

Aukštakalnis, E. Asmenų nenutrūkstamo sekimo tyrimas. Elektronikos inžinerijos *magistro* baigiamasis projektas / vadovas doc. dr. Darius Andriukaitis; Kauno Technologijos Universitetas, Elektros ir Elektronikos fakultetas, Elektronikos Inžinerijos katedra.

Reikšminiai žodžiai:

Sekimas, asmenys, tyrimas, algoritmas, metodas, programa, eksperimentas, savybės.

Kaunas, 2017. 95 psl.

SANTRAUKA

Darbo tikslas - ištirti nenutrūkstamą asmenų sekimą pritaikius pasirinktą metodą ar metodus ir algoritmą, bei panaudoti kompiuterinę programą vaizdo apdorojimui. Tyrimas atliekamas su asmenimis ir su kompiuterine programa *MATLAB*.

Pirmoje šio darbo dalyje literatūros analizėje yra nagrinėjami objektų aptikimui ir sekimui taikomi metodai ir algoritmai. Kai kurie yra tik teoriškai aprašyti, kiti pritaikomi ir praktikoje. Daugelis autorių yra atlikę eksperimentų, kurie turi ne tik privalumų, bet ir trūkumų, tačiau visa tai yra tobulinama. Dauguma asmenų sekimui taikomų algoritmų yra aprašyti tik matematiškai, todėl neturi jokios praktinės vertės.

Antroje dalyje yra išvardinti visi objektų sekimui ir aptikimui taikomi metodai, kurie pritaikomi praktikoje. Taip pat aprašyta objektų aptikimas ir sekimas panaudojant *MATLAB* programinę aplinką. Tai yra objektų aptikimui taikoma veido detekcija su aprėpties langu, bei objekto sekimui taikoma fono substrakcija. Taip pat aprašyti objektų savybėms išskirti naudojami *MATLAB* aplinkoje metodai bei *SVM* klasifikatorius, kuris palygina objektų savybes tarpusavyje.

Eksperimentinėje šio darbo dalyje atliekamas ir aprašomas tyrimas. Iš dviejų sukurtų vaizdų įrašų atliekami bandymai panaudojant *MATLAB* programą su nenutrūkstamu asmenų sekimu. Pirmame vaizdo įrašė yra trys asmenys, o antrame aštuoni. Atliekant eksperimentą su pirmu vaizdo įrašu buvo panaudoti du būdai asmenų savybių nustatymui, tai *LPB* ir *HOG*. Iš šių būdų nustatytas, kuris geriausiai atlieka savybių nustatymą. Atliekant antrą eksperimentą su daugiau asmenų buvo jau naudojama daugiau savybių. Po to buvo atliekamas mašininis apmokymas *MATLAB* programinėje aplinkoje su *SVM*, kuris palygina asmenų savybes ir nustato ar asmuo yra atpažintas ar naujas. Pagal tai yra sudaromi rezultatai ir apibendrinimas apie atliktą eksperimentą.

Aukstakalnis, Edvinas. Research of the continuous tracking of objects. Final project of Electronics Engineering *master* degree/ supervisor doc. dr. Darius Andriukaitis; Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, department of Electronics Engineering.

Keywords:

Tracking, person (humans), research, algorithm, method, program, experiment, features.

Kaunas, 2017. 95 p.

SUMMARY

The aim of the project is to analyse continuous tracking of humans within selected method or methods together with related algorithm and use computer program for image processing. Research is done monitoring people and proceeded in *MATLAB* computer program.

In the first part of the project methods and algorithms used in detection and tracking of the objects are analysed. Some of them are described only theoretically, while others are also applied practically. Most of the authors performed experiments, which show not only advantages, but also disadvantages of the described systems, although improvements are being introduced. The majority of algorithms used in tracking of objects are depicted only mathematically, therefore have no practical value.

In the second part, all the practically applied methods used for observation and detection of objects are being described. Detection and tracking of objects in *MATLAB* environment is also presented. Face detection with region of interest used for detection of the object and background subtraction is applied for object tracking. Moreover, methods used in *MATLAB* environment for picking out features of objects and *SVM* classificatory parameter, which compares them between each other, are analysed in detail.

In the experimental part of the project research is depicted, where it is attempted to apply *MATLAB* written program for continuous people tracking towards the video clip with multiple people appearing in it. During the experiment with first video, used two approaches for picking out the features of people were employed: *LPB* and *HOG*. The best approach for picking out the features was determined. Second video uses more approaches for picking out the features of people. Also, machine learning was done in *MATLAB* environment together with *SVM*, which performs comparison of features and identifies if the person was detected before or is newly appeared one. Depending on this data results and conclusions of the experiment are composed.

SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS

RGB	trijų spalvų sistema: geltona, žalia ir mėlyna (angl. <i>Red-Green-Blue</i>)
MOG	Gauso algoritmų mišinys (angl. <i>Mixture of Gaussian</i>)
HOG	orientuotos gradientų histogramos, naudojama objektų savybių išskyrimui vaizdų apdorojime (angl. <i>Histograms of Gradients</i>)
LBP	vietinis dvejetainis modelis, naudojama objektų savybių išskyrimui vaizdų apdorojime (angl. <i>Local Binary Pattern</i>)
SURF	svarbių savybių greitinimas, naudojamas vaizdų apdorojime objektų aptikimui (angl. <i>Speeded up Robust Features</i>)
SVM	atraminių vektorių mašinos (angl. <i>Support Vector Machine</i>)
BLOB	dvejetainis didelis objektas, naudojamas vaizdų apdorojime (angl. <i>Binary Large Object</i>)
GMM	Gauso mišinių modelis (angl. <i>Gaussian Mixture Model</i>)
IR	infraraudonųjų spindulių kamera (angl. <i>InfraRed</i>)
IP	stebėjimo kamera, paremta interneto protokolu (angl. <i>Internet Protocol</i>)
HSV	HSV spalvų sistema: atspalvis, sotis ir reikšmė (angl. <i>Hue-Saturation-Value</i>)
KNN	algoritmas paremtas artimiausių kaimyninių savybių suradimu ir palyginimu (angl. <i>K-Nearest Neighbour algorithm</i>)

TURINYS

ĮVADAS	13
1. LITERATŪROS ŠALTINIŲ APŽVALGA	14
2. METODINĖ DALIS	38
2.1 Pasirinkto objektų aptikimo ir sekimo būdų tyrimas.....	39
2.2 Pasirintų metodų pritaikymas <i>MATLAB</i> programos aplinkoje.....	43
2.2.1 Objektų detekcija panaudojant aprėpties langus	43
2.2.2 Veido detekcija naudojant <i>MATLAB</i> programą.....	43
2.2.3 Detektuojantis objektus aprėpties langas.....	44
2.2.3 Teigiami ir neigiami atvaizdai.....	45
2.2.4 Keleto objektų sekimas panaudojant aprėpties langus ir fono substrakciją	45
3.1 Savybių išskyrimo tipai	48
2.9 <i>SVM</i> klasifikatorius	49
3. EKSPERIMENTINĖ DALIS	52
3.1 Eksperimentiniai rezultatai	52
3.1.1 Paprogramių (funkcijų) ir pagrindinės programos aprašymas	52
3.1.2 <i>MATLAB</i> mašininis apmokymas	60
3.1.3 Programos algoritmas ir jo aprašymas.....	62
3.1.4 Eksperimentiniai rezultatai su mažu asmenų skaičiumi	65
3.1.5 Eksperimentiniai rezultatai su dideliu asmenų skaičiumi.....	71
IŠVADOS IR PASIŪLYMAI	74
INFORMACIJOS ŠALTINIŲ SĄRAŠAS.....	76
PRIEDAI	82
Priedas 1. Pagrindinė <i>MATLAB</i> programa.....	82
Priedas 2. <i>MATLAB</i> programa asmenų klasifikatorius.....	91
Priedas 3. <i>MATLAB</i> funkcija savybėms išskirti	92
Priedas 4. <i>MATLAB</i> mašininio apmokymo <i>SVM</i> klasifikatorių tikslumai.....	93

IVADAS

Šiais laikais viena iš didžiausių problemų yra terorizmas, vagystės ir žmogaus sauga. Dažniausiai žiniasklaida informuoja apie įvykdytus teroro aktus ir išpuolius, pasikėsinimus, sprogimus, susišaudymus, apiplėšimus ir įkaitų paėmimą. Todėl ieškoma įvairių būdų su tuo kovoti ir apsisaugoti nuo šių pavojingų veiksmų.

Apsaugos sistemos jau naudojamos seniai. Tačiau užbėgti pavojingiems įvykiams į priekį iki šiol dar nepavyksta. Beveik visose viešosiose erdvėse, ypatingai kur yra dideli žmonių susibūrimai, naudojamos paprastos apsaugos sistemos, metalo detektoriai, dūmų detektoriai ir kiti. Stebėjimo sistemos su vaizdo kameromis atsirado jau vėliau. Jos pagerino viešųjų erdvių apsaugą, tačiau aptikti įtartinus asmenis yra kol kas neįmona. Sunku sužinoti, kuris asmuo įjėjęs į viešąją erdvę gali būti pavojingas visuomenei. Jis gali atlikti teroro aktą, vagystę ar kitus veiksmus, kurie gali būti mirtinai pavojingi aplinkiniams. Stebėjimo kameros, įvairūs dujiniai jutikliai ir metalo detektoriai padeda kažkiek apsaugoti viešąją erdvę nuo pavojingų veiksmų, bet yra asmenų, kurie šias sistemas gali praeiti nepatebėti ir neaptikti. Todėl šiais laikais tobulinamos ir kuriamos naujos apsaugos ir stebėjimo sistemos, kurios bent minimaliai sugebėtų aptikti įtartinus asmenis.

Dabar siekiama griežtesnės ir geresnės apsaugos viešosiose erdvėse, visi asmenys turi būti stebimi nuolatos, nuo tada kai jie įjėjęs į viešąją erdvę. Tai sumažina tikimybę atlikti pavojingą veiksmą aplinkiniams, nes asmenys bus stebimi ir sekami skirtingose vietose. Pritaikius specialius metodus ir algoritmus, nuolatos galima sekti asmenų veidus ir siluetus, sukurti skirtingus identifikavimo kodus, o tai sustiprina apsaugą ir palengvina aptikti įtartinus asmenis. Visai tai reikalauja nemažų investicijų, tačiau užtikrina didesnę žmonių apsaugą, o šiais laikais tai yra labai svarbu.

Dabar internete galima rasti įvairių vaizdo įrašų, mokslinių straipsnių ir žurnalų apie asmenų identifikavimą, panaudojant vaizdo stebėjimo sistemas. Aptikti įtartinus asmenis dar kol kas nėra jokio praktinio pritaikymo, tačiau jau yra aprašyta metodų ir algoritmų, kaip turi veikti tokia apsaugos stebėjimo sistema.

Ši tema dabartiniiais laikais apie asmenų sekimą nėra nauja, tačiau aktuali. Užsienyje yra labiau išvysta ir plėtojama, tačiau Lietuvoje apie tai mažai kalbama. Dažniausiai visada tenka jau jau pritaikyti paruoštą produktą. Todėl įgyvendinant tikslą galima pasinaudoti užsienio šalių literatūros šaltiniais ir bandymais, o galutinį produktą sukurti čia. Visa tai gali būti svarbu ne tik savo šaliai, bet ir kitoms šalims.

Magistrinio darbo tikslas yra išanalizuoti ir pritaikyti pasirinktus metodus asmenų detektavimui (aptikimui), identifikavimui ir sekimui panaudojant vaizdo stebėjimo sistemą (vaizdo kameras). Svarbiausia yra sekti nuolatos ir identifiuoti asmenis, bei juos atpažinti daug, kiekvieną kartą pasirodant vaizdo stebėjimo zonoje. Surasti ir ištirti įvairius metodus, kurie galėtų atskirti asmenis ir jiems suteikti unikalų identifikavimo kodą ir visa tai pritaikyti vaizdo apsaugos sistemose. Įvykdyti darbo tikslą, yra keliami šie darbo uždaviniai:

- ištirti objektų detektavimo, identifikavimo ir sekimo metodus;
- pritaikyti pasirinktus metodus judančių objektų stebėjimui;
 - išanalizuoti objektų savybių išskyrimą;
 - pritaikyti mašininį apmokymą objektų identifikavimui ir sekimui;
- atlikti eksperimentus su *MATLAB* programiniu paketu;
- apibendrinti gautus rezultatus ir išvadas.

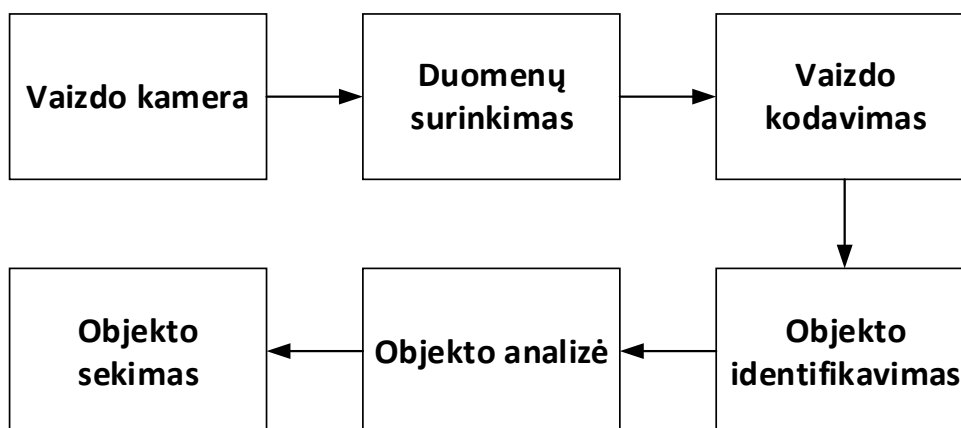
Šio darbo temos aktualumas yra apsaugos susitiprinimas viešosiose erdvesė, žmonių susibūrimo vietose, taip pat apsaugos sistemos patobulinimas vagysčių atvejais (bankai, prekybos centrai, automobilių stovėjimo aikštelės ir kt.) ir nuolatinis asmenų stebėjimas ir sekimas padidinto saugumo zonose. Jei asmuo pasirodo keletą kartų, tai būtų galimybė įspėti, kad tas asmuo yra įtartinas ir apie tai įspėti apsaugą. Šios temos naujumas asmenų sekimas juos identifiuojant unikaliais kodais, kurie yra pastovūs tam pačiam pasirodusiam asmeniui daug kartų stebėjimo zonoje. Visai tai, kad veiktų yra reikalinga atlikti asmenų savybių išskyrimą pagal objektų formą ir spalvas.

Tyrimų objektas yra sekami asmenys ir pasirinkta programa vaizdo apdorojimui. Metodai bus ieškomi ir analizuojami iš mokslinių straipsnių ir mokslinės literatūros.

1. LITERATŪROS ŠALTINIŲ APŽVALGA

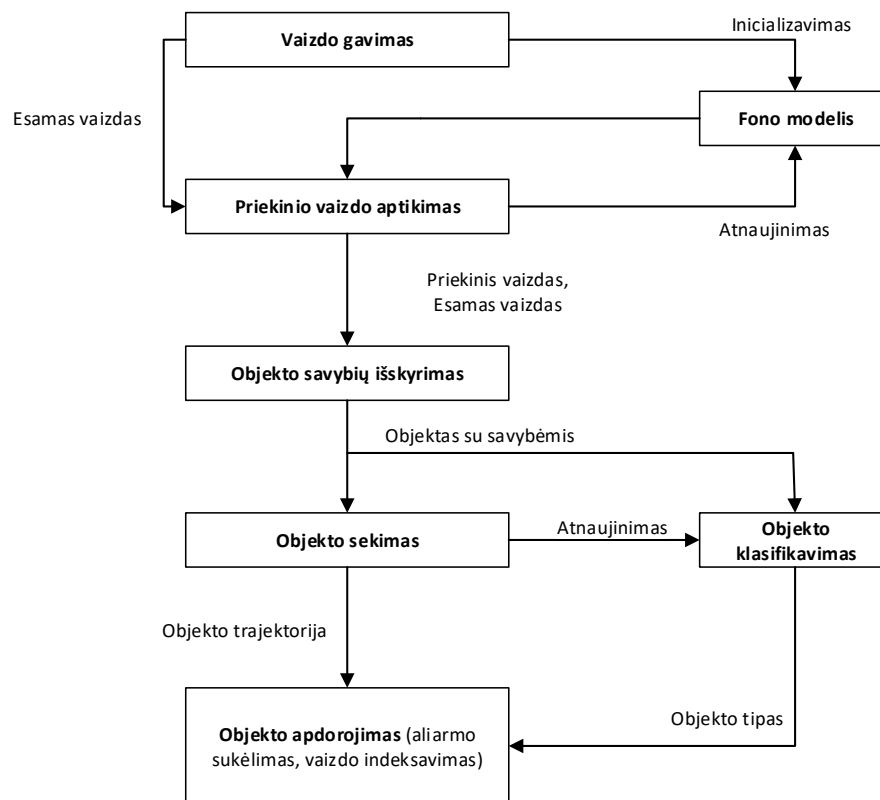
Informacijos apie judančių objektų ir asmenų atpažinimą bei sekimą yra aprašyta moksliniuose straipsniuose, kurie yra skelbiami internete bei moksliniuose žurnaluose. Apie tai yra pristatoma daug algoritmų ir metodų, kurie iki šiol yra analizuojami ir bandomi pritaikyti praktiškai. Tačiau konkretaus pritaikymo iki šiol nėra, tik teoriniai aprašymai ir tyrimai. Todėl yra keliamos problemos ir ieškoma įvairių būdų, kaip tai pritaikyti praktikoje.

Siekiant praktikoje pritaikyti asmenų sekimą reikia pradėti nuo siūlomų metodų ir algoritmų, kurie yra aprašyti moksliniuose straipsniuose ir lietratūroje. Didžiausią dėmesį straipsnių autoriai skiria stebėjimo sistemoms, kurioms taikomi specialūs algoritmai ir metodai, kad sistemos sugebėtų aptikti asmenis bei juos sekti. Šaltinyje [1] yra aprašomas metodas, kaip aptikti ir sekti objektus, panaudojant vaizdo stebėjimo sistemą. Yra pateikta seka, kurioje parodyta, kokia eilės tvarka yra atliekamas objekto sekimas 1.1 pav.



1.1 pav. Stebėjimo sistemos blokinė schema [1]

Tačiau tikslaus šios sistemos aprašymo nėra, nes šis metodas pratikoje nebuvo pritaikytas. Tačiau yra pateiktas algoritmas, kuriuo remiasi autoriai ir kaip galima sukurti tokią stebėjimo sistemą, kuri sugebėtų identifikuoti ir sekti objektus 1.2 pav.



1.2 pav. Objekto identifikavimas ir sekimas panaudojant specialų vaizdo apdorojimo algoritmą su fono modeliu [1]

Pateiktame algoritme yra svarbiausi du aspektai priekinio plano aptikimas (angl. *foreground detection*) ir fono modelis (angl. *background detection*). Toliau atliekamas vaizdo apdorojimas, kurio metu yra išskiriamos objekto savybės, klasifikavimas ir sekimas.

Anksčiau buvo paminėta, kad tokių vaizdo stebėjimo sistemų dar praktikoje nėra taikoma ir tokios sistemos yra apibrėžiamos kaip „neuronų tinklas“. „Neuronų tinklas“ – vienas iš geriausių matematiškai pagrįstų algoritmų. Jis gali būti pritaikytas stebėjimo sistemoms, kurios pagal šį algoritmą sugebėtų indentifikuoti asmenis ar objektus ir juos sekti, bet tai kol kas yra sunku įgyvendinti šioje srityje ir šis algoritmas yra laikomas kaip „juodoji dėžė“. Taip pat autoriai įvardina objekto sekimo būdus [1] tai pagal tašką, siluetą, segmentavimą ir fono substrakciją, tačiau nesigilinama į šiuos būdus.

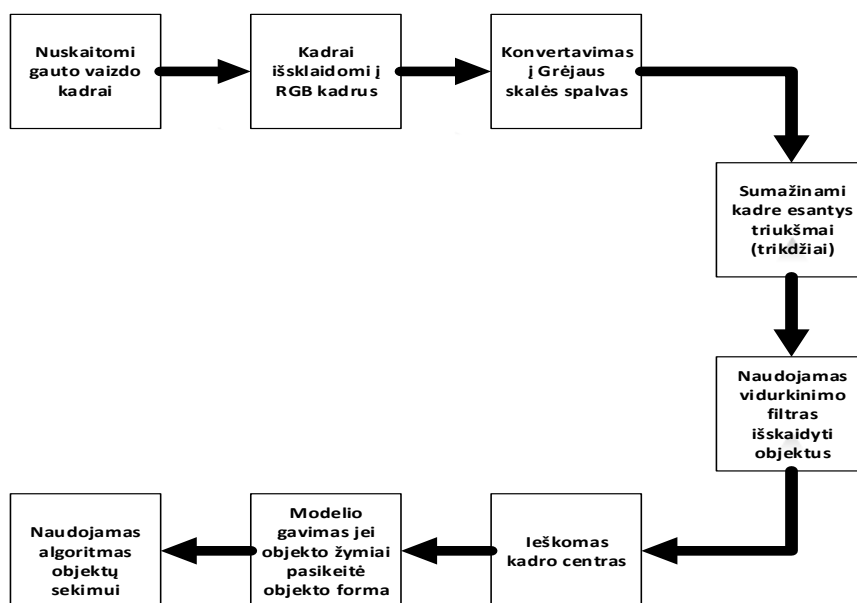
Šaltinio [2], autoriai atliko eksperimentą su vaizdo kamera, kuri sugeba aptikti bei sekti objektus. Bandydami jie susidūrė su problemomis, kurios turi įtakos vaizdo apdorojimui:

- staigus objekto judesys;
- triukšmai, atsirandantys vaizde;
- objekto išvaizdos pasikeitimas ir aplinkos apšvietimas;

- nestabili objekto struktūra;
- okliuzija (objekto uždengimas);
- realaus laiko apdorojimas.

Šios problemos gali sukelti daug sunkumų aptinkant ir sekant objektus. Staigus objekto judesys, kurio vaizdo kamera gali nesususpėti aptikti, triukšmus taip pat reikia kuo labiau sumažinti, kad būtų tikslus objekto aptikimas ir sekimas. Objekto išvaizdo pokyčiai, aplinkos apšvietimas ir nestabili struktūra, tai veiksniai sukeliantys didelį vaizdo apdorojimo vėlinimą, o realaus laiko apdorojimas yra sudėtingas, nes sistemos turi būti greitos, patikimos, kad tiesiogiai būtų atliekamas objektų sekimas. Okliuzijos yra sunku išvengti nes objektas gali dažnai būti už kliūtis, o tai kamera nesugebės pastebėti.




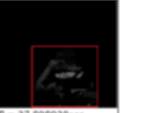







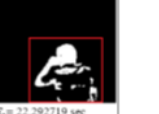
Autorių sukurtos sistemos blokinė schema pateikta 1.3 pav.



1.3 pav. Sukurtos stebėjimo sistemos veikimo seka [2]

Blokinėje schemoje parodyta, kad gaunamas vaizdas iš kameros yra suskaidomas į kadrus (gali būti ir realaus laiko apdorojimas). Sekantis žingsnis – kadru ištraukimas į *RGB* kadrus, konvertavimas į Grėjaus skalės spalvas (juodai baltas vaizdo formatas), triukšmų sumažinimas kadre, naudojamas vidurkinimo filtras, kuris išskiria judančius objektus, objekto centro suradimas, modelio gavimas, jei objekto forma žymiai pakito ir naudojamas įvairų formų atitikimo algoritmas objektų sekimui.

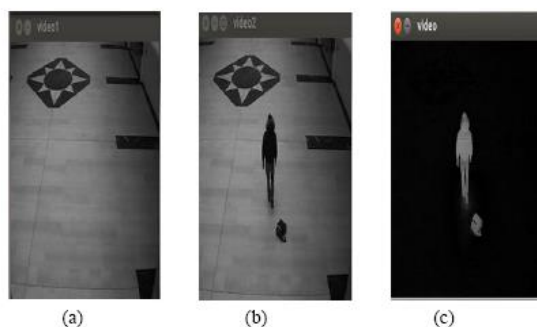
Šaltinyje [3] autoriai aprašo tris išskyrimo, aptikimo ir sekimo metodus, kurie buvo eksperimentiškai atlikti. Tai yra kadru skirtumų metodas (angl. *Frame difference method*), apytikslis medianos metodas (angl. *Aproximate median method*) ir siūlomas autorių metodas – patobulintas kadru skirtumų metodas. Eksperimento rezultatai pateikti 1.4 pav.

Metodai	Kadru skaičius Nuskaityta 20 kadru	Kadru skaičius Nuskaityta 40 kadru	Kadru skaičius Nuskaityta 60 kadru	Kadru skaičius Nuskaityta 80 kadru
Kadru skirtumų metodas	 $E_t = 11.077711 \text{ sec}$	 $E_t = 16.681195 \text{ sec}$	 $E_t = 21.470045 \text{ sec}$	 $E_t = 27.898023 \text{ sec}$
Apytikslis medianos metodas	 $E_t = 9.875688 \text{ sec}$	 $E_t = 19.083763 \text{ sec}$	 $E_t = 28.075134 \text{ sec}$	 $E_t = 36.398993 \text{ sec}$
Pasiūlytas metodas	 $E_t = 6.107433 \text{ sec}$	 $E_t = 11.642959 \text{ sec}$	 $E_t = 17.501209 \text{ sec}$	 $E_t = 22.292719 \text{ sec}$

1.4 pav. Objektų sekimui panaudoti metodai ir apdorojimo laikai [3]

Iš atlikto eksperimento matyti, kad nuskaitant judančio objektų kuo didesnę kadru skaičių, tuo padidėja vaizdo apdorojimo trukmė, tačiau yra tiksliau sekamas asmuo bei jo veiksmai. Autorių siūlomas metodas yra beveik panašus į medianos metodą, tačiau šio metodo privalumas yra kadru apdorojimo laikas (angl. *elapsed time - E_t*). Visi šie metodai remiasi fono subtrakcija (atėmimu). Apie šį metodą jau buvo minėta anksčiau, kad tai yra vienas iš tinkamų vaizdo apdorojimo algoritmų, kuris naudojamas objektų sekimui.

Šaltinyje [4] autoriai aprašo asmenų okliuzijos (uždengimo) ir be priežiūros paliktų objektų aptikimo metodus. Anksčiau buvo minėta okliuzijos problema, todėl autoriai skiria didelį dėmesį okliuzijos aptikimui ir be priežiūros paliktiems objektams viešosiose erdvėse. Tai gali būti palikta kuprinė, bagažas ar krepšys, kurio viduje gali būti įvairių sprogstamųjų ar nuodingų medžiagų, kurios gali pakenkti ar sužaloti aplinkinius. Panaudojant aspsagos vaizdo kameras yra aptinkamas asmuo, kuris palieka savo daiktą 1.5 pav.



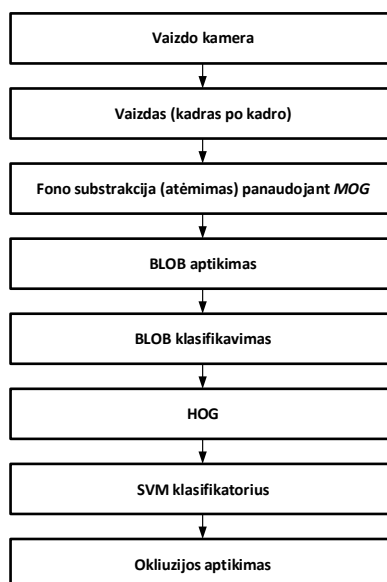
1.5 pav. a – fonas be sekamo asmens, b- fonas su sekamu asmeniu ir c- sekamas asmuo su atlikta fono subtrakcija [4]

Matyti tai, kad fono subtrakcija yra naudinga ne tik asmenų sekimui, bet ir paliktų daiktų aptikimui. Pagal šį algoritmą pirmiausiai yra užfiksuojamas fonas, po to fonas su objektu ir atlikus fono subtrakciją (atėmimą) galima aptikti ir išskirti objektą su jo paliktu daiktu.

Okliuzijos aptikimui yra naudojami du metodai, kuriuos autoriai panaudoja eksperimente:

- *MOG* (Gauso algoritmų mišinys)
- *HOG* (orientuotų gradientų histograma)

Šie metodai yra naudojami kompiuterinėse vaizdo apdorojimo sistemose, tam kad būtų nustatomos asmenų savybės. Pagal tai stebėjimo sistema gali būti apmokoma ir atpažinti tuos pačius objektus keletą kartų ir identifikuoti kaip atpažintą objektą ar asmenį. Okliuzijos atvejus gali išskirti asmenis ir juos sekti po okliuzijos nuolatos. Pagrindinis okliuzijos aptikimo algoritmas pateiktas 1.6 pav.



1.6 pav. Okliuzijos aptikimo seka [4]

Tačiau *HOG* ir *MOG* vien tik šiuos taikyti nepakanka, tam reikia naudoti *SVM* klasifikatorių. Šis naudojamas tam, kad palygintų mažiausiai bent dvi duomenų grupes, kurios yra kiekvieno aptikto objekto duomenys. *SVM* šiuos duomenis lygina kiekvieną kartą ir nustato ar tai yra tas pats objektas ar kitas, todėl okliuzijos aptikimui tai yra geriausias metodas išskirti uždengtus asmenis. Tačiau šie algoritmai yra aprašyti matematiškai ir praktikoje mažai taikomi.

Šaltinio [5] autoriai siūlo naują metodą, kuris pagerina žmonių aptikimą bei sekimą kompleksinėje aplinkoje, kurioje gali būti minios žmonių ar skirtingi fonai. Šį metodą autoriai pritaiko ne tik statinėms vaizdo kameroms, bet ir dinaminėms. Kaip anksčiau buvo minėta pagrindinė problema kompleksinėje aplinkoje yra okliuzija.

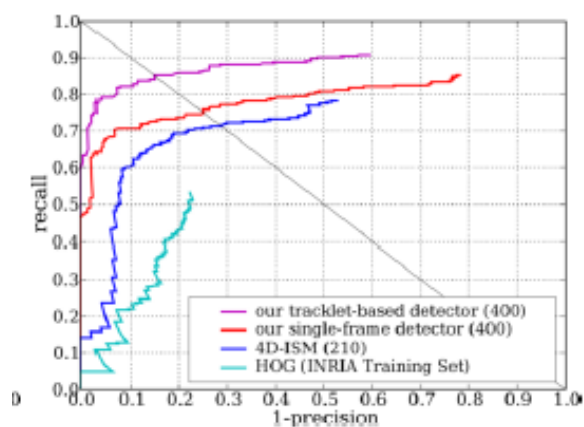
Autoriai atliko eksperimentus su okliuzija panaudojant du skirtingus detektorius (būdus), tai pėsčiųjų detekcija ir asmens galūnių pagrindu paremtas detektorius. Pėsčiųjų detekcija yra paremta objektų suskaidymu į dalis pagal kiekvieną objekto galūnę. 1.7 pav. Kiekviena kūno dalis yra detekuojama skirtingais aprėpties langais, kad būtų kuo tikslesnis objekto išskyrimas iš netvarkingos aplinkos.



1.7 pav. Pėsčiųjų detekcija panaudojant objekto suskaidymą į dalis pagal galūnes [5]

Objekto detektavimas paremtas Gauso hierarchiniu modeliu, kuris pagerina žmogaus detektavimą vaizdo sekoje, bei sumažina okliuzijos galimybę.

Autoriai atliko du eksperimentus. Pirmo eksperimento tikslas buvo palyginti objektų aptikimą naudojant du skirtingus detekcijos būdus. Vienas jų yra vieno objekto aptikimas kadre (angl. *single-frame detector*) ir objekto suskaidymas į fragmentus, kaip anksčiau buvo minėta pagal objekto galūnes (angl. *Tracklet-based detector*). Šie detekcijos būdai buvo palyginti atliekant vaizdų sekas, kuriuose yra dalina arba visiška asmenų okliuzija. Atlikus bandymą, buvo nelengva nustatyti asmenis, kurie yra visiškai nematomi, o kurie yra tik dalinai matomi. Tačiau autoriai, tiksliai įvertino panaudojant vieno kadro detektorius, kuriame asmuo yra matomas, daugiau negu 50 proc, tada asmuo yra fiksuojamas tame kadre. Tačiau objekto suskaidymas į fragmentus yra tikslesnis detekcijos būdas 1.8 pav



1.8 pav. Dviejų detektorių tikslumo palyginimas [5]

Objektų detekcijos būdas suskaidymas į segmentus yra tikslesnis, dėl to, kad dalinai uždengtus asmenis galima išskirti suteikus jiems atskirą aprėpties langą. Jei asmenys yra arti vienas kito, jie nebus sujungti į tą patį aprėpties langą.

Antras eksperimentas buvo atliktas papildyti pirmą eksperimentą, tačiau šiuo atveju vaizdų sekoje buvo detektuojama didesnis asmenų skaičius 1.9 pav. Gauti eksperimento rezultatai parodo, kad jei asmenys yra uždengti kitų asmenų tam tikrą laiką, tada jie neaptinkami po okliuzijos, kai jau yra matomi vaizde.



