KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

Marius Matulionis

SKAITMENINIŲ FOTOGRAFIJŲ PALYGINIMO METODŲ TYRIMAS

Baigiamasis magistro projektas

Vadovas
Doc. dr. Armantas Ostreika

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

SKAITMENINIŲ FOTOGRAFIJŲ PALYGINIMO METODŲ TYRIMAS

Baigiamasis magistro projektas

Informatika (kodas 621I10003)

Vadovas
Doc. dr. Armantas Ostreika

Recenzentas
Lekt. Julijus Jakutavičius

Projektą atliko
Marius Matulionis

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos

(Fakultetas)

Marius Matulionis

(Studento vardas, pavardė)

Informatika, 621I10003

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Pavadinimas“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 ___ m. _____ ____ d.

Kaunas

Patvirtinu, kad mano, **Mariaus Matulionio**, baigiamasis projektas tema „Skaitmeninių fotografijų palyginimo metodų tyrimas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

_____ _____

(vardą ir pavardę įrašyti ranka) *(parašas)*

TURINYS

Matulionis, Marius. Skaitmeninių fotografijų palyginimo metodų tyrimas. Magistro baigiamasis projektas / vadovas doc. dr. Armantas Ostreika; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: Kompiuterinė rega

Reikšminiai žodžiai: *Kompiuterinė rega, algoritmai, fotografijų palyginimas*

Kaunas, 2017. 84 p.

SANTRAUKA

Šiame darbe tiriami kompiuterinės regos algoritmai gebantys palyginti skaitmenines fotografijas. Vaizdų palyginimo algoritmus galima skirstyti į dvi pagrindines kategorijas: pikselių palyginimu paremti algoritmai ir savybių aptikimu ir palyginimu paremti algoritmai. Pagrindinis dėmesys yra skiriamas savybių aptikimu paremtiems algoritmams, nes lyginant su pikselių palyginimu paremtais algoritmais, jie pasižymi ženkliai didesniu atsparumu fotografijų modifikacijoms ir iškraipymams tokiems kaip raiškos, spalvų pokytis, vaizdo pasukimas ar suliejimas, tačiau dėl didelio jų universalumo, jų veikimas ir rezultatai nėra iki galo ištirti.

Šis darbas susideda iš vaizdo palyginimo algoritmų ir juos naudojančios programinės įrangos analizės bei tyrimo projektavimo ir eksperimentinės dalies. Analizės dalyje apžvelgiama populiari vaizdų palyginimo programinė įranga, išbandomas jos veikimas bei bandoma nustatyti kokie algoritmai naudojami. Eksperimentų planavimo dalyje aprašomi eksperimento planai, pradiniai duomenys, eksperimentų rezultatai, darbo pabaigoje pateikiamos išvados.

Atlikus eksperimentus ištirta algoritmų greitaveika, nustatytas atsparumas pasukimui, vaizdo pasukimui, vaizdo suliejimui, spalvų bei mastelio pokyčiui bei kitoms modifikacijoms. Didžiausia greitaveika pasižymėjo ORB algoritmas, jo veikimo greitis buvo nuo 2 iki 500 kartų greitesnis lyginant su kitais algoritmais, taip pat jis pasižymėjo labai dideliu tikslumu.

Matulionis, Marius. Master‘s thesis in Investigation Of Methods For Digital Photography Comparison / supervisor assoc. prof. Armantas Ostreika. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: computer vision

Key words: computer vision, algorithms, photo comparison

Kaunas, 2017. 84 p.

SUMMARY

In this paper we compare computer vision algorithms that can be used to compare digital photographs. We can separate photo comparison algorithms into two separate categories: pixel based algorithms and feature based algorithms. In this paper we focus on feature based algorithms because they are much more versatile and can be used to compare photos that have different resolotions and variuos distrortions which can not be compared with regular pixel based algorithms. But because feture based algorithms are so versatile, their abilities and weaknesses aren't fully reseached yet.

This paper consist of analysis and experimentation parts. In the analysis part we look at photo comparison software and algorithms that they use to compare digital photographs. In the experimentation part we plan our experiments, write the conditions for the experiments and then test the selected algorithms.

After completing the experiments we got data on how fast the algorithms work, how resistant they are to variuos distortions and photograph modifications. The algorithms with had the best run time and showed the most consistent accuracy was ORB algorithm.

Terminų ir santrumpų žodynas

FAST	Kampų aptikimo algoritmas
SIFT	(angl. *Scale-invariant feature transform*) kompiuterinės regos algoritmas skirtas aptikti objektus vaizduose
SURF	(angl. *Speeded Up Robust Features*) kompiuterinės regos algoritmas
BRISK	(angl. *Binary Robust Invarian Scalable Keypoints*) kompiuterinės regos algoritmas
ORB	(angl. *Oriented BRIEF and Rotated FAST*) kompiuterinės regos algoritmas
FREAK	(angl. *Fast Retina Keypoint*) kompiuterinės regos algoritmas
AGAST	Kompiuterinės regos algoritmas skirtas kampų aptikimui
BRIEF	(angl. *Binary Robus Independant Features*) kompiuterinės regos algoritmas
GFTT	(angl. *Good Featires to Track*) kompiuterinės regos algoritmas
KAZE	kompiuterinės regos algoritmas
AKAZE	kompiuterinės regos algoritmas

1. ĮVADAS

Problemos aktualumas

Šiuolaikinės fotografijų peržiūros programos dažnai turi galimybes ieškoti bei palyginti fotografijas. Tačiau, jų rezultatai, bei efektai dažnai skiriasi, be to nuolat atsiranda naujų, patobulintų algoritmų, dėl to jų galimybės, privalumai bei trūkumai ne visada aiškūs. Dėl šios priežasties atsiranda poreikis ištirti jų efektyvumą.

Darbo tikslas ir uždaviniai

Išanalizuoti fotografijų palyginimo algoritmus bei juos naudojančią programinę, ištirti algoritmų veikimą bei efektyvumą.

1. Atlikti vaizdo palyginimo algoritmų bei juos naudojančios programinės įrangos analizę.
2. Atlikti populiarių vaizdo palyginimo algoritmų greitaveikos tyrimą.
3. Ištirti algoritmų tikslumą bei atsparumą esant įvairiems fotografijų iškraipymams ir modifikacijoms.

Darbo struktūra

Šis darbas susideda iš dviejų pagrindinių dalių: analizės bei vaizdų palyginimo metodų tyrimo projektavimo ir eksperimentinės dalies.

Analizėje atliekama esamų sprendimų bei vaizdų palyginimo ir paieškos algoritmų apžvalga.

Eksperimento dalyje planuojami ir vykdomi eksperimentai bei pateikiami apibendrinti rezultatai ir išvados

2. VAIZDŲ PALYGINIMO ALGORITMŲ IR PROGRAMINĖS ĮRANGOS ANALIZĖ

Šioje darbo dalyje bus apžvelgiama vaizdų palyginimo programinė įranga bei populiariausi algoritmai, skirti vaizdų paieškai ir palyginimui.

2.1. Fotografijų palyginimo programinės įrangos analizė

Bolide Software sukurta nuotraukų palyginimo programinė įranga palaiko raw, jpeg, j2k, bmp, gif, png, tiff, tga ir dar keletą kitų vaizdų formatų. Ji atpažįsta skaitmenines fotografijas ir automatiškai parenka geriausią iš visų dublikatų esančių kompiuteryje, taip suteikiant galimybę perkelti ar ištrinti didžiąją dalį kopijų vos keliais mygtukų paspaudimais.

Vienas iš išskirtinių šios programos bruožų yra tai, kad ji naudoja ir savybėmis pagrįstą vaizdų paiešką, kuri suteikia galimybę efektyviai palyginti pasuktus, apkirptus ar kitaip redaguotus vaizdus. Ši programa yra mokama [1]. Programos ekranvaizdis ir rezultatai pateikiami 1.1 priede.

ionForge sukurta fotografijų palyginimo programa, pavadinimu vienu metu palyginti dvi fotografijas ir parodo skirtumus tarp jų, taip pat siūlo ir keletą rezultatų atvaizdavimo būdų, tačiau kaip, matome iš rezultatų, ši programa naudoja įprastus pikselių palyginimo algoritmus, o ne pažangesnius nuotraukų turiniu paremtus algoritmus. Ši programa yra nemokama. Programos ekranvaizdis ir rezultatai pateikiami 1.2 priede.

Resemble.js - tai atviro kodo internetinė vaizdų analizės ir palyginimo programa, veikianti interneto naršyklėje. Ji naudoja *HTML5 File API* vaizdų duomenų gavimui ir *PhantomCSS* biblioteką vaizdų palyginimui. Ši programa taip pat siūlo keletą vaizdų palyginimo algoritmų bei rezultatų atvaizdavimo būdų. Ši programa yra nemokama [2]. Programos ekranvaizdis ir rezultatai pateikiami 1.3 priede.

Anti-twin - tai programa skirta vienodų failų bei fotografijų paieškai kompiuteryje. Vienos iš pagrindinių savybių, tai baitų paieška, kuri leidžia rasti bet kokios rūšies failų kopijas, identiškų ar panašių failų pavadinimų paieška ir pikseliais paremtas vaizdų palyginimas, tačiau kaip ir daugelis kitų programų, ji nepalaiko turiniu pagrįsto fotografijų palyginimo. Ši programa yra nemokama asmeniniam naudojimui [3]. Programos ekranvaizdis ir rezultatai pateikiami 1.4 priede.

FlounderCraft Image comparator - tai atviro kodo vaizdų palyginimo programa gebanti palyginti kelių skirtingų formatų vaizdus, tačiau naudoja tik pačius paprasčiausius pikseliais paremtus algoritmus. Pasak autoriaus, ši programa dar nėra visiškai užbaigta [4]. Programos ekranvaizdis ir rezultatai pateikiami 1.5 priede.

Image Comparer - tai dar viena labai paprasta vaizdų palyginimo programa naudojanti pikseliais paremtus vaizdų palyginimo algoritmus, be to ši programa nesiūlo jokio papildomo funkcionalumo. Ši programa yra nemokama. Programos ekranvaizdis ir rezultatai pateikiami 1.6 priede.

DiffImg - tai paprasta vaizdų palyginimo programa leidžianti vartotojui palyginti du vienodos raiškos paveikslėlius, turinti galimybę keisti keletą paveikslėlio palyginimo parametrų bei atvaizduojanti paveikslėlių informaciją, taip pat ir jų histogramas. Kaip ir didžioji dalis programų ji naudoja pikseliais paremtus algoritmus [5]. Programos ekranvaizdis ir rezultatai pateikiami 1.7 priede.

1 lentelė. Programinės įrangos palyginimas

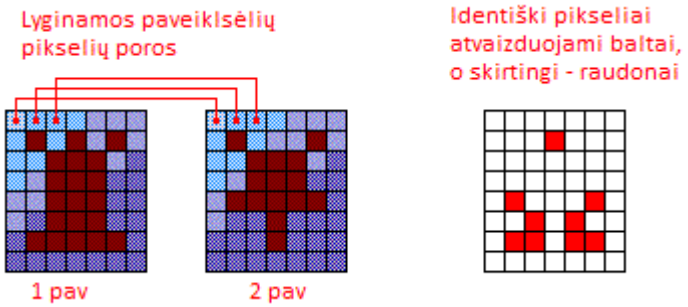
Programos pavadinimas	Kaina	Naudoja baitais paremtus vaizdų paieškos algoritmus	Naudoja pikseliais paremtus vaizdų paieškos algoritmus	Naudoja turiniu paremtus vaizdų paieškos algoritmus	Atvirojo kodo
Bolidesoft *Image comparer*	**$34.95**	Ne	Taip	Taip	Ne
ImageDiff	Nemokama	Ne	Taip	Ne	Ne
Resemble.js	Nemokama	Ne	Taip	Ne	Taip
Anti-Twin	Nemokama	Taip	Taip	Ne	Ne
FlounderCraft Image comparator	Nemokama	Ne	Taip	Ne	Taip
Image Comparer	Nemokama	Ne	Taip	Ne	Ne
DiffImg	Nemokama	Ne	Taip	Ne	Ne

Kaip matome 1 lentelėje, didžioji dalis vaizdų palyginimo programų naudoja pikseliais paremtus algoritmus, taip yra dėl lengvo jų realizavimo. Iš septynių apžvelgtų programų, tik viena naudojo pažangesnius, savybėmis paremtus algoritmus, tačiau ši programa buvo mokama.

2.2. Vaizdų palyginimo ir paieškos algoritmų apžvalga

2.2.1 Klasikinis tikrinimas

Klasikinio tikrinimo metu yra lyginamas kiekvienas abiejų paveiksliukų pikselis. T. y. pirmo paveikslio pirmas pikselis yra lyginamas su antro paveikslio pirmu pikseliu, antras - su antru ir t.t. Tokiu būdu tikrinami paveiksliukai turi būti tokios pat raiškos ir suspaudimo, kitu atveju bus nustatyta, kad paveikslėliai yra skirtingi. Toks algoritmas labiausiai tinkamas paveikslėlių kopijų paieškai, nes kiekvienas pikselių spalvos skirtumas yra aptinkamas kaip paveikslėlių neatitikimas



1 pav. Pikseliais pagrįsto tikrinimo veikimas

2.2.2 Pikseliais paremtas tapatumo tikrinimas

Šis algoritmas yra labai panašus į klasikinį paveikslėlių tikrinimą, tačiau leidžia tam tikrą kiekį neatitinkančių pikselių. Algoritmas išskaido kiekvieną pikselį į tris spalvas: raudoną, žalią ir mėlyną. Tada tikrinama kiekvienos spalvos esama vertė su verte, kurios tikimasi. Susumuojami pikseliai, kurių vertės yra identiškos. Galutinis rezultatas yra identiškų pikselių skaičius, padalintas iš visų pikselių skaičiaus. Šis algoritmas tinkamas kai paveikslėliuose gali būti nedidelis kiekis skirtingų pikselių, tačiau kaip ir klasikinis tikrinimas, šis algoritmas nėra tinkamas, jei paveikslėliuose yra poslinkių ar iškraipymų.

2.2.3 Pikseliais paremtas panašumo tikrinimas

Šis algoritmas suskaido kiekvieną pikselį į tris spalvas: raudoną, žalią ir mėlyną, tada tikrina pirmo paveikslio kiekvieno pikselio spalvas su antro paveikslio spalvomis ir suskaičiuoja procentinį panašumą. Šio algoritmo galutinis rezultatas yra visų pikselių spalvų nukrypimas nuo pradinio paveikslėlio.

2.2.4 Blokais pagrįstas tapatumo tikrinimas

Blokais pagrįsto tapatumo algoritmas padalina paveikslą į kvadratinius pasirinkto dydžio blokus. Kiekvieno bloko reikšmė - vidutinis visų bloke esančių spalvų reikšmių vidurkis. Jei paveikslėlio plotis ar aukštis nėra bloko dydžio kartotinis, tai paveikslėlio dešinysis kraštas ar apačia atitinkamai apkarpoma. Tada palyginamas blokų panašumas, o galutinis rezultatas gaunamas padalinus identiškų blokų skaičių iš visų blokų skaičiaus

2.2.5 Blokais pagrįstas panašumo tikrinimas

Šis algoritmas veikia labai panašiai kaip ir blokais pagrįstas tapatumo tikrinimas, tačiau čia apskaičiuojamas iš procentinės kiekvieno bloko spalvos reikšmės. [6]

2.2.6 Histogramos tikrinimas

Spalvų histograma atvaizduoja paveikslėlio spalvų pasiskirstymą naudojantis tam tikru kiekiu kibirėlių, kur kiekvienas histogramos kibirėlis nurodo spalvą kvantuotoje spalvų erdvėje.

Siekiant palyginti paveikslėlius naudojantis histogramomis, reikia palyginti vieno paveikslėlio histogramos spalvų kibirėlius su kito paveikslėlio spalvų histogramos atitinkamais kibirėliais. Galutinis rezultatas gaunamas palyginus atitinkamus spalvų kategorijų dažnius.

2.2.7 Palyginimas naudojantis *Bitiesinis* (*Bilinear*) filtru.

Šis algoritmas sumažina paveikslėlius iki pasirinkto dydžio, tada vėl padidina iki originalaus dydžio, naudodamasis *Bitiesiniu* filtru. Tokio proceso metu gaunamas paveikslėlio suliejimas, nes šis filtras skaičiuoja esamo pikselio reikšmę pagal šalia jo esančio kaimyninio pikselio reikšmę. Dėl to prarandama dalis dažniausiai nereikalingos paveikslėlio informacijos, tokios kaip triukšmas. Dėl šios priežasties, šis algoritmas gali būti taikomas įvairiose srityse. Galutinis algoritmo rezultatas – sulietų paveikslėlių pikselių spalvų vidutinis nuokrypis.

2.2.8 SURF algoritmas

SURF(angl. *Speeded Up Robust Features*) algoritmas - tai vienas iš populiariausių turiniu ir savybėmis paremtų vaizdo atpažinimo algoritmų. Šis algoritmas buvo sukurtas siekiant užtikrinti didelį vaizdų savybių aptikimo greitį trijuose etapuose: aptikime, apibūdinime ir palyginime. Pasak autorių, SURF gali veikti greičiau nei SIFT, neprarasdamas raktinių taškų kokybės. SURF algoritmas susideda iš keturių pagrindinių etapų. Kelių mastelių analizėje, priešingai nei SIFT algoritme, naudojamas ne Gauso, o dėžės filtravimo metodai, kurie veikia panašiu principu, tačiau dėžės filtravimas yra greitesnis. Raktinių taškų aptikimo žingsnyje *Hessian* operatorius yra naudojamas siekiant pasirinkti raktinių taškų kandidatus, tada patikrinama ar pasirinktų taškų reikšmės yra virš pasirinkto slenksčio. Tada šių taškų vieta ir mastelis yra išgryninami naudojant kvadratinį išdėstymą. Savybių apibūdinimo etape atliekant analizę, taškų pasirinkimas dėžės erdvėje suteikia mastelio ir vertimo nekintamumą, siekiant sukurti pasukimo nekintamumą, orientacija yra nustatoma pagal vietinio gradiento orientacijos pasiskirstymą. 64 dimensijų deskriptorius sukuriamas naudojantis erdvinės lokalizacijos tinkleliu. Savybių lyginimo etape, palyginami anksčiau sukurti kelių paveikslėlių deskriptoriai. Tai atliekama skaičiuojant Euklidinį atstumą tarp visų potencialiai vienodų raktinių taškų porų. [7]

Hessian detektorius [8]:

$$\mathcal{H}(\vec{x},\sigma) = \begin{bmatrix} L_{xx}(\vec{x},\sigma) & L_{xy}(\vec{x},\sigma) \\ L_{xy}(\vec{x},\sigma) & L_{yy}(\vec{x},\sigma) \end{bmatrix} \qquad (1)$$

SURF privalumai:

- Ženkliai greitesnis už SIFT;
- Labiau pritaikytas lygiagretiems skaičiavimams;
- Atsparus triukšmui;

Trūkumai:

- Mažesnis tikslumas, lyginant su SIFT;
- Ne toks atsparus pasukimams;
- Apsaugotas patentu;

2.2.9 SIFT algoritmas

Šis algoritmas susideda iš keturių pagrindinių etapų:

a) Mastelio-erdvės ekstremumų aptikimo

Naudojantis Gauso branduoliu randame mastelio erdvę:

$$L(x, y, \sigma_D) = G(x, y, \sigma_D) * I(x, y) \quad (2)$$

Siekdami sumažinti sąnaudas aproksimuojame Laplaso-Gauso operatorių

$$\sigma\nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(k\sigma) - G(\sigma)}{k\sigma - \sigma} \quad (3)$$

Apibūdiname Gauso skirtumą

$$(k-1)\sigma^2\nabla^2 G \approx G(k\sigma) - G(\sigma) \quad (4)$$

$$D(\sigma) \equiv (G(k\sigma) - G(\sigma)) * I \quad (5)$$

b) Raktinių taškų aptikimo

$$D(\vec{x}) = D + \frac{\partial D^T}{\partial \vec{x}}\vec{x} + \frac{1}{2}\vec{x}^T\frac{\partial^2 D^T}{\partial \vec{x}^2}\vec{x} \quad (6)$$

Siekiant gauti tikruosius vietos ekstremumus, minimizuojame Teiloro eilutę

$$\hat{x} = -\frac{\partial^2 D}{\partial x^2}^{-1}\frac{\partial D}{\partial x} \quad (7)$$

Atmetame raktinius taškus turinčius per mažą kontrastą

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \quad (8)$$

c) Orientacijos priskyrimo

Naudojantis taško masteliu pasirenkame tinkamą paveikslėlį

$$L(x, y) = G(x, y, \sigma) * I(x, y) \quad (9)$$

Suskaičiuojame gradientų svarbą ir orientaciją naudojantis baigtiniais skirtumais

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (10)$$

$$\theta(x, y) = \tan^{-1}\left(\frac{(L(x, y+1) - L(x, y-1))}{(L(x+1, y) - L(x-1, y))}\right) \quad (11)$$

d) Raktinių taškų aprašo sukūrimo

Visi šie etapai turi eiti eilės tvarka ir po kiekvieno etapo turi būti atliekamas filtravimas siekiant palikti tik pakankamai didelio kontrasto raktinius taškus. [7]

```
parse options
load image
resample image to double size
for each octave
    create Gaussian blur intervals
    create difference-of-Gaussian intervals
    compute edges for each interval
end for
search each octave for stable extrema
create keypoints at dominant orientations of extrema
for each keypoint
    rotate sample grid to keypoint orientation
    sample region and create descriptor
end for
optionally save pyramid images
save output images
save descriptors
```

2 pav. SIFT algoritmo pseudo-kodas [9]

Privalumai:

- Didelis tikslumas
- Yra įtrauktas į OpenCV biblioteką
- Atsparus objektų pridengimui
- Reliatyviai efektyvus lyginant su kitais algoritmais

Trūkumai:

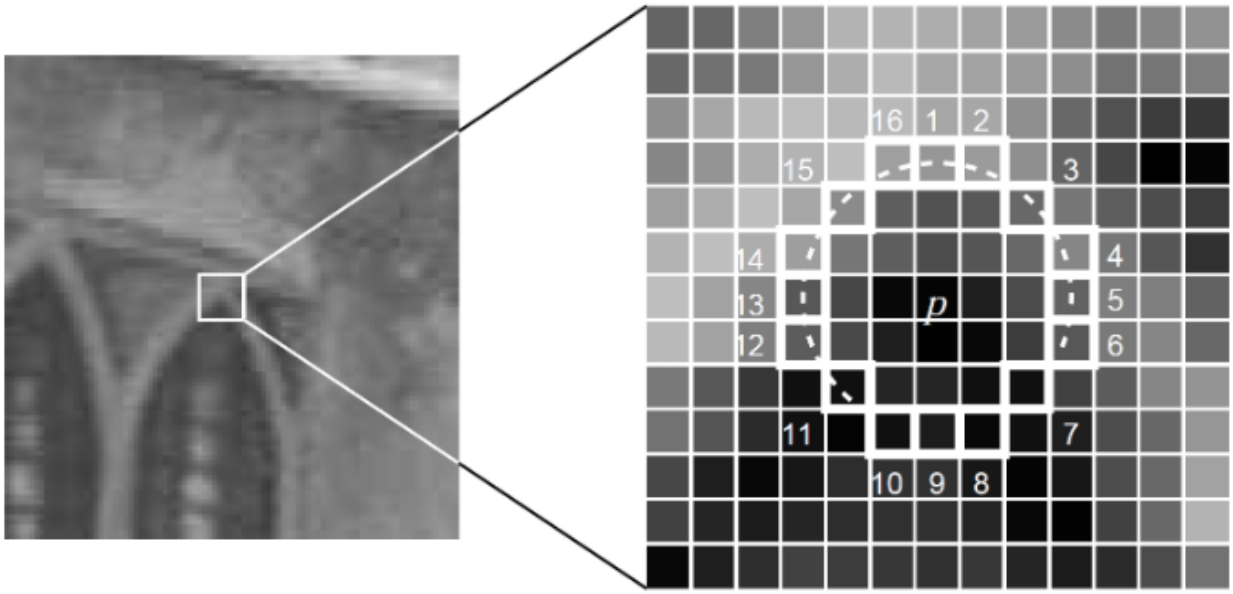
- Ganėtinai lėtas
- Neatsparus apšvietimo pokyčiams ir vaizdų suliejimui
- Apsaugotas patentu

SIFT algoritmo variacijos:

- PCA-SIFT
- SURF
- GLOH
- HOG

2.2.10 FAST algoritmas

Šis algoritmas skiriasi nuo anksčiau minėtų tuo, kad jo veikimas pagrįstas ne paveikslėlio duomenis, o jame atvaizduotų objektų savybėmis. Šiuo atvejo – kampais. Remiantis autoriais, kampų aptikimas yra daug svarbesnis nei kraštinių, nes kampai yra daug intuityvesnis požymis, nes jie parodo stiprų dvimatį intensyvumo pokytį, taip juos išskiriant nuo artimiausių taškų.



3 pav. FAST algoritmo veikimas [10]

FAST algoritmas veikia paveikslėlyje pasirinkdamas pikselį „p“. Numatoma, kad šio pikselio intensyvumas yra $I_{p,}$ tai pikselis kuris gali būti nustatytas kaip dominantis taškas. Tada nustatoma intensyvumo slenksčio vertė T procentais. Apsvarstoma šešiolikos tašką „p“ ratu supančių taškų vertė. Kad pikselis būtų aptiktas kaip dominantis taškas, tai N iš gretimų Šešiolikos pikselių turi būti T verte daugiau arba mažiau už I_p . Siekiant kiek įmanoma pagreitinti algoritmo veikimą, pirmiausiai patikrinami pirmas, penktas, devintas ir tryliktas pikseliai esantys rate ir jų reikšmė palyginama su I_p. siekiant kad pasirinktas pikselis būtų laikomas kampu, bent trys iš keturių pasirinktų taškų turi tenkinti slenksčio sąlygą. Jei mažiausiai trys iš keturių pasirinktų taškų netenkina slenksčio sąlygos, tada pikselis p nėra laikomas kampu. Jei taip atsitinka, tada tikrinama ar 12 iš 16 pikselių atitinka sąlygą, jei neatitinka, tada pasirenkamas kitas pikselis ir šie veiksmai kartojami tol, kol patikrinami visi paveikslėlyje esantys taškai. [11] [10]

Privalumai:

- Labai greitas
- Puikiai aptinka paveikslėlių savybes

Trūkumai

- Neatsparus dideliems triukšmo lygiams
- Priklausomas nuo slenkstinio žingsnio

FAST algoritmo variacijos:

- FAST-ER
- AGAST
- AST
- SUSAN

2.2.11 BRIEF algoritmas

BRIEF tai pirmasis išleistas dvejetainis deskriptorius. Šis deskriptorius pasižymi tuo, kad geba rasti dvejetainius tekstus(angl. *Strings)* tiesiogiai, neskaičiuojant pačių deskriptorių. Pirmiausia jis pasiima sulieto paveikslėlio gabalėlį bei pasirenka rinkinį n koordinačių porų tam tikru unikaliu būdu. Tada palygina kai kurių pikselių porų intensyvumą, pavyzdžiui, jei pirma pora yra p ir q, ir p intensyvumas yra mažesnis už q intensyvumą tada rezultatas yra 1, kitu atveju – 0. Tokie veiksmai pritaikomi visoms n koordinačių porų, taip gauname n dimensijų turintį bitų masyvą. n reikšmės dažniausiai būna 128, 256 ar 512. BRIEF yra tik detektorius, todėl siekiant palyginti vaizdus būtinai reikės naudotis ir savybių detektoriumi, pvz. SIFT ar SURF. BRIEF pasižymi savo greičiu bei puikiu atpažinimu, jei vaizdų pasisukimas plokštumoje nėra labai didelis. [12] [13]

2.2.12 ORB algoritmas

ORB tai labai greitas dvejetainis deskriptorius paremtas BRIEF, kuris yra nepriklausomas nuo pasukimo ir atsparus triukšmui. ORB pasižymi tuo, kad jis yra žymiai greitesnis nei SIFT bei daugeliu atvejų pasižymi tokiu pat tikslumu. Šio algoritmo autoriai išbandė jo efektyvumą įvairiose realaus gyvenimo situacijose, įskaitant objektų atpažinimą bei kelio sekimą išmaniajame telefone. ORB susideda iš visiems gerai žinomo FAST detektoriaus ir BRIEF deskriptoriaus, dėl šios priežasties jis yra vadinamas ORB(angl. *Oriented FAST and Rotated BRIEF*). Šie du metodai pasižymi dideliu našumu bei mažais skaičiavimo kaštais. [14]

2.2.13 KAZE algoritmas

KAZE algoritmas raktinius taškus randa panašiai kaip ir SIFT algoritmas, jis nustato mastelio erdvę naudodamasis logaritminiais žingsniais susidedančiais iš oktavų ir žemesnio lygio rinkinių. KAZE algoritmas visada dirba su originalia paveikslėlio raiška nekurdamas mažesnės raiškos kopijų kiekvienai oktavai kaip daro SIFT algoritmas.

2.2.14 AKAZE algoritmas

AKAZE algoritmas susideda iš trijų pagrindinių etapų: nelinijinės piramidės sukūrimo, raktinių taškų suradimo ir deskriptoriaus generavimo. Pirmajame etape, AKAZE naudoja Perona-Malik lygtį siekiant sukurti nelinijinio mastelio piramidę. Siekiant sukurti skirtingus žemesnius piramidės lygius, šis metodas išskirsto paveikslėlį į didėjančių mastelių rinkinius naudodamas (12) formulę. Siekiant paspartinti išskirstymo procesą, AKAZE pritaiko greitą išreikštinės difuzijos schemą, kuri aproksimuoja sprendimus pagal iteracijas. Kiekviena iteracija gali išskirti paveikslėlį su mažo mastelio žingsniu. Kadangi greita išreikštinė difuzija remiasi besikeičiančiu mastelio žingsniu, ji gali ženkliai sumažinti iteracijų skaičių.

$$\frac{\partial I}{\partial t} = div\left(\frac{\nabla I}{1+|\nabla I|^2/k^2}\right) \qquad (12)$$

Antrajame etape etape surandami raktinai taškai. Kai piramidė sukonstruota, apskaičiuojame *Hessian* matricos normalizuotą determinantą. Tada, kiekviename žemesniame lygyje esantys lokalūs maksimumai parenkami kaip raktinių taškų kandidatai. Siekiant surasti mastelio ekstremumus, potencialius taškas yra palyginamas su kitais kandidatuojančiais taškais esančiais tam tikro langeliuose esančiuose jį supančiuose sluoksniuose. Trečiajame etape AKAZE pristato

modifikuotą lokalių skirtumų dvejetainį deskriptorių. Šio deskriptoriaus generavimas susideda iš trijų žingsnių: pagrindinės orientacijos nustatymo, pavyzdinės iškarpos pasukimo ir dvejetainio deskriptoriaus generavimo. Pirmiausia naudojant į SURF panašų, histogramomis paremtą metodą nustatoma pagrindinė orientacija, tada pavyzdinis šablonas pasukamas pagal raktinio taško kryptį ir galiausiai lyginant tinklelius keliuose spalvų kanaluose sugeneruojamas dvejetainis deskriptorius [15].

2.2.15 MSER algoritmas

MSER tai vienas iš dėmių paveikslėliuose aptikimo algoritmų, jis susideda iš penkių pagrindinių etapų:

- Paprasto paveikslėlio apšvietimo slenkstinių žingsnių suradimo patikrinant visus intensyvumo lygius nuo juodos iki baltos spalvos.
- Susijungusių komponenčių išskyrimo
- Slenksčio suradimo, kai ekstremalus regionas yra stabiliausias. T.y. Kvadratėlio reliatyvus augimas yra lokalus minimumas. Tačiau dėl diskrečių paveikslėlio savybių, aukščiau ir žemiau esantys regionai gali būti supainioti su tikruoju regionu, tokiu atveju regionas vistiek laikomas maksimaliu.
- Regiono aproksimavimo su elipse
- Regiono išsaugojimo kaip savybių [16]

2.2.16 GFTT algoritmas

GFTT(*angl. Good Features To Track*) tai modifikuotas *Harris* kampų detektorius. Modifikavimą 1994 metais pasiūlė *J. Shi* ir *C. Tomasi*. Pagrindinis modifikavimas – kampų įvertinimo formulės pakeitimas iš (13) į (14)

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 \qquad (13)$$

$$R = \min(\lambda_1, \lambda_2) \qquad (14)$$

Jei λ_1 ir λ_2 yra aukščiau slenkstinės ribos λ_{min} tada regionas laikomas kampu. GFTT suranda paveikslėlyje esančius kampus kurie pasižymi ryškiausiomis savybėmis.

2.3. Algoritmų ir programinės įrangos apžvalgos apibendrinimas

Atlikus algoritmų analizę pastebėta, kad beveik visi vaizdų palyginimo ir apieškos algoritmai yra skirstomi į dvi pagrindines grupes: pikseliais paremtus ir savybėmis paremtus algoritmus. Praktikoje dažniausiai taikomi pikseliais paremti algoritmai, nes jų realizavimas yra žymiai paprastesnis, tačiau priešingai nei savybėmis paremti algoritmai, pikseliais paremti algoritmai dažniausiai yra neatsparūs vaizdų iškraipymams ir modifikacijoms.

Atlikus literatūros apžvalgą pastebėta, kad didžioji dalis tirtų algoritmų turi po keletą variacijų, taip pat pastebėta, kad skirtinguose šaltiniuose, dėl skirtingų testavimo duomenų gaunami skirtingi algoritmų efektyvumo rezultatai. Be to kiekvienas iš analizuotų algoritmų turi tam tikrų trūkumų, dėl to visada bus galimybė patobulinti jų veikimą.

2 lentelė. SIFT ir SURF palyginimas esant skirtingoms modifikacijoms. [17]

Metodas	Veikimo laikas	Mąstelio pokytis	Pasukimas	Suliejimas	Apšvietimo pokytis
SIFT	Vidutinis	Puikus	Puikus	Puikus	Vidutinis
SURF	Puikus	Geras	Vidutinis	Geras	Puikus

3. TYRIMO ĮRANKIO PROJEKTAVIMAS

3.1. Reikalavimai sprendimui

3.1.1 Funkciniai reikalavimai

- Lyginamos fotografijos turi būti atvaizduojamos ekrane
- Palyginimo rezultatai turi būti atvaizduojami grafiškai.

3.1.2 Nefunkciniai reikalavimai

- Tyrimas atliekamas naudojant OpenCV programinę įrangą.

3.2. **Sprendimo realizavimui naudojamos technologijos**:

- Windows 10 operacinė sistema
- OpenCV 3.1 kompiuterinės regos biblioteka
- Visual Studio 2015 programinė įranga
- C++ programavimo kalba

Šiam tyrimo įrankiui realizuoti buvo nuspręsta pasirinkti OpenCV biblioteką nes ji yra nemokama, atviro kodo, o tai leidžia, atsiradus poreikiui modifikuoti pačios bibliotekos kodą, ko dažniausiai negalima padaryti naudojant programinę įrangą kaip MATLAB. Taip pat OpenCV yra sukurta naudojantis C/C++ programavimo kalba, tai yra svarbu nes šiame tyrime bus tiriama ir algoritmų greitaveika, dėl šios priežasties C++ veikimo greitis yra didelis pranašumas lyginant su kitomis programavimo kalbomis.

Tyrimo įrankio veikimas:

Pirmiausia atidaromi norimi palyginti paveikslėliai kurie iš karto paverčiami vienspalviais ir išsaugomi kaip matrica, tai būtina padaryti nes beveik visi savybių aptikimo ir palyginimo algoritmai geba apdoroti tik vienspalvius vaizdus, be to spalvotų vaizdų apdorojimas užtrunka ilgiau. Sekančiame žingsnyje sukuriamas norimo algoritmo detektorius ir deskriptorius. Detektorius pagal paveikslėlių duomenis aptinka raktinius taškus bei išsaugo juos kaip vektorius, toliau deskriptorius pagal paveikslėlių duomenis ir anksčiau atrastus raktinius taškus sukuria savybių aprašus, savybių aprašai saugomi kaip matricos. Siekiant ištirti greitaveiką po kiekvieno žingsnio fiksuojamas ir į ekraną išvedama tam tikrų skaičiavimų trukmė. Visi gauti aprašai "*Brute Force*" palyginami tarpusavyje, taip pat apskaičiuojamas minimalus ir maksimalus Euklidiniai atstumai tarp gautų aprašų. Siekiant išfiltruoti klaidingai aptinkamus raktinius taškus ir palengvinti gautų rezultatų apdorojimą į ekraną atvaizduojamos tik tos savybės kurių reikšmė yra mažesnė arba lygi dvigubam minimaliam ankstesniame žingsnyje apskaičiuotam Euklidiniam atstumui.

4. VAIZDŲ PALYGINIMO METODŲ TYRIMO PROJEKTAVIMAS IR EKSPERIMENTINĖ DALIS

Šiame skyriuje aprašomi eksperimento planai, bei atliktų eksperimentų rezultatai.

4.1. Eksperimentų planavimas

Eksperimentų metu planuojama ištirti realizuotų fotografijų palyginimo metodų veikimą ir tikslumą lyginant dvi fotografijas vienu metu, taip bus užtikrinamas gautų duomenų tikslumas, taip pat kiekvienas palyginimas bus atliekamas mažiausiai 5 kartus, taip bus siekiama išvengi anomalijų galinčių atsirasti dėl kompiuteryje esančios operacinės sistemos ar programinės įrangos kuri galėtų pakenkti tiksliems tyrimo rezultatams, ypač atliekant greitaveikos tikrinimą. Pirmiausia bus bandoma palyginti dvi visiškai skirtingas fotografijas, taip bus nustatomas algoritmų tikslumas. Toliau bus bandoma lyginti fotografijas turinčias tam tikrų sintetinių ir natūralių iškraipymų, galimų modifikacijų sąrašas pateikiamas žemiau. Visos fotografijos naudojamos eksperimentuose bus 2 megapikselių ir didesnės raiškos, tuo bus siekiama tiksliau pamatyti algoritmų veikimo greitį, taip pat pamatyti algoritmų elgesį ir rezultatus esant dideliems apdorojamų duomenų kiekiams.

Galimos fotografijų modifikacijos ir iškraipymai:

- Vaizdo pasukimas
- Nuotraukos raiškos pokytis
- Vaizdo suliejimas
- Apšvietimo pokyčiai
- Spalvų pokyčiai

4.1.1 **Eksperimentams naudojama techninė įranga**.

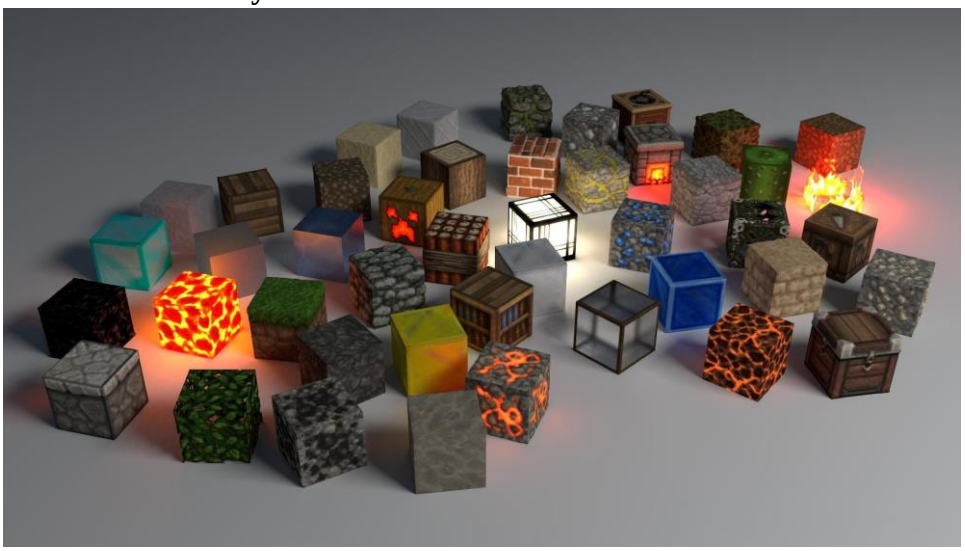
Eksperimentai bus atliekami naudojantis Lenovo Y50-70 nešiojamu kompiuteriu kurio techniniai parametrai yra:

Procesorius	Intel® Core™ i5-4210H, 3M podėlis, taktinis dažnis iki 3.5GHz
Operatyvioji atmintis	8GB
Vaizdo plokštė	Intel HD Graphics 4600; NVidia GTX860M 4GB

4.1.2 Algoritmų tikslumo ir greitaveikos tyrimas.

Šioje dalyje bus tiriamas algoritmų veikimo greitis ir tikslumas lyginant du paveikslėlius. Algoritmų aptiktos bei sutampančios paveikslėlių savybės bus atvaizduojamos paveikslėliuose, algoritmų greitaveikos rezultatai bus pateikiami grafikuose (8-72 priedai).

