmo taikymo galimybę ribotų resursų aparatuose.

Šią problemą adresavo *LiteFlowNet* algoritmo autoriai ir savo darbu siekė sukurti žymiai paprastesnį optiniam srautui skaičiuoti skirtą dirbtinių neuronų tinklą, kuris tikslumu nenusileistų ankstesnėms *FlowNet* versijoms. *LiteFlowNet* autoriams išsikeltą uždavinį pavyko įvykdyti ir tą atspindi jų darbo rezultatas – sukurtas net 30 kartų mažesnis (30 kartų mažiau svorio koeficientų) ir 1,36 karto spartesnis algoritmas. Šio algoritmo struktūra pateikta 1.8 pav.



1.8 pav. *LiteFlowNet* tinklo struktūra [13]

Iš neuronų tinklo struktūros matyti, kad ir šio tinklo įėjimą sudaro du vienodi sąsūkos dirbtinių neuronų tinklai (apibraukti raudona punktyrine linija kairėje pusėje), kurių sąsūkos filtrai vienodi, tai leidžia iš apdorojamų gretimų vaizdų srauto kadrų išskirti tuos pačius požymius, o tarp šių požymių skirtinguose kadruose atsiradusį erdvinį poslinkį įvertina likusi tinklo dalis (apibraukta mėlyna punktyrine linija dešinėje pusėje). Erdvinis poslinkis tarp apdorojamų kadrų vertinamas skirtinguose požymių lygmenyse, o tai laidžia įvertinti skirtingų amplitudžių ir skirtingo dydžio sričių poslinkį. Trijose pakopose įvertintas optinis srautas apdorojamas ir apjungiamas į vieningą rezultatą, kurio aukštis ir plotis atitinka įėjimo kadrų atitinkamus matmenis.

Ši neuronų tinklo architektūra leido apmokyti tinklą taip, kad juo apskaičiuotas optinis srautas pagal populiariausius optinio srauto tikslumo vertinimo duomenų rinkinius prilygsta *FlowNet2* tinklui, o tam tikrais atvejais net jį pranoksta.

LiteFlowNet puikiai įrodo, kad sąsūkos dirbtinių neuronų tinklai gali būti sėkmingai pritaikyti optinio srauto skaičiavimo uždaviniui spręsti ir svarbiausia, kad šio tipo algoritmus galima optimizuoti neaukojant tikslumo. Atsižvelgiant į spartų specializuotos ir į dirbtinių neuronų tinklų taikymą orientuotos aparatinės įrangos tobulėjimą, galima tikėtis dar geresnių tokio tipo optinio srauto algoritmų, kurių sparta bus pakankama taikyti realiojo laiko uždaviniuose.

1.3. Judesio požymių taikymas vaizdų atpažinimo uždaviniuose

Iki šiol šiame skyriuje aptarti algoritmai skirti gana paprastoms judesio vaizde savybėms išgauti. Aptarti judesio aptikimo algoritmai leidžia vieno vaizdą sudarančio taško tikslumu įvertinti ar stebimame taške matomas kitimas, kuris vertinamas kaip judesys. Optinio srauto algoritmai leidžia apie regimąjį apdorojamo vaizdų srauto judesį pasakyti daugiau – nustatyti kiekvieno vaizdo taško erdvinio poslinkio tarp apdorojamų kadrų amplitudę ir kryptį. Ši informacija vertinga ir išsamiai apibūdina regimąjį judesį, tačiau vien šių duomenų nepakanka praktiniams judesio analizės algoritmų taikymams ir šie duomenys dažniausiai papildomai apdorojami siekiant išgauti konkrečiam taikymo atvejui svarbią informaciją. Tokio apdorojimo pavyzdys – judesio aptikimas vaizdo stebėjimo sistemoje su automatinio įrašymo aptikus judesį. Šiuo atveju judesio aptikimo arba optinio srauto algoritmo rezultatas gali būti papildomai apdorojamas, pavyzdžiui, judančių taškų kiekis palyginamas su nustatyta slenkstine verte, ją viršijus skelbiamas judesio aptikimo signalas. Tačiau tai tik labai paprastas judesio vaizde požymių panaudos atvejis, o realiuose uždaviniuose sutinkami daug sudėtingesni judesio analizės algoritmų pritaikymo būdai.

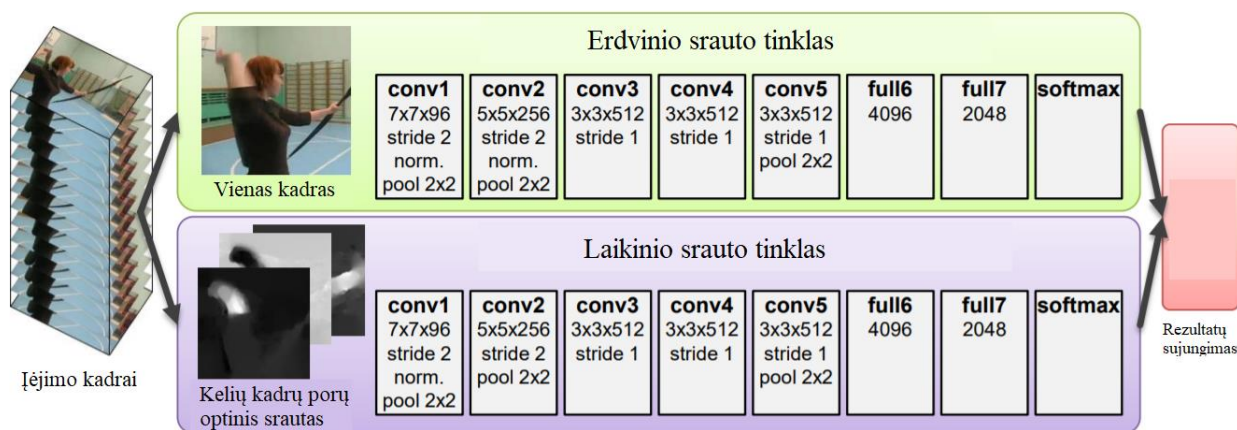
Šiame poskyryje apžvelgti sėkmingo judesio analizės algoritmų taikymo svarbiuose kompiuterinės regos uždaviniuose pavyzdžiai. Rastų taikymo pavyzdžių apžvalga suskirstyta pagal uždavinį, kuriam taikomi judesio analizės algoritmais išgauti duomenys, ir pateikta 1.3.1–1.3.3 skirsniuose.

1.3.1. Judesio požymių taikymas žmogaus veiksmams atpažinti

Pastaruoju metu įvairiuose vaizdų atpažinimo uždaviniuose pavyko pasiekti ypač gerų rezultatų. Nesant griežtų aparatinės įrangos našumo apribojimų, galima pasiekti labai aukštą vaizdų atpažinimo tikslumą, kartais viršijantį žmogaus vaizdų atpažinimo tikslumą. Verta pastebėti, kad didžioji dalis žmonių naudojamos vaizdinės informacijos saugoma vaizdo įrašų pavidalu. Šiuolaikiniams vaizdo įrašams būdinga aukšta skiriamoji geba ir kadrų dažnis, taigi, ir šiuos įrašus ar gyvus vaizdų srautus apdorojančiai įrangai tenka didesnis krūvis. Tai apsunkina vaizdų atpažinimo uždavinius. Tačiau šis iššūkis suteikia ir naujų galimybių vaizdų atpažinimo srityse. Iš vaizdų srauto galima išgauti judesio informaciją, kuri leidžia geriau ir tiksliau pažinti vaizde esančią informaciją. Viena tokios informacijos panaudojimo būdų – žmogaus veiksmų atpažinimas vaizdų srautuose.

Žmogaus veiksmų atpažinimas – gana sudėtingas vaizdų atpažinimo uždavinys, nors kai kuriuos veiksmus nesunku atpažinti iš pavienių kadrų, tačiau yra ir sudėtingesnių pavyzdžių, į kuriuos net ir žmogui tenka kurį laiką pasižiūrėti, kad suprastų ką stebimas žmogus daro. Siekiant įvertinti tokį vaizdų kitimą laike gali būti naudojami šiame darbe tiriami judesio analizės algoritmai.

Šią idėją savo tyrimuose įvertino Karen'as Simonyan'as ir Andrew Zisserman'as [14], jie veiksmams atpažinti pasiūlė naudoti dviejų srautų sąsūkos dirbtinių neuronų tinklą, kurį sudaro du atskiri neuronų tinklai. Juose vaizdo ir optinio srauto duomenys apdorojami lygiagrečiai, atskiruose tinklo kanaluose. Šio dirbtinių neuronų tinklo architektūra pateikta 1.9 pav.



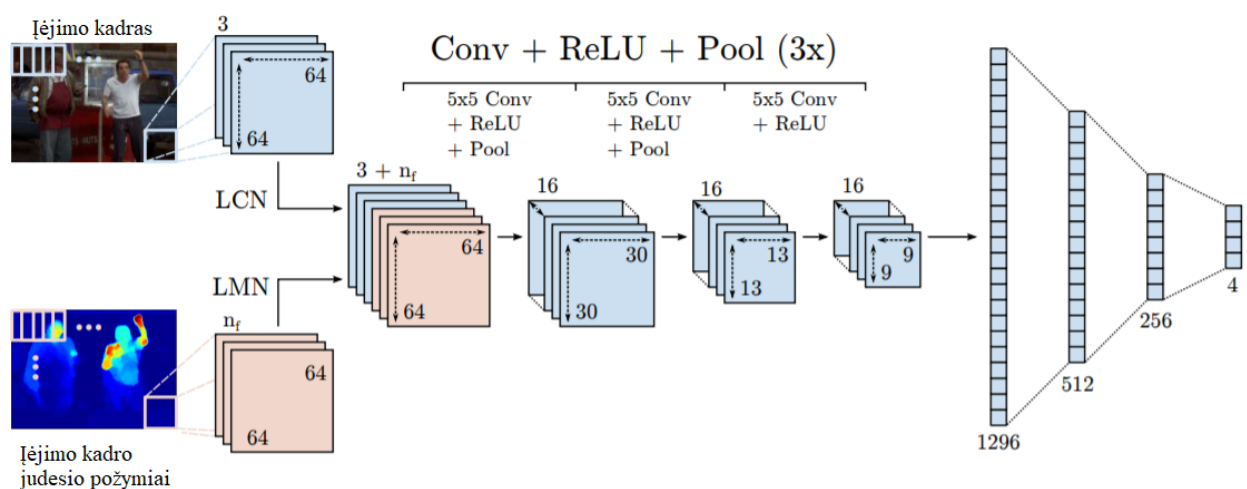
1.9 pav. Dviejų srautų dirbtinių neuronų tinklo veiksmams atpažinti architektūra [14]

Iš aukščiau pateiktos dirbtinių neuronų tinklo architektūros matyti, kad erdvinis (vaizdo) srautas ir laikinis (optinis) srautas apdorojami vienodos architektūros tinklais, kuriuos sudaro penki sąsūkos filtrų sluoksniai, du pilnai sujungti dirbtinių neuronų sluoksniai ir minkštojo maksimumo sluoksnį tinklo gale. Šių dviejų tinklų rezultatus autoriai apjungia dviem būdais – skaičiuojant vidurkį ir apmokant atraminių vektorių (angl. *support vector machine*) klasifikatorių.

Žmogaus veiksmams atpažinti judesio požymius (optinio srauto duomenis) pritaikė ir Charles'as Han'as, Evelyn Mei ir Chao Wang'as [15], pasiūlę suderinti objektų atpažinimo algoritmą su žmonių

veiksmų atpažinimo užduotimi ir iš apdorojamų vaizdų išskiria tik tas sritis, kuriose aptiktas žmogus, judesio požymiai gaunami išskiriant kelių kadrų duomenis iš šių sričių, kurie apdorojami dirbtinio neuronų tinklo, kuris iš kadrų sekos išmoksta išskirti žmonių veiksmus atskirti padedančius požymius. Šio algoritmo autoriai pabrėžia, kad naudojama judesio informacija leido žymiai padidinti veiksmų atpažinimo tikslumą.

Dar vienas optinio srauto naudojimo žmogaus veiksmams atpažinti pavyzdys – *MoDeep* algoritmas [16]. Šio algoritmo autoriai žmogaus pozicijai atpažinti irgi pritaikė sąsūkos dirbtinių neuronų tinklą, tačiau šis tinklas skiriasi tuo, kad jame judesio ir vaizdo informacija sudedama į vieną masyvą ir siunčiama į tinklo įėjimą (slenkančio lango principu). Tinklo architektūra pateikta 1.10 pav.



1.10 pav. *MoDeep* algoritmo dirbtinių neuronų tinklo architektūra [16]

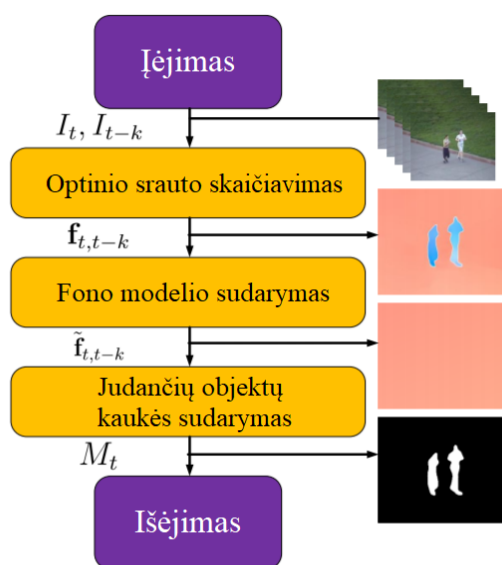
Tinklą sudaro trys pasikartojantys sąsūkos filtrų, aktyvavimo ir didžiausios reikšmės išrinkimo sluoksnių deriniai, o jų išėjimas patenka į keturių sluoksnių pilnai sujungtų dirbtinių neuronų tinklą. Algoritmo pristatančio straipsnio išvadose autoriai pabrėžia, kad judesio požymių naudojimas leido jų pasiūlytam algoritmui pasiekti aukštesnį tikslumą nei kitais tuo metu geriausiais laikytais algoritmais.

Svarbu pažymėti, kad autoriai pabrėžė, kad optinio srauto skaičiavimas yra gana daug skaičiavimo resursų reikalaujantis procesas, todėl bandymams optinis srautas eksperimentų duomenims buvo apskaičiuotas iš anksto, o ne realiuoju laiku. Tai rodo, kad net ir optiniam srautui skaičiuoti netaikant tokių didelių sąsūkos dirbtinių neuronų tinklų, kaip *FlowNet2*, šis vaizdų apdorojimo algoritmas yra sudėtingas ir reikalaujantis laiko. Dėl šios priežasties šio darbo metu tiriamų optinio srauto algoritmų spartai skirtas ypatingas dėmesys.

1.3.2. Judesio požymių taikymas judantiems objektams aptikti ir sekti

Dar vienas svarbus ir daug tyrėjų bendruomenės dėmesio sulaukiantis kompiuterinės regos uždavinys – judančių objektų vaizde aptikimas ir išskyrimas iš fono. Šiame darbe jau aptarti judesio aptikimo algoritmai, kuriuose modeliuojamas vaizdo fonas ir lyginamas su apdorojamu kadru, aptikus skirtumų – šie požymiai ir atitinkamai atnaujinamas sudarytas fono modelis. Tokie metodai judesiui aptikti naudoja tik vaizdinę informaciją, tačiau tokį rezultatą galima pasiekti ir naudojant optinį srautą. Optinio srauto algoritmo rezultatas, dėl savo struktūros panašumo į skaitmeninius vaizdo signalus, gali būti apdorojamas įprastais vaizdų apdorojimo metodais. Paprasčiausiu atveju – galima judesio vektorių amplitudei pritaikyti slenkščio funkciją ir išskirti vaizdo sritis, kuriose regimas slenkstinę vertę viršijantis judesys. Tai duotų panašų rezultatą, kaip ir judesio aptikimo algoritmai, kuriuose naudojamas gretimų kadrų skirtumo metodas, tačiau judantiems objektams aptikti naudojant optinį srautą galima detalesnę judesio analizę. Išrinkus vaizde sritis, kuriose pastebėtas pasirinktos amplitudės judesys – galima šių sričių papildoma analizė, pavyzdžiui, pagal judesio kryptį. Tai leistų vaizde išskirti atskirus objektus pagal jų judėjimo kryptį, o ypač padėtų tada, kai objektai dalinai vienas kitą uždengia ir judesio aptikimo algoritmais būtų sujungti į vieną sritį. Tokioje situacijoje optinis srautas leistų šioje srityje išskirti atskirus objektus, kai jų judėjimas nėra vienodas.

Panašią idėją savo tyrimuose įgyvendino Junjie Huang, Wei Zou, Zheng Zhu ir Jiagang Zhu [17], kurie įgyvendino optinio srauto duomenų naudojimu paremtą judančių objektų aptikimo algoritmą. Šio algoritmo veikimo diagrama pateikta 1.11 pav.



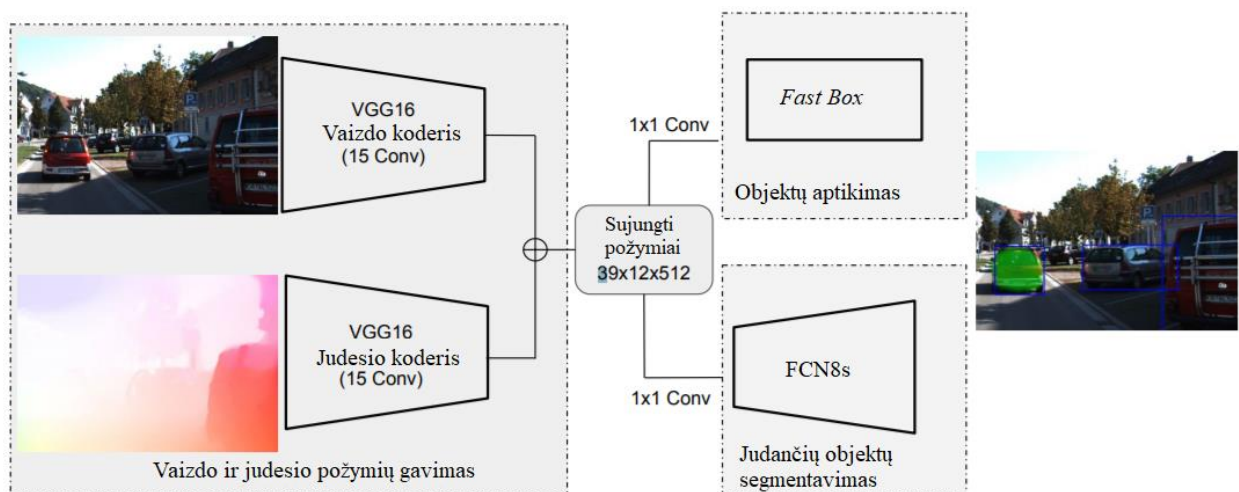
1.11 pav. Optinio srauto duomenų analize paremto judančių objektų aptikimo algoritmo architektūra [17]

Šio algoritmo vaizdų apdorojimo seka prasideda optinio srauto tarp kadrų poros skaičiavimu. Apskaičiuotas optinis srautas naudojamas apskaičiuoti homografinę matricą. Pagal apskaičiuotą

matricą sudaromas optinio srauto fono modelis, kuris, kaip ir anksčiau apžvelgtuose algoritmuose, lyginamas su apskaičiuotu optiniu srautu ir išskiriamos aptikto judesio sritys.

Optiniam srautui skaičiuoti straipsnio autoriai naudojo jau aptartą *FlowNet2* algoritmą ir pabrėžė šio algoritmo tikslumą. Panaudotas tikslus optinio srauto algoritmas ir sėkmingai įgyvendintos jo apdorojimo idėjos leido šio judančių objektų aptikimo algoritmo autoriams pasiekti išskirtinai aukštą judančių objektų aptikimo tikslumą ir, pagal autorių pateikiamus rezultatus, pagerinti daugelio geriausių iki šiol pasiūlytų tokio tipo algoritmų tikslumą.

Objektų aptikimo ir jų klasifikavimo į judančius ir nejudančius uždavinius į vieną apjungė *MODNet* [18] algoritmo autoriai, kurie šiam uždaviniui pritaikė jau anksčiau matytą ir šiame darbe minėtą dviejų srautų sąsūkos dirbtinių neuronų tinklo architektūrą, kurioje lygiagrečiai apdorojami vaizdo signalo ir apskaičiuoto optinio srauto duomenys, o šių atskirų dirbtinių neuronų tinklo šakų apskaičiuotus požymius analizuoja dvi išėjimo tinklo šakos. Pirmoji skirta judančioms sritims išskirti, o antroji – aptikti objektams. Šio algoritmo rezultatas – aptikti objektai ir nustatyta ar kiekvienas iš jų juda ar ne. Šio tinklo struktūra, įėjimo ir išėjimo signalų pavyzdžiai pateikti 1.12 pav.



