

ŠIAULIŲ UNIVERSITETAS
TECHNOLOGIJOS FAKULTETAS
ELEKTRONIKOS KATEDRA

Raimondas Zemblys

**SPAUSDINTINIO MONTAŽO PLOKŠČIŲ SURINKIMO
KOKYBĖS ĮVERTINIMAS KOMPIUTERINE REGA
Magistro darbas**

Vadovas

doc. dr. G. Daunys

ŠIAULIAI, 2008

SANTRAUKA

Zemblys R., Spausdintinio montažo plokščių surinkimo kokybės įvertinimas kompiuterine rega: Signalų technologijos magistro darbas / mokslinis vadovas doc. dr. G. Daunys; Šiaulių universitetas, Technologijos fakultetas, Elektronikos katedra, -Šiauliai, 2008. – 42p.

Elektronikos gamybos rinkoje produkcijos kokybei užtikrinti buvo pasikliaujama žmogaus vizualia apžiūra ir elektroniniais testais. Pasirodžius personaliniams kompiuteriams, gamybos pramonėje pradėta naudoti „kompiuterinė rega“. Procesas, kuriame naudojamas optinis vaizdo sensorius, pvz.: kamera, apibūdinamas kaip *automatinė optinė kontrolė* – AOK (*angl. Automated Optical Inspection - AOI*). Darbo tikslas - sukurti spausdintinio montažo plokščių kokybės įvertinimo sistemą panaudojant kompiuterinę regą, išrinkti spausdintinio montažo plokščių kokybės įvertinimui efektyviausius vaizdo apdorojimo algoritmus ir jų parametrus bei įvertinti jų efektyvumą. Taip pat nagrinėjama aplinkos veiksnių (apšvietimo, naudojamos techninės įrangos ir kt.) įtaka sistemos darbo efektyvumui.

SUMAARY

Zemblys R., Evaluation of Assembling Quality of Printed Circuit Boards Using Computer Vision: Master thesis of Signal Technology / research advisor Assoc. Dr. G. Daunys; Šiauliai University, Faculty of Technology, Electronics Department, -Šiauliai, 2008. – 42p.

Historically the electronics manufacturing market has relied on a combination of human visual inspection and electrical test methods to ensure product quality. With the advent of the personal computer, the use of "machine vision" in industrial applications gradually became more common. Process where optical sensors (i.e., cameras) are used to make specific pass/fail decisions is usually described as *Automated Optical Inspection* or *AOI*.

There are discussed problems of designing AOI system in this paper. Main goal is to select most efficient image analysis algorithm and to study other parameters that have impact for designing reliable AOI system.

TURINYS

| | |
|---|----|
| Įžanga | 6 |
| 2. Automatinės optinės kontrolės sistemos | 8 |
| 2.1. AOK sistemos sudėtis | 8 |
| 2.2. Vaizdo atpažinimo metodikos | 9 |
| 2.2.1. Tikslus formų gavimas naudojant Furjė koeficientus ir laiko perstūmimą | 9 |
| 2.2.2. Formos žymių projekcijos ir palyginimas daliniam objektų atpažinimui | 10 |
| 2.2.3. Giminingų modelių atpažinimas naudojant kelių lygių autokonvoliuciją | 12 |
| 2.3. Pasirinkta vaizdo apdorojimo metodika | 13 |
| 2.3.1. Sobel operatorius | 13 |
| 2.3.2. Canny edge detection | 14 |
| 2.3.3. Laplace operatorius | 15 |
| 2.3.4. Eigen decomposition metodas | 15 |
| 2.3.5. Corner detection | 16 |
| 2.3.6. Harris corner detector | 16 |
| 2.3.7. Vaizdų palyginimas | 17 |
| 3. Spausdintinio montažo plokščių kokybės įvertinimo sistema | 19 |
| 3.1. Techninė įranga | 19 |
| 3.2. Programinė įranga | 20 |
| 4. Sistemos efektyvumo tyrimai | 25 |
| 4.1. Fotografavimo režimo įtaka vaizdo analizei | 25 |
| 4.2. Kameros parametrų įtaka analizės rezultatams | 30 |
| 4.3. Optimalių algoritmų parametrų nustatymas | 31 |
| 4.4. Optimalaus segmento dydžio nustatymas | 36 |
| 5. Išvados ir siūlymai | 39 |
| Literatūra | 41 |

LENTELĖS

| | |
|--|----|
| 1 lentelė. Neatitinkančių segmentų skaičius spalvotame vaizde naudojant skirtingus Sobel operatoriaus nustatymus, kai segmento dydis 50x50px ir koreliacijos koeficientas 0.8..... | 26 |
| 2 lentelė. Neatitinkančių segmentų skaičius spalvotame vaizde naudojant skirtingus Canny algoritmo nustatymus, kai segmento dydis 50x50px ir koreliacijos koeficientas 0.9..... | 27 |
| 3 lentelė. Rezultatai naudojant skirtingus Eigen decomposition algoritmo parametrus, kai koreliacijos koeficientas 0.9, segmento dydis – 50x50, 75x75, 100x100 pikselių. | 33 |
| 4 lentelė. Neatitinkančių segmentų skaičius naudojant skirtingus Harris Corner Detection algoritmo nustatymus. | 35 |
| 5 lentelė. Pikselių perskaičiavimas į milimetrus..... | 36 |

PAVEIKSLAI

| | | |
|---------|---|----|
| 1 pav. | Tipinės automatinės optinės kontrolės veikimo principas | 8 |
| 2 pav. | Kamera fotografuoja atskirus gaminio segmentus | 9 |
| 3 pav. | WARP metodo žingsniai | 10 |
| 4 pav. | Panašumo tarp dviejų formos aprašų įvertinimas WARP metodu..... | 10 |
| 5 pav. | Etaloninis (kairėje) ir tiriamas (dešinėje) vaizdas, bei jų intensyvumų histogramos..... | 11 |
| 6 pav. | Ying Shan siūlomo metodo schema | 11 |
| 7 pav. | Kelių objektų MSA transformacijos | 12 |
| 8 pav. | Pradinis (dešinėje) ir Sobel operatoriumi apdorotas vaizdas | 14 |
| 9 pav. | Canny algoritmu apdorotas vaizdas | 14 |
| 10 pav. | Laplace operatoriumi apdorotas vaizdas | 15 |
| 11 pav. | Eigen decomposition metodu apdorotas vaizdas | 15 |
| 12 pav. | Corner detection algoritmu apdorotas vaizdas | 16 |
| 13 pav. | Harris corner detection algoritmu apdorotas vaizdas..... | 17 |
| 14 pav. | Suprojektuotas vaizdų analizės standas..... | 19 |
| 15 pav. | Šviesos diodų (kairėje) ir kaitrinių lempų (dešinėje) apšvietimo moduliai | 20 |
| 16 pav. | Programinės įrangos moduliai..... | 21 |
| 17 pav. | Programinės įrangos blokinė schema | 21 |
| 18 pav. | Pagrindinis programinės įrangos langas..... | 22 |
| 19 pav. | Vaizdo analizės algoritmas..... | 23 |
| 20 pav. | Gauto vaizdo analizės rezultatas | 24 |
| 21 pav. | Klaidingai identifikuotų segmentų skaičius naudojant <i>Corner detection</i> algoritmą su įvairiais parametrais | 27 |
| 22 pav. | Rezultatai su nespaltotu vaizdu naudojant <i>Harris corner detector</i> algoritmą..... | 28 |
| 23 pav. | Rezultatai su spalvotu vaizdu naudojant <i>Harris corner detector</i> algoritmą | 29 |
| 24 pav. | Rezultatai su <i>Harris corner detection</i> algoritmu, kai segmento dydis 75x75 px, koreliacijos koeficientas – 0.8..... | 29 |
| 25 pav. | Tyrimo rezultatai, kai <i>Laplace</i> operatoriaus parametras aperture size yra 7. | 32 |
| 26 pav. | Plokštės be realių defektų tyrimas parenkant skirtingus <i>Corner detection</i> parametrus, segmento dydžius ir koreliacijos koeficientus. | 34 |
| 27 pav. | Plokštės su realiais defektais tyrimas parenkant skirtingus <i>Corner detection</i> parametrus, segmento dydžius ir koreliacijos koeficientus. | 34 |
| 28 pav. | Rezultatai nenaudojant papildomo apšvietimo | 37 |
| 29 pav. | Rezultatai naudojant šviesos diodų apšvietimą..... | 37 |
| 30 pav. | Rezultatai naudojant kaitrinių lempų apšvietimą..... | 38 |
| 31 pav. | Rezultatai optiškai priartinus tiriamą plokštę..... | 39 |

IŽANGA

Šiame darbe nagrinėjama galimybė įvertinti spausdintinio montažo plokščių surinkimo kokybę panaudojant kompiuterinę regą.

Darbo tikslas - sukurti spausdintinio montažo plokščių kokybės įvertinimo sistemą panaudojant kompiuterinę regą.

Darbo uždaviniai:

- ištirti automatinės optinės kontrolės sistemų veikimo principus;
- suprojektuoti gaminių kokybės tikrinimo sistemos prototipą;
- sukurti programinę įrangą plokščių surinkimo kokybės įvertinimui;
- atlikti sistemos efektyvumo tyrimus;
- išrinkti spausdintinio montažo plokščių kokybės įvertinimui efektyviausius vaizdo apdorojimo algoritmus ir jų parametrus;
- ištirti išorinių veiksnių įtaką sistemos darbo efektyvumui.

Elektronikos gamybos rinkoje produkcijos kokybei užtikrinti buvo pasikliaujama žmogaus vizualia apžiūra ir elektroniniais testais. Pasirodžius personaliniams kompiuteriams, gamybos pramonėje pradėta naudoti „kompiuterinė rega“. Procesas, kuriame naudojamas optinis vaizdo sensorius, pvz.: kamera, apibūdinamas kaip automatinė optinė kontrolė – AOK (*angl. Automated Optical Inspection - AOI*)^[1].

Tokios sistemos buvo pradėtos naudoti apie 1980 metus. Buvo tikimasi, kad jos pakeis žmones, kurių darbas yra mažiau efektyvus tikrinant produkcijos kokybę. Pirmosios AOK sistemos nepasižymėjo dideliu greičiu, efektyvumu, lengvumu naudotis, buvo brangios, tačiau dabar AOK sistemų patikimumas nuolat auga link galutinio tikslo – 100% defektų aptikimo be klaidingų išspėjimų.

Elektronikos pramonėje yra trys pagrindinės vietos, kur gali būti naudojama AOK:

1. Padengimas lydmetaliu mišiniu (*angl. solder paste*). Tai yra pirmas ir sunkiausiai kontroliuojamas procesas spausdintinio montažo plokščių surinkime. AOK sistema, paremta standartinės kameros naudojimu, paprastai identifikuoja lydmetaliu mišinio buvimą ar nebuvimą ant takelių, o taip pat takelių išsišakojimo problemas. Lydmetaliu mišinio kiekio matavimui paprastai reikia kitokios technologijos, kaip pvz.: lazerinė

trianguliacija. Šiame proceso etape surasti defektai yra lengvai ištaisomi, o grįžtamasis ryšys su padengimo lydmetaliu procesu yra labai greitas.

2. Komponentų montavimo kontrolė. Tikrinamas komponento buvimas/nebuvimas, simetriškumas, poliškumas, kartais raidinis žymėjimas ant komponento. Rasti defektai pašalinami sąlyginai lengvai, atsakomoji reakcija komponentų montavimo procesui sąlyginai greita ir efektyvi.
3. Baigto gaminio kontrolė. Gali būti tikrinami tiek komponentų išdėstymas, tiek reikiamų vietų padengimas lydmetaliu. Šioje fazėje rasti defektai dažnai sunkiau ištaisomi, proceso atsakomoji reakcija lėta, tačiau tai leidžia didesnę paviršiaus padengimą komponentais, bei galima naudoti tik vieną AOK sistemą surinkimo linijoje, jei tai yra įmanoma ir finansiškai apsimoka.

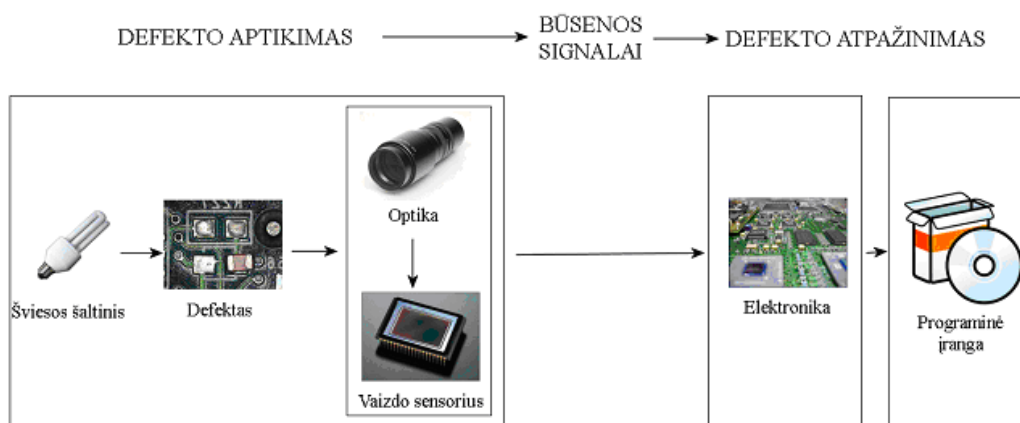
Šį darbą sudaro keli skyriai, kuriuose pateikiami AOK sistemos komponentų aprašymai, vaizdo apdorojimui naudojamų metodų analizė, aprašomi sistemos projektavimo ir efektyvumo tyrimo darbai, pateikiamos sistemos funkcionavimo ir vaizdo apdorojimo algoritmų blokinės schemos, tyrimo rezultatai ir išvados.

2. AUTOMATINĖS OPTINĖS KONTROLĖS SISTEMOS

2.1. AOK sistemos sudėtis

Tipinė AOK sistema susideda iš 4 komponentų (1 pav.):

- šviesos šaltinio;
- vaizdo įvedimo prietaiso, tokio kaip analoginė ar skaitmeninė kamera, skeneris, video mikroskopas, medicininės diagnostikos įranga;
- vaizdo įvedimo į kompiuterį plokštės (*angl. Frame grabber*) arba kitos sąsajos vaizdo duomenų perkėlimui į kompiuterį (pvz.: USB, IEEE 1394).
- programinės įrangos vaizdo atpažinimui, analizavimui, matavimui, apdorojimui ir saugojimui.



1 pav. Tipinės automatinės optinės kontrolės veikimo principas

Sistema gauna objekto vaizdą, kuris yra apšviestas koku nors šviesos šaltiniu, ir persiunčia jį į kompiuterį, kur vaizdas yra analizuojamas. Pagrindinis komponentas, kurio daromi patikimi sprendimai apie gaminio kokybę yra programinė įranga. Problema yra pasirinkti vaizdo analizavimo ir atpažinimo algoritmą taip, kad būtų tenkinami tokie reikalavimai kaip sistemos kaina, apdorojimo greitis, patikimumas.

Naudojamas šviesos šaltinis gali būti vienspalvis šviesos diodų (*angl. LED*), ar kitokia balta šviesa, taip pat kai kuriose sistemose gali būti naudojamas ir spalvotas apšvietimas.

Dažniausia tokiose sistemose naudojami dviejų tipų vaizdo jutikliai: CCD, t.y. surištojo krūvio prietaisų (*angl. Charge Coupled Devices*) bei CMOS, t.y. papildytojo metalo oksido ir puslaidininkio darinių (*angl. Complementary Metal Oxide Semiconductor*)^[2]. Gali būti naudojami linijiniai (*angl. linescan*), t.y. tokie vaizdo sensoriai kuriuose vaizdo jutiklio taškai išdėstyti viena linija, arba matricos

tipo, kai jutiklio taškai išdėlioti ne viena linija. Kai kuriose sistemose naudojami keli vaizdo jutikliai, kitose kamera yra slenkama pagal objekto paviršių ir taip gaunami atskirų segmentų vaizdai (2 pav.).