4.1.3 Pradiniai duomenys



4 pav. Pirmasis paveikslėlis palyginimui [18]

5 pav. Antrasis paveikslėlis palyginimui [18]

Abu paveikslėliai yra *fullHD* raiškos(1080x1920). Tokie paveikslėliai eksperimentui buvo pasirinkti dėl keleto priežasčių: Kadangi daugiausiai bus skiriama algoritmams paremtiems savybių išskyrimu, didelės raiškos paveikslėliai turi daug detalių bei savybių, taip pat yra labai skirtingi, dėl to lyginant juos pasirinktais algoritmais bus galima pamatyti kaip teisingai veikia algoritmai. Taip pat didesni paveikslėliai bus apdorojami ilgiau, dėl to bus aiškiai matoma algoritmų atliekamų skaičiavimų trukmė.

6 pav. Trečiasis paveikslėlis palyginimui

7 pav. Ketvirtasis paveikslėlis palyginimui

Trečiasis ir ketvirtasis paveikslėliai palyginimui atvaizduoja tokį pat objektą, tačiau fotografijos darytos skirtingais atstumais, dėl to atsirado nedidelis spalvų ir mastelio pokytis. Abi fotografijos yra 3 megapikselių raiškos (2048x1536). Lyginant šias fotografijas bus bandoma ištirti algoritmų atsparumą spalvų ir mastelio pokyčiams.

8 pav. Penktasis paveikslėlis palyginimui

9 pav. Šeštasis paveikslėlis palyginimui

Penktasis ir šeštasis yra identiški tačiau šeštasis paveikslėlis yra pasuktas 45 laipsnių kampu. Lyginant šiuos paveikslėlius bus bandoma nustatyti vaizdo algoritmų atsparumą pasukimui.

10 pav. Septintasis paveikslėlis palyginimui

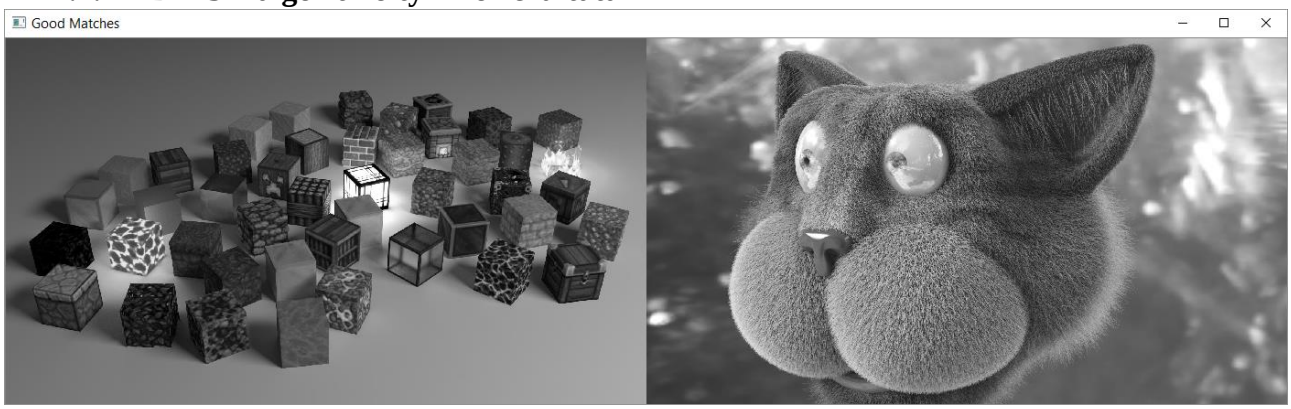
Septintasis paveikslėlis, tai penktojo paveikslėlio kopija kuriai pritaikytas Gauso suliejimas. Lyginant penktąjį ir šeštąjį paveikslėlius bus siekiama nustatyti algoritmų atsparumą suliejimui ir triukšmui.

11 pav. Aštuntasis paveikslėlis palyginimui

Tyrime bus lyginamos 3 ir 6 megapiksleių raiškos aštuntojo paveikslėlio kopijos, taip bus siekiama nustatyti algoritmų atsparumą raiškos pokyčiams.

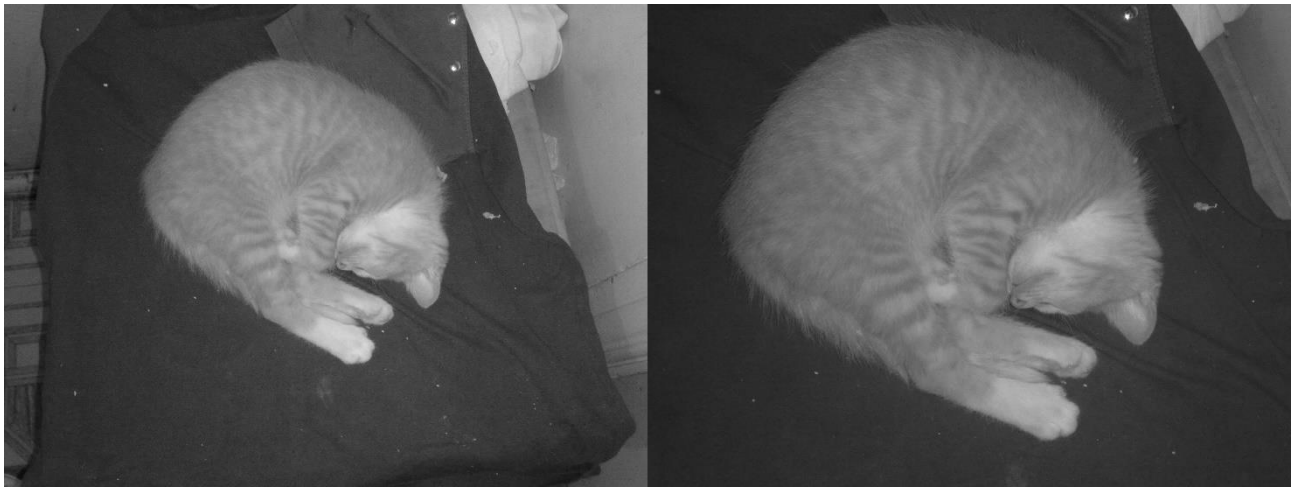
4.2. Tyrimo rezultatai

4.2.1 BRISK algoritmo tyrimo rezultatai



12 pav. BRISK algoritmo rezultatai lyginant skirtingus paveikslėlius

Kaip matome BRISK algoritmas paveikslėliuose nerado jokių sutampančių savybių, Raktinius taškus surado per šiek tiek mažiau nei 7 sekundes, deskriptorių skaičiavimas truko maždaug sekundę, o visi skaičiavimai vidutiniškai truko apie 18,5 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (2 priedas, 2.1 pav.).



13 pav. BRISK algoritmo rezultatai lyginant vizualiai panašias fotografijas

Atliekant vizualiai panašių fotografijų palyginimą BRISK detektorius nerado jokių sutampančių savybių. Visi skaičiavimai vidutiniškai užtruko 0,38 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (2 priedas, 2.2 pav.).

14 pav. BRISK algoritmo rezultatai lyginant pasuktas fotografijas

Lyginant pasuktas fotografijas su BRISK algoritmu, šis algoritmas rado 2 sutampančias savybės iš kurių abi buvo aptiktos teisingai. Raktinių taškų radimas truko apie 5 sekundes, o visų skaičiavimų laikas vidutiniškai buvo 120.8 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (2 priedas, 2.3 pav.).

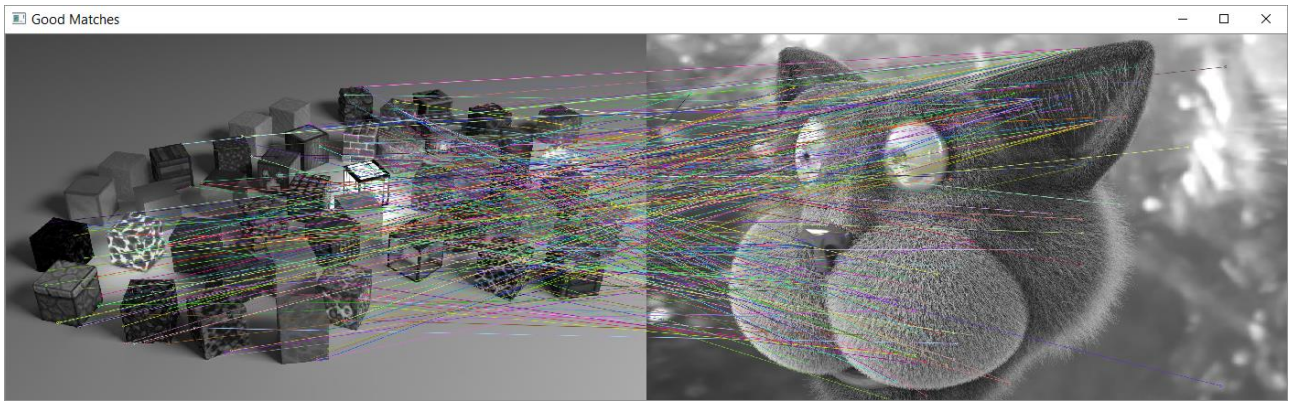
15 pav. BRISK algoritmo rezultatai lyginant fotografijas su sulietu vaizdu

Atliekant fotografijų su sulietu vaizdu palyginimą BRISK algoritmas rado 2 sutampančias savybes iš kurių abi buvo teisingos. Skaičiavimai užtruko 14 sekundžių. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (2 priedas, 2.4 pav).

16 pav. BRISK algoritmo rezultatai lyginant skirtingos raiškos fotografijas

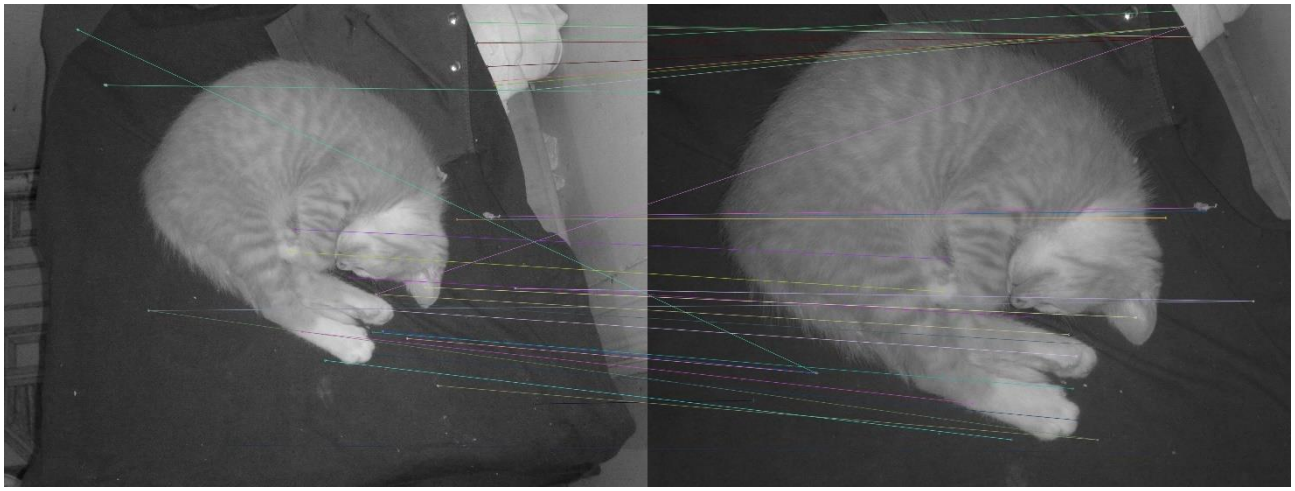
BRISK algoritmas skirtingos raiškos fotografijose aptiko tik vieną teisingai sutampančią savybę ir ji buvo teisinga tačiau skaičiavimai truko net 196.2 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (2 priedas, 2.5 pav.).

4.2.2 SIFT algoritmo tyrimo rezultatai



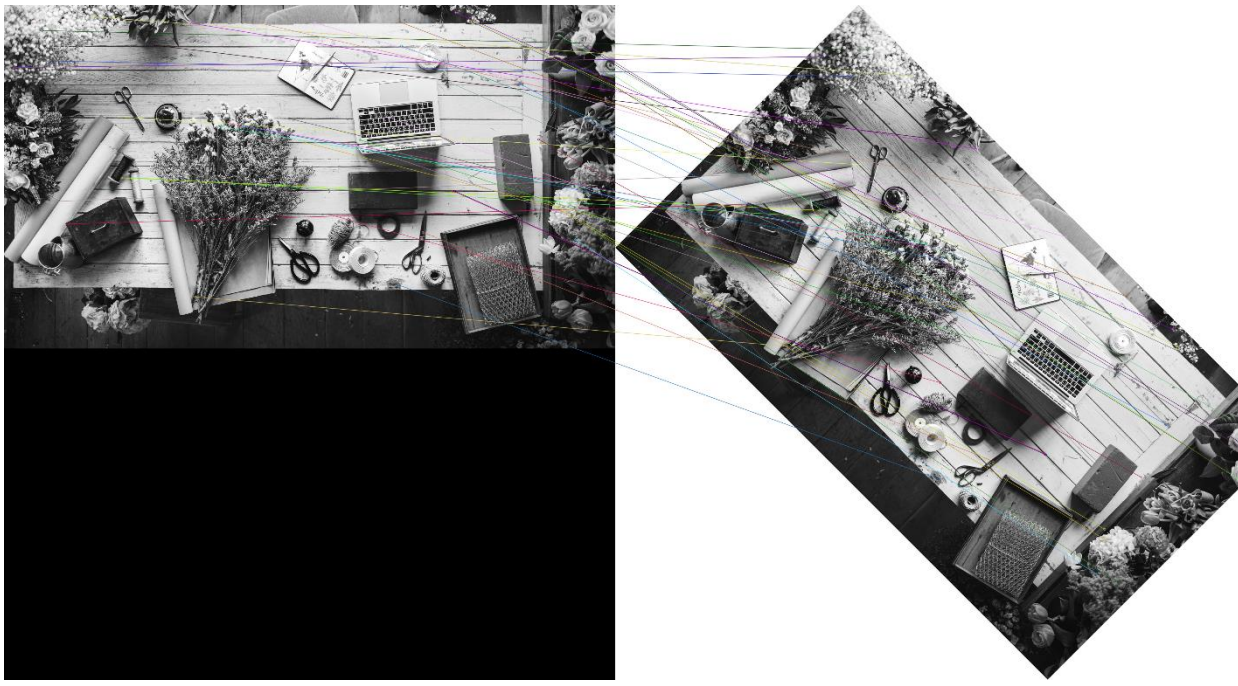
17 pav. SIFT algoritmo rezultatai lyginant skirtingus paveikslėlius

SIFT algoritmas šiame eksperimente pasirodė gana prastai, nors jis ir garsėja savo tikslumu, tačiau šioje situacijoje jis aptiko net 213 atitinkančių savybių kurių neturėjo aptikti, paveikslėlių palyginimas vidutiniškai užtruko apie 31 sekundę o tai yra beveik dvigubai ilgiau nei BRISK. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (3 priedas, 3.1 pav.).



18 pav. SIFT algoritmo rerzultatai lyginant vizualiai panašias fotografijas

Lyginant panašias fotografijas SIFT detektorius ir deskriptorius aptiko 37 sutampančias savybes, tačiau tik 14 iš jų buvo aptiktos teisingai. Visi skaičiavimai vidutiniškai truko 23.2 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (3 priedas, 3.2 pav.).



19 pav. SIFT algoritmo rezultatai lyginant pasuktas fotografijas

Lyginant fotografijas su pasuktu vaizdu SIFT algoritmas aptiko 46 sutampančias savybes, iš jų, 43 buvo aptiktos teisingai. Skaičiavimai vidutiniškai truko 80.22 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (3 priedas, 3.3 pav.).

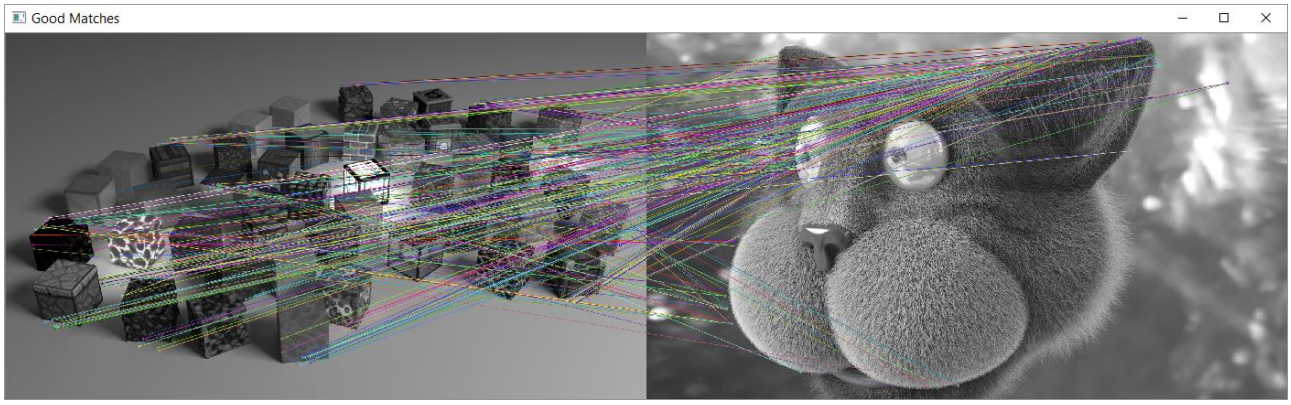
20 pav. SIFT algoritmo rezultatai lyginant fotografijas su sulietu vaizdu

Lyginant fotografijas su sulietu vaizdu SIFT algoritmas rado vieną sutampančią savybę kuri buvo teisinga. Skaičiavimai vidutiniškai truko 37,98 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (3 priedas, 3.4 pav.).

21 pav. SIFT algoritmo rezultatai lyginant skirtingos raiškos fotografijas

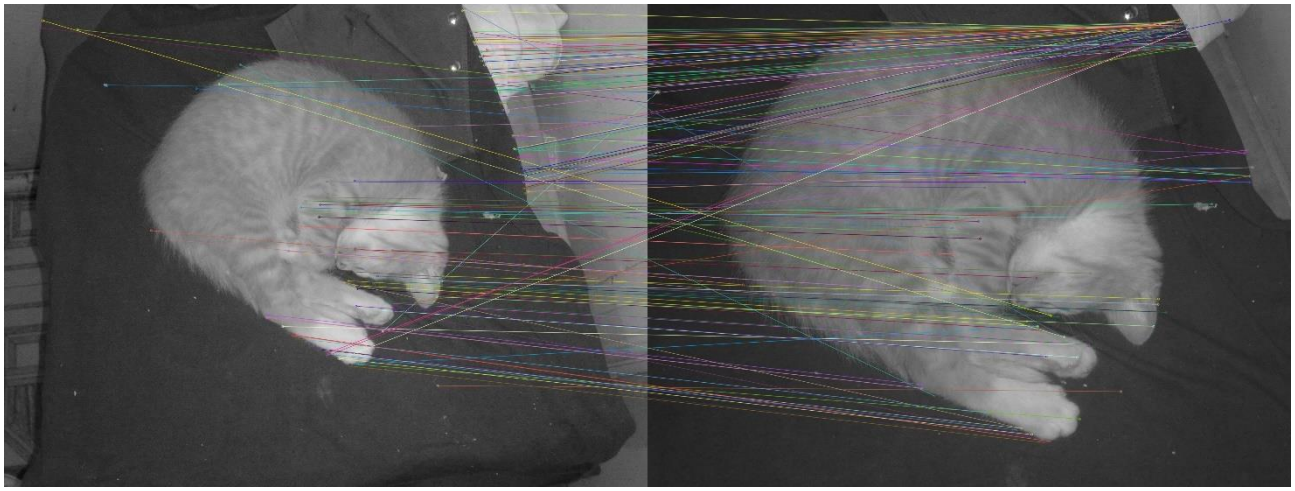
Atlikus skirtingos raiškos fotografijų palyginimą su SIFT algoritmu pastebėta, kad jis rado 25 sutampančias savybes ir jos visos buvo aptiktos teisingai. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (3 priedas, 3.5 pav.).

4.2.3 SURF algoritmo tyrimo rezultatai



22 pav. SURF algoritmo rezultatai lyginant skirtingus paveikslėlius

SURF algoritmas aptiko net 356 sutampančias savybes, nors jis turėtų būti ženkliai greitesnis už SIFT tačiau šiuo atvejo jis skaičiavimus atliko per 28 sekundes, o tai yra tik maždaug 10% greičiau nei SIFT. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (4 priedas, 4.1 pav.).



23 pav. SURF algoritmo rezultatai lyginant panašias fotografijas

Lyginant panašias fotografijas SURF detektorius ir deskriptorius aptiko net 144 sutampančias savybes tačiau iš jų teisingos buvo tik 82. Visi skaičiavimai truko 7 sekundes. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (4 priedas, 4.2 pav.).

24 pav. SURF algoritmo rezultatai lyginant pasuktas fotografijas

Lyginant pasuktas fotografijas SURF algoritmo skaičiavimai vidutiniškai užtruko 62.3 sekundės. 19 iš 23 aptiktų savybių buvo teisingos. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (4 priedas, 4.3 pav.).

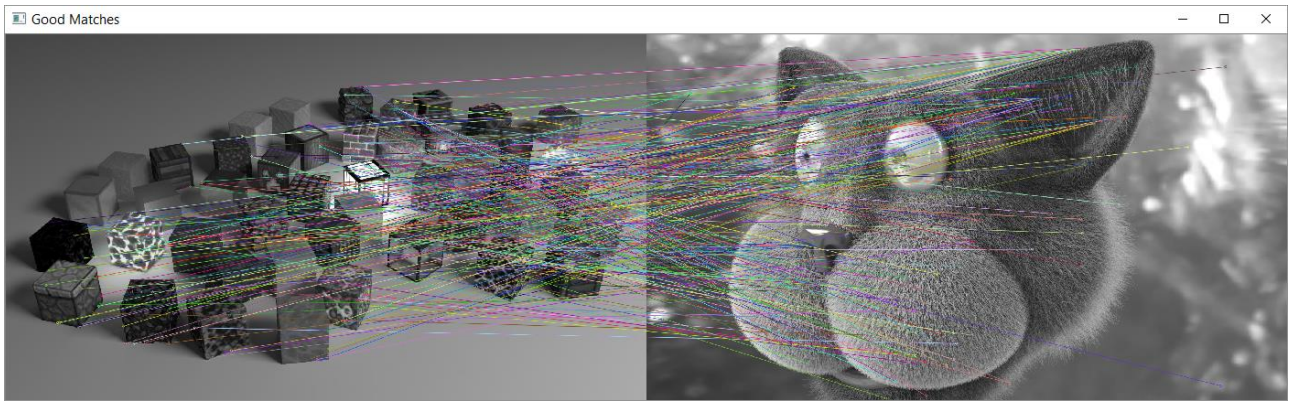
25 pav. SURF algoritmo rezultatai lyginant fotografijas su sulietu vaizdu

SURF algoritmas fotografijose su sulietu vaizdu rado 74 sutampančias savybes ir visos iš jų buvo aptiktos teisingai. Skaičiavimai vidutiniškai truko 45,29 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (4 priedas, 4.4 pav.).

26 pav. SURF algoritmo rezultatai lyginant skirtingos raiškos fotografijas

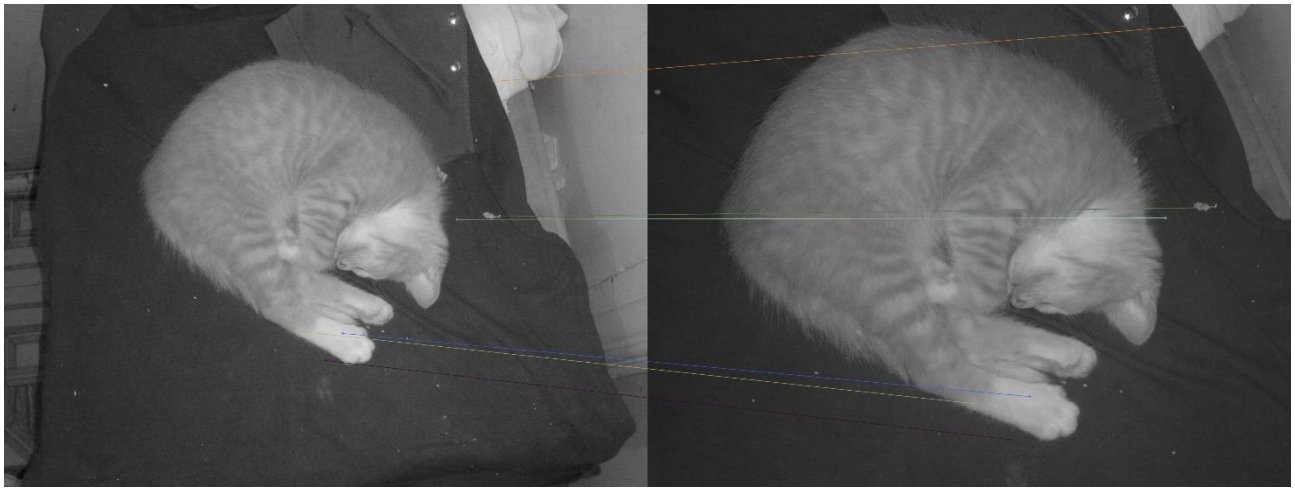
6 ir 3 megapikselių raiškos fotografijų palyginime SURF algoritmas aptiko net 712 sutampančių savybių, o maždaug 700 iš jų buvo aptiktos teisingai. Skaičiavimai vidutiniškai užtruko 192.6 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (4 priedas, 4.5 pav.).

4.2.4 SURF detektoriaus ir SIFT deskriptoriaus tyrimo rezultatai



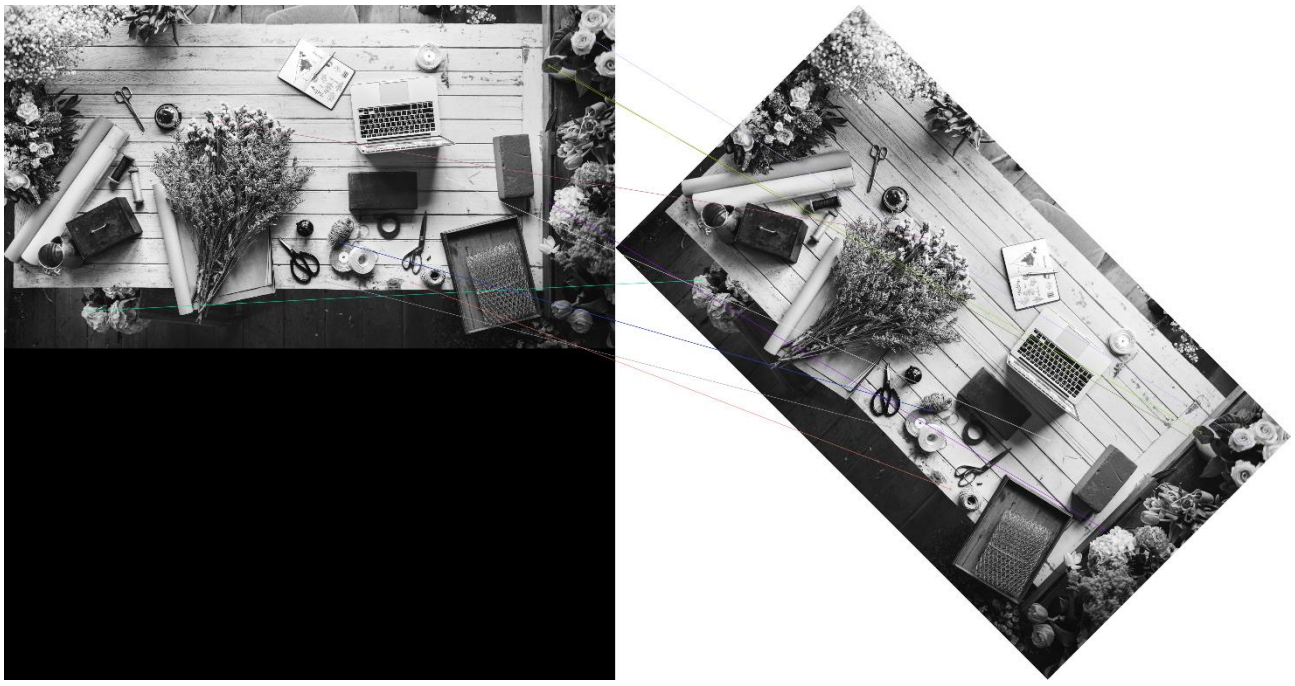
27 pav. SURF detektoriaus ir SIFT deskriptoriaus rezultatai lyginant skirtingus paveikslėlius

SURF detektoriaus ir SIFT deskriptoriaus kombinacija visus skaičiavimus atliko per 16 sekundžių, o tai yra greičiau nei SURF IR SIFT metodai atskirai. Taip pat šis metodas aptiko 365 sutampančias savybes. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (5 priedas, 5.1 pav.).



28 pav. SURF detektoriaus ir SIFT deskriptoriaus rezultatai lyginant panašias fotografijas

Lyginant panašias fotografijas SURF detektorius ir SIFT deskriptorius rado 7 sutampančias savybes ir visos iš jų buvo teisingos. Skaičiavimai vidutiniškai truko 10.8 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (5 priedas, 5.2 pav.).



29 pav. SURF detektoriaus ir SIFT deskriptoriaus rezultatai lyginant pasuktas fotografijas

Atlikus pasuktų fotografijų palyginimą su SURF detektoriumi ir SURF deskriptoriumi buvo nustatyta, kad ši algoritmų kombinacija aptiko 11 teisingai sutampančių savybių, o visų skaičiavimų laikas truko 39,59 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (5 priedas, 5.3 pav.).

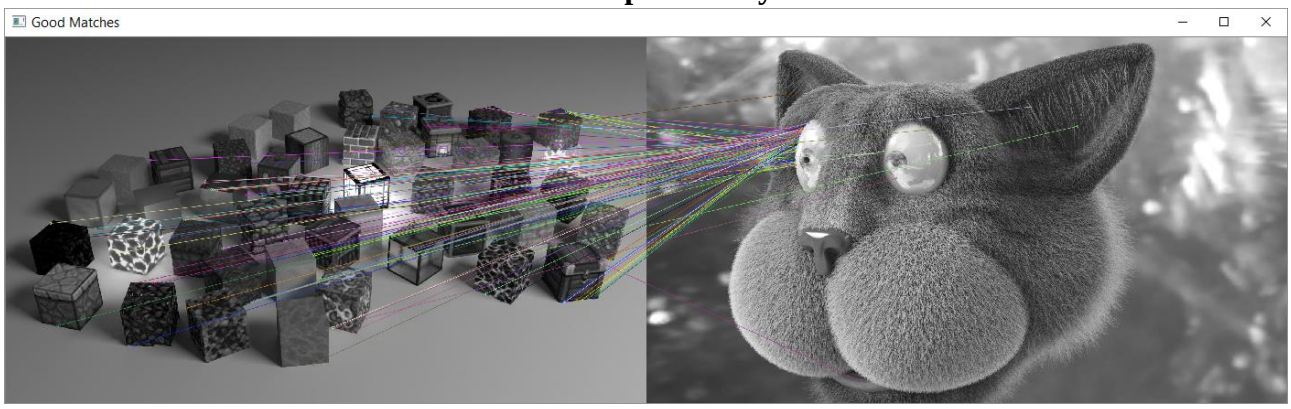
30 pav. SURF detektoriaus ir SIFT deskriptoriaus rezultatai lyginant fotografijas su sulietu vaizdu

SURF detektorius ir SIFT deskriptorius fotografijose su sulietu vaizdu iš viso rado 6 sutampančias savybes, visos jos buvo aptiktos teisingai, o skaičiavimai vidutiniškai truko19,2 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (5 priedas, 5.4 pav.).

31 pav. SURF detektoriaus ir SIFT deskriptoriaus rezultatai lyginant skirtingos raiškos fotografijas

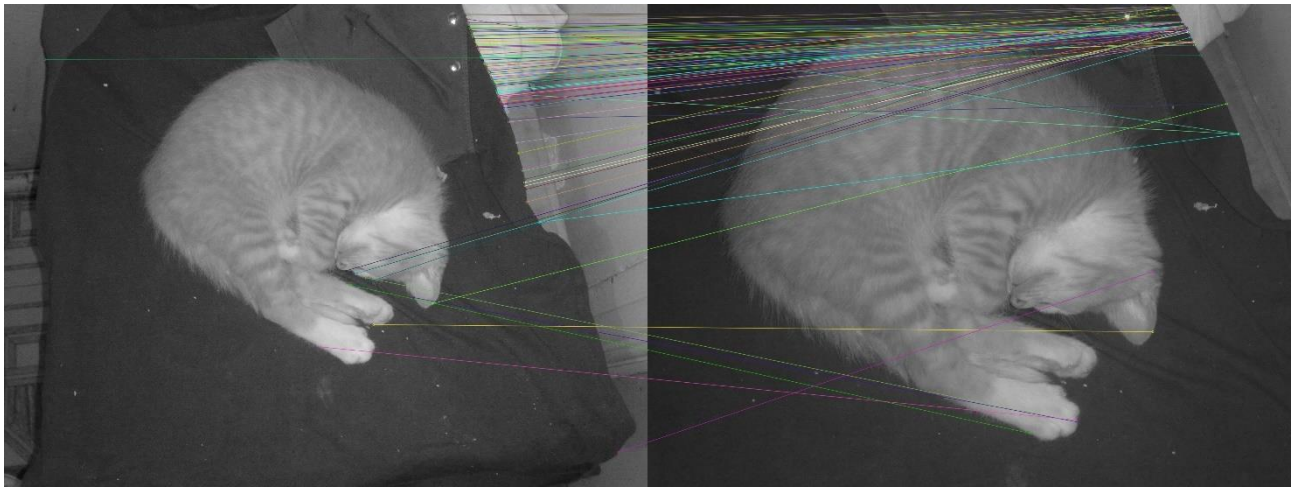
SURF detektorius ir SIFT deskriptorius ieškodamas savybių skirtingos raiškos fotografijose užtruko 108.6 sekundės bei rado 6 sutampančias savybes iš kurių 2 buvo aptiktos klaidingai. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (5 priedas, 5.5 pav.).

4.2.5 FAST detektoriaus ir SURF deskriptoriaus tyrimo rezultatai



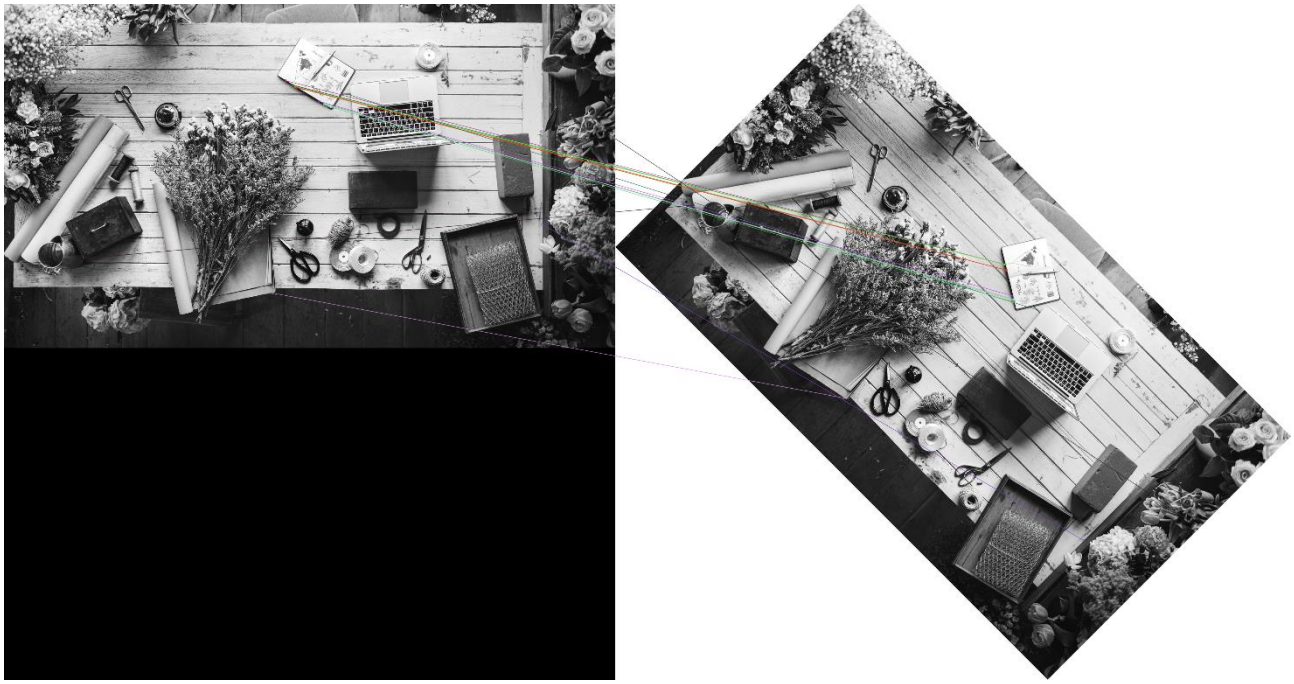
32 pav. FAST detektoriaus ir SURF deskriptoriaus rezultatai lyginant skirtingus paveikslėlius

FAST ir SURF kombinacija raktinius taškus rado per 0.45 sekundės, deskriptoriaus skaičiavimus atliko per 10 sekundžių, tačiau savybių palyginimas užtruko labai ilgai. Taip pat šis metodas rado 161 sutampančią savybę. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (6 priedas, 6.1 pav.).



33 pav. FAST detektoriaus ir SURF deskriptoriaus rezultatai lyginant panašias fotografijas

Lyginant panašias fotografijas FAST detektorius ir SURF deskriptorius aptiko 90 sutampančių savybių, tačiau tik mažiau nei 10 iš jų buvo aptiktos teisingai. Skaičiavimai vidutiniškai užtruko apie 2.48 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (6 priedas, 6.2 pav.).



34 pav. FAST detektoriaus ir SURF deskriptoriaus rezultatai lyginant pasuktas fotografijas

FAST detektorius ir SURF deskriptorius pasuktose fotografijose aptiko 18 sutampančių savybių, tačiau 6 iš jų buvo aptiktos klaidingai. Be to skaičiavimai užtruko net 284 sekundes. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (6 priedas, 6.3 pav.).

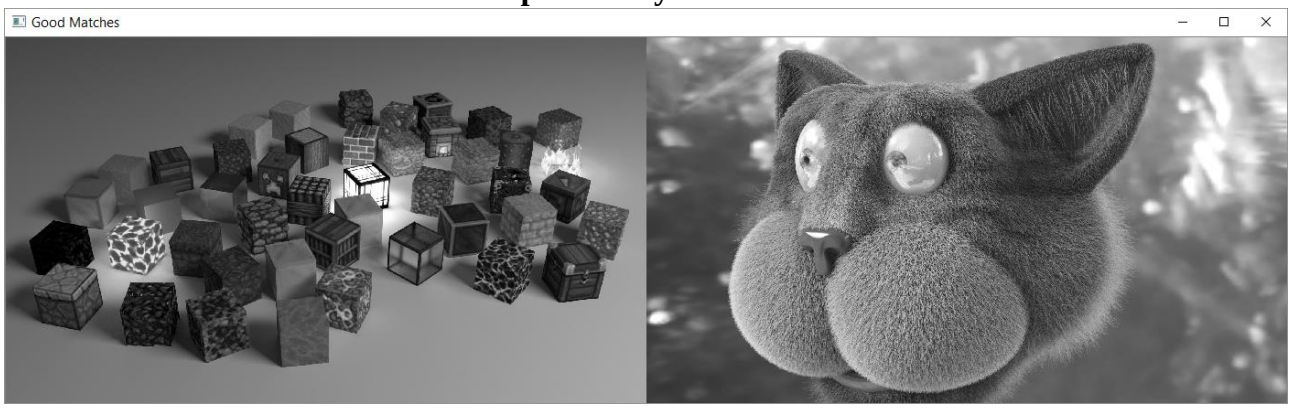
35 pav. FAST detektoriaus ir SURF deskriptoriaus rezultatai lyginant fotografijas su sulietu vaizdu

FAST detektorius ir SURF deskriptorius aptiko net 367 sutampančias savybes tačiau tik apie 20 iš jų buvo teisingos. Skaičiavimai truko 39.27 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (6 priedas, 6.4 pav.).

