



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

# **Trimačio spausdinimo defektų aptikimo sistemos sukūrimas ir tyrimas**

Baigiamasis magistro projektas

---

**Dalius Kataržis**

Projekto autorius

**Lekt. dr. Kęstas Rimkus**

Vadovas

---

**Kaunas, 2021**



**Kauno technologijos universitetas**

Elektros ir elektronikos fakultetas

# **Trimačio spausdinimo defektų aptikimo sistemos sukūrimas ir tyrimas**

Baigiamasis magistro projektas

Valdymo technologijos (6211EX014)

---

**Dalius Kataržis**

Projekto autorius

**Lekt. dr. Kęstas Rimkus**

Vadovas

**Lekt. dr. Kęstutis Brazauskas**

Recenzentas

---

**Kaunas, 2021**



**Kauno technologijos universitetas**

Elektros ir Elektronikos Fakultetas

Dalius Kataržis

## **Trimačio spausdinimo defektų aptikimo sistemos sukūrimas ir tyrimas**

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Dalius Kataržis

*Patvirtinta elektroniniu būdu*

Kataržis, Dalius. Trimačio spausdinimo defektų aptikimo sistemos sukūrimas ir tyrimas. Magistro baigiamasis projektas. Vadovas lekt. Kęstas Rimkus; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): elektronikos inžinerija (inžinerijos mokslai)

Reikšminiai žodžiai: kokybės patikros sistema, vaizdinė analizė, trimatis spausdinimas.

Kaunas, 2021. 51 p.

### **Santrauka**

Baigiamasis magistro projektas yra skirtas trimačio spausdinimo metu atsirandančių defektų aptikimo sistemai sukurti ir ištirti. Ilgai trunkančio spausdinimo proceso metu atsiradę defektai gali nebūti pastebėti iš karto ir sugadinti ne tik gaminamą detalę, bet ir padaryti žalos spausdintuvui. Pirmoje darbo dalyje išskiriami defektai galintys pakenkti spausdintuvui, aptariami defektų aptikimo sistemų veikimo principai, naudojami algoritmai. Antroje dalyje aprašyti sistemos pritaikymo trimačiam spausdintuvui metodai ir atliekamas šablono modelio paruošimas. Tiriamoji dalis susideda iš sistemos sukūrimo - naudojamų metodų palyginimo, defektams aptikti naudojamų parametrų nustatymo, ir iš sukurtos sistemos tyrimo – tikrinama įtaka spausdinių išvaizdai, spausdinimo laikui, ir išbandomas veikimo efektyvumas su skirtingais defektų tipais.

Sukurta defektų aptikimo sistema remiasi spalvinio segmentavimo ir paveikslų atimties metodais nustatyti ne vietoje esančiai ar trūkstamai medžiagai, bei briaunų nustatymo ir šablono paieškos metodais įvertinti vidinių struktūrų defektams. Sistemos defektų aptikimo efektyvumas – 86,7%, kai naudojamos medžiagos atspalvis skiriasi nuo spausdinimo platformos. Norint spausdinimui panaudoti panašaus į spausdinimo platformą atspalvio medžiagą, rekomenduojama platformą padengti kitos spalvos danga, priešingu atveju sistemos efektyvumas krenta iki 65%. Visų bandymų metu, kokybiškų detalių vaizduose nebuvo klaidingai aptiktų defektų.

Kataržis, Dalius. Development and analysis of 3D print defects detection system. Master's Final Degree Project / supervisor lekt. Kęstas Rimkus; Faculty of Electrical and Electronics engineering, Kaunas University of Technology.

Study field and area (study field group): electronics engineering (engineering science).

Keywords: quality assurance, defect detection, 3D printing, visual inspection, machine vision.

Kaunas, 2021. 51 p.

### **Summary**

The final master's thesis aims at developing and analysing a system for defect detection in material extrusion-based 3D printing process. Long manufacturing times so common in 3D printing, may lead to unnoticed defects that pose threat not only to the part being manufactured, but to the printer itself. In the first part of the thesis, we pick out defects that pose the greatest danger to printing equipment and discuss various methods for defect detection. The second part of the thesis focuses on adapting the system to specific 3D printer in question and explaining how template images were made. The third and final part consists of defect detection system analysis and parameter selection.

The developed defect detection system is based on color segmentation and image subtraction methods for detecting zones with missing or too much material, and on edge detection and template matching for detecting layer shift defects. The efficiency of said system is 86,7% when used on parts made from different color material than the print bed. For using parts with similar color to the print bed, it is recommended to coat the print bed with a different color otherwise, system efficiency plummets to 65%. In any case, all experiments done on quality parts with no defects shows no false-positive detections.

## Turinys

Lentelių sąrašas.....	8
Paveikslų sąrašas.....	9
Santrumpų ir terminų sąrašas.....	10
Įvadas .....	11
<b>1. Teorinė dalis .....</b>	<b>12</b>
1.1. Trimačio spausdinimo būdai .....	12
1.2. Defektai pasitaikantys spausdinimo metu .....	13
1.3. Vizualinės patikros sistemos .....	14
1.3.1. Vaizdo kamera.....	14
1.3.2. Darbo aplinka ir apšvietimas.....	15
1.4. Vaizdo apdorojimo metodai .....	15
1.4.1. Segmentavimas pagal ryškumą.....	15
1.4.2. Segmentavimas pagal spalvą.....	16
1.4.3. Briaunų nustatymas.....	16
1.4.4. Vaizdo briaunų nustatymo būdų palyginimas.....	19
1.5. Vaizdų palyginimo metodai .....	19
1.5.1. Paveikslų atimties metodas .....	19
1.5.2. Koreliacija ir kiti panašumo nustatymo metodai.....	20
1.6. Programinė įranga.....	20
1.6.1. Bibliotekų palyginimas .....	21
<b>2. Metodologinė dalis.....</b>	<b>22</b>
2.1. Naudojama eksperimento įranga .....	22
2.1.1. Spausdintuvas ir jo modifikacijos .....	22
2.1.2. Kamera.....	23
2.1.3. Spausdintuvo valdiklio paruošimas .....	23
2.1.4. Fotografijų surinkimas .....	25
2.1.5. Šablono modelio paruošimas .....	26
2.2. Objekto išskyrimo iš fono metodai.....	27
2.2.1. Objekto išskyrimas segmentacijos metodais.....	27
2.2.2. Briaunų nustatymo metodo panaudojimas.....	29
2.2.3. Vidinių struktūrų išskyrimas .....	30
<b>3. Tiriamoji dalis .....</b>	<b>32</b>
3.1. Defektų aptikimo sistemos parametrų nustatymas .....	32
3.1.1. Defektų aptikimas naudojant segmentavimą .....	32
3.1.2. Defektų aptikimas naudojant šablono paieškos metodus .....	35
3.2. Sudarytos defektų aptikimo sistemos veikimo principas .....	39
3.3. Defektų aptikimo sistemos įtaka spausdinamos detalės išvaizdai .....	40
3.4. Defektų aptikimo sistemos veikimo greitis .....	41
3.4.1. Šablono vaizdų išskyrimo greitis .....	42
3.4.2. Darbo stabdymo ir pratęsimo trukmė, galimos optimizacijos .....	43
3.5. Defektų aptikimo sistemos efektyvumo tyrimas .....	43
3.5.1. Defektų aptikimo žalios spalvos detalėse eksperimentas .....	44

3.5.2. Defektų aptikimo pilkos spalvos detalėse eksperimentas.....	45
3.5.3. Defektų aptikimo kai spausdinamos dvi detalės eksperimentas .....	47
<b>Rezultatai ir Išvados .....</b>	<b>49</b>
<b>Literatūros sąrašas.....</b>	<b>50</b>
<b>Priedai .....</b>	<b>52</b>
1 priedas. Nuotraukų ir šablonų rinkinys naudotas parametrų nustatyti .....	52
2 priedas. Maskavimo įtaka šablono paieškos algoritmams.....	53
3 priedas. Šablono paieškos algoritmų skaičiavimo laikas .....	54
4 priedas. Bandymams naudotas defektų aptikimo sistemos programos kodas .....	55
5 priedas. Defektų aptikimo sistemos skaičiavimo laikas.....	57
6 priedas. Šablono vaizdų apdirbimo skaičiavimo laikas .....	58
7 priedas. Defektų aptikimo žalios spalvos detalėse eksperimento nuotraukos.....	59
8 priedas. Defektų aptikimo pilkos spalvos detalėse eksperimento nuotraukos .....	61
9 priedas. Defektų aptikimo kai spausdinamos dvi detalės eksperimento nuotraukos .....	63

## Lentelių sąrašas

1 lentelė. Dažnai pasitaikantys spausdinimo defektai ir jų priežastys .....	13
2 lentelė. Vaizdo briaunų nustatymo būdų palyginimas .....	19
3 lentelė. Vaizdo atpažinimo bibliotekų palyginimas .....	21
4 lentelė. Tyrimui aktualios kameros specifikacijos.....	23
5 lentelė. Tyrimui aktualios spausdintuvo valdiklio specifikacijos [25] .....	24
6 lentelė. Defekto įvertis gautas vaizdų atimties metodu, prieš ir po mažų plotų zonų filtravimo ...	34
7 lentelė. Panašumo skaičiavimo ir briaunų nustatymo metodų palyginimas.....	37
8 lentelė. Bandomų kompiuterių konfigūracijos .....	41
9 lentelė. Defektų aptikimo žalios spalvos detalėse rezultatai .....	44
10 lentelė. Defektų aptikimo pilkos spalvos detalėse rezultatai.....	46
11 lentelė. Defektų aptikimo aptikimo kai spausdinamos dvi detalės rezultatai.....	47



## Paveikslų sąrašas

1.1 pav. Segmentavimo pagal slenkstinę vertę pavyzdys. [10].....	16
1.2 pav. RGB ir HSV spalvinių palečių palyginimas, spalvų pasiskirstymas [11] .....	16
1.3 pav. Sobel operatoriumi atliktas briaunų radimas [13] .....	17
1.4 pav. Canny operatoriumi atliktas briaunų radimas [15].....	18
1.5 pav. Paveikslų atimties metodo eiga, spausdintinių plokščių patikrai [19].....	19
2.1 pav. Spausdintuvas su gaubtu ir LED apšvietimu .....	22
2.2 pav. Naudojama kamera. ....	23
2.3 pav. Kameros vaizdo centro ir spausdinimo centro kalibravimas .....	23
2.4 pav. settings.g failo kodo iškarpa.....	24
2.5 pav. Patikra.g failo kodas .....	24
2.6 pav. Darbo pratęsimo, galvutės valymo kodas .....	25
2.7 pav. Kodas nuotraukoms padaryti, gavus signalą. ....	25
2.8 pav. Patikros kodo įterpimo šliaužiklis, ir modeliuoto šablono pavyzdys. ....	26
2.9 pav. Spausdinimo platformos įtaka segmentacijai pagal ryškumo vertę.....	27
2.10 pav. Segmentavimo pagal ryškumo vertę pritaikymas modeliuoto šablono vaizdui.....	27
2.11 pav. Segmentavimo pagal spalvą rezultatas. ....	28
2.12 pav. Segmentavimo pagal spalvą ir filtravimo rezultatas.....	28
2.13 pav. Spalvinės segmentacijos pritaikymas šablonui .....	29
2.14 pav. Šablono tarpelių užpildymas .....	29
2.15 pav. Briaunos gautos <i>Canny</i> algoritmu. ....	30
2.16 pav. Sobel ir Laplaso operatoriais paremtas kraštinių radimas .....	30
2.17 pav. Briaunų aptikimo ir segmentavimo apjungimas.....	31
3.1 pav. Sluoksnių pasislinkimo defekto išskyrimas po segmentacijos pagal spalvą.....	33
3.2 pav. Spausdinimo ore, atsiklijavimo ir prilaikančiųjų konstrukcijų defektų išskyrimas po segmentacijos pagal spalvą. ....	33
3.3 pav. Defekto įverčiai po segmentavimo, sugrupuoti pagal defekto tipą .....	35
3.4 pav. Maskavimo įtaka paieškos koeficientui .....	36
3.5 pav. Maskavimo įtaka paieškos greičiui.....	36
3.6 pav. Sluoksnių pasislinkimo aptikimas koreliacijos metodu.....	38
3.7 pav. Paieškos laikas skirtingais panašumo skaičiavimo metodais.....	38
3.8 pav. Darbo sustabdymo įtaka detalių iš PLA plastiko išvaizdai.....	40
3.9 pav. Darbo sustabdymo įtaka detalių iš PETG plastiko išvaizdai .....	41
3.10 pav. Defektų aptikimo sistemos greitaveika.....	42
3.11 pav. Šablono vaizdų išskyrimo greitaveika .....	43
3.12 pav. Klaidingai aptikti defektai žalios spalvos detalėse .....	45
3.13 pav. Klaidingai aptikti defektai pilkos spalvos detalėse.....	47

## Santrumpų ir terminų sąrašas

### Santrumpos:

MEX – Material Extrusion;

PLA – Polyactic acid;

PETG - Polyethylene terephthalate glycol;

Px – pixel, pikselis.

### Terminai:

**CoreXY** – roboto tipas, kai manipulatorius judinamas XY plokštumoje panaudojant dvi stacionarias pavaras.

**Dilate** – morfologinė vaizdo operacija, dvispalviuose vaizduose padidinti vienos iš spalvų plotą.

**Erode** - morfologinė vaizdo operacija, dvispalviuose vaizduose sumažinti vienos iš spalvų plotą.

**Opening** - morfologinė vaizdo operacija, pritaikanti Erode, o po to Dilate operacijas. Naudojama triukšmams šalinti.

**Closing** - morfologinė vaizdo operacija, pritaikanti Dilate, o po to Erode operacijas. Naudojama tarpeliams šalinti.

**Bilateral** – vaizdo triukšmų filtravimo būdas.

## Įvadas

Populiarėjant trimačio spausdinimo technologijai ir tobulėjant supratimui apie šios technologijos pritaikymą, vis dažniau imamasi didelių projektų, reikalaujama tikslių matmenų ir sudėtingų formų gaminių. Neretai tokių gaminių spausdinimas trunka parą ar ilgiau ir sunaudoja didelius kiekius medžiagos.

Tačiau populiarėjant trimačio spausdinimo technologijai vis dažniau tenka išgirsti atsiliepimus apie nepavykusius spausdinius ir tarsi iš niekur atsiradusius defektus. Neretai tai sukelia operatoriaus žinių trūkumas, neteisingai parinkti parametrai ar nekokybiškas naudojamas polimeras. Tačiau nereikia atmesti ir mechaninių problemų, galinčių atsirasti netikėtai ir sutrukdyti net ir labiausiai pasiruošusiam operatoriui. Čia ir iškyla šiame darbe sprendžiama problema – jei spausdinimas atliekamas be priežiūros, o defektai gali atsirasti atsitiktinai, kas nutinka kai defektas nepastebimas iš karto?

Priklausomai nuo atsiradusio defekto pobūdžio, toliau tęsiant darbą gali būti tik veltui naudojama medžiaga, bet gali būti ir pažeidžiama spausdinimo galvutė, joje esantys jutikliai, kaitinimo elementai. Laiku sustabdžius darbą, šių padarinių galima išvengti, o tą atlikti nereikalaujant operatoriaus dėmesio galima panaudojus automatines kokybės stebėjimo sistemas.

Pramonėje naudojamos kokybės stebėjimo sistemos dažnai remiasi palyginimu su jau pagamintomis kokybiškomis detalėmis, etaloniniais dirbiniais. Trimatis spausdinimas dažniau yra naudojamas prototipams ar kitiems vienetiniams gaminiams ir neturima realaus atitiktens kokybės palyginimui. Todėl šiame darbe bus naudojamos kompiuteriu generuotais šablonais kokybės patikrai atlikti.

Projekto tikslas – sukurti ir išbandyti sistemą galinčią identifikuoti kritinius MEX trimačio spausdinimo metu pasitaikančius defektus.

Projekto uždaviniai:

1. susipažinti su pagrindiniais vaizdo atpažinimo techninei įrangai keliamais reikalavimais ir jais remiantis sukonstruoti defektų aptikimo sistemos prototipą;
2. atlikus literatūros apžvalgą pasirinkti du kuriamoje defektų aptikimo sistemoje naudojamus vaizdų palyginimo metodus;
3. sukurti algoritmą defektų aptikimui ir remiantis bandymų rezultatais nustatyti jo parametrus;
4. įvertinti ar atliekant patikrą daroma įtaka spausdinamų detalių išvaizdai;
5. atlikti sistemos greitaveikos tyrimą;
6. atlikti sistemos patikimumo tyrimus su skirtingu skaičiumi ir skirtingų spalvų detalėmis.

## 1. Teorinė dalis

### 1.1. Trimačio spausdinimo būdai

Darbe bus nagrinėjami MEX spausdinimo proceso metu atsirandantys defektai. Siekiant geriau suprasti iš kur atsiranda specifiniai, tik šiai technologijai pritaikomi sprendimai, reikia išsiaiškinti pagrindinius veikimo principus ir skirtumus tarp skirtingų trimačio spausdinimo technologijų. Standartas ISO/ASTM 52900 [1] išskiria šias pagrindines trimačio spausdinimo kategorijas:

- BJT – (angl. *binder jetting*), procesas kurio metu skysta rišamoji medžiaga naudojama miltelių pavidalo medžiagoms sutvirtinti.
- DED – (angl. *directed energy deposition*), procesas kurio metu lazerio ar plazmos šaltinio energija naudojama išlydyti tiekiamai medžiagai.
- MEX – (angl. *material extrusion*), procesas kurio metu spausdinimui naudojama medžiaga tiekama per tūtą ar purkštuką.
- MJT – (angl. *material jetting*), procesas kurio metu medžiaga spausdinimui tiekama lašelių pavidalu.
- PBF – (angl. *powder bed fusion*), procesas kurio metu šiluminė energija naudojama miltelių pavidalo medžiagoms sutvirtinti.
- SHL – (angl. *sheet lamination*), procesas kurio metu sujungiami tam tikros formos medžiagos lakštai.
- VPP – (angl. *vat photopolymerization*), procesas kurio metu šviesa naudojama šviesoje kietėjančio polimero formavimui.

Prieš standarto įsteigimą, MEX spausdinimo procesas skirtingų gamintojų vadintas ir geriau gali būti žinomas kaip - FDM (angl. *Fused Deposition Modelling*), FFF (angl. *Fused Filament Fabrication*), PJP (angl. *Plastic Jet Printing*) ar MEM (angl. *Melted and Extruded Modelling*) procesas. Šiuo procesu gali būti spausdinamos įvairios medžiagos – betonas, keramika, metalai ar šokoladas, tačiau plačiausiai naudojamas termoplastikas [2].

Termoplastikas spausdinimui dažniausiai tiekiamas ritėse, kalibruoto diametro gijos pavidalu. Spausdinimo proceso metu, pusiau skysto pavidalo medžiaga tiekama per kompiuteriu valdomą purkštuką ant spausdinimo platformos. Detalės spausdinimas atliekamas sluoksniais, naujo sluoksnio medžiagą tiekiant tiesiai ant sukietėjusios prieš tai buvusio sluoksnio medžiagos. Biudžetinės klasės įrenginiuose pasikliaujama pastoviu gijos skersmeniu, nuspėjamu medžiagos elgesiu ir idealiai kalibruotais parametrais detalei atspausdinti. Neturint grįžtamojo ryšio su vykstančiu procesu, nesunku nuspėti, kad netikėtai pasikeitus nors vienam parametru atsiranda spausdinamos detalės defektai.

## 1.2. Defektai pasitaikantys spausdinimo metu

Dažnai pasitaikantys defektai ir subendrintos jų atsiradimo priežastys [3] nurodytos pirmoje lentelėje.

1 lentelė. Dažnai pasitaikantys spausdinimo defektai ir jų priežastys

Defektai	Atsiradimo priežastys
Smulkios matmenų paklaidos	Plastiko variacija, netinkamas pasiruošimas darbui
Didelės matmenų paklaidos, kitos deformacijos	Mechaninės problemos
Darbo plokštumoje pasislinkę sluoksniai	Mechaninės problemos
Užsikimšusi spausdinimo galvutė	Plastiko variacija, Mechaninės problemos
Spausdinio atsiklijavimas nuo spausdinimo stalo	Mechaninės problemos, netinkamas pasiruošimas darbui
Neatkuriamos smulkios detalės	Netinkamas pasiruošimas darbui
Prilipę sudegusio plastiko gabaliukai	Plastiko variacija
Atraminė konstrukcija deformuota	Plastiko variacija, netinkamas pasiruošimas darbui
Įtrūkimai tarp sluoksnių	Netinkamas pasiruošimas darbui
Tarpeliai tarp perimetro ir užpildo takelių	Plastiko variacija, netinkamas pasiruošimas darbui
Spausdinimas ore	Netinkamas pasiruošimas darbui, <b>kiti atsiradę defektai</b>

Defektai kuriuos sukelia mechaninės spausdintuvo problemos kaip atsilaisvinę sinchroniniai dirželiai ar dantračiai, nesukalibruota spausdinimo platforma ir laidų pažeidimai, gali atsirasti bet kuriuo momentu ir paprastai kelia didžiausią riziką. Atsiklijavę, deformuotos detalės ir ne vietoje spausdinamas plastikas gali prilipti prie spausdinimo galvutės ir ją apgaubti, dažniausiai nepataisomai apklijuojant jutiklius ir kaitinimo elementus, o pažeistas temperatūros jutiklis gali įtakoti gaisro atsiradimą. Todėl šių problemų keliamus defektus svarbu atpažinti ir sustabdyti darbą.

Defektai kuriuos sukelia plastiko kokybė (nevienodas diametras, svetimkūniai) paprastai apgadina spausdinamą detalę, tačiau nekelia didesnio pavojaus. Prilipę plastiko gabaliukai ar detalės paviršiuje atsiradę tarpeliai dėl staiga sumažėjusio plastiko diametro gali būti pašalinti papildomo apdirbimo po spausdinimo metu ir todėl šio darbo metu atskirai ši defektų kategorija nebus nagrinėjama.

Netinkamas pasiruošimas darbui (netinkama temperatūra, spausdinimo greitis ar kiti parametrai) gali sukelti tokius pat defektus kaip ir plastiko ar mechaninės problemos, tačiau atidžiau atliekant pasiruošimo darbus galima šiuos neatitikimus pastebėti ir ištaisyti. Sėkmingai atspausdinus vieną detalę, tuo pačiu kodu atsispausdins ir kita, jei neatsiras nuo kodo nepriklausančių pakitimų. Todėl šio darbo metu atskirai ši defektų kategorija taip pat nebus nagrinėjama.

Apibendrinant defektus ir įvertinant jų padarinius nuspręsta, kad patikros metu bus ieškomi šie defektai:

- Spausdinimo ore, nes dažniausiai pasireiškia kaip kito defekto padarinys ir kelia didžiausią riziką spausdinimo galvutės pažeidimui.
- Atsiklijavimo ir prilaiakančiųjų konstrukcijų, nes tiesiogiai sukelia spausdinimo ore defektą, kuris kelia riziką spausdinimo galvutės pažeidimui.
- Sluoksnių pasislinkimo ir detalės deformacijų, nes yra pirmieji simptomai signalizuojantys apie mechanines spausdintuvo problemas.

### 1.3. Vizualinės patikros sistemos

Tam, kad kokybės kontrolės sistema galėtų atpažinti defektus, jai reikalingi darbo aplinkos jutikliai. Nagrinėjamu atveju tam puikiai tinka vaizdo atpažinimo sistema. Egzistuoja komercinės vaizdo atpažinimo sistemos [4], pritaikytos specifiniams tikslams, produktų grupėms ar pramonės šakoms. Dažniausiai tokias sistemas sudaro kamera, darbo vietos apšvietimas ir kompiuteris su specialia programine įranga. Tačiau neradus reikiamo sprendimo, gali tekti vaizdo atpažinimo sistemą susikurti patiems, pagal savo konkrečius reikalavimus. Tokiu atveju galime optimizuoti darbo aplinką, jos apšvietimą, kameros ir objektyvo pasirinkimą, naudojamus vaizdo atpažinimo algoritmus bei jų parametrus.

#### 1.3.1. Vaizdo kamera

Pirmas žingsnis atliekamas prieš bet kokią vaizdo atpažinimo operaciją, tai to vaizdo gavimas. Geri atpažinimo rezultatai gaunami tik tada, kai gera ir gauto vaizdo kokybė. Todėl yra svarbu pasirinkti vaizdo kamerą tenkinančią specifinės užduoties reikalavimus – šviesos spektro, jutiklio raiškos bei tipo, kadru dažnio ir duomenų perdavimo greitaveikos.

Nespalvotus vaizdus fiksuojančios kameros paprastai yra geresnis pasirinkimas jei užduočiai nereikia spalvų. Taip yra dėl to, kad šios kameros neturi spalvinių filtrų ir yra jautresnės. Tačiau jei stebimiems objektams reikia ultravioletinio ar infraraudono apšvietimo, šie šviesos ruožai turi patekti į kameros jutiklio fiksuojamą spektrą. Žinant šviesos spektro ribas, lengviau pasirinkti jutiklio tipą.

