UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET
ODSJEK ZA AUTOMATIKU I ELEKTRONIKU

Brojanje objekata od interesa na pokretnoj traci korištenjem industrijske kamere

ZAVRŠNI RAD

- PRVI CIKLUS STUDIJA -

Student:
Amar Begović

Mentor:
Doc.dr Emir Sokić, dipl.ing.el.

Sarajevo,
juli 2018.

Doc.dr Emir Sokić, dipl.ing.el.
Odsjek za automatiku i elektroniku
Sarajevo, 09.07.2018.

Postavka zadatka završnog rada I ciklusa:
Brojanje objekata od interesa na pokretnoj traci korištenjem industrijske kamere

Vizuelna inspekcija je veoma važan dio kontrole kvaliteta proizvoda u industrijskom okruženju. Jedan od najvažnijih parametara procesa je kvantifikacija količine (broja proizvoda) koji se kreću pokretnom trakom u određenom vremenu. Ukoliko specifičnosti proizvodnog procesa ne dozvoljavaju brojanje objekata na uobičajen način, tada se korištenje industrijske kamere i obrada slike nameću kao jedino moguće rješenje. U okviru završnog rada potrebno je implementirati sistem koji omogućava automatsko brojanje objekata od interesa koji prolaze pokretnom trakom u odgovarajućem vremenskom periodu.

Koncept i metode rješavanja:
Rad se sastoji iz sljedećih cjelina:

- pregled literature,
- analize načina rada industrijske kamere Sony XC-55,
- analize metoda za izdvajanje objekata od interesa i njihovo automatsko labeliranje, brojanje objekata na frejmu i korelaciju izbrojanih objekata od interesa kroz frejmove sa ili bez referentnih tačaka/objekata, te sa ili bez promjene međusobne dispozicije objekata,
- usporedba pristupa ukoliko se koristi interno (kamera) odnosno eksterno (senzori) hardversko ili softversko trigerovanje,
- implementacija odgovarajućih algoritama u C++/OpenCV okruženju,
- integracija sistema u model ETFcam v.1.0.,
- analize eksperimentalnih rezultata na sintetiziranom skupu uzoraka (npr. kružni i pravougaoni uzorci).

Polazna literatura:

[1] Batchelor, Bruce G., and Paul F. Whelan. Intelligent vision systems for industry. Springer Science & Business Media, 2012.

[2] Torras, Carme, ed. Computer vision: theory and industrial applications. Springer Science & Business Media, 2012.

[3] Demant, Christian, C. Demant, and Bernd Streicher-Abel. Industrial image processing. Springer-Verlag, 1999.

[4] Szeliski, Richard. Computer vision: algorithms and applications. Springer Science & Business Media, 2010.

[5] Sonka, Milan, Vaclav Hlavac, and Roger Boyle. Image processing, analysis, and machine vision. Cengage Learning, 2014.

[6] Russ, John C. The image processing handbook. CRC press, 2016.

[7] Costa, Luciano da Fontoura Da, and Roberto Marcondes Cesar Jr. Shape analysis and classification: theory and practice. CRC Press, Inc., 2000.

[8] Costa, Corrado, et al. "Shape analysis of agricultural products: a review of recent research advances and potential application to computer vision." Food and Bioprocess Technology 4.5 (2011): 673-692.

Doc.dr Emir Sokić, dipl.ing.el.

Univerzitet u Sarajevu
Elektrotehnički fakultet
Odsjek za automatiku i elektroniku

Izjava o autentičnosti radova

Završni rad
I ciklus studija

Ime i prezime: Amar Begović
Naslov rada: Brojanje objekata od interesa na pokretnoj traci korištenjem industrijske kamere
Vrsta rada: Završni rad I ciklusa studija
Broj stranica: 58

Potvrđujem:

- da sam pročitao dokumente koji se odnose na plagijarizam, kako je to definirano Statutom Univerziteta u Sarajevu, Etičkim kodeksom Univerziteta u Sarajevu i pravilima studiranja koja se odnose na I i II ciklus studija, integrirani studijski program I i II ciklusa i III ciklus studija na Univerzitetu u Sarajevu, kao i uputama o plagijarizmu navedenim na web stranici Univerziteta u Sarajevu;
- da sam svjestan univerzitetskih disciplinskih pravila koja se tiču plagijarizma;
- da je rad koji predajem potpuno moj, samostalni rad, osim u dijelovima gdje je to naznačeno;
- da rad nije predat, u cjelini ili djelimično, za stjecanje zvanja na Univerzitetu u Sarajevu ili nekoj drugoj visokoškolskoj ustanovi;
- da sam jasno naznačio prisustvo citiranog ili parafraziranog materijala i da sam se referirao na sve izvore;
- da sam dosljedno naveo korištene i citirane izvore ili bibliografiju po nekom od preporučenih stilova citiranja, sa navođenjem potpune reference koja obuhvata potpuni bibliografski opis korištenog i citiranog izvora;
- da sam odgovarajuće naznačio svaku pomoć koju sam dobio pored pomoći mentora i akademskih tutora/ica.

Sarajevo, 09. juli 2018.

_____

Amar Begović

Sažetak

Računarska vizija je sredstvo putem kojeg se rješavaju problemi koji zahtijevaju nekonvecionalna rješenja bez korištenja senzora. U ovom radu prikazan je sistem računarske vizije koji za cilj ima brojanje objekata od interesa. U tu svrhu prvo su analizirani načini rada industrijske kamere *Sony XC-55*, potom su analizirane metode za izdvajanje objekata od interesa. Ekstrakcija značajki je također analizirana, jer pomoću značajki će biti vršena korelacija između frejmova. Informacija od interesa predstavlja broj objekata koji se kreću pokretnom trakom i prolaze kroz vidljivo polje industrijske kamere. Cilj je ekstraktovati tu informaciju i napraviti korelaciju kroz frejmove. Aplikacija koja omogućava automatsko brojanje i labeliranje objekata od interesa je razvijena pomoću programskog jezika *C++* uz korištenje *OpenCV* biblioteka. Proces kojim se opisuje izdvajanje objekata i njihovo brojanje se može podijeliti u korake: segmentacija slike, morfološka obrada slike, opisivanje objekata i ekstrakcija značajki, korelacija objekata kroz frejmove, brojanje i labeliranje objekata od interesa. Eksperimentalni rezultati su pokazali da razvijeni algoritam radi uspješno pri relativno malim brzinama pokretne trake, kao i da algoritam ima izvršavanje u realnom vremenu.

Abstract

Computer vision is tool for solving problems which require unconventional solutions without usage of sensors. In this paper, computer vision sistem is shown which has aim to count objects of interest. In that purpose, first thing analysed are modes of work of industrial camere *Sony XC-55.*, after that methods for segregation objects of interest. Extraction of features is also analysed, because features are going to be used for correlation between frames. Information od interest represents number of objects on moving conveyor which are passing through visible field of industrial camera. Aim is to extract that information and create correlation through frames. Application which enables automatic counting and labeling objects of interest is developed within programming language *C++* with usage of *OpenCV* libraries. Process which describes segregation of objects and their counting can be divided in steps: image segmentation, morphological image processing, segregation of objects and extraction of features, correlation of objects through frames, counting and labeling objects of interest. Experimental results have shown that developed algorithm works successfully under relatively small moving conveyor velocities, and that algorithm has performance in real time.

Sadržaj

Popis slika

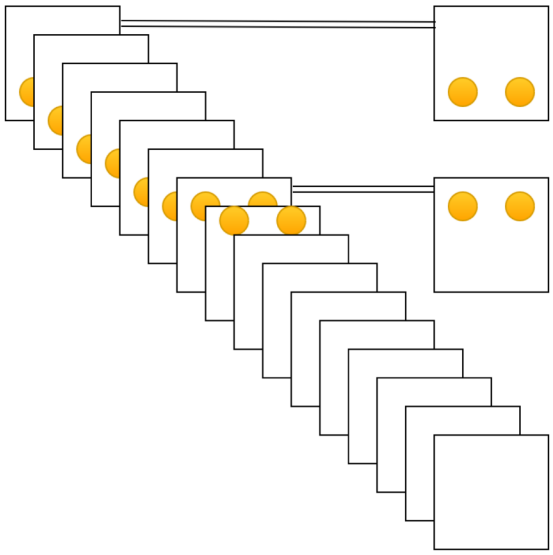
Popis tabela

Poglavlje 1

Uvod

1.1 Obrazloženje teme

U današnjem svijetu potreba za vizuelnom inspekcijom postaje sve veća zbog svojih mogućnosti. Računarska vizija, iako relativno nova nauka, mora da bude u stalnom porastu u smislu broja inženjera koji je izučavaju i doprinosa koje donose. Sa svojom sve većom popularnosti, polako postaje neizbježni faktor u industrijskim aplikacijama, ponajviše zbog toga što iz jednog signala, koji predstavlja slika, postoji mogućnost da se ekstraktuje veliki broj informacija. U ovom radu informacija koja se želi ekstraktovati je broj objekata koji se kreću pokretnom trakom. Ulazni signal predstavlja trenutni frejm koji sadrži informaciju o dijelu pokretne trake koji pokriva kamera u trenutku akvizicije slike. Iz tog signala cilj je ekstraktovati informaciju o postojanju objekata na vidljivom dijelu pokretne trake. Tu već postaje jasno zbog čega se izučavana nauka naziva računarska vizija, jer taj dio trake koji je proglašen vidljivim, postaje uistinu vidljiv i računaru kroz signal koji se šalje od strane kamere. Nakon akvizicije slike i slanja iste računaru, ona se obrađuje i pokušava se ekstraktovati informacija od interesa. U daljem toku rada biće detaljno pojašnjeno kako se ekstraktuje informacija od interesa, u ovom slučaju informacija o broju objekata u vidljivom polju kamere. Naravno, nije dovoljno ekstraktovati informaciju o broju objekata na svakom frejmu zasebno, nego je potrebno napraviti korelaciju između frejmova i identifikovati objekte kroz frejmove. Prikaz promjene pozicije objekata u vidljivom polju kamere je dat na slici 1.1.



Slika 1.1: Promjena pozicije objekata kroz frejmove

1.2 Pregled korištenih metoda obrade slike

U ovom potpoglavlju bit će dat uvid u važnost izbora adekvatne kamere. Nakon toga će biti pobrojane i ukratko objašnjene različite metode korištene za izdvajanje objekata od interesa, kao i metode za opisivanje izdvojenih objekata. Također će biti pobrojani osnovni koraci na primjeru aplikacije za automatsko brojanje i labeliranje objekata od interesa na pokretnoj traci korištenjem industrijske kamere *Sony XC-55*.

Slika koja predstavlja izlazni signal kamere je određenog kvaliteta i gubitak kvaliteta teško može biti softverski nadoknađen [2]. Jako bitno odrediti kakav kvalitet slike je od interesa i odabrati adekvatnu kameru, jer u suprotnom može doći do realizacije sistema koji je u startu problematičan zbog neadekvatnog izbora kamere. Naime, prilikom vršenja eksperimenata opisanih u radu, korištena je industrijska kamera *Sony XC-55*. Navedena kamera posjeduje interne trigere koji mogu biti rješenje problema sa akvizicijom slike objekata koji se kreću relativno brzo ili relativno sporo.

Metode za izdvajanje objekata od interesa mogu se grubo podijeliti na segmentaciju i morfološku obradu slike. Analizirane metode segmentacije su:

- segmentacija pragom (eng. *thresholding*),
- segmentacija bazirana na ivicama (eng. *edge-based segmentation*),
- segmentacija bazirana na regionima (eng. *region-based segmentation*) [3].

Svaka od ovih metoda se bavi izdvajanjem objekata na slici, samo u drugačijoj reprezentaciji. Segmentacija pragom je najstariji metod segmentacije koji sliku u prostoru sivih tonova transformiše u binarnu sliku na osnovu praga koji određuje vrijednost svakog piksela. Segmentacija bazirana na ivicima i segmentacija bazirana na regionima isto tako vraćaju binarne slike, s tim što prva od navedenih segmentaciju vrši na osnovu diskontinuiteta u boji ili teksturi, dok druga segmentacija umjesto detekcije granica među regionima, direktno konstruiše regione.

