

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Andrius Paulavičius

Automatinis vaizdų jungimas į panoramas

Magistro darbas

Darbo vadovas

doc. dr. Eimutis Karčiauskas

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Automatinis vaizdų jungimas į panoramas

Magistro darbas

Recenzentas

dr. Š. Packevičius

2010-05-

Vadovas

doc. dr. E. Karčiauskas

2010-05-

Atliko

IFM 4/1 gr. stud. A. Paulavičius

2010-05-

Kaunas, 2010

Santrauka

Šiame darbe pateikiamas apibendrintas automatinio vaizdų jungimo į panoramas algoritmas, detalai aptariami algoritmo žingsniai ir galimas našumo didinimas lygiagretaus programavimo priemonėmis. Tyrime aprašyti vaizdų jungimo bandymai panaudojant kelias populiarių vaizdų bruožų alternatyvas. Taip pat pristatoma praktinė vaizdų jungimą panaudojanti aplikacija.

Vaizdų jungimas į panoramas yra plačiai ištyrinėta kompiuterinio matymo problema. Ilgą laiką vaizdų jungimui buvo būtinos atsargiai nustatytais parametrais padarytos nuotraukos, rankinis parametru nustatymas programinėje įrangoje ar specializuota ir brangi fotografinė įranga. Šiuolaikiniai, klaidoms atsparūs vaizdų bruožai leidžia automatizuoti vaizdų tarpusavio padėčių nustatymą ir sujungti fotografijas, padarytas su įprastais, vartotojo lygio, rankose laikomais fotoaparatais, iš vartotojų nereikalaujant papildomų nustatymų.

Summary

Automatic image stitching into panoramas

This study presents a generalization of an automated image stitching algorithm, describes its steps in great detail and discusses possible performance improvements by use of parallel execution. Results of stitching experiments using a couple of modern and popular image feature alternatives are shown. A practical application using an implementation of automated image stitching is presented.

Image stitching into panoramas is a well researched problem in computer vision. Until recently image stitching required photographs taken with carefully calibrated parameters, manual positioning in software or specialized and expensive photography equipment. With the advent of more robust image features, it is now possible to automate image parameter estimation and stitch images taken with regular, hand held, consumer level cameras, with no additional user input.

Turinys

1. Įvadas.....	7
2. Vaizdų sujungimas.....	8
2.1. Vaizdų sujungimo metodai.....	8
2.2. Keturių parametų fotoaparato modelis.....	8
2.3. Automatizuotas vaizdų sujungimas.....	9
2.3.1. Vaizdų bruožų sutapatinimas.....	9
2.3.2. Persidengiančių vaizdų radimas.....	10
2.3.3. Fotoaparatų parametų nustatymas.....	11
2.4. Vaizdų suliejimas	16
2.4.1. Sujungimas be suliejimo.....	16
2.4.2. Tiesinis suliejimas.....	16
2.4.3. Daugiajuostis suliejimas.....	17
2.5. Nepilnų panoramų apdorojimas.....	21
2.5.1. Tekstūrų sintezė.....	21
2.6. Vaizdų bruožai.....	22
2.6.1. SIFT bruožai.....	23
2.6.2. SURF bruožai.....	25
3. Projektuojama sistema.....	27
3.1. Funkciniai reikalavimai.....	27
3.2. Panašių produktų tyrimas.....	29
3.3. Panoraminų vaizdų vizualizacija.....	31
3.3.1. Vizualizacijos priemonės naršyklėse.....	32
3.4. Architektūra.....	33
3.4.1. Vaizdų jungimo komponento architektūra.....	34
3.4.2. Web aplikacijos architektūra.....	38
3.4.3. Duomenų bazės architektūra.....	39
3.5. Vartotojo sąsaja.....	41
3.6. Testavimas.....	44
3.7. Rezultatai.....	45
4. Eksperimentai.....	47
4.1. Veikimo laiko įvertinimas.....	47
4.2. Sujungimo kokybė.....	47
4.3. Palyginimas su alternatyvomis.....	48
5. Išvados.....	50
6. Literatūra.....	51
7. Terminų ir santrumpų žodynas.....	54
8. Priedai.....	55
8.1. Konferencijos „Information Society and University studies“ 2010 pranešimo medžiaga.....	55

Paveikslų sąrašas

1 pav. Persidengiančių vaizdų bruožai. Vaizdų bruožai, sutapatinti bruožai, didžiausias homografija suderinamų sutapatinimų rinkinys.....	11
2 pav. Klaidoms atsparios atstumo funkcijos komponentės kai	13
3 pav. Panoramos fragmentas prieš ir po globalaus posūkio transformacijos.....	16
4 pav. Tiesinio vaizdų suliejimo komponentai. Pradiniai vaizdai , vaizdai bendrose sferinėse koordinatėse , vaizdų tiesio suliejimo svoriai	17
5 pav. Daugiajuosčio suliejimo komponentai. Persidengiančių vaizdų dažnių juostos ir jų svorinės funkcijos kai k kinta nuo 1 iki 3.....	19
6 pav. Vaizdų sujungimo strategijos. Jungimas be suliejimo, jungimas su tiesiniu suliejimu ir jungimas su daugiajuosčiu suliejimu.....	20
7 pav. Prasto panoramos skylių užpildymo tekstūrų sinteze pavyzdys.....	22
8 pav. Trūkstamos informacijos apdorojimo strategijos. Sujungta panorama su informacijos trūkumais, skylių užpildymas tekstūrų sinteze, pilnai užpildyto turinio iškirpimas.....	22
9 pav. SIFT bruožo apibūdinimas sudarytas iš gradiento histogramų aplink bruožo tašką.....	25
10 pav. Sistemos panaudojimo atvejai.....	29
11 pav. Hugin vartotojo sąsaja.....	30
12 pav. PTgui vartotojo sąsaja.....	30
13 pav. Panoweaver vartotojo sąsaja.....	31
14 pav. Sferinės panoramos vizualizavimas.....	32
15 pav. Sistemos komponentai.....	33
16 pav. Vaizdų jungimo programos klasių diagrama.....	35
17 pav. Vaizdas bendrose sferinėse koordinatėse, kai jis kerta koordinatės pradžia.....	37
18 pav. Web aplikacijos struktūra.....	38
19 pav. Esybių diagrama.....	39
20 pav. Duomenų bazės schema.....	40
21 pav. Bendras vartotojo sąsajos išdėstymas.....	41
22 pav. Panoramų išrinkimo vartotojo sąsaja.....	42
23 pav. Vartotojo profilio peržiūros vartotojo sąsaja.....	42
24 pav. Panoramos peržiūros vartotojo sąsaja.....	43
25 pav. Panoramos kūrimo vartotojo sąsaja.....	43
26 pav. Vartotojo sąsajos formos pavyzdys.....	44
27 pav. Projektui parašyto kodo eilučių kiekiai.....	46
28 pav. Veikimo laiko įvertinimas.....	47
29 pav. Sujungimo kokybė.....	48
30 pav. Panoramų, sukurtų iš to pačio vaizdų rinkinio skirtingomis programomis palyginimas. Projekte realizuota programa, Hugin, PTgui, Panoweaver.....	49

1. Įvadas

Panoramos yra didelio matymo kampo vaizdai. Panoramos, apimančios visą cilindrinę ar sferinę erdvę, esančią aplinkui stebėtoją, dažnai naudojamos kaip virtualios ekskursijos, leidžiant po jas žvalgytis. Detalus realios aplinkos atvaizdavimas virtualioje erdvėje yra labai vertingas tokiose srityse kaip turizmas. Panoramų kūrimą apsunkina aparatūrinių priemonių kaštai, programinių įrankių sudėtingumas, palaikomų sistemų apribojimai.

Tradiciškai panoraminiai vaizdai išgaunami sudėtingomis specializuotomis aparatūrinėmis priemonėmis, pvz. hiperboliniais veidrodžiais. Programinė įranga leidžia panoramas išgauti sujungiant įprastas nuotraukas. Vaizdų sujungimo algoritmai bando nustatyti vaizdų, su dalinai persidengiančiu turiniu, geometrinius santykius ir šiuos vaizdus sulieti, sprendžiant tokias problemas kaip kintančią ekspoziciją.

Panoraminių vaizdų sujungimas yra nuodugnai ištirtas[1, 2, 3, 4] ir taikomas komercinėse programose. Vaizdų jungimo problemos esmė – transformacijos matricos arba homografijos paieška kiekvienam vaizdui. Šis paieškos procesas įprastai reikalauja pradinių parametrų (dažniausiai vartotojams pele nurodant apytiksles padėtis) arba griežtos vaizdų tvarkos. Pavyzdžiui, PhotoStitch programinė įranga, tiekama kartu su Canon skaitmeniniais fotoaparatais, reikalauja arba griežto horizontalus perrinkimo, arba kvadratinės matricos nuotraukų tvarkos. Dauguma vaizdų jungimo programų turi vartotojo sąsają apytiksliam vaizdų pozicionavimui pele.

2. Vaizdų sujungimas

2.1. Vaizdų sujungimo metodai

Vaizdų sujungimo metodai skiriami į dvi kategorijas – tiesioginius [5, 6, 7] ir paremtus bruožais [8, 9, 10].

Tiesioginių metodų esmė – rasti tokias transformacijas tarp vaizdų, kurios minimizuotų persidengiančių vaizdų nepanašumo įverčius bendroje erdvėje. Šie metodai išnaudoja visą vaizdų informaciją, tačiau jiems būtina tiksliai parinkti pradinius parametrus.

Bruožais paremtų metodų esmė – rasti tokias transformacijas tarp vaizdų, kurios minimizuotų bruožų taškų, sutapatintų keliuose vaizduose, padėčių skirtumus bendroje erdvėje. Šiems metodams nebūtina tiksliai parinkti pradinius parametrus ir jais grįstų optimizuojamų tikslo funkcijų sudėtingumas daug mažesnis, nei grįstų tiesioginiais metodais. Bruožais pagrįsti metodai išnaudoja mažiau vaizduose laikomos informacijos. Tradiciniai vaizdų bruožai neturi invariantiškumo savybių, leidžiančių rasti geometrines transformacijas tarp neapibrėžta tvarka pateiktų vaizdų.

2.2. Keturių parametru fotoaparato modelis

Panoramos konstruojamos jungiant nuotraukas, darytas iš bendro optinio centro. Dažnai naudojamas plačiai ištyrinėtas vieno parametro modelis, kai fotoaparatas sukamas tik aplink Y ašį ir vaizdai sujungiami į cilindrinės panoramas.

Transformaciją tarp vaizdo koordinačių ir jų atitikmenų kitame vaizde nusako homografijos matrica:

$$\tilde{u}_i = H_{ij} \tilde{u}_j, \quad (1)$$

kur H_{ij} – homografija iš vaizdo j į vaizdą i , \tilde{u}_i ir \tilde{u}_j homogeninės koordinatės vaizduose i ir j .

Bendriausias vaizdo fotoaparato modelis nusakomas keturiais parametrais: posūkio kampų vektoriumi $\theta = [\theta_{ix} \ \theta_{iy} \ \theta_{iz}]$ ir židinio nuotoliu f . Tokiu atveju homografijos matricą galima išreikšti vaizdų posūkio matricų R_i ir mastelio keitimo matricų K_i sandaugomis:

$$H_{ij} = K_i R_i R_j^T K_j^{-1} \quad (2)$$

$$K_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$R_i = e^{\begin{bmatrix} 0 & -\theta_{iz} & \theta_{iy} \\ \theta_{iz} & 0 & -\theta_{ix} \\ -\theta_{iy} & \theta_{ix} & 0 \end{bmatrix}} = \begin{bmatrix} 2(x^2-1)s^2-1 & 2xys^2-2zcs & 2xzs^2+2yys \\ 2xys^2+2zcs & 2(y^2-1)s^2+1 & 2yys^2-2xcs \\ 2xzs^2-2yys & 2yys^2+2xcs & 2(z^2-1)s^2+1 \end{bmatrix}, \quad (4)$$

kur posūkio vektorius išreikštas normalizuota forma $[\theta_{ix} \ \theta_{iy} \ \theta_{iz}] = [x \ y \ z] \cdot \theta$, $c = \cos(\theta/2)$ ir $s = \sin(\theta/2)$.

2.3. Automatizuotas vaizdų sujungimas

Tiesioginiams vaizdų jungimo metodams privaloma pradinė kalibracija, todėl automatizuotam vaizdų jungimui galima remtis tik bruožais pagrįstais metodais. Pagal bruožų padėtis nustatytų vaizdų porų homografijų sujungimas sukauptų dideles paklaidas ir nebūtų įmanoma įvertinti sujungimo iš kelių pusių apribojimų. Dėl šių priežasčių, vaizdų fotoaparatus modelių parametru nustatymui reikia vienu metu vertinti visų persidengiančių vaizdų bruožus [11].

Galima išskirti tokius automatinio vaizdų jungimo etapus:

- Vaizdų transformacijoms invariantiškų bruožų išgavimas;
- Vaizdų bruožų sutapatinimas;
- Persidengiančių vaizdų radimas;
- Vaizdų fotoaparatus geometrinių parametru nustatymas;
- Vaizdų suliejimas bendroje erdvėje.

2.3.1. Vaizdų bruožų sutapatinimas

Kiekvienas vaizdas gali persidengti su bet kuriuo kitu ir tas pats bruožo taškas gali kartotis keliuose vaizduose, todėl kiekvienam bruožui reikia ieškoti kelių artimiausių kaimynų jų apibūdinimų erdvėje. Artimiausių kaimynų paieška didelės dimensijos erdvėje neturi efektyvaus algoritmo, pilnas perrinkimas reikalautų $O(n^2)$ palyginimų. Tipinis vaizdas turi šimtus bruožų ir kai kiekvieno bruožo aprašymo vektoriai didelės dimensijos (pvz. SIFT – 128), bruožų sutapatinimas pilnu perrinkimu gali virsti daugiausiai laiko reikalaujančiu algoritmo žingsniu. Bruožų sutapatinimą galima atlikti žymiai greičiau, apsiribojant apytiksliais artimiausiais kaimynais. Tai įmanoma padaryti panaudojant k-d medį. Šis medis yra dvejetainis erdvės išskaidymas, kuris rekursyviai erdvę dalina per vidurkius didžiausio kitimo dimensijoje. Naudojant šią duomenų struktūrą, apytiksliai artimiausius kaimynus įmanoma rasti per $O(n \log n)$ palyginimų [12].

2.3.2. Persidengiančių vaizdų radimas

Po bruožų sutapatinimo yra žinomos vaizdų poros, kurios turi daugiausiai artimų bruožų, o tuo pačiu ir didžiausią potencialiai persidengiantį plotą. Iš didžiausio persidengiančio vaizdų rinkinio bus sujungiamą galutinė panorama. RANSAC (RANDOM SAMPLE CONSENSUS) [13] metodu galima išrinkti didžiausius rinkinius sutapatintų bruožų, kurie yra geometriškai suderinami, t. y. tarp jų egzistuoja homografija (žr. pav. 1). Ar vaizdai persidengia galima spręsti iš RANSAC patvirtintų ir atmestų sutapatinimų santykio.

2.3.2.1. RANSAC metodas

RANSAC yra modelio parametrų nustatymo iš eksperimentinių duomenų metodas, kuriame modelio parametrai nustatomi iš atsitiktinai išrinkto minimalaus duomenų egzempliorių rinkinio. Procedūra kartojama daug kartų, ieškant geriausiai su eksperimentiniais duomenimis sutampančių parametrų. Kai bandymų kiekis pakankamai didelis, tikimybė rasti geriausius parametrus artėja į vieną, o atsitiktinis parinkimas užtikrina atsparumą klaidoms eksperimentiniuose duomenyse.

Homografijai tarp dviejų vaizdų nustatyti DLT (Direct Linear Transformation) metodu [14] reikia keturių šių vaizdų sutapatintų bruožų porų padėčių. Homografijos H elementai randami išsprendus tiesinę lygčių sistemą:

$$\begin{bmatrix} 0 & 0 & 0 & -u_{j1x} & -u_{j1y} & -1 & u_{i1x} \cdot u_{j1y} & u_{i1y} \cdot u_{j1x} \\ u_{j1x} & u_{j1y} & 1 & 0 & 0 & 0 & -u_{i1x} \cdot u_{j1x} & -u_{i1y} \cdot u_{j1x} \\ 0 & 0 & 0 & -u_{j2x} & -u_{j2y} & -1 & u_{i2x} \cdot u_{j2y} & u_{i2y} \cdot u_{j2y} \\ u_{j2x} & u_{j2y} & 1 & 0 & 0 & 0 & -u_{i2x} \cdot u_{j2x} & -u_{i2y} \cdot u_{j2x} \\ 0 & 0 & 0 & -u_{j3x} & -u_{j3y} & -1 & u_{i3x} \cdot u_{j3y} & u_{i3y} \cdot u_{j3y} \\ u_{j3x} & u_{j3y} & 1 & 0 & 0 & 0 & -u_{i3x} \cdot u_{j3x} & -u_{i3y} \cdot u_{j3x} \\ 0 & 0 & 0 & -u_{j4x} & -u_{j4y} & -1 & u_{i4x} \cdot u_{j4y} & u_{i4y} \cdot u_{j4y} \\ u_{j4x} & u_{j4y} & 1 & 0 & 0 & 0 & -u_{i4x} \cdot u_{j4x} & -u_{i4y} \cdot u_{j4x} \end{bmatrix} \cdot \begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \end{bmatrix} = \begin{bmatrix} -u_{i1y} \\ u_{i1x} \\ -u_{i2y} \\ u_{i2x} \\ -u_{i3y} \\ u_{i3x} \\ -u_{i4y} \\ u_{i4x} \end{bmatrix}$$

$$H_{33} = 1 \tag{5}$$

Kiekviename bandyme rastos homografijos kokybė įvertinama pagal tai, kiek vieno vaizdo sutapatintų bruožų padėčių transformuotų H yra artimi kito vaizdo bruožų padėtimis.



1 pav. Persidengiančių vaizdų bruožai. Vaizdų bruožai, sutapatinti bruožai, didžiausias homografija suderinamų sutapatinimų rinkinys

2.3.2.2. Persidengimo patvirtinimas

Ar vaizdai persidengia, galima spręsti iš RANSAC patvirtintų ir atmestų sutapatinimų santykio, naudojant tiesinę slenkstinę sąlygą:

$$n_i > \alpha + \beta \cdot n_f, \quad (6)$$

kur n_i - RANSAC rastą homografiją tenkinančių sutapatintų bruožų kiekis, n_f - RANSAC atmestų sutapatintų savybių kiekis.

