

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Ernestas Kardzys

**VAIZDO ATPAŽINIMO SISTEMOS
PRITAIKYMAS OBJEKTŲ KLASIFIKAVIMUI**

Magistro darbas

Darbo vadovas
prof. Eduardas Bareiša

Kaunas, 2012

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Ernestas Kardzys

**VAIZDO ATPAŽINIMO SISTEMOS
PRITAIKYMAS OBJEKTŲ KLASIFIKAVIMUI**

Magistro darbas

Recenzentas

doc. dr. Vytautas Pilkauskas

2012-05-23

Vadovas

prof. Eduardas Bareiša

Atliko

IFM-0/2 gr. stud.

Ernestas Kardzys

2012-05-23

Kaunas, 2012

Adaptation of Video Recognition System for Object Classification

SUMMARY

Video recognition systems are becoming more popular in everyday life. They are used in wide range of areas: traffic control, military, medicine etc.

This paper is exemplified with the project “Universal Video Recognition Systems” developed by Liudas Drejeris, Ernestas Kardzys and Karolis Kriščiūnas.

This paper concentrates on the problems encountered during creation of video recognition system: the selection of algorithm for video recognition, the selection of system architecture used, comparison of JSON and XML formats for information interchange.

The created video recognition system is tested with three different kinds of objects (books, cups and toys) and is 92% accurate.

System was created using Java and C++ programming languages. It uses SURF algorithm for image processing. System has clients for Windows, Linux and Android OS, server works on Linux OS.

Vaizdo atpažinimo sistemos pritaikymas objektų klasifikavimui

SANTRAUKA

Vaizdo atpažinimo sistemos tampa vis svarbesnės kasdieniniame gyvenime. Jos yra naudojamos įvairiose srityse – eismo saugumo užtikrinime, karinėje pramonėje, medicinoje.

Šio darbo praktinė realizacija yra „Universali vaizdo atpažinimo sistema“, sukurta Liudo Drejerio, Ernesto Kardzio ir Karolio Kriščiūno. Darbas tyrinėja problemas, su kuriomis yra susiduriama kuriant vaizdo atpažinimo sistemą: nuotraukų apdorojimo algoritmo pasirinkimą, naudojamą sistemos architektūrą, JSON ir XML formatų palyginimas informacijos perdavimui.

Sistema buvo bandoma su trimis objektų klasėmis: „knygomis“, „puodeliais“, „žaislais“ ir buvo gautas vidutinis sėkmingai atpažintų objektų procentas - 92%.

Sistema parašyta naudojant Java ir C++ programavimo kalbas, naudoja SURF algoritmą objektų savybių išskyrimui. Sistema turi klientus Windows, Linux, Android OS, serveris veikia ant Linux OS.

Turinys

| | | |
|----------|---|-----------|
| 1 | ĮVADAS | 10 |
| 2 | ANALITINĖ DALIS | 12 |
| 2.1 | VAIZDŲ ATPAŽINIMO ALGORITMAI..... | 12 |
| 2.2 | BRIAUNŲ APTIKIMO ALGORITMAI..... | 12 |
| 2.2.1 | <i>Gauso funkcija (Gaussian function)</i> | 13 |
| 2.2.2 | <i>Gradientų aptikimas</i> | 13 |
| 2.2.3 | <i>Kany briaunų aptikėjas</i> | 14 |
| 2.2.4 | <i>„Naudingų taškų“ aptikimo algoritmai</i> | 15 |
| 2.2.4.1 | Hariso kampų aptikimo algoritmas..... | 15 |
| 2.2.4.2 | SUSAN kampų aptikimo algoritmas | 16 |
| 2.2.5 | <i>Nekintančių savybių aptikimo algoritmai</i> | 17 |
| 2.3 | ALGORITMŲ PALYGINIMAI | 17 |
| 2.3.1 | <i>Hariso ir SUSAN algoritmų palyginimas</i> | 17 |
| 2.3.2 | <i>SURF, Hariso, Gauso, Hesiano – Laplaso algoritmų palyginimas</i> | 19 |
| 2.4 | INFORMACIJOS PERDAVIMO PROTOKOLO FORMATAS | 20 |
| 2.4.1 | <i>Tyrimo aprašymas</i> | 20 |
| 2.4.2 | <i>Tyrimo rezultatai</i> | 21 |
| 2.5 | OBJEKTŲ ATPAŽINIMO BIBLIOTEKOS | 23 |
| 2.5.1 | <i>OpenCV biblioteka</i> | 24 |
| 2.5.2 | <i>Insight Segmentation and Registration Toolkit (ITK)</i> | 24 |
| 2.5.3 | <i>Integrating Vision Toolkit (IVT)</i> | 24 |
| 2.5.4 | <i>Visualization Toolkit (VTK)</i> | 25 |
| 2.5.5 | <i>Bibliotekų palyginimas</i> | 25 |
| 2.6 | SIFT IR SURF DESKRIPTORIŲ PALYGINIMAS..... | 25 |
| 2.7 | IŠVADOS..... | 26 |
| 3 | PROJEKTINĖ DALIS..... | 27 |
| 3.1 | PROJEKTO TIKSLAS..... | 27 |
| 3.2 | REIKALAVIMAI SISTEMAI IR SISTEMOS FUNKCIJOS | 27 |
| 3.3 | PANAŠŪS PROJEKTAI | 29 |
| 3.3.1 | <i>Google Googles</i> | 29 |
| 3.3.2 | <i>GemIdent</i> | 29 |
| 3.3.3 | <i>Bartendroid</i> | 30 |
| 3.4 | NUOTRAUKOS APDOROJIMAS | 30 |
| 3.5 | SISTEMOS ARCHITEKTŪROS APRAŠYMAS..... | 30 |
| 3.5.1 | <i>Serverio architektūros aprašymas</i> | 31 |
| 3.5.1.1 | Prisijungiančių klientų aptarnavimas..... | 31 |
| 3.5.1.2 | Objektų paieška, šalinimas, atnaujinimas ir pridėjimas..... | 31 |

| | | |
|----------|---|-----------|
| 3.5.2 | <i>Kliento aprašymas</i> | 31 |
| 3.6 | SERVERIO ARCHITEKTŪRA | 32 |
| 3.6.1 | <i>Serverio komponentų diagramos</i> | 32 |
| 3.6.2 | <i>SocketService komponentas</i> | 33 |
| 3.6.3 | <i>MySQLService komponentas</i> | 34 |
| 3.6.4 | <i>Actions komponentas</i> | 35 |
| 3.6.5 | <i>ObjectRecognition komponentas</i> | 36 |
| 3.6.6 | <i>Main komponentas</i> | 38 |
| 3.7 | KLIENTO ARCHITEKTŪRA | 39 |
| 3.7.1 | <i>Kliento komponentų diagrama</i> | 39 |
| 3.7.2 | <i>LiveImageProcessingPackage komponento diagrama</i> | 39 |
| 3.7.3 | <i>ClientObjectRecognitionPackage komponento diagrama</i> | 41 |
| 3.7.4 | <i>MainPackage komponento diagrama</i> | 43 |
| 3.8 | DUOMENŲ BAZĖS STRUKTŪRA | 44 |
| 3.9 | SISTEMOS ĮDIEGIMAS..... | 45 |
| 3.10 | IŠVADOS | 45 |
| 4 | TYRIMO DALIS | 46 |
| 4.1 | OBJEKTŲ KLASIFIKAVIMO ALGORITMO APRAŠYMAS | 47 |
| 4.2 | REIKALINGAS SISTEMOS PAKEITIMAS | 49 |
| 5 | EKSPERIMENTINĖ DALIS | 50 |
| 5.1 | TECHNINĖ IR PROGRAMINĖ ĮRANGA..... | 50 |
| 5.2 | EKSPERIMENTO TIKSLAS | 50 |
| 5.3 | REALIZACIJA | 50 |
| 5.4 | EKSPERIMENTO ATLIKIMO SĄLYGOS | 53 |
| 5.5 | EKSPERIMENTO ATLIKIMO TVARKA..... | 54 |
| 5.6 | EKSPERIMENTŲ REZULTATAI..... | 56 |
| 5.7 | EKSPERIMENTO IŠVADOS..... | 57 |
| 5.8 | PASTEBĖTI SISTEMOS TRŪKUMAI..... | 57 |
| 6 | IŠVADOS | 59 |
| 7 | LITERATŪRA | 60 |
| 8 | TERMINŲ IR SANTRUMPŲ ŽODYNAS | 62 |
| 9 | PRIEDAI | 64 |
| 9.1 | JSON FORMATO ŽINUTĖS INFORMACIJOS APSIKEITIMUI | 64 |
| 9.1.1 | <i>Objekto pridėjimas</i> | 64 |
| 9.1.2 | <i>Objekto išrinkimas</i> | 65 |
| 9.1.3 | <i>Informacinė žinutė</i> | 66 |
| 9.1.4 | <i>Klaidos žinutė</i> | 67 |
| 9.1.5 | <i>Žinutė, jog objektas atpažintas</i> | 67 |
| 9.1.6 | <i>Žinutė, jog objektas neatpažintas</i> | 69 |

| | | |
|-----|--|----|
| 9.2 | MOBILIOJO TELEFONO HTC Evo 3D PARAMETRAI | 70 |
| 9.3 | APKLAUSOS ANKETA..... | 70 |
| 9.4 | DUOMENŲ BAZĖS LENTELIŲ APRAŠYMAS..... | 71 |
| 9.5 | NAUDOTOS KOMPILIAVIMO PROGRAMINĖS ĮRANGOS PARAMETRAI | 73 |
| 9.6 | EKSPERIMENTO REZULTATAI..... | 74 |
| 9.7 | UAB „INFORMACINIŲ TECHNOLOGIJŲ ORGANIZACIJA“ ĮDIEGIMO AKTAS..... | 77 |
| 9.8 | VĮ „REGISTRŲ CENTRAS“ ĮDIEGIMO AKTAS | 78 |
| 9.9 | KTU TECHNORAMOS PADĖKOS RAŠTAS..... | 79 |

Paveikslukų sąrašas

| | |
|---|----|
| 1 PAV. ORIGINALUS PAVEIKSLIUKAS „LENA“ | 12 |
| 2 PAV. PAVEIKSLIUKO „LENA“ BRIAUNOS | 13 |
| 3 PAV. STABILUMO ANALIZĖ | 18 |
| 4 PAV. ATSPARUMAS TRIUKŠMAMS | 18 |
| 5 PAV. ATPAŽINIMO LAIKAS | 20 |
| 6 PAV. JSON FORMATO ATMINTIES IR CPU PANAUDOJIMAS | 22 |
| 7 PAV. XML FORMATO ATMINTIES IR CPU PANAUDOJIMAS | 23 |
| 8 PAV. SERVERIO PANAUDOS ATVEJŲ DIAGRAMA | 28 |
| 9 PAV. KLIENTUI PANAUDOS ATVEJŲ DIAGRAMA | 28 |
| 10 PAV. BENDRA SISTEMOS PAKETŲ DIAGRAMA | 32 |
| 11 PAV. SOCKETSERVICE KOMPONENTO KLASIŲ DIAGRAMA | 33 |
| 12 PAV. MYSQLSERVICE KOMPONENTO DIAGRAMA | 34 |
| 13 PAV. ACTIONS KOMPONENTO PAKETŲ DIAGRAMA | 36 |
| 14 PAV. OBJECTRECOGNITION KOMPONENTO KLASIŲ DIAGRAMA | 37 |
| 15 PAV. MAIN KOMPONENTO KLASIŲ DIAGRAMA | 38 |
| 16 PAV. KLIENTO PAGRINDINĖ KOMPONENTŲ DIAGRAMA | 39 |
| 17 PAV. LIVEIMAGEPROCESSINGPACKAGE KOMPONENTO DIAGRAMA | 40 |
| 18 PAV. CLIENTOBJECTRECOGNITIONPACKAGE KOMPONENTO DIAGRAMA | 42 |
| 19 PAV. MAINPACKAGE KOMPONENTO DIAGRAMA | 43 |
| 20 PAV. DUOMENŲ BAZĖS STRUKTŪRA | 45 |
| 21 PAV. OBJEKTŲ KLASIFIKAVIMO ALGORITMAS | 47 |
| 22 PAV. OBJEKTŲ PAIEŠKOS ALGORITMAS | 48 |
| 23 PAV. PRINCIPINĖ UNIVERSALIOS VAIZDO ATPAŽINIMO SISTEMOS SCHEMA | 51 |
| 24 PAV. SERVERIO IŠVESTIES PAVYZDINIS VAIZDAS | 52 |
| 25 PAV. WINDOWS KLIENTO NAUJO OBJEKTO ĮVEDIMO LANGAS | 52 |
| 26 PAV. KLIENTO, SKIRTO ANDROID OS, NAUJO OBJEKTO ĮVEDIMO LANGAS | 53 |
| 27 PAV. TESTUOJAMI OBJEKTAI | 54 |
| 28 PAV. EKSPERIMENTO TVARKA | 55 |

Lentelių sąrašas

| | |
|---|----|
| 1 LENTELĖ. SKIRTINGŲ ALGORITMŲ GREIČIŲ PALYGINIMAI | 19 |
| 2 LENTELĖ. SIUNTIMO LAIKŲ MATAVIMAS..... | 21 |
| 3 LENTELĖ. JSON FORMATO CPU IR ATMINTIES NAUDOJIMAS | 21 |
| 4 LENTELĖ. XML FORMATO CPU IR ATMINTIES NAUDOJIMAS..... | 22 |
| 5 LENTELĖ. SKIRTINGŲ BIBLIOTEKŲ PALYGINIMAS | 25 |
| 6 LENTELĖ. SOCKETSERVICE KOMPONENTO METODŲ APRAŠYMAS | 33 |
| 7 LENTELĖ. MYSQLSERVICE KOMPONENTO METODŲ APRAŠYMAS | 34 |
| 8 LENTELĖ. ACTIONS PAKETO METODŲ APRAŠYMAS..... | 36 |
| 9 LENTELĖ. OBJECTRECOGNITION KOMPONENTO METODŲ APRAŠYMAS | 37 |
| 10 LENTELĖ. MAIN KOMPONENTO METODŲ APRAŠYMAS | 38 |
| 11 LENTELĖ. LIVEIMAGEPROCESSINGPACKAGE KOMPONENTO METODŲ APRAŠYMAS..... | 40 |
| 12 LENTELĖ. CLIENTOBJECTRECOGNITIONPACKAGE KOMPONENTO METODŲ APRAŠYMAS | 42 |
| 13 LENTELĖ. MAINPACKAGE KOMPONENTO METODŲ APRAŠYMAS | 43 |
| 14 LENTELĖ. EKSPERIMENTO REZULTATŲ APIBENDRINIMAS | 57 |
| 15 LENTELĖ. MOBILIOJO TELEFONO HTC EVO 3D PARAMETRAI | 70 |
| 16 LENTELĖ. LENTELĖS OBJECT STRUKTŪRA | 71 |
| 17 LENTELĖ. LENTELĖS KEYPOINTS STRUKTŪRA | 72 |
| 18 LENTELĖ. LENTELĖS OBJECTTYPE STRUKTŪRA..... | 72 |
| 19 LENTELĖ. LENTELĖS BOOKS STRUKTŪRA..... | 72 |
| 20 LENTELĖ. LENTELĖS CUPS STRUKTŪRA | 73 |
| 21 LENTELĖ. LENTELĖS TOYS STRUKTŪRA | 73 |
| 22 LENTELĖ. NAUDOTOS PROGRAMINĖS ĮRANGOS PARAMETRAI | 73 |
| 23 LENTELĖ. EKSPERIMENTAI SU HESIANO SLENKSČIU 5000 | 74 |
| 24 LENTELĖ. EKSPERIMENTAI SU HESIANO SLENKSČIU 3000 | 75 |

1 ĮVADAS

Šiuolaikiniame pasaulyje vis aktualesnės tampa vaizdo atpažinimo technologijos. Jos yra plačiai naudojamos pramonėje, transporte, kariniais tikslais ir vis labiau veržiasi į kasdienį paprastų žmonių gyvenimą.

Pramonėje vaizdų atpažinimo technologijos yra naudojamos norint atpažinti produktų brūkšninius kodus, produktus, turinčius defektų, ir kitur. Oro uostuose ir viešajame transporte vaizdo technologijos padeda padidinti ir užtikrinti žmonių saugumą, atpažindamos nusikaltėlius ar ieškomus asmenis. Taip pat vaizdo atpažinimo technologijos naudojamos viešajame transporte, siekiant sumažinti avaringumą miestų gatvėse. Vaizdo atpažinimo technologijos montuojamos į karinę techniką, siekiant tiksliai nustatyti ir sunaikinti taikinį, taip išvengiant pašalinių aukų.

Viena didžiausių pasaulio automobilių gamintojų „General Motors“ (JAV), gaminanti „Chevrolet“, „Buick“, „GMC“ ir „Cadillac“ markių automobilius, naudoja ir tyrinėja vaizdo atpažinimo technologijos savo gaminamuose automobiliuose. Jos Elektros ir valdymo integracijos laboratorijos (*Electrical & Controls Integration Lab*) [26] mokslininkai tyrinėja sistemas, kurios ateityje leis sukurti ypatingus automobilių modelius, kurie:

- Niekada nesusidurs;
- Nereikalaus remonto, bus efektyvūs ir draugiški aplinkai;
- Atpažins ir paryškins kelio ženklus, kelio skiriamąsias linijas ir pan.

Šiame darbe yra analizuojama vaizdų atpažinimo sistema, skirta atpažinti aplinkoje esančius objektus. Bus analizuojama kliento – serverio principu veikianti sistema. Mobiliojo telefono klientas atsakingas už unikalių objekto požymių išskyrimą ir bendravimą su serveriu, o serveris – už klientų aptarnavimą, objektų paiešką bei administravimą. Sistema bus bandoma su šiomis objektų klasėmis: knygomis, puodeliais ir žaislais.

Šiame darbe yra nagrinėjamos problemos, kurios atsiranda kuriant vaizdo atpažinimo sistemą, veikiančią kliento – serverio principu. Vaizdo atpažinimo sistema, veikianti kliento - serverio principu, sukelia tokias pagrindines problemas:

- Duomenų perdavimo problema. Bendraujant klientui su serveriu reikia perduoti duomenis iš vienos sistemos dalies į kitą.
- Duomenų saugojimo problema. Perduodamus duomenis gali reikėti išsaugoti vėlesniam panaudojimui.
- Nuotraukos apdorojimo algoritmo problema. Reikia pasirinkti tinkamą algoritmą nuotraukų apdorojimui.

- Sistemos naudojamų bibliotekų (*library*) problema. Reikia pasirinkti tinkamas bibliotekas, kad būtų galima greičiau ir efektyviau realizuoti sistemą.

2 ANALITINĖ DALIS

2.1 Vaizdų atpažinimo algoritmai

Objektų atpažinimo algoritmai yra skirstomi į kelias pagrindines grupes: briaunų aptikimo, „naudingų taškų“ aptikimo, nekintančių savybių aptikimo (*affine invariant feature detection*). Visos šios algoritmų grupės turi savų privalumų ir trūkumų, todėl dažniausiai naudojamos šių algoritmų kombinacijos ar modifikacijos.

Siekiant patobulinti objektų atpažinimo algoritmus beveik kiekvienas algoritmas turi savas modifikacijas, leidžiančias pagerinti kiekvieno algoritmo savybes, veikiant specifinėmis sąlygomis.

2.2 Briaunų aptikimo algoritmai

Pagrindinis briaunų aptikimo algoritmo tikslas yra aptikti paveiksliuke vietas (briaunas), kuriose ryškumas (*brightness*) ryškiai kinta arba turi ryškių pertrūkimų (*discontinuity*). Klasikinis briaunų aptikimo pavyzdys galėtų būti paveiksliukas „Lena“ (1 pav.), naudojamas briaunų aptikimo bandymams (2 pav.).



1 pav. Originalus paveiksliukas „Lena“



2 pav. Paveiksluko „Lena“ briaunos

2.2.1 Gauso funkcija (Gaussian function)

Gauso funkcija:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (2.1.),$$

kur $a, b, c > 0$, $e \approx 2,718281828$.

Pirmoji Gauso funkcijos išvestinė:

$$f(x) = \frac{-x}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (2.2.)$$

2.2.2 Gradientų aptikimas

Vaizdų atpažinime yra dirbama su skaitmeniniais paveikslukais ir yra apdorojami diskretūs duomenys [1]. Analizuojant paveiksluką yra tikrinami pilkos spalvos pokyčiai tarp gretimų dviejų pikselių. Kadangi paveikslukai yra dvimačiai (2D), šie pokyčiai turi būti išanalizuoti x ir y ašių kryptimis.

Paveiksluko I visų taškų intensyvumo skirtumas yra ekvivalentiškas sąsūkai (*convolution*) paveiksluko I su atitinkamu filtro branduoliu k . Šis branduolys:

$$k_x = [-1, 0, 1] \quad (2.3.)$$

Paveikslukuose būna triukšmų. Norint sumažinti triukšmų trikdžius, filtro branduoliai gali būti padidinti, siekiant tolygaus duomenų apdoravimo. Reikia turėti omeny, kad filtro branduolio dydis įtakoja skaičiavimų greitį. Briaunų aptikimas yra vienas pradinių žingsnių, apdorojant paveikslukus.