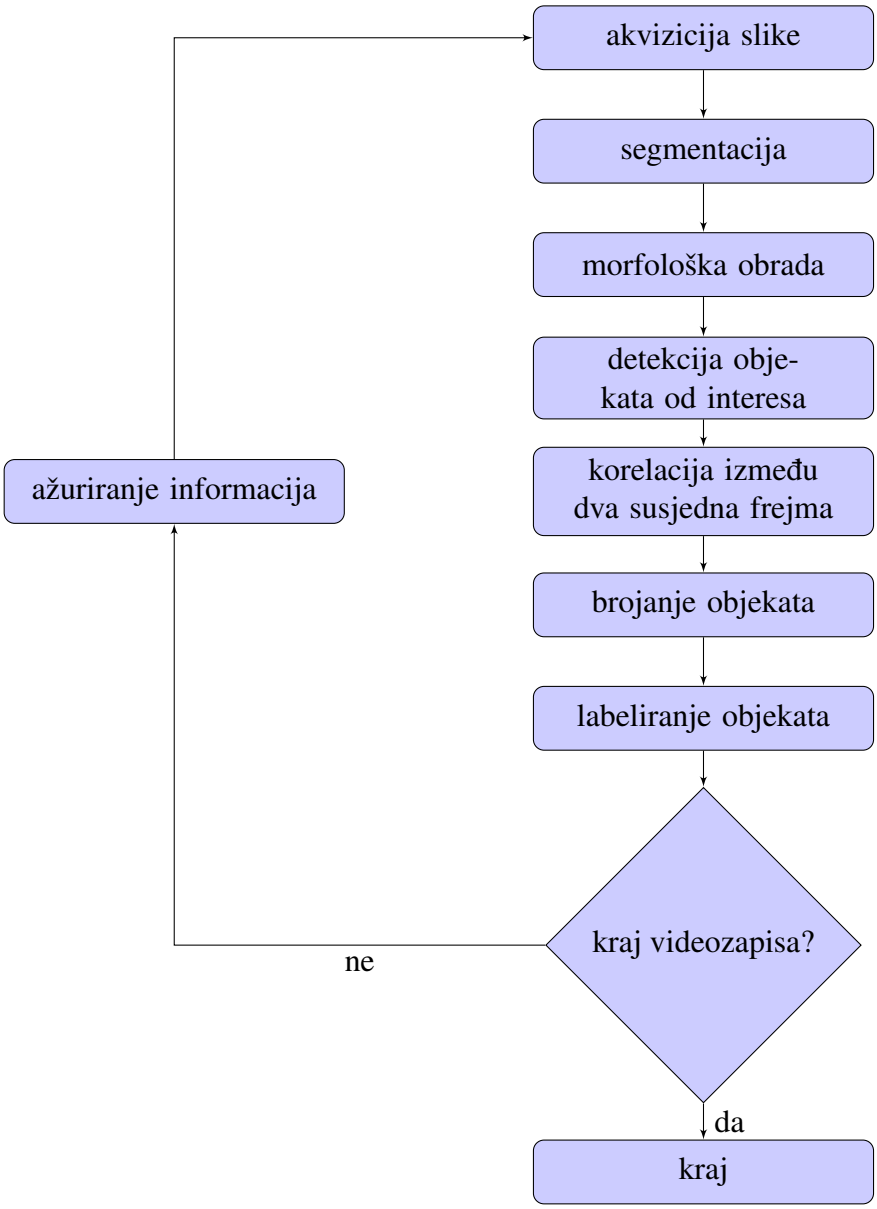
Za razliku od segmentacije koja se uglavnom primijenjuje na slike u prostoru sivih tonova, morfološka obrada slike kao ulazni parametar prima binarnu sliku i ima za cilj modifikovati sliku tako da bude pogodna za opisivanje izdvojenih objekata. Morfološka obrada slike se može podijeliti na osnovne morfološke operacije:

- erozija (eng. *erode*),
- dilatacija (eng. *dilate*),
- otvaranje (eng. *opening*),
- zatvaranje (eng. *closing*) [4].

Nakon uspješnog izdvajanja objekata od interesa na slici potrebno je te iste objekte opisati, kako bi se mogla napraviti korelacija među frejmovima u kojoj će objekti zadržati svoju jedinstvenost. Opisivanje izdvojenih objekata se vrši pomoću lančanog koda (eng. *chain code*), koji objekat opisuje konturom, odnosno sekvencom susjednih piksela u unaprijed određenom smjeru.

Nakon opisivanja objekata od interesa određuju se značajke samih objekata. Značajke koje će biti od interesa u ovom radu su površina konture i centar mase (centroid) konture. Na osnovu centra mase vrši se korelacija objekata kroz frejmove, dok površina služi za provjeravanje da li detektovane konture odgovaraju unaprijed poznatim objektima od interesa. Također površina konture će biti ključna informacija u dijelu rada u kojem je cilj očuvanje stvarnog oblika objekta od interesa prilikom ulaska/izlaska iz vidljivog polja kamere.

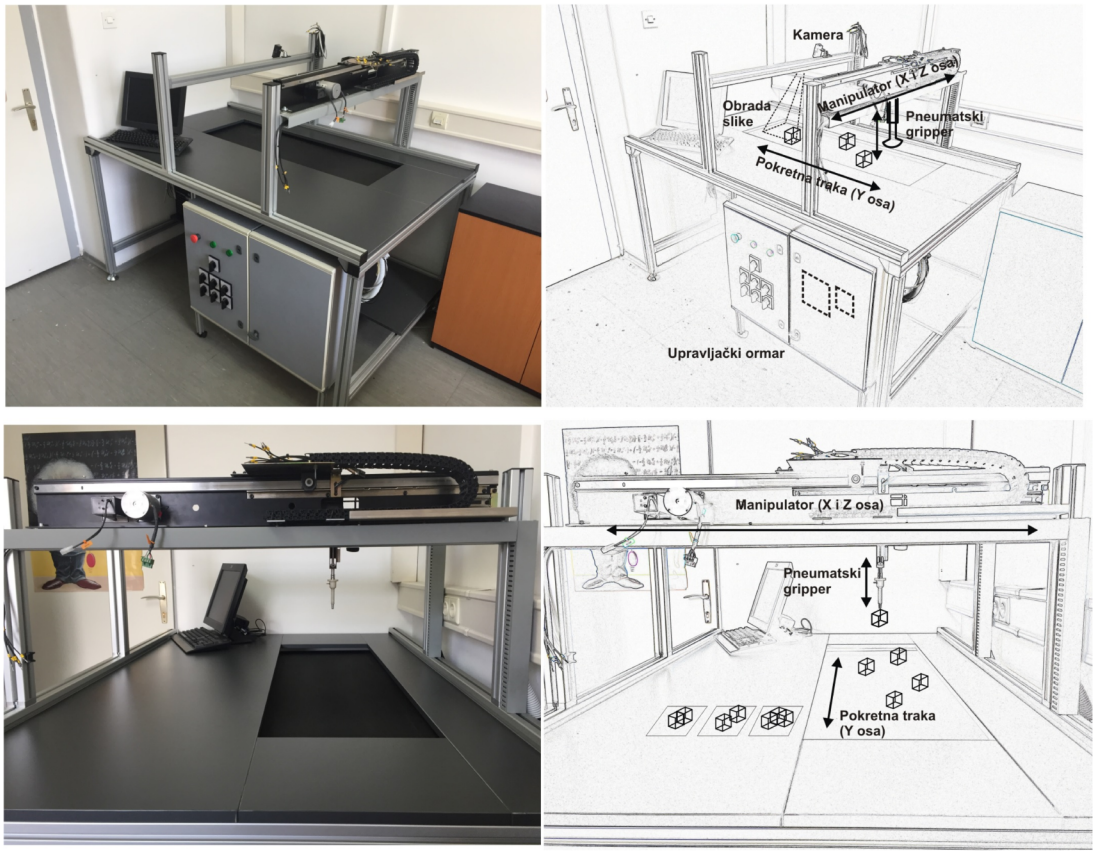
Koraci koji će biti korišteni na razvijenoj aplikaciji za automatsko brojanje i labeliranje objekata od interesa su predstavljeni na dijagramu 1.2.



Slika 1.2: Dijagram aktivnosti razvijene aplikacije za brojanje i labeliranje objekata na video zapisu

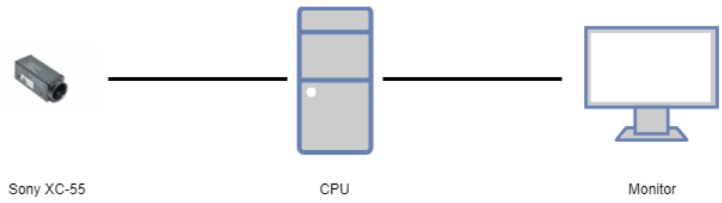
1.3 Eksperimentalna postavka

Razvijena aplikacija biće integrisana u model *ETFCam v.1.0.* i testiranje će biti vršeno na tom modelu. Sam izgled modela je prikazan na slici 1.3 zajedno sa funkcionalnostima koje posjeduje.



Slika 1.3: Izgled modela *ETFCam v1.0.*

Slanje podataka funkcioniše tako da se izlazni signal sa kamere šalje na računar na kojem se dalje vrši analiza i obrada tog signala pomoću programskog jezika *C++* zajedno sa bibliotekama koje nudi *OpenCV*, a koje su konkretno namijenjene za analizu i obradu slike. Nakon ekstrakcije željene informacije, prikaz rezultata se može vidjeti na monitoru koji je pozicioniran na modelu. Principijelna šema je data na slici 1.4.



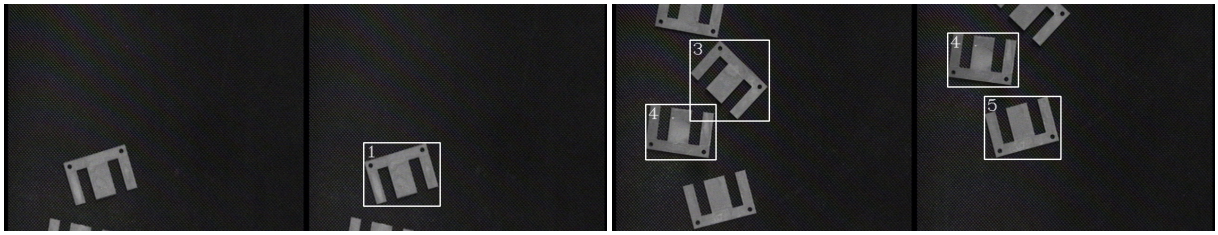
Slika 1.4: Principijelna šema povezivanja kamere i računara

Realni problemi koji će biti analizirani se mogu podijeliti u tri testna scenarija:

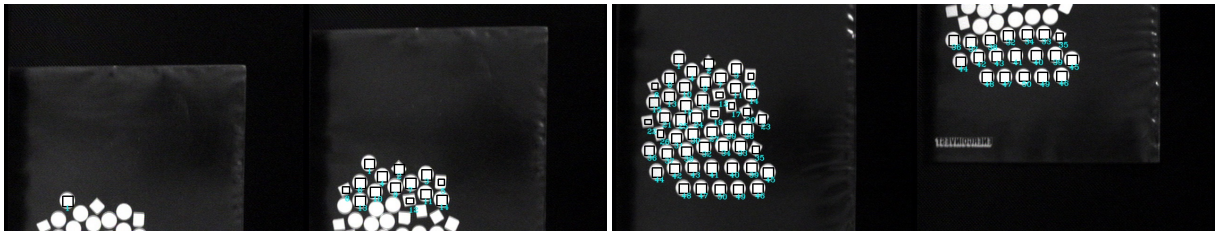
- Testni scenarij 1 - Objekti koji se kreću trakom, a imaju izgled slova 'E',
- Testni scenarij 2 - Objekti kružnog i pravougaonog oblika koji imaju relativno malu površinu i nalaze se na relativno maloj udaljenosti,

- Testni scenarij 3 - Objekti koji se koriste kao podloga u koje se ubacuju šarafi, a nazivaju se diblovi.

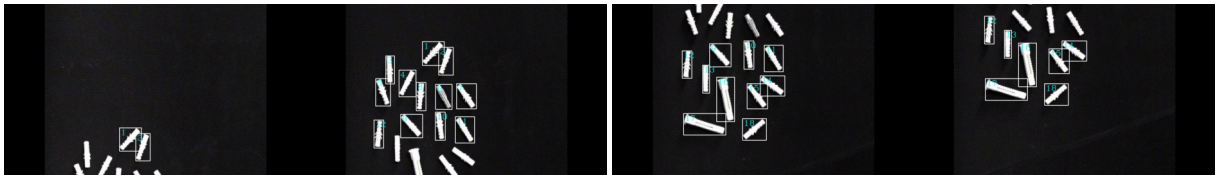
U radu će biti pojašnjeno zbog čega su baš izabrani ovi testni scenariji, kao i problemi koje objekti navedenih izgleda mogu uzrokovati. Prikaz željenog rezultata na svakom od testnih scenarija je dat na slikama 1.5, 1.6 i 1.7.