2 pav. Kamera fotografuoja atskirus gaminio segmentus

2.2. Vaizdo atpažinimo metodikos

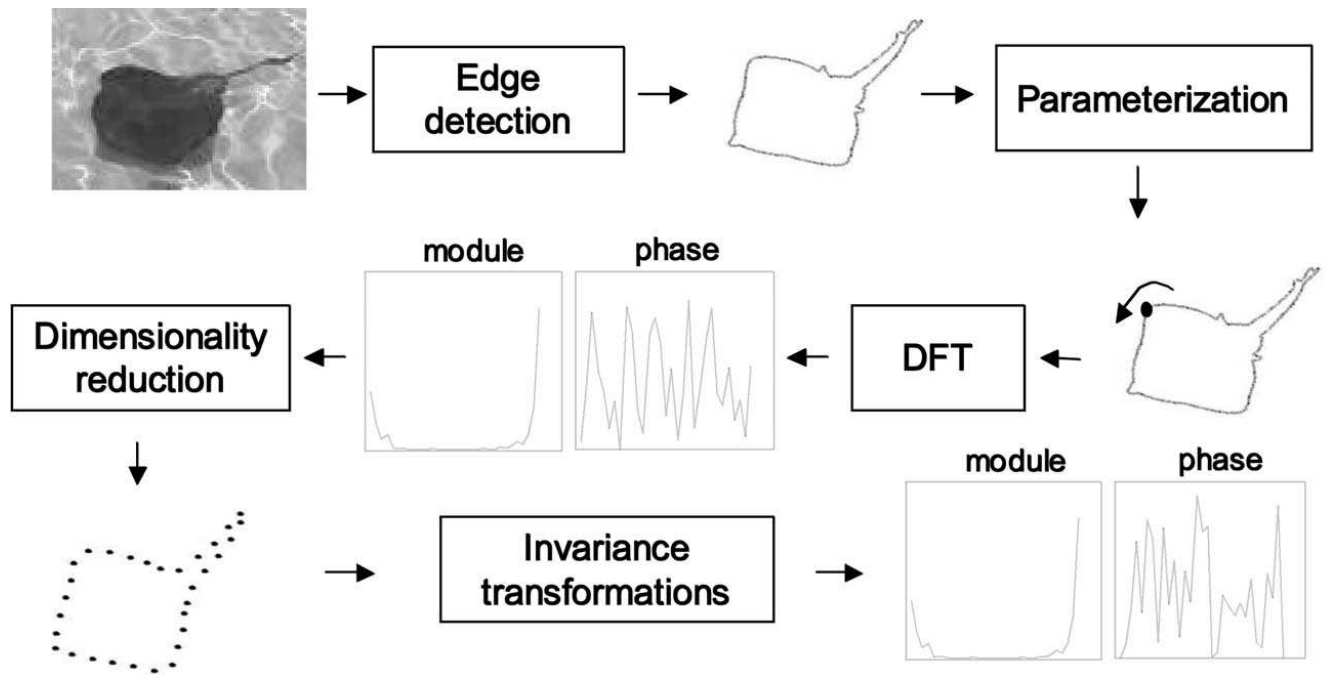
Vaizdo atpažinimo sistemose naudojama daug įvairių metodų, kai kurie iš jų yra ^[3]:

- Forma paremtas atpažinimas;
- 1D metrologija – matavimas;
- 2D metrologija – kraštų radimas;
- 3D metrologija – stereo vaizdas;
- Sričių analizė (angl. Blob analysis);
- Morfologija;
- Brūkšninio kodo atpažinimas;
- Optinis simbolio atpažinimas (angl. *Optical Character Recognition – OCR*).

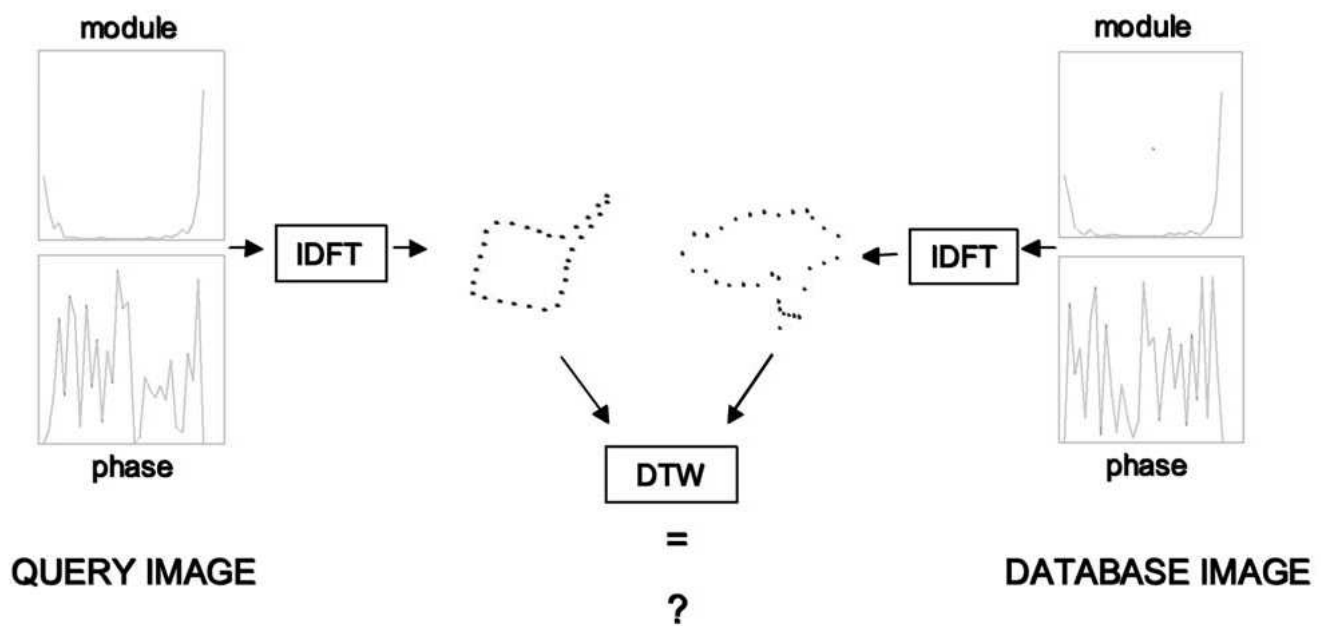
Šie būdai gali remtis tradicinėmis vaizdo apdorojimo metodikomis arba gali būti naudojami labai sudėtingi sprendimai sprendimų apie kokybę priėmimui.

2.2.1. Tikslus formų gavimas naudojant Furjė koeficientus ir laiko perstumimą.

WARP – Ilaria Bartolini pasiūlytas naujas Furjė transformacija paremtas metodas formų gavimui ir palyginimui ^[4]. Specifinis šio metodo bruožas – naudojama Furjė koeficientų fazė ir DTW (Dynamic Time Warping) atstumas. Fazės informacija duoda daug geresnę objekto kraštų aprašymą nei tik Furjė koeficientų amplitudė, o DTW dydis leidžia sulyginti vaizdus net esant mažiems fazės poslinkiams. Metodas leidžia palyginti atskirų objektų formas.



3 pav. WARP metodo žingsniai

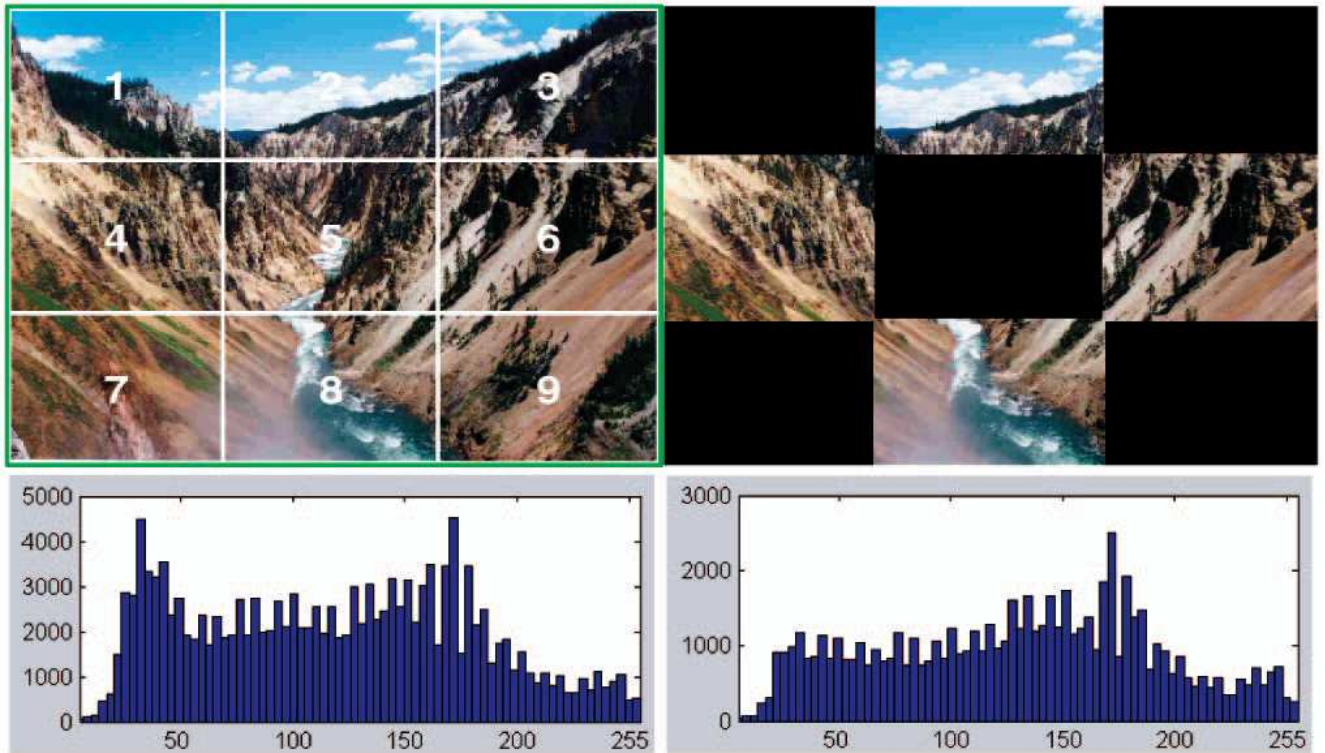


4 pav. Panašumo tarp dviejų formos aprašų įvertinimas WARP metodu

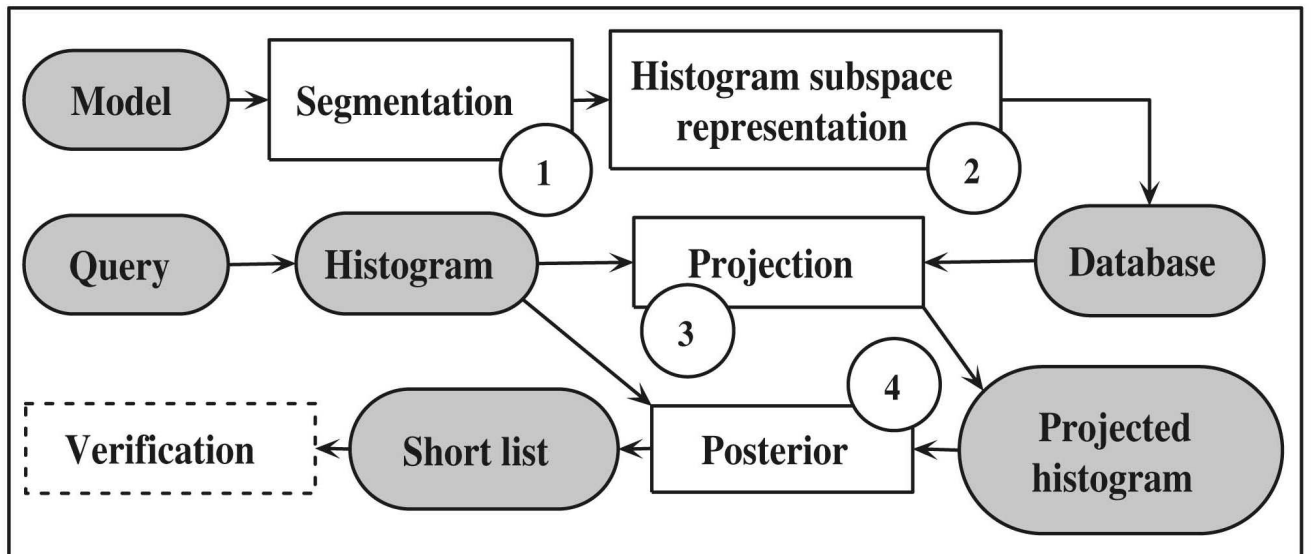
2.2.2. Formos žymių projekcijos ir palyginimas daliniam objektų atpažinimui.

Formos žymių histogramos arba prototipinės formos (*angl. prototypical shapes, shapemes*) plačiai naudojamos 2D/3D formų palyginimui ir atpažinimui [5]. Naudojant dalinai matomo objekto formos histogramas, galima jį atpažinti palyginus su duomenų bazėje esančiomis viso objekto formos histogramomis. Ying Shan siūlo aprašyti objektą keliomis formų histogramomis, ir atpažinimą vykdyti

dviem žingsniais: apskaičiuoti tiriamos histogramos projekciją ant visų modelio histogramų ir tada skaičiuoti atitikimo įvertį tarp tiriamos histogramos ir gautos projekcijos.



5 pav. Etaloninis (kairėje) ir tiriamas (dešinėje) vaizdas, bei jų intensyvumų histogramos



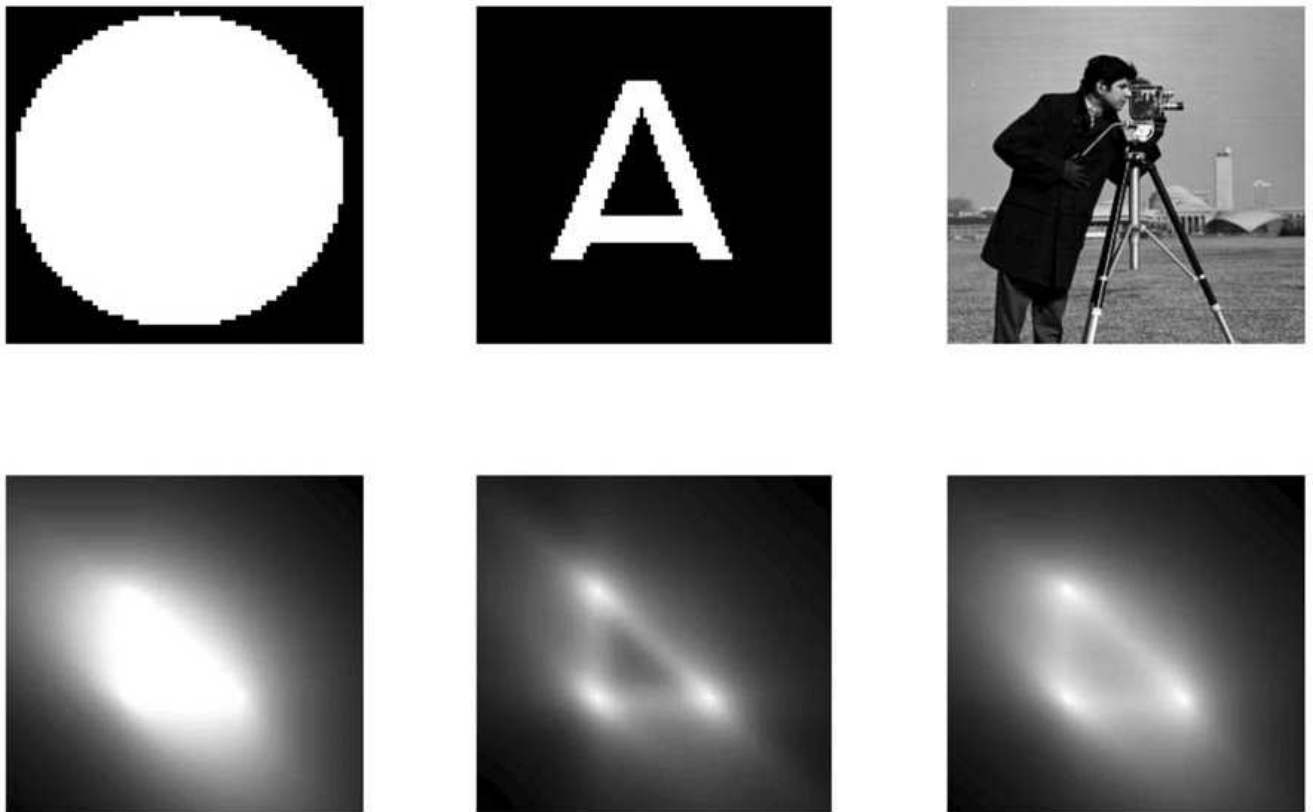
6 pav. Ying Shan siūlomo metodo schema

Bloke nr.1 (6 pav.) tiriamas vaizdas suskaidomas į atskiras dalis, ir tų dalių histogramos apskaičiuojamos bloke nr.2. Bloke nr.3 panaudojant duomenų bazėje esančių modelių histogramas

apskaičiuojamos gautų histogramų projekcijos. Rezultatai gaunami bloke nr.4. Taip pat sugeneruojamas neprivalomas kelių histogramų sąrašas patikrinimui.

