



**Kauno technologijos universitetas**  
Elektros ir elektronikos fakultetas

**Vaizdo apdorojimo algoritmų laukiniams gyvūnams atpažinti  
tyrimas ir realizavimas įterptinėje sistemoje**  
Baigiamasis magistro studijų projektas

---

**Aistė Kukanauskaitė**  
Projekto autorė

**Doc. Mindaugas Knyva**  
Vadovas

---

**Kaunas, 2023**



**Kauno technologijos universitetas**  
Elektros ir elektronikos fakultetas

# **Vaizdo apdorojimo algoritmų laukiniams gyvūnams atpažinti tyrimas ir realizavimas įterptinėje sistemoje**

Baigiamasis magistro studijų projektas  
Elektronikos inžinerija (6211EX012)

---

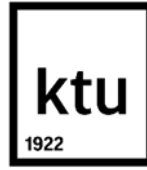
**Aistė Kukanauskaitė**  
Projekto autorė

**Doc. Mindaugas Knyva**  
Vadovas

**Prof. Darius Andriukaitis**  
Recenzentas

---

**Kaunas, 2023**



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

Aistė Kukanaukaitė

## **Vaizdo apdorojimo algoritmų laukiniams gyvūnams atpažinti tyrimas ir realizavimas įterptinėje sistemoje**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama kitų asmenų autoriaus ar kitų teisių, laikydamasi Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjusi;
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Aistė Kukanaukaitė

*Patvirtinta elektroniniu būdu*

Kukanauskaitė, Aistė. Vaizdo apdorojimo algoritmų laukiniams gyvūnams atpažinti tyrimas ir realizavimas įterptinėje sistemoje. Magistro baigiamasis projektas / vadovas doc. Mindaugas Knyva; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Elektronikos inžinerija, inžinerijos mokslai.

Reikšminiai žodžiai: gyvūnų aptikimo sistemos, termovizorius, kompiuterinė vizija.

Kaunas, 2023. 55 p.

### **Santrauka**

Laukinių gyvūnų stebėjimo ir aptikimo sistemos yra plačiai naudojamos norint tirti gyvūnų migracijos kelius, aplinkos jiems daromą įtaką, saugoti žmonių gyvenamas arba pavojingas teritorijas nuo laukinių gyvūnų, užkirsti kelią autoįvykiams ir kitoms nelaimėms. Stebėti ir aptikti gyvūnus galima ir tokiomis sistemomis, kurių principas – kameromis nuskaitomų vaizdų apdorojimas (1.2 poskyris). Kamera gaunamų vaizdų apdorojimu paremtos sistemos yra neinvazinės, joms nereikia papildomų prietaisų blokų ar stotelių (1.4 poskyris). Darbe bus tiriami ir projektuojami algoritmai gyvūnų aptikimui bei pasirinktas algoritmas realizuojamas įterptinėje sistemoje.

Algoritmų, skirtų apdoroti nuotraukas, tyrimui ir kūrimui panaudota Kauno Technologijos Universiteto, Elektros ir elektronikos fakulteto mokslininkų kurta sistema, aprašyta 2.1 poskyryje. Viena iš sistemos dalių – termovizorine kamera – gautos nuotraukos, kurių paruošimas tolimesniems darbo žingsniams aprašomas 2.2 poskyryje. Išanalizuoti ir suprojektuoti keturi algoritmai, skirti nuotraukose aptikti gyvūnų objektus, aprašomi 3 skyriuje. Pagal parametrus, aprašytus 4.1 poskyryje, ištirtos ir palygintos skirtingos metodikos; metodikų analizė ir vienos metodikos pasirinkimas perkėlimui į įterptinę sistemą aprašytas 4.2 skyriuje. Pasirinkta metodika, paremta judesio fiksavimu.

Pasirinkta metodika realizuota įterptinės sistemos mikrovaldiklyje; dalis metodikos žingsnių akceleruota implementavus juos FPGA plokštėje. Ištirtos ir palygintos skirtingų metodikų, įgyvendintų tiek mikrovaldikliuose, tiek įterptinėje sistemoje, realizavimo sąlygos bei algoritmų vykdymo trukmės, tyrimų rezultatai pateikiami 5 skyriuje. Taip pat atliktas pasirinktos metodikos tyrimas skirtingais nuotraukų rinkiniais.

Kukanauskaitė, Aistė. Image Processing Algorithms for Animal Detection Research and Realization in Embedded System. Master's Final Degree Project / supervisor Assoc. Prof. Mindaugas Knyva. Faculty of Electricity and Electronics. Kaunas University of Technology.

Study field and area (study field group): Electronic Engineering, Engineering Sciences.

Keywords: animal detection systems, thermovisor, computer vision.

Kaunas, 2023. 55 p.

### **Summary**

Wild animal monitoring and detection systems are widely used to study animal migration routes and the effects of the environment on them, to protect human-inhabited or dangerous areas from wild animals, and prevent car accidents and other disasters. Monitoring and detection of animals is also possible with such systems that process images scanned by cameras (subsection 1.2). Systems based on processing of images received by the camera are non-invasive, do not require processing device blocks or stations (subsection 1.4). In this work, algorithms for animal detection will be researched and designed as well as the selected algorithm will be implemented in an embedded system.

The system developed by the scientists of Kaunas University of Technology, Faculty of Electricity and Electronics, described in subsection 2.1, was used for the research and development of algorithms for processing images. One of the parts of the system – a thermal imaging camera – was used to capture images, which were processed for further work as described in subsection 2.2. Four algorithms for detecting animal objects in images were analyzed and designed, as depicted in Section 3. According to parameters described in subsection 4.1, different methodologies were investigated and compared; the analysis of techniques and the selection of one methodology for transfer to the embedded system is described in subsection 4.2. The chosen methodology is based on motion capture.

The selected methodology was implemented on a microcontroller of the embedded system; some of the steps of the methodology were accelerated by implementing them on the FPGA board. The actualization details of different methodologies, implemented both in microcontrollers and embedded systems, as well as algorithms execution times, were examined and compared in Section 5. A study of the chosen methodology with different photo sets was also carried out.

## Turinys

<b>Lentelių sąrašas .....</b>	<b>7</b>
<b>Paveikslų sąrašas .....</b>	<b>8</b>
<b>Santrumpų ir terminų sąrašas .....</b>	<b>10</b>
<b>Įvadas.....</b>	<b>11</b>
<b>1. Laukinių gyvūnų atpažinimo ir stebėjimo metodai ir sistemos .....</b>	<b>12</b>
1.1. Spindulio nutraukimo metodas .....	12
1.2. Dengiamo ploto metodas .....	13
1.3. Kitokio tipo sistemos .....	17
1.4. Apibendrinimas .....	24
<b>2. Duomenų atrinkimas ir paruošimas .....</b>	<b>25</b>
2.1. Duomenų surinkimo sistema .....	25
2.2. Duomenų atrinkimas ir paruošimas.....	27
<b>3. Metodikos gyvūnams aptikti nuotraukose .....</b>	<b>29</b>
3.1. Metodika, paremta filtracija dvišaliu filtru.....	29
3.2. Metodika, paremta Canny kraštų radimo algoritmu.....	32
3.3. Metodika, paremta spalvų kvantavimu artimiausio vidurkio algoritmu .....	35
3.4. Metodika gyvūnams aptikti nuotraukų rinkinyje, paremta judesio aptikimu.....	37
3.5. Metodikos pasirinkimas.....	40
<b>4. Metodikos realizavimas eLinux sistemoje.....</b>	<b>44</b>
4.1. Metodikos realizavimas įterptinės sistemos mikroprocesoriuje.....	44
4.2. Metodikos dalių akceleravimas FPGA .....	46
4.3. Metodikos tyrimai, aptinkant gyvūnus įvairiose nuotraukose.....	49
<b>Išvados .....</b>	<b>52</b>
<b>Literatūros sąrašas .....</b>	<b>53</b>
<b>Priedai.....</b>	<b>56</b>
1 priedas. Metodikos, paremtos dvišaliu filtravimu, algoritmas .....	56
2 priedas. Metodikos, paremtos Canny kraštų radimo algoritmu, algoritmas.....	58
3 priedas. Metodikos, paremtos spalvų kvantavimu, algoritmas.....	59
4 priedas. Metodikos, paremtos judesio aptikimu, algoritmas .....	60

## Lentelių sąrašas

<b>1 lentelė.</b> Vairuotojų greičio apibūdinimo lentelė priklausomai nuo aliarmo švyturių būsenos [7].	15
<b>2 lentelė.</b> Metodikų tyrimo parametrais rezultatus apibūdinanti lentelė .....	42
<b>3 lentelė.</b> Personalinio kompiuterio ir PYNQ plokštės specifikacijų palyginimo lentelė.....	44
<b>4 lentelė.</b> Metodikų vykdymo laikų trukmių lentelė .....	49

## Paveikslų sąrašas

<b>1 pav.</b> Spindulio nutraukimo sistema .....	12
<b>2 pav.</b> Dengiamo ploto sistemos modelis [14] .....	13
<b>3 pav.</b> Dramblius atbaidyti skirtos sistemos modelis [4] .....	14
<b>4 pav.</b> Didelių gyvūnų aptikimo sistemos veikimo principo aiškinamasis brėžinys [7] .....	15
<b>5 pav.</b> Gyvūno aptikimas: a) bepilotis dronas virš ir už sekamo objekto; b) terminė nuotrauka; c) terminis paveikslas po apdorojimo sukurtu algoritmu [12] .....	16
<b>6 pav.</b> Grafinės vartotojo sąsajos vaizdas bepiločio drono skridimo metu [12] .....	16
<b>7 pav.</b> Jutiklių duomenų klasifikavimo modeliai: a) akustinio sensoriaus; b) seisminio sensoriaus [13] .....	18
<b>8 pav.</b> Mikro doplerio signatūra nuo atskirų kūno dalių: a) žmogaus; b) arklio [13] .....	19
<b>9 pav.</b> Jutiklių tinkle sistemos veikimo modelis [13] .....	19
<b>10 pav.</b> Duomenų gavimo, perdavimo tarp nešlių ir siuntimo sistemos modelis [5] .....	20
<b>11 pav.</b> Sistemos architektūros dalys: a) jutimo prietaisų; b) sekimo prietaisų [15] .....	21
<b>12 pav.</b> Galios suvartojimo praktinis tyrimas 12 valandų eigoje [15] .....	22
<b>13 pav.</b> Atpažinimo sistemos modelis [16] .....	23
<b>14 pav.</b> Kelio atkarpoje Kaunas – Šakiai išdėstyta duomenų surinkimo sistema .....	25
<b>15 pav.</b> Stulpelio viršutinė dalis su stogeliu apsaugai nuo kritulių .....	25
<b>16 pav.</b> Pagrindinio sistemos prietaisų bloko, pritaisyto prie stulpo, vaizdas .....	26
<b>17 pav.</b> Duomenų surinkimo sistemos principinė schema .....	26
<b>18 pav.</b> Terminėmis kameromis gautos nuotraukos: a ir b – su kamera, pritaisyta prie pagrindinio sistemos prietaisų bloko, c ir d – su kamera, pritaisyta prie antrinio prietaisų bloko .....	27
<b>19 pav.</b> Apkirptos terminėmis kameromis gautos nuotraukos: a ir b – su kamera, pritaisyta prie pagrindinio sistemos prietaisų bloko, c ir d – su kamera, pritaisyta prie antrinio prietaisų bloko ...	28
<b>20 pav.</b> Dvišaliu filtru paremtos metodikos algoritmas .....	29
<b>21 pav.</b> Dvišalio filtro metodikos įgyvendinimo žingsniai .....	32
<b>22 pav.</b> Canny algoritmu paremtos metodikos algoritmas .....	32
<b>23 pav.</b> Canny algoritmo metodikos įgyvendinimo žingsniai .....	35
<b>24 pav.</b> Spalvų kvantavimo algoritmu paremtos metodikos algoritmas .....	35
<b>25 pav.</b> Spalvų kvantavimo metodikos įgyvendinimo žingsniai .....	37
<b>26 pav.</b> Judesio aptikimu paremtos metodikos algoritmas .....	38
<b>27 pav.</b> Judesio aptikimo metodika analizuoti kadrai .....	40
<b>28 pav.</b> Metodikų tyrimo pagal metrikų įverčius grafikas .....	42
<b>29 pav.</b> Metodikų tyrimo pagal laiką grafikas .....	42
<b>30 pav.</b> Metodikų algoritmo trukmės skirtinguose procesoriuose palyginamasis grafikas .....	44
<b>31 pav.</b> Metodikų algoritmo trukmės skirtinguose procesoriuose, kai nevykdoma gyvūnų objektų aptikimo dalis, palyginamasis grafikas .....	45
<b>32 pav.</b> Metodikų algoritmo trukmės skirtinguose procesoriuose, kai vykdoma ir kai nevykdoma gyvūnų objektų aptikimo dalis, palyginamasis grafikas .....	45
<b>33 pav.</b> Apibendrintas judesio aptikimu paremtos metodikos algoritmas akceleracijai su FPGA ...	46
<b>34 pav.</b> xOpenCV bibliotekos veikimą nusakanti schema [35] .....	47
<b>35 pav.</b> Metodikos algoritmo trukmės neatliekant buferizacijos ir buferizuojant skirtingus procesus palyginamasis grafikas .....	47
<b>36 pav.</b> Filtravimo operacijos trukmės negreitinant, buferizuojant ir greitinant bei buferizuojant palyginamasis grafikas .....	48



<b>37 pav.</b> Objektų išplėtimo operacijos trukmės negreitinant, buferizuojant ir buferizuojant bei greitinant palyginamasis grafikas .....	48
<b>38 pav.</b> Metodikų algoritmo trukmės skirtinguose procesoriuose ir FPGA, nevykdant gyvūnų objektų aptikimo dalies, palyginamasis grafikas .....	49
<b>39 pav.</b> Gyvūnų objektų tyrimo, atpažįstant juos skirtingose situacijose paremta judesiu metodika, rezultatai .....	50
<b>40 pav.</b> Metodikos, paremtos judesiu, tyrimo skirtingais duomenų rinkiniais rezultatai .....	51

## Santrumpų ir terminų sąrašas

### Santrumpos:

IR – infraraudonieji spinduliai.

PIR – pasyvinis infraraudonųjų spindulių jutiklis.

### Terminai:

**E-Linux** – įterptinė sistema, kurios procesoriuje veikia speciali įterptinė Linux operacinė sistema.

**Formantai** – tai smailios viršūnės signalo spektre, turinčios aukštą energijos lygį, ypač pastebimos analizuojant žmogaus tariamų balsių signalus.

**Gauso (MVG) klasifikatorius** – Gauso tikimybių pasiskirstymo dėsnio pagrindu paremtas nelinearinis mašininio mokymo algoritmas klasifikavimui ir regresijai.

**Delsimui tolerantiškų tinklų** (angl. *Delay Tolerant Networking, DTN*) **technologija** – kompiuterių tinklo architektūra, kuri naudojama norint užtikrinti ryšį tinkluose, kurie yra nevienalyčiai ir ne visuomet turi stabilų ryšį.

**Dvišalis filtras** – Gauso filtro veikimu pagrindu paremtas filtras, turintis papildomą dedamąją komponentę – pikselių intensyvumo skirtumo ištraukimą ir filtravimo skaičiavimą.

**Euklidinis (pikselių) atstumas** – tai yra tiesioginis atstumas tarp dviejų pikselių linija.

**Radiometriniai skirtumai** – itin maži skirtumai, tam tikroje skalėje nusakantys jos nevientisumą.

**Požymių detektacija** (angl. *feature detection*) – procesas, kurio metu neuroninė sistema įvertina jai suteiktą informaciją tam, kad suprastų esmines užuominas apie objektus ir gebėtų jas skirti nuo objektų gaubiančio triukšmo ir aplinkos.

**Požymių apibrėžimas** (angl. *feature description*) – procesas, kurio metu apibrėžiami objektą nusakantys požymiai, unikalūs tam objektui.

**Konvoliucija** (angl. *convolution*) – tai yra dviejų funkcijų operacija, kuria gaunama trečia funkcija, nusakanti, kaip kinta pirmoji funkcija antrosios atžvilgiu.

**Supremumas** (angl. *supremum*) – tai yra poaibiui priklausantis skaičius, esantis didesnis arba lygus visiems duotos aibės poaibio skaičiams.

**Intensyvus skaičiavimų algoritmas** (angl. *computationally intensive algorithm*) – tai algoritmas, kuriam įvykdyti ir tinkamai funkcionuoti reikalingas palyginti didelis kiekis operacijų.

**Morfologinės operacijos** – tai yra vaizdų apdorojimo operacijos, kurios apdoroja vaizdus priklausomai nuo juose esančių objektų formos.

**Segmentavimo operacijos** – tai yra vaizdų apdorojimo operacijos, kurių metu paveikslai yra dalinami į skirtingas zonas priklausomai nuo pikselių charakteristikų tam, kad būtų galima atpažinti objektus, jų ribas ir lengviau bei efektyviau analizuoti vaizdus.

## Įvadas

Daugelį metų laukinių gyvūnų stebėjimo ir aptikimo sistemos buvo ir yra naudojamos norint stebėti gyvūnų elgesį dėl įvairių priežasčių: sužinoti bei tirti gyvūnų migracijos kelius ir būdus bei aplinkos jiems daromą įtaką, saugoti žmonių gyvenamas arba pavojingas teritorijas nuo laukinių gyvūnų, užkirsti kelią autoįvykiams ir kitoms nelaimėms, kurių priežastis – žmogaus ar mašinos susidūrimas su laukiniu gyvūnu [1] [2]. Dėl šių priežasčių yra sukurta daug gyvūnų stebėjimo, aptikimo sistemų, kurias gali sudaryti sensorių, bevielio ryšio modulių, algoritmams vykdyti skirtų procesorių ir įspėjamųjų ženklų, dinamiškai informaciją perduodančių signalais, tinklas [3] [4] [5]. Šių sudėtingų sistemų egzistavimas pagerina žmonių, gyvenančių ar keliaujančių šalia laukinių gyvūnų teritorijų, gyvenseną, padeda mokslininkų tyrimams zoologijos moksle [4] [6] [7]. Todėl buvo pasirinkta sukurti laukinių gyvūnų aptikimo algoritmą, realizuojamą E-Linux platformoje tam, kad būtų įgyvendinamas sistemos mobilumas ir ši ateityje galėtų padėti vairuotojams sužinoti apie gresiantį susidūrimą su laukiniu gyvūnu, taip prisidedant prie didesnio vairuotojų saugumo keliuose.

**Darbo tikslas** – sukurti vaizdo apdorojimo algoritmą, skirtą aptikti laukinius gyvūnus, ir patalpinti jį *E-Linux* sistemoje.

### Uždaviniai:

- atlikti laukinių gyvūnų stebėjimo ir aptikimo metodikų apžvalgą;
- aprašyti pasirinkto gyvūnų stebėjimo ir aptikimo metodo įrangą;
- suprojektuoti ir įgyvendinti vaizdo apdorojimo algoritmą;
- realizuoti vaizdo apdorojimo algoritmą E-Linux sistemoje.

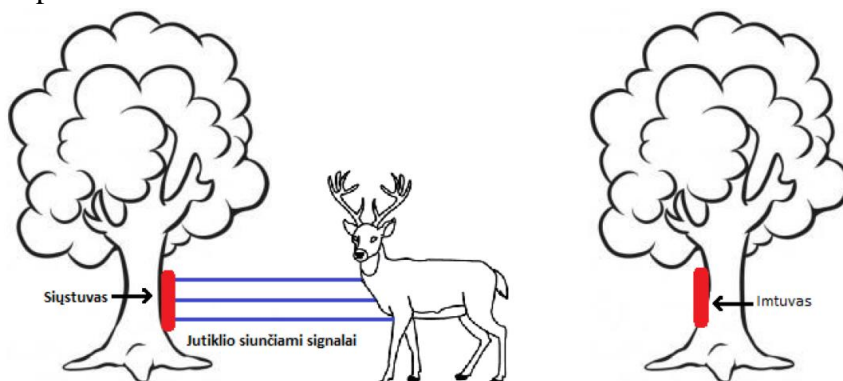
## 1. Laukinių gyvūnų atpažinimo ir stebėjimo metodai ir sistemos

Kiekvienais metais laukinių gyvūnų ir automobilių susidūrimų keliuose kiekis didėja visame pasaulyje [8] [9]. Nors Covid-19 pandemijos pradžioje įvairiose valstybėse dėl jų taikomos politikos ir sankcijų ligą suvaldyti laukinių gyvūnų ir automobilių susidūrimų skaičiai buvo sumažėję, pandemijai progresuojant tokių tipų avarių kiekis vėl pradėjo augti. Tai viena iš priežasčių, kodėl yra kuriamos avarines situacijas minimizuoti skirtos sistemos: jos gali saugoti konkrečias laukinių gyvūnų rūšis nuo išnykimo dėl dažno šių rūšių gyvūnų patekimo į važiuojamąsias kelio dalis, taip pat yra plačiąja prasme skirtos stebėti ir aptikti laukinius gyvūnus [10] [3]. Yra svarbu gebėti įspėti automobilyje esantį žmogų apie galimą susidūrimo pavojų su laukiniu gyvūnu, nes taip galima sumažinti tiek pačių vairuotojų stresą važiuojant avaringuose kelio ruožuose, tiek ir sukliudyti nelaimingesniems atsitikimams, galintiems baigtis mirtimi.

Mokslinėse publikacijose, tiriančiose bei pasiūlančiose gyvūnų stebėjimo ir aptikimo technologijas, jas galima suskirstyti į spindulio nutraukimo [3] arba dengiamo ploto metodikas naudojančias sistemas [1]. Spindulio nutraukimo metodikoje naudojami siųstuvai ir imtuvai, turintys mikrobangų, infraraudonųjų spindulių arba lazerinius jutiklius [3] [11], o dengiamo ploto metoduose naudojami ir apdorojami aktyvinių mikrobangų, aktyvinių ir pasyvinių infraraudonųjų spindulių jutiklių signalai [4] [7] arba apdorojami kamera gaunami vaizdai [12].

### 1.1. Spindulio nutraukimo metodas

Vienas iš paprasčiausių gyvūnų aptikimo būdų yra įdiegti sistemas, kuriose naudojamas spindulio nutraukimo (angl. *break-the-beam systems*) metodas. Tokio tipo sistemos veikimo principas – siųstuvo perduodamas signalas į imtuvą. Sistema aptinka gyvūną, kai jo kūnas blokuoja signalą arba kai imtuvo priimamas signalas užfiksuojamas žemesnis nei nustatyto lygio. Apibendrintas sistemos vaizdas matomas 1 pav.



1 pav. Spindulio nutraukimo sistema

Kaip galima matyti pavaizduotame paveikslėlyje (žr. 1 pav.), trys jutiklio skleidžiami signalai perduodami siųstuvu ir yra priimami tam tikru atstumu nutolusiu imtuvu. Kai gyvūnas patenka į spindulių sklaidimo tiesę, imtuvas nebegali priimti dalies ar visų signalų, kuriuos riboja gyvūno kūnas. Sistemos „protas“ – procesorius, apdorojantis gaunamus signalus imtuve – gali užtikrinti, kad būtų imtasi atitinkamų veiksmų, kai sužinoma, kad buvo kirsti signalų spinduliai, pavyzdžiui, įjungti aliarmo signalą ar kitaip iškomunikuoti reikiamą informaciją.

Spindulio nutraukimo metodiką naudojančiose sistemose įprastai pasitelkiami pasyviniai PIR, ultragarso jutikliai [11], panaudojami kaip sudėtinės sistemos dalys, viena su kita nesusijusios; taip

pat gali būti panaudojamas įvairių – akustinių, seisminių, ultragarsinių – jutiklių tinklas, iš kurio nuskaitomi jutiklių duomenys apdorojami, jungiami vieni su kitais [13].

Kalbant apie sistemas, kuriose jutikliai panaudojami kaip atskiros sistemos dalys, straipsnyje „Design of Wild Animal Detection and Rescue System with Passive Infrared and Ultrasonic Sensor based Microcontroller“, rašytame Yusman, A. Finawan ir Rusli, apibūdinama sistema, susidedanti iš PIR ir ultragarsinio jutiklių bei mikrovaldiklio, apdorojančio gautus duomenis iš PIR jutiklio ir atsakingo už signalo ultragarsiniam jutikliui generavimą [11]. Minėtame straipsnyje panaudotas vienas ultragarso keitiklis pulsui išsiųsti ir jo aidui priimti ir vienas pasyvinis PIR jutiklis (pasyviniu jutiklis vadinamas dėlto, nes tik priima IR spinduliuotę iš išorės, o pats nespinduliuoja)

Yusman, A. Finawan ir Rusli aprašytoje sistemoje PIR jutiklis naudojamas gyvūnams aptikti, kai šie kerta jutiklio apimamą plotą, o ultragarso jutiklis - gyvūnams atbaidyti. Straipsnyje pasidalinta tyrimu, kurio metu ištirta gyvūnų reakcija į skleidžiamas skirtingo dažnio ultragarso bangas, priklausomai nuo gyvūnų atstumo iki jutiklio. Nustatyta, kad mažiausias atstumas, kuriuo gyvūnai parodė dirglumą ir pasitraukė nuo projektuotos sistemos, kai buvo išspinduliuota 25 kHz banga, yra 15 metrų, o atstumas, kuriuo toliausiai nuo sistemos esantys gyvūnai galėjo būti aptikti PIR jutikliu – 5 metrai.

Apibendrinant spindulio nutraukimo metodiką naudojančias sistemas galima teigti, kad šių sistemų duomenų apdorojimo algoritmų sudėtingumas priklauso nuo sistemoje naudojamų jutiklių kiekio ir paskirties. Šiose sistemose judančių gyvūnų aptikimo linija yra dažniausiai arba visiškai linijinė, tačiau naudojant daugiau nei vieną jutiklį arba jutikliui spinduliuojant signalą kampu, gali būti, kad pačio signalo perdavimo / aptikimo linija tampa platesnė – išplatėja siunčiamo signalo spindulio plotis, dėlto signalus jutikliais galima registruoti ilgesnį laiko tarpą, taip padidinant tikimybę teisingai atpažinti gyvūną.