Slika 1.5: Testni scenarij 1



Slika 1.6: Testni scenarij 2



Slika 1.7: Testni scenarij 3

Poglavlje 2

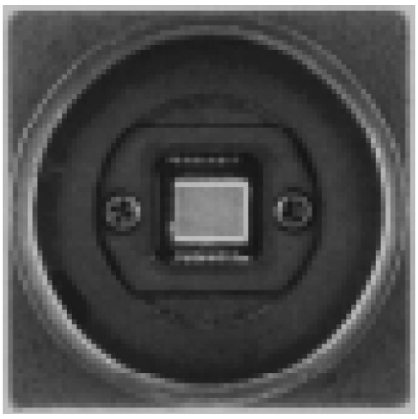
Analiza načina rada industrijske kamere *Sony XC-55*

Sony XC-55 kamera predstavlja moćan alat koji može biti iskorišten u kompjuterskoj viziji. Izlaz kamere je crno-bijela slika u *VGA* (eng. *video graphic array*) formatu. Jedna od njenih najvažnijih osobina je to što koristi progresivno skeniranje prilikom akvizicije slike. Bazirana je na *CCD* (eng. *charge-coupled device*) tehnologiji koja u suštini predstavlja senzore koji sakupljaju svjetlost i pretvaraju je u digitalne podatke pogodne za obradu. Rezolucija *CCD* senzora je obično predstavljena njegovom veličinom u inčima, tj. kod kamere *Sony XC-55* ta veličina iznosi $1/3$" (jedna trećina inča). Velika prednost ove kamere je što nudi slike sa kvadratnim pikselima, pogodnim za kasniju obradu. Veličina jednog piksela iznosi $7.4\mu m \times 7.4\mu m$ i nema potrebe za dimenzionalnom korekcijom jer je horizontalna rezolucija piksela jednaka vertikalnoj. Postoje dvije postavke pri progresivnom skeniranju slike:

- *1I* postavka, *2:1 interlace*
  Omogućava 2:1 prepletanje koje sve parne i neparne piksele isprepliće svako 1/60 sekundi i generiše kao izlazne signale.

- *1N* postavka, *Non-interlace*
  Prepliće piksele neovisno o parnosti svako 1/30 sekundi i generiše kao izlazni signal. Pri ovoj postavci, omogućeno je uključenje trigera *E-DONPISHA II* koji služi za preciznu akviziciju slike objekata koji se kreću velikim brzinama.

Težina kamere iznosi $110g$, dok su dimenzije kamere: $29mm \times 29mm \times 67mm$, a sam prikaz objektiva kamere se nalazi na slici 2.1.



Slika 2.1: Objektiv kamere *Sony XC-55*

Postoji više pristupa trigerovanju industrijske kamere *Sony XC-55* koji se globalno mogu podijeliti na interno i eksterno. U ovom radu pristup koji će biti korišten predstavlja interno trigerovanje u normalnom režimu rada tj. rad sa normalnom akvizicijom slika (eng. *normal shutter*).

2.1 Interno trigerovanje

Pri postavci *1N* kamere *Sony XC-55*, interni triger se može postaviti u dva režima rada:

- *normalni režim*; omogućava postavljanje brzine trigerovanja između 1/100 i 1/8000 sekundi; izlaz, koji je posljedica rada sa internim trigerom, predstavlja kontinualan signal.
- *E-DONPISHA II*; omogućava postavljanje brzine trigerovanja između 1/4 i 1/100000 sekundi; koristi se za akviziciju slike objekata koji se kreću velikim brzinama i omogućava da se sa velikom preciznošću uhvati njihova pozicija na slici, odnosno omogućava akviziciju što kvalitetnije slike.

Pristup trigerovanja u normalnom režimu će biti korišten uz pretpostavku da objekti koji se kreću trakom imaju relativno male brzine kretanja, odnosno nema potrebe da zaustavljenjem brojanja i labeliranja jer je traka uvijek u pokretu i objekti prolaze kroz vidljivo polje kamere.

2.2 Eksterno trigerovanje

Za ovaj pristup koristi se opcija **Restart Reset** na kameri *Sony XC-55* koja je namijenjena za eksterno trigerovanje i ima veliku primjenu u industriji u slučajevima kada nije potrebno uzimati slike objekata koji se kreću velikim brzinama, odnosno nema potrebe za velikom količinom slika, jer se iz manjeg broja mogu izvući potrebne informacije.

Pristup eksternog trigerovanja se tipično koristi pomoću senzora za detekciju objekata, uz pretpostavku da je frekvencija pojavljivanja objekata na pokretnoj traci relativno mala. Pristup objedinjuje sisteme u kojima ili je objekata koje je potrebno brojati malo ili pokretna traka nije uvijek u funkciji tj. nema potrebe za akvizicijom slika jer se objekti ne kreću trakom.

Još jedan od korisnih atributa je pojačanje koje se može koristiti na tri načina, sa ograničenjem od $0dB$ do $18dB$:

- *AGC* - automatski regulira pojačanje u odnosu na svjetlost u svrhu prikupljanja što kvalitetnije slike,
- *F* - fiksirano pojačanje od $0dB$,
- *M* - pojačanje koje se ručno mijenja pomoću pojačivača na samoj kameri. [5]

* * *

Iako postoje dva načina internog trigerovanja industrijske kamere *Sony XC-55*, u ovom radu biće obrađen samo normalni režim rada, dok režim rada naziva *E-DONPISHA II* neće bti razmatran zbog nemogućnosti korištenja istog na primjeru industrijske kamere *Sony XC-55* koji posjeduje *Elektrotehnički fakultet u Sarajevu*. Pristup eksternog trigerovanja isto tako neće biti obrađen jer rezultat razvijenog algoritma ni u kakvom smislu neće biti ugrožen korištenjem eksternog trigera u odnosu na interni u normalnom režimu. Smjernica za budući rad svakako obuhvata rad sa eksternim trigerom u cilju što veće industrijske primjene sa razvijenim algoritmom.

Poglavlje 3

Analiza metoda za izdvajanje objekata od interesa

U ovom poglavlju će biti analizirane metode za izdvajanje objekata od interesa, od kojih je polazna segmentacija. Segmentacija slike je jedan od najvažnijih koraka pri analizi slike i ekstrakciji informacija iz iste, odnosno lociranja objekata od interesa koji se pojavljuju na slici. U ovom poglavlju biće obrađena segmentacija slike na načine:

- segmentacije pragom (eng. *thresholding*),
- segmentacije bazirane na ivicama (eng. *edge-based segmentation*),
- segmentacije bazirane na regionima (eng. *region-based segmentation*).

Naknadno u ovom poglavlju će biti dat osvrt na morfološku obradu slike i na osnovne metode koje se koriste u istoj. Morfološka obrada može biti ključna pri izdvajanju objekata od interesa, jer se u njoj može postići slika adekvatna za kasniju identifikaciju i opisivanje kontura.

3.1 Segmentacija pragom

Segmentacija pragom sivih tonova (eng. *gray-level thresholding*) na slikama jedan je od najstarijih segmentacijskih procesa. Veliki broj objekata na slikama je okarakterisan konstantnom svjetlošću koju apsorbuje i reflektuje. Iz toga slijedi da postoji razlika između objekta od interesa i podloge na kojoj se nalazi. Ta razlika je ključna u segmentaciji i pokušava se dodatno naglasiti u cilju pronalaska kontura koje opisuju objekte od interesa.
Kompletna segmentacija slike R predstavlja formiranje konačnog skupa regija $R_1, ..., R_s, ...$

$$R = \bigcup_{i=1}^{s} R_i, \qquad R_i \neq R_j, \qquad i \neq j \tag{3.1}$$

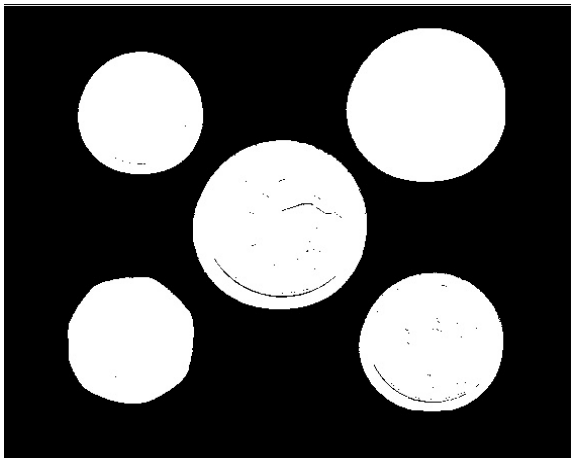
Segmentacija pragom je transformacija ulazne slike f koja daje izlaznu (binarnu sliku) g po funkciji:

$$g(i,j) = \begin{cases} 1 & \text{za} \quad f(i,j) \geq T, \\ 0 & \text{za} \quad f(i,j) < T \end{cases} \tag{3.2}$$

gdje T označava prag, $g(i,j) = 1$ objekat od interesa na slici, $g(i,j) = 0$ pozadinu na slici ili obratno [3].

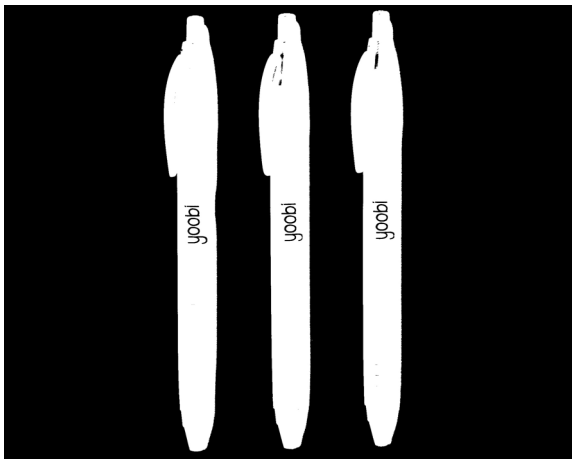
Segmentacija pragom daje dobre rezultate u slučaju da se objekti ne dodiruju i da je slika vrlo dobre kvalitete, odnosno vrlo jasno se razdvajaju objekti od podloge na crno-bijeloj slici. Rezultat segmentacije pragom prikazan je na slikama 3.2 i 3.4.

Slika 3.1: Primjer 1 - Originalna slika[1]



Slika 3.2: Primjer 1 - Rezultat segmentacije pragom

Slika 3.3: Primjer 2 - Originalna slika[2]



Slika 3.4: Primjer 2 - Rezultat segmentacije pragom

Rezultat segmentacije pragom na prikazan na slikama 3.2 i 3.4 otvara mogućnost pristupanja opisivanju kontura jer iz ovog rezultata bi relativno jednostavno bilo prebrojati objekte, no međutim veliki problem predstavljaju objekti koji se međusobno preklapaju, odnosno dodiruju i ova metoda bez dodatne obrade slike ne daje potrebne rezultate za labeliranje i prebrojavanje objekata.
Postoje razne varijacije segmentacije pragom poput: adaptivne segmentacije pragom (eng. *adaptive thresholding*), polu-segmentacije pragom (eng. *semi-thresholding*), segmentacije pragom sa opsegom (eng. *band thresholding*) itd. Po potrebi korištenja određenog algoritma koji koristi segmentaciju pragom biće napravljen pregled i dat primjer istog.

[1]Rezolucija slike: 562×467
[2]Rezolucija slike: 2048×2048

3.2 Segmentacija bazirana na ivicama

Suština metode se bazira se na ivicama pronađenim operatorima detekcije ivica, gdje ivice predstavljaju diskontinuitete u boji, teksturi i slično. Rezultat dobiven detekcijom ivica sam po sebi ne može se koristiti kao rezultat segmentacije. Dodatni koraci moraju biti poduzeti jer kontura koja opisuje objekat dobivena detekcijom ivica može imati diskontinuitet i time gubi svojstvo zatvorene konture kojom određujemo postojanje objekta na određenom dijelu slike. Fokus će biti na metodama koje zahtijevaju minimalno poznavanje apriori informacija u cilju što opširnijeg spektra objekata koji će se moći detektovati.

3.2.1 *Canny* detekcija ivica

Jedna od najvažnijih metoda za lociranje ivica objekata na slici će biti opisana u ovom potpoglavlju. Metoda se bazira na traženju maksimuma u pravcu gradijenta i optimalna je za detekciju ivica nastalih "bijelim šumom". Optimalnost detektora ogleda se u tri kriterija:

- detekcijski kriterij; zahtijeva da važne (stvarne) ivice moraju biti locirane,
- pozicioni kriterij; zahtijeva da udaljenost između stvarne i pronađene pozicije ivice bude minimalna,
- kriterij jedinstvenosti; minimizira broj lociranja ivice i tako spriječava lažne rezultate, poput detekcije već detektovane ivice.