2.2.3. Giminingų modelių atpažinimas naudojant kelių lygių autokonvoliuciją.

Objekto atpažinimui naudojama jo transformacija naudojant kelių lygių autokonvoliuciją (angl. multiscale autoconvolution – MSA) [6]. Siūlomas metodas pagrįstas tikimybinio vaizdo funkcijos įvertinimu. Metodas gali būti taikomas tiesiogiai izoliuotiems objektams ir nėra būtina išskirti objekto ribų ar reikšmingų taškų. Skaičiavimo greitis gana didelis, nes naudojama greita Furjė transformacija. Transformacijos koeficientai naudojami kaip objektų deskriptoriai jų klasifikacijai.



7 pav. Kelių objektų MSA transformacijos

2.3. Pasirinkta vaizdo apdorojimo metodika

Kadangi ankščiau minėti metodai skirti daugiau atskirų objektų atpažinimui bei palyginimui, yra gan sudėtingi, bei vaizdo atpažinimas vykdomas pakankamai ilgai, darbe pasirinkta naudoti tradicinius vaizdo apdorojimo algoritmus. Pasirinkti vaizdo apdorojimo algoritmai:

- Sobel operator
- Canny edge detector
- Laplace operator
- Eigen decomposition
- Corner detection
- Harris corner detector

Šie algoritmai vaizdo apdorojimo procese naudojami kaip priemonė sumažinti informacijos kiekį, t.y. naudojami vaizdo bruožų išskyrimui (*angl. Feature Extraction*). Gali būti išskiriami kraštai (*angl. Edge Detection*), sritys (*angl. Blob Detection*) ar kampai (*angl. Corner Detection*)^[7].

2.3.1. Sobel operatorius.

Sobel operatorius apskaičiuoja vaizdo ryškumo perėjimą (*angl. Gradient*) kiekviename taške kas parodo didžiausią galimą pokytį iš šviesaus į tamsų ir pokyčio intensyvumą ta kryptimi^[8]. Naudojamas 3x3 sąsūkų branduolių porą kuri naudojama apskaičiuoti išvestines horizontalia bei vertikalia kryptimi. Jei pradinį vaizdą pažymėsime A , o G_x ir G_y bus kiekviename taške apskaičiuotos išvestinės, tai jos apskaičiuojamos taip:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \text{ and } G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad (1)$$

Kiekviename vaizdo taške gaunamos pokyčio reikšmės apskaičiuojamos pagal formulę:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2)$$

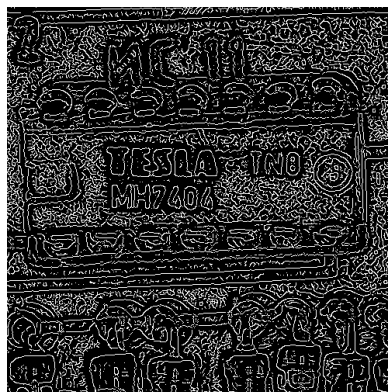


8 pav. Pradinis (dešinėje) ir Sobel operatoriumi apdorotas vaizdas

2.3.2. Canny edge detection.

Canny edge detection algoritmas naudoja kelių pakopų algoritmą vaizde esančių kraštų aptikimui ^[9]. Taip pat naudojami keletas reguliuojamų parametru (Gauso filtro dydis, slenksčiai), kurie turi įtakos skaičiavimo greičiui bei gaunamam rezultatui. Naudojamos pakopos:

- Triukšmo sumažinimas - vaizdas apdorojamos Gauso filtru.
- Ryškumo perėjimo radimas. Gali būti naudojami keli būdai vertikalių, horizontalių ir įstrižių kraštų radimui (pvz.: Roberts, Prewitt, Sobel).
- Ne maksimumų pašalinimas (*angl. non-maximum suppression*) – ieškoma ar pokyčio reikšmė yra lokalus maksimumas.
- Kraštų suradimas nustatant vaizdo ir histerezės slenkstinius lygius – reikalingi 2 slenksčiai: viršutinis ir žemutinis. Pritaikius viršutinį slenkstinį lygį, pažymimi kraštai, kurie gali būti tikri. Pradedant jais ir naudojant ankščiau gautą informaciją, randami visi vaizdo kraštai. Apatinis slenkstinis lygis leidžia surasti silpnus kraštus ir jie atmetami.



9 pav. Canny algoritmu apdorotas vaizdas

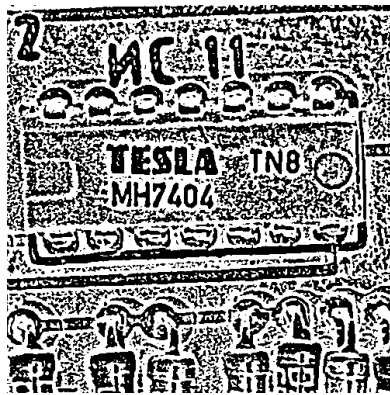
2.3.3. Laplace operatorius

Vienas iš pirmųjų ir dažniausiai naudojamų sričių detektorių yra Laplace operatorius (*angl. Laplacian of the Gaussian – LoG*)^[10]. Sričių detektoriai randa vaizdo vietas, kurios yra šviesesnės arba tamsesnės nei aplink esanti regionas. *Laplace* operatorius yra antros eilės diferencialinis operatorius n matavimų erdvėje, nusakomas kaip perėjimo nukrypimas (*angl. divergence*):

$$\Delta = \nabla^2 = \nabla \cdot \nabla \quad (3)$$

ekvivalentiškai, *Laplace* operatorius yra suma antros eilės dalinių išvestinių:

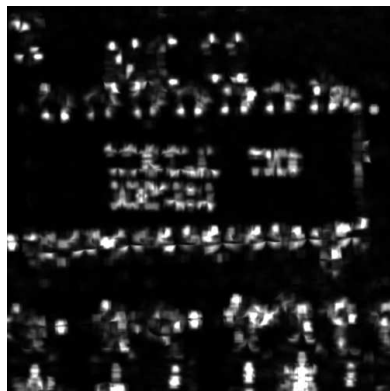
$$\Delta = \sum_{i=1}^n \frac{\partial^2}{\partial x_i^2} \quad (4)$$



10 pav. Laplace operatoriumi apdorotas vaizdas

2.3.4. Eigen decomposition metodas

Eigen decomposition - tai apdorojamos matricos suskaidymas į kanoninę formą, kur ji nusakoma tikrinėmis vertėmis (*angl. eigenvalues*) ir tikriniais vektoriais (*angl. eigenvectors*). Šiame darbe kaip charakteringos vaizdo vietos naudojamos tikrinės vertės.



11 pav. Eigen decomposition metodu apdorotas vaizdas

2.3.5. Corner detection

Corner detection algoritmas arba labiau bendras terminas – reikšmingų taškų suradimas (*angl. interest point detection*) yra kompiuterinės regos sistemose naudojamas būdas nustatyti tam tikros rūšies požymius apdorojamuose vaizduose. Bendru atveju kampų suradimo algoritmas (*The Moravec corner detection algoritmas*) testuoja kiekvieną vaizdo tašką, o kampas gali būti nusakytas kaip dviejų kraštų susikirtimas. Paprasčiausias būdas surasti vaizdo kampus, yra naudoti formulę:

$$D_x^2 D_{yy} + D_y^2 D_{xx} - 2D_x D_y D_{xy} \quad (5)$$

kur D_x – pirmoji vaizdo taško išvestinė, D_y – antroji vaizdo taško išvestinė. Kraštai gaunami kaip lokalūs funkcijos maksimumai.



12 pav. Corner detection algoritmu apdorotas vaizdas

2.3.6. Harris corner detector

Harris ir *Stephens* patobulino šį algoritmą įprastą kampų radimo algoritmą ^[11]. Jei pradinis vaizdas yra I , tai paėmus vaizdo fragmentą iš srities (u,v) ir paslinkus per (x,y) , gaunama kvadratinių skirtumų suma tarp šių dviejų sričių, pažymėta S , yra gaunama pagal formulę:

$$S(x, y) = \sum_u \sum_w w(u, w) (I(u, v) - I(u - x, v - y))^2 \quad (6)$$

Harris matrica A yra gaunama suapvalinus S naudojant antros eilės Teiloro eilutės skleidinį:

$$S(x, y) \approx S(0,0) + (x, y) \nabla S + \frac{1}{2} (x, y) A \begin{pmatrix} x \\ y \end{pmatrix} \quad (7)$$

kur ∇S ir A žymi perėjimo vektorių ir antrųjų S išvestinių matricas.

Kampas yra aprašomas kaip didelis S nukrypimas (*angl. variation*) visomis vektoriaus (x, y) kryptimis. Analizuojant tikrines A vertes, šis aprašymas gali būti išreikštas taip: A turi turėti 2 dideles tikrines vertes reikšmingame taške. Atsižvelgiant į tikrinių verčių dydžius, galima daryti tokias išvadas:

- jei $\lambda_1 \approx 0$ ir $\lambda_2 \approx 0$ tai tiriamoje vietoje kampo nėra;
- jei $\lambda_1 \approx 0$ o λ_2 yra teigiama reikšmė – tiriamoje vietoje rastas kraštas;
- jei λ_1 ir λ_2 yra didelės, teigiamos ir skirtingos reikšmės – tiriamoje vietoje rastas kampas;

Tikslus tikrinių verčių apskaičiavimas užima daug laiko, todėl *Harris corner detector* algoritme naudojama sekanti funkcija M_c , kur κ yra reguliuojamas jautrumo parametras:

$$M_c = \lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 = \det(A) - \kappa \text{trace}^2(A) \quad (8)$$

κ reikšmė turi būti nustatyta bandymų būdu. Literatūroje kaip galima šio parametro reikšmė nurodoma 0.04 - 0.15 ^[12].



13 pav. Harris corner detection algoritmu apdorotas vaizdas

2.3.7. Vaizdų palyginimas

Tiriamų vaizdų palyginimas atliekamas naudojant normalizuotą koreliaciją ^[16]. Koreliacijos funkciją vaizdai galima apskaičiuoti naudojant šią lygtį:

$$\gamma(u, v) = \frac{\sum_x \sum_y [f(x, y) - \bar{f}_{u,v}] [t(x-u, y-v) - \bar{t}]}{\sqrt{\sum_x \sum_y [f(x, y) - \bar{f}_{u,v}]^2 \sum_x \sum_y [t(x-u, y-v) - \bar{t}]^2}} \quad (9)$$

Koreliacinė funkcija parodo ar vaizdai yra panašūs, ir kiek jie panašūs. Tai parodo koreliacinės funkcijos reikšmė. Jei normalizuotos koreliacinės funkcijos reikšmė lygi 1 arba -1, tai sakoma, kad yra idealus vaizdų sutapimas. Laikoma, kad patikimai atpažinta, kai normalizuotos koreliacijos reikšmė yra $\geq 0,9$ arba $\leq -0,9$.

3. SPAUSDINTINIO MONTAŽO PLOKŠČIŲ KOKYBĖS ĮVERTINIMO SISTEMA

3.1. Techninė įranga

Gavus finansavimą iš TARPTAUTINIŲ MOKSLO IR TECHNOLOGIJŲ PLĖTROS PROGRAMŲ AGENTŪROS iš projekto E13807 – „EKO-FACTORY: Intelektuali konvejerinės gamybos identifikavimo sistema” lėšų remiantis AOK koncepcija, buvo suprojektuotas ir pagamintas spausdinto montažo plokščių kokybės tikrinimo stendo prototipas (14 pav.).

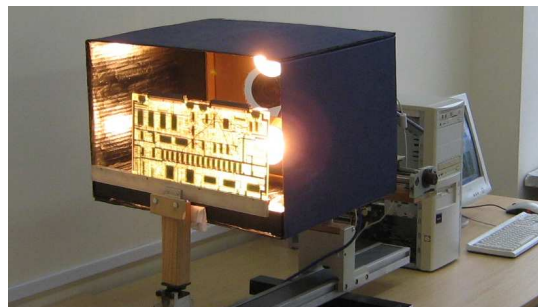


14 pav. Suprojektuotas vaizdų analizės stendas

Stendo veikimo principas:

- gaminys patalpinamas į specialų rėmelį, galima keisti jo atstumą iki kameros, pasisukimo kampą.
- Vaizdui gauti naudojama buitinė didelės skiriamosios gebos CANON EOS 400D fotokamera, prie kompiuterio prijungta USB sąsaja.
- Gautam vaizdui apdoroti naudojamas kompiuteris su specialiai gaminio kokybės tikrinimui sukurta programine įranga.

Norint gauti kokybišką tiriamo objekto vaizdą, būtina sumažinti aplinkos įtaką, todėl reikalingas geras papildomas apšvietimas. Atliekant tiriamuosius darbus su stendu, naudoti šviesos diodų (*angl. LED*) ir kaitrinių lempų (Philips Spotone 100W E27) apšvietimo moduliai (15 pav.).



15 pav. Šviesos diodų (kairėje) ir kaitrinių lempų (dešinėje) apšvietimo moduliai

3.2. Programinė įranga

Kuriant gaminių defektų identifikavimo sistemos prototipą, buvo suprojektuota ir parašyta speciali programinė įranga. Programinė įranga parašyta C++ programavimo kalba, naudotas nemokamas wxDev-C++ kompiliatorius^[17].

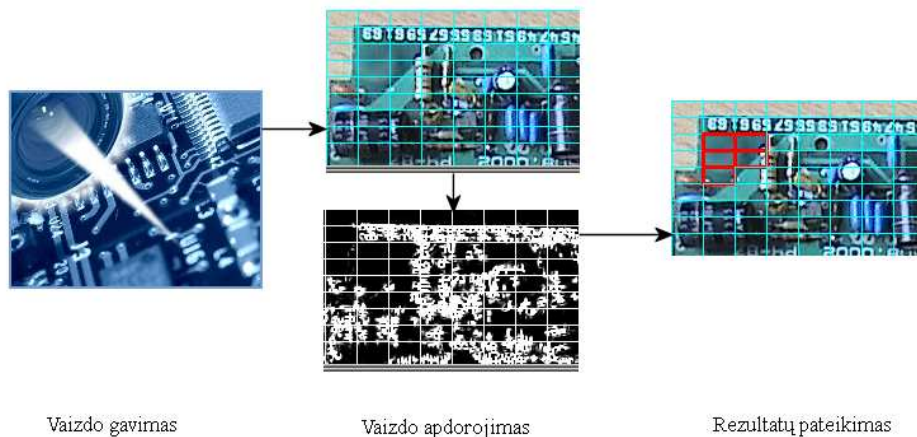
Programinės įrangos veikimo principas: kamera nufotografuoja tiriamos plokštės vaizdą, vaizdas apdorojamas pasirinktu algoritmu ir pateikiami tyrimo rezultatai. Radus neatitikimą, sistema perspėja operatorių apie tiriamos plokštės defektą, išsaugo ir pateikia plokštės atvaizdą su pažymėtomis defektų vietomis.

