

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS IR GAMTOS MOKSLŲ FAKULTETAS
ELEKTRONIKOS KATEDRA

Alexander Abramovich

ERDVINIO VAIZDO ALGORITMŲ Palyginimas
Magistro darbas

Vadovas

doc. dr. D. Dervinis

ŠIAULIAI, 2014

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS IR GAMTOS MOKSLŲ FAKULTETAS
ELEKTRONIKOS KATEDRA

TVIRTINU

Katedros vedėjas

prof. dr. Gintautas Daunys

2014 06

ERDVINIO VAIZDO ALGORITMŲ Palyginimas
Magistro darbas

Vadovas

doc. dr. D. Dervinis

2014 06

Recenzentas

Atliko

RM12 gr. stud.

Alexander Abramovich

2014 06

ŠIAULIAI, 2014

Abramovich A. Comparison of Stereo Vision Algorithms: Master's Thesis in Electronic Engineering / academic advisor: Doc. Dr. Dervinis D., Šiauliai University, Faculty of Technology, Department of Electronics. – Šiauliai, 2014. 48 pp.

SUMMARY

Computer stereo vision is a receiving of stereo information from digital images. This field of science is rather new and its popularity is increasing rapidly. Computer stereo vision is applied in robotics, manufacturing industry, everyday life and other spheres.

The aim of the Thesis is to analyze and compare the stereo vision algorithms. In order to achieve the aim of the Research, the following tasks are determined: to classify the stereo vision algorithms, to study the methods of algorithm design, to create the method of assessment of stereo vision algorithms and to assess the stereo vision algorithms basing on the ground of this method.

Basing on academic works of various authors, the stereo vision algorithms are classified and divided into two groups in accordance with the modes of their functioning: local and global algorithms. Several algorithms are chosen from each group. A based local algorithm with different mathematical solutions is chosen from the group of local algorithms, and the dynamic programming is chosen from the group of the global algorithms. The chosen algorithms are tested in the course of the Research. To compare the mentioned algorithms, the assessment method is prepared, the main criteria of which are the correlation index and the running time of algorithm. All abovementioned algorithms are tested by the means of the method of assessment. The best results are chosen basing on the method and results of the tests.

The results of the conducted Research show that the tested algorithms are not perfect, but they are appropriate for use with some limitations. Moreover, the imperfection of algorithms evidences that the abovementioned field of science has not been studied completely what affords to the scientists the ground for further development of existing algorithms and creation of the new ones.

TURINYS

ĮVADAS	8
1. KOMPIUTERINĖS ERDVINĖS REGOS IR JOS ALGORITMŲ ANALIZĖ	9
1.1. Stereo sistemos modelis.	9
1.2. Disparitiškumas ir vaizdo gylis	11
1.3. Horopteris	12
1.4. Standartinis stereo sistemos modelis.....	14
1.5. Algoritmai.	16
1.5.1. Bazinis lokalinis sprendimas	16
1.5.2. Bazinio lokalinio sprendimo patobulinimas.	17
1.5.3. Atitikimo skaičiavimo algoritmai.	18
1.5.4. Verčių sumavimas.	20
1.6. Standartiniai stereo vaizdų rinkiniai skirti algoritmų testavimui.	21
2. EKSPERIMENTINIS TYRIMAS	22
2.1 Bazinis lokalinis sprendimas.....	22
2.1.2 Bazinio lokalinio sprendimo algoritmo tyrimo rezultatai.	25
2.1.3. Bazinio lokalinio algoritmo tyrimo išvados.....	33
2.3. Dinaminis programavimas.	37
2.3.1. Dinaminio programavimo algoritmo aprašymas.	37
2.3.2. Dinaminio programavimo algoritmo bandymų rezultatai.....	39
2.3.3. Dinaminio programavimo algoritmo bandymų išvados.	42
2.4. Išbandytų algoritmų suvestinė.	44
IŠVADOS	46
LITERATŪRA.....	47
PRIEDAI.....	49

LENTELĖS

2.1 lentelė. Geriausi SAD be filtravimo bandymų rezultatai.....	26
2.2 lentelė. Geriausi, pagal algoritmo veikimo laiką, SAD be filtravimo bandymų rezultatai.....	28
2.3 lentelė . SAD algoritmo rezultatai po medianos filtravimo.....	29
2.4 lentelė. SAD algoritmo be filtravimo ir po medianos filtravimo.....	30
2.5 lentelė. SAD algoritmo rezultatai po Vinerio filtro pritaikymo.....	31
2.6 lentelė. SAD algoritmo rezultatų palyginimas be filtravimo ir po Vinerio filtro pritaikymo.....	31
2.7 lentelė. SSAD algoritmo bandymų rezultatai be filtravimo	priedas Nr. 8
2.8 lentelė. SSAD algoritmo bandymų rezultatai po medianos filtravimo	priedas Nr. 8
2.9 lentelė. SSAD algoritmo bandymų rezultatai po medianos ir Vinerio filtravimo	priedas Nr. 8
2.10 lentelė. SAD algoritmo bandymų rezultatų skirtumas be filtravimo i po jo.....	32
2.11 lentelė. STAD algoritmo bandymų rezultatai be filtravimo	priedas Nr. 9
2.12 lentelė. STAD algoritmo bandymų rezultatai po medianos filtravimo	priedas Nr. 9
2.13 lentelė. STAD algoritmo bandymų rezultatai po medianos ir Vinerio filtravimo	priedas Nr. 9
2.14 lentelė. STAD algoritmo bandymų rezultatų skirtumas be filtravimo i po jo.....	33
2.15 lentelė. Geriausių basic approach bandymų rezultatų suvestinė.....	34
2.16 lentelė. Geriausių tarp-piksleninės analizės bandymų rezultatų suvestinė.....	36
2.17 lentelė. Dinaminio programavimo bandymų rezultatai be filtravimo pritaikymo.....	39
2.18 lentelė. Dinaminio programavimo bandymų rezultatai pritaikant medianos filtrą.....	40
2.19 lentelė. Dinaminio programavimo bandymų rezultatai pritaikant medianos ir Vinerio filtrus.....	41
2.20 lentelė. Geriausių dinaminio programavimo algoritmo bandymų rezultatų suvestinė.....	42

PAVEIKSLAI

1.1 pav. Dažniausiai naudojamas kompiuterinio erdvinio vaizdo modelis.	9
1.2 pav. Kompiuterinės regos modelis sudarytas iš šviesos šaltinio ir vienos kameros.	10
1.3 pav. Nekalibruoto erdvinio modelio kamerų plokštumos	10
1.4 pav. „Standartizuotos“ (kalibruotos) stereo kameros modelis.	11
1.5 pav. Vaizdo gylio skaičiavimo modelis.	12
1.6 pav. Horopteris.	12
1.7 pav. Diskretizuotas horopteris.	13
1.8 pav. Apribotas diskretizuotas horopteris.	13
1.9 pav. Erdvinės sistemos struktūrinė veikimo schema.	14
1.10 pav. Atitinkamų taškų paieška epipoliarinėje tiesėje.	17
1.11 pav. „Winner takes all“ algoritmo atvaizdavimas.	17
1.12 pav. Stereo vaizdų pikselių „svorių“ skirtumas.	19
1.13 pav. Fiksuoto lango skaičiavimas epipoliarinėje tiesėje.	19
1.14 pav. a) fiksuoto lango algoritmo skaičiavimo rezultatas; b) idealus/etaloninis rezultatas.	20
1.15 pav. Netikslumai gaunami prie „perėjimo“ sričių.	20
1.16 pav. Netikslumų pašalinimas, mažinant papildomų laukų kiekį.	21
2.1 pav. Basic approach algoritmas.	23
2.2 pav. Maksimalus SAD koreliacijos koeficientai be filtravimo.	26
2.3 pav. Bendrojo koreliacijos koeficiento priklausomybė nuo parinkto lango dydžio ir disparitiškumo.	27
2.4 pav. Algoritmo veikimo laiko priklausomybė nuo lango dydžio ir disparitiškumo.	28
2.5 pav. Bazinio lokalinio algoritmo geriausias rezultatas pagal kor. koeficientą.	35
2.6 pav. Geriausias tarp-pikselinės analizės rezultato erdvinis atvaizdavimas.	37
2.7 pav. Dinaminio programavimo algoritmas.	38
2.8 pav. Dinaminio programavimo algoritmo disparitiškumo atvaizdavimas: a) prieš; b) po filtravimo.	43
2.9 pav. Geriausio dinaminio programavimo algoritmo gauto disparitiškumo erdvinis atvaizdavimas.	44

PRIEDAI

- 1 PRIEDAS. Stereo kameros kalibravimo principas.
- 2 PRIEDAS. Stereo vaizdų pora, apskaičiuotas ir atraminis (angl. *groundtruth*) disparitiškumas.
- 3 PRIEDAS. Vidinių kameros iškraipymų taisymas po kalibravimo.
- 4 PRIEDAS. Tsukuba stereo porų rinkinys.
- 5 PRIEDAS. Bazinio lokalinio sprendimo (angl. *basic approach*) MatLab2013b programos tekstas.
- 6 PRIEDAS. Bazinio lokalinio sprendimo tyrimo rezultatai.
- 7 PRIEDAS. Triukšmų, atsirandančių nuo mažo slenkančio lango parinkimo, atvaizdavimas.
- 8 PRIEDAS. SSAD algoritmo bandymų suvestinė.
- 9 PRIEDAS. STAD algoritmo bandymų suvestinė.
- 10 PRIEDAS. Medianos ir Vinerio filtravimo pavyzdys.
- 11 PRIEDAS. Tarp pikselinės analizės tyrimo rezultatai.
- 12 PRIEDAS. Dinaminio programavimo programos tekstas.
- 13 PRIEDAS. Dinaminio programavimo tyrimo rezultatai.

IVADAS

Šiais laikais, informacinių technologijų greitas tobulėjimas, padidino automatizuotų ir robotizuotų sistemų populiarumą. Šis populiarumas atsispindi ne tik pramonėje ar moksle, bet ir mūsų kasdieniniame gyvenime. Dėl to atsiranda efektyvių gautos informacijos apdorojimo metodų paklausa, ypač kompiuterinės regos duomenys. Efektyvus gautų duomenų apdorojimas ir panaudojimas leidžia padidinti sistemos našumą ir praplėsti jos panaudojimo sritis.

Kompiuterinė rega (mašininė rega) – techninių ir programinių sprendimų rinkinys, leidžiantis gauti skaitmeninį vaizdą, jį apdoroti ir pateikti rezultatai tokia forma, kuri bus galima praktiškai panaudoti realaus laiko mašinose. Kompiuterinė rega skirstoma į monokuliarinę ir binokuliarinę. Jos atitinkamai skirtos apdoroti plokščius ir erdvinius vaizdus.

Nepaisant didelių pasiekimų šioje srityje, efektyvus kompiuterinės regos, kaip automatizuotos ir nepriklausomos valdymo sistemos, panaudojimas yra įmanomas tik labai aukšto lygio technologiniuose procesuose. Kitais atvejais kompiuterinės regos užduotis apsiriboja kitų sistemų papildymu. Tai praplėčia žmogaus vaizdo apdorojimo galimybes. Reikia pažymėti, kad neegzistuoja universalūs vaizdo apdorojimo metodai, kurie leistų suformuoti standartinį sprendimų algoritmą gaminant kompiuterinės regos sistemas. Todėl, projektuojant arba ruošiant kompiuterinės regos specialistus, labai svarbios yra publikacijos ir kiti moksliniai darbai, kuriuose yra aprašomi atskiri matematiniai algoritmai sprendžiantys keliamas užduotis.

Šiame darbe planuojama aprašyti, išbandyti ir palyginti kelis skirtingus erdvinio vaizdo algoritmus. Nustatyti nuo kokių konfigūravimo parametrų priklauso jų veikimas ir gaunamų rezultatų kokybė. Algoritmų palyginimas turi būti atliekamas vadovaujantis sukurta ir aprašyta įvertinimo metodiką. Algoritmų bandymuose planuojama naudoti standartinius erdvinio vaizdų rinkinius, tokius kaip „tsukuba“ [3], „cones“ ir pan. Kiekvieno algoritmo bandymo rezultatai bus suvedami į atskiras lentelės, o iš jų bus išrenkami geriausi ir vadovaujantys jais daromos išvados.

Darbo tikslas :

- išanalizuoti ir palyginti erdvinio vaizdo algoritmus.

Darbo uždaviniai:

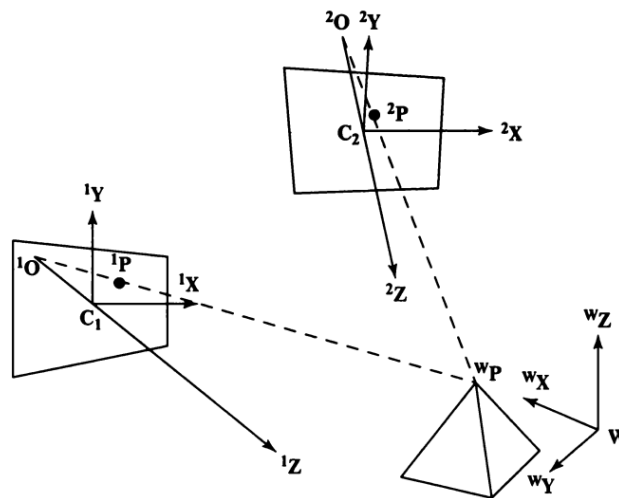
- suklasifikuoti erdvinio vaizdo algoritmus;
- apžvelgti erdvinio vaizdo sudarymo būdus ir metodus;
- sukurti erdvinio vaizdo vertinimo metodiką ir kriterijus;
- sukurti vertinimo metodiką ir vadovaujantis ja palyginti erdvinio vaizdo algoritmus;
- įvertinti sukurtos metodikos rezultatus.

1. KOMPIUTERINĖS ERDVINĖS REGOS IR JOS ALGORITMŲ ANALIZĖ.

Šiame skyriuje bus analizuojama ir pateikiama keletą skirtingų technologinių ir matematinių trimatės kompiuterinės regos sprendimų .

1.1. Stereo sistemos modelis.

Kompiuterinėje regoje stereo vaizdo gavimui dažniausiai naudojama sistema sudaryta iš dviejų kamerų, kurios stebi tą patį erdvės dalį. Toks modelis yra pavaizduotas 1.1 paveiksle. Vaizdo



1.1 pav. Dažniausiai naudojamas kompiuterinio erdvinio vaizdo modelis.

kamerų koordinatinių sistemas C1 ir C2 parenkamos taip, kad z ašies neigiama kryptis nurodinėtų objekto atstumą „nuo kameros“. Bet skaičiavimuose galima naudoti koordinatinių modelį, kuris visiškai neįtakos gautų rezultatų.

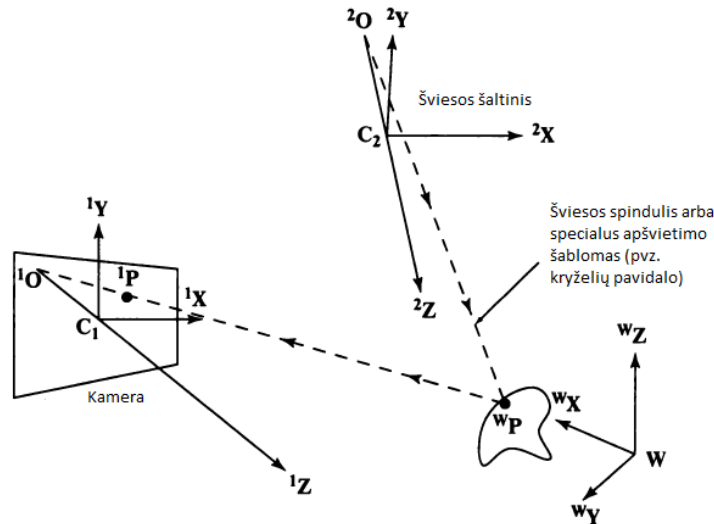
Iš pavyzdžio, kuris pateiktas 1.1 paveiksle matome, kad kameros yra nukreiptos į stebimą erdvės dalį. Ši dalis irgi turi savo koordinatinių sistemą W. Galima teigti, kad taško ${}^W P$ koordinatės yra $({}^W P_x, {}^W P_y, {}^W P_z)$, o jas galima gauti apskaičiavus ${}^W P^1 O$ ir ${}^W P^2 O$ spindulių susikirtimo tašką. Teoriškai šie skaičiavimai yra ganėtinai paprasti, bet praktiškai skaičiavimas yra apsunkinamas paklaidomis ir kitais faktoriais.

Trimačių koordinatinių skaičiavimui mums reikia žinoti šiuos elementus:

- kameros C1 koordinatės W koordinatinių sistemoje. Taip pat reikalingi ir kiti parametrai (pvz. židinio nuotolis), kuriuos galima surašyti matricos pavidale;
- analogiškai skaičiavimams bus reikalingi C2 kameros koordinatės ir jos parametrai;
- reikia nustatyti atitikimą tarp taško ${}^W P$ ir jo atvaizdų ${}^1 P$ ir ${}^2 P$;
- reikalingas matematinis algoritmas/formulė pagal kurią galima paskaičiuoti ${}^W P$ taško koordinatės, turint ${}^W P^1 O$ ir ${}^W P^2 O$ spindulius.

Minėta kompiuterinės regos modelį galima pakeisti į :

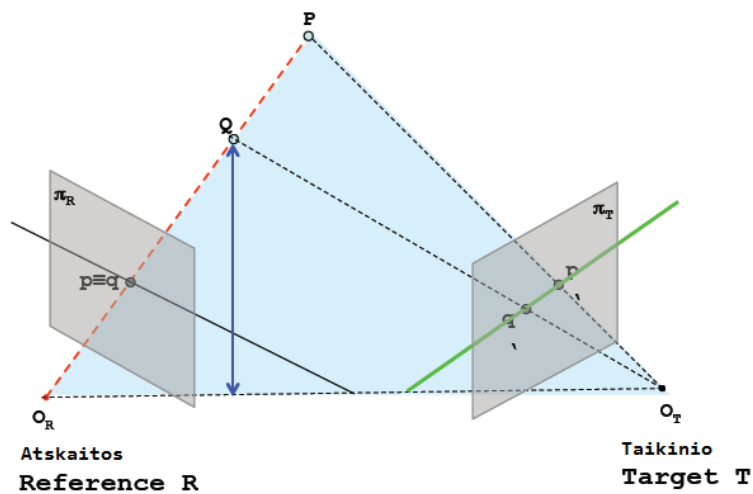
- Sistemą kurioje viena kamera bus pakeista šviesos šaltiniu arba specialiu šviesos šablonu (pvz. kryželių pavidalu). Skaičiavimas bus analogiškas kaip ir dviejų kamerų modelyje. Toks modelis yra pavaizduotas 1.2 paveiksle.



1.2 pav. Kompiuterinės regos modelis sudarytas iš šviesos šaltinio ir vienos kameros.

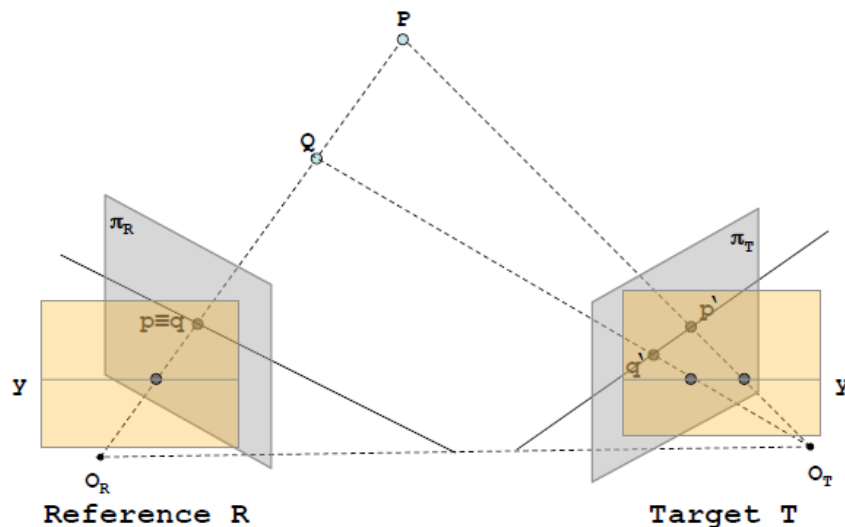
- Sistemą, kai jau žinomi stebimo objekto parametrai. Pvz. jei jau žinomas piramidės, pavaizduotos 1.1 paveiksle, aukštis. Tada jau žinoma viena koordinatė ${}^W P_z$, o tai reiškia kad taškas yra z plokštumoje. Tokia sąlyga labai palengvina skaičiavimus. Realybėje toks modelis labai apriboja kompiuterinės regos panaudojimą.

Kaip 1.1 paveiksle ir buvo pavaizduota dažniausiai naudojama sistema sudaryta iš dviejų vaizdo kamerų. Kiekvienoje kameroje gauname stebimo taško projekciją (${}^1 P$ ir ${}^2 P$ taškai iš 1.1 pav.). Taigi viena kamera galima pavadinti kaip atskaitos (angl. *reference*), o kita taikinio/tikslinė (angl. *target*). Iš minėtų kamerų abiejuose gautuose vaizduose reikia rasti tuos pačius taškus.



