

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
MATEMATINĖS INFORMATIKOS KATEDRA

Kursinis darbas

Kompiuterinė rega ir jos taikymai biologijoje ir medicinoje
(Computer vision and its applications in biology and medicine)

Atliko: 3 kurso 1 grupės studentas

Vytautas Jankauskas (parašas)

Darbo vadovas:

lekt. Irus Grinis (parašas)

Vilnius
2016

Turinys

Ivadas	2
1. Susiję darbai	3
2. OpenCV kompiuterinės regos biblioteka	4
3. Programos prototipo architektūros projektavimas	5
3.1. Video įvesties šaltinis – VideoCapture klasė	5
3.2. Fono atimtis (ang. Background subtraction).....	6
3.3. Paveikslėlio slenkstis (ang. Picture threshold).....	7
3.4. Judančio objekto kontūrų radimas.....	7
4. Programos prototipo kūrimas	10
4.1. Algoritmo ypatybės	11
4.1.1. Triukšmas paveikslėlyje.....	11
4.1.2. Žmogaus nueito atstumo matavimas	11
4.1.3. Identifikuojamas judančio žmogaus kontūras.....	11
4.1.4. Apskaičiuojamas kontūro centras	11
4.1.5. Atstumas tarp etaloninio objekto šoninių kraštinių.....	11
4.1.6. Nueito atstumo matavimo vienetas	12
4.1.7. Žmogaus judėjimo grafikas	12
4.1.8. Grafiko braižymas	12
5. Programos veikimo bandymai ir analizė	13
6. Rezultatai ir išvados	16
Santrauka	18
Summary	19
Literatūros sąrašas	20

Įvadas

Pastaraisiais dešimtmečiais objektų atpažinimas, aptikimas ir sekimas įgauna vis didesnę svarbą. Tai lemia smarkiai auganti informacijos apie objekto tapatybę, lokaciją, judėjimą paklausa kariuomenėje, gamybos sektoriuje, apsaugos sistemose, transporto sektoriuje, sveikatos apsaugoje, medicinoje. Kompiuterinė rega tobulėja dėl mažėjančių vaizdo kamerų kainų ir gerėjančios jų kokybės bei todėl, jog regos algoritmai tampa vis patikimesni, veikia sklandžiau [1]. Didėjanti žmonių darbo jėgos kaina skatina ieškoti ir kurti automatizuotas, robotizuotas sistemas, kuriose eliminuojamas arba iki minimumo sumažinamas žmogaus įsikišimo poreikis.

Šiais laikais vis daugėja žmonių, kuriems reikalingos slaugos paslaugos. Didėja pensinio amžiaus žmonių skaičius. Šie veiksniai kelia įvairių tipų slaugytojų paklausą, todėl atitinkamai didėja ir slaugytojų darbo kaštai. Dėl šių priežasčių kuriamos įvairios kompiuterinės regos sistemos, mažinančios slaugytojų poreikį ir lengvinančios jų darbą. Gebėjimas aptikti ir sekti judančius objektus, žmones leidžia nustatyti jų buvimą ar nebuvimą stebimoje aplinkoje, analizuoti jų judėjimą toje erdvėje. Šie veiksniai gali būti naudojami nustatinėjant asmens buvimo poziciją, nustatant tapatybę. Jie taip pat gali būti praktiškai panaudojami sveikatos apsaugoje.

Reabilitacijos įstaigose, kai žmonės atsigavinėja po traumų ar operacijų jiems būtina mankštintis, atlikinėti judėjimo pratimus. Norint užtikrinti šių veiksmų atlikimą, reikalinga slaugytojo priežiūra. Tačiau, kaip alternatyvus priežiūros būdas buvo sugalvota kompiuterinė programa, gebanti aptikti ir sekti žmogaus judėjimą bei apskaičiuoti jo nueitą atstumą.

Kursinio darbo **tikslas** – sukurti programos, gebančios apskaičiuoti žmogaus nueinamą atstumą, naudojant tik kompiuterio kamerą, prototipą. Programa taip pat turi išvesti žmogaus judėjimo grafiką nustatytoje erdvėje.

Šio kursinio darbo tikslui pasiekti iškelti šie **uždaviniai**:

1. Ištirti, kokiais kompiuterinės regos įrankiais galima sukurti programą.
2. Suprojektuoti programos prototipo architektūrą.
3. Sukurti programos prototipą.
4. Išbandyti programos veikimą realioje aplinkoje ir išanalizuoti bandymų rezultatus.

Šio kursinio darbo **rezultatas** – veikiantis kompiuterinės regos programos prototipas, apskaičiuojantis žmogaus nueitą atstumą ir sukuriantis žmogaus judėjimo grafiką nustatytoje erdvėje.

1. Susiję darbai

Žmonių atpažinimo ir sekimo problemos kompiuterinės regos mokslo srityje yra pakankamai senos, egzistuojančių sekimo metodų apžvalgą galima rasti [6].