Išskiriamos dvi pagrindinės jutiklių technologijos – CCD ir CMOS. CCD (angl. *Charge-coupled device*) jutikliai yra pigesni, tačiau reaguoja tik į matomos šviesos spektrą, jautresni akinimui, bei karščio keliamiems trikdžiams. Šiuolaikiniai CMOS (angl. *Complementary Metal Oxide Semiconductor*) jutikliai yra efektyvesni energijos suvartojimo atžvilgiu ir turi didesnę kadru fiksavimo dažnį [5].

Priklausomai nuo fiksuojamo objekto matmenų reikia pasirinkti tarp zonos ir linijos nuskaitymo kamerų. Įprastinės vaizdo kameros vaizdą nuskaito dvimatėje erdvėje iš karto, tačiau linijos nuskaitymo kameros fiksuoja vieno pikselio pločio liniją ir iš daugybės tokių nuotraukų galima išgauti bendrą objekto vaizdą. Linijos kameros puikiai tinka didelio ilgio (pvz. žurnalų puslapiams ruluose) bei greitai judantiems objektams fiksuoti.

Nuo vaizdo raiškos priklauso gauto vaizdo tikslumas ir detalumas, o nuo kadru dažnio – kaip greitai judančius objektus bus galima stebėti. Paprastai šie parametrai atvirkščiai proporcingi – mažėjant vaizdo raiškai, galima pasiekti didesnę kadru dažnį. Šiuolaikiniai jutikliai siūlo raiškas iki 48 Megapikselių ir nesunkiai pasiekia 60 kadru per sekundę dažnį [5].

Žinant naudojamą raišką bei kadru dažnį, galima pasirinkti duomenų perdavimo sąsają, nuo kurios priklausys maksimalus duomenų perdavimo greitis bei atstumas. Populiariausi standartai – *USB*, *CameraLink*, *GigE* ir *CoaXPress* [6]. *USB3 Vision* standartas leidžia greičius iki 350 MB/s – užtekinai įprastoms vaizdo atpažinimo kameroms. Naudojamo kabelio ilgis ribojamas iki 3 metrų, tačiau šie kabeliai pigūs ir nereikalauja papildomos aparatūrinės įrangos signalui priimti. Likę standartai yra nuolat atnaujinami, didinama jų sparta. Pavyzdžiui dabartinis *10GigE Vision* standartas leidžia komunikacijų atstumus iki 100 m., bei greičius iki 1,25 GB/s. *CameraLink* greitis – 850 MB/s,

o maksimalus atstumas – 10 metrų, tačiau perduodamas signalas labiau atsparus trikdžiams. Šiuo metu kuriama nauja *CoaXPress* karta *CXP-12* žada duomenų perdavimo spartą iki 5 GB/s. Todėl jei nekeliami ypatingi reikalavimai duomenų perdavimo spartai ar atstumui, galima rinktis USB3 standartu komunikuojančią kamerą dėl pigios instaliacijos.

### **1.3.2. Darbo aplinka ir apšvietimas**

Gera vaizdo kokybė priklauso ne tik nuo kameros, bet ir nuo apšvietimo. Esant nepakankamam apšvietimui, atsiranda šešėliai, sunkiau atskirti objekto kraštus, o kai apšvietimas per stiprus gali atsirasti nepageidaujami atspindžiai. Tačiau šviestuvo parinkimas neapsiriboja vien tinkamu šviesos srauto parametru radimu – skirtingi objektai gali būti geriau išskiriami priklausomai nuo lempučių tipų, išdėstymo, šviesos sklaidymo ar filtravimo [7].

Šiuo metu pramonėje dažniausiai naudojami LED šviestuvai [8]. Tobulėjant technologijai, LED apšvietimas tapo pranašesnis už konkurentus dėl savo paprastumo, stabilumo ir gyvavimo laiko, tačiau fluorescenciniai šviestuvai vis dar pranašesni dėl mažesnės kainos. Jei reikalingas platesnis spalvų spektras, galima naudoti ksenonines ar gyvsidabrinės lemputes.

Pašaliniai šviesos šaltiniai, tokie kaip patalpos apšvietimas ir saulė, ar net šalia dirbančių mašinų indukcinės lemputės, gali turėti didelę įtaką vaizdo atpažinimo kokybei bei pastovumui. Tai galima išspręsti panaudojus ryškią pulsuojančią šviesą kuri tiesiog sumenkina aplinkos įtaką, tačiau šis sprendimas tinkamas tik su kai kuriais lempučių tipais, taip pat yra sunkiau techniškai įgyvendinamas. Jei tiksli spalvinė informacija nėra reikalinga, galima panaudoti optinius filtrus, o kitais atvejais belieka viską sumontuoti į šviesos nepraleidžiantį aptvarą.

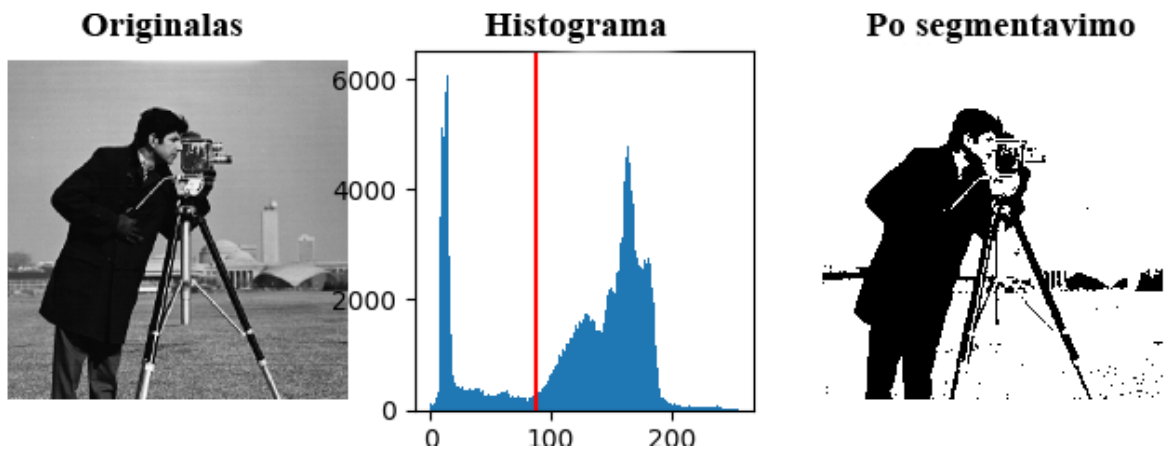
Nuo objekto formos ir medžiagos priklauso šviesos reakcija, todėl yra daugybė šviesos sklaidymo, koncentravimo ar pozicionavimo būdų pritaikytų skirtingiems produktams [9]. Pavyzdžiui, išsklaidyta šviesa geriau tinka tekstui ant metalinių paviršių stebėti, o apšvietus objektą iš nugaros, lengviau atpažinti kontūrus. Dėl didelės apšvietimo daromos įtakos tolesniam vaizdų atpažinimui, visada rekomenduojama išbandyti kelis apšvietimo variantus ir pasirinkti geriausią prieš pereinant prie tolesnių sistemos dalių.

### **1.4. Vaizdo apdorojimo metodai**

Kitas etapas objekto atpažinimo uždavinyje – gauto vaizdo apdirbimas, modifikavimas, siekiant išgauti norimą informaciją ar sumažinti jos perteklių. Pagrindiniai vaizdo apdorojimo būdai:

#### **1.4.1. Segmentavimas pagal ryškumą.**

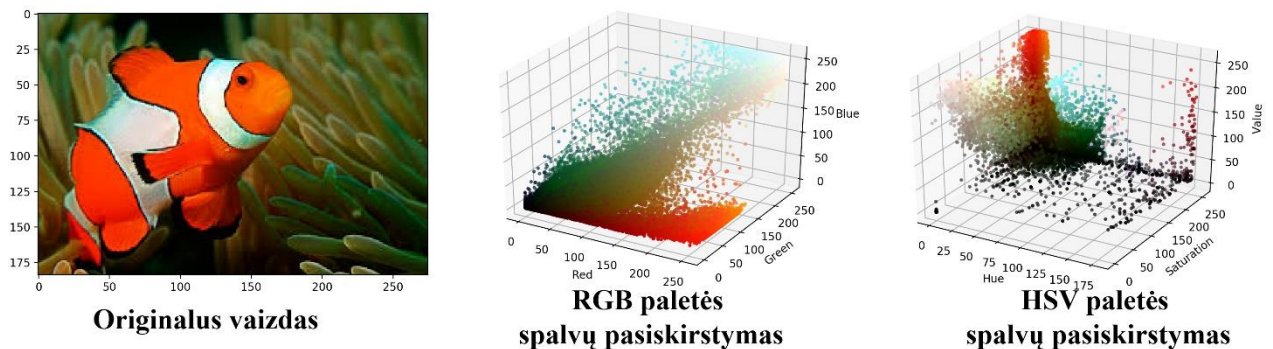
Segmentavimas, tai vaizdo dalinimas į objektus atstovaujančius regionus, atskyrimas nuo fono. Dažnai tam naudojamas slenkstinės vertės radimas. Tai procesas kai vaizdą sudarantys pikseliai pakeičiami į juodus arba baltus, priklausomai nuo to ar jų ryškumas viršija tam tikrą vertę. Tokiu būdu atskiriami tamsūs objektai šviesiame fone, ar atvirkščiai. Realioje aplinkoje šis metodas veikia prastai, dėl ryškumą įtakojančių veiksnių kaip apšvietimas ir atspindžiai, tačiau pramoninėje aplinkoje kur galima kontroliuoti ne tik apšvietimą, bet ir foną, slenkstinės vertės radimas vis dar taikytinas.



1.1 pav. Segmentavimo pagal slenkstinę vertę pavyzdys. [10]

### 1.4.2. Segmentavimas pagal spalvą.

Tai vienspalvio regiono išskyrimas panašiai kaip slenkstinės vertės radimo atveju, paprastai atliekamas HSV spalvinėje paletėje. HSV – tai atspalvis (*Hue*), sodrumas (*Saturation*) ir apšvietumo vertė (*Value*). Šis spalvinis modelis patogus tuo, kad norint išskirti spalvą užtenka pasirinkti atspalvį, nereikia galvoti kaip sumaišyti RGB dedamąsias. Tuo pačiu tai reiškia, kad visi panašūs atspalviai yra netoli vienas kito, o ne išsibarstę po visą spalvinę paletę, taip supaprastinant segmentavimo ribų nustatymą.



1.2 pav. RGB ir HSV spalvinių paletėlių palyginimas, spalvų pasiskirstymas [11]

### 1.4.3. Briaunų nustatymas.

Briaunų (kraštinių, kontūrų) nustatymo veikimo principas – ieškoma kur staiga pasikeičia vaizdo ryškumas, kas gali reikšti pokytį gylyje, medžiagoje ar apšvietime. Tuomet šios vietos išskiriamos iš aplinkos. Gautas rezultatas – vientisos spalvos fonas, su priešinga spalva atvaizduotomis linijomis, idealiu atveju atitinkančiomis objekto kraštines. Šio metodo privalumas tas, kad iki šimto kartų sumažinama nuotraukoje esanti informacija, kas ne tik pagreitina tolimesnį vaizdo apdorojimą, bet ir sumažina jo failo dydį [12]. Dažnai kraštinių radimui naudojami *Sobel*, *Canny* ir *Laplaso* operatoriai.

#### Sobel operatorius

Analizuojant paveiksluką, tikrinamas ryškumo pokytis tarp gretimų pikselių x ir y ašių kryptimis. Ryškumo skirtumas tarp paveikslo  $A$  pikselių yra lygus  $A$  sąsūkai su filtro branduoliu  $k$ :



$$k_x = [-1, 0, 1] \quad (1.1)$$

Siekiant sumažinti triukšmų įtaką rezultatams, galima išplėsti filtro branduolius taip suglotninant duomenis. Briaunų radimas įprastai yra vienas pirmų etapų vaizdų atpažinimo operacijoje, o filtro branduolio dydis turi įtakos skaičiavimo greičiui, todėl egzistuoja daugybė branduolio variacijų su skirtingomis charakteristikomis. *Sobel* filtro branduolys yra priimtas kaip vienas iš geresnių kompromisų tarp greičio ir glotninimo [13]:

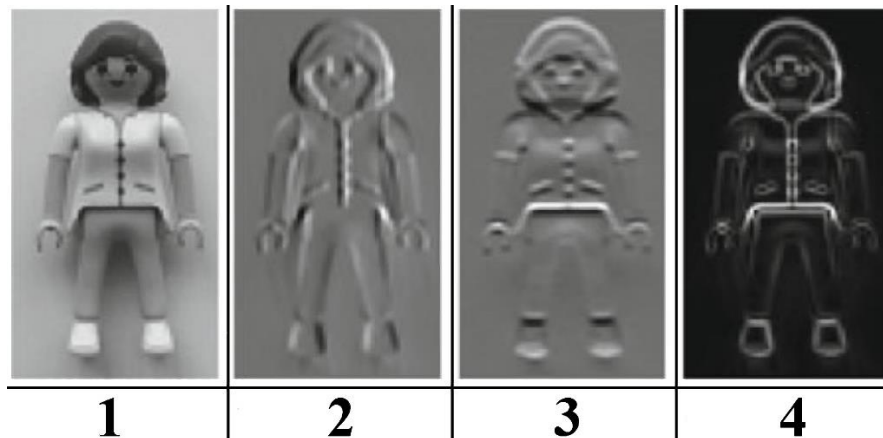
$$k_{S,x} = \frac{1}{4} \cdot \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad k_{S,y} = \frac{1}{4} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (1.2)$$

Tuomet gradientai  $A_x$  ir  $A_y$  yra lygūs duoto vaizdo  $A$  ir filtrų branduolių sąsūkai:

$$A_x = A * k_{S,x}, \quad A_y = A * k_{S,y} \quad (1.3)$$

Duotus gradientus taip pat galima išreikšti kaip dydį  $A_G$  ir kryptį  $A_\theta$ , o skaičiavimams pagreitinti  $A_G$  yra aproksimuojamas:

$$A_G = |A_x| + |A_y|, \quad A_\theta = \arctan\left(\frac{A_y}{A_x}\right) \quad (1.4)$$



1.3 pav. Sobel operatoriumi atliktas briaunų radimas [13]

1.3 paveiksle matyti darbo eiga naudojant *Sobel* operatorių – pirmu numeriu pažymėta originali nuotrauka, antru – gradientas x ašies kryptimi, trečiu – gradientas y ašies kryptimi, o ketvirtu – gradiento dydis (ryškesnė spalva žymi didesnę  $A_G$  vertę). Matyti, kad dėl glotninimo gaunamos platesnės kraštinės, todėl sunku nustatyti tikslią jų vietą.

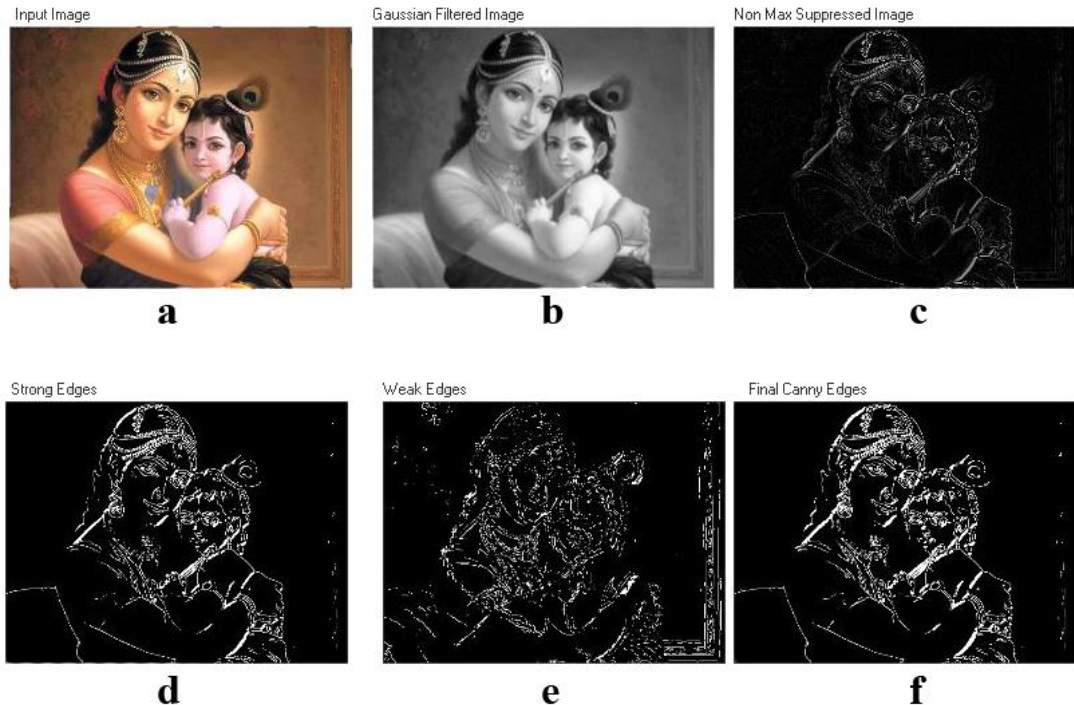
### Canny operatorius

Iki šiol dažnai naudojamas algoritmas yra John'o Canny pasiūlytas *Canny* briaunų nustatymas. Savo publikacijoje [14] Canny aprašė algoritmą skirtą pagerinti tuo metu naudojamiems briaunų aptikimo metodams ir suformulavo kelis optimizavimo kriterijus:

- Didelis signalo ir triukšmo santykis gradiento išėjime. Vadinasi, reikia nepraleisti pažymėti realių briaunų ir nepažymėti triukšmo sukurtų trikdžių.
- Maža pozicijos variacija. Pažymėtos briaunos turi būti kuo arčiau realių briaunų.

- Vienas realios briaunos pikselis turi atitikti tik vieną aptiktos briaunos pikselį.

Canny taip pat įrodė, kad pirmo lygio Gauso filtro išvestinė yra gera minėtų kriterijų aproksimacija. Vaizdo sąsūka su šiuo filtru atlieka glotninimą ir gradiento skaičiavimą vienu metu, yra geras našumo ir greičio derinys.



1.4 pav. Canny operatoriumi atliktas briaunų radimas [15]

Bendru atveju *Canny* briaunų aptikimo algoritmas vyksta tokia eiga:

1. Glotninimo, triukšmo pašalinimo naudojant Gauso filtrą (*b*).
2. Gradiento dydžio  $A_G$  ir krypties  $A_\theta$  skaičiavimo naudojant 1.4 formules.
3. Ne maksimalių reikšmių nuslėpimo (*c*). Taip pasiekama, kad kiekviena briauna būtų vieno taško pločio.  $A_G$  turi būti modifikuotas prieš slenkstinės vertės radimą, kadangi glotninimas išsklaido gradiento reikšmes. Tuo tikslu analizuojami  $3 \times 3$  pikselių ploteliai aplink kiekvieną pikselį  $p$ , randami du pikseliai su artimiausia gradiento kryptimi ir jei vienas iš jų turi didesnę gradiento dydį nei  $p$ ,  $p$  prilyginamas nuliui.
4. Dvigubo slenkščio metodo pritaikymo. Pikseliai, kurių gradiento dydis viršija aukštąjį slenkstį, klasifikuojami kaip stiprios briaunos (*d*), o kurių gradiento dydis yra žemiau apatinės slenkstinės vertės, kaip ne briaunos. Likusi dalis pažymima kaip silpnos briaunos (*e*) – tai taškai kurie galimai yra briaunos;
5. Paslepamos visos silpnos briaunos, kurios nesiliečia su stipriomis briaunomis, gaunamas galutinis briaunų vaizdas (*f*).

Remiantis Canny kriterijais yra pasiūlytos kelios alternatyvos, pavyzdžiui *Deriche* [16] filtras kuris optimizuotas į greitį, arba Freeman'o ir Adelson'o [17] kvadratūrinės filtrų poros skirtos geriau atpažinti kitokių rūšių kontūrams.

#### 1.4.4. Vaizdo briaunų nustatymo būdų palyginimas

Ramūnas Laskauskas ir Vytautas Vyšniauskas savo straipsnyje [18] išanalizavo aštuonis populiariausius vaizdo kontūrų nustatymo būdus. Tyrimui buvo panaudota 100 paveikslų su įvairiais elementais. Paveikslai verčiami nespaltvotais ir suglotninami priklausomai nuo jo turinio. Kiekvienam paveiksliui autoriai parinko optimalius parametrus, kad geriausiai matytųsi kontūrai ir kiekvienam metodui fiksavo slenkstines reikšmes. Pasak autorių, geriausiai kontūrai išskiriami, kai slenkstinės reikšmės arti 100% ir siaurame diapazone. Rezultatai matomi 2 lentelėje.

2 lentelė. Vaizdo briaunų nustatymo būdų palyginimas

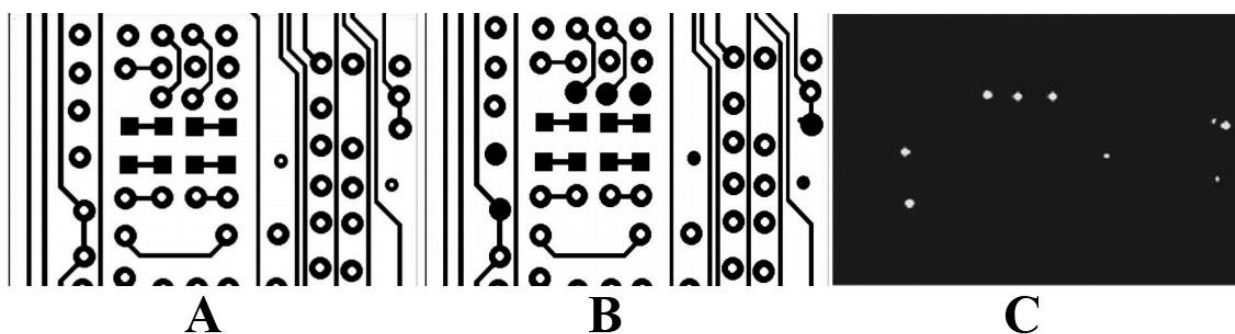
Metodas	Slenkstinės reikšmės diapazonas		Dažniausiai pasitaikė (%)
	Min.	Max.	
Canny	95,0	98,0	96,0 (25%)
Prewitt	97,5	99,5	99,0 (51%)
Roberts	97,0	99,0	99,0 (50%)
Sobel	97,0	99,5	99,0 (52%)
Laplacian	99,5	100	100 (49%)
LoG	97,0	99,5	99,0 (46%)
Zerocross	99,5	100,0	100,0 (67%)
Marr-Hildreth	99,5	100,0	100,0 (49%)

Apibendrinant pastebėta, kad briaunų išskyrimo kokybė mažai priklauso nuo pasirinkto metodo, o labiau nuo pačio paveikslo kokybės. Taip pat nepastebėta jokio ryšio tarp pasirinkto metodo ir paveikslo scenų grupių.

#### 1.5. Vaizdų palyginimo metodai

##### 1.5.1. Paveikslų atimties metodas

Naudojant paveikslų atimties metodą, kiekviename pikselyje skaičiuojamas skirtumas tarp vaizdų. Vaizdai turi būti vienodos raiškos ir spalvinės gamos, todėl metodas dažniau naudojamas dvispalviams vaizdams, pavyzdžiui spausdintinių plokščių takelių defektams išskirti. Paveikslų atimties metodo eiga pavaizduota 1.5 paveiksle. Prieš paveikslų atimties operaciją etaloninis (A) ir tikrinamas (B) vaizdai turi būti sutapdinti, tuomet naudojantis XOR loginiu operatoriumi išskiriami nesutampantys pikseliai (C).



1.5 pav. Paveikslų atimties metodo eiga, spausdintinių plokščių patikrai [19]

## 1.5.2. Koreliacija ir kiti panašumo nustatymo metodai

Norint skaitine reikšme įvertinti vaizdų panašumą, galima pasinaudoti keletu skirtingų panašumo skaičiavimo metodų. Metodai pritaikyti įvertinti rezultato tikslumą atliekant šabloninio vaizdo paiešką didesniame, tikrinamame vaizde. Šiame darbe bus išbandomi du panašumo skaičiavimo metodai, tai kvadratinių skirtumų suma (angl. *Sum of Squared Differences*) [20]:

$$SSD(u, v) = \sum_{i=0}^M \sum_{j=0}^N (f(i, j) - g(i + u, j + v))^2 \quad (1.5)$$

Koreliacija (angl. *Cross Correlation*) :

$$CC(u, v) = f(i, j) \cdot g(i + u, j + v) \quad (1.6)$$

Čia  $u$  ir  $v$  – koordinatės kuriose skaičiuojamas įvertis,  
 $i$  ir  $j$  – lyginamų vaizdų pikseliai,  
 $M$  ir  $N$  – vaizdo raiška atitinkamai  $x$  ir  $y$  kryptimis,  
 $f$  - vaizdas kuriame vykdoma paieška,  
 $g$  – ieškomas šablonas.