1.3 pav. Nekalibruoto erdvinio modelio kamerų plokštumos

1.3 paveiksle matome kad du taškai Q ir P guli ant tos pačios tiesės (raudonai pažymėta). Jų projekcija atskaitos kameros plokštumoje π_R yra tas pats taškas ($P \equiv Q$). Taikinio plokštumoje π_T galima nubraižyti tiesę ant kurios bus išdėstyti Q ir P taškai, priklausantys raudonai tiesei. Pavaizduotame modelyje skaičiavimai bus atliekami dvimatėje erdvėje. Turi būti žinomi plokštumų posūkio kampai ir kiti kintamieji. Šių skaičiavimų palengvinimui galima „standartizuoti“ (atlikti epipoliarinį apribojimą) stereo kamera. Tai reiškia, kad abi plokštumos π_R ir π_T turi sutapti, tai reiškia, kad visa sistema pasidaro vienmatė. Toks atvejis yra pavaizduotas 1.4 paveiksle. Dažniausiai „standartizavimo“ procesas atliekamas virtualiai, nes fiziškai (techniškai) tai pasiekti labai sudėtinga. 1.3 ir 1.4 paveikslų atvejai yra pavaizduoti 1 priede.



1.4 pav. „Standartizuotos“ (kalibruotos) stereo kameros modelis.

1.2. Disparitiškumas (angl. *disparity*) ir vaizdo gylis (angl. *depth*).

Po erdvinės (stereo) sistemos „standartizavimo“ (stereo kalibravimo), kai atliktas epipoliarinis apribojimas (angl. *epipolar constrain*), galima pereiti prie vaizdo gylio skaičiavimo. Pats populiariausias būdas yra trianguliacijos principais pagrįstas skaičiavimas (PQ_RQ_T ir Ppp' trikampiai). Jis yra pavaizduotas 1.5 paveiksle.

$$\frac{B}{Z} = \frac{(b+x_T)-x_R}{Z-f}; \quad (1.1)$$

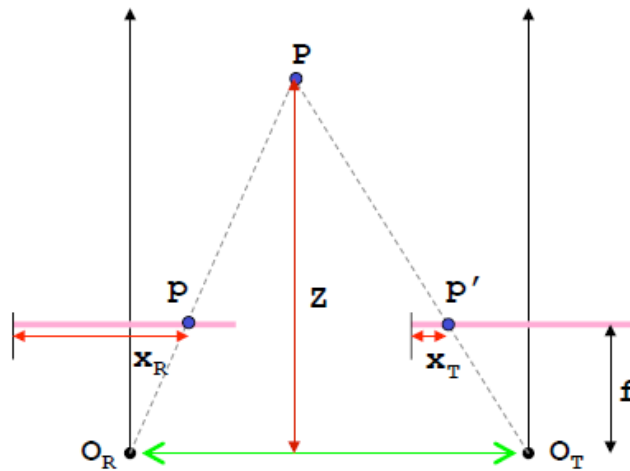
$$Z = \frac{b \cdot f}{x_R - x_T} = \frac{b \cdot f}{d} \quad (1.2)$$

Čia : b – bazinė linija(tiesė);

f – žydinio nuotolis;

$x_R - x_T = d$ - ieškomo taško koordinacių skirtumas.

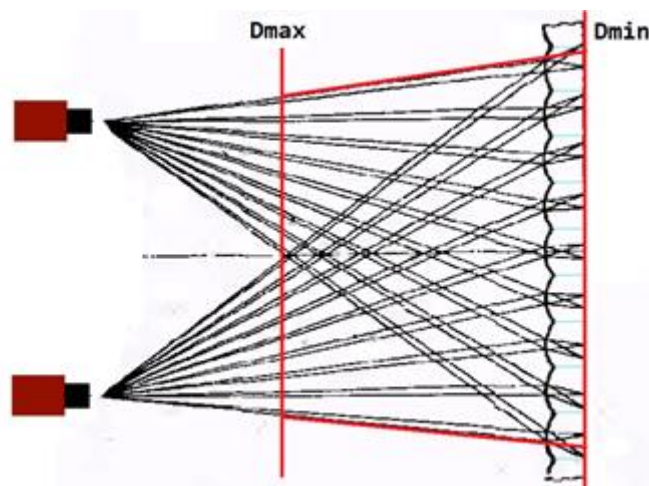
Disparitiškumas apibūdina dviejų atitinkamų taškų skirtumą X koordinatinių sistemoje. Jis dažniausiai atvaizduojamas juodai-pilku paveikslu (angl. *grey image*). Disparitiškumas gautas iš dviejų stereo vaizdų yra pavaizduotas 2 priede. Reikia paminėti tai, kad disparitiškumas yra didesnis tų taškų, kurie yra arčiau kameros. Analogiškai kuo toliau taškas nuo kameros tuo disparitiškumas mažėja .



1.5 pav. Vaizdo gylio skaičiavimo modelis.

1.3. Horopteris (angl. *horopter*).

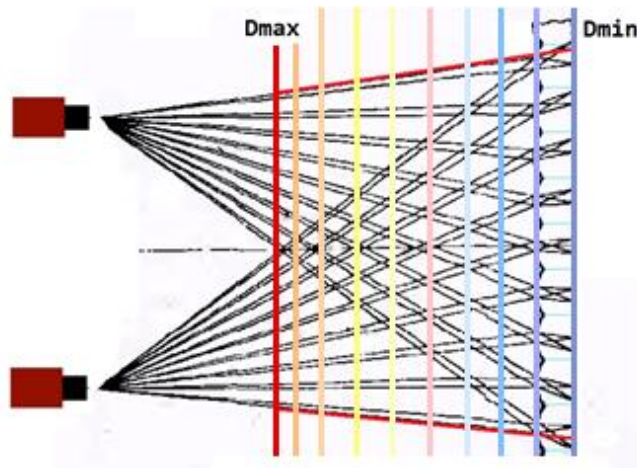
Horopteris – tai regėjimo riba, matoma erdvės dalis. Šis terminas naudojamas stereo sistemos apibūdinimui. Turint bazinę liniją (tiesę) ir židinio nuotolį visus taškus galima apriboti horopteryje, o jo ribas galima apriboti diapozone $[D_{min}, D_{max}]$. Horopteris yra pavaizduotas 1.6 paveiksle. Ribinės vertės galima apskačiuoti panaudojant 1.2 formulę: $\frac{b \cdot f}{D_{min}}$ ir $\frac{b \cdot f}{D_{max}}$.



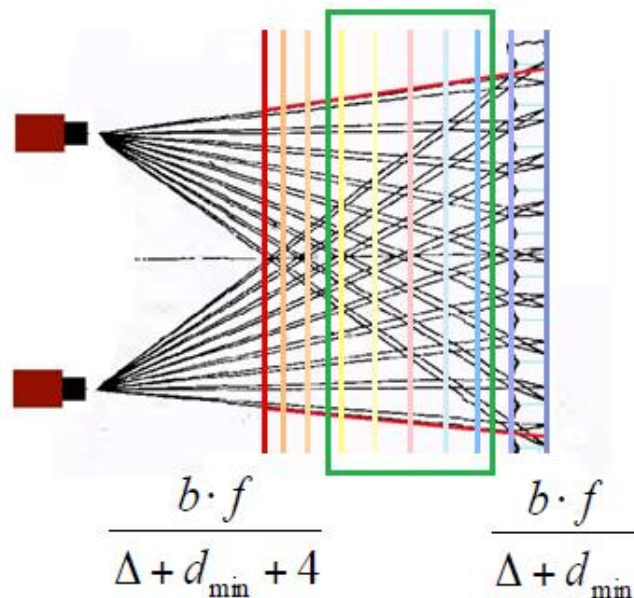
1.6 pav. Horopteris.

Turimą horopterį galima diskretizuoti. Priklausomai nuo atstumo nuo kamerų, jis dalinamas į lygiagrečias plokštumas. Geresnius rezultatus galima gauti diskretizuojant ir pritaikant tarp-piksulinę analizę (angl. sub-pixel estimation). Diskretizuotas horopteris yra pavaizduotas 1.7 paveiksle.

Dažnai būna taip, kad mus domina tik viena tam tikra horopterio dalis, todėl galima keisti diskretizavimo ribas. Pvz. diskretizuojame į penkias plokštumas vidinę horopterio dalį. Gauname kad ribos bus lygios $[\Delta + D_{\min}, \Delta + D_{\min} + 4]$ (1.8 paveikslas).



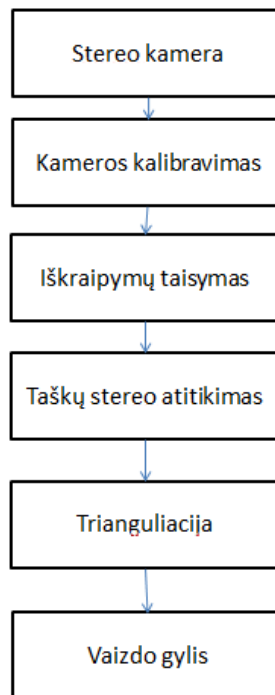
1.7 pav. Diskretizuotas horopteris.



1.8 pav. Apskritas diskretizuotas horopteris.

1.4. Standartinis stereo sistemos modelis.

1.9 paveiksle yra pateiktas struktūrinis standartinės erdvinės sistemos modelis.



1.9 pav. Erdvinės sistemos struktūrinė veikimo schema.

Kalibravimas.

Kalibravimas – tai vidinių iš išorinių kameros parametrų gavimas, panaudojant specialiai gautus vaizdus iš kalibruojamos kameros. Kalibravimas – tai pats pirmas žingsnis, kuris atliekamas prieš bet kokios kompiuterinės regos užduoties sprendimą.

Vidinių kameros parametrų matrica sudaryta iš penkių elementų (1.3 formulė): židinio nuotolis (du parametrai), kameros centras (principinis taškas – angl. *principal point*, du parametrai) ir pikselių posūkio kampo koeficientas (kreivumo koeficientas tarp x ir y ašies).

$$A = \begin{pmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.3)$$

Čia:

f_x ir f_y – židinio nuotolis;

u_0 ir v_0 – kameros centras;

γ – posūkio kampas.

$$\gamma = f_y \cdot \tan \varphi; \quad (1.4)$$

Čia:

φ – pikselių posūkio kampas.

Be tiesinių vidinių kameros parametrų, po kalibravimo gaunami ir netiesiniai parametrai, kurie kitaip vadinami iškraipymo koeficientais. Visi išvardinti parametrai aprašo tik pačią kamerą, bet ne aplinką aplink ją, ir keičiasi tik tada, kai pakeičiami kameros nustatymai.

Išorinius parametrus sudaro R, T vektoriai. R aprašo posūkio parametrus ir gali būti atvaizduotas kaip 3x1 vektorius arba 3x3 matrica. T – postūmio 3x1 vektorius. Jais aplinkos koordinačių sistema pakeičiama į kameros koordinačių sistemą. Atvirkščiai, nei vidiniai parametrai, išoriniai yra susieti su fotografuojama aplinka, todėl kiekvienai nuotraukos porai gali būti skirtingi išoriniai parametrai.

Populiariausi kalibravimo algoritmai:

- Klasikinis – Roger Y. Tsai algoritmas[1]. Susideda iš dviejų etapų. Pirmame etape gaunami išoriniai parametrai, o paskui apskaičiuojami vidiniai.
- „Nauja lanksti kalibravimo technologija“ - ZhengyouZhang sukurtas algoritmas[1]. Jo veikimas yra pagrįstas plokščio, šachmatų lentos pavidalo, objekto naudojimu.
- Autokalibravimas [1] – parametrai gaunami tiesiai iš gautų vaizdų, be specialių kalibravimo objektų naudojimo.

Kalibravimo algoritmai yra įgyvendinti OpenCV bibliotekoje ir MatLab programinėje įrangoje.

Iškraipymų taisymas.

Iškraipymų taisymas susideda iš dviejų etapų:

- Vidinių kameros iškraipymų taisymas (tangentiniai ir radialiniai iškraipymai – 3 priedas)
 - Stereo sistemos „standartizavimas“ (epipoliarinis apribojimas – 1 priedas).

Skirtuminio vaizdo (disparitiškumo) radimas.

Pagrindinė kompiuterinės (mašininės) stereo regos užduotis. Dažniausiai susiveda į atitinkamų taškų radimą skirtingose vaidose, gautų iš stereo sistemos. Galimi algoritmai bus aprašyti žemiau pateiktose punkuose.

Trianguliacija ir vaizdo gylis

Kaip ir buvo parašyta 1.2 skyriuje skirtuminius ir vaizdo gylio žemėlapius (angl. *disparity and depth maps*) galima apskaičiuoti trianguliacijos principu. Skaičiavimui bus reikalingas bazinės linijos ilgis ir židinio nuotolis:

$$Z = \frac{b \cdot f}{d}; \quad (1.5)$$

$$X = Z \cdot \frac{x_R}{f}; \quad (1.6)$$

$$Y = Z \cdot \frac{y_R}{f}; \quad (1.7)$$

1.5. Algoritmai.

Skirtingų algoritmų palyginimui ir klasifikavimui bus naudojami D. Scharstein ir R.Szeliski [3, 3 skyrius 10 psl.] pateikti būdai. Beveik visi standartiniai kompiuterinės erdvinės regos algoritmai susideda iš tam tikrų skaičiavimo žingsnių (angl. *building blocks*). Nors visi jie algoritmų dažna sudedamoji dalis, jų matematinis įgyvendinimas yra skirtingas, priklausomai nuo parinkto skaičiavimo būdo.

Standartiniai skaičiavimo žingsniai:

- 1) taškų atitikimo skaičiavimas (angl. *matching cost computation*);
- 2) atitinkamų taškų surinkimas (agregacija) (angl. *cost (support) aggregation*);
- 3) Disparitiškumo skaičiavimas ir optimizavimas (angl. *disparity computation and optimization*);
- 4) Disparitiškumo patikslinimas (angl. *disparity refinement*).

Faktiškai šių paminėtų žingsnių panaudojimas priklauso nuo konkretaus algoritmo.

Lokaliniuose (angl. *local (window based)*) algoritmuose, kur duoto taško skirtumo skaičiavimas priklauso tik nuo intensyvumo (taško svorio) apribotame lange, kur rezultatai apvalinami sudedant arba atliekant kitus matematinius veiksmus su gautais taškais. Tokiems algoritmams reikalingas tik 1, 2 ir 3 žingsniai, su „laimėtojas gauna viską“ (angl. *winner takes all*) strategija [3, 12psl].

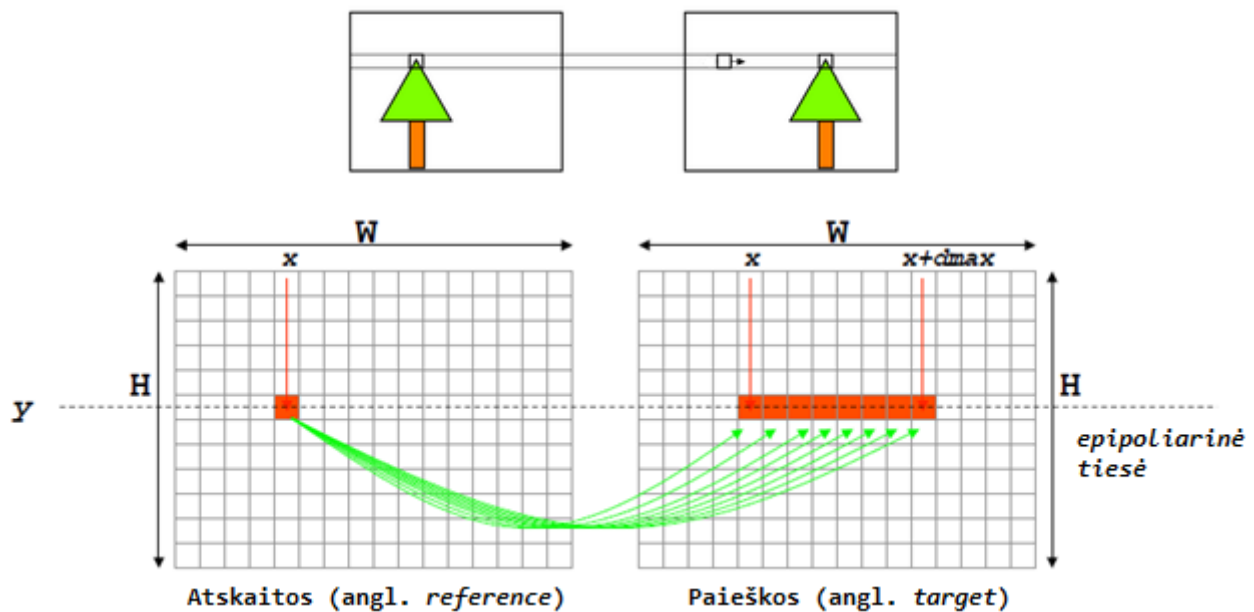
Kita vertus, globalūs (angl. *global*) algoritmai padaro plokštumų paklaidų prielaidas (angl. *make explicit smoothness assumptions*) ir tik tada sprendžia optimizavimo užduotis. Tokie algoritmai neatlieka antro agregacijos žingsnio.

Be minėtų žingsnių taip pat naudojami standartiniai matematiniai skaičiavimo modeliai [3, 15 psl.]:

- 1) Papildomos filtravimo funkcijos (pvz. *Laplacian of Gaussian filtering, Bilateral filtering* ir pan.) ;
- 2) Artimų pikselių vidurkio skaičiavimas;
- 3) Ir kiti papildomi veiksmai galintys įtakoti galutinį rezultatą.

1.5.1. Bazinis lokalinis sprendimas (angl. **Basic approach**).

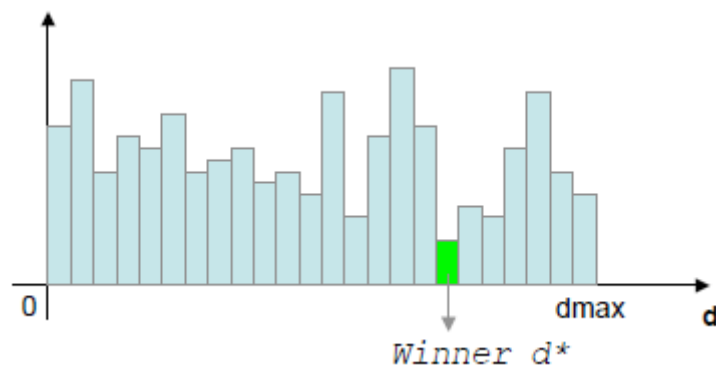
Pats paprasčiausias algoritmas yra pagrįstas atitinkamų taškų paieška epipoliarinėje tiesėje. Kaip aukščiau jau buvo parašyta, erdvinė sistema turi būti „standartizuota“, tai reiškia kad turi būti atliktas epipoliarinis apribojimas. Po šio veiksmo dviejose stereo vaizdo epipoliarinės tiesės sutampa. Reiškia visą skaičiavimą galima atlikti vienoje tiesėje (1.10 pav.) Galima panaudoti neuronų tinkluose naudojama algoritmą „Nugalėtojas gauna viską“ (angl. *Winner takes all*).



1.10 pav. Atitinkamų taškų paieška epipoliarinėje tiesėje.

Skaičiavimas susideda iš dviejų etapų:

- 1) Skaičiuojamas pasirinkto ploto intensyvumas (1.10 pav. atskaitos vaizdas) ir skirtumas (1.10 pav. paieškos vaizdas) – 1 žingsnis (1.5 skyrius);
- 2) Reikiamo taško koordinatės parinkimas (1.10 pav. paieškos vaizdas) panaudojant „winner takes all“ algoritmą (1.11 pav.) – 3 žingsnis (1.5 skyrius).



1.11 pav. „Winner takes all“ algoritmo atvaizdavimas.

1.5.2. Bazinio lokalinio sprendimo patobulinimas.

Egzistuoja dvi tobulinimo strategijos (visai atvejais daroma prielaida, kad ir pirmu ir antru atveju stereo vaizdas dalinai nufiltruojamas):

- Lokaliniai algoritmai – naudojama paprasta „winner takes all“ disparitiškumo skaičiavimo strategija su papildomais filtrais arba skaičiavimo algoritmais leidžiančiais padidinti viso skaičiavimo atsparumą triukšmams (angl. SNR – *signal to noise ratio*) (1, 2 žingsniai ir „winner takes all“). Tokio skaičiavimo trūkumas yra tai, kad vienas taškas pažymėtas atskaitos paveiksle

(angl. *reference image*) gali atitikti keliems taškams taikinio (angl. *target*) paveiksle, turinčius tokius pat parametrus.

