

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ INŽINERIJOS KATEDRA

Kęstutis Murauskas

## **Judančių vaizdo objektų sekimas ir analizė**

Magistro darbas

Darbo vadovas

doc. dr. Eimutis Karčiauskas

Kaunas, 2009

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMŲ INŽINERIJOS KATEDRA

Kęstutis Murauskas

## **Judančių vaizdo objektų sekimas ir analizė**

Magistro darbas

Recenzentas

doc. dr. Eugenijus Mačikėnas

2009-05

Vadovas

doc. dr. Eimutis Karčiauskas

2009-05

Atliko

IFM-3/2 gr. studentas  
Kęstutis Murauskas

2009-05

Kaunas, 2009

# TURINYS

1. ĮVADAS .....	9
2. KOMPIUTERINIŲ VAIZDŲ APDOROJIMO ANALIZĖ .....	11
2.1. Tikslas .....	11
2.2. Problematika .....	11
2.3. Kompiuterinis vaizdas .....	11
2.4. Problemos kylančios apdorojant vaizdus .....	12
2.5. Judančio vaizdo sekimo apibrėžimas .....	13
2.5.1. Taškų sekimas .....	13
2.5.2. Branduolio pagrindo sekimas (angl. Kernel Tracking) .....	15
2.5.3. Siluetu paremtas sekimas .....	15
2.6. Fonų atimtis ir segmentavimas .....	16
2.6.1. Segmentavimo apibrėžimas .....	16
2.6.2. Fonų atimties apibrėžimas .....	16
2.6.3. Segmentavimo algoritmai .....	17
2.7. Kontūro aproksimacija .....	22
2.8. 3D vaizdo analizė 2D vaizde .....	23
2.9. Judančio vaizdo objekto aptikimas .....	23
2.9.1. Objekto aptikimas naudojant filtrus .....	23
2.9.2. Objekto aptikimas pagal šabloną .....	24
2.9.3. Objekto aptikimas pagal histogramą .....	25
2.10. Išvados ir priimtas sprendimas .....	27
2.10.1. Priimto sprendimo tikslai .....	27
3. PROJEKTINĖ DALIS .....	28
3.1. Tikslas .....	28
3.2. Bendra apžvalga .....	28
3.3. Pasirinktos technologijos .....	28
3.3.1. OpenCV vaizdo apdorojimo biblioteka .....	28
3.3.2. EMGU .....	28
3.4. Statinis sistemos vaizdas .....	30
3.5. Panaudojimo atvejai .....	30
3.6. Stebėjimo proceso apibendrinta sekos diagrama .....	32

3.7. Judančio vaizdo objekto stebėjimo ir analizės algoritmo aprašymas .....	33
3.8. Išvados .....	33
4. JUDANČIŲ VAIZDO OBJEKTŲ SEKIMO IR ANALIZĖS PROGRAMOS KOKYBĖS TYRIMAS .....	34
4.1. Tikslas .....	34
4.2. Vartotojo sąsaja.....	34
4.3. Sistemos testavimas .....	34
4.4. Sistemos saugumas .....	34
4.5. Išvados .....	34
5. JUDANČIŲ VAIZDO OBJEKTŲ SEKIMO IR ANALIZĖS TESTAVIMO EKSPERIMENTAI.....	35
5.1. Tikslas .....	35
5.2. Eksperimento objektas .....	35
5.3. Eksperimentas .....	35
5.3.1. Eksperimentų aprašymas .....	35
5.3.2. Eksperimentų rezultatai .....	36
5.4. Išvados .....	38
6. IŠVADOS .....	39
7. LITERATŪRA .....	40
8. TERMINŲ IR SANTRAUKŲ ŽODYNAS .....	42
PRIEDAI.....	43

Murauskas K. Judančių vaizdo objektų sekimas ir analizė: Programų inžinerijos magistro darbas / vadovas doc. dr. E. Karčiauskas; Kauno technologijos universitetas, Informatikos fakultetas, Programų inžinerijos katedra. – Kaunas, 2009. – 51 p.

## **SANTRAUKA**

Sporto varžybų įrašai yra intensyviai studijuojami ir šių įrašų analizė susilaukia vis daugiau dėmesio dėl potencialios komercinės naudos ir pramogų funkcinės galimybės. Didėjantis vaizdo apdorojimo aplikacijų poreikis, skatina mokslininkus atlikti eksperimentus įvairiais vaizdo analizės aspektais. Pagrindinės sporto vaizdo įrašų tyrimo sritys yra: metimų charakteristikų tyrimas, objektų sekimas, ryškiausių įvykių išgavimas.

Daugiausia tyrimų yra atliekama su populiarių sporto šakų vaizdo įrašais: krepšinio, tinklinio, futbolo. Remdamasis šių tyrimų rezultatais, vaizdo objektų sekimą ir analizę bus bandoma pritaikyti nagrinėti mėtomo rutulio charakteristikoms.

Šio darbo tikslas – sukurti programinę įrangą judančių vaizdo objektų sekimui ir analizei, atlikti šios programinės įrangos analizę.

Projektinėje dalyje, naudojantis pasirinktais metodais, kurie apžvelgiami analizės dalyje, ir panaudojant atvirojo kodo vaizdų apdorojimo biblioteką „OpenCV“ yra pateikiamas realizuotos programinės įrangos pavyzdys. Naudojantis šia sukurta programa yra atliekami eksperimentiniai tyrimai, kuriais remiantis bus daromos išvados dėl sukurto metodo veikimo teisingumo, jo teigiamų ir neigiamų savybių.

Murauskas K. Moving video object tracking and analysis: Master's Work in Software Engineering / supervisor doc. dr. E. Karčiauskas; Department of Software Engineering, Faculty of Informatics, Kaunas University of Technology. – Kaunas, 2009. – 51 p.

## SUMMARY

Sports videos, as important multimedia contents, have been extensively studied, and sports video analysis is receiving more and more attention due to the potential commercial benefits and entertaining functionalities. The urgent requirements for multimedia applications therefore motivate the researches in various aspects of video analysis. Major research issues of sports video analysis include: shot characteristic analysis, highlight extraction and object tracking.

Most researches are carried out with the popular sports videos: basketball, volleyball, football. Based on these studies, the video object tracking and analysis is adapted to do research on shot-put.

The purpose of this work is to create „Moving video object tracking and analysis“ software and perform analysis based on this software.

The design part of the document will show the solution of the software, which is based on methods described in analysis part and open source „OpenCV“ vision library. Experiments are done using „Moving video object tracking and analysis“ software and conclusion findings will be given about methods correctness, positive and negative sides of work.

**Keywords:** Video object tracking, video segmentation, contour extraction, background subtraction, computer vision.

## PAVEIKSLAI

1 pav. Automobilio vaizdo matymas žmogaus akimis ir kompiuteriu [11].....	12
2 pav. 2D objekto atvaizdavimo pasikeitimas nuo žiūrėjimo kampo [11] .....	12
3 pav. objekto taškų atitikimas .....	13
4 pav. Objekto taškų atitikimas: (a) Visi galimos kadro t-1 objekto taškų asociacijos su kadro taškais; (b) unikalių asociacijų rinkinys; (c) kelių kadro taškų asociacijos. ....	14
5 pav. Skirtingi judesio apribojimai: (a) Artumas (b) Maksimalus kitimas (c) Mažas kitimas (d) Bendras judėjimas (e) Griežtumas .....	14
6 pav. Parametruota kvadrato transformacija .....	15
7 pav. Kontūrų kitimas.....	15
8 pav. FDI segmentavimas [5].....	21
9 pav. PFDI segmentavimas [5].....	22
10 pav. Kontūro aproksimacija .....	22
11 pav. Kontūro apibrėžimas geometrinės figūromis .....	23
12 pav. Histogramų metodų palyginimų reikšmės .....	26
13 pav. EMGU architektūra.....	29
14 pav. Statinis sistemos vaizdas .....	30
15 pav. Panaudojimo atvejai .....	30
16 pav. Stebėjimo proceso sekos diagrama .....	32
17 pav. Judančio vaizdo objekto sekimo ir analizės algoritmo žingsniai .....	33
18 pav. Rutulio metimo trajektorijos analizė.....	37

## **LENTELĖS**

1 lentelė. Vaizdo įrašų charakteristikos .....	35
2 lentelė. Vaizdo įrašų analizės trukmė .....	36
3 lentelė. Rutulio aptikimo kiekis .....	37



## 1. ĮVADAS

### **Darbo tema**

Šiame darbe yra nagrinėjama judančių vaizdo objektų išskyrimo problema ir judančio objekto terminu yra apibūdinamas rutulys.

### **Aktualumas**

Vaizdų apdorojimas – duomenų apdorojimo sistemos panaudojimas vaizdams sudaryti, skaityti, analizuoti, suvokti, gerinti, atvaizduoti. Vaizdų apdorojimo populiarumas šiomis dienomis didėja. Populiarumui įtakos turėjo didelis vaizdo įrašų ir transliacijų poreikis pramogoms, aplikacijoms, kurios skirtos saugumui užtikrinti, mokymuisi, telemedicinai.

Objektų sekimas yra plačiai naudojamas:

- Judesio pobūdžio paremti objekto sekimai – pavyzdžiui, žmogaus identifikavimas pagal eiseną;
- Automatizuotos sistemos – stebi ir ieško neįprastų įvykių;
- Vaizdo indeksavimas – automatinis komentavimas ir vaizdų gavimas iš duomenų bazės;
- Kompiuterio ir žmogaus bendravime – gestų atpažinimas;
- Automobilių srauto stebėjimas – sudaroma automobilių srautų statistika;
- Transporto priemonių navigacijai – vaizdu paremto kelio sudarymas ir kliūčių išvengimas.

Sporto vaizdo įrašų apdorojimo tematika yra įdomus dalykas tiriamajam darbui, nes aiškiai apibrėžtos žaidimo taisyklės suteikia gausią žinių sritį analizei [19].

Šiame darbe nagrinėjamas vaizdų apdorojimo pritaikymas sporto aplinkoje. Vienu atveju tai gali būti skirta sporto rungtynių žiūrovui, pateikiant jam tam tikrą informaciją, kitu atveju tai gali būti panaudota analizei, kuri skirta pagerinti sportininko rezultatus.

### **Darbo tikslas.**

Išanalizuoti egzistuojančius vaizdo objekto sekimo ir analizės sprendimus, kurie naudojami sporto vaizdo įrašų apdorojimui ir sukurti eksperimentinę sistemą, kuri atliktų mėtomo rutulio trajektorijos sekimą ir analizę.

## **Darbo metodika.**

Darbą sudaro įžanga, keturi skyriai: *2 analitinė dalis, 3 projektinė dalis, 4 Tyrimo dalis, 5 eksperimentinė dalis*; išvados, naudotos literatūros sąrašas, terminų ir santraukų žodynas, priedai.

Darbo apimtis 51 puslapis, jame yra 3 lentelės, 18 paveikslų, literatūros sąrašą sudaro 20 šaltinių.

### **Darbo planas:**

1. mokslinės literatūros analizė;
2. programinės įrangos sukūrimas;
3. programinės įrangos kokybės įvertinimas;
4. eksperimentas;
5. rezultatų pateikimas ir įvertinimas.

Pirmame etape analizuojama mokslinė literatūra, apžvelgiami egzistuojantys sprendimai, naudojami metodai, inžinerinės problemos, jų sprendimo būdų analizė.

Antrame etape, pagal atliktą apžvalgą yra siūlomas vaizdo apdorojimo modelis, kuriama jo architektūra.

Trečiame etape pateikiamas sukurtos programinės įrangos kokybės analizė. Joje tyrinėjamos programų sistemos kokybės tobulinimo galimybės.

Ketvirtame etape pateikiamas sukurtos programinės įrangos bei jos patobulinimų eksperimentinis tyrimas.

Penktame etape itin glaustai išdėstoma nuomonė apie svarbiausius tezėse paliestus klausimus: projektavimo metu priimti sprendimai, atliktų tyrimų esmė ir gauti rezultatai.

## **2. KOMPIUTERINIŲ VAIZDŲ APDOROJIMO ANALIZĖ**

### **2.1. Tikslas**

Pagrindinis šios darbo dalies tikslas – išanalizuoti vaizdo apdorojimo būdus, objektų segmentavimo, galinio fono sudarymo, objektų atpažinimo metodus, išryškinti esmines vaizdų apdorojimo problemas, rasti ir apžvelgti iškeltų problemų sprendimo būdus.

### **2.2. Problematika**

Egzistuoja daugybę sprendimų skirtų sekti judančius vaizdo objektus, tačiau kiekvienas iš jų yra pritaikytas tam tikriems atvejams. Dažniausiai eksperimentai yra atliekami su populiariomis sporto šakomis: krepšinis, tinklinis, futbolas, beisbolas.

Norėdami analizuoti sportininko mėtomą rutulį susiduriame su rutulio sekimo vaizde problema, kadangi nėra universalus metodo, kuris tiktų judančio vaizdo objekto aptikimui ir leistų atlikti to objekto analizę. Remiantis kitoms sporto šakoms sukurtais egzistuojančiais sprendimais, ieškosime metodologijos, kuri leistų įgyvendinti mėtamo rutulio aptikimą ir analizę.

