



UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET
ODSJEK ZA AUTOMATIKU I ELEKTRONIKU

Određivanje apsolutne pozicije i orijentacije objekta na osnovu digitalnih slika i eksternih senzorskih signala

ZAVRŠNI RAD
- PRVI CIKLUS STUDIJA -

Student:
Dženana Puščul

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

Sarajevo,
septembar 2018.

Postavka zadatka završnog rada I ciklusa:
Određivanje apsolutne pozicije i orijentacije objekta na osnovu digitalnih
slika i eksternih senzorskih signala

Ukoliko se kamera ili sistem računarske vizije koristi za određivanje pozicije pokretnih objekata, tada postoji nekoliko faktora koji utiču na određivanje apsolutne pozicije i orijentacije nekog objekta u datom vremenskom trenutku, a to su: prostorne transformacije koje unosi kamera, kašnjenja zbog različitog načina i trajanja uzimanja slike (softversko/hardversko, eksterno/interno okidanje), utjecaji osvjetljenja i sl. Zbog toga postoje određene poteškoće prilikom određivanja tačne apsolutne pozicije i orijentacije objekta u datom vremenskom trenutku, i njegove estimacije u budućnosti bazirane na sinhronizaciji sa eksternim signalima (fotoelektrični senzori, enkoderi, motori i sl.). U okviru ovog rada je potrebno implementirati sistem koji omogućava određivanje apsolutne pozicije i orijentacije objekta na digitalnoj slici, i implementirati mogućnost estimacije trajektorije objekta u budućem vremenu.

Koncept i metode rješavanja:

Rad treba da se sastoji iz sljedećih cjelina:

- pregleda literature,
- analize postupaka kalibracije kamere,
- analize metoda za određivanje pozicije i orijentacije oblika,
- implementacije odgovarajućih algoritama u C++/OpenCV okruženju,
- integracije sistema u prototip ETFcam v.1.0,
- analize eksperimentalnih rezultata.

Polazna literatura:

[1] Sonka, M., Hlavac, V., Boyle, R., "Image Processing, Analysis, and Machine Vision", Thomson, 2008.

[2] Lee, E.A., Varaiya, P., "Structure and Interpretation of Signals and Systems", Electrical Engineering & Computer Science University of California, Berkeley, 2000

Izjava autora

Ja dolje potpisani izjavljujem da ovaj podnesak predstavlja moj vlastiti rad i da, prema mom saznanju i vjerovanju, u njemu nije sadržan bilo koji dio koji je prethodno objavljen ili napisan od strane drugih osoba, niti materijal koji je prihvaćen u svrhu dodjele diplome bilo kojeg stupnja na bilo kojoj visokoškolskoj ili naučnoj instituciji u svijetu. Autorska prava doktorske disertacije ostaju pri autoru disertacije i Elektrotehničkom fakultetu Univerziteta u Sarajevu.

Dženana Puščul

U Sarajevu,
septembar 2018.

Sažetak

Računarska vizija je u današnje vrijeme postala veoma važan alat u industriji, čiji je osnovni cilj ekstrakcija bitnih informacija sa slike. Ono što je bitno na slici varira od aplikacije do aplikacije. Određivanje pozicije i orijentacije igra važnu ulogu u industrijskim sistemima, kako za interpretaciju rezultata procesiranja slike, tako i za pozicioniranje raznih industrijskih pomagala (kao što je na primjer prihvatnica (*eng. gripper*)). U ovom radu bit će date teorijske osnove kalibracije i obrade slike te će biti objašnjeni načini nalaženja pozicije i orijentacije objekata na slici. Potrebno je provesti sljedeće korake:

- kalibracija kamere,
- segmentiranje slike,
- morfološka obrada slike,
- određivanje pozicije i orijentacije.

U sklopu rada bit će razvijena aplikacija na ETFCam industrijskom modelu.

Abstract

The computer vision has become a very important tool in the industry today. Its basic purpose is to extract the essential information from the image. What matters in the image varies from application to application. Determining position and orientation plays an important role in industrial systems, both for the interpretation of image processing results and for positioning various industrial aids (such as a gripper). In this paper, theoretical basics of calibration and image processing and the ways of determining location and orientation of the objects in the figure will be explained. The following steps are required:

- camera calibration,
- image segmentation,
- morphological image processing,
- determination of position and orientation.

As part of this paper, an application will be developed on the ETFCam industrial model.

Abstract

Sadržaj

Popis slika	viii
Popis tabela	ix
1 Uvod	1
1.1 Obrazloženje teme	1
1.2 Reprezentacija slike u računarima	1
1.3 Sony XC-55 kamera	3
2 Kalibracija kamere	5
2.1 Pinhole model kamere	5
2.2 Radijalna distorzija	8
2.3 Kalibracija kamere u <i>OpenCV</i> -u	9
3 Pretprocesiranje slike	10
3.1 Segmentacija pragom (<i>Thresholding</i>)	10
3.2 Detekcija ivica	11
3.2.1 Detekcija ivica zasnovana na gradijentu	11
3.2.2 Canny metoda detekcije ivica	12
3.3 Regijski bazirana segmentacija	13
3.4 Morfološka obrada slike	14
3.4.1 Erozija	14
3.4.2 Dilatacija	15
3.5 Segmentacija u <i>OpenCV</i> -u	15
4 Određivanje pozicije i orijentacije	16
4.1 Određivanje pozicije	16
4.2 Određivanje orijentacije	18
4.2.1 Osa najmanje inercije	18
4.2.2 Signatura ugaone udaljenosti	19
5 Postavka eksperimenta	20
5.1 ETFCam - model industrijskog sistema za automatsku vizuelnu inspekciju	20
5.1.1 Modicon M241 programabilni logički kontroler (PLC)	22
5.1.2 Senzor ET318B	22
5.1.3 Inkrementalni enkoder	23
5.1.4 Modbus protokol	23
5.2 Opis rada aplikacije	24

6	Eksperimentalni rezultati	26
6.1	Analiza tačnosti određene pozicije i orijentacije	27
6.2	Analiza i estimacija kašnjenja u izvršavanju algoritma pri realnim uslovima . .	29
6.3	Estimacija trajektorije u budućem vremenu	31
6.4	Problemi određivanja pozicije	32
	Prilozi	34
A	Kod za kalibraciju kamere	35
B	Određivanje pozicije i orijentacije kroz <i>OpenCV</i>	36
	Lista publikacija	38
	Literatura	38

Popis slika

1.1	<i>(a) Originalna slika; (b) Originalna slika projektirana na rešetku sa fotodetektora; (c) Izračunat prosječni svjetlosni intenzitet svake ćelije; (d) Reprerentacija slike pod (c) pomoću cjelobrojnih vrijednosti koje odgovaraju određenim razinama sive [1]</i>	2
1.2	<i>Sony XC-55 [2]</i>	3
1.3	<i>SG-257 Board [2]</i>	4
1.4	<i>Moguća vremena ekspozicije za Sony XC-55 [2]</i>	4
2.1	<i>Pinhole model kamere [3]</i>	5
2.2	<i>Geometrijski model pinhole kamere [4]</i>	6
2.3	<i>Geometrija modela: C-optički centar, P – glavna tačka, ϕ –fokalna udaljenost [3]</i>	6
2.4	<i>Slika 2.4. a) Slika sa distorzijom; b) Slika na kojoj je uklonjena distorzija [3]</i>	9
3.1	<i>a) Slika u prostoru sivih tonova; b) Slika nakon segmentacije pragom</i>	11
3.2	<i>a) Originalna slika; b) Rezultat primjene Laplaciana [5]</i>	12
3.3	<i>a) Originalna slika; b) Rezultat vertikalnog Prewittovog filtera; c) Rezultat horizontalnog Prewittovog filtera; d) Mapa kvantizirane orijentacije; e) Mapa amplituda gradijenta; f) Amplitude nakon supresije ne-maksimuma; g) Dva praga: crveni pikseli su iznad nižeg, a ispod višeg nivoa; h) Određene ivice [3]</i>	13
3.4	<i>a) Binarna slika; b) Slika nakon erozije</i>	14
3.5	<i>a) Binarna slika; b) Slika nakon dilatacije</i>	15
4.1	<i>Pravougaonik se mijenja sa orijentacijom objekta [5]</i>	16
4.2	<i>Centroid, centar opisane kružnice i centroid sa uračunatim težinama [5]</i>	18
5.1	<i>ETFCam</i>	21
5.2	<i>Principijelna šema</i>	21
5.3	<i>Modicon TM241CEC24R PLC [6]</i>	22
5.4	<i>Senzor ET318B [7]</i>	23
5.5	<i>Enkoder [8]</i>	24
5.6	<i>Tok odvijanja algoritma</i>	25
6.1	<i>a) Koordinati sistem postavljen u stvarnom svijetu u odnosu na kojeg se mjeri pozicija; b) područje koje će slika obuhvatiti</i>	26
6.2	<i>Primjer određivanja apsolutne pozicije i orijentacije</i>	27
6.3	<i>Objekti nad kojima je proveden eksperiment</i>	28
6.4	<i>a) Referentna (stvarna) vrijednost; b) Traka pređe put Δx</i>	29
6.5	<i>Isti objekat pri različitim brzinama trake: a) pri 33%; b) pri 40%; c) pri 50%; d) pri 60%; e) pri 66% ; f) pri 83%; g) pri 100%</i>	30

6.6	<i>Crvenom bojom naznačena je trajektorija u budućem vremenu za dati objekat .</i>	31
6.7	<i>Problem se javlja ukoliko je objekat preveliki</i>	32
A.1	<i>Određivanje unutrašnjih kornera uzorka u OpenCV-u</i>	35
B.1	<i>a) Originalna slika; b) Segmentacija pragom; c) Erozija; d) Dilatacija; e) Konture; f) Određene su pozicija i orijentacija</i>	36
B.2	<i>Novčići nemaju orijentaciju</i>	37

Popis tabela

6.1	<i>Eksperimentalni rezultati određivanja pozicije i greške mjerenja</i>	28
6.2	<i>Eksperimentalni rezultati određivanja orijentacije i greške mjerenja</i>	28
6.3	<i>Trenutna i estimirana pozicija objekta</i>	31

Poglavlje 1

Uvod

1.1 Obrazloženje teme

Računarska vizija ima za cilj određivanje korisnih informacija sa slika. Problem predstavlja kompleksnost vizualnih podataka. Sami objekti na slici mogu biti preklapljeni i izobličeni, može ih biti veliki broj, te slika može sadržavati razne šumove i distorzije. Za jedan algoritam računarske vizije čak i određivanje konture objekta ne predstavlja lagan zadatak [3]. Računarska vizija mora da uzme, procesira, analizira digitalnu sliku, te odredi tražene informacije.

Glavni akcenat u ovom radu će biti dat određivanju pozicije i orijentacije objekata na slici, te će biti analizirani faktori koji utječu na njihovo nalaženje, kao što su: prostorne transformacije koje unosi kamera i kašnjenja zbog različitog načina i trajanja uzimanja slike. Bit će objašnjeni načini reprezentacije slike u samom računaru, kalibracija kamere, neke metode pretprocesiranja slike i nalaženja pozicije i orijentacije. U sklopu ovog rada bit će razvijena aplikacija, čiji će rezultati primjene biti analizirani u poglavlju 6.

1.2 Reprezentacija slike u računaru

Intuitivno, slika se može shvatiti kao nešto što ljudsko oko, ili neki drugi senzor, "snimi" kada svjetlost padne na njega u određenom trenutku vremena. Slika koja se dobija je dvodimenzionalna. Ona se može modelirati kontinualnom funkcijom dvije promjenljive $f(x,y)$, gdje x i y predstavljaju koordinate u ravni. Vrijednosti koje poprima funkcija crno-bijele (*eng. grayscale*) slike najčešće odgovaraju jačini svjetlosti u određenoj tački, ali mogu predstavljati i neke druge fizikalne veličine kao što su temperatura, udaljenost od posmatrača itd. Budući da se radi o kontinualnoj funkciji, koordinate su realni brojevi (najčešće nenegativni) kao i same vrijednosti funkcije.