## 1.2. Dengiamo ploto metodas

Dengiamo ploto metodiką naudojančių sistemų veikimo principas – naudojami jutikliai, kurių aptikimo signalo forma yra kūgio arba aplink jutiklį esančio ovalo formos ir gyvūnai yra aptinkami pasinaudojant gaunamais paveikslėliais, apdorojamais realiu laiku iš kamerų, judesio ir / arba kūno šilumą aptinkančiais davikliais, taip pat jutikliais, registruojančiais jų pačių siūstas ir atsispindėjusias bangas [12] [1] [7].



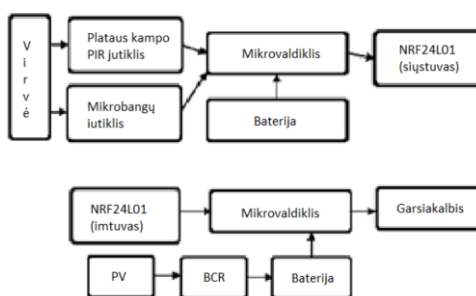
2 pav. Dengiamo ploto sistemos modelis [14]

Kaip galima matyti pavaizduotame modelyje (žr. 2 pav.) pasinaudojama šalia važiuojamosios kelio dalies pastatytais radarais, nukreiptais nuo kelio, kurie aptinka besiantinančius gyvūnus. Kitoje kelio pusėje matomas aliarmo signalas, įjungiamas tuomet, kai radarai aptinka gyvūnus arčiau kelio.

Pasyvinės dengiamo ploto tipo sistemos aptinka gyvūnus tik iš gaunamo signalo, populiariausi pasirinkimai – infraraudonųjų spindulių ir vaizdo kameros [4] [12]. Aktyvinės dengiamojo ploto sistemos siunčia signalą stebimo ploto ribose ir gyvūnams aptikti apdoroja signalo atspindį, paprasčiausias to įgyvendinimas – panaudoti mikrobanginį radarą [7].

Pasyvinę dengiamo ploto sistemą įgyvendino straipsnio „Animal Presence Detection for Elephants and Extruding Method Based on Bee Frequency“ autoriai E. Maulana, C. S. Nugroho ir A. B. Dianisma. Šios sistemos tikslas – apsaugoti prie laukinės gamtos gyvenančius žmones nuo dramblių kaimenių, kurios migracijos, maisto bei vandens paieškos metu įsiveržia į gyvenamas teritorijas ir sukelia neigiamų padarinių žmonėms bei pavojų jų gyvybei [4].

Sistema susideda iš dviejų dalių: dramblių atpažinimo bei bičių simulatoriaus. Sistemos principinis modelis pateikiamas toliau.



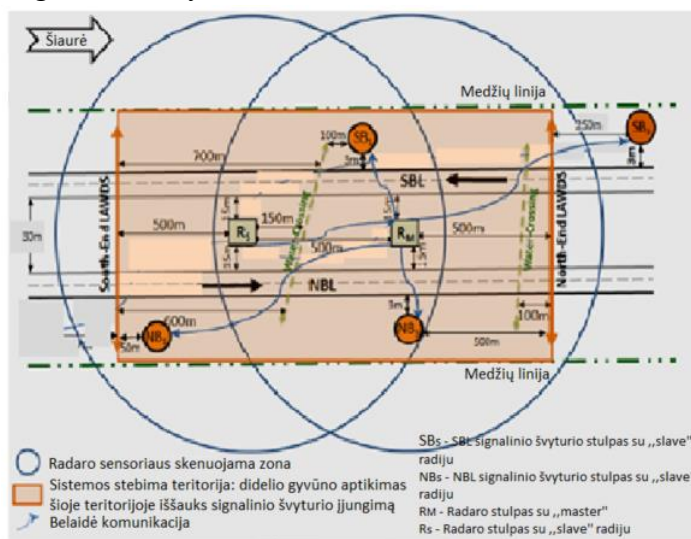
**3 pav.** Dramblius atbaidyti skirtos sistemos modelis [4]

Kaip galima matyti sistemos modelyje, dramblių aptikimo dalis susideda iš plataus kampo PIR (pasyvaus infraraudonųjų spindulių, angl. *PIR* – *Passive InfraRed*) jutiklio bei mikrobangų jutiklio. Belaidis radijo bangų modulis naudojamas duomenims perduoti tarp dramblių aptikimo ir bičių simulatoriaus dalių. Kadangi drambliai yra gyvos būtybės ir skleidžia šilumą, PIR jutiklis panaudojamas aptikti infraraudonuosius spindulius. Skaitmeniniai mikrobangų jutikliai veikia doplerio efekto principu ir aptinka dramblio judėjimą – pasiuntus radijo bangų signalą, jis atsimuša nuo judančio gyvūno ir yra aptinkamas jutiklio, o paties priimto signalo dažnis yra pasislinkęs priklausomai nuo atstumo tarp signalo šaltinio ir gyvūno. Priimti jutiklių duomenys radijo bangomis nusiunčiami į pagrindinį mikrovaldiklį, kuriame yra apdorojami. Apdorojus duomenis ir nustatčius, kad buvo aptiktas dramblys, įjungiami motorinis vibratorius ir garsiakalbis. Iš motoro sklindanti vibracija priverčia bites išskristi iš avilio ir dėl sukeltos agresijos pulti dramblių. Garsiakalbis pagarsina užregistruotą bičių skleidžiamą garsą. Taip drambliai yra labiau atbaidomi nuo patekimo į teritoriją.

Ištestavus sistemą straipsnio autorių pasidalintų rezultatų duomenimis, PIR jutiklio didžiausias registruojamas atstumas, kai aptinkamas gyvūnas, yra 9 metrai, didžiausias kampas jutiklio atžvilgiu – 180° prie 5 metrų, o didžiausias atstumas, kada aptinkamas gyvūnas mikrobanginiu jutikliu – 16 metrų. Ištestavus radijo bangų siuntimo modulį, gauti rezultatai parodė, kad nepažeisti siųstuvo siūsti duomenys gali būti priimti už 200 metrų, jeigu tarp siųstuvo ir imtuvo nėra kliūties. Kitu atveju, kai kliūtis tarp siųstuvo ir imtuvo egzistuoja, duomenys gali būti priimti už mažiau nei 200 metrų ir su didesne klaidų tikimybe.

Aktyvinę dengiamo ploto sistemą suprojektavo ir aprašė A. Mukherjee, S. Stolpner, X. Liu, U. Vrenozaj, C. Fei ir A. Sinha straipsnyje „Large Animal Detection and Continuous Traffic Monitoring on Highways“ [7]. Ši sistema aptinka didelius gyvūnus, tokius, kaip elnius ar briedžius,

prisiartinusius prie važiuojamosios kelio dalies ir įspėja vairuotojus apie gyvūno pasirodymą įjungdama mirksinčius signalinius švyturius.



4 pav. Didelių gyvūnų aptikimo sistemos veikimo principo aiškinamasis brėžinys [7]

Kaip galima matyti pavaizduotame sistemos aiškinamajame brėžinyje (žr. 4 pav.), du jutikliai, kuriems energija tiekama saulės baterijomis, skenuoja 1,5 kilometro ilgio ruožą greitkelyje ir šalia jo, apimamas plotas – iki šalia greitkelio augančių medžių. Kiekviena jutiklio stotelė susideda iš 360° skenuojančio radaro, procesoriaus, apdorojančio duomenis, belaidžio radijo ryšio siųstuvo, skirto įjungti signalinius švyturius, bei saulės kolektorių ir baterijų. Prietaisai, išskyrus radarą, nuo žvarbių oro sąlygų yra apsaugoti atitinkamuose korpusuose, pačios stotelės pritvirtintos greitkelio viduryje esančiose nepervažiuojamose atkarpose tam, kad dėl stotelių įrengimo nereikėtų keisti gamtinės aplinkos šalia greitkelio.

Sistemos algoritmai apdoroja iš radarų gaunamus duomenis apie užfiksuojamus judančių objektų pėdsakus. Programiškai yra išfiltruojami itin trumpi pėdsakai, kuriuos palieka aukštai virš kelio lygio skrendantys paukščiai arba lapai, taip pat dėl nuolatinių eismo būsenos atnaujinimų sistemos programinė įranga sugeba eliminuoti netikro aliarmo signalus, galinčius kilti esant lėtesniam eismui dėl eismo įvykių ar kelio remonto darbų. Apdoroti pėdsakai tuomet yra vertinami pagal tuo metu vyraujančią eismo srautą ir klasifikuojami į automobilių arba gyvūnų pėdsakus.

1 lentelė. Vairuotojų greičio apibūdinimo lentelė priklausomai nuo aliarmo švyturių būsenos [7]

Greičio vidurkis	Kai švyturiai išjungti (km/h)	Kai švyturiai įjungti (km/h)	Absoliutus greičio sumažėjimas (km/h)	Greičio sumažėjimas (%)
Dienos metu	105,58	89,30	16,28	15,4
Nakties metu	110,11	93,58	16,53	15,0

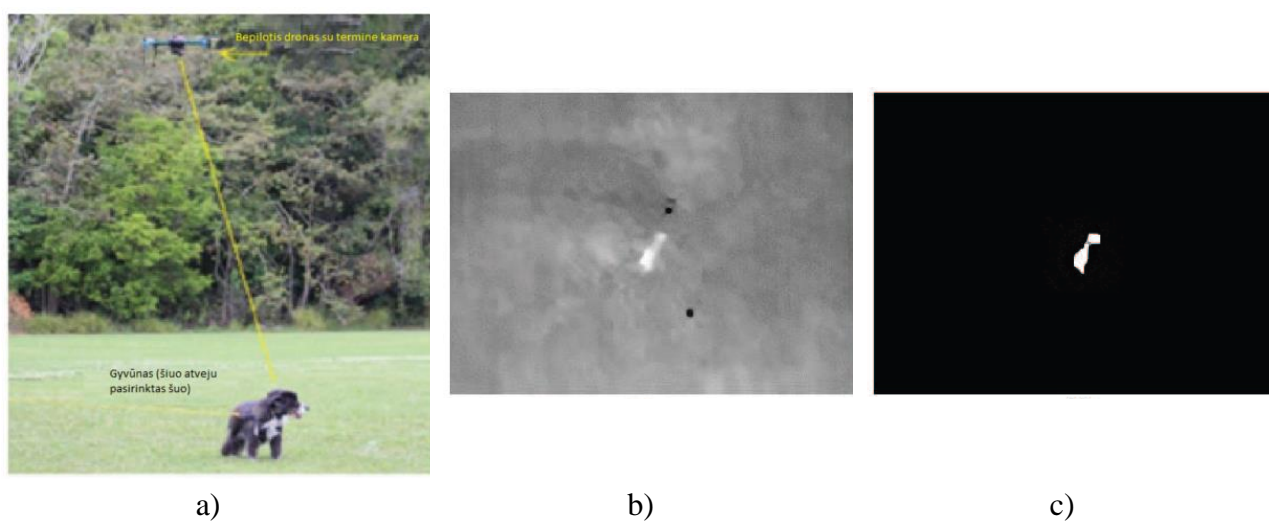
Autorių sistemos eksploatacijos metu atliktas tyrimas, kurio rezultatai matomi 1 lentelėje, parodė, kad signaliniams švyturiams esant išjungtiems, vairuotojai važiuoja apie 15 – 15,4 % greičiau ir sulėtina greitį, kai švyturiai būna įjungti.

Verta paminėti ir pasyvinę sistemą, naudojančią terminę kamerą ir kompiuterinės vizijos algoritmus gyvūnams iš jų aplinkos išskirti, kuri yra aprašoma S. Ward, J. Hensler, B. Alsalam ir L. F. Gonzalez straipsnyje „Autonomous UAVs Wildlife Detection Using Thermal Imaging, Predictive Navigation and Computer Vision“ [12]. Aprašoma sistema geba identifikuoti gyvūno šilumos signalūrą iš anksto



nustatytame aukštyje, nustatyti objekto GPS koordinatės, jas perduoti ir atvaizduoti grafinėje vartotojo sąsajoje realiu laiku.

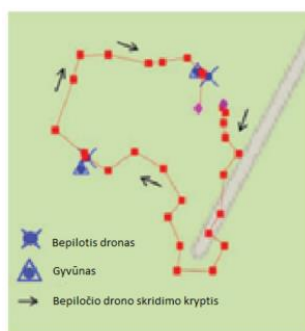
Sistemos architektūra susideda iš dviejų dalių. Desantinė (transportuojama oru) dalis susideda iš keleto motorų turinčio bepiločio orlaivio – drono, autopiloto, terminės kameros, mikrokompiuterio ir GPS modulio, pastarieji keturi elementai išdėstyti ant drono. Paveikslų atpažinimo algoritmas yra įdiegtas į mikrokompiuterio, prie kurio tiesiogiai prijungti kamera ir GPS modelis. Mikroprocesorius gauna GPS koordinatės, kurios perduodamos autopiloto modeliui, ir tuomet kamera užfiksuoti vaizdai skenuojami laukinių gyvūnų atpažinimo algoritmu. Atpažinto gyvūno koordinatės siunčiamos į kontrolės stotelę (angl. Ground Control Station, GCS) tolimesniam apdorojimui. Kai kontrolės stotelėje gaunamos gyvūno koordinatės, programoje atvaizduojamas geografinis žemėlapis su objekto lokacija ir bepiločio drono skridimo taku. Sistema gali veikti tiek prisijungusi prie tinklo, tiek ne (angl. *online ir offline*); dirbant neprisijungus prie tinklo, alternatyvus žemėlapis įkeliamas į mikrokompiuterį grafinės vartotojo sąsajos pagalba.



**5 pav.** Gyvūno aptikimas: a) bepilotis dronas virš ir už sekamo objekto; b) terminė nuotrauka; c) terminis paveikslas po apdorojimo sukurtu algoritmu [12]

5 pav. pateikiama sistema gaunamos gyvūno nuotraukos aptikimas ir apdorojimas – visų pirma nufotografuojama nuotrauka, straipsnio autorių kurtu algoritmu apdorojamos nuotraukos pikselio vertės – panaudojamas metodas pagal specifines ypatybes apskaičiuoja ir atpažįsta „dėmes“ (angl. blobs), kuomet gausus pikselių kiekis su itin panašiomis vertėmis pastebimi vienas šalia kito.

Sistemos testavimo metu tyrimo objektas buvo atpažintas ir jo lokacija nustatyta abu kartus, kai bepilotis dronas skrido virš jo.



**6 pav.** Grafinės vartotojo sąsajos vaizdas bepiločio drono skridimo metu [12]



6 pav. galima matyti tiek tyrimo objekto – šuns – lokaciją abu kartus, kai jis buvo užfiksuotas ir atpažintas, tiek bepiločio drono skridimo kelią. GPS lokacijos tikslumas buvo apibrėžtas 2 metrų spinduliu nuo gyvūno aptikimo vietos.

Apibendrinant aptartas dengiamo ploto tipo sistemas galima teigti, kad dėl sistemose naudojamų jutiklių ir kamerų tipo šios sistemos gali aptikti tiek prie sistemos prisiartinusį gyvūną, tiek iš kurios pusės jis priartėjo, netgi nustatyti atstumą nuo sistemos iki gyvūno. Tačiau dėl sudėtingesnių jutikliais spinduliuojamų ir priimamų signalų, taip pat dėl kameromis gaunamų vaizdų apdorojimo sistemų veikimo modeliai ir principai tampa sudėtingesni, reikia realizuoti tikslesnį duomenų filtravimą ir analizavimą.

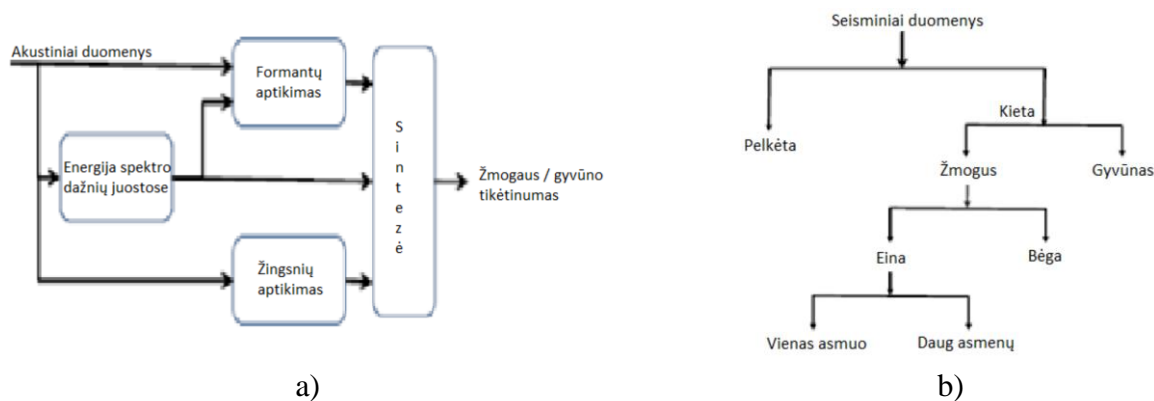
### **1.3. Kitokio tipo sistemos**

Kitokio tipo sistemose gali būti naudojami kabeliai, įkasti po žeme, įvairaus tipo jutikliai, sudarantys jutiklių tinklą, bei ant gyvūno kūno nešiojami prietaisai. Kabelius naudojančiose sistemose po žeme įkasami tam tikro ilgio dvigubo kabelio jutikliai, kurie aptinka didelius ir vidutinio dydžio gyvūnus, kertančius kabelį visame jo ilgyje. Jutiklių tinklų sistemose panaudojant duomenų apdorojimo ir apjungimo algoritmus iš keleto jutiklių duomenų gaunami apjungti rezultatai, padedantys atskirti ir atpažinti jutiklių matymo lauką kertančius/ribojančius gyvūnus. Ant gyvūnų kūnų nešiojamo tipo sistemose gali būti naudojami įvairaus tipo sensoriai (judėjimo, temperatūros, drėgmės ir pan.), kuriais gaunami duomenys apdorojami ir sprendžiama apie gyvūno buvimo vietą, jį supančią aplinką.

#### **1.3.1. Jutiklių tinklo metodika**

Jutiklių tinklo panaudojimu paremta sistema aprašyta T. Damarla, A. Mehmood, J. Sabatier straipsnyje „Detection of people and animals using non-imaging sensors“. Aprašoma spindulio nutraukimo sistema, kurioje panaudojami akustiniai, seisminiai ir ultragarso jutikliai, jų duomenys rūšiuojami, klasifikuojami [13]. Sistemos tikslas yra sukurti algoritmus, kurie gebėtų atskirti žmogų nuo gyvūno pasinaudojant minėtų jutiklių rodmenimis; kiekvienas iš jutiklių, autorių nuomone, suteikia unikalias ypatybes apie tiriamus objektus. Ultragarso jutikliai šioje jutiklių sistemoje veikia tokiu pačiu principu, koku veikia ir jutikliai, aprašyti „Design of Wild Animal Detection and Rescue System with Passive Infrared and Ultrasonic Sensor based Microcontroller“ sistemoje. Kiti jutikliai, panaudoti šioje sistemoje, yra akustiniai jutikliai ir seisminis jutiklis. Akustinių jutiklių dalį sudaro keturių mikrofonų sistema, kurioje trys mikrofonai išdėstyti ratu ir vienas nukreiptas į viršų; seisminiu jutikliu matuojamos žemės vibracijos ir laukiama, kol virpėjimas pasikeis prie sistemos artinantis žmogui arba gyvūnui.

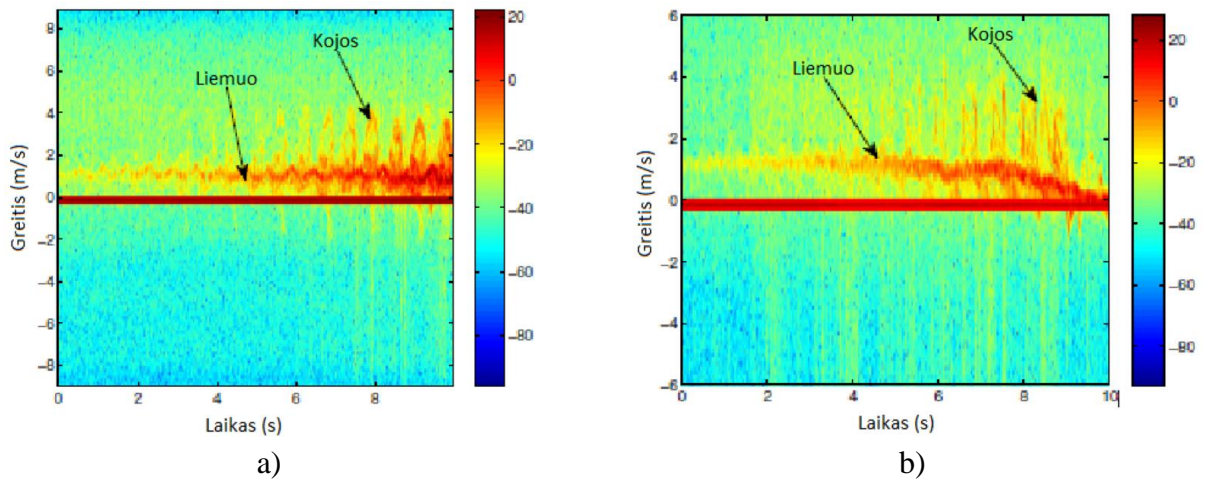
Jutiklių tinklo sistemoje kiekvieno jutiklio suteikiama informacija buvo iširta pirmiausia pavieniui, o tuomet pateiktas bendras sistemos modelis, padedantis atskirti žmones nuo gyvūnų. Toliau pateikiamuose paveikslėliuose galima matyti akustinio, seisminio ir ultragarsinio sensoriaus gražinamų duomenų klasifikavimą.



**7 pav.** Jutiklių duomenų klasifikavimo modeliai: a) akustinio sensoriaus; b) seismo sensoriaus [13]

Akustinio jutiklio duomenys ištiriami apžvelgiant energiją keliose žmogaus balso dažnių spektro juostose, taip pat pasinaudojama žinomais formantais ir moduliacijos charakteristikomis bei bandoma detektuoti žingsnius. Algoritmas, atpažįstantis žmogaus balsą, susideda iš formanto (nešlio dažnio) ir išskleisto spektro. Jeigu aplink nešlio dažnį išskleistas spektras yra virš tam tikro nustatyto lygio, galima teigti, kad tai yra žmogaus balsas. Analizuojant energiją priimto garso spektre, jeigu yra girdimas žmogaus balsas, tyrimo būdu straipsnių autoriaus atskirtose dažnių juostose galima užfiksuoti tam tikras energijos ypatybes, kurios išsaugomos kaip vektoriai ir, remiantis daugiamačiu Gauso (MVG) klasifikatoriumi, yra panaudojamos klasifikuoti garsą kaip žmogaus balsą arba gyvūno skleidžiamus garsus. Taip pat žingsnių atpažinimo dalyje akustiniais jutikliais gautuose duomenyse ieškota žmogaus ir gyvūno žingsnių signalūrų; manoma, kad ritminga žmogaus žingsnių seka patenka į 1 – 2 Hz dažnių juostą, o ritmingas gyvūno ėjimas – į 2,5 – 3 Hz dažnių juostą. Ištiriant harmonikų išsidėstymą dažnių juostoje, galima nuspėti sekas ir jas panaudoti klasifikuojant ėjimą į gyvūno arba žmogaus.

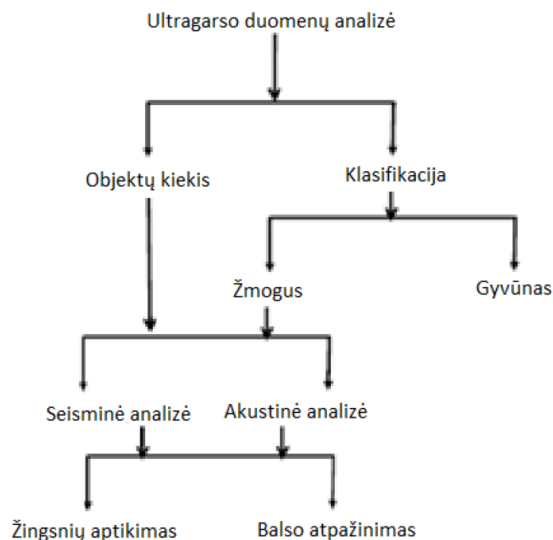
Seismo jutiklio duomenys ištiriami detektuojant žingsnius jutiklio apimamame plote. Nors atskirai gyvūno ir atskirai žmogaus ritmingų žingsnių sekas jutiklio atiduodamo signalo spektre galima pastebėti, tačiau kartu einančių daugiau nei vieno žmogaus ir / ar gyvūno signalūros persipina ir dėlto jas atskirti sunku. Tam, kad išeitų panaudoti turimus duomenis, straipsnio autoriai aptarė išankstines sąlygas, kurių laikantis analizuojami duomenys, pavyzdžiui, kad jeigu taku eina žmonės, jie eina vienas paskui kitą kelių metrų atstumu, nes takas yra siauras; jeigu taku eina gyvūlys, jis yra vedinas žmogaus, ir panašiai. Kadangi gyvūno ir žmogaus žingsnių sekos signalūros skiriasi, pasinaudojant anksčiau minėtu MGK klasifikatoriumi pavyksta atskirti gyvūno ir žmogaus žingsnius. Seismo jutiklio duomenų klasifikavimo medyje (žr. 7 pav. b dalis) nurodoma, kad duomenų analizavimas priklauso nuo paviršiaus, kuriuo eina žmogus arba gyvūnas (pelkėta vietoje ar kietoje žemėje išmintu taku); atpažinus žemės paviršių, duomenys analizuojami išskiriant gyvūną ir žmogų pagal anksčiau minėtą klasifikavimo modelį ir sprendžiamas žmonių kiekis ir jų veiksmai.