1.12 pav. *MODNet* algoritmo sąsūkos dirbtinių neuronų tinklo struktūrą, įėjimo ir išėjimo signalų pavyzdžiai [18]

Kaip matyti iš aukščiau pateiktos struktūros, šiame algoritme optinio srauto ir vaizdo duomenys apdorojami atskirais sąsūkos dirbtinių neuronų tinklais. Išgauti judesio ir vaizdo požymiai analizuojami siekiant išgauti apdorojamame kadre aptiktus objektus ir jų klasę (judantis ar nejudantis). Tokio rezultato vizualizacija pateikta 1.12 pav. dešinėje pusėje esančiame paveikslėlyje, kuriame mėlynais stačiakampiais pažymėti aptikti objektai, o žalia spalva pažymėtas judantis automobilis.

MODNet autoriai šį algoritmą orientavo į naudojimą autonominėse transporto priemonėse, todėl tinklui mokytis ir tinklo patikrai naudotas autonominio automobilio veikimo sąlygas atitinkantis vaizdo duomenų rinkinys. Savo tyrimų rezultatais autoriai įrodo, kad ir šiame optinio srauto taikymo

pavyzdyje judesio požymiai teikia didelę naudą ir padidina algoritmo tikslumą, tačiau *MODNet* autoriai pabrėžia svarbų šio algoritmo privalumą – gebėjimą aptikti iki šiol tinklui nežinomų (mokymo duomenų rinkinyje nesančių) klasių objektus vien iš judesio požymių. Ši savybė prideda šiam algoritmui patikimumo esant nenumatytiems atvejams, o taikant intelektualiąsias sistemas realiomis sąlygomis tokie atvejai neišvengiami, todėl ši savybė svarbi norint įgyvendinti autonominį transportą.

Kaip matyti iš apžvelgtų straipsnių, rega grįsta judesio analizė naudinga norint aptikti judančius objektus ir tobulinant objektų atpažinimo ir lokalizavimo uždavinius, tačiau judesio informacija gali padėti ir sprendžiant ilgalaikį aptiktų objektų sekimą. Objektų sekimas leidžia nustatyti stebimų objektų judėjimo trajektoriją ir gyvendinti aukštesnio lygio vaizdų analizės sistemas, pavyzdžiui, miesto stebėjimo sistemose įgyvendinti miesto infrastruktūros užimtumo analizę, stebėti eismo intensyvumą ir įvairias kitas sistemas.

Vienas iš pavyzdžių, kur optinis srautas naudojamas daugelio objektų sekimo algoritmams tobulinti – D. Beaupre ir N. Saunier'io straipsnis [19], kuriame autoriai tyrė optinio srauto naudojimo fono atmetimo algoritmuose įtaką daugelio objektų sekimo algoritmų tikslumui ir pastebėjo, kad pritaikius optinio srauto duomenis, sumažėjo objektų sekimo klaidų kiekis situacijose, kai du sekami objektai yra arti vienas kito arba dalinai uždengiami.

Optinio srauto naudą judantiems objektams sekti savo straipsnyje taip pat įvertino K. Kale, S. Pawar'as ir P. Dhulekar'as [20] ir kitame tyrime S. Hua, M. Kapoor'is ir D. Anastasiu [21], kurie tyrė judančių objektų sekimo miestų stebėjimo sistemose algoritmus.

Iš sėkmingų optinio srauto algoritmo taikymo judantiems objektams aptikti ir sekti pavyzdžių gausos akivaizdu, kad šioje srityje optinio srauto algoritmai yra perspektyvūs ir leidžia toliau tobulinti jau sukurtas vaizdų analizės sistemas ir išplėsti šių sistemų veikimą pritaikius ne tik analizuojamo vaizdų srauto vaizdines, bet ir judesio savybes.

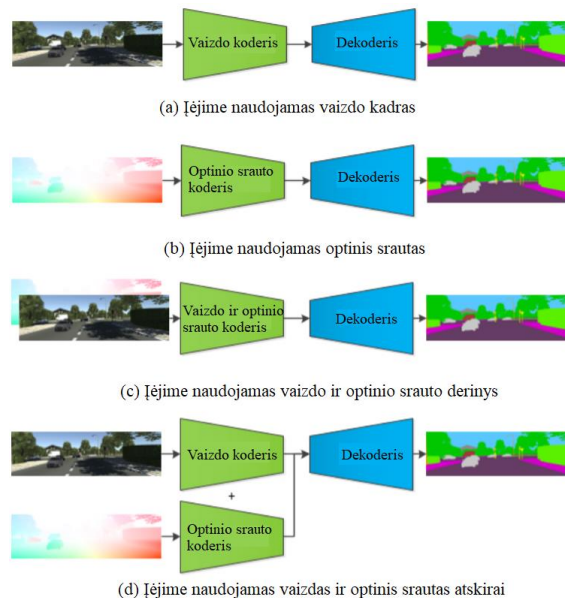
1.3.3. Judesio požymių taikymas vaizdams segmentuoti

Vaizdų segmentavimas – sudėtingas vaizdų analizės uždavinys, kurio tikslas vieno vaizdą sudarančio taško tikslumu padalinti analizuojamą vaizdą į sritis, kurias sudaro atskiri objektai. Pavyzdžiui, autonominio automobilio kamera stebimame vaizde norime atskirti ir pažymėti kelią, kelkraščius, kitus automobilius, pėsčiuosius, dangų, kitas atskiras sritis ir nustatyti kuriai sričiai priklauso kiekvienas vaizdo taškas.

Neseniai pasiekta giliųjų dirbtinių neuronų tinklų sėkmė įvairiose srityse lėmė jų taikymą daugelyje sričių, ne išimtis ir vaizdų segmentavimas, tačiau naujausioje mokslinėje literatūroje randama pavyzdžių, kur vaizdams segmentuoti naudojami ir judesio požymiai. Šis uždavinys ypatingai aktualus autonominiam kelių transportui vystyti, be to, autonominiai automobiliai įprastai

apdoroja nuolatinį vaizdų srautą, o tai leidžia apskaičiuoti optinį srautą ir apjungti šiuos uždavinius siekiant pagerinti vaizdų segmentavimo rezultatą.

H. Rashed'as, S. Yogammani'is, A. El-Sallab'as, P. Križek'as ir M. El-Helw'as savo straipsnyje [22] pateikė tyrimo rezultatus, kuriame įvertino skirtingus optinio srauto duomenų apjungimo su vaizdo duomenimis sąsūkos dirbtinių neuronų tinklo įėjime būdus. Tirtus optinio srauto panaudojimo būdus iliustruoja 1.13 pav.



1.13 pav. Optinio srauto taikymo vaizdams segmentuoti būdai, įvertinti straipsnyje [22]

Iš iliustracijos matyti, kad tyrimai atlikti vaizdams segmentuoti naudojant vien vaizdo informaciją, vien optinį srautą, jų derinį viename sąsūkos dirbtinių neuronų tinkle ir jų analizę atskirais sąsūkos dirbtinių neuronų tinklais, išskirtus požymius vėliau apjungiant prieš perduodant į vaizdų segmentavimo tinklo dalį. Iš šių keturių būdų aukščiausią tikslumą pasiekė paskutinis būdas, kuriame vaizdo ir optinio srauto požymiai išgaunami atskirais sąsūkos filtrais, o tai patvirtina kitų mokslininkų tyrimuose padarytas išvadas, kad tokia tinklo struktūra geriausiai tinka judesio ir vaizdo požymiams išgauti ir derinti.

Dar didesnį optinio srauto panaudojimo vaizdams segmentuoti dirbtinių neuronų tinklo įėjime variacijų kiekį įvertino P. Viswanath'as su bendraautorais [23], kurie optinio srauto skirtingas reprezentacijas pridėjo prie spalvoto vaizdo kaip papildomus kanalus. Viso išbandytos tokios variacijos: tinklo įėjime naudojamas vien vaizdas, prie vaizdo pridėtas normuotų optinio srauto vektorių amplitudžių kanalas, prie vaizdo pridedami normuotų optinio srauto amplitudžių ir kampo kanalai, prie vaizdo pridedami trys spalvinės optinio srauto reprezentacijos kanalai (paaiškinta 1.2.2 skirsnyje), prie vaizdo pridedami optinio srauto x ir y kryptimis dedamųjų kanalai, prie vaizdo pridedamas optinio srauto amplitudės kanalas. Iš visų įvardintų būdų geriausius rezultatus autorių atliktame tyrime pasiekė vaizdo ir optinio srauto amplitudžių kanalo derinys, tačiau šio algoritmo

autoriai nenaudojo dviejų srautų dirbtinių neuronų tinklo architektūros, kurios efektyvumą pabrėžė kitų šiame darbe apžvelgtų straipsnių autoriai.

1.4. Trumpas skyriaus apibendrinimas

Šiame skyriuje apžvelgti dažniausiai sutinkami rega grįsti judesio analizės algoritmai, išanalizuoti jų veikimo principai. Apžvelgti algoritmai suskirstyti į judesio aptikimo ir optinio srauto algoritmų kategorijas. Prie pirmosios pridėti ir fono atmetimo algoritmai, nes šie daugeliu atvejų sutampa.

Judesio aptikimo algoritmais apdorojant vaizdų srautą sudarančią kadrų seką vaizde išskiriamos sritys, kuriose regimas vaizdo kitimas (kinta vaizdą sudarančių taškų intensyvumo vertės). Yra žinomas ne vienas judesio aptikimo algoritmo įgyvendinimo būdas. Šie būdai skiriasi savo tikslumu, jautrumu triukšmui, apšvietimo pokyčiams ir reikalauja atlikti skirtingą kiekį skaičiavimų.

Kita šiame skyriuje aptartų regimojo judesio analizės algoritmų grupė – optinio srauto algoritmai, kurių tikslas iš dviejų vaizdų sekos kadrų nustatyti erdvinį vaizdo poslinkį tarp šių kadrų. Optinis srautas gana išsamiai apibūdina regimąjį judesį nes nustatoma vaizdo komponentų judėjimo kryptis ir poslinkio atstumas, kuris įvertinamas vaizdą sudarančiais taškais. Yra žinoma įvairių optinio srauto algoritmų, kai kurie naudojami jau daugiau nei trisdešimt metų, tačiau nuolat atsiranda naujų, naujus metodus taikančių optinio srauto algoritmų, kuriais dažniausiai siekiama padidinti tikslumą ir spartą. Po pastaruoju metu stebimos giliųjų (daugelio sluoksnių) dirbtinių neuronų tinklų padarytos pažangos įvairiose srityse, atsirado ne vienas bandymas juos taikyti ir optiniam srautui skaičiuoti, šie bandymai susilaukė sėkmės ir pritraukė dar didesnę tyrėjų bendruomenės dėmesį, todėl ši tyrimų sritis išlieka aktuali.

Trečiame poskyryje apžvelgtos judesio analizės algoritmais gautos informacijos apie judesį vaizde taikymo įvairiuose kompiuterinės regos ir vaizdų atpažinimo uždaviniuose galimybės. Iš atliktos apžvalgos rezultatų matoma nenuginčijama judesio analizės algoritmų nauda intelektualiosiose vaizdų atpažinimo sistemose, tokiuose taikymuose, kaip intelektualiosios miesto stebėjimo sistemos ar autonominis kelių transportas.

Iš apžvalgos rezultatų išryškėjo optinio srauto algoritmų naudingumas, jų spartos ir tikslumo svarba, taikymo galimybės. Šis darbas orientuotas į rega grįstų judesio analizės algoritmų taikymą realiojo laiko vaizdų atpažinimo uždaviniuose, kurie vis dažniau vykdomi įterptinėse sistemose, todėl šiame darbe bandyta atskleisti judesio analizės algoritmų taikymo tokiose elektroninėse sistemose galimybes.

2. ATRINKTI JUDESIO ANALIZĖS ALGORITMAI IR JŲ TAIKYMAS VAIZDŲ ATPAŽINIMO REALIUOJU LAIKU UŽDAVINIUOSE

Pirmajame skyriuje pateikta apžvalga atskleisti dažniausiai naudojamų rega grįstų judesio analizės algoritmų veikimo principai, jų galimybės, būdingos savybės ir galimos naujos taikymo sritys. Šiame skyriuje pateikti eksperimentiniam tyrimui parinkti judesio analizės algoritmai, pasiūlytas ir paaiškintas naujas objektų aptikimo ir sekimo algoritmas, kuriame pritaikytas pasirinktas judesio analizės algoritmas.

Sudarant šio darbo eksperimentinių tyrimų planą atsižvelgta į tiriamiems judesio analizės algoritmams būdingą didelį atliekamų skaičiavimų kiekį, todėl nuspręsta atlikti pasirinktų algoritmų spartos tyrimą. Be to, planuojant šio darbo eksperimentinius tyrimus įvertintas vaizdų analizės algoritmų vykdymo specializuotuose įterptinėse sistemose populiarėjimas. Dėl šios priežasties eksperimentiniais tyrimais siekta įvertinti galimybę tiriamus algoritmus naudoti būtent tokiose, ypač ribotų resursų kompiuterizuotose sistemose.

2.1. Atrinkti judesio analizės algoritmai

Šiame poskyryje pateikti eksperimentiniams tyrimams pasirinkti judesio analizės algoritmai, kurie buvo įgyvendinti ir ištirti siekiant įvertinti jų savybes ir tinkamumą realiojo laiko vaizdų atpažinimo uždaviniams spręsti.

Dėl pirmajame skyriuje aptartų plačių optinio srauto taikymo vaizdų atpažinimo uždaviniuose galimybių ir sėkmingo sąsūkos dirbtinių neuronų tinklų taikymo optiniam srautui skaičiuoti pavyzdžių, nuspręsta atlikti pradinį šių algoritmų palyginimą su klasikiais algoritmais, kuriuose netaikomi dirbtinių neuronų tinklai. Palyginimui pasirinkti algoritmai patiekti 2.1.1 skirsnyje.

Šio darbo eksperimentiniams tyrimams parinkti konkretūs judesio aptikimo ir optinio srauto algoritmai pateikti atitinkamai 2.1.2 ir 2.1.3 skirsniuose.

2.1.1. Pradiniam palyginimui atrinkti optinio srauto algoritmai

Optinio srauto apžvalgoje įvertintas *FlowNet2* algoritmas, pagal algoritmo autorius, pranoksta iki šiol pasiūlytus optinio srauto algoritmus, kuriuose nenaudojami dirbtiniai neuronų tinklai. Siekiant skaičiuoti optinį srautą realiuoju laiku ypatingai svarbi algoritmo sparta. Siekiant susipažinti su šių algoritmų skirtumais ir išskirtinėmis savybėmis, nuspręsta atlikti pradinį *FlowNet2* palyginimą su pora kitų algoritmų, kurių veikimas paprastesnis ir nereikalauja tokios našios aparatinės įrangos.

Šiam tyrimui pasirinkti šie trys algoritmai: *FlowNet2* [12], *Farneback* [8], *DIS* [9]. Toks algoritmų pasirinkimas leis palyginti jau kurį laiką gerai žinomą ir dažnai naudojamą *Farneback*

algoritmą su dviem naujais, bet veikimo principu visiškai skirtingais algoritmais ir leis geriau suprasti skirtingų tipų algoritmams būdingomis savybėmis.

2.1.2. Eksperimentiniam spartos tyrimui atrinkti judesio aptikimo algoritmai

Šio darbo pirmame skyriuje apžvelgti judesio aptikimo algoritmai gali būti naudingi ne tik pačiam judesio faktui aptikti, bet ir judančioms sritims vaizde išskirti, o tai naudinga detalesnei vaizdų analizei. Taikant šiuos algoritmus realiuoju laiku svarbu per vieno kadro periodą atlikti judesio aptikimą ir kitas norimas operacijas, o tai reiškia, kad kuo daugiau laiko užtrunka judesio aptikimas tuo mažiau laiko lieka kitoms operacijoms. Dėl šios priežasties nuspręsta atlikti kelių gerai žinomų ir plačiai naudojamų judesio aptikimo algoritmų spartos tyrimą juos įgyvendinus įterptinėse sistemose.

Šiam tyrimui parinkti septyni plačiai taikomi judesio aptikimo algoritmai:

- CNT (angl. *CouNTing*) – skaičiavimu grįstas algoritmas [24].
- MOG (angl. *Mixture Of Gaussians*) – Gauso skirstinių mišinio modeliu grįstas algoritmas [25].
- MOG2 – patobulintas Gauso skirstinių mišinio modeliu grįstas algoritmas [3].
- KNN (angl. *K-Nearest Neighbors*) – *K* artimiausiųjų kaimynų klasifikavimo algoritmu grįstas algoritmas [4].
- GMG (A. Godbehere'as, A. Matsukawa, K. Goldberg'as) – pagal autorių pavardes pavadintas algoritmas [26].
- GSOC (angl. *Google Summer Of Code*) – GSOC renginio metu sukurtas ir pasiūlytas algoritmas [27].
- LSBP (angl. *Local SVD Binary Pattern*) – matricos skaidymu singuliariomis reikšmėmis grįstas algoritmas [28].

Šiame sąrašė pateikti pavadinimai atitinka atvirojo kodo vaizdų apdorojimo ir analizės bibliotekoje *OpenCV* pateiktus pavadinimus, kai kurių algoritmų autoriai nesuteikė savo algoritmams pavadinimo, todėl patogumui toliau šiame darbe bus naudojami aukščiau pateikti pavadinimai.

2.1.3. Eksperimentiniam spartos ir tikslumo tyrimui atrinkti optinio srauto algoritmai

Didžioji dalis šiame darbe apžvelgtų judesio analizės algoritmų taikymo pavyzdžių naudoja optinio srauto algoritmus, todėl nuspręsta išrinkti ir ištirti dažniausiai naudojamus arba naujus, tačiau perspektyvius optinio srauto algoritmus, kurie gali būti sėkmingai įgyvendinti įvairioje aparaturoje. Šiomis savybėmis pasižymi bibliotekoje *OpenCV* įgyvendinti optinio srauto algoritmai, kurie gali būti įgyvendinti tiek įterptinėse sistemose, tiek asmeniniuose kompiuteriuose ar specializuotuose

duomenų apdorojimo serveriuose, taip pat palaikomas įvairių platformų aparatinis skaičiavimų spartinimas, todėl pasirinkta išsamiau ištirti šiuos konkrečius optinio srauto algoritmus:

- DIS (angl. *Dense Inverse Search*) – atvirkštinės vaizdų sąsajos paieškos principu grįstas algoritmas [9].
- *LK-SparseToDense* – *Lucas-Kanade* algoritmu grįstas tankiojo (angl. *dense*) optinio srauto algoritmas [29].
- *Farnebeck* – algoritmo autoriaus vardu pavadintas algoritmas [8].
- PCA (angl. *Principle Component Analysis*) – principinių komponentų analize grįstas algoritmas [30].
- *DeepFlow* – požymių sutapdinimo (angl. *matching*) dirbtiniais neuronų tinklais principu veikiantis algoritmas [31].
- *SimpleFlow* – neiteracinis, subtiesinis optinio srauto algoritmas [32].
- TVL1 (angl. *Total Variation L1 norm*) – pilnutinės variacijos pirmąja norma grįstas algoritmas [33].

Šiame darbe atlikti aukščiau pateiktų optinio srauto spartos ir tikslumo tyrimai. Šių tyrimų tikslas rasti optimalų optinio srauto algoritmą taikyti riboto našumo aparatinėje įrangoje, todėl bus ieškoma kompromiso tarp spartos ir tikslumo, kuris būtų tinkamas 2.2 poskyryje pateiktam optinio srauto algoritmui taikyti objektams aptikti ir sekti.

2.2. Pasiūlytas naujas objektų aptikimo ir sekimo, taikant optinį srautą, algoritmas

Dėl pastaruoju metu sparčiai tobulėjančių dirbtinių neuronų tinklų pagrindu veikiančių algoritmų ir tobulėjančios specializuotos jiems įgyvendinti skirtos aparatinės įrangos progreso, šiuos algoritmus galime taikyti kasdieninėje veikloje, asmeniniuose kompiuteriuose ar išmaniuosiuose telefonuose.

Ypač sėkmingai šie algoritmai taikomi vaizdų atpažinimo uždaviniams spręsti. Vienas iš svarbesnių naujų pasiekimų šioje srityje – sėkmingas ir efektyvus sąsūkos dirbtinių neuronų tinklų pritaikymas objektų aptikimo uždaviniui spręsti. Šį uždavinį sudaro objektų vaizde lokalizavimas ir jų atpažinimas (objekto klasės priskyrimas), tai leidžia atpažinti sudėtingus vaizdus, kuriuos sudaro daugelis objektų. Žymus šio tipo algoritmų žingsnis į priekį žengtas kai buvo pasiūlyta šiuos uždavinius apjungti į vieną ir atlikti viena kovoliucinio dirbtiniu neuronų tinklo tiesioginio sklaidimo (angl. *forward propagation*) operacija, tą sėkmingai pavyko pasiekti tokiais algoritmais kaip YOLO (angl. *You Only Look Once*) [34], SSD (angl. *Single Shot Detector*) [35].

