

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Diplomski sveučilišni studij strojarstva

Diplomski rad

**PRIMJENA STROJNOG VIDA U DIMENZIONALNOJ
KONTROLI PROIZVODA**

Rijeka, ožujak 2019.

Tomislav Bazina

0035177436

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Diplomski sveučilišni studij strojarstva

Diplomski rad

**PRIMJENA STROJNOG VIDA U DIMENZIONALNOJ
KONTROLI PROIZVODA**

Mentor: Prof. dr. sc. Zoran Jurković

Komentor: Prof. dr. sc. Duško Pavletić

Rijeka, ožujak 2019.

Tomislav Bazina

0035177436

TEHNIČKI FAKULTET

Povjerenstvo za diplomske ispite diplomskog sveučilišnog studija strojarstva

Klasa: 602-04/18-05/07

Ur. br.: 2170-15-12-18-3

Rijeka, 13. 3. 2018.

Z A D A T A K za diplomski rad

Pristupnik: **TOMISLAV BAZINA**

JMBAG: 0035177436

Naziv zadatka: **Primjena strojnog vida u dimenzionalnoj kontroli proizvoda**

Title (English): **Machine vision application in workpieces dimensional control**

Polje znanstvenog područja: strojarstvo

Grana znanstvenog područja: proizvodno strojarstvo

Sadržaj zadatka:

U uvodnom dijelu rada potrebno je dati pregled stanja u području strojnog vida, s naglaskom na postojeća rješenja, kao i trendove razvoja po pitanju softvera i hardvera. Na konkretnom primjeru potrebno je prikazati mogućnosti primjene strojnog vida u laboratorijskom okruženju s ciljem dimenzijske kontrole proizvoda u stvarnom vremenu. Također potrebno je opisati predloženu strukturu sustava strojnog vida, razviti programsku podršku, te provesti potrebna mjerenja kao i analizu dobivenih rezultata.

U radu treba navesti korištenu literaturu, druge izvore informacija, kao i eventualno dobivenu pomoć. *Diplomski rad mora biti napisan prema Uputama za pisanje diplomskih radova koje su objavljene na web stranicama Diplomskog sveučilišnog studija strojarstva.*

Zadano: **19. 3. 2018.**

Zadatak vrijedi do **16. 5. 2019.**

Mentor:

Izv. prof. dr. sc. Zoran Jurković



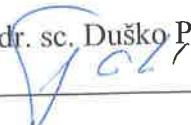
Predsjednik Povjerenstva:

Prof. dr. sc. Kristian Lenić



Komentor:

Prof. dr. sc. Duško Pavletić



Zadatak preuzeo dana: 19. 03. 2018.



(potpis pristupnika)

Dostaviti:

- Mentor (2x) Evidencija studija (2x) Pristupnik (1x) Predsjednik povjerenstva (1x)

IZJAVA O SAMOSTALNOJ IZRADI RADA

Ja, Tomislav Bazina, pod punom materijalnom, kaznenom i moralnom odgovornošću, izjavljujem da sam jedini autor završnog rada pod naslovom „Primjena strojnog vida u dimenzionalnoj kontroli proizvoda“, te da u navedenom radu nisu, bez pravilnog referenciranja, korišteni dijelovi tuđih radova.

Tomislav Bazina

ZAHVALA

Zahvaljujem mentorima prof. dr. sc. Zoranu Jurkoviću i prof. dr. sc. Dušku Pavletiću na svoj pruženoj pomoći i korisnim savjetima pri izradi ovog diplomskog rada. Također, zahvaljujem doc. dr. sc. Draženu Brščiću i asistentici Maji Marković na uloženom trudu i vremenu. Posebno zahvaljujem poduzeću Kovinotokarska Radnja Mandić Nandor na omogućenom korištenju mjernih instrumenata, posuđenoj opremi i fleksibilnom radnom vremenu. Veliko hvala obitelji na pruženoj podršci i Virni na bezuvjetnoj potpori i ogromnoj pomoći.

SADRŽAJ

1. UVOD	1
2. STRUKTURA I LANAC OBRADE SUSTAVA STROJNOG VIDA	3
2.1. Struktura sustava strojnog vida.....	3
2.1.1. Način formiranja digitalne slike	4
2.2. Mjereni proizvodi	6
2.2.1. Mjerene veličine ručnim putem i pripadajuće tolerancije	8
2.2.2. Mjerene veličine optičkim putem.....	9
2.3. Predloženi optički mjerni sustav.....	14
2.3.1. Kamera Basler ace USB 3.0 acA2500-60uc	14
2.3.2. Objektiv Edmund Optics fiksne žarišne duljine 16 mm	18
2.3.3. Stolna LED svjetiljka u obliku prstena	21
2.3.4. Kalibracija optičkog mjernog sustava	21
2.3.5. Postavljanje optičkog sustava.....	26
2.4. Digitalna obrada slike	29
2.4.1. Algoritam <i>Gaussian Convolution</i>	30
2.4.2. Algoritam <i>Kuwahara</i>	31
2.4.3. Algoritam <i>Minima</i>	31
2.4.4. Algoritam <i>Gradient Magnitude</i>	33
2.4.5. Algoritam <i>Seeded Watershed</i>	33
2.4.6. Algoritam <i>Remove Edge Objects</i>	35
2.4.7. Algoritam <i>Fill Holes</i>	35
2.4.8. Algoritam <i>Apply Mask</i>	36
2.4.9. Algoritam <i>Label</i>	36
2.4.10. Algoritam <i>Measure</i>	37
2.4.11. Algoritam <i>Working Distance Correction</i>	38
2.4.12. Algoritam <i>Binary Area Closing</i>	38
2.4.13. Algoritam <i>Segment O-ring</i>	39

2.4.14.	Algoritam <i>Set Pixel Size</i>	40
2.4.15.	Algoritam predobrade slike	41
3.	PROGRAMSKA PODRŠKA MJERNOG SUSTAVA	42
3.1.	Rukovanje kamerom i glavni prozor grafičkog sučelja programske podrške	45
3.2.	Kalibracija mjernog sustava pomoću programske podrške	47
3.3.	Mjerni algoritam i postupak mjerenja predmeta.....	54
4.	MJERENJA I ANALIZA SPOSOBNOSTI SUSTAVA	62
4.1.	Postupak i rezultati mjerenja proizvoda mikrometarskim vijkom.....	62
4.2.	Analiza sposobnosti ručnog mjernog sustava (mjerenja mikrometarskim vijkom)	70
4.2.1.	Sposobnost sustava pri mjerenju poprečnog presjeka O-prstena.....	74
4.2.2.	Sposobnost sustava pri mjerenju unutarnjeg promjera mikrometarskim vijkom....	76
4.2.3.	Sposobnost sustava pri mjerenju vanjskog promjera mikrometarskim vijkom	81
4.3.	Postupak i rezultati mjerenja proizvoda u optičkom mjernom sustavu.....	84
4.4.	Analiza sposobnosti optičkog mjernog sustava.....	86
4.4.1.	Sposobnost optičkog sustava pri mjerenju unutarnjeg promjera O-prstena.....	90
4.4.2.	Sposobnost optičkog sustava pri mjerenju vanjskog promjera O-prstena	91
4.5.	Analiza osjetljivosti optičkog mjernog sustava	95
4.6.	Usporedba optičkog (automatiziranog) i ručnog mjernog sustava	99
5.	ZAKLJUČAK	104
	LITERATURA.....	105
	POPIS OZNAKA I KRATICA	108
	SAŽETAK.....	111
	PRILOZI.....	112
	Prilog 1.: Programski kod podrške mjernom sustavu	112
1.1.	Skripta „main.py“	112
1.2.	Skripta „cam.py“	114
1.3.	Skripta „event_handlers.py“	119
1.4.	Skripta „flowcharts.py“	121

1.5.	Skripta „mainwindow.py“	127
1.6.	Skripta „ptrees.py“	133
1.7.	Skripta „visualize.py“	137
1.8.	Skripta „nodes.py“	144
1.9.	Skripta „process.py“	169
Prilog 2.: Programski kod za analizu sposobnosti mjernog sustava		172
2.1.	Skripta „measurement_system_analysis.R“	172
Prilog 3.: Potvrde o umjeravanju.....		177
3.1.	Potvrda o umjeravanju mikrometarskog vijka <i>Mitutoyo</i> MDC-25SX.....	177
3.2.	Potvrda o umjeravanju kontrolne palice NM.KPR.002	178
Prilog 4.: Mjerni listovi s izmjerenim vrijednostima veličina mikrometarskim vijkom		180
4.1.	Mjerni list s izmjerenim vrijednostima poprečnog presjeka [mm] O-prstena	180
4.2.	Mjerni list s izmjerenim vrijednostima unutarnjeg promjera [mm] O-prstena.....	181
4.3.	Mjerni list s izmjerenim vrijednostima vanjskog promjera [mm] O-prstena.....	182
Prilog 5.: Rezultati mjerenja O-prstena pomoću optičkog mjernog sustava		183
5.1.	Rezultati prvog mjerenja svih O-prstena	183
5.2.	Rezultati drugog mjerenja svih O-prstena	186
5.3.	Rezultati trećeg mjerenja svih O-prstena	189
5.4.	Rezultati četvrtog mjerenja svih O-prstena.....	192
5.5.	Rezultati petog mjerenja svih O-prstena.....	195
5.6.	Rezultati šestog mjerenja svih O-prstena.....	198
5.7.	Rezultati sedmog mjerenja svih O-prstena	201
5.8.	Rezultati osmog mjerenja svih O-prstena	204
5.9.	Rezultati devetog mjerenja svih O-prstena	207
Prilog 6.: Tolerancije mjerenih veličina O-prstena opće namjene [7]		210
Prilog 7.: Nacrt objektiva Edmund Optics žarišne duljine 16 mm [14].....		211

1. UVOD

Strojni vid, kao područje umjetne inteligencije, se u posljednjih nekoliko godina intenzivno razvija. Njegova primjena u robotici, i automatiziranoj kontroli je od velikog interesa, kako mnogim istraživačima, tako i mnogim inženjerima. U industriji postoje brojni primjeri uspješne implementacije sustava strojnog vida, pomoću kojih se snižavaju troškovi i povećava kvaliteta proizvodnje. U ovom radu će naglasak biti na optičkim sustavima za kontrolu proizvoda, koji imaju ogroman potencijal za razvoj i primjenu. Konstantno se razrađuju novi algoritmi, novi uređaji i naprednije metode za primjenu u automatiziranim sustavima za optičku kontrolu kvalitete. S druge strane, takav razvoj ima zadaću pratiti zahtjeve za konstantnim povećanjem kvalitete procesa proizvodnje i kontrole, te samih proizvoda. Automatizirana optička kontrola je interdisciplinarna tematika, koja obuhvaća područja optike, matematike, računarstva, strojarstva i elektrotehnike.

Rad je strukturiran na tri cjeline. Cilj rada je razraditi cijeli sustav za kontrolu dimenzija odabranog proizvoda. Automatizirani optički sustavi se uglavnom projektiraju na temelju jednog proizvoda, odnosno skupine sličnih proizvoda. Proizvod, čije dimenzije će se kontrolirati je O-prsten. O-prsteni su brtveni elementi relativno jednostavnog oblika i široke upotrebe. Proizvode se u velikim serijama, što ih čini idealnim za automatiziranu optičku kontrolu. S obzirom na navedeno, vizualna kontrola O-prstena je već bila predmet istraživanja. Istraživanja su bila fokusirana na kontrolu dimenzija vanjskog i unutarnjeg promjera, te površinskih pogrešaka tijekom proizvodnje [1]. Pri kontroli brtvenih elemenata složenijeg poprečnog presjeka razvijeni su sustavi za 3D inspekciju svih strana elementa laserom [2]. Također, izrađeni su i specijalni industrijski sustavi kontrole velike propusne moći. Neki sustavi su bez pokretnih dijelova, poput *Planar2D* [3] sustava. Proizvođač tvrdi da sustav može izmjeriti $100 \div 1000$ O-prstena u vremenu od 0,1 s uz preciznost od $12 \mu\text{m}$. Drugi sustavi imaju mogućnost kontrole O-prstena i sličnih okruglih proizvoda sa svih strana, te automatiziranog pakiranja ispravnih proizvoda [4]. Treći tipovi sustava su bazirani na vibracijskim i rotacijskim stolovima, te mlaznicama za odvajanje ispravnih od neispravnih proizvoda [5]. Proizvođač tvrdi da je propusna moć takvog sustava 150 O-prstena u minuti. Sustav predložen u radu neće imati takve sposobnosti, no softver sustava će biti razvijen od potpuno besplatnih rješenja otvorenog koda.

U prvom dijelu rada bit će pojašnjene osnove sustava strojnog vida i elementi, koje svaki takav sustav mora sadržavati. Lanac obrade sustava strojnog vida će biti prikazan u obliku piramide. Kao i u mnogim drugim sustavima, pretpostavka kvalitete je dobra osnova sustava, odnosno baza

piramide. U sustavu strojnog vida je to proces prikupljanja slika za obradu. U svim sljedećim koracima je teško, ili nemoguće, nadoknaditi loše prikupljene informacije u prvom koraku. U prvom dijelu rada će biti opisan i mjereni proizvod, odnosno mjerene veličine tog proizvoda pomoću mikrometarskog vijka i optičkim putem. Bit će predstavljene i komponente optičkog sustava, te način postavljanja sustava prije puštanja u rad. Na kraju prvog dijela će biti pojašnjene osnove digitalne obrade slike i algoritmi korišteni u programskoj podršci optičkog sustava.

Drugi dio rada će biti vezan uz programsku podršku mjernog sustava. Objasniti će se pravilno korištenje programskog rješenja od samog pokretanja do spremanja rezultata mjerenja. Bit će prikazan način upravljanja parametrima kamere, postupak pripreme kalibracijskog i mjernog algoritma, te sam postupak provođenja mjerenja.

U posljednjem dijelu rada bit će analizirani dobiveni rezultati mjerenja. Prvo će se analizirati sposobnost ručnog mjernog sustava, a zatim automatiziranog. Nakon analize osjetljivosti optičkog sustava na kalibraciju, dva sustava će biti uspoređena i vizualno prikazana korelacija rezultata mjerenja.

2. STRUKTURA I LANAC OBRADE SUSTAVA STROJNOG VIDA

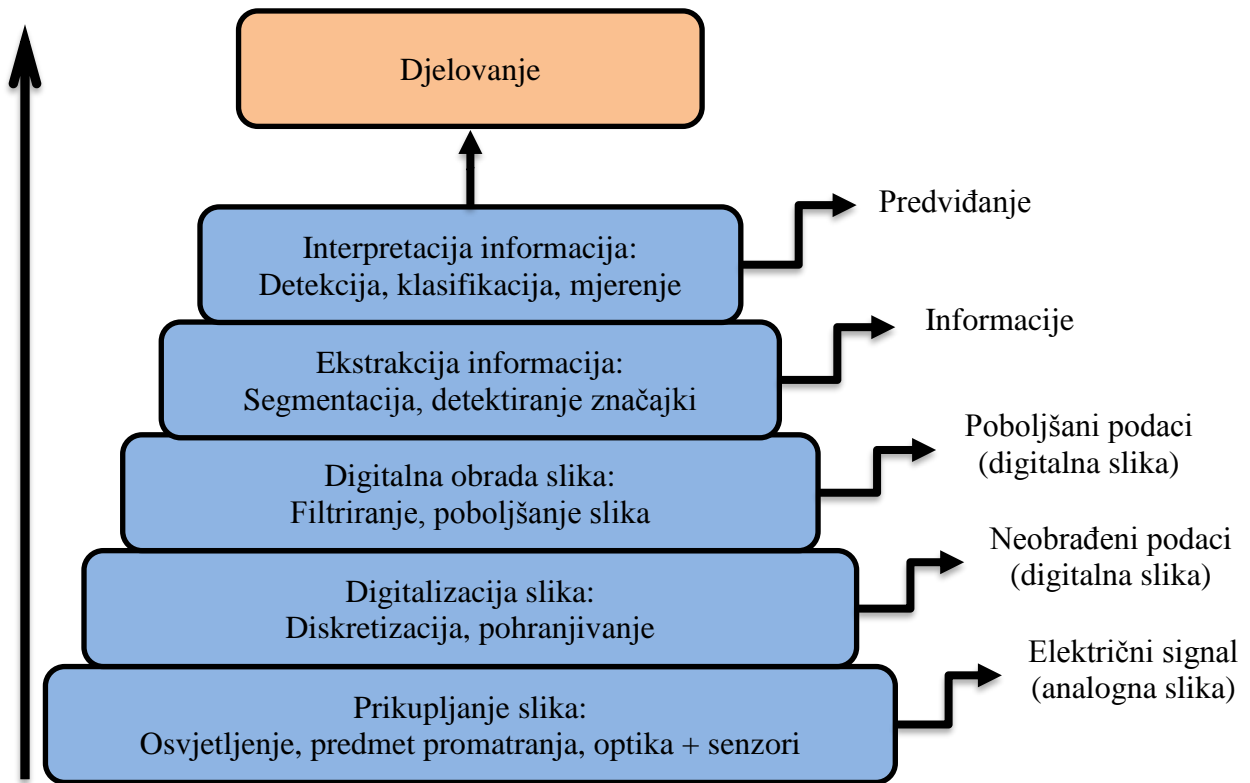
U narednih nekoliko potpoglavlja su pojašnjeni osnovni principi rada sustava strojnog vida, način pribavljanja i digitalizacije slika, te postupak digitalne obrade slika. Uz to je odabran i proizvod, za čiju dimenzijsku kontrolu je projektiran optički mjerni sustav. Definirane su karakteristike, oznake i mjerene veličine odabranog proizvoda, te komponente, kalibracija i postavljanje predloženog automatiziranog mjernog sustava.

2.1. Struktura sustava strojnog vida

Lanac obrade sustava strojnog vida se, po uzoru na [6], sastoji od koraka prikazanih na slici 2.1. Kao baza piramide sustava se nalazi postupak prikupljanja slika. Taj postupak je iznimno važan čimbenik koji diktira kvalitetu cijelog sustava. Najniža razina sustava podrazumijeva svu strojnu opremu potrebnu za izvršenje zadatka sustava, poput kvalitetnog osvjetljenja, predmeta promatranja, te sustava leća i senzora za prikupljanje slika. Informacije sa slike, koje nisu prikupljene tijekom prvog koraka, teško se mogu povratiti su sljedećim koracima obrade.

Svi naredni koraci obrade su vezani uz softversku podršku sustava strojnog vida. Robusna računalna podrška je također vrlo važan sastavni dio sustava. U drugom koraku se analogni signal digitalizira i pretvara u diskretan oblik čitljiv računalu. Postupak pretvaranja analognog signala u digitalni neminovno povlači sa sobom gubitak informacija. Digitalna slika uglavnom sadrži i smetnje, odnosno „šum“, što otežava prikupljanje bitnih informacija. Stoga postoje razni algoritmi za digitalnu obradu slike, od kojih su neki pojašnjeni u potpoglavlju 2.4, pomoću kojih se u određenoj mjeri uklanja „šum“ i povećava kontrast područja interesa (ROI) slike. Ekstrakcija informacija se vrši na poboljšanoj slici, te također uključuje brojne algoritme za segmentaciju i detektiranje značajki slike. Nakon što su dobivene relevantne informacije za izvršavanje zadatka sustava, potrebno ih je interpretirati na pravilan način, donijeti odluku i izvršiti djelovanje.

U slučaju primjene strojnog vida pri mjerenju predmeta, što je zadatak ovog rada, ne postoji korak u kojem se djeluje na sustav. Izlaz sustava su rezultati mjerenja, odnosno izmjerene vrijednosti mjerenih veličina opisanih u potpoglavlju 2.2.2. Pri primjeni sustava u praksi je moguće uvesti djelovanje na proizvodni sustav temeljem praćenja izmjerenih vrijednosti veličina u vremenu, no to je izvan opsega ovog rada.



Slika 2.1. Lanac obrade sustava strojnog vida [6]

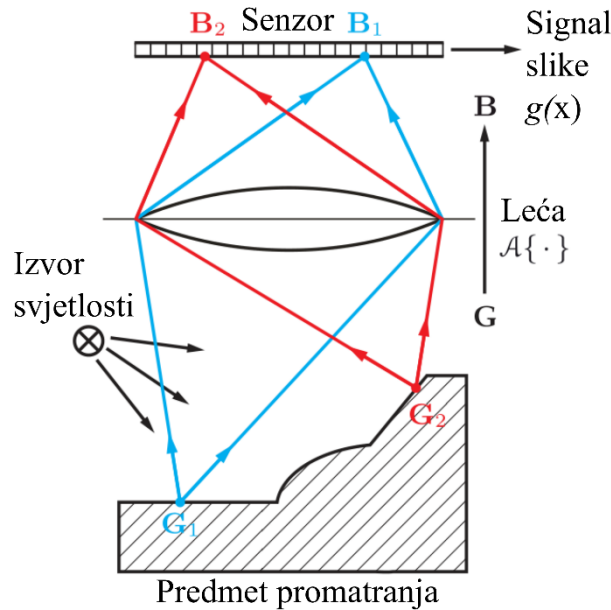
2.1.1. Način formiranja digitalne slike

Prilikom prikupljanja slika za analizu koristi se već opisan sustav, prikazan na slici 2.2. Prema [6], sustav se sastoji od izvora svjetlosti, senzora, leće kamere i promatranog predmeta. Svjetlosne zrake iz izvora svjetlosti osvjetljavaju promatrani predmet. Dio svjetlosti se odbija od točke **G** predmeta, prolazi kroz leću i konvergira u jednu točku **B** na senzoru. Idealno svaka točka predmeta odgovara točno jednoj točki na senzoru prema jednadžbi [6]:

$$\mathbf{B} = \mathcal{A}(\mathbf{G}), \quad (2.1)$$

gdje je:

- $\mathcal{A}(\cdot)$ [/] Prijenosna funkcija idealne leće,
- \mathbf{B} [/] Točka senzora optičkog sustava,
- \mathbf{G} [/] Točka promatranog predmeta optičkog sustava.



Slika 2.2. Način formiranja digitalne slike [6]

Senzori u digitalnim kamerama su dvodimenzionalni, što rezultira izlaznim dvodimenzionalnim signalom slike prema izrazu [6]:

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_Q(\mathbf{x}) \end{pmatrix}, \quad (2.2)$$

gdje je:

- $\mathbf{g}(\mathbf{x})$ [/] Funkcija za dodjeljivanje vektorske vrijednosti na poziciji \mathbf{x} na slici,
- $g_i(\mathbf{x})$ [/] Funkcija za dodjeljivanje vrijednosti pojedinoj dimenziji vektora na poziciji \mathbf{x} na slici, $i \in [1, 2, \dots, Q]$
- \mathbf{x} [/] Dvodimenzionalni vektor koordinata slike,
- Q [/] Prostorna dimenzija vrijednosti slike (*image channel*).

Dvodimenzionalni vektor koordinata slike je određen jednadžbom [6]:

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad (2.3)$$

gdje je:

- x [/] x -koordinata slike,
- y [/] y -koordinata slike.

Pri digitalnoj obradi slike, ne koristi se kontinuirani signal $\mathbf{g}(\mathbf{x})$, nego diskretni \mathbf{g}_{mn} . Diskretizacija signala nastaje u dva pogleda. Prvo se signal prostorno uzorkuje pomoću mreže od $M \times N$ piksela, a zatim se i same vrijednosti kvantiziraju u K razina [6]:

$$\mathbf{g}_{mn} \in [0, 1, 2, \dots, K - 1]^Q, \quad (2.4)$$

gdje je:

\mathbf{g}_{mn} [/]	Diskretne vrijednosti intenziteta svakog piksela slike,
m [/]	Prva prostorna koordinata piksela slike, $m \in [0, 1, 2, \dots, M - 1]$,
n [/]	Druga prostorna koordinata piksela slike, $n \in [0, 1, 2, \dots, N - 1]$,
$M \times N$ [/]	Rezolucija slike u pikselima,
K [/]	Broj razina koje može poprimiti vrijednost intenziteta piksela slike.

U slučaju prikazivanja slike u nijansama sive boje, prostorna dimenzija vrijednosti intenziteta slike je $Q = 1$, svaki piksel može poprimiti $K = 255$ različitih vrijednosti, a funkcija za dodjeljivanje vrijednosti svakom pikselu je sljedeća [6]:

$$\mathbf{g}(\mathbf{x}) = g(\mathbf{x}). \quad (2.5)$$

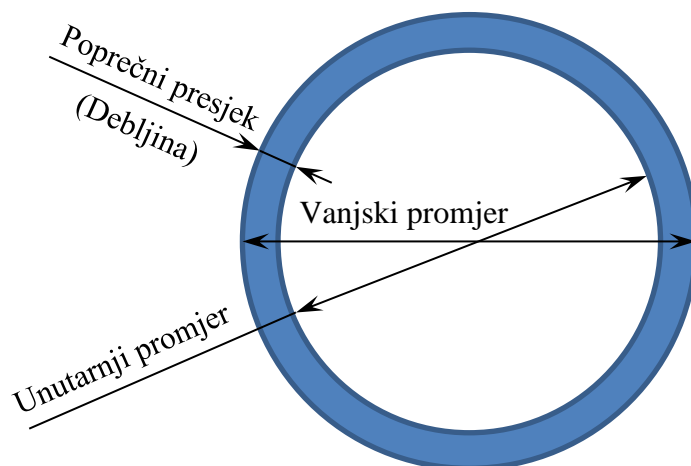
U slučaju prikazivanja digitalnih slika u boji, prostorna dimenzija vrijednosti je $Q = 3$, svaki piksel može poprimiti $K = 255$ vrijednosti u svakoj od tri dimenzije, a funkcija za dodjeljivanje vrijednosti je [6]:

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} g_1(\mathbf{x}) \rightarrow \text{crvena} \\ g_2(\mathbf{x}) \rightarrow \text{zelena} \\ g_3(\mathbf{x}) \rightarrow \text{plava} \end{pmatrix}, \quad (2.6)$$

U optičkom sustavu, koji je predstavljenom u ovom radu, slike se prilikom prikupljanja pretvaraju iz slika u boji u slike u nijansama sive boje.

2.2. Mjereni proizvodi

Odabrani proizvodi, za čije mjerenje je osmišljen optički sustav, su O-prsteni od nitrilne gume. O-prsteni su kružnog poprečnog presjeka i služe kao brtveni elementi opće namjene. Skica O-prstena s mjerenim veličinama prikazana je na slici 2.3. O-prsteni se kataloški označavaju pomoću unutarnjeg promjera i poprečnog presjeka.



Slika 2.3. O-prsten s mjerenim veličinama

Za mjerenje su odabrani O-prsteni nominalnog vanjskog promjera 15,8 mm i nominalnog poprečnog presjeka 2,4 mm. Takvih 10 O-prstena je sortirano i numerički označeno brojevima 1÷10. Sortirani O-prsteni 15,8 x 2,4 mm, uz još dvije varijante većih O-prstena, prikazani su na slici 2.4.



Slika 2.4. Sortirani O-prsteni zbog identifikacije pri mjerenju

Postupak sortiranja je nužan zbog identifikacije svakog O-prstena tijekom mjerenja i povezivanja izmjerenih vrijednosti s pripadajućim O-prstenom. U ostatku rada, proizvodi su označeni na način da se prvo upisuje nazivni unutarnji promjer O-prstena, zatim deseterostruka nazivna vrijednost poprečnog presjeka i na kraju identifikacijski broj:

- 15,8 x 24_1,
- 15,8 x 24_2,
- ...
- 15,8 x 24_10.

Provedena su mjerenja veličina O-prstena u dva različita mjerna sustava. Prvo su, zbog dobivanja referentnih vrijednosti veličina, izvršena mjerenja O-prstena mikrometarskim vijkom, a zatim pomoću optičkog mjernog sustava.

2.2.1. Mjerene veličine ručnim putem i pripadajuće tolerancije

Prije mjerenja proizvoda optičkim putem, isti su izmjereni mikrometarskim vijkom. Mjerene su one veličine prikazane na slici 2.3:

- Poprečni presjek O-prstena,
- Unutarnji promjer O-prstena,
- Vanjski promjer O-prstena.

Poprečni presjek O-prstena je mjeran kao ulazni podatak u kalibracijsku funkciju optičkog mjernog sustava, a unutarnji i vanjski promjeri zbog mogućnosti usporedbe dobivenih rezultata. Kalibracijska funkcija je definirana u potpoglavlju 2.3.4. Postupak, mjerni instrumenti pomoću kojih su izvršena ručna mjerenja i rezultati istih prikazani su u potpoglavlju 4.1. Originalni mjerni listovi se nalaze u prilogu 4.

Kako bi se varijabilnosti mjernog sustava mogle usporediti s postavkama procesa, potrebno je znati tolerancije pri proizvodnji O-prstena. Tolerancije metričkih O-prstena opće namjene nalaze se u prilogu 6, a preuzete su iz [7]. Iz priloga je moguće očitati tolerancije unutarnjeg promjera i poprečnog presjeka. Nazivna mjera vanjskog promjera je sastavljena mjera i računa se prema:

$$\text{Vanjski promjer} = \text{Unutarnji promjer} + 2 \times \text{Poprečni presjek}, \quad (2.7)$$

Pomoću iste jednadžbe (2.7) računaju se i tolerancije vanjskog promjera.

Gornja i donja granica procesa se računaju pomoću nazivne mjere i tolerancije:

$$\begin{aligned} LSL &= \text{Nazivna mjera} - \text{Tolerancija}, \\ USL &= \text{Nazivna mjera} + \text{Tolerancija}, \end{aligned} \quad (2.8)$$

gdje je:

<i>USL</i> [mm]	Gornja granica procesa specificirana od strane klijenta (<i>Upper Specification Limit</i>),
<i>LSL</i> [mm]	Donja granica procesa specificirana od strane klijenta (<i>Lower Specification Limit</i>).

Ukupna širina tolerancijskog polja *T* je dobivena izrazom:

$$T = USL - LSL = 2 \times \text{Tolerancija}, \quad (2.9)$$

gdje je:

<i>T</i> [mm]	Ukupna širina tolerancijskog polja mjerene veličine.
---------------	--

Korištenjem jednadžbi (2.7) i (2.8), te tolerancija iz priloga 6, dobiveni su podaci o mjerenim veličinama O-prstena u procesu proizvodnje. Ti podaci su prikazani u tablici 2.1.

Tablica 2.1. Nazivne mjere i tolerancije mjerenih veličina O-prstena [7]

Mjereni veličina	Nazivna mjera [mm]	Tolerancije ± [mm]	LSL [mm]	USL [mm]
Poprečni presjek	2,4	0,08	2,32	2,48
Unutarnji promjer	15,8	0,23	15,57	16,03
Vanjski promjer	20,6	0,39	20,21	20,99

Zbog pojave deformacije O-prstena, čak i pri djelovanju malim silama, potrebna je posebna pažnja pri mjerenjima mikrometarskim vijkom.

2.2.2. Mjerene veličine optičkim putem

Mjerenje O-prstena optičkim putem spada u beskontaktno mjerne metode. Na taj način je eliminirana greška mjerenja zbog deformacije predmeta. Također, trajanje optičkog mjerenja je znatno kraće od mjerenja mikrometarskim vijkom, stoga se može izmjeriti veći udio proizvoda. Treća prednost optičkih mjerenja je mogućnost mjerenja znatno većeg broja veličina i istovremeno

vršenje mjerenja svih veličina. No, za kvalitetan optički mjerni sustav su potrebna znatna financijska ulaganja.

Veličine O-prstena, mjerene optičkim mjernim sustavom, posebno za unutarnji, a posebno za vanjski promjer, su sljedeće:

- Opseg (*Perimeter*),
- Maksimalni Feret-ov promjer (*Feret Max*),
- Minimalni Feret-ov promjer (*Feret Min*),
- Promjer okomit na Feret-ov minimalni promjer (*Feret PerpMin*),
- Kut maksimalnog Feret-ovog promjera (*Feret MaxAng*),
- Kut minimalnog Feret-ovog promjera (*Feret MinAng*),
- Maksimalni polumjer (*Radius Max*),
- Minimalni polumjer (*Radius Min*),
- Prosječni polumjer (*Radius Mean*),
- Standardna devijacija polumjera (*Radius StdDev*),
- Podček-ov oblik kruga (*PodcheckShapes Circle*),
- Podček-ov oblik elipse (*PodcheckShapes Ellipse*),
- Površina (*Size*),
- Konveksnost (*Convexity*),
- Konveksni opseg (*ConvexPerimeter*),
- Koordinate centra (*Center dim0, Center dim1*).

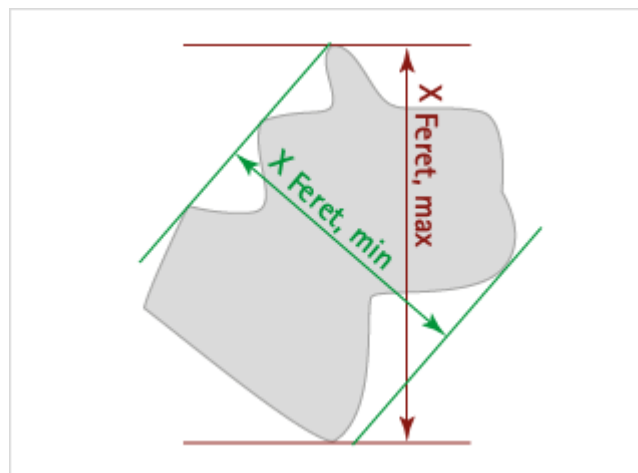
Gotovo sve navedene mjerene veličine se oslanjaju na precizno postavljanje lančanog koda (*chain code*) po unutarnjem, odnosno vanjskom promjeru O-prstena. Postupak postavljanja lančanog koda je opisan u [8]. Prvo se vrši pretraživanje vrhova po digitaliziranoj reprezentaciji promjera O-prstena u smjeru kazaljke na satu i obrnuto od smjera kazaljke na satu. Nakon toga se odabire početna točka, dodaju se lančani kodovi po obodu, te se na njih primjenjuje uvjet linearnosti. Točka lanca, koja ne zadovolji uvjet linearnosti, uzima se kao početna točka sljedećeg lančanog koda. Na taj način se dobiva reprezentacija vanjskog oboda predmeta pomoću poligona. Ukupna duljina poligona se računa kao suma duljina linearnih segmenata koje ga sačinjavaju. U [8] je također predstavljena metoda za procjenu duljine digitaliziranih linearnih segmenata uz što manju grešku.

Opseg (*Perimeter*) O-prstena je proračunat kao ukupna duljina lančanog koda po obodu, uzimajući u obzir udubljenja i izbočine [9].

Feret-ov promjer se često koristi pri procjeni dimenzija čestica nepravilnog oblika. Definiran je kao udaljenost između dva paralelna pravca koja dodiruju predmet u jednoj točki, prema slici 2.5. Posebno za vanjski, a posebno za unutarnji promjer O-prstena se mjeri sljedećih pet Feret-ovih vrijednosti [9]:

- Maksimalni Feret-ov promjer (*Feret Max*):
 - Najveća udaljenost između dva paralelna pravca koja dodiruju mjereni predmet.
- Minimalni Feret-ov promjer (*Feret Min*):
 - Najmanja udaljenost između dva paralelna pravca koja dodiruju mjereni predmet.
- Promjer okomit na Feret-ov minimalni promjer (*Feret PerpMin*):
 - Udaljenost između dva paralelna pravca, mjerena pod kutom od 90° u odnosu na kut minimalnog Feret-ovog promjera.
- Kut maksimalnog Feret-ovog promjera (*Feret MaxAng*):
 - Kut, u odnosu na x -os slike, pri kojem je izmjeren maksimalni Feret-ov promjer.
- Kut minimalnog Feret-ovog promjera (*Feret MinAng*):
 - Kut, u odnosu na y -os slike, pri kojem je izmjeren minimalni Feret-ov promjer.

Navedene mjerene veličine su posebno važne pri procjeni eliptičnosti O-prstena.



Slika 2.5. Skica maksimalnog (*Feret Max*) i minimalnog (*Feret Min*) Feret-ovog promjera [10]

Pri mjerenju O-prstena je vrlo bitno odrediti vrijednosti promjera. Zbog toga su optičkim putem mjerene sljedeće veličine [9]:

- Maksimalni polumjer (*Radius Max*),
- Minimalni polumjer (*Radius Min*),
- Prosječni polumjer (*Radius Mean*),
- Standardna devijacija polumjera (*Radius StdDev*).

Radijusi se mjere na način da se prvo odredi centroid, odnosno geometrijska sredina poligona opisanog lančanim kodom. Zatim se računaju udaljenosti od centroida do piksela na poligonu. Navedene izmjerene vrijednosti radijusa se, njihovim udvostručavanjem, lako mogu prikazati kao vrijednosti promjera. Pri analizi sposobnosti optičkog mjernog sustava i usporedbi s mjerenjima mikrometarskim vijkom, korištena je veličina prosječnog polumjera (*Radius Mean*).

Dodatne mjerene veličine za ocjenu kvalitete oblika O-prstena su:

- Podcbeck-ov oblik kruga (*PodcbeckShapes Circle*),
- Podcbeck-ov oblik elipse (*PodcbeckShapes Ellipse*).

Podcbeck-ovi oblici prikazuju sličnost mjenog oblika idealnom geometrijskom liku. Pri procjeni sličnosti idealnom liku se u obzir uzimaju vanjske dimenzije mjenog predmeta, ali i površinske nepravilnosti sadržane u reprezentaciji predmeta lančanim kodom [11]. Podcbeck-ovi oblici se ne mjere direktno, nego su izvedeni iz drugih izmjerenih vrijednosti. Podcbeck-ov oblik kruga (*PodcbeckShapes Circle*) je definiran izrazom [9]:

$$PodcbeckShapes Circle = \frac{4 \times Size}{\pi \times Feret PerpMin^2}, \quad (2.10)$$

gdje je:

<i>Size</i> [mm ²]	Površina poligona kao reprezentacije mjenog predmeta,
<i>Feret PerpMin</i> [mm]	Udaljenost između dva paralelna pravca, mjerena pod kutom od 90° u odnosu na kut minimalnog Feret-ovog promjera.

Podcbeck-ov oblik elipse je određen na način [9]:

$$PodcbeckShapes Ellipse = \frac{4 \times Size}{\pi \times Feret Min \times Feret PerpMin}, \quad (2.11)$$

gdje je:

Feret Min [mm] Najmanja udaljenost između dva paralelna pravca koja dodiruju mjereni predmet.

Iz jednadžbi (2.10) i (2.11) je vidljivo da su u slučaju idealnog kruga, odnosno idealne elipse vrijednosti Podczeck-ovog oblika jednake 1. Što su vrijednosti udaljenije od 1, to je oblik predmeta manje sličan idealnom geometrijskom liku. Navedena spoznaja se, osim u rezultatima mjerenja u prilogu 5, koristi i za identifikaciju O-prstena u algoritmu predobrade slike (potpoglavlje 2.4.15).

Kao još jedna mjera kvalitete oblika O-prstena se može uzeti njegova konveksnost (*Convexity*). U idealnom slučaju O-prsten je potpuno konveksan, odnosno ne postoje udubljenja na njegovom obodu. Konveksnost O-prstena je izvedena veličina i određuje se indirektno, prema izrazu [9]:

$$Convexity = \frac{ConvexPerimeter}{Perimeter}, \quad (2.12)$$

gdje je:

Convexity [/] Konveksnost mjenog predmeta,
ConvexPerimeter [mm] Ukupna duljina lančanog koda po obodu mjenog predmeta bez udubljenja,
Perimeter [mm] Ukupna duljina lančanog koda po obodu mjenog predmeta, uzimajući u obzir udubljenja i izbočine.

Vrijednost konveksnosti mjenog predmeta se može kretati u intervalu $0 \div 1$, gdje se 1 odnosi na idealno konveksan predmet.

Posljednje mjerene veličine optičkim putem su koordinate centroida (*Center dim0*, *Center dim1*), odnosno geometrijskog težišta O-prstena. Pošto se mjere koordinate centra posebno za unutarnji promjer, a posebno za vanjski promjer O-prstena, moguće je izračunati njihovu međusobnu ekscentričnost:

$$Eccentricity = \sqrt{(Center\ dim0_{out} - Center\ dim0_{in})^2 + (Center\ dim1_{out} - Center\ dim1_{in})^2}, \quad (2.13)$$

gdje je:

Eccentricity [mm] Ekscentričnost unutarnjeg i vanjskog promjera O-prstena,
Center dim0_{out} [mm] Koordinate centra vanjskog promjera O-prstena po *x*-osi,
Center dim0_{in} [mm] Koordinate centra unutarnjeg promjera O-prstena po *x*-osi,

$Center\ dim1_{out}$ [mm]	Koordinate centra vanjskog promjera O-prstena po y -osi,
$Center\ dim1_{in}$ [mm]	Koordinate centra unutarnjeg promjera O-prstena po y -osi.

Ekscentričnost O-prstena ukazuje na višak, odnosno manjak materijala na pojedinim dijelovima O-prstena.

2.3. Predloženi optički mjerni sustav

Kao što je navedeno u potpoglavlju 2.1, svaki sustav strojnog vida, pa tako i optički mjerni sustav se sastoji od hardvera i pripadajuće programske podrške. Programska podrška sustava, bazirana na rješenjima otvorenog koda, predstavljena je u poglavlju 3. Hardverske komponente sustava, prikazane na slici 2.6, su sljedeće:

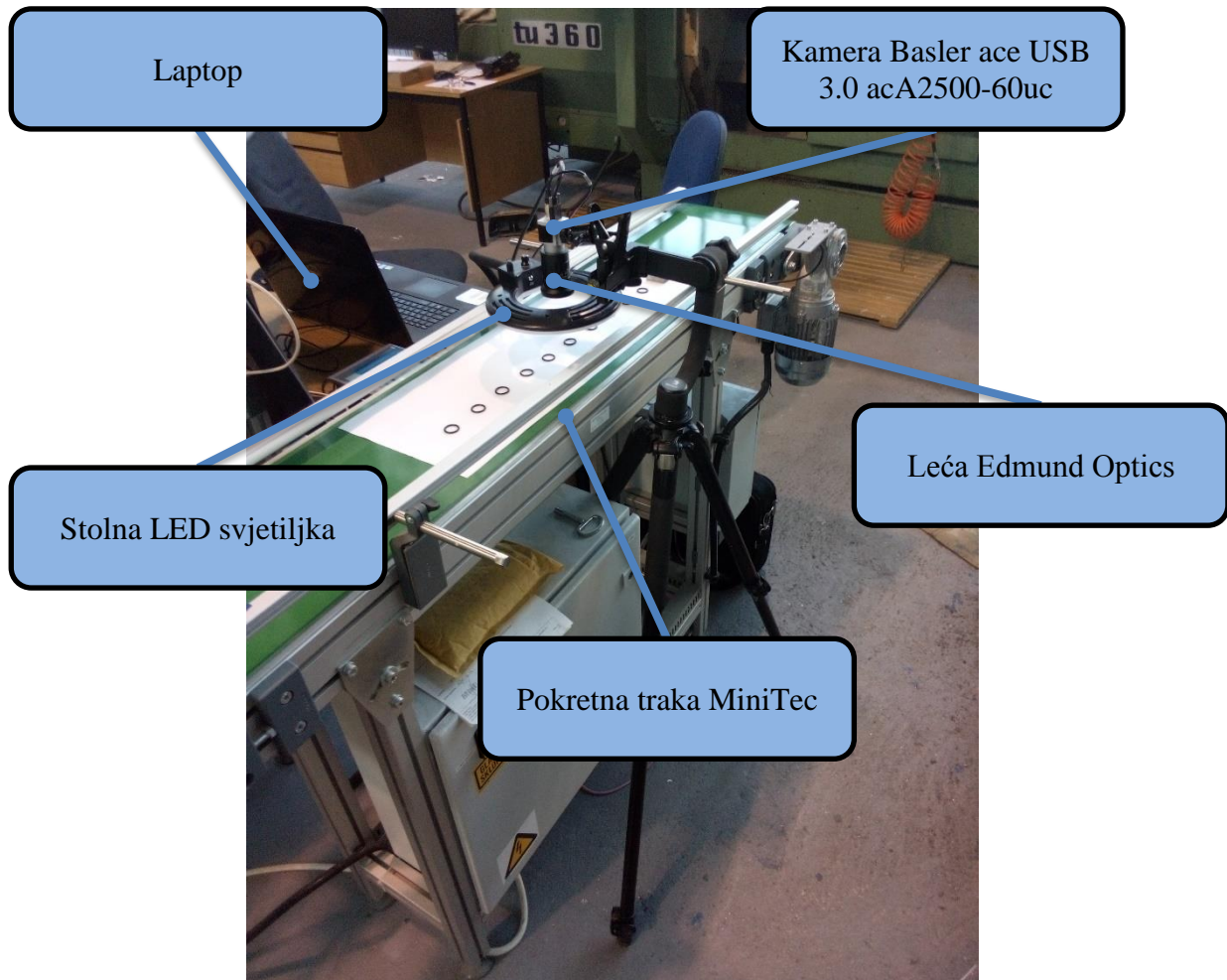
- Kamera Basler ace USB 3.0 acA2500-60uc,
- Leća Edmund Optics fiksne žarišne duljine 16 mm,
- Stolna LED svjetiljka u obliku prstena,
- Modularna pokretna traka MiniTec,
- Laptop + Ubuntu operativni sustav + programska rješenja otvorenog koda.

Uz navedene komponente, potrebno je napomenuti da je kao podloga za O-prsten korišten čisti bijeli papir. To je bilo nužno kako bi se povećao kontrast između mjerenog predmeta i podloge, što je rezultiralo preciznijim rezultatima mjerenja.

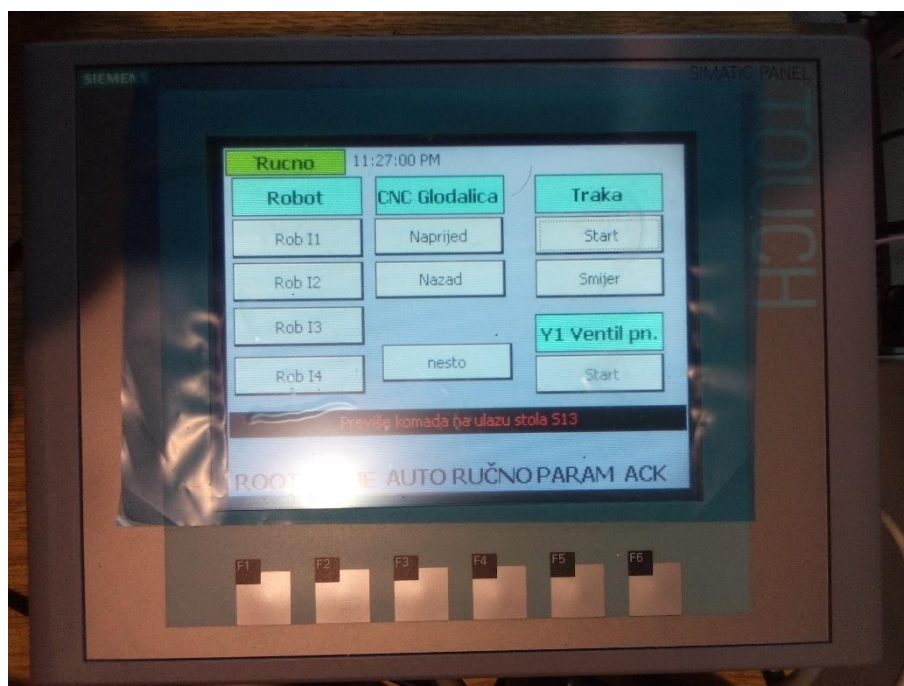
O-prsteni se pomoću pokretne trake MiniTec pokreću kroz vidno polje kamere, dobivene slike se digitalno obrađuju i vrši se mjerenje. Pokretna traka nije integrirana u automatski mjerni sustav, odnosno ne pomiče se pomoću programske podrške sustava, nego pomoću Siemens SIMATIC programabilnog logičnog kontrolera (*Programmable logic controller*, PLC). PLC je prikazan na slici 2.7. Pokretna traka se pomiče pritiskom gumba *Start*, a smjer joj se mijenja pritiskom gumba *Smjer*.

2.3.1. Kamera Basler ace USB 3.0 acA2500-60uc

Samo prikupljanje slika za mjerenje se vrši pomoću kamere Basler ace USB 3.0 acA2500-60uc. Postupak formiranja digitalne slike je opisan u potpoglavlju 2.1.1. Navedena kamera pripada tipu kamera s površinskim senzorom, tj. osjetni elementi (pikseli) su raspoređeni u dvodimenzionalnu mrežu. Rezultat izlaganja senzora svjetlosti je dvodimenzionalna slika.



Slika 2.6. Komponente optičkog mjernog sustava



Slika 2.7. Siemens SIMATIC PLC kontroler za upravljanje pokretnom trakom

Karakteristike navedene kamere su sljedeće [12]:

- Priključak za montiranje objektiva: *C-Mount*
- Potrebna snaga:
 - ≈ 3.8 W pri 5 VDC (volti istosmjerne struje),
 - ≈ 4.1 W (maksimalno),
- Razlučivost: 2592 x 2048 piksela,
- Tip senzora: CMOS senzor s tehnologijom progresivnog skeniranja (*Progressive scan*),
- Format senzora: 1",
- Dimenzije svakog piksela: 4.8 μm x 4.8 μm ,
- Broj slika u sekundi:
 - 60 FPS (pri brzom načinu očitavanja sa senzora),
 - 47 FPS (pri normalnom načinu očitavanju sa senzora),
- Dimenzije (duljina x širina x visina):
 - 29.3 mm x 29 mm x 29 mm (bez objektiva),
 - 48.2 mm x 29 mm x 29 mm (s objektivom),
- Težina: 80 g,
- Sučelje za prijenos podataka i napajanje: USB 3.0, maksimalna brzina prijenosa 5 Gbit/s,
- Temperatura kućišta tijekom rada: 0 ÷ 50 °C,
- Popratna programska podrška: *Basler pylon Camera Software Suite* [13].

Potrebno se pridržavati uputa proizvođača u vezi s montiranjem i održavanjem kamere, te osiguranjem radnih uvjeta i pravilnog odvođenja topline prema [12].

CMOS senzori pružaju brojne prednosti u odnosu na standardne CCD senzore, poput mogućnosti očitavanja proizvoljnog područja interesa, velikog raspona radne temperature, niske potrošnje energije i niskih proizvodnih troškova. Tehnologija progresivnog skeniranja omogućuje očitavanje svih linija piksela senzora sekvencijalno, što rezultira manjom distorzijom predmeta u gibanju na slici.

Programsko rješenje, opisano u poglavlju 3, omogućuje podešavanje raznih parametara kamere [12]:

MaxNumBuffer

Maksimalan broj međuspremnik za istovremeno pohranjivanje slika,

<i>AcquisitionFrameRate</i>	Gornji limit u FPS,
<i>PixelFormat</i>	Format digitalizirane izlazne slike,
<i>Gain</i> [dB]	Digitalno pojačanje izlaznog signala, odnosno pojačanje osvjetljenja,
<i>ExposureTime</i> [ms]	Vremensko trajanje izlaganja senzora svjetlosti tijekom prikupljanja slika,
<i>Width</i> [px]	Duljina izlazne slike (ROI) u pikselima,
<i>Height</i> [px]	Visina izlazne slike (ROI) u pikselima,
<i>OffsetX</i> [px]	Pomak ROI po <i>x</i> -osi senzora u pikselima,
<i>OffsetY</i> [px]	Pomak ROI po <i>y</i> -osi senzora u pikselima,

Osnovni parametri, između kojih je potrebno pronaći kompromis kod mjerenja u stvarnom vremenu, su digitalno pojačanje signala (*Gain*) i trajanje izlaganja senzora svjetlosti (*Exposure time*). Povećavanje oba parametra rezultira svjetlijom slikom. Dulje vrijeme izlaganja svjetlosti povećava rizik od distorzije promatranog predmeta zbog gibanja. Također, u sustavima strojnog vida visoke propusne moći je to vrijeme ograničeno. Digitalno pojačanje signala jednostavno povećava intenzitet cijelog signala, uključujući i „šum“ kao nuspojavu. Tablica 2.2 prikazuje vrijednosti parametara kamere korištene za dobivanje rezultata mjerenja optičkim putem. Programsko rješenje je ograničeno na 50 FPS-a. Slike su prilikom prikazivanja pretvorene u 8-bitni format, tj. nijanse sive boje (postavka *Mono8*), a svaki piksel senzora je izložen svjetlosti 9,5 ms. Pri mjerenju su korištene slike maksimalne moguće rezolucije.

Tablica 2.2. Postavke parametara kamere korištene pri mjerenju

Parametar	Vrijednost
<i>MaxNumBuffer</i>	10
<i>AcquisitionFrameRate</i>	50 FPS
<i>PixelFormat</i>	Mono8
<i>Gain</i>	0 dB
<i>ExposureTime</i>	9,5 ms
<i>Width</i>	2592 px
<i>Height</i>	2048 px
<i>OffsetX</i>	0 px
<i>OffsetY</i>	0 px

2.3.2. Objektiv Edmund Optics fiksne žarišne duljine 16 mm

Leće se, kako je navedeno u potpoglavlju 2.1.1, koriste za preslikavanje točki sa promatranog predmeta na senzor. U idealnom slučaju, svaka točka promatranog predmeta **G** je mapirana u točno jednu točku **B** na senzoru. U tom slučaju snop divergirajućih svjetlosnih zraka iz točke **G**, nakon prolaska kroz leću, konvergira u samo jednu točku **B** na senzoru, kao što je prikazano na slici 2.8. Leće s takvim principom rada se nazivaju sabirne leće. U realnosti ne postoje idealne leće, već samo približno idealne. Leće su rotacijski simetrični predmeti zakrivljenog oblika, uglavnom izrađeni od stakla. Os rotacije leće naziva se optička os.

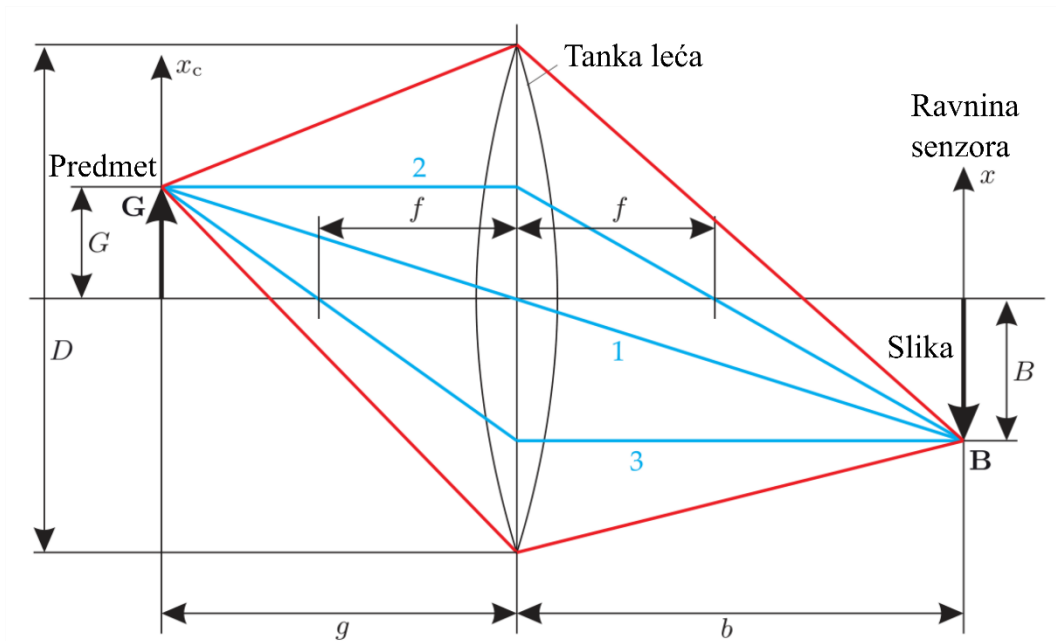
Na slici 2.8 su prisutne sljedeće oznake [6]:

B [/]	Točka senzora optičkog sustava,
G [/]	Točka promatranog predmeta optičkog sustava,
<i>G</i> [mm]	Udaljenost točke promatranog predmeta od optičke osi leće,
<i>B</i> [mm]	Udaljenost točke senzora od optičke osi leće,
<i>D</i> [mm]	Promjer leće,
<i>g</i> [mm]	Udaljenost promatranog predmeta leće,
<i>b</i> [mm]	Udaljenost senzora od leće,
<i>f</i> [mm]	Žarišna udaljenost leće.

Kako bi princip rada leće sa slike bio primjenjiv, žarišna udaljenost leće mora biti mnogo veća od njezine debljine. Zato se leća naziva tankom lećom. Žarišna udaljenost leće *f* je karakteristika same leće, a interpretira se kao udaljenost žarišne točke od leće u kojoj svjetlosne zrake, koje upadaju okomito u odnosu na leću, sijeku žarišnu os. Primjer takve zrake je označen plavom bojom i brojem 2 na slici 2.8. Primjenom sličnosti trokuta, može se dobiti sljedeći odnos [6]:

$$\frac{G}{g} = -\frac{B}{b}, \quad (2.14)$$

Negativan predznak u jednadžbi (2.14) znači da je na dobivenoj slici predmet okrenut naopako.



Slika 2.8. Princip rada tanke leće [6]

Još jedna važna karakteristika optičkog sustava je odnos veličine predmeta na slici i stvarne veličine promatranog predmeta [6]:

$$V = \frac{B}{G} = -\frac{b}{g}, \quad (2.15)$$

gdje je:

V [/] Odnos veličine predmeta na slici i stvarne veličine promatranog predmeta.

Iz jednadžbe (2.15) je vidljivo da je veličina predmeta na slici ovisna o njegovoj udaljenost od leće g . Što je veća udaljenost g , to je predmet manji na slici, a vrijedi i obrnuto.

Također, važno je napomenuti da se u praksi koriste optički sustavi sačinjeni od više leća, no prethodno navedene relacije vrlo vjerodostojno opisuju i takve sustave.

Za potrebe optičkog mjernog sustava korišten je objektiv Edmund Optics sljedećih karakteristika [14]:

- Priključak za montiranje: *C-Mount*
- Žarišna udaljenost: 16 mm,
- Otvor objektiva: $f/1,8 \div f/16$,
- Optimizirana radna udaljenost: 400 ÷ 2000 mm,
- Radna udaljenost: > 100 mm,

- Vidno polje: 101,3 mm – 43,2 °,
- Maksimalna duljina: 60,4 mm,
- Minimalna duljina: 57,8 mm,
- Vanjski promjer: 38,8 mm
- Maksimalni podržani format senzora: 1",
- Distorzija: < 3,5%,
- Broj leća (grupa): 9 (7).

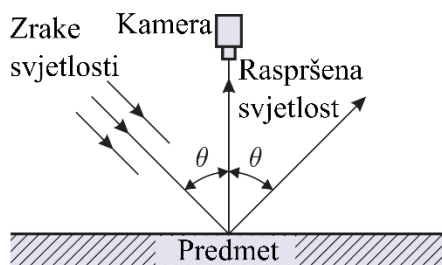
Nacrt navedenog objektiva nalazi se u prilogu 7. Objektiv je predviđen za rad na minimalnoj radnoj udaljenosti od 100 mm. Navedena udaljenost se računa od prednjeg dijela objektiva do mjerenog predmeta. Na manjim udaljenostima od navedene nije moguće dobiti oštru sliku predmeta. Objektivu je moguće prilagoditi otvor i fokus zakretanjem dva prstena na kućištu objektiva, a zadovoljavajuće vrijednosti se mogu zaključati pomoću dva vijka.

Otvor objektiva i udaljenost promatranog predmeta određuju dubinsku oštrinu (*Depth of field*), vrlo važan parametar optičkih sustava. Dubinska oštrina predstavlja zonu udaljenosti od objektiva, u kojoj su predmeti prikazani prihvatljivom oštrinom. Ta zona može biti plitka (malo područje) i duboka (veliko područje). U opisanom mjernom sustavu bitno je da se svi predmeti od interesa, tj. O-prsteni i etaloni, nalaze u toj zoni. Otvor objektiva diktira količinu propuštene svjetlosti do senzora. Što je veći otvor objektiva, više svjetlosti dolazi do senzora, odnosno izlazna slika je svjetlija. Vrijedi i obrnuto. Iznos otvora objektiva se određuje relativno u odnosu na žarišnu udaljenost, tj. $f/1,8$ podrazumijeva veliki otvor, a $f/16$ mali otvor. Utjecaj otvora objektiva na dubinsku oštrinu je obrnuto proporcionalan, odnosno povećavanjem otvora se smanjuje zona dubinske oštrine, a smanjenjem otvora povećava.

Udaljenost predmeta od objektiva također utječe na dubinsku oštrinu. Što je predmet bliže objektivu, zona dubinske širine je plića i obrnuto. Uzimajući sve navedeno u obzir, prilikom postavljanja optičkog sustava, potrebno je pronaći optimalne vrijednosti otvorenosti objektiva, udaljenosti predmeta i parametara kamere, prema poglavlju 2.3.1. Rezultat postavljanja sustava mora biti zadovoljavajuća oštrina slike, zadovoljavajuća veličina predmeta na slici (zbog uočavanja potrebnih značajki) i zadovoljavajuće osvjetljenje izlazne slike (zbog kontrasta između predmeta od interesa i pozadine).

2.3.3. Stolna LED svjetiljka u obliku prstena

Optimalno osvjetljenje je jedna od bitnih pretpostavki kvalitetnog sustava strojnog vida. U optičkom mjernom sustavu je korišteno osvjetljenje tamnog polja, prikazano na slici 2.9. U takvom sustavu izvor svjetlosti nije usmjeren direktno u kameru, nego pod određenim kutom θ . Jedino svjetlost raspršena od predmeta promatranja dolazi do senzora kamere.



Slika 2.9. Shema osvetljenja tamnog polja [6]

Osvjetljenjem tamnog polja se mogu postići visoki kontrasti između promatranog predmeta i pozadine [6]. Sa slike 2.6 je vidljiv prstenasti oblik svjetiljke, odabran kako bi svjetlost upadala iz svih kutova. Na taj način je osigurano osvjetljenje unutrašnjeg i vanjskog promjera O-prstena.

LED svjetiljke su okarakterizirane visokom trajnošću, opadajućim intenzitetom svjetlosti tijekom životnog vijeka, te otpornošću na udarce i vibracije. Također, sva svjetlost je emitirana u uskom vidljivom spektru, što rezultira visokim učinkom.

Karakteristike korištene svjetiljke su sljedeće:

- Boja osvetljenja: 5500 K,
- Snaga: 24 W,
- Broj dioda: 120,
- Svjetlosni tok: 2500 lm,
- Mogućnost prigušenja svjetlosti.

2.3.4. Kalibracija optičkog mjernog sustava

Prije svakog mjerenja, optički mjerni sustav je potrebno kalibrirati. Standardan pristup kalibraciji, kod dimenzijskih mjerenja, je određivanje veličine svakog piksela slike u mjernim jedinicama za duljinu. To su najčešće milimetri, kao u slučaju predloženog optičkog sustava, odnosno mikrometri u slučaju optičke mikroskopije. Konačni rezultat kalibracije je mm/px omjer vezan

samo za slike prikupljene u istim uvjetima u kojima je izvršena kalibracija. Ukoliko se na slici 2.8 udaljenost točke senzora od optičke osi B prikaže u pikselima, rezultat kalibracije odgovara izrazu:

$$mm/px \text{ omjer} = \frac{G}{B}, \quad (2.16)$$

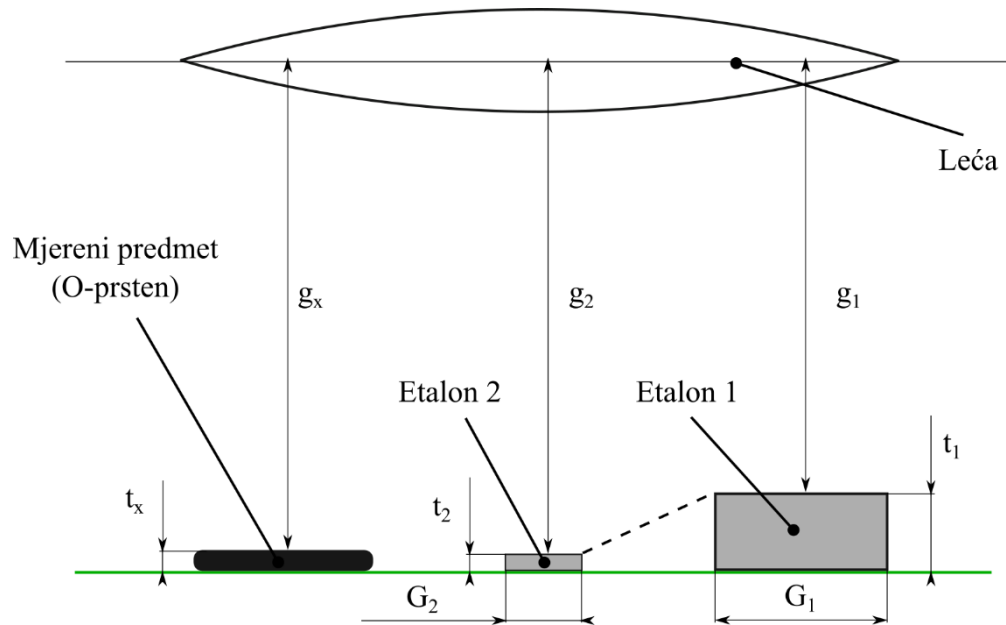
gdje je:

$mm/px \text{ omjer}$ [mm/px]	Veličina jednog piksela slike u mjernim jedinicama milimetra,
G [mm]	Udaljenost točke promatranog predmeta od optičke osi leće,
B [px]	Udaljenost točke senzora od optičke osi leće u pikselima.

Uzimajući u obzir jednadžbe (2.15) i (2.16), može se zaključiti da i $mm/px \text{ omjer}$ ovisi o udaljenosti predmeta od objektiva g . Što je veća udaljenost predmeta od objektiva, to je slika na senzoru manja, odnosno veličina jednog piksela u milimetrima je manja. Iz navedenog se može zaključiti da razlučivost kamere ima velik utjecaj na rezultate mjerenja i da je mjereni predmet potrebno držati što bliže objektivu.