8 pav. Mikro doplerio signalūra nuo atskirų kūno dalių: a) žmogaus; b) arklio [13]

Ultragarso jutikliais gauti duomenys apdoroti išskiriant judančių žmogaus ir gyvūno Doplerio signalūras. 8 pav. pavaizduoti tipiniai Doplerio greičiai, kurie yra proporcingi Doplerio dažniams nuo skirtingų žmogaus (a) ir arklio (b) kūno dalių jiems einant. Tyrimo metu užfiksuota, kad Doplerio gražinimas (angl. *Doppler return*) nuo žmogaus juosmens yra stipresnis negu nuo gyvūno juosmens. Energijos pasiskirstymas įvairiose dažnių juostose taip pat skiriasi, lyginant gyvūno ir žmogaus tyrimo duomenis, dėlto klasifikacijai pasinaudota 40 Gauso klasifikatoriumi išskiriamų ypatybių priklausomai nuo to, kaip kinta energija skirtingose dažnių juostose. Išskiriant tris klases klasifikavimo metu (žmogaus, gyvūno ir kitų) pasiektas 95 % tikslumas, o naudojant dvi klases (žmogaus ir kitų, į antrąją įeina ir gyvūnai), pasiektas 98 % tikslumas.

Bendras visos sistemos veikimo modelis pateikiamas toliau.



9 pav. Jutiklių tinkle sistemos veikimo modelis [13]

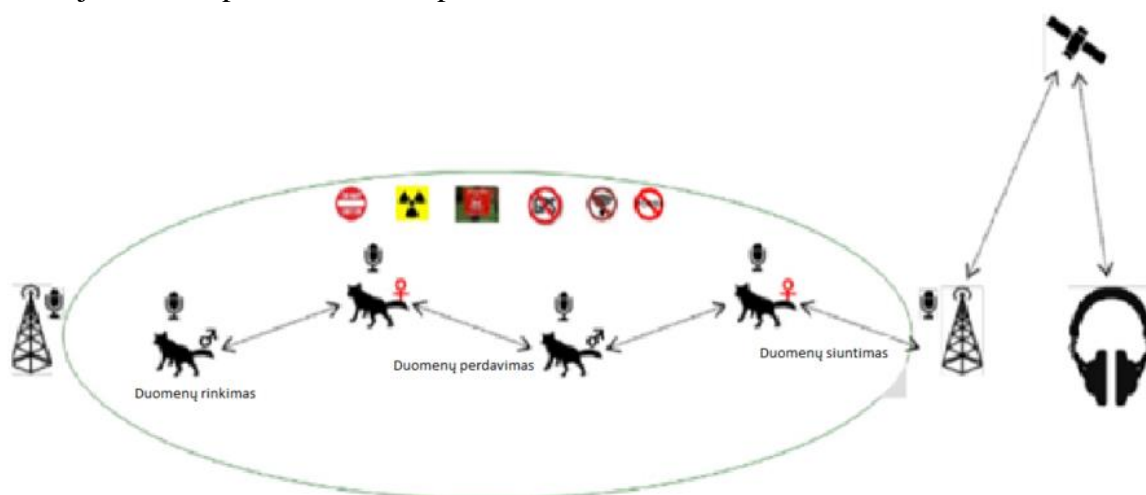
Sistemos, sudarytos iš jutiklių tinklo, veikimo modelyje (žr. 9 pav.) matoma, kad pirmiausia analizuojami ultragarso jutiklių duomenys, padedantys atskirti, ar eina gyvūnai, ar žmonės, bei atpažinti jų kiekį. Jeigu atpažįstama, kad eina žmonės, toliau pasinaudojama seisminio ir akustinio jutiklių duomenimis ir detektuojami žingsniai bei balsas. Ultragarso jutikliai buvo patalpinti 20 metrų toliau nuo akustinių ir seisminių jutiklių, dėlto nebuvo galima apjungti visų trijų jutiklių duomenų

klasifikatoriaus modeliui. Tačiau, kai ultragarso jutikliu aptinkami judantys gyvi objektai, tuomet jau galima pasitelkti akustinius ir seisminius jutiklius tolimesniam objektų tyrimui.

### 1.3.2. Ant gyvūno kūno nešiojami prietaisai

Laukinėse vietose, kuriose nėra išvystyta miestų infrastruktūra ir jos teikiami patogumai – transportas, elektra ir komunikacijos, taip pat negyvenamose vietose gali būti nepatogu talpinti sistemas, kurios reikalautų didelių energijos sąnaudų ir nuolatinės priežiūros. Dėlto „Dependable Wildlife DTN: Wearable Animal Resource Optimization for Sustainable Long-Term Monitoring“ straipsnio autoriai K. Nakagawa, D. Shimotoku, J. Kawase ir H. H. Kobayashi apžvelgia sistemą, galinčią sutaupyti energijos ir sumažinti eksploataavimo išlaidas dėlto, kad veikia jutikliais gaunamų duomenų apsikeitimo tarp prietaisų, pritaistytų ant laukinių gyvūnų, principu [5].

Straipsnio autorių apžvelgta sistema yra paremta gyvūno – gyvūnui duomenų perdavimo mechanizmu ir delsimui tolerantiškų tinklų (angl. *Delay Tolerant Networking, DTN*) technologija. Šis metodas nereikalauja žmogaus įsikišimo tam, kad duomenys būtų fiksuojami ir saugomi nepažeisti, jo modelis pavaizduotas 10 pav.



10 pav. Duomenų gavimo, perdavimo tarp nešlių ir siuntimo sistemos modelis [5]

Visų pirma, gyvūno dėvimas jutiklis surenka duomenis, charakterizuojančius individualų gyvūną ir jo aplinką. Tuomet duomenys yra perduodami tarp gyvūnų peršokos (angl. *multihop*) kolektyvinių veiksmų metu, pavyzdžiui, gyvūnams veisiantis. Galiausiai duomenys perduodami į serverį, kai vienas iš gyvūnų nešlių patenka į ryšio plotą, pavyzdžiui, mobilų nešlių tinklą.

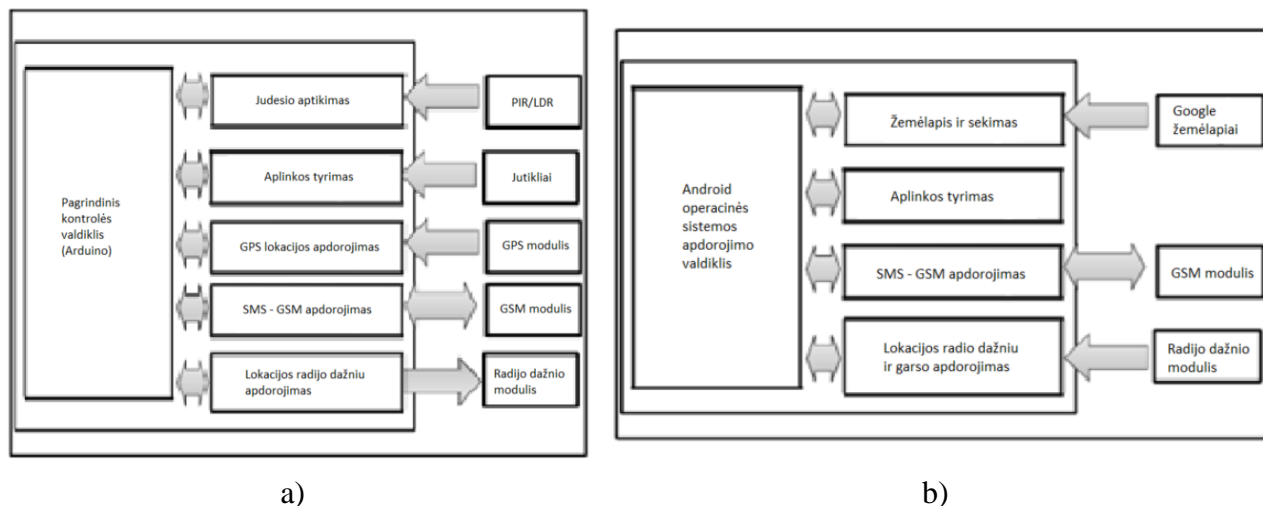
Viena iš problemų, su kuria susidūrė tyrimo autoriai, buvo galios suvartojimas. Belaidė komunikacija aprašytoje sistemoje yra vienintelis būdas perduoti duomenis iš gyvūno gyvūnui tam, kad jie galiausiai pasiektų serverį, dėlto padidėja suvartojama energija, tiekiamą iš nešiojamos baterijos. Tam, kad būtų galima prailginti nešiojamos sistemos laiką, implementuota peršokos komunikacija, kuri atvirkštiniu būdu pasinaudoja laukinių gyvūnų elgsena, taip saugant energijos išteklius sistemoje. Kitaip tariant, bevielė komunikacija sistemoje aktyvuojama tik tuomet, kai jutikliais aptinkami gyvūno elgsenos pokyčiai, indikuojantys, kad buvo sutiktas kitas gyvūnas.

Sekanti apsvairstyta problema yra laukinių gyvūnų mirtingumas. Dėl plėšrūnų ir atšiaurių laukinės gamtos sąlygų gyvūnai, nešiojantys jutiklių sistemas, gali numirti ir taip duomenys tarp mirusio nešlio gyvūno ir kito nešlio gyvūno nebūtų perduodami. Dėlto autoriai pasinaudojo gyvūnų nešlių įtraukimo

modeliu, kurio įgyvendinimo metu periodiškai tam tikras kiekis gyvūnų nešlių su nešiojamais jutikliais išleidžiami į laukinę gamtą; taip jutiklių užfiksuojami duomenys perduodami iš kartos į kartą.

Autoriai sistemos tyrimo metu nustatė, kad, norint išspręsti anksčiau minėtas problemas, būtina užtikrinti tam tikrą kiekį gyvūnų nešlių, kurie iš naujo būtų paleidžiami į laukinę gamtą ir tai reikėtų daryti kas metus. Palyginę savo sistemos modelį su jau egzistuojančiais panašaus tipo modeliais autoriai parodė, kad, kartą per metus įtraukus 120 naujų gyvūnų nešlių, gyvūnų susitikimas, kurio metu buvo aktyvuojamas duomenų perdavimas, padidėjo 1,28 karto (lyginamojoje sistemoje kas tris mėnesius buvo įtraukiama 30 naujų gyvūnų nešlių).

Kitokia sistema, paremta ant gyvūno kūno nešiojamu prietaisu su jutikliais, aprašyta C. So-In, C. Phaudphut, N. Weeramongkonlert, K. Wijitsopon.. U. Kokaew, B. Waikham ir S. Saitod straipsnyje „Mobile Animal Tracking Systems Using Light Sensor for Efficient Power and Cost Saving Motion Detection“ [15]. Straipsnio autorių siūloma sistema pasinaudoja Android operacinės sistemos mobiliaisiais prietaisais, susiejamais su SMS - GSM ir radijo ryšį priimančia IOIO plokšte, kuria galima vykdyti gyvūnų elgsenos stebėjimą tiek abstrakčiai, tiek ir smulkmeniškai.



11 pav. Sistemos architektūros dalys: a) jutimo prietaisų; b) sekimo prietaisų [15]

Sistema susideda iš dviejų dalių – jutiklinio ir sekimo prietaisų. Jutiklinis prietaisas surenka informaciją apie aplinką, kurioje yra gyvūnas, ir siunčia duomenis mobiliais ir belaidžiais tinklais į sekimo prietaisą. Šiame prietaise yra pagrindinis kontrolės valdiklis (angl. *Core Control Unit*) ir 5 moduliai:

- judesio aptikimo - naudojamas aptikti gyvūno judėjimą tam, kad nustatyti, ar dabartinė gyvūno lokacija pasikeitė, ir priklausomai nuo to, ar reikia siųsti lokacijos informaciją į sekimo prietaisą.
- aplinkos tyrimo – prijungtas prie išorinių temperatūros ir drėgmės sensorių stebėti aplinką.
- GPS lokacijos apdorojimo – naudojamas įjungti lokacijos apdorojimo logiką pozicijos duomenims perduoti į SMS – GSM modulį.
- SMS – GSM apdorojimo – kai sukaupiami aplinkos ir lokacijos duomenys, modulis naudojamas perduoti duomenis simbolių formatu SMS ryšiu į sekimo prietaisą. Taip pat modulis priima

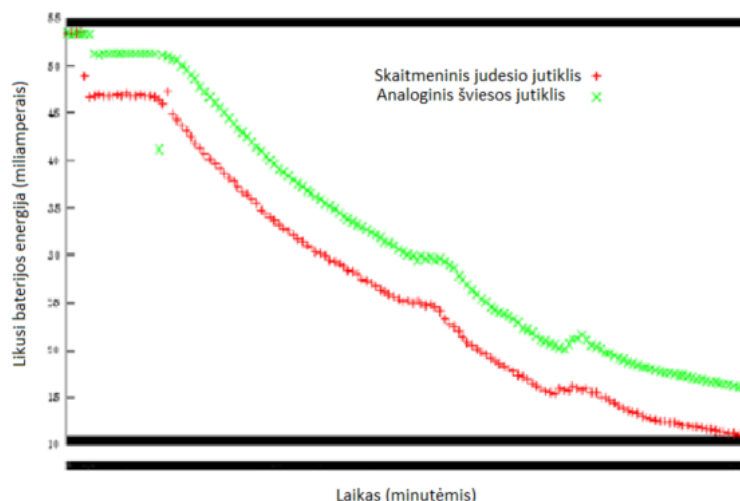
komandas, siūstas SMS ryšiu iš sekimo prietaiso, kurios nusako atlikti tam tikrą užduotį, pvz., nustatyti specifinę lygį, kurį pasiekus aktyvuojamos GPS funkcijos.

- lokacijos radijo dažniu apdorojimo – radijo dažnio bangas siunčia sekimo prietaisui, pasinaudojant signalo lygio matavimu kaip rodikliu, kuris parodo, kiek toli yra gyvūnas nuo mobilus prietaiso.

Sekimo prietaisas skirtas vartotojui sekti gyvūną, jo ieškoti, rasti ir stebėti, taip pat įspėti mobilus prietaiso naudotoją, kuomet jis prisiartina prie ieškomo objekto. Prietaisas susideda iš Android operacinės sistemos kontrolės valdiklio ir 4 modulių:

- SMS – GSM apdorojimo – priima aplinkos ir lokacijos duomenis GSM tinklams naudojantis SMS ryšiu bei saugo juos duomenų bazėje apdoroti vėliau. Taip pat kontrolės perdavimo logikos dalyje šis modulis siunčia komandas į jutiklinį prietaisą atlikti tam tikras užduotis.
- Aplinkos stebėjimo – duomenų bazėje turint aplinkos duomenis, žmogaus akimi suvokiamo teksto pavidalu atvaizduoja temperatūros ir drėgmės informaciją procentais.
- Žemėlapių ir sekimo – atlieka tris funkcijas: atvaizduoja tikslią GPS jutikliu gautą lokaciją pasinaudojant Google žemėlapių funkcionalumu, apskaičiuoja kryptį iki ieškomo objekto nuo dabartinės pozicijos ir atvaizduoja gyvūno nueitą taką įvertinant gyvūno elgsenos analizės istoriją.
- Lokacijos ir garso radijo ryšiu apdorojimo – kartu su IOIO sąsaja prisijungia prie išorinės radijo ryšio grandinės.

Sistemos kūrėjai ištestavo prototipą panaudodami gyvūną, kurio judėjimą stebėjo mobiliojo ryšio prietaiso ekrane. Taip pat dėl energijos suvartojimo sąnaudų taupymo buvo iširtas šviesos jutiklio panaudojimas vietoj judesio jutiklio (12 pav.).



12 pav. Galios suvartojimo praktinis tyrimas 12 valandų eigoje [15]

Kaip galima matyti 12 pav., analoginis šviesos jutiklis energiją vartojo taupiau. Dėlto, naudojantis šviesos jutikliu, kai norima stebėti gyvūno aplinkos pasikeitimą, prietaisą būtų galima naudoti ilgiau negu naudojantis judesio jutikliu, kuris per visą tyrimo laiką išėikvojo didesniąją dalį baterijos.

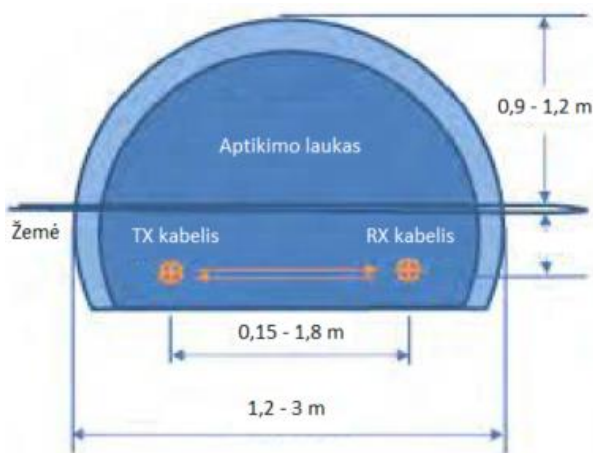


Apibendrinant apžvelgtas ant gyvūno kūno nešiojamas jų aptikimo ir stebėjimo sistemas galima teigti, kad šios sistemos gana tiksliai gali nusakyti gyvūno buvimo vietą ir jo būseną, jeigu panaudojami jutikliai tirti, pavyzdžiui, gyvūno kūno temperatūrai ar aplinkos drėgmei suvokiant, kaip nuo šių parametrų priklauso gyvūno elgsena. Tačiau naudojant prietaisus, dedamus ant gyvūno kūno ir skirtus aptikti gyvūnų pasirodymą tam tikrose vietovėse, reikia pasirūpinti tinkamu kiekiu gyvūnų nešlių, taip pat rengti specialias stoteles, priimančias ir saugančias per tam tikrą atstumą prisiartinančio gyvūno siunčiamus duomenis.

### 1.3.3. Sistemos po žeme

Ankstesniuose skyriuose aprašytos sistemos, tiek dengiamo ploto, tiek spindulio nutraukimo, tiek ir ant gyvūno kūno nešiojami prietaisai veikia virš žemės. C. Druta ir A. S. Alden rašytame straipsnyje „Evaluation of a Buried Cable Roadside Animal Detection system“ apibūdinama kitokio tipo sistema, aptinkanti didelius ir vidutinio dydžio gyvūnus ir implementuota 300 metrų dvigubo kabelio (angl. *dual – cable*) jutikliu, esančiu po žeme [16].

Sistemos pagrindinė sudedamoji dalis yra modulinis po žeme pasirinktame perimetre įkasamas koaksialinis kabelis, veikiantis kaip intruzijos aptikimo kabelis ir pagrįstas atskirtų koaksialinių kabelių (angl. *ported coaxial cable, SC2*) technologija. Aptikimo laukas yra formuojamas radijo bangų signalais, kurie yra perdušami jutiklio kabeliais (vienas – siųstuvo tipo, kitas – imtuvo). Siųstuvo tipo kabelis paskirsto radijo bangų signalus per kabelio ilgį, o imtuvo tipo kabelis priima signalus ir siunčia juos į procesorių (13 pav.).



13 pav. Atpažinimo sistemos modelis [16]

Kaip galima matyti 13 pav., radijo signalai formuoja elektromagnetinį atpažinimo lauką aplink jutiklio kabelius, kurio pagalba galima aptikti ir identifikuoti objektą (šiuo atveju, gyvūną), kertantį minėtą lauką. Aptikus gyvūną, kirtusį formuojamą lauką, procesorius iššaukia aliarmo signalą. Tarpas tarp kabelių priklauso nuo vietos sąlygų ir nuo to, kaip giliai užkasti kabeliai. Sistemai reikalingas duomenų apdorojimo procesorius turi būti prijungtas prie kabelių, taip sujungtame tinkle keičiamasi informacija tarp jutiklių ir procesoriaus bei tarp procesoriaus ir aliarmo atvaizduoti skirtų tinklo dalių.

Apžvelgus aptartos sistemos veikimo principą galima teigti, kad ši sistema yra invazinė – jai įdiegti būtinas žmonių ir mašinų darbas (kabelių užkasimas, pagrindinio procesoriaus tvirtinimas ir kt.); taip pat yra tiksliai – didžiausias pasiektas gyvūnų atpažinimo tikslumas, nustačius tinkamus sistemos parametrus (pakėlus aptikimo lygį iki 15.8 dB ir padidinus kabelių galios faktorių iki 3 dBm), buvo 99,5 %.

#### 1.4. Apibendrinimas

Buvo apžvelgtos laukinių gyvūnų stebėjimo ir aptikimo sistemos, skirstomos į spindulio nutraukimo ir dengiamo ploto sistemas, taip pat kitokio tipo metodikos – jutiklių tinklo ir po žeme užkasamo jutiklio sistemos bei ant gyvūno kūno nešiojami prietaisai.

Spindulio nutraukimo sistemos pasižymi lengvu signalų apdorojimu, tačiau būtent dėlto, kad šiose sistemose naudojami paprasti signalai, jomis negalima nusakyti reikšmingos informacijos apie signalą kirtusį gyvūną – iš kurios pusės jis kirto sistemą, koks jo dydis ir kt.

Dengiamo ploto sistemose signalų apdorojimas yra sudėtingesnis dėl pačių signalų tipų, kuriuos priima šiose sistemose naudojami jutikliai, dėlto iš gaunamų duomenų galima išgauti anksčiau minėtą reikalingą informaciją. Tačiau šio tipo sistemose įprastai naudojami jau didesni jutikliai, kuriems reikia specialių pritvirtinimo stotelių ir komunikacijos infrastruktūros, o diegiant mobilias sistemas būtina įtraukti ir su gyvūnų atpažinimu nesusijusias technologijas, kaip kad vietos sekimą, lokalizaciją žemėlapyje ir kt.

Pasitelkiant jutiklių tinklo metodiką, padidėja apdorojamų duomenų kiekis priklausomai nuo jutiklių skaičiaus, tačiau apdorojant skirtingo tipo duomenis galima atrasti gyvūnų signatūras, būdingas tik tam tikrai rūšiai, ir panaudoti tokias sistemas ne tik stebėti ir atpažinti gyvūnus, bet ir juos klasifikuoti. Tačiau daug jutiklių įtraukiančios, kaip ir po žeme užkasto kabelio, sistemos reikalauja papildomų resursų tinkamai jutiklius išdėstyti, pritvirtinti, patalpinti po žeme, dėlto sistemos eksploatavimas gali tapti brangus.

Ant gyvūno kūno nešiojamų prietaisų sistemos yra vienos arčiausiai laukinio gyvūno esančių sistemų, kurios padeda stebėti jo aplinką ir elgseną, taip pat ir aptikti gyvūną pasirodant tam tikrose teritorijose. Tačiau šių sistemų kūrimas ir implementavimas neapsiriboja vien tik prietaisų, tvirtinamų prie gyvūnų kūnų, sudarymu, tačiau yra reikalingos ir tam tikru atstumu viena nuo kitos nutolusios stotelės, registruojančios gyvūnus ir jų siunčiamus duomenis, tinkami prietaiso maitinimo energija metodai užtikrinti sistemos ilgaamžiškumą.

Naudojant sistemas, kurių pagrindas yra kamerų nuskaitomų vaizdų apdorojimas, būtina užtikrinti tinkamą kameros matymo lauką norimai stebėti teritorijai, taip pat procesorių, gebantį realiu laiku apdoroti gaunamus vaizdus juose aptinkant ir atpažįstant gyvūnus. Tačiau šiam metodui nėra būtinos atskiros stotelės, per atstumą priimančios duomenis ir juos kaupiančios, taip pat sistemos eksploatavimo kaina nėra brangesnė už anksčiau minėtų jutiklių tinklo ir po žeme kasamo kabelio metodikas naudojančių sistemų. Todėl pasirinkta laukinių gyvūnų stebėjimo ir aptikimo sistemai naudoti kamerą ir vaizdo apdorojimo algoritmą talpinti procesoriuje su E-Linux platforma.



## 2. Duomenų atrinkimas ir paruošimas

Duomenų surinkimo sistemą sukūrė Kauno Technologijos universiteto mokslininkai ir buvo gauta galimybė pasinaudoti duomenimis, gautais minėta sistema. Tolimesniuose poskyriuose bus aprašoma duomenų surinkimo sistema bei ja surinktų duomenų paruošimas tolimesniam naudojimui.

### 2.1. Duomenų surinkimo sistema

Duomenų surinkimo sistemos realus vaizdas kelyje Kaunas – Šakiai iš viršaus pateikiamas toliau.



14 pav. Kelio atkarpoje Kaunas – Šakiai išdėstyta duomenų surinkimo sistema

Kaip galima matyti 14 pav., prie kelio vienodu atstumu yra išdėstyti stulpeliai, vaizde paženklinėti ASM1 – ASM8, ant kurių į priešingą keliui pusę yra nukreipti PIR jutikliai. Prie šių stulpelių taip pat yra primontuoti LoRa moduliai, skirti komunikuoti su pagrindiniu sistemos prietaisų bloku.



15 pav. Stulpelio viršutinė dalis su stogeliu apsaugai nuo kritulių

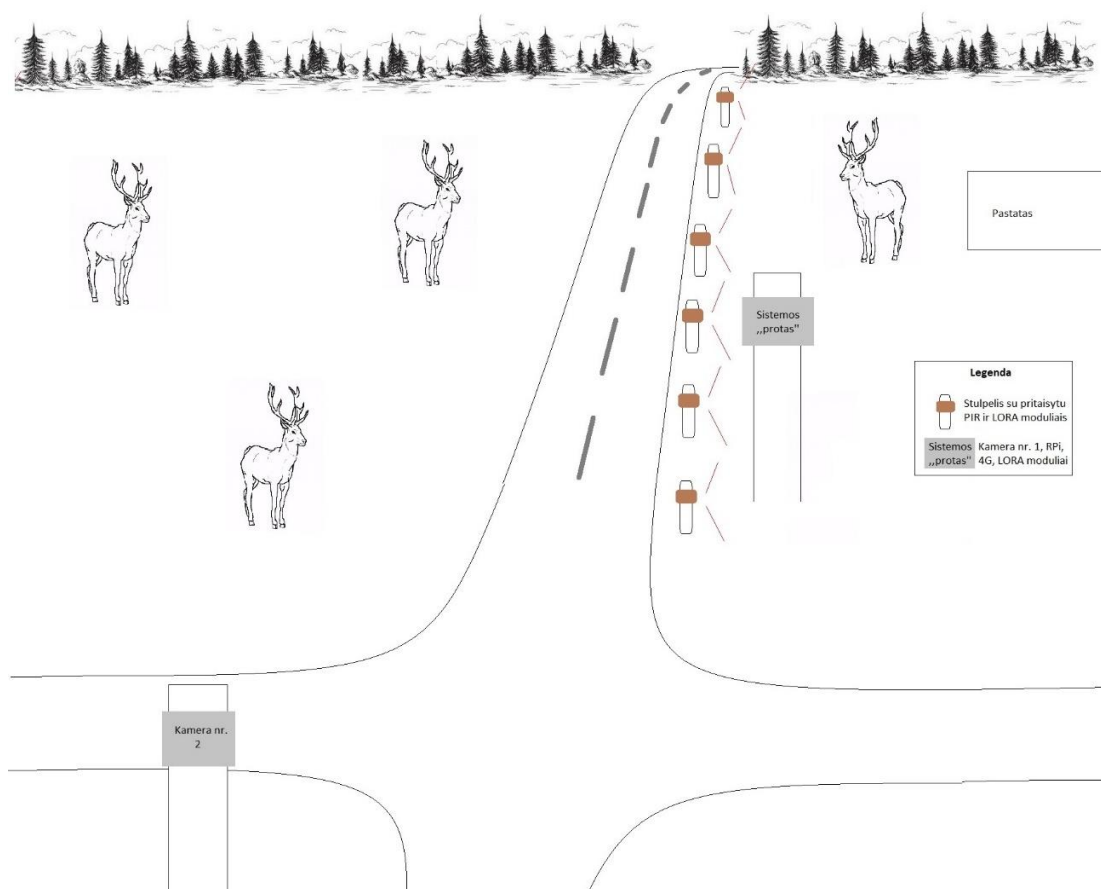
Vieno iš stulpelių viršutinę dalį su pritaisytu stogeliu apsaugai nuo kritulių galima matyti 15 pav.