1.9 pav. Eksperimento rezultatai [5]

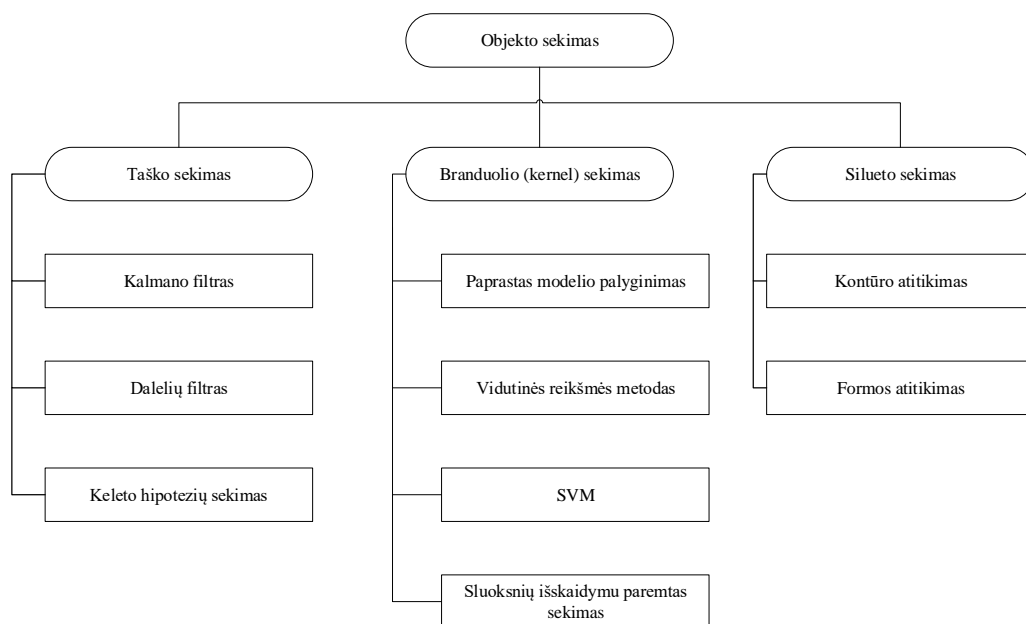
Šaltinyje [6] yra išsamiai aprašyti objektų atpažinimo ir sekimo būdai. Taip pat paminėti sunkumai, su kuriais galima susidurti panaudojant paminėtus metodus. Šaltinyje [1] jau buvo paminėti tokie patys sunkumai tačiau minimo šaltinio autoriai išskiria ir apribojimus ir trūkumus kiekvienam išvardintam metodui 1.1 lentelėje.

1.1 lentelė. Objektų atpažinimo metodų privalumai ir trūkumai [6]

Metodai		Tikslumas	Skaičiavimo laikas	Komentariai
Fono subtrackcija	MOG	Vidutinis	Vidutinis	+ Maži atminties reikalavimai - Sudėtingo fono nesugeba apdoroti
	Apytikslis medianos	Mažesnis už vidutinį	Vidutinis	+ Nereikia imti papildomų kadro iš vaizdo. Sukuria tinkamo fono modelį - Skaičiavimui reikia buferio su neseniai naudotų pikselių reikšmėmis
Optinis srautas		Vidutinis	Aukštas	+ Pateikia visą objekto judėjimo informaciją - Reikalauja didelės apimties skaičiavimų
Kadro skirtumas		Aukštas	Mažesnis už vidutinį	+ Paprasčiausias metodas. Gerai atliekamas kai yra statinis fonas - Fonas turi būti be judančių objektų

Jau anksčiau vienas fono subtrakcijos algoritmų buvo paminėtas taikant *GOM*. Pagal tai galima padaryti išvadą, kad šis metodas yra labiausiai paplitęs atliekant fono subtrakciją.

Toliau pateikta blokinė schema, kurioje aprašyti objektų sekimo būdai 1.10 pav.



1.10 pav. Objektų sekimo metodai [6]

Pagal tai matyti, kad yra trys pagrindiniai metodai, pagal kuriuos galima atlikti objekto sekimą. Tai yra taško sekimas, centro (angl. *kernel*) ir silueto. Branduolio sekimas yra aprašomas daugybe algoritmų, kurie yra sudėtingi. Autoriai šį metodą aprašo, tačiau daug dėmesio neskiriama šiam metodui analizuoti.

Šaltinio [7] autoriai aprašo paliktų objektų automatinį aptikimą, kai objektas yra statinis (nejudantis) tam tikrą laiką yra signalizuojama, kad paliktas įtartinas objektas. Ši sistema susideda iš dviejų kaskadinių (pakopinių) modulių: Bajeso (angl. *Bayesian*) objekto sekimo ir aptikimo bei stebėjimo sistemos modulių. Šis žmonių sekimo modulis sugeba susidoroti su okliuzija bei triukšmais, kurie atsiranda sekant objektus.

Objekto aptikimo ir stebėjimo sistemos modulis remiasi tuo, kad yra pačiame „viršuje“ asmenys, kurie palieka daiktus be priežiūros ir juos sistema identifikuoja tam tikrame vaizdo regione. Paliktas daiktas yra vaizdo pirmame plane ir jis turi būti statiškas, negali jis būti judinamas..

Autoriai buvo atlikę eksperimentą ir nustatė, kad po penkių sekundžių kai asmens daiktas ar objektas buvo paliktas be priežiūros, buvo perspėjamas operatorius vaizduoklio ekrane apie paliktą daiktą 1.11 pav.



1.11 pav. Asmuo bei juo daiktai yra atskiriami į atskirus spalvų regionus pagal kuriuos identifikuoja asmenis bei paliktus jų daiktus [7]

Autoriai pabrėžia tai, kad dar šia sistemą reikia tobulinti, nes sukurti algoritmai negali identifikuoti kiekvieno asmens bei su juo susijusių daiktų. Siekiant tai padaryti, reikia keisti kiekvieno asmens išvaizdos modelį tai yra, kad turėtų daug atskirų spalvos regionų kurie kiekvieną asmenį individualiai sektų ir identifikuotų asmenis.

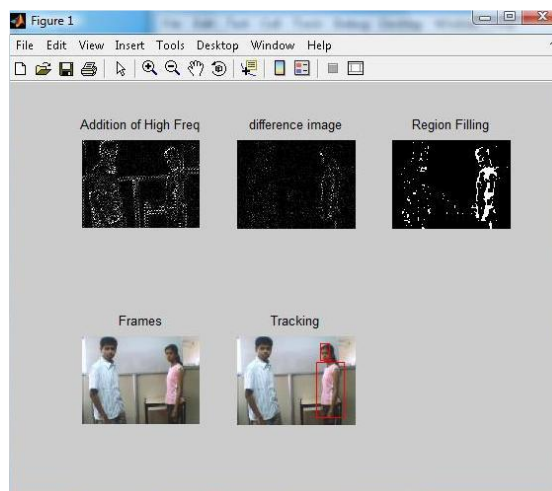
Šaltinyje [8] autoriai išsamiai aprašo apie įtartinių asmenų aptikimą ir sekimą, kai asmuo turi su savimi ginklą. Ginklas yra aptinkama taikant kadru metodą, kai kiekvieno asmens veiksmas su ginklu yra suskaidomas į kadrus. Pagal tai nesunku susekti asmenį, kuris yra pavojingas visuomenei.



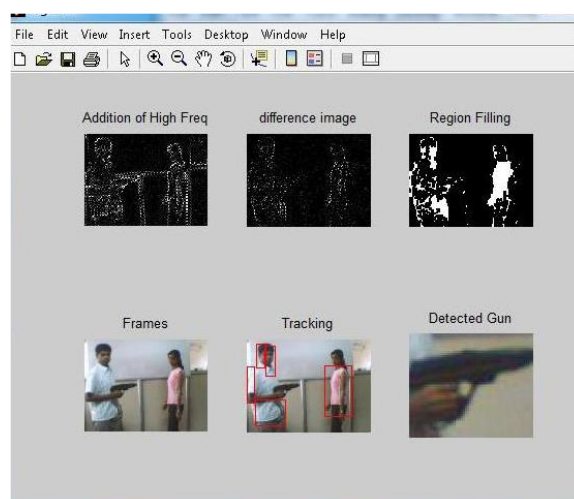
1.12 pav. Eksperimentas, kai įtartinas asmuo yra aptiktas su ginklu, bei taikant kadru metodą sekamas kiekvienas asmens veiksmas [8]

Šio šaltinio autoriai išsamiai aprašo stebėjimo sistemą, kurią sudaro šie elementai: *Web* kamera, *IP* kamera, asmeninis kompiuteris, *Atmega 8535* mikrovaldiklis ir *MATLAB* programinis paketas.

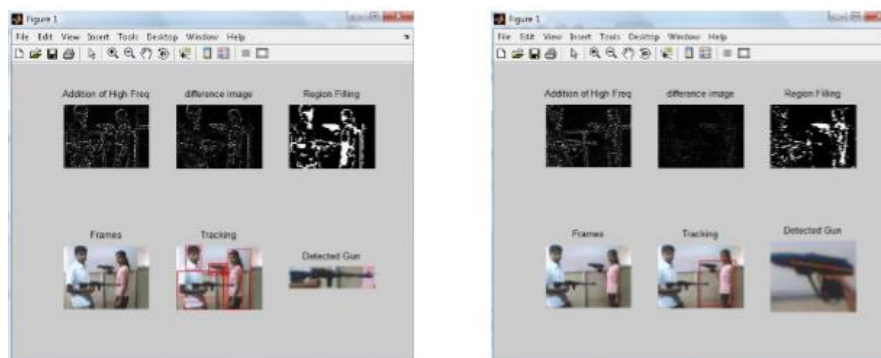
Autoriai šią sukurtą sistemą pritaikė įtartino asmens sekimui (šiuo atveju asmuo esantis su ginklu). Ši sistema identifikuoja (tikrina) asmenis. Jei asmuo ginklo neišsitraukia tada sistema jį laiko neįtartinu ir jo neseka, jei asmuo ginklą išsitraukia, tada sistema asmenį fiksuoja kaip įtartiną ir signalizuoja apie galima pavojų. Buvo atlikti eksperimentai su skirtingais atvejais, kai asmenys buvo be ginklų 1.13 pav., kai vienas iš asmenų buvo aptiktas su ginklu 1.14 pav., kai abu asmenys buvo aptikti su ginklais 1.15 pav., ir kai minioje asmuo buvo aptiktas su ginklu.



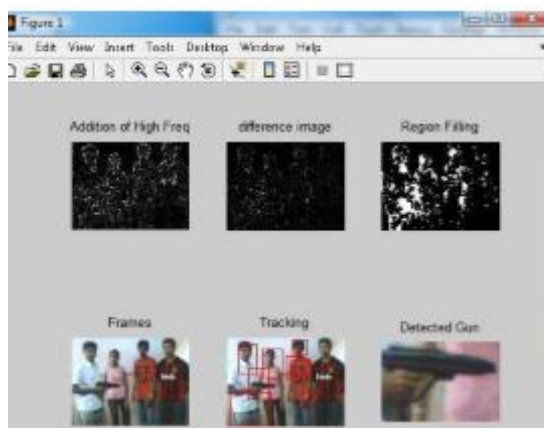
1.13 pav. Eksperimento rezultatai, kai sekamas asmuo yra aptiktas be ginklo, tokį asmenį stebėjimo sistema fiksuoja kaip neįtartiną [8]



1.14 pav. Eksperimento rezultatai, kai sekamas asmuo yra aptiktas su ginklu, tokį asmenį stebėjimo sistema jau fiksuoja įtartiną [8]



1.15 pav. Eksperimento rezultatai, kai abu sekami asmenys yra aptiktas su ginklu, sistema fiksuoja skirtingus ginklus [8]



1.16 pav. Eksperimento rezultatai, kai abu sekami asmenys miniokje yra aptiktas su ginklu, [8]

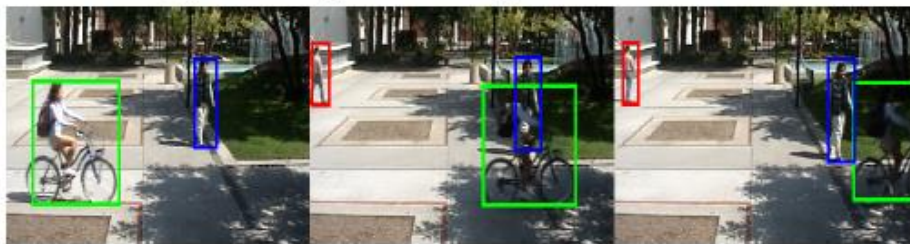
Tačiau ši sistema turi trūkumų. Jei asmuo išeis iš stebėjimo sistemos zonos jis nebus sekamas ir neaptinkamas su ginklu. Taip pat lemia šios sistemos veikimą aplinkos apšvietumas, jei bus tamsu, tada stebėjimo sistema nesugebės aptikti asmens. Kiti trūkumai yra tai, kad taikant algoritmus reikia didelės duomenų bazės, kuri turi atpažinti įvairius ginklų tipus. Šiuo atveju buvo aptinkamas tik pistoletas ir automatas tačiau kitokio ginklo ši sistema nesugebėtų aptikti.

Šaltinio [9] autorių siūlomas metodas yra stebėti objektus ir juos sekti iš daugelio kamerų, kurios yra skirtingose vietose. Šiai stebėjimo sistemai yra pritaikomas jungtinis tikimybinis modelis, todėl įmanoma sekti asmenis ir naudojant išvaizdos modelį. Taip pat galima koduoti asmenis ir juos identifikuoti įvairiomis spalvomis stačiakampiais aprėpties langais, kiekvieną užfiksuotą judantį objektą stebėjimo zonoje.



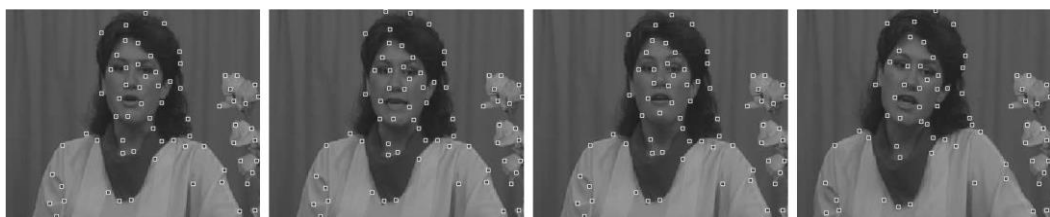
1.17 pav. Okliuzijos problema, kai du asmenys persidengia ir po to vienas iš asmenų nėra identifikuojamas [9]

Pagrindinė problema, kurią autoriai išsprendė yra okliuzija, kai vienas asmuo uždengia kitą asmenį ir po to stebėjimo sistema nesugeba identifiкуoti asmenį kitos spalvos aprėpties stačiakampiu 1.17 pav. Pagal paveikslėlį matyti, kad su dviračiu važiuojantis asmuo „perima“ mėlyną stačiakampį iš kito asmenio ir daugiau jo nebedetektuoja. 1.18 pav. parodyta, kaip po okliuzijos asmuo yra identifikuojamas kitos spalvos aprėpties stačiakampiu.

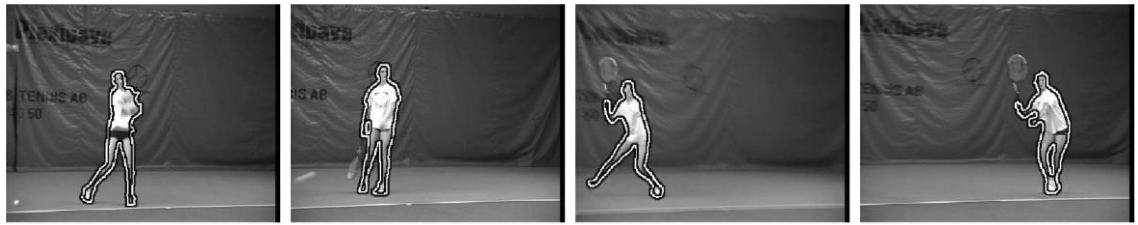


1.18 pav. Okliuzijos problema išspręsta, po okliuzijos abu asmenys identifiкуojami skirtingų spalvų stačiakampiais [9]

Straipsnyje [10] autorai pateikia daug metodų, kurie taikomi objektų stebėjimui. Autoriai skirsto asmenų sekimo metodus į tris tipus: primityvus geometrinis modelis, naudojant kontūrus ir nustatant objektą pagal taškus. Tačiau kiekvienas metodas turi trūkumų. Taikant objekto sekimą pagal taškus, reikia objektą aptikti kiekviename naujame kadre ir pagal tai jį atpažinti, o taikant objekto sekimą pagal geometriją ar kontūrus, reikia iš naujo kiekviename vaizde identifiкуoti objektą ir tik po to su stebėjimo sistema, kuri pagal pasirinktą algoritmą apdoroja vaizdą ir objektas yra sekamas.



1.19 pav. Asmens detektavimas ir sekimas taikant taškų metodą [10]



1.20 pav. Asmens detektavimas ir sekimas taikant kontūrų metodą [10]

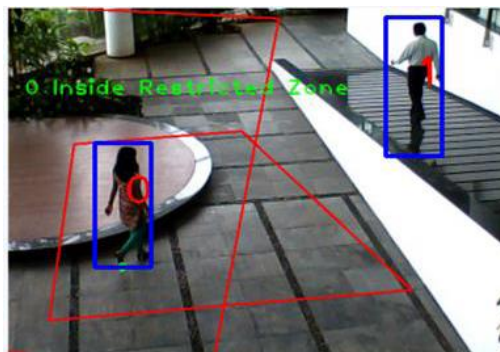
Šaltinyje [11] autoriai pateikia stebėjimo sistemas, kurios atlieka visas pagrindines funkcijas objekto sekimui. Tai yra objekto judesio aptikimas, objekto klasifikavimas, sekimas ir objekto elgsenos analizė. Tačiau tyrimai prasideda tik nuo algoritmų ir funkcijų apibrėžimų, ir tai siekiama pereiti prie sudėtingos aplinkos analizės, kur yra svarbiausia aptikti asmenis ir jų elgsenas.



1.21 pav. Pagrindiniai etapai įtartinų asmenų sekimui [11]

Šiame straipsnyje yra svarbu ne tik asmenų sekimas, bet ir priegos kontrolė, pagal kurią bus identifikuojami asmenys, taip pat paliktų daiktų aptikimas ir signalizacija. Siekiamą tokią apsaugos sistemą įgalinti viešosiose ar privačiose erdvėse.

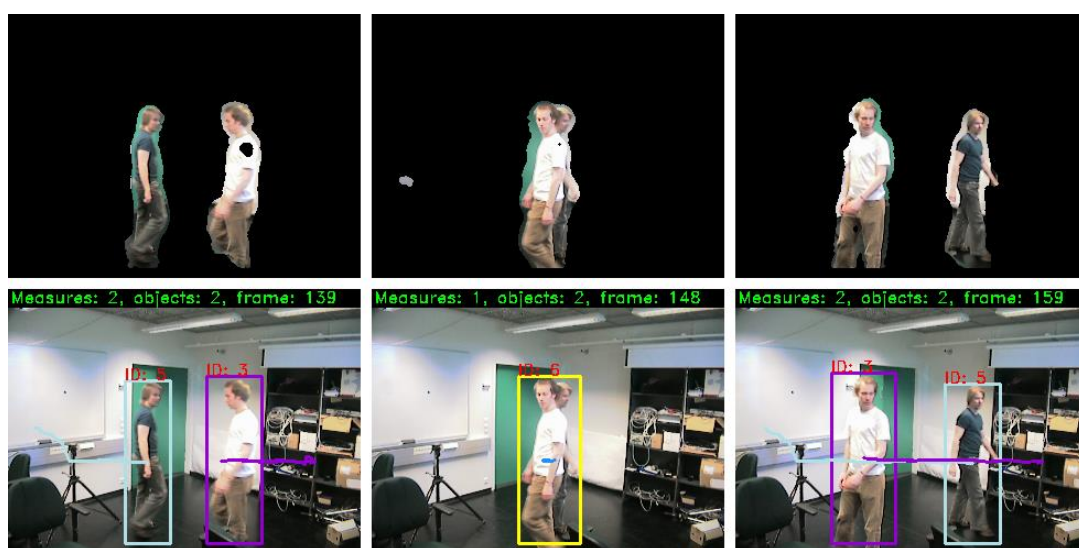
Daugiakamerinė stebėjimo sistema yra efektyvus būdas stebėti teritoriją iš įvairių pusių. Autoriai [12] panaudodami daugiakamerinę stebėjimo sistemą bei pritaikę sudėtingus algoritmus *GMM* (Gauso mišinių algoritmus), šią sistemą naudoja įtartinų asmenų sekimui ir aptikimui.



1.22 pav. Asmens aptikimas apribotoje zonoje [12]

Padidinto saugumo zonose, autoriai sugalvojo panaudoti vaizdo stebėjimo kameras, kurios stebi apribotą zoną (angl. *restricted zone*), į kurią asmenys negali įeiti 1.23 pav. Asmuo įžengęs į šią zoną, sistema įjungia signalizaciją ir įspėja, kad kažkas yra neleistinoje zonoje. Pagal tai galima spręsti, kad asmuo gai būti įtartinas ir tai reiktų atkreipti dėmesį.

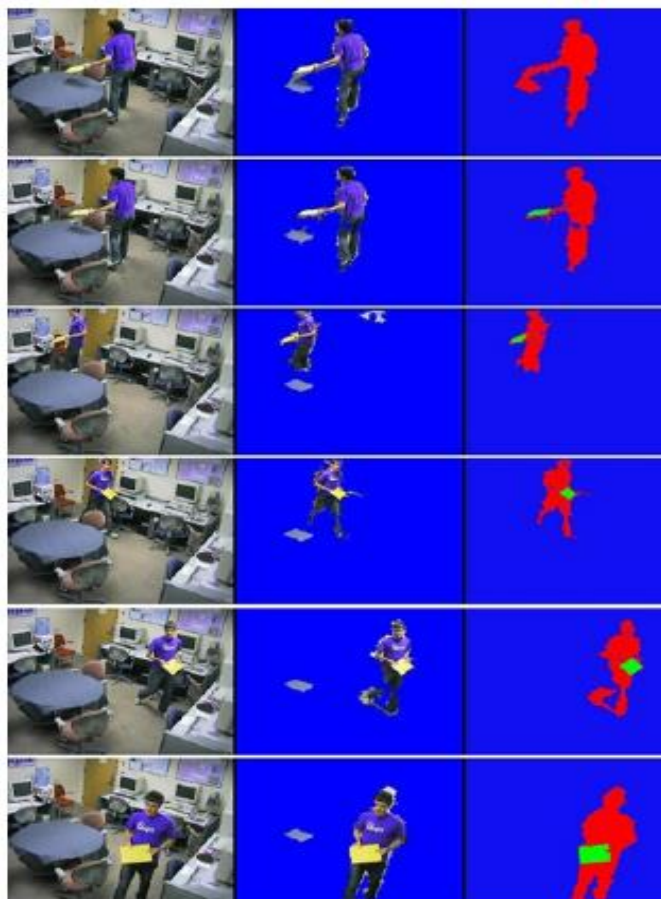
Šaltinyje [13] yra aprašytas asmenų sekimo būdas, panaudojant spalvų kombinacijas, tekstūros ir judėjimo savybes bei tekstūros pagrindu paremta fono substrakcija (atėmimas). Ši sistema gali sekti kelis asmenis skirtingose sąlygose, kai sistemos greitaveika yra 10 -15 kadru per sekundę ir panaudojus asmeninį kompiuterį, kurio procesorius turi būti 3 GHz taktinio dažnio 1.23 pav.



1.23 pav. Dviejų asmenų sekimas panaudojant fono substrakciją bei spalvų kombinacijas [13]

Ateityje ši sistema bus patobulinta. Ši sistema sugebės sekti asmenis esant, įvairiam apšvietimui, nes šiuo metu tai yra didelė problema. Dienos metu asmenis yra nesunku aptikti ir sekti, tačiau nakties metu tai yra sudėtinga. Taip pat bus sekami ne tik asmenys, bet ir kitokių formų objektai, kaip mašinos, dviračiai, motociklai ir kita.

Autoriai [14] pristato stebėjimo sistemą kuri aptinka ir seka asmenis, bei su jais turimus daiktus. Naudojant specialius spalvų algoritmus ir fono substrakciją išskiria asmenis į tam tikrus spalvų derinius ir su jais susijusius daiktus 1.24 pav. Jei asmuo turi su savimi daiktą, panaudojus fono substrakciją, asmuo išskiriamas iš aplinkos suteikiama jam spalva ir kita spalva suteikiama asmens turimam daiktui, šiuo atveju knygai.



1.24 pav. Asmens sekimas panaudojus spalvų algoritmus bei fono substrakciją [14]

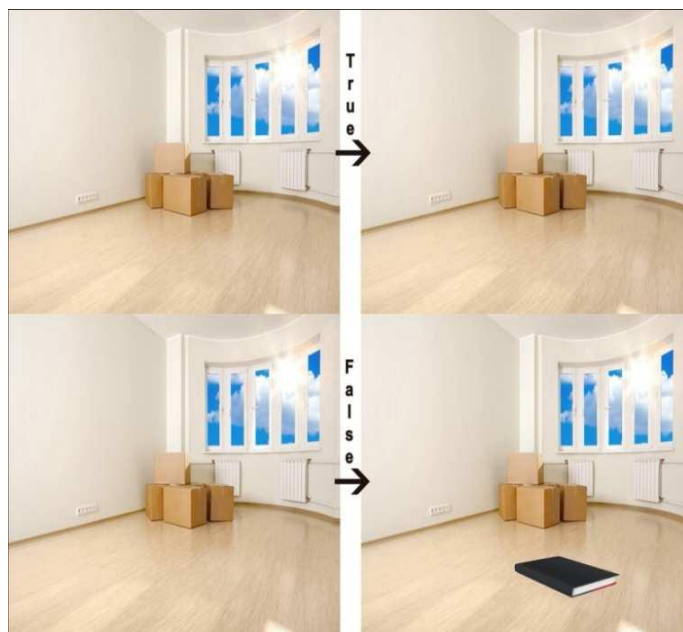
Tačiau ši sistema turi ir trūkumų. Jei yra keli asmenys vienoje vietoje, tada sistema nesugebės priskirti skirtingų spalvų kiekvienam asmeniui. Taip pat viena iš problemų jei asmenys yra panašiai apsirengę, tada sistemai pagal panašias išvaizdos savybes bus sunku išskirti asmenis. Todėl ši sistema yra tobulinama.

Šaltinyje [15] pateikta uosto apsaugos stebėjimo sistema, kuri stebi įtartinų asmenų išsibrovimus į teritoriją ir pakrantes. Čia yra naudojami aktyvūs akustiniai ir IR spindulių jutikliai, kurie ypač reikalingi stebėti nakties metu.



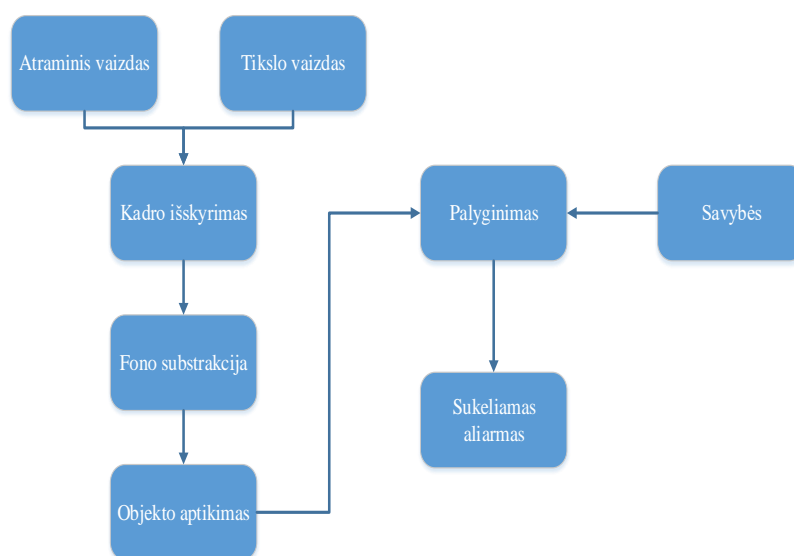
1.25 pav. Naudojant IR kamerą nakties metu aptinkami asmenys [15]

Straipsnyje [16] aprašomas objektų aptikimo metodas panaudojant statinę kamerą. Bandymas buvo atliekamas mažoje patalpoje, tačiau galima ir didesnėse patalpose taikyti šį metodą. Autorių siūlomas algoritmo veikimas yra toks, sudaromi du vaizdai, viena iš jų yra tikslo vaizdas su daiktu, o kitas atraminis vaizdas be daikto. Atlikus jų skirtumą gaunamas tik vieno daikto vaizdas 1.26 pav



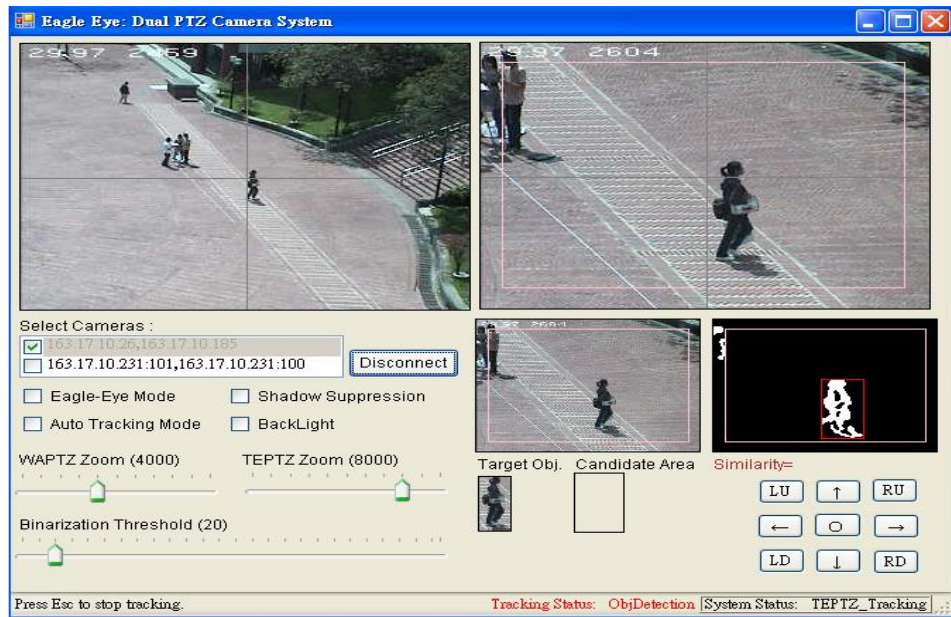
1.26 pav. Viršuje yra atraminis vaizdas be knygos, apačioje tikslo vaizdas su objektu [16]

Toliau pateiktas pagrindinis algoritmas:



1.27 pav. Algoritmas, kuris naudojamas objektų aptikimui [16]

Straipsnyje [17] yra aprašyta stebėjimo sistemos, kurioje naudojamos aktyvios stebėjimo kameros su aukšta raiška, kad būtų įmanoma priartintus vaizdą, aiškiai matyti asmenis. Dabartinės stebėjimo sistemos naudoja tik pasyvias kameras, kurios nesugeba priartinti vaizdo dėl ribotos raiškos. Tačiau naudojant aktyvias kameras ir priartinus vaizdą prie įtartino asmens yra didelė tikimybė, kad bus nepastebėti aplinkui esantys įtartini asmenys. Šio šaltinio autoriai pristato *Eagle Vision* (liet. *erelio akis*) stebėjimo sistemą 1.28 pav.



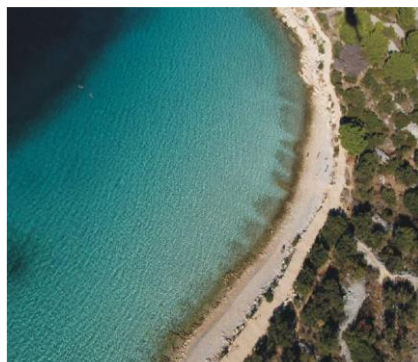
1.28 pav. *Eagle Vision* stebėjimo sistema [17]

Ši sistema yra ypatinga tuo, kad galima priartinti vaizdą, stebėti asmenį bei sekti jį, taip pat matyti kas ir aplinkui vyksta. Naudojant aukštos raiškos kameras galima kuo aiškiau matyti asmenis, bei pritaikius algoritmą panaudoti fono substrakciją, apie kurią buvo kalbėta anksčiau, asmenų sekimui. Tačiau labai didelės teritorijas stebėti su šia sistema yra sunku, nes yra būtina susieti bent keletą aktyvių kamerų, kurios stebėtų visą teritoriją.