Pošto se $mm/px \text{ omjer}$ mijenja ovisno o udaljenosti predmeta od leće, osmišljen je način kalibracije sustava pomoću dva etalona. Etalonima su unaprijed poznate sve potrebne dimenzije, a odabiru se na način da je visina jednog etalona niža od visine mjenenog predmeta, a visina drugog etalona viša od iste veličine mjenenog predmeta. Mjerodavna veličina O-prstena je njihov poprečni presjek. Na slici 2.10 su prikazane sve potrebne veličine etalona i mjenenih predmeta za kalibraciju optičkog sustava:

g_1 [mm]	Udaljenost od leće do etalona 1,
g_2 [mm]	Udaljenost od leće do etalona 2,
g_x [mm]	Udaljenost od leće do mjenenog predmeta (O-prstena),
t_1 [mm]	Visina etalona 1,
t_2 [mm]	Visina etalona 2,
t_x [mm]	Visina mjenenog predmeta (O-prstena),
G_1 [mm]	Mjerena veličina etalona 1 unaprijed poznate vrijednosti,
G_2 [mm]	Mjerena veličina etalona 2 unaprijed poznate vrijednosti.



Slika 2.10. Kalibracija optičkog mjernog sustava

Uz sve navedene veličine na slici, potrebno je uvesti još tri:

- B_1 [px] Mjerena veličina etalona 1 preslikana na senzor (sliku),
- B_2 [px] Mjerena veličina etalona 2 preslikana na senzor (sliku),
- B_x [px] Mjerena veličina predmeta preslikana na senzor (sliku).

Navedene dvije veličine (B_1 i B_2) je potrebno izmjeriti na slici za kalibraciju pomoću odgovarajuće programske podrške, te ih iskazati u pikselima.

Udaljenost senzora od leće b je konstantna u slučaju mjerenja etalona 1 i etalona 2 prema slici 2.10. Prema jednadžbi (2.14), mogu se postaviti odnosi posebno za etalon 1, a posebno za etalon 2:

$$\frac{G_1}{g_1} = -\frac{B_1}{b}, \quad \frac{G_2}{g_2} = -\frac{B_2}{b}. \quad (2.17)$$

Iz relacije (2.17) se može izraziti udaljenost senzora od leće:

$$b = -\frac{g_1 B_1}{G_1} = -\frac{g_2 B_2}{G_2}. \quad (2.18)$$

Daljnjim sređivanjem slijedi:

$$\frac{G_2}{B_2} = \frac{g_2}{g_1} \times \frac{G_1}{B_1}. \quad (2.19)$$

Sa slike 2.10 je vidljiv odnos između visina etalona i udaljenosti od leće do etalona:

$$g_1 + t_1 = g_2 + t_2. \quad (2.20)$$

Izlučivanjem udaljenosti od leće do etalona 2 iz jednadžbe (2.20) se dobiva:

$$g_2 = g_1 + t_1 - t_2. \quad (2.21)$$

Nakon što se izraz (2.21) uvrsti u (2.19), dobiva se:

$$\frac{G_2}{B_2} = \frac{g_1 + t_1 - t_2}{g_1} \times \frac{G_1}{B_1} = \left(1 + \frac{t_1 - t_2}{g_1}\right) \times \frac{G_1}{B_1}. \quad (2.22)$$

Sređivanjem izraza (2.22) se dobiva:

$$\frac{\frac{G_2}{B_2}}{\frac{G_1}{B_1}} = 1 + \frac{t_1 - t_2}{g_1}. \quad (2.23)$$

Iz jednadžbe (2.23) se može izraziti udaljenost od leće do etalona 1:

$$g_1 = \frac{t_1 - t_2}{\frac{\frac{G_2}{B_2}}{\frac{G_1}{B_1}} - 1}. \quad (2.24)$$

Pomoću relacije (2.24) se može direktno izračunati g_1 pošto su visine (t_1 i t_2) i mjerene veličine (G_1 i G_2) etalona unaprijed poznate, a vrijednosti veličina na senzoru (B_1 i B_2) su izmjerene. Udaljenost g_1 je u potpoglavlju 3.2 označena kao *Working distance* etalona.

Ako se u izrazu (2.22) umjesto indeksa drugog etalona (2) ubaci indeks mjerenog predmeta (x) proizvoljne visine, dobiva se odnos:

$$\frac{G_x}{B_x} = \left(1 + \frac{t_1 - t_x}{g_1}\right) \times \frac{G_1}{B_1}. \quad (2.25)$$

Uvrštavanjem jednadžbe (2.16) u (2.25), dobiva se kalibracijska funkcija:

$${}^{mm}/{}_{px} \text{ omjer}(t_x) = \left(1 + \frac{t_1 - t_x}{g_1}\right) \times {}^{mm}/{}_{px} \text{ omjer}_1, \quad (2.26)$$

gdje je:

${}^{mm}/{}_{px} \text{ omjer}(t_x)$ [mm/px]	Kalibracijska funkcija u ovisnosti o visini mjerenog predmeta,
${}^{mm}/{}_{px} \text{ omjer}_1$ [mm/px]	Veličina jednog piksela slike u mjernim jedinicama milimetra, na visini etalona 1.

Kalibracijska funkcija je zapravo interpolacija ${}^{mm}/{}_{px} \text{ omjer}$ vrijednosti između etalona 1 i etalona 2 na proizvoljnoj visini t_x . Ovakvim načinom kalibracije je moguće mjeriti veličine predmeta neovisno o udaljenosti od objektiva. Isto tako je moguće mjeriti veličine predmeta različitih visina, uz preduvjet poznavanja tih visina prije mjerenja.

U predloženom optičkom mjernog sustavu su, prema tablici 4.1, mjereni O-prsteni poprečnog presjeka 2,397 mm. To je ulazna veličina t_x kalibracijske funkcije. Za određivanje ${}^{mm}/{}_{px} \text{ omjer}$ -a korišteni su sljedeći čelični etaloni, prikazani na slici 2.11:

- Etalon 20 mm (etalon 1):
 - Visina: $t_1 = 9,010$ mm,
 - Mjerena veličina: $G_1 = 20,000$ mm.
- Etalon 2 mm (etalon 2):
 - Visina: $t_2 = 2,000$ mm,
 - Mjerena veličina: $G_2 = 8,950$ mm.

Etaloni su za kalibraciju pripremljeni na način da su približno paralelno položeni na bijelu papirnatu podlogu i obojeni crnim marker.



Slika 2.11. Etaloni 20 mm (lijevo) i 2 mm (desno) pripremljeni za kalibraciju sustava

Pri mjerenju veličina B_1 i B_2 u pikselima, korišten je minimalni Feret-ov promjer opisan u potpoglavlju 2.2.2. U praksi, zbog raznih utjecajnih čimbenika, dolazi do greške prilikom mjerenja navedenih veličina. Greška u umjeravanju sustava može znatno utjecati na rezultat mjerenja, kako je pokazano u potpoglavlju 4.5. Stoga je praktična preporuka da mjerene veličine etalona G_1 i G_2 budu što veće, odnosno zauzimaju velik dio slike na temelju koje se kalibrira sustav. Apsolutna greška kod mjerenja B_1 i B_2 je približno jednaka, bez obzira mjere li se veće ili manje veličine, no dijeljenje prema izrazu (2.16) unosi znatno manju grešku ako se i u brojniku i u nazivniku nalaze veće vrijednosti.

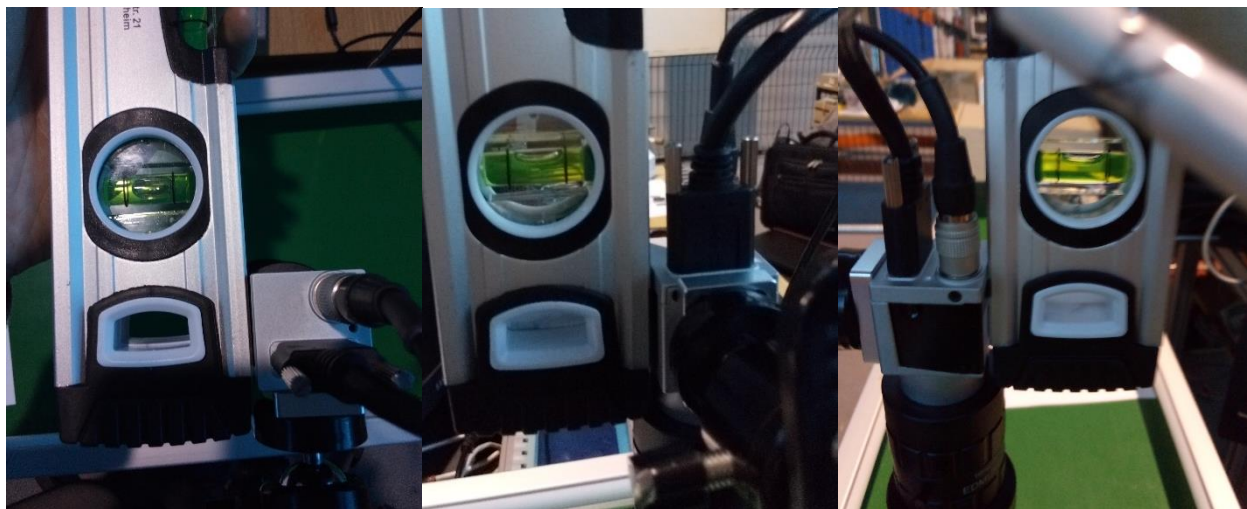
2.3.5. Postavljanje optičkog sustava

Prilikom postavljanja optičkog sustava potrebno je pratiti sljedeće korake:

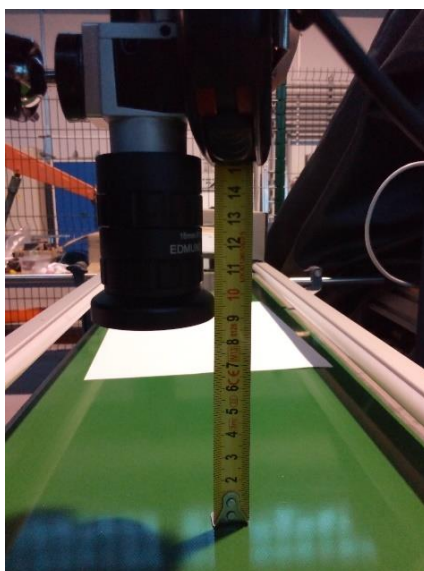
1. Provjeriti komponente sustava zbog mogućih oštećenja,
2. Provjeriti vodoravnost pokretne trake libelom, prema slici 2.12,
3. Postaviti tripod, na njega montirati kameru, te ju spojiti na računalo pomoću USB 3.0 sučelja,
4. Namjestiti kameru na način da optička os kamere bude okomita na pokretnu traku, kako prikazuje slika 2.13,
5. Postaviti kameru tako da udaljenost od objektiva do pokretne trake bude ~ 94 mm, prema slici 2.14,
6. Rotirati prstene na kućištu objektiva dok otvor objektiva ne bude postavljen na $f/5,6$, a fokus na 0,1 m. Navedeni parametri kamere pružaju odgovarajuću dubinu oštine i osvjetljenje slike. Učvrstiti dva vijka na kućištu da ne dođe do slučajnog pomicanja prstena.



Slika 2.12. Provjera horizontalnosti pokretne trake libelom



Slika 2.13. Namještanje pozicije kamere libelom



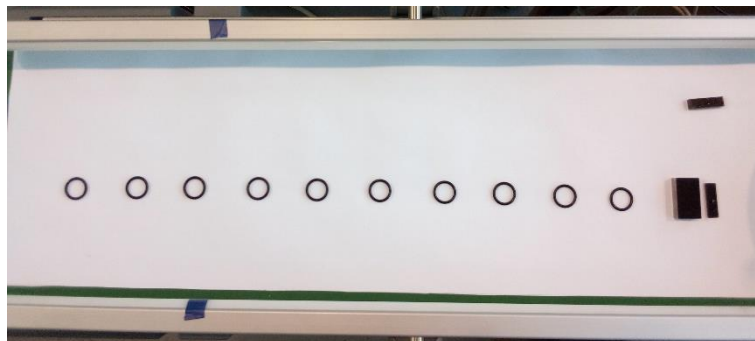
Slika 2.14. Postavljanje kamere na pravilnu radnu udaljenost (Working distance)

7. Učvrstiti LED svjetiljku za radni stol i postaviti prsten oko kamere prema slici 2.15. Udaljenost svjetiljke od pokretne trake treba biti ~ 75 mm,



Slika 2.15. Učvršćivanje LED svjetiljke i postavljanje prstena oko kamere

8. Na pokretnu traku položiti čisti bijeli papir, te na njega poslagati O-prstene i etalone u približno jednakim razmacima, kako je prikazano na slici 2.16. O-prsteni i etaloni bi se trebali kretati što bliže sredini slike tijekom kalibriranja i mjerenja.



Slika 2.16. Postavljanje bijele podloge, te priprema O-prstena i etalona

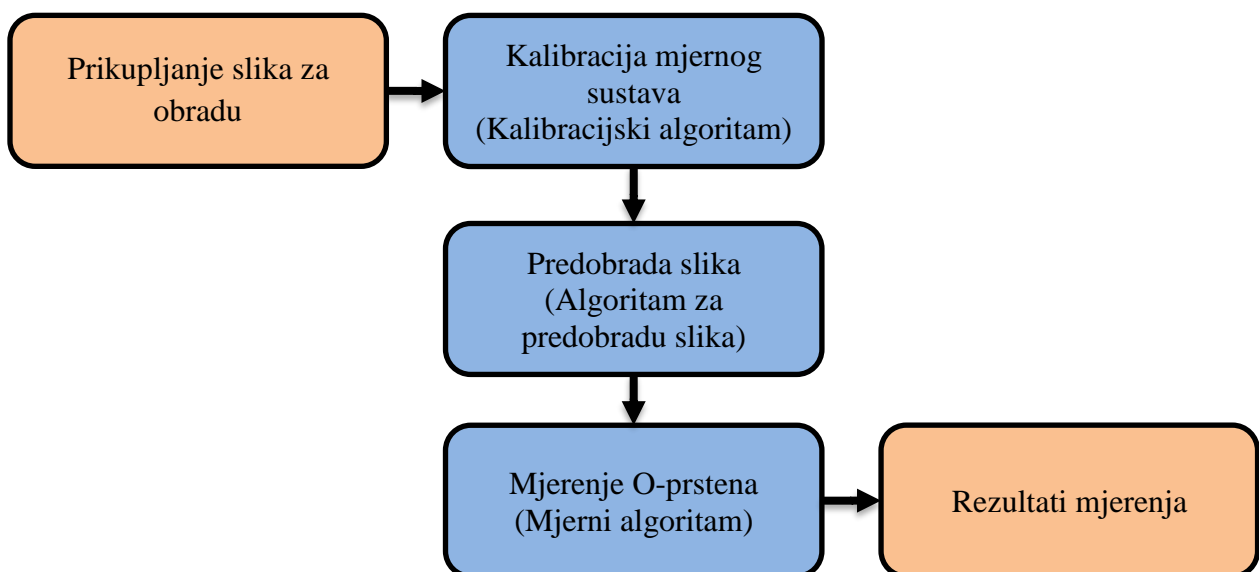
9. Upaliti LED svjetiljku, pokrenuti programsku podršku mjernom sustavu i pripremiti PLC kontroler za upravljanje pokretnom trakom, prikazan na slici 2.7.
10. Nastaviti postupak kalibracije i mjerenja, kao što je opisano u poglavlju 3.

2.4. Digitalna obrada slike

Digitalna obrada slike je područje računarske znanosti koje se bavi razvojem i implementacijom algoritama i metoda za obradu digitalnih slika. Digitalna obrada slika je bazirana na području digitalne obrade signala. Kao što je opisano u potpoglavlju 2.1.1, analogni signal slike $g(\mathbf{x})$ se pretvara u digitalni g_{mn} prostornom diskretizacijom (uzorkovanjem) i vrijednosnom diskretizacijom (kvantiziranjem). U sljedećih nekoliko potpoglavlja su ukratko pojašnjeni linearni i nelinearni algoritmi za filtriranje, segmentacijski algoritmi, algoritmi za filtriranje binarnih slika, algoritmi za obilježavanja područja slike, algoritmi za mjerenja i algoritmi za dodjeljivanje mjernih jedinica pikselima. Implementacije navedenih algoritama se preuzete iz [9]. Uz navedene standardne algoritme digitalne obrade slika, izrađeni su i algoritmi prilagođeni problemu mjerenja O-prstena, poput:

- Algoritma za dobivanje konstanti kalibracijske funkcije prema jednadžbi (2.26),
- Algoritma za segmentaciju unutarnjeg i vanjskog promjera O-prstena,
- Algoritma za predobradu slika.

Svi navedeni algoritmi se koriste u programskoj podršci mjernog sustava kao sastavni dijelovi koraka za izvršavanje optičkih mjerenja. Na slici 2.17 su prikazani spomenuti koraci uz pripadajuće algoritme programske podrške. Detaljniji dijagrami tokova pojedinih koraka objašnjeni su u poglavlju 3. Sastavni algoritmi složenijih algoritama programske podrške, uz njihovu namjenu, prikazani su u tablici 2.3. Algoritam za predobradu slika je pojašnjen u potpoglavlju 2.4.15.



Slika 2.17. Koraci i algoritmi potrebni za izvršavanje optičkih mjerenja O-prstena

Tablica 2.3. Algoritmi programske podrške raščlanjeni na sastavne algoritme

Namjena algoritma	Kalibracijski algoritam	Mjerni algoritam
Filtriranje	<i>Gaussian Convolution</i>	<i>Kuwahara</i>
	<i>Kuwahara</i>	<i>Gradient Magnitude</i>
	<i>Gradient Magnitude</i>	
Segmentacija	<i>Minima</i>	<i>Minima</i>
	<i>Seeded Watershed</i>	<i>Segment O-ring</i>
		<i>Seeded Watershed</i>
Filtriranje binarnih slika	<i>Remove Edge Objects</i>	<i>Remove Edge Objects</i>
	<i>Fill Holes</i>	<i>Binary Area Closing</i>
Obilježavanje područja	<i>Label</i>	<i>Label</i>
Kalibracija i primjena kalibracije	<i>Working Distance Correction</i>	<i>Set Pixel Size</i>
Mjerenje	<i>Measure</i>	<i>Measure</i>
Vizualna kontrola rezultata	<i>Apply Mask</i>	<i>Apply Mask</i>

Svim algoritmima iz tablice 2.3 su u nastavku pojašnjeni ulazni parametri i prikazani primjeri upotrebe.

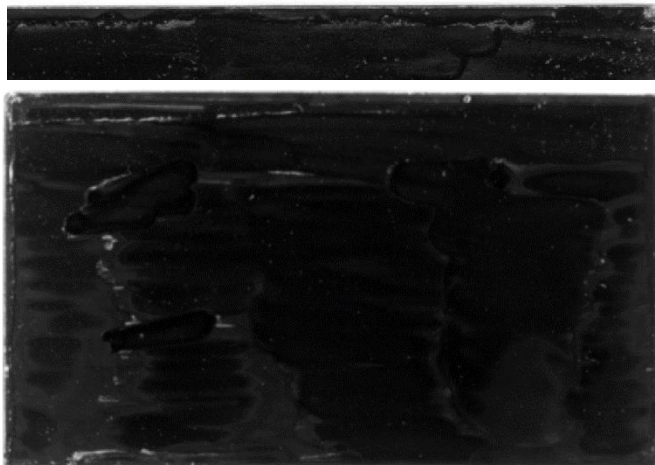
2.4.1. Algoritam *Gaussian Convolution*

Algoritam *Gaussian Convolution* pripada grupi linearnih filtera, a koristi se za zamaglivanje slike. Zamaglivanje slike je uglavnom prvi korak u postupku mjerenja predmeta optičkim putem. Ono se koristi kako bi se uklonio „šum“ sa rubova mjerenog predmeta i dobile jasnije konture [15].

Za zamaglivanje se koristi Gaussov *kernel*, odnosno strukturni element zvonastog oblika. *Kernel* prelazi preko slike mijenjajući vrijednosti pojedinih piksela pomoću susjednih piksela. Pošto je *kernel* zvonastog oblika, najveći utjecaj na izlaznu vrijednost piksela imaju njegovi neposredni susjedi. U programskom rješenju je korištena implementacija iz [9], sa sljedećim parametrima:

sigmas Standardna devijacija normalne raspodjele, koja određuje izgled *kernela*, odnosno način zamaglivanja slike.

Primjer upotrebe algoritma za zamaglivanje etalona je prikazan na slici 2.18. U gornjem dijelu slike se nalazi izoštren predmet, a u donjem zamagljen. Parametri algoritma korišteni pri kalibraciji prikazani su na slici 3.12.



Slika 2.18. Primjer algoritma *Gaussian Convolution* (dolje) na mjernom etalonu

2.4.2. Algoritam *Kuwahara*

Algoritam *Kuwahara* isto služi za zamagljivanje slike, no zbog načina rada pripada skupini nelinearnih filtera. Algoritam je opisan u [16]. *Kuwahara* algoritam ima uključen mehanizam za očuvanje rubova predmeta na slici. Vrijednost svakog piksela se računa na način da se *kernel* pomiče po susjedstvu promatranog piksela, koji se uvijek mora nalaziti ispod pokretnog strukturnog elementa. Pomicanje se vrši sve dok varijanca piksela ispod kernela ne bude najniža. Zatim se računa prosječna vrijednost svih piksela ispod kernela i dodjeljuje promatranom pikselu. Implementacija algoritma je preuzeta iz [9], zajedno s parametrima:

- shape* Definiira oblik *kernela* koji se pomiče po susjedstvu promatranog piksela,
- size* Određuje veličinu pokretnog *kernela*,
- threshold* Služi kao mehanizam za sprečavanje pojave umjetnih rubova na pojedinim mjestima zbog „šuma“ na slici.

Algoritam se koristi prilikom kalibracije i prilikom mjerenja, a korišteni parametri su navedeni na slikama 3.12 i 3.18. Slika 2.19 prikazuje primjena algoritma na O-prstenima. U lijevom dijelu slike se nalazi O-prsten prije primjene algoritma, a u desnom nakon.

2.4.3. Algoritam *Minima*

Algoritam *Minima* služi za pronalaženje pojedinačnih piksela, ili skupina piksela istog intenziteta, koji su okruženi pikselima višeg intenziteta. Algoritam pripada skupini segmentacijskih algoritama. U programskom rješenju je korištena implementacija prema [9].



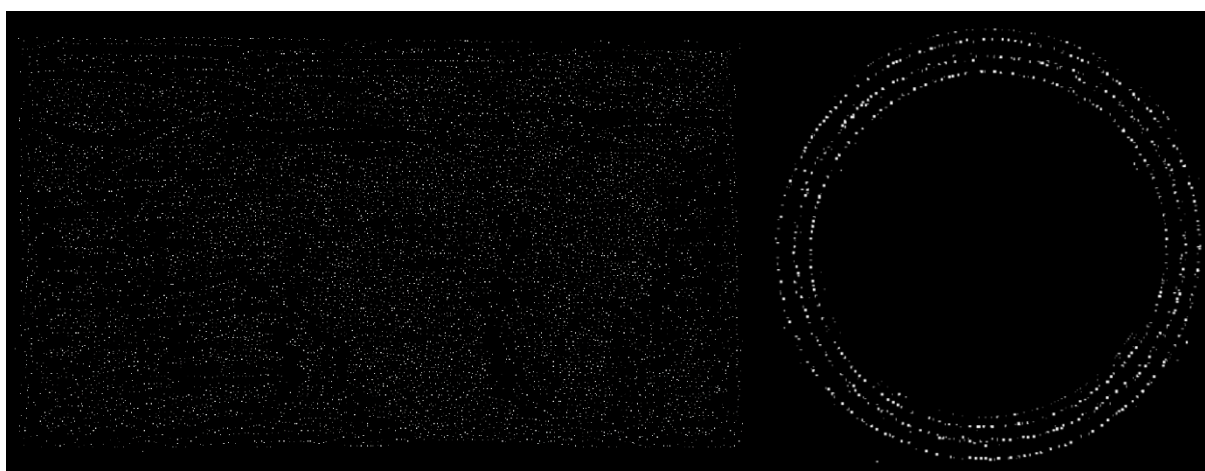
Slika 2.19. Primjena algoritma Kuwahara (desno) na O-prstenu

Algoritmu *Minima* je moguće podesiti sljedeće parametre:

connectivity Parametar određuje definiciju susjednih piksela. Ukoliko je vrijednost postavljena na *4-connectivity*, susjednim pikselima se smatraju 4 piksela iznad, ispod, lijevo i desno od promatranog piksela. Pri vrijednosti *8-connectivity*, susjedima se dodatno smatraju i 4 piksela dijagonalno od promatranog.

outType Određuje tip podataka izlazne slike.

U lijevom dijelu slike 2.20 je prikazana primjena algoritma na etalonu, a u desnom na O-prstenu.



Slika 2.20. Primjena algoritma Minima na etalonu i O-prstenu

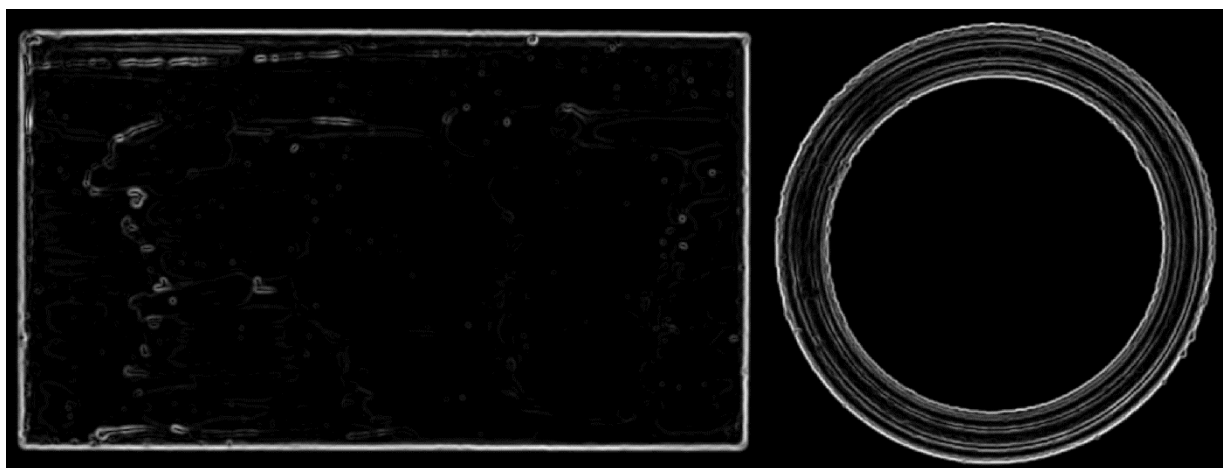
Vrijednosti parametara algoritma *Minima* pri kalibraciji su dane na slici 3.12, a pri mjerenju na slici 3.18.

2.4.4. Algoritam *Gradient Magnitude*

Algoritmom se prvo računaju gradijenti, odnosno razlike u intenzitetima susjednih piksela po x -osi i po y -osi slike. Kao rezultat se dobivaju vektori, kojima se određuje magnituda (modul). Također, slika se prije računanja gradijenata zamagljuje algoritmom *Gaussian Convolution* (potpoglavlje 2.4.1). Korištena je implementacija algoritma prema [9], s parametrom:

sigmas Standardna devijacija normalne raspodjele, koja određuje izgled *kernels*, odnosno način zamagljivanja slike.

Algoritam pripada skupini linearnih filtera, a koristi se za kalibraciju i za mjerenje. Primjena algoritma na etalonu je prikazana u lijevom dijelu slike 2.21, a na O-prstenu u desnom dijelu slike. Vrijednosti parametara pri kalibraciji se nalaze na slici 3.12, a pri mjerenju na slici 3.18.



Slika 2.21. Primjena algoritma *Gradient Magnitude* na etalonu (lijevo) i O-prstenu (desno)

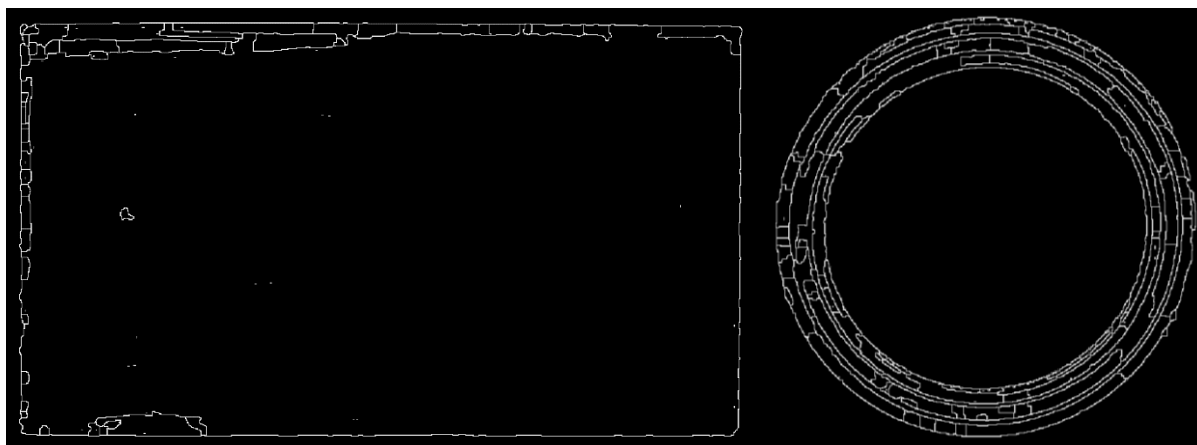
2.4.5. Algoritam *Seeded Watershed*

Algoritam *Seeded Watershed* pripada skupini segmentacijskih algoritama. Potrebne su mu dvije ulazne slike. Jedna slika sadrži početno sjeme (*seed*), a na drugoj se vrši operacija slijevanja (*watershed*). Operacija slijevanja se može vizualizirati kao vodeni tok. Voda izvire iz sjemena i širi se prema područjima više vrijednosti. Područja visokog intenziteta piksela su poput zidova koji ograničavaju tok vode. Ulazna slika sa sjemenjem je rezultat algoritma *Minima*, a slika po kojoj se vrši slijev je rezultat algoritma *Gradient Magnitude*. Algoritam uključuje i metodu za

sjedinjenje regija tijekom slijevanja. U programskom rješenju je korištena implementacija algoritma prema [9], uz parametre:

- connectivity* Parametar određuje definiciju susjednih piksela. Ukoliko je vrijednost postavljena na *4-connectivity*, susjednim pikselima se smatraju 4 piksela iznad, ispod, lijevo i desno od promatranog piksela. Pri vrijednosti *8-connectivity*, susjedima se dodatno smatraju i 4 piksela dijagonalno od promatranog.
- maxDepth* Određuje maksimalnu razliku intenziteta piksela unutar pojedinog područja kao uvjet sjedinjenja regija. Ukoliko je razlika manja od parametra, može se spajati s drugim regijama.
- maxSize* Definira maksimalnu veličinu pojedine regije kao uvjet sjedinjenja. Ako se unutar regije nalazi manje piksela od parametra, regija je pogodna za spajanje.
- outType* Određuje tip podataka izlazne slike.
- sortOrder* Definira način slijevanja. Opcija *low first* podrazumijeva kretanje od područja najnižeg intenziteta piksela slike prema područjima višeg intenziteta piksela, a opcija *high first* podrazumijeva suprotan smjer širenja.
- no gaps* Opcija podrazumijeva širenje slijeva sve dok se područja ne dotaknu.
- uphill only* Opcija ograničava slijevanje na samo jedan smjer, odnosno samo lokalno širenje regija.

Watershed algoritam se koristi kao glavni algoritam segmentacije promatranog predmeta od pozadine, a koristi se i pri kalibraciji i pri mjerenju. Primjer primjene na etalonu (lijevo) i O-prstenu (desno) prikazuje slika 2.22.



Slika 2.22. Primjena algoritma *Seeded Watershed* na etalonu (lijevo) i O-prstenu (desno)

Vrijednosti parametara algoritma *Seeded Watershed* pri kalibraciji se nalaze na slici 3.12, a pri mjerenju na slici 3.18.

2.4.6. Algoritam *Remove Edge Objects*

Algoritam pripada skupini algoritama za filtriranje binarnih slika. Algoritam uklanja spojene objekte koji dodiruju rubove slike. Prema implementaciji algoritma iz [9], moguće je podesiti samo parametar:

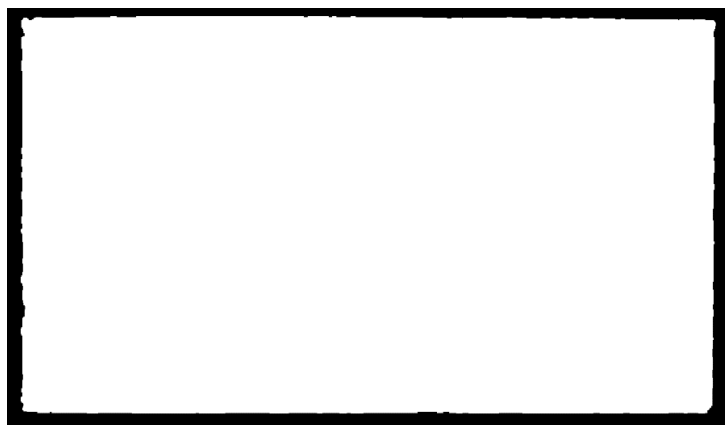
connectivity Parametar određuje definiciju susjednih piksela. Ukoliko je vrijednost postavljena na *4-connectivity*, susjednim pikselima se smatraju 4 piksela iznad, ispod, lijevo i desno od promatranog piksela. Pri vrijednosti *8-connectivity*, susjedima se dodatno smatraju i 4 piksela dijagonalno od promatranog.

2.4.7. Algoritam *Fill Holes*

Pomoću algoritma *Fill Holes* se popunjava prostor unutar objekta bijelom bojom (odnosno jedinicama) na binarnim slikama. Implementacija algoritma je preuzeta iz [9], a moguće je podesiti jedna parametar:

connectivity Parametar određuje definiciju susjednih piksela. Ukoliko je vrijednost postavljena na *4-connectivity*, susjednim pikselima se smatraju 4 piksela iznad, ispod, lijevo i desno od promatranog piksela. Pri vrijednosti *8-connectivity*, susjedima se dodatno smatraju i 4 piksela dijagonalno od promatranog.

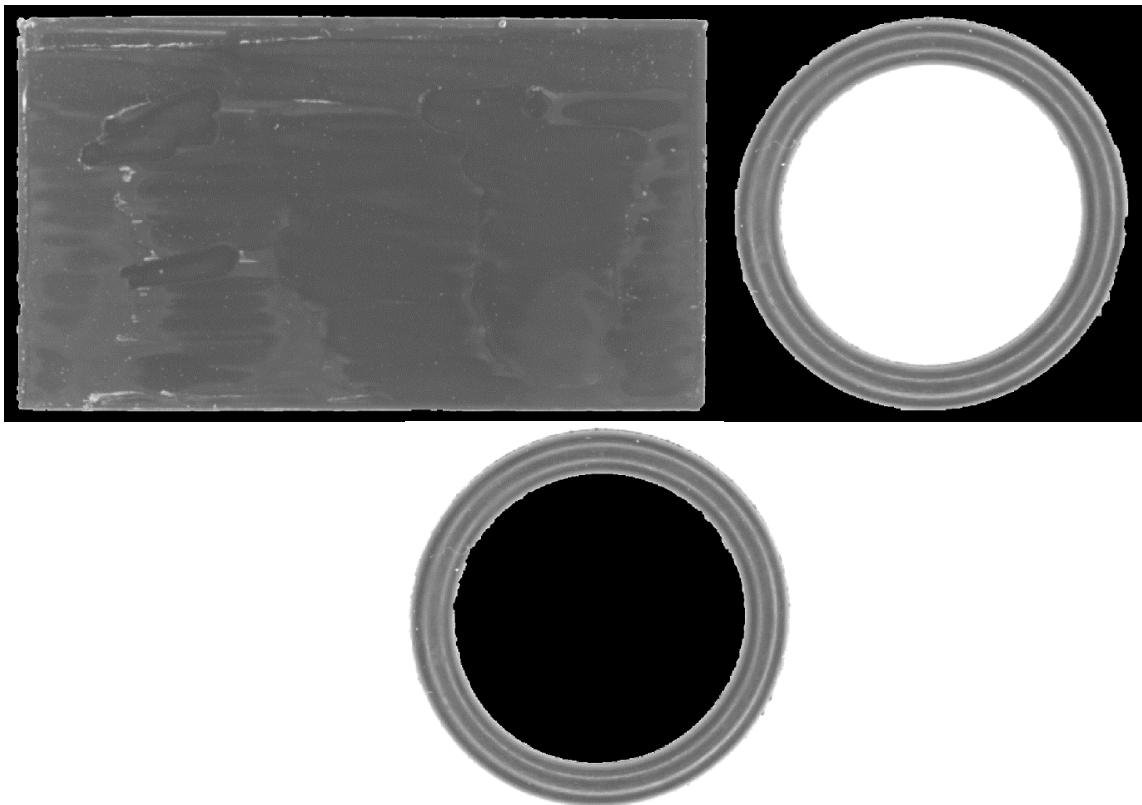
Primjena algoritma na etalonu se nalazi na slici 2.23.



Slika 2.23. Primjena algoritma *Fill Holes* na etalonu

2.4.8. Algoritam *Apply Mask*

Algoritam *Apply Mask* jednostavno preklapa jednu binarnu sliku preko druge. Algoritam se primjenjuje prilikom kalibracije i prilikom mjerenja, prema slici 2.24. Ulaz u algoritam su dvije slike. Prva slika je originalna slika na kojoj se nalazi promatrani predmet, a druga slika je binarna slika na kojoj je predmet od interesa segmentiran od pozadine. Preklapanje slika služi vizualnoj kontroli provedenog postupka segmentiranja predmeta i procjeni kvalitete dobivene konture. Algoritam se koristi za provjeru konture etalona (lijevo), te vanjskog (desno) i unutarnjeg (dolje) promjera O-prstena, kao što je prikazano na slici 2.24.



Slika 2.24. Primjena algoritma Apply Mask na etalonu, te vanjskom i unutarnjem promjeru O-prstena

2.4.9. Algoritam *Label*

Algoritam *Label* služi za obilježavanje spojenih područja na slici različitim brojem. Pozadina je označena brojem 0, prvi objekt brojem 1, drugi brojem 2, itd. Algoritmom se segmentirani objekti pripremaju za mjerenje.

Implementacija algoritma je preuzeta iz [9], sa sljedećim parametrima:

- connectivity* Parametar određuje definiciju susjednih piksela. Ukoliko je vrijednost postavljena na *4-connectivity*, susjednim pikselima se smatraju 4 piksela iznad, ispod, lijevo i desno od promatranog piksela. Pri vrijednosti *8-connectivity*, susjedima se dodatno smatraju i 4 piksela dijagonalno od promatranog.
- minSize* Određuje minimalan broj piksela koji određuju veličinu predmeta. Ukoliko predmet ima manje piksela od navedenog parametra, pridružuje se pozadini.
- maxSize* Definira maksimalan broj piksela koji određuju veličinu predmeta. Ako se predmet ima više piksela od vrijednosti parametra, pridružuje se pozadini.

Primjer vizualne reprezentacije primjene algoritma *Label* se nalazi na slici 2.25. Na slici su vidljiva dva etalona, označena različitom bojom (broj 1 → bijela boja, broj 2 → nijansa sive boje) i pozadina označena crnom bojom (broj 0). Vrijednosti parametara algoritma pri kalibraciji se nalaze na slici 3.12, a pri mjerenju na slici 3.18.



Slika 2.25. Primjena algoritma *Label* na etalonima za kalibraciju

2.4.10. Algoritam *Measure*

Algoritam *Measure* je zapravo skup algoritama za mjerenje odabranih veličina etalona, odnosno O-prstena. Rezultati mjerenja se mogu prikazati u pikselima ili u drugim mjernim jedinicama. Za prikazivanje rezultata u drugim mjernim jedinicama, mora biti određena veličina piksela algoritmom *Set Pixel Size*. Implementacija algoritma je preuzeta iz [9]. Kao ulaz u algoritam nužno je označiti sve potrebne mjerene veličine, objašnjene u potpoglavlju 2.2.2.

Osim mjerenih veličina, potrebno je i postaviti vrijednost parametru:

connectivity Parametar određuje definiciju susjednih piksela. Ukoliko je vrijednost postavljena na *4-connectivity*, susjednim pikselima se smatraju 4 piksela iznad, ispod, lijevo i desno od promatranog piksela. Pri vrijednosti *8-connectivity*, susjedima se dodatno smatraju i 4 piksela dijagonalno od promatranog.

Algoritmom mjerene veličine pri kalibraciji se nalaze na slici 3.12, a pri mjerenju na slici 3.18.

2.4.11. Algoritam *Working Distance Correction*

Algoritam je posebno izrađen za određivanje konstanti kalibracijske funkcije pomoću dva etalona. Ulaz u algoritam su rezultati mjerenja u pikselima, dobiveni pomoću *Measure* algoritma, i sljedeći parametri:

<i>feature</i>	Parametar određuje na temelju koje izmjerene veličine etalona se vrši kalibracija sustava.
<i>standard 1 height</i>	Visina etalona 1, t_1 [mm]
<i>standard 1 measure</i>	Mjerena veličina etalona 1, G_1 [mm]
<i>standard 2 height</i>	Visina etalona 2, t_2 [mm]
<i>standard 2 measure</i>	Mjerena veličina etalona 2, G_2 [mm]

Pomoću jednadžbi (2.16) i (2.24) se računaju vrijednosti mm/px omjer₁ prvog etalona i udaljenosti od leće do istog etalona g_1 . Izračunate vrijednosti su, uz visinu etalona t_1 , izlazi iz algoritma, koji potpuno određuju kalibracijsku funkciju (2.26). Vrijednosti parametara pri kalibraciji su navedene na slici 3.12.

2.4.12. Algoritam *Binary Area Closing*

Algoritam *Binary Area Closing* pripada skupini algoritama za filtriranje binarnih slika. Pomoću morfoloških transformacija se zatvaraju sva bijela područja na slici. Namjena mu je popunjavanje prostora između detektirane konture unutarnjeg i vanjskog promjera O-prstena.

Implementacija je preuzeta iz [9], a moguće je podesiti sljedeće parametre:

- connectivity* Parametar određuje definiciju susjednih piksela. Ukoliko je vrijednost postavljena na *4-connectivity*, susjednim pikselima se smatraju 4 piksela iznad, ispod, lijevo i desno od promatranog piksela. Pri vrijednosti *8-connectivity*, susjedima se dodatno smatraju i 4 piksela dijagonalno od promatranog.
- filterSize* Definira veličinu regije u pikselima. Sva spojena, crno obojena, područja manje veličine od parametra se pretvaraju u bijela područja.
- edgeCondition* Određuje na koji način se tretiraju pikseli izvan granica same slike. Mogu se tretirati kao pozadina crne boje (*background*) ili promatrani predmet bijele boje (*object*).

Primjena algoritma na O-prstenu sa slike 2.22 je prikazana na slici 2.26. Postavke parametara algoritma pri mjerenju su navedene na slici 3.18.



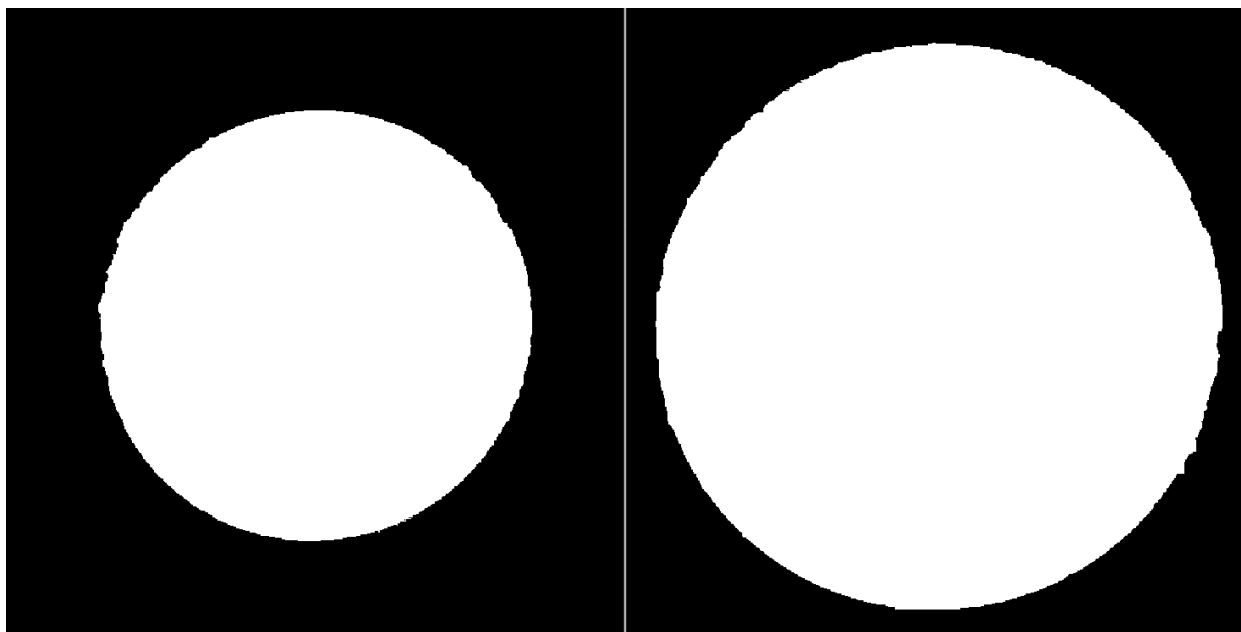
Slika 2.26. Primjena algoritma *Binary Area Closing* na O-prstenu

2.4.13. Algoritam *Segment O-ring*

Algoritam *Segment O-ring* je sastavljen od nekoliko algoritama. Ulaz u algoritam je slika 2.26, na koju se, kako bi se vanjski promjer pripremio za mjerenje, primjenjuje algoritam *Fill Holes* (potpoglavlje 2.4.7). U svrhu pripreme unutarnjeg promjera za mjerenje, na ulaznoj slici 2.26 se napravi inverzija crne i bijele boje, a zatim primijeni algoritam *Remove Edge Objects* (potpoglavlje 2.4.6). Rezultati navedenih operacija su prikazani na slici 2.27.

Algoritmu je moguće samo podesiti jedan parametar, koji prilikom mjerenja poprima vrijednost prikazanu na slici 3.18:

connectivity Parametar određuje definiciju susjednih piksela. Ukoliko je vrijednost postavljena na *4-connectivity*, susjednim pikselima se smatraju 4 piksela iznad, ispod, lijevo i desno od promatranog piksela. Pri vrijednosti *8-connectivity*, susjedima se dodatno smatraju i 4 piksela dijagonalno od promatranog.



Slika 2.27. Primjena algoritma *Segment O-ring* kako bi se pripremio unutarnji (lijevo) i vanjski (desno) promjer za mjerenje

2.4.14. Algoritam *Set Pixel Size*

Algoritam *Set Pixel Size* služi za definiranje veličine svakog piksela na slici u mjernim jedinicama duljine. Rezultat kalibracije, tj. mm/px omjer na visini mjenog predmeta se povezuje s pikselima slike, kako bi se rezultirajuća mjerenja mogla izraziti u milimetrima. Algoritam je preuzet iz [9], a moguće mu je podesiti jedan parametar:

units Parametar određuje mjernu jedinicu u kojoj se mjeri veličina jednog piksela, a moguće ga je postaviti na mm ili μm .

Postavka parametra prilikom mjerenja je navedena na slici 3.18.

2.4.15. Algoritam predobrade slike

Algoritam predobrade slike se pokreće programskom podrškom onog trenutka kad započnu mjerenja u stvarnom vremenu. Njega nije moguće mijenjati pomoću grafičkog sučelja programa, nego samo programskim kodom. Algoritam se mora izvršavati u vrlo kratkom vremenskom roku, stoga ga je potrebno optimizirati. Ulaz u algoritam je slika dobivena pomoću kamere, a izlaza iz algoritma većinom nema, ili je znatno manja slika samo jednog O-prstena, spremna za mjerenje. Funkcija algoritma je predobrada slike, provjera ispunjenja svih potrebnih uvjeta i identifikacija područja slike (ROI) u kojem se nalazi O-prsten. Koraci algoritma su sljedeći:

1. Detektiranje središta slike po x -osi.
2. *Thresholding* slike između donje granice intenziteta piksela 30 i gornje granice intenziteta piksela 240.
3. Popunjavanje prostora unutar detektiranog objekta pomoću algoritma *Fill Holes* (potpoglavlje 2.4.7).
4. Obilježavanje spojenih područja na slici primjenom algoritma *Label* (potpoglavlje 2.4.9), uz parametar $minSize=10000$.
5. Ukoliko je pronađen barem jedan objekt, prelazi se na idući korak algoritma, inače se prekida izvršavanje algoritma.
6. Izvršavanje mjerenja svih pronađenih objekata na slici algoritmom *Measure* (potpoglavlje 2.4.10). Mjerene veličine su koordinate centra, Podczeck-ov oblik elipse, te minimalne i maksimalne koordinate objekta po obje osi slike. Mjerene veličine su pojašnjene u potpoglavlju 2.2.2
7. Za svaki pronađeni objekt se provjerava nalazi li se Podczeck-ov oblik elipse u intervalu $0,98 \div 1,02$. Ukoliko se nalazi, objekt zadovoljava uvjet i prelazi se na idući korak. Na taj način se ugrubo provjerava je li detektirani objekt zapravo O-prsten.
8. Za svaki objekt, koji je zadovoljio prethodni uvjet, provjerava se uvjet položaja predmeta. Ukoliko se izmjerena koordinata centra objekta nalazi u intervalu ± 150 px od središta slike po x -osi, uvjet položaja je zadovoljen i prelazi se na idući korak.
9. Koristeći izmjerene minimalne i maksimalne koordinate objekta, stvara se ROI u obliku pravokutnika. Sa svake strane objekta se dodaje 50 px.
10. Dobiveno područje interesa, koje sadrži O-prsten, prosljeđuje se prema algoritmu za mjerenje, ali samo ako je prošlo minimalno 10 sekundi od prethodno dobivenog ROI. Ovaj uvjet služi tome da se isti O-prsten ne mjeri više od jednom. Algoritam za mjerenje je pojašnjen u potpoglavlju 3.3.

3. PROGRAMSKA PODRŠKA MJERNOG SUSTAVA

Kao što je već navedeno, nezaobilazni dio automatiziranog mjernog sustava je programska podrška. Ona uglavnom mora biti osmišljena samo za rješavanje pojedinog problema strojnog vida. Podrška obuhvaća setove instrukcija za upravljanje pojedinim uređajima, algoritme za predobradu i obradu podataka, te alate za interpretaciju i prikazivanje rezultata. U narednih nekoliko potpoglavlja bit će pojašnjena struktura programske podrške mjernog sustava, korišteni programski paketi, korišteni algoritmi i način korištenja izrađenog programskog rješenja.

Programsko rješenje za automatsku kontrolu dimenzija O-prstena je izrađeno pomoću programskog jezika *Python*. Pri izradi su također korišteni mnogi moduli otvorenog koda:

- *Numpy* [17, 18]:
 - Programski paket za znanstveno računalno programiranje, čija je osnovna namjena učinkovito rukovanje matricama.
- *Scipy* [19]:
 - Programski paket za znanstveno računalno programiranje, koji sadrži niz algoritama za numeričku analizu.
- *pypylon* [13]:
 - Programski paket koji sadrži alate i upravljačke programe za rukovanje kamerama proizvođača Basler AG.
- *DIPLib* [9]:
 - Biblioteka i razvojno okruženje za kvantitativnu analizu slike, bazirano na algoritmima za preciznu i učinkovitu obradu slika.
- *PyQtGraph* [20]:
 - Modul za izradu grafičkog sučelja, primarno namijenjen za upotrebu u područjima znanosti i inženjerstva.
- *multiprocess* [21, 22]:
 - Modul za distribuiranu obradu podataka istovremenim korištenjem više računalnih jezgri.

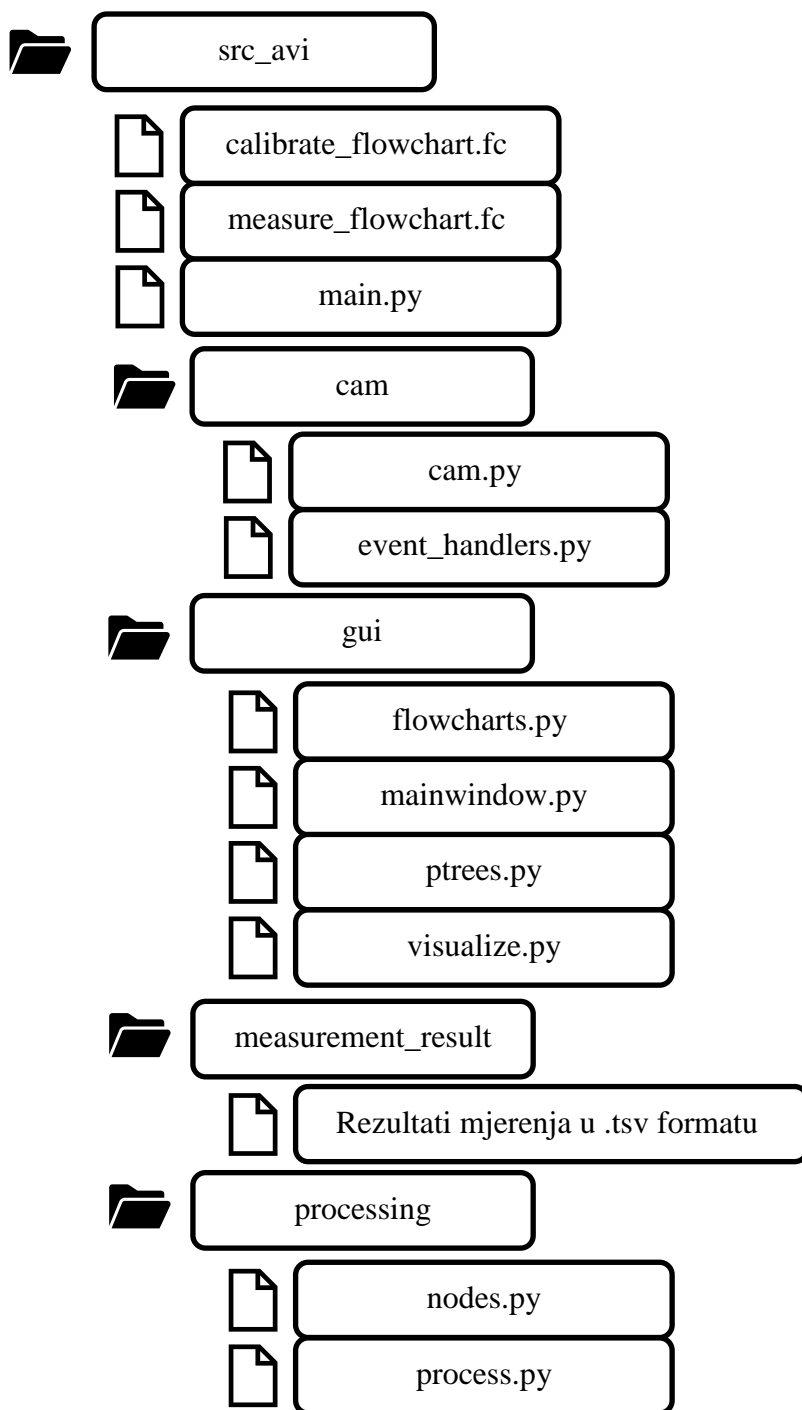
Struktura programskog rješenja prikazana je na slici 3.1. Programski kod svih skripti nalazi se u prilogu 1. Unutar glavnog foldera (*src_avi*) nalaze se četiri dodatna foldera, koja obuhvaćaju određenu cjelinu programskog koda, te skripta *main.py* (prilog 1.1). Pomoću navedene skripte se pokreće cijelo programsko rješenje kao zasebna aplikacija.

Folder *cam* sadrži skripte *cam.py* (prilog 1.2) i *event_handlers.py* (prilog 1.3), pomoću kojih su definirane naredbe za upravljanje kamerom Basler ace acA2500-60uc. Skripta *cam.py* sadrži objekt za uključivanje i isključivanje kamere, postavljanje parametara kamere prema potpoglavlju 2.3.1, pokretanje i zaustavljanje snimanja, te brojanje prikazanih slika u sekundi (FPS). Skripta *event_handlers.py* sadrži rutine za upravljanje događajima prilikom snimanja. Konkretno, skriptom su definirana dva objekta čije se funkcije izvršavaju svaki put kad se „uhvati“ jedna slika s kamere. Jedan objekt služi samo za prikazivanje slika u grafičkom sučelju programskog rješenja, a drugi poziva funkcije za predobradu i obradu slika, te samo mjerenje.

Unutar foldera *gui* se nalaze skripte *flowcharts.py* (prilog 1.4), *mainwindow.py* (prilog 1.5), *ptrees.py* (prilog 1.6) i *visualize.py* (prilog 1.7). Navedenim skriptama su definirani objekti, koji služe kao gradbeni blokovi grafičkog sučelja programskog rješenja. Pomoću skripte *mainwindow.py* je definiran izgled i funkcionalnost glavnog prozora programskog rješenja. Unutar skripte *flowcharts.py* se nalaze dva objekta, koji definiraju dva dijagrama toka korištena u programskom rješenju. Pomoću prvog objekta je određena grafička reprezentacija dijagrama toka koji se koristi za kalibraciju sustava, zajedno s podesivim parametrima i bibliotekom blokova za obradu slike. Drugi objekt definira iste značajke drugog dijagrama toka, koji se koristi za vršenje samog mjerenja. Skriptom *ptrees.py* su uspostavljene dvije sheme parametara u obliku stabla, te njihovi međusobni odnosi. Oba parametarska stabla se pojavljuju unutar glavnog prozora programskog rješenja na način da je istovremeno moguće mijenjati parametre samo jednog stabla, ovisno o kontekstu. Prvim parametarskim stablom su određene vrste postavki, te minimalne, maksimalne i zadane vrijednosti, koje se odnose na kameru. Drugim parametarskim stablom su definirane postavke koje se odnose na postupak i rezultate kalibracije. U posljednjoj skripti unutar foldera (*visualize.py*) se nalaze objekti, koji sadrže funkcije odgovorne za vizualni prikaz „uhvaćenih“ slika sa kamere. Skriptom se definiraju tri različita prozora grafičkog sučelja. Unutar prvog prozora se kontinuirano prikazuje snimka sa kamera i mjeri FPS, te se može sačuvati odgovarajuća slika. Unutar drugog prozora se na jednoj sačuvanoj slici može odrediti područje interesa (ROI), izvršiti uvid u raspodjelu intenziteta piksela na slici, te pripremiti ROI za prilagodbu kalibracijskog, odnosno mjernog dijagrama toka.

Folder *processing* sadrži skripte *nodes.py* (prilog 1.8) i *process.py* (prilog 1.9). U tim skriptama je prikazana sama logika svih algoritama za predobradu, obradu i mjerenja. Skripta *nodes.py* sadrži algoritme za vizualizaciju, filtriranje, binarne operacije, aritmetičke operacije, segmentaciju, kalibraciju sustava i mjerenje O-prstena. Algoritmi su programirani u obliku blokova (čvorova, *nodes*), kako bi se mogli proizvoljno dodavati i oduzimati sa kalibracijskog, odnosno mjernog

dijagrama toka. U skripti *process.py* se nalaze objekti koji omogućuju izvršavanje algoritma predobrade slika. Predobrada slika se vrši na znatno većem broju slika nego sama obrada i mjerenja. Stoga ju je, zbog mogućnosti primjene rješenja u realnom vremenu potrebno proračunavati na više jezgri računala. Sve potrebne funkcije za komunikaciju između različitih procesa i izvršavanje programskog koda na više jezgri računala se nalaze u skripti *process.py*.



Slika 3.1. Struktura programske podrške mjernog sustava

Unutar foldera *measurement_result* se nalaze rezultati mjerenja. Korisnik može tablične rezultate mjerenja O-prstena pohraniti u *.tsv* (*Tab-separated values*) datoteke. To je jednostavan tekstualan format datoteke za pohranu tabličnih struktura i njihovu razmjenu između baza podataka.

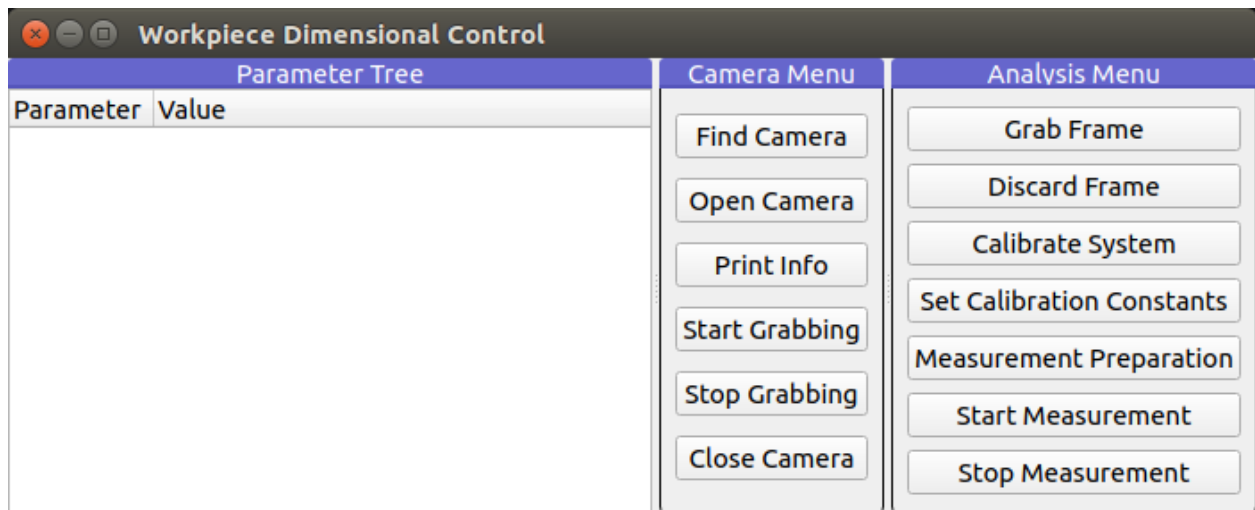
Unutar glavnog foldera (*src_avi*) se nalaze još dvije datoteke *calibrate_flowchart.fc* i *measure_flowchart.fc*. To su datoteke posebnog formata za pohranjivanje dijagrama toka radi njihove kasnije uporabe. Bilo bi vrlo nepraktično prilikom svakog pokretanja programskog rješenja nanovo definirati dijagrame toka. Na ovaj način ih je moguće jednom definirati, a zatim svaki put samo učitati iz pohranjenih datoteka, te po potrebi promijeniti određene parametre. Datoteka *calibrate_flowchart.fc* u sebi sadrži strukturu dijagrama toka za kalibraciju mjernog sustava, a *measure_flowchart.fc* strukturu dijagrama toka za vršenje mjerenja unutarnjeg i vanjskog promjera O-prstena.

U narednih nekoliko potpoglavlja biti će pojašnjen postupak korištenja programske podrške mjernog sustava, od uključivanja kamere, pa sve do vršenja mjerenja O-prstena u realnom vremenu.

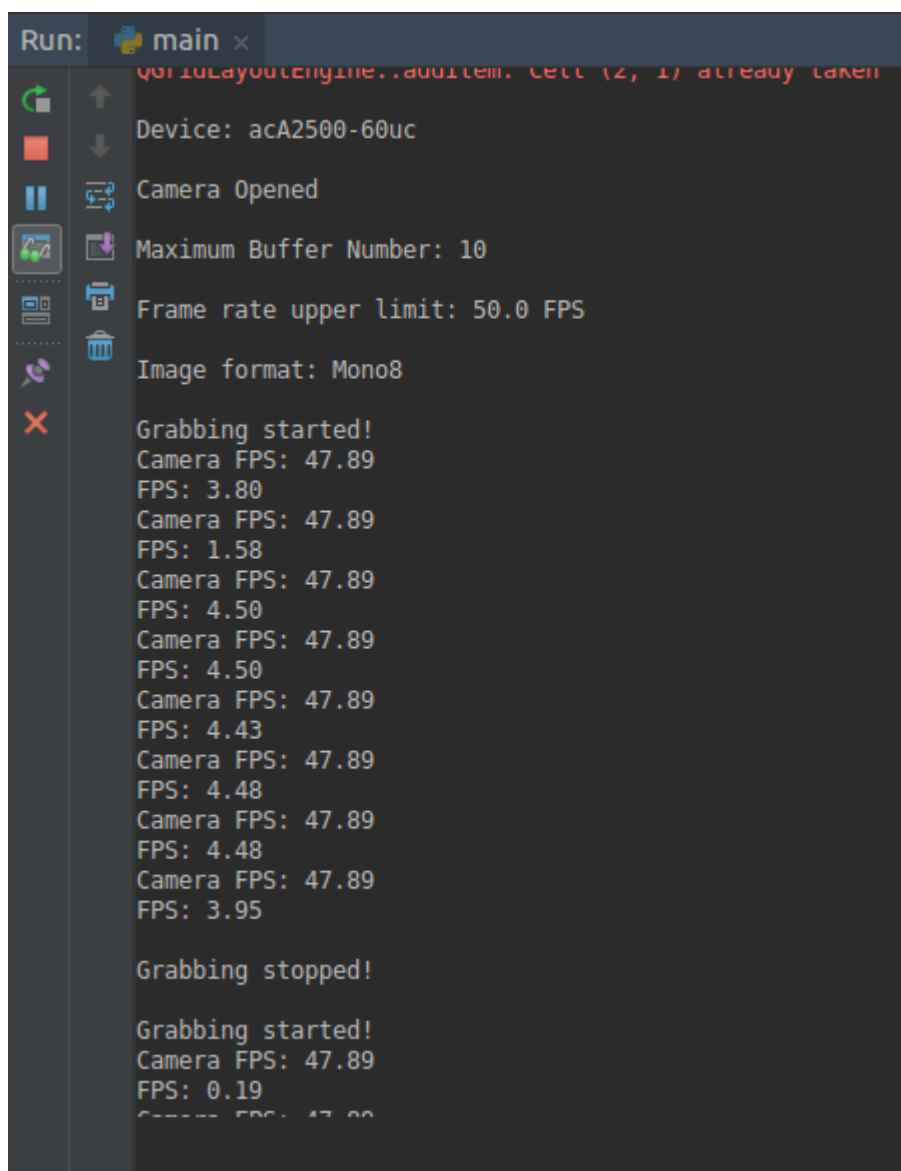
3.1. Rukovanje kamerom i glavni prozor grafičkog sučelja programske podrške

Nakon pokretanja programskog rješenja za mjerenje O-prstena, vidljiv je glavni prozor, kao na slici 3.2. Sučelje prozora je podijeljeno na tri dijela. U lijevom dijelu prozora se nalazi prostor za parametarsko stablo (*Parameter Tree*), kojim se podešavaju postavke kamere i kalibracije sustava, ovisno o kontekstu. U središnjem dijelu sučelja se nalaze gumbi pomoću kojih se izvršavaju aktivnosti vezane za kameru (*Camera Menu*), a u desnom dijelu gumbi pomoću kojih se otvaraju dodatni prozori za analizu slike, podešavanje kalibracijskog i mjernog dijagrama toka, te samo mjerenje u realnom vremenu (*Analysis Menu*).