Kompleksiškesnę keleto žmonių atpažinimo ir sekimo techniką naudojant aukšto dažnio radijo bangas ir kelias vaizdo kameras aprašo Mandeljc [4]. Pagrindinis tokio metodo pranašumas yra asmens identifikavimo trikdžių pašalinimas. Bandymų rezultatai, palyginus su vien tik radijo bangas naudojančiu metodu, parodo didesnę lokalizavimo tikslumą. Taip pat toks metodas geriau apsaugo nuo identifikavimo trikdžių atsiradimo, kurie neišvengiami vien tik kompiuterinę regą naudojančiuose metoduose.

Kompiuterinė rega pasitelkiama medicinoje ir norint išvengti ar bent jau supaprastinti pasikartojančius, daug laiko reikalaujančius tyrimus, reikalaujančius aukšto lygio kvalifikacijos specialistų. Tokie tyrimai turi ir aukštą klaidų atsiradimo lygį. Linder sukūrė kompiuterinę regą pagrįstą maliarijos diagnozavimo metodą [3]. Šis įrankis, naudojant objektų atpažinimą, gydytojui pateikia tik tuos kraujo mėginio regionus, kuriuose maliarijos infekcijos tikimybė didžiausia.

Judesių ir judėjimo modelių atpažinimas medicinoje pradedamas automatizuoti ir stebėjimų tikslais. Hashemi sukurtas įrankis neinvaziniam su autizmu susijusio elgesio matavimui kūdikių veiksmuose [2]. Autizmo požymius kūdikių elgesyje įmanoma pastebėti jau pirmaisiais jų gyvenimo metais. Tačiau tai yra labai daug laiko ir kvalifikuotų specialistų reikalaujantis darbas, kuriame tiriamas kūdikių elgesys, jų veido išraiškos. Sukurtas algoritmas matuoti autizmo spektro sutrikimą rodančius veiksnius sekant atitinkamas veido charakteristikas. Metodo bandymų rezultatai parodė galimybę tokią elgesio matavimo techniką pritaikyti ir realiuose klinikiuose kūdikių elgesio vertinimuose.

Kompiuterinės regos taikymas medicinoje galimas ir higienos tikslais. Chen sukurtas įrankis aptinkantis sveikatos apsaugos darbuotojų ir pacientų kontaktą [5] taip įvertinant darbuotojų rankų higieną ir numatytų apsaugos priemonių dėvėjimo taisyklių laikymąsi. Prie paciento lovos kamų buvo pritvirtinti davikliai, fiksuojantys kada sveikatos apsaugos darbuotojo rankos patekdavo į nustatytą zoną virš lovos. Taip pat buvo identifikuojamas darbuotojo liemuo, veidas ir šios zonos buvo lyginamos su tikromis veido apsaugos kaukių, pirštinių, medicininių chalātų spalvomis.

2. OpenCV kompiuterinės regos biblioteka

Kompiuterinė rega sparčiai vystosi jau keletą dešimtmečių. Prie jos plėtros smarkiai prisidėjo ir biblioteka OpenCV. OpenCV – atviro kodo kompiuterinės regos ir save mokančių sistemų programinės įrangos biblioteka. Pirmoji bibliotekos alfa versija išleista dar 2000-aisiais metais. OpenCV 1.0 versija išleista 2006 metais. Biblioteka turi daugiau nei 2500 optimizuotų algoritmų. Šiuos algoritmus galima įvairiai taikyti – vaizdų aptikimui ir atpažinimui, objektų identifikavimui ir sekimui, panašių vaizdų atpažinimui, klasifikavimui ir pan.

Šiam kursinio darbo kompiuterinės programos kūrimui OpenCV biblioteka buvo pasirinkta todėl, kad yra atviro kodo, taip pat turi didelę vartotojų bendruomenę (daugiau nei 47 tūkst. vartotojų) ir daugiau nei 7 mln. atsisiuntimų. Įprasti bibliotekos panaudojimo pavyzdžiai – aptikti, atpažinti, sekti žmones, judančius objektus, sekti akių judesius, išgauti 3D modelius iš objektų, rasti panašius paveikslėlius iš paveikslėlių duomenų bazės, klasifikuoti žmogaus judesius ir pan.

OpenCV biblioteką aktyviai naudoja tiek tyrėjų grupės, tiek valstybinės institucijos, tiek ir pasaulyje žinomos įmonės (Google, Yahoo, Honda, Microsoft). Biblioteką naudoja ir įvairūs startuoliai, gebėdami atrasti naujų pritaikymo būdų.

Ši kompiuterinės regos biblioteka turi C++, C, Python, JAVA ir MATLAB programavimo kalbų sąsajas bei palaiko Windows, Linux, Mac OS bei Android operacines sistemas. Todėl programuoti naudojant šią biblioteką gali skirtingų įgūdžių turintys programuotojai.

3. Programos prototipo architektūros projektavimas

Kompiuterinės programos prototipo kūrimui buvo pasirinkta OpenCV bibliotekos Python programavimo kalbos sąsaja. Python yra sąlyginai paprasta ir lengvai skaitoma kalba, leidžianti programos kūrėjui išreikšti savo idėją naudojant mažiau kodo eilučių, bet tuo pačiu metu neprarandant kodo suprantamumo.