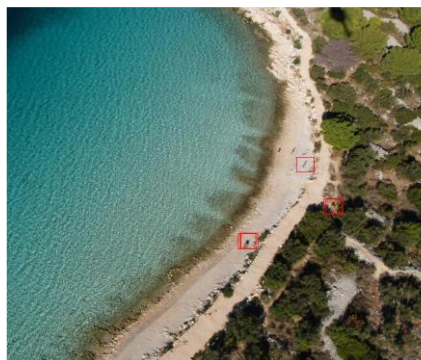
Straipsnyje [18] autoriai pristato dviejų etapų segmentavimo metodą, kuris skirtas aptikti įtartinius asmenis didelėse teritorijose, išskyrus gyvenvietes ir miestus. Naudojama aukštos raiškos skaitmeninė kamera, kuri gali aptikti objektus daugiau negu už 200 metrų. Gautas vaizdas iš kameros suskirstomas į keletą mažesnių dalinių vaizdų ir taip paruošiamas vaizdas apdorojimui.

Pirmas apdorojimo etapas yra vaizdo dalių apdorojimas į *YcbCr* spalvų modelį. Mėlynos spalvos chromas ir raudonos spalvos chromas naudojamas vaizdo segmentavimui. Kiekviena vaizdo dalis yra segmentuojama ir sugrupuojama. Po to sekantis etapas yra apdorojimo etapas,

kuriame naudojama sugrupuota vaizdų informacija (seka) yra verčiama į duomenų rinkinį. Galiausiai sprendimų priėmimo modelis įvertina, kuriame vaizde yra labiausiai tikėtinas aptiktas objektas. Pagal tai aptinkamas objektas ar asmuo.



1.29 pav. Originalus vaizdas [18]



1.30 pav. Vaizdas su aptiktais asmenimis [18]

Šaltinio [19] autoriai naudoja dalijimo ir normalizuotą kros-koreliacijos (angl. *Partitioning and Normalized Cross Correlation - PNCC*) algoritimą, kuris aptinka judančius objektus. Šio algoritmo vykdymo laikas yra pakankamai mažas. Tai padidina apdorojamo vaizdo greitaveiką. Šio šaltinio autoriai naudoja stebėjimo sistemas stebėti įtartinus asmenis bei aptikti neveiklius (angl. *idle*) objektus, gatvėse, perėjose ir keliuose 1.31 ir 1.32 pav.



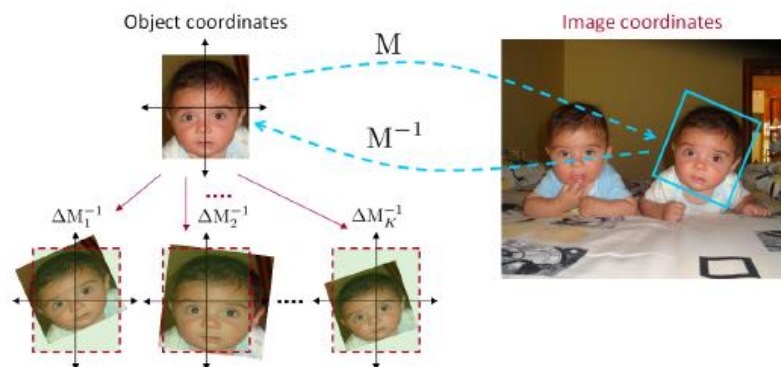
1.31 pav. Objekto aptikimas ir sekimas panaudojant *PNCC* [19]



1.32 pav. Neveiklaus (*idle*) objekto aptikimas, kuriuo metu įsjungia aliarmas [19]

Straipsnyje [20] autoriai plačiai aprašo objektų aptikimo ir sekimo būdus. Tie metodai jau yra anksčiau paminėti, kaip objekto aptikimas ir sekimas pagal kontūrus ir panaudojant fono substrakciją. Tačiau daug dėmesio skiriama okliuzijos tipams. Okliuzija gali būti dalinė arba pilna, paties objekto okliuzija, tarp objektinė okliuzija bei pagal fono scenos struktūrą. Paties objekto okliuzija yra tada kai viena bent iš objektų dalių uždengia kitą objekto dalį, tarp objektinė okliuzija gali būti tik tada kai vienas su kitu objektai prasilenkia ir fono okliuzija atsiranda tik tada, kai objektas yra uždengtas kito statinio objekto (stulpas, medis).

Taip pat aprašoma objektų detekavimo tipai ir veido detektavimas pagal koordinates. 1.33 pav. parodyta, kaip asmens veidą galima pagal koordinates aptikti, kai veidas gali būti įvairiose pozicijose. Tai yra labai svarbu, nes stebimo asmens veidas, niekada nebus vienoje pozicijoje, jis gali būti pasisukęs į vieną ar kitą pusę.



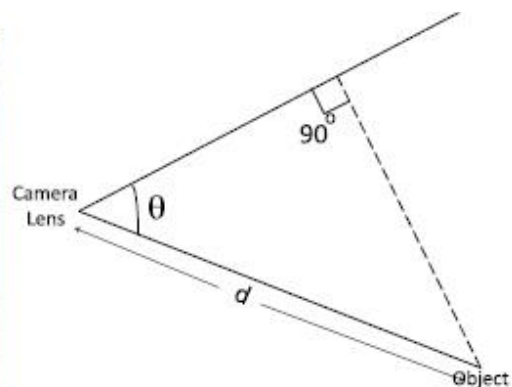
1.33 pav. Asmens veido detekcija, kai keičiama veido pozicija pagal koordinates [20]



1.34 pav. Objekto pagal kontūrus aptikimo būdai [20]

1.34 pav. parodyta pagal kontūrus objekto aptikimo būdai pagal kontūrus. Pirmas iš būdų yra tai, objekto apibrėžtas regionas, kai objektas yra apribotas tam tikru vaizdo regionu ir sekantis yra eliptinis regionas, kai asmens vaizdas yra išskiriamas elipse, o jos centras yra objekto centras. Kitas būdas tai yra išskirtas tik objekto siluetas, sekantis būdas yra, kai objektas yra išskirtas į kūno dalis ir paskutinis būdas yra objektas išskirtas pagal skeleto struktūrą.

Šaltinio [21] autoriai pristato realaus laiko judančio objekto aptikimo ir sekimo sistemą. Šią sistemą sudaro robotas su dviem kameromis 1.35 pav., bevielio ryšio modelis ir asmeninis kompiuteris.



1.35 pav. Robotas su dviem kameromis ir kameros stebėjimo kampo schema [21]

Robotas fiksuoja su kameromis judančius objektus ir perduoda bevielio ryšio moduliui, kuris bevieliu ryšiu perduoda duomenis į kompiuterį. Kompiuteris pagal užsiduotą algoritną apdoroja gautus vaizdus. Ši sistema vadinama realaus laiko sistema, nes yra greita (greitai perduodami duomenys į kompiuterį su mažais vėlinimais) tačiau yra keletas veiksnių, kurie gali įtakoti šios sistemos greitaveiką. Tai gali būti didelis atstumas iki objekto, objekto judėjimo greitis ir kameros kampas, kurie gali įtakoti roboto užfiksuoto vaizdo apdorojimo laiką. Tai pat greitaveiką įtakoja vaizdo fiksavimo dažnis, bevielio ryšio pralaidumas, serverio greitis ir gautas duomenų kiekis, kuris priklauso nuo fono sudėtingumo. Taikiant dideliems atstumams, ši sistema netinka, tačiau naudoti mažose patalpose sekti ir aptikti objektus, ši sistema yra tinkama.

Šaltiniuose [23] ir [24] aprašoma paliktų daiktų be priežiūros viešosiose erdvėse aptikimą. Šaltinyje [23] aprašomas neprižiūrimų arba paliktų daiktų aptikimas panaudojant laikinę logiką. Tai yra algoritmas, kuris gali aptikti daug paliktų objektų. Paveikslėlyje 1.36 pav. pavaizduota kadrų seka, kurioje yra parodytas paliktas bagažas. Paveiksle *c* idenfikuojamas bagažas, kuris yra jau palikas be asmens. Paveiksle *d* sistema fiksuoja, kad šeimininkas negrįžta per tam tikrą nustatytą laiko tarpą ir įjungiamas šešiasdešimt sekundžių laikmatis. Po šešiasdešimt sekundžių įsijungia aliarmas, kuris įspėja apie paliktą bagažą ir sistema jį fiksuoja raudonu stačiakampiu. Jei šeimininkas grįžta ir pasiima bagažą, tada sistema išjungia aliarmą ir daugiau bagažo nebefiksuoja.



1.36 pav. Kadrų seka, kuriose parodyta paliktas be priežiūros bagažas [22]

Šaltinyje [23] taip pat aprašytas tų pačių autorių laikinis algoritmas, kuris skaičiuoja laiką nuo tada, kai šeimininkas be priežiūros palieka bagažą ar kitus savo daiktus, t'čiau įvertina šiuo atveju ir okliuziją. Stebint paliktą objektą viešojoje erdvėje, kur yra daug žmonių, šie gali bet kuriuo metu užstoti objektą 1.37 pav.



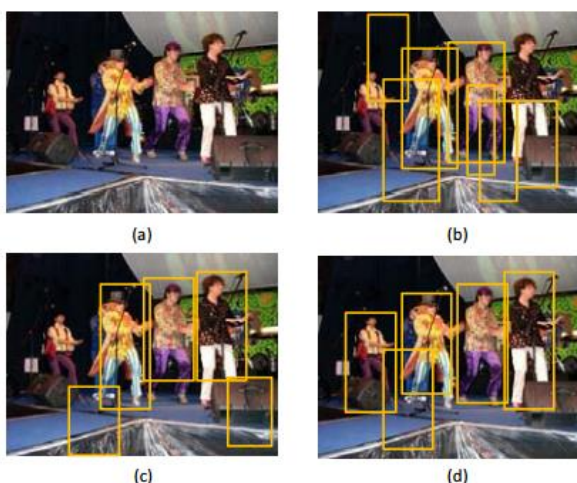
1.37 pav. Kadru seką, kuriose parodyta paliktas be priežiūros objektas, kai šį uždengia asmenys [23]

Straipsnis [24] yra susijęs su [23], kad čia taip pat yra neprižiūrimų ar paliktų objektų aptikimas kompleksinėje (sudėtingoje) aplinkoje. Šio straipsnio autoriai pristato realaus laiko stebėjimo sistemas, kurios aptinka paliktus objektus viešosiose erdvėse. Objekto detekcijai yra naudojamas Gauso mišrusis algoritmas. Tačiau šis metodas tinka tik statiškiems objektams, kurie yra nejudantys. Šis algoritmas veikia taip, iš pradžių segmentuojami objektai, kurie yra palikti, po to lyginamas visas bendras vaizdas su fonu.



1.38 pav. Aptiktas paliktas objektas, kuris po to yra uždengiamas [24]

Autoriai [26] pristato viena iš patobulintų metodų su *HOG-LPB* savybių taikymų objektų aptikimui, tai padidintas vietinės struktūros *HOG-LPB* detektorius. Pirmiausiai pristatė jau anksčiau naudojamus metodus tokius kaip *LSLBP* (angl. *local structured local binary pattern*) ir *LSHOG* (angl. *local structured histogram of gradients*). Eksperimentiškai autoriai įrodė, kad patobulinant šiuos abu metodus, galima tiksliai nustatyti objekto vietos struktūrą, bei pagerinti objekto detekciją. 1.39 pav. pateikta objektų detekcijos rezultatai.



1.39 pav. Objektų detekcija: a – originalus kadras, b – *SVM+HOG* detekcija, c – *SVM+LPB* detekcija, d – autorių siūlomas metodas [26]

Pagal pateikto paveikslio rezultatus matyti, kad autorių siūlomas metodas yra efektyvus. Tiksliai detektuojami objektai ir nėra jokių netikslumų. Šį metodą pritaikė programoje *PASCAL VOC2007*, kuri tinka objektų detektavimui.

Autoriai eksperimentą suskirstė į tris dalis: pirmoje dalyje tikrino vietos struktūrinį aprašymą pagal *LSHOG* savybes, antra dalis su *LSLBP* savybėmis įvertinant efektyvumą trilineinės interpoliacijos ir objekto vietos struktūrinį aprašymą, trečia dalis rezultatų palyginimas derinant skirtingas programas ir ketvirta dalis siūlomo metodo (padidinto vietos struktūrizuoto *HOG-LPB*)

paremtu objektų aptikimu, panaudojant *PASCAL VOC2007* programą. Čia taip pat yra vykdomas *SVM* mašininis apmokymas, su kuriuo yra palyginamos savybės.

\|

2. METODINĖ DALIS

Šioje dalyje aprašomi objektų aptikimo ir sekimo metodai, kurie buvo paminėti literatūros apžvalgoje. Taip pat bus aprašomas praktinis pritaikymas panaudojant kompiuterines programas. Naudojami būdai ir algoritmai programose, funkcijos ir programų šablonai. Ši dalis suteiks informacijos atlikti eksperimentams.

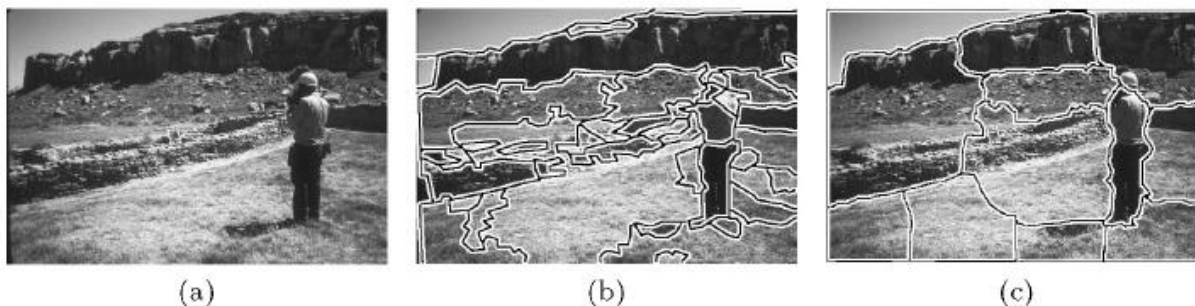
2.1 Objektų aptikimo ir sekimo metodai

Moksliniuose straipsniuose ir literatūros šaltiniuose yra aprašyta metodų ir apie objektų aptikimą ir sekimą. Pagal tuos metodus yra sudaryta daug algoritmų, kurie aprašyti matematiškai, tačiau ne visi yra realizuoti. Šių šaltinių [10], [6], [14], [19] ir [21] autoriai plačiausiai aprašo algoritmus ir metodus, kurie jau eksperimentiškai yra atlikti. Populiariausi objekto detekcijai (aptikimui) ir sekimui naudojami metodai pateikti 1 lentelėje:

1 lentelė. Objektų aptikimo būdai

Objektų aptikimo būdai	Objektų sekimo būdai pagal aptikimo būdus
Objekto segmentavimas	Vidurkio pokyčio grupavimas (<i>angl. Mean-Shift Clustering</i>)
Objektų taškų detektavimas	Vaizdo sukarpymas į segmentus
Objekto centro detektavimas	Fono substrakcija (atėmimas)
Objekto veido detektavimas	Asmens sekimas pagal aprėpties langą

Pateikti segmentavimo būdai (2.1 pav.)



2.1 pav. Segmentavimo būdai: a- vaizdo segmentavimas, b- vidurkio pokyčio segmentavimas ir c- vaizdo sukarpymas į dalis [10]

Objekto sekimui naudojami būdai pateikti 2 lentelėje:

2 lentelė. Objektų sekimo būdai

Sekimo būdai	Metodai (algoritmai)
Taško sekimas	Deterministinis, statistinis
Branduolio (<i>kernel</i>) sekimas	Šablonu ir išvaizda pagrįstas modelis, multi- peržiūrų išvaizdos modelis
Silueto sekimas	Kontūro pokyčiai, formų atitikimas
Fono substakcija	Objekto išskyras iš fono

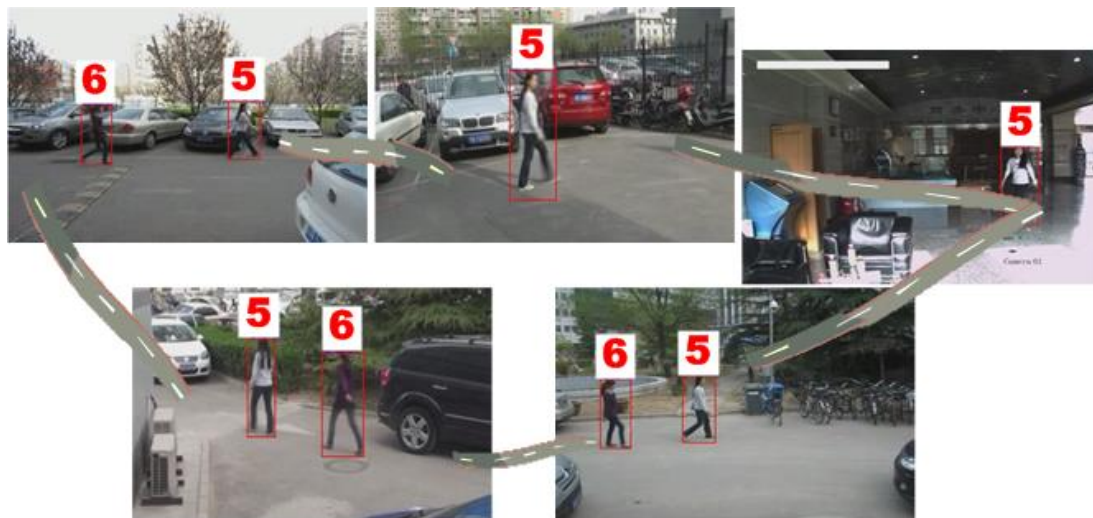
Praktikoje labiausiai paplitęs fono substrakcijos metodas. Tai vienas iš dažniausiai praktikoje taikomų metodų sekti asmenis. Šis metodas tiksliai atkuria asmens formą ir siluetą bei judesius. Tačiau šis metodas reikalauja didelių skaičiavimų ir algoritmų.

2.1 Pasirinkto objektų aptikimo ir sekimo būdų tyrimas

Anksčiau paminėtų objektų aptikimo sekimo metodų yra daug, tačiau ne visi yra praktiškai bandyti. Ankstesniame skyriuje buvo išvardinti atskiri objekto aptikimo ir sekimo metodai. Tačiau iš jų pasirinkti metodai yra objektų aptikimas panaudojant aprėpties langus, bei fono substrakcija. Šie metodai buvo pasirinkti dėl to, kad jie yra tikslūs ir tinkamiausi naudoti objektų aptikimui ir sekimui. Taip pat jie yra aprašyti *OpenCV* ir *MATLAB* programinėse aplinkose.

Pagrindinės problemos sekti nenutrūkstamai objektą tai yra sudėtingi algoritmai, kurie naudojami vaizdo apdorojimui. Taip pat yra susiduriama techninėmis problemomis, tokiomis, kaip tinkamų vaizdo kamerų įrangos ir serverių parinkimas. Realios laiko sistemoms reikia greitos sistemos tam kad vaizdas būtų apdorojamas realiu laiku. Aptikti asmenį užtenka statinės kameros, kuri yra nejudri ir platus stebėjimo kampas. Tačiau sekti asmenį, jau neužtenka naudoti statinės kameros, kuri viską stebi vienoje vietoje. Šiuo atveju reikia naudoti dinaminę kamerą, kuri tam tikru kampu galėtų sukstis, ir taip būtų sekamas asmuo nuolatos. Tačiau dar yra kita problema sekant objektus, neužtenka vien tik juos stebėti vienoje patalpoje ar zonoje. Šiuo atveju turi būti asmenys sekami visoje viešojoje erdvėje. Tam reikia, kad tarpusavyje būtų sujungtos kameros ir viena bendra stebėjimo sistema 2.2 pav. Bet tada galima išvengti dinaminių kamerų ir naudoti statines kameras, kurias sumontavus tam tikrais kampais ir atitikų kaip vieną dinaminę kamerą. Naudojant dinamines kameras būtų neišvengiama problema su kintančiu fonu. Naudojant fono substrakciją reikalingas statinis fonas. Šiuo atveju fonas nebus statinis ir fono substrakcijos metodo

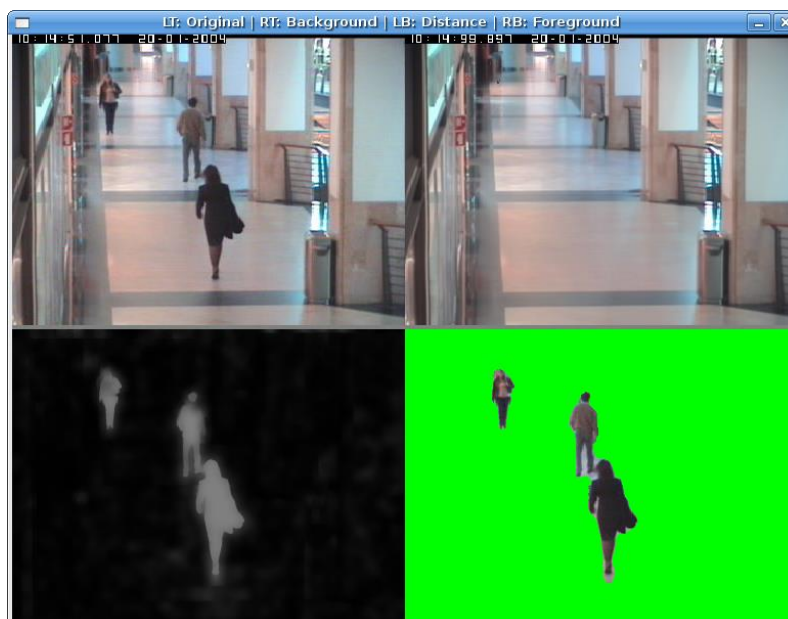
neišeitų panaudoti. Tačiau pasirinkus statines kameras ir panaudojus jų daugiau, galima taikyti šį pasirinktą metodą [27].



2.2 pav. Asmenų sekimas su aprėpties langais panaudojus keletą kamerų skirtingose vietose [27]

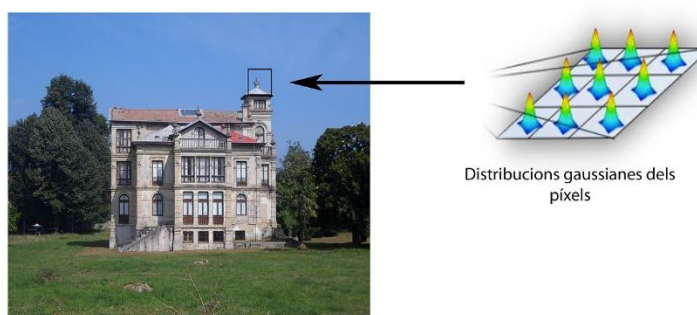
Pritaikius tinkamus algoritmus ir metodus objektų aptikimui ir sekimui, bei sukūrus stebėjimo sistemą iš daugelio kamerų, turime tobulą stebėjimo sistemą tačiau dar bus susiduriama su problemomis. Literatūros apžvalgoje paminėtos pagrindinės problemos, su kuriomis autoriai jau buvo susidūrę atliekant asmens sekimą. Tai yra kameros raiškos apribojimai, kurie gali įtakoti vaizdo kokybei, taip pat staigus objekto judesys, triukšmas ir trikdžiai vaizde, objekto išvaizdos (aprangos) pasikeitimas, aplinkos apšvietimas, nestabili objekto padėtis (objektas gali būti nusiųkęs nuo kameros ar būti skirtingoje pozijoje), okliuzija (susiduriama, kai yra sekamas objektas, kur yra daug žmonių) ir realaus laiko apdorojimas (reikalingos didelės greیتaveikos stebėjimo sistemos, kurios vaizdą apdorotų realiu laiku).

Pasirinktas objektų sekimo metodas fono substrakcija 2.3 pav. Šis metodas yra praktiškai yra išbandytas ir naudojamas. Panaudojus *MATLAB* programa, kuri sugeba matematiškai apskaičiuoti daug algoritmų ir sugeneruoti filtrus, kurie reikalingi vaizdo apdorojimui. Kaip pateiktame paveikslėlyje matyti keturi vaizdai. Pirmas yra originalus vaizdas su objektais, antras vaizdas yra tik statinis fonas be jokių objektų, trečias vaizdas yra panaudojus Grėjaus skalę sukuriamas juodai-baltas vaizdas, asmenys išskiriami iš fono ir fonas panaikinamas. Ketvirtas vaizdas taip pat išskiriami asmenys tačiau panaudojamas kitos spalvos fonas.



2.3 pav. Fono substrakcijos metodas [4]

Fono substrakcijos būdui naudojami Gauso mišinių algoritmai (*angl. Gaussian mixture methods*). Vienas iš pagrindinių metodų naudojamas Gauso skirstinys, kuris taikomas kiekvienam vaizdo pikseliui 2.4 pav. Taikant šį skirstinį yra tikimybiškai nustatomas kiekvieno pikselio intensyvumas.



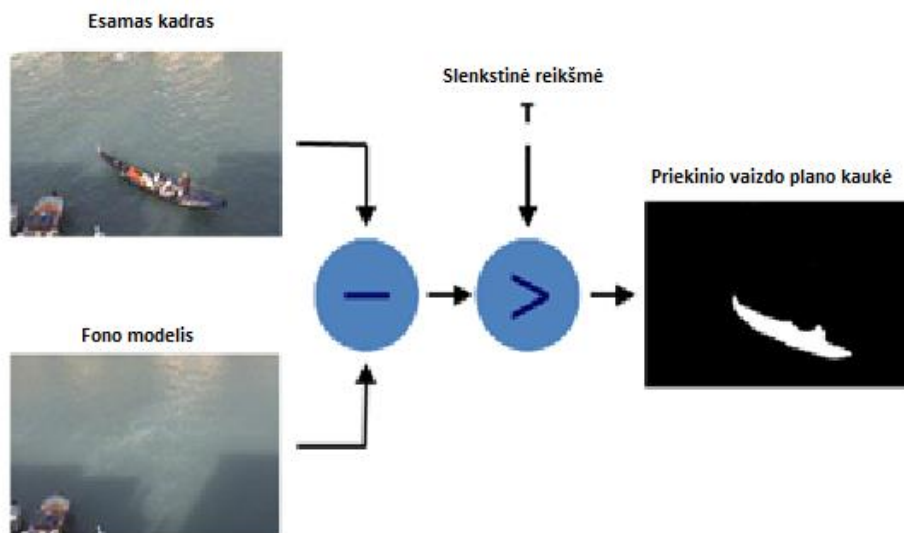
2.4 pav. Gauso skirstinys kiekvienam vaizdo pikseliui [28]

Fono substrakcijos metodas prasideda nuo dalinimo dalies, kai judantys objektai ar priekinio vaizdo planas (*foreground*) padalinamas į dalis (segmentuojamas) ir atskiriamas nuo fono. Tai yra paprasčiausias būdas paimti vaizdą kaip foną ir paimti kadrus kurie yra gauti tuo pačiu laiku t lyginti su foniniu vaizdu B . Panaudojant aritmetinius veiksmus, objektas padalinamas į dalis, panaudojant vaizdo atėmimo kompiuterinių vaizdų apdorojimo technologiją, kai kiekvienas vaizde

esančio elemento pikselis, yra priskiriamas $I(t)$ ir imama pikselio reikšmė $P[I(t)]$. Visa tai atimama su atitinkamais vaizdo elementais, kurie yra toje pačioje padėtyje fone, kuris yra $P[B]$.

$$P[F(t)] = P[I(t)] - P[B] \quad (2.1)$$

Pritaikant formulę (2.1) gaunamas rezultatas, kuris parodytas 2.5 pav.



2.5 pav. Vykdoma atėmimo operacija pritaikant slenkstinę reikšmę [29]

Svarbiausia yra tai, kad pirmiausia reikia užfiksuoti foninį vaizdą be objekto, o po to užfiksuoti esamą vaizdą su objektu ir atlikti atimtį pagal formulę 2.1, tačiau reikia užfiksuoti foninį vaizdą tam tikru laiko momentu, kad po to būtų pagal jį lyginamas pikselių intensyvumas. Tačiau yra ribinė vertė (angl. *threshold*), kurią reikia įvertinti, nes ši slenkstinė reikšmė suteikia tikslesnį fono atėmimą. Atliekant fono atėmimo operaciją yra svarbu gauti tikslų galutinį rezultatą – objekto vaizdą be fono. Tačiau reikia įvertinti pikselių intensyvumą tam tikroms vaizdo elemento vietoms, kurios galėjo kisti dviejuose kadruose. Tačiau yra didelė problema, kad šis metodas veiks tik tada, kai visi priekinio plano vaizdo elementai yra judantys, o fono vaizdo elementai turi būti statiški (nejudrūs). Tam ir naudojama ribinės reikšmės formulės:

$$|P[F(t)] - P[F(t+1)]| < \text{ribinė vertė} \quad (2.2)$$

Formulė (2.2) reiškia skirtumą tarp vaizdo pikselių intensyvumo pradinio laiko momentu t ir po to praėjusiam laiko momentui $t+1$, turi būti didesnis už ribinę reikšmę. Šio metodo vaizdo

tikslumas priklauso nuo objekto judėjimo greičio. Jei objektas juda greitai tada reikalinga didesnė ribinė reikšmė. Jei objektus yra statiškas (nejudrus), tada pikselių skirtumas neviršins slenkstinės reikšmės ir objektas nebus fiksuojamas dinaminės kaukės vaizde.

Siekiant tobulinti apsaugos sistemą, neužtenka aptikti ir sekti asmenis. Jau anksčiau buvo minėta, kad stebėjimo sistema aptinka paliktus ir daiktus ar kai asmuo išsitraukia ginklą. Šaltinyje [8] buvo eksperimentiškai atliktas asmens veiksmų sekimas, bei ginklo aptikimas. Pritaikant fono substrakciją įtartinų asmenų sekimui bei ginklų aptikimą, galima sukurti šiuolaikinę ir patikimą apsaugos sistemą viešosiose erdvėse.

2.2 Pasirintų metodų pritaikymas *MATLAB* programos aplinkoje

2.2.1 Objektų detekcija panaudojant aprėpties langus

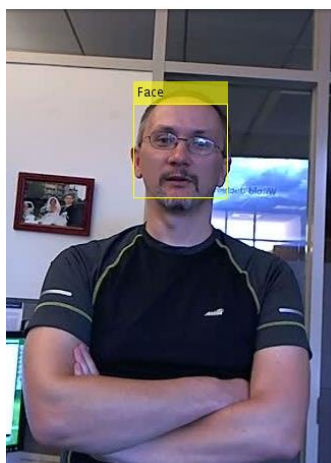
Viešosiose erdvėse yra labai svarbu sekti kiekvieną asmenį ir identifikuoti jį unikaliu kodu. Tačiau tai yra labai sudėtinga, nes operatorius stebėdamas vaizdą ekrane, negali pastebėti skirtingų asmenų, kai yra aplink daug žmonių. Tačiau tai yra išsprendžiama problema panaudojus veido detekciją (aptikimą) panaudojant aprėpties langą, kurio pagalba galima stebėti asmenis bei juos sekti.

Jau anksčiau buvo minėtą algoritmų ir būdų, kurie pritaikomi asmenų aptikimui ir sekimui. Tačiau visai tai siekiant įgyvendinti, reikia įsigyti stebėjimo (vaizdo apsaugos kameros, monitoriai) bei programinę įrangą. Tačiau tai galima sukurti nereikalaujant didelių investicijų. Paprasčiausias būdas tai yra pritaikyti kompiuterį su kameromis bei *MATLAB* programą, su kuria galima atlikti daug su vaizdo apdorojimu susijusių operacijų. Siekiant įgyvendinti šio darbo tikslą galima panaudoti *MATLAB* programą sukuriant aprėpties langus bei fono substrakciją.

2.2.2 Veido detekcija naudojant *MATLAB* programą

Veido detekciją (aptikimą) atliekant su *MATLAB* programa yra susiduriama su problema. Tinkamo algoritmo parinkimas veido detekcijai. Tačiau programinis *MATLAB* paketas, turi daug integruotų funkcijų ir algoritmų, kuriuos galima panaudoti vaizdo apdorojime. Tai palengvina visą programavimą ir sumažina programos apimtį. Tačiau siekiant, kad ši programa būtų panaudota vaizdo apdorojimo pritaikyme, reikia ne tik taikyti tinkamus algoritmus, bet ir sukurti sąsają su išorinėmis vaizdo kameromis, kurios bus naudojamos veido detekcijai (2.6 pav.). Veido detekcija vykdoma su aprėpties langu. Apie šį objektų aptikimo metodą buvo kalbėta

anksčiau. Tačiau aprėpties langus galima naudoti ne tik veido detekcijai, bet aptikti visą judantį objektą.



2.6 pav. Veido detekcija panaudojus *MATLAB* programą [30]

Panaudojus aprėpties langą, suteikia galimybę ne tik detektuoti asmenis, bet ir išskirti asmenų savybes, identifikuoti ir sekti.