Da bi se signal slike mogao predstaviti u računaru, pomenuta funkcija se mora digitalizirati, odnosno izvršiti diskretizacija i domena i kodomena. Funkcija dobijena nakon diskretizacije naziva se digitalna funkcija slike. Ovakve funkcije obično imaju domen dat izrazom (1.1), gdje su koordinate x i y prirodni brojevi, a x_m i y_n predstavljaju najveće moguće vrijednosti x i y , respektivno.

$$R = \{(x,y) | 0 < x < x_m + 1 \wedge 0 < y < y_n + 1 \wedge x \in N \wedge y \in N\} \quad (1.1)$$

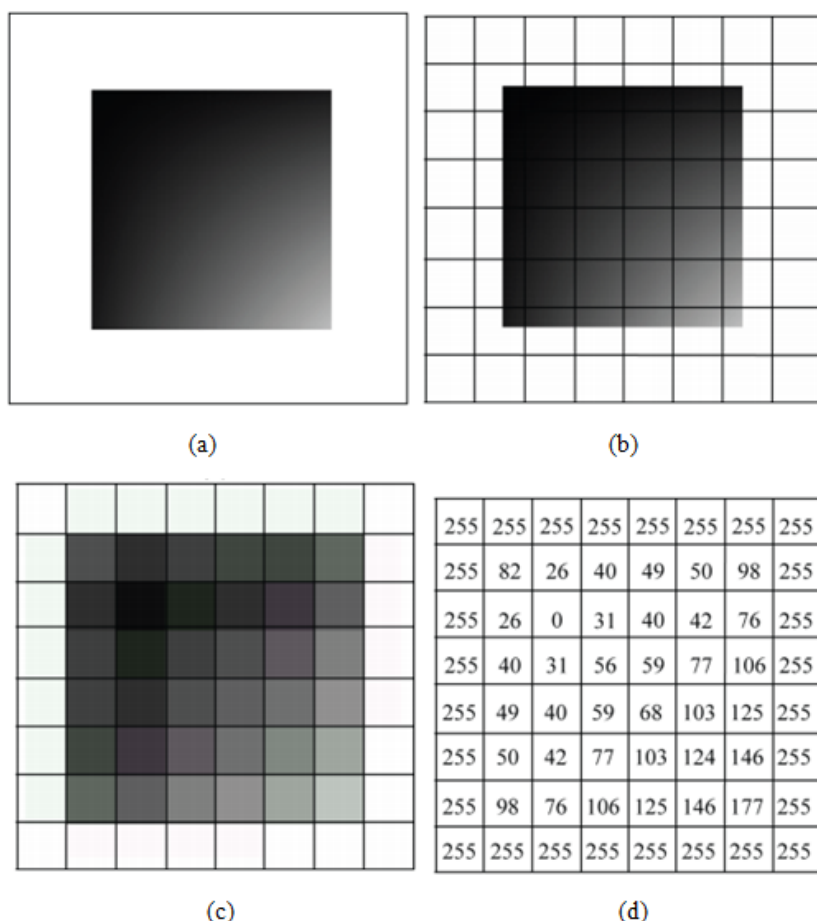
Kodomen je također ograničen i može poprimiti samo određene vrijednosti. Te vrijednosti se nazivaju nivoi (razine) sive (*eng. gray levels*). Konvencionalno, najmanja vrijednost koju

ovakve funkcije mogu poprimiti je crna, a najviša bijela [5]. Jedan od načina diskretizacije slike je njeno projektiranje na rešetku fotodetektorskih ćelija. Svaka ćelija ima mogućnost da proračuna prosječni intenzitet svjetlosti projektiran na njega, time diskretizirajući sliku prostorno. Ovaj postupak je prikazan na slikama 1.1 (a) i 1.1 (b) [1].

Diskretizacija kodomena se vrši kvantizacijom koja svakoj vrijednosti proračunatog intenziteta svjetlosti dodjeljuje cjelobrojnu vrijednost određenu razinom sive. Na ovaj način kodomen je postao skup k vrijednosti. Obično, k ima vrijednost 256, gdje crna predstavlja vrijednost 0, a bijela 255. Ovaj postupak je prikazan na slici 1.1 (c).

Nakon izvršenih diskretizacija dobijena je digitalna verzija početne slike. Da bi bilo moguće izvršavati bilo kakve operacije nad slikom u računar, ona mora biti predstavljena nekom diskretnom strukturom podataka. Najčešće je to matrica. Elementi matrice predstavljaju cjelobrojne vrijednosti dodijeljene nivoima sive prilikom kvantizacije. Na slici 1.1 (d) je prikazana reprezentacija digitalne slike kao matrice brojeva.

Beskonačno mala tačka uzorkovanja u rešetki naziva se *piksel*. U stvarnosti, piksel nije beskonačno mali, nego ima konačne dimenzije. Piksel se može predstaviti kao nedjeljiva jedinica slike [5]. Kvalitet slike zavisi od broja piksela kojim je ona predstavljena. Veći broj piksela daje bolju kvalitetu slike.



Slika 1.1: (a) Originalna slika; (b) Originalna slika projektirana na rešetku sa fotodetektora; (c) Izračunat prosječni svjetlosni intenzitet svake ćelije; (d) Reprezentacija slike pod (c) pomoću cjelobrojnih vrijednosti koje odgovaraju određenim razinama sive [1]

1.3 Sony XC-55 kamera

CCD (*eng. charged-couple device*) predstavlja elektronički uređaj koji se koristi za kreiranje slika objekata, pohranu informacija ili prenos električnog naboja. Ovaj uređaj prikuplja svjetlost koja dolazi sa objekta ili električni naboj. CCD prima ovaj optički ili električni ulaz i pretvara ga u elektronički izlaz. Ovaj elektronički signal se dalje procesira kako bi se stvorila slika ili dale korisnicima vrijedne informacije [9]. Sony XC-55 predstavlja kompaktnu, laganu, monohromatsku CCD kameru. CCD koristi kvadratne piksele najpogodnije za računarsku viziju, koji osiguravaju jednaku rezoluciju duž horizontalne i vertikalne ose. Piksel ima dimenzije $4.7 \mu\text{m} \times 4.7 \mu\text{m}$. Na slici 1.2 prikazana je spomenuta kamera.



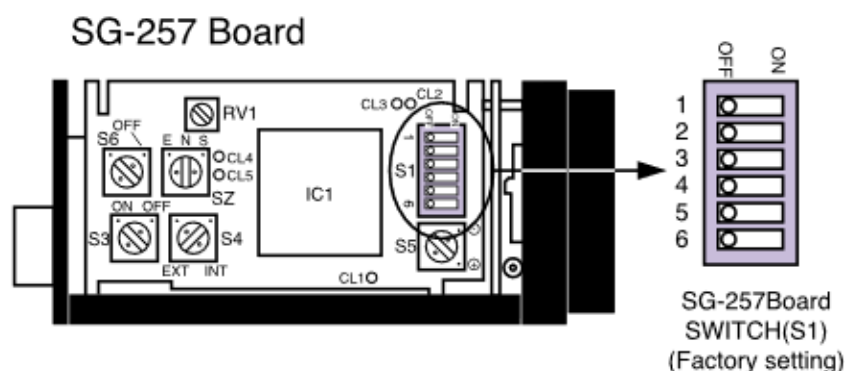
Slika 1.2: Sony XC-55 [2]

Na stražnjoj strani kamere nalazi se dvanaestopinski multikonektor, prekidač za odabir pojačanja (*eng. gain switch*), prekidač za odabir tipa signala (*eng. signal switch*) i kontrolni ulaz za odabir manuelnog pojačanja. Multikonektor se povezuje na DC IN/SYNC CCXC-12Pxxx kabal. Gain switch može biti automatski (A), fiksirani (F) i manualni (M). Automatski gain na izlazu daje ujednačene nivoe slike, u ovisnosti od osvijetljenosti objekta. Fiksirani gain ima najveći odnos signal/šum (*eng. signal-to-noise*). U fabrici, manualni gain se podešava na fiksiranu osjetljivost tako da odgovara standardnom objektu. Gain se može podesiti pomoću manual gain control volume. Pomoću sklopke Signal switch može se odabrati da li će izlazni signal sa kamere biti isprepleten (*eng. Interlaced, 1I*) ili neisprepleten (*Noninterlaced, 1N*). Ako je odabran neisprepleteni signal, kamera će na izlazu dati sve piksele CCD-a kontinualno u 1/30 sekundi. Ako je odabran isprepleten signal, svi podaci u pikselima će se pojaviti na izlazu u dvije uzastopne iteracije, od kojih svaka traje 1/60 sekundi. Podaci iz iteracija se onda kombinuju (prepliću) kako bi se dobila cijela slika [2].

Svaka slika počinje sa izlaganjem senzora u kameri svjetlosti. Koliko će slika biti svijetla ili tamna ovisi od toga koliko svjetlosti dolazi do leća objektiva, odnosno senzora. Izloženost senzora svjetlosti kontrolišu poklopac (*eng. shutter*) i otvor (*eng. aperture*). Poklopac (shutter) se otvara i zatvara i tako počinje odnosno završava izloženost senzora svjetlosti. Kroz otvor (aperture) svjetlost dolazi u kameru. Mijenjajući ove dvije veličine mijenja se i kvalitet slike (zamagljenost, oštrina, osvijetljenje itd.).

Poklopac (*eng. shutter*) je otvoren samo kada se kreira slika. Inače je zatvoren i ne dopušta svjetlosti da dopre do senzora. Vrijeme ekspozicije (*eng. shutter speed*) predstavlja vrijeme koje je

poklopac otvoren. Što je ono veće to manje svjetlosti dopire do senzora pa će slika biti mračnija. Osim izloženosti senzora, pomoću spomenutog vremena također se kontroliše način na koji je kretanje zabilježeno kamerom. Objekat koji se kreće će biti zamagljeniji (*eng. blurred*) što je vrijeme ekspozicije veće. Dakle, ako je potrebno dobiti preciznu i oštru sliku objekta koji se kreće, potrebno je birati manje vrijeme po potrebi [10]. Na Sony XC-55 modelu kamere, vrijeme ekspozicije se podešava pomoću prekidača 4, 5 i 6 na SG-257 tabli (*eng. SG-257 board*) koja je prikazana na slici 1.3.



Slika 1.3: SG-257 Board [2]

U tabeli datoj na slici 1.4 prikazano je kako se trebaju postaviti navedeni prekidači da bi se dobilo određeno vrijeme ekspozicije (1 - ON, 0 - OFF).

PREKIDAČI			VRIJEME EKSPOZICIJE
4	5	6	
0	0	0	OFF
1	0	0	1/100sec.
0	1	0	1/250sec.
1	1	0	1/500sec.
0	0	1	1/1000sec.
1	0	1	1/2000sec.
0	1	1	1/4000sec.
1	1	1	1/8000sec.

Slika 1.4: Moguća vremena ekspozicije za Sony XC-55 [2]

Okidanje (*eng. triggering*) se može vršiti interno ili eksterno. Interno se to radi softverski, a eksterno pomoću vanjskog signala koji se dovodi na kameru [2]. Ova kamera će biti korištena u eksperimentalnom dijelu.

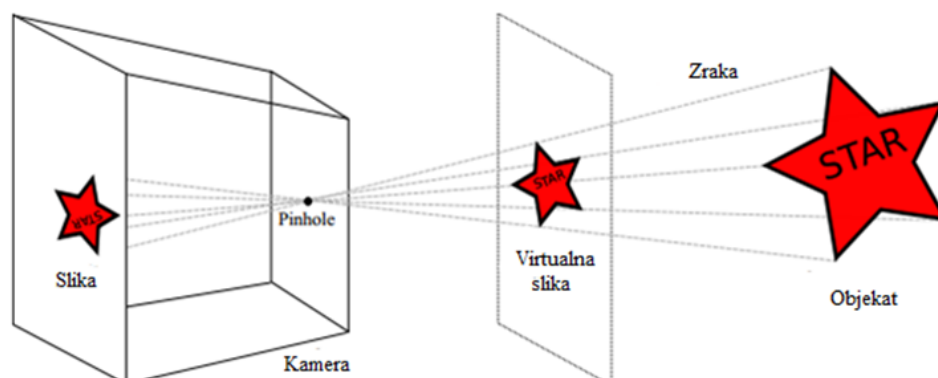
Poglavlje 2

Kalibracija kamere

Kalibracija kamere je proces kojim se određuju tačne vrijednosti parametara kamere. Ona igra glavnu ulogu u svim aplikacijama računarske vizije [11]. U većini slučajeva, pri razmatranju ovog problema koristi se pinhole model kamere. Međutim, ovaj model ne uzima u obzir sočiva, što znači da njegova analiza ne obuhvata probleme distorzije. O pinhole modelu kamere i radijalnoj distorziji koja nastaje na sočivu, bit će riječi u ovom poglavlju.

2.1 Pinhole model kamere

Pinhole model kamere je jednostavan geometrijski model koji opisuje način na koji se tačke iz realnog svijeta projektuju na sliku. Pinhole kamera se može predstaviti kao zatvorena kutija sa malim otvorom kroz koji ulaze zrake svjetlosti i formiraju invertovanu sliku na suprotnoj strani od otvora, koja se naziva ravan slike. Činjenica da se ovakvim projektovanjem dobije invertovana slika je nezgodna, te se zbog toga ispred kutije "postavlja" virtualna ravan slike, na kojoj se projektira neinvertovana slika. Ovako nešto nije moguće konstruisati u stvarnosti, ali je matematički ekvivalentno istinskom pinhole modelu kamere, s tim da je slika obrnuta. Na slici 2.1 prikazan je ovaj model.