## 1.6. Programinė įranga

Vaizdų atpažinimui ir apdorojimui yra sukurta daug bibliotekų palengvinančių programavimą:

- *OpenCV* tai atviro kodo biblioteka, suteikianti prieigą prie daugiau kaip 500 algoritmų ir funkcijų skirtų realaus laiko vaizdų apdorojimui ir kompiuterinės regos realizavimui. Veikia populiariausiose operacinėse sistemose, programuojama C, C++ ir *Python* kalbomis [21].
- *Insight Segmentation and Registration Toolkit (ITK)* yra kita atviro kodo biblioteka skirta paveikslukų segmentavimui ir registravimui, labiau orientuota medicininiais duomenimis. Veikia populiariausiose operacinėse sistemose, programuojama C++ ir *Python* kalbomis [22].
- *Integrating Vision Toolkit (IVT)* yra dar viena atviro kodo biblioteka, parašyta ir programuojama C++ kalba. Autoriai pažymi lengvai suprantamą ir modifikuojamą kodą kaip vieną iš pagrindinių privalumų [23].
- *Visualization Toolkit (VTK)* tai atviro kodo biblioteka programuojama C++, *Python* ir *Java* kalbomis, palaikanti lygiagretų apdorojimą, turinti daugybę algoritmų ir integruotų duomenų bazių. [24]

### 1.6.1. Bibliotekų palyginimas

Bibliotekoms palyginti buvo nustatyti projektuojamos sistemos reikalavimai ir sudaryta trečia lentelė.

3 lentelė. Vaizdo atpažinimo bibliotekų palyginimas

-	<b>OpenCV</b>	<b>ITK</b>	<b>IVT</b>	<b>VTK</b>
Turi realizaciją C++ programavimo kalba	+	+	+	+
Turi realizaciją <i>Python</i> programavimo kalba	+	+	-	+
Veikia Linux ir Windows operacinėse sistemose	+	+	+	+
Turi segmentavimo pagal spalvą galimybę	+	+	+	+
Turi realizuotą SURF algoritmą	+	-	-	-
Leidžia nuskaityti vaizdą tiesiogiai iš kameros	+	-	+	-

Iš apžvelgtų bibliotekų tinkamiausia yra *OpenCV*, kuri turi plačiausią algoritmų pasiūlą, gali būti programuojama *Python* kalba ir taip pat leidžia tiesiogiai nuskaityti vaizdą iš kameros.

## 2. Metodologinė dalis

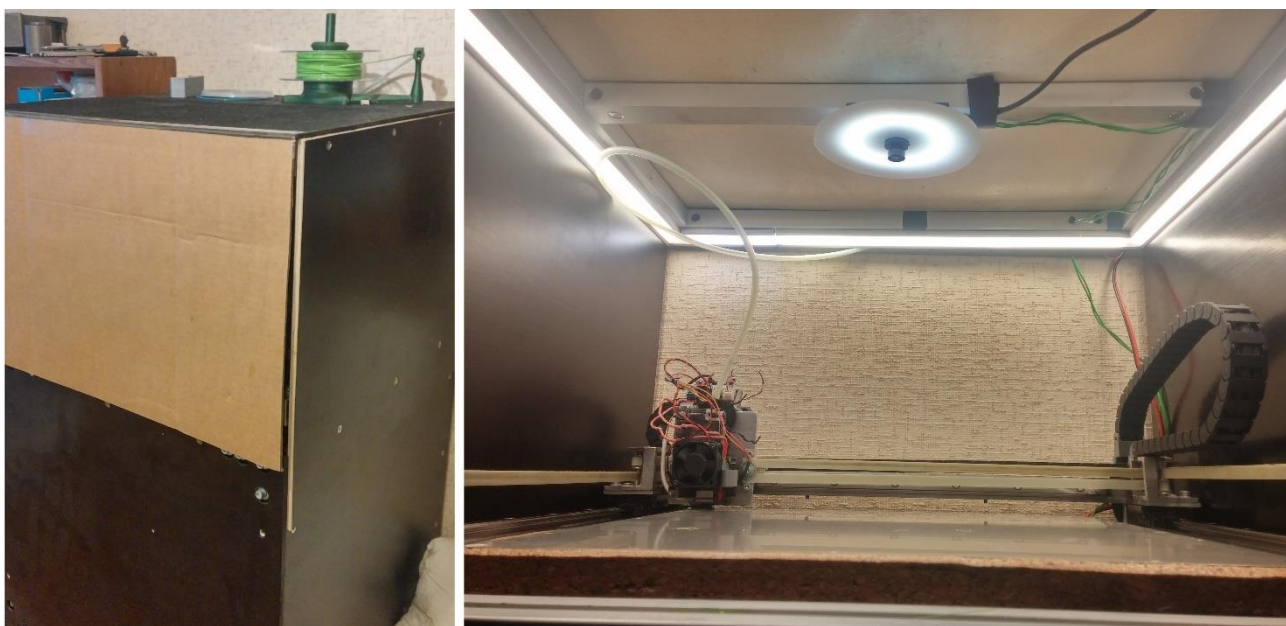
### 2.1. Naudojama eksperimento įranga

#### 2.1.1. Spausdintuvas ir jo modifikacijos

Tyrime naudojamas šio darbo autoriaus sukurtas ir pagamintas CoreXY tipo MEX spausdintuvas su Z ašimi judančiu spausdinimo staliuku. XY darbo plotas 300x300mm, *SKF LLTHR 15* linijinės kreipiančiosios, pozicionavimui naudojamos 200 žingsnių Nema23 pavaros, teorinis pozicionavimo tikslumas nustačius 16 mikro-žingsnių – 8,84μm.

Z ašies darbinis aukštis 0-400mm, pozicionavimui naudojama 400 žingsnių Nema23 pavara su 25:1 sliekiniu reduktoriumi ir HTD 5M perdava, apvalios 12mm kreipiančiosios, teorinis vieno žingsnio pozicionavimo tikslumas – 0,01mm.

Siekiant panaikinti pašalinio apšvietimo įtaką detalės nuotraukoms, aplink darbinį plotą sumontuotas atidaromas gaubtas. Jo viduje pritaikytas darbo vietos apšvietimas – perimetru 30° kampu į spausdinimo platformą nukreiptos juostos ir žiedinis aplink kamerą sumontuotą tiesiai virš spausdinamos detalės. Apšvietimui panaudoti 2835 tipo šviesos diodai, juostoje sumontuoti 120 LED/m žingsniu, deklaruojamas aukštas spalvų atkūrimo indeksas (>90). Siekiant sušvelninti šešėliams, kiekvienam šviesos šaltiniui sumontuoti šviesą sklaidantys difuzoriai.



2.1 pav. Spausdintuvas su gaubtu ir LED apšvietimu

### 2.1.2. Kamera

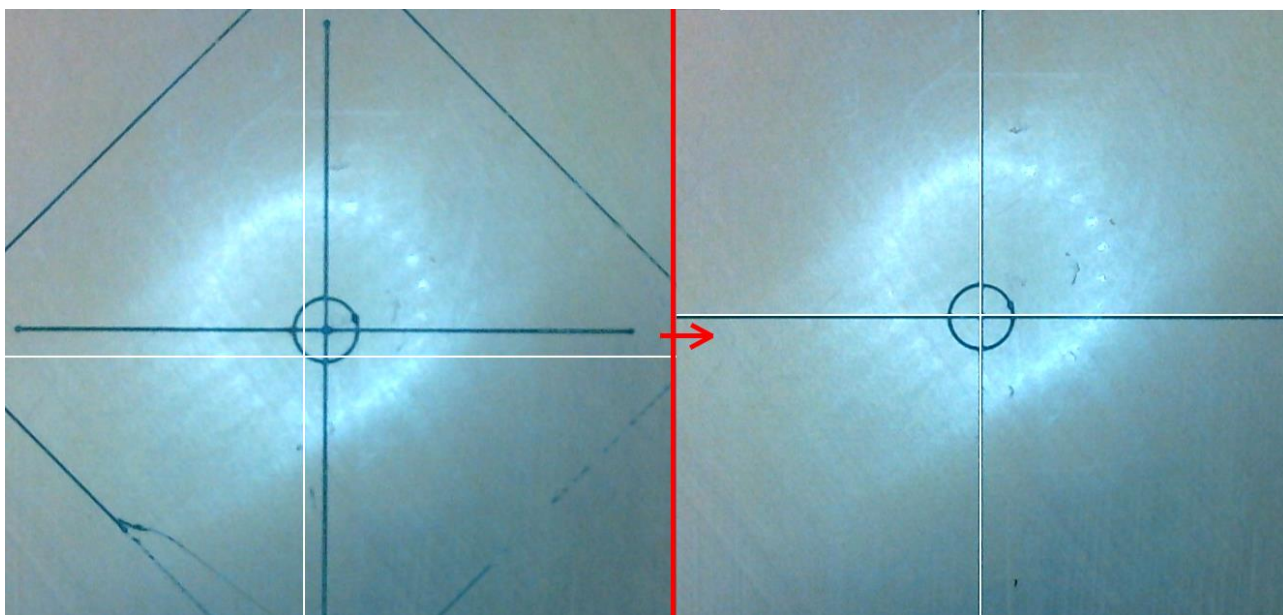
Kamera sumontuota tiesiai virš spausdinimo stalo centro. Tokia pozicija pasirinkta įvertinus ieškomų defektų pobūdį, bei siekiant išgauti kuo tolygesnį fono vaizdą, efektyvesniam objektų išskyrimui. Kameros pozicija ir posūkio kampas spausdintuvo ašių atžvilgiu suderintas spausdinant taikinio formos objektą, ir reguliuojant kameros tvirtinimo mechanizmą siekiant sutaptinti centrus, bei pasisukimą. Derinimo eigos nuotrauka 2.3 paveiksle. Svarbiausi kameros parametrai pažymėti ketvirtoje lentelėje.

4 lentelė. Tyrimui aktualios kameros specifikacijos

<b>Jutiklis</b>	4MP OV4689
<b>Raiška ir maksimalus kadrų dažnis</b>	1920X1080 ir 60 kadrai per sekundę; 1280 x 720 ir 120 kadrai per sekundę; 640x360 ir 330 kadrai per sekundę.
<b>Sąsaja</b>	USB 2.0
<b>Objektyvas</b>	Mažo iškraipymo



2.2 pav. Naudojamos kameros išvaizda.



2.3 pav. Kameros vaizdo centro ir spausdinimo centro kalibravimas

### 2.1.3. Spausdintuvo valdiklio paruošimas

Naudojamas *Duet 2 Wifi* valdiklis tinka šiam tyrimui, dėl prieinamų papildomų įėjimų ir išėjimų bei nesudėtingo jų programavimo tiesiai naršyklėje, taip pat dėl plačių *makro* failų pritaikymo galimybių nepriklausomai nuo gaminiui spausdinti naudojamo g-kodo. Svarbiausios valdiklio specifikacijos pateiktos penktoje lentelėje.

5 lentelė. Tyrimui aktualios spausdintuvo valdiklio specifikacijos [25]

<b>Pavadinimas</b>	Duet 2 WiFi
<b>Komunikacijos galimybės</b>	2.4GHz WiFi, USB sąsaja, SD kortelės skaitytuvas, nuoseklusis prievadas, SPI sąsaja išplėtimo moduliams
<b>Įėjimai ir išėjimai</b>	3 temperatūros jutiklių įėjimai (PT1000 arba 100k termistoriui) 5 galinio judesio jutiklių įėjimai 3 kaitinimo elementų išėjimai (7A) 3 reguliuojami ir 2 nereguliuojami ventiliatorių išėjimai (1A) 7 laisvai programuojami skaitmeniniai įėjimai/išėjimai

Tyrimui įgyvendinti, valdiklyje reikėjo sukurti kelias *makro* komandas bei nustatyti papildomus parametrus. *Duet 2 Wifi* valdiklyje panaudota RRF 3.2 programinės įrangos versija.

Naudojami įėjimai ir išėjimai aprašomi *settings.g* faile, kuris nuskaitomas kiekvieną kartą įjungiant spausdintuvą. Kodo atsakingo už įėjimų ir išėjimų priskyrimą kontaktams *settings.g* faile iškarpa pateikta 2.4 paveiksle. Komanda M950 priskiria funkciją (P – išėjimas, J – įėjimas) atitinkamam kontaktui C.

```

86 M950 P5 C"exp.18";           ;Komanda fotografavimui atlikti
87 M950 P6 C"e0heat" Q65000    ;Perimetro LED juosta, PWM dažnis 65kHz
88 M950 P7 C"fan2"             ;LED ziedas
89 M950 J1 C"exp.24";          ;Darbo stabdymo iejimas
90 M950 J2 C"exp.19";          ;Darbo pratesimo iejimas
91

```

2.4 pav. settings.g failo kodo iškarpa

Programa rašoma į *Patikra.g* makro failą, kuris iškviečiamas panaudojant *M98 P"Patikra.g"* komandą. Ši komanda į spausdinamo objekto kodą įterpiama ruošiant šablono modelius, pasirinktais momentais. *Patikra.g* makro faile esantis kodas įsimeina dabartinę poziciją, patraukia spausdinimo galvutę į šalį, įjungia darbo vietos apšvietimą ir siunčia signalą vaizdų surinkimo valdikliui. Kodas ir jo paaiškinimas pateiktas 2.5 paveiksle.

```

4
5 M83           ; relative extruder moves
6 G1 E-3 F3600 ; retract 3mm of filament
7 G91           ; relative positioning
8 G1 Z5 F360   ; lift Z by 5mm
9 G90           ; absolute positioning
10 G1 X0 Y335 F6000 ; go to X=0 Y=0
11 M42 P6 S1    ; ijuniamas apsvietmas
12 M42 P7 S1
13 M42 P5 S1    ;Signalas fotografavimui
14 G4 P200      ;laukia 0,2s
15 M42 P5 S0    ; nutraukia signala
16

```

2.5 pav. Patikra.g failo kodas

Gavus signalą iš duomenų surinkimo valdiklio aktyvuojama *Trigger2.g* makro faile esanti *M24* komanda, kuri aktyvuoja už darbo pratęsimą atsakingą failą *resume.g*. Taip daroma dėl to, nes rankiniu būdu valdymo panelėje pratęsus darbą aktyvuojamas tas pats *resume.g* failas, ir modifikuojant kodą nereikia prižiūrėti dviejų skirtingų failų. Pratęsimo faile yra nurodytos judesio virš prie spausdintuvo pritvirtinto šepetėlio, bei grįžimo į prieš sustabdymą išsaugotą vietą komandos. Šepetėlis naudojamas nuvalyti ištryškusiame plastikui nuo spausdinimo galvutės tam, kad nesukurti



papildomų prilipusio plastiko gabaliukų ant detalės, ir taip išvengti papildomų defektų. Darbo pratęsimo kodas 2.6 paveiksle.

```
1  
2 G1 X-10 Y350 ;valymo procedura  
3 G1 X-10 Y300  
4 G1 X-10 Y350  
5 G1 X-10 Y300  
6 G1 X0 Y350  
7  
8 G1 R1 X0 Y0 Z5 F6000 ; go to 5mm above position of the last print move  
9 G1 R1 X0 Y0 ; go back to the last print move  
10 M83 ; relative extruder moves  
11 G1 E3 F3600 ; extrude 3mm of filament  
12  
13
```

2.6 pav. Darbo pratęsimo, galvutės valymo kodas

### 2.1.4. Fotografijų surinkimas

Fotografijų surinkimui panaudotas *Raspberry Pi 1 B+* mikrovaldiklis. Tyrimo metu valdiklis tik surenka detalių fotografijas tolesniam apdorojimui todėl didelė greیتaveika nereikalinga. Siekiant realiu laiku įgyvendinti sukurtą defektų aptikimo algoritmą galimas nesudėtingas atnaujinimas į naujesnės kartos *Raspberry Pi* valdiklį.

Abejims valdikliams tarpusavyje komunikuoti panaudoti skaitmeniniai signalai, signalizuojantys apie darbo stabdymą ir momentą nuotraukai padaryti, bei darbo pratęsimą. *Raspberry Pi* valdiklyje panaudota *Raspberry Pi OS 5.4* versija, programa parašyta *Python* programavimo kalba. Programa laukia skaitmeninio signalo iš spausdintuvo valdiklio signalizuojančio apie patikros pradžią, išsaugo nuotrauką ir pratęsia spausdintuvo darbą. Kodas pateiktas 2.7 paveiksle.

```
import subprocess  
import RPi.GPIO as GPIO  
import time  
  
GPIO.setmode(GPIO.BOARD)  
GPIO.setup(12, GPIO.IN, pull_up_down=GPIO.PUD_UP)  
GPIO.setup(13, GPIO.OUT)  
i=0  
try:  
    while True:  
        input_state = GPIO.input(12)  
        if input_state == 0:  
            subprocess.call("fswebcam -d /dev/video0 -r 1920x1080 -S0 --no-banner --set brightness=120 --set contrast=5 | "+str(i)+"pic.png", shell=True)  
            i=i+1  
            time.sleep(0.2)  
            GPIO.output(13, GPIO.HIGH)  
            time.sleep(0.2)  
            GPIO.output(13, GPIO.LOW)  
        except KeyboardInterrupt:  
GPIO.cleanup()
```

2.7 pav. Kodas nuotraukoms padaryti, gavus signalą.

Siekiant užtikrintai gauti įvairių defektų vaizdų, buvo taikomi metodai imituojantys defekto atsiradimą:

- Nuotraukos su detalės atsiklijavimo defektu gautos rankiniu būdu judinant atsiklijavusią detalę į įvairias spausdinimo stalo vietas.
- Deformuotų detalių vaizdai gauti keičiant modelio kraštinių santykį prieš spausdinimą, imituojant atsilaisvinusio dantračio keliamas pasekmes.
- Sluoksnių pasislinkimo vaizdai gauti sutrukdant spausdinimo galvutės judesį, prieš tai sumažinus žingsninių pavarų srovę. Dėl to būna praleidžiamas atsitiktinis skaičius žingsnių ir imituojama atsilaisvinusių dirželių ar perkaitusių pavarų keliami defektai. Dėl

neprognozuojamo defekto dydžio, dalis nuotraukų buvo modifikuotos vaizdo redagavimo programa, kopijuojant ir paslenkant aukščiausią sluoksnį.

- Spausdinimo ore vaizdai gauti trumpam sustabdžius plastiko tiekimą nenutraukiant spausdinimo ir po kelių sluoksnių atstatant plastiko tiekimą.
- Prilaikančiųjų konstrukcijų defektai imituojami deformuojant konstrukcijas rankiniu būdu.
- Kokybiškų detalių vaizdai renkami kiekviename sluoksnyje iki atliekant aukščiau nurodytus veiksmus defektams išgauti.

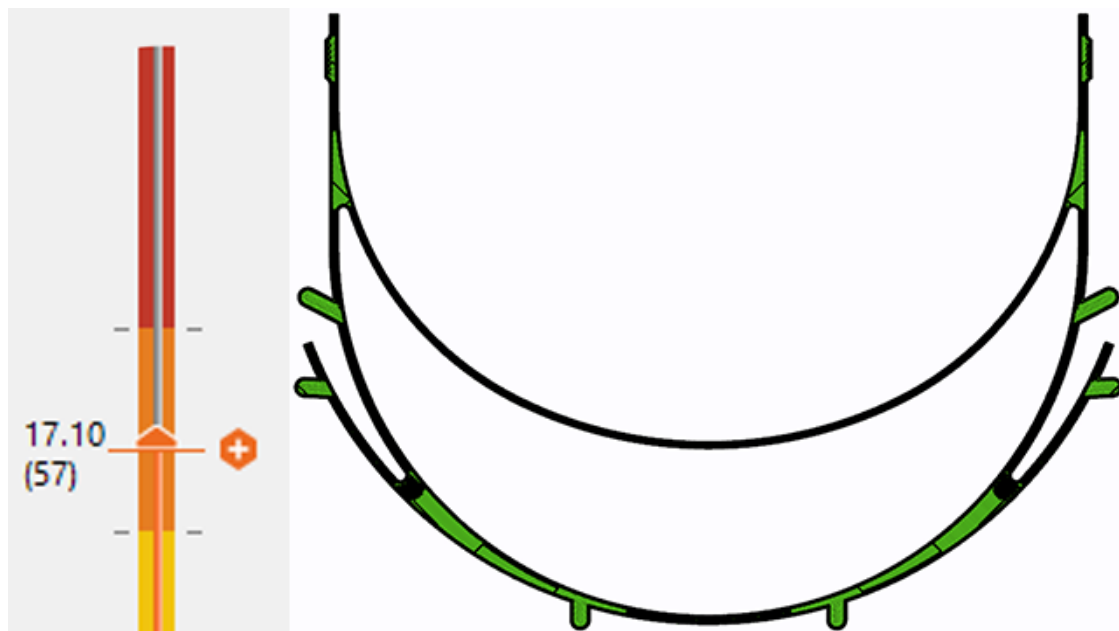
### 2.1.5. Šablono modelio paruošimas

Šablono modeliui sukurti naudojama modifikuota *PrusaSlicer 2.3.0* versija. Spausdinimo platformos tekstūra pakeičiama į vienspalvę, šiuo atveju baltą, lengvesniam detalės kontūrų išskyrimui. *Printer Settings > General > Size and Coordinates > Bed Shape > Texture*.

Komanda *M98 P“Patikra.g“* įterpiama pasirinktuose sluoksniuose panaudojant šalia šliaužiklio esančiu mygtuku, pavyzdys matomas 2.8 paveikslo kairėje pusėje.

Šablonas yra paruošiamas sudarytas iš dviejų dalių – šiuo metu spausdinamo sluoksniu išskirto juoda spalva ir visų jau atspausdintų sluoksnių išskirtą žalia spalva. Tai atliekama tam, kad būtų galima atskirai aptikti defektus paveikiančius tik vieną sluoksnį net jei jis yra mažesnis už visą detalę, o ne tik defektus paveikiančius visą detalę.

Pasirenkame *View > Top*, ir išsaugome reikalingus šablono vaizdus. Vieno šablono pavyzdys 2.8 paveikslo dešinėje pusėje. Tyrimui naudojamų objektų šablonai ruošiami rankiniu būdu koreguojant mastelį priklausomai nuo spausdinamo sluoksniu aukščio. Norint tiriamą sistemą naudoti nuolatos, rekomenduojama šį procesą automatizuoti.



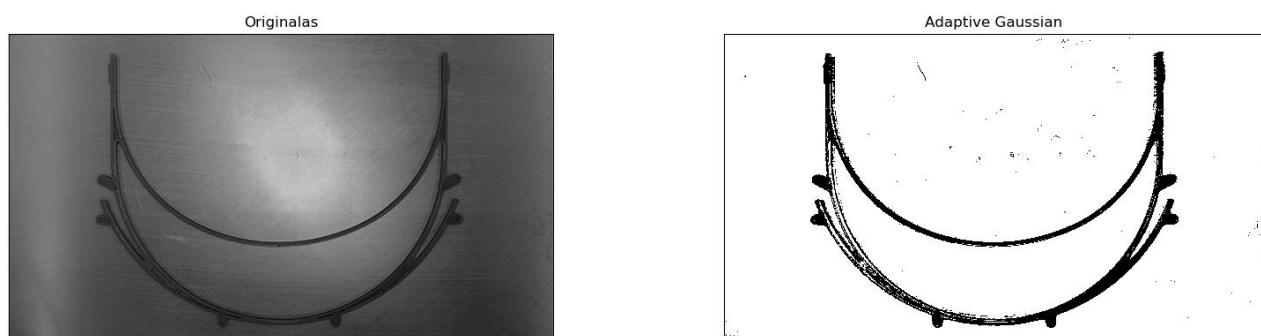
2.8 pav. Patikros kodo įterpimo šliaužiklis, ir modeliuoto šablono pavyzdys.

## 2.2. Objekto išskyrimo iš fono metodai

Kaip buvo aptarta 1.5 skyriuje, defektus galima aptikti lyginant detalės ir šablono vaizdus paveikslų atimties ar panašumo nustatymo metodais. Paveikslų atimties metodu palyginsime vaizdus po segmentavimo, o šablono paieškos metodu ieškosime atitinkamų vidinių struktūrų briaunų. Kiekvienam defektui bus suteikiamas įvertis ir pagal šio įvertčio dydį bus sprendžiama ar tai mažas, galbūt toleruotinas defektas, kaip prilipęs plastiko gabaliukas, ar tai didesnis neatitikimas, kaip pasislinkę sluoksniai, nuvirtusios atramos ar atsiklijavusi detalė.

### 2.2.1. Objekto išskyrimas segmentacijos metodais

Priklausomai nuo naudojamo plastiko spalvos ir ryškumo, pritaikius segmentacijos pagal ryškumą metodą, galima nesudėtingai išskirti objektą iš fono. Tačiau su šiame tyrime naudojamomis plastiko spalvomis nepavyko pasiekti patenkinamų rezultatų nei tradiciniais, nei adaptyviais segmentavimo pagal ryškumą metodais – buvo išskiriamos spausdinimo platformos dalys, arba neišskiriamos spausdinamos detalės dalys. Geriausias gautas adaptyvaus segmentavimo pagal ryškumą rezultatas pavaizduotas 2.9 paveiksle.



2.9 pav. Spausdinimo platformos įtaka segmentacijai pagal ryškumo vertę.

Segmentavimas pagal ryškumo vertę idealiai tinka išskirti detalės sluoksniams šablone. Pavyzdys 2.10 paveiksle.