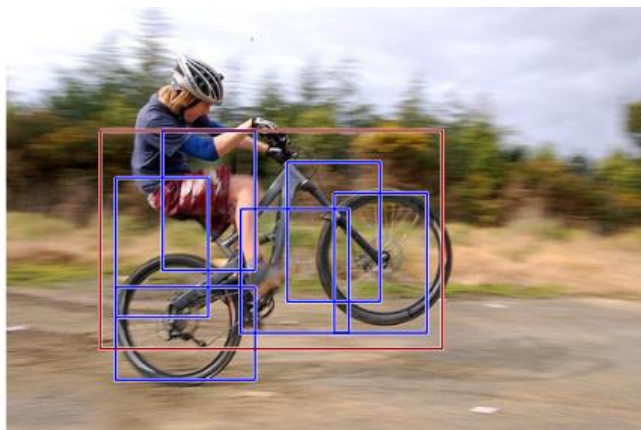
2.2.3 Detektuojantis objektus aprėpties langas

Objektų detekavimas su aprėpties langu yra patikimas būdas aptikti, sekti ir identifikuoti asmenis. Kadre pasirodęs asmuo yra detektuojamas aprėpties langu pagal jo esamus matmenis. Šie matmenys gali kisti, kai objektas artėja arba tolsta nuo vaizdo kameros. Taip ir turi būti su aprėpties langu, kurio matmenys turi kisti pagal asmens matmenis. Panaudojant *MATLAB* programą, kurioje yra funkcija su aprėpties langu, visa tai galima atlikti.

Aprėpties lango veikimas vyksta pagal pikselių pokyčius. Iš vaizdo imami kadrai ir atliekamas jų skirtumas. Programa yra apmokyta taip, kad atlikus kadrų skirtumas, per visą kadro langą slankioja klasifikatorius, kuris pažymi pikselių pokytį. Jei tarp gretimų kadrų įvyko pikselių pokytis yra nubrėžiamas stačiakampis langas aplink judančius objektus. Detektuojant asmenis, didžiausias pikselių pokytis pasireiškia ten kur asmuo juda. Todėl, bet koks objekto judesys yra fiksuojamas aprėpties langu.

2.2.3 Teigiami ir neigiami atvaizdai

Programa *MATLAB* naudoja apmokytą vaizdo žymeklį, kuris naudojamas teigiamiems ir neigiamiems atvaizdams sukurti. Jis pažymi teigiamo atvaizdo svarbų vaizdą, kuris yra apibrėžiamas stačiakampiais aprėpties langais (2.7 pav.).



2.7 pav. Dviratininkas su dviračiu detektuojamas stačiakampiais aprėpties langais [20]

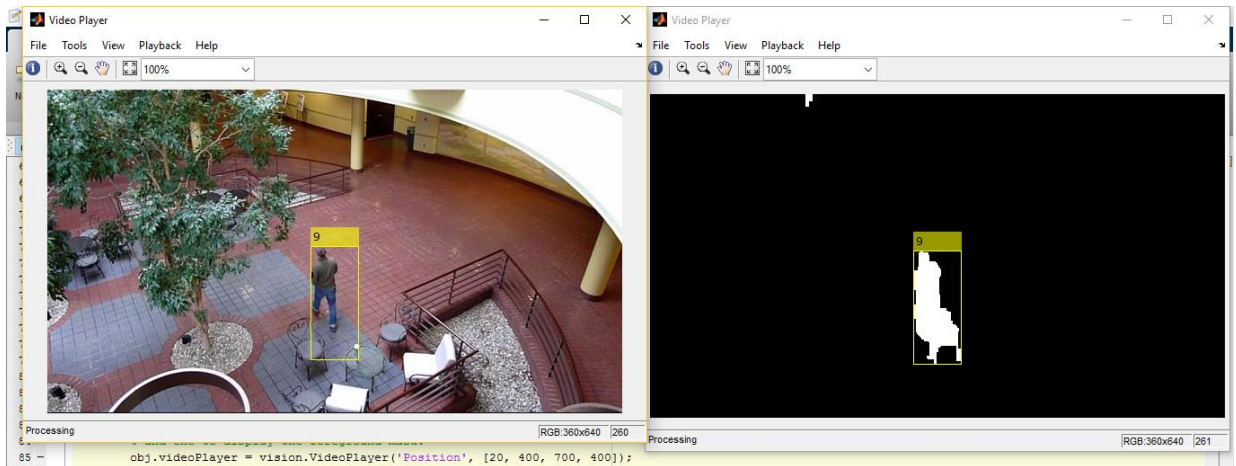
Tai pat galima rankiniu būdu apibrėžti teigiamus atvaizdus dviem būdais. Pirmas būdas yra apibrėžti stačiakampio sritis didesniame atvaizde, tačiau sritys turi apibrėžti tik svarbius (dominančius) objektus, kuriuos būtina detektuoti. Antras būdas iškirpti detektuojantį objektą ir išsaugoti kaip atskirą atvaizdą. Tada vėl galima apibrėžti sritį, kurioje yra pilnas atvaizdas, kaip buvo paminėta pirmame variante.

Neigiami atvaizdai nėra tiksliai apibrėžti. *MATLAB* funkcija *BoundingBox* savaime sukuria neigiamus atvaizdus iš vartotojo gaunamų neigiamų atvaizdų, kuriuose nėra svarbūs esantys objektai atvaizde, tokie kaip žmonės ir automobiliai. Prieš apmokant kiekvieną naują etapą, funkcija valdo detektorius, kuris jau buvo ankstesniuose etapuose apmokytas atpažinti neigiamus atvaizdus. Tačiau jei objektai kurie buvo aptikti anksčiau, yra neteisingai priskirti kaip teigiami, kuriuos detektorius fiksuoja kaip neigiamus atvaizdus, tada šiuo atveju kiekviename naujame etape detektorius yra apmokytas ištaisyti klaidas, padarytas ankstesniuose etapuose.

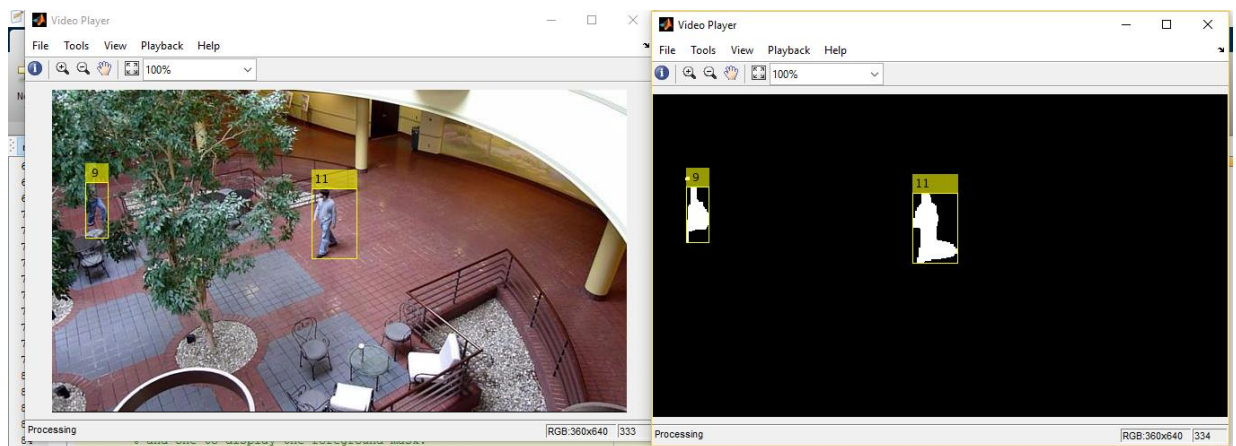
2.2.4 Keleto objektų sekimas panaudojant aprėpties langus ir fono substrakciją

Apie apriepies langus ir fono substrakciją buvo minėta jau anksčiau. Tačiau šių metodų realizavimas panaudojus *MATLAB* programoje yra įmanomas ir jau programinių pavyzdžių. Vienas yra tokių programos šablonų *MultiObjectTracking*, kuriame jau yra sukurtas vaizdo įrašas

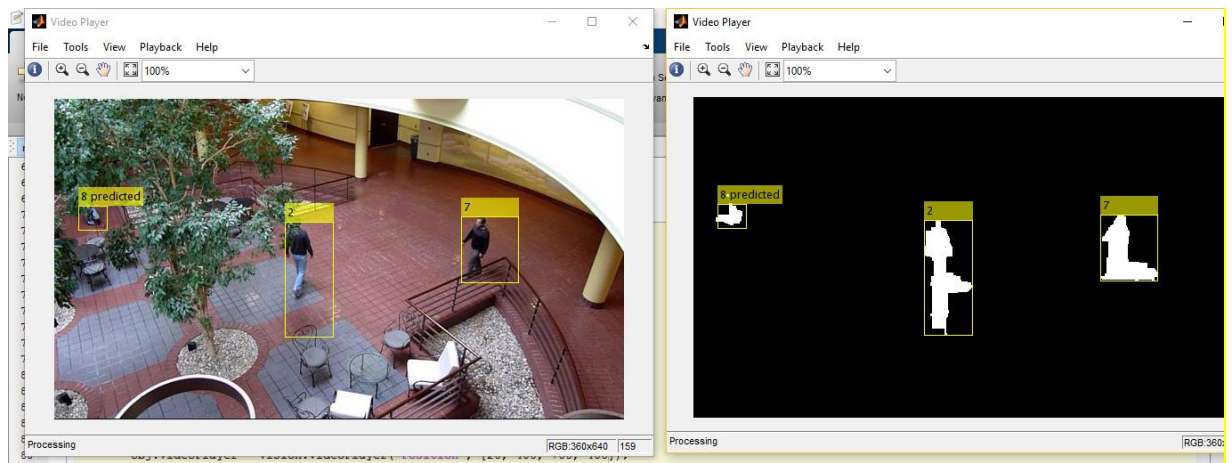
su keliais asmenimis. Šiame vaizdo yra naudojama dinaminio vaizdo kaukė (fondo subtrakcija), ir objektų identifikavimas bei daug funkcijų, kurias reikia apjungti. Apjungius šias funkcijas *setupSystemObjects*, *initializeTracks*, *detectObjects*, *predictNewLocationsOfTracks*, *updateAssignedTracks*, *deleteLostTracks*, *createNewTracks*, *displayTrackingResults*, gaunamas galutinis rezultatas asmenų sekimas, identifikavimas tam tikrais kodais, asmens detektavimas panaudojus stačiakampį aprėpties langą, fondo subtrakcija pagal žmogaus siluetą, bei asmens aptikimą kai jis yra už kliūtis (medžio ar kito objekto) panaudojus vaizdo įrašą 2.8 -2.11.



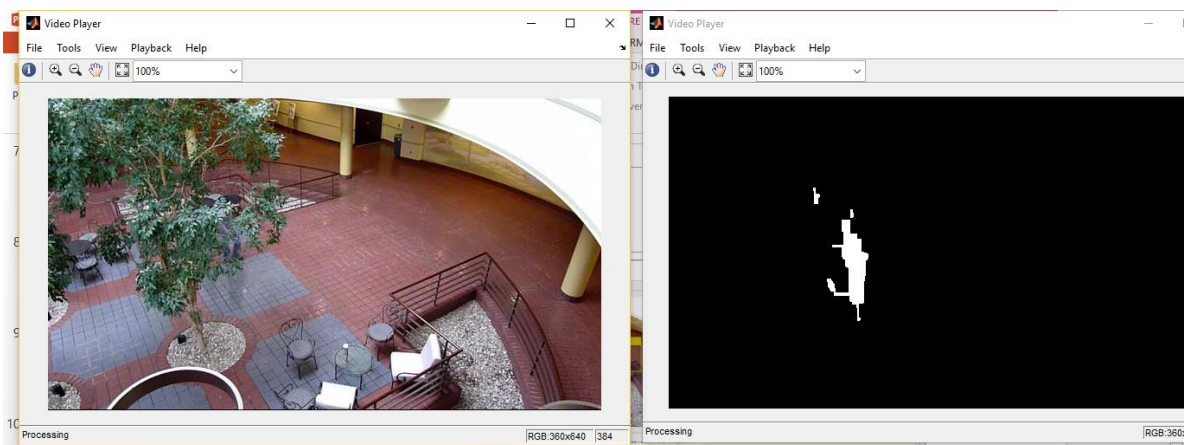
2.8 pav. Vieno asmens sekimas su identifikavimo kodu [31]



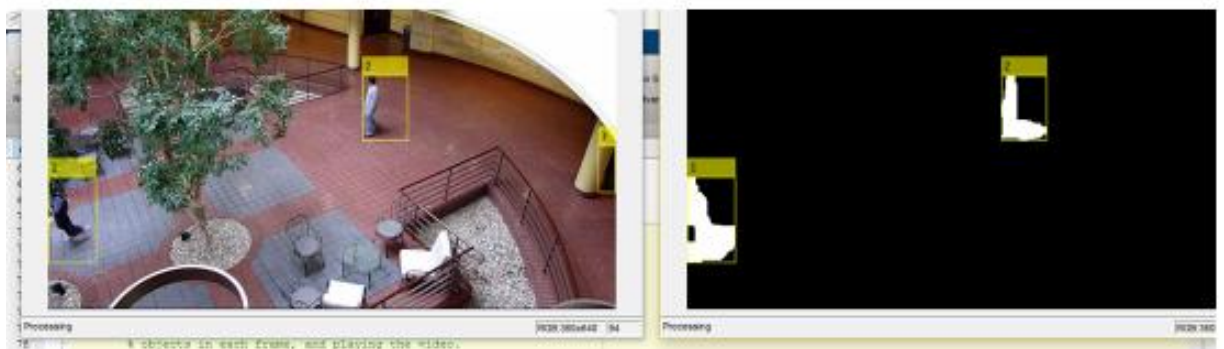
2.9 pav. Dviejų asmenų sekimas ir identifikavimas skirtingais kodais [31]



2.10 pav. Trijų asmenų sekimas su identifikavimo kodais [31]



2.11 pav. Asmens siluetas, kai asmuo yra už kliūtis [31]



2.12 pav. Asmenų sekimas (asmuo 2 identifikuojamas kitu kodu) [31]

Atlikus bandymą su programos šablonu *MultiObjectTracking*, gauti rezultatai tokie: kiekvienas asmuo sekamas nenutrūkstamai panaudojant aprėpties langą su identifikavimo kodu

- originalus vaizdas ir vaizdas su dinamine vaizdo kauke arba atlikta fono substrakcija;

- objektai abiejuose vaizduose yra detektuojami ir sekami, asmuo kuris yra netiksliai aptinkamas identifikuojamas kaip prognozuojamas (angl. *predicted*).

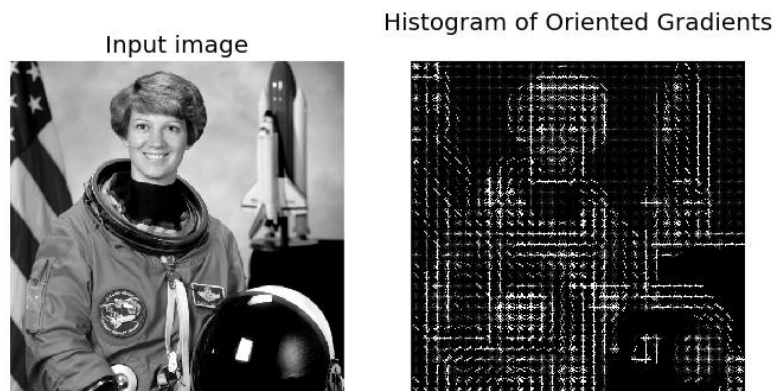
Panaudojus du *MATLAB* programos langus, kuriuose vaizduojama *RGB* pikselių formatu vaizdas ir fono subtrakcija (dinaminė vaizdo kaukė) galima nesunkiai stebėti ir sekti asmenis, kai abiejuose languose asmuo žymimas stačiakampiu langu.

Buvo atlikus eksperimentą pastebėta ne tik privalumų, bet ir trūkumų, kuriuos reikia tobulinti. Tai yra šie trūkumai:

- Realaus laiko apdorojimas
- Toks pat asmens identifikavimo kodas turi išlikti visada, kai asmuo palieka stebėjimo zoną
- Kelių kamerų susiejimas, kad galima stebėti ir sekti asmenis keliose vietose
- *MATLAB* ir vaizdo kamerų sąsaja (tačiau galima atlikti apdorojimą su vaizdo įrašu)
- Asmenų okliuzija (persidengimas)
- Šešėlių įtaka
- Netiksli objekto forma

3.1 Savybių išskyrimo tipai

Programa *MATLAB* palaiko įvairius savybių išskyrimo tipus: vietiniai dvejetainiai modeliai - *LPB*, orientuotų gradientų histogramos - *HOG*. *LBP* ypatybės dažnai naudojamos, veidų aptikimui todėl, nes šis būdas puikiai atvaizduoja aukštos raiškos atvaizdus. *HOG* ypatybės yra dažnai naudojamos tam, kad aptiktų pilnus objektus tokius kaip žmones ar automobilius. Jie tiksliai fiksuoja bendrą objekto formą. *HOG* vizualizacijoje įmanoma pamatyti moters kontūrus 2.13 pav.



2.13 pav. Orientuotų gradientų histogramos *HOG* pavyzdys [25]

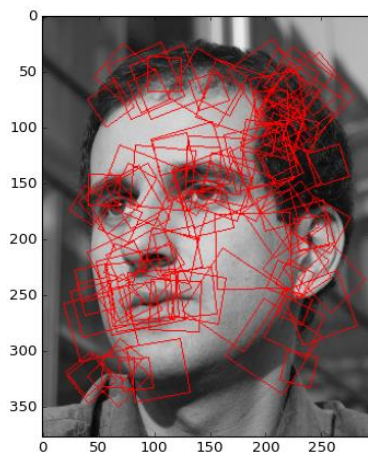


2.14 pav. Vietinio dvejetainio modelio *LPB* pavyzdys [32]

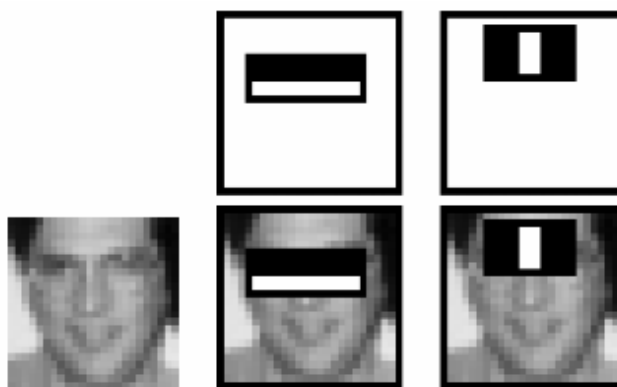
2.9 SVM klasifikatorius

Anksčiau jau buvo minima apie asmenų savybių palyginimą. Stebint ir sekant asmenis yra svarbu juos identifikuoti. Todėl čia yra susiduriama su problema – asmenų atpažinimas. Asmuo kadre gali pasirodyti ne vieną kartą, o daugiau kartų. Todėl reikia juos atpažinti ir suteikti jiems identifikavimo kodą. Suteikti skirtingus identifikavimo kodus kiekvienam asmeniui nėra sudėtinga, tačiau suteikti tokį patį identifikavimo kodą tokiam pačiam asmeniui yra sudėtinga. Tada reikia remtis išvaizdos modeliu ir nustatyti, kad tai yra toks pat asmuo, kuris buvo kadre pasirodęs anksčiau.

Literatūroje ir moksliniuose straipsniuose yra aprašyta įvairiausių išvaizdos modelių, tačiau ne visi pritaikomi praktikoje. Jau paminėti tokie kaip *HOG* ir *LPB*, kurie yra praktiškai taikomi, taip pat yra jau integruoti *MATLAB* programiniame pakete, kurie gali būti naudojami įvairiam vaizdo apdorojimui. Taip pat yra *SURF*, *HAAR*, *HSV* spalvų histogramos, *colorMoments*, *gaborWavelet*, *waveletTrasnform* [36], [37], [38]. Šie yra pagrindiniai *MATLAB* programos savybių išskyrimo metodai. Apie *HOG* ir *LBP* savybes jau buvo kalbėta anksčiau. *SURF* savybių išskyrimo metodas pateiktas 2.15 pav.

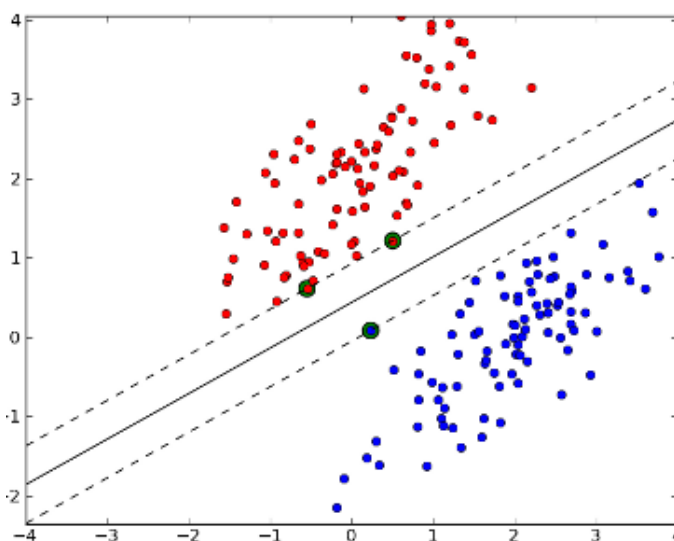


2.15 pav. *SURF* savybių išskyrimo metodas [33]



2.16 pav. HAAR savybių išskyrimo metodas [34]

Asmenų savybių nustatymas yra labai svarbu, kai reikia juos atpažinti. Tačiau vien to neužtenka. Reikia asmenų savybes palyginti tarpusavyje ir nustatyti ar tos savybės yra vienodos ar skirtingos. Visą savybių palyginimą atlieka *SVM* klasifikatorius. Tačiau, kad *SVM* klasifikatorius pradėtų lyginti savybes, jam reikia pradinių duomenų. Padavus savybių duomenis klasifikatoriui šis juos lygina tarpusavyje ir priskiria tam tikrai savybių grupei 2.17 pav.



2.17 pav. *SVM* klasifikatorius [35]

Iš pateikto paveikslėlio matyti dvi duomenų grupės ir atraminiai mašininiai vektoriai. Čia pateiktas *SVM* linijinis klasifikatorius. Duomenų grupės yra pavaizduotos mėlynais ir raudonais taškeliais. Jos yra lyginamos pagal atraminius vektorius priešingose pusėse. Pagrindinis vektorius šias grupes atskiria bei sudaro atraminių vektorių nuokrypį nuo pagrindinio vektoriaus. Kaip buvo minėta anksčiau *SVM* klasifikatorius atlieka duomenų grupių palyginimą. Tačiau, kad *SVM* pradėtų palyginimą jam reikia mažiausiai dviejų pradinių grupių, kurios pagal pateiktą paveikslėlį

yra sugrupuojamos pagal atraminius vektorius. Atsiradus naujai grupei atliekas lyginimas su dviem esamom grupėm, kurios buvo pradinės.

SVM klasifikatorių pritaikant *HOG* ir *LBP* savybių nustatymo metodams, galima atlikti savybių grupavimą, bei jas tarpusavyje lyginti. Taip galima identifikuoti ir atpažinti esamus ir naujus asmenis. Tačiau kiekvieną kartą aptikus asmenis, reikia jų išvaizdos savybes kaupti duomenų bazėje, kad po to būtų galima pateikti duomenis *SVM* klasifikatoriui. Pasirinkus vieną iš savybių nustatymo metodų *HOG* ar *LBP*, galima nustatyti, kuris metodas tiksliau identifikuoja asmenis.

3. EKSPERIMENTINĖ DALIS

Šioje dalyje su programiniu paketu *MATLAB* bus tiriamas ir atliekamas eksperimentas su asmenų sekimu, panaudojus aptikimo, sekimo ir savybių išskyrimo metodus. Kuriant vaizdo apdorojimo sistemą reikės taip pat ir įrangos, kuri bus naudojama šiam tyrimui bei asmenų, kurie bus stebiami per vaizdo kamerą. Tokios kuriamos sistemos tikslas yra greitaveika ir kokybė. Jei greitaveika bus maža, tokią vaizdo apdorojimo sistemą negalėsime priskirti realaus laiko sistemai, tačiau kitu atveju išauga sistemos resursai ir tada reikia naudoti brangią sistemą. Kokybė tai pat yra svarbus kriterijus šiai sistemai. Vaizdo raiška ir asmens aptikimas tai yra svarbiausi kokybės kriterijai. Tačiau jei vaizdas yra aukštos raiškos tada reikia, kad vaizdo sistema būtų didelės greitaveikos. Todėl reikia pasirinkti optimalų variantą, aukšta vaizdo raiška bus mažai atsižvelgiama, tačiau tikslus asmens aptikimas ir sekimas yra šio darbo tikslas.

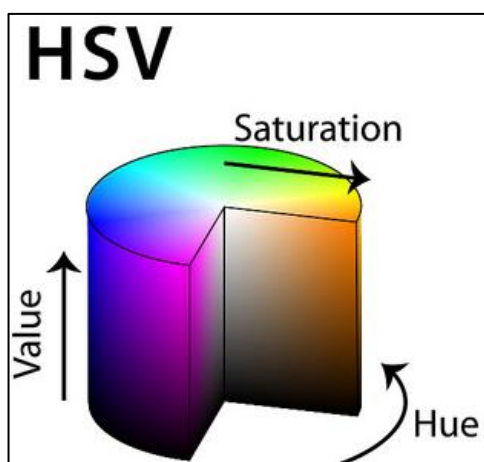
3.1 Eksperimentiniai rezultatai

Atliekamas pirmas eksperimentas, kurio pagrindinis tikslas yra iširti ir apibendrinti asmenų atpažinimo tikslumą. Naudojantis *MATLAB* programiniu paketu, yra sukuriama programa, kuri sugebės, priimti vaizdą iš vaizdo kameros (šiam eksperimentui bus naudojamas jau sukurtas vaizdo įrašas), detektuoti asmenis, atskirti juos pagal savybes ir sekti nuolatos. Visa tai reikalauja ne vienos, o daug funkcijų, kurias reikia apsašyti bei panaudoti funkcijas, kurios yra integruotos *MATLAB* programiniame pakete. *MultiObjectTracking* – *MATLAB* programos šablonas, kuris jau yra naudojamas kelių objektų sekimui. Panaudojus šį šabloną bei įterpus papildomų funkcijų ir paprogramių bus sukuriama programa, kurios tikslas bus asmenų detektavimas, identifikavimas unikaliais kodais, savybių išskyrimas, atpažinimas ir sekimas.

3.1.1 Paprogramių (funkcijų) ir pagrindinės programos aprašymas

Anksčiau jau buvo minėta, kad bus naudojamas jau sukurtas programos šablonas objektų atpažinimui ir sekimui, kuris tinka šio darbo tikslui. Tačiau vien tik jo neužtenka panaudoti, jis tik bus naudojamas kaip pagrindinė programa. Todėl reikia naudoti papildomų programų (paprogramių), kurios apjungs visą bendrą programą. Pagrindinėje programoje *MultiObjectTracking* yra svarbiausia nustatyti pradinis parametrus bei sukurti duomenų bazę. Tam, kad objektai būtų aptikti gautame vaizde, turi būti nustatytos pradinės (slenkstinės) reikšmės, kurias viršijus, objektas bus fiksuojamas kadre. Nes jei nebus nustatytos slenkstinės reikšmės, tada gali būti netikslus objektų identifikavimas ir sekimas (kadre gali būti sekami visi judantys

objektai), kuriuos sekti nėra tikslinga. Nustatomos minimalios slenkstinės reikšmės, kurios programoje apsirąšomos kaip *min_person_width* ir *min_person_height*. Tai yra minimalūs objekto matmetys, asmens plotis ir aukštis, nuo kurių jis yra aptinkamas ir sekamas. Taip pat svarbu pagrindinėje programoje sukurti duomenų masyvus, kuriuose bus kaupiami duomenys kiekvieno sekamo objekto (identifikavimo ir savybių) *duomenysTemp*, *duomenysIDTemp*, *duomenys*, *duomenysID*, tam kad galima juos panaudoti tolimesniuose apdorojimo etapuose. Reikia paminėti tai, kad yra sukuriami laikini ir pastovūs duomenų masyvai. Šioje paprogramėje atliekama *RGB* konvertacija į *HSV* spalvų skalę. Ši konvertacija yra reikalinga tam, kad būtų panaikinti šešėliai dinaminio vaizdo kaukėje. Nes jei nebus pašalinti šešėliai, jie taip yra fiksuojami kaip judantis objektai ir sekami kaip ir asmuo. *HSV* spalvų skalę sudaro trys dedamosios *saturation* – įsisotinimas, *value* – reikšmė ir *hue* – atspalvis 3.1 pav.



3.1 pav. *HSV* spalvų skalės dedamosios: *saturation* – įsisotinimas, *hue* – atspalvis ir *value* – reikšmė

Panaikinus *value* dedamąją, pradingsta šešėliai, nes *value* dedamoji nustato dydį atspalvio, sotis ir atspalvis programoje yra susumuojami:

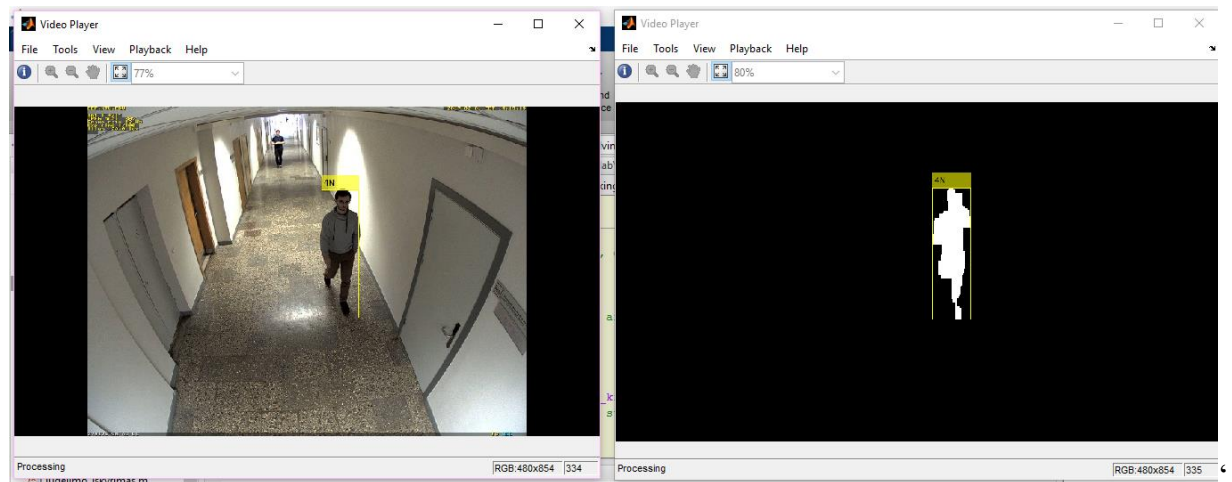
$$HS = H + S \quad (3.1)$$

Taip pat normalizuojama suma, kad reikšmės būtų iki vieneto:

$$HS_{norm} = \frac{HS}{\max(HS)} \quad (3.2)$$

Paprogramėje, kurioje sukuriami sisteminiai objektai, naudojama funkcija *setupSystemObjects*. Tai reiškia, kad šioje paprogramėje vyksta gaunamo vaizdo apdorojimas iš vaizdo kameros (realaus laiko vaizdo sistema) ar vaizdo įrašo. Funkcija *vision.VideoFileReader* nuskaito pasirinktą vaizdo šaltinį. Kad vaizdas po viso įvykdyto apdorojimo būtų matomas

MATLAB programoje reikia panaudoti integruotą papildomą funkciją *vision.VideoPlayer*, kuris atlieka kaip vaizdo leistuvo funkciją ir gautas vaizdas po apdorojimo yra matomas *MATLAB* aplinkoje. Tačiau svarbiausia yra nustatyti vaizdo dydį pikseliais, nes kaip anksčiau buvo minėta, kad tai lemia atkuriamo vaizdo greitį. Todėl reikia pasirinkti optimaliausią vaizdo dydį, tam kad kokybė ir greitaveika būtų priimtimiausia. Taip pat jau daug kartų minėta fono substrakcija, kurią šiuo atveju irgi buvo panaudota asmenų sekimui. Pasinaudojus anksčiau minėta *MATLAB* funkcija *vision.VideoPlayer* sukuriamas papildomas vaizdo leistuvo langas su atlikta fono substrakcija 3.1 pav.



a)

b)

3.2 pav. *MATLAB* aplinkoje sukurti du vaizdo leistuvai: a - originalus vaizdas, b - fono substrakcija

Tačiau to nepakanka, kad būtų atvaizduotų fono substrakcija, todėl tam yra naudojamos papildomos funkcijos kaip *vision.ForegroundDetector* ir *vision.Blobanalysis*, kurios naudojamos sukurti dinaminio vaizdo kaukę, tam kad būtų sekami tik asmenų siluetai. Tačiau dar reikia paminėti, kad dinaminio vaizdo kaukė yra atvaizduojama ne *RGB* vaizdo skalėje, o Grėjaus (juodai balta) skalėje. Ši skalė jau yra aprašyta anksčiau paminėtose programose, kur naudojamas jau anksčiau minėtas Gauso mišinių modelis, kuris *MATLAB* programoje apsirašo funkcija *vision.ForegroundDetector*. Ši funkcija palygina judančius pikselius su fonu, taip sukurdamas dinaminio vaizdo kaukę bei fono modelį. Tačiau yra svarbu nustatyti parametrus, kurie įtakoja asmens silueto sekimą. Svarbiausi parametrai yra apsimokymų skaičius *NumTrainingFrames*, kuris nustato po kiek kadrų bus apsimokoma, gaunamų iš vaizdo, tačiau šiuo atveju yra naudojamas standartinis skaičius kuris nustatytas programoje. Taip pat yra svarbu fono slenkstinis skaičius (angl. *MinimumBackgroundRatio*), kuris nustato fono modelio slenkstį, iki kiek tikimybiškai

pikseliai gali būti neužfiksuojami. Šis slenkstinis skaičius yra taip pat nustatytas standartiškai pagal programą.