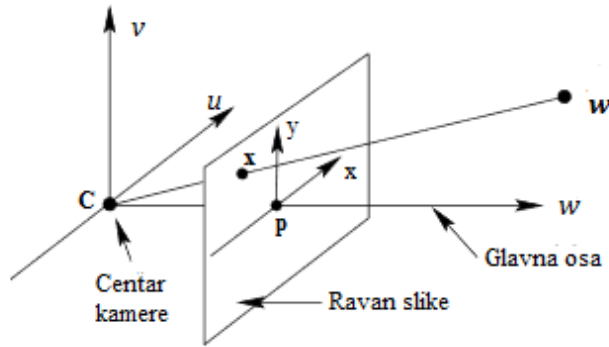


Slika 2.1: Pinhole model kamere [3]

Mali otvor na kutiji koji pušta zrake svjetlosti predstavlja optički centar. Neka on predstavlja centar trodimenzionalnog koordinatnog sistema koji predstavlja realni svijet. Taj koordinatni sistem ima tri ose: u , v i w raspoređene u prostoru kao na slici 2.2. Sada se svaka tačka realnog

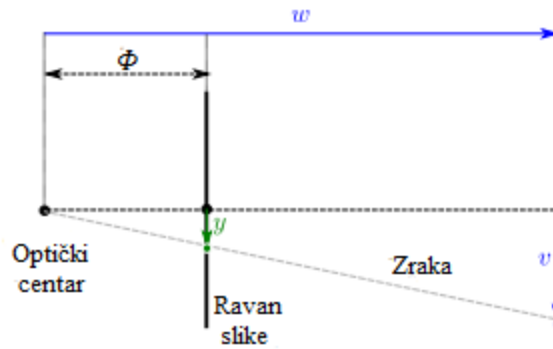
svijeta može opisati pomoću ovog koordinatnog sistema kao $\mathbf{w} = [u \ v \ w]^T$. Virtualna slika će biti formirana na virtualnoj ravni koja se još naziva i ravan slike, koja je okomita na osu w (optička ili glavna osa). Tačka u kojoj se sijeku ravan slike i optička osa naziva se glavna ili principalna tačka. Fokalna udaljenost predstavlja udaljenost optičkog centra od glavne tačke.

Neka je Oxy koordinatni sistem u ravni slike, postavljen kao na slici 2.2. Potrebno je odrediti poziciju $\mathbf{x} = [x \ y]^T$ u koju se preslikava tačka $\mathbf{w} = [u \ v \ w]^T$. Ovakvo preslikavanje naziva se perspektivna projekcija. Pozicija \mathbf{x} zapravo predstavlja tačku u kojoj zraka svjetlosti siječe ravan slike [3].



Slika 2.2: Geometrijski model pinhole kamere [4]

Na slici 2.3, uočavaju se dva slična trougla. Na osnovu te sličnosti slijedi odnos dat relacijom (2.1) iz kojeg slijedi (2.2).



Slika 2.3: Geometrija modela: C – optički centar, P – glavna tačka, ϕ – fokalna udaljenost [3]

$$\frac{y}{v} = \frac{\phi}{w} \quad (2.1)$$

$$y = \phi \frac{v}{w} \quad (2.2)$$

Analogno se dobija i izraz za x , koji je dat relacijom (2.3).

$$x = \phi \frac{u}{w} \quad (2.3)$$

Pikseli ne moraju da budu kvadratnog oblika, tako da fokalne udaljenosti ne moraju da budu

iste u x i y pravcima. Stoga, relacije (2.2) i (2.3) postaju relacije (2.4) i (2.5), respektivno. ϕ_x i ϕ_y predstavljaju fokalne udaljenosti u x i y pravcima.

$$y = \phi_y \frac{v}{w} \quad (2.4)$$

$$x = \phi_x \frac{u}{w} \quad (2.5)$$

Sljedeća stvar koja se treba uzeti u obzir je ta što se pozicija $\mathbf{x} = [0 \ 0]^T$ u ravni slike nalazi u glavnoj tački. Međutim, to ne mora uvijek biti slučaj. Matematički, ovaj problem se rješava dodavanjem offset parametara δ_x i δ_y . Ovi parametri predstavljaju koordinate u pikselima glavne tačke u odnosu na koordinatni sistem Oxy . Sada model poprima oblik dat relacijama (2.6) i (2.7).

$$y = \phi_y \frac{v}{w} + \delta_y \quad (2.6)$$

$$x = \phi_x \frac{u}{w} + \delta_x \quad (2.7)$$

Ukoliko ose x i y nisu okomite uvodi se faktor iskrivljenja γ (relacije (2.8) i (2.9)). Kako se ovo rijetko dešava, faktor iskrivljenja je skoro uvijek 0.

$$y = \phi_y \frac{v}{w} + \delta_y \quad (2.8)$$

$$x = \phi_x \frac{u + \gamma v}{w} + \delta_x \quad (2.9)$$

Često je potrebno odrediti položaj tačke \mathbf{w} u nekog drugom trodimenzionalnom koordinatnom sistemu. Neka je \mathbf{w}' položaj tačke \mathbf{w} u novom koordinatnom sistemu. Odnos između \mathbf{w}' i \mathbf{w} dat je relacijom (2.10), odnosno relacijom (2.11). U ovim relacijama, Ω predstavlja matricu rotacije, a τ translacioni vektor.

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{21} & \omega_{22} & \omega_{23} \\ \omega_{31} & \omega_{32} & \omega_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (2.10)$$

$$\mathbf{w}' = \Omega \mathbf{w} + \tau \quad (2.11)$$

Sada je moguće formirati potpuni pinhole model kamere koristeće relacije (2.8), (2.9) i (2.10). Koordinate $[x \ y]^T$ i $[u \ v \ w]^T$ su povezane relacijama (2.12) i (2.13).

$$x = \frac{\phi_x(\omega_{11}u + \omega_{12}v + \omega_{13}w + \tau_x) + \gamma(\omega_{21}u + \omega_{22}v + \omega_{23}w + \tau_y)}{\omega_{31}u + \omega_{32}v + \omega_{33}w + \tau_z} + \delta_x \quad (2.12)$$

$$y = \frac{\phi_y(\omega_{21}u + \omega_{22}v + \omega_{23}w + \tau_y)}{\omega_{31}u + \omega_{32}v + \omega_{33}w + \tau_z} + \delta_y \quad (2.13)$$

Postoje dvije vrste parametara u ovom modelu: unutrašnji (eng. intrinsic) i vanjski (eng. extrinsic). Unutrašnji parametri ($\phi_x, \phi_y, \gamma, \delta_x, \delta_y$) opisuju samu kameru, a vanjski (Ω, τ) opisuju

poziciju i orijentaciju kamere u realnom svijetu. Unutrašnji parametri se najčešće zapisuju u formi matrice, kao što je dato relacijom (2.14).

$$\Lambda = \begin{bmatrix} \phi_x & \gamma & \delta_x \\ 0 & \phi_y & \delta_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Nalaženje ovih parametara predstavlja glavni problem kalibracije. Da bi se dobile linearne projekcije jednačine potrebno je promijeniti reprezentaciju koordinata $[x \ y]^T$ i $[u \ v \ w]^T$ u homogene koordinate.

Homogene koordinate tačke \mathbf{x} obilježavaju se sa $\tilde{\mathbf{x}}$, i date su relacijom (2.15), gdje je λ proizvoljni faktor skaliranja.

$$\tilde{\mathbf{x}} = \lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.15)$$

Slično, homogene koordinate tačke \mathbf{x} date su relacijom (2.16).

$$\tilde{\mathbf{w}} = \lambda \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} \quad (2.16)$$

Model kamere u homogenim koordinatama je dat sa (2.17).

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \phi_x & \gamma & \delta_x & 0 \\ 0 & \phi_y & \delta_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \tau_x \\ \omega_{21} & \omega_{22} & \omega_{23} & \tau_y \\ \omega_{31} & \omega_{32} & \omega_{33} & \tau_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} \quad (2.17)$$

Model u matričnom obliku je predstavljen sa (2.18), [3].

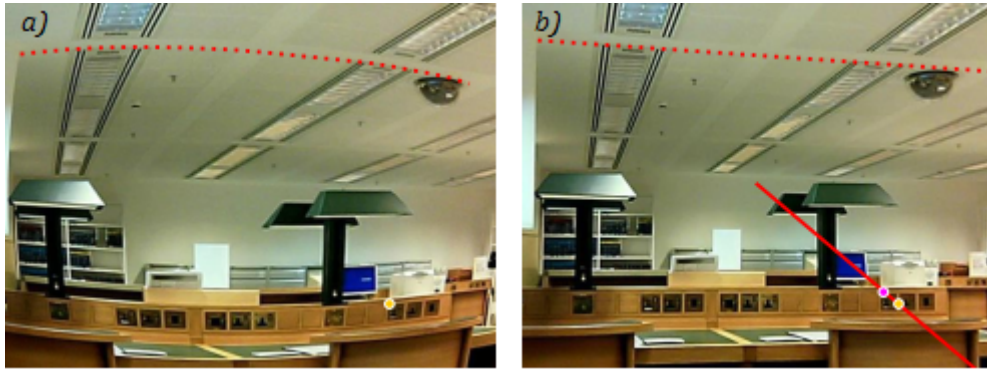
$$\lambda \tilde{\mathbf{x}} = \begin{bmatrix} \Lambda & 0 \end{bmatrix} \begin{bmatrix} \Omega & \tau \\ 0^T & 1 \end{bmatrix} \tilde{\mathbf{w}} \quad (2.18)$$

2.2 Radijalna distorzija

U pinhole modelu kamere izuzima se činjenica da stvarne kamere gotovo uvijek imaju leće (sočiva) koje skupljaju svjetlost koja dolazi izvana i fokusira je na ravan slike. Leće dovode do raznih devijacija od standardnog pinhole modela. Jedna od devijacija je i tzv. dvodimenzionalno savijanje (*eng. 2D warping*) kod kojeg tačke odstupaju od njihovih očekivanih pozicija i "kreću" se duž radijalnih linija povučenih iz centra slike. Ova pojava se naziva radijalna distorzija (*eng. radial distortion*) i ovisi od udaljenosti tačke od centra. Ona se može veoma jednostavno uočiti na slici, jer se linije koje su ravne u stvarnom svijetu na slici prikazuju zakrivljene, što je vidljivo na slikama 2.4 a) i 2.4 b). U stvarnosti, radijalna distorzija se dešava kada je vidno polje leća veliko.

Radialna distorzija se obično modelira kao polinomijalna funkcija udaljenosti od centra r . Konačna pozicija (x', y') se može odrediti pomoću originalne pozicije (x, y) , relacijama (2.19) i (2.20), u kojima parametri β_1 i β_2 predstavljaju stepen distorzije.

$$x' = x(1 + \beta_1 r^2 + \beta_2 r^4) \quad (2.19)$$



Slika 2.4: Slika 2.4. a) Slika sa distorzijom; b) Slika na kojoj je uklonjena distorzija [3]

$$y' = y(1 + \beta_1 r^2 + \beta_2 r^4) \quad (2.20)$$

Za tačne rezultate obrade slike, potrebno je uzeti u obzir prethodno opisanu pojavu [3].

2.3 Kalibracija kamere u *OpenCV*-u

OpenCV obezbeđuje gotove funkcije potrebne za kalibraciju kamere. U razvoju aplikacije koristit će se funkcija ***cv::calibrateCamera*** koja kao ulaz prima vektor vektora koordinata tačaka (dvodimenzionalnu matricu koordinata tačaka) kalibracionog uzorka u koordinatnog sistemu koji je postavljen u stvarnom svijetu. Drugi argument ove funkcije je vektor vektora projekcija pomenutih tačaka na ravan slike. Treći parametar predstavlja dimenzije slike. Kao četvrti i peti parametar proslijeđuju se matrice u koje će funkcija upisati unutrašnje (*eng. intrinsic*) parametre i koeficijente distorzije, respektivno. Pored pomenutih argumenata koji su nužni, u funkciju se mogu proslijediti i neki kriteriji kojima se može specificirati način izračunavanja unutrašnjih parametara i koeficijenata distorzije. Funkcija vraća finalnu grešku ponovne projekcije (*eng. reprojection error*) koja je tipa *double*.

Matrice unutrašnjih parametara i distorzionih koeficijenata se proslijeđuju kao treći i četvrti parametar funkciji ***cv::undistort***. Prvi parametar je slika sa koje je potrebno ukloniti distorziju, dok se kao drugi parametar proslijeđuje slika u koju će se smjestiti prvi parametar nakon uklanjanja distorzije. Ova funkcija ne vraća ništa (povratni tip je *void*).

Nakon uklanjanja distorzije, potrebno je dobiti perspektivnu transformaciju.