Buvo vykdomas tyrimas ieškant, kokie komponentai reikalingi veikiančiam programos prototipui sukurti. Tolesniuose skirsniuose pateikiami reikalingi komponentai su naudojimo pavyzdžiais.

3.1. Video įvesties šaltinis – VideoCapture klasė

Pirmiausia, norint apdoroti vaizdinę medžiagą, reikia ją pateikti OpenCV bibliotekai suprantamu formatu. Tam naudojama VideoCapture klasė. Argumentas šios klasės objektui gali būti įvesties prietaiso (kompiuterinės vaizdo kameros) indeksas arba video failo pavadinimas. Tuomet su funkcija `read()` galima vieną po kito imti kadrus iš video failo ir atlikti norimus veiksmus su jais. OpenCV naudoja Numpy – optimizuotą skaitinių operacijų biblioteką su MATLAB stiliaus sintakse.

```

1 # Importuoti OpenCV ir Numpy bibliotekas:
2 import numpy as np
3 import cv2
4
5 # Paduoti įvesties šaltinį - kompiuterio vaizdo kamerą:
6 cap = cv2.VideoCapture(0)
7
8 # Imti kadrus vieną po kito iš įvesties šaltinio:
9 while(True):
10     ret, frame = cap.read()

```

1 kodo fragmentas. Naudojimo pavyzdys



1 pav. Vienas kadras iš video įvesties.

3.2. Fono atimtis (ang. Background subtraction)

Kitas žingsnis norint gebėti aptikti objektą kadre yra fono atimties technika. Tikslas – atskirti „nejudantį“ paveikslėlio foną nuo „judančio“ objekto (žmogaus ar transporto priemonės). Tai galima padaryti turint du tos pačios vietos paveikslėlius, iš kurių tik viename bus judantis objektas. Reikia atimti nuotrauką su objektu iš nuotraukos be objekto. Atimties veiksmu laikysime iš einamojo kadro atimamą pirmąjį, etaloninį paveikslėlį.

OpenCV `absdiff(image1, image2)` funkcija apskaičiuoja skirtumą tarp kiekvieno elemento dviejuose elementų masyvuose.

```

1 import cv2
2 import time
3
4 # Įkeliami paveikslėlius - be judančio objekto ir su judančiu objektu:
5 img1 = cv2.imread('initFrame.jpg')
6 img2 = cv2.imread('frame1.jpg')
7
8 # Paverčiame paveikslėlius į pustoninius (gray scale):
9 img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
10 img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)
11
12 # Skaičiuojame skirtumą tarp jų:
13 difference = cv2.absdiff(img1, img2)

```

2 kodo fragmentas. Naudojimo pavyzdys

3.3. Paveikslėlio slenkstis (ang. Picture threshold)

Fono atimties skirtumo vizualizavimui naudojamas paveikslėlio slenkščio metodas: jei pikselio reikšmė yra didesnė už slenkstinę vertę, tam pikseliui priskiriama atitinkama – dažniausiai baltos spalvos vertė. Kitu atveju pikselį paverčiame juodu. Paveikslėlio šaltinis turėtų būti sudarytas iš pilkų pustonijų (ang. gray scale). Taip pat reikia nustatyti slenkstinę vertę, pagal kurią bus klasifikuojamos pikselių reikšmės.

```

1 # Importuoti OpenCV ir Numpy bibliotekas:
2 import numpy as np
3 import cv2
4
5 # Įkrauname paveikslėlį ir paverčiame jį į sudarytą iš pustonijų:
6 img = cv2.imread('img1.png', 0)
7 img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
8
9 # Sukuriame slenkstinę paveikslėlio versiją:
10 thresh = cv2.threshold(img, 60, 255, 0)[1]

```

3 kodo fragmentas. Naudojimo pavyzdys

3.4. Judančio objekto kontūrų radimas

Turint slenkstinę (juodai baltą) paveikslėlio versiją, galima joje rasti judančio objekto koordinatas. OpenCV bibliotekos findContours(a, b, c) funkcija į vieną kontūrą sujungia visus tarpusavyje

besijungiančius tos pačios spalvos ar vienodo intensyvumo pikselius. Kad būtų aiškiau, paveikslėlyje aplink kontūrą apibrėžiamas mažiausio įmanomo perimetro stačiakampis.

```

1 # Importuoti OpenCV ir Numpy bibliotekas:
2 import numpy as np
3 import cv2
4
5 # Įkrauname paveikslėlį ir paverčiame jį į sudarytą iš pustonų:
6 img = cv2.imread('img1.png',0)
7 img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
8
9 # Sukuriame slenkstinę paveikslėlio versiją:
10 thresh = cv2.threshold(img, 60, 255, 0)[1]
11
12 # Randame judančio objekto kontūrus ir apibrėžiame aplink jį
13 # stačiakampį:
14 (cnts, _) = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
15     cv2.CHAIN_APPROX_SIMPLE)
16 x,y,w,h = cv2.boundingRect(cnt)
17 cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)