Algoritam koji opisuje rad metode *Canny* detekcije ivice izgleda na sljedeći način:

1. Konvolucija slike f sa *Gaussianom* promjera standardne devijacije σ.

$$F(x,\sigma) = f(x) * G(x,\sigma) \tag{3.3}$$

2. Estimacija lokalne ivice normalnih smjerova $\mathbf{n}$ za svaki piksel na slici.

$$\mathbf{n} = \frac{\nabla(G*f)}{|\nabla(G*f)|} \tag{3.4}$$

3. Lociranje pozicije ivica koristeći izraz (3.5), koji je poznat pod imenom nemaksimalna supresija (eng. *non-maximal suppression*).

$$\frac{\partial^2}{\partial \mathbf{n}^2} G*f = 0 \tag{3.5}$$

4. Izračunavanje amplitude detektovane ivice opisano je jednačinom (3.6). Uvažavajući izraz $G_n = \frac{\partial G}{\partial \mathbf{n}} = \mathbf{n}\nabla G$, kao i osobinu asocijativnosti operacija konvolucije i derivacije u izrazu (3.5), iz jednačine (3.4) slijedi:

$$|G_n * f| = |\nabla(G*f)| \tag{3.6}$$

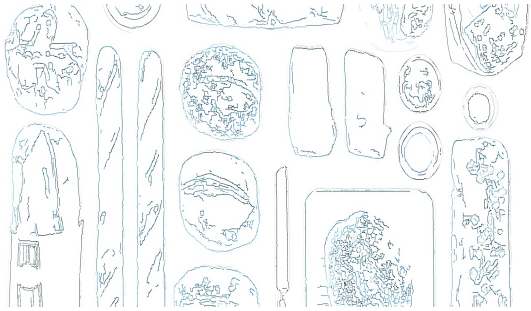
5. Segmentacija ivica na slici unaprijed određenom histerezom s ciljem eliminacije lažnih ivica. Sve ivice sa vrijednošću amplitude veće od t_1 su označene kao stvarne. Vrši se pretraga svih piksela u rasponu amplitude $[t_0, t_1]$ gdje se označavaju susjedni pikseli sa prethodno označenim pikselima koji sadržavaju ivice.

6. Ponavljanje koraka (1) do (5) sa rastućim vrijednostima standardne devijacije σ.

7. Prikupljanje finalnih informacija o detektovanim ivicama sa različitim vrijednostima standardne devijacije σ.

Canny detekcija ivica predstavlja relativno komplikovanu metodu, koja je međutim dala veoma dobre reultate i napravila veliku kontribuciju u detekciji ivica [3]. Rezultat *Canny* algoritma za detekciju ivica je prikazan na slikama 3.8 i 3.6 [3]. Postoji još veliki broj metoda koje su bazirane na segmentaciji ivica, no međutim u ovom radu neće biti obrađene.

Slika 3.5: Primjer 3 - Originalna slika[4]



Slika 3.6: Primjer 3 - Rezultat *Canny* detekcije

Slika 3.7: Primjer 4 - Originalna slika[5]

Slika 3.8: Primjer 4 - Rezultat *Canny* detekcije

3.3 Segmentacija bazirana na regionima

Cilj prethodno opisane segmentacije je bio pronalazak granica između regiona, dok u ovoj vrsti segmentacije cilj je direktno konstruisati regione. Rezultati ove dvije segmentacije nisu isti, međutim u većini slučajeva je korisno koristiti kombinaciju ove dvije segmentacije. Segmentacija bazirana na regionima je uspješnija kod slika gdje se javlja šum i samim tim je teže pronaći granice. Homogenost je važno svojstvo regiona i koristi se kao glavni kriterij za segmentaciju, gdje je glavna ideja podijeliti sliku u zone sa maksimalnom homogenosti. Kriterij homogenosti

[3] slike 3.8 i 3.6 su invertovane radi kvalitetnijeg prikaza
[4] Rezolucija slike: 1400×561
[5] Rezolucija slike: $x \times y$

može biti boja, tekstura, oblik itd. Dodatna pretpostavka potrebna u ovoj sekciji je da regioni moraju zadovoljiti sljedeće uslove:

$$H(R_i) = \text{TRUE}, \quad i = 1, 2, ..., S \tag{3.7}$$

$$H(R_i \cup R_j) = \text{FALSE}, \quad i \neq j, \quad R_i \quad \text{susjedan sa} \quad R_j \tag{3.8}$$

gdje je S ukupan broj regiona na slici, a $H(R_i)$ binarna homogena procjena regiona R_i. Rezultujući regioni moraju biti i homogeni i maksimalni, gdje se za maksimalni kaže da je takav da kriterij homogenosti neće biti tačan nakon spajanja regiona sa bilo kojim susjednim.

Tri osnovna pristupa na kojima se zasniva segmentacija bazirana na regionima su

- Spajanje (eng. *merging*),
- Razdvajanje (eng. *splitting*),
- Razdvajanje i spajanje (eng. *splitting and merging*).

Kod prvog pristupa polazi se od toga da svaki piksel predstavlja jedinstven region. Skoro pa sigurno je da ovi regioni ne zadovoljavaju jednačinu (3.8) i cilj je da se regioni spajaju sve dok jednačina (3.7) ostane zadovoljena.

Razdvajanje regiona je suprotna radnja od spajanja regiona i polazna stavka je da je cijela slika predstavljena kao jedan region koji obično ne zadovoljava uvjet (3.7). Cilj pristupa je sekvencijalno razdvajati regione dok ne zadovolje jednačine (3.7) i (3.8). Rezultat ovog pristupa je sličan prvom, ali ne daje nužno iste rezultate čak i ako je korišten isti kriterij homogenosti.

Kombinacija prethodno opisana dva pristupa rezultira sa metodom koja daje prednosti oba pristupa. Razdvoji i spoji pristup (eng. *Split-and-merge approach*) najčešće se koristi kod piramidalno i kvadratno predstavljenih reprezentacija slika. U slučaju nehomogenosti bilo kojeg regiona u bilo kojem piramidalnom nivou (isključujući najniži), on se dalje dijeli na četiri podregiona. Drugi dio koji se odnosi na spajanje kaže da ako postoje četiri regiona približno iste homogenosti u bilo kojem piramidalnom nivou, oni se spajaju u jedan region koji pripada gornjem piramidalnom nivou [3].

3.3.1 *Watershed* segmentacija

Veliki problem pri izdvajanju objekata od interesa i njihovim kasnijim prebrojavanjem predstavlja međusobno dodirivanje, odnosno preklapanje. U tim slučajevima je vrlo teško odrediti određeni diskontinuitet kojim bismo razdvojiti dva različita objekta. Rješenje za ovakve tipove problema daje *Watershed* segmentacija. *Watershed* algoritam je iterativan, tako da se slika iterativno erodira, i u svakom koraku odvojeni regioni koji su nestali u prethodnom koraku se definiraju kao ultimativne erodirajuće tačke (eng. *Ultimate eroded points (UEPs)*) i spašavaju kao rezultujuća slika, zajedno sa brojem iteracije. Spašavanje je neophodno jer će regioni biti drugačijih veličina i neće svi nestati u istom broju iteracija. Proces se nastavlja sve dok se slika potpuno ne obriše. Nakon toga, vrši se dilatacija slike na klasičan nacin, uz dodatnu logičku funkciju da nijedan novi piksel ne može biti uključen ako stvara konekciju sa prethodno odvojenim pikselom ili regionom, ili ako nije uključen na originalnoj slici. U svakoj iteraciji dilatacije, slika se poredi sa odgovarajućom iteracijom erozije preko logičke funkcije "ILI". Ovaj proces

dešava se tako da se regioni vraćaju u svoje originalne granice, samo što linije koje razdvajaju objekte se javljaju na mjestima gdje su se prethodno dodirivala dva različita objekta [6]. Na slici 3.10 prikazan je rezultat *Watershed* segmentacije.

Slika 3.9: Primjer 5 - Originalna slika[6]

Slika 3.10: Primjer 5 - Rezultat *Watershed* segmentacije

3.4 Morfološka obrada slike

Morfološke operacije su jako blisko povezane sa segmentacijom, jer su morfološke operacije najpogodnije za primjenu nakon segmentacije pragom, odnosno najpogodnije su za obradu binarnih slika. Za definisanje morfoloških operacija potrebno je iskoristiti jednačinu koja opisuje segmentaciju pragom (3.2). Ova jednačina je polazna u definisanju morfoloških operacija, jer se sve morfološke operacije definišu preko nje. Također, radi definisanja morfoloških operacija, potrebno je transformisati jednačinu (3.2) da bude funkcija dvije varijable i to: ulazne slike i praga.

$$g(f,T) = \begin{cases} 1 & \text{za} \quad f \geq T, \\ 0 & \text{za} \quad f < T \end{cases} \tag{3.9}$$

Prvi korak je konvolucija binarne slike f sa strukturnim elementom s, koji predstavlja dio slike kojim se provjerava da li se na određenoj poziciji pikseli slike slažu sa strukturnim elementom. Obično se strukturni element definiše kao matrica 3×3 [4]. Rezultat konvolucije c je također binarna slika.

$$c = g * s \tag{3.10}$$

U nastavku svaka od operacija će zasebno biti definisana na osnovu jednačina (3.9) i (3.10).

3.4.1 Erozija i dilatacija

Operacije erozije i dilatacije mogu grubo biti objašnjene kao dodavanje ili oduzimanje piksela sa binarne slike prema odgovarajućim pravilima, koja zavise od redoslijeda susjednih piksela. Erozija uklanja odgovarajuće piksele sa regiona na slici, sa ciljem uklanjanja piksela koji su nebitni sa daljnju obradu slike. Najjednostavniji primjer predstavljaju pikseli koji nisu uklonjeni

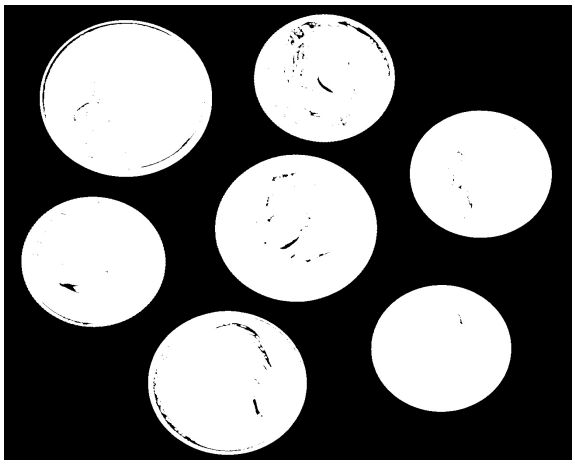
[6]Rezolucija slike: $x \times y$

korištenjem segmentacije praga jer predstavljaju određeni odsjaj kreiran svjetlošću, a u suštini ne predstavljaju nikakav objekat od interesa. Najjednostavniji oblik erozije je uklanjanje bilo kog piksela koji je susjedan sa pikselom koji predstavlja pozadinu. Ovakva radnja i te kako utiče na veličinu regiona od interesa, tj. u velikoj mjeri može smanjiti veličinu objekta na slici ili u nekim slučajevima region se može razdvojiti na dva regiona. Erozija u potpunosti može otkloniti šumove ili linijske defekte, jer oni uglavnom zauzimaju jedan do dva piksela. Komplementarna operacija eroziji je dilatacija koja služi za dodavanje odgovarajućih piksela. Klasičan način za definisanje dilatacije je analogan definiciji erozije, a to je da dilatacija dodaje bilo koji piksel koji je susjedan sa regionom od interesa. Suprotno eroziji, ovakva radnja uzrokuje povećanje u veličini regiona od interesa. Dilatacija predstavlja vrlo moćno rješenje kada je cilj zanemariti teksturu objekta od interesa koja može predstavljati problem, tako što će širenjem regiona popuniti rupe koje je prouzrokovala tekstura. Postoje razna pravila za odlučivanje koje piksele treba dodati, a koje ukloniti, kao i za formiranje kombinacije erozije i dilatacije [6]. Jednačine koje opisuju eroziju i dilataciju, izvedene iz (3.9) i (3.10), u nastavku date su kao (3.11) i (3.12) [4]. Rezultat morfoloških operacija dilatacije i erozije prikazan je na slikama 3.12 i 3.15

$$dilate(f,s) = g(c,1) \tag{3.11}$$

$$erode(f,s) = g(c,S) \tag{3.12}$$

Slika 3.11: Primjer 4 - Binarna slika nakon segmentacije pragom



Slika 3.12: Primjer 4 - Rezultat morfološke operacije dilatacije

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	0
0	0	1	1	1	1	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Slika 3.13: Prikaz piksela u binarnom zapisu prije i nakon dilatacije [1]