2.3.3. Fotoaparatus parametru nustatymas

2.3.3.1. Reliatyvių būsenų nustatymas

Rastų homografijų sujungimas sukauptų dideles paklaidas. Kokybiškam būsenų radimui, reikia ieškoti visų vaizdų fotoaparatus parametru kartu. Tai galima atlikti ryšulinio koregavimo metodu [15].

Fotoaparatus parametru optimizavimo procedūra atliekama su geriausiai persidengiančių vaizdų pora ir kartojama įterpiant po naują vaizdą, geriausiai persidengiantį su jau optimizuotu rinkiniu. Naujai

įterpto vaizdo fotoaparato pradiniai parametrai nustatomi tokie patys kaip jau optimizuoto vaizdo, su kuriuo jis geriausiai persidengia. Persidengimo kokybė vertinama pagal tarpusavyje sutapatintų bruožų kiekį. Optimizuojama funkcija optimizavime dalyvaujančių vaizdų bruožus projektuoja į visus kitus optimizavime dalyvaujančius vaizdus, kuriuose yra šių bruožų atitikmenys ir sumuoja atstumų tarp jų kvadratus. Šią funkciją reikia minimizuoti fotoaparato parametrų atžvilgiu.

Sutapatinto bruožo taško i – ajame ir j – ajame vaizduose $u_i^k \Leftrightarrow u_j^l$ projekcijos skirtumas yra:

$$r_{ij}^k = u_i^k - p_{ij}^k, \quad (7)$$

kur p_{ij}^k yra taško atitinkančio u_i^k vaizde j projekcija vaizde i .

$$\tilde{p}_{ij}^k = K_i R_i R_j^T K_j^{-1} \tilde{u}_j^l \quad (8)$$

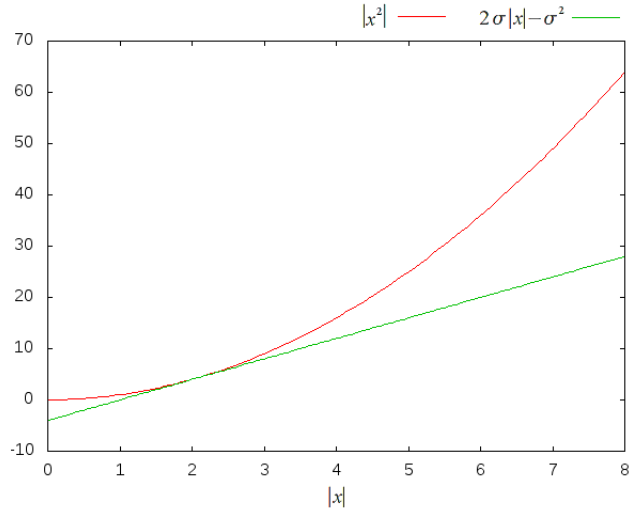
Optimizuojama funkcija:

$$e = \sum_{i=1}^n \sum_{j \in I(i)} \sum_{k \in F(i,j)} f(r_{ij}^k), \quad (9)$$

kur n – vaizdų kiekis, $I(i)$ – vaizdai persidengiantys su vaizdu i , $F(i,j)$ – sutapatinti vaizdų i ir j bruožų taškai. f – atsparumą klaidoms suteikianti funkcija.

$$f(x) = \begin{cases} |x^2| & \text{kai } |x| < \sigma \\ 2\sigma|x| - \sigma^2 & \text{kai } |x| \geq \sigma \end{cases} \quad (10)$$

Funkcija $f(x)$ suteikia atsparumą klaidoms duomenyse. Ji naudoja antro laipsnio normą greitam konvergavimui gerai sutapatintiems nariams ir pirmo laipsnio normą tolimiems nariams. Pirmos iteracijos metu $\sigma = \infty$, likusiųjų $\sigma = 2$ pikselius atitinkantis atstumas. Klaidos funkcijos komponentes iliustruoja pav. 2.



2 pav. Klaidoms atsparios atstumo funkcijos komponentės kai $\sigma=2$

Tai yra netiesinis mažiausių kvadratų uždavinys, kurį sprendžia Levenberg-Marquardt algoritmas [16]. Šis algoritmas yra Gauso-Niutono iteracinio optimizavimo algoritmo modifikacija, kurioje įvestas slopinimo parametras, pagreitinantis konvergavimą ir leidžiantis rasti minimumą prastai parinkus pradinius parametrus.

Levenberg-Marquardt algoritmo iteracija yra tiesinių lygčių sistemos sprendimas:

$$(J^T J + \lambda \cdot \text{diag}(J^T J))(\Delta \Theta) = J^T r, \quad (11)$$

kur Θ – visų fotoaparato parametrai surašyti į vieną vektorių, $\Delta \Theta$ – ieškomas parametru pokytis, r – projektavimo paklaidos surašytos į vieną vektorių, J – Jakobiano matrica $J = \frac{\delta r}{\delta \Theta}$, λ – slopinimo parametras, kurio vertė keičiama priklausomai nuo minimizuojamos funkcijos kitimo greičio (funkcijai smarkiai pagerėjus – mažinama, mažai – didinama), diag – matricos diagonalizavimo funkcija.

Jakobianą sudarančios išvestinės skaičiuojamos pagal jų analitines išraiškas. Pvz., išvestinė pagal i – tojo fotoaparato parametru θ_x :

$$\frac{\delta r_{ij}^k}{\delta \theta_{ix}} = 0 - \frac{\delta p_{ij}^k}{\delta \theta_{ix}} = - \left(\frac{\delta p_{ij}^k}{\delta \tilde{p}_{ij}^k} \frac{\delta \tilde{p}_{ij}^k}{\delta \theta_{ix}} \right) \quad (12)$$

$$\frac{\delta p_{ij}^k}{\delta \tilde{p}_{ij}^k} = \frac{\delta [x/z, y/z]}{\delta [x, y, z]} = \begin{bmatrix} 1/z & 0 & -x/z^2 \\ 0 & 1/z & -y/z^2 \end{bmatrix} \quad (13)$$

$$\frac{\delta \tilde{p}_{ij}^k}{\delta \theta_{ix}} = K_i \frac{\delta R_i}{\delta \theta_{ix}} R_j^T K_j^{-1} \tilde{u}_j \quad (14)$$

$$\frac{\delta R_i}{\delta \theta_{ix}} = \frac{\delta e^{[\theta_i]}}{\delta \theta_{ix}} = e^{[\theta_i]} \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (15)$$

Išvestinės pagal kitus parametrus gaunamos analogiškai. Jose besiskiriančios komponentės:

$$\frac{\delta R_i}{\delta \theta_{ix}} = \frac{\delta e^{[\theta_i]}}{\delta \theta_{ix}} = e^{[\theta_i]} \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (16)$$

$$\frac{\delta R_i}{\delta \theta_{iy}} = e^{[\theta_i]} \cdot \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (17)$$

$$\frac{\delta R_i}{\delta \theta_{iz}} = e^{[\theta_i]} \cdot \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (18)$$

$$\frac{\delta K_i}{\delta \theta_{if}} = \frac{\delta \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix}}{\delta f_i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (19)$$

$$\frac{\delta R_j^T}{\delta \theta_{jx}} = (e^{[\theta_j]})^T \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad (20)$$

$$\frac{\delta R_j^T}{\delta \theta_{jy}} = (e^{[\theta_j]})^T \cdot \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (21)$$

$$\frac{\delta R_j^T}{\delta \theta_{jz}} = (e^{[\theta_j]})^T \cdot \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (22)$$

$$\frac{\delta K_j^{-1}}{\delta \theta_{jf}} = \frac{\delta \begin{bmatrix} f_j^{-1} & 0 & 0 \\ 0 & f_j^{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix}}{\delta f_j} = \begin{bmatrix} -f_j^{-2} & 0 & 0 \\ 0 & -f_j^{-2} & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (23)$$

Praktiškai vienas vaizdas persidengia su nedideliu kiekiu kitų vaizdų. Tai reiškia, kad Jakobiano matrica yra praretinta ir optimizavimo algoritmą galima paspartinti, neskaičiuojant didelių matricių sandaugų, o užpildant sudaugintos formos submatricas:

$$(J^T J)_{4i:4(i+1), 4j:4(j+1)} = \sum_{k \in F(i, j)} \left(\frac{\delta r_{ij}^k}{\delta \Theta_i} \right)^T \frac{\delta r_{ij}^k}{\delta \Theta_j} \quad (24)$$

$$(J^T r)_{4i:4(i+1)} = \sum_{j \in I(i)} \sum_{k \in F(i, j)} \left(\frac{\delta r_{ij}^k}{\delta \Theta_i} \right)^T r_{ij}^k. \quad (25)$$

2.3.3.2. Globalios būsenos nustatymas

Po reliatyvių fotoaparataų būsenų nustatymo lieka nežinoma fotoaparataų padėtis pasaulio koordinatinių atžvilgiu. Jei vieno fotoaparato posūkio parametrus laikome globaliais, išėjime galima gauti banguotą panoramą, nes mažai tikėtina, kad fotografavimo metu fotoaparatas buvo laikomas tobulai tiesiai (žr. pav. 3). Šį efektą galima pataisyti naudojant euristicą, kad įprastai fotografuojant panoramas fotoaparatas nesukamas horizonto atžvilgiu, dėl ko X ašių vektoriai atsiduria plokštumoje[10]. Surašius visų fotoaparataų posūkių matricų X ašių kryptis, reliatyvias vienam pasirinktam fotoaparatai, į matricą X, galima rasti jiems statmeną vektorių Y, išsprendžiant lygčių sistemą:

$$(X X^T) Y = [0 0 0]^T. \quad (26)$$

Globalią posūkio būseną nusako pasirinkto fotoaparato posūkio transformacija padauginta iš transformacijos pasukančios Y iki vertikalios padėties t. y. posūkio per ašį $A = Y \times [0 \ 1 \ 0]^T$ kampu $\theta = \arccos(Y \cdot [0 \ 1 \ 0]^T)$.

Posūkio kampu per ašį transformacijos matrica T, sudaryta pagal Rodriguez'o formuluotę:

$$T = I \cdot \cos(\theta) + \begin{bmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{bmatrix} \cdot \sin(\theta) + (1 - \cos(\theta)) \cdot A A^T, \quad (27)$$

kur I – vienetinė matrica, A – posūkio ašis, θ – posūkio kampas.



3 pav. Panoramos fragmentas prieš ir po globalaus posūkio transformacijos

2.4. Vaizdų suliejimas

Žinant n vaizdų $I^i(x, y)$ fotoaparato parametrus galima juos pervesti į bendras sferines koordinates $I^i(\theta, \phi)$, trečią sferinių koordinačių parametą, nusakantį atstumą, laikant konstanta. Vaizdus sujungiant į galutinę panoramą $I(\theta, \phi)$, svarbu panaudoti kokybišką suliejimo metodą ekspozicijos pokyčių, optinių vaizdų iškraipymų ir padėčių paklaidų kompensavimui. Kelios suliejimo strategijos pavaizduotos pav. 6.

2.4.1. Sujungimas be suliejimo

Paprasčiausias $I(\theta, \phi)$ sudarymas gali vykti kiekvienam išėjimo taškui išrenkant jį geriausiai atitinkantį tašką iš vieno atskiro vaizdo:

$$I(\theta, \phi)_{\text{be suliejimo}} = I^i(\theta, \phi), \text{ kai } W^i(\theta, \phi) = \max(W(\theta, \phi)) \quad (28)$$

Kadangi mažiausi optiniai iškraipymai pasireiškia nuotraukų centruose, pasirinkta svorinė funkcija $W^i(\theta, \phi)$ lygi 1 vaizdo centre ir tiesiškai mažėja iki 0 kraštuose.

2.4.2. Tiesinis suliejimas

Tiesinis suliejimas leidžia kompensuoti nevienodą ekspoziciją tarp persidengiančių vaizdų. Panaudojama visa persidengiančių vaizdų informacija, jų svarbumą įvertinant pagal svorines funkcijas $W^i(\theta, \phi)$. Tiesinio suliejimo komponentus iliustruoja pav. 4.

Tiesiniu suliejimu išėjimo vaizdas gaunamas taip:

$$I(\theta, \phi)_{\text{tiesinis}} = \frac{\sum_{i=1}^n I^i(\theta, \phi) W^i(\theta, \phi)}{\sum_{i=1}^n W^i(\theta, \phi)} \quad (28)$$



4 pav. Tiesinio vaizdų suliejimo komponentai. Pradiniai vaizdai $I^i(x, y)$, vaizdai bendrosė sferinėse koordinatėse $I^i(\theta, \phi)$, vaizdų tiesio suliejimo svoriai $W^i(\theta, \phi)$

2.4.3. Daugiajuostis suliejimas

Tiesinis suliejimas nėra atsparus klaidoms ir esant objektyvo sukeliams iškreipimams, judesiui ar mažom fotoaparatu parametru nustatymo paklaidoms, suliejime aukšto dažnio detalės tampa miglotos. Šią problemą galima dalinai išspręsti naudojant daugiajuostį suliejimą [17]. Metodo idėja yra suskaidyti vaizdus į juostas pagal dažnius, suliejant žemo dažnio juostas dideliuose plotuose, o aukšto dažnio mažuose plotuose.

Didžiausių dažnių juostos ir jų svoriai suformuojami taip:

$$B_1^i(\theta, \phi) = I^i(\theta, \phi) - I^i(\theta, \phi) * g_\sigma(\theta, \phi) \quad (29)$$

$$W_{max}^i(\theta, \phi) = \begin{cases} 1, & \text{kai } W^i(\theta, \phi) = \max(W(\theta, \phi)) \\ 0, & \text{kitu atveju} \end{cases} \quad (30)$$

$$W_1^i(\theta, \phi) = W_{max}^i(\theta, \phi) * g_\sigma(\theta, \phi), \quad (31)$$

kur $g_\sigma(\theta, \phi)$ – 2D Gauso funkcija su standartiniu nuokrypiu σ , pervesta į sferines koordinates, $B_1^i(\theta, \phi)$ – dažnių juosta kurios bangų ilgis nuo 0 iki σ , $W_1^i(\theta, \phi)$ – vaizdo i svorinė funkcija bangų ilgiui nuo 0 iki σ , * – sąsūkos operacija.

Tarpinės dažnių juostos ir jų svoriai suformuojamos taip:

$$I_{k+1}^i(\theta, \phi) = I_k^i(\theta, \phi) * g_\sigma(\theta, \phi) \quad (32)$$

$$B_{(k+1)}^i(\theta, \phi) = I_k^i(\theta, \phi) - I_{k+1}^i(\theta, \phi) \quad (33)$$

$$W_{(k+1)}^i(\theta, \phi) = W_k^i(\theta, \phi) * g_\sigma(\theta, \phi). \quad (34)$$

Paskutinė juosta sudaroma iš visų dar nepaskirstytų žemų dažnių:

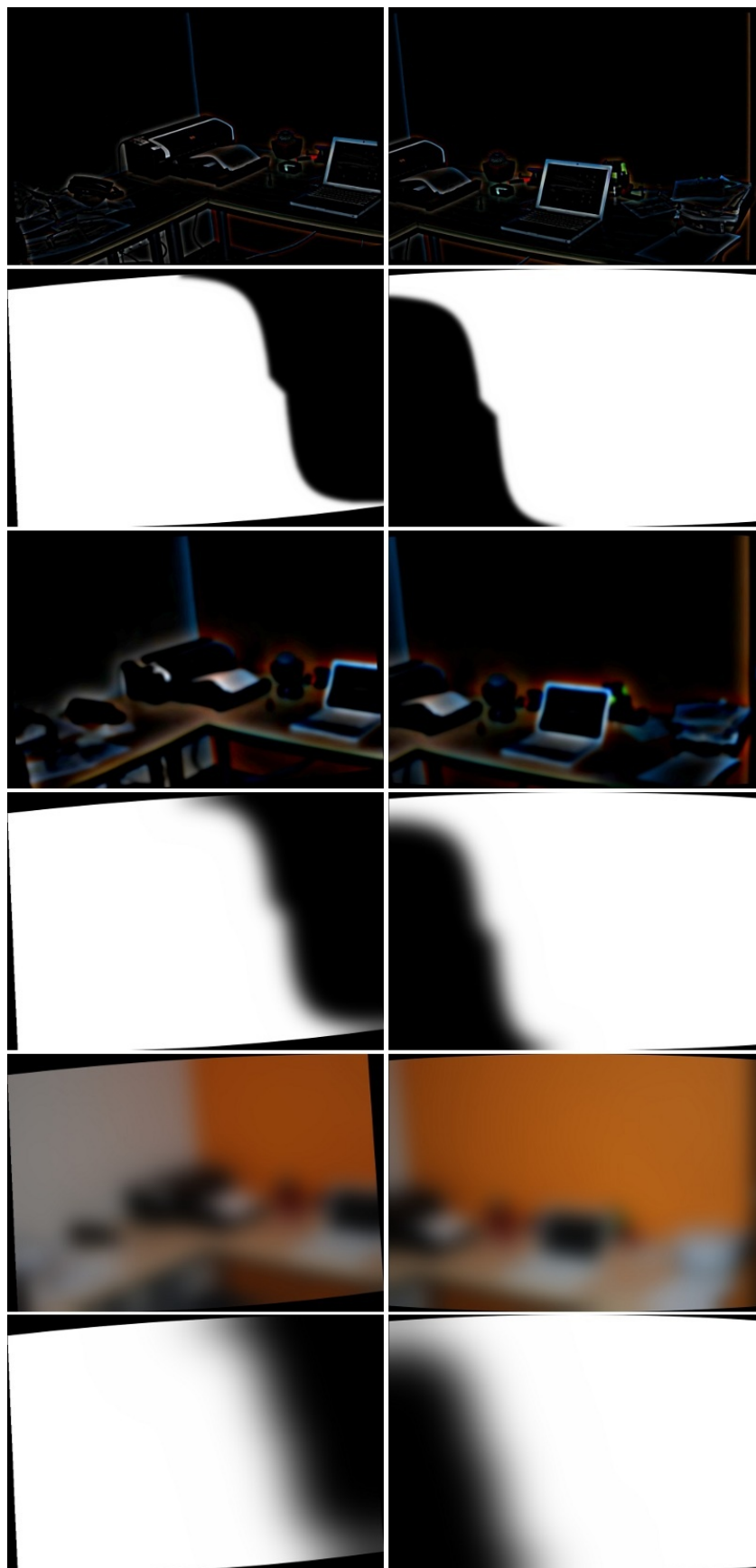
$$B_{(paskutinis)}^i(\theta, \phi) = I_{paskutinis-1}^i(\theta, \phi). \quad (35)$$

Daugiajuosčio suliejimo komponentes iliustruoja pav. 5.