- Globalūs algoritmai – priešingi lokaliniais algoritmams, globaliuose praleidžiamas agregacijos žingsnis. Dažniausiai jie yra pagrįsti energijos minimizavimo metodais. Tikslas yra rasti disparitiškumo funkciją d kuri minimizuoja „globalią“ energiją:

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d) \quad (1.8)$$

Funkcija $E_{data}(d)$ parodo kaip gerai disparitiškumo funkcija atitinka stereo porai. $E_{smooth}(d)$ funkcija parodo parinkto algoritmo filtravimo (angl. *smoothness*) lygį. [6]

1.5.3. Atitikimo skaičiavimo (angl. *matching cost computation*) algoritmai.

Pikselių pagrįstas skaičiavimai (angl. *Pixel-based matching costs*):

- Absoliutus skirtumas :

$$e(x, y, d) = |I_R(x, y) - I_T(x + d, y)| \quad (1.9)$$

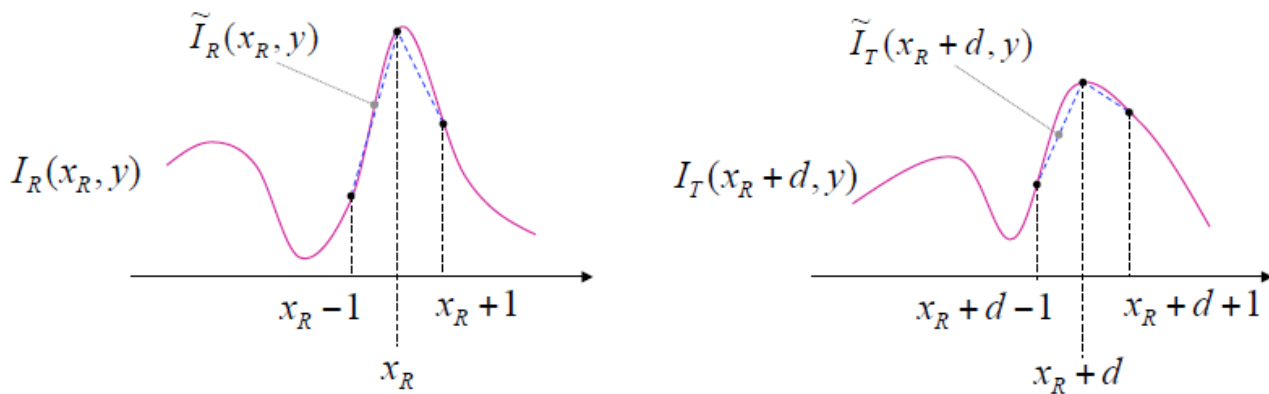
- Kvadratų skirtumas:

$$e(x, y, d) = (I_R(x, y) - I_T(x + d, y))^2 \quad (1.10)$$

- Tvirtos atitikimo priemonės (angl. *Robust matching measures (M-estimators)*):

Pvz. sutrumpintas absoliutus minimumas (angl. *Truncated absolute differences (TAD)*):

$$e(x, y, d) = \min\{|I_R(x, y) - I_T(x + d, y)|, T\} \quad (1.11)$$



1.12 pav. Stereo vaizdų pikselių „svorių“ skirtumas.

Fiksuoto dydžio slenkančiu langu pagrįsti skaičiavimai (1.13 pav.) (angl. *area-based matching costs*):

- Absoliučių skirtumų suma (*SAD* – *Sum of Absolut edifferences*):

$$C(x, y, d) = \sum |I_R(x, y) - I_T(x + d, y)| \quad (1.12)$$

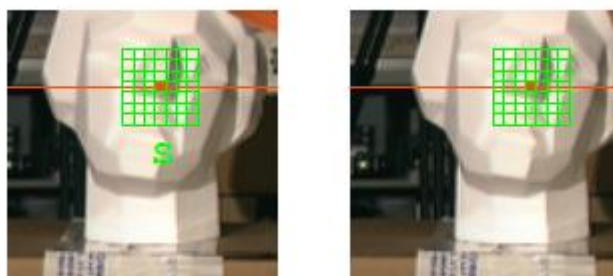
- Absoliučių skirtumų kvadratų suma (*SSAD* – *Sum of Squared differences*):

$$C(x, y, d) = \sum (I_R(x, y) - I_T(x + d, y))^2 \quad (1.13)$$

- Sum of truncated absolute differences (*STAD* – *Sum of truncated absolute differences*):

$$C(x, y, d) = \sum \min \{|I_R(x, y) - I_T(x + d, y)|, T\} \quad (1.14)$$

- Normalized cross correlation;
- Zero mean normalized cross coreliation.

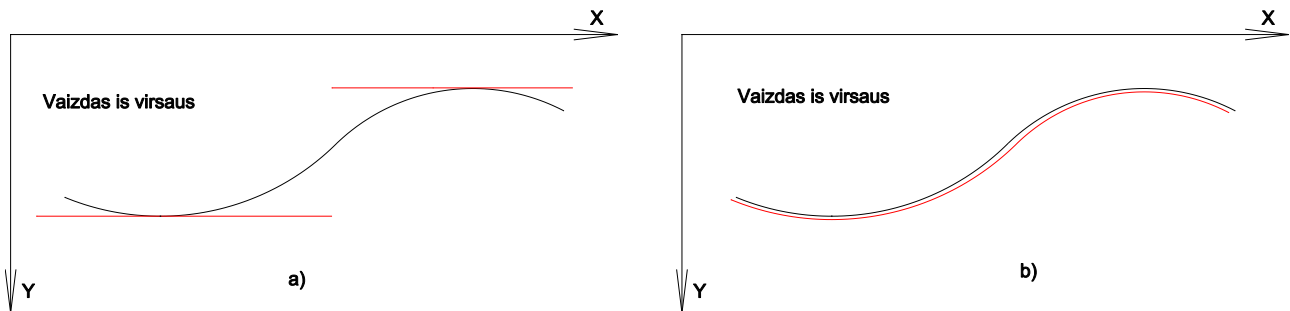


1.13 pav. Fiksuoto lango skaičiavimas epipoliarinėje tiesėje.

1.5.4. Verčių sumavimas (angl. *cost aggregation*).

Po atitikimo skaičiavimo atliekamas rezultatų surinkimas į bendrą „vaizdą“. Naudojant fiksuoto lango algoritmus aptinkamos šios klaidos:

1. Neryškiai aptinkamos lygiagrečios erdvinio vaizdo sritys. Dažnai pasitaikanti problema bandant išmatuoti apvalią arba pasukta kampu figūrą. Toks atvejis yra pavaizduotas 1.14 paveiksle.



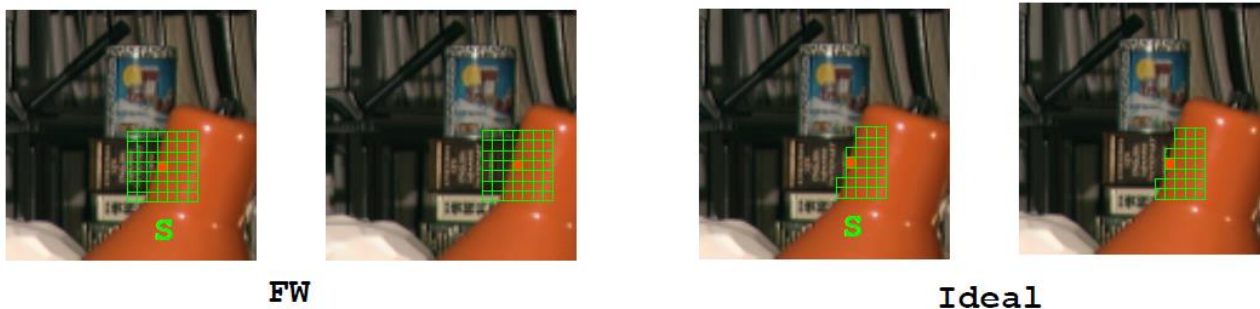
1.14 pav. a) fiksuoto lango algoritmo skaičiavimo rezultatas; b) idealus/etaloninis rezultatas.

2. Netiksliai aptinkami vaizdo gylio ribiniai taškai (perėjimai nuo vieno dydžio prie kito). Dėl fiksuoto lango dydžio ir jos parametrų nelankstumo, perėjimo sritys, aptinkamos netiksliai. Toks atvejis pavaizduotas 1.15 paveiksle. Panaudojant statistikos apdorojimo metodus galima dalinai sumažinti paklaidos dydį.

Taip pat tikslui pasiekti galima sumažinant papildomų laukų kiekį, t.y. į skaičiavimą įtraukti tik tas sritis kurios turi tą patį dispariškumą arba vaizdo gylį.



1.15 pav. Netikslumai gaunami prie „perėjimo“ sričių.



1.16 pav. Netikslumų pašalinimas, mažinant papildomų laukų kiekį.

- a) Netiksliai aptinkamos pasikartojančios ir vienodos sritys. Šiuo atveju galima taikyti tokius pat algoritmus, kurie buvo aprašyti 2 punkte. Vienintelis skirtumas gali būti toks, kad papildomų laukų kiekį galima ne tik mažinti bet ir didinti.

1.6. Standartiniai stereo vaizdų rinkiniai skirti algoritmų testavimui.

Kiekvienais metais yra sukuriama vis nauji erdvinės regos algoritmai. Kiekvienos naujos metodikos tikslas yra kuo tiksliau ir greičiau apskaičiuoti disparitiškumą. Dažnai algoritmai apdoroja ne tik atskirus stereo vaizdų rinkinius, bet ir realiaame laike gaunamą informaciją. Todėl jų išbandymams reikalingi kalibruoti stereo porų rinkiniai, o gautiems rezultatams įvertinti reikalingas atraminis disparitiškumas (angl. *groundtruth*). Šiai užduočiai išspręsti kuriami standartiniai erdvinės regos stereo vaizdų porų rinkiniai turintys atraminį disparitiškumą.

Vienas iš populiariausių rinkinių pasaulyje yra Tsukuba. Pagrindiniai elementai pavaizduoti rinkinyje yra galva ir lempa. Šis rinkinys buvo sukurtas Japonijos Tsukuba universitete. Dažniausiai šis rinkinys yra sintetinis (sukurtas erdvinio modeliavimo programų pagalba), kas leidžia keisti vaizdų parametrus : apšvietimą, scenos apžiūros kampą, vaizdų dydį, taip pat galima simuliuoti realaus laiko veikimo modelį. Vienas iš tsukuba stereo porų rinkinių yra pavaizduoti 4 priede [7].

Be minėto rinkinio, yra ir kiti populiarius tarp tyrinėtojų vaizdai: Teddy, Venus, Cones ir kiti. Kiekvienas rinkinys išsiskiria skirtingomis ypatybėmis:

- objektų spalvos panašios arba monotoniškos;
- skirtingi objektai vaizduose išdėstyti skirtingais kampais, tai pasunkina jų kontūrų radimą;
- objektų formos yra skirtingos.

Visų šių stereo porų autoriai yra Daniel Scharstein, Padma Ugbabe ir Rick Szeliski. Pilną aprašymą ir atraminius disparitiškumus galima rasti [16] šaltinyje

2. EKSPERIMENTINIS TYRIMAS

2.1 Bazinis lokalinis sprendimas (angl. *basic approach*).

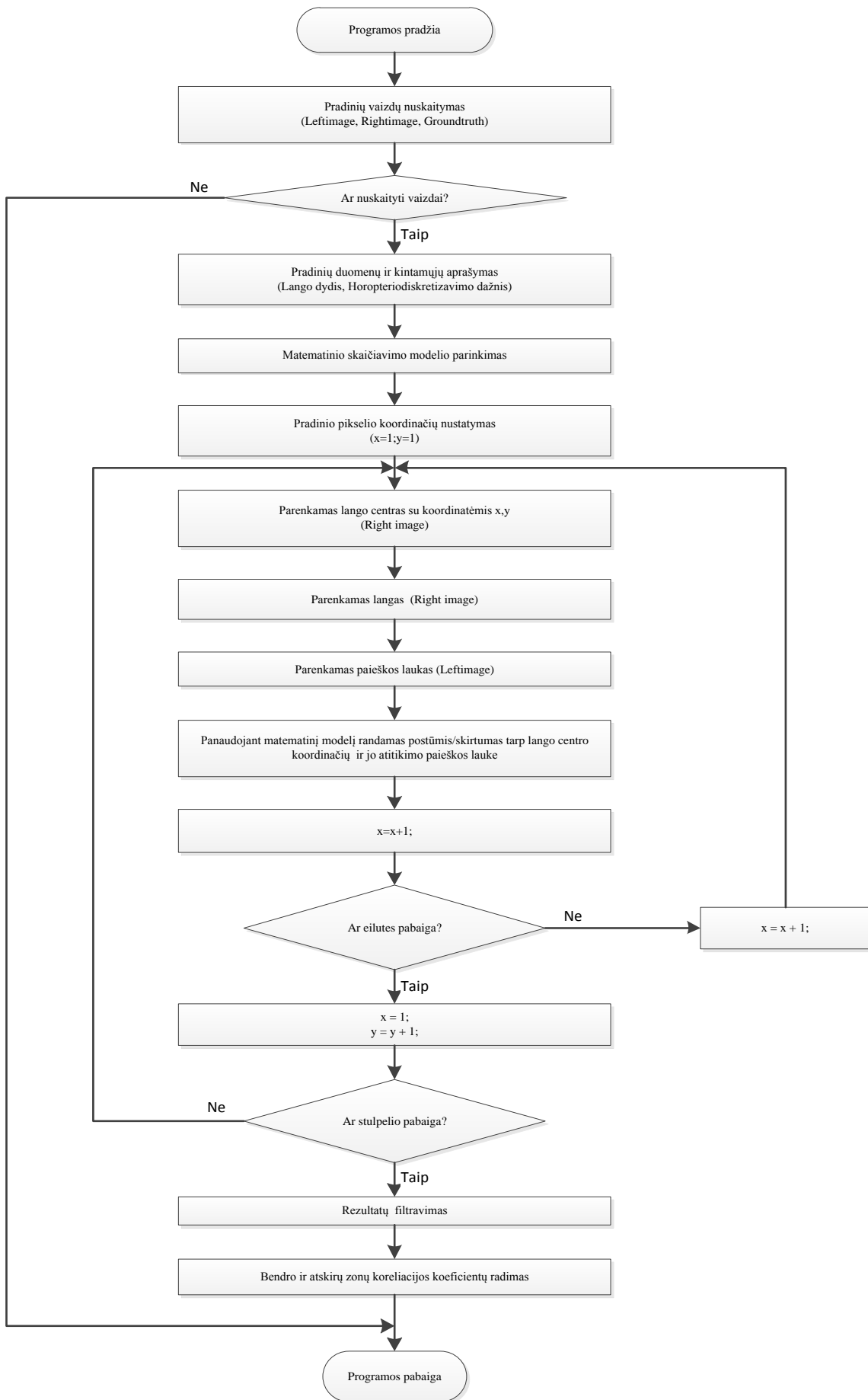
2.1.1 Bazinio lokalinio sprendimo algoritmo aprašymas.

Pirmas algoritmas kuris bus tiriamas yra bazinis lokalinis algoritmas. Jis yra pavaizduotas 2.1 paveiksle.

Šitam algoritmui ištirti panaudota „Tsukuba“ stereo porų rinkinį. Taip pat rezultatams gauti reikalingas atraminis disparitiškumas (angl. *groundtruth*).

Prieš paleidžiant programą reikia ją sukonfigūruoti, parenkant keturis svarbius parametrus:

1. Teisingai parinkti ir epipoliariškai apriboti stereo porų vaizdai;
2. disparitiškumo gylis – horopterioidiskretizavimo dažnis, t.y. į kiek sluoksnių bus išdalintas horopterioidis;
3. slenkančio lango dydis pikseliais;
4. naudojamas matematinis modelis (SAD, SSD, STAD – 1.5.3. skyrius).



2.1 pav. Basic approach algoritmas.

Po aukščiau išvardintų parametrų parinkimo paleidžiama programa „Run and time“ funkcija. Ši funkcija visiškai nesiskiria nuo įprastos paleidimo funkcijos „Run“, tik leidžia išmatuoti laiką, kurį sunaudojo kompiuteris atlikdamas pateiktą užduotį. Atlikimo laikas yra svarbus algoritmo įvertinimo kriterijus.

Programos pradžioje visi naudojami vaizdai iš spalvotų paverčiami į juodai–pilkus (angl. *gray picture*). Kairysis yra atskaitos (angl. *reference*) vaizdas, o dešinysis paieškos (angl. *target*) vaizdas. Po šių veiksmų atliekamas lango išskyrimas iš „atskaitos“ (angl. *reference*) vaizdo ir jo atitikimo paieška „paieškos“ (angl. *target*) paveiksle. Pagrindinė slenkančio lango charakteristika yra lango centro koordinatės, todėl lango dydžio parinkimui naudojama „puse lango dydžio“ charakteristika. O pats langas apskaičiuojamas pagal šia formulę:

$$W_x = 2 * half_block_size + 1; \quad (2.1)$$

$$W_y = 2 * half_block_size + 1; \quad (2.2)$$

Kaip buvo minėta, naudojami vaizdai turi būti epipoliariškai apriboti, todėl visi veiksmai (skaičiavimas ir atitikimo paieška) atliekami toje pačioje x ašyje. Reikia atkreipti dėmesį į slenkančio lango dydžio parinkimą vaizdo kraštinėse (kai lango centras yra kraštinėje). Tokiais atvejais langas sumažėja per pusę ir į jį įeina tik ta dalis vaizdo, kuri egzistuoja.

Atitikimo paieška atliekama ne visame vaizde, o tik atskirame lauke. Kaip ir slenkantis langas taip ir paieškos laukas yra toje pačioje x ašyje. Lauko dydis ir koordinatės priklauso nuo dviejų parametrų:

- 1) slenkančio lango koordinatės;
- 2) disparitiškumo gylio.

Atitikmens paieškai naudojami 1.5.3 skyriuje aprašyti algoritmai. Kai gaunamos lango atitikmens koordinatės, apskaičiuojamas disparitiškumas. Paimamos lango koordinatės atskaitos vaizde (angl. *reference picture*) ir jo atitikimo koordinatės paieškos vaizde (angl. *target picture*) ir skaičiuojamas postūmis pikseliais. Gauti rezultatai surašomi į atskirą masyvą, kuris algoritmo pabaigoje atvaizduoja disparitiškumo žemėlapi.

Po šių veiksmų paieškos lango koordinatės keičiamos vienu pikselių pagal x arba y ašį ir išnaujo atliekamas aukščiau aprašytas algoritmas.

Rezultatų įvertinimui bus naudojami koreliacijos koeficientai. Juos galima išskirti į dvi grupės: bendras ir tiksliniai koreliacijos koeficientai. Bendras koeficientas parodo kaip algoritmo apskaičiuotas disparitiškumas koreliuoja su atraminiu vaizdu, bet jis negali tikslingai aprašyti algoritmo kokybę, todėl bus išskirtos kelios svarbios vaizdo zonos, kurių koreliacija su atraminiu disparitiškumo vaizdu (angl. *groundtruth*) bus skaičiuojama atskirai. Taip pat įvertinime yra svarbus

algoritmo atlikimo laikas, kuris parodo kaip greitai vykdomas testuojamas algoritmas.

Algoritmas įgyvendintas naudojant MatLab 2013b programą. Programos tekstas yra pateiktas 5 priede.

2.1.2 Bazinio lokalinio sprendimo (angl. *basic approach*) algoritmo tyrimo rezultatai.

Kaip buvo minėta šio basicapproach algoritmo tyrimas yra atliktas MatLab 2013b programinio paketo pagalba. Programos tekstas yra pateiktas 5 priede. Tyrimas buvo atliktas ACER ASPIRE V3-551G kompiuteriu (CPU: AMD A8-4500M 2,8 GHz; GPU: AMD Radeon HD 7640G). Išvardinti parametrai yra svarbūs, nes bus skaičiuojamas algoritmo veikimo laikas, o jis tiesiogiai priklauso nuo kompiuterio pajėgumo ir naudojamų technologijų. Taip atlikimo laikas yra svarbus projektuojant realaus laiko erdvines regos sistemas, turint ribotus išteklius.

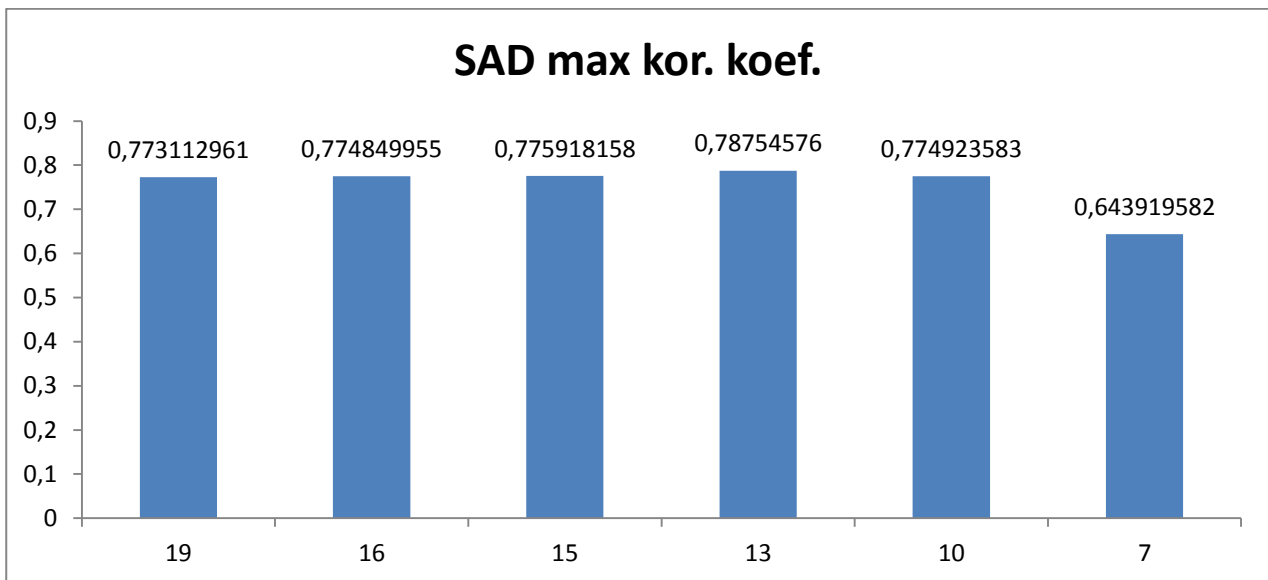
Pirmiausia prieš atliekant tyrimą, reikėjo pasirinkti kalibruotos (epipoliariškai apribotos) stereo vaizdų poros. Taip pat rezultatams įvertinti reikalingas „atraminis“ disparitiškumas. Todėl buvo parinktas dažnai naudojamas tyrimuose „Tsukuba“ rinkinys. Šis rinkinys buvo parinktas dėl kelių priežasčių:

- Vaizdų dydis yra 288x384 pikselio. Tai yra mažos rezoliucijos vaizdai, todėl skaičiavimo atlikimo laikas yra mažesnis ir mažiau apkraunamas kompiuteris.
- Dažnai naudojamas tyrimuose, todėl galima palyginti gautus rezultatus su kitų žmonių rezultatais.
- Lengvai randamas ir prieinamas „atraminio“ disparitiškumo vaizdas, nes daugelis internete randamų stereo porų vaizdų neturi paruošto „atraminio“ disparitiškumo.