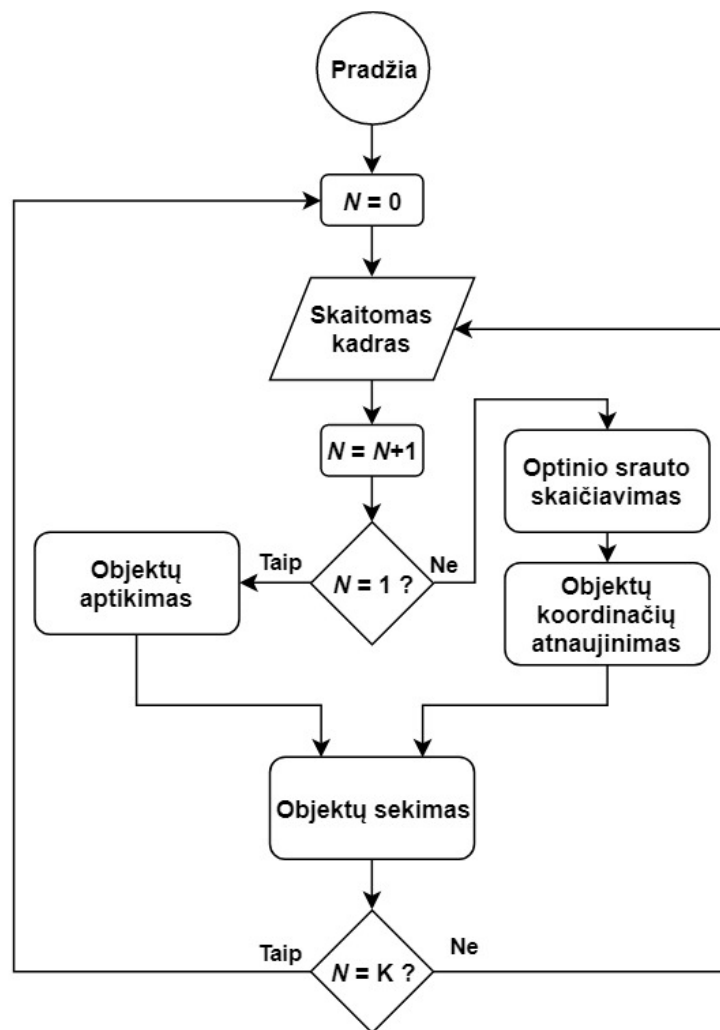
Šis pakeitimas leido žymiai sparčiau atlikti objektų atpažinimą, o pasitelkus itin našią specializuotą aparatinę įrangą, kuri leidžia skaičiavimus atlikti lygiagrečiai, tikslus objektų atpažinimas tapo pasiekiamas ir realiuoju laiku.

Dėl minėtos objektų aptikimo algoritmų sėkmės šie algoritmai imti taikyti ir daugelio objektų sekimo uždaviniuose, kurių tikslas fiksuoti sekamų objektų poziciją skirtinguose kadruose. Tokia analizė leidžia įvertinti objekto judėjimo trajektoriją, greitį ir kitas dominančias savybes. Objektams sekti taikant objektų aptikimo algoritmus, sekimas paremtas aptiktų objektų klasėmis ir pozicijomis vaizde, detaliau vienas tokio tipo objektų sekimo algoritmas aptartas 2.2.3 skirsnyje.

Žinoma, kad šiuolaikiniai objektų aptikimo algoritmai specializuotoje aparatūroje jau gali veikti realiuoju laiku, tačiau daugelyje praktinio taikymo sričių svarbi aparatūros kaina, dydis ir energetinis efektyvumas, todėl atsiranda poreikis šiuos algoritmus įgyvendinti pigesnėse, mažesnėse ir mažiau energijos reikalaujančiose sistemose. Tokioms sistemoms objektų aptikimo uždavinys nėra lengvas ir veikimas realiuoju laiku sunkiau pasiekiamas. Šio darbo metu pasiūlytas minėtos problemos sprendimas objektų aptikimo ir sekimo algoritmui spartinti taikant optinio srauto algoritmą.

2.2.1. Pasiūlytas algoritmas

Pagal šiame darbe atlikto optinio srauto spartos tyrimo rezultatus (pateikti 4 skyriuje) žinoma, kad šiuolaikiniai optinio srauto algoritmai gali būti skaičiuojami realiuoju laiku net įterptinėse sistemose ir optinio srauto skaičiavimas, priklausomai nuo apdorojamų vaizdų dydžio, gali būti daug spartesnis nei objektų aptikimas, kuriam, norint pasiekti aukštą tikslumą, net ir mažam vaizdo signalui apdoroti reikia naudoti daugelio sluoksnių dirbtinių neuronų tinklą, todėl galimos tokios uždavinio sąlygos, kai vienos kadrų poros optiniam srautui skaičiuoti reikia daug mažiau laiko nei vieno kadro objektams aptikti. Remiantis šia išvada ir tuo, kad įprastomis sąlygomis vaizdinėje medžiagoje objektai tarp kadrų pajuda santykinai nedaug, pasiūlytas objektų aptikimo ir sekimo realiuoju laiku algoritmas. Pasiūlytame algoritme, siekiant spartesnio algoritmo veikimo, po objektų aptikimo pirmame kadre, tam tikrame skaičiuje ateinančių kadrų aptiktų objektų pozicija atnaujinama pagal tiems objektams būdingą judesį, nustatytą iš apskaičiuoto optinio srauto. Taip dalis objektų aptikimo skaičiavimų pakeičiama optinio srauto ir objektų padėties atnaujinimo skaičiavimu taip sutaupant laiko. Tokiu būdu apskaičiuotos aptiktų objektų padėties gali būti naudojamos objektų sekimo uždaviniui. 2.1 pav. pateikta pasiūlyto algoritmo struktūrinė schema.



2.1 pav. Pasiūlyto daugelio objektų sekimo algoritmo struktūrinė schema

Pateiktoje schemoje prieš pirmojo kadro skaitymą inicializuojamas kadrų skaitiklis N po kiekvieno kadro skaitymo ji didinamas ir, priklausomai nuo jo vertės, atliekamas objektų aptikimas arba optinio srauto skaičiavimas ir prieš tai aptiktų objektų padėčių atnaujinimas pagal apskaičiuotą optinį srautą tų objektų ribose. Šios procedūros kartojamos kol N pasiekia nustatytą vert K , o tada N nustatomas iš naujo ir iki šiol apibūdintas procesas kartojamas. Toks algoritmas pirmąjį kadrą iš K ilgio kadrų sekos naudoja objektams aptikti, o likusius šios sekos kadrus naudoja optiniam srautui skaičiuoti ir atnaujinti pirmame kadre aptiktų objektų padėtį pagal jų srityse nustatytą optinį srautą.

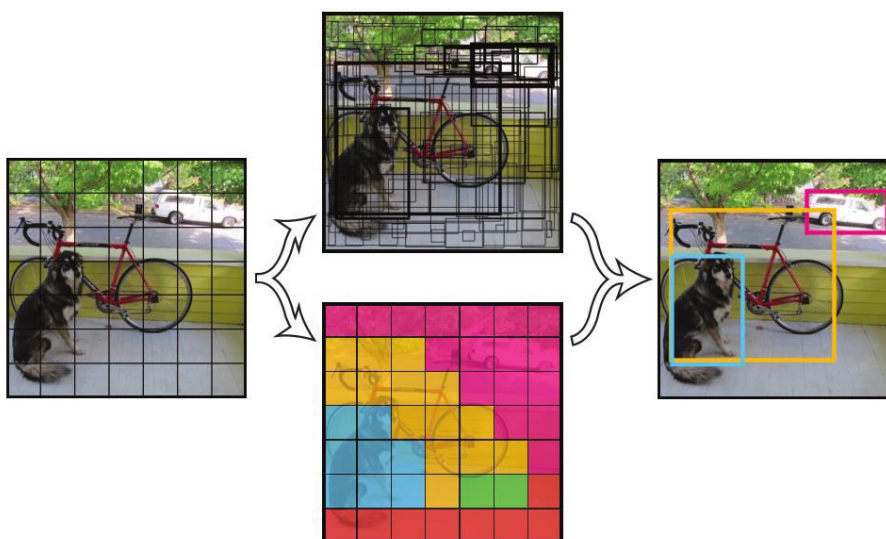
Teisingai parinkus sudaryto algoritmo sudedamąsias dalis galima gauti spartesnę algoritmą be žymaus tikslumo sumažėjimo. Šią hipotezę siekta patvirtinti šio darbo eksperimentinių tyrimų metu, kurių rezultatai pateikti 4 skyriuje.

Toliau šiame poskyryje pateiktos ir apibūdintos tyrimams parinktos pasiūlyto algoritmo sudedamosios dalys.

2.2.2. Objektų aptikimas vaizde

Objektų aptikimas vienas iš vaizdų atpažinimo uždavinių. Šio tipo algoritmais siekiama apdorojamame įėjimo vaizde aptikti ir lokalizuoti tam tikrus dominančius objektus, atpažinti kokie tai objektai. Tai detalesnis vaizdų atpažinimas nei įprastas vaizdų klasifikavimas, kurio metu įėjime pateikiamas tam tikro objekto vaizdas, o išėjime gaunamas vienareikšmis atsakymas apie tai, kuriai žinomai klasei tas objektas priklauso. Objektams aptikti gali būti naudojamas ir paprastas vaizdų klasifikatorius jį taikant atskiroms dalims, pavyzdžiui, slenkančio lango principu, o kiekvieno klasifikavimo poziciją apdorojamo vaizdo atžvilgiu ir rezultatus registruoti. Taip galima atrinkti vietas, kur rasti žinomų klasių objektai ir įgyvendinti objektų aptikimo uždavinį. Tai neefektyvu, nes klasifikavimas atliekamas daugybę kartų ir dažnai ten, kur jokios dominančio vaizdo informacijos nėra. Šiai problemai spręsti pasiūlytas ne vienas dominančių sričių (angl. *region of interest*) atrinkimas sumažinant nereikalingų skaičiavimų kiekį, tačiau šie siūlymai neprilygsta iš esmės kitokią idėją taikančiam pasiūlymui objektų aptikimo uždavinį įgyvendinti vienu dirbtinių neuronų tinklu, kuris mokomas nustatyti kur ir kokie objektai matomi apdorojamame vaizde. Šią idėją savo pasiūlytame algoritme įgyvendino algoritmo YOLO (angl. You Only Look Once) [34] autorius. Toliau šiame skirsnyje pateiktas YOLO algoritmo veikimo paaiškinimas.

Šį algoritmą struktūriškai galima išskirti į dvi dalis – požymių išskyrimo ir objektų aptikimo. Pirmąją dalį sudaro daugelio sluoksnių sąsūkos dirbtinių neuronų tinklas, apmokytas klasifikavimo uždaviniui, kurio paskirtis šiame algoritme iš vaizdo išgauti jame esančius požymius, leidžiančius aptikti objektus. Antroji dalis skirta pagal išgautus požymius aptikti objektus. Šios dalies struktūra priklauso nuo pasirinktų algoritmo parametrų. Pirmiausia reikia paminėti, kad apdorojamas vaizdas dalinamas į $S \times S$ vienodo dydžio segmentų ir kiekvienas segmentas bus apdorojamas atskirai. Tokį sudalinimą iliustruoja 2.2 pav.



2.2 pav. YOLO algoritmo veikimo skirtingų etapų iliustracija [34]

Kiekviename segmente skaičiuojami šie rezultatai: kai galimų klasių kiekis lygus C – skaičiuojama kiekvienos klasės buvimo šiame segmente tikimybė ir B kiekis objekto objektą gaubiančių kontūrų, o kiekvieną iš šių kontūrų sudaro kontūro viršutinio kairiojo kampo koordinatės, plotis, aukštis ir užtikrintumo, kad šis kontūras gaubia kokį nors objektą, įvertis. Tai pasiekama pirmosios algoritmo dalies išgautus požymius prijungiant prie pilnai jungto (angl. *fully connected*) dirbtinių neuronų sluoksnio, kurį sudarančių neuronų kiekis lygus $S \times S \times (B \cdot 5 + C)$. Tai leidžia objektų aptikimo uždavinį spręsti kaip regresijos problemą ir sudaryti tikslo funkciją, kad algoritmo išėjimai būtų kuo artimesni turimo sužymėto mokymo duomenų rinkinio duomenims. Toks algoritmas gali būti apmokytas objektų aptikimo uždaviniui spręsti ir yra pripažintas vienu geriausių šios problemos sprendimo būdų kai siekiama skaičiavimus atlikti realiuoju laiku.

Nors YOLO algoritmas spartesnis už paprastu klasifikavimu grįstus objektų aptikimo algoritmus, jo tikslumas mažesnis. Pagal autoriaus tyrimus ir atliktas apžvalgas algoritmas prasčiau aptinka mažus objektus. Šią problemą adresavo naujesnės YOLO versijos: YOLOv2 [36] ir YOLOv3 [37] ir sėkmingai pritaikė YOLO patobulinimus, leidusius padidinti algoritmo tikslumą mažiems objektams (pritaikytas aptikimas keliais masteliais) ir bendrą aptikimo tikslumą.

Šis algoritmas dėl savo spartos ir tikslumo tinkamas realiojo laiko uždaviniams ir pasirinktas naudoti pasiūlytam objektų sekimo algoritmui įgyvendinti ir ištirti, tačiau pasiūlytam objektų sekimo algoritmui galima naudoti ir kitą objektų aptikimo algoritmą.

2.2.3. Objektų sekimas vaizde

Objektų sekimas vaizde – uždavinys, kurio tikslas analizuojant vaizdų srautą nustatyti to paties objekto poziciją kiekviename vaizdų srauto kadre jai kintant. Tai leidžia stebėti objektų judėjimo trajektoriją, greitį ir kitas dominančias savybes. Kai vaizde matomas tik vienas judantis objektas, jo sekimas gali būti įgyvendintas naudojant pirmame skyriuje apžvelgtus judesio aptikimo algoritmus, objekto padėtį atnaujinant pagal judesio aptikimo rezultatus (būtų matomas aptikto judesio koordinatė kitimas). Vieno objekto sekimas yra gana paprastas uždavinys, tačiau realybėje dažniau sutinkamas poreikis sekti daugelį objektų. Kai analizuojamame vaizdų sraute sekami keli objektai iškyla ne tik objekto padėties atnaujinimo, bet ir aptiktų objektų identifikavimo uždavinys. Daugelio objektų sekimo algoritmas turėtų kiekvienam stebimam objektui priskirti identifikatorių ir išlaikyti priskirtą identifikatorių kiekviename vaizdų srauto kadre.

Stiprus objektų atpažinimo algoritmų spartos šuolis lėmė, kad objektų aptikimas gali būti vykdomas realiuoju laiku ir analizuojant vaizdų srautą, kiekviename kadre gali būti aptinkami visi dominantys objektai. Turint šiuos duomenis, norint įgyvendinti daugelio objektų sekimą, lieka identifikatorių priskyrimo uždavinys. Atsižvelgiant į tai, kad šiuolaikiniams vaizdų srautams būdingas aukštas kadrų dažnis, dažnai galima daryti prielaidą, kad tarp dviejų gretimų kadrų objektai

vaizde pajuda santykinai nedaug, o tai reiškia, kad galiojant šiai sąlygai sekamų objektų identifikatorius galima priskirti remiantis mažiausio atstumo tarp aptiktų objektų kriterijumi.

Ši idėja įgyvendinta SORT (angl. *Simple Online and Realtime Tracking*) [38] daugelio objektų sekimo algoritme, kuris paremtas sekimu naudojant objektų aptikimą. Šis algoritmas objektams sekti naudoja kiekvieno kadro aptiktų objektų sąrašą, pirmame kadre kiekvienam objektui priskiriamas unikalus identifikatorius, gavus antro kadro aptikimų sąrašą skaičiuojamas kiekvienam pirmojo kadro objektui skaičiuojamas jį ribojančio stačiakampio ir kiekvieno iš antrame kadre aptiktus objektus ribojančių stačiakampių sankirtos ir sąjungos plotų santykį IOU (angl. *Intersection Over Union*). Po to, pritaikant *Vengrišką* priskyrimo algoritmą pagal apskaičiuotą IOU parametą, atliekamas pirmojo kadro objektų identifikatorių priskyrimas antrajame kadre aptiktiems objektams. Šiame algoritme taip pat objektų judesiui modeliuoti ir filtruoti (kai, pavyzdžiui, dalį laiko objektą užstoja kitas objektas) naudojamas gerai žinomas ir daugelyje sričių pritaikytas Kalmano filtras. *SORT* algoritmas yra paprastas, bet efektyvus, sparta ir tikslumu pranokstantis daugelį kitų pasiūlytų algoritmų, dėl to jis pritaikytas baigiamajame darbe pasiūlyto algoritmo vaizdų sekimo etape.

2.3. Trumpas skyriaus apibendrinimas

Šiame skyriuje pateikti tyrimams pasirinkti judesio aptikimo ir optinio srauto algoritmai, siekiant atskleisti judesio analizės algoritmų taikymo vaizdų atpažinimo uždaviniuose galimybes, pasiūlytas daugelio objektų atpažinimo ir sekimo algoritmas, kuriame naudojant optinį srautą didinama algoritmo veikimo sparta. Pasiūlytas algoritmas įgyvendintas specializuotoje aparatūroje ir ištirtas, tyrimo rezultatai pateikti ketvirtame skyriuje.

3. DUOMENŲ PARUOŠIMAS JUDESIO ANALIZĖS ALGORITMŲ EKSPERIMENTINIAM TYRIMUI

Tinkamas duomenų parinkimas ir paruošimas ypatingai svarbus vaizdų technologijų srities tyrimuose. Nuo tyrimams parinktų duomenų gali priklausyti tyrimo rezultatai, teisingai parinkti duomenys gali padėti atskleisti svarbias tyrimo objekto savybes, todėl itin svarbu apgalvotai pasirinkti ir paruošti duomenis.

Šiame skyriuje pateikti ir paaiškinti duomenų baigiamojo darbo eksperimentiniams tyrimams pasirinkimai. Baigiamajame darbe suplanuoti eksperimentai skiriasi savo specifika ir siekiamais tikslais, todėl jiems paruošti skirtingi duomenų rinkiniai, kurie pateikti atskiruose šio skyriaus poskyriuose.

3.1. Duomenų paruošimas pradiniam optinio srauto algoritmų palyginimui

Pradiniu optinio srauto algoritmų palyginimu siekiama įvertinti pagrindinius skirtumus tarp sąsūkos dirbtinių neuronų tinklus naudojančių optinio srauto algoritmų ir paprastesnių, mažiau skaičiavimo resursų reikalaujančių algoritmų.

Šiuo palyginimu siekiama vizualiai įvertinti skirtingais optinio srauto algoritmais apskaičiuoto optinio srauto vizualizacijos skirtumus, kai optinis srautas skaičiuojamas tam pačiam vaizdų srautui.

Šiam bandymui pasirinkta naudoti du vaizdo įrašus, pirmasis įrašas įrašytas išmaniojo telefono kamera, namų sąlygomis, šiame įrašė matomas kompiuterio pelė ratu ant stalo judinanti ranka, juda didelė vaizdo dalis ir dėl susidariusių šešėlių kinta apšvietimas, o antras įrašas paimtas iš itin populiarios bibliotekos *OpenCV* dokumentacijos, šiame įrašė matomi nekintančiame vaizde vaikstantys žmonės, kurių plotas, palyginus su fono plotu, nedidelis, vaizdas gerai apšviestas. Pasirinktas vaizdo įrašų derinys leis tiriamus algoritmus palyginti skirtingomis sąlygomis, pasirinktų vaizdo įrašų kadrai pavyzdžiai pateikti 3.1 pav.



3.1 pav. Pradiniam optinio srauto palyginimui pasirinktų vaizdo įrašų kadrai

Judesio analizės algoritmų apžvalgoje pateikta, kad *FlowNet2* algoritmas tai sąsūkos dirbtinių neuronų tinklas, kuris sudarytas ir apmokytas skaičiuoti optinį srautą. Apmokymas – ko gero svarbiausias tokių algoritmų etapas, nuo kurio priklauso algoritmo veikimas, todėl siekiant įvertinti algoritmų autorių pristatytą algoritmą pasinaudota *FlowNet2* autorių pateiktu apmokytu algoritmo modeliu. Šiam modeliui reikia nurodyti du kadrus kaip atskirus vaizdo failus ir kur išsaugoti rezultatą arba tekstinius failus, kuriuose parašyti kadrų porų ir rezultatų failų pavadinimai. Dėl tokio modelio įėjimo ir išėjimo formato parašyta programa, suskaidanti vaizdo įrašus į atskirus kadrus bei sudaranti kadrų porų ir rezultatų failų pavadinimus. Programai naudota *Python* programavimo kalba, vaizdų apdorojimui naudota *OpenCV* biblioteka.