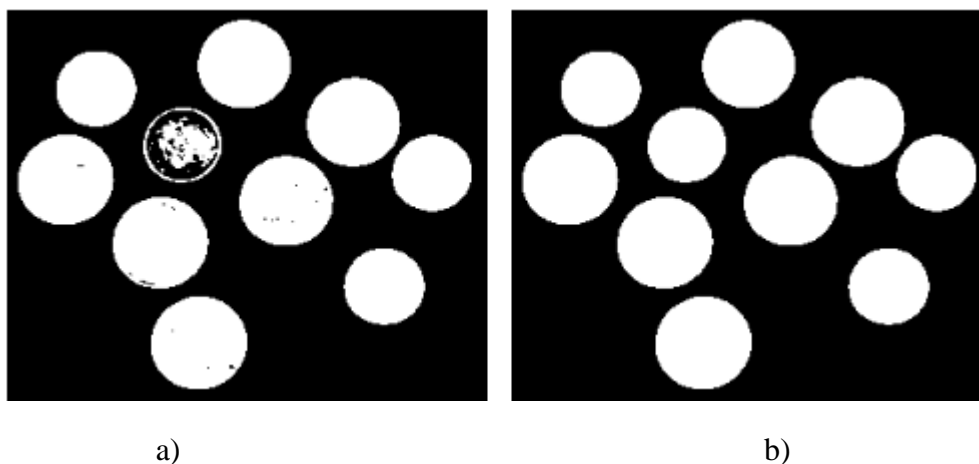
Atkuriant fono substrakciją yra svarbiausia, kad funkcija kuo tiksliau atkurtų asmens siluetą bei sukurtų kuo tikslesnį stačiakampį asmens sekimui, tam yra naudojama funkcija *BlobAnalyser*. Ši funkcija gautą kadrą fiksuoja į dvejetainį kadrą ir viskas yra laikoma tik bitų dydžiais. Kur yra aprėpties langas (angl. *Bounding box*), yra fiksuojamas asmuo ir bitai igauna teigiamą reikšmę *true*, ir tada pikseliai nubrėžiai baltą sekamo asmens siluetą, todėl svarbu šios funkcijos nustatyti parametrus tokius kaip *BoundingBoxOutputPort*, *AreaOutputPort*, į teisingą reikšmę - *true*. Tai pat svarbu šioje funkcijoje nustatyti minimalią reikšmę *Minimum BlobArea*, nuo kurios fiksuojamas dinaminės vaizdo kaukės sekamas asmuo. Šią reikšmę galima keisti pagal poreikį.

Inicializuojant sekamus asmenis (angl. *Initalize tracks*) yra svarbu nustatyti jiems identifikavimo kodus, aprėpties langus taip pat inicializuoti Kalmano filtrą, kuris prognozuoja asmenų judėjimą, kadrų skaičių (angl. *age*), kuris skaičiuoja sekamo asmens kadrus nuo pat pradžios, nuo tada kai jis užfiksuojamas kadre. Taip pat svarbu objekto būseną ar jis yra tame vaizde prieš tai pasirodęs: A- atpažintas, N - naujas ir I - identifikuotas, bei savybės *featureSave* ir *featureCount*, kurios saugomos duomenų bazėje. Paprogramėje, kur inicializuojami sekami asmenys yra sugeneruojami visi šie masyvai, kuriuose bus kaupiami ir saugomi duomenys. Programiškai yra sudaroma bendra struktūra (angl. *struct*), kurioje yra visi šie išvardinti sekamo asmens parametrai.

Parogramėje nuoseklus kadrų nuskaitymas (angl. *read next frame*) vykdoma paprasčiausia operacija. Naudojama funkcija *readFrame*. Gautas vaizdas yra suskaidomas į kadrus, tačiau svarbu yra nustatyti kadrų nuskaitymo žingsnį, kuris šiuo atveju bendroje programoje yra nustatytas automatiškai. Kadrų nuskaitymas vyksta nuosekliai ir kadrai yra skaičiuojami pagrindinėje programoje bei kadrų skaičius patalpinamas į masyvą. Pagal kiekvieną kadro numerį nustatomos objekto savybės, būsenos ir sekimas.

Objektų aptikimui naudojama funkcija *detectObjectsframe*, tai reiškia kad objektai detektuojami kiekviename kadre. Morfologinė operacija atliekama su dinamine vaizdo kauke. Šios abi operacijos yra aprašytos bendroje paprogramėje. Čia yra naudojama dinaminio vaizdo kaukė *mask* kaip anksčiau buvo minėta dvejetainiam kadrai, kuris naudojamas fono substrakcijai. Morfologinė operacija pašalina papildomius trikdžius aptiktiems objektas bei užpildo kiaurymes, kurios atsiranda dinaminio vaizdo kaukėje, tam kad objekto siluetas būtų tikslios formos ir tolygus

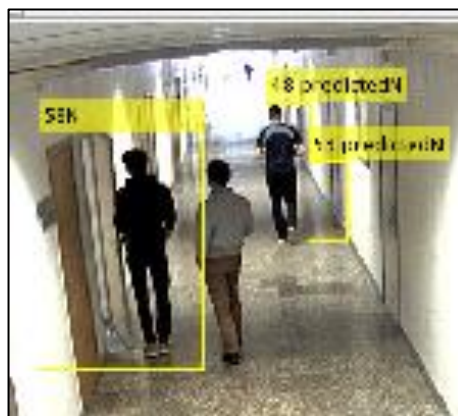
3.3 pav.



3.3 pav. a – originalus vaizdas, b – vaizdas po morfologinės operacijos

Operacijos *imopen* ir *imclose* naudojamos tam, kad atsiradusius dvejetainio vaizdo netikslumus ir trikdžius, pašalintų, panaudojant stačiakampį ir kvadratą, kurie nuosekliai slenka per visą aprėpties langą. Šios operacijos yra nuoseklios tai reiškia, kad iš pradžių vykdoma morfologinė atidarymo operaciją ir po to uždarymo. Kaip pavaizduota 3.2 pav. naudojama operacija, kuri užpildo visas aprėpties lange susidariusias tuštumas.

Egzistuojančių kadre asmenų prognozuojamas (nuspėjamas) sekimas (angl. *Predict New Locations of Existing Tracks*) atliekamas panaudojant Kalmano filtrą. Šioje paprogramėje naudojama funkcija *predictNewLocationsOfTracks*, kurioje yra sukuriamas ciklas tam, kad būtų kiekvienas kadre vis atsirandantis naujas objektas sekamas su prognozuojamu sekimu. Nuspėjamas sekimas ypač svarbus yra tada, kai objektas yra uždengtas, už kliūtis ar ir išeinantis iš stebėjimo zonos, kai jo matmetys mažėja. Tada ši funkcija dar kurį laiką objektą fiksuoja kaip nuspėjamą (angl. *predicted*) su identifikavimo kodu 3.3 pav.



3.4 pav. Prognozuojamas objekto sekimas

Iš pradžių cikle yra surandamas sekamo objekto dabartinis (esamas) centras (angl. *current centroid*) ir panaudojamas Kalmano filtras. Taip pat apsirąšomas nuspėjamas objekto centras kartu su Kalmano filtru ir aprėpties langu, kuris seka objektą pagal nuspėjamą jo centrą (angl. *predicted centroid*). Kai objektas yra prastai matomas ar užsidengtas, tada suveikia Kalmano filtras ir aprėpties langas seka tik tam tikrą laiką objektą su identifikavimo kodu bei užrašu *predicted*.

Paprogramėje, kurioje yra priskyrimai sekamiems objektams (angl. *Assign Detections to Tracks*) sudaro daugelis funkcijų tokių kaip *assignments*, *unassignedTracks*, *unassignedDetections*, *detectionToTrackAssignment*. Šios funkcijos naudojamos, tam kad visus sekamus objektus patikrintų. Ir tuo atveju, kai objektas naujas ir kai jau buvo identifikuotas bei atpažintas. Tai reiškia sekančiame kadre objektas yra toks pat kaip buvo ankstesniame kadre užfiksuotas. Tai veikia jei objektas buvo mažai pajudėjęs iš vietos arba stovėjo vietoje. Todėl yra slenkstinė reikšmė *costOfNonAssignment*, kuri nustato objekto sutapimą. Šios funkcijos nėra svarbios šioje programoje tačiau, patobulina asmenų aptikimą ir sekimą.

Priskirtų sekamų asmenų atnaujinimas (angl. *Update Assigned Tracks*) yra svarbus tuo, kad antrą kartą aptiktus asmenis kadre reikia atnaujinti jų parametrus. Jeigu jie prieš tai buvo dingę iš kadro ar uždengti kliūtis ir jie vėl pasirodė, reikia atnaujinti jų identifikavimo kodą, objekto centrą, aprėpties langą. Funkcija *updateAssignedTracks* atlieka visą sekamų asmenų atnaujinimą. Jau anksčiau buvo paminėtas Kalmano filtras, kuris padeda teisingai nustatyti prognozuojamo, sekamo objekto centrą, taip pat aprėpties langas pasikeičia (angl. *replace*) ant atnaujinto objekto, bei nustato iš naujo asmens kadrų skaičių (angl. *age*), kuris parodo kiek laiko asmuo yra vaizde. Tai pat svarbu atnaujinimo matomumas (angl. *visibility*), kuriame asmens stebėjimo skaičius yra inrementuojams vienetu ir pradinė reikšmė sugrįžta į nulį tam, kad atnaujinimas būtų atliktas. Šis atnaujinimas yra svarbus, kai asmuo iš stebėjimo zonos buvo išėjęs ir vėl sugrįžo arba tada, kai pasirodo naujas asmuo stebėjimo zonoje. Visas šis atnaujinimas remiasi kiekvieno asmens identifikavimo kodu, kuris šioje funkcijoje naudojamas. Čia visada sukasi ciklas.

Priešinga funkcijai priskirtų sekamų asmenų atnaujinimui yra nepriskirtų sekamų asmenų atnaujinimas (angl. *Update Unassigned Tracks*). Naudojama funkcija *updateUnassignedTracks*, kurioje taip pat sukasi ciklas, kuriame sunkiai pastebimi objektai yra padaromi kaip nematomi ir jų kadrų skaičių bei pasirodymo laiką (*age*) inkrementuoja vienetu tam, kad jie būtų visada nefiksuojami. Ši funkcija reikalinga tada, kai objektai yra nutolę ir sunkiai pastebimi.

Ištrynimasis buvusių sekamų objektų (angl. *Delete Lost tracks*) reikalingas tuo, kad išėjusių objektų savybių duomenys, būtų panaikinami ir ne kaupiami, nes tai sukelia papildomus trikdžius, kurie paveikia gaunamą vaizdą. Funkcija *deleteLostTracks* visai tai atlieka: ištrina masyvą, kuriame yra sukaupti sekimo duomenys ir savybes. Tačiau šioje funkcijoje yra svarbu nustatyti

kadrų skaičiaus slenkstinę reikšmę, kurią pasiekus asmens duomenys ištrinami. Yra sudaromos dvi sąlygos, kai reikia ištrinti buvusių asmenis. Jei kadrų skaičius viršija slenkstinę reikšmę *ageThreshold* bei matomumo reikšmę *visibility*, tada objektas ištrinamas.

Naujų sekamų asmenų sekimas (angl. *Create New Tracks*) yra panašiai kaip inicializavimas, tačiau, jau šioje paprogramėje yra funkcija *createNewTracks*, kuri jau sugeneruoja naujus identifikuotus ir sekamus objektus pagal užsiduotus parametus. Jau yra minėta anksčiau, kad yra ciklas, kuris nuolatos sukuria naujam sekamam objektui savybes, kurios buvo paminėtos inicializavime *initializeTracks*. Ciklas vyksta iki objekto centro sekimo pabaigos. Kai objektas yra ištrinamas tada ir dingsta objekto centro koordinatės ir ciklas nustoja veikti. Šioje funkcijoje taip pat yra apsirąšomas Kalmano filtras, tam kad objektas būtų nuspėjamos sekamo objekto centro koordinatės. Anksčiau buvo minėta, kad aprėpties stačiakampis seka objektą pagal centrą ir inkrementuoja objekto identifikavimo kodą vienetu. Taip pat yra sukuriamas masyvas saugoti sekamų asmenų duomenis.

Paprogramėje (angl. *Display Tracking results*) išvedami sekimo rezultatai į ekraną. Čia naudojama funkcija *DisplayTrackingResults*. Iš pradžių kadrai ir dinaminė kaukės vaizdai verčiami į 8 bitų *RGB* skalę su 256 spalvų reikšmėmis. Ši konvertacija nėra būtina, tačiau užtikrinama, kad gauti kadrai *RGB* formato ir bus tikslus kadrų apdorojimas. Šioje paprogramėje yra nustatoma mažiausia kadrų reikšmė, tam, kad nustatyti slenkstį, nuo kada sekami objektai yra fiksuojami. Sekami yra fiksuojami į patikimus (angl. *reliable*) ir nepatikimus (angl. *unreliable*). Tarp jų skirtumas yra toks, kad patikimi yra jau seni objektai, kurie buvo anksčiau užfiksuoti, o nepatikimi tai nauji objektai. Todėl šioje funkcijoje yra aprašomi patikimi ir nepatikimi objektai ir abiem priskiriamos būsenos naujas „N“. Tiek patikimiems (seniems) ir nepatikimiems (naujiems) objektams yra sukuriamas prognozuojamas aprėpties langas, sukuriamas identifikavimo kodas ir nustatoma būsena. Taip originaliame ir fono substrakcijos vaizduose yra sukuriami keturkampiai aprėpties langai, kurie seka objektus. Šioje paprogramėje naujus objektus detektuoja raudonu keturkampio formos aprėpties langu, o geltonu aprėpties langu detektuoja tada, kai jau objektas yra atpažintas. Tačiau kiekvieną kartą, jei tas pats asmuo pasirodo antrą kartą, jis iš naujo yra identifikuojamas raudonu aprėpties langu, bet po to jis yra atpažįstamas ir identifikuojamas senu buvusi kodu bei geltonu aprėpties langu. Taip yra dėl to, kad kiekvieną kartą, nesvarbu ar atsiradus naujam ar jau ankčiau identifikuotam asmeniui visada fiksuojamas kaip naujas, nes vykdomas *SVM* klasifikatoriaus savybių grupių palyginimas. Galutiniame rezultate originalus ir dinamio vaizdo kaukės vaizdai atvaizduojami *MATLAB* programos leistuvuose su asmenų aprėpties langais ir identifikavimo kodais.

Funkcija *identifikuotiasmeni* atlieka asmenų identifikavimą, kai asmuo pasirodo antrą ar daugiau kartų tame pačiame vaizde (kadre). Čia yra visas pagrindas asmenų sekimui, kad tam pačiam asmeniui yra suteikiamas tas pats identifikavimo kodas. Šis procesas yra ciklas, kuris vyksta nuo pat pradžios iki kiek asmenų bus fiksuojama vaizde. Galima teigti, kad tai begalinis ciklas, kuris vyks nuolatos, kol asmenys bus matomi. Iš pradžių yra vykdoma *if* sąlyga, kuri naujus asmenis identifikuoja kaip „N“ ir išsaugo jų duomenis masyve. Yra skaitiklis, kuris skaičiuoja asmenis, tačiau šioje funkcijoje yra svarbus *SVM* klasifikatorius, kuris atlieka asmenų savybių palyginimą. Kaip anksčiau buvo minėta, kad *SVM* pradeda lyginti grupes, tik tada, kai yra jau sudarytos bent trys grupės, jei mažiau yra grupių tada lyginimo *SVM* klasifikatorius neatlieka. Šiuo atveju, duomenų bazėje turi būti sukaupta bent dviejų asmenų savybės (angl. *features*), ir kai atsiranda trečio asmens duomenys, tik tada pradedami jie lyginti su pirmais dviejų asmenų duomenimis. Todėl naudojamas skaitiklis, kuris suskaičiuoja iki trijų asmenų ir tada pradeda veikti *SVM* klasifikatorius. Šis palygina trečią duomenų grupę su dviem pirmomis ir jei kažkurios grupės savybės sutampa, tada asmenį fiksuoja kaip atpažintą „A“, kitu atveju asmenį tik suteikia jam identifikavimo kodą ir toliau lygina sekančias duomenų grupes. Jei atsiranda naujas asmuo, tiesiog duomenų bazę papildo naujo asmens savybių taškais. Ši paprogramėje vyksta mašininis mokymas (angl. *Machine learning*), apie kurį išsamiau bus kalbama sekančiame skyriuje.

Funkcija *papildyti_dataset* vykdo ciklą tokį patį kaip anksčiau minėta funkcija *identifikuotiasmeni*. Duomenys yra kaupiami tol kol vyksta asmenų stebėjimas. Sąlyga *if* nustato asmens būseną ar jis yra naujas ir ši programa vyksta tik tada kai asmuo yra naujas. Tada pagal aprėpties stačiakampį asmuo yra iškerpamas ir jo iškirptas kadras yra lyginamas su slenkstinėmis užsiduotomis reikšmėmis, kurios jau buvo anksčiau minėtos, tai mažiausias aukštis (*min person height*, *min person width*). Šitie parametrai turi būti tokie patys kaip buvo naudojami mašininiam apmokymui. Toliau yra inkrementuojamas savybių kiekis vienetu, tam, kad būtų turima daugiau savybių mašininiam mokymui.

Paprogramėje pradinių grupių kūrimas, yra naudojama funkcija *pradines_grupes*. Šioje funkcijoje yra kuriamos pradinės grupės *SVM*. Kaip anksčiau buvo minėta, kad *SVM* atlieka palyginimą, kai jau yra bent trys grupės. Šiuo atveju yra sukuriamos grupės asmuo tampa identifikuotas „I“ tada duomenų bazę papildoma savybių taškais, kurie bus reikalingi apmokymui. Taip pat yra svarbu, kad apmokymui būtų naudojama vienodo dyžio aprėpties langai. Nes jei asmuo buvo aptiktas nutolęs, tai reiškia, kad aprėpties langas bus mažas, jei arčiau vaizdo kameros, tada bus didesnis aprėpties langas. Todėl yra naudojamos papildomos funkcijos *ClasiPerHeight* ir *ClasiPerWidth*, kurios sulygina visus aprėpties langus. Po kiekvieno kadro yra didinamas savybių kiekis tam, kad būtų išsaugotos asmens savybė ir tiksliau klasifikatorius identifikuotų asmenis.

Kuo daugiau savybių pateiksime *SVM* klasifikatoriui tuo tiksliai identifikuos tą patį žmogų, tačiau apdorojimo laikas didėja ir reikia daug sistemos resursų apdorojimui.

Programos pradžios funkcija yra *programos_pradzia*. Šioje paprogramėje sukuriama grupė duomenų, tai yra vienas grupės narių, kuriame nėra kadre užfiksuotų asmenų, tik pats fonas. Pradžioje naudojama tik dvidešimt kadru tam, kad būtų sukurtas statinio fono modelis. Šio modelio savybės taip pat yra įtraukiamos į apmokymų grupę ir fiksuojamas kaip grupės narys. Tai suteikia didesnę tikslumą klasifikatoriui.

3.1.2 *MATLAB* mašininis apmokymas

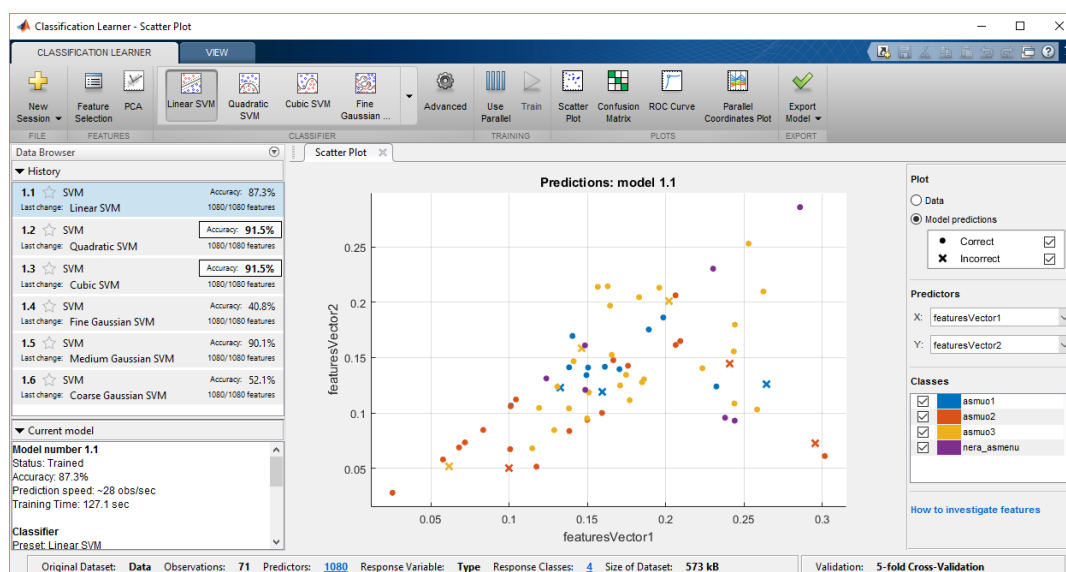
Visai neseniai *MATLAB* programiniame pakete buvo pridėtas papildomas įrankis (angl. *tool*) mašininis apmokymas (angl. *machine learning*) apie kurį buvo paminėta ankstesnėje dalyje. Šis įrankis yra išskirtinis tuo, kad galima atlikti su juo specialius apmokymus, tačiau reikia turėtų duomenų rinkinius, kurie bus naudojami apmokymui. Papildžius duomenų, jis atlieka apmokymą ir įkelia apmokymo programą į pagrindinę programą (*main*).

Objektų identifikavimui ir sekimui mašininis apmokymas yra svarbus. Šiuo atveju svarbiausia dalis yra *SVM* klasifikatoriaus apmokymas. *MATLAB* pakete yra galimybė pasirinkti *SVM* klasifikatoriaus tipus. Kaip ankstesnėje dalyje buvo paminėta, kad yra sukuriamos asmenų savybių grupės tam, kad galėtų *SVM* klasifikatorius jas palyginti. Apsirašyti programiškai tai yra sunku, nes sukauptus vis naujus duomenis reikia lyginti, o tai turi vykti netrūkstamai jau su vis daugiau sukauptų duomenų grupių. Todėl *MATLAB* mašininis apmokymas tai vykdo, jam suteikus tam tikras duomenų grupes. Šio darbo tikslas yra atpažinti tuos pačius asmenis su tais tokiais pačiais identifikavimo kodais, kai asmuo pasirodo tame pačiame vaizde du kartus arba daugiau kartų. Todėl svarbu yra kaupti duomenų bazėje asmenų savybes, jas kiekvieną kartą lyginti ir pagal anksčiau minėtus metodus juos atpažinti. Anksčiau minėti tokie savybių išskyrimo metodai buvo kaip *HOG* ir *LBP*, kurie naudojami asmenų savybėms nustatyti. Programiškai aprašyti ir pasirinkti iš vieną metodų, galima *SVM* klasifikatorių apmokyti, kad šis kiekvieną kartą lygintų asmenų savybes ir nustatų ar tos savybės sutampa .

Eksperimentas buvo atliekamas taikant *HOG* ir *LBP* savybes. Buvo tiriama trijų asmenų savybės. Pirmas bandymas su *SVM* klasifikatoriumi buvo naudojama *HOG* savybės. Duomenų bazėje buvo išsaugoma kuo daugiau kiekvieno asmens kadru, kad savybių nustatymas būtų kuo tikslesnis. Tačiau jei iškerpama daugiau kadru, tuo yra ilgesnis savybių nustatymo laikas. Pateikus duomenų bazę su *HOG* savybės į mašininį apmokymą buvo atliekamas *SVM* klasifikatoriaus apmokymas, kurio tikslas yra apmokyti *SVM* klasifikatorių palyginti asmenų savybių grupes

tarpusavyje. Prieš atliekant apmokymą yra galimybė nustatyti *SVM* klasifikatorius pasitikrinimų skaičių, tai reiškia, kad po apsimokymo *SVM* klasifikatorius pasilieka savybių iš duomenų bazės pasitikrinimui. Tai yra naudinga, nes nustatomas *SVM* klasifikatoriaus tikslumas. Taip pat galima pasirinkti klasifikatorių tipus, kurie lemia apmokymų tikslumą.

Atlikus mašininį apmokymą su linijiniu *SVM* klasifikatoriumi rezultatas gautas su 87,3 proc tikslumas. Šiuo atveju yra lyginama trijų asmenų savybių grupės ir taip pat fono savybių grupė, kiekviena atpažįstama su pateiktu procentų tikslumu. Toks rezultatas yra patenkinamas. Tačiau siekiama tikslesnio rezultato. Todėl galima atlikti apmokymus su kitokiais *SVM* klasifikatoriaus tipais. Apmokymų tikslumai procentais pateikti 3.5 pav

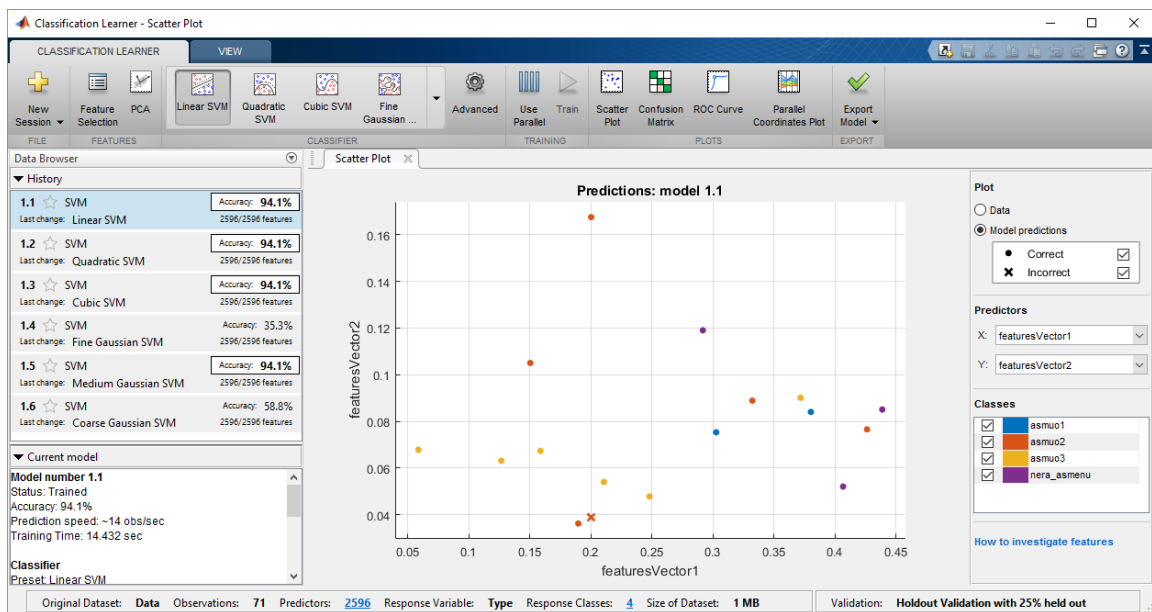


3.5 pav. *SVM* klasifikatoriaus apmokymas naudojant *HOG* išskyrimo savybes, kurio didžiausias tikslumas 91,5 proc naudojant kubinio ir kvadratinio tipo *SVM* klasifikatorius

Iš pateikto 3.5 paveikslėlio galima daryti išvadą, kad *SMV* kubinio ir kvadratinio klasifikatorių tikslumas yra didžiausias iš visų klasifikatorių taikant *HOG* savybių išskyrimo būdą. Tačiau galima ir atlikti su kitokiomis savybėmis kaip *LBP*.

Bandymas atliekamas toks pat kaip su *HOG* savybėmis. Naudojamos trijų asmenų savybių grupės, tačiau šiuo atveju naudojama *LBP* savybės. Kurio turi daug skirtumų lyginant su *HOG* savybėmis, apie kurias buvo kalbama metodinėje dalyje. Atlikus eksperimentą su *SVM* klasifikatoriaus visais tipais, rezultatai buvo daug geresni negu atliekant su *HOG* išskyrimo savybėmis. Didžiausias tikslumas gautas yra kubiniu ir kvadratinu *SVM* klasifikatoriais kurų tikslumas yra 94,1 proc . Tai yra labai aukštas tikslumas ir tai reiškia, kad neteisingai atpažinti asmenį yra tikimybė 5,9 proc. Šiuo atveju su *LBP* savybėmis *SVM* klasifikatorius palygina savybių

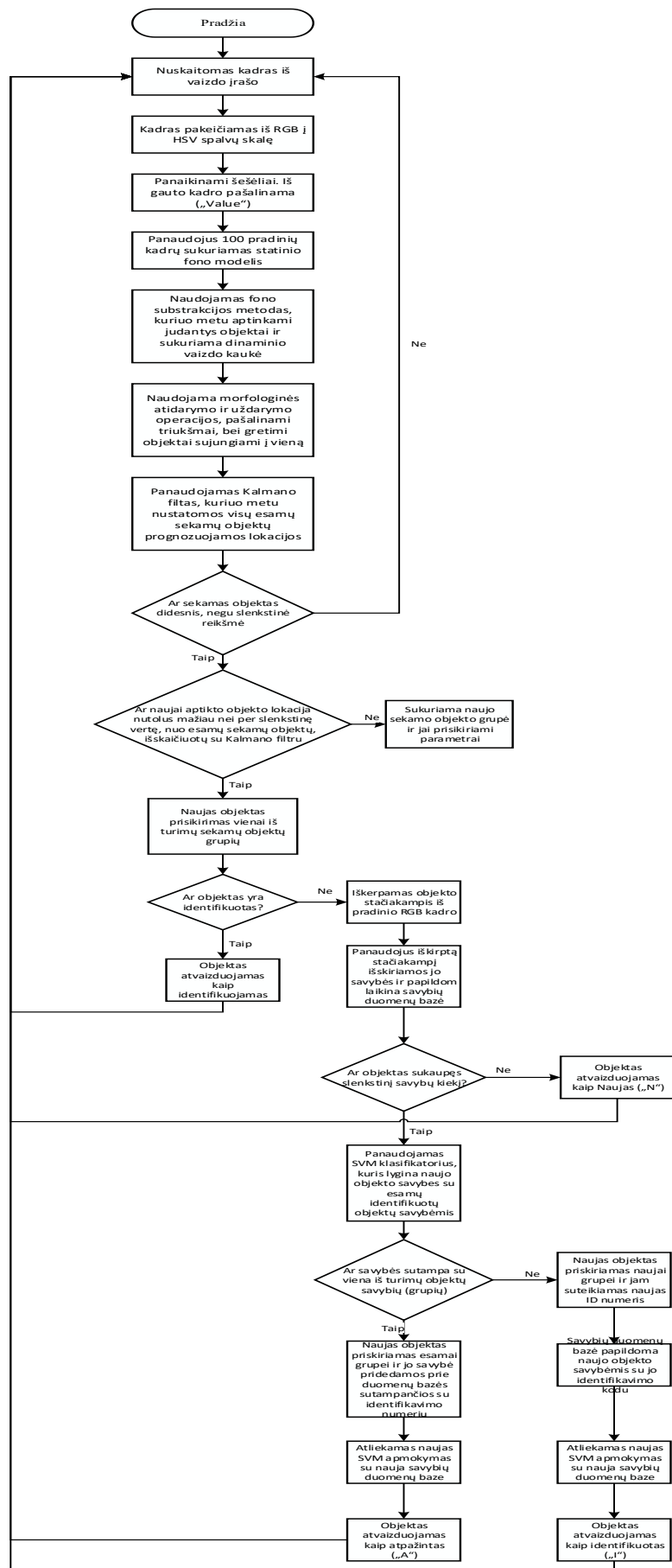
grupės ir didesniu tikslumu nustato ar aptiktų asmenų savybės yra tos pačios. Pagal tai atpažįstamas tas pats objektas, tokiu pačiu identifikavimo kodu. Vienas iš galimų veiksmų, kuris įtakoja tikslumą naudojant *HOG* savybių nustatymo metodą, tai aprėpties langas. Šis aprėpia ne tik, pačio siluetą, bet ir fono dalį. Kai su *HOG* metodu apdojamas detektuotas asmenuo, paaimamas visas aprėpties langas, kuriame yra objektas ir fonas. Todėl savybės kaupiamos ne tik asmens, bet ir fono dalies savybės. Tai ir gali sukelti netikslumų lyginant savybes. *HOG* savybių metodas naudoja *RGB* spalvų skalę, o *LBP* tik Grėjaus spalvų skalę. Dėl to tikslumas *LBP* savybių metodo yra tikslesnis, nes nėra daug įvairių spalvų ir tekstūros kontrastų.