Sobelio filtro branduolys yra geras derinys tarp greičio ir duomenų tolygumo:

$$k_{S,x} = \frac{1}{4} \cdot \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, k_{S,y} = \frac{1}{4} \cdot \begin{bmatrix} -1 & 2 & 1 \\ 0 & 0 & 2 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.4.)$$

Paveiksluko I sąskaitai su $k_{S,x}$ ir $k_{S,y}$ yra x ir y gradientai I_x ir I_y :

$$I_x = I * k_{S,x}, I_y = I * k_{S,y} \quad (2.5)$$

Alternatyvus I_x ir I_y yra gradient dydis I_G ir orientacija I_θ . Dėl greičio, I_G yra dažnai aproksimuojama kaip dydžių suma:

$$I_G = |I_x| + |I_y|, I_\theta = \arctan\left(\frac{I_x}{I_y}\right) \quad (2.6.)$$

2.2.3 Kany briaunų aptikėjas

Vienas populiariausių briaunų aptikimo algoritmas yra 1986-ais metais Džono Kany (John Canny) [2] paskelbtas Kany briaunų aptikėjas (*Canny Edge Detector*). Kany suformulavo kelis kriterijus, kurie leidžia paspartinti briaunų aptikimą:

- Geras aptikimas (*good detection*). Turi būti maža tikimybė, jog nepavyks teisingai pažymėti realius briaunų taškus ar kad neteisingai bus pažymėti ne briaunų taškai.
- Gera lokalizacija (*good localization*). Pažymėti briaunų taškai turi būti kiek galima arčiau tikros briaunos centro.
- Tik vienas atsakas į vieną briauną (*only one response to a single edge*). Vienas tikras briaunos pikselis turi nurodyti tik vieną briaunos pikselį. Atstumas tarp išrinkto briaunos pikselio turi būti maksimalus.

Kany parodė, kad pirmo lygio Gauso išvestinė yra gera optimalaus sprendinio aproksimacija aukščiau apibrėžtiems kriterijams. Paveiksluko I sąsūka su šiuo filtru atlieka tolygumo ir gradiento skaičiavimo operacijas tuo pačiu metu. Kadangi yra dirbama su dvimačiu vaizdu, tad dvi sąsūkos turi būti atliktos: x ašiai ir y ašiai.

Kita šių filtrų savybė yra ta, kad jie yra geras našumo ir greičio derinys. I_x ir I_y gali būti apskaičiuojamos kaip:

$$g_x(x, y) = \sqrt{2\pi}\sigma \cdot \frac{\delta g(x)}{\delta x} \cdot g(y) \quad (2.7.)$$

$$g_y(x, y) = \sqrt{2\pi}\sigma \cdot \frac{\delta g(y)}{\delta y} \cdot g(x) \quad (2.8.)$$

Reikia turėti omenyje, jog reikšmė $\sqrt{2\pi}\sigma$ tarnauja kaip kompensuojantis dydis, kuris sumažina išvestinės amplitudės sumažėjimą dėl padidėjusio σ . Reikšmių iš g_x ir g_y paėmimas veda prie branduolių $k_{C,x}$ ir $k_{C,y}$.

Kany filtras susideda iš šių žingsnių:

1. Tolygaus gradiento apskaičiavimo. Tam suskaičiuojama I_G reikšmė (panaudojant formulę 2.6), panaudojant paveiksluką I ir $k_{C,x}$ ir $k_{C,y}$.
2. Ne maksimalaus nuslėpimo. Norint gauti unikalius atsakus kiekvienam tikram briaunos taškui, I_G turi būti dar kartą suskaičiuotas prieš pradžia, kadangi

išlyginimas išsklaido gradiento reikšmes. Pabaigoje visų viršūnių, kurios neturi maksimalios reikšmes gradiento kryptimi, dydis yra sumažinamas (t.y. nustatomas į nulį). Tai gali būti pasiekama analizuojant kaimyninių 3 x 3 pikselių plotelių apie kiekvieną tašką p . Pirmiausia identifikuojami du kaimyniniai pikseliai, kurie yra arčiausiai taško p . Jeigu vienas taškas turi didesnę gradiento reikšmę negu taškas p , tai p dydis yra nustatomas į nulį.

3. Slenksčio išskyrimo. Paskutiniame žingsnyje yra pikseliai yra suskirstomi į „briaunų“ ir „ne briaunų“, remiantis jų gradiento reikšme. Labiau tikėtina, kad pikseliai su didesne gradiento reikšme yra briaunos taškai. Vietoje vieno slenksčio yra naudojami du. Pirmiausia visi pikseliai, kurių gradiento dydis yra didesnis negu t_h yra iškarto klasifikuojami kaip „briaunos“. Šie pikseliai yra naudojami kaip tarpinės antram žingsniui: visi šiems pikseliams gretimi pikseliai yra taip pat laikomi „briaunų“ pikseliais, jeigu jų gradiento dydis yra didesnis už t_i . Šis procesas yra kartojamas tol, kol daugiau nerandami jokie papildomi pikseliai.

2.2.4 „Naudingų taškų“ aptikimo algoritmai

Kampų aptikimo algoritmų tikslas yra aptikti paveiksliuke esančius kampus. Aptikus kampus galima supaprastinti paveiksliukų apdorojimo procesą, kadangi taip gaunama pradinė informacija apie paveiksliuką ir jame esančius objektus: jeigu paveiksliuke esančių objektų kampai nesutampa su jau mums žinomų objektų kampais, galime daryti prielaidą, jog šis objektas mums dar yra nepažįstamas ir nežinomas.

Yra du plačiausiai naudojami kampų aptikimo algoritmai: Hariso (*Harris*) ir SUSAN.

2.2.4.1 Hariso kampų aptikimo algoritmas

Krisas Harisas (*Chris Harris*) ir Maikas Stefensas (*Mike Stephens*) (kai kurioje literatūroje nurodomas ir trečias autorius Plesis (*Plessey*) savo straipsnyje [3] pasiūlė naują kampų ir briaunų aptikimo metodą. Šis metodas remiasi vietine auto koreliacijos funkcija. Hariso kampų aptikimo algoritmas yra plačiai naudojamas dėl jo nekintamumo pasukimui (*rotation*), mastelio kitimui (*scale*), apšvietimo pokyčiams (*illumination variation*) ar paveiksliuko „triukšmams“ (*image noise*) [4]. Nors šis algoritmas pasiūlytas 1986-iais metais, tačiau jis iki šiol laikomas vienu geriausių ir plačiausiai naudojamų.

Robertas Kolinsas (*Robert Collins*) savo paskaitoje [5] pateikia Hariso kampų aptikimo algoritmą:

1. Paskaičiuoti paveiksliuko I_x ir I_y išvestines:

$$I_x = G_\sigma^x * I, I_y = G_\sigma^y * I \quad (2.9)$$

2. Paskaičiuoti reikšmes kiekvienam pikseliui:

$$I_{x2} = I_x \cdot I_x, I_{y2} = I_y \cdot I_y, I_{xy} = I_x \cdot I_y \quad (2.10)$$

3. Paskaičiuoti išvestinių sumas kiekvienam pikseliui:

$$S_{x2} = G_{\sigma'} * I_{x2}, S_{y2} = G_{\sigma'} * I_{y2}, S_{xy} = G_{\sigma'} * I_{xy} \quad (2.11)$$

4. Kiekvienam pikseliui (x, y) sudaryti matricą

$$H(x, y) = \begin{bmatrix} S_{x2}(x, y) & S_{xy}(x, y) \\ S_{xy}(x, y) & S_{y2}(x, y) \end{bmatrix} \quad (2.12)$$

5. Suskaičiuoti detektoriaus R atsaką kiekvienam pikseliui:

$$R = Det(H) - k (Trace(H))^2, kur k = 0,04 - 0,06 \quad (2.13)$$

6. Suskaičiuoti R slenkstį (*threshold*).

2.2.4.2 SUSAN kampų aptikimo algoritmas

Stefanas Smitas (*Stephen M. Smith*) ir Michaelis Bredy (*J. Michael Brady*) aprašė SUSAN (*Smallest Univalve Segment Assimilating Nucleus*) [6] kampų aptikimo algoritmą. Nors šis algoritmas pasiūlytas 1997-ais metais, tačiau jis iki šiol yra plačiai naudojamas dėl savo gerų veikimo charakteristikų ir greičio.

SUSAN algoritmas susideda iš kelių žingsnių. Turime paveiksluką I.

1. Kiekvieno pikselio ryškumas (*brightness*) palyginamas su centrinio pikselio ryškumu:

$$c(\vec{r}, \vec{r}_0) = \begin{cases} 1, & \text{jei } |I(\vec{r}) - I(\vec{r}_0)| \leq t \\ 0, & \text{priešingu atveju} \end{cases} \quad (2.14)$$

kur \vec{r}_0 - centrinio taško padėtis, \vec{r} - bet kurio kito taško padėtis, $I(\vec{r})$ - taško ryškumas, $I(\vec{r}_0)$ - centrinio taško ryškumas, t - ryškumo slenkstis, c - palyginimo rezultatas.

2. Šie palyginimai yra padaromi kiekvienam pikseliui ir iš visų jų padaroma n :

$$n(\vec{r}_0) = \sum_{\vec{r}} c(\vec{r}, \vec{r}_0) \quad (2.15)$$

3. Nustatoma g reikšmė:

$$g = \frac{3n_{max}}{4} \quad (2.16)$$

kur n_{max} yra didžiausia įgyjama n reikšmė.

4. SUSAN algoritmo reikšmė gaunama:

$$R(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0), & \text{jei } n(\vec{r}_0) < g \\ 0, & \text{priešingu atveju} \end{cases} \quad (2.17)$$

Kaip rašo algoritmo autoriai, šis algoritmas duoda gerus rezultatus, tačiau geriau vartoti šią išraišką, vietoje aprašytos (2.13):

$$c(\vec{r}, \vec{r}_0) = e^{-\left(\frac{I(\vec{r}) - I(\vec{r}_0)}{t}\right)^6} \quad (2.18)$$

2.2.5 Nekintančių savybių aptikimo algoritmai

Nekintančių savybių algoritmo pavyzdys yra SURF (*Speeded Up Robust Features*) algoritmas. SURF algoritmas veikia išskirdamas pagrindinius taškus (*salient points*) iš paveiksluko ir suskaičiuoja aplink pagrindinius taškus esančių taškų deskriptorius (*descriptor*), kurie nekinta keičiant mastelį, objekto pasukimą ar apšvietimą [7]. Kaip rašo algoritmo autoriai, tyrimų metu nustatyta, kad SURF algoritmas yra labai našus [8].

SURF algoritmas susideda iš keturių pagrindinių dalių:

1. Integralinio paveiksluko generavimo.
2. Greito Hesiano detektoriaus (*Fast-Hessian detector*) „įdomių taškų aptikimo“.
3. Deskriptoriaus orientacijos priskyrimo (nebūtina).
4. Deskriptoriaus generavimo.

$$I_{\Sigma}(x, y) = \sum_{i=0}^x \sum_{j=0}^y I(i, j) \quad (2.19)$$

Integralinis paveikslukas (2.19) yra naudojamas algoritmo veikimo paspartinimui. Kai naudojamas integralinis paveikslukas, būtina perskaityti keturias pikselių reikšmes paviršiaus integralo skaičiavimui bet kokio dydžio paveikslukams.

$$H(x, y) = \det \begin{pmatrix} \frac{\delta^2 f}{\delta x^2} & \frac{\delta^2 f}{\delta x \delta y} \\ \frac{\delta^2 f}{\delta x \delta y} & \frac{\delta^2 f}{\delta y^2} \end{pmatrix} \quad (2.20)$$

$$H(\bar{x}) = D_{xx}(\bar{x})D_{yy}(\bar{x}) - (0,9D_{xy}(\bar{x}))^2, \bar{x} = (x, y, s) \quad (2.21)$$

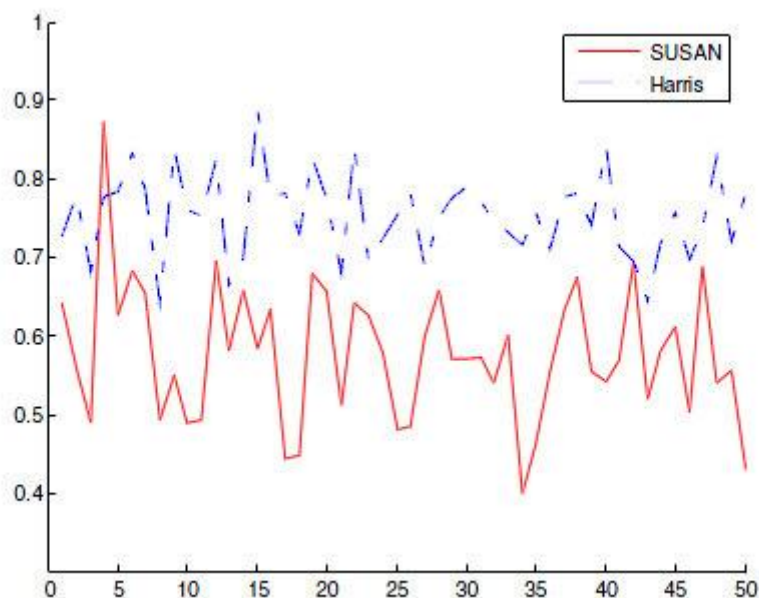
Šio algoritmo veikimo rezultatas yra seka taškų ir deskriptorių.

2.3 Algoritmų palyginimai

2.3.1 Hariso ir SUSAN algoritmų palyginimas

Autoriai [9] palygino Hariso ir SUSAN algoritmus esant skirtingoms veikimo sąlygoms. Būtina paminėti, kad ir Hariso, ir SUSAN algoritmai tenkina Kany iškeltas sąlygas geram briaunų aptikimui (skyrius 2.2.3).

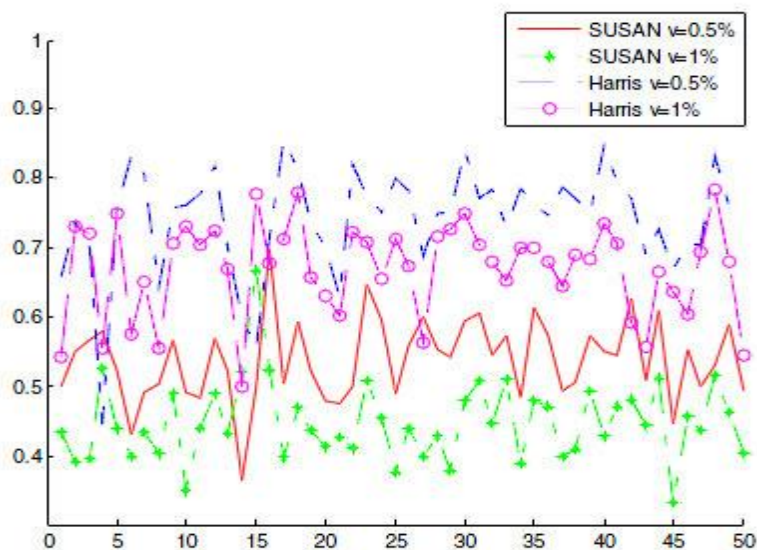
Straipsnyje yra analizuojami Hariso ir SUSAN algoritmai stabilumo, atsparumo triukšmams, sudėtingumo ir veikimo greičio atžvilgiu. Taip pat buvo analizuojama, kaip algoritmai atpažįsta paveikslukus.



3 pav. Stabilumo analizė

Analizuojant stabilumą (3 pav.) buvo apskaičiuota, kad Hariso algoritmo vidutinis stabilumas yra 75,41%, o SUSAN algoritmo – 57,82%. Skirtumas yra apie 25%.

Analizuojant triukšmus (4 pav.) buvo pasirinktas Gauso baltas triukšmas, kintantis nuo 0,5% iki 1%. Hariso algoritmas veikia geriau negu SUSAN algoritmas. Tačiau autoriai pažymi, jog jų abiejų našumas yra žemesnis, kai padidėja triukšmo faktorius.



4 pav. Atsparumas triukšmams

Stabilumo analizei buvo pasirinktas 320 x 240 pikselių dydžio paveiksliukas. SUSAN algoritmas vidutiniškai veikė 4,2125 s, o Hariso – 1,8686 s. Hariso algoritmas yra žymiai greitesnis (maždaug 2,5 karto).

2.3.2 SURF, Hariso, Gauso, Hesiano – Laplaso algoritmų palyginimas

Buvo bandomi apžiūros taško (*viewpoint*), didinimo ir posūkio (*rotation*), bei apšvietimo (*lighting*) pasikeitimai.

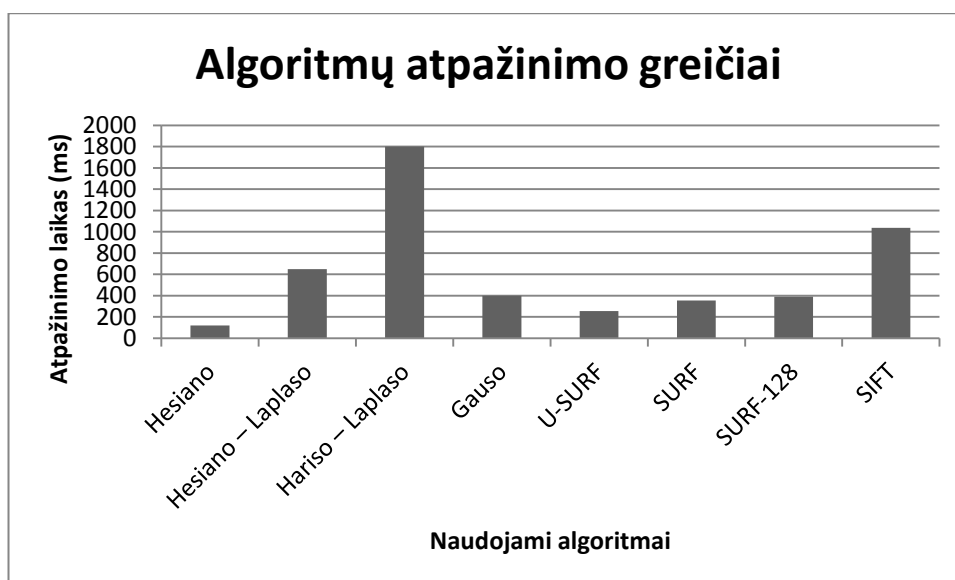
Greitasis Hesiano (*Fash-Hessian*) algoritmas yra daugiau kaip 3 kartus greitesnis už Gauso algoritmą, bei 5 kartus greitesnis už Hesiano – Laplaso (*Hessian-Laplace*) algoritmą.

Lyginant SURF algoritmą su analogiškais „nekintančių savybių aptikimo algoritmais“ GLOH, SIFT ir PCA-SIFT, buvo pastebėta, kad SURF algoritmas veikia greičiau. Kaip pažymi SURF algoritmo autoriai, SURF algoritmas veikia žymiai greičiau negu analogiški algoritmai (kartais net daugiau kaip 10%), esant tam pačiam tikslumui.

Buvo analizuojama, per kiek laiko greitasis Hesiano, Hesiano – Laplaso, Hariso – Laplaso, Gauso bei U-SURF, SURF, SURF-128 bei SIFT algoritmai atpažįsta paveiksliuką (1 lentelė, 5 pav.):

1 lentelė. Skirtingų algoritmų greičių palyginimai

| Algoritmas | Atpažinimo laikas (ms) |
|--------------------------|------------------------|
| Hesiano | 120 |
| Hesiano – Laplaso | 650 |
| Hariso – Laplaso | 1800 |
| Gauso | 400 |
| U-SURF | 255 |
| SURF | 354 |
| SURF-128 | 391 |
| SIFT | 1036 |



5 pav. Atpažinimo laikas

Kaip matome, greičiausias yra Hesiano algoritmas, SURF algoritmas nuo jo nedaug atsilieka.

2.4 Informacijos perdavimo protokolo formatas

Duomenų keitimuisi tarp kelių bendradarbiaujančių sistemų reikalingas sutartas duomenų apsikeitimo formatas. Šiuo metu yra populiariausi du formatai: XML [10] ir JSON [11].

XML formatas yra Standartizuotos apibendrintos žymėjimo kalbos (*Standard Generalized Markup Language, SGML*) poaibis (*subset*). XML formato didžiausias privalumas yra jo universalumas - šią kalbą lengvai galima pritaikyti įvairiems tikslams [12].

JSON formatas yra „lengvas“ (*lightweight*) duomenų apsikeitimo formatas, lengvai suprantamas žmonių ir lengvai apdorojamas informacinių sistemų. Kompiuterinės sistemos gali lengvai apdoroti JSON formato dokumentus.

2.4.1 Tyrimo aprašymas

Autoriai [13] lygina XML ir JSON duomenų formatus. Jie analizuoja, kaip efektyviai keičiamasi duomenimis naudojant JSON ir XML formatus.

Buvo testuojama kliento - serverio architektūros principu veikianti sistema. Klientas siųsdavo žinutes į serverį, o serveris jas priimdavo. Testų metu buvo fiksuojama:

- Nusiųstų objektų skaičius (20 000, 40 000, 60 000, 80 000, 100 000);
- Laikas nusiųsti visiems objektams;
- Vidutinis vieno objekto siuntimo laikas;

- Kliento CPU panaudojimas;
- Serverio CPU panaudojimas;
- Atminties panaudojimas.