Funkcija ***cv::getPerspectiveTransform*** prima dva argumenta. Prvi argument je vektor koordinata četiri tačke na slici. Drugi argument je vektor koordinata korespondirajućih tačkaka u odredišnoj slici (koordinate u koje će se preslikati tačke iz prvog argumenta). Rezultat ove funkcije je matrica perspektivne transformacije koja se primjenjuje na sliku pomoću funkcije ***cv::warpPerspective***. Ona kao ulaz prima sliku na koju je potrebno primijeniti transformaciju, sliku u koju će funkcija smjestiti rezultat, matricu perspektivne transformacije i dimenzije rezultatne slike, respektivno. Povratni tip ove funkcije je *void* [12].

Poglavlje 3

Pretprocesiranje slike

Pretprocesiranje je termin koji se koristi za operacije nad slikom na najnižem nivou apstrakcije. Ono u većini slučajeva dovodi do gubitka informacija koje nosi slika. Međutim, nekada je korisno izgubiti informacije koje nemaju značaj za neki specifični zadatak u obradi i analizi slike, jer je potreban manji memorijski prostor za smještanje slike. Glavni zadatak pretprocesiranja je da se priguše neželjene distorzije ili da se naglase značajke slike koje su važne za dalju obradu. U [13] metode pretprocesiranja su podijeljene u četiri kategorije: transformacije svjetline piksela, geometrijske transformacije, metode pretprocesiranja koje koriste lokalno susjedstvo (*eng. local neighborhood*) i obnova slike [13].

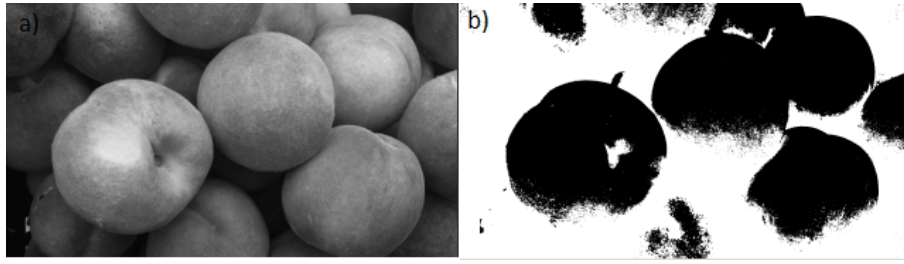
Segmentacija dijeli sliku na njene sastavne dijelove ili objekte. Nivo do kojeg se vrši ovakva podjela slike ovisi od problema koji je razmatran, odnosno segmentacija završava onda kada su objekti od interesa na slici izolirani [14]. Segmentacijom se vrši provjera svakog piksela da bi se ustanovilo da li pripada objektu od interesa ili ne. Rezultat ovog procesa je binarna slika [15]. Generalno, algoritmi za segmentaciju slike su bazirani na dva svojstva intenziteta (vrijednosti koje može poprimiti funkcija slike): diskontinuitet i sličnost. U prvoj kategoriji, cilj je podijeliti sliku na osnovu naglih promjena u intenzitetu. U ovu kategoriju spadaju algoritmi za detekciju tačaka, linija i ivica. U drugoj kategoriji, težnja je da se napravi podjela slike u regije koji slični prema unaprijed definisanom kriteriju. Ovdje spadaju thresholding, regijski bazirana segmentacija itd. U ovom radu detaljnije će biti opisane neke metode metode segmentacije.

3.1 Segmentacija pragom (*Thresholding*)

Najjednostavniji način segmentacije sastoji se u nalaženju piksela čiji se intenzitet svjetlosti (*eng. brightness*) nalazi u specificiranom opsegu. Svi pikseli koji su van opsega se smatraju pozadinom. Rezultat ovog načina segmentacije je najčešće binarna slika, sa crnim i bijelim regijama. Ne postoji općeprihvaćena konvencija koje regije predstavljaju regije od interesa [5]. Matematički, segmentacija pragom predstavlja transformaciju funkcije slike f u binarnu sliku g na način dat formulom (3.1), gdje je T prag segmentacije, $g(i, j)$ je 1 za piksele koji pripadaju objektima, a 0 za piksele koji pripadaju pozadini. Na slici 3.1 može se vidjeti rezultat primjene ove segmentacije.

$$g(i, j) = \begin{cases} 1 & \text{za } f(i, j) \geq T \\ 0 & \text{za } f(i, j) < T \end{cases} \quad (3.1)$$

Ova metoda je pogodna ako se objekti ne dodiruju i ako se jasno razlikuju od pozadine [13].



Slika 3.1: a) Slika u prostoru sivih tonova; b) Slika nakon segmentacije pragom

3.2 Detekcija ivica

Detektori ivica lociraju piksele gdje funkcija slike naglo mijenja vrijednost. Ovaj postupak do-
vodi do značajnog smanjenja informacija koje slika nosi, ali u većini slučajeva kada se sprovodi,
informacije koje se gube nisu bitne za dalju obradu i analizu.

3.2.1 Detekcija ivica zasnovana na gradijentu

Ivica je svojstvo individualnog piksela i računa se pomoću vrijednosti funkcije slike u njegovoj
okolini. Ona predstavlja vektorsku varijablu koja se sastoji od dvije komponente: amplitude i
smjera (*eng. direction*). Amplituda je zapravo amplituda gradijenta koja se računa prema izrazu
(3.2), a smjer predstavlja smjer gradijenta zakrenut za -90° . Smjer gradijenta ψ daje smjer
maksimalnog rasta funkcije i računa se pomoću relacije (3.3), gdje $f(x,y)$ predstavlja funkciju
slike, a $\arg(x, y)$ predstavlja ugao koji zaklapa prava koja prolazi kroz koordinatni početak i
kroz (x, y) , sa x -osom.

$$|\text{grad}f(x,y)| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (3.2)$$

$$\psi = \arg\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) \quad (3.3)$$

Ukoliko je bitno samo poznavanje amplituda ivice, obično se koristi linearni diferencijalni ope-
rator Laplacian [13].

Laplacian se definiše izrazom (3.4) i invarijantan je na rotaciju, a time i na smjer.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3.4)$$

Ovaj operator naglašava tačke, linije i ivice te izoštrava sliku, a prigušava uniformne i blago
promjenljive regije.

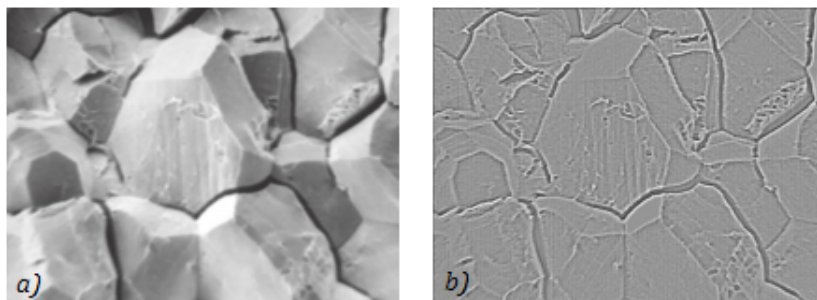
Laplacian se može izračunati pomoću konvolucione matrice (*eng. kernel*). Njegova klasična
3x3 konvoluciona matrica prikazana je izrazom (3.5).

$$L = \begin{bmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (3.5)$$

Ova matrica oduzima vrijednosti funkcije slike osam susjednih piksela od vrijednosti funkcije
srednjeg piksela pomnožene sa 8. Ako se ova matrica primjeni na regiju ujednačenog inten-
ziteta svjetlosti (vrijednosti funkcije slike), rezultat će biti jednak nuli. Kada je u susjedstvu

prisutna varijacija u formi tačke, linije ili ivice, Laplacian ima nenultu vrijednost. Ta vrijednost može biti ili pozitivna ili negativna, u ovisnosti od toga sa koje strane ivice (ili tačke) leži centralni piksel. Kako bi se prikazali pikseli koji nakon primjene ovog operatora imaju negativnu vrijednost, uobičajeno je da se doda vrijednost srednjeg sivog nivoa (ako crna ima vrijednost 0, a bijela 255, srednja siva ima vrijednost 128).

Rezultat primjene ovog operatora na sliku 3.3 a) je prikazan na slici 3.3 b) [5].



Slika 3.2: a) Originalna slika; b) Rezultat primjene Laplaciana [5]

Postoji još operatora koji su bazirani na gradijentu, kao što su: Robertsov, Prewittov, Sobelov, Robinsonov te Kirschov operator. Više o ovim operatorima može se naći u [13].

3.2.2 Canny metoda detekcije ivica

Canny metoda predstavlja optimalnu metodu za detekciju ivica. Njena optimalnost se ogleda u odnosu na tri kriterija:

- važne ivice moraju biti detektovane i ne smije biti lažnih detekcija;
- udaljenost između stvarne i locirane ivice treba biti minimalna;
- detektor treba da svaku ivicu detektuje jednom, odnosno ne smije postojati više odziva na jedinstvenu ivicu [13].

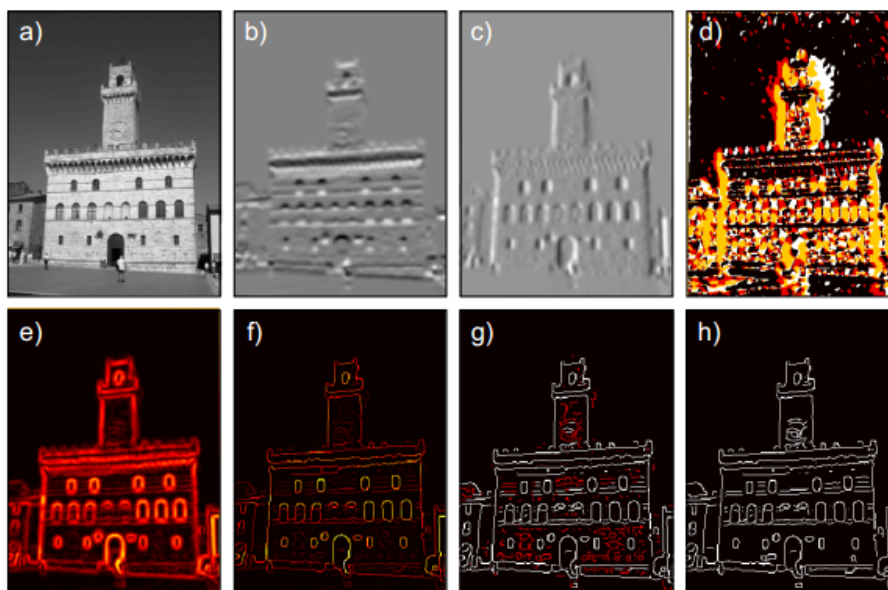
Da bi se detektovale ivice na slici, potrebno je prvo zamagliti sliku, te izvršiti konvoluciju funkcije slike sa parom ortogonalnih derivacijskih filtera (npr. sa Prewittovim filterima), kako bi se dobile dvije slike koje sadrže derivacije u horizontalnim i vertikalnim smjerovima, respektivno.

Jednostavan pristup bio bi izračunavanje amplitude piksela (x , y), te dodjeljivanje ivice tom pikselu ukoliko je amplituda veća od određene kritične vrijednosti. Ovo se označava kao *thresholding* i u većini slučajeva daje nedovoljno dobre rezultate jer amplituda ima velike vrijednosti na ivici, ali i na susjednim pozicijama.

Da bi se izbjegli nepoželjni odzivi, Canny metoda koristi supresiju nemaksimuma (*eng. non-maximum suppression*). Orijentacija gradijenta se kvantizuje tako da može poprimiti jednu od četiri vrijednosti: 0° , 45° , 90° ili 135° . Amplituda piksela postaje nula ako bilo koji od susjednih piksela koji su vertikalni u odnosu na gradijent ima veću vrijednost. Na ovaj način zadržavaju se gradijenti na ivici, ali ne i oni na susjednim pozicijama. Sada je moguće dobiti ivice poređenjem preostalih nenulatih amplituda sa fiksiranom vrijednosti.

Međutim, i dalje ivice neće biti u potpunosti tačno detektovane (neki dio ivice će biti izgubljen, a negdje gdje nije ivica, ovaj metod će pokazati da jeste). Zbog toga se uzima u obzir činjenica da su ivice u stvarnom svijetu kontinualne, te se definiraju dva praga: viši i niži. Svi pikseli čija je amplituda iznad višeg praga se smatraju ivicom, s tim da se uzima u obzir da tu spada i

nekoliko piksela koji nisu zapravo ivice. Pikseli koji imaju amplitudu iznad nižeg praga i koji su povezani sa postojećom ivicom (definisanom gornjim pragom) se također smatraju ivicom [3]. Na slici 3.4 je prikazano sprovođenje ove metode.