Kiekvienai juostai persidengiantys vaizdai tiesiškai suliejami, o galinis rezultatas gaunamas susumavus visų juostų vertes:

$$B(\theta, \phi)_k = \frac{\sum_{i=1}^n B_k^i(\theta, \phi) W_k^i(\theta, \phi)}{\sum_{i=1}^n W_k^i(\theta, \phi)} \quad (36)$$

$$I(\theta, \phi)_{daugiajuostis} = \sum_{i \in k} B(\theta, \phi)_i. \quad (37)$$



5 pav. Daugiajuosčio suliejimo komponentai. Persidengiančių vaizdų dažnių juostos $B_k^i(\theta, \phi)$ ir jų svorinės funkcijos $W_k^i(\theta, \phi)$ kai k kinta nuo 1 iki 3



6 pav. Vaizdų sujungimo strategijos. Jungimas be suliejimo, jungimas su tiesiniu suliejimu ir jungimas su daugiajuosčiu suliejimu

2.5. Nepilnų panoramų apdorojimas

Kai jungiami vaizdai nepilnai padengia visą sferinę erdvę, panoramoje atsiranda skylės. Kai skylės nepageidautinos panoramą galima iškirpti iki pilnai užpildyto turinio ribų arba skylių turinį eksterpoliuoti. Skirtingas panoramų su skylėmis apdorojimo strategijas iliustruoja pav. 8.

2.5.1. Tekstūrų sintezė

Tekstūrų sintezė yra vaizdų generavimas pasinaudojant mažu pavyzdiniu vaizdu, kuriame ieškoma dėsningumų didelio vaizdo užpildymui. Tekstūrų sintezės neparimetrizuotais bandymais metodas [18] gali būti panaudotas vaizdo skylių užpildymui.

Šio metodo esmė – pasinaudoti sintezuojamo pikselio kaimynų intensyvumų kvadratinio langu, rasti šiam langui artimiausią bandymų vaizde ir sintezuojamam pikseliui priskirti reiškę iš rasto lango centro. Langų atstumas įvertinamas pagal juos sudarančių atitinkamų padėčių elementų skirtumų kvadratų sumą. Reikšmės lango kraštuose mažiau svarbios nei jo centre, tad elementų skirtumas įvedama svorinė funkcija pagal jų padėtį paieškos lange.

Atstumas tarp pikselių langų d įvertinamas taip:

$$d = \sum_{i \in (A-B)*G} i^2, \quad (38)$$

kur A – vieno pikselių lango intensyvumų matrica, B – kito pikselių lango intensyvumų matrica, G – dvimatė Gauso funkcija, $*$ – atitinkamų padėčių matricių elementų sandauga.

Panoramos skylės galima užpildyti kaip bandymų vaizdą panaudojus visus užpildytus panoramos plotus. Sintezę pradedama vykdyti neužpildytiems pikseliams, kurie yra užpildytų kaimynai. Procedūra kartojama kol visi pikseliai užpildomi.

Tipinėse panoramose, jei skylės nėra mažos, tekstūrų sintezė sugeneruoja nenatūralias struktūras arba pasiremdama naujai sukurtomis reikšmėmis pradeda generuoti šiukšles (žr. pav. 7).



7 pav. Prasto panoramos skylių užpildymo tekstūrų sinteze pavyzdys



8 pav. Trūkstamos informacijos apdorojimo strategijos. Sujungta panorama su informacijos trūkumais, skylių užpildymas tekstūrų sinteze, pilnai užpildyto turinio iškirpimas

2.6. Vaizdų bruožai

Lokalūs fotometriniai vaizdų bruožai sėkmingai taikomi objektų atpažinimui, tekstūrų atpažinimui ir vaizdų jungimui. Paprasčiausias bruožas yra taškų intensyvumų vektorius, tarp tokių vektorių panašumą galima įvertinti tarpusavio koreliacija. Intensyvumų vektorių bruožai reikalauja didelės vektorių dimensijos, todėl operacijų su jais sudėtingumas yra per aukštas praktiniam panaudojimui. Išskiriami 4 vaizdų bruožų tipai:

- Pasiskirstymu grįsti bruožai. Šios grupės bruožai naudoja histogramas atvaizduoti skirtingas išvaizdos charakteristikas ar formą. Paprasčiausias bruožas yra taškų intensyvumų pasiskirstymas atvaizduotas histograma. Zabih ir Woodfill [19] pasiūlė metodą atsparų apšvietumo pokyčiams. Jis remiasi taškų tvarkos ir intensyvumo santykių histogramomis. Šis metodas taip pat reikalauja didelės aprašymo vektoriaus dimensijos. Lowe [20] pasiūlė masteliui invariantišką bruožų transformaciją(SIFT), kuri apjungia masteliui invariantiškų regionų atpažinimą ir tų regionų aprašymą paremtą gradiento pasiskirstymu.
- Erdviniai - dažniniai bruožai. Daug metodų aprašo vaizdo dažnių sudėtį. Furje transformacija vaizdo turinys išskaidomas į bazines funkcijas. Pritaikymo lokaliai aprašymui problemas sprendžiama Gaboro transformacija.
- Diferencialiniai bruožai. Rinkinys vaizdo išvestinių apskaičiuotų iki norimo laipsnio apytiksliai nusako taško kaimynus. Florack ir kt. [21] pasiūlė diferencialinius invariantus, kurie apjungia lokalių išvestinių komponentes, kad išgautų invariantiškumą posūkiams. Stabilus išvestinių aproksimavimas gaunamas panaudojus sąsūką su Gauso funkcijų išvestinėmis.
- Kiti metodai. Van Gool ir kt. [22] pasiūlė apibendrintus momentų invariantus. Momentai apibūdina regiono formos ir intensyvumo pasiskirstymą

Vaizdų bruožai invariantiški tik ribotam kiekiui vaizdų transformacijų tipų, dauguma pasižymi invariacija tik poslinkiui plokštumoje. Sferinių panoramų jungimui reikia bruožų kurie būtų invariantiški mastelio ir žiūrėjimo kampo pokyčiams, kuriuos, kai posūkio kampas pakankamai mažas, galima interpretuoti kaip afiniasias transformacijas. Tobuli vaizdų bruožai turėtų būti invariantiški apšvietimo pokyčiams ir afiniosioms transformacijoms, taip pat jie turėtų gerai išsiskirti tarpusavyje, kokybiškai kaimynų paieškai.

Remiantis Mikolajczyk ir kt. [23] lokalių bruožų tyrimų rezultatais, bendru atveju tiksliausi pagal teisingų ir neteisingų sutapatintų bruožų kiekių santykį veikia SIFT bruožai.

2.6.1. SIFT bruožai

SIFT(Scale Invariant Feature Transform) [24] bruožai yra rutulio formos vaizdo plotai, turintys padėtį, mastelį ir kryptį. Bruožų paieška vykdoma įvairiuose masteliuose ir padėtyse, todėl SIFT apibūdinimai yra invariantiški postūmiui, posūkiui ir mastelio keitimo transformacijoms.

2.6.1.1. SIFT bruožų radimas.

SIFT bruožai randami ieškant ekstremumų vaizdo ir Gauso filtrų sąsūkų skirtumų (DoG) erdvėje, kuri gali būti interpretuojama, kaip Laplaso erdvės aproksimacija. Jos vaizdai gaunami sudarant vaizdų piramidę atliekant sąsūkas su Gauso funkcijomis ir imant artimų piramidės narių skirtumus.

DoG vaizdas $D(x,y,\sigma)$ gaunamas taip:

$$D(x,y,\sigma) = L(x,y,k_i\sigma) - L(x,y,k_j\sigma) \quad (39)$$

$$L(x,y,k\sigma) = G(x,y,k\sigma) * I(x,y), \quad (40)$$

kur $I(x,y)$ - vaizdas, $G(x,y,k\sigma)$ - 2D Gauso funkcija.

Sąsukos vaizdai sugrupuojami oktavomis, atitinkančiomis σ padvigubėjimą. k parenkama taip, kad būtų gautas fiksuotas vaizdų kiekis oktavoje. DoG vaizdai gaunami imant gretimų oktavos Gauso filtrais sulietų vaizdų skirtumus.

Sugeneravus DoG vaizdus, bruožų taškai yra randami pagal lokalius ekstremumus keliuose masteliuose. Tai atliekama palyginant kiekvieną DoG vaizdų tašką su 8 kaimynais tame pačiame mastelyje ir 9 atitinkamais kaimyniniais taškais kiekviename kaimyniniame mastelyje. Jei taškas atitinka minimumą arba maksimumą pagal visus lyginamus jis laikomas galimu bruožo taško kandidatu. Tokiu išrinkimu gaunama daug taškų, kurių dalis yra nestabilūs. Žemo kontrasto taškai yra atmetami pagal slenkstinę funkciją, taip pat atmetami taškai kurių aplinkos glodumas itin žemas.

2.6.1.2. SIFT bruožo apibūdinimas.

SIFT bruožo apibūdinimai yra 4x4 gradiento krypčių histogramos išsidėsčiusios aplink bruožo tašką, su 8 kryptimis kiekvienoje. Šios histogramos yra skaičiuojamos bruožui atitinkamo mastelio ir orientacijos vaizdo gradiento vertėms aplinkui bruožo tašką. SIFT bruožo apibūdinimo vektorius sudarymą iš gradiento iliustruoja pav. 9.

Gradiento dydžiai ir kryptys nustatoma taip:

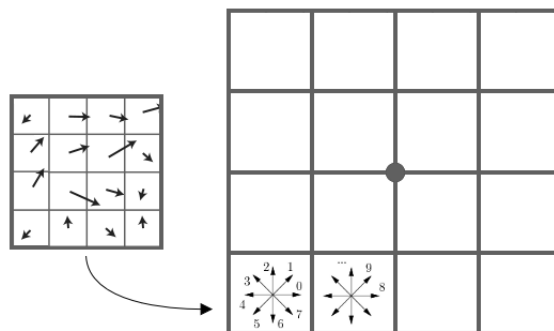
$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (41)$$

$$\theta(x,y) = \tan^{-1} \left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \right), \quad (42)$$

čia $m(x,y)$ - gradiento vektorius ilgis, $\theta(x,y)$ - gradiento vektorius kryptis, $L(x,y)$ - DoG skaičiavimui panaudotas rasto ekstremumo mastelio L .

Ilgiausias toks gradiento vektorius laikomas viso bruožo orientacija. Histogramų kryptys užrašomos reliatyviai bruožo orientacijos, tuo išgaunamas invariantiškumas posūkio transformacijoms. Histogramos vertės padauginamos iš svorinės Gauso funkcijos su $\sigma = 1.5$ kartų bruožo taško mastelio, kas suteikia didesnius svorius elementams arčiau bruožo taško. Iš histogramų sudarytas apibūdinimo vektorius normalizuojamas, kas suteikia invariacijos apšvietimo pokyčiams. Netiesinio apšvietimo efektų eliminavimui, vektoriaus reikšmės apribojamos iki 0.2 ir jis pakartotinai normalizuojamas.

128 elementų apibūdinimo vektorius yra gan ilgas, tačiau sutapatinto savybės yra teisingos 50% atveju kai fotoaparo posūkis neviršija 50 laipsnių. SIFT apibūdinimai yra invariantiški nedideliems afiniems fotoaparato parametrų pokyčiams.



9 pav. SIFT bruožo apibūdinimas sudarytas iš gradientų histogramų aplink bruožo tašką

2.6.2. SURF bruožai

SURF(Speeded Up Robust Features) [25] bruožai sukurti stengiantis paspartinti bruožų išgavimą. Jų paieška vykdoma naudojant 2D Haro bangelių filtrus Hesiano determinanto aproksimavimui. SURF išgavimas efektyviai išnaudoja tarpinius skaičiavimo vaizdus ir yra kelis kartus greitesnis už SIFT.

2.6.2.1. SURF bruožų radimas

SURF apibūdinimas pagrįstas Hesiano (antro laipsnio išvestinių) matricos determinantu, kuris aproksimuojamas Haro bangelėmis. SURF skirtingiems masteliams išgauti nenaudoja tradicinės rekursyviai Gauso filtrais sulietų vaizdų piramidės, o tiesiog naudoja skirtingo dydžio atpažinimo filtrus.

2.6.2.2. SURF bruožų apibūdinimas

SURF bruožų apibūdinimai aprašo kaip taškų intensyvumai pasiskirstę atitinkamo mastelio SURF bruožo aplinkoje. Šis aprašymas panašūs į SIFT, tačiau naudojami Haro bangelių filtrai, kas padidina toleranciją klaidom ir reikalauja mažiau skaičiavimo laiko. Haro bangelės yra filtrai kurie gali būti panaudoti rasti horizontalius ar vertikalius gradientus. Apibūdinimai išgaunami visiem bruožų taškams priskiriant orientaciją ir konstruojant nuo mastelio priklausančius langus kuriuose sudaromi 64-dimensijų vektoriai. Visi skaičiavimai atliekami reliatyviai surasto taško masteliui r .

Orientacijos išgavimui suskaičiuojama taškų, esančių $6r$ atstumu nuo rasto bruožo taško žingsniais r , reakcija į $4r$ ilgio Haro bangeles. Rezultatas padauginamas iš svorinės Gauso funkcijos su standartiniu nuokrypiu $2,5r$. Bruožo dominuojanti kryptis nustatoma sumuojant visas reakcijas, jų sudaromą vektorinę erdvę sukant iki $\frac{\pi}{3}$.

SURF bruožo apibūdinimui išrenkami taškai $20r$ plote aplink bruožo tašką, kuris yra pasuktas pagal nustatytą bruožo orientaciją. Apibūdinimo plotas padalinamas į 4×4 regionus. Kiekviename regione panaudojamos $2r$ dydžio Haro bangelės. Pagal reakcijas suskaičiuojamas regiono apibūdinimas

$$\left[\sum dx \quad \sum dy \quad \sum |dx| \quad \sum |dy| \right], \quad (43)$$

kur dx ir dy atitinkama reakcija į horizontaliai ir vertikalčiai pritaikytas bangeles.

Tokios sukaupotos reikšmės iš visų 4 regionų surašytos į vieną vektorių sudaro 64 elementų SURF bruožo apibūdinimą. Toks apibūdinimas invariantiškas posūkio, mastelio, šviesumo ir kontrasto pokyčiams.

3. Projektuojama sistema

Automatinis panoramų kūrimas nereikalaujantis papildomo vartotojo įsikišimo yra potenciali internetu teikiama paslauga, analogiška dabartiniams socialiniams tinklams pagrįstais fotografijų ar filmų kūrimu ir dalinimusi, skirta atsitiktiniams vartotojams.

Projektuojamos sistemos paskirtis – interneto naršykle naršoma panoraminių vaizdų valdymo sistema su minimalių vartotojo pastangų reikalaujančiomis panoramų kūrimo priemonėmis.

3.1. Funkciniai reikalavimai

Išskirtos dvi reikalavimų grupės – vaizdų jungimo programos, kuria gali naudotis vaizdų jungimo tyrėjas, ir web aplikacijos, kuri panaudos ribotą vaizdų jungimo programos funkcionalumą.

Vaizdų jungimo programos funkciniai reikalavimai:

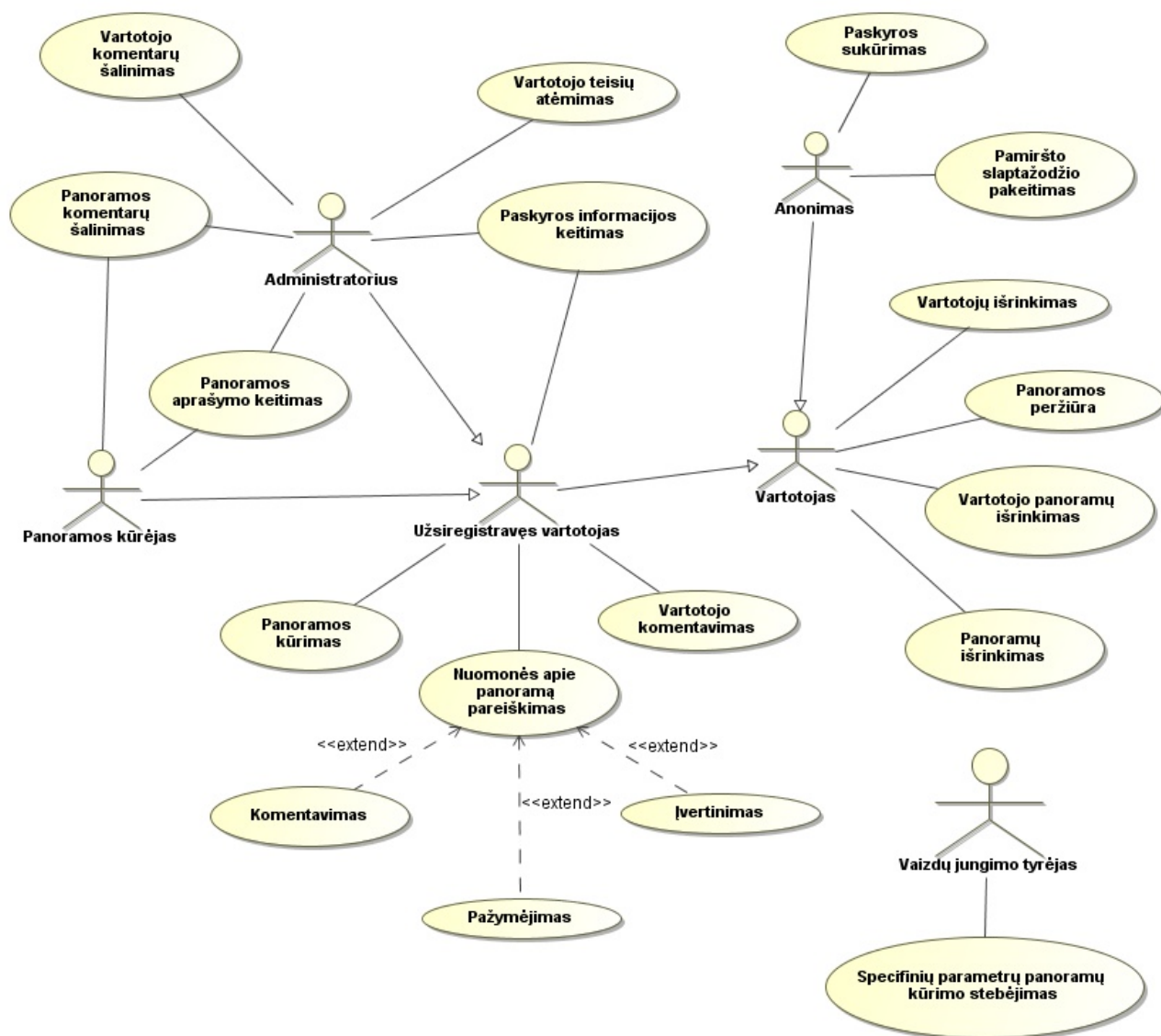
- Programa turi rasti ir sujungti persidengiančius bendro optinio centro vaizdus iš vartotojo pateikiamo neapibrėžtos tvarkos vaizdų rinkinio.
- Programa turi leisti stebėti vaizdų jungimo rezultatus, laisvai parenkant įvairius programos parametrus. Leidžiami keisti parametrai:
 - įėjimo ir išėjimo vaizdų dydžių ribojimai;
 - vaizdų bruožų tipas;
 - bruožų kiekio ribojimas;
 - vaizdų persidengimo nustatymo slenkstinės reikšmės;
 - globalios padėties nustatymo metodas;
 - suliejimo metodas;
 - gijų kiekis;
 - tarpinių ir iliustracinių vaizdų generavimas;
 - pranešimų detalumo lygis.