Programinę įrangą galima suskirstyti į 3 modulius (16 pav.):

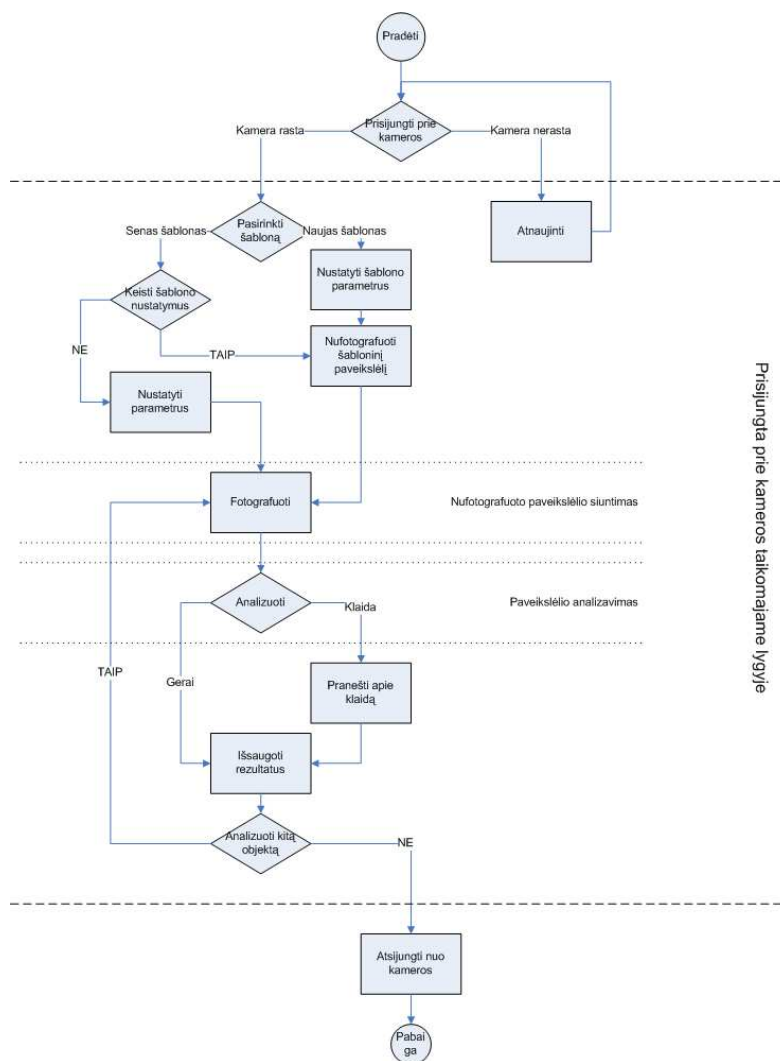
1. Vaizdo gavimas. Tai sąsaja su išoriniais įrenginiais – kameromis. Apima prisijungimą prie kamerų, jų parametrų nustatymą, nuotolinį fotografavimą, gauto vaizdo parsisiuntimą į kompiuterį apdorojimui.
2. Vaizdo apdorojimas. Apima vaizdo apdorojimo algoritmo pasirinkimą, jo parametrų nustatymą, vaizdo apdorojimą, defektų atpažinimą.
3. Rezultatų pateikimas. Tai sąsaja su vartotoju.

Startavus programą, jungiamasi prie kamerų (17 pav.). Jei jos neprijungtos prie kompiuterio, yra išjungtos, ar dėl kitų priežasčių neįmanoma prisijungti, vartotojas apie tai informuojamas. Sėkmingai prisijungus, užkraunami nustatymai. Kadangi naudojantis sistema bus norima tirti ne vieną gaminį, numatyta galimybė kiekvienam gaminiui parinkti skirtingus kamerų, vaizdo apdorojimo ir kitus nustatymus. Programai startavus, užkraunami vėliausiai naudoto šablono nustatymai. Jei norima tirti kitą gaminį, pasirenkamas jau sukurtas jo šablonas, arba sukuriamas naujas. Sukūrus naują

šabloną, pasiūlomi standartiniai nustatymai, kuriuos vėliau galima pakeisti. Taip pat pasiūloma nufotografuoti šabloninį naujo gaminio vaizdą, pagal kurį bus lyginami kiti gaminiai.



16 pav. Programinės įrangos moduliai



17 pav. Programinės įrangos blokinė schema

Nufotografuotas vaizdas iš laikinosios kameros atmintinės parsiuočiomas į kompiuterį. Kiekvienam gautam vaizdai suteikiamas unikalus vardas: *camID_time_imgName*.

camID – kameros identifikacinis numeris, skiriasi kiekvienoje kameroje.

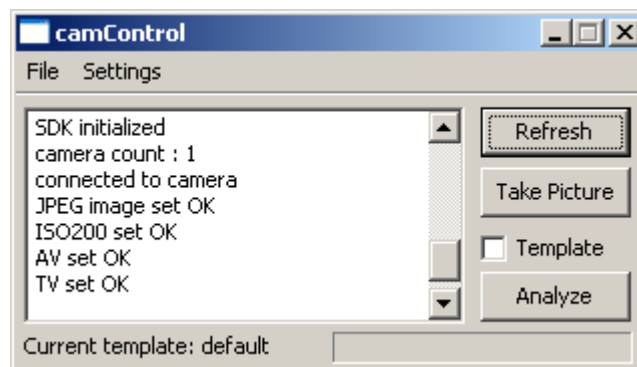
time – laikas Unix Timestamp formatu (nuo 1970 m. Sausio 1d. 00:00 val. praėjęs sekundžių skaičius) ^[18].

imgName – kameros suteiktas originalus paveikslėlio vardas.

Naujai gautas vaizdas, palyginamas su prieš tai gautu šablonu. Radus neatitikimų, vartotojas apie tai informuojamas. Analizės rezultatai Išsaugomi HTML formatu.

Baigus darbą su programa, atsijungiama nuo kameros.

Sąsaja su kamera realizuota naudojant EDSDK 2.0 biblioteką. Tai biblioteka, suteikianti galimybę nesudėtingai prisijungti prie CANON skaitmeninių kameros, jas valdyti, keisti duomenimis tarp kameros ir kompiuterio ^[19]. Sukurta programinė įranga leidžia nuotoliniu būdu fotografuoti, pasirinkti kameros jautrumą (*angl. ISO Speed*), išlaikymo (*angl. Shutter Speed*) ir diafragmos atvėrimo (*angl. Apperture Value*) parametrus.

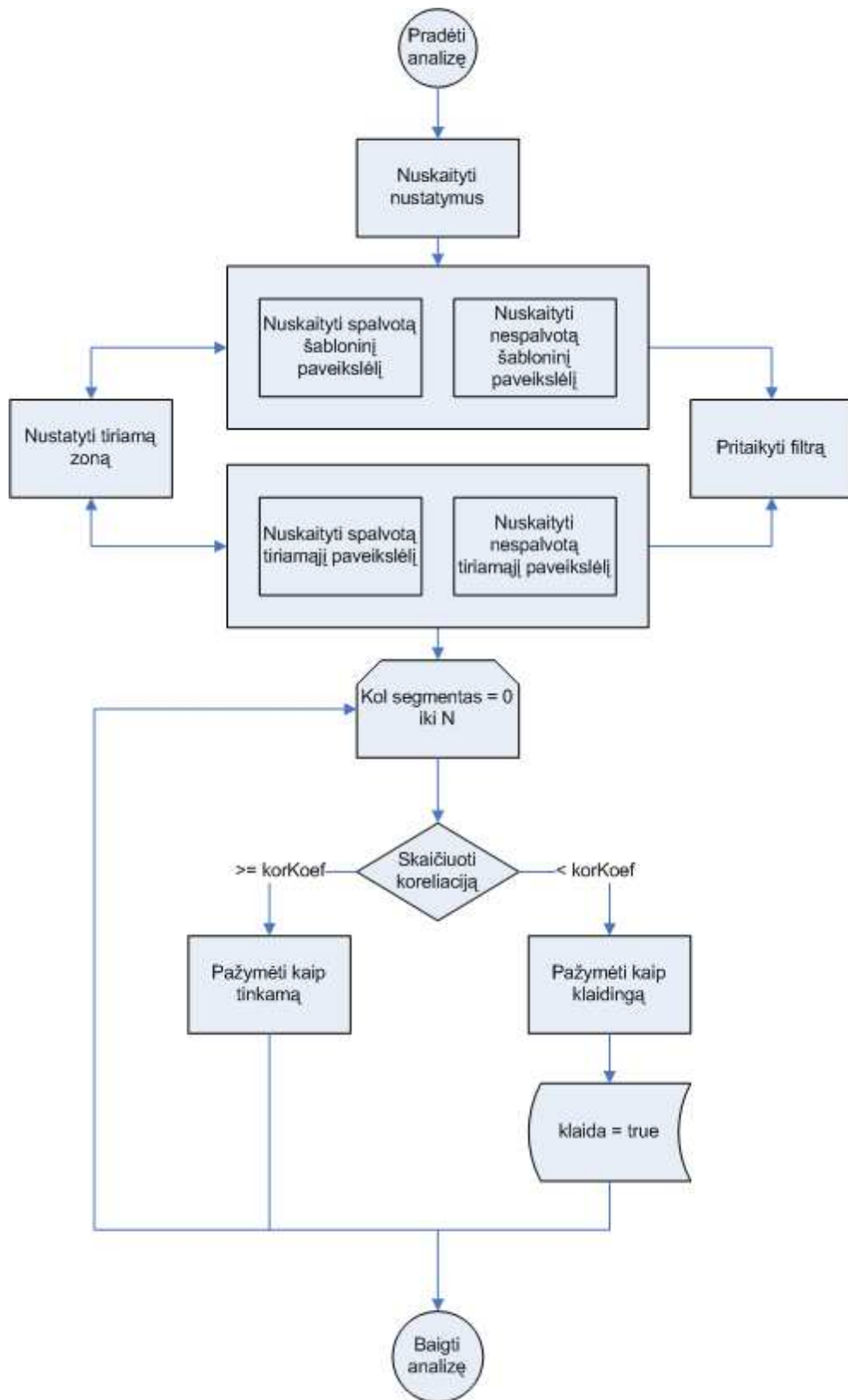


18 pav. Pagrindinis programinės įrangos langas

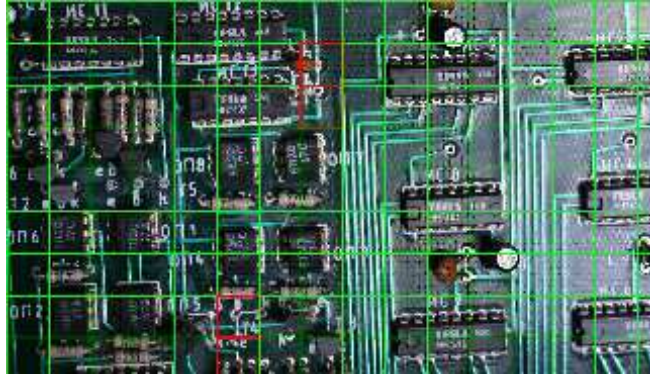
Vaizdo apdorojimo algoritmas (19 pav.):

- Gaunamas etaloninis gaminio vaizdas
- Etaloninis vaizdas apdorojamas pasirinktu vaizdo apdorojimo algoritmu
- Gaunamas tiriamo gaminio vaizdas
- Tiriamas vaizdas apdorojamas tokiu pat vaizdo apdorojimo algoritmu kaip ir etaloninis vaizdas
- Pagal parinktą segmento dydį, nustatoma segmento vieta abiejuose vaizduose ir suskaičiuojamas koreliacijos koeficientas

- Palyginus su nustatyta koreliacijos koeficiento reikšme, segmentas pažymimas kaip tinkamas (žaliai) arba kaip klaidingas (raudonai) (20 pav.).



19 pav. Vaizdo analizės algoritmas



20 pav. Gauto vaizdo analizės rezultatas

Naudojamoje programinėje įrangoje, koreliacija atliekama naudojant OpenCV bibliotekos funkciją *cvMatchTemplate*.

Kameros bei vaizdo apdorojimo algoritmų parametrai saugomi config.ini faile, *INI* (*initialization file*)^[20] formatu, jo palaikymas realizuotas naudojant *CIniFile* klasę^[21].

4. SISTEMOS EFEKTYVUMO TYRIMAI

Ekspperimentai vykdomi tokia tvarka: imama testuojama plokštė, ji patalpinama į rėmelį ir nufotografuojama 2 kartus. Pirmą nuotrauką naudojamas kaip etaloninis gaminio vaizdas, antra – kaip gaminio be defektų vaizdas. Po to iš plokštės pašalinami keli elementai (rezistoriai, kondensatoriai ir kt.), plokštė vėl patalpinama į rėmelį ir dar kartą nufotografuojama. Taip gaunamas plokštės su defektais vaizdas. Fotografavimas ir gautų vaizdų analizavimas vykdomas naudojant sukurta programinę įrangą, t.y. nuotoliniu būdu. Kameros parametrai taip pat keičiami nuotoliniu būdu, taip imituojant realios sistemos darbą (kai vartotojas tiesiogiai nekontaktuoja su technine įranga).

Grafikuose x ašyje atidedami algoritmo parametrai, y ašyje – klaidingai identifikuotų segmentų (segmentų, kuriuose realaus defekto nebuvo, tačiau programinė įranga vis tiek jų identifiko kaip neatitinkantį etalono) skaičius procentais. Jis apskaičiuojamas pagal formulę:

$$rez = \frac{(def - def_err)}{segx * segy} * 100, \text{ kur} \quad (10)$$

def – neatitinkančių segmentų kiekis;

def_err – neatitinkančių segmentų, kurie identifikuoja plokštės defektą kiekis;

$segx, segy$ – segmentų skaičius horizontalia ir vertikalio kryptimi.

4.1. Fotografavimo režimo įtaka vaizdo analizei.

Kadangi visi pasirinkti vaizdo apdorojimo algoritmai apdoroja pilkus (*angl. grey scale*) vaizdus, tikrinama prielaida, kad geriau nustatyti kameras į nespaltoto fotografavimo režimą. Tiriama plokštė be realių defektų, todėl teoriškai visi segmentai turėtų būti identifikuoti kaip atitinkantys šabloną. Tyrimui naudojami šie vaizdo apdorojimo algoritmai: *Sobel operator*, *Canny edge detector*, *Corner detection* ir *Harris corner detector*. Taip pat naudojami skirtingi segmento dydžiai (50x50 px ir 75x75 px) ir skirtingi koreliacijos koeficientai (0.9 ir 0.8).

Naudojant *Sobel* algoritmą, klaidingai identifikuotų segmentų skaičius buvo didelis (1 lentelė). Tik esant tam tikriems algoritmo parametrams, visi segmentai identifikuoti teisingai. Mažiau klaidų daroma buvo parinkus didesnę segmentą ir mažesnę koreliacijos koeficientą. Segmentai spalvotame paveikslėlyje buvo identifikuojami tiksliau.