### **2.3. Kompiuterinis vaizdas**

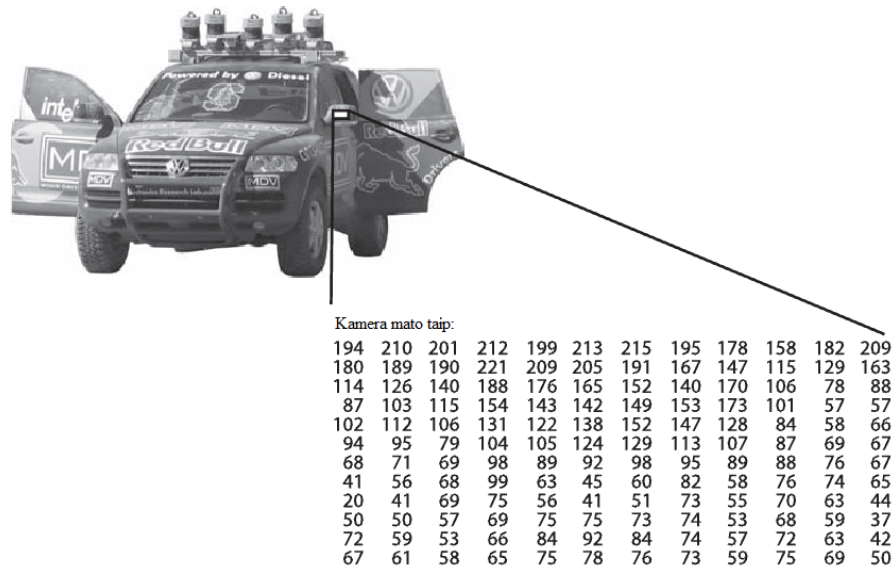
Remiantis šaltiniu [11] apibrėšime kompiuterinio vaizdo sąvoką.

Kompiuterinis vaizdas yra vaizdo šaltinio duomenų transformacija į sprendimą ar naują atvaizdavimą. Visos šios transformacijos yra atliekamos tam, kad pasiektume užsibrėžtą tikslą. Įeinantys duomenys gali turėti kontekstinės informacijos. Sprendimas gali būti apibrėžtas taip: ar vaizdo kadre yra rutulys arba kiek rutulių yra kadre. Nauju sprendimu gali tapti spalvoto paveikslu pavertimas nespalvotu arba kameros judesio pašalinimas iš vaizdo kadru sekos.

Žmogaus smegenys suskaido vaizdo signalą į daugybę kanalų, kurie perduoda skirtingą informaciją jūsu smegenims. Smegenys susifokusuoja ties svarbiomis vaizdo dalimis, kad galėtų atlikti jų analizę, kai tuo tarpu kitų vaizdo dalių analizė yra sustabdoma. Smegenys kuria vaizdus atsižvelgdamos į asociacijas, kurios atsirado gyvenant pasaulyje.

Mašininės vaizdo sistemos atveju, kompiuteris gauna skaičių matricą iš kameros ar kietojo disko. Didžiausiai daliai nėra jokių šablonų atpažinimui, fokusavimo ir apertūros automatinio valdymo. 1 pav. matome automobilį. Šiame paveiksle mes matome šoninį automobilio veidrodėlį, o kompiuteris mato tik skaičių matricą. Kiekvienas skaičius iš

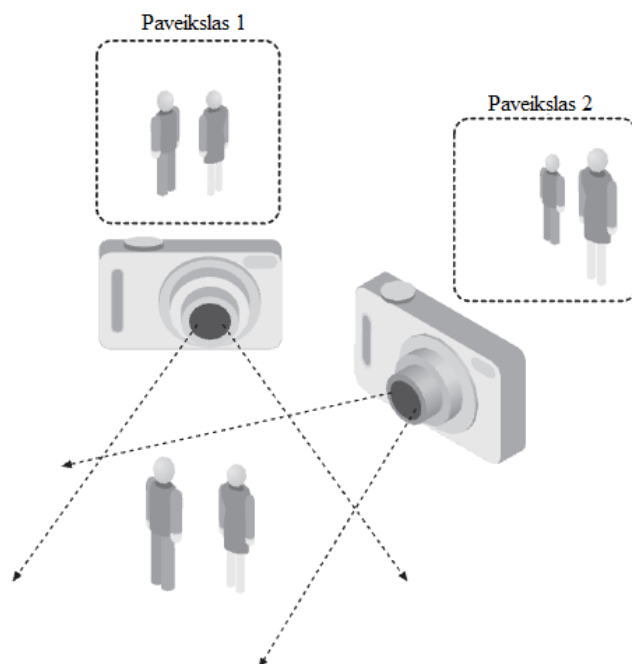
matricos perduoda daugiau triukšmo elementų ir tik truputėlį naudingos informacijos, bet kompiuteris mato tik šią skaičių matricą, todėl užduotimi tampa šį triukšmų pilną skaičių matricą paversti į žmogaus akį suprantamą vaizdą – „šoninis veidrodis“.



1 pav. Automobilio vaizdo matymas žmogaus akimis ir kompiuteriu [11]

## 2.4. Problemos kylančios apdorojant vaizdus

Jeigu turime trimačio vaizdo (3D) dvimatį vaizdą (2D), tai nėra unikalaus būdo kaip atkurti 3D signalą. Ši problema neturi nei unikalaus, nei geriausio sprendimo. Net jei turėtume idealius duomenis, tai tas pats 2D vaizdas gali apibūdinti daugybę 3D vaizdų. 2 pav. matome, kaip tas pats vaizdas skiriasi atvaizdavimu 2D vaizde, kai keičiasi žiūrėjimo kampas.



2 pav. 2D objekto atvaizdavimo pasikeitimas nuo žiūrėjimo kampo [11]

Dauguma duomenų yra sugadinti triukšmų ir iškraipymų. Šie sugadinimai atsiranda dėl įvairių pasaulio kitimų: judėjimas, žaibavimas, oro sąlygos, atspindžiai; defektų lęšiuose ar mechaniniame montavime, sensoriaus riboto vaizdo sujungimo į visumą laiko (vaizdo suliejimo atvejis), elektrinių signalų triukšmų sensoriuje ar kitoje elektronikoje, vaizdo glaudinimo. Taip pat objektą sudėtinga gali būti sekti dėl:

- sudėtingo objekto judesio;
- neaiškios objekto prigimties;
- dalinio ar pilno objektų persidengimo;
- sudėtingos objektų formos.

## 2.5. Judančio vaizdo sekimo apibrėžimas

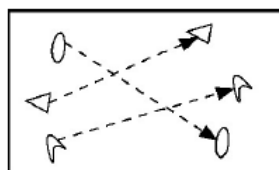
Objekto sekimo tikslas yra sukurti objekto judėjimo trajektoriją per laiką, nustatant objekto koordinates kiekviename vaizdo kadre. Objektų sekimas ir atitikimas tarp egzempliorių gali būti atliktas atskirai arba kartu. Kai sekimas ir atitikimas yra atliekamas atskirai, tai visų pirma per visus vaizdo kadrus, sekimo algoritmas pagal objekto srities vidurkį aptinka objektus ir vėliau per visus kadrus ieško objekto atitikimų. Kai sekimas ir atitikimas yra atliekamas kartu, tai objekto srities atitikimai yra atliekami iteraciniu būdu, atnaujinant objekto srities ir vietos informaciją pagal prieš tai buvusį kadrą.

Objekto sekimas gali būti atliekamas pagal:

- Spalvą
- Kraštus
- Tekstūrą
- Optinius parametrus (pvz. ryškumas)

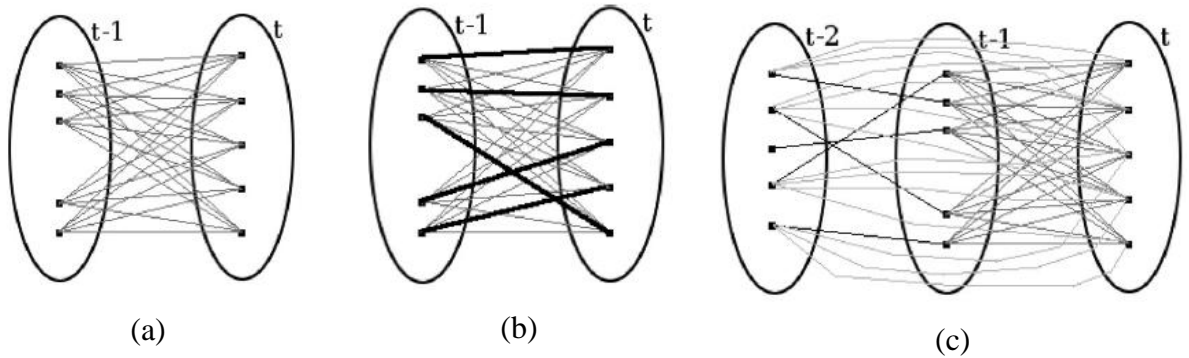
### 2.5.1. Taškų sekimas

Šiuo atveju vienas po kito einančiuose kadruose aptiktas objektas yra apibūdinamas taškais ir sąryšis tarp taškų yra paremtas pagal prieš tai buvusią objekto būklę, kuri gali įtraukti objekto poziciją ar judėjimą. Toks sekimas reikalauja mechanizmo aptikti objektus kiekviename kadre. Objektų atitikimo pavyzdys pateikiamas 3 pav.



3 pav. objekto taškų atitikimas

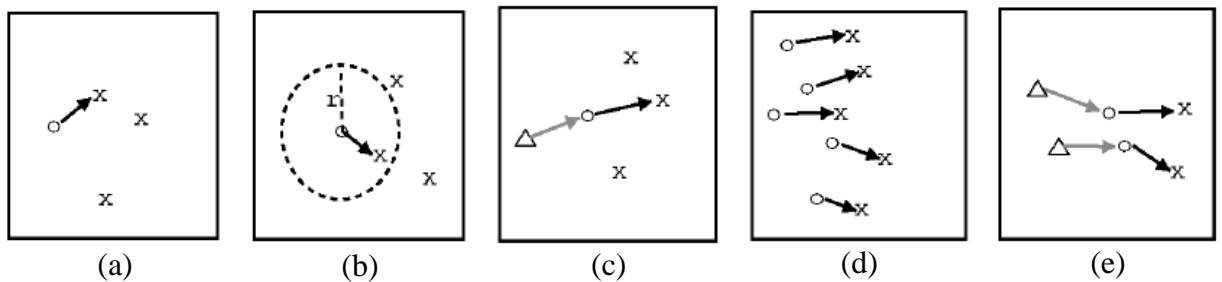
Taškų atitikimas tarp kadru tampa sudėtinga problema kai atsiranda objektų persidengimų, objekto neaptikimų, objektų atsiradimo ir išnykimo kadre atveju. Šis metodas kiekvieną objektą kadre  $t-1$  asocijuoja su objektu kadre  $t$ , naudodamas rinkinį judesio apribojimų. Taškų atitikimų mažinimas yra formuluojamas kaip kombinatorikos optimizavimo problema. 4 pav. pateikiami taškų atitikimas tarp kadru  $t-1$  ir  $t$ . [14]



4 pav. Objekto taškų atitikimas: (a) Visi galimos kadro  $t-1$  objekto taškų asociacijos su kadro taškais; (b) unikalių asociacijų rinkinys; (c) kelių kadru taškų asociacijos.

Atitikimų sudėtingumas dažniausiai yra įvertinamas pagal apribojimus (5 pav.):

- Artumas – laikoma, kad objektų padėtis žymiai nesikeičia tarp kadru;
- Maksimalus kitimas – apriboja taškų atitikimus aplink objektą;
- Mažas kitimas – laikoma, kad objekto kryptis ir greitis drastiškai nesikeičia;
- Bendras judėjimas – kaimyniniai objektai yra laikomi panašūs;
- Griežtumas (angl. rigidity) – tam tikro objekto atstumas tarp dviejų taškų nesikeičia.

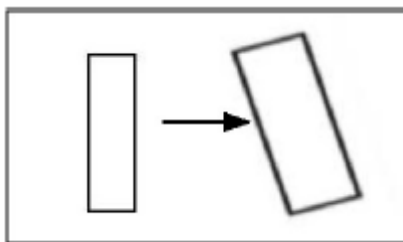


5 pav. Skirtingi judesio apribojimai: (a) Artumas (b) Maksimalus kitimas (c) Mažas kitimas (d) Bendras judėjimas (e) Griežtumas

Taškinis sekimas yra naudingas mažų objektų sekimui, kurie atvaizduojami vienu tašku. Didesnių objektų sekimui yra reikalingi keletas taškų. Kai objekto sekimui yra naudojamas keletas taškų tai susiduriame su taškų, kurie priklauso tam pačiam objektui, automatinio grupavimo problema. Taip atsitinka todėl, kad reikia atskirti skirtingus objektus, atskirti patį objektą nuo galinio fono.

### 2.5.2. Branduolio pagrindo sekimas (angl. Kernel Tracking)

Branduolys – yra objekto forma ir išvaizda. Branduoliu gali būti laikoma paprasčiausios figūros. Dažniausia tai yra kvadratas arba elipsės formos figūra su jai priskirta histograma [2, 14]. Objektų sekimas yra atliekamas apskaičiuojant objektų judėjimą pagal branduolį vienas po kito einančiame kadre. Šis judėjimas dažniausiai yra parametrinė transformacija – keitimas, pasukimas (6 pav.).