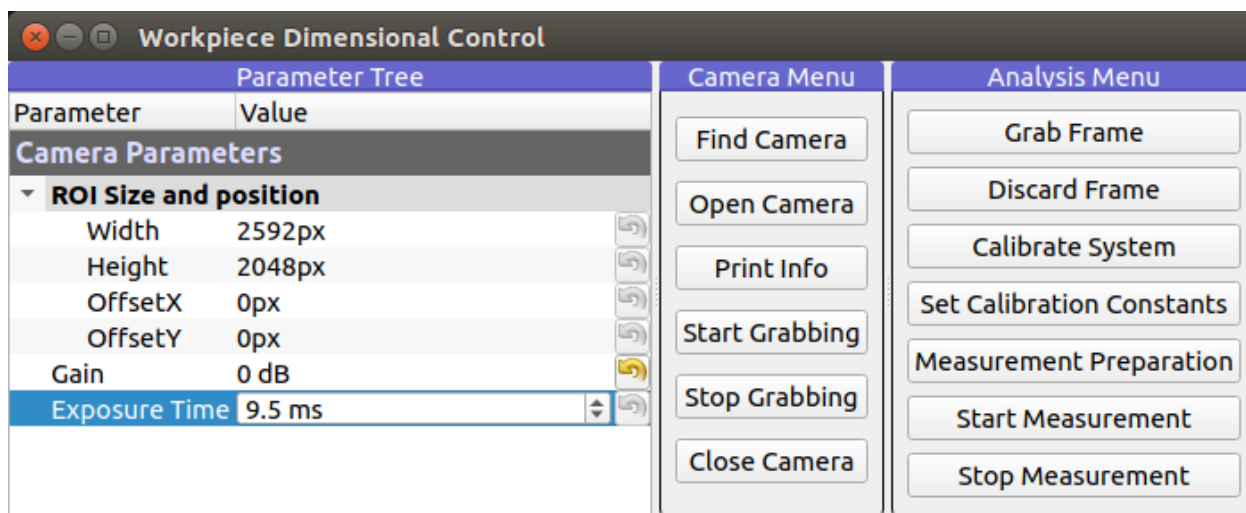
Uz glavni prozor, tijekom izvršavanja programskog rješenja, prisutna je i konzola za ispis informacija o izvršavanju, prikazana na slici 3.3. U konzoli se prikazuju statusne poruke, poput uključivanja i isključivanja kamere, FPS, početnih postavki kamere, promjene postavki kamere, te pokretanja i zaustavljanja određenih radnji. Nakon što je mjerni sustav postavljen, kao u potpoglavlju 2.3.5, potrebno je pronaći spojenu kameru pomoću gumba *Find Camera* (slika 3.2). Nakon pronalazanja kamere, potrebno ju je uključiti i učitati početne postavke pomoću gumba *Open Camera*. Pritiskom navedenog gumba otvara se parametarsko stablo s određenim postavkama kamere, kao na slici 3.4. Parametri kamere sa slike su pojašnjeni u potpoglavlju 2.3.1, te se iste vrijednosti koriste za dobivanje rezultata mjerenja iz priloga 5.



Slika 3.2. Glavni prozor programske podrške mjernog sustava



Slika 3.3. Konzola za ispis informacija o statusu izvršavanja programskog rješenja



Slika 3.4. Glavni prozor programske podrške mjernog sustava s parametarskim stablom kamere

Nakon postavljanja parametara kamere slijedi postupak kalibracije mjernog sustava pomoću dva etalona različite visine, prema potpoglavlju 2.3.4.

3.2. Kalibracija mjernog sustava pomoću programske podrške

Kako bi se mjerni sustav kalibrirao, odnosno odredio mm/px omjer na visini mjerenih O-prstena, prvo je potrebno centrirati etalone za umjeravanje. Pritiskom gumba *Start Grabbing* se otvara novi prozor, poput onog na slici 3.5, u kojem se određenom brzinom prikazuju slike s kamere. Broj prikazanih slika u sekundi (FPS) je ispisan u donjem desnom kutu prozora. Osi označavaju koordinatni sustav slike u pikselima. Slike je moguće uvećavati i umanjivati koliko je to potrebno. U desnom dijelu prozora je prikazan histogram slike s razinama za kontrolu raspona bijele i crne boje na slici. Nakon što su etaloni centrirani i postavke kamere zadovoljavajuće, potrebno je jednu sliku spremiti za daljnju obradu pritiskom gumba *Grab Frame* na glavnom prozoru programske podrške, prikazane na slici 3.4.

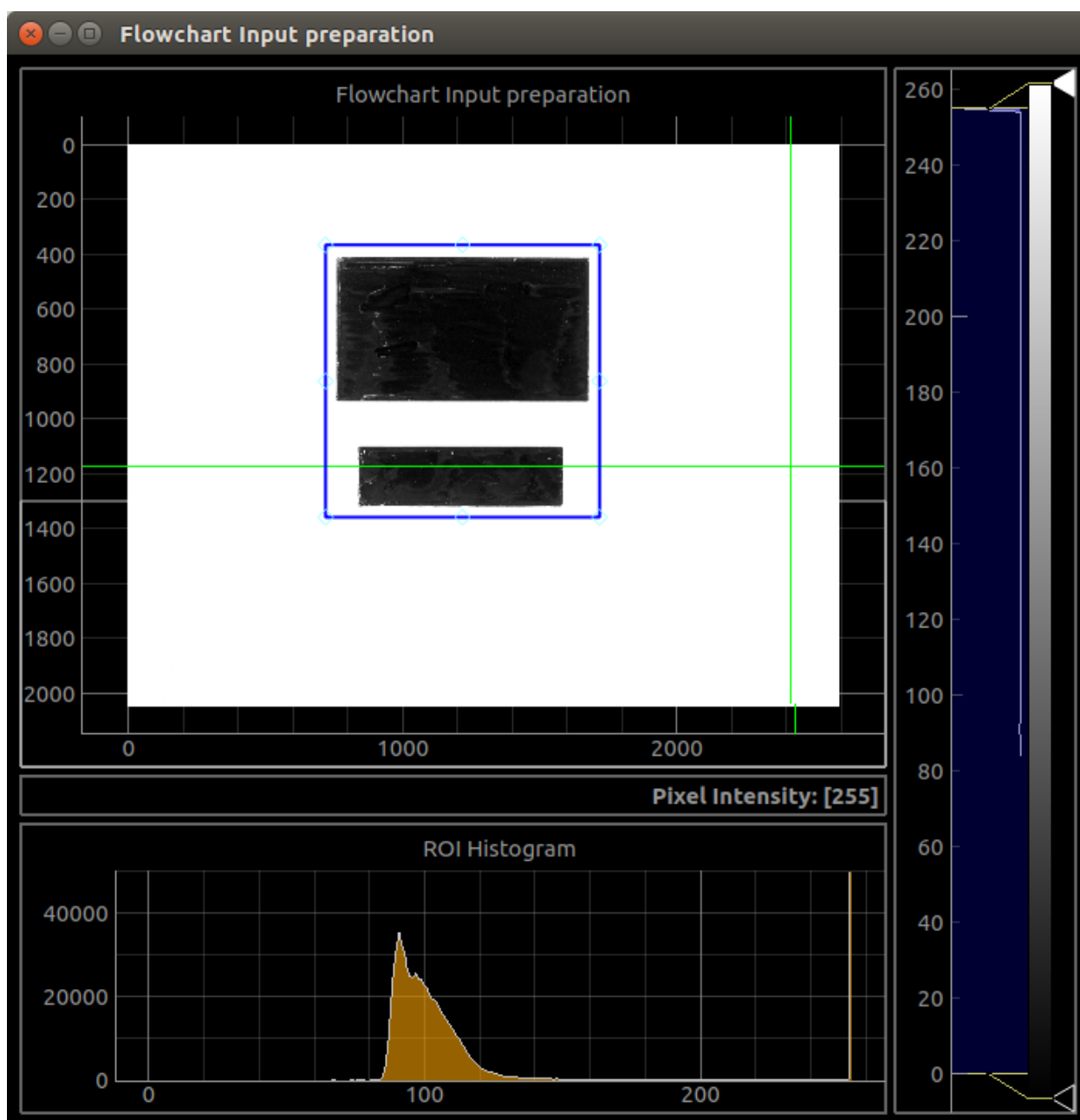
Pritiskom gumba se otvara novi prozor za pripremu slike za kalibraciju mjernog sustava, prikazan na slici 3.6. Nakon toga se može zatvoriti prethodno otvoren prozor na slici 3.5 pritiskom gumba *Stop Grabbing*. Grafičko sučelje prozora za pripremu slike je podijeljeno na 3 dijela. Glavni dio zauzima sama slika s pripadajućim koordinatnim osima u pikselima. Na slici je plavom bojom označeno područje interesa (ROI) unutar kojeg se nalaze etaloni. ROI je potrebno ručno pomicati i prilagoditi mu veličinu na način da obuhvaća oba etalona. Na taj način se izolira samo dio slike, koji služi kao ulaz u dijagram toka za kalibraciju sustava. Osim ROI, na slici se nalazi i pomični kursor označen zelenom bojom. Intenzitet pojedinog piksela ispod kursora je također ispisan (*Pixel Intensity*). U donjem dijelu grafičkog sučelja se nalazi histogram raspodjele intenziteta piksela

ispod ROI-a. Pomoću njega se može zaključiti o raspodjeli intenziteta piksela predmeta od interesa (etalona) i raspodjeli intenziteta piksela pozadine. S desne strane prozora na slici 3.6 se nalaze već spomenute razine kontrole raspona bijele i crne boje. Nakon što je ROI unutar kojeg se nalaze etaloni pravilno označen, može se prijeći na sljedeći korak pritiskom gumba *Calibrate System*.



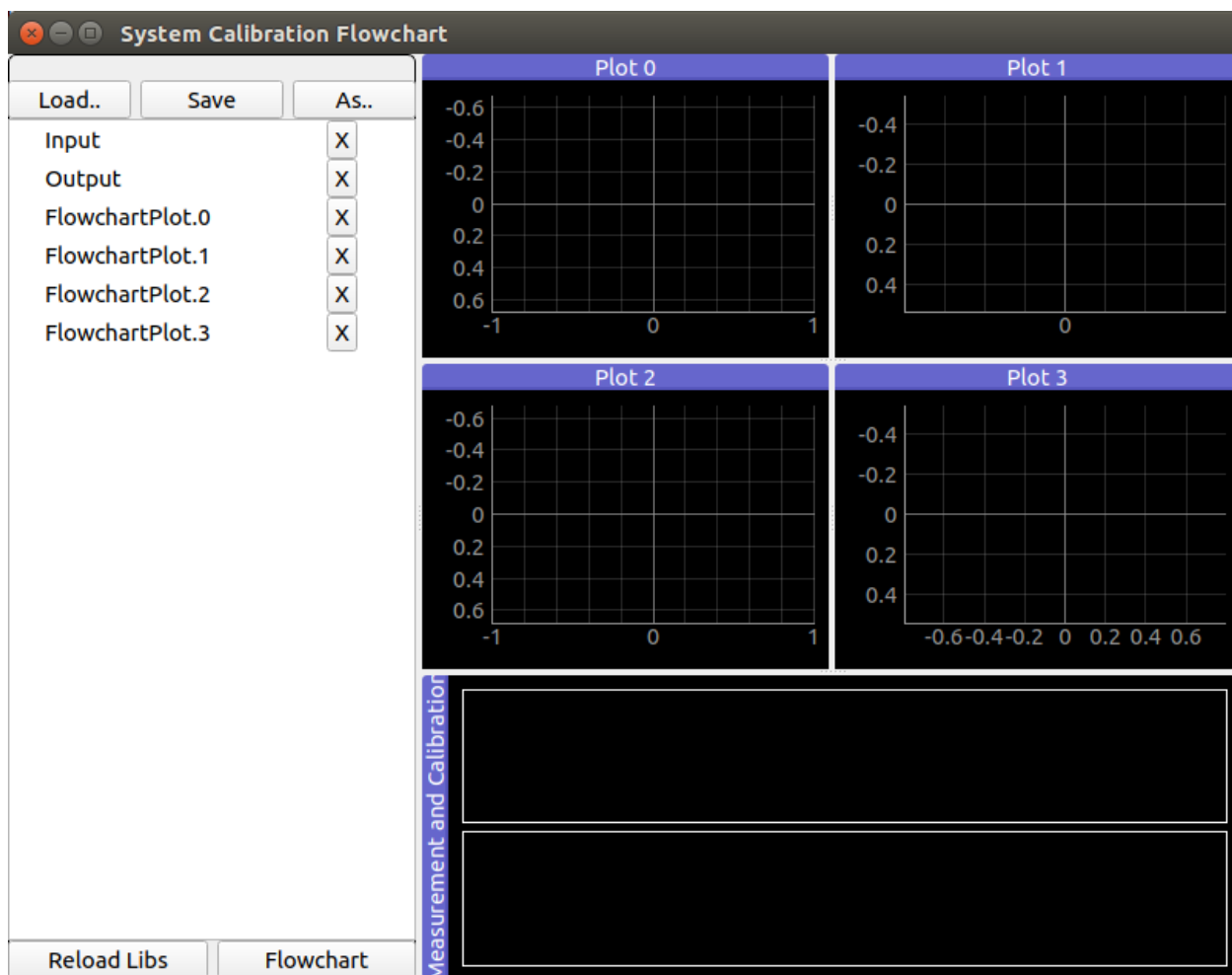
Slika 3.5. Prozor za prikazivanje niza slika dobivenih pomoću kamere s centriranim etalonima

Pritiskom gumba se otvara novi prozor koji vizualizira korake i rezultate dijagrama toka za kalibraciju sustava (slika 3.7). Na početku je prozor prazan, jer spremljeni dijagram toka nije još učitani. Pritiskom tipke *Load* sa slike 3.7, te odabirom spremljene datoteke *calibrate_flowchart.fc* se učitava dijagram toka sa diska. Nakon učitavanja se dobiva prozor nalik na sliku 3.8. U lijevom dijelu prozora su prikazani zasebni blokovi koji čine algoritam za kalibraciju sustava. Svaki blok ima vlastite podesive parametre pomoću kojih se regulira rezultat umjeravanja. Promjenom bilo kojeg parametra se algoritmi automatski osvježavaju i prikazuju ažurirane rezultate. Četiri slike na gornjoj desnoj strani prikazuju izlaze nakon četiri različita koraka algoritma. Slike se mogu pomicati, uvećavati i umanjivati, kako bi se vizualno pregledali izlazi pojedinih blokova.

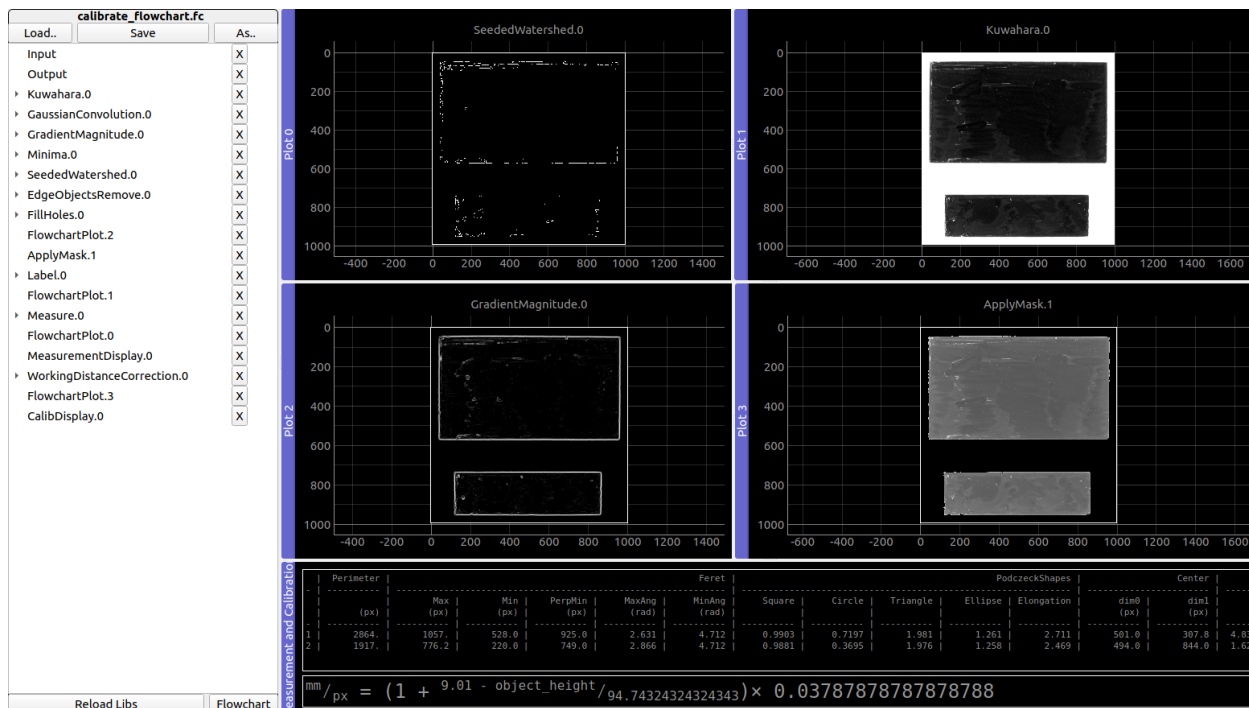


Slika 3.6. Prozor za pripremu slike na temelju koje se kalibrira optički sustav

U donjem desnom kutu prozora sa slike 3.8 prikazani su rezultati mjerenja etalona u pikselima i dobivena kalibracijska funkcija za računanje mm/px omjer-a prema potpoglavlju 2.3.4. Funkcija prikazana na slici 3.9 je dobivena na temelju izmjerenog promjera *Feret Min* i unaprijed poznatih širina i visina etalona.



Slika 3.7. Prozor za vizualizaciju kalibracijskog dijagrama toka prije učitavanja



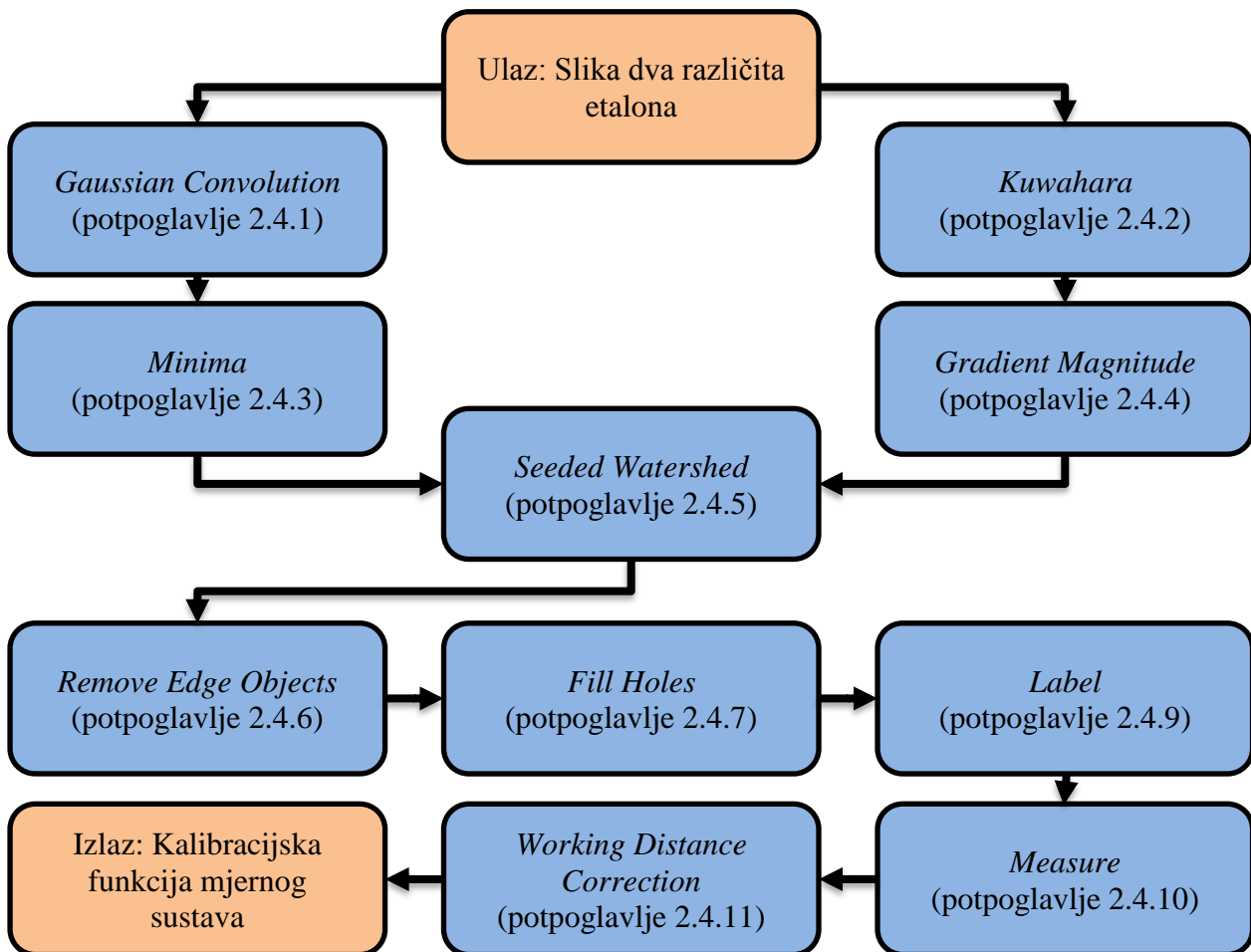
Slika 3.8. Prozor za vizualizaciju kalibracijskog dijagrama toka nakon učitavanja

	Perimeter		Feret				PodceckShapes				
	(px)	Max (px)	Min (px)	PerpMin (px)	MaxAng (rad)	MinAng (rad)	Square	Circle	Triangle	Ellipse	Elongation
1	2864.	1057.	528.0	925.0	2.631	4.712	0.9983	0.7197	1.981	1.261	2.711
2	1917.	776.2	220.0	749.0	2.866	4.712	0.9881	0.3695	1.976	1.258	2.469

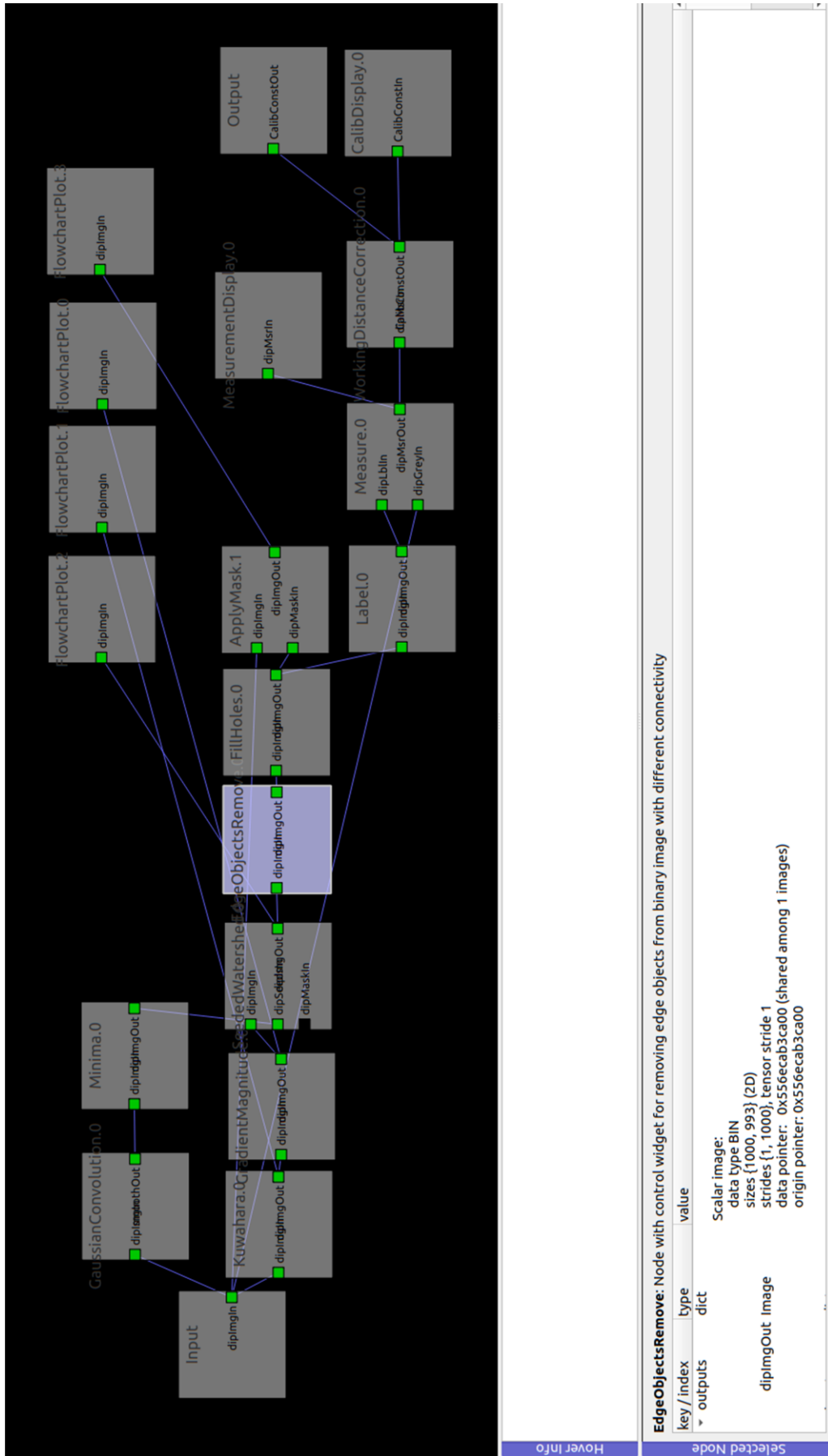
$$\text{mm}/\text{px} = (1 + 9.01 \cdot \text{object_height}/94.743243243243) \times 0.03787878787878788$$

Slika 3.9. Rezultati mjerenja etalona i kalibracijska funkcija

Pritiskom gumba *Flowchart* u donjem lijevom kutu slike 3.8 otvara se shema dijagrama toka za kalibraciju sustava, prikazana na slici 3.11. Na toj shemi se, po potrebi, mogu dodavati i oduzimati blokovi, koji zajedno čine jedan složeni algoritam kalibracije sustava. Dijagram toka istog algoritma s pripadajućim gradbenim blokovima, te poveznicama na potpoglavlja u kojima su pojašnjeni, prikazan je na slici 3.10. Parametri svakog bloka dijagrama toka prikazani su na slici 3.12. Ti isti parametri su korišteni za dobivanje rezultata kalibracije sa slike 4.13 pri svakom od devet provedenih mjerenja O-prstena.



Slika 3.10. Dijagram toka algoritma za umjeravanje optičkog sustava



Slika 3.11. Shema kalibracijskog dijagrama toka programskog rješenja

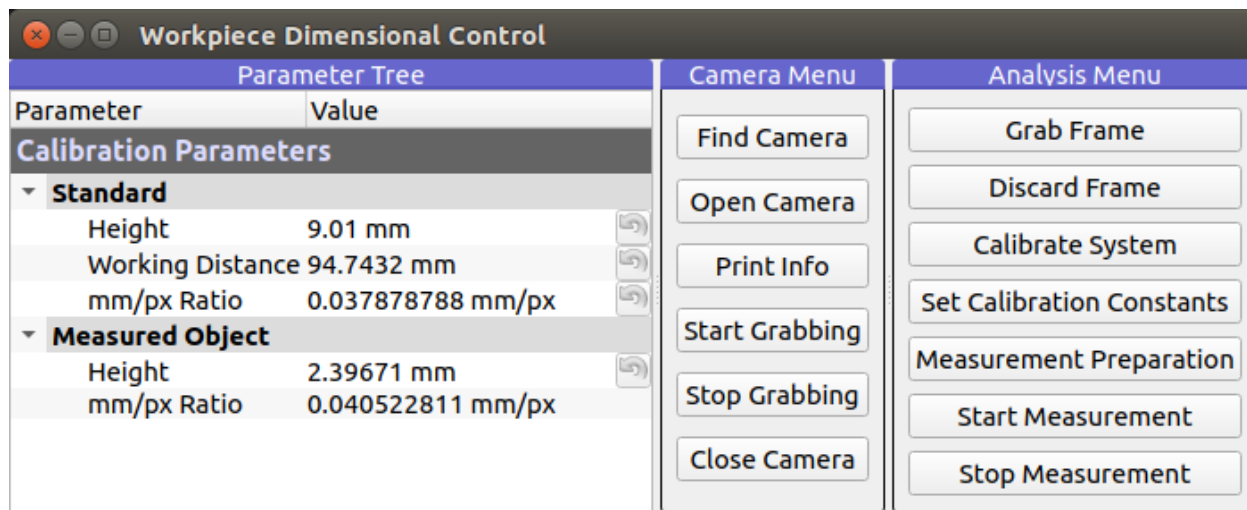
The image shows a parameter tree for an optical measurement algorithm. The tree is organized into two columns. The left column contains input and processing steps, and the right column contains output and measurement steps. Each step has various parameters, some with dropdown menus, checkboxes, or input fields.

Step Name	Parameter	Value / Option
Input	Input	X
	Output	X
	Kuwahara.0	X
Kuwahara.0	shape	rectangular
	size	1
	threshold	0
GaussianConvolution.0	sigmas	1
GradientMagnitude.0	sigmas	2
Minima.0	connectivity	8-Connectivity
	outType	binary
SeededWatershed.0	connectivity	8-Connectivity
	maxDepth	1
	maxSize	0
	outType	binary
	sortOrder	low first
	no gaps / uphill only	<input type="checkbox"/>
EdgeObjectsRemove.0	connectivity	8-Connectivity
FillHoles.0	connectivity	8-Connectivity
	FlowchartPlot.2	X
ApplyMask.1	X	
Label.0	connectivity	8-Connectivity
	minSize	1e+04
	maxSize	0
FlowchartPlot.1	X	
Measure.0	connectivity	8-Connectivity
	Perimeter	<input checked="" type="checkbox"/>
	Feret	<input checked="" type="checkbox"/>
	CartesianBox	<input type="checkbox"/>
	Minimum	<input type="checkbox"/>
	Maximum	<input type="checkbox"/>
	SolidArea	<input type="checkbox"/>
	ConvexArea	<input type="checkbox"/>
	ConvexPerimeter	<input type="checkbox"/>
	AspectRatioFeret	<input type="checkbox"/>
	Radius	<input type="checkbox"/>
	P2A	<input type="checkbox"/>
	Roundness	<input type="checkbox"/>
	Circularity	<input type="checkbox"/>
PodceckShapes	<input checked="" type="checkbox"/>	
Solidity	<input type="checkbox"/>	
Convexity	<input type="checkbox"/>	
EllipseVariance	<input type="checkbox"/>	
Eccentricity	<input type="checkbox"/>	
Center	<input checked="" type="checkbox"/>	
FlowchartPlot.0	X	
MeasurementDisplay.0	X	
WorkingDistanceCorrection.0	feature	Feret Min
	standard 1 height	9.01 mm
	standard 1 measure	20 mm
	standard 2 height	2 mm
	standard 2 measure	8.95 mm
FlowchartPlot.3	X	
CalibDisplay.0	X	

Slika 3.12. Parametarsko stablo algoritma za umjeravanje optičkog mjernog sustava

Nakon što je kalibracijski algoritam prilagođen uvjetima mjerenja O-prstena, potrebno je pritisnuti gumb *Set Calibration Constants* na glavnom prozoru grafičkog sučelja podrške, prikazanom na slici 3.4. Lijevi dio glavnog prozora se zatim osvježava, te prikazuje dobivene parametre kalibracije, kao na slici 3.13. Ti parametri su zapravo argumenti kalibracijske funkcije, pojašnjeni

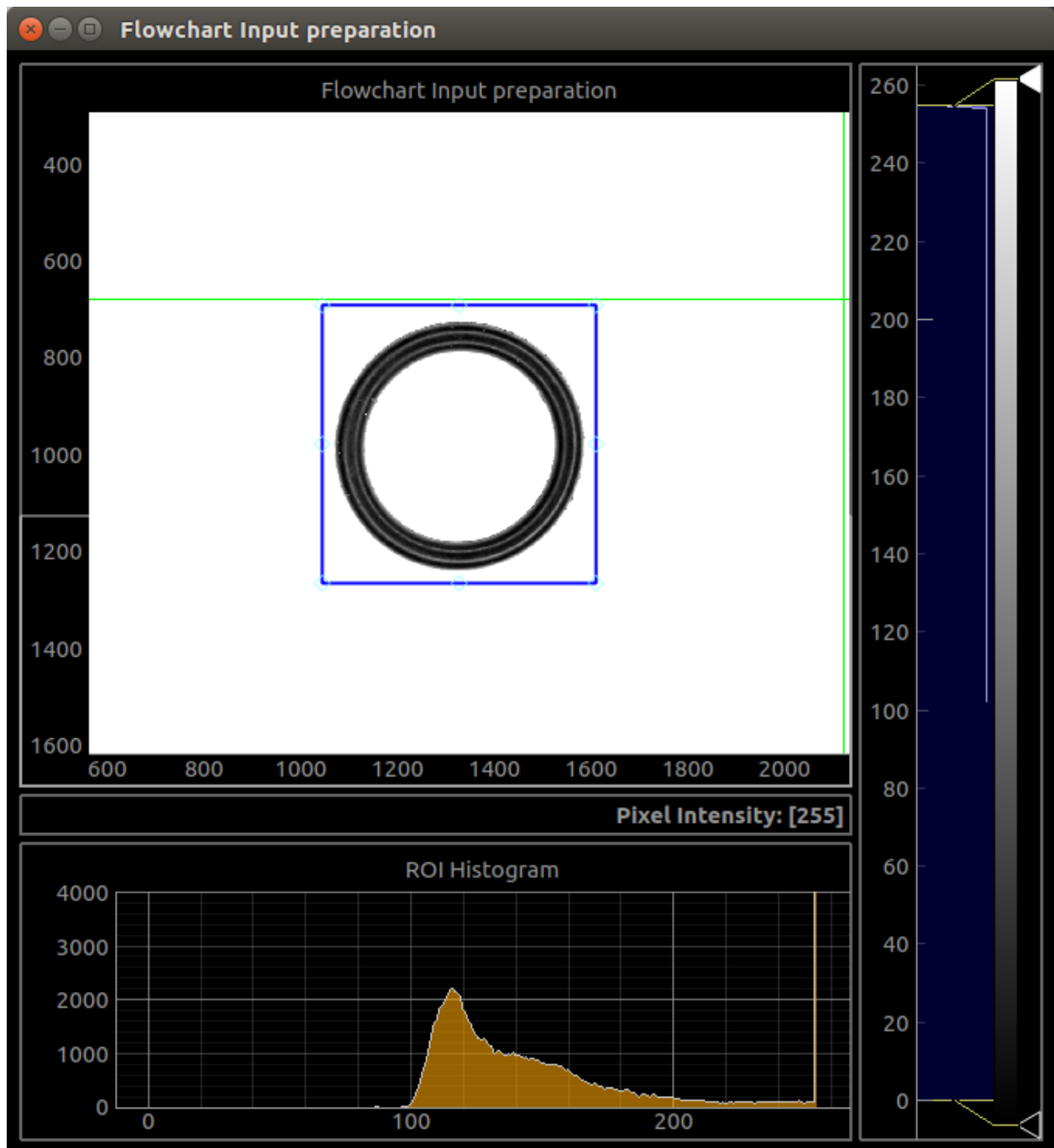
u potpoglavlju 2.3.4. Glavni argument funkcije je visina mjenenog objekta (*Measured Object* → *Height*). Vrijednost visine objekta od 2,397 mm je dobivena kao prosjek svih izmjerenih vrijednosti poprečnog presjeka O-prstena, prema tablici 4.1. Nakon dovršenog postupka kalibracije slijedi određivanje koraka algoritma za samo mjerenje O-prstena.



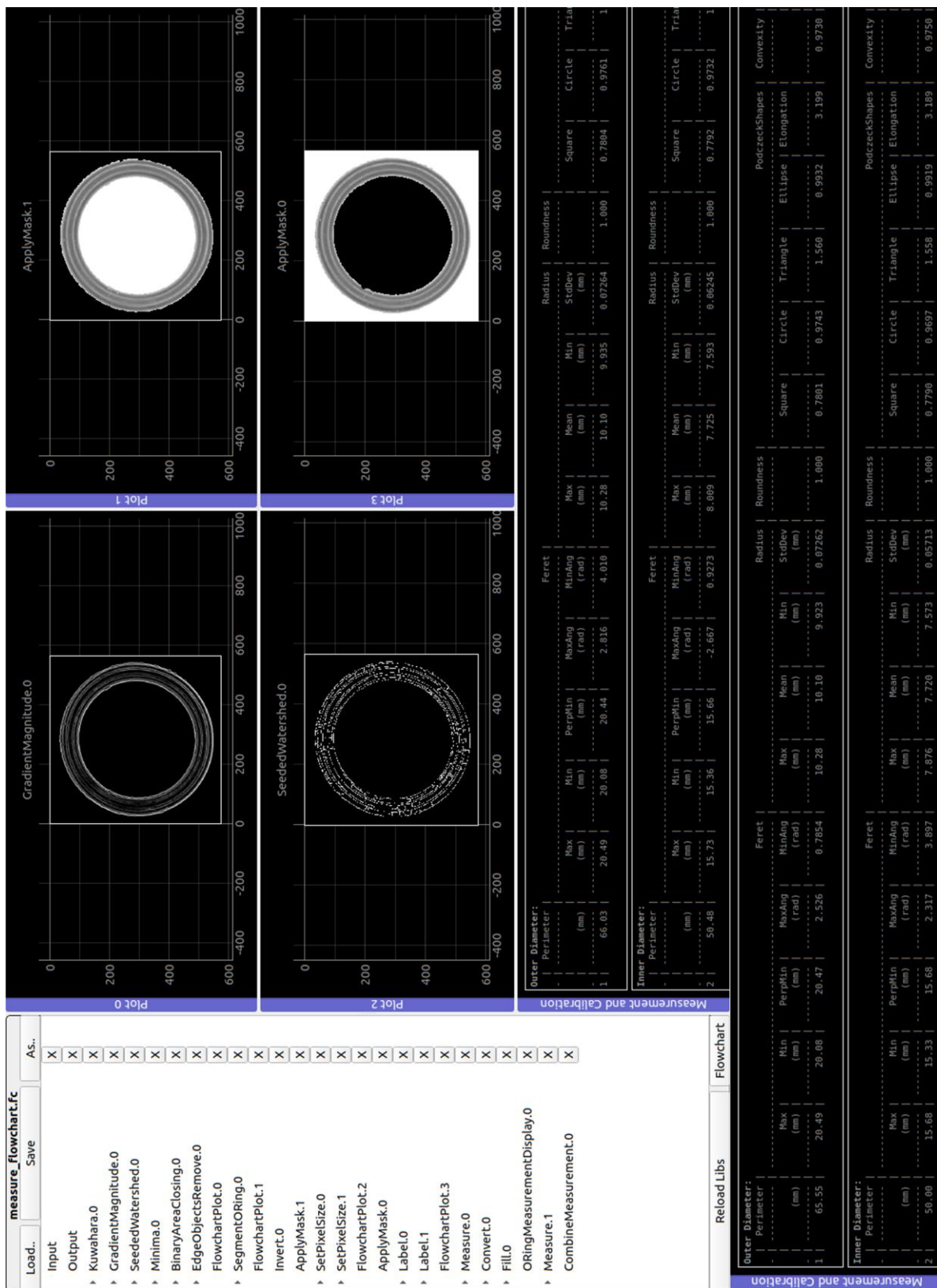
Slika 3.13. Glavni prozor programske podrške mjernog sustava s dobivenim parametrima kalibracije

3.3. Mjerni algoritam i postupak mjerenja predmeta

Postupak prilagođavanja mjernog algoritma se, kao i postupak umjeravanja, vrši na slici jednog O-prstena. Slika O-prstena treba odražavati uvjete koji vladaju u trenutku mjerenja. Postupak pribavljanja slike prati postupak opisan u potpoglavlju 3.2. Pritiskom gumba *Start Grabbing* na glavnom prozoru (slika 3.13), otvara se novi prozor u kojem se prikazuje niz slika s kamere. O-prsten je zatim potrebno centrirati na slici zbog što manjeg utjecaja distorzije. Gumbom *Grab Frame* se otvara novi prozor za pripremu slike, prikazan na slici 3.14. Na njemu je potrebno prilagoditi ROI (plava boja) da obuhvaća cijeli O-prsten. Nakon što je priprema slike završila, pritiskom gumba *Measurement Preparation*, na glavnom prozoru (slika 3.13), otvara se novi prozor s praznim dijagramom toka, kao na slici 3.7. Nakon učitavanja datoteke *measure_flowchart.fc* se dijagram toka za mjerenje O-prstena može vizualizirati slikom 3.15. Prozor je interaktivan, te sadrži četiri dijagrama za vizualizaciju izlaza pojedinih algoritama. Sa slike je vidljivo da je korišteno više algoritama nego pri kalibraciji mjernog sustava. Također, na slici su prikazane i izlazne vrijednosti algoritma za mjerenje. U donjem dijelu slike su prikazane izmjerene vrijednosti mjerenih veličina, koje se odnose na vanjski promjer O-prstena (*Outer Diameter*), odnosno unutarnji promjer O-prstena (*Inner Diameter*). Mjerene veličine su pojašnjene u potpoglavlju 2.2.2.

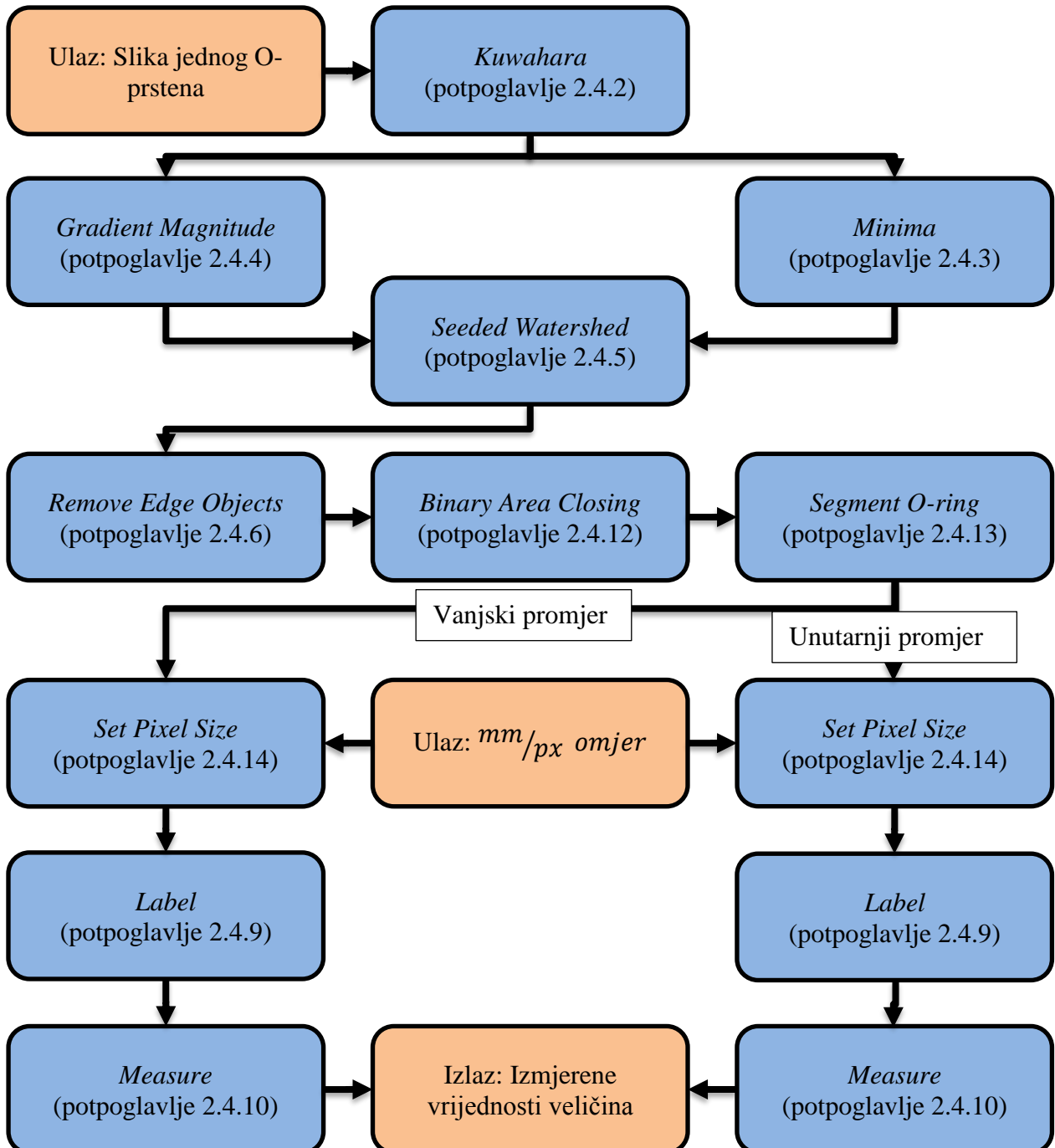


Slika 3.14. Prozor za pripremu slike na temelju koje se prilagođava mjerni algoritam sustava

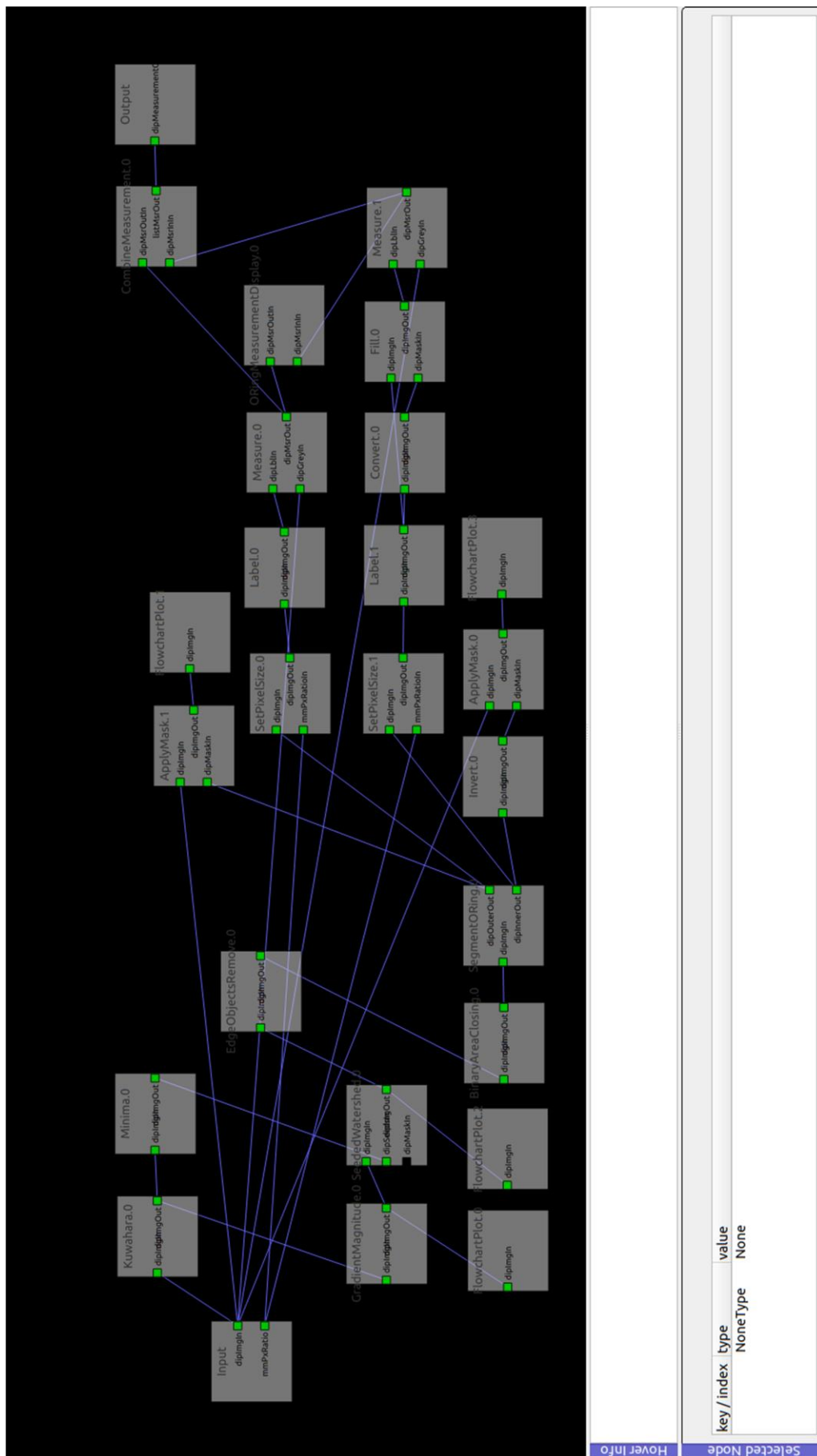


Slika 3.15. Prozor za vizualizaciju mjernog dijagrama toka nakon učitavanja

Pritiskom gumba *Flowchart* na slici 3.15, otvara se dijagram toka za mjerenje O-prstena. Prozor programskog rješenja s dijagramom toka prikazan je na slici 3.17. Pomoću prethodno definiranih gradbenih blokova je moguće prilagoditi algoritam, te pratiti ulazne i izlazne vrijednosti pojedinih blokova. Shemom na slici 3.16 je prikazan dijagram toka algoritma za mjerenje odabranih veličina O-prstena.

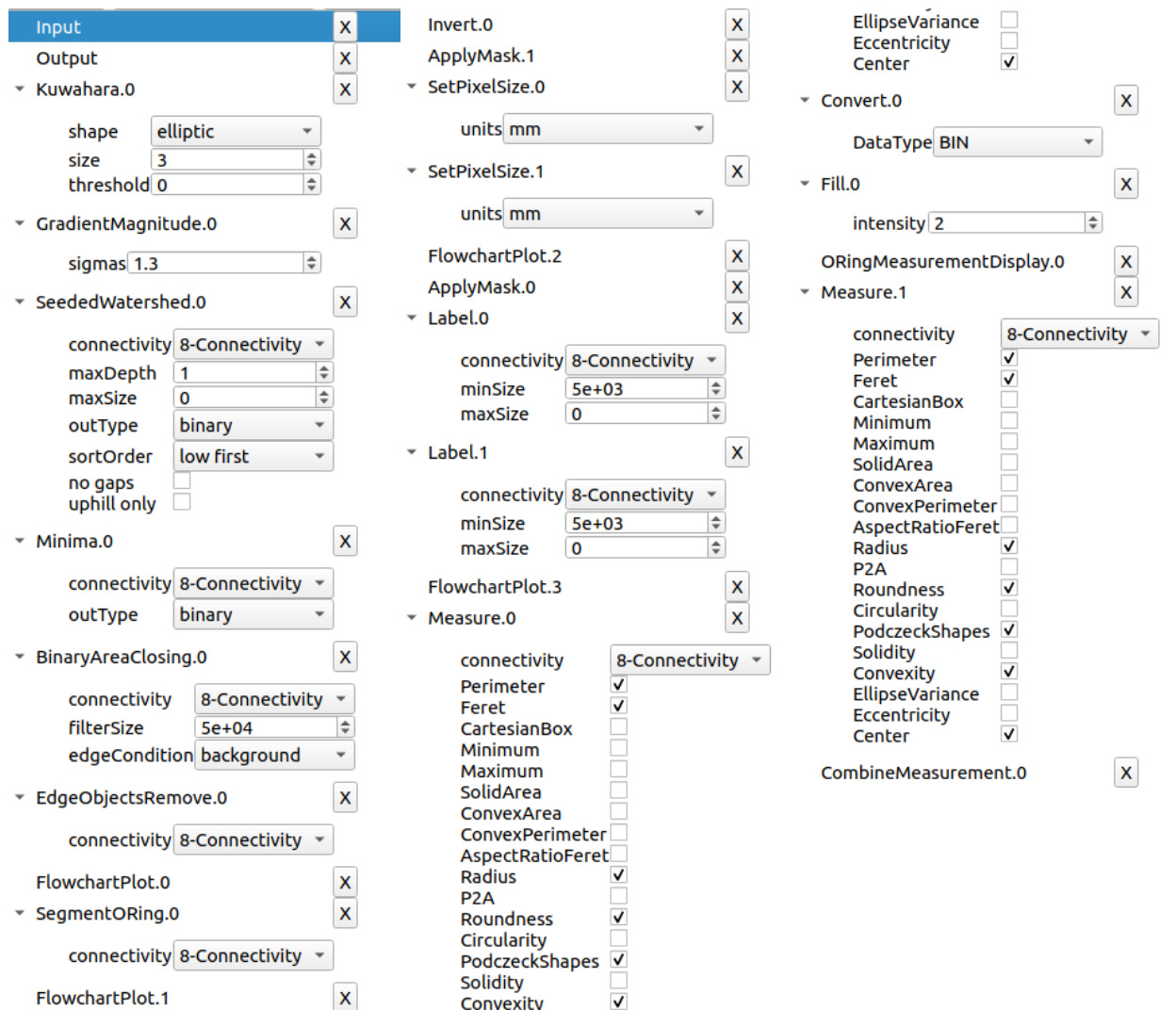


Slika 3.16. Dijagram toka mjernog algoritma optičkog sustava



Slika 3.17. Shema mjernog dijagrama toka programskog rješenja

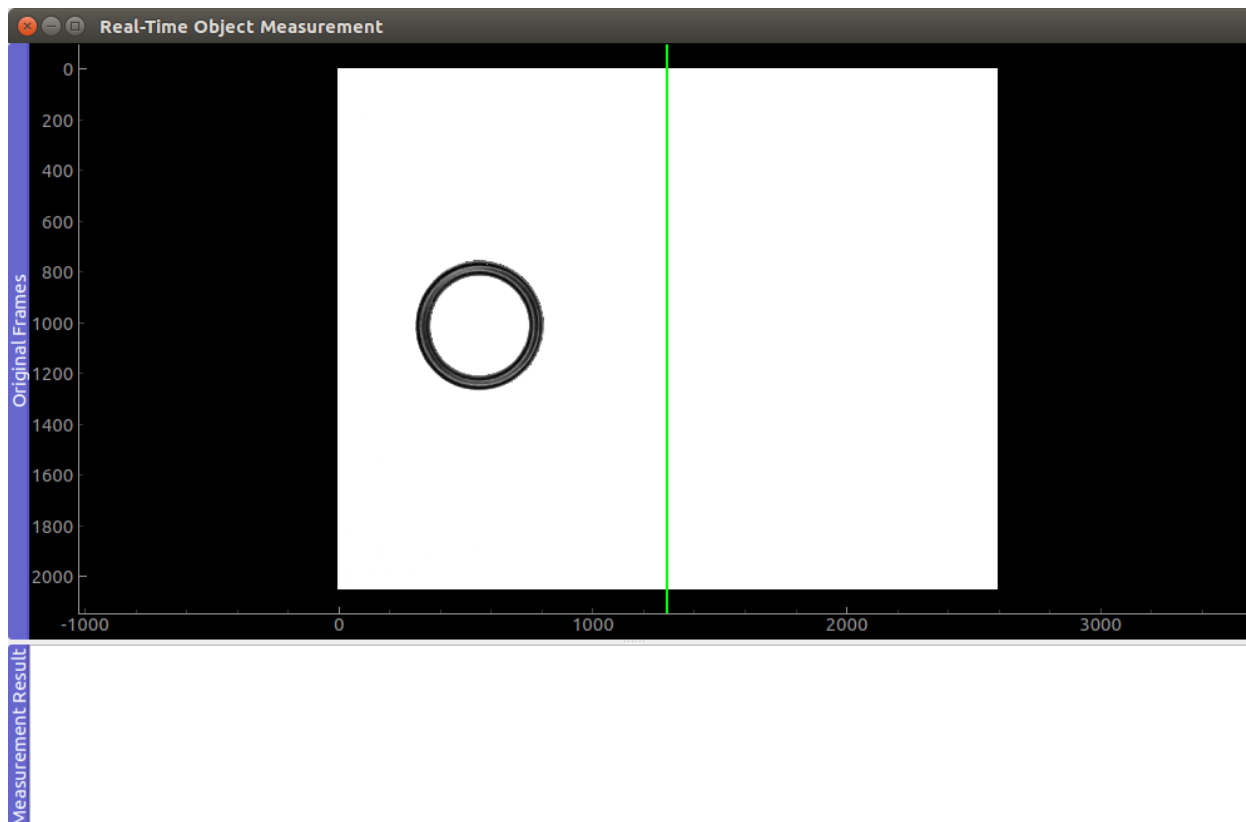
Nakon što su gradbeni blokovi algoritma definirani, potrebno je postaviti parametre svakog pojedinog bloka. Parametarsko stablo s već postavljenim parametrima mjernog algoritma se nalazi na slici 3.18. Moguće postavke parametara su pojašnjene za svaki algoritam zasebno u potpoglavlju 2.4. Korištenjem istih parametara su dobiveni konačni rezultati mjerenja O-prstena u prilogu 5.



Slika 3.18. Parametarsko stablo mjernog algoritma programske podrške

Nakon što je željeni mjerni algoritam potpuno određen, može se prijeći na mjerenja O-prstena u realnom vremenu. Gumbima *Start Measurement* i *Stop Measurement* glavnog prozora (slika 3.13) se otvara, odnosno zatvara novi prozor programske podrške. Prozor grafičkog sučelja, prikazan na slici 3.19, je podijeljen na dva dijela. U gornjem dijelu se vidi niz slika na kojima se mjere O-prsteni, te je zelenom linijom označena sredina slika s kamere. Donji dio slike je rezerviran za tablični prikaz rezultata mjerenja. Svaka slika s kamere prolazi kroz algoritam predobrade, koji je objašnjen u potpoglavlju 2.4.15. Svrha predobrade je identifikacija pozicije O-prstena na slici i

slanje manje slike, koja sadrži samo jedan O-prsten, na mjerenje pomoću mjernog algoritma. Algoritmi predobrade se izvršavaju paralelno, na tri posebna procesa, zbog potrebe za mjerenjima u realnom vremenu. Mjerenje se vrši samo kad je centar O-prstena u intervalu ± 150 piksela od centra slike (zelene linije). To pravilo je uvedeno kako bi se umanjio utjecaj distorzije slike zbog leće na rezultate mjerenja. Također, kako bi se uštedilo na vremenu mjerenja, veličina slike, koja se šalje na mjerenje, sadrži samo 50 piksela dodatka sa svake strane O-prstena. Mjerni algoritam je znatno sporiji i složeniji od algoritma predobrade i algoritma kalibracije.



Slika 3.19. Prozor programske podrške za vršenje mjerenja i prikazivanje rezultata

Slika 3.20 prikazuje prozor programske podrške s izvršenim mjerenjem jednog O-prstena. U donjem dijelu slike su tablično prikazani rezultati mjerenja O-prstena. Svaki stupac tablice označava jednu mjerenu veličinu i pripadajuću mjernu jedinicu. Rezultatima mjerenja jednog O-prstena pripadaju dva reda tablice. Redovi su označeni na način da se prvo prikazuje redni broj O-prstena prilikom mjerenja, a zatim oznaka odnosi li se mjerenje na vanjski ili unutarnji promjer (*in*, *out*). Primjeri oznaka redova su sljedeći:

- Prvi O-prsten, vanjski promjer $\rightarrow 1_out$,
- Prvi O-prsten, unutarnji promjer $\rightarrow 1_in$,
- Drugi O-prsten, vanjski promjer $\rightarrow 2_out$,

- Drugi O-prsten, unutarnji promjer $\rightarrow 2_in$,
- ...

Navedenim načinom označavanja redova su prikazani i rezultati optičkih mjerenja u prilogu 5.

Kako bi se rezultati mjerenja spremili u već spomenutu .tsv datoteku, potrebno je aktivirati padajući izbornik desnim klikom miša na tablicu s rezultatima i odabrati naziv datoteke.



Slika 3.20. Prozor programske podrške s izvršenim mjerenjem jednog O-prstena

Nakon završenih mjerenja, u sljedećem poglavlju slijedi analiza sposobnosti optičkog mjernog sustava.

4. MJERENJA I ANALIZA SPOSOBNOSTI SUSTAVA

U narednim potpoglavljima je opisan postupak mjerenja 10 odabranih O-prstena, predstavljenih u potpoglavlju 2.2. Proizvodi su prvo izmjereni mikrometarskim vijkom, a zatim u automatiziranom sustavu. Također, prikazane su postavke i parametri sustava pri kojima je mjerenje izvršeno, a dobiveni rezultati su međusobno uspoređeni. Za procjenu varijabilnosti sustava korištena je *Six Sigma* metodologija, a procjenu korelacije između mjerenja intraklasni koeficijent korelacije (ICC). Uz sposobnost automatiziranog sustava analizirana je i osjetljivost istog na kvalitetu kalibracije.

4.1. Postupak i rezultati mjerenja proizvoda mikrometarskim vijkom

Na 10 odabranih O-prstena mjerene su veličine vanjskog promjera, unutarnjeg promjera i poprečnog presjeka. Postupak mjerenja je sljedeći:

1. Mjerenje vrše tri različita operatera (A, B i C),
2. Svaki operater mjeri sve O-ringove slučajnim odabirom,
3. Svaki O-ring se mjeri devet puta (tri puta po operateru),
4. Mjerni instrumenti (mikrometarski vijci) su kalibrirani prije početka mjerenja,
5. Izmjerene vrijednosti se upisuju u mjerne listove za pojedinu mjerenu veličinu.

Opisani postupak podrazumijeva da svaki od 3 operatera izmjeri svaki od 10 O-prstena 3 puta, što rezultira s ukupnim brojem od 90 ($3 \times 10 \times 3$) mjerenja po svakoj mjerenoj veličini.

Mikrometarski vijak korišten za mjerenje vanjskog promjera O-prstena je *Mitutoyo Digmatic Micrometer MDC-25SX*. Potvrda o umjeravanju navedenog uređaja se nalazi u prilogu 3.1.

Karakteristike mikrometarskog vijka su sljedeće:

- Mjerno područje: $0 \div 25$ mm,
- Razlučivost: 1 μm ,
- Mjerna nesigurnost: $U = 1,1$ μm pri faktoru pokrivanja $k = 2$ (vjerojatnost 95%).

Mikrometarski vijak za vanjska mjerenja je kalibriran pomoću kontrolne palice, NM.KPR.002, nazivne mjere 25 mm. Potvrda o umjeravanju etalona nalazi se u prilogu 3.2. Kalibraciju mikrometarskog vijka prije mjerenja prikazuju slike 4.1 i 4.2. Provjerena mu je greška mjerenja na početku i na kraju mjernog područja, te je ustanovljeno poklapanje s potvrdom o umjeravanju.



Slika 4.1. Kalibracija mikrometarskog vijka za vanjske mjere na početku mjernog područja



Slika 4.2. Kalibracija mikrometarskog vijka za vanjske mjere pomoću kontrolne palice

Za mjerenje unutarnjeg promjera O-prstena korišten je mikrometarski vijak za unutrašnja mjerenja s dodirom u tri točke. Karakteristike mikrometarskog vijka su:

- Mjerno područje: $12 \div 16$ mm,
- Razlučivost: $5 \mu\text{m}$.

Mikrometarski vijak za unutrašnja mjerenja je kalibriran kontrolnim prstenom, NM.KPR.009, nazivne mjere 16,001 mm. Kalibraciju prikazuje slika 4.3.



Slika 4.3. Kalibracija mikrometarskog vijka za unutrašnja mjerenja kontrolnim prstenom

Slike 4.4, 4.5 i 4.6 prikazuju primjere mjerenja poprečnog presjeka, te vanjskog i unutarnjeg promjera O-prstena. Posebna pažnja je posvećena pritisku na O-prsten prilikom mjerenja jer su proizvodi izrađeni od vrlo elastične nitrilne gume, te su podložni elastičnim deformacijama.



Slika 4.4. Primjer mjerenja poprečnog presjeka O-prstena mikrometarskim vijkom



Slika 4.5. Primjer mjerenja vanjskog promjera O-prstena mikrometarskim vijkom



Slika 4.6. Primjer mjerenja unutarnjeg promjera O-prstena mikrometarskim vijkom

U narednim tablicama 4.1, 4.2 i 4.3 prikazane su izmjerene vrijednosti vanjskog i unutarnjeg promjera, te poprečnog presjeka O-ringa. Kao što je vidljivo iz tablica, svaki mjeritelj je izmjerio svaki O-prsten tri puta. Na dnu svake tablice je prikazan prosjek vrijednosti mjerene veličine za svaki O-prsten, kao njena procjena, te iznos standarde devijacije, kao mjera ponovljivosti mjerenja. Standardna devijacija je određena pomoću izraza za procjenu standardne devijacije cijele populacije na temelju uzorka [23]:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}, \quad (4.1)$$

gdje je:

- s [mm] Procjena standardne devijacije populacije na temelju uzorka,
- n [/] Veličina uzorka,
- x_i [mm] Izmjerene vrijednost i -tog uzorka,
- \bar{x} [mm] Prosjek izmjerenih vrijednosti uzorka.