Web aplikacijos funkciniai reikalavimai:

- Sistema turi teikti joje sukurtų panoramų peržiūros per naršyklę paslaugą.

- Užsiregistravę vartotojai turi turēt galimybę kurti panoramas, užkraunant savo nuotraukų rinkinį.
- Panoramos gali būti išrenkamos pagal naujumą, įvertinimą per laikotarpį ar populiarumą per laikotarpį, pažymėjimų raktažodžiu kiekį ar vykdant tekstinę paiešką jų pavadinimuose ir aprašymuose.
- Galima peržiūrėti vartotojo paskyros informaciją, jo sukurtas panoramas naujumo tvarka ir vartotojo sukurtas žymas.
- Vartotojai gali būti išrenkami pagal jų sukurtų panoramų populiarumą per laikotarpį arba jų įvertinimą per laikotarpį .
- Vartotojai neturintys paskyros gali laisvai užsiregistruoti, pateikę galiojantį elektroninio pašto adresą ir patvirtinę, kad tas adresas tikrai jų. Pamiršus paskyros slaptažodį vartotojas gali inicijuoti jo pakeitimą, keitimo nuorodą atsiunčiant į paskyros savininko paštą.
- Užsiregistravę vartotojai gali komentuoti, žymėti ir vertinti kitų sukurtas panoramas, komentuoti kitus vartotojus.
- Panoramos kūrėjas gali redaguoti jos savybes, šalinti jam paliktus komentarus.
- Administratoriai gali redaguoti visas panoramas ir atimti vartotojo teises.

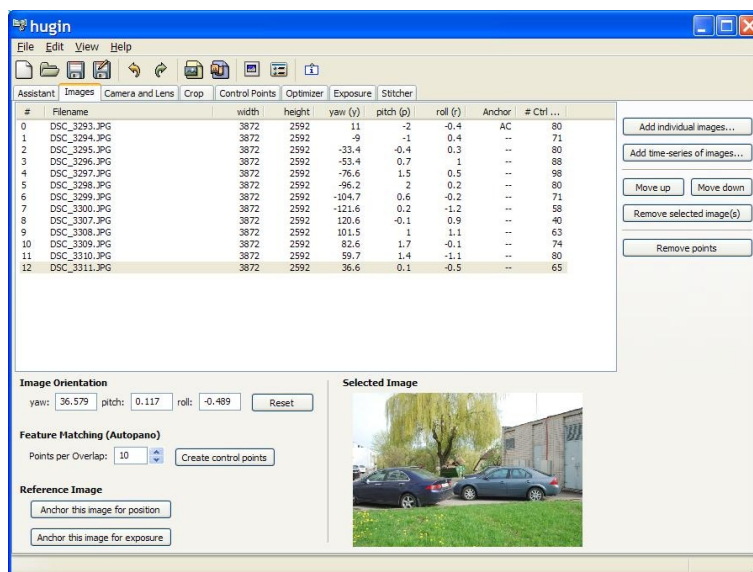
Sistemos panaudojimo atvejus iliustruoja pav. 7.



10 pav. Sistemos panaudojimo atvejai

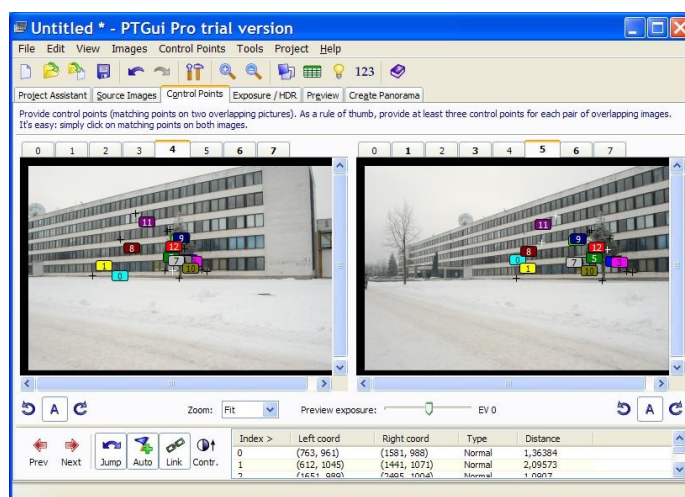
3.2. Panašių produktų tyrimas

Hugin [26] yra nuo platformos nepriklausanti atviro kodo vaizdų jungimo programa. Vartotojas turi galimybę keisti įvairiausias vaizdų jungimo parametrus, dėl to vartotojo sąsaja gan sudėtinga. Vaizdų tarpusavio santykius apibrėžiančius valdymo taškus reikia įvesti rankiniu būdu. Naujausia versija turi įskiepių sistemą leidžiančią integruoti automatinį valdymo taškų išgavimą. Hugin sąsajos pavyzdys pateiktas pav. 11.



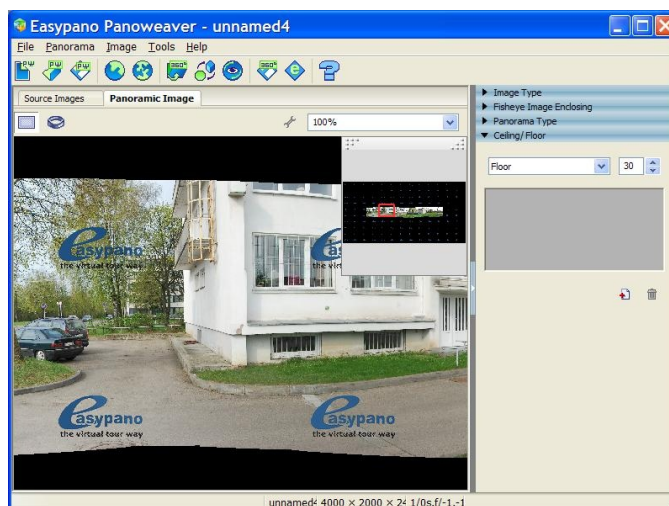
11 pav. Hugin vartotojo sąsaja

PTgui [27] yra brandi komercinė Windows ir Mac OS skirta vaizdų jungimo programa. Apribotų galimybių demonstracinė versija leidžia kurti panoramas su vandens ženklais. Valdymo taškus vaizdų fotoaparato parametrų nustatymui bando rasti automatiškai, bet juos galima nurodyti ir rankiniu būdu su pele. PTgui sąsajos pavyzdys pateiktas pav. 12.



12 pav. PTgui vartotojo sąsaja

Panoweaver [28] yra komercinė Windows ir Mac OS skirta vaizdų jungimo programa. Jos vartotojo sąsaja bando paslėpti jungimui reikalingų parametrų sudėtingumą. Vaizdų fotoaparato parametrus bandoma nustatyti automatiškai, jei tarp jungimui pateiktų vaizdų yra tokių, kurie neturi aiškaus persidengimo su kitais, programa liepia valdymo taškus nurodyti rankiniu būdu. Programa moka eksportuoti sukurtas panoramas jos kūrėjų teikiamoms Flash ir Java grįstoms peržiūros programoms. Panoweaver sąsajos pavyzdys pateiktas pav. 13.



13 pav. Panoweaver vartotojo sąsaja

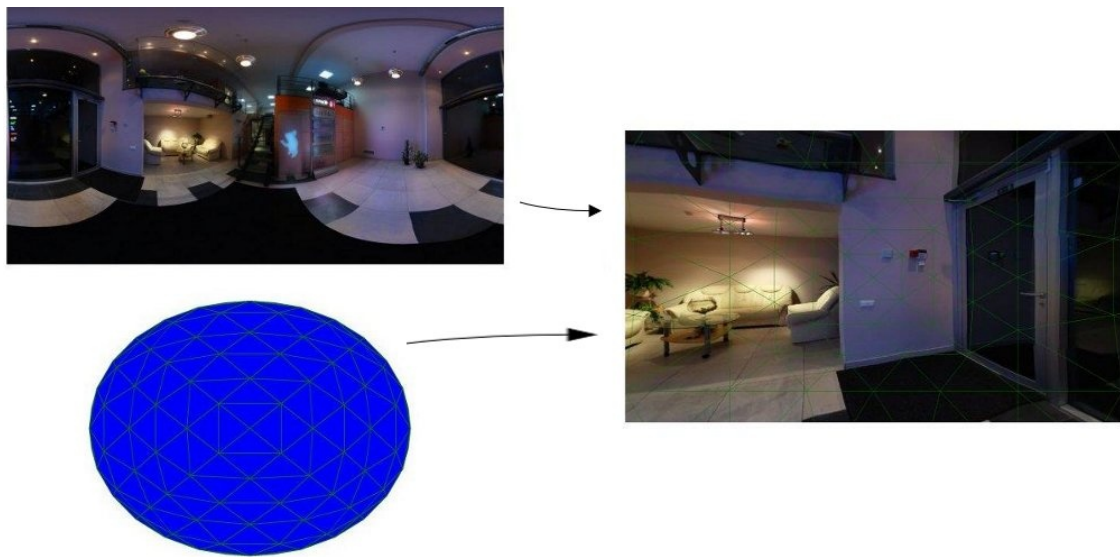
Google Street View [29] yra nemokama Google maps pagrįsta sujungtų panoraminių vaizdų peržiūros sistema. Sferinėse panoramose fiksuojamas didžiausių pasaulio miestų pagrindinių gatvių vaizdai. Panoramos fotografuotos iš važiuojančių mašinų su specializuotais panoraminiais fotoaparatais. Vizualizacijai naudojamas Flash. Panoramomis galima vaikščioti fotografuotų kelių kryptimis.

panoramas.lt Siūlo komercines panoraminių vaizdų kūrimo paslaugas, panaudojant jų turimą aparatūrinę ir programinę įrangą. Tinklapyje demonstruojama jau sukurtos panoramos panaudojant Quicktime VR [30], per turinį galima naršyti žemėlapiu pagalba.

Šiuo metu nerasta sistema leidžianti vartotojams automatiškai kurti panoramas, nereikalaujant papildomų mokesčių, nededant vandens ženklų ir leidžiant sukurtas panoramas talpinti internete.

3.3. Panoraminių vaizdų vizualizacija

Panoramų vaizdus nebūtina versti į cilindrinės ar sferinės koordinatės ir paišyti specializuotomis priemonėmis. Galima pasinaudoti standartinėmis 3D grafikos API, sferų ar cilindrų geometrines figūras generuojant iš daugiakampių, panoraminį vaizdą panaudojant kaip tekstūrą (žr. pav. 14).



14 pav. Sferinės panoramos vizualizavimas

3.3.1. Vizualizacijos priemonės naršyklėse

Quicktime VR. Viena iš pirmųjų ir plačiausiai naudojama panoraminių vaizdų vizualizavimo sistema. Apple Quicktime paketo dalis. Ši naršyklės įskiepi turi įdiegta 61% interneto vartotojų. Įskiepis veikia tik Apple kompiuteriuose ir Windows operacinėje sistemoje. Įmanoma nurodyti aktyvias panoramos dalis, reaguojančias į vartotojo pelės paspaudimą, navigacijai per panoramų rinkinį. Sistema uždara, jos neįmanoma išplėsti papildomomis funkcijomis.

Flash Turi įdiegta 99% vartotojų. Flash įskiepis neturi išskirtinių priemonių panoramoms vizualizuoti, tačiau nesudėtinga poligoninių 3D objektų vaizdavimą realizuoti panaudojant egzistuojančią vektorinės 2D grafikos infrastruktūrą. Taip pat galima išnaudoti įvykių sistemą interaktyvių elementų integravimui.

Java. Panoraminių vaizdų įrankį galima realizuoti Java apletu. Egzistuoja bendros paskirties Java skirti 3D grafikos paketai. Įdiegta įskiepi turi 77% interneto vartotojų. Java naršyklės įskiepiams būdingas ilgas paleidimo laikas.

Javascript. Veikia visose naršyklėse, tačiau visose interpretuojama šiek tiek skirtingai. Įmanoma panoraminius vaizdus realizuoti paišant po tašką, tačiau paišymo API per lėtas praktiniam didesnių panoramų vaizdavimui.

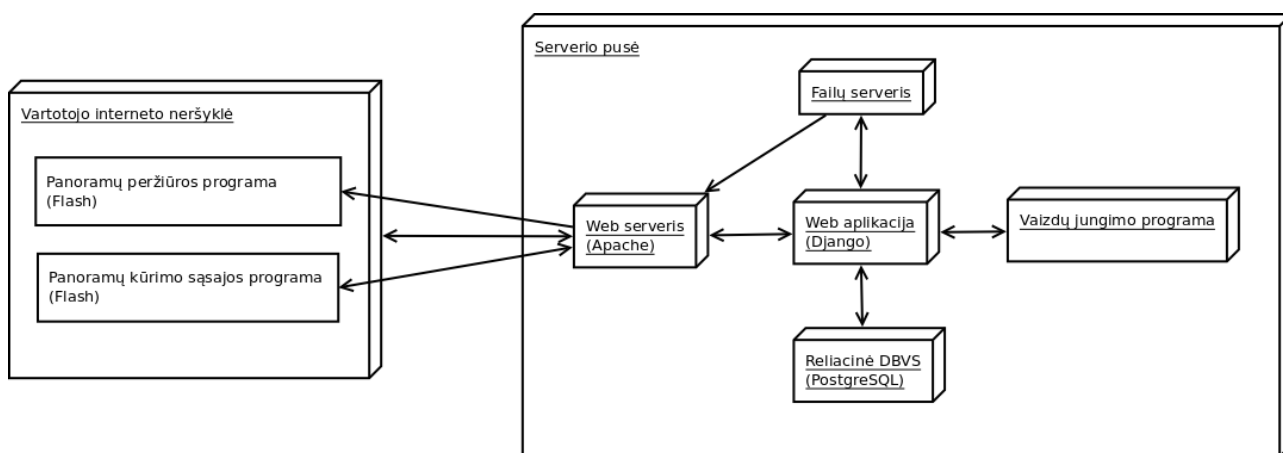
VRML įskiepiai. VRML yra standartizuotas bylų formatas 3D interaktyviai vektoriniai grafikai atvaizduoti. Prieš dešimtmetį buvo sukurta daugybė VRML skirtos programinės įrangos. Dėl privačių formatų ir riboto to laikmečio interneto ryšio pralaidumo, dauguma iniciatyvų sužlugo.

Daug VRML įskiepių seniai nebepalaikomi. Labai mažas kiekis vartotojų turi įdiegtą VRML įskiepi.

Vartotojų kiekių duomenys paimti iš [31]. Projekte vizualizacijai pasirinkta naudoti Flash aplinką.

3.4. Architektūra

Sistemą sudaro: vaizdų jungimo programa, web aplikacija ir vartotojo sąsajos programos, skirtos panoramų kūrimui ir peržiūrai. Sistemos komponentus iliustruoja pav. 15.



15 pav. Sistemos komponentai

Pasirinktos sistemos realizavimo priemonės:

- Vaizdų jungimo programa realizuota C++ kalba, dėl uždavinio sudėtingumo kylančio aukšto našumo poreikio ir suderinamumo su dauguma skaičiavimams skirtų programinių bibliotekų.
- Web aplikacija realizuota Python kalba ir Django karkasu, leidžiančiu greitai kurti duomenų baze grįstas web aplikacijas pasinaudojant ORM sąsaja.
- Duomenų saugojimui pasirinkta PostgreSQL reliacinė duomenų bazė. Ši sistema išsivystė iš komercinio produkto Ingres ir yra brandžiausia atviro kodo DBVS.
- Vartotojo sąsajos realizacijai pasirinkta haXe kalba, leidžianti rašyti kompaktišką kodą, kompiliuojamą į Action Script Virtual Machine 3 bait kodą, kuris bus vykdomas interneto naršyklės Flash įskiepyje.

3.4.1. Vaizdų jungimo komponento architektūra

Programa realizuota plačiai naudojant C++ standartinės bibliotekos (STL) algoritmus. Dauguma komponentų suprojektuota STL skatinamu funkcijiniu stiliumi, vengiant objektuose laikyti būsenos kintamuosius. Kintančios būsenos neegsirtavimas leidžia kodą našiai atlikti lygiagrečiai, negaištant procesoriaus laiko sinchronizacijos mechanizmuose, reikalinguose bendrų kintamųjų apsaugojimui nuo rašymo konfliktų. Būsena saugoma išskirtiniais atvejais, kaip optimizavimo priemonė. Programos klasių diagrama pateikta pav. 16.

- Boost – bibliotekų rinkinys siekiantis įgyvendinti trūkstamą C++ standartinės bibliotekos funkcionalumą. Projekte panaudota priėjimui prie operacinės sistemos teikiamų lygiagretaus programavimo priemonių ir kokybiškam atsitiktinių skaičių generavimui.
- GSL(GNU Scientific Library) – matematinių skaičiavimų biblioteka. Projekte panaudota matricinėms operacijoms ir tiesinių lygčių sprendimui.
- FLANN(Fast Library for Approximate Nearest Neighbors) – greitos paieškos didelės dimensijos erdvėse priemonės.
- OpenCV(Open Computer Vision) – vaizdų mapinuliavimo priemonės.
- VW(NASA Vision Workbench) – vaizdų mapinuliavimo priemonės. Projekte panaudota vaizdų geometrinėms transformacijoms.
- Rob Hess SIFT [32] – SIFT bruožų išgavimo biblioteka.
- Googletest – C++ kodo testavimo karkasas.

3.4.1.2. Lygiagretūs komponentai

Programa atlieka didelės apimties skaičiavimus, o daugiausiai laiko reikalaujantys skaičiavimai atliekami su daug duomenų, kurie dažnai yra nepriklausomi tarpusavyje. Tokius veiksmus galima efektyviai atlikti lygiagrečiose gijose, siekiant padidinti našumą, programą vykdant kompiuteriuose su šiuolaikiniais kelis branduolius turinčiais procesoriais.

