



Kauno technologijos universitetas
Elektros ir elektronikos fakultetas
Automatikos katedra

MAGISTRO BAIGIAMASIS DARBAS

T000M041

Laminuotos medienos drožlių plokštės kokybės tikrinimo prototipo kūrimas ir tyrimas
The development and analysis of laminated chipboard quality assurance prototype

Ataskaita

Atliko: EAM 9/5 gr. stud.

Artūras Tamulynas

Darbo vadovas:

Dr. Renaldas Urniežius

KAUNAS, 2015

SANTRAUKA

Iki šiol laminuotų medienos drožlių plokščių kokybės nustatymo uždavinius sprendavo kokybės kontrolierius, naudodamas įvairius matavimo įrankius, pasikliaudamas savo regos bei kitais pojūčiais ir įgūdžiais. Kadangi žmogus tapo lėčiausia grandimi gamybos procese, šio darbo tikslas yra sukurti laminuotos medienos drožlių plokštės kokybės tikrinimo sistemos prototipą, skirtą skylių po gręžimo patikrinimui, panaudojant kompiuterinę vaizdo analizės sistemą.

Siekiant ištirti kokie yra naudojami vaizdo analizės objektų atpažinimui metodai, pirmame šio darbo skyriuje atlikta literatūros apžvalga. Išsiaiškinti nestandartiniai metodai, jų veikimo principai ir panaudojimo sritys.

Antrame darbo skyriuje yra analizuojami prototipo kūrimui pasirinkti metodai tokie kaip: Sobel operator, Canny edge detector, Laplace operator, Eigen decomposition, Moravec corner detection, Harris corner detector. Išsiaiškinta, kad visi šie algoritmai turi skirtingus parametrų, leidžiančių apdoroti vaizdą kiekius bei galimybes. Taip pat pasirinktas vaizdų palyginimo metodas – tiesinė koreliacija.

Trečioje darbo dalyje nagrinėjama kuriamo prototipo techninė bei programinė įrangos. Išvardintos techninės įrangos komponentai, jų tarpusavio sąsaja bei įtaka sistemos veikimui, plačiau panagrinėjami esminių dalių, tokių kaip kamera parametrai. Ištirta apšvietimo intensyvumo svarba rezultatų gavimui. Toliau pateikiama programinės įrangos bei bendra sistemos veikimo blokinės schemas. Išanalizuotas bei iliustruotas sistemos veikimo principas. Toliau darbe atliekamas sistemos patikrinimas bei efektyvumo tyrimas. Bandoma kameros parametrų, apšvietimo stiprumo, pikselių kiekio įtaka sistemos patikimumui. Parenkami tinkamiausi algoritmų parametrai prie skirtingų veiksmų, ieškomas optimaliausias variantas, tiksliam ir greitam defektų suradimui.

Raktiniai žodžiai: Vaizdo apdorojimo metodai, kokybės tikrinimas, kameros parametrai, defektai.

SUMMARY

So far, laminated chipboard determining the quality challenges of quality controller, using a variety of measurement tools, relying on their own vision and other senses and skills. Because the man has become the slowest link in the production process, the aim is to create a laminated chipboard quality inspection system prototype for the inspection hole after drilling, using a computerized image analysis system.

In order to investigate what is used for image analysis methods to identify objects on the first section of this paper carried literature review. Finding out non-standard methods, their principles of operation and application areas.

The second chapter analyzes the development of prototypes to choose methods such as Sobel operator, Canny edge detector, Laplace operator, Eigen decomposition, Moravec corner detection, Harris corner detector. They found that all of these algorithms has different parameters, allowing image processing volumes and opportunities. Also selected image comparison method - linear correlation.

The third part deals with assembling the prototype hardware and software. The listed hardware components, their mutual connection and influence the functioning of the more one looks at the key parts such as camera settings. The influence of light intensity of the importance of the results obtained. The following software and overall system performance flow diagram. Analyzed and illustrated the principles of operation. Further work is carried out the verification and performance test. Trying to camera settings, lighting strength, pixel how much influence the reliability of the system. Selects the most appropriate algorithm parameters to different factors, wanted the best option, for precise and rapid remediation of defects.

Keywords: Image processing methods, quality control, camera settings, defects.

TURINYS

1.	VAIZDO ANALIZĖS SISTEMA KOKYBĖS KONTROLEI	9
1.1	Objektų atpažinimo metodai.....	10
1.2	Objekto formų atpažinimas pasitelkus Furjė koeficientus ir laiko perstūmimą	10
1.3	Dalinai matomo objekto atpažinimas ir histogramų palyginimas	12
1.4	Objektų palyginimas ir atpažinimas naudojant autokonvoliuciją.....	13
2.	PASIRINKTI VAIZDO APDOROJIMO METODAI.....	14
2.1.	Sobel operatorius	14
2.2.	Canny Edge detection.....	15
2.3.	Laplace operatorius.....	16
2.4.	Eigen išskaidymo metodas	17
2.5.	Moravec corner detection	18
2.6.	Harris corner detector	19
2.7.	Vaizdų palyginimas	21
3.	MEDIENOS DROŽLIŲ PLOKŠTĖS KOKYBĖS TIKRINIMO SISTEMA	22
3.1.	Techninė įranga	22
3.2.	Programinė įranga.....	23
4.	SISTEMOS TIKRINIMAS. EFEKTYVUMO TYRIMAS	28
4.1.	Kameros įtaka vaizdo tyrimui.....	29
4.2	Vaizdo kameros parametrų įtaka tyrimui	38
4.3.	Algoritmų optimalių parametrų nustatymas	39
4.4.	Segmento dydžio įtaka.....	44
5.	Išvados	47
6.	Literatūra.....	48

IVADAS

Nors mediena yra vienas iš atsinaujinančių energijos šaltinių, tačiau pagal sunaudojimo mastą - gana ribotas resursas, todėl yra svarbu įvairiais kriterijais optimizuoti medienos gaminių produkcijos gamybą. Svarbiausias siekis yra sumažinti atliekų kiekį ir padidinti išeią. Tai būtų galima atlikti turint galimybę tiksliai įvertinti apdorojamų baldinių plokščių kokybę – kokybės klasę, defektus ir kt. Iki šiol laminuotų medienos drožlių plokščių kokybės nustatymo uždavinius sprendavo žmogus, naudodamas įvairius matavimo įrankius, pasikliaudamas savo regos bei kitais pojūčiais ir įgūdžiais. Tačiau tokiu būdu kokybės vertinimo uždavinys negali būti išspręstas optimaliai – žmogiškasis faktorius nėra pilnai tinkamas preciziškai įvertinti visų kokybės ypatybių. Be to, toks žmogaus dalyvavimas medžio baldų gamyboje pasidarė nepageidaujamas dar ir todėl, kad jis tapo lėčiausia grandimi, automatizuotame gamybos procese. Taip pat žmogaus darbo kokybę priklauso nuo jo patirties, nuotaikos, nuovargio ir kitų aspektų. Todėl sunku atlikti greitą darbo režimo pakeitimą ar operatyvų brokuotų detalių atsiradimo priežasties panaikinimą, kai defektų nustatymą atlieka žmogus.

Prasidėjus spraçiam informacinių technologijų vystymuisi, atsivėrė naujos galimybės šias technologijas pritaikyti įvairiose pramonės šakose. Viena iš jų yra medienos produkcijos gamyba, jos optimizavimas – įvairių procesų, tarp jų ir kokybiškų bei defektuotų produktų nustatymo kompiuterizavimas.

Tokie sprendimai leidžia pagerinti:

- Medžio plokščių apdorojimo proceso didesnę našumą ir efektyvesnę aparatūros išnaudojimą;
- Kokybiškesnės produkcijos gamybą;
- Reakciją į pakitusį gamybinės linijos darbo režimą.

Tyrimo problemos aktualumas:

Iki šiol laminuotų medienos drožlių plokščių kokybės nustatymo uždavinius sprendavo kokybės kontrolierius, naudodamas įvairius matavimo įrankius, pasikliaudamas savo regos bei kitais pojūčiais ir įgūdžiais. Tačiau išsivysčius informacinėm technologijom, gamybos pramonėje vis dažniau ir plačiau naudojama kompiuterinė vaizdo analizė. Tai toks procesas, kai naudojamas optiniai, infraraudonųjų spindulių, lazeriniai, ultragarsiniai jutikliai ir vaizdo kamera.¹

¹ Sistema gali dirbti tiek suderinta su kitais jutikliais tiek atskirai, priklausomai nuo uždavinio sprendimo sudėtingumo bei kintamųjų, kuriuos norima stebėti gausos.

Minimos kompiuterinės sistemos buvo pradėtos taikyti pramonėje praėjusio amžiaus pabaigoje ir nuo tol tobulinamos. Pagrindinis tikslas: sumažinti kiek įmanoma arba visai pašalinti žmogiškojo faktoriaus įtaką gamybos bei kokybės tikrinimo procesuose. [2] Vystymosi pradžioje kompiuterinės vaizdo analizės sistemos buvo salyginai brangios, jomis buvo sudėtinga naudotis, o ir efektyvumu nei greičiu jos nepasizymėjo, tačiau dabartinių kokybės tikrinimo sistemų savybės stipriai išstobulėjo, o patikimumas ir efektyvumas priartėjo 100%.

Tyrimo objektas – laminuotų medienos drožlių plokščių kokybės užtikrinimas po skylių išgręžimo centrų.

Darbo tikslas - sukurti laminuotos medienos drožlių plokštės kokybės tikrinimo sistemos prototipą skirtą skylių po gręžimo patikrinimui panaudojant kompiuterinę vaizdo analizės sistemą.

Darbo uždaviniai:

- Ištirti kompiuterinės vaizdo analizės sistemų kokybės tikrinimui veikimo principus;
- Sukurti prototipą su programine įranga baldinių plokščių po gręžimo kokybės tikrinimui;
- Išrinkti medžio pjuvenų plokščių kokybės įvertinimui efektyviausius vaizdo apdorojimo algoritmus ir jų parametrus;
- Atlikti sistemos patikimumo tyrimus.

Laminuotos medienos drožlių plokštės baldų gamybos pramonėje yra trys pagrindinės vietos, kur gali būti naudojama kompiuterinė vaizdo analizės sistema kokybės tikrinimui:

1. Laminato (plėvelės) apvilkinimo ant medienos drožlių plokštės kokybė. Tai pirmas ir vienas iš svarbiausių procesų, gaminių srautas yra nestabdomas, nes tai tiesiogiai įtakoja gaminių padengimą laminatu. Greičių svyravimai – pagrindinė „raukšlių“ ir įtrūkimo atsiradimo priežastis. Taigi svarbiausias faktorius yra palaikyti vienodą gaminių srautą bei užtikrinti pastovų ir nekintantį klijų sluoksnį prieš prieš dengiant plokštes tekstūrine plėvele. Šiame gamybos etape pastebėti nekokybiškas detales yra ganėtinai paprasta, o brokas lengvai ištaisomas (1pav.).



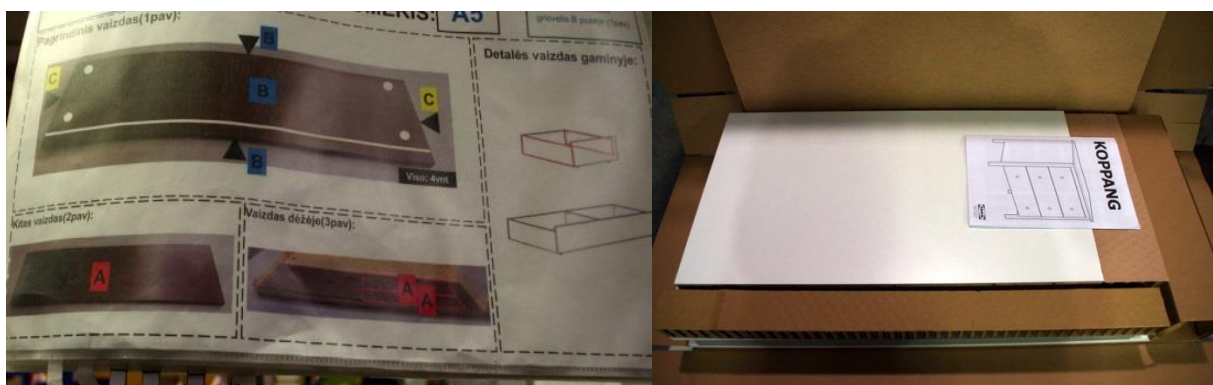
1 pav. Laminavimo linija ir defektų pavyzdžių lentelė

2. Skylių išgręžimas. Daugiausia dėmesio reikalaujanti gamybos proceso dalis yra išgręžtų skylių kokybė, atitikimas brėžiniui ir minimalios paklaidos išlaikymas. Menkiausias staklių išsiderinimas, gali įtakoti pakankamai didelius nuokrypius nuo užduoties (brėžinio) iki rezultato gavimo (sugręžtos detalės). Defektuotas detales pastebėti yra sudėtinga be specialios matavimo įrangos ir tai reikalauja nemažai laiko, apie 10 min. vienai sugręžtai detaliai patikrinti. Pagamintas brokuotas detales ištaisyti yra labai sudėtinga, kai kuriais atvejais neįmanoma. Svarbiausias faktorius yra greitaveikiškumas kokybės užtikrinime, patikrinti kas 10 gaminį nstabdant gamybinės linijos darbo ir šiam procesui skirti neilgiau kaip 1 minutę laiko (2 pav.).



2 pav. Skylių išgręžimo staklės bei defektų pavyzdžiai

3. Pakavimo linija. Gaminiai yra kompleksiniai, surenkami pirkėjo, svarbiausia supakuotų detalių kiekis, rūšys, instrukcijos buvimas. Pagaminti baldai iš atskirų detalių yra pakuojami į kartonines dėžes keliais sluoksniais, optimaliai išnaudojant pakuotės matmenis. Šis procesas yra standartizuotas (pirma guldomi baldų šonai, tuomet smulkesnės detalės (baldų kojos, stalčių priekinės detalės) priklausomai nuo baldo, galiausiai sudedami viršutinės ir apatinės gaminio dalys ir instrukcija). Pakavimo linijoms taip pat galima pritaikyti kompiuterinės vaizdo analizės sistemą, kuri susidėtų iš kelių segmentų skritų stebėti atskiriems pakavimo linijos darbo etapams (sluoksniams). Taip būtų užtikrinamas supakuoto gaminio detalių kiekių ir rūšių atitikimas (3 pav.).



3 pav. Pakuotės sudėjimo pavyzdžiai

Šiame darbe pateikti skyriai, kuriuose aprašomi laminuotos medienos drožlių kokybės tikrinimo prototipo sistemos komponentai, vaizdo analizei naudojami filtrai, algoritmai pateikiamos jų blokinės schemos, aptariami ir kiti metodai. Pateikiama informacija apie atliktus kokybės tikrinimo sistemos efektyvumo tikrinimo darbus bei išorinių veiksnių įtaką. Galiausiai išdėstomos tiriamojo darbo išvados, pastabos ir rezultatai.

1. VAIZDO ANALIZĖS SISTEMA KOKYBĖS KONTROLEI

Pastaruoju metu yra siūlomos standartizuotos industrinės kokybės tikrinimo sistemos, pritaikytos konkrečiai pramonės šakai ar tam tikram produktui tirti. Nepaisant to, yra nemaža gamybinės pramonės dalis, kurios produkcijai tokių standartizuotų sistemų nepritaikyta, dėl tam tikrų priežasčių, pvz: gaminamas produktas visiškai naujas rinkoje, produktas - nestandartinis. Tokiais atvejais projektuojami, programuojami ir diegiami kokybės tikrinimo sistemos pagal konkrečius reikalavimus. Pagrindiniai kompiuterizuotos vaizdo analizės sistemos kokybės kontrolei komponentai yra:

1. Kompiuteris;
2. Kameros;
3. Apšvietimas;
4. Vaizdo analizės programa.

Tikrinamo objektas yra apšviečiamas, kad išvengti nepageidaujamų triukšmų, tuomet nufotografuojamas ir jo atvaizdas siunčiamas į sistemą tuomet analizuojamas. Šiame etape svarbiausias vaidmuo atitenka vaizdo analizės programai. Siekiant užtikrinti programinės įrangos patikimumą ir greitaveiką bei likusios techninės įrangos kainą, būtina parinkti optimalų vaizdo analizės metodą ir algoritmą defektų atpažiniui. Apšvietimas taip pat labai svarbus faktorius, jis turi būti parenkamas pagal tikrinamo objekto paviršiaus savybes, t.y. ne per intensyvus, kad neatsirastų nepageidaujamo blizgėjimo ant paviršiaus ir ne per silpnas, kad išvengti šešėlių ir kitų triukšmų. Labiausiai paplitęs šviesos diodų apšvietimas dėl ekonomiško, tolygaus spindulių paskirstymo ant paviršiaus ir ilgaamžiskumo. Vaizdo apdorojimo prietaisai taip pat skirstomi:

- Universalūs pramoniniai prietaisai su integruotom vaizdo kamerom, turintys atskirą šviesos šaltinio modulį ir net vaizdo apdorojimo programą. Tokie prietaisai paplitę ir siūlomi žymių gamintojų, tokių kaip Omron, Festo ir kt [3]. Tokius prietaisus nesunku konfigūruoti. Tačiau jų funkcijos ne visada tenkina reikalavimus sistemai.²
- Individualiai projektuojamos sistemos pagal konkrečius reikalavimus. Naudojami standartiniai komponentai ar jų rinkiniai (kamos. Tokiais atvejais, kai fotografuojamas objektas yra didelių gabaritų, ar reikia patikrinti daugiau nei vieną jo paviršių, naudojama daugiau vaizdo jutiklių, kameros gali būti sujungtos į matricą, kitu kamera yra slenkama virš objekto paviršiaus, o vaizdas segmentuojamas.

² Poole, Ian. *Automatic optical inspection, AOI systems*. In: *Radio-electronics: Resources and analysis for electronics engineers* [žiūrėta 2015 m. balandžio 7 d.]. Prieiga per internetą: http://www.radio-electronics.com/info/t_and_m/ate/aoi-automatic-automated-optical-inspection.php.