36 pav. FAST detektoriaus ir SURF deskriptoriaus rezultatai lyginant skirtingos raiškos fotografijas

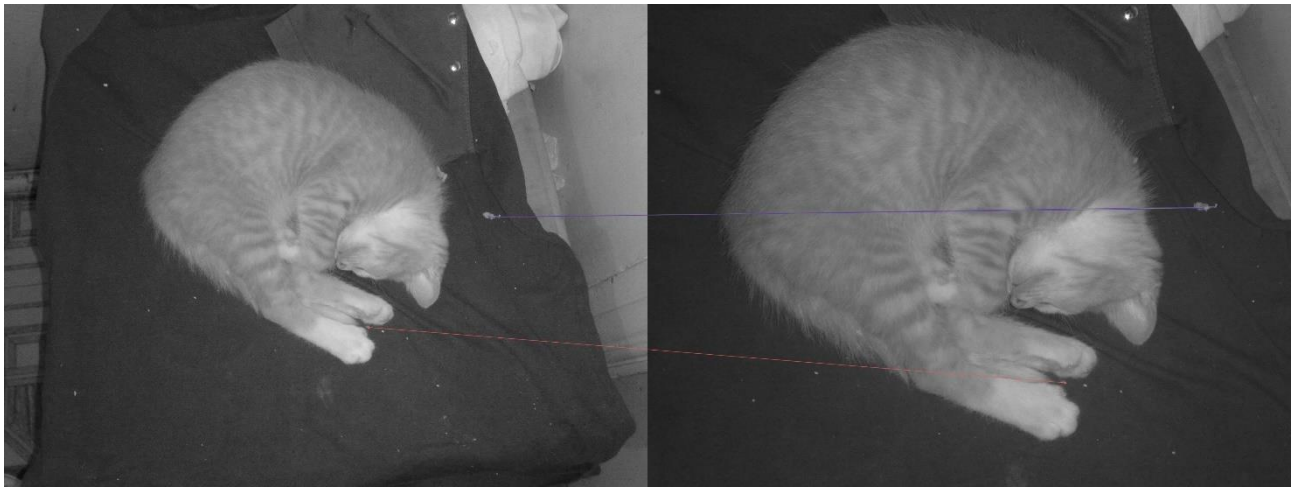
FAST detektorius ir SURF deskriptorius lyginant 3 ir 6 megapikselių raiškos fotografijas aptiko 59 sutampančias savybes, tačiau tik 25 iš jų buvo aptiktos teisingai. Be to, skaičiavimai užtruko labai ilgai, net 802.4 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (6 priedas, 6.5 pav.).

4.2.6 ORB detektoriaus ir deskriptoriaus tyrimo rezultatai



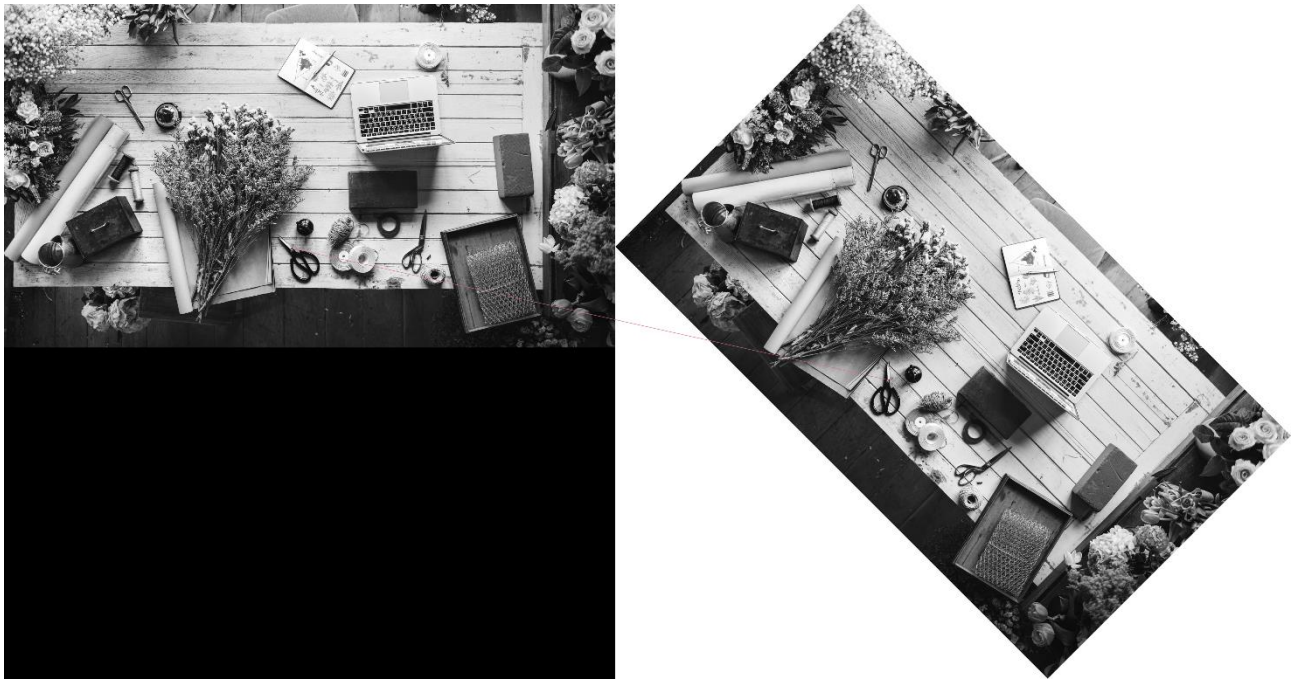
37 pav. ORB algoritmo rezultatai lyginant skirtingus paveikslėlius

ORB algoritmas paveikslėlius palygino labai greitai, vidutinė skaičiavimų trukmė buvo mažiau nei viena sekundė, taip pat šis metodas neaptiko nei vienos sutampančios savybės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (7 priedas, 7.1 pav.).



38 pav. ORB algoritmo rezultatai lyginant panašias fotografijas

Lyginant panašias fotografijas ORB detektorius ir deskriptorius aptiko tik 4 sutampančias savybes, tačiau visos jos buvo aptiktos teisingai, o skaičiavimai truko tik 0.58 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (7 priedas, 7.2 pav.).



39 pav. ORB algoritmo rezultatai lyginant pasuktas fotografijas

Pasuktose fotografijose ORB algoritmas aptiko tik vieną sutampančią savybę, kuri buvo teisinga, o jo skaičiavimo laikas truko tik 1,24 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (7 priedas, 7.3 pav.).

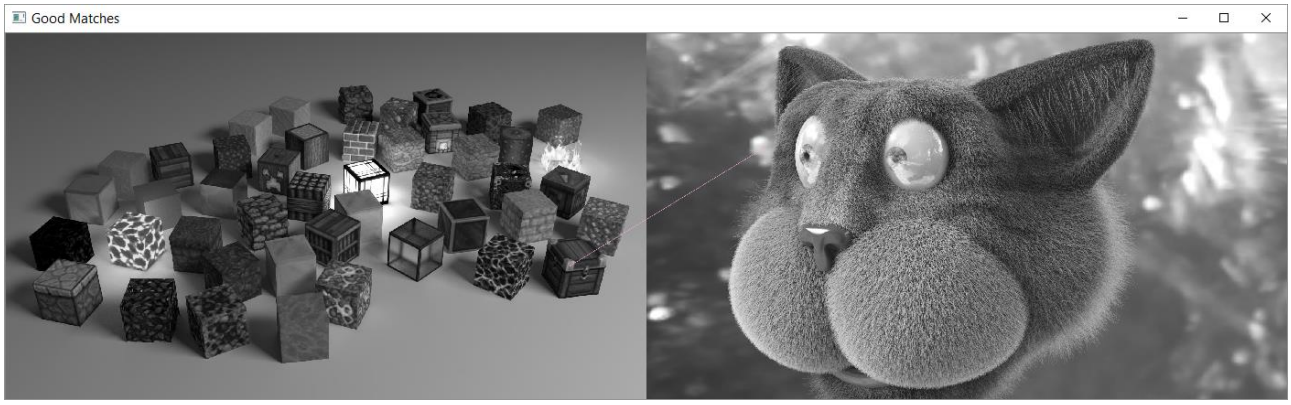
40 pav. ORB detektoriaus ir deskriptoriaus rezultatai lyginant fotografijas su sulietu vaizdu

Fotografijose su sulietu vaizdu ORB detektorius ir deskriptorius aptiko 6 sutampančias savybes ir visos iš jų buvo teisingos, o skaičiavimai truko labai trumpai, vos 0.72 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (7 priedas, 7.4 pav.).

41 pav. ORB detektoriaus ir deskriptoriaus rezultatai lyginant skirtingos raiškos fotografijas

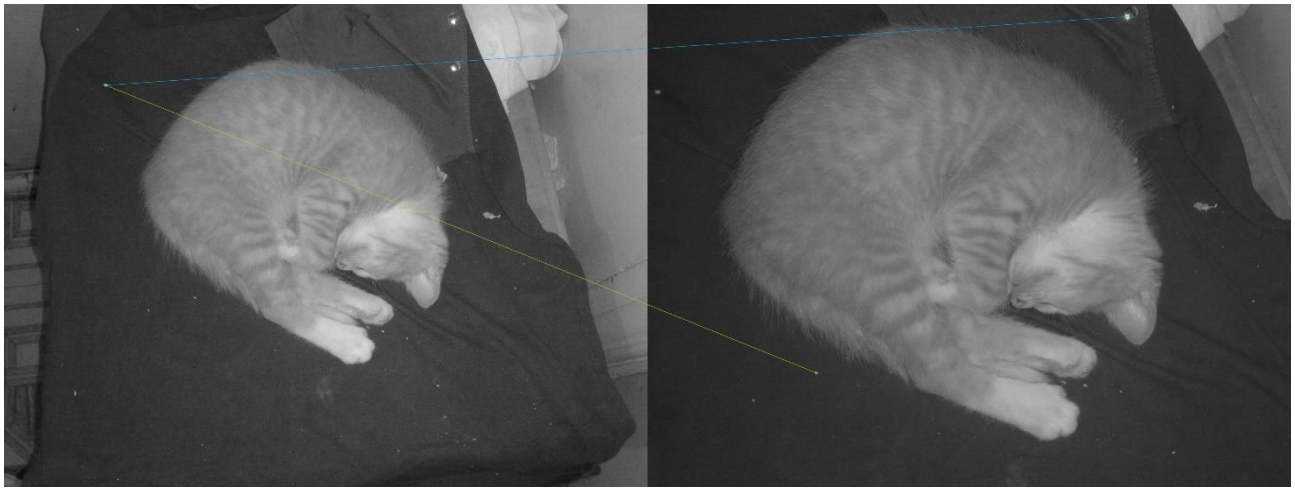
ORB algoritmas skirtingos raiškos fotografijose rado net 9 sutampančias savybes iš kurių visos jos buvo aptiktos teisingai, be to ORB skaičiavimai užtruko tik 1.94 sekundės, o tai yra daug greičiau nei bet kuris kitas eksperimentų algoritmas. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (7 priedas, 7.5 pav.).

4.2.7 FREAK detektoriaus ir MSER deskriptoriaus tyrimo rezultatai



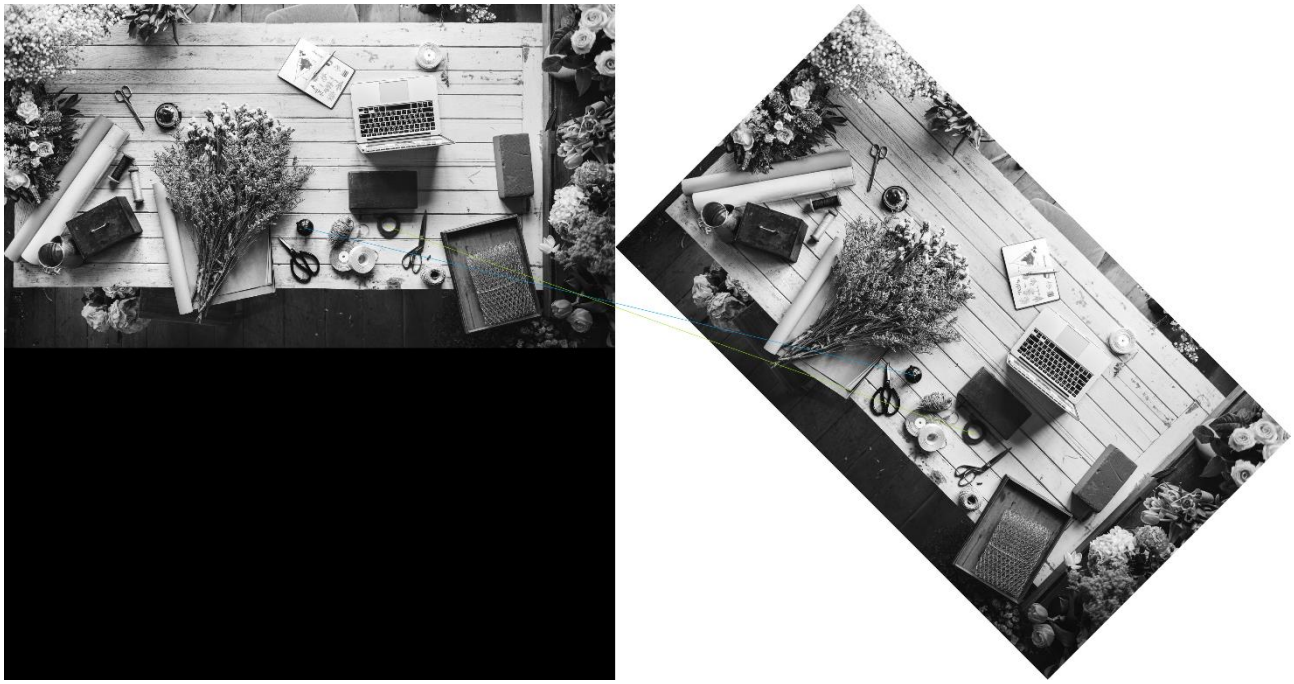
42 pav. FREAK detektoriaus ir MSER deskriptoriaus rezultatai lyginant skirtingus paveikslėlius

FREAK ir MSER kombinacija aptiko tik vieną sutampančią savybę, o visus skaičiavimus atliko per 3.5 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (8 priedas, 8.1 pav.).



43 pav. FREAK detektoriaus ir MSER deskriptoriaus rezultatai lyginant panašias fotografijas

Lyginant panašias fotografijas FREAK detektorius ir MSER deskriptorius aptiko tik dvi sutampančias savybes tačiau nei viena iš jų nebuvo aptikta teisingai. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (8 priedas, 8.2 pav.).

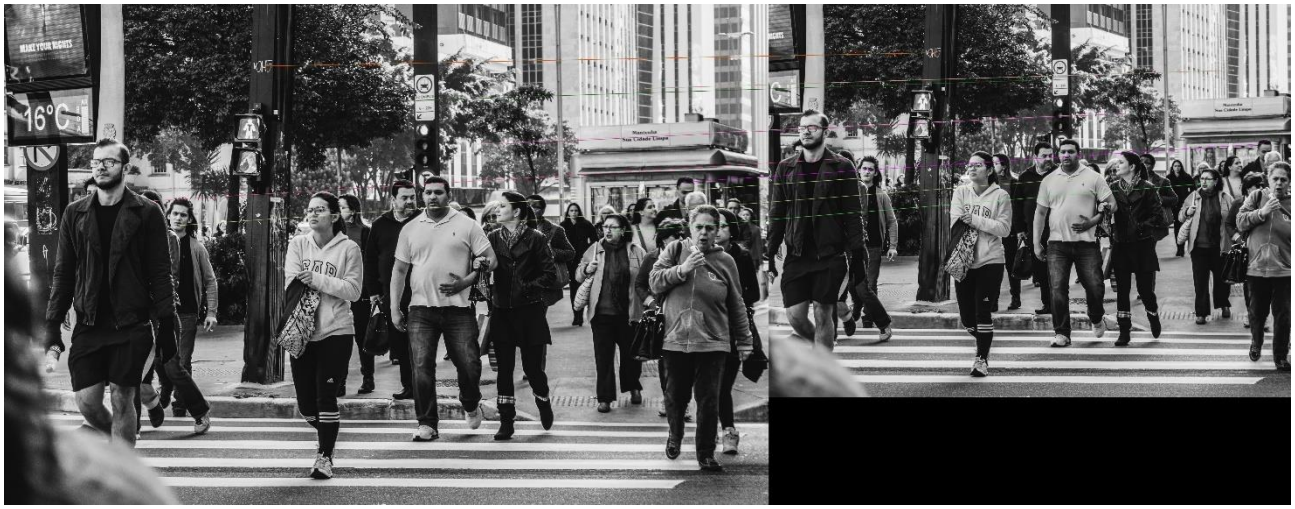


44 pav. FREAK detektoriaus ir MSER deskriptoriaus rezultatai lyginant pasuktas fotografijas

Lyginant pasuktas fotografijas, FREAK detektorius ir MSER deskriptorius aptiko dvi sutampančias savybes, o skaičiavimai vidutiniškai truko 17,54 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (8 priedas, 8.3 pav.).

45 pav. FREAK detektoriaus ir MSER deskriptoriaus rezultatai lyginant fotografijas su sulietu vaizdu

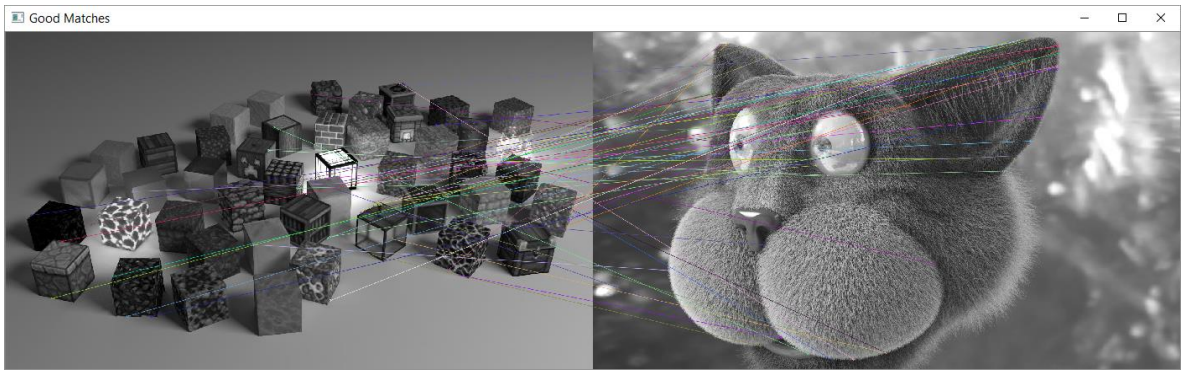
FREAK detektorius ir MSER deskriptorius fotografijose su sulietu vaizdu iš viso aptiko 5 sutampančias savybes ir visos iš jų buvo teisingos. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (8 priedas, 8.4 pav.).



46 pav. FREAK detektoriasu ir MSER deskriptoriaus rezultatai lyginant skirtingos raiškos fotografijas

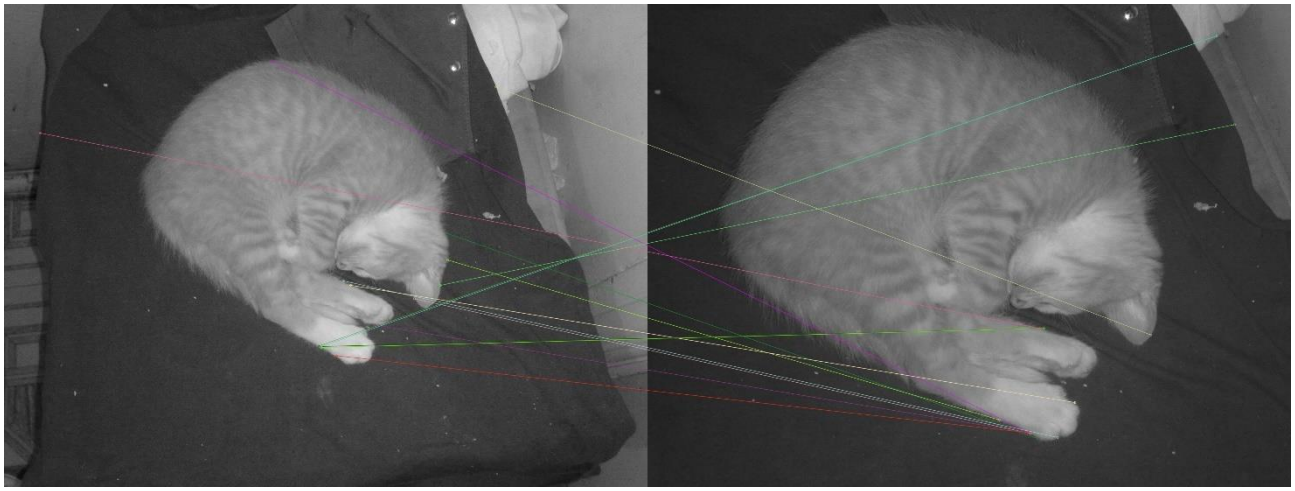
FREAK detektorius ir MSER deskriptorius rado 16 sutampančių savybių iš kurių visos buvo aptiktos teisingai, o skaičiavimai truko tik 33.5 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (8 priedas, 8.5 pav.).

4.2.8 FREAK detektoriaus ir AGAST deskriptoriaus



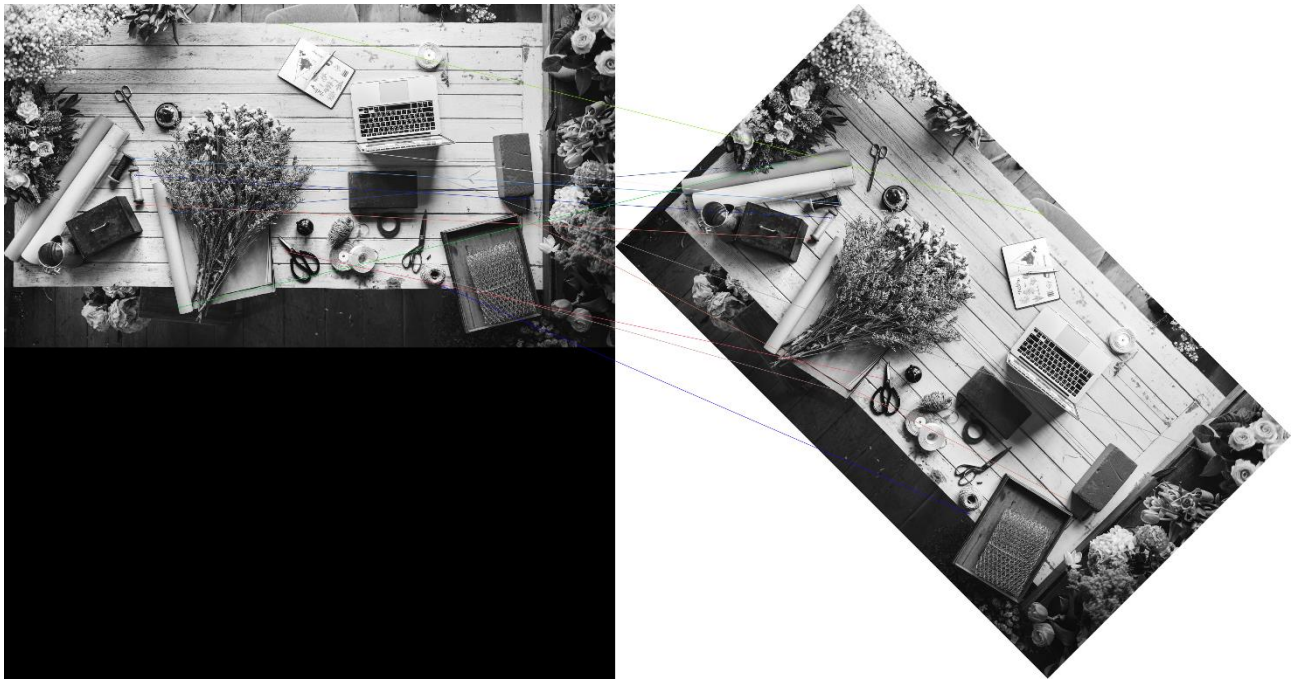
47 pav. FREAK detektoriaus ir AGAST deskriptoriaus rezultatai lyginant skirtingus paveikslėlius

FREAK ir AGAST skaičiavimai užtruko net 100 sekundžių, taip pat skaičiavimų metu buvo aptiktos 64 sutampančios savybės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (9 priedas, 9.1 pav.).



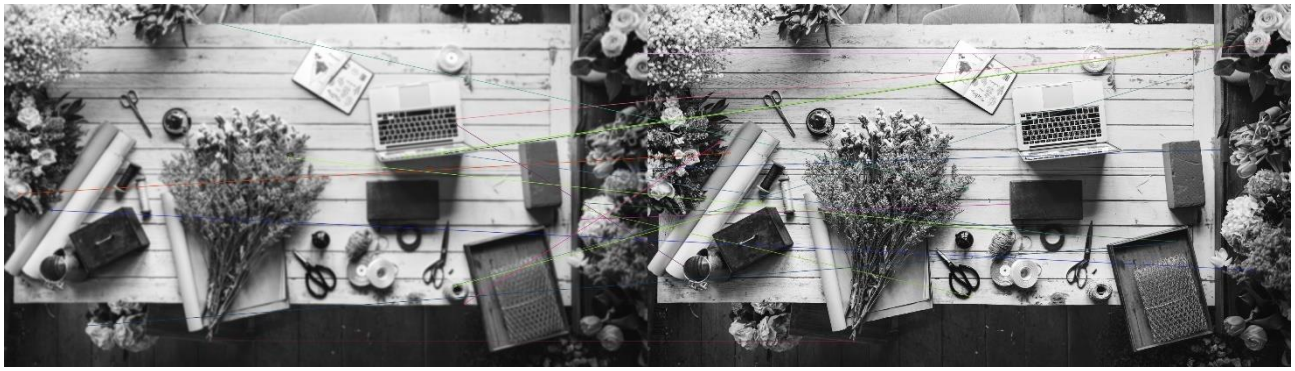
48 pav. FREAK detektoriaus ir AGAST deskriptoriaus rezultatai lyginant panašias fotografijas

Panašių fotografijų palyginime FREAK detektorius ir AGAST deskriptorius aptiko net 16 sutampančių, tačiau tik viena iš jų buvo teisinga. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (9 priedas, 9.2 pav.).



49 pav. FREAK detektoriaus ir AGAST deskriptoriaus rezultatai lyginant pasuktas fotografijas

Bandant palyginti pasuktas fotografijas su FREAK detektoriumi ir AGAST deskriptoriumi buvo nustatyta, kad jų skaičiavimo laikas vidutiniškai truko 282 sekundės, buvo aptiktos 12 sutampančių savybių, iš kurių 3 buvo aptiktos klaidingai. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (9 priedas, 9.3 pav.).



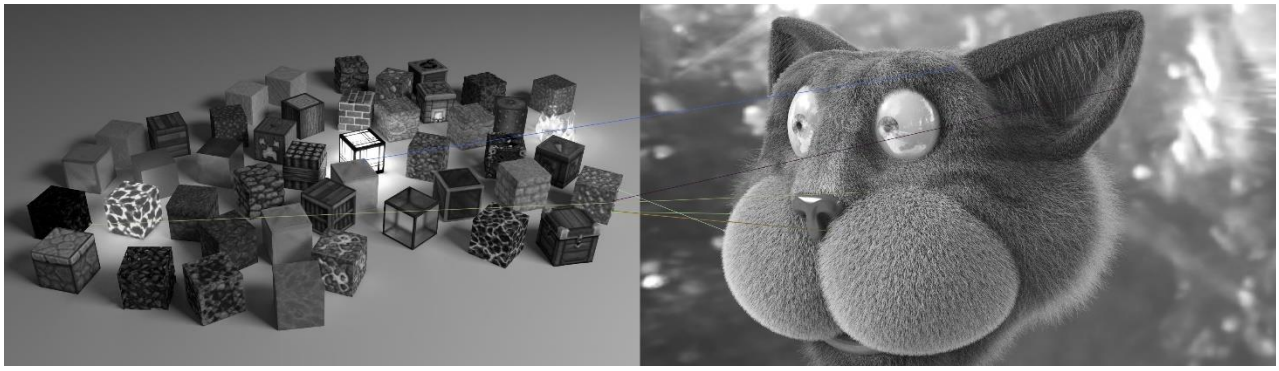
50 pav. FREAK detektoriaus ir AGAST deskriptoriaus rezultatai lyginant fotogrfijas su sulietu vaizdu

Lyginant fotografijas su sulietu vaizdu FREAK detektorius ir AGAST deskriptorius aptiko 22 sutampančias savybes iš kurių tik viena buvo aptikta teisingai. Visi skaičiavimai užtruko 36,9 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (99 priedas, 9.4 pav.).

51 pav. FREAK detektoriaus ir AGAST deskriptoriaus rezultatai lyginant skirtingos raikos fotografijas

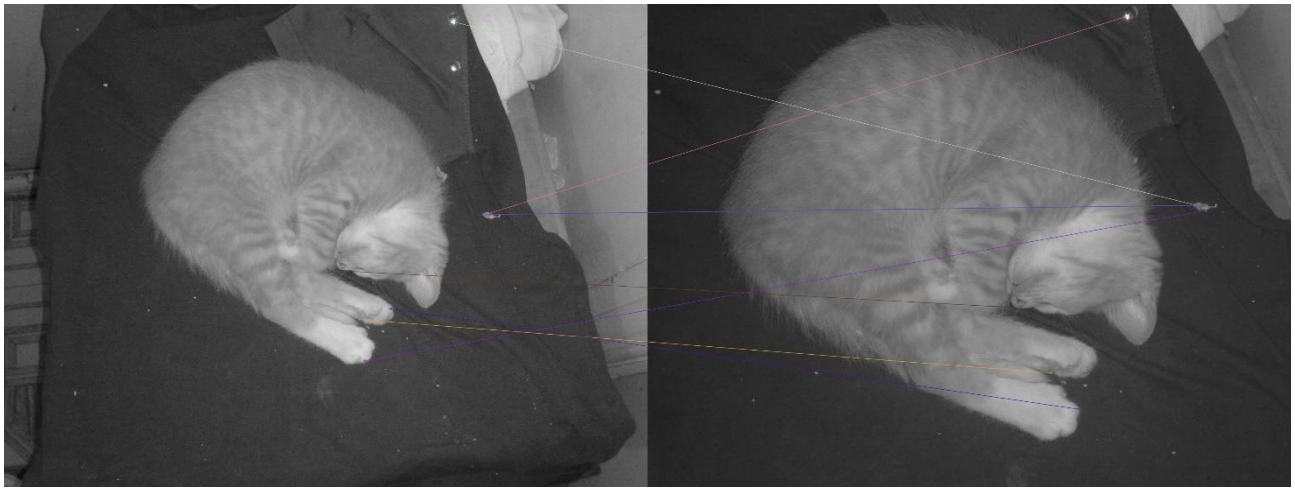
Lygindamas 3 ir 6 megapikselių raiškos fotografijas FREAK detektorius ir AGAST deskriptorius rado 10 sutampančių savybių, tačiau iš jų net 7 buvo aptiktos klaidingai. Skaičiavimai užtruko 734 sekundes. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (9 priedas, 9.5 pav.).

4.2.9 BRIEF detektoriaus ir BRISK deskriptoriaus



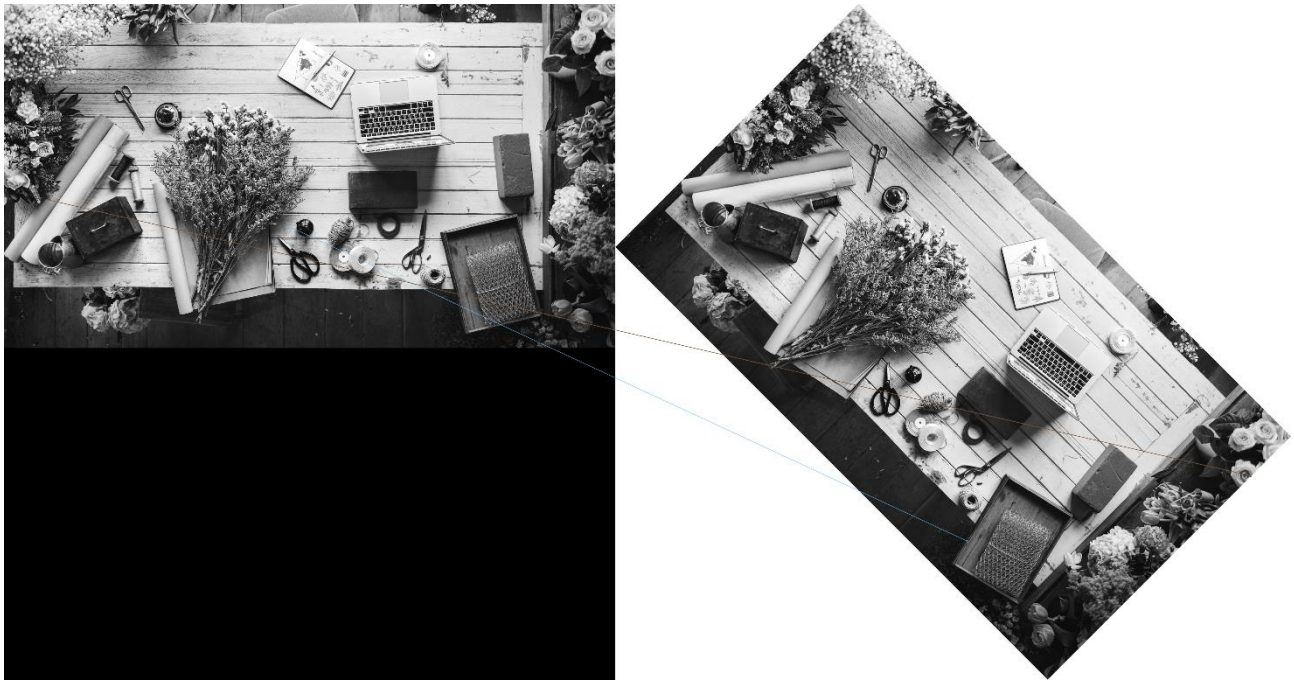
52 pav. BRIEF detektoriaus ir BRISK deskriptoriaus rezultatai lyginant skirtingus paveikslėlius

BRIEF detektorius ir BRISK deskriptorius aptiko 7 sutampančias, o skaičiavimai vidutiniškai užtruko apie 14 sekundžių. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (10 priedas, 10.1 pav.).



53 pav. BRIEF detektoriaus ir BRISK deskriptoriaus rezultatai lyginant panašias fotografijas

Lyginant panašias fotografijas BRIEF detektorius ir BRISK deskriptorius aptiko 8 sutampančias savybes tačiau tik pusė iš jų buvo aptiktos teisingai. Skaičiavimai vidutiniškai truko 0.41 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (10 priedas, 10.2 pav.).



54 pav. BRIEF detektoriaus ir BRISK deskriptoriaus rezultatai lyginant pasuktas fotografijas

BRIEF detektorius ir BRISK deskriptorius pasuktose fotografijose rado 3 sutampančias savybes, bet visos jos buvo aptiktos klaidingai, o skaičiavimai užtruko 75 sekundes. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (10 priedas, 10.3 pav.).

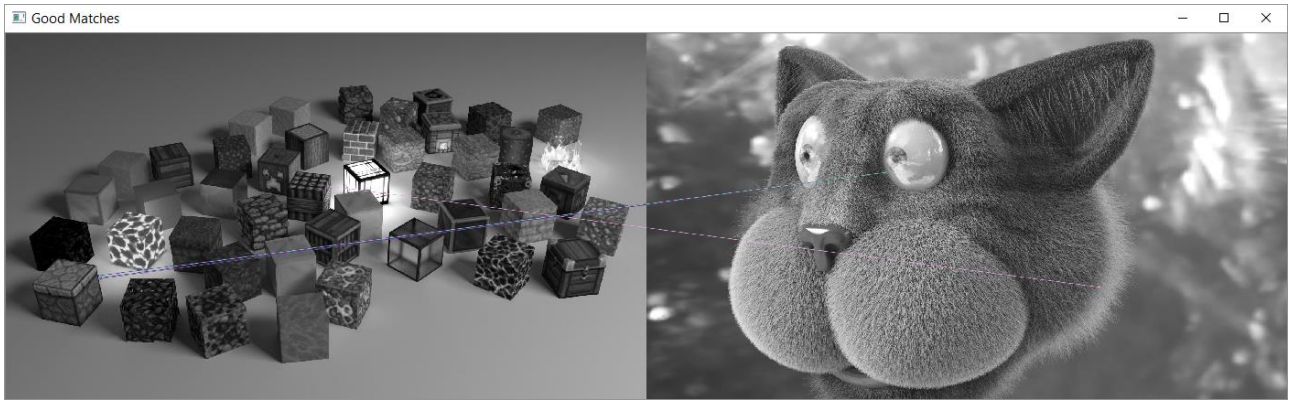
55 pav. BRIEF detektoriaus ir BRISK deskriptoriaus rezultatai lyginant fotogrfijas si sulietu vaizdu

Atlikus fotografijų su sulietu vaizdu palyginimą su BRIEF detektoriumi ir BRISK deskriptoriumi buvo nustatyta, kad jis aptiko 2 sutampančias savybes iš kurių abi buvo teisingos. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (10 priedas, 10.4 pav.).

56 pav. BRIEF detektoriaus ir BRISK deskriptorius rezultatai lyginant skirtingos raiškos fotografijas

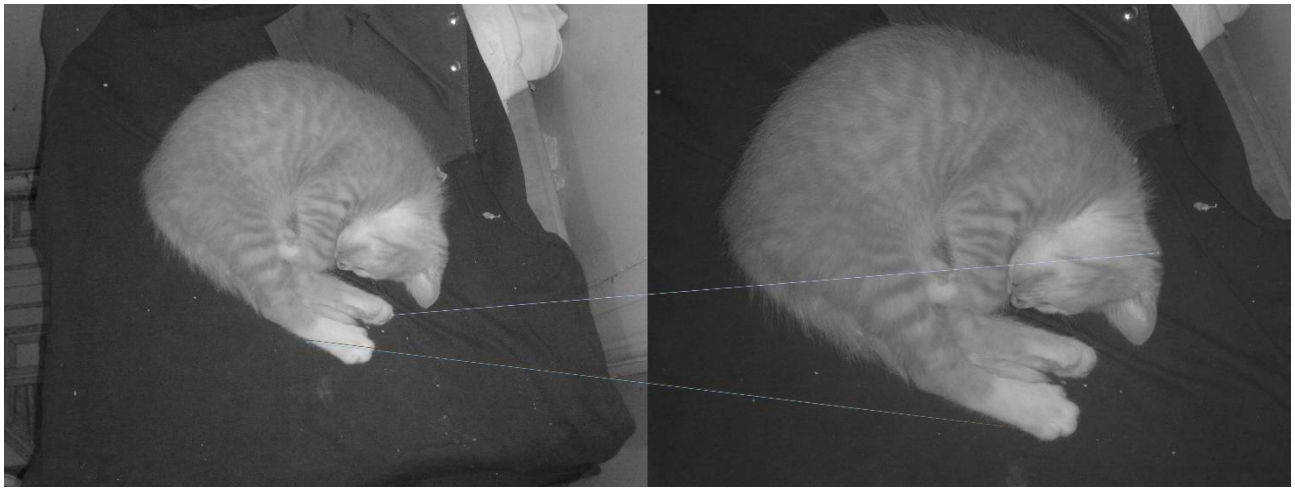
Skirtingos raiškos fotografijose BRIEF detektorius ir BRISK deskriptorius rado 2 sutampančias savybes iš kurių abi buvo aptiktos teisingai. Skaičiavimai vidutiniškai užtruko 121.68 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (10 priedas, 10.5 pav.).

4.2.10 BRIEF detektoriaus ir SURF deskriptoriaus tyrimo rezultatai



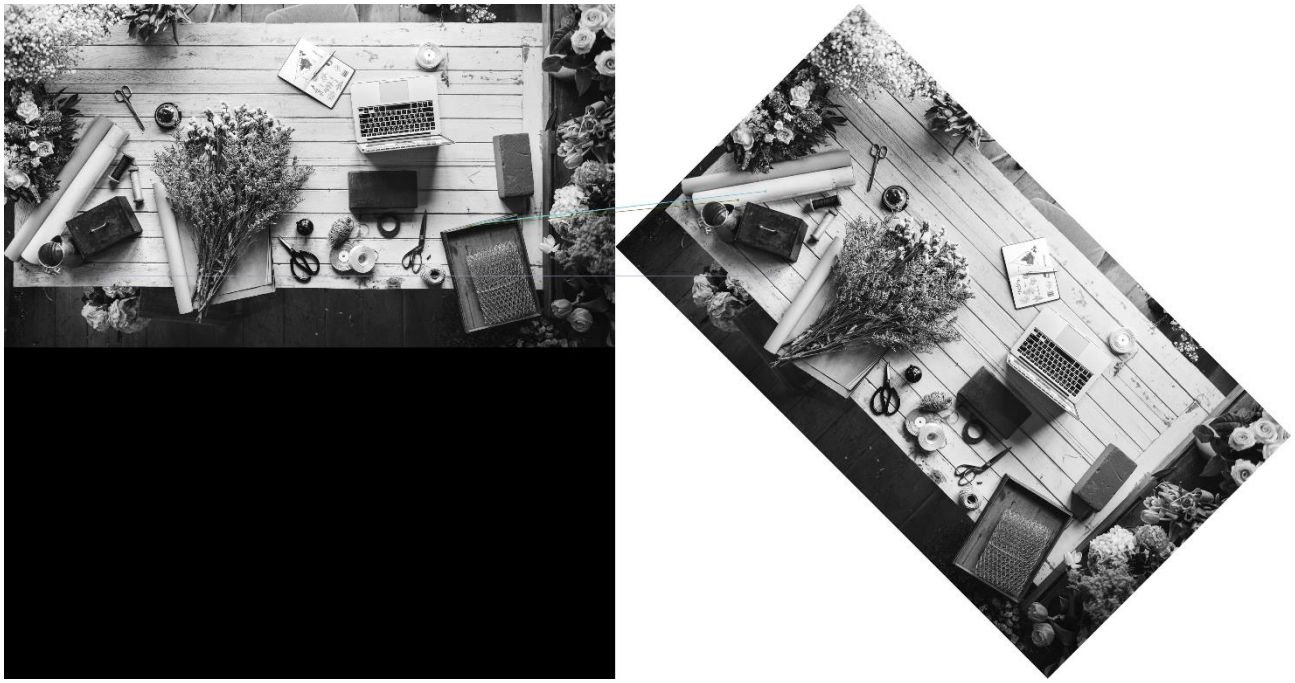
57 pav. BRIEF detektoriaus ir SURF deskriptoriaus rezultatai lyginant skirtingus paveikslėlius

BRIEF ir SURF kombinacija skaičiavimus vidutiniškai atliko per 13 sekundžių ir rado 5 sutampančias savybes. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (11 priedas, 11.1 pav.).