Izmjerene vrijednosti mikrometarskim vijkom (ručna metoda) služe za kasniju usporedbu s izmjerenim vrijednostima u automatiziranom sustavu. Originalni mjerni listovi se nalaze u prilogima 4.1, 4.2 i 4.3.

Tablica 4.1. Izmjerene vrijednosti poprečnog presjeka [mm] (debljine) O-prstena mikrometerskim vijkom

Mjeritelj	Broj mjerenja	15,8 x		15,8 x		15,8 x		15,8 x		15,8 x		15,8 x	
		24_1	24_2	24_3	24_4	24_5	24_6	24_7	24_8	24_9	24_10		
A	1	2,430	2,412	2,385	2,375	2,387	2,387	2,370	2,408	2,406	2,389		
	2	2,444	2,418	2,358	2,372	2,395	2,371	2,372	2,412	2,407	2,403		
	3	2,450	2,419	2,356	2,384	2,386	2,391	2,367	2,418	2,397	2,408		
B	1	2,433	2,418	2,383	2,376	2,388	2,395	2,370	2,420	2,399	2,402		
	2	2,438	2,417	2,380	2,368	2,399	2,387	2,356	2,411	2,400	2,396		
	3	2,443	2,412	2,387	2,373	2,397	2,388	2,368	2,433	2,400	2,402		
C	1	2,438	2,429	2,370	2,371	2,396	2,392	2,361	2,417	2,396	2,397		
	2	2,444	2,412	2,369	2,377	2,385	2,395	2,356	2,422	2,407	2,402		
	3	2,430	2,413	2,382	2,376	2,395	2,376	2,373	2,413	2,404	2,390		
	Prosjek:	2,439	2,417	2,374	2,375	2,392	2,387	2,366	2,417	2,402	2,399		
	St. dev.:	0,0070	0,0055	0,0117	0,0045	0,0054	0,0083	0,0066	0,0075	0,0043	0,0063		
Prosjek		2,397											
St. dev.		0,023											

Tablica 4.2. Izmjerene vrijednosti unutarnjeg promjera [mm] O-prstena mikrometaskim vijkom

Mjeritelj	Broj mjerenja	15,8 x		15,8 x		15,8 x		15,8 x		15,8 x		15,8 x		15,8 x		
		24_1	24_2	24_3	24_4	24_5	24_6	24_7	24_8	24_9	24_10	24_11	24_12	24_13	24_14	24_15
A	1	15,870	15,770	15,910	15,900	15,740	15,900	15,860	15,900	15,790	15,860	15,900	15,860	15,900	15,790	15,870
	2	15,870	15,880	15,820	15,850	15,780	15,770	15,820	15,770	15,820	15,820	15,850	15,820	15,850	15,840	15,870
	3	15,860	15,790	15,740	15,780	15,770	15,750	15,770	15,770	15,750	15,770	15,740	15,770	15,740	15,760	15,850
B	1	15,750	15,680	15,770	15,800	15,730	15,740	15,900	15,740	15,900	15,900	15,670	15,900	15,670	15,680	15,750
	2	15,760	15,700	15,740	15,780	15,710	15,730	15,790	15,730	15,790	15,790	15,700	15,790	15,700	15,710	15,680
	3	15,740	15,650	15,740	15,700	15,690	15,780	15,820	15,780	15,820	15,820	15,640	15,820	15,640	15,630	15,790
C	1	15,730	15,710	15,720	15,790	15,710	15,770	15,810	15,770	15,810	15,810	15,680	15,810	15,680	15,720	15,740
	2	15,760	15,670	15,690	15,760	15,670	15,680	15,750	15,680	15,750	15,750	15,700	15,750	15,700	15,680	15,710
	3	15,750	15,750	15,680	15,740	15,730	15,710	15,770	15,710	15,770	15,770	15,700	15,770	15,700	15,700	15,760
	Prosjeck:	15,788	15,733	15,757	15,789	15,726	15,759	15,810	15,759	15,810	15,810	15,731	15,810	15,723	15,780	
	St. dev.:	0,0600	0,0723	0,0709	0,0586	0,0354	0,0617	0,0474	0,0617	0,0474	0,0474	0,0868	0,0474	0,0638	0,0698	
Prosjeck	15,760															
St. dev.	0,068															

Tablica 4.3. Izmjerene vrijednosti vanjskog promjera [mm] O-prstena mikrometaskim vijkom

Mjeritelj	Broj mjerenja	15,8 x		15,8 x		15,8 x		15,8 x		15,8 x		15,8 x	
		24_1	24_2	24_3	24_4	24_5	24_6	24_7	24_8	24_9	24_10		
A	1	20,453	20,385	20,301	20,505	20,541	20,287	20,503	20,347	20,465	20,467		
	2	20,475	20,212	20,411	20,438	20,463	20,378	20,400	20,336	20,246	20,411		
	3	20,582	20,337	20,395	20,407	20,328	20,412	20,513	20,328	20,343	20,348		
B	1	20,422	20,400	20,498	20,361	20,463	20,433	20,561	20,239	20,332	20,514		
	2	20,514	20,423	20,418	20,476	20,413	20,498	20,576	20,520	20,342	20,431		
	3	20,535	20,466	20,398	20,425	20,422	20,420	20,517	20,402	20,349	20,425		
C	1	20,481	20,522	20,428	20,467	20,540	20,460	20,502	20,312	20,472	20,521		
	2	20,607	20,513	20,550	20,450	20,484	20,493	20,525	20,545	20,502	20,487		
	3	20,594	20,494	20,465	20,526	20,476	20,473	20,595	20,380	20,493	20,565		
	Prosjeck:	20,518	20,417	20,429	20,451	20,459	20,428	20,521	20,379	20,394	20,463		
	St. dev.:	0,066	0,099	0,070	0,050	0,066	0,066	0,056	0,098	0,091	0,066		
Prosjeck	20,446												
St. dev.	0,084												

4.2. Analiza sposobnosti ručnog mjernog sustava (mjerjenja mikrometarskim vijkom)

Pri analizi sposobnosti ručnog mjernog sustava korišten je programski jezik *R* [24] s paketima *SixSigma* [25] i *irr* [26]. Svrha programskog okruženje *R* je statistička obrada i vizualizacija podataka. *SixSigma* paket je skup alata za primjenu *Six Sigma* metodologije, odnosno primjenu znanstvenih metoda za poboljšanje procesa i unapređenje kvalitete. Paket *irr* sadrži funkcije za izračunavanje raznih koeficijenata, koji služe za procjenu pouzdanosti i podudaranja između različitih procjenitelja (mjeritelja). Kompletan programski kod za analizu sposobnosti mjernog sustava se nalazi u prilogu 2.

Osnova *Six Sigma* metodologije je DMAIC krug, koji definira pet faza za primjenu *Six Sigma* metodologije unutar organizacije:

1. *Define* faza,
 - Određivanje opsega projekta, željenih ciljeva, te svih sudionika i njihovih odgovornosti,
2. *Measure* faza,
 - Prikupljanje podataka i vizualna reprezentacija procesa,
3. *Analyze* faza,
 - Analiziranje podataka, opisivanje glavnih problema i primjena statističkih metoda,
4. *Improve* faza,
 - Pronalaženje rješenja koristeći metoda poput *brainstorming-a*,
5. *Control* faza,
 - Implementacija rješenja iz prethodne faze i održavanje pozitivnog učinka.

Za procjenu sposobnosti mjernog sustava će biti korišteni alati iz *Measure* i *Analyze* faze DMAIC kruga, konkretno kontrolne karte, studije ponovljivosti i obnovljivosti (R&R) i ANOVA analiza.

Kontrolne karte su statistički alati, koji vizualno prikazuju kretanje pojedinih varijabli uključenih u proces. Vrijednosti varijabli se upisuju prema redoslijedu pojave u procesu na *y*-os, a identifikacijske oznake praćenih objekata se upisuju na *x*-os.

Kontrolne karte sadrže još i razine:

1. Prosječnu vrijednost – pokazatelj sredine oko koje se rasipaju vrijednosti,
2. Gornju i donju kontrolnu granicu – granice udaljene tri standardne devijacije od prosječne vrijednosti (pod pretpostavkom normalne razdiobe uključuju 99,7% podataka i ukazuju na pojavu iznimki u procesu).

U nastavku su korištena dva različita, no komplementarna tipa kontrolnih karti:

- \bar{x} – karta → prikazuje kretanje prosječnih vrijednosti praćenih veličina
- R – karta → prikazuje kretanje raspona (razlika najmanje i najveće) vrijednosti praćenih veličina

ANOVA (*Analysis of Variance*) je statistička metoda, koja ukazuje na izvore varijabilnosti unutar procesa. Njenom primjenom je moguće odrediti koliki udio u ukupnoj varijabilnosti ima određeni izvor. U slučaju mjernog sustava, izvore varijabilnosti je moguće prikazati slikom 4.7 [25]. Varijabilnost između mjerenih dijelova i varijabilnost mjernog sustava određuju ukupnu varijabilnost. U kvalitetnom mjernom sustavu je udio varijabilnosti zbog samog sustava relativno nizak u odnosu na udio varijabilnosti između dijelova. Ponovljivost sustava podrazumijeva podudaranje između dobivenih rezultata pri uzastopnom mjerenju iste veličine u istim uvjetima mjerenja, s obzirom na [27]:

- Proceduru mjerenja,
- Mjeritelja, mjerni uređaj i mjerno mjesto,
- Kratko vremensko razdoblje između uzastopnih mjerenja.

Obnovljivost sustava podrazumijeva podudaranje rezultata mjerenja iste mjerne veličine pri uzastopnom mjerenju istim ili različitim mjernih instrumentom, od strane više različitih mjeritelja. Obnovljivost obuhvaća utjecaj mjeritelja na varijabilnost mjernog sustava [27].

$$\text{Ukupna varijabilnost} \left\{ \begin{array}{l} \text{Varijabilnost sustava (R\&R)} \\ \text{Varijabilnost između dijelova} \end{array} \right\} \left\{ \begin{array}{l} \text{Ponovljivost} \\ \text{Obnovljivost} \end{array} \right\} \left\{ \begin{array}{l} \text{Procjenitelji} \\ \text{Interakcije} \end{array} \right\}$$

Slika 4.7. Izvori varijabilnosti mjernog sustava [25]

Kao kriterij kvalitete mjernog sustava, može se odrediti udio varijabilnosti mjernog sustava u ukupnoj varijabilnosti prema izrazu [25]:

$$\%R\&R = \frac{\sigma_{R\&R}}{\sigma_{uk}}, \quad (4.2)$$

gdje je:

- $\%R\&R$ [%] Udio varijabilnosti mjernog sustava u ukupnoj varijabilnosti,
- $\sigma_{R\&R}$ [mm] Varijabilnost mjernog sustava (standardna devijacija sustava pomnožena s 5,15 da obuhvaća 99% podataka),
- σ_{uk} [mm] Ukupna varijabilnost (ukupna standardna devijacija pomnožena s 5,15 da obuhvaća 99% podataka).

S obzirom na $\%R\&R$, mjerne sustave možemo podijeliti na:

- $\%R\&R < 10\%$ → Kvalitetan mjerni sustav,
- $10\% < \%R\&R < 30\%$ → Prihvatljiv mjerni sustav,
- $\%R\&R > 30\%$ → Loš mjerni sustav.

Rezultat ANOVA analize uključuje i statistički pokazatelj p -vrijednost, koji ukazuje na prihvaćanje, odnosno odbacivanje nul-hipoteze. Nul-hipoteza prilikom analize glasi da su prosjeci vrijednosti nezavisnih grupa (izvora varijabilnosti) jednaki. Ukoliko je p -vrijednost < 0.05 , nul-hipoteza se može odbaciti i zaključiti da postoji razlika između prosjeka vrijednosti pojedinih grupa.

Statistički pokazatelj odnosa varijabilnosti mjernog sustava i tolerancijskog polja procesa je definiran izrazom [27]:

$$R\&R/T = \frac{\sigma_{R\&R}}{USL - LSL} \times 100\%. \quad (4.3)$$

gdje je:

- $R\&R/T$ [%] Udio varijabilnosti mjernog sustava u toleranciji procesa.
- T [mm] Ukupna širina tolerancijskog polja mjerene veličine prema jednadžbi (2.9),
- USL [mm] Gornja granica procesa specificirana od strane klijenta (*Upper Specification Limit*) prema jednadžbi (2.8),
- LSL [mm] Donja granica procesa specificirana od strane klijenta (*Lower Specification Limit*) prema jednadžbi (2.8),

Uz standardne alate *Six Sigma* metodologije, kao dodatan statistički alat za procjenu pouzdanosti procjenitelja (mjeritelja ili mjernog sustava) koristi se intraklasni koeficijent korelacije (*Intraclass correlation coefficient*, ICC). ICC obuhvaća i stupanj korelacije i stupanj podudaranja između mjerenja. Koeficijent je primjenjiv u slučajevima kad više različitih procjenitelja (mjeritelji ili mjerni sustavi) donosi sud (izmjerena vrijednost) o određenim subjektima (proizvodima). Postoji nekoliko različitih načina računanja ICC-a, ovisno o načinu provođenja mjerenja. Način računanja ICC-a definiraju sljedeći parametri [28]:

model (<i>model</i>)	Odabran je <i>Two-Way Mixed-Effects</i> model, koji podrazumijeva da su jedino odabrani procjenitelji od interesa (nema generalizacije na opću populaciju), te se procjenjuje njihova pouzdanost.
tip (<i>type</i>)	Odabran je <i>single rater</i> tip jer se u primjeni planira koristiti izmjerena vrijednost jednog procjenitelja kao procjena mjerene veličine (ne koristi se prosjek izmjerenih vrijednosti više procjenitelja).
definicija (<i>definition</i>)	Koristi se <i>consistency</i> definicija jer se u obzir uzima dosljednost rezultata između procjenitelja, a ne apsolutno slaganje.

Uzimajući u obzir prethodno definirane parametre, ICC se računa prema izrazu [28]:

$$ICC(C, 1) = \frac{MS_R - MS_E}{MS_R + (k - 1)MS_E}, \quad (4.4)$$

gdje je:

ICC(C, 1) [/]	Intraklasni koeficijent korelacije (<i>Intraclass correlation coefficient</i>),
MS_R [/]	Kvadratna sredina svih stupaca,
MS_E [/]	Kvadratna sredina greške,
k [/]	Broj procjenitelja (mjeritelja ili mjernih sustava).

Pri procjeni pouzdanosti (korelacije i dosljednosti) između procjenitelja pomoću ICC koeficijenta, u praksi se koriste sljedeće pravila [28]:

- $ICC < 0,5 \rightarrow$ Slaba pouzdanost,
- $0,5 \leq ICC < 0,75 \rightarrow$ Umjerena pouzdanost,
- $0,75 \leq ICC < 0,9 \rightarrow$ Dobra pouzdanost,
- $ICC \geq 0,9 \rightarrow$ Odlična pouzdanost.

4.2.1. Sposobnost sustava pri mjerenju poprečnog presjeka O-prstena

Kako bi se procijenio intraklasni koeficijent korelacije pri mjerenju poprečnog presjeka O-prstena mikrometarskim vijkom, potrebno je dobivene rezultate iz tablice 4.1 preoblikovati u novu tablicu 4.4. Tablica 4.4. je oblikovana na način da svaki stupac označava procjenitelja (mjeritelja), a svaki red subjekta (O-prsten). Vrijednosti unutar tablice su aritmetička sredina vrijednosti triju mjerenja, koja je izvršio svaki mjeritelj nad svakim O-prstenom.

Tablica 4.4. Prosječne izmjerene vrijednosti poprečnog presjeka [mm] O-prstena po mjeritelju

O-prsten	A	B	C
15,8 x 24_1	2,441	2,438	2,437
15,8 x 24_2	2,416	2,416	2,418
15,8 x 24_3	2,366	2,383	2,374
15,8 x 24_4	2,377	2,372	2,375
15,8 x 24_5	2,389	2,395	2,392
15,8 x 24_6	2,383	2,390	2,388
15,8 x 24_7	2,370	2,365	2,363
15,8 x 24_8	2,413	2,421	2,417
15,8 x 24_9	2,403	2,400	2,402
15,8 x 24_10	2,400	2,400	2,396

Pomoću programskog paketa *irr* i okruženja *R* je procijenjen ICC koeficijent:

```

266 Single Score Intraclass Correlation
267
268 Model: twoway
269 Type : consistency
270
271 Subjects = 10
272 Raters = 3
273 ICC(C,1) = 0.973
274
275 F-Test, H0: r0 = 0 ; H1: r0 > 0
276 F(9,18) = 108 , p = 8.34e-14
277
278 95%-Confidence Interval for ICC Population Values:
279 0.923 < ICC < 0.993

```

Visoka ICC vrijednost ukazuje na odličnu pouzdanost pri procjeni debljine O-prstena od strane mjeritelja A, B i C. Mjerenjem navedene veličine mikrometarskim vijkom, po proceduri opisanoj u potpoglavlju 4.1, postiže se visok stupanj korelacije i dosljednosti.

Korištenjem programskog paketa *SixSigma* je provedena analiza ponovljivosti i obnovljivosti na temelju rezultata mjerenja iz tablice 4.1 u svrhu određivanja varijabilnosti mjernog sustava. Analiza je prikazana grafički (slika 4.8) i numerički, zajedno s pripadajućom ANOVA analizom s i bez interakcija:

```

282 Complete model (with interaction):
283
284           Df  Sum Sq  Mean Sq  F value  Pr(>F)
285 o ring           9 0.04206 0.004674 107.957 8.34e-14
286 operators         2 0.00007 0.000036   0.843   0.447
287 o_ring:operators 18 0.00078 0.000043   0.846   0.641
288 Repeatability     60 0.00307 0.000051
289 Total            89 0.04599
290
291 alpha for removing interaction: 0.05
292
293
294 Reduced model (without interaction):
295
296           Df  Sum Sq  Mean Sq  F value  Pr(>F)
297 o ring           9 0.04206 0.004674  94.690 <2e-16
298 operators         2 0.00007 0.000036   0.739   0.481
299 Repeatability    78 0.00385 0.000049
300 Total            89 0.04599
301
302 Gage R&R
303
304           VarComp  %Contrib
305 Total Gage R&R 4.935812e-05 8.76
306 Repeatability  4.935812e-05 8.76
307 Reproducibility 0.000000e+00 0.00
308 operators      0.000000e+00 0.00
309 Part-To-Part 5.138195e-04 91.24
310 Total Variation 5.631776e-04 100.00
311
312           StdDev  StudyVar  %StudyVar
313 Total Gage R&R 0.007025533 0.0421532 29.60
314 Repeatability  0.007025533 0.0421532 29.60
315 Reproducibility 0.000000000 0.0000000 0.00
316 operators      0.000000000 0.0000000 0.00
317 Part-To-Part  0.022667586 0.1360055 95.52
318 Total Variation 0.023731363 0.1423882 100.00
319
320 Number of Distinct Categories = 4

```

Iz numeričke analize je vidljivo da je udio varijabilnosti zbog samog sustava u cjelokupnoj varijabilnosti 8,76%, te prevladavajući utjecaj ima varijabilnost između dijelova u udjelu 91,24%. Prema prethodno definiranim kriterijima, mjerni sustav se može smjestiti u kvalitetne mjerne sustave. Navedene tvrdnje su grafički prikazane u gornjem lijevom dijagramu na slici 4.8. Na slici

su O-prsteni označeni brojevima 1 ÷ 10, a mjeritelji slovima A, B i C. Iz R – kontrolne karte se može zaključiti da se rasponi pri mjerenjima svakog O-prstena od strane svakog operatera nalaze unutar kontrolnih granica, što ukazuje na stabilnost procesa. Vrijednosti na \bar{x} – kontrolnoj karti se smiju nalaziti izvan kontrolnih granica jer ukazuju na varijabilnost između mjerenih dijelova. Gornja dva dijagrama na desnoj strani slike (*measurements by o_ring* i *measurements by operators*) prikazuju rasipanja izmjerenih vrijednosti po O-prstenima, odnosno mjeriteljima. Pomoću donjeg desnog dijagram se može zaključiti da nema interakcija između mjeritelja i O-prstena, pošto se vrijednosti mijenjaju po sličnom pravilu.

Proizvodnja O-prstena se može promatrati kao proces s donjom granicom $LSL = 2,32$ mm i gornjom granicom $USL = 2,48$ mm, prema potpoglavlju 2.2. Statistički pokazatelj odnosa varijabilnosti mjernog sustava i tolerancijskog polja, prema jednadžbi (4.3) glasi:

$$R\&R/T = \frac{\sigma_{R\&R}}{USL - LSL} \times 100\% = \frac{0.0422}{2,48 - 2,32} \times 100\% = 26\%$$

Varijabilnost, čiji je uzrok ponovljivost i obnovljivost samog mjernog sustava, zauzima 26% širine tolerancijskog polja, što je znatno više nego udio varijabilnosti mjernog sustava u cjelokupnoj varijabilnosti.

4.2.2. Sposobnost sustava pri mjerenju unutarnjeg promjera mikrometarskim vijkom

Postupak mjerenja unutarnjeg promjera mikrometarskim vijkom za unutrašnja mjerenja je opisan u potpoglavlju 4.1. Tablica 4.5 prikazuje prosječne izmjerene vrijednosti veličine unutarnjeg promjera po mjeritelju, a izvedena je iz tablice 4.2. Svaka vrijednost unutar tablice je prosjek triju izmjerenih vrijednosti, a vrijednosti se koriste za procjenu ICC-a.

Rezultat procjene ICC-a pomoću programskog paketa *irr* i programskog okruženja *R* je sljedeći:

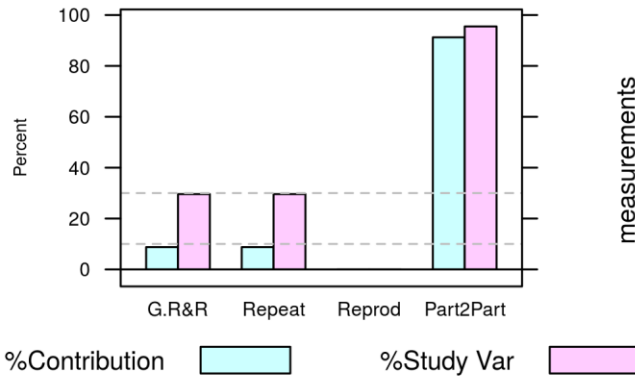
```

330 Single Score Intraclass Correlation
331
332 Model: twoway
333 Type : consistency
334
335 Subjects = 10
336 Raters = 3
337 ICC(C,1) = 0.463
338
339 F-Test, H0: r0 = 0 ; H1: r0 > 0
340 F(9,18) = 3.59 , p = 0.0101
341
342 95%-Confidence Interval for ICC Population Values:
343 0.07 < ICC < 0.804

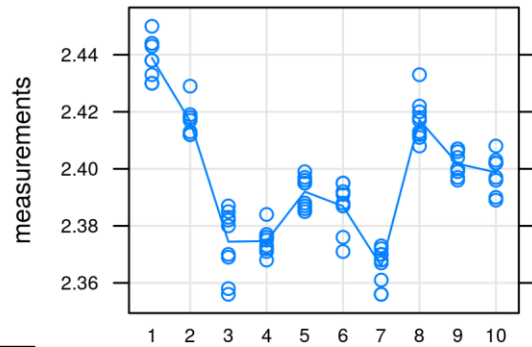
```

O-Ring Cross-section (Micrometer)

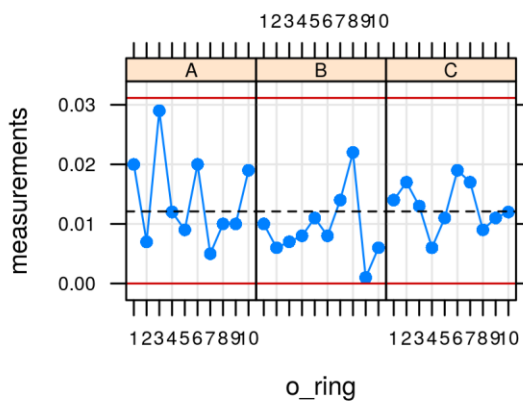
Components of Variation



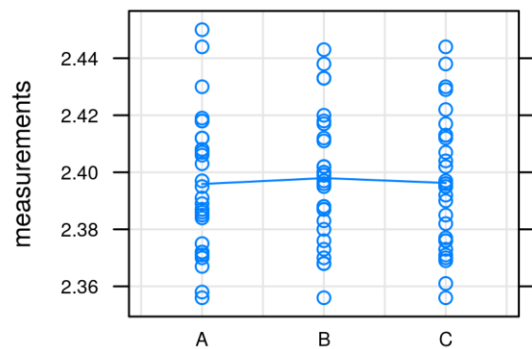
measurements by o_ring



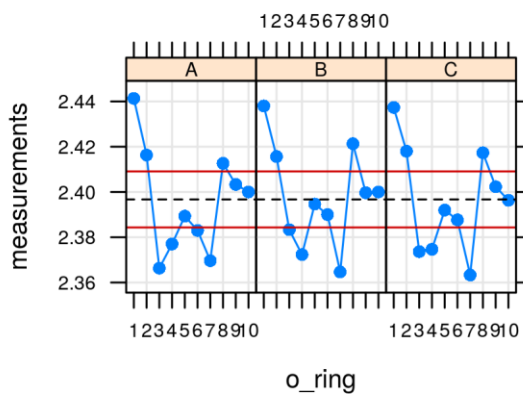
R Chart by operators



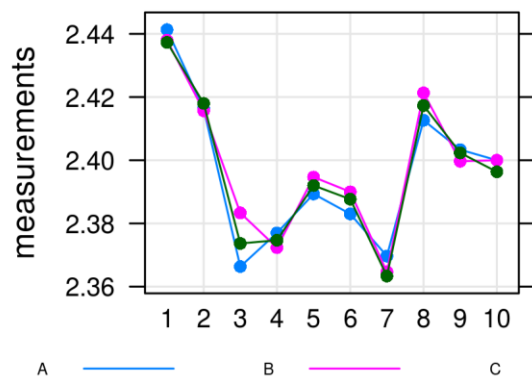
measurements by operators



\bar{x} Chart by appr



o_ring:operators Interaction



Six Sigma O-Ring R&R

Slika 4.8. Studija ponovljivosti i obnovljivosti pri mjerenju poprečnog presjeka O-prstena mikrometarskim vijkom

Tablica 4.5. Prosječne izmjerene vrijednosti unutarnjeg promjera [mm] O-prstena po mjeritelju

O-prsten	A	B	C
15,8 x 24_1	15,867	15,750	15,747
15,8 x 24_2	15,813	15,677	15,710
15,8 x 24_3	15,823	15,750	15,697
15,8 x 24_4	15,843	15,760	15,763
15,8 x 24_5	15,763	15,710	15,703
15,8 x 24_6	15,807	15,750	15,720
15,8 x 24_7	15,817	15,837	15,777
15,8 x 24_8	15,830	15,670	15,693
15,8 x 24_9	15,797	15,673	15,700
15,8 x 24_10	15,863	15,740	15,737

ICC koeficijent u iznosu od 0,463 ukazuje na slabu (granična vrijednost s umjerenom) pouzdanost, odnosno korelaciju i dosljednost između mjeritelja A, B i C.

ANOVA analiza i analiza R&R na temelju mjernih rezultata iz tablice 4.2 su prikazane kao izlaz *SixSigma* paketa:

```

346 Complete model (with interaction):
347
348           Df Sum Sq Mean Sq F value    Pr(>F)
349 o_ring      9 0.0773  0.00859    3.589   0.0101
350 operators   2 0.1781  0.08904   37.190 4.05e-07
351 o_ring:operators 18 0.0431  0.00239    1.337   0.1989
352 Repeatability 60 0.1075  0.00179
353 Total       89 0.4060
354
355 alpha for removing interaction: 0.05
356
357
358 Reduced model (without interaction):
359
360           Df Sum Sq Mean Sq F value    Pr(>F)
361 o_ring      9 0.0773  0.00859    4.452   1e-04
362 operators   2 0.1781  0.08904   46.128 6.01e-14
363 Repeatability 78 0.1506  0.00193
364 Total       89 0.4060
365
366 Gage R&R
367
368           VarComp %Contrib
369 Total Gage R&R 0.0048339791 86.72
370 Repeatability 0.0019302849 34.63

```

371	Reproducibility	0.0029036942	52.09	
372	operators	0.0029036942	52.09	
373	Part-To-Part	0.0007403113	13.28	
374	Total Variation	0.0055742904	100.00	
375				
376		StdDev	StudyVar	%StudyVar
377	Total Gage R&R	0.06952682	0.4171609	93.12
378	Repeatability	0.04393501	0.2636100	58.85
379	Reproducibility	0.05388594	0.3233156	72.17
380	operators	0.05388594	0.3233156	72.17
381	Part-To-Part	0.02720866	0.1632520	36.44
382	Total Variation	0.07466117	0.4479670	100.00
383				
384	Number of Distinct Categories = 1			

Iz analize je vidljivo da prevladavajući utjecaj na ukupnu varijabilnost ima varijabilnost samog mjernog sustava u iznosu od 86,72%. Od tog postotka na obnovljivost (varijabilnost zbog mjerenja od strane različitih mjeritelja) otpada 52,09%, a na ponovljivost 34,63%. Navedene numeričke vrijednosti se mogu grafički vizualizirati pomoću gornjeg lijevog dijagrama na slici 4.9. Varijabilnost mjernog sustava prema *Six Sigma* metodologiji iznosi $\sigma_{R\&R} = 0,417$ mm. Iz *R* – kontrolne karte sa slike 4.9 je vidljivo da rasponi mjerenja mjeritelja A znatno odstupaju od preostalih mjeritelja. Isto se može zaključiti i za prosječne vrijednosti mjerenja prema \bar{x} – kontrolnoj karti.

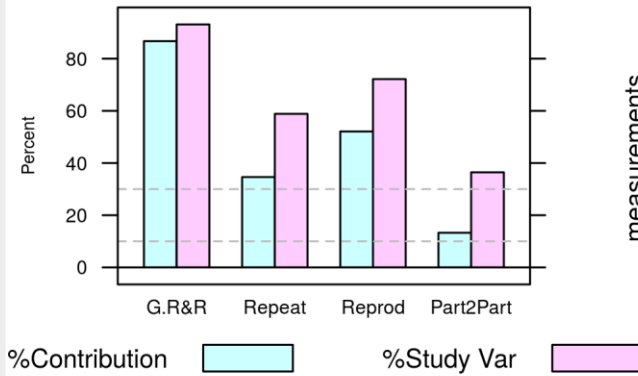
Donja granica vrijednosti veličine unutarnjeg promjera O-prstena, specificirana od strane klijenta, iznosi $LSL = 15,57$ mm, a gornja granica $USL = 16,03$ mm. Tolerancije procesa su opisane u potpoglavlju 2.2. Odnos varijabilnosti mjernog sustava i tolerancijskog polja procesa, prema jednadžbi (4.3) glasi:

$$R\&R/T = \frac{\sigma_{R\&R}}{USL - LSL} \times 100\% = \frac{0,417}{16,03 - 15,57} \times 100\% = 90,7\%$$

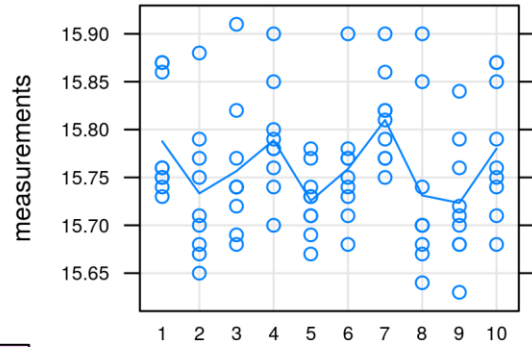
Uzimajući u obzir sve statističke pokazatelje navedene u ovom potpoglavlju, može se zaključiti da navedeni mjerni sustav nije adekvatan za kontrolu veličine unutarnjeg promjera O-prstena.

O-Ring Inner Diameter (Micrometer)

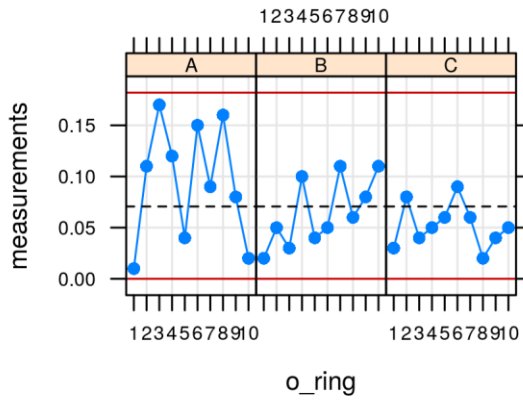
Components of Variation



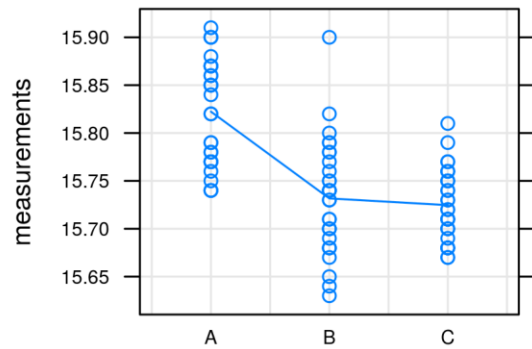
measurements by o_ring



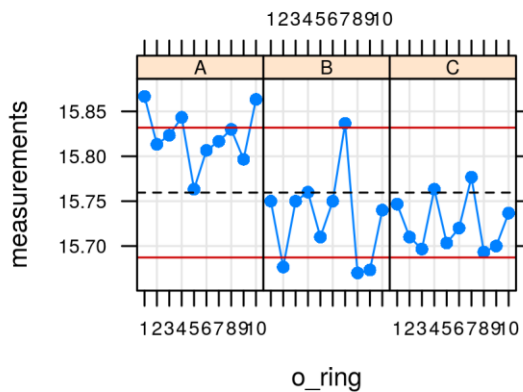
R Chart by operators



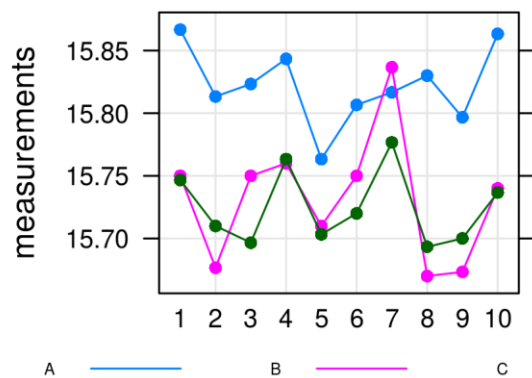
measurements by operators



\bar{x} Chart by appr



o_ring:operators Interaction



Six Sigma O-Ring R&R

Slika 4.9. Studija ponovljivosti i obnovljivosti pri mjerenju unutarnjeg promjera O-prstena mikrometarskim vijkom

4.2.3. Sposobnost sustava pri mjerenju vanjskog promjera mikrometarskim vijkom

U narednom potpoglavlju bit će interpretirane vrijednosti statističkih pokazatelja sposobnosti mjernog sustava kod mjerenja veličine vanjskog promjera mikrometarskim vijkom. Procedura mjerenja je predstavljena u potpoglavlju 4.1. Prosječne izmjerene vrijednosti veličine vanjskog promjera po mjeritelju su izvedene iz tablice 4.3 i prikazane pomoću tablice 4.6. Svaka vrijednost unutar tablice je prosjek triju izmjerenih vrijednosti, a vrijednosti se koriste za procjenu ICC-a.

Tablica 4.6. Prosječne izmjerene vrijednosti vanjskog promjera [mm] O-prstena po mjeritelju

O-prsten	A	B	C
15,8 x 24_1	20,503	20,490	20,561
15,8 x 24_2	20,311	20,430	20,510
15,8 x 24_3	20,369	20,438	20,481
15,8 x 24_4	20,450	20,421	20,481
15,8 x 24_5	20,444	20,433	20,500
15,8 x 24_6	20,359	20,450	20,475
15,8 x 24_7	20,472	20,551	20,541
15,8 x 24_8	20,337	20,387	20,412
15,8 x 24_9	20,351	20,341	20,489
15,8 x 24_10	20,409	20,457	20,524

U nastavku je procijenjen ICC pomoću programskog paketa *irr* i programskog okruženja *R*:

```

394 Single Score Intraclass Correlation
395
396 Model: twoway
397 Type : consistency
398
399 Subjects = 10
400 Raters = 3
401 ICC(C,1) = 0.617
402
403 F-Test, H0: r0 = 0 ; H1: r0 > 0
404 F(9,18) = 5.83 , p = 0.000753
405
406 95%-Confidence Interval for ICC Population Values:
407 0.248 < ICC < 0.873

```

Intraklasni koeficijent korelacije ICC, iznosa 0,617 ukazuje na umjerenu pouzdanost između mjeritelja A, B i C kod mjerenja vanjskog promjera O-prstena mikrometarskim vijkom.

Na temelju rezultata mjerenja iz tablice 4.3, dobivena je R&R studija i prateća ANOVA analiza pomoću *SixSigma* paketa:

```

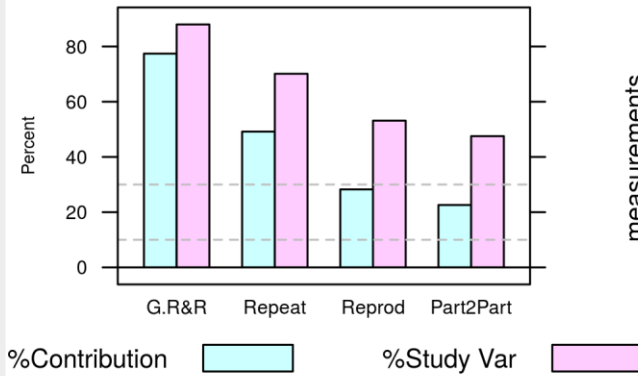
410 Complete model (with interaction):
411
412           Df Sum Sq Mean Sq F value    Pr(>F)
413 o_ring      9  0.1804  0.02005    5.832 0.000753
414 operators   2  0.1423  0.07117   20.707 2.15e-05
415 o_ring:operators 18  0.0619  0.00344    0.850 0.636897
416 Repeatability 60  0.2427  0.00404
417 Total      89  0.6273
418
419 alpha for removing interaction: 0.05
420
421
422 Reduced model (without interaction):
423
424           Df Sum Sq Mean Sq F value    Pr(>F)
425 o_ring      9  0.1804  0.02005    5.134 1.87e-05
426 operators   2  0.1423  0.07117   18.229 3.20e-07
427 Repeatability 78  0.3046  0.00390
428 Total      89  0.6273
429
430 Gage R&R
431
432           VarComp %Contrib
433 Total Gage R&R    0.006146849 77.41
434 Repeatability    0.003904510 49.17
435 Reproducibility 0.002242339 28.24
436 operators        0.002242339 28.24
437 Part-To-Part     0.001793476 22.59
438 Total Variation    0.007940325 100.00
439
440           StdDev StudyVar %StudyVar
441 Total Gage R&R    0.07840184 0.4704110 87.98
442 Repeatability      0.06248608 0.3749165 70.12
443 Reproducibility    0.04735334 0.2841200 53.14
444 operators          0.04735334 0.2841200 53.14
445 Part-To-Part       0.04234946 0.2540967 47.53
446 Total Variation    0.08910850 0.5346510 100.00
447
448 Number of Distinct Categories = 1

```

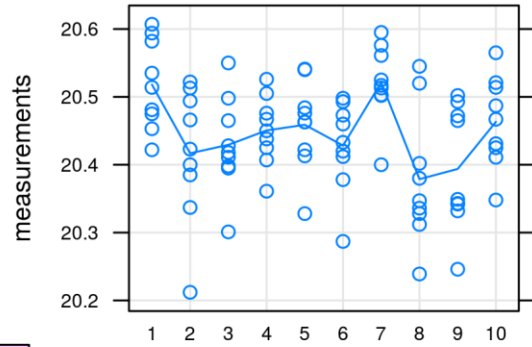
Kao i kod mjerenja unutarnjeg promjera, mjerni sustav ima najveći utjecaj na varijabilnost od 77,41%. No, u ovom slučaju je glavni uzrok varijabilnost zbog ponovljivosti (49,17%), odnosno ponavljanja mjerenja od strane istog mjeritelja u istim uvjetima. Navedeno se može vizualno prikazati gornjim lijevim dijagramom na slici 4.10. Tu tvrdnju potvrđuje i iznimni slučaj na *R* – kontrolnoj karti na slici. Varijabilnost mjernog sustava iznosi $\sigma_{R\&R} = 0,470$ mm.

O-Ring Outer Diameter (Micrometer)

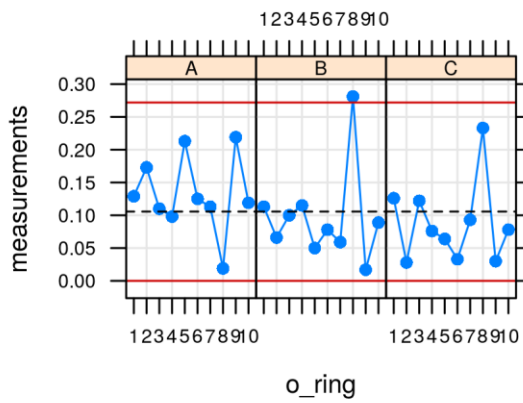
Components of Variation



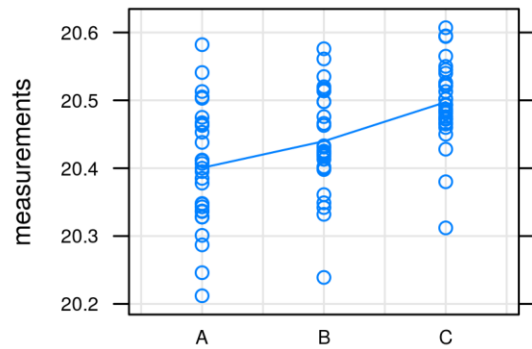
measurements by o_ring



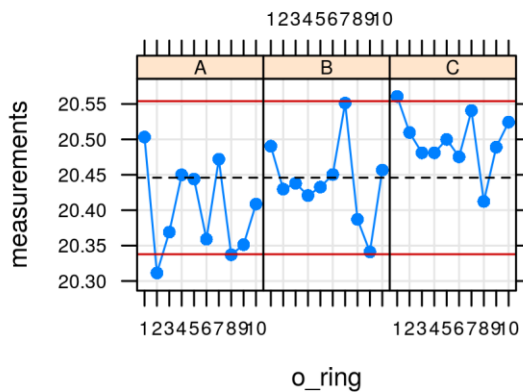
R Chart by operators



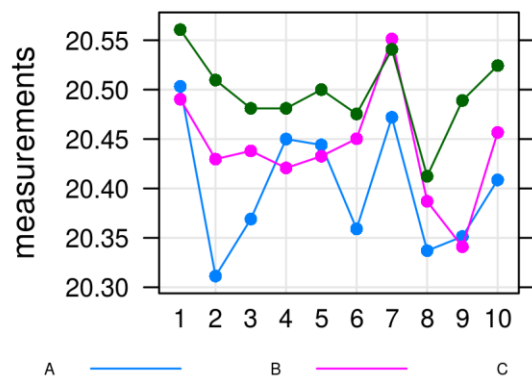
measurements by operators



\bar{x} Chart by appr



o_ring:operators Interaction



Six Sigma O-Ring R&R

Slika 4.10. Studija ponovljivosti i obnovljivosti pri mjerenju vanjskog promjera O-prstena mikrometarskim vijkom

Proizvodnja O-prstena se može prikazati kao proces s donjom granicom $LSL = 20,21$ mm i gornjom granicom $USL = 20,99$ mm. Tolerancije proizvoda su pojašnjene u potpoglavlju 2.2. Varijabilnost mjernog sustava u odnosu na tolerancijsko polje procesa, prema izrazu (4.3) glasi:

$$R\&R/T = \frac{\sigma_{R\&R}}{USL - LSL} \times 100\% = \frac{0.470}{20,99 - 20,21} \times 100\% = 60,3\%$$

Pomoću pokazatelja navedenih u ovom potpoglavlju se može zaključiti da opisani mjerni sustav ima visok stupanj varijabilnosti i nije primjenjiv za kontrolu dimenzija vanjskog promjera O-prstena.

4.3. Postupak i rezultati mjerenja proizvoda u optičkom mjernom sustavu

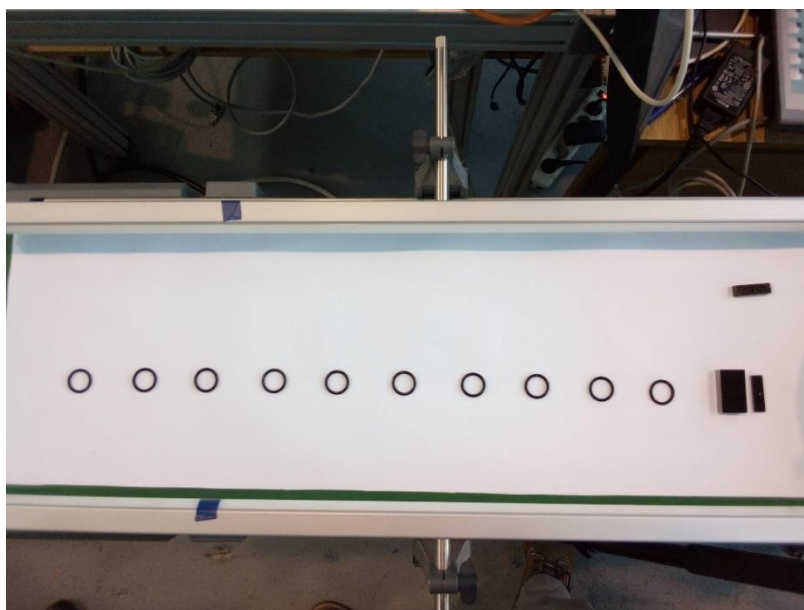
U optičkom (automatiziranom) mjernom sustavu, opisanom u potpoglavlju 2.3, mjereno je 10 identičnih O-prstena, kao i u potpoglavlju 4.1. O-prsteni su poredani na bijelu podlogu po redu, od 15,8 x 24_1 do 15,8 x 24_10, optički sustav je kalibriran, te je izvršeno jedno mjerenja na svim proizvodima. Navedeni postupak je ponovljen 9 puta, kako bi broj mjerenja bio u skladu s ručnim mjerenjima. Na taj način je izvršeno 90 (9×10) mjerenja. Prednost automatiziranog mjernog sustava je u tome, što je istovremeno moguće mjeriti više veličina. Kako je opisano u potpoglavlju 2.2.2, mjerene veličine su sljedeće:

- Opseg (*Perimeter*),
- Maksimalni Feret-ov promjer (*Feret Max*),
- Minimalni Feret-ov promjer (*Feret Min*),
- Maksimalni polumjer (*Radius Max*),
- Minimalni polumjer (*Radius Min*),
- Prosječni polumjer (*Radius Mean*),
- Standardna devijacija polumjera (*Radius StdDev*),
- Podcbeck-ov oblik kruga (*PodcbeckShapes Circle*),
- Podcbeck-ov oblik elipse (*PodcbeckShapes Ellipse*),
- Konveksnost (*Convexity*),
- Koordinate centra (*Center dim0, Center dim1*).

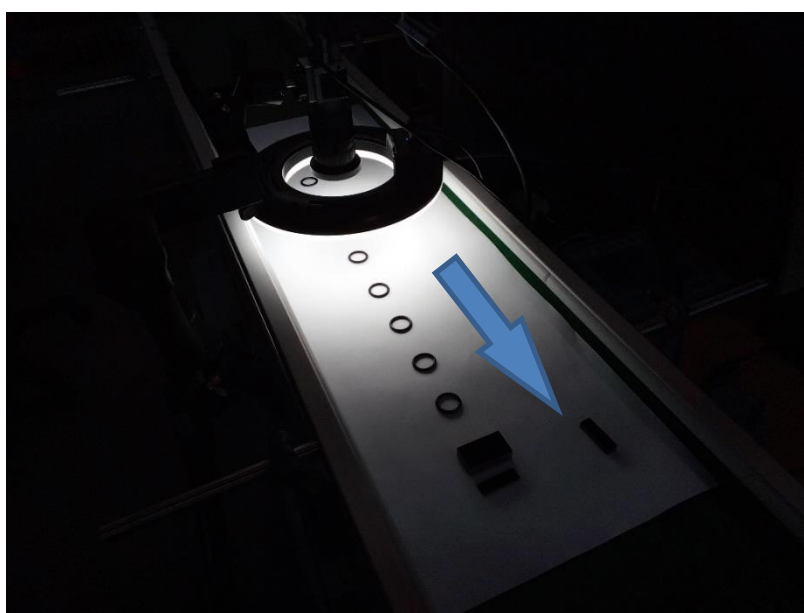
Nakon što je optički sustav postavljen prema potpoglavlju 2.3.5, na bijelu papirnatu podlogu su postavljeni O-prsteni i etaloni za kalibraciju po redu, što je prikazano na slici 4.11.

Postupak kalibracije i mjerenja prati korake opisane u poglavlju 3:

1. Pozicioniranje etalona u centralni dio područja interesa slike (*Region of interest, ROI*),
2. Prilagođavanje parametara kalibracijskog algoritma i samo umjeravanje sustava,
3. Pozicioniranje prvog O-prstena u centralni dio ROI,
4. Prilagođavanje parametara mjernog algoritma,
5. Automatizirano mjerenje svih 10 O-prstena po redu, uz pomicanje pokretne trake, kao što je prikazano na slici 4.12,
6. Spremanje rezultata mjerenja u .tsv (*Tab-separated values*) formatu datoteke.



Slika 4.11. Postavljanje O-prstena i etalona za kalibraciju prije automatiziranog mjerenja



Slika 4.12. Automatizirano mjerenje O-prstena uz pomicanje pokretne trake

Za umjeravanje optičkog sustava korišteni su sljedeći čelični etaloni:

- Etalon 20 mm:
 - Visina: 9,010 mm,
 - Širina: 20,000 mm,
- Etalon 2 mm:
 - Visina: 2,000 mm,
 - Širina: 8,950 mm.

Postupak umjeravanja etalona opisan je u potpoglavlju 2.3.4. Prethodno opisanim optičkim mjernim sustavom nije moguće direktno mjeriti poprečni presjek (debljinu) O-prstena, već je to ulazni parametar za kalibraciju sustava. Parametri kalibracije za svako mjerenje, s upisanim rednim brojem mjerenja se nalaze na slici 4.13. Na slici su prikazani parametri koji se odnose na etalon (*Standard*) i mjereni O-prsten (*Measured Object*). Rezultat umjeravanja je mm/px omjer na visini mjenog objekta (*Height*).

Izlazni rezultati programskog rješenja svih devet mjerenja prikazani su u prilogu 5.

4.4. Analiza sposobnosti optičkog mjernog sustava

Pri analizi sposobnosti optičkog (automatiziranog) mjernog sustava korišten je programski jezik R [24] s paketima *SixSigma* [25] i *irr* [26]. Navedeni paketi, *Six Sigma* metodologija i svi statistički pokazatelji sposobnosti sustava su definirani u potpoglavlju 4.2. Programski kod za primjenu metodologije se nalazi u prilogu 2.

Iz rezultata mjerenja, prema prilogu 5, izdvojene su prosječne vrijednosti polumjera (*Radius Mean*) i udvostručene, kako bi se dobila procjena veličine unutarnjeg, odnosno vanjskog promjera O-prstena. Odabrane su navedene vrijednosti kako bi bilo moguće usporediti automatizirani i ručni mjerni sustav. Tablica 4.7 prikazuje navedene izdvojene vrijednosti unutarnjeg promjera, dobivene mjerenjem pomoću optičkog sustava. Unutar tablice su izračunate i prosječne vrijednosti, te standardne devijacije izmjerenih vrijednosti unutarnjeg promjera svakog O-prstena pojedinačno, kao i svih O-prstena zajedno. Mjerenja unutar tablice su također povezana i s pripadajućim rezultatima kalibracije (mm/px omjer). U tablici 4.8 su vidljive izdvojene prosječne vrijednosti (*Radius Mean*) vanjskog promjera O-prstena, te prosjeci i standardne devijacije po O-prstenu.

Parameter Tree		Parameter Tree	
Parameter	Value	Parameter	Value
Calibration Parameters 1		Calibration Parameters 2	
▼ Standard		▼ Standard	
Height	9.01 mm	Height	9.01 mm
Working Distance	103.687 mm	Working Distance	112.544 mm
mm/px Ratio	0.03206524 mm/px	mm/px Ratio	0.032206119 mm/px
▼ Measured Object		▼ Measured Object	
Height	2.39671 mm	Height	2.39671 mm
mm/px Ratio	0.034110403 mm/px	mm/px Ratio	0.03409861 mm/px
Parameter Tree		Parameter Tree	
Parameter	Value	Parameter	Value
Calibration Parameters 3		Calibration Parameters 4	
▼ Standard		▼ Standard	
Height	9.01 mm	Height	9.01 mm
Working Distance	102.61 mm	Working Distance	112.996 mm
mm/px Ratio	0.032051282 mm/px	mm/px Ratio	0.032102729 mm/px
▼ Measured Object		▼ Measured Object	
Height	2.39671 mm	Height	2.39671 mm
mm/px Ratio	0.034117015 mm/px	mm/px Ratio	0.033981596 mm/px
Parameter Tree		Parameter Tree	
Parameter	Value	Parameter	Value
Calibration Parameters 5		Calibration Parameters 6	
▼ Standard		▼ Standard	
Height	9.01 mm	Height	9.01 mm
Working Distance	129.394 mm	Working Distance	114.496 mm
mm/px Ratio	0.032385355 mm/px	mm/px Ratio	0.032102729 mm/px
▼ Measured Object		▼ Measured Object	
Height	2.39671 mm	Height	2.39671 mm
mm/px Ratio	0.034040563 mm/px	mm/px Ratio	0.033956976 mm/px
Parameter Tree		Parameter Tree	
Parameter	Value	Parameter	Value
Calibration Parameters 7		Calibration Parameters 8	
▼ Standard		▼ Standard	
Height	9.01 mm	Height	9.01 mm
Working Distance	120.87 mm	Working Distance	104.339 mm
mm/px Ratio	0.03227597 mm/px	mm/px Ratio	0.032 mm/px
▼ Measured Object		▼ Measured Object	
Height	2.39671 mm	Height	2.39671 mm
mm/px Ratio	0.034041918 mm/px	mm/px Ratio	0.034028243 mm/px
Parameter Tree		Parameter Tree	
Parameter	Value	Parameter	Value
Calibration Parameters 9			
▼ Standard			
Height	9.01 mm		
Working Distance	125.76 mm		
mm/px Ratio	0.032154341 mm/px		
▼ Measured Object			
Height	2.39671 mm		
mm/px Ratio	0.033845223 mm/px		

Slika 4.13. Parametri kalibracije optičkog mjernog sustava s brojem mjerenja (gore desno)

Tablica 4.7. Izmjerene prosječne vrijednosti unutarnjeg promjera ($\text{Radius Mean} \times 2$) [mm] O-prstena pomoću optičkog sustava

mm/px omjer	Broj mjerenja	15,8 x		15,8 x		15,8 x		15,8 x		15,8 x		15,8 x	
		24_1	24_2	24_3	24_4	24_5	24_6	24_7	24_8	24_9	24_10		
0,03411	1	15,872	15,804	15,725	15,757	15,702	15,821	15,897	15,911	15,753	15,756		
0,03410	2	16,089	16,496	15,739	15,862	15,748	15,690	15,818	15,859	15,679	15,759		
0,03412	3	16,094	15,803	15,736	15,758	15,777	15,700	15,829	15,938	15,720	15,766		
0,03398	4	15,808	15,747	15,663	15,697	15,682	15,637	15,750	15,773	15,792	15,704		
0,03404	5	15,831	16,000	15,698	15,719	15,730	15,656	15,805	15,820	15,653	15,818		
0,03396	6	15,786	15,715	15,662	15,679	15,675	15,625	15,740	15,893	15,615	15,704		
0,03404	7	15,838	15,981	15,709	15,723	15,726	15,658	15,795	15,736	15,671	15,715		
0,03403	8	16,051	15,767	15,697	15,720	15,720	15,655	15,786	15,759	15,668	15,736		
0,03385	9	15,882	15,803	15,607	15,765	15,624	15,570	15,973	15,661	15,581	15,635		
	Prosjek:	15,917	15,902	15,693	15,742	15,709	15,668	15,821	15,817	15,681	15,733		
	St. dev.:	0,1249	0,2438	0,0426	0,0535	0,0451	0,0689	0,0731	0,0917	0,0657	0,0513		
Prosjek	15,768												
St. dev.	0,131												

Tablica 4.8. Izmjerene prosječne vrijednosti vanjskog promjera (*Radius Mean* × 2) [mm] O-prstena pomoću optičkog sustava

mm/px omjer	Broj mjerenja	15,8 x		15,8 x		15,8 x		15,8 x		15,8 x		15,8 x	
		24_1	24_2	24_3	24_4	24_5	24_6	24_7	24_8	24_9	24_10		
0,03411	1	20,797	20,632	20,594	20,554	20,554	20,719	20,588	20,329	20,529	20,529	20,529	20,529
0,03410	2	20,555	20,424	20,584	20,548	20,591	20,571	20,715	20,610	20,518	20,538	20,538	20,538
0,03412	3	20,571	20,642	20,607	20,552	20,616	20,577	20,730	20,610	20,564	20,539	20,539	20,539
0,03398	4	20,737	20,559	20,516	20,470	20,495	20,500	20,630	20,525	20,149	20,475	20,475	20,475
0,03404	5	20,769	20,019	20,557	20,507	20,560	20,518	20,671	20,567	20,487	20,425	20,425	20,425
0,03396	6	20,712	20,526	20,508	20,454	20,478	20,478	20,612	20,524	20,431	20,456	20,456	20,456
0,03404	7	20,775	20,345	20,572	20,504	20,560	20,531	20,682	20,568	20,500	20,495	20,495	20,495
0,03403	8	20,535	20,587	20,565	20,510	20,555	20,530	20,676	20,588	20,508	20,495	20,495	20,495
0,03385	9	20,480	20,217	20,442	20,388	20,419	20,415	20,408	20,468	20,391	20,387	20,387	20,387
	Prosjek:	20,659	20,439	20,549	20,499	20,536	20,538	20,635	20,532	20,453	20,482	20,482	20,482
	St. dev.:	0,1221	0,2113	0,0519	0,0542	0,0611	0,0836	0,0966	0,0888	0,1254	0,0523	0,0523	0,0523
Prosjek	20,532												
St. dev.	0,121												

4.4.1. Sposobnost optičkog sustava pri mjerenju unutarnjeg promjera O-prstena

Intraklasni koeficijent korelacije je korišten za određivanje pouzdanosti između različitih mjerenja istog O-prstena. Na temelju vrijednosti iz tablice 4.7, te korištenjem paketa *irr* i programskog okruženja *R*, dobiveni su sljedeći rezultati:

```

505 Single Score Intraclass Correlation
506
507 Model: twoway
508 Type : consistency
509
510 Subjects = 10
511 Raters = 9
512 ICC(C,1) = 0.454
513
514 F-Test, H0: r0 = 0 ; H1: r0 > 0
515 F(9,72) = 8.49 , p = 1.57e-08
516
517 95%-Confidence Interval for ICC Population Values:
518 0.231 < ICC < 0.758

```

Prema kriterijima iz potpoglavlja 4.2, ICC = 0,454 ukazuje na slabu korelaciju i dosljednost između različitih mjerenja iste veličine.

ANOVA i R&R analiza rezultata mjerenja su provedene korištenjem *SixSigma* paketa:

```

521 One-way ANOVA (single appraiser):
522
523           Df Sum Sq Mean Sq F value Pr(>F)
524 o_ring      9 0.6642 0.07380   6.885 2.81e-07
525 Repeatability 80 0.8574 0.01072
526 Total      89 1.5216
527
528 Gage R&R
529
530           VarComp %Contrib
531 Total Gage R&R 0.010718046 60.46
532 Repeatability 0.010718046 60.46
533 Part-To-Part 0.007008617 39.54
534 Total Variation 0.017726663 100.00
535
536           StdDev StudyVar %StudyVar
537 Total Gage R&R 0.10352799 0.6211680 77.76
538 Repeatability 0.10352799 0.6211680 77.76
539 Part-To-Part 0.08371749 0.5023049 62.88
540 Total Variation 0.13314152 0.7988491 100.00
541
542 Number of Distinct Categories = 1

```

Pošto u automatiziranom sustavu ne postoji utjecaj mjeritelja na ukupnu varijabilnost, moguće je promatrati samo varijabilnost zbog ponovljivosti (*Repeatability*). Doprinos varijabilnosti sustava

ukupnoj varijabilnosti je 60,46%. Navedeni iznos je previsok, te ga je potrebno smanjiti kako bi se optički sustav mogao koristiti u svrhu mjerenja unutarnjeg promjera O-prstena. Varijabilnost sustava iznosi $\sigma_{R\&R} = 0,621$ mm. Vizualizacija R&R studije je prikazana na slici 4.14. Gornji lijevi dijagram također potvrđuje veliku varijabilnost sustava. Također, sa R – kontrolne karte na slici je vidljiva iznimka u rasponu pri mjerenju O-prstena 15,8 x 24_2. Iznimku u mjerenju potvrđuje i gornji desni dijagram na slici.

Uzimajući prosječne vrijednosti unutarnjeg promjera O-prstena iz tablice 4.7, proizvodnja O-prstena se može prikazati kao proces s donjom granicom $LSL = 15,57$ mm i gornjom granicom $USL = 16,03$ mm. Tolerancije procesa su definirane u potpoglavlju 2.2. Odnos varijabilnosti mjernog sustava i tolerancijskog polja procesa, prema jednadžbi (4.3) glasi:

$$R\&R/T = \frac{\sigma_{R\&R}}{USL - LSL} \times 100\% = \frac{0.621}{16,03 - 15,57} \times 100\% = 135,0\%$$

Varijabilnost optičkog mjernog sustava je znatno visoka, te ju je potrebno smanjiti prije primjene sustava za kontrolu veličine unutarnjeg promjera O-prstena.

4.4.2. Sposobnost optičkog sustava pri mjerenju vanjskog promjera O-prstena

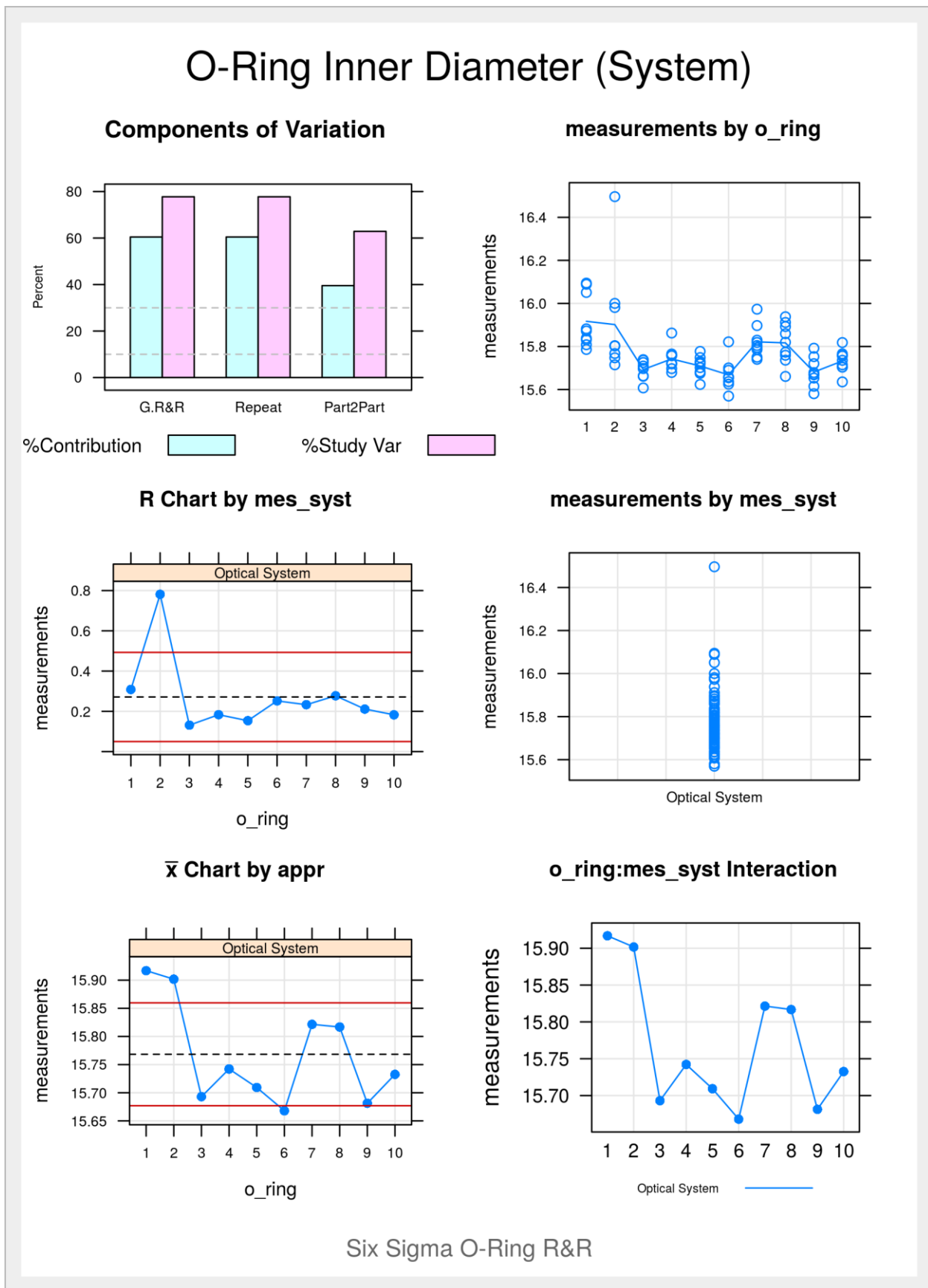
Procjena ICC-a između 9 mjerenja je izvršena na temelju izmjerenih vrijednosti vanjskog promjera proizvoda prema tablici 4.8. Izlazni rezultati paketa *irr* su sljedeći:

```

458 Single Score Intraclass Correlation
459
460 Model: twoway
461 Type : consistency
462
463 Subjects = 10
464 Raters = 9
465 ICC(C,1) = 0.325
466
467 F-Test, H0: r0 = 0 ; H1: r0 > 0
468 F(9,72) = 5.33 , p = 1.46e-05
469
470 95%-Confidence Interval for ICC Population Values:
471 0.128 < ICC < 0.657

```

ICC vrijednost od 0,325 upućuje na slabu korelaciju i dosljednost između različitih mjerenja vanjskog promjera O-prstena u automatiziranom sustavu.