Kitiems šiame tyrime naudotiems optinio srauto algoritmams tokio paruošimo nereikėjo, todėl naudoti pradiniai vaizdo įrašai įrašai.

3.2. Duomenų paruošimas atrinktų optinio srauto ir judesio aptikimo algoritmų spartai ir optinio srauto algoritmų tikslumui tirti

Šiame darbe tiriamų optinio srauto ir judesio aptikimo algoritmų maksimali veikimo sparta priklauso tik nuo apdorojamų vaizdų dydžio ir nepriklauso nuo apdorojamų vaizdų turinio, todėl šių algoritmų spartos palyginimui iš laisvai pasirinkto vaizdo įrašo sudarytos dviejų skirtingų raiškų šio įrašo kopijos, 320×240 ir 160×120 taškų. Tokios, palyginti nedidelės, raiškos pasirinktos dėl vaizdų apdorojimui būdingų aukštų skaičiavimo resursų sąnaudų ir atsižvelgus į tai, kad norimas šių algoritmų veikimas realiuoju laiku net juos įgyvendinus įterptinėse sistemose. Šiems vaizdo įrašams jokio papildomo apdorojimo neprireikė.

Nors šiame darbe siekiama atskleisti judesio analizės algoritmų taikymo vaizdų atpažinimo realiuoju laiku uždaviniuose galimybes ir daug dėmesio skiriama algoritmų spartai, taip pat svarbus ir šių algoritmų tikslumas.

Šiame darbe nuspręsta atlikti tiriamų optinio srauto algoritmų tikslumo tyrimą ir pagal tyrimo rezultatus palyginti žinomus optinio srauto algoritmus. Optinio srauto tikslumo įvertinimas nėra lengva užduotis dėl duomenų, kuriems būtų nustatytas visiškai tikslus optinis srautas, trūkumo. Daugelis žinomų duomenų rinkinių yra sintetiniai arba labai nedideli, tačiau tokių duomenų rinkinių yra. Šio darbo metu optinio algoritmo tikslumui tirti pasirinktas *Middlebury* optinio srauto duomenų rinkinys [2], kuriame yra kadrų sekų rinkinys su pateiktu tikruoju (angl. *ground truth*) optiniu srautu. Tai leidžia tiriamais algoritmais apskaičiuvus optinį srautą jį palyginti su tikruoju optiniu srautu.

Vienas optinio srauto įvertinimo būdų – vidutinės optinio srauto vektorių galų paklaidos EPE (angl. *End Point Error*) kriterijus, kuris įvertina vidutinį atstumą tarp apskaičiuoto ir tikrojo judesio vektoriaus. Vienam taškui ši paklaida gali būti apskaičiuota pagal (3.1).

$$E_{EP} = \sqrt{(u - u_{gt})^2 + (v - v_{gt})^2}, \quad (3.1)$$

čia E_{EP} – klaida pagal EPE, u ir v – taškui apskaičiuotas judesys tarp kadrų kiekviena dvimačio vaizdo kryptimi, u_{gt} ir v_{gt} – tikrojo optinio srauto tame pačiame taške.

Optinio srauto algoritmų tikslumo tyrimo metu algoritmų tikslumas vertintas pagal EPE, naudojant tuos *Middlebury* optinio srauto duomenų rinkinio vaizdus, kuriems yra pateiktas tikslus optinis srautas. Naudotą vaizdų sąrašą sudaro: *RubberWhale*, *Hydrangea*, *Dimetrodon*, *Grove2*, *Grove3*, *Urban2*, *Urban2*, *Venus* vaizdų sekos.

3.3. Duomenų paruošimas pasiūlyto objektų aptikimo ir sekimo, taikant optinį srautą, algoritmui tirti

Daugelio objektų aptikimo ir sekimo uždavinys sudėtinis, todėl jam įvertinti reikalingas duomenų rinkinys, kuriame būtų pateikti pradiniai vaizdai, jie būtų naudojami objektams aptikti, o kartu šiame rinkinyje turėtų būti sužymėti teisingi sekimo rezultatai, pagal kuriuos būtų įvertintas tiriamo algoritmo tikslumas.

Šiame darbe nagrinėjamos judesio analizės algoritmų taikymo objektams aptikti ir sekti vaizde galimybės ir viename iš pasiūlyto algoritmų etape nuspręsta naudoti optinį srautą, o tam reikia, kad pasirinkto duomenų rinkinio vaizdų sekos būtų nuosekliai išrinktos ir vaizdo įrašo, kad būtų būdingas natūralus judesys ir būtų galima apskaičiuoti optinį srautą gretimų kadrų poroms.

Visomis minėtomis savybėmis pasižymi daugelio objektų sekimo iššūkio MOT15 [40] pasiūlytas vaizdo sekų rinkinys, kuris pateikiamas kaip kiekvieną seką sudarančių kadrų rinkinys ir tai sekai apskaičiuotas teisingas sekimo rezultatas, pateiktas tekstinio failo pavidale, kuriame kiekvienam kadrai parašyti tame kadre esančių objektų padėtis, jiems priskirtas identifikatorius ir pateikta kita naudinga informacija, kuri padeda spręsti ir kitus uždavinius, pavyzdžiui, vien objektų aptikimo.

Šiame darbe pasirinkta naudoti MOT15 mokymo duomenų rinkinį, kurį sudaro 11 kadrų sekų ir vaizdų sekimo tose sekose teisingi rezultatai, skirti palyginti tiriamą sekimo algoritmą su idealium. Viso šiame duomenų rinkinyje 5500, kadrų, kuriuose beveik 40 tūkstančių objektų. Duomenų rinkinį sudarančios vaizdų sekos skiriasi sąlygomis, kuriomis jos buvo įrašytos, žemiau trumpai apibūdinta kiekviena iš sekų:

- *Venice-2* – didelėje aikštėje vaikstantys žmonės, kamera nejuda, matomi įvairaus dydžio objektai, skirtingos objektų judėjimo kryptys.
- *KITTI-17* – saulėtą dieną vaikstantys pėstieji, nejudanti kamera.
- *KITTI-13* – intensyvi miesto scena, nufilmuota į priekį judančia kamera.
- *ADL-Runde-8* – naktį nufilmuoti pėstieji, judanti kamera, kameros drebinimas.
- *ADL-Runde-6* – iš apačios nufilmuoti praeiviai prekybos centre, nejudanti kamera.
- *ETH-Pedcross2* – pėstieji ir automobiliai gatvėje, nufilmuoti iš judančios platformos.
- *ETH-Sunnyday* – miesto gatvės scena, nufilmuota saulėtą dieną iš judančios platformos.

- *ETH-Bahnhof* – praeiviai mieste, nufilmuoti iš judančios platformos, kameros drebėjimas.
- *PETS09-S2L1* – iš stebėjimo kameros (stacionarios) nufilmuoti 8 kameros matymo lauke vaikstantys pėstieji (panaudota ir 3.1 poskyryje pateiktame antrajame vaizdo įrašė).
- *TUD-Campus* – iš šono nufilmuoti praeiviai, kamera nejuda.
- *TUD-Stadtmitte* – gatvėje nufilmuoti praeiviai, kamera nejuda.

Pagrindinės duomenų rinkinio savybės pateiktos 3.1 lentelėje.

3.1 lentelė. Pasirinktos objektų sekimo duomenų rinkinio vaizdo sekos iš MOT15 [40]

Pavadinimas	Kadru dažnis, Hz	Raiška	Vaizdų kiekis	Sekami objektai	Objektų aptikimų	Vidutiniškai objektų kadre
Venice-2	30	1920×1080	600	26	7141	11,9
KITTI-17	10	1224×370	145	9	683	4,7
KITTI-13	10	1242×375	340	42	762	2,2
ADL-Runde-8	30	1920×1080	654	28	6783	10,4
ADL-Rundle-6	30	1920×1080	525	24	5009	9,5
ETH-Pedcross2	14	640×480	837	133	6263	7,5
ETH-Sunnyday	14	640×480	354	30	1858	5,2
ETH-Bahnhof	14	640×480	1000	171	5415	5,4
PETS09-S2L1	7	768×576	795	19	4476	5,6
TUD-Campus	25	640×480	71	8	359	5,1
TUD-Stadtmitte	25	640×480	179	10	1156	6,5
Iš viso:	-	-	5500	500	39905	7,3

Iš 3.1 lentelėje pateiktų duomenų matyti, kad vaizdų sekos skiriasi savo kadru dažniu, raiška, ilgiu ir objektų tankiu.

Objektų sekimo uždavinys gana sudėtingas ir nėra paprasta objektyviai įvertinti objektų sekimo algoritmų rezultatus. Šiam tikslui yra žinoma būtų įvertinti daugelio objektų sekimo tikslumą. Pasirinkto duomenų rinkinio autoriai objektų sekimo algoritmams vertinti siūlo naudoti jų sudarytą objektų sekimo tikslumo metrikų sąrašą, kurį sudaro šie pagrindiniai įverčiai:

- FP (angl. *False Positive*) – klaidingai aptikti objektai.
- FN (angl. *False Negative*) – neaptikti objektai.
- R (angl. *Recall*) – teisingai aptiktų TP (angl. *True Positive*) objektų ir tikro esančių objektų kiekio, kurį sudaro FN ir TP suma, santykis, išreiškiamas vieneto dalimis arba procentais.
- P (angl. *Precision*) – TP ir visų aptiktų objektų kiekio santykis, kurį sudaro TP ir FP suma, išreiškiama vieneto dalimis arba procentais.

- FAR (angl. *False Alarm Rate*) – vidutinis viename kadre klaidingai aptiktų objektų kiekis.
- GT (angl. *Ground Truth*) – tikrasis objektų judėjimo trajektorijų kiekis.
- MT (angl. *Mostly Tracked*) – nustatytų trajektorijų, kurių didžioji dalis teisinga, kiekis.
- PT (angl. *Mostly Tracked*) – nustatytų trajektorijų, kurių dalis teisinga, kiekis.
- ML (angl. *Mostly Tracked*) – nustatytų trajektorijų, kurių didžioji dalis neteisinga, kiekis.
- IDS (angl. *ID Switches*) – vieno objekto trajektorijoje objekto identifikatoriaus pasikeitimų kiekis.
- FM (angl. *Fragmentation*) – trajektorijų trūkių kiekis (kai tarp kadru pametamas objektas ir trajektorija nebepratesiama).
- MOTA (angl. *Multi-Object Tracking Accuracy*) – sudėtinis daugelio objektų sekimo algoritmų taiklumo įvertis, skaičiuojamas pagal 3.2 išraišką.
- MOTP (angl. *Multi-Object Tracking Precision*) – daugelio objektų sekimo tikslumas, įvertinantis sekamo objekto padėties nustatymo tikslumą, skaičiuojamas pagal (3.3).

$$MOTA = 1 - \frac{\sum_t (FN_t + FT + IDS_t)}{\sum_t GT_t}, \quad (3.2)$$

čia t – kadro numeris, GT – tikras kadre esančių objektų kiekis, o kiti žymėjimai sutampa su prieš tai pateiktomis santrumpomis,

$$MOTP = \frac{\sum_{i,t} d_{i,t}}{\sum_t c_t}, \quad (3.3)$$

čia $d_{i,t}$ – kadre t esančio teisingai atspėto objekto gaubiančiojo stačiakampio i ir jį atitinkančio tikrojo objekto gaubiančiojo stačiakampio plotų persidengimo įvertis (anksčiau minėtas IOU), c_t – teisingai aptiktų objektų kadre t kiekis.

Šiame darbe atliktų eksperimentinių objektų sekimo algoritmo tyrimų metu naudotos čia pateiktos objektų sekimo algoritmų įvertinimo metrikos, naudojant aukščiau apibūdintą MOT15 duomenų rinkinį. Didelis kiekis objektų aptikimo ir sekimo algoritmų įvertinimo metrikų leidžia palyginti skirtingas tiriamo algoritmo savybes ir tiksliau įvertinti pasiūlyto algoritmo tikslumo priklausomybę nuo keičiamų parametrų.

3.4. Objektų aptikimo algoritmo modelio pasirinkimas

Pasiūlyto objektų aptikimo ir sekimo algoritmo sudėtyje pasirinkta naudoti optimizuotą objektų aptikimo algoritmo YOLOv3 modifikaciją, vadinamą YOLOv3-tiny [37]. Objektų aptikimo ir sekimo algoritmams įvertinti pasirinktame duomenų rinkinyje sekami objektai yra pėstieji, todėl baigiamojo darbo bandymams reikia pėsčiuosius atpažinti apmokyto objektų aptikimo dirbtinių neuronų tinklo. Baigiamojo darbo eksperimentiniams pasiūlyto algoritmo tyrimams pasirinkta naudoti COCO duomenų rinkiniu [41] apmokytą YOLOv3-tiny objektų aptikimo modelį, kuris gali aptikti ne tik pėsčiuosius, bet ir dar 90 dažniausiai pasitaikančių objektų. COCO duomenų rinkinys naudojamas daugelio objektų atpažinimo algoritmams mokyti, nes yra pakankamai didelis ir įvairus.

3.5. Trumpas skyriaus apibendrinimas

Šiame skyriuje paaiškintas duomenų baigiamojo darbo eksperimentiniams tyrimams pasirinkimas, pateiktos ir paaiškintos tyrimuose naudotos metrikos.

Judesio aptikimo ir optinio srauto spartos bandymams pasirinktas MOT15 duomenų rinkinyje esantis PETS09-S2L1 vaizdo įrašas, prieš bandymus padarytos dvi skirtingų raiškų šio įrašo kopijos.

Optinio srauto algoritmų tikslumui įvertinti pasirinkta naudoti *Middlebury* duomenų rinkinį, kuriame pateiktos kadrų poros ir jas atitinkantis tikslus optinis srautas. Pateiktą tikslų optinį srautą galima naudoti kaip etaloną ir palyginti tiriamais algoritmais apskaičiuotu optiniu srautu. Optinio srauto tikslumui įvertinti pasirinktas EPE kriterijus.

Pasiūlyto objektų aptikimo ir sekimo algoritmui tirti pasirinkta naudoti MOT15 duomenų rinkinį, kuriame yra visi duomenys, kurių reikia, kad būtų galima apskaičiuoti optinį srautą, pritaikyti objektų aptikimo algoritmą, atlikti objektų sekimą ir palyginti gautus rezultatus su rankiniu būdu sužymėtais teisingais objektų sekimo atsakymais. Objektų sekimo algoritmams įvertinti naudotos MOT15 duomenų rinkinio autorių siūlytos metrikos, kurios paaiškintos 3.3 skirsnyje.

Šiame skyriuje pateikti duomenys panaudoti eksperimentiniuose tyrimuose, o atliktų tyrimų rezultatai pateikti 4 skyriuje.

4. JUDESIO ANALIZĖS ALGORITMŲ TYRIMO REZULTATAI

Šiame skyriuje pateikti ir aptarti baigiamojo darbo metu atliktų eksperimentinių tyrimų rezultatai. Eksperimentai vykdyti pagal sudarytą eksperimentų planą ir padalinti į tokius atskirus etapus:

- pradinis skirtingo tipo optinio srauto algoritmų palyginimas,
- judesio aptikimo ir optinio srauto algoritmų spartos tyrimas,
- optinio srauto algoritmų tikslumo tyrimas,
- pasiūlyto objektų aptikimo ir sekimo algoritmo tyrimas.

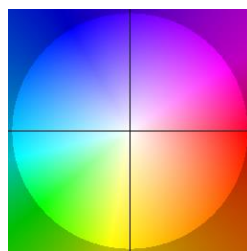
Toliau šiame skyriuje pateikti judesio analizės algoritmų tyrimo rezultatai. Kiekvieno atlikto tyrimo etapo rezultatai pateikti atskiruose poskyriuose.

4.1. Pradinio optinio srauto algoritmų palyginimo rezultatai

Šio tyrimo tikslas – išsiaiškinti skirtingo tipo optinio srauto algoritmų svarbiausias išskirtines savybes. Tyrimo metu palyginti *FlowNet2*, *Farneback* ir *DIS* optinio srauto algoritmai, skaičiuojant optinį srautą tiems patiems vaizdo įrašams. Tyrimams naudoti du vaizdo įrašai: kintančio apšvietimo vaizdo įrašas, kuriame matoma ant stalo kompiuterio pelę judinanti ranka, ir MOT15 objektų sekimo algoritmų įvertinimo duomenų rinkinyje esantis PETS09-S2L1 vaizdo įrašas, kuriame matomi stacionaria kamera nufilmuoti vaikštantys žmonės.

Tiriami algoritmai įgyvendinti *Python* programavimo kalba, naudojant *OpenCV* biblioteką. Eksperimentams atlikti naudotas specializuotas kompiuteris su aštuonių branduolių *Intel® Xeon® Processor E5-2630 v3* centriniu procesoriniu įrenginiu, *NVIDIA Quadro P4000* grafiniu procesoriniu įrenginiu ir 30 GB operatyviaja atmintimi. Tokios aukšto našumo įrangos prireikė *FlowNet2* algoritmui įgyvendinti.

Parašyta programa apskaičiuoto vaizdo įrašų optiniam srautui vaizdinti naudotas optinio srauto vaizdinimo spalvomis HSV spalvų erdvėje būdas. Naudojant šį vaizdinimo būdą taško spalva priklauso nuo optinio srauto vektoriaus krypties tame taške, o intensyvumas – nuo šio vektoriaus ilgio. Tokio vaizdinimo pavyzdys pateiktas 4.1 pav.



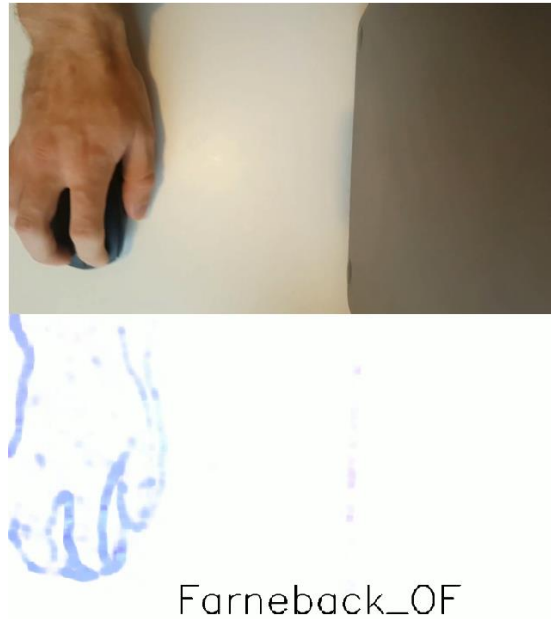
4.1 pav. Optinio srauto vaizdinimas spalvomis

4.1 pav. parodyta, kaip tokiu būdu vaizdinamo optinio srauto spalva priklauso nuo nustatyto judesio greičio ir krypties. Pavyzdžiui, jeigu 4.1 pav. esančio vidinio apskritimo spindulys lygus

vienetui, tuomet jeigu apskaičiuotas judesys tarp kadrų poros tame taške yra nukreiptas į dešinę ir lygus vienam – taškas bus ryškios raudonos spalvos, o jeigu jis nukreiptas į apačią – taškas bus geltonos spalvos. Paaiškintu principu galima vaizdžiai pateikti apskaičiuotą optinį srautą aiškiai vizualiai išskiriant skirtingo greičio ir skirtingų krypčių judesį.

4.1.1. *Farneback* algoritmu apskaičiuotas optinis srautas

Parašytoje programoje įgyvendintu *Farneback* algoritmu apskaičiuoto optinio srauto pavyzdžiai pateikti 4.2 ir 4.3 pav.