2.10 pav. Segmentavimo pagal ryškumo vertę pritaikymas modeliuoto šablono vaizdui

Kadangi ruošiant kodą gaminiui reikia žinoti kokį plastiką naudosime, žinant jo spalvą galima pasirinkti tinkamus parametrus ir panaudoti segmentavimo pagal spalvą algoritimą. Dirbama HSV spalvinėje paletėje. Segmentavimo pagal spalvą rezultatas 2.11 paveiksle.



2.11 pav. Segmentavimo pagal spalvą rezultatas.

Iš rezultato matome, kad objektas išskiriamas pakankamai tiksliai, tačiau yra pašalinių triukšmų tiek fone, tiek detalės kraštinėse. Šiuos triukšmus galima pašalinti naudojant *Erode* ir *Dilate* funkcijas. Šiuo atveju panaudota funkcija *Opening*, apjungianti *Erode* ir *Dilate* į vieną komandą. Rezultatas 2.12 paveiksle.



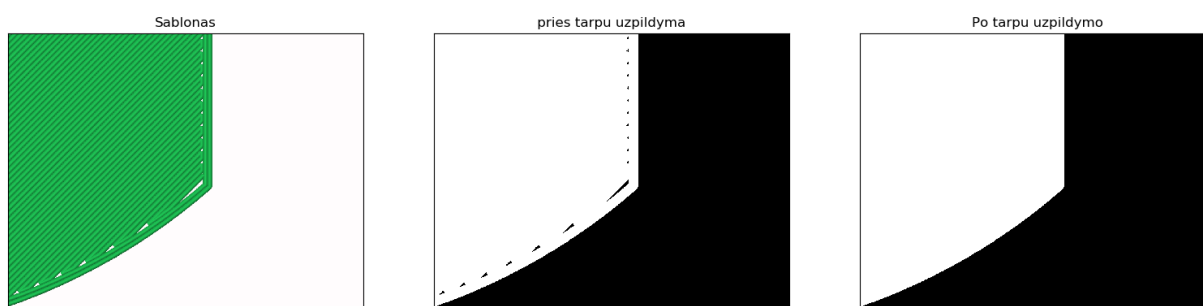
2.12 pav. Segmentavimo pagal spalvą ir filtravimo rezultatas

Vaizde po segmentavimo gerai išskiriami objekto išoriniai kontūrai, todėl nesunku pastebėti ne tik didelius, objekto padėtį keičiančius defektus, bet ir ant spausdinamo objekto išorės esančius smulkesnius defektus. Tačiau jei spausdinama detalė yra siaurėjanti į viršų, mažesnio ploto sluoksniai bus paslepiami ir juose esantys defektai liks nepastebėti. Siekiant išskirti šiuos sluoksnius ir juose esančius defektus bus naudojamas briaunų nustatymas. Tam kad būtų galima išskirti tik spausdinamo sluoksnio briaunas, šablonai paruošiami iš dviejų spalvų – juoda spalva žymi naujausią sluoksnį. Kaip ir segmentavimo pagal ryškumą būdu, panaudojus spalvinį segmentavimą gali būti išskiriami šio sluoksnio kontūrai tolesniam darbui. Spalvinio segmentavimo privalumas tas, kad galima atskirai išskirti kiekvieną modelio sudedamąją dalį (kraštinės, užpildą, prilaikančias konstrukcijas). Pavyzdys 2.13 paveiksle.



2.13 pav. Spalvinės segmentacijos pritaikymas šablonui

Šablonai modeliuojami remiantis idealiu plastiko išspaudimu ir tiksliu spausdinimo galvutės judėjimu. Todėl vietose kur netelpa nustatyto pločio plastiko juosta yra paliekami tarpeliai. Realiose detalėse šių tarpelių nesimato, nes išsilydęs plastikas yra takus ir juos užpildo. Siekiant kuo labiau sutaptinti šablono ir realių detalių vaizdus, šie tarpeliai yra užpildomi panaudojus *Closing* funkciją. Priartintas šablono vaizdas ir tarpelių užpildymo eiga matomi 2.14 paveiksle.



2.14 pav. Šablono tarpelių užpildymas

## 2.2.2. Briaunų nustatymo metodo panaudojimas

Siekiant išskirti spausdinamo objekto dalis kurios nepastebimos segmentavimo būdu, pasirinkta naudoti briaunų nustatymo algoritmus. Briaunų radimas gali suteikti informacijos apie vidinius detalės kontūrus, kartu pastebint užpildo defektus ar siaurėjančiuose sluoksniuose pasitaikančius defektus.

Metodo privalumas tas, kad nereikia iš anksto žinoti spalvos, kaip segmentavimo atveju. Prieš tai nuotrauka filtruojama *bilateral* filtru, kurio vertė parinkta siekiant pašalinti kuo daugiau stalo rašto įtakotų trikdžių, tuo pat metu nepašalinant tikrų objekto kraštinių. Naudojamo filtro dydis – 5; sigma vertės – 150. Tačiau kaip matyti iš 2.15 paveikslo, net ir po nuotraukos filtravimo pritaikius *Canny* briaunų radimo algoritmą, nepavyko idealiai išskirti visų objekto briaunų tuo pat metu neįvedant daugybės pašalinių, spausdinimo stalo rašto kuriamų, trikdžių.



2.15 pav. Briaunos gautos *Canny* algoritmu.

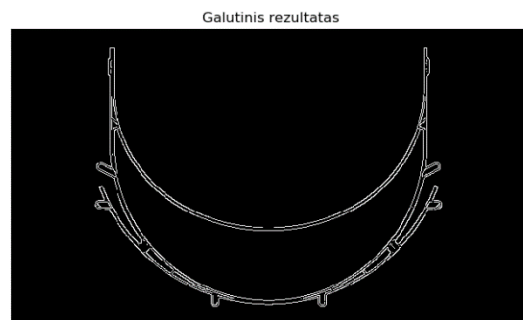
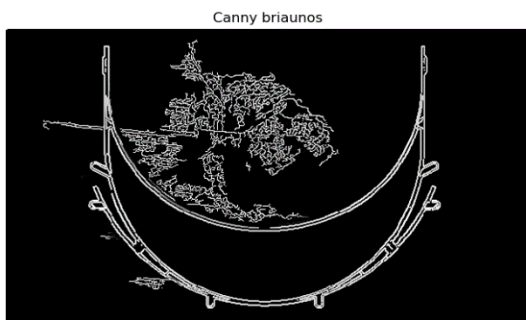
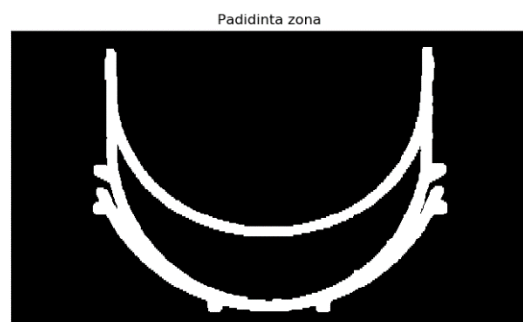
Papildomai išbandytas briaunų aptikimas naudojant *Sobel* ir *Laplace* operatorius. Rezultatas matomas 2.16 paveiksle. Naudojant Laplaso operatoriumi paremtą kraštinių nustatymo algoritmą, dėl spausdinimo stalo rašto sukuriama per daug triukšmų ir kai kurios kraštinės nustatomos silpnai. Todėl tik *Sobel* ir *Canny* briaunos bus naudojamos tolesniuose tyrimuose.



2.16 pav. Sobel ir Laplaso operatoriais paremtas kraštinių radimas

### 2.2.3. Vidinių struktūrų išskyrimas

Apjungiant spalvinio segmentavimo ir briaunų nustatymo metodus galima išskirti objekto vidines struktūras bei pašalinti stalo rašto kuriamus trikdžius. Jautriu *Canny* algoritmu gautos briaunos ir spalvinės segmentacijos būdu gautos detalės ribos apjungiamos panaudojant *bitwise\_and* funkciją, paliekančią tik bendrus abiejų vaizdų pikselius. Prieš tai segmentacijos rezultatas išplečiamas panaudojus *Dilate* funkciją, siekiant palikti kuo galima tikslesnę *Canny* briaunų rezultatą. Po apjungimo lieka tik briaunos patenkančios į spalvinio segmentavimo zoną – vadinasi, tik detalės briaunos. Darbo eiga ir rezultatai matomi 2.17 paveiksle.



2.17 pav. Briaunų aptikimo ir segmentavimo apjungimas

### 3. Tiriamoji dalis

#### 3.1. Defektų aptikimo sistemos parametrų nustatymas

Tam, kad išsirinkti galutinėje defektų aptikimo sistemoje naudojamus briaunų nustatymo ir šablono paieškos metodus, bei nustatyti sistemos parametrus, buvo atlikti bandymai su įvairių formų, spalvų ir defektų spausdiniais. Įvertinus rezultatus buvo parinkti metodai išskiriantys didžiausią dalį defektų.

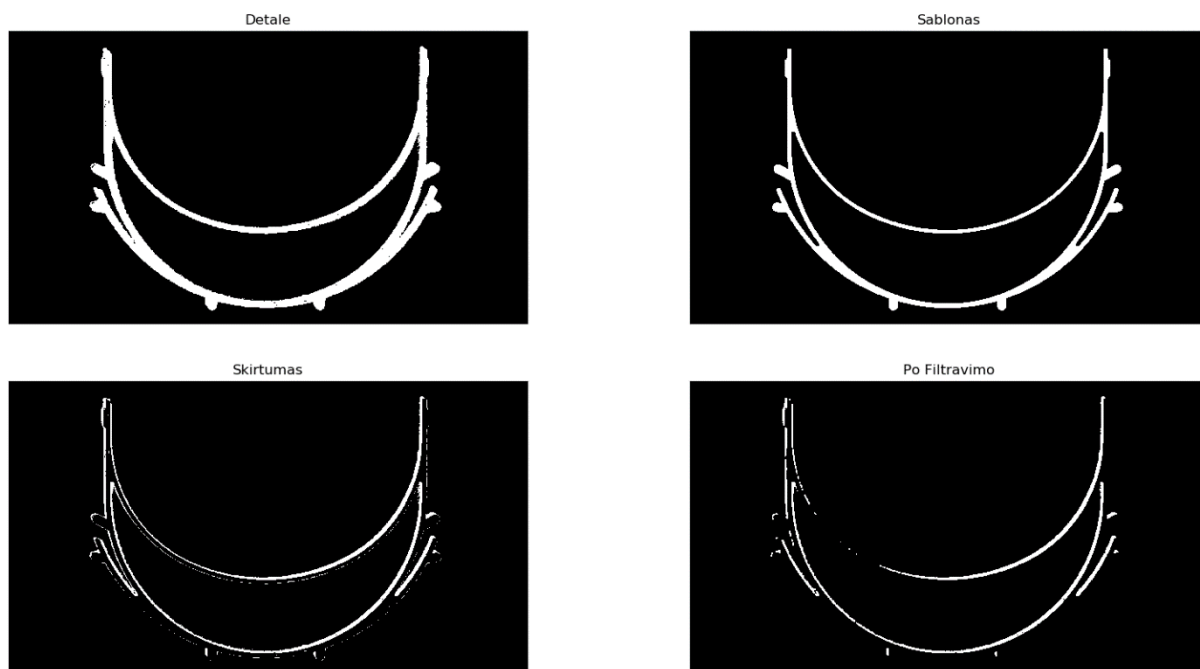
Tam tikslui buvo paruošta 25-ios skirtingų detalių bei jas atitinkančių šablonų fotografijos. Nuotraukos suskirstytos į grupes po 5, pagal fotografuojamą detalę. Kiekvienoje grupėje pirmos dvi detalės yra kokybiškos, be defektų. Likę trys nuotraukos vaizduoja skirtingus defektus – spausdinimo ore, atsiklijavimo ir sluoksnių pasislinkimo. Naudotos nuotraukos pirmame priede, o detalesnis paaiškinimas žemiau:

- Numeriais nuo 1 iki 15, pažymėtos *Prusa Face Shield* [26] modelio detalės. Numeriai 1-5 spausdinti žalia spalva, 6-10 – oranžine, o 11-15 – pilka. Pilko plastiko atspalvis parinktas kuo artimesnis spausdinimo stalo spalvai, siekiant patikrinti detalės išskyrimo iš fono kokybę.
- Siekiant patikrinti algoritmų veikimą su labai paprastomis detalėmis neturinčiomis išskirtinių savybių, kampų ir nesiskiriančių tarp skirtingų sluoksnių, sumodeliuotas žiedo formos modelis spausdintas žalios spalvos plastiku. Numeriais nuo 16 iki 20 pažymėtos šio modelio fotografijos.
- Kiekvienoje aukščiau minėtoje grupėje pirmos dvi nuotraukos yra be detalės defektų. Trečia - vaizduoja situaciją kai defektas nebuvo aptiktas iš karto ir plastikas buvo liejamas ore bei susipynė į atsitiktinės formos gniužulą. Ketvirtoje nuotraukoje detalė yra atsiklijavusi nuo spausdinimo stalo, o penktoje pavaizduotas sluoksnių pasislinkimo defektas.
- Numeriais nuo 21 iki 25 pažymėti skirtingi pusskritulio formos detalės sluoksniai spausdinti žalios spalvos plastiku. Šios detalės modelis buvo sukurtas šio darbo autoriaus, siekiant patikrinti vidinių struktūrų bei prilaikančiųjų konstrukcijų defektų nustatymo kokybę. Todėl 23 ir 24 numeriais pažymėtos nuotraukos vaizduoja sluoksnių pasislinkimo defektą kai įtakojami sluoksniai yra „apsupti“ žemiau esančių, jau atspausdintų sluoksnių. 25 numeriu pažymėtas prilaikančiųjų konstrukcijų defektas.

##### 3.1.1. Defektų aptikimas naudojant segmentavimą

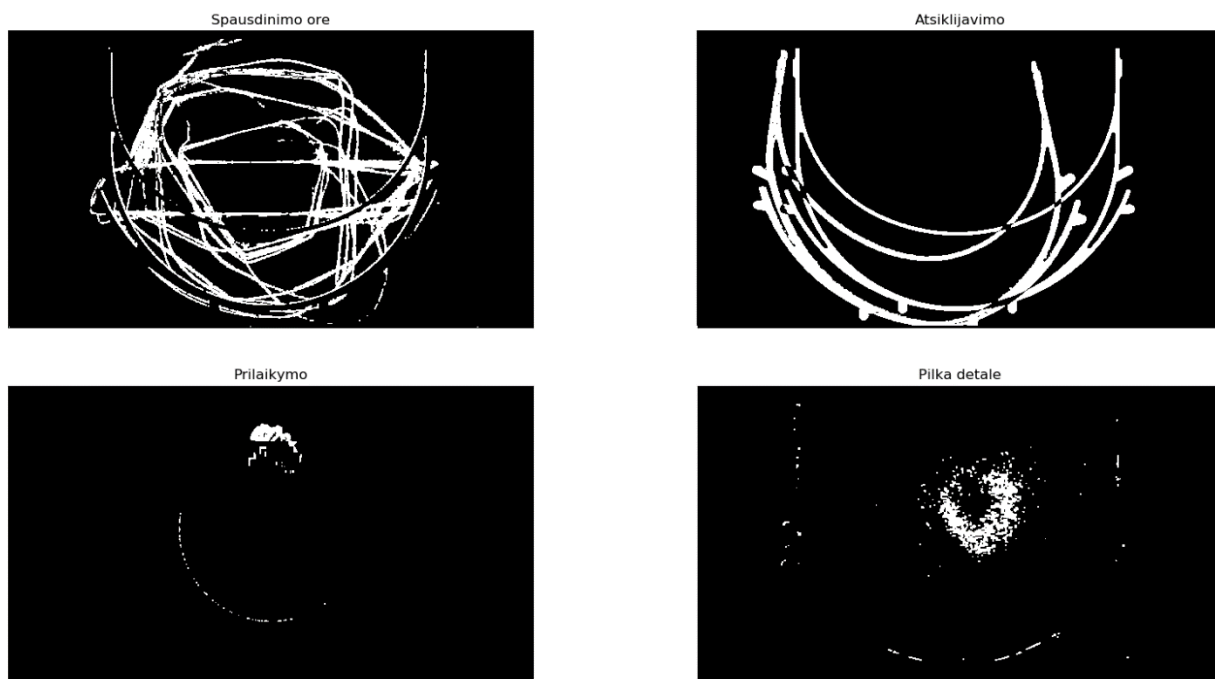
Atlikus spausdinamo objekto ir šablono segmentavimą 2.2.1 skyrelyje aprašytu metodu, galima gautiems vaizdams pritaikyti *bitwise\_xor* funkciją, taip išskiriant visus ne bendrus pikselius. Tokiu būdu pažymimas ne tik už šablono ribų esantis plastikas, bet ir šablono vietos kuriose jo trūksta. Pritaikius *Erosion* ir *Dilation* filtrus, gauname tik didesnių nei pasirinktas filtro dydis defektų vaizdą. Defekto plotas (pikseliais) apskaičiuojamas naudojant *countNonZero* funkciją. Pagal defekto plotą, galima spręsti apie defekto svarbą ir nuspręsti ar stabdyti darbą. 3.1 paveiksle pavaizduotas šablonas ir detalė su pasislinkusių sluoksnių defektu, bei defekto išskyrimo ir filtravimo vaizdai.





3.1 pav. Sluoksnių pasislinkimo defekto išskyrimas po segmentacijos pagal spalvą.

Analogiškai išskiriami spausdinimo ore, detalės atsiklijavimo ir prilaikančiųjų konstrukcijų defektai. Pilku plastiku spausdinta detalė išskirta nekokybiškai ir klaidingai pažymimi defektai. Išskirtų defektų vaizdai 3.2 pav.



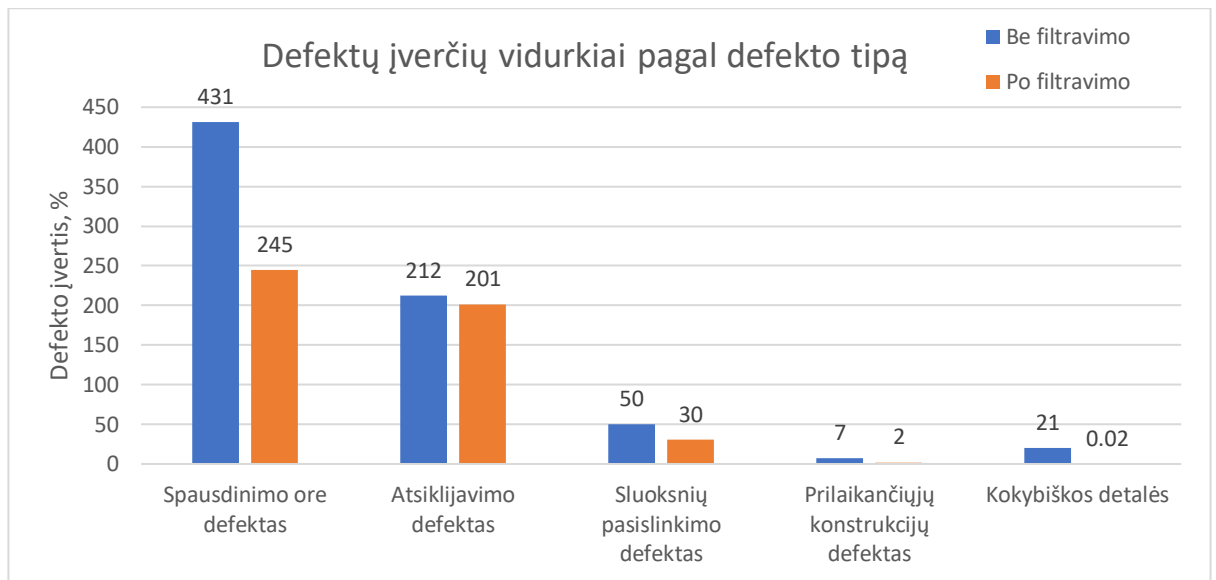
3.2 pav. Spausdinimo ore, atsiklijavimo ir prilaikančiųjų konstrukcijų defektų išskyrimas po segmentacijos pagal spalvą.

Kaip defekto įvertis panaudotas išskirto defekto ploto ir šablono ploto santykis, išreikštas procentais. Palyginti įverčiai iškarto po vaizdų palyginimo ir kai iš defekto vaizdo pašalinamos visos mažesnės nei 8 px ploto zonos. Rezultatai pateikti šeštoje lentelėje.

6 lentelė. Defekto įvertis gautas vaizdų atimties metodu, prieš ir po mažų plotų zonų filtravimo

Scenos Numeris	Be filtravimo	Po filtravimo
	Įvertis, %	
1	12.81	0.00
2	32.62	0.16
3	296.54	201.79
4	219.19	215.12
5	57.28	30.88
6	14.22	0.00
7	29.20	0.00
8	145.79	38.23
9	207.08	197.06
10	54.04	25.91
11	155.64	83.84
12	77.05	26.68
13	347.64	239.53
14	240.05	184.25
15	119.03	47.86
16	28.44	0.00
17	39.33	0.00
18	933.73	498.60
19	209.97	190.77
20	132.99	94.83
21	3.49	0.00
22	3.99	0.00
23	3.94	0.00
24	3.96	0.00
25	7.16	2.05

Kaip matome iš rezultatų, šis defektų aptikimo metodas dvi neigiamas savybes. Pirma iš jų – jeigu detalė yra siaurėjanti (nr. 21-25), aukščiau esantys sluoksniai gali būti visiškai uždengiami žemiau esančių sluoksnių taip užmaskuojant tik tam tikrą sluoksnį įtakojančius defektus (pasislinkimą ar atraminių konstrukcijų deformacijas). Be to, spalvinio segmentavimo būdu nepavyksta kokybiškai išskirti detalių kurių spalva sutampa su spausdinimo stalo spalva (nr. 11-15). Atsižvelgiant į tai, 3.3 paveiksle atvaizduoti defektų įverčių vidurkiai, sugrupuoti pagal defekto tipą, neskaičiuojant taisyklingai neiškirtų, pilkos spalvos detalių.



3.3 pav. Defekto įverčiai po segmentavimo, sugrupuoti pagal defekto tipą

Detalių neturinčių defektų (nr. 1, 2, 6, 7 ir t.t.) įverčių prieš filtravimą vidurkis yra 20.5%, o po filtravimo – 0,02%. Kitų defektų vidurkiai po filtravimo sumažėjo, tačiau nenukrito žemiau 2%. Iš to galima spręsti, kad filtras efektyviai pašalina fono triukšmus ir smulkius neatitikimus tarp šablono ir detalės, tuo pačiu nepašalindamas svarbių defektų.

Papildomai pastebima, kad spausdinimo ore (nr. 3, 8, 18) įverčiai kinta nuo 38% iki 500%, priklausomai nuo detalės ploto bei kaip ilgai defektas buvo nepastebėtas. Atsiklijavimo defektų (nr. 4, 9, 19) įverčiai viršija 100%, nes išskiriamos ir atsiklijavusios detalės ir trūkstamo plastiko zonos. Žemiausias defekto įvertis yra 2%, o aukščiausias kokybiškos detalės įvertis – 0,2%.

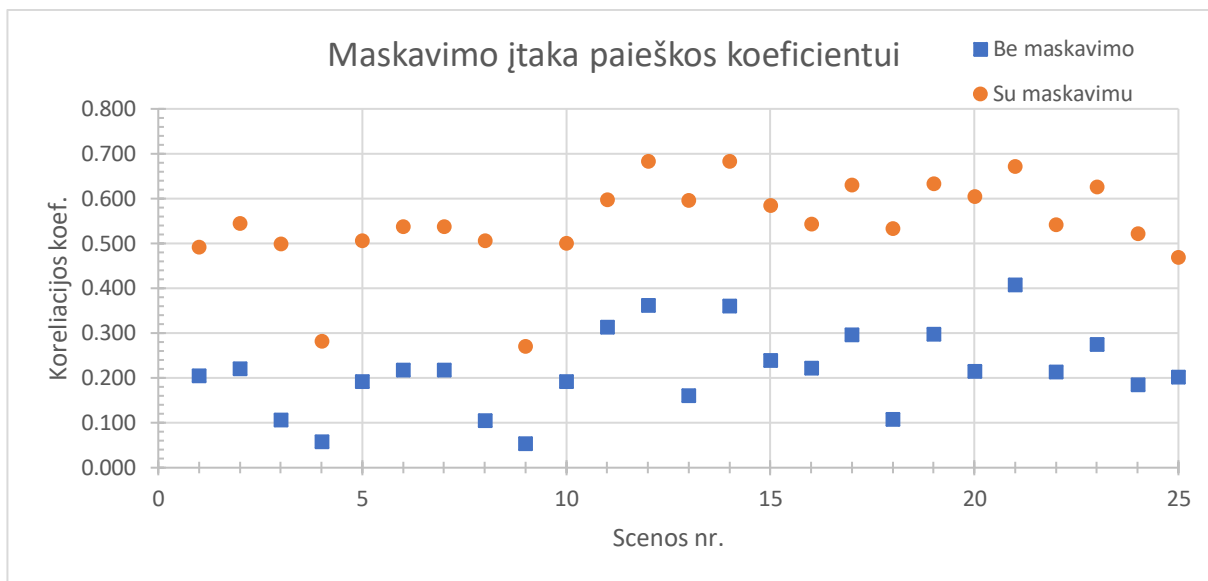
### 3.1.2. Defektų aptikimas naudojant šablono paieškos metodus

Siekiant išskirti spausdinamos detalės defektus kurie lieka nepastebėti po segmentavimo, pasinaudota 2.2.3 skyrelyje aprašyta metodika ir išskirtos detalės briaunos. Kadangi spausdinamos detalės pozicija ir posūkio kampas lengvai prognozuojami ruošiantis darbui, bei turi nesikeisti darbo metu, galima panaudoti šablono paieškos metodus šablono ir detalės pozicijoms palyginti.