```

4 kodo fragmentas. Naudojimo pavyzdys

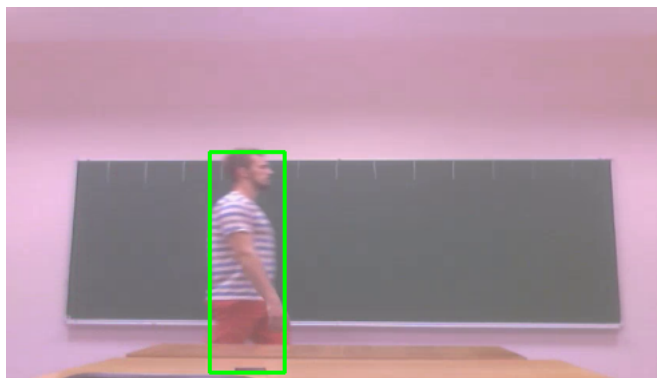


(a) Įprastas kadras su judančiu žmogumi.



(b) Slenkstinė versija.

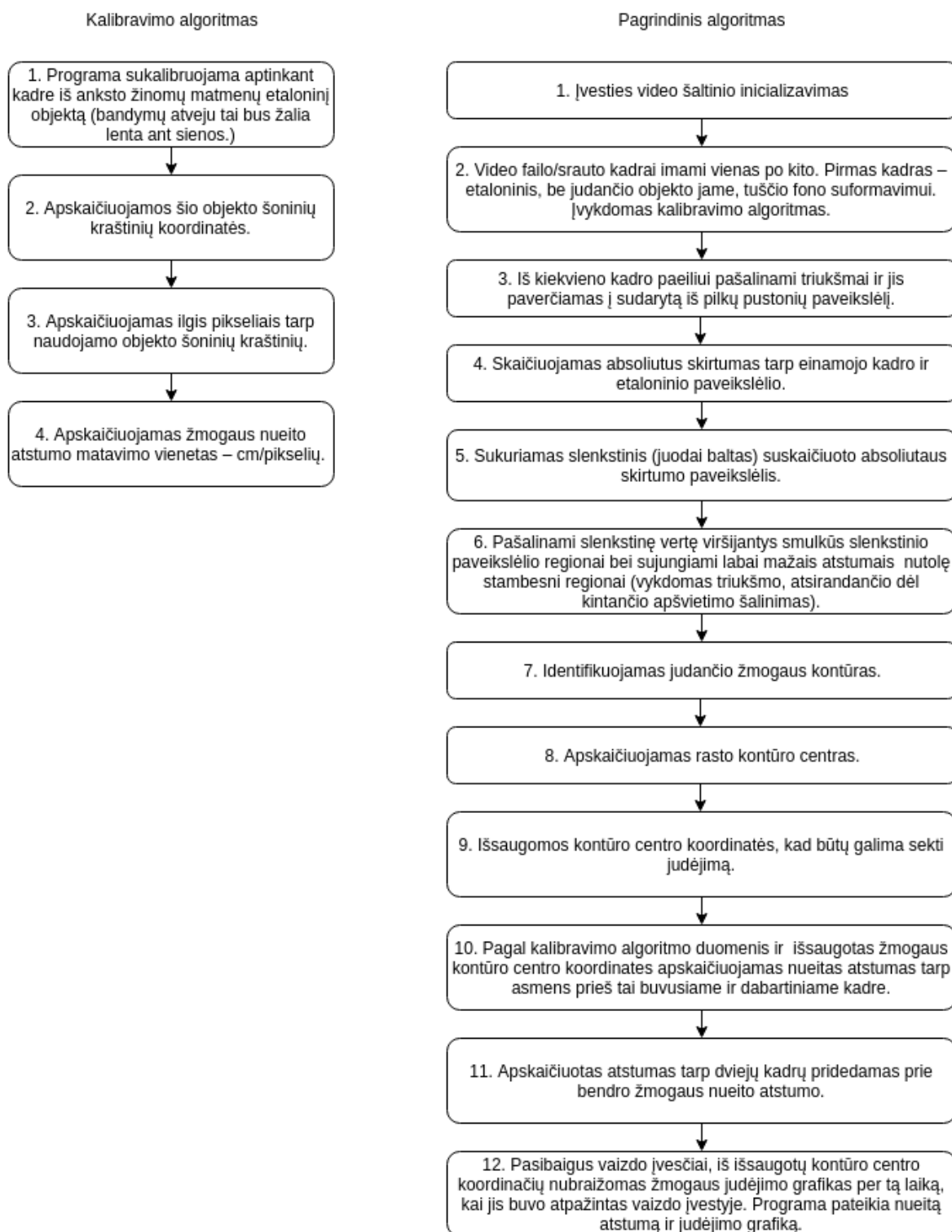
2 pav. Paveikslėlio slenkstis.



3 pav. Judančio objekto kontūrai apibrėžti stačiakampiu.

4. Programos prototipo kūrimas

Pasirinkus pagrindinius kompiuterinės regos komponentus, reikalingus šio kursinio darbo programos prototipui, buvo imtasi jo kūrimo.