Po stereo porų parinkimo reikėjo parinkti matematinį modelį. Pirmiausias buvo parinktas SAD (angl. sum of absolute differences) modelis. Jo įvertinimui ir tyrimui reikėjo atlikti daug bandymų, keičiant pradinius parametrus (slenkančio lango dydis ir horoptero diskretizavimo dažnis). Eksperimentiniu būdu buvo nustatyta, kad puse slenkančio lango dydis (pilnas langas apskaičiuojamas pagal 14 ir 15 formulės) turi būti ribose nuo 1 iki 15 pikselių, o horoptero diskretizavimas turi kisti nuo 7 iki 19 sluoksnių. Nepatenkantys į aprašytas ribas parametrai netinka eksperimentams su „tsukuba“ (288x384 pikselio dydžio), nes gauti rezultatai turi mažus koreliacijos koeficientus ir negali įtakoti tyrimui. Kiekvienam parametru rinkiniui išbandyti reikia nuo 30 iki 90 sekundžių. Tai užima daug laiko, todėl programa buvo pakeista taip, kad vienu programos paleidimui buvo atliekami iškart keli eksperimentai. Buvo parenkamas diskretizavimo dažnis, o lango dydis, po kiekvieno bandymo, automatiškai buvo keičiamas. Visi gauti duomenys buvo įrašomi į masyvą ir toliau apdorojami.

Gauti bandymų rezultatai yra pateikti 6 priede. Pradiniai rezultatai gauti taikant tik matematinį apdorojimą be tolimesnio filtravimo pritaikymo. Iš gautų rezultatų matome prie kokių

parametrų buvo pasiekti maksimalus koreliacijos koeficientai. Taip pat vadovaujantis bandymo rezultatais galima patvirtinti nustatytas parametrų ribas (parenkamus bandymų mažiausius ir didžiausius slenkančio lango ir disparitiškumo dydžius). Dydžiausių gautų bendrų koreliacijos koeficientų, be filtravimo pritaikymo, priklausomybė nuo parinkto disparitiškumo yra pavaizduota 2.2 paveiksle. Iš jo matome kad koreliacijos koeficientai mažėja kai disparitiškumas yra mažesnis nei 7 ir kai didesnis nei 19. Todėl ir buvo parinktos tokios ribos.



2.2 pav. Maksimalus SAD koreliacijos koeficientai be filtravimo.

Geriausi SAD rezultatai be filtravimo yra pateikti 2.1 lentelėje. Apibendrinant pateiktus duomenis, bendro koreliacijos koeficiento priklausomybė nuo parinkto lango dydžio ir disparitiškumo yra pavaizduota 2.3 paveiksle, o algoritmo veikimo laiko priklausomybė nuo disparitiškumo ir lango dydžio 2.4 paveiksle.

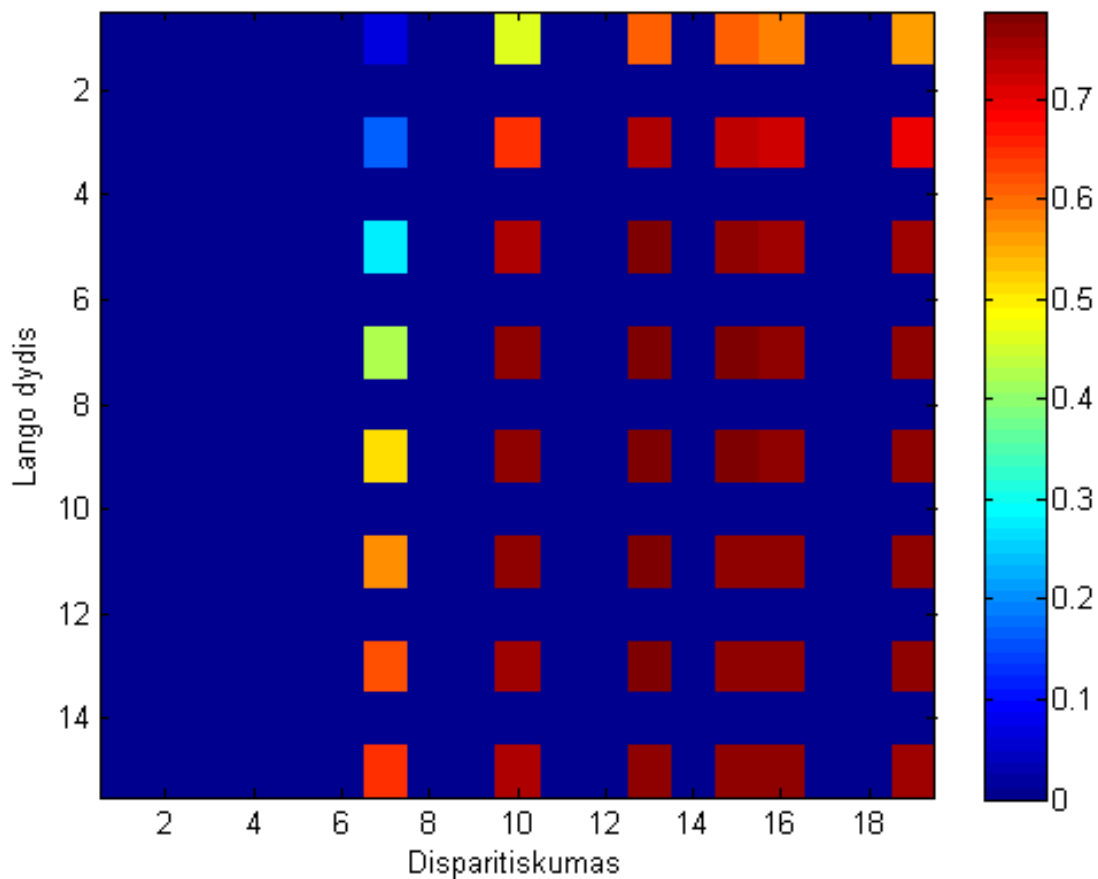
2.2 lentelė

Geriausi SAD be filtravimo bandymų rezultatai.

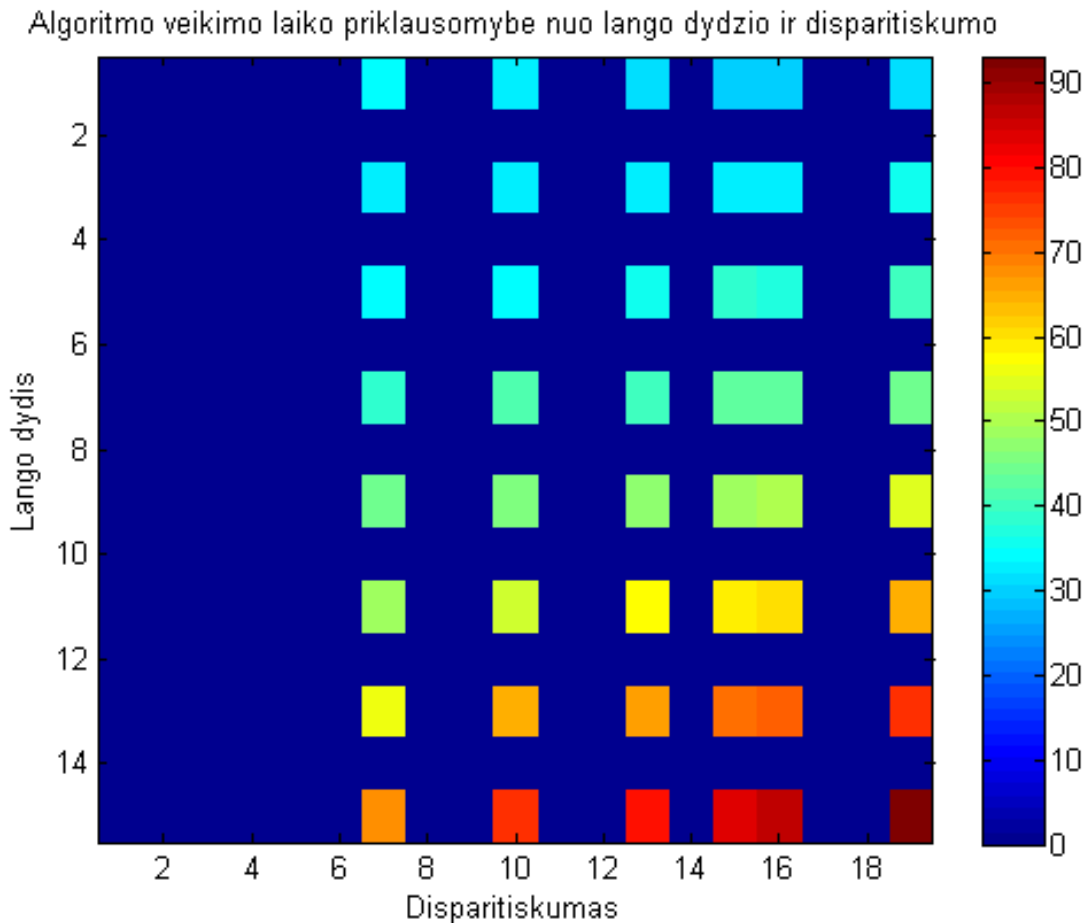
Nr.	Puse lango dydis	Disparitiškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor.koef.	Kameros kor.koef.	Stalo kor.koef.	Galvos kor.koef.	Lempos kor.koef.
1	7	13	40,4510593	0,78754576	0,186616	0,580917	0,756575	0,55607	0,569598
2	9	13	47,7987064	0,784653183	0,189808	0,597023	0,727358	0,545149	0,574356
3	5	13	36,2858326	0,784333638	0,169706	0,549092	0,750894	0,552752	0,55723
4	7	15	42,3074712	0,775918158	0,186616	0,501567	0,755894	0,535167	0,566083
5	9	15	48,4383105	0,775322707	0,189808	0,597023	0,727358	0,525337	0,566799
6	7	10	41,0906634	0,774923583	0,186616	0,580917	0,760516	0,407434	0,562605
7	9	16	50,7159251	0,774849955	0,189808	0,597023	0,727358	0,525267	0,566783

8	7	16	42,744274	0,774133298	0,186616	0,501567	0,755894	0,535167	0,565853
9	9	19	54,0699466	0,773112961	0,189808	0,597023	0,727358	0,525267	0,566783
10	7	19	43,9766819	0,770036907	0,186616	0,501567	0,755894	0,535167	0,566066
11	9	10	46,2074962	0,76982997	0,189808	0,597023	0,728196	0,389368	0,566743
12	5	15	38,5166469	0,766997874	0,169706	0,432046	0,749583	0,529628	0,551399
13	5	16	36,9098366	0,761149443	0,169706	0,432046	0,749583	0,52831	0,550496
14	5	19	39,936256	0,753563532	0,169706	0,432046	0,740812	0,52831	0,555385
15	5	10	33,6026154	0,741548275	0,169706	0,549092	0,756396	0,398039	0,44568
16	15	7	67,6420336	0,643919582	0,121209	0,592693	0,526272	0,148891	0,28183
17	13	7	56,4255617	0,625285212	0,108673	0,599238	0,521856	0,084159	0,285217
18	11	7	48,7191123	0,571260887	0,162324	0,604513	0,523417	0,011947	0,251828

Bendro koreliacijos koeficiento priklausomybė nuo disparitiskumo ir lango dydžio



2.3 pav. Bendrojo koreliacijos koeficiento priklausomybė nuo parinkto lango dydžio ir disparitiškumo.



2.4 pav. Algorithmo veikimo laiko priklausomybe nuo lango dydžio ir disparitiškumo.

2.1 lentelėje rezultatai išdėstyti pagal bendrąjį koreliacijos koeficientą mažėjančia tvarka. Matome kad nuo pirmojo iki penkioliktojo bandymo rezultato koreliacijos koeficientų skirtumas yra labai mažas. Todėl įvertinime tampa svarbus algoritmo atlikimo laikas. Taip pat papildomam įvertinimui įvesti atskirų vaizdo objektų koreliacijos koeficientai, bet jie tik papildo pagrindinius rezultatus.

Įvertinus visus aukščiau pateiktus duomenis ir įvertinimo kriterijus išrinkti penki geriausi SAD rezultatai, jie yra pateikti 2.2 lentelėje.

2.2 lentelė

Geriausi, pagal algoritmo veikimo laiką, SAD be filtravimo bandymų rezultatai.

Nr.	Puse lango dydis	Disparitiškumas	Algorithmo veikimo laikas	Bendras kor. koef.	Fono kor.koef.	Kameros kor.koef.	Stalo kor.koef.	Galvos kor.koef.	Lempos kor.koef.
1	5	10	33,6026154	0,741548	0,169706	0,549092	0,756396	0,398039	0,44568
2	5	13	36,2858326	0,784334	0,169706	0,549092	0,750894	0,552752	0,55723
3	5	16	36,9098366	0,761149	0,169706	0,432046	0,749583	0,52831	0,550496
4	5	15	38,5166469	0,766998	0,169706	0,432046	0,749583	0,529628	0,551399

5	5	19	39,936256	0,753564	0,169706	0,432046	0,740812	0,52831	0,555385
6	7	13	40,4510593	0,787546	0,186616	0,580917	0,756575	0,55607	0,569598
7	7	10	41,0906634	0,774924	0,186616	0,580917	0,760516	0,407434	0,562605
8	7	15	42,3074712	0,775918	0,186616	0,501567	0,755894	0,535167	0,566083
9	7	16	42,744274	0,774133	0,186616	0,501567	0,755894	0,535167	0,565853
10	7	19	43,9766819	0,770037	0,186616	0,501567	0,755894	0,535167	0,566066
11	9	10	46,2074962	0,76983	0,189808	0,597023	0,728196	0,389368	0,566743
12	9	13	47,7987064	0,784653	0,189808	0,597023	0,727358	0,545149	0,574356
13	9	15	48,4383105	0,775323	0,189808	0,597023	0,727358	0,525337	0,566799
14	9	16	50,7159251	0,77485	0,189808	0,597023	0,727358	0,525267	0,566783
15	9	19	54,0699466	0,773113	0,189808	0,597023	0,727358	0,525267	0,566783

Vizualiai įvertinus gautus rezultatus galima pastebėti, kad papildomai prie SAD galima pritaikyti filtravimo funkcijas. Triukšmai ypač pastebimi, kai parenkamas mažas slenkantis langas. Tokie gauti rezultatai pavaizduoti 7 priede. Taip pat reikia pabrėžti kad triukšmų lygis labai įtakoja bendrąjį koreliacijos koeficientą. Todėl visi bandymai atlikti su mažais slenkančiais langais turi mažesnę bendrą koreliaciją nei kitose bandymuose, nors ir vizualiai atrodo geriau.

Rezultatų filtravimui buvo panaudotos šios funkcijos:

- Medianos filtras (MatLab funkcija: medfilt2);
- Vinerio (Wiener) filtras (MatLab funkcija: wiener2).

Laiko taupymo sumetimais filtravimo funkcijos buvo išbandytos tik su geriausiai bandymų rezultatais be filtravimo (2.2 lentelė). Pirmiausiai disparitiškumas buvo filtruojamas tik medianos filtrų, o vėliau buvo pritaikytas dar Vinerio filtras. 2.3 lentelėje yra pateikti eksperimento rezultatai po medianos filtravimo. Patogesniai palyginimui duomenys 2.3 lentelėje išdėstyti tokia pat tvarka kaip ir 2.2 lentelėje. Palyginimo patogumui 2.4 lentelėje pateikti bendrų koreliacijos koeficientų ir algoritmo veikimo laikų skirtumas. Taip pat yra apskaičiuotas rezultatų vidurkis.

2.3 lentelė

SAD algoritmo rezultatai po medianos filtravimo

Nr.	Puse lango dydis	Disparitiškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor. koef.	Kameros kor. koef.	Stalo kor. koef.	Galvos kor. koef.	Lempos kor. koef.
1	5	10	35,4902275	0,749763	0,176116	0,560527	0,765753	0,402318	0,458974
2	5	13	39,0782505	0,790104	0,176116	0,560527	0,760903	0,557737	0,56135
3	5	16	38,5634472	0,76735	0,176116	0,444963	0,759739	0,536849	0,553585
4	5	15	40,716261	0,773178	0,176116	0,444963	0,759739	0,537151	0,554318
5	5	19	41,6990673	0,762296	0,176116	0,444963	0,759739	0,536849	0,559789
6	7	13	42,1514702	0,789678	0,189421	0,592078	0,763345	0,558166	0,570071
7	7	10	39,0626504	0,779014	0,189421	0,592078	0,766961	0,412476	0,570046

8	7	15	43,4930788	0,778392	0,189421	0,524517	0,762635	0,537144	0,56611
9	7	16	47,8767069	0,776719	0,189421	0,524517	0,762635	0,537144	0,566052
10	7	19	45,8798941	0,773365	0,189421	0,524517	0,762635	0,537144	0,566029
11	9	10	45,396291	0,77098	0,194235	0,603009	0,734823	0,391407	0,568174
12	9	13	50,1387214	0,785748	0,194235	0,603009	0,734268	0,545565	0,574523
13	9	15	49,9047199	0,776568	0,194235	0,603009	0,734101	0,525569	0,566838
14	9	16	52,260335	0,776477	0,194235	0,603009	0,734101	0,525569	0,566838
15	9	19	51,1527279	0,774752	0,194235	0,603009	0,734101	0,525569	0,566838

2.4 lentelė

SAD algoritmo be filtravimo ir po medianos filtravimo.

Nr.	Puse lango dydis	Disparitiškumas	Koreliacijos koeficientų skirtumas	Algoritmo atlikimo laiko skirtumas [sekundėmis]
1	5	10	0,008214651	1,887612
2	5	13	0,00577006	2,792418
3	5	16	0,006200496	1,653611
4	5	15	0,006180058	2,199614
5	5	19	0,008732232	1,762811
6	7	13	0,002132409	1,700411
7	7	10	0,004090352	-2,02801
8	7	15	0,002474033	1,185608
9	7	16	0,002585593	5,132433
10	7	19	0,003328397	1,903212
11	9	10	0,001150137	-0,81121
12	9	13	0,001094903	2,340015
13	9	15	0,001245599	1,466409
14	9	16	0,001626763	1,54441
15	9	19	0,001638773	-2,91722
Vidurkis			0,003764297	1,320808

Matome, kad medianos filtras visiškai neįtakoja rezultatų. Bendras koreliacijos koeficientas padidėja tik tūkstantosiomis dalimis. Vidutiniškai filtravimas užima apie 1,32 sekundes. Išsamesni rezultatai pateikti 6 priede.

Sekantis filtras kuris buvo išbandytas buvo Vinerio. Tai yra žemo dažnio filtras, kuris dažnai naudojamas pilkų (angl. *grey scale*) paveikslų filtravimui. Vinerio filtravimo rezultatai pateikti 2.5 lentelėje (greičiausiai veikiančys viršuje), o jų palyginimas su duomenimis be filtravimo 2.6 lentelėje. Matome, kad pavyko pagerinti bendrą koreliacijos koeficientą. Vidutiniškai koeficientas pagerėjo 3 procentais ir kai kurių rezultatų bendrojo koreliacijos koeficiento dydis pasiekė 0,8 ribas. Statistikoje 0,8 dydžio koeficientas laikomas kaip geros koreliacijos požymis. Aišku filtravimo atlikimas reikalauja kompiuterio resursų panaudojimo, todėl pailgėja algoritmo atlikimo laikas. Vidutiniškai tai sudaro 1,38 sekundes. Palyginus su viso algoritmo atlikimo laiku tai labai

maža dalis viso sunaudoto laiko, todėl norint pasiekti geresnius SAD algoritmo rezultatus reikia naudoti Vinerio filtrą.

2.5 lentelė

SAD algoritmo rezultatai po Vinerio filtro pritaikymo

Nr.	Puse lango dydis	Disparitiškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor.koef.	Kameros kor.koef.	Stalo kor.koef.	Galvos kor.koef.	Lempos kor.koef.
1	5	10	36,4262335	0,768402	0,226225	0,613131	0,798572	0,423437	0,474666
2	5	13	38,5634472	0,805741	0,227103	0,616499	0,796171	0,586833	0,576239
3	5	15	39,5774537	0,788678	0,227789	0,481833	0,797571	0,565505	0,567676
4	5	16	40,0298566	0,782944	0,227922	0,483269	0,79826	0,566173	0,567491
5	5	19	40,8566619	0,778199	0,228006	0,484732	0,79893	0,566931	0,574216
6	7	10	41,184264	0,792241	0,223043	0,657042	0,794585	0,431551	0,59205
7	7	13	42,3542715	0,80143	0,223081	0,658964	0,795391	0,584627	0,582369
8	7	15	44,1326829	0,789773	0,22306	0,578979	0,796257	0,561615	0,576406
9	7	19	46,176296	0,784905	0,223057	0,579698	0,796832	0,562306	0,576718
10	7	16	46,8159001	0,788159	0,223059	0,579235	0,796468	0,561865	0,576484
11	9	10	48,516311	0,782977	0,221681	0,665887	0,759272	0,41136	0,590402
12	9	15	50,7627254	0,786667	0,221704	0,665309	0,763184	0,547918	0,576117
13	9	13	50,8407259	0,796201	0,221704	0,665568	0,762062	0,569933	0,585627
14	9	16	53,1027404	0,786582	0,221704	0,665306	0,763198	0,547934	0,576126
15	9	19	56,0355592	0,784965	0,221704	0,665246	0,763481	0,54825	0,576297

2.6 lentelė

SAD algoritmo rezultatų palyginimas be filtravimo ir po Vinerio filtro pritaikymo.