58 pav. BRIEF detektoriaus ir SURF deskriptoriaus rezultatai lyginant panašias fotografijas

Lyginant panašias fotografijas BRIEF detektorius ir SURF deskriptorius aptiko 3 sutampančias savybes, tačiau visos iš jų buvo klaidingos. Visi skaičiavimai truko apie 4 sekundes. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (11 priedas, 11.2 pav.).



59 pav. BRIEF detektoriaus ir SURF deskriptoriaus rezultatai lyginant pasuktas fotografijas

Lyginant pasuktas fotografijas BRIEF detektorius ir SURF deskriptorius užtruko 31.7 sekundės bei rado 5 sutampančias savybes iš kurių tik 1 buvo aptikta teisingai. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (11 priedas, 11.3 pav.).

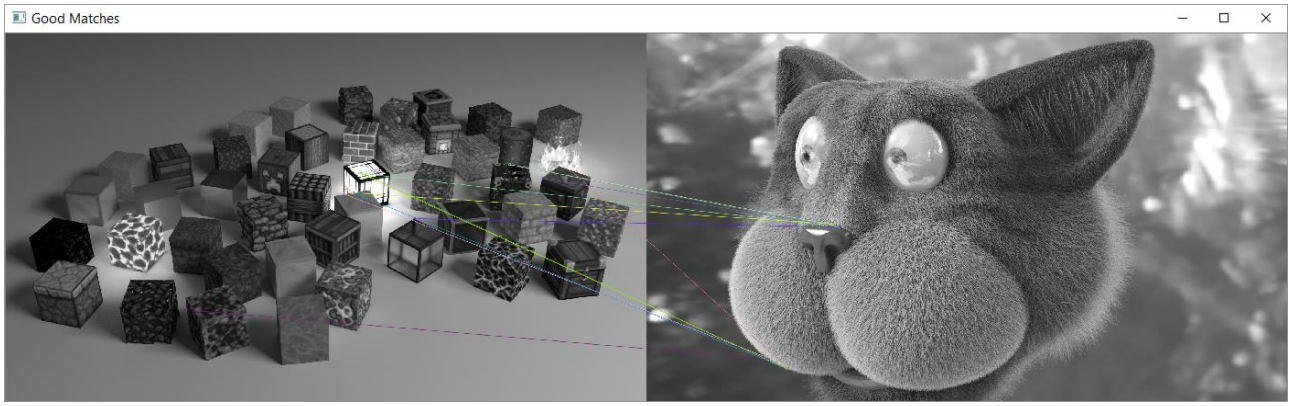
60 pav. BRIEF detektoriaus ir SURF deskriptoriaus rezultatai lyginant fotografijas su sulietu vaizdu

BRIEF detektorius ir SURF deskriptorius fotografijose su sulietu vaizdu aptiko 2 sutampančias savybes iš kurių abi buvo aptiktos teisingai. Visi skaičiavimai užtruko vidutiniškai 12.5 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (11 priedas, 11.4 pav.).

61 pav. BRIEF detektoriaus ir SURF deskriptoriaus rezultatai lyginant skirtingos raiškos forografijas

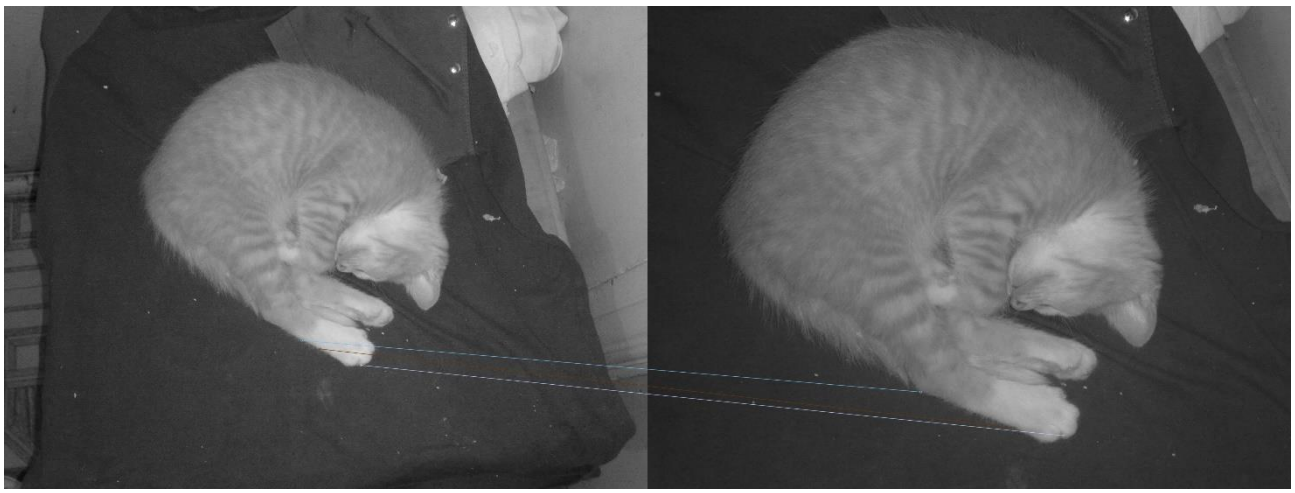
BRIEF detektorius ir SURF deskriptorius lygindamas 3 ir 6 megapikselių raiškos fotografijas užtruko 91.14 sekundės bei aptiko 1 teisingai sutampančią savybę. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (11 priedas, 11.5 pav.).

4.2.11 ORB detektoriaus ir GFTT deskriptoriaus tyrimo rezultatai



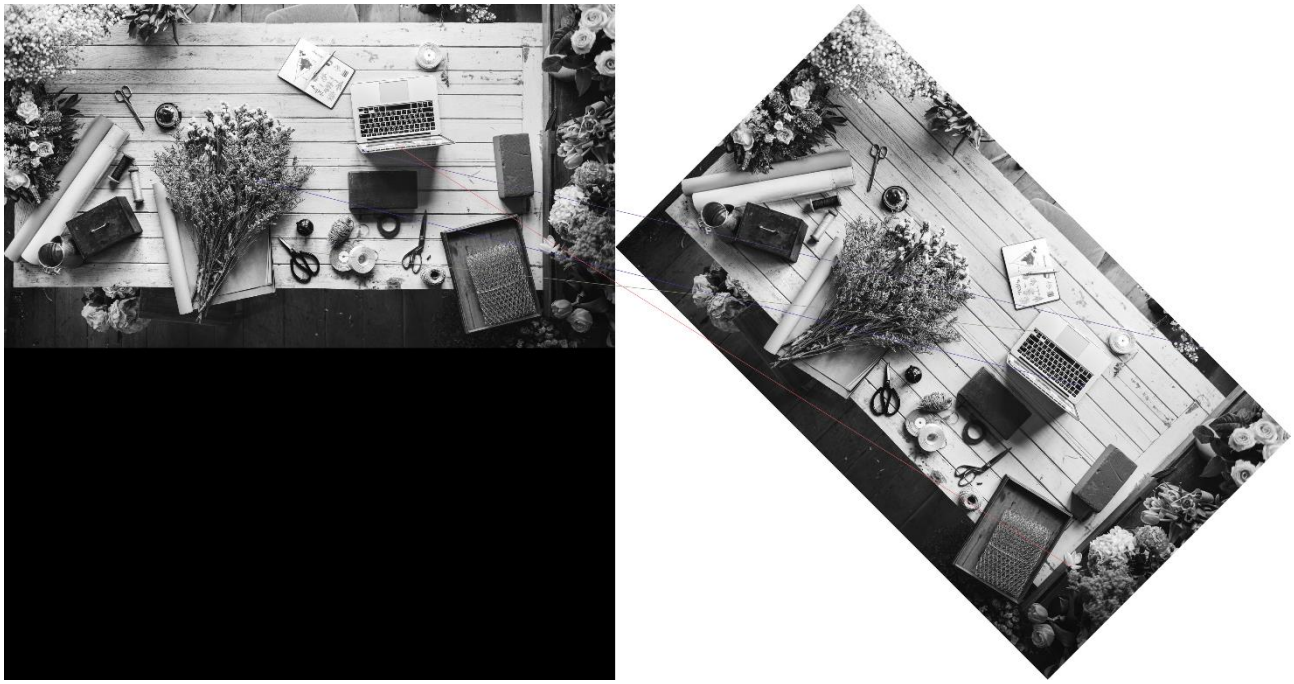
62 pav. ORB detektoriaus ir GFTT deskriptoriaus rezultatai lyginant skirtingus paveikslėlius

ORB ir GFTT visus skaičiavimus atliko per mažiau nei 5 sekundes bei rado 12 sutampančių savybių. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (12 priedas, 12.1 pav.).



63 pav. ORB deskriptoriaus ir GFTT detektoriaus rezultatai lyginant panašias fotografijas

Atlikus panašių fotografijų palyginimą ORB detektorius ir GFTT deskriptorius rado tik tris sutampančias savybes iš kurių 2 buvo aptiktos klaidingai. Skaičiavimai vidutiniškai truko 8.3 sekundės Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (12 priedas, 12.2 pav.).



64 pav. ORB deskriptoriaus ir GFTT detektoriaus rezultatai

Pasuktose fotografijose ORB deskriptorius ir GFTT detektorius iš viso rado 4 sutampančias savybes, tačiau nei viena iš jų nebuvo aptikta teisingai. Skaičiavimai vidutiniškai truko 10 sekundžių. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (12 priedas, 12.3 pav.).

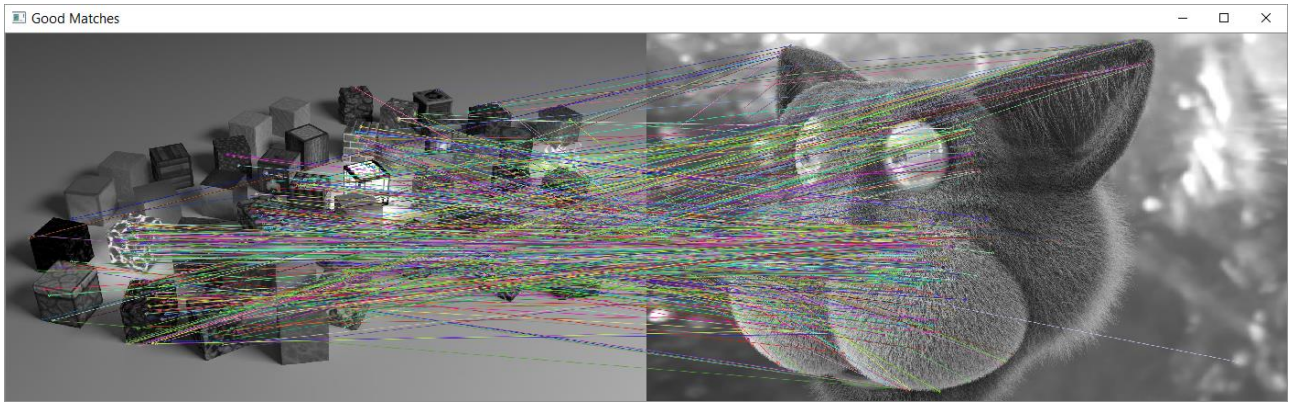
65 pav. ORB detektoriaus ir GFTT deskriptoriaus rezultatai

Lyginant fotografijas su sulietu vaizdu ORB detektorius ir GFTT deskriptorius iš viso aptiko 32 sutampančias savybes, tačiau iš jų tik 26 buvo aptiktos teisingai, o visi skaičiavimai užtruko 4.78 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (12 priedas, 12.4 pav.).

66 pav. ORB detektoriaus ir GFTT deskriptoriaus rezultatai lyginant skirtingos raiškos fotografijas

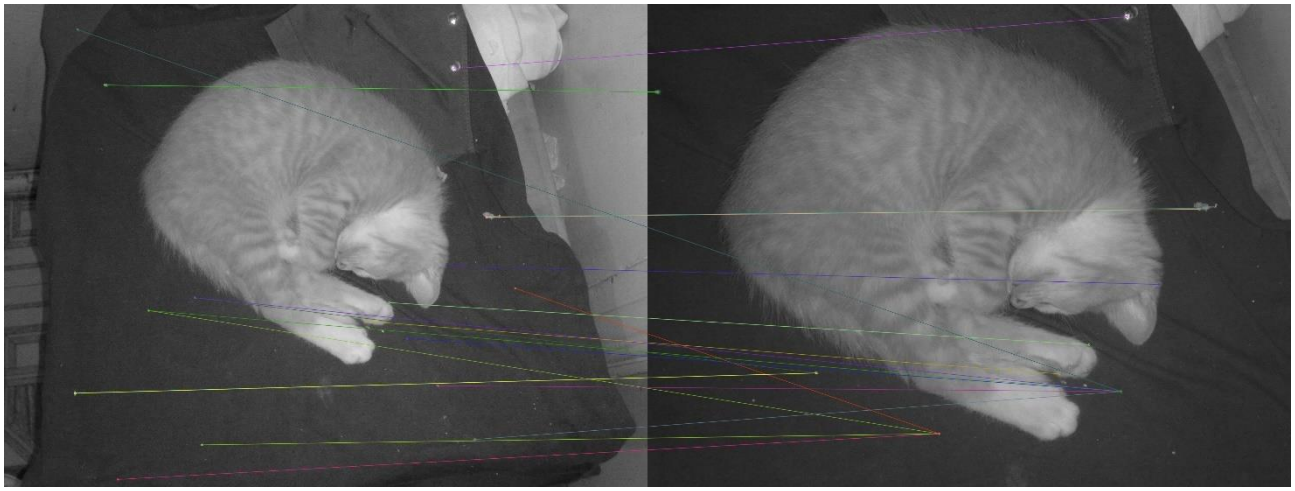
ORB detektorius ir GFTT deskriptorius skirtingos raiškos fotografijose aptiko 12 sutampančių savybių iš kurių tik 3 buvo teisingos, skaičiavimai truko 11.3 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (12 priedas, 12.5 pav.).

4.2.12 KAZE detektoriaus ir deskriptoriaus tyrimo rezultatai



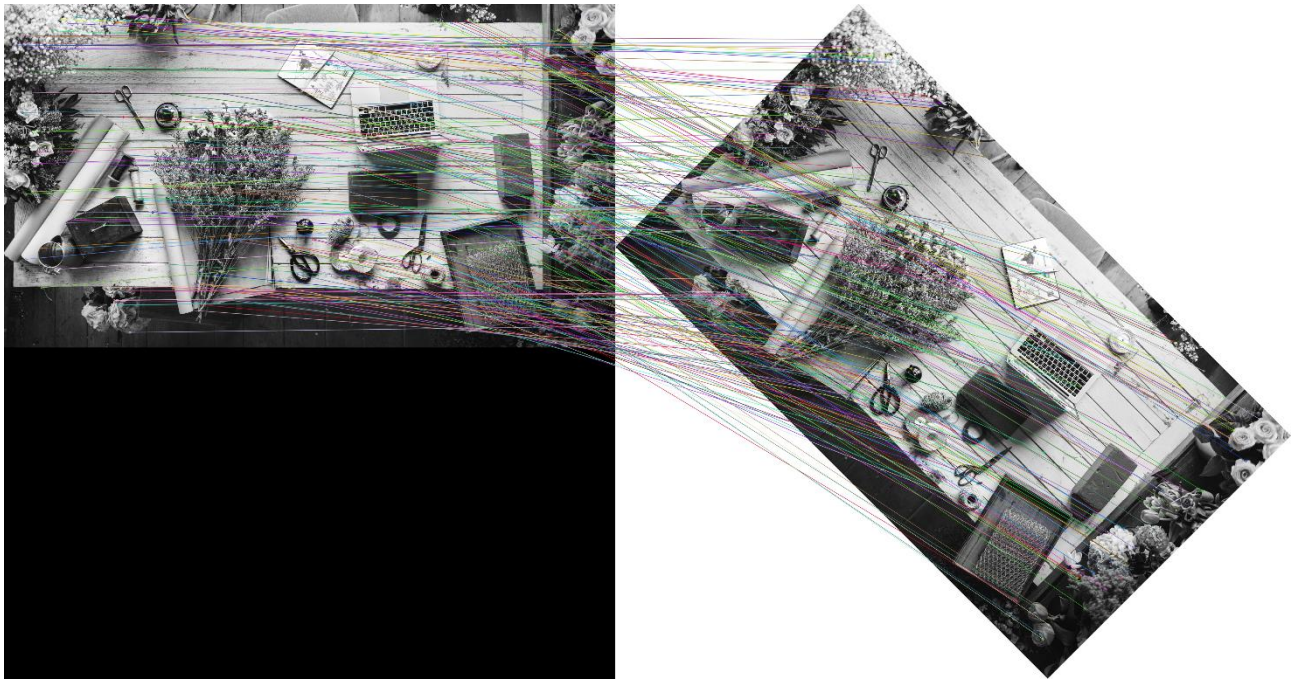
67 pav. KAZE algoritmo rezultatai lyginant skirtingus paveikslėlius

KAZE rado daugiausiai savybių, net 739, be to šio algoritmo skaičiavimo laikas buvo vienas iš ilgiausių. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (13 priedas, 13.1 pav.).



68 pav. Kaze algoritmo rezultatai lyginant panašias fotografijas

Lyginant panašias fotografijas KAZE detektorius ir deskriptorius aptiko net 24 sutampančias savybes, tačiau tik 7 iš jų buvos teisingos. Skaičiavimai truko daugiau nei 100 sekundžių. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (13 priedas, 13.2 pav.).



69 pav. KAZE algoritmo rezultatai lyginant pasuktas fotografijas

Lyginant pasuktas fotografijas KAZE algoritmas aptiko net 282 sutampančias savybes, iš kurių visos buvo aptiktos teisingai, tačiau skaičiavimai užtruko net 286.8 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (13 priedas, 13.3 pav.).

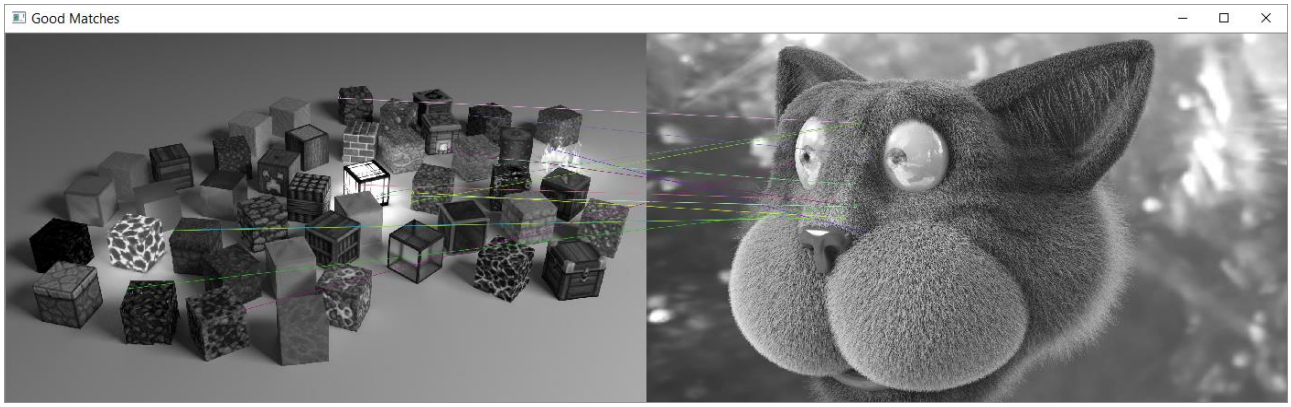
70 pav. KAZE detektoriaus rezultatai lyginant fotografijas su sulietu vaizdu

Fotografijose su sulietu vaizdu KAZE algoritmas aptiko net 33 sutampančias savybes bei visos iš jų buvo aptiktos teisingai. Nors KAZE ir pasižymėjo dideliu tikslumu, tačiau jo skaičiavimai užtruko net 192 sekundes. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (13 priedas, 13.4 pav.).

71 pav. KAZE detektoriaus ir deskriptoriaus rezultatai lyginant skirtingos raiškos fotografijas

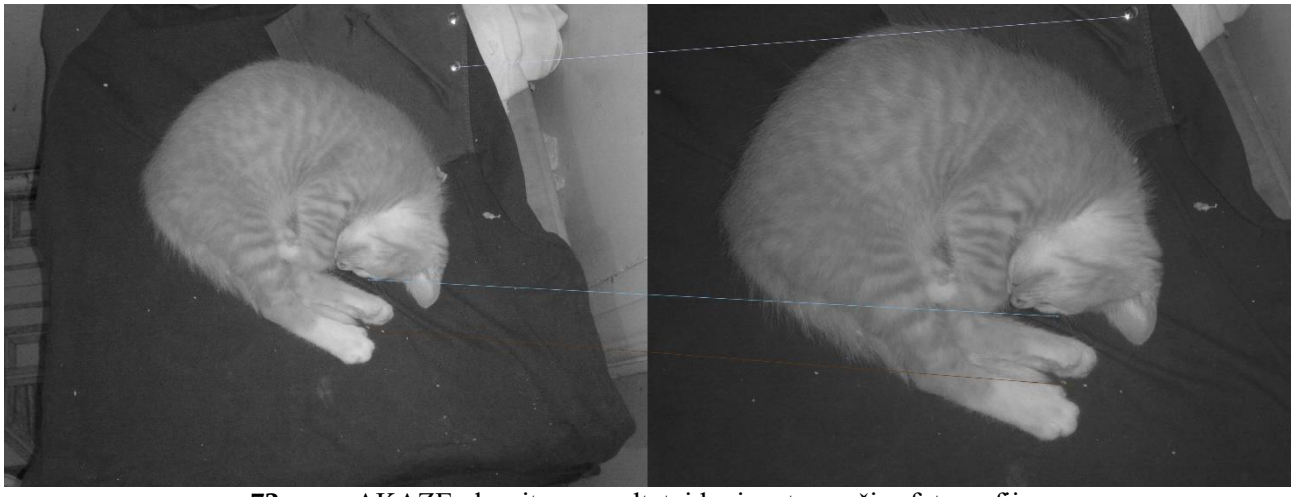
KAZE algoritmo skaičiavimai lyginant skirtingos raiškos fotografijas užtruko 777.8 sekundės, skaičiavimų metu buvo rastos 67 sutampančios savybės be to jos visos buvo aptiktos teisingai. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (13 priedas, 13.5 pav.).

4.2.13 AKAZE detektoriaus ir deskriptoriaus tyrimo rezultatai



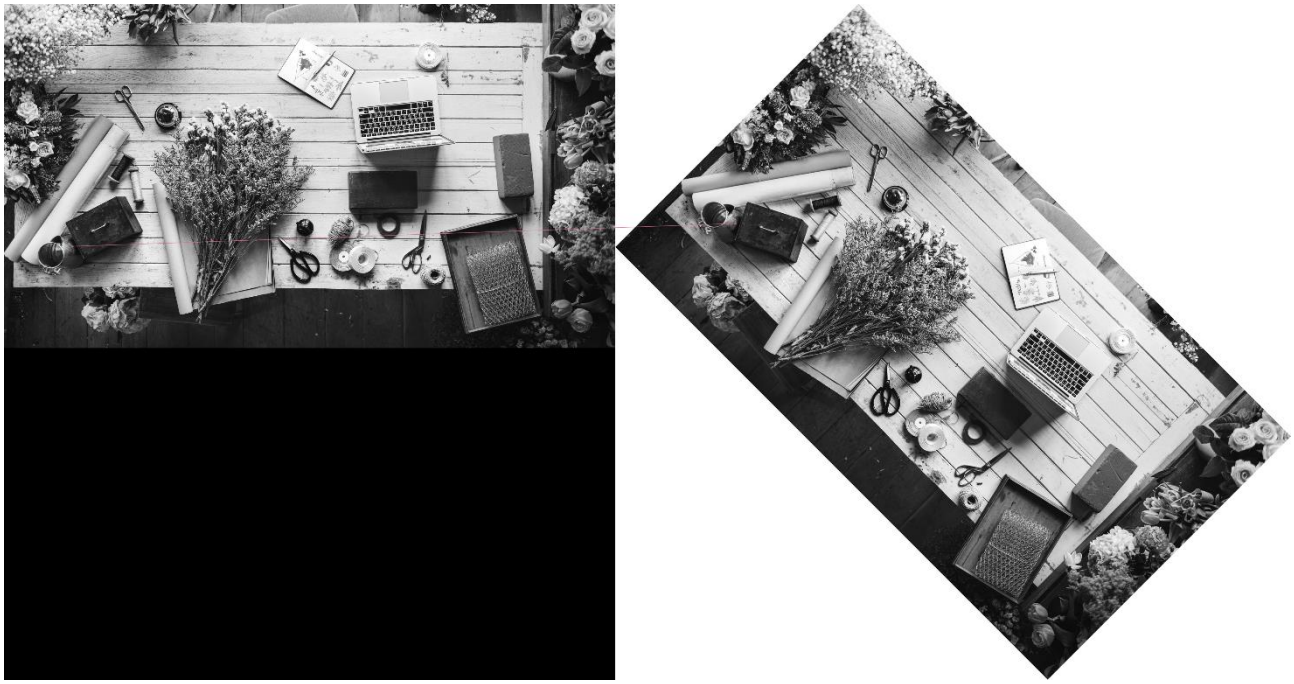
72 pav. AKAZE algoritmo rezultatai lyginant skirtingus paveikslėlius

AKAZE rezultatai buvo šiek tiek geresni nei KAZE, skaičiavimai vidutiniškai užtruko apie 40 sekundžių bei buvo rasta 14 sutampančių savybių. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (14 priedas, 14.1 pav.).



73 pav. AKAZE algoritmo rezultatai lyginant panašias fotografijas

AKAZE algoritmas panašiuose paveikslėliuose rado 3 teisingai sutampančias savybes. Skaičiavimai truko šiek tiek mažiau nei 40 sekundžių. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (14 priedas, 14.2 pav.).



74 pav. AKAZE algoritmo rezultatai lyginant pasuktas fotografijas

Lyginant pasuktas fotografijas AKAZE iš viso aptiko tik vieną sutampančią savybę, ji buvo teisinga, tačiau skaičiavimai užtruko ilgiausiai- net 337,6 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (14 priedas, 14.3 pav.).

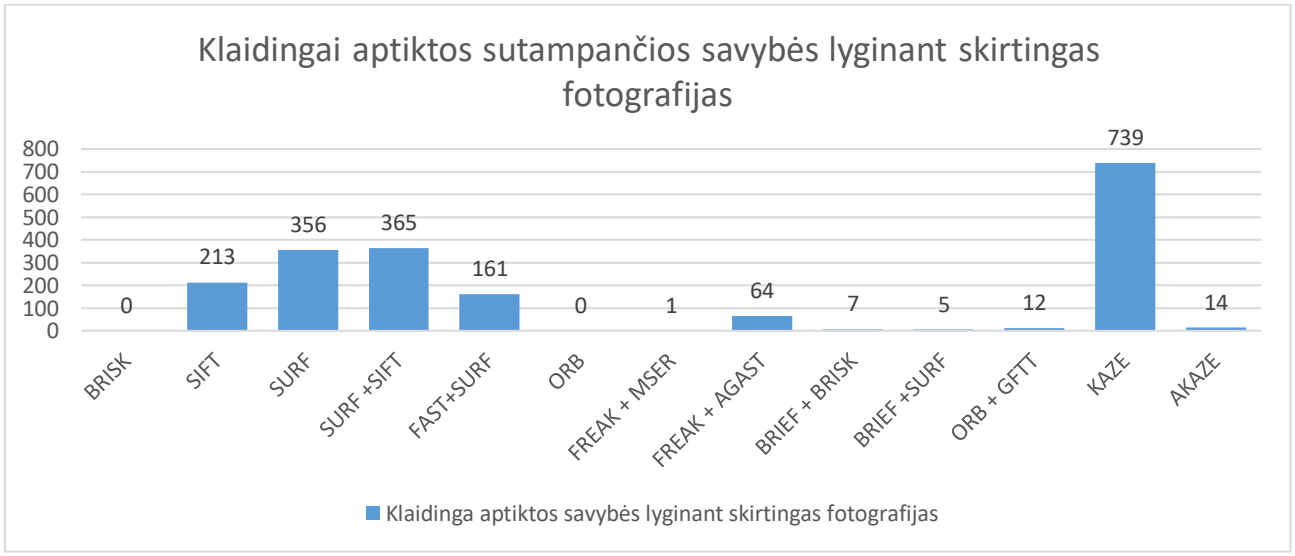
75 pav. AKAZE algoritmo rezultatai lyginant fotografijas su sulietu vaizdu

AKAZE algoritmas rado tik vieną sutampančią savybę, o skaičiavimai užtruko 165.6 sekundės. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (14 priedas, 14.4 pav.).

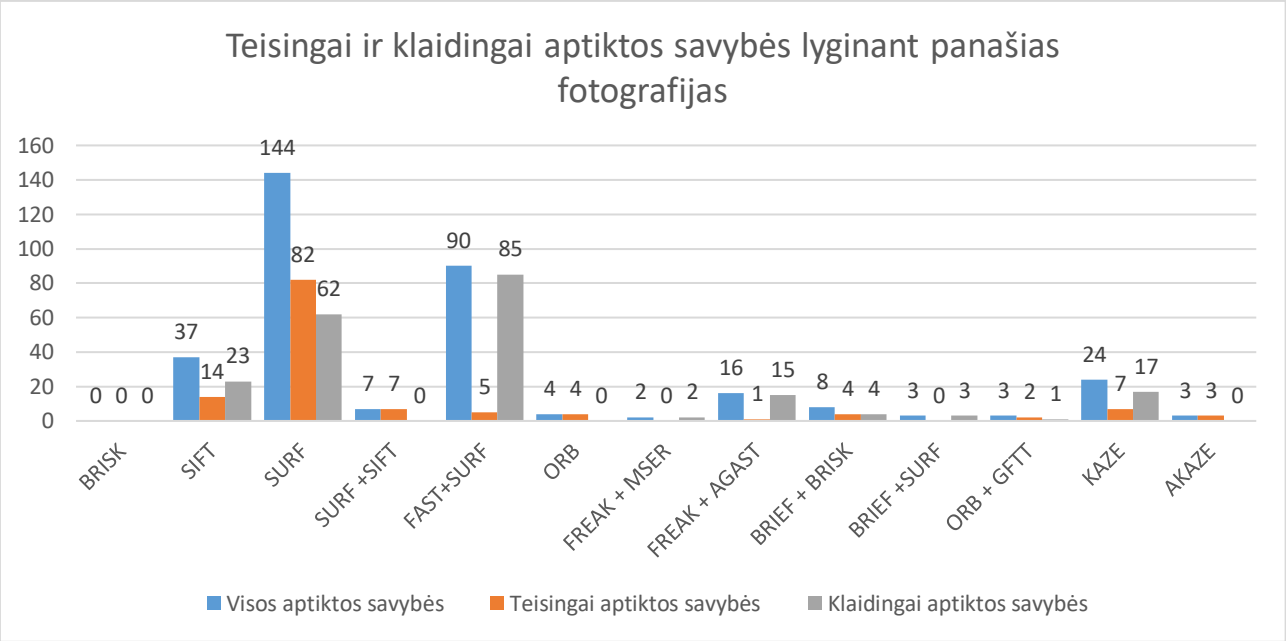
76 pav. AKAZE detektoriaus ir deskriptoriaus rezultatai lyginant skirtingos raiškos fotografijas

AKAZE algoritmo skaičiavimai užtruko ilgiausiai, net 1037,4 sekundės be to iš 5 rastų savybių tik 2 buvo teisingos. Detalesni greitaveikos tyrimo rezultatai pateikiami grafike (14 priedas, 14.5 pav.).

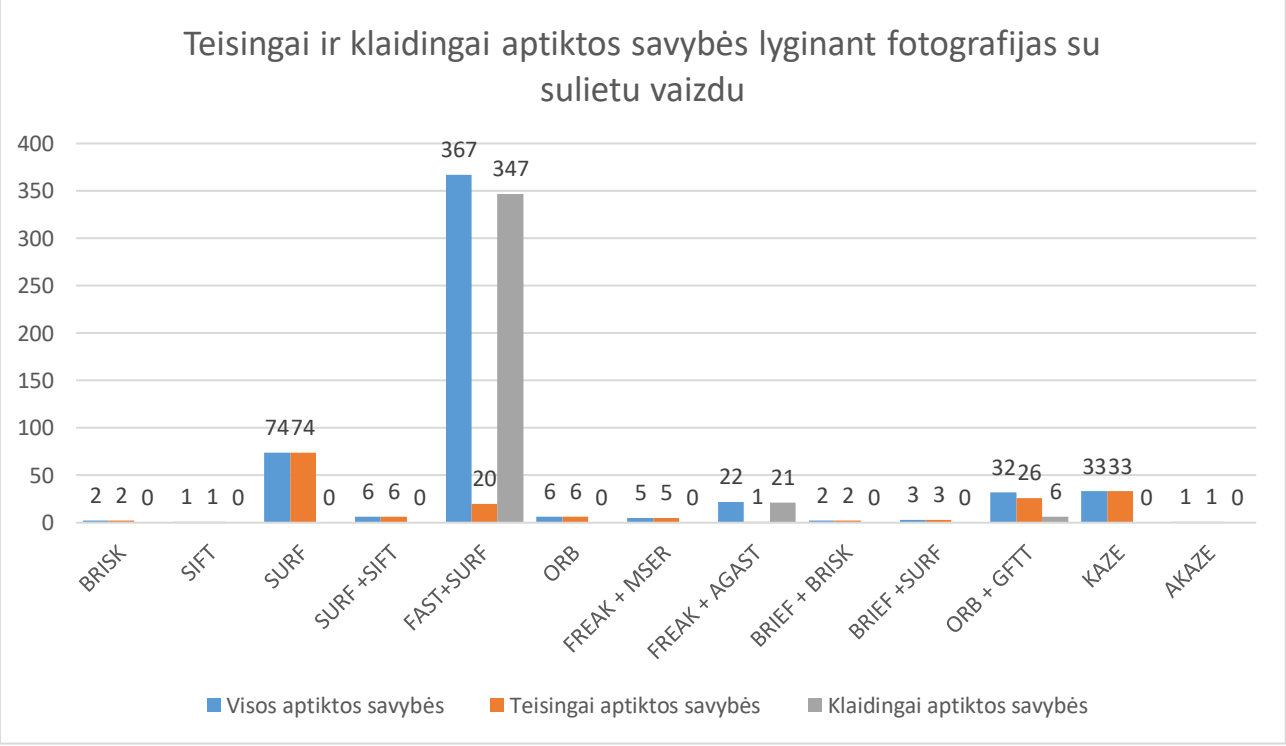
4.3. Rezultatų apibendrinimas



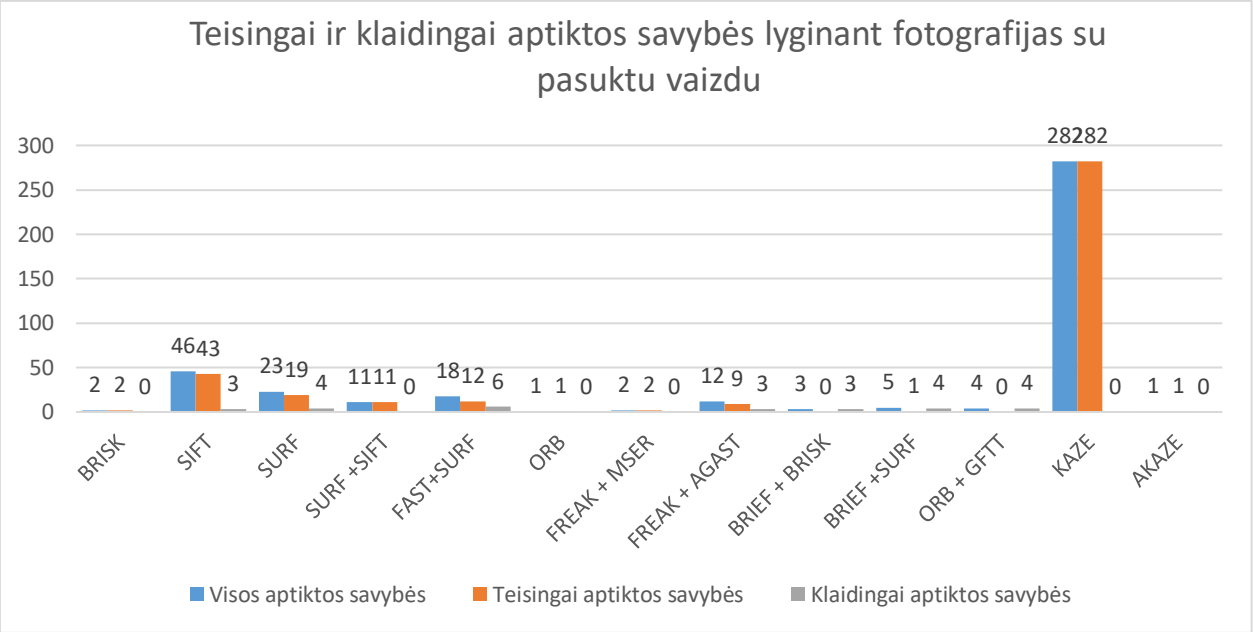
Lyginant skirtingas fotografijas daugiausiai klaidingai sutampančių savybių aptiko KAZE algoritmas (739), SURF algoritmas (356), SIFT detektorius ir SURF deskriptorius (365) bei SIFT algoritmas (213). Jokių sutampančių savybių skirtinguose paveikslėliuose neaptiko tik BRISK ir ORB algoritmai taip pat jie greičiausiai atliko visus skaičiavimus.



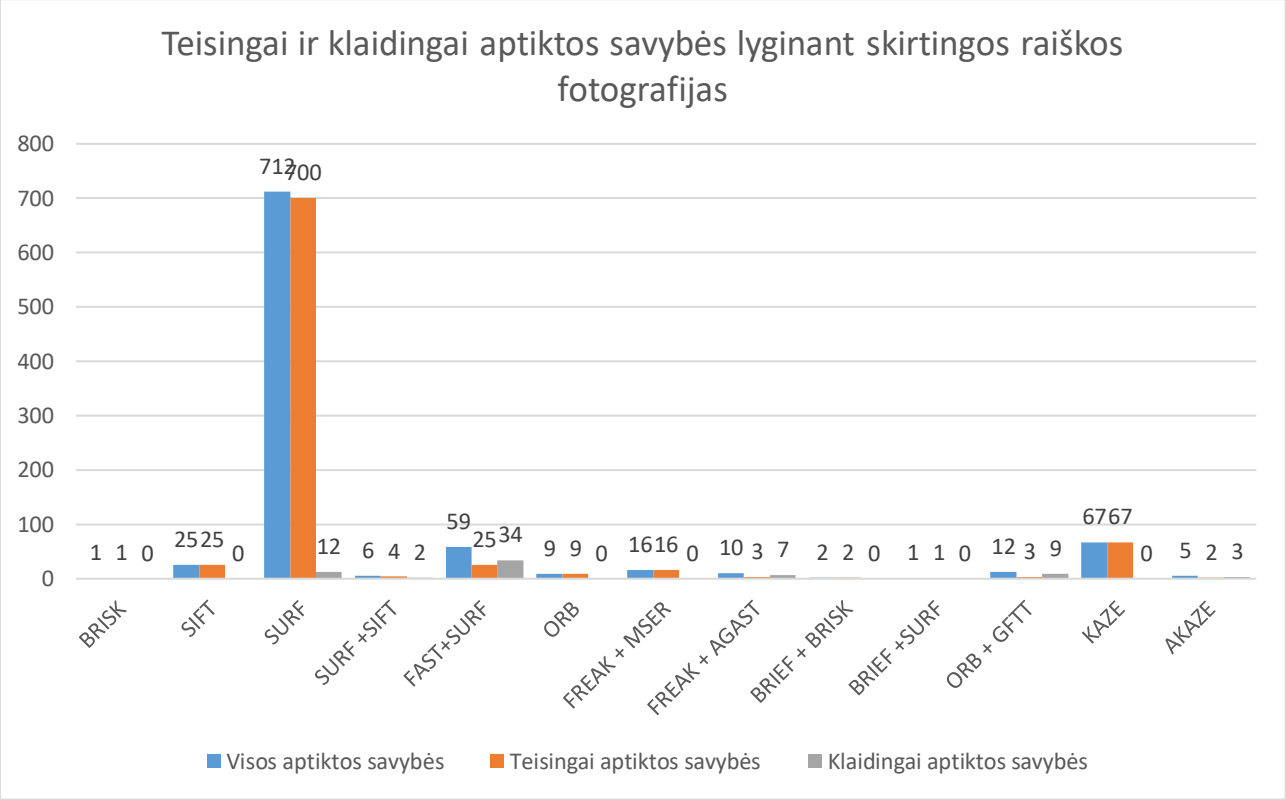
Panašių fotografijų palyginime tiksliausias buvo SURF detektorius ir SIFT deskriptorius bei ORB ir AKAZE algoritmai, tačiau ORB skaičiavimus atliko 20 kartų greičiau nei SURF detektorius ir SIFT deskriptorius ir beveik 80 kartų greičiau nei AKAZE



Didžiausiu atsparumu sulietam vaizdui pasižymėjo SURF, KAZE ir ORB algoritmai, o prasčiausias rezultatas buvo pasiektas FAST detektoriaus ir SURF deskriptoriaus bei FREAK detektoriaus ir AGAST deskriptoriaus, kurie beveik visas savybes atliko klaidingai.