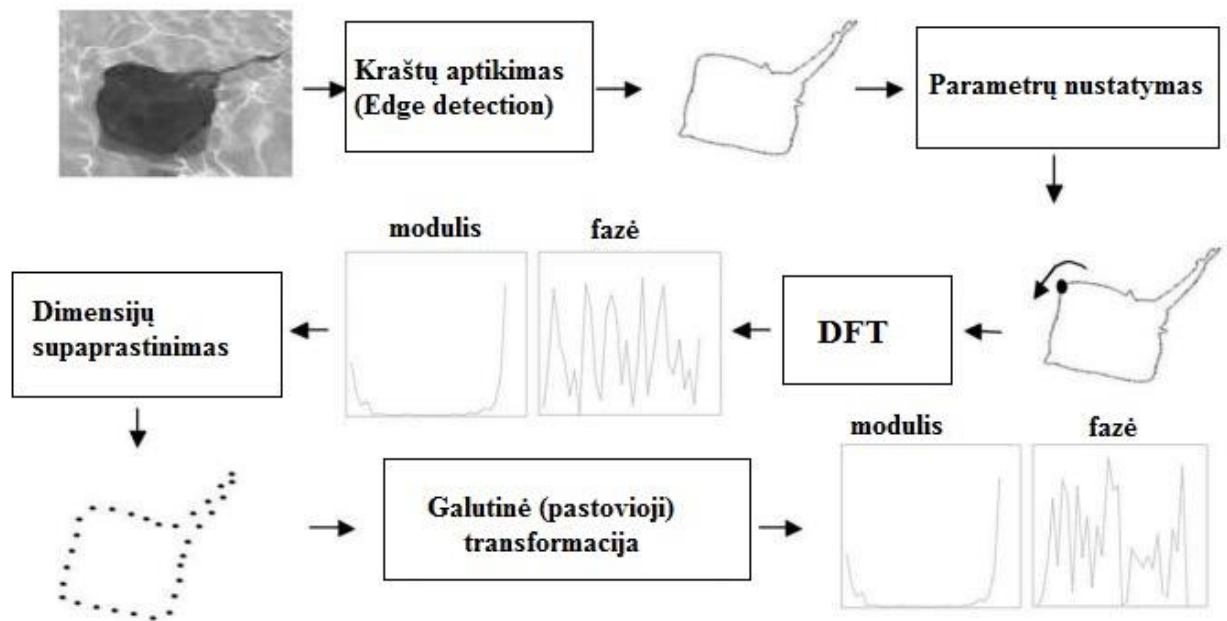
1.1 Objektų atpažinimo metodai

Objektų atpažinimo ir bruožų išskyrimo sistemos kuriamos remiantis įvairiausiais principais, bet dažniausiai naudojami yra tokie metodai:

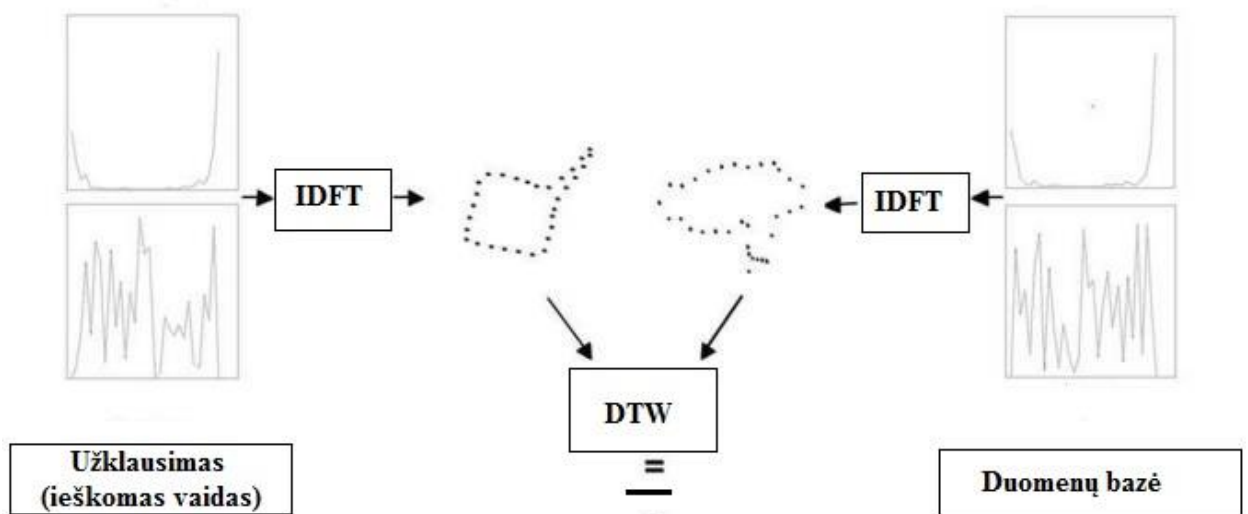
- Brūkšninio kodo atpažinimas;
- Sričių analizė;
- Forma paremtas atpažinimas;
- Morfologija;
- 1D metrologija, tai yra matavimas;
- 2D metrologija, tai yra kraštų ir kampų radimas;
- 3D metrologija – objekto matavimai trimatėje erdvėje.

1.2 Objekto formų atpažinimas pasitelkus Furjė koeficientus ir laiko perstūmimą

Tai gana naujoviškas tiesiogine Furjė transformacija pagrįstas metodas objektų vaizdų palyginimui ir bendrų bruožų (metmenų) suradimui, šis metodas vadinamas WARP, kas reiškia išvertus iš anglų kalbos metmenys (5 pav.). Šio metodo kūrėjas ir pradininkas Ilaria Bartolinis [4]. Metodika remiasi metmenų kitimo laike (Dynamic Time Warping, DTW) apskaičiavimu. Naudojama sistemoje kaip judančio kūno atpažinimas iš bendrų bruožų, kaip pavyzdžiui automobilių numerių suregistravimas į bendrą duomenų bazę, kurie važinėje miesto gatvėse ar kitose saugomose zonose. Metodas yra daug informatyvesnis lyginant su Furjė pasiūlytu algoritmu, nes gali įvertinti vaizdų metmenis esant dideliems ir labai mažiems fazių poslinkiams. Taip pat leidžia palyginti skirtingų kūnų formas (6 pav.).



5 pav. WARP metodo žingsniai³



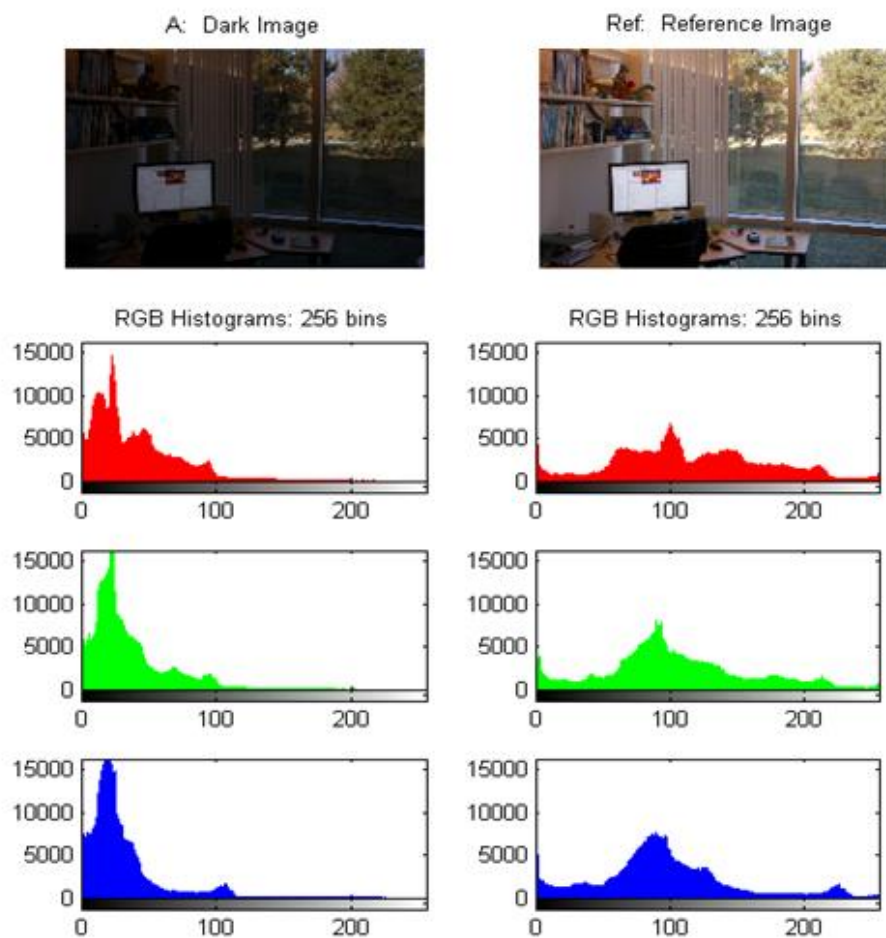
6 pav. Panašumo tarp dviejų formos aprašų palyginimas taikant WARP metodą⁴

³ Bartolini, Ilaria. *WARP: Accurate Retrieval of Shapes Using Phase of Fourier Descriptors and Time Warping Distance*. In: *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 27, no. 1, january 2005 [žiūreta 2015 m. kovo 20d.]. Prieiga per internetą: <http://www-db.disi.unibo.it/courses/SI-M/papers/BCP05.pdf>.

⁴ Bartolini, Ilaria. *WARP: Accurate Retrieval of Shapes Using Phase of Fourier Descriptors and Time Warping Distance*. In: *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 27, no. 1, january 2005 [žiūreta 2015 m. kovo 20d.]. Prieiga per internetą: <http://www-db.disi.unibo.it/courses/SI-M/papers/BCP05.pdf>.

1.3 Dalinai matomo objekto atpažinimas ir histogramų palyginimas

Formos ir žymių histogramos ar prototipinės formos (angl. prototypical shapes, shapemes) labai plačiai naudojamos tiek dvimačių, tiek trimačių objektų atpažinimui ir palyginimui. Panaudojant nepilnai (dalinai arba ne ryškiai) matomo objekto histogramas, galima jį palyginti (atpažinti) su duombazėje esančiomis viso pilnai matomo objekto histogramomis. Yra siūloma aprašyti tiriamą objektą keliomis jo formų ar spalvų histogramomis, ir palyginimą atpažinimui vykdyti tokiais žingsniais: matematiškai apibrėžti tiriamos histogramos projekciją, jos ekstremumo taškus ir amplitudę ir tuomet įvertinti atitikimą tarp etaloninės ir tiriamo objekto vaizdo histogramos gauta projekcija [6].

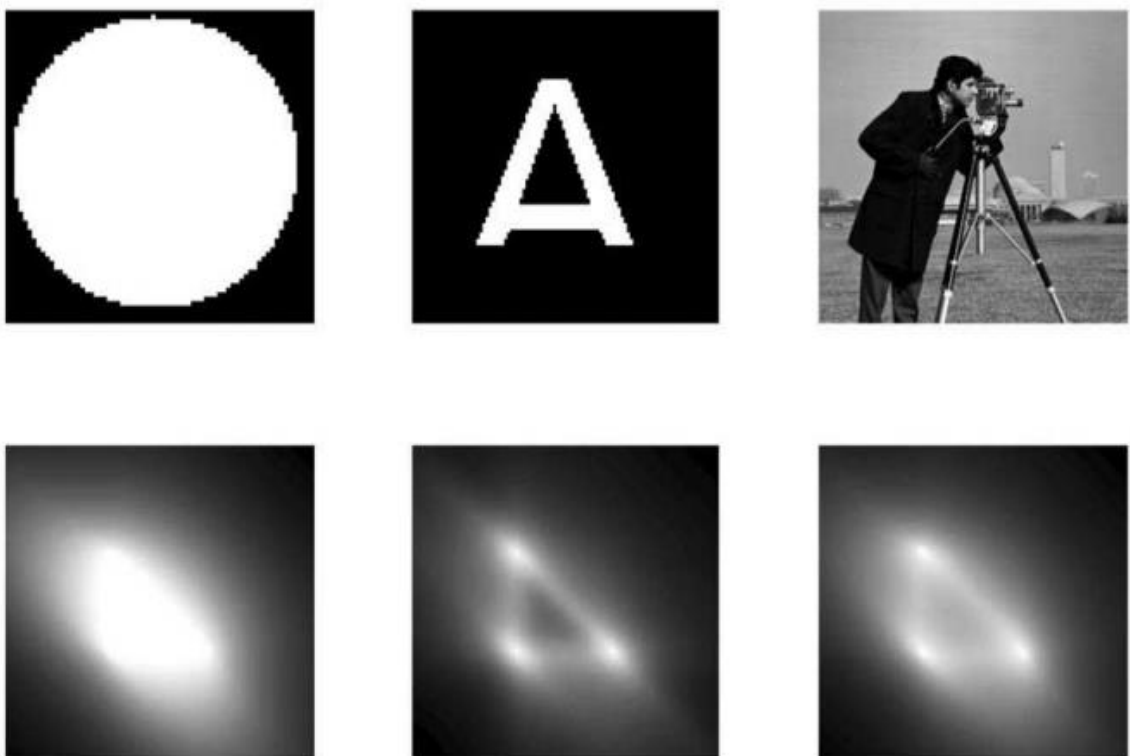


7 pav. Etaloninis dešinėje, bei tiriamasis vaizdas kairėje, bei jų intensyvumo histogramos⁵

⁵Publikuota mokomoji medžiaga. *Imhistmatch. Adjust histogram of image to match N-bin histogram of reference image.* In: *MathWorks* [žiūrėta 2015 m. vasario 12 d.]. Prieiga per internetą: <http://se.mathworks.com/help/images/ref/imhistmatch.html>.

1.4 Objektų palyginimas ir atpažinimas naudojant autokonvoliuciją

Šis siūlomas metodas grindžiamas tikimybinio vaizdo atpažinimo įvertinimu, tai kai objekto vaizdas transformuojamas naudojant kelių dimencijų autokonvoliucija (angl. Multiscale autoconvolution), šis algoritmas literatūroje daugiau žinomas tokiu trumpiniu, kaip MSA [7]. Pagrindinis šio metodo bruožas, ar privalumas toks, kad jis gali būti taikomas izoliuotiems objektams kai nėra būtinybės išskirti objekto ribų arba išskirtinių taškų. Vaizdo palyginimo apskaičiavimo greitis ganėtinai didelis, nes naudojama greitoji Furjė transformacija (angl. Fast Furje Transformation, FFT). Gauti FFT koeficientai naudojami objektų klasifikacijai.



8 pav. Kelių objektų MSA transformacijos⁶

⁶ Rahtu, Esa. Salo, Mikko and Heikkila, Janne. *Affine Invariant Pattern Recognition Using Multi-Scale Autoconvolution* In: *University of Helsinki, Rolf Nevanlinna Institute* [žiūrėta 2015 m. kovo 15 d.]. Prieiga per internetą: http://www.rni.helsinki.fi/~msa/pub/TPAMI_MSA.pdf.

2. PASIRINKTI VAIZDO APDOROJIMO METODAI

Visi anksčiau paminėti metodai daugiau skirti atskirų objektų palyginimui bei atpažinimui, yra ganėtinai sudėtingi - vaizdų atpažinimas vykdomas nepakankamai greitai, todėl darbe pasirinkti naudoti paprastesni vaizdo apdorojimo algoritmai:

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

Išvardinti metodai vaizdo apdorojimo procesuose naudojami kaip filtai skirti sumažinti gautos informacijos kiekį, t.y. išskiriami ir naudojami reikšmingi (esminiai) objekto bruožai ar sritys. Dažniausiai išskiriamos objekto kraštinės (*ang. Edge Detection*), reikšmingos vaizdo formos (*ang. Blob Detection*) taip pat kampai (*ang. Corner detection*). Visi paminėti algoritmai buvo paimti iš viešai prieinamų bibliotekų [8, 9, 10, 11].

2.1. Sobel operatorius

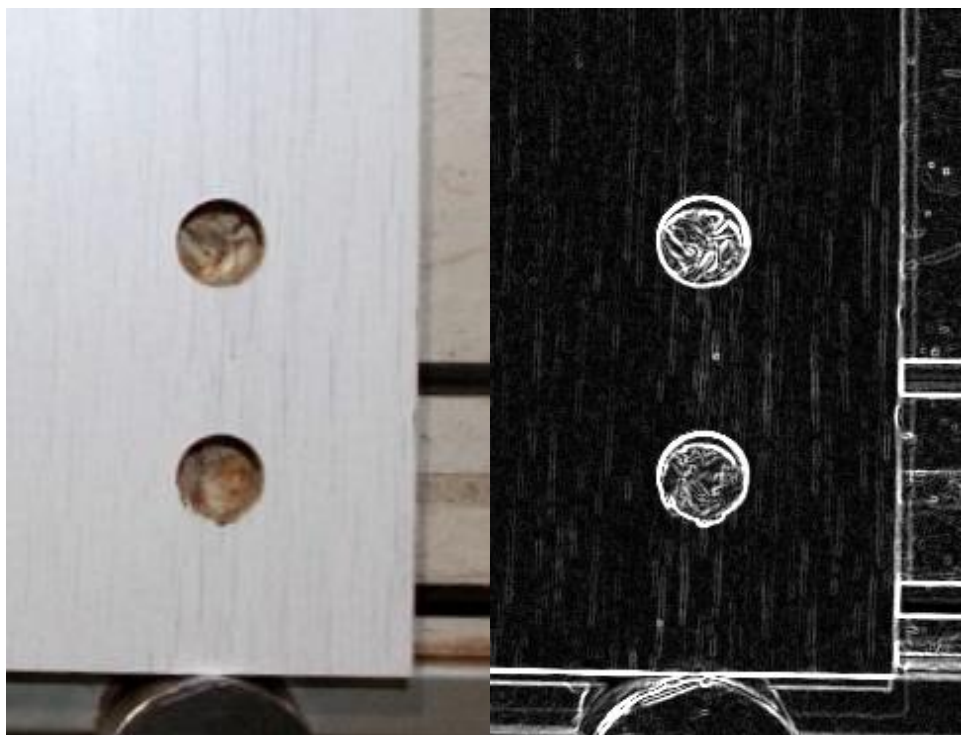
Sobel operatoriu apskaičiuoja vaizdo ryškumo perėjimą (*angl. Gradient*) kiekviename taške, kas parodo didžiausią galimą pokytį iš šviesaus į tamsų ir pokyčio intensyvumą ta kryptimi. Naudojamas 3x3 sąsūkų branduolių porą, kuri naudojama apskaičiuoti išvestines horizontalia ir vertikalia kryptimi. Jei pradinį vaizdą pažymėsime A , o G_x ir G_y bus kiekviename taške apskaičiuotos išvestinės, kurios apskaičiuojamos taip [12]:

$$G_x = \begin{pmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{pmatrix} * A \text{ ir } G_y = \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * A$$

Kiekviename vaizdo taške gaunamos pokyčio reikšmės apskaičiuojamos pagal formulę:

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

Toliau galima išskaičiuoti gradiento kryptį: $\Theta = \arctan\left(\frac{G_y}{G_x}\right)$



9 pav. Realus vaizdas (kairėje) ir Sobel operatoriumi apdorotas vaizdas (dešinėje) ⁷

2.2. Canny Edge detection

Canny edge detection pasižymi tuo, kad šis metodas naudoja kelių žingsnių algoritmą kraštų aptikimui esančių vaizde. Taip pat pasitelkiami keli reguliuojami parametrai (Gauso filtro dydis, slenksčiai), kurie padidina skaičiavimo greitį bei turi įtakos gautam rezultatui.

Kaip pavyzdį pateiksim Gauso filtro 5x5 matricą⁸:

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A$$

Naudojamos pakopos:

- Gauso filtru – naudojamas pirminiam mvaizdo apdorojimui, spalvų panaikinimas, triukšmo mažinimas, naudojamo vaizdo informacijos kiekio sumažinimas.
- Sričių su dideliu kontrastu (ryškių vietų) radimas. Ryškumo perėjimas, toks žingsnis naudojamas palengvinti horizontalių, vertikalinių, bei įstrižių kraštinių radimui. Tokų pat modelį naudoja ir kiti metodai kaip Sobel;

⁷*AForge.Imaging.Filters Namespace* In: *AForge.NET Framework* [žiūrėta 2015 m. vasario 8d.]. Prieiga per internetą: <http://www.aforogenet.com/aforge/framework/docs/html/cdf93487-0659-e371-fed9-3b216efb6954.htm>

⁸ Mokomoji medžiaga. *Kraštai ir Linijos.Canny* In: *Vilniaus Universitetas, Matematikos ir Informatikos fakultetas* [žiūrėta 2015 m. vasario 15 d.]. Prieiga per internetą: http://uosis.mif.vu.lt/~neutrino/slides/10_2.pdf.