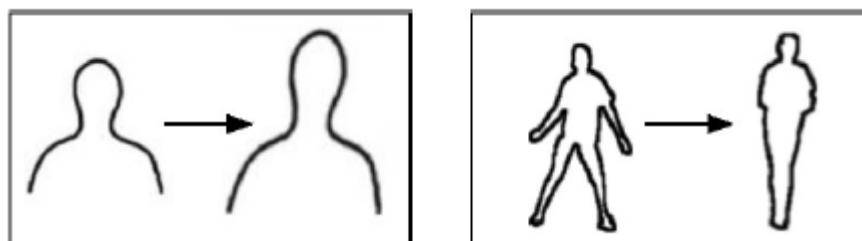


6 pav. Parametrizuota kvadrato transformacija

Dažniausiai objektų sekimui naudojamas metodas yra šablono atitikimas. Šio metodo veikimo principas yra pagrįstas srities paieška pagal objekto šabloną, kuris buvo apibūdintas prieš tai buvusiame kadre. Intensyvumas ir spalvos yra dažniausiai naudojamos savybės šablonams sudaryti. Šio metodo trūkumas yra daug atliekamų skaičiavimų.

### 2.5.3. Siluetu paremtas sekimas

Siluetu paremtas sekimas yra atliekamas įvertinant objekto sritis kiekviename kadre. Šis sekimo metodas naudoja informaciją įrašyta objekto viduje [20]. Siluetai yra sekami figūros atitikimo arba kontūro atitikimo metodu (7 pav.). Atliekant tokio pobūdžio objektų sekimą yra naudojamas objektų segmentavimas.



7 pav. Kontūrų kitimas

Objektai gali būti sudėtingi: ranka, galva, petys. Šie objektai negali būti apibendrinti įprastomis geometrinėmis figūromis. Šiems objektams sekti yra naudojamas kontūru paremtas objekto sekimas. Šio metodo esmė yra surasti prieš tai buvusio kadro sritis su apskaičiuotais vidurkiais. Šis metodas remiasi objektų spalvų histogramomis, objektų kraštais ar objektų kontūrais. Didžiausias privalumas yra tai, kad galima sekti daugybę objekto formų.

## 2.6. Fonų atimtis ir segmentavimas

### 2.6.1. Segmentavimo apibrėžimas

Vaizdo skaidymo į segmentus tikslas yra išskirti mus dominčias sritis. Kiekvienas segmentavimo algoritmas turi dvi problemas: geras kriterijus skaidymui ir algoritmas, kuris leistų pasiekti efektyvų segmentavimą.

Daugelis segmentavimo algoritmų segmentuoja kiekvieną kadrą, kad nustatytų objektus, tačiau toks metodas gali naudoti daug skaičiavimo resursų. Jei tikslas yra nustatyti judantį objektą, tai kiekvieno kadro segmentavimas nėra reikalingas. Tačiau toks variantas objektų sekime yra ribotas ir negali susidoroti su sudėtingomis vaizdo scenomis [17].

Segmentavimas gali būti paremtas [9]:

- Požymiu (angl. feature-based) – vartotojo paprašoma pasirinkti rinkinį pikselių priklausančių skirtingiems objektams ir pagal pasirinktų pikselių savybes, objektai yra klasifikuojami į vieną iš vartotojo nurodytų objektų tipų.
- Kontūru (angl. contour-based) – vartotojas turi apytikriai pažymėti objekto kontūrus ir automatizuotas algoritmas priderina nupieštus kontūrus pagal tikrus objekto kontūrus.
- Sritimi (angl. region-based) – vartotojui leidžiama vaizde sukurti objekto formą.

### 2.6.2. Fonų atimties apibrėžimas

Objektų radimas gali būti atliekamas sukuriant pradinį galinį foną ir vėliau galinis fonas yra lyginamas su kiekvienu nauju kadru. Tada bet koks didelis skirtumas tarp kadro yra identifikuojamas kaip judantis objektas. Dažniausiai yra naudojamas sujungtų komponentų algoritmas [11], kuris suranda atitinkamų segmentuotų objektų sujungtas sritis.

Galinio fono atnaujinimas yra atliekamas ten, kur atsiranda naujas objektas ir jis tam tikrą laiką nejuda, t.y. tampa statiniu objektu, visa kita yra laikoma arba judančių objektu arba triukšmu. Šio algoritmo teigiama savybė yra ta, kad gali atmesti pasikeitimus atsirandančius dėl laikino didelio vaizdo sujaukimo [15].

Siekiant sukurti fono atimties operaciją atsparią atspindžiams, vienas iš būdų yra sumodeliuoti galinį foną statišką kiekviename pikselyje [13]. Tada įvertinant yra suskaičiuojamas ryškumo ir spalvų iškreipimas RGB spalvų paletėje. Kiekvienas pikselis yra aprašomas  $(E_i, s_i, a_i, b_i)$ , kur  $E_i$  yra vektorius su apskaičiuotomis pikselio raudonos, žalios,



mėlynos komponentių vidurkiais per buvusių  $N$  galinio fono kadru:  $s_i$  – vektorius, kuriame laikomos standartiniai spalvų nukrypimų reikšmės,  $a_i$  – ryškumo iškraipymo reikšmės,  $b_i$  – chromatinių iškraipymų reikšmės.

Daugumos galinio fono atimties algoritmų trūkumas yra mėginimas išspręsti dviejų klasių (angl. two-class) klasifikavimo problemą, kai turime neapibūdintą vieną klasę. Algoritmas turi nuspręsti ar pikselis yra galinis fonas ar ne. Galinio fono kaita gali būti sudaryta apžvelgiant pikselius per laiko tarpą ir darant prielaidas apie apšvietimo kaitą.

Kai mes turime modelius sukurtus judintiems objektams, tada mes galime tai klasifikuoti kaip dviejų klasių problemą, išskiriant galinio fono ir priekinio fono objektus. Tai galime apžvelgti šiuo pavyzdžiu. Įsivaizduokim galinio fono pikselis yra atvaizduojamas spalva  $C_b$ , laikomas turėti spalvą  $C_o$  tam tikrame kadre, kai mes spėjame, kad jis bus uždengtas pirminio fono objekto spalvos pikseliu  $C_f$ . Jeigu  $C_o$  labai panašus į  $C_f$  ir net nepanašus į  $C_b$ , net tada  $C_b$  pikselis mūsų algoritmo bus laikomas kaip galinio fono pikselis [12].

### 2.6.3. Segmentavimo algoritmai

Įprastiniai segmentavimo metodai yra naudojami, kai vaizdo kamera yra nejudanti, tačiau egzistuoja metodai, kurie sugeba prisitaikyti prie kintančios aplinkos [3, 10].

#### „Heikkila ir Olli“ algoritmas

Pikselis yra pažymimas priklausantis pirminiam fonui jeigu,

$$|I_t - B_t| > \tau$$

kur  $\tau$  yra ribinė reikšmė. Mažų vaizdo įrašų sričių pašalinimas yra atliekamas filtruojant  $3 \times 3$  langelių branduoliu (angl. kernel).

Galinio fono atnaujinimas yra atliekamas pagal tokią formulę

$$B_{t+1} = \alpha I_t + (1 - \alpha)B_t$$

kur  $\alpha$  yra laikomas maža reikšme, kad išvengtume dirbtinių judančio vaizdo objekto „uodegų“ susidarymo.

Galinio fono korekcijos atliekamos tokiu principu:

1. Jeigu pikselis yra pažymėtas priekiniu fonu ilgiau negu  $m$  per paskutinius  $M$  kadru, tada galinis fonas yra atnaujinamas taip  $B_{t+1} = I_t$ . Tokia korekcija atliekama tam, kad būtų kompensuoti staigūs apšvietimo pasikeitimai ir naujų statinių objektų atsiradimas
2. Jeigu pikselis dažnai keičia būseną iš priekinio fono į galinį foną, tai jis yra slepiamas nuo įterpimo į priekinį foną.

### „Adaptive mixture of Gaussians“ algoritmas

Kiekvienas pikselis yra kuriamas atskirai „maišant“ K kiekį „Gaussians“.

$$P(I_t) = \sum_{i=1}^K \omega_{i,t} \eta(I_t; \mu_{i,t}, \Sigma_{i,t})$$

kur  $K=4$  arba  $K=3..5$ . Taip pat laikoma, kad  $\Sigma_{i,t} = \sigma_{i,t}^2 I$ .

Galinio fono atnaujinamas yra atliekamas prieš aptinkant pirminį foną. Jeigu  $I_t$  atitinką komponentą  $i$ , pvz.  $I_t$  yra tarp  $\mu_{i,t}$  nuokrypio  $\lambda$ , tada  $i$ -tasis komponentas yra atnaujinamas taip:

$$\begin{aligned}\omega_{i,t} &= \omega_{i,t-1} \\ \mu_{i,t} &= (1 - p)\mu_{i,t-1} + pI_t \\ \sigma_{i,t}^2 &= (1 - p)\sigma_{i,t-1}^2 + p(I_t - \mu_{i,t})^T (I_t - \mu_{i,t})\end{aligned}$$

kur  $p = \alpha \Pr(I_t | \mu_{i,t-1}, \Sigma_{i,t-1})$

Komponentai, kurie neatitinka  $I_t$  yra atnaujinami taip:

$$\begin{aligned}\omega_{i,t} &= (1 - \alpha)\omega_{i,t-1} \\ \mu_{i,t} &= \mu_{i,t-1} \\ \sigma_{i,t}^2 &= \sigma_{i,t-1}^2\end{aligned}$$

Jeigu  $I_t$  neatitinka jokio komponento, tai labiausiai panašus komponentas yra pakeičiamas nauju komponentu  $\mu_{i,t} = I_t$ .

Po atnaujinimų, svoriai  $\omega_{i,t}$  yra pakartotinai normalizuojami.

Pirminis fonas aptinkamas tokia tvarka. Visi komponentai surikiuojami mažėjančia tvarka pagal  $\omega_{i,t} / \|\Sigma_{i,t}\|$ . Tada tegul

$$B = \operatorname{argmin}_b \left( \frac{\sum_{i=1}^b \omega_{i,t}}{\sum_{i=1}^K \omega_{i,t}} > T \right)$$

Komponentai nuo 1..B yra laikomi galiniu fonu. Tada jei  $I_t$  neatitinka nė vieno iš šių komponentų, tai pikselis yra pažymimas kaip priklausantis pirminiam fonui. Tada pikseliai yra suskirstomi į segmentus panaudojant sujungtų komponentų algoritmą.

### „PFINDER“ algoritmas

Šis algoritmas naudoja paprastą schemą, kur galinis fonas yra modeliuojamas viena reikšme, kur atnaujinimas atliekamas taip:

$$B_t = (1 - \alpha)B_{t-1} + \alpha I_t$$

Pirminio fono pikseliai kuriami pagal vidurkius ir kovariacijas, kurie yra rekursiškai atnaujinami. Toks principas reikalauja tuščio pirminio kadro.

### „W<sup>4</sup>“ algoritmas

Pikselis yra pažymimas priklausantis pirminiam fonui jeigu

$$|M - I_t| > D \text{ arba } |N - I_t| > D$$

kur parametrai M, N, D apibūdina minimalią, maksimalią ir didžiausią tarp kadru apžvelgiamą absoliutų skirtumą galinio fono vaizde. Šie parametrai yra įvertinami per pirmąsias kelias vaizdo įrašo sekundes ir periodiškai yra atnaujinami toms vaizdo dalims, kuriose nėra pirminio fono objektų. Gautas pirminio fono vaizdas yra ardomas (angl. erode), kad būtų pašalinti 1 pikselio triukšmai. Tada įvykdomas sujungtų komponentų algoritmas ir pašalinami mažos sritys. Likusios vaizdo sritys yra išplečiamos (angl. dilate) ir paskui ardomos.

### „LOTS“ algoritmas

Yra naudojami trys galinio fono modeliai: pirminis, antrinis, senas galinis fonas.

Jie yra atnaujinami taip:

1. Jei pikselis nėra pažymėtas kaip priklausantis pirminiam fonui, tai pirminis galinis fonas yra atnaujinamas taip:

$$B_{t+1} = \alpha I_t + (1 - \alpha)B_t$$

Jei pikselis yra pažymėtas kaip priklausantis pirminiam fonui, tai pirminis galinis fonas yra atnaujinamas taip:

$$B_{t+1} = \beta I_t + (1 - \beta)B_t$$

$\alpha$  yra pasirenkamas iš intervalo [0,0000610351.. 0,25], o  $\beta=0.25\alpha$ .

2. Pikseliuose, kur dabartinio vaizdo reikšmės nėra labai skirtingos nuo dabartinių antrinio galinio fono reikšmių ir  $\alpha$  reikšmė lygi tokiai pačiai reikšmei, kuri buvo panaudota pirminiam galiniam fonui, antrinis galinis fonas yra atnaujinamas taip:

$$B_{t+1} = \alpha I_t + (1 - \alpha)B_t$$

Pikseliuose, kur skirtumas yra didelis, antrinis galinis fonas yra atnaujinamas taip:

$$B_{t+1} = I_t$$

3. Senasis galinis fonas yra dabartinio vaizdo kopija nuo 9000 iki 18000 kadru prieš tai.

Pirminio fono aptikimas yra paremtas histereze su prisitaikančia riba. Atliekami pataisymai:

1. Mažos pirminio fono sritys yra atmetamos;
2. Pikseliai, kurių reikšmės virs numatytos ribos dabartiniame kadre, yra lyginami su prieš tai buvusio kadro pikseliais. Didelis pakitimas yra laikomas kaip apšvietimo pasikeitimas. Bendra ribinė reikšmė laikinai yra padidinama.

3. Kiekvieno objekto srities pikselių reikšmės yra lyginamos su pirminio ir antrinio galinių fonų atitinkančiomis sritimis, kad surasti bendrą ryškumo vidurkį. Šis žingsnelis atliekamas pašalinti dirbtinai sukurtus objektus dėl apšvietimo kaitos.