3.6 pav. SVM klasifikatoriai naudojant *LBP* tipo savybes, kurių tikslumas 94,1 proc

3.1.3 Programos algoritmas ir jo aprašymas

Darbo užduotis yra atlikti bandymą panaudojant *MATLAB* programinį paketą ir vaizdo sistemą. Visa tai, kad tiksliai veiktų būtina sudaryti algoritmą su pasirinktais metodais, kurie buvo paminėti metodinėje dalyje. Kadangi vaizdo apdorojimo sistemą sudaro aparatinė įranga ir programinė dalis. Aparatinę įrangą sudarytų vaizdo kameros ir kompiuteris. Tačiau programinę dalį sudarytų visas *MATLAB* progamos veikimo algoritmas, kuris būtų sudarytas iš daugelio sudėtinių funkcijų 3.7 pav.



3.7 pav. Programos algoritmas

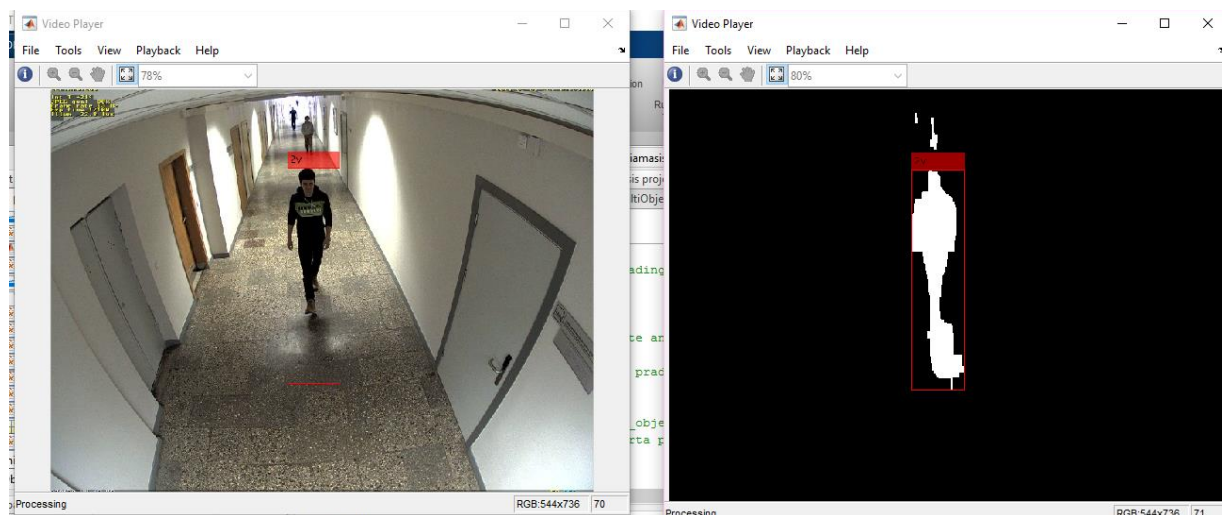
Sukurtos programos algoritmas pradeda veikti nuo gaunamo vaizdo iš vaizdo kameros. Tai gali būti realaus laiko vaizdas ar jau sukurtas vaizdo įrašas, tačiau algoritmas yra išliks toks pat. Iš vaizdo įrašo yra nuskaitomi kadrai. Funkcija *readframe* atlieka kadru nuskaitymą iš vaizdo, tai jau buvo minėta anksčiau. Gautas kadras yra *RGB* skalės spalvų, todėl jį reikia paversti į *HSV* spalvų skalę. Šis keitimas yra reikalingas dėl to, kad būtų pašalinta šešėlių įtaka, nes buvo paminėta ankstesnėje dalyje, kad pagrindinėje programoje, kur atliekamas kadru skaitymas, sekami objektai yra judantys objektai, tai reiškia kad ir šešėliai yra judantys objektai. Tačiau šešėliai neturi spalvos, dėl to pakeitus spalvų skalę bei nustačius atspalvio ir „value“ reikšmes, panaikinama šešėlių įtaka. Atliekant fono substrakciją (išskyrimą) yra sukuriamas statinio fono modelis, tai yra užfiksuojamas kadre tik fonas. Tam reikia panaudoti šimtą pradinių kadru be jokių judančių objektų. Kai yra naudojamas fono išskyrimo metodas yra aptinkami tik judantys objektai ir sukuriama dinaminio vaizdo kaukė, kurioje objektai yra pavaizduoti baltai juodame fone, ir matomas tik jų siluetas ir judėjimas. Po to sekantis etapas yra morfologinė operacija. Šioje operacijoje vyksta atidarymas ir uždarymas objekto, pašalinami triukšmai tam kad vaizdas būtų tikslus, taip pat gretimi objektai yra sujungiami vienas į kitą tam, kad neliktų skylių dinaminės kaukės vaizde (žr. 3.2 pav.). Po to seka Kalmano filtro panaudojimas sekamų objektų nuspėjamos naujoms lokacijoms nustatyti. Įvykus visos šioms operacijoms priartėja *if* sąlyga, kurioje keliama sąlyga: „Ar objektas yra didesnis, negu slenkstinės reikšmės“. Jei ši sąlyga netenkinama, tada objektas yra nefiksuojamas ir grįžtama į pradinį etapą į kadro nuskaitymą iš vaizdo. Jei sąlyga tenkinama, tada vėl yra vykdoma kita *if* sąlyga: „Ar naujai aptikto objekto lokacija nutolus mažiau nei per slenkstinę vertę, nuo esamų sekamų objektų, išskaičiuotų su Kalmano filtru“. Jei ši sąlyga nevykdoma, tada yra sukuriama nauja grupė ir jai suteikiami parametrai. O jei ši sąlyga vykdoma tada naujas objektas priskiriamas vienai iš turimų objektų duomenų grupių. Toliau yra vykdoma *if* sąlyga „Ar objektas yra identifikuotas“. Jei ši sąlyga tenkinama tada ekrane matome, kad objektas yra atvaizduotas kaip identifikuotas „I“ ir grįžtama į algoritmo pradžią, kur vyksta iš naujo kadru nuskaitymas. Tačiau jei ši *if* sąlyga nevykdoma, tada vykdoma objekto iškirpimo operacija. Iš pradinio *RGB* kadro iškerpamas objekto stačiakampis tam, kad programa galėtų iš objekto išskirti savybes. Tačiau iškirpimas gali būti vykdomas ir tada kai yra sukurta naujo sekamo objekto grupė. Atlikus iškirpimo operaciją, panaudojamas iškirptas stačiakampis kaip anksčiau buvo minėta, išskirti objekto savybes ir papildyti laikiną savybių duomenų bazę.

Įvykdžius šias operacijas iki objekto savybių išsaugojimo laikinoje duomenų bazėje, toliau vėl yra vykdoma *if* sąlyga „Ar objektas yra sukaupęs slenkstinį savybių kiekį“. Jei objektas nėra sukaupęs reikiamą savybių, jis yra atvaizduojamas kaip naujas „N“ ir algoritmas ties šiuo punktu

užsibaigia ir grįžta į pradžią į kadro nuskaitymą iš vaizdo, kaip ir identifikuotas objektas. Bet jei ši sąlyga vykdoma tada yra naudojamas *SVM* klasifikatorius, kuris lygina ir nustato ar naujo objekto savybės su esamų identifikuotų objektų savybėmis sutampa. Toliau vėl vykdoma *if* sąlyga „Ar savybės sutampa su viena iš turimų objektų savybių grupių“. Po šios sąlygos seka du algoritmai, vienas iš jų tenkina sąlyga, kitas netenkina. Kuris tenkina tada naujas objektas priskiriamas esamai grupei ir jo savybės pridedamos prie bendros duomenų bazės sutapusių grupės *ID* numeriu. Po to atliekamas naujas *SVM* klasifikatoriaus apmokymas jau su nauja savybių duomenų baze. Po apmokymo objektas atvaizduojamas ekrane kaip atpažintas „A“ ir šis algoritmas grįžta į pradžią. Jei ši *if* sąlyga netenkinama tada naujas objektas priskiriamas naujai grupei ir jam suteikiamas naujas *ID* numeris. Tada duomenų bazė, kurioje yra saugomos objektų savybės, papildoma bazė naujo objekto savybėmis su identifikavimo kodu *ID*. Po to atliekamas naujas *SVM* klasifikatoriaus mokymas jau su nauja papildyta savybių duomenų baze. Po šio apmokymo objektas atvaizduojamas ekrane kaip identifikuotas „I“ ir visas algoritmas grįžta į pradžią. Šis procesas yra niekada nesibaigiantis, nebent tik tada kai nebus gaunamas vaizdas iš vaizdo kameros.

3.1.4 Eksperimentiniai rezultatai su mažu asmenų skaičiumi

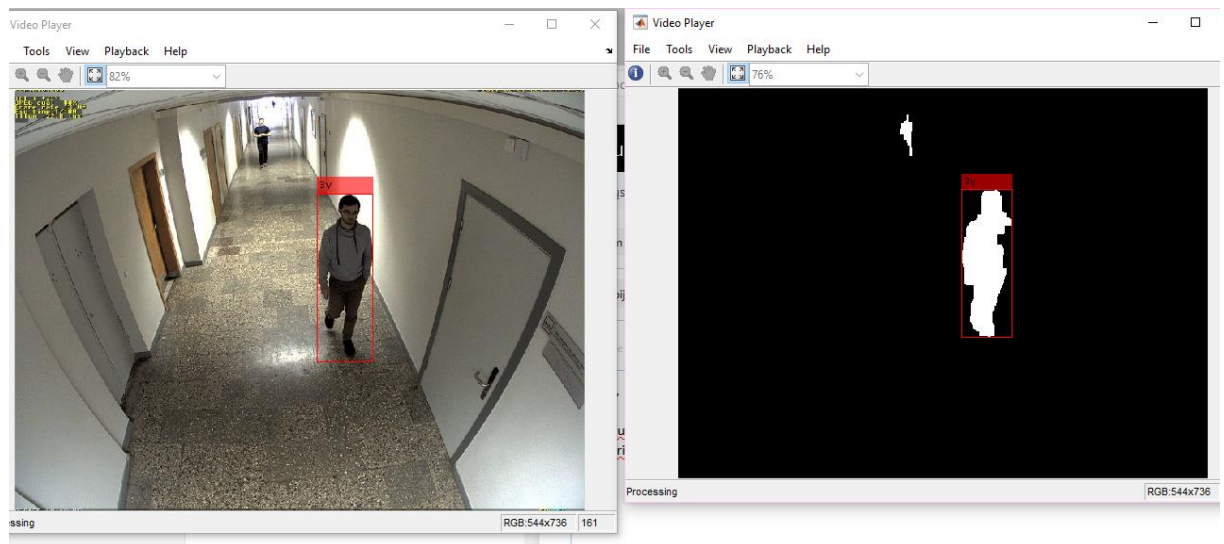
Atlikus mašininį apmokymą, buvo atliktas eksperimentas su vaizdo įrašu, kuriame yra trys asmenys. Pirmas bandymas buvo atliktas pritaikant *LBP* metodo objektų išskyrimo savybes. Identifikavimo eksperimento rezultatai pateikti paveikslėliuose 3.8 pav – 3.10 pav.



a)

b)

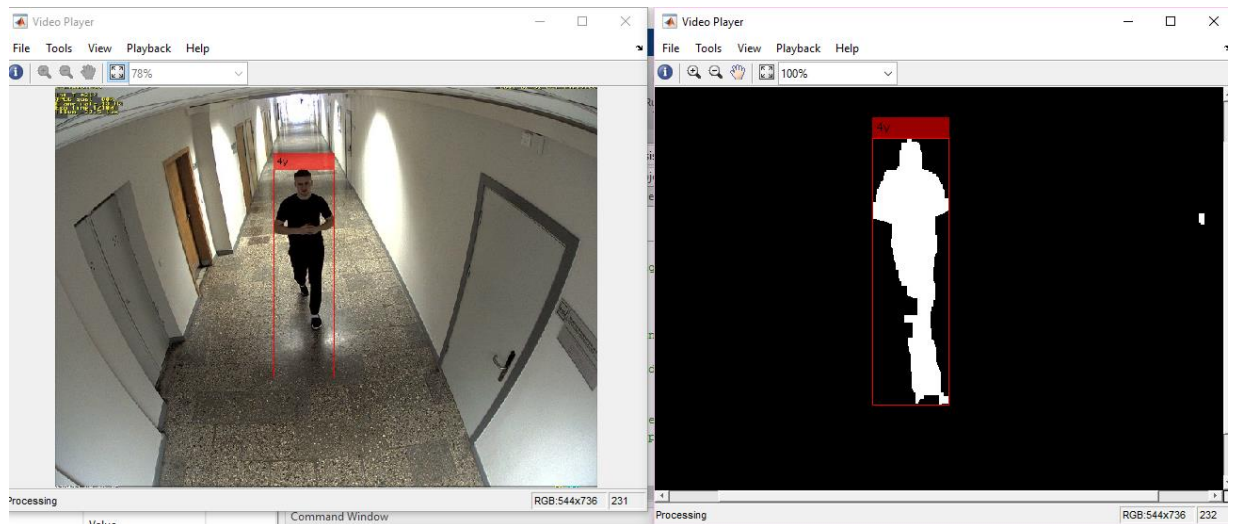
3.8 pav. Identifikuojamas pirmas asmuo: a – originalus vaizdas b- fono substrakcija



a)

b)

3.9 pav. Identifikuojamas antras asmuo: a – originalus vaizdas b- fono substrakcija

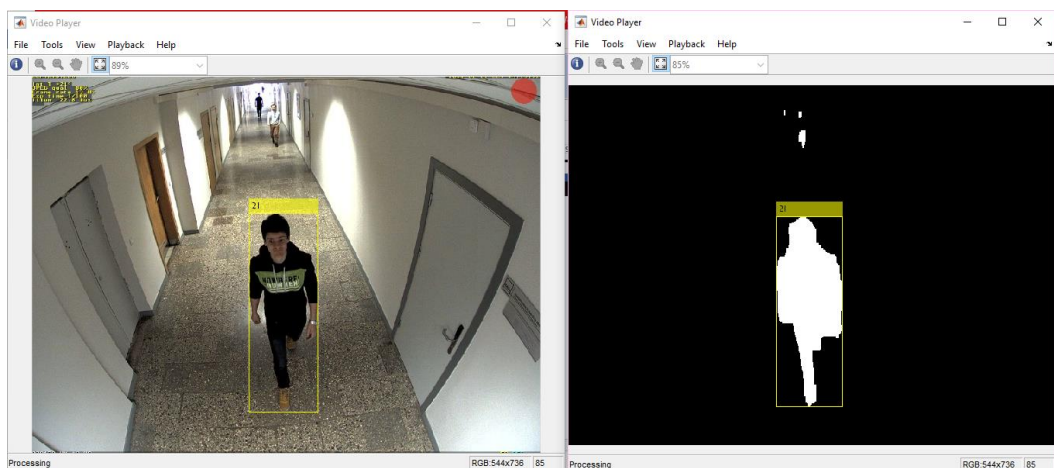


a)

b)

3.10 pav. Identifikuojamas trečias asmuo: a – originalus vaizdas b- fono substrakcija

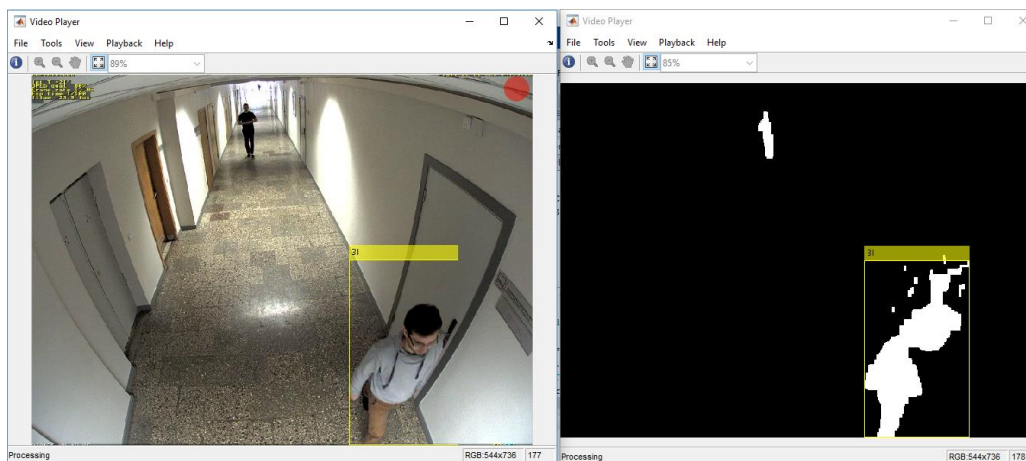
Kaip matyti iš pateiktų paveikslėlių, identifikavimas atliekamas su raudonu aprėpties langu. Kodai suteikiami didėjimo tvarka. Pirmas asmuo yra identifikuojamas skaičiumi 2 dėl to, kad pirmas skaičius priskiriamas fonui, kad vyktų apmokymas. Toliau pateikta identifikavimo kodų priskyrimas kiekvienam asmeniui.



a)

b)

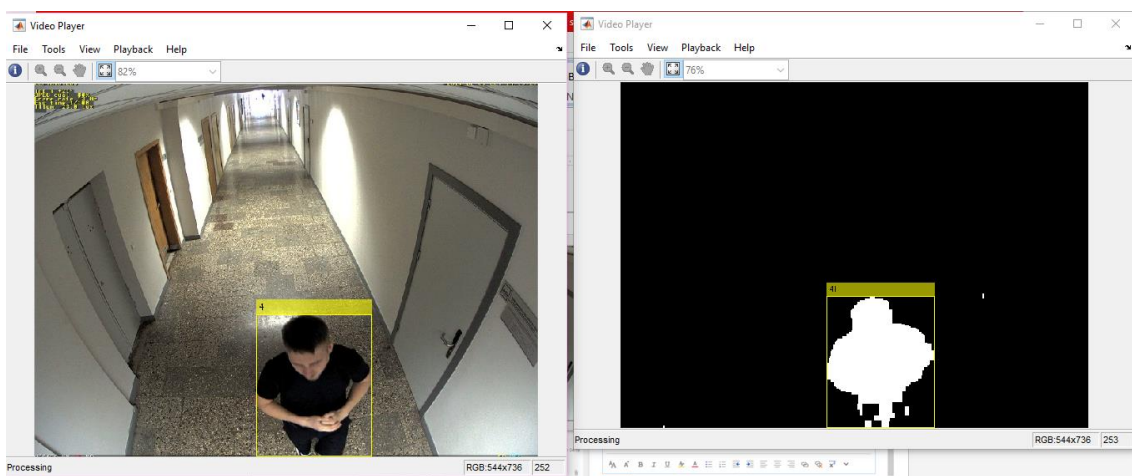
3.11 pav. Pirmas asmuo po identifikavimo: a – originalus vaizdas b- fono substrakcija



a)

b)

3.12 pav. Antras asmuo po identifikavimo: a – originalus vaizdas b- fono substrakcija

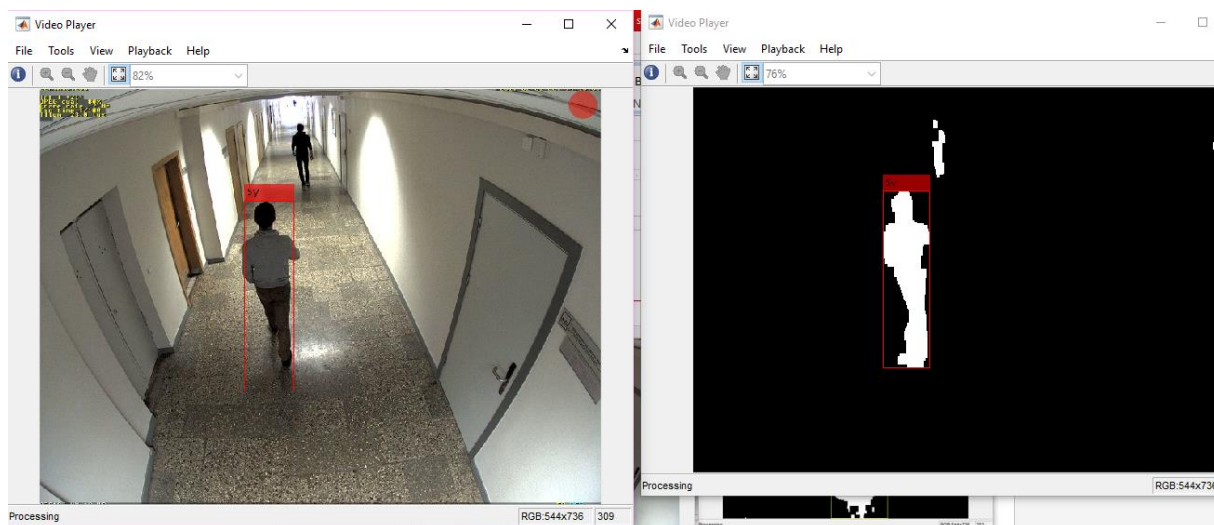


a)

b)

3.13 pav. Trečias asmuo po identifikavimo: a – originalus vaizdas b- fono substrakcija

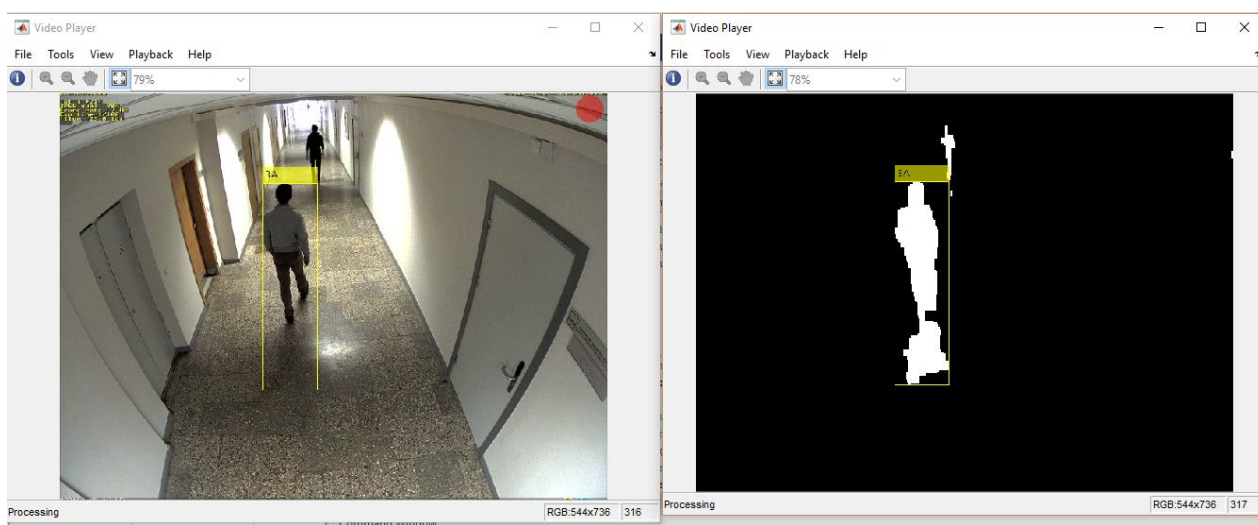
Iš pateiktų paveikslėlių 3.11 -3.13 pav., visiems asmenims suteikiami identifikavimo kodai, bei pažymimami geltonais aprėpties langais. Pagal seką yra suteikiami identifikavimo kodai su indeksu „I“. Pirmas asmuo – „2I“, antras asmuo – „3I“, trečias asmuo – „4I“.



a)

b)

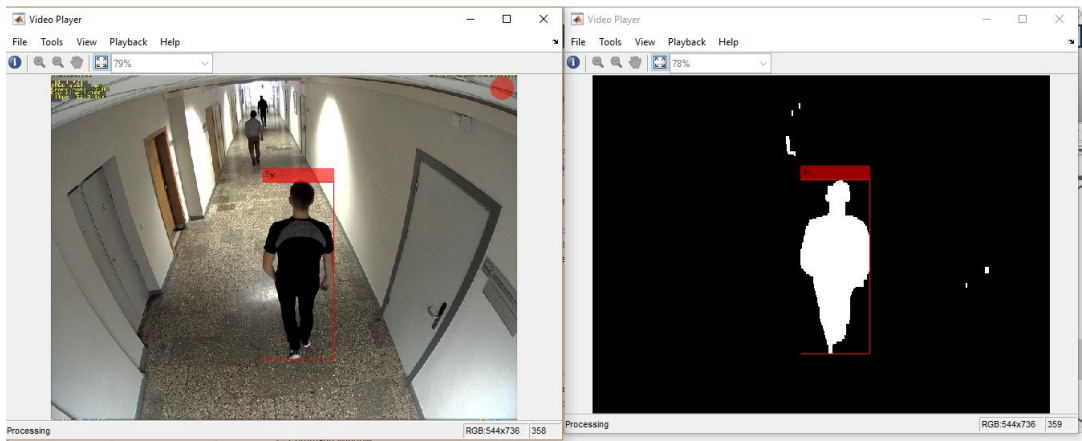
3.14 pav. Iš naujo identifikuojamas antras asmuo: a – originalus vaizdas b- fono substrakcija



a)

b)

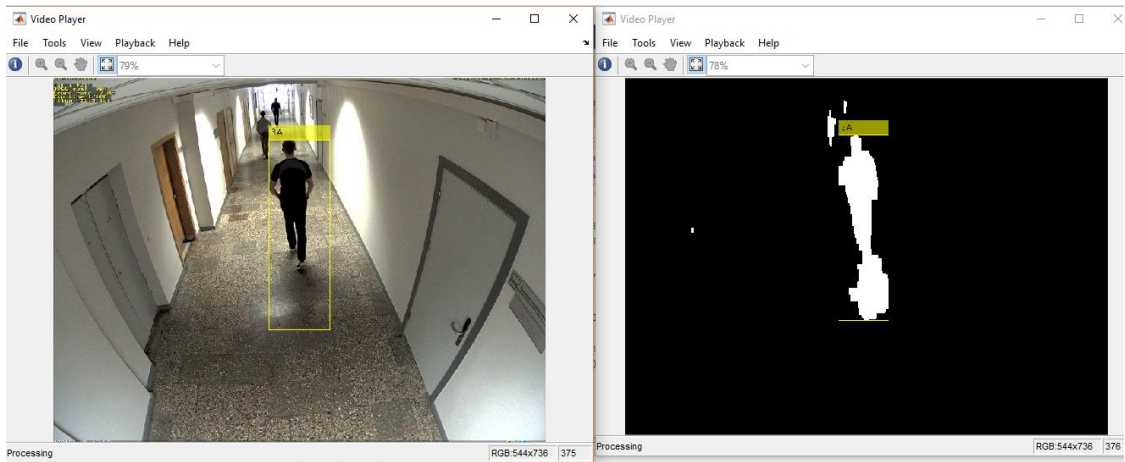
3.15 pav. Atpažįstamas antras asmuo: a – originalus vaizdas b- fono substrakcija



a)

b)

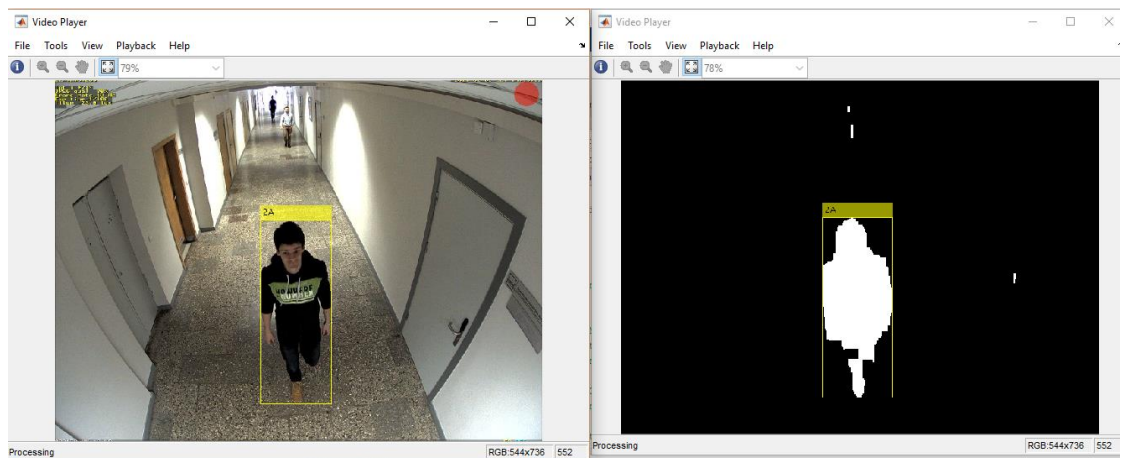
3.16 pav. Iš naujo identifikuojamas trečias asmuo: a – originalus vaizdas b- fono substrakcija



a)

b)

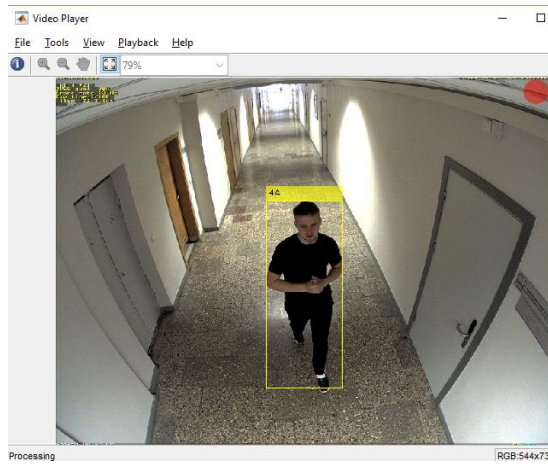
3.17 pav. Atpažįstamas trečias asmuo: a – originalus vaizdas b- fono substrakcija



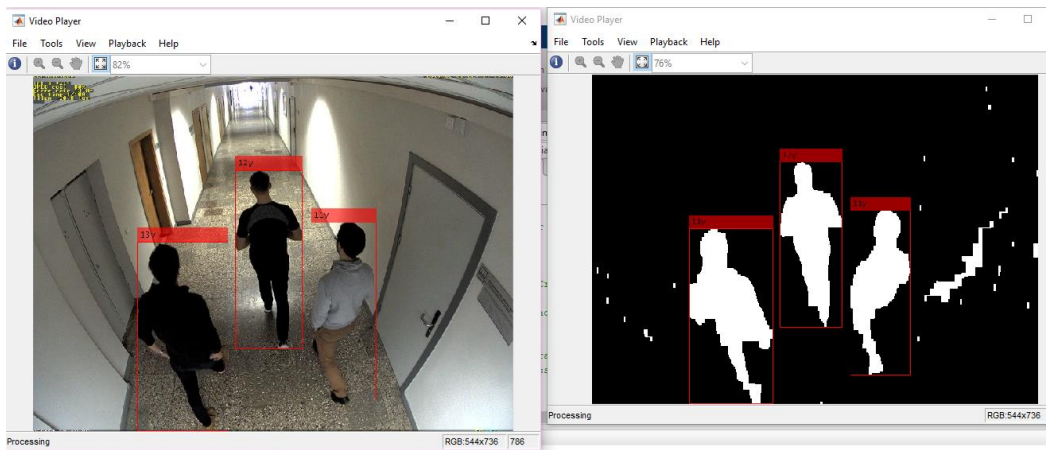
a)

b)

3.18 pav. Atpažįstamas pirmas asmuo: a – originalus vaizdas b- fono substrakcija



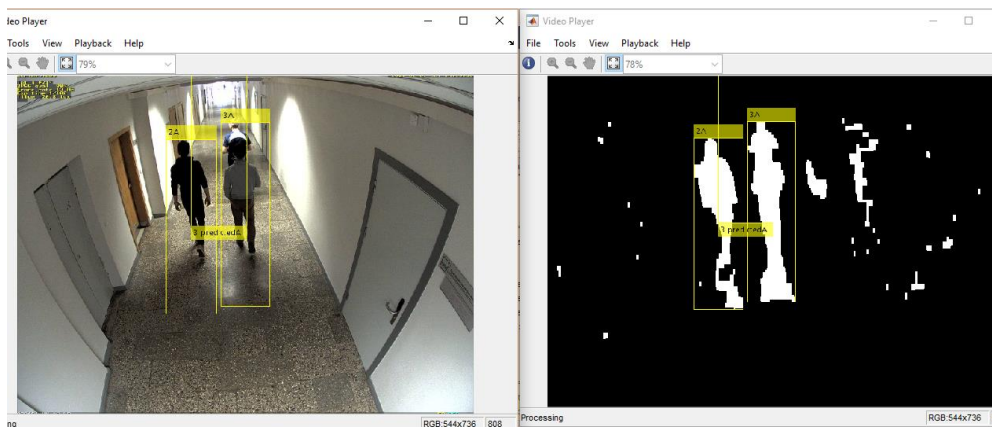
3.19 pav. Atpažįstamas trečias asmuo su teisingu kodu



a)

b)

3.20 pav. Identifikuojami visi trys asmenys vienu metu: a – originalus vaizdas b- fono substrakcija



a)

b)

3.21 pav. Atpažįstami visi trys asmenys vienu metu: a – originalus vaizdas b- fono substrakcija

Iš gautų rezultatų, galima pamatyti trūkumą. Trečias asmuo yra nevisada tiksliai atpažįstamas pagal identifikavimo kodą. Kai kada yra suteikiama antro asmens identifikavimo kodas. Tai gali būti dėl savybių netinkamo priskyrimo. Trečias asmuo identifikuojamas blogu kodu, tada kai jis yra tamsesnėje zonoje, todėl tai gali būti kitos savybės. Tačiau objektų sekimas ir identifikavimas yra pakankamai tikslus.