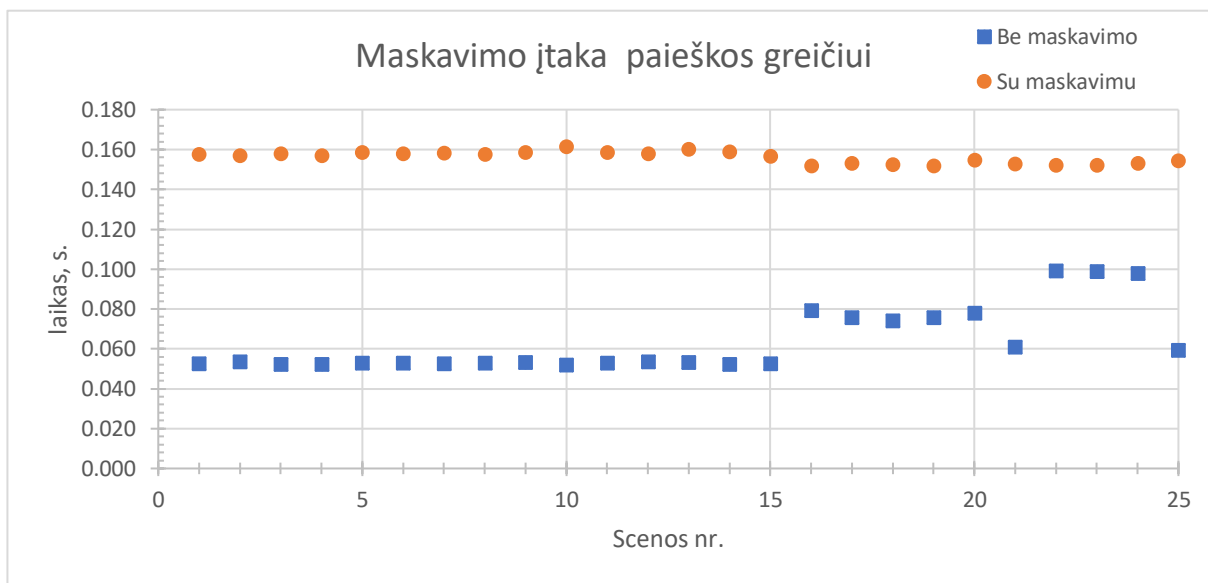
Iš bendro šablono vaizdo iškerpama tik dalis su norimomis palyginti briaunomis (atskirai tikrinama spausdinamas sluoksnis, bei visi buvę sluoksniai) ir slenkant šią iškarpa virš detalės briaunų vaizdo, kiekviename taške pasirinktu metodu skaičiuojamas panašumo koeficientas. Tuomet surandamos labiausiai šablono atitinkančio atskaitos taško koordinatės. Lyginant iškarpos koordinates tarp šablono ir detalės, pagal jų nuokrypį galima nustatyti sluoksnių pasislinkimo ar detalės atsiklijavimo defektus.

Verta pažymėti, kad atsiradus sluoksnių pasislinkimo defektui tikslesnis koreliacijos įvertis visada bus naujoje sluoksnio vietoje, kadangi net jei prieš tai buvęs sluoksnis yra identiškas, dalis jo yra uždengta nauju, pasislinkusiu sluoksniu. Papildomai siekiant sumažinti ankstesnių sluoksnių įtaką aptikimo tikslumui kai ieškoma tik naujausio sluoksnio, šablono iškarpos vaizdas užmaskuojamas ir ieškomi tik balti briaunų pikseliai. Tai reiškia, kad turint tik spausdinamo sluoksnio briaunų šablono, galima dideliu tikslumu rasti atitikmenį detalės vaizde, net jei jis apsuptas kitų briaunų ar spausdinimo

stalo kuriamų trikdžių. Palyginimas naudojant paiešką su maskavimu ir be jo, *TM\_CCORR\_NORMED* koreliacijos skaičiavimo metodu naudojant *Canny* briaunas, pavaizduotas 3.4 ir 3.5 paveiksluose, rezultatai pateikti antrame priede.



3.4 pav. Maskavimo įtaka paieškos koeficientui



3.5 pav. Maskavimo įtaka paieškos greičiui

Kaip matome iš rezultatų, užmaskuotos detalės paieška duoda tikslesnius koreliacijos įverčius. Vidutinis koreliacijos koeficientas nenaudojant maskavimo – 0,22, o su maskavimu – 0,54. Tačiau maskavimas prailgina skaičiavimo laiką vidutiniškai 2,5 karto (be maskavimo skaičiavimai vidutiniškai trunka – 0,064 s, su – 0,156 s).

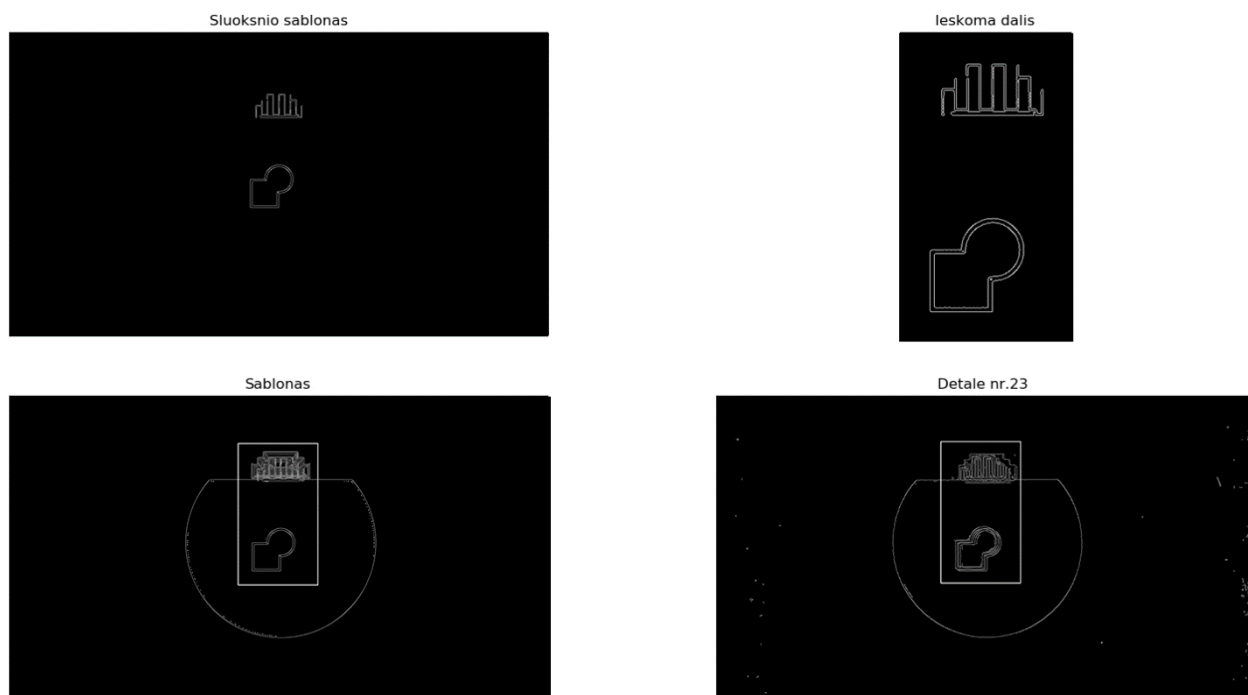
Naudojamoje *OpenCV* bibliotekos versijoje šablono maskavimą palaiko tik du panašumo skaičiavimo metodai – *TM\_SQDIFF* ir *TM\_CCORR\_NORMED*. Šiems metodams palyginti, buvo atlikti bandymai matuojant nuokrypį tarp dabartinio slauksnio šablono ir detalės koordinatų. Taip pat buvo palyginti *Canny* ir *Sobel* briaunų nustatymo algoritmai. Rezultatai pateikti septintoje lentelėje.

7 lentelė. Panašumo skaičiavimo ir briaunų nustatymo metodų palyginimas.

Scenos Numeris	<i>TM_CCORR_NORMED</i> metodas				<i>TM_SQDIFF</i> metodas			
	<i>Canny</i> briaunos		<i>Sobel</i> briaunos		<i>Canny</i> briaunos		<i>Sobel</i> briaunos	
	Nuokrypis, px	Koreliacijos koeficientas	Nuokrypis, px	Koreliacijos koeficientas	Nuokrypis, px	Kvadratų skirtumų suma	Nuokrypis, px	Kvadratų skirtumų suma
1	0.00	0.41	3.61	0.39	0.00	10098	3.61	17746
2	1.41	0.53	1.41	0.53	1.41	13735	1.41	24237
3	2.83	0.48	2.24	0.48	2.83	13561	2.24	23892
4	91.00	0.23	268.37	0.39	91.00	13732	91.00	24239
5	14.14	0.49	14.14	0.45	14.14	13733	14.14	24258
6	2.24	0.50	2.24	0.54	2.24	13745	2.24	24261
7	2.24	0.51	2.24	0.56	2.24	13561	2.24	24239
8	2.24	0.45	2.24	0.49	2.24	13648	2.24	23958
9	107.00	0.21	234.48	0.39	107.00	13746	110.00	24260
10	14.14	0.44	14.14	0.47	14.14	13758	14.14	24279
11	1.00	0.56	1.00	0.45	1.00	10086	1.00	18520
12	1.00	0.58	1.00	0.62	1.00	13735	1.00	24336
13	1.00	0.51	1.00	0.53	1.00	13469	1.00	23921
14	111.00	0.57	111.00	0.62	111.00	13737	111.00	24312
15	17.80	0.49	1.00	0.52	17.80	13715	1.00	24227
16	1.41	0.45	9.49	0.50	1.41	2174	9.00	4235
17	1.00	0.61	1.00	0.51	1.00	2174	1.00	4234
18	0.00	0.52	611.93	0.45	0.00	2232	5.10	4320
19	455.24	0.50	455.24	0.51	455.24	2174	455.24	4234
20	29.21	0.47	506.58	0.50	29.21	2174	33.30	4235
21	0.00	0.67	0.00	0.64	0.00	14588	0.00	25707
22	1.00	0.52	1.41	0.59	1.00	2606	1.41	2813
23	14.32	0.65	2.24	0.55	14.32	2606	2.24	2866
24	12.40	0.66	2.24	0.57	12.40	2606	2.24	2837
25	7.21	0.46	2.00	0.47	7.21	14591	1.41	25729

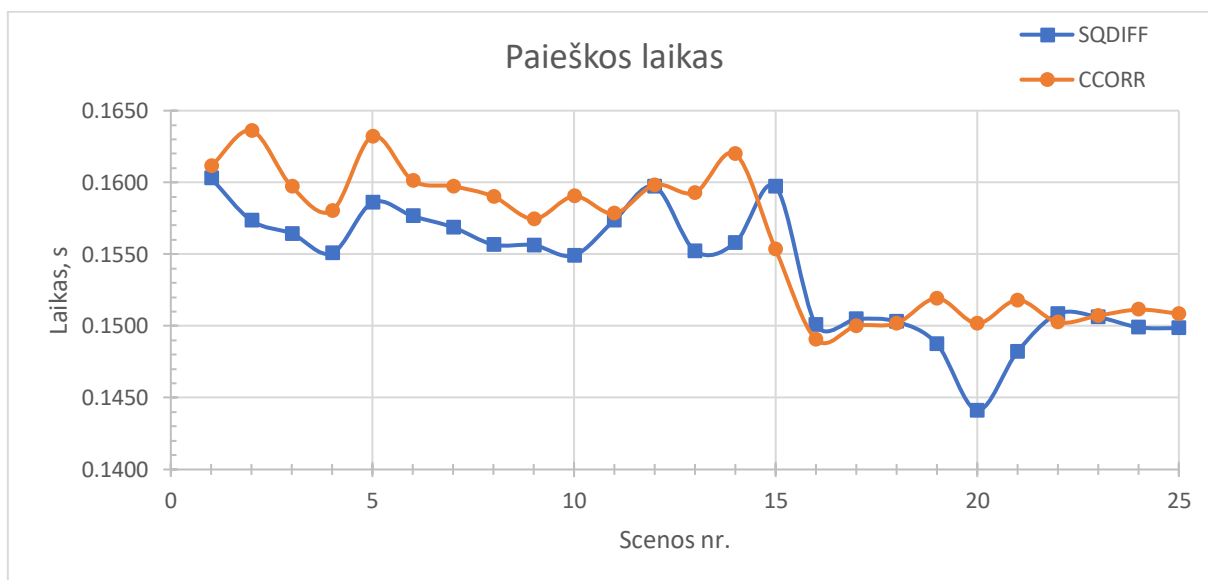
Lentelėje pilka spalva pažymėti akivaizdžiai neteisingai nustatyti detalės pozicijos nuokrypiai (vadinasi, neteisingai nustatytos spausdinamo sluoksnio koordinatės). Iš gautų rezultatų matyti, kad detalėse be defektų (nr. 1,2,6,7 ir t.t.) nuokrypis neviršija 2,3 px. Tačiau naudojant *Sobel* briaunas gautas nuokrypis nuotraukoje nr. 16 yra artimas 10 px, nors detalė yra be defektų. Vadinasi, koordinatės aptiktos neteisingai ir toliau bus naudojamas *Canny* briaunų nustatymo metodas. Sluoksnių pasislinkimo atvejais (nr. 5, 10, 15, 20, 23 ir 24) gautas nuokrypis didesnis nei 10 px, o detalės atsiklijavimo atvejai (nr. 4, 9, 14 ir 19) taip pat aptinkami teisingai. Tačiau spausdinimo ore defektai (nr. 3, 7, 13 ir 17) ženkliai neįtakojo nei rasto nuokrypio, nei koreliacijos koeficiento. Verta paminėti, kad sluoksnių pasislinkimo defektai buvo atpažinti net ir tais atvejais, kai to nepavyko

atlikti segmentavimo palyginimo metu (nr. 23 ir 24). Aptikimo pavyzdys su pažymėta didžiausios koreliacijos zona - 3.6 paveiksle.



3.6 pav. Sluoksnių pasislinkimo aptikimas koreliacijos metodu

Kaip matyti iš rezultatų, naudojant *Canny* briaunų vaizdus abejais paieškos metodais gauti vienodi pozicijų nuokrypiai ir nepasitaikė su didele paklaida aptiktų koordinatčių. Tam, kad išsirinkti toliau naudojamą paieškos metodą, išmatuotas paieškos laikas. Rezultatai pateikti trečiame priede ir 3.7 paveiksle.



3.7 pav. Paieškos laikas skirtingais panašumo skaičiavimo metodais

Vidutinis paieškos laikas naudojant *TM\_SQDIFF* metodą – 0,1540s, o *TM\_CCORR\_NORMED* – 0,1561s. Atsižvelgiant į tai, kad paieška *TM\_SQDIFF* trunka vidutiniškai 2,1ms greičiau, bei kad

naudojant *Sobel* briaunas buvo gauta mažiau klaidingų aptikimų nei naudojant *Sobel* briaunas *TM\_CCORR\_NORMED* metodu, defektų aptikimo sistemai pasirinkta naudoti *TM\_SQDIFF* metodą su *Canny* briaunomis.

### 3.2. Sudarytos defektų aptikimo sistemos veikimo principas

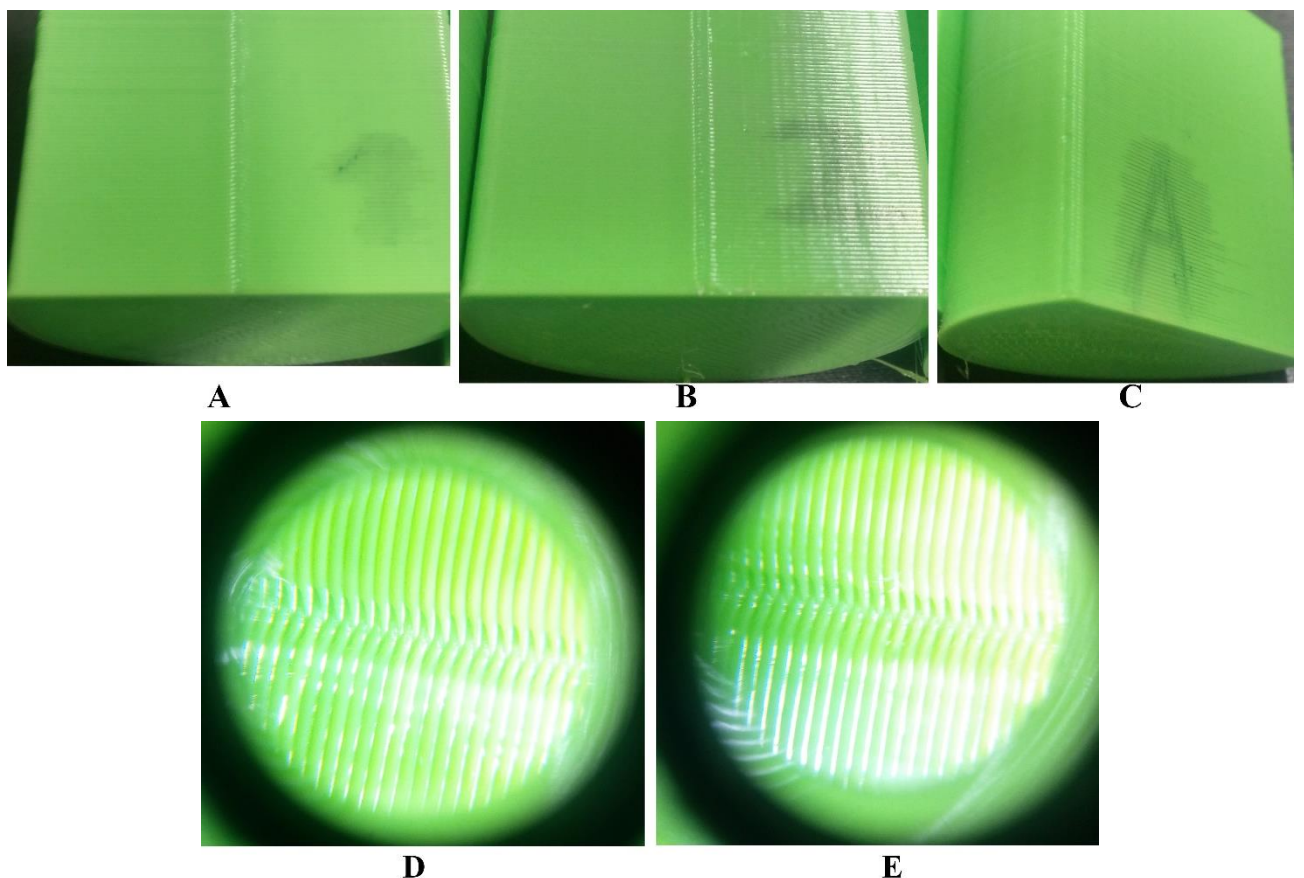
Remiantis atliktų bandymų rezultatais, nuspręsta naudoti spalvinį segmentavimą spausdinimo ore ir nuvirtusių prilaikančiųjų konstrukcijų aptikimui, bei *Canny* briaunų nustatymą kartu su *TM\_SQDIFF* paieškos metodu detalės atsiklijavimui ir sluoksnių pasislinkimui nustatyti. Programos kodas ketvirtame priede, o kodo paaiškinimas ir bendras sistemos veikimo aprašymas toliau:

1. Ruošiant spausdinamos detalės kodą spausdinimui, pažymimi sluoksniai kuriuose norima atlikti patikrą. Sluoksnių vaizdai išsaugomi numeruojant iš eilės, pradedant vienetu. Remiantis naudojamo plastiko spalva, išsaugomi parametrai reikalingi spalvinei segmentacijai atlikti. Papildomai pažymima ar plastiko ir spausdinimo stalo paviršiaus atspalviai yra panašūs.
2. Spausdinant detalę, pasiekus pažymėtą patikrai sluoksnį, darbas stabdomas ir spausdinimo galvutė patraukiama į šalį. Spausdintuvo valdiklis siunčia signalą patikrai pradėti.
3. Užfiksuojama spausdinamos detalės nuotrauka. Pagal sluoksnio numerį parenkamas tinkamas šablonas, nuskaitomi naudojamo plastiko spalvos parametrai. *Tyrimo metu, nuotrauka užkraunama iš atminties.*
4. Iš šablono vaizdo, segmentavimo būdu išskiriamos zonos žyminčios dabartinį sluoksnį ir bendrą detalės vaizdą. Maži tarpeliai pašalinami panaudojus *closing* filtrą. Spausdinamos detalės vaizdas filtruojamas *bilateral* filtru.
5. Atliekama segmentavimu pagrįsta defektų paieška:
  - a. Spausdinamos detalės vaizdas pagal nuskaitytus parametrus segmentuojamas HSV spalvinėje paletėje.
  - b. Šablono zona žyminti bendrą detalės vaizdą ir spausdinamos detalės zona išskirta segmentavimo būdu palyginamos tarpusavyje, panaudojant *bitwise\_xor* funkciją. Gautas išskirto defekto vaizdas filtruojamas *opening* filtru ir suskaičiuojami visi balti pikseliai plotui nustatyti.
  - c. Defekto plotas lyginamas su detalės šablono plotu ir pagal gautą santykį nusprendžiama apie defekto aptikimą. Jei santykis viršija ir 0,4% ir jei plastiko ir spausdinimo stalo atspalviai skirtingi, signalizuojama apie defekto atsiradimą ir stabdomas spausdinimas.
6. Atliekama briaunų palyginimu paremta defektų paieška:
  - a. Ketvirtame etape išskirtai dabartinio sluoksnio zonai pritaikomas *Canny* briaunų nustatymo algoritmas.
  - b. Šablono briaunų vaizdas apkarpomus paliekant nustatyto dydžio zoną aplink išskirtas briaunas. Nustatomos šio vaizdo atskaitos taško koordinatės, kurios bus naudojamos palyginimui.
  - c. Spausdinamos detalės vaizdui po filtravimo pritaikomas *Canny* briaunų nustatymo algoritmas. Paliekamos tik briaunos patenkančios į segmentavimo būdu išskirtą zoną. Taip pašalinama didelė dalis stalo rašto kuriamų briaunų, todėl galima naudoti jautresnį briaunų nustatymo algoritmą.

- d. Apkirptas šablono briaunų vaizdas lyginamas su detalės briaunų vaizdu *TM\_SQDIFF* būdu ir nustatomas taško su geriausiu atitikmeniu koordinatės.
  - e. Skaičiuojamas atstumas tarp šablono iškarpos ir surasto atitinkamo detalės taško koordinatė. Jam viršijus 2,5 pikselio signalizuojama apie defekto atsiradimą ir stabdomas spausdinimas.
7. Jei buvo aptiktas defektas – spausdintuvo valdikliui perduodamas pauzės signalas, palaikoma spausdinimo stalo temperatūra siekiant išvengti galimo detalės atsiklijavimo, o spausdinimo galvutės kaitinimas išjungiamas. Operatoriui nusprendus apie klaidingai aptiktą defektą galima darbą pratęsti.
  8. Jei defektas neaptiktas, spausdintuvo valdikliui perduodamas darbo pratęsimo signalas, spausdinimo galvutė automatiškai nuvaloma šepetėliu ir grąžinama į poziciją prieš patikrą. Spausdinimas tęsiamas iki sekančio patikros signalo.

### 3.3. Defektų aptikimo sistemos įtaka spausdinamos detalės išvaizdai

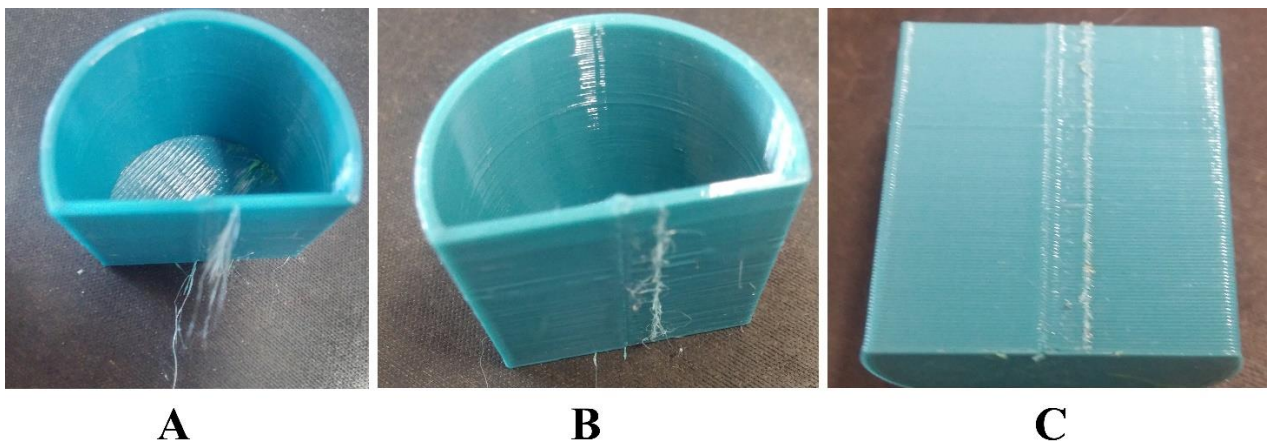
Siekiant patikrinti ar nuolatinis darbo stabdymas detalės fotografavimui nesukelia papildomų kosmetinių defektų, buvo paruoštas modelis su sulygiuota ir matomoje vietoje esančia sluoksnių keitimo vieta. Viena detalė buvo spausdinama įprastai, nestabdant darbo. Kita – kiekvieno sluoksnių keitimo metu patraukiant spausdinimo galvutę į šalį, tarsi būtų atliekamas defektų aptikimas. Išbandytos dvi plastiko rūšys – PLA ir PETG. Rezultatai pateikti 3.8 ir 3.9 paveiksluose.