- Silpnų taškų (minimumų) pašalinimas (*ang. Weak Edges suspresion*) – ieškomi lokalūs maksimumai [14];
- Kraštinių radimas nustatant vaizdo ir histerezės skirtingus, priešingus lygius – reikalingi du vaizdo apdorojimo slenksčiai: žemas ir viršutinis. Pritaikant žemą slenkstinį lygį, pažymimos kraštinės ir taškai, kurie negali būti tikri, turi būti nufiltruoti. Viršutinis lygis suteikia informaciją ir sužymi kraštus, kurie – tikri. Panaudojus abu slenkstinius lygius pagal jų surinktą informaciją, randami visi vaizdo taškai, kraštinės.



10 pav. Canny edge detection apdorotas vaizdas (dešinėje)⁹

2.3. Laplace operatorius

Vienas populiariausių sričių detektorių yra Laplace operatorius [16] (*angl. Laplacian of the Gaussian – LoG*) [17]. Kraštams surasti gali būti naudojami ir antros eilės išvestinėmis pagrįsti operatoriai, ypač dažnai naudojamas Laplaso operatorius, kuris nusakomas formule:

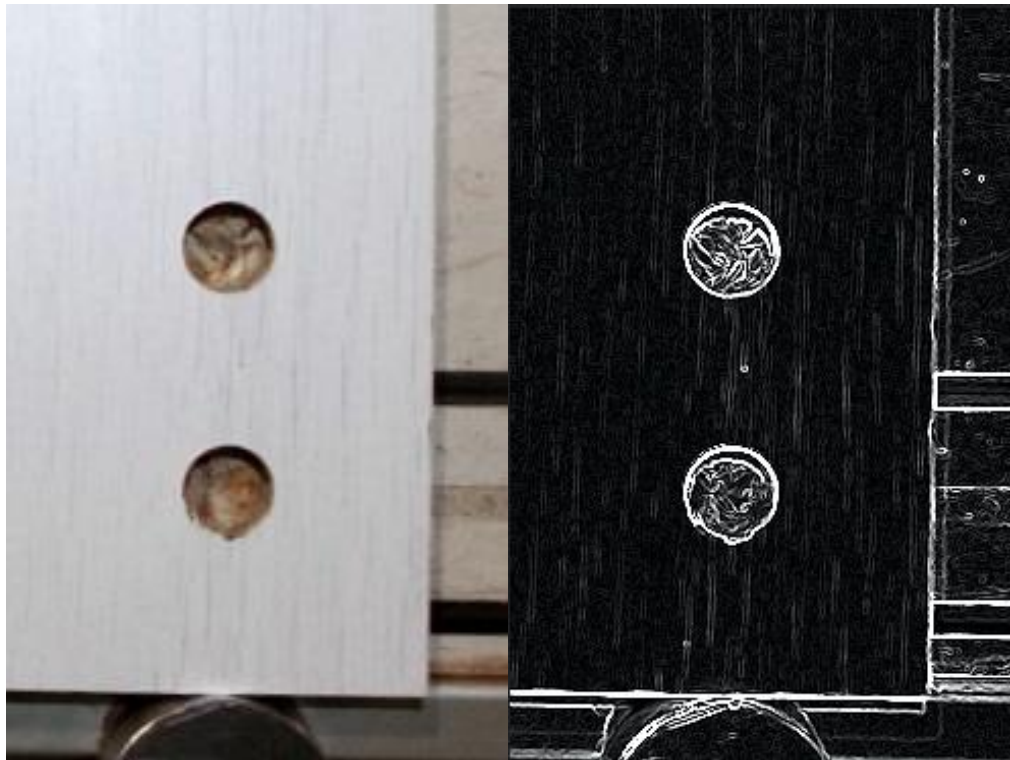
$$\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

Kraštų atpažinimo algoritmas randa vaizdo vietas, kurios yra šviesesnės arba tamsesnės nei aplink esantis regionas.

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

$$\Delta = \sum_{i=1}^n \frac{\partial^2}{\partial x_i^2}$$

⁹ Dr.Bharadi, Vinayak Ashok. *Canny Edge Detection in C#* In: *Code Project, For those who code* [žiūrėta 2015 m. sausio 14 d.] Panaudotas filtravimo programos kodas. Prieiga per internetą: <http://www.codeproject.com/Articles/93642/Canny-Edge-Detection-in-C>.



11 pav. Laplace operatoriumi apdorotas vaizdas (dešinėje) ¹⁰

2.4. Eigen išskaidymo metodas

Šis metodas vaizdų apdorojime naudojamas dimensijų mažinimui, paveikslo svorio mažinimui ir triukšmų šalinimui. *Eigen išskaidymo metodas* (Ang. *Eigen decomposition*) - tai apdorojamų matricių skaidymas į kanonines formas, kur jos aprašomos tikriniais vektoriais (ang. *eigenvectors*) ir tikrinėmis vertėmis (ang. *eigenvalues*) [19].

Eigen metodas vaizdus apdoroja pirmiausia juos išskaidydamas į tokias matricas:

$$X_l = \frac{1}{r} \begin{bmatrix} x_{11} + \dots + x_{r1} & x_{12} + \dots + x_{r1} \\ x_{d1} + \dots + x_{m1} & x_{d2} + \dots + x_{m2} \end{bmatrix};$$

Kintamieji nusakomi tikrinėmis vertėmis ir vektoriais [20].

$$\hat{x}_{h1}^T \hat{x}_{h2} + x_{11}x_{22} + x_{21}x_{12} + x_{31}x_{42} + x_{41}x_{32} = 0.$$

1. ¹⁰ Pawar, Kunal. *Implementation of Laplacian of Gaussian Edge Detection Algorithm using C#* In: *Code Project, For those who code* [žiūrėta 2015 m. sausio 14 d.] Panaudotas filtravimo programos kodas. Prieiga per internetą: <http://www.codeproject.com/Articles/37299/Implementation-of-Laplacian-of-Gaussian-Edge-Detec>



12 pav. Eigen decomposition apdorotas vaizdas (dešinėje)¹¹

2.5. Moravec corner detection

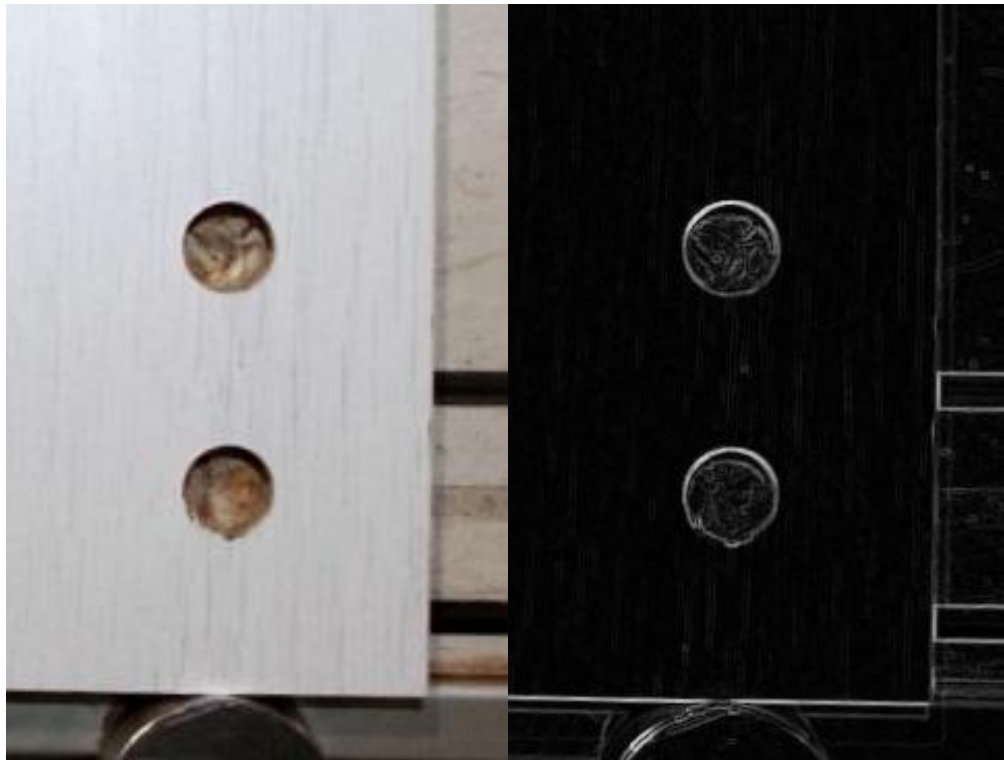
Moravec corner detection algoritmas dažniausiai naudojamas kampų arba reikšmingų taškų suradimui ir išskyrimui. Šis metodas kompiuterizuotos vaizdo analizės sistemose naudojamas kaip būdas nustatyti tam tikros rūšies sritis apdorojamuose vaizduose. Bendru atveju kampų suradimo algoritmas tikrina kiekvieną vaizdo tašką. Kampai apibūdinami kaip dviejų linijų (vertikalios ir horizontalios) susikirtimo vieta, taškas vertinamas kaip lokalus maksimumas. Pagrindinė šio operatoriaus problema yra, kad jis nėra izotropinis, jei kraštinė nėra ryškiam fone, t.y. nėra ryškaus susikirtimo, toks kampas gali būti neteisingai atpažintas. Tai vienas pirmųjų kampų aptikimo algoritmų, ir apibrėžia kampą, kaip vieną tašką ir srities maksimumą [22].

Paprasčiausias būdas surasti vaizdo kampus, yra naudoti formulę [23]:

$$D_x^2 D_{yy} D_{y^2} D_{xx} - 2D_x D_y D_{xy}$$

čia D_x – pirmoji vaizdo taško išvestinė, D_{xx} – antroji vaizdo taško išvestinė. Kraštai gaunami kaip lokalūs funkcijos maksimumai.

¹¹ Sergio Andrés Gutiérrez Rojas. *Multiple face detection and recognition in real time*. In: *Code Project, For those who code* [žiūrėta 2015 m. balandžio 14 d.] Panaudotas filtravimo programos kodas. Prieiga per internetą: <http://www.codeproject.com/Articles/239849/Multiple-face-detection-and-recognition-in-real>.



12 pav. Realus vaizdas (dešinėje) ir Corner detection apdorotas vaizdas (kairėje) ^[12]

2.6. Harris corner detector

Harris ir *Stephens* patobulino šį algoritmą iš įprasto kampų radimo algoritmo [24]. Tarkim, kad pradinį vaizdą pažymėsime I , tai pasirinkus vaizdo fragmentą iš srities (u, v) ir pastūmus jį per (x, y) , gaunamos kvadratinės skirtumų sumos tarp paminėtų dviejų sričių, jei pažymėsime sumą S , tuomet ji gaunama pagal formulę [25]:

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

Harris matrica „ A “ yra gaunama suapvalinus sumą „ S “ naudojant antros eilės Teiloro eilutės skleidinį [26]:

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

, čia ∇S ir A žymi perėjimo vektorių ir antrųjų S išvestinių matricas.

Tuomet kampas yra apibūdinamas kaip didelis S nuokrypis (*ang. variation*) visomis vektoriaus (x, y) kryptimis. Nagrinėjant tikrines A vertes, paminėtas apibūdinimas gali būti aprašomas taip:

1. A turi turėti du didelius koeficientus (tikrines vertes) reikšmingame taške. Atsižvelgiant į šių koeficientų dydžius, galima padaryti tokias išvadas:

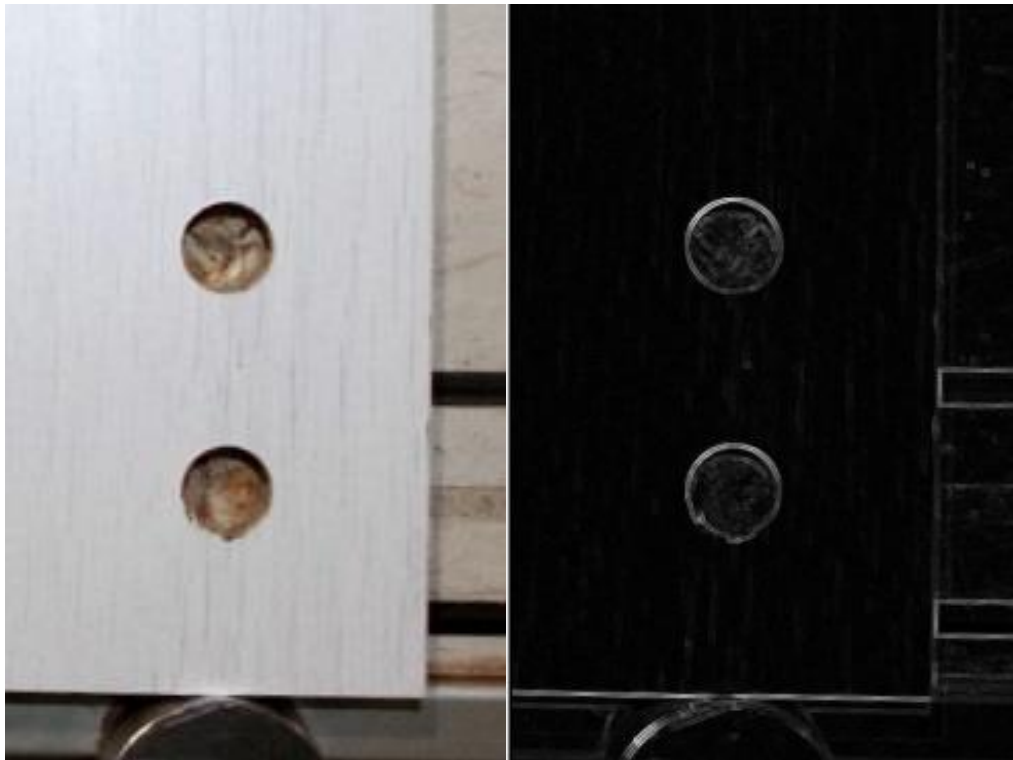
¹² *AForge.Imaging.Filters Namespace* In: *AForge.NET Framework* [žiūrėta 2015 m. vasario 8d.] Prieiga per internetą: <http://www.aforge.net/aforge/framework/docs/html/cdf93487-0659-e371-fed9-3b216efb6954.htm>.

- Jei $\lambda_1 \approx 0$ ir $\lambda_2 \approx 0$ tai tiriamojo vietoje kampas nerastas;
- Jei $\lambda_1 \approx 0$, o λ_2 yra teigiama reikšmė – tiriamojoje vietoje surasta kampo kraštas ar objekto kraštinė;
- Jei λ_1 ir λ_2 yra didelės, teigiamos ir skirtingos reikšmės – tai tiriamojoje vietoje surastas kampas.

Tikslus koeficientų apskaičiavimas reikalauja daug laiko, todėl *Harris corner detector* algoritme naudojama toliau aprašyta funkcija M_c , kur yra reguliuojamas jautrumo parametras (κ) [27]:

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

κ - reikšmė yra nustatyta bandymų metodu.



13 pav. Realus vaizdas (dešinėje) ir Harris corner detector apdorotas vaizdas (kairėje) ¹³

¹³ Souza, Cesar. *Harris Corners Detector in C#* (2010.05.12) In: *Accord.Net Framework. Cesar Souza blog for sharing knowledge efficiently*. [žiūrėta 2015 m. sausio 5 d.] Panaudotas programos kodas Harris filtrui. Prieiga per internetą: <http://crsouza.com/2010/05/harris-corners-detector-in-c/>.

2.7. Vaizdų palyginimas

Tiriamų vaizdų palyginimui atlikti naudojama tiesinė Pirsono koreliacija. Koreliacijos funkcija vaizdai yra apskaičiuojama pagal sekančią lygtį:

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

Koreliacija – statistinis ryšys tarp dviejų kintamųjų, reiškinių sąryšis. Terminą *koreliacija* pirmas biologijoje panaudojo Kiuvje (1806). Matematiškai aprašė prancūzas O.Brave (1846). Brave buvo kristalografas, jis Gauso dėsnį pritaikė dvimačiui atvejui.

Koreliacinė funkcija parodo ar du kintamieji, šiuo atveju vaizdai, turi ryšį ir koks to ryšio stiprumas, t.y. kiek panašūs yra tiriamieji vaizdai. Tai parodo koreliacijos koeficientas, funkcijos reikšmė, tai yra koreliacijos stiprumo matas [29]. Jeigu dviejų tiriamųjų vaizdų koreliacijos koeficientas lygus nuliui, tai tie vaizdai neturi jokio ryšio ir yra statistiškai nepriklausomi. Žemiau pateikta koreliacijos koeficiento reikšmių skalė (Lentelė 1) :

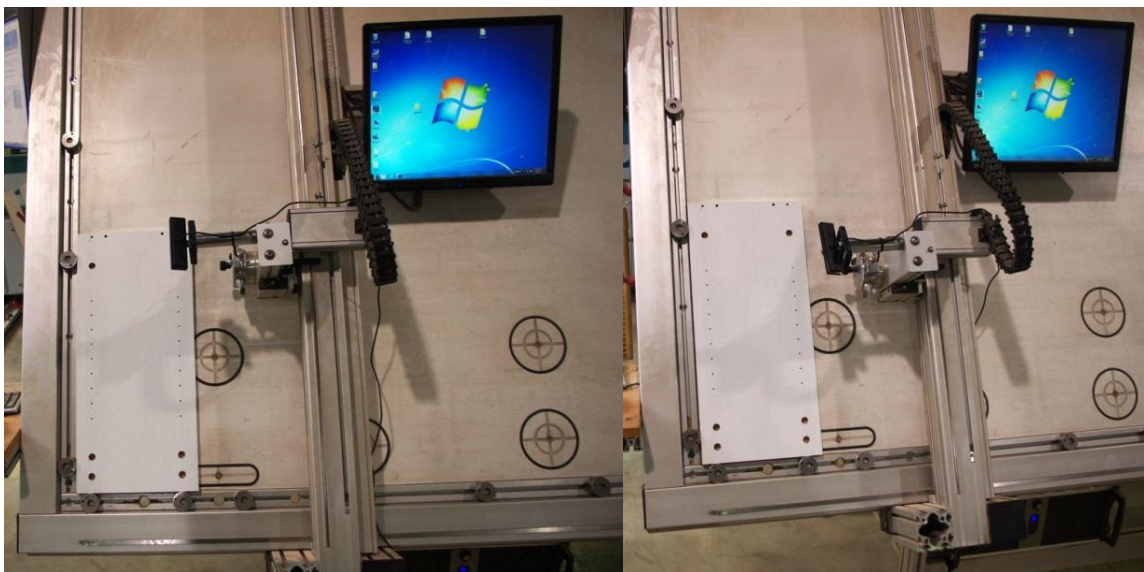
Lentelė 1: Koreliacijos koeficiento reikšmės.

Koreliacijos koeficiento reikšmių skalė										
Labai stipri	Stipri	Vidutinė	Silpna	Labai silpna	Nėra ryšio	Labai silpna	Silpna	Vidutinė	Stipri	Labai stipri
-1	nuo -1 iki -0,7	nuo -0,7 iki -0,5	nuo -0,5 iki -0,2	nuo -0,2 iki 0	0	nuo 0,2 iki 0	nuo 0,5 iki 0,2	nuo 0,7 iki 0,5	nuo 1 iki 0,7	1

3. MEDIENOS DROŽLIŲ PLOKŠTĖS KOKYBĖS TIKRINIMO SISTEMA

3.1. Techninė įranga

Tyrimas buvo atliekamas baldų gamyklos patalpose, UAB „IKEA Industry Lietuva“, įmonė skyrė tiek finansinę paramą techninei įrangai įsigyti, t.y. - foto kameros, šviestuvai, taip pat ir dalį laboratorinės įrangos: matavimo stendą, kompiuterinę įrangą. Bandymams atlikti buvo sumontuotas pastovus darbo vietos apšvietimas, apie 800lm. Iš įmonės lėšų buvo suprojektuotas ir pagamintas laminuotos medienos drožlių plokštės sugręžtų skylių kokybės tikrinimo stendo prototipas.