4.2 pav. Pirmojo vaizdo įrašo kadras (viršuje) ir *Farneback* algoritmu apskaičiuotas optinis srautas (apačioje)



4.3 pav. Antrojo vaizdo įrašo kadras (kairėje pusėje) ir *Farneback* algoritmu apskaičiuotas optinis srautas (dešinėje pusėje)

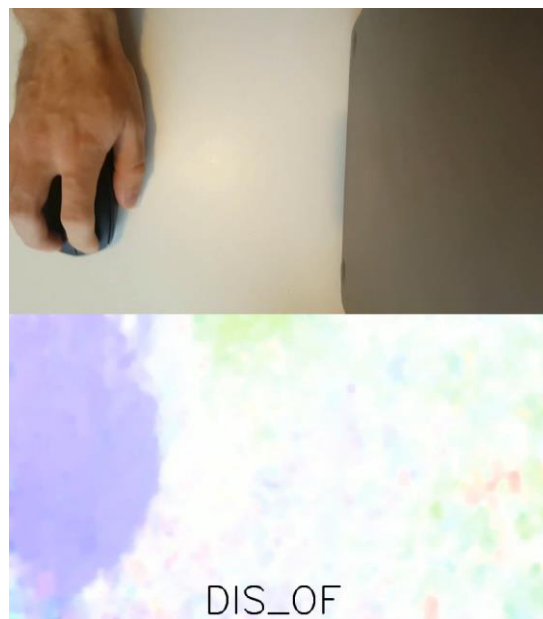
4.2 ir 4.3 pav. pastebima, kad *Farneback* algoritmu gerai apskaičiuojamas optinis srautas ties objektų briaunomis (didelio kontrasto srityse), tačiau objektų vidinėse srityse optinio srauto rezultatas lygus nuliui (balta spalva), o neteisingas rezultatas, nes, pavyzdžiui, pirmojo įrašo atveju visa ranka

juda vienodai, tačiau judesys nustatytas tik ties rankos ir stalo ribomis. Taip gali būti dėl judančio objekto spalvos tolygumo objekto srityje, tačiau tai nėra norimas rezultatas.

Šio bandymo metu vieno kadro optinio srauto skaičiavimas užtruko mažiau nei kadro atvaizdavimo trukmė, kuri pirmajam vaizdo įrašui lygi apie 42 ms, o antrajam – 100 ms, todėl parašyta programa gali būti taikoma realiuoju laiku.

4.1.2. DIS algoritmu apskaičiuotas optinis srautas

Parašytoje programoje įgyvendintu DIS algoritmu apskaičiuoto optinio srauto pavyzdžiai pateikti 4.4 ir 4.5 pav.



4.4 pav. Pirmojo vaizdo įrašo kadras (viršuje) ir DIS algoritmu apskaičiuotas optinis srautas (apačioje)



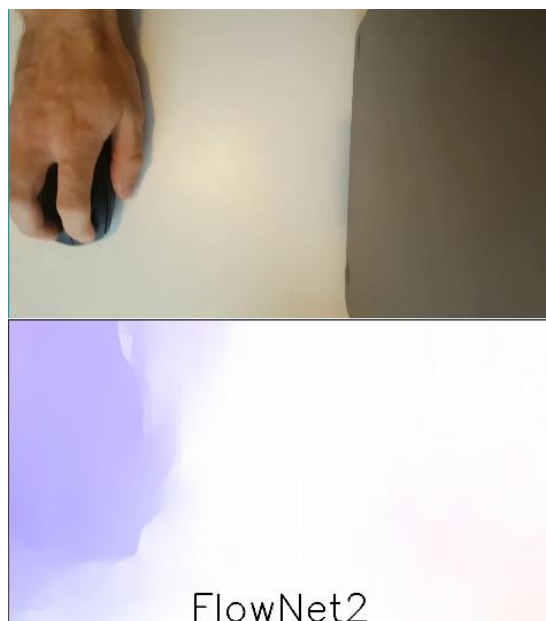
4.5 pav. Antrojo vaizdo įrašo kadras (kairėje pusėje) ir DIS algoritmu apskaičiuotas optinis srautas (dešinėje pusėje)

Iš 4.4 ir 4.5 pav. matyti, kad šis algoritmas geriau nei *Farneback* apskaičiuoja optinį srautą objektų vidinėse dalyse, kur spalva pastovi, tačiau pirmojo vaizdo įrašo rezultatuose matomas žymus triukšmas dėl prastų apšvietimo sąlygų. Tai nepasireiškia antrajame vaizdo įrašė, nes šiame

apšvietimo sąlygos geros ir pastovios. Šio algoritmo skaičiavimo sparta tuo pačiu kompiuteriu, kaip ir *Farneback* algoritmo bandymo metu, buvo pakankama apdoroti vaizdo įrašus realiuoju laiku. Tiksliau optinio srauto algoritmų sparta įvertinta 4.2 poskyryje pateiktais bandymų rezultatais. Tačiau tai leidžia šiuos algoritmus taikyti realiuoju laiku.

4.1.3. *FlowNet2* algoritmu apskaičiuotas optinis srautas

Parašytoje programoje įgyvendintu *FlowNet2* algoritmu apskaičiuoto optinio srauto pavyzdžiai pateikti 4.6 ir 4.7 pav.



4.6 pav. Pirmojo vaizdo įrašo kadras (viršuje) ir *FlowNet2* algoritmu apskaičiuotas optinis srautas (apačioje)



4.7 pav. Antrojo vaizdo įrašo kadras (kairėje pusėje) ir *FlowNet2* algoritmu apskaičiuotas optinis srautas (dešinėje pusėje)

Iš 4.6 ir 4.7 pav. matyti, kad *FlowNet2* algoritmu apskaičiuotas optinis srautas pasižymi tolygumu objektų vidaus srityse, atsparumu smulkiam triukšmui, ryškiais judančių objektų kontūrais, tačiau esant prastoms apšvietimo sąlygoms (pirmasis įrašas) rezultatai prastesni, matomas apskaičiuotas judesys srityse, kur jo nėra, ypač kai kinta didelę vaizdo dalį užimantys šešėliai arba atspindžiai. Be

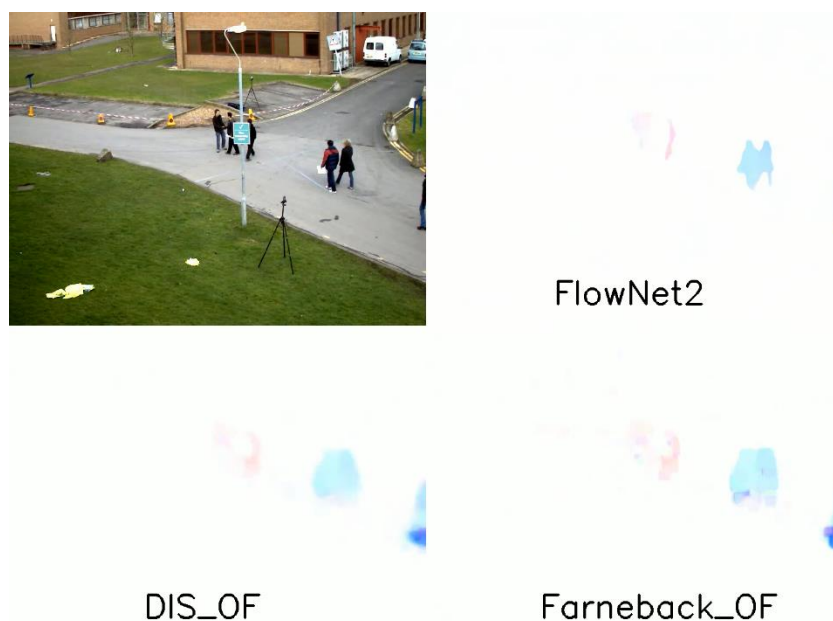
to, svarbu įvertinti, kad net naudojant ypač našų kompiuterį su skaičiavimus spartinančiu grafiniu procesoriniu įrenginiu, vienos kadrų poros apdorojimas truko tarp 1 ir 2 sekundžių, (kadrų apdorojimo dažnis 1 Hz ir 0,5 Hz atitinkamai), o tai žymiai mažesnis dažnis nei vaizdo įrašų kadrų dažniai (24 Hz pirmame vaizdo įraše ir 10 Hz antrame). Todėl šio algoritmo sparta, naudojant pasirinktą aparatūrą, nepakankama taikyti realiojo laiko uždaviniuose.

4.1.4. Skirtingais algoritmais apskaičiuoto optinio srauto palyginimas

Šiame skirsnyje palygintas ir apibendrintas *Farneback*, DIS ir *FlowNet2* algoritmais apskaičiuotas to paties įrašo optinis srautas, jo pavyzdžiai tam pačiam vaizdo įrašo kadrai pateikti 4.8 ir 4.9 pav.



4.8 pav. Skirtingais algoritmais apskaičiuotas pirmojo įrašo kadro optinis srautas



4.9 pav. Skirtingais algoritmais apskaičiuotas antrojo įrašo kadro optinis srautas

Iš 4.8 ir 4.9 pav. pateiktų palyginimų aiškiai pastebimos jau aptartos kiekvieno algoritmo savybės. Esant prastoms apšvietos sąlygoms, šešėliams ir atspindžiams *FlowNet2* ir DIS algoritmais apskaičiuotas optinis srautas pasižymi neaiškiais kontūrais ir dideliu triukšmu, tačiau jiems būdingas apskaičiuotos krypties tolygumas judančių objektų srityse. Tuo tarpu *Farneback* algoritmas ne toks jautrus apšvietimo sąlygoms, tačiau optinį srautą gerai apskaičiuoja tik ties objektų kraštais.

Esant geroms apšvietimo sąlygoms aiškiausiai atrodo *FlowNet2* rezultatai, (pastovus fonas, ryškūs kontūrai, tolygumas judančių objektų vidaus srityse), tačiau šis algoritmas reikalauja žymiai daugiau skaičiavimo resursų. Šio bandymo metu gauti DIS algoritmo rezultatai gali būti prilyginami *FlowNet2* algoritmu gautiems rezultatams, tačiau DIS algoritmas žymiai spartesnis.

Iš pradinio pasirinktų optinio srauto algoritmų palyginimo galima daryti išvadą, kad pasirinktais optinio srauto algoritmais gauti rezultatai ženkliai skiriasi (skirtumas matosi net vizualiai analizuojant optinį srautą), o tinkamas optinio srauto algoritmas konkrečiam uždaviniui priklauso nuo uždavinio sąlygų ir reikalavimų šiam taikymui. Šio darbo metu siekiama atskleisti judesio analizės algoritmų taikymo realiojo laiko uždaviniuose galimybes, todėl tolimesniuose bandymuose jis nebus naudojamas. *Farneback* ir DIS algoritmai pakankamai spartūs, kad skaičiuotų optinį srautą realiuoju laiku, tačiau norint padaryti objektyvias išvadas, jų sparta ir tikslumas nustatyti kitame bandyme, kurio rezultatai pateikti 4.2 poskyryje.

4.2. Judesio aptikimo ir optinio srauto algoritmų spartos tyrimo rezultatai

Vienas iš svarbiausių reikalavimų, keliamų judesio analizės algoritmams, norint juos taikyti realiojo laiko uždaviniuose, yra sparta. Dėl šios priežasties nuspręsta įgyvendinti pasirinktus judesio aptikimo ir optinio srauto algoritmus tokiems taikymams būdingoje aparatinėje įrangoje – dviejuose skirtinguose ARM architektūros procesoriuose. Pasirinkti mikroprocesoriai naudojami populiariose platformose ir yra tinkami siekiant mažesnės kainos ir mažiau energijos vartojančias, tačiau vaizdų analizei pakankamai našias sistemas.

Pirmoji bandymams naudota sistema – bandymams ir prototipams dažnai naudojamas *Raspberry Pi 3* įterptinis kompiuteris, jame skaičiavimus atlieka keturių branduolių ARM A53 architektūros centrinis procesorinis įrenginys, veikiantis 1,2 GHz taktiniu dažniu.

Antroji bandymams pasirinkta sistema – *NVIDIA Jetson Nano* modulis, skirtas intelektualiosioms sistemoms įgyvendinti įterptinėse sistemose. Šiame modulyje naudojama vienlūstė sistema, kurią sudaro keturių branduolių ARM A57 architektūros centrinis procesorinis įrenginys, veikiantis 1,43 GHz taktiniu dažniu, ir 128 branduolių *Maxwell* architektūros grafinis procesorinis įrenginys.

Bandymams naudoti dviejų, raiškų vaizdo įrašai, palyginti 7 judesio aptikimo ir 7 optinio srauto algoritmai. Bandymų metu kiekvienas algoritmas pritaikytas visam vaizdo įrašui ir suskaičiuota vidutinė vieno kadro apdorojimo trukmė ir ją atitinkantis kadrų dažnis.

Judesio aptikimo algoritmų, įgyvendintų *Raspberry Pi 3* ir *Jetson Nano* sistemose, spartos matavimų rezultatai pateikti atitinkamai 4.1 ir 4.2 lentelėse. Šiose lentelėse pateikta kiekvieno algoritmo vidutinė vieno kadro apdorojimo trukmė esant skirtingoms raiškoms ir šią trukmę atitinkantis maksimalus kadrų apdorojimo dažnis.

4.1 lentelė. Judesio aptikimo algoritmų spartos matavimo rezultatai (*Raspberry Pi 3*)

Algoritmas	160×120 raiška		320×240 raiška	
	T, ms	f, Hz	T, ms	f, Hz
CNT	0,629	1588,882	1,765	566,484
MOG	3,683	271,543	14,072	71,061
MOG2	4,849	206,247	18,357	54,474
KNN	4,980	200,788	18,464	54,16
GMG	6,421	160,223	23,024	43,433
GSOC	16,084	62,173	53,076	18,841
LSBP	22,254	44,936	75,487	13,247

4.2 lentelė. Judesio aptikimo algoritmų spartos matavimo rezultatai (*Jetson Nano*)

Algoritmas	160×120 raiška		320×240 raiška	
	T, ms	f, Hz	T, ms	f, Hz
CNT	0,235	4248,128	0,641	1559,168
MOG	1,334	749,640	5,060	197,635
MOG2	1,311	762,664	4,785	208,971
KNN	1,499	667,315	5,627	177,723
GMG	3,035	329,488	12,152	82,291
GSOC	4,438	225,334	16,878	59,249
LSBP	7,963	125,584	26,159	38,227

Iš 4.1 ir 4.2 lentelėse pateiktų matavimo rezultatų matyti, kad judesio aptikimo algoritmų vykdymo sparta ženkliai skiriasi ir priklauso nuo pasirinkto algoritmo, vaizdo įrašo raiškos ir algoritmą vykdančios aparatūros. Bandymams naudotų vaizdų taškų skaičius skiriasi keturis kartus ir visų algoritmų spartos esant atitinkamoms raiškoms santykis yra artimas keturiems. Tai parodo artimą tiesinei kadro taškų kiekio ir jo apdorojimo trukmės priklausomybę. Tai leidžia pagal naudojamos aparatinės įrangos galimybes nustatyti maksimalią realiuoju laiku apdoroti galimą vaizdų raišką.

Verta pastebėti, kad CNT algoritmas nuo kitų išsiskiria savo sparta jį vykdant *Raspberry Pi 3* sistemoje 320×240 raiškos vaizdų srautui galimas 566 Hz viršijantis kadrų dažnis, *Jetson Nano* sistema pasiekiamas net 1559 Hz kadrų dažnis. Tai gerokai viršija daugumos realiojo laiko uždavinių reikalavimus, todėl šis algoritmas gali būti taikomas vaizdų pradiniam apdorojimui daugelyje vaizdų atpažinimo uždavinių. Pasirinkus praktikoje dažnai naudojamą 25 Hz kadrų dažnį vieno kadro periodas yra lygus 40 ms, taigi *Jetson Nano* sistemoje naudojant CNT algoritmą 25 Hz kadrų dažnio 320×240 raiškos vaizdų srautui judesiui aptikti sunaudojama tik 1,6 % turimo laiko biudžeto, todėl didžioji kadro periodo dalis gali būti panaudota norimam vaizdų atpažinimo uždaviniui vykdyti.

Svarbu paminėti, kad į bandymo metu matuotą kadro apdorojimo trukmę neįtraukta vaizdų skaitymo trukmė, nes ji priklauso nuo pasirinkto duomenų įvesties būdo ir norint sumažinti.

Optinio srauto algoritmų, įgyvendintų *Raspberry Pi 3* ir *Jetson Nano* sistemose, spartos matavimų rezultatai pateikti atitinkamai 4.3 ir 4.4 lentelėse.

4.3 lentelė. Optinio srauto algoritmų spartos matavimo rezultatai (*Raspberry Pi 3*)

Algoritmas	160×120 raiška		320×240 raiška	
	T, ms	f, Hz	T, ms	f, Hz
DIS	4,111	243,256	18,065	55,355
LK-SparseToDense	46,010	21,735	181,281	5,516
Farneback	93,963	10,642	416,349	2,402
PCA	108,807	9,191	457,932	2,184
DeepFlow	738,678	1,354	2985,137	0,335
SimpleFlow	768,740	1,301	4928,000	0,203
TVL1	3549,35	0,282	13894,610	0,072

4.4 lentelė. Optinio srauto algoritmų spartos matavimo rezultatai (*Jetson Nano*)

Algoritmas	160×120 raiška		320×240 raiška	
	T, ms	f, Hz	T, ms	f, Hz
DIS	1,888	529,527	6,990	143,071
LK-SparseToDense	16,107	62,085	64,990	15,387
Farneback	40,182	24,887	164,512	6,079
PCA	41,584	24,048	172,874	5,785
DeepFlow	229,067	4,366	911,475	1,097
SimpleFlow	280,089	3,570	1113,077	0,898
TVL1	326,370	3,064	1136,842	0,880

Iš 4.3 ir 4.4 lentelėse pateiktų optinio srauto algoritmų spartos matavimų rezultatų matyti, kad optinio srauto skaičiavimas reikalauja daugiau laiko nei judesio aptikimo algoritmų. Taip pat matyti,

kad dalies algoritmų sparta, juos įgyvendinus pasirinktose įterptinėse sistemose, nėra pakankama realiojo laiko uždaviniams. TVL1, *SimpleFlow* ir *DeepFlow* algoritmais skaičiuojant 320×240 raiškos optinį srautą *Raspberry Pi 3* sistemoje, kadrų dažnis nesiekė 0,5 Hz. Naudojant *Raspberry Pi 3* sistemą tik trijų algoritmų maksimalus dažnis viršijo 10 Hz esant 160×120 taškų raiškai ir tik vienas – esant didesnei raiškai. Naudojant *Jetson Nano* sistemą pasiekta geresnių rezultatų, tačiau pasirinkus 320×240 taškų raišką tik DIS algoritmo vykdymas vienam kadru vidutiniškai truko mažiau nei anksčiau minėta praktikoje dažnai sutinkama 40 ms riba.

Iš gautų rezultatų savo sparta išsiskiria DIS algoritmas ir yra potencialus taikyti šiame darbe pasiūlytame objektų aptikimo ir sekimo algoritme, tačiau norint geriau atskleisti tiriamų algoritmų savybes atlikti optinio srauto algoritmų tikslumo įvertinimo eksperimentai, kurių rezultatai pateikti 4.3 poskyryje.

4.3. Optinio srauto algoritmų tikslumo tyrimo rezultatai

Optinio srauto algoritmų tikslumo matavimo metu aštuoniems *Middlebury* duomenų rinkinio pavyzdžiams buvo apskaičiuota vidutinė pasirinktų algoritmų klaida pagal EPE kriterijų (angl. *End Point Error*). Optinio srauto skaičiavimo tikslumas pasirinktais algoritmais nepriklauso nuo aparatinės įrangos, kuria vykdomas algoritmas, todėl rezultatai sutampa ir pateikti vienoje lentelėje. 4.5 lentelėje pateikti matavimų rezultatai, lentelėje raidėmis nuo a iki h pažymėti *Middlebury* duomenų rinkinio pavyzdžiai pagal tokį eiliškumą: a – *Dimetrodon*, b – *Grove2*, c – *Grove3*, d – *Hydrangea*, e – *RubberWhale*, f – *Urban2*, g – *Urban3*, h – *Venus*.

4.5 lentelė. Optinio srauto tikslumo matavimo rezultatai

Algoritmas	Vidutinė EPE klaida								
	a	b	c	d	e	f	g	h	Vidurkis
DeepFlow	0,08	0,17	0,70	0,17	0,12	0,37	0,46	0,28	0,29
LK-SparseToDense	0,17	0,27	0,92	0,23	0,18	0,69	0,69	0,34	0,44
DIS	0,15	0,32	0,85	0,25	0,22	0,64	2,02	0,40	0,61
PCA	0,19	0,36	1,14	0,36	0,28	0,90	1,20	0,51	0,62
TVL1	0,18	0,16	0,76	0,19	0,16	3,58	2,07	0,30	0,92
Farneback	0,86	0,49	1,26	0,49	0,33	1,51	3,09	1,44	1,18
SimpleFlow	1,57	2,18	2,89	3,23	1,35	6,86	4,68	2,93	3,21

Iš 4.5 lentelėje pateiktų rezultatų matyti, kad optinis srautas tiksliausiai apskaičiuotas naudojant *DeepFlow* algoritmą, taip pat matyti, kad lėčiausi algoritmai nėra tiksliausi, o tai ypač svarbu pasirenkant optinio srauto algoritmą taikyti realiojo laiko uždaviniams. Svarbu paminėti, kad DIS algoritmas yra sparčiausias iš tirtų algoritmų ir sparta gerokai lenkia visus kitus bandymuose tirtus algoritmus, tačiau pagal tikslumą jis yra trečioje vietoje iš septynių algoritmų. Dėl minėtų DIS

algoritmo savybių jis pasirinktas pasiūlyto objektų aptikimo ir sekimo algoritmo tyrimo bandymams. Šio tyrimo rezultatai pateikti 4.4 poskyryje.