2.4.2 Tyrimo rezultatai

Matuojant siuntimo laikus buvo gauti šie rezultatai (2 lentelė):

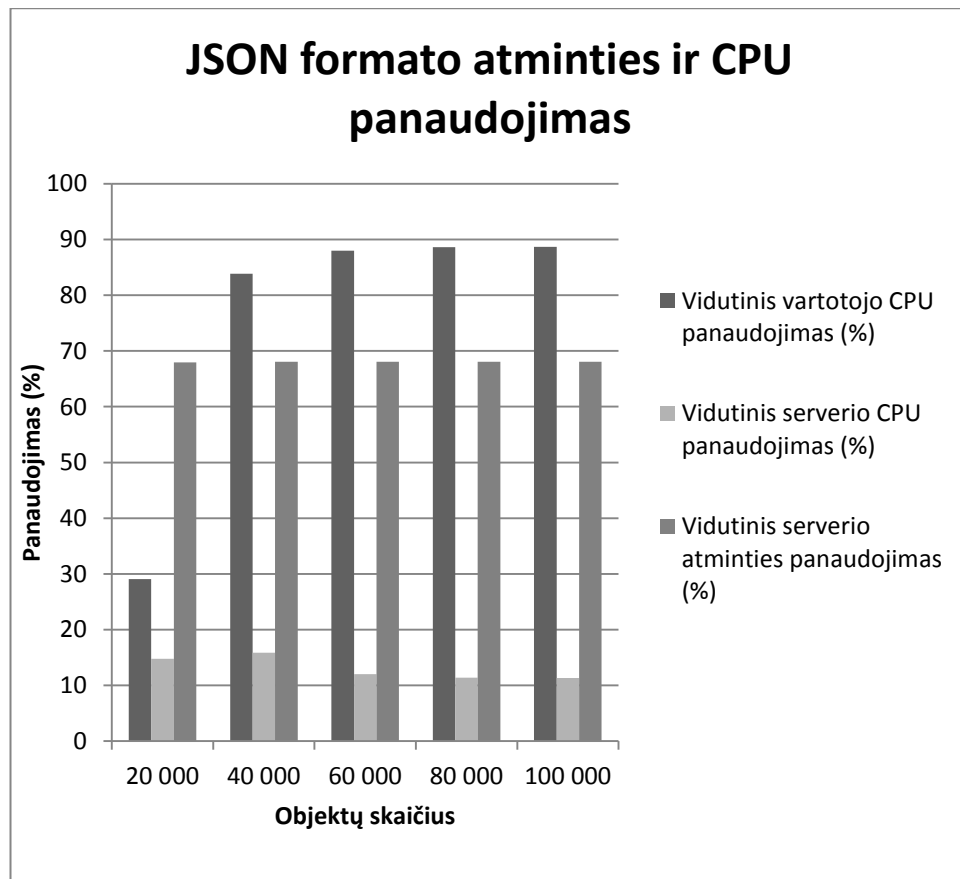
2 lentelė. Siuntimo laikų matavimas

| | JSON | XML |
|---------------------------------|---------|-----------|
| Testas 1, objektų skaičius | 20 000 | 20 000 |
| Testas 1, visas laikas (ms) | 2213,15 | 61333,68 |
| Testas 1, vidutinis laikas (ms) | 0,11 | 3,07 |
| Testas 2, objektų skaičius | 40 000 | 40 000 |
| Testas 2, visas laikas (ms) | 3127,99 | 123854,59 |
| Testas 2, vidutinis laikas (ms) | 0,08 | 3,10 |
| Testas 3, objektų skaičius | 60 000 | 60 000 |
| Testas 3, visas laikas (ms) | 4552,38 | 185936,27 |
| Testas 3, vidutinis laikas (ms) | 0,08 | 3,10 |
| Testas 4, objektų skaičius | 80 000 | 80 000 |
| Testas 4, visas laikas (ms) | 6006,72 | 247639,81 |
| Testas 4, vidutinis laikas (ms) | 0,08 | 3,10 |
| Testas 5, objektų skaičius | 100 000 | 100 000 |
| Testas 5, visas laikas (ms) | 7497,36 | 310017,47 |
| Testas 5, vidutinis laikas (ms) | 0,07 | 3,10 |

Matuojant CPU ir atminties panaudojimą buvo gauti šie rezultatai (3 lentelė, 6 pav.):

3 lentelė. JSON formato CPU ir atminties naudojimas

| Objektų skaičius | Vidutinis vartotojo CPU panaudojimas (%) | Vidutinis serverio CPU panaudojimas (%) | Vidutinis serverio atminties panaudojimas (%) |
|------------------|--|---|---|
| 20 000 | 29,07 | 14,80 | 67,97 |
| 40 000 | 83,84 | 15,84 | 68,07 |
| 60 000 | 88,01 | 11,99 | 68,06 |
| 80 000 | 88,65 | 11,36 | 68,06 |
| 100 000 | 88,70 | 11,30 | 68,06 |



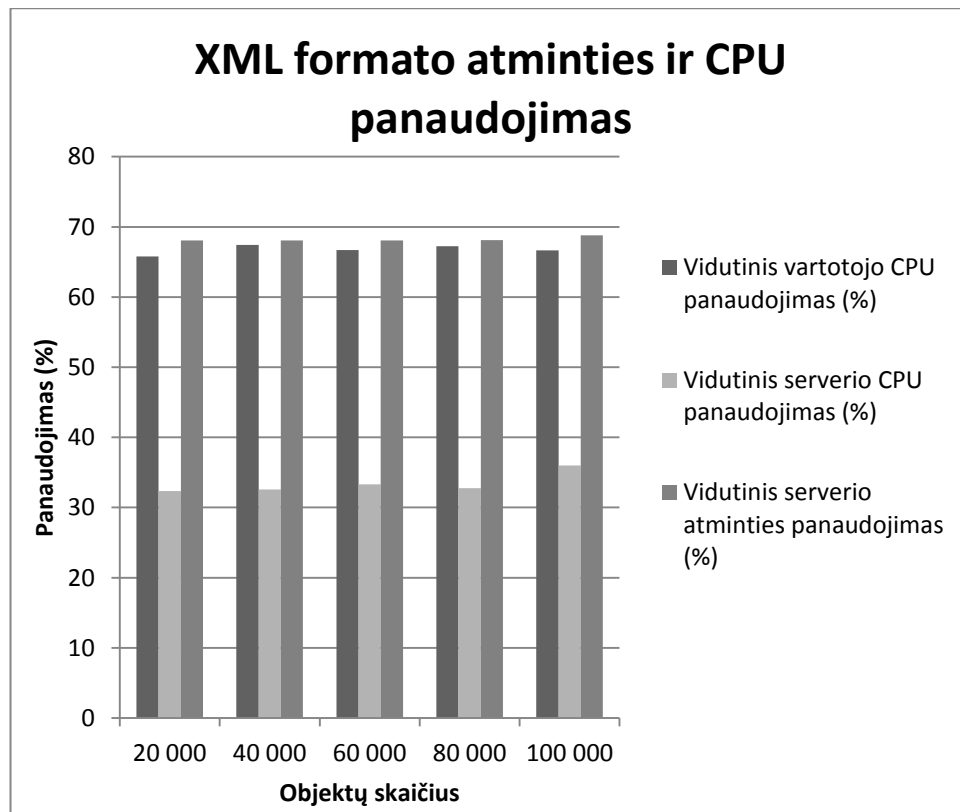
6 pav. JSON formato atminties ir CPU panaudojimas

Kaip matome, naudojant JSON formatą, atminties ir CPU panaudojimas lieka pastovus ir nepriklauso nuo objektų skaičiaus.

Testuojant bandomą XML formatą buvo gauti šie rezultatai (4 lentelė, 7 pav.):

4 lentelė. XML formato CPU ir atminties naudojimas

| Objektų skaičius | Vidutinis vartotojo CPU panaudojimas (%) | Vidutinis serverio CPU panaudojimas (%) | Vidutinis serverio atminties panaudojimas (%) |
|------------------|--|---|---|
| 20 000 | 65,80 | 32,36 | 68,08 |
| 40 000 | 67,43 | 32,57 | 68,08 |
| 60 000 | 66,69 | 33,31 | 68,08 |
| 80 000 | 67,24 | 32,76 | 68,11 |
| 100 000 | 66,64 | 36 | 68,79 |



7 pav. XML formato atminties ir CPU panaudojimas

Kaip matome, naudojant XML formatą, atminties ir CPU panaudojimas taip pat lieka pastovus ir nepriklauso nuo objektų skaičiaus. Tačiau resursų panaudojimas yra didesnis, negu naudojant JSON.

Išvados:

1. Duomenų siuntimas JSON formatu yra greitesnis negu naudojant XML.
2. Siunčiant mažus duomenų paketus XML formatu, jų siuntimo laikas yra mažesnis negu JSON.
3. JSON formato siuntimo laikai nedaug priklauso nuo duomenų kiekio.
4. JSON formato objektai naudoja daugiau kliento (vartotojo) CPU laiko negu XML formato žinutės. Tačiau serveryje viskas atvirkščiai.

2.5 Objektų atpažinimo bibliotekos

Objektų atpažinimui ir apdorojimui yra sukurta daug bibliotekų:

- *OpenCV* [14].
- *Insight Segmentation and Registration Toolkit (ITK)* [15].
- *Integrating Vision Toolkit (IVT)* [16].
- *Visualization Toolkit (VTK)* [17].

2.5.1 OpenCV biblioteka

OpenCV (*OpenSource Computer Vision*) yra biblioteka, skirta realaus laiko vaizdų apdorojimui ir kompiuterinio matymo (*computer vision*) realizavimui. OpenCV suteikia lengvai naudojamą kompiuterinio matymo infrastruktūrą, kartu su didele biblioteka (daugiau kaip 500) algoritmų ir funkcijų, kurios pritaikytos realaus laiko kompiuteriniam matymui. Ši biblioteka plačiai naudojama visame pasaulyje, ji buvo atsisiųsta daugiau kaip 3 305 000 kartų. Biblioteka yra labai plačiai naudojama, pradedant nuo interaktyvaus meno, baigiant kasyklų analizėmis ir aukšto lygio robotų pramone.

OpenCV yra parašyta panaudojant optimizuotą C/C++ kodą, veikia Windows, Linux, Mac OS X, Android OS operacinėse sistemose. Biblioteka yra nemokama komerciniam ir nekomerciniam naudojimui pagal BSD licenciją.

2.5.2 Insight Segmentation and Registration Toolkit (ITK)

ITK yra biblioteka, skirta paveikslukų segmentavimui ir registravimui. Pagrindinės šios bibliotekos savybės:

- Skirta dirbti su medicininiais duomenimis.
- Yra bibliotekos versijos C++, Tcl, Python, Java programavimo kalboms.
- Veikia Windows, Linux, Mac OS X operacinėse sistemose.
- Naudoja CMake kompiliavimo konfigūravimui.
- Palaiko daugiagijį apdorojimą (*mult-threaded processing*).
- Skirtas Matomo žmogaus projekto (*Visible Human Project*) [30] nuotraukų apdorojimui.

ITK pradedant nuo 3.6 versijos yra platinama pagal BSD licenciją. 4.0 programos versija yra platinama pagal Apache 2.0 licenciją.

2.5.3 Integrating Vision Toolkit (IVT)

IVT yra biblioteka, skirta paveikslukų apdorojimui. Pagrindinės šios bibliotekos savybės:

- Turi realizuotą Hariso kampų detektorius.
- Parašyta C++ programavimo kalba.
- Yra realizuotas SIFT algoritmas.
- Gali segmentuoti paveikslukus pagal spalvas.
- Veikia Windows, Linux, Mac OS X operacinėse sistemose.
- Turi realizuotą K - D medžio algoritmą.

IVT platinama pagal atvirojo kodo BSD licenciją.

2.5.4 Visualization Toolkit (VTK)

VTK yra biblioteka, skirta 3D kompiuterinei grafikai, paveiksliukų apdorojimui ir vizualizavimui. Jos pagrindinės savybės:

- Turi realizacijas C++, Tcl, Perl, Python, Java programavimo kalbomis.
- Realizuoti daugiakampių sumažinimo, tekstūravimo metodai.
- Leidžia lygiagrečiai apdoroti paveiksliukus.
- Veikia Windows, Linux, Mac OS X operacinėse sistemose.

VTK platinama pagal atvirojo kodo BSD licenciją.

2.5.5 Bibliotekų palyginimas

5 lentelė. Skirtingų bibliotekų palyginimas

| | OpenCV | ITK | IVT | VTK |
|--|--------|-----|-----|-----|
| Turi realizaciją C++ programavimo kalba | + | + | + | + |
| Veikia Linux, Windows, Mac OS X operacinėse sistemose | + | + | + | + |
| Turi realizuotą SURF algoritimą | + | – | – | – |
| Leidžia nuskaityti vaizdą iš kameros | + | – | + | – |

Iš apžvelgtų bibliotekų (5 lentelė) labiausiai tinkama yra OpenCV, kuri turi realizuotą SURF algoritimą, taip pat leidžia nuskaityti vaizdą iš kameros.

2.6 SIFT ir SURF deskriptorių palyginimas

Siekiant paspartinti vaizdo atpažinimui reikalingus skaičiavimus stengiamasi naudoti kuo mažiau dimensijų turinčius deskriptorius [27]. Deskriptoriai turi būti nepriklausomi nuo analizuojamo objekto:

1. Pasukimo;
2. Apšvietimo;
3. Atstumo pokyčių.

Nors SIFT deskriptorius vienas populiariausių [28], tačiau jo didelis dimensijų kiekis (128-D) lemia, kad šis deskriptorius ne itin tinkamas mobiliuose prietaisuose veikiančioms ir/arba didelę duomenų bazę turinčioms sistemoms. Naudojant SURF deskriptorių [29] pasiekiamas geresnis teisingų atpažinimų skaičius ir trumpesnis skaičiavimų laikas, lyginant su SIFT ar jo modifikacijomis, taigi vartotojui greičiau bei tiksliau pateikiama su objektu susieta virtuali informacija.

2.7 Išvados

Atlikus literatūros analizę buvo padarytos šios išvados:

1. Nei vienas iš apžvelgtų algoritmų negali klasifikuoti objektų arba tam nėra pritaikytas.
2. Objektų atpažinimo algoritmai gali aptikti objektus, nustatyti unikalias objekto savybes, išskirti objektą aprašančią informaciją ir ją panaudoti konkretaus objekto atpažinimui.
3. Apžvelgti algoritmai negali išskirti charakteringų objektų savybių, kurias būtų galima panaudoti klasifikuojant objektus į klases (tipus).
4. SURF algoritmas yra tinkamesnis objektų atpažinimui (ypač realizuojant klientą mobiliajame telefone) negu SIFT, nes jis tiksliau ir greičiau atpažįsta objektus.
5. Yra tikslinga ne realizuoti SURF algoritmą, bet panaudoti šio algoritmo realizaciją OpenCV bibliotekoje.
6. Informacijos perdavimui tarp kliento ir serverio yra tikslinga naudoti JSON formatą, o ne XML.

3 PROJEK TINĖ DALIS

3.1 Projekto tikslas

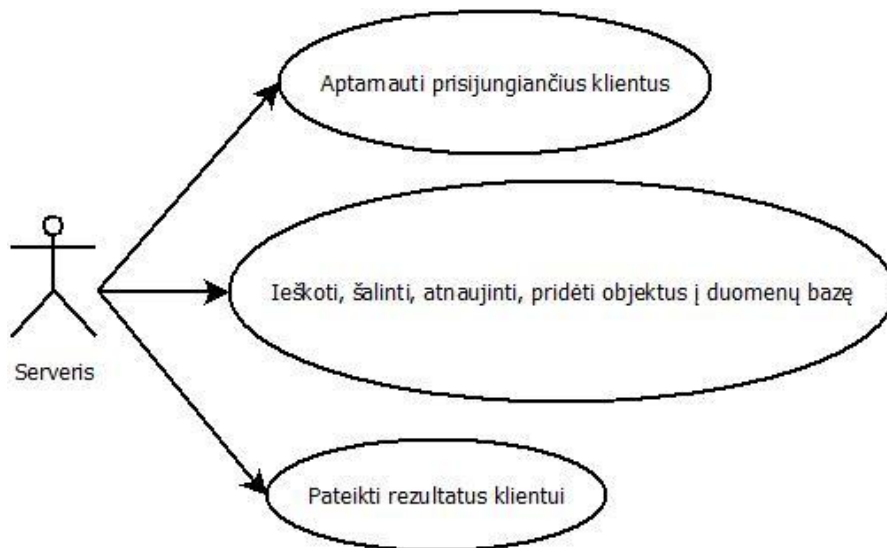
Projekto tikslas yra sukurti vaizdo atpažinimo sistemą, kurios pagrindinės funkcijos būtų šios:

- Įvesti naują objektą į sistemą (pavyzdžiui, knygą).
- Išsaugoti objektą duomenų bazėje.
- Atpažinti seniau įvestą objektą.
- Redaguoti (atnaujinti, pašalinti) jau egzistuojantį objektą.
- Parodyti klientui operacijų rezultatus.

3.2 Reikalavimai sistemai ir sistemos funkcijos

Esminiai reikalavimai sistemai yra šie:

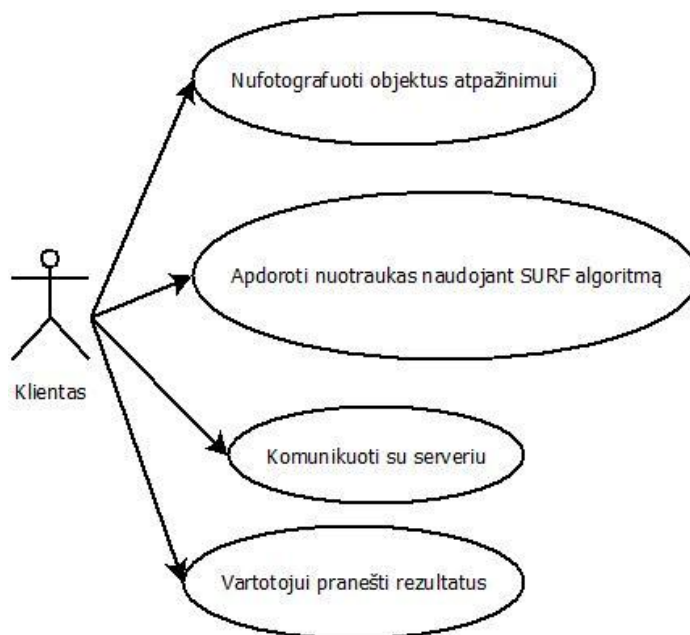
- Turi veikti naudodama kliento - serverio architektūrą.
- Turi turėti klientus Windows, Linux, Android OS operacinėms sistemoms.
- Nuotraukų apdorojimui turi naudoti SURF algoritimą.
- Objektų informaciją sistema turi išsaugoti duomenų bazėje.
- Sistema duomenų apsikeitimui tarp kliento ir serverio turi naudoti JSON formatu užkoduotus duomenis.
- Sistema turi taupiai naudoti mobilaus telefono baterijos energiją.
- Sistema turi galėti klasifikuoti objektus į klases, pavyzdžiui, „knygos“, „obuoliai“, „riešutai“ ir pan.
- Sistemos reakcijos laikas - ne ilgiau 5 sekundžių.
- Sistema turi naudoti atviro kodo ir nemokamą programinę įrangą.
- Perduodant duomenis tinklu turi naudoti kiek galima mažiau tinklo srauto.



8 pav. Serverio panaudos atvejų diagrama

Serverio (8-as paveikslukas) pagrindinės funkcijos yra šios:

- Aptarnauti prisijungiančius klientus.
- Ieškoti, šalinti, atnaujinti, pridėti objektus į duomenų bazę.
- Pateikti rezultatus klientui.



9 pav. Klientui panaudos atvejų diagrama

Kliento (9 pav.) pagrindinės funkcijos yra šios:

- Nufotografuoti objektus atpažinimui.
- Apdoroti nuotraukas naudojant SURF algoritmą.
- Komunikuoti su serveriu.

- Klientui pranešti rezultatus.

3.3 Panašūs projektai

Yra keletas panašių projektų, skirtų objektų atpažinimui ir nuotraukų analizei.

3.3.1 Google Googles

Kompanijos „Google“ sukurta nemokama „Google Googles“ [18] programa, veikianti Android OS ir iOS operacinėse sistemose. Ši programa yra skirta ieškoti panašių paveiksliukų internete ir pateikti šių paveiksliukų aprašymus.

Programos savybės:

- Žyminių kodų (*barcodes*) paieška.
- QR kodų paieška.
- Populiarių aplinkos vietų (*landmarks*) paieška.
- Užsienio kalbomis parašyto teksto vertimas.
- Teksto analizavimas, panaudojant OCR.

Programa dirba su šiomis objektų klasėmis:

- Tekstu.
- Prekių ženklais.
- Knygomis.
- Vizitinių kortelių kontaktine informacija.
- Populiariomis aplinkos vietomis.

Ši programa nėra objektų atpažinimo programinis sprendimas. Tai yra paieškos pagal paveiksliuką programinis sprendimas.

3.3.2 GemIdent

Stenfordo universiteto sukurta „GemIdent“ [19] programa, kurios pagrindinė paskirtis yra aptikti išsiskiriančius regionus paveiksliukuose. Pagrindinės šios programos savybės yra šios:

- Spalvų parinkimas.
- Apmokymas pagal šablonus.
- Klasifikavimas.

Ši programa yra parašyta naudojantis Java programavimo kalba. Ji neturi kliento Android OS operacinei sistemai.

3.3.3 Bartendroid

Lundo universiteto magistro studijų studentų Sebastiano Olssono ir Philipo Akessono sukurta programa „Bartendroid“ [20] yra skirta nufotografuoti ir nuotraukoje aptikti alkoholinių gėrimų buteliams ir, panaudojant šią informaciją, vartotojui pasiūlyti optimalų gėrimų derinį, kad vartotojas galėtų sau pasidaryti kokteilį. Pagrindinės šios sistemos savybės:

- Sistema nuotraukų apdorojimui naudoja SURF algoritmą.
- Veikia Android OS.
- Sistema veikia kliento - serverio principu.