14 pav. Kokybės tikrinimo stendo prototipas

Stendo veikimo principas:

- Gaminys pastatomas į stacionarų rėmelį taip, kad jo atstumas bei pasukimo kampas nesikeistų;
- Vaizdui gauti naudojama internetinė didelės skiriamosios gebos, plataus kampo *GENIUS WIDECAM F100 HD* vaizdo kamera, prijungta prie kompiuterio per USB sąsaja;
- Vaizdui apdoroti naudojamas kompiuteris su specialiai sukurta gaminio kokybės patikrinimui programine įranga.

Norint gauti kokybišką tiriamo objekto vaizdą, būtina sumažinti aplinkos įtaką, todėl reikalingas geras apšvietimas.

3.2. Programinė įranga

Programinė įrangos prototipas laminuotų medienos plokščių kokybės vertinimui buvo kuriamas naudojant C# programavimo kalbą, naudotas nemokamas *Microsoft Visual C# 2010 Express* kompiliatorius, *Aforge.NET* biblioteka bei kitus viešus šaltinius [28] ir atvirus programų kodus¹⁴.

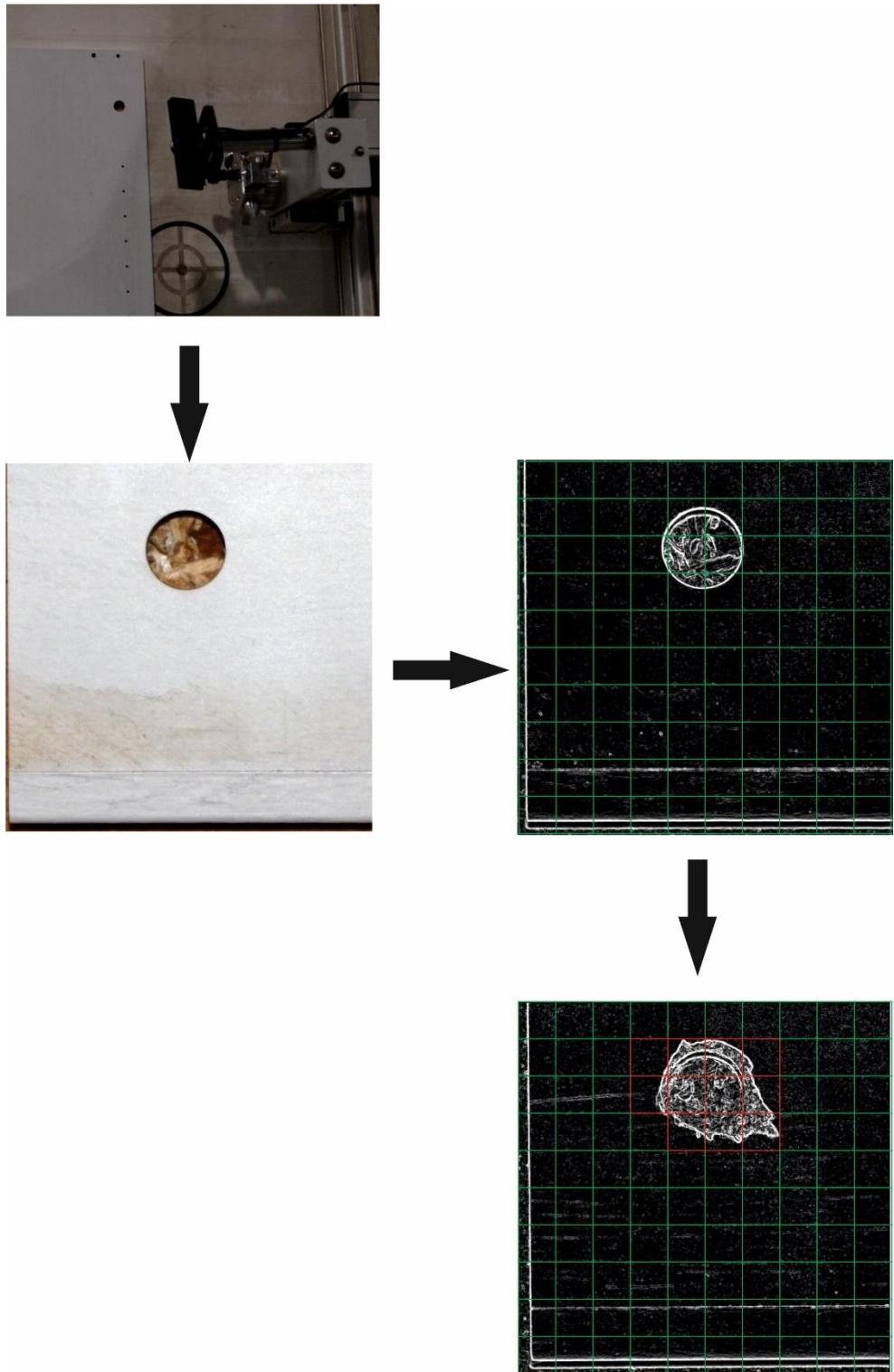
Programinės įrangos veikimo algoritmas: pirmiausia kamera nufotografuoja tikrinamos medienos drožlių plokštės atvaizdą, vaizdas apdorojamas pasirinktu metodu ir pateikiami analizės (nuskaitymo) rezultatai. Aptikus neatitikimą, programinė įranga įspėja sistemos operatorių apie tikrinamos baldinės plokštės defektus, tuomet išsaugo ir pateikia programos lange plokštės vaizdą su pažymėtomis vietomis.

Programos veikimą galima suskaidyti į šiuos etapus (15pav.):

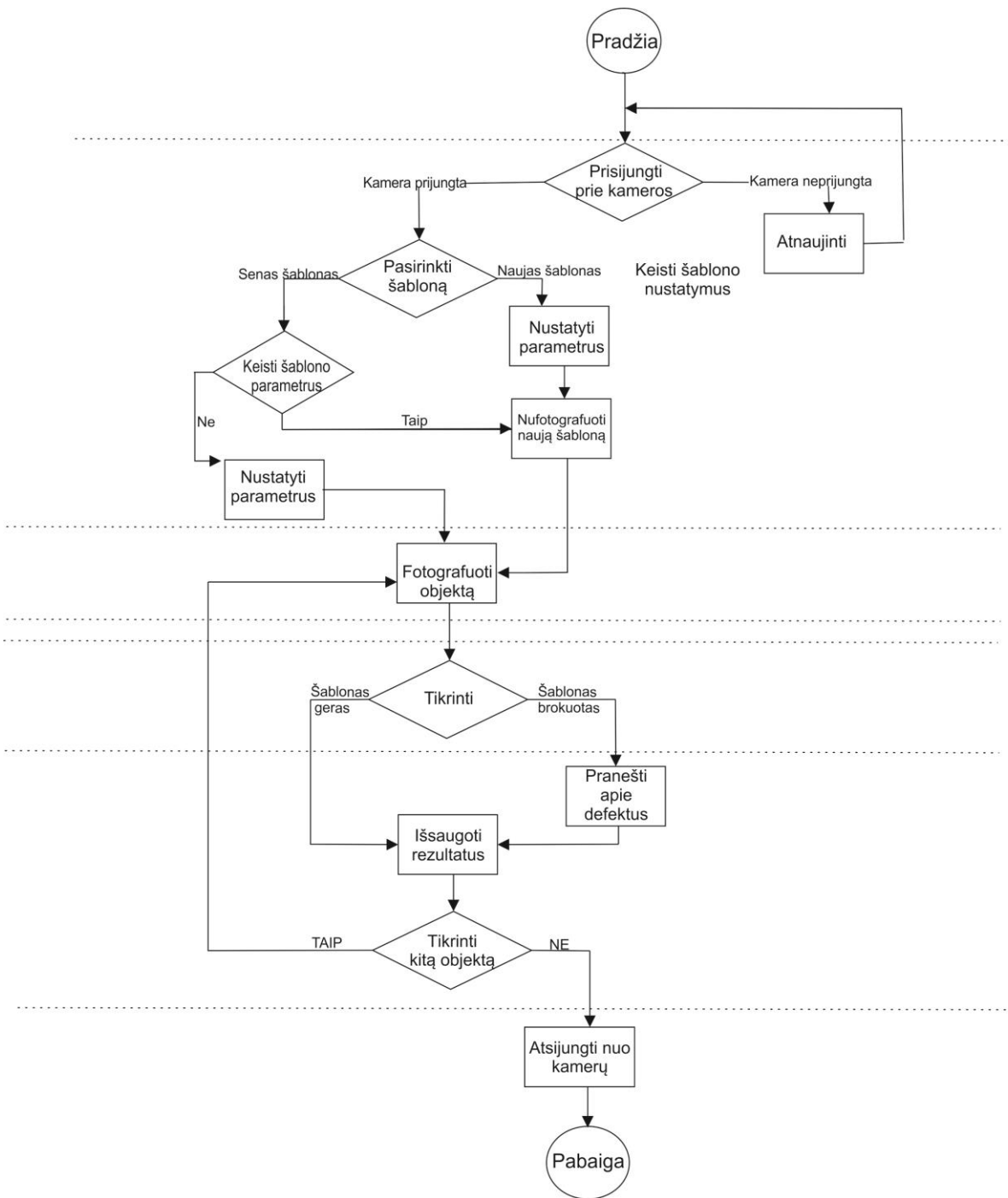
1. Objekto (plokštės paviršiaus) vaizdo užfiksavimas, tai sąsaja su kamera. Šiame etape yra prisijungiama prie kameros(ų), nustatomi arba pasirenkami kameros parametrai, tada nufotografuojama plokštė ir gautas vaizdo parsiončiamas į kompiuterį apdorojimui.
2. Pasirenkamas algoritmas, nustatomi jo parametrai, apdorojimas gautas vaizdas ir ieškoma neatitikimų su etalonu.
3. Rezultatų pateikimas programos lange.

Paleidus programą, galima prisijungti prie vaizdo kameros, taip pat pasirinkti kamerą (tobulinant sistemą, atsiras galimybė prisijunti prie grupės kamerų) (16 pav.). Jei kameros neprijungtos prie sistemos kompiuterio ar dėl kitokių priežasčių neįmanoma prisijungti, vartotojas apie tai informuojamas. Su tikslu, kad naudojama sistema bus norima tikrinti ne vieną gaminį, todėl numatyta galimybė kiekvienam tiriamam gaminiui pasirinkti skirtingus parametrus, tiek kameros, tiek vaizdo apdorojimo ir kitus. Paleidus programą, jei visos sąlygos išpildytos, t.y. kameros pajungtos, tuomet užkraunami paskutinį kartą naudoto šablono parametrai. Jei analizuojamas kitas gaminytis, pasirenkamas kitas jau sukurtas to gaminio šablonas, jei tokio nėra - sukuriamas naujas.

¹⁴ Dr.Bharadi, Vinayak Ashok. *Canny Edge Detection in C#* In: *Code Project, For those who code* [žiūrėta 2015 m. sausio 14 d.] Panaudotas filtravimo programos kodas. Prieiga per internetą: <http://www.codeproject.com/Articles/93642/Canny-Edge-Detection-in-C>.

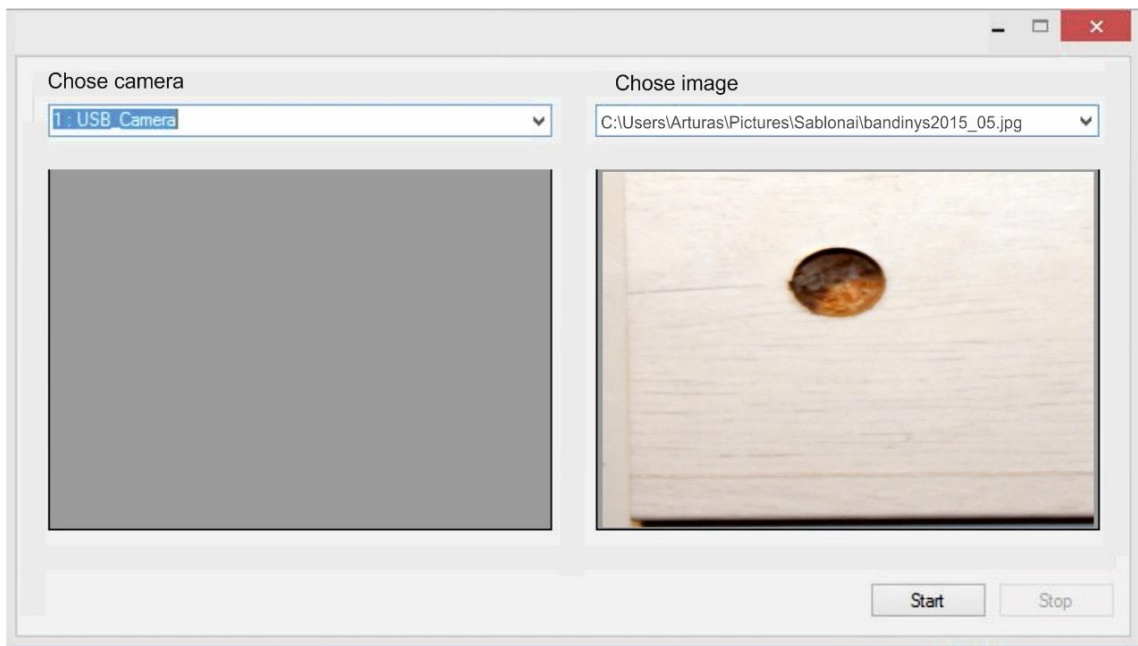


15 pav. Programinės įrangos veikimo principas



16 pav. Programinės įrangos veikimo blokinė schema

Užfiksuotas kameros vaizdas persiunčiamas į kompiuterį. Kiekvienam gautam vaizdui suteikiamas unikalus pavadinimas. Naujai gautas paveikslas, palyginamas su prieš tai padarytu to paties gaminio vaizdu, t.y. šablonu. Radus neatitikimų, vartotojas apie tai informuojamas. Baigus darbą su programa, atsijungiama nuo kamerų.

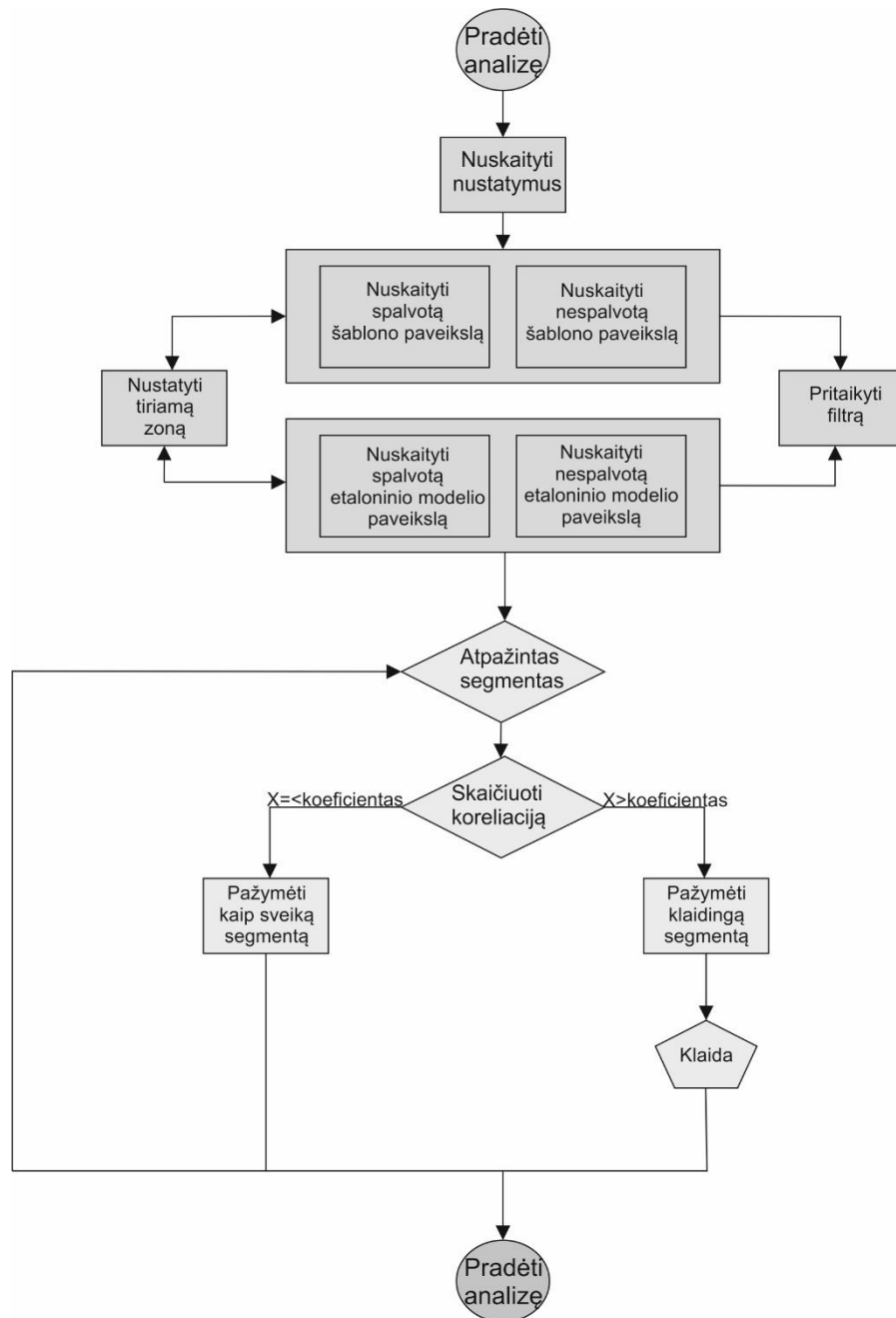


17 pav. Pradinis programos langas ¹⁵

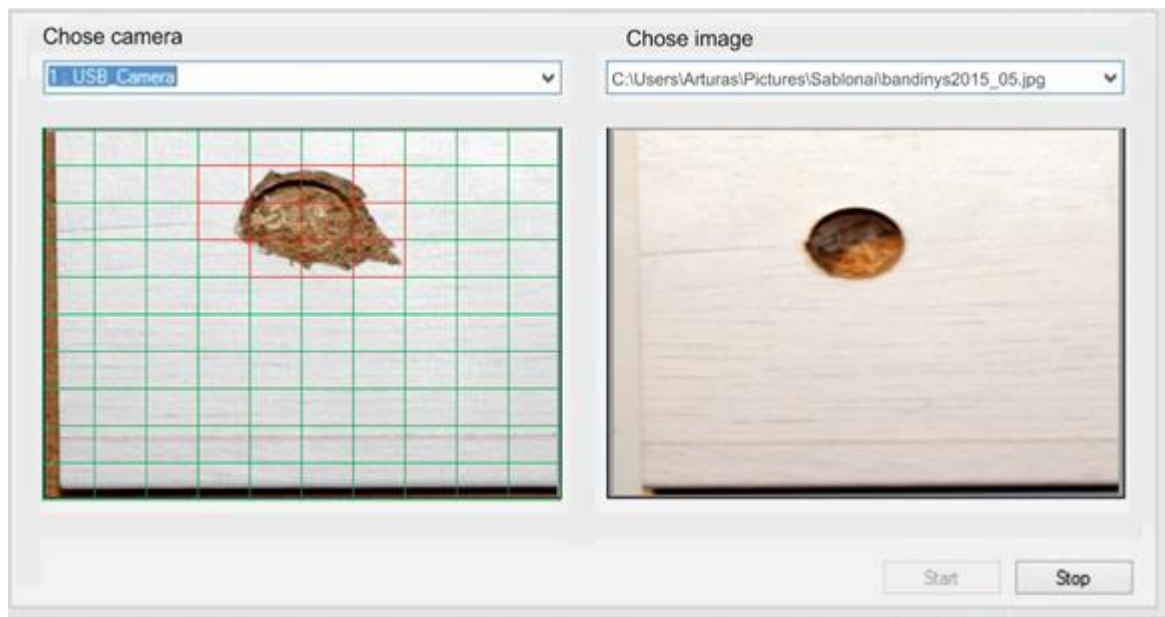
Kompiuterinio vaizdo analizės algoritmas (18 pav.):

- Gaunamas etaloninis modelis, sugręžtos plokštės vaizdas;
- Etaloninis modelis apdorojamas pasirinktu vaizdo apdorojimo algoritmu;
- Gaunamas tiriamo modelio vaizdas;
- Tiriamojo modelio vaizdas apdorojamas tokiu pat kaip ir etaloninio modelis vaizdo apdorojimo algoritmu;
- Pagal tai koks yra parinktas segmento dydis, nustatoma segmento vieta abiejuose atvaizduose ir suskaičiuojamas koreliacijos koeficientas;
- Palyginus su nustatytu koreliacijos koeficiento dydžiu, segmentas pažymimas kaip tinkamas, t.y. kokybiškas (žalia spalva) arba kaip netinkamas - defektuotas (raudona spalva) (19 pav.).