1 lentelė. Neatitinkančių segmentų skaičius spalvotame vaizde naudojant skirtingus Sobel operatoriaus nustatymus, kai segmento dydis 50x50px ir koreliacijos koeficientas 0.8

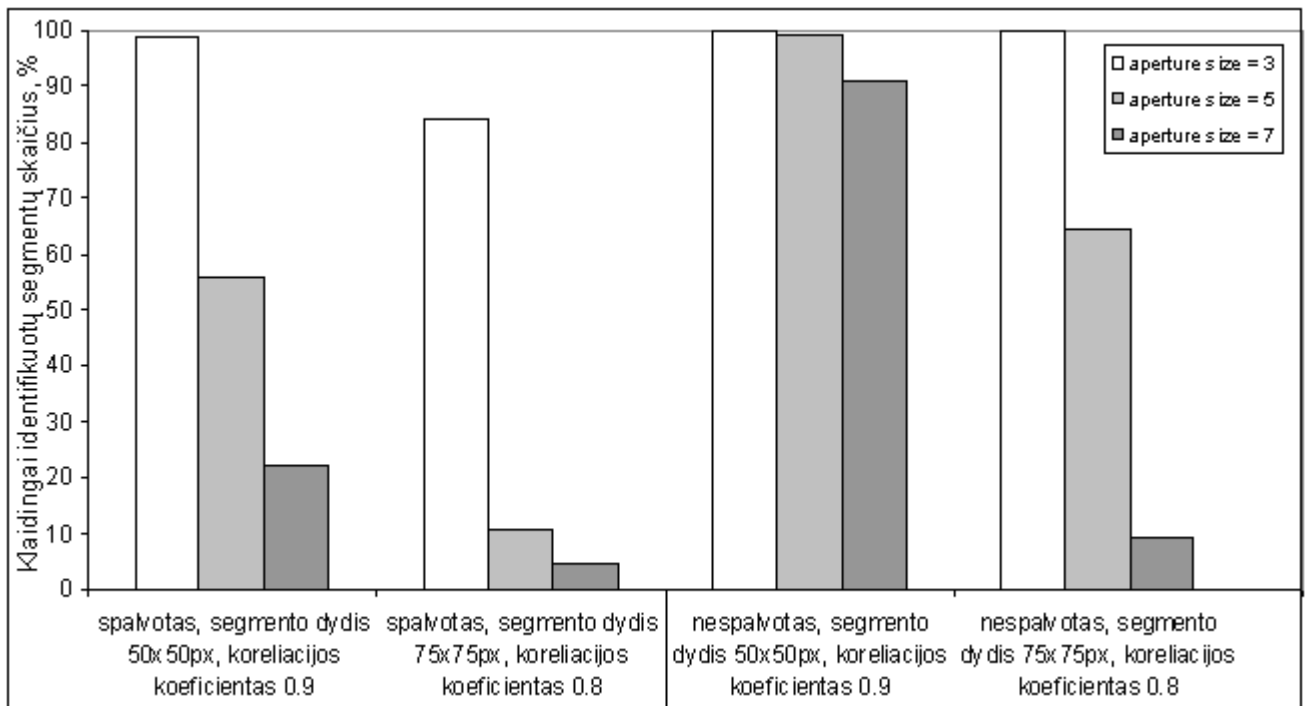
| Algoritmo nustatymai | Neatitinkančių segmentų skaičius | | Neatitinkančių segmentų skaičius, % | |
|------------------------------------|----------------------------------|-------------|-------------------------------------|-------------|
| | Spalvotas | Nespalvotas | Spalvotas | Nespalvotas |
| xorder=0, yorder=1, apertureSize=3 | 76 | 247 | 19,00 | 61,75 |
| xorder=0, yorder=1, apertureSize=5 | 39 | 66 | 9,75 | 16,50 |
| xorder=0, yorder=1, apertureSize=7 | 31 | 45 | 7,75 | 11,25 |
| xorder=0, yorder=2, apertureSize=3 | 400 | 400 | 100,00 | 100,00 |
| xorder=0, yorder=2, apertureSize=5 | 204 | 400 | 51,00 | 100,00 |
| xorder=0, yorder=2, apertureSize=7 | 100 | 397 | 25,00 | 99,25 |
| xorder=1, yorder=0, apertureSize=3 | 64 | 103 | 16,00 | 25,75 |
| xorder=1, yorder=0, apertureSize=5 | 40 | 45 | 10,00 | 11,25 |
| xorder=1, yorder=0, apertureSize=7 | 30 | 34 | 7,50 | 8,50 |
| xorder=1, yorder=1, apertureSize=3 | 400 | 400 | 100,00 | 100,00 |
| xorder=1, yorder=1, apertureSize=5 | 253 | 400 | 63,25 | 100,00 |
| xorder=1, yorder=1, apertureSize=7 | 130 | 342 | 32,50 | 85,50 |
| xorder=1, yorder=2, apertureSize=3 | 400 | 400 | 100,00 | 100,00 |
| xorder=1, yorder=2, apertureSize=5 | 400 | 400 | 100,00 | 100,00 |
| xorder=1, yorder=2, apertureSize=7 | 344 | 400 | 86,00 | 100,00 |
| xorder=2, yorder=0, apertureSize=3 | 400 | 400 | 100,00 | 100,00 |
| xorder=2, yorder=0, apertureSize=5 | 217 | 376 | 54,25 | 94,00 |
| xorder=2, yorder=0, apertureSize=7 | 108 | 208 | 27,00 | 52,00 |
| xorder=2, yorder=1, apertureSize=3 | 400 | 400 | 100,00 | 100,00 |
| xorder=2, yorder=1, apertureSize=5 | 400 | 400 | 100,00 | 100,00 |
| xorder=2, yorder=1, apertureSize=7 | 359 | 400 | 89,75 | 100,00 |
| xorder=2, yorder=2, apertureSize=3 | 400 | 400 | 100,00 | 100,00 |
| xorder=2, yorder=2, apertureSize=5 | 400 | 400 | 100,00 | 100,00 |
| xorder=2, yorder=2, apertureSize=7 | 400 | 400 | 100,00 | 100,00 |

Canny algoritmas pasirodė netinkamas defektų atpažinimui, nes buvo neteisingai identifikuojama nuo 80% iki 100% segmentų (2 lentelė- rezultatai, kai segmento dydis 50x50 pikselių ir koreliacijos koeficientas 0.9). Tai parodo, jog *Canny* algoritmas labai jautrus net mažiausiems pokyčiams. Nežymūs fotografuojamo vaizdo pakitimai galėjo atsirasti dėl netolygaus bendrojo apšvietimo, ir dėl to, kad norint gauti pakankamai ryškų ir šviesų vaizdą, buvo padidintas kameros jautrumas. Geresni rezultatai gaunami tiriant spalvotą vaizdą.

Naudojant *Corner detection* algoritmą taip pat didelė dalis segmentų buvo identifikuoti kaip neatitinkantys etalono (21 pav). Geresni rezultatai gauti tiriant spalvotą vaizdą, bei imant didesnę segmento dydį, bei mažesnę koreliacijos koeficientą. Tiriant spalvotą vaizdą, parametrui *aperture size* esant 7, segmento dydžiui 75x75px ir koreliacijos koeficientui 0.8, klaidingai identifikuotų segmentų skaičius buvo ~4,5 %.

2 lentelė. Neatitinkančių segmentų skaičius spalvotame vaizde naudojant skirtingus Canny algoritmo nustatymus, kai segmento dydis 50x50px ir koreliacijos koeficientas 0.9

| Algoritmo nustatymai | | | Neatitinkančių segmentų skaičius | | Neatitinkančių segmentų skaičius, % | |
|----------------------|--------|---------------|----------------------------------|-------------|-------------------------------------|-------------|
| xorder | yorder | aperture size | Spalvotas | Nespalvotas | Spalvotas | Nespalvotas |
| 0 | 1 | 3 | 76 | 247 | 19 | 61,75 |
| | | 5 | 39 | 66 | 9,75 | 16,5 |
| | | 7 | 31 | 45 | 7,75 | 11,25 |
| | 2 | 3 | 400 | 400 | 100 | 100 |
| | | 5 | 204 | 400 | 51 | 100 |
| | | 7 | 100 | 397 | 25 | 99,25 |
| 1 | 0 | 3 | 64 | 103 | 16 | 25,75 |
| | | 5 | 40 | 45 | 10 | 11,25 |
| | | 7 | 30 | 34 | 7,5 | 8,5 |
| | 1 | 3 | 400 | 400 | 100 | 100 |
| | | 5 | 253 | 400 | 63,25 | 100 |
| | | 7 | 130 | 342 | 32,5 | 85,5 |
| | 2 | 3 | 400 | 400 | 100 | 100 |
| | | 5 | 400 | 400 | 100 | 100 |
| | | 7 | 344 | 400 | 86 | 100 |
| 2 | 0 | 3 | 400 | 400 | 100 | 100 |
| | | 5 | 217 | 376 | 54,25 | 94 |
| | | 7 | 108 | 208 | 27 | 52 |
| | 1 | 3 | 400 | 400 | 100 | 100 |
| | | 5 | 400 | 400 | 100 | 100 |
| | | 7 | 359 | 400 | 89,75 | 100 |
| | 2 | 3 | 400 | 400 | 100 | 100 |
| | | 5 | 400 | 400 | 100 | 100 |
| | | 7 | 400 | 400 | 100 | 100 |



21 pav. Klaidingai identifikuotų segmentų skaičius naudojant *Corner detection* algoritmą su įvairiais parametrais

Naudojant *Harris corner detector* algoritmą nespaltvotame vaizde, klaidingai identifikuota iki 97% segmentų. Tiriant spalvotą vaizdą, klaidingai identifikuota iki 48% segmentų. Esant kai kuriems algoritmo parametrams, klaidingai identifikuota ~1% segmentų spalvotame ir ~3% nespaltvotame vaizde.

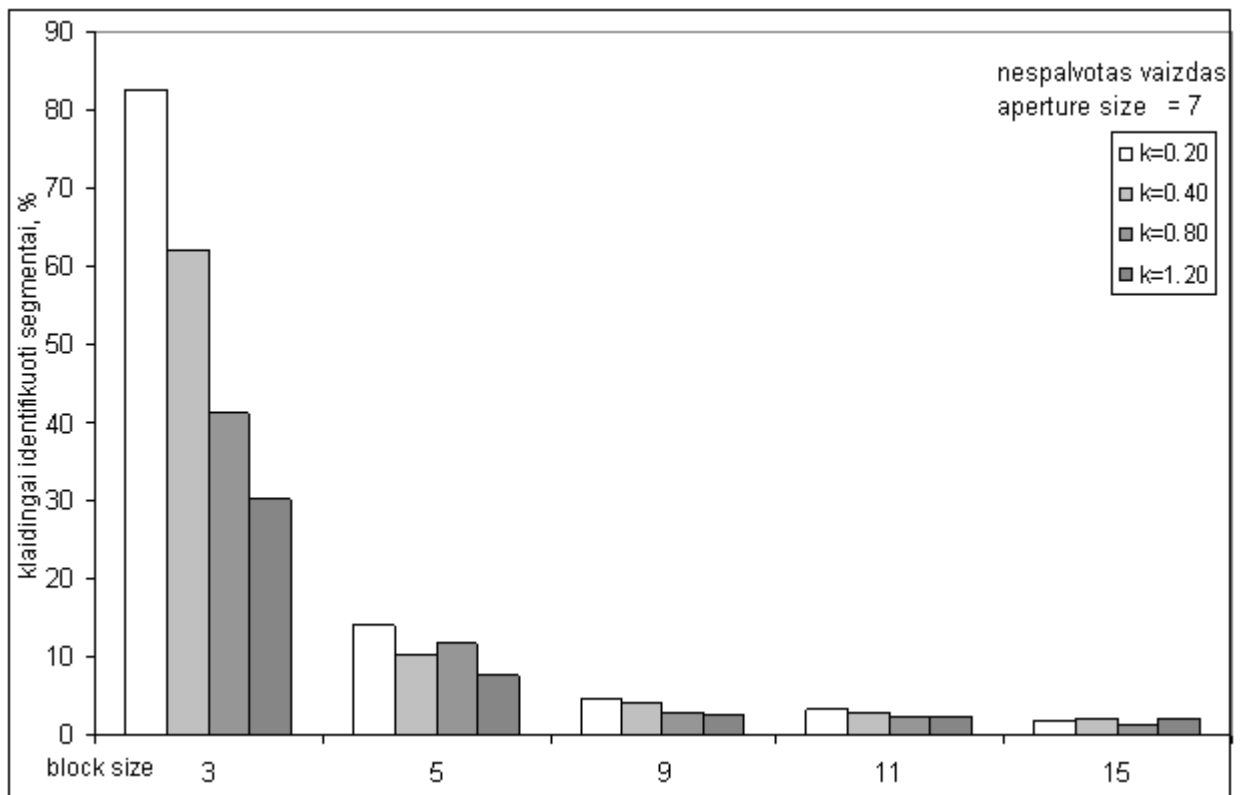
22 pav. ir 23 pav. pavaizduoti tyrimo rezultatai kai naudojami šie algoritmo parametrai:

aperture size: 7;

block size: 3, 5, 9, 11, 15;

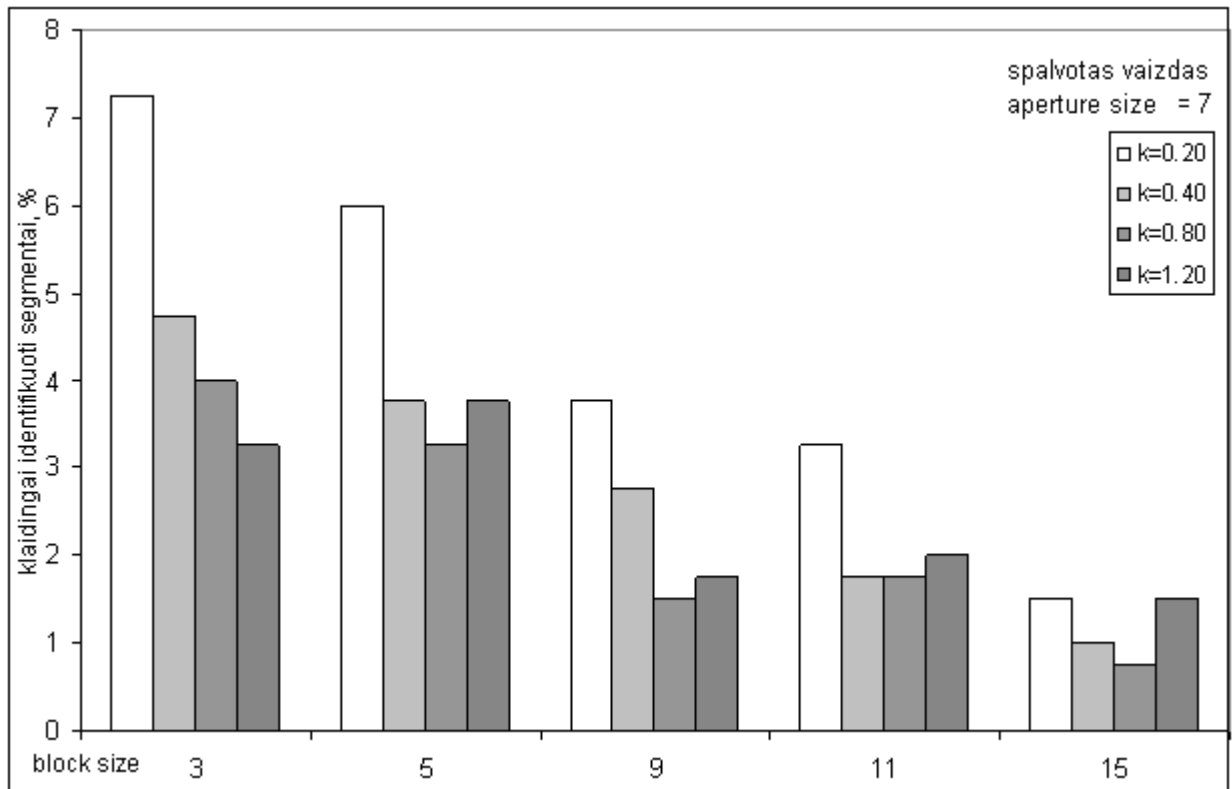
k: 0.2, 0.4, 0.8, 1.2.