Pagrindiniai šios programos trūkumai yra du:

- Ribota paskirtis (ji yra skirta dirbti tik su alkoholiniais gėrimais).
- Nufotografuota nuotrauka yra nusiunčiama į serverį apdoroti. Jeigu nenaudojamas *Wi-Fi* tinklas, nuotraukos nusiuntimas gali užimti daug laiko.

3.4 Nuotraukos apdorojimas

Kadangi Universalio vaizdo atpažinimo sistema dirba su nuotraukomis, būtina pasirinkti, kur apdoroti nuotrauką: kliente ar serveryje.

Apdorojant nuotrauką serveryje yra šie privalumai:

- Galima išnaudoti galingesnę serverio procesorių nuotraukų apdorojimui paspartinti.
- Klientas neapkraunamas papildomomis (nuotraukos apdorojimo) užduotimis.

Apdorojant nuotrauką serveryje yra šie trūkumai:

- Dirbant su daug klientų reikės galingo serverio.
- Siunčiant visų klientų nuotraukas į serverį bus išnaudojama daug tinklo srauto.

Apdorojant nuotrauką kliente yra šie privalumai:

- Sutaupoma tinklo srauto, nes tinklu siunčiama tikrai informacija apie požymius, bet ne pati nuotrauka.
- Nuotrauką galima apdoroti esant skirtingoms rezoliucijoms.

Apdorojant nuotrauką kliente yra šie trūkumai:

- Daugiau apkraunamas kliento procesorius.
- Greičiau eikvojama mobiliojo telefono baterijos energija.

Sistemos realizacijai buvo pasirinkta, kad nuotrauka bus apdorojama kliente.

3.5 Sistemos architektūros aprašymas

Sistema sukurta naudojant kliento - serverio architektūrą. Jos veikimo principas:

- Klientas nufotografuoja norimą atpažinti ar išsaugoti objektą naudodamas vaizdo kamerą.

- Objektas yra apdorojamas naudojant SURF algoritmą ir jo duomenys yra serializuojami panaudojant JSON formatą.
- JSON formato žinutė yra pasiunčiamas į Ubuntu Linux 11.10 [21] operacinėje sistemoje veikiančią serverį.
- Šiame serveryje esančio MySQL [22] duomenų bazėje objektas yra surandamas arba išsaugojamas.
- Operacijos rezultatas yra grąžinamas klientui.

3.5.1 Serverio architektūros aprašymas

Serveryje naudojamos šios technologijos:

- MySQL duomenų bazių serveris.
- Ubuntu Linux 11.10 64-ių bitų operacinė sistema.
- OpenCV 2.1 biblioteka.
- MySQL/C++ Connector [23] biblioteka.
- Boost [24] biblioteka.

Pagrindinės serverio atliekamos operacijos yra dvi: prisijungiančių klientų aptarnavimas ir objektų paieška, šalinimas, atnaujinimas bei pridėjimas.

3.5.1.1 *Prisijungiančių klientų aptarnavimas*

Klientas, norėdamas dirbti su sistema, privalo prisijungti prie serverio. Standartiškai serveris veikia naudodamas 10 000-ąją jungtį. Jeigu klientas sėkmingai prisijungia prie serverio, jis gali atlikti objektų paieškos, šalinimo, atnaujinimo ir pridėjimo operacijas.

Klientas su serveriu bendrauja naudodamas JSON formatu užkoduotas žinutes (*messages*). Žinučių formatas ir pavyzdžiai pateikti Priedų 9.1.1 - 9.1.6 skyriuose.

3.5.1.2 *Objektų paieška, šalinimas, atnaujinimas ir pridėjimas*

Klientas norėdamas atlikti paieškos, šalinimo, atnaujinimo arba pridėjimo veiksmą privalo atsiųsti atitinkamo formato žinutę į serverį. Serveryje žinutė yra apdorojama ir atliekamos operacijos duomenų bazėje.

Klientas norėdamas ieškoti arba pridėti naują objektą JSON formato žinutėje atsiunčia objekto tipą (JSON žinutės *data.object-data.type* parametras). Pagal objekto tipą sistema nustato, kokiai kategorijai priklauso objektas.

3.5.2 Kliento aprašymas

Kliente naudojamos šios technologijos:

- OpenCV 2.1 biblioteka.

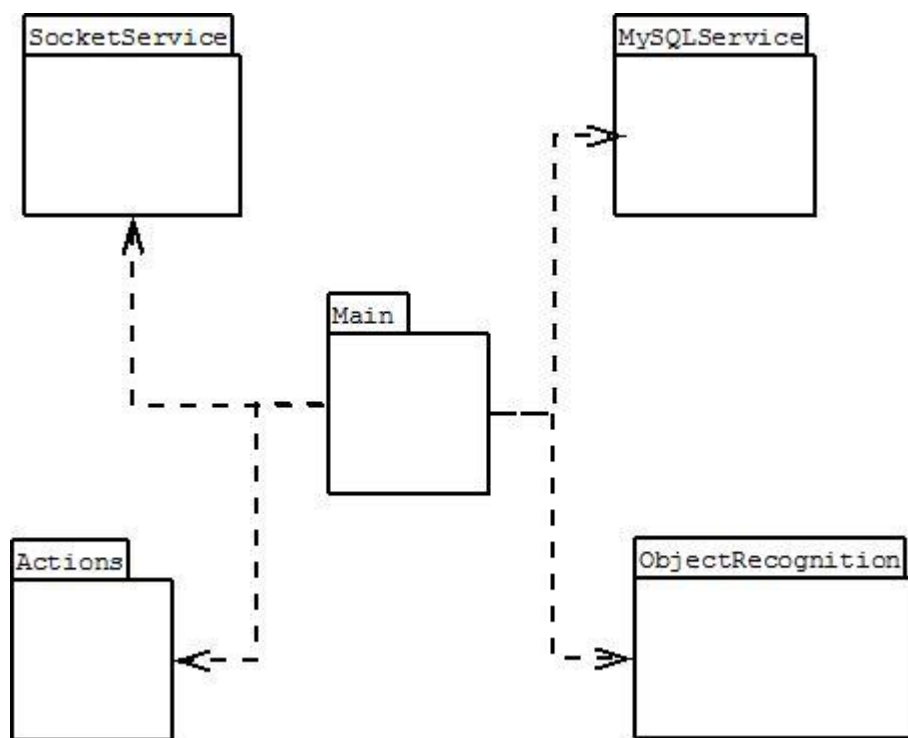
- QT SDK [24] (Windows, Linux klientams).
- Android SDK (Android OS klientui).

Pagrindinės kliento atliekamos operacijos yra objekto pridėjimas, šalinimas, redagavimas bei paieška.

Klientas, nufotografavęs objektą, jo nuotrauką apdoroja su SURF algoritmu, užkoduoja informaciją JSON formatu, persiunčia serveriui. Serveris, gavęs žinutę, atlieka atitinkamas operacijas duomenų bazėje ir pasiunčia klientui atsakymą.

3.6 Serverio architektūra

3.6.1 Serverio komponentų diagramos



10 pav. Bendra sistemos paketų diagrama

Serveris sudarytas iš kelių pagrindinių komponentų (10 pav.):

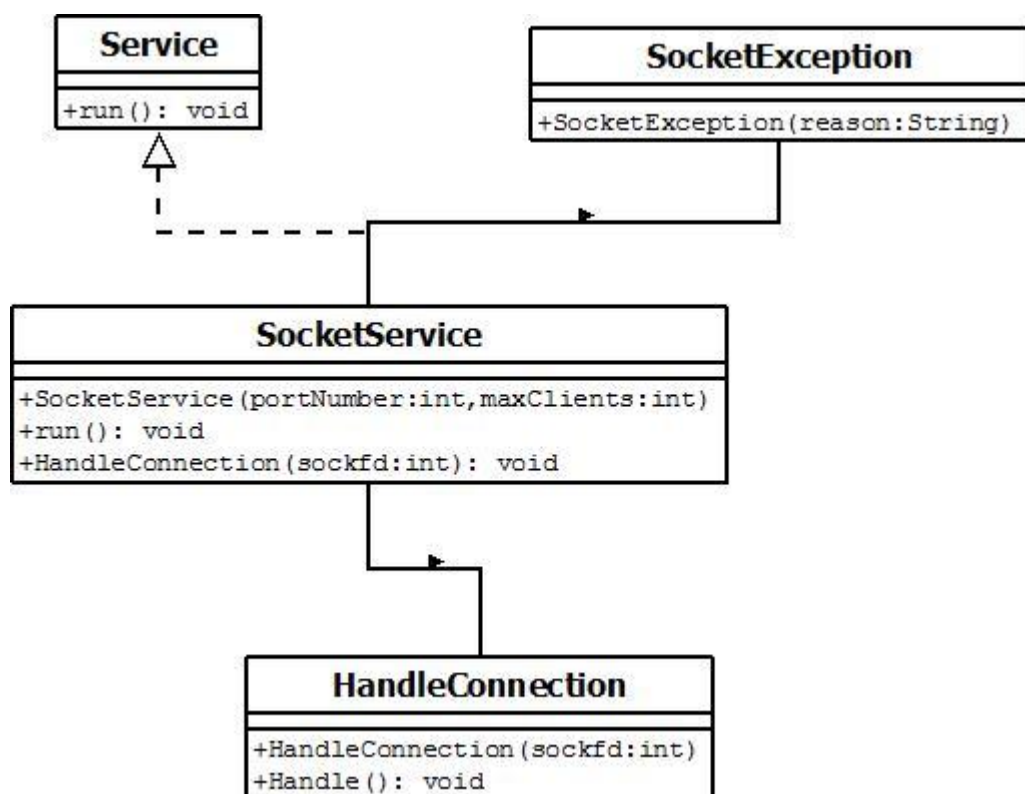
- SocketService. Sistemos komponentas, atsakingas už darbą su tinklu. Naudojantis šiuo komponentu serveris komunikuoja su klientų per TCP/IP protokolą ir 10 000-ąją jungtį (*port*).
- MySQLService. Sistemos komponentas, atsakingas už darbą su duomenų baze. Naudojantis šiuo komponentu serveris atlieka įterpimo (*create*), skaitymo (*read*), atnaujinimo (*update*), pašalinimo (*delete*) operacijas (*CRUD operations*). Šis komponentas naudoja *MySQL/C++ Connector* biblioteką.

- **Actions.** Sistemos komponentas, atsakingas už kliento pasirinktų veiksmų apdorojimą. Informacija apie kliento pasirinktą veiksmą (pridėti, pašalinti, atpažinti, atnaujinti) yra perduodama šiam komponentui, o jis atlieka norimą veiksmą ir klientui praneša veiksmo atlikimo rezultatą - ar veiksmą pavyko atlikti, ar ne.
- **ObjectRecognition.** Sistemos komponentas, atsakingas už objektų atpažinimą. Jame yra atliekamos paveiksliuko atpažinimo ir palyginimo operacijos.
- **Main.** Pagrindinis sistemos komponentas. Jo paskirtis - susieti visus kitus komponentus.

3.6.2 SocketService komponentas

Šio paketo (11 pav., 6 lentelė) paskirtis yra apdoroti prisijungimus prie serverio, naudojant soketus (*socket*). Serveris standartiškai veikia naudodamas 10000 jungtį, tačiau ją galima pakeisti serverio konfigūracijoje. Kadangi serveris yra parašytas naudojant Linux operacinės sistemos, standartiškai yra naudojami UNIX tipo soketai (*socket*).

Klientai su serveriu bendraus panaudodami vieną populiariausių informacijos apsikeitimo formatų - JSON. Žinučių formatai pateikti magistrinio darbo priedų 9.1 punkte.



11 pav. SocketService komponento klasių diagrama

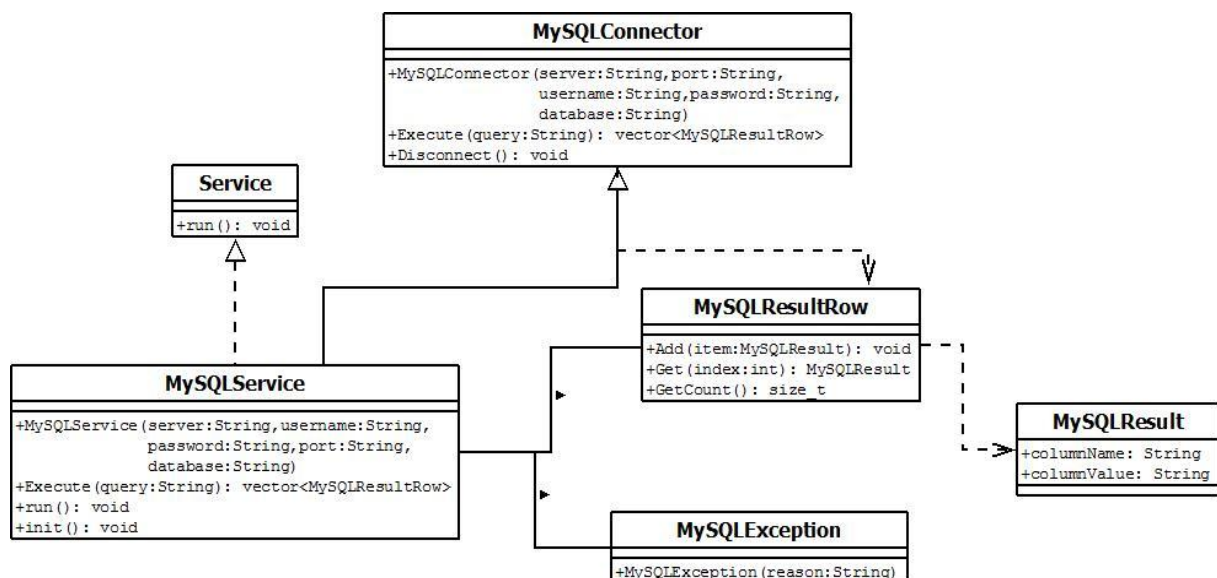
6 lentelė. SocketService komponento metodų aprašymas

| Klasė | Metodas | Aprašymas |
|-------------------------|--|--|
| Service | run() | Atlikti serviso veiksmus |
| SocketException | SocketException(reason:String) | Soketų komponento išimtis |
| SocketService | SocketService(portNumber:int, maxClients:int) | Konstruktorius. Parametrai nurodo, kokią standartinę jungtį naudoti bei kiek gali būti daugiausiai klientų |
| | run() | Paleidžia soketų komponentą, laukia klientų prisijungimų |
| | HandleConnection(sockfd:int) | Prisijungus klientui išskviečiamas metodas, kuris sukuria kliento objektą |
| HandleConnection | HandleConnection(sockfd:int) | Kliento objektas |
| | Handle() | Apdoroja kliento prisijungimą |

3.6.3 MySQLService komponentas

Šio komponento (12 pav., 7 lentelė) paskirtis yra dirbti su MySQL duomenų baze (prisijungimas prie duomenų bazės, atsijungimas, užklausų siuntimas ir gavimas, rezultatų apdorojimas ir pan.) ir suteikti API sąsają kitoms sistemos dalims.

Modulis sudarytas iš kelių dalių, kuriose yra API sąsaja, skirta vartotojui prieiti prie MySQL funkcijų ir jas panaudoti darbui su duomenų baze: objektų pridėjimui, pašalinimui, redagavimui ir paieškai.



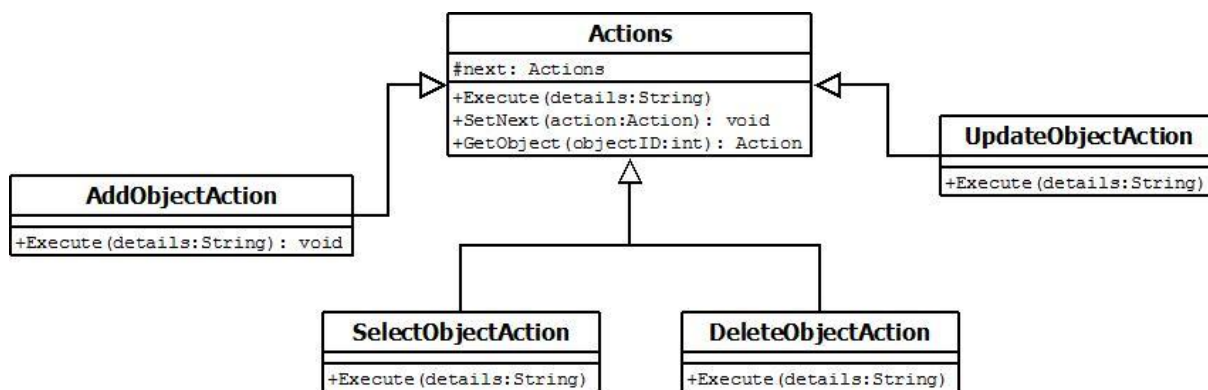
12 pav. MySQLService komponento diagrama

7 lentelė. MySQLService komponento metodų aprašymas

| Klasė | Metodas/kintamasis | Aprašymas |
|-----------------------|---|---|
| Service | run() | Atlikti serviso veiksmus |
| MySQLException | MySQLException(reason:String) | Duomenų bazės komponento išimtis |
| MySQLService | MySQLService(server:String, username:String, password:String, port:String, database:String) | Konstruktorius. Parametrai nurodo serverį, vartotojo prisijungimo duomenis ir duomenų bazę |
| | run() | Paleidžia operacijų su MySQL servišą |
| | Execute(query:String) | Įvykdo užklausą |
| | init() | Inicializuoja servišą - prisijungia prie duomenų bazių serverio, pasiruošia vykdyti užklausas |
| MySQLConnector | MySQLConnector(server:String, username:String, password:String, port:String, database:String) | Konstruktorius. Parametrai nurodo serverį, vartotojo prisijungimo duomenis ir duomenų bazę |
| | Execute(query:String): vector<MySQLResultRow> | Įvykdo užklausą |
| | Disconnect() | Atsijungia nuo duomenų bazių serverio |
| MySQLResultRow | Add(item: MySQLResult) | Prideda užklaustos vykdymo rezultato eilutę į rezultatus |
| | Get(index: int) : MySQLResult | Grąžina užklaustos vykdymo rezultata |
| | GetCount() : size_t | Grąžina užklaustos vykdymo rezultatų skaičių |
| MySQLResult | columnName | Stulpelio reikšmė |
| | columnValue | Stulpelio pavadinimas |

3.6.4 Actions komponentas

Šio komponento (13 pav., 8 lentelė) pagrindinė paskirtis yra atlikti veiksmus su objektais: pridėti, redaguoti ar pašalinti. Vartotojo pasiūsta viena iš JSON formatu pateiktų žinučių Universalios vaizdo atpažinimo sistemos serveriui yra persiunčiama Actions paketo klasėms, kurios apdoroja šį veiksmą: prideda, redaguoja ar pašalina objektą.



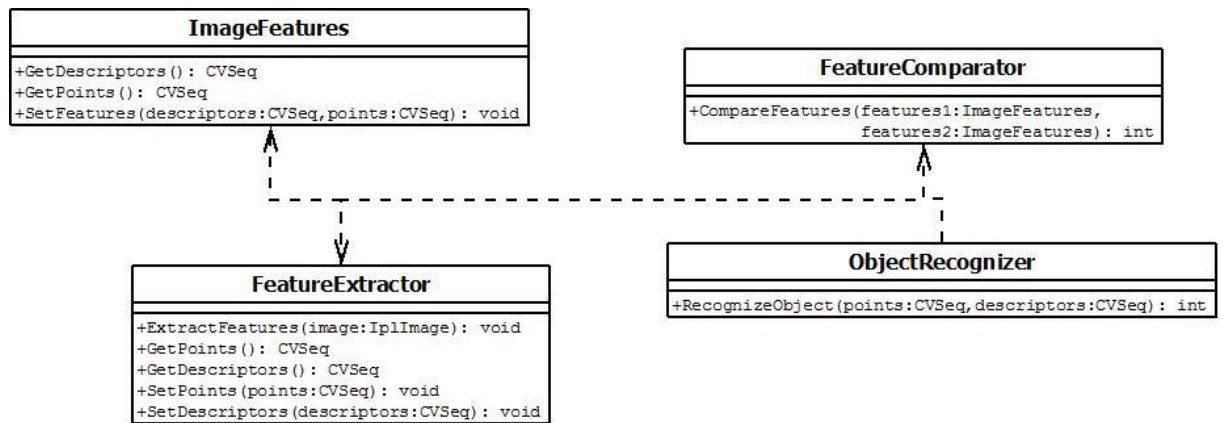
13 pav. Actions komponento paketų diagrama

8 lentelė. Actions paketo metodų aprašymas

| Klasė | Metodas/kintamasis | Aprašymas |
|---------------------------|---------------------------------------|-------------------------------------|
| Actions | next: Actions | Elementų sąrašas |
| | Execute(details: String) | Įvykdo užklausą |
| | SetNext(action: Action) | Nustato sekantį elementą |
| | GetObject(objectID : int) : Action | Grąžina nurodytą sąrašo elementą |
| AddObjectAction | Execute(details: String) | Prideda objektą |
| SelectObjectAction | Execute(details: String) | Suranda objektą |
| DeleteObjectAction | Execute(details: String) | Ištrina objektą |
| UpdateObjectAction | Execute(details: String) | Atnaujina objektą |

3.6.5 ObjectRecognition komponentas

Šio komponento (14 pav., 9 lentelė) paskirtis yra išanalizuoti pateiktus serveriui paveikslėlius ir išskirtų iš jų unikalius taškus bei jų deskriptorius, kuriuos vėliau šis modulis panaudos objektų palyginimui ir atpažinimui. Taip pat šis modulis yra naudojamas paveikslėlių konvertavimui iš spalvotų į nespalvotus.