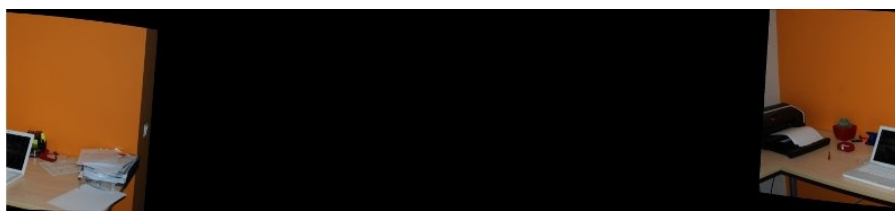
Lygiagretumas įvestas vektorinėms procedūroms, kurių vykdymui sugaištama daugiausiai laiko. Šių procedūrų argumentų duomenų kolekcijos išskaidomos kiekvienai gijai. Gijos atlieka procedūrą, rezultatus rašydamos į atitinkančias rezultato vektoriaus pozicijas. Pagrindinė gija tęsia darbą, kai sulaukia visų skaidyme sukurtų gijų darbo pabaigos.

Skirtingo tipo lygiagretūs veiksmai:

- Vektorinės funkcijos su tarpusavyje nepriklausomais duomenimis. Tokios funkcijos neturi bendros kintančios būsenos, todėl nereikalingi papildomi sinchronizavimo mechanizmai. Programoje tokį funkcijų tipą atitinka nepriklausomos atskirų vaizdų atžvilgiu funkcijos: vaizdo bruožų išgavimas, vaizdo dažnių juostų išgavimas ir kaimynų paieška vaizdo bruožams.
- Pradinių daugiajuosčio suliejimo svorinių funkcijų generavimas. Ši procedūra atliekama lygiagrečiai atskirų vaizdų atžvilgiu. Procedūra pagal slenkstinę funkciją modifikuoja

bendrą būseną, gali ištikti rašymo konfliktai. Pradinės ir pakeistos reikšmės išlaiko slenkstinės funkcijos tikrinamos sąlygos savybes ir rezultatai bet koku atveju gaunami teisingi. Šiai procedūrai taip pat nereikalingi sinchronizavimo mechanizmai.

- Vaizdo transformavimas į bendras sferines koordinates. Ši funkcija nepriklausoma duomenų atžvilgiu, tačiau rezultatų kiekis, o tuo pačiu ir rezultato rašymo pozicija, gali kisti, todėl rašymą į rezultatų vektorių reikia apsaugoti kritine sekcija. Rezultatų kiekis gali kisti, nes vaizdui kertant sferinės koordinatės θ pradžią, jis išsidėsto abiejuose išskleisto išėjimo panoraminio vaizdo kraštuose ir dėl atminties taupymo turi būti suskaidytas į du atskirus vaizdus (žr. pav. 17).
- Fotoaparato parametrų nustatyme naudojamų matricių sudarymas. Persidengiančių vaizdų poros bruožai įtakoja tik mažas matricių dalis, kurių sudaryme dalyvauja tų vaizdų parametrų išvestinės, tad šią procedūrą galima atlikti lygiagrečiai persidengiančių vaizdų porų atžvilgiu. Vienas vaizdas gali persidengti su keliais kitais, todėl jo įtakojamus matricių plotus gali bandyti modifikuoti kelios gijos ir rezultatų matricių modifikavimo veiksmus reikia apsaugoti kritinėmis sekcijomis.
- Vaizdų Suliejimo funkcijos. Šias funkcijas nepatogu skaidyti pagal argumentą, nes dideli taškų plotai gali būti perdengti keliais vaizdais ir tikėtini didžiuliai rašymo konfliktai. Šios funkcijos atliekamos lygiagrečiai rezultato atžvilgiu, gijoms paskirstant pildomus išėjimo vaizdo plotus.



17 pav. Vaizdas bendrose sferinėse koordinatėse, kai jis kerta koordinatės θ pradžią

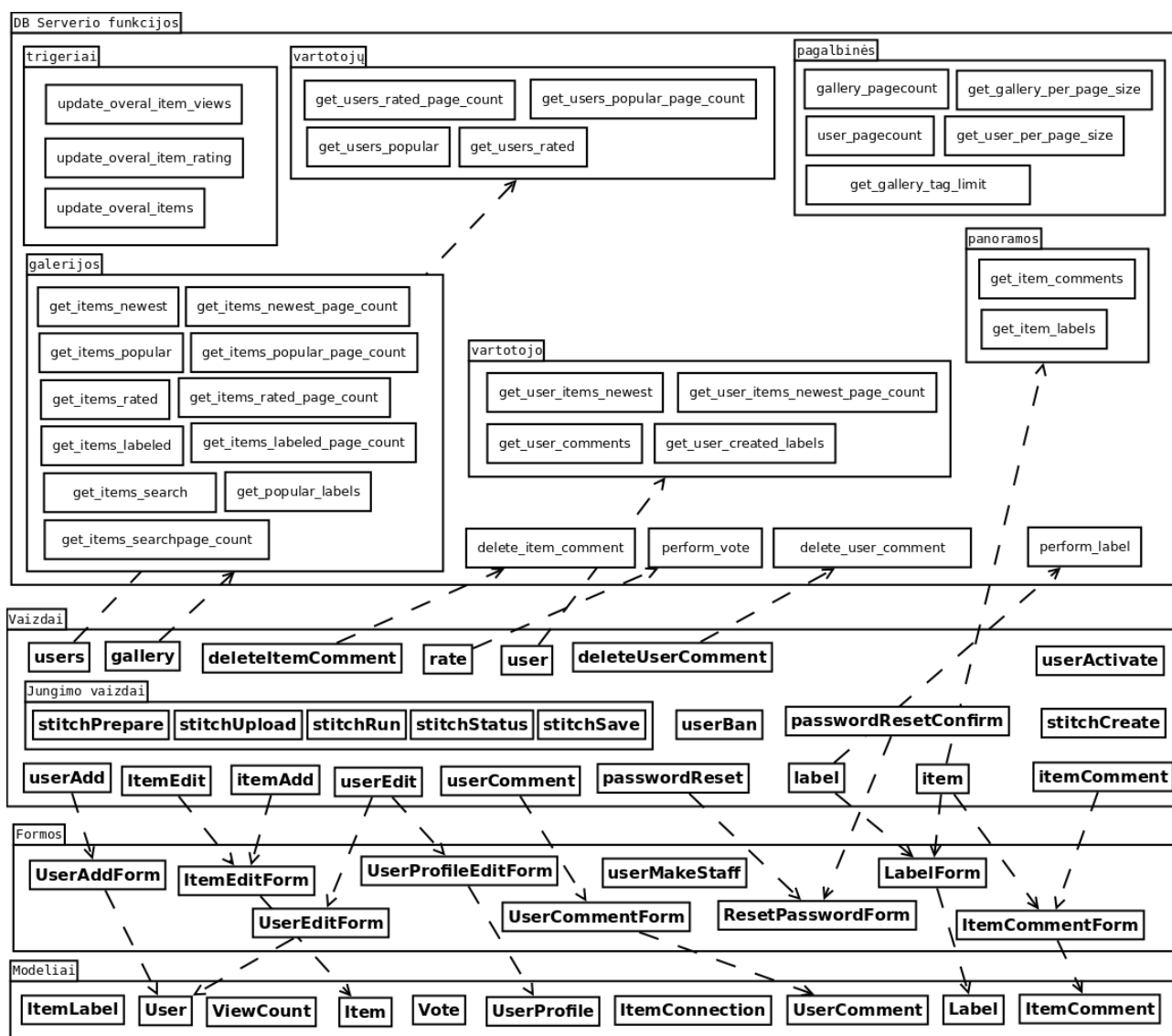
Lėčiausias programos žingsnis – nuoseklus fotoaparato parametrų optimizavimas. Šis procesas yra iteratyvus ir nėra aiškus jo išlygiagretinimo būdo.

3.4.2. Web aplikacijos architektūra

Su sistema vartotojai sąveikauja per interneto naršyklę naudojant http protokolą. Kreipiniai apdorojami pagal kreipimosi adresą kviečiant atitinkamą vaizdo funkciją, kuri sugeneruoja atsakymo html kodą. Aplikacijos struktūra pateikta pav. 18.

Panaudojami papildomi Django web karkaso moduliai:

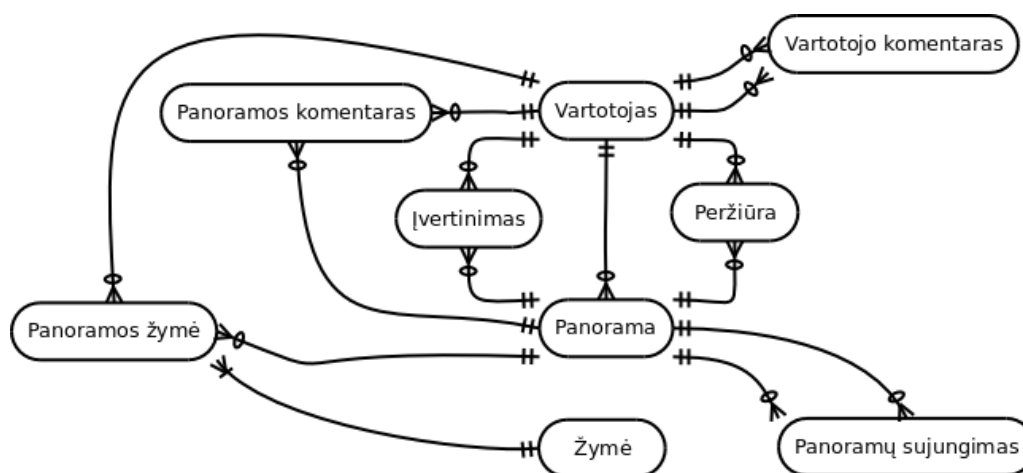
- Auth – vartotojo autentifikacijos ir teisių ribojimo modulis;
- i18n – vertimus įgalinantis modulis;
- CSRF – apsaugojimo nuo padirbtų užklausų modulis;
- Notifications – vartotojo pranešimų saugojimo modulis.



18 pav. Web aplikacijos struktūra

3.4.3. Duomenų bazės architektūra

Visi kintami aplikacijos duomenys, išskyrus vaizdus, saugomi duomenų bazėje. Nustatytos duomenų esybės pavaizduotos 19 pav.



19 pav. Esibių diagrama

Visos duomenų lentelės sudarytos trečioje normalinėje formoje. Django paketo standartiniai moduliai naudoja savo vartotojo lentelę kurioje egzistuoja dauguma norimų panaudoti vartotojo laukų. Papildomiems laukams talpinti sukurta nauja vartotojo profilio lentelė, su vartotojo lentele susieta išoriniu raktu.

Sistemos našumo pagerinimui, lentelės papildytos pertekliniais nenormalizuotais laukais (Panorama – bendru įvertinimu ir peržiūrų kiekiu, Vartotojas – jo sukurtų panoramų kiekiu), kurių reikšmės perskaičiuojamos trigeriais iškviečiamomis procedūromis. Pirminiams, išoriniams raktams ir datų laukams sukurti indeksai.

Sudėtingos ir daug duomenų išrenkančios užklausos realizuotos duomenų bazės procedūromis parašytomis PostgreSQL PL/pgSQL kalba. Paprasčiausias užklausas, kurios neturi potencialo būti pagreitintos išsaugojus procedūros vykdymo planą, leidžiama generuoti Django ORM.

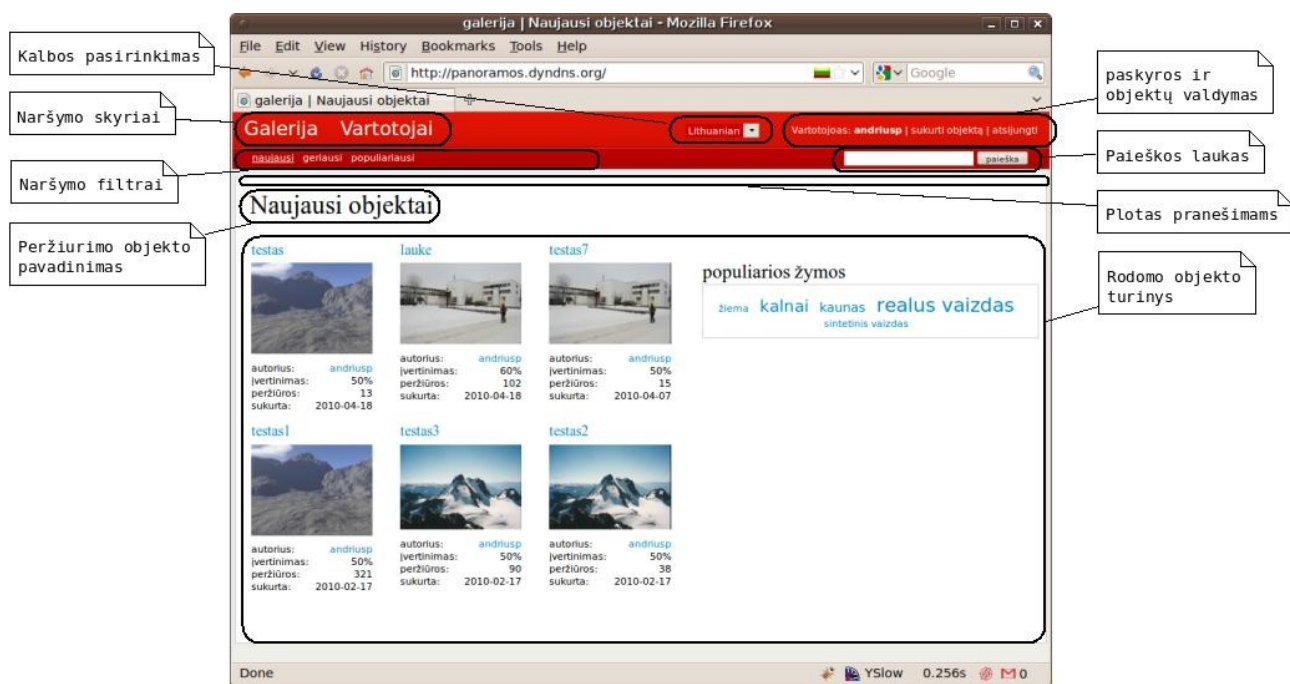
Suprojektuotą duomenų bazę sudaro:

- 10 lentelių;
- 32 procedūrų;
- 3 poschemės.

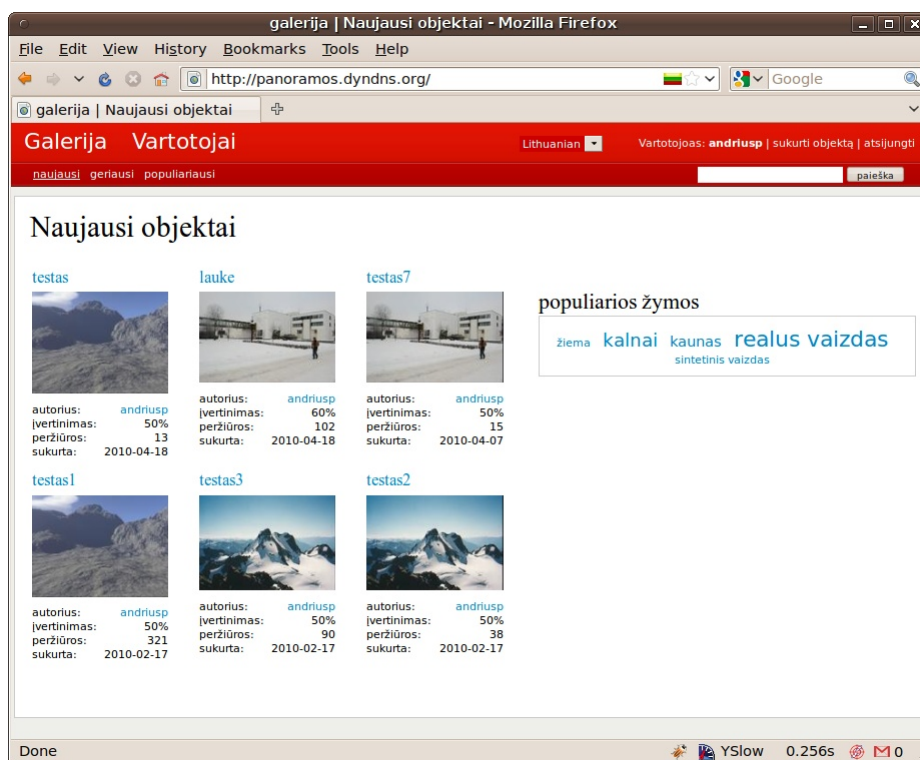
Duomenų bazės schemą iliustruoja 20 pav.

3.5. Vartotojo sąsaja

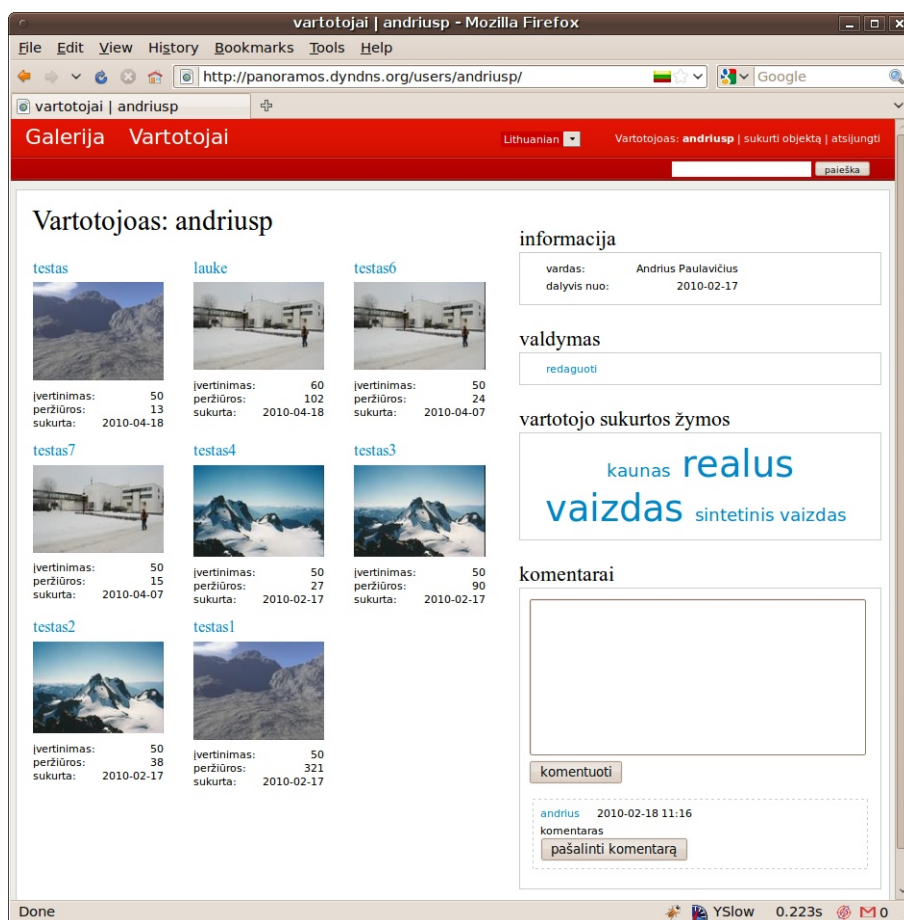
Bendras vartotojo sąsajos išdėstymas pavaizduotas pav. 21. Specifinių vaizdų fragmentai – pav. 22–26.