3.1.5 Eksperimentiniai rezultatai su dideliu asmenų skaičiumi

Anksčiau buvo atliktas eksperimentas su mažu asmenų skaičiumi. Rezultatas buvo tikslus, *SVM* klasifikatoriaus didžiausias tikslumas gautas 94,1 proc pritaikant *LBP* išskyrimo savybes, nes asmenys dideliu tikslumu buvo identifikuoti ir atpažinti. Tačiau šiuo atveju *SVM* klasifikatorius naudojo tik keturias grupes: pirmo asmens savybės, antro asmens savybės, trečio asmens savybės ir fono savybės. Todėl panaudojus *HOG* ar *LBP* savybių išskyrimą, klasifikatorius labai tiksliai atskiria skirtingus asmenis.

Sekantis eksperimentas buvo atliktas su dideliu asmenų skaičiumi. Šiuo atveju vaizdo įrašai buvo sukurti, kuriuose buvo aštuoni asmenys. Šie buvo filmuoti to pačio koridoriaus galuose. Asmenų nenutrūkstamui sekimui būtina juos identifikuoti kodais ir juos sekantį kartą atpažinti su vienodu prieš tai suteiktu identifikavimo kodu. Programos algoritmas yra toks pats ir programa yra tokia pati. Tačiau su didesniu asmenų skaičiumi yra susiduriama su savybių išskyrimo problema, nes šiuo atveju yra daugiau apmokymui reikalingų grupių: aštuoni asmenys ir fonas. Todėl *SVM* klasifikatoriui yra sunkiau suklasifikuoti grupes. Pradžioje buvo atliekamas bandymas su *HOG* ir *LBP* savybių išskyrimo būdais. Tikslumai buvo nepakankami, nes didžiausi tikslumai su *LBP* gauta 63,6 proc, o su *HOG* 61,8 proc, parodyta Priede 4. Naudojant abi savybes kartu tikslumas pagerėjo iki 73,3 proc. Todėl šiuo atveju buvo panaudota daugiau objektų išskyrimo savybių: *hsvHistogram*, *colorAutoCorrelogram*, *colorMoments*, *gaborWavelet*, *waveletTransform*.

SVM klasifikatoriaus tikslumai procentais su įvairiomis savybėmis pateikti 3.1 lentelėje.

3.1 lentelė. SVM klasifikatorių tikslumai pagal asmenų išskyrimo savybes

Savybių išskyrimo būdai	SVM klasifikatoriaus tikslumas %	SVM klasifikatoriaus tikslumas % pašalinus nesvarbių asmenų kadrus
Visos	64,4 (<i>quadratic SVM</i>)	68,3 (<i>quadratic SVM</i>)
<i>HOG, LPB</i>	73,3 (<i>cubic SVM</i>)	75,6 (<i>quadratic SVM</i>)
<i>HOG, LPB, HSV Histogram</i>	71,1 (<i>cubic SVM</i>)	78 (<i>cubic, linear SVM</i>)
<i>HOG, LBP, HSV Histogram, ColorautoCorrelogram</i>	71,1 (<i>quadratic SVM</i>)	78 (<i>quadratic SVM</i>)
<i>HSV Histogram, ColorautoCorrelogram</i>	40 (<i>quadratic SVM</i>)	43,9 (<i>quadratic SVM</i>)
<i>ColorautoCorrelogram, colorMoments</i>	28,9 (<i>cubic SVM</i>)	29,3 (<i>quadratic, cubic SVM</i>)
<i>ColorautoCorrelogram, colorMoments, waveletTransforms</i>	33,3 (<i>quadratic SVM</i>)	34,1 (<i>quadratic SVM</i>)
<i>HSV Histogram, ColorautoCorrelogram, colorMoments, waveletTransform</i>	44,4 (<i>linear SVM</i>)	51,2 (<i>linear SVM</i>)
<i>LBP, HSV Histogram, autoCorrelogram, colorMoments, meanAmplitude, msEnergy</i>	53,3 (<i>quadratic SVM</i>)	56,1 (<i>quadratic SVM</i>)
<i>LBP, HOG, HSV Histogram, ColorautoCorrelogram, meanAmplitude, msEnergy</i>	64,4 (<i>quadratic SVM</i>)	68,3 (<i>quadratic SVM</i>)

Didžiausias tikslumas yra gautas, kai buvo panaudotos tik *LPB* ir *HOG* savybės. Tačiau 73,3 proc. Tikslumas nėra labai didelis objektų atpažinimui. Tai reiškia, kad objektą, kuris pasirodys antrą kartą kadre jis gali būti neatpažintas su 26,7 proc tikimybe. Sekant asmenis, nevisi bus atpažinti tokiu pačiu indentifikavimo kodu. Kaip matyti iš 3.1 lentelės, kad jei naudojamos tik kitos savybės, tada tikslumas mažėja. Tai reiškia, kad *HOG* ir *LPB* savybės tiksliausia išskiria asmenų savybės. Todėl šios savybės yra tinkamiausios naudoti kompiuterinėje vizualizacijoje.

Pašalinus nesvarbių asmenų (asmenys, kurie pasirodo vaizdo įrašė tik vieną kartą) kadrus, tikslumas padidėja keliais procentais. Pagal tai galima spręsti, kad nesvarbių asmenų kadru skaičius yra mažas ir apmokymui yra nereikalingi. Gautas didžiausias tikslumas 78 proc pritaikius *HOG*, *LBP* ir *HSV Histogram* išskyrimo savybių tipus.

Taip pat be *SVM* klasifikatoriaus yra naudojamas *KNN* klasifikatorius, kuris taip atlieka savybių grupių palyginimą *MATLAB* mašininiam apmokyme. Priede 4 yra keletą vaizdų, kuriuose pateikta *KNN* klasifikatoriaus tikslumai naudojant anksčiau minėtas *LBP* ir *HOG* išskyrimo savybes.

IŠVADOS IR PASIŪLYMAI

Šio darbo tikslas buvo ištirti ir pritaikyti praktiškai asmenų nenutrūkstamą sekimą, pritaikyti savybes pagal kurias nustatomi ir identifikuojami asmenys naudojant *MATLAB* programą.

Tikslui įvykti buvo iškelti uždaviniai, kurie buvo atlikti.

- Literatūros šaltiniuose yra aprašyta įvairiausių būdų ir metodų objektų aptikimui ir sekimui. Iš pradžių objektą reikia detektuoti (aptikti) gautame kadre ir po to jį sekti. Dauguma metodų yra su algoritmais, tačiau ne visi yra atlikti eksperimentiškai, tik aprašyti matematiniais modeliais. Dažniausiai aptikimui yra naudojami segmentavimo, fono substrakcijos, aprėpties lango, objekto centro nustatymo ir objekto taškų detektavimo metodai. Lyginant šiuos metodus plačiausiai taikomas objekto aptikimas su aprėpties langu. Tai yra iš paprastesnių būdų, kaip aptikti asmenis pagal jų matmenis ir juos identifikuoti su kodu. Labiausiai šis metodas taikomas *MATLAB* ir *OpenCV* programinėse aplinkose. Objekto sekimui naudojami tai pat panašūs metodai kaip ir objektų aptikimui. Objektą fiksuojant vaizde yra svarbu ne tik jį aptikti, bet identifikuoti ir sekti nenutrūkstamai stebėjimo zonoje.
- Išnanalizavimus objektų aptikimo ir sekimo metodus, buvo pasirinkta objektų detekcija su aprėpties langu ir objektų sekimas panaudojant dinaminio vaizdo kaukę arba fono substrakciją. Taikant aprėpties lango metodą buvo susiduriama su asmens matmenų pokyčio, asmens centro nustatymo, prognozuojama asmens detekcija ir mažų objektų detekcijos problemomis. Visai buvo išspręsta panaudojus integruotas ir naujas funkcijas. Dinaminės kaukės vaizde pagrindinė problema buvo šešėlių sekimas. Pritaikius *HSV* spalvų skalę buvo panaikinta šešėlių įtaka ir vaizdas tapo aiškesnis.
- Tyrimas buvo realizuotas panaudojant *MATLAB* programinį paketą bei papildomus įrankius, tokius kaip mašininį apmokymą. Bandytas iš pradžių buvo atliekamas su trimis asmenimis, dėl to, kad būtų įmanoma panaudojant *SVM* klasifikatorių lyginti grupes (*SVM* klasifikatoriui reikia mažiausiai trijų grupių palyginimui). Iš gautų vaizdo įrašų su mažu ir dideliu asmenų skaičiumi, buvo iškerpami iš jų kadrai ir sutalpinami į duomenų bazę. Bandytas buvo atliekamas panaudojus *HOG* ir *LBP* savybes, kai yra mažas asmenų skaičius (šiuo atveju trys asmenys). Atlikus mašininį apmokymą su *SVM* klasifikatoriumi buvo nustatyta, kad asmenys yra tiksliau atpažįstami panaudojus *LBP* išskyrimo savybes. Šiuo atveju didžiausias tikslumas gautas 94,1 proc. Taikant *HOG* savybes tikslumas buvo mažesnis. Taip yra dėl to, kad *HOG* savybės ima

asmens savybes iš aprėpties lango, kuriame yra ne tik sekamas asmuo, bet ir dalis fono, kuri taip pat apdorojama. *HOG* savybės naudoja *RGB* spalvas, o *LBP* savybės naudoja tik Grėjaus skalės spalvas.

Atlikus eksperimentą su didesnius asmenų skaičiumi (šiuo atveju aštuoni asmenys), *SVM* klasifikatoriaus tikslumas sumažėja jei tik panaudojamos *LPB* ar *HOG* savybės. Taikant *LBP* išskyrimo savybes tikslumas dižiausias gautas 63,6 proc su kai buvo naudojama 16x16 celių asmens kadrai išskirti. *HOG* didžiausias tikslumas gautas 61,8 proc su, kai 16x16 buvo nustatyta celių. Todėl buvo siekiama tikslumą padidinti panaudojus daugiau išskyrimo savybių. Panaudojus abu kartu *LPB* ir *HOG* savybes tikslumas padidėjo iki 73,3 proc. Naudojant kitas išskyrimo savybes tikslumas buvo mažesnis.

Pašalinus iš duomenų bazės netinkamus asmenis, tikslumas padidėjo apie 2 proc. Dižiausias tikslumas buvo panaudojus *HOG*, *LBP* kartu su *HSV Histogram* išsaugo iki 78 proc. Iš to galima spręsti, kad asmenys, kurių kadru skaičius yra mažas, sumažina *SVM* klasifikatoriaus tikslumą. Todėl gauti didelį tikslumą reikia duomenų bazėje patalpinti kiekvieno asmens kuo didesnę kadru skaičių.

Papildomai buvo atlikta bandymas su *KNN* klasifikatoriumi asmenų savybėmis palyginti. Tačiau tikslumas yra mažesnis 2 proc. už *SVM* klasifikatoriaus rezultatus. Bandymas buvo atiktas panaudojus *HOG* ir *LBP* atskirai ir kartu. Didžiausias tikslumas buvo kai šių savybių išskyrimo tipai buvo panaudoti kartu 73,2 proc.

- Šią sukurtą sistemą galima taikyti asmenų aptikimui ir sekimui, tačiau, kad ši sistemai tinkamai veiktų reikalingas kalibravimas. Iš pradžių reikia statinio fono vaizdo, tam, kad būtų atliktas asmenų detektavimas. Sistema turi apsimokyti ir atskirti foną ir judančius objektos. Pradiniam apsimokymui reikia trijų asmenų tam, kad *SVM* klasifikatorius galėtų atlikti asmenų savybių palyginimą bei identifikuoti ir aptikti asmenis. Atlikus visą sistemos kalibravimą, sistemą toliau identifikuoti, aptikti ir sekti objektus be jokių trikdžių ir pradinių apsimokymų.

Pasiūlymai būtų, tokią sukurtą sistema realizuoti realiuoju laiku, sukurti daugiakamerinę stebėjimo sistemą, kuri būtų sinchronizuota ir atliko asmenų detektavimą, atpažinimą ir sekimą realiuoju laiku. Taip pat panaudi daugiau objektų išskyrimo savybių arba jas paotbulinti pritaikius įvairius algoritmus, kad asmuo būtų atpažįstamas kuo tiksliau, bei sumažinti *SVM* klasifikatoriaus apsimokymo laiką iki minimalaus.

INFORMACIJOS ŠALTINIŲ SĄRAŠAS

1. PRASAD I. Bhosle, LOKESH Bijole. Video Object Tracking and Segmentation Using Artificial Neural Network for Surveillance System [interaktyvus]. International Journal of Engineering Research & Technology (IJERT). Vol. 2 Issue 11, November 2013 [žiūrėta 2015m. lapkričio 12 d.]. Prieiga per internetą: <http://www.ijert.org/view-pdf/6556/video-object-tracking-and-segmentation-using-artificial-neural-network-for-surveillance-system>
2. Prof. A.P. BODKHEL, Prof. S.A. NIRMAL, Snehal A. THAKRE. A Literature Review on Different models for Human and Vehicle Tracking [ineraktyvus]. International Journal of Innovative Research in Computer and Communication Engineering Vol. 3, Issue 1, January 2015 [žiūrėta 2015 m. gruodžio 3 d.]. Prieiga per internetą: http://www.ijrccce.com/upload/2015/january/26_A_Literature.pdf
3. SEEMA Kumari, MANPREET Kaur, BIRMOHAN Singh. DETECTION AND TRACKING OF MOVING OBJECT IN VISUAL SURVEILLANCE SYSTEM [interaktyvus]. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (ISO 3297: 2007 Certified Organization) Vol. 2, Issue 8, August 2013 [žiūrėta 2015 m. lapkričio 14 d.]. Prieiga per internetą: http://www.ijareeie.com/upload/2013/august/20E_DETECTION.pdf
4. M. CHITRA, M.KALAISELVI GEETHA, L. MENAKA. Occlusion and Abandoned Object Detection for Surveillance [interaktyvus]. Applications International Journal of Computer Applications Technology and Research Volume 2– Issue 6, India [žiūrėta 2015 m. gruodžio 5 d.]. Prieiga per internetą: <http://www.ijcat.com/archives/volume2/issue6/ijcatr02061014.pdf>
5. Mykhaylo Andriluka, Jaeyeon Jung, Stefan Roth, Bernt Schiele. People-Tracking-by-Detection and People-Detection-by-Tracking [interaktyvus]. Computer Science Department, TU Darmstadt, Germany [žiūrėta 2015 m. gruodžio 1 d.] Prieiga per internetą: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.408.334&rep=rep1&type=pdf>
6. Hemangi R. Patil, Prof. K. S. Bhagat. Object Detection and Motion Based Tracking of Moving Objects: a Survey [interaktyvus]. International Journal of Innovative Research in Computer and Communication Engineering Vol. 3, Issue 1, December 2014 [žiūrėta 2015 m. gruodžio 1 d.] Prieiga per internetą: <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/sTrack20Technical20Report.pdf>

7. Martin SPRENGLER, Bernt SCHIELE. Automatic Detection and Tracking of Abandoned Objects [interaktyvus]. Perceptual Computing and Computer Vision Computer Science, Department ETH Zurich, Switzerland [žiūrėta 2015 m. gruodžio 7 d.]. Prieiga per internetą: <http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.2.5407>
8. Suspicious Object Detection and Tracking [interaktyvus]. [žiūrėta 2015 m. gruodžio 9 d.]. Prieiga per internetą: http://www.spit.ac.in/wp-content/uploads/profktalele/project/2009-10/%284%29Suspicious%20Object%20Recognition/report%20Suspicious_Object_Detection.pdf
9. Jinman KANG, Isaac COHEN, Gérard MEDIONI. Persistent Objects Tracking Across Multiple Non Overlapping Cameras [interaktyvus]. IEEE Workshop on Motion and Video Computing (WACV/MOTION'05)-Volume 2 January 05-07, 2005. IRIS, Computer Vision Group, University of Southern California, Los Angeles [žiūrėta 2015 m. gruodžio 6 d.]. Prieiga per internetą: <http://iris.usc.edu/outlines/papers/2005/kang-motion05.pdf>
10. Alper YILMAZ, Omar JAVED, Mubarak SHAH. Object Tracking: A Survey [interaktyvus]. ACM Computing Surveys, Vol. 38, No. 4, Article 13, Publication date: December 2006 [žiūrėta 2015 m. lapkričio 27 d.]. Prieiga per internetą: <http://www.ppgia.pucpr.br/~alceu/pdi/Video%20Segmentation%20and%20Tracking/Yilmaz.pdf>
11. Sumalatha RAMACHANDRAN ,Lakshmi Harika PALIVELA. Detection of Static and Dynamic Behavior in Forbidden Areas of a Complex Environment [interaktyvus]. Department of Information Technology, Anna University, Chennai, India [žiūrėta 2015 m. gruodžio 5 d.]. Prieiga per internetą: <http://www.aensiweb.com/old/anas/Special4%20EEICC%202014/33-39.pdf>
12. Reena Kumari Behera, Pallavi Kharade, Suresh Yerva, Pranali Dhane, Ankita Jain, and Krishnan Kutty. Multi-Camera Based Surveillance System [interaktyvus]. Center for Research in Engineering Sciences and Technology (CREST), KPIT Cummins Infosystems Limited, Pune 411057, India [žiūrėta 2015 m. gruodžio 7 d.]. Prieiga per internetą: <http://www.kpit.com/downloads/research-papers/multi-camera-based-surveillance-system.pdf>
13. Valtteri TAKALA and Matti PIETIKAINEN. Multi-Object Tracking Using Color, Texture and Motion [interaktyvus]. Machine Vision Group, Infotech Oulu and Dept. of Electrical and Information Engineering, P.O. Box 4500 FIN-90014 University of Oulu, Finland [žiūrėta 2015 m. gruodžio 2 d.]. Prieiga per internetą: <http://www.ee.oulu.fi/mvg/files/pdf/VS2007.pdf>

14. Martí Balcells, Daniel DeMenthon, David Doermann. AN APPEARANCE BASED APPROACH FOR CONSISTENT LABELING OF HUMANS AND OBJECTS IN VIDEO [interaktyvus]. Language and Media Processing Laboratory, University of Maryland, College Park, MD, 20742, USA [žiūrėta 2015 m. lapkričio 29 d.]. Prieiga per internetą: <http://lampsrv02.umiacs.umd.edu/pubs/Papers/martibalcellscapellades-04/martibalcellscapellades-04.pdf>
15. NAKANO Masaki, SHIBA Hiroshi, KAWAHARA Akihiro, TABUCHI Toru, KOBAYASHI Minoru, YAGI Masahiro. Harbor Monitoring Network System for Detecting Suspicious Objects Approaching Critical Facilities in Coastal Areas [interaktyvus]. NEC Technical Journal/Vol.9 No.1/Special Issue on Solutions for Society - Creating a Safer and More Secure Society [žiūrėta 2015 m. gruodžio 3d.]. Prieiga per internetą: http://in.nec.com/en_IN/images/140125.pdf
16. SOWMIYA.M, PONGODI.P.C, SHARMILA Bhargavi.V. Cognitive Object Detection Using Binary Large Object Algorithm [interaktyvus]. International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 13 Issue 2 –MARCH 2015 [žiūrėta 2015 m. gruodžio 10 d.]. Prieiga per internetą: <http://www.ijetcse.com/wp-content/plugins/ijetcse/file/upload/docx/446Cognitive-Object-Detection-Using-Binary-Large-Object-Algorithm-pdf.pdf>
17. Hsien-Chou LIAO, Wei-Yi CHEN. EAGLE-EYE: A DUAL-PTZ-CAMERA SYSTEM FOR TARGET TRACKING IN A LARGE OPEN AREA [interaktyvus]. Chaoyang University of Technology, Department of Computer Science and Information Engineering 168 Jifong E. Rd., Wufong Township Taichung County, 41349, Taiwan (R.O.C.), ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2010, Vol.39, No.3 [žiūrėta 2015 m. lapkričio 19 d.]. Prieiga per internetą: <http://itc.ktu.lt/itc393/Liao393.pdf>
18. VLADAN PAPIC, HRVOJE TURIC, HRVOJE DUJMIC. Two-stage segmentation for detection of suspicious objects in aerial and longrange surveillance applications [interaktyvus]. Department of polytechnics, Department of electronics, University of Split Teslina, CROATIA [žiūrėta 2015 m. lapkričio 21 d.]. Prieiga per internetą: <http://www.ipsar.fesb.hr/pdf/wseasPrag09.pdf>
19. K. KAUSALYA, S. CHITRAKALA. Idle Object Detection in Video for Banking ATM Applications [interaktyvus]. Research Journal of Applied Sciences, Engineering and Technology, Easwari Engineering College, Anna University of Technology, Chennai, Tamilnadu, India, Published: December 15, 2012 [žiūrėta 2015 m. gruodžio 3 d.]. Prieiga per internetą: <http://maxwellsci.com/print/rjaset/v4-5350-5356.pdf>

20. PORIKLI, F.; YILMAZ, A. Object Detection & Tracking [interaktyvus]. MITSUBISHI ELECTRIC RESEARCH LABORATORIES, January 2012 [žiūrėta 2015 m. gruodžio 6 d.]. Prieiga per internetą: <http://www.merl.com/publications/docs/TR2012-003.pdf>
21. Yamini Nimmagadda, Karthik Kumar, Yung-Hsiang Lu, and C. S. George Lee. Real-time Moving Object Recognition and Tracking Using Computation Offloading [interaktyvus]. The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 18-22, 2010, Taipei, Taiwan [žiūrėta 2015 m. spalio 28 d.]. Prieiga per internetą: <https://engineering.purdue.edu/HELPS/Publications/papers/NimmagaddaKumarLuLee2010IROS.pdf>
22. Medha BHARGAVA, Chia-Chih CHEN, M. S. RYOO, J. K. AGGARWAL. Detection of object abandonment using temporal logic [interaktyvus]. Springer-Verlag 2009 [žiūrėta 2015 m. lapkričio 15 d.]. Prieiga per internetą: http://cvrc.ece.utexas.edu/Publications/Medha_Bhargava_mva08.pdf
23. Medha BHARGAVA, Chia-Chih CHEN, M. S. RYOO, J. K. AGGARWAL. Detection of Abandoned Objects in Crowded Environments [interaktyvus]. Computer and Vision Research Center Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, Texas, USA. Prieiga per internetą: http://michaelryoo.com/papers/AVSS2007_Bhargava.pdf
24. Ying-li TIAN, Rogerio FERIS, Arun HAMPAPUR. Real-Time Detection of Abandoned and Removed Objects in Complex Environments [interaktyvus]. IBM T.J. Watson Research Center [žiūrėta 2015 m. gruodžio 10 d.]. Prieiga per internetą: <http://rogerioferis.com/publications/FerisECCVW08.pdf>
25. Viola, P.; Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features [interaktyvus]. MITSUBISHI ELECTRIC RESEARCH LABORATORIES [žiūrėta 2016 m. spalio 10 d.]. Prieiga per internetą: <http://www.merl.com/publications/docs/TR2004-043.pdf>
26. Junge Zhang, Kaiqi Huang, Yinan Yu and Tieniu Tan. Boosted Local Structured HOG-LBP for Object Localization [interaktyvus]. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science. [žiūrėta 2017 m. sausio 23 d.]. Prieiga per internetą: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.362.6498&rep=rep1&type=pdf>

27. Adrian Rosebrock. Pedestrian Detection OpenCV [interaktyvus]. Object Detection, Tutorials, November 9, 2015. [žiūrėta 2016 m. gruodžio 29 d.]. Prieiga per internetą: <http://www.pyimagesearch.com/2015/11/09/pedestrian-detection-opencv/>
28. Birgi Tamersoy. Background subtraction [interaktyvus]. The University of Texas at Austin, September 29, 2009 [žiūrėta 2015 m. lapkričio 27 d.]. Prieiga per internetą: http://www.cs.utexas.edu/~grauman/courses/fall2009/slides/lecture9_background.pdf
29. OpenCV 3.2.0-dev, Open Source Computer Vision [interaktyvus]. How to Use Background Subtraction Methods [žiūrėta 2015 m. gruodžio 5 d.]. Prieiga per internetą: http://docs.opencv.org/trunk/d1/dc5/tutorial_background_subtraction.html
30. Face Detection and Tracking Using CAMShift [interaktyvus]. MathWorks 2014 [žiūrėta 2016 m. birželio 15 d.]. Prieiga per internetą: <https://se.mathworks.com/help/vision/examples/face-detection-and-tracking-using-camshift.html>
31. Motion-Based Multiple Object Tracking [interaktyvus]. MathWorks 2015 [žiūrėta 2016 m. gruodžio 29 d.]. Prieiga per internetą: <https://se.mathworks.com/help/vision/examples/motion-based-multiple-object-tracking.html>
32. Face Recognition with Local Binary Patterns [interaktyvus]. LBP Face Recognition System [žiūrėta 2017 m. vasario 18 d.]. Prieiga per internetą: <http://www.advancedsourcecode.com/lbpface.asp>
33. Speeded-Up Robust Features [interaktyvus]. [žiūrėta 2017 m. vasario 27 d.]. Prieiga per internetą: <http://mahotas.readthedocs.io/en/latest/surf.html>
34. Face Detection using Haar Cascades [interaktyvus]. OpenCV 3.2.0-dev, Open Source Computer Vision [žiūrėta 2017 m. kovo 5 d.]. Prieiga per internetą: http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html
35. Simon Tong, Daphne Koller. Support Vector Machine Active Learning with Applications to Text Classification [interaktyvus]. Computer Science Department Stanford University, USA. Journal of Machine Learning Research, 2001 [žiūrėta 2016 m. lapkričio 28 d.]. Prieiga per internetą: <http://www.jmlr.org/papers/volume2/tong01a/tong01a.pdf>
36. Kalyan Roy, Joydeep Mukherjee [interaktyvus]. Image Similarity Measure using Color Histogram, Color Coherence Vector, and Sobel Method. Jadavpur University, School of Education Technology, M Tech Scholar, Kolkata, India. International Journal of Science and Research [žiūrėta 2017 m. balandžio 22 d.]. Prieiga per internetą: <https://www.ijsr.net/archive/v2i1/IJSRON2013311.pdf>

37. Zhi Zhou, Glen Hong Ting Ong and Eam Khwang Teoh [interaktyvus]. Soft-biometric Detection based on Supervised Learning. School of Electrical and Electronic Engineering Nanyang Technological University Singapore, Singapore. 2014 13th International Conference on Control, Automation, Robotics & Vision Marina Bay Sands, Singapore, 10-12th December 2014 [žiūrėta 2017 m. balandžio 23 d.]. Prieiga per internetą: <http://ieeexplore.ieee.org/abstract/document/7064310/>
38. Yunsheng Jiang, Jinwen Ma [interaktyvus]. Combination Features and Models for Human Detection. Department of Information Science, School of Mathematical Sciences and LMAM, Peking University, Beijing, China [žiūrėta 2017 m. kovo 31 d.]. Prieiga per internetą: http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Jiang_Combination_Features_and_2015_CVPR_paper.pdf

PRIEDAI

Priedas 1. Pagrindinė *MATLAB* programa

```
function multiObjectTracking ()

% Sukuriami sisteminiai objektai is vaizdo iraso, detektuojami ir pateikiami
obj = setupSystemObjects();

tracks = initializeTracks(); % Sukuriamas naujas masyvas aptiktiems objektams

nextId = 2; % pradeda nuo 2 nes 1 yra be zmoniu grupe

duomenysTemp=[];
duomenysIDTemp=[];
duomenys=[];
duomenysID=[];

trainedClassifier=[];

%Parametrai _____
min_person_width=50;
min_person_height=200;
ClasiPerHeight=380;
ClasiPerWidth=150;
dataThreshold= 15; %sukaupus N frame features masyvu, asmuo lyginamas su turimomis
grupemis
counter1=0;

programos_pradzia(); %atlieka pradinius konfigravimus

% Aptinkami
while ~isDone(obj.reader)
    frame = readFrame();

    %spalvu konvertavimas i HSV color map, kad pasalintu seseliu itaka
    [ H,S,~ ] = RGB_HSV( frame );
    abu_HS=H+S; % sudedu Hue ir Saturation
    abu_HS_norma=abu_HS/(max(max(abu_HS))); % normalizuoju suma, kad reiksmes iki 1
    butu

    [centroids, bboxes, mask] = detectObjects(S);
    predictNewLocationsOfTracks();
    [assignments, unassignedTracks, unassignedDetections] = ...
        detectionToTrackAssignment();

    updateAssignedTracks();
    updateUnassignedTracks();
    deleteLostTracks();
```

```

createNewTracks();
papildyti_dataset();

    identifikuotiAsmeni(); % kai aptinkamas naujas asmuo, atliekamas is naujo SVM
    apsimokymas

    displayTrackingResults();

end

%% Sukuriami sisteminiai objektai
function obj = setupSystemObjects()

    obj.reader = vision.VideoFileReader('hall_eef.mp4');

    obj.videoPlayer = vision.VideoPlayer('Position', [50, 300, 600, 400]);
    obj.maskPlayer = vision.VideoPlayer('Position', [740, 300, 600, 400]);

    obj.detector = vision.ForegroundDetector('NumGaussians', 5, ...
        'NumTrainingFrames', 100, 'MinimumBackgroundRatio', 0.6);

    obj.blobAnalyser = vision.BlobAnalysis('BoundingBoxOutputPort', true, ...
        'AreaOutputPort', true, 'CentroidOutputPort', true, ...
        'MinimumBlobArea', 8000);
end

%% Inicializuojami objektai

function tracks = initializeTracks()
    % sukuriami nauji masyvai
    tracks = struct(...
        'id', {}, ...
        'bbox', {}, ...
        'kalmanFilter', {}, ...
        'age', {}, ...
        'totalVisibleCount', {}, ...
        'consecutiveInvisibleCount', {},...
        'busena', {},...
        'featureSave', {},...
        'featureCount', {});
end

%% Vaizdo kadru nuskaitymas
function frame = readFrame()
    frame = obj.reader.step();
end

```

%% Objektų aptikimas ir morfologinė operacija

```
function [centroids, bboxes, mask] = detectObjects(frame)
```

```
    % Aptinkamas fonas
```

```
    mask = obj.detector.step(frame);
```

```
    % Morfologine operacija
```

```
    mask = imopen(mask, strel('rectangle', [8,3]));
```

```
    mask = imclose(mask, strel('rectangle', [15, 15]));
```

```
    mask = imfill(mask, 'holes');
```

```
    [~, centroids, bboxes] = obj.blobAnalyser.step(mask);
```

```
end
```

%% Prognozuojama nauja vieta esamiems objektams

```
function predictNewLocationsOfTracks()
```

```
    for i = 1:length(tracks)
```

```
        bbox = tracks(i).bbox;
```

```
        predictedCentroid = predict(tracks(i).kalmanFilter);
```

```
        predictedCentroid = int32(predictedCentroid) - bbox(3:4) / 2;
```

```
        tracks(i).bbox = [predictedCentroid, bbox(3:4)];
```

```
    end
```

```
end
```

%% Priskyrimas esamiems objektas

```
function [assignments, unassignedTracks, unassignedDetections] = ...
```

```
    detectionToTrackAssignment()
```

```
    nTracks = length(tracks);
```

```
    nDetections = size(centroids, 1);
```

```
    cost = zeros(nTracks, nDetections);
```

```
    for i = 1:nTracks
```

```
        cost(i, :) = distance(tracks(i).kalmanFilter, centroids);
```

```
    end
```

```
    costOfNonAssignment = 20;
```

```
    [assignments, unassignedTracks, unassignedDetections] = ...
```

```
        assignDetectionsToTracks(cost, costOfNonAssignment);
```

```
end
```

%%

% Priskirtų sekamų asmenų atnaujinimas

```
function updateAssignedTracks()
```

```
    numAssignedTracks = size(assignments, 1);
```

```
    for i = 1:numAssignedTracks
```

```

trackIdx = assignments(i, 1);
detectionIdx = assignments(i, 2);
centroid = centroids(detectionIdx, :);
bbox = bboxes(detectionIdx, :);
correct(tracks(trackIdx).kalmanFilter, centroid);

tracks(trackIdx).bbox = bbox;

% atnaujina objekto kadru skaiciu
tracks(trackIdx).age = tracks(trackIdx).age + 1;

tracks(trackIdx).totalVisibleCount = ...
    tracks(trackIdx).totalVisibleCount + 1;
tracks(trackIdx).consecutiveInvisibleCount = 0;
end
end

function updateUnassignedTracks()
for i = 1:length(unassignedTracks)
ind = unassignedTracks(i);
tracks(ind).age = tracks(ind).age + 1;
tracks(ind).consecutiveInvisibleCount = ...
    tracks(ind).consecutiveInvisibleCount + 1;
end
end

%% Istrinami senai nepasirodantys asmenys

function deleteLostTracks()
if isempty(tracks)
return;
end

invisibleForTooLong = 20;
ageThreshold = 8;

ages = [tracks(:).age];
totalVisibleCounts = [tracks(:).totalVisibleCount];
visibility = totalVisibleCounts ./ ages;

lostInds = (ages < ageThreshold & visibility < 0.6) | ...
    [tracks(:).consecutiveInvisibleCount] >= invisibleForTooLong;

% Istrinami dingę objektai
tracks = tracks(~lostInds);
end

```

```

%% Sukuriami nauji objektai
function createNewTracks()
    centroids = centroids(unassignedDetections, :);
    bboxes = bboxes(unassignedDetections, :);

    for i = 1:size(centroids, 1)

        centroid = centroids(i,:);
        bbox = bboxes(i, :);

        % Sukuriamas objektas su Kalmano filtru
        kalmanFilter = configureKalmanFilter('ConstantVelocity', ...
            centroid, [200, 50], [100, 25], 100);

        % Sukuriamas naujas objektas
        newTrack = struct(...
            'id', nextId, ...
            'bbox', bbox, ...
            'kalmanFilter', kalmanFilter, ...
            'age', 1, ...
            'totalVisibleCount', 1, ...
            'consecutiveInvisibleCount', 0, ...
            'busena', 'N',...
            'featureSave', 'Y',...
            'featureCount', 0);

        % Pridedamas objektas i masyva
        tracks(end + 1) = newTrack;

        % Increment the next id.
        nextId = nextId + 1;
    end
end

%% Display Tracking Results
% I ekrana isvedami sekimo rezultatai
function displayTrackingResults()
    % Convert the frame and the mask to uint8 RGB.
    frame = im2uint8(frame);
    mask = uint8(repmat(mask, [1, 1, 3])) .* 255;

    minVisibleCount = 25;
    if ~isempty(tracks)

        % Trumpai pasirode asmenys
        reliableTrackInds = ...
            [tracks(:).busena] ~= 'N';
        reliableTracks = tracks(reliableTrackInds);

        UNreliableTrackInds = ...