Prie kelio prisiartinus šilumą skleidžiančiam kūnui, kamera reaguoja į jutiklių suveikimą. Pasinaudojant kameromis taip pat yra renkami duomenys nuotraukų pavidalu ir saugomi duomenų bazėje. Naudojami DS-2TD2617-6/PA termovizoriai, jais gaunamų nuotraukų raiška yra 320 x 240 pikselių [17]. Sistemoje panaudotas termovizorius dėlto, kad pasirinkta pasinaudoti gyvo kūno skleidžiama šiluma ir jos fiksavimu. Įprastomis, šviesos atspindį fiksuojančiomis kameromis gautuose vaizduose gali būti sunku atpažinti gyvūną dėl jo susilieimo su aplinkos spalvine gama arba dėl oro sąlygų. Terminė kamera pašalina šią problemą – ja gautose nuotraukose gyvūnai nesusilieja su pievos aplinka, kadangi žemė skleidžia itin mažą kiekį šilumos, lyginant su gyvūno kūnu.



**16 pav.** Pagrindinio sistemos prietaisų bloko, pritaisyto prie stulpo, vaizdas

Pagrindinis sistemos prietaisų blokas (žr. 16 pav.) yra pritaisytas prie stulpo, esančio šalia kelio; bloke yra kamera, Raspberry Pi kompiuteris, nuotolinį ryšį užtikrinantys 4G ir LoRa moduliai. Antras papildomas prietaisų blokas, kurį sudaro nuotolinį ryšį užtikrinantys 4G ir LoRa moduliai bei kamera, pritaisytas prie kito stulpo kiek toliau nuo pagrindinio sistemos prietaisų bloko ir stulpelių prie kelio. Abu prietaisų blokus ir stulpelių išsidėstymą galima matyti toliau sistemos principinėje schemoje.

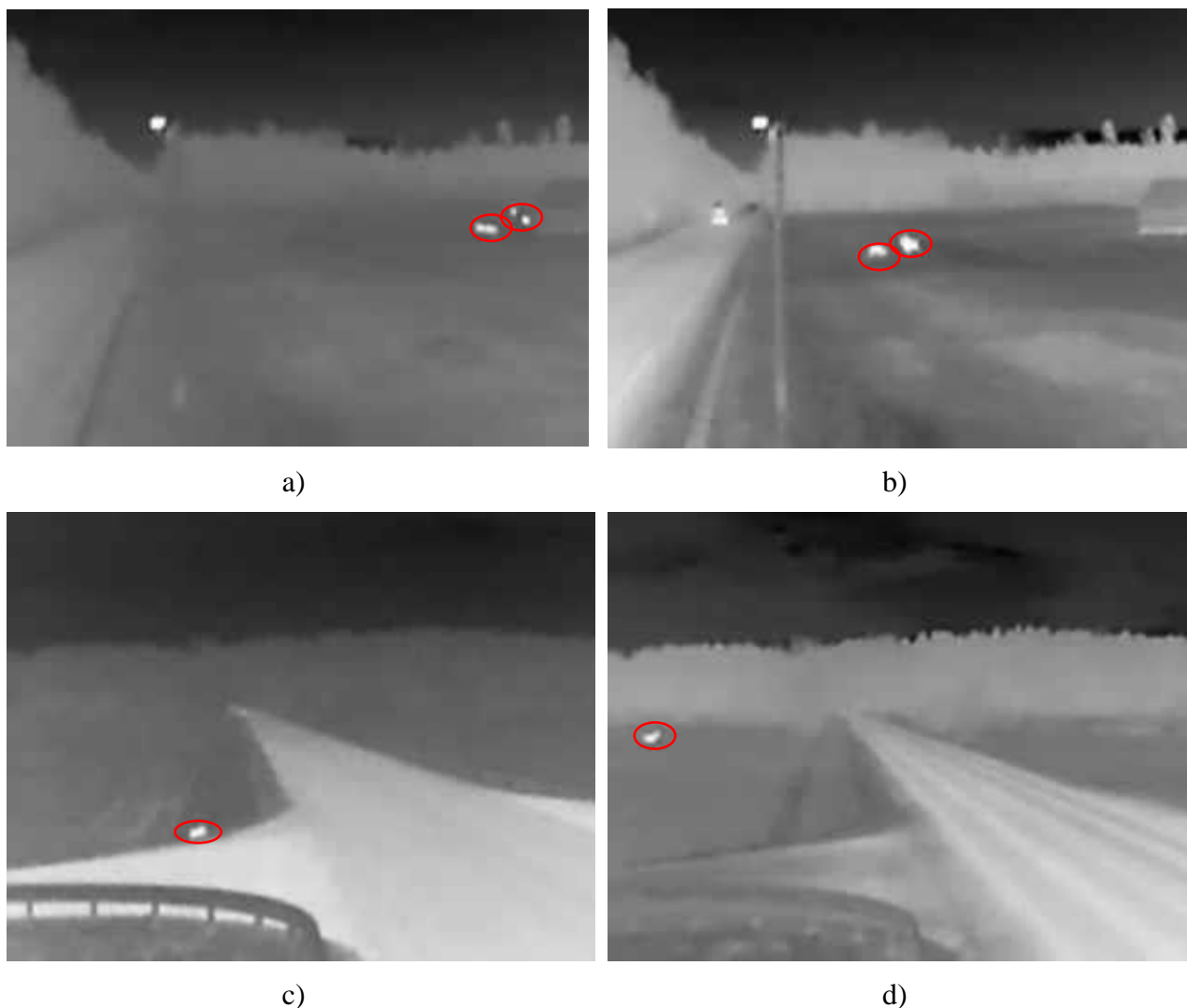


**17 pav.** Duomenų surinkimo sistemos principinė schema

Kaip galima matyti 17 pav., antroji kamera buvo pritaisyta prie stulpo, esančio sankryžos kairėje pusėje už kelio, kertančio lauką.

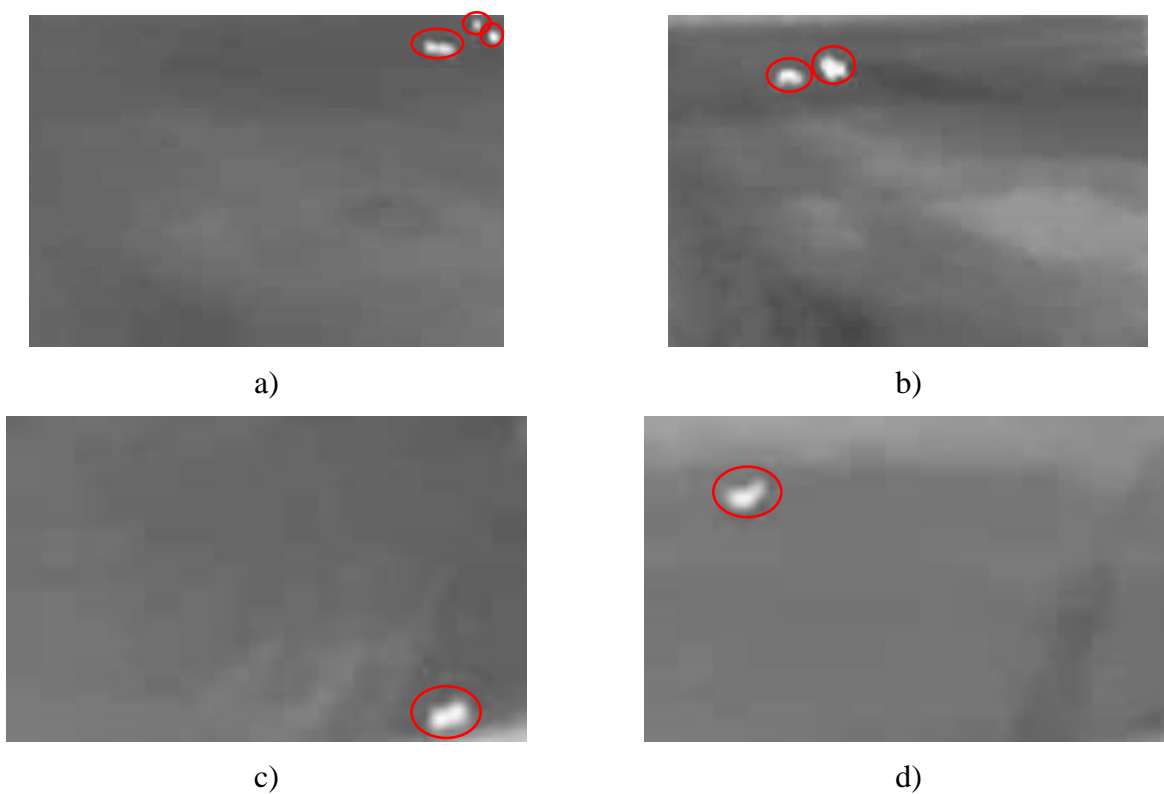
## 2.2. Duomenų atrinkimas ir paruošimas

Poskyryje nr. 2.1 aprašytomis kameromis rinktos nuotraukos, pavaizduotos toliau. Nuotraukose galima matyti raudonais apskritimais apvestus gyvūnus.



**18 pav.** Terminėmis kameromis gautos nuotraukos: a ir b – su kamera, pritaisyta prie pagrindinio sistemos prietaisų bloko, c ir d – su kamera, pritaisyta prie antrinio prietaisų bloko

Kaip galima matyti 18 pav., abejomis kameromis gautuose vaizduose buvo užfiksuoti gyvūnai skirtingose kelio pusėse. Apdorojant nuotraukas jos buvo apkirptos taip, kad būtų matyti tik naudingas plotas – pievos šalia kelio. Kitu atveju, nuotraukose paliekant kelią bei dangų, atsiranda trukdžių, kurie neleidžia tinkamai įvertinti, ar tai gyvūno objektas, ar ne, dėl kameros užfiksuojamų atspindžių bei kitų, ne tik gyvūnų, objektų skleidžiamos šilumos. Apkirptų nuotraukų pavyzdys pateikiamas 19 pav.



**19 pav.** Apkirptos terminėmis kameromis gautos nuotraukos: a ir b – su kamera, pritaisyta prie pagrindinio sistemos prietaisų bloko, c ir d – su kamera, pritaisyta prie antrinio prietaisų bloko

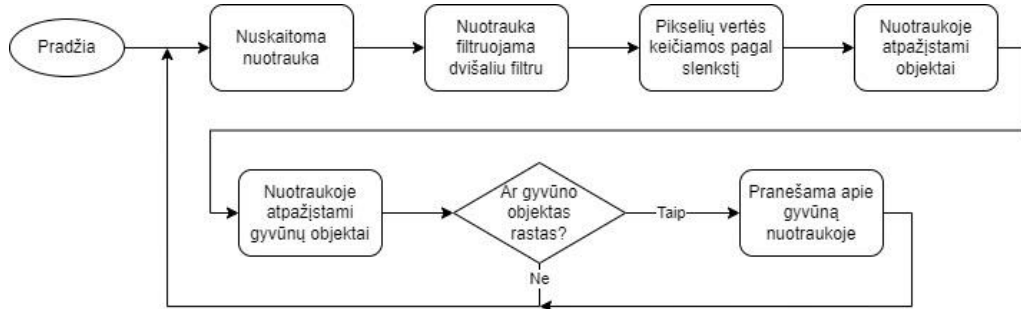
Aukščiau pavaizduotuose paveikslėliuose matomos apkirptos nuotraukos, kurios buvo naudojamos eksperimentuojant ir projektuojant algoritmus. Dalyse a ir b apkirptos nuotraukos, gautos kamera, pritaisyta prie pagrindinio sistemos prietaisų bloko, yra siauresnės už dalyse c ir d matomas apkirptas nuotraukas. Taip yra dėlto, kad naudingas plotas pievose dėl besiskiriančių kamerų išdėstymo pozicijų ir matymo kampo skiriasi.

### 3. Metodikos gyvūnams aptikti nuotraukose

Atrinkus ir paruošus duomenis, skirtus testuoti metodikas, palyginti gaunamus rezultatus, pasirinkta pirmiausia išbandyti metodikas, skirtas aptikti gyvūnus vienoje nuotraukoje. Toliau bus aprašomos trys skirtingos metodikos.

#### 3.1. Metodika, paremta filtracija dvišaliu filtru

Pirmojo metodo gyvūnams aptikti vykdymo algoritmas pateikiamas 20 pav.



20 pav. Dvišaliu filtru paremtos metodikos algoritmas

Apibendrinant dvišaliu filtru paremtą metodiką (žr. 20 pav.), galima matyti, jog atliekamas nuotraukos filtravimas dvišaliu filtru (angl. *bilateral filter*), pikselių verčių keitimas parenkant ir nustatant slenkstį (angl. *thresholding*), nuotraukoje egzistuojančių objektų kontūrų nustatymas ir rastų objektų palyginimas su fiksuotais gyvūnų objektų kontūrais. Tuomet, pagal tai, ar buvo aptiktas gyvūnas nuotraukoje, arba įvyktų pranešimo siuntimas, arba būtų laukiama naujo kadro.

##### 3.1.1. Filtracija dvišaliu filtru

Pasirinkta filtruoti nuotrauką dvišaliu filtru, skirtu pašalinti Gauso triukšmą paveiksluose [18]. Šis filtras yra nelinijinis, Gauso triukšmą šalinantis, išlyginamasis filtras, išsaugantis objektų kraštų aštrumą. Nuotraukos pikseliai yra filtruojami skaičiuojant svorinį vidurkį, kuris priklauso nuo artimų pikselių intensyvumo verčių. Dvišalio filtro pagrindinė formulė pateikiama toliau:

$$I_{\text{filtruotas}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|); \quad (1)$$

čia  $I_{\text{filtruotas}}$  – filtruotas paveikslas,  $I$  – filtruojamas originalus paveikslas įėjime,  $x$  – konkretaus filtruojamo pikselio koordinatės,  $\Omega$  – langas, kurio centro koordinatės yra  $x$ , taip  $x_i \in \Omega$  yra apibrėžiamas kaip vis kitas pikselis,  $f_r$  – diapazono branduolys, skirtas išlyginti skirtumus tarp intensyvumo verčių (gali būti Gauso funkcija),  $g_s$  – erdvinis arba sferinis branduolys, skirtas išlyginti skirtumus tarp koordinačių (gali būti Gauso funkcija).

Svoris  $W_p$  kiekvienam pikseliui lange yra priskiriamas skaičiuojant erdvinį artumą (naudojamas  $g_s$  branduolys) ir intensyvumo verčių skirtumą (naudojamas  $f_r$  branduolys). Kadangi naudojami Gauso branduoliai, vienam iš  $\Omega$  lange esančių pikselių  $(k, l)$  priskiriamas svoris  $w(i, j, k, l)$ , nusakantis šio pikselio įtaką filtruojant centrinį pikselį  $(i, j)$ , matematiškai būtų užrašomas taip:

$$w(i, j, k, l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2} - \frac{\|I(i, j) - I(k, l)\|^2}{2\sigma_r^2}\right); \quad (2)$$

čia  $\sigma_d$  ir  $\sigma_r$  – išlyginimą nusakantys parametrai,  $I(i, j)$  ir  $I(k, l)$  – pikselių  $(i, j)$  ir  $(k, l)$  intensyvumo vertės.

Apskaičiuojamus svorius lango  $\Omega$  pikseliams, jie yra sunormuojami ir apskaičiuojamas intensyvumas centriniam lango pikseliui:

$$I_D(i, j) = \frac{\sum_{k,l} I(k,l)w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}, \quad (3)$$

čia  $I_D(i,j)$  – centrinio pikselio  $(i,j)$  filtruota intensyvumo vertė.

Verta paminėti, kad kraštų ryškumas filtruojamame paveiksle išsaugomas dėlto, kad svoriai priklauso ne tik nuo euklidinio pikselių atstumo, bet ir nuo radiometrinių skirtumų (pvz., diapazono skirtumų, tokių, kaip spalvos intensyvumas, gylio nuotolis ir kt.) [19]. Metodui įgyvendinti parinktas  $\Omega$  lango dydis –  $5 \times 5$  pikselių; pasirinktas toks lango dydis, kad būtų galima filtrą naudoti realaus laiko sistemoje. Taip pat parinktas atstumas pikselių kaimynystei nustatyti skaičiuojant svorinius koeficientus, lygus 9; išlyginimą nusakantys parametrai  $\sigma_d$  ir  $\sigma_r$  parinkti vienodi, jų vertės yra 75.

### 3.1.2. Slenksčio nustatymas

Slenksčio nustatymas (angl. *thresholding*) yra atliekamas tam, kad būtų galima iš pilkų atspalvių nuotraukos, kurioje pikselių vertės kinta 0 – 255 diapazone, gauti binarinio tipo vaizdą, kuriame pikselių vertės yra tik juoda (0) arba balta (1) [20]. Dėlto binarinis vaizdas, reikalingas kitame skyrelyje aprašytam metodikos žingsniui, gaunamas pasinaudojant šia matematine išraiška:

$$I_{apd.}(x, y) = \begin{cases} I_{max}, & \text{jeigu } I_{orig.}(x, y) > \text{slenkstį} \\ 0, & \text{jeigu } I_{orig.}(x, y) \leq \text{slenkstį} \end{cases} \quad (4)$$

čia  $I_{apd.}(x,y)$  – po slenksčio nustatymo operacijos gauto paveikslo pikselių koordinatės,  $I_{orig.}(x,y)$  – originalaus paveikslo pikselių koordinatės,  $I_{max}$  – maksimali leidžiama iš anksto nurodyta pikselio intensyvumo vertė, slenkstis – nustatyta slenksčio vertė, pagal kurią vertinama, kokią reikšmę priskirti konkrečiam pikseliui.

Metodui įgyvendinti pasirinkta slenkstinė vertė lygi 180.

### 3.1.3. Objektų kontūrų radimas

Objektų kontūrų radimo algoritmui reikalingas binarinio tipo paveikslėlis. Kadangi atliekami skaičiavimai tik su dviem intensyvumo vertėmis, galima teigti, kad toje dviejų pikselių poroje, kur vienas pikselio intensyvumas pasikeičia iš 0 į 1, yra kontūro kraštas [21]. Taip galima surasti visus tokius pikselių intensyvumo pasikeitimus ir priskirti juos sudarančius pikselius, nutolusius vienas nuo kito vieno pikselio atstumu, konkrečioms grupėms. Atskira užfiksuotų pikselių intensyvumo pasikeitimo grupė ir yra vadinama objekto kontūru.

Atliekant objektų kontūrų radimą, pasirinkta suspausti horizontalius, vertikalius ir įstrižus segmentus, išsaugant tik būtinas kontūro formą nusakyti pikselių koordinatas. Taip pat programiškai atmetami nuotraukoje rasti objektai, kurie kertasi su nuotraukos kraštu – taip minimizuojamas netinkamų objektų aptikimas.

### 3.1.4. Gyvūnų objektų kontūrų atitikmenų radimas

Atlikus objektų kontūrų radimą, kontūrų ribose išrenkami išskirtiniai požymiai, kitaip – ypatybės, kurios apibūdina objekto unikalumą. Iš nuotraukų, gautų sistemoje įdiegta kamera, iš anksto išrinkti paveikslėliai su gyvūnais, ir gyvūnų objektų nuotraukos apkarpytos ir įtrauktos į duomenų rinkinį. Pasinaudojant šia duomenų baze ir žinant, kad duomenų bazėje esančioje nuotraukoje tikrai yra gyvūnas, jo objektui paveiksle taip pat išrenkami išskirtiniai požymiai. Šie iš realiu laiku gaunamos nuotraukos išrinkti požymiai ir iš gyvūnų objektų nuotraukų gaunami požymiai bei jų apibrėžimai lyginami tarpusavyje. Požymių išrinkimas kompiuterinėje vizijoje vadinamas požymių detektacija (angl. *feature detection*), o požymių apibūdinimas tam, kad būtų galima juos atrasti ne tik originalioje, bet ir kitose nuotraukose, vadinamas požymių apibrėžimu (angl. *feature description*). Išskirtinių ypatybių išrinkimo ir apibrėžimo algoritmai yra implementuoti OpenCV (liet. atviro kodo kompiuterinės vizijos, open source computer vision) bibliotekoje, šiam metodikos etapui įgyvendinti pasirinkta pasinaudoti SIFT algoritmu, o požymiai skirtingose nuotraukose palyginami naudojantis tiesioginio pritaikymo (angl. *Brute-Force*) atitikmens radimo algoritmu [22].

SIFT, kitaip – masteliui invariantiškų požymių transformacijos (angl. *scale invariant feature transform*) algoritmas susideda iš keturių žingsnių. Pirmiausia pasinaudojant Gauso Laplasu (angl. *Laplacian of Gaussian*) prie skirtingų mastelio keitimo parametro verčių bei atliekant Gauso skirtumo (angl. *Difference of Gaussians*) aproksimaciją potencialūs pagrindiniai požymiai randami prie skirtingų mastelių ir skirtingose erdvės vietose. Tuomet potencialūs požymių taškai yra filtruojami pašalinant žemo kontrasto taškus pagal ekstremumo taško intensyvumą bei pašalinant objektų kraštų taškus. Trečiame žingsnyje atliekamas orientacijos priskyrimas pagrindinių požymių taškams: sukuriama tokio pačio mastelio ir toje pačioje vietoje esantys taškų junginiai, kurie skiriasi savo kryptimi – posūkio kampais. Galiausiai, požymiui apibrėžti pažymimos pagrindinio taško kaimyninių pikselių vertės, orientacijos diagramos sukuriama iš suskaidytos kaimyninių pikselių verčių matricos ir išsaugomos vektorių pavidalu.

Tiesioginio pritaikymo atitikmens radimo algoritme atstumo metrika palyginami abiejose nuotraukose rastų požymių apibrėžimai ir pagal trumpiausią atstumą tarp jų surandami ypatybių taškai, potencialiai atitinkantys panašias vietas nuotraukose.

Atliekant gyvūnų aptikimą nuotraukose rastuose objektuose, tiek objektams, tiek apkarpytomis gyvūnų nuotraukoms randami pagrindiniai požymiai ir jų apibrėžimai. Tuomet šie požymiai palyginami ir, jeigu objekto kontūruose randami požymiai sutampa su duomenų rinkinyje esančio gyvūno požymiais, teigiama, kad nuotraukoje yra gyvūnas.

### 3.1.5. Rezultatų aptarimas

21 pav. galima matyti nepakeistą nuotrauką su gyvūnais ir pirmosios metodikos įgyvendinimo žingsnių iliustracijas: filtraciją dvišaliu filtru, slenksčio nustatymą bei gyvūnų kontūrų atpažinimą.

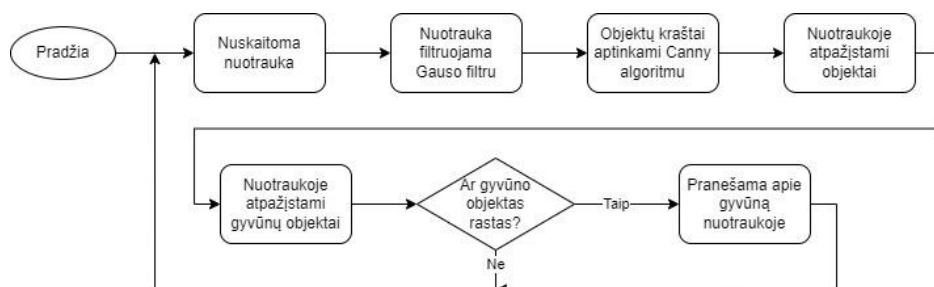


21 pav. Dvišalio filtro metodikos įgyvendinimo žingsniai

Kaip galima matyti 21 pav., filtracija dvišaliu filtru pašalina nuotraukoje egzistuojančią triukšmą, ją suglotnina, tačiau objektų, matomų baltos spalvos pavidalu, kraštai išlieka tokie pat ryškūs. Parinktas tinkamas slenkstis leidžia objektų atpažinimo algoritmui pateikti objektus vaizduojančias baltas dėmes (angl. *blob*) juodame fone. Pavyzdinėje nuotraukoje aptikti du gyvūnų objektai; tiek originalioje, tiek metodikos žingsnius iliustruojančiose nuotraukose gyvūnų objektai apvesti raudonos spalvos stačiakampiais.

### 3.2. Metodika, paremta Canny kraštų radimo algoritmu

Antrojo metodo gyvūnams aptikti vykdymo algoritmas pateikiamas 22 pav.



22 pav. Canny algoritmu paremtos metodikos algoritmas

Antrojo metodo gyvūnams nuotraukose aptikti įgyvendinimo metu (žr. 22 pav.) buvo atliktas nuotraukos suliejimas Gauso filtru (angl. *Gaussian filter*), objektų kraštų aptikimas pasinaudojant Canny algoritmu, nuotraukoje egzistuojančių objektų kontūrų nustatymas ir rastų objektų palyginimas su fiksuotais gyvūnų objektų kontūrais. Tuomet, pagal tai, ar buvo aptiktas gyvūnas nuotraukoje, arba vyktų pranešimo siuntimas, arba būtų laukiama naujo kadro.