Nr.	Puse lango dydis	Disparitiškumas	kor.koef. skirtumas	laiko atlikimo skirtumas
1	5	10	0,026853826	2,823618
2	5	13	0,02140729	2,277615
3	5	16	0,027528198	2,667617
4	5	15	0,01594586	1,51321
5	5	19	0,024635365	0,920406
6	7	13	0,004694821	0,733205
7	7	10	0,026505966	1,263608
8	7	15	0,013854399	1,825212
9	7	16	0,238957285	1,794011
10	7	19	0,014868383	2,199614
11	9	10	0,018329171	0,608404
12	9	13	0,00167624	0,717605
13	9	15	0,011343896	2,324415
14	9	16	0,02135065	0,124801
15	9	19	0,013469427	-0,96721
Vidurkis			0,03209472	1,388409

Kitas basic approach algoritmas, kuris buvo išbandytas yra SSAD (Sum of Squared differences – 1.5.3 skyrius). Kaip ir aukščiau aprašyti bandymai pirmiausiai buvo atliktas rezultatų įvertinimas be filtravimo panaudojimo, išrinkti geriausi rezultatai ir jiems pritaikytos filtravimo funkcijos. Vertinimo metodika visiškai nesikeičia ir svarbiausi įvertinimo kriterijai yra bendras koreliacijos koeficientas ir algoritmo atlikimo laikas.

Po geriausių rezultatų be filtravimo (8 priedas 2.7 lentelė) išrinkimo, pritaikomi Medianos ir Vinerio filtrai. Medianos filtravimo rezultatai pateikti 8 priedo 2.8 lentelėje, o Vinerio filtravimo 8 priedo 2.9 lentelėje. Iš gautų duomenų matome, kad medianos filtras beveik neįtakoja bendrąjį koreliacijos koeficientą, bet Vinerio filtro pritaikymas vidutiniškai 2,7%, tam tikrais atvejais ir 5%, pagerina galutinį rezultatą. Taip pat reikia pastebėti, kad Vinerio filtravimas buvo atliekamas beveik tokį pat laiką kaip ir medianos filtravimas. O tokiais atvejais, kai naudojami abu filtrai sunaudojimo laikas dvigubai sumažėja. Visi palyginimo duomenys yra 2.10 lentelėje.

2.10 lentelė

SAD algoritmo bandymų rezultatų skirtumas be filtravimo i po jo.

Nr.	Puse lango dydis	Disparitiškumas	Medianos filtras		Medianos ir Vinerio filtras	
			kor.koef. skirtumas	laiko atlikimo skirtumas	kor.koef. skirtumas	laiko atlikimo skirtumas
1	5	10	0,011799468	1,7628113	0,029943332	0,2496016
2	3	13	0,017846196	0,9516061	0,044692246	2,1216136
3	3	15	0,017784148	0,3276021	0,044623115	0,1404009
4	3	19	0,025093282	0,2652017	0,054481065	0,8580055
5	3	16	0,019703442	0,6708043	0,047199115	1,2792082
6	5	13	0,005983378	2,3868153	0,021310287	0,624004
7	5	19	0,010190436	2,7456176	0,026488528	0,3432022
8	5	16	0,007594973	0,624004	0,023419419	3,5100225
9	5	15	0,006433903	0,1872012	0,021822238	0,3588023
10	7	10	0,006087404	0,5928038	0,01965028	5,0544324
11	7	15	0,003745255	2,028013	0,015286134	0,1404009
12	7	13	0,003069437	0	0,014702832	0,2496016
13	7	16	0,00394444	0,4212027	0,015736161	2,4648158
14	9	10	0,003297745	3,9780255	0,015137101	0,9672062
15	7	19	0,005015484	0,3744024	0,016958623	1,2168078
Vidurkis			0,009839266	1,1544074	0,027430032	1,305208367

Paskutinis Bazinis lokalinis algoritmas kuris buvo išbandytas yra STAD (angl. *Sum of truncated absolute differences*). Jo išbandymui panaudotas visiškai toks pat aukščiau aprašytas įvertinimo metodas. 9 priede 2.11 lentelėje pateikti bandymų rezultatai be filtravimo, 2.12 lentelėje

- po medianos filtravimo, 2.13 lentelėje – po medianos ir Vinerio filtravimų. Palyginus anksčiau gautus ir šio STAD tyrimo duomenys matome, kad naudojant filtravimą galima pagerinti rezultatą net iki 13%. Visi duomenys parodantys kaip filtravimas įtakoja galutinį rezultatą yra pateikti 2.14 lentelėje.

2.14 lentelė

STAD algoritmo bandymų rezultatų skirtumas be filtravimo i po jo.

Nr.	Puse lango dydis	Disparitiškumas	Medianos filtras		Medianos -> Vinerio filtras	
			kor.koef. skirtumas	laiko atlikimo skirtumas[s]	kor.koef. skirtumas	laiko atlikimo skirtumas
1	3	10	0,031791704	2,0436131	0,0718852	1,092007
2	3	13	0,035427641	4,0872262	0,07290086	5,304034
3	1	10	0,061625706	-1,669211	0,13351487	-1,2636081
4	3	19	0,041741705	3,7596241	0,08010725	5,1324329
5	3	16	0,036710783	2,4180155	0,0731643	0,8424054
6	3	15	0,033787271	-0,436803	0,06964691	-0,7176046
7	5	10	0,015673453	2,9328188	0,04339232	2,1372137
8	5	13	0,017166356	0,4056026	0,04165796	3,4788223
9	5	16	0,01717616	1,6692107	0,04193462	0,1716011
10	5	15	0,016563254	-0,780005	0,04049912	2,8860185
11	5	19	0,020456046	0,8736056	0,04600777	2,4180155
12	7	13	0,011414933	-1,138807	0,03239551	1,9188123
13	7	15	0,010814557	-0,577204	0,03058697	0,8892057
14	7	16	0,01219789	12,339679	0,03295761	0,3900025
15	7	19	0,013602203	1,1388073	0,03499376	0,8112052
Vidurkis			0,025076644	1,8044116	0,05637633	1,699370893

2.1.3. Bazinio lokalinio algoritmo tyrimo išvados.

2.1.2 dalyje buvo atliktas Basic approach algoritmų tyrimas buvo išbandyti trys algoritmai, kurie skiriasi tarpusavyje panaudojamu matematiniu modeliu. Buvo gauti duomenys iš kurių buvo išrenkami geriausi. Pagrindiniai vertinimo kriterijai buvo bendras koreliacijos koeficientas ir algoritmo veikimo laikas. Taip pat buvo nustatyta, kad geram algoritmo veikimui sistemą reikia teisingai sukonfigūruoti (teisingai nustatyti parametrus: disparitiškumas, ir slenkančio lango dydis). Kiekvienas erdvinis vaizdas yra skirtingo dydžio ir pobūdžio, todėl minėtus parametrus eksperimentiniu ir statistiniu būdu reikia nustatinti individualiai. Tai yra, reikia rasti algoritmo gaunamus maksimumus (bendras koreliacijos koeficientas ir atlikimo laikas), ir tam tikrose ribose nuo maksimumo atlikinėti tyrimą. Taip pat bandymų metu buvo nustatyta, kad galima galutinį rezultatą pagerinti naudojant filtrus. Darbe panaudoti medianos (aukšto dažnio) ir Vinerio (žemo

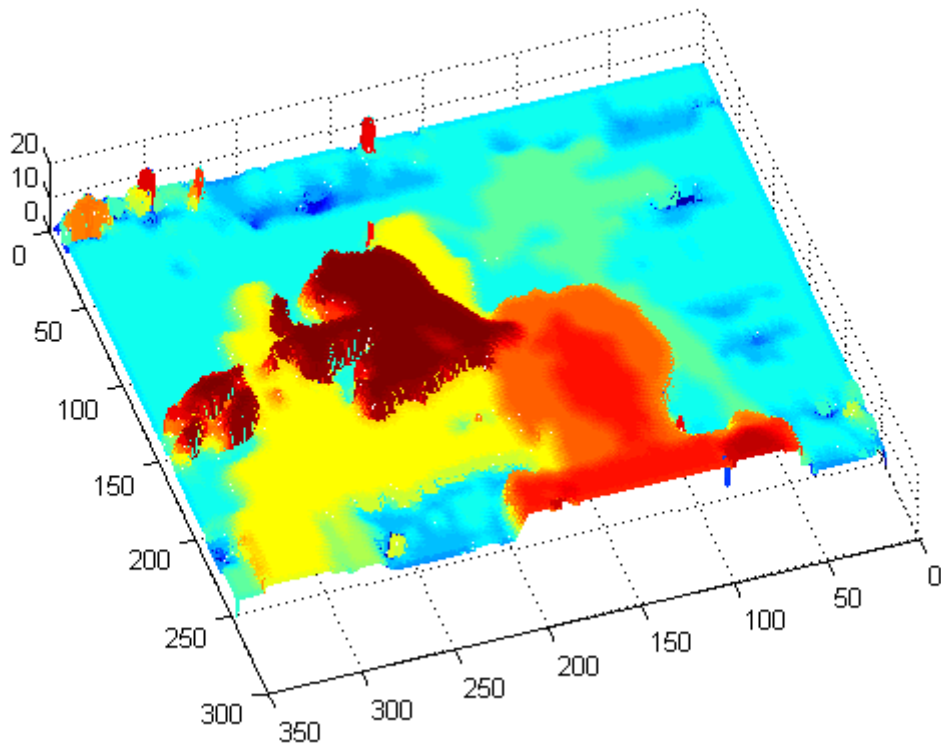
dažnio) filtrai. Jais galutinis rezultatas buvo pagerintas nuo 1% iki 8% (10 priede pateikti filtravimo pvz.). Filtrų naudojimas prailgina algoritmo veikimo laiką, bet pailgėjimas, palyginus su algoritmo veikimo laiku be filtravimo, yra labai mažas. Todėl jis beveik neįtakoja bendram visos sistemos veikimui.

2.15 lentelėje pateikti trys geriausių bazinių lokalinių algoritmų rezultatai. Iš jų matome, kad geriausias bendras koreliacijos koeficientas buvo lygus 0,805741 (SAD algoritmas). Jo erdvinis atvaizdavimas yra parodytas 2.5 paveiksle. Bet reikia atkreipti dėmesį į tai, kad šio algoritmo atlikimo laikas yra 38,6 sekundės. Greičiausias atlikimo laikas yra 34,1 sekundė (SSAD algoritmas). Jo koreliacijos bendras koeficientas yra lygus 0,755. Palyginus abu geriausius rezultatus matome, kad bendras koreliacijos koeficientas skiriasi 5%, o atlikimo laikas 4,5 sekundės. Todėl renkantis algoritmą vartotojas turi įvertinti visus sprendimo variantus ir pasirinkti tinkamiausią.

2.15 lentelė

Geriausių basic approach bandymų rezultatų suvestinė.

Algoritmas	Puse lango dydis	Disparitetai	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor. koef.	Kameros kor. koef.	Stalo kor. koef.	Galvos kor. koef.	Lempos kor. koef.	Geriausias pagal
SAD	5	13	38,56345	0,805741	0,2271	0,616499	0,7961	0,5868	0,576239	Kor. Koef.
SAD	5	10	36,42623	0,768402	0,2262	0,613131	0,7985	0,4234	0,474666	Laiką
SAD	5	13	38,56345	0,805741	0,2271	0,616499	0,7961	0,5868	0,576239	Laiką
SAD	5	15	39,57745	0,788678	0,2277	0,481833	0,7975	0,5655	0,567676	Laiką
SSAD	5	13	36,19223	0,797998	0,1896	0,626211	0,7924	0,5556	0,57477	Kor. Koef.
SSAD	5	10	34,11742	0,755131	0,1881	0,624903	0,7980	0,3634	0,466809	Laiką
SSAD	3	13	36,03623	0,784439	0,1871	0,54103	0,7863	0,5955	0,527933	Laiką
SSAD	3	15	34,74142	0,770146	0,1907	0,389656	0,7907	0,5707	0,546541	Laiką
STAD	3	15	36,19223	0,756689	0,0350	0,324974	0,7730	0,5594	0,526162	Kor. Koef.
STAD	3	10	34,30462	0,616815	0,1791	0,432035	0,7785	0,2190	0,250482	Laiką
STAD	3	13	38,61025	0,752092	0,0717	0,43758	0,7724	0,5738	0,458553	Laiką
STAD	1	10	32,40141	0,578016	0,1282	0,50728	0,7032	0,2257	0,183377	Laiką



2.5 pav. Bazinio lokalinio algoritmo geriausias rezultatas pagal kor. koeficientą.

2.2 Bazinio lokalinio algoritmo tobulinimas panaudojant tarp pikselinę analizę (angl. Sub-pixel estimation).

Pagrindinis trūkumas basic approach algoritmo yra tai kad, panaudojant fiksuotą slenkantį langą, galutiniame disparitiškumo vaizde gaunami figūrų kontūrai yra netikslūs ir neatitinka realybės. Tokios paklaidos pavyzdys yra pavaizduotas 1.15 paveiksle. Šių paklaidų įvertinimui buvo įvesti papildomi atskirų objektų koreliacijos koeficientai. Iš aukščiau atliktų tyrimų matome, kad jie kinta nuo 0,1 iki 0,8 ribose. Šiai problemai išspręsti yra pritaikoma tarp-pikselinė analizė.

Skaitmeninis vaizdas yra kvantuotas (diskretizuotas) į pikselius kurie surašyti į tam tikrą matricą. Dažniausiai galvojama, kad pikselis yra mažiausias skaitmeninio vaizdo skaičiavimo vienetas, todėl objektai vaizde gali būti netikslūs. Šios paklaidos atsiranda ir prieš duomenų apdorojimą (pvz. Vaizdo jutiklio paklaida, vaizdo duomenų kodavimas) taip pat ir po apdorojimo (pvz. dėl slenkančio lango dydžio atliekant basic approach algoritmą)[9].

Aukščiau atliktame tyrime disparitiškumui gauti buvo naudojamas elementarus koordinačių skirtumas. Panaudojant tarp pikselinę analizę parinkto taško disparitiškumui skaičiuoti, įvertinami šalia jo esantys disparitiškumo lygiai. Jų pagalba atliekama korekcija ir randamas reikalingas disparitiškumas.

Šiam patobulinimui išbandyti naudojamas toks pat algoritmas aprašytas 2.1.1. skyriuje. Visi

gauti duomenys yra pateikti 11 priede. Bandymų parametrai palikti tokie pat kaip ir ankstesniame tyrime:

- Naudojami trijų tipų algoritmai (SAD, SSAD, STAD);
- Pusė slenkančio lango dydis kinta nuo 1 iki 15 pikselių ribose;
- Disparitiškumo diskretizavimas kinta nuo 7 iki 15 ribose;
- Pritaikyti Medianos ir Vynerio filtrai.

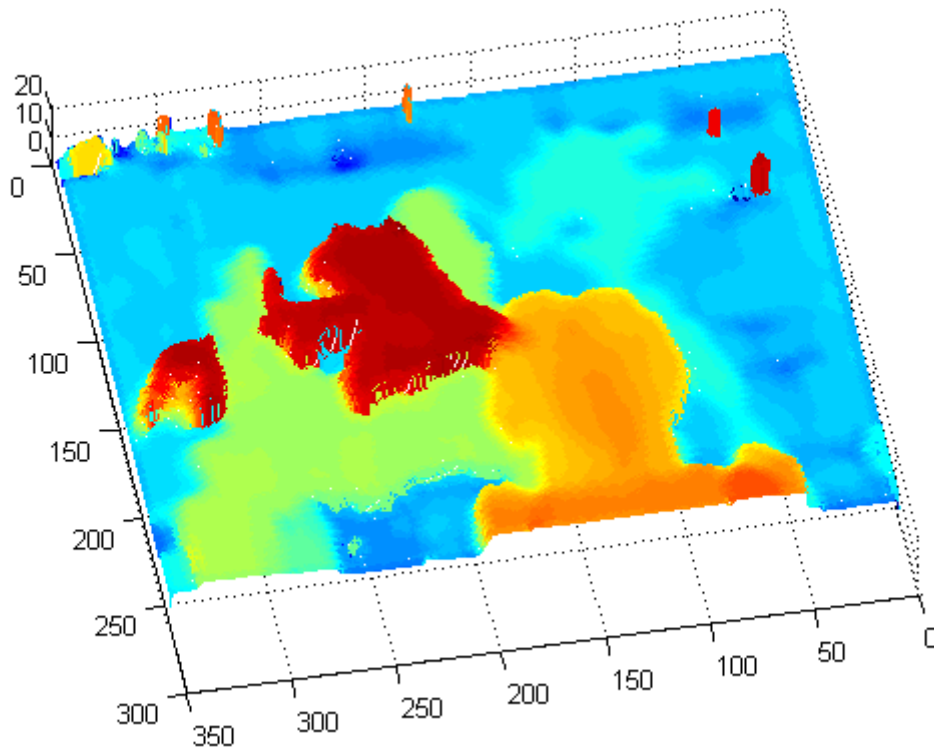
Visų išbandytų algoritmų, be filtravimo pritaikymo, bendrojo koreliacijos koeficiento rezultatai yra labai panašūs į paprastus Basic approach rezultatus (2.1.2 skyrius). Tik prie mažo disparitiškumo lygio (nuo 13 ir mažiau) bendras koreliacijos koeficientas, dėl gaunamų triukšmų stipriai sumažėja iki 0,2 ir mažiau. Taip pat labai pailgėja (kai kuriais atvejais dvigubai ar dar daugiau) algoritmo veikimo laikas. Tai parodo, kad sistema yra daugiau apkraunama, ir tokio algoritmo įgyvendinimui reikalingi didesni skaičiavimo resursai. Kai kurie trūkumai išsprendžiami pritaikant filtravimo funkcijas. Jų dėka bendrasis visų algoritmų (SAD, SSAD, STAD) koreliacijos koeficientas padidėja iki 0,8, net prie žemo disparitiškumo (13 ir 10 diskretizavimo lygio) bendros koreliacijos yra apie 0,7 lygio.

Išskirtinis tarp-piksulinės analizės bruožas yra tas, kad pavyko padidinti atskirų objektų koreliacijos koeficientų lygius. Tai reiškia kad atskirų objektų disparitiškumas ir ribos apskaičiuojamos tiksliau. Kai kurių objektų koeficientai buvo pagerinti net 50%. Visi tarp-piksulinės analizės bandymų rezultatai yra pateikti 11 priede. 2.16 lentelėje pateikiu geriausias gautus rezultatus, o 2.6 paveiksle yra pavaizduotas aukščiausia koreliacijos koeficientą turinčio rezultato erdvinis atvaizdavimas.

2.16 lentelė

Geriausių tarp-piksulinės analizės bandymų rezultatų suvestinė.

Algoritmas	Pusė lango dydis	Disparitiškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor. koef.	Kameros kor. koef.	Stalo kor. koef.	Galvos kor. koef.	Lempos kor. koef.	Geriausias pagal
SAD	7	15	52,83754	0,791955	0,2570	0,623117	0,8065	0,5444	0,580904	Kor. Koef.
SAD	3	15	37,20624	0,772459	0,2272	0,421591	0,7820	0,5708	0,560062	Laiką
SAD	3	13	38,04864	0,753184	0,2309	0,577577	0,7958	0,5642	0,39835	Laiką
SAD	5	19	42,44787	0,780147	0,2571	0,517672	0,8114	0,5549	0,57684	Laiką
SSAD	7	15	48,90631	0,78508	0,2468	0,675903	0,7971	0,5042	0,579177	Kor. Koef.
SSAD	1	13	36,31703	0,750128	0,1434	0,505113	0,7087	0,5727	0,514623	Laiką
SSAD	3	15	37,81464	0,772103	0,2173	0,43047	0,7959	0,5606	0,551381	Laiką
SSAD	5	15	39,87386	0,784679	0,2402	0,535285	0,8062	0,5204	0,570317	Laiką
MAD	3	15	40,62266	0,758122	0,0334	0,361388	0,7737	0,5481	0,530588	Kor. Koef
MAD	3	13	39,32785	0,705259	0,0823	0,507854	0,7904	0,5686	0,345087	Laiką
MAD	3	16	40,26386	0,755567	0,1073	0,359342	0,7733	0,5457	0,524704	Laiką
MAD	3	15	40,62266	0,758122	0,0334	0,361388	0,7737	0,5481	0,530588	Laiką

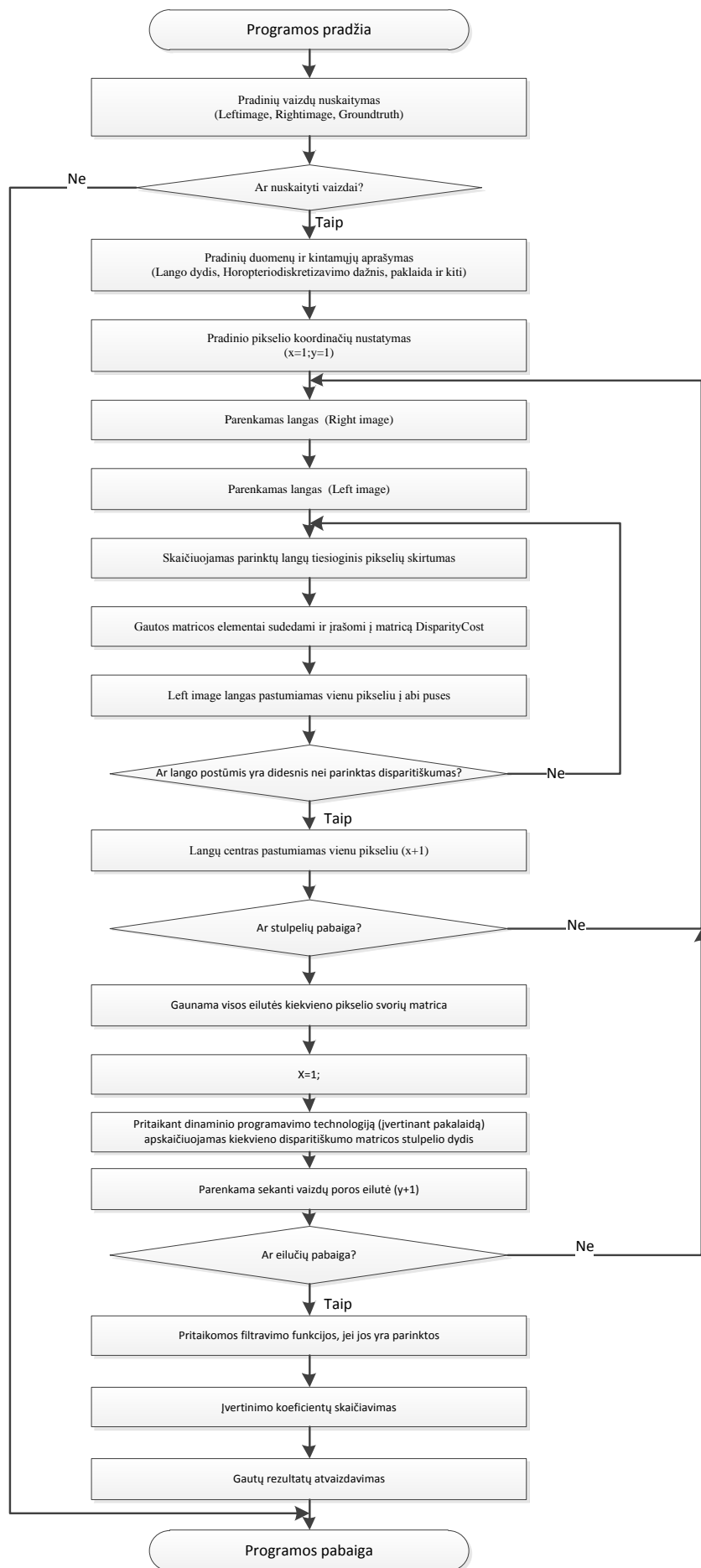


2.6 pav. Geriausias tarp-piksulinės analizės rezultato erdvinis atvaizdavimas.