Naudojamas segmento dydis – 50x50 pikselių, koreliacijos koeficientas – 0.9.

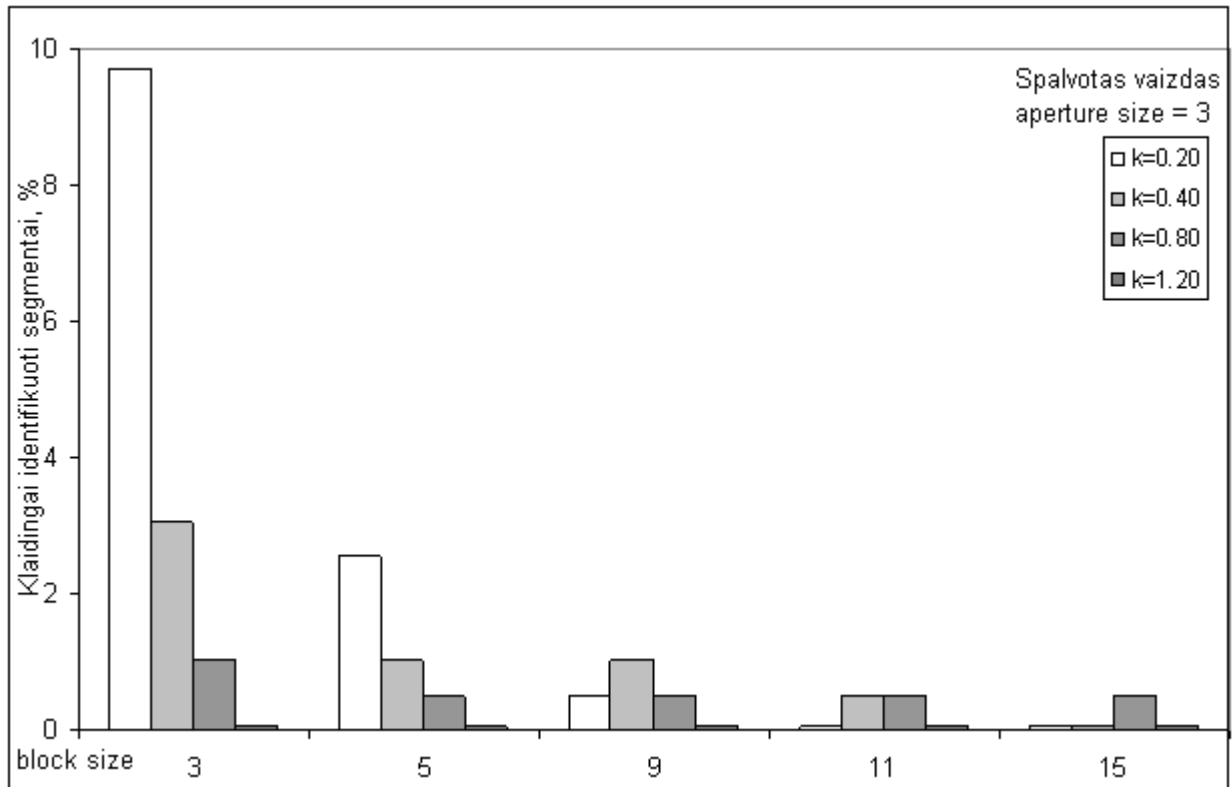


22 pav. Rezultatai su nespaltvotu vaizdu naudojant *Harris corner detector* algoritmą

Padidinus segmento dydį ir sumažinus koreliacijos koeficientą, esant kai kuriems parametrams ir spalvotame ir nespaltvotame vaizde visi segmentai identifikuoti teisingai (24 pav.).



23 pav. Rezultatai su spalvotu vaizdu naudojant *Harris corner detector* algoritmą



24 pav. Rezultatai su *Harris corner detection* algoritmu, kai segmento dydis 75x75 px, koreliacijos koeficientas – 0.8

4.2. Kameros parametrų įtaka analizės rezultatams.

Pagrindiniai fotokameros parametrai kurie turi įtakos gaunamo vaizdo kokybei yra išlaikymas, diafragmos atvėrimas ir jautrumas ^[22]. Didinant išlaikymo parametą, į kameros vaizdo sensorių patenka daugiau šviesos, todėl vaizdas gaunamas šviesesnis. Tačiau galimi pašaliniai triukšmai. Tam kad gauti vienodo šviesumo vaizdą, didinant išlaikymo parametą, buvo mažinamas diafragmos atvėrimas. Šių parametrų įtakos sistemos efektyvumui nepastebėta.

Paveiksluojant tą patį vaizdą fotokamera duoda geros kokybės, tačiau šiek tiek skirtingus vaizdus, todėl atliekant vaizdų analizę, kai kurie vaizdų apdorojimo algoritmai juos traktuodavo kaip skirtingus. Pastebėta, kad tam įtakos turi automatinė CANON fotokameros fokusavimo ir vaizdo stabilizavimo sistemos, bei nustatytas jautrumas, todėl fotografuojant reikėtų naudoti rankinį režimą, o kameros parametrus nustatyti taip, kad būtų gaunama kuo mažiau pašalinių triukšmų.

Eksperimentų metu kai naudojamas papildomas apšvietimą, kameros parametrai nustatomi į tokius:

- išlaikymas – 1/125
- diafragmos atvėrimas – 5.6
- jautrumas - ISO200

Nenaudojant apšvietimo, kameros jautrumas padidinamas iki ISO800.

4.3. Optimalių algoritmų parametrų nustatymas.

Kaip matosi 22 pav., esant kai kuriems algoritmų parametrams, klaidingai identifikuotų segmentų skaičius sumažėja, todėl kiekvienam pasirinktam vaizdo apdorojimo algoritmui buvo parenkami įvairūs galimi parametrai, ir tiriama kurie iš jų yra optimaliausi.

Taip pat keičiamas segmento dydis (50x50, 75x75 ir 100x100), bei koreliacijos koeficientas (0.9 ir 0.8). Kadangi *Canny* algoritmas ankstesniame tyrime parodė didelį neteisingai identifikuotų segmentų skaičių, toliau jis eksperimentuose nenaudojamas.

Sobel metodas naudojamoje programinėje įrangoje realizuotas naudojant OpenCV funkciją *cvSobel*. Ji apskaičiuoja pirmos, antros, trečios arba mišrios eilės vaizdo išvestines naudodama *sobel* operatorių. Funkcijos prototipas:

```
void cvSobel( const CvArr* src, CvArr* dst, int xorder, int yorder, int aperture_size=3 );
```

src – pradinis vaizdas;
dst – apdorotas vaizdas;
xorder – išvestinės eilė x kryptimi
yorder – išvestinės eilė y kryptimi;
aperture size – sobel branduolio dydis. Gali būti 1, 3, 5 ir 7.

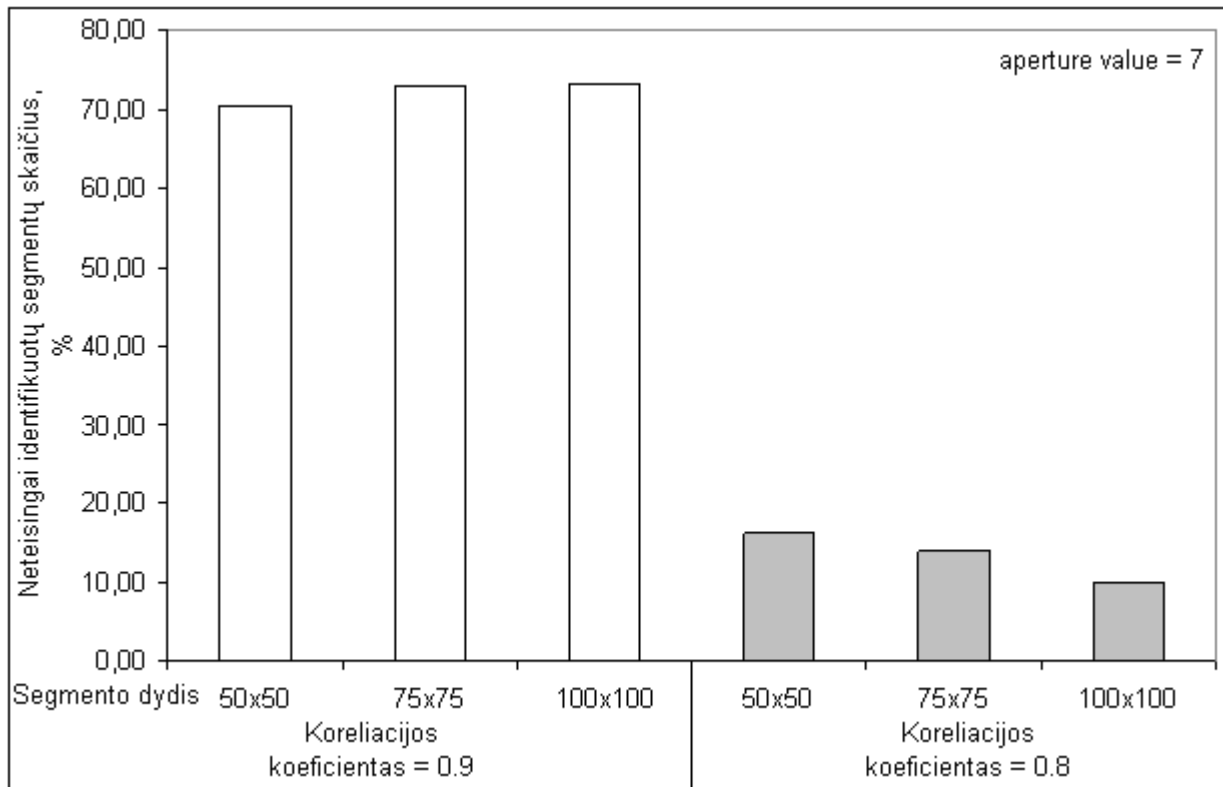
Sobel operatorius labai jautrus, todėl net esant mažiausiems neatitikimams, segmentas identifikuojamas kaip klaidingas. Norint pašalinti galimus triukšmus, prieš apdorojant vaizdą *Sobel* operatoriumi, prieš tai jis buvo apdorotas *Gaussian* filtru. Geriausi rezultatai gaunami kai xorder = 1, yorder = 0, apertureSize = 7. (1 lentelė) Taip pat naudojant šį operatorių reikėtų sumažinti koreliacijos koeficientą, bei padidinti segmento dydį.

Laplace operatorius realizuotas naudojant OpenCV funkciją *cvLaplace*. Jos prototipas:

```
void cvLaplace( const CvArr* src, CvArr* dst, int aperture_size=3 );
```

src – pradinis vaizdas;
dst – apdorotas vaizdas;
aperture size –branduolio dydis. Gali būti 1, 3, 5 ir 7.

Algoritmas labai jautrus bet kokiems pokyčiams, todėl nuo 10% iki 70% segmentų identifikuojami klaidingai. Geresni rezultatai gaunami esant mažesniai koreliacijos koeficientui, bei didesniai segmento dydžiui (25 pav.).



25 pav. Tyrimo rezultatai, kai *Laplace* operatoriaus parametras aperture size yra 7.

Eigen decomposition metodas realizuotas funkcija `cvCornerEigenMinVal`. Jos prototipas:

```
void cvCornerMinEigenVal( const CvArr* image, CvArr* eigenval, int block_size, int aperture_size=3 );
```

`image` – pradinis vaizdas;

`eigenval` – suskaičiuotos minimalios tikrinės vertės.

`block_size` - tyrimui naudojamo bloko dydis

`aperture size` –branduolio dydis. Gali būti 1, 3, 5 ir 7.

Esant mažam *aperture_size* parametrai (1, 3) ir mažam *blok_size* parametrai iki 1,3% segmentų (tiriant plokštę be defektų) ir iki 21% segmentų (tiriant plokštę su defektais) identifikuojami kaip klaidingi. Padidinus šiuos parametrus, klaidingai identifikuojamu segmentų sumažėja. Visi realūs defektai identifikuojami teisingai. Geriausi rezultatai gaunami, kai *aperture_size* = 5 ir 7, *blok_size* = 9, 11 ir 15 (3 lentelė).

3 lentelė. Rezultatai naudojant skirtingus Eigen decomposition algoritmo parametrus, kai koreliacijos koeficientas 0.9, segmento dydis – 50x50, 75x75, 100x100 pikselių.

| aperture size | block size | Segmento dydis, px | Koreliacijos koeficientas | Neteisingai identifikuotų segmentų skaičius, % |
|---------------|------------|--------------------|---------------------------|--|
| 5 | 5 | 50 x 50 | 0.9 | 21,02 |
| 7 | 5 | | | 13,15 |
| 5 | 9 | | | 0,93 |
| 7 | 9 | | | 1,02 |
| 5 | 15 | | | 0,19 |
| 7 | 15 | | | 0,19 |
| 5 | 5 | 75 x 75 | | 8,64 |
| 7 | 5 | | | 5,56 |
| 5 | 9 | | | 0,00 |
| 7 | 9 | | | 0,00 |
| 5 | 15 | | | 0,00 |
| 7 | 15 | | | 0,00 |
| 5 | 5 | 100 x 100 | | 3,57 |
| 7 | 5 | | | 2,50 |
| 5 | 9 | | | 0,00 |
| 7 | 9 | | | 0,00 |
| 5 | 15 | | | 0,00 |
| 7 | 15 | | | 0,00 |

Corner detection algoritmas realizuotas funkcija `preCornerDetect`. Jos prototipas:

```
void cvPreCornerDetect( const CvArr* image, CvArr* corners, int aperture_size=3 );
```

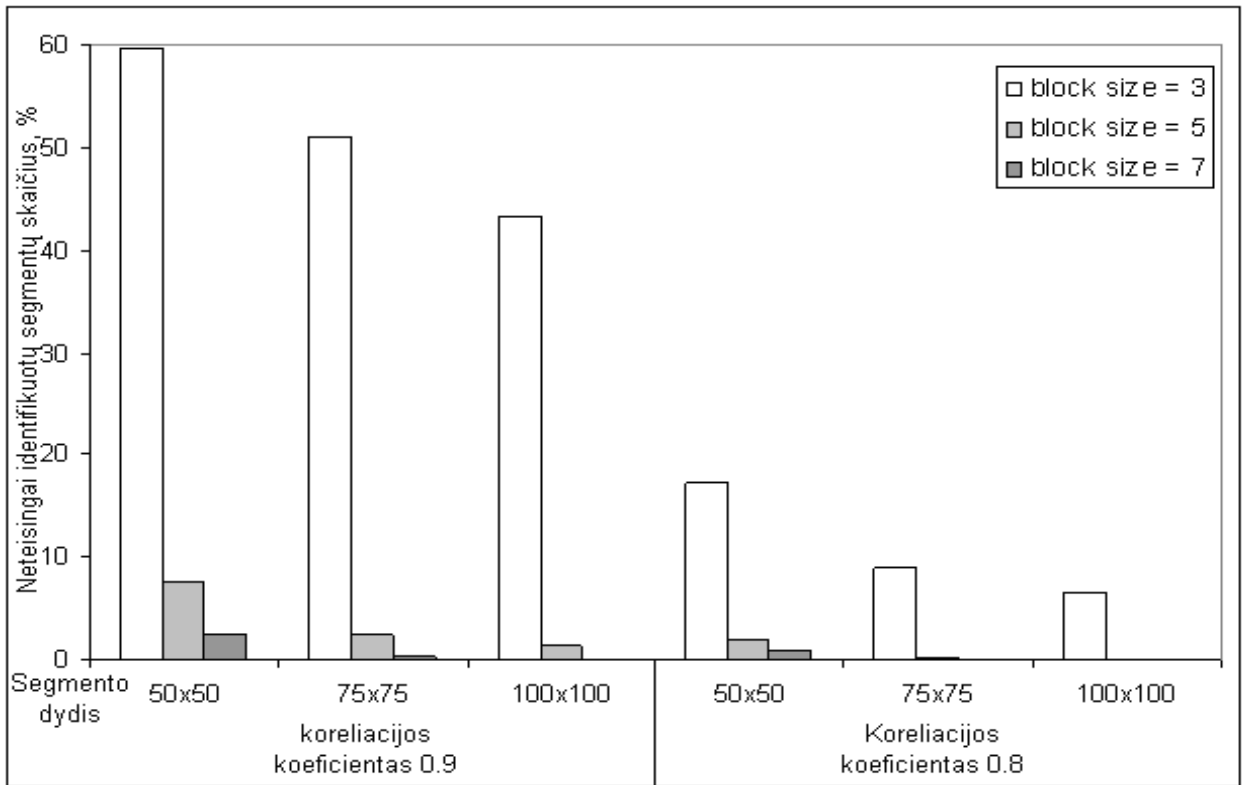
`image` – pradinis vaizdas;