3.8 pav. Darbo sustabdymo įtaka detalių iš PLA plastiko išvaizdai



PLA plastikas yra spausdinamas žemoje temperatūroje (~190°C) ir greitai kietėja, todėl lengviau kontroliuoti staigius spausdinimo greičio pokyčius, darbo stabdymą. Detalių spausdintų nesustojant (3.8 pav. A ir D) ir detalių spausdintų kiekviename sluoksnyje atliekant patikrą (3.8 pav. B ir E) kokybė pastebimai nesiskiria, o sluoksnių keitimo vieta gali būti užmaskuojama detalės kampe (3.8 pav C).



3.9 pav. Darbo sustabdymo įtaka detalių iš PETG plastiko išvaizdai

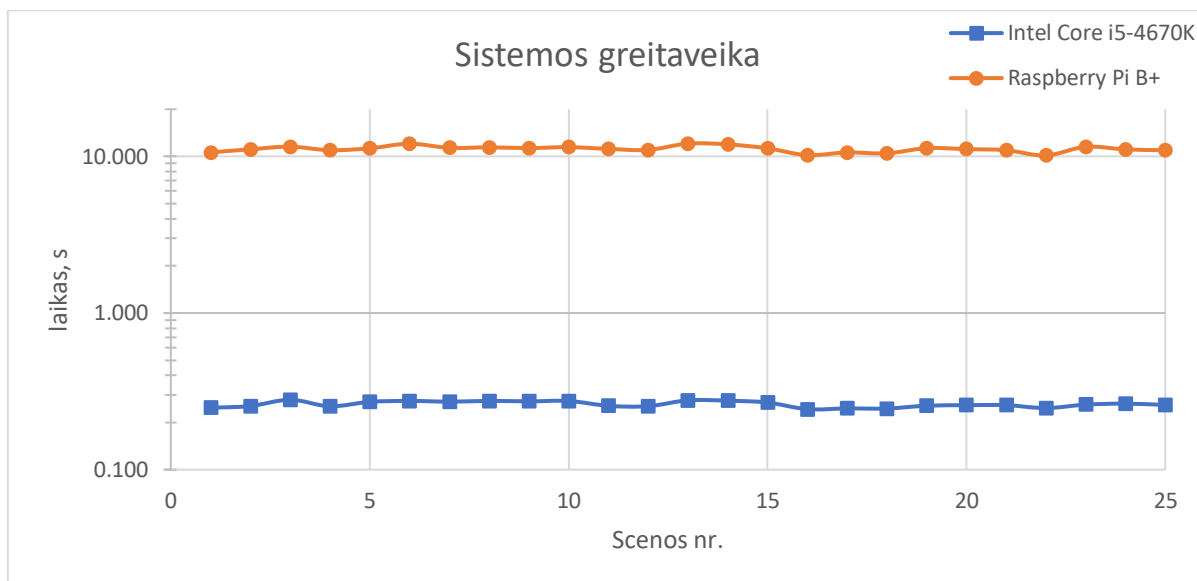
PETG plastikas yra klampesnis, spausdinamas aukštesnėje temperatūroje (240°C) ir todėl sunkiau suvaldomi staigūs spausdinimo greičio pokyčiai. Staiga nutraukus plastiko tiekimą, dėl susidariusio spaudimo spausdinimo galvutėje ištrykšta šiek tiek plastiko. Traukiant spausdinimo galvutę į šalį šis plastikas prilimpa prie spausdinamos detalės ir sukuria plaukelius primenantį kosmetinį defektą. Detalės apdirbimo po spausdinimo metu šie defektai nesunkiai pašalinami, tačiau norint padidinti defektų aptikimo sistemos tikslumą reikia šiuos defektus pašalinti dar darbo eigoje. Tai pasiekama padidinus akceleraciją ir laisvų judesių (*angl. travel*) greitį, todėl ženkliai sumažinamas ištryškusio plastiko kiekis kol spausdinimo galvutė nepatraukta nuo detalės. 3.9 pav. pavaizduotos detalės spausdintos PETG plastiku, kai akceleracijos dydis 500mm/s<sup>2</sup> (A), kai akceleracijos dydis 1500mm/s<sup>2</sup> (B) ir po plaukelių pašalinimo (C).

### 3.4. Defektų aptikimo sistemos veikimo greitis

Sistemos greitaveika buvo išbandyta asmeniniame kompiuteryje su *Intel Core i5-4670K* procesoriumi ir *Raspberry Pi B+* vienos plokštės kompiuteryje. Eksperimentas kartotas tris kartus ir apskaičiuotas vidurkis. Rezultatai pateikti penktame priede ir 3.10 paveiksle, o naudojama eksperimento įranga – aštuntoje lentelėje.

8 lentelė. Bandomų kompiuterių konfigūracijos

Sistema	Asmeninis kompiuteris	<i>Raspberry Pi B+</i>
Procesorius	<i>Intel Core i5-4670K</i> , 3,8GHz	ARM1176JZF-S, 700MHz
Operatyvinės atminties kiekis	16 GB	512 MB
Operacinė sistema	Windows 10, 64 bitų	<i>Raspberry Pi OS 5.4</i>
<i>Python</i> versija	2.7.18	2.7.18
<i>OpenCV</i> versija	4.2.0	4.2.0



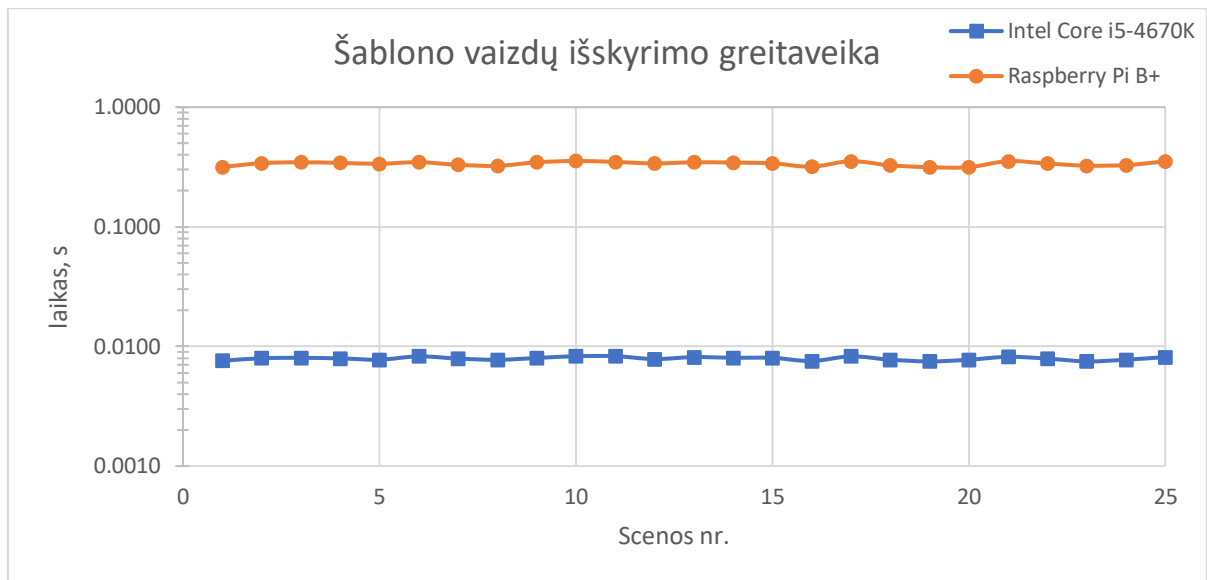
3.10 pav. Defektų aptikimo sistemos greitimeika

Vidutinis sistemos veikiančio asmeniniame kompiuteryje patikros greitis – 0,262s, o *Raspberry Pi B+* – 11,145s. Vadinasi, sistema asmeniniame kompiuteryje veikia 42,5 karto greičiau.

### 3.4.1. Šablono vaizdų išskyrimo greitis

Kadangi šio darbo metu naudojami šablonų vaizdai yra ruošiami rankiniu būdu, buvo nuspręsta apjungti dabar spausdinamo ir jau atspausdintų sluoksnių vaizdus į vieną. Taip dvigubai sumažinama darbo ruošiant šablonus, tačiau sugaištama laiko vaizdams atskirti vykdant programą. Šablono modeliui sudarytam iš dviejų spalvų, pritaikomas segmentavimo pagal ryškumą algoritmas vaizdams atskirti. Po to, dabartinio sluoksnio vaizdas apdirbamas *Canny* briaunų nustatymo algoritmu. Norint padidinti sistemos greitimeiką, šie veiksmai galėtų būti atlikti dar ruošiant šabloną.

Siekiant įvertinti šablono vaizdų išskyrimo įtaką galutinei sistemos greitimeikai, buvo išmatuotas šablono segmentavimo ir briaunų išskyrimo greitis ir rezultatai pateikti šeštame priede ir 3.11 paveiksle.



3.11 pav. Šablono vaizdų išskyrimo greitimeika

Vidutinis šablono vaizdų išskyrimo asmeniniame kompiuteryje greitis – 7,9ms, o *Raspberry Pi B+* – 337ms. Vadinas, šablono vaizdų apdirbimui sugaištama tik 3% visos defektų aptikimo sistemos veikimo laiko ir išankstinis vaizdų išskyrimas neturės ženklios įtakos bendrai sistemos greitimeikai.

### 3.4.2. Darbo stabdymo ir pratęsimo trukmė, galimos optimizacijos

*Prusa Face Shield* modelis tyrime naudojamu spausdintuvu atspausdinamas per 1 valandą 34 minutes ir 23 sekundes. Tas pats modelis spausdinamas sustojant kas antrame sluoksnyje nuotraukai užfiksuoti (viso 33 sustojimai) buvo atspausdintas per 1 valandą 41 minutę ir 47 sekundes. Vadinas, darbo sustabdymas, nuotraukos fiksavimas ir darbo pratęsimas trunka apytiksliai 13,5 sekundės. Atsižvelgiant į tai, kad pratęsiant darbą spausdinimo galvutė papildomai sugaišta laiko ištryškusio plastiko nuvalymui, nuo nuotraukos užfiksavimo iki sekančio sluoksnio spausdinimo pradžios praeina ne mažiau nei pusė viso sugaišto laiko, tai yra – ne mažiau nei 6,75 sekundės.

Norint kiek galima mažiau įtakoti spausdinimo laiką, defektų patikrą galima atlikti kol spausdinimo galvutė juda tarp fotografavimo ir spausdinimo pozicijų. Vadinas, patikrą reiktų atlikti greičiau nei per 6,75 sekundės. Nepriklausomi bandymai [27] parodė, kad vieno branduolio sintetinių skaičiavimų greitis yra apie 1,78 karto didesnis *Raspberry Pi 2 B* kompiuteryje, lyginant su *Raspberry Pi B+*. Vadinas, defektų patikra *Raspberry Pi 2 B* kompiuteryje užtruktų apytiksliai 6,26 sekundės ir būtų atlikta dar prieš spausdinimui prasitęsiant.

### 3.5. Defektų aptikimo sistemos efektyvumo tyrimas

Naudojami algoritmai ir jų parametrai parinkti 3.1 skyriuje, naudojant mišrų rinkinį kokybiškų ir nekokybiškų detalių vaizdų. Siekiant patikrinti sistemos efektyvumą išskiriant skirtingus defektų tipus, jos veikimas išbandytas su anksčiau nenaudotomis nuotraukomis.

Atlikti trys eksperimentai – defektų aptikimo žalios spalvos detalėse, defektų aptikimo pilkos spalvos detalėse ir defektų aptikimo kai spausdinamos dvi identiškios detalės vienu metu. Eksperimentais siekiama iširti kiekvienos rūšies defektų aptikimo kokybę ir įsitikinti, kad defektai neaptinkami

kokybiškose detalėse. Eksperimentuose naudotos nuotraukos pateiktos septintame, aštuntame ir devintame priede.

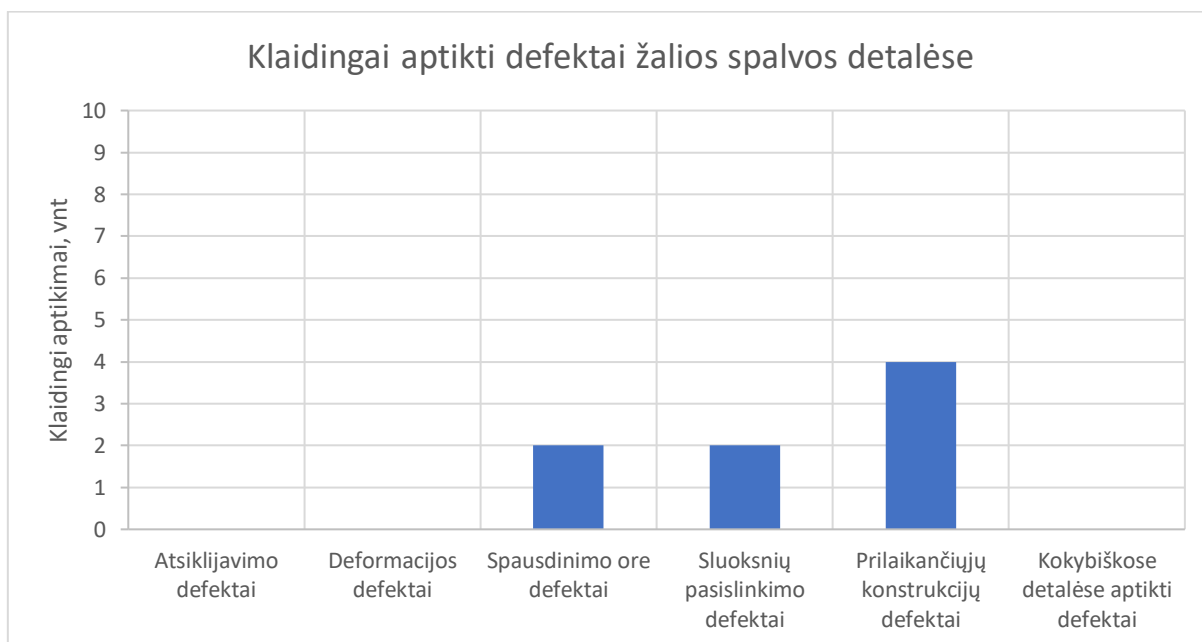
### 3.5.1. Defektų aptikimo žalios spalvos detalėse eksperimentas

Šiame eksperimente panaudotos fotografijos suskirstytos po dešimt vienetų kiekvienam defektų tipui, kartu su dešimt kokybiškų detalių be defektų fotografijų. Detalės spausdintos žalios spalvos PLA plastiku. Kiekviena defektų kategorija, išskyrus prilaikančiųjų konstrukcijų defektus, sudaryta iš penkių šio darbo autoriaus sukurtu pusskritulio modelio detalių ir penkių *Prusa Face Shield* modelio detalių. Prilaikančiųjų konstrukcijų defektai gali atsirasti tik pusskritulio formos detalėje, todėl ši kategorija sudaryta tik iš šio modelio detalių fotografijų. Fotografijos pateikiamos sukurtai defektų aptikimo sistemai ir pažymimas defekto įvertis gautas segmentavimo būdu, bei nuokrypio vertė gaunama šablono paieškos metodu. Pagal 3.2 skyrelyje aprašytas ribines vertes nustatoma ar aptiktas defektas. Devintoje lentelėje pateikti gauti rezultatai, kur pilka spalva pažymėtos fotografijos su klaidingai aptiktais (ar neaptiktais) defektais.

9 lentelė. Defektų aptikimo žalios spalvos detalėse rezultatai

Defektas		Atsiklijavimo	Deformacijos	Spausdinimo ore	Pasislinkimo	Prilaikančiųjų konstrukcijų	Kokybiškos detalės
Scenos nr.							
1	Įvertis, %	53.46	14.96	0.05	0.00	0.00	0.00
	Nuokrypis, px	157.28	13.04	1.00	6.40	1.00	2.00
2	Įvertis, %	69.08	36.32	0.42	0.00	0.42	0.24
	Nuokrypis, px	148.82	16.28	1.41	5.00	1.41	1.00
3	Įvertis, %	68.60	27.66	3.05	0.00	0.42	0.24
	Nuokrypis, px	163.77	111.00	1.00	6.08	1.41	0.00
4	Įvertis, %	127.57	22.54	0.42	0.00	0.56	0.00
	Nuokrypis, px	296.50	128.72	1.41	1.00	1.41	1.41
5	Įvertis, %	168.31	33.32	0.00	0.00	0.56	0.00
	Nuokrypis, px	519.20	112.97	1.41	1.41	1.41	1.41
6	Įvertis, %	207.84	119.42	144.51	51.30	0.00	0.00
	Nuokrypis, px	136.30	45.04	2.24	13.89	1.00	0.00
7	Įvertis, %	168.55	160.65	142.48	35.83	0.00	0.00
	Nuokrypis, px	61.29	139.70	2.24	13.00	2.00	0.00
8	Įvertis, %	180.79	102.21	66.27	62.88	0.58	0.13
	Nuokrypis, px	84.72	5.00	1.41	18.38	1.00	0.00
9	Įvertis, %	209.30	135.58	145.09	43.57	1.26	0.10
	Nuokrypis, px	179.00	154.00	1.41	13.89	1.00	0.00
10	Įvertis, %	174.45	154.27	85.14	28.66	0.20	0.13
	Nuokrypis, px	16.00	15.03	2.24	9.43	1.41	0.00

Kaip matyti iš rezultatų, atsiklijavimo ir deformacijos defektai aptinkami 100% tikslumu, viršijamas ne tik defekto įvertis, bet ir nuokrypio limitas. Spausdinimo ore defektai aptinkami tik pagal segmentavimo būdu gaunamą įvertį, todėl kai defektas lokalizuotas detalės ribose įvertis neviršija ribos ir defektas lieka neaptiktas. Autoriaus patirtis ruošiant defektų fotografijas rodo, kad traukiant spausdinimo galvutę į šalį patikrai atlikti, spausdinimo ore defektas dažnai būna nutempiamas už detalės ribų ir gali būti atpažintas. Sluoksnių pasislinkimo defektas patikimai aptinkamas pagal nuokrypio dydį, tačiau jei defektas įtakoja plačiausius detalės sluoksnius (nr.6-10), gali būti aptinkamas ir pagal defekto įvertį. Dėl neidealios detalės ir šablono lygiavimo, bei galimai nepastovaus briaunų radimo, nuokrypiai mažesni nei 2,5 pikselio nežymimi kaip defektai. Todėl jei sluoksniai pasislinkę nežymiai, defektas lieka neaptiktas. Prilaikančiųjų konstrukcijų defektai aptinkami panašiai kaip ir spausdinimo ore – jei defektas neišeina už numatytų šablono ribų, jis lieka nepastebėtas. Visos kokybiškos detalės pažymėtos kaip be defektų, o didžiausias aptiktas padėties nuokrypis nuo šablono – 2 pikseliai. Klaidingai aptiktų defektų skaičius pagal defektų tipą pavaizduotas 3.12 paveiksle.



3.12 pav. Klaidingai aptikti defektai žalios spalvos detalėse

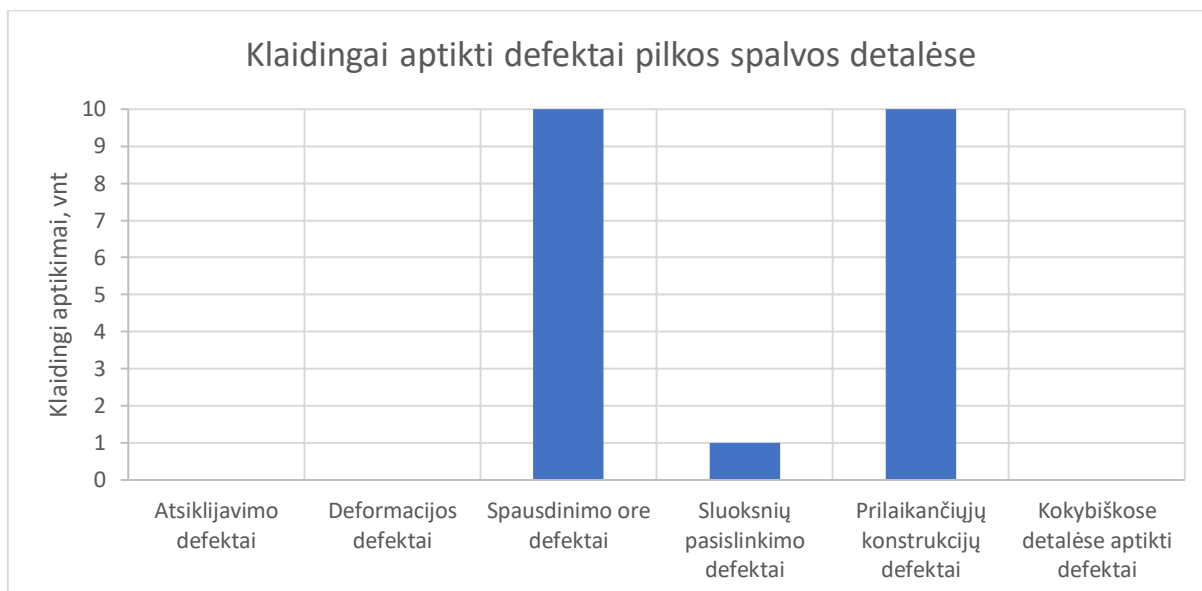
### 3.5.2. Defektų aptikimo pilkos spalvos detalėse eksperimentas

Šiame eksperimente panaudotos fotografijos yra sugrupuotos analogiškai prieš tai buvusiam, defektų aptikimo žalios spalvos detalėse eksperimentui, tačiau spausdintos spalva panašia į spausdinimo stalo (pilka). Sistemos parametruose pažymima, kad stalo ir detalės atspalviai panašūs ir defektai vertinami tik pagal pozicijos nuokrypį. Dešimtoje lentelėje pateikti gauti rezultatai, kur pilka spalva pažymėtos fotografijos su klaidingai aptiktais (ar neaptiktais) defektais.

10 lentelė. Defektų aptikimo pilkos spalvos detalėse rezultatai

Defektas		Atsiklijavimo	Deformacijos	Spausdinimo ore	Pasislinkimo	Prilaikančiųjų konstrukcijų	Kokybiškos detalės
Scenos nr.							
1	Įvertis, %	48.16	40.50	4.56	3.97	0.27	0.30
	Nuokrypis, px	160.90	43.00	1.00	5.00	1.00	1.00
2	Įvertis, %	55.13	42.66	6.32	3.88	4.60	0.49
	Nuokrypis, px	151.65	42.00	0.00	14.87	1.00	1.00
3	Įvertis, %	120.85	40.50	5.77	3.93	0.27	0.88
	Nuokrypis, px	468.03	111.02	1.41	10.00	1.41	1.00
4	Įvertis, %	47.95	46.69	9.87	3.97	0.27	1.99
	Nuokrypis, px	186.34	21.93	1.00	6.32	2.00	1.00
5	Įvertis, %	12.93	41.69	7.51	3.92	4.21	31.04
	Nuokrypis, px	314.77	103.17	1.41	10.30	1.00	1.00
6	Įvertis, %	409.23	397.90	135.34	55.40	0.27	86.21
	Nuokrypis, px	57.01	188.00	1.00	14.87	1.00	1.00
7	Įvertis, %	430.59	398.45	132.43	48.24	4.62	61.71
	Nuokrypis, px	196.00	10.00	2.00	18.38	1.41	0.00
8	Įvertis, %	405.99	394.83	150.45	62.37	0.27	44.78
	Nuokrypis, px	35.01	14.87	1.00	17.03	1.00	1.00
9	Įvertis, %	430.07	393.80	147.34	46.51	5.01	31.86
	Nuokrypis, px	98.01	11.18	1.00	1.00	1.00	1.00
10	Įvertis, %	419.32	399.83	106.62	35.94	0.27	30.50
	Nuokrypis, px	27.46	91.20	1.00	11.05	1.00	1.00

Kaip matyti iš rezultatų, atsiklijavimo ir deformacijos defektai aptinkami 100% tikslumu net ir nenaudojant segmentavimo būdu gauto įverčio. Tačiau segmentavimo įverčio atsisakymas taip pat reiškia, kad spausdinimo ore ir prilaikančiųjų konstrukcijų defektai lieka neaptikti. Vienas iš sluoksnių pasislinkimo defektų (nr. 9) lieka neaptiktas ir detalės nuokrypis apskaičiuotas tik 1 pikselio dydžio. Nuotraukoje vaizduojamas pasislinkimas yra apie 2mm dydžio, kas atitinka apie 16 pikselių, todėl daroma išvada, kad didesnę panašumą šablonui turėjo prieš tai esantis sluoksnis ir buvo klaidingai aptikta naujo, pasislinkusio sluoksnio vieta. Visos eksperimente naudotos kokybiškos detalės teisingai atpažintos kaip be defektų, o didžiausias pasitaikęs padėties nuokrypis yra vieno pikselio dydžio. Klaidingai aptiktų defektų skaičius pagal defektų tipą pavaizduotas 3.13 paveiksle.