### 3.2.1. Suliejimas Gauso filtru

Prieš naudojant Canny algoritmą išskirti objektų kraštus, būtina nuotraukoje minimizuoti triukšmus. Dėl to pasirinkta filtruoti nuotrauką Gauso filtru [23]. Nuotraukai pritaikomas filtras, aprašomas formule:

$$K(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2}\right); \quad 1 \leq x, y \leq (2k + 1); \quad (5)$$

čia  $K$  – filtro branduolys – kaukė,  $x$  ir  $y$  – konkretaus filtruojamo pikselio koordinatės,  $\sigma$  – suliejimą nusakantis parametras,  $(2k + 1)$  – filtro kaukės dydis – eilučių ir stulpelių kiekis.

Pavyzdinis filtras, kuomet parametras  $k$  lygus 1, taip filtro eilučių ir stulpelių skaičius lygus 3, parodytas toliau:

$$I_{\text{filtruotas}} = \frac{1}{28} \begin{bmatrix} 2 & 4 & 2 \\ 4 & 8 & 4 \\ 2 & 4 & 2 \end{bmatrix} * I_{\text{originalus}}; \quad (6)$$

čia  $I_{\text{filtruotas}}$  – filtruotas paveikslas,  $I_{\text{originalus}}$  – originalus paveikslas įėjime, koeficientų matrica – matrica, nusakanti svorinę gretimų pikselių daromą įtaką apskaičiuojant centrinio pikselio intensyvumo vertę. Žvaigždės simbolis „\*“ filtro veikimą apibūdinančioje formulėje nusako konvoliucijos operaciją: filtravimas pradedamas nuo viršutinio kairiojo kampo, centrinį filtro pikselį padedant arčiausiai kairiojo viršutinio paveikslo kampo esančio pikselio taip, kad filtro kaukė liestų kraštinius ir kampinį filtro pikselius. Apskaičiavus centrinio pikselio vertę, filtras slenkamas vieno pikselio žingsniu į kairę, priėjus eilutės pabaigą – į sekančios eilutės pradžios pikselį; skaičiavimo operacija atliekama buvusioms pikselių vertėms, naujosios pikselių intensyvumo vertės saugomos atmintyje.

Naudojant šį filtrą prarandamas objektų kraštų ryškumas, tačiau nuotrauka yra suliejama, taip pašalinant triukšmą ir sulyginant pasikeitimus tarp fono ir nereikšminių objektų, kurių nereikėtų aptikti, pvz., balų atspindžių [24]. Objektų kraštų ryškumo užtikrinti prieš panaudojant Canny algoritmą nėra būtina. Filtro kaukės dydis, pasirinktas įgyvendinti metodui, yra 3x3 pikselių.

### 3.2.2. Canny kraštų aptikimo algoritmas

Kadangi objekto briauna paveiksle gali būti pasisukusi įvairiomis kryptimis, Canny algoritmas įprastai naudoja keturis filtras aptikti horizontalias, vertikalias ir įstrižas briaunas sulietame paveiksle. [25] Atliekant kraštų aptikimo operaciją (šioje metodikoje pasirinktas naudoti Sobelio kraštų aptikimo operatorius) pritaikomos dvi konvoliucijos kaukės, veikiančios  $x$  ir  $y$  kryptimis, matomos toliau:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}; \quad (7)$$

Naudojantis kaukėmis, atliekama filtravimo procedūra paveiksliui konvoliucijos principu. Ant vis naujo centrinio pikselio patrauktos kaukės leidžia suskaičiuoti objekto briaunos gradientą ir kryptį:

$$G = \sqrt{G_x^2 + G_y^2}; \quad \theta = \arctan\left(\frac{G_y}{G_x}\right); \quad (8)$$

čia  $G$  – krašto gradiento stiprumas,  $\theta$  – krašto gradiento kryptis,  $G_x$  ir  $G_y$  – atitinkamomis  $x$  ir  $y$  kryptimis veikiančiomis kaukėmis filtruotos pikselių grupės intensyvumų pirmos eilės išvestinės. Krašto gradiento kryptis pagal savo reikšmę priskiriama vienai iš keturių galimų reikšmių:  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ .

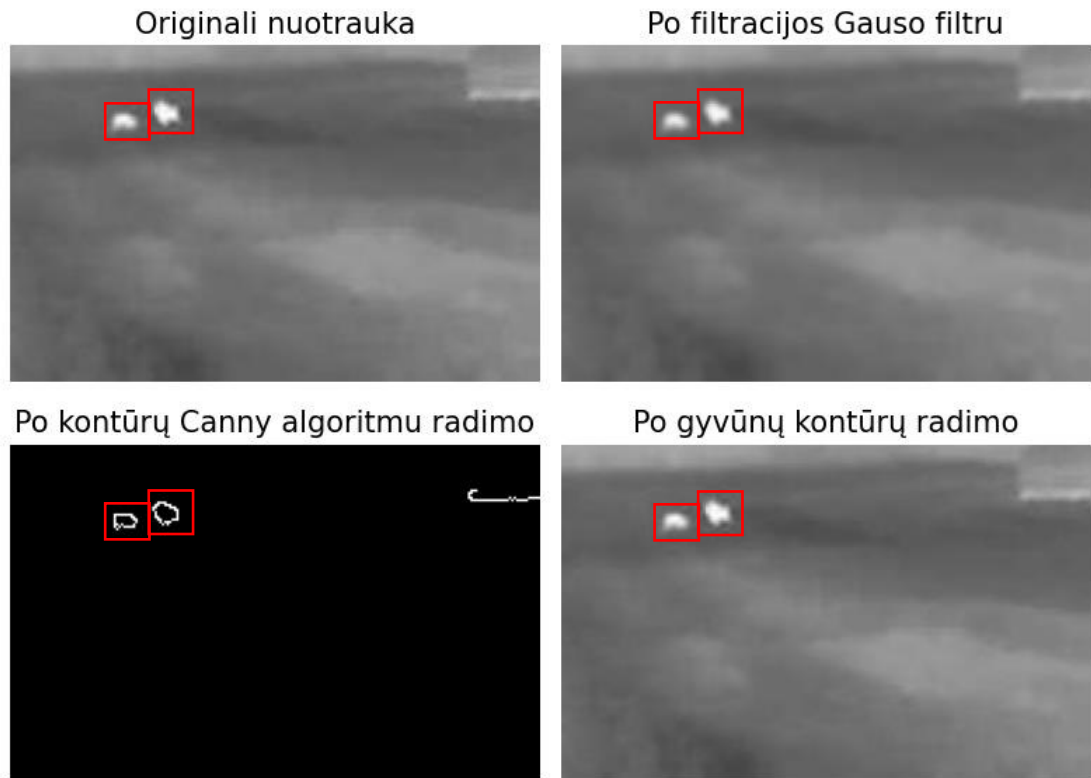
Toliau taikoma kraštų ploninimo (angl. *edge thinning*) technika [26]. Kiekvienam pikseliui gradiento paveiksle yra palyginamas briaunos stiprumas tarp dabartinio pikselio ir pikselių teigiama ir neigiama gradiento kryptimi. Jeigu briaunos stiprumas dabartiniam pikseliui yra didžiausias, lyginant su kitais pikseliais, esančiais po ta pačia kauke ir laikomais tos pačios krypties gradiento dalimi, šis pikselis bus laikomas objekto krašto dalimi. Kitu atveju jis bus atmetamas.

Po kraštų ploninimo operacijos atliekamas dvigubo slenksčio nustatymas stiprias ir silpnas gradiento vertes turintiems pikseliams, kurie laikomi objekto briaunos dalimi. [25] Kadangi dalis pikselių gali būti laikomi krašto dalimi, tačiau skaičiavimuose atsidūrę atsitiktinai dėl spalvinio diapazono arba triukšmo, yra būtina kraštais laikyti tik tuos pikselius, kurie turi aukštą gradiento vertę. Tai atliekama pasirenkant aukštą ir žemą slenkstines vertes. Jeigu pikselio vertė yra aukštesnė už aukštą slenkstinę vertę, jis laikomas stipriuotu briaunos pikseliu. Kitu atveju, jeigu pikselio vertė yra tarp abiejų slenkstinių verčių, šis pikselis laikomas silpnuoju briaunos pikseliu; jeigu pikselio vertė yra žemesnė už žemesnę slenkstinę vertę, pikselis atmetamas. Galiausiai analizuojant silpnuosius pikselius, priimami tie, kurie liečiasi su stipriausiais pikseliais; kiti silpnieji pikseliai atmetami, o stiprieji priimami kaip objektų kraštai.

Canny algoritmo išeigoje gaunamas binarinis paveikslėlis, kuriame aukščiausio intensyvumo – baltos spalvos – pikseliai atitinka objektų kraštus, o juodos spalvos – žemiausio intensyvumo – pikseliai atitinka foną ir objektų užpildą. Binarinis paveikslas toliau apdorojamas objektų kontūrų radimo algoritmu, aprašytu 3.1.3 skyrelyje, bei jame ieškoma gyvūnų objektų pasinaudojant technika, aprašyta 3.1.4 skyrelyje.

### 3.2.3. Rezultatų aptarimas

Paveiksle nr. 23 galima matyti nepakeistą nuotrauką su gyvūnais ir antrosios metodikos įgyvendinimo žingsnių iliustracijas: filtraciją Gauso filtru, objektų kontūrų nustatymą Canny algoritmu bei gyvūnų kontūrų atpažinimą.

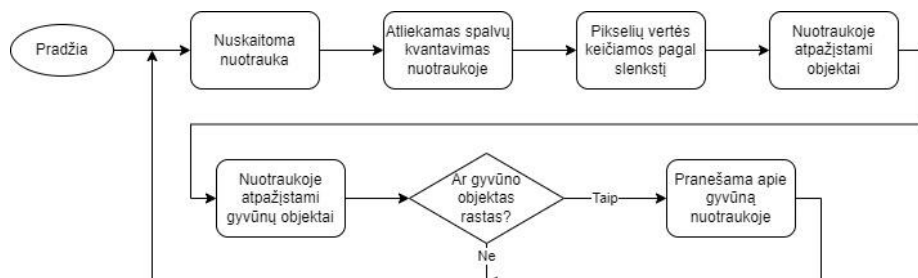


23 pav. Canny algoritmo metodikos įgyvendinimo žingsniai

Atlikus filtraciją Gauso filtru, galima matyti, kad baltų objektų kontūrai viršuje dešinėje esančioje nuotraukoje tampa blankesni nei nuotraukoje viršuje kairėje (žr. 23 pav.). Canny algoritmu apdorota nuotrauka suteikia informaciją apie iliustracijoje rastus objektus baltų kontūrų pavidalu. Kaip galima matyti, buvo išryškinti ne tik gyvūnų objektų kontūrai, bet ir toliau nuo jų matomo statinio apačios apybraiža. Pasinaudojus gyvūnų objektų aptikimo algoritmo dalimi, aptikti du gyvūnai; tiek originalioje, tiek metodikos žingsnius iliustruojančiose nuotraukose gyvūnų objektai apvesti raudonos spalvos stačiakampiais.

### 3.3. Metodika, paremta spalvų kvantavimu artimiausio vidurkio algoritmu

Trečiojo metodo gyvūnams aptikti vykdymo algoritmas pateikiamas 19 pav.



24 pav. Spalvų kvantavimo algoritmu paremtos metodikos algoritmas

Apibendrinant spalvų kvantavimo algoritmu paremtą metodiką (žr. 24 pav.) galima matyti, jog atliekamas nuotraukos spalvų kvantavimas, paremtas artimiausio vidurkio (angl. *K-Means*) metodu, pikselių verčių keitimas parenkant ir nustatant slenkstį, nuotraukoje egzistuojančių objektų kontūrų

nustatymas ir rastų objektų palyginimas su fiksuotais gyvūnų objektų kontūrais. Tuomet, pagal tai, ar buvo aptiktas gyvūnas nuotraukoje, arba vyktų pranešimo siuntimas, arba būtų laukiama naujo kadro.

### 3.3.1. Spalvų kvantavimas artimiausio vidurkio algoritmu

Spalvų kvantavimas yra paveikslo redagavimo metodika, leidžianti sumažinti spalvų kiekį, naudojamą vaizduoti paveiksliui [27]. Buvo pasirinkta pritaikyti šį metodą kaip pirminį nuotraukos apdorojimą siekiant sumažinti pilkų atspalvių kiekį nuotraukoje. Taip spalvinė paveikslo gama būtų sugrupuojama į nurodytą kiekį spalvų grupių, siekiant išskirti kamera fiksuojamuose vaizduose objektus, kurie būtų šilumą spinduliuojantys ir dėlto itin išsiskiriantys iš bendro fonu.

Spalvų kvantavimo procedūrai atlikti buvo pasirinkta pasinaudoti artimiausio vidurkio algoritmu [28]. Taikant šį algoritmą, įėjimo duomenų – paveikslo pikselių – intensyvumo vertės yra grupuojamos į nurodytą kiekį grupių (angl. *clusters*) priklausomai nuo vertės dydžio.

Visų pirma, atliekamas pirmasis algoritmo žingsnis - atsitiktinai yra parenkamas nurodytas kiekis centrinių grupių taškų – tai yra pikselių intensyvumo vertės. Tuomet paskaičiuojamas euklidinis atstumas (angl. *euclidean distance*) nuo kiekvieno taško iki centrinių taškų ir pikseliai atitinkamai yra pažymimi kaip priklausantys tai grupei, iki kurio centrinio pikselio intensyvumo yra mažiausias atstumas, pagal šią formulę:

$$G_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \forall j, 1 \leq j \leq k \right\}; \quad (9)$$

čia  $k$  – grupių kiekis,  $m_i$  ir  $m_j$  – atitinkamai naujos priskiriamos ir dabartinės priskirtos pikseliui grupės centrinis taškas,  $x_p$  – analizuojamas pikselis,  $G_i^{(t)}$  – konkreti grupė.

Tuomet yra atliekamas antrasis žingsnis - paskaičiuojamas vidurkis iš kiekvienai grupei priklausančių pikselių intensyvumų verčių; šie apskaičiuoti įverčiai tampa naujais grupių centrniais taškais pagal formulę, pateiktą toliau:

$$m_i^{(t+1)} = \frac{1}{|G_i^{(t)}|} \sum_{x_j \in G_i^{(t)}} x_j; \quad (10)$$

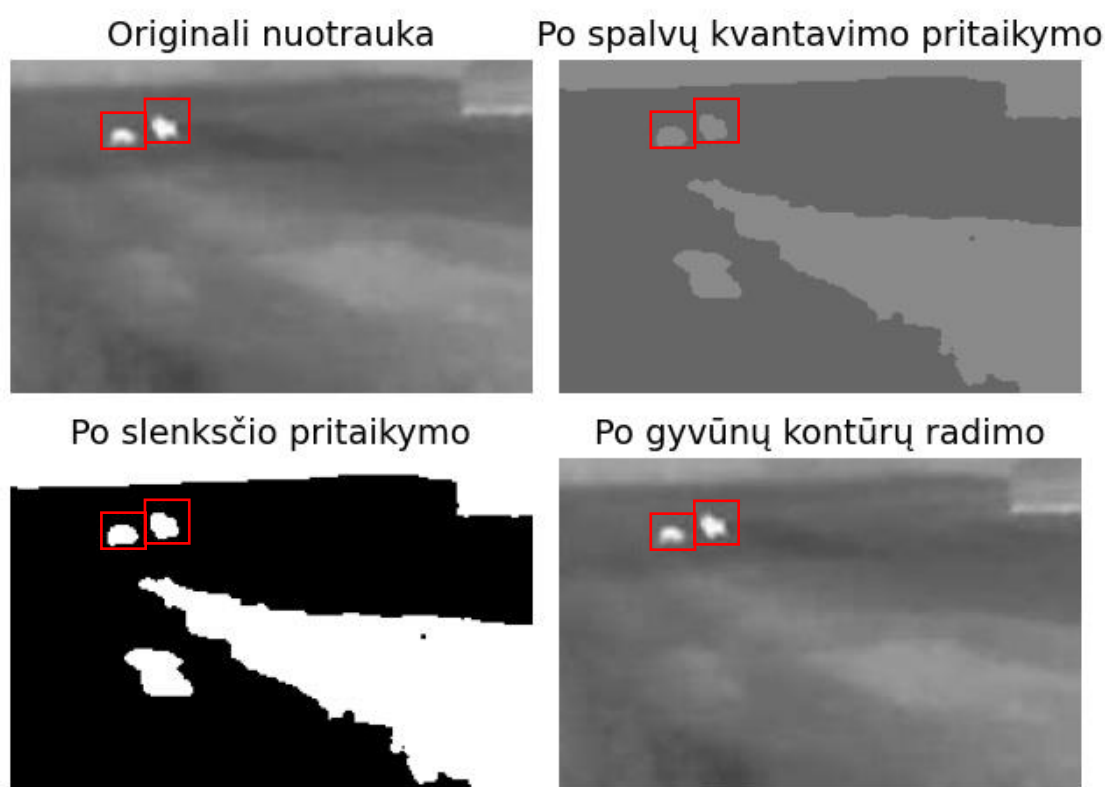
čia  $m_i^{(t+1)}$  – naujai skaičiuojamas centrinis grupės taškas,  $G_i^{(t)}$  – konkreti grupė,  $x_j$  – pikselių, priklausančių konkrečiai grupei  $G_i^{(t)}$ , intensyvumo vertės.

Pikselių priskyrimo grupėms pagal atstumą iki jų centrų ir pačių grupių centrų perskaičiavimo žingsniai atliekami tol, kol centriniai grupių taškai savo pozicijos nebekeičia arba yra pasiekiamas nurodytas kriterijus, pavyzdžiui, įvykdomas numatytas kiekis iteracijų, pasiekiamas tam tikras tikslumas ir kt.

Nustačius tikslius grupių centrus pikselių intensyvumo verčių pavidalu, atliekamas spalvų kvantavimas - visi paveikslo pikseliai pakeičiami tomis intensyvumo vertėmis, kurias turi centriniai jiems priskirtų grupių pikseliai [28]. Pasirinkta paveikslą prieš apdorojant spalvų kvantavimo metodu jį filtruoti filtru, aprašytu 3.2.1 skyrelyje, tam, kad informacijos nesuteikiantys objektai būtų suliejami su fonu. Pačiam algoritmui vykdyti nuspręsta kurti dvi grupes, į kurias bus skirstomos pikselių intensyvumo vertės, maksimalus iteracijų kiekis – 200, maksimalus tikslumas – 0.5. Toliau tęsiant paveikslo analizę rasti objektų kontūrai, aprašyti 3.1.3 skyrelyje ir šie kontūrai tapatinti su fiksuotais gyvūnų objektų kontūrais, ši technika aprašyta 3.1.4 skyrelyje.

### 3.3.2. Rezultatų aptarimas

Toliau pateikta nuotrauka su gyvūnais ir trečiosios metodikos žingsnius – spalvų kvantavimą, slenksčio pritaikymą – iliustruojantys paveikslai.



25 pav. Spalvų kvantavimo metodikos įgyvendinimo žingsniai

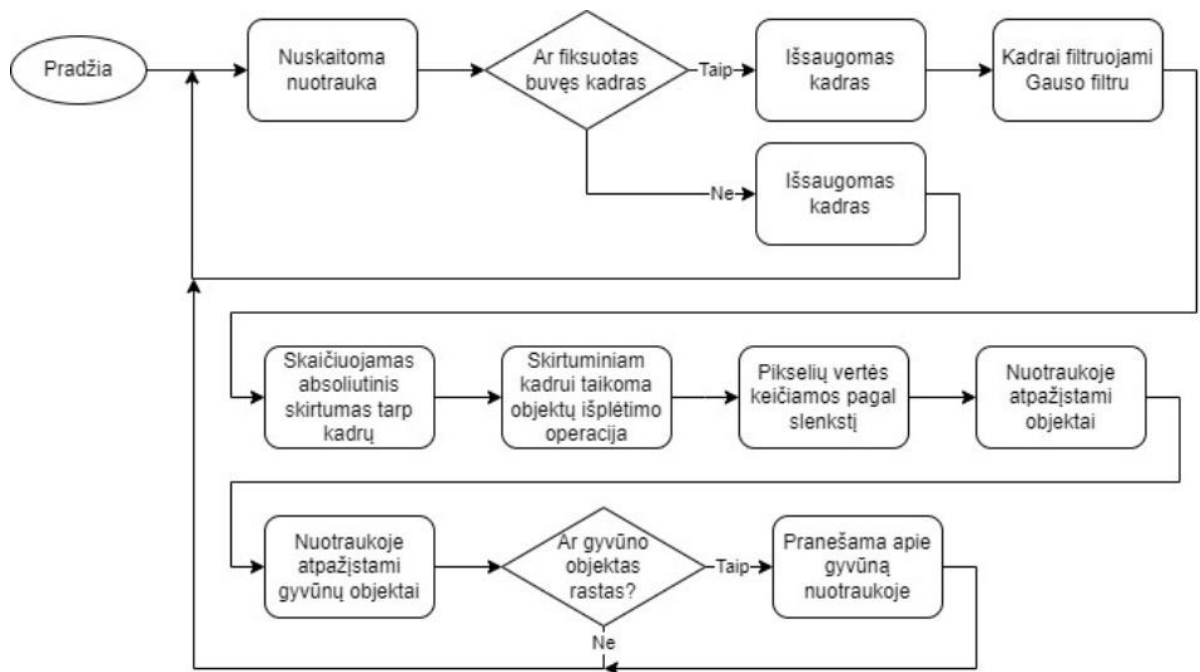
Kvantuojant spalvas buvo pasirinkta jas skirstyti į dvi grupes. Todėl 25 pav. dešinėje viršuje galima matyti dvi skirtingo šviesumo pilkas spalvas, į kurias suskirstyti visi kiti nuotraukoje egzistuojantys atspalviai. Po slenksčio pritaikymo šviesesnės pilkos atspalvis tampa baltu, tamsesnės – juodu. Verta paminėti, kad į baltą spalvą turi būti konvertuojama šviesiausia spalva, kadangi dėl šiluminės nuotraukos ypatybių gyvūnai visuomet turės šviesiausią atspalvį nuotraukoje. Pavyzdinėje nuotraukoje aptikti du gyvūnai, tiek originalioje, tiek metodikos žingsnius iliustruojančiose nuotraukose gyvūnų objektai apvesti raudonos spalvos stačiakampiais.

### 3.4. Metodika gyvūnams aptikti nuotraukų rinkinyje, paremta judesio aptikimu

Atlikus ir išanalizavus rezultatus apdorojant nuotraukas kaip vienetus, nuspręsta atlikti gyvūnų objektų aptikimą nuotraukų rinkinyje, kitaip tariant, analizuoti vieną po kitos iš kameros gaunamas nuotraukas ir bandyti aptikti gyvūnus lyginant prieš tai buvusią ir dabartinę nuotrauką. Toliau bus aprašyta metodika, skirta būtent darbui su dviem iš eilės einančiomis nuotraukomis.

#### 3.4.1. Metodikos apibūdinimas

Buvo pasirinkta išbandyti metodiką, padedančią aptikti gyvūnų judesį vienas po kito einančiuose kadruose. Metodikos veikimo algoritmas pavaizduotas 21 pav.



26 pav. Judesio aptikimu paremtos metodikos algoritmas

Proceso (žr. 26 pav.) eigoje nuotraukos buvo filtruojamos Gauso filtru, kurio veikimo principas paaiškintas 3.2.1 skyrelyje, tuomet atliekamas absoliutinio skirtumo skaičiavimas tarp buvusio ir dabartinio kadro ir gautam skirtuminiam kadrai pritaikoma objektų išplėtimo (angl. *dilate*) operacija. Toliau atliekamas pikselių verčių keitimas nustatytus slenkstį, nuotraukoje egzistuojančių objektų kontūrų nustatymas ir rastų objektų palyginimas su fiksuotais gyvūnų objektų kontūrais.

### 3.4.2. Absoliutinis skirtumas tarp kadru

Kiekviena iš termovizoriaus gaunama nuotrauka yra paveikslas, sudarytas iš tam tikro kiekio pikselių su jų intensyvumo vertėmis. Todėl nuotrauka gali būti vertinama kaip pikselių intensyvumų matrica arba masyvas, ir dėlto galima atlikti absoliutaus skirtumo skaičiavimą tarp skirtingų kadru. [29] Tai leidžia pastebėti, kuriose paveikslo vietose įvyko itin didelis pokytis tarp pikselių intensyvumo verčių – tose vietose atsirado naujas objekto kraštas arba užimamas plotas, dėlto galima daryti išvadą, kad objektas pajudėjo. Absoliutaus skirtumo išraiška:

$$P_{skirtuminis}(i) = |P_{buvęs}(i) - P_{dabartinis}(i)|; \quad (11)$$

čia  $P_{skirtuminis}(i)$  – gaunamas skirtuminis kadras,  $P_{buvęs}(i)$  – prieš dabartinį ejęs kadras,  $P_{dabartinis}(i)$  – dabartinis kadras.

Modulis skaičiuojant absoliutinį skirtumą reikalingas tam, kad būtų fiksuojamas pokytis tarp pikselių, išlaikant intensyvumo vertę tinkamame diapazone – nuo 0 iki 255.

### 3.4.3. Objektų išplėtimas

Objektų išplėtimo operacija suteikia galimybę paveiksle užpildyti kiaurymes, sujungti greta esančius plotus – maži skirtumai apskaičiuavus absoliutinį skirtumą tarp kadru yra didinami tam, kad būtų lengviau atpažinti objektų kontūrus atliekant vėlesnius metodikos žingsnius [30]. Objektų kontūrai paveiksle plečiami pasinaudojant pilkų atspalvių paveikslo tipo objektų išplėtimo formule:

$$(P \oplus f)(x) = \sup_{y \in E} [P(y) + f(x - y)]; \quad (12)$$

čia  $P$ ,  $P(x)$  bei  $P(y)$  – paveikslas ir atitinkamai paveikslo su išplėtais objektais ir su nemodifikuotais objektais pikselis,  $f(x-y)$  – struktūrinė funkcija,  $\sup_{y \in E}$  – supremumas, išrenkamas iš tam tikro kiekio arba visos aibės pikselių, kai renkama vertė tam pikseliui  $y$ , kuris priklauso nurodytai aibei  $E$ .