Slika 3.14: Primjer 4 - Binarna slika nakon segmentacije pragom

Slika 3.15: Primjer 4 - Rezultat morfološke operacije erozije

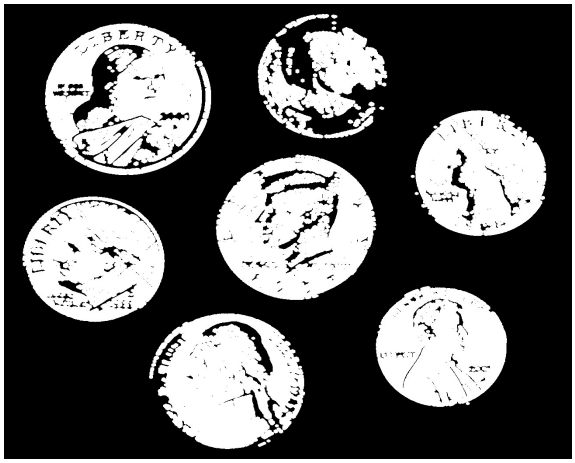
3.4.2 Morfološko otvaranje i zatvaranje

Kombinacija erozije praćena sa dilatacijom se naziva morfološko otvaranje, referirajući se na mogućnost da otvori prostor između objekata koji se dodiruju. Otvaranje predstavlja najčešće korištenu metodu sa uklanjanje šuma sa binarnih slika. Izvršavanje istih operacija u obrnutom redoslijedu (dilatacija praćena erozijom) proizvodi drugačiji rezultat. Ovakva sekvence naziva se morfološkim zatvaranjem, referirajući na mogućnost zatvaranja prekida u objektima, tj. njegovim spajanjem. Postoji nekoliko parametara koji mogu biti podešeni u operacijama erozije i dilatacije, s akcentom na formiranje obrasca kojim se tretiraju susjedni pikseli, kao i dodatna pravila za dodavanje i uklananje piksela, kao i broj iteracija izvršavanja morfoloških operacija [6]. Jednačina koja opisuje morfološko otvaranje je data kao (3.13), dok morfološko zatvaranje opisuje jednačina (3.14) [4]. Rezultat morfoloških operacija otvaranja i zatvaranja prikazan je na slikama 3.17 i 3.19.

$$open(f,s) = dilate(erode(f,s),s) \tag{3.13}$$

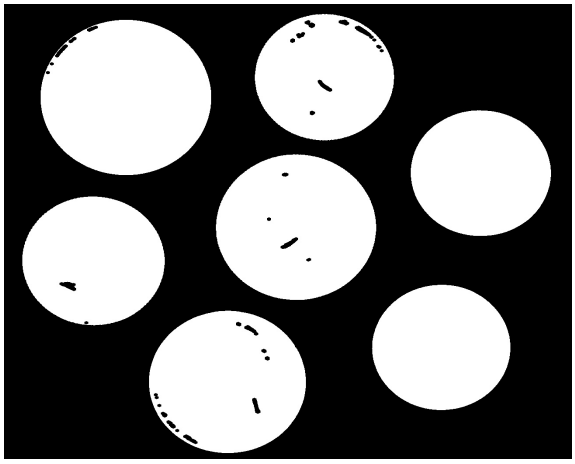
$$close(f,s) = erode(dilate(f,s),s) \tag{3.14}$$

Slika 3.16: Primjer 4 - Binarna slika nakon segmentacije pragom



Slika 3.17: Primjer 4 - Rezultat morfološke operacije otvaranja

Slika 3.18: Primjer 4 - Binarna slika nakon segmentacije pragom



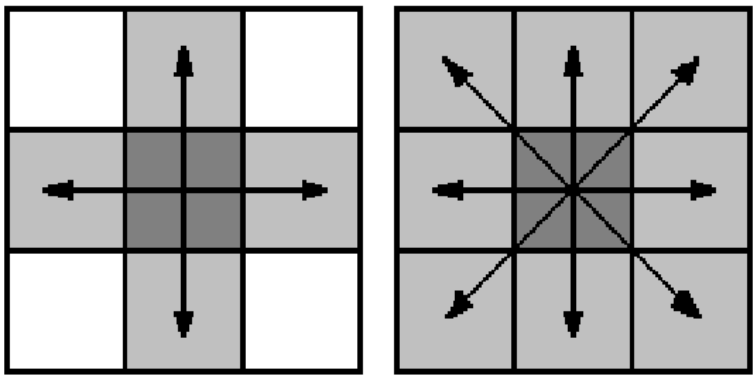
Slika 3.19: Primjer 4 - Rezultat morfološke operacije zatvaranja

3.5 Opisivanje regiona na slici

Nakon izdvajanja regiona na slici pomoću segmentacije i morfoloških operacija, potrebno ih je opisati i predstaviti u formi pogodnoj za daljnju obradu dobijenih podataka. U nastavku izdvojeni regioni od interesa će biti referirani kao konture. Jedan od jednostavnih i popularnih pristupa u opisivanju kontura je lančani kod (eng. *Chain Code*).

3.5.1 Lančani kod

Lančani kod, u literaturi još poznat kao *Freeman*-ov kod, predstavlja konturu kao sekvencu povezanu susjednim pikselima određenog smjera. Najčešće predstavljanje smjera je 4-struko ili 8-struko, što je prikazano na slici 3.20 [7].



Slika 3.20: 4-struka and 8-struka povezanost

Konvencija je da se smjerovi kodiraju 2-bitno (4-struka povezanost) i 3-bitno (8-struka povezanost), počevši od smjera sa zapadnom orijentacijom i u smjeru suprotnom kazaljci na satu. U *OpenCV*-u postoji implementirana metoda *cv::findContours* koja će biti iskorištena, a zasnovana je na 8-strukoj povezanosti. Algoritam je vrlo jednostavan, a počinje detektovanjem prvog piksela koji predstavlja dio regiona od interesa, a ujedno predstavlja njegovu ivicu. Dalje, algoritmom se prolazi kroz susjedne piksele prethodnog piksela koji također predstavljaju ivice istog regiona, sve do povratka do početnog piksela. Time je region opisan konturom i

metoda *cv::findContours* vraća niz tačaka koji opisuju određenu konturu, odnosno njene ivice bazirane na 8-strukoj povezanosti. Na osnovu stečene informacije o poziciji piksela konture, dalje postaje pogodno određivanje osobina objekata od interesa poput površine, centra mase i slično.

3.6 Pregled efikasnosti navedenih metoda

U tabeli 3.1 su data vremena izvršavanja prethodno navedenih metoda tj. brzini samih metoda segmentacije pragom, morfološkim operacijama erozije i dilatacije, kao i brzini metoda *Canny* detekcije ivica, *watershed* segmentacije i *cv::findContours* metode za detekciju kontura na binarnoj slici. Podaci su nastali primjenom algoritama nad testnim slikama 3.1, 3.3 i 3.5.

Naziv metode	**Vrijeme izvršavanja metode[ms]**		
	Primjer 1	**Primjer 2**	**Primjer 3**
	Rezolucija slike		
	562×467	2048×2048	1400×561
cv::threshold	2.69823	1.15293	0.201909
cv::erode	0.50308	3.46614	0.942071
cv:dilate	0.362942	2.34487	0.528381
cv::Canny	3.72215	3.7076	6.78234
cv::watershed	0.447598	5.96099	1.77053
cv::findContours	2.06812	4.66783	1.65752

Tabela 3.1: Prikaz brzine izvršavanja određenih *OpenCV* metoda[7]

U tabeli 3.1 se vidi da se operacije segmentacije pragom i morfološke operacije izvršavaju poprilično brzo pri čemu to vrijeme raste proporcionalno rezoluciji, dok kompleksnije operacije poput *Canny* detekcije ivica, *watershed* segmentacije i *cv::findContours* zahtjevaju više vremena za procesiranje. Operacije segmentacije pragom, dilatacije i erozije u istim testnim scenarijima odražavaju svoje vrijeme izvršavanja u skladu sa rezolucijom slike, tj. rezolucija slike predstavlja jedini parametar koji nezanemarivo utiče na vrijeme izvršavanja metode. U *OpenCV* metoda *cv::erode* i *cv::dilate* postoji parametar koji određuje broj iteracija koji određuje koliko puta će se primijeniti razmatrana morfološka operacija i ukoliko je taj parametar različit od 1, vrijeme izvršavanja metode se dodatno povećava. Metoda *cv::Canny* direktno ovisi o broju objekata koji se nalaze na analiziranoj slici i što je veći broj objekata, veći je i broj detektovanih ivica tih istih objekata i vrijeme izvršavanja se dodatno povećava. Rezultat toga je vidljiv na slici 3.6 gdje može primijetiti veliki broj detektovanih ivica. Metode *cv::watershed* i *cv::findContours* također direktno ovise o broju objekata koji se nalaze na slici, no međutim kako metoda *cv::findContours* vraća konture kao niz tačaka koji najreprezentativnije opisuju oblik samog objekta, tako može doći do problema sa objektima kružnog oblika, jer je potreban

[7]Specifikacije računara:
Procesor: Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz
RAM memorija: 3859MiB
Operativni sistem: Ubuntu 16.04
Qt: 5.10.1
OpenCV: 3.4.1

veliki broj tačaka koji će opisati isti. U tom slučaju vrijeme izvršavanja dodatno raste i može se primijetiti na primjeru 1 da je vrijeme izvršavanja metode cv::findContours veće od vremena izvršavanja iste metode na primjeru 3, iako slika 3.5 ima puno veću rezoluciju od slike 3.1.

* * *

Dalje u radu će se koristiti segmentacija pragom, erozija, dilatacija i metoda za traženje kontura koje će biti bazne metode za razvijeni algoritam koji će kasnije biti detaljno objašnjen. Metode *watershed* segmentacije i *Canny* detekcije ivica su izbjegnute, kako zbog nezadovoljavajuće brzine izvršavanja, tako i zbog ograničenja na rad sa objektima koji se ne dodiruju. Smjernica za budući rad predstavlja segmentaciju baziranu na regionima, gdje bi se u budućnosti algoritam svodio na rad sa regionima od interesa, i gdje bi se lokalno primjenjivale morfološke operacije u cilju uštede vremena obrade slike. Također planirano je poboljšanje algoritma, tako da u budućnosti bude omogućen i rad sa objektima koji se dodiruju, kao i sa objektima koji se preklapaju.

Poglavlje 4

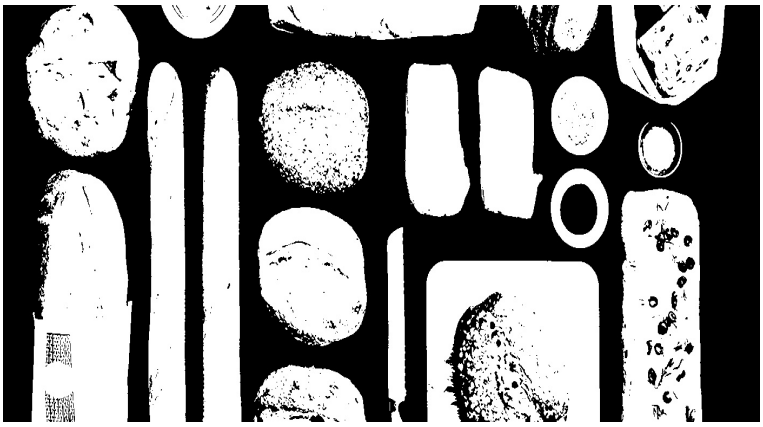
Automatsko labeliranje i brojanje objekata na frejmu

Labeliranje objekata od interesa je usko povezano sa izdvajanjem, jer izdvojene objekte treba označiti i prebrojati. Trenutni fokus u ovom poglavlju će biti labeliranje objekata i prebrojavanje na jednom frejmu, dok će u poglavlju 5 biti napravljena veza između frejmova i dat uvid u algoritam koji će se izvršavati u realnom vremenu i biti osnova brojanja objekata od interesa na pokretnoj industrijskoj traci. U ovoj fazi, kao što je navedeno, fokus je na jednom frejmu i biće dat uvid u rješenje tog problema.

Na slikama 4.1 i 4.2 prikazana originalna slika i slika u prostoru sivih tonova na primjeru 3, nad kojim će se dalje vršiti obrada i analiza. Nakon odgovarajuće segmentacije pragom, objekti su izdvojeni od svoje pozadine, ali i dalje postoji problem, jer nije jasno koje konture na slici treba labelirati i prebrojati, što je i prikazano na slici 4.3.