### **„Halevy“ algoritmas**

Visuose pikseliuose, kur  $S(I_t)$  yra glodinta (angl. smooth)  $I_t$  versija, galinis fonas yra atnaujinamas taip:

$$B_{t+1} = \alpha S(I_t) + (1 - \alpha)B_t$$

Pirminio fono pikseliai yra identifikuojami pagal  $S(I_t - B_t)$  maksimumą. Šis metodas naudojama koeficientą  $\alpha=[0.3..0.5]$  ir remiasi išsiskyrimo efekto, nustatant skirtumus tarp kadru, principu.

Reikalingas kadru skaičius  $t$ , kad nusistovėtų galinis fonas po iniciacijos, yra apskaičiuojamas:

$$(1 - \alpha)^t < 0.1$$

### **„Cutler“ algoritmas**

Šis algoritmas naudoja spalvotus vaizdus, kadangi tai duoda geresnį segmentavimą nei monochromatiniai vaizdai, tokiose vietose kur mažas ryškumas, pvz. objektas tamsiame šešėlyje.

Galinis fonas yra įvertinamas laikina mediana pagal paskutinius  $N$  kadrus, kur  $N$  kinta nuo 50 iki 200.

Pikselis pažymimas priklausantis pirminiam fonui jeigu

$$\sum_{C \in R, G, B} |I_t(C) - B_t(C)| > K\sigma$$

kur  $\sigma$  yra triukšmų nuokrypio koeficientas, o  $K$  apytiksliai pasirinkta konstanta (dažniausiai būna lygi 10). Objektų atrinkimui šis metodas taip pat naudoja šablonų atitikimą.

### **„Wallflower“ algoritmas**

Naudojami du regresyvūs modeliai:

$$B_t = - \sum_{k=1}^p a_k B_{t-k}$$

$$\bar{I}_t = - \sum_{k=1}^p a_k I_{t-k}$$

kartu su galinio fono ribine reikšme, kuri apskaičiuojama:

$$\varepsilon(e_t^2) = \varepsilon(B_t^2) + \sum_{k=1}^p a_k \varepsilon(B_t B_{t-k})$$

Pikseliai pažymimi galiniu fonu jeigu

$$\tau = 4\sqrt{\varepsilon(e_t^2)}$$

$$|I_t - B_t| < \tau \text{ ir } < |I_t - \bar{I}_t|\tau$$

Koeficientas  $a_k$  yra atnaujinamas kiekviename kadre. Jeigu daugiau nei 70% vaizdo yra klasifikuojama kaip pirminis fonas, tai algoritmas yra atmetamas ir pakeičiamas „atsarginiu“ algoritmu.

### **„Frame differencing“ algoritmas**

Jeigu nėra kameros judesio, tai kadrų diferencijavimo metodas yra naudojamas judančių objektų segmentavimui [5]. FDI yra suformuotas vaizdas lyginant du kadrus. FDI pikselio reikšmė yra pakeičiama į 255, jei skirtumas didelis, kitu atveju įrašomas 0. Žemiau pateikiama formulė, kur  $n$  yra kadrų skaičius, o  $T_d$  slenkstis (angl. threshold).

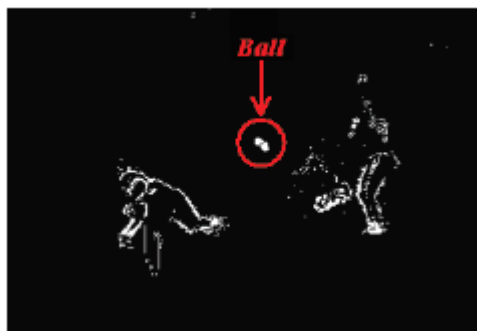
$$FDI_n(x, y) = \begin{cases} 255, & \text{if } |Intensity_n(x, y) - Intensity_{n-1}(x, y)| > T_d \\ 0, & \text{otherwise} \end{cases}$$

Tačiau galime pastebėti, kad kamuoliukas tampa didesnis (8 pav.), nes matuojamas bendras kadrų skirtumo intensyvumas. Jeigu vaizdas ryškus, tai turime padaryti kamuoliuko ryškumą didesni. Šiam tikslui pasiekti yra naudojamas PFDI, kuris apibrėžiamas formule:

$$PFDI_n(x, y) = \begin{cases} 255, & \text{if } Intensity_n(x, y) - Intensity_{n-1}(x, y) > T_d \\ 0, & \text{otherwise} \end{cases}$$

Tokiu būdu sėkmingai segmentuojama sritis, kurioje yra kamuoliukas ir kur dviejų kadrų skirtumo intensyvumas yra didesnis (9 pav.).

Atlikus segmentavimą yra naudojamos morfologinės operacijos pašalinti triukšmus ir užpildyti sritis.



8 pav. FDI segmentavimas [5]

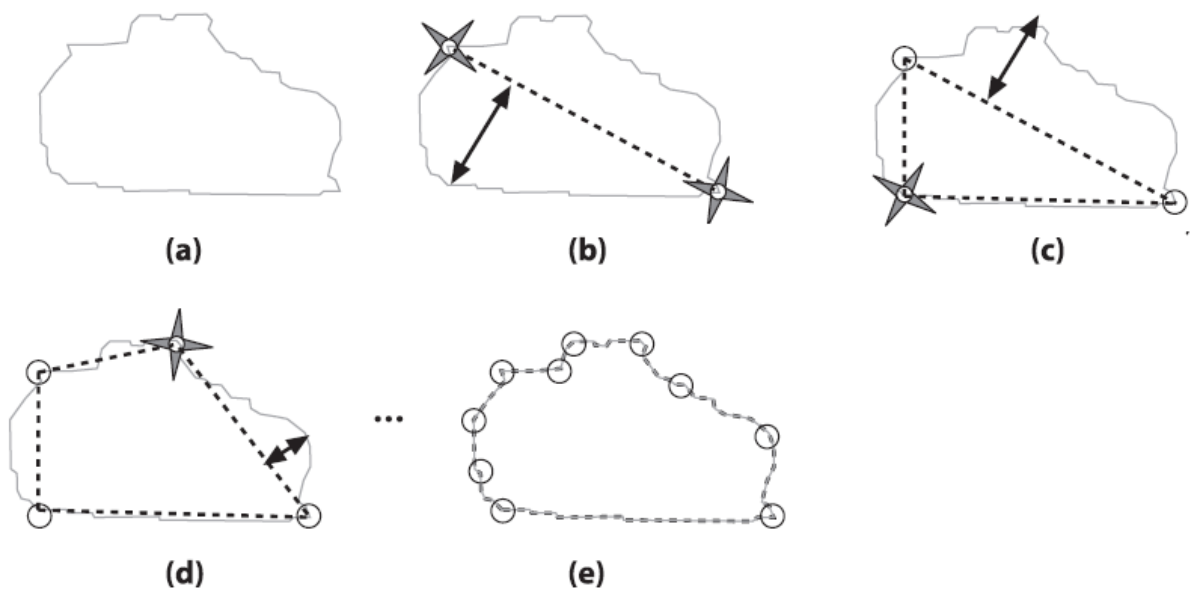


9 pav. PFDI segmentavimas [5]

Objektų sėkmingam segmentavimui yra naudojamas kas antras ar trečias kadras (nustatyta eksperimentavimo metu), nes tokiu atveju išgaunama pilnesnė objekto forma [4, 5].

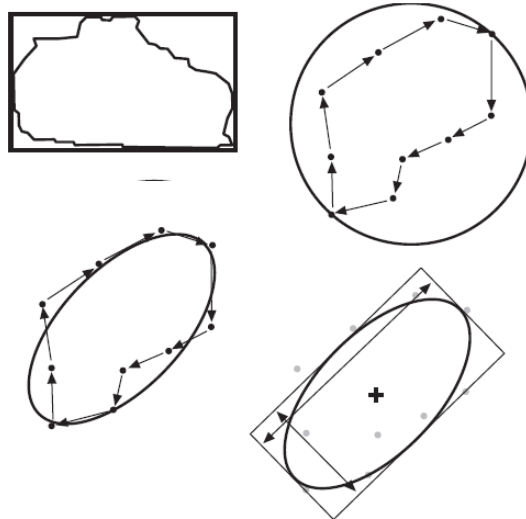
## 2.7. Kontūro aproksimacija

10 pav. pavaizduota kontūro aproksimacija. Pradžioje turime kontūrą (a). Algoritmas išrenka du ekstremumo taškus ir sujungia juos linija (b). Tada ieškomas taškas, toliausiai nutolęs nuo nubrėžtos linijos, ir pridamas į aproksimaciją. Procesas yra iteracinis (d - e). Toliau renkamas kitas toliausiai nutolęs taškas ir tai atliekama tol, kol atstumas nuo taško iki linijos yra mažesnis nei apibrėžtos reikšmės.



10 pav. Kontūro aproksimacija

Atlikus aproksimaciją yra skaičiuojamos kontūro charakteristikos kaip ilgis ar plotas, kontūro momentai, aplink kontūrą apibrėžtas stačiakampis (11 pav.).



11 pav. Kontūro apibrėžimas geometrinės figūromis

Vaizdo objekto kontūro įvertinimas gali būti atliktas aktyvaus kontūro metodu. Šis metodas sėkmingai pritaikytas spręsti tokioms problemoms kaip vaizdo segmentavimas, objekto aptikimas, pozicijos nustatymas ir sekimas vaizdo kadruose [15].

## 2.8. 3D vaizdo analizė 2D vaizde

Kameros kalibravimas yra būtinas kai norime susieti žaidėjų ar kamuolio pozicijas su realaus pasaulio koordinatėmis [6]. Geometrinė 3D realaus pasaulio koordinatinių  $(x, y, z)$  transformacija į 2D vaizdo koordinates  $(u', v')$  gali būti išreikšta taip:

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \text{ kur } \begin{matrix} u' = \frac{u}{w} \\ v' = \frac{v}{w} \end{matrix}$$

Siekiant atlikti sekimą 3D vaizde reikia atlikti kelis žingsnius. Visų pirma objektas yra aptinkamas ir sekamas skirtingų kamerų vaizduose. Tada skirtingi 2D vaizdai yra integruojami tam, kad būtų gautas bendras 3D vaizdas panaudojant žinomus judesio metodus [8]. Trimačiam atvaizdavimui suformuoti yra naudojama Hough transformacija [7].

## 2.9. Judančio vaizdo objekto aptikimas

### 2.9.1. Objekto aptikimas naudojant filtrus

Visų pirma yra atliekamas objekto segmentavimas, kuris įvykdomas fonų atimties operacijos metu. Tada atlikus segmentavimą, atliekame morfologines operacijas triukšmams pašalinti.

Turint segmentuotą vaizdą yra atliekama objekto kandidato paieška. Paieškai yra panaudojamas trijų lygių filtras [5, 4] :

- Objekto dydžio filtras – objektai panaikinami jei jie nepatenka į apibrėžtą reikšmių aibę.
- Objekto formos filtras - objektas gali įgyti skirtingą formą, nei ji yra iš tikrųjų, tačiau formatinis koeficientas (angl. aspect ratio) turi išlikti apibrėžtoje reikšmių aibėje.
- Objekto pilnumo filtras – kai kurie objektai skirtingų formų gali būti praleisti mūsų dydžio ir formos filtrų dėl tinkamo dydžio ir formato koeficiento. Tam yra sukuriamas objekto pilnumo koeficientas  $D_f$ . Objektai, kurių pilnumo koeficientas yra mažesnis už slenkstį (angl. threshold).  $D_f$  apibrėžiame tokia formule:

$$D_f = S_{obj} / A_{b-box},$$

kur  $S_{obj}$  yra objekto dydis, o  $A_{b-box}$  yra aplink objekto apibrėžto kvadrato srities dydis.

Po filtravimo, objektai yra skirstomi į izoliuotus ir persidengiančius. Tai yra nustatoma pagal artimiausiai esančius objektus kadre. Objektas yra laikomas izoliuotu, jeigu nėra kaimyninių objektų per apibrėžto objekto pločio vidurkio atstumą, kitu atveju tai yra laikomas persidengiančiu objektu.

Jeigu nepavyksta aptikti judančio objekto dėl kažkokių priežasčių, tai objekto judėjimo trajektorija gali būti nuspėta ir atvaizduota pasinaudojant „Kalmano“ filtru [5,18].