Pasirinkta objektus išplėsti pasinaudojant tolygia struktūrinio elemento funkcija  $f$ :

$$f(x) = \begin{cases} 0, & x \in B, \\ 255, & \text{kitaip,} \end{cases}; \quad B \in E; \quad (13)$$

čia  $f(x)$  – struktūrinio elemento / branduolio pikselis,  $B$  – tam tikras kiekis pikselių, kuris priklauso nurodai aibei  $E$ ,  $E$  – visi paveikslo pikseliai.

Sumodeliavus branduolį, kurio pavidalas yra  $5 \times 5$  dydžio vienetų matrica, ir pasirinkus gražinti ne supremumą, o maksimalaus intensyvumo vertę, objektų išplėtimo formulė supaprastėja:

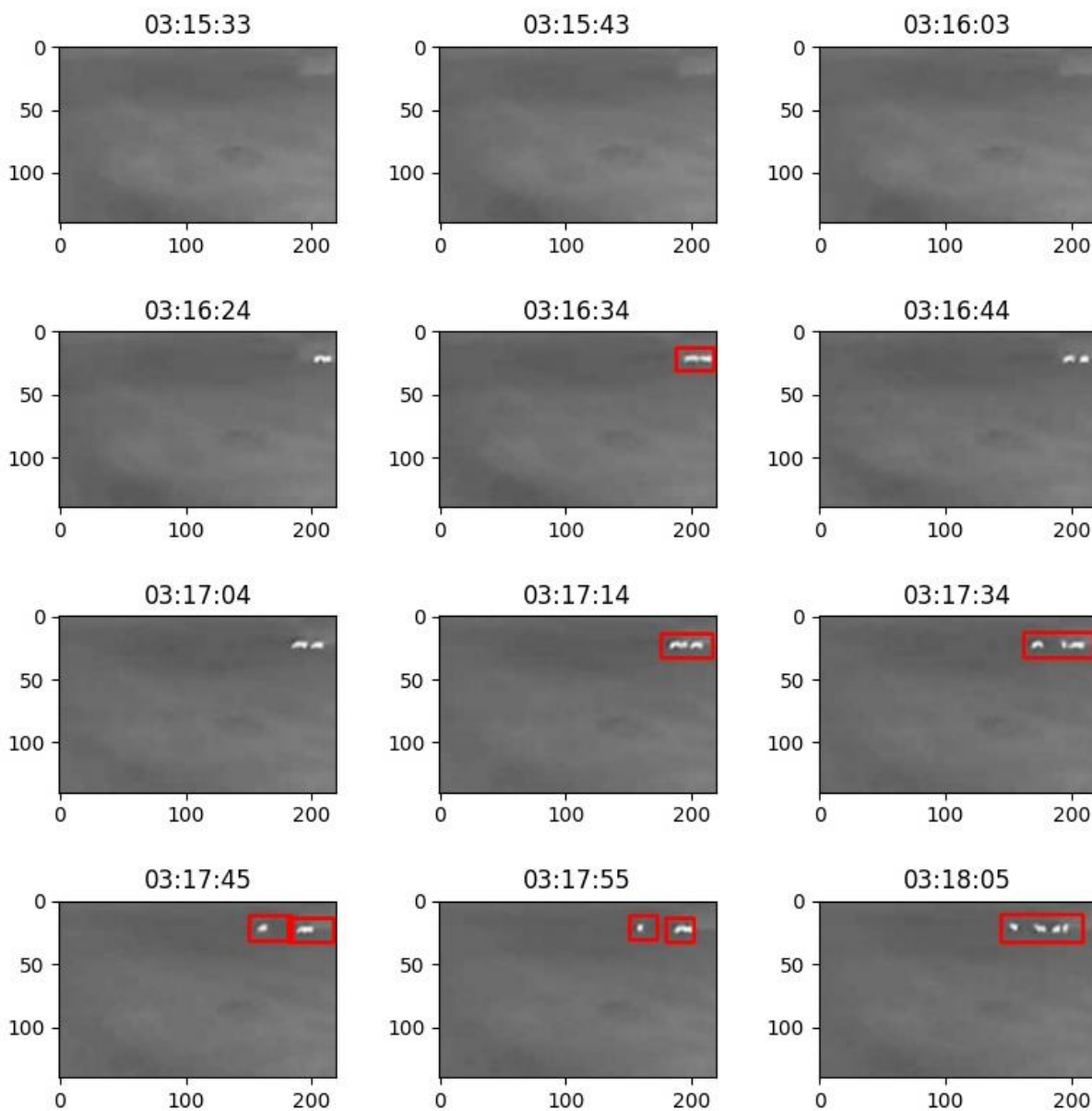
$$(P \oplus f)(x) = \sup_{y \in E} [P(y) + f(x - y)] = \max_{z \in B} [P(z)]; \quad (14)$$

čia  $P(y)$  ir  $P(z)$  – atitinkamai sena pikselio intensyvumo vertė bei pikselių vertės, dengiamos branduolio,  $f(x-y)$  - struktūrinio elemento funkcija,  $\max_{z \in B}$  – maksimalios pikselių vertės.

Objektų išplėtimo metodas veikia judančio lango principu [31]. Branduolys yra slenkamas per paveikslą ir nauja pikselio, esančio branduolio centre, vertė nustatoma parenkant maksimalią intensyvumo vertę iš pikselių, dengiamų branduolio. Atlikus šiuos veiksmus, yra pritaikomas slenkščio nustatymo algoritmas binariniam paveiksliui gauti, aprašytas 3.1.2 skyrelyje, kontūrų objekte nustatymas sekant 3.1.3 skyrelyje aprašytu principu ir pritaikomas 3.1.4 apibūdintas algoritmas, kuriame atpažįstami gyvūnams priklausantys kontūrai.

#### 3.4.4. Rezultatų aptarimas

Iliustracijoje nr. 27 pavaizduoti kadrai, fiksuoti kas 10 – 20 sekundžių, ir analizuoti judesio aptikimo metodika. Visuose kadruose aptinkamų ir judančių gyvūnų objektai pažymėti raudonais stačiakampiais. Stebint vienas po kito einančius kadrus (žr. 27 pav.) galima matyti, kad antroje eilutėje pirmame ir paskutiniame kadre bei trečioje eilutėje pirmame kadre nėra pažymėti gyvūnai, nors jų kontūrai nuotraukose yra matomi. Taip yra dėlto, kad gyvūno objekto kontūro plotą nusakantis stačiakampis kertasi su kadro kraštu. Dėlto galima teigti, kad kai prieš tai buvusiame kadre gyvūnas nebuvo aptiktas, o dabartiniame kadre buvo aptiktas, tačiau jo objekto plotas ribojasi su kadro kraštu – gyvūnas pirmą kartą įėjo į kameros aprėpiamą vaizdą ir apie jo judėjimą galima spręsti iš būsimų kadru (nusakoma situacija tarp kadru 03:16:03 ir 03:16:24).



**27 pav.** Judesio aptikimo metodika analizuoti kadrai

Kai prieš tai buvusiam kadre gyvūnas buvo aptiktas ir pastebėtas jo judėjimas, o dabartiniame kadre gyvūnas yra ties kameros aprėpiamo vaizdo riba (žr. 27 pav.), galima teigti, kad gyvūnas juda link stebimo lauko krašto dešinėn tolyn nuo pavojingos gatvės kairėje (nusakoma situacija – tarp kadru 03:16:34 ir 03:16:44 ir tarp kadru 03:16:44 ir 03:17:04) – dėlto pavojus nėra fiksuojamas.

### **3.5. Metodikos pasirinkimas**

Atlikus gyvūnų objektų atpažinimą nuotraukose skirtingomis metodikomis, tam, kad būtų galima išskirti vieną metodiką, realizuojamą įterptinėje sistemoje, buvo parinkti parametrai, aprašomi tolimesniame skyrelyje.



### 3.5.1. Metodikų įvertinimo parametrai

Norint palyginti metodikas, buvo parinkti šie parametrai:

1. vidutinis metodikos vykdymo laikas sekundėmis;
2. metodikos jautrumas procentais;
3. metodikos specifiškumas procentais;
4. metodikos tikslumas procentais.

Vidutinis metodikos vykdymo laikas apima laiką, kurį užtrunka metodiką realizuojantis programuotas algoritmas, kaip įėjimą algoritme priimantis nuotrauką, o kaip išėjimą turintis atsaką, ar gyvūnas nuotraukoje pastebėtas, ar ne. Vidutinio laiko matavimo formulė pateikiama toliau:

$$laikas_{vid} = \frac{\sum_{i=1}^N t}{N}; \quad (15)$$

čia  $N$  – bandymų kiekis,  $t$  – vieno bandymo trukmė sekundėmis.

Metodikos jautrumas, specifiškumas ir tikslumas yra statistiniai matai, tyrimo atveju kiekybiškai įvertinantys, kiek metodika yra tinkama naudoti realiame scenarijuje [32]. Jautrumo matas nusako, kaip tinkamai metodika įvertina teigiamus rezultatus – gyvūno objekto aptikimą; skaičiavimo formulė pateikiama toliau:

$$jautrumas = \frac{TP}{TP+FN}; \quad (16)$$

čia  $TP$  (angl. *True Positive*) – teisingų teigiamų įverčių kiekis,  $FN$  (angl. *False Negative*) – neteisingų neigiamų įverčių kiekis. Specifiškumo matas parodo, kaip tinkamai metodika neatpažįsta gyvūnų objektų ten, kur jų iš tiesų nėra – kitaip tariant, kaip gerai įvertina ir nuotraukas, kuriose nėra gyvūnų.

$$specifiškumas = \frac{TN}{TN+FP}; \quad (17)$$

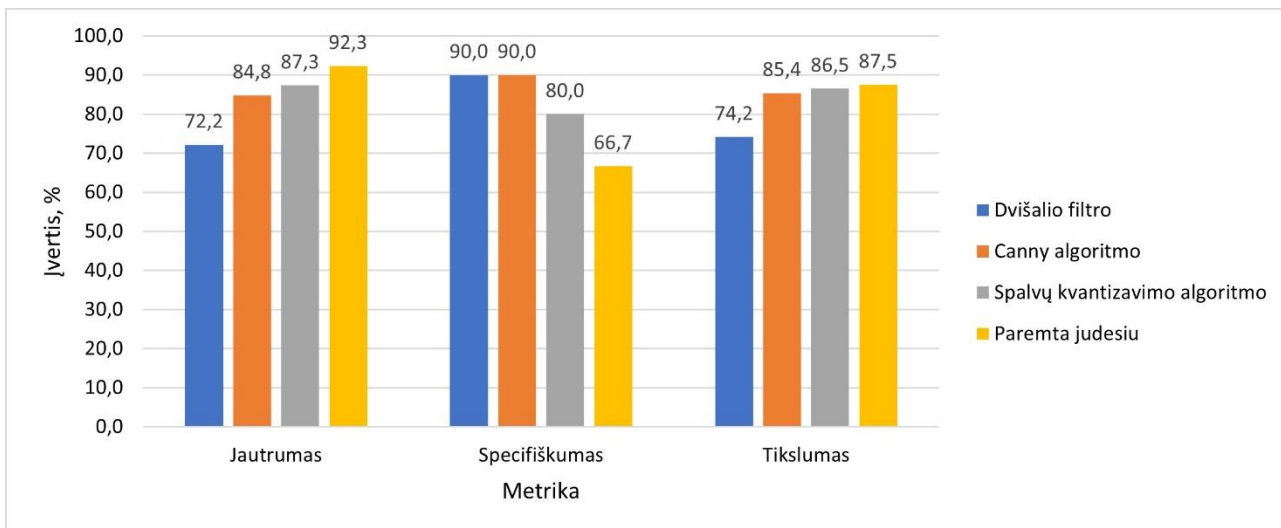
čia  $TN$  (angl. *True Negative*) – teisingų neigiamų įverčių kiekis,  $FP$  (angl. *False Positive*) – neteisingų teigiamų įverčių kiekis. Tikslumo matas nusako, kiek teisingai metodika aptinka gyvūnų objektų, atsižvelgiant į visus vertinimo rezultatus. Skaičiavimo formulė pateikiama toliau:

$$tikslumas = \frac{TN+TP}{TN+TP+FN+FP}; \quad (18)$$

čia  $TN$  – teisingų neigiamų įverčių kiekis,  $TP$  – teisingų teigiamų įverčių kiekis,  $FN$  – neteisingų neigiamų įverčių kiekis,  $FP$  – neteisingų teigiamų įverčių kiekis.

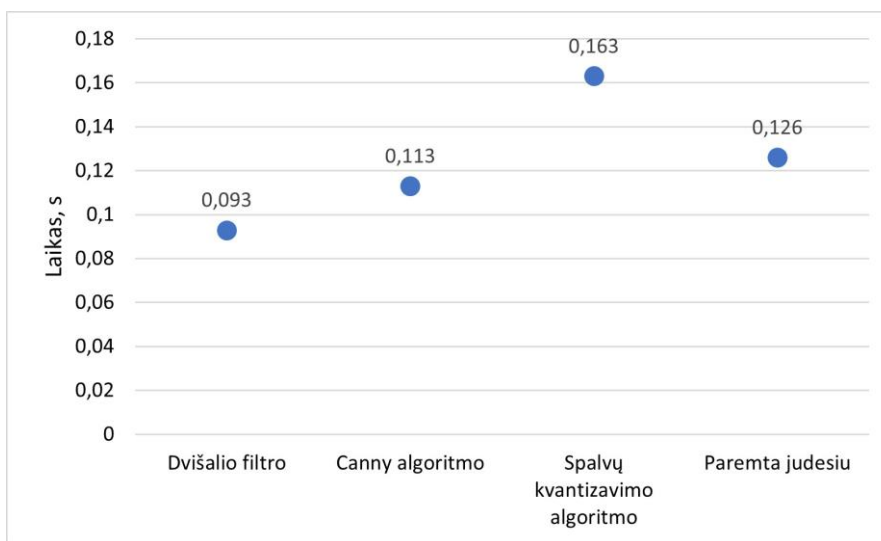
### 3.5.2. Metodikų tyrimas pagal parametrus

Pasinaudojant ankstesniame poskyryje aprašytais parametrais, buvo iširtos įgyvendintos metodikos. Buvo parinkti 89 kadrai, einantys vienas po kito, kadrai buvo įvertinti stebėtojo akimi ir parinkta tokia seka, kurioje gyvūnai judėtų į ir iš kadro, t.y., kad dalyje kadrų gyvūnai matomi, dalyje – ne. Pasirinkti paveikslai buvo paduodami į kiekvienos metodikos vykdymo algoritmo įėjimą. Dėlto galima lyginti metodikas, apdorojančias vieną nuotrauką, su metodika, kurioje analizuojama kadrų – prieš tai ėjusio ir dabartinio – pora. Toliau pateikiami metodikų tyrimo rezultatus nusakantys grafikai, kuriuose pavaizduoti metodikų įverčiai pagal metrikas bei vidutinę metodikų algoritmų atlikimo trukmę.



28 pav. Metodikų tyrimo pagal metrikų įverčius grafikas

Viršuje pavaizduotame metodikų tyrimo įverčių grafike pagal metrikas galima matyti, jog didžiausią jautrumą turi judesiu paremta metodika, didžiausią specifiškumą turi dvišalio filtro ir Canny algoritmo metodikos, o didžiausiu tikslumu pasižymi dvi – spalvų kvantavimo algoritmo bei judesiu paremta – metodikos.



29 pav. Metodikų tyrimo pagal laiką grafikas

Metodikų tyrimo pagal laiką grafike 29 pav. galima matyti, jog greičiausiai veikiančią programinę algoritmą turi dvišalio filtro metodika, lėčiausiai – spalvų kvantizavimo metodika. Toliau pateikiama metodikų tyrimo rezultatus apibendrinanti lentelė su tiksliais įverčių reikšmėmis.

2 lentelė. Metodikų tyrimo parametrais rezultatus apibūdinanti lentelė

Metodika	Parametrai			
	Vidutinis laikas, s	Jautrumas, %	Specifiškumas, %	Tikslumas, %
Dvišalio filtro	0,093	72,15	90,00	74,15
Canny algoritmo	0,113	84,81	90,00	85,39
Spalvų kvantizavimo algoritmo	0,163	87,34	80,00	86,51
Paremta judesiu	0,126	92,31	66,67	87,50

Įvertinus realizuotas metodikas aprašytais parametrais, lentelėje nr. 2 pateikiamuose rezultatuose galima matyti, jog vidutinis laikas metodikoms atlikti kinta diapazone 0,093 – 0,163 s, lėčiausiai ir greičiausiai veikiančių algoritmų metodikas skiria 70 ms. Jautrumo metrikos įverčiai skirtingoms metodikoms kinta 72,15 – 92,31 % diapazone, jautriausia metodikos įvertį turi metodika, paremta judesiu. Specifiškiausios metodikos ties 90 % yra dvišalio filtro ir Canny algoritmo, mažiausiai specifiška yra metodika, paremta judesiu, ties 66,67 %. Didžiausią tikslumą pasiekė metodika, paremta judesio atpažinimu, turinti 87,50 % įvertį; antroje vietoje yra spalvų kvantavimo metodika, kurios tikslumas – 86,51 %.

Vertinant gautus rezultatus ir kiekvienos individualios metodikos kūrimo ypatumus, pasirinkta įterptinėje sistemoje realizuoti metodiką, paremtą judesiu. Ši metodika turi didžiausią jautrumo įvertį, dėlto galima teigti, kad ji aptinka gyvūnus nuotraukose, kuriuose jie iš tiesų egzistuoja, geriausiai, lyginant su kitomis metodikomis. Taip pat šios metodikos tikslumas yra didžiausias – tai parodo, kad ši metodika, lyginant su kitomis metodikomis, tiksliausiai aptinka gyvūnų objektus nuotraukose.

#### 4. Metodikos realizavimas eLinux sistemoje

Pasirinkta judesiu paremta metodika buvo realizuota įterptinėje sistemoje (ĮS). Įterptinės sistemos pasižymi tuo, kad yra sukurtos atlikti specifinę užduotį ar užduočių grupę ir nėra skirtos vykdyti visapusiškas užduotis. Metodika gyvūnams aptikti turi vieną paskirtį – realiu laiku pranešti, kada kameros gaunamoje nuotraukoje pastebimas gyvūnas. Dėlto tinkamai metodikos integracijai į ĮS reikalinga komunikacija su kamera bei bevielio ryšio tinklas, kuriuo gauta informacija siunčiama į duomenų bazę bei signalai perduodami perspėjamiems atšvaitams. Pasirinkta PYNQ plokštė, turinti mikrovaldiklį bei FPGA plokštę, skirtą akceleruoti algoritmus, ir jungtį tinklo kabeliui [33].

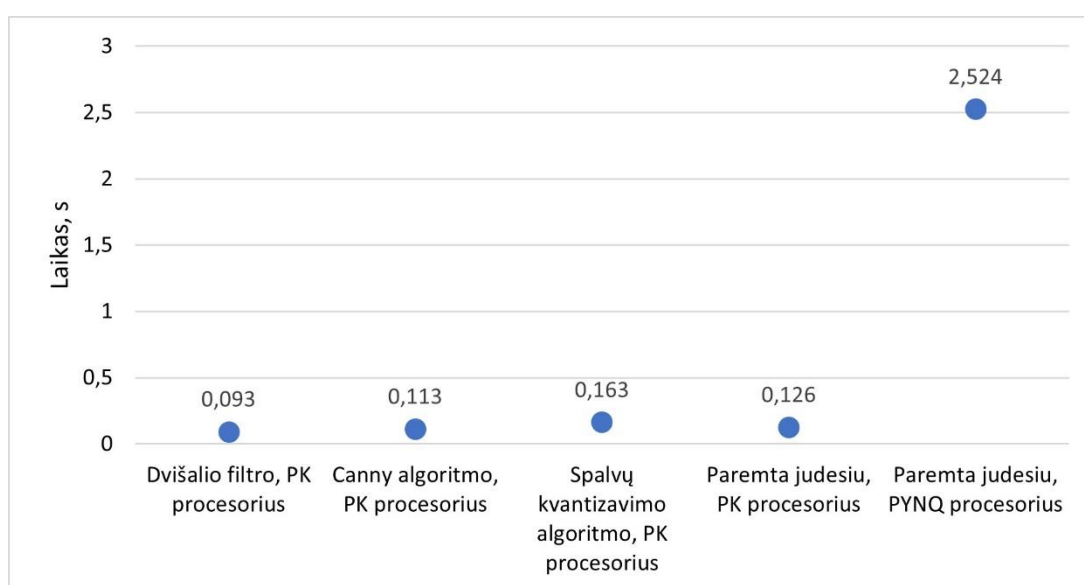
##### 4.1. Metodikos realizavimas įterptinės sistemos mikroprocesoriuje

Metodikų kūrimo ir įvertinimo bei palyginimo etape jos buvo realizuojamos personaliniame kompiuteryje. Tuomet pasirinkta metodika buvo perkelta į PYNQ plokštę ir realizuota veikti šios plokštės mikrovaldiklyje [34]. Personalinio kompiuterio ir PYNQ plokštės specifikacijų palyginimo lentelė pateikiama toliau.

3 lentelė. Personalinio kompiuterio ir PYNQ plokštės specifikacijų palyginimo lentelė

Specifikacija	Personalinis kompiuteris	PYNQ plokštė
Operacinė sistema	Windows 10	Debian GNU/Linux
Procesoriaus branduolių kiekis	4	2
Procesoriaus CPU dažnis, GHz	1.60	1
RAM atminties kiekis	8 GB	512 MB

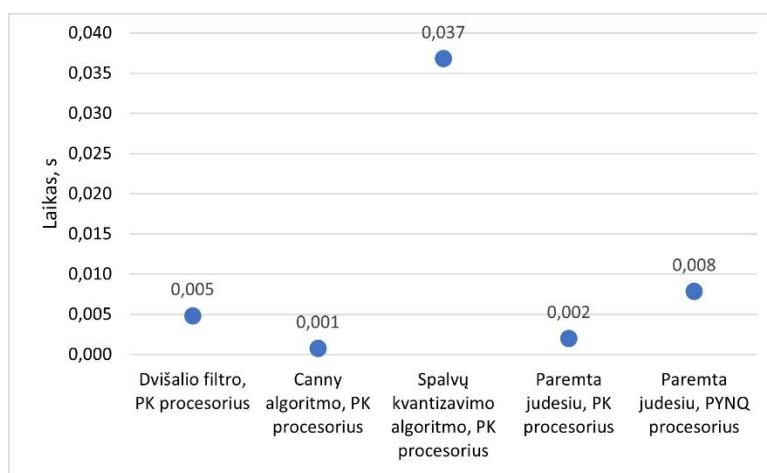
Kaip galima matyti lentelėje nr. 3, personalinio kompiuterio procesoriaus branduolių skaičius yra dvigubai didesnis už PYNQ plokštėje esančio Dual ARM Cortex – A9 mikrovaldiklio, kuris turi du branduolius. Personaliniame kompiuteryje esančio procesoriaus maksimalus CPU dažnis yra 600 MHz didesnis nei PYNQ plokštėje esančio mikrovaldiklio; RAM tipo atmintis personaliniame kompiuteryje – 8 GB, PYNQ plokštei priklausančiam mikrovaldikliui – 512 MB, mažesnė apie 15,6 kartų. Minėtos specifikacijos turi įtakos vidutiniam metodikos algoritmo vykdymo laikui – algoritmas trunka ilgiau, kai yra naudojamas PYNQ plokštė.



30 pav. Metodikų algoritmo trukmės skirtinguose procesoriuose palyginamasis grafikas

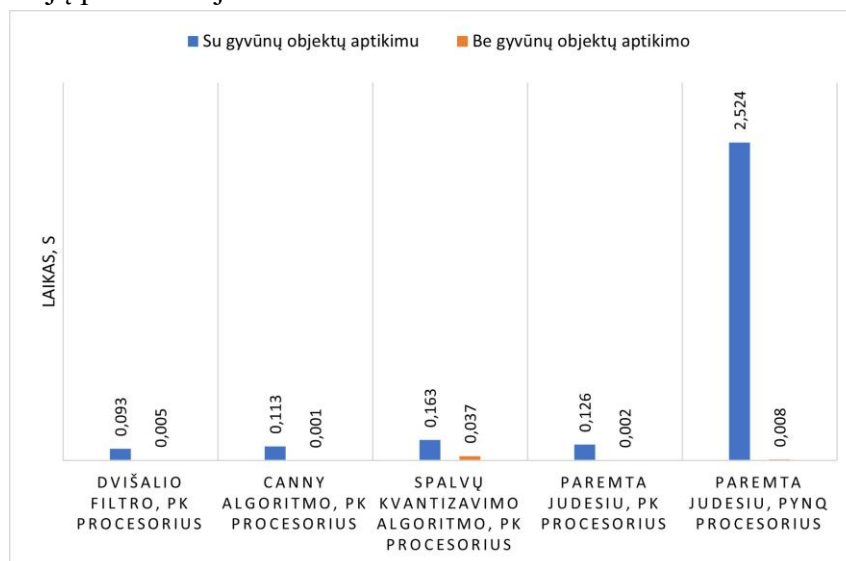
Apžvelgiant vidutinius metodikų vykdymo laikus skirtinguose procesoriuose, 30 pav. (čia PK – personalinio kompiuterio) galima matyti, jog pasirinktos ĮS realizuoti metodikos vykdymo laikas yra apie 2,5 sekundės – tiek nuo tos pačios metodikos, tiek ir nuo kitų metodikų vykdymo laiko personalinio kompiuterio procesoriuje skiriasi apie 21 kartą. Taip yra dėl minėtų skirtingų procesorių specifikacijų - CPU dažnių ir branduolių kiekio.

Buvo nuspręsta patikrinti, kuri metodikos dalis užima daugiausiai jos algoritmo vykdymo laiko. Kiekvienos metodikos eigoje paskutinis žingsnis yra gyvūnų objektų atpažinimas (žr. 3.1.4 skyrelį). Dėlto nuspręsta pakoreguoti algoritmus ir analizuoti jų vykdymo laiką be gyvūnų objektų iš nuotraukose aptiktų elementų kontūrų radimo. Toliau esančiame paveiksle pateikiamas metodikų vykdymų laikas skirtinguose procesoriuose.