21 pav. Bendras vartotojo sąsajos išdėstymas

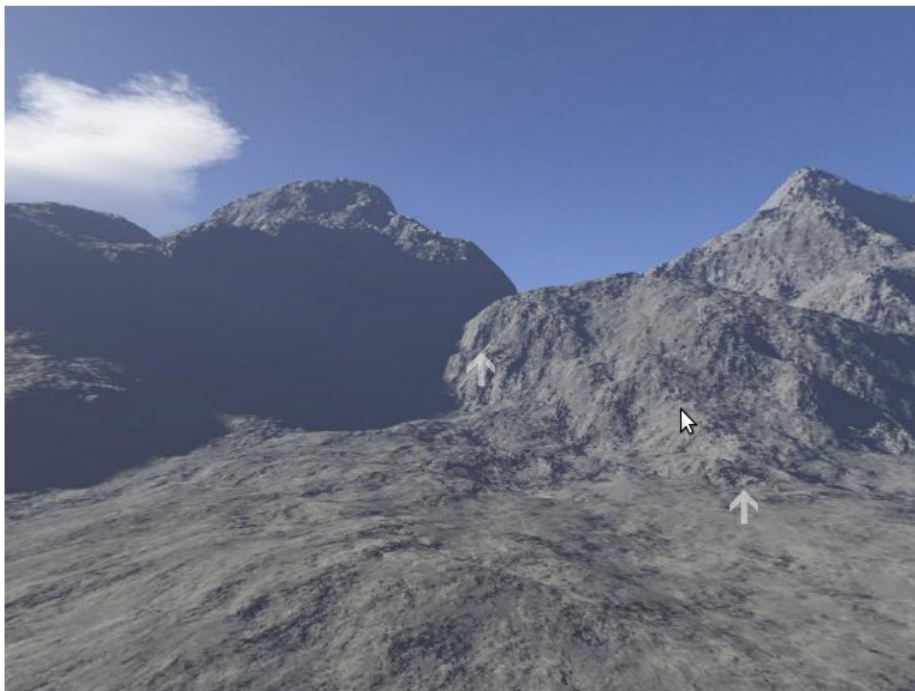


22 pav. Panoramų išrinkimo vartotojo sąsaja



23 pav. Vartotojo profilio peržiūros vartotojo sąsaja

testas1
autorius: andriusp



komentari

komentuoti

paulius 2010-05-10 00:58
komentaras komentaras komentaras

statistika

sukurta: 2010-02-17
įvertinimas: 50%
peržiūros: 320

jūsų įvertinimas:

50 %

žymos

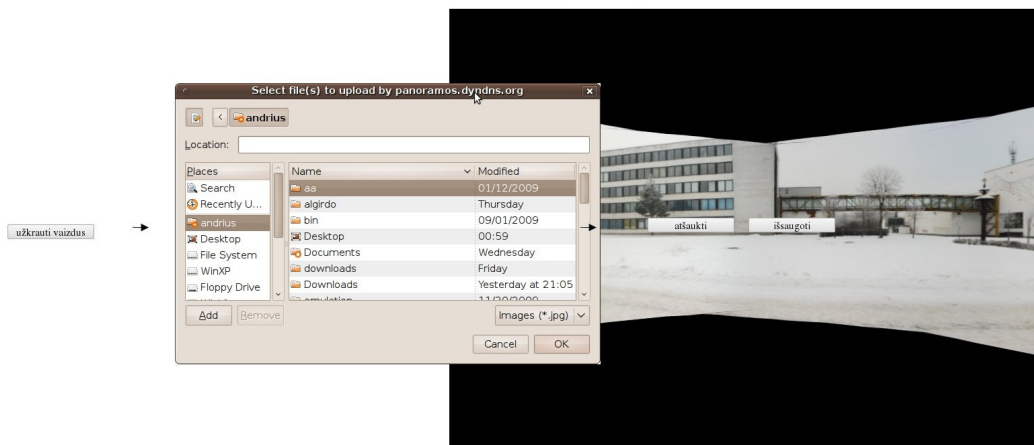
kalnai sintetinis
vaizdas

pažymėti

24 pav. Panoramos peržiūros vartotojo sąsaja

testas6 sukurti panoramą

testas6 sukurti panoramą



25 pav. Panoramos kūrimo vartotojo sąsaja

redaguoti paskyrą andriusp

Vardas:	<input type="text" value="Andrius"/>
Pavardė:	<input type="text" value="Paulavičius"/>
E-pašto adresas:	<input type="text" value="apaulavicius@gmail.com"/>
Apie:	<input type="text"/>
Trumas apie:	<input type="text"/>

26 pav. Vartotojo sąsajos formos pavyzdys

3.6. Testavimas

Sistemos veikimo teisingumui patikrinti atliktas testavimas:

- Unit testai – automatizuotas žemo lygio vaizdų jungimo funkcionalumo testavimas;
- Sisteminiai testai – vaizdų jungimo su kontroliniais vaizdų rinkiniais patikrinimas;
- Vartotojo sąsajos testavimas – rankinis web aplikacijos elgesio atitikimo reikalavimams tikrinimas.

Unit testai vykdomi su Google Googletest karkasu. Sudarytas 21 testinių scenarijų rinkinys, jame tikrinamos 79 sąlygos. Tikrinami funkcijų rezultatai su ribinėmis jų parametrų reikšmėmis ir žinomos rezultatų savybės.

Funkcijos, generuojančios nurodomą kiekį atsitiktinių nesikartojančių sveikų skaičių nuo 0 iki nurodytos reikšmės, testų pvz.:

```
TEST(RandomGenerator, extremesCheck) {
    vector<int> vec = randomUniqueInts(3, 3);
    sort(vec.begin(), vec.end());
    int vecExpect[] = {0, 1, 2};
    EXPECT_TRUE(equal(vec.begin(), vec.end(), vecExpect));

    vector<int> failVec1 = randomUniqueInts(3, -1);
    EXPECT_TRUE(failVec1.empty());

    vector<int> failVec2 = randomUniqueInts(3, 4);
```

```

    int failExpect2[] = {0, 0, 0, 0};
    EXPECT_TRUE(equal(failVec2.begin(), failVec2.end(), failExpect2));
}

TEST(RandomGenerator, boundsCheck) {
    const int tryCount = 30;
    const int iterationSizeMul = 10;
    const int sizeToTake = 10;

    bool hadLower = false, hadHigher = false, hadSame = false;
    for (int i = 0; i < tryCount; i++)
    {
        int maxVal = (i+1) * iterationSizeMul;
        vector<int> vec = randomUniqueInts(maxVal, sizeToTake);
        sort(vec.begin(), vec.end());

        if (*vec.begin() < 0)
            hadLower = true;

        if (*vec.end() >= maxVal)
            hadHigher = true;

        if (unique(vec.begin(), vec.end()) - vec.begin() != (int)vec.size())
            hadSame = true;
    }
    EXPECT_TRUE(!hadLower);
    EXPECT_TRUE(!hadHigher);
    EXPECT_TRUE(!hadSame);
}

```

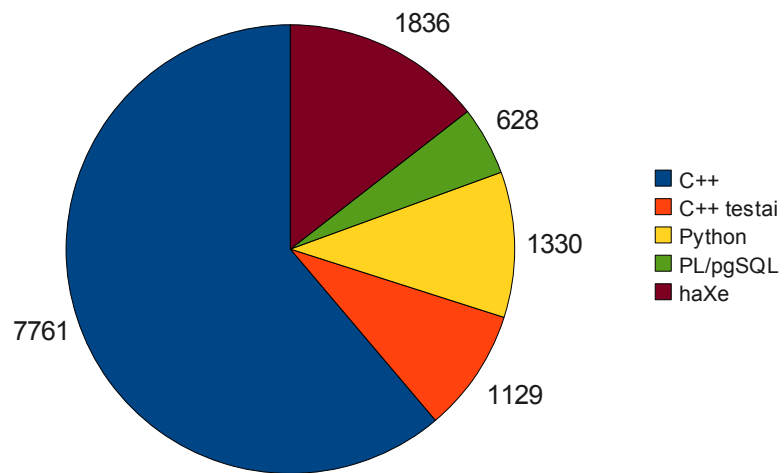
Programoje naudojami euristiniai metodai ir atsitiktinių skaičių generatoriai, tad sunku ištestuoti aukšto lygio funkcionalumą. Apie aukšto lygio funkcionalumo teisingumą sprendžiama iš sisteminių testų su keliais kontrolinių vaizdų rinkiniais.

3.7. Rezultatai

Realizuota automatinio vaizdų jungimo programa pagal šiame tyrime aprašytus žingsnius. Nerealizuota tekstūrų sintezė, nes bendru atveju ja gaunami nepatenkinami rezultatai.

Pilnai realizuota aprašyta web aplikacija. Vaizdų jungimo programa iš web aplikacijos paleidžiama su geriausiais bendro atvejo nustatymais, vaizdų fotoaparato parametrų nustatymui naudojant SIFT bruožus, vaizdus suliejant daugiajuosčiu 3 dažnių juostų suliejimu.

Projektui parašyto programų kodo kiekis iliustruoja pav. 27.



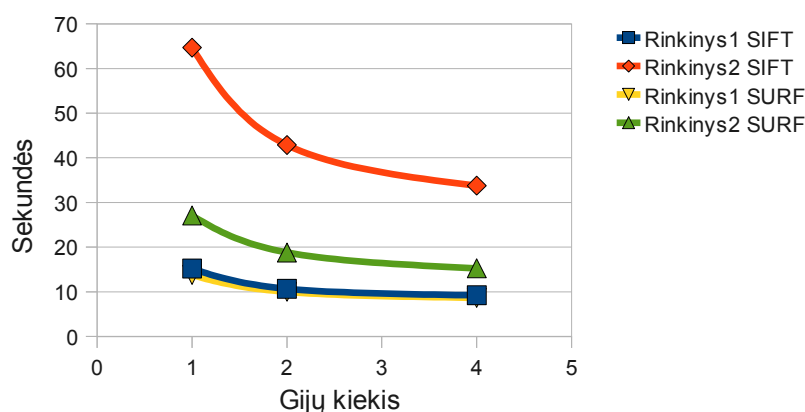
27 pav. Projektui parašyto kodo eilučių kiekiai

4. Eksperimentai

Visuose eksperimentuose kiekvienam vaizdui vertinama iki 500 geriausių bruožų, vienam vaizdui vertinama iki 4 geriausių persidengimų, naudojamos 3–jų dažnių juostų daugiajuostis suliejimas, išėjime gaunamos 2000 pikselių pločio panoramos.

4.1. Veikimo laiko įvertinimas

Veikimo laikas įvertintas atliekant vaizdų jungimą su tais pačiais vaizdų rinkiniais, kintant panaudojamų vaizdų bruožų tipui ir panaudojamų gijų skaičiui. Bandymas atliktas su keturių branduolių 2.6GHz Intel Core2 Q9400 procesorių turinčiu kompiuteriu. Pirmas vaizdų rinkinys sudarytas iš 7 nuotraukų, antras – iš 27. Eksperimento rezultatai pateikti pav. 28.



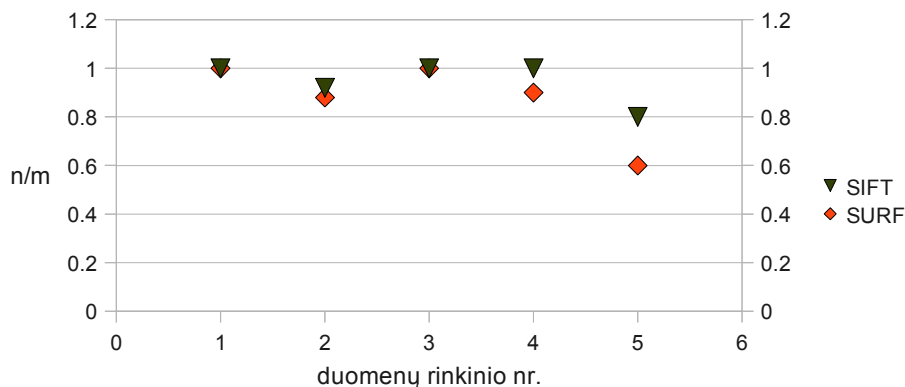
28 pav. Veikimo laiko įvertinimas

SURF bruožai išgaunami žymiai greičiau nei SIFT. Jų dvigubai mažesnės dimensijos apibūdinimo vektoriai smarkiai paspartina kaimyninių bruožų paiešką. Didinat gijų kiekį, žymiai ryškesnis paspartėjimas matomas su dideliais vaizdų rinkiniais. Taip yra, nes dauguma lygiagrečių veiksmų gijoms išskaidomi aukštame lygyje. Neturint pakankamai duomenų, ne visada visi procesoriai turi darbo. Didinant gijų kiekį gaunamas vis mažesnis pagreitis, dėl kompiuterio atminties ir spartinančiosios atminties apribojimų.

4.2. Sujungimo kokybė

Sunku apibrėžti panoramų jungimo kokybės kriterijus. Automatiniame jungime naudojamas atsitiktinių skaičių generatorius ir apytiksliai euristiniai metodai, kurie net tos pačios programos skirtinguose bandymuose gali duoti neviseškai vienodus rezultatus. Įvertinimui pasirinktas kriterijus

$\frac{n}{m}$, kur n – žmogaus suvokiamas apytiksliai teisingomis reliatyviomis padėtimis programos sujungtų vaizdų kiekis ir m – žmogus suvokiamas visų persidengiančių vaizdų kiekis. Eksperimento rezultatai pateikti pav. 29.



29 pav. Sujungimo kokybė

Panaudojant SURF bruožus surandama tiek pat arba mažiau persidengimų nei naudojant SIFT. Ne visi persidengimai randami vaizdų rinkiniuose, kuriuose tarp vaizdų nėra aiškių išskirtinių bendrų bruožų, arba bruožai smarkiai skiriasi dėl atspindžių ar objektų judesio.

4.3. Palyginimas su alternatyvomis

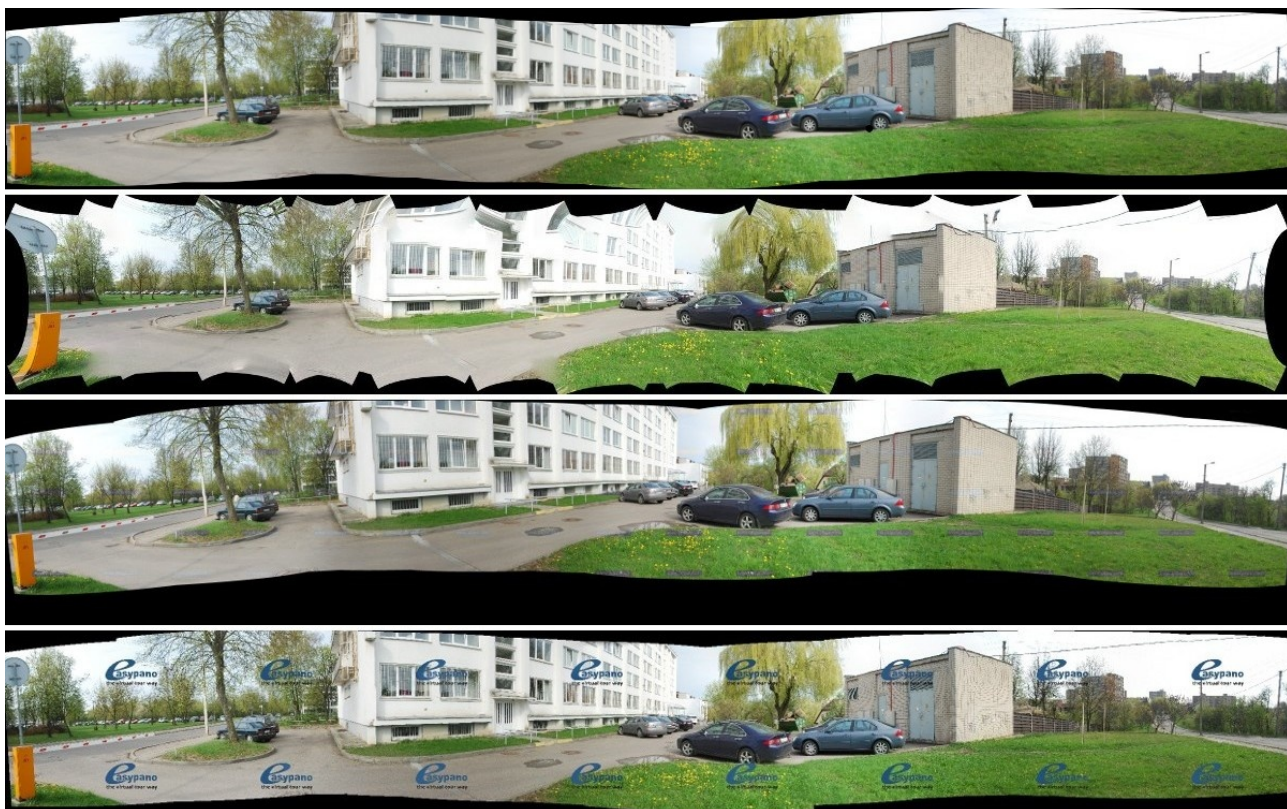
Atliktas palyginimas tarp realizuotos programos ir jos alternatyvų, naudojant tuos pačius vaizdų rinkinius. Bandyje kaip įmanoma daugiau parametrų buvo parinkta automatiškai. Hugin kontrolinių taškų nustatymui panaudotas SIFT įskiepis. Bandyką su tuo pačiu rinkiniu iliustruoja pav. 30.

Pastebėjimai:

- Naudojant nedidelius vaizdų rinkinius, visos programos duoda šiek tiek skirtingus, bet vizualiai teisingus rezultatus.
- Alternatyvios programos, kai pateiktuose jungimui rinkiniuose yra vaizdų, kuriems nerandama persidengimo su kitais vaizdais, atsisako jungti panoramą arba reikalauja bruožų taškus parinkti rankiniu būdu. Realizuota programa vaizdus, kuriems nerandama persidengimų, ignoruoja.
- Hugin programoje galima rankiniu būdu nustatyti labai daug detalių parametrų, kurie teoriškai leidžia kompensuoti įvairius vaizdų iškraipymus ir pasiekti aukštą kokybę. Toks

parametrų parinkimas yra sudėtingas ir reikalauja daug vartotojo pastangų. Veikimo laikas ženkiai didesnis nei kitų programų.