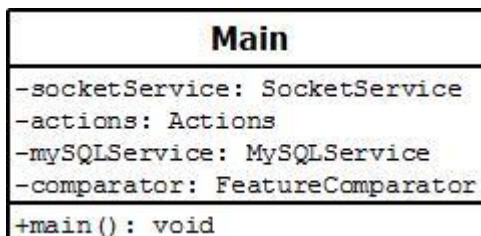
14 pav. ObjectRecognition komponento klasių diagrama

9 lentelė. ObjectRecognition komponento metodų aprašymas

| Klasė | Metodas/kintamasis | Aprašymas |
|--------------------------|---|--|
| ImageFeatures | GetDescriptors() : CVSeq | Gražina deskriptorių sąrašą |
| | GetPoints() : CVSeq | Gražina taškų sąrašą |
| | SetFeatures(descriptors: CVSeq, points: CVSeq) | Nustato taškus ir taškų deskriptorius |
| FeatureComparator | CompareFeatures(features1: ImageFeatures, features2: ImageFeatures) : int | Palygina du paveikslukus. Gražina jų panašumą |
| FeatureExtractor | ExtractFeatures(image: IplImage) | Iš paveiksluko išskiria jį aprašančius taškus ir deskriptorius |
| | GetPoints() : CVSeq | Gražina taškus |
| | GetDescriptors() : CVSeq | Gražina deskriptorius |
| | SetPoints(points: CVSeq) | Nustato taškus |
| ObjectRecognizer | SetDescriptors(descriptors: CVSeq) | Nustato deskriptorius |
| | RecognizeObject(points: CVSeq, descriptors: CVSeq) : int | Atlieka objektų atpažinimą. Gražina objektų panašumą |

3.6.6 Main komponentas

Šio komponento (15 pav., 10 lentelė) paskirtis yra sujungti visus komponentus į vieną visumą.



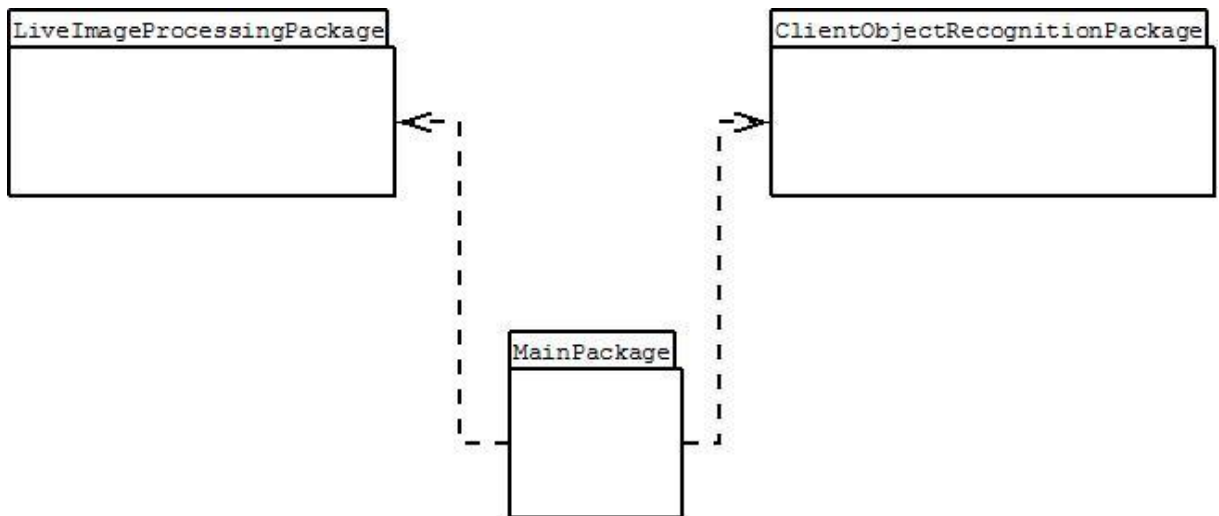
15 pav. Main komponento klasių diagrama

10 lentelė. Main komponento metodų aprašymas

| Klasė | Metodas/kintamasis | Aprašymas |
|-------|-------------------------------|---|
| Main | socketService: SocketService | Soketų komponento objektas |
| | actions: Actions | Veiksmų komponento objektas |
| | mySQLService: MySQLService | Duomenų bazių komponento objektas |
| | comparator: FeatureComparator | Objektų palyginimo komponento objektas |
| | main() | Pagrindinis metodas, paleidžiantis visą sistemą |

3.7 Kliento architektūra

3.7.1 Kliento komponentų diagrama



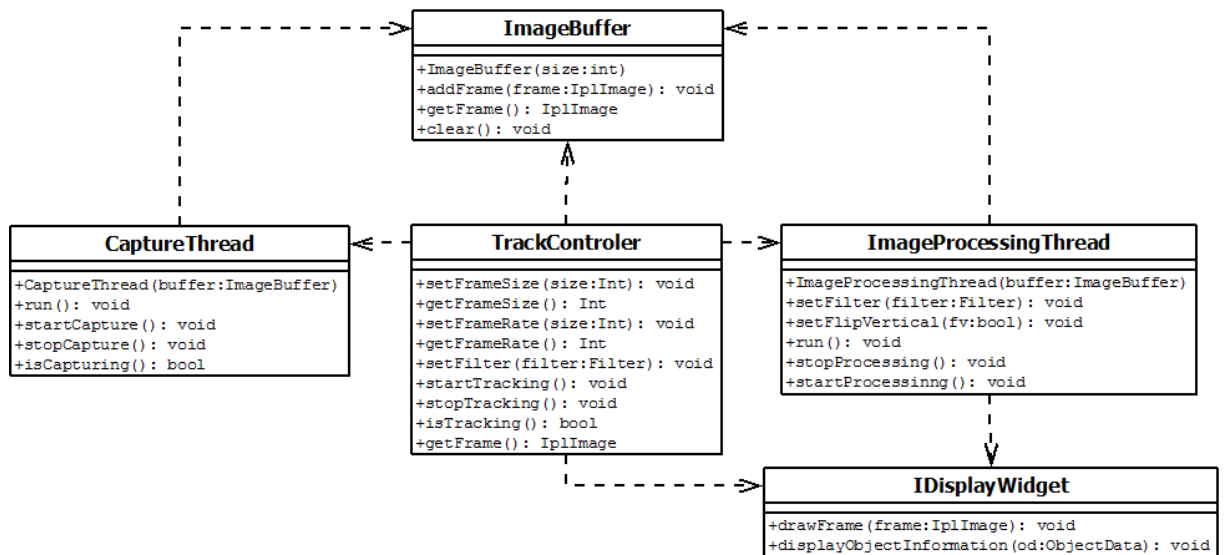
16 pav. Kliento pagrindinė komponentų diagrama

Klientas (16 pav.) sudarytas iš kelių pagrindinių komponentų:

- LiveImageProcessingPackage. Sistemos komponentas, atsakingas už iš kameros paimto vaizdo atvaizdavimą ekrane.
- ClientObjectRecognitionPackage. Sistemos komponentas, atsakingas už iš kameros paimtos nuotraukos apdorojimą.
- MainPackage. Sistemos komponentas, atsakingas už kliento sistemos paleidimą ir kliento sistemos veikimą.

3.7.2 LiveImageProcessingPackage komponento diagrama

Šiame pakete (17 pav., 11 lentelė) esančios klasės naudojamos paimti kameros vaizdą ir jį atvaizduoti ekrane. Už kameros vaizdo paėmimą ir jo atvaizdavimą ekrane bus atsakingos dvi klasės – CaptureThread ir ImageProcessingThread. Sąsają (*interface*) IDisplayWidget realizuojanti klasė bus atsakinga už iš kameros paimto vaizdo apdorojimą (spalvoto kadro pavertimą juodai baltu (*grayscale*), pasukimą, apvertimą ir pan.) bei atvaizdavimui ekrane.



17 pav. LiveImageProcessingPackage komponento diagrama

11 lentelė. LiveImageProcessingPackage komponento metodų aprašymas

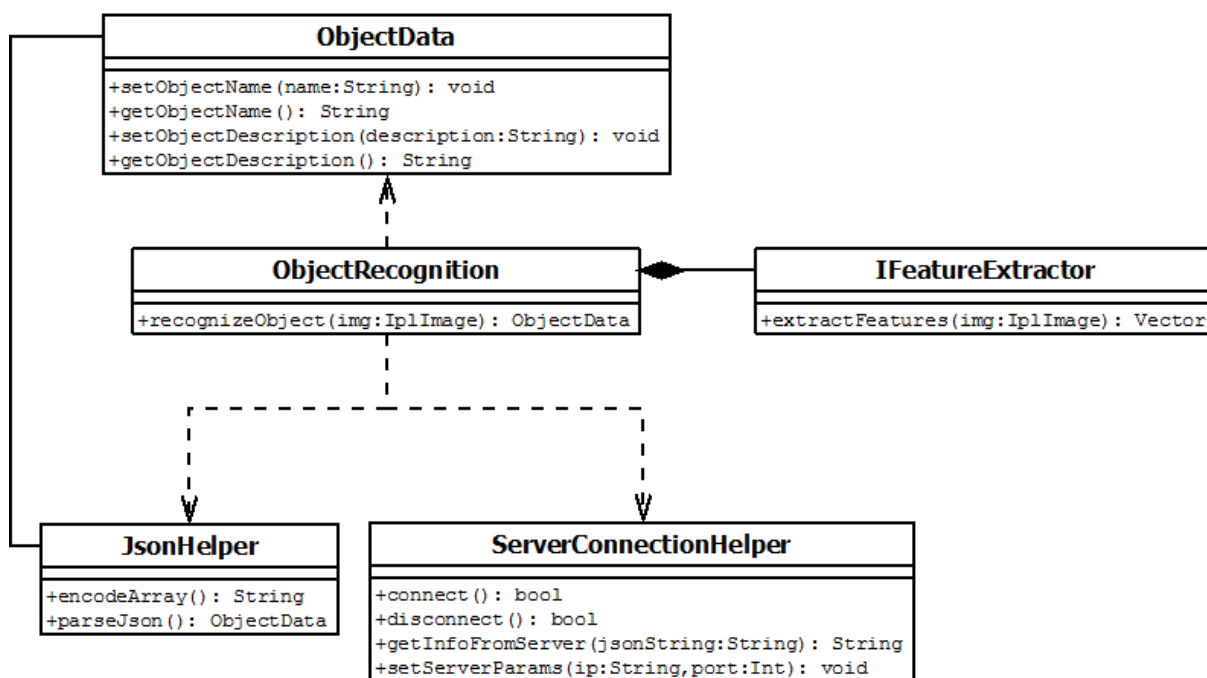
| Klasė | Metodas/kintamasis | Aprašymas |
|------------------------|------------------------------------|---|
| ImageBuffer | ImageBuffer(size: int) | Klasės konstruktorius. |
| | addFrame(frame: IplImage) | Įdeda kadra į buferį. |
| | getFrame() : IplImage | Grąžina naujausią kadra iš buferio. |
| | clear() | Išvalo buferį. |
| CaptureThread | CaptureThread(buffer: ImageBuffer) | Klasės konstruktorius. |
| | run() | Metodas, kviečiamas gijos veikimo metu. Jei klasės būseną nustatyta į „CAPTURING“, ima iš kameros vieną kadra ir patalpina jį buferyje. |
| | startCapture() | Inicijuoja vaizdo ėmimą iš kameros. |
| | stopCapture() | Stabdo vaizdo ėmimą iš kameros. |
| | isCapturing() | Grąžina informaciją, ar gija ima kadrus iš kameros ir rašo į buferį. |
| TrackController | setFrameSize(size: int) | Nustato kadro dydį. |
| | getFrameSize() : int | Grąžina kadro dydį. |

| Klasė | Metodas/kintamasis | Aprašymas |
|------------------------------|--|---|
| | setFrameRate(size: int) | Nustato kadru dažnį. |
| | getFrameRate() : int | Grąžina kadru dažnį. |
| | setFilter(filter: Filter) | Nustato filtrą |
| | startTracking() | Pradedama imti vaizdą iš kameros, jį atvaizduoti ekrane. Jei objektas atpažintas, aktyvuoja sekimo algoritmą. |
| | stopTracking() | Nustoja imti vaizdą iš kameros, jį atvaizduoti ekrane. |
| | isTracking() : bool | Grąžina, ar objektas yra sekamas. |
| | getFrame() : IplImage | Grąžina naujausią buferio kadrą. |
| ImageProcessingThread | ImageProcessingThread(buffer: ImageBuffer) | Klasės konstruktorius. |
| | setFilter(filter: Filter) | Nustato filtrą. |
| | setFlipVertical(fv: bool) | Apverčia paveikslėlį. |
| | run() | Sukūrus giją, ima kadrą iš buferio ir kviečia IDisplayWidget sąsajos metodą drawFrame(). |
| | stopProcessing() | Pradedama paveikslėlio apdorojimą. |
| | startProcessing() | Stabdo paveikslėlio apdorojimą. |
| IDisplayWidget | drawFrame(frame: IplImage) | Atvaizduoti objektą ekrane. |
| | displayObjectInformation(od: ObjectData) | Pateikia informaciją apie objektą ekrane. |

3.7.3 ClientObjectRecognitionPackage komponento diagrama

Šis komponentas (18 pav., 12 lentelė) skirtas iš kameros paimto kadro apdorojimui, nuotraukoje esančių objekto požymių išskyrimui bei kreipimuisi į serverį. Tėvinę klasę IFeatureExtractor paveldi ObjectRecognition paketo klasę, realizuojančią konkretų požymių išskyrimo algoritmą.

Klientas serveriui iš nuotraukos išskirtus objekto požymius perduoda JSON formatu. Iš serverio gautą informaciją apie atpažintą objektą enkapsuliuoja klasė ObjectData, kuri vėliau naudojama duomenų apie atpažintą objektą pateikimui vartotojui.



18 pav. ClientObjectRecognitionPackage komponento diagrama

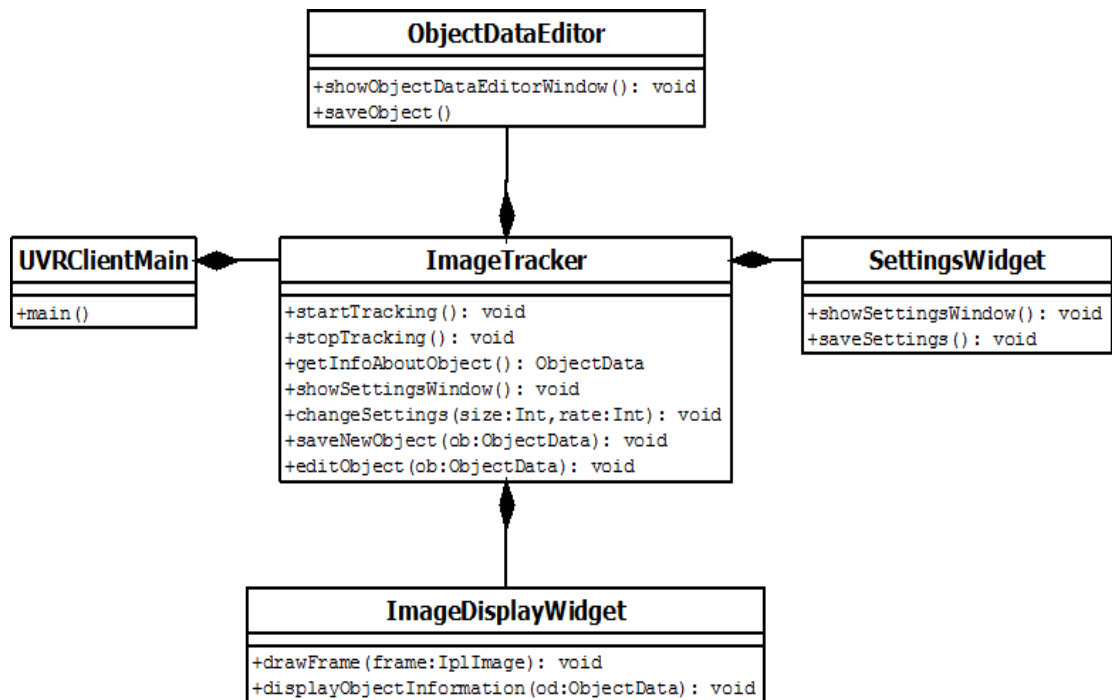
12 lentelė. ClientObjectRecognitionPackage komponento metodų aprašymas

| Klasė | Metodas/kintamasis | Aprašymas |
|--------------------------|---|--|
| ObjectData | setObjectName(name: String) | Nustato objekto vardą. |
| | getObjectName() : String | Grąžina objekto vardą. |
| | setDescription(description: String) | Nustato objekto aprašymą. |
| | getDescription() : String | Grąžina objekto aprašymą. |
| ObjectRecognition | recognizeObject(img: IplImage) : ObjectData | Išskiria norimo atpažinti objekto požymius, gauna reikiamą JSON bei asinchroniškai kreipiasi į serverį. Naudoja atitinkamai FeatureExtractor, JSONHelper ir ServerConnectionHelper klasių metodus. |
| IFeatureExtractor | extractFeatures(img: IplImage) : Vector | Išskiria objekto požymius. |
| JsonHelper | encodeArray() : String | Masyvą perkoduoja JSON formatu. |
| | parseJson() : ObjectData | JSON formato masyvą |

| Klasė | Metodas/kintamasis | Aprašymas |
|-------------------------------|---|--|
| | | perkoduoja į masyvą. |
| ServerConnectionHelper | connect() : bool | Prisijungia prie serverio. |
| | disconnect() : bool | Atsijungia nuo serverio |
| | getInfoFromServer(jsonString : String) : String | Gauna atsakymą iš serverio. |
| | setServerParams(ip: String, port : int) | Nustato prisijungimo prie serverio parametrus. |

3.7.4 MainPackage komponento diagrama

Šiame pakete (19 pav., 13 lentelė) esančios klasės realizuoja kliento vartotojo sąsają bei koordinuoja kitus kliento dalies objektus.



19 pav. MainPackage komponento diagrama

13 lentelė. MainPackage komponento metodų aprašymas

| Klasė | Metodas/kintamasis | Aprašymas |
|-------------------------|------------------------------|---|
| ObjectDataEditor | showObjectDataEditorWindow() | Parodo objekto informacijos redagavimo langą. |
| | saveObject() | Išsaugo naują informaciją. |
| UVRClientMain | main() | Paleidžia programą. |

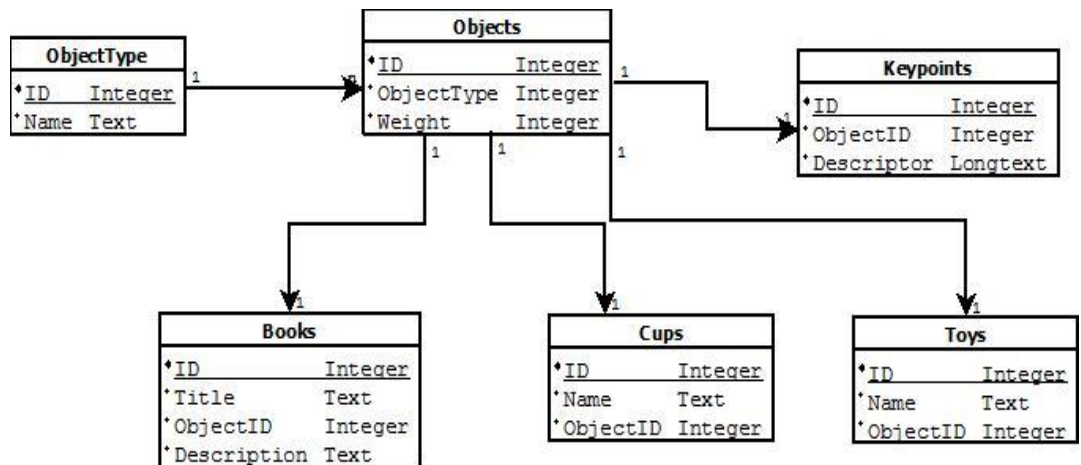
| Klasė | Metodas/kintamasis | Aprašymas |
|---------------------------|---|--|
| SettingsWidget | showSettingsWidget() | Parodo nustatymų langą. |
| | saveSettings() | Validuoja ir išsaugo nustatymus. |
| ImageTracker | startTrackgin() | Pradedą iš kameros imamą vaizdą vaizduoti ekrane. |
| | stopTracking() | Sustabdo iš kameros paimto vaizdo pateikimą ekrane (sustabdo kadą). |
| | getInfoAboutObject() : ObjectData | Apdoroja objekto atpažinimo mygtuko nuspaudimą. |
| | showSettingsWindow() | Parodo nustatymų koregavimo langą. |
| | changeSettings(size : int, rate: int) | Pakeičia programos nustatymus. |
| | saveNewObject(ob: ObjectData) | Apdoroja naujo objekto išsaugojimo mygtuko nuspaudimą. |
| | editObject(ob: ObjectData) | Apdoroja objekto informacijos redagavimo išsaugojimo mygtuko nuspaudimą. |
| ImageDisplayWidget | drawFrame(frame: IplImage) | Piešia kadą ekrane. |
| | displayObjectInformation(ob: ObjectData) | Piešia informaciją apie objektą. |

3.8 Duomenų bazės struktūra

Sistemai veikiant kliento - serverio principu, sistemos duomenims būtina duomenų saugojimo vieta. Kaip duomenų bazių valdymo sistema buvo pasirinkta MySQL duomenų bazių valdymo sistema, kadangi:

- Ji veikia Linux OS.
- Yra nemokama.
- Sistemos kūrėjai turi patirties dirbant su šia duomenų bazių valdymo sistema.
- MySQL duomenų bazių valdymo sistemos kūrėjai pateikia nemokamą MySQL/C++ Connector biblioteką darbui su MySQL duomenų baze.