### 2.9.2. Objekto aptikimas pagal šablona

Objektas aptinkamas išskyrus skirtumus tarp dviejų kadrų. Tada imama taškų spalvos informacija, tačiau spalva gali keistis dėl skirtingų žiūrėjimo kampų, apšvietimo [6]. Objektai gali būti aptikti remiantis sritimis, kurios nusakomos šablonais [1]. Sekamas objektas yra apibūdinamas sąsajų nuorodomis  $\mathbf{r} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M)^T$  pirmame vaizdo kadre. Taško  $u$  chromatiškumo lygio intensyvumas laiko momentu  $T$  yra apibūdinamas  $f(u, t)$ . Tada vektorius  $f(\mathbf{r}, t)$  susideda iš visos srities intensyvumo reikšmių laiko momentu  $t$  ir tai yra vadinama šablonu. Inicijacijos metu yra apskaičiuojamas pirmo paveikslo nuorodų šablonas  $f(\mathbf{r}, 0)$ . Šablonų atitikimas yra atliekamas skaičiuojant judesio parametrus  $\mu(t)$ , kuris sumažina intensyvumo skirtumus tarp nuorodinio šablono ir dabartinio šablono. Žemiau pateikiama formulė:

$$\mu(t) = \underset{\text{size } \mu}{\operatorname{argmin}} \|f(\mathbf{r}, 0) - f(\mathbf{g}(\mathbf{r}, \mu), t)\|_2$$



Funkcija  $g(r, \mu)$  apibūdina geometrinę transformaciją, kuri yra parametrizuojama vektoriumi  $\mu$ . Funkcijos  $\mu(f)$  paieškos atlikimas jėgos metodu (angl. brute-force) reikalauja daug skaičiavimų, todėl efektyviau yra aproksimuoti  $\mu$  per linijinę sistemą

$$\hat{\mu}(t+1) = \hat{\mu}(t) + A(t+1)(f(r, 0) - f(g(r, \hat{\mu}(t)), t+1))$$

Objektai gali būti aptikti remiantis sritimis, kurios nusakomos histogramomis [1]. Objektas yra žymimas nuorodų sritimi, kuri pažymima  $R(x(t))$ , kur  $x(t)$  susideda iš skirtingų laikų srities parametrų. Informacija laikoma nuorodų srityje yra naudojama modeliuoti judantį objektą. Informacija gali susidėti iš spalvos, achromatinės reikšmės, gradiento. Kiekvieno laiko momentu  $t$  ir kiekvienai būsenai  $x(t)$  judančio objekto apibūdinamas susideda iš tikimybių tankio funkcijos  $p(x(t))$  pasirinktų funkcijų iš srities  $R(x(t))$ .

### 2.9.3. Objekto aptikimas pagal histogramą

Spalvotų vaizdų histogramos yra naudojamos vaizdo objektų sekimui, dėl paprastumo, universalumo, greitumo. Taip pat histogramų naudojimas plačiai taikomas spalvotų objektų atpažinimui pagal spalvos indeksavimą [16]. Kraštų, spalvos, kampų histogramos suformuoja savybių tipą, kuris perduodamas klasifikatoriams atpažinti objektus. Histogramos yra klasikinis įrankis naudojamas kompiuteriniuose vaizdų apdorojimuose.

Histogramos suteikia galimybę palyginti histogramas tarpusavyje pagal apsibrėžtą panašumo kriterijų. Galimi histogramų lyginimo metodai [11]:

#### „Koreliacijos (angl. Correlation)“ metodas

Tobulas atitikimas yra laikomas, kai gauta reikšmė yra lygi 1; visiškas neatitikimas yra laikomas, kai gauta reikšmė lygi -1; gavus reikšmę 0 – koreliacija neegzistuoja.

$$d_{corel}(H_1, H_2) = \frac{\sum_i H_1'(i)H_2'(i)}{\sqrt{\sum_i H_1'^2(i)H_2'^2(i)}}$$

kur

$$H_k'(i) = H_k(i) - \left(\frac{1}{N}\right) \left(\sum_j H_k(j)\right)$$

ir  $N$  lygus histogramos dėtuvų (angl. bin) skaičiui.

#### „Chi-square“ metodas

Tobulas atitikimas yra laikomas, kai gauta reikšmė yra lygi 0; visiškas neatitikimas yra laikomas prie neapibrėžtos reikšmės.

$$d_{chi-square}(H_1, H_2) = \sum_i \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)}$$

**„Susikirtimų (angl. Intersection)“ metodas**

Gautos didelės reikšmės reiškia didelį atitikimą, o gautos mažos reikšmės reiškia prastą atitikimą. Jei histograma yra normalizuota, tai visiškasis atitikimas atitinka reikšmę 1, o visiškasis neatitikimas atitinka reikšmę 0.

$$d_{intersection}(H_1, H_2) = \sum_i \min(H_1(i), H_2(i))$$

**„Bhattacharyya atstumo (angl. Bhattacharyya distance)“ metodas**

Gautos mažos reikšmės reiškia didelį atitikimą, o gautos didelės reikšmės mažą atitikimą. Tobulus atitikimas yra, kai gauta reikšmė yra lygi 0, o visiškasis neatitikimas yra, kai gauta reikšmė lygi 1.

$$d_{Bhattacharyya}(H_1, H_2) = \sqrt{1 - \frac{\sum_i \sqrt{H_1(i) * H_2(i)}}{\sqrt{\sum_i H_1(i) * \sum_i H_2(i)}}}$$

**„EMD (angl. Earth Mover’s Distance)“ metodas**

Apšvietimo pasikeitimas gali sukelti spalvų reikšmių poslinkį ir tai įtakoja anksčiau aptartų metodų žlugimą lyginant histogramas. Vietoj histogramų atitikimo metodo mes galime panaudoti histogramų poslinkio metodą. Šis metodas skaičiuoja kiek darbo reiks perkelti vienos histogramos formą į kitą, įtraukiant kintančias histogramos dalis į naują poziciją. Šis metodas veikia N dimensijose.

Tobulus atitikimas yra laikomas, kai gauta reikšmė yra lygi 0; visiškasis neatitikimas yra laikomas, kai gauta reikšmė lygi 1.

**Metodų apibendrinimas**

Histogramos Pavyzdys	Atitikimų reikšmės				
	Correlation:	Chisquare:	Intersection	Bhattacharyya:	EMD:
Tikslus atitikimas	1.0	0.0	1.0	0.0	0.0
Pusiau atitinka	0.7	0.67	0.5	0.55	0.5
Visiškas neatitikimas	-1.0	2.0	0.0	1.0	1.0

12 pav. Histogramų metodų palyginimų reikšmės

Greičiausiai veikia susikirtimų metodas, tačiau šis metodas nėra labai tikslus. „Chi-square“ ir „Bhattacharyya“ metodai veikia lėčiau, bet gaunami rezultatai yra daug tikslesni. Geriausius rezultatus pateikia „Earth Mover’s Distance“ metodas, tačiau jis yra daug lėtesnis, lyginant su kitais histogramų palyginimo metodais, skaičiavimo atžvilgiu [11].

## **2.10. Išvados ir priimtas sprendimas**

Vaizdų apdorojimas yra sudėtingas procesas. Priklausomai nuo nagrinėjamos situacijos ir aplinkybių yra taikomi skirtingi vaizdų apdorojimo metodai. Didžiausią neigiamą įtaką vaizdų apdorojimui turi triukšmai.

Remiantis kitose sporto šakose pritaikytais metodais, bus mėginama atlikti judančio vaizdo objekto (rutulio) sekimą.

### **2.10.1. Priimto sprendimo tikslai**

Rutulio mėtymas bus filmuojamas stacionaria kamera, todėl vaizdo įrašo segmentavimui bus naudojamas kadro skirtumo (angl. frame difference) algoritmas. Segmentavus vaizdą, objektų kontūrams rasti bus naudojamas aktyvus kontūro metodas. Pagal rasto objekto kontūro charakteristikas bus sprendžiama ar aptiktas objektas yra mūsų mėtomas rutulys.

## **3. PROJEKTINĖ DALIS**

### **3.1. Tikslas**

Šios dalies paskirtis yra pateikti pasirinktų technologijų pagrindimą ir realizuotos sistemos esminius architektūros vaizdus ir įvertinti realizuotos sistemos architektūrinį sprendimą.

### **3.2. Bendra apžvalga**

Šiame skyriuje pateikiama:

- Pasirinktos technologijos – aprašomi pasirinktų technologijų kriterijai.
- Statinis sistemos vaizdas – paketų tarpusavio sąveikos diagrama.
- Panaudojimo atvejai - aprašomi sistemos aktoriai ir jų veiksmi.
- Apibendrinta sekos diagrama - pateikiama stebėjimo ir analizės apibendrintas procesas.
- Pagrindinio algoritmo aprašymas - pateikiami pagrindinio algoritmo žingsniai.

### **3.3. Pasirinktos technologijos**

#### **3.3.1. „OpenCV“ vaizdo apdorojimo biblioteka**

„OpenCV“ yra atviro kodo kompiuterinių vaizdų apdorojimo biblioteka. Biblioteka yra parašyta C ir C++ kalbomis ir veikia šiose operacinėse sistemose: Linux, Windows, Mac OS X.

„OpenCV“ buvo suprojektuota atlikti efektyvius skaičiavimus ir didžiausias dėmesys sutelktas ties realaus laiko aplikacijomis. „OpenCV“ yra parašyta optimizuota C kalba ir gali naudotis kelių branduolių procesorių teikiamomis galimybėmis.

Ši biblioteka turi supaprastintą kompiuterinių vaizdų apdorojimo infrastruktūrą. Biblioteka turi daugiau nei 500 funkcijų, kurios apima daugelį vaizdo sričių ir taip pat suteikia atlikti tokias galimybes: gamyklinių produktų tikrinimą, medicinoje naudojamų vaizdų apdorojimą, kameros kalibravimą, erdvinių vaizdų sudarymą.

#### **3.3.2. EMGU**

EMGU – skirtingų platformų (angl. cross platform) „OpenCV“ vaizdo apdorojimo bibliotekos .NET įvynioklis (angl. wrapper) leidžiantis iškviešti „OpenCV“ funkcijas suderinamose .NET kalbose: C#, VB, VC++, IronPython.

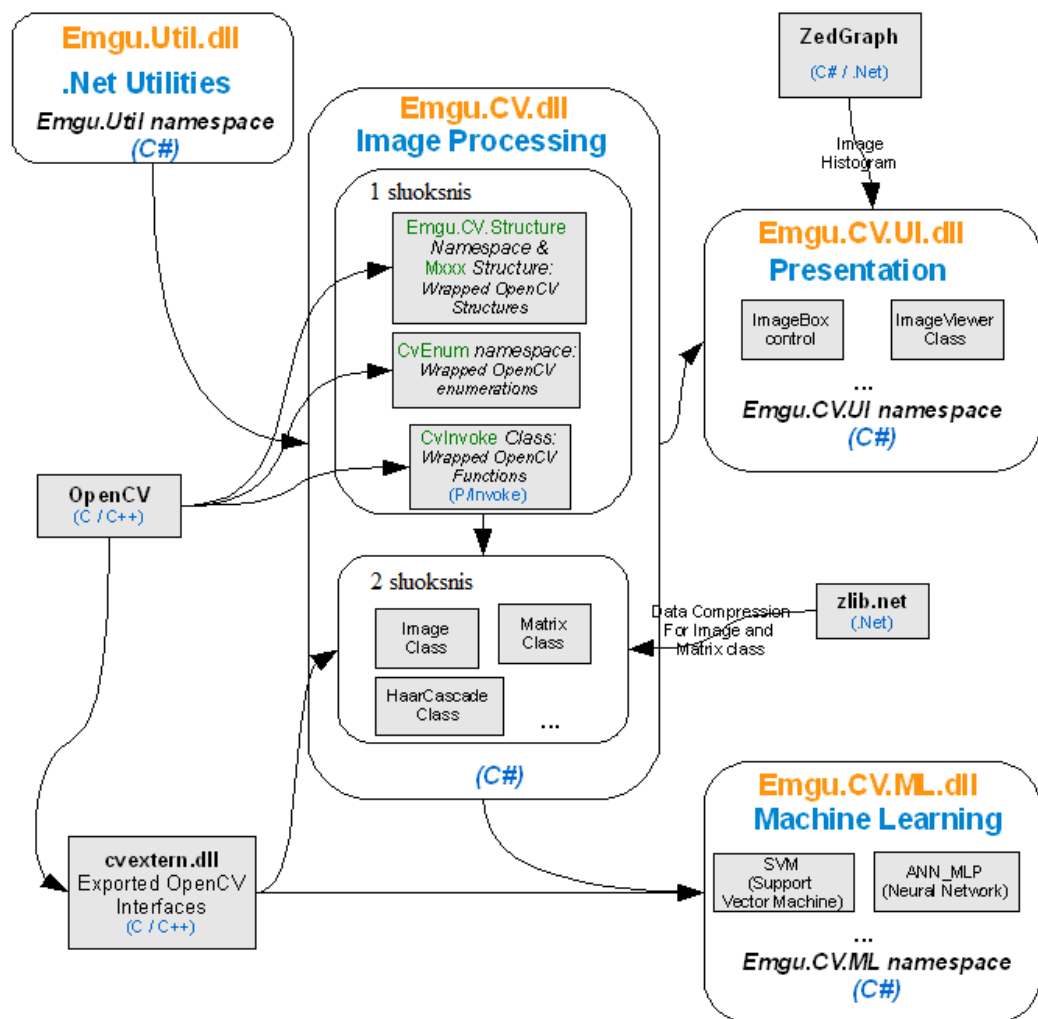
EMGU yra parašyta išimtai tik C# kalba ir puikiai tiko prie pasirinktos programinės įrangos kūrimo įrankio „Visual Studio 2005“ ir pasirinktos kalbos C#. EMGU papildomos savybės:

- Vaizdų klasė su pagrindinėmis spalvomis ir spalvų gyliais
- Automatinis šiukšlių surinkimas
- Vaizdo aprašymas keitimu i nuoseklią XML formą (angl. serializable)
- Automatinio užbaigimo (angl. Intellisense) funkcijos palaikymas
- Galimybė naudoti vaizdų klasę arba tiesiogiai išsikviesti „OpenCV“ funkcijas
- Vaizdo pikselio apdorojimo pagrindinės funkcijos