**31 pav.** Metodikų algoritmo trukmės skirtinguose procesoriuose, kai nevykdoma gyvūnų objektų aptikimo dalis, palyginamasis grafikas

Kaip galima matyti 31 pav., nevykdant gyvūnų objektų aptikimo dalies ilgiausiai – 36,8 ms – trunka spalvų kvantizavimo algoritmo metodika, veikianti personalinio kompiuterio procesoriuje. Antroje vietoje yra paremta judesiu metodika, kuri buvo parinkta realizuoti įterptinėje sistemoje – jos vykdymo laikas yra 7,86 ms, kai algoritmas vyksta ĮS procesoriuje, ir apie 2 ms, kai algoritmas vyksta didesnių specifikacijų procesoriuje.



**32 pav.** Metodikų algoritmo trukmės skirtinguose procesoriuose, kai vykdoma ir kai nevykdoma gyvūnų objektų aptikimo dalis, palyginamasis grafikas

Apibendrinant metodikų vykdymo laikus skirtinguose procesoriuose tiek atliekant, tiek neatliekant gyvūnų objektų aptikimą, 32 pav. galima matyti gautas trukmes. Ilgiausiai vykdant gyvūnų objektų aptikimą trunka judesiu paremtos metodikos algoritmas, kuris pasirinktas realizuoti PYNQ ir veikiantis šios plokštės mikroprocesoriuje, trumpiausiai – dvišalio filtro metodikos algoritmas. Kai gyvūnų objektų algoritmas nevykdomas, ilgiausiai trunka spalvų kvantizavimu paremtos metodikos algoritmas, trumpiausiai – Canny algoritmu paremtos metodikos algoritmas. Galima teigti, kad trukmių išsidėstymai yra neproporcingi, t.y. vykdant gyvūnų objektų aptikimą ir be jo, algoritmų vykdymų laikas sumažėja nevienodai. Taip yra dėlto, kad skirtingos metodikos tiek teisingai, tiek neteisingai atpažįsta skirtingą kiekį gyvūnų. Kalbant apie metodikas, paremtas judesiu, algoritmo realizavimą PYNQ plokštėje esančiame mikrovaldiklyje, toks skirtumas tarp trukmių, kai vykdomas ir nevykdomas gyvūnų aptikimas, matomas dėl to, kad algoritmas, kuris naudojamas gyvūnams aptikti (žr. 3.1.4 skyrelį), yra intensyvus atliekamais skaičiavimais atskiruose žingsniuose (angl. *computationally intensive*).

#### 4.2. Metodikos dalių akceleravimas FPGA

Atlikus algoritmų vykdymo trukmių tyrimą skirtinguose procesoriuose, buvo nuspręsta dalį pasirinktos metodikos žingsnių akceleruoti pasitelkiant FPGA. Apibendrintas metodikos veikimo algoritmas su dalimi, kurioje veiksmai akceleruojami FPGA, yra toliau.



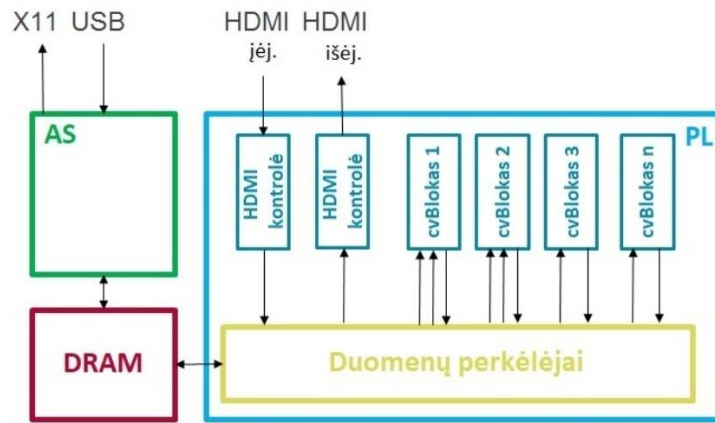
33 pav. Apibendrintas judesio aptikimu paremtos metodikos algoritmas akceleracijai su FPGA

Algoritmas, paremtas judesio aptikimu, detaliam paaiškintam 3.4 poskyryje. Metodikos algoritme (žr. 33 pav.) išskirta operacijų su kadru dalis – tai yra segmentacijos ir morfologinės operacijos, kurių metu yra apdorojamas ir tolimesniems žingsniams ruošiamas kadras. Iš minėtų rūšių operacijų akceleracijai į FPGA perkeltos filtravimo Gauso filtru ir objektų išplėtimo operacijos.

Akceleracijai FPGA atlikti yra pasitelkiamos paviršinės aparatinės įrangos bibliotekos (angl. *overlays or hardware libraries*). Šių bibliotekų pagalba algoritmai yra perkeliami iš procesoriaus į programavimo logiką – FPGA; jos naudojamos akceleruoti programinę įrangą arba pritaikyti aparatinę įrangą konkrečiam uždaviniui spręsti.

Atliekant judesio aptikimu paremtos metodikos dalies akceleraciją FPGA, pasinaudota xfOpenCV – kompiuterinės vizijos paviršinėmis aparatinės įrangos bibliotekomis, skirtomis dirbti su PYNQ plokštėje esančia programine logika. Valdėti šias bibliotekas aplikacijoje galima Python programavimo kalba, taip pat jose palaikomi algoritmai, akceleruojami FPGA – filtravimas 2D filtrais, objektų mažinimo operacijos.

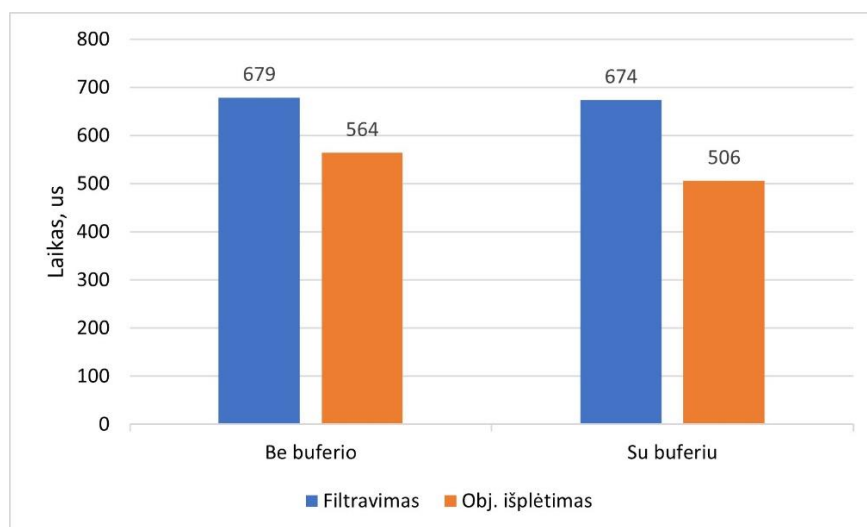
Paviršinės aparatinės įrangos bibliotekos veikimo principą paaiškinanti schema pavaizduota toliau.



34 pav. xfOpenCV bibliotekos veikimą nusakanti schema [35]

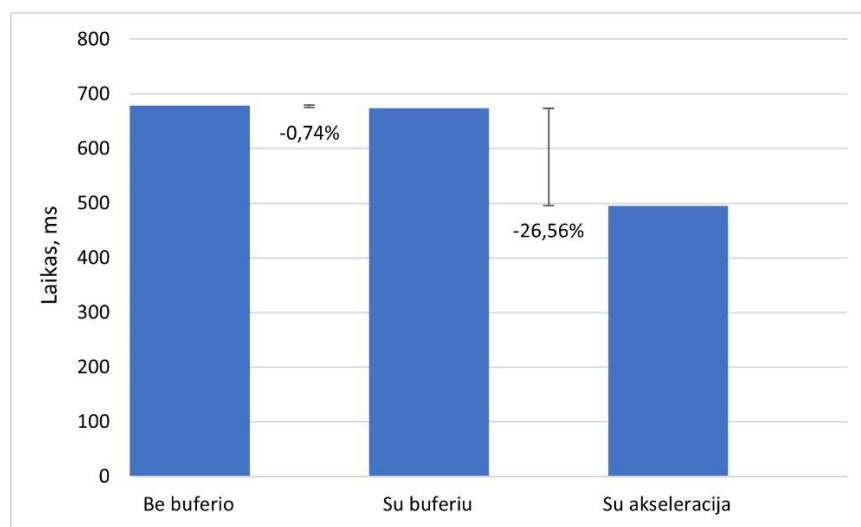
Tiek xfOpenCV, tiek kitos paviršinės aparatinės įrangos bibliotekos sąveikauja su AS – apdorojimo sistema (angl. *processing system*) ir PL – programavimo logika (žr. 34 pav.). Algoritmas vykdomas apdorojimo sistemoje, programinis algoritmo kodas įkeliamas į DRAM atmintį. Šioje atmintyje taip pat saugomi ir algoritmo vykdymo metu reikalingi objektai, pvz. nuotrauka, signalas ir pan. Programuojamoje logikoje HDMI kontrolės procesų pagalba galima priimti HDMI signalą ir apdorotus perduoti į išėjimą, pvz.: atvaizduoti nuotrauką ar signalą. Duomenų surinkimo sistemoje nuotraukos priimamos tinklo kabeliu, dėlto HDMI funkcionalumas nebuvo panaudotas. Bibliotekos programinio kodo išdėstymas programuojamoje logikoje atliekamas cvBlokų pagalba, DRAM atmintyje egzistuojantys objektai pasiekiami duomenų perkėlėjais, juose laikomi ir HDMI įėjime priimami signalai, jeigu naudojamas HDMI funkcionalumas. Paviršinės įrangos bibliotekos procesus, išdėstomus programuojamoje logikoje, bei duomenų paskirstymą ir jiems rezervuojamą vietą perkėlėjuose nusako apdorojimo sistemoje vykdomas pagrindinis algoritmas.

Prieš atliekant Gauso filtro ir objektų išplėtimo operacijų akceleraciją, buvo iširta, kiek įtakos metodikos algoritmui turi buferių naudojimas. Buferiu laikoma vieta atmintyje, kuri nedaloma rezervuojama tam tikram resursui laikyti (angl. *contiguous allocated memory*). Palyginamasis grafikas, kuriame vaizduojama filtravimo ir objektų išplėtimo operacijų trukmė, kai neatliekama ir kai atliekama buferizacija, pateikiamas toliau.



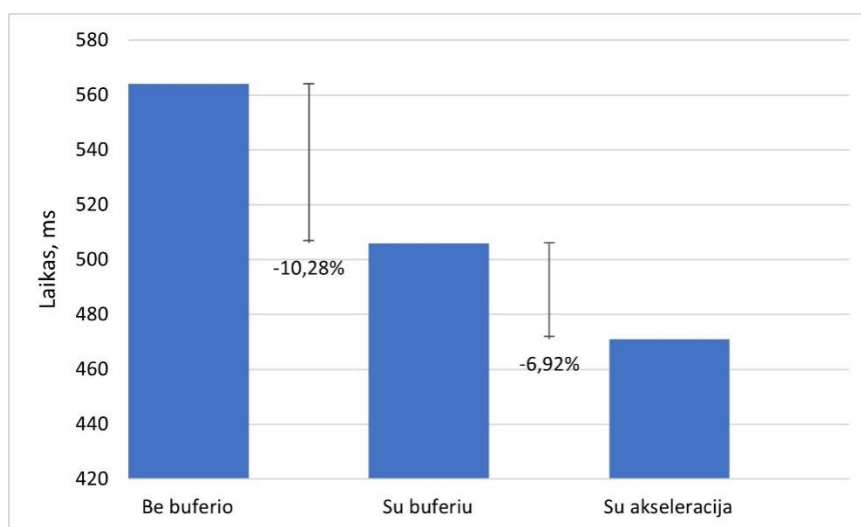
35 pav. Metodikos algoritmo trukmės neatliekant buferizacijos ir buferizuojant skirtingus procesus palyginamasis grafikas

Kaip galima matyti 35 pav., atliekant buferizaciją, tiek filtravimo, tiek objektų išplėtimo operacija trunka trumpiau. Galima teigti, kad dėlto, jog nedalomai išskiriama atmintis nuotraukoms prieš jas apdorojant ir po apdorojimo, algoritmo vykdymas trunka trumpiau, nes tokiu atveju atminties paskirstymas nėra išskaidomas ir tampa paprasčiau nuskaityti pikselių rinkinį. Lyginant buferizuotus filtravimo ir objektų išplėtimo procesus galima matyti, kad buferizuotas objektų išplėtimo procesas trunka trumpiau nei buferizuotas filtravimo procesas. Taip yra dėlto, kad nors objektų išplėtimo procesas yra sudėtingesnė skaičiavimais operacija, tačiau palyginus su filtravimo procesu, apdorojamų pikselių kiekis yra mažesnis plečiant objektus. Dėlto pačių operacijų kiekis atliekant objektų plėtimą yra mažesnis – papildomai atliekant buferizaciją išryškėja skirtumas tarp minėtosios ir filtravimo operacijų. Palyginamieji filtravimo ir objektų išplėtimo operacijų trukmių grafikai, kuomet neatliekama nei akceleracija, nei buferizavimas, kai atliekamas tik buferizavimas ir kai atliekami tiek buferizavimas, tiek akceleracija, pateikiami 36 – 37 pav.



**36 pav.** Filtravimo operacijos trukmės negreitinant, buferizuojant ir greitinant bei buferizuojant palyginamasis grafikas

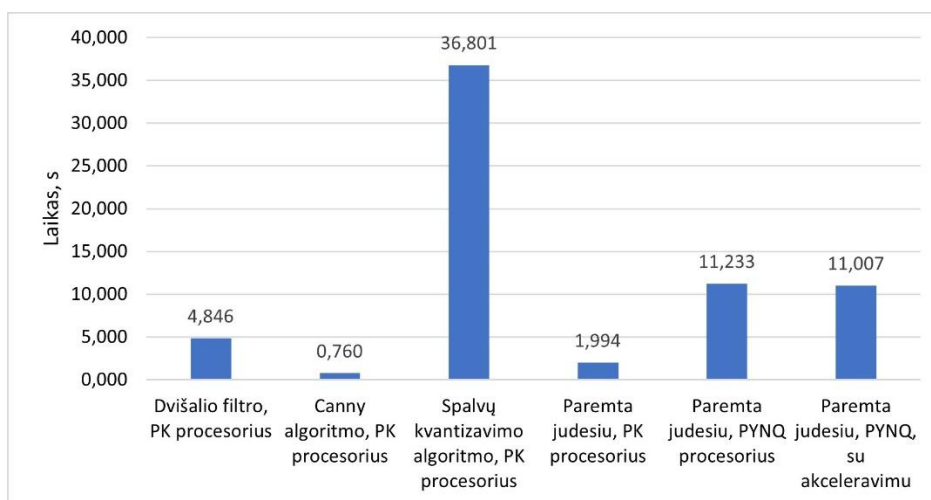
Atlikus buferizavimą ir akceleraciją filtravimo procesui, 36 pav. galima matyti, kad akceleruojant filtravimo operacija pagreitinama apie 28 %, lyginant su neakceleruotu ir nebuferizuotu filtravimu, o lyginant su tik buferizuotu filtravimo procesu, akceleravimas suteikia apie 1 % pagreitėjimą.



**37 pav.** Objektų išplėtimo operacijos trukmės negreitinant, buferizuojant ir buferizuojant bei greitinant palyginamasis grafikas



Kaip galima matyti 37 pav., buferizuotas ir pagreitintas objektų išplėtimo procesas veikia apie 32 % greičiau, lyginant su neakceleruotu ir nebuferizuotu objektų išplėtimu, o lyginant su tik buferizuotu objektų išplėtimo procesu, akceleravimas suteikia apie 7 % pagreitėjimą. Visų tirtų metodikų, realizuotų personalinio kompiuterio procesoriuje, bei judesiu paremtos metodikos, realizuotos personalinio kompiuterio procesoriuje, PYNQ procesoriuje ir akceleruotos FPGA, vykdymų trukmių palyginamasis grafikas pateikiamas toliau.



**38 pav.** Metodikų algoritmo trukmės skirtinguose procesoriuose ir FPGA, nevykdant gyvūnų objektų aptikimo dalies, palyginamasis grafikas

Lyginant vykdymo trukmes tirtų metodikų algoritams (žr. 38 pav.) galima teigti, kad akceleruojant dalį judesiu paremtos metodikos algoritmo, jos vykdymo trukmė yra didesnė už kitų metodikų, išskyrus spalvų kvantizavimo metodikos algoritmo bei tos pačios judesiu paremtos metodikos algoritmo, vykdyto įterptinės sistemos mikroprocesoriuje, vykdymo trukmę. Apibendrinanti metodikų vykdymo trukmes lentelė pateikiama toliau.

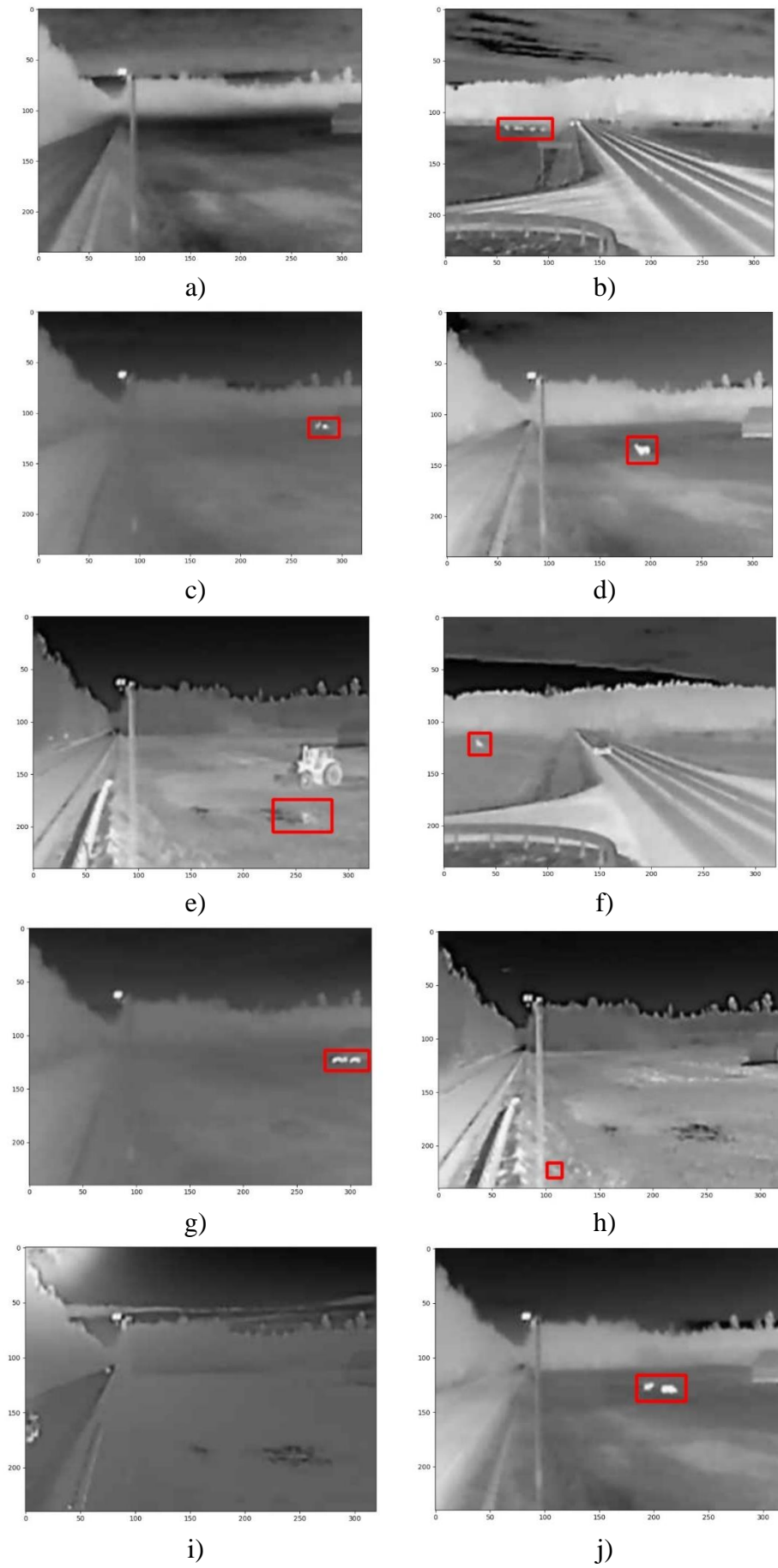
**4 lentelė.** Metodikų vykdymo laikų trukmių lentelė

Metodikos	Vidutinis vykdymo laikas, ms
Dvišalio filtro, personalinio kompiuterio procesorius	4,85
Canny algoritmo, personalinio kompiuterio procesorius	0,76
Spalvų kvantizavimo algoritmo, personalinio kompiuterio procesorius	36,8
Paremta judesiu, personalinio kompiuterio procesorius	1,99
Paremta judesiu, įterptinės sistemos procesorius	11,23
Paremta judesiu, akceleruota FPGA	11,00

Kaip galima matyti lentelėje nr. 4, trumpiausiai veikianči išliko Canny algoritmu paremta metodika. Pasirinktą metodiką, paremtą judesiu, realizuoti įterptinėje sistemoje esančiame procesoriuje ir dalį metodikos algoritmo perkelti į FPGA, taip pagreitinant metodikos vykdymo laiką, pavyko.

### 4.3. Metodikos tyrimai, aptinkant gyvūnus įvairiose nuotraukose

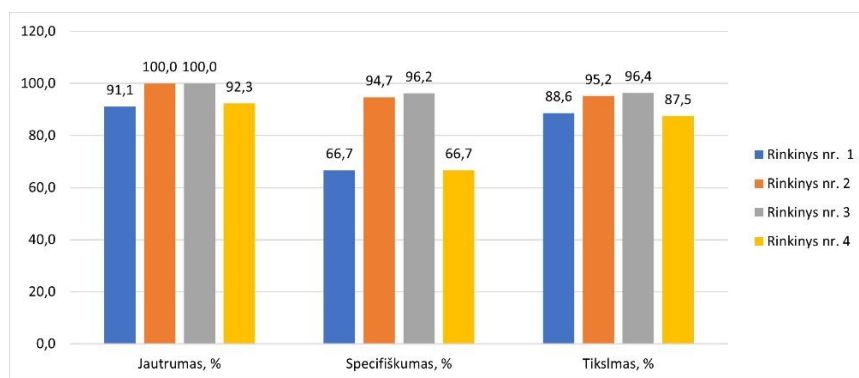
Atlikus pasirinktos metodikos implementavimą įterptinėje sistemoje bei dalies procesų akceleravimą FPGA, nuspręsta panaudoti sukauptų nuotraukų rinkinius – iš jų išrinkta 10 skirtingas situacijas vaizduojančių nuotraukų ir atliktas tyrimas, parodantis, kaip teisingai metodika geba išskirti gyvūnų objektus įvairaus tipo nuotraukose.



**39 pav.** Gyvūnų objektų tyrimo, atpažįstant juos skirtingose situacijose paremta judesiu metodika, rezultatai

Aptariant gautus rezultatus (žr. 39 pav.) galima teigti, kad daugumoje nuotraukų metodikos algoritmas geba atpažinti gyvūnų objektus. Tiek metų laikas, tiek paros laikas daugelyje matomų atvejų nedaro įtakos metodikos tikslumui. Taip yra todėl, kad dėl termovizorinės kameros ypatumo stebėti kūnų spinduliuojamą šilumą krituliai, paros laikas ir kiti trukdžiai, kurie galėtų varžyti algoritmo veikimą, šiuo atveju netrukdo. Taip pat yra užfiksuota atvejų, kai algoritmas atpažįsta ir nuotraukoje yra pažymimas itin mažas arba neaiškiai matomas gyvūnas (žr 39 pav., e ir h dalys). Taip nutikus, reikėtų papildomai apdoroti nuotrauką stengiantis padidinti jos raišką, kad realia žmogaus akimi būtų galima stebėti, ar nuotraukoje vis dėlto iš tiesų yra gyvūnas, arba priskirti tokią nuotrauką prie neteisingai atpažintųjų sąrašo, kadangi dėl nuotraukos raiškos nėra įmanoma nusakyti joje pastebimo objekto pobūdžio. Apibendrinant galima daryti išvadą, kad 6 iš 10 įvairaus paros ir metų laiko nuotraukų gyvūnas buvo atpažintas teisingai (žr 39 pav., b, c, d, f, g, j dalys). Taip pat 2 iš 10 nuotraukų algoritmas atpažino gyvūną, tačiau neatliekant tolimesnio apdorojimo neįmanoma nusakyti, ar atpažintas objektas iš tiesų yra gyvūnas (žr 39 pav., e, h dalys); 2 iš 10 nuotraukų, kuriose gyvūnai nepastebėti, algoritmas gyvūnų objektų taip pat neaptiko (žr 39 pav., a, i dalys).

Atlikus eksperimentą – metodika ištyrus 10 skirtingų situacijų vaizduojančias nuotraukas – buvo parinkti 4 duomenų rinkiniai ir atliktas metodikos tyrimas norint patikrinti, kaip metodika geba skirti gyvūnų objektus kadrų rinkiniuose, kuriuose nuotraukos gautos skirtingomis kameromis (žr. kamerų išdėstymą, 17 pav.) ir skiriasi nuotraukų su gyvūnų objektais kiekis. Tyrimo rezultatai toliau.



**40 pav.** Metodikos, paremtos judesiu, tyrimo skirtingais duomenų rinkiniais rezultatai

Kaip galima matyti tyrimo rezultatuose (žr. 40 pav.), metodikos įvertinimo parametrai gaunami skirtingi, kai keičiasi nuotraukų kiekis rinkinyje bei nuotraukų su gyvūnais kiekis rinkinyje.