3.13 pav. Klaidingai aptikti defektai pilkos spalvos detalėse

### 3.5.3. Defektų aptikimo kai spausdinamos dvi detalės eksperimentas

Eksperimentas atliktas siekiant išsiaiškinti sistemos reakciją kai spausdinamos dvi identiškos detalės vienu metu, ir vienoje iš jų pasitaiko defektas. Naudojamas 50% sumažintas pusskritulio formos modelis, detalės spausdinamos žalios spalvos PLA plastikui. Atsiklijavimo defekto nuotraukose viena detalė yra atsiklijavusi, o kita likusi originalioje pozicijoje. Sluoksnių pasislinkimo defektų nuotraukas sudarytos iš penkių (nr.6-10) kų tik įvykusio pasislinkimo vaizdų, kai įtakojama tik viena detalė ir penkių (nr. 1-5) nuotraukų užfiksuotų sekančiame sluoksnyje, kai defekto paveiktos abi detalės. Dešimtoje lentelėje pateikti gauti rezultatai, kur pilka spalva pažymėtos fotografijos su klaidingai aptiktais (ar neaptiktais) defektais.

11 lentelė. Defektų aptikimo aptikimo kai spausdinamos dvi detalės rezultatai

Scenos nr. \ Defektas		Atsiklijavimo	Pasislinkimo	Kokybiškos detalės
1	Įvertis, %	25.02	0.00	0.00
	Nuokrypis, px	2.00	5.83	1.00
2	Įvertis, %	73.18	0.00	0.00
	Nuokrypis, px	2.00	3.61	2.00
3	Įvertis, %	98.16	0.00	0.00
	Nuokrypis, px	2.00	5.83	2.00
4	Įvertis, %	97.51	0.00	0.00
	Nuokrypis, px	2.24	6.40	1.41
5	Įvertis, %	84.16	0.00	0.00
	Nuokrypis, px	2.00	9.43	1.00
6	Įvertis, %	97.34	0.00	0.00
	Nuokrypis, px	2.00	2.24	0.00

7	Įvertis, %	82.97	0.00	0.00
	Nuokrypis, px	2.24	1.00	1.00
8	Įvertis, %	55.69	0.00	0.00
	Nuokrypis, px	2.00	1.00	2.00
9	Įvertis, %	94.89	0.00	0.00
	Nuokrypis, px	2.00	1.00	1.41
10	Įvertis, %	92.48	0.00	0.00
	Nuokrypis, px	2.00	2.00	1.00

Kaip matyti iš rezultatų, visi atsiklijavimo defektai aptinkami teisingai. Defekto įverčiai gauti segmentavimo būdu neviršija 100%, nes viena detalė likusi savo vietoje ir taip padidina bendrą detalės plotą, tuo pačiu mažinant defekto plotą. Kokybiškos detalės taip pat atpažintos tinkamai, ir didžiausias užfiksuotas nuokrypis nuo šablono yra dviejų pikselių dydžio. Kai sluoksnių pasislinkimo defekto paveikta tik viena detalė (scenos nr. 6-10), nuokrypio vertės neviršijo nustatytos 2,5 pikselio ribos. Tai įvyko dėl to, nes kokybiškos detalės dabartinis sluoksnis kartu su pažeistos detalės prieš tai buvusiu sluoksniu buvo panašesnis į šabloną. Tačiau atvejai kai sluoksnių pasislinkimo paveikta tik viena detalė gali būti užfiksuojami tik iš karto po defekto atsiradimo. Spausdinant sekantį sluoksnį defektas atsiranda ant abiejų detalių ir tampa aptinkamas (scenos nr. 1-5).



## Rezultatai ir Išvados

1. Remiantis literatūroje pateiktomis rekomendacijomis, sukonstruotas defektų aptikimo sistemos prototipas sudarytas iš vaizdo kameros sumontuotos virš stalo centro, darbo vietos apšvietimo su šviesos išsklaidymo elementais ir gaubto skirto pašalinio apšvietimo įtakai sumažinti.
2. Atlikus literatūros apžvalgą, defektų aptikimui pasirinkta naudoti paveikslų atimties ir šablono paieškos metodus.
3. Naudojant segmentavimo pagal spalvą ir paveikslų atimties metodus apskaičiuojamas defekto ir šablono plotų santykis išreikštas procentais. Remiantis bandymų rezultatais parinkta defekto atsiradimo riba – 0,4%. Naudojant *Canny* briaunų nustatymą ir kvadratinių skirtumų suma paremtą šablono paieškos metodą, palyginamos naujausio šablono sluoksnio ir jo atitiktens spausdinamoje detalėje koordinatės. Jei koordinacių nuokrypis viršija 2,5 pikselio, nustatomas defektas.
4. Didinant spausdinimo galvutės akceleraciją galima išvengti patikros sukeliamų, kosmetinių spausdinamos detalės defektų atsiradimo. Naudojant PETG plastiką, pažeidimai tampa nepastebimi kai spausdinimo galvutės akceleracija viršija 1500mm/s<sup>2</sup>.
5. Vidutinis defektų aptikimo sistemos veikiančios asmeniniame kompiuteryje su Intel i5-4670K procesoriumi skaičiavimų greitis – 0,262s, o *Raspberry Pi B+* kompiuteryje – 11,145s. Šablono vaizdo segmentavimas ir briaunų nustatymas užima tik 3% bendro sistemos skaičiavimo laiko, todėl išankstinis šablono apdirbimas ženkliai nepagreitins sistemos darbo. Spausdinimo galvutei patraukti nuo spausdinamos detalės ir sugrįžti spausdinimui pratęsti užtrunka 13,5 sekundės. Norint patikrai sugaišti kuo mažiau laiko, rekomenduojama defektų aptikimą atlikti kol spausdinimo galvutė grįžta darbui pratęsti. Vadinasi, skaičiavimai turi būti atlikti greičiau nei per 6,75 sekundės. Silpniausias rekomenduojamas kompiuteris atitinkantis šiuos reikalavimus yra *Raspberry Pi 2*.
6. Defektų aptikimo sistemos patikimumui įvertinti atlikti trys eksperimentai:
  - 6.1. Kai spausdinimui naudojamo plastiko spalva skiriasi nuo spausdinimo stalo, defektų aptikimo sistemos efektyvumas – 86,7%. Visos detalės be ieškomų defektų identifikuotos teisingai, bei atsiklijavimo ir deformacijos defektai aptikti 100% atvejų. Spausdinimo ore ir prilaikančiųjų konstrukcijų defektai aptinkami tik jei defektas išeina už detalės ribų. Bandyne naudotose nuotraukose spausdinimo ore defektai aptikti 80% atvejų, o prilaikančiųjų konstrukcijų – 60%. Sluoksnių pasislinkimo defektai atpažinti 80% tikslumu.
  - 6.2. Kai spausdinimui naudojamo plastiko ir spausdinimo stalo atspalviai panašūs, spalvinis segmentavimas nebeveikia tinkamai ir sistemos efektyvumas krenta iki 65%. Visos detalės be ieškomų defektų identifikuotos teisingai, bei atsiklijavimo ir deformacijos defektai aptikti 100% atvejų. Sluoksnių pasislinkimo defektai atpažinti 90% atvejų. Tačiau spausdinimo ore ir prilaikančiųjų konstrukcijų defektai neaptikti nei karto (0%). Dažnai spausdinant tokia medžiagos spalva, rekomenduojama spausdinimo stalą padengti kitokios spalvos danga.
  - 6.3. Jei spausdinamos dvi detalės vienu metu, sluoksnių pasislinkimo defektai aptinkami ne iš karto, o tik kai paveiktos abi detalės. Vienos detalės atsiklijavimas gali būti pastebėtas tik kai naudojamo plastiko spalva skiriasi nuo spausdinimo stalo.

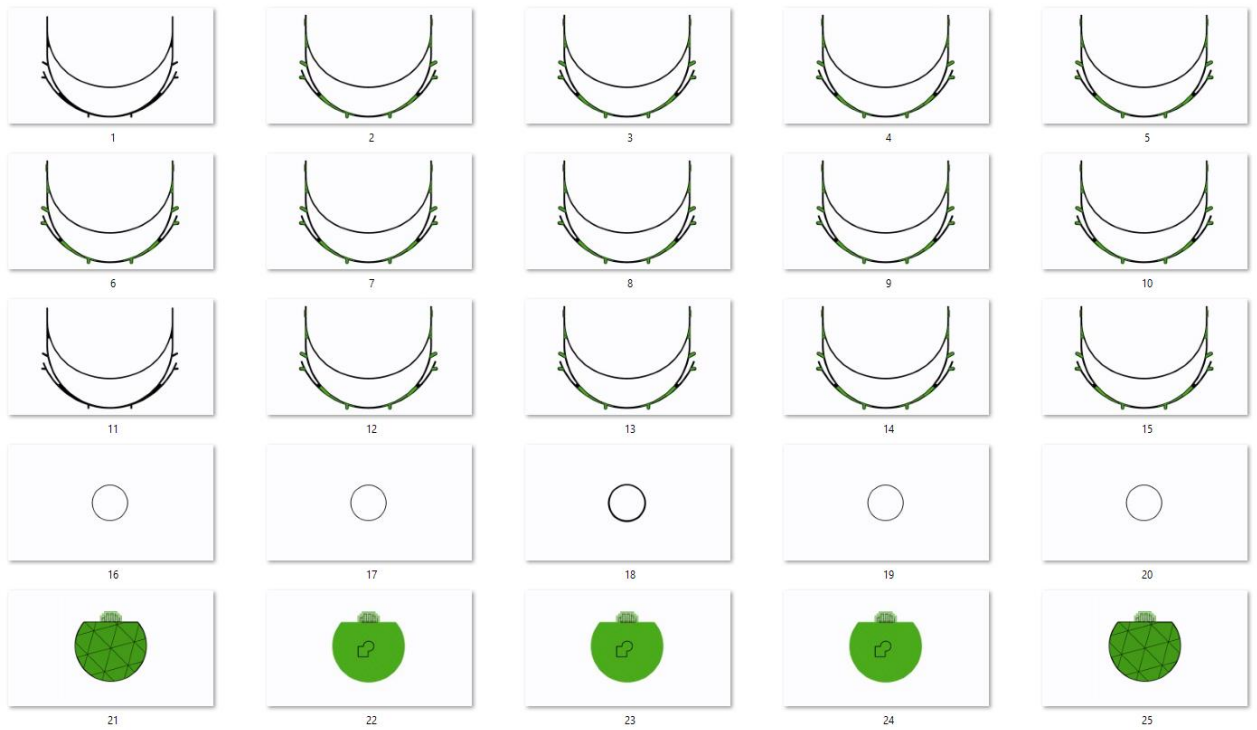
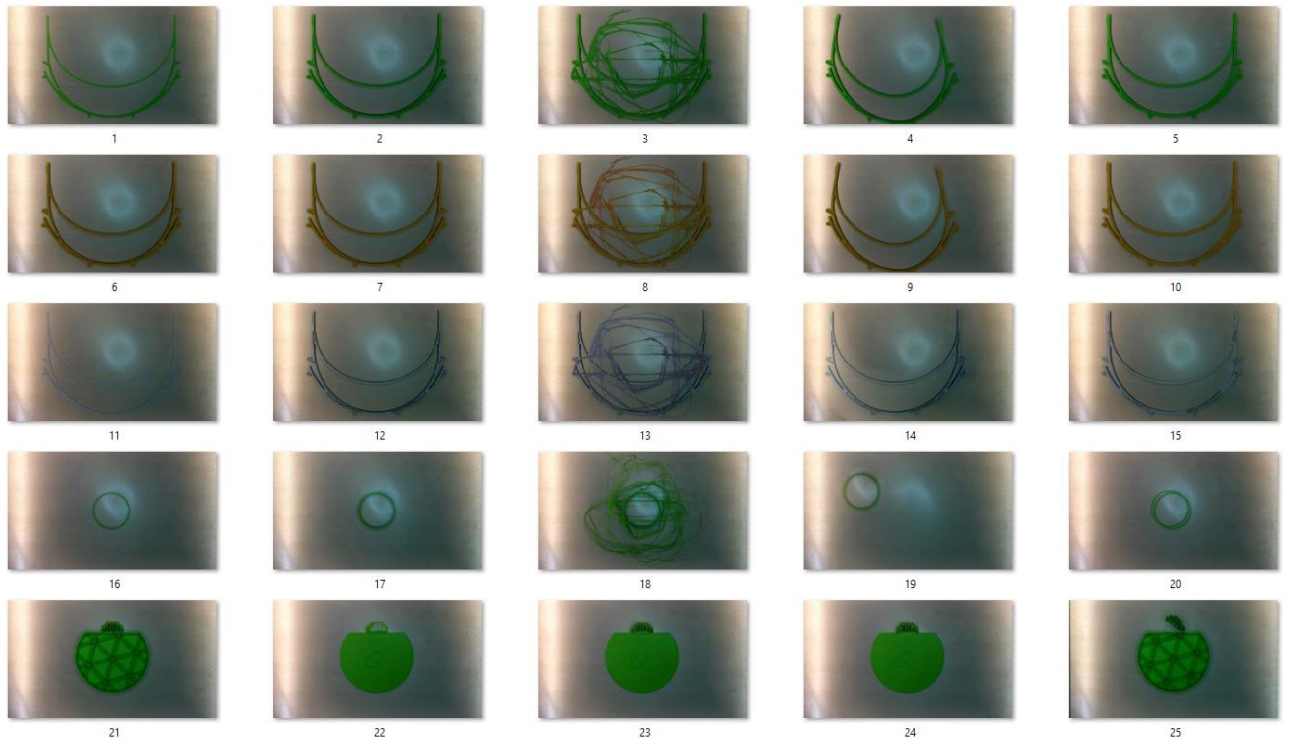
## Literatūros sąrašas

- [1] International Organization for Standardization, ISO/ASTM 52900, 2015, Additive manufacturing — General principles — Terminology.
- [2] BARNATT, Christopher. *3D Printing: Third Edition.*: CreateSpace Independent Publishing Platform, 2016. ISBN 9781539655466.
- [3] GUNAYDIN, Kadir and Halit TÜRKMEN. "Common FDM 3D Printing Defects", in *International Congress on 3D Printing (Additive Manufacturing) Technologies and Digital Industry*, 2018.
- [4] LAPPAS, Nikos. Machine Vision: 8 Companies that enable Industrial Robots to see. [interaktyvus]. [žiūrėta 2019-10-12]. Prieiga per: <https://www.smarterchains.com/smart-factories-news/industrial-machine-vision>
- [5] AHMAD, Majeed. "Machine Vision and the Evolution of Image Sensors", *ECN: Electronic Component News*, vol. 63, no. 3, pp. 34-35, 2019.
- [6] PONCE DE LEON, Rusty. "Avoid Costly Missteps When Selecting a Machine Vision Interface - Now and in the Future.," *Quality*, vol. SpecialIssue, pp. 12-15, 2019.
- [7] National Instruments. A Practical Guide to Machine Vision Lighting. [interaktyvus]. [žiūrėta 2019-10-17]. Prieiga per: <https://www.ni.com/en-us/innovations/white-papers/12/a-practical-guide-to-machine-vision-lighting.html>
- [8] McMENAMIN, Edward. "Lighting: A Back-To-Basics Look At Led Illumination For Machine Vision Applications," *Quality. Vision & Sensors*, pp. 14-15, Mar. 2018.
- [9] LEWIS, John. "How to Choose the Right Lighting for Machine Vision Applications.," *Quality*, vol. 52, no. 10, pp. 26-33, 2013.
- [10] Thresholding. [interaktyvus]. [žiūrėta 2019-10-24]. Prieiga per: [https://scikit-image.org/docs/dev/auto\\_examples/segmentation/plot\\_thresholding.html](https://scikit-image.org/docs/dev/auto_examples/segmentation/plot_thresholding.html)
- [11] Image Segmentation Using Color Spaces in OpenCV + Python. [Online]. [žiūrėta 2019-10-24]. Prieiga per: <https://realpython.com/python-opencv-color-spaces/>
- [12] DAVIES, E.R. *Computer and Machine Vision, Theory, Algorithms, Practicalities.*: Academic Press, Inc., 2012. ISBN 9780123869081.
- [13] TREIBER, Marco. *An Introduction to Object Recognition.*: Springer Science & Business Media, 2010. ISBN 9781849962353.

- [14] CANNY, John. "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.
- [15] BHARADI, Vinayak Ashok. Canny Edge Detection in C#. [interaktyvus]. [žiūrėta 2020-01-02]. Prieiga per: <https://www.codeproject.com/Articles/93642/Canny-Edge-Detection-in-C>
- [16] DERICHE, Rachid. "Using Canny's criteria to derive a recursively implemented edge detector," *International Journal of Computer Vision*, vol. 1, pp. 167-187, 1987.
- [17] FREEMAN, W.T. and E.H. ADELSON. "The Design and Use of Steerable Filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891-906, 1991.
- [18] LASKAUSKAS, Ramūnas and Vytautas VYŠNIAUSKAS. "Vaizdo kontūrų nustatymo būdų analizė," *Jaunųjų mokslininkų darbai*, vol. 5, no. 21, pp. 21-26, 2008.
- [19] AJAY, Pal and Chauhan SINGH and Bhardwaj SHARAT. "Detection of Bare PCB Defects by Image Subtraction Method using Machine Vision," in *Proceedings of the World Congress on Engineering 2011*, 2011.
- [20] BADRUL, M., and Y. SHAHRUL and A. ALIANA and A. NAZREN, and N. WAFI. "Template Matching using Sum of Squared Difference and Normalized Cross Correlation," in *2015 IEEE Student Conference on Research and Development*, 2015.
- [21] Open Source Computer Vision. [interaktyvus]. [žiūrėta 2020-1-12]. Prieiga per: <https://opencv.org/>
- [22] Insight Segmentation and Registration Toolkit (ITK). [interaktyvus]. [žiūrėta 2020-1-12]. Prieiga per: <https://itk.org/>
- [23] Integrating Vision Toolkit: IVT. [interaktyvus]. [žiūrėta 2020-1-12]. Prieiga per: <http://ivt.sourceforge.net/>
- [24] VTK - The Visualization Toolkit. [interaktyvus]. [žiūrėta 2020-1-12]. Prieiga per: <https://vtk.org/>
- [25] Duet3D Wiki. *Hardware overview*. [interaktyvus]. [žiūrėta 2020-10-01]. Prieiga per: [https://duet3d.dozuki.com/Wiki/Hardware\\_Overview](https://duet3d.dozuki.com/Wiki/Hardware_Overview)
- [26] Prusa Printers. *PRUSA PRO Face Shield*. [interaktyvus]. [žiūrėta 2020-10-01]. Prieiga per: <https://www.prusaprinters.org/prints/32714-prusa-pro-face-shield>
- [27] MBTechWorks. *RPi 1 B+ vs RPi 2 B Performance*. [interaktyvus]. [žiūrėta 2021-03-21]. Prieiga per: <https://www.mbtchworks.com/hardware/rpi-1-b-vs-rpi-2-b-performance.html>

## Priedai

### 1 priedas. Nuotraukų ir šablonų rinkinys naudotas parametrms nustatyti



## 2 priedas. Maskavimo įtaka šablono paieškos algoritmams

Scenos numeris	Be maskavimo		Su maskavimu	
	Koeficientas	Laikas, s	Koeficientas	Laikas, s
1	0.206	0.052	0.492	0.158
2	0.221	0.054	0.544	0.157
3	0.107	0.052	0.500	0.158
4	0.058	0.052	0.283	0.157
5	0.193	0.053	0.507	0.159
6	0.218	0.053	0.538	0.158
7	0.218	0.053	0.538	0.158
8	0.105	0.053	0.507	0.157
9	0.054	0.053	0.271	0.159
10	0.192	0.052	0.500	0.162
11	0.314	0.053	0.598	0.158
12	0.362	0.053	0.683	0.158
13	0.161	0.053	0.597	0.160
14	0.360	0.052	0.683	0.159
15	0.240	0.052	0.584	0.157
16	0.223	0.079	0.543	0.152
17	0.297	0.076	0.630	0.153
18	0.107	0.074	0.533	0.153
19	0.298	0.076	0.633	0.152
20	0.215	0.078	0.605	0.155
21	0.408	0.061	0.672	0.153
22	0.213	0.099	0.542	0.152
23	0.275	0.099	0.627	0.152
24	0.186	0.098	0.522	0.153
25	0.202	0.059	0.469	0.154
<b>Vidurkis</b>	0.217	0.064	0.544	0.156

### 3 priedas. Šablono paieškos algoritmų skaičiavimų laikas

Scenos nr.	<i>SQDIFF</i>	<i>CCORR</i>
	laikas, s	laikas, s
1	0.1603	0.1611
2	0.1573	0.1636
3	0.1564	0.1597
4	0.1551	0.1580
5	0.1586	0.1632
6	0.1577	0.1601
7	0.1569	0.1597
8	0.1557	0.1590
9	0.1556	0.1575
10	0.1549	0.1590
11	0.1574	0.1578
12	0.1597	0.1598
13	0.1552	0.1593
14	0.1558	0.1620
15	0.1597	0.1553
16	0.1501	0.1491
17	0.1505	0.1500
18	0.1503	0.1502
19	0.1487	0.1519
20	0.1441	0.1502
21	0.1482	0.1518
22	0.1508	0.1503
23	0.1506	0.1507
24	0.1499	0.1511
25	0.1498	0.1509
<b>Vidurkis</b>	0.1540	0.1561

#### 4 priedas. Bandymams naudotas defektų aptikimo sistemos programos kodas

```
import cv2 as cv
import numpy as np
import math
import csv

import parametrai
f= open("Tyrimas0.csv", "ab+")
ff=csv.writer(f, delimiter=',', quotechar="", quoting=csv.QUOTE_MINIMAL)
ff.writerow(["Sluoksni Numeris", "Defekto dydis", "Nuokrypis", "Korr koef", "laikas", "Rastas defektas"])

for i in range(1,26):

    t1 = cv.getTickCount()
    imgT = cv.imread('sabl/%s.png'%i)
    img = cv.imread('foto/%s.png'%i)

    gray_imgT = cv.cvtColor(imgT, cv.COLOR_BGR2GRAY)
    bil = cv.bilateralFilter(img,5,150,150)
    ret,th = cv.threshold(gray_imgT,188,255,cv.THRESH_BINARY_INV)
    retL,thL = cv.threshold(gray_imgT,70,255,cv.THRESH_BINARY_INV)
    kernel = np.ones((3,3),np.uint8) #tarpeliu uzpildymo filtro dydis
    closing = cv.morphologyEx(th, cv.MORPH_CLOSE, kernel)
    closingL = cv.morphologyEx(thL, cv.MORPH_CLOSE, kernel)

    hsv_img = cv.cvtColor(bil, cv.COLOR_BGR2HSV)
    obj = cv.inRange(hsv_img, parametrai.color, parametrai.color_d)
    defekt = cv.bitwise_xor(obj, th)
    kernel2 = np.ones((3,3),np.uint8) #triuksmu salinimo filtro dydis
    defektF = cv.morphologyEx(defekt, cv.MORPH_OPEN, kernel2,iterations = 2)

    Defekto_p = cv.countNonZero(defektF)
    Detales_p = cv.countNonZero(th)
    Santykisd = (Defekto_p * 1.0 / Detales_p) * 100

    edgesL = cv.Canny(closingL,10,245,3,L2gradient = True)

    koord = np.argwhere(edgesL>0)
    y1,x1 = koord.min(axis=0)
    y2,x2 = koord.max(axis=0)
    margin = 10
    cropped = edgesL[y1-margin:y2+margin, x1-margin:x2+margin]

    edges = cv.Canny(bil,30,45,3,L2gradient = True)
    kernel3 = np.ones((5,5),np.uint8) #padidinama zona pries palyginima
    dilate = cv.dilate(obj,kernel3,iterations = 1)
```

```

edgesF = cv.bitwise_and(edges,obj)

res = cv.matchTemplate(edgesF,cropped,cv.TM_CCORR_NORMED, mask=cropped)
min_val, max_val, min_loc, max_loc = cv.minMaxLoc(res)

nuokrypis_x = (max_loc[0] - (x1-margin))
nuokrypis_y = (max_loc[1] - (y1-margin))
nuokrypis = math.sqrt(nuokrypis_x ** 2 + nuokrypis_y ** 2)

t2 = cv.getTickCount()
time = (t2 - t1)/ cv.getTickFrequency()

if nuokrypis > 2.5:
    defektas = 1
elif Santykisd > 0.4 and parametrai.stalas == 0:
    defektas =1
else:
    defektas =0

ff.writerow([i, Santykisd, nuokrypis,max_val, time, defektas])

f.close()