¹⁵ Stephens, Rod. *Compare two images to find differences in C#*. In: *C# Helper*. [žiūrėta 2015 m. balandžio 20 d.]. Prieiga per internetą: <http://csharp-helper.com/blog/2011/01/compare-two-images-to-find-differences-in-c/>.



18 pav. Vaizdo analizės algoritmas



19 pav. Vaizdo analizės rezultatas¹⁶

Programinė įranga, koreliaciją atlieka naudojant Aforge.NET bibliotekos funkciją *ExhaustiveTemplateMatching*¹⁷.

4. SISTEMOS TIKRINIMAS. EFEKTYVUMO TYRIMAS

Sistemos tikrinimas vykdomas tokia tvarka: imama tikrinamas gaminys (plokštė), jis patalpinamas į rėmelį ir nufotografuojamas du kartus. Pirmą nuotrauką panaudojama kaip etaloninis plokštės vaizdas, antra – kaip nedefektos plokštės vaizdas. Po to gaminys tikslingai sugadinamas, pažeidžiamas laminato sluoksnis, deformuojama išgręžta skylė ir tuomet plokštė dar kartą nufotografuojama. Taip gaunamas defektuotos plokštės nuotrauka. Fotografavimas ir gautų vaizdų apdorojimas vykdomas tik naudojant specialiai sukurtą programinę įrangą kokybės tikrinimui. Kamera sumontuota stacionariai, jos pozicija nekeičiama testuojamos plokštės atžvilgiu, vaizdo parametrai (filtrai) taip pat keičiami naudojant programą, tokiu būdu imituojamas realios sistemos darbas.

Toliau grafikuose pateikti rezultatai apie klaidingai identifikuotus gaminio neatitikimus etaloniniam vaizdai: x ašyje pažymini ir atidedami keičiami algoritmų parametrai, o y ašyje – santykinis klaidingai identifikuotų defektų skaičius procentais, t.y. realių defektų nebuvo, tačiau programa vis tiek apskaičiuoja ir įvertina kaip neatitinkimą etaloniniam modeliui). Taip nutinka

¹⁶ Krarup, Jacob. *Simple image comparison in .NET*. In: *Code Project. For those who code*. [žiūrėta 2015 m. balandžio 15 d.]. Prieiga per internetą: <http://www.codeproject.com/Articles/374386/Simple-image-comparison-in-NET>.

¹⁷ AForge.NET biblioteka. *ExhaustiveTemplateMatching Class*. In: *Aforge.NET Framework*. [žiūrėta 2015 m. sausio 9 d.]. Prieiga per internetą: <http://www.aforge.net.com/framework/docs/html/17494328-ef0c-dc83-1bc3-907b7b75039f.htm>.

dėl laminato plėvelės faktūros, raštų intensyvumo ir šešėlių (kai nenaudojamas stiprus apšvietimas).

Santykinis klaidingai identifikuotų defektų skaičius procentais apskaičiuojamas pagal formulę:

$$rez = \frac{(obj - obj_kld)}{segm} * 100\% , \text{ čia}$$

obj - neatitikimų skaičius;

obj_kld - klaidingai nustatytų neatitikimų skaičius leistina paklaida;

segm - bendras segmentų (skylių kadru) skaičius.

4.1. Kameros įtaka vaizdo tyrimui

Pasirinkti vaizdo analizės algoritmai apdoroja nespaltvotus (*angl. grey scale*) vaizdus, todėl galima daryti prielaidą, kad geriausiai būtų nustatyti kamera į nespaltvoto fotografavimo režimą. Iš pradžių tikrinamas gaminys yra be pažeidimų, todėl teoriškai visi segmentai turi būti identifikuoti kaip atitinkantys etaloną. Vaizdo analizei naudojami šie vaizdo apdorojimo metodai: *Sobel operator*, *Harris corner detector*, *Difference edge detector* ir *Canny edge detector*. Taip pat tyrimui paimami skirtingi tiriamo vaizdo dydžiai ir skirtingi koreliacijos koeficientai, bet nemažesni nei 0,8.

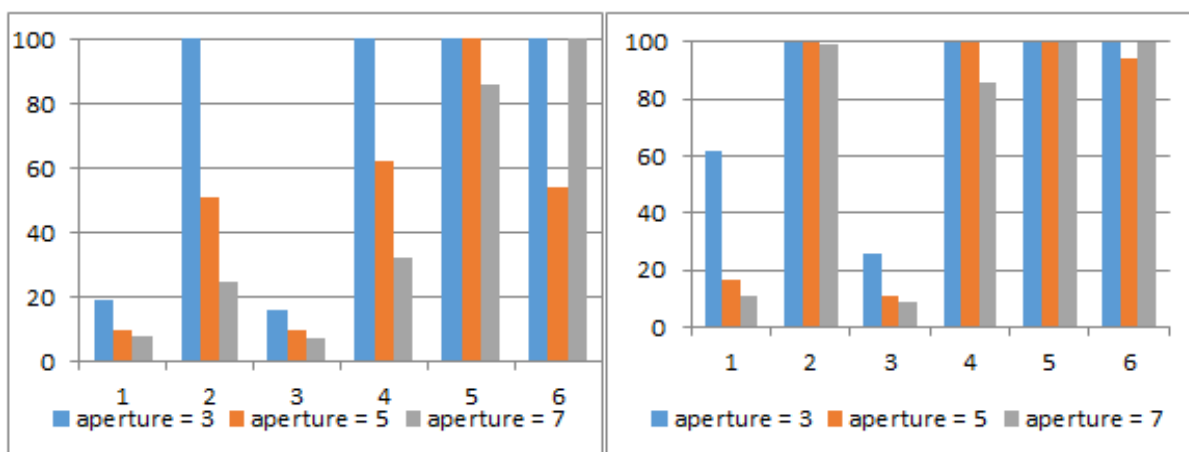
Klaidingai identifikuotų segmentų skaičius buvo didelis, naudojant *Sobel* algoritimą (1 lentelė). Visi segmentai identifikuoti teisingai, tik tuomet kai yra nustatyti tam tikri algoritmo parametrai. Parinkus didesnę segmentą ir mažesnę koreliacijos koeficientą, daroma mažiau klaidų. Segmentai, kurie tiriami spalvotame vaizde buvo identifikuojami kur kas tiksliau.

Naudojant *Canny edge detector* algoritimą buvo neteisingai identifikuota nuo 80% iki 100% segmentų, todėl jis pasirodė netinkamas atpažįstant defektus (2 lentelė- rezultatai, kai segmento dydis 480x480 pikselių ir koreliacijos koeficientas 0.9). Todėl išsiaiškinta, kad *Canny edge detector* algoritmas yra itin jautrus net mažiausiems pokyčiams. Yra daroma prielaida, kad nedideli fotografuojamo vaizdo pakitimai galimai atsirado dėl nevienodai pasiskirsčiusio apšvietimo. Todėl yra svarbu akcentuoti, kad pakankamai ryškus ir šviesus vaizdas gaunamas tik padidinus kameros jautrumą. Tiriant spalvotą vaizdą gauti geresni rezultatai.

Taip pat didelė dalis segmentų buvo identifikuoti kaip neatitinkantys etalonio modelio naudojant *Corner detection* algoritimą (22, 23, 24, 25pav). Imant didesnę segmento dydį ir mažesnę koreliacijos koeficientą bei tiriant spalvotą vaizdą, gauti geresni rezultatai. Tiriamas spalvotas vaizdas buvo naudojant tokius parametrus: *aperture = 7* (*diafragmos atidarymas*), segmento dydis 640x640px, koreliacijos koeficientas 0.8. Nustatyta, kad sistemos paklaida buvo iki 5,5 % (3 lentelė).

1 lentelė. Klaidingai identifikuotų neatitikčių skaičius naudojant Sobel operatorių. Koreliacijos koeficientas 0,8. Tiriama vaizdo dydis 480x480px.

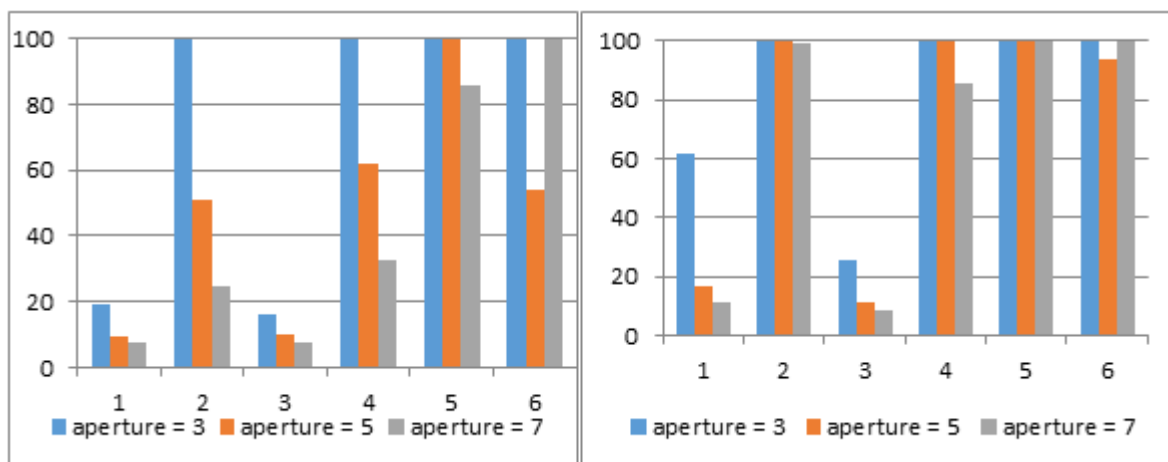
Nustatymai	Klaidingai identifikuotų neatitikčių skaičius, vnt		Klaidingai identifikuotų neatitikčių skaičius, %	
	Spalvotas	Nespalvotas	Spalvotas	Nespalvotas
int x = 0; int y = 0; aperture = 3	123	38	61,75%	19%
int x = 0; int y = 0; aperture = 5	33	20	16,5%	9,75%
int x = 0; int y = 0; aperture = 7	23	16	11,25%	7,75%
int x = 0; int y = 1; aperture = 3	200	200	100%	100%
int x = 0; int y = 1; aperture = 5	200	102	100%	51%
int x = 0; int y = 1; aperture = 7	198	50	99,25%	25%
int x = 1; int y = 0; aperture = 3	53	32	25,75%	16%
int x = 1; int y = 0; aperture = 5	22	20	11,25%	10%
int x = 1; int y = 0; aperture = 7	17	15	8,5%	7,5%
int x = 1; int y = 1; aperture = 3	200	200	100%	100%
int x = 1; int y = 1; aperture = 5	200	127	100%	62,25%
int x = 1; int y = 1; aperture = 7	171	65	85,5%	32,5%
int x = 1; int y = 2; aperture = 3	200	200	100%	100%
int x = 1; int y = 2; aperture = 5	200	200	100%	100%
int x = 1; int y = 2; aperture = 7	200	172	100%	86%
int x = 2; int y = 2; aperture = 3	200	200	100%	100%
int x = 2; int y = 2; aperture = 5	188	108	94%	54%
int x = 2; int y = 2; aperture = 7	200	200	100%	100%



20 pav. Tyrimo su Sobel rezultatai. Koreliacijos koeficientas 0,8. Dešinėje spalvoto vaizdo analizė, kairėje – nespalvoto.

2 lentelė. Klaidingai identifikuotų neatitikčių skaičius naudojant Canny edge detection algoritmą. Koreliacijos koeficientas 0,9. Tiriama vaizdo dydis 480x480px.

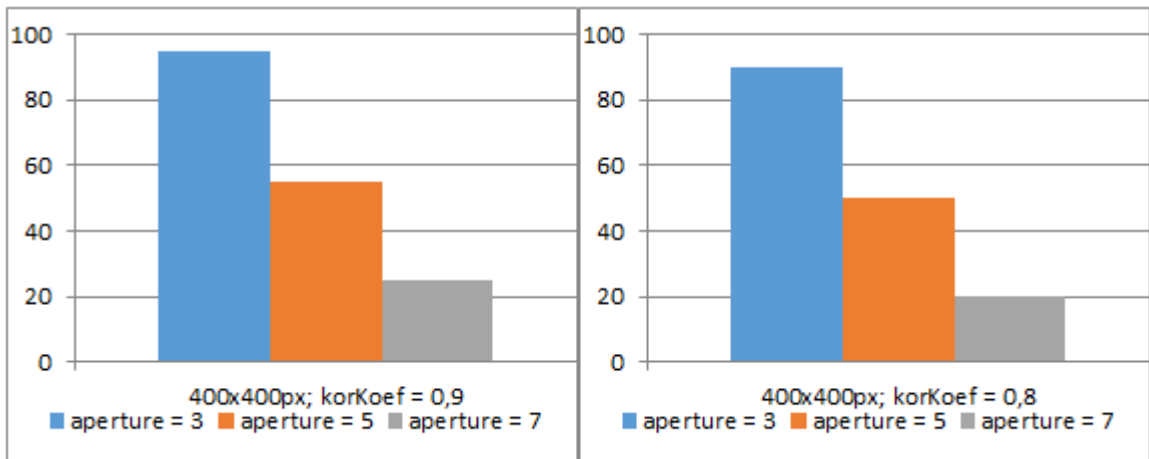
Nustatymai	Klaidingai identifikuotų neatitikčių skaičius, vnt		Klaidingai identifikuotų neatitikčių skaičius, %	
	Spalvotas	Nespalvotas	Spalvotas	Nespalvotas
int x = 0; int y = 0; aperture = 3	38	123	19%	61%
int x = 0; int y = 0; aperture = 5	20	33	9,75%	16,5%
int x = 0; int y = 0; aperture = 7	16	23	7,75%	11,25%
int x = 0; int y = 1; aperture = 3	200	200	100%	100%
int x = 0; int y = 1; aperture = 5	102	200	51%	100%
int x = 0; int y = 1; aperture = 7	50	198	25%	99,25%
int x = 1; int y = 0; aperture = 3	32	53	16%	25,75%
int x = 1; int y = 0; aperture = 5	20	22	10%	11,25%
int x = 1; int y = 0; aperture = 7	15	17	7,5%	8,5%
int x = 1; int y = 1; aperture = 3	200	200	100%	100%
int x = 1; int y = 1; aperture = 5	127	200	62,25%	100%
int x = 1; int y = 1; aperture = 7	65	171	32,5%	85,5%
int x = 1; int y = 2; aperture = 3	200	200	100%	100%
int x = 1; int y = 2; aperture = 5	200	200	100%	100%
int x = 1; int y = 2; aperture = 7	172	200	86%	100%
int x = 2; int y = 2; aperture = 3	200	200	100%	100%
int x = 2; int y = 2; aperture = 5	108	188	54%	94%
int x = 2; int y = 2; aperture = 7	200	200	100%	100%



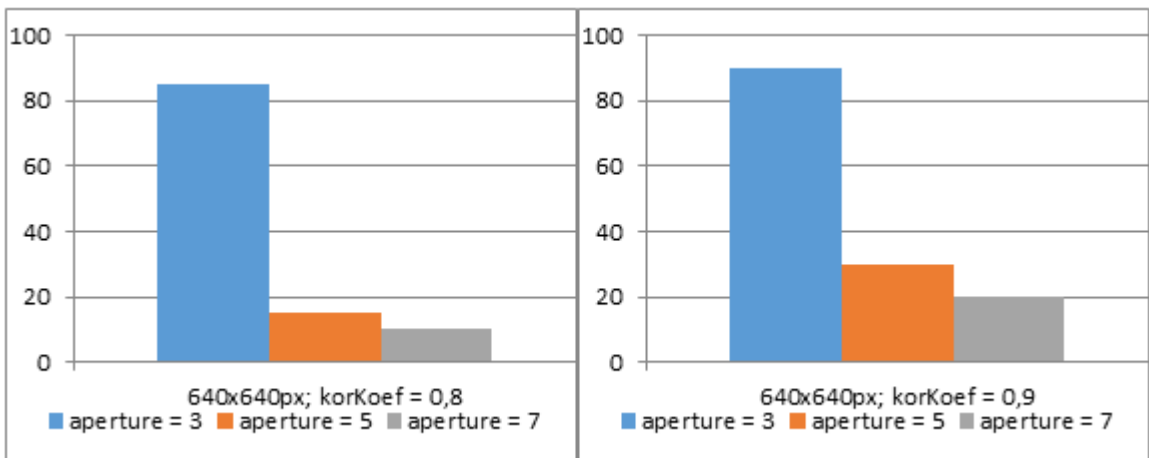
21 pav. Klaidingai identifikuotų neatitikčių skaičius naudojant Canny edge detection algoritmą. Kairėje spalvoto vaizdo analizė, dešinėje – nespalvoto.

3 lentelė. Klaidingai identifikuotų neatitikčių skaičius naudojant Corner detection algoritmą.