2.3. Dinaminis programavimas (angl. *dynamic programming*).

2.3.1. Dinaminio programavimo algoritmo aprašymas.

Iš aukščiau atliktų angl. basic approach algoritmų bandymų matome, kad gaunami rezultatai yra su triukšmais, kurie labai įtakoja koreliacijos koeficientą. Juose kiekvienas disparitiškumo taškas buvo skaičiuojamas atskirai pagal pasirinktą matematinį modelį. Gautiems triukšmams pašalinti naudojami medianos ir vynerio filtrai. Bet ir po jų panaudojimo rezultatai nėra didesni nei 0,8 dydžio. Šiuos triukšmus galima panaikinti pritaikant glotnumo apribojimą (angl. smoothness constraint). Kitaip tariant, kiekvieno taško diparitiškumą įtakoja ne tik tiesiogiai apskaičiuotas dydis, bet ir šalia esantys taškai. Pavyzdžiui, eilutėje šalia esančių pikselių disparitiškumas gali skirtis ± 3 ribose. Bet iškyla problema, kaip parinkti optimalų disparitiškumą. Pirmiausia randamas disparitiškumas, panaudojant slenkantį langą, o paskui atliekamas jo pakeitimas priklausomai nuo šalia atskiro taško esamų kitų taškų disparitiškumų. Šiai užduočiai išspręsti gerai tinka dinaminio programavimo technologijos. Dinaminį programavimą galima priskirti prie globalių algoritmų (1.5.2 skyrius).



2.7 pav. Dinaminio programavimo algoritmas.

Naudojamo dinaminio programavimo algoritmas yra pavaizduotas 2.7 paveiksle, o programos kodas yra pateiktas 12 priede.

Prieš programos paleidimą ją reikia sukonfigūruoti. Kaip ir basic approach algoritme reikia pasirinkti disparitiškumą ir slenkančio lango dydį. Laiko taupymo sumetimais slenkančio lango dydis yra kintantis. Šis kitimas yra įgyvendintas „for“ ciklo pagalba, kuriame slenkančio lango dydis kinta ribose nuo 1 iki 15. Skirtingai nuo basic approach algoritmų atsiranda dar vienas parametras naudojama kaip „bauda“ už disparitiškumo pikselių skirtumą tarpusavyje.

Programos pradžioje nuskaitomi pradiniai vaizdai (stereo pora ir atraminio disparitiškumo vaizdas). Visi vaizdai, jei jie yra spalvoti paverčiami juodai-pilkais (angl. Gray scale). Parenkami numatyti parametrai (lango dydis, disparitiškumas ir kiti parametrai). Po auščiau aprašytų žingsnių įjungiamas ciklas atliekantis kairiojo (programos kintamasis : leftI) vaizdo skanavimą. Pirmiausia vaizdo matricoje parenkama eilutė ir stulpelis. Šis taškas yra slenkančio lango centras. Pagal lango centrą apskaičiuojamas slenkančio lango dydis ir jo koordinatės. Taip pat nustatomos mažiausios ir didžiausios disparitiškumo ribos (jos tiesiogiai priklauso nuo parinkto disparitiškumo prieš programos paleidimą).

Toliau iš kairiojo vaizdo (programos kintamasis : leftI) iškerpamas slenkantis langas ir duomenys išsaugomos atskiroje matricoje. Toks pat langas turi būti iškirptas ir iš dešiniojo vaizdo (programos kintamasis : rightI), tik šiuo atveju langas turi būti slenkantis ir lango koordinatės kinta po vieną pikselį į abi puses nuo lango centro numatytose disparitiškumo ribose. Duomenys taip yra įrašomi į atskirą matricą. Toliau skaičiuojamas gautų matricų tiesioginis skirtumas, o gautos matricos duomenys sudedami tarpusavyje ir gaunamas koeficientas, atvaizduojantis langų skirtumų energiją, įrašomas į atskirą matricą. Tokiu būdu gaunama kairiojo ir dešiniojo vaizdų eilučių koeficientų rinkinys, kuris yra panaudojamas atliekant dinaminį programavimą. Jo eigoje įvertinami gauti koeficientai ir disparitiškumo taškų skirtumas tarpusavyje. Pabaigoje gaunamas optimaliosios eilutės vaizdo gylis. Po šio žingsnio programoje parenkama kita apdorojamo vaizdo eilutė ir iš naujo atliekamas aprašytas algoritmas.

2.3.2. Dinaminio programavimo algoritmo bandymų rezultatai.

Šio algoritmo bandymui ir įvertinimui panaudojau tokią pat metodiką, kuri buvo panaudota įvertinant bazinius lokalius algoritmus. Pirmiausia buvo atliekami tyrimai nepritaikant filtravimų funkcijų. Buvo keičiami tik baziniai algoritmo parametrai : disparitiškumas ir slenkančio lango dydis. Darbo laiko taupymui programa buvo patobulinta „for“ ciklu. Juo atliekant bandymą automatiškai keičiasi slenkančio lango dydis, o gauti duomenys įrašomi į atskirą masyvą. Programos tekstas yra pateiktas 12 priede, o dinaminio programavimo visi gauti bandymų rezultatai 13 priede.

Dinaminio programavimo bandymų rezultatai be filtravimo pritaikymo.

Nr.	Puse lango dydis	Disparitiškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor.koef.	Kameros kor.koef.	Stalo kor.koef.	Galvos kor.koef.	Lempos kor.koef.
1	1	19	78,9989	0,81046	0,13200	0,60483	0,76278	0,59956	0,58621
2	3	19	80,8709	0,80510	0,20237	0,62820	0,80997	0,57372	0,56804
3	5	19	90,1061	0,79934	0,19533	0,62401	0,79467	0,55003	0,56791
4	1	16	65,4736	0,81048	0,13200	0,60483	0,76278	0,59956	0,58629
5	3	16	68,9836	0,80516	0,20237	0,62820	0,80997	0,57372	0,56830
6	5	16	74,1784	0,79934	0,19533	0,62401	0,79467	0,55003	0,56789
7	1	15	62,2288	0,81053	0,13200	0,60483	0,76278	0,59956	0,58638
8	3	15	65,7544	0,80551	0,20237	0,62820	0,80997	0,57431	0,56963
9	5	15	70,6216	0,79934	0,19533	0,62401	0,79467	0,55022	0,56776
10	1	13	56,5347	0,82080	0,13200	0,60483	0,76278	0,64727	0,59152
11	3	13	62,9308	0,81877	0,20237	0,62820	0,81452	0,61375	0,58164
12	5	13	64,0696	0,81436	0,19533	0,62401	0,79634	0,58523	0,59454
13	1	10	47,0655	0,78793	0,13200	0,60483	0,76278	0,57880	0,46286
14	3	10	51,1059	0,77604	0,20237	0,62820	0,81452	0,43240	0,42645
15	5	10	55,2399	0,79090	0,19533	0,62401	0,79888	0,43922	0,51485

2.17 lentelėje pateikti bandymų rezultatai be filtravimo pritaikymo. Žalia spalva yra pažymėtas rezultatas turintis geriausią bendrąjį koreliacijos koeficientą. Geltona spalva pažymėtas greičiausiai veikiantis rezultatas. Iš lentelės matome kad bendrasis koreliacijos koeficientas kinta ribose nuo 0,77(14 bandymas 17 lentelė) iki 0,82(10 bandymas 17 lentelė) ribose. Skirtumas yra tik 5%, todėl svarbiu įvertinimo kriterijumi tampa algoritmo veikimo laikas.

Sekantys bandymai buvo atlikti pridėdant prie algoritmo filtravimo funkcijas. Jos buvo pritaikomos jau prie gauto rezultato (apskaičiuoto disparitiškumo). Pirmas panaudotas filtras buvo medianos (aukšto dažnio) filtras. 2.18 lentelėje yra pateikti bandymų rezultatai po medianos filtravimo. Lengvesniam rezultatų palyginimui pagrindiniai parametrai paimti tokie pat kaip 2.17 lentelėje. Geriausias gautas bendrasis koreliacijos koeficientas yra lygus 0,822891 (10 bandymas 2.18 lentelė) ir pažymėtas žalia spalva, o greičiausias bandymas buvo atliktas per 47,4243 sekundės (13 bandymas 2.18 lentelė) ir yra pažymėtas geltona spalva. Rezultatai, palyginus su aukščiau gautais (be filtravimo pritaikymo), pakito tik apie 1,5 procento.

Dinaminio programavimo bandymų rezultatai pritaikant medianos filtrą.

Nr.	Puse lango dydis	Disparit iškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor.koef.	Kameros kor.koef.	Stalo kor.koef.	Galvos kor.koef.	Lempos kor.koef.
1	1	19	76,09729	0,812539	0,130188	0,608767	0,774699	0,599373	0,588381
2	3	19	82,27493	0,806513	0,209235	0,630146	0,809584	0,57482	0,570589
3	5	19	85,25455	0,799523	0,19559	0,62519	0,795617	0,551158	0,569581
4	1	16	65,89482	0,812539	0,130188	0,608767	0,774699	0,599373	0,588381
5	3	16	69,71685	0,806533	0,209235	0,630146	0,809584	0,57482	0,5707
6	5	16	72,79007	0,799523	0,19559	0,62519	0,795617	0,551158	0,569581
7	1	15	62,104	0,812528	0,130188	0,608767	0,774699	0,599373	0,588283
8	3	15	65,14602	0,806802	0,209235	0,630146	0,809584	0,575263	0,571723
9	5	15	69,34244	0,799512	0,19559	0,62519	0,795617	0,551158	0,569489
10	1	13	59,17118	0,822891	0,130188	0,608767	0,774699	0,647203	0,593565
11	3	13	63,22721	0,82009	0,209235	0,630146	0,81385	0,61525	0,585428
12	5	13	63,88241	0,814715	0,19559	0,62519	0,796803	0,586474	0,596268
13	1	10	47,4243	0,792909	0,130188	0,608767	0,774699	0,581168	0,472981
14	3	10	51,68313	0,779843	0,209235	0,630146	0,81385	0,437972	0,437743
15	5	10	53,49274	0,791427	0,19559	0,62519	0,798868	0,439911	0,517489

Paskutinis filtras kuris buvo pritaikytas yra Vinerio (žemo dažnio) filtras. Rezultatai yra surašyti į 2.19 lentelė. Kaip aukščiau pateiktose lentelėse žalia spalva pažymėtas bandymas turintis geriausią koreliacijos koeficientą (10 bandymas 2.19 lentelė) ir yra lygus 0,8369, o geltona spalva greičiausias bandymas (13 bandymas 2.19 lentelė) ir yra lygus 48,03271 sekundėms. Bendras visų rezultatų bendrų koreliacijos skirtumas sumažėjo iki 3 % ir kinta ribose nuo 0,79 iki 0,83. Todėl svarbiu įvertinimo koeficientu tampa algoritmo veikimo laikas. Taip pat reikia pastebėti kad papildomi koreliacijos koeficientai buvo pagerinti apie 10 procentų.

Dinaminio programavimo bandymų rezultatai pritaikant medianos ir Vinerio filtrus.

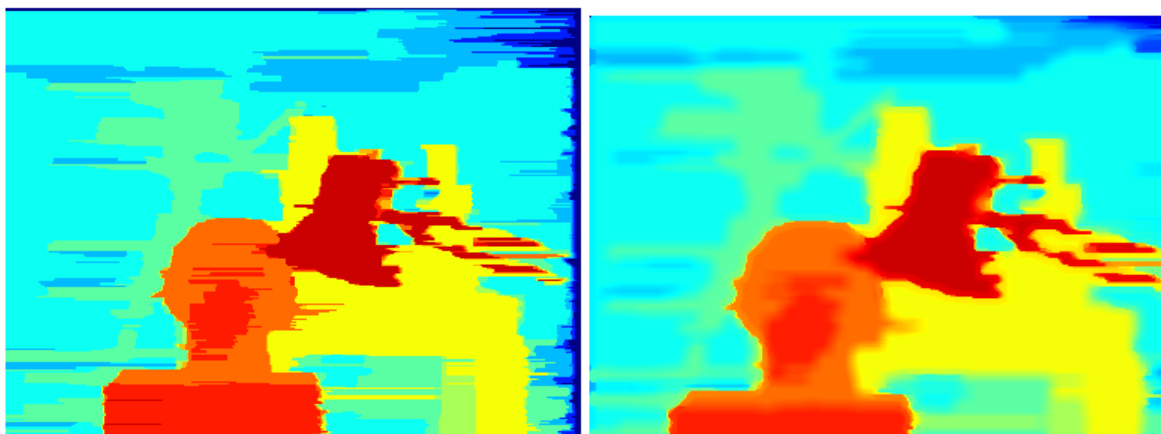
Nr.	Puse lango dydis	Disparit iškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor.koef.	Kameros kor.koef.	Stalo kor.koef.	Galvos kor.koef.	Lempos kor.koef.
1	1	19	81,01132	0,824653	0,195254	0,700142	0,809053	0,620569	0,601698
2	3	19	82,46213	0,817554	0,265156	0,718009	0,836331	0,596199	0,582788
3	5	19	87,54776	0,808516	0,234812	0,69386	0,819374	0,572135	0,579908
4	1	16	70,79325	0,824653	0,195254	0,700142	0,809053	0,620569	0,601698
5	3	16	81,88492	0,817574	0,265156	0,718009	0,836329	0,596194	0,582863
6	5	16	77,12689	0,808516	0,234812	0,69386	0,819374	0,572135	0,579908
7	1	15	65,03682	0,824646	0,195254	0,700142	0,809051	0,620565	0,601614
8	3	15	75,33288	0,817819	0,265156	0,718009	0,836317	0,596629	0,583747
9	5	15	73,13327	0,808501	0,234812	0,693861	0,819369	0,572126	0,579787
10	1	13	60,38799	0,8369	0,194962	0,700095	0,807856	0,675104	0,614721
11	3	13	68,14124	0,832045	0,265148	0,717923	0,839479	0,641123	0,602165
12	5	13	66,84643	0,8244	0,234802	0,694054	0,81931	0,610924	0,610462
13	1	10	48,03271	0,809029	0,190066	0,697481	0,804697	0,610743	0,492835
14	3	10	52,10433	0,794517	0,264972	0,715437	0,836602	0,452601	0,452141
15	5	10	55,02155	0,803529	0,234421	0,691281	0,817495	0,457318	0,535387

2.3.3. Dinaminio programavimo algoritmo bandymų išvados.

Buvo atliktas dinaminio programavimo algoritmo bandymai. Gauti duomenys (13 priedas) iš kurių buvo išrenkami geriausi rezultatai. Taip pat rezultatams pagerinti buvo pritaikomos filtravimo funkcijos. 2.20 lentelėje yra surašyti visų bandymų, be filtravimo, po medianos, po vinerio filtravimo, geriausi rezultatai. Šioje lentelėje pirmame stulpelyje surašyti filtrų pavadinimai: tuščias laukelis – filtras nepanaudotas, M – medianos filtras, MV – pirmiausia pritaikytas medianos, o paskui vinerio filtras. Filtravimo pritaikymo rezultatai yra pavaizduoti 2.8 paveiksle, o geriausio gauto rezultato erdvinis disparitiškumo atvaizdavimas 2.9 paveiksle. Kaip aukščiau buvo minėta (1.5.2 skyrius), tiriamas algoritmas yra globalus. Jame nėra apskaičiuojamas kiekvienas atskiras pikselis, o vyksta visos eilutės duomenų surinkimas ir apdorojimas. Medianos ir vinerio filtravimo dėka išėjo pagerinti koreliacijos koreficientą apie 1,5 procento, o vidutiniškai algoritmo laikas pailgėja apie 0,5 sekundės. Tai yra labai mažas dydis palyginus su viso algoritmo veikimo laiku, todėl galima teigti, kad filtro įdėjimas į šį algoritmą stipriai neįtakoja jo veikimą.

Geriausių dinaminio programavimo algoritmo bandymų rezultatų suvestinė.

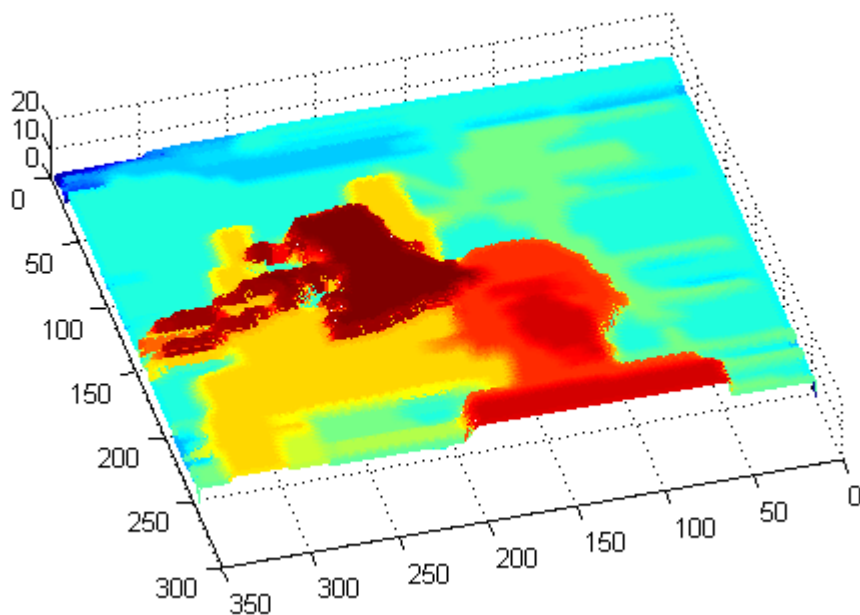
Filtrai	Puse lango dydis	Disparitiškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor. koef.	Kameros kor. koef.	Stalo kor. koef.	Galvos kor. koef.	Lempos kor. koef.	Geriausias pagal
	1	13	56,53476	0,82080	0,1320	0,60483	0,7627	0,6472	0,59152	Kor. Koef.
	1	10	47,0655	0,78793	0,1320	0,60483	0,7627	0,5788	0,46286	Laiką
	3	10	51,10593	0,77604	0,2023	0,62820	0,8145	0,4324	0,42645	Laiką
	5	10	55,23995	0,79090	0,1953	0,62401	0,7988	0,4392	0,51485	Laiką
M	1	13	59,17118	0,82289	0,1301	0,60876	0,7746	0,6472	0,59356	Kor. Koef.
M	1	10	47,4243	0,79290	0,1301	0,60876	0,7746	0,5811	0,47298	Laiką
M	3	10	51,68313	0,77984	0,2092	0,63014	0,8138	0,4379	0,43774	Laiką
M	5	10	53,49274	0,79142	0,1955	0,62519	0,7988	0,4399	0,51748	Laiką
MV	1	13	60,38799	0,8369	0,1949	0,70009	0,8078	0,6751	0,61472	Kor. Koef
MV	1	10	48,03271	0,80902	0,1900	0,69748	0,8046	0,6107	0,49283	Laiką
MV	3	10	52,10433	0,79451	0,2649	0,71543	0,8366	0,4526	0,45214	Laiką
MV	5	10	55,02155	0,80352	0,2344	0,69128	0,8174	0,4573	0,53538	Laiką



a)

b)

2.8 pav. Dinaminio programavimo algoritmo disparitiškumo atvaizdavimas: a) prieš; b) po filtravimo.