4 pav. Programos algoritmas.

4.1. Algoritmo ypatybės

Kuriant veikiančią programos prototipo algoritmą, verta plačiau pakomentuoti jo žingsnius, kuriuos įgyvendinant kilo sunkumų ar buvo įvesti algoritmą gerinantys patobulinimai.

4.1.1. Triukšmas paveikslėlyje

Pirmiausia, algoritmas yra jautrus apšvietimo pokyčiams vaizdo įvestyje (3 algoritmo žingsnis 4-ajame paveikslėlyje). Šios problemos sprendimas susidėjo iš kelių etapų. Buvo bandoma išlaikyti kiek įmanoma pastovesnį apšvietimą – bandymo patalpoje naudojant dirbtinį apšvietimą. Kita sprendimo dalis buvo sureguliuoti slenkstinę vertę. Jei ji per žema, tai slenkstiniame paveiksliuke atvaizduojami ir dėl kintančio apšvietimo nežymiai pasikeitę pikselių blokai. Tačiau, jei slenkstinė vertė bus per aukšta, tuomet, priklausomai nuo fono ir žmogaus odos bei aprangos spalvų, paveiksliuke bus atvaizduojami tik daliniai, mažesni, sutrūkinėję kontūrai. Arba žmogus nebus atpažintas apskritai. Galiausiai, buvo panaudotos morfologinės paveikslėlio transformacijos ir taip sumažintas triukšmas paveikslėlyje.

4.1.2. Žmogaus nueito atstumo matavimas

Norint apskaičiuoti žmogaus nueitą atstumą, programai reikia pateikti etaloninio objekto, esančio vaizdo įvesties įrašo regos lauke, realius išmatavimus (1-as kalibravimo algoritmo žingsnis). Todėl šiai problemai spręsti prototipo bandymams buvo pasirinkta patalpa su išmatuoto dydžio žalia rašymo lenta, kabančia ant sienos. Tokiu būdu buvo įgalintas žmogaus nueito atstumo apskaičiavimas.

4.1.3. Identifikuojamas judančio žmogaus kontūras

7-ajame pagrindinio algoritmo žingsnyje identifikuojamas žmogaus kontūras. Pagal nutylėjamą laikoma, kad tai yra vienintelis ir didžiausias patalpoje judantis objektas. Tokia prielaida leidžia išvengti žmogaus atpažinimo bei identifikavimo būtinumo. Šie metodai bus įgyvendinti kitoje programos prototipo versijoje.

4.1.4. Apskaičiuojamas kontūro centras

Žmogaus nueitam atstumui matuoti naudojamas surasto kontūro centras (8-as žingsnis). Darant tokią prielaidą dalinai sumažėja algoritmo tikslumas skaičiuojant nueitą atstumą. Tačiau tokiu būdu įmanoma sekti asmens judėjimą ir tuo atveju, kai dalį jo kūno užstoja kiti objektai.

4.1.5. Atstumas tarp etaloninio objekto šoninių kraštinių

Daroma prielaida, kad objektas, naudojamas kalibravimo algoritme užima didžiąją dalį vaizdo įvesties regėjimo lauko. Dėl to nueitas atstumas skaičiuojamas tik to objekto ribose. Tai gali pa-

didinti matavimo paklaidą, jei vaizdo kameros regėjimo lauke žmogus daugiau judės už etaloninio objekto ribų.

4.1.6. Nueito atstumo matavimo vienetas

Algoritmo 15 žingsnyje apskaičiuojamas nueito atstumo matavimo vienetas – pikselio įvertinimas centimetrais. Dėl to matavimo rezultatai pateikiami metrinėje sistemoje bei yra pakankamai tikslūs.

4.1.7. Žmogaus judėjimo grafikas

Žmogaus judėjimo grafiko duomenys yra atpažinto žmogaus kontūro horizontalaus centro koordinatės kiekviename kadre. Braižomame grafike vaizduojamas pilnas žmogaus judėjimas vaizdo įvesties regėjimo lauke, t.y. ne tik žinomų matmenų objekto ribose, bet ir už jų. Tačiau tai nedaro įtakos žmogaus nueito atstumo skaičiavimui.

4.1.8. Grafiko braižymas

Būtina paminėti, kad prototipas nėra pilnai automatizuotas. Tyrėjui reikia pačiam nustatyti vaizdo įvesties tipą. Taip pat būtina nustatyti, kokio pločio yra etaloninis objektas, atpažįstamas kalibravimo algoritme. Žmogaus judėjimo grafikas braižomas naudojant Python kalbos matplotlib grafikų braižymo biblioteką.