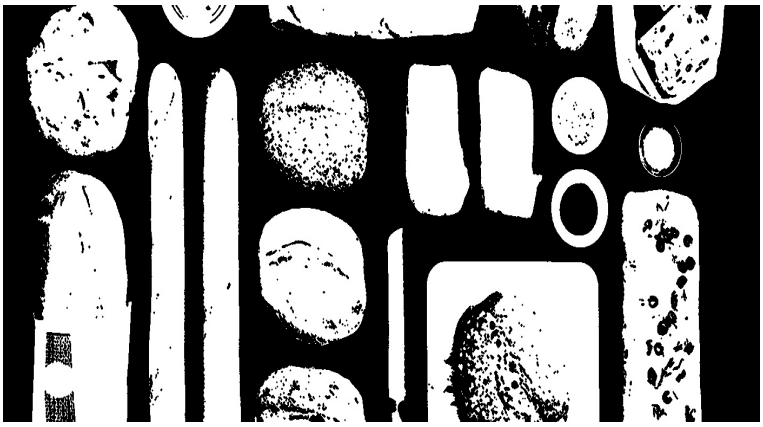
Slika 4.1: Primjer 3 - Originalna slika

Slika 4.2: Primjer 3 - Slika u prostoru sivih tonova

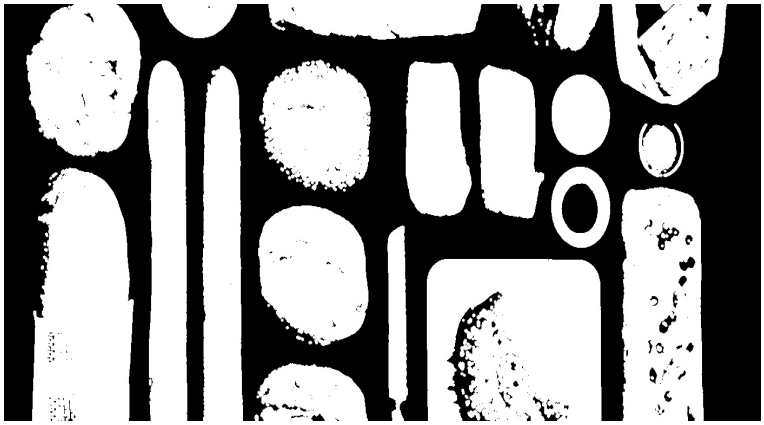


Slika 4.3: Primjer 3 - Rezultat segmentacije pragom

Rješenje za ovaj problem daju morfološke operacije, od kojih je početna u ovom algoritmu: erozija. U odnosu na rezultat koji je dala segmentacija pragom, može se primijetiti da sitniji regioni ili su nestali, ili su se spojili sa većim regionima. U nastavku je izvršena druga morfološka operacija: dilatacija, koja je dala još bolje rezultate, jer su manji regioni skoro pa nestali, odnosno proširili su se na potrebnim mjestima i spojili sa većim u cilju što boljeg izdvajanja. Rezultat ovih operacija je prikazan na slikama 4.4 i 4.5.



Slika 4.4: Primjer 3 - Rezultat erozije



Slika 4.5: Primjer 3 - Rezultat dilatacije

Nakon dovoljnog broja iteracija dilatacije dobijamo relativno dobru sliku za prebrojavanje i označavanje kontura (4.6) koje predstavljaju objekte od interesa. Glavni problem u ovom dijelu predstavljaju konture relativno male površine, koje pri pretpostavci da su unaprijed poznati objekti od interesa korisniku, u dijelu algoritma za brojanje i labeliranje će biti ignorisane, odnosne neće biti izbrojane i označene. Finalni rezultat je prikazan na slici 4.7, dok slika 4.6 također predstavlja prikaz svih kontura koje će biti prebrojane i označene uz pretpostavku da su konture relativno male površine ignorisane.

Slika 4.6: Primjer 3 - Rezultat nakon dovoljnog broja iteracija dilatacije

Slika 4.7: Primjer 3 - Labelirani i prebrojani objekti na frejmu

4.1 Značajke segmenata slike

Pozadina koja se krije iza brojanja objekata i labeliranja na frejmu direktno je povezana sa potpoglavljem Lančani kod iz poglavlja 3 i *OpenCV* metodom *cv::findContours*. Rezultat koji vraća spomenuta metoda se manifestuje kao niz kontura u kojem je svaka kontura opisana nizom tačaka koji opisuju krajnje tačke konture, odnosno ivice konture.

Da bi rezultat metode *cv::findContours* bio koristan potrebno je iz njega odrediti značajke od interesa za ovaj rad, a to su:

- površina konture (eng. *contour area*) i
- centar mase konture (eng. *center of gravity*).

Ukratko će biti opisan način određivanja spomenutih značajki, koji je poprilično intuitivan.

4.1.1 Površina konture

Površina konture A je određena jednačinom (4.1) i opisuje samu površinu u pikselima. N predstavlja broj tačaka koji opisuju konturu, dok su x_i i y_i pikseli frejma koji mogu imati vrijednost 0 ili 1.

$$A = \frac{1}{2}\left|\sum_{i=0}^{N-1}(x_i y_{i+1} - x_{i+1} y_i)\right| \tag{4.1}$$

4.1.2 Center mase konture

Pošto je određena površina konture, na osnovu te vrijednosti se određuje i centar mase konture, u literaturi još poznat pod nazivom centroid. Prema jednačini (4.2) određuju se koordinate tačke koja predstavlja center mase [8].

$$\begin{cases} g_x = \frac{1}{6A}\sum_{i=0}^{N-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \\ g_y = \frac{1}{6A}\sum_{i=0}^{N-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \end{cases} \tag{4.2}$$

Kao što je i navedeno na osnovu rezultata metode *cv::findContours* određena je površina svake od kontura, kao i centar mase kontura, koji će biti ključne osobine potrebne za povezivanje kontura kroz frejmove u narednom poglavlju. Također, određen je i pravougaonik koji opisuje 4 najdalje tačke konture i služi za labeliranje svakog od objekata. Stranice navedenog pravougaonika su paralelne stranicama frejma koji se analizira. Algoritam koji izvršava pomenute radnje napisan je kao pseudokod, a dat u algoritmu 1.

Samo labeliranje objekata je razvijeno tako da svaki registrovani objekat ima svoju metodu labeliranja i dok se određeni objekat nalazi u trenutnom frejmu, poziva se metoda koja labelira taj isti objekat. Metoda korištena za labeliranje je *cv::drawContours* koja kao parametre prima niz tačaka koje opisuju konturu i pravougaonik koji služi za označavanje objekta od interesa. Metoda je korištena tako da modificira originalnu sliku 4.1 i na njoj označi pronađene objekte od interesa.

Algoritam 1 Brojanje i labeliranje objekata na jednom frejmu

```
procedure NUMBERING AND LABELING(frame)
    kernel_er ← structuringElement(3,3)
    kernel_dil ← structuringElement(4,4)
    cvtColor(frame, grey)
    threshold(grey, tresh)
    erode(tresh, er, kernel_er)
    dilate(er, dil, kernel_dil)
    processed ← dil
    findContours(processed, contours)
    i ← 0
    while i < contours.size() do
        if contours[i].area() > minArea&contours[i].area() < maxArea then
            realcontours.pushback(contours[i])
        i ← i + 1
    i ← 0
    while i < realcontours.size() do
        boundRect[i] ← boundingRect(realcontours[i])
        drawContours(frame, boundRect[i], id)
        i ← i + 1
    return frame
```

* * *

U ovom poglavlju opisan je proces obrade slike metodama analiziranim u poglavlju 3. Kako je i naglašeno, korištene metode su bile redom: segmentacija pragom, morfološke operacije: erozija i dilatacija i metoda za detekciju kontura *cv::findContours*. Na osnovu rezultata koje vraća navedena metoda za detekciju kontura, određene su značajke konture od interesa, tj. određena je površina i centar mase kontura, na osnovu kojih će u poglavlju 5 biti izvršena korelacija između frejmova.

Poglavlje 5

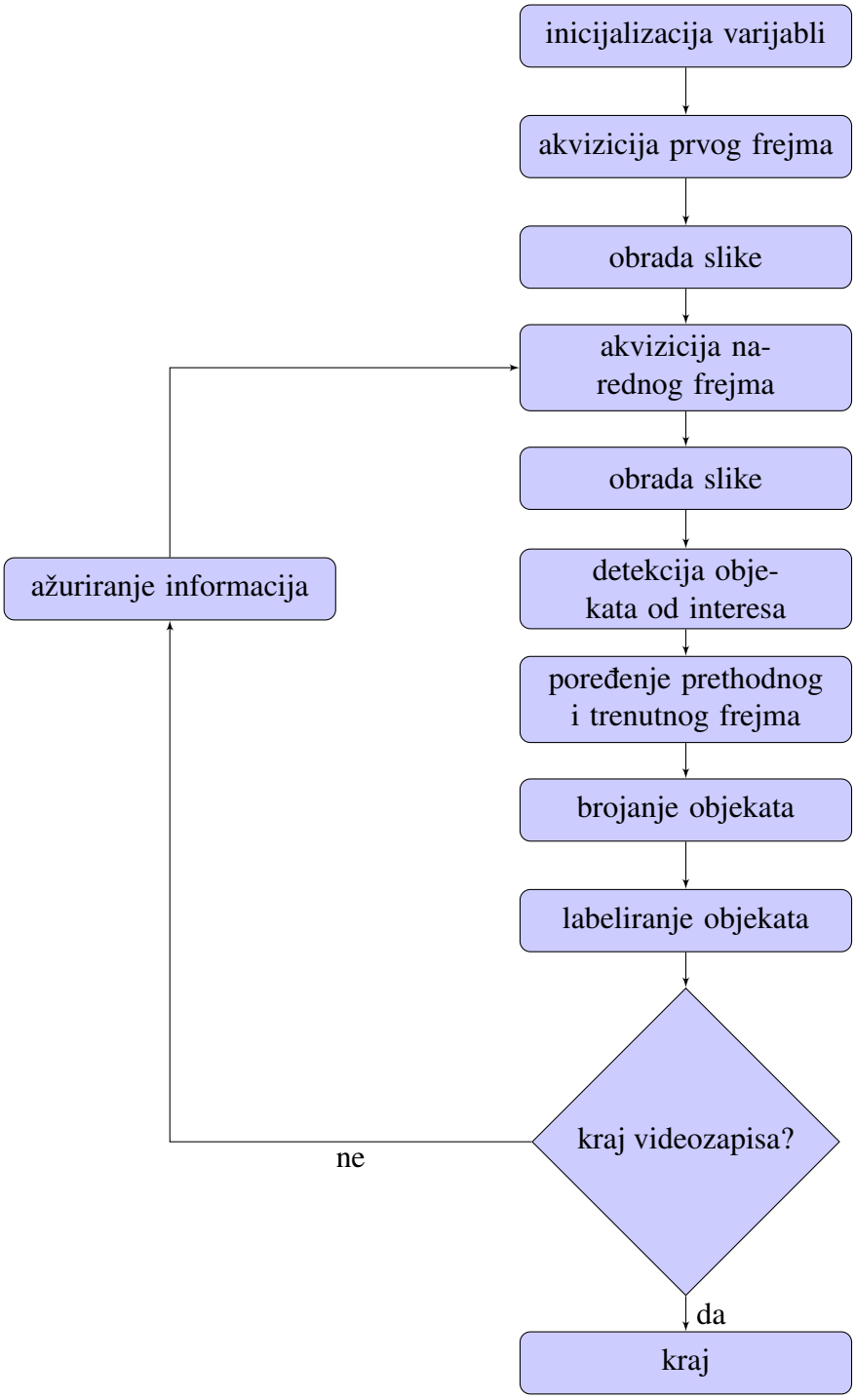
Korelacija izbrojanih objekata od interesa kroz frejmove

U ovom poglavlju će biti povezani objekti kroz sekvencu frejmova, odnosno biće riješen problem identifikacije objekata na različitim frejmovima. Potrebno je u svakom trenutku znati da li objekat sa prethodnog frejma odgovara objektu na trenutnom frejmu i da li se na trenutnom frejmu pojavio novi objekat. Također biće pojašnjeno kako se prevazišao problem nastao izgledom objekata oblika slova 'U' i 'E' [3], kako je i prikazano u Testnom scenariju 1. Korišteni pristup obuhvatio je relativno otežane uslove, poput relativno male međusobne dispozicije objekata, kao i obrade informacija bez korištenja referentnih tačaka, odnosno objekata. Na slici 5.1 dat je uvid u dijagram aktivnosti razvijenog algoritma.