Slika 3.3: a) Originalna slika; b) Rezultat vertikalnog Prewittovog filtera; c) Rezultat horizontalnog Prewittovog filtera; d) Mapa kvantizirane orijentacije; e) Mapa amplituda gradijenta; f) Amplitude nakon supresije ne-maksimuma; g) Dva praga: crveni pikseli su iznad nižeg, a ispod višeg nivoa; h) Određene ivice [3]

3.3 Regijski bazirana segmentacija

Važno svojstvo regija je njihova homogenost koja predstavlja glavni kriterij kod ove segmentacije. Ovaj kriterij može biti baziran na homogenosti u smislu razina sive, boja, teksture itd. Sprovodi se pomoću tzv. rasta regija (*eng. region growing*). Jednostavnije verzije rasta regija su spajanje regija (*eng. region merging*) i razdvajanje regija (*eng. region splitting*).

Kod spajanja regija prvo je potrebno definisati početnu metodu za segmentiranje slike u mnogo malih regija koje zadovoljavaju određeni uslov homogenosti. Zatim se definiše kriterij koji regije moraju zadovoljiti da bi se spojile. Nakon što više ne postoje dvije regije koje zadovoljavaju kriterij spajanja, potrebno je stati sa procesom. Početna segmentacija se može izvršiti tako da svaki piksel predstavlja zasebnu regiju ili da se koriste regije od 2x2, 4x4 ili 8x8 piksela.

Razdvajanje regija je suprotno spajanju regija i počinje sa jednom regijom koja predstavlja cijelu sliku. Definiše se kriterij dijeljenja i regije se dijele na više njih dok god je uslov dijeljenja zadovoljen. Iako se ovaj pristup čini kao dualan spajanju regija, ove dvije metode ne daju istu segmentaciju čak i kad se koristi isti kriterij homogenosti.

Ove metode se najčešće kombinuju kako bi se iskoristile prednosti i jednog i drugog pristupa [13].

Objekti na slikama se često dodiruju ili preklapaju što može predstavljati problem pri segmentaciji. Metoda koja se obično koristi pri razdvajanju objekata koji se dodiruju naziva se watershed segmentacija [5]. Postoje dva osnovna pristupa watershed segmentaciji slike. Kod prvog, potrebno je naći nizvodni (*eng. downstream*) put od svakog piksela do lokalnog minimuma

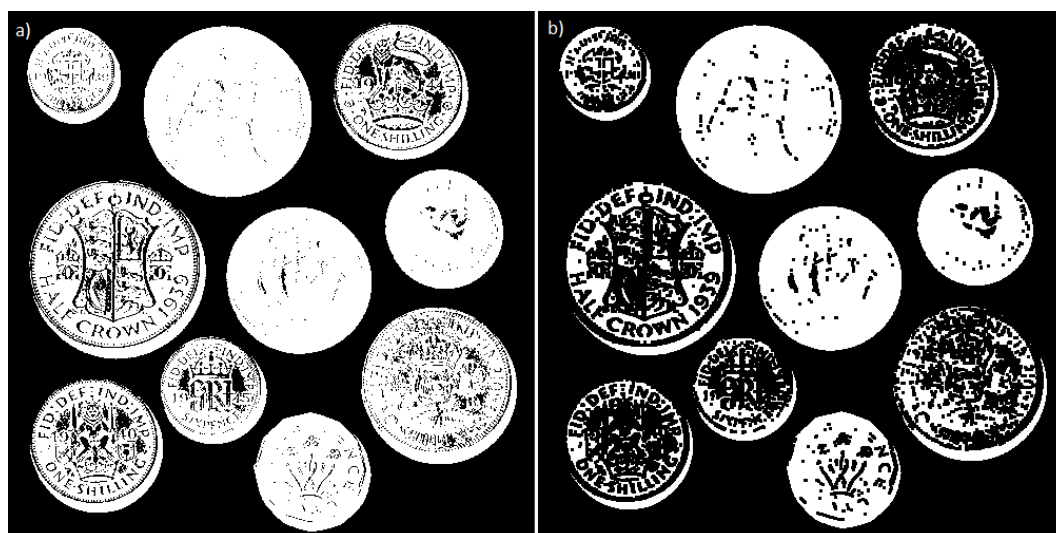
funkcije slike. Sliv (*eng. catchment basins*) se definiše kao skup piksela čiji nizvodni putevi završavaju u istom minimumu. Međutim, za digitalne površine ne postoji pravilo za jednoznačno određivanje nizvodnih puteva, te zahtjeva dosta računanja. Drugi pristup je praktičniji; umjesto nalaženja nizvodnih puteva, slivovi se "popunjavaju" odozdo. Svaki sliv predstavlja jedan lokalni minimum. Ako se slika zamisli kao topografska površina koja se uranja u vodu, svi slivovi koji se nalaze ispod nivoa vode, se počinju puniti. Ukoliko se dva sliva spoje, pravi se brana duž "uzvišenja" na slici, koja predstavlja watershed liniju[13].

3.4 Morfološka obrada slike

Rezultat segmentacije slike ponekad nije dovoljno zadovoljavajući za dalju obradu. Morfološke operacije se provode nad slikom nakon same segmentacije, kako bi se istakla tražena svojstva. U ovom poglavlju će detaljnije biti riječi o eroziji i dilataciji. Erozija i dilatacija su najkorištenije operacije nad binarnim slikama koje se obično dobijaju kao rezultat segmentacije. Predstavljaju operacije dodavanja ili oduzimanja piksela prema određenim pravilima, koja ovise od susjednih piksela.

3.4.1 Erozija

Erozija (*eng. erosion*) ima za cilj da ukloni piksele koji ne bi trebali biti tu. Za primjer se može uzeti segmentacija pragom, koja u većini slučajeva ne daje sasvim tačne rezultate, nego postoje pikseli koji se smatraju objektom, a zapravo pripadaju pozadini i obrnuto. Te piksele je potrebno ukloniti, za što se često koristi erozija. Najjednostavniji način erozije je da se "isključe" oni pikseli koji se dodiruju sa pozadinom. Ovo dovodi do smanjenja dimenzija objekta što može predstavljati problem, a u nekih slučajevima dolazi i do rastavljanja objekta na manje dijelove [5]. Primjer erozije dat je na slici 3.5.



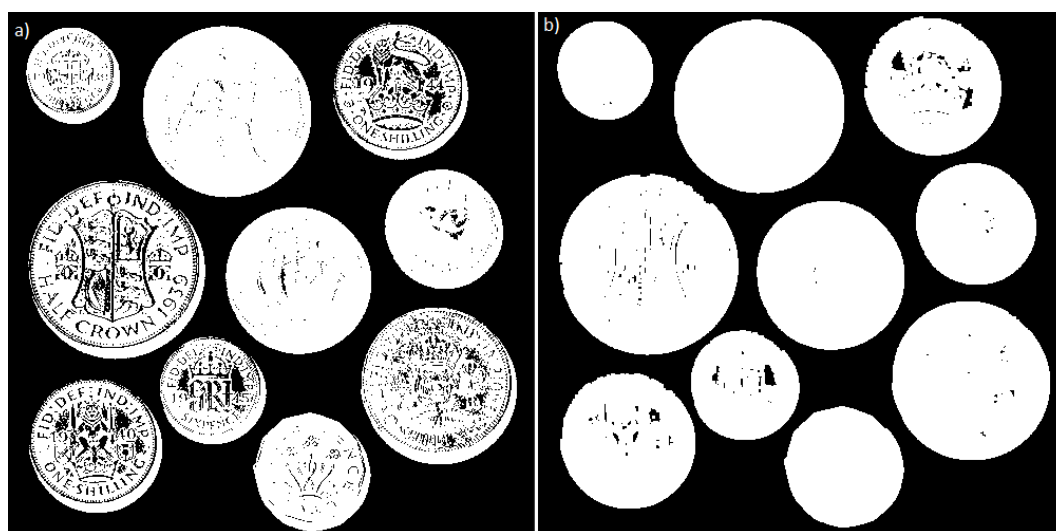
Slika 3.4: a) Binarna slika; b) Slika nakon erozije

3.4.2 Dilatacija

Komplementarna operacija eroziji je dilatacija (*eng. dilation*), koja se koristi za dodavanje piksela. Ova operacija se obično izvodi tako da dodaje "sloj piksela" oko svih objekata i regija, te popunjava male šupljine u njima ako postoje. Problemi se mogu javiti zbog povećanja dimenzija.

Ove operacija se obično koriste zajedno, a redoslijed primjene ovisi o željenom rezultatu. Primjena erozije pa dilatacije na sliku se naziva otvaranje (*opening*), a primjena dilatacije pa erozije predstavlja zatvaranje (*eng. closing*). U nekim slučajevima, otvaranje se može primjeniti za razdvajanje objekata koji su veoma blizu, pa su se nakon segmentacije spojili [5].

Dilatacija je prikazana na slici 3.6.



Slika 3.5: a) Binarna slika; b) Slika nakon dilatacije

3.5 Segmentacija u *OpenCV*-u

Metoda segmentacije koja će se koristiti u razvoju aplikacije je segmentacija pragom. U *OpenCV*-u postoji funkcija *cv::threshold*. Prvi argument ove funkcije je slika (sa jednim kanalom) koju je potrebno segmentirati. Drugi argument je binarna slika u koju će funkcija smjestiti rezultat segmentacije pragom. Vrijednost praga segmentacije se proslijeđuje kao treći argument. Kao četvrti argument proslijeđuje se maksimalna vrijednost koja ovisi od tipa segmentacije pragom, koji se proslijeđuje kao peti argument. Više o ovoj kao i o drugim funkcijama segmentacije (*cv::Canny*, *cv::watershed*) može se naći u [12].

Nakon segmentacije pristupa se morfološkoj obradi slike. *OpenCV* funkcije *cv::erode* i *cv::dilate* vrše eroziju odnosno dilataciju nad slikom koja je proslijeđena kao prvi argument funkcije. Drugi argument funkcije je slika u koju će funkcija upisati rezultat primjene erozije odnosno dilatacije. Kao treći argument se proslijeđuje element za strukturiranje (*eng. structuring element*, obično se generiše pomoću funkcije *cv::getStructuringElement()*). Četvrti argument je tačka koja predstavlja "sidro" strukturiranog elementa. Zadnja tri argumenta su redom: broj iteracija koliko će se erozija odnosno dilatacija primjeniti, metoda ekstrapolacije piksela i vrijednost granice u slučaju konstantne granice. Funkcije ne vraćaju ništa [12].

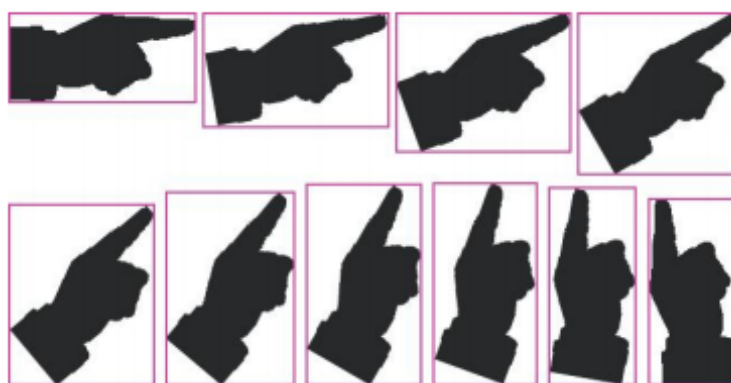
Poglavlje 4

Određivanje pozicije i orijentacije

Pozicija i orijentacija objekta na slici su često potrebne za interpretaciju informacija koje su rezultat procesiranja slike. Ukoliko su dovoljno dobro nađene ivice objekta, nalazak pozicije i orijentacije predstavlja čist matematički problem. Pozicija kao i orijentacija nepravilnih objekata može se definirati na više načina, od kojih će neki biti obrađeni u nastavku.

4.1 Određivanje pozicije

Pod određivanjem pozicije podrazumijeva se nalaženje koordinata tačke koja je važna za objekat čiju je poziciju potrebno odrediti. Postoji više ovakvih tačaka i načina njihovog određivanja. Najjednostavniji način je određivanje minimalne i maksimalne granice objekta na slici. One se određuju nalaženjem piksela sa najmanjim odnosno najvećim koordinatama u horizontalnom i vertikalnom smjeru. Ove granične koordinate definiraju pravougaonik u kojem se nalazi cijeli objekat. Kao tačka koja označava poziciju objekta, uzima se tačka presjeka dijagonala pravougaonika. Ovakvo određivanje pozicije je neprecizno jer se nalaženje tačke bazira samo na koordinatama nekoliko piksela. Također postoji problem orijentacije objekta; pravougaonik ne mora biti isti za različite orijentacije istog objekta, što se može primjetiti na slici 3.4.