**5 lentelė.** Nuotraukų bendrai ir nuotraukų su gyvūnais kiekį rinkiniuose apibūdinanti lentelė

	Rinkinys nr. 1	Rinkinys nr. 2	Rinkinys nr. 3	Rinkinys nr. 4
Kiek iš viso nuotraukų rinkinyje	89	84	84	17
Kiek iš viso nuotraukų su gyvūnais rinkinyje	80	8	4	13

Lyginant bendrą nuotraukų kiekį ir nuotraukų su gyvūnais kiekį rinkiniuose (žr. 5 lentelė) su atitinkamų metodikų parametru vertėmis galima teigti, kad aukštesnės metodikų parametru vertės gaunamos tiriant metodiką tais nuotraukų rinkiniais, kuriuose didžiausias nuotraukų kiekis, bet mažiausias nuotraukų su gyvūnais kiekis. Todėl galima daryti išvadą, kad kuo ilgiau gyvūnas būna teritorijoje, kuri yra stebima kamera, tuo labiau judesio metodika paremtas algoritmas gali neteisingai vertinti situaciją – gyvūno pozicija nuotraukoje turi kisti tam, kad būtų fiksuotas judėjimas ir metodikos algoritmas galėtų atpažinti gyvūno objektą.

## Išvados

1. Atlikus gyvūnų aptikimo ir stebėjimo metodikų apžvalgą, buvo nustatyta, kad įprastos metodikos reikalauja didesnių eksploatacijos išlaidų dėl sistemų specifiškumo, neretai yra invazinės savo principu dėl būtinybės pasitelkti mašinų ar žmonių darbą jas tvirtinti stebėjimo vietose arba dėl pačių sistemų tvirtinimo prie gyvūnų kūnų. Taip pat pastebėta, kad kuo daugiau sistemoje panaudojama jutiklių arba kuo didesnę informaciją neša patys jutiklių signalai, tuo pasiekiamas didesnis gyvūnų atpažinimo tikslumas.
2. Buvo iširtos 4 skirtingos metodikos, skirtos aptikti gyvūnų objektus nuotraukose. Judesiu paremta metodika yra jautriausia specifiškam sistemos uždaviniui, jos jautrumo matas siekia apie 91 %. Todėl galima daryti išvadą, kad ši metodika gražina geriausią teigiamą rezultatą, lyginant su kitomis metodikomis. Taip pat judesiu paremta metodika siekia apie 87 % tikslumą, dėlto galima teigti, kad ši metodika tiksliausiai aptinka gyvūnus nuotraukose. Todėl įterptinėje sistemoje nuspręsta realizuoti judesiu paremtą metodiką.
3. Pasirinkta metodika realizuota įterptinės sistemos procesoriuje, dalis jos žingsnių akceleruota FPGA. Lyginant metodikų, vykdytų personaliniame kompiuteryje, trukmes su įterptinėje sistemoje įgyvendintu algoritmu galima teigti, kad metodikų vykdymo trukmėms įtaką daro ne tiek pačių algoritmų vykdymo principas bei optimizavimas, bet ir realizavimo aplinka. Pasirinktos metodikos vidutinis vykdymo laikas įterptinės sistemos procesoriuje trunka apie 20 kartų ilgiau, lyginant su jos vykdymo trukme personaliniame kompiuteryje.
4. Akceleruojant judesiu paremtą metodiką FPGA, užtikrintas tam tikrų procesų lygiagretinimas – dėl to pasiekta greitesnė algoritmo vykdymo trukmė negu neakceleruojant nė vieno proceso; filtravimo procesas pagreitintas apie 26,5 %, objektų plėtimo funkcija pagreitinta apie 7 %.
5. Ištyrus judesiu paremtą metodiką kadrų rinkiniais, turinčiais skirtingus nuotraukų su gyvūnais ir be kiekius, gautas vidutinis metodikos tikslumas apie 91 %. Metodikos jautrumo, specifiškumo ir tikslumo įverčiai gauti žemesni tais atvejais, kai metodika tirta rinkiniu su didesniu nuotraukų su gyvūnais kiekiu. Dėlto galima teigti, kad metodikos gebėjimas teisingai vertinti gyvūno pasirodymą kameros stebimoje teritorijoje priklauso nuo to, kiek ilgai gyvūnas būna minėtoje teritorijoje. Gyvūnas turi judėti ir nuotraukoje turi kisti jo pozicija tam, kad algoritmas teisingai fiksuotų judesį ir galėtų aptikti gyvūnų objektus.

## Literatūros sąrašas

- [1] M. P. Huijser, P. T. McGowen, W. Camel, A. Hardy, P. Wright ir A. P. Clevenger, „Animal Vehicle Crash Mitigation Using advanced Technology Phase I: Review, Design and Implementation,“ Western Transportation Institute - Montana State University, Montana, 2006.
- [2] D. Forslund ir J. Bjärkefur, „Night Vision Animal Detection,“ *2014 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1-6, 8-11 Birželis 2014.
- [3] M. P. Huijser, T. D. Holland, M. Blank, M. C. Greenwood, P. T. McGowen, B. Hubbard ir S. Wang, „The Comparison of Animal Detection Systems in a Test-Bed: A Quantitative Comparison of System Reliability and Experiences with Operation and Maintenance,“ Federal Highway Administration; Montana Department of Transportation, Helena, Montana, 2009.
- [4] E. Maulana, C. S. Nugrho ir A. B. Dianisma, „Animal Presence Detection for Elephants and Extruding Method Based on Bee Frequency,“ *2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, pp. 1-4, 2018.
- [5] K. Nakagawa, D. Shimotoku, J. Kawase ir H. H. Kobayashi, „Dependable Wildlife DTN: Wearable Animal Resource Optimization for Sustainable Long-Term Monitoring,“ *2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application (DependSys)*, pp. 49-56, 2020.
- [6] M. Desholm, „Thermal Animal Detection System (TADS). Development of a method,“ *NERI Technical Report No 440.*, p. 27, 2003.
- [7] A. Mukherjee, S. Stolpner, X. Liu, U. Vrenozaj, C. Fei ir A. Sinha, „Large Animal Detection and Continuous Traffic Monitoring on Highways,“ *2013 IEEE SENSORS*, pp. 1-3, 3-6 Lapkritis 2013.
- [8] M. A. Mumma ir J. O. Abraham, „Elevated wildlife-vehicle collision rates during the COVID-19 pandemic,“ *Scientific Reports*, 2021.
- [9] T. Krukowicz, F. Krzysztof ir P. Chrobot, „Spatiotemporal Analysis of Road Crashes with Animals in Poland,“ 23 01 2022. [Tinkle] <https://www.mdpi.com/2071-1050/14/3/1253>. [Kreiptasi 17 05 2022].
- [10] E. COMmission, „European Commission,“ 2018 - 2023. [Tinkle] [https://webgate.ec.europa.eu/life/publicWebsite/index.cfm?fuseaction=search.dspPage&n\\_proj\\_id=6860](https://webgate.ec.europa.eu/life/publicWebsite/index.cfm?fuseaction=search.dspPage&n_proj_id=6860). [Kreiptasi 17 05 2022].
- [11] Yusman, A. Finawan ir Rusli, „Design of Wild Animal Detection and Rescue System with Passive Infrared and Ultrasonic Sensor based Microcontroller,“ *Emerald Reach Proceedings Series*, t. Vol. 1, pp. 415-422.
- [12] S. Ward, J. Hensler, B. Alsalam ir L. F. Gonzalez, „Autonomous UAVs Wildlife Detection Using Thermal Imaging, Predictive Navigation and Computer Vision,“ įtraukta *2016 IEEE Aerospace Conference*, 2016.

- [13] T. Damarla, A. Mehmood ir J. Sabatier, „Detection of people and animals using non-imaging sensors,“ įtraukta *14th International Conference on Information Fusion*, Chicago, Illinois, 2012.
- [14] L. Daemer, „Electronic specifier,“ Electronic specifier, 27 gegužė 2020. [Tinkle] <https://www.electronicspecifier.com/products/sensors/digitisers-used-for-road-radar-to-detect-wild-animals>. [Kreiptasi 15 gegužė 2022].
- [15] ". So-In, C. Phaudphut, S. Tesana, N. Weeramongkonlert, K. Phaudphut, S. Tesana, N. Weeramongkonlert, K. Wijitsopon, U. KoKaew ir B. Waikh, „Mobile animal tracking systems using light sensor for efficient power and cost saving motion detection,“ įtraukta *2012 8th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, 2012.
- [16] C. Druta ir A. S. Alden, „Evaluation of a Buried Cable Roadside Animal Detection System,“ Virginia Center for Transportation Innovation and Research, Charlottesville, Virginia, 2015.
- [17] Hikvision, „Hikvision - Thermal Products - Security thermal cameras - heatpro series,“ [Tinkle] <https://www.hikvision.com/europe/products/Thermal-Products/Security-thermal-cameras/heatpro-series/ds-2td2617-6-pa/>. [Kreiptasi 10 12 2022].
- [18] M. Elad, „On the Origin of the Bilateral Filter and Ways to Improve It,“ *IEEE Transactions on Image Processing*, pp. 1141 - 1151, 10 2022.
- [19] S. Paris, P. Kornprobst, J. Tumblin ir F. Durand, „Bilateral Filtering: Theory and Applications,“ *Computer Graphics and Vision*, pp. 1 - 73, 2009.
- [20] P. Roy, S. Dutta, N. Dey, G. Dey, S. Chakraborty ir R. Ray, „Adaptive Thresholding: A comparative study,“ įtraukta *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCCICT)*, 2014.
- [21] EDUCBA, „Introduction to openCV findContours,“ EDUCBA, [Tinkle] <https://www.educba.com/opencv-findcontours/>. [Kreiptasi 26 11 2022].
- [22] D. G. Lowe, „Distinctive Image Features from Scale-Invariant Keypoints,“ *International Journal of Computer Vision*, p. 91–110, 2004.
- [23] R. Fisher, S. Perkins, A. Walker ir E. Wolfart, „Gaussian smoothing,“ HIPR2, [Tinkle] <https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>. [Kreiptasi 28 11 2022].
- [24] Automatic Addison, „Pros and Cons of Gaussian Smoothing,“ Automatic Addison, [Tinkle] <https://automaticaddison.com/pros-and-cons-of-gaussian-smoothing/>. [Kreiptasi 30 11 2022].
- [25] T. AI, „What is a Canny Edge Detection Algorithm?,“ Towards AI, 17 09 2020. [Tinkle] <https://towardsai.net/p/computer-vision/what-is-a-canny-edge-detection-algorithm>. [Kreiptasi 30 11 2022].
- [26] R. Fisher, S. Perkins, A. Walker ir E. Wolfart, „Thinning,“ HIPR2, [Tinkle] <https://homepages.inf.ed.ac.uk/rbf/HIPR2/thin.htm>. [Kreiptasi 30 11 2022].
- [27] M. T. Orchard ir C. A. Bouman, „Color Quantization of Images,“ *IEEE Transactions on Signal Processing*, pp. 2677 - 2690, 12 1991.

- [28] Education Ecosystem (LEDU), „Understanding K-means Clustering In Machine Learning,“ Towards Data Science, 13 09 2018. [Tinkle] <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>. [Kreiptasi 26 11 2022].
- [29] Wolfram MathWorld, „Absolute Difference,“ Wolfram MathWorld, 13 12 2022. [Tinkle] <https://mathworld.wolfram.com/AbsoluteDifference.html>. [Kreiptasi 01 12 2022].
- [30] MathWorks, „Morphological dilation and erosion,“ MathWorks, [Tinkle] <https://se.mathworks.com/help/images/morphological-dilation-and-erosion.html>. [Kreiptasi 01 12 2022].
- [31] HIPR2, „Dilation,“ HIPR2, [Tinkle] <https://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>. [Kreiptasi 10 12 2022].
- [32] D. G. Altman, D. Machin, T. N. Bryant ir M. J. Gardner, „Calculating confidence intervals for proportions and their differences,“ įtraukta *Statistics With Confidence*, London, BMJ Publishing Group, 1898, pp. 28-33.
- [33] Xilinx, „Xilinx,“ Advanced Micro Devices, Inc, [Tinkle] <https://www.xilinx.com/support/university/xup-boards/XUPPYNQ-Z2.html>. [Kreiptasi 26 03 2023].
- [34] Design & Reuse, „Design & Reuse,“ Design & Reuse (D&R), [Tinkle] <https://www.design-reuse.com/articles/16875/the-arm-cortex-a9-processors.html>. [Kreiptasi 26 03 2023].
- [35] K. Denolf, J. Lo ir M. Qasaimeh, „PYNQ - Computer Vision,“ 2019. [Tinkle] <https://github.com/Xilinx/PYNQ-ComputerVision>. [Kreiptasi 20 03 2023].
- [36] P. Chakravorty, „What Is a Signal?,“ *IEEE Signal Processing Magazine*, t. 5, nr. 35, pp. 175-177, 2018.
- [37] Simplilearn, „What is image processing? Overview, applications, benefits, and who should learn it,“ Simplilearn, 24 04 2022. [Tinkle] <https://www.simplilearn.com/image-processing-article>. [Kreiptasi 17 05 2022].
- [38] V. R. K. K. a. G. B. E. Rublee, „ORB: An efficient alternative to SIFT or SURF,“ *2011 International Conference on Computer Vision*, pp. 2564-2571, 2011.

## Priedai

### 1 priedas. Metodikos, paremtos dvišaliu filtravimu, algoritmas

```
def tikrinti_ar_koordinates_liecia_nuotraukos_krasta(plotis, aukstis, x, y, pl, auk):
    return x <= 0 or (x + pl) >= (plotis - 1) or y <= 0 or (y + auk) >= (aukstis - 1)

def padidinti_keturkampiai(objekto_keturk, plotis, aukstis):
    x1 = objekto_keturk[0]-2 if objekto_keturk[0] > 2 else objekto_keturk[0]
    x2 = (objekto_keturk[0] + objekto_keturk[2])+2 if (objekto_keturk[0] + objekto_keturk[2])+2 < (plotis)
else objekto_keturk[0] + objekto_keturk[2]
    y1 = objekto_keturk[1]-2 if objekto_keturk[1] > 2 else objekto_keturk[1]
    y2 = (objekto_keturk[1] + objekto_keturk[3])+2 if (objekto_keturk[1] + objekto_keturk[3])+2 < (aukstis)
else objekto_keturk[1] + objekto_keturk[3]
    return (x1, y1), (x2, y2)

def ar_taskas_yra_keturkampio_viduje(keturk_koords,pt):
    return keturk_koords[0] < pt[0] < keturk_koords[0]+keturk_koords[2] and keturk_koords[1] < pt[1] <
keturk_koords[1]+keturk_koords[3]

def rasti_gyvuno_objekta(nuotrauka, gyvunu_objektu_keliai_direktorijoje, koordinates_objekto_keturk):
    nuotrauka2 = nuotrauka
    nuotrauka2 = cv2.cvtColor(nuotrauka2, cv2.COLOR_BGR2GRAY)

    gyvunas_buvo_rastas = False
    visu_rastu_gyvunu_kauke = []
    for obj_kelias in gyvunu_objektu_keliai_direktorijoje:
        gyv_obj = plt.imread(obj_kelias)
        gyv_obj = cv2.cvtColor(gyv_obj, cv2.COLOR_BGR2GRAY)

        # Inicializuojamas SIFT algoritmo detektorius
        sift_detektorius = cv2.SIFT_create()
        # Surandami pagrindiniai taskai ir pozymiai
        pagr_t_1, poz_1 = sift_detektorius.detectAndCompute(gyv_obj, None)
        pagr_t_2, poz_2 = sift_detektorius.detectAndCompute(nuotrauka2, None)
        # Inicializuojamas BF tasku ir pozymiu lygintojas
        if poz_1 is None or poz_2 is None:
            continue
        bf_lygintojas = cv2.BFMatcher()
        # Surandami sutampantys pozymiai
        sutampantys_pozymiai = bf_lygintojas.knnMatch(poz_1, poz_2, k=2)
        if matches and len(matches[0]) == 1:
            continue
        elif not matches:
            continue

        tinkami_pagr_t = [[m] for m, n in sutampantys_pozymiai if m.distance < n.distance]
        tinkamu_pagr_t_koords = [pagr_t_2[sutamp_pozymis[0].trainIdx].pt for sutamp_pozymis in tinkami_pagr_t]

        ar_gyvunas_rastas = False
        for (x,y) in tinkamu_pagr_t_koords:
            cv2.circle(nuotrauka2, (int(x),int(y)), radius=3, color=(0, 0, 255), thickness=-1)
            ar_gyvunas_rastas = [ar_taskas_yra_keturkampio_viduje(keturk_koords,(int(x),int(y))) for
keturk_koords in koordinates_objekto_keturk]
            if True in ar_gyvunas_rastas:
                visu_rastu_gyvunu_kauke.extend(ar_gyvunas_rastas)

        if True in visu_rastu_gyvunu_kauke:
            gyvunas_buvo_rastas = True

    return nuotrauka2, gyvunas_buvo_rastas

def main():
    # Apdorojama nuotrauka
    filtruota_nuotrauka = cv2.bilateralFilter(nuotrauka, 9, 75, 75)
    filtruota_pilka_nuotrauka = cv2.cvtColor(filtruota_nuotrauka, cv2.COLOR_BGR2GRAY)
    _, slenkstine_nuotrauka = cv2.threshold(filtruota_pilka_nuotrauka, 180, 255, cv2.THRESH_BINARY)
    # Randami objektø kontūrai
    konturai, hierarchija = cv2.findContours(slenkstine_nuotrauka, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    # Patikrinama, ar kontūrai liečia krasta
    aukstis, plotis = nuotrauka.shape[:2]
    liecia_krasta = []
    for konturas in konturai:
        x, y, pl, auk = cv2.boundingRect(konturas)
        liecia_krasta.append(tikrinti_ar_koordinates_liecia_nuotraukos_krasta(plotis, aukstis, x, y, pl, auk))
    # Randami objektus apibreziantys keturkampiai
```



```
koordinates_objekto_keturk = []
for indeksas in range(len(konturai)):
    if not liecia_krasta[indeksas]:
        objekto_keturk = cv2.boundingRect(konturai[indeksas])

        pradzia, galas = padidinti_keturkampiai(objekto_keturk, plotis, aukstis)
        koordinates_objekto_keturk.append([pradzia[0],pradzia[1],galas[0]-pradzia[0],galas[1]-pradzia[1]])

# Tikrinama, ar nuotraukoje yra gyvunu objektu
nuotrauka_su_galimais_gyvunu_objjs, ar_gyvunas_rastas = rasti_gyvuno_objekta(nuotrauka,
gyvunu_objektu_keliai_direktorijoje, koordinates_objekto_keturk)
```

## 2 priedas. Metodikos, paremtos Canny kraštų radimo algoritmu, algoritmas

```
def patikrinti_ar_obj_konturai_liecia_nuotraukos_krasta(konturai, plotis, aukstis):
    liecia_krasta = []
    for konturas in konturai:
        x, y, w, h = cv2.boundingRect(konturas)
        liecia_krasta.append(tikrinti_ar_koordinates_liecia_nuotraukos_krasta(plotis, aukstis, x, y, w, h))
    return liecia_krasta

def pagrindinis_algoritmas(slenkstis, src_gray, nuotrauka, gyvunu_objektu_keliai_direktorijoje):
    # Aptinkami krastai pasinaudojant Canny algoritmu
    canny_algoritmo_vaizdas = cv2.Canny(nuotrauka, slenkstis, slenkstis * 2)
    # Randami objektu konturai
    konturai, _ = cv2.findContours(canny_algoritmo_vaizdas, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    # Patikrinama, ar rasti objektu konturai neliecia nuotraukos krasto
    aukstis, plotis = nuotrauka.shape[:2]
    ar_liecia_krasta = patikrinti_ar_obj_konturai_liecia_nuotraukos_krasta(konturai, plotis, aukstis)
    koordinates_objekto_keturk = []

    for indeksas in range(len(konturai)):
        if not ar_liecia_krasta[indeksas]:
            obj_keturk = cv2.boundingRect(konturai[indeksas])
            pradzia, galas = padidinti_keturkampi(obj_keturk, plotis, aukstis)
            koordinates_objekto_keturk.append([pradzia[0],pradzia[1],galas[0]-pradzia[0],galas[1]-pradzia[1]])

    # Tikrinama, ar nuotraukoje yra gyvunu objektu
    nuotrauka_su_galimais_gyvunu_objs, ar_gyvunas_rastas = rasti_gyvuno_objekta(nuotrauka,
    gyvunu_objektu_keliai_direktorijoje, koordinates_objekto_keturk)

    return canny_algoritmo_vaizdas, nuotrauka_su_galimais_gyvunu_objs, ar_gyvunas_rastas

def main()
    filtruota_nuotrauka = cv2.blur(nuotrauka, (3,3))
    slenkstis = 100

    canny_algoritmo_vaizdas, nuotrauka_su_galimais_gyvunu_objs, ar_gyvunas_rastas =
    pagrindinis_algoritmas(slenkstis, filtruota_nuotrauka, nuotrauka, gyvunu_objektu_keliai_direktorijoje)
```

### 3 priedas. Metodikos, paremtos spalvų kvantizavimu, algoritmas

```
def kmeans_spalvu_kvantavimas(nuotrauka, klusteriu_kiekis=8, kiek_kartoti=1):
    auk, pl = nuotrauka.shape[:2]
    bandiniai = np.zeros([auk*pl,3], dtype=np.float32)
    skaitiklis = 0

    for x in range(auk):
        for y in range(pl):
            bandiniai[skaitiklis] = nuotrauka[x][y]
            skaitiklis += 1

    kriterijai = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 200, 0.5)
    _, pavadinimai, centrai = cv2.kmeans(
        bandiniai,
        klusteriu_kiekis,
        None,
        kriterijai,
        kiek_kartoti,
        cv2.KMEANS_RANDOM_CENTERS
    )

    centrai = np.uint8(centrai)
    rezultatas = centrai[pavadinimai.flatten()]
    return rezultatas.reshape((nuotrauka.shape))

def main():
    # Apdorojama nuotrauka
    filtruota_nuotrauka = cv2.blur(nuotrauka, (3,3))

    # Atliekamas kmeans spalvu kvantizavimas
    kmeans_rezultatas = kmeans_spalvu_kvantavimas(filtruota_nuotrauka, klusteriu_kiekis=2)
    kmeans_pilkas_rezultatas = cv2.cvtColor(kmeans_rezultatas, cv2.COLOR_BGR2GRAY)
    slenkstinis_rezultatas = cv2.threshold(kmeans_pilkas_rezultatas, 0, 255, cv2.THRESH_BINARY +
    cv2.THRESH_OTSU)[1]

    # Randami objektu konturai
    konturai, _ = cv2.findContours(slenkstinis_rezultatas, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[-2:]

    aukstis, plotis = nuotrauka.shape[:2]
    ar_liecia_krasta = patikrinti_ar_obj_konturai_liecia_nuotraukos_krasta(konturai, plotis, aukstis)

    per_mazas = []
    uzimamo_ploto_slenkstis = 5
    for indeksas in range(len(konturai)):
        uzimamas_plotas = cv2.contourArea(konturai[indeksas])
        if uzimamas_plotas < uzimamo_ploto_slenkstis:
            per_mazas.append(True)
        else:
            per_mazas.append(False)

    koordinates_objekto_keturk = []
    for indeksas in range(len(konturai)):
        if not ar_liecia_krasta[indeksas] and not per_mazas[indeksas]:
            obj_keturk = cv2.boundingRect(konturai[indeksas])

            pradzia, galas = padidinti_keturkampii(obj_keturk, plotis, aukstis)
            koordinates_objekto_keturk.append([pradzia[0],pradzia[1],galas[0]-pradzia[0],galas[1]-pradzia[1]])

    # Patikrinama, ar rasti objektu konturai neliecia nuotraukos krasto
    nuotrauka_su_galimais_gyvunu_objs, ar_gyvunas_rastas = rasti_gyvuno_objekta(nuotrauka,
    gyvunu_objektu_keliai_direktorijoje, koordinates_objekto_keturk)
```

## 4 priedas. Metodikos, paremtos judesio aptikimu, algoritmas

```
def main():
    kadru_keliai_direktorijoje_testavimui.sort()
    buves_kadras = None

    for _, pav_failas in enumerate(kadru_keliai_direktorijoje_testavimui, start=2): #[:1]:
        pav = plt.imread(pav_failas)
        # Apdorojamas naujas kadras
        paruostas_kadras = cv2.cvtColor(pav, cv2.COLOR_BGR2GRAY)
        paruostas_kadras = cv2.GaussianBlur(src=paruostas_kadras, ksize=(5,5), sigmaX=0)

        # Pirmas kadras issaugomas kaip buves
        if buves_kadras is None:
            buves_kadras = paruostas_kadras
            continue

        # Atliekamas kadru skirtumas
        kadru_skirtumas = cv2.absdiff(src1=buves_kadras, src2=paruostas_kadras)
        # Paruostas kadras tampa buvusiu ateities kadru
        buves_kadras = paruostas_kadras

        # Atliekamas objektu ispletimas
        branduolys = np.ones((5, 5))
        kadru_skirtumas = cv2.dilate(kadru_skirtumas, branduolys, iterations=2)
        slenkstinis_kadras = cv2.threshold(src=kadru_skirtumas, thresh=20, maxval=255,
type=cv2.THRESH_BINARY)[1]
        # Randami objektu konturai
        konturai, _ = cv2.findContours(image=slenkstinis_kadras, mode=cv2.RETR_EXTERNAL,
method=cv2.CHAIN_APPROX_SIMPLE)

        # Patikrinama, ar rasti objektu konturai nelielia nuotraukos krasto
        aukstis, plotis = pav.shape[:2]
        ar_liecia_krasta = patikrinti_ar_obj_konturai_liecia_nuotraukos_krasta(konturai, plotis, aukstis)

        koordinates_objekto_keturk = []
        for indeksas in range(len(konturai)):
            if not ar_liecia_krasta [indeksas]:
                obj_keturk = cv2.boundingRect(konturai[indeksas])

                pradzia, galas = padidinti_keturkampii(obj_keturk , plotis, aukstis)
                koordinates_objekto_keturk .append([pradzia[0],pradzia[1],galas[0]-pradzia[0],galas[1]-
pradzia[1]])

        # Tikrinama, ar nuotraukoje yra gyvunu objektu

        nuotrauka_su_galimais_gyvunu_objs, ar_gyvunas_rastas = rasti_gyvuno_objekta(pav,
gyvunu_objektu_keliai_direktorijoje, koordinates_objekto_keturk)
```