### EMGU architektūra

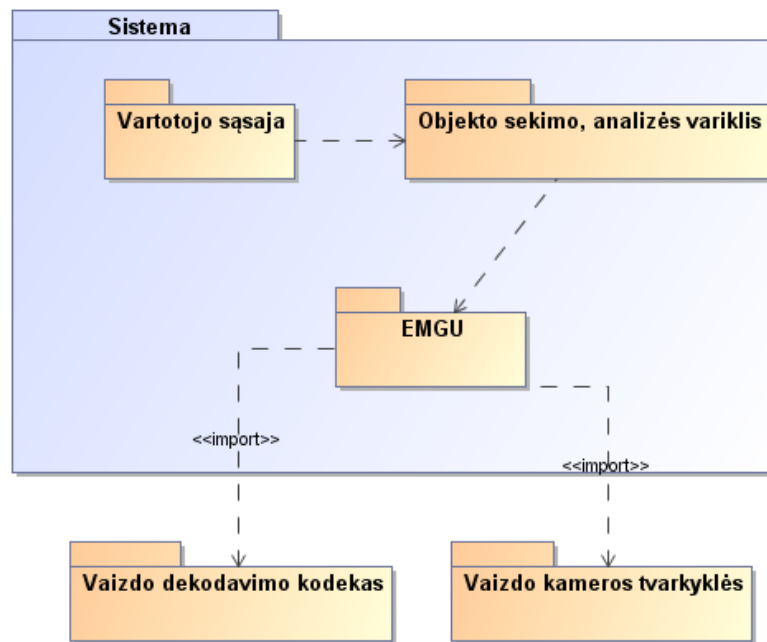
EMGU susideda iš dviejų įvynioklių sluoksnių:

- 1 sluoksnis susideda iš funkcijų, struktūrų, išvardinimo (angl. enumeration) susiejimų su „OpenCV“ funkcijomis
- 2 sluoksnis naudoja .NET funkcijas



13 pav. EMGU architektūra

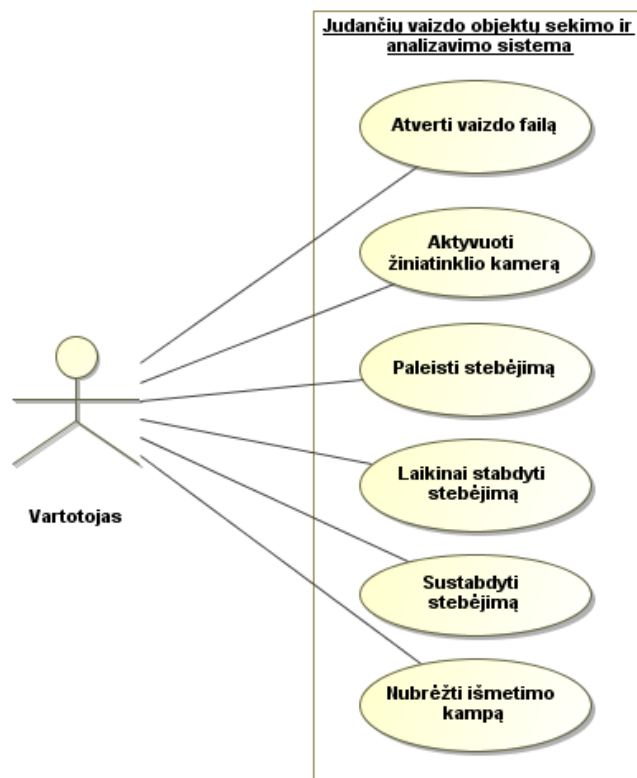
### 3.4. Statinis sistemos vaizdas



14 pav. Statinis sistemos vaizdas

### 3.5. Panaudojimo atvejai

Sistemoje vartotojai nėra skirstomi į atskiras grupes, todėl panaudojimo atvejai yra taikomi visiems sistemos vartotojams.



15 pav. Panaudojimo atvejai

**Panaudojimo atvejis:** Atverti vaizdo failą  
**Aprašas:** Atidaromas pasirinktas vartotojo AVI tipo video failas  
**Vartotojas/Aktorius:** Vartotojas  
**Prieš sąlyga:** Nevyksta stebėjimas iš vaizdo kameros arba neanalizuojamas vaizdo failas  
**Sužadinimo sąlyga:** Aktyvuojamas meniu punktas ir pasirenkamas norimas vaizdo failas  
**Po sąlyga:** Atidarytas vaizdo failas

**Panaudojimo atvejis:** Aktyvuoti žiniatinklio kamerą  
**Aprašas:** Vykdomas stebėjimas, kai vaizdo informacija yra gaunama iš žiniatinklio kameros  
**Vartotojas/Aktorius:** Vartotojas  
**Prieš sąlyga:** Neatidarytas vaizdo failas ar stebėjimas iš vaizdo kameros  
**Sužadinimo sąlyga:** Aktyvuotas meniu punktas  
**Po sąlyga:** Programa gauna vaizdo signalą iš vaizdo kameros

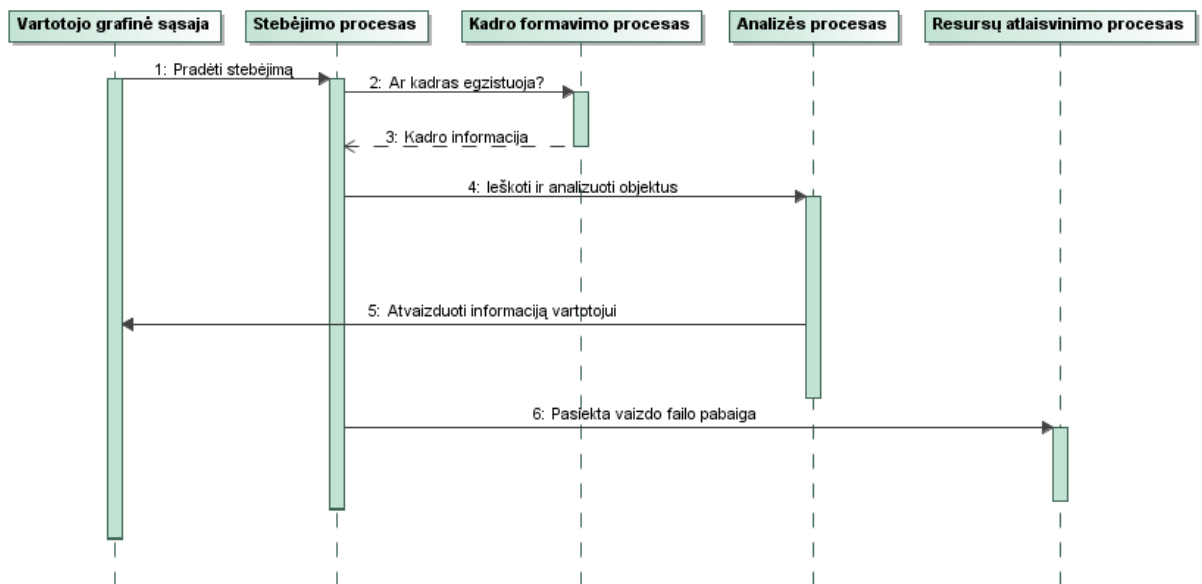
**Panaudojimo atvejis:** Paleisti stebėjimą  
**Aprašas:** Pradedamas judančio vaizdo objekto stebėjimas ir analizė  
**Vartotojas/Aktorius:** Vartotojas  
**Prieš sąlyga:** Laikina sustabdytas stebėjimas  
**Sužadinimo sąlyga:** Aktyvuotas meniu punktas  
**Po sąlyga:** Vykdomas objektų stebėjimas ir analizė

**Panaudojimo atvejis:** Laikina stabdyti stebėjimą  
**Aprašas:** Kažkuriuo laiko momentu stebėjimas laikina sustabdomas  
**Vartotojas/Aktorius:** Vartotojas  
**Prieš sąlyga:** Vyksta judančio vaizdo objekto stebėjimas  
**Sužadinimo sąlyga:** Aktyvuotas meniu punktas  
**Po sąlyga:** Judančio vaizdo objekto stebėjimas laikina sustabdytas

<b>Panaudojimo atvejis:</b>	Sustabdyti stebėjimą
<b>Aprašas:</b>	Vartotojas sustabdo judančio vaizdo objekto stebėjimą
<b>Vartotojas/Aktorius:</b>	Vartotojas
<b>Prieš sąlyga:</b>	Vyksta judančio vaizdo objekto stebėjimas
<b>Sužadinimo sąlyga:</b>	Aktyvuotas meniu punktas
<b>Po sąlyga:</b>	Judančio vaizdo objekto stebėjimas sustabdytas

<b>Panaudojimo atvejis:</b>	Nubrėžti išmetimo kampą
<b>Aprašas:</b>	Vartotojui nubrėžiamas stebėjimo kampas
<b>Vartotojas/Aktorius:</b>	Vartotojas
<b>Prieš sąlyga:</b>	Vyksta judančio vaizdo objekto stebėjimas
<b>Sužadinimo sąlyga:</b>	Vaizdo analizė baigta
<b>Po sąlyga:</b>	Nubrėžtas išmetimo kampas

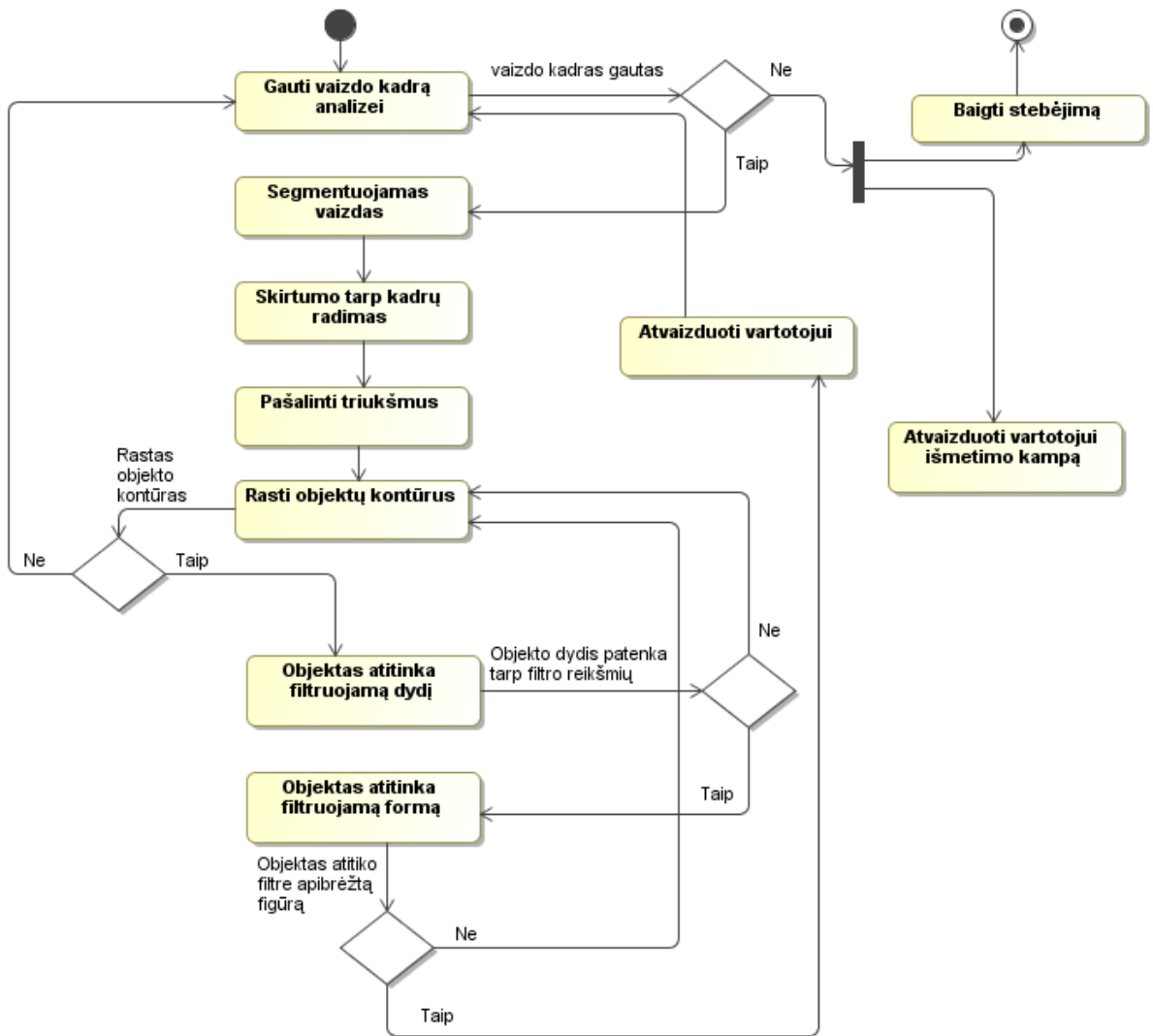
### 3.6. Stebėjimo proceso apibendrinta sekos diagrama



16 pav. Stebėjimo proceso sekos diagrama



### 3.7. Judančio vaizdo objekto stebėjimo ir analizės algoritmo aprašymas



17 pav. Judančio vaizdo objekto sekimo ir analizės algoritmo žingsniai

### 3.8. Išvados

Remiantis aprašyta architektūra, buvo realizuota programinė įranga, kuri atlieka judančio objekto sekimą ir analizę.

## **4. JUDANČIŲ VAIZDO OBJEKTŲ SEKIMO IR ANALIZĖS PROGRAMOS KOKYBĖS TYRIMAS**

### **4.1. Tikslas**

Tiriamoji dalis skirta „Judančių vaizdo objektų sekimas ir analizė“ programinės įrangos kokybės analizei.