`corners` – surasti pradinio vaizdo kampai;

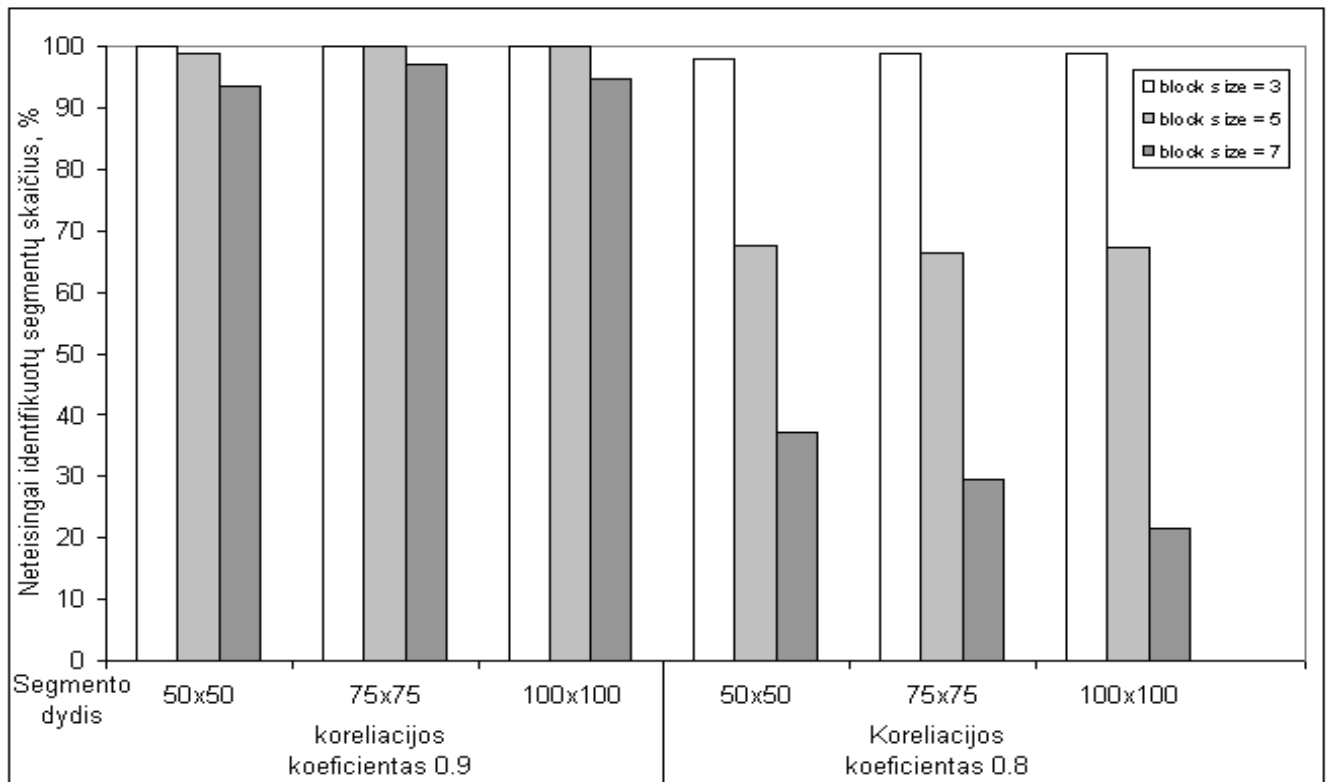
`aperture size` – branduolio dydis. Gali būti 1, 3, 5 ir 7.

Algoritmas jautrus bet kokiems pakitimams, todėl net tiriant plokštę be defektų iki 60% segmentų identifikuojami kaip klaidingi (26 pav.). Tiriant plokštę su realiai esančiais defektais, jie atpažįstami, tačiau didelė dalis segmentų, kuriuose nėra defektų, taip pat identifikuojami kaip klaidingi (27 pav.).

Geresni rezultatai gaunami padidinus segmento dydį, bei sumažinus koreliacijos koeficientą.



26 pav. Plokštės be realių defektų tyrimas parenkant skirtingus *Corner detection* parametrus, segmento dydžius ir koreliacijos koeficientus.



27 pav. Plokštės su realiais defektais tyrimas parenkant skirtingus *Corner detection* parametrus, segmento dydžius ir koreliacijos koeficientus.

Vaizdo apdorojimas *Harris corner detector* algoritmu OpenCV bibliotekoje atliekamas naudojant funkciją `cvCornerHarris`. Jos prototipas:

```
void cvCornerHarris( const CvArr* image, CvArr* harris_responce, int block_size, int aperture_size=3, double k=0.04 );
```

`image` – pradinis vaizdas;

`harris_responce` – apdorotas vaizdas;

`block_size` – naudojamo bloko dydis

`aperture size` – branduolio dydis. Gali būti 1, 3, 5 ir 7;

`k` – pasirenkamas jautrumo parametras.

Kadangi algoritmas panašus į *Eigen decomposition*, testuojant naudojamas pastovus parametras `aperture_size = 7`. Keičiami kiti 2 parametrai. Algoritmas gerai atpažįsta realius defektus, o taip pat tik iki 1% segmentų atpažįstami neteisingai. Parinkus didesnes `blok_size` (9, 11, 15) ir `k` (0.08, 0.12) reikšmes, klaidingai identifikuotų segmentų nebelieka (4 lentelė).

4 lentelė. Neatitinkančių segmentų skaičius naudojant skirtingus *Harris Corner Detection* algoritmo nustatymus.

| block size | k | segmento dydis | Koreliacijos koeficientas | Neatitinkančių segmentų skaičius | | | Neatitinkančių segmentų skaičius, % | |
|------------|------|----------------|---------------------------|----------------------------------|------------------------|------|-------------------------------------|------------------------|
| | | | | Be realių defektų | Su realiais defektais* | | Be realių defektų | Su realiais defektais* |
| 9 | 0.40 | 50x50 px | 0.9 | 2 | 43 | (32) | 0,19 | 1,02 |
| 11 | 0.80 | | | 1 | 37 | (33) | 0,09 | 0,37 |
| 15 | 1.20 | | | 1 | 35 | (33) | 0,09 | 0,19 |
| 9 | 0.40 | 75x75 px | | 0 | 20 | (20) | 0 | 0 |
| 11 | 0.80 | | | 0 | 19 | (19) | 0 | 0 |
| 15 | 1.20 | | | 0 | 19 | (19) | 0 | 0 |
| 9 | 0.40 | 100x100 px | | 0 | 10 | (10) | 0 | 0 |
| 11 | 0.80 | | | 0 | 10 | (10) | 0 | 0 |
| 15 | 1.20 | | | 0 | 10 | (10) | 0 | 0 |
| 9 | 0.40 | 50x50 px | 0.8 | 1 | 32 | (30) | 0,09 | 0,19 |
| 11 | 0.80 | | | 1 | 32 | (30) | 0,09 | 0,19 |
| 15 | 1.20 | | | 1 | 30 | (30) | 0,09 | 0 |
| 9 | 0.40 | 75x75 px | | 0 | 19 | (19) | 0 | 0 |
| 11 | 0.80 | | | 0 | 19 | (19) | 0 | 0 |
| 15 | 1.20 | | | 0 | 17 | (17) | 0 | 0 |
| 9 | 0.40 | 100x100 px | | 0 | 9 | (9) | 0 | 0 |
| 11 | 0.80 | | | 0 | 9 | (9) | 0 | 0 |
| 15 | 1.20 | | | 0 | 9 | (9) | 0 | 0 |

* - skliausteliuose nurodytas segmentų skaičius, atitinkantis realaus defekto vietą.

Rasta, kad optimaliausi *Harris corner detector* algoritmo parametrai yra tokie: $\text{block_size} \geq 9$; $\text{aperture size} = 7$; $k \geq 0.08$.

4.4. Optimalaus segmento dydžio nustatymas.

Nustatant optimaliausią segmento dydį, dydžiai parenkami nuo 10x10 pikselių iki 150x150 pikselių didinant kas 10. Koreliacijos koeficientas – 0.9. Išmatavus nufotografuotas plokštes pikseliai perskaičiuojami į milimetrus (5 lentelė). Tyrimui naudojami 3 vaizdo apdorojimo algoritmai: *Sobel*, *Eigen decomposition* ir *Harris corner detector*.

5 lentelė. Pikselių perskaičiavimas į milimetrus

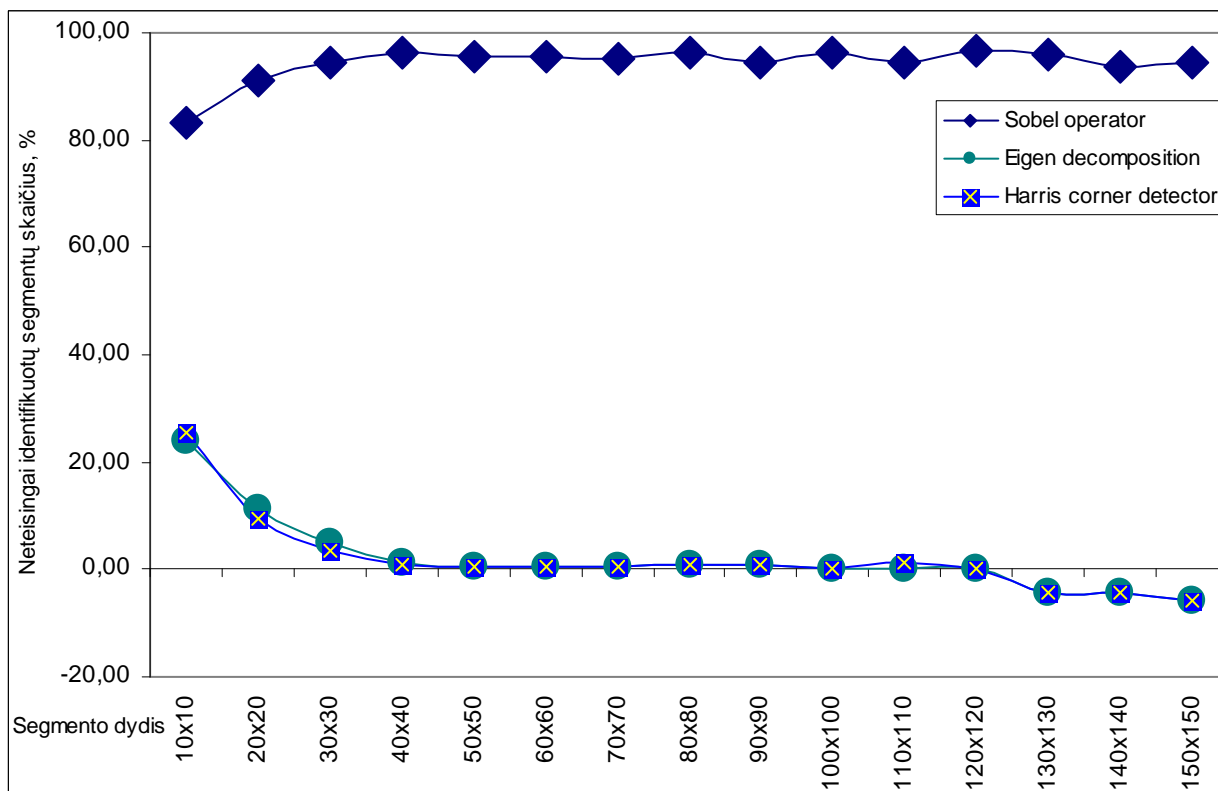
| Pikseliai | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 110 | 120 | 130 | 140 | 150 |
|-------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|
| Milimetrai | 0,9 | 1,8 | 2,7 | 3,6 | 4,5 | 5,3 | 6,2 | 7,1 | 8,0 | 8,9 | 9,8 | 10,7 | 11,6 | 12,5 | 13,4 |
| Milimetrai (optiškai priartinus) | 0,4 | 0,9 | 1,3 | 1,7 | 2,2 | 2,6 | 3 | 3,5 | 3,9 | 4,4 | 4,8 | 5,2 | 5,7 | 6,1 | 6,5 |

Kaip matome 28 pav. nenaudojant papildomo apšvietimo ir esant mažam segmento dydžiui (10x10 px – 30x30 px) neteisingai identifikuotų segmentų skaičius gana didelis. Naudojant *Sobel* operatorių, neteisingai identifikuotų segmentų skaičius labai didelis esant bet kokiam segmento dydžiui, todėl norint naudoti šį vaizdo apdorojimo metodą, reikėtų sumažinti koreliacijos koeficientą.

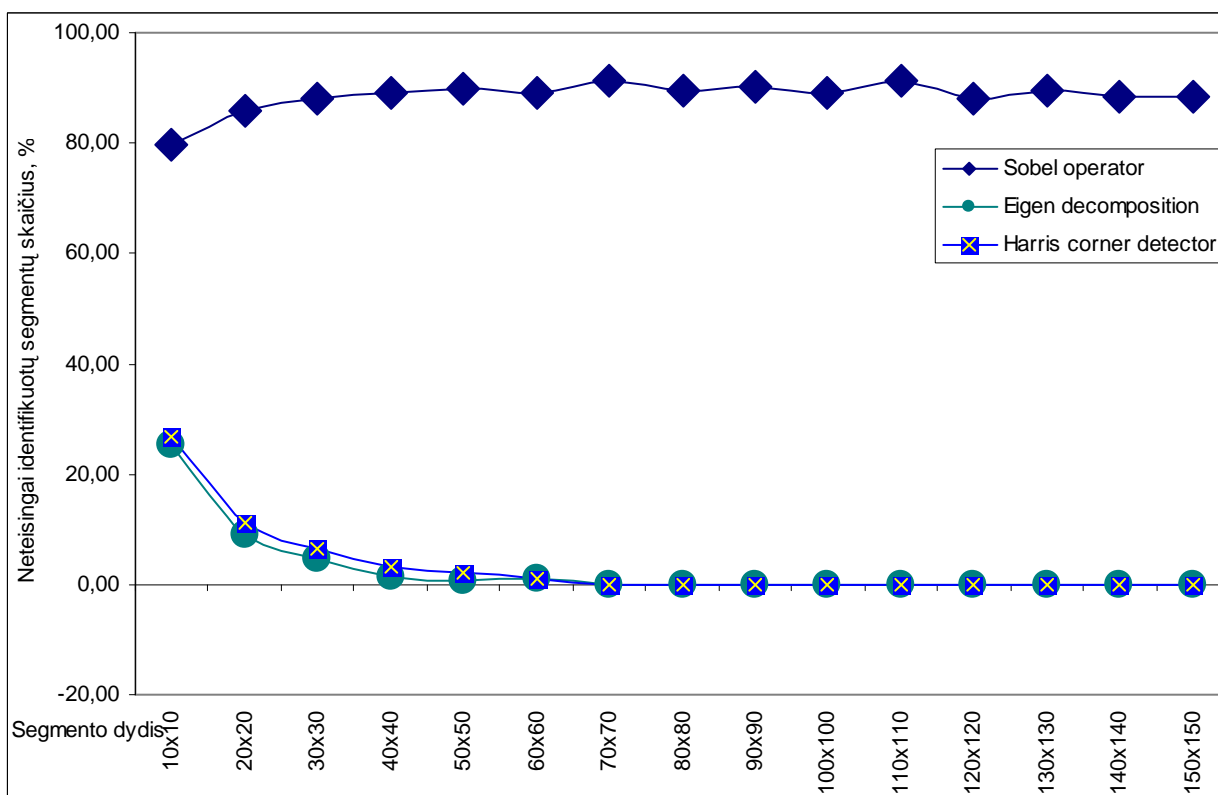
Rezultatai naudojant *Eigen decomposition* ir *Harris corner detection* metodus labai panašūs. Segmento dydį nustačius į 50x50px – 120x120px, klaidingai identifikuotų segmentų nebelieka, arba jų skaičius labai mažas (~1%). Pastebėta, kad segmentai identifikuojami kaip neatitinkantys šablono vaizdo pakraščiuose, nes dalinant vaizdą į fiksuoto dydžio segmentus, kraštuose lieka mažesni segmentai nei nustatyta (žr. 8 pav. dešinėje).