Pasukimui pats atspariausias buvo KAZE algoritmas, o mažiausiai atsparus - ORB detektorius ir GFTT deskriptorius kurie visas sutampančias savybes aptiko klaidingai.



Skirtingos raiškos fotografijose geriausiai savybes aptiko KAZE algoritmas, tačiau dėl skaičiavimų laiko, kuris buvo daugiau nei 1000 sekundžių, šį algoritmą būtų labai sudėtinga pritaikyti aukštos raiškos fotografijų palyginime.

3 lentelė. Visų eksperimentų metu aptiktos teisingos ir klaidingos savybės

Detektorius	Deskriptorius	Visuose eksperimentuose teisingai aptiktų savybių suma	Visuose eksperimentuose klaidingai aptiktų savybių suma	Teisingas savybių aptikimas, %
BRISK	BRISK	5	0	100,0
SIFT	SIFT	322	239	25,8
SURF	SURF	1333	434	67,4
SURF	SIFT	395	367	7,1
FAST	SURF	695	633	8,9
ORB	ORB	20	0	100,0
FREAK	MSER	26	3	88,5
FREAK	AGAST	124	110	11,3
BRIEF	BRISK	22	14	36,4
BRIEF	SURF	17	12	29,4
ORB	GFTT	63	32	49,2
KAZE	KAZE	1145	756	34,0
AKAZE	AKAZE	24	17	29,2

3 lentelėje pateikiama kiekvieno algoritmo per visus eksperimentus rastų savybių bei klaidingai rastų savybių sumos. Kaip matome, didžiausiu tikslumu pasižymėjo BRISK ir ORB algoritmai, visos jų aptiktos savybės buvo teisingos, KAZE algoritmas puikiai aptiko savybes sulietuose, pasuktuose bei skirtingos raiškos fotografijos, tačiau jis rado labai daug sutampančių savybių skirtinguose paveikslėliuose, dėl to labai sumažėjo jo tikslumas.

5. IŠVADOS

1. Atlikus vaizdų palyginimo algoritmų ir juos naudojančios programinės įrangos analizę nustatyta, kad dėl greito veikimo ir paprastos realizacijos dažniausiai naudojami pikseliais paremti algoritmai. Tačiau jie yra visiškai neatsparūs bet kokioms fotografijų modifikacijoms.
2. Greitaveikos tyrimo metu nustatyta, kad greičiausiai veikia ORB algoritmas, net lyginant 3 ir 6 megapikselių raiškos fotografijas jo skaičiavimai truko mažiau nei 2 sekundes, kai kitų algoritmų skaičiavimai truko nuo 11 iki 1100 sekundžių. Lėčiausiai skaičiavimus atliko KAZE ir AKAZE algoritmai.
3. Algoritmų tikslumo ir atsparumo modifikacijoms tyrimo metu nustatyta, kad tiksliausiai veikia ORB ir BRISK algoritmai, nors jie surado nedidelį kiekį (1-9) kiekviename eksperimente sutampančių savybių, tačiau visos jų aptiktos sutampančios savybės buvo teisingos.
    - Bandant palyginti skirtingas fotografijas daugiausiai klaidingų savybių aptikę algoritmai buvo: KAZE, kuris surado net 739 sutampančias savybes kurių neturėjo būti, SURF (356 klaidingai sutampančios savybės) bei SIFT (215 klaidingai sutampančių savybių)
    - Lyginant panašias fotografijas su spalvų ir mastelio pokyčiais tiksliausi buvo SURF detektoriaus ir SIFT deskriptoriaus kombinacija bei ORB algoritmas. Prasčiausius rezultatus parodė FAST detektoriaus ir SURF deskriptoriaus kombinacija bei FREAK detektoriaus ir AGAST deskritoriaus kombinacija, daugiau nei 90% jų aptiktų sutampančių savybių buvo klaidingos
    - Lyginant fotografijas su pasuktu vaizdu tiksliausias buvo KAZE algoritmas, jis aptiko 282 teisingai sutampančias savybes, tačiau skaičiavimai truko net 286 sekundes. Mažiausiai tikslūs šioje eksperimentų dalyje buvo BRIEF detektorius ir BRISK deskriptorius bei ORB detektorius ir GFTT deskriptorius - visos jų surastos sutampančios savybės buvo klaidingos.
    - Tiriant algoritmų atsparumą vaizdo suliejimui buvo nustatyta, kad jam atspariausi buvo SURF ir KAZE algoritmai, tačiau KAZE skaičiavimai truko beveik 4 kartus ilgiau nei SURF.
    - Bandant palyginti skirtingos raiškos fotografijas daugiausia sutampančių savybių rado SURF algoritmas, tačiau dalis iš jų buvo klaidingos. Atspariausi raiškos pokyčiui buvo SIFT, ORB ir KAZE algoritmai, kurių visos sutampančios savybės buvo teisingos.

6. LITERATŪRA

[1] „Image Comparer Duplicate Photo Finder and Remover,“ [Tinkle]. Available: http://www.bolidesoft.com/imagecomparer.html. [Kreiptasi 08 12 2015].

[2] „Resemble.js,“ [Tinkle]. Available: https://huddle.github.io/Resemble.js/. [Kreiptasi 20 12 2015].

[3] „Anti-Twin (Freeware) - Find and delete duplicate files or similar images,“ [Tinkle]. Available: http://www.anti-twin.com/. [Kreiptasi 12 01 2016].

[4] „An Image Comparison Program,“ [Tinkle]. Available: http://www.flounder.com/imagecomparator.htm. [Kreiptasi 20 12 2015].

[5] „DiffImg The Hive xbee tools,“ [Tinkle]. Available: http://thehive.xbee.net/index.php?module=pages&func=display&pageid=11. [Kreiptasi 12 01 2016].

[6] „Details about the algorithm for image comparison,“ [Tinkle]. Available: https://www.qfs.de/qftest/manual/en/tech_imagealgorithmdetails.html. [Kreiptasi 01 12 2015].

[7] M. Guerrero, „A Comparative Study of Three Image Matcing,“ 2011.

[8] t. T. L. V. G. Herbart Bay, „SUF: Speeded Up Robust Features“.

[9] J. Rihan, „An Exploration of the SIFT Operator,“ 2005.

[10] E. R. a. T. Drummond, „Machine learning for high-speed corner detection,“ *European Conference on Computer Vision,* 2006.

[11] E. R. a. T. Drummond, „Fusing points and lines for high performance tracking,“ *IEEE International Conference on Computer Vision,* 2005.

[12] „BRIEF (Binary Robust Independent Elementary Features),“ [Tinkle]. Available: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_brief/py_brief.html. [Kreiptasi 05 01 2017].

[13] M. L. V. S. C. &. F. P. Calonder, „Brief: Binary robust independent elementary features.,“ įtraukta *European conference on computer vision*, Springer Berlin Heidelberg., 2010.

[14] A. V. J. S. J. a. V. K. H. Kulkarni, „Object recognition with ORB and its Implementation on FPGA,“ *International Journal of Advanced Computer Research 3.3,* 2013.

[15] A. M. T.-A. J. C. P.-O. a. E. A. R.-A. Takacs, „"Dedicated feature descriptor for outdoor augmented reality detection.",“ *Pattern Analysis and Applications,* pp. 1-12, 2016.

[16] D. L.-H. W. P. Forss, „Region detectors“.

[17] G. L. J. Oubong, „A Comparison of SIFT, PCA-SIFT amd SURF,“ *international Juornal of Image Processing.*

[18] „Full HD 1080p 3D Wallpapers, Desktop Backgrounds HD, Pictures and Images,“ [Tinkle]. Available: https://wallpaperscraft.com/catalog/3d/1920x1080.

[19] M. S. a. K.Hemachandran, „Content Based Image Retrieval using Color and Texture,“ *Signal & Image Processing : An International Journal,* t. 3, 2012.

[20] „How to get Color Histogram of an Image,“ [Tinkle]. Available: http://se.mathworks.com/matlabcentral/answers/181624-how-to-get-color-histogram-of-an-image. [Kreiptasi 03 06 2016].

[21] „MATLAB,“ [Tinkle]. Available: https://en.wikipedia.org/wiki/MATLAB. [Kreiptasi 02 06 2016].

[22] „Python Tutorial - Image Histogram - 2016,“ [Tinkle]. Available: http://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_image_histogram_calcHist.php. [Kreiptasi 08 06 2016].

[23] „Python (programming language),“ [Tinkle]. Available: https://en.wikipedia.org/wiki/Python_(programming_language). [Kreiptasi 20 05 2016].

[24] „What is C++ Programming Language - Definition from Techopedia,“ [Tinkle]. Available: https://www.techopedia.com/definition/26184/c-programming-language. [Kreiptasi 15 05 2016].

[25] „Histogram Calculation — OpenCV 2.4.13.0 documentation,“ [Tinkle]. Available: http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_calculation/histogram_calculation.html. [Kreiptasi 20 05 2016].

[26] „Displaying a histogram of image data,“ [Tinkle]. Available: http://stackoverflow.com/questions/28519355/displaying-a-histogram-of-image-data. [Kreiptasi 15 05 2016].

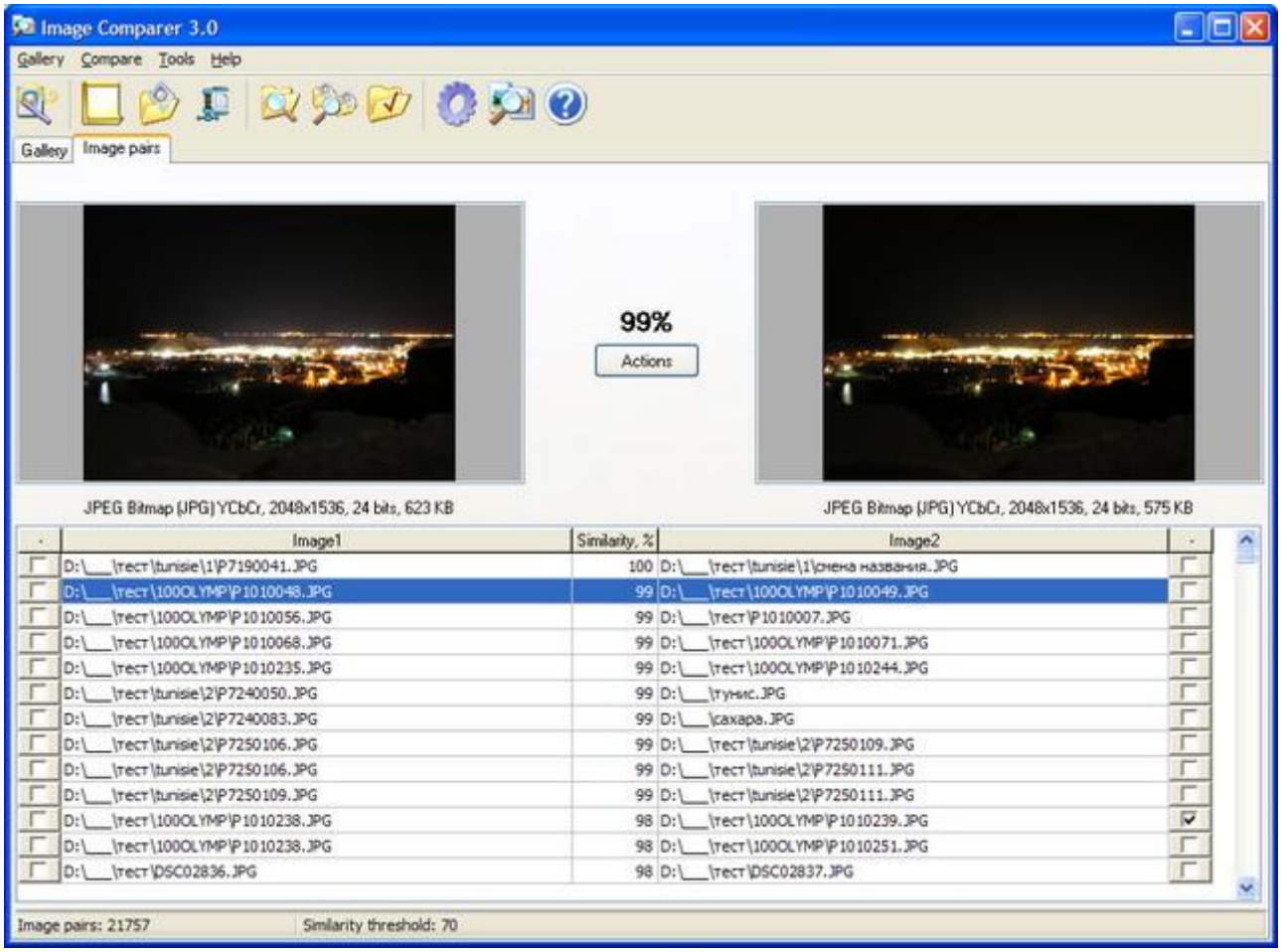
[27] „The Java Programming Language,“ [Tinkle]. Available: http://groups.engin.umd.umich.edu/CIS/course.des/cis400/java/java.html. [Kreiptasi 15 06 2016].

[28] L. J. a. O. Gwun, „A comparison of SIFT, PCA-SIFT and surf,“ *International juornal of image processing,* t. 3, nr. 4, pp. 143-152.

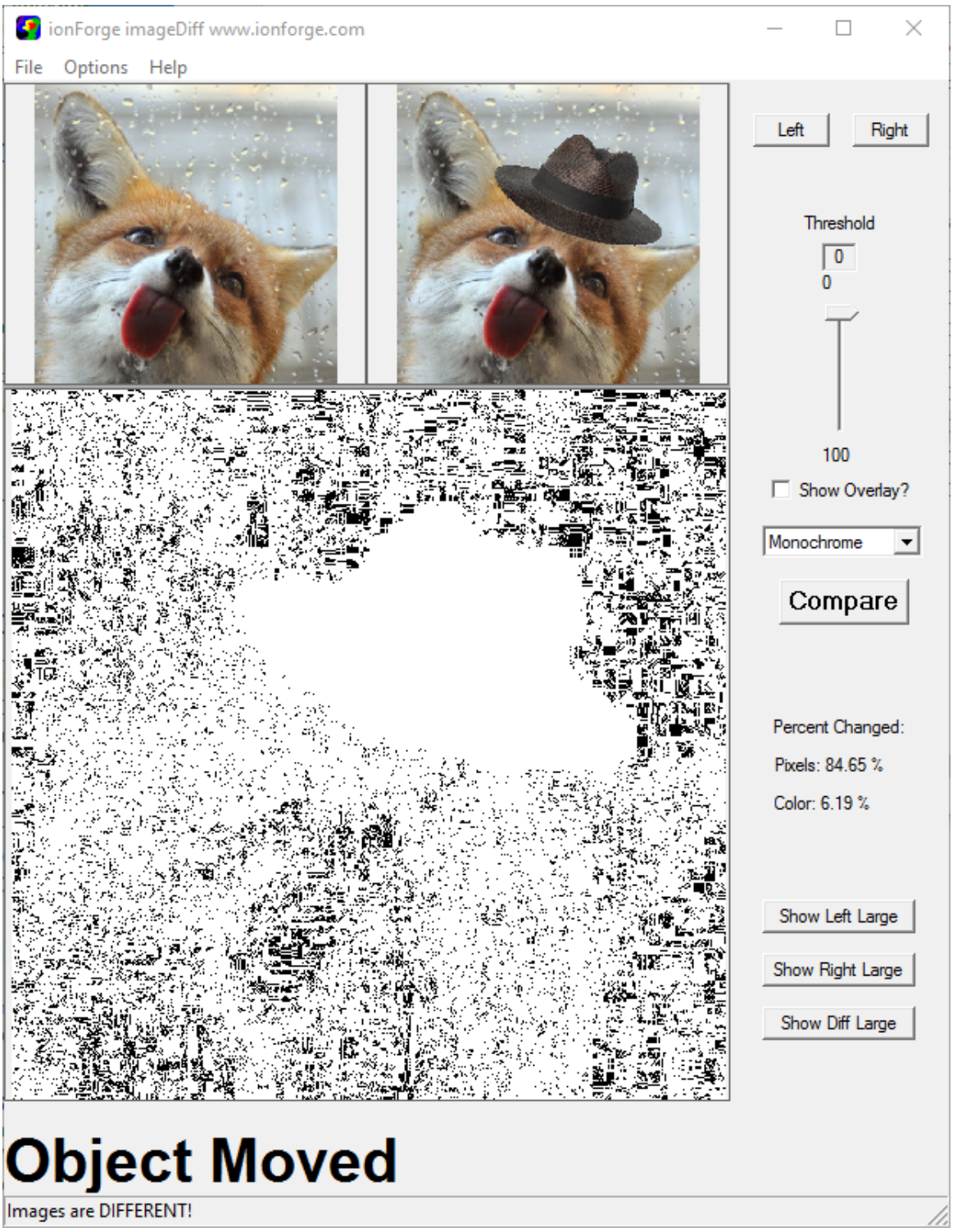
7. PRIEDAI

Priedų sąrašas

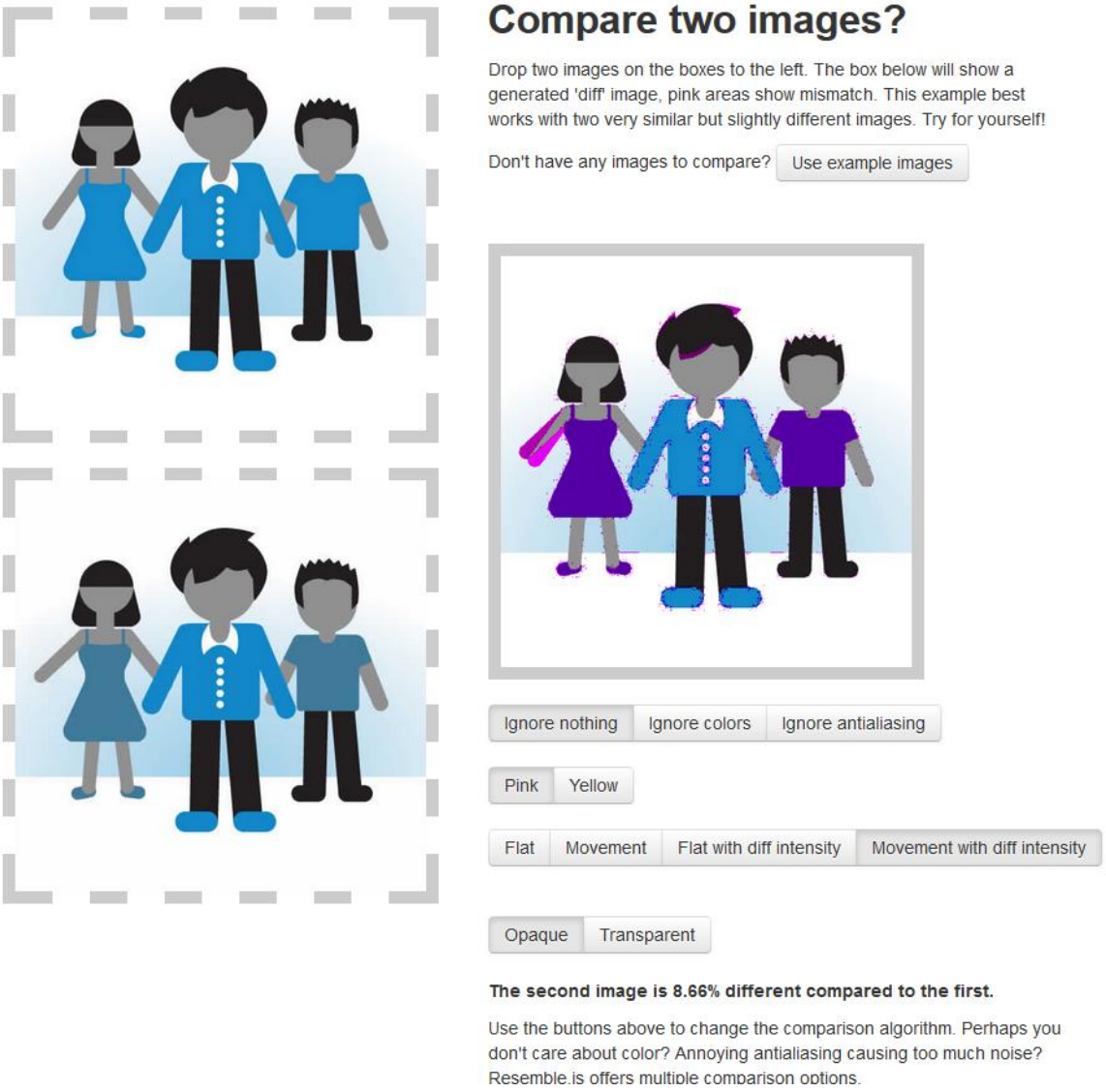
1. Priedas. Fotografijų palyginimo programų ekranvaizdžiai ir rezultatai



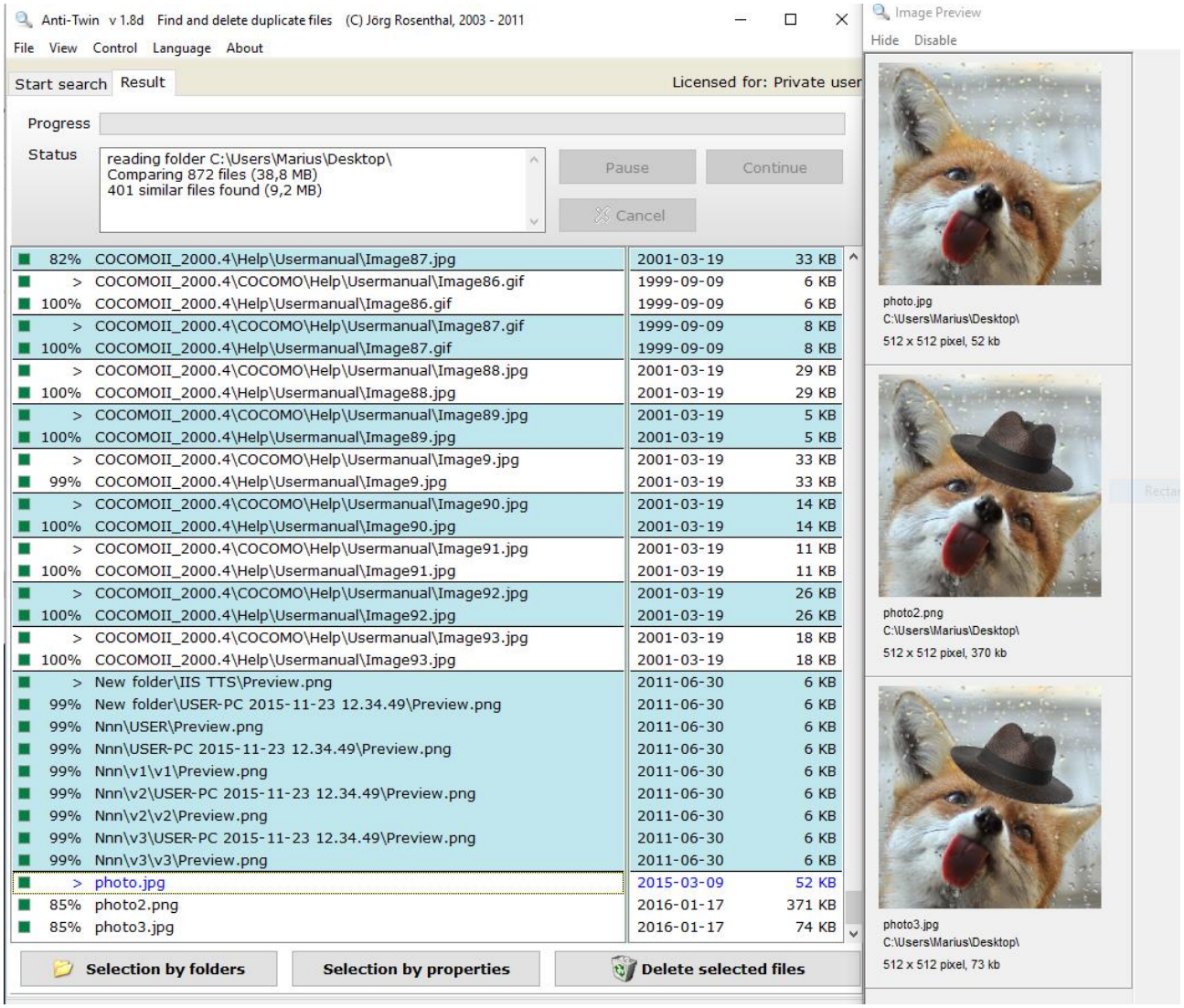
1.1 pav. *Image comparer* programos darbo rezultatai **[4]**



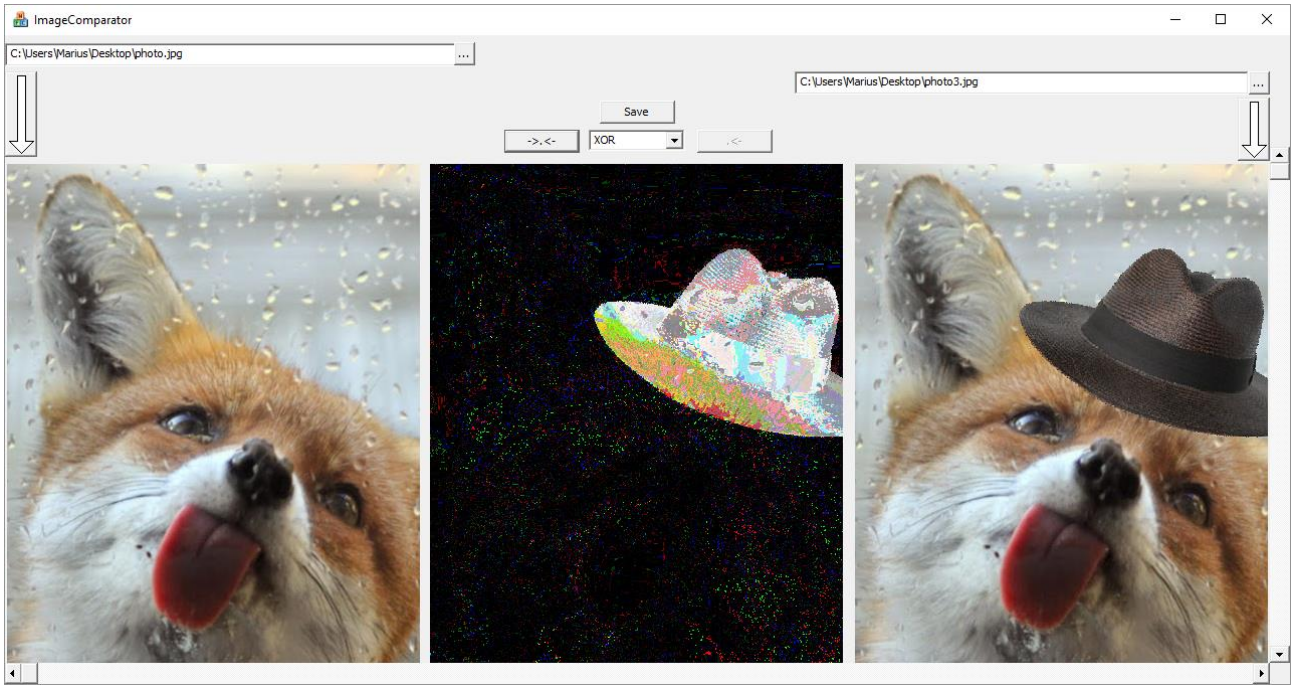
1.2 pav. *ionForge ImageDiff* programos rezultatai



1.3 pav. Resemble.js rezultatai **[2]**

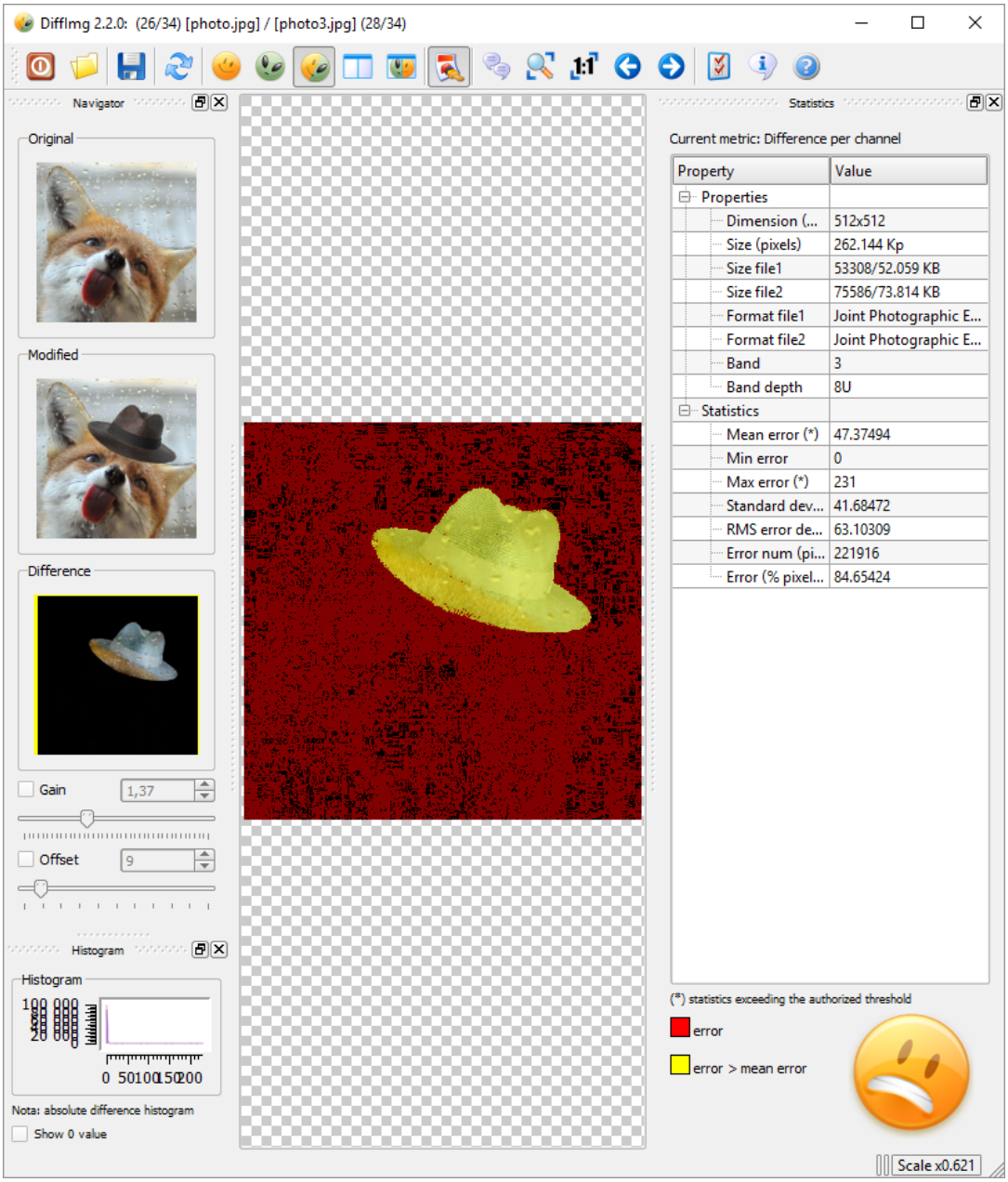


1.4 pav. *Anti-Twin* programos rezultatai



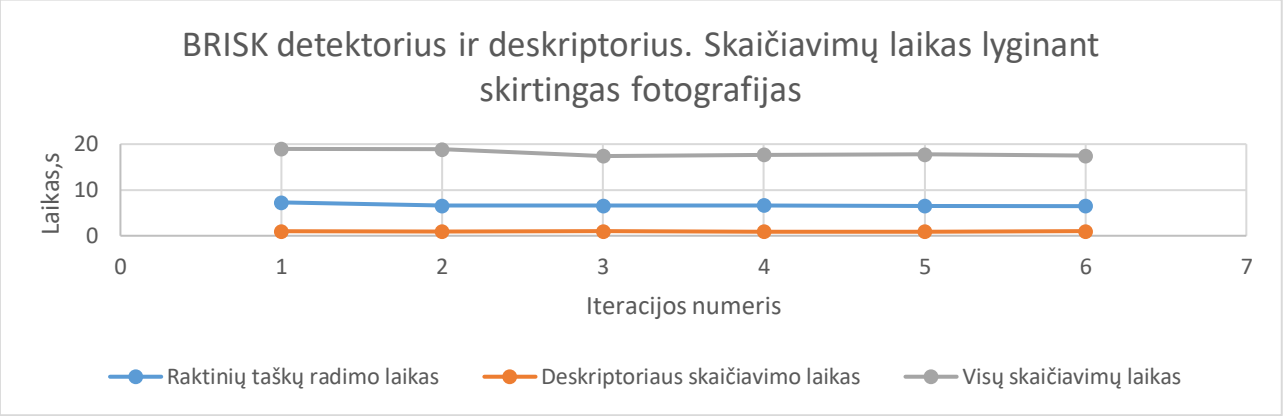
1.5 pav. FlounderCraft Image comparator rezultatai

1.6 pav. Image comparer rezultatai

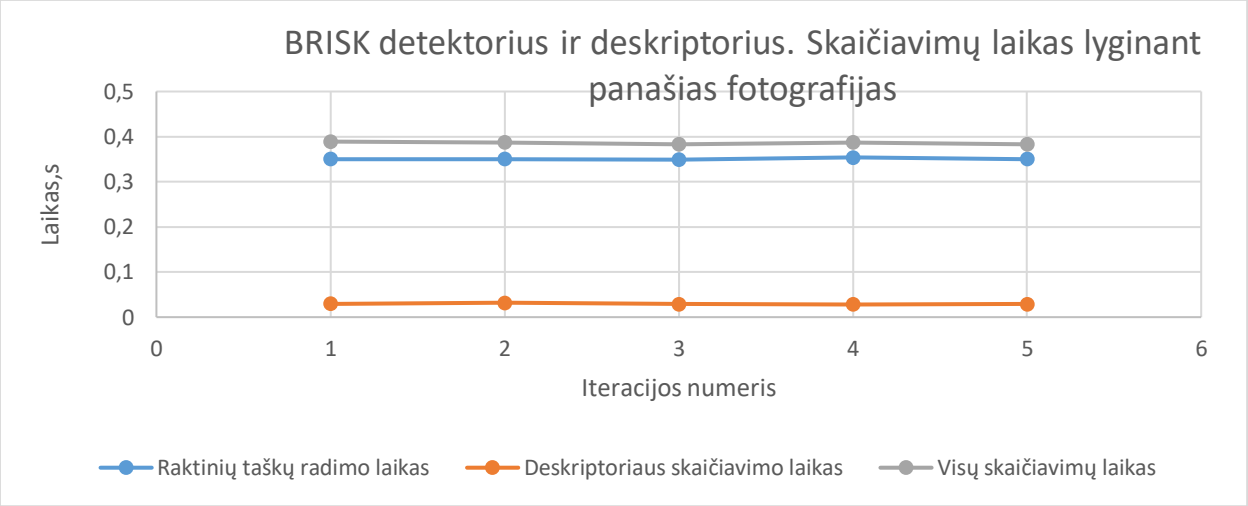


1.7 pav. *DiffImg* rezultatai

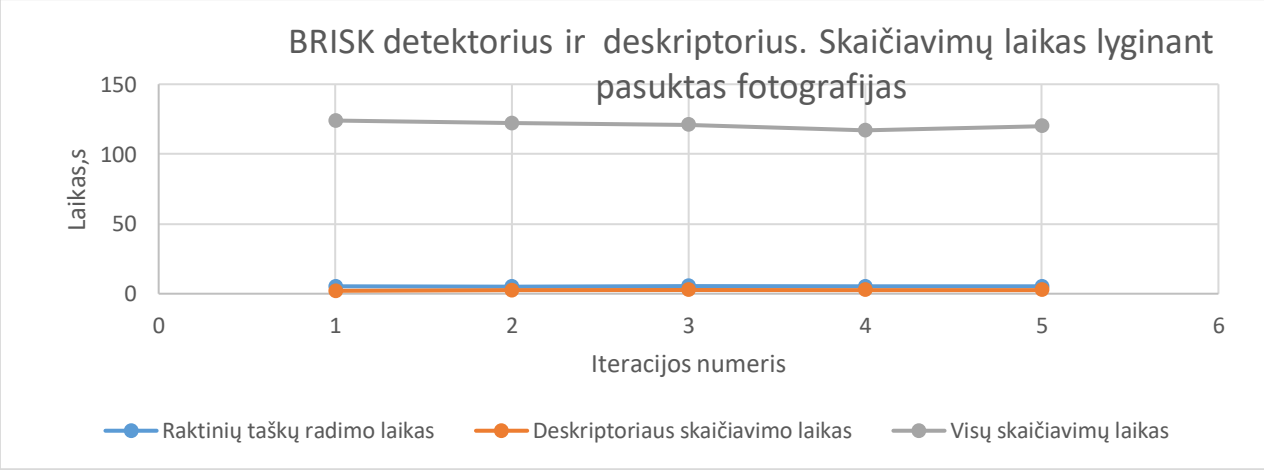
2. Priedas. BRISK algoritmo greitaveikos tyrimo rezultatai



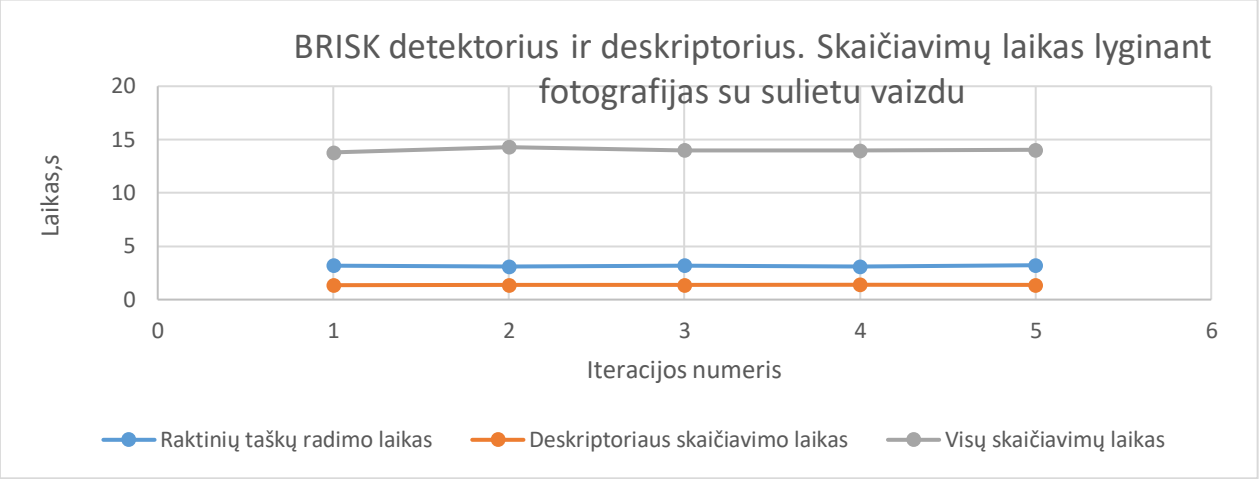
2.1 pav.