4.4. Pasiūlyto objektų aptikimo ir sekimo algoritmo tyrimo rezultatai

Šiame poskyryje pateikti pasiūlyto objektų aptikimo ir sekimo algoritmo tyrimo rezultatai. Pasiūlyto algoritmo idėja paremta prielaida, kad optinio srauto skaičiavimas užtrunka trumpiau nei objektų aptikimas. Pagal 4.2 poskyryje pateiktus rezultatus matyti, kad yra tokių optinio srauto algoritmų, kuriuos įgyvendinus net įterptinėse sistemose jų sparta viršija įprastus realiojo laiko uždavinių reikalavimus, todėl, derinant šį algoritmą su objektų aptikimo algoritmu, galimas bendro algoritmo veikimas realiuoju laiku. Šių bandymų metu siekiama įvertinti galimybę pasiūlytu algoritmu aukojant dalį tikslumo pasiekti žymiai didesnę algoritmo veikimo spartą. Algoritmui įvertinti naudota MOT15 [40] pasiūlyta metodika.

Pasiūlytas algoritmas remiasi objektų sekimo pagal objektų aptikimą principu, tačiau pakeičiant dalies kadrų objektų aptikimo etapą spartesniu aptiktų objektų padėties atnaujinimu pagal optinį srautą. Prieš įgyvendinant pasiūlytą algoritmą svarbu suprasti kaip objektų sekimo tikslumas priklauso nuo minėtų būdų pakeistų objektų aptikimo etapų kiekio. Siekiant išsiaiškinti sukurto objektų sekimo algoritmo tikslumo priklausomybę nuo kadrų, kuriuose pakeičiamas objektų aptikimo etapas, kiekio, atlikti eksperimentiniai bandymai. Jų rezultatai pateikti 4.4.1 skirsnyje.

4.4.1. Objektų sekimo tikslumo priklausomybė nuo objektų aptikimo dažnio

Įgyvendinant pasiūlytą objektų aptikimo ir sekimo algoritmą galima pasirinkti kokiu periodiškumu bus naudojamas objektų aptikimo algoritmas ir kokiam kiekiui kadrų po aptikimo bus naudojamas objektų padėčių atnaujinimas pagal optinį srautą. Atlikto eksperimentinio tyrimo metu pasiūlytas algoritmas įgyvendintas atliekant vieną aptikimą per N kadrų, kur N kinta nuo 1 iki 10.

Visomis tirtomis algoritmo konfigūracijomis atliktas objektų sekimo rezultatų vertinimas MOT15 objektų sekimo algoritmų vertinimo kriterijus kiekvienai iš vienuolikos pasirinktų sekų atskirai ir apskaičiuotas bendras rezultatas visam duomenų rinkiniui. Tyrimuose, kurių rezultatai pateikti 4.4.1 skirsnyje, naudoti tikrieji objektų aptikimo rezultatai (atitinka idealų objektų aptikimą), jie pateikti kartu su duomenų rinkiniu. Taip siekta nustatyti objektų sekimo tikslumo priklausomybę būtent nuo kadrų, kuriuose objektų aptikimo etapas pakeistas optinio srauto skaičiavimu, skaičiaus. 4.6 lentelėje pateikti objektų sekimo įvertinimo rezultatai, kai objektų aptikimas vykdomas kiekvienam kadrui ir optinio srauto algoritmas nenaudojamas. Tai atitinka objektų sekimą naudojant idealų objektų aptikimo algoritmą ir SORT objektų sekimo algoritmą. Šie rezultatai naudojami kaip etaloniniai ir geriausia, galimi pasiekti pasiūlytu algoritmu.

4.6 lentelė. Objektų sekimo algoritmo rezultatai, objektus aptinkant kiekviename kadre

Vaizdo įrašo pavadinimas	R	P	FAR	GT	MT	PT	ML	FP	FN	IDS	FM	MOT A	MOT P
<i>TUD-Stadtmitte</i>	99,2	100	0,00	10	10	0	0	0	9	0	0	99,22	97,97
<i>TUD-Campus</i>	98,3	100	0,00	8	8	0	0	0	6	0	0	98,33	92,46
<i>PETS09-S2L1</i>	98,9	96,2	0,22	19	19	0	0	174	48	0	2	95,04	94,71
<i>ETH-Bahnhof</i>	93,7	71,6	2,01	171	121	38	12	2014	343	9	116	56,31	90,00
<i>ETH-Sunnyday</i>	96,3	98,0	0,10	30	27	3	0	37	69	0	5	94,29	90,92
<i>ETH-Pedcross2</i>	94,3	93,1	0,52	133	88	29	16	439	354	9	7	87,19	92,67
<i>ADL-Rundle-6</i>	98,7	100	0,00	24	24	0	0	0	63	1	1	98,72	97,54
<i>ADL-Rundle-8</i>	99,2	100	0,00	28	27	1	0	0	54	0	0	99,20	95,85
<i>KITTI-13</i>	74,8	83,1	0,34	42	14	20	8	116	192	8	6	58,53	83,96
<i>KITTI-17</i>	98,5	87,1	0,69	9	9	0	0	100	10	1	0	83,75	90,37
<i>Venice-2</i>	99,3	100	0,00	26	26	0	0	0	45	0	0	99,37	98,42
Iš viso:	97,0	93,1	0,52	500	373	91	36	2880	1193	28	137	89,72	94,69

4.6 lentelėje R atitinka anksčiau aptartą *Recall*, o P – *Precision*. Kaip matyti iš apibendrintų rezultatų (paskutinė eilutė), objektų sekimas gana tikslus, iš 500 teisingų objektų sekimo trajektorijų didžiąja dalimi teisingai atliktas 373 objektų sekimas (MT), dalinai teisingai – 91 (PT) ir didžiąją dalimi neteisingai atliktas tik 36 objektų sekimas (ML). MOTA ir MOTP atitinkamai lygūs 89,72 % ir 94,69 %. Šie rezultatai gali būti naudojami kaip atskaitos taškas kitų bandymų metu. 4.7 lentelėje pateikti bandymo, kai kas antrame kadre objektų aptikimo etapas pakeistas optinio srauto skaičiavimu ir objektų padėties atnaujinimu pagal optinį srautą.

4.7 lentelė. Objektų sekimo algoritmo rezultatai, objektus aptinkant 1 iš 2 kadru

Vaizdo įrašo pavadinimas	R	P	FAR	GT	MT	PT	ML	FP	FN	IDS	FM	MOT A	MOT P
<i>TUD-Stadtmitte</i>	98,9	99,9	0,01	10	10	0	0	1	12	0	0	98,88	96,06
<i>TUD-Campus</i>	98,0	99,4	0,03	8	8	0	0	2	7	0	0	97,49	88,58
<i>PETS09-S2L1</i>	98,8	96,1	0,23	19	19	0	0	181	54	0	3	94,75	90,87
<i>ETH-Bahnhof</i>	92,6	70,4	2,10	171	116	42	13	2102	403	10	117	53,55	86,36
<i>ETH-Sunnyday</i>	95,7	97,6	0,12	30	26	4	0	43	80	0	5	93,38	86,80
<i>ETH-Pedcross2</i>	92,3	91,8	0,62	133	78	37	18	516	481	31	49	83,59	89,22
<i>ADL-Rundle-6</i>	98,5	99,9	0,01	24	24	0	0	6	76	1	1	98,34	95,61
<i>ADL-Rundle-8</i>	98,9	99,9	0,01	28	27	1	0	9	75	0	1	98,76	93,77
<i>KITTI-13</i>	71,0	81,6	0,36	42	13	24	5	122	221	7	6	54,07	79,26
<i>KITTI-17</i>	95,5	89,4	0,53	9	9	0	0	77	31	1	7	84,04	85,84
<i>Venice-2</i>	98,9	99,9	0,01	26	26	0	0	7	79	0	6	98,80	96,84
Iš viso:	96,2	92,6	0,56	500	356	108	36	3066	1519	50	195	88,38	91,94

4.7 lentelėje R yra *Recall*, o P – *Precision*. Pagal 4.7 lentelėje pateiktus rezultatus, dvigubai sumažinę objektų aptikimo algoritmo taikymo dažnį, tik 1,34 % nukrito MOTA rezultatas ir 2,75 % nukrito MOTP rezultatas.

Padidinus objektų aptikimo algoritmo taikymo periodą iki 5 kadrų gauti 4.8 lentelėje pateikti rezultatai.

4.8 lentelė. Objektų sekimo algoritmo rezultatai, objektus aptinkant 1 iš 5 kadrų

Vaizdo įrašo pavadinimas	R	P	FAR	GT	MT	PT	ML	FP	FN	IDS	FM	MOT A	MOT P
TUD-Stadtmitte	97,1	98,1	0,1	10	10	0	0	22	34	7	9	94,55	90,10
TUD-Campus	91,6	95,6	0,2	8	6	2	0	15	30	3	5	86,63	82,22
PETS09-S2L1	88,3	90,0	0,5	19	15	4	0	437	525	71	123	76,92	82,90
ETH-Bahnhof	88,8	67,5	2,3	171	102	41	28	2317	605	13	125	45,80	81,17
ETH-Sunnyday	92,6	94,8	0,3	30	22	8	0	94	137	0	8	87,57	82,10
ETH-Pedcross2	81,2	86,1	1,0	133	41	63	29	821	1178	81	103	66,79	84,23
ADL-Rundle-6	94,9	97,4	0,2	24	24	0	0	127	254	46	58	91,48	90,47
ADL-Rundle-8	97,1	98,6	0,1	28	27	1	0	96	199	13	30	95,46	87,45
KITTI-13	57,3	67,6	0,6	42	5	29	8	209	325	11	23	28,48	69,96
KITTI-17	83,5	81,4	0,9	9	6	3	0	130	113	10	14	62,96	78,66
Venice-2	96,2	98,6	0,2	26	26	0	0	98	271	31	55	94,40	93,01
Iš viso:	90,8	89,2	0,8	500	284	151	65	4366	3671	286	553	79,14	86,55

4.8 lentelėje R yra *Recall*, o P – *Precision*. Iš 4.8 lentelėje pateiktų rezultatų matyti, kad objektus aptinkant viename kadre iš penkių, gaunamas 79,14 % MOTA ir 86,55 % MOTP rezultatas. Tai parodo, kad net penkis kartus sumažinus objektų aptikimo etapo taikymą išlaikoma didžioji dalis objektų sekimo tikslumo. Vis dar 284 iš 500 sekimo trajektorijų yra beveik ar visiškai teisingos (MT stulpelis). Šie rezultatai patvirtina idėją, kad objektų padėties atnaujinimą atliekant pagal suskaičiuotą optinį srautą, galima gauti gana tikslų objektų padėties nustatymą.

Padidinus objektų aptikimo algoritmo taikymo periodą iki 10 kadrų (aptikimas atliekamas vieną kartą per dešimt kadrų), gauti rezultatai pateikti 4.9 lentelėje. 4.9 lentelėje R yra *Recall*, o P – *Precision*. Iš 4.9 lentelėje pateiktų rezultatų matyti, kad dešimt kartų sumažinus objektų aptikimo algoritmo taikymą, MOTA nukrito iki 59,74 %, o MOTP iki 81,63 %. Beveik arba visiškai teisingai atliktas 186 iš 500 objektų sekimas, o tai jau mažiau nei pusė. Objektų aptikimą atliekant viename iš dešimties kadrų, objektų sekimo tikslumas ženkliai sumažėjo, todėl nuspręsta toliau objektų aptikimo algoritmo periodo nedidinti.

4.9 lentelė. Objektų sekimo algoritmo rezultatai, objektus aptinkant 1 iš 10 kadro

Vaizdo įrašo pavadinimas	R	P	FAR	GT	MT	PT	ML	FP	FN	IDS	FM	MOT A	MOT P
<i>TUD-Stadtmitte</i>	87,6	90,9	0,6	10	8	2	0	102	143	20	20	77,08	85,23
<i>TUD-Campus</i>	80,8	83,1	0,8	8	4	4	0	59	69	9	15	61,84	76,03
<i>PETS09-S2L1</i>	64,5	70,0	1,6	19	3	15	1	1235	1589	144	226	33,69	76,90
<i>ETH-Bahnhof</i>	79,4	60,3	2,8	171	73	65	33	2828	1113	37	188	26,54	75,65
<i>ETH-Sunnyday</i>	85,5	88,8	0,6	30	12	15	3	200	269	2	22	74,65	77,62
<i>ETH-Pedcross2</i>	70,1	73,0	1,9	133	23	71	39	1624	1872	108	161	42,46	80,05
<i>ADL-Rundle-6</i>	87,6	91,3	0,8	24	17	7	0	420	623	86	95	77,46	83,79
<i>ADL-Rundle-8</i>	91,1	93,4	0,7	28	22	6	0	440	601	45	96	83,99	81,77
<i>KITTI-13</i>	29,9	38,8	1,1	42	0	18	24	359	534	11	22	18,64	63,06
<i>KITTI-17</i>	64,0	62,9	1,8	9	2	6	1	258	246	19	30	23,43	72,71
<i>Venice-2</i>	91,5	95,2	0,5	26	22	4	0	328	609	62	70	86,01	89,04
Iš viso:	80,8	80,4	1,4	500	186	213	101	7853	7668	543	945	59,74	81,63

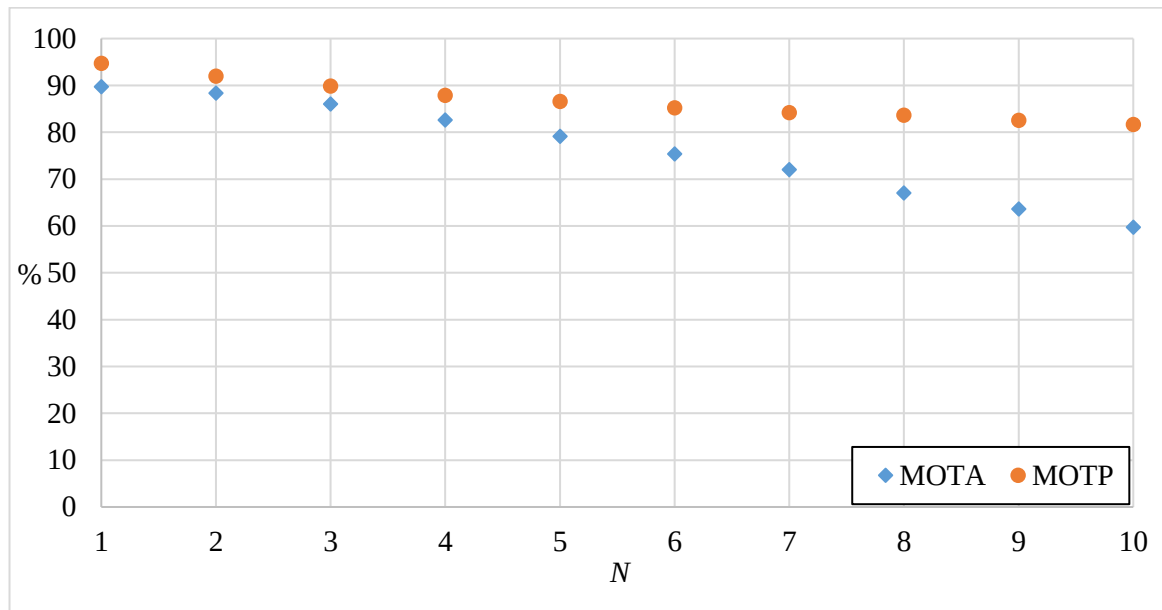
Eksperimentinio tyrimo metu iš viso atlikta 10 tokio tipo bandymų, kuriuose objektų aptikimas atliekamas viename kadre iš N skaičiaus kadro, kai N kinta nuo 1 iki 10. 4.10 lentelėje pateikti apibendrinti šių tyrimų rezultatai (pateiktas tik apibendrintas viso duomenų rinkinio rezultatas).

4.10 lentelė. Objektų sekimo tikslumo priklausomybė nuo aptikimo dažnio

Aptikimas vykdomas 1 kadre iš:	R	P	FAR	GT	MT	PT	ML	FP	FN	IDS	FM	MOT A	MOT P
1	97,0	93,1	0,52	500	373	91	36	2880	1193	28	137	89,72	94,69
2	96,2	92,6	0,56	500	356	108	36	3066	1519	50	195	88,38	91,94
3	94,5	92,1	0,59	500	330	127	43	3247	2206	134	311	86,00	89,86
4	93,1	90,4	0,72	500	312	141	47	3946	2763	232	439	82,61	87,90
5	90,8	89,2	0,79	500	284	151	65	4366	3671	286	553	79,14	86,55
6	88,9	87,6	0,92	500	270	165	65	5043	4424	373	702	75,34	85,19
7	87,0	86,2	1,01	500	242	181	77	5571	5168	416	771	72,05	84,21
8	84,9	83,6	1,21	500	225	194	81	6647	6021	496	859	67,01	83,6
9	82,7	82,3	1,29	500	218	192	90	7085	6914	532	910	63,59	82,53
10	80,8	80,4	1,43	500	186	213	101	7853	7668	543	945	59,74	81,63

4.10 lentelėje R yra *Recall*, o P – *Precision*. 4.10 lentelė geriausiai apibūdina objektų sekimo tikslumo priklausomybę nuo kadro, kuriuose objektų aptikimas pakeistas jų sekimu pagal optinį srautą, skaičiaus. Kaip matyti iš MT stulpelio (beveik arba visiškai teisingai atliktų sekimų skaičius), net sumažinus objektų aptikimo dažnį iki vienos šeštosios, didžiosios dalies objektų sekimas atliktas didžiąją dalimi teisingai, MOTP yra daugiau nei 75 %, o MOTP daugiau nei 85 %.

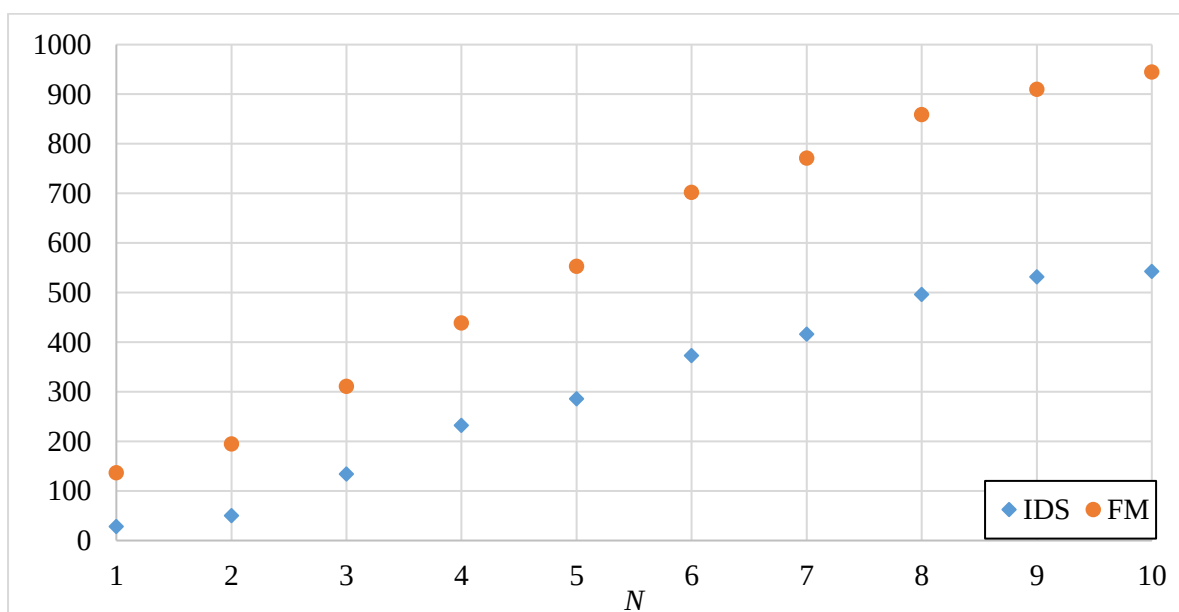
Vaizdžiai MOTA ir MOTP priklausomybė nuo objektų aptikimo dažnio iliustruota 4.10 pav.



4.10 pav. MOTA ir MOTP, kai objektų aptikimas vykdomas viename iš N kadrų

Skaičius N atitinka kadrų skaičių, kuriam atliekamas vienas objektų aptikimas. Iš 4.10 pav. matyti, kad MOTA įvertis, kuris įvertina sekimo tarp kadrų tikslumą, labiau priklauso nuo N ir kai N pasiekia 5, pradeda staigiau mažėti. Tuo tarpu MOTP įvertis, kuris apibūdina objektų padėties nustatymo tikslumą, didėjant N mažėja ne taip staigiai.