Slika 4.14. Studija ponovljivosti pri mjerenju unutarnjeg promjera O-prstena pomoću optičkog mjernog sustava

Za određivanje varijabilnosti optičkog mjernog sustava, provedena je ANOVA analiza:

```

474 One-way ANOVA (single appraiser):
475
476           Df Sum Sq Mean Sq F value    Pr(>F)
477 o_ring      9  0.4094  0.04549    4.076 0.000247
478 Repeatability 80  0.8927  0.01116
479 Total      89  1.3021
480
481 Gage R&R
482
483           VarComp %Contrib
484 Total Gage R&R 0.011158794    74.53
485 Repeatability 0.011158794    74.53
486 Part-To-Part 0.003814052    25.47
487 Total Variation 0.014972846   100.00
488
489           StdDev StudyVar %StudyVar
490 Total Gage R&R 0.10563519 0.6338112    86.33
491 Repeatability 0.10563519 0.6338112    86.33
492 Part-To-Part 0.06175801 0.3705481    50.47
493 Total Variation 0.12236358 0.7341815   100.00
494
495 Number of Distinct Categories = 1

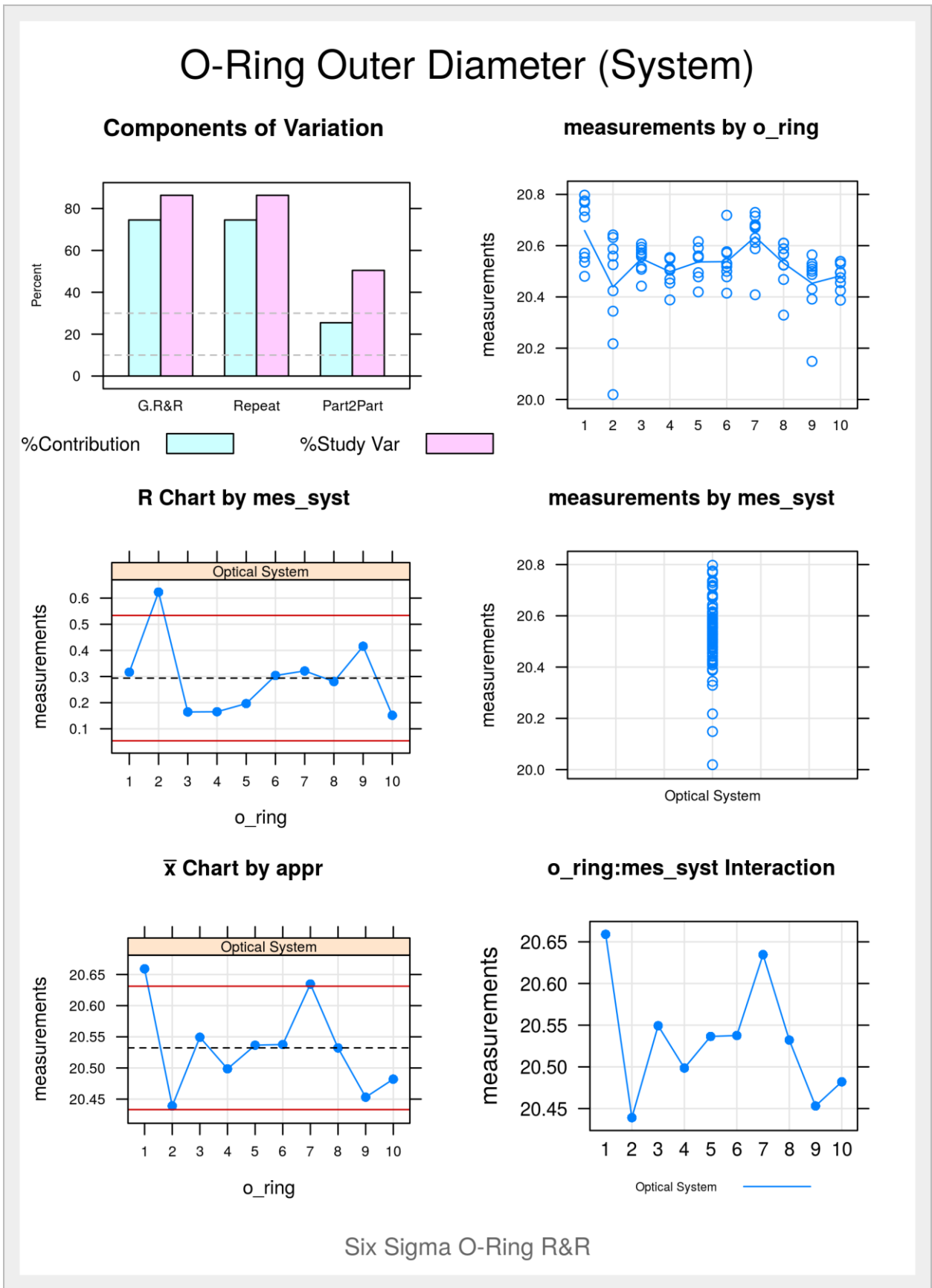
```

Varijabilnost sustava ima velik udio od 74,53% u ukupnoj varijabilnosti, a po apsolutnoj vrijednosti iznosi $\sigma_{R\&R} = 0,634$ mm. Kriterij od maksimalnog udjela u totalnoj varijabilnosti od 30% nije zadovoljen. Grafički prikaz R&R analize se nalazi na slici 4.15. Varijabilnost zbog samog sustava i zbog razlike između dijelova je prikazana na gornjem lijevom dijagramu na slici. Kao i kod analize unutarnjeg promjera, vidljiva je iznimka na R – kontrolnoj karti. Raspon između različitih mjerenja vanjskog promjera O-prstena 15,8 x 24_2 je znatno veći od ostalih raspona među mjerenjima iste veličine.

U procesu proizvodnje, vrijednost veličine vanjskog promjera O-prstena je potrebno održavati između donje granice $LSL = 20,21$ mm i gornje granice $USL = 20,99$ mm. Odnos varijabilnosti mjernog sustava i tolerancijskog polja procesa, prema jednadžbi (4.3) glasi:

$$R\&R/T = \frac{\sigma_{R\&R}}{USL - LSL} \times 100\% = \frac{0.634}{20,99 - 20,21} \times 100\% = 81,3\%$$

Varijabilnost automatiziranog optičkog sustava je i u ovom slučaju vrlo visoka, te je potrebno povisiti kvalitetu sustava prije njegove implementacije.



Slika 4.15. Studija ponovljivosti pri mjerenju vanjskog promjera O-prstena pomoću optičkog mjernog sustava

4.5. Analiza osjetljivosti optičkog mjernog sustava

Jedna od bitnih karakteristika automatiziranog mjernog sustava je njegova osjetljivost na kvalitetu kalibracije. Kalibracija se provodi pomoću dva etalona i jednadžbe optičke leće prema proceduri iz potpoglavlja 2.3.4. Rezultat kalibracije je mm/px omjer na visini mjerenog predmeta, tj. O-prstena. Pomoću omjera se svakom pikselu sa slike dodjeljuje fizikalna jedinica od jednog milimetra.

Kako bi se procijenila osjetljivost mjernog sustava, prvo je određena korelacija između mm/px omjer-a i izmjerenih vrijednosti promjera za svaki O-prsten na temelju tablica 4.7 i 4.8. Pearsonov koeficijent korelacije i pripadajuće p -vrijednosti su izračunati pomoću programskog okruženja *R* i paketa *Hmisc* [29], te prikazani u tablici 4.9. U tablici su dodatno istaknute vrijednosti korelacije O-prstena, gdje je p -vrijednost $\leq 0,05$, odnosno, korelacija je statistički signifikantna. Vrijednosti promjera otprilike 50% O-prstena su jako rastuće korelirani s porastom mm/px omjer-a, što bi u idealnom slučaju bilo očekivano za sve O-prstene. U slučaju O-prstena 15,8 x 24_7, izmjerene vrijednosti unutarnjeg promjera čak opadaju s porastom mm/px omjer-a, što se protivi logici procesa mjerenja.

Tablica 4.9. Pearsonov koeficijent korelacije i pripadajuće p -vrijednosti između rezultata kalibracije i izmjerenih vrijednosti promjera O-prstena

O-prsten	Unutarnji promjer		Vanjski promjer	
	Pearsonov koeficijent korelacije	p -vrijednost	Pearsonov koeficijent korelacije	p -vrijednost
15,8 x 24_1	0,493	$1,78 \times 10^{-1}$	0,287	$4,54 \times 10^{-1}$
15,8 x 24_2	0,389	$3,01 \times 10^{-1}$	0,375	$3,20 \times 10^{-1}$
15,8 x 24_3	0,988	$5,98 \times 10^{-7}$	0,987	$7,31 \times 10^{-7}$
15,8 x 24_4	0,347	$3,60 \times 10^{-1}$	0,997	$7,65 \times 10^{-9}$
15,8 x 24_5	0,893	$1,17 \times 10^{-3}$	0,951	$8,20 \times 10^{-5}$
15,8 x 24_6	0,819	$6,96 \times 10^{-3}$	0,839	$4,66 \times 10^{-3}$
15,8 x 24_7	-0,246	$5,23 \times 10^{-1}$	0,812	$7,86 \times 10^{-3}$
15,8 x 24_8	0,715	$3,02 \times 10^{-2}$	0,149	$7,03 \times 10^{-1}$
15,8 x 24_9	0,583	$9,92 \times 10^{-2}$	0,560	$1,17 \times 10^{-1}$
15,8 x 24_10	0,803	$9,14 \times 10^{-3}$	0,875	$2,00 \times 10^{-3}$

Pri procjeni osjetljivosti, nad istim podacima iz tablica 4.7 i 4.8 izvršena je linearna regresija, kako bi se opisala linearna ovisnost izmjerenih vrijednosti po svakom O-prstenu i rezultata kalibracije. Koeficijenti smjera pravca definiraju osjetljivost mjernog sustava na kalibraciju, te su prikazani u tablici 4.10.

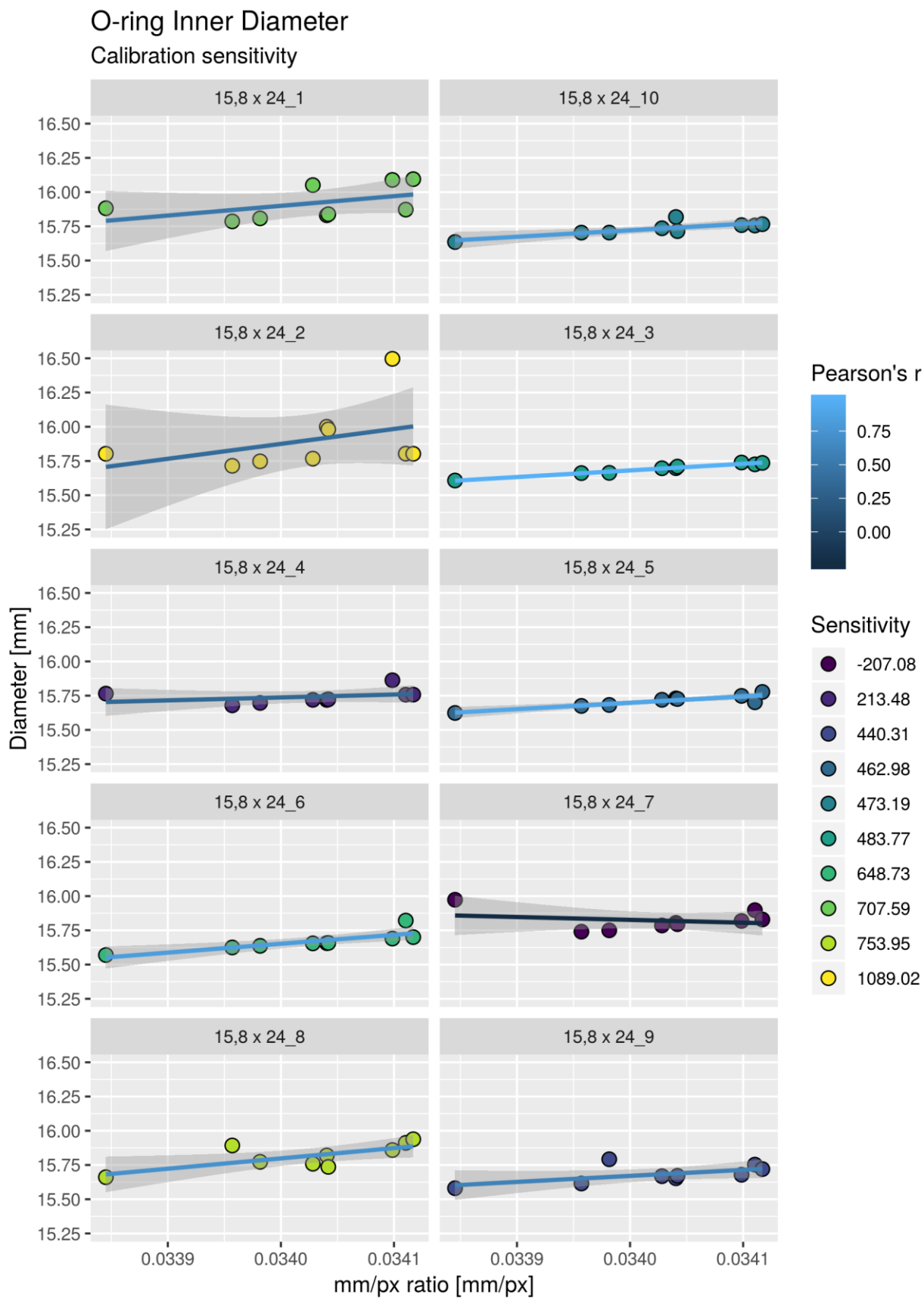
Tablica 4.10. Osjetljivost optičkog mjernog sustava na kalibraciju

O-prsten	Unutarnji promjer	Vanjski promjer
	Osjetljivost na kalibraciju [1/px]	Osjetljivost na kalibraciju [1/px]
15,8 x 24_1	707,59	402,72
15,8 x 24_2	1089,02	910,26
15,8 x 24_3	483,78	589,46
15,8 x 24_4	213,48	621,05
15,8 x 24_5	462,98	668,13
15,8 x 24_6	648,73	806,97
15,8 x 24_7	-207,08	901,25
15,8 x 24_8	753,95	151,75
15,8 x 24_9	440,31	806,59
15,8 x 24_10	473,19	526,16

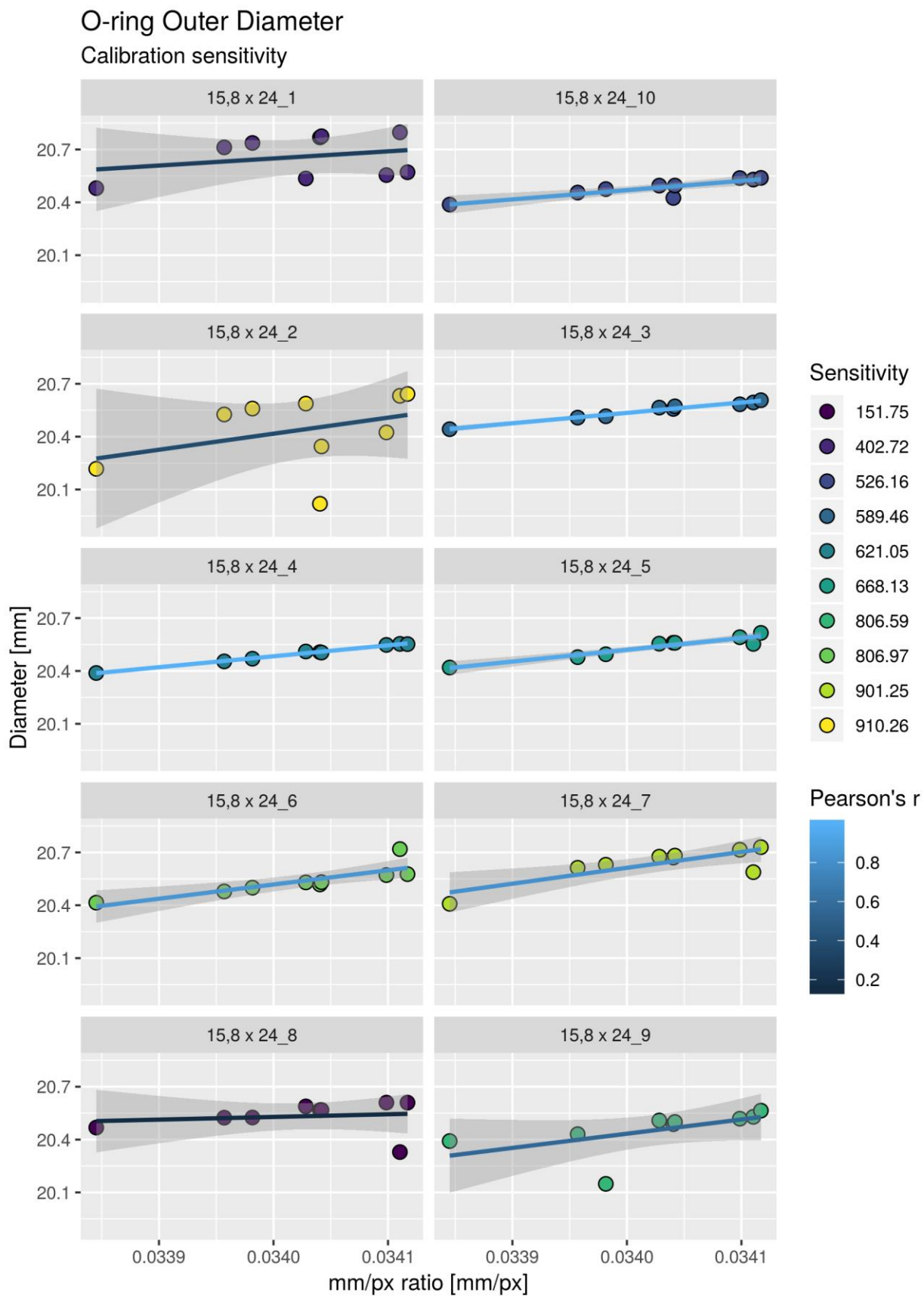
Vizualni prikaz koeficijenata korelacije iz tablice 4.9 i osjetljivosti mjernog sustava iz tablice 4.10 dan je na slici 4.16 u slučaju mjerenja unutarnjeg promjera, odnosno na slici 4.17 u slučaju mjerenja vanjskog promjera. Slike su dobivene pomoću programskog okruženja *R* i programskog paketa *ggplot2* [30]. Na slikama su prikazane i regresijske krivulje, te 95%-tni intervali povjerenja. Boja regresijske krivulje označava iznos Pearsonovog koeficijenta korelacije, tj. svjetlija nijansa plave boje znači jaču korelaciju, a tamnija slabiju. Točke na svakom dijagramu su obojane na način da reflektiraju pripadajuću osjetljivost, odnosno nagib regresijskog pravca.

Raspon između najveće i najmanje vrijednosti mm/px omjer-a je 0,00027 mm/px. Uzimajući najvišu moguću osjetljivost (najgori slučaj) iz tablice 4.10 u iznosu od $\sim 1000 px^{-1}$, može se zaključiti sljedeće:

Greška umjeravanja sustava od 0,3 μm uzrokuje grešku rezultata mjerenja od 0,3 mm.



Slika 4.16. Osjetljivost izmjerenih vrijednosti veličine unutarnjeg promjera O-prstena na kvalitetu kalibracije mjernog sustava



Slika 4.17. Osjetljivost izmjerenih vrijednosti veličine vanjskog promjera O-prstena na kvalitetu kalibracije mjernog sustava

4.6. Usporedba optičkog (automatiziranog) i ručnog mjernog sustava

Pomoću analiza provedenih u potpoglavljima 4.2.2, 4.2.3, 4.4.1 i 4.4.2 izrađena je tablica 4.11, iz koje je vidljivo da su varijabilnosti unutar optičkog sustava više od varijabilnosti pri mjerenju mikrometarskim vijkom. Jedini mjerni sustav koji zadovoljava sve kriterije je onaj za mjerenje poprečnog presjeka O-prstena mikrometarskim vijkom.

Tablica 4.11. Usporedba varijabilnosti i korelacija automatiziranog i ručnog mjernog sustava

Mjerni sustav:	Optički			Ručni (mikrometarski vijak)		
	%R&R [%]	R&R/T [%]	ICC(C,1) [/]	%R&R [%]	R&R/T [%]	ICC(C,1) [/]
Unutarnji promjer	60,5	135,0	0,454	86,7	90,7	0,463
Vanjski promjer	74,5	81,3	0,325	77,4	60,3	0,617
Poprečni presjek	-	-	-	8,8	26,0	0,973

Pri usporedbi sustava, moguće je pratiti i dosljednost izmjerenih vrijednosti po svakom O-prstenu u automatiziranom i ručnom mjernom sustavu. Kao procjena vrijednosti veličine unutarnjeg promjera svakog O-prstena, mjerenog optičkim sustavom, se može uzeti prosječna vrijednost svih mjerenja navedenog O-prstena (prosjeci po stupcima prema tablici 4.7). Ukoliko se isto učini i za mjerenja mikrometarskim vijkom (prosjeci po stupcima prema tablici 4.2), može se odrediti korelacija između dobivenih procjena veličina pomoću paketa *irr*:

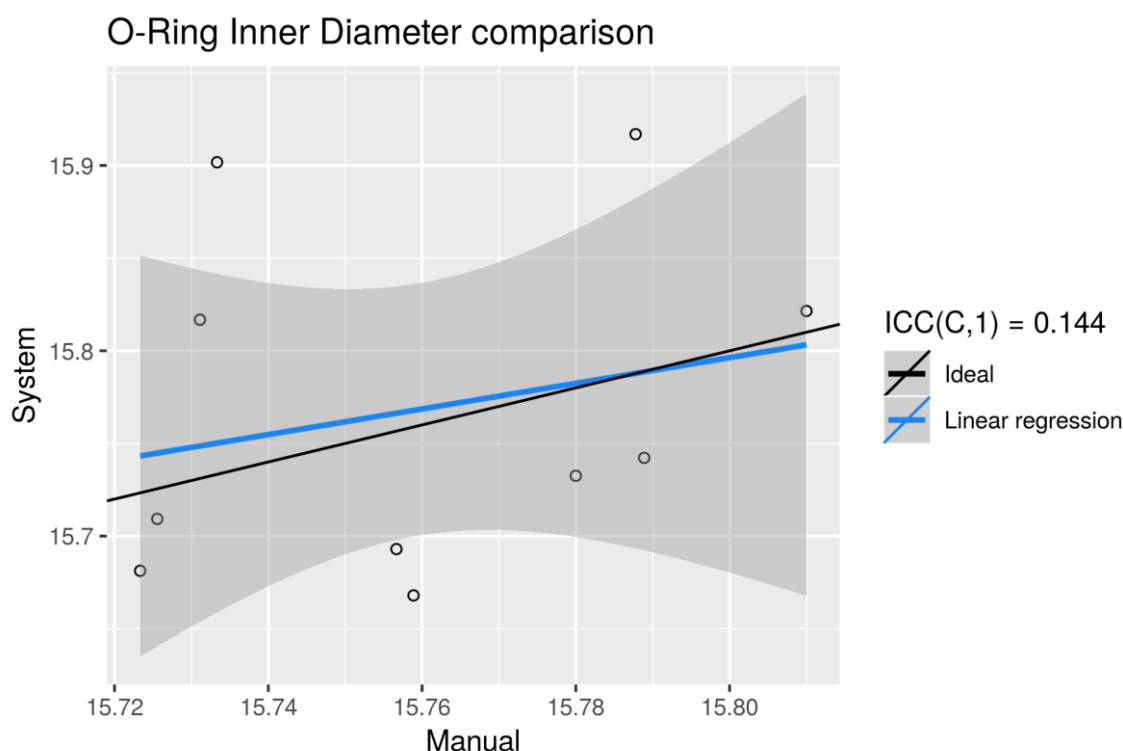
```

552 Single Score Intraclass Correlation
553
554 Model: twoway
555 Type : consistency
556
557 Subjects = 10
558 Raters = 2
559 ICC(C,1) = 0.144
560
561 F-Test, H0: r0 = 0 ; H1: r0 > 0
562 F(9,9) = 1.34 , p = 0.336
563
564 95%-Confidence Interval for ICC Population Values:
565 -0.501 < ICC < 0.687

```

ICC vrijednost 0,144 ukazuje na slabu pouzdanost između procjena unutarnjeg promjera O-prstena automatiziranim, odnosno ručnim putem. Pomoću istih vrijednosti, prikazanih u tablici 4.12, može

se konstruirati dijagram na slici 4.18, u kojem su procjene veličina mikrometarskim vijkom upisane na x -osi, a procjene veličina pomoću optičkog sustava na y -osi. Također, crnom bojom je upisana linija idealnog poklapanja mjerenja. Ukoliko bi se pojedine procjene nalazile na toj liniji, to bi ukazivalo na potpuno poklapanje procjena unutarnjeg promjera O-prstena u oba mjerna sustava. Osim procijenjenih vrijednosti, na dijagramu se nalazi i regresijski pravac (plava boja) s 95%-tnim intervalom povjerenja. Sa slike je vidljivo da je dosljednost između procijenjenih vrijednosti niska, na što ukazuje i niski ICC. Vrijednosti su u velikoj mjeri rasipane oko idealnog pravca, iako se regresijska krivulja relativno dobro poklapa s istim.



Slika 4.18. Usporedba procjene vrijednosti unutarnjeg promjera [mm] O-prstena u slučaju optičkog mjerenja (System) i mjerenja mikrometarskim vijkom (Manual)

U slučaju mjerenja veličine vanjskog promjera, može se uzeti prosjek svih stupaca tablice 4.3 kao procjena promjera pri mjerenju mikrometarskim vijkom. Prosječne vrijednosti stupaca tablice 4.8 definiraju procjenu vanjskog promjera pri mjerenju optičkim sustavom. Navedene procjene prikazane su u tablici 4.12. Kako bi se ocijenila pouzdanost između optičkog i ručnog mjernog sustava, izračunat je ICC pomoću paketa *irr*:

```

578 Single Score Intraclass Correlation
579
580 Model: twoway
581 Type : consistency
582
583 Subjects = 10

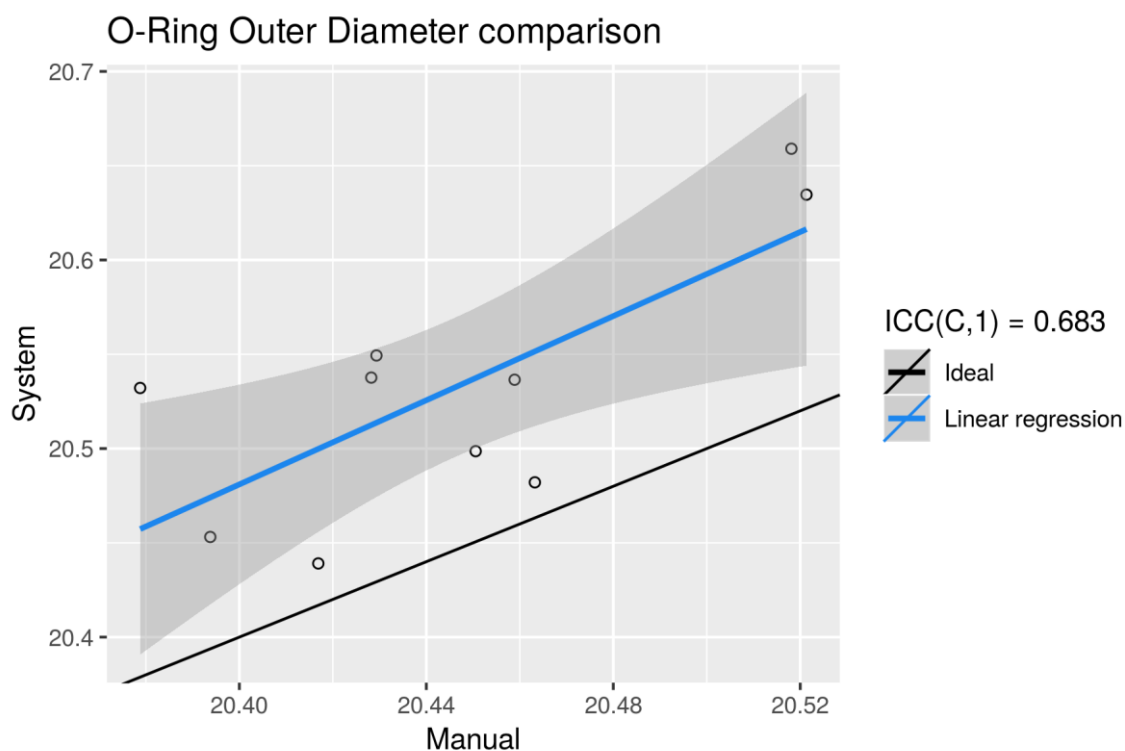
```

```

584   Raters = 2
585   ICC(C,1) = 0.683
586
587   F-Test, H0: r0 = 0 ; H1: r0 > 0
588       F(9,9) = 5.3 , p = 0.0103
589
590   95%-Confidence Interval for ICC Population Values:
591   0.137 < ICC < 0.91

```

ICC vrijednost od 0,683, prema kriterijima iz potpoglavlja 4.2, ukazuje na umjerenu korelaciju između navedenih mjernih sustava. Na dijagramu na slici 4.19 je prikazana usporedba procijenjenih vrijednosti vanjskog promjera O-prstena iz tablice 4.12. Na x -osi su vrijednosti dobivene pomoću mikrometarskog vijka, a na y -osi vrijednosti dobivene pomoću optičkog sustava. Pravac crne boje označava pravac idealnog poklapanja mjerenja dobivenih pomoću oba mjerna sustava. Pravac plave boje je regresijski pravac ucrtanih vrijednosti. Uz njega je ucrtan i 95%-tni interval povjerenja. Uzimajući u obzir uži interval povjerenja i višu ICC vrijednost, može se zaključiti da su izmjerene vrijednosti vanjskog promjera znatno dosljednije nego vrijednosti unutarnjeg promjera. Također, vidljiv je i pomak regresijskog pravca u odnosu na idealan pravac, što ukazuje na sustavnu grešku prilikom mjerenja. Optičkim mjernim sustavom su konstantno mjerene više vrijednosti vanjskog promjera O-prstena, nego pomoću mikrometarskog vijka.



Slika 4.19. Usporedba procjene vrijednosti vanjskog promjera [mm] O-prstena u slučaju optičkog mjerenja (System) i mjerenja mikrometarskim vijkom (Manual)

Tablica 4.12. Procjene veličina vanjskog i unutarnjeg promjera dobivene pomoću prosjeka svih mjerenja optičkim, odnosno ručnim sustavom

Mjerni sustav: O-prsten	Optički		Ručni (mikrometarski vijak)	
	Procjena veličine vanjskog promjera [mm]	Procjena veličine unutarnjeg promjera [mm]	Procjena veličine vanjskog promjera [mm]	Procjena veličine unutarnjeg promjera [mm]
15,8 x 24_1	20,659	15,917	20,518	15,788
15,8 x 24_2	20,439	15,902	20,417	15,733
15,8 x 24_3	20,549	15,693	20,429	15,757
15,8 x 24_4	20,499	15,742	20,451	15,789
15,8 x 24_5	20,536	15,709	20,459	15,726
15,8 x 24_6	20,538	15,668	20,428	15,759
15,8 x 24_7	20,635	15,821	20,521	15,810
15,8 x 24_8	20,532	15,817	20,379	15,731
15,8 x 24_9	20,453	15,681	20,394	15,723
15,8 x 24_10	20,482	15,733	20,463	15,780

Također, prosjek svih mjerenja vanjskog promjera O-prstena optičkim sustavom, prema tablici 4.8, nalazi se znatno bliže nazivnoj vrijednosti, nego prosjek svih mjerenja mikrometarskim vijkom prema tablici 4.3. Navedena činjenica može ukazivati na deformaciju O-prstena tijekom ručnih mjerenja.

Na kraju se mogu i analizirati sustavne greške mjerenja između automatiziranog i ručnog sustava u tablici 4.13. Tablica je dobivena oduzimanjem procijenjenih veličina dobivenih mikrometarskim vijkom od onih optičkim sustavom. Pomoću tablice se jasno vidi da su sustavne greške prilikom mjerenja vanjskog promjera sve pozitivne, odnosno optički sustav je pozitivno pristran pri takvim mjerenjima, ili je ručni sustav negativno pristran. Za razliku od toga, prilikom mjerenja veličine unutarnjeg promjera se ne može zaključiti o pristranosti sustava. Prosječne greške prilikom unutrašnjih mjerenja su znatno niže, a prilikom vanjskih mjerenja više, što je vidljivo iz poklapanja regresijskih i idealnih pravaca sa slika 4.18 i 4.19. No, standardna devijacija greške unutrašnjih mjerenja je znatno viša od standardne devijacije greške vanjskih mjerenja, na što ukazuju rasipanja

vrijednosti na navedenim slikama. Rasponi grešaka unutrašnjih i vanjskih mjerenja se praktički poklapaju.

Tablica 4.13. Sustavne pogreške u procijenjenim vrijednostima veličina izmjerenih automatiziranim, odnosno ručnim mjernim sustavom

O-prsten	Sustavna pogreška unutarnjeg promjera [mm]	Sustavna pogreška vanjskog promjera [mm]
15,8 x 24_1	0,129	0,141
15,8 x 24_2	0,168	0,022
15,8 x 24_3	-0,064	0,120
15,8 x 24_4	-0,047	0,048
15,8 x 24_5	-0,016	0,078
15,8 x 24_6	-0,091	0,109
15,8 x 24_7	0,011	0,113
15,8 x 24_8	0,086	0,153
15,8 x 24_9	-0,042	0,059
15,8 x 24_10	-0,047	0,019
Prosjek:	0,0088	0,0863
St. dev.:	0,089	0,048
Raspon:	0,168	0,153

5. ZAKLJUČAK

Cilj rada je bila razrada i primjena optičkog sustava kontrole za inspekciju dimenzija odabranog proizvoda. Predloženi sustav ne zadovoljava stroge uvjete kvalitete koje nalaže *Six Sigma* metodologija, stoga bi za njegovu primjenu u praksi bilo potrebno uvesti određene dorade. Također, propusna moć sustava je mala u usporedbi s industrijskim rješenjima, no kako je predloženi sustav prototipno rješenje, taj pokazatelj nije od presudne važnosti.

Prva dorada sustava može ići u smjeru kalibracije kamere i određivanja matrice kamere. Ispravljanjem radijalne i tangencijalne distorzije se mogu poboljšati rezultati mjerenja. Za kalibraciju kamere postoje rješenja otvorenog koda, poput *OpenCV*-a, koja sadrže potrebne funkcije i upute [31].

Nakon toga se može provjeriti zadovoljava li raspoloživa oprema uvjete optičkog sustava. Primjer ulaganja u adekvatniju opremu bi bila nabava telecentričnog objektiva i zamjena postojećeg. Primjenom takve leće, uvećanje predmeta ne ovisi o udaljenosti od objektiva, stoga se izbjegava predloženi način kalibracije sustava. Kalibracija telecentričnog sustava je znatno jednostavnija, što rezultira manjom unesenom greškom.

Treća dorada sustava može uključiti naprednije algoritme za procjenu rubova predmeta. Jedan takav algoritam je predložen u [15]. U izrađenom sustavu su se rubovi predmeta procijenili na razini jednog piksela. Naprednijim načinom, prema [15], rubovi predmeta su procijenjeni unutar pojedinog piksela, na tzv. *subpixel* razini, pomoću Gaussovog integralnog modela.

Postoji još mnogo utjecajnih elemenata na kvalitetu dobivenih rezultata, poput brzine pokretne trake, kvalitete osvjetljenja, preciznosti etalona za umjeravanje, okomitosti optičke osi i promatranih predmeta, itd.

Kao nadogradnju postojećeg rješenja moguće je uklopiti metode za interpretaciju dobivenih rezultata mjerenja. Na taj način se mogu identificirati nesukladni proizvodi i djelovati na sam proces proizvodnje.

Vrijednost ovog rada je u izbjegavanju korištenja komercijalnih programskih rješenja, te u sistematizaciji postojećih besplatnih modula otvorenog koda. Takvi moduli se mogu uspješno koristiti za izradu prototipa i provjeru izvedivosti sustava strojnog vida. Problemi koji se javljaju pri implementaciji sustava strojnog vida u praksi su iznimno složeni i često ih je moguće riješiti samo principom pokušaja i pogrešaka.

LITERATURA

- [1] Jiang, B. C.; Jiang, S.: »Machine Vision Based Inspection of Oil Seals«, *Journal of Manufacturing Systems*, svez. 17, br. 3, pp. 159-166, 1998.
- [2] Scholz, O. i dr.: »Fully-Automatic Surface Inspection of O-Ring Seals«, *SAE Technical Paper 2015-01-0596*, 2015.
- [3] Productivity Quality: »O-ring Inspection - Productivity Quality«, s Interneta, <https://www.planarinspect.com/o-ring-inspection/>, 11.03.2019.
- [4] Seal & Design Inc.: »O-Ring Inspection Equipment | sealanddesign.com«, s Interneta, <http://blog.sealanddesign.com/o-rings/o-ring-inspection-equipment>, 11.03.2019.
- [5] Cincinnati Automation Ltd.: »O - Ring Inspection Systems - Cincinnati Automation«, s Interneta, <https://www.cincinnatiautomation.com/page/o-ring-inspection-systems>, 11.03.2019.
- [6] Beyerer, J.; León, F. P.; Frese, C.: »Machine Vision: Automated Visual Inspection: Theory, Practice and Applications«, Springer-Verlag, Berlin, 2015.
- [7] Marco Rubber & Plastics: »O-Ring Tolerances International General Purpose«, s Interneta, <https://www.marcorubber.com>, 11.03.2019..
- [8] Vossepoel, A. M.; Smeulders, A. W. M.: »Vector code probability and metrication error in the representation of straight lines of finite length«, *Computer Graphics and Image Processing*, svez. 20, pp. 347-368, 1982.
- [9] Luengo, C.: »DIPlib 3.0.alpha, A Library for Quantitative Image Analysis«, s Interneta, <https://diplib.github.io/diplib-docs/>, 11.03.2019.
- [10] Sympatec GmbH: »Particle Characterisation«, s Interneta, <https://www.sympatec.com/en/particle-measurement/glossary/particle-shape/>, 11.03.2019.
- [11] Podczek, F.; Newton, J. M.: »A Shape Factor to Characterize The Quality of Spheroids«, *Journal of Pharmacy and Pharmacology*, svez. 46, br. 2, pp. 82-85, 1994.

- [12] Basler AG: »Basler ace USB 3.0 - Product Documentation (User's Manual)«, s Interneta, <https://www.baslerweb.com/en/sales-support/downloads/document-downloads/basler-ace-usb-3-0-users-manual/>, 11.03.2019.
- [13] Basler AG: »pylon Open Source Projects«, s Interneta, <https://www.baslerweb.com/pylon>, 11. 03. 2019.
- [14] Edmund Optics: »16mm Focal Length, HP Series Fixed Focal Length Lens«, s Interneta, <https://www.edmundoptics.com/p/16mm-Focal-Length-HP-Series-Fixed-Focal-Length-Lens/28990/>, 11.03.2019.
- [15] Duan, Z. i dr.: »High Precision Edge Detection Algorithm for Mechanical Parts«, *Measurement Science Review*, svez. 18, br. 2, pp. 65-71, 2018.
- [16] Kuwahara, M.; Hachimura, K.; Kinoshita, M.: »Image enhancement and left ventricular contour extraction techniques applied to radioisotope angiocardiograms«, *Automedica*, svez. 3, br. 2, pp. 107-119, 1980.
- [17] van der Walt, S.; Colbert, S. C.; Varoquaux, G.: »The NumPy Array: A Structure for Efficient Numerical Computation«, *Computing in Science & Engineering*, svez. 13, br. 2, pp. 22-30, 2011.
- [18] Oliphant, T. E.: »Guide to NumPy«, Trelgol Publishing, SAD, 2006.
- [19] Jones, E.; Oliphant, T.; Peterson, P.: »SciPy: Open Source Scientific Tools for Python«, s Interneta, <http://www.scipy.org/>, 11.03.2019.
- [20] Campagnola, L.: »PyQtGraph - Scientific Graphics and GUI Library for Python«, s Interneta, <http://www.pyqtgraph.org/>, 11.03.2019.
- [21] McKerns, M. i dr.: »Building a framework for predictive science«, *Proceedings of the 10th Python in Science Conference*, Austin, 2011.
- [22] McKerns, M.; Aivazis, M.: »pathos: a framework for heterogeneous computing«, s Interneta, <http://trac.mystic.cacr.caltech.edu/project/pathos>, 11.03.2010.

- [23] United Kingdom Accreditation Service: »The Expression of Uncertainty and Confidence in Measurement, 3rd edition«, Feltham, Middlesex, 2012.
- [24] R Core Team: »R: A Language and Environment for Statistical Computing«, R Foundation for Statistical Computing, s Interneta, <http://www.R-project.org/>, 11.03.2013.
- [25] Cano, E. L.; Moguerza, J. M.; Redchuk, A.: »Six Sigma with R«, Springer, New York, 2012.
- [26] Gamer, M. i dr.: »Package 'irr', Various Coefficients of Interrater Reliability and Agreement«, s Interneta, <https://cran.r-project.org/web/packages/irr/>, 11.03.2019.
- [27] Zaimović-Uzunović, N. i dr.: »Proizvodna Mjerenja, prvo izdanje«, Univerzitet u Zenici, Mašinski fakultet, Zenica, 2009.
- [28] Koo, T. K.; Li, M. Y.: »A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research«, *Journal of Chiropractic Medicine*, svez. 15, br. 2, p. 155–163, 2016.
- [29] Harrell, F. E.: »Hmisc: Harrell Miscellaneous, R package version 4.2-0«, s Interneta, <https://cran.r-project.org/package=Hmisc>, 11.03.2019.
- [30] Wickham, H.: »ggplot2: Elegant Graphics for Data Analysis«, Springer-Verlag, New York, 2016.
- [31] OpenCV team: »Camera Calibration - OpenCV documentation«, s Interneta, https://docs.opencv.org/3.4.3/dc/dbb/tutorial_py_calibration.html, 11.03.2019.

POPIS OZNAKA I KRATICA

U [μm]	Mjerna nesigurnost
k [/]	Faktor pokrivanja
s [mm]	Procjena standardne devijacije populacije na temelju uzorka
n [/]	Veličina uzorka
x_i [mm]	Izmjerene vrijednost i -tog uzorka
\bar{x} [mm]	Prosjek izmjerenih vrijednosti uzorka, aritmetička sredina vrijednosti procesa
$\%R\&R$ [%]	Udio varijabilnosti mjernog sustava u ukupnoj varijabilnosti
$R\&R/T$ [%]	Udio varijabilnosti mjernog sustava u toleranciji procesa
$\sigma_{R\&R}$ [mm]	Varijabilnost mjernog sustava (standardna devijacija sustava pomnožena s 5,15 da obuhvaća 99% podataka)
σ_{uk} [mm]	Ukupna varijabilnost (ukupna standardna devijacija pomnožena s 5,15 da obuhvaća 99% podataka)
USL [mm]	Gornja granica procesa specificirana od strane klijenta (<i>Upper Specification Limit</i>)
LSL [mm]	Donja granica procesa specificirana od strane klijenta (<i>Lower Specification Limit</i>)
σ_{pr} [mm]	Standardna devijacija vrijednosti procesa
$ICC(C, 1)$ [/]	Intraklasni koeficijent korelacije (<i>Intraclass correlation coefficient</i>)
MS_R [/]	Kvadratna sredina svih stupaca
MS_E [/]	Kvadratna sredina greške
k [/]	Broj procjenitelja (mjeritelja ili mjernih sustava)
$\mathcal{A}(\cdot)$ [/]	Prijenosna funkcija idealne leće
\mathbf{B} [/]	Točka senzora optičkog sustava
\mathbf{G} [/]	Točka promatranog predmeta optičkog sustava
G [mm]	Udaljenost točke promatranog predmeta od optičke osi leće
B [mm]	Udaljenost točke senzora od optičke osi leće
D [mm]	Promjer leće
g [mm]	Udaljenost promatranog predmeta leće
b [mm]	Udaljenost senzora od leće

f [mm]	Žarišna udaljenost leće
V [/]	Odnos veličine predmeta na slici i stvarne veličine promatranog predmeta
g_1 [mm]	Udaljenost od leće do etalona 1
g_2 [mm]	Udaljenost od leće do etalona 2
g_x [mm]	Udaljenost od leće do mjenog predmeta (O-prstena)
t_1 [mm]	Visina etalona 1
t_2 [mm]	Visina etalona 2
t_x [mm]	Visina mjenog predmeta (O-prstena)
G_1 [mm]	Mjerena veličina etalona 1 unaprijed poznate vrijednosti
G_2 [mm]	Mjerena veličina etalona 2 unaprijed poznate vrijednosti
B_1 [px]	Mjerena veličina etalona 1 preslikana na senzor (sliku)
B_2 [px]	Mjerena veličina etalona 2 preslikana na senzor (sliku)
B_x [px]	Mjerena veličina predmeta preslikana na senzor (sliku)
$g(\mathbf{x})$ [/]	Funkcija za dodjeljivanje vrijednosti na poziciji \mathbf{x} na slici
\mathbf{x} [/]	Dvodimenzionalni vektor koordinata slike
x [/]	x -koordinata slike
y [/]	y -koordinata slike
Q [/]	Prostorna dimenzija vrijednosti slike (<i>image channel</i>)
\mathbf{g}_{mn} [/]	Diskretne vrijednosti intenziteta svakog piksela slike
m [/]	Prva prostorna koordinata piksela slike, $m \in [0, 1, 2, \dots, M - 1]$
n [/]	Druga prostorna koordinata piksela slike, $n \in [0, 1, 2, \dots, N - 1]$
$M \times N$ [/]	Rezolucija slike u pikselima
K [/]	Broj razina koje može poprimiti vrijednost intenziteta piksela slike
T [mm]	Ukupna širina tolerancijskog polja mjerene veličine
θ [rad]	Kut upada svjetlosti
PLC	Programabilni logički kontroler (<i>Programmable logic controller</i>)
FPS	Broj prikazanih slika u sekundi (<i>Frames per second</i>)
R&R	Ponovljivost i obnovljivost (<i>Repeatability and reproducibility</i>)
ROI	Područje interesa slike (<i>Region of interest</i>)

mm/px omjer [mm/px]	Veličina jednog piksela slike u mjernim jedinicama milimetra
<i>Size</i> [mm ²]	Površina poligona kao reprezentacije mjerenog predmeta
<i>Feret PerpMin</i> [mm]	Udaljenost između dva paralelna pravca, mjerena pod kutom od 90° u odnosu na kut minimalnog Feret-ovog promjera
<i>Feret Min</i> [mm]	Najmanja udaljenost između dva paralelna pravca koja dodiruju mjereni predmet
<i>Convexity</i> [/]	Konveksnost mjerenog predmeta
<i>ConvexPerimeter</i> [mm]	Ukupna duljina lančanog koda po obodu mjerenog predmeta bez udubljenja
<i>Perimeter</i> [mm]	Ukupna duljina lančanog koda po obodu mjerenog predmeta, uzimajući u obzir udubljenja i izbočine
<i>Eccentricity</i> [mm]	Ekscentričnost unutarnjeg i vanjskog promjera O-prstena
<i>Center dim0_{out}</i> [mm]	Koordinate centra vanjskog promjera O-prstena po <i>x</i> -osi
<i>Center dim0_{in}</i> [mm]	Koordinate centra unutarnjeg promjera O-prstena po <i>x</i> -osi
<i>Center dim1_{out}</i> [mm]	Koordinate centra vanjskog promjera O-prstena po <i>y</i> -osi
<i>Center dim1_{in}</i> [mm]	Koordinate centra unutarnjeg promjera O-prstena po <i>y</i> -osi
<i>Feret Max</i> [mm]	Maksimalni Feret-ov promjer
<i>Feret MaxAng</i> [rad]	Kut maksimalnog Feret-ovog promjera
<i>Feret MinAng</i> [rad]	Kut minimalnog Feret-ovog promjera
<i>Radius Max</i> [mm]	Maksimalni polumjer
<i>Radius Min</i> [mm]	Minimalni polumjer
<i>Radius Mean</i> [mm]	Prosječni polumjer
<i>Radius StdDev</i> [mm]	Standardna devijacija polumjera
<i>PodczechShapes Circle</i>	Podczech-ov oblik kruga
<i>PodczechShapes Ellipse</i>	Podczech-ov oblik elipse

SAŽETAK

U prvom dijelu rada je predložen automatizirani optički sustav kontrole dimenzija O-prstena. Opisane su mjerene veličine O-prstena, te komponente, kalibracija i postavljanje optičkog sustava. Predstavljena je i programska podrška mjernog sustava, izrađena pomoću programskog jezika *Python* i programskih paketa otvorenog koda. Prikazan je način korištenja grafičkog sučelja programskog rješenja i pojašnjeni su algoritmi kalibracije, mjerenja i predobrade digitalnih slika.

U drugom dijelu rada je prikazan postupak mjerenja O-prstena mikrometarskim vijkom, te pomoću optičkog mjernog sustava. Sposobnost oba sustava je analizirana primjenom *Six Sigma* metodologije. Ispitana je osjetljivost optičkog mjernog sustava na kvalitetu kalibracije. Automatizirani i ručni mjerni sustav su uspoređeni pomoću statističkih pokazatelja ponovljivosti i obnovljivosti, te intraklasnog koeficijenta korelacije. Na kraju su predložene mjere povećanja kvalitete optičkog mjernog sustava.

Ključne riječi: optički mjerni sustav, mjerenje O-prstena, *Six Sigma*, programska podrška otvorenog koda

SUMMARY

In the first section of the thesis, an automated visual inspection system for O-ring measurement is proposed. Measured O-ring features are also defined. Optical system components, calibration and setup are described. Measurement system software solution, developed using the Python programming language and the open-source software, is also introduced. Instructions for using software solution graphical interface are composed, and the algorithms for calibration, measurement and digital image preprocessing are defined.

The second part of thesis presents methods for measuring the O-ring using micrometer and the optical measurement system. The capability of each system is analyzed using the Six Sigma methodology. Optical measurement system sensitivity to the calibration is examined. Automated and manual measurement systems were compared by means of repeatability and reproducibility and intraclass correlation coefficient. Finally, the measures to increase the quality of the optical measurement system are proposed.

Keywords: optical measurement system, O-ring measurement, Six Sigma, open source software solution

PRILOZI

Prilog 1.: Programski kod podrške mjernom sustavu

1.1. Skripta „main.py“

```
1 import sys
2
3 # Numpy
4 import numpy as np
5
6
7 # Pylon
8 from pypylon import pylon
9 from pypylon import genicam
10 import cv2
11 import PyDIP as dip
12 import scipy.misc
13
14 # GUI imports
15 import pyforms
16 # from gui.pyforms import MeasuringApp
17 from gui.mainwindow import MeasuringApp
18 from pyqtgraph import QtGui
19
20
21 def save_cam_params(cam, fn):
22     """
23     Save cam parameters to file fn
24
25     :param cam:
26     :param fn:
27     :return:
28     """
29     try:
30         pylon.FeaturePersistence.Save(fn, cam.GetNodeMap())
31         print("Features saved to {}".format(fn))
32     except:
33         print("\nFeatures saving failed!")
34
35 def grab_one_pic(cam, cs=pylon.PixelType_RGB8packed):
36     """
37     Grab one image, convert it to required color space
38
39     :param cam:
40     :return:
41     """
42     grab = cam.GrabOne(100)
43     converter = pylon.ImageFormatConverter()
44     converter.OutputPixelFormat = cs
45     converter.OutputBitAlignment = pylon.OutputBitAlignment_MsbAligned
46     img = converter.Convert(grab)
47     return img.GetArray()
48
49
50 if __name__ == "__main__":
51     # save_cam_params(cam, "Features.pfs")
52     # pyforms.start_app(MeasuringApp)
53     app = QtGui.QApplication(sys.argv)
```



```

54     GUI = MeasuringApp()
55     sys.exit(app.exec_())
56     """
57     # Korekcijski faktor debljine etalona
58     k_et = (104.5 - 2.366 + 2.13) / (104.5 - 9. +2.13)
59     print("Korekcijski faktor debljine etalona: {}".format(k_et))
60     pixel_size_corr = k_et * 0.035650623885918005
61     print("Korigirani Omjer Pixel/mm: {}".format(pixel_size_corr))
62     # grab_show_cv(cam)
63     img = dip.Image(grab_one_pic(cam, pylon.PixelType_Mono8))
64     img.Show()
65     # Thresholding image
66     thr = dip.Threshold(img, 'background', parameter=8)
67     print("Threshold: {}".format(thr[1]))
68     thr[0].Show()
69     # Gaussian Smoothing
70     smooth = dip.Gauss(img, sigmas=[0])
71     smooth.Show()
72     thr_sm = dip.Threshold(smooth, 'background', parameter=10)
73     print("Gaussian smoothed threshold: {}".format(thr_sm[1]))
74     thr_sm[0].Show()
75     # Opening with kernel
76     mask = dip.Opening(thr_sm[0], dip.SE(21, 'elliptic'))
77     mask.Show()
78     # Segmenting inner and outer parts
79     seg = dip.Convert(dip.EdgeObjectsRemove(mask), 'UINT8')
80     seg[seg > 0] = 100
81     seg[mask == 0] = 250
82     # Assigning physical dimension to segmented px
83     seg.SetPixelSize(dip.PixelSize(dip.PhysicalQuantity(
84         pixel_size_corr, "mm")))
85     # seg.SetPixelSize(dip.PixelSize(dip.PhysicalQuantity(1, "px")))
86     seg.Show()
87     # Assigning physical dimension to img px
88     img.SetPixelSize(dip.PixelSize(dip.PhysicalQuantity(
89         pixel_size_corr, "mm")))
90     # img.SetPixelSize(dip.PixelSize(dip.PhysicalQuantity(1, "px")))
91     # Labeling inner and outer objects
92     inner_lbl = dip.Label(seg == 100, minSize=100)
93     outer_lbl = dip.Label(seg > 99, minSize=100)
94     inner_lbl.Show()
95     outer_lbl.Show()
96     # Measuring
97     m_outer = dip.MeasurementTool.Measure(
98         outer_lbl, img, features=['Feret', 'Radius', 'Center'])
99     m_inner = dip.MeasurementTool.Measure(
100         inner_lbl, img, features=['Feret', 'Radius', 'Center'])
101     print("\n\n\t\t~~~~ Outer Diameter Measures ~~~~")
102     print(m_outer)
103     print("\n\n\t\t~~~~ Inner Diameter Measures ~~~~")
104     print(m_inner)
105     # pixel_size = 20. / m_outer['Feret'][1][1]
106     # print('Pixel/mm Omjer etalona: {}'.format(pixel_size))
107     # dip.Show(dip.ObjectToMeasurement(outer_lbl, m_outer['Feret']))
108     """
109
110     # finally:
111     #     pass
112     # cam.close()

```

1.2. Skripta „cam.py“

```
1 # Pylon
2 from pypylon import pylon
3 from pypylon import genicam
4
5 from pyqtgraph import QtCore
6 import pyqtgraph as pg
7
8 class CamObject():
9     def __init__(self):
10        # FPS counter
11        self.fps = 0.
12        self.fps_count = 0
13        self.fps_start = 0.
14        # Frame object
15        self.frame = None
16
17    def __enter__(self):
18        return self
19
20    def __exit__(self, exc_type, exc_val, exc_tb):
21        self.close()
22
23    def create(self):
24        try:
25            self.cam = pylon.InstantCamera(
26                pylon.TlFactory.GetInstance().CreateFirstDevice())
27            print("\nDevice: {}".format(
28                self.cam.GetDeviceInfo().GetModelName()))
29            # Software trigger configuration registering
30            self.cam.RegisterConfiguration(
31                pylon.SoftwareTriggerConfiguration(),
32                # pylon.AcquireContinuousConfiguration(),
33                pylon.RegistrationMode_ReplaceAll,
34                pylon.Cleanup_Delete
35            )
36            # Enabling camera events
37            self.cam.GrabCameraEvents = True
38        except Exception as e:
39            print("\nCamera NOT FOUND!")
40            print(e)
41            raise
42
43    def open(self):
44        """
45        Create camera object instance
46        """
47        # Camera instance
48        try:
49            self.cam.Open()
50            print("\nCamera Opened")
51        except Exception as e:
52            print('\nCamera Open FAILED!')
53            print(e)
54            raise
55
56        # Check if camera supports events.
57        try:
58            if not genicam.IsAvailable(self.cam.EventSelector):
59                raise genicam.RuntimeException(
```

```

60         "The device doesn't support events.")
61     except genicam.GenericException as e:
62         # Error handling.
63         print("An exception occurred.", e.GetDescription())
64
65     # Maximum number of buffers for simultaneous image grabbing
66     try:
67         self.cam.MaxNumBuffer = 10
68         print("\nMaximum Buffer Number: {}".format(
69             self.cam.MaxNumBuffer.GetValue()))
70     except Exception as e:
71         print("\ncam.MaxNumBuffer FAILED!")
72         print(e)
73         raise
74
75     # Frame rate upper limit for for simultaneous image grabbing
76     try:
77         self.cam.AcquisitionFrameRateEnable.SetValue(True)
78         self.cam.AcquisitionFrameRate = 50
79         print("\nFrame rate upper limit: {} FPS".format(
80             self.cam.AcquisitionFrameRate.GetValue()))
81     except Exception as e:
82         print("\ncam.AcquisitionFrameRate FAILED!")
83         print(e)
84         raise
85
86     # Pixel format
87     try:
88         self.cam.PixelFormat = "Mono8"
89         print("\nImage format:
{}".format(self.cam.PixelFormat.GetValue()))
90     except Exception as e:
91         print("\ncam.PixelFormat FAILED!")
92         print(e)
93         raise
94
95     def close(self):
96         try:
97             self.cam.Close()
98             print("\nCamera Closed!")
99     except Exception as e:
100         print("\nClosing FAILED!")
101         print(e)
102         raise
103
104     def print_info(self):
105         """
106         Print Camera info
107         """
108         try:
109             print("\n\t\t\t\t\t~~~~~ Camera Info ~~~~~")
110             print("\nDevice Vendor Name: {}".format(
111                 self.cam.DeviceVendorName.GetValue()))
112             print("\nCamera Sensor")
113             print("\tWidth: {}
px".format(self.cam.SensorWidth.GetValue()))
114             print("\tHeight: {} px".format(
115                 self.cam.SensorHeight.GetValue()))
116             print("\nROI")
117             print("\tMax Width: {}
px".format(self.cam.WidthMax.GetValue()))

```

```

118         print("\tMax Height: {}
px".format(self.cam.HeightMax.GetValue()))
119         print("\nDevice Link Speed: {} MB/sec".format(
120             self.cam.DeviceLinkSpeed.GetValue()*1e-6))
121         print("\nGain: {} dB".format(self.cam.Gain.GetValue()))
122         print("\tMax: {} dB".format(self.cam.Gain.Max))
123         print("\tMin: {} dB".format(self.cam.Gain.Min))
124         print("\nExposure time: {}
us".format(self.cam.ExposureTime()))
125         print("\tMax: {}".format(self.cam.ExposureTime.Max))
126         print("\tMin: {}".format(self.cam.ExposureTime.Min))
127     except Exception as e:
128         print('\nCamera Print Info FAILED!')
129         print(e)
130         raise
131
132
133     def set_default_params(self):
134         """
135         MaxNumBuffer: 8
136         """
137         print("\n\t\t\t\t~ Setting Default Parameters ~")
138         # Amplification of camera signal
139         try:
140             self.cam.GainAuto.SetValue("Off")
141             self.cam.Gain = 2.28869
142             print("\nGain: {} dB".format(self.cam.Gain()))
143         except Exception as e:
144             print("\ncam.Gain FAILED!")
145             print(e)
146             raise
147
148         # Exposure time
149         try:
150             self.cam.ExposureAuto.SetValue("Off")
151             self.cam.ExposureTime = 60000 # us
152             print("\nExposure Time: {} us".format(
153                 self.cam.ExposureTime()))
154         except Exception as e:
155             print("\ncam.ExposureTime FAILED!")
156             print(e)
157             raise
158
159         # ROI Size and position
160         try:
161             if genicam.IsWritable(self.cam.Width):
162                 self.cam.Width = self.cam.Width.Max
163                 # self.cam.Width.SetValue(1248)
164             if genicam.IsWritable(self.cam.Height):
165                 self.cam.Height = self.cam.Height.Max
166                 # self.cam.Height.SetValue(750)
167             if genicam.IsWritable(self.cam.OffsetX):
168                 self.cam.OffsetX = self.cam.OffsetX.Min
169                 # self.cam.OffsetX.SetValue(544)
170             if genicam.IsWritable(self.cam.OffsetY):
171                 self.cam.OffsetY = self.cam.OffsetY.Min
172                 # self.cam.OffsetY.SetValue(800)
173             print("\nImage ROI")
174             print("\tOffsetX: {} px".format(self.cam.OffsetX.GetValue()))
175             print("\tOffsetY: {} px".format(self.cam.OffsetY.GetValue()))
176             print("\tWidth: {} px".format(self.cam.Width.GetValue()))
177             print("\tHeight: {} px".format(self.cam.Height.GetValue()))

```

```
178         except Exception as e:
179             print("\nSetting ROI FAILED!")
180             print(e)
181             raise
182
183     def set_gain(self, gain):
184         """
185         Set amplification of camera signal
186         """
187         try:
188             self.cam.Gain = gain
189             # print("\nGain: {} dB".format(self.cam.Gain()))
190         except Exception as e:
191             print("\ncam.Gain FAILED!")
192             print(e)
193             raise
194
195     def set_exposure_time(self, exp_time):
196         """
197         Set camera exposure time
198         """
199         try:
200             self.cam.ExposureTime = exp_time
201             # print("\nExposure Time: {}
us".format(self.cam.ExposureTime()))
202         except Exception as e:
203             print("\ncam.ExposureTime FAILED!")
204             print(e)
205             raise
206
207     def set_width(self, width):
208         """
209         Set Camera ROI Width
210         """
211         try:
212             if genicam.IsWritable(self.cam.Width):
213                 self.cam.Width = width
214         except Exception as e:
215             print("\ncam.Width FAILED!")
216             print(e)
217             raise
218
219     def set_height(self, height):
220         """
221         Set Camera ROI Height
222         """
223         try:
224             if genicam.IsWritable(self.cam.Height):
225                 self.cam.Height = height
226         except Exception as e:
227             print("\ncam.Height FAILED!")
228             print(e)
229             raise
230
231     def set_offsetx(self, offsetx):
232         """
233         Set Camera ROI X Axis Offset
234         """
235         try:
236             if genicam.IsWritable(self.cam.OffsetX):
237                 self.cam.OffsetX = offsetx
238         except Exception as e:
```

```
239         print("\ncam.OffsetX FAILED!")
240         print(e)
241         raise
242
243     def set_offsety(self, offsety):
244         """
245         Set Camera ROI Y Axis Offset
246         """
247         try:
248             if genicam.IsWritable(self.cam.OffsetY):
249                 self.cam.OffsetY = offsety
250         except Exception as e:
251             print("\ncam.OffsetY FAILED!")
252             print(e)
253             raise
254
255     def start_grabbing(self,
strategy=pylon.GrabStrategy_LatestImageOnly):
256         try:
257             self.cam.StartGrabbing(
258                 strategy, pylon.GrabLoop_ProvidedByInstantCamera)
259             print('\nGrabbing started!')
260         except Exception as e:
261             print('\nGrabbing FAILED!')
262             print(e)
263             raise
264
265     def stop_grabbing(self):
266         try:
267             self.cam.StopGrabbing()
268             print('\nGrabbing stopped!')
269         except Exception as e:
270             print("\nStop Grabbing FAILED!")
271             print(e)
272             raise
273
274     def frame_burst(self, joinable_queue):
275         """
276         Grab frames from camera using software trigger, scheduling and
event
277         handler
278         """
279         if self.fps_count == 0:
280             self.fps_start = pg.ptime.time()
281         if self.cam.IsGrabbing():
282             if self.cam.WaitForFrameTriggerReady(
283                 300, pylon.TimeoutHandling_ThrowException):
284                 self.cam.ExecuteSoftwareTrigger()
285                 self.frame = joinable_queue.get()
286                 self.fps_count += 1
287             if self.fps_count >= 50:
288                 fps_time = pg.ptime.time() - self.fps_start
289                 self.fps = self.fps_count / fps_time
290                 print("Camera FPS: {:.2f}".format(
291                     self.cam.ResultingFrameRate.GetValue()))
292                 print("FPS: {:.2f}".format(self.fps))
293                 self.fps_count = 0
294                 QtCore.QTimer.singleShot(1, self.frame_burst)
295
```

1.3. Skripta „event_handlers.py“

```

1 # Pypylon
2 from pypylon import pylon
3
4 # Numpy
5 import numpy as np
6
7 # Time
8 import time
9
10 # Pyqtgraph
11 import pyqtgraph as pg
12 from pyqtgraph import QtCore
13
14 from gui.visualize import SignalChange
15
16
17 class FrameGrabEventHandler(pylon.ImageEventHandler):
18     def __init__(self, frame_queue):
19         """
20         Image event handler for showing grabbed frame and FPS in
FramePlotter
21         """
22         super().__init__()
23         self.frame_queue = frame_queue
24
25     def OnImageGrabbed(self, cam, grabResult):
26         if grabResult.GrabSucceeded():
27             imgArray = grabResult.GetArray()
28             self.frame_queue.put(imgArray)
29
30
31 class FrameMeasureEventHandler(pylon.ImageEventHandler):
32     def __init__(self, frame_queue, preproc_queue, procParallel):
33         """
34         Image event handler for Real-Time measurement window
35
36         :param frame_queue: Queue for displaying frames in window
37
38         :param procParallel: Object for parallel processing
39         """
40         super().__init__()
41         self.frame_queue = frame_queue
42         self.preproc_queue = preproc_queue
43         self.procParallel = procParallel
44         self.numProc = procParallel.getNumProc()
45         self.frame_num = 0
46         # Time delay in seconds between measurement
47         self.timeDelay = 10
48
49     def checkTimeDelay(self):
50         """
51         Convenience function for measuring time between frames for
measuring
52
53         :return: True of False
54         """
55         if self.startTime:
56             if (pg.ptime.time() - self.timeDelay) > self.startTime:
57                 # self.startTime = pg.ptime.time()