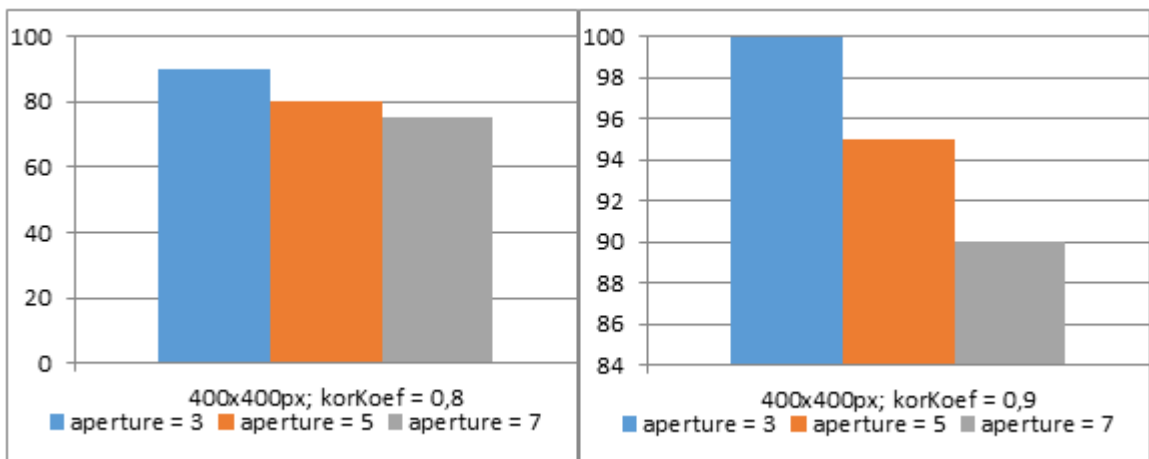
Nustatymai		Klaidingai identifikuotų neatitikčių skaičius, %	
		Spalvotas	Nespalvotas
400x400px; korKoef = 0,9	korKoef = 0,8; aperture = 3	95	100
	korKoef = 0,8; aperture = 5	55	95
	korKoef = 0,8; aperture = 7	25	90
400x400px; korKoef = 0,8	korKoef = 0,9; aperture = 3	90	90
	korKoef = 0,9; aperture = 5	50	80
	korKoef = 0,9; aperture = 7	20	75
640x640px; korKoef = 0,9	korKoef = 0,8; aperture = 3	90	100
	korKoef = 0,8; aperture = 5	30	80
	korKoef = 0,8; aperture = 7	20	17
640x640px; korKoef = 0,8	korKoef = 0,9; aperture = 3	85	100
	korKoef = 0,9; aperture = 5	15	70
	korKoef = 0,9; aperture = 7	10	10



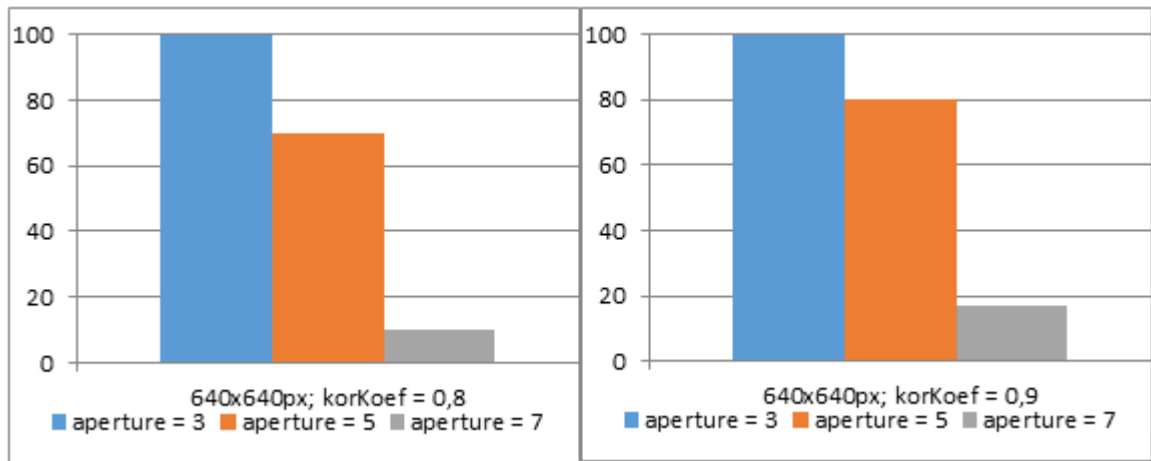
22 pav. Klaidingai identifikuotų segmentų procentinis kiekis. Spalvotų vaizdų 400x400px.



23 pav. Klaidingai identifikuotų segmentų procentinis kiekis. Spalvotų vaizdų 640x640px.



24 pav. Klaidingai identifikuotų segmentų procentinis kiekis. Nespalvotų vaizdų 400x400px.



25 pav. Klaidingai identifikuotų segmentų procentinis kiekis. Nespalvotų vaizdų 640x640px.

Ištirta, kad iki 97% segmentų buvo identifikuoti klaidingai, kada spalvotam vaizdai apdoroti buvo naudojamas *Harris corner detector* algoritmas, o nespalvotame - iki 48% segmentų. Skaičiavimo rezultatai pateikti 4 lentelėje. (26 pav.) ir (27 pav.) pavaizduoti tyrimo rezultatai kai naudojami šie algoritmo parametrai:

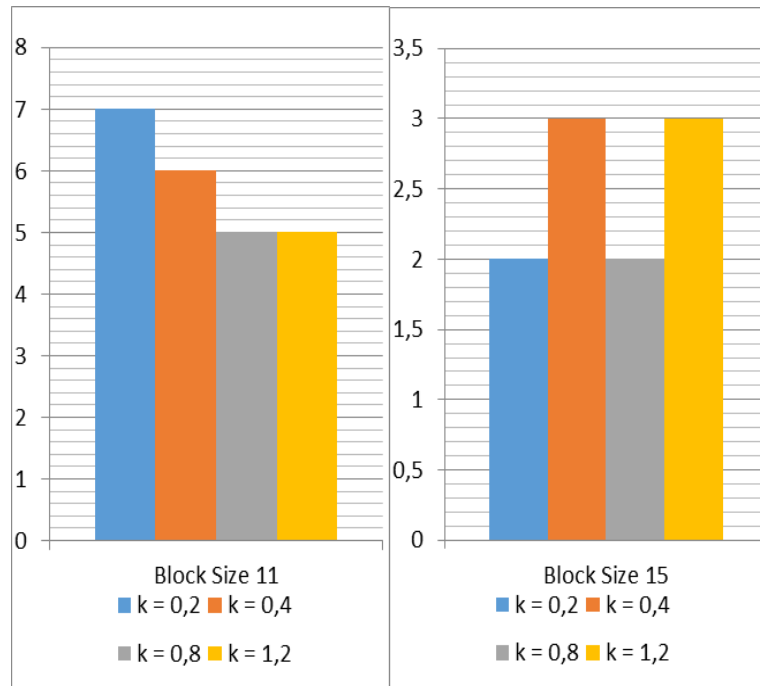
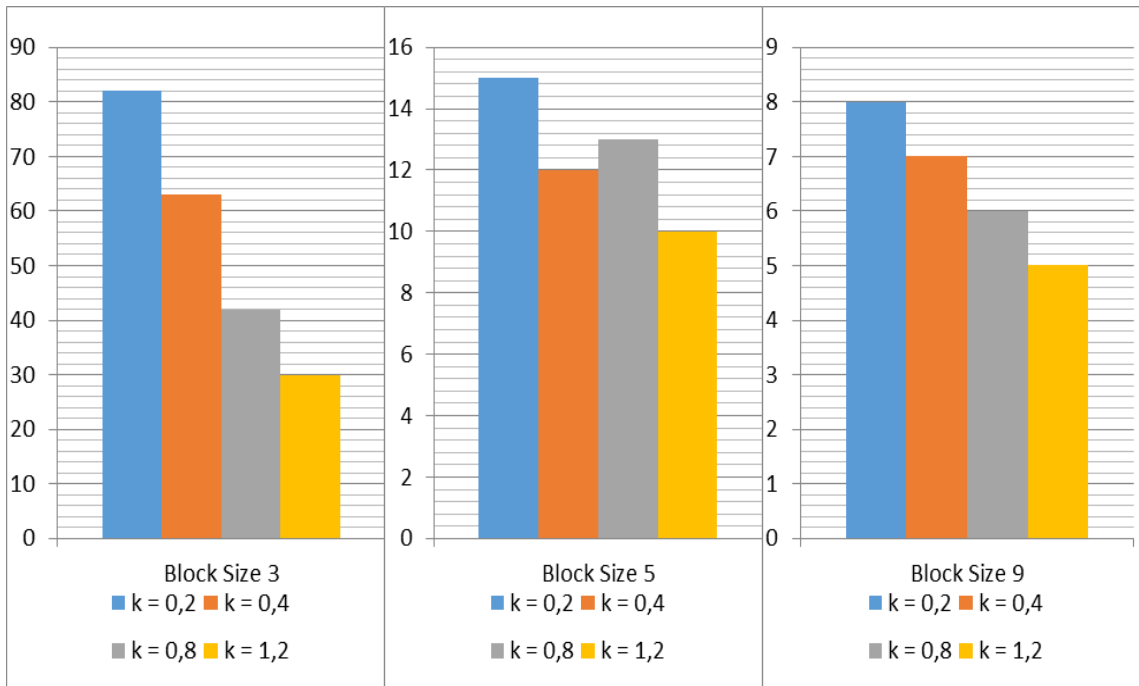
- aperture: 7 – diafragmos atidarymo laikas;
- block size: 3, 5, 9, 11, 15 – tyrimui naudojamo bloko dydis.
- k: 0.2, 0.4, 0.8, 1.2 – jautrumo koeficientas

Naudojamas segmento (išgręžtos skylės) dydis – 150x150px (D=13,4mm), koreliacijos koeficientas 0,9.

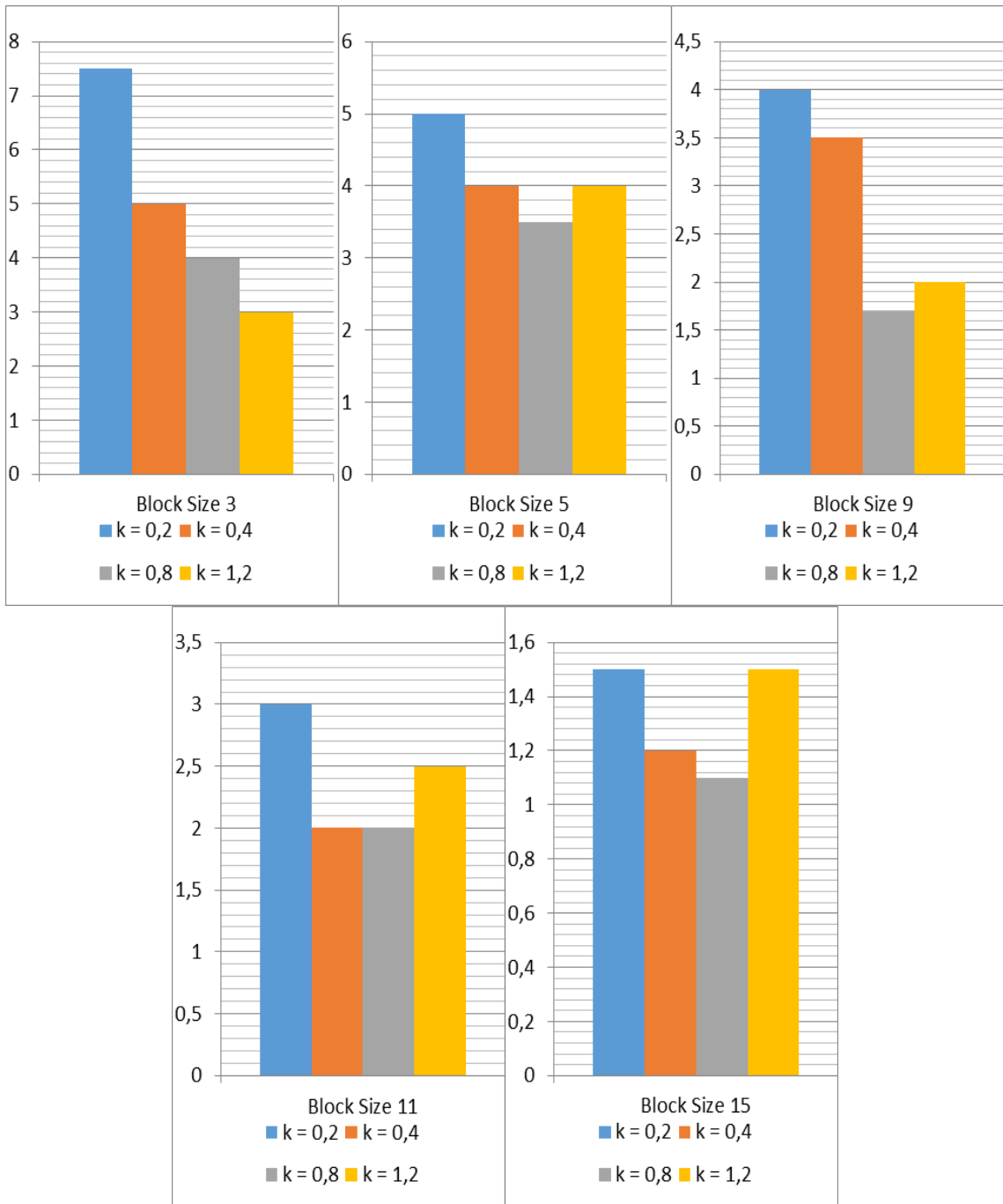
Nustatyta, kad ~1% spalvotame ir ~3% nespalvotame vaizde buvo identifikuoti klaidingai naudojant tokius algoritmo parametrus: block size – 15; k – 0.8; 0.4. (28 pav.).

4 lentelė. Klaidingai identifikuotų neatitikčių skaičius naudojant Harris corner detection algoritmą.

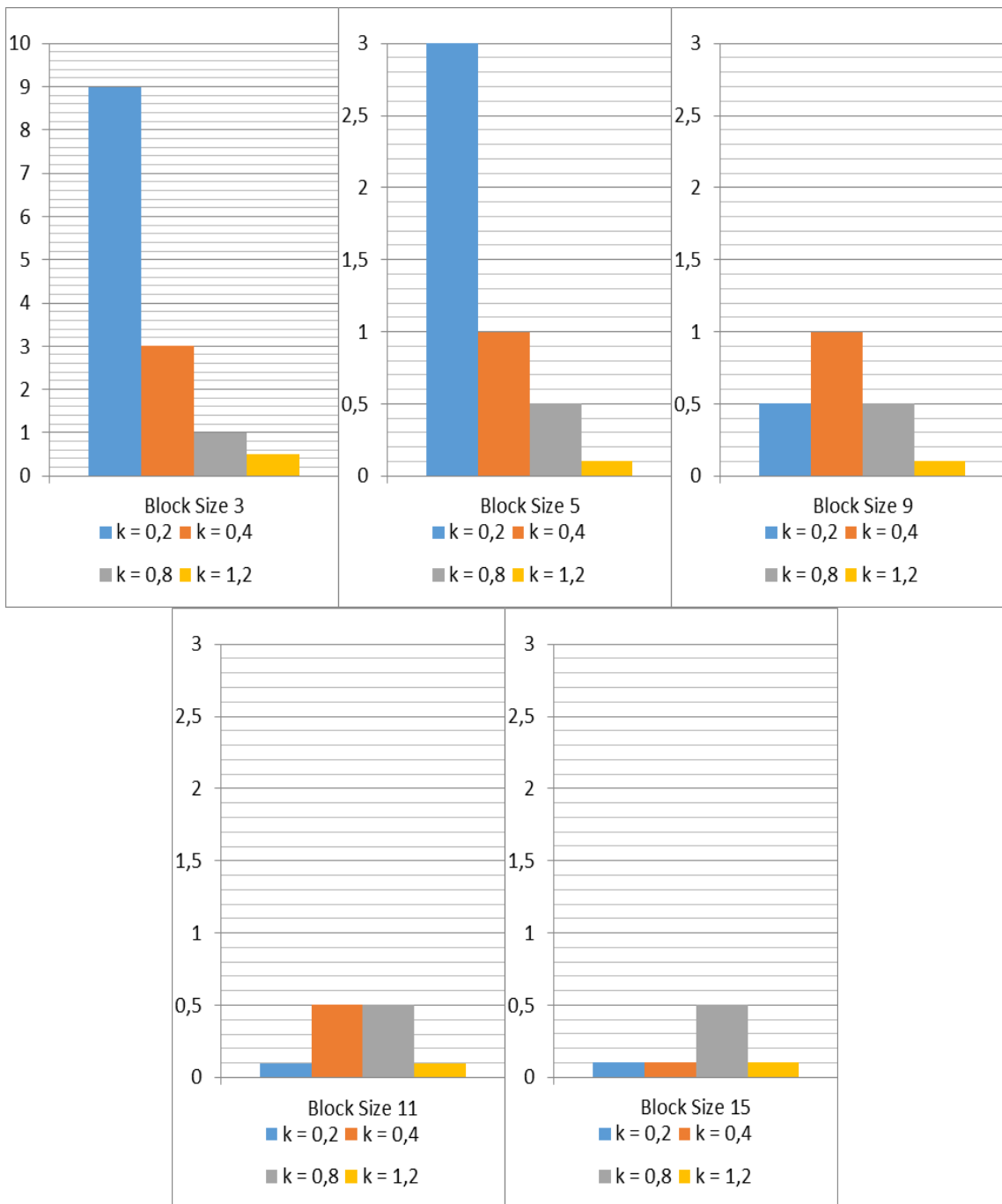
Nustatymai		Klaidingai identifikuotų neatitikčių skaičius, %	
		Nespalvotas	Spalvotas
Block Size 3	k = 0,2	7,5	82
	k = 0,4	5	63
	k = 0,8	4	42
	k = 1,2	3	30
Block Size 5	k = 0,2	5	15
	k = 0,4	4	12
	k = 0,8	3,5	13
	k = 1,2	4	10
Block Size 9	k = 0,2	4	8
	k = 0,4	3,5	7
	k = 0,8	1,7	6
	k = 1,2	2	5
Block Size 11	k = 0,2	3	7
	k = 0,4	2	6
	k = 0,8	2	5
	k = 1,2	2,5	5
Block Size 15	k = 0,2	1,5	2
	k = 0,4	1,2	3
	k = 0,8	1,1	3
	k = 1,2	1,5	3



26 pav. Harris Corner detector tyrimo rezultatai. Spalvotas vaizdas.



27 pav. Harris Corner detector algoritmo tyrimo rezultatai. Nespaltotas vaizdas.



28 pav. Harris Corner detector algoritmo tyrimo rezultatai. Koreliacijos koeficientas 0,8

4.2 Vaizdo kameros parametrų įtaka tyrimui

Gaunamo vaizdo kokybei turi įtakos pagrindiniai fotokameros parametrai: išlaikymas, diafragmos atvėrimas, jautrumas. Vaizdas yra šviesesnis, kada yra didinamas išlaikymo parametras, todėl į kameros vaizdo sensorių patenka daugiau šviesos, dėl šios priežasties gali atsirasti pašaliniai triukšmai. Siekiant išgauti tolygaus šviesumo vaizdą, buvo didinamas ne tik išlaikymo parametras, bet ir mažinamas diafragmos atvėrimas. Sistemos efektyvumui šie parametrai didelės įtakos neturėjo.

Naudojant GENIUS WideCam F100 kamerą, vaizdas buvo sufokusuotas rankiniu būdu, kameros parametrai nustatyti taip, kad būtų kuo mažiau pašalinių triukšmų. Automatinis fokusavimas nors ir duodai geros kokybės vaizdus, tačiau nėra tinkamas, nes tas pats objektas gali būti užfiksuotas skirtingai, priklausomai nuo išorinių veiksnių: apšvietimas, spalvų intensyvumas ir kt. Tokius vaizdus apdorojimo algoritmai atpažintų kaip skirtingus, o programinė įranga identifikuotų klaidingai.

Bandymų metu, kai naudotas papildomas apšvietimas, kameros jautrumas nustatytas - ISO200, nenaudojant apšvietimo, kameros jautrumas padidinamas iki ISO800.

4.3. Algoritmų optimalių parametru nustatymas

Pagal informaciją, kuri yra pateikta rezultatų lentelėse ir grafikuose, nustatyta, kad esant kai kuriems algoritmų parametrams, klaidingai identifikuojamų bandymų skaičius sumažėja. Atliekant bandymus buvo parenkami įvairūs parametrai ir jų rinkiniai, taip siekiant surasti optimaliausią, kiekvienam pasirinktam vaizdo apdorojimo algoritmui.