```

```

[tracks(:).busena] == 'N';
UNreliableTracks = tracks(UNreliableTrackInds);

if ~isempty(reliableTracks)

    bboxes = cat(1, reliableTracks.bbox);
    .
    ids = int32([reliableTracks(:).id]);

    busenos = int32([reliableTracks(:).busena]);
    busenos = cellstr(char(busenos));

    % Create labels for objects indicating the ones for
    % which we display the predicted rather than the actual
    % location.
    labels = cellstr(int2str(ids));
    predictedTrackInds = ...
        [reliableTracks(:).consecutiveInvisibleCount] > 0;
    isPredicted = cell(size(labels));
    isPredicted(predictedTrackInds) = {'predicted'};
    labels = strcat(labels, isPredicted, busenos);

    frame = insertObjectAnnotation(frame, 'rectangle', ...
        bboxes, labels);
    mask = insertObjectAnnotation(mask, 'rectangle', ...
        bboxes, labels);
end

% Nauja: atvaizduoti ir nepatikimus track
if ~isempty(UNreliableTracks)

    bboxes2 = cat(1, UNreliableTracks.bbox);
    ids2 = int32([UNreliableTracks(:).id]);

    featureSaves = int32([UNreliableTracks(:).featureSave]);
    featureSaves = cellstr(char(featureSaves));
    labels2 = cellstr(int2str(ids2));
    labels2 = strcat(labels2, featureSaves);

    % nubrezia raudona langa
    frame = insertObjectAnnotation(frame, 'rectangle', ...
        bboxes2, labels2, 'Color', 'red');

    % Nubrezia raudona langa kaukes vaizde
    mask = insertObjectAnnotation(mask, 'rectangle', ...
        bboxes2, labels2, 'Color', 'red');
end

```



```

end

% Atvaizduoja dinamines kaukes vaizda arba fono substrakcija
obj.maskPlayer.step(mask);
obj.videoPlayer.step(frame);
end

%% funkcija asmens identifikavimas
function identifikuotiAsmeni()

for i=1:length(tracks)
if tracks(i).busena == 'N' && tracks(i).featureCount > dataThreshold
if counter1<3 % pirmos 3 grupes bus saugojamos is karto be klasifikavimo
pradines_grupes(i);
counter1=counter1+1;
else
%Sitos dvi eilutes skirtos is viso masyvo isrinkti tik i
%asmens savybes
ind_naujos_g=(duomenysIDTemp==tracks(i).id);
testFeature= duomenysTemp(ind_naujos_g,:);

% prognozuojamas asmens judejimas pritaikant klasifikatoriui
[imagepred, ~] = predict(trainedClassifier,testFeature);

result= mode (imagepred); %isrenkama dzniausiai priskiria grupe

if ~(result == 1) %vadinasi sutapo su kazkokia grupe, nes 0 grupe reiskia naujas asmuo
tracks(i).busena= 'A';
tracks(i).id= result; %naujam track, priskiriama esamo ID
else
tracks(i).busena= 'T';
end

%duomenu baze papildoma naujo asmens feature taskais
ind_naujos_g=(duomenysIDTemp==tracks(i).id);
indeksai(1:sum(ind_naujos_g),1)= tracks(i).id;
duomenys = [duomenys;duomenysTemp(ind_naujos_g,:)];
duomenysID =[duomenysID; indeksai];

Apmokymas();
end
end
end
end

%% Papildoma funkcija zymeti ir identifikuoti asmenis
function papildyti_dataset()

for i = 1:length(tracks)
tracks(i).featureSave = 'n'; %rodo, kad sitas frame no save

```

```

if (tracks(i).busena=='N')

    bboxes = cat(1, tracks(i).bbox);
    iskirptas_asmuo = imcrop(frame,bboxes);

    %isskiriu kiekvieno track feature points
    if size(iskirptas_asmuo,1)> min_person_height &&
size(iskirptas_asmuo,2)>min_person_width
        feature_set=extractFeatures3(iskirptas_asmuo, ClasiPerHeight, ClasiPerWidth);
%ClasiPerHeight, ClasiPerWidth sitie privalo buti koki pat, kaip ir buvo naudota apmokant
clasifier

        duomenysTemp=[duomenysTemp;feature_set];
        duomenysIDTemp=[duomenysIDTemp; tracks(i).id];

        %padidinu savybiu kieki
        tracks(i).featureCount=tracks(i).featureCount+1;
        tracks(i).featureSave = 'y'; %rodo, kad sitas frame issaugotas kaip asmens features
    end
end
end
end

%% Programos pradzios funkcija. pirmu grupiu kurimas
function pradines_grupes( index )
    tracks(index).busena= 'I'; %identifikuotas - tai naujas zmogus ir jam nauja grupe
    %duomenu baze papildoma naujo asmens feature taskais
    ind_naujos_g=(duomenysIDTemp==tracks(index).id);
    indeksai(1:sum(ind_naujos_g),1)= tracks(index).id;
    duomenys = [duomenys;duomenysTemp(ind_naujos_g,:)];
    duomenysID =[duomenysID; indeksai];
    if counter1==2 %jeigu 2 jau trecias asmuo
        Apmokymas();
    end
end

%% Programos pradzios funkcija. sukura 1 grupes nari, is nuotrauku be zmoniu
function programos_pradzia()
    for i=1:1:20
        frame = readFrame();

        feature_set=extractFeatures3(frame, ClasiPerHeight, ClasiPerWidth);
%ClasiPerHeight, ClasiPerWidth sitie privalo buti koki pat, kaip ir buvo naudota apmokant
clasifier
        duomenys=[duomenys;feature_set];
        duomenysID=[duomenysID; 1];
    end
end

%% apmoko nauja clasifier
function Apmokymas()

```

```
featuresVector= duomenys;  
featuresVector(:,length(duomenys)+1)= duomenysID;  
[trainedClassifier, ~] = trainClassifier_cubicSVM2(featuresVector); %features Vector  
paskutinis stulpelis privalo tureti ID asmenu  
end  
%%  
displayEndOfDemoMessage(mfilename)  
end
```

Priedas 2. *MATLAB* programa asmenų klasifikatorius

```
%Programa asmenu klasifikatorius
% 1 etapas paimama duomenu baze
% 2 etapas is kiekvieno kadro sukuriamos savybes
% 3 etapas sukuriama duomenu lentele, kuri vadinasi DATA
% 4 etapas vyksta apmokymas
%% 1 etapas Paimami duomenys is duomenu bazes
imset = imageSet('trainingImages','recursive');

%% 2 etapas sukuriamos kiekvieno asmens savybes

numPer=length(imset); % Asmenu skaicius duomenu bazeje
n=1;
for i = 1:numPer
    numImg= imset(1,i).Count; % numeris nuotrauku kategorijoje
    for ii = 1:numImg
        %paima viena nuotrauka is data set
        path = imset(1,i).ImageLocation{ 1,ii}; %paima lokacija is data set
        image=imread(path); %nuskaito nuotrauka is path lokacijos
%istraukia is nuotraukos features
        featuresVector(n,:) = extractFeatures4( image, 380, 150); % input image and [height withd]
        n=n+1;
    end
end

%% 3 etapas Sukuriama lentele iskirtoms savybems
Data      = array2table(featuresVector);
Data.Type  = getImageLabels(imset);

%% 4 etapas Naudojamas klasifikatoriaus apmokymas
classificationLerner
```

Priedas 3. *MATLAB* funkcija savybėms išskirti

%% Funkcija savybiu isskyrimas

```
function [ featuresVector ] = extractFeatures4( image, height, width )  
image2 = imresize(image, [height width]);
```

%% skaiciuoja ivairius savybiu isskyrimo budus

%cellsize nustato kiek savybiu turesime

```
HOG = extractHOGFeatures(image2,'CellSize',[64 64]);
```

%cellsize nustato kiek savybiu turesime

```
LBP = extractLBPFeatures(rgb2gray(image2),'CellSize',[100 100]);
```

% isskiriamos kadre esancio objekto savybes

```
hsvHist = hsvHistogram(image2);
```

```
autoCorrelogram = colorAutoCorrelogram(image2);
```

```
color_moments = colorMoments(image2);
```

% for gabor filtrui reikia greiaus skales spalvu

```
img = double(rgb2gray(image2))/255;
```

```
[meanAmplitude, msEnergy] = gaborWavelet(img, 4, 6); % 4 = skaliu nr, 6 = orientaciju nr
```

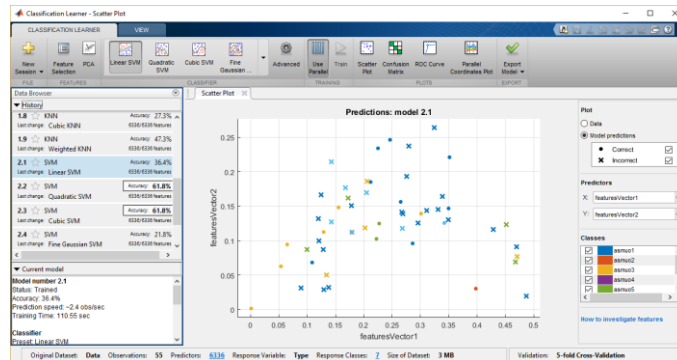
```
wavelet_moments = waveletTransform(image2);
```

% Visas savybes sudeda i viena masyva

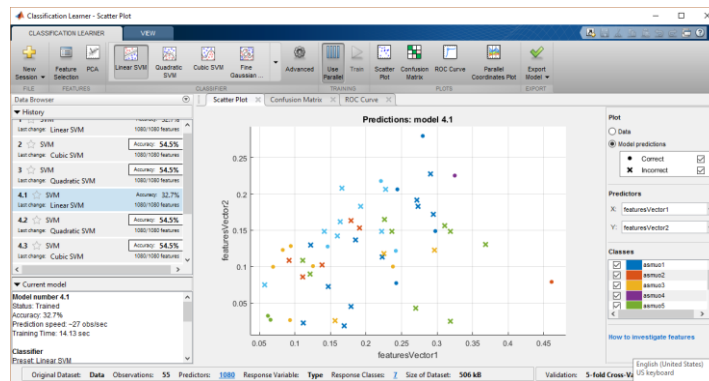
```
featuresVector = [ autoCorrelogram color_moments wavelet_moments ];
```

Priedas 4. *MATLAB* mašininio apmokymo *SVM* klasifikatorių tikslumai

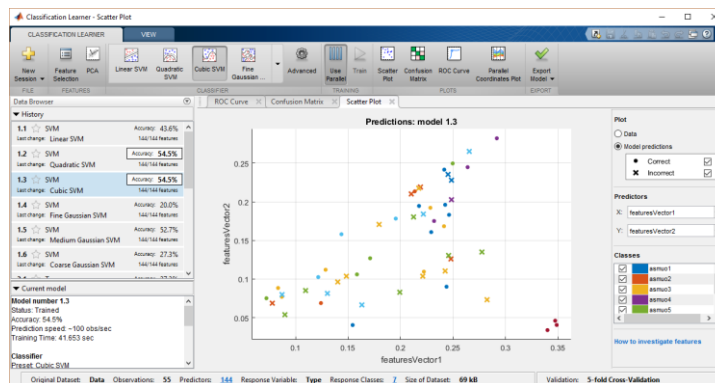
SVM klasifikatorius su *HOG*, kai taikoma nustatyta 16x16 celių



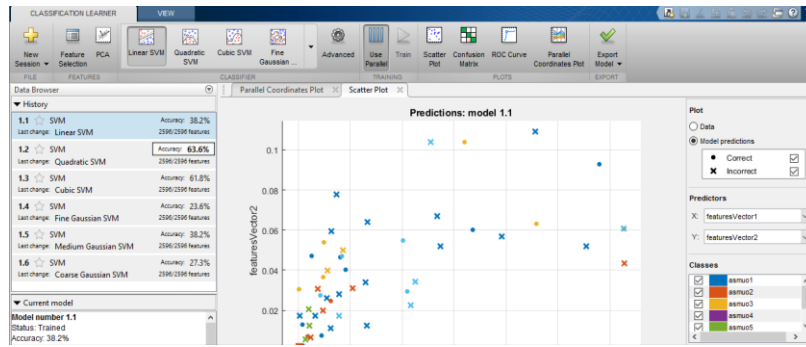
SVM klasifikatorius su *HOG*, kai nustatyta 32x32 celių



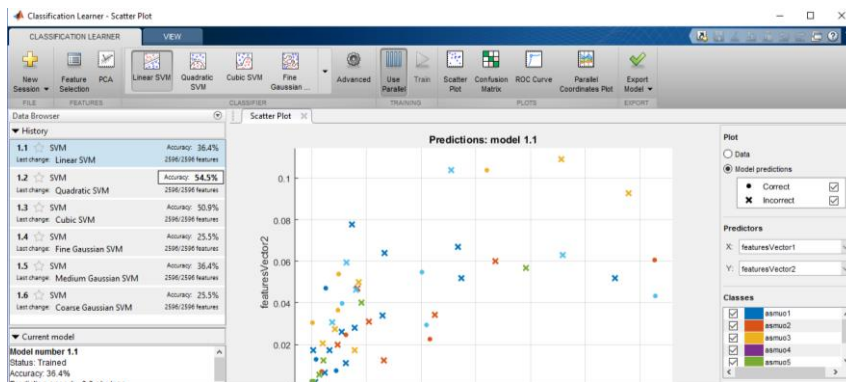
SVM klasifikatorius su *HOG*, kai nustatyta 64x64 celių



SVM klasifikatorius su *LBP*, kai nustatyta 16x16 celių

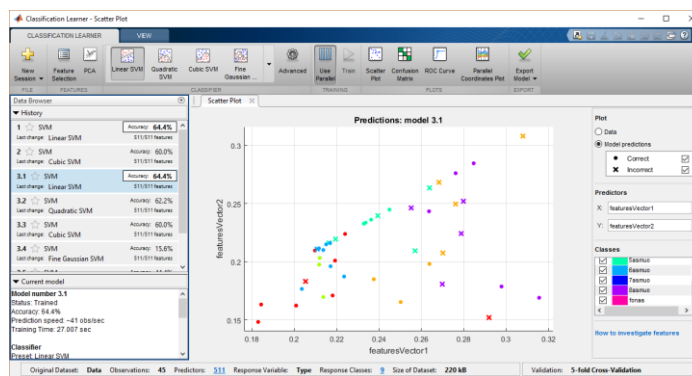


SVM klasifikatorius su LBP, kai nustatyta 32x32 celių

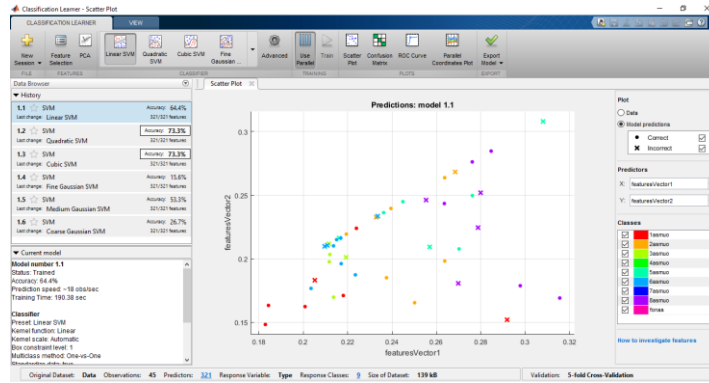


Toliau pateikti SVM klasifikatoriaus tikslumai, su daugiau asmenų išskyrimo savybių tipais

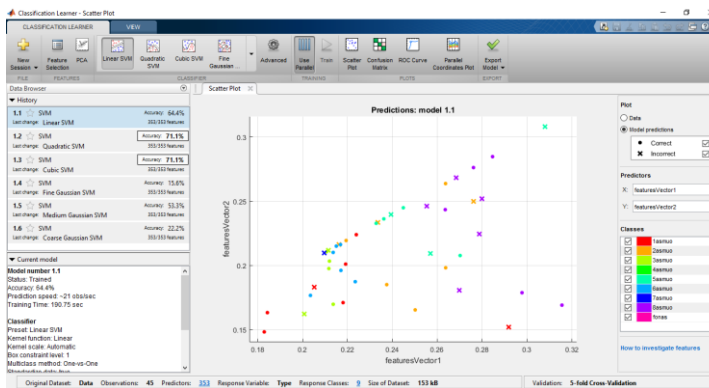
Visos savybės



HOG, LBP

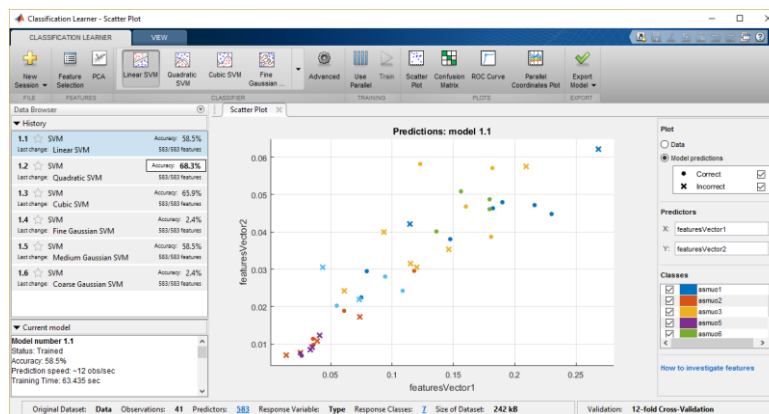


HOG, LBP, HSV Histogram, ColorautoCorrelogram

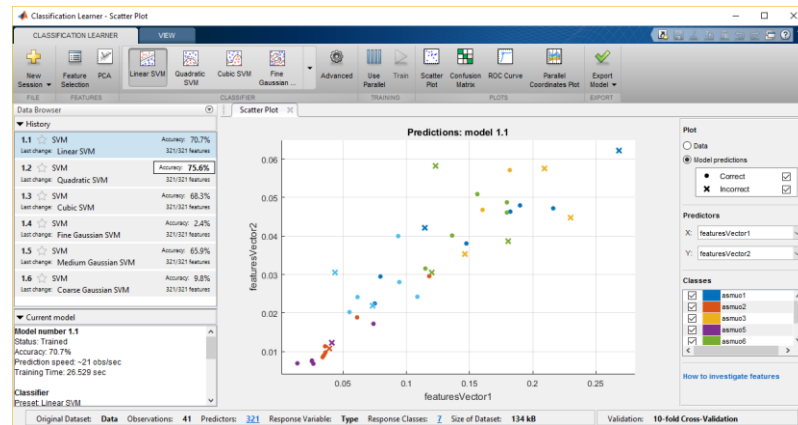


Toliau pateikti SVM klasifikatoriaus tikslumai su daugiau asmenų išskirimo savybių tipais, kai atmestos dvi asmenų savybių grupės.

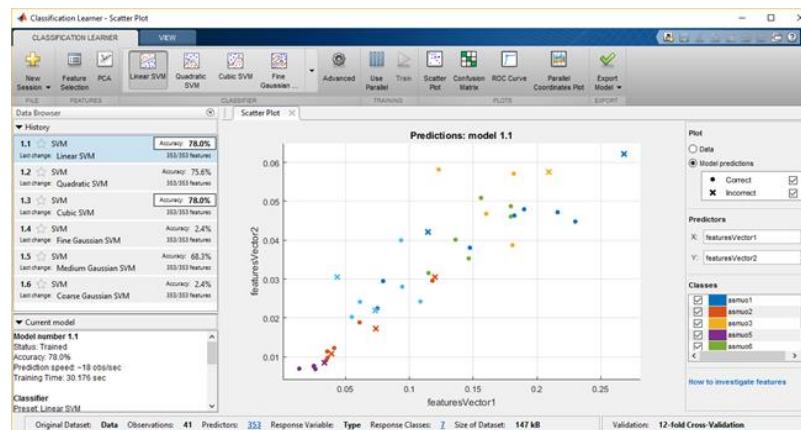
SVM klasifikatorius su visais išskirimo savybių tipais



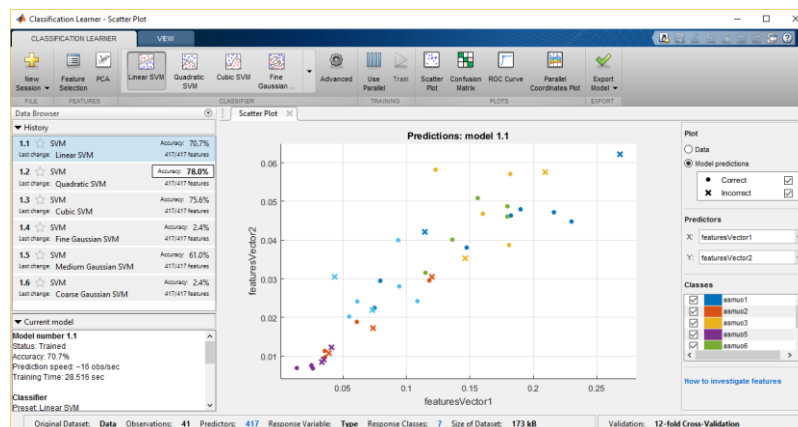
LPG, HOG kartu panaudojus nustačius 16x16 celių



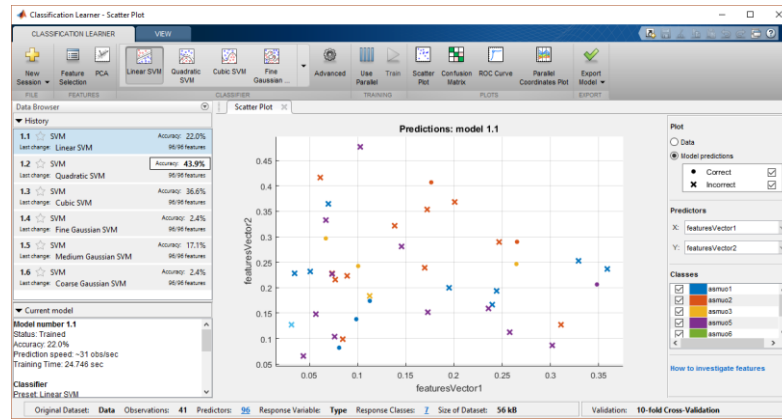
LPG, HOG su HSV Histogram



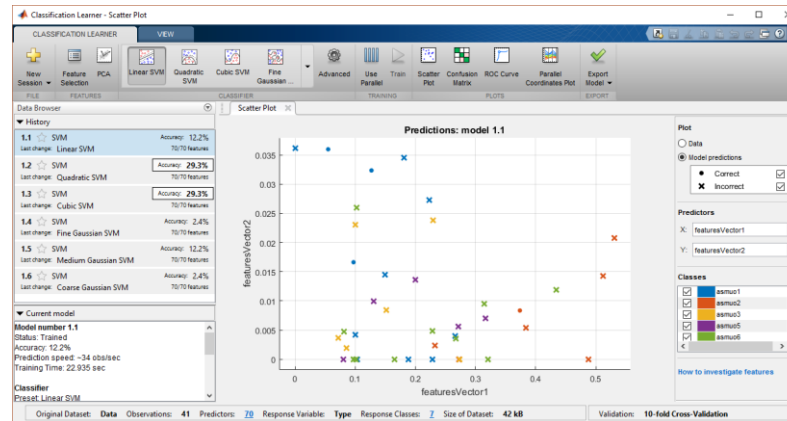
LPG, HOG, HSV Histogram, ColorautoCorrelogram



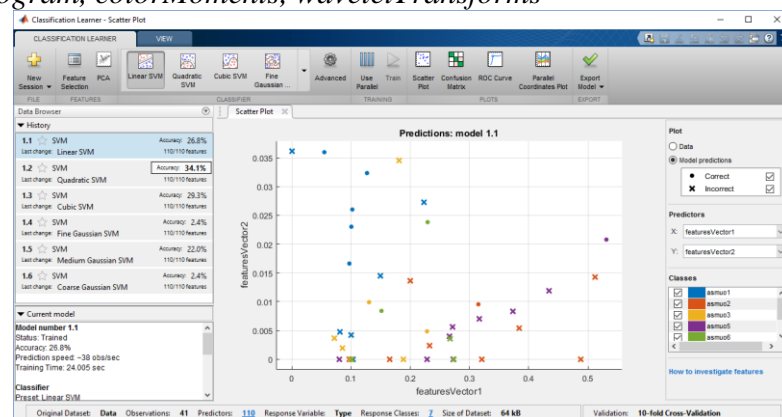
HSV Histogram, ColorautoCorrelogram



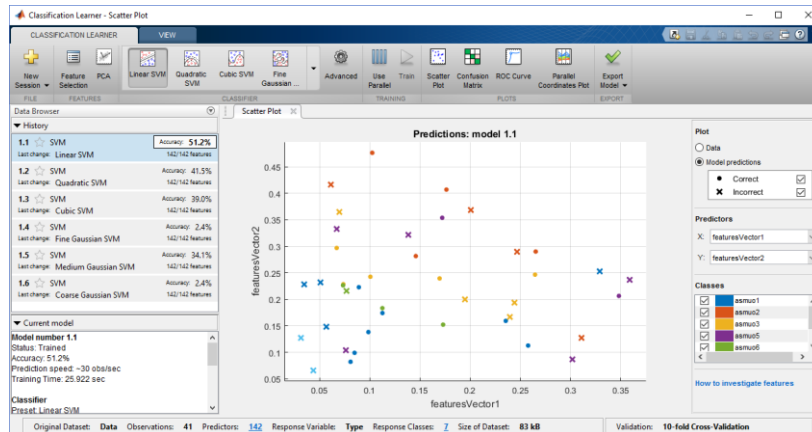
ColorautoCorrelogram, colorMoments



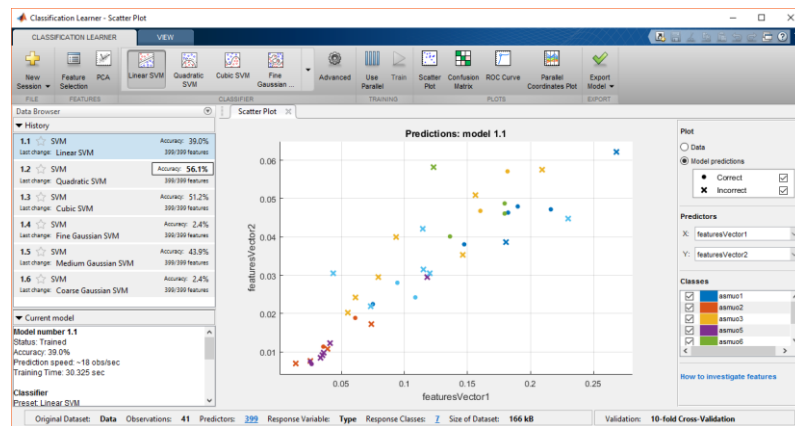
ColorautoCorrelogram, colorMoments, waveletTransforms



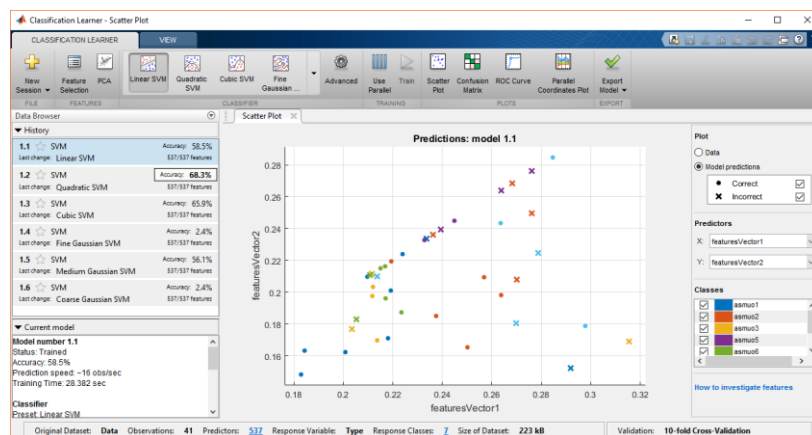
HSV Histogram, ColorautoCorrelogram, colorMoments, waveletTransform



LBP, HSV Histogram, autoCorrelogram, colorMoments, meanAmplitude, msEnergy

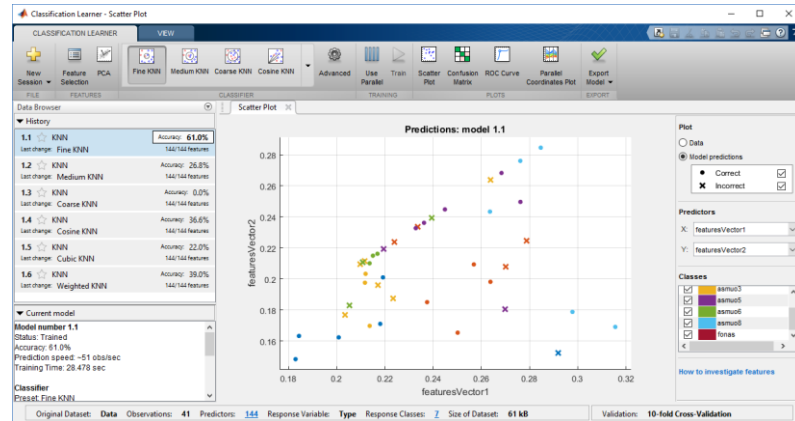


LBP, HOG, HSV Histogram, ColorautoCorrelogram, meanAmplitude, msEnergy

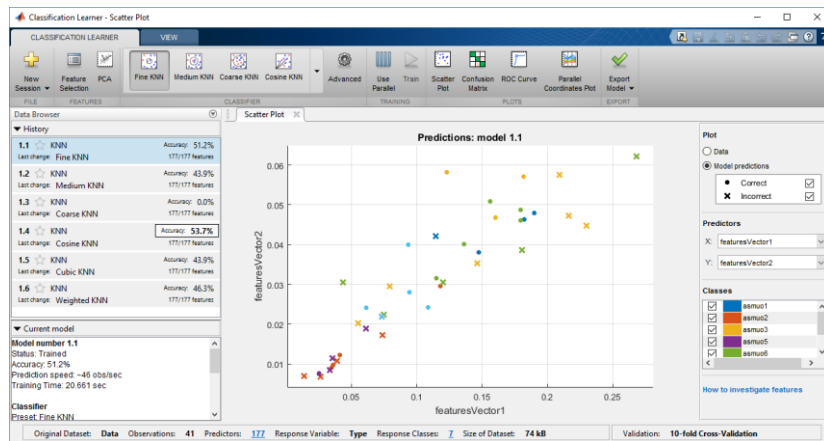


MATLAB mašininio apmokymo *KNN* klasifikatorių tikslumai

HOG išskyrimo savybės (pašalinus dvejų asmenų savybių grupes) 16x16 celių



LBP išskyrimo savybės (pašalinus dvejų asmenų savybių grupes) 16x16 celių



HOG, LBP išskyrimo savybės 16x16 celių