5. Programos veikimo bandymai ir analizė

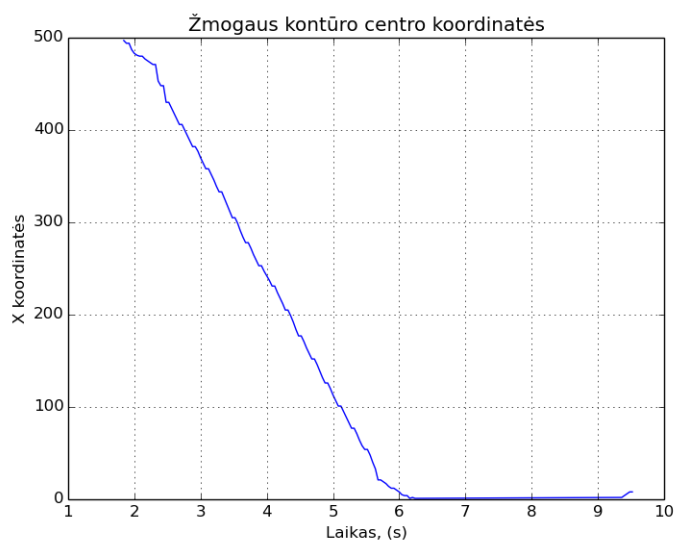
Sukūrus programos prototipą buvo būtina jo veikimą išbandyti kasdieninėmis sąlygomis. Bandymui buvo pasirinkta patalpa, kurioje veikia dirbtinis apšvietimas ir yra žinomų matmenų objektas – žalia rašymo lenta, pakabinta ant sienos. Prieš bandymą lentos plotis buvo padalintas į 14 vieno-
dų (21 cm ilgio) atkarpų, kad būtų suskaičiuotas realus žmogaus nueitas atstumas. Bandymo metu buvo filmuojama patalpos siena, ant kurios kabėjo žalia lenta. Iš pradžių buvo gautas fonas – be jokio judančio objekto kadre. Vėliau žmogus pradėjo vaikščioti palei sieną. Iš viso buvo atlikti 4 to-
kio tipo įrašai, kiekviename iš jų žmogaus ėjimo bei apsisukimo trajektorijos buvo skirtingos taip norint išmatuoti algoritmo tikslumą esant skirtingam judėjimo sudėtingumui. Žemiau pateiktoje lentelėje pavaizduoti bandymų rezultatai.

Bandymas	Faktiškai nueitas atstumas (cm)	Programos suskaičiuotas nueitas atstumas (cm)	Judėjimo trajektorija
1	300	298,5	Greitis pastovus, kryptis nesikeičia.
2	594	589,2	Žmogus pakeičia judėjimo kryptį vieną kartą.
3	1113	1091,7	Greitis nepastovus, kryptis keičiama penkis kartus.
4	1764	1787,7	Kryptis pakeičiama du kartus, o judėjimo trukmė ilgiausia.

1 lentelė. Bandymų rezultatai.

Iš rezultatų matoma, jog pirmo – elementariausio bandymo metu programos paklaida buvo 1,5 proc. Kintant judėjimo trajektorijai kituose bandymuose, kinta ir paklaidos dydis – antrame, trečiame ir ketvirtame bandymuose paklaidos dydis atitinkamai 0,2 proc., 2 proc. ir 1,3 proc. Tačiau dydis kinta nepriklausomai nuo žmogaus judėjimo krypties, apsisukimų skaičiaus. Galima teigti, kad algoritmo tikslumas anksčiau apibūdintomis sąlygomis yra pakankamai aukštas.

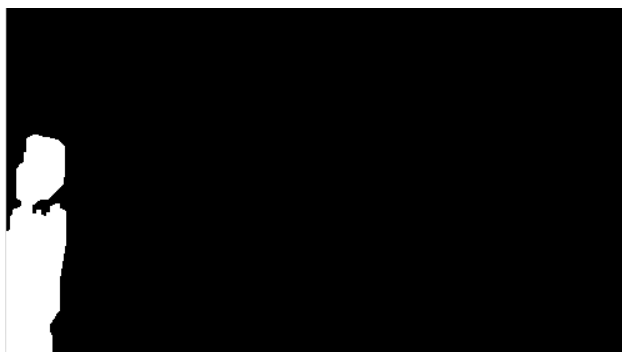
Žemiau pateikiami asmens judėjimo grafikai kiekviename bandyme. Juose vaizduojama žmogaus kontūro centro trajektorija vaizdo įvesties horizontalioje (X) ašyje pikseliais. Laikas matuojamas nuo tada, kai asmuo buvo pirmą kartą atpažintas kadre iki tada, kai jis buvo atpažintas paskutinį kartą.



5 pav. Žmogaus judėjimas pirmojo bandymo metu.

Pirmojo bandymo grafike matoma, kad judėjimas buvo įprastas, ėjimas buvo pastovaus greičio ir į vieną pusę. Tačiau, maždaug ties 2,5s ir 450 pikselių verte galima pastebėti, kad grafikas turi minimalių trūkių. Jie atsirado, nes tuo metu žmogus krito sienos ir lentos pradžios ribą. Programos prototipas neprisitaiko prie kintančio kadro fono už judančio žmogaus, todėl atpažįstamas kontūras dalinai pakeičia savo formą. Tai galima pastebėti ir žemiau pateikiamuose paveikslėliuose. Vadinasi, algoritmą reikia tobulinti ir pritaikyti jį prie besikeičiančio fono už sekamo judančio žmogaus.