- Realizuota programa mažiau atspari neteisingiems persidengimų nustatymui nei kitos alternatyvos. Keli neteisingi nustatyti persidengimai gali sugadinti visų optimizuojamų vaizdų fotoaparato parametrus.



30 pav. Panoramų, sukurtų iš to pačio vaizdų rinkinio skirtingomis programomis palyginimas. Projekte realizuota programa, Hugin, PTgui, Panoweaver

5. Išvados

1. Išanalizavus literatūrą, sudarytas apibendrintas automatinio vaizdų jungimo algoritmas. Sėkmingai realizuota ir išbandyta algoritmo realizacija su dviejų rūšių vaizdų bruožais.
2. Šiuolaikiniai vaizdų bruožai leidžia rasti persidengiančius vaizdus ir jų santykinę padėtį, nereikalaujant vartotojo įsikišimo.
3. Neturint pilnų sferinės ar cilindrinės panoramos aplinkos nuotraukų, susidarančias skylės galima užpildyti tekstūrų sintezės metodu. Praktiškai, vizualiai patenkinami rezultatai gaunami tik tuo atveju, kai užpildomos panoramos skylės yra labai mažos.
4. Dauguma pateikto algoritmo žingsnių operuoja dideliais kiekiais tarpusavyje nepriklausomų duomenų. Šiuos žingsnius patogiu atlikti lygiagrečiai, keliant našumą, jungimą vykdančią su moderniais, daug procesoriaus branduolių turinčiais kompiuteriais.
5. Realizuotoje programoje, naudojant keturias gijas keturių procesoriaus branduolių sistemoje, gautas pagreitėjimas artimas 1.7 kartų vienos gijos atžvilgiu. Didinant gijų kiekį gaunamas vis mažesnis pagreitėjimas, dėl kompiuterio atminties ir spartinančiosios atminties apribojimų. Realizuota programa neefektyviai išnaudotų procesorius su daugiau nei keturiais branduoliais.
6. SURF bruožai išgaunami žymiai greičiau nei SIFT. Jų dvigubai mažesnės dimensijos apibūdinimo vektoriai smarkiai paspartina kaimyninių bruožų paiešką, tačiau praktiniuose bandymuose SURF bruožai teisingai sutapatunami rečiau nei SIFT.
7. Automatizuotas vaizdų jungimas neįmanomas, kai tarp šių vaizdų nėra aiškių bendrų bruožų, arba bruožai smarkiai skiriasi dėl atspindžių ar judesio.
8. Realizuota vaizdų jungimo programa pasiekia rezultatus, suliginamus su jos komerciniais analogais, tačiau programa linkusi patvirtinti neteisingus persidengimus. Tolimesniam vystyme reikėtų modifikuoti persidengimų patvirtinimo funkciją, jos slenksčius nustatant iš didelių, žinomų parametrų vaizdų rinkinių.

6. Literatūra

- [1] Chia-Yen Chen ir Reinhard Klette, „Image stitching - comparisons and new techniques“, In CAIP '99: Proceedings of the 8th International Conference on Computer Analysis of Images and Patterns, London, UK, 1999.
- [2] Richard Szeliski, „Image alignment and stitching: A tutorial“, Technical report, Microsoft Research, 2004.
- [3] R. Szeliski ir H. Shum. „Creating full view panoramic image mosaics and environment maps“. Computer Graphics (SIGGRAPH'97), 1997.
- [4] H.Y. Shum ir R. Szeliski, „Panoramic image mosaics“, Microsoft Research, 1997.
- [5] H. Sawhney ir R. Kumar. „True multi-image alignment and its application to mosaicing and lens distortion correction“. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999.
- [6] H. Shum ir R. Szeliski. „Construction of panoramic mosaics with global and local alignment“. International Journal of Computer Vision, 2000.
- [7] M. Irani ir P. Anandan. „About direct methods“. Vision Algorithms: Theory and Practice, number 1883 in LNCS, Springer-Verlag, Corfu, Greece, 1999.
- [8] P. McLauchlan, A. Jaenicke. „Image mosaicing using sequential bundle adjustment“. Image and Vision Computing, 2002.
- [9] D. Capel ir A. Zisserman. „Automated mosaicing with super-resolution zoom“. In Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR98), June 1998.
- [10] M. Brown, „Multi-Image Matching Using Invariant Features“, University of British Columbia, 2005.
- [11] P. McLauchlan, A. Jaenicke. „Image mosaicing using sequential bundle adjustment“. Image and Vision Computing, 2002.
- [12] Marius Muja ir David G. Lowe, „Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration“, International Conference on Computer Vision Theory and Applications, 2009.

- [13] M. Fischler ir R. Bolles. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 1981.
- [14] R. Hartley ir A. Zisserman. „Multiple View Geometry in Computer Vision“. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [15] W. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. „Bundle adjustment: A modern synthesis“. In *Vision Algorithms: Theory and Practice*, number 1883 in LNCS. Springer-Verlag, Corfu, Greece, September 1999.
- [16] Kaj Madsen Hans Bruun ir Ole Tingleff Imm „Methods for non-linear least squares problems“, 2004.
- [17] P. Burt ir E. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 1983.
- [18] Alexei A. Efros ir Thomas K. Leung. „Texture Synthesis by Non-Parametric Sampling“. *Proceedings of the International Conference on Computer Vision-Volume 2*. 1999.
- [19] R. Zabih ir J. Woodfill, “Non-Parametric Local Transforms for Computing Visual Correspondance,” *Proc. Third European Conf. Computer Vision*, 1994.
- [20] D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *Int’l J. Computer Vision*, vol. 2, no. 60, 2004.
- [21] L. Florack, B. ter Haar Romeny, J. Koenderink, M. Viergever, “General Intensity Transformations and Second Order Invariants,” *Proc. Seventh Scandinavian Conf. Image Analysis*, 1991.
- [22] L. Van Gool, T. Moons, and D. Ungureanu, “Affine/Photometric Invariants for Planar Intensity Patterns,” *Proc. Fourth European Conf. Computer Vision*, 1996.
- [23] K. Mikolajczyk ir C. Schmid, „A performance evaluation of descriptors“, in *Proceedings of the Interational Conference and Pattern Recognition*, 2003.
- [24] Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints”, *International Journal of Computer Vision*, 2004.
- [25] Bay, Ess, ir kt. „SURF: Speeded up robust features“, 2008.
- [26] Hugin. <http://hugin.sourceforge.net/>. [žiurèta 2010 05 07]

- [27] PTGui. <http://www.ptgui.com/>. [žiurėta 2010 05 07]
- [28] Panoweaver <http://www.easypano.com/panorama-software.html> [žiurėta 2010 05 07]
- [29] Google Street View <http://maps.google.com/help/maps/streetview/> [žiurėta 2010 05 07]
- [30] S. Chen. „QuickTime VR – An image-based approach to virtual environment navigation“. In SIGGRAPH'95, volume 29, 1995.
- [31] Flash player skvarvos analizė, http://www.adobe.com/products/player_census/flashplayer/, [Žiūrėta 2010 05 07]
- [32] Rob Hess realizuota SIFT bruožų paieška <http://web.engr.oregonstate.edu/~hess/> [žiurėta 2010 05 07]

7. Terminų ir santrumpų žodynas

API – programavimo sąsaja (Application programming interface)

DLT – Tiesioginė linijinė transformacija (Direct linear transformation)

SIFT – masteliui invariantiška savybių transformacija (Scale invariant feature transform)

SURF – pagreitinotos klaidom atsparios savybės (Speeded Up Robust Features)

RANSAC – atsitiktinių bandymų suderinamumas (Random sample consensus)

DoG – skirtumai tarp gretimų mastelių vaizdo ir Gauso filtro sąsukų (Difference of Gaussian)

ORM – objektinio ir reliacinio modelio sąryšis (Object-relational mapping)

VRML – Virtualios realybės modeliavimo kalba (Virtual reality modeling language)

STL – C++ metaprogramavimu grįsta standartinė biblioteka (Standard Template Library)

i18n – programos pritaikymas vertimui (internationalisation)

CSRF – tarptinklinis užklausos padirbimas (Cross-site request forgery)

8. Priedai

8.1. Konferencijos „Information Society and University studies“ 2010 pranešimo medžiaga

Automatinis vaizdų jungimas į panoramas

Andrius Paulavičius

Kauno technologijos universitetas, Informatikos
fakultetas, Sistemų analizės katedra, Kaunas
apaulavicius@gmail.com

Santrauka—Šiame straipsnyje apibendrintas automatizuoto įprastų nuotraukų jungimo į panoraminius vaizdus algoritmas, nereikalaujantis papildomų vartotojo pastangų.

Raktažodžiai – panoraminiai vaizdai, vaizdų bruožai, vaizdų suliejimas.

I. ĮVADAS

Panoramos yra didelio matymo kampo vaizdai. Kai panoramos apima visą cilindrinę ar sferinę erdvę esančią aplinkui stebėtoją, jas galima panaudoti kaip virtualias ekskursijas, kurios plačiai naudojamos tokiose srityse kaip turizmas. Panoramų kūrimą apsunkina aparatūrinių priemonių kaštai, programinių įrankių sudėtingumas ir apribojimai.

Vaizdų sujungimo metodai skiriami į dvi kategorijas – tiesioginius [1, 2] ir paremtus bruožais [3, 4]. Tiesioginių metodų esmė – rasti transformacijas tarp vaizdų, kurios minimizuotų persidengiančių vaizdų nepanašumo įverčius bendroje erdvėje. Šie metodai išnaudoja visą vaizdų informaciją, tačiau jiems būtina tiksliai parinkti pradinius parametrus. Bruožais paremtų metodų esmė – rasti transformacijas tarp vaizdų kurios minimizuotų bruožų taškų, sutapatintų keliuose vaizduose, padėčių skirtumus bendroje erdvėje. Jiems nebūtina tiksliai parinkti pradinius parametrus ir jų optimizavimo tikslo funkcijos apskaičiavimo sudėtingumas daug mažesnis.

Automatiniam vaizdų jungimui galima išskirti į tokius etapus [5]:

- Vaizdų invariantiškų bruožų išgavimas;
- Vaizdų bruožų sutapatinimas;
- Persidengiančių vaizdų radimas;

- Vaizdų fotoaparato geometrinių parametru nustatymas;
- Vaizdų suliejimas.

II. FOTOAPARATO GEOMETRINIS MODELIS

Panoramos jungiamos iš nuotraukų darytų iš bendro optinio centro. Dažnai naudojamas plačiai ištyrinėtas vieno parametro modelis, kai fotoaparatas sukamas tik aplink Y ašį ir vaizdai sujungiami į cilindrinės panoramas.

Transformaciją tarp vaizdo koordinatų ir jų atitikmenų kitame vaizde nusako homografijos matrica:

$$\tilde{u}_i = H_{ij} \tilde{u}_j \quad (1),$$

kur H_{ij} – homografija iš vaizdo j į vaizdą i , \tilde{u}_i ir \tilde{u}_j homogeninės koordinatės vaizduose i ir j .

Bendriausias vaizdo fotoaparato modelis nusakomas keturiais parametrais: posūkio kampų vektoriumi $\theta = [\theta_{ix} \ \theta_{iy} \ \theta_{iz}]$ ir židinio nuotoliu f . Tokiu atveju homografijos matricą galima išreikšti vaizdų posūkio matricų R_i ir mastelio keitimo matricų K_i sandaugomis:

$$H_{ij} = K_i R_i R_j^T K_j^{-1} \quad (2)$$

$$K_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$R_i = e^{\begin{bmatrix} 0 & -\theta_{iz} & \theta_{iy} \\ \theta_{iz} & 0 & -\theta_{ix} \\ -\theta_{iy} & \theta_{ix} & 0 \end{bmatrix}}$$

$$(4) \quad \begin{bmatrix} 2(x^2-1)s^2-1 & 2xys^2-2zcs & 2xzs^2+2yys \\ 2xys^2+2zcs & 2(y^2-1)s^2+1 & 2yzs^2-2xcs \\ 2xzs^2-2yys & 2yzs^2+2xcs & 2(z^2-1)s^2+1 \end{bmatrix},$$

kur posūkių vektorius išreikštas normalizuota forma $[\theta_{ix} \ \theta_{iy} \ \theta_{iz}] = [x \ y \ z] \cdot \theta$, $c = \cos(\theta/2)$ ir $s = \sin(\theta/2)$.

III. VAIZDŲ BRUOŽAI

Lokalių fotometrinių bruožų aprašymai sėkmingai taikomi objektų atpažinimui, tekstūrų atpažinimui ir vaizdų tarpusavio padėčių nustatymui. Paprasčiausio bruožo pavyzdys yra taškų intensyvumų vektorius. Nauji vaizdų bruožai yra dalinai invariantiški afiniosioms transformacijoms ir leidžia rasti atitikmenis tarp vaizdų dengiančių bendrą erdvę, jei žiūrėjimo kampo pokyčiai nėra dideli.

A. SIFT bruožai

SIFT(Scale Invariant Feature Transform) [6] bruožai yra rutulio formos vaizdo plotai, turintys padėtį, mastelį ir kryptį. Bruožų paieška vykdoma įvairiuose masteliuose ir padėtyse, todėl SIFT apibūdinimai yra invariantiški postūmiui, posūkiui ir mastelio keitimo transformacijoms. SIFT bruožai randami ieškant ekstremumų vaizdo ir Gauso filtrų sąsūkių skirtumų (DoG) erdvėje. Jų apibūdinimai yra 128 elementų vektoriai, nusakantys gradiento krypties histogramas aplink bruožų taškus.

B. SURF bruožai

SURF(Speeded Up Robust Features) [7] bruožai sukurti stengiantis paspartinti bruožų išgavimą. Jų paieška vykdoma naudojant 2D Haro bangelių filtrus Hesiano determinanto aproksimavimui. SURF išgavimas efektyviai išnaudoja tarpinius skaičiavimo vaizdus ir yra kelis kartus greitesnis už SIFT. Šių bruožų apibūdinimai yra 64 elementų vektoriai nusakantys šviesumų intensyvumų pasiskirstymą aplink bruožų taškus.

IV. BRUOŽŲ SUTAPATINIMAS

Kiekvienas vaizdas gali persidengti su bet kuriuo kitu ir vienas bruožo taškas kartotis keliuose vaizduose, todėl kiekvienam bruožui reikia ieškoti kelių artimiausių kaimynų jų apibūdinimų erdvėje. Nėra efektyvesnio būdo didelių dimensijų objektų kaimynų išrinkimui nei visų objektų palyginimas su visais, tačiau įmanoma rasti apytikslius kaimynus panaudojant k-d medį. Šis medis yra dvejetainis erdvės išskaidymas, kuris rekursyviai erdvę dalina per vidurkius didžiausio kitimo dimensijoje. Naudojant šią duomenų struktūrą, apytikslius artimiausius kaimynus įmanoma rasti per $O(n \log n)$ laiką [8].

V. PERSIDENGIANČIŲ VAIZDŲ RADIMAS

Po bruožų sutapatinimo yra žinomos vaizdų poros, kurios turi daugiausiai artimų bruožų, o tuo pačiu ir didžiausią potencialiai persidengiantį plotą. Iš didžiausio

persidengiančio vaizdų rinkinio bus sujungiami galutinė panorama. RANSAC metodu galima išrinkti didžiausius rinkinius sutapatintų bruožų, kurie yra geometriškai suderinami, t. y. tarp jų egzistuoja homografija (žr. paveikslą 1). Ar vaizdai persidengia galima spręsti iš RANSAC patvirtintų ir atmestų sutapatinimų santykio.

A. RANSAC

RANSAC (RANdom SAMple Consensus) [9] yra modelio parametrų nustatymo iš eksperimentinių duomenų metodas, kuriame modelio parametrai nustatomi iš atsitiktinai išrinkto minimalaus duomenų egzempliorių rinkinio. Procedūra kartojama daug kartų, ieškant geriausiai su eksperimentiniais duomenimis sutampančių parametrų. Kai bandymų kiekis pakankamai didelis, tikimybė rasti geriausius parametrus artėja į vienetą, o atsitiktinis parinkimas užtikrina atsparumą klaidoms eksperimentiniuose duomenyse.

Homografijai tarp dviejų vaizdų nustatyti DLT (Direct Linear Transformation) metodu [10] reikia keturių šių vaizdų sutapatintų bruožų porų padėčių. Homografijos H elementai randami išsprendus tiesinę lygčių sistemą:

$$\begin{bmatrix} 0 & 0 & 0 & -u_{j1x} & -u_{j1y} & -1 & u_{i1x} \cdot u_{j1y} & u_{i1y} \cdot u_{j1y} \\ u_{j1x} & u_{j1y} & 1 & 0 & 0 & 0 & -u_{i1x} \cdot u_{j1x} & -u_{i1y} \cdot u_{j1x} \\ 0 & 0 & 0 & -u_{j2x} & -u_{j2y} & -1 & u_{i2x} \cdot u_{j2y} & u_{i2y} \cdot u_{j2y} \\ u_{j2x} & u_{j2y} & 1 & 0 & 0 & 0 & -u_{i2x} \cdot u_{j2x} & -u_{i2y} \cdot u_{j2x} \\ 0 & 0 & 0 & -u_{j3x} & -u_{j3y} & -1 & u_{i3x} \cdot u_{j3y} & u_{i3y} \cdot u_{j3y} \\ u_{j3x} & u_{j3y} & 1 & 0 & 0 & 0 & -u_{i3x} \cdot u_{j3x} & -u_{i3y} \cdot u_{j3x} \\ 0 & 0 & 0 & -u_{j4x} & -u_{j4y} & -1 & u_{i4x} \cdot u_{j4y} & u_{i4y} \cdot u_{j4y} \\ u_{j4x} & u_{j4y} & 1 & 0 & 0 & 0 & -u_{i4x} \cdot u_{j4x} & -u_{i4y} \cdot u_{j4x} \end{bmatrix} \cdot \begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \end{bmatrix} = \begin{bmatrix} -u_{i1y} \\ u_{i1x} \\ -u_{i2y} \\ u_{i2x} \\ -u_{i3y} \\ u_{i3x} \\ -u_{i4y} \\ u_{i4x} \end{bmatrix}$$

$$H_{33} = 1. \quad (5)$$

Kiekviename bandyme rastos homografijos kokybė įvertinama pagal tai, kiek vieno vaizdo sutapatintų bruožų padėčių transformuotų H yra artimi kito vaizdo bruožų padėtimis.