### **4.2. Vartotojo sąsaja**

Realizuotos programinės įrangos vartotojo sąsaja yra paprasta ir lengvai suprantama. Vartotojo sąsajos funkcionalumas yra panašus į įprastinių vaizdo grotuvų vartotojo sąsajas. Meniu punktai buvo suklasifikuoti pagal atliekamas funkcijas, pateikiami tikslūs ir aiškūs pranešimai vartotojui. Programa gali naudotis bet kuris vartotojas, turintis elementarias naudojimosi kompiuteriu žinias.

### **4.3. Sistemos testavimas**

Vaizdinė informacija kompiuteryje yra saugoma skaičių matrica. Žmogaus akys, žiūrėdamos į šiuos skaičius, negali sudaryti bendro vaizdo. Automatinis testavimas taip pat yra neįmanomas, nes reikia sudaryti vaizdo matricas, kurios aprašo atvaizduojamo pikselio informaciją.

Sistemos testuojamumas yra sudėtingas, nes norint įsitikinti sukurto algoritmo veikimo teisingumu, reikia stebėti kaip programa atvaizduoja patį vaizdą ir kokius rezultatus mums pateikia. Gavus rezultatus turime įvertinti ar skaičiavimai buvo atlikti teisingai. Realus vaizdo įrašas suskaidomas į kadrus ir turėdami atvaizduotą objekto judėjimo trajektoriją lyginame objekto poziciją realiame vaizdo įrašė su eksperimento metu gauta objekto judėjimo trajektorija, todėl šios sistemos testavimas yra imlus darbui procesas.

### **4.4. Sistemos saugumas**

Sistema nesaugo jokių vartotojo asmeninių duomenų ar viešai neplatinamos informacijos, todėl sistemoje nebuvo realizuotas joks saugumą užtikrinantis mechanizmas.

### **4.5. Išvados**

Šio skyriaus dalyje atlikta realizuotos judančių vaizdo objektų sekimo ir analizės programos sistemos kokybės analizė.

## 5. JUDANČIŲ VAIZDO OBJEKTŲ SEKIMO IR ANALIZĖS TESTAVIMO EKSPERIMENTAI

### 5.1. Tikslas

Atliktų eksperimentinių tyrimų tikslas – patvirtinti arba paneigti darbo metu suformuotas hipotezes ir teiginius, įvertinti programos sugebėjimą užfiksuoti išmetamą rutulį, nustatyti kiek vidutiniškai užtrunka programa apdorodama vaizdo įrašą.

Svarbiausi uždaviniai, kurių atžvilgiu bus vykdomi tyrimai:

- ✓ Įvertinti programos veikimo tikslumą;
- ✓ Išsiaiškinti sistemos stipriąsias ir silpnąsias puses.

### 5.2. Eksperimento objektas

Eksperimento metu bus nagrinėjama „Judančių vaizdo objektų sekimas ir analizė“ programos charakteristikos.

### 5.3. Eksperimentas

#### 5.3.1. Eksperimentų aprašymas

Eksperimentams buvo naudojamas tokių parametrų kompiuteris:

- Intel Core 2 Duo T8100 2,1 GHz procesorius
- 4GB operatyvioji atmintis
- ATI Mobility Radeon HD 3650 vaizdo plokštė, turinti 512 MB operatyviosios atminties.
- Windows Vista x64bit

Buvo nufilmuoti vaizdo įrašai, kuriuose užfiksuotas rutulio metimas. Rutulio metimai atlikti taip, kad kamera metimą fiksuotų iš skirtingų kampų. Vėliau vaizdo įrašai buvo leidžiami su sukurta programine įranga ir analizuojamas programos veikimo metu užfiksuotų rezultatų teisingumas.

1 lentelėje pateikiami nufilmuotų vaizdo failų charakteristikos.

*1 lentelė. Vaizdo įrašų charakteristikos*

Failo pavadinimas	Vaizdo įrašo skiriamoji geba	Kadrų skaičius per sekundę (FPS)	Vaizdo įrašo trukmė
MVI_0001	640 x 480	10	6s 100ms
MVI_0002	640 x 480	10	3s 900ms
MVI_0003	640 x 480	10	5s 200ms

Failo pavadinimas	Vaizdo įrašo skiriamoji geba	Kadru skaičius per sekundę (FPS)	Vaizdo įrašo trukmė
MVI_0004	640 x 480	10	5s 500ms
MVI_0005	640 x 480	10	4s 600ms
MVI_0006	640 x 480	10	5s 200ms
MVI_0007	640 x 480	10	3s 200ms
MVI_0008	640 x 480	10	4s 800ms
MVI_0009	640 x 480	10	4s 100ms

### 5.3.2. Eksperimentų rezultatai

Atlikus eksperimentus, informacija buvo suklasifikuota ir pateikiama žemiau esančiose lentelėse.

2 lentelėje pateikiami vaizdo įrašo analizės trukmės. Tas pats vaizdo įrašas buvo leidžiamas kelis kartus, kad būtų galima apskaičiuoti tikslesnį analizės laiką.

Turint vaizdo įrašo apdorojimo laiko vidurkius, buvo išvesta formulė, kuri apskaičiuoja kiek laiko užtruks vaizdo įrašo analizė:

$$\text{vaizdo įrašo analizės trukmė} = \text{vaizdo įrašo trukmė} \times k,$$

kur vėlinimo koeficientas  $k=1,4$ . Vėlinimo koeficientas buvo paskaičiuotas taip

$$k = \frac{\sum_{i=1}^N \frac{\text{VaizdoĮrašoAnalizėsTrukmėsVidurkis}_i}{\text{VaizdoĮrašoTrukmė}_i}}{N}$$

kur  $i$  - vaizdo įrašas,  $N$  – vaizdo įrašų kiekis.

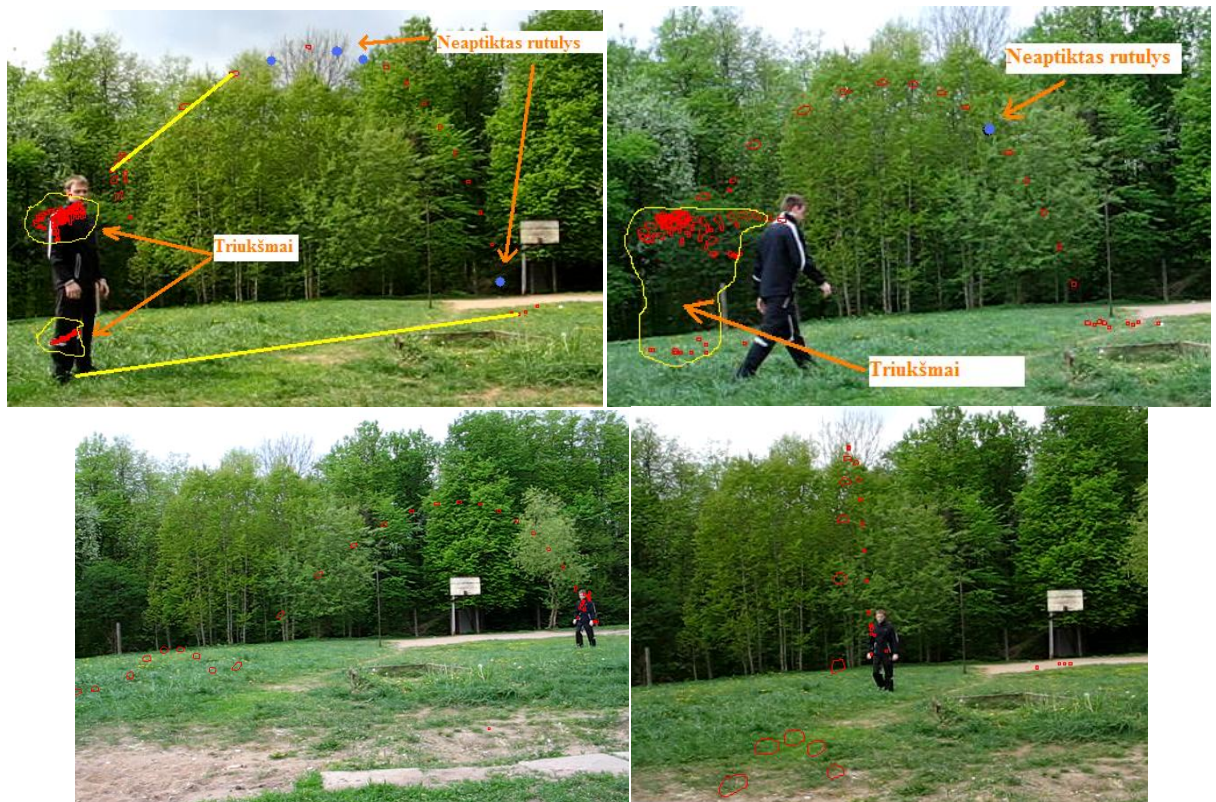
2 lentelė. Vaizdo įrašų analizės trukmė

Failo pavadinimas	Vaizdo įrašo analizės trukmė			
	Bandymas 1	Bandymas 2	Bandymas 3	Vidurkis
MVI_0001	8s 769ms	8s 752ms	8s 688ms	8s 736ms
MVI_0002	5s 329ms	5s 413ms	5s 805ms	5s 516ms
MVI_0003	7s 572ms	7s 296ms	7s 465ms	7s 444ms
MVI_0004	7s 442ms	7s 775ms	7s 540ms	7s 586ms
MVI_0005	6s 373ms	6s 620ms	6s 351ms	6s 448ms
MVI_0006	7s 488ms	7s 697ms	7s 588ms	7s 591ms
MVI_0007	4s 492ms	4s 339ms	4s 573ms	4s 468ms
MVI_0008	6s 902ms	5s 756ms	6s 607ms	6s 422ms
MVI_0009	5s 844ms	6s 194ms	6s 165ms	6s 068ms

Po analizės trukmės tyrimo, buvo tiriamas programos sugebėjimas fiksuoti išmestą rutulį. Tyrimo metu, rutulio fiksavimas buvo skaičiuojamas nuo rutulio išmetimo momento iki rutulys pasiekia žemę. 3 lentelėje pateikiami rutulio fiksavimo lyginamieji duomenys. Rutulio aptikimo skaičiavimo pavyzdžiai pateikiami 1 ir 2 prieduose.

3 lentelė. Rutulio aptikimo kiekis

Failo pavadinimas	Rutulio atvaizdavimo kiekis vaizdo įrašė	Rutulio aptikimo kiekis programos veikimo metu	Rutulio aptikimas procentais
MVI_0001	18	15	83,33%
MVI_0002	21	21	100,00%
MVI_0003	20	18	90,00%
MVI_0004	21	21	100,00%
MVI_0005	17	16	94,12%
MVI_0006	24	23	95,83%
MVI_0007	16	16	100,00%
MVI_0008	24	22	91,67%
MVI_0009	23	22	95,65%
<b>Bendras programos rutulio aptikimas procentais:</b>			<b>94,51%</b>



18 pav. Rutulio metimo trajektorijos analizė

Analizuojant 18 pav. gautus rezultatus, matome, kad algoritmas ne visada užfiksuoja mėtomą rutulį tam tikrose situacijose ir nepašalina visų triukšmų.

Siekiant sumažinti atvaizduojamus triukšmus, filtre būtų galima didinti objekto srities plotą, tačiau objektų filtras gali nufiltruoti atpažintą rutulį kai jis yra numetamas į tolį ir rutulio kontūro srities plotas pasidaro mažas. Taip pat atliekant išmetimo veiksmą, rutulys yra fiksuojamas ir judančios sportinės aprangos baltos juostelės sukuria triukšmus, kurie suformuoja objektą panašų į elipsės formą.

Išmetus rutulį atpažinimas kartais neįvyksta, nes supanašėjus rutulio ir aplinkos spalvų tonams, dėl ryškumo trūkumo, vieta kur randasi rutulys suformuojama kaip maži triukšmai, todėl po segmentavimo, atlikus morfologines operacijas, algoritmas sritį pažymi kaip nepakitusią.

#### **5.4. Išvados**

Atlikus eksperimentus, buvo nustatytas programos rutulio aptikimo procentas. Kadangi egzistuoja rutulio sekimo klaidų, tai bendras rutulio aptikimo procentas yra iki 94,51%. Vaizdo įrašo apdorojimo trukmė užtrunka 1,4 karto ilgiau nei vaizdo įrašo trukmė.

Pastebėta, kad triukšmų šalinimo efektyvumas turėtų būti gerinamas. Daugiausia triukšmų pasitaiko rutulio išmetimo metu. Taip pat sekimo algoritmas turi trūkumų, kurių metu yra neatpažįstamas rutulys.

## 6. IŠVADOS

1. Vaizdo įrašų apdorojimas plačiai tiriamas moksliniuose darbuose ir ypač daug dėmesio skiriama analizuoti sporto vaizdo įrašus.
2. Analizės metu apžvelgus egzistuojančius mokslinius sprendimus, mėtomam rutuliui aptikti buvo pasirinktas kadrų skirtumo metodas. Šio metodo vienas svarbiausių kriterijų, kad kamera laikoma fiksuotoje padėtyje.
3. Tinkamo judančio vaizdo objekto atrinkimui buvo sukurtas filtras, kuris filtruoja rastus objektus pagal kontūro srities plotą ir objekto formą. Kadangi išmestas rutulys, filmuojant, gali prarasti savo formą, todėl objektai filtruojami pagal elipsės formą.
4. Programa buvo realizuota C# programavimo kalba, panaudojant EMGU .NET įvynioklį, skirtą OpenCV vaizdo apdorojimo bibliotekai.
5. Eksperimentų metu nustatyta, kad „Judančių vaizdo objektų sekimas ir analizė“ programos efektyvumas aptikti išmestą rutulį yra iki 94,51% ir vaizdo įrašo analizė užtrunka 1,4 karto ilgiau nei vaizdo įrašo trukmė.
6. Sukurto metodo tobulinimo kryptys:
  - 6.1. Atsirandančių triukšmų šalinimo efektyvumo gerinimas;
  - 6.2. Mėtomo rutulio atvaizdavimas rutulio judėjimo trajektorijoje nuspėjimo būdu, kai rutulys yra neaptinkamas objektų segmentavimo algoritmo;
  - 6.3. Objektų segmentavimo ir filtravimo algoritmų tobulinimas.