Antrojo bandymo grafike matoma analogiška problema. Dėl atpažinto žmogaus kontūro pokyčio šiam kertant ribą tarp sienos ir žalios lentos susidaro nedideli įtrūkimai tarp grafiko taškų. Tokią pačią tendenciją galima pastebėti ir 3-iojo bei 4-ojo bandymų grafikuose. Nepaisant didėjančio žmogaus judėjimo sudėtingumo, paklaidos atsiranda tik žmogui kertant etaloninio objekto ribas.

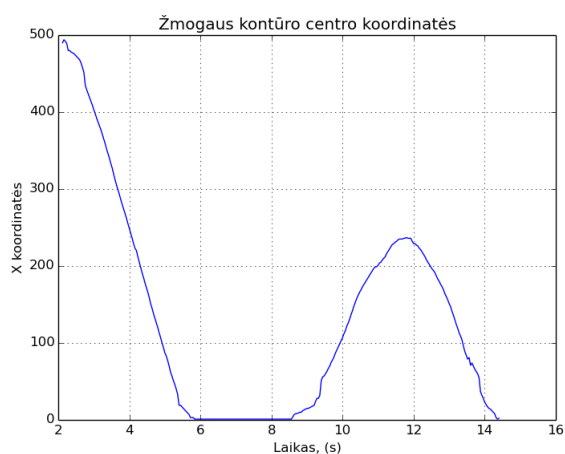


(a) Priešais sieną.

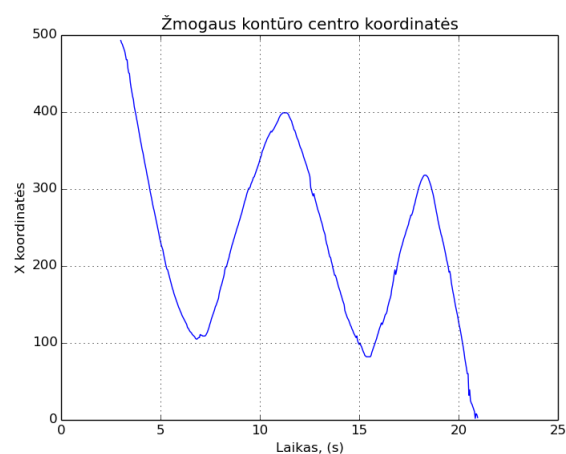


(b) Priešais žalią lentą.

6 pav. Žmogaus kontūro dydis.

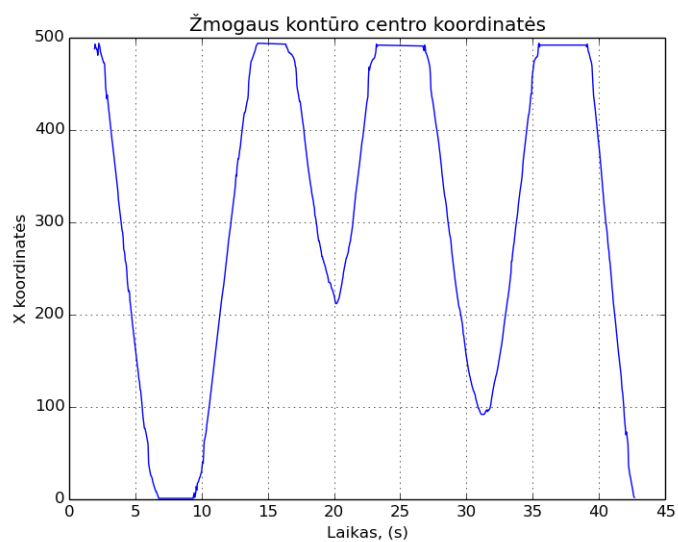


(a) 2-ojo bandymo metu.



(b) 3-ojo bandymo metu.

7 pav. Žmogaus judėjimo grafikas.



8 pav. Žmogaus judėjimas 4-ojo bandymo metu.

6. Rezultatai ir išvados

Šiais laikais kompiuterinė rega greitai vystosi ir tobulėja, todėl atsiranda vis daugiau jos pritaikymo būdų įvairiose srityse, įskaitant mediciną ir biologiją. Taip pat galima tobulinti jau esamus jos taikymus greitėjant kompiuterių našumui ir mažėjant geros kokybės vaizdo kamerų kainoms.

Šiame kursiniame darbe buvo apžvelgtos kompiuterinės regos taikymo galimybės medicinoje bei biologijoje. Taip pat buvo sukurtas kompiuterinės regos programos prototipas, pasižymintis šiomis savybėmis:

1. Nustatytoje aplinkoje atpažįsta judantį objektą – žmogų.
2. Geba sekti judantį žmogų.
3. Naudojant etaloninį objektą, esantį vaizdinės medžiagos kadre, apskaičiuoja kokį atstumą nuėjo žmogus per visą vaizdo įrašo trukmę.
4. Nubrėžia žmogaus kontūro centro judėjimo horizontalioje koordinačių ašyje kreivę per laiką, nuo kada vaizdo medžiagoje judantis žmogus pirmą kartą atpažįstamas iki kol jis buvo atpažįstamas paskutinį kartą.