Slika 4.1: *Pravougaonik se mijenja sa orijentacijom objekta [5]*

Druga tačka koju možemo koristiti za određivanje pozicije je centar granične kružnice. Efikasan način za nalaženje ove kružnice je prethodno nalaženje mnogougla u kojem se nalazi cijeli objekat te opisivanje kružnice oko njega. Za objekte koji su šupljikasti ili su neobičnog izgleda,

ova tačka ne mora biti na samom objektu. Za ovakve objekte poželjno je uzeti u obzir koordinate svih piksela koji predstavljaju objekat. Na ovaj način se dobija centroid, tačka čije su x i y koordinate date relacijama (3.4) i (3.5), respektivno, gdje je P broj piksela koji predstavljaju objekat.

$$C_x = \frac{\sum_i x_i}{P} \quad (3.4)$$

$$C_y = \frac{\sum_i y_i}{P} \quad (3.5)$$

Ukoliko se u obzir uzmu samo pikseli koji definiraju ivicu, koristeći relacije (3.4) i (3.5) neće ponekad se ne dobiju ispravne koordinate centroida, ukoliko je jedan dio ivice kompleksniji (sadrži više piksela) od ostatka ivice. Neka ivicu objekta definiše n piksela i neka su njihove koordinate date sa (x_i, y_i) , gdje je $i = 0, 1, \dots, n$ i tačke (x_0, y_0) i (x_n, y_n) predstavljaju jednu istu tačku. Tada se koordinate centroida za ovaj slučaj mogu izračunati pomoću relacija (3.6) i (3.7).

$$C_x = \frac{\sum_i (x_i + x_{i-1})^2 (y_i - y_{i-1})}{P} \quad (3.6)$$

$$C_y = \frac{\sum_i (y_i + y_{i-1})^2 (x_i - x_{i-1})}{P} \quad (3.7)$$

U ovom slučaju P se računa prema (3.8).

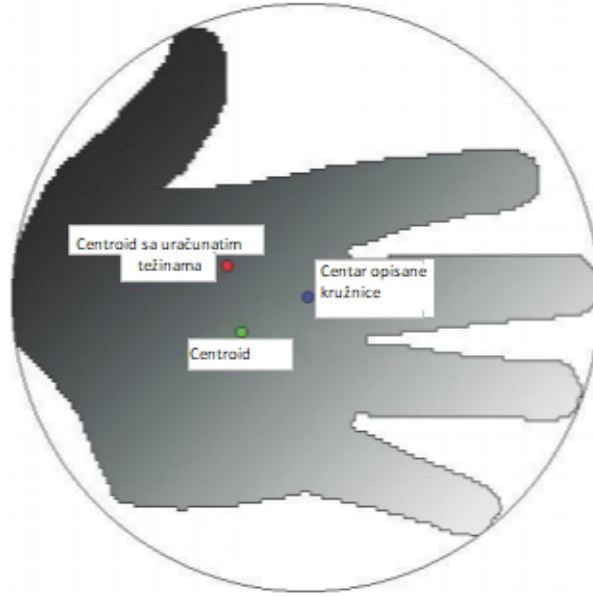
$$P = \frac{\sum_i (x_i + x_{i-1})(y_i - y_{i-1})}{2} \quad (3.8)$$

Kod nalaženja koordinata centroida, svaki piksel se uzima sa jednakom važnošću. U nekim slučajevima neki pikseli su važniji od ostalih. Na primjer, kao kriterij "težine" piksela može se uzeti jačina svjetlosti, pa da tamniji pikseli imaju veću vrijednost od svjetlijih i obrnuto. Svakoј razini sive se pridružuje "težina" i ona se uzima u obzir prema relacijama (3.9) i (3.10), gdje v_i predstavlja težinu piksela (x_i, y_i) [5].

$$C_x = \frac{\sum_i v_i x_i}{\sum_i v_i} \quad (3.9)$$

$$C_y = \frac{\sum_i v_i y_i}{\sum_i v_i} \quad (3.10)$$

Sve navedene tačke prikazane su na slici 3.5.



Slika 4.2: Centroid, centar opisane kružnice i centroid sa uračunatim težinama [5]

4.2 Određivanje orijentacije

Određivanje orijentacije objekta predstavlja puno veći izazov od određivanja pozicije, pogotovo za simetrične objekte kod kojih određivanje orijentacije nekada nije ni moguće. Međutim, čak i za objekte čija je orijentacija jasna ljudskom oku, njeno nalaženje može biti problem u smislu zahtjeva na tačnost određene orijentacije i brzinu izračunavanja [16]. U nastavku će biti razmotrene neke metode nalaženja orijentacije.

4.2.1 Osa najmanje inercije

Svaki objekat posjeduje osu za koju ima najmanji moment inercije, i time može stabilno rotirati oko nje [16]. To je linija kod koje je suma kvadrata rastojanja ose inercije i tačaka konture minimalna [5]. Najpogodniji način za dobijanje orijentacije je izračunavanjem elemenata tenzora inercije prema relacijama (3.11), (3.12) i (3.13).

$$M_{xx} = \sum_i x_i^2 - \frac{(\sum_i x_i)^2}{P} \quad (3.11)$$

$$M_{yy} = \sum_i y_i^2 - \frac{(\sum_i y_i)^2}{P} \quad (3.12)$$

$$M_{xy} = \sum_i x_i y_i - \frac{(\sum_i x_i)(\sum_i y_i)}{P} \quad (3.13)$$

Orijentacija se može predstaviti uglom koji zaklapa osa inercije sa x -osom u pozitivnom smjeru od horizontale. Taj ugao se može naći korištenjem relacije (3.14).

$$\theta = \arctan\left(\frac{M_{xx} - M_{yy} + \sqrt{(M_{xx} - M_{yy})^2 + 4M_{xy}^2}}{2M_{xy}}\right) \quad (3.14)$$

Korištenje ove metode može izazvati probleme kod objekata koji su simetrični. Kod osno simetričnih objekata može se desiti da proračun ne bude jasan, odnosno da prednja i zadnja strana objekta nisu jednoznačno određene. Objekti koji su simetrični u odnosu na jednu tačku imaju oblik sličan ili isti krugu ili kvadratu. Mala promjena njihove konture dovodi do velikog zakretanja ose najmanje inercije.

Da bi se izbjegli ovi problemi potrebno je ispuniti sljedeće uslove:

- objekat treba da je dovoljno dug toliko da se primarna i sekundarna osa mogu razlikovati;
- objekat reba da bude jasno asimetričan, potrebno je moći odrediti prednju i zadnju stranu objekta [16].

4.2.2 Signatura ugaone udaljenosti

Signatura ugaone udaljenosti (*eng. distance-versus-angle signature*) predstavlja vektor čiji su elementi udaljenosti između tačke na konturi i centroida. Broj elemenata nije fiksno određen; moguće je uzeti 360 tačaka na konturi sa rezolucijom 1° . Signatura ugaone udaljenosti kružnice predstavlja vektor kod koga su svi elementi isti. Određivanje ove signature je jednostavno u teoriji, ali u praksi se nailazi na probleme. Na primjer, ako je objekat geometrijski složen, može se desiti da linija od centroida do tačke na konturi presječe konturu više puta. Tada se tačka na konturi bira (obično se uzme najudaljenija od centra gravitacije).

Glavna prednost signature je što se može koristiti nad objektima koji imaju slične primarni i sekundarni moment inercije, te za izobličene figure. Preciznost ove procedure ne može biti bolja od rezolucije kojom je potpis uziman [16].

Poglavlje 5

Postavka eksperimenta

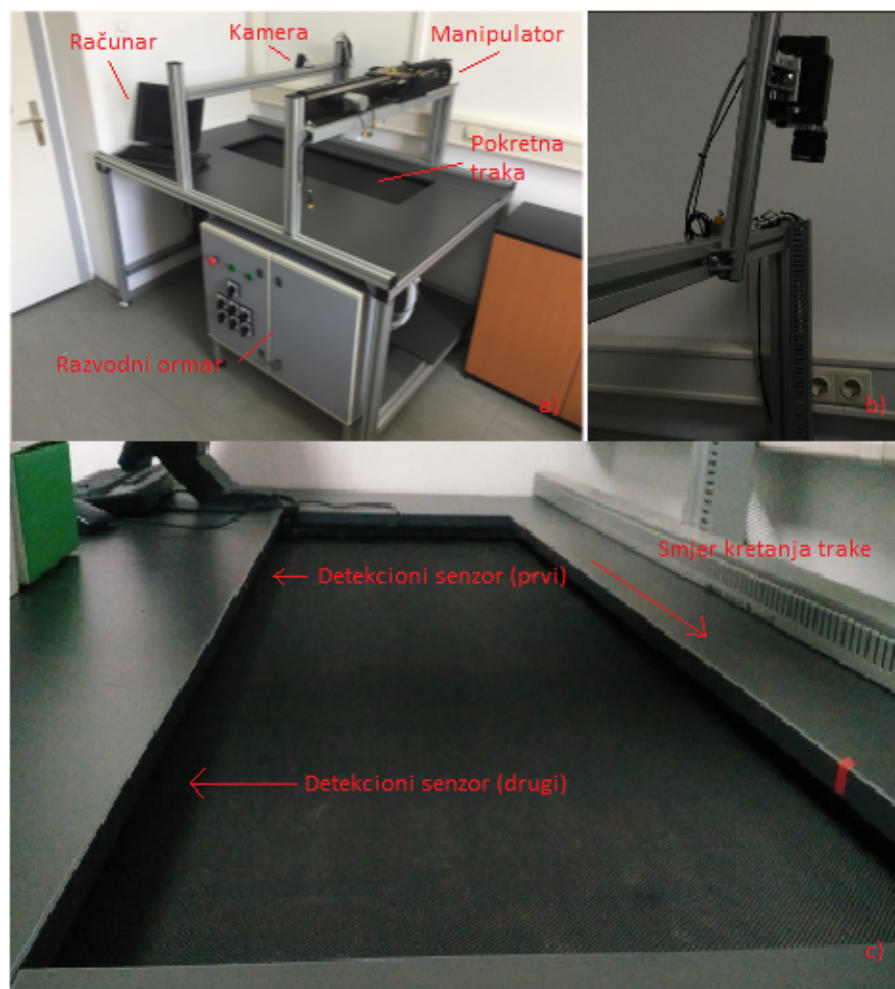
Eksperimentalni dio ovog rada će biti razvijen na ETFCam sistemu. U ovom poglavlju bit će rečeno nešto o ovom sistemu te o njegovim komponentama koje se koriste u eksperimentu. Također će biti opisan način rada aplikacije.

5.1 ETFCam - model industrijskog sistema za automatsku vizuelnu inspekciju

Na slici 5.1 prikazan je izgled ovog sistema. Sistem se sastoji od:

- pokretne trake koju pokreće DC motor,
- računara,
- manipulatora na koji će biti pričvršćen pneumatski gripper koji se uz pomoć dva dvofazna step motora može kretati po x i z osi,
- kamere koja je postavljena iznad trake,
- enkodera i dva senzora prisustva koji su postavljeni sa strane trake,
- programabilnog logičkog kontrolera koji je smješten u razvodnom ormaru.

Ovaj sistem ima mogućnosti obrade slike i upravljanja manipulatorom preko step motora. Izlazni signali iz senzora i enkodera se vode na digitalne ulaze u PLC. Pomoću PLC-a vrši se i upravljanje step motorima. Kamera je spojena sa računarom, čija je glavna uloga obrada slike. Računar i PLC su spojeni pomoću serijske komunikacije. U ovom radu akcenat je stavljen na dio sistema koji je vezan za obradu slike. Na slici 5.2 prikazana je principijelna šema. U nastavku će biti opisane komponente sistema koje učestvuju u eksperimentu.



Slika 5.1: *ETFcam*



Slika 5.2: *Principijelna šema*

5.1.1 Modicon M241 programabilni logički kontroler (PLC)

Modicon TM241CEC24R je PLC kojeg proizvodi Schneider Electric. Ovaj kontroler se može koristiti u mnogim industrijskim aplikacijama. Posjeduje 14 digitalnih ulaza, 10 digitalnih izlaza te port za serijsku komunikaciju. Na slici 5.3 prikazan je izgled ovog kontrolera.



Slika 5.3: Modicon TM241CEC24R PLC [6]

Konfiguracija i programiranje se vrši pomoću SOMachine softvera koji podržava sljedeće načine programiranja:

- instrukcijska lista (IL)
- strukturirani tekst (ST)
- funkcijski blok dijagram (FBD)
- sekvencijalni funkcijski dijagram (SFC)
- ljestvičasti dijagram (LD)

Za potrebe ove aplikacije koristit će se strukturirani tekst i funkcijski blok dijagram. Napajanje je 100...240V AC [6].

5.1.2 Senzor ET318B

Za detekciju objekta na pokretnoj traci na ETFcam sistema ugrađena su dva energetska reflektirajuća svjetlosna skenera ET318B. Izgled senzora prikazan je na slici 5.4. Ovaj senzor koristi crvenu zraku (vidljivi spektar) za detekciju objekta. ET318B može detektovati objekat na udaljenosti 1000 mm ukoliko se koristi aksijalna optika, a 450 mm ukoliko se koristi 90° optika. Senzor ne može detektovati objekat na udaljenosti ispod 1 mm odnosno 5 mm, respektivno. Napaja se sa 10...30V DC. Smješten je u plastičnom kućištu.