2.9 pav. Geriausio dinaminio programavimo algoritmo gauto disparitiškumo erdvinis atvaizdavimas.

2.4. Išbandytų algoritmų suvestinė.

Šiame darbe buvo atliktas kelių kompiuterinės erdvinės regos algoritmų bandymai ir jų įvertinimas. Pirmiausia buvo atlikti basic approach algoritmų bandymai. Išvardinti algoritmai yra priskiriami prie lokalinių/vietinių algoritmų (1.5.2 skyrius), nes skaičiavimas atliekamas su kiekvienu pikseliu atskirai, nepaisant šalia esančių taškų. Pilnas bandymų aprašymas yra 2 skyriuje. Apibendrinant atliktą tyrimą galima padaryti basic approach algoritmo trūkumų ir privalumų suvestinę:

Privalumai:

- Paprasti algoritmų matematiniai modeliai;
- Palyginus su kitais išbandytais algoritmais yra greičiausiai veikiantys;
- Lengvai sukonfigūruojamas reikalingai užduočiai;
- Galima pritaikyti daugybe filtravimo funkcijų.

Trūkumai:

- Didelis triukšmų lygis, nes neįvertinami šalia esantys pikseliai;
- Objektų kraštai netikslus, nes jie tiesiogiai priklauso nuo slenkančio lango dydžio;
- Gerai veikia sukonfigūruotas, bet jei pastoviai keičiasi vaizdo porų prigimtis yra galimybė kad triukšmų kiekis gali būti labai didelis.
- Nėra optimalių filtravimo funkcijų kurias galima būtų pritaikyti algoritmui, todėl žmogus pats, priklausomai nuo gaunamų rezultatų, turi parinkti jam reikalingą filtravimo funkciją.

Sekantis algoritmas kuris buvo išbandytas tai tarp-pikseline (angl. sub-pixel estimation) analizė (6.2 skyrius). Tai yra patobulintas basic approach algoritmų rinkinys. Jo veikimas nesiskiria veikimu nuo įprasto algoritmo, tik disparitiškumo taškai priklauso nuo šalia esančių taškų. Atlikus bandymus buvo pastebėta, kad bendrasis koreliacijos koeficientas beveik nepakito, palyginus su įprastu algoritmu. Bet buvo pagerinti atskirų objektų koreliacijos koeficientai, o tai reiškia kad objektų formos yra atpažinamos geriau. Taip reikia pastebėti kad pailgėja algoritmo veikimo laikas, todėl renkantis šį algoritmą reikia įvertinti turimas technines galimybes.

Pirmi bandymai buvo atlikti su lokaliniais/vietiniais algoritmais, todėl paskutiniam tyrimui buvo parinktas globalus algoritmas (1.5.2 skyrius). Šiai užduočiai buvo parinktas dinaminio programavimo algoritmas. Pilnas tyrimo aprašymas yra padarytas 2.3 skyriuje. Gauti koreliacijos koeficientai (bendri ir atskirų objektų) buvo geresni nei kitų tyrimų. O veikimo laikas yra didesnis nei aukščiau aprašytų algoritmų. Dinaminio programavimo algoritmų privalumai ir trūkumai:

Privalumai:

- Palyginus su kitais algoritmais, geriausi rezultatai;
- Pakankamai tiksliai atpažinami objektų kontūrai;
- Galimybe keisti ir tobulinti jau esama algoritmą.

Trūkumai:

- „Skanavimo tiesės“ efektas, nes apdorojimas vyksta visos tiesės o ne kiekvieno pikselio atskirai;
- Algoritmo veikimo laikas tiesiogiai priklauso atliekamų matematinių skaičiavimų.

Visi išvardinti bandymai buvo atlikti MatLab2013b programa. Skaičiavimus atliko bendros paskirties procesorius AMD A8-4500M. Jis nėra pritaikytas vaizdų apdorojimui. Kompiuteriai turintys NVIDIA vaizdų kortos palaikančios CUDA tvarkyklės (angl. *driver*) keliais kartais pagreitinti visus skaičiavimus. O tai leidžia visus išvardintus algoritmus išbandyti realaus laiko (angl. *real-time*) sistemose. Taip pat realaus laiko sistemoms įgyvendinimui galima panaudoti FPGA (angl. *Field-Programmable Gate Array*).

Paskutinis tikslas kuri reikėjo pasiekti šiame darbe buvo erdviųjų vaizdų algoritmų įvertinimo metodikos sukūrimas. Pagrindiniai įvertinimo kriterijai yra koreliacijos koeficientas ir algoritmo atlikimo laikas, taip pat papildomai buvo įvesti atskirų objektų koreliacijos koeficientai, kurie atvaizduoja kaip gerai yra atpažinami objektai (jų kontūrai ir pan.). Koreliacijai apskaičiuoti yra reikalingas atraminis disparitiškumas, sintetinis vaizdas kuris yra pateikiamas kartu su stereo vaizdų pora. Tyrimuose buvo panaudotas „tsukuba“ stereo vaizdų rinkinys. Po atliktų tyrimų galima teigti kad mano aprašyta metodika pakankamai gerai aprašo ir įvertina naudojamus kompiuterinės erdvinės regos algoritmus.

IŠVADOS

Kompiuterinės erdvinės regos sistemos kūrimas yra sudėtingas procesas. Jis priklauso nuo daugelio faktorių: nuo techninio įgyvendinimo, nuo parinkto algoritmo, nuo stebimos aplinkos ir kitų faktorių. Darbe buvo išanalizuotas standartinis kompiuterinės erdvinės regos sistemos modelis, ir jo sudedamosios dalys. Taip pat buvo nustatyta, kad prieš erdvinės sistemos modelio naudojimą, jį reikia sukalibruoti ir paruošti darbui.

Sekantys atliktas svarbus žingsnis buvo algoritmų parinkimas ir jų klasifikavimas. Pagal veikimo principus jie buvo išskirti į dvi grupes: lokalius ir globalius. Iš kiekvienos grupės buvo paimti keli algoritmai, su kuriais ir buvo atliekamas eksperimentinis tyrimas. Iš lokalių algoritmų buvo išrinkti trys bazinio lokalinio sprendimo algoritmai : SAD (angl. *Sum of Absolute differences*), SSAD (angl. *Sum of Squared differences*), STAD (angl. *Sum of truncated absolute differences*). Išvardintų algoritmų rezultatų tobulinimui ir triukšmų pašalinimui buvo pritaikyti medianos ir vinerio filtrai, o objektų kontūrų tikslesniam atpažinimui pritaikyta tarp pikselinė analizė. Iš globalių algoritmų buvo išrinktas dinaminis programavimas. Jo rezultatams gerinti irgi buvo pritaikyti medianos ir vinerio filtrai.

Algoritmų rezultatų palyginimui ir įvertinimui skurta įvertinimo metodika. Pagrindinis jos elementas yra bendras koreliacijos koeficientas ir algoritmo veikimo laikas. Taip pat įvesti papildomi (tiksliniai) koreliacijos koeficientai, kurie parodo kaip tiksliai atpažinami objektai. Vadovaujantys išvardintais kriterijais išrinkti geriausi bandymų rezultatai. Jų vizualinis įvertinimas taip pat patvirtina kad naudojama metodika tinkama kompiuterinės erdvinės regos įvertinimui.

LITERATŪRA

1. **Drennan M.** An Implementation of Camera Calibration Algorithms. Department of Electrical and Computer Engineering Clemson University.
2. **Nalpantidis Lazaros, Georgios Christou Sirakoulis, Antonios Gasteratos** Review of stereo vision algorithms: from software to hardware. International Journal of Optomechatronics, 2: 435–462, 2008.
3. **D. Scharstein, R. Szeliski**, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms Int. Jour. Computer Vision, 47(1/2/3):7–42, 2002. Prieiga per internetą: <<http://vision.middlebury.edu/stereo/data/scenes2001/>>
4. **E. Trucco, A. Verri**, Introductory Techniques for 3-D Computer Vision, PrenticeHall, 1998.
5. **G. Bradsky, A. Kaehler**, Learning OpenCV, O'Reilly, 2008
6. **Olga Veksler** Stereo Correspondence by Dynamic Programming on a Tree. University of Western Ontario Computer Science Department, Middlesex College 361 London ON N6A 5B7 Canada.
7. Realistic CG Stereo Image Dataset with Ground Truth Disparity Maps , **Sara Martull** University of Tsukuba, Japan, **Martin Peris** Cyberdyne Inc., Japan, **Kazuhiro Fukui** University of Tsukuba, Japan.
8. **Federico Tombari, Stefano Mattoccia, Luigi Di Stefano** Segmentation-based adaptive support for accurate stereo correspondence. Prieiga per internetą: <http://vision.deis.unibo.it/~smatt/stereo_segment_support.html/>
9. **Robert B. Fisher** Sub-pixel estimation, University of Edinburgh.
10. **Chuen-Horng Lin, Cheng-Hsin Kuo, Li-Jung Fu** A stereo matching algorithm based on adaptive windows, International Journal of Electronic Commerce Studies Vol.3, No.1, pp.21-34, 2012. Prieiga per internetą: <<http://www.academic-journals.org/ojs2/index.php/ijecs/article/viewFile/1011/98>>
11. **Emmanouil Z. Psarakis, Georgios D. Evangelidis** An Enhanced Correlation-Based Method for Stereo Correspondence with Sub-Pixel Accuracy, Department of Computer Engineering and Informatics University of Patras, 26500 Patras, Greece. Prieiga per internetą: <<http://xanthippi.ceid.upatras.gr/people/evangelidis/encc/>>
12. **F. Tombari, S. Mattoccia, L. Di Stefano** "Segmentation-based adaptive support for accurate stereo correspondence " IEEE Pacific-Rim Symposium on Image and Video Technology (PSIVT 2007). December 17-19, 2007, Santiago, Chile.

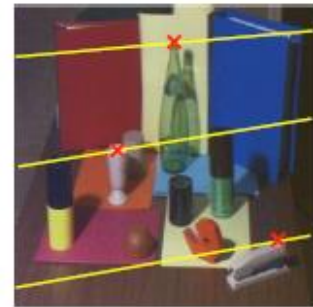
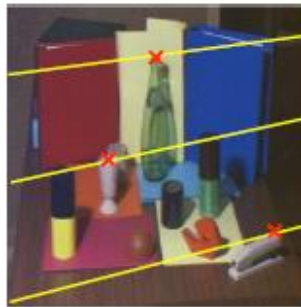
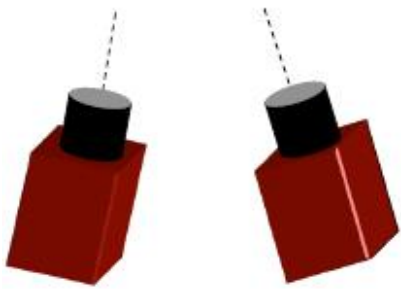
13. **S. Mattocchia, F. Tombari, L. Di Stefano**, "Stereo vision enabling precise border localization within a scanline optimization framework" 8th Asian Conference on Computer Vision ([ACCV 2007](#)) November 18-22, 2007, Tokyo, Japan.

14. **Anonymous**. Correlation based similarity measures-Summary. Prieiga per internetą:
<<http://siddhantahuja.wordpress.com/category/stereo-vision/>>

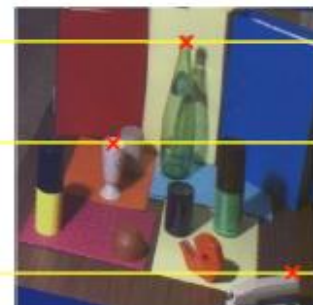
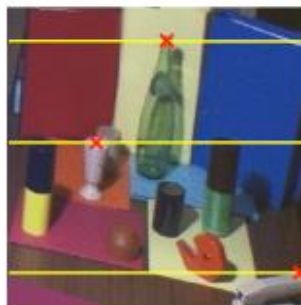
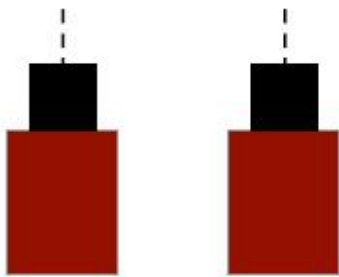
15. **Anonymous**. Camera Calibration and 3D Reconstruction. Prieiga per internetą:
<http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html#camera_calibrationopencv>

PRIEDAI

Stereo kameros kalibravimo principas:



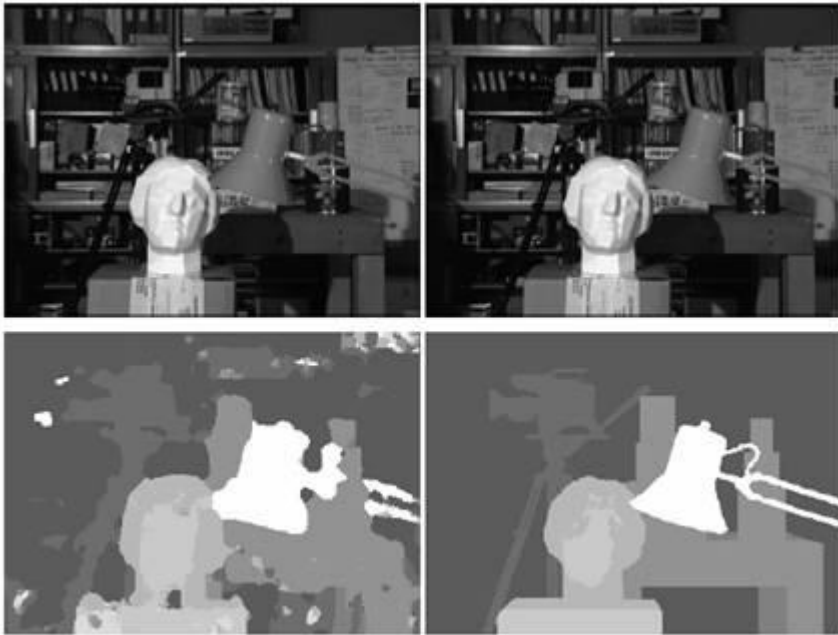
Nestandartizuotas stereo sistemos modelis



Standartizuotas stereo sistemos modelis

2 PRIEDAS

Stereo vaizdų pora, apskaičiuotas ir atraminis (angl. *groundtruth*) disparitiškumas:



Vidinių kameros iškraipymų taisymas po kalibravimo.



Tsukuba stereo porų rinkinys.

Kairysis vaizdas:



Dešinysis vaizdas:



„Atraminis“ disparitiškumas (angl. groundtruth):



Bazinio lokalinio sprendimo (angl. *basic approach*) MatLab2013b programa.

```

clear all
close all
clc
%% -----
% vaizdų informacijos nuskaitymas
iminfo('imL.png');
iminfo('imR.png');
iminfo('gt.png');
masyvas = zeros(15,15); % galutinių duomenų surinkimas

for z = 5 : 2 : 5
% GroundTruth nuskaitymas iš failo
[GT, map] = imread('gt.png');
% jei reikalingas groundtruth atvaizdavimas
% figure(5)
% imshow(GT,map);
% -----
% vaizdų nuskaitymas ir jų transformavimas į "pilką" (angl. grey) formata
hIdtc = vision.ImageDataTypeConverter;
hCsc = vision.ColorSpaceConverter('Conversion','RGB to intensity');
leftI3chan = step(hIdtc,imread('imL.png'));
leftI = step(hCsc,leftI3chan);
rightI3chan = step(hIdtc,imread('imR.png'));
rightI = step(hCsc,rightI3chan);
% -----
% galima pakeisti į GPU masyva bet veikia tik su Nvidia vaizdo kortomis
% kurios palaiko CUDA technologija
% rightI = gpuArray(rightI);
% leftI = gpuArray(leftI);
% -----
% originalių vaizdų atvaizdavimas
figure(1), clf;
subplot(1,2,1), subimage(rightI3chan), title('Right image');
subplot(1,2,2), subimage(leftI3chan), title('Left image');
% -----
% vaizdų skirtumo atvaizdavimas
figure(2)
imshow(rightI)
imshow(cat(3,rightI,leftI,leftI)), axis image;
% -----
% bazinių kintamųjų aprašymas
t = cputime; % algoritmo veikimo laiko skaitiklio kintamasis
Dbasic = zeros(size(leftI), 'single');
Dbasic_module = zeros(size(leftI), 'single');
% panašių pikselių paieškai naudojamas langas
disparityRange = double(13); % horopterio diskretizavimas/disparitiškumas
% (2*halfBlockSize+1)-ant-(2*halfBlockSize+1) slenkantis langas.
halfBlockSize = z; % puse lango dydis
blockSize = 2*halfBlockSize+1; % pilnas slenkantis langas
tmats = cell(blockSize);
% Proceso progreso atvaizdavimo langas
hWaitBar = waitbar(0, 'Performing basic block matching...');
nRowsLeft = size(leftI, 1);

%% -----
for m = 1 : nRowsLeft % eilutes
% nustatomos vaizdo lango/bloko minimalios ir maksimalios ribos
minr = max(1,m-halfBlockSize);
maxr = min(nRowsLeft,m+halfBlockSize);

```

```

for n=1 : size(leftI,2)          % stulpeliai
    %slenkančio lango ribinių verčių nustatymas pagal X ašį
    minc = max(1,n-halfBlockSize);
    maxc = min(size(leftI,2),n+halfBlockSize);
    % aprašome paieškos lauko ribas
    mind = max( -disparityRange, 1-minc );
    maxd = min( disparityRange, size(leftI,2)-maxc );
    % paimamas langas ir atskaitos paveikslo
    template = rightI(minr:maxr,minc:maxc);
    % lango centras
    templateCenter = floor((size(template)+1)/2);
    % roi - region of interest
    roi = [minc+templateCenter(2)+mind-1 minr+templateCenter(1)-1 maxd-
mind+1 1];
    if isempty(tmats(size(template,1),size(template,2)))
        tmats(size(template,1),size(template,2)) =
vision.TemplateMatcher('ROIInputPort',true);
%pakeitimas algoritmo 'Metric', 'Maximum absolute difference' ir pan.
    end
    thisTemplateMatcher = tmats(size(template,1),size(template,2));
    % įjungiamas paieškos objektas
    loc = step(thisTemplateMatcher, leftI, template, roi);
    Dbasic(m,n) = loc(1) - roi(1) + mind;
    % modulis
    if Dbasic(m,n)<=0
        Dbasic_module(m,n) = 0;
    else
        Dbasic_module(m,n) = loc(1) - roi(1) + mind;
    end;
end
waitbar(m/nRowsLeft,hWaitBar);
end

close(hWaitBar)

figure(7)
imshow(Dbasic_module,[]);

GT = imcrop(GT,[19 19 347 251]);
imwrite(GT, map, 'GT_png.png')
figure(3)
imshow(GT,map)

Dbasic_module = imcrop(Dbasic_module,[19 19 347 251]);
figure(4)
imshow(Dbasic_module,[]), axis image, colormap('jet'), colorbar;
caxis([0 disparityRange]);
title('Depth map from basic block matching');

% filtravimo funkcijos
k = medfilt2(Dbasic_module)
k = wiener2(k, [9 9])
% Jei reikia rezultatų atvaizdavimas
% figure(5)
% imshow(k,[]), axis image, colormap('jet'), colorbar;
% caxis([0 disparityRange]);
% title('Depth map from basic block matching');

r = corr2(k, GT)          %bendras koreliacijos koeficientas
% -----
%Iškerpamos 5 dalys iš groundtruth ir skaičiuojami tiksliniai koreliacijos
koeficientai
GT_fonas = imcrop(GT,[5 85 70 110]);

```



```

GT_kamera = imcrop(GT,[40 25 120 90]);
GT_stalas = imcrop(GT,[240 170 340 240]);
GT_galva = imcrop(GT,[90 120 100 160]);
GT_lempa = imcrop(GT,[170 70 110 110]);

figure (5)
subplot(1,5,1), subimage(GT_fonas,map), title('Fonas');
subplot(1,5,2), subimage(GT_kamera,map), title('Kamera');
subplot(1,5,3), subimage(GT_stalas,map), title('Stalas');
subplot(1,5,4), subimage(GT_galva,map), title('Galva');
subplot(1,5,5), subimage(GT_lempa,map), title('Lempa');

k_fonas = imcrop(k,[5 85 70 110]);
k_kamera = imcrop(k,[40 25 120 90]);
k_stalas = imcrop(k,[240 170 340 240]);
k_galva = imcrop(k,[90 120 100 160]);
k_lempa = imcrop(k,[170 70 110 110]);

figure (6)
subplot(1,5,1), imshow(k_fonas,[]), title('Fonas');
subplot(1,5,2), imshow(k_kamera,[]), title('Kamera');
subplot(1,5,3), imshow(k_stalas,[]), title('Stalas');
subplot(1,5,4), imshow(k_galva,[]), title('Galva');
subplot(1,5,5), imshow(k_lempa,[]), title('Lempa');

r_fonas = corr2(k_fonas, GT_fonas) %fono koreliacijos koeficientas
r_kamera = corr2(k_kamera, GT_kamera) %kameros koreliacijos koeficientas
r_stalas = corr2(k_stalas, GT_stalas) %stalo koreliacijos koeficientas
r_galva = corr2(k_galva, GT_galva) %galvos koreliacijos koeficientas
r_lempa = corr2(k_lempa, GT_lempa) %lempos koreliacijos koeficientas

% irasymas duomeniu i masyva
masyvas (z,1) = z;
masyvas (z,2) = disparityRange;
masyvas (z,3) = cputime - t; %laikas
masyvas (z,4) = r; %bendras koreliacijos koeficientas
masyvas (z,5) = r_fonas; % fono koreliacija
masyvas (z,6) = r_kamera; % kameros koreliacija
masyvas (z,7) = r_stalas; % stalo koreliacija
masyvas (z,8) = r_galva; % galvos koreliacija
masyvas (z,9) = r_lempa; % lempos koreliacija

end
% Ciklo pabaiga

```

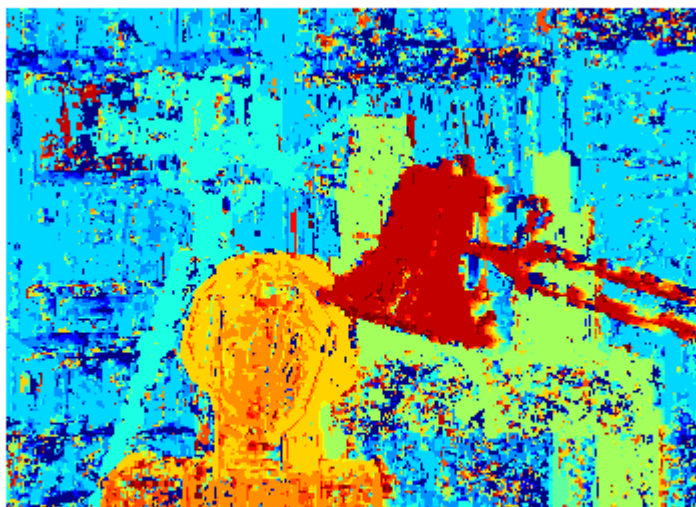
6 PRIEDAS

Bazinio lokalinio sprendimo tyrimo rezultatai.