```



## 5 priedas. Defektų aptikimo sistemos skaičiavimų laikas

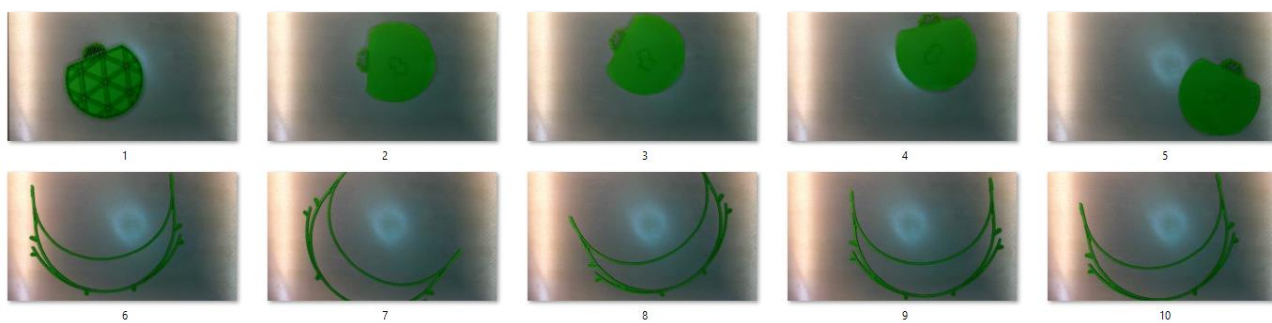
Scenos nr.	Skaičiavimo laikas, s							
	Intel Core i5-4670K			Vidurkis	Raspberry Pi B+			Vidurkis
1	0.244	0.251	0.249	0.248	10.062	11.226	10.513	10.600
2	0.251	0.257	0.256	0.255	11.009	10.819	11.455	11.094
3	0.271	0.282	0.286	0.279	11.297	11.743	11.552	11.531
4	0.254	0.254	0.253	0.254	11.365	10.604	10.868	10.946
5	0.273	0.268	0.272	0.271	11.268	11.049	11.464	11.260
6	0.275	0.275	0.274	0.275	12.041	12.297	11.789	12.042
7	0.262	0.279	0.274	0.272	10.827	11.753	11.411	11.330
8	0.280	0.271	0.273	0.275	11.666	11.316	11.279	11.420
9	0.273	0.273	0.273	0.273	11.272	11.484	11.145	11.300
10	0.274	0.275	0.274	0.274	11.445	11.463	11.424	11.444
11	0.254	0.256	0.256	0.256	10.935	11.116	11.450	11.167
12	0.259	0.250	0.253	0.254	10.921	10.969	11.091	10.994
13	0.277	0.276	0.278	0.277	12.385	12.117	11.696	12.066
14	0.272	0.278	0.277	0.276	11.907	11.697	12.158	11.921
15	0.265	0.273	0.266	0.268	11.392	11.626	10.774	11.264
16	0.245	0.239	0.246	0.243	10.029	10.370	10.139	10.179
17	0.245	0.246	0.246	0.246	10.854	10.888	10.057	10.600
18	0.241	0.248	0.248	0.246	10.574	10.661	10.125	10.453
19	0.252	0.262	0.254	0.256	11.258	11.366	11.231	11.285
20	0.261	0.252	0.261	0.258	11.208	10.604	11.565	11.126
21	0.260	0.257	0.257	0.258	10.828	10.612	11.390	10.943
22	0.243	0.249	0.250	0.247	9.915	10.081	10.413	10.136
23	0.259	0.263	0.257	0.260	11.481	11.762	11.286	11.510
24	0.267	0.259	0.265	0.264	11.013	10.822	11.408	11.081
25	0.258	0.258	0.260	0.259	10.887	11.184	10.747	10.939
<b>Vidurkis:</b>	0.261	0.262	0.262	<b>0.262</b>	11.114	11.185	11.137	<b>11.145</b>

## 6 priedas. Šablono vaizdų apdirbimo skaičiavimų laikas

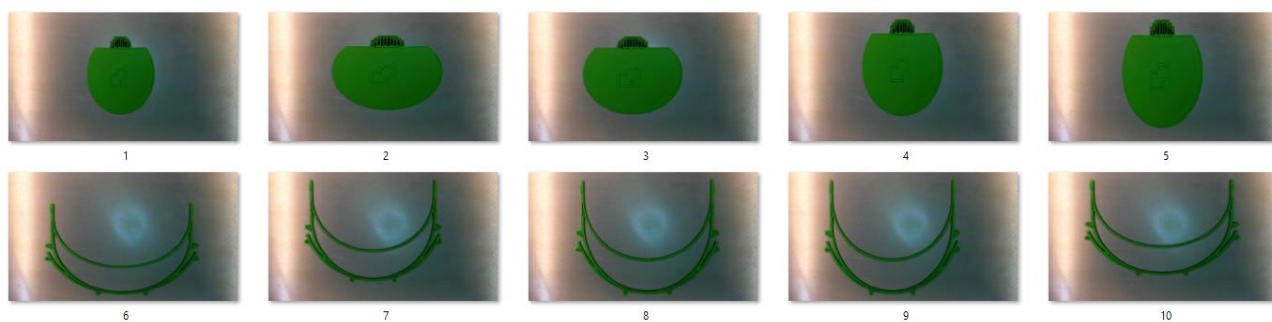
Scenos nr.	Skaičiavimo laikas, s							
	Intel Core i5-4670K			Vidurkis	Raspberry Pi B+			Vidurkis
1	0.0077	0.0076	0.0075	0.0076	0.3259	0.3154	0.3075	0.3163
2	0.0079	0.0085	0.0075	0.0080	0.3323	0.3792	0.3105	0.3407
3	0.0078	0.0081	0.0081	0.0080	0.3362	0.3573	0.3453	0.3463
4	0.0074	0.0081	0.0083	0.0079	0.3262	0.3517	0.3489	0.3422
5	0.0079	0.0076	0.0077	0.0078	0.3338	0.3340	0.3414	0.3364
6	0.0082	0.0083	0.0083	0.0083	0.3510	0.3465	0.3423	0.3466
7	0.0074	0.0083	0.0080	0.0079	0.3134	0.3478	0.3284	0.3299
8	0.0081	0.0075	0.0076	0.0077	0.3317	0.3051	0.3334	0.3234
9	0.0083	0.0077	0.0080	0.0080	0.3659	0.3337	0.3389	0.3462
10	0.0082	0.0082	0.0085	0.0083	0.3558	0.3388	0.3706	0.3551
11	0.0090	0.0079	0.0080	0.0083	0.3686	0.3275	0.3494	0.3485
12	0.0082	0.0074	0.0079	0.0078	0.3392	0.3301	0.3474	0.3389
13	0.0081	0.0081	0.0082	0.0081	0.3361	0.3498	0.3527	0.3462
14	0.0080	0.0082	0.0080	0.0080	0.3570	0.3333	0.3426	0.3443
15	0.0076	0.0087	0.0078	0.0080	0.3399	0.3510	0.3274	0.3394
16	0.0080	0.0073	0.0074	0.0076	0.3353	0.3204	0.2987	0.3181
17	0.0082	0.0085	0.0081	0.0083	0.3513	0.3714	0.3389	0.3539
18	0.0077	0.0078	0.0078	0.0077	0.3225	0.3348	0.3223	0.3265
19	0.0073	0.0078	0.0074	0.0075	0.2937	0.3482	0.3034	0.3151
20	0.0078	0.0076	0.0078	0.0077	0.3196	0.3119	0.3135	0.3150
21	0.0082	0.0082	0.0081	0.0082	0.3587	0.3476	0.3560	0.3541
22	0.0077	0.0080	0.0080	0.0079	0.3182	0.3407	0.3552	0.3380
23	0.0074	0.0077	0.0073	0.0075	0.3036	0.3460	0.3240	0.3245
24	0.0080	0.0078	0.0075	0.0077	0.3255	0.3401	0.3191	0.3283
25	0.0080	0.0082	0.0082	0.0081	0.3471	0.3670	0.3426	0.3522
<b>Vidurkis:</b>	0.0079	0.0080	0.0079	<b>0.0079</b>	0.3355	0.3412	0.3344	<b>0.3370</b>

## 7 priedas. Defektų aptikimo žalios spalvos detalėse eksperimento nuotraukos

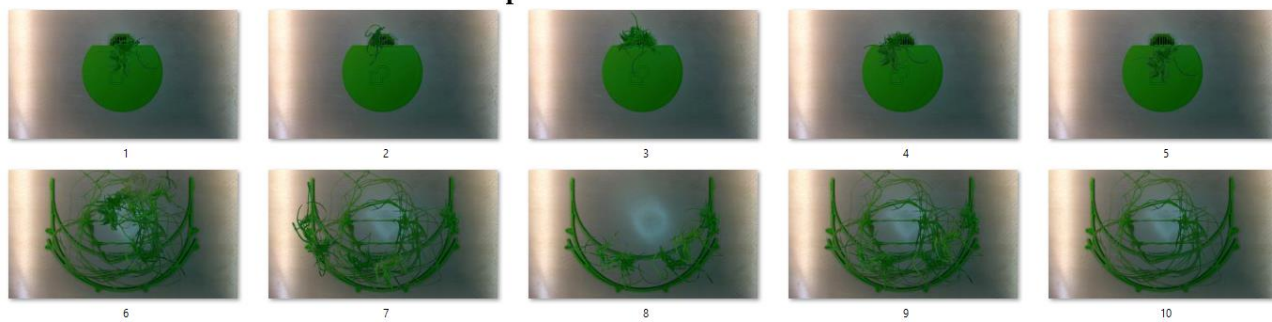
### Atsiklijavimo defektai



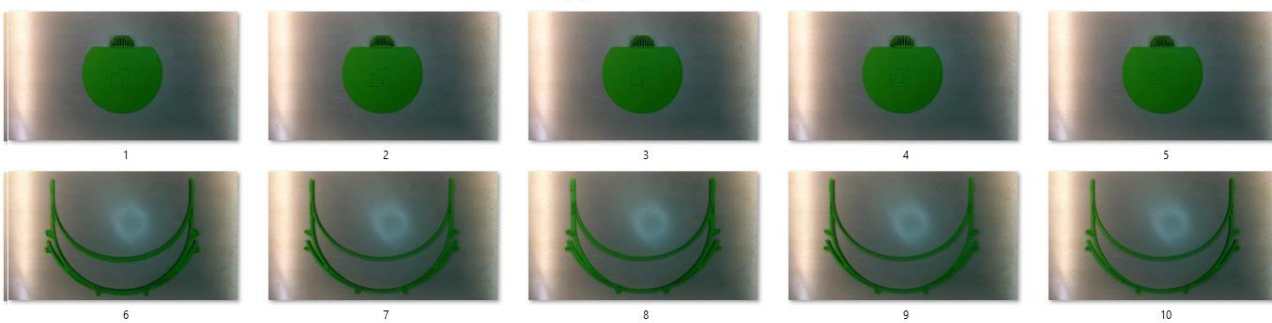
### Deformacijos defektai



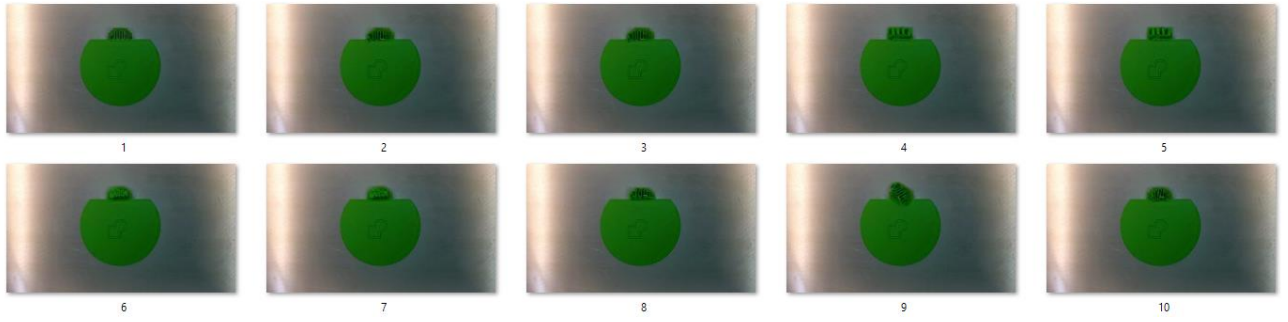
### Spausdinimo ore defektai



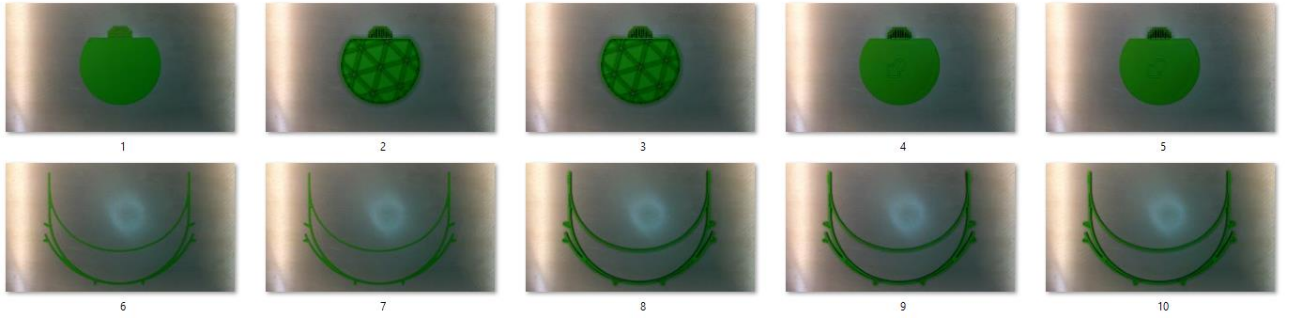
### Sluoksnių pasislinkimo defektai



### Prilaikančiųjų konstrukcijų defektai

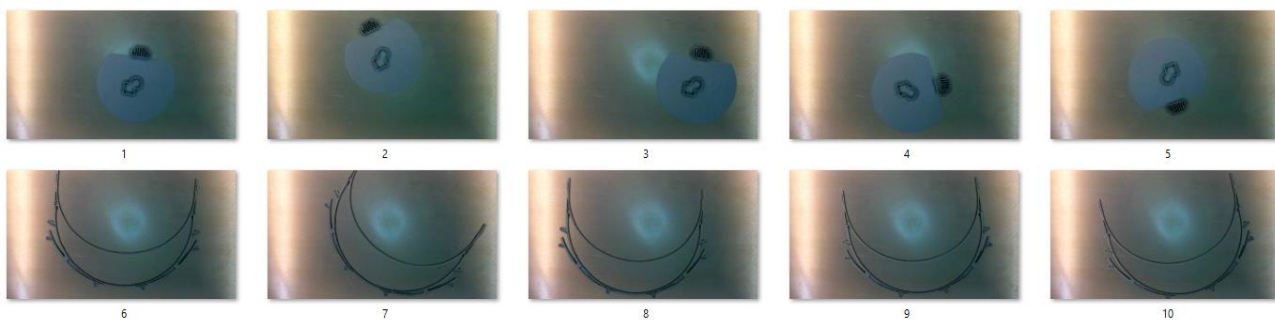


### Kokybiškos detalės

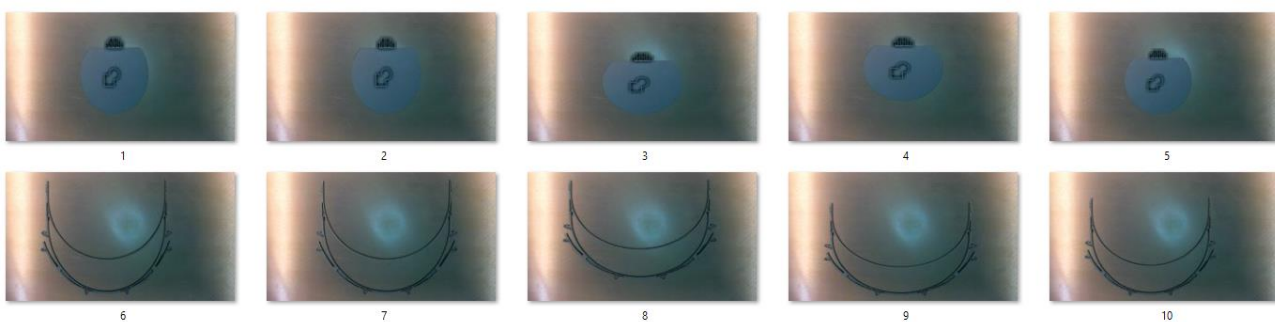


## 8 priedas. Defektų aptikimo pilkos spalvos detalėse eksperimento nuotraukos

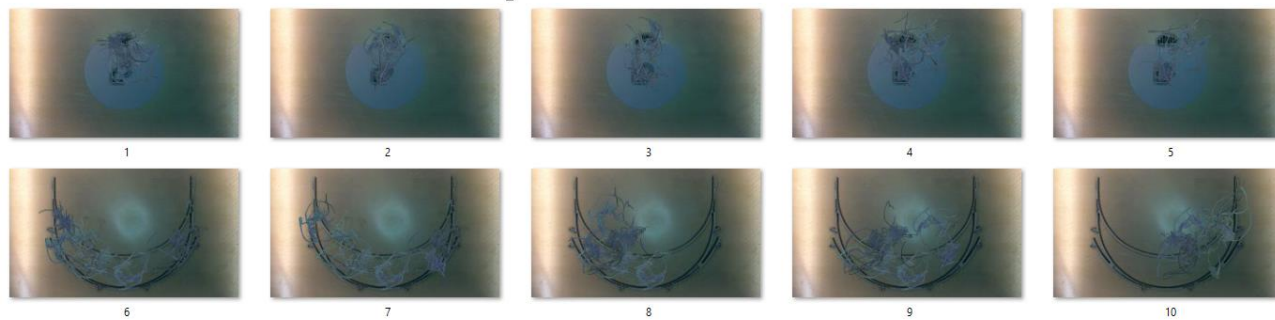
### Atsiklijavimo defektai



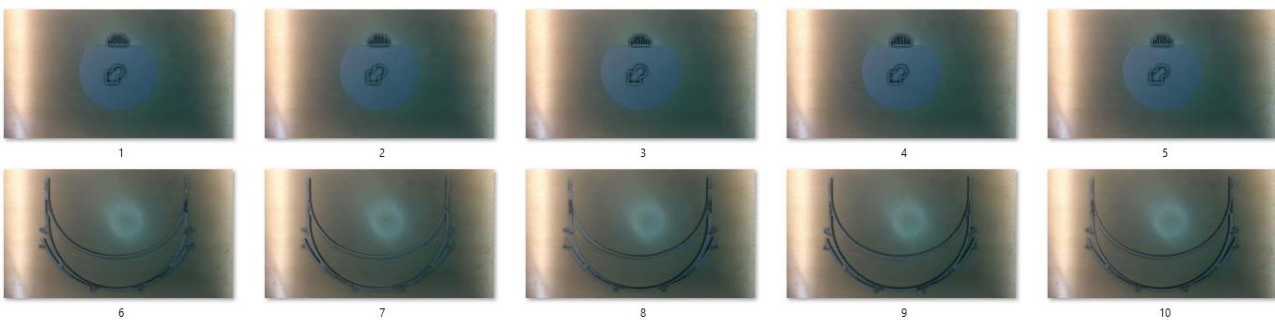
### Deformacijos defektai



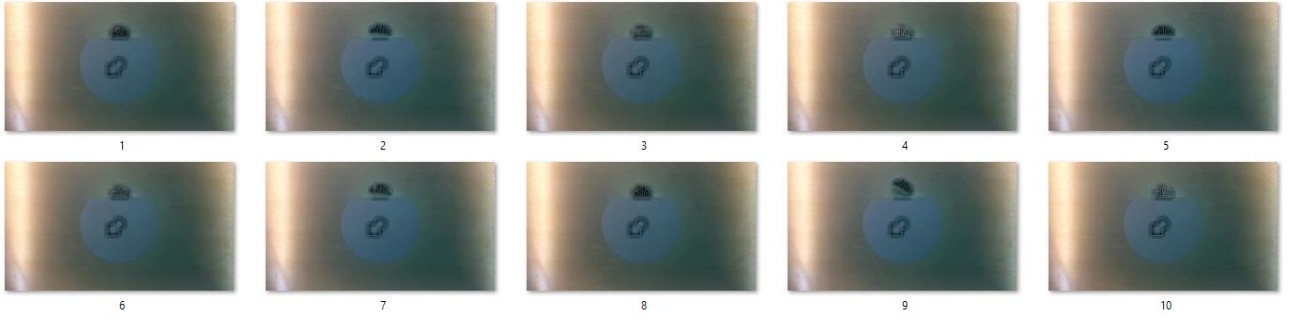
### Spausdinimo ore defektai



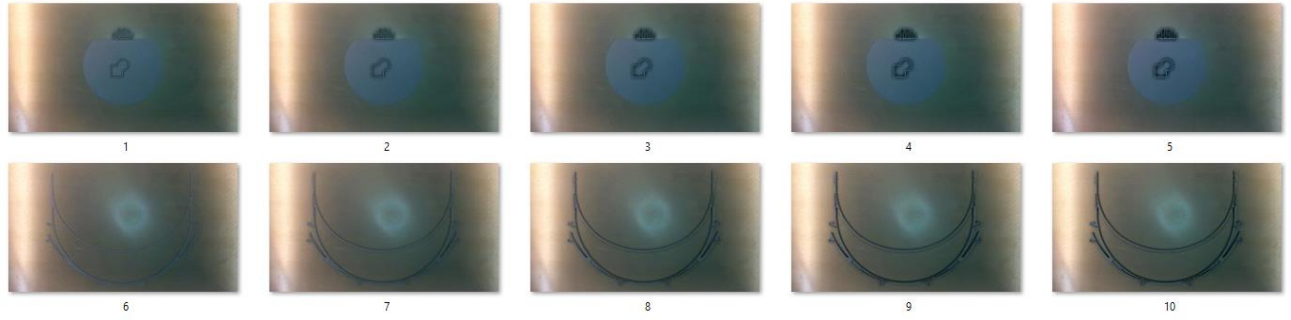
### Sluoksnių pasislinkimo defektai



### Prilaikančiųjų konstrukcijų defektai

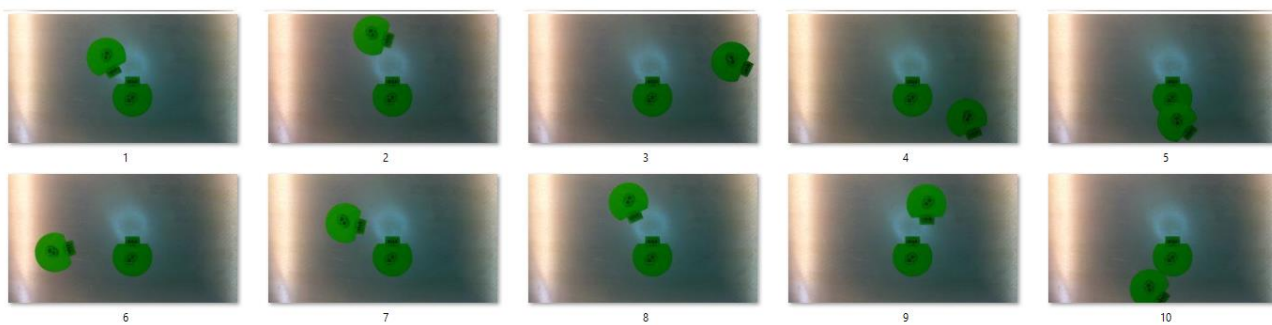


### Kokybiškos detalės

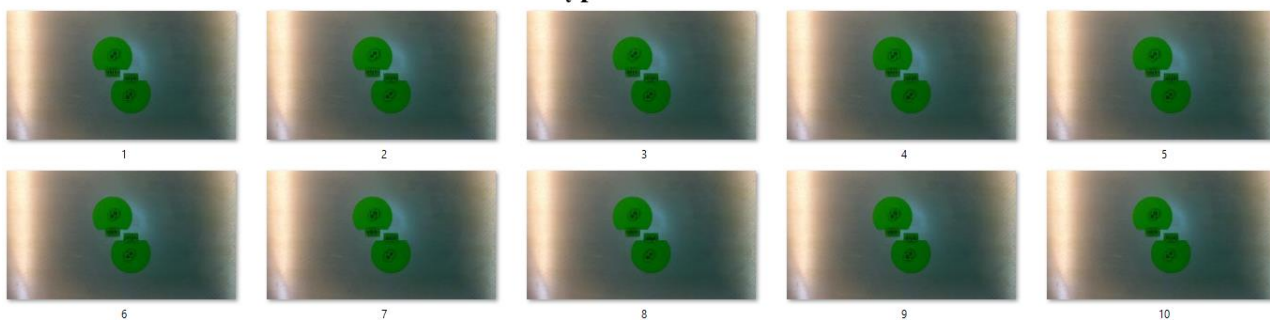


## 9 priedas. Defektų aptikimo kai spausdinamos dvi detalės eksperimento nuotraukos

### Atsiklijavimo defektai



### Sluoksnių pasislinkimo defektai



### Kokybiškos detalės