Sistemos duomenys yra saugomi reliacinėje duomenų bazėje, kurios apibendrinta duomenų bazės lentelių schema yra pateikta 20-ame paveiksliuke.



20 pav. Duomenų bazės struktūra

Detalesnis sistemos duomenų bazės lentelių aprašymas yra pateiktas Priedo 9.4 punkte.

3.9 Sistemos įdiegimas

Sistema buvo pristatyta 2012 m. KTU Technoramos parodoje, kurioje vienintelė iš KTU Informatikos fakulteto darbų gavo rektoriaus padėką. Sistema įdiegta UAB „Informacinių technologijų organizacija“ ir VĮ „Registų centras“.

3.10 Išvados

Apibendrinant galima teigti, jog:

1. Sistema gali įvesti, redaguoti, šalinti ir atpažinti objektus.
2. Pasirinktas SURF algoritmas yra tinkamas objektų atpažinimo sistemai realizuoti.
3. JSON formato naudojimas padeda sutaupyti tinklo srauto (palyginus su XML). Taip pat užtikrina perduodamų duomenų integralumą.
4. OpenCV bibliotekos naudojimas padėjo ženkliai sumažinti sistemos kūrimui skirtą laiką.
5. Nufotografuotą objekto nuotrauką apdorojant kliente, o ne serveryje, yra sutaupoma tinklo srauto. Tačiau dėl intensyvaus mobiliojo telefono procesoriaus naudojimo yra greičiau eikvojama mobiliojo telefono baterija.
6. Bandant sistemą reikia pasirinkti objektus, kurių unikalios savybės yra ne tik spalva (pavyzdžiui, spalvotos kortelės). Nes SURF algoritmas dirba tik su nespalvotomis nuotraukomis, tad jeigu objektas daugiau neturi išskirtinių savybių (be spalvos), realizuota objektų atpažinimo sistema veikia netiksliai.

4 TYRIMO DALIS

Analizuojant sistemą buvo atliekama sistemą naudojusią vartotojų apklausa Valstybės įmonėje „Registru centras“ ir UAB „Informacinių technologijų organizacija“ ir buvo analizuojami vartotojų pateikti atsakymai į anketos klausimus. Anketa pateikta Priedų 9.3 punkte.

Apklausti sistemos vartotojai išskyrė pagrindinį sistemos trūkumą: negalima sistemos objektų išskirstyti į skirtingus tipus (klases). Iki šiol sistemoje visi objektai dedami į vieną bevarde klasę ir nėra sistemos vartotojui parašoma, kokiam tipui priklauso objektas.

Patobulinta sistema bus vertinama pagal šiuos matavimo vienetus:

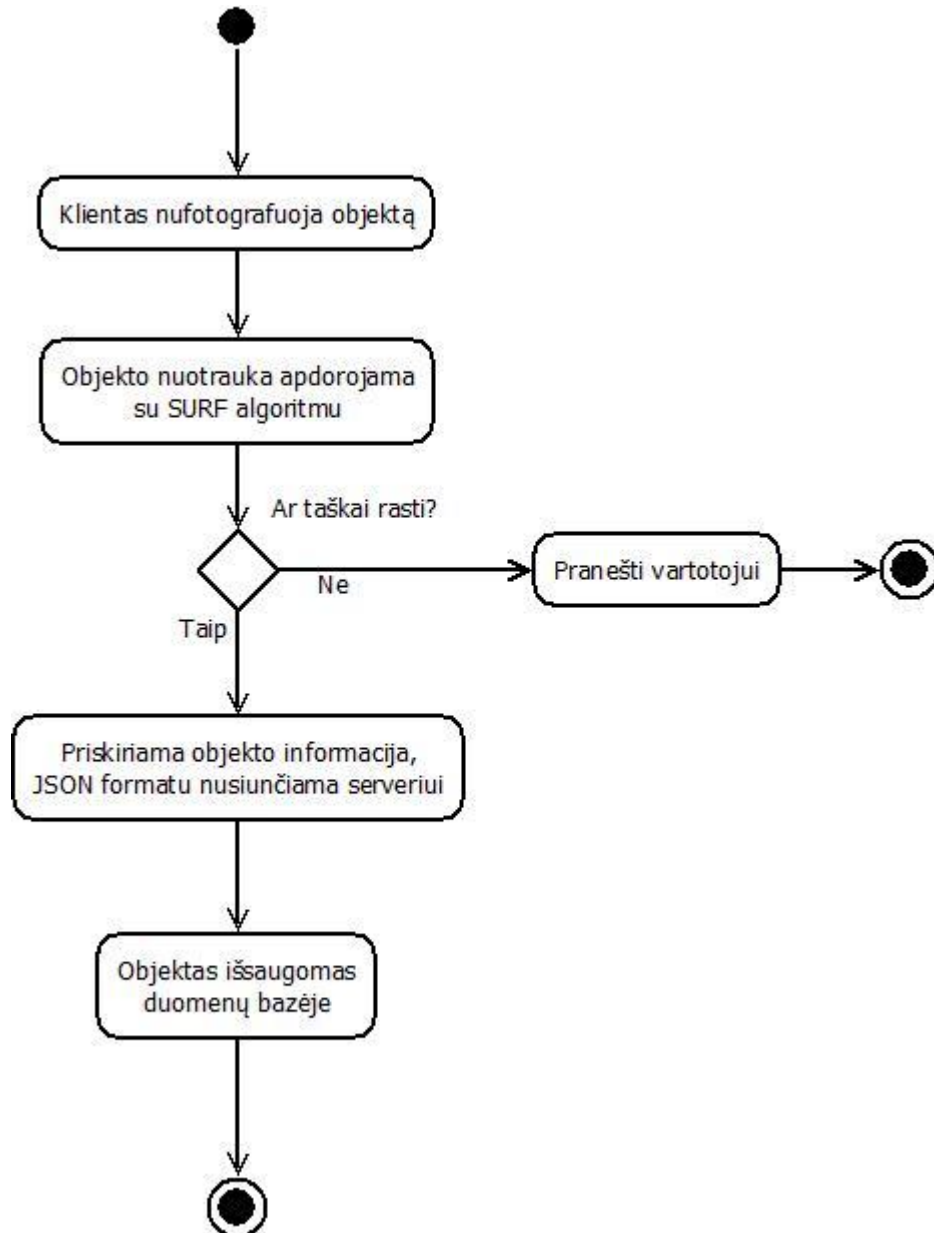
- Objektų klasės atpažinimo tikslumas procentais.
- Sistemos veikimo stabilumas (ar buvo nesklandumų sistemos veikimo darbe, ar sistema "nenulūžo" ir pan.). Tai bus vertinama žymintis visus nesklandumus, iškilusius eksperimentų su sistema metu.

Vertinant objektų atpažinimo tikslumą bus vertinama, kaip tiksliai sistema atpažino seniau įvestus objektus. Kadangi SURF algoritmo ypatybė yra ta, kad jis dirba tikrai su nespalvotais paveiksliukais, objektų atpažinimo tikslumo testavimui reikės pasirinkti objektus, kurie turėtų daugiau unikalių savybių - ne tik spalvą.

Testuojant objektų atpažinimo tikslumo požiūriu bus įvertinamos šios sistemos ypatybės:

- Tiksliai suklasifikuotų į klases objektų procentas. Bus tikrinama, kiek kartų sistema sėkmingai atpažino ir priskyre objekto klasę.
- Kokio dydžio baitais paketas yra siunčiamas tinklu iš kliento serveriui.
- Kaip sistemos veikimo tikslumą paveikia pakeista Hesiano slenksčio reikšmė.

4.1 Objektų klasifikavimo algoritmo aprašymas

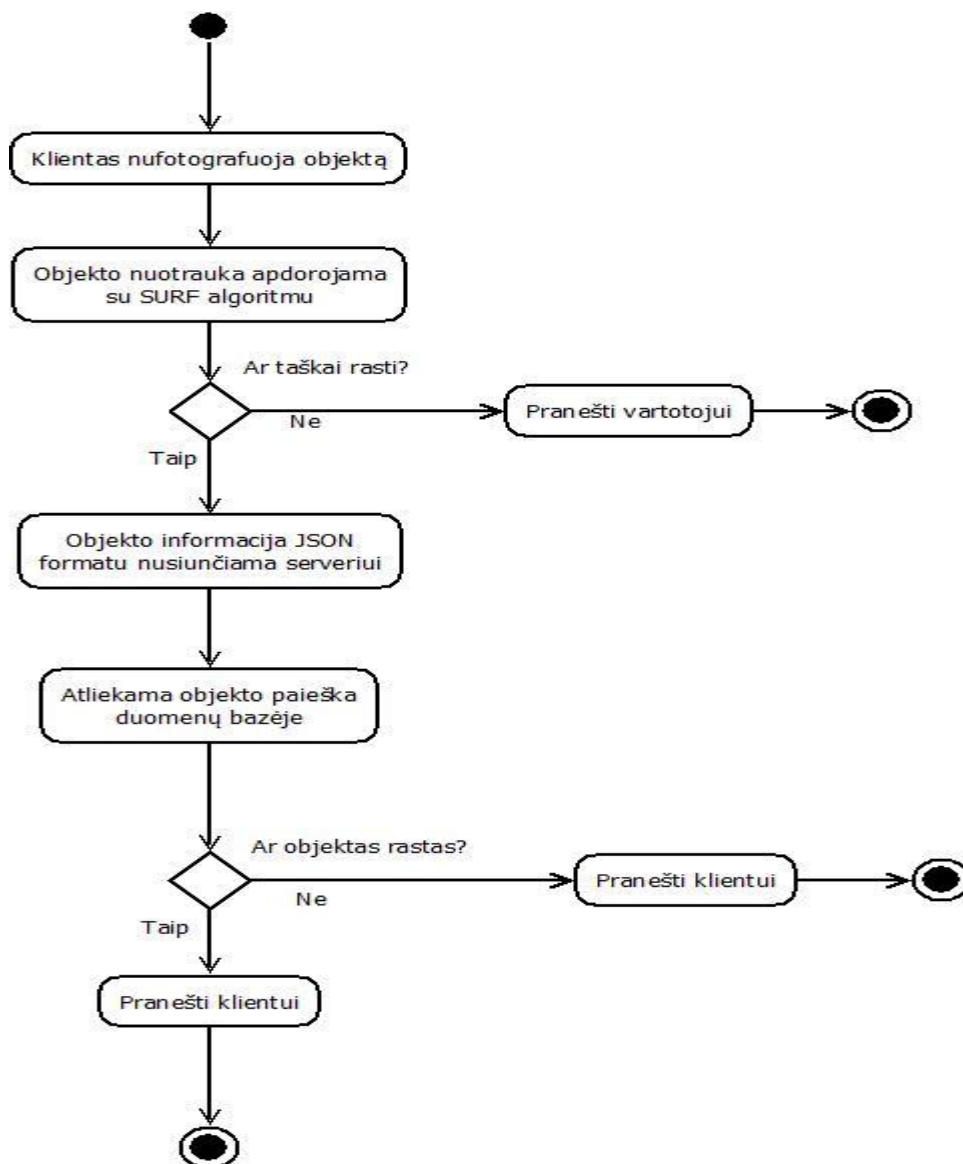


21 pav. Objektų klasifikavimo algoritmas

Objektų klasifikavimas (jo schema pateikta 21 paveiksliuke) bus vykdomas šiuo principu:

1. Klientas nufotografuoja objektą.
2. Objekto nuotrauka iš spalvotos yra paverčiama į nespaltvotą (*grayscale*).
3. Objekto nespaltvota nuotrauka yra apdorojama su SURF algoritmu. SURF algoritmas iš nespaltvotos nuotraukos išskiria „naudingus taškus“ (*interest points*) ir juos aprašančią informaciją.
4. Vartotojas objektui priskiria pavadinimą (pavyzdžiui, „rankinukas“) ir spėjamą objekto klasę (pavyzdžiui, „moteriškos rankinės“).

5. „Naudingų taškų“ informacija, taip pat objekto pavadinimas bei spėjama objekto klasė serializuojama JSON formatu (JSON formato žinutės pateiktos Priedų 9.1 punkte).
6. JSON formato objekto informacija perduodama serveriui.
7. Objekto informacija yra išsaugoma duomenų bazėje. Objektas priskiriamas nurodytai klasei.



22 pav. Objektų paieškos algoritmas

Ieškant objektų (algoritmas pateiktas 22 paveiksliuke) sistema veikia tokiu principu:

1. Klientas nufotografuoja objektą.
2. Objekto nuotrauka iš spalvotos yra paverčiama į nespalvotą (*grayscale*).

3. Objekto nespalvota nuotrauka yra apdorojamas su SURF algoritmu. SURF algoritmas iš nespalvotos nuotraukos išskiria „naudingus taškus“ (*interest points*) ir juos aprašančią informaciją.
4. „Naudingų taškų“ informacija serializuojama JSON formatu.
5. Serializuota informacija perduodama į serverį.
6. Serveris pagal perduotą „naudingų taškų“ informaciją ieško turimų objektų klasių duomenų bazėje objekto.
7. Radus objektą serveris perduoda klientui informaciją apie objekto pavadinimą bei objekto klasę.
8. Neradus objekto, serveris praneša klientui, kad toks objektas nerastas.

4.2 Reikalingas sistemos pakeitimas

Norint realizuoti objektų klasifikavimo algoritimą, sistemoje reikia atlikti šiuos pakeitimus:

1. Pataisyti JSON žinučių formatą, kad būtų galima pridėti galimybę siunčiant objekto informaciją kartu siųsti ir objektų kategoriją.
2. Android OS kliente reikia pridėti spėjamos objekto kategorijos pasirinkimas bei atitinkamai pataisytas programos išėties tekstas (*source code*).
3. Serveryje reikia pridėti naujus objektų tipus („puodeliai“ ir „žaislai“) bei pridėti programinį kodą, leidžiantį administruoti šių tipų objektus.
4. Pridėti duomenų bazės lentelės objektų tipams „puodeliai“ ir „žaislai“.

5 EKSPERIMENTINĖ DALIS

5.1 Techninė ir programinė įranga

Projektui testuoti buvo naudojama:

- HTC Evo 3D mobilusis telefonas. Šio mobiliojo telefono parametrai pateikti Priedų 9.2 punkte.
- VMWare Workstation 8.0 pagrindu veikiantis virtualus kompiuteris su 4 GB RAM, 2 procesoriais su dviem branduoliais kiekviename, Ubuntu Linux 11.10 64-ių bitų operacine sistema.
- Testavime naudotas kompiuteris su Intel i7-2600K procesoriumi (8-i branduoliai, 3,4 GHz), 16 GB RAM, Windows 7 SP1 Ultimate 64-ių bitų OS.

5.2 Eksperimento tikslas

Norėdami patikrinti, ar veikia vaizdų atpažinimo sistema, realizavome ją. Po realizacijos išbandėme šią sistemą ir tikrinome, ar:

- Sistema veikia ir gali klasifikuoti objektus.
- Ar sistema gali atpažinti objektus.
- Ar galima į sistemą pridėti, pašalinti objektus, taip pat atnaujinti jau egzistuojančių objektų informaciją.
- Ar sistema teisingai klasifikuoja objektus.
- Išskyla problemos sistemos veikime.

5.3 Realizacija

Sistema, pavadinta „Universalio vaizdo atpažinimo sistema“ (autoriai: Liudas Drejeris, Ernestas Kardzys ir Karolis Kriščiūnas, IFME-0/2 grupė), buvo sukurta naudojant C++, Java programavimo kalbas ir QT SDK bei Android SDK.

Sistemos serveris parašytas C++ programavimo kalba ir veikia Linux OS. Klientai parašyti:

- Linux, Windows OS naudojant atvirojo kodo ir nemokamą QT SDK bei C++ programavimo kalbą.
- Mobiliajam telefonui skirtas Android OS klientas buvo parašytas naudojant Android SDK ir Java programavimo kalbą.

Principinė Universalios vaizdo atpažinimo sistemos schema yra pateiktą 23-ame paveiksliuke.



23 pav. Principinė Universalios vaizdo atpažinimo sistemos schema

Vartotojas nufotografuoja objektą, jį apdoroja su SURF algoritmu ir serializuotą JSON formatu persiunčia į serverį. Šiame serveryje veikia Ubuntu Linux OS ir MySQL duomenų bazė. Serverio programinė įranga atlieka CRUD operacijas duomenų bazėje (paieškos, įterpimo, šalinimo, redagavimo) ir apie gautus rezultatus informuoja klientą.

Serveris atlieka šias funkcijas:

- Aptarnauja prisijungiančius klientus.
- Ieško, šalina, atnaujina, prideda objektus į duomenų bazę.
- Pateikia rezultatus klientui.

Serverio išvedamas pavyzdinis rezultatas yra pateiktas 24-ame paveiksliuke.

```

ernestas@ubuntu: ~/Desktop/Magistrinis/UVRSServer/dist/Debug/GNU-Linux-x86
0.014471084810793;0.014494650997222;-0.001253661350347;0.001284599886276;0.00984
0254671872;0.095419049263000;0.030982408672571;0.184363231062889;-0.076455309987
068;0.246189817786217;-0.000651699781884;0.168645218014717;-0.031977117061615;0.
206668078899384;0.007029886357486;0.295505344867706;-0.026339273899794;0.3240827
32200623;-0.067585349082947;0.203762471675873;-0.001429760595784;0.0130555965006
35;0.000983995501883;0.014242249540985;-0.009414294734597;0.017422040924430;-0.0
00609168957453;0.026723828166723;0.000000000000000;0.000000000000000;0.017411442
473531;0.017575811594725;-0.008171675726771;0.008171675726771;-0.000481855706312
;0.000481855706312;0.059761658310890;0.145676821470261;0.171384707093239;0.22092
6538109779;-0.105092458426952;0.480291098356247;0.090606786310673;0.111428096890
450;-0.102241098880768;0.157195031642914;-0.097899571061134;0.197891235351562;0.
029567262157798;0.160824552178383;-0.006661793217063;0.276190280914307;-0.008523
581549525;0.019532822072506;-0.005776872392744;0.013893226161599;0.0085530690848
83;0.020524973049760;-0.009695686399937;0.028295511379838;0.000000000000000;0.00
000000000000000;0.001960701541975;0.001960701541975;-0.001115946797654;0.001115946
797654;0.000000000000000;0.000000000000000;0.013115778565407;0.015344941988587;0
.007938285358250;0.019321737810969;-0.006567299365997;0.019966337829828;-0.00372
8282870725;0.004261351656169;-0.002609463408589;0.011114511638880;-0.00502308877
1850;0.035462621599436;-0.002232026075944;0.039703361690044;-0.011767718940973;0
.024943169206381;-0.002283510519192;0.002717015100643;0.002168200444430;0.003544
608363882;-0.001486900611781;0.001778879668564;0.004635971039534;0.0066514168865
98;" } } }
[*] Got action: add. Title: puodelis , Description: raudonas

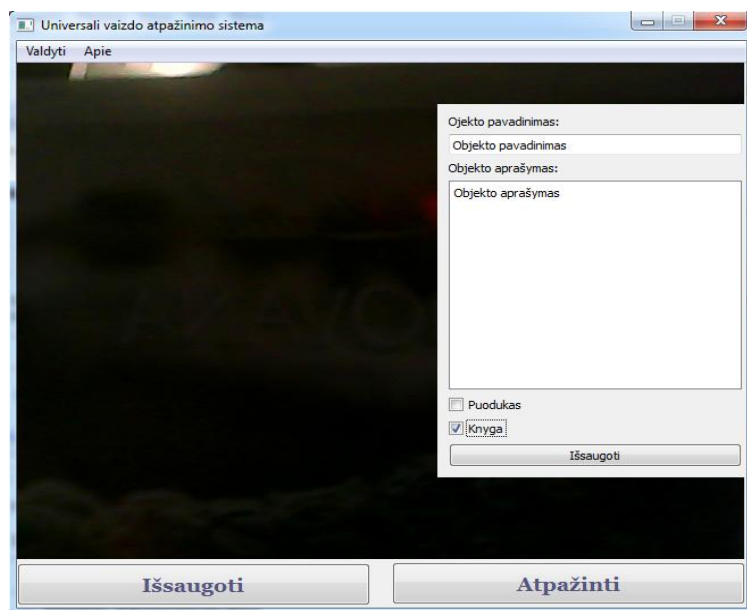
```

24 pav. Serverio išvesties pavyzdinis vaizdas

Klientas atlieka šias funkcijas:

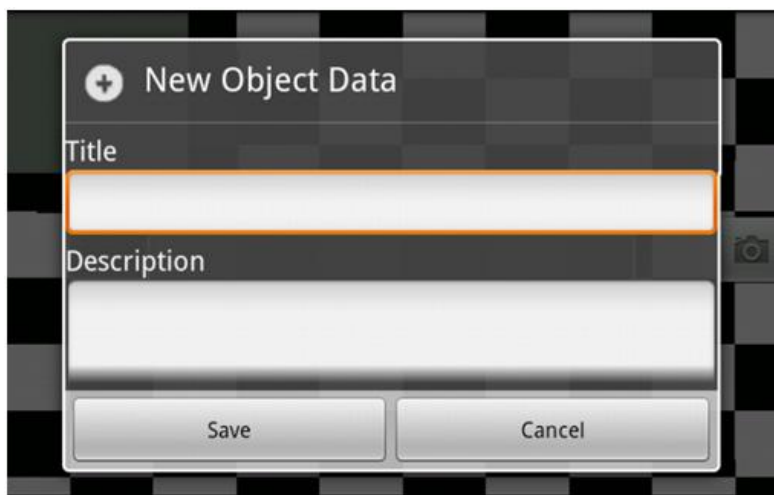
- Nufotografuoja objektus.
- Apdoroja nuotraukas su SURF algoritmu.
- Komunikuoja su serveriu.