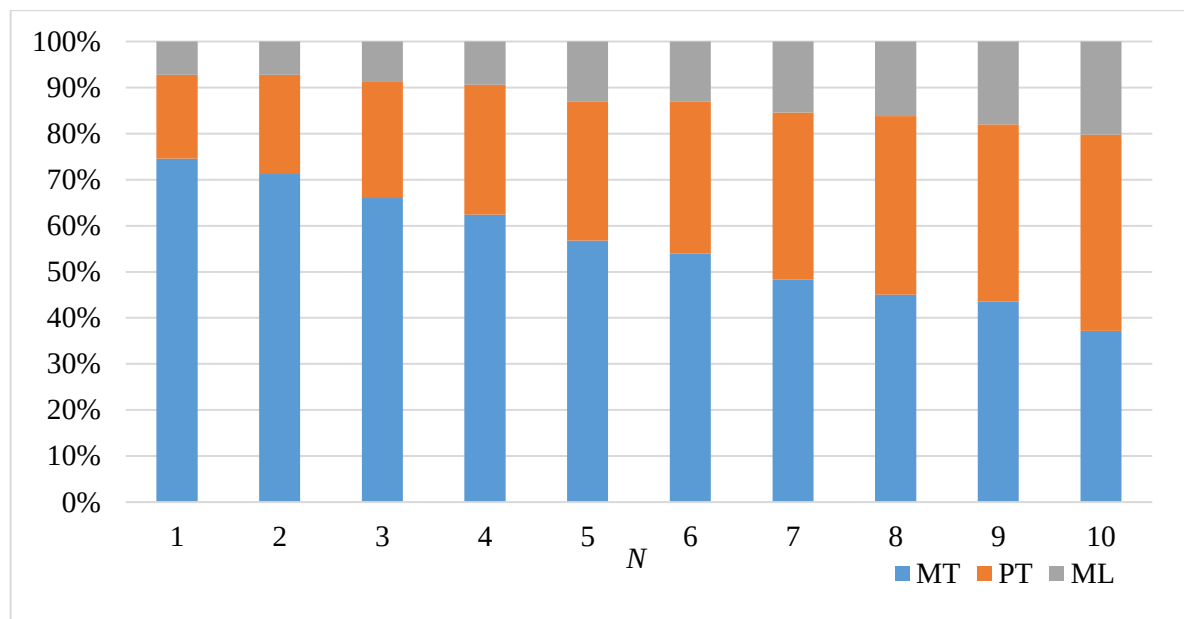
4.11 pateiktas objektų identifikatorių pokyčių (IDS) ir trajektorijų trūkių (FM) priklausomybės nuo N grafikas.



4.11 pav. IDS ir FM, kai objektų aptikimas vykdomas viename iš N kadrų kadrų

Iš 4.11 matomas pastovus objektų identifikatorių pokyčių ir objektų trajektorijų trūkių skaičiaus didėjimas, didinant N . Tai parodo, kad, siekiant tikslaus objektų sekimo, negalima parinkti per didelio N . Suprantama, kad kai vaizde atsiranda naujas objektas, jo trajektorija nėra sekma tol, kol neįvykdomas objektų aptikimas, ir šiam objektui priskiriamas identifikatorius.

4.12 pav. pateiktas visų bandymams naudotų sekamų objektų trajektorijų sekimo rezultatų pasiskirstymas į tris kategorijas: beveik arba visiškai tiksliai atliktas sekimas, iš dalies teisingai atliktas sekimas ir didžiąją dalimi klaidingai atliktas sekimas.



4.12 pav. MT, PT ir ML pasiskirstymas, kai objektai aptinkami viename iš N kadru

Iš 4.12 pav. matyti, kad bent iš dalies teisingai atliktas apie 80 % objektų sekimas (PT dalies viršutinė riba), net kai N lygus 10. Parametras N gali būti parinktas pagal konkretaus uždavinio tipą ir jam keliamus tikslumo reikalavimus ar naudojamos aparatūros galimybes.

Šiame skirsnyje pateikti rezultatai leidžia suprasti kiek objektų sekimo tikslumo aukojama dalyje kadru objektų aptikimą keičiant jų padėties atnaujinimu pagal optinį srautą, tačiau labai svarbu žinoti kokį spartos padidėjimą duoda toks tikslumo paaukojimas. Minėtas algoritmo spartos padidėjimas priklauso nuo pasirinktos aparatūros ir optinio srauto algoritmo, todėl nuspręsta atlikti pasiūlyto objektų aptikimo ir sekimo algoritmo spartos priklausomybės nuo objektų aptikimo dažnio tyrimą, įgyvendinus algoritmą specializuotoje aparatinėje įrangoje. Objektų sekimo spartos priklausomybės nuo objektų aptikimo dažnio tyrimo rezultatai pateikti 4.4.2 skirsnyje.

4.4.2. Objektų sekimo spartos priklausomybė nuo objektų aptikimo dažnio

Šiame skirsnyje pateiki objektų sekimo spartos priklausomybės nuo objektų aptikimo dažnio tyrimo rezultatai. Šiuo tyrimu siekta įvertinti gaunamą objektų sekimo algoritmo spartos padidėjimą, pritaikant optinio srauto algoritmą.

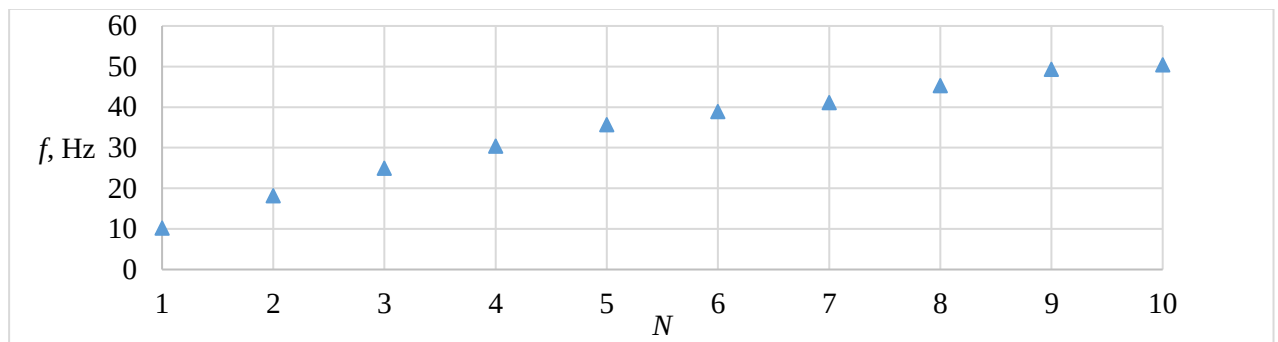
Algoritmas įgyvendintas specializuotoje *Jetson Nano* sistemoje, kurioje objektų aptikimo etapui naudojamas aparatinis spartinimas grafiniu procesoriniu įrenginiu. Tai leidžia tokioje nedidelėje ir mažai energijos vartojančioje sistemoje (didžiausia momentinė galia – 5 W) realiuoju laiku vykdyti daugelio objektų aptikimo ir sekimo algoritmą.

Bandymų metu vykdytas objektų aptikimo ir sekimo algoritmas objektų aptikimo etapą vykdamas viename iš N kadrų, esant N vertėms nuo 1 iki 10. Bandymams naudotas 320×240 taškų raiškos vaizdo įrašas. Bandymų rezultatai pateikti 4.11 lentelėje. Lentelėje atskirai pateikta kiekvieno algoritmo etapo vykdymo trukmė ir atitinkamas maksimalus algoritmo vykdymo kadrų dažnis.

4.11 lentelė. Objektų sekimo spartos priklausomybė nuo aptikimo dažnio

Aptikimas vykdomas 1 kadre iš:	Aptikimo trukmė, ms	Optinio srauto skaičiavimo trukmė, ms	Objektų padėties atnaujinimo trukmė, ms	Objektų sekimo trukmė, ms	Maksimalus kadrų dažnis, Hz
1	91,27	0,00	0,00	7,03	10,17
2	91,54	2,69	1,21	7,04	18,11
3	91,88	2,54	1,15	7,06	24,91
4	92,30	2,43	1,18	6,92	30,37
5	92,26	2,51	1,11	6,54	35,62
6	92,27	2,40	1,15	6,96	38,85
7	92,13	2,55	1,18	7,21	41,09
8	92,48	2,39	1,15	6,89	45,23
9	92,40	2,42	1,16	6,84	49,30
10	92,78	2,34	1,20	7,28	50,41

Iš 4.11 lentelėje pateiktų rezultatų matyti, kad objektų aptikimą atliekant vienam ir trijų kadrų, gaunamas daugeliui realiojo laiko uždavinių būdingas kadrų dažnis (beveik 25 Hz). Toliau didinant kadrų, kuriuose objektams sekti naudojamas optinis srautas, skaičių pasiektas net iki 50 Hz kadrų dažnis. Tai įrodo, kad pasiūlytas algoritmas gali veikti realiuoju laiku net tokiose nedidelėse sistemose, kaip *Jetson Nano*. Vaizdžiai rezultatus iliustruoja 4.13 pav.



4.13 pav. Maksimalus objektų sekimo dažnis, kai objektai aptinkami viename iš N kadrų

4.5. Trumpas skyriaus apibendrinimas

Šiame skyriuje atlikti eksperimentiniai judesio analizės algoritmų tyrimai. Atlikto pradinio optinio srauto algoritmų palyginimo metu atskleistos išskirtinės skirtingų tipų optinio srauto algoritmų savybės. Sąsukos dirbtinių neuronų tinklais grįstas *FlowNet2* algoritmas geriausiai apskaičiuoja optinį srautą, tačiau reikalauja ypatingai našios aparatūros ir nėra tinkamas daugelyje realiojo laiko taikymo atvejų. Iš palygintų algoritmų savo sparta ir patenkinamu tikslumu išsiskyrė DIS optinio srauto algoritmas, kuris pasirinktas naudoti darbo metu pasiūlyto objektų sekimo algoritmui įgyvendinti.

Antrojo eksperimentinio tyrimo metu įvertinta pasirinktų optinio srauto ir judesio aptikimo algoritmų sparta. Iš bandymų rezultatų nustatyta, kad šių algoritmų sparta labai priklauso nuo algoritmo ir kinta plačiame intervale.

Trečiojo eksperimentinio tyrimo metu įvertinus pasirinktų optinio srauto algoritmų tikslumą pastebėta, kad algoritmų vykdymo trukmė nėra proporcinga algoritmo tikslumui ir padaryta išvada, kad renkantis optinio srauto algoritmą realiojo laiko uždaviniams, svarbu atsižvelgti į algoritmo spartos ir tikslumo santykį.

Baigiamajame darbe pasiūlytas objektų aptikimo ir sekimo algoritmas, kuriame algoritmo veikimui paspartinti dalyje kadrų objektų aptikimo etapas pakeistas optinio srauto skaičiavimu ir objektų padėčių atnaujinimu pagal suskaičiuotą optinį srautą. Ketvirtuoju eksperimentiniu tyrimu bandyta išsiaiškinti, kaip nuo taip pakeistų kadrų apdorojimo skaičiaus priklauso algoritmo tikslumas ir veikimo sparta. Bandymais patvirtinta, kad pasiūlytu būdu galima žymiai padidinti algoritmo veikimo spartą. Įgyvendinus algoritmą *Jetson Nano* sistemoje pasiektas iki 50 Hz kadrų dažnis (objektų aptikimas vykdomas viename kadre iš 10). Dar vienas pasiūlyto algoritmo privalumas – galimybė rinktis tikslesnį objektų aptikimo algoritmą išlaikant veikimo realiojo laiko galimybę.

APIBENDRINIMAS. IŠVADOS

Baigiamojo darbo metu atliktas rega grįstų judesio analizės algoritmų tyrimas. Tyrimo rezultatų pateikimas šiame aiškinamajame rašte padalintas į keturias dalis, kurios pateiktos atskiruose skyriuose.

Pirmajame skyriuje pateikti rega grįstų judesio analizės algoritmų analitinės apžvalgos rezultatai. Atliktos apžvalgos metu atskleisti dažniausiai naudojamų judesio aptikimo ir optinio srauto algoritmų veikimo principai, atskleistos judesio analizės algoritmų taikymo realiojo laiko vaizdų atpažinimo uždaviniuose. Iš apžvalgos rezultatuose pateiktų pavyzdžių matyti, kad judesio analizės algoritmai yra sėkmingai pritaikyti ne viename vaizdų atpažinimo uždavinyje. Iš apžvelgtų judesio analizės algoritmų taikymo pavyzdžių išsiskiria jų taikymas veiksmų atpažinimo, judančių objektų aptikimo ir sekimo bei vaizdų segmentavimo uždaviniuose. Šiuose uždaviniuose pavyko pritaikyti judesio analizės algoritmais išgautas judesio savybes ir taip padidinti tikslumą.

Antrajame skyriuje pateikti eksperimentiniams tyrimams pasirinkti judesio aptikimo ir optinio srauto algoritmai. Pasirinkti tokie algoritmai, kuriuos pavyktų įgyvendinti naudojant įvairaus tipo ir galimybių aparatinę įrangą. Antrajame skyriuje taip pat pasiūlytas objektų aptikimo ir sekimo algoritmas, kuriame pritaikytas spartus optinio srauto algoritmas. Optinio srauto skaičiavimas šiame algoritme leidžia praleisti santykinai ilgai trunkantį objektų aptikimo etapą dalyje vaizdų srauto įrašų atnaujinant jau aptiktų objektų padėtį pagal iš optinio srauto apskaičiuotą objektų judėjimo kryptį ir greitį.

Trečiajame skyriuje pateikti duomenų judesio analizės algoritmų eksperimentiniam tyrimui pasirinkti duomenys. Paaiškinti reikalavimai, kuriuos turi atitikti pasirinkti duomenys. Pradiniam skirtingų tipų optinio srauto algoritmų palyginimui pasirinkti du skirtingi vaizdo įrašai, leidę atskleisti svarbias lygintų algoritmų savybes. Optinio srauto ir judesio aptikimo algoritmų spartai tirti paruošti du skirtingų raiškų vaizdo įrašai, pasirinktos raiškos atitinka realiojo laiko uždaviniams būdingas sąlygas ir tokiems uždaviniams naudojamos aparatinės įrangos galimybės. Optinio srauto tikslumui įvertinti pasirinktas *Middlebury* duomenų rinkinys, leidžiantis tiriama algoritmais suskaičiuotą optinį srautą palyginti su pateiktu visiškai teisingu optiniu srautu. Tikslumui vertinti pasirinktas judesio vektorių galų paklaidos kriterijus EPE (angl. *End Point Error*). Objektų sekimo algoritmui įvertinti pasirinktas tam plačiai naudojamas MOT15 (angl. *Multiple Object Tracking challenge 2015*) duomenų rinkinys, kuriam būdingas didelis duomenų kiekis ir duomenų įvairovė. Pasirinktą duomenų rinkinį sudaro vienuolika vaizdo įrašų ir juos atitinkančių visiškai teisingų objektų aptikimo ir objektų sekimo rezultatų. Teisingi objektų aptikimo ir sekimo rezultatai gali būti naudojami tiriamo algoritmo rezultatams įvertinti.

Ketvirtajame skyriuje pateikti atliktų eksperimentinių tyrimų rezultatai. Pradinis skirtingų tipų optinio srauto algoritmų palyginimas leido atskleisti išskirtines kiekvieno tipo algoritmų savybes. Nors sąsūkos dirbtinių neuronų tinklais grįstu *FlowNet2* algoritmu gautas geriausias rezultatas iš lygintų algoritmų, pastebėta, kad šiam algoritmui vykdyti reikia ypač našios aparatinės įrangos ir net naudojant tokią įrangą gaunama sparta netenkina daugelio realiojo laiko uždavinių reikalavimų. Šiuo darbu siekiama atskleisti judesio analizės algoritmų taikymo realiojo laiko uždaviniuose galimybes ir algoritmų veikimo sparta yra ypač svarbi, todėl tolimesniuose tyrimuose atsisakyta *FlowNet2* algoritmo. Judesio aptikimo ir optinio srauto algoritmų spartos tyrimas parodė, kad šie algoritmai gali būti įgyvendinti šiuolaikinėse įterptinėse sistemose ir net veikti realiuoju laiku. Po optinio srauto algoritmų tikslumo tyrimo tolimesniems bandymams išrinktas tikslumu ir sparta išsiskyręs DIS (angl. *Dense Inverse Search*) optinio srauto algoritmas. DIS algoritmas pritaikytas pasiūlytam objektų aptikimo ir sekimo algoritmui įgyvendinti. Atlikus pasiūlyto algoritmo tikslumo ir spartos priklausomybės nuo objektų aptikimo etapo vykdymo dažnio nustatyta, kad pasiūlytu algoritmu galima žymiai padidinti objektų aptikimo ir sekimo spartą aukojant nedidelę dalį tikslumo. Šiuo algoritmu nedidelėje ir mažai energijos vartojančioje *Jetson Nano* įterptinėje sistemoje pavyko įgyvendinti objektų aptikimo ir sekimo algoritmą, kuris veikia iki 50 Hz kadrų dažniu, naudojant 320×240 taškų raiškos vaizdų srautą. Tai leidžia kai kuriuos vaizdų atpažinimo uždavinius vykdyti prie duomenų šaltinio, pavyzdžiui, pačioje vaizdų srautą fiksuojančioje kameroje, išvengiant duomenų perdavimo vėlinimo, duomenų saugumui kylančių grėsmių ir padidinant vaizdų analizės sistemų autonomiškumą.

Iš atlikto tyrimo rezultatų padarytos šios išvados:

1. Regimojo judesio analizei dažniausiai naudojami optinio srauto algoritmai.
2. Taikant judesio analizės algoritmus vaizdų atpažinimo uždaviniuose pasiekiamas didesnis vaizdų atpažinimo tikslumas, didžiausia nauda pastebėta objektų aptikimo ir sekimo, vaizdų segmentavimo, veiksmų atpažinimo uždaviniuose.
3. Ypatingai geri optinio srauto skaičiavimo rezultatai pasiekiami taikant intelektualiuosius metodus, tačiau šiuo metu pasiūlyti optinio srauto skaičiuojantys dirbtiniai neuronų tinklai gana sudėtingi ir lėti.
4. DIS optinio srauto algoritmas išsiskiria ypač didele sparta ir aukštu tikslumu, todėl yra tinkamas taikyti realiojo laiko uždaviniuose.
5. DIS optinio srauto algoritmas gali būti sėkmingai taikomas pasiūlyto objektų aptikimo ir sekimo algoritmo spartai didinti.

LITERATŪRA

1. GONZALEZ, R. C.; WOODS, R. E. 1993. *Digital Image Processing*: vadovėlis. Boston, MA, USA: Addison-Wesley Longman Publishing Co. 185 p.
2. SINGLA, N.; MENDEL, J. M. 2014. Motion detection based on frame difference method, *International Journal of Information and Computing Technoogy*. 4: 1559–1565.
3. ZORAN, Z. 2004. Improved Adaptive Gaussian Mixture Model for Background Subtraction, *Proceedings - International Conference on Pattern Recognition*. 2: 28–31.
4. ZORAN, Z.; HEIJDEN, F. 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction, *Pattern Recognition Letters*. 27: 773–780.
5. LAKSHMI, V. A.; ANITHA, E.; SELVAKUMARI, S. 2014. Moving object detection by background subtraction, *Procedings of International Conference On Global Innovations in Coputing Technology*. 2: 2320–9801.
6. LUCAS, B. D.; KANADE, T. 1981. An iterative image registration technique with an application to stereo vision, *Procedings of the 7th International Joint Conference on Artificial Intelligence*. 674–679.
7. HORN, B. D.; SCHUNCK, B. 1981. Determining optical flow, *Artificial Intelligence*. 17: 185–203.
8. FARNEBACK G. 2003. Two-Frame Motion Estimation Based on Polynomial Expansion, *Image analysis*: 363–370.
9. KROEGER, T.; TIMOFTE, R.; DAI, D.; GOOL, L. V. 2016. Fast optical flow using dense inverse search, *Procedings of ECCV 2016*: 471–488.
10. WEINZAEPHEL, P.; JEROME, R.; HARCHAOUI, Z.; SCHMID, C. 2013. DeepFlow: large displacement optical flow with deep matching, *Procedings of IEEE International Conference on Computer Vision 2013*: 1385–1392.
11. FISCHER, P.; DOSOVITSKIY, A.; ILG, E.; HAUSSER, P. 2015. FlowNet: learning optical flow with convolutional cetworks, *Procedings of IEEE International Conference on Computer Vision 2015*: 2758–2767.
12. ILG, E.; MAYER, N.; SAIKIA, T.; KEUPER, M.; DOSOVITSKIY, A.; BROX, T. 2016. FlowNet 2.0: evolution of optical flow estimation with deep networks, *IEEE Conference on Computer Vision and Pattern Recognition 2017*: 1647–1656.
13. HUI, T. W.; TANG, X.; LOY, C. C. 2018. LiteFlowNet: a lightweight convolutional neural network for optical flow estimation, *IEEE Conference on Computer Vision and Pattern Recognition 2018*: 8981–8989.
14. SIMONYAN, K.; ZISSERMAN, A. 2014. Two-stream convolutional networks for action recognition in videos, *Advances in Neural Information Processing Systems*. 1: 568–576.
15. HAN, C.; WANG, C.; MEI, E. 2017. YOLO-based adaptive window two-stream convolutional neural network for video classification, *Stanford CS231n project reports*: 707–714.
16. JAIN, A.; TOMPSON, J.; LECUN, Y.; BREGLER, C. 2014. MoDeep: A deep learning framework using motion features for human pose estimation, *12th Asian Conference on Computer Vision*. 9004: 302–315.