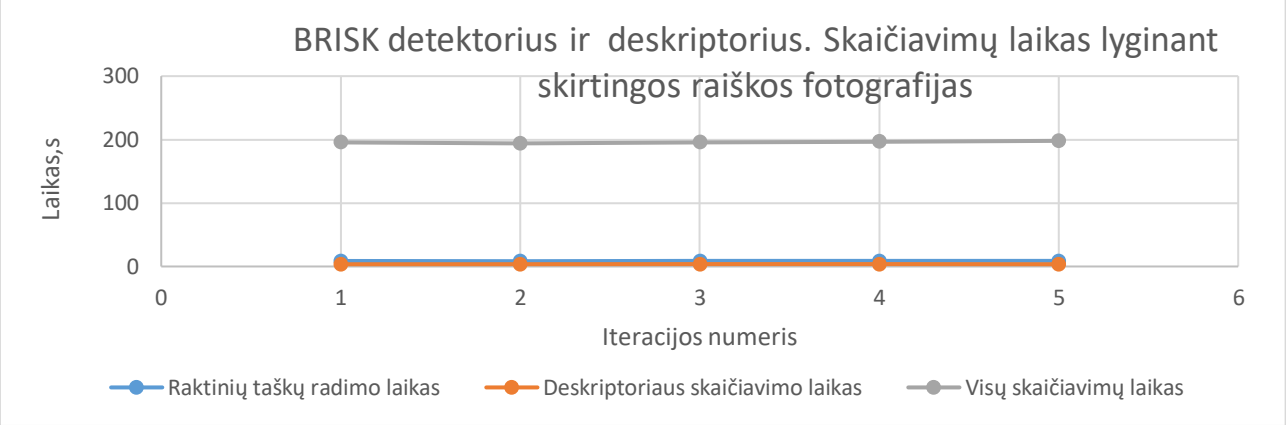
2.2 pav.



2.3 pav.

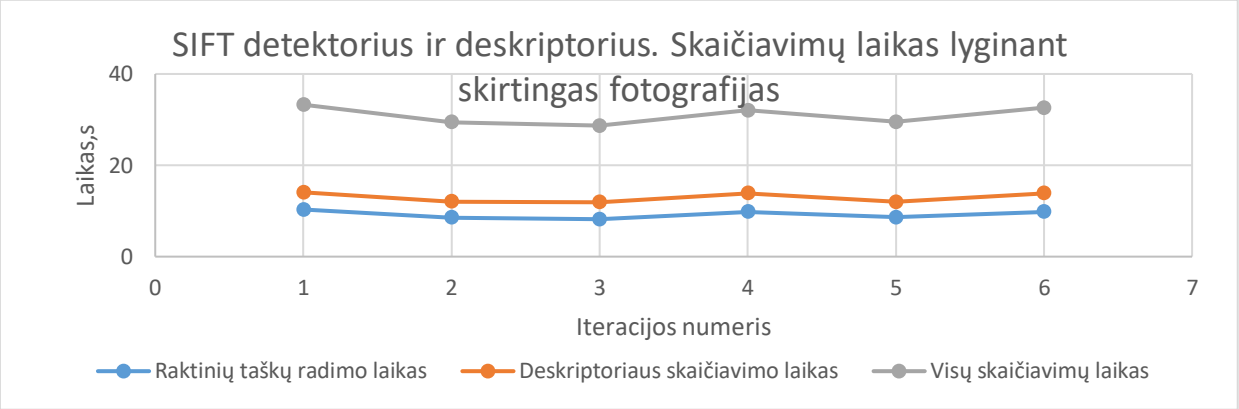

BRISK detektorius ir deskriptorius. Skaičiavimų laikas lyginant fotografijas su sulietu vaizdu
Laikas,s
20
15
10
5
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

2.4 pav.

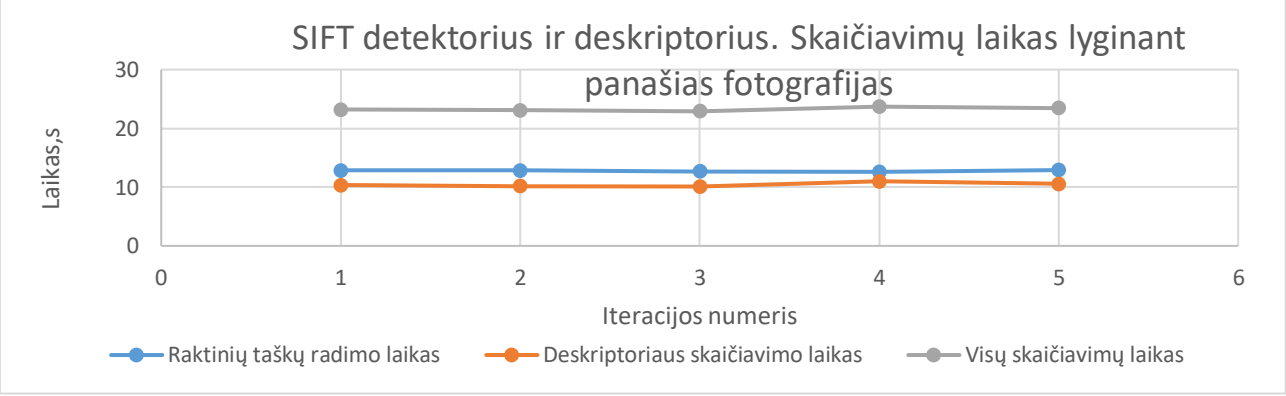

BRISK detektorius ir deskriptorius. Skaičiavimų laikas lyginant skirtingos raiškos fotografijas
Laikas,s
300
200
100
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

2.5 pav.

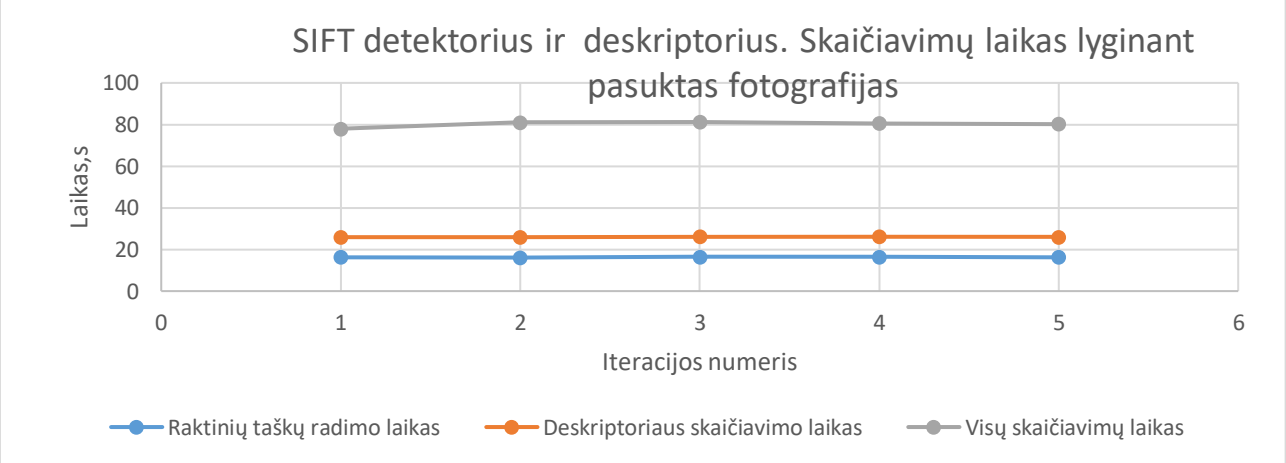
3. Priedas. SIFT algoritmo greitaveikos tyrimo rezultatai



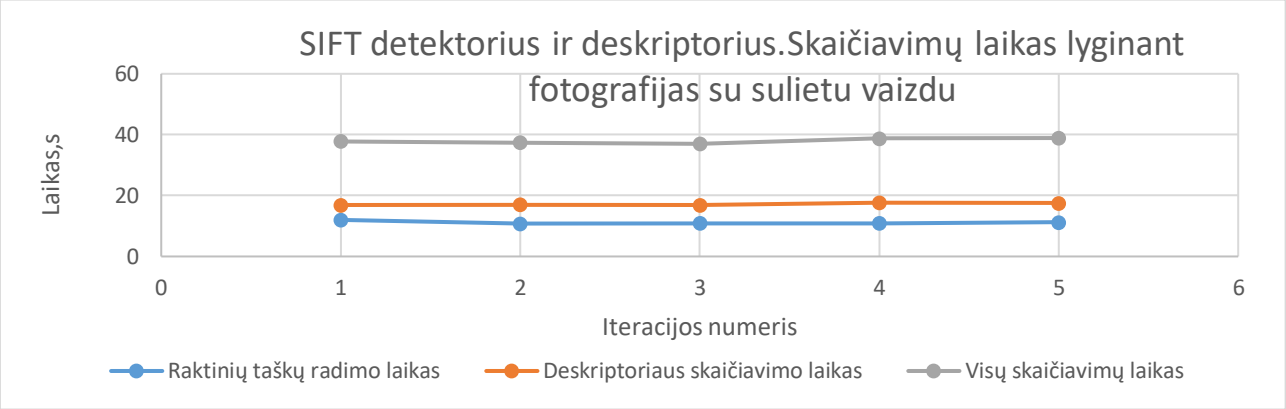
3.1 pav.



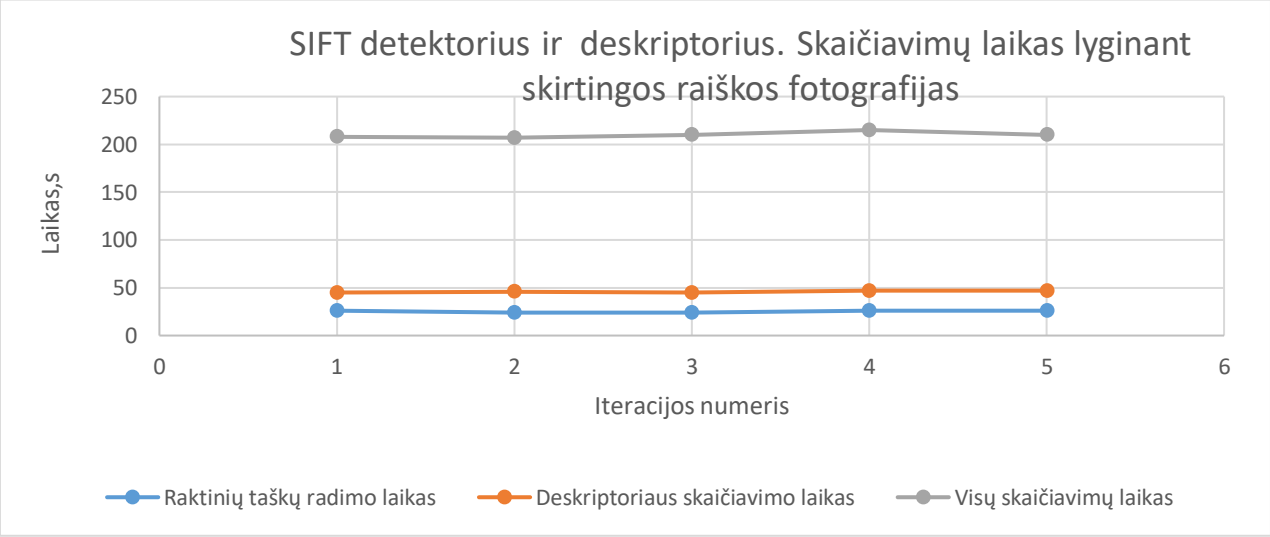
3.2 pav.



3.3 pav.

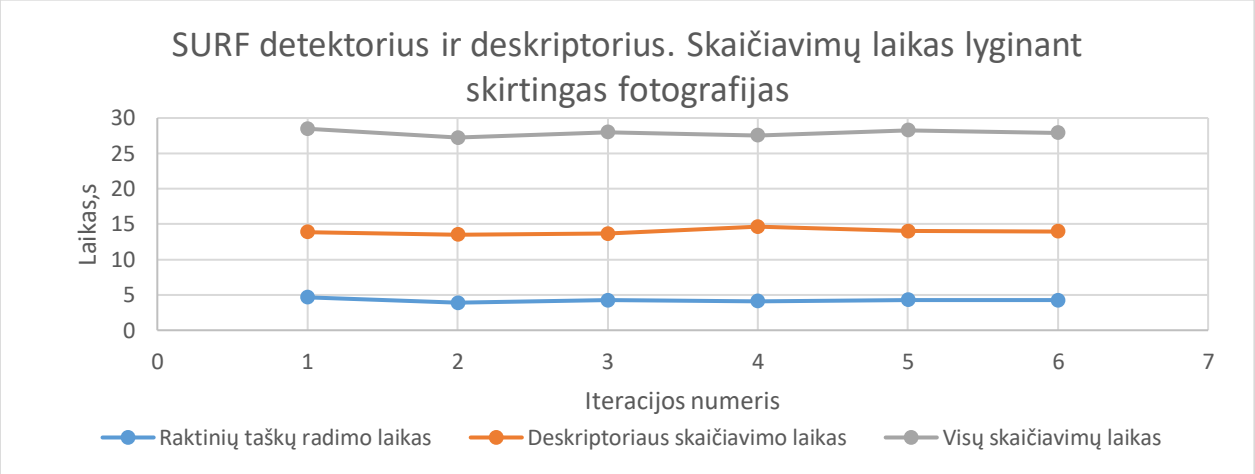

SIFT detektorius ir deskriptorius.Skaičiavimų laikas lyginant fotografijas su sulietu vaizdu
Laikas,s
60
40
20
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

3.4 pav.

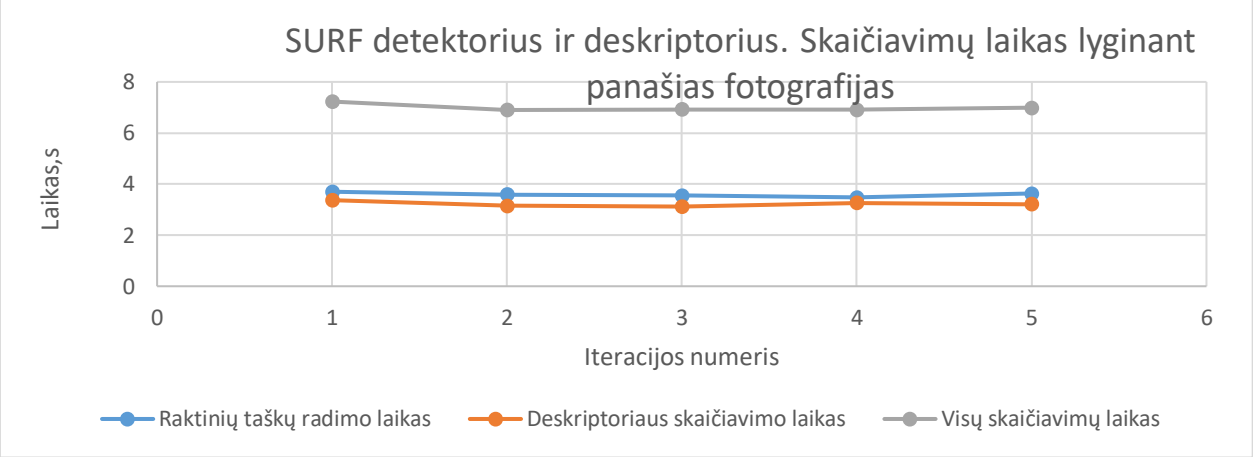

SIFT detektorius ir deskriptorius. Skaičiavimų laikas lyginant skirtingos raiškos fotografijas
Laikas,s
250
200
150
100
50
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

3.5 pav.

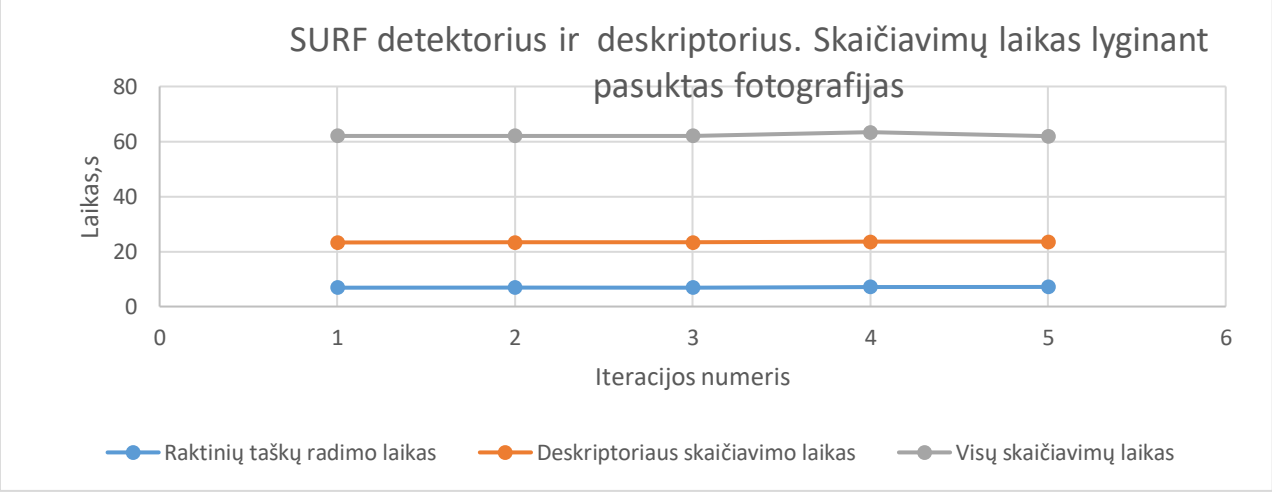
4. Priedas. SURF algoritmo greitaveikos tyrimo rezultatai



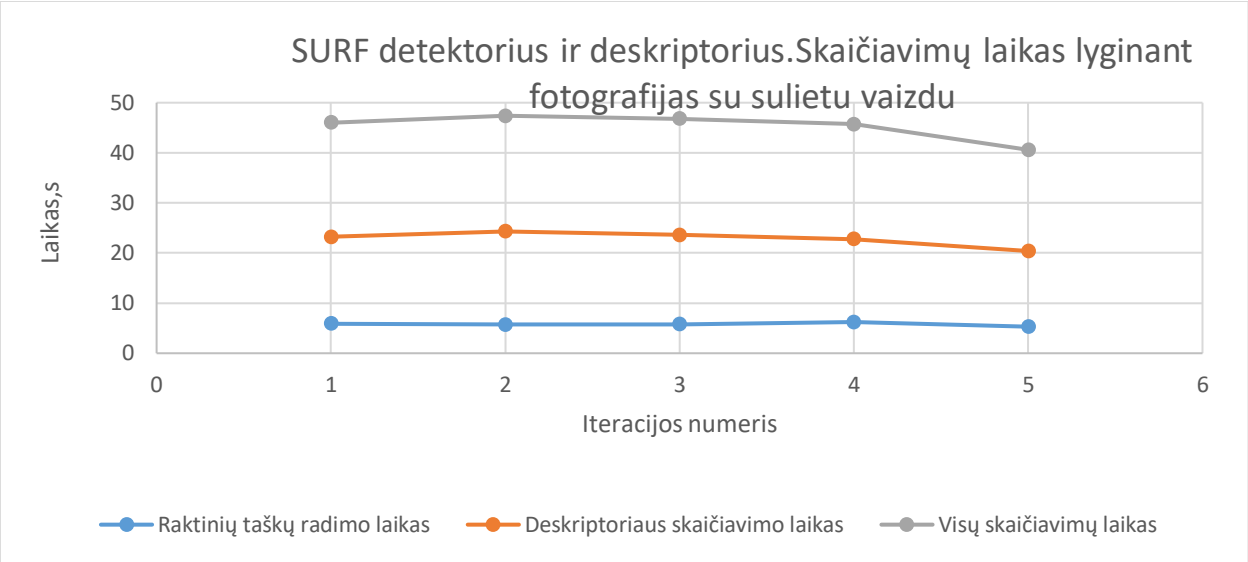
4.1 pav.



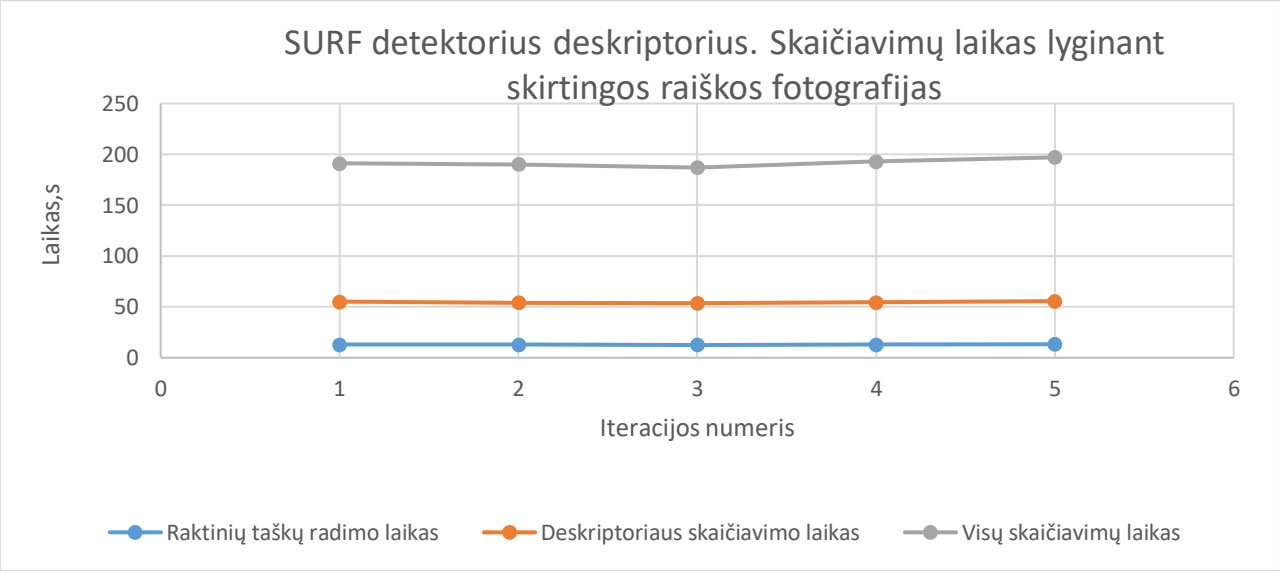
4.2 pav.



4.3 pav.

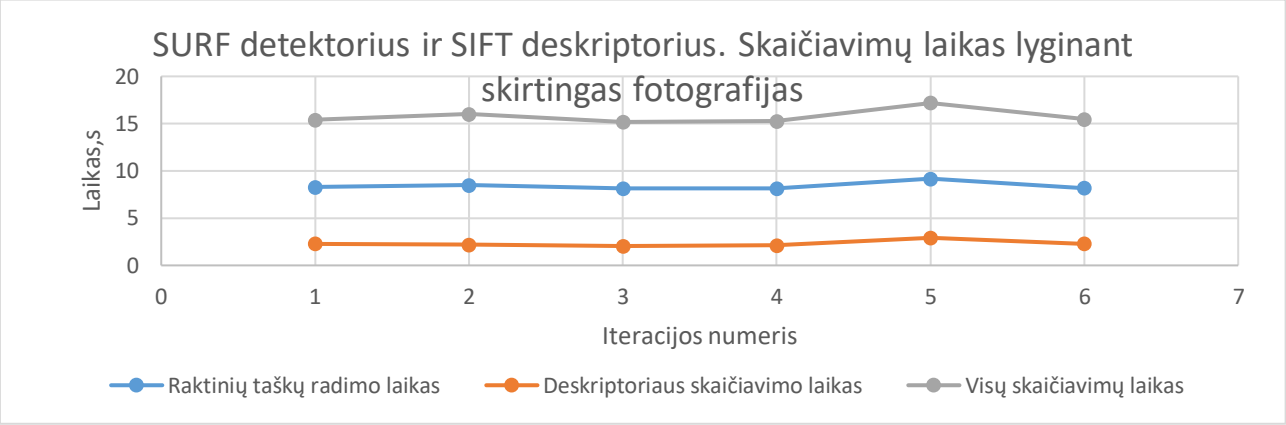

SURF detektorius ir deskriptorius.Skaičiavimų laikas lyginant fotografijas su sulietu vaizdu
Laikas,s
50
40
30
20
10
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

4.4 pav.

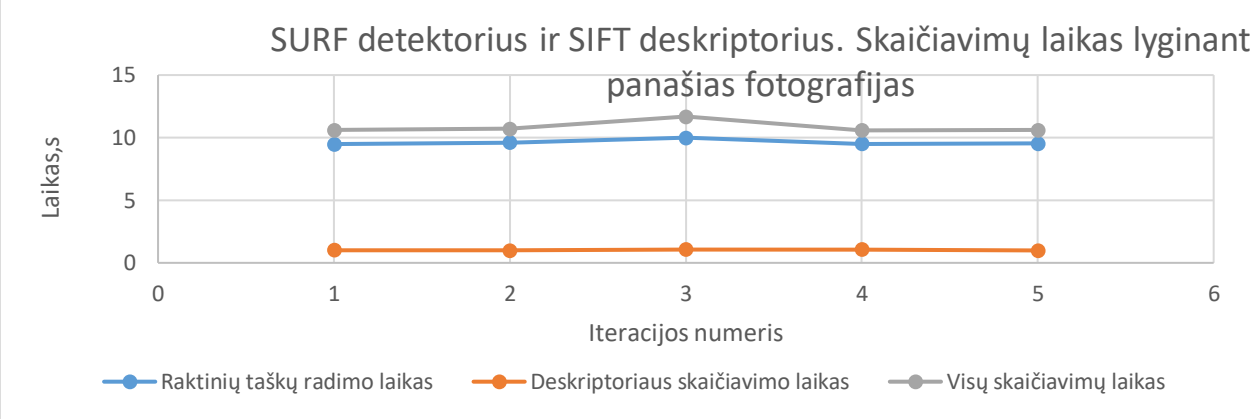

SURF detektorius deskriptorius. Skaičiavimų laikas lyginant skirtingos raiškos fotografijas
Laikas,s
250
200
150
100
50
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

4.5 pav.

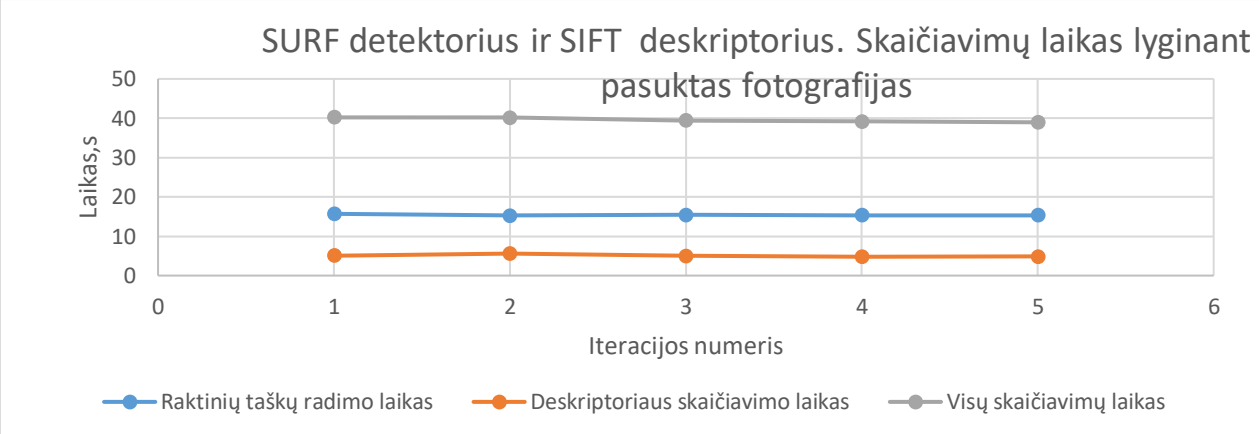
5. Priedas. SURF detektoriaus ir SIFT deskriptoriaus greitaveikos tyrimo rezultatai



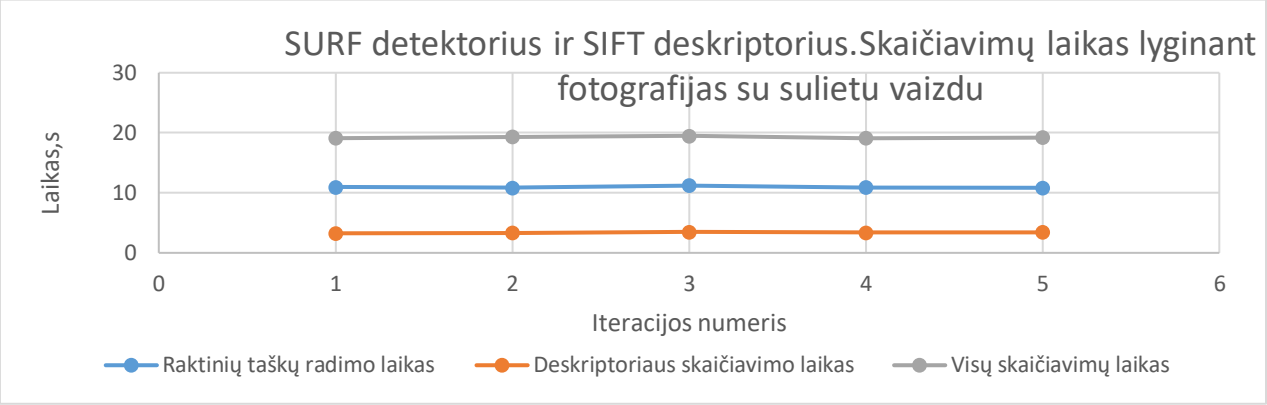
5.1 pav.



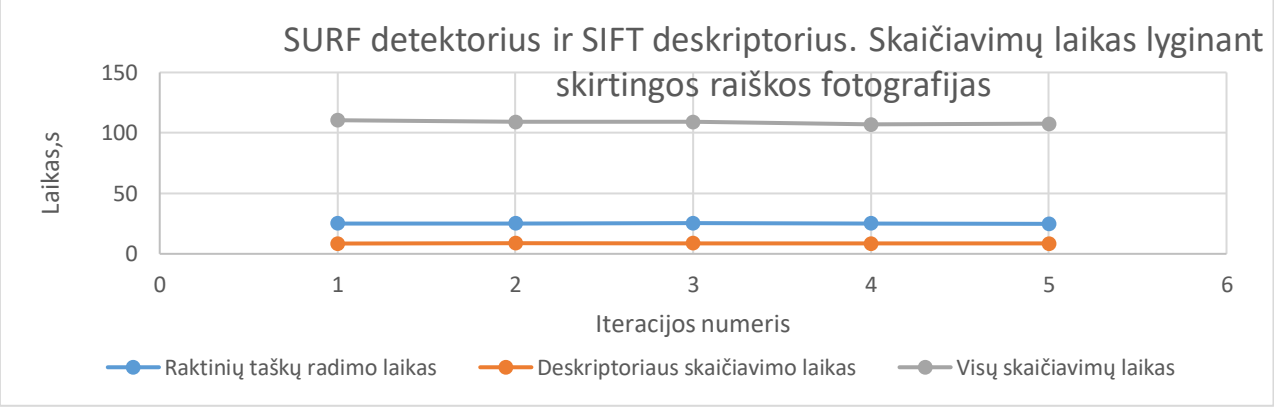
5.2 pav.



5.3 pav.

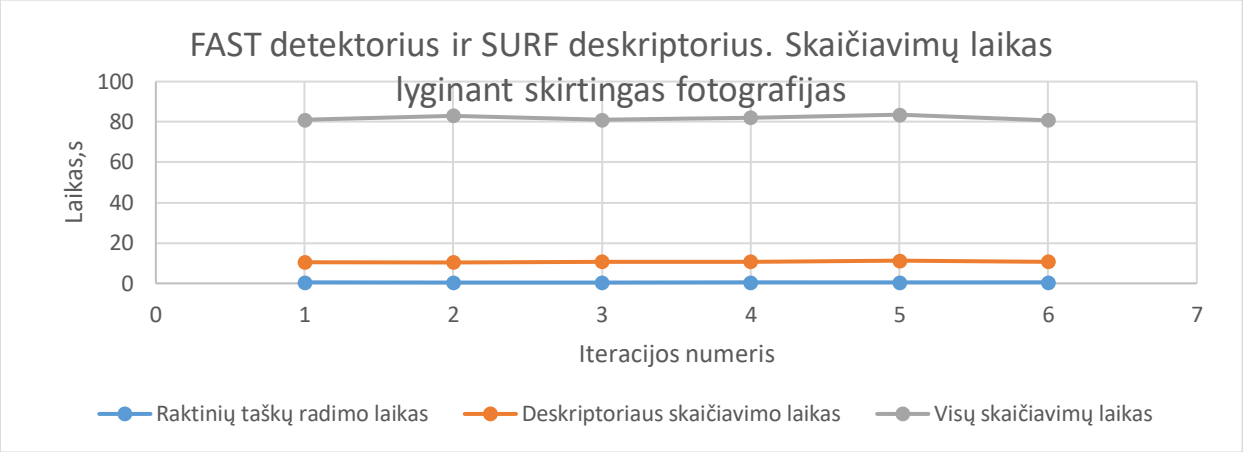

SURF detektorius ir SIFT deskriptorius.Skaičiavimų laikas lyginant fotografijas su sulietu vaizdu
Laikas,s
30
20
10
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

5.4 pav.

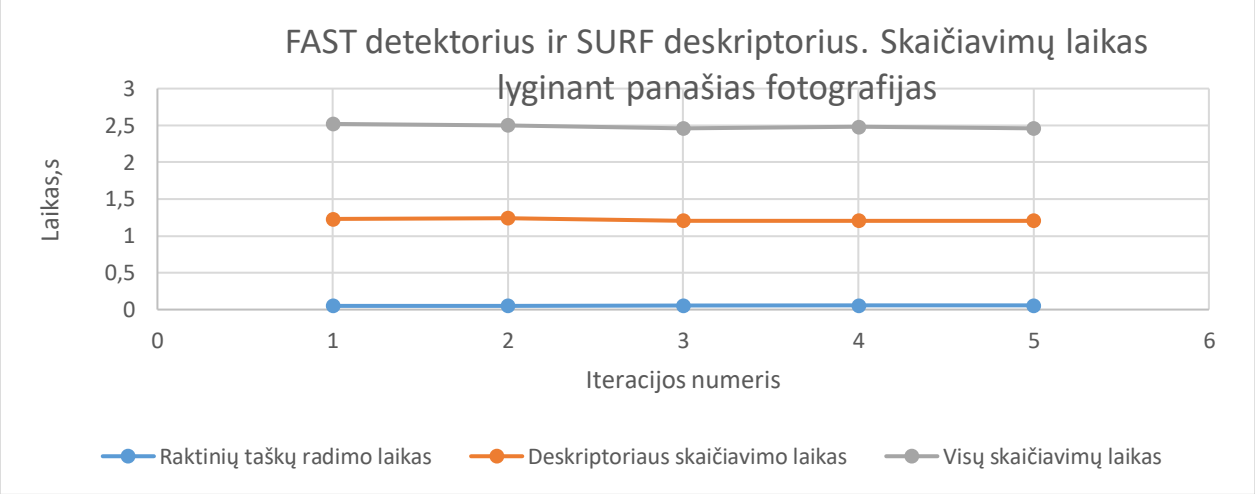

SURF detektorius ir SIFT deskriptorius. Skaičiavimų laikas lyginant skirtingos raiškos fotografijas
Laikas,s
150
100
50
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

5.5 pav.

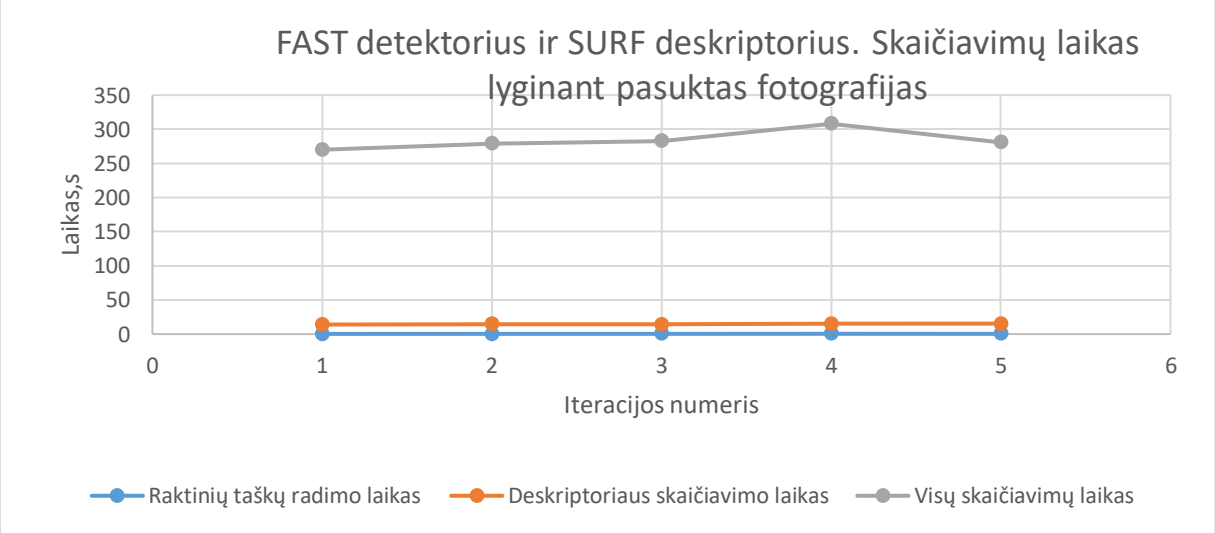
6. Priedas. FAST detektoriaus ir SURF deskriptoriaus greitaveikos tyrimo rezultatai



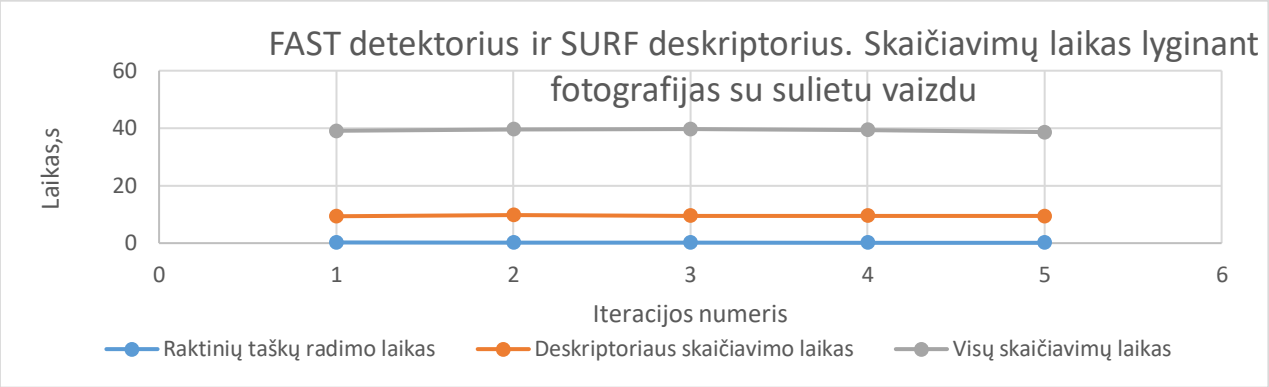
6.1 pav.



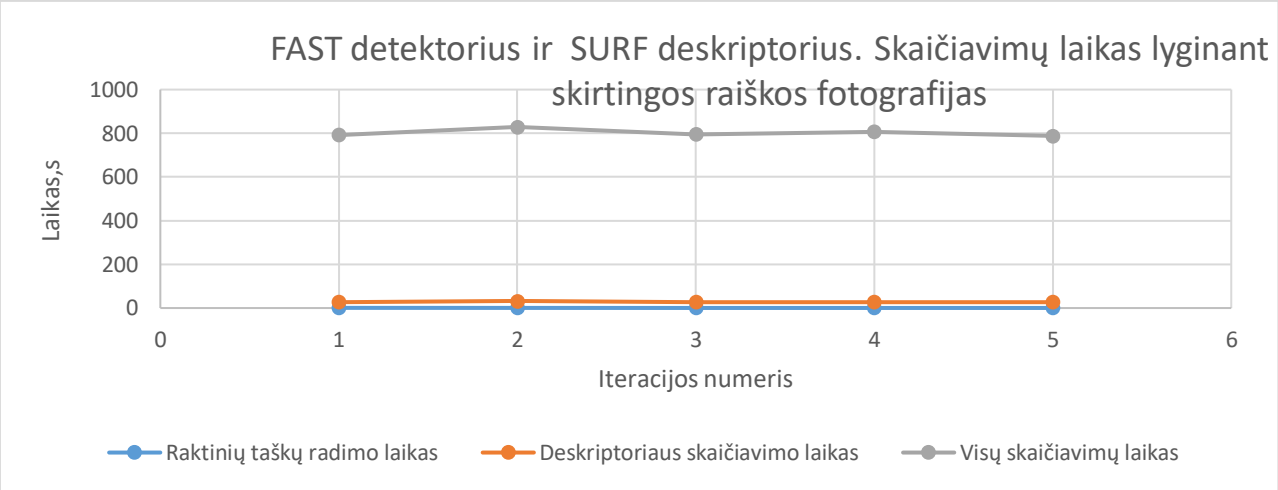
6.2 pav.



6.3 pav.