Be to buvo keičiamas koreliacijos koeficientas (0.9 ir 0.8) ir segmento dydis (75x75, 100x100, 150x150). *Canny edge detection* algoritmas nebuvo naudojamas tolimesniuose tyrimuose, nes ankstesnių bandymų metu buvo išsiaiškinta, kad jis nėra tinkamas dėl didelio neteisingai identifikuotų defektų skaičiaus.

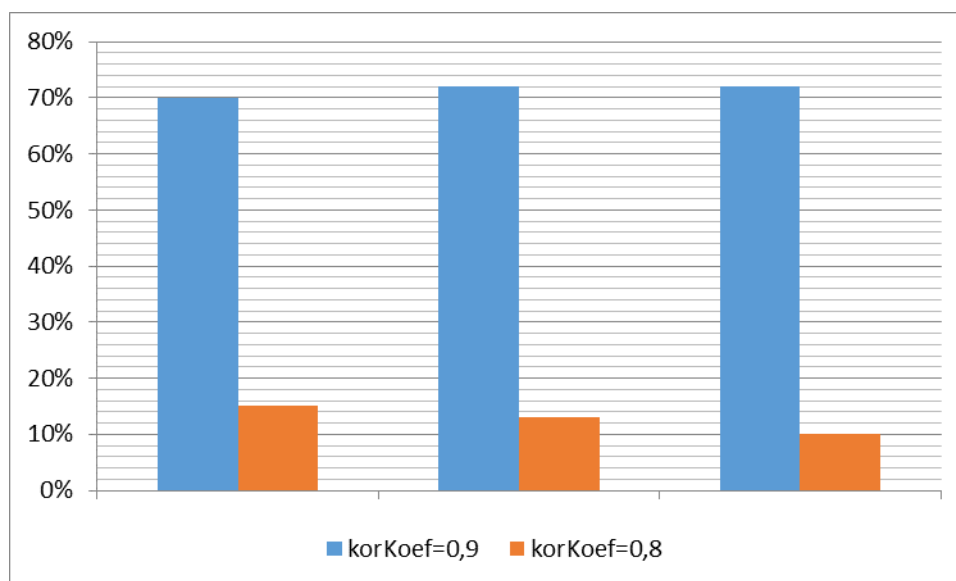
Sobel operator metodas sukurtoje programinėje įrangoje realizuotas naudojant Aforge.NET funkciją *image filter Sobel*. Nododama *Sobel operator* programa apskaičiuoja pirmos, antros ir trečios eilės vaizdo išvestines.

Funkcijos kintamieji:

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

Dėl itin jautraus *Sobel* operatoriaus, net esant mažiausiems neatitikimams, objekto vaizdo segmentas identifikuojamas kaip klaidingas. Pirmiausiai vaizdas buvo apdorotas *Gaussian* filtru ir tik po to *Sobel* operatoriumi, taip siekiant pašalinti galimus triukšmus. Taikant šį operatorių būtina padidinti segmento dydį ir sumažinti koreliacijos koeficientą. Nustatyta, kad geriausi rezultatai gaunami prie tokių *Sobel operator algoritmo* parametru: kai $x = 1$, $y = 0$, $aperture = 7$. (1 lentelė).

Laplace operatorius [16] realizuojamas naudojant Aforge.NET viešą biblioteką internetinėje prieigoje. Iširta, kad nuo 10% iki 70% plokštės segmentai buvo identifikuojami klaidingai, nes algoritmas labai jautrus, bet kokiems pokyčiams. Esant didesniai segmento dydžiui ir mažesniai koreliacijos koeficientui buvo gauti geresni rezultatai. (29pav).



29 pav. Tikrinimo rezultatai Laplace operatoriumi, kai aperture = 7.

Eigen decomposition metodas realizuotas per atviro kodo programą pateiktą C# bibliotekoje ir funkciją iš Accord.Image bibliotekos [20, 21, 22].

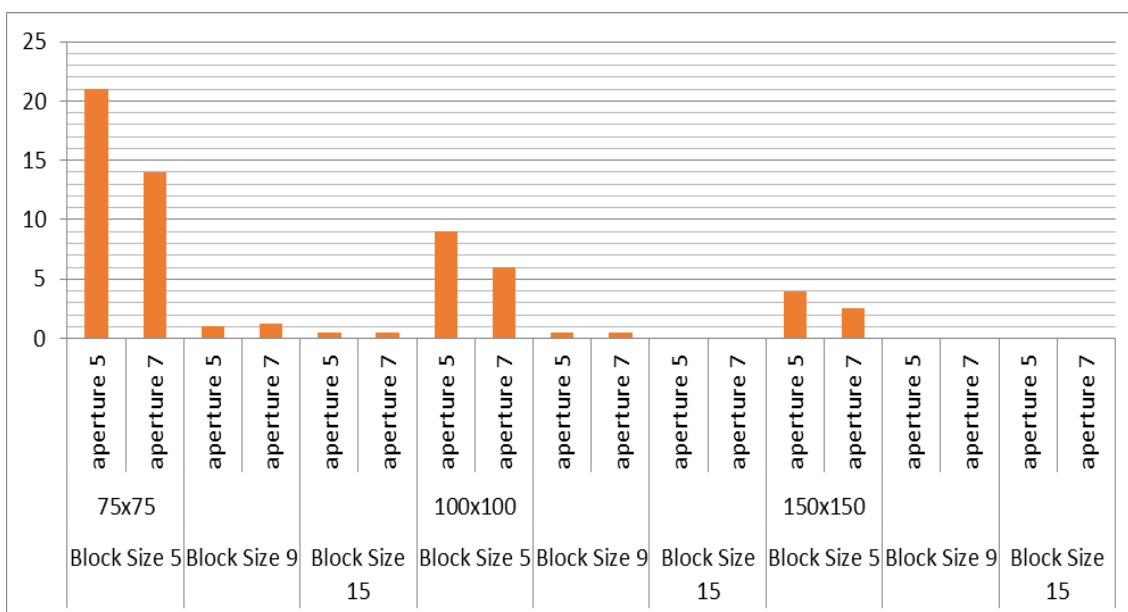
Pagrindiniai parametrai:

- *block_size* - tyrimui naudojamą bloko dydis;
- *aperture* –branduolio dydis. Gali būti 1, 3, 5 ir 7.

Atlikus bandymus paaiškėjo, kad iki 1,3% objekto vaizdų, kai gaminys nepažeistas ir iki 21% - kai gaminys pažeistas, identifikuoti klaidingai esant mažam *aperture* (1, 3) ir mažam *blok_size* parametrui. Rezultatai pagerėja padidinus šiuos parametrus. Išsiaiškinta, kad kai *aperture* = 5 ir 7, *blok_size* = 9, 11 ir 15 buvo gauti geriausi rezultatai (5 lentelė) ir (30 pav.).

5 lentelė. Klaidingai identifikuotų neatitikčių skaičius naudojant Eigen decomposition algoritmą. Koreliacijos koeficientas – 0,9.

Nustatymai			Klaidingai identifikuotų neatitikčių skaičius, %
Block Size 5	75x75	aperture 5	21
		aperture 7	14
Block Size 9		aperture 5	1
		aperture 7	1,2
Block Size 15		aperture 5	0,5
		aperture 7	0,5
Block Size 5	100x100	aperture 5	9
Block Size 9		aperture 7	6
		aperture 5	0,5
Block Size 15		aperture 7	0,5
		aperture 5	0
aperture 7		0	
Block Size 5	150x150	aperture 5	4
Block Size 9		aperture 7	2,5
		aperture 5	0
Block Size 15		aperture 7	0
		aperture 5	0
aperture 7		0	

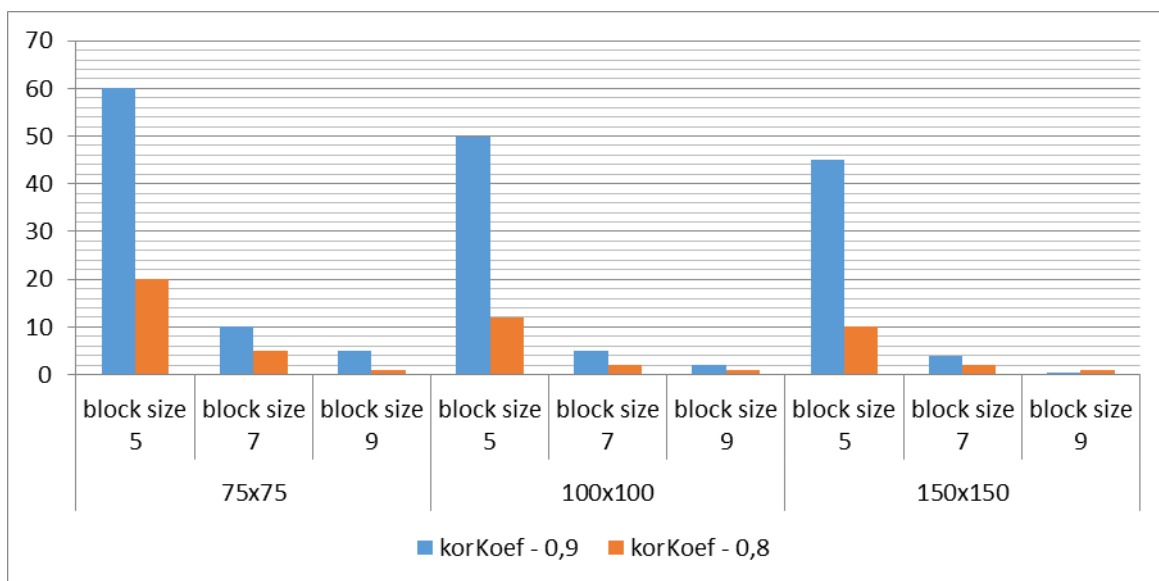


30 pav. Eigen decomposition algoritmo tyrimo rezultatai. Koreliacijos koeficientas 0,9

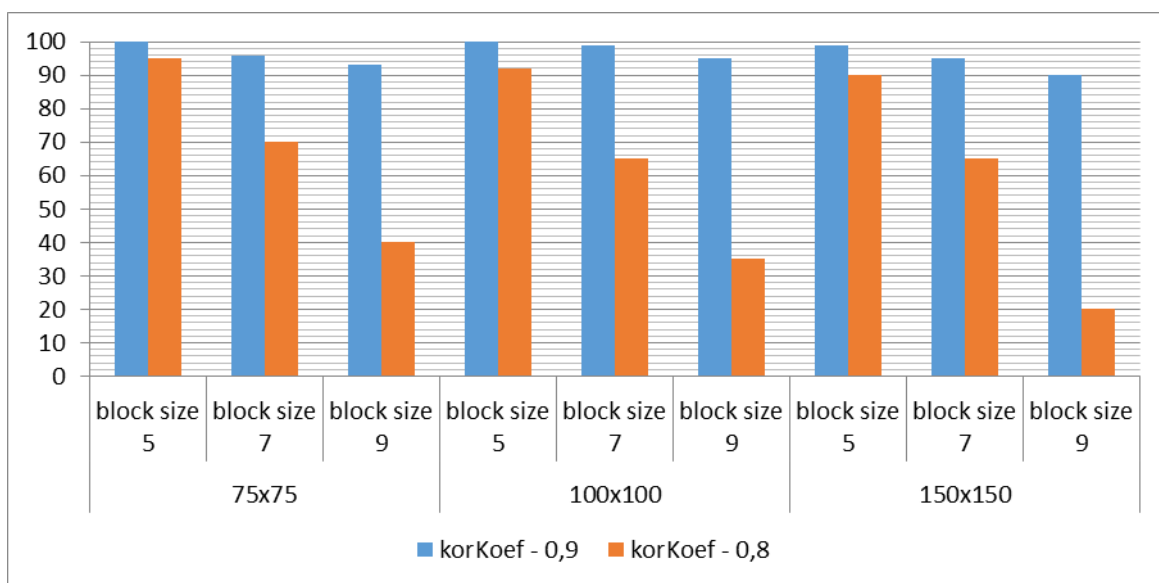
Corner detection algoritmas naudojamas per funkciją *differenceEdgesFilters*¹⁸ iš Aforge.Net bibliotekos.

Keičiamas parametras: *aperture* – branduolio dydis gali būti 1, 3, 5 ir 7. Tikrinant nepažeistą plokštę iki 60% segmentų identifikuojami kaip klaidingi, nes algoritmas jautrus esant, bet kokiems pokyčiams (31 pav.). O kai tiriama brokuota plokštė, pažeidimai aptinkami, tačiau taip pat klaidingai identifikuojamos ir nepažeistos plokštės vietos. (32 pav.).

Sumažinus koreliacijos koeficientą ir padidinus segmento dydį buvo gaunami geresni rezultatai.



31 pav. Corner detection algoritmo tyrimo rezultatai, su skirtingais parametrais. Be realių defektų.



32 pav. Corner detection algoritmo rezultatai, su skirtingais parametrais. Su realiais defektais.

¹⁸ Interneto šaltinis. *DifferenceEdgeDetector Class In: Aforge.NET Framework*. [žiūrėta 2015 m. balandžio 5 d.]. <http://www.aforgenet.com/framework/docs/html/d0eb5827-33e6-c8bb-8a62-d6dd3634b0c9.htm>.

Vaizdo apdorojimas *Harris corner* detector algoritmas realizuojamas per atviro kodo programą pateiktą C# bibliotekoje *HarrisCornersDetector* [29].

Parenkami parametrai:

- *block_size* – tiriamo bloko dydis
- *aperture* – diafragmos atvėrimo dydis. Gali būti 1, 3, 5 ir 7;
- *k* – keičiamas jautrumo koeficientas.

Atliekant bandymus buvo naudotas pastovus parametras *aperture* = 7, o kiti du – keičiami, nes šis algoritmas yra labai panašus į *Eigen decomposition*. Naudojant *Harris corner* detector algoritmą gerai atpažįstami ne tik realūs defektai, bet taip pat neteisingai identifikuojami tik iki 1% segmentų. Klaidingai identifikuotų segmentų nebelieka, kai parenkamos didesnės *block_size* (9, 11, 15) ir *k* (0.08, 0.12) reikšmės (6 lentelė).

Nustatyta, kad geriausi rezultatai gaunami, kai *Harris corner detector* algoritmo parametrai parenkami tokie: *block_size* ≥ 9; *aperture* size = 7; *k* ≥ 0.8.

6 lentelė. Klaidingai atpažinti defektai naudojant *Harris corner detector* algoritmą.

Nustatymai				Klaidingai identifikuotų neatitikčių skaičius, %	
				Be realių defektų	Su realiais defektais
korKoef=0, 9	75x75	k=0,4	block size 5	0,2	1
		k=0,8	block size 7	0,1	0,4
		k=1,2	block size 9	0,1	0,2
	100x100	k=0,4	block size 5	0	0
		k=0,8	block size 7	0	0
		k=1,2	block size 9	0	0
	150x150	k=0,4	block size 5	0	0
		k=0,8	block size 7	0	0
		k=1,2	block size 9	0	0
korKoef=0, 8	75x75	k=0,4	block size 5	0,1	0,2
		k=0,8	block size 7	0,1	0,2
		k=1,2	block size 9	0,1	0,1
	100x100	k=0,4	block size 5	0	0
		k=0,8	block size 7	0	0
		k=1,2	block size 9	0	0
	150x150	k=0,4	block size 5	0	0
		k=0,8	block size 7	0	0
		k=1,2	block size 9	0	0

4.4. Segmento dydžio įtaka

Atliekant tyrimą buvo nustatytas optimaliausias segmento dydis, dydžiai keičiami nuo 30x30 pikselių iki 150x150 pikselių, o koreliacijos koeficientas buvo parinktas 0.9. Fotokamera užfiksuotos medžio drožlių plokštės buvo išmatuotos, o pikseliai perskaičiuoti į milimetrus (7 lentelė). Toliau atliekant tyrimui buvo taikomi trys vaizdo apdorojimo algoritmai: *Eigen decomposition*, *Sobel operator*, *Harris corner detector*.

7 lentelė. Pikselių vertimas į milimetrus.

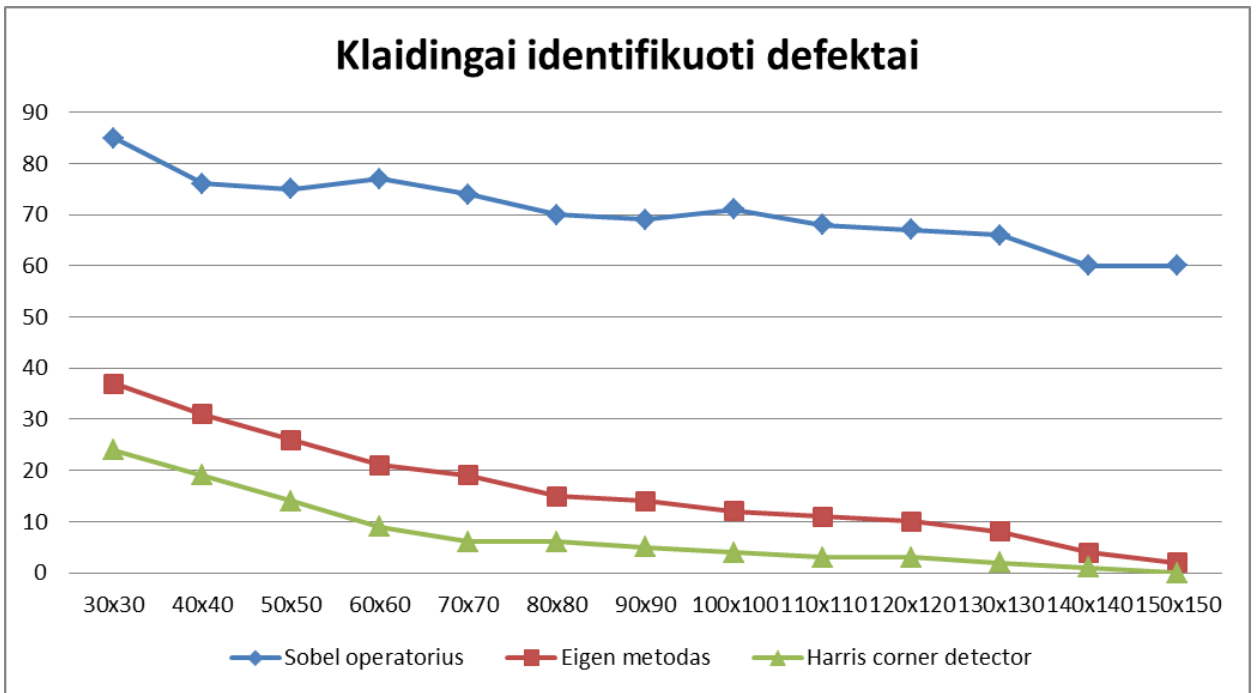
Pikseliai	30	40	50	60	70	80	90	100	110	120	130	140	150
Milimetrai	2,7	3,6	4,5	5,4	6,3	7,2	8,1	9	9,9	10,8	11,7	12,6	13,5

Neteisingai identifikuotų segmentų skaičius yra didelis, kai nėra naudojamas papildomas apšvietimas ir yra mažas segmento dydis (30x30 px – 50x50 px) (33 pav.). Naudojant *Sobel* operatorių, būtina sumažinti koreliacijos koeficientą, nes esant bet kokiam tiriamos plokštės ploto dydžiui neteisingai identifikuojamas segmentų skaičius yra labai didelis.