```

```

58         return True
59     else:
60         return False
61     return True
62
63     def OnImageEventHandlerRegistered(self, cam):
64         """
65         Start parallel processing when event handler is registered
66         """
67         self.procParallel.start()
68         # self.startTime = pg.ptime.time()
69         self.startTime = None
70
71     def OnImageGrabbed(self, cam, grabResult):
72         """
73         Called in internal thread on every image grabbed from camera
74         """
75         if grabResult.GrabSucceeded():
76             imgArray = grabResult.GetArray()
77             # Put frame in queue for displaying
78             self.frame_queue.put(imgArray)
79             # print('Put {} frame for display'.format(self.frame_num))
80             # self.frame_num += 1
81             if not self.procParallel.ifInputQueueFull():
82                 # Add frame to preprocessing if queue not full
83                 self.procParallel.addInput(imgArray)
84                 # print('Parallel addInput')
85             # Grab all results
86             while not self.procParallel.ifOutputQueueEmpty():
87                 # print('Acquiring Parallel Output')
88                 res = self.procParallel.getOutput()
89                 # Preprocessing outputs None if object is not centered
90                 if res is not None:
91                     # Preprocessing outputs np.ndarray if object is
centered
92                     if isinstance(res, np.ndarray):
93                         # print('Parallel Output is frame')
94                         # Do not put frames for preprocessing if the same
object
95                         if self.checkTimeDelay():
96                             # Put centered object in queue for
measurement
97                             self.preproc_queue.put(res)
98                             self.startTime = pg.ptime.time()
99                             # print('Parallel Output None')
100
101                         # self.frame_num += 1
102                         # if self.frame_num >= self.numProc:
103                         #     self.procParallel.join()
104                         #     # now = time.time()
105                         #     preprocImgArrays = [
106                         #         self.procParallel.getOutput() for _ in
range(self.numProc)
107                         #     ]
108                         #     # print(time.time() - now)
109                         #     [self.frame_queue.put(preprocImgArray)
110                         #      for preprocImgArray in preprocImgArrays]
111                         #     self.frame_num = 0
112
113     def OnImageEventHandlerDeregistered(self, cam):
114         """
115         Stop parallel processing when event handler is deregistered

```



```

116         """
117         self.procParallel.stop()
118         self.procParallel.join()

```

1.4. Skripta „flowcharts.py“

```

1 import pyqtgraph as pg
2 from pyqtgraph.flowchart import Flowchart, Node
3 import pyqtgraph.flowchart.library as fclib
4 from pyqtgraph.flowchart.library.common import CtrlNode
5 from pyqtgraph.Qt import QtCore, QtGui
6 from pyqtgraph.dockarea import DockArea, Dock
7
8 # Itertools
9 import itertools
10
11 # Plot Widget
12 from gui.visualize import FlowchartPlotWidget
13
14 # Custom nodes
15 from processing.nodes import (
16     FlowchartPlotNode, GaussianConvolutionNode, ThresholdNode,
17     RangeThresholdNode, OpeningNode, ClosingNode, DilationNode,
ErosionNode,
18     EdgeObjectsRemoveNode, ConvertNode, InvertNode, CannyNode,
GradientNode,
19     GradientMagnitudeNode, GradientDirectionNode, WatershedNode,
20     SeededWatershedNode, MinimaNode, MaximaNode, ApplyMaskNode,
CreateMaskNode,
21     FillNode, BinaryPropagationNode, FillHolesNode, LabelNode,
MeasureNode,
22     MeasurementDisplayNode, WorkingDistanceCorrectionNode,
CalibDisplayNode,
23     SegmentORingNode, ORingMeasurementDisplayNode, SetPixelSizeNode,
24     KuwaharaNode, BilateralFilterNode, BinaryAreaClosingNode,
25     BinaryAreaOpeningNode, BinaryClosingNode, BinaryOpeningNode,
26     BinaryDilationNode, BinaryErosionNode, OperatorPlusNode,
27     CombineMeasurementNode
28 )
29
30
31 class FlowchartCalibrateWindow:
32     def __init__(self):
33         """
34         Flowchart inside new window for system calibration
35         """
36         # Graphics window
37         self.win = QtGui.QMainWindow()
38         self.win.setWindowTitle('System Calibration Flowchart')
39         # Dock Area
40         dockArea = DockArea()
41         # Central widget
42         self.win.setCentralWidget(dockArea)
43
44         # Window Docks
45         fcWidgetDock = Dock(
46             'Flowchart Widget', size=(1, 1), hideTitle=True)
47         displayDock = Dock('Measurement and Calibration', size=(1, 1))
48         plotDocks = [
49             Dock('Plot {}'.format(i), size=(1, 1)) for i in range(4)
50         ]

```

```

51     dockArea.addDock(fcWidgetDock, 'left')
52     dockArea.addDock(displayDock, 'right', fcWidgetDock)
53     dockArea.addDock(plotDocks[0], 'top', displayDock)
54     dockArea.addDock(plotDocks[1], 'right', plotDocks[0])
55     dockArea.addDock(plotDocks[2], 'top', displayDock)
56     dockArea.addDock(plotDocks[3], 'right', plotDocks[2])
57
58     # Window size
59     self.win.resize(800, 600)
60
61     # Flowchart
62     self.fc = Flowchart(
63         terminals={
64             'dipImgIn': {'io': 'in'},
65             'CalibConstOut': {'io': 'out'},
66         }
67     )
68     # row, column, rowspan, colspan
69     fcWidgetDock.addWidget(self.fc.widget())
70
71     # Plot widgets
72     self.plt_widg = [FlowchartPlotWidget() for _ in
range(len(plotDocks))]
73     [plotDocks[i].addWidget(
74         self.plt_widg[i]) for i in range(len(plotDocks))]
75
76     # Graphics layout for displays
77     displayLayout = pg.GraphicsLayoutWidget(border='w')
78     displayDock.addWidget(displayLayout)
79
80     # Display widgets
81     self.disp_widg = []
82     self.disp_widg.append(displayLayout.addLabel(
83         '', colspan=2, justify='left'))
84     displayLayout.nextRow()
85     self.disp_widg.append(displayLayout.addLabel(
86         '', colspan=2, justify='left'))
87
88     # Flowchart library copy - custom nodes available to user
89     self.fc_library = fclib.LIBRARY.copy()
90     [self.fc_library.addNodeType(
91         nd, [('dipImage', 'Display')]) for nd in (
92         FlowchartPlotNode, MeasurementDisplayNode, CalibDisplayNode,
93         ORingMeasurementDisplayNode
94     )]
95     # Filter nodes
96     [self.fc_library.addNodeType(
97         nd, [('dipImage', 'Filters')]) for nd in (
98         GaussianConvolutionNode, GradientNode, GradientMagnitudeNode,
99         GradientDirectionNode, EdgeObjectsRemoveNode, KuwaharaNode,
100         BilateralFilterNode
101     )]
102     # Binary Filter nodes
103     [self.fc_library.addNodeType(
104         nd, [('dipImage', 'Binary')]) for nd in (
105         BinaryClosingNode, BinaryOpeningNode, BinaryErosionNode,
106         BinaryDilationNode, BinaryAreaClosingNode,
BinaryAreaOpeningNode,
107         BinaryPropagationNode, FillHolesNode
108     )]
109     # Segmentation nodes
110     [self.fc_library.addNodeType(

```

```

111         nd, [('dipImage', 'Segmentation')]) for nd in (
112         ThresholdNode, RangeThresholdNode, MinimaNode, MaximaNode,
113         WatershedNode, SeededWatershedNode, CannyNode,
SegmentORingNode
114     ])
115     # Morphological nodes
116     [self.fc_library.addNodeType(
117         nd, [('dipImage', 'Morphological')]) for nd in (
118         DilationNode, ErosionNode, OpeningNode, ClosingNode)]
119     # Image nodes
120     [self.fc_library.addNodeType(
121         nd, [('dipImage', 'Image')]) for nd in (
122         ConvertNode, FillNode, LabelNode, SetPixelSizeNode
123     )]
124     # Arithmetics nodes
125     [self.fc_library.addNodeType(
126         nd, [('dipImage', 'Arithmetics')]) for nd in (
127         InvertNode, ApplyMaskNode, CreateMaskNode, OperatorPlusNode
128     )]
129     # Measurement nodes
130     [self.fc_library.addNodeType(
131         nd, [('dipImage', 'Measurement')]) for nd in (
132         MeasureNode, WorkingDistanceCorrectionNode,
CombineMeasurementNode
133     )]
134     self.fc.setLibrary(self.fc_library)
135
136     # Plot nodes and Widget connection
137     plt_nodes_x = (-20+x*120 for x in range(len(self.plt_widg)))
138     plt_nodes = [self.fc.createNode(
139         'FlowchartPlot', pos=(x, -60)) for x in plt_nodes_x]
140     [nd.set_fcPlotWidget(
141         widg) for nd, widg in zip(plt_nodes, self.plt_widg)]
142
143     # Connecting plot and display widgets with nodes
144     self.fc.sigFileLoaded.connect(self.setFlowchartPlotWidgets)
145     self.fc.sigFileLoaded.connect(self.setDisplayWidgets)
146     self.fc.sigChartChanged.connect(self.setFlowchartPlotWidgets)
147     self.fc.sigChartChanged.connect(self.setDisplayWidgets)
148
149     def setFlowchartPlotWidgets(self):
150         """
151         Connect Flowchart plot widgets with nodes after loading fc file
152         """
153         nd_list = []
154         for name, node in self.fc.nodes().items():
155             if isinstance(node, FlowchartPlotNode):
156                 nd_list.append(node)
157         [nd.set_fcPlotWidget(widg) for nd, widg in zip(nd_list,
self.plt_widg)]
158
159     def setDisplayWidgets(self):
160         """
161         Connect Display widgets with nodes
162         """
163         for name, node in self.fc.nodes().items():
164             if isinstance(node, MeasurementDisplayNode):
165                 node.setDisplayWidget(self.disp_widg[0])
166             if isinstance(node, CalibDisplayNode):
167                 node.setDisplayWidget(self.disp_widg[1])
168
169     def setInput(self, dip_img):

```

```
170         """
171         Set Flowchart input
172         """
173         self.fc.setInput(dipImgIn=dip_img)
174
175     def fc_process(self, dip_img):
176         """
177         Process data with display=False (speed increase)
178         """
179         return self.fc.process(dipImgIn=dip_img) ['CalibConstOut']
180
181     def output(self):
182         """
183         Process data through flowchart with display=True and return
output
184         """
185         return self.fc.output() ['CalibConstOut']
186
187     def show(self):
188         self.win.show()
189
190     def close(self):
191         self.win.hide()
192
193
194 class FlowchartMeasureWindow:
195     def __init__(self):
196         """
197         Flowchart inside new window for measuring objects
198         """
199         # Initial mm/px ratio
200         self.mmPxRatio = 1
201         # Graphics window
202         self.win = QtGui.QMainWindow()
203         self.win.setWindowTitle('Object Measurement Flowchart')
204         # Dock Area
205         dockArea = DockArea()
206         # Central widget
207         self.win.setCentralWidget(dockArea)
208
209         # Window Docks
210         fcWidgetDock = Dock(
211             'Flowchart Widget', size=(1, 1), hideTitle=True)
212         displayDock = Dock('Measurement and Calibration', size=(1, 1))
213         plotDocks = [
214             Dock('Plot {}'.format(i), size=(1, 1)) for i in range(4)
215         ]
216         dockArea.addDock(fcWidgetDock, 'left')
217         dockArea.addDock(displayDock, 'right', fcWidgetDock)
218         dockArea.addDock(plotDocks[0], 'top', displayDock)
219         dockArea.addDock(plotDocks[1], 'right', plotDocks[0])
220         dockArea.addDock(plotDocks[2], 'top', displayDock)
221         dockArea.addDock(plotDocks[3], 'right', plotDocks[2])
222
223         # Window size
224         self.win.resize(800, 600)
225
226         # Flowchart
227         self.fc = Flowchart(
228             terminals={
229                 'dipImgIn': {'io': 'in'},
230                 'mmPxRatio': {'io': 'in'},
```

```

231         'dipMeasurementOut': {'io': 'out'},
232     }
233 )
234 # row, column, rowspan, colspan
235 fcWidgetDock.addWidget(self.fc.widget())
236
237 # Plot widgets
238 self.plt_widg = [FlowchartPlotWidget() for _ in
range(len(plotDocks))]
239 [plotDocks[i].addWidget(
240     self.plt_widg[i]) for i in range(len(plotDocks))]
241
242 # Graphics layout for displays
243 displayLayout = pg.GraphicsLayoutWidget(border='w')
244 displayDock.addWidget(displayLayout)
245
246 # Display widgets
247 self.disp_widg = []
248 self.disp_widg.append(displayLayout.addLabel(
249     '', colspan=2, justify='left'))
250 displayLayout.nextRow()
251 self.disp_widg.append(displayLayout.addLabel(
252     '', colspan=2, justify='left'))
253
254 # Flowchart library copy - custom nodes available to user
255 self.fc_library = fclib.LIBRARY.copy()
256 [self.fc_library.addNodeType(
257     nd, [('dipImage', 'Display')]) for nd in (
258     FlowchartPlotNode, MeasurementDisplayNode, CalibDisplayNode,
259     ORingMeasurementDisplayNode
260 )]
261 # Filter nodes
262 [self.fc_library.addNodeType(
263     nd, [('dipImage', 'Filters')]) for nd in (
264     GaussianConvolutionNode, GradientNode, GradientMagnitudeNode,
265     GradientDirectionNode, EdgeObjectsRemoveNode, KuwaharaNode,
266     BilateralFilterNode
267 )]
268 # Binary Filter nodes
269 [self.fc_library.addNodeType(
270     nd, [('dipImage', 'Binary')]) for nd in (
271     BinaryClosingNode, BinaryOpeningNode, BinaryErosionNode,
272     BinaryDilationNode, BinaryAreaClosingNode,
BinaryAreaOpeningNode,
273     BinaryPropagationNode, FillHolesNode
274 )]
275 # Segmentation nodes
276 [self.fc_library.addNodeType(
277     nd, [('dipImage', 'Segmentation')]) for nd in (
278     ThresholdNode, RangeThresholdNode, MinimaNode, MaximaNode,
279     WatershedNode, SeededWatershedNode, CannyNode,
SegmentORingNode
280 )]
281 # Morphological nodes
282 [self.fc_library.addNodeType(
283     nd, [('dipImage', 'Morphological')]) for nd in (
284     DilationNode, ErosionNode, OpeningNode, ClosingNode)]
285 # Image nodes
286 [self.fc_library.addNodeType(
287     nd, [('dipImage', 'Image')]) for nd in (
288     ConvertNode, FillNode, LabelNode, SetPixelSizeNode
289 )]

```

```

290     # Arithmetics nodes
291     [self.fc_library.addNodeType(
292         nd, [('dipImage', 'Arithmetics')]) for nd in (
293         InvertNode, ApplyMaskNode, CreateMaskNode, OperatorPlusNode
294     )]
295     # Measurement nodes
296     [self.fc_library.addNodeType(
297         nd, [('dipImage', 'Measurement')]) for nd in (
298         MeasureNode, WorkingDistanceCorrectionNode,
CombineMeasurementNode
299     )]
300     self.fc.setLibrary(self.fc_library)
301
302     # Plot nodes and Widget connection
303     plt_nodes_x = (-20+x*120 for x in range(len(self.plt_widg)))
304     plt_nodes = [self.fc.createNode(
305         'FlowchartPlot', pos=(x, -60)) for x in plt_nodes_x]
306     [nd.set_fcPlotWidget(
307         widg) for nd, widg in zip(plt_nodes, self.plt_widg)]
308
309     # Connecting plot and display widgets with nodes
310     self.fc.sigFileLoaded.connect(self.setFlowchartPlotWidgets)
311     self.fc.sigFileLoaded.connect(self.setDisplayWidgets)
312     self.fc.sigChartChanged.connect(self.setFlowchartPlotWidgets)
313     self.fc.sigChartChanged.connect(self.setDisplayWidgets)
314
315     def __getstate__(self):
316         """
317         When pickling retain only Flowchart related attributes
318         Flowchart objects can't be pickled
319
320         :return: self.__dict__
321         """
322         state = self.__dict__.copy()
323         flowchart_state = {
324             key:value for key, value in state.items() if
325             key in ['fc', 'fc_process', 'mmPxRatio']}
326     }
327     return flowchart_state
328
329     def setFlowchartPlotWidgets(self):
330         """
331         Connect Flowchart plot widgets with nodes after loading fc file
332         """
333         nd_list = []
334         for name, node in self.fc.nodes().items():
335             if isinstance(node, FlowchartPlotNode):
336                 nd_list.append(node)
337         [nd.set_fcPlotWidget(widg) for nd, widg in zip(nd_list,
self.plt_widg)]
338
339     def setDisplayWidgets(self):
340         """
341         Connect Display widgets with nodes
342         """
343         for name, node in self.fc.nodes().items():
344             if isinstance(node, ORingMeasurementDisplayNode):
345                 node.setDisplayWidgetList(self.disp_widg)
346
347     def setInput(self, dip_img, mmPxRatio):
348         """
349         Set Flowchart input

```

```

350         """
351         self.mmPxRatio = mmPxRatio
352         self.fc.setInput(dipImgIn=dip_img, mmPxRatio=mmPxRatio)
353
354     def fc_process(self, dip_img):
355         """
356         Process data with display=False (speed increase) and return
output
357         """
358         return self.fc.process(
359             dipImgIn=dip_img,
mmPxRatio=self.mmPxRatio)['dipMeasurementOut']
360
361     def output(self):
362         """
363         Process data through flowchart with display=True
364         """
365         return self.fc.output()['dipMeasurementOut']
366
367     def show(self):
368         self.win.show()
369
370     def close(self):
371         self.win.hide()
372

```

1.5. Skripta „mainwindow.py“

```

1 import sys
2
3 # Visualize frames
4 from gui.visualize import FrameWindow, FlowchartPrepWindow,\
5     MeasureRealtimeWindow, SignalChange
6
7 # Flowchart frames
8 from gui.flowcharts import FlowchartCalibrateWindow,
FlowchartMeasureWindow
9
10 # PyQtGraph
11 import pyqtgraph as pg
12 from gui.ptrees import CameraParams, CalibrationParams
13 from pyqtgraph.parameterstree import ParameterTree
14 from pyqtgraph import QtCore, QtGui
15 from pyqtgraph.dockarea import DockArea, Dock
16 from pyqtgraph import ptime
17
18 # Numpy
19 import numpy as np
20
21 # DIPlib
22 import PyDIP as dip
23
24 # Camera object
25 from cam.cam import CamObject
26 from cam.event_handlers import FrameGrabEventHandler,
FrameMeasureEventHandler
27 from pypylon import pylon
28
29 # Parallel processing
30 from processing.process import ProcessParallel
31

```

```
32 # Queue module
33 import queue
34
35
36 class MeasuringApp(CamObject, QtGui.QMainWindow):
37     def __init__(self, *args, **kwargs):
38         CamObject.__init__(self)
39         QtGui.QMainWindow.__init__(self)
40         self.resize(700, 150)
41         self.setWindowTitle('Workpiece Dimensional Control')
42         self.area = DockArea()
43         self.setCentralWidget(self.area)
44
45         # Window Docks
46         self.camMenuDock = Dock('Camera Menu', size=(1, 1))
47         self.procMenuDock = Dock('Analysis Menu', size=(1, 1))
48         self.paramDock = Dock('Parameter Tree', size=(250, 150))
49         self.area.addDock(self.camMenuDock, 'right')
50         self.area.addDock(self.procMenuDock, 'right')
51         self.area.addDock(self.paramDock, 'left')
52
53         # Parameter Tree
54         self._parameter_tree = ParameterTree()
55         self._parameter_tree.setWindowTitle('Parameter Tree')
56         self.paramDock.addWidget(self._parameter_tree)
57
58         # Camera Menu
59         self._cam_menu = pg.LayoutWidget(parent=self.camMenuDock)
60         self._cam_buttons = self.cam_buttons()
61         [btn.resize(btn.minimumSizeHint()) for btn in self._cam_buttons]
62         [self._cam_menu.addWidget(btn, row=i, col=0) for btn, i in zip(
63             self._cam_buttons, range(len(self._cam_buttons)))]
64         self.camMenuDock.addWidget(self._cam_menu)
65
66         # Analysis menu
67         self._proc_menu = pg.LayoutWidget(parent=self.procMenuDock)
68         self._proc_buttons = self.process_buttons()
69         [btn.resize(btn.minimumSizeHint()) for btn in self._proc_buttons]
70         [self._proc_menu.addWidget(btn, row=i, col=1) for btn, i in zip(
71             self._proc_buttons, range(len(self._proc_buttons)))]
72         self.procMenuDock.addWidget(self._proc_menu)
73
74         # Original frames window
75         self._orig_frame = FrameWindow("Original")
76
77         # Flowchart input preparation window
78         self._prep_flow = FlowchartPrepWindow('Flowchart Input
preparation')
79
80         # Calibrate system window
81         self._calib_window = FlowchartCalibrateWindow()
82         # Calibration Parameters
83         self._calib_params = None
84
85         # Measurement preparation window
86         self._measure_prep_window = FlowchartMeasureWindow()
87
88         # Realtime measurement window
89         self._realtime_measure_window = MeasureRealtimeWindow()
90
91         # Show Main Window
92         self.show()
```



```
93
94     # Result Queue init
95     self.preproc_queue = queue.Queue()
96
97     # FPS display count
98     self.fps_display_count = 0
99
100    def cam_buttons(self):
101        """
102        Menu buttons as list
103        """
104        # Camera create button
105        cam_create = QtGui.QPushButton('Find Camera',
parent=self._cam_menu)
106        cam_create.clicked.connect(self.create)
107        # Camera open button
108        cam_open = QtGui.QPushButton('Open Camera',
parent=self._cam_menu)
109        cam_open.clicked.connect(self.open)
110        # Camera print info
111        cam_print_info = QtGui.QPushButton('Print Info',
parent=self._cam_menu)
112        cam_print_info.clicked.connect(self.print_info)
113        # Camera start grabbing
114        cam_start_grab = QtGui.QPushButton(
115            'Start Grabbing', parent=self._cam_menu)
116        cam_start_grab.clicked.connect(self.start_grabbing)
117        # Camera stop grabbing
118        cam_stop_grab = QtGui.QPushButton(
119            'Stop Grabbing', parent=self._cam_menu)
120        cam_stop_grab.clicked.connect(self.stop_grabbing)
121        # Camera close
122        cam_close = QtGui.QPushButton('Close Camera',
parent=self._cam_menu)
123        cam_close.clicked.connect(self.close)
124        return [
125            cam_create, cam_open, cam_print_info, cam_start_grab,
126            cam_stop_grab, cam_close
127        ]
128
129    def process_buttons(self):
130        """
131        Process buttons as list
132        """
133        # Grab single frame button
134        proc_grab_frame = QtGui.QPushButton(
135            'Grab Frame', parent=self._proc_menu)
136        proc_grab_frame.clicked.connect(self.grab_single_frame)
137        proc_discard_frame = QtGui.QPushButton(
138            'Discard Frame', parent=self._proc_menu)
139        proc_discard_frame.clicked.connect(self.discard_frame)
140        proc_calib_sys = QtGui.QPushButton(
141            'Calibrate System', parent=self._proc_menu)
142        proc_calib_sys.clicked.connect(self.calibrate_system)
143        proc_set_calib_const = QtGui.QPushButton(
144            'Set Calibration Constants', parent=self._proc_menu)
145        proc_set_calib_const.clicked.connect(self.set_calib_const)
146        proc_measure_prep = QtGui.QPushButton(
147            'Measurement Preparation', parent=self._proc_menu)
148        proc_measure_prep.clicked.connect(self.measurement_preparation)
149        proc_realtime_start = QtGui.QPushButton(
150            'Start Measurement', parent=self._proc_menu)
```

```
151
proc_realtime_start.clicked.connect(self.start_realtime_measurement)
152     proc_realtime_stop = QtGui.QPushButton(
153         'Stop Measurement', parent=self._proc_menu)
154
proc_realtime_stop.clicked.connect(self.stop_realtime_measurement)
155     return [
156         proc_grab_frame, proc_discard_frame, proc_calib_sys,
157         proc_set_calib_const, proc_measure_prep, proc_realtime_start,
158         proc_realtime_stop
159     ]
160
161     def open(self):
162         """
163         Reimplementation of open function to update GUI numbers and
sliders
164         """
165         super().open()
166         self._parameter_tree.clear()
167         self._parameter_tree.addParameters(
168             CameraParams(name='Camera Parameters', cam_obj=self),
showTop=True)
169
170     def close(self):
171         """
172         Reimplementation of close function to update GUI numbers and
sliders
173         """
174         super().close()
175         self._parameter_tree.clear()
176
177     def set_default_params(self):
178         """
179         Reimplementation of set_default_params function to update GUI
180         """
181         super().set_default_params()
182         self._cam_gain.value = self.cam.Gain()
183         self._cam_exposure.value = self.cam.ExposureTime()
184         self._cam_width.value = self.cam.Width.GetValue()
185         self._cam_height.value = self.cam.Height.GetValue()
186         self._cam_offsetx.value = self.cam.OffsetX.GetValue()
187         self._cam_offsety.value = self.cam.OffsetY.GetValue()
188
189     def start_grabbing(self,
strategy=pylon.GrabStrategy_LatestImageOnly):
190         """
191         Reimplementation of start_grabbing to show new window
192         """
193         if not self.cam.IsGrabbing():
194             self._orig_frame.show()
195             self.frame_queue = queue.Queue()
196             fg_hndl = FrameGrabEventHandler(self.frame_queue)
197             # Frame grab event registering
198             self.cam.RegisterImageEventHandler(
199                 fg_hndl,
200                 pylon.RegistrationMode_ReplaceAll,
201                 pylon.Cleanup_Delete
202             )
203             super().start_grabbing(strategy)
204             self.frame_burst(graphicsObject=self._orig_frame)
205
206     def stop_grabbing(self):
```

```

207         """
208         Reimplementation of stop_grabbing to close new window
209         """
210         if self.cam.IsGrabbing():
211             super().stop_grabbing()
212             self._orig_frame.close()
213
214         def frame_burst(self, graphicsObject, updateFrame=True,
215 updateFPS=True,
216 updateMeasurementResult=False, numProc=1,
217 fps_display=5):
218             """
219             Grab frames from camera using software trigger, scheduling and
220             event
221             handler.
222             frame_burst reimplementation for FPS, frame and measurement
223             result
224             update.
225             :param graphicsObject: object which methods are called and
226             graphics updated
227             :param updateFrame: bool
228             :param updateFPS: bool
229             :param updateMeasurementResult: bool
230             :param numProc: 1
231             :param fps_display: int
232             """
233             if self.fps_count == 0:
234                 if updateFPS:
235                     # Update graphicsObject FPS label
236                     graphicsObject.update_fps(self.fps)
237                     self.fps_start = pg.ptime.time()
238             if self.cam.IsGrabbing():
239                 for _ in range(numProc):
240                     if self.cam.WaitForFrameTriggerReady(
241                         300, pylon.TimeoutHandling_ThrowException):
242                         self.cam.ExecuteSoftwareTrigger()
243                     frames = [self.frame_queue.get() for _ in range(numProc)]
244                     self.fps_display_count += 1
245                     # print('FPS_display_count:
246                     {}.format(self.fps_display_count))
247                     if updateFrame and not (self.fps_display_count %
248                     fps_display):
249                         self.fps_display_count = 0
250                         # Update graphicsObject ImageItem
251                         [graphicsObject.update_frame(frame) for frame in frames]
252                         self.fps_count += numProc
253                     if updateMeasurementResult:
254                         while not self.preproc_queue.empty():
255                             graphicsObject.update_measurement_result(
256                                 self._measure_prep_window.fc_process(
257                                     dip.Image(self.preproc_queue.get())
258                                 ))
259                     # Average FPS on number of frames
260                     if self.fps_count >= 10:
261                         fps_time = pg.ptime.time() - self.fps_start
262                         self.fps = self.fps_count / fps_time
263                         print("Camera FPS: {:.2f}".format(
264                             self.cam.ResultingFrameRate.GetValue()))
265                         print("FPS: {:.2f}".format(self.fps))
266                         self.fps_count = 0
267                     QtCore.QTimer.singleShot(

```

```
262         1, lambda : self.frame_burst(graphicsObject,
263                                     updateFrame,
264                                     updateFPS,
265                                     numProc,
266                                     ))
267
268     def grab_single_frame(self):
269         self._prep_flow.set_frame(self._orig_frame.grab_single_frame())
270         self._prep_flow.show()
271
272     def discard_frame(self):
273         self._prep_flow.close()
274
275     def calibrate_system(self):
276         """
277         Show Calibration flowchart and set input frame
278         """
279         self._calib_window.setInput(self._prep_flow.get_dip_slice())
280         self._calib_window.show()
281
282     def set_calib_const(self):
283         """
284         Set and show calibration parameters in parameterTree
285         """
286         standard_dict = self._calib_window.output()
287         self._calib_params = CalibrationParams(
288             name='Calibration Parameters', standard_dict=standard_dict)
289         self._parameter_tree.clear()
290         self._parameter_tree.addParameters(self._calib_params,
showTop=True)
291
292     def measurement_preparation(self):
293         """
294         Show Measurement preparation flowchart and set input frame
295         """
296         self._measure_prep_window.setInput(
297             self._prep_flow.get_dip_slice(),
298             self._calib_params.obj_mm_px.value()
299         )
300         self._measure_prep_window.show()
301
302     def start_realtime_measurement(
303         self, strategy=pylon.GrabStrategy_LatestImageOnly):
304         """
305         Show window for realtime measurement and start grabbing frames
from cam
306         """
307         if not self.cam.IsGrabbing():
308             self._realtime_measure_window.show()
309             procParallel = ProcessParallel(
310                 self._measure_prep_window, numberProc=3
311             )
312             # Queue for storing frames
313             self.frame_queue = queue.Queue()
314             # Queue for storing measurement result list
315             self.preproc_queue = queue.Queue()
316             self.fm_hndl = FrameMeasureEventHandler(
317                 self.frame_queue, self.preproc_queue, procParallel
318             )
319             # Frame grab event registering
320             self.cam.RegisterImageEventHandler(
321                 self.fm_hndl,
```

```

322         pylon.RegistrationMode_ReplaceAll,
323         pylon.Cleanup_Delete
324     )
325     super().start_grabbing(strategy)
326     self.frame_burst(
327         graphicsObject=self._realtime_measure_window,
328         updateFPS=False, updateMeasurementResult=True,
329         fps_display=10,
330         # numProc=procParallel.getNumProc()
331     )
332
333     def stop_realtime_measurement(self):
334         """
335         Close window for realtime measurement and stop grabbing frames
336         """
337         if self.cam.IsGrabbing():
338             super().stop_grabbing()
339             self._realtime_measure_window.close()
340             self.cam.DeregisterImageEventHandler(self.fm_hndl)

```

1.6. Skripta „ptrees.py“

```

1 # PyQtGraph
2 import pyqtgraph as pg
3 import pyqtgraph.parameterTree.parameterTypes as pTypes
4 from pyqtgraph.parameterTree import Parameter, ParameterTree,
ParameterItem, \
5     registerParameterType
6
7
8 class CameraParams(pTypes.GroupParameter):
9     def __init__(self, cam_obj, **opts):
10         """
11         Camera parameters group
12
13         :param cam_obj: camera object
14
15         :param opts: dictionary with options passed to GroupParameter
16         """
17         opts['type'] = 'group'
18         # Camera object instance
19         self.cam = cam_obj.cam
20         # Functions for changing parameters
21         self.set_width = cam_obj.set_width
22         self.set_height = cam_obj.set_height
23         self.set_offsetx = cam_obj.set_offsetx
24         self.set_offsety = cam_obj.set_offsety
25         self.set_gain = cam_obj.set_gain
26         self.set_exposure_time = cam_obj.set_exposure_time
27         # Amplification of camera signal manual control
28         self.cam.GainAuto.SetValue("Off")
29         # Exposure Time manual control
30         self.cam.ExposureAuto.SetValue("Off")
31         pTypes.GroupParameter.__init__(self, **opts)
32
33         self.addChildren([
34             {
35                 'name': 'ROI', 'title': 'ROI Size and position',
36                 'type': 'group', 'children':
37                 [

```

```

38         # Camera Width Parameter
39         {
40             'name': 'Width', 'type': 'int',
41             'value': self.cam.Width(),
42             'default': 2592,
43             'decimals': 4,
44             'limits': (self.cam.Width.Min,
self.cam.Width.Max),
45             'step': 32,
46             'suffix': 'px', 'siPrefix': False
47         },
48         # Camera Height Parameter
49         {
50             'name': 'Height', 'type': 'int',
51             'value': self.cam.Height(),
52             'default': 2048,
53             'decimals': 4,
54             'limits': (self.cam.Height.Min,
self.cam.Height.Max),
55             'step': 2,
56             'suffix': 'px', 'siPrefix': False
57         },
58         # Camera OffsetX Parameter
59         {
60             'name': 'OffsetX', 'type': 'int',
61             'value': self.cam.OffsetX(),
62             'default': 0,
63             'decimals': 4,
64             'limits': (self.cam.OffsetX.Min,
self.cam.OffsetX.Max),
65             'step': 32,
66             'suffix': 'px', 'siPrefix': False
67         },
68         # Camera OffsetX Parameter
69         {
70             'name': 'OffsetY', 'type': 'int',
71             'value': self.cam.OffsetY(),
72             'default': 0,
73             'decimals': 4,
74             'limits': (self.cam.OffsetY.Min,
self.cam.OffsetY.Max),
75             'step': 2,
76             'suffix': 'px', 'siPrefix': False
77         },
78     ]
79     },
80     {
81         'name': 'Gain', 'type': 'float', 'value':
self.cam.Gain(),
82         'default': 12,
83         'decimals': 5,
84         'limits': (self.cam.Gain.Min, self.cam.Gain.Max),
85         'step': 0.05,
86         'suffix': ' dB', 'siPrefix': False
87     },
88     {
89         'name': 'Exposure Time', 'type': 'float',
90         'value': self.cam.ExposureTime()*1e-6,
91         'default': 9500*1e-6,
92         'decimals': 12,
93         'limits': (81*1e-6, 150000*1e-6),
94         'step': 100*1e-6,

```

```
195         'suffix': 's', 'siPrefix': True
196     },
197 ]
198
199     # Connect OnValueChanged
200     self.connect_recursive(self, self.OnValueChanged)
201
202     def connect_recursive(self, param, func):
203         """
204         Recursively connect func to sigValueChanged of all children of
205 param
206         """
207         param.sigValueChanged.connect(func)
208         if param.hasChildren():
209             [self.connect_recursive(ch, func) for ch in param.children()]
210
211     def OnValueChanged(self, param, value):
212         """
213         Function which handles parameter value changes
214         """
215         if param.name() == 'Width':
216             self.child('ROI', 'OffsetX').setLimits((
217                 0, self.cam.Width.Max - value))
218             self.set_width(value)
219         if param.name() == 'Height':
220             self.child('ROI', 'OffsetY').setLimits((
221                 0, self.cam.Height.Max - value))
222             self.set_height(value)
223         if param.name() == 'OffsetX':
224             self.child('ROI', 'Width').setLimits((
225                 self.cam.Width.Min, self.cam.Width.Max - value))
226             self.set_offsetx(value)
227         if param.name() == 'OffsetY':
228             self.child('ROI', 'Height').setLimits((
229                 self.cam.Height.Min, self.cam.Height.Max - value))
230             self.set_offsety(value)
231         if param.name() == 'Gain':
232             self.set_gain(value)
233         if param.name() == 'Exposure Time':
234             self.set_exposure_time(value*1e6)
235
236 class CalibrationParams(pTypes.GroupParameter):
237     def __init__(self, standard_dict, **opts):
238         """
239         Calibration parameters group
240
241         :param standard_dict:
242         dictionary with standard height, working distance and mm/px ratio
243
244         :param opts: dictionary with options passed to GroupParameter
245         """
246         opts['type'] = 'group'
247         super().__init__(**opts)
248
249         self.addChildren([
250             {
251                 'name': 'Standard', 'title': 'Standard',
252                 'type': 'group', 'children':
253                 [
254                     # Standard height
255                     {
```

```

156         'name': 'Height', 'type': 'float',
157         'value': standard_dict['std_height'],
158         'default': standard_dict['std_height'],
159         'limits': (0, None),
160         'step': 0.001,
161         'decimals': 6,
162         'suffix': ' mm', 'siPrefix': False
163     },
164     # Standard working distance
165     {
166         'name': 'Working Distance', 'type': 'float',
167         'value': standard_dict['std_work_dist'],
168         'default': standard_dict['std_work_dist'],
169         'limits': (0, None),
170         'step': 0.01,
171         'decimals': 6,
172         'suffix': ' mm', 'siPrefix': False
173     },
174     # Standard mm/px ration
175     {
176         'name': 'mm/px Ratio', 'type': 'float',
177         'value': standard_dict['std_mm_px'],
178         'default': standard_dict['std_mm_px'],
179         'limits': (0, None),
180         'step': 0.00001,
181         'decimals': 8,
182         'suffix': ' mm/px', 'siPrefix': False
183     },
184 ]
185 },
186 {
187     'name': 'Object', 'title': 'Measured Object',
188     'type': 'group', 'children':
189     [
190         # Measured object height
191         {
192             'name': 'Height', 'type': 'float',
193             'value': 2.3967111111,
194             'default': 2.3967111111,
195             'limits': (0, None),
196             'step': 0.001,
197             'decimals': 6,
198             'suffix': ' mm', 'siPrefix': False
199         },
200         # Measured object mm/px ration
201         {
202             'name': 'mm/px Ratio', 'type': 'float',
203             'value': 0,
204             'limits': (0, None),
205             'decimals': 8,
206             'suffix': ' mm/px', 'siPrefix': False,
207             'readonly': True
208         },
209     ]
210 },
211 ])
212
213 # Parameters to attributes
214 self.std_height = self.child('Standard', 'Height')
215 self.std_work_dist = self.child('Standard', 'Working Distance')
216 self.std_mm_px = self.child('Standard', 'mm/px Ratio')
217 self.obj_height = self.child('Object', 'Height')

```



```

218         self.obj_mm_px = self.child('Object', 'mm/px Ratio')
219         # Setting object mm/px ratio initial value
220         self.obj_mm_px_calc()
221
222         # Connect obj_mm_px_calc
223         self.connect_recursive(self, self.obj_mm_px_calc)
224
225     def obj_mm_px_calc(self):
226         """
227         Calculate mm/px ratio on object height image using thin lens
equation
228         and linear interpolation
229         """
230         self.obj_mm_px.setValue(
231             (1. + (self.std_height.value() - self.obj_height.value()) /
232              self.std_work_dist.value()) * self.std_mm_px.value()
233         )
234
235     def connect_recursive(self, param, func):
236         """
237         Recursively connect func to sigValueChanged of all children of
param
238         except obj_mm_px
239         """
240         if param is not self.obj_mm_px:
241             param.sigValueChanged.connect(func)
242             if param.hasChildren():
243                 [self.connect_recursive(ch, func) for ch in
param.children()]
244

```

1.7. Skripta „visualize.py“

```

1 # Pylon
2 from pypylon import pylon
3
4 # Numpy
5 import numpy as np
6
7 #PyDIP
8 import PyDIP as dip
9
10 from pyqtgraph import QtCore, QtGui
11 from pyqtgraph.dockarea import DockArea, Dock
12 import pyqtgraph.widgets.RemoteGraphicsView
13 import pyqtgraph as pg
14 pg.setConfigOptions(imageAxisOrder='row-major')
15
16 # Math
17 import math
18
19
20 class FrameWindow:
21     def __init__(self, title):
22         """
23         Window for showing grabbed frames
24         """
25         # Graphics window
26         self.win = pg.GraphicsLayoutWidget()
27         self.win.setWindowTitle(title)
28         self.win.resize(600, 400)

```

```

29     # View Box
30     self.view = pg.ViewBox(invertY=True, lockAspect=1)
31     # Image item
32     self.img = pg.ImageItem(border='w')
33     self.view.addItem(self.img)
34     # Plot for axes
35     self.win.addPlot(
36         row=1, col=1, title='Original Frames', viewBox=self.view)
37     # FPS label
38     self.fpsLabel = self.win.addLabel(
39         'FPS', row=2, col=1, justify='right')
40     # Image histogram
41     self.hist = pg.HistogramLUTItem(image=self.img,
fillHistogram=False)
42     self.hist.setLevels(0, 255)
43     self.win.addItem(self.hist, row=1, col=2, rowspan=2)
44
45     def update_frame(self, frame):
46         self.img.setImage(frame)
47
48     def update_fps(self, fps_value):
49         self.fpsLabel.setText(
50             text="FPS: {:.2f}".format(fps_value), bold=True)
51
52     def grab_single_frame(self):
53         return self.img.image
54
55     def show(self):
56         self.win.show()
57
58     def close(self):
59         self.win.hide()
60
61
62 class FlowchartPrepWindow:
63     """
64     Window for preparing image for flowchart
65     """
66     def __init__(self, title):
67         # Empty objects
68         self.frame = None
69         self.frame_dip = None
70         self.frame_dip_roi = None
71         # Graphics window
72         self.win = pg.GraphicsLayoutWidget()
73         self.win.setWindowTitle(title)
74         self.win.resize(600, 600)
75         self.win.centralWidget.setBorder({'color': (255, 255, 255, 100),
76                                         'width': 2})
77         # View Box
78         self.view = pg.ViewBox(invertY=True, lockAspect=1)
79         # Image item
80         self.img = pg.ImageItem(border='w')
81         self.view.addItem(self.img)
82         # Plot for axes
83         self.plot = self.win.addPlot(
84             row=1, col=1, rowspan=2, title='Flowchart Input preparation',
85             viewBox=self.view
86         )
87         self.plot.showGrid(x=True, y=True, alpha=0.8)
88         # Empty label for proper layout
89         self.win.addLabel(row=2, col=1)

```

```

90     # Crosshair
91     self.vline = pg.InfiniteLine(
92         angle=90, pen={'color': 'g', 'width': 1}, movable=False)
93     self.hline = pg.InfiniteLine(
94         angle=0, pen={'color': 'g', 'width': 1}, movable=False)
95     self.plot.addItem(self.vline, ignoreBounds=True)
96     self.plot.addItem(self.hline, ignoreBounds=True)
97     self.proxy = pg.SignalProxy(
98         self.plot.scene().sigMouseClicked, rateLimit=60,
99         slot=self.on_mouse_move
100    )
101    # Pixel intensity label
102    self.px_lbl = self.win.addLabel(
103        'Pixel Intensity: ', row=3, col=1, bold=True,
justify='right')
104    # Image histogram
105    self.hist = pg.HistogramLUTItem(
106        image=self.img, fillHistogram=True)
107    self.hist.setLevels(0, 255)
108    self.win.addItem(self.hist, row=1, col=2, rowspan=4)
109    # ROI selection
110    self.roi = pg.ROI(
111        pos=(0, 0),
112        pen={'color': 'b', 'width': 2},
113    )
114    # ROI Scaling Handles
115    self.roi.addScaleHandle(pos=(0, 0), center=(1, 1))
116    self.roi.addScaleHandle(pos=(0, 1), center=(1, 0))
117    self.roi.addScaleHandle(pos=(1, 0), center=(0, 1))
118    self.roi.addScaleHandle(pos=(1, 1), center=(0, 0))
119    self.roi.addScaleHandle(pos=(0, 0.5), center=(1, 0.5))
120    self.roi.addScaleHandle(pos=(0.5, 0), center=(0.5, 1))
121    self.roi.addScaleHandle(pos=(0.5, 1), center=(0.5, 0))
122    self.roi.addScaleHandle(pos=(1, 0.5), center=(0, 0.5))
123    self.plot.addItem(self.roi)
124    self.roi.setZValue(10)
125    # Connect histogram plotting function to ROI change
126    self.roi.sigRegionChangeFinished.connect(self.plot_hist_roi)
127    # Histogram plot
128    self.hist_plot = self.win.addPlot(
129        row=4, col=1, title='ROI Histogram')
130    self.hist_plot.showGrid(x=True, y=True, alpha=0.8)
131
132    def set_frame(self, frame):
133        """
134        Set current frame and ROI in ViewBox
135        """
136        # Frame object
137        self.frame = frame
138        # DIP Image object
139        self.frame_dip = dip.Image(self.frame)
140        # Set frame to ImageItem
141        self.img.setImage(self.frame)
142        # Set ROI size according to frame
143        self.roi.setPos((0, 0))
144        self.roi.setSize((self.frame.shape[1], self.frame.shape[0]))
145        self.roi.maxBounds = QtCore.QRectF(
146            0, 0, self.frame.shape[1], self.frame.shape[0])
147
148    def plot_hist_roi(self):
149        """
150        Update ROI histogram according to ROI size

```

```

151         """
152         # Slicing compatible with dip Image
153         slice_tup = self.roi.getArraySlice(
154             self.frame, self.img, axes=(0, 1), returnSlice=False)
155         slice_tup = [
156             slice(tup[0], tup[1]-self.roi.pen.width()) for tup in
slice_tup[0]
157         ][::-1]
158         # print("Slice: {}".format(slice_tup))
159         # print("DIP Image: {}".format(self.frame_dip))
160         self.frame_dip_roi = self.frame_dip[slice_tup]
161         self.hist_plot.clear()
162         # self.frame_dip[slice_tup].Show()
163         frame_hist_y, frame_hist_x = dip.Histogram(self.frame_dip_roi)
164         # Plot Histogram
165         self.hist_plot.plot(
166             np.asarray(frame_hist_x[0]), np.asarray(frame_hist_y),
167             stepMode=False, fillLevel=0, fillBrush=(255, 165, 0, 150))
168
169     def on_mouse_move(self, event):
170         pos = event[0]
171         if self.plot.sceneBoundingRect().contains(pos):
172             coord = self.plot.vb.mapSceneToView(pos)
173             if (0 < coord.x() < self.frame_dip.Sizes()[0]) and (
174                 0 < coord.y() < self.frame_dip.Sizes()[1]):
175                 self.px_lbl.setText('Pixel Intensity: {}'.format(
176                     self.frame_dip[int(coord.x()), int(coord.y())])
177                 self.vline.setPos(coord.x())
178                 self.hline.setPos(coord.y())
179
180     def get_dip_slice(self):
181         return self.frame_dip_roi.Copy()
182
183     def show(self):
184         self.win.show()
185
186     def close(self):
187         self.win.hide()
188
189
190 class FlowchartPlotWidget(pg.GraphicsLayoutWidget):
191     """
192     PlotWidget displaying for dip.Image
193     """
194     def __init__(self, title=None):
195         super().__init__()
196         # View Box
197         self.view = pg.ViewBox(invertY=True, lockAspect=1)
198         # Image item
199         self.img = pg.ImageItem(border='w')
200         self.view.addItem(self.img)
201         # Plot for axes
202         self.plot = self.addPlot(
203             row=1, col=1, rowspan=1, title=title,
204             viewBox=self.view
205         )
206         self.plot.showGrid(x=True, y=True, alpha=0.8)
207
208     def setImage(self, dip_img):
209         """
210         Convenience function for displaying dip.Image inside plot
211         """

```

```
212         self.img.setImage(np.asarray(dip_img, dtype=np.uint32))
213
214     def setTitle(self, title):
215         """
216         Convenience function for setting plot title
217         """
218         self.plot.setTitle(title)
219
220
221 class MeasureRealtimeWindow:
222     def __init__(self):
223         """
224         Window for real-time measurement
225         """
226         # Graphics window
227         self.win = QtGui.QMainWindow()
228         self.win.setWindowTitle('Real-Time Object Measurement')
229         # Dock Area
230         dockArea = DockArea()
231         # Central widget
232         self.win.setCentralWidget(dockArea)
233
234         # Window Docks
235         origFramesDock = Dock('Original Frames', size=(100, 100))
236         # preprocFramesDock = Dock('Preprocessed Frames', size=(100,
237 100))
237         measureResultDocks = Dock('Measurement Result', size=(100, 60))
238         dockArea.addDock(measureResultDocks, 'left')
239         dockArea.addDock(origFramesDock, 'top', measureResultDocks)
240         # dockArea.addDock(preprocFramesDock, 'right', origFramesDock)
241
242         # Window size
243         self.win.resize(900, 600)
244
245         # Graphics View
246         viewOrig = pg.GraphicsView()
247         # viewPreproc = pg.GraphicsView()
248         origFramesDock.addWidget(viewOrig)
249         # preprocFramesDock.addWidget(viewPreproc)
250
251         """
252 Measurement result table
253 data2 = {
254     'FeretMin': [12.3, 17.5],
255     'FeretMax': [20.7, 21.9],
256 }
257 data2 = pd.DataFrame(data=data2)
258 w.setData(data2.values)
259 w.setHorizontalHeaderLabels(data2.columns)
260 w.appendRow([11, 11])
261         """
262
263         # PlotItems
264         plotOrig = pg.PlotItem()
265         # plotPreproc = pg.PlotItem()
266         plotOrig.invertY(True)
267         # plotPreproc.invertY(True)
268         plotOrig.setAspectLocked(True)
269         # plotPreproc.setAspectLocked(True)
270         viewOrig.setCentralItem(plotOrig)
271         # viewPreproc.setCentralItem(plotPreproc)
272
```

```

273     # ImageItems
274     self.imgOrig = pg.ImageItem(border='w')
275     # Auto Downsampling
276     # self.imgOrig.setAutoDownsample(True)
277     # self.imgPreproc = pg.ImageItem(border='w')
278     plotOrig.addItem(self.imgOrig)
279     # plotPreproc.addItem(self.imgPreproc)
280
281     # X-axis center line
282     self.xAxisCenter = None
283     self.vLine = plotOrig.addLine(x=0, pen={'color': 'g', 'width':
284     2})
285
286     # TableWidget for displaying measurement result
287     self.resultTable = pg.TableWidget(sortable=False)
288     # Formating for decimal numbers in table
289     self.resultTable.setFormat('%%.4f')
290     measureResultDocks.addWidget(self.resultTable)
291
292     # Column labels for result table
293     self.columnLabels = None
294     # Row labels for result table
295     self.rowLabels = []
296
297     # Number of measured objects
298     self.numMeasured = 0
299
300     def update_frame(self, frame):
301         """
302         Update frame inside ImageItem with np.array
303
304         :param frame: np.array frame
305         """
306         self.imgOrig.setImage(frame)
307         xAxisCenter = frame.shape[1] // 2
308         if xAxisCenter != self.xAxisCenter:
309             self.vLine.setValue(xAxisCenter)
310             self.xAxisCenter = xAxisCenter
311
312     def update_measurement_result(self, resultList):
313         """
314         Update table with measurement results. Column labels are set only
315         on
316         first call
317
318         :param resultList: list of dip.Measurement objects
319         """
320         # Number of measured objects
321         self.numMeasured += 1
322         # Measured features
323         features = [feat.name for feat in resultList[0].Features()]
324         # Outer and inner diameter mark
325         whichDiam = ['out', 'in'] * math.ceil(len(resultList) / 2)
326
327         # Set column labels if not already set
328         if self.columnLabels is None:
329             # Number of values for each measured feature
330             num_val = [feat.numberValues for feat in
331             resultList[0].Features()]
332             feat_num = [[feat] * num for feat, num in zip(features,
333             num_val)]
334             # Flatten list of features

```

```

331         feat_num = [feat for sublist in feat_num for feat in sublist]
332         # List of feature values
333         values = [
334             ' '.join((val.name, ' '.join([' ', str(val.units), ''])))
335             for val in resultList[0].Values()
336         ]
337         self.columnLabels = [
338             ' '.join((feat, val)) for feat, val in zip(feat_num,
values)
339         ]
340         #
self.resultTable.setHorizontalHeaderLabels(self.columnLabels)
341
342         # Append rows at the end of table
343         for dipMsr, i in zip(resultList, range(len(resultList))):
344             # For every dip.Measurement object
345             for obj in dipMsr.Objects():
346                 # For every measured object inside dip.Measurement object
347                 combinedList = [dipMsr[feat][obj] for feat in features]
348                 # Flatten combined list
349                 combinedList = [
350                     comb for sublist in combinedList for comb in sublist
351                 ]
352                 self.resultTable.appendRow(combinedList)
353                 # Append row Labels
354                 self.rowLabels.append(
355                     ' '.join((str(self.numMeasured), whichDiam[i]))
356                 )
357
358             # Set Row labels in table
359             self.resultTable.setVerticalHeaderLabels(self.rowLabels)
360             self.resultTable.verticalHeadersSet = True
361             self.resultTable.setHorizontalHeaderLabels(self.columnLabels)
362             self.resultTable.horizontalHeadersSet = True
363             self.resultTable.resizeColumnsToContents()
364
365         def setTableColumnLabels(self, headerLabels):
366             """
367             Set result table column labels with list of strings
368
369             :param headerLabels: list of column headers
370             """
371             self.resultTable.setHorizontalHeaderLabels(headerLabels)
372
373         def appendResultRow(self, measurementResult):
374             """
375             Append measurement results at the end of table
376
377             :param measurementResult: list or np.ndarray of measurement
results
378             """
379             self.resultTable.appendRow(measurementResult)
380
381
382         # def update_preproc_frame(self, frame):
383         #     self.imgPreproc.setImage(frame)
384
385         def show(self):
386             self.win.show()
387
388         def close(self):
389             self.win.hide()

```

```

390
391
392 class SignalChange(QtCore.QObject):
393     """
394     Signal for communication between threads
395     """
396     updateGraphics = QtCore.pyqtSignal()
397     updateFPS = QtCore.pyqtSignal()
398

```

1.8. Skripta „nodes.py“

```

1 import pyqtgraph as pg
2 from pyqtgraph.flowchart import Node
3 import pyqtgraph.flowchart.library as fclib
4 from pyqtgraph.flowchart.library.common import CtrlNode
5 from pyqtgraph.Qt import QtCore, QtGui
6
7 # Numpy
8 import numpy as np
9
10 # PyDIP
11 import PyDIP as dip
12
13 # Operators
14 import operator
15
16
17 class FlowchartPlotNode(Node):
18     """
19     Node for displaying dip.Image in FlowchartPlotWidget
20     """
21     nodeName = 'FlowchartPlot'
22
23     def __init__(self, name):
24         super().__init__(name=name, terminals={
25             'dipImgIn': {'io': 'in', 'optional': True}
26         })
27         self.fc_plot_widget = None
28
29     def set_fcPlotWidget(self, fc_plot_widget):
30         self.fc_plot_widget = fc_plot_widget
31
32     def process(self, dipImgIn=None, display=True):
33         """
34         Node process function. If display=False, skip displaying img
35         """
36         if display and self.fc_plot_widget is not None:
37             if dipImgIn is None:
38                 # Empty image
39                 self.fc_plot_widget.setImage(np.zeros((1, 1)))
40             else:
41                 self.fc_plot_widget.setImage(dipImgIn)
42                 # Plot title from connected node
43                 title = ' '.join([n.name() for n in
self.dependentNodes()])
44                 self.fc_plot_widget.setTitle(title)
45
46
47 class GaussianConvolutionNode(CtrlNode):
48     """

```



```

49     Node with control widget for convolution with Gaussian kernel
50     """
51     nodeName = 'GaussianConvolution'
52     uiTemplate = [
53         ('sigmas', 'spin', {'value': 1.0, 'step': 0.1, 'bounds': [0.,
10.]}))
54     ]
55
56     def __init__(self, name):
57         super().__init__(name=name, terminals={
58             'dipImgIn': {'io': 'in'},
59             'smoothOut': {'io': 'out'}}
60         })
61
62     def process(self, dipImgIn, display=True):
63         """
64         Node process function. If display=False, no effect
65         """
66         sigmas = self.ctrls['sigmas'].value()
67         return {'smoothOut': dip.Gauss(dipImgIn, [sigmas])}
68
69
70 class ThresholdNode(CtrlNode):
71     """
72     Node with control widget for thresholding with different methods
73     """
74     nodeName = 'Threshold'
75     uiTemplate = [
76         ('method', 'combo', {'values': [
77             'background', 'volume', 'fixed', 'isodata', 'otsu',
'minerror',
78             'triangle',
79         ]}),
80         ('parameter', 'spin', {
81             'value': 8.0, 'step': 0.1, 'bounds': [0., 255.]})
82     ]
83
84     def __init__(self, name):
85         super().__init__(name=name, terminals={
86             'dipImgIn': {'io': 'in'},
87             'thrImgOut': {'io': 'out'},
88             'thrValOut': {'io': 'out'},
89         })
90
91     def process(self, dipImgIn, display=True):
92         """
93         Node process function. If display=False, no effect
94         """
95         method = self.stateGroup.state()['method']
96         parameter = self.ctrls['parameter'].value()
97         thr = dip.Threshold(dipImgIn, method=method,
parameter=parameter)
98         return {'thrImgOut': thr[0], 'thrValOut': thr[1]}
99
100
101 class RangeThresholdNode(CtrlNode):
102     """
103     Node with control widget for range thresholding
104     """
105     nodeName = 'RangeThreshold'
106     uiTemplate = [
107         ('lowerBound', 'spin', {

```

```

108         'value': 30, 'step': 1, 'bounds': [0, 255]
109     }),
110     ('upperBound', 'spin', {
111         'value': 100, 'step': 1, 'bounds': [0, 255]
112     }),
113     ('foreground', 'spin', {
114         'value': 1, 'step': 1, 'bounds': [0, 1]
115     }),
116     ('background', 'spin', {
117         'value': 0, 'step': 1, 'bounds': [0, 1]
118     }),
119 ]
120
121 def __init__(self, name):
122     super().__init__(name=name, terminals={
123         'dipImgIn': {'io': 'in'},
124         'dipImgOut': {'io': 'out'},
125     })
126
127 def process(self, dipImgIn, display=True):
128     """
129     Node process function. If display=False, no effect
130     """
131     lowerBound = self.ctrls['lowerBound'].value()
132     upperBound = self.ctrls['upperBound'].value()
133     foreground = self.ctrls['foreground'].value()
134     background = self.ctrls['background'].value()
135     thr = dip.RangeThreshold(
136         dipImgIn, lowerBound=lowerBound, upperBound=upperBound,
137         foreground=foreground, background=background
138     )
139     return {'dipImgOut': thr}
140
141
142 class OpeningNode(CtrlNode):
143     """
144     Node with control widget for opening with different shape and size
145     """
146     nodeName = 'Opening'
147     uiTemplate = [
148         ('shape', 'combo', {'values': [
149             'elliptic', 'diamond', 'rectangular', 'octagonal',
'parabolic'
150         ]}),
151         ('size', 'spin', {
152             'value': 11, 'step': 2, 'bounds': [1, 99]})
153     ]
154
155 def __init__(self, name):
156     super().__init__(name=name, terminals={
157         'dipImgIn': {'io': 'in'},
158         'dipImgOut': {'io': 'out'},
159     })
160
161 def process(self, dipImgIn, display=True):
162     """
163     Node process function. If display=False, no effect
164     """
165     shape = self.stateGroup.state()['shape']
166     size = self.ctrls['size'].value()
167     struct_element = dip.SE(size, shape)
168     img = dip.Opening(dipImgIn, se=struct_element)

```

```
169         return {'dipImgOut': img}
170
171
172 class ClosingNode(CtrlNode):
173     """
174     Node with control widget for Closing with different shape and size
175     """
176     nodeName = 'Closing'
177     uiTemplate = [
178         ('shape', 'combo', {'values': [
179             'elliptic', 'diamond', 'rectangular', 'octagonal',
'parabolic'
180         ]}),
181         ('size', 'spin', {
182             'value': 11, 'step': 2, 'bounds': [1, 99]})
183     ]
184
185     def __init__(self, name):
186         super().__init__(name=name, terminals={
187             'dipImgIn': {'io': 'in'},
188             'dipImgOut': {'io': 'out'},
189         })
190
191     def process(self, dipImgIn, display=True):
192         """
193         Node process function. If display=False, no effect
194         """
195         shape = self.stateGroup.state()['shape']
196         size = self.ctrls['size'].value()
197         struct_element = dip.SE(size, shape)
198         img = dip.Closing(dipImgIn, se=struct_element)
199         return {'dipImgOut': img}
200
201
202 class DilationNode(CtrlNode):
203     """
204     Node with control widget for Dilation with different shape and size
205     """
206     nodeName = 'Dilation'
207     uiTemplate = [
208         ('shape', 'combo', {'values': [
209             'elliptic', 'diamond', 'rectangular', 'octagonal',
'parabolic'
210         ]}),
211         ('size', 'spin', {
212             'value': 11, 'step': 2, 'bounds': [1, 99]
213         }),
214     ]
215
216     def __init__(self, name):
217         super().__init__(name=name, terminals={
218             'dipImgIn': {'io': 'in'},
219             'dipImgOut': {'io': 'out'},
220         })
221
222     def process(self, dipImgIn, display=True):
223         """
224         Node process function. If display=False, no effect
225         """
226         shape = self.stateGroup.state()['shape']
227         size = self.ctrls['size'].value()
228         struct_element = dip.SE(size, shape)
```

```

229         img = dip.Dilation(dipImgIn, se=struct_element)
230         return {'dipImgOut': img}
231
232
233 class ErosionNode(CtrlNode):
234     """
235     Node with control widget for Erosion with different shape and size
236     """
237     nodeName = 'Erosion'
238     uiTemplate = [
239         ('shape', 'combo', {'values': [
240             'elliptic', 'diamond', 'rectangular', 'octagonal',
'parabolic'
241         ]}),
242         ('size', 'spin', {
243             'value': 11, 'step': 2, 'bounds': [1, 99]
244         }),
245     ]
246
247     def __init__(self, name):
248         super().__init__(name=name, terminals={
249             'dipImgIn': {'io': 'in'},
250             'dipImgOut': {'io': 'out'},
251         })
252
253     def process(self, dipImgIn, display=True):
254         """
255         Node process function. If display=False, no effect
256         """
257         shape = self.stateGroup.state()['shape']
258         size = self.ctrls['size'].value()
259         struct_element = dip.SE(size, shape)
260         img = dip.Erosion(dipImgIn, se=struct_element)
261         return {'dipImgOut': img}
262
263
264 class EdgeObjectsRemoveNode(CtrlNode):
265     """
266     Node with control widget for removing edge objects from binary image
with
267     different connectivity
268     """
269     nodeName = 'EdgeObjectsRemove'
270     uiTemplate = [
271         ('connectivity', 'combo', {'values': [
272             '8-Connectivity', '4-Connectivity'
273         ]}),
274     ]
275
276     def __init__(self, name):
277         super().__init__(name=name, terminals={
278             'dipImgIn': {'io': 'in'},
279             'dipImgOut': {'io': 'out'},
280         })
281
282     def process(self, dipImgIn, display=True):
283         """
284         Node process function. If display=False, no effect
285         """
286         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
287             connectivity = 2
288         else:

```

```
289         connectivity = 1
290         img = dip.EdgeObjectsRemove(dipImgIn, connectivity=connectivity)
291         return {'dipImgOut': img}
292
293
294 class ConvertNode(CtrlNode):
295     """
296     Node with control widget for converting image data type
297     """
298     nodeName = 'Convert'
299     uiTemplate = [
300         ('DataType', 'combo', {'values': [
301             'BIN', 'UINT8', 'UINT16', 'UINT32', 'SINT8', 'SINT16',
'SINT32',
302             'SFLOAT', 'DFLOAT', 'SCOMPLEX', 'DCOMPLEX'
303         ]}),
304     ]
305
306     def __init__(self, name):
307         super().__init__(name=name, terminals={
308             'dipImgIn': {'io': 'in'},
309             'dipImgOut': {'io': 'out'},
310         })
311
312     def process(self, dipImgIn, display=True):
313         """
314         Node process function. If display=False, no effect
315         """
316         dtype = self.stateGroup.state()['DataType']
317         img = dip.Convert(dipImgIn, dtype)
318         return {'dipImgOut': img}
319
320
321 class InvertNode(Node):
322     """
323     Node for inverting pixel intensities
324     """
325     nodeName = 'Invert'
326
327     def __init__(self, name):
328         super().__init__(name=name, terminals={
329             'dipImgIn': {'io': 'in'},
330             'dipImgOut': {'io': 'out'},
331         })
332
333     def process(self, dipImgIn, display=True):
334         """
335         Node process function. If display=False, no effect
336         """
337         img = dip.Invert(dipImgIn)
338         return {'dipImgOut': img}
339
340
341 class CannyNode(CtrlNode):
342     """
343     Node with control widget for Canny edge detector
344     """
345     nodeName = 'Canny'
346     uiTemplate = [
347         ('sigmas', 'spin', {'value': 1.0, 'step': 0.1, 'bounds': [0.,
10.]}),
```

```

348         ('lower', 'spin', {'value': 0.5, 'step': 0.01, 'bounds': [0.,
1.]}),
349         ('upper', 'spin', {'value': 0.9, 'step': 0.01, 'bounds': [0.,
1.]}),
350     ]
351
352     def __init__(self, name):
353         super().__init__(name=name, terminals={
354             'dipImgIn': {'io': 'in'},
355             'dipImgOut': {'io': 'out'},
356         })
357
358     def process(self, dipImgIn, display=True):
359         """
360         Node process function. If display=False, no effect
361         """
362         sigmas = [self.ctrls['sigmas'].value()]
363         lower = self.ctrls['lower'].value()
364         upper = self.ctrls['upper'].value()
365         img = dip.Canny(dipImgIn, sigmas, lower, upper)
366         return {'dipImgOut': img}
367
368
369 class GradientNode(CtrlNode):
370     """
371     Node with control widget for calculating image Gradient
372     """
373     nodeName = 'Gradient'
374     uiTemplate = [
375         ('sigmas', 'spin', {'value': 1.0, 'step': 0.1, 'bounds': [0.,
10.]}),
376     ]
377
378     def __init__(self, name):
379         super().__init__(name=name, terminals={
380             'dipImgIn': {'io': 'in'},
381             'dipImgOut': {'io': 'out'},
382         })
383
384     def process(self, dipImgIn, display=True):
385         """
386         Node process function. If display=False, no effect
387         """
388         sigmas = [self.ctrls['sigmas'].value()]
389         img = dip.Gradient(dipImgIn, sigmas)
390         return {'dipImgOut': img}
391
392
393 class GradientMagnitudeNode(CtrlNode):
394     """
395     Node with control widget for computing Gradient Magnitude
396     """
397     nodeName = 'GradientMagnitude'
398     uiTemplate = [
399         ('sigmas', 'spin', {'value': 1.0, 'step': 0.1, 'bounds': [0.,
10.]}),
400     ]
401
402     def __init__(self, name):
403         super().__init__(name=name, terminals={
404             'dipImgIn': {'io': 'in'},
405             'dipImgOut': {'io': 'out'},