## 7. LITERATŪRA

- [1] BAJRAMOVIC, F.; DEUTSCH, B; GRABL, CH; DENZLER, J, *Efficient Adaptive Combination of Histograms for Real-Time Tracking*. EURASIP Journal on Image and Video Processing Volume 2008, Article ID 528297, p: 11
- [2] BUGEAU, A.; PEREZ, P., *Track and Cut: Simultaneous Tracking and Segmentation of Multiple Objects with Graph Cuts*. EURASIP Journal on Image and Video Processing Volume 2008, Article ID 317278, p: 14
- [3] CIGDEM EROGLU ERDE; MURAT TEKALP, A.; BÜLENT SANKUR, *Video Object Tracking With Feedback of Performance Measures*. // IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 4, 2003
- [4] HUA-TSUNG CHEN; HSUAN-SHEN CHEN; SUH-YIN LEE, *Physics-Based Ball Tracking in Volleyball Videos with its Applications to Set Type Recognition and Action Detection*. // Acoustics, Speech and Signal Processing, IEEE International Conference on Volume 1, 2007, p: I-1097 - I-1100
- [5] HUA-TSUNG CHEN; HSUAN-SHENG CHEN; MING-HO HSIAO; WEN-JIIN TSAI; SUH-YIN LEE, *A Trajectory-Based Ball Tracking Framework with Visual Enrichment for Broadcast Baseball Videos*. // Journal of Information Science and Engineering 24, 2008, p. 143-157
- [6] HUA-TSUNG CHEN; MING-CHUN TIEN; YI-WEN CHEN; WEN-JIIN TSAI; SUH-YIN LEE, *Physics-based ball tracking and 3D trajectory reconstruction with applications to shooting location estimation in basketball video*. // Journal of Visual Communication and Image Representation, 2009, Volume 20 , Issue 3, p. 204-216, ISSN:1047-3203
- [7] JINCHANG REN; JAMES ORWELL; GRAEME A. JONES, *Estimating the Position of a Football from Multiple Image Sequences*. // One Day Symposium on Spatiotemporal Image Processing, Royal Statistical Society, London, 2004
- [8] JINCHANG REN; ORWELL, J.; JONES, G.A.; MING XU, *A general framework for 3D soccer ball estimation and tracking* // International Conference on Image Processing, 2004, Volume 3, p: 1935 – 1938
- [9] MARCOTEGUI, B.; ZANOQUERA, F.; CORREIA, P.; ROSA, R.; MARQUES, F.; MECH, R.; WOLLBORN, M., *A video object generation tool allowing friendly user interaction*. // International Conference on Image Processing, 1999, Volume 2, p: 391 – 395



- [10] MCIVOR M., A., *Background Subtraction Techniques*. // Image and Vision Computing, Hamilton, New Zealand, 2000.
- [11] ROST BRADSKI, G; KAEHLER, A. *Learning OpenCV: Computer Vision with the OpenCV Library*. 2008. ISBN 10: 0-596-51613-4 | ISBN 13: 9780596516130
- [12] SENIOR, A., *Tracking people with probabilistic appearance models*. // Proceedings of the ECCV Workshop on Performance Evaluation of Tracking and Surveillance Systems, 2002, p: 48-55.
- [13] SENIOR, A; HAMPAPUR, A; YING-LI TIAN; BROWN, L; PANKANTI, S; BOLLE, R., *Appearance models for occlusion handling*. // Image and Vision Computing In Performance Evaluation of Tracking and Surveillance, 2006, Vol. 24, No. 11., p: 1233-1243.
- [14] THUNDER ROCKS, *Object tracking*. // Engineering/Technical seminars. Prieiga per internetą <http://enggseminars.uuuq.com/data/new/Object%20tracking.rar>
- [15] TSECHPENAKIS, G.; AKRIVAS, G.; ANDREOU, G.; STAMOU, G.; KOLLIAS, S., *Knowledge – Assisted Video Analysis and Object Detection*. // Proceedings of European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems, konferencija, 2002. Prieiga per internetą: <http://www.image.ece.ntua.gr/php/savepaper.php?id=208> [žiūrėta 2009-01-20]
- [16] VERGES, J.; ARANDA, J.; SANFELIU, A., *Object tracking system using colour histograms*. Spanish symposium on Pattern recognition and Image analysis, p:225-230, 2001, ISBN:84-8021-351-5
- [17] WANG, Y.; DOHERTY, J. F.; VAN DYCK, R. E., *Moving Object Tracking in Video*. // Proceedings of the 29th Applied Imagery Pattern Recognition Workshop, 2000, p: 95, ISBN:0-7695-0978-9
- [18] WELCH, G.; BISHOP, G., *An Introduction to the Kalman Filter*. // Technical Report: TR95-041, 1995
- [19] XINGUO YU; FARIN, D., *Current and Emerging Topics in Sports Video Processing*. // IEEE International Conference on Multimedia and Expo, 2005, p: 526- 529, ISBN: 0-7803-9331-7.
- [20] YOKOYAMA, M.; POGGIO, T., *A contour-based moving object detection and tracking*. // 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, p: 271- 276, ISBN: 0-7803-9424-0

## 8. TERMINŲ IR SANTRAUKŲ ŽODYNAS

**AVI** ( angl. Audio Video Interleave) – įvairialypės terpės garso ir vaizdo formatas

**.NET** – programinės įrangos karkasas skirtas Microsoft Windows operacinėms sistemoms. Jis sudarytas iš didelės suprogramuotų sprendimų bazės, skirtos išvengti dažniausiai pasitaikančių programavimo sprendimų.

**EMGU** – skirtingų platformų (angl. cross platform) OpenCV vaizdo apdorojimo bibliotekos

**.NET** įvynioklis (angl. wrapper) leidžiantis iškviešti OpenCV funkcijas suderinamose .NET kalbose: C#, VB, VC++, IronPython.

**OpenCV** – atviro kodo kompiuterinė vaizdo biblioteka.

**FPS** – kadrų skaičius per sekundę

**2D** – dviejų dimensijų vaizdas

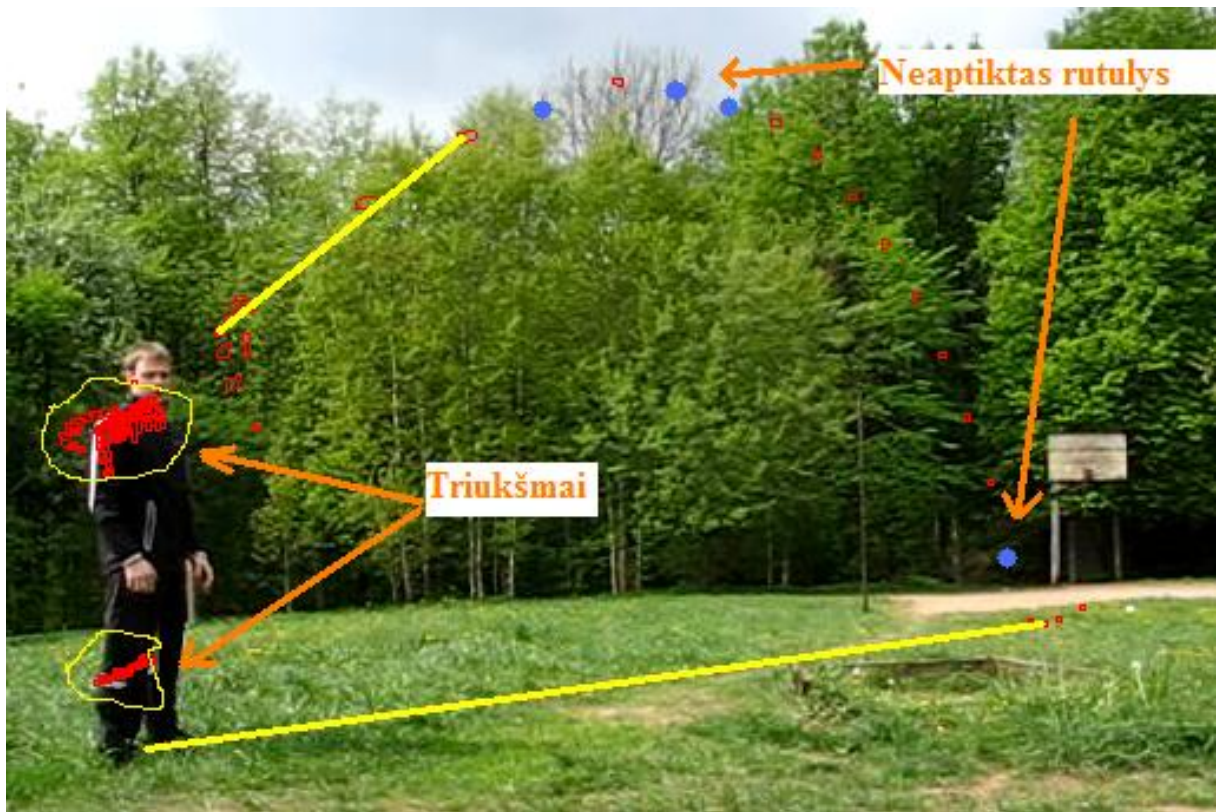
**3D** – trijų dimensijų vaizdas

**FDI** – kadrų diferencijavimo vaizdas (angl. Frame Difference Image)

**PFDI** – teigiamas kadrų diferencijavimo vaizdas (angl. Positive Frame Difference Image)

**PRIEDAI**

1 PRIEDAS. MVI\_0001 vaizdo įrašo analizė





Vaizdo kadrai išdėstyti eiliškumu iš kairės į dešinę

1 → 2

3 → 4

5 → 6





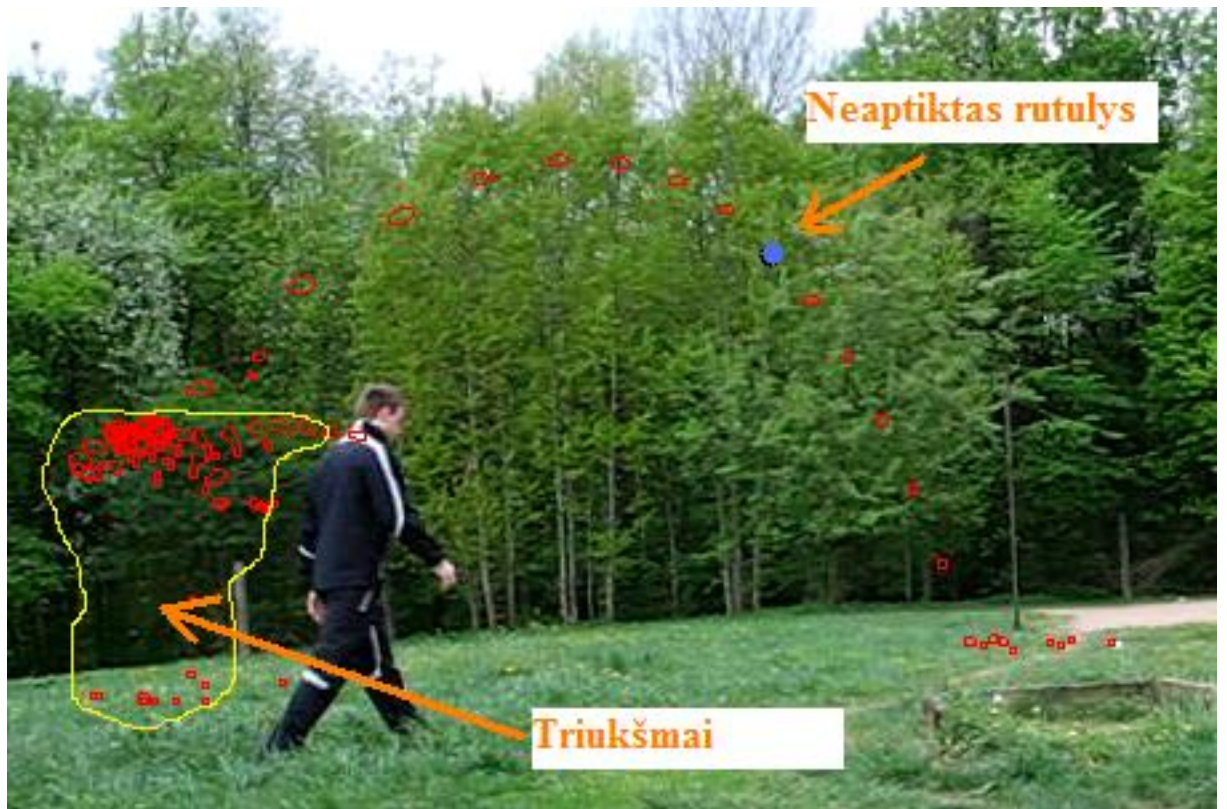








2 PRIEDAS. MVI\_0008 vaizdo įrašo analizė





Vaizdo kadrai išdėstyti eiliškumu iš kairės į dešinę

1 → 2

3 → 4

5 → 6