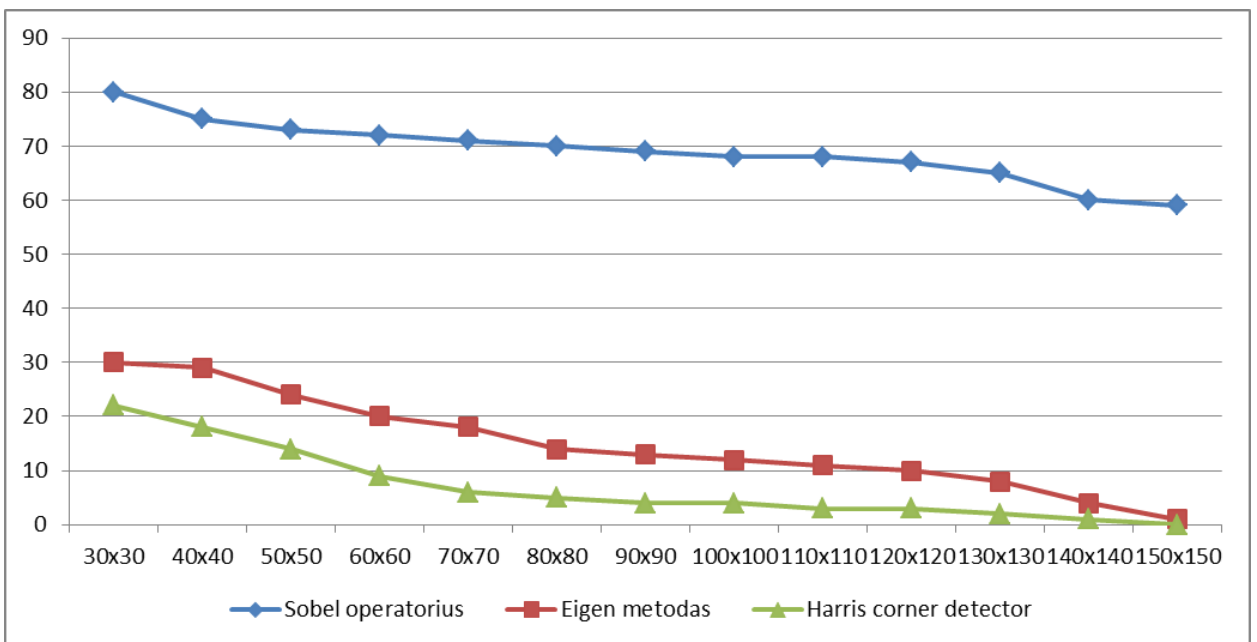
Harris corner detection ir *Eigen decomposition* metodų naudojimas davė labai panašius rezultatus. Klaidingai identifikuotų segmentų kiekis sumažėja, kada jo dydis yra nustatomas į 50x50px – 120x120px.

Fotografuojant didelį plokštės plotą prie tokių 130x130 px, neidentifikuoti plokštės pažeidimai, t.y. plokštė buvo atpažinta kaip neturinti jokių defektų. (33 pav.).

Panaudojus papildomą, sustiprintą apšvietimą, visi pažeidimai identifikuojami daug tiksliau nepaisant tiriamo segmento dydžio (34 pav.) Naudojant *Sobel operator* daromų klaidų skaičius sumažėja, bet jis vis tiek lieka palyginti didelis. Kai segmento dydis būna nuo 50x50px tada identifikuojami visi pažeidimai, o klaidos nėra daromos.



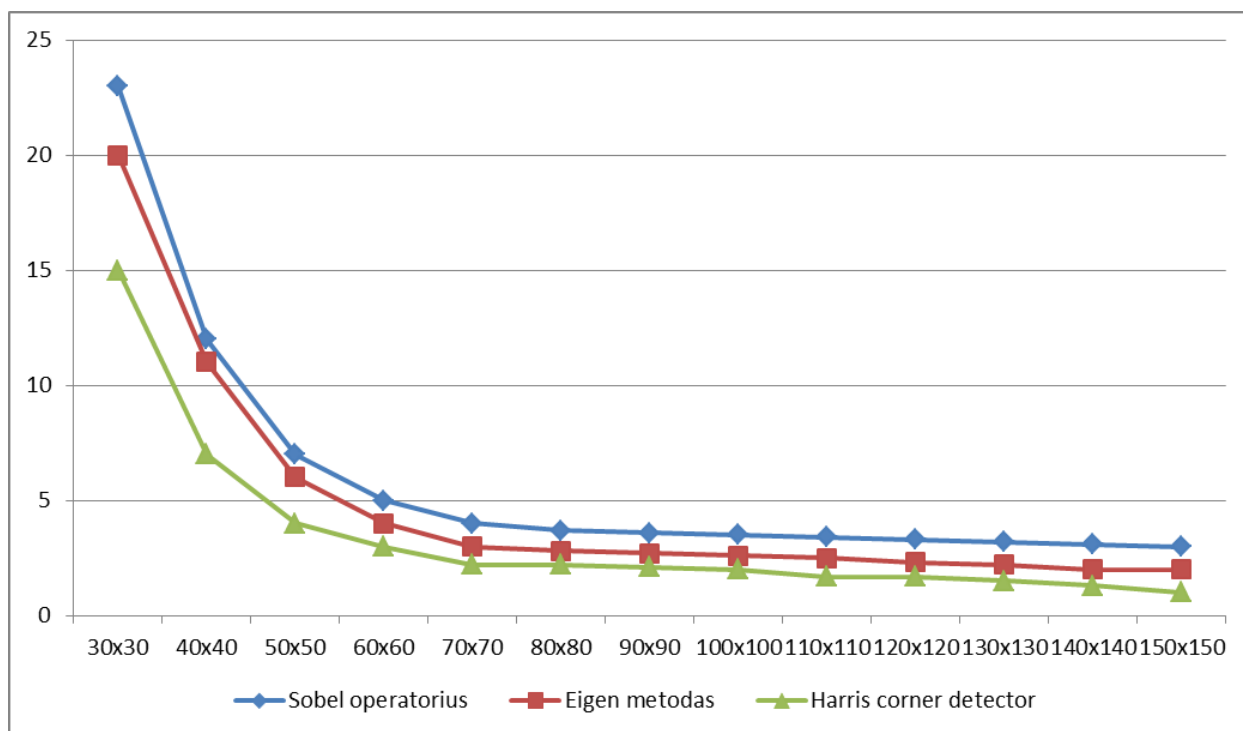
33 pav. Rezultatai be papildomo apšvietimo



34 pav. Rezultatai kai naudojamas papildomas stendo apšvietimas.

Atliekant tyrimus, plokštės vaizdas buvo optiškai priartinamas, taip siekiant išgauti kuo geresnę skiriamąją gebą. Rezultatai buvo daug geresni panaudojus optinį priartinimą (35 pav.), tačiau taip galima ištirti tik mažąją plokštės ploto dalį, taip prarandant viso patikrinimo greitį. Išsiaiškinta, kad esant segmento dydžiui 30x30px – 50x50px buvo daroma daugiausia klaidų (*Sobel*: 22,5% - 7%, *Harris corner detector*: 15,4% – 4,3%, *Eigen decomposition*: 20,4% - 5%).

Defektai buvo identifikuojami ir klaidos buvo daromos tik vaizdo kraštuose, esant didesniai segmento dydžiui.



35 pav. Rezultatai naudojant optinį priartinimą

Nustatyta, kad gauti geresnės kokybės vaizdus ir geresnius rezultatus būtina įrengti papildomą apšvietimą (800lm), tinkamiausi algoritmai ir jų parametrai: Sobel operator : Int x = 1, Int y = 0, aperture \geq 6; Eigen decomposition: block_size \geq 9; aperture \geq 6; Harris corner detector: block_size \geq 9, aperture = 7, k \geq 0.8.

Tobulinant sistemą, reikėtų sumontuoti ir suderinti kelias aukštos raiškos kameras, kad paspartinti kokybės tikrinimo procesą, ir nereikėtų rankiniu būdu stumti kamerą iki nustatytų taškų.

5. Išvados

- 1) Ištirti kompiuterinės vaizdo analizės sistemų veikimo principai.
- 2) Suprojektuotas ir pagamintas laminuotos medienos drožlių plokštės kokybės tikrinimo prototipas.
- 3) Išrinkti patikimiausi laminuotos medienos drožlių plokštės išgręžtų skylių kokybės tikrinimui vaizdo apdorojimo algoritmai, suderinti labiausiai tinkantys parametrai: Sobel operator: $\text{Int } x = 1, \text{Int } y = 0, \text{aperture} \geq 6$;
Eigen decomposition: $\text{block_size} \geq 9; \text{aperture} \geq 6$;
Harris corner detector: $\text{block_size} \geq 9, \text{aperture} = 7, k \geq 0.8$.
- 4) Norint gauti geresnės kokybės vaizdus, būtina įrengti papildomą apšvietimą: 800lm.
- 5) Tobulinant sistemą, reikėtų sumontuoti ir suderinti >2 vnt. aukštos raiškos kameras, kad paspartinti kokybės tikrinimo procesą ir padidinti sistemos patikimumą.

6. Literatūra

1. Peter Corke, "Robotics, vision and control", Springer tracks in advanced robotics, Vol 73, 2011.
2. Poole, Ian. *Automatic optical inspection, AOI systems*. In: *Radio-electronics: Resources and analysis for electronics engineers* [žiūrėta 2015 m. balandžio 7 d.]. Prieiga per internetą: http://www.radio-electronics.com/info/t_and_m/ate/aoi-automatic-automated-optical-inspection.php.
3. Loenas, Liudas. Įmonės pristatymas. In: *Industrinės vaizdo sistemos* [žiūrėta 2015 m. kovo 20 d.]. Prieiga per internetą: <http://visionsystems.lt/about/>.
4. Bartolini, Ilaria. *WARP: Accurate Retrieval of Shapes Using Phase of Fourier Descriptors and Time Warping Distance*. In: *IEEE Transactions on Pattern analysis and machine intelligence, vol. 27, no. 1, january 2005* [žiūrėta 2015 m. kovo 20d.]. Prieiga per internetą: <http://www-db.disi.unibo.it/courses/SI-M/papers/BCP05.pdf>.
5. D. G. Lowe, "Distinctive Image Features from Scale Invariant Keypoints", *International Journal of Computer Vision*, vol. 60, no. 2, 2004, psl. 91-110.
6. H.S.Sawhney, Y.Shan, R.Kumar and B.Matei. *Partial Object Matching with Shapeme Histograms*. In: *Ying Shan's Homepage* [žiūrėta 2015 m. balandžio 10 d.]. Prieiga per internetą: <http://www.shanfamily.com/PDFs/eccv.ying.2004.pdf>.
7. Rahtu, Esa. Salo, Mikko and Heikkila, Janne. *Affine Invariant Pattern Recognition Using Multi-Scale Autoconvolution* In: *University of Helsinki, Rolf Nevanlinna Institute* [žiūrėta 2015 m. kovo 15 d.]. Prieiga per internetą: http://www.rni.helsinki.fi/~msa/pub/TPAMI_MSA.pdf.
8. Dr.Bharadi, Vinayak Ashok. *Canny Edge Detection in C#* In: *Code Project, For those who code* [žiūrėta 2015 m. sausio 14 d.] Panaudotas filtravimo programos kodas. Prieiga per internetą: <http://www.codeproject.com/Articles/93642/Canny-Edge-Detection-in-C>.
9. Pawar, Kunal. *Implementation of Laplacian of Gaussian Edge Detection Algorithm using C#* In: *Code Project, For those who code* [žiūrėta 2015 m. sausio 14 d.] Panaudotas filtravimo programos kodas. Prieiga per internetą: <http://www.codeproject.com/Articles/37299/Implementation-of-Laplacian-of-Gaussian-Edge-Detection>.
10. Accord.Imaging Namespace In: *Accord.NET Framework* [žiūrėta 2015 m. balandžio 6d.]. Prieiga per internetą: http://accord-framework.net/docs/html/T_Accord_Imaging_HarrisCornersDetector.htm.

11. *AForge.Imaging.Filters Namespace* In: *AForge.NET Framework* [žiūrėta 2015 m. vasario 8d.]. Prieiga per internetą: <http://www.aforgenet.com/aforge/framework/docs/html/cdf93487-0659-e371-fed9-3b216efb6954.htm>.
12. Mokomoji medžiaga. *Kraštai ir Linijos*. In: *Vilniaus Universitetas, Matematikos ir Informatikos fakultetas* [žiūrėta 2015 m. vasario 15 d.]. Prieiga per internetą: http://uosis.mif.vu.lt/~neutrino/slides/10_2.pdf.
13. H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “SURF: Speeded Up Robust Features”, *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346--359, 2008.
Le-Jie Zhang, Zeng-Guang Hou, Min TAN, „Kalman Filter and Vision Localization Based Potential Field Method for Autonomous Mobile Robots „, [žiūrėta:2014-05-16]
14. Mokomoji medžiaga. *Kraštai ir Linijos. Canny - non maximus suppression* In: *Vilniaus Universitetas, Matematikos ir Informatikos fakultetas* [žiūrėta 2015 m. vasario 15 d.]. Prieiga per internetą: http://uosis.mif.vu.lt/~neutrino/slides/10_2.pdf.
15. Peter Corke, “Robotics, vision and control”, *Springer tracks in advanced robotics*, Vol 73, 2011.
16. Pawar, Kunal. *Implementation of Laplacian of Gaussian Edge Detection Algorithm using C#* In: *Code Project, For those who code* [žiūrėta 2015 m. sausio 14 d.] Panaudotas filtravimo programos kodas. Prieiga per internetą: <http://www.codeproject.com/Articles/37299/Implementation-of-Laplacian-of-Gaussian-Edge-Detect>
17. P.Kasparaitis. *Skaitmeninis vaizdų apdorojimas. Vaizdo apdorojimas. Sričių kraštai* 2008 10 28. In: *Vilniaus Universitetas, Matematikos ir Informatikos fakultetas* [žiūrėta 2015 m. vasario 15 d.]. Prieiga per internetą: <http://www.mif.vu.lt/~pijus/SVA/vdkr.pdf>.
18. E. Rosten and T. Drummond, Machine learning for high-speed corner detection, *Proceedings of the 9th European conference on Computer Vision*, vol. 1, pp. 430- 443, 2006.
19. Abdi, Herve. *The Eigen-Decomposition: Eigenvalues and Eigenvectors*. In: *The University of Texas at Dallas*. [žiūrėta 2015 m. kovo 20 d.]. Prieiga per internetą: <http://wwwpub.utdallas.edu/~herve/Abdi-EVD2007-pretty.pdf>.
20. Saitwal, Kishor and Maciejewski, Anthony. *Analysis of Eigendecomposition for Sets of Correlated Images at Different Resolutions* In: *College of Engineering* [žiūrėta 2015 m. sausio 20 d.]. Prieiga per internetą: <https://www.engr.colostate.edu/~aam/pdf/conferences/76.pdf>.

21. Sergio Andrés Gutiérrez Rojas. *Multiple face detection and recognition in real time*. In: *Code Project, For those who code* [žiūrėta 2015 m. balandžio 14 d.] Panaudotas filtravimo programos kodas. Prieiga per internetą: <http://www.codeproject.com/Articles/239849/Multiple-face-detection-and-recognition-in-real>.
22. Interneto šaltinis. *EigenValueDecomposition Class*. In: *Accord.NET Framework*. [žiūrėta 2015 m. balandžio 3 d.]. Prieiga per internetą: http://accord-framework.net/docs/html/T_Accord_Math_Decompositions_EigenvalueDecomposition.htm.
23. Straipsnis enciklopedijoje. *Corner Detection. The Moravec corner detection algorithm*. In: *Wikipedia. The free encyclopedia*. [žiūrėta 2015 m. balandžio 20d.] Prieiga per internetą: http://en.wikipedia.org/wiki/Corner_detection#LoG.2C_DoG.2C_and_DoH_feature_detection.
24. *AForge.Imaging.Filters Namespace* In: *AForge.NET Framework* [žiūrėta 2015 m. vasario 8d.] Prieiga per internetą: <http://www.aforgenet.com/aforge/framework/docs/html/cdf93487-0659-e371-fed9-3b216efb6954.htm>.
25. Harris, Chris and Stephens, Mike. *A Combined Corner and Edge Detector* Plessey Research Roke Manor, United Kingdom, 1988m. In: *BMVA The British Machine Vision Association and Society for Pattern Recognition*. [žiūrėta 2015 m. vasario 10 d.]. Prieiga per internetą: <http://www.bmva.org/bmvc/1988/avc-88-023.pdf>.
26. Straipsnis enciklopedijoje. *Corner Detection*. In: *Wikipedia. The free encyclopedia*. [žiūrėta 2015 m. sausio 20 d.]. Prieiga per internetą: http://en.wikipedia.org/wiki/Corner_detection#The_Harris_.26_Stephens_.2F_Plessey_.2F_Shi.E2.80.93Tomas_i_corner_detection_algorithm.
27. Derpanis, Konstantinos G. *The Haris Corner Detector*.(2004 10 27) In: *Department of Computer Science and Engineering. Indian Institute of Technology Kanpur*. [žiūrėta 2015 m. sausio 11 d.] Prieiga per internetą: http://www.cse.iitk.ac.in/users/vision/dipakmj/papers/2-harris_detector.pdf.
28. Straipsnis viešojoje enciklopedijoje. *Corner Detection*. In: *Wikipedia. The free Encyclopedia* [žiūrėta 2015 m. vasario 10 d.]. Prieiga per internetą: http://en.wikipedia.org/wiki/Corner_detection#The_Harris_.26_Stephens_.2F_Plessey_.2F_Shi.E2.80.93Tomas_i_corner_detection_algorithm.
29. Souza, Cesar. *Harris Corners Detector in C#* (2010.05.12) In: *Accord.Net Framework. Cesar Souza blog for sharing knowledge efficiently*. [žiūrėta 2015 m. sausio 5 d.] Panaudotas programos kodas Harris filtrui. Prieiga per internetą: <http://crsouza.com/2010/05/harris-corners-detector-in-c/>.

30. Mokomoji medžiaga. *Tiesinė koreliacija*. In: *Aleksandro Stulginskio Universitetas*. [žiūrėta 2014 m. lapkričio 8 d.]. Prieiga per internetą: http://www.asu.lt/nm/failai/Matematinis_modeliavimas/18.htm.
31. B. Steder, G. Grisetti, C. Stachniss, W. Burgard, Visual SLAM for Flying Vehicles, *IEEE Transactions on Robotics*, vol. 24, no. 5, 2008.
G. Reitmayr, E. Eade, T. W. Drummond. Semi-automatic Annotations in Unknown Environments, 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2007.
32. Souza, Cesar. *Harris Corners Detector in C#* (2010.05.12) In: *Accord.Net Framework*. *Cesar Souza blog for sharing knowledge efficiently*. [žiūrėta 2015 m. sausio 5 d.] Panaudotas programos kodas Harris filtrui. Prieiga per internetą: <http://crsouza.com/2010/05/harris-corners-detector-in-c/>.
33. G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces", 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nov. 2013.
34. B. Dai, D. Zhangb, H. Liua, S. Suna, and K. Lic, "Evaluation of Face Recognition Techniques", *Image Processing and Photonics for Agricultural Engineering*, vol. 7489, 2009.
D. Scaramuzza and F. Fraundorfer Visual Odometry: Part I - The First 30 Years and Fundamentals. *IEEE Robotics and Automation Magazine*, Volume 18, issue 4, 2011.
35. AFroge.NET biblioteka. *ExhaustiveTemplateMatching Class*. In: *Aforge.NET Framework*. [žiūrėta 2015 m. sausio 9 d.]. Prieiga per internetą: <http://www.aforgenet.com/framework/docs/html/17494328-ef0c-dc83-1bc3-907b7b75039f.htm>.