B. Persidengimo patvirtinimas

Norint nustatyti ar vaizdai persidengia, galima naudoti tikimybinę funkciją, priklausančią nuo atmestų ir patvirtintų sutapatinamų bruožų santykio. Vaizdų poros patikrinimo slenkstinė funkcija:

$$n_i > \alpha + \beta \cdot n_f, \quad (6)$$

kur n_i – RANSAC rastą homografiją tenkinančių sutapatintų bruožų kiekis, n_f – visų sutapatintų bruožų kiekis.



Paveikslas 1. Persidengiantys vaizdai su jų bruožais; pagal apibūdinimus rasti bruožų sutapatinimai; didžiausias homografija suderinamų sutapatinimų rinkinys

VI. FOTOAPARATŲ PARAMETRŲ NUSTATYMAS

C. Reliatyvių būsenų nustatymas

Rastų homografijų sujungimas sukauptų dideles paklaidas. Kokybiškam būsenų radimui, reikia ieškoti visų vaizdų fotoaparatus parametrų kartu. Tai galima atlikti ryšulinio koregavimo metodu [11].

Fotoaparatus parametrų optimizavimo procedūra atliekama su geriausiai persidengiančių vaizdų pora ir kartojama įterpiant po naują vaizdą, geriausiai persidengiantį su jau optimizuotu rinkiniu. Naujai įterpto vaizdo fotoaparatus pradiniai parametrai nustatomi tokie patys kaip jau optimizuoto vaizdo su kuriuo jis geriausiai persidengia. Persidengimo kokybė vertinama pagal tarpusavyje sutapatintų bruožų kiekį. Optimizuojama funkcija bruožus projektuoja į optimizavime dalyvaujančius vaizdus, kuriuose yra jų atitikmenys ir sumuoja atstumų tarp jų kvadratus. Šią funkciją reikia minimizuoti fotoaparatus parametrų atžvilgiu.

Sutapatinto bruožo taško i – ajame ir j – ajame vaizduose $u_i^k \Leftrightarrow u_j^l$ projekcijos skirtumas yra:

$$r_{ij}^k = u_i^k - p_{ij}^k, \quad (7)$$

kur p_{ij}^k yra taško atitinkančio u_i^k vaizde j projekcija vaizde i .

$$\tilde{p}_{ij}^k = K_i R_i R_j^T K_j^{-1} \tilde{u}_j^l \quad (8)$$

Optimizuojama funkcijas yra:

$$e = \sum_{i=1}^n \sum_{j \in I(i)} \sum_{k \in F(i,j)} f(r_{ij}^k), \quad (9)$$

kur n – vaizdų kiekis, $I(i)$ – vaizdai persidengiantys su vaizdu i , $F(i,j)$ – sutapatinti vaizdų i ir j bruožų taškai. f – atsparumą klaidoms suteikianti funkcija.

$$f(x) = \begin{cases} |x|^2, & \text{kai } |x| < \sigma \\ 2\sigma|x| - \sigma^2, & \text{kai } |x| \geq \sigma \end{cases} \quad (10)$$

Funkcija $f(x)$ suteikia atsparumą klaidoms duomenyse. Ji naudoja antro laipsnio normą greitam konvergavimui gerai sutapatintiems nariams ir pirmo laipsnio normą tolimiems nariams.

Tai yra netiesinis mažiausių kvadratų uždavinys, kurį sprendžia Levenberg-Marquardt algoritmas. Šis algoritmas yra Gauso-Niutono iteracinio optimizavimo algoritmo modifikacija, kurioje įvestas slopinimo parametras pagreitinantis konvergavimą ir leidžiantis rasti minimumą prastai parinkus pradinius parametrus.

Levenberg-Marquardt algoritmo iteracija yra tiesinių lygčių sistemos sprendimas:

$$(J^T J + \lambda \cdot \text{diag}(J^T J))(\Delta \Theta) = J^T r, \quad (11)$$

kur Θ – visų fotoaparatus parametrai surašyti į vieną vektorių, $\Delta \Theta$ ieškomas parametrų pokytis, r – projektavimo paklaidos surašytos į vieną vektorių, J – Jakobiano matrica $J = \frac{\delta r}{\delta \Theta}$, λ – slopinimo parametras, kurio vertė keičiama priklausomai nuo minimizuojamos funkcijos kitimo greičio (funkcijai smarkiai pagerėjus – mažinama, mažai – didinama), diag – matricos diagonalizavimo funkcija.

Jakobianą sudarančios išvestinės skaičiuojamos pagal jų analitinės išraiškas. Pvz., išvestinė pagal i – tojo fotoaparatus parametru θ_x :

$$\frac{\delta r_{ij}^k}{\delta \theta_{ix}} = 0 - \frac{\delta p_{ij}^k}{\delta \theta_{ix}} = - \left(\frac{\delta p_{ij}^k}{\delta \tilde{p}_{ij}^k} \frac{\delta \tilde{p}_{ij}^k}{\delta \theta_{ix}} \right) \quad (12)$$

$$\frac{\delta \tilde{p}_{ij}^k}{\delta \theta_{ix}} = K_i \frac{\delta R_i}{\delta \theta_{ix}} R_j^T K_j^{-1} \tilde{u}_j^l = K_i R_i \cdot \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} R_j^T K_j^{-1} \tilde{u}_j^l \quad (13)$$

$$\frac{\delta p_{ij}^k}{\delta \tilde{p}_{ij}^k} = \frac{\delta [x/z, y/z]}{\delta [x, y, z]} = \begin{bmatrix} 1/z & 0 & -x/z^2 \\ 0 & 1/z & -y/z^2 \end{bmatrix} \quad (14)$$

Praktiškai vienas vaizdas persidengia su nedideliu kiekiu kitų vaizdų. Tai reiškia, kad Jakobiano matrica yra parėnta ir algoritmą galima paspartinti neskaiciuojant didelių matricų sandaugų, o užpildant sudaugintos formos submatricas:

$$(J^T J)_{4i:4(i+1), 4j:4(j+1)} = \sum_{k \in F(i, j)} \left(\frac{\delta r_{ij}^k}{\delta \Theta_i} \right)^T \frac{\delta r_{ij}^k}{\delta \Theta_j} \quad (15)$$

$$(J^T r)_{4i:4(i+1)} = \sum_{j \in I(i)} \sum_{k \in F(i, j)} \left(\frac{\delta r_{ij}^k}{\delta \Theta_i} \right)^T r_{ij}^k. \quad (16)$$

I. Globalios būsenos nustatymas

Po reliatyvių fotoaparatus būsenų nustatymo lieka nežinoma fotoaparatus padėtis pasaulio koordinatų atžvilgiu. Jei vieno fotoaparatus posūkiu parametrus laikome globaliais, išėjime galima gauti banguotą panoramą, nes mažai tikėtina, kad fotografavimo metu kamera buvo laikoma tobulai tiesiai (žr. paveikslą 2). Šį efektą galima pataisyti naudojant euristiką, kad įprastai fotografuojant panoramas fotoaparatus nesukamas horizonto atžvilgiu, dėl ko jų X ašių vektoriai randasi plokštumoje. Surašius visų fotoaparatus posūkių matricių X ašių kryptis reliatyvias pasirinktam fotoaparatus į matricę X, galima rasti jiems statmeną vektorių Y išsprendžiant lygčių sistemą:

$$(X X^T) Y = [0 \ 0 \ 0]^T. \quad (17)$$

Globalią posūkiu būseną nusako pasirinkto fotoaparatus posūkiu transformacija padauginta iš transformacijos pasukančios Y iki vertikalios padėties t. y. posūkiu per ašį $A = Y \times [0 \ 1 \ 0]^T$ kampu $\theta = \arccos(Y \cdot [0 \ 1 \ 0]^T)$.

Posūkiu kampu per ašį transformacijos matrica:

$$T = I \cdot \cos(\theta) + \begin{bmatrix} 0 & -Z_3 & Z_2 \\ Z_3 & 0 & -Z_1 \\ -Z_2 & Z_1 & 0 \end{bmatrix} \cdot \sin(\theta) + (1 - \cos(\theta)) \cdot A A^T, \quad (18)$$

kur I – vienietinė matrica, A – posūkiu ašis, θ – posūkiu kampas.



Paveikslas 2. Panoramos fragmentas prieš ir po globalaus posūkiu transformacijos

VII. VAIZDŲ SULIEJIMAS

Žinant n vaizdų $I^i(x, y)$ fotoaparatus parametrus galima juos pervesti į bendras sferines koordinates $I^i(\theta, \phi)$, trečią sferinių koordinatų parametru, nusakantį atstumą, laikant konstanta. Vaizdus sujungiant į galutinę panoramą $I(\theta, \phi)$, svarbu panaudoti kokybišką suliejimo metodą ekspozicijos pokyčių, vaizdų iškreipimų ir paklaidų kompensavimui. Suliejimo strategijos pavaizduotos paveiksle 4.

A. Sujungimas be suliejimo

Paprasciausias $I(\theta, \phi)$ sudarymas gali vykti kiekvienam išėjimo taškui išrenkant jį geriausiai atitinkantį tašką iš vieno atskiro vaizdo:

$$I(\theta, \phi)_{\text{be suliejimo}} = I^i(\theta, \phi), \text{ kai } W^i(\theta, \phi) = \max(W(\theta, \phi)). \quad (19)$$

Kadangi mažiausi optiniai iškreipimai pasireiškia nuotraukų centruose, pasirinkta svorinė funkcija $W^i(\theta, \phi)$ lygi 1 vaizdo centre ir tiesiškai mažėja iki 0 kraštuose.

B. Tiesinis suliejimas

Tiesinis suliejimas leidžia kompensuoti nevienodą ekspoziciją tarp persidengiančių vaizdų. Panaudojama visa persidengiančių vaizdų informacija, jos svarbumą įvertinant pagal svorinę funkciją:

$$I(\theta, \phi)_{\text{tiesinis}} = \frac{\sum_{i=1}^n I^i(\theta, \phi) W^i(\theta, \phi)}{\sum_{i=1}^n W^i(\theta, \phi)}. \quad (20)$$

C. Daugiaujustis suliejimas

Tiesinis suliejimas nėra atsparus klaidoms ir esant objektyvo sukeliams iškreipimams, judesiu ar mažom fotoaparatus parametru nustatymo paklaidoms, suliejime aukšto dažnio detalės tampa miglotos. Šią problemą galima dalinai išspręsti naudojant daugiaujustį suliejimą [12]. Metodo idėja yra suskaidyti vaizdus į juostas pagal dažnius, suliejant žemo dažnio juostas dideliuose plotuose, o aukšto dažnio mažuose plotuose.

Didžiausių dažnių juostos ir jų svoriai suformuojami taip:

$$B_1^i(\theta, \phi) = I^i(\theta, \phi) - I^i(\theta, \phi) * g_\sigma(\theta, \phi) \quad (21)$$

$$W_{\max}^i(\theta, \phi) = \begin{cases} 1, & \text{kai } W^i(\theta, \phi) = \max(W(\theta, \phi)) \\ 0, & \text{kitu atveju} \end{cases} \quad (22)$$

$$W_1^i(\theta, \phi) = W_{\max}^i(\theta, \phi) * g_\sigma(\theta, \phi), \quad (23)$$

kur $g_\sigma(\theta, \phi)$ – 2D Gauso funkcija su standartiniu nuokrypiu σ , perversa į sferines koordinates, $B_1^i(\theta, \phi)$ – dažnių juosta kurios bangų ilgis nuo 0 iki σ , $W_1^i(\theta, \phi)$ – vaizdo i svorinė funkcija bangų ilgiui nuo 0 iki σ , * – sąsūkos operacija.

Kitos dažnių juostos ir jų svoriai suformuojamos taip:

$$I_{k+1}^i(\theta, \phi) = I_k^i(\theta, \phi) * g_\sigma(\theta, \phi) \quad (24)$$

$$B_{(k+1)}^i(\theta, \phi) = I_k^i(\theta, \phi) - I_{k+1}^i(\theta, \phi) \quad (25)$$

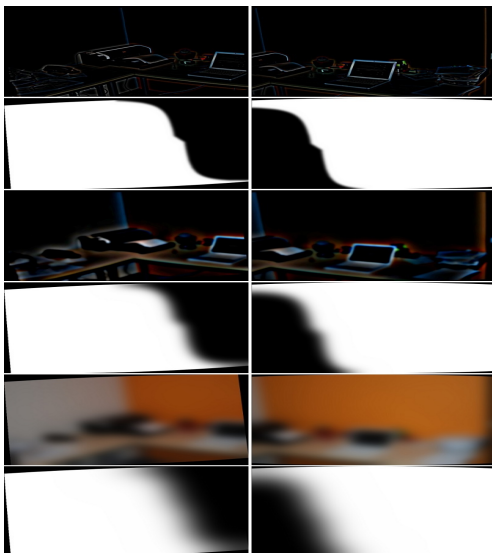
$$W_{(k+1)}^i(\theta, \phi) = W_k^i(\theta, \phi) * g_\sigma(\theta, \phi) \quad (26)$$

Paskutinė juosta sudaroma iš visų dar nepaskirstytų žemų dažnių (žr. paveikslą 3).

Kiekvienai juostai persidengiantys vaizdai tiesiškai suliejami, o galinis rezultatas gaunamas susumavus visų juostų vertes:

$$B(\theta, \phi)_k = \frac{\sum_{i=1}^n B_k^i(\theta, \phi) W_k^i(\theta, \phi)}{\sum_{i=1}^n W_k^i(\theta, \phi)} \quad (27)$$

$$I(\theta, \phi)_{daugiaujustis} = \sum_{i=k} B(\theta, \phi)_i \quad (28)$$



Paveikslas 3. Persidengiančių vaizdų dažnių juostos $B_k^i(\theta, \phi)$ ir jų svorinės funkcijos $W_k^i(\theta, \phi)$ kai k kinta nuo 1 iki 3

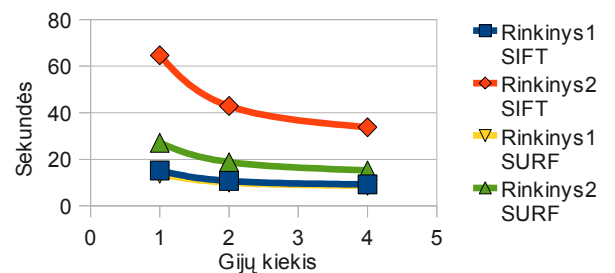


Paveikslas 4. vaizdų sujungimas be suliejimo, panaudojant tiesinį suliejimą ir panaudojant daugiajuostį suliejimą

VIII. REZULTATAI.

Buvo realizuotas iš aprašytų žingsnių sudarytas algoritmas C++ kalba, panaudojant atviro kodo NASA Vision Workbench ir OpenCV paketus. Realizacijos paspartinimui panaudotos lygiagrečios programavimo priemonės.

Lygiagrečiai atliekami tarpusavyje duomenimis nepriklausomi veiksmai (bruožų išgavimas, vaizdo bruožų kaimynų paieška, vaizdų dažnių juostų išgavimas, vaizdų transformavimas, svorinių funkcijų generavimas ir parametrų nustatymo submatricių sudarymas) ir suliejimas (atskiroms gijoms išėjimo vaizdo pildymą paskirsčius pagal vertikalų plotą).



Paveikslas 5. Veikimo laiko įvertinimas

Bandytas atliktas su keturių branduolių 2.6GHz Intel Core2 Q9400 procesorių turinčiu kompiuteriu. Kiekvienam vaizdui vertinama iki 500 geriausių bruožų, vienam vaizdui vertinama iki 4 geriausių persidengimų, naudojamos 3 dažnių juostų suliejimas, išėjime gaunamos 2000 pikselių pločio panoramos. Pirmas rinkinys sudarytas iš 7 nuotraukų, antras iš 27. Laikai buvo vertinti tik tais atvejais, kai abiejų rūšių bruožais buvo nustatyti teisingi fotoaparato parametrai. Veikimo laiko įvertinimas pateiktas paveiksle 5.

IX. IŠVADOS

Šiuolaikiniai vaizdų bruožai leidžia rasti persidengiančius vaizdus ir jų santykinę padėtį nereikalaujant vartotojo įsikišimo.

Automatizuotas vaizdų jungimas neįmanomas, kai tarp jų nėra aiškių bendrų bruožų (pvz. buto lubos), arba kai bruožai persidengiančiuose vaizduose smarkiai skiriasi dėl atspindžių ar permatomumo.

Dauguma pateikto algoritmo žingsnių operuoja dideliais kiekiais tarpusavyje nepriklausomų duomenų. Šiuos žingsnius nesudėtinga atlikti lygiagrečiai duomenų atžvilgiu.

SURF bruožai išgaunami žymiai greičiau nei SIFT, jų dvigubai mažesnės dimensijos apibūdinimo vektoriai smarkiai paspartina kaimyninių bruožų paiešką. Tačiau praktiniuose bandymuose SURF bruožai teisingai sutapatinami rečiau nei SIFT.

LITERATŪRA

- [1] H. Shum ir R. Szeliski. „Construction of panoramic mosaics with global and local alignment“, International Journal of Computer Vision, 2000.
- [2] H. Sawhney and R. Kumar. „True multi-image alignment and its application to mosaicing and lens distortion correction“, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999.
- [3] P. McLauchlan, A. Jaenicke. „Image mosaicing using sequential bundle adjustment“, Image and Vision Computing, 2002.
- [4] D. Capel ir A. Zisserman. „Automated mosaicing with super-resolution zoom“, Proceedings of the International Conference on Computer Vision and Pattern Recognition, 1998.
- [5] M. Brown, „Multi-Image Matching Using Invariant Features“, University of British Columbia, 2005.
- [6] D. G. Lowe „Distinctive Image Features from Scale-Invariant Keypoints“, International Journal of Computer Vision, 2004.
- [7] H. Bay, A. Ess, T. Tuytelaars, L. V. Gool. „SURF: Speeded up robust features“, Computer Vision and Image Understanding, 2008.
- [8] Marius Muja ir David G. Lowe, „Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration“, International Conference on

Computer Vision Theory and Applications, 2009.

- [9] M. Fischler ir R. Bolles, „Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography“, Communications of the ACM, 1981.
- [10] R. Hartley ir A. Zisserman. „Multiple View Geometry in Computer Vision“. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [11] W. Triggs, P. McLauchlan, R. Hartley, ir A. Fitzgibbon. „Bundle adjustment: A modern synthesis“. Vision Algorithms: Theory and Practice, 1999.
- [12] P. Burt ir E. Adelson. „A multiresolution spline with application to image mosaics“, ACM Transactions on Graphics, 1983.
- [13] M. Brown, R. Hartley ir D. Nister, „Minimal Solutions for Panoramic Stitching“, International Conference on Computer Vision and Pattern Recognition, Minneapolis, 2007