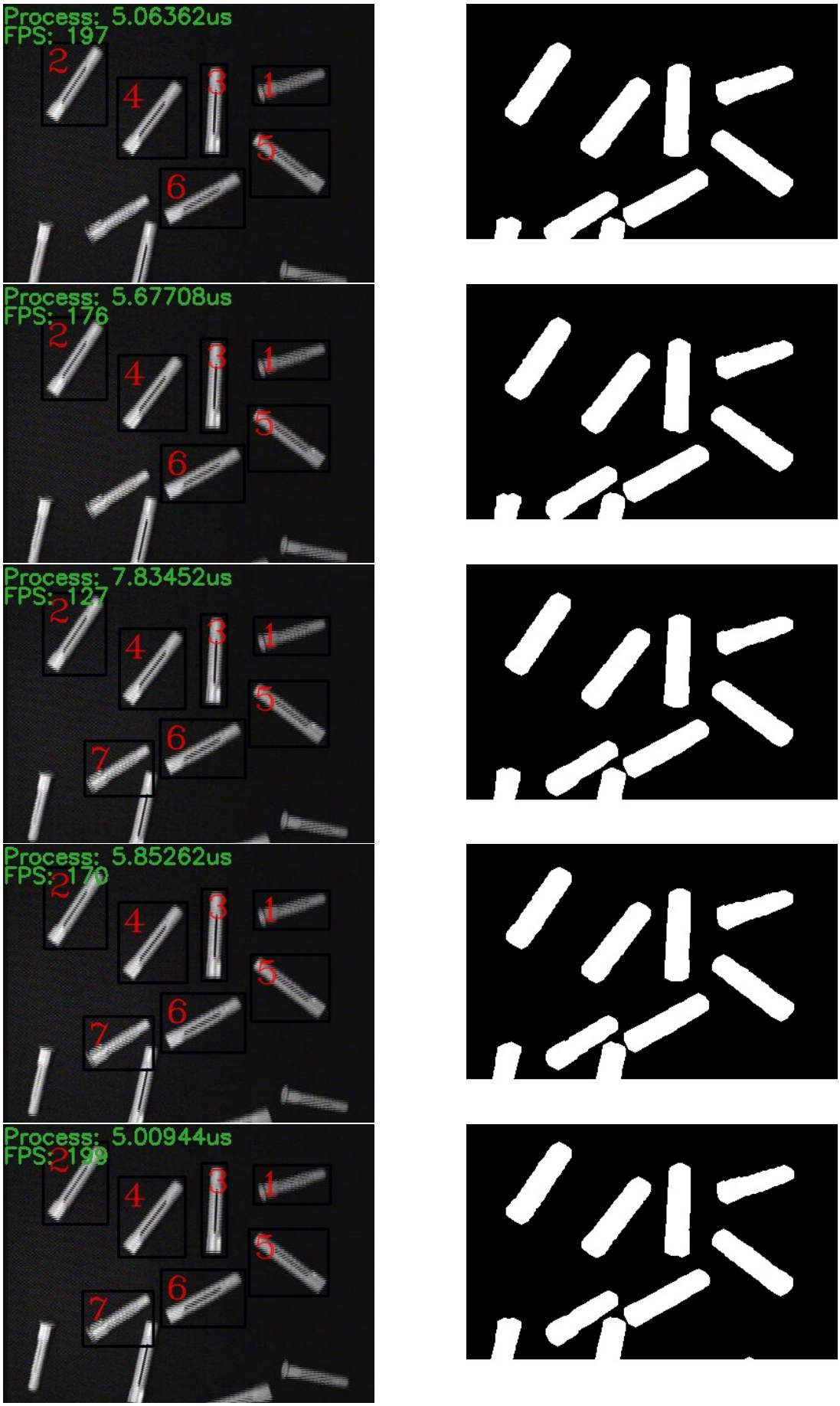
Kao što je navedeno u poglavlju 4, ključna informacija za prevazilaženje problema identifikacije objekata predstavlja centar mase (centroid) koji opisuje konturu objekta. Centar mase registrovanog objekta se provjerava svaki frejm, i uz dozvoljenu razliku po y-osi (smjer kretanja objekata na pokretnoj traci pri ekperimentalnoj postavci), identifikuje kao objekat iz prethodnog frejma. Također, uzeta je u obzir i razlika po x-osi koja predstavlja promjenu dispozicije određenog objekta uslijed nesavršenosti samog objekta. Uvid u implementaciju poređenja centroida je dat u algoritmu 2.

Kao što je i naglašeno u prethodnim poglavljima, relativno male konture neće biti uzete u obzir jer su to uglavnom konture nastale šumom ili zbog neadekvatnog osvijetljenja. Taj problem je riješen nakon ekstrakcije informacije o površini koju kontura zauzima u pikselima. Relativno male konture u odnosu na veličinu slike neće biti smatrane kao konture od interesa, odnosno neće biti registrovani kao objekti. No međutim, ako postoji potreba za registrovanjem manjih kontura kao objekata od interesa, parametre je moguće podesiti tako da i manje konture budu objekti od interesa.

Na slikama 5.2 i 5.4 je prikazan rezultat ukupnog rješenja, koje obuhvata algoritme 2, 3, 4 i 5, odnosno prikazano je brojanje i labeliranje objekata frejm po frejm. Na slikama je prikazano, iako postoji promjena pozicije objekata po y-osi, da objekti zadržavaju svoju stvarnu numeraciju, odnosno jednoznačno su identifikovani. Na slici 5.1 je prikazan dijagram aktivnosti koji prikazuje tok izvršavanja pomenutih radnji, dok je sama implementacija prikazana u algoritmima 3, 4, 2 i 5.



Slika 5.1: Dijagram aktivnosti razvijenog rješenja za brojanje i labeliranje objekata na video zapisu



Slika 5.2: Rezultat razvijenog algoritma za brojanje i labeliranje objekata na video zapisu

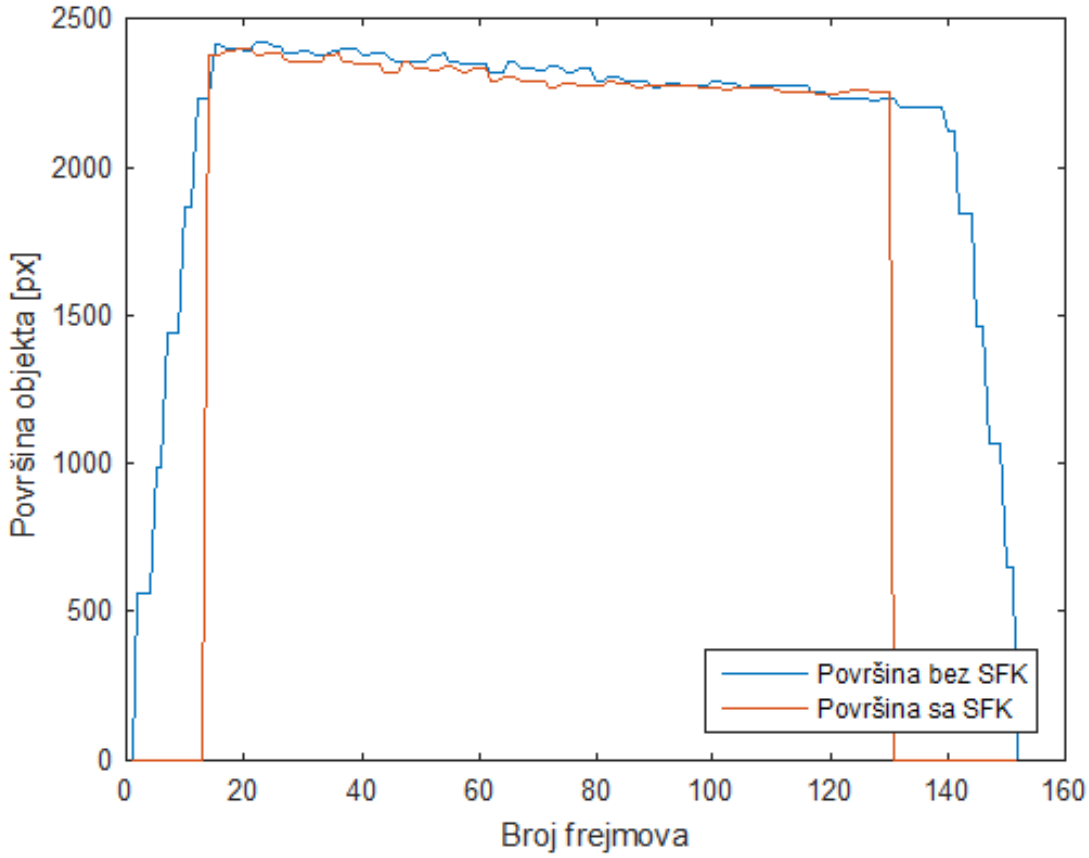
Prilikom detekcije objekata koji ulaze u vidljivo područje kamere, odnosno objekata koji izlaze iz istog, može doći do dva problema:

1. objekti prilikom svog pojavljivanja (ulaska u vidljivo područje kamere) imaju relativno veliku promjenu svoje površine, kao i u slučaju objekata nepravilnog izgleda centar mase može imati veliku dispoziciju što u velikoj mjeri utiče na rad algoritma.

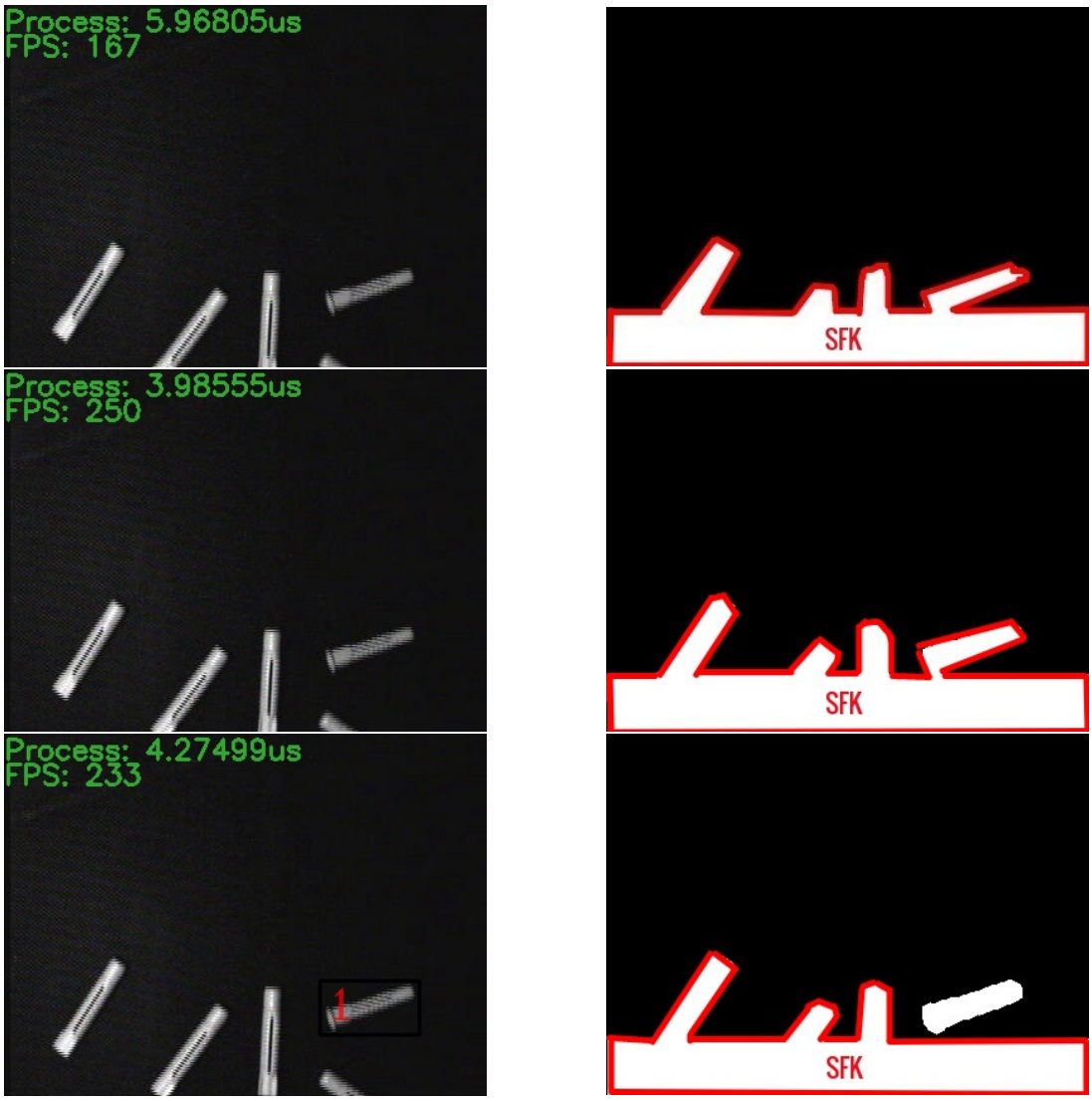
2. objekti čiji izgled podsjeća na slova 'U' i 'E' mogu prilikom ulaska u vidljivo područje biti dovoljno velike površine da se registruju kao stvarni objekti, no u zavisnosti od njihove orijentacije može se desiti da jedan stvarni objekat bude registrovan kao dva ili više.