Pavyzdinis Linux ir Windows klientų naujų objektų įvedimo langas yra pateiktas 25-ame paveiksliuke.



25 pav. Windows kliento naujo objekto įvedimo langas

Android OS kliento naujo objekto įvedimo langas yra pateiktas 26-ame paveiksluke.



26 pav. Kliento, skirto Android OS, naujo objekto įvedimo langas

Serveris yra realizuotas kaip komandinės eilutės Linux programa, klientai - kaip programos Linux, Windows, Android OS operacinėms sistemoms.

5.4 Eksperimento atlikimo sąlygos

Siekiant, kad vaizdo atpažinimo programinės įrangos testavimo rezultatus kiek galima mažiau įtakotų aplinkos sąlygos, testuoti parinkta aplinka su tokiomis charakteristikomis:

- Dirbtiniu apšvietimu. Analizuojamo objekto kairėje ir dešinėje buvo pastatytos dvi lempos su 60 W galios lempute.
- Po objektu esančiu baltu popieriaus lapu.
- Naudojama techninė ir programinė įranga aprašyta 5.1 skyriuje.

Objektų atpažinimo testavimas buvo atliekamas su šiomis objektų klasėmis (jų vaizdas pateiktas 27 paveiksluke):

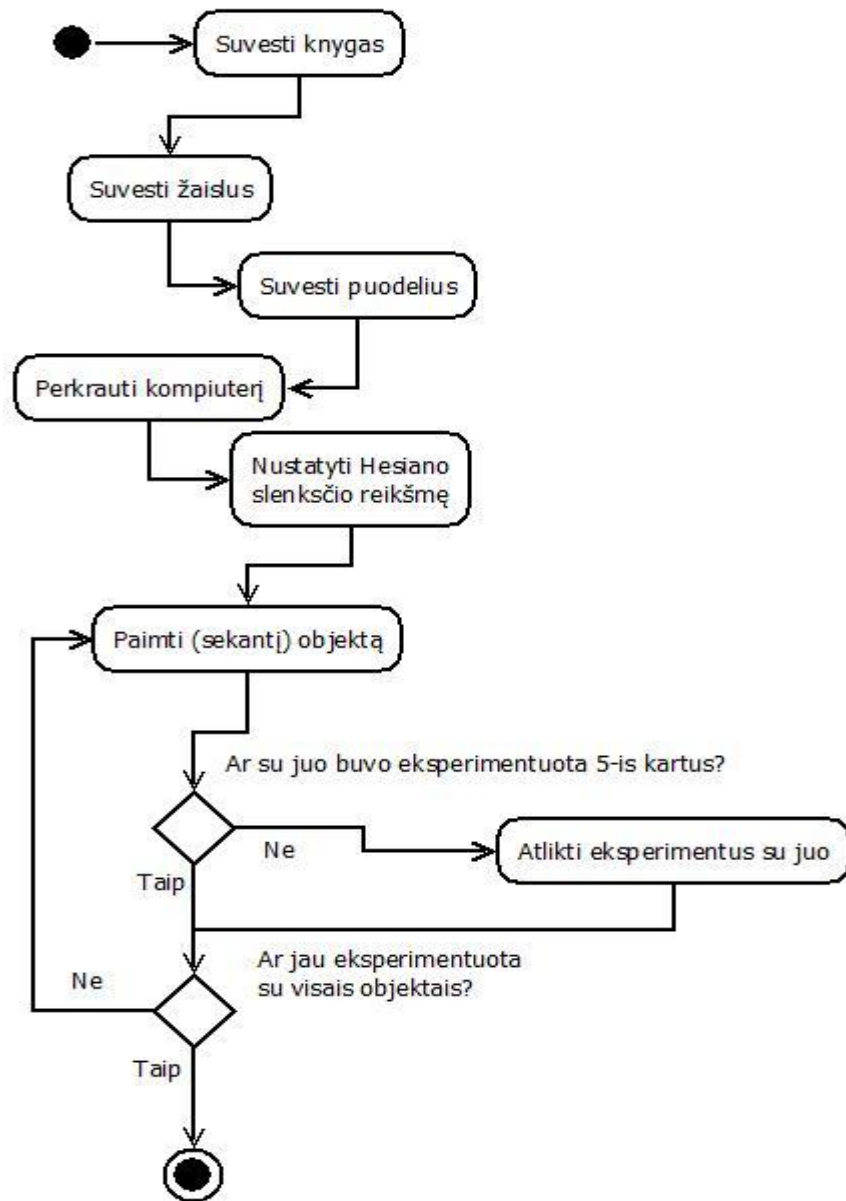
- Knygomis: Joseph Murphy „Jūsų pasąmonės galia“, Helen Fouche Gaines „Cryptanalysis: A Study of Ciphers and Their Solution“, Sun Zi „Karo menas“, Steven Feuerstein ir Bill Pribyl „Oracle PL/SQL Programming 2nd Edition“.
- Puodeliais: „Jacobs“ puodelis, puodelis su rože, Ožiaragio puodelis, puodelis su tigru.
- Žaislais: Panda, Vėžlys, Meškiukas.



27 pav. Testuojami objektai

5.5 Eksperimento atlikimo tvarka

Eksperimentuojant su sistema buvo keičiamos pradinės SURF algoritmo reikšmės ir ieškoma optimalių. Buvo rastos Hesiano slenksčio reikšmės 5000 ir 3000, su kuriomis sistema veikia geriausiai. Kiti SURF algoritmo parametrai buvo palikti standartiniai.



28 pav. Eksperimento tvarka

Eksperimentas (jo schema matoma 28-ame paveiksluke) buvo atliekamas tokia tvarka:

1. Įvedami visi objektai.
2. Nustatoma Hesiano slenksčio reikšmė - 5000.
3. Atpažįstant kiekvieną klasės objektą jis parodomas po 5-is kartus ir iš jų išvedamas sėkmingai suklasifikuotų objektų vidurkis.
4. Nustatoma Hesiano slenksčio reikšmė - 3000.
5. Vėl pakartojamas bandymas su objektų klasėmis - kiekvienas klasės objektas parodomas po 5-is kartus ir iš jų išvedamas sėkmingų bandymų vidurkis.

Atliekant bandymą su Hesiano slenksčio reikšme 5000 Android OS skirtame kliente buvo nustatyta Hesiano slenksčio reikšmė 5000. Ant stalo buvo sudėti keturi balti popieriaus lapai, ant kurių buvo padėti bandymų objektai ir su jais buvo bandoma sistema.

Objektai į sistemą buvo įvedami iš eilės, pagal klases. Pirmiausia buvo įvesta objektų klasė „knygos“, tada - „žaislai“ ir paskutinė buvo suvesta klasė „puodeliai“. Suvedus visas objektų klases, buvo perkrautas virtualus kompiuteris ir tada buvo atliekami bandymai su objektų klasėmis.

Bandant atpažinti objektų klases jie buvo imami atsitiktine tvarka. Pabandžius atpažinti vieną objektą ir užfiksavus bandymo rezultatą (ar buvo teisingai atpažinta objekto klasė), jis padedamas atgal prie likusių. Tada atsitiktine tvarka imamas kitas objektas, bandoma atpažinti objekto klasę, užfiksuojamas atpažinimo rezultatas ir t.t.

Atlikus visus atpažinimo bandymus, duomenys iš duomenų bazės buvo ištrinti, Android OS kliente buvo pakeista Hesiano slenksčio reikšmė į 3000, iš naujo įvedamos visos objektų klasės į sistemą, perkraunamas virtualus kompiuteris ir kartojami atpažinimo bandymai fiksuojant gautus rezultatus.

5.6 Eksperimentų rezultatai

Atliekant bandymus su Hesiano slenksčio reikšme 5000 buvo bandyta klasifikuoti objektus. Gauti šie rezultatai:

- Klasės „puodeliai“ sėkmingai atpažintų objektų procentas - 25%, vidutinis objekte rastų taškų skaičius - 13.
- Klasės „knygos“ sėkmingai atpažintų objektų procentas - 70%, vidutinis objekte rastų taškų skaičius - 11.
- Klasės „žaislai“ sėkmingai atpažintų objektų procentas - 60%, vidutinis objekte rastų taškų skaičius - 8.

Atlikus bandymus su Hesiano slenksčio reikšme 5000 buvo rastas vidutinis sėkmingai atpažintų bandymų procentas - 52%.

Pakartotinai atliekant bandymus su Hesiano slenksčio reikšme 3000 buvo bandyta klasifikuoti objektus. Gauti šie rezultatai:

- Klasės „puodeliai“ sėkmingai atpažintų objektų procentas - 95%, vidutinis objekte rastų taškų skaičius - 39.
- Klasės „knygos“ sėkmingai atpažintų objektų procentas - 95%, vidutinis objekte rastų taškų skaičius - 27.
- Klasės „žaislai“ sėkmingai atpažintų objektų procentas - 87%, vidutinis objekte rastų taškų skaičius - 16.

Atlikus bandymus su Hesiano slenksčio reikšme 5000 buvo rastas vidutinis sėkmingai atpažintų bandymų procentas - 92%.

Apibendrinti eksperimento rezultatai yra pateikti 14-oje lentelėje.

14 lentelė. Eksperimento rezultatų apibendrinimas

| Klasė | Hesiano slenksčio reikšmė | Vidutinis rastų taškų skaičius | Vidutinis sėkmingų bandymų procentas (%) |
|-----------|---------------------------|--------------------------------|--|
| Puodeliai | 5000 | 13 | 25 |
| | 3000 | 39 | 95 |
| Knygos | 5000 | 11 | 70 |
| | 3000 | 27 | 95 |
| Žaislai | 5000 | 8 | 60 |
| | 3000 | 16 | 87 |

Kaip matome iš bandymų rezultatų, kuo didesnis objekte rastų taškų skaičius, tuo tiksliau objektas yra klasifikuojamas.

5.7 Eksperimento išvados

Atlikus eksperimentą galime teigti:

1. Objekto klasifikavimo tikslumas priklauso nuo objekte rastų taškų skaičiaus. Kuo daugiau taškų objekte yra randama, tuo tiksliau klasifikuojamas objektas.
2. Sukurta objektų atpažinimo sistema su nustatyta Hesiano slenksčio reikšmė 5000 blogai veikia su sferiniais objektais (bandytu atveju - puodeliais).
3. Didėjant rastų taškų skaičiui didėja iš kliento į serverį siunčiamų paketų dydis.
4. Sistema geriausiai veikia su plokščiais objektais (pavyzdžiui, knygomis).
5. Eksperimentuojant su objektais ir dirbant su sistema nebuvo pastebėta jokių sistemos „lūžimų“.

5.8 Pastebėti sistemos trūkumai

Atlikus eksperimentą buvo pastebėti keli sistemos trūkumai. Būtų tikslinga padaryti šiuos sistemos pakeitimus:

1. Suspausti siunčiamas JSON formato žinutes panaudojant suspaudimo (*compression*) algoritmą, pavyzdžiui, ZIP. Taip būtų sutaupoma tinklo srauto ir greičiau vyktų informacijos apsikeitimas tarp kliento ir serverio.

2. Pasirinkimą kliente, ar norima nuotrauką apdoroti lokaliai, ar siųsti nuotrauką į serverį apdorojimui. Taip būtų galima pritaikyti klientą ir ne tokiems galingiems mobiliojo ryšio telefonams.
3. Reikėtų patobulinti Universalios vaizdo atpažinimo sistemos klientą, kad jis vartotų mažiau elektros energijos. Padirbus su šiuo klientu daugiau 6 valandas pilnai įkrautos baterijos lygis buvo sumažėjęs iki 30%. Dėl intensyvaus belaidžio ryšio (*Wi-Fi*), vaizdo kameros bei procesoriaus naudojimo greitai išsikrauna mobiliojo telefono baterija.

6 IŠVADOS

Atlikus magistrinį darbą buvo padarytos šios išvados:

1. Atlikus objektų atpažinimo algoritmų analizę buvo pasirinktas SURF algoritmas, nes jis veikia greičiau negu kiti apžvelgti algoritmai. Taip pat yra šis algoritmas yra realizuotas OpenCV bibliotekoje.
2. Teisingai atpažintų ir suklasifikuotų objektų procentas priklauso nuo nufotografuotame objekte aptiktų taškų skaičiaus.
3. Kuriant objektų atpažinimo sistemą, pagrindinės problemos yra šios:
 - Duomenų perdavimo problema.
 - Duomenų saugojimo problema.
 - Nuotraukos apdorojimo algoritmo problema.
4. Pasiūlytas vaizdo atpažinimo sistemos sprendimas, kuris:
 - Leidžia atpažinti ir klasifikuoti objektus 92% tikslumu.
 - Veikia naudodamas kliento - serverio architektūrą.
 - Turi klientus Android, Linux, Windows operaciniams sistemoms.
 - Naudoja SURF algoritmą objektų atpažinimui.
 - Buvo testuotas su 3-is objektų klasėmis: knygomis, puodeliais ir žaislais.
5. Hesiano slenksčio reikšmės pakeitimas padeda pagerinti sistemos veikimo tikslumą. Sistema geriausiai veikia su plokščiais objektais (pavyzdžiui, knygomis ar vaistų dėžutėmis), kadangi visas objektas yra vienoje plokštumoje.
6. Siuntinėjant JSON formato žinutes buvo pasiektas maksimalus žinutės dydis (156 kilobaitai). Tad, būtų tikslinga žinutes suspausti (pavyzdžiui, panaudojant ZIP algoritmą) ir tada perduoti tinklu.

7 LITERATŪRA

1. Treiber, M., „An Introduction to Object Recognition: Selected Algorithms for a Wide Variety of Applications“, London, 2010, ISBN 978-1-84996-234-6.
2. Canny, J.F., „An Computational Approach to Edge Detection“, „IEEE Transactions on Pattern Analysis and Machine Intelligence“, 8(6): 679 – 698, 1986.
3. Harris, Ch.; ir Stephens, M. „A Combined Corner and Edge Detector“, Cambridge, „Alvey Vission Conference“, 1988.
4. Schmid, C.; Mohr, R.; ir Bauckhage, C. „Evaluation of interest pint detectors“, „International Journal of Computer Vision“, 2000, 37(2):151–172, ISSN: 0920-5691.
5. Collins, R. „Lecture 6: Harris Corner Detection“, Penn State University's Computer Science and Engineering Department.
6. Smith, S. M.; ir Brady M. L. „SUSAN – A New Approach to Low Level Image Processing“, "International Journal of Computer Vision", 1997, 23(1): 45–78, ISSN: 0920-5691.
7. Švab, J.; Krajnik, T.; Faigl, J.; ir Preučil L. „FPGA Based SpeededUp Robust Features“, „Technologies for practical robot applications“, 2009, 35 – 41, ISBN: 978-1-4244-4991-0.
8. Bay, H.; Tuytelaars, T.; ir Van Gool, L. „SURF: Speeded Up Robust Features“, „Computer Vision and Image Understanding“, 2008, tomas 110, nr. 3, 346 – 359, ISBN: 3-540-33832-2.
9. Zou, L.; Chen, J.; Zhang, J.; ir Dou, L., „The Comparison of Two Typical Corner Detection Algorithms“, Jiangxi, „2008 Second International Symposium on Intelligent Information Technology Application (Iita)“, 2008, ISBN: 1424440637
10. W3C, Extensible Markup Language (XML) 1.0 (Fourth Edition) [žiūrėta 2012-03-17]. Prieiga per internetą: < <http://www.w3.org/TR/2006/REC-xml-20060816/>>.
11. JSON [žiūrėta 2012-03-17]. Prieiga per internetą: < <http://www.json.org/>>.
12. Hunter, D.; Rafter, J. ir kt., „Beginning XML, 4th Edition“, 2007, p. 6 - 8, ISBN: 978-0470114872
13. Nurseitov, N.; Paulson, M.; Reynolds, R.; Izurieta, C.; „Comparison of JSON and XML Data Interchange Formats: A Case Study“ [žiūrėta 2012-03-17], 2009. Prieiga per internetą: < <http://www.cs.montana.edu/izurieta/pubs/caine2009.pdf>>.
14. OpenCV (Open Source Computer Vision) [žiūrėta 2012-04-30]. Prieiga per internetą: < <http://opencv.willowgarage.com/wiki/>>.
15. Insight Toolkit [žiūrėta 2012-04-30]. Prieiga per internetą: <<http://www.itk.org/>>.
16. Integrating Vision Toolkit [žiūrėta 2012-04-30]. Prieiga per internetą: <<http://ivt.sourceforge.net/>>.

17. Visualization Toolkit [žiūrėta 2012-04-30]. Prieiga per internetą: <<http://www.vtk.org/>>.
18. Google, „Google Goggles“ [žiūrėta 2012-04-30]. Prieiga per internetą: <<http://www.google.com/mobile/goggles/>>.
19. GemIdent [žiūrėta 2012-04-30]. Prieiga per internetą: <<http://gemident.com/>>.
20. S. Olsson, P. Akesson. Distributed Mobile Computer Vision and Applications on Android Platform. *Magistro darbas*. Lund University. Prieiga per internetą: <<http://www.maths.lth.se/media/thesis/2009/akesson-olsson-FMA820.pdf>>.
21. Ubuntu [žiūrėta 2012-04-30]. Prieiga per internetą: <<http://www.ubuntu.com/>>.
22. Oracle, MySQL [žiūrėta 2012-04-30]. Prieiga per internetą: <<http://www.mysql.com/>>.
23. Oracle, MySQL/C++ Connector [žiūrėta 2012-04-30]. Prieiga per internetą: <<http://www.mysql.com/products/connector/>>.
24. Boost [žiūrėta 2012-04-30]. Prieiga per internetą: <<http://www.boost.org/>>.
25. Nokia, QT SDK [žiūrėta 2012-04-30]. Prieiga per internetą: <<http://qt.nokia.com/>>.
26. „General Motors“ Elektros ir valdymo integracijos laboratorijos interneto svetainė [žiūrėta 2010-11-06]. Prieiga per internetą: <<http://www.gm.com/vehicles/innovation/research/overview/eci.jsp>>
27. L. Drejeris; E. Kardzys; K. Kriščiūnas; „Universalios vaizdo atpažinimo sistemos kūrimas ir tyrimas. Magistrinio projekto pilna dokumentacija“. Kauno technologijos universitetas, 2012.
28. Sang Hwa Lee, Junyeong Choi, and Jong-Il Park „Interactive E-Learning System Using Pattern Recognition and Augmented Reality“, IEEE Transactions on Consumer Electronics, Vol. 55, No. 2, MAY 2009
29. Robert K. McConnell „Off-the-Shelf System for Color and Multispectral Based Recognition and Control“, Technologies for Practical Robot Applications, 2008. TePRA 2008, IEEE.
30. U.S. National Library of Medicine, Visible Human Project [žiūrėta 2012-05-21]. Prieiga per internetą: <http://www.nlm.nih.gov/research/visible/visible_human.html>

8 TERMINŲ IR SANTRUMPŲ ŽODYNAS

- **Objektas** (*object*) – briaunų, linijų, kampų ir spalvų visuma.
- **Objektų atpažinimas** (*object recognition*) – veiksmų seka, kurios metu atpažįstamas objektas ir informacija apie objektą išsaugoma duomenų bazėje.
- **Objekto tipas** - apibendrinančios objekto rūšies pavadinimas, kuriam priklauso objektas. Pavyzdžiui, „alyvinio obuolio“ tipas yra „obuolys“, o automobilio „Mazda Premacy“ tipas yra „automobilis“.
- **Kampų aptikimas** (*corner detection*) – operacija, kurios metu aptinkami objekto kampai.
- **Briaunų aptikimas** (*edge detection*)– operacija, kurios metu yra aptinkamos objekto briaunos.
- **Briauna** (*edge*) – linija, jungianti du taškus.
- **Kampas** (*corner*) – dviejų briaunų susikirtimas. Briaunas susikerta kampu, matuojamu laipsniais.
- **„Naudingas taškas“** (*interest point*) – paveiksluko taškas, kuris:
 - Turi aiškų, matematinį apibrėžimą;
 - Turi aiškiai apibrėžtą padėtį paveiksliuke;
 - Turi daug naudingos informacijos, esančios aplink šį tašką, kuri bus naudojama paveiksluko apdorojime;
 - Yra pastovioje vietoje, nekeičiantis pozicijos apskaitant paveiksluką ar keičiant paveiksluko ryškumą.
- **Gradientas** - diferencialinis operatorius, skaliarinį lauką atvaizduojantis į vektorinį lauką taip, kad kiekvienas vektorinio lauko vektorius būtų nukreiptas skaliarinio lauko reikšmių didėjimo kryptimi, o jo modulis būtų lygus kryptinei išvestinei šiame taške.
- **JSON** (*JavaScript Object Notation*) - duomenų apsikeitimo formatas.
- **XML** (*Extensible Markup Language*) - duomenų apsikeitimo formatas.
- **CPU** (*Central Processing Unit*) - pagrindinis kompiuterio procesorius.
- **TCP/IP** (*Transmission Control Protocol/Internet Protocol*) - duomenų perdavimo tinklu protokolų rinkinys.
- **Jungtis** (*port*) - programinis operacinės sistemos komponentas, naudojamas komunikacijai
- **CRUD** (*Create, Read, Update, Delete*) operacijos - pagrindinės operacijos su duomenų bazės duomenimis - sukūrimo, skaitymo, atnaujinimo, pašalinimo operacijos.