```

```

406         })
407
408     def process(self, dipImgIn, display=True):
409         """
410         Node process function. If display=False, no effect
411         """
412         sigmas = [self.ctrls['sigmas'].value()]
413         img = dip.GradientMagnitude(dipImgIn, sigmas)
414         return {'dipImgOut': img}
415
416
417 class GradientDirectionNode(CtrlNode):
418     """
419     Node with control widget for computing Gradient Direction
420     """
421     nodeName = 'GradientDirection'
422     uiTemplate = [
423         ('sigmas', 'spin', {'value': 1.0, 'step': 0.1, 'bounds': [0.,
10.]}),
424     ]
425
426     def __init__(self, name):
427         super().__init__(name=name, terminals={
428             'dipImgIn': {'io': 'in'},
429             'dipImgOut': {'io': 'out'},
430         })
431
432     def process(self, dipImgIn, display=True):
433         """
434         Node process function. If display=False, no effect
435         """
436         sigmas = [self.ctrls['sigmas'].value()]
437         img = dip.GradientDirection(dipImgIn, sigmas)
438         return {'dipImgOut': img}
439
440
441 class WatershedNode(CtrlNode):
442     """
443     Node with control widget for Watershed
444     """
445     nodeName = 'Watershed'
446     uiTemplate = [
447         ('connectivity', 'combo', {'values': [
448             '8-Connectivity', '4-Connectivity'
449         ]}),
450         ('maxDepth', 'spin', {
451             'value': 1., 'step': 0.01, 'bounds': [0., 255.]
452         }),
453         ('maxSize', 'spin', {
454             'value': 0, 'step': 1, 'bounds': [0, None]
455         }),
456         ('outType', 'combo', {'values': [
457             'binary', 'labels'
458         ]}),
459         ('sortOrder', 'combo', {'values': [
460             'low first', 'high first'
461         ]}),
462         ('algorithm', 'combo', {'values': [
463             'correct', 'fast'
464         ]}),
465     ]
466

```

```
467     def __init__(self, name):
468         super().__init__(name=name, terminals={
469             'dipImgIn': {'io': 'in'},
470             'dipMaskIn': {'io': 'in', 'optional': True},
471             'dipImgOut': {'io': 'out'},
472         })
473
474     def process(self, dipImgIn, dipMaskIn, display=True):
475         """
476         Node process function. If display=False, no effect
477         """
478         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
479             connectivity = 2
480         else:
481             connectivity = 1
482         maxDepth = self.ctrls['maxDepth'].value()
483         maxSize = int(self.ctrls['maxSize'].value())
484         flags = {
485             self.stateGroup.state()['outType'],
486             self.stateGroup.state()['sortOrder'],
487             self.stateGroup.state()['algorithm']
488         }
489         if dipMaskIn is None:
490             img = dip.Watershed(
491                 dipImgIn, connectivity=connectivity,
492                 maxDepth=maxDepth, maxSize=maxSize, flags=flags
493             )
494         else:
495             img = dip.Watershed(
496                 dipImgIn, mask=dipMaskIn, connectivity=connectivity,
497                 maxDepth=maxDepth, maxSize=maxSize, flags=flags
498             )
499         return {'dipImgOut': img}
500
501
502     class SeededWatershedNode(CtrlNode):
503         """
504         Node with control widget for Watershed starting at seeds
505         """
506         nodeName = 'SeededWatershed'
507         uiTemplate = [
508             ('connectivity', 'combo', {'values': [
509                 '8-Connectivity', '4-Connectivity'
510             ]}),
511             ('maxDepth', 'spin', {
512                 'value': 1., 'step': 0.01, 'bounds': [0., 255.]
513             }),
514             ('maxSize', 'spin', {
515                 'value': 0, 'step': 1, 'bounds': [0, None]
516             }),
517             ('outType', 'combo', {'values': [
518                 'labels', 'binary'
519             ]}),
520             ('sortOrder', 'combo', {'values': [
521                 'low first', 'high first'
522             ]}),
523             ('no gaps', 'check', {'value': False}),
524             ('uphill only', 'check', {'value': False}),
525         ]
526
527     def __init__(self, name):
528         super().__init__(name=name, terminals={
```



```

529         'dipImgIn': {'io': 'in'},
530         'dipSeedsIn': {'io': 'in'},
531         'dipMaskIn': {'io': 'in', 'optional': True},
532         'dipImgOut': {'io': 'out'},
533     })
534
535     def process(self, dipImgIn, dipSeedsIn, dipMaskIn=None,
display=True):
536         """
537         Node process function. If display=False, no effect
538         """
539         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
540             connectivity = 2
541         else:
542             connectivity = 1
543         maxDepth = self.ctrls['maxDepth'].value()
544         maxSize = int(self.ctrls['maxSize'].value())
545         flags = {
546             self.stateGroup.state()['outType'],
547             self.stateGroup.state()['sortOrder'],
548         }
549         if self.ctrls['no gaps'].isChecked():
550             flags.add('no gaps')
551         if self.ctrls['uphill only'].isChecked():
552             flags.add('uphill only')
553         if dipMaskIn is None:
554             img = dip.SeededWatershed(
555                 dipImgIn, seeds=dipSeedsIn, connectivity=connectivity,
556                 maxDepth=maxDepth, maxSize=maxSize, flags=flags
557             )
558         else:
559             img = dip.SeededWatershed(
560                 dipImgIn, seeds=dipSeedsIn, mask=dipMaskIn,
561                 connectivity=connectivity, maxDepth=maxDepth,
maxSize=maxSize,
562                 flags=flags
563             )
564         return {'dipImgOut': img}
565
566
567 class MinimaNode(CtrlNode):
568     """
569     Node with control widget for local minima
570     """
571     nodeName = 'Minima'
572     uiTemplate = [
573         ('connectivity', 'combo', {'values': [
574             '8-Connectivity', '4-Connectivity'
575         ]}),
576         ('outType', 'combo', {'values': [
577             'binary', 'labels'
578         ]}),
579     ]
580
581     def __init__(self, name):
582         super().__init__(name=name, terminals={
583             'dipImgIn': {'io': 'in'},
584             'dipImgOut': {'io': 'out'},
585         })
586
587     def process(self, dipImgIn, display=True):
588         """

```

```

589     Node process function. If display=False, no effect
590     """
591     if self.stateGroup.state()['connectivity'] == '8-Connectivity':
592         connectivity = 2
593     else:
594         connectivity = 1
595     output = self.stateGroup.state()['outType']
596     img = dip.Minima(dipImgIn, connectivity=connectivity,
output=output)
597     return {'dipImgOut': img}
598
599
600 class MaximaNode(CtrlNode):
601     """
602     Node with control widget for local minima
603     """
604     nodeName = 'Maxima'
605     uiTemplate = [
606         ('connectivity', 'combo', {'values': [
607             '8-Connectivity', '4-Connectivity'
608         ]}),
609         ('outType', 'combo', {'values': [
610             'binary', 'labels'
611         ]}),
612     ]
613
614     def __init__(self, name):
615         super().__init__(name=name, terminals={
616             'dipImgIn': {'io': 'in'},
617             'dipImgOut': {'io': 'out'},
618         })
619
620     def process(self, dipImgIn, display=True):
621         """
622         Node process function. If display=False, no effect
623         """
624         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
625             connectivity = 2
626         else:
627             connectivity = 1
628         output = self.stateGroup.state()['outType']
629         img = dip.Minima(dipImgIn, connectivity=connectivity,
output=output)
630         return {'dipImgOut': img}
631
632
633 class CreateMaskNode(CtrlNode):
634     """
635     Node with control widget for creating mask
636     """
637     nodeName = 'CreateMask'
638     uiTemplate = [
639         ('relation', 'combo', {'values': [
640             '<=', '>=', '<', '>', '==', '!='
641         ]}),
642         ('intensity', 'spin', {
643             'value': 1, 'step': 1, 'bounds': [0, 255]})
644     ]
645
646     def __init__(self, name):
647         super().__init__(name=name, terminals={
648             'dipImgIn': {'io': 'in'},

```

```

649         'dipImgOut': {'io': 'out'},
650     })
651     self.op = {
652         '<=': operator.le, '>=': operator.ge, '<': operator.lt,
653         '>': operator.gt, '==': operator.eq, '!=': operator.ne
654     }
655
656     def process(self, dipImgIn, display=True):
657         """
658         Node process function. If display=False, no effect
659         """
660         rel = self.stateGroup.state()['relation']
661         intensity = self.ctrls['intensity'].value()
662         img = self.op[rel](dipImgIn, intensity)
663         return {'dipImgOut': img}
664
665
666     class ApplyMaskNode(Node):
667         """
668         Node for applying mask
669         """
670         nodeName = 'ApplyMask'
671
672         def __init__(self, name):
673             super().__init__(name=name, terminals={
674                 'dipImgIn': {'io': 'in'},
675                 'dipMaskIn': {'io': 'in'},
676                 'dipImgOut': {'io': 'out'},
677             })
678
679         def process(self, dipImgIn, dipMaskIn, display=True):
680             """
681             Node process function. If display=False, no effect
682             """
683             img = dipImgIn * dipMaskIn
684             return {'dipImgOut': img}
685
686
687     class FillNode(CtrlNode):
688         """
689         Node for filling pixels under mask (optional) with intensity
690         """
691         nodeName = 'Fill'
692         uiTemplate = [
693             ('intensity', 'spin', {
694                 'value': 0, 'step': 1, 'bounds': [0, 255]})
695         ]
696
697         def __init__(self, name):
698             super().__init__(name=name, terminals={
699                 'dipImgIn': {'io': 'in'},
700                 'dipMaskIn': {'io': 'in', 'optional': True},
701                 'dipImgOut': {'io': 'out'},
702             })
703
704         def process(self, dipImgIn, dipMaskIn, display=True):
705             """
706             Node process function. If display=False, no effect
707             """
708             intensity = self.ctrls['intensity'].value()
709             if dipMaskIn is None:
710                 img = dipImgIn.Copy().Fill(intensity)

```

```

711         else:
712             img = dipImgIn.Copy()
713             img[dipMaskIn] = intensity
714             return {'dipImgOut': img}
715
716
717 class BinaryPropagationNode(CtrlNode):
718     """
719     Node with control widget for BinaryPropagation
720     """
721     nodeName = 'BinaryPropagation'
722     uiTemplate = [
723         ('connectivity', 'combo', {'values': [
724             '8-Connectivity', '4-Connectivity'
725         ]}),
726         ('iterations', 'spin', {
727             'value': 0, 'step': 1, 'bounds': [0, None]
728         }),
729         ('edgeCondition', 'combo', {'values': [
730             'background', 'object'
731         ]}),
732     ]
733
734     def __init__(self, name):
735         super().__init__(name=name, terminals={
736             'dipImgIn': {'io': 'in'},
737             'dipMaskIn': {'io': 'in', 'optional': True},
738             'dipImgOut': {'io': 'out'},
739         })
740
741     def process(self, dipImgIn, dipMaskIn, display=True):
742         """
743         Node process function. If display=False, no effect
744         """
745         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
746             connectivity = 2
747         else:
748             connectivity = 1
749         iterations = int(self.ctrls['iterations'].value())
750         edgeCondition = self.stateGroup.state()['edgeCondition']
751         if dipMaskIn is None:
752             img = dip.BinaryPropagation(
753                 dipImgIn, connectivity=connectivity,
754                 iterations=iterations, edgeCondition=edgeCondition
755             )
756         else:
757             img = dip.BinaryPropagation(
758                 dipImgIn, dipMaskIn, connectivity=connectivity,
759                 iterations=iterations, edgeCondition=edgeCondition
760             )
761         return {'dipImgOut': img}
762
763
764 class FillHolesNode(CtrlNode):
765     """
766     Node with control widget for FillHoles
767     """
768     nodeName = 'FillHoles'
769     uiTemplate = [
770         ('connectivity', 'combo', {'values': [
771             '8-Connectivity', '4-Connectivity'
772         ]}),

```

```

773     ]
774
775     def __init__(self, name):
776         super().__init__(name=name, terminals={
777             'dipImgIn': {'io': 'in'},
778             'dipImgOut': {'io': 'out'},
779         })
780
781     def process(self, dipImgIn, display=True):
782         """
783         Node process function. If display=False, no effect
784         """
785         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
786             connectivity = 2
787         else:
788             connectivity = 1
789         img = dip.FillHoles(dipImgIn, connectivity=connectivity)
790         return {'dipImgOut': img}
791
792
793 class LabelNode(CtrlNode):
794     """
795     Node with control widget for Labeling connected regions in binary
image
796     """
797     nodeName = 'Label'
798     uiTemplate = [
799         ('connectivity', 'combo', {'values': [
800             '8-Connectivity', '4-Connectivity'
801         ]}),
802         ('minSize', 'spin', {
803             'value': 0, 'step': 1, 'bounds': [0, None]
804         }),
805         ('maxSize', 'spin', {
806             'value': 0, 'step': 1, 'bounds': [0, None]
807         }),
808     ]
809
810     def __init__(self, name):
811         super().__init__(name=name, terminals={
812             'dipImgIn': {'io': 'in'},
813             'dipImgOut': {'io': 'out'},
814         })
815
816     def process(self, dipImgIn, display=True):
817         """
818         Node process function. If display=False, no effect
819         """
820         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
821             connectivity = 2
822         else:
823             connectivity = 1
824         minSize = int(self.ctrls['minSize'].value())
825         maxSize = int(self.ctrls['maxSize'].value())
826         img = dip.Label(
827             dipImgIn, connectivity=connectivity,
828             minSize=minSize, maxSize=maxSize
829         )
830         return {'dipImgOut': img}
831
832
833 class MeasureNode(CtrlNode):

```

```

834     """
835     Node with control widget for Measuring one or more features
836     """
837     nodeName = 'Measure'
838     uiTemplate = [
839         ('connectivity', 'combo', {'values': [
840             '8-Connectivity', '4-Connectivity'
841         ]}),
842         ('Perimeter', 'check', {'value': False}),
843         ('Feret', 'check', {'value': True}),
844         ('CartesianBox', 'check', {'value': False}),
845         ('Minimum', 'check', {'value': False}),
846         ('Maximum', 'check', {'value': False}),
847         ('SolidArea', 'check', {'value': False}),
848         ('ConvexArea', 'check', {'value': False}),
849         ('ConvexPerimeter', 'check', {'value': False}),
850         ('AspectRatioFeret', 'check', {'value': False}),
851         ('Radius', 'check', {'value': False}),
852         ('P2A', 'check', {'value': False}),
853         ('Roundness', 'check', {'value': False}),
854         ('Circularity', 'check', {'value': False}),
855         ('PodczeckShapes', 'check', {'value': False}),
856         ('Solidity', 'check', {'value': False}),
857         ('Convexity', 'check', {'value': False}),
858         ('EllipseVariance', 'check', {'value': False}),
859         ('Eccentricity', 'check', {'value': False}),
860         ('Center', 'check', {'value': False}),
861     ]
862
863     def __init__(self, name):
864         super().__init__(name=name, terminals={
865             'dipLblIn': {'io': 'in'},
866             'dipGreyIn': {'io': 'in'},
867             'dipMsrOut': {'io': 'out'},
868         })
869
870     def process(self, dipLblIn, dipGreyIn, display=True):
871         """
872         Node process function. If display=False, no effect
873         """
874         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
875             connectivity = 2
876         else:
877             connectivity = 1
878         features = []
879         for name, val in self.ctrls.items():
880             if name != 'connectivity' and val.isChecked():
881                 features.append(name)
882         msr = dip.MeasurementTool.Measure(
883             dipLblIn, grey=dipGreyIn, features=features,
884             connectivity=connectivity
885         )
886         return {'dipMsrOut': msr}
887
888
889     class MeasurementDisplayNode(Node):
890         """
891         Node for displaying dip.Measurement in FlowchartPlotWidget
892         """
893         nodeName = 'MeasurementDisplay'
894
895         def __init__(self, name):

```

```

896     super().__init__(name=name, terminals={
897         'dipMsrIn': {'io': 'in'}}
898     })
899     self.fc_display_widget = None
900
901     def setDisplayWidget(self, display_widget):
902         self.fc_display_widget = display_widget
903
904     def process(self, dipMsrIn, display=True):
905         """
906         Node process function. If display=False, skip displaying text
907         """
908         if display and self.fc_display_widget is not None:
909             if dipMsrIn is None:
910                 # Empty text
911                 self.fc_display_widget.setText('')
912             else:
913                 # Measurement result as string
914                 disp_text = str(dipMsrIn).replace("\n", "<br>")
915                 disp_text = disp_text.replace(" ", "&nbsp;")
916                 disp_text = ''.join((
917                     '<tt style="font-size:8pt">', disp_text, '</tt>'))
918                 self.fc_display_widget.setText(disp_text)
919
920
921 class WorkingDistanceCorrectionNode(CtrlNode):
922     """
923     Node with control widget for correcting object distance from lens
using
924     thin lens equation
925     """
926     nodeName = 'WorkingDistanceCorrection'
927     uiTemplate = [
928         ('feature', 'combo', {'values': [
929             'Feret Min', 'Feret Max'
930         ]}),
931         ('standard 1 height', 'spin', {
932             'value': 0, 'step': 0.001, 'bounds': [0, None], 'suffix':
'mm',
933             'siPrefix': True
934         }),
935         ('standard 1 measure', 'spin', {
936             'value': 0, 'step': 0.001, 'bounds': [0, None], 'suffix':
'mm',
937             'siPrefix': True
938         }),
939         ('standard 2 height', 'spin', {
940             'value': 0, 'step': 0.001, 'bounds': [0, None], 'suffix':
'mm',
941             'siPrefix': True
942         }),
943         ('standard 2 measure', 'spin', {
944             'value': 0, 'step': 0.001, 'bounds': [0, None], 'suffix':
'mm',
945             'siPrefix': True
946         }),
947     ]
948
949     def __init__(self, name):
950         super().__init__(name=name, terminals={
951             'dipMsrIn': {'io': 'in'},
952             'CalibConstOut': {'io': 'out'},

```

```

953     })
954
955     def process(self, dipMsrIn, display=True):
956         """
957         Node process function. If display=False, no effect
958         """
959         if self.stateGroup.state()['feature'] == 'Ferret Min':
960             # Measured image size
961             img_size_1, img_size_2 = (dipMsrIn['Ferret'][1][1],
962                                     dipMsrIn['Ferret'][2][1])
963         else:
964             # Measured image size
965             img_size_1, img_size_2 = (dipMsrIn['Ferret'][1][0],
966                                     dipMsrIn['Ferret'][2][0])
967         # print("Image sizes: {} - {}".format(img_size_1, img_size_2))
968         # Standard height difference
969         height_1 = self.ctrls['standard 1 height'].value()
970         height_2 = self.ctrls['standard 2 height'].value()
971         height_diff_12 = height_1 - height_2
972         # print("Height difference: {}".format(height_diff_12))
973         obj_size_1 = self.ctrls['standard 1 measure'].value()
974         obj_size_2 = self.ctrls['standard 2 measure'].value()
975         # print("Object sizes: {} - {}".format(obj_size_1, obj_size_2))
976         # mm/px ratio
977         mm_px_1 = obj_size_1 / img_size_1
978         mm_px_2 = obj_size_2 / img_size_2
979         # print("mm/px ratios: {} - {}".format(mm_px_1, mm_px_2))
980         # Working distance
981         work_dist_1 = height_diff_12 / (mm_px_2 / mm_px_1 - 1.)
982         # print("Working distance 1: {}".format(work_dist_1))
983         return {'CalibConstOut': {'std_height': height_1,
984                                   'std_work_dist': work_dist_1,
985                                   'std_mm_px': mm_px_1}}
986
987
988     class CalibDisplayNode(Node):
989         """
990         Node for displaying Calibration equation in FlowchartPlotWidget
991         """
992         nodeName = 'CalibDisplay'
993
994         def __init__(self, name):
995             super().__init__(name=name, terminals={
996                 'CalibConstIn': {'io': 'in'}
997             })
998             self.fc_display_widget = None
999
1000         def setDisplayWidget(self, display_widget):
1001             self.fc_display_widget = display_widget
1002
1003         def process(self, CalibConstIn, display=True):
1004             """
1005             Node process function. If display=False, skip displaying text
1006             """
1007             if display and self.fc_display_widget is not None:
1008                 if CalibConstIn is None:
1009                     # Empty text
1010                     self.fc_display_widget.setText('')
1011                 else:
1012                     # Measurement result as string
1013                     disp_text = (
1014                         '<sup>mm</sup>&frac{1}{<sub>px</sub>} = (1 + '

```



```

1015         '<sup>{} - object_height</sup>&fracl;<sub>{}</sub>)'
1016         '&times; {}'.format(
1017             CalibConstIn['std_height'],
CalibConstIn['std_work_dist'],
1018             CalibConstIn['std_mm_px']
1019         )
1020         # print(disp_text)
1021         disp_text = ''.join((
1022             '<tt style=\"font-size:18pt\">', disp_text,
'</tt>'))
1023             self.fc_display_widget.setText(disp_text)
1024
1025
1026 class SegmentORingNode(CtrlNode):
1027     """
1028     Node with control widget for Segmenting O-ring outer and inner
diameter
1029     """
1030     nodeName = 'SegmentORing'
1031     uiTemplate = [
1032         ('connectivity', 'combo', {'values': [
1033             '8-Connectivity', '4-Connectivity'
1034         ]}),
1035     ]
1036
1037     def __init__(self, name):
1038         super().__init__(name=name, terminals={
1039             'dipImgIn': {'io': 'in'},
1040             'dipOuterOut': {'io': 'out'},
1041             'dipInnerOut': {'io': 'out'},
1042         })
1043
1044     def process(self, dipImgIn, display=True):
1045         """
1046         Node process function. If display=False, no effect
1047         """
1048         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
1049             connectivity = 2
1050         else:
1051             connectivity = 1
1052         outer = dip.FillHoles(dipImgIn, connectivity=connectivity)
1053         inner = dip.EdgeObjectsRemove(
1054             dip.Invert(dipImgIn), connectivity=connectivity)
1055         return {'dipOuterOut': outer,
1056             'dipInnerOut': inner}
1057
1058
1059 class ORingMeasurementDisplayNode(Node):
1060     """
1061     Node for displaying O-Ring dip.Measurements in Flowchart Labels
1062     """
1063     nodeName = 'ORingMeasurementDisplay'
1064
1065     def __init__(self, name):
1066         super().__init__(name=name, terminals={
1067             'dipMsrOutIn': {'io': 'in', 'optional': True},
1068             'dipMsrInIn': {'io': 'in', 'optional': True}
1069         })
1070         self.fc_display_widget_list = None
1071
1072     def setDisplayWidgetList(self, display_widget_list):
1073         """

```

```

1074     Set list of two labels for displaying measurement results
1075     """
1076     self.fc_display_widget_list = display_widget_list
1077
1078     def process(self, dipMsrOutIn, dipMsrInIn, display=True):
1079         """
1080         Node process function. If display=False, skip displaying text
1081         """
1082         if display and self.fc_display_widget_list is not None:
1083             if dipMsrOutIn is None and dipMsrInIn is None:
1084                 # Empty text
1085                 self.fc_display_widget_list[0].setText('')
1086                 self.fc_display_widget_list[1].setText('')
1087             elif dipMsrOutIn is not None and dipMsrInIn is None:
1088                 # Measurement result as string
1089                 outer_disp = str(dipMsrOutIn).replace("\n", "<br>")
1090                 outer_disp = outer_disp.replace(" ", "&nbsp;")
1091                 outer_disp = ''.join((
1092                     '<tt style="font-size:8pt"> <b>Outer
Diameter:</b><br>',
1093                     outer_disp, '</tt>'))
1094                 self.fc_display_widget_list[0].setText(outer_disp)
1095                 # Empty text
1096                 self.fc_display_widget_list[1].setText('')
1097             elif dipMsrOutIn is None and dipMsrInIn is not None:
1098                 # Measurement result as string
1099                 inner_disp = str(dipMsrInIn).replace("\n", "<br>")
1100                 inner_disp = inner_disp.replace(" ", "&nbsp;")
1101                 inner_disp = ''.join((
1102                     '<tt style="font-size:8pt"> <b>Inner
Diameter:</b><br>',
1103                     inner_disp, '</tt>'))
1104                 # Empty text
1105                 self.fc_display_widget_list[0].setText('')
1106                 self.fc_display_widget_list[1].setText(inner_disp)
1107             else:
1108                 # Measurement result as string
1109                 outer_disp = str(dipMsrOutIn).replace("\n", "<br>")
1110                 outer_disp = outer_disp.replace(" ", "&nbsp;")
1111                 outer_disp = ''.join((
1112                     '<tt style="font-size:8pt"> <b>Outer
Diameter:</b><br>',
1113                     outer_disp, '</tt>'))
1114                 inner_disp = str(dipMsrInIn).replace("\n", "<br>")
1115                 inner_disp = inner_disp.replace(" ", "&nbsp;")
1116                 inner_disp = ''.join((
1117                     '<tt style="font-size:8pt"> <b>Inner
Diameter:</b><br>',
1118                     inner_disp, '</tt>'))
1119                 self.fc_display_widget_list[0].setText(outer_disp)
1120                 self.fc_display_widget_list[1].setText(inner_disp)
1121
1122
1123     class SetPixelSizeNode(CtrlNode):
1124         """
1125         Node with control widget for assigning physical quantity to pixels
1126         """
1127         nodeName = 'SetPixelSize'
1128         uiTemplate = [
1129             ('units', 'combo', {'values': [
1130                 'mm', 'um'
1131             ]}),

```

```

1132     ]
1133
1134     def __init__(self, name):
1135         super().__init__(name=name, terminals={
1136             'dipImgIn': {'io': 'in'},
1137             'mmPxRatioIn': {'io': 'in'},
1138             'dipImgOut': {'io': 'out'},
1139         })
1140
1141     def process(self, dipImgIn, mmPxRatioIn, display=True):
1142         """
1143         Node process function. If display=False, no effect
1144         """
1145         units = self.stateGroup.state()['units']
1146         img = dipImgIn.Copy()
1147         img.SetPixelSize(
1148             dip.PixelSize(dip.PhysicalQuantity(mmPxRatioIn, units)))
1149         return {'dipImgOut': img}
1150
1151
1152 class KuwaharaNode(CtrlNode):
1153     """
1154     Node with control widget for Kuwahara-Nagao non-linear, edge-
1155     preserving
1156     smoothing filter
1157     """
1158     nodeName = 'Kuwahara'
1159     uiTemplate = [
1160         ('shape', 'combo', {'values': [
1161             'elliptic', 'rectangular', 'diamond', 'line'
1162         ]}),
1163         ('size', 'spin', {
1164             'value': 11, 'step': 2, 'bounds': [1, 99]}),
1165         ('threshold', 'spin', {
1166             'value': 0., 'step': 0.01, 'bounds': [0, None]}),
1167     ]
1168
1169     def __init__(self, name):
1170         super().__init__(name=name, terminals={
1171             'dipImgIn': {'io': 'in'},
1172             'dipImgOut': {'io': 'out'},
1173         })
1174
1175     def process(self, dipImgIn, display=True):
1176         """
1177         Node process function. If display=False, no effect
1178         """
1179         shape = self.stateGroup.state()['shape']
1180         size = self.ctrls['size'].value()
1181         threshold = self.ctrls['threshold'].value()
1182         kernel = dip.Kernel(size, shape)
1183         img = dip.Kuwahara(dipImgIn, kernel=kernel, threshold=threshold)
1184         return {'dipImgOut': img}
1185
1186 class BilateralFilterNode(CtrlNode):
1187     """
1188     Node with control widget for Bilateral filtering (non-linear,
1189     edge-preserving smoothing)
1190     """
1191     nodeName = 'BilateralFilter'
1192     uiTemplate = [

```

```

1193         ('method', 'combo', {'values': [
1194             'xysep', 'full', 'pwnlinear'
1195         ]}),
1196         ('spatialSigmas', 'spin', {
1197             'value': 2., 'step': 0.01, 'bounds': [0, None]}),
1198         ('tonalSigma', 'spin', {
1199             'value': 30., 'step': 0.1, 'bounds': [0, None]}),
1200         ('truncation', 'spin', {
1201             'value': 2., 'step': 0.01, 'bounds': [0, None]}),
1202     ]
1203
1204     def __init__(self, name):
1205         super().__init__(name=name, terminals={
1206             'dipImgIn': {'io': 'in'},
1207             'dipImgOut': {'io': 'out'},
1208         })
1209
1210     def process(self, dipImgIn, display=True):
1211         """
1212         Node process function. If display=False, no effect
1213         """
1214         method = self.stateGroup.state()['method']
1215         spatialSigmas = [self.ctrls['spatialSigmas'].value()]
1216         tonalSigma = self.ctrls['tonalSigma'].value()
1217         truncation = self.ctrls['truncation'].value()
1218         img = dip.BilateralFilter(
1219             dipImgIn, method=method, spatialSigmas=spatialSigmas,
1220             tonalSigma=tonalSigma, truncation=truncation,
1221         )
1222         return {'dipImgOut': img}
1223
1224
1225     class BinaryAreaClosingNode(CtrlNode):
1226         """
1227         Node with control widget for BinaryAreaClosing of areas smaller than
1228         filterSize
1229         """
1230         nodeName = 'BinaryAreaClosing'
1231         uiTemplate = [
1232             ('connectivity', 'combo', {'values': [
1233                 '8-Connectivity', '4-Connectivity'
1234             ]}),
1235             ('filterSize', 'spin', {
1236                 'value': 0, 'step': 1, 'bounds': [0, None]
1237             }),
1238             ('edgeCondition', 'combo', {'values': [
1239                 'background', 'object'
1240             ]}),
1241         ]
1242
1243     def __init__(self, name):
1244         super().__init__(name=name, terminals={
1245             'dipImgIn': {'io': 'in'},
1246             'dipImgOut': {'io': 'out'},
1247         })
1248
1249     def process(self, dipImgIn, display=True):
1250         """
1251         Node process function. If display=False, no effect
1252         """
1253         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
1254             connectivity = 2

```

```

1255     else:
1256         connectivity = 1
1257         filterSize = int(self.ctrls['filterSize'].value())
1258         edgeCondition = self.stateGroup.state()['edgeCondition']
1259         img = dip.BinaryAreaClosing(
1260             dipImgIn, connectivity=connectivity,
1261             filterSize=filterSize, edgeCondition=edgeCondition
1262         )
1263         return {'dipImgOut': img}
1264
1265
1266 class BinaryAreaOpeningNode(CtrlNode):
1267     """
1268     Node with control widget for BinaryAreaOpening of areas smaller than
1269     filterSize
1270     """
1271     nodeName = 'BinaryAreaOpening'
1272     uiTemplate = [
1273         ('connectivity', 'combo', {'values': [
1274             '8-Connectivity', '4-Connectivity'
1275         ]}),
1276         ('filterSize', 'spin', {
1277             'value': 0, 'step': 1, 'bounds': [0, None]
1278         }),
1279         ('edgeCondition', 'combo', {'values': [
1280             'background', 'object'
1281         ]}),
1282     ]
1283
1284     def __init__(self, name):
1285         super().__init__(name=name, terminals={
1286             'dipImgIn': {'io': 'in'},
1287             'dipImgOut': {'io': 'out'},
1288         })
1289
1290     def process(self, dipImgIn, display=True):
1291         """
1292         Node process function. If display=False, no effect
1293         """
1294         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
1295             connectivity = 2
1296         else:
1297             connectivity = 1
1298             filterSize = int(self.ctrls['filterSize'].value())
1299             edgeCondition = self.stateGroup.state()['edgeCondition']
1300             img = dip.BinaryAreaOpening(
1301                 dipImgIn, connectivity=connectivity,
1302                 filterSize=filterSize, edgeCondition=edgeCondition
1303             )
1304             return {'dipImgOut': img}
1305
1306
1307 class BinaryClosingNode(CtrlNode):
1308     """
1309     Node with control widget for BinaryClosing with iterations
1310     """
1311     nodeName = 'BinaryClosing'
1312     uiTemplate = [
1313         ('connectivity', 'combo', {'values': [
1314             '4-8 Connectivity', '8-4 Connectivity', '8-Connectivity',
1315             '4-Connectivity',
1316         ]}),

```

```
1317         ('iterations', 'spin', {
1318             'value': 3, 'step': 1, 'bounds': [0, None]
1319         }),
1320         ('edgeCondition', 'combo', {'values': [
1321             'special', 'background', 'object'
1322         ]}),
1323     ]
1324
1325     def __init__(self, name):
1326         super().__init__(name=name, terminals={
1327             'dipImgIn': {'io': 'in'},
1328             'dipImgOut': {'io': 'out'},
1329         })
1330
1331     def process(self, dipImgIn, display=True):
1332         """
1333         Node process function. If display=False, no effect
1334         """
1335         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
1336             connectivity = 2
1337         elif self.stateGroup.state()['connectivity'] == '4-
Connectivity':
1338             connectivity = 1
1339         elif self.stateGroup.state()['connectivity'] == '4-8
Connectivity':
1340             connectivity = -1
1341         elif self.stateGroup.state()['connectivity'] == '8-4
Connectivity':
1342             connectivity = -2
1343         iterations = int(self.ctrls['iterations'].value())
1344         edgeCondition = self.stateGroup.state()['edgeCondition']
1345         img = dip.BinaryClosing(
1346             dipImgIn, connectivity=connectivity,
1347             iterations=iterations, edgeCondition=edgeCondition
1348         )
1349         return {'dipImgOut': img}
1350
1351
1352     class BinaryOpeningNode(CtrlNode):
1353         """
1354         Node with control widget for BinaryOpening with iterations
1355         """
1356         nodeName = 'BinaryOpening'
1357         uiTemplate = [
1358             ('connectivity', 'combo', {'values': [
1359                 '4-8 Connectivity', '8-4 Connectivity', '8-Connectivity',
1360                 '4-Connectivity',
1361             ]}),
1362             ('iterations', 'spin', {
1363                 'value': 3, 'step': 1, 'bounds': [0, None]
1364             }),
1365             ('edgeCondition', 'combo', {'values': [
1366                 'special', 'background', 'object'
1367             ]}),
1368         ]
1369
1370     def __init__(self, name):
1371         super().__init__(name=name, terminals={
1372             'dipImgIn': {'io': 'in'},
1373             'dipImgOut': {'io': 'out'},
1374         })
1375
```

```
1376     def process(self, dipImgIn, display=True):
1377         """
1378         Node process function. If display=False, no effect
1379         """
1380         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
1381             connectivity = 2
1382         elif self.stateGroup.state()['connectivity'] == '4-
Connectivity':
1383             connectivity = 1
1384         elif self.stateGroup.state()['connectivity'] == '4-8
Connectivity':
1385             connectivity = -1
1386         elif self.stateGroup.state()['connectivity'] == '8-4
Connectivity':
1387             connectivity = -2
1388         iterations = int(self.ctrls['iterations'].value())
1389         edgeCondition = self.stateGroup.state()['edgeCondition']
1390         img = dip.BinaryOpening(
1391             dipImgIn, connectivity=connectivity,
1392             iterations=iterations, edgeCondition=edgeCondition
1393         )
1394         return {'dipImgOut': img}
1395
1396
1397 class BinaryDilationNode(CtrlNode):
1398     """
1399     Node with control widget for BinaryDilation with iterations
1400     """
1401     nodeName = 'BinaryDilation'
1402     uiTemplate = [
1403         ('connectivity', 'combo', {'values': [
1404             '4-8 Connectivity', '8-4 Connectivity', '8-Connectivity',
1405             '4-Connectivity',
1406         ]}),
1407         ('iterations', 'spin', {
1408             'value': 3, 'step': 1, 'bounds': [0, None]
1409         }),
1410         ('edgeCondition', 'combo', {'values': [
1411             'background', 'object'
1412         ]}),
1413     ]
1414
1415     def __init__(self, name):
1416         super().__init__(name=name, terminals={
1417             'dipImgIn': {'io': 'in'},
1418             'dipImgOut': {'io': 'out'},
1419         })
1420
1421     def process(self, dipImgIn, display=True):
1422         """
1423         Node process function. If display=False, no effect
1424         """
1425         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
1426             connectivity = 2
1427         elif self.stateGroup.state()['connectivity'] == '4-
Connectivity':
1428             connectivity = 1
1429         elif self.stateGroup.state()['connectivity'] == '4-8
Connectivity':
1430             connectivity = -1
1431         elif self.stateGroup.state()['connectivity'] == '8-4
Connectivity':
```

```
1432         connectivity = -2
1433         iterations = int(self.ctrls['iterations'].value())
1434         edgeCondition = self.stateGroup.state()['edgeCondition']
1435         img = dip.BinaryDilation(
1436             dipImgIn, connectivity=connectivity,
1437             iterations=iterations, edgeCondition=edgeCondition
1438         )
1439         return {'dipImgOut': img}
1440
1441
1442 class BinaryErosionNode(CtrlNode):
1443     """
1444     Node with control widget for BinaryErosion with iterations
1445     """
1446     nodeName = 'BinaryErosion'
1447     uiTemplate = [
1448         ('connectivity', 'combo', {'values': [
1449             '4-8 Connectivity', '8-4 Connectivity', '8-Connectivity',
1450             '4-Connectivity',
1451         ]}),
1452         ('iterations', 'spin', {
1453             'value': 3, 'step': 1, 'bounds': [0, None]
1454         }),
1455         ('edgeCondition', 'combo', {'values': [
1456             'object', 'background'
1457         ]}),
1458     ]
1459
1460     def __init__(self, name):
1461         super().__init__(name=name, terminals={
1462             'dipImgIn': {'io': 'in'},
1463             'dipImgOut': {'io': 'out'},
1464         })
1465
1466     def process(self, dipImgIn, display=True):
1467         """
1468         Node process function. If display=False, no effect
1469         """
1470         if self.stateGroup.state()['connectivity'] == '8-Connectivity':
1471             connectivity = 2
1472         elif self.stateGroup.state()['connectivity'] == '4-
Connectivity':
1473             connectivity = 1
1474         elif self.stateGroup.state()['connectivity'] == '4-8
Connectivity':
1475             connectivity = -1
1476         elif self.stateGroup.state()['connectivity'] == '8-4
Connectivity':
1477             connectivity = -2
1478             iterations = int(self.ctrls['iterations'].value())
1479             edgeCondition = self.stateGroup.state()['edgeCondition']
1480             img = dip.BinaryErosion(
1481                 dipImgIn, connectivity=connectivity,
1482                 iterations=iterations, edgeCondition=edgeCondition
1483             )
1484             return {'dipImgOut': img}
1485
1486
1487 class OperatorPlusNode(Node):
1488     """
1489     Node for operator plus
1490     """
```



```

1491     nodeName = 'OperatorPlus'
1492
1493     def __init__(self, name):
1494         super().__init__(name=name, terminals={
1495             'dipImgOneIn': {'io': 'in'},
1496             'dipImgTwoIn': {'io': 'in'},
1497             'dipImgPlusOut': {'io': 'out'},
1498         })
1499
1500     def process(self, dipImgOneIn, dipImgTwoIn, display=True):
1501         """
1502         Node process function. If display=False, no effect
1503         """
1504         img = dipImgOneIn + dipImgTwoIn
1505         return {'dipImgPlusOut': img}
1506
1507
1508 class CombineMeasurementNode(Node):
1509     """
1510     Node for making list of input dip.Measurement objects
1511     """
1512     nodeName = 'CombineMeasurement'
1513
1514     def __init__(self, name):
1515         super().__init__(name=name, terminals={
1516             'dipMsrOutIn': {'io': 'in'},
1517             'dipMsrInIn': {'io': 'in'},
1518             'listMsrOut': {'io': 'out'},
1519         })
1520
1521     def process(self, dipMsrOutIn, dipMsrInIn, display=True):
1522         """
1523         Node process function. If display=False, no effect
1524         """
1525         lst = [dipMsrOutIn, dipMsrInIn]
1526         return {'listMsrOut': lst}

```

1.9. Skripta „process.py“

```

1 import PyDIP as dip
2 import numpy as np
3
4 # Multiprocessing
5 # import multiprocessing
6 import multiprocessing as multiprocessing
7
8 class ProcessParallel:
9     def __init__(self, measureFlowchart, numberProc=1):
10         """
11         Object for parallel processing and preprocessing of image frames
12         """
13         # Flowchart object, queues and processes
14         self.measureFlowchart = measureFlowchart
15         self.input_queue = multiprocessing.JoinableQueue(1)
16         self.output_queue = multiprocessing.Queue()
17         self.numberProc = numberProc
18         self.processes = [
19             ProcessQueue(self.input_queue, self.output_queue)
20             for _ in range(self.numberProc)
21         ]
22

```

```

23     @staticmethod
24     def preprocess_frame(frame):
25         """
26         Hardcoded function for real-time frame preprocessing. Returns
image
27         for measurement of None
28
29         :param frame: 2D array
30         :return: np.array or None
31         """
32         img = dip.Image(frame)
33         img_center_x = img.Sizes()[0]//2
34         # Range Thresholding
35         thr = dip.RangeThreshold(
36             img, lowerBound=30, upperBound=240,
37             foreground=1.0, background=0.0
38         )
39         # Fill holes
40         fill = dip.FillHoles(thr)
41         # Label and measure
42         lbl = dip.Label(fill, connectivity=1, minSize=10000)
43         if dip.GetObjectLabels(lbl):
44             msr = dip.MeasurementTool.Measure(
45                 lbl, img, ['Center', 'Minimum', 'Maximum',
'PodczeckShapes'])
46             for obj in msr.Objects():
47                 # Ellipse Podczeck
48                 if 0.98 <= msr['PodczeckShapes'][obj][3] <= 1.02:
49                     # Object location near x-axis center
50                     if (img_center_x-150 <= msr['Center'][obj][0] <=
51                         img_center_x+150):
52                         # Expand box around object for measurement
53                         min_x = int(msr['Minimum'][obj][0] - 50)
54                         min_y = int(msr['Minimum'][obj][1] - 50)
55                         max_x = int(msr['Maximum'][obj][0] + 50)
56                         max_y = int(msr['Maximum'][obj][1] + 50)
57                         # Only one object can be measured
58                         return np.asarray(
59                             img[min_x:max_x, min_y:max_y], np.uint32
60                         )
61             # Return None if no object is near x-axis center
62             return None
63
64     def start(self):
65         """
66         Start parallel execution
67         """
68         [proc.start() for proc in self.processes]
69
70     def addInput(self, frame):
71         """
72         Add frame to input Queue
73         """
74         self.input_queue.put((self.preprocess_frame, frame))
75
76     def addMeasurementInput(self, measured_object):
77         """
78         Add image of measured object to input Queue for measuring
79         :param measured_object: np.ndarray
80         """
81         mes_obj = dip.Image(measured_object)
82         self.input_queue.put((self.measureFlowchart.fc_process, mes_obj))

```

```
83
84     def getOutput(self):
85         """
86         Retrieve Queue output
87         """
88         return self.output_queue.get()
89
90     def getNumProc(self):
91         """
92         Return number of processes
93         """
94         return self.numberProc
95
96     def ifInputQueueFull(self):
97         """
98         Check if input queue is full
99
100        :return: True or False
101        """
102        return self.input_queue.full()
103
104     def ifOutputQueueFull(self):
105         """
106         Check if output queue is full
107
108        :return: True or False
109        """
110        return self.output_queue.full()
111
112     def ifOutputQueueEmpty(self):
113         """
114         Check if output queue is empty
115
116        :return: True or False
117        """
118        return self.output_queue.empty()
119
120     def join(self):
121         """
122         Wait for processing to finish
123         """
124         self.input_queue.join()
125
126     def stop(self):
127         """
128         Stop parallel execution
129         """
130         [self.input_queue.put(None) for _ in range(self.numberProc)]
131
132
133 class ProcessQueue(multiprocessing.Process):
134     def __init__(self, input_queue, output_queue):
135         """
136         Process for taking data from input_queue and writing into
137         output_queue.
138         Exit with poison pill (None)
139         """
140         multiprocessing.Process.__init__(self)
141         self.input_queue = input_queue
142         self.output_queue = output_queue
143
144     def run(self):
```

```

144     """
145     Infinite loop with poison pill exit
146     """
147     while True:
148         input_tup = self.input_queue.get()
149         if input_tup is None:
150             self.input_queue.task_done()
151             break
152         func, args = input_tup
153         res = func(args)
154         self.input_queue.task_done()
155         self.output_queue.put(res)

```

Prilog 2.: Programski kod za analizu sposobnosti mjernog sustava

2.1. Skripta „measurement_system_analysis.R“

```

1 library(readr)
2 library(qualityTools)
3 library(SixSigma)
4 library(irr)
5 library(ggplot2)
6 library(Hmisc)
7 library(dplyr)
8 library(tidyr)
9
10 r_r_calc <- function(filename, output_image, fig_title, fig_subtitle){
11
12     Measurement_Data <- read.csv2(
13         filename
14     )
15     # ICC
16     ratings <- t(rbind(
17         colMeans(Measurement_Data[Measurement_Data$Mjeritelj == 'A', 3:12]),
18         colMeans(Measurement_Data[Measurement_Data$Mjeritelj == 'B', 3:12]),
19         colMeans(Measurement_Data[Measurement_Data$Mjeritelj == 'C', 3:12])
20     ))
21     icc_data <- icc(
22         ratings = ratings,
23         model = "twoway",
24         type = "consistency",
25         unit = "single",
26         conf.level = 0.95
27     )
28     print(icc_data)
29     cat("\n\n")
30     # Factor with 90 values for operators
31     operators <- factor(rep(Measurement_Data$Mjeritelj, each=10))
32     # View(operators)
33     # Each O-ring (10 parts) is measured 3 times by 3 operators
34     o_ring <- factor(rep(rep(1:10, each=3), 3))
35     # View(o_ring)
36     # 10 parts are measured 3 times each
37     run_num <- factor(rep(Measurement_Data$Broj.mjerenja, 10))
38     # View(run_num)
39     # Measurements
40     measurements <- c(
41         as.numeric(unlist(Measurement_Data[Measurement_Data$Mjeritelj=='A',
3:12])),

```

```
42   as.numeric(unlist(Measurement_Data[Measurement_Data$Mjeritelj=='B',
3:12])),
43   as.numeric(unlist(Measurement_Data[Measurement_Data$Mjeritelj=='C',
3:12]))
44   )
45   # View(measurements)
46   # Measurements as data frame
47   measurements_df = data.frame(operators, o_ring, run_num, measurements)
48   # Graphics device
49   png(filename = output_image, width = 1949, height = 2700,
50       units = 'px', res = 300)
51   # View(measurements_df)
52   r_r_study = ss.rr(
53     var = measurements,
54     part = o_ring,
55     appr = operators,
56     data = measurements_df,
57     sigma = 6,
58     main = fig_title,
59     sub = fig_subtitle
60   )
61   # Reset device
62   dev.off()
63   # r_r_study
64 }
65
66 r_r_autom <- function(filename, output_image, fig_title, fig_subtitle){
67   # Load input data from automated measurements
68   Measurement_Data_Automated <- read_delim(
69     filename, ";", escape_double = FALSE,
70     col_types = cols(`Broj mjerenja` = col_factor(
71       levels = c("1", "2", "3", "4", "5", "6", "7", "8", "9"))),
72     locale = locale(decimal_mark = ","), skip = 1)
73   # View(Measurement_Data_Automated)
74   # ICC
75   icc_data <- icc(
76     ratings = t(Measurement_Data_Automated[3:12]),
77     model = "twoway",
78     type = "consistency",
79     unit = "single",
80     conf.level = 0.95
81   )
82   print(icc_data)
83   cat("\n\n")
84   # Optical measurement system
85   mes_syst <- factor(rep('Optical System', 90))
86   # Each O-ring (10 parts) is measured 3 times by 3 operators
87   o_ring <- factor(rep(1:10, each=9))
88   # 10 parts are measured, 9 times each
89   run_num <- factor(rep(Measurement_Data_Automated$`Broj mjerenja`, 10))
90   # Measurements
91   measurements <- as.numeric(unlist(Measurement_Data_Automated[3:12]))
92   # Measurements as data.frame
93   measurements_df <- data.frame(mes_syst, o_ring, run_num, measurements)
94   # Graphics device
95   png(filename = output_image, width = 1949, height = 2700,
96       units = 'px', res = 300)
97   # View(measurements_df)
98   r_r_study = ss.rr(
99     var = measurements,
100    part = o_ring,
101    appr = mes_syst,
```

```
102     data = measurements_df,
103     sigma = 6,
104     main = fig_title,
105     sub = fig_subtitle
106   )
107   # Reset device
108   dev.off()
109   # r_r_study
110 }
111
112 icc_comparison <- function(input_manual, input_automated, output_image,
113                             fig_title){
114   # Load input data from automated measurements
115   Measurement_Data_Automated <- read_delim(
116     input_automated, ";", escape_double = FALSE,
117     col_types = cols(`Broj mjerenja` = col_factor(
118       levels = c("1", "2", "3", "4", "5", "6", "7", "8", "9"))),
119     locale = locale(decimal_mark = ","), skip = 1)
120   # Load input data from manual measurements
121   Measurement_Data <- read.csv2(
122     input_manual
123   )
124   # Formatting data in data.frame
125   autom_manual <- data.frame(t(rbind(
126     colMeans(Measurement_Data[3:12]),
127     colMeans(Measurement_Data_Automated[3:12]))),
128     row.names = 1:10)
129   names(autom_manual) <- c("Manual", "System")
130   # ICC
131   icc_data <- icc(
132     ratings = autom_manual,
133     model = "twoway",
134     type = "consistency",
135     unit = "single",
136     conf.level = 0.95
137   )
138   print(icc_data)
139   cat("\n\n")
140   # Graphics device
141   png(filename = output_image, width = 1800, height = 1200,
142     units = 'px', res = 300)
143   # Plot data
144   ggplot(data = autom_manual, aes(x = Manual, y = System)) +
145     geom_point(shape=1) +
146     geom_smooth(aes(color="Linear regression"), method = lm) +
147     geom_abline(aes(slope = 1, intercept = 0, color="Ideal"),
148       show.legend = TRUE) +
149     scale_color_manual(values = c("black", "dodgerblue2")) +
150     labs(color = paste("ICC(C,1) =", round(icc_data$value, 3)),
151       title = fig_title)
152   ggsave(filename = output_image)
153   # Reset device
154   dev.off()
155
156 }
157
158 capability_study <- function(input_file){
159   # Load input file
160   Measurement_Data_Capability <- read.csv2(input_file, row.names=1)
161   # O-Ring names
162   names(Measurement_Data_Capability) <- 1:10
163   # Graphics device for autom_in
```

```
164 png(filename = "O-Ring_Automated_In_Cp.png", width = 1800, height =
1900,
165     units = 'px', res = 300)
166 # Capability study for autom_in
167 ss.study.ca(as.numeric(Measurement_Data_Capability["autom_in",]),
168             LSL = 15.57, USL = 16.03, Target = 15.8,
169             f.main = "O-Ring Inner Diameter (System)",
170             f.sub = "Six Sigma Capability Analysis")
171 dev.off()
172 # Graphics device for manual_in
173 png(filename = "O-Ring_Manual_In_Cp.png", width = 1800, height = 1900,
174     units = 'px', res = 300)
175 # Capability study for manual_in
176 ss.study.ca(as.numeric(Measurement_Data_Capability["manual_in",]),
177             LSL = 15.57, USL = 16.03, Target = 15.8,
178             f.main = "O-Ring Inner Diameter (Manual)",
179             f.sub = "Six Sigma Capability Analysis")
180 dev.off()
181 # Graphics device for autom_out
182 png(filename = "O-Ring_Automated_Out_Cp.png", width = 1800, height =
1900,
183     units = 'px', res = 300)
184 # Capability study for autom_out
185 ss.study.ca(as.numeric(Measurement_Data_Capability["autom_out",]),
186             LSL = 20.21, USL = 20.99, Target = 20.6,
187             f.main = "O-Ring Outer Diameter (System)",
188             f.sub = "Six Sigma Capability Analysis")
189 dev.off()
190 # Graphics device for manual_out
191 png(filename = "O-Ring_Manual_Out_Cp.png", width = 1800, height = 1900,
192     units = 'px', res = 300)
193 # Capability study for manual_out
194 ss.study.ca(as.numeric(Measurement_Data_Capability["manual_out",]),
195             LSL = 20.21, USL = 20.99, Target = 20.6,
196             f.main = "O-Ring Outer Diameter (Manual)",
197             f.sub = "Six Sigma Capability Analysis")
198 dev.off()
199 # Graphics device for manual_thick
200 png(filename = "O-Ring_Manual_Thick_Cp.png", width = 1800, height =
1900,
201     units = 'px', res = 300)
202 # Capability study for manual_thick
203 ss.study.ca(as.numeric(Measurement_Data_Capability["manual_thick",]),
204             LSL = 2.32, USL = 2.48, Target = 2.4,
205             f.main = "O-Ring Cross-section (Manual)",
206             f.sub = "Six Sigma Capability Analysis")
207 dev.off()
208 }
209
210 calib_sensitivity <- function(input_file, fig_title, output_image){
211 # Load input data from automated measurements
212 Measurement_Data_Automated <- read_delim(
213     input_file, ";", escape_double = FALSE,
214     col_types = cols(`Broj mjerenja` = col_factor(
215         levels = c("1", "2", "3", "4", "5", "6", "7", "8", "9"))),
216     locale = locale(decimal_mark = ","), skip = 1)
217 # Remove column from data.frame
218 Measurement_Data_Automated <- subset(
219     Measurement_Data_Automated, select = -`Broj mjerenja`)
220 # Change column name
221 names(Measurement_Data_Automated)[1] <- "mm/px ratio"
222 # Correlation matrix
```

```
223 print("Correlation Matrix")
224 corr_mat <- rcorr(as.matrix(Measurement_Data_Automated), type =
"pearson")
225 Pearson_R <- matrix(rep(corr_mat$r[1,2:11], each=9))
226 print(corr_mat$r[1,])
227 print(corr_mat$P[1,])
228 cat("\n\n")
229 # Linear regression coefficients
230 reg_coeff <- c()
231 for (ring in colnames(Measurement_Data_Automated[2:11])){
232   print(ring)
233   linMod <- lm(Measurement_Data_Automated[[ring]] ~ `mm/px ratio`,
234               data = Measurement_Data_Automated)
235   print(linMod)
236   reg_coeff <- cbind(reg_coeff, linMod$coefficients[2])
237 }
238 reg_coeff <- round(reg_coeff, 2) %>% rep(each=9) %>% factor()
239 # Graphics device
240 png(filename = output_image, width = 1949, height = 2700,
241      units = 'px', res = 300)
242 # Plot data in grid
243 Measurement_Data_Automated %>%
244   gather(-`mm/px ratio`, key = "var", value = "value") %>%
245   ggplot(aes(x = `mm/px ratio`, y = value)) +
246   geom_point(aes(fill = reg_coeff), shape=21, size = 3) +
247   geom_smooth(aes(color = Pearson_R), method = lm) +
248   scale_fill_viridis_d(name = "Sensitivity", option = "D") +
249   scale_color_continuous(name="Pearson's r") +
250   facet_wrap(~var, ncol = 2, scales = "fixed") +
251   xlab("mm/px ratio [mm/px]") +
252   ylab("Diameter [mm]") +
253   labs(title = fig_title, subtitle = "Calibration sensitivity")
254 # Save data
255 ggsave(filename = output_image)
256 # Reset device
257 dev.off()
258 }
259
260 print('ICC and R&R for O-ring cross-section')
261 r_r_calc(filename = "Measurement_Data_Thick.csv",
262          output_image = "O-Ring_Thick_R_R.png",
263          fig_subtitle = "Six Sigma O-Ring R&R",
264          fig_title = "O-Ring Cross-section (Micrometer)")
265
266 print('ICC and R&R for O-ring inner diameter')
267 r_r_calc(filename = "Measurement_Data_In.csv",
268          output_image = "O-Ring_In_R_R.png",
269          fig_subtitle = "Six Sigma O-Ring R&R",
270          fig_title = "O-Ring Inner Diameter (Micrometer)")
271
272 print('ICC and R&R for O-ring outer diameter')
273 r_r_calc(filename = "Measurement_Data_Out.csv",
274          output_image = "O-Ring_Out_R_R.png",
275          fig_subtitle = "Six Sigma O-Ring R&R",
276          fig_title = "O-Ring Outer Diameter (Micrometer)")
277
278 print('ICC and R&R for O-ring automated outer diameter')
279 r_r_autom(filename = "Measurement_Data_Automated_Out.csv",
280          output_image = "O-Ring_Automated_Out_R_R.png",
281          fig_subtitle = "Six Sigma O-Ring R&R",
282          fig_title = "O-Rign Outer Diameter (System)")
283
```



```

284 print('ICC and R&R for O-ring automated inner diameter')
285 r_r_autom(filename = "Measurement_Data_Automated_In.csv",
286           output_image = "O-Ring_Automated_In_R_R.png",
287           fig_subtitle = "Six Sigma O-Ring R&R",
288           fig_title = "O-Rign Inner Diameter (System)")
289
290 print('ICC (System - Manual) for O-ring inner diameter')
291 icc_comparison(input_manual = "Measurement_Data_In.csv",
292               input_automated = "Measurement_Data_Automated_In.csv",
293               output_image = "O-Ring_Comparison_In.png",
294               fig_title = "O-Ring Inner Diameter comparison")
295
296 print('ICC (System - Manual) for O-ring outer diameter')
297 icc_comparison(input_manual = "Measurement_Data_Out.csv",
298               input_automated = "Measurement_Data_Automated_Out.csv",
299               output_image = "O-Ring_Comparison_Out.png",
300               fig_title = "O-Ring Outer Diameter comparison")
301
302 print("Capability study")
303 capability_study(input_file = "Measurement_Data_Capability.csv")
304
305 print("Calibration (System) sensitivity O-ring Outer Diameter")
306 calib_sensitivity(input_file = "Measurement_Data_Automated_Out.csv",
307                  fig_title = "O-ring Outer Diameter",
308                  output_image = "O-
Ring_Calibration_Sensitivity_Out.png")
309
310 print("Calibration (System) sensitivity O-ring Inner Diameter")
311 calib_sensitivity(input_file = "Measurement_Data_Automated_In.csv",
312                  fig_title = "O-ring Inner Diameter",
313                  output_image = "O-Ring_Calibration_Sensitivity_In.png")

```

Prilog 3.: Potvrde o umjeravanju

3.1. Potvrda o umjeravanju mikrometarskog vijka *Mitutoyo* MDC-25SX

Issue No./発行No.		18105095	
CERTIFICATE OF INSPECTION / 検査成績書			
1. Item/対象製品			
Product name/品名	Digimatic Micrometer	Measuring range/測定範囲	0-25mm
Model/符号	MDC-25SX	Resolution/最小表示量	0.001mm
Code No./コードNo.	293-821-30	Serial No./製造No.	66544609
2. Result of Inspection/検査結果			
Inspection standard: Mitutoyo standard 1)		Standard temperature/標準温度 20 °C	
Performance/性能			
		Permissible error. /許容値	Measured value /実測値
Flatness of measuring face /平面度	Anvil /アンビル	0.3	0.0
	Spindle /スピンドル	0.3	0.0
Parallelism of measuring faces /平行度		2.0	1.3
[μm]			
Measuring length /測定長(mm)	Permissible error /許容値	Error /器差	Uncertainty of measurement /測定の不確かさ
0.00	± 2	0	U = 1.1 [k=2]
7.70		0	
12.90		0	
17.60		+1	
22.80		+1	
25.00		+1	
1) Based on: JISB7502(1994), ISO3611(1978), DIN863-1(1999) Traceable to: NMIJ/AIST by JCSS No.0030, NIST via No.683/290402-17*, No.683/281740-12 PTB via No.50308PTB14* * Confirmation of technical capability by the transfer standard.(ISO/IEC 17043)			
3. Judgment /総合判定	Passed /合格		この検査成績書には安心してご利用いただけます よう出荷時の検査データを記載していますが、 校正証明書取得用にはご利用できません。
4. QC Manager	T.Hoshino		
Mitutoyo Corporation			

3.2. Potvrda o umjeravanju kontrolne palice NM.KPR.002



List 1 od 2

POTVRDA O UMJERAVANJU BR.01/2014

Podnositelj zahtjeva: KOVINOTOKARSKA RADNJA Mandić Nandor

Mjerilo: KONTROLNA PALICA

Mjerno područje: 25mm

Proizvođač:

Vrsta:

Serijski broj/oznaka: NM.KPR.002

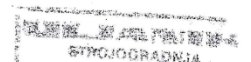
Datum izvršenja umjeravanja: 23.01.2014.

Rezultati mjerenja: Rezultati mjerenja prikazani su u tablici 1 na listu 2

Datum izdavanja: 28.01.2014.

Mjerenje izvršio:

Vučetić Željko

Odgovorna osoba
rukovoditelj Kontrole kvaliteteKONTROLA KVALITETE
Gržinić Luciano dipl.ing.



List 2 od 2

Potvrda o umjeravanju broj:01/2014

- Postupak mjerenja: Umjeravanje je izvršeno sukladno dokumentu :Uputa za rad u izdavaoni mjernih instrumenata br.15 broj: RU266-001. Korišten je uređaj za mjerenje duljina-TRIMOS HORIZON H 2000
- Sljedivost: Sljedivost je osigurana :
-referentni dokumenti
-horizontalni mjerni instrument TRIMOS Horizon H 2000, 750 50 0005 03 sa samokontroliranim procesorom (S/N 1335/A)
- Metoda mjerenja: Mjerenje je izvršeno direktnom metodom.
- Uvjeti mjerenja: Mjerenja su izvršena pri temperaturi okoliša (20 ± 1)°C
- Mjesto mjerenja: Izdavaonica instrumenata
- Napomena: Max.pogreška $3 \mu\text{m} + L(\text{mm})/300 = \mu\text{m}$

Rezultati mjerenja :
Tablica 1 : Rezultati mjerenja duljina kontrolne palice

Nazivna duljina	Izmjerena duljina	Odstupanje	Mjerna nesigurnost
mm	mm	μm	μm
25	24,999	1 μm	

Prilog 4.: Mjerni listovi s izmjerenim vrijednostima veličina mikrometarskim vijkom

4.1. Mjerni list s izmjerenim vrijednostima poprečnog presjeka [mm] O-prstena

PROIZVOD: 15,8 x 2,4		MJERENA VELIČINA: POPREČNI PRESJEK (DEBLJINA)										
MJERMI VREĐAJ: HITUTOYO DIGMATIC		MICROMETER NDC-25X										
MJERITELJ	BR. MJER.	15,8x2,4-1	15,8x2,4-2	15,8x2,4-3	15,8x2,4-4	15,8x2,4-5	15,8x2,4-6	15,8x2,4-7	15,8x2,4-8	15,8x2,4-9	15,8x2,4-10	15,8x2,4-11
A	1.	2,430	2,412	2,385	2,375	2,387	2,387	2,370	2,408	2,406	2,389	
	2.	2,444	2,418	2,358	2,372	2,395	2,371	2,372	2,412	2,407	2,403	
	3.	2,450	2,419	2,356	2,384	2,386	2,391	2,367	2,418	2,397	2,408	
B	1.	2,433	2,418	2,383	2,376	2,388	2,395	2,370	2,420	2,399	2,402	
	2.	2,438	2,417	2,380	2,368	2,399	2,387	2,356	2,411	2,400	2,396	
	3.	2,443	2,412	2,387	2,373	2,397	2,388	2,368	2,433	2,400	2,402	
C	1.	2,438	2,429	2,370	2,371	2,396	2,392	2,361	2,417	2,396	2,397	
	2.	2,444	2,412	2,369	2,377	2,385	2,395	2,356	2,422	2,407	2,402	
	3.	2,430	2,413	2,382	2,376	2,395	2,376	2,373	2,413	2,404	2,390	

DATUM:
18.12.2018.

4.2. Mjerni list s izmjerenim vrijednostima unutarnjeg promjera [mm] O-prstena

PROIZVOD: 15,8 x 2,4		MJERENA VELIČINA: UNUTARNJI PROMJER									
MJERNI UREĐAJ: MIKROMETAR ZA UNUTARNJIA MJERENJA											
MJERITELJ	BR. MJER.	15,8x2,4-1	15,8x2,4-2	15,8x2,4-3	15,8x2,4-4	15,8x2,4-5	15,8x2,4-6	15,8x2,4-7	15,8x2,4-8	15,8x2,4-9	15,8x2,4-10
A	1.	15,87	16,08 15,77	15,81	15,90	16,07 15,74	15,90	15,86	15,90	15,79	15,87
	2.	15,87	15,88	15,82	15,85	15,78	15,77	15,82	15,85	15,84	15,87
	3.	15,86	15,79	15,74	15,78	15,77	15,75	15,77	15,74	15,76	15,85
B	1.	15,75	15,68	15,77	15,80	15,73	15,74	15,90	15,67	15,68	15,75
	2.	15,76	15,70	15,74	15,78	15,71	15,73	15,79	15,70	15,71	15,68
	3.	15,74	15,65	15,74	15,70	15,69	15,78	15,82	15,64	15,63	15,79
C	1.	15,73	15,71	15,72	15,79	15,71	15,77	15,81	15,68	15,72	15,74
	2.	15,76	15,67	15,69	15,76	15,67	15,68	15,75	15,70	15,68	15,71
	3.	15,75	15,75	15,68	15,74	15,73	15,71	15,77	15,70	15,70	15,76

DATUM:
19.12.2018.