Segmento dydžiui pasiekus 130x130 px, buvo nebeatpažinti realiai plokštėje esantys defektai, t.y. visi segmentai buvo identifikuojami kaip atitinkantys šabloną (28 pav. tai pavaizduota kaip neigiamas neteisingai identifikuotų segmentų skaičius).

Kaip papildomą apšvietimą panaudojus šviesos diodų apšvietimo modulį (15 pav. dešinėje), visi realūs defektai atpažįstami esant bet kokiam tiriamam segmento dydžiui. (29 pav.) Taip pat sumažėja *Sobel* operatoriaus daromų klaidų skaičius, tačiau jis išlieka didelis (~88%). Optimalus segmento dydis, kai atpažįstami visi defektai, ir nedaroma klaidų (išskyrus vaizdo pakraščiuose kur segmento dydis nebeatitinka nustatyto) yra nuo 50x50px.

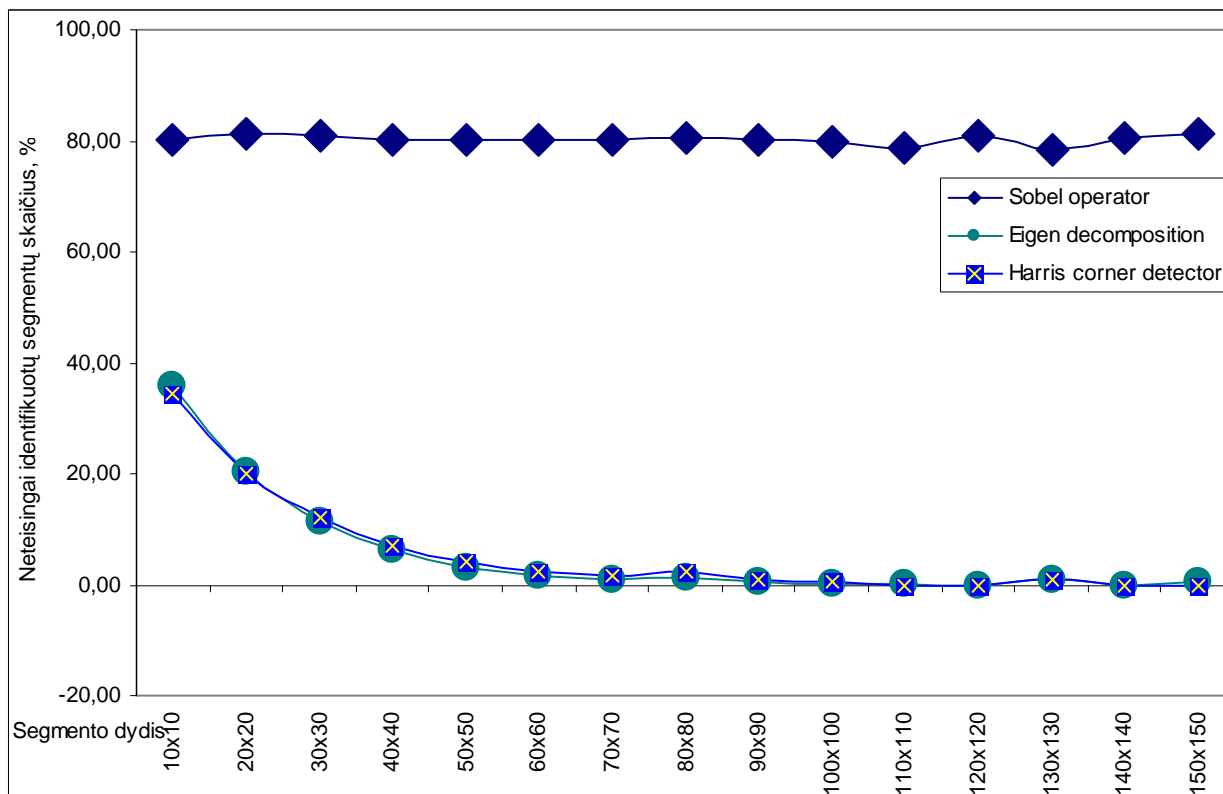


28 pav. Rezultatai nenaudojant papildomo apšvietimo



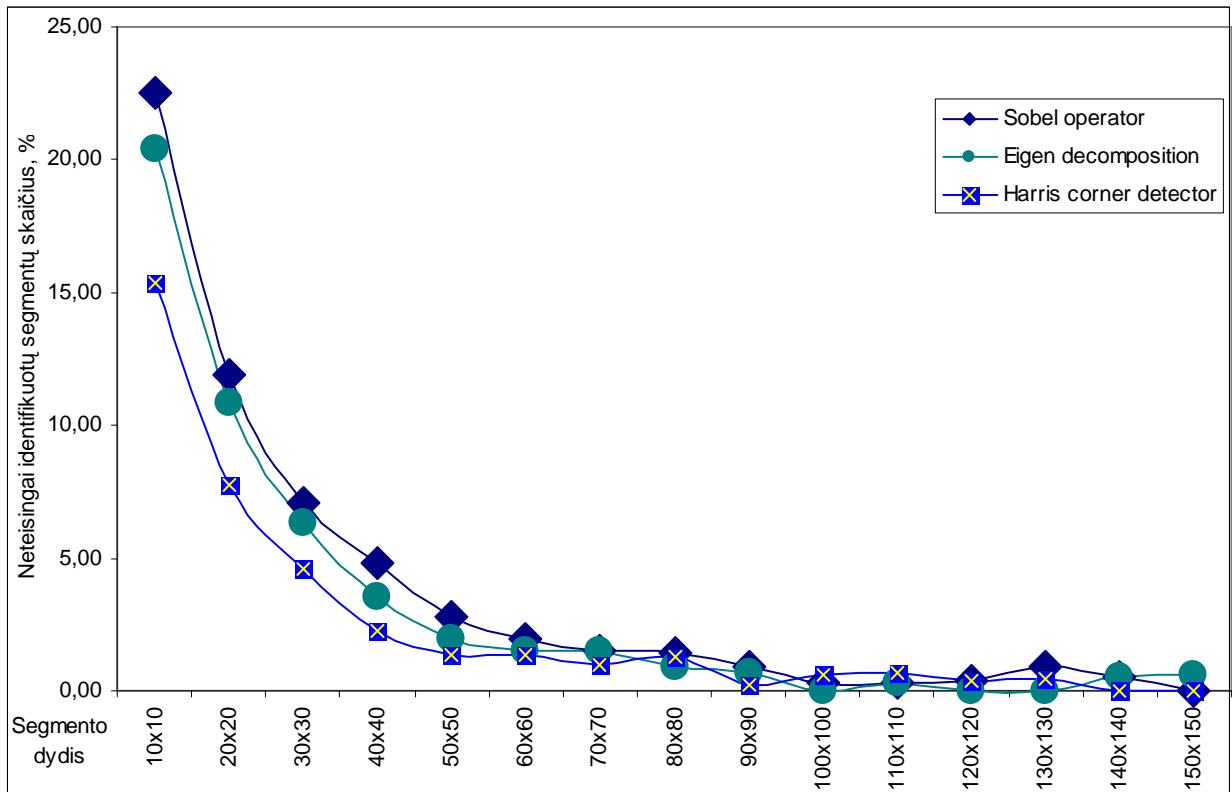
29 pav. Rezultatai naudojant šviesos diodų apšvietimą

Naudojant kaitrinių lempų apšvietimą ir vaizdą apdorojant Sobel operatoriumi, neteisingai identifikuotų segmentų skaičius yra apie 80% (30 pav.). Naudojant kitus 2 metodus, visi realūs defektai atpažįstami, tačiau segmento dydis, kai nedaroma klaidų – 80x80px ir didesnis.



30 pav. Rezultatai naudojant kaitrinių lempų apšvietimą

Norint gauti geresnę tiriamos plokštės vaizdo skiriamąją gebą, vaizdas buvo optiškai priartintas. Rezultatai naudojant visus 3 metodus buvo geri, tačiau taip galima ištirti mažesnę plokštės dalį. Daugiausia klaidų buvo daroma segmento dydžiui esant 10x10px – 50x50px. (*Sobel*: 22,5% - 3%, *Eigen decomposition*: 20,4% - 2%, *Harris corner detector*: 15,4% – 1,3%). Esant didesniai segmento dydžiui, visi realūs defektai buvo atpažįstami, o klaidos daromos tik vaizdo kraštuose.



31 pav. Rezultatai optiškai priartinus tiriamą plokštę

5. IŠVADOS IR SIŪLYMAI

- Ištyrus automatinės optinės kontrolės sistemas suprojektuotas ir pagamintas spausdintinio montažo plokščių surinkimo kokybės testavimo stendas.
- Sukurta programinė įranga vaizdo apdorojimui.
- Programinė įranga kurta naudojant C++ programavimo kalbą, todėl nereikalingi papildomi įrankiai ją naudojant.
- Išrinkti geriausiai kokybės įvertinimui tinkantys vaizdo apdorojimo algoritmai bei jų parametrai. Geriausiai tam tinkami *Sobel*, *Eigen decomposition* ir *Harris corner detector* algoritmai. Jei nenaudojamas papildomas apšvietimas, nerekomenduojama naudoti *Sobel* algoritmo, nes jis labai jautrus net mažiausiems vaizdo pokyčiams. Esant galimybei fotografuojant tiriamą plokštę, ją reikėtų optiškai priartinti. Rekomenduojami algoritmų nustatymai:

Sobel:

```
xorder = 1;
yorder = 0;
aperture_size = 7.
```

Eigen decomposition:

block_size ≥ 9 ;
 aperture size ≥ 5 .

Harris corner detector:

block_size ≥ 9 ;
 aperture size ≥ 7 ;
 $k \geq 0.08$

- Norint gauti geros kokybės vaizdus, būtina įrengti gerą papildomą apšvietimą.
- Nustatyta kad optimalus segmento dydis naudojant šviesos diodų apšvietimą – 50x50px (4.5x4.5 mm) ir didesnis, naudojant kaitrinių lempų apšvietimą – 80x80px (7.1x7.1 mm) ir didesnis.
- Nenaudojant apšvietimo, segmento dydžio nereikėtų parinkti didesnio nei 120x120px (10.7x10.7 mm) nes gali būti nebeatpažįstami realūs defektai.
- Segmento dydis turi būti parenkamas toks, kad tiriamas vaizdas išsidalintų į lygias dalis, kitu atveju galimi klaidingi išpėjimai apie defektą.
- Tinkamiausias koreliacijos koeficientas - 0.9 (naudojant Sobel metodą vaizdo apdorojimui – 0.8).
- Fotografuojamas vaizdas turėtų būti gaunamas su visa įmanoma informacija, šiuo atveju – kamera turėtų fotografuoti spalvotu režimu ir gautą vaizdą programiškai reikėtų paversti į tokį formatą, koks naudojamas jį apdorojant.
- Tobulinant sistemą, prarandamą spalvinę informacija būtų galima panaudoti kaip papildomą priemonę kokybės vertinimui.
- Fotografuojant naudoti rankinį režimą.
- Fotografavimo parametrus nustatyti taip, kad būtų gaunama kuo mažiau pašalinių triukšmų. Eksperimentų metu naudoti parametrai kai naudojamas papildomas apšvietimas buvo:
 - išlaikymas – 1/125;
 - diafragmos atvėrimas – 5.6;
 - kameros jautrumas - ISO200.
- Magistrinio darbo rezultatai naudojami vykdant projekto E!3807 – „EKO-FACTORY: Intelektuali konvejerinės gamybos identifikavimo sistema” darbus.
- Darbo tema į žurnalą “Information Technology and Control” parašytas straipsnis “Image Analysis Problems In AOI Systems”.

LITERATŪRA

1. Automated Optical Inspection for Electronics Manufacturing, prieiga per internetą: <http://www.home.agilent.com/agilent/editorial.jsp?cc=IN&lc=eng&ckey=205277&nid=-11144.0.00&id=205277>
2. Vaizdo jutikliai ir jų ateities vaizdai, prieiga per internetą: <http://rtn.elektronika.lt/rtn/0303/jutikliai.html>
3. Nello Zuech. Machine Vision Software. Posted 11/03/2005. prieiga per internetą: <http://www.machinevisiononline.org/public/articles/articlesdetails.cfm?id=2641>
4. Ilaria Bartolini, Paolo Ciaccia, and Marco Patella. WARP: Accurate Retrieval of Shapes Using Phase of Fourier Descriptors and Time Warping distance. IEEE Transactions on pattern analysis and machine intelligence, vol. 27, no. 1, January 2005
5. Ying Shan, Harpreet S. Sawhney, Bogdan Matei and Rakesh Kumar. Shapeme Histogram Projection and Matching for Partial Object Recognition. IEEE Transactions on pattern analysis and machine intelligence, vol. 28, no. 4, April 2006
6. Esa Rahtu, Mikko Salo, and Janne Heikkila. Affine Invariant Pattern Recognition Using Multiscale Autoconvolution. IEEE Transactions on pattern analysis and machine intelligence, vol. 27, no. 6, June 2005
7. Gao, Q., Zhang, L., Zhang, D., Yang, J. On Image Matrix Based Feature Extraction Algorithms. SMC-B(37), No. 5, October 2007, 1373-1374.
8. T. A. ABBASI, M. U. ABBASI. A Proposed FPGA Based architecture for Sobel Edge Detection Operator. J. of Active and Passive Electronic Devices, Vol. 2, 2007, 271–277.
9. Canny J. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, Nov. 1986, 639-643.
10. Xin Wang. Laplacian Operator-Based Edge Detectors. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 5, 2007, 886-890.
11. Konstantinos G. Derpanis, The Harris Corner Detector, October 27, 2004
12. T. Lindeberg (1998). "Feature detection with automatic scale selection". International Journal of Computer Vision 30 (2)
13. Lindeberg T. Edge detection and ridge detection with automatic scale selection. International Journal of Computer Vision, 30, 2, 1998, 117-154.
14. Bose T. Digital Signal and Image Processing. John Wiley & Sons, Inc. 2004.
15. F. Mohanna, F. Mokhtarian. Performance Evaluation of Corner Detection Algorithms under Similarity and Affine Transforms. BMVC 2001.

16. J. P. Lewis, Fast Normalized Cross-Correlation. Industrial Light & Magic
17. wxDev-C++, prieiga per internetą: <http://wxdsgn.sourceforge.net/index.php>
18. Ben Sintay. UNIX TIME STAMP . COM, prieiga per internetą: <http://www.unixtimestamp.com/>
19. Canon EOS Digital SDK “EDSDK2.0 API Programming Reference”, Revision 2.0, 5/28/2007
20. INI file, from Wikipedia, the free encyclopedia, prieiga per internetą: http://en.wikipedia.org/wiki/INI_file
21. CIniFile Class by Adam Clauss, prieiga per internetą: <http://www.codeproject.com/file/CIniFile.asp>
22. Curtin's Guide to Digital Cameras And Other Photographic Equipment, prieiga per internetą: <http://www.shortcourses.com/guide/>