- **Soketas** (*socket*) - IP adreso ir jungties (*port*) kombinacija, nurodanti komunikacijos sąsają tarp dviejų programų.
- **Branduolys** - matematinė operacija.
- **Sąsūka** - matematinė operacija.
- **Segmentas** - paveiksluko dalis.
- **Segmentavimas** - procesas, kai paveikslukas yra išskaidomas į kelis segmentus (dalis).
- **SIFT** (*Scale Invariant Feature Transform*) - paveiksluko savybių išskyrimo algoritmas.
- **GPU** (*Graphics Processing Unit*) - specialus kompiuterio procesorius, kurio pagrindinė paskirtis yra greitai apdoroti vaizdą ir jį pateikti kompiuterio monitoriaus ekrane.
- **PPI** (*Pixels Per Inch*) - taškų skaičius kvadratiname colyje.
- **OCR** (*Optical Character Recognition*) - paveiksluko pavertimas į tekstą.
- **DARPA** (*Defense Advanced Research Projects Agency*) - JAV agentūra, kuri kuria naujas technologijas ir pritaiko jas JAV kariuomenės reikmėms.
- **Biblioteka** (*library*) - funkcijų rinkinys.
- **GNU** – „GNU is Not Unix“, plačiau žinoma kaip GNU Project. Projektas skirtas vystyti atviro kodo programinę įrangą ir pateikti ją visuomenei nemokamam naudojimui.
- **GNU GPL** – nemokama atviro kodo licenzija, pagal kurią platinama daugelis populiarių ir nemokamų produktų: Linux OS, Apache HTTP serveris, MySQL duomenų bazių serveris ir daugelis kitų.
- **SURF** – „Speed Up Robust Features“. Vienas iš plačiausiai naudojamų vaizdo atpažinimo algoritmas, pagrįstas nuo mastelio nepriklausančių savybių aptikimu.
- **2D** – dvimatis (dviejų dimensijų – pločio ir ilgio).
- **3D** – trimatis (trijų dimensijų – ilgio, pločio ir aukščio).
- **Klasė** - objekto tipas (pavyzdžiui, „indai“, „kėdės“ ir pan.).
- **Klasifikavimas** - klasės nustatymas ir priskyrimas.
- **Objektų klasifikavimas** - objekto priskyrimas klasei.
- **Hesiano slenkstis** (*Hessian threshold*) - dydis, nurodantis SURF algoritmo jautrumą. Bus išskirti tik tie taškai, kurių reikšmės yra didesnės negu Hesiano slenkstis.
- **Matomo žmogaus projektas** (*Visible Human Project*) – projektas, kurio tikslas yra pateikti žmogaus kūno nuotraukas visuomenei.

9 PRIEDAI

9.1 JSON formato žinutės informacijos apsikeitimui

9.1.1 Objekto pridėjimas

```
{
  "data":
  {
    "action-desc":
    {
      "action": "add-object"
    },
    "object-data":
    {
      "id": "0",
      "type": "<tipas>",
      "descriptors": "<deskriptoriai>",
      "title": "<pavadinimas>",
      "author": "<autorius>"
    }
  }
}
```

Pavyzdžiui:

```
{
  "data":
  {
    "action-desc":
    {
      "action": "add-object"
    },
    "object-data":
    {
      "id": "0",
      "type": "book",

```



```
        "descriptors": "!0;1.5;56;45...",
        "title": "Nusikaltimas ir bausmė",
        "author": "Feodoras Dostojevskis"
    }
}
}
```

9.1.2 Objekto išrinkimas

```
{
  "data":
  {
    "action-desc":
    {
      "action": "select-object"
    },
    "object-data":
    {
      "id": "<objekto numeris>",
      "type": "<objekto tipas>",
      "descriptors": "!0;1.5;56;45..."
    }
  }
}
```

Pavyzdžiui:

```
{
  "data":
  {
    "action-desc":
    {
      "action": "select-object"
    },
    "object-data":
    {
```

```
        "id": "11",
        "type": "cup"
    }
}
```

9.1.3 Informacinė žinutė

```
{
  "data":
  {
    "message-desc":
    {
      "type": "info-message"
    },
    "message-body":
    {
      "message": "<žinutės tekstas>"
    }
  }
}
```

Pavyzdžiui:

```
{
  "data":
  {
    "message-desc":
    {
      "type": "info-message"
    },
    "message-body":
    {
      "message": "Mano žinutė"
    }
  }
}
```

9.1.4 Klaidos žinutė

```
{
  "data":
  {
    "message-desc":
    {
      "type": "error-message"
    },
    "message-body":
    {
      "message": "<klaidos žinutės tekstas>"
    }
  }
}
```

Pavyzdžiui:

```
{
  "data":
  {
    "message-desc":
    {
      "type": "error-message"
    },
    "message-body":
    {
      "message": "Wrong data format!"
    }
  }
}
```

9.1.5 Žinutė, jog objektas atpažintas

```
{
  "data":
  {
```

```
"message-desc":
{
  "type": "object-recognized-message"
},
"message-body":
{
  "id": "10",
  "message": "Object recognised!",
  "type": "<tipas>",
  "title": "<pavadinimas>",
  "description": "<aprašymas>"
}
}
```

Pavyzdžiui:

```
{
  "data":
  {
    "message-desc":
    {
      "type": "object-recognized-message"
    },
    "message-body":
    {
      "id": "10",
      "message": "Object recognised!",
      "type": "book",
      "title": "Trys muškietininkai",
      "description": "Gera knyga"
    }
  }
}
```

9.1.6 Žinutė, jog objektas neatpažintas

```
{
  "data":
  {
    "message-desc":
    {
      "type": "error-message"
    },
    "message-body":
    {
      "message": "<tekstas>"
    }
  }
}
```

Pavyzdžiui:

```
{
  "data":
  {
    "message-desc":
    {
      "type": "error-message"
    },
    "message-body":
    {
      "message": "Sorry, object not recognised!"
    }
  }
}
```

9.2 Mobiliojo telefono HTC Evo 3D parametrai

15 lentelė. Mobiliojo telefono HTC Evo 3D parametrai

| Parametras | Reikšmė |
|----------------------|--|
| Palaikomi 2G tinklai | GSM 850 / 900 / 1800 / 1900 |
| Palaikomi 3G tinklai | HSDPA 900 / 2100 arba HSDPA 850 / 2100, HSDPA 900 / 1700 / 2100 |
| Ekranas | 540 x 960 pikselių, 4,3 colio (≈256 PPI pikselių tankumas) |
| Atmintis | 1 GB pastovi atmintis, 1 GB RAM |
| GPRS | Iki 114 kbps |
| EDGE | Iki 560 kbps |
| WLAN | Wi-Fi 802.11 b/g/n, DLNA, Wi-Fi stotelė |
| Kamera | 5 MP, 2560x1920 pikselių |
| OS | Android OS, v2.3 (Gingerbread) |
| Lustų rinkinys | Qualcomm MSM8660 |
| CPU | 2 branduolių 1,4 GHz |
| GPU | Adreno 220 |
| Baterija | 1730 mAh |

9.3 Apklausos anketa

Projekto pavadinimas: „Universali vaizdo atpažinimo sistema“

Įvertinkite žemiau pateiktus teiginius skalėje nuo 1 iki 10, kur 1 – žemiausias įvertinimas, o 10 - aukščiausias. Jeigu charakteristika Jums netinka (ar nežinote) – įrašykite raidę N.

- pasirinkta Linux OS yra tinkama serveriui.
- projektas veikia tinkamai.
- sistemos klientais naudotis yra paprasta.
- darbas su sistema vyksta greitai.
- sistema galima išmokti greitai naudotis.
- sistemą rekomenduočiau naudoti kitiems.

Pažymėkite, kiek laiko dirbote su projektu.

- mažiau nei 1 diena.
- 1 – 3 dienos.
- 3 – 7 dienos.

__ 7 dienos ir daugiau.

Pažymėkite kiek operacijų (įterpimo, redagavimo, šalinimo, atpažinimo) atlikote:

__ iki 5.

__ 5 - 20.

__ 20 - 50.

__ 50 - 100.

__ daugiau nei 100.

Įrašykite pastebėtą didžiausią sistemos privalumą:

Įrašykite pastebėtą didžiausią sistemos trūkumą:

Savais žodžiais apibūdinkite projektą:

9.4 Duomenų bazės lentelių aprašymas

Lentelėje Objects (16 lentelė) yra saugoma bendra informacija apie objektą.

16 lentelė. Lentelės Object struktūra

| Lauko pavadinimas | Lauko tipas | Duomenų pavyzdys | Aprašymas |
|-------------------|-------------|------------------|--|
| ID | Integer | 1 | Unikalus identifikatorius, apibūdinantis įrašą |
| ObjectType | Integer | 5 | Nurodo objekto tipą (iš lentelės objecttype) |
| weight | Integer | 2 | Nurodo laukelio svorį, nenaudojamas |

Lentelėje Keypoints (17 lentelė) yra saugoma aptiktų taškų informacija.

17 lentelė. Lentelės Keypoints struktūra

| Lauko pavadinimas | Lauko tipas | Duomenų pavyzdys | Aprašymas |
|-------------------|-------------|------------------|--|
| ID | Integer | 1 | Unikalus identifikatorius, apibūdinantis įrašą |
| ObjectID | Integer | 5 | Nurodo objekto numerį (sutampa su objects.ID laukeliu) |
| descriptor | Longtext | !0;1.5;56;45... | Saugomas paveiksluko taškų deskriptorius. |

Lentelėje Objecttype (18 lentelė) yra saugomos objektų klasės.

18 lentelė. Lentelės Objecttype struktūra

| Lauko pavadinimas | Lauko tipas | Duomenų pavyzdys | Aprašymas |
|-------------------|-------------|------------------|--|
| ID | Integer | 1 | Unikalus identifikatorius, apibūdinantis įrašą |
| Name | Text | Obuolys | Nurodo objekto klasę. |

Lentelėje Books (19 lentelė) yra saugoma objekto klasės „Knygos“ informacija.

19 lentelė. Lentelės Books struktūra

| Lauko pavadinimas | Lauko tipas | Duomenų pavyzdys | Aprašymas |
|-------------------|-------------|---------------------|---|
| ID | Integer | 1 | Unikalus identifikatorius, apibūdinantis įrašą |
| Title | Text | Trys muškietininkai | Nurodo knygos pavadinimą |
| ObjectID | Integer | 2 | Nurodo knygos numerį (sutampa su objects.ID laukeliu) |

| Lauko pavadinimas | Lauko tipas | Duomenų pavyzdys | Aprašymas |
|-------------------|-------------|------------------|------------------|
| Description | Text | Puiki knyga | Knygos aprašymas |

Lentelėje Cups (20 lentelė) yra saugoma objekto klasės „Puodelis“ informacija.

20 lentelė. Lentelės Cups struktūra

| Lauko pavadinimas | Lauko tipas | Duomenų pavyzdys | Aprašymas |
|-------------------|-------------|------------------|---|
| ID | Integer | 1 | Unikalus identifikatorius, apibūdinantis įrašą |
| Name | Text | Žalias puodelis | Nurodo puodelio pavadinimą |
| ObjectID | Integer | 2 | Nurodo puodelio numerį (sutampa su objects.ID laukeliu) |

Lentelėje Toys (21 lentelė) yra saugoma objekto klasės „Žaislai“ informacija.

21 lentelė. Lentelės Toys struktūra

| Lauko pavadinimas | Lauko tipas | Duomenų pavyzdys | Aprašymas |
|-------------------|-------------|------------------|---|
| ID | Integer | 1 | Unikalus identifikatorius, apibūdinantis įrašą |
| Name | Text | Žalias puodelis | Nurodo žaislo pavadinimą |
| ObjectID | Integer | 2 | Nurodo žaislo numerį (sutampa su objects.ID laukeliu) |

9.5 Naudotos kompiliavimo programinės įrangos parametrai

Kompiliavimui naudota programinė yra aprašyta 22-oje lentelėje.

22 lentelė. Naudotos programinės įrangos parametrai

| Pavadinimas | Versija | Paskirtis |
|-------------|---------|-----------|
|-------------|---------|-----------|

| Pavadinimas | Versija | Paskirtis |
|-----------------|---------|-------------------------------|
| G++ | 4.6.1 | C++ kalbos kompiliatorius |
| NetBeans | 7.1 | Programavimo IDE |
| OpenCV | 2.3.1 | Biblioteka darbui su vaizdais |
| MySQL | 5.1.58 | Duomenų bazių serveris |
| Boost | 1.46.1 | C++ funkcijų biblioteka |

9.6 Eksperimento rezultatai

Eksperimentuojant su objektais buvo gauti rezultatai su Hesiano slenksčio reikšmėmis 5000 ir 3000. Eksperimentuojant buvo surinkti duomenys apie siunčiamų paketų dydį bei objekte rastų taškų skaičių.

Eksperimentų rezultatai pateikiami 23-oje ir 24-oje lentelėse. Eksperimentuojant, buvo ieškoma, ar teisingai objektas buvo priskirtas savo klasei.

23 lentelė. Eksperimentai su Hesiano slenksčiu 5000

| Daiktas | Bandymas 1 | Bandymas 2 | Bandymas 3 | Bandymas 4 | Bandymas 5 | Rastų taškų skaičius | Paketo dydis |
|---|-------------|-------------|-------------|-------------|-------------|----------------------|--------------|
| Panda | Priskirta | Priskirta | Priskirta | Vėžlys | Priskirta | 22 | 55278 |
| Vėžlys | Neatpažinta | Neatpažinta | Neatpažinta | Neatpažinta | Neatpažinta | 1 | 2412 |
| Meškiukas | Priskirta | Priskirta | Priskirta | Priskirta | Priskirta | 2 | 7207 |
| Knyga "Oracle PL/SQL Programming 2nd Edition" | Nepriskirta | Nepriskirta | Nepriskirta | Nepriskirta | Nepriskirta | 4 | 12048 |
| Knyga "Cryptanalysis: A Study of Their Ciphers and Their Solution" | Priskirta | Priskirta | Priskirta | Nepriskirta | Priskirta | 2 | 7200 |
| Knyga "Karo menas" | Priskirta | Priskirta | Priskirta | Priskirta | Priskirta | 36 | 91233 |
| Knyga "Pasąmonės galia" | Priskirta | Nepriskirta | Priskirta | Priskirta | Priskirta | 2 | 7193 |
| Puodelis su tigrū | Priskirta | Nepriskirta | Nepriskirta | Nepriskirta | Nepriskirta | 15 | 38384 |
| Puodelis su rože | Nepriskirta | Nepriskirta | Nepriskirta | Nepriskirta | Nepriskirta | 7 | 25012 |
| Jacobs puodelis | Nepriskirta | Priskirta | Nepriskirta | Nepriskirta | Priskirta | 12 | 31223 |

| Daiktas | Bandymas 1 | Bandymas 2 | Bandymas 3 | Bandymas 4 | Bandymas 5 | Rastų taškų skaičius | Paketo dydis |
|---------------------------|-------------|------------|-------------|------------|-------------|----------------------|--------------|
| Ožiaragio puodelis | Nepriskirta | Priskirta | Nepriskirta | Priskirta | Nepriskirta | 19 | 48036 |

24 lentelė. Eksperimentai su Hesiano slenkščiu 3000

| Daiktas | Bandymas 1 | Bandymas 2 | Bandymas 3 | Bandymas 4 | Bandymas 5 | Rastų taškų skaičius | Paketo dydis |
|---|------------|-------------|------------|-------------|------------|----------------------|--------------|
| Panda | Priskirta | Nepriskirta | Priskirta | Nepriskirta | Priskirta | 10 | 26359 |
| Vėžlys | Priskirta | Priskirta | Priskirta | Priskirta | Priskirta | 14 | 35933 |
| Meškiukas | Priskirta | Priskirta | Priskirta | Priskirta | Priskirta | 24 | 60072 |
| Knyga "Oracle PL/SQL Programming 2nd Edition" | Priskirta | Priskirta | Priskirta | Priskirta | Priskirta | 9 | 23975 |
| Knyga "Cryptanalysis: A Study of Ciphers and Their Solution" | Priskirta | Priskirta | Priskirta | Priskirta | Priskirta | 40 | 98497 |
| Knyga "Karo menas" | Priskirta | Priskirta | Priskirta | Priskirta | Priskirta | 35 | 86332 |
| Knyga "Pasąmonės galia" | Priskirta | Priskirta | Priskirta | Priskirta | Priskirta | 23 | 57668 |
| Puodelis su tigrū | Priskirta | Priskirta | Priskirta | Priskirta | Priskirta | 29 | 72056 |
| Puodelis su rože | Priskirta | Priskirta | Priskirta | Priskirta | Priskirta | 23 | 57649 |
| Jacobs puodelis | Priskirta | Priskirta | Priskirta | Nepriskirta | Priskirta | 39 | 95973 |
| Ožiaragio puodelis | Priskirta | Priskirta | Priskirta | Priskirta | Priskirta | 64 | 156003 |

Ekperimentuojant buvo pastebėta, kad sumažinus Hesiano slenkstį objektai buvo klasifikuojami tiksliau. Tačiau dėl padidėjusio aptiktų taškų skaičius labai išaugo siunčiamų JSON paketų dydis (buvo pasiektas maksimalus 156 kilobaitų paketo dydis).

Sumažinus Hesiano slenkstį nebuvo pastebėtas ryškus objekto nuotraukos apdorojimo sulėtėjimas.

Nors kuriant Android OS kliento programinę įrangą buvo siekiama, kad elektros energija būtų naudojama kiek galima efektyviau, to visiškai pasiekti nepavyko. Testavimui naudotas „HTC Evo 3D“ telefonas turi galingą bateriją (1730 mAh), tačiau pasinaudojus

klientu daugiau nei 6 valandas, buvo pastebėta, kad baterijos lygis sumažėjęs iki 30%. Tai įtakoja belaidžio ryšio (*Wi-Fi*) tinklo, kameros, bei intensyvus procesoriaus (apdorojant nuotrauką) naudojimas.

9.7 UAB „Informacinių technologijų organizacija“ įdiegimo aktas



UAB „INFORMACINIŲ TECHNOLOGIJŲ ORGANIZACIJA“

Savanorių pr. 184, 44150 Kaunas

Tel.: 8~671 1 39 58

Faks.: 8~37 300 489

El.paštas: info@ito.lt

PROGRAMŲ SISTEMOS PERDAVIMO IR APROBAVIMO AKTAS

20 11 m. gruodžio 12 d.

Programų sistemos pavadinimas „Universali vaizdo atpažinimo sistema“

Kūrinio tipas programinė įranga

Programų sistemos sukūrimo data 20 11 m. gruodžio 12 d.

Kūrinio įteikimo UŽSAKOVUI data 20 11 m. gruodžio 12 d.

Užsakovo arba trečiojo asmens Kūrinio aprobavimo rezultatas:

Programinė įranga sukurta ir įdiegta sėkmingai. Darbas priimtas ir laikomas tinkamu.

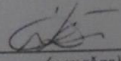
Kūrinio aprobavimo data 20 11 m. gruodžio 12 d.

Kūrinio originalo saugotojas - UAB „Informacinių technologijų organizacija“

AUTORIUS

Liudas Drejeris

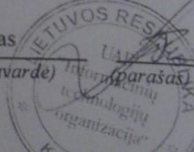
(vardas, pavardė)


(parašas)

Paulius Lazauskas

(vardas, pavardė)

UŽSAKOVAS



Paulius Lazauskas

Direktorius

9.8 VI „Registru centras“ įdiegimo aktas



VALSTYBĖS ĮMONĖ REGISTRŲ CENTRAS
Vincio Kudirkos g. 18-3, 03105 Vilnius,
tel. (8 5) 268 8202, faks. (8 5) 268 8311, el. p. info@registrucentras.lt.
Duomenys kaupiami ir saugomi Juridinių asmenų registre, kodas 124110246

Kauno Technologijos universitetas
Informatikos fakultetas
Studentų 50-411, LT 3031 Kaunas
Tel. (8 37) 300350
Faksas (3 37) 300352

PROGRAMŲ SISTEMOS PERDAVIMO IR APROBAVIMO AKTAS

2011 m. Gruodžio 19 d.

Programų sistemos pavadinimas „Universali vaizdo atpažinimo sistema“

Kūrinio tipas Programinė įranga

Programų sistemos sukūrimo data 2011 m. Gruodžio 15 d.

Kūrinio įteikimo UŽSAKOVUI data 2011 m. Gruodžio 19 d.

Užsakovo arba trečiojo asmens Kūrinio aprobavimo rezultatas:

Sistema įdiegta sėkmingai ir atitinka iškeltus reikalavimus.

Kūrinio aprobavimo data 2011 m. Gruodžio 19 d.

Kūrinio originalo saugotojas Ernestas Kardzys

AUTORIUS

Ernestas Kardzys

(vardas, pavardė)

E. Kardzys
(parašas)

UŽSAKOVAS

Algirdas Remeikis

(vardas, pavardė)

Remeikis
(parašas)

Valstybės įmonės Registrų centro
Informacinių technologijų centro
IS konstravimo departamento viršininkas
Asmens pareigų pavadinimas

Antanas Krikščiūnas
(Parašas)

Antanas Krikščiūnas
(Vardas ir pavardė)

Parengė
Algirdas Remeikis
2011 12 19

Remeikis

9.9 KTU Technoramos padėkos raštas