Ovi slučajevi su izbjegnuti korištenjem softverski forsirane konture (**SFK**), koja je dovoljno velika (uz pretpostavku da aproksimativno znamo površinu kontura stvarnih objekata) da sama ne bude registrovana kao objekat. U slučaju ulaska novih objekata, oni neće biti registrovani kao stvarni objekti sve dok ne prestanu biti dio softverski forsirane konture, odnosno svom svojom stvarnom površinom ne budu u trenutnom frejmu, što je prikazano u algoritmu 4. Poređenje površine prvog objekta koji se analizira dato je na slici 5.3, gdje je prikazana površina objekta sa korištenjem **SFK** i bez korištenja iste.

Kako se može primijetiti na slici 5.2, detekcija novog objekta počinje tek kada on cijelom svojom konturom bude u potpunosti u trenutnom frejmu. To je omogućeno softverskim forsiranjem konture relativno velike površine na dijelu frejma gdje se pojavljuju objekti, tj. u donjem dijelu frejma, kao i dijelu frejma gdje objekti nestaju iz vidljivog polja kamere, tj. gornjem dijelu frejma.



Slika 5.3: Površina objekta sa korištenjem SFK i bez korištenja SFK



Slika 5.4: Rezultat razvijenog algoritma za brojanje i labeliranje objekata na video zapisu

* * *

Vidi se da je napravljena korelacija između frejmova na osnovu koje je omogućeno identifikovanje i povezivanje objekata. Korelacija je izvršena pomoću centra mase kontura koje opisuju objekte od interesa. Pored korelacije, razvijena je i softverski forsirana kontura koja je omogućila očuvanje površine kontura dok god se one nalaze u vidljivom polju kamere. Pokazani su pseudokodovi iz kojih je kreiran algoritam pomoću programskog jezika *C++* koji se nalazi u prilogu B.

Algoritam 2 Poređenje centroida kontura prethodnog i trenutnog frejma

```
1:  procedure CHECK CENTROIDS(contoursCurr, contoursLast)
2:      currCounter ← 0
3:      i ← 0
4:      j ← 0
5:      while i < contoursCurr.size() do
6:          while j < contoursLast.size() do
7:              dx ← contoursCurr[i].center.x − contoursLast[j].center.x
8:              dy ← contoursCurr[i].center.y − contoursLast[j].center.y
9:              if abs(dx) < min_center&abs(dy) < min_center then
10:                 currCounter ← currCounter + 1          ▷ Trenutni broj objekata na frejmu
11:             i ← j + 1
12:         i ← i + 1
13:     k ← totalCounter − currCounter
14:     while k < totalCounter do
15:         totalObjects[k].updateCenterOfObject()
16:         k ← k + 1
17:     return currCounter
```

Algoritam 3 Segmentacija i morfološka obrada slike

```
1:  procedure IMAGE PROCESSING(frame)
2:      kernel_er ← structuringElement(3,3)
3:      kernel_dil ← structuringElement(4,4)
4:      cvtColor(frame, grey)
5:      threshold(grey, tresh)
6:      erode(tresh, er, kernel_er)
7:      dilate(er, dil, kernel_dil)
8:      processed ← dil
9:      processed(Range(processed.rows − 50, processed.rows), Range(0, processed.cols)) ←
    255                                   ▷ dodavanje softverski forsirane konture
10:     return processed
```

Algoritam 4 Provjeravanje da li je objekat dio SFK

```
1:  procedure CHECKSFC(contours)
2:      i ← 0
3:      while i < contours.size() do
4:          if contours[i] is not part of SFC &contours[i].area() is within limits then
5:              newContours.pushback(contours[i])
6:          i ← i + 1
7:      return newContours
```

Algoritam 5 Detekcija objekata na prethodnom i narednom frejmu, brojanje i labeliranje

```
procedure DETECTION, NUMBERING AND LABELING(lastFrame, currentFrame)
    findContours(lastFrame, contours1)
    findContours(currentFrame, contours2)
    contours1 ← checkSFC(contours1)
    contours2 ← checkSFC(contours2)
    currCounter ← checkCentroids(contours1, contours2)
    if contours2.size() > contours1.size() then
        registerNewObject()
        totalCounter ← totalCounter + 1
        currCounter ← currCounter + 1
    i ← totalCounter − currCounter
    while i < totalCounter do
        totalObjects[i].label(currentFrame)
    return currentFrame
```

Poglavlje 6

Analiza eksperimentalnih rezultata

U ovom poglavlju biće prikazani rezultati implementiranog algoritma pomoću programskog jezika *C++* uz korištenje *OpenCV* biblioteka. Kroz više primjera biće dat uvid u tačnost samog algoritma pri različitim brzinama pokretne trake. Parametri metoda korištenih za obradu slike dati su u tabeli 6.1. Eksperimenti su vršeni nad videozapisima snimljenim na *Elektrotehničkom fakultetu u Sarajevu* kamerom *Sony XC-55* koja se nalazi na modelu *ETFCam v1.0*. Frejmovi ekstraktovani iz videozapisa su rezolucije 640×480.

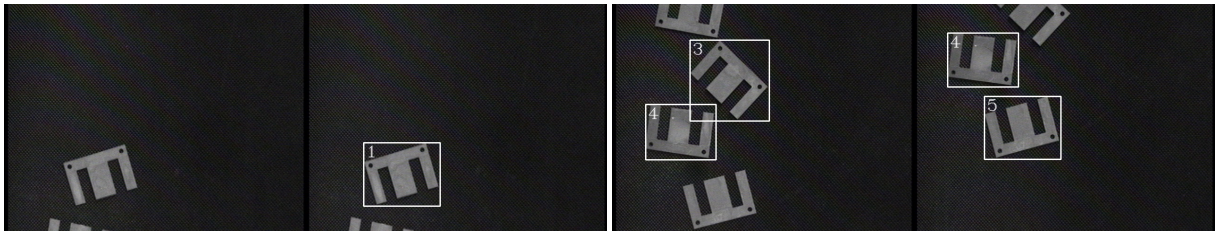
Parametar	**Testni scenarij 1**	**Testni scenarij 2**	**Testni scenarij 3**
Threshold	85	85	85
Strukturni element erozije	$I_{3\times3}$	$I_{3\times3}$	$I_{3\times3}$
Strukturni element dilatacije	$I_{5\times5}$	$I_{4\times4}$	$I_{4\times4}$
Broj iteracija erozije	1	10	1
Broj iteracija dilatacije	3	3	3
Minimalna dispozicija centroida [px]	25	10	25
Minimalna površina konture [px]	3000	50	300
Maksimalna površina konture [px]	20000	20000	20000

Tabela 6.1: Parametri metoda korištenih za obradu slike[1]

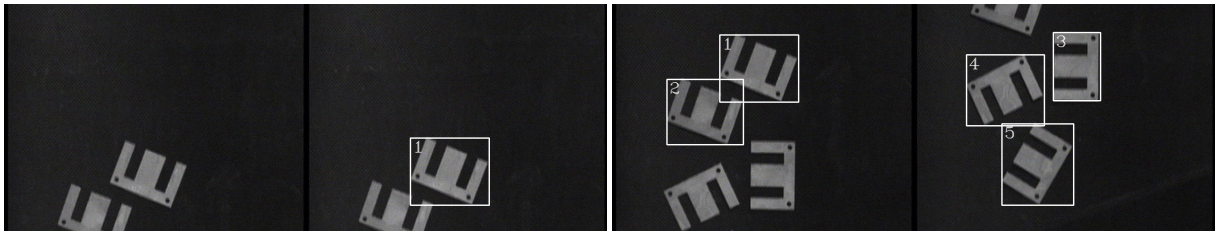
Prvi razmatrani scenarij predstavljaju objekti oblika slova 'E', koji pri ulasku u vidljivo polje kamere tipično imaju problema zbog oblika, no međutim korištenjem **SFK** taj problem je izbjegnut. Rezultat algoritma je prikazan na slikama 6.1, 6.2, 6.3 i 6.4. Prosječna brzina izvršavanja algoritma na testnom scenariju 1 iznosi 1/188 [s], što omogućava obradu od 188 [fps][2]. Pošto kamera generiše brzinu od 1/30 [s], zaključuje se da algoritam radi u realnom vremenu.

[1] $I_{a\times a}$ - kvadratna matrica jedinica (eng. *all-ones matrix*), $a \in N$

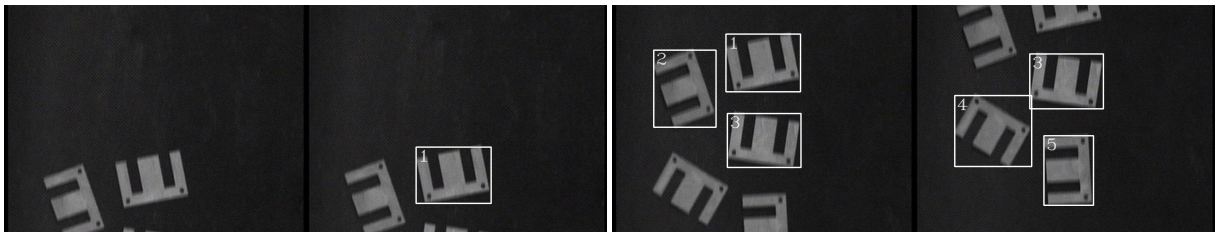
[2] fps (frames per second) - frejmova u sekundi



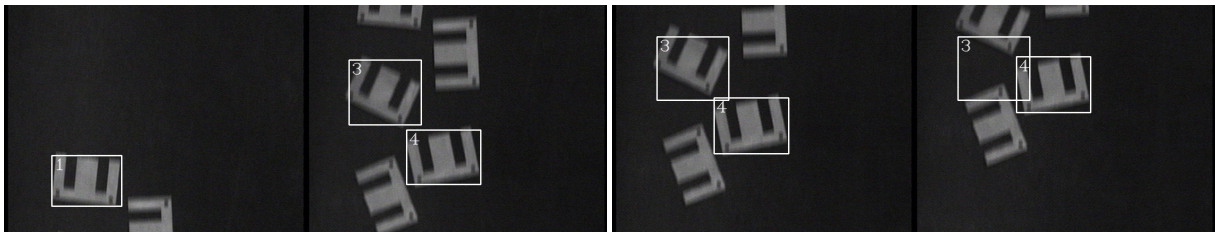
Slika 6.1: Testni scenarij 1 - Rezultat algoritma pri normalizovanoj brzini trake od 27[%]



Slika 6.2: Testni scenarij 1 - Rezultat algoritma pri normalizovanoj brzini trake od 39[%]



Slika 6.3: Testni scenarij 1 - Rezultat algoritma pri normalizovanoj brzini trake od 45[%]



Slika 6.4: Testni scenarij 1 - Rezultat algoritma pri normalizovanoj brzini trake od 62[%]

Literatura

[1] Batchelor, B. G., Whelan, P. F., Intelligent vision systems for industry. Springer Science & Business Media, 2012.

[2] Demant, C., Demant, C., Streicher-Abel, B., Industrial image processing. Springer, 1999.

[3] Sonka, M., Hlavac, V., Boyle, R., Image processing, analysis, and machine vision. Cengage Learning, 2014.

[4] Szeliski, R., Computer vision: algorithms and applications. Springer Science & Business Media, 2010.

[5] Application Guide: Sony XC-55/55B, Sony Corporation, 2000, datum pristupa: 25.03.2018., dostupno na: "http://www.sony.co.jp/ISP/"

[6] Russ, J. C., The image processing handbook. CRC press, 2016.

[7] Ibrahim, Z., Abdul Rahman Al-Attas, S., “Wavelet-based printed circuit board inspection algorithm”, Vol. 12, 01 2005, str. 201-213.

[8] Yang, M., Kpalma, K., Ronsin, J., “A survey of shape feature extraction techniques”, 2008.