Atlikus programos bandymus kasdieninėmis sąlygomis matoma, kad tiksliai dabartinės prototipo versijos veikimui reikalingos nesikeičiančio (arba labai silpnai besikeičiančio) apšvietimo sąlygos. Taip pat matoma, kad algoritmo tiksliai veikimui daro įtaką ir kontrastingas fonas už sekamo žmogaus.

Atlikti bandymai parodė, kad sukurto programos prototipo skaičiavimų paklaida maža. Tačiau sklandžiam jo veikimui bet kokia atsitiktinai parinkta aplinka netinka. Reikalinga patalpa, kurioje nesikeistų (arba tik minimaliai keistųsi) apšvietimas, nebūtų daugiau judančių objektų bei būtų etaloninis objektas, kurio išmatavimai yra žinomi.

Ateityje planuojama sukurti programą, kurioje bus patobulintos šios dabartinio prototipo ypatybės:

1. Programos veikimui reikalingas pirmas vaizdo kadras, kuriame nėra judančio objekto – žmogaus, o yra tik fonas. Numatoma pašalinti pirmo kadro be judančio objekto būtinybę.
2. Prototipe daroma prielaida, kad bet kuris didžiausias judantis objektas yra žmogus. Kitoje prototipo versijoje bus vykdomas ne tik asmens atpažinimas, bet ir identifikavimas.
3. Etaloninio objekto su žinomais matmenimis atpažinimo būtinybė vaizdo įvestyje. Numatoma patobulinti šią algoritmo veikimo sąlygą.
4. Dabartinė programos versija gali atpažinti tik vieną judantį objektą ir gali veikti nenuspėjamai tuo atveju, jei judančių objektų daugiau nei vienas. Numatoma programos savybė – daugiau nei vieno žmogaus atpažinimas ir sekimas.

5. Prototipo veikimo tikslumas mažėja, jei vaizdo įvesties kadruose atsiranda triukšmo. Kadangi daugeliu atvejų triukšmas atsiranda dėl kintančio apšvietimo, numatoma patobulinti triukšmo normalizavimą tolesnėse programos versijose.

Santrauka

Kursiniame darbe buvo siekiama sukurti kompiuterinės regos programos prototipą, kuris aptiktų ir sektų žmogaus judėjimą, apskaičiuotų nueitą atstumą ir nubrėžtų judėjimo grafiką. Rašto darbe buvo pateikti kompiuterinės regos taikymo medicinoje pavyzdžiai. Aprašyti pagrindiniai metodai, reikalingi sukurti šią programą. Taip pat apibūdintas prototipo sukūrimas ir jo veikimo algoritmo ypatybės. Galiausiai, šiame darbe buvo išbandytas ir išanalizuotas programos veikimas kasdieninėmis sąlygomis. Kursinio darbo rezultatas – suprogramuotas kompiuterinės regos programos prototipas, kuris seka žmogaus judėjimą, apskaičiuoja jo nueitą atstumą ir sukuria judėjimo grafiką.

Summary

The objective of this course project was to develop a prototype of a computer vision program, able to detect, track the movement and measure the distance of a person and to create a plot of the movement. The possible applications of the computer vision in medicine and biology were also reviewed. General methods, necessary for this program were described in this project. Furthermore, the development and the workflow of the algorithm was characterized. The program was tested and the results were analysed. The outcome of this course project: a developed prototype of a computer vision program which tracks the movement, measures the distance the person has walked and creates a plot of the movement.

Literatūros sąrašas

- [1] G. Bradski and A. Kaehler. *Learning opencv: computer vision with the opencv library.* ” O’Reilly Media, Inc.”, 2008.
- [2] J. Hashemi, M. Tepper, T. Vallin Spina, A. Esler, V. Morellas, N. Papanikolopoulos, H. Egger, G. Dawson, and G. Sapiro. Computer vision tools for low-cost and noninvasive measurement of autism-related behaviors in infants. *Autism research and treatment*, 2014, 2014.
- [3] N. Linder, R. Turkki, M. Walliander, A. Mårtensson, V. Diwan, E. Rahtu, M. Pietikäinen, M. Lundin, and J. Lundin. A malaria diagnostic tool based on computer vision screening and visualization of plasmodium falciparum candidate areas in digitized blood smears. *Plos one*, 9(8):e104855, 2014.
- [4] R. Mandeljc, S. Kovačič, M. Kristan, J. Perš, et al. Tracking by identification using computer vision and radio. *Sensors*, 13(1):241–273, 2012.
- [5] P. M. Polgreen. Using computer vision and depth sensing to measure healthcare worker-patient contacts and personal protective equipment adherence within hospital rooms. In *Idweek 2014*. Idsa, 2014.
- [6] A. Yilmaz, O. Javed, and M. Shah. Object tracking: a survey. *Acm computing surveys (csur)*, 38(4):13, 2006.