17. HUANG, J.; ZOU, W.; ZHU, Z.; ZHU, J. 2018, *Optical flow based real-time moving object detection in unconstrained scenes* [interaktyvus]. [žiūrėta 2019 m. gegužės 22 d.]. Prieiga per internetą: <<https://arxiv.org/pdf/1807.04890.pdf>>
18. SIAM, M.; MAHGOUB, H.; ZAHARAN, M.; YOGAMANI, S.; JAGERSAND, M.; SALLAB, A. A. 2018. MODNet: Motion and Appearance based Moving Object Detection Network for Autonomous Driving, *IEEE International Conference on Intelligent Transportation Systems 2018*: 2859–2864.
19. BEAUPRE, D.; BILODEAU, G.; SAUNIER, N. 2018, *Improving Multiple Object Tracking with Optical Flow and Edge Preprocessing* [interaktyvus]. [žiūrėta 2019 m. gegužės 22 d.]. Prieiga per internetą: <<https://arxiv.org/pdf/1801.09646.pdf>>
20. KALE, K.; PAWAR, S.; DHULEKAR, P. 2015. Moving object tracking using optical flow and motion vector estimation, *4th International Conference on Reliability, Infocom Technologies and Optimization*: 1–6.
21. HUA, S.; KAPOOR, M.; ANASTASIU, D. 2018. Vehicle Tracking and Speed Estimation from Traffic Videos, *IEEE Conference on Computer Vision and Pattern Recognition 2018*: 1533–1537.
22. RASHED, H.; YOGAMANI, S.; EL-SALLAB, A.; KRIZEK, P.; EL-HELW, M. 2019, *Optical Flow augmented Semantic Segmentation networks for Automated Driving*. [interaktyvus]. [žiūrėta 2019 m. gegužės 22 d.]. Prieiga per internetą: <<https://arxiv.org/pdf/1901.07355.pdf>>
23. VISWANATH, P.; SISTU, G.; ILIE, M.; YOGAMANI, S.; HORGAN, J. 2018. Early Fusion of Dense Optical Flow with Image for Semantic Segmentation in Autonomous Driving, *Proceedings for the 26th AIAI Irish Conference on Artificial Intelligence and Cognitive Science*: 126–137.
24. *The fastest opencv background subtraction* [interaktyvus]. 2017. BackgroundSubtractorCNT [žiūrėta 2019 m. gegužės 22 d.]. Prieiga per internetą: <<https://sagi-z.github.io/BackgroundSubtractorCNT/>>
25. KAEWTRAKULPONG, P.; BOWDEN, R. 2002. An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection, *Proceedings of 2nd European Workshop on Advanced Video-Based Surveillance Systems 2002*: 135–144.
26. GODBEHERE, A.; MATSUKAWA, A.; GOLDBERG, K. 2002. An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection, *2012 American Control Conference*: 4305–4312.
27. *BackgroundSubtractorGSOC* [interaktyvus]. 2018. OpenCV Documentation [žiūrėta 2019 m. gegužės 22 d.]. Prieiga per internetą: <https://docs.opencv.org/4.1.0/d2/d55/group_bgsegm.html>
28. GUO, L.; XU, D.; QIANG, Z. 2016. Background Subtraction Using Local SVD Binary Pattern, *IEEE Conference on Computer Vision and Pattern Recognition 2016*: 1159–1167.
29. BOUGUET, J. 2000. Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm, *Intel Corporation. Microprocessor Research Labs*. 1.
30. WULFF, J.; BLACK, M. 2015. Efficient sparse-to-dense optical flow estimation using a learned basis and layers, *IEEE Conference on Computer Vision and Pattern Recognition 2015*: 120–130.

31. WEINZAEPFEL, P.; REVAUD, J.; HARCHAOUI, Z.; SCHMID, C. 2013. DeepFlow: Large displacement optical flow with deep matching, *IEEE International Conference on Computer Vision 2013*: 1385–1392.
32. TAO, M.; BAI, J.; KOHLI, P.; PARIS, S. 2012. SimpleFlow: A Non-iterative, Sublinear Optical Flow Algorithm, *Computer Graphics Forum*, 31: 345–353.
33. ZACH, C.; POCK, T.; BISCHOF, H. 2007. A Duality Based Approach for Realtime TV-L1 Optical Flow, *Pattern Recognition*, 4713: 214–223.
34. REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. 2016. You Only Look Once: Unified, Real-Time Object Detection, *IEEE Conference on Computer Vision and Pattern Recognition 2016*: 779–788.
35. LIU, W.; ANGUELOV, D.; ERHAN, D.; SZEGEDY, C.; REED, S.; FU, C.; BERG, A. 2016. SSD: Single Shot MultiBox Detector, *European Conference on Computer Vision 2016*: 21–37.
36. REDMON, J.; FARHADI, A. 2017. YOLO: Better, Faster, Stronger, *IEEE Conference on Computer Vision and Pattern Recognition 2017*: 6517–6525.
37. REDMON, J.; FARHADI, A. 2017. YOLOv3: An Incremental Improvement [interaktyvus]. [žiūrėta 2019 m. gegužės 22 d.]. Prieiga per internetą: <<https://pjreddie.com/media/files/papers/YOLOv3.pdf>>
38. BEWLEY, A.; ZONGYUAN, G.; OTT, L.; RAMOS, F. UPCROFT, B. 2016. Simple online and realtime tracking, *IEEE International Conference on Image Processing 2016*: 3464–3468.
39. *Optical Flow Evaluation Datasets* [interaktyvus]. 2009. The Middlebury Computer Vision Pages [žiūrėta 2019 m. gegužės 22 d.]. Prieiga per internetą: <<http://vision.middlebury.edu/flow/data/>>
40. *2D MOT 2015* [interaktyvus]. 2015. Multiple Object Tracking Benchmark [žiūrėta 2019 m. gegužės 22 d.]. Prieiga per internetą: <https://motchallenge.net/data/2D_MOT_2015/>
41. LIN, T.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLAR, P.; ZITNICK, C. L. 2016. Microsoft COCO: Common Objects in Context, *European Conference on Computer Vision 2014*: 740–755.

PRIEDAI

A priedas. Judesio analizės algoritmų tyrimo rezultatai



REGA GRĮSTŲ JUDESIO ANALIZĖS ALGORITMŲ TYRIMAS

Gediminas STANKEVIČIUS

Vadovas doc. dr. Dalius Matuzevičius

Svarbiausi judesio analizės algoritmo tyrimo rezultatai

1 lentelė. Judesio aptikimo algoritmų spartos tyrimo rezultatai (*Jetson Nano*)

Algoritmas	160×120 raiška		320×240 raiška	
	T, ms	f, Hz	T, ms	f, Hz
CNT	0,235	4248,128	0,641	1559,168
MOG	1,334	749,640	5,060	197,635
MOG2	1,311	762,664	4,785	208,971
KNN	1,499	667,315	5,627	177,723
GMG	3,035	329,488	12,152	82,291
GSOC	4,438	225,334	16,878	59,249
LSBP	7,963	125,584	26,159	38,227

2 lentelė. Optinio srauto algoritmų spartos tyrimo rezultatai (*Jetson Nano*)

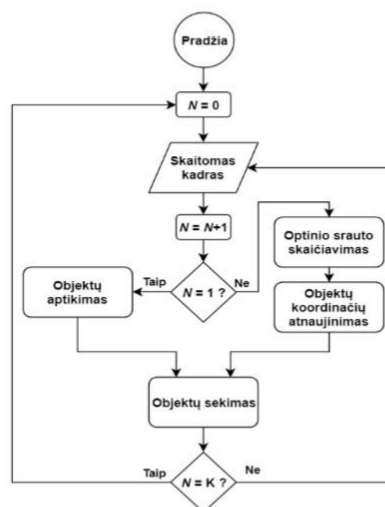
Algoritmas	160×120 raiška		320×240 raiška	
	T, ms	f, Hz	T, ms	f, Hz
DIS	1,888	529,527	6,990	143,07
LK-SD	16,107	62,085	64,990	15,387
Farneback	40,182	24,887	164,512	6,079
PCA	41,584	24,048	172,874	5,785
DeepFlow	229,067	4,366	911,475	1,097
SimpleFlow	280,089	3,570	1113,07	0,898
TVL1	326,370	3,064	1136,84	0,880

3 lentelė. Optinio srauto algoritmų tikslumo tyrimo rezultatai

Algoritmas	Vidutinė EPE klaida								Vidurkis
	a	b	c	d	e	f	g	h	
DeepFlow	0,08	0,17	0,70	0,17	0,12	0,37	0,46	0,28	0,29
LK-SD	0,17	0,27	0,92	0,23	0,18	0,69	0,69	0,34	0,44
DIS	0,15	0,32	0,85	0,25	0,22	0,64	2,02	0,40	0,61
PCA	0,19	0,36	1,14	0,36	0,28	0,90	1,20	0,51	0,62
TVL1	0,18	0,16	0,76	0,19	0,16	3,58	2,07	0,30	0,92
Farneback	0,86	0,49	1,26	0,49	0,33	1,51	3,09	1,44	1,18
SimpleFlow	1,57	2,18	2,89	3,23	1,35	6,86	4,68	2,93	3,21

Lentelėje raidėmis nuo a iki h pažymėti *Middlebury* duomenų rinkinio pavyzdžiai pagal tokį eiliškumą: a – *Dimetrodon*, b – *Grove2*, c – *Grove3*, d – *Hydrangea*, e – *RubberWhale*, f – *Urban2*, g – *Urban3*, h – *Venus*.

Pasiūlyto objektų aptikimo ir sekimo algoritmo tyrimo rezultatai



1 pav. Pasiūlyto daugelio objektų sekimo algoritmo struktūrinė schema

4 lentelė. Maksimalus kadrų dažnis, kai N yra nuo 1 iki 10 (*Jetson Nano*)

Aptikimas vykdomas 1 kadre iš:	Maksimalus kadrų dažnis, Hz
1	10,17
2	18,11
3	24,91
4	30,37
5	35,62
6	38,85
7	41,09
8	45,23
9	49,30
10	50,41

5 lentelė. MOTA ir MOTP, kai N yra nuo 1 iki 10 (*Jetson Nano*)

Aptikimas vykdomas 1 kadre iš:	MOTA	MOTP
1	89,72	94,69
2	88,38	91,94
3	86,00	89,86
4	82,61	87,90
5	79,14	86,55
6	75,34	85,19
7	72,05	84,21
8	67,01	83,6
9	63,59	82,53
10	59,74	81,63

B priedas. Pranešimo XXII-oje Lietuvos jaunųjų mokslininkų konferencijoje medžiaga



VILNIAUS GEDIMINO
TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS

REGA GRĮSTŲ JUDESIO ANALIZĖS ALGORITMŲ TYRIMAS

22-OJI JAUNŲJŲ MOKSLININKŲ KONFERENCIJA
„MOKSLAS – LIETUVOS ATEITIS: ELEKTRONIKA IR ELEKTROTECHNIKA“

Gediminas STANKEVIČIUS, gediminas.stankevicius@stud.vgtu.lt

Vadovas: doc. dr. Dalius MATUZEVIČIUS

Elektroninių sistemų katedra

Vilnius, 2019



VILNIAUS GEDIMINO
TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS

[Gediminas STANKEVIČIUS]

2019-03-15

Jaunųjų mokslininkų konferencija
„Mokslas – Lietuvos ateitis: Elektronika ir elektrotechnika“

Rega grįsta judesio analizė

- Analizuojami vaizdų srautai, kuriuos sudaro kadro seka
- Judesiu vadinamas vaizdo kitimas tarp kadro (laike)
- Tikslas – iš vaizdų srauto duomenų išgauti norimą informaciją
 - Judesio aptikimas
 - Judančių objektų judėjimo greičio, krypties nustatymas
 - Veiksmų, veiklų atpažinimas



Aktualumas, taikymo sritys

- **Tokio tipo algoritmai naudojami aktyviai vystomoms užduotims spręsti:**
 - Saugos sistemose (judesio aptikimas, žmonių, veidų aptikimas)
 - Robotikoje bei autonomiam transportui kurti (patalpų 3D žemėlapių sudarymas, odometrija, kliūčių aptikimas)
 - 3D objektų skenavimas

3 10



Atlikti tyrimai

- **7 judesio aptikimo / fono atmetimo algoritmų spartos tyrimas įterptinėje sistemoje**
- **7 optinio srauto (angl. *Optical Flow*) algoritmų spartos ir tikslumo tyrimas įterptinėje sistemoje**
 - Tikslumui įvertinti naudotas judesio vektorių pabaigos klaidos (angl. *End Point Error*) kriterijus EPE
 - Tikslumo tyrimui naudota dažnai optinio srauto algoritmams vertinti naudojama *Middlebury* [1] duomenų bazė
 - Kadrai + tikslus optinis srautas

Programinis kodas: <https://github.com/gediminas-stan/jmk2019>

4 10

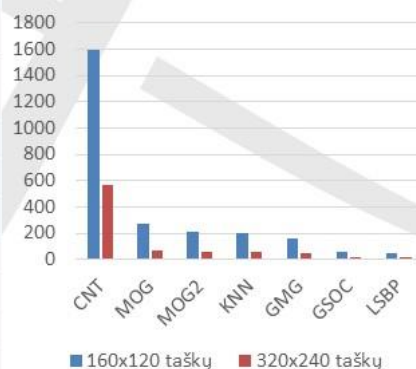


Tyrimų rezultatai

- Judesio aptikimo algoritmų spartos tyrimo rezultatai (CPU: 4×ARM Cortex-A53, 1,2 GHz)

Algoritmas	160×120		320×240	
	T, ms	f, Hz	T, ms	f, Hz
CNT [2]	0.629	1588.882	1.765	566.484
MOG [3]	3.683	271.543	14.072	71.061
MOG2 [4]	4.849	206.247	18.357	54.474
KNN [5]	4.98	200.788	18.464	54.16
GMG [6]	6.421	160.223	23.024	43.433
GSOC [7]	16.084	62.173	53.076	18.841
LSBP [8]	22.254	44.936	75.487	13.247

Kadrų apdorojimo dažnis



Programinis kodas: <https://github.com/gediminas-stan/jmk2019>

5 10

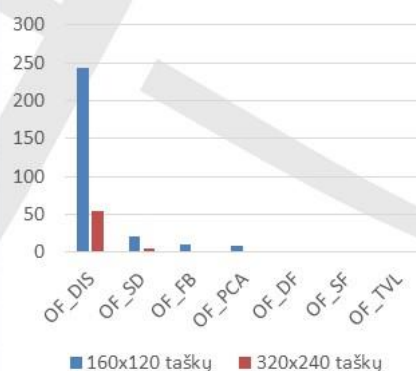


Tyrimų rezultatai

- Optinio srauto algoritmų spartos tyrimo rezultatai (CPU: 4×ARM Cortex-A53, 1,2 GHz)

Algoritmas	160×120		320×240	
	T, ms	f, Hz	T, ms	f, Hz
OF_DIS [9]	4.111	243.256	18.065	55.355
OF_SD [10]	46.01	21.735	181.281	5.516
OF_FB [11]	93.963	10.642	416.349	2.402
OF_PCA [12]	108.807	9.191	457.932	2.184
OF_DF [13]	738.678	1.354	2985.137	0.335
OF_SF [14]	768.74	1.301	4928	0.203
OF_TVL [15]	3549.35	0.282	13894.61	0.072

Kadrų apdorojimo dažnis



Programinis kodas: <https://github.com/gediminas-stan/jmk2019>

6 10



Tyrimų rezultatai

- **Optinio srauto algoritmų tikslumo tyrimo rezultatai naudojant Middlebury [1] vertinimo duomenų bazę**

Algoritmas	Vidutinė EPE klaida								
	Dimetrodon	Grove2	Grove3	Hydrangea	RubberWhale	Urban2	Urban3	Venus	Vidurkis
OF_DF [13]	0.085	0.171	0.702	0.17	0.12	0.367	0.462	0.281	0.295
OF_SD [10]	0.17	0.274	0.925	0.232	0.18	0.689	0.688	0.339	0.437
OF_DIS [9]	0.151	0.322	0.853	0.247	0.219	0.645	2.019	0.393	0.606
OF_PCA [12]	0.187	0.359	1.143	0.364	0.284	0.898	1.196	0.509	0.618
OF_TVL [15]	0.179	0.156	0.757	0.194	0.156	3.578	2.069	0.303	0.924
OF_FB [11]	0.859	0.492	1.265	0.489	0.328	1.512	3.088	1.445	1.185
OF_SF [14]	1.568	2.177	2.891	3.232	1.353	6.856	4.677	2.934	3.211

 Programinis kodas: <https://github.com/gediminas-stan/jmk2019>

7 10



Išvados

- Iš tirtų judesio aptikimo algoritmų sparčiausias CNT [2]
- Pagal tyrimo rezultatus geriausias spartos ir tikslumo santykis OF_DIS [9] algoritmo
- Lėčiausi optinio srauto algoritmai nėra tiksliausi, todėl svarbu objektyviai įvertinti alternatyvius algoritmus
- Tolimesniuose tyrimuose pasirinktas naudoti OF_DIS [9] algoritmas

8 10



Literatūra

- [1] Daniel Scharstein, 2009. *The Middlebury Computer Vision Optical Flow Page, Datasets*. [Interaktyvus]. [Žiūrėta 2019 m. kovo 15 d.]. Prieiga per internetą: <<http://vision.middlebury.edu/flow/data/>>
- [2] Sagi Zeevi, 2016. The fastest opencv background subtraction, *BackgroundSubtractorCNT*. [Interaktyvus]. [Žiūrėta 2019 m. kovo 15 d.]. Prieiga per internetą: <<https://sagi-z.github.io/BackgroundSubtractorCNT/>>
- [3] Pakorn KaewTraKulPong, Richard Bowden. 2002. An improved adaptive background mixture model for real-time tracking with shadow detection. *Video-Based Surveillance Systems*, pages 135–144. Springer.
- [4] Zoran Živkovic. 2004. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE.
- [5] Zoran Živkovic and Ferdinand van der Heijden. 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780.
- [6] Andrew B Godbehere, Akihiro Matsukawa, and Ken Goldberg. 2012. Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. In *American Control Conference (ACC)*, 2012, pages 4305–4312. IEEE.
- [7] OpenCV documentation, 2018. *BackgroundSubtractorGSOC*. [Interaktyvus]. [Žiūrėta 2019 m. kovo 15 d.]. Prieiga per internetą: <https://docs.opencv.org/4.0.1/d4/dd5/classcv_1_1bgsegm_1_1BackgroundSubtractorGSOC.html>
- [8] L. Guo, D. Xu, and Z. Qiang. 2016. Background subtraction using local svd binary pattern. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1159–1167.
- [9] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. 2016. Fast optical flow using dense inverse search. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- [10] OpenCV documentation, 2018. *OpticalFlowSparseToDense*. [Interaktyvus]. [Žiūrėta 2019 m. kovo 15 d.]. Prieiga per internetą: <https://docs.opencv.org/4.0.1/d2/d84/group_optflow.html#gad6aa63f2703202806fe18dc1353b5f4b>
- [11] G. Farneback, 2003. Two-Frame Motion Estimation Based on Polynomial Expansion. *Proceedings of the 13th Scandinavian Conference on Image Analysis*.
- [12] J. Wulff and M. J. Black. 2015. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [13] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. 2013. Deepflow: Large displacement optical flow with deep matching. In *Computer Vision (ICCV)*, 2013 IEEE International Conference on, pages 1385–1392. IEEE.
- [14] M. Tao, J. Bai, P. Kohli, and S. Paris. 2012. Simpleflow: A non-iterative, sublinear optical flow algorithm. In *Computer Graphics Forum*.
- [15] Christopher Zach, Thomas Pock, and Horst Bischof. 2007. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition*, pages 214–223. Springer

9 10



Ačiū už dėmesį

10 10