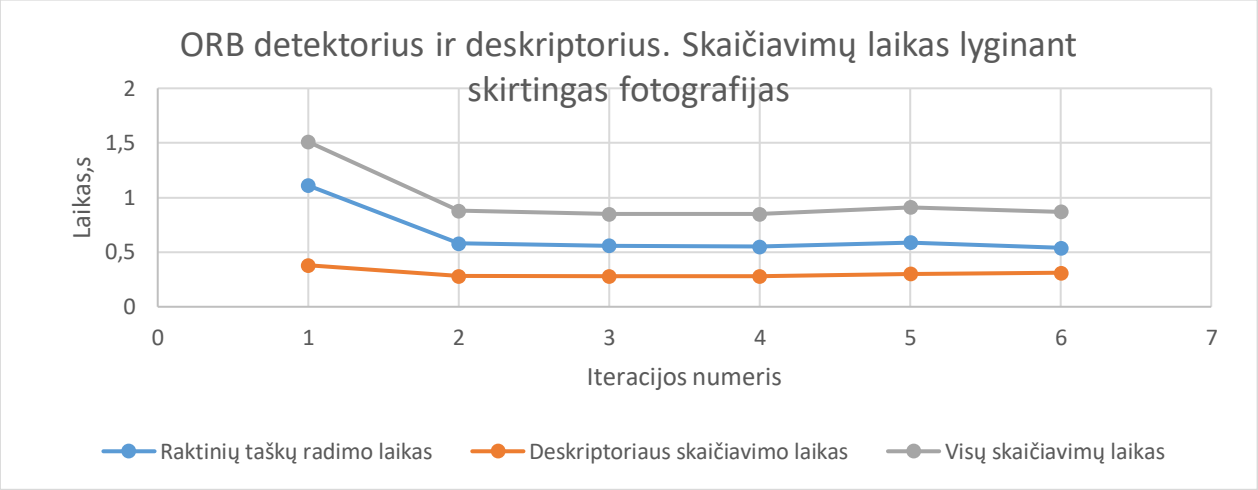

FAST detektorius ir SURF deskriptorius. Skaičiavimų laikas lyginant fotografijas su sulietu vaizdu
Laikas,s
60
40
20
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

6.4 pav.

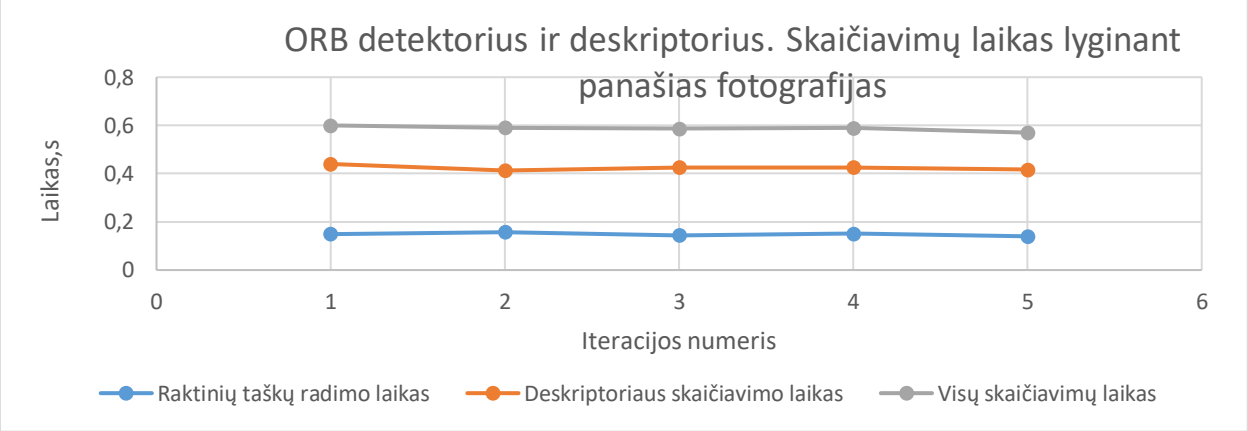

FAST detektorius ir SURF deskriptorius. Skaičiavimų laikas lyginant skirtingos raiškos fotografijas
Laikas,s
1000
800
600
400
200
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

6.5 pav.

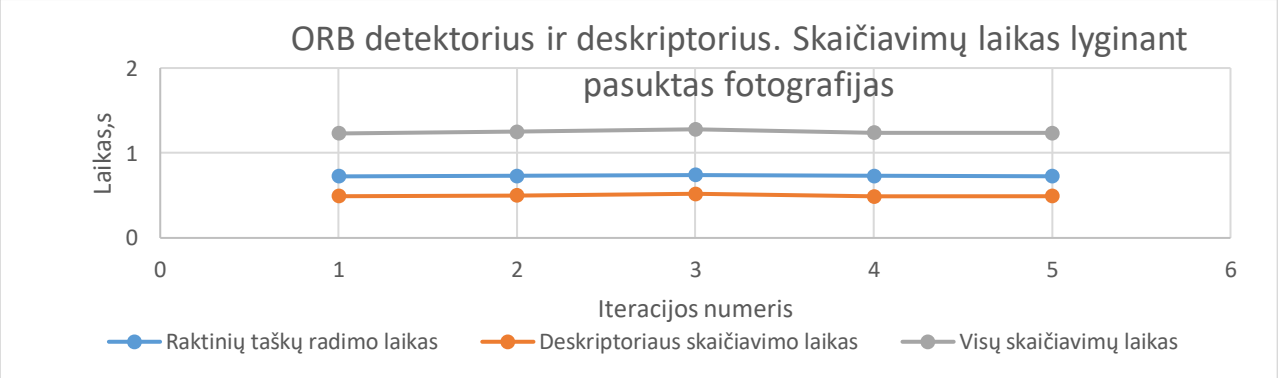
7. Priedas ORB detektoriaus ir deskriptoriaus greitaveikos tyrimo rezultatai



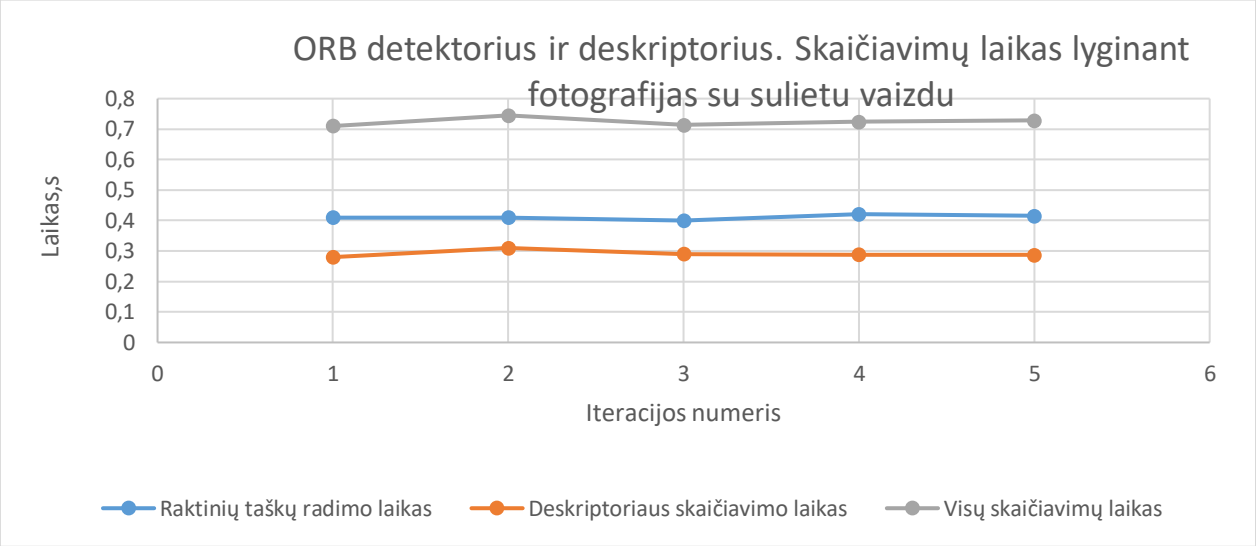
7.1 pav.



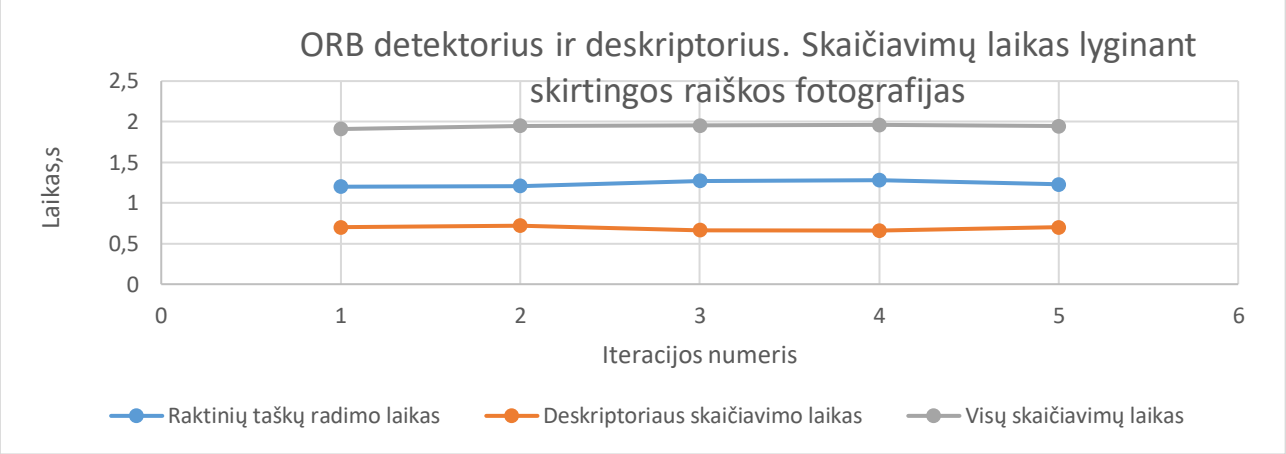
7.2 pav.



7.3 pav.

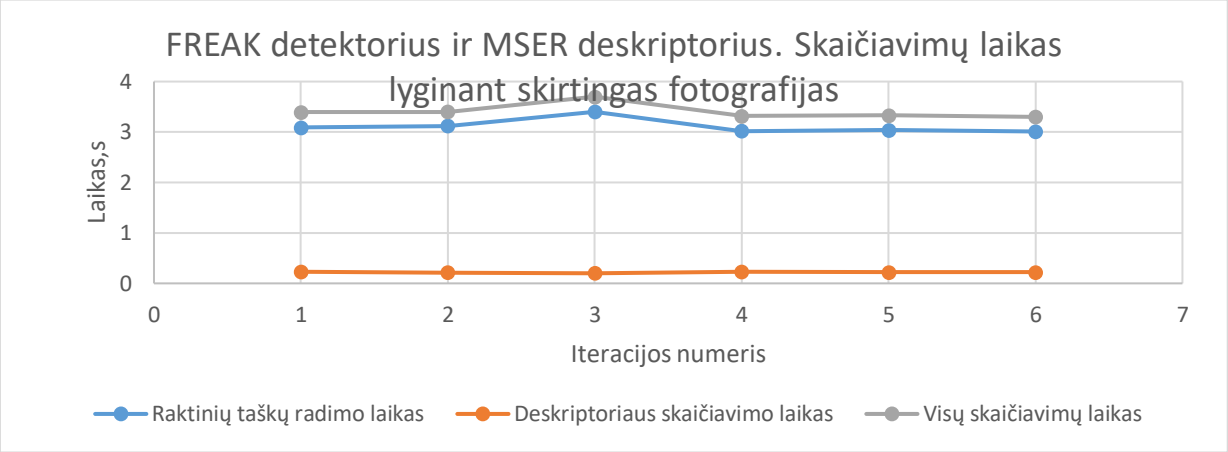

ORB detektorius ir deskriptorius. Skaičiavimų laikas lyginant fotografijas su sulietu vaizdu
Laikas,s
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

7.4 pav.

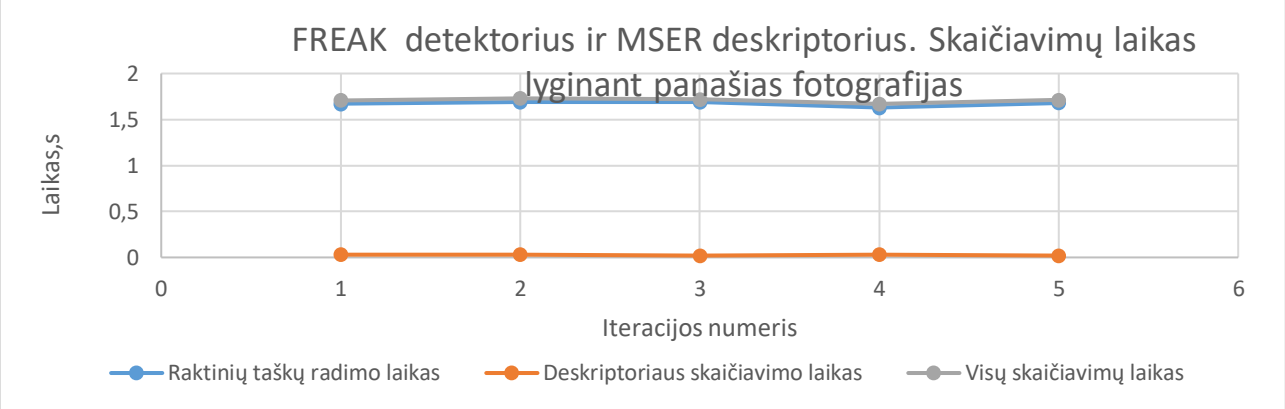

ORB detektorius ir deskriptorius. Skaičiavimų laikas lyginant skirtingos raiškos fotografijas
Laikas,s
2,5
2
1,5
1
0,5
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

7.5 pav.

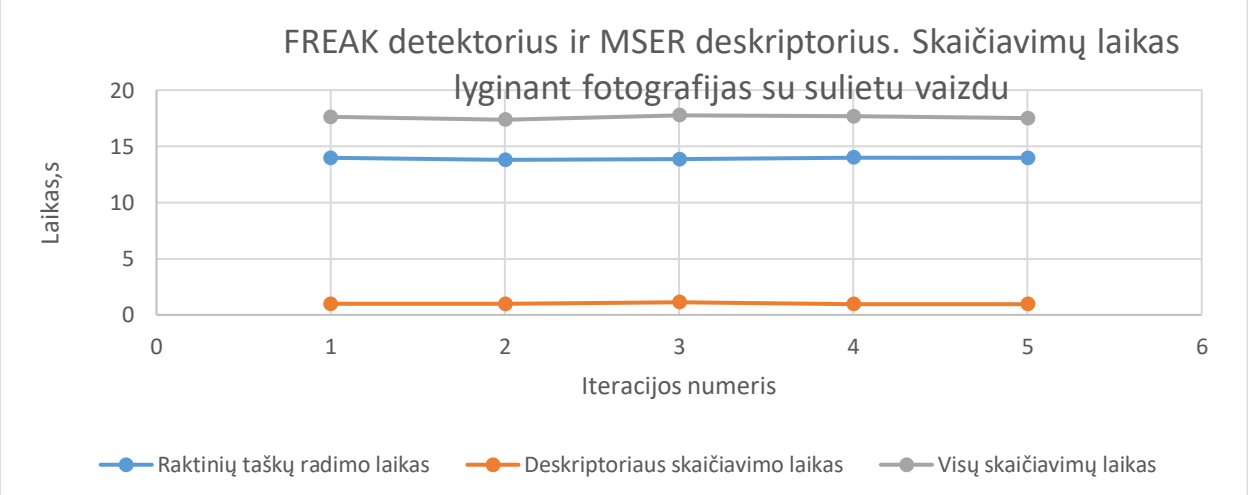
8. Priedas. FREAK detektoriaus ir MSER deskriptoriaus greitaveikos tyrimo rezultatai



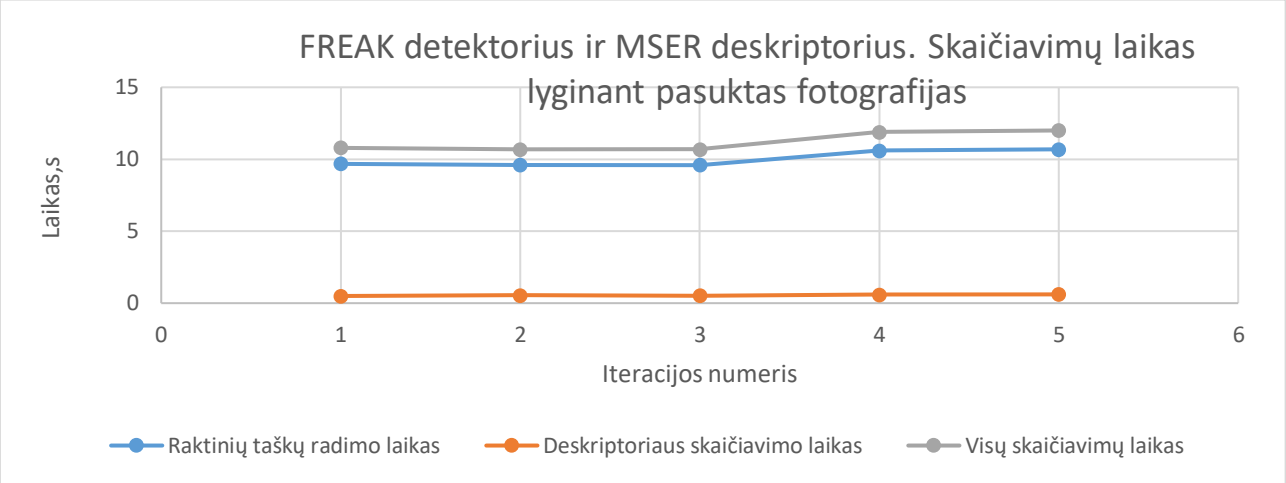
8.1 pav.



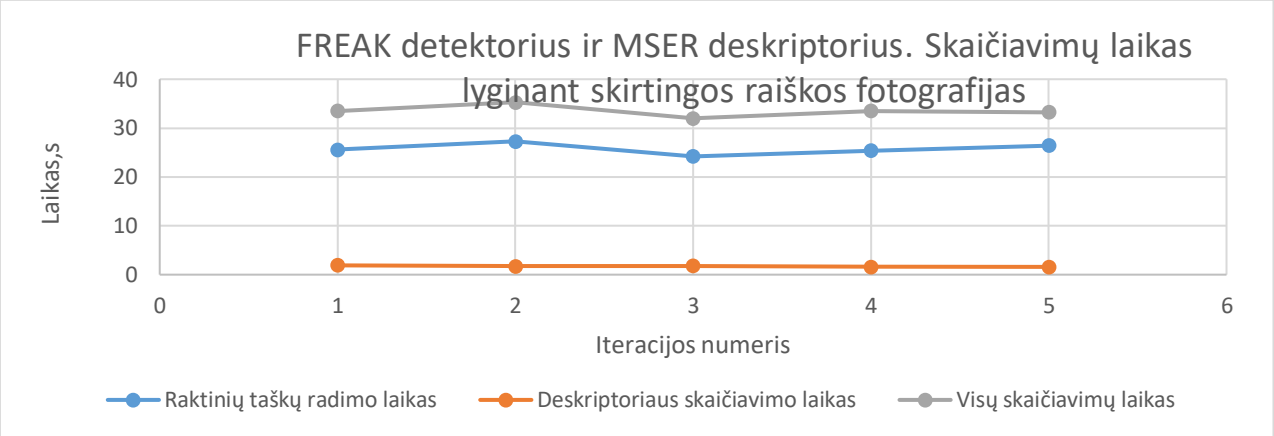
8.2 pav.



8.3 pav.

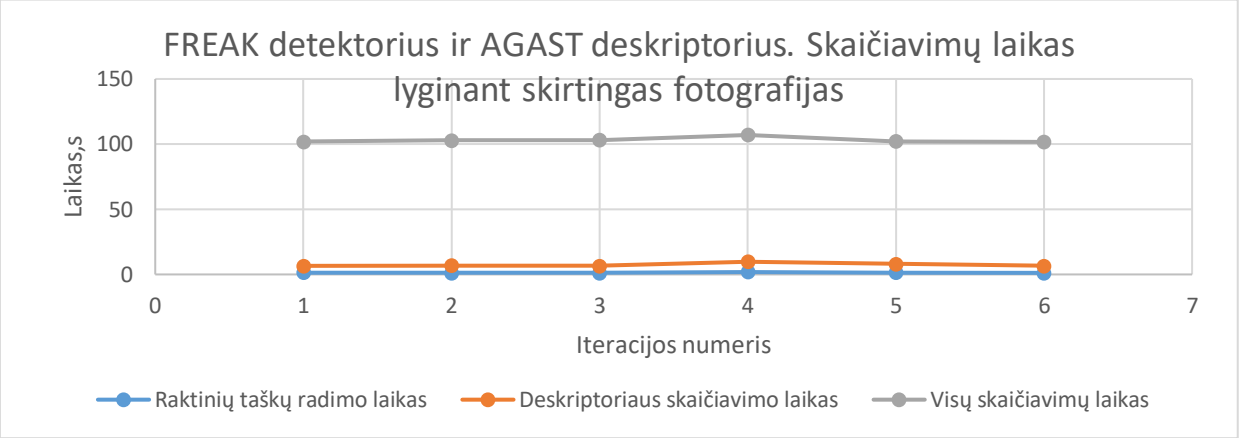

FREAK detektorius ir MSER deskriptorius. Skaičiavimų laikas lyginant pasuktas fotografijas
Laikas,s
15
10
5
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

8.4 pav.

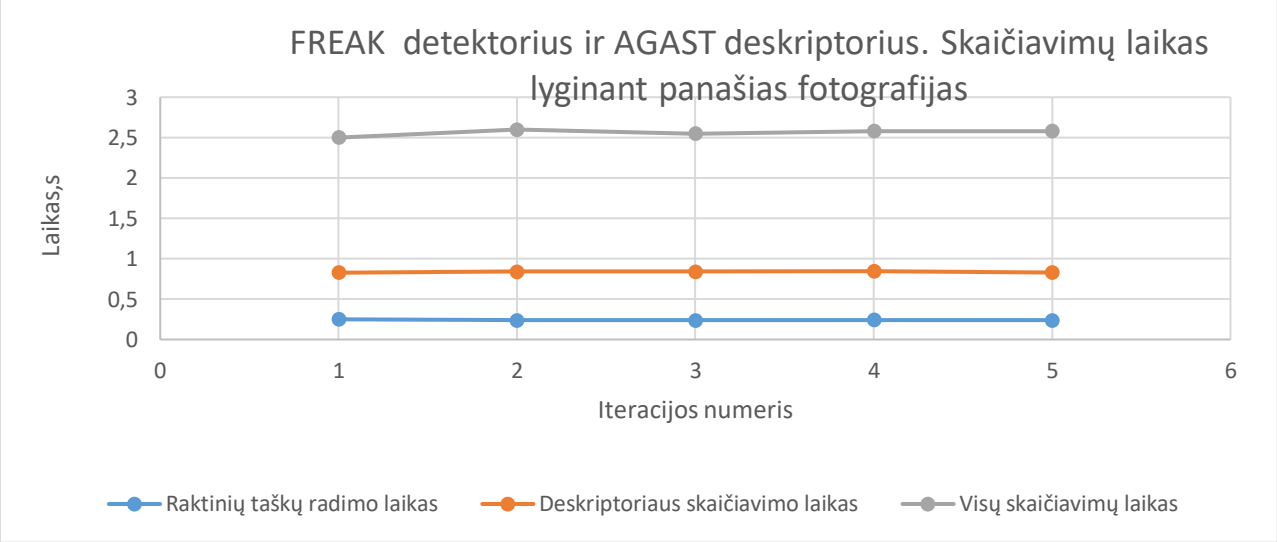

FREAK detektorius ir MSER deskriptorius. Skaičiavimų laikas lyginant skirtingos raiškos fotografijas
Laikas,s
40
30
20
10
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

8.5 pav.

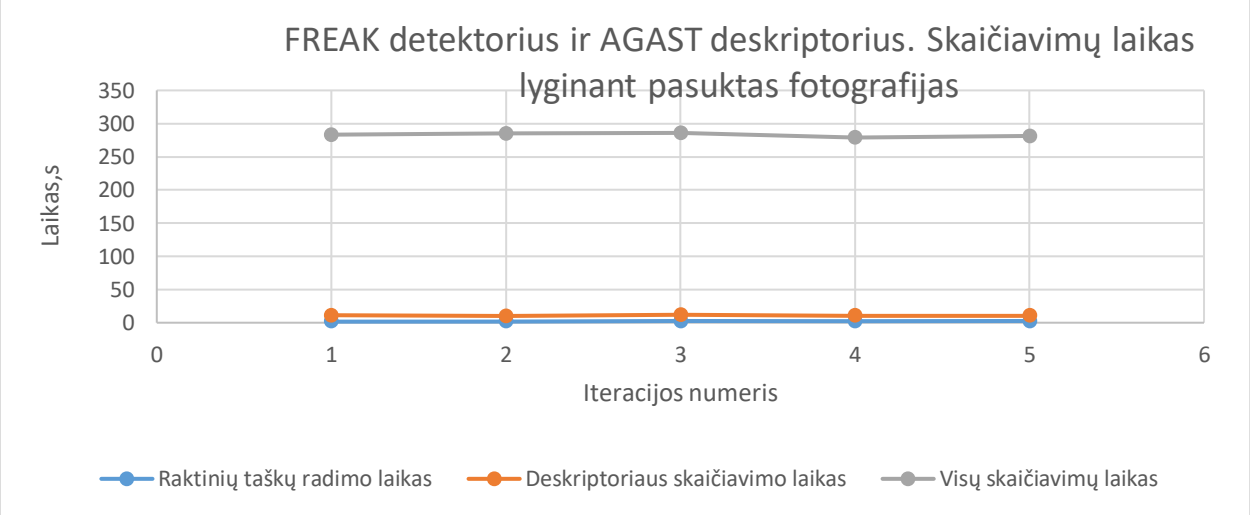
9. Priedas. FREAK detektoriaus ir AGAST deskriptoriaus greitaveikos tyrimo rezultatai



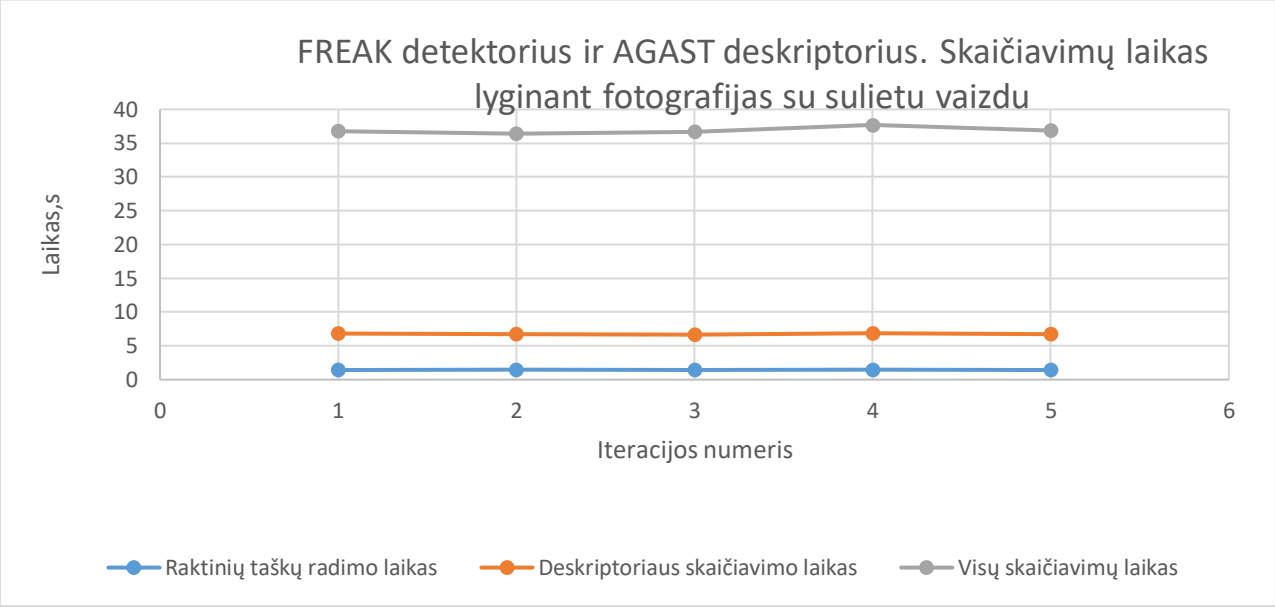
9.1 pav.



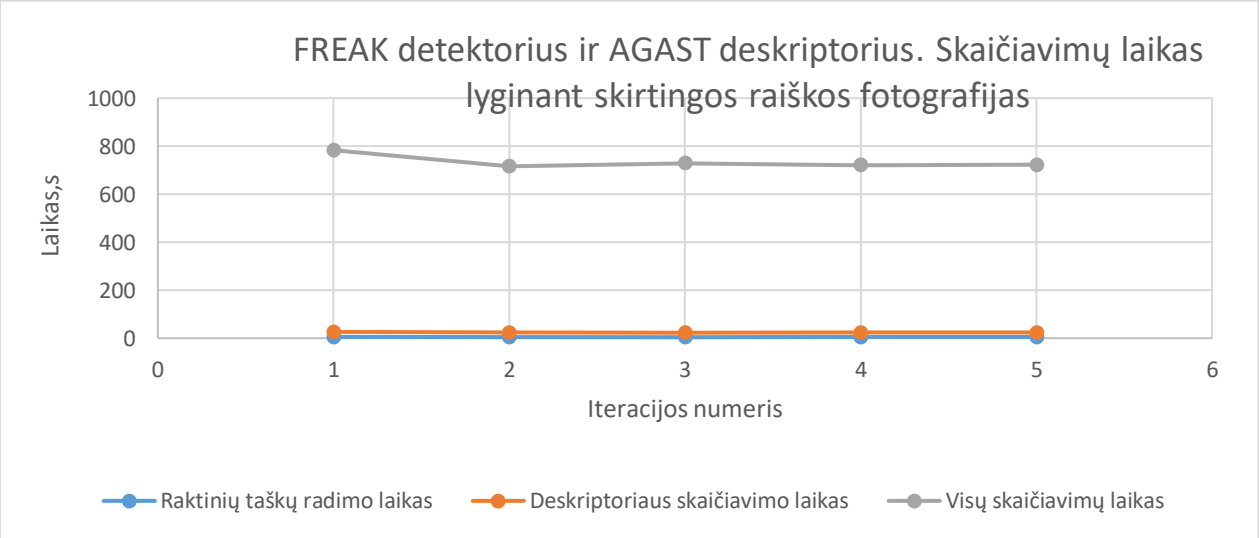
9.2 pav.



9.3 pav.

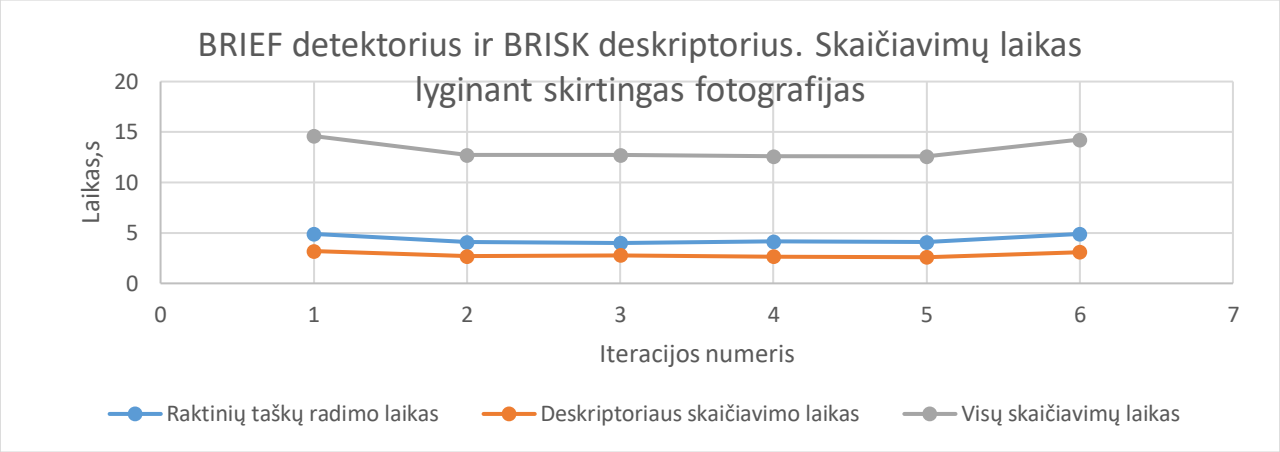

FREAK detektorius ir AGAST deskriptorius. Skaičiavimų laikas lyginant fotografijas su sulietu vaizdu
Laikas,s
40
35
30
25
20
15
10
5
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

9.4 pav.

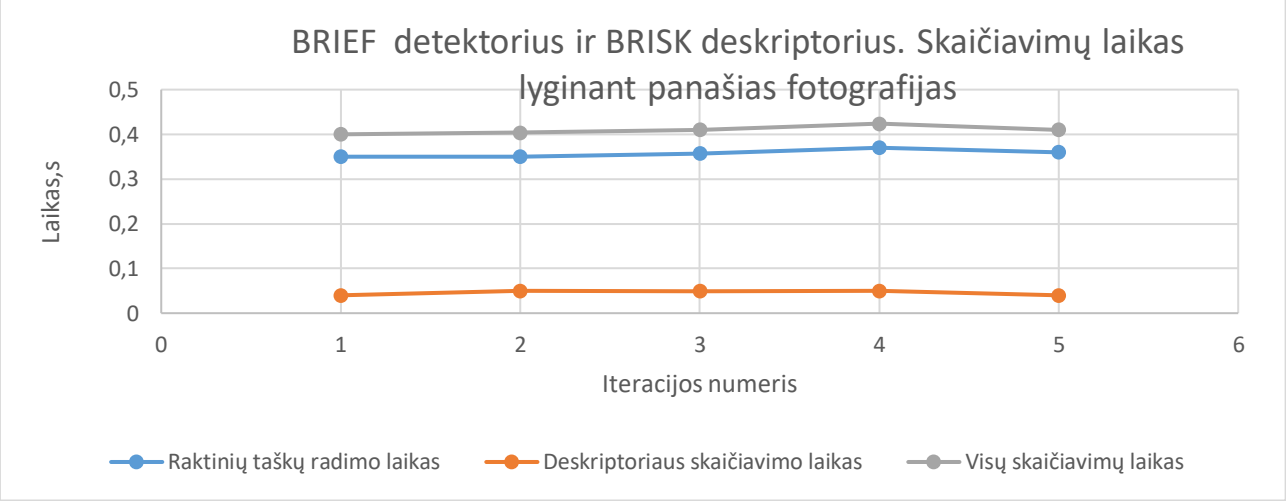

FREAK detektorius ir AGAST deskriptorius. Skaičiavimų laikas lyginant skirtingos raiškos fotografijas
Laikas,s
1000
800
600
400
200
0
0
1
2
3
4
5
6
Iteracijos numeris
Raktinių taškų radimo laikas
Deskriptoriaus skaičiavimo laikas
Visų skaičiavimų laikas

9.5 pav.

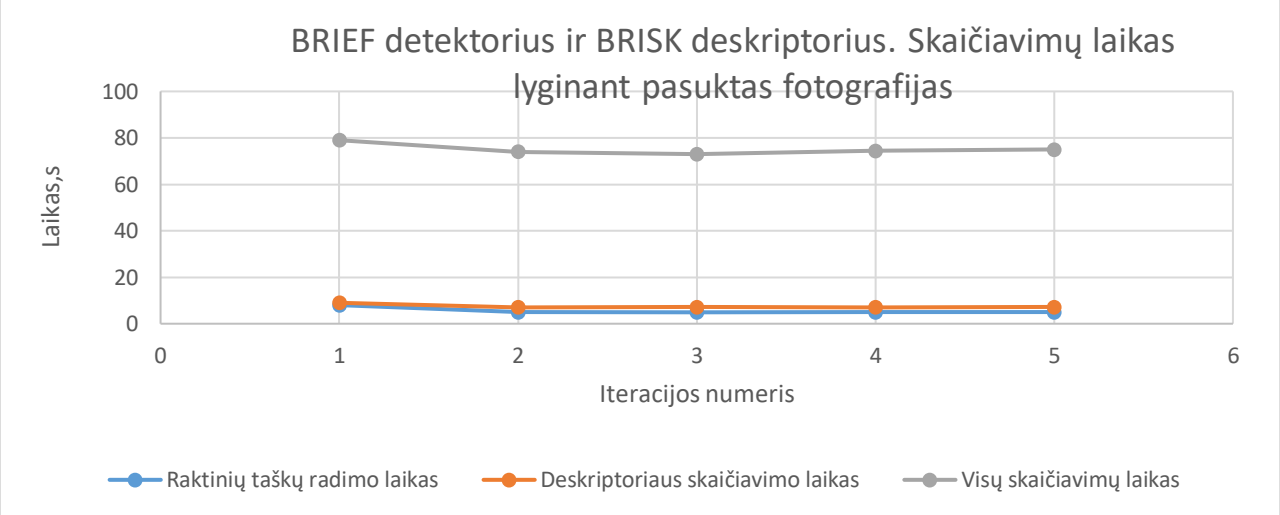
10. priedas. BRIEF detektoriaus ir BRISK deskriptoriaus greitaveikos tyrimo rezultatai



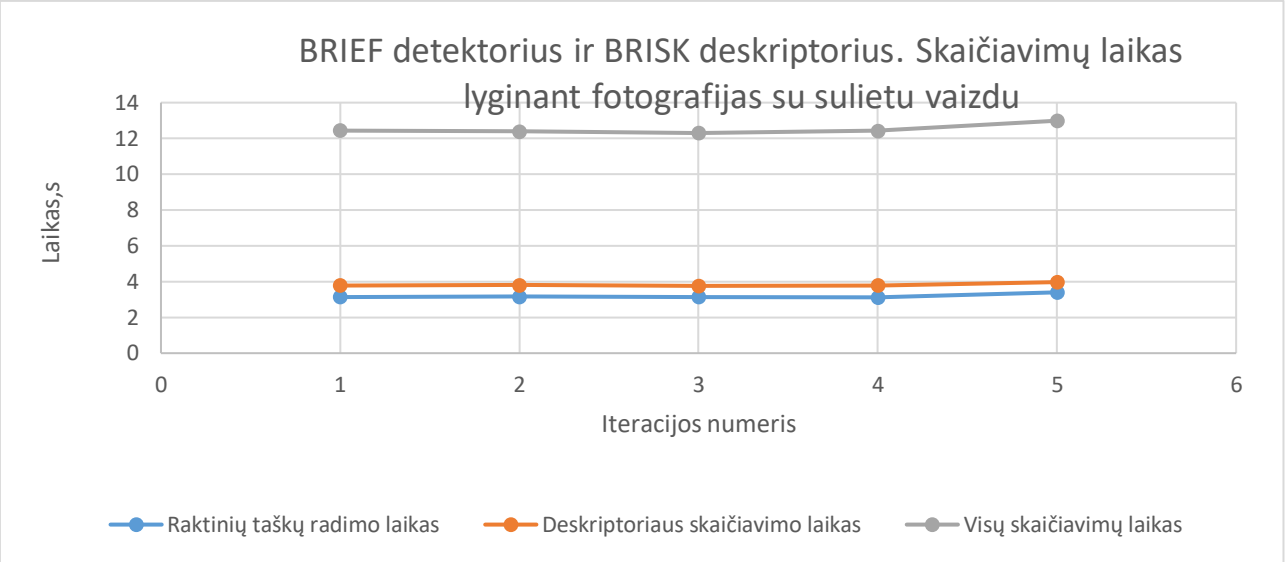
10.1 pav.



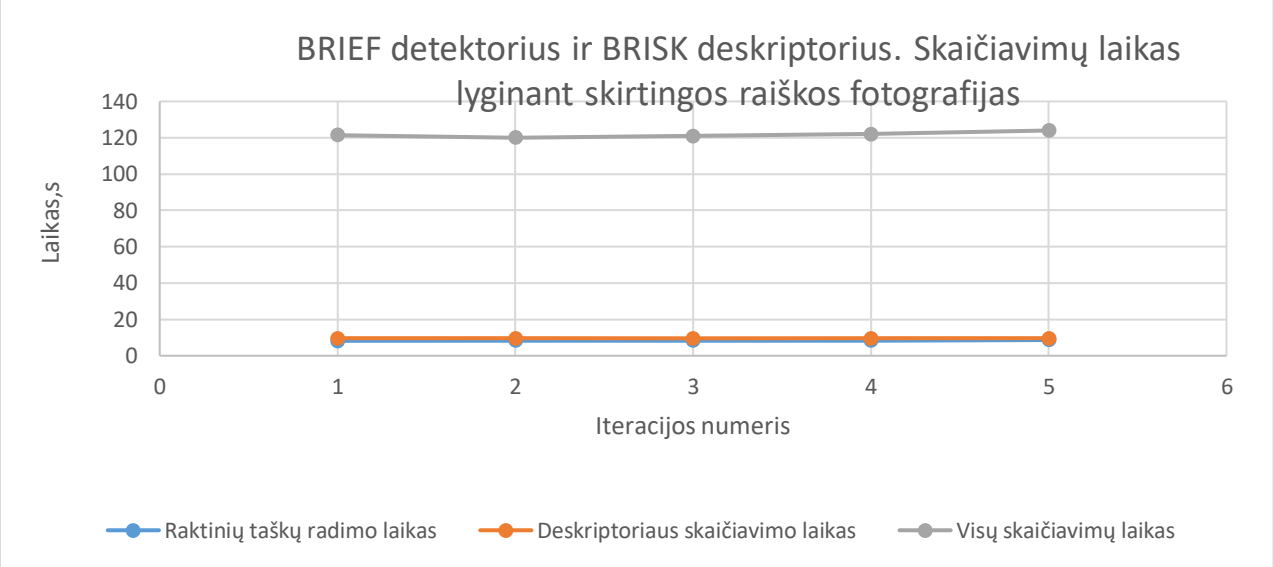
10.2 pav.