Ovaj senzor ima sposobnost učenja (*eng. teach-in*), odnosno prije upotrebe potrebno ga je naučiti na koji način da reaguje na pojavu objekta u "vidnom" području senzora. Učenje se može obaviti pomoću refleksije o objekat ili refleksije o pozadinu (u ovom slučaju pozadina mora biti fiksna). Na senzoru se nalazi dugme za učenje. Postoje dva načina učenja: operativni nivo 1 i 2. U prvom slučaju dugme za učenje se drži sve dok se na LED indikatoru koji se nalazi na senzoru ne pojavi žuto svjetlo. Za drugi način učenja potrebno je držati dugme za učenje sve dok indikator ne počne treperiti (smjenjuju se zelena i žuta svjetlost na indikatoru). Nakon obavljanja ovog koraka senzor je spreman za upotrebu.

Pomoću dugmeta za učenje moguće je podesiti logiku kojom senzor radi. Za to je potrebno pritisnuti dugme dok se na indikatoru ne pojavi zeleno svjetlo, nakon čega će senzor raditi suprotnom logikom od one koja je prethodno bila podešena. Ukoliko se nakon pojave zelenog svjetla na indikatoru javi žuto svjetlo u trajanju od 2s senzor radi u pozitivnoj logici, a ukoliko se na indikatoru pojavljuje naizmjenično zelena i žuta svjetlost u trajanju od 2s senzor radi u negativnoj logici [7].



Slika 5.4: *Senzor ET318B [7]*

5.1.3 Inkrementalni enkoder

Inkrementalni enkoderi se koriste u strojevima i instalacijama. Zahvaljujući njihovom jednostavnom dizajnu, inkrementalni enkoderi su ekonomičniji od apsolutnih. S druge strane, oni samo indiciraju promjenu pozicije i ne mogu detektovati kretanje bez napajanja.

Rotacijsko kretanje se procesira u inkrementalnom enkoderu na način da se ugaoni inkrementi pamte pomoću točka sa fiksnim brojem pulseva po rotaciji. Dio sa integriranom optoelektronikom generira električne signale i na izlazu daje pulseve. Rezolucija enkodera predstavlja broj tamnih segmenata na točku [8]. Izgled enkodera dat je na slici 5.5.

5.1.4 Modbus protokol

Modicon programabilni kontroleri mogu komunicirati jedni s drugima i sa drugih uređaja preko različitih mreža. Podržane mreže uključuju Modicon Modbus i Modbus Plus industrijske mreže



Slika 5.5: *Enkoder [8]*

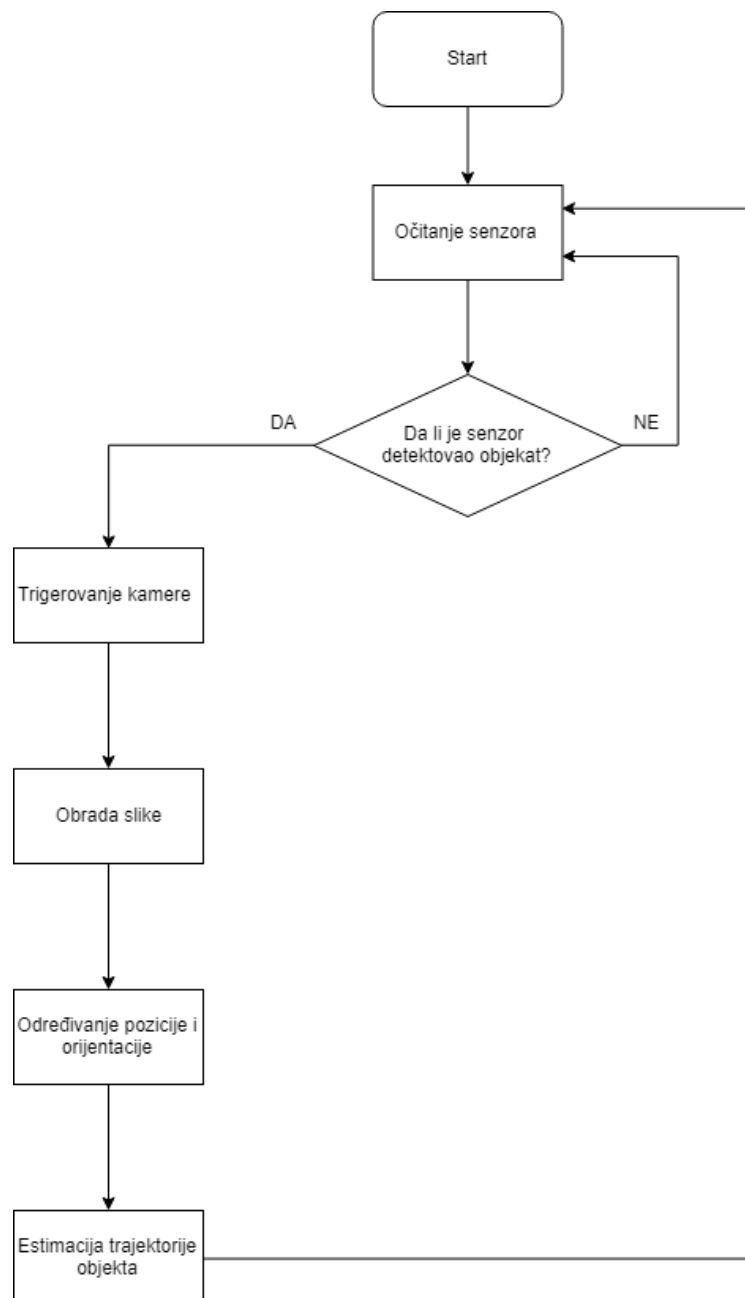
i standardne mreže kao što su MAP i Ethernet. Zajednički jezik korišten od strane svih Modicon kontrolera je Modbus protokol. On definiše strukturu poruke neovisno preko koje mreže se vrši komunikacija.

Za vrijeme komunikacije preko Modbus-a protokol definiše kako će se svaki uređaj znati adresu drugog uređaja sa kojim komunicira, prepoznati poruke koje su mu poslate te iz njih izdvojiti potrebne informacije. Uređaji komuniciraju koristeći master-slave tehniku. Kod ove tehnike samo master može inicirati transakcije. Ostali uređaji (slave uređaji) odgovaraju na zahtjev mastera izvršavajući akciju koja je zahtjevana [17].

U ETFCam sistemu za komunikaciju računara sa PLC-om koristi se ovaj protokol. Računar je master, dok je PLC postavljen kao slave.

5.2 Opis rada aplikacije

Za rad aplikacije su potrebna očitavanja sa senzora i enkodera koji su spojeni na PLC, u kojem se vrši njihovo "pakovanje" u markere koji se šalju preko serijske komunikacije prema računaru. Na računaru, pomoću *OpenCV*-a se vrši njihovo raspakiranje. Ukoliko se prvi senzor aktivirao (što znači da je detektovao objekat), pomoću računara softverski se trigeruje kamera. Na dobijenoj slici se prvo vrši uklanjanje distorzije, perspektivna transformacija i uparivanje koordinatnog sistema slike i stvarnog koordinatnog sistema. Zatim se vrši određivanje pozicije i orijentacije objekata na slici. Tok odvijanja ovog algoritma prikazan je na slici 5.6.



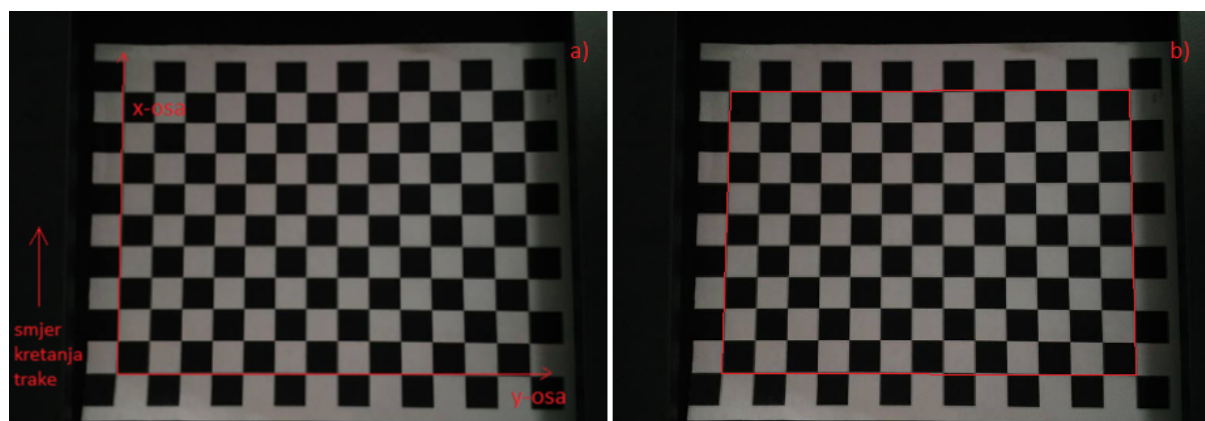
Slika 5.6: Tok odvijanja algoritma

Poglavlje 6

Eksperimentalni rezultati

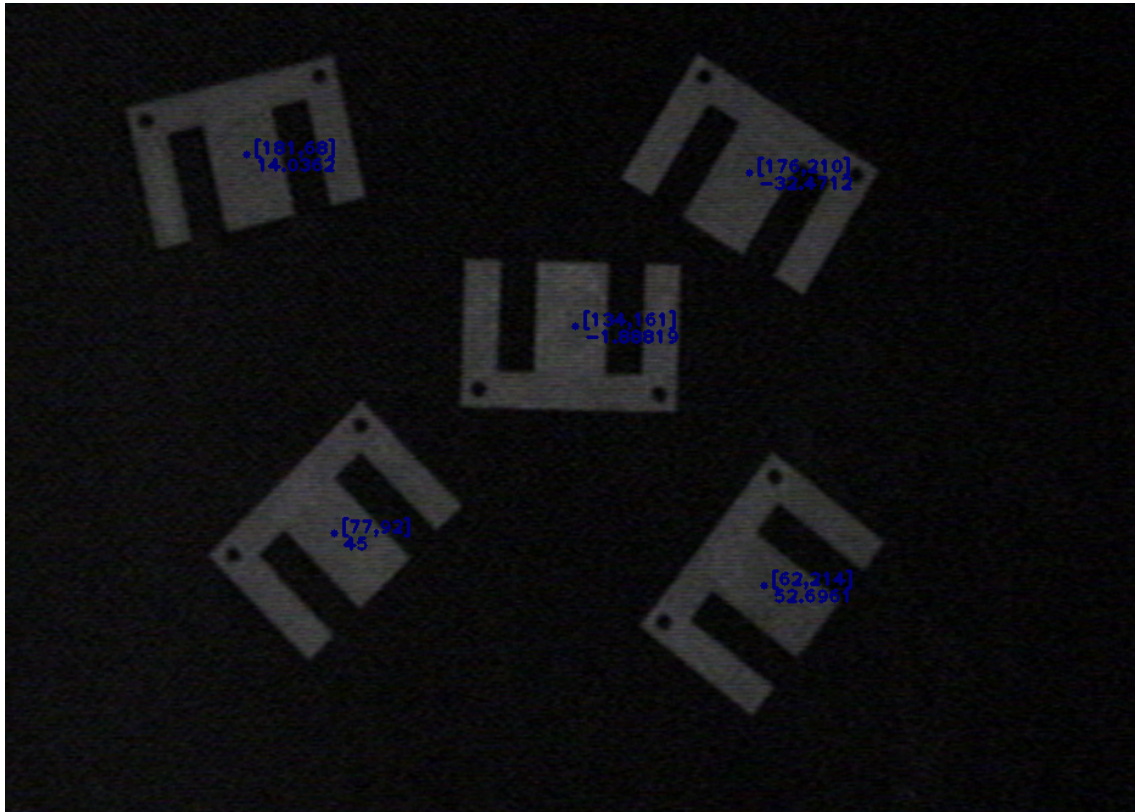
U ovom poglavlju bit će prikazani eksperimentalni rezultati dobijeni primjenom razvijene aplikacije u sklopu ETFcam sistema. Bit će analizirana tačnost određene pozicije i orijentacije na primjeru četiri različita objekta te kašnjenje uzimanja slike u odnosu na brzinu pokretne trake pomoću koje se kreću objekti. Također će biti analizirana mogućnost određivanja trajektorije u budućem vremenu.

Na ploče koje ograničavaju traku postavljeni su crveni indikatori koji pokazuju gdje je na slici donji lijevi i gornji desni ugao. y -osa koordinatnog sistema u stvarnom svijetu postavljena je tako da bude poravnata sa crvenim indikatorom, odnosno da se poklapa sa y -osom koordinatnog sistema na slici. x -osa postavljena je 2.5 cm od ploče. Koordinatni sistem u stvarnom svijetu i područje koje će biti prikazano na slici je prikazano na 6.1.