4.3. Mjerni list s izmjerenim vrijednostima vanjskog promjera [mm] O-prstena

PROIZVOD: 15,8 x 2,4		MJERENA VELIČINA: VANJSKI PROMJER									
MJERNI VEŠTAJ: MITUTOYO DIGMATIC											
MICROMETER MDC-25SX											
MJERITELJ	BR. MJER.	15,8x2,4-1	15,8x2,4-2	15,8x2,4-3	15,8x2,4-4	15,8x2,4-5	15,8x2,4-6	15,8x2,4-7	15,8x2,4-8	15,8x2,4-9	15,8x2,4-10
A	1.	20,453	20,385	20,301	20,505	20,541	20,287	20,503	20,347	20,465	20,467
	2.	20,475	20,212	20,411	20,438	20,463	20,378	20,400	20,536	20,246	20,411
	3.	20,582	20,337	20,385	20,407	20,328	20,412	20,513	20,328	20,343	20,348
B	1.	20,422	20,400	20,438	20,361	20,463	20,433	20,561	20,239	20,332	20,514
	2.	20,514	20,423	20,418	20,476	20,413	20,498	20,576	20,520	20,342	20,431
	3.	20,535	20,466	20,388	20,425	20,422	20,420	20,517	20,402	20,349	20,425
C	1.	20,481	20,522	20,428	20,467	20,540	20,460	20,502	20,312	20,472	20,521
	2.	20,607	20,513	20,550	20,450	20,484	20,493	20,525	20,545	20,502	20,487
	3.	20,594	20,494	20,465	20,526	20,476	20,473	20,595	20,380	20,493	20,565

DATUM:
18.12.2018.

Prilog 5.: Rezultati mjerenja O-prstena pomoću optičkog mjernog sustava

5.1. Rezultati prvog mjerenja svih O-prstena

	Perimeter [mm]	Feret Max [mm]	Feret Min [mm]	Feret PerpMin [mm]	Feret MaxAng [rad]	Feret MinAng [rad]	Radius Max [mm]
1_out	68,15602	21,12203	20,64768	20,95088	-1,84555	0,30288	10,59824
1_in	51,80185	16,10752	15,76277	16,10539	-1,71958	3,00904	8,06455
2_out	66,68040	20,87211	20,56857	20,70501	2,05345	0,00000	10,46564
2_in	52,43849	16,03889	15,69079	15,92956	-1,74608	0,00000	8,02957
3_out	67,50683	20,84143	20,53446	20,67090	-2,34231	0,00000	10,44910
3_in	51,65050	15,88395	15,69459	15,86717	-2,01235	-0,32175	7,95836
4_out	73,95212	20,92417	20,58514	20,72597	-2,57936	1,59977	10,50679
4_in	52,76056	16,02833	15,68852	15,97663	-2,94670	-1,12104	8,03085
5_out	67,48206	20,78175	20,52586	20,74294	2,27835	3,92699	10,43175
5_in	50,92030	15,90105	15,65368	15,82252	2,90338	3,92699	7,96597
6_out	68,27537	20,89991	20,73195	20,89375	1,93281	0,32175	10,47535
6_in	52,75749	15,97595	15,79282	15,89893	1,69065	3,94032	8,00258
7_out	65,93659	20,84617	20,43213	20,77324	1,68724	0,00000	10,48103
7_in	56,08083	16,31080	15,72490	16,30477	-1,59799	0,00000	8,24521
8_out	69,26481	20,66406	20,25815	20,65595	2,08319	0,54042	10,38030
8_in	51,34586	16,38099	15,69776	16,11289	1,60204	4,17868	8,43905
9_out	66,89444	20,76775	20,50443	20,55339	2,36432	1,65063	10,42650
9_in	51,69471	15,90017	15,74985	15,88388	3,02549	1,31019	7,96971
10_out	66,91221	20,75079	20,49434	20,55360	-1,95497	1,61425	10,41019
10_in	51,23743	15,89303	15,74985	15,85312	2,75192	1,31019	7,96437

	Radius Mean [mm]	Radius Min [mm]	Radius StdDev [mm]	Roundness □	PodczeckShapes Square □	PodczeckShapes Circle □	PodczeckShapes Triangle □
1_out	10,39854	10,24428	0,07012	1,00000	0,78501	0,98504	1,57002
1_in	7,93605	7,77665	0,05293	1,00000	0,77929	0,97111	1,55858
2_out	10,31591	10,15804	0,04190	1,00000	0,78511	0,99304	1,57022
2_in	7,90179	7,78956	0,05170	1,00000	0,78450	0,98389	1,56901
3_out	10,29695	10,11769	0,04398	1,00000	0,78469	0,99250	1,56937
3_in	7,86273	7,75122	0,03299	1,00000	0,77990	0,98220	1,55981
4_out	10,27681	10,09446	0,07311	1,00000	0,77780	0,98359	1,55559
4_in	7,87851	7,75420	0,05362	1,00000	0,77771	0,97235	1,55542
5_out	10,27683	10,10766	0,04188	1,00000	0,77919	0,98171	1,55838
5_in	7,85089	7,71861	0,03804	1,00000	0,78175	0,98473	1,56349
6_out	10,35936	10,17476	0,04404	1,00000	0,77851	0,98355	1,55701
6_in	7,91075	7,79253	0,03291	1,00000	0,78305	0,99036	1,56611
7_out	10,29401	10,17457	0,04880	1,00000	0,78419	0,98207	1,56838
7_in	7,94858	7,76173	0,12395	1,00000	0,77311	0,94935	1,54623
8_out	10,16453	9,66186	0,12493	1,00000	0,77699	0,97024	1,55398
8_in	7,95573	7,72623	0,10999	1,00000	0,78600	0,97498	1,57199
9_out	10,26453	10,15494	0,04060	1,00000	0,78528	0,99747	1,57056
9_in	7,87640	7,78487	0,02725	1,00000	0,77901	0,98350	1,55803
10_out	10,26461	10,16780	0,03958	1,00000	0,78560	0,99738	1,57120
10_in	7,87797	7,78272	0,02669	1,00000	0,78091	0,98781	1,56182

	PodczechShapes Ellipse []	PodczechShapes Elongation []	Convexity []	Center dim0 [mm]	Center dim1 [mm]	SolidArea [mm ²]	Size [mm ²]	ConvexPerimeter [mm]
1_out	0,99950	3,22677	0,96362	12,04715	12,19931	291.860,00000	339,58483	65,67650
1_in	0,99222	3,21601	0,96671	12,05149	12,21546	170.047,00000	197,83440	50,07735
2_out	0,99963	3,19471	0,97560	12,02014	12,09781	287.366,00000	334,35597	65,05340
2_in	0,99886	3,26946	0,95089	12,00115	12,13030	168.543,00000	196,08447	49,86325
3_out	0,99909	3,23907	0,96232	11,97970	12,04373	286.263,00000	333,07261	64,96326
3_in	0,99300	3,25174	0,96031	11,96910	12,06528	166.929,00000	194,21818	49,60065
4_out	0,99032	3,53429	0,88119	12,11310	12,01960	285.207,00000	331,84393	65,16574
4_in	0,99021	3,29171	0,94353	12,11212	11,95734	167.558,00000	194,93258	49,78111
5_out	0,99210	3,24718	0,96062	12,02180	11,94881	285.129,00000	331,75318	64,82485
5_in	0,99535	3,20232	0,97241	12,01458	11,96715	166.416,00000	193,62362	49,51525
6_out	0,99122	3,26678	0,95732	12,04448	12,11362	289.832,00000	337,22404	65,36153
6_in	0,99702	3,30231	0,94608	12,05203	12,11317	168.999,00000	196,61619	49,91285
7_out	0,99846	3,16301	0,98444	11,98014	12,04210	286.066,00000	332,84339	64,91079
7_in	0,98436	3,43826	0,89549	11,94510	11,96003	170.404,00000	198,21952	50,22004
8_out	0,98929	3,35195	0,92738	11,95502	11,99016	279.438,00000	325,13159	64,23484
8_in	1,00076	3,13448	0,98022	11,91608	11,90107	170.867,00000	198,80710	50,33030
9_out	0,99985	3,22107	0,96880	12,04414	11,99975	284.435,00000	330,94569	64,80708
9_in	0,99187	3,25121	0,96103	12,07408	11,99488	167.503,00000	194,88488	49,68006
10_out	1,00026	3,22456	0,96862	12,01914	11,98613	284.414,00000	330,92126	64,81283
10_in	0,99429	3,22389	0,96973	12,04747	11,97922	167.589,00000	194,98145	49,68660

5.2. Rezultati drugog mjerenja svih O-prstena

	Perimeter [mm]	Feret Max [mm]	Feret Min [mm]	Feret PerpMin [mm]	Feret MaxAng [rad]	Feret MinAng [rad]	Radius Max [mm]
1_out	65,31713	21,02325	20,28049	20,86153	-1,67805	0,28379	10,56637
1_in	51,02975	16,20373	15,99225	16,16274	2,73256	4,71239	8,11533
2_out	64,55152	20,73838	20,18638	20,69786	-1,63332	0,00000	10,38042
2_in	55,82033	17,05875	16,05744	16,92792	-2,98302	4,38784	8,70814
3_out	67,08020	20,83244	20,52736	20,66376	-2,36082	0,00000	10,45383
3_in	51,44712	15,99839	15,68536	15,78766	-2,18659	0,00000	8,06635
4_out	71,37796	20,86456	20,48288	20,84883	-2,46736	2,28386	10,48511
4_in	52,16257	16,22911	15,67500	16,18007	-2,98335	-1,52735	8,16495
5_out	66,91854	20,83874	20,51326	20,79437	-3,05804	-0,33746	10,45579
5_in	50,60894	15,91529	15,66791	15,89387	-2,09101	-0,41451	7,96640
6_out	69,23776	20,87807	20,54136	20,64253	2,73348	1,71969	10,47293
6_in	51,77034	15,85054	15,66209	15,80243	2,55836	3,94921	7,94251
7_out	68,41919	20,89566	20,67041	20,83422	1,94676	0,79910	10,47207
7_in	52,88959	15,96201	15,80653	15,92161	1,71874	0,56673	8,00077
8_out	67,28670	20,85745	20,49326	20,76605	1,68055	0,00000	10,48929
8_in	51,80709	16,18632	15,70995	15,92355	2,29359	3,42764	8,18103
9_out	67,26543	20,80859	20,42811	20,73612	2,58768	3,64869	10,44751
9_in	52,23991	15,96649	15,56272	15,90638	2,45451	1,01530	7,99302
10_out	67,63697	20,76101	20,49634	20,68573	2,36665	1,61954	10,42060
10_in	51,13447	15,89778	15,74472	15,86527	2,79814	0,78540	7,97044

	Radius Mean [mm]	Radius Min [mm]	Radius StdDev [mm]	Roundness □	PodczechShapes Square □	PodczechShapes Circle □	PodczechShapes Triangle □
1_out	10,27736	10,07939	0,10956	1,00000	0,78404	0,97047	1,56809
1_in	8,04445	7,94176	0,02718	1,00000	0,78634	0,99064	1,57269
2_out	10,21202	10,05366	0,09095	1,00000	0,78411	0,97369	1,56823
2_in	8,24806	7,74181	0,22292	1,00000	0,78539	0,94856	1,57077
3_out	10,29180	10,10486	0,04734	1,00000	0,78447	0,99222	1,56893
3_in	7,86943	7,73849	0,03392	1,00000	0,78554	0,99370	1,57108
4_out	10,27388	10,09691	0,07002	1,00000	0,77682	0,97172	1,55365
4_in	7,93123	7,67822	0,11513	1,00000	0,77832	0,96006	1,55664
5_out	10,29572	10,15486	0,04893	1,00000	0,78051	0,98034	1,56102
5_in	7,87418	7,75377	0,04112	1,00000	0,78213	0,98168	1,56426
6_out	10,28564	10,13088	0,04605	1,00000	0,78367	0,99291	1,56734
6_in	7,84482	7,75811	0,03425	1,00000	0,78102	0,98560	1,56205
7_out	10,35754	10,16692	0,04387	1,00000	0,78277	0,98882	1,56554
7_in	7,90922	7,78781	0,03290	1,00000	0,78091	0,98710	1,56182
8_out	10,30490	10,18766	0,04749	1,00000	0,78390	0,98498	1,56779
8_in	7,92968	7,64708	0,09250	1,00000	0,78953	0,99177	1,57906
9_out	10,25919	10,13677	0,06379	1,00000	0,78048	0,97897	1,56095
9_in	7,83935	7,71570	0,06327	1,00000	0,77968	0,97127	1,55936
10_out	10,26884	10,16952	0,04751	1,00000	0,78124	0,98559	1,56247
10_in	7,87940	7,79260	0,02756	1,00000	0,78078	0,98657	1,56157

	PodczechShapes Ellipse []	PodczechShapes Elongation []	Convexity □	Center dim0 [mm]	Center dim1 [mm]	SolidArea [mm]	Size [mm ²]	ConvexPerimeter [mm]
1_out	0,99827	3,10690	0,99218	11,90168	12,13526	285.293,00000	331,71450	64,80660
1_in	1,00120	3,14926	0,99331	11,90441	12,15805	174.809,00000	203,25308	50,68841
2_out	0,99837	3,11266	0,99643	11,90350	12,09616	281.767,00000	327,61477	64,32138
2_in	0,99998	3,27224	0,93823	11,56125	12,35811	183.610,00000	213,48265	52,37250
3_out	0,99881	3,21999	0,96809	11,98754	12,04979	286.183,00000	332,74932	64,93958
3_in	1,00018	3,21577	0,96541	11,98631	12,06773	167.311,00000	194,52690	49,66765
4_out	0,98908	3,42101	0,91105	12,10575	11,99562	285.313,00000	331,73775	65,02879
4_in	0,99099	3,21414	0,96085	12,18259	11,96512	169.974,00000	197,39997	50,12042
5_out	0,99378	3,21126	0,97186	12,08109	12,05612	286.343,00000	332,93535	65,03560
5_in	0,99584	3,17989	0,98103	12,06555	12,05192	167.515,00000	194,76874	49,64903
6_out	0,99780	3,31629	0,93762	12,01975	11,98530	285.794,00000	332,29702	64,91872
6_in	0,99443	3,26616	0,95620	12,01730	12,00868	166.268,00000	193,30256	49,50258
7_out	0,99665	3,27433	0,95500	12,03046	12,11822	289.926,00000	337,10136	65,34002
7_in	0,99428	3,31347	0,94396	12,04043	12,11927	169.043,00000	196,52793	49,92556
8_out	0,99809	3,22603	0,96639	11,95365	12,06135	286.913,00000	333,59810	65,02510
8_in	1,00526	3,20067	0,96656	12,00351	12,03011	169.875,00000	197,50694	50,07465
9_out	0,99373	3,23258	0,96443	12,00803	11,96724	284.343,00000	330,60992	64,87285
9_in	0,99272	3,27185	0,94734	11,97112	11,96999	166.012,00000	193,00723	49,48881
10_out	0,99470	3,25788	0,95963	12,04402	11,99595	284.876,00000	331,22965	64,90664
10_in	0,99412	3,21645	0,97196	12,06526	11,98545	167.754,00000	195,03501	49,70046

5.3. Rezultati trećeg mjerenja svih O-prstena

	Perimeter [mm]	Feret Max [mm]	Feret Min [mm]	Feret PerpMin [mm]	Feret MaxAng [rad]	Feret MinAng [rad]	Radius Max [mm]
1_out	65,21811	21,02389	20,30041	20,86014	-1,67321	0,28806	10,54008
1_in	51,02000	16,21047	16,00088	16,17146	2,70713	4,71239	8,11913
2_out	66,89083	20,86034	20,57256	20,70903	2,05200	0,00000	10,46102
2_in	52,02543	16,04479	15,69230	15,95440	-1,76986	3,11303	8,03015
3_out	67,60343	20,86927	20,57228	20,73487	-2,34348	2,93195	10,45964
3_in	51,69125	15,90314	15,69383	15,79618	-2,41691	0,00000	7,99827
4_out	72,89099	20,92822	20,52984	20,72284	-2,57936	-0,78540	10,48533
4_in	52,57106	16,03144	15,69383	15,93265	-2,94670	4,71239	8,03414
5_out	66,77998	20,88577	20,52640	20,80536	-2,43598	2,65164	10,46606
5_in	51,31907	15,93425	15,72439	15,92079	-2,00859	2,76109	8,00658
6_out	68,19168	20,83846	20,54973	20,82917	2,31335	0,77241	10,44043
6_in	51,62701	15,85454	15,65672	15,80146	2,99909	0,78540	7,93372
7_out	68,46373	20,93317	20,70903	20,84550	1,94101	0,00000	10,49652
7_in	52,32129	15,97496	15,82296	15,93023	2,02871	0,57638	8,00918
8_out	67,37642	20,85607	20,50433	20,77726	2,14882	0,00000	10,46690
8_in	52,59911	16,27814	15,78484	16,21436	1,83153	3,82478	8,22147
9_out	67,26895	20,83095	20,49513	20,80683	2,58994	0,82981	10,44849
9_in	52,43879	15,94725	15,64643	15,94713	2,46685	4,03373	7,98929
10_out	67,68053	20,75495	20,49785	20,68726	-2,58961	1,60782	10,42590
10_in	51,82346	15,99218	15,72794	15,83029	2,44986	4,71239	8,06403

	Radius Mean [mm]	Radius Min [mm]	Radius StdDev [mm]	Roundness □	PodczechShapes Square □	PodczechShapes Circle □	PodczechShapes Triangle □
1_out	10,28540	10,09689	0,11102	1,00000	0,78482	0,97245	1,56964
1_in	8,04722	7,94850	0,02801	1,00000	0,78602	0,99023	1,57203
2_out	10,32107	10,20580	0,03955	1,00000	0,78552	0,99357	1,57105
2_in	7,90129	7,78871	0,05239	1,00000	0,78313	0,98073	1,56626
3_out	10,30336	10,13798	0,04386	1,00000	0,78184	0,98766	1,56368
3_in	7,86802	7,72139	0,03542	1,00000	0,78451	0,99239	1,56901
4_out	10,27613	10,09352	0,06953	1,00000	0,77978	0,98360	1,55956
4_in	7,87881	7,74655	0,05554	1,00000	0,77978	0,97797	1,55957
5_out	10,30798	10,18662	0,05231	1,00000	0,78156	0,98178	1,56313
5_in	7,88859	7,76106	0,04461	1,00000	0,78087	0,98197	1,56174
6_out	10,28833	10,09072	0,04478	1,00000	0,77682	0,97581	1,55364
6_in	7,85021	7,75896	0,03342	1,00000	0,78233	0,98697	1,56466
7_out	10,36486	10,21214	0,04137	1,00000	0,78191	0,98904	1,56382
7_in	7,91460	7,81019	0,03129	1,00000	0,78071	0,98734	1,56142
8_out	10,30478	10,17675	0,04771	1,00000	0,78300	0,98385	1,56599
8_in	7,96905	7,68231	0,09458	1,00000	0,77964	0,96638	1,55929
9_out	10,28211	10,14435	0,06433	1,00000	0,77874	0,97667	1,55748
9_in	7,86006	7,71374	0,05628	1,00000	0,77756	0,97136	1,55513
10_out	10,26939	10,14450	0,05137	1,00000	0,78125	0,98561	1,56250
10_in	7,88287	7,79445	0,03881	1,00000	0,78378	0,99149	1,56756

	PodczechShapes Ellipse []	PodczechShapes Elongation []	Convexity []	Center dim0 [mm]	Center dim1 [mm]	SolidArea [mm]	Size [mm ²]	ConvexPerimeter [mm]
1_out	0,99926	3,10210	0,99415	11,92485	12,17243	285.529,00000	332,34739	64,83680
1_in	1,00079	3,14735	0,99378	11,91839	12,18239	174.736,00000	203,38758	50,70268
2_out	1,00016	3,20660	0,97307	11,99611	12,09622	287.518,00000	334,66252	65,08931
2_in	0,99711	3,24251	0,95829	11,97421	12,12393	168.462,00000	196,06504	49,85527
3_out	0,99547	3,23938	0,96131	11,99812	12,05198	286.523,00000	333,50437	64,98768
3_in	0,99886	3,25038	0,96045	11,99005	12,07321	167.098,00000	194,48088	49,64682
4_out	0,99285	3,48290	0,89355	12,11259	12,00088	285.014,00000	331,74678	65,13204
4_in	0,99285	3,27925	0,94677	12,11808	11,94860	167.537,00000	194,98022	49,77295
5_out	0,99512	3,19739	0,97489	12,06124	12,02706	286.755,00000	333,77441	65,10303
5_in	0,99423	3,22068	0,97009	12,05428	12,02002	167.952,00000	195,48655	49,78412
6_out	0,98908	3,27240	0,95237	12,01116	11,97066	285.665,00000	332,50569	64,94385
6_in	0,99609	3,25629	0,95934	11,99944	11,99014	166.304,00000	193,54737	49,52779
7_out	0,99556	3,27059	0,95495	12,07047	12,13257	289.993,00000	337,54335	65,37921
7_in	0,99403	3,27521	0,95464	12,08200	12,13309	169.087,00000	196,78903	49,94775
8_out	0,99694	3,23054	0,96484	11,97257	12,06335	286.584,00000	333,57537	65,00769
8_in	0,99267	3,23127	0,95697	12,07513	11,99794	171.444,00000	199,54299	50,33552
9_out	0,99152	3,22928	0,96583	12,02368	11,99747	285.304,00000	332,08549	64,97061
9_in	0,99002	3,28827	0,94634	12,00150	11,99954	166.705,00000	194,01413	49,62498
10_out	0,99472	3,26093	0,95871	12,04423	12,01450	284.616,00000	331,28468	64,88608
10_in	0,99794	3,24055	0,95956	12,05565	12,00848	167.678,00000	195,14434	49,72782

5.4. Rezultati četvrtog mjerenja svih O-prstena

	Perimeter [mm]	Feret Max [mm]	Feret Min [mm]	Feret PerpMin [mm]	Feret MaxAng [rad]	Feret MinAng [rad]	Radius Max [mm]
1_out	69,13962	21,04633	20,59667	20,92541	-1,86906	3,43305	10,55712
1_in	52,60465	16,01056	15,68857	16,00566	-1,66645	-0,15991	8,04185
2_out	66,90689	20,79488	20,52488	20,66081	2,05710	0,00000	10,43151
2_in	52,39115	15,98309	15,63471	15,96296	-1,69014	2,96174	8,01926
3_out	66,70678	20,74840	20,45692	20,59285	-2,38979	0,00000	10,39359
3_in	51,03538	15,80948	15,62457	15,79651	-2,05848	-0,32175	7,92464
4_out	71,92380	20,83205	20,43346	20,63913	-2,53594	1,63592	10,46427
4_in	51,89141	16,08694	15,59755	15,86941	-2,49556	4,71239	8,15762
5_out	66,30668	20,73983	20,41726	20,66441	-2,33882	-0,32175	10,40942
5_in	50,44060	15,90124	15,61610	15,80457	1,92431	-0,45066	8,00756
6_out	68,01823	20,74751	20,47238	20,71267	2,28084	3,92699	10,40936
6_in	51,33500	15,79329	15,59457	15,73874	2,52434	3,92699	7,91742
7_out	67,23234	20,83305	20,53935	20,73491	1,93434	0,86854	10,42561
7_in	52,72073	15,88719	15,71472	15,83486	1,86819	3,92699	7,96129
8_out	67,18398	20,77854	20,42294	20,69479	1,67730	0,00000	10,44748
8_in	51,20692	16,06327	15,60735	15,83531	2,39510	0,46365	8,11996
9_out	64,99485	20,53220	19,87923	20,25303	2,57563	4,71239	10,41009
9_in	49,84321	16,05000	15,59755	15,93737	2,74143	4,71239	8,03735
10_out	67,25707	20,72538	20,43222	20,61479	2,43279	1,61521	10,38924
10_in	51,41203	15,85699	15,68299	15,83671	2,52981	1,33971	7,94281

	Radius Mean [mm]	Radius Min [mm]	Radius StdDev [mm]	Roundness □	PodczechShapes Square □	PodczechShapes Circle □	PodczechShapes Triangle □
1_out	10,36846	10,19858	0,06735	1,00000	0,78352	0,98193	1,56704
1_in	7,90404	7,74123	0,05626	1,00000	0,78160	0,97545	1,56320
2_out	10,27965	10,12875	0,04616	1,00000	0,78294	0,99032	1,56589
2_in	7,87371	7,74083	0,05302	1,00000	0,78014	0,97287	1,56027
3_out	10,25782	10,14537	0,04005	1,00000	0,78467	0,99248	1,56935
3_in	7,83163	7,69264	0,03298	1,00000	0,78068	0,98317	1,56135
4_out	10,23477	10,03352	0,06985	1,00000	0,78071	0,98412	1,56141
4_in	7,84848	7,72102	0,05770	1,00000	0,78174	0,97829	1,56348
5_out	10,24740	10,12186	0,04742	1,00000	0,78181	0,98353	1,56363
5_in	7,84103	7,70334	0,04175	1,00000	0,78253	0,98447	1,56506
6_out	10,24989	10,09422	0,04506	1,00000	0,77816	0,97929	1,55632
6_in	7,81835	7,72508	0,03338	1,00000	0,78226	0,98688	1,56453
7_out	10,31498	10,15281	0,04466	1,00000	0,78498	0,99004	1,56996
7_in	7,87501	7,75381	0,03543	1,00000	0,78305	0,98945	1,56610
8_out	10,26261	10,13899	0,04790	1,00000	0,78285	0,98366	1,56570
8_in	7,88662	7,63204	0,08251	1,00000	0,79079	0,99237	1,58158
9_out	10,07432	9,77051	0,14508	1,00000	0,79118	0,98877	1,58237
9_in	7,89597	7,74359	0,07542	1,00000	0,78774	0,98160	1,57548
10_out	10,23759	10,12289	0,05113	1,00000	0,78159	0,98634	1,56318
10_in	7,85215	7,76328	0,02900	1,00000	0,77983	0,98328	1,55967

	PodczechShapes Ellipse []	PodczechShapes Elongation []	Convexity □	Center dim0 [mm]	Center dim1 [mm]	SolidArea [mm]	Size [mm ²]	ConvexPerimeter [mm]
1_out	0,99761	3,28512	0,94770	12,00530	12,15395	292.437,00000	337,69129	65,52396
1_in	0,99517	3,28562	0,94852	12,00570	12,17123	169.974,00000	196,26458	49,89645
2_out	0,99687	3,21747	0,97002	11,96334	12,07218	287.522,00000	332,01570	64,90105
2_in	0,99330	3,27791	0,94839	11,94766	12,09756	168.623,00000	194,70336	49,68699
3_out	0,99908	3,21503	0,96974	11,94348	11,99700	286.258,00000	330,55610	64,68816
3_in	0,99399	3,22815	0,96780	11,93187	12,01933	166.872,00000	192,68139	49,39213
4_out	0,99403	3,45256	0,90054	12,08110	11,94831	285.124,00000	329,24661	64,77058
4_in	0,99534	3,22569	0,95601	12,06459	11,90647	167.573,00000	193,49896	49,60888
5_out	0,99544	3,19707	0,97597	12,01338	11,98456	285.651,00000	329,85517	64,71346
5_in	0,99635	3,17212	0,98079	11,99496	11,98115	167.254,00000	193,13290	49,47166
6_out	0,99078	3,27838	0,95121	11,97353	11,94222	285.749,00000	329,96833	64,69951
6_in	0,99601	3,25043	0,96077	11,96588	11,96321	166.285,00000	191,99778	49,32126
7_out	0,99947	3,22720	0,96767	12,01388	12,07500	289.508,00000	334,30903	65,05866
7_in	0,99701	3,31844	0,94276	12,01610	12,07159	168.757,00000	194,85463	49,70288
8_out	0,99676	3,23333	0,96338	11,90224	12,02468	286.530,00000	330,87019	64,72351
8_in	1,00686	3,18783	0,97259	11,93815	12,00312	169.251,00000	195,44124	49,80320
9_out	1,00737	3,16551	0,97896	11,85500	12,13223	275.854,00000	318,54209	63,62729
9_in	1,00298	3,10550	0,99799	11,83786	11,94744	169.578,00000	195,82000	49,74307
10_out	0,99515	3,24516	0,96144	11,99684	11,95568	285.092,00000	329,20966	64,66344
10_in	0,99291	3,24223	0,96316	12,00669	11,94806	167.743,00000	193,68487	49,51816

5.5. Rezultati petog mjerenja svih O-prstena

	Perimeter [mm]	Feret Max [mm]	Feret Min [mm]	Feret PerpMin [mm]	Feret MaxAng [rad]	Feret MinAng [rad]	Radius Max [mm]
1_out	68,58702	21,07287	20,63086	20,92169	-1,86751	0,28950	10,56351
1_in	51,90711	16,03831	15,70691	16,03560	-1,59627	2,98894	8,03466
2_out	64,44119	20,33236	19,86117	20,12485	-2,70790	2,65469	10,22730
2_in	50,56622	16,09946	15,89694	16,06715	-2,64579	4,71239	8,06093
3_out	66,93823	20,79764	20,49242	20,62858	-2,36545	0,00000	10,40574
3_in	51,23709	15,85812	15,66339	15,81345	1,99796	3,02216	7,94364
4_out	71,50307	20,81627	20,47017	20,78478	-2,58126	2,22187	10,44849
4_in	52,32180	15,95980	15,62462	15,86290	-2,51217	4,71239	8,01251
5_out	67,09094	20,82893	20,49122	20,74176	-3,06307	-0,34739	10,44679
5_in	50,86343	15,89709	15,65604	15,87076	-2,05647	-0,39479	7,95345
6_out	68,53858	20,74802	20,48832	20,57111	3,01988	1,76244	10,41932
6_in	51,47640	15,82740	15,61061	15,80408	2,90496	3,94660	7,92973
7_out	66,90225	20,85356	20,61840	20,75537	1,95039	0,81103	10,44273
7_in	51,89506	15,95079	15,75938	15,90674	1,81215	3,94237	7,99722
8_out	67,18370	20,79948	20,45838	20,73070	2,15108	0,00000	10,44885
8_in	52,63670	16,13142	15,68049	15,90235	2,43686	0,24781	8,15984
9_out	66,70135	20,77317	20,36258	20,72072	2,58768	0,86217	10,43690
9_in	51,89568	15,93193	15,54942	15,88641	2,43181	0,78540	7,97601
10_out	64,99081	20,57650	20,40288	20,53679	2,22952	3,99073	10,33255
10_in	49,95274	16,00287	15,72674	15,93098	3,02647	1,57080	8,00493

	Radius Mean [mm]	Radius Min [mm]	Radius StdDev [mm]	Roundness □	PodczechShapes Square □	PodczechShapes Circle □	PodczechShapes Triangle □
1_out	10,38429	10,19922	0,06603	1,00000	0,78472	0,98525	1,56944
1_in	7,91547	7,77257	0,05382	1,00000	0,78140	0,97452	1,56281
2_out	10,00952	9,76786	0,07712	1,00000	0,78761	0,98968	1,57522
2_in	8,00005	7,89732	0,02735	1,00000	0,78702	0,99145	1,57404
3_out	10,27865	10,09730	0,04221	1,00000	0,78519	0,99314	1,57039
3_in	7,84901	7,71464	0,03291	1,00000	0,78140	0,98547	1,56280
4_out	10,25337	10,03964	0,06793	1,00000	0,77659	0,97381	1,55317
4_in	7,85955	7,73663	0,05303	1,00000	0,78300	0,98197	1,56599
5_out	10,28006	10,15691	0,05023	1,00000	0,78101	0,98240	1,56202
5_in	7,86492	7,73019	0,04259	1,00000	0,78210	0,98233	1,56421
6_out	10,25911	10,10177	0,04574	1,00000	0,78445	0,99477	1,56890
6_in	7,82824	7,74529	0,03285	1,00000	0,78012	0,98113	1,56025
7_out	10,33527	10,17253	0,04212	1,00000	0,78427	0,99198	1,56854
7_in	7,90225	7,80760	0,03113	1,00000	0,78261	0,98721	1,56521
8_out	10,28354	10,16116	0,04749	1,00000	0,78332	0,98426	1,56664
8_in	7,90980	7,73475	0,09820	1,00000	0,78701	0,98807	1,57402
9_out	10,24375	10,09774	0,06399	1,00000	0,78120	0,97746	1,56240
9_in	7,82663	7,68945	0,06242	1,00000	0,77871	0,97046	1,55743
10_out	10,21241	10,13137	0,03109	1,00000	0,78192	0,98907	1,56383
10_in	7,90906	7,80795	0,04661	1,00000	0,78434	0,98585	1,56868

	PodczechShapes Ellipse []	PodczechShapes Elongation []	Convexity []	Center dim0 [mm]	Center dim1 [mm]	SolidArea [mm]	Size [mm ²]	ConvexPerimeter [mm]
1_out	0,99914	3,25475	0,95647	12,04067	12,15167	292.304,00000	338,71017	65,60137
1_in	0,99491	3,23645	0,96221	12,03553	12,17027	169.863,00000	196,81190	49,94567
2_out	1,00282	3,16939	0,97997	11,82666	12,31442	271.679,00000	314,81074	63,15064
2_in	1,00206	3,14086	0,99649	11,88820	12,05464	173.478,00000	201,01936	50,38861
3_out	0,99974	3,21855	0,96850	11,95357	12,01786	286.448,00000	331,92447	64,82944
3_in	0,99491	3,23097	0,96657	11,93977	12,04012	167.033,00000	193,54652	49,52399
4_out	0,98878	3,43496	0,90740	12,10288	11,99181	285.143,00000	330,41229	64,88207
4_in	0,99694	3,27835	0,94857	12,07754	11,95144	167.485,00000	194,06680	49,63113
5_out	0,99442	3,22105	0,96802	12,05027	11,99243	286.469,00000	331,94880	64,94546
5_in	0,99581	3,19954	0,97558	12,03779	11,98802	167.710,00000	194,33216	49,62136
6_out	0,99879	3,30338	0,94480	11,99540	11,94395	285.323,00000	330,62087	64,75522
6_in	0,99328	3,25236	0,95946	11,98126	11,96240	166.117,00000	192,46539	49,38963
7_out	0,99857	3,20819	0,97398	12,03734	12,09558	289.640,00000	335,62323	65,16179
7_in	0,99644	3,25345	0,96102	12,04267	12,09695	169.319,00000	196,18385	49,87224
8_out	0,99736	3,23007	0,96576	11,93648	12,02986	286.703,00000	332,21995	64,88339
8_in	1,00205	3,26299	0,94944	11,84391	12,06223	169.541,00000	196,24643	49,97543
9_out	0,99466	3,21094	0,97041	11,99860	11,95104	284.451,00000	329,61043	64,72791
9_in	0,99149	3,25734	0,95187	11,96070	11,95030	166.020,00000	192,36110	49,39813
10_out	0,99557	3,15850	0,99011	11,91673	11,95971	282.742,00000	327,63011	64,34830
10_in	0,99865	3,12149	0,99729	11,92296	11,94682	169.587,00000	196,51062	49,81761

5.6. Rezultati šestog mjerenja svih O-prstena

	Perimeter [mm]	Feret Max [mm]	Feret Min [mm]	Feret PerpMin [mm]	Feret MaxAng [rad]	Feret MinAng [rad]	Radius Max [mm]
1_out	68,16026	21,03108	20,57516	20,85811	-1,86906	0,29706	10,55027
1_in	51,53513	15,98667	15,66806	15,98395	-1,71359	3,01724	8,00595
2_out	66,36248	20,75930	20,46332	20,64331	2,03371	3,03094	10,40436
2_in	51,96200	15,94579	15,58625	15,85791	-1,80286	0,00000	7,99125
3_out	66,90270	20,76982	20,46364	20,61150	-2,35966	0,12970	10,39152
3_in	51,25417	15,84514	15,62021	15,72208	2,00665	0,00000	7,95963
4_out	69,90619	20,76971	20,36728	20,61927	-2,53636	1,65000	10,44016
4_in	51,52413	15,92061	15,58625	15,82395	-2,51217	4,71239	7,98448
5_out	66,40835	20,74841	20,40246	20,62796	-2,33999	-0,32175	10,41811
5_in	50,47679	15,83448	15,61760	15,80305	-2,10643	-0,39479	7,93036
6_out	68,42896	20,73871	20,47606	20,51001	2,31566	4,71239	10,41838
6_in	51,19529	15,80514	15,58625	15,72208	2,71639	4,71239	7,92099
7_out	67,37990	20,82139	20,56777	20,73148	1,89616	0,81103	10,43213
7_in	52,65700	15,90006	15,70333	15,84740	1,68853	0,78540	7,96016
8_out	66,92464	20,77082	20,44210	20,67980	1,68055	0,00000	10,44430
8_in	51,93056	16,23658	15,65294	16,17234	1,71772	0,43241	8,15755
9_out	67,01914	20,72216	20,31626	20,67353	2,58768	0,86217	10,40897
9_in	51,92299	15,88900	15,52685	15,78668	2,47586	0,75093	7,94686
10_out	67,21904	20,69816	20,41918	20,61515	2,36315	1,64210	10,40382
10_in	51,43720	15,87306	15,68812	15,78999	-2,80361	4,71239	7,99701

	Radius Mean [mm]	Radius Min [mm]	Radius StdDev [mm]	Roundness □	PodczeckShapes Square □	PodczeckShapes Circle □	PodczeckShapes Triangle □
1_out	10,35602	10,16661	0,06709	1,00000	0,78496	0,98589	1,56992
1_in	7,89303	7,75702	0,05277	1,00000	0,78138	0,97522	1,56276
2_out	10,26297	10,14624	0,04075	1,00000	0,78333	0,98867	1,56667
2_in	7,85740	7,74734	0,05361	1,00000	0,78443	0,98166	1,56886
3_out	10,25396	10,08543	0,04354	1,00000	0,78316	0,99000	1,56632
3_in	7,83076	7,68063	0,03413	1,00000	0,78436	0,99221	1,56872
4_out	10,22712	10,01249	0,06793	1,00000	0,78270	0,98439	1,56541
4_in	7,83959	7,72171	0,05323	1,00000	0,78271	0,98161	1,56542
5_out	10,23919	9,97272	0,05287	1,00000	0,78258	0,98552	1,56517
5_in	7,83750	7,71188	0,04010	1,00000	0,78183	0,98378	1,56366
6_out	10,23919	10,02486	0,04777	1,00000	0,78430	0,99695	1,56860
6_in	7,81246	7,69928	0,03294	1,00000	0,78232	0,98747	1,56464
7_out	10,30624	10,02169	0,05009	1,00000	0,78277	0,98879	1,56555
7_in	7,87015	7,75518	0,03611	1,00000	0,78185	0,98643	1,56369
8_out	10,26208	10,14155	0,04719	1,00000	0,78260	0,98498	1,56520
8_in	7,94630	7,62709	0,11754	1,00000	0,78407	0,96624	1,56813
9_out	10,21537	10,07482	0,06401	1,00000	0,78036	0,97642	1,56073
9_in	7,80744	7,66555	0,06161	1,00000	0,78109	0,97815	1,56218
10_out	10,22776	10,12904	0,04402	1,00000	0,78054	0,98436	1,56107
10_in	7,85195	7,76480	0,03322	1,00000	0,78167	0,98884	1,56335

	PodczechShapes Ellipse []	PodczechShapes Elongation []	Convexity []	Center dim0 [mm]	Center dim1 [mm]	SolidArea [mm ²]	Size [mm ²]	ConvexPerimeter [mm]
1_out	0,99944	3,24093	0,96011	11,99024	12,11850	292.153,00000	336,87353	65,44104
1_in	0,99489	3,22363	0,96613	11,98544	12,13278	169.721,00000	195,68741	49,78968
2_out	0,99737	3,19676	0,97516	11,90435	12,01481	286.976,00000	330,90405	64,71390
2_in	0,99877	3,25867	0,95361	11,88357	12,04448	168.155,00000	193,88400	49,55165
3_out	0,99715	3,22115	0,96694	11,97511	11,98925	286.474,00000	330,32636	64,69096
3_in	0,99868	3,23469	0,96429	11,96505	12,00948	167.070,00000	192,62484	49,42394
4_out	0,99657	3,36578	0,92489	12,06185	11,95950	285.066,00000	328,70283	64,65538
4_in	0,99658	3,23632	0,96074	12,03457	11,91882	167.427,00000	193,04456	49,50129
5_out	0,99642	3,20065	0,97387	12,00926	12,00854	285.635,00000	329,35893	64,67334
5_in	0,99546	3,18778	0,97933	11,98108	11,99877	167.346,00000	192,96039	49,43341
6_out	0,99860	3,29958	0,94472	11,98857	11,91848	285.651,00000	329,37738	64,64632
6_in	0,99608	3,23915	0,96294	11,98657	11,93570	166.267,00000	191,70584	49,29787
7_out	0,99666	3,23609	0,96489	11,97819	12,04942	289.466,00000	333,77521	65,01449
7_in	0,99548	3,31175	0,94338	11,98009	12,04617	168.762,00000	194,56778	49,67553
8_out	0,99644	3,22205	0,96751	11,91351	12,01004	286.915,00000	330,83487	64,75049
8_in	0,99831	3,19837	0,96651	11,98171	11,87618	172.156,00000	198,48247	50,19144
9_out	0,99359	3,23418	0,96324	11,96779	11,92390	284.248,00000	327,75961	64,55530
9_in	0,99452	3,26786	0,94892	11,92730	11,92918	166.053,00000	191,45908	49,27101
10_out	0,99381	3,24758	0,96149	11,98470	11,95237	284.944,00000	328,56215	64,63042
10_in	0,99526	3,24053	0,96294	12,00625	11,94454	167.955,00000	193,63263	49,53113

5.7. Rezultati sedmog mjerenja svih O-prstena

	Perimeter [mm]	Feret Max [mm]	Feret Min [mm]	Feret PerpMin [mm]	Feret MaxAng [rad]	Feret MinAng [rad]	Radius Max [mm]
1_out	68,81075	21,07857	20,63460	20,91826	-1,81382	0,29706	10,57032
1_in	52,25811	16,07507	15,72548	16,07399	-1,68753	3,01327	8,03976
2_out	64,24430	20,68371	20,08287	20,67701	-1,67964	3,10589	10,34845
2_in	50,50628	16,12254	15,89758	15,99970	-2,26499	4,71239	8,06654
3_out	67,68601	20,82177	20,56047	20,66077	-2,35966	-0,02631	10,43438
3_in	52,05890	15,90515	15,65928	15,79545	-2,48825	0,00000	8,00305
4_out	72,12943	20,82013	20,46806	20,66899	-2,56583	1,55467	10,45464
4_in	52,28054	15,95582	15,62524	15,86353	-2,51986	4,71239	8,01085
5_out	66,76051	20,80031	20,49595	20,71347	-2,33999	2,77442	10,43617
5_in	51,17571	15,89145	15,66823	15,86438	-2,17187	2,91812	7,95612
6_out	68,45464	20,79259	20,50533	20,78399	2,31335	0,77131	10,43187
6_in	51,24226	15,80986	15,62953	15,79339	2,52910	4,26263	7,92361
7_out	68,53127	20,86214	20,65209	20,78498	1,92403	3,99073	10,45206
7_in	52,99784	15,95215	15,75954	15,87665	1,69489	3,94192	7,99197
8_out	67,62306	20,79652	20,46911	20,76772	1,68398	3,17861	10,45146
8_in	50,91832	15,91521	15,62524	15,89758	-1,61787	0,00000	7,97405
9_out	67,24612	20,78980	20,38924	20,74370	2,60539	4,00376	10,44843
9_in	52,26931	15,91801	15,57229	15,88842	2,64446	0,99289	7,96808
10_out	66,69038	20,70737	20,41244	20,65315	2,90259	0,78540	10,39847
10_in	51,39093	15,86784	15,69332	15,76141	2,82527	4,71239	7,96230

	Radius Mean [mm]	Radius Min [mm]	Radius StdDev [mm]	Roundness □	PodczeckShapes Square □	PodczeckShapes Circle □	PodczeckShapes Triangle □
1_out	10,38750	10,22385	0,06912	1,00000	0,78519	0,98617	1,57037
1_in	7,91919	7,75435	0,05571	1,00000	0,77921	0,97061	1,55841
2_out	10,17225	10,00661	0,09943	1,00000	0,78277	0,96802	1,56554
2_in	7,99053	7,89865	0,02877	1,00000	0,78843	0,99745	1,57686
3_out	10,28601	10,10919	0,04768	1,00000	0,78241	0,99136	1,56482
3_in	7,85461	7,71308	0,03609	1,00000	0,78354	0,98904	1,56708
4_out	10,25219	10,08610	0,06736	1,00000	0,78095	0,98467	1,56189
4_in	7,86150	7,74275	0,05243	1,00000	0,78317	0,98219	1,56635
5_out	10,27990	10,17932	0,04798	1,00000	0,78192	0,98512	1,56385
5_in	7,86299	7,74002	0,04201	1,00000	0,78142	0,98263	1,56283
6_out	10,26555	10,08664	0,04300	1,00000	0,77671	0,97568	1,55343
6_in	7,82896	7,70109	0,03279	1,00000	0,77996	0,98277	1,55991
7_out	10,34090	10,16945	0,04181	1,00000	0,78278	0,99029	1,56555
7_in	7,89744	7,78536	0,03359	1,00000	0,78315	0,98978	1,56630
8_out	10,28395	10,14251	0,04803	1,00000	0,78154	0,98078	1,56307
8_in	7,86821	7,72166	0,05204	1,00000	0,78286	0,97970	1,56573
9_out	10,24986	10,10173	0,06469	1,00000	0,78019	0,97639	1,56038
9_in	7,83540	7,70240	0,06071	1,00000	0,77934	0,97254	1,55868
10_out	10,24763	10,10315	0,05220	1,00000	0,78256	0,98477	1,56511
10_in	7,85746	7,77754	0,02728	1,00000	0,78413	0,99408	1,56827

	PodczechShapes Ellipse []	PodczechShapes Elongation []	Convexity □	Center dim0 [mm]	Center dim1 [mm]	SolidArea [mm]	Size [mm ²]	ConvexPerimeter [mm]
1_out	0,99973	3,26449	0,95380	12,02295	12,18150	292.460,00000	338,91791	65,63167
1_in	0,99212	3,25088	0,95620	12,02819	12,20247	169.973,00000	196,96083	49,96938
2_out	0,99666	3,10603	0,99728	11,78541	12,04947	280.492,00000	325,04877	64,06983
2_in	1,00386	3,13265	0,99656	11,76827	12,07417	173.052,00000	200,54169	50,33233
3_out	0,99619	3,25073	0,95936	11,96368	12,03662	286.804,00000	332,36344	64,93499
3_in	0,99764	3,27308	0,95216	11,96956	12,06414	167.257,00000	193,80528	49,56846
4_out	0,99433	3,46441	0,89880	12,10942	11,99351	285.095,00000	330,38296	64,82980
4_in	0,99717	3,27658	0,94936	12,08187	11,95049	167.529,00000	194,12628	49,63294
5_out	0,99558	3,20959	0,97219	12,06237	12,04014	286.456,00000	331,96016	64,90415
5_in	0,99493	3,22033	0,96967	12,04883	12,03332	167.616,00000	194,23406	49,62351
6_out	0,98894	3,29226	0,94630	11,98918	11,96038	285.647,00000	331,02149	64,77856
6_in	0,99307	3,24116	0,96390	11,97669	11,98282	166.151,00000	192,52707	49,39246
7_out	0,99666	3,28496	0,95186	12,00206	12,09198	289.950,00000	336,00919	65,23199
7_in	0,99713	3,32230	0,94070	12,01175	12,09297	169.104,00000	195,95032	49,85504
8_out	0,99508	3,25165	0,95945	11,97040	12,02806	286.688,00000	332,22901	64,88087
8_in	0,99677	3,19935	0,97409	11,93824	12,03198	167.815,00000	194,46583	49,59887
9_out	0,99337	3,23457	0,96308	11,99770	11,94947	284.747,00000	329,97968	64,76349
9_in	0,99229	3,28366	0,94654	11,96623	11,94940	166.406,00000	192,82373	49,47483
10_out	0,99638	3,22061	0,97005	12,01024	11,98641	284.689,00000	329,91131	64,69276
10_in	0,99839	3,23868	0,96417	12,01564	11,97467	167.374,00000	193,95477	49,54935

5.8. Rezultati osmog mjerenja svih O-prstena

	Perimeter [mm]	Feret Max [mm]	Feret Min [mm]	Feret PerpMin [mm]	Feret MaxAng [rad]	Feret MinAng [rad]	Radius Max [mm]
1_out	65,10748	20,96919	20,25990	20,81724	-1,67321	0,29146	10,51833
1_in	50,86646	16,15600	15,99327	16,12939	2,64309	4,71239	8,08587
2_out	66,86814	20,85198	20,54010	20,68316	2,05269	2,71183	10,44658
2_in	52,09525	15,99407	15,68143	15,93182	-1,87102	3,19418	8,01458
3_out	67,79673	20,81497	20,55306	20,62112	-2,34348	0,00000	10,43212
3_in	52,00447	15,86198	15,65299	15,75508	2,00950	0,00000	7,95195
4_out	72,34703	20,86065	20,53103	20,74685	-2,53594	2,36630	10,47257
4_in	52,66472	15,94313	15,63123	15,93433	-2,93959	1,82812	7,98934
5_out	66,92143	20,81363	20,47181	20,72908	-2,28793	2,80121	10,45008
5_in	50,52305	15,87967	15,65251	15,86026	-2,10350	-0,44442	7,95297
6_out	67,84686	20,82150	20,50985	20,74658	2,73498	1,24905	10,44027
6_in	51,45191	15,81772	15,61870	15,77695	2,55061	3,98575	7,92106
7_out	68,45362	20,85376	20,67147	20,78612	1,92403	1,00148	10,44650
7_in	52,11717	15,94575	15,75497	15,87315	1,69489	3,94453	7,97452
8_out	67,00102	20,81817	20,48146	20,75664	1,68218	3,16381	10,46089
8_in	51,23556	15,94099	15,67748	15,90751	-1,64986	0,06443	7,97773
9_out	67,44961	20,78817	20,38417	20,73159	2,10111	4,00789	10,43116
9_in	52,30265	15,91744	15,56084	15,88155	2,63491	4,11010	7,97455
10_out	66,44095	20,71739	20,43294	20,59651	2,36200	1,49033	10,40597
10_in	50,63224	15,93834	15,72105	15,82313	2,42116	4,71239	8,03599

	Radius Mean [mm]	Radius Min [mm]	Radius StdDev [mm]	Roundness □	PodczechShapes Square □	PodczechShapes Circle □	PodczechShapes Triangle □
1_out	10,26773	10,07086	0,11164	1,00000	0,78526	0,97305	1,57051
1_in	8,02547	7,93889	0,02607	1,00000	0,78424	0,99010	1,56848
2_out	10,29371	10,15119	0,04336	1,00000	0,78356	0,99076	1,56712
2_in	7,88364	7,77131	0,05131	1,00000	0,78140	0,97927	1,56280
3_out	10,28239	10,11411	0,04377	1,00000	0,78368	0,99452	1,56736
3_in	7,84859	7,70221	0,03403	1,00000	0,78470	0,99264	1,56941
4_out	10,25525	10,08426	0,07173	1,00000	0,77591	0,97764	1,55182
4_in	7,85978	7,73577	0,05205	1,00000	0,77914	0,97316	1,55827
5_out	10,27744	10,14714	0,04990	1,00000	0,78180	0,98307	1,56361
5_in	7,85992	7,74141	0,04113	1,00000	0,78178	0,98235	1,56355
6_out	10,26488	10,10726	0,04438	1,00000	0,77787	0,97911	1,55574
6_in	7,82737	7,73557	0,03211	1,00000	0,78096	0,98438	1,56192
7_out	10,33787	10,18477	0,04256	1,00000	0,78145	0,98949	1,56290
7_in	7,89276	7,79589	0,03253	1,00000	0,78261	0,98903	1,56522
8_out	10,29411	10,18240	0,04635	1,00000	0,78304	0,98378	1,56609
8_in	7,87959	7,72038	0,04835	1,00000	0,78206	0,98135	1,56411
9_out	10,25411	10,10911	0,06575	1,00000	0,78147	0,97832	1,56294
9_in	7,83422	7,70376	0,05906	1,00000	0,77994	0,97300	1,55989
10_out	10,24747	10,15412	0,03866	1,00000	0,78377	0,99001	1,56755
10_in	7,86789	7,78540	0,03389	1,00000	0,78168	0,98885	1,56337

	PodczechShapes Ellipse []	PodczechShapes Elongation []	Convexity □	Center dim0 [mm]	Center dim1 [mm]	SolidArea [mm]	Size [mm ²]	ConvexPerimeter [mm]
1_out	0,99982	3,10491	0,99399	11,90338	12,13421	286.018,00000	331,18634	64,71635
1_in	0,99853	3,14846	0,99410	11,90426	12,14569	174.713,00000	202,30391	50,56652
2_out	0,99766	3,20680	0,97134	11,96937	12,06151	287.484,00000	332,88386	64,95156
2_in	0,99491	3,25716	0,95507	11,95591	12,08971	168.612,00000	195,21975	49,75471
3_out	0,99781	3,25711	0,95748	11,96533	12,02261	286.845,00000	332,14394	64,91373
3_in	0,99911	3,27856	0,95231	11,95881	12,04272	167.143,00000	193,51876	49,52462
4_out	0,98792	3,46811	0,89845	12,07896	11,98202	285.427,00000	330,50201	65,00012
4_in	0,99203	3,30329	0,94267	12,06664	11,93624	167.611,00000	194,06183	49,64542
5_out	0,99542	3,21527	0,96994	12,03519	11,97816	286.520,00000	331,76762	64,90954
5_in	0,99539	3,18162	0,98122	12,02148	11,97405	167.614,00000	194,07804	49,57399
6_out	0,99041	3,25850	0,95482	12,00411	11,95372	285.849,00000	330,99065	64,78139
6_in	0,99435	3,25280	0,95990	12,00663	11,97871	166.207,00000	192,44074	49,38852
7_out	0,99498	3,28255	0,95273	12,01048	12,08884	289.980,00000	335,77403	65,21802
7_in	0,99645	3,26841	0,95590	12,01849	12,09105	169.040,00000	195,71534	49,81889
8_out	0,99700	3,21839	0,96919	11,97633	12,04474	287.491,00000	332,89196	64,93667
8_in	0,99575	3,21408	0,97009	11,95036	12,04562	168.445,00000	195,03680	49,70288
9_out	0,99500	3,24462	0,96097	11,99152	11,95834	285.206,00000	330,24611	64,81692
9_in	0,99305	3,28587	0,94536	11,96312	11,96004	166.475,00000	192,74758	49,44481
10_out	0,99793	3,20701	0,97331	11,99716	11,95588	284.863,00000	329,84894	64,66756
10_in	0,99527	3,17676	0,98008	12,02315	11,94879	167.933,00000	194,44857	49,62387

5.9. Rezultati devetog mjerenja svih O-prstena

	Perimeter [mm]	Feret Max [mm]	Feret Min [mm]	Feret PerpMin [mm]	Feret MaxAng [rad]	Feret MinAng [rad]	Radius Max [mm]
1_out	65,16145	20,87438	20,24787	20,71773	-1,68127	0,28605	10,47426
1_in	50,25754	15,96928	15,85578	15,93554	-2,25712	3,01724	8,00335
2_out	65,07978	20,58271	19,82864	20,55131	-1,65476	2,71497	10,39985
2_in	49,86094	15,90167	15,77187	15,80572	-2,15467	0,00000	7,95630
3_out	66,43064	20,67791	20,37482	20,51020	-2,36314	0,00000	10,37045
3_in	50,96493	15,76232	15,57081	15,74041	2,00755	2,83871	7,89719
4_out	69,96832	20,70952	20,31843	20,60561	-2,46736	-0,78540	10,39448
4_in	51,64681	16,12752	15,54988	16,09162	-2,54428	-1,08246	8,09721
5_out	66,33530	20,65016	20,38366	20,58454	-3,05791	2,75571	10,36784
5_in	50,33277	15,76886	15,56143	15,76736	-1,81577	2,88282	7,90716
6_out	67,90823	20,67245	20,39022	20,65348	2,31335	0,78540	10,36847
6_in	51,39988	15,74454	15,52921	15,69768	2,90287	3,97457	7,87970
7_out	65,22836	20,66333	20,30713	20,44251	1,91845	0,00000	10,36125
7_in	53,92412	16,40446	15,81874	15,90436	-1,67621	-1,24905	8,37850
8_out	67,02517	20,69878	20,38321	20,66546	1,67893	0,04996	10,40480
8_in	50,67478	15,84278	15,53496	15,80572	-1,63921	0,00000	7,93199
9_out	67,13276	20,67929	20,27361	20,62206	2,09157	0,86418	10,37571
9_in	51,62502	15,82292	15,49640	15,74789	2,59750	0,76412	7,92146
10_out	66,68151	20,57222	20,33832	20,53123	2,47045	1,60782	10,32609
10_in	51,09629	15,77525	15,61134	15,76476	2,75670	1,29250	7,91154

	Radius Mean [mm]	Radius Min [mm]	Radius StdDev [mm]	Roundness []	PodczechShapes Square []	PodczechShapes Circle []	PodczechShapes Triangle []
1_out	10,24021	10,07628	0,09425	1,00000	0,78530	0,97720	1,57060
1_in	7,94099	7,85665	0,02196	1,00000	0,78392	0,99312	1,56784
2_out	10,10869	9,76116	0,16331	1,00000	0,78840	0,96853	1,57681
2_in	7,90125	7,82900	0,02319	1,00000	0,78669	0,99951	1,57339
3_out	10,22115	10,07232	0,04228	1,00000	0,78533	0,99332	1,57067
3_in	7,80371	7,66784	0,03289	1,00000	0,78056	0,98313	1,56112
4_out	10,19422	10,02896	0,06703	1,00000	0,78018	0,97951	1,56036
4_in	7,88251	7,62682	0,10811	1,00000	0,77852	0,95787	1,55703
5_out	10,20971	10,08991	0,04719	1,00000	0,78035	0,98388	1,56070
5_in	7,81181	7,68226	0,03933	1,00000	0,78130	0,98179	1,56261
6_out	10,20737	10,02136	0,04197	1,00000	0,77718	0,97693	1,55437
6_in	7,78477	7,66114	0,03238	1,00000	0,78080	0,98347	1,56160
7_out	10,20418	9,83378	0,06623	1,00000	0,78792	0,99657	1,57585
7_in	7,98661	7,71538	0,15028	1,00000	0,79567	1,00762	1,59133
8_out	10,23404	10,11953	0,04680	1,00000	0,78113	0,98099	1,56227
8_in	7,83042	7,68398	0,04930	1,00000	0,78447	0,98171	1,56894
9_out	10,19573	10,05967	0,06424	1,00000	0,78093	0,97751	1,56186
9_in	7,79027	7,65767	0,05954	1,00000	0,78101	0,97853	1,56202
10_out	10,19364	10,08035	0,05093	1,00000	0,78175	0,98600	1,56349
10_in	7,81775	7,73473	0,02770	1,00000	0,78007	0,98354	1,56013

	PodczechShapes Ellipse []	PodczechShapes Elongation []	Convexity []	Center dim0 [mm]	Center dim1 [mm]	SolidArea [mm]	Size [mm ²]	ConvexPerimeter [mm]
1_out	0,99988	3,12160	0,99094	11,88633	12,06135	287.583,00000	329,42607	64,57123
1_in	0,99812	3,14714	0,99563	11,87812	12,07223	172.915,00000	198,07398	50,03796
2_out	1,00383	3,16187	0,98127	11,93025	11,92219	280.470,00000	321,27813	63,86074
2_in	1,00165	3,13558	0,99798	11,82120	12,02547	171.202,00000	196,11174	49,76014
3_out	0,99992	3,21264	0,97064	11,88504	11,94797	286.499,00000	328,18435	64,48035
3_in	0,99384	3,23334	0,96597	11,87644	11,97232	167.017,00000	191,30866	49,23048
4_out	0,99336	3,37856	0,92167	12,03566	11,90395	285.152,00000	326,64136	64,48790
4_in	0,99124	3,20240	0,96419	12,12477	11,86981	170.365,00000	194,80243	49,79717
5_out	0,99357	3,21234	0,97254	11,97533	11,96376	285.837,00000	327,42603	64,51368
5_in	0,99479	3,19191	0,97924	11,96251	11,95713	167.359,00000	191,70271	49,28771
6_out	0,98954	3,28496	0,94858	11,91967	11,86437	285.723,00000	327,29429	64,41644
6_in	0,99414	3,26462	0,95547	11,90790	11,88457	166.179,00000	190,33728	49,11091
7_out	1,00322	3,15672	0,98606	11,89766	11,99999	285.544,00000	327,09039	64,31889
7_in	1,01307	3,28716	0,93703	11,94321	12,15723	174.773,00000	200,17940	50,52873
8_out	0,99457	3,23812	0,96343	11,88976	11,97734	287.242,00000	329,03545	64,57434
8_in	0,99882	3,19860	0,97428	11,86211	11,97972	168.158,00000	192,62026	49,37150
9_out	0,99431	3,24638	0,96009	11,93026	11,88578	285.024,00000	326,49474	64,45355
9_in	0,99441	3,26267	0,95263	11,89426	11,88373	166.398,00000	190,59387	49,17960
10_out	0,99535	3,24134	0,96549	11,95525	11,91917	284.971,00000	326,43402	64,38051
10_in	0,99321	3,23902	0,96494	11,96957	11,91422	167.606,00000	191,98107	49,30462

Prilog 6.: Tolerancije mjerenih veličina O-prstena opće namjene [7]

(800) 775-6525
sales@marcorubber.com
www.marcorubber.com

O-Ring Tolerances International General Purpose

O-Ring Tolerances International General Purpose

Dynamic O-Ring Gland Width and Depth and Diameter Default Recommendations

O-RING TOLERANCES CROSS-SECTION GENERAL PURPOSE INTERNATIONAL (Millimeters)			
O-Ring Cross Section (mm)	Tolerance (mm)	O-Ring Cross Section (Inches)	Tolerance (Inches)
0.00 to 1.79	+/- 0.07	0.00 to 0.069	+/- 0.003
1.80 to 2.64	+/- 0.08	0.070 to 0.102	+/- 0.003
2.65 to 3.54	+/- 0.09	0.103 to 0.138	+/- 0.004
3.55 to 5.29	+/- 0.10	0.139 to 0.209	+/- 0.004
5.30 to 6.99	+/- 0.13	0.210 to 0.274	+/- 0.005
7.00 to 8.00	+/- 0.15	0.275 to 0.314	+/- 0.006
8.01 to 10.00	+/- 0.20	0.315 to 0.394	+/- 0.008
10.01 to 15.00	+/- 0.25	0.395 to 0.590	+/- 0.010
15.01 to 25.00	+/- 0.35	0.591 to 0.984	+/- 0.014
25.01 to 100.00	+/- 0.45	0.985 to 3.937	+/- 0.018

O-RING TOLERANCES INSIDE DIAMETER INTERNATIONAL GENERAL PURPOSE (Millimeters)					
O-Ring Inside Diameter (mm)	Tolerance (mm)	O-Ring Inside Diameter (mm)	Tolerance (mm)	O-Ring Inside Diameter (mm)	Tolerance (mm)
0.00 to 1.79	+/- 0.12	100.01 to 103.00	+/- 0.85	307.01 to 311.00	+/- 2.21
1.80 to 2.80	+/- 0.13	103.01 to 106.00	+/- 0.87	311.01 to 315.00	+/- 2.24
2.81 to 4.00	+/- 0.14	106.01 to 109.00	+/- 0.89	315.01 to 320.00	+/- 2.27
4.01 to 5.30	+/- 0.15	109.01 to 112.00	+/- 0.91	320.01 to 325.00	+/- 2.30
5.31 to 7.10	+/- 0.16	112.01 to 115.00	+/- 0.93	325.01 to 330.00	+/- 2.33
7.11 to 8.50	+/- 0.17	115.01 to 118.00	+/- 0.95	330.01 to 335.00	+/- 2.36
8.51 to 9.75	+/- 0.18	118.01 to 122.00	+/- 0.97	335.01 to 340.00	+/- 2.40
9.76 to 11.80	+/- 0.19	122.01 to 125.00	+/- 0.99	340.01 to 345.00	+/- 2.43
11.81 to 13.20	+/- 0.21	125.01 to 128.00	+/- 1.01	345.01 to 350.00	+/- 2.46
13.21 to 15.00	+/- 0.22	128.01 to 132.00	+/- 1.04	350.01 to 355.00	+/- 2.49
15.01 to 16.00	+/- 0.23	132.01 to 136.00	+/- 1.07	355.01 to 360.00	+/- 2.52
16.01 to 17.00	+/- 0.24	136.01 to 140.00	+/- 1.09	360.01 to 365.00	+/- 2.56
17.01 to 19.00	+/- 0.25	140.01 to 142.50	+/- 1.11	365.01 to 370.00	+/- 2.59
19.01 to 20.60	+/- 0.26	142.51 to 145.00	+/- 1.13	370.01 to 375.00	+/- 2.62
20.61 to 21.20	+/- 0.27	145.01 to 147.50	+/- 1.14	375.01 to 379.00	+/- 2.64
21.21 to 22.40	+/- 0.28	147.51 to 150.00	+/- 1.16	379.01 to 383.00	+/- 2.67
22.41 to 23.60	+/- 0.29	150.01 to 152.50	+/- 1.18	383.01 to 387.00	+/- 2.70
23.61 to 25.00	+/- 0.30	152.51 to 155.00	+/- 1.19	387.01 to 391.00	+/- 2.72
25.01 to 26.50	+/- 0.31	155.01 to 157.50	+/- 1.21	391.01 to 395.00	+/- 2.75
26.51 to 28.00	+/- 0.32	157.51 to 160.00	+/- 1.23	395.01 to 400.00	+/- 2.78
28.01 to 29.00	+/- 0.33	160.01 to 162.50	+/- 1.24	400.01 to 406.00	+/- 2.82
29.01 to 30.00	+/- 0.34	162.51 to 165.00	+/- 1.25	406.01 to 412.00	+/- 2.85
30.01 to 31.50	+/- 0.35	165.01 to 167.50	+/- 1.28	412.01 to 418.00	+/- 2.89
31.51 to 33.50	+/- 0.36	167.51 to 170.00	+/- 1.29	418.01 to 425.00	+/- 2.93
33.51 to 34.50	+/- 0.37	170.01 to 172.50	+/- 1.31	425.01 to 429.00	+/- 2.96

(800) 775-6525

Prilog 7.: Nacrt objektiva Edmund Optics žarišne duljine 16 mm [14]