10.3 pav.

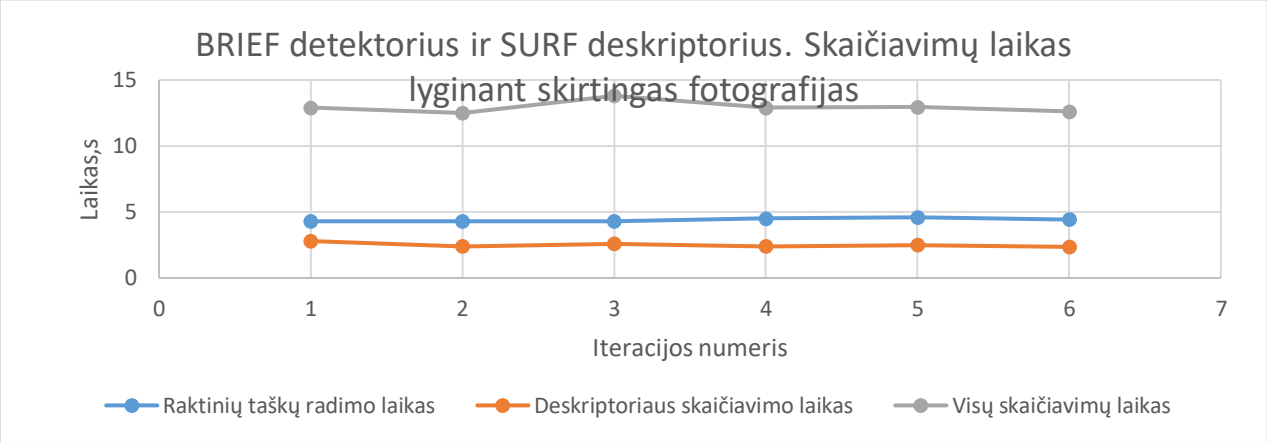


10.4 pav.

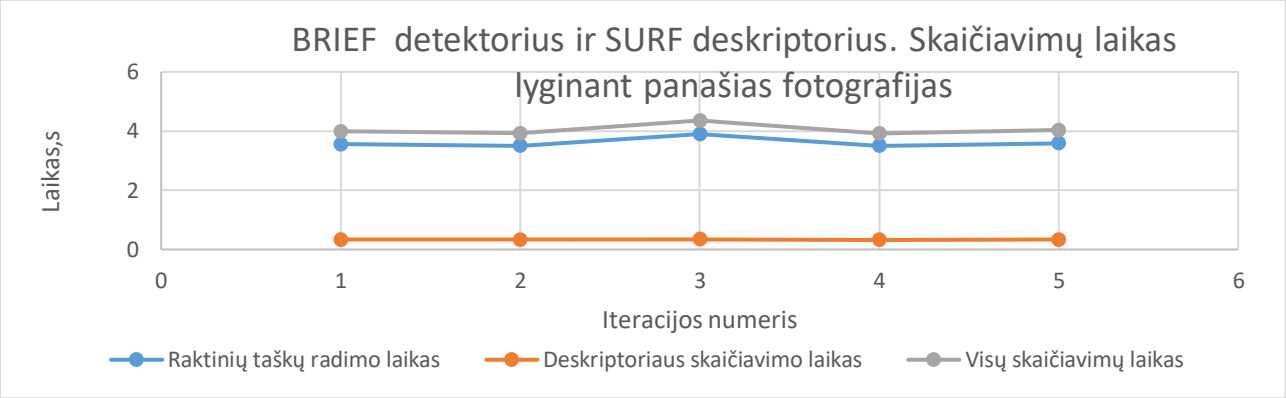


10.5 pav.

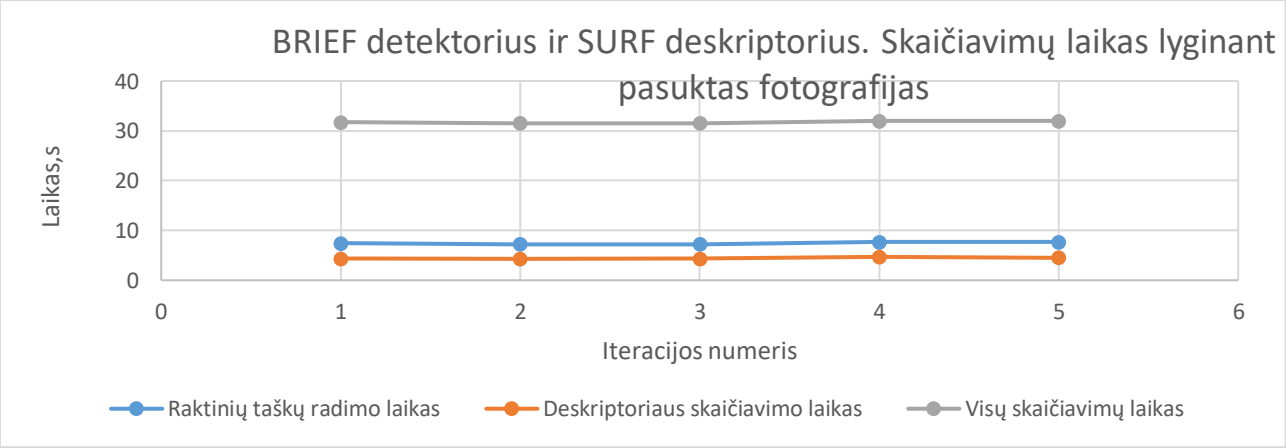
11. priedas. FAST detektoriaus ir SURF deskriptoriaus greitaveikos tyrimo rezultatai



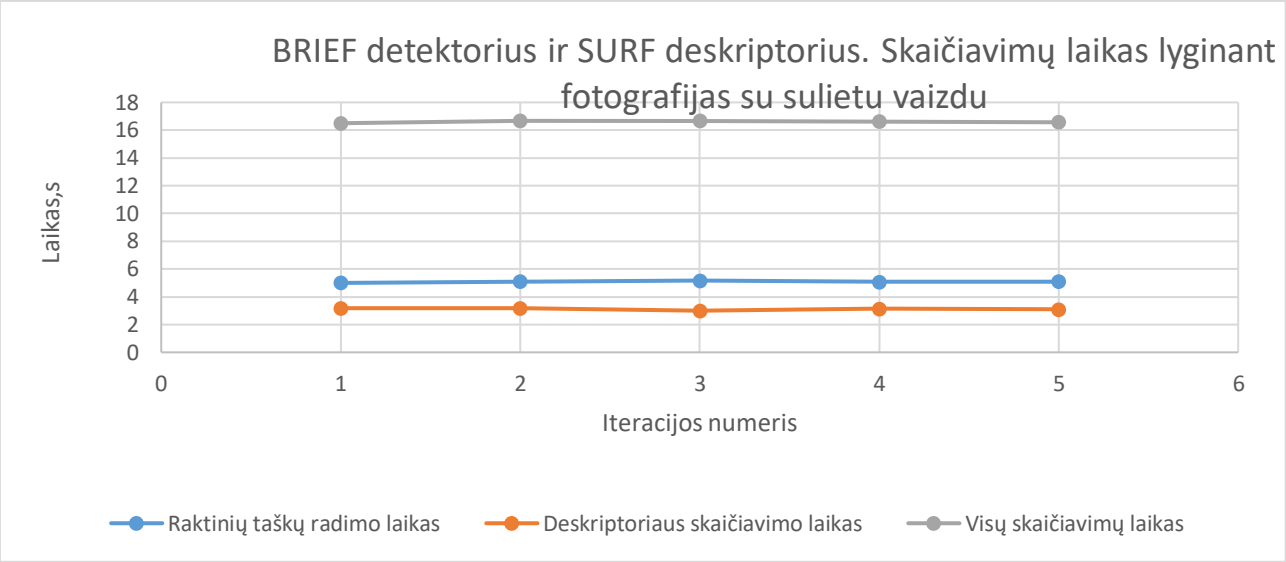
11.1 pav.



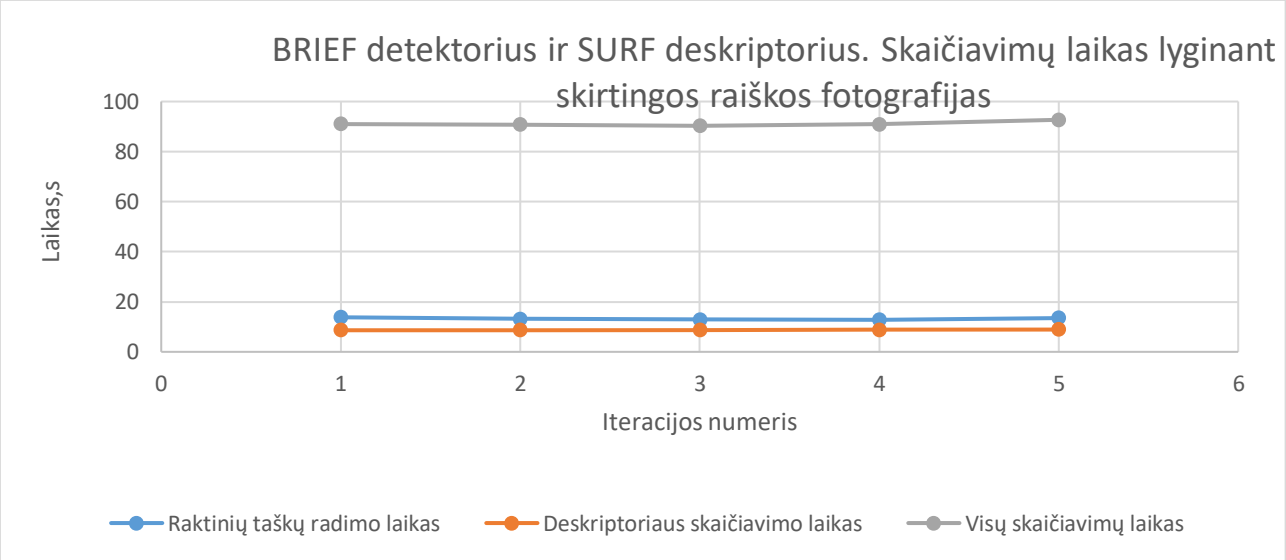
11.2 pav.



11.3 pav.

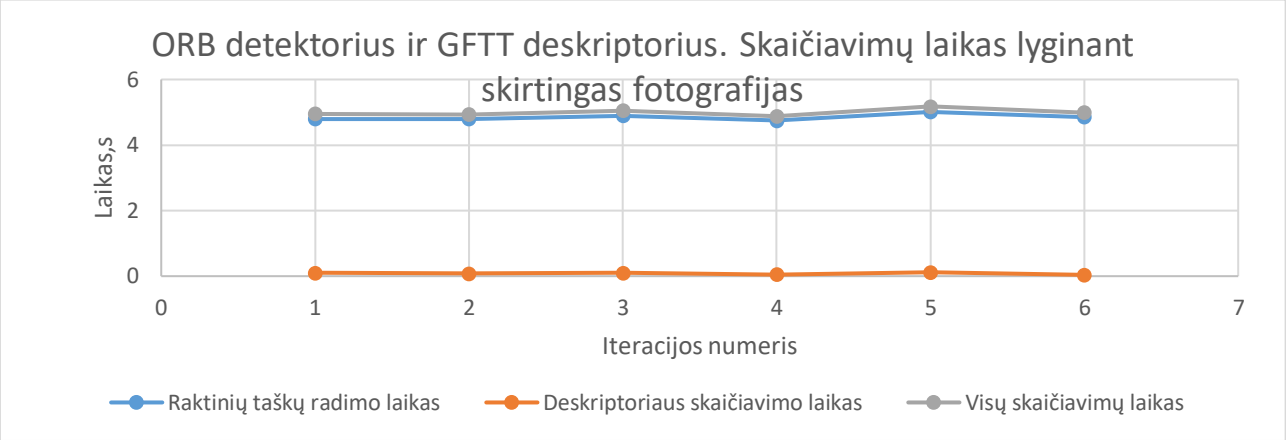


11.4 pav.

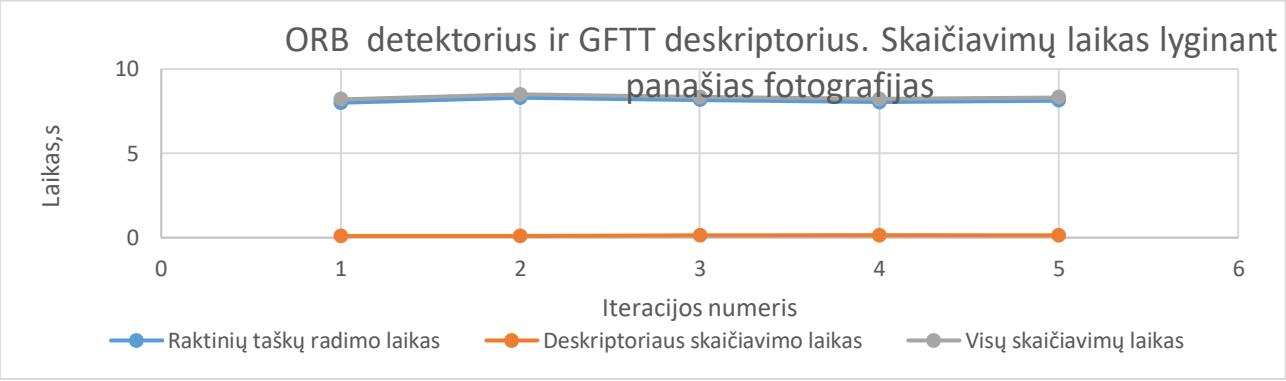


11.5 pav.

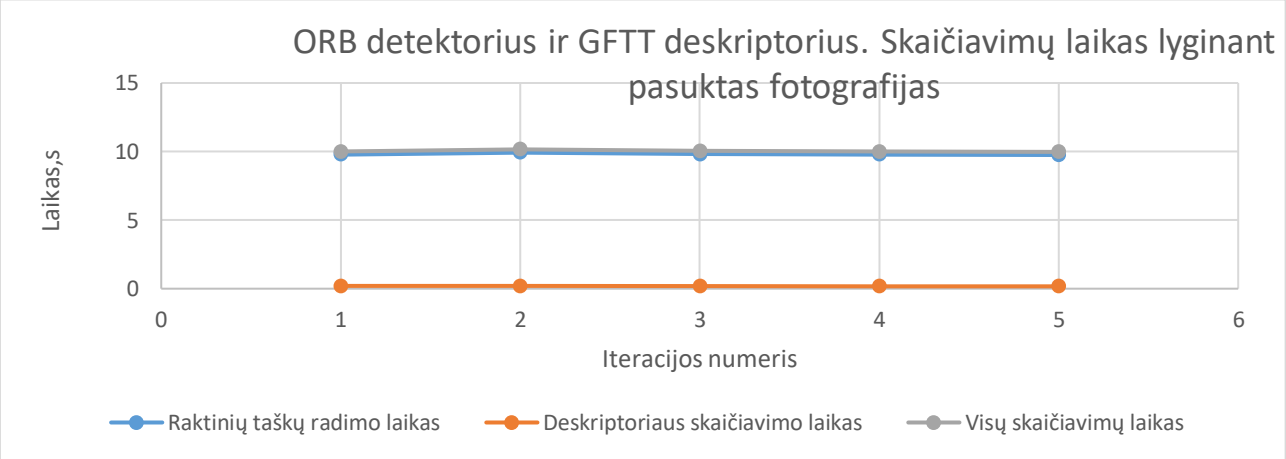
12. priedas. ORB detektoriaus ir GFTT deskriptoriaus greitaveikos tyrimo rezultatai



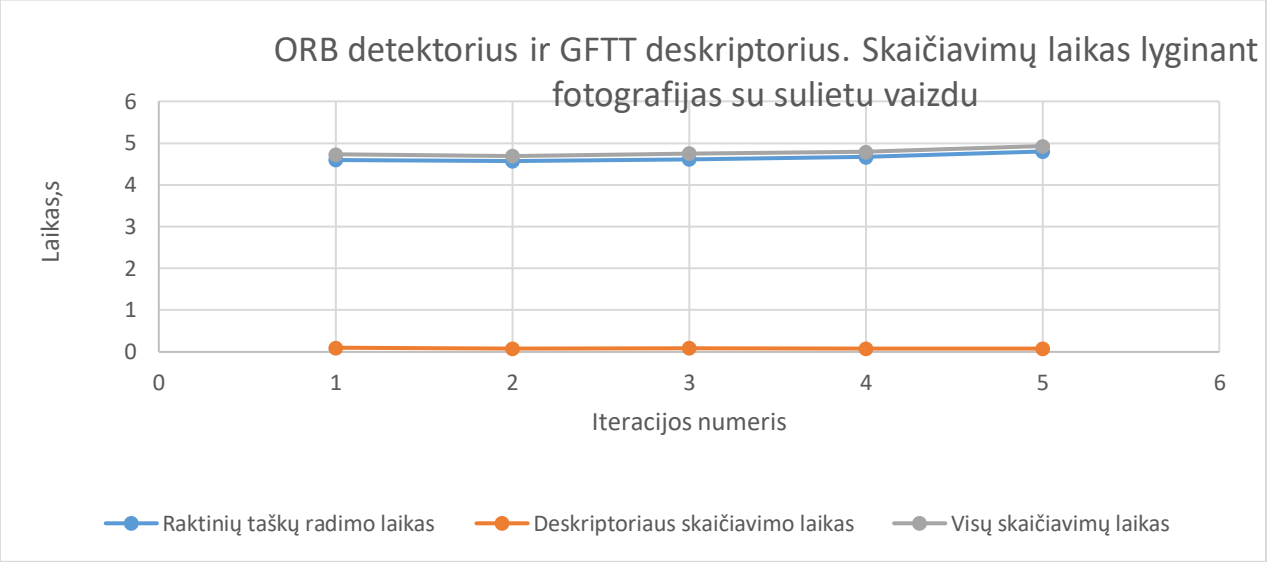
12.1 pav.



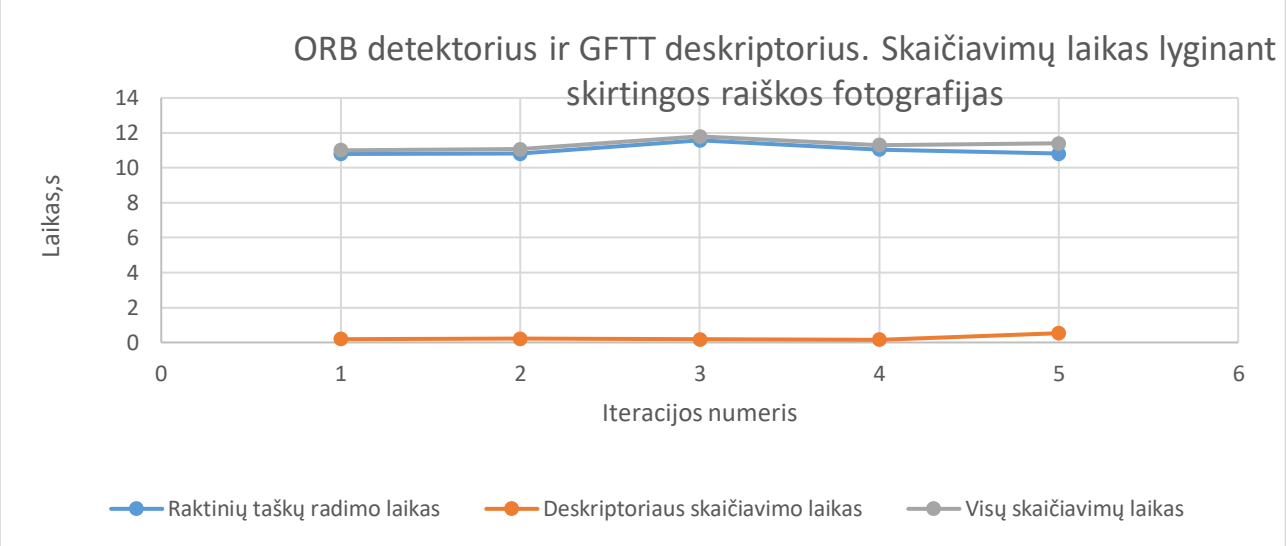
12.2 pav.



12.3 pav.

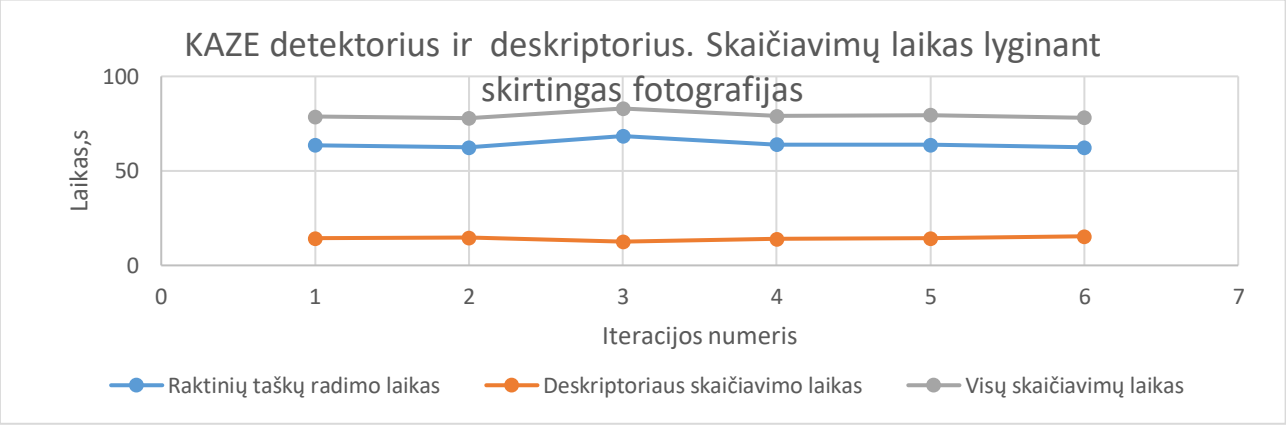


12.4 pav.

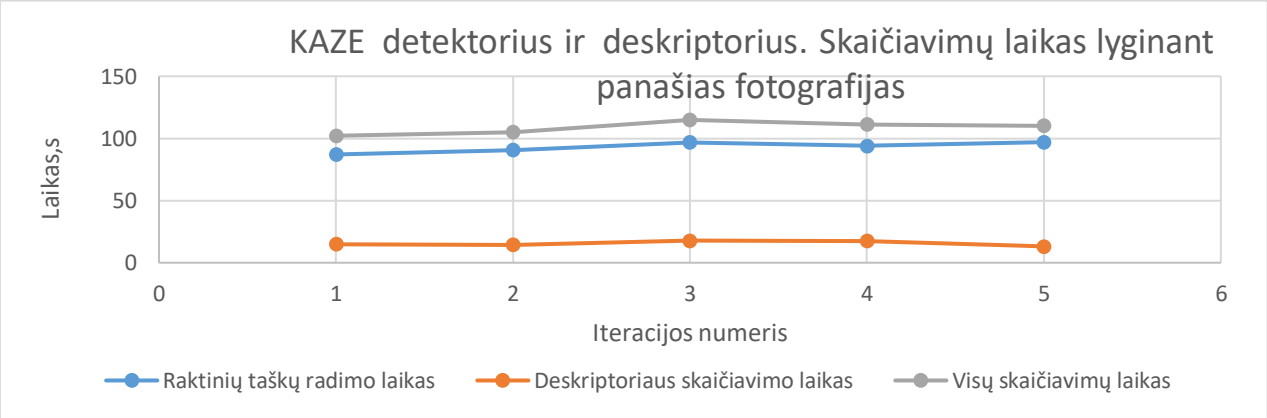


12.5 pav.

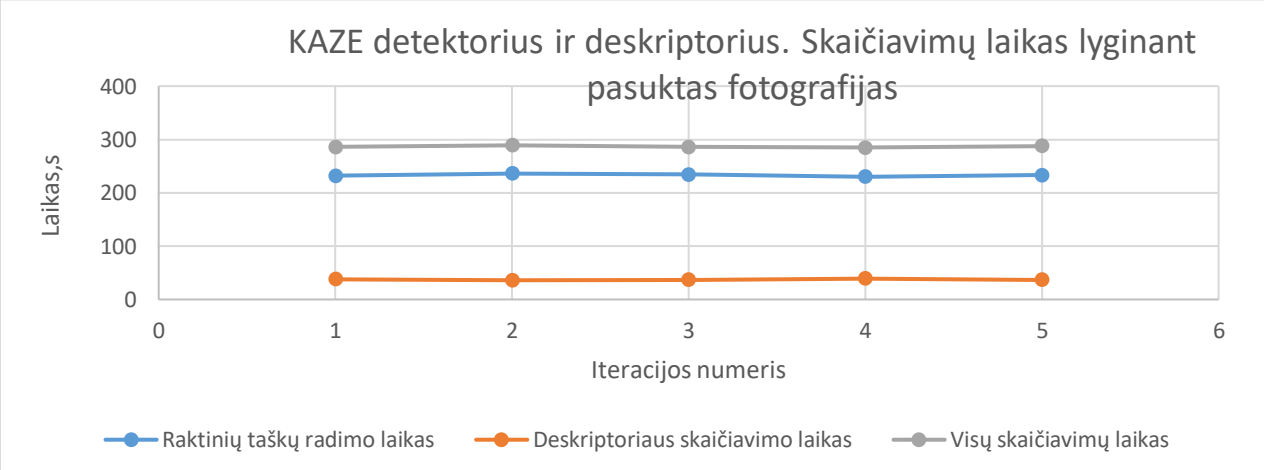
13. priedas. KAZE detektoriaus ir deskriptoriaus greitaveikos tyrimo rezultatai



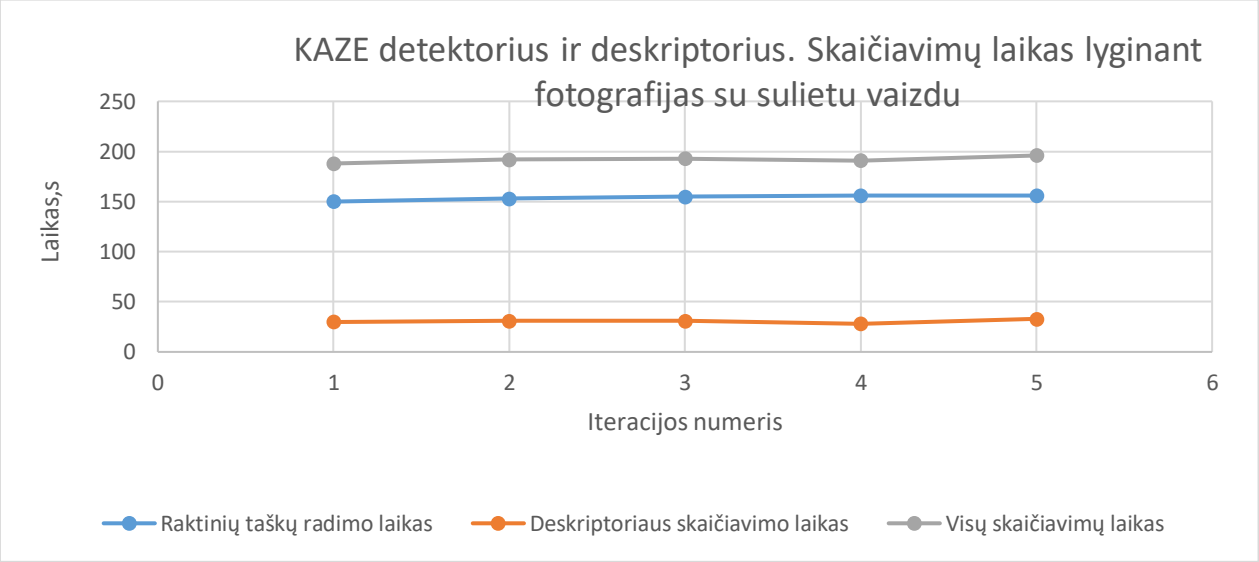
13.1 pav.



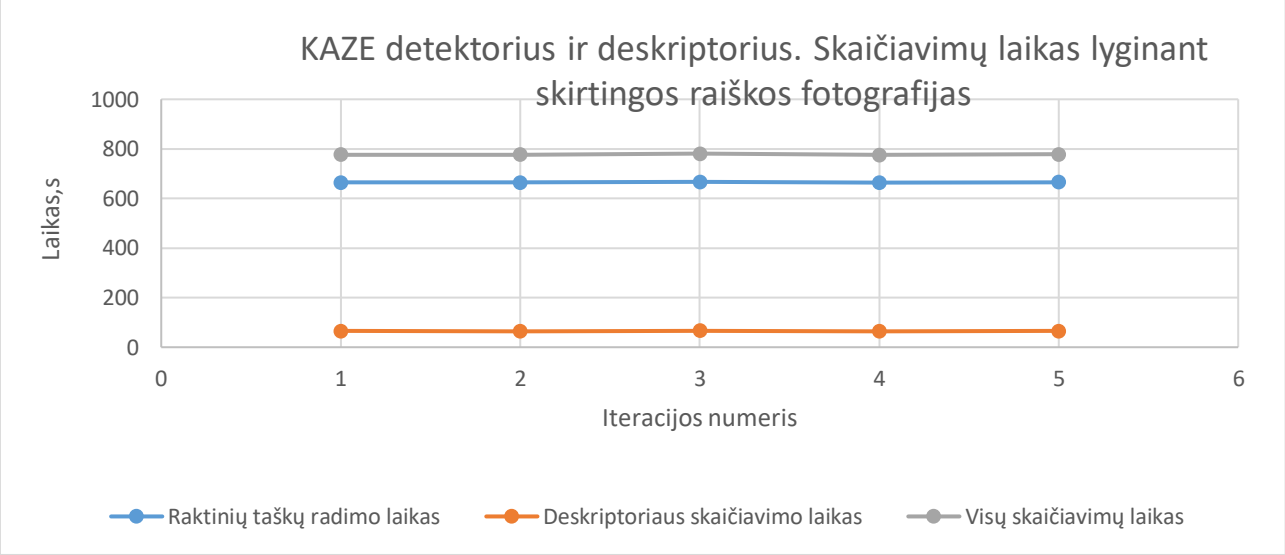
13.2 pav.



13.3 pav.

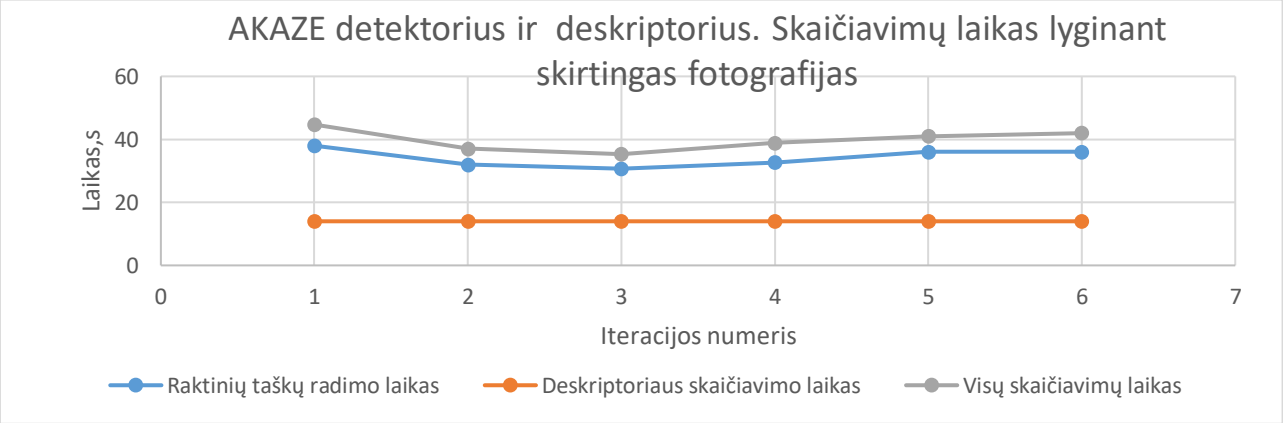


13.4 pav.

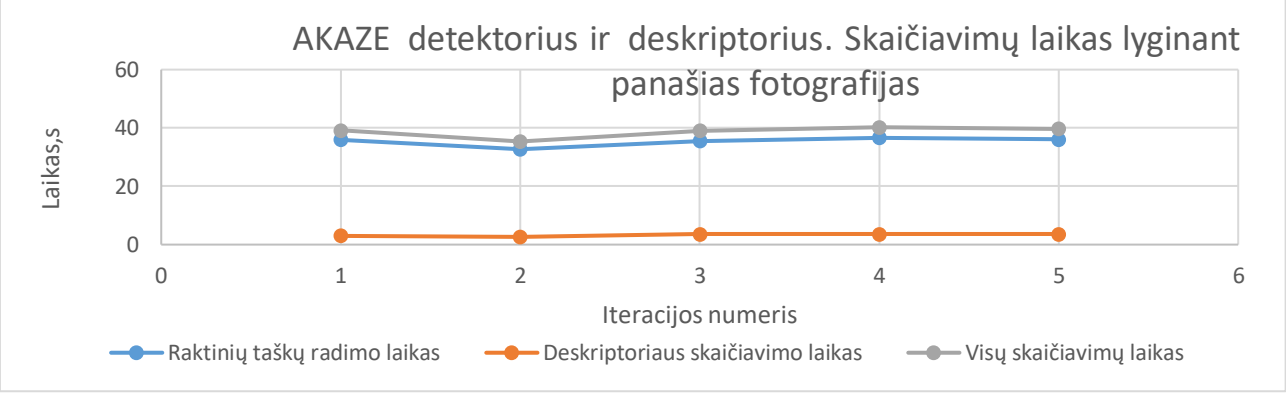


13.5 pav.

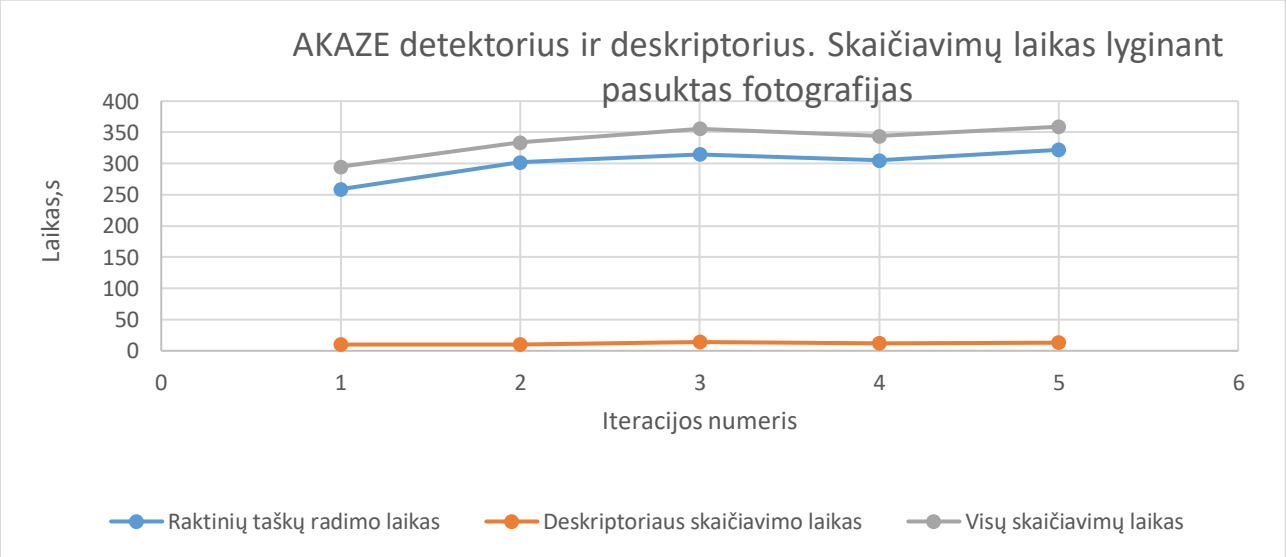
14. priedas. AKAZE detektoriaus ir deskriptoriaus greitaveikos tyrimo rezultatai



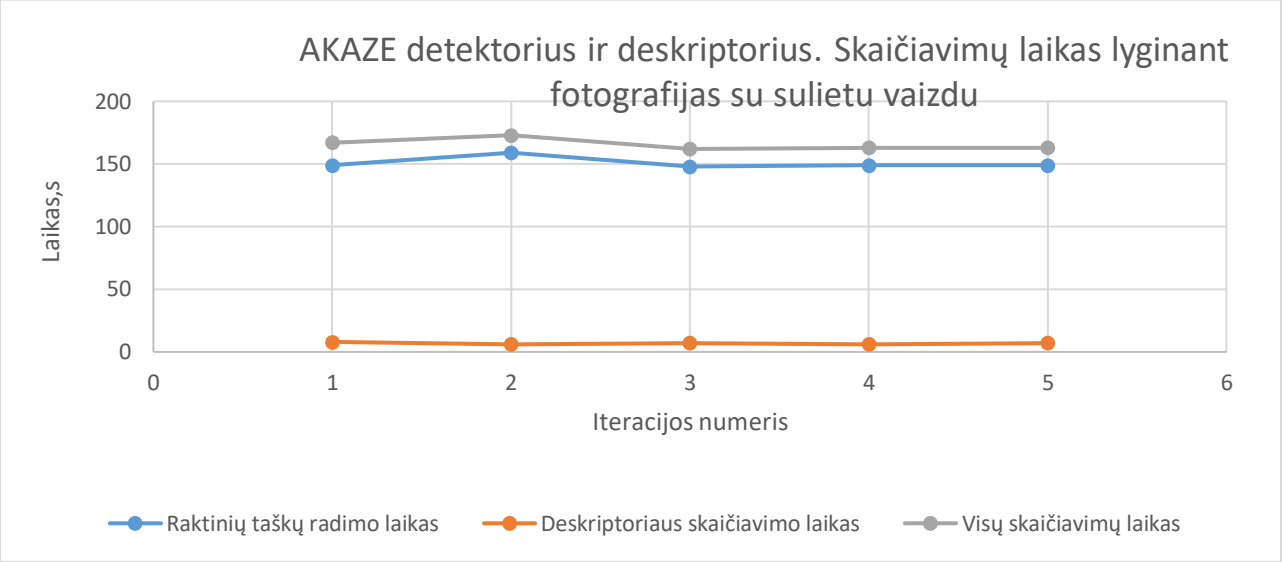
14.1 pav.



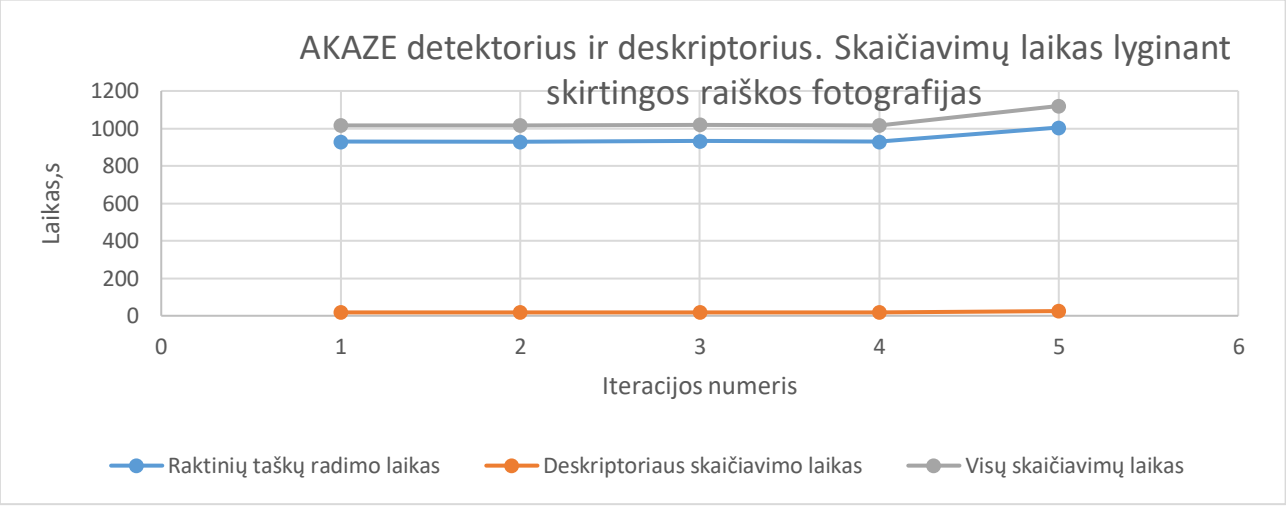
14.2 pav.



14.3 pav.



14.4 pav.



14.5 pav.