Slika 6.1: a) Koordinatni sistem postavljen u stvarnom svijetu u odnosu na kojeg se mjeri pozicija; b) područje koje će slika obuhvatiti

Određena pozicija na slikama data je u milimetrima u odnosu na stvarni koordinatni sistem (6.2).



Slika 6.2: Primjer određivanja apsolutne pozicije i orijentacije

6.1 Analiza tačnosti određene pozicije i orijentacije

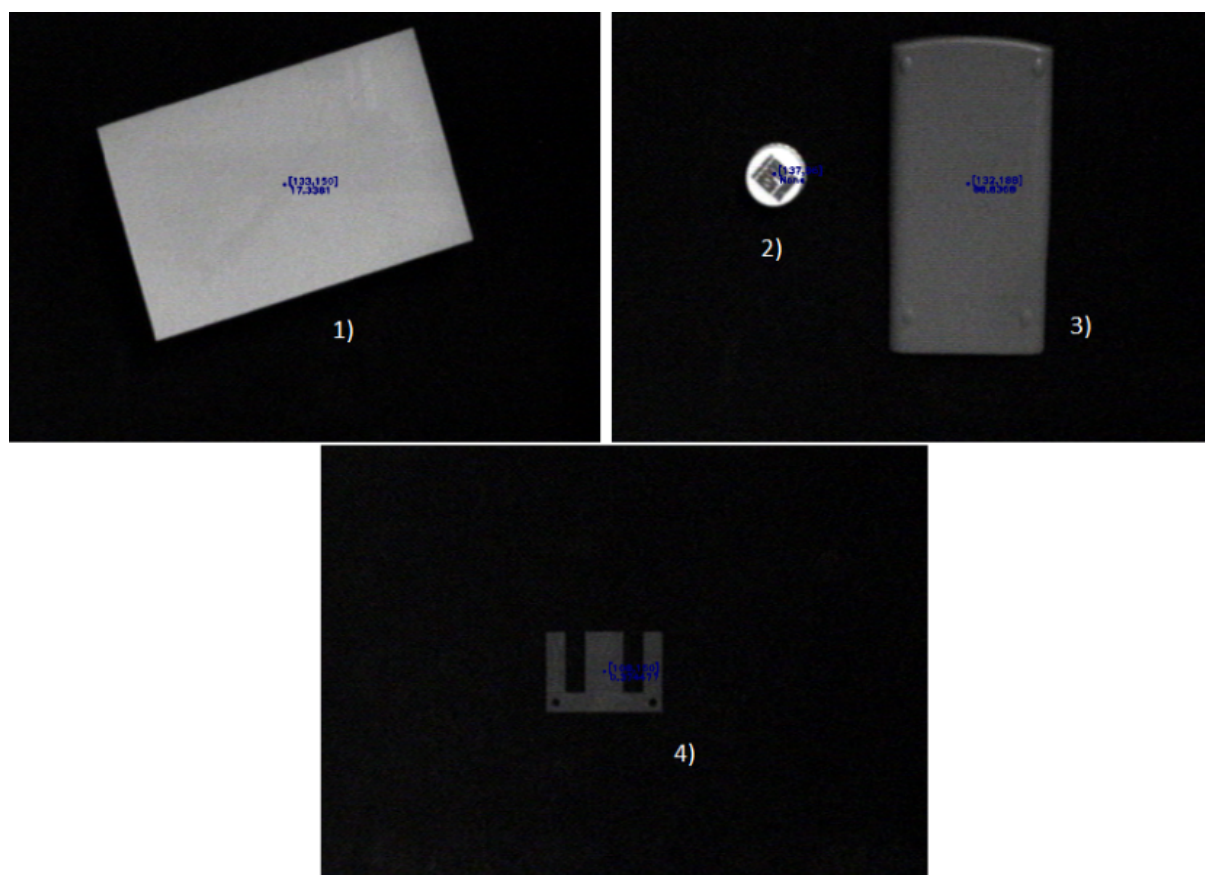
Određivanje tačnosti bit će sprovedeno upoređivanjem stvarne pozicije i orijentacije objekata sa onima dobijenim na slici pomoću apsolutne i relativne greške pojedinačnog mjerenja. Relacije pomoću kojih se računaju ove greške date su izrazima (6.1) i (6.2), respektivno, gdje je $x_{mjereno}$ izmjerena, a $x_{stvarno}$ stvarna vrijednost.

$$\Delta x = x_{mjereno} - x_{stvarno} \quad (6.1)$$

$$g = \frac{\Delta x}{x_{stvarno}} * 100\% \quad (6.2)$$

Na slici 6.3 nalaze se objekti nad kojima je proveden eksperiment.

Rezultati eksperimenta i vrijednosti pomenutih grešaka nalaze se u tabelama 6.1 i 6.2.



Slika 6.3: Objekti nad kojima je proveden eksperiment

Broj mjerjenja	Određena pozicija (mm)	Stvarna pozicija (mm)	Apsolutna greška po x (Δx)	Apsolutna greška po y (Δy)	Relativna greška po x (g_x)	Relativna greška po y (g_y)
1	(135, 152)	(133, 150)	2	2	1.48%	1.31%
2	(137, 89)	(137, 86)	0	-3	0%	3.37%
3	(131, 184)	(132, 188)	-1	-4	-0.76%	-2.17%
4	(105, 145)	(108, 150)	-3	-5	-2.85%	-3.45%

Tabela 6.1: Eksperimentalni rezultati određivanja pozicije i greške mjerenja

Broj mjerjenja	Stvarna orijentacija (°)	Određena orijentacija (°)	Apsolutna greška	Relativna greška
10	17	17	0	0%
12	0	0	0	0%
15	88	88	0	0%
18	0	0	0	0%

Tabela 6.2: Eksperimentalni rezultati određivanja orijentacije i greške mjerenja

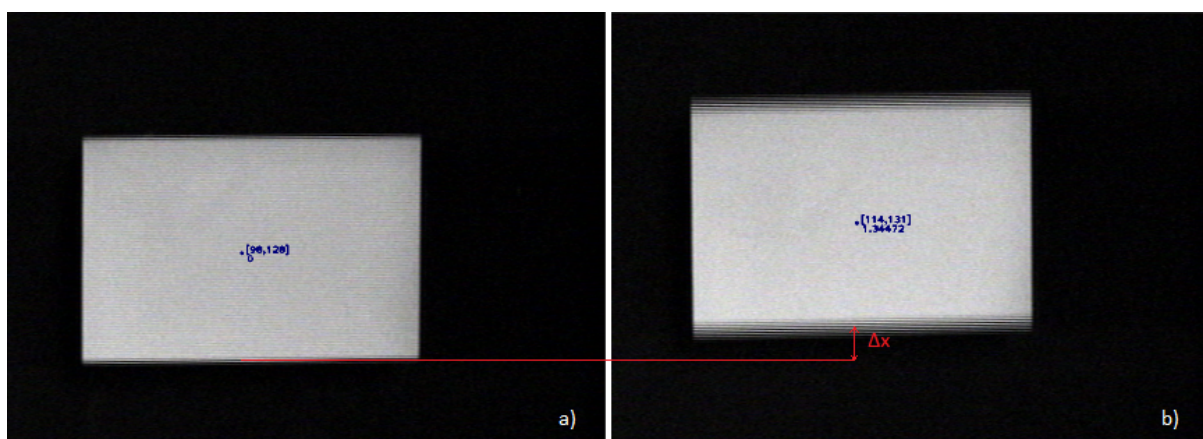
Može se zaključiti da aplikacija radi sa zanemarivim odstupanjima.

6.2 Analiza i estimacija kašnjenja u izvršavanju algoritma pri realnim uslovima

Na slici 6.5 prikazan je isti objekat postavljen na približno istu poziciju slikan pri različitim brzinama trake.

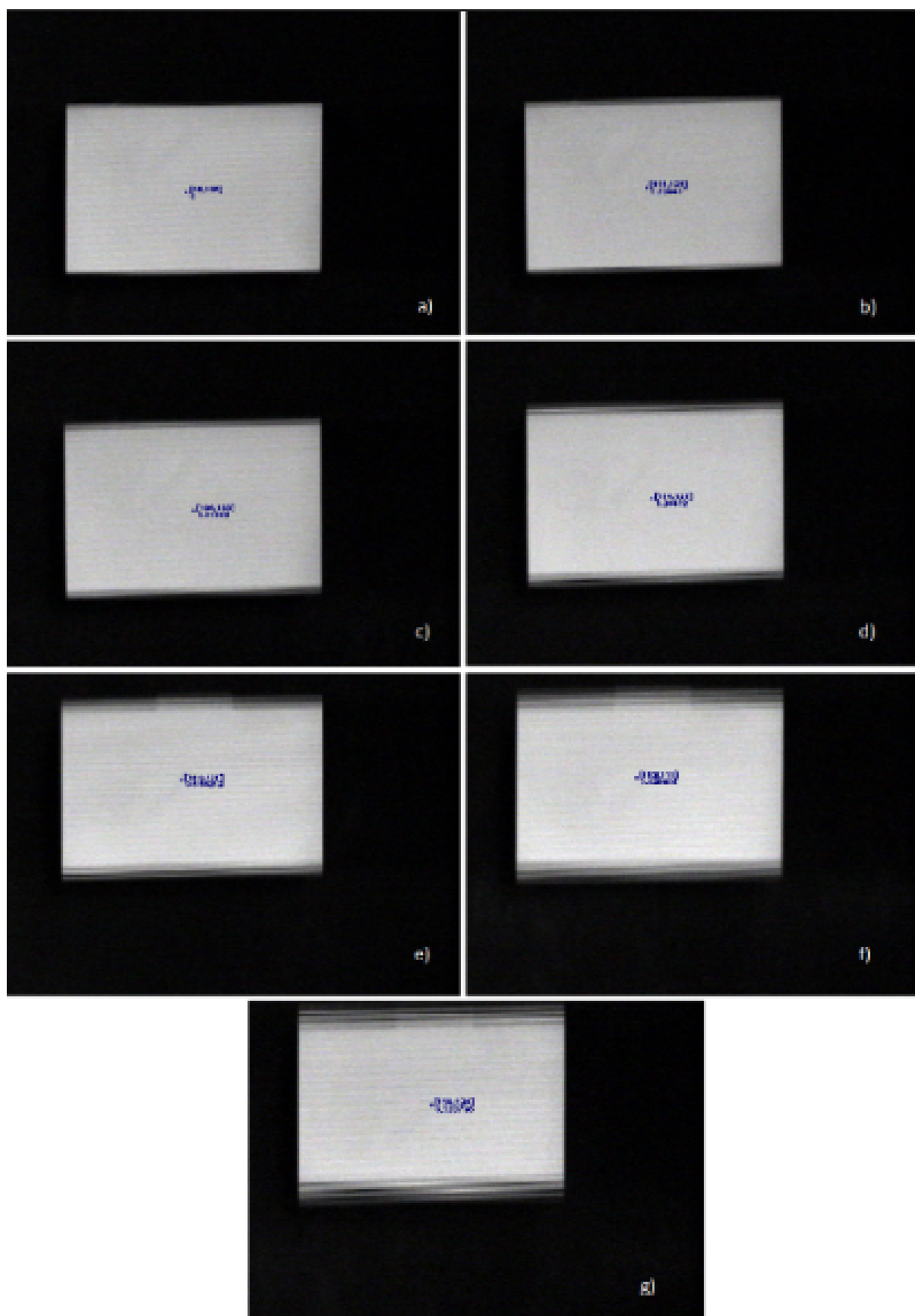
Trigerovanje kamere se vrši softverski kada računar preko Modbus-a dobije informaciju da je senzor detektovao objekat. Brzina od 100% označava nominalnu brzinu trake. Primjetno je da postoji određeno kašnjenje između detekcije senzora i okidanja kamere, koji se javlja zato što je potrebno određeno vrijeme da informacija o detekciji objekta dođe do računara.

Put koji objekat pređe za vrijeme ovog kašnjenja označen je sa Δx (6.4). Brzina trake od 33% je dosta mala, te je Δx pri njoj zanemarivo. Zbog toga će se u daljim razmatranjima slika uslikana pri ovoj brzini smatrati referentnom. Δx se povećava povećanjem brzine. Pomoću očitavanja sa enkodera može se zaključiti da je brzina od 4% reda 0.5 m/s. Uzimajući u obzir da se brzina trake mijenja približno linearno sa promjenom napona napajanja možemo (približno) izračunati vrijeme koje protekne između detekcije senzora i okidanja kamere. Zbog uzimanja približnih vrijednosti, nije moguće egzaktno utvrditi potrebno vrijeme. Ono je reda nekoliko ms. Na slici 6.5 prikazan je isti objekat postavljen na približno istu poziciju slikan pri različitim brzinama trake.



Slika 6.4: a) Referentna (stvarna) vrijednost; b) Traka pređe put Δx