Priedamas excel failas: 6_priedas_basic_approach.xlsx

Triukšmų, atsirandančių nuo mažo slenkančio lango parinkimo, atvaizdavimas.

Depth map from basic block matching



SSAD algoritmo bandymų suvestinė

2.7 lentelė

SSAD algoritmo bandymų rezultatai be filtravimo

Nr.	Puse lango dydis	Disparitiškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor.koef.	Kameros kor.koef.	Stalo kor.koef.	Galvos kor.koef.	Lempos kor.koef.
1	5	10	33,867817	0,725187	0,130889	0,520878	0,749185	0,333124	0,430723
2	3	13	33,914617	0,739746	0,114885	0,429857	0,703447	0,527635	0,495041
3	3	15	34,601021	0,725522	0,113564	0,317585	0,692175	0,500531	0,517732
4	3	19	34,647822	0,700872	0,113564	0,303167	0,657184	0,483800	0,508703
5	3	16	34,663422	0,718468	0,113564	0,315025	0,674525	0,490439	0,515385
6	5	13	35,568228	0,776688	0,130889	0,520151	0,739626	0,520494	0,555057
7	5	19	36,176631	0,746655	0,130889	0,401754	0,738358	0,490286	0,539246
8	5	16	36,473033	0,754618	0,130889	0,403035	0,738358	0,496876	0,544368
9	5	15	37,955043	0,761284	0,130889	0,403351	0,738358	0,503825	0,549206
10	7	10	38,282645	0,759082	0,156874	0,536865	0,748232	0,335638	0,533838
11	7	15	39,842655	0,769233	0,156874	0,524905	0,739799	0,502664	0,561554
12	7	13	40,373058	0,777194	0,156874	0,531159	0,740477	0,515309	0,559184
13	7	16	41,948668	0,763849	0,156874	0,524905	0,739799	0,496823	0,558660
14	9	10	42,635073	0,756005	0,154425	0,532654	0,708202	0,302557	0,553054
15	7	19	43,758280	0,759664	0,156874	0,524905	0,739799	0,487270	0,556417

2.8 lentelė

SSAD algoritmo bandymų rezultatai po medianos filtravimo

Nr.	Puse lango dydis	Disparitiškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor.koef.	Kameros kor.koef.	Stalo kor.koef.	Galvos kor.koef.	Lempos kor.koef.
1	5	10	35,630628	0,736987	0,139358	0,564859	0,761420	0,344341	0,450268
2	3	13	32,963011	0,757593	0,135465	0,471098	0,730525	0,552565	0,507978
3	3	15	34,928623	0,743306	0,135465	0,352842	0,729103	0,527111	0,528342
4	3	19	34,913023	0,725965	0,135465	0,341252	0,705097	0,518224	0,530940
5	3	16	33,992617	0,738171	0,135465	0,35271	0,716663	0,521232	0,528889
6	5	13	37,955043	0,782671	0,139358	0,564859	0,753270	0,526881	0,558820
7	5	19	38,922249	0,756846	0,139358	0,433692	0,751936	0,500282	0,545945
8	5	16	37,097037	0,762213	0,139358	0,434038	0,751936	0,504996	0,548684
9	5	15	37,767842	0,767718	0,139358	0,434038	0,751936	0,510655	0,552128
10	7	10	38,875449	0,765169	0,168263	0,556925	0,755336	0,344309	0,546212
11	7	15	41,870668	0,772978	0,168263	0,547809	0,747575	0,507338	0,564560
12	7	13	40,373058	0,780263	0,168263	0,550728	0,748281	0,519745	0,560904
13	7	16	41,527466	0,767793	0,168263	0,547809	0,747575	0,502034	0,562348
14	9	10	46,613098	0,759303	0,156611	0,535987	0,714284	0,306559	0,559727
15	7	19	43,383878	0,764680	0,168263	0,547809	0,747575	0,494392	0,561600

SSAD algoritmo bandymų rezultatai po medianos ir Vinerio filtravimo filtravimo

Nr.	Puse lango dydis	Disparitiškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor.koef.	Kameros kor.koef.	Stalo kor.koef.	Galvos kor.koef.	Lempos kor.koef.
1	5	10	34,117418	0,755131	0,188146	0,624903	0,798079	0,363459	0,466808
2	3	13	36,036231	0,784439	0,187112	0,54103	0,786395	0,595523	0,527932
3	3	15	34,741422	0,770145	0,190712	0,389656	0,790712	0,570781	0,546541
4	3	19	35,505827	0,755353	0,193664	0,37852	0,771630	0,564577	0,551156
5	3	16	35,942630	0,765667	0,191702	0,391453	0,781162	0,565466	0,548077
6	5	13	36,192232	0,797998	0,189656	0,626211	0,792430	0,555643	0,574770
7	5	19	36,519834	0,773144	0,191336	0,479216	0,796196	0,530859	0,562004
8	5	16	39,983056	0,778037	0,191938	0,477545	0,794954	0,535162	0,563943
9	5	15	37,596241	0,783107	0,191384	0,47598	0,793886	0,539584	0,566909
10	7	10	43,337077	0,778732	0,208591	0,61886	0,785583	0,362552	0,569905
11	7	15	39,983056	0,784519	0,208478	0,619144	0,782379	0,532882	0,576151
12	7	13	40,622660	0,791896	0,20849	0,618814	0,781804	0,546065	0,573203
13	7	16	44,413484	0,779585	0,208478	0,61915	0,783060	0,528562	0,574525
14	9	10	43,602279	0,771142	0,173638	0,584774	0,740072	0,321772	0,583241
15	7	19	44,975088	0,776623	0,208478	0,619145	0,783645	0,520383	0,573797

STAD algoritmo bandymų suvestinė

2.11 lentelė

STAD algoritmo bandymų rezultatai be filtravimo

Nr.	Puse lango dydis	Disparitiškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor.koef.	Kameros kor.koef.	Stalo kor.koef.	Galvos kor.koef.	Lempos kor.koef.
1	3	10	33,212612	0,544929	0,106920	0,321089	0,633710	0,172911	0,225083
2	3	13	33,306213	0,679191	0,045805	0,304246	0,612617	0,458356	0,405343
3	1	10	33,665015	0,444500	0,032145	0,276487	0,423581	0,177190	0,168989
4	3	19	34,491821	0,662892	0,087469	0,231938	0,588147	0,404070	0,455681
5	3	16	35,802229	0,680810	0,087526	0,229346	0,597835	0,436982	0,477418
6	3	15	36,909836	0,687042	0,017941	0,237326	0,604406	0,446044	0,484665
7	5	10	37,128238	0,540778	0,176090	0,296229	0,689780	0,156447	0,140745
8	5	13	37,674241	0,681749	0,134298	0,287206	0,675116	0,452638	0,329785
9	5	16	40,248258	0,701741	0,046003	0,253152	0,644249	0,434147	0,445470
10	5	15	40,388658	0,709962	0,046480	0,255312	0,648519	0,442664	0,463027
11	5	19	41,745867	0,683749	0,045460	0,244117	0,643509	0,391182	0,408328
12	7	13	45,24029	0,668189	0,121751	0,209623	0,643301	0,467990	0,272830
13	7	15	46,753499	0,701787	0,031777	0,204811	0,630753	0,481993	0,410406
14	7	16	48,032707	0,693913	0,031777	0,211200	0,630753	0,473664	0,399337
15	7	19	51,261928	0,677621	0,031440	0,212035	0,630753	0,438845	0,379825

2.12 lentelė

STAD algoritmo bandymų rezultatai po medianos filtravimo

Nr.	Puse lango dydis	Disparitiškumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor.koef.	Kameros kor.koef.	Stalo kor.koef.	Galvos kor.koef.	Lempos kor.koef.
1	3	10	35,256226	0,576721	0,138322	0,356307	0,700874	0,200227	0,235181
2	3	13	37,393439	0,714619	0,045305	0,357879	0,687502	0,517125	0,436373
3	1	10	31,995805	0,506126	0,069743	0,364377	0,568278	0,204862	0,167135
4	3	19	38,251445	0,704634	0,072252	0,275681	0,677140	0,463917	0,478963
5	3	16	38,220245	0,717520	0,072238	0,280359	0,681310	0,497825	0,495833
6	3	15	36,473033	0,720829	0,011929	0,284606	0,684078	0,502093	0,504276
7	5	10	40,061056	0,556451	0,191626	0,314514	0,711346	0,163188	0,139073
8	5	13	38,079844	0,698915	0,161517	0,313910	0,704379	0,487192	0,339661
9	5	16	41,917468	0,718917	0,077785	0,283416	0,685257	0,466758	0,454715
10	5	15	39,608653	0,726525	0,077785	0,283486	0,689449	0,474762	0,473199
11	5	19	42,619473	0,704205	0,078333	0,269605	0,685257	0,420967	0,422825
12	7	13	44,101482	0,679604	0,120682	0,231985	0,667800	0,476547	0,281176
13	7	15	46,176296	0,712601	0,027606	0,232151	0,656114	0,490142	0,419236
14	7	16	60,372387	0,706111	0,027606	0,237982	0,656114	0,483359	0,410810
15	7	19	52,400735	0,691223	0,028061	0,232593	0,656114	0,449824	0,395234

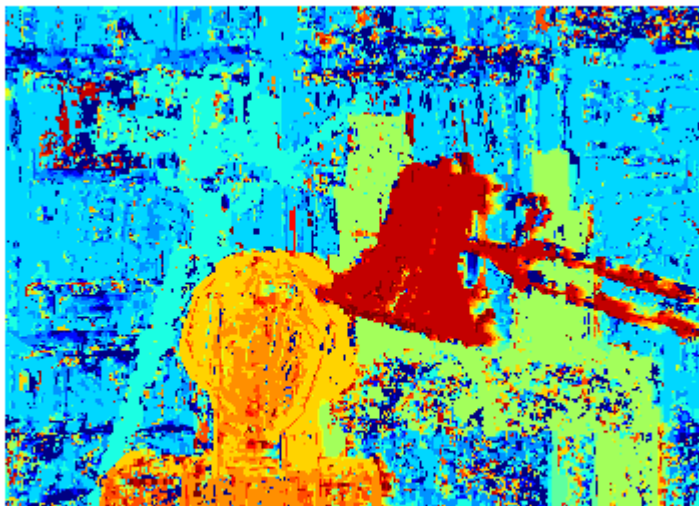
STAD algoritmo bandymų rezultatai po medianos ir Vinerio filtravimo filtravimo

Nr.	Puse lango dydis	Disparitiskumas	Algoritmo veikimo laikas	Bendras kor. koef.	Fono kor.koef.	Kameros kor.koef.	Stalo kor.koef.	Galvos kor.koef.	Lempos kor.koef.
1	3	10	38,610247	0,752092	0,071781	0,437580	0,772440	0,573884	0,458552
2	3	13	32,401407	0,578015	0,12821	0,507280	0,703267	0,225740	0,183377
3	1	10	39,624254	0,742999	0,110609	0,313220	0,774689	0,515102	0,501676
4	3	19	36,644634	0,753974	0,107335	0,322517	0,772801	0,555164	0,517999
5	3	16	36,192232	0,756689	0,035032	0,324974	0,773002	0,559455	0,526162
6	3	15	39,265451	0,584170	0,198254	0,366928	0,761023	0,173928	0,141902
7	5	10	41,153063	0,723407	0,158336	0,361910	0,757439	0,527122	0,352602
8	5	13	40,419859	0,743676	0,112725	0,329889	0,747283	0,508791	0,472061
9	5	16	43,274677	0,750461	0,110624	0,327620	0,748205	0,515431	0,490698
10	5	15	44,163883	0,729757	0,116339	0,305059	0,752319	0,457138	0,439867
11	5	19	47,159102	0,700584	0,118041	0,254196	0,731045	0,503661	0,291819
12	7	13	47,642705	0,732374	0,068051	0,255456	0,718572	0,517188	0,433694
13	7	15	48,422710	0,726870	0,069788	0,259852	0,722754	0,511697	0,425553
14	7	16	52,073133	0,712615	0,073087	0,243855	0,729043	0,476921	0,410239
15	7	19	38,610247	0,752092	0,071781	0,437580	0,772440	0,573884	0,458552

Medianos ir Vinerio filtravimo pavyzdys.

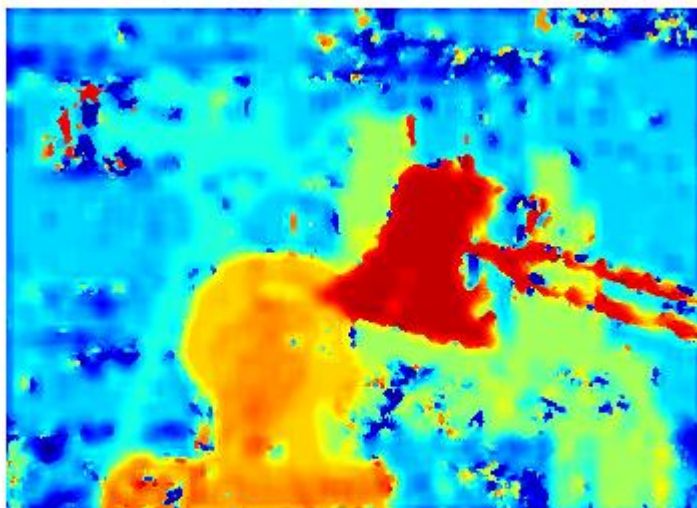
Prieš:

Depth map from basic block matching



Po:

Depth map from basic block matching



Tarp pikselinės analizės tyrimo rezultatai.

Pridedamas excel failas : 11_priedas_sub_pixel_estimation.xlsx

Dinaminio programavimo programos tekstas:

```

clear all
close all
clc
masyvas = zeros(15,15);

for z = 1 : 2 : 1

% GroundTruth nuskaitymas is failo
[GT, map] = imread ('gt.png');
% -----
% vaizdų nuskaitymas ir jų transformavimas į "pilką" formata
hIdtc = vision.ImageDataTypeConverter;
hCsc = vision.ColorSpaceConverter('Conversion','RGB to intensity');
leftI3chan = step(hIdtc,imread('imL.png'));
leftI = step(hCsc,leftI3chan);
rightI3chan = step(hIdtc,imread('imR.png'));
rightI = step(hCsc,rightI3chan);
% -----
% originalių vaizdų atvaizdavimas
figure(1), clf;
subplot(1,2,1), subimage(rightI3chan), title('Right image');
subplot(1,2,2), subimage(leftI3chan), title('Left image');
% -----
% vaizdų skirtumo atvaizdavimas
figure(2)
imshow(rightI)
imshow(cat(3,rightI,leftI,leftI)), axis image;
% -----
% baziniu kintamųjų aprašymas
t = cputime;
disparityRange = double(13);
nRowsLeft = size(leftI, 1);
halfBlockSize = z;
Ddynamic = zeros(size(leftI), 'single');
finf = 1e3;
disparityCost = finf*ones(size(leftI,2), 2*disparityRange + 1, 'single');
disparityPenalty = 0.5;
hWaitBar = waitbar(0,'Using dynamic programming for smoothing...');
for m=1:nRowsLeft
    disparityCost(:) = finf;
    minr = max(1,m-halfBlockSize);
    maxr = min(nRowsLeft,m+halfBlockSize);
    for n=1:size(leftI,2)
        minc = max(1,n-halfBlockSize);
        maxc = min(size(leftI,2),n+halfBlockSize);
        mind = max(-disparityRange, 1-minc );
        maxd = min( disparityRange, size(leftI,2)-maxc );
        for d=mind:maxd
            disparityCost(n, d + disparityRange + 1) =
sum(sum(abs(leftI(minr:maxr, (minc:maxc)+d) - rightI(minr:maxr,minc:maxc))));
        end
    end
end

optimalIndices = zeros(size(disparityCost), 'single');
cp = disparityCost(end,:);
for j=size(disparityCost,1)-1:-1:1
    cfinf = (size(disparityCost,1) - j + 1)*finf;
    [v,ix] = min([cfinf cfinf cp(1:end-4)+3*disparityPenalty;
                 cfinf cp(1:end-3)+2*disparityPenalty;
                 cp(1:end-2)+disparityPenalty;

```

```

        cp(2:end-1);
        cp(3:end)+disparityPenalty;
        cp(4:end)+2*disparityPenalty cfinf;
        cp(5:end)+3*disparityPenalty cfinf cfinf], [], 1);
    cp = [cfinf disparityCost(j,2:end-1)+v cfinf];
    optimalIndices(j,2:end-1) = (2:size(disparityCost,2)-1) + (ix - 4);
end
[~,ix] = min(cp);
Ddynamic(m,1) = ix;
for k=1:size(Ddynamic,2)-1
    Ddynamic(m,k+1) = optimalIndices(k, max(1, min(size(optimalIndices,2),
round(Ddynamic(m,k)) ) ) );
end
waitbar(m/nRowsLeft, hWaitBar);
end
close(hWaitBar);
Ddynamic = Ddynamic - disparityRange - 1;

figure(3), clf;
imshow(Ddynamic,[]), axis image, colormap('jet'), colorbar;
caxis([0 disparityRange]);
GT = imcrop(GT, [19 19 347 251]);
imwrite(GT, map, 'GT_png.png')
figure(4)
imshow(GT,map)
k = imcrop(Ddynamic, [19 19 347 251]);
figure(5)
imshow(Ddynamic,[]), axis image, colormap('jet'), colorbar;
caxis([0 disparityRange]);

k = medfilt2(k)
k = wiener2(k, [9 9])
figure(6)
imshow(k,[]), axis image, colormap('jet'), colorbar;
caxis([0 disparityRange]);
r = corr2(k, GT) %bendras koreliacijos koeficientas
% -----
%iskerpu 5 dalys is ground truth ir skaiciuoju koreliacijos koeficienta
GT_fonas = imcrop(GT, [5 85 70 110]);
GT_kamera = imcrop(GT, [40 25 120 90]);
GT_stalas = imcrop(GT, [240 170 340 240]);
GT_galva = imcrop(GT, [90 120 100 160]);
GT_lempa = imcrop(GT, [170 70 110 110]);

figure (7)
subplot(1,5,1), subimage(GT_fonas,map), title('Fonas');
subplot(1,5,2), subimage(GT_kamera,map), title('Kamera');
subplot(1,5,3), subimage(GT_stalas,map), title('Stalas');
subplot(1,5,4), subimage(GT_galva,map), title('Galva');
subplot(1,5,5), subimage(GT_lempa,map), title('Lempa');

k_fonas = imcrop(k, [5 85 70 110]);
k_kamera = imcrop(k, [40 25 120 90]);
k_stalas = imcrop(k, [240 170 340 240]);
k_galva = imcrop(k, [90 120 100 160]);
k_lempa = imcrop(k, [170 70 110 110]);

figure (8)
subplot(1,5,1), imshow(k_fonas,[]), title('Fonas');
subplot(1,5,2), imshow(k_kamera,[]), title('Kamera');
subplot(1,5,3), imshow(k_stalas,[]), title('Stalas');
subplot(1,5,4), imshow(k_galva,[]), title('Galva');
subplot(1,5,5), imshow(k_lempa,[]), title('Lempa');

```

```

r_fonas = corr2(k_fonas, GT_fonas)           %fono koreliacijos koeficientas
r_kamera = corr2(k_kamera, GT_kamera)       %kamos koreliacijos koeficientas
r_stalas = corr2(k_stalas, GT_stalas)      %stalo koreliacijos koeficientas
r_galva = corr2(k_galva, GT_galva)        %galvos koreliacijos koeficientas
r_lempa = corr2(k_lempa, GT_lempa)        %lempos koreliacijos koeficientas

% irasymas duomeniu i masyva
masyvas (z,1) = z;
masyvas (z,2) = disparityRange;
masyvas (z,3) = cputime - t;
masyvas (z,4) = r;
masyvas (z,5) = r_fonas;
masyvas (z,6) = r_kamera;
masyvas (z,7) = r_stalas;
masyvas (z,8) = r_galva;
masyvas (z,9) = r_lempa;
end

```

```

%laikas
%bendras koreliacijos koeficientas
% fono koreliacija
% kameros koreliacija
% stalo koreliacija
% galvos koreliacija
% lempos koreliacija

```

Dinaminio programavimo tyrimo rezultatai.

Priedamas excel failas : 13_priedas_dinamic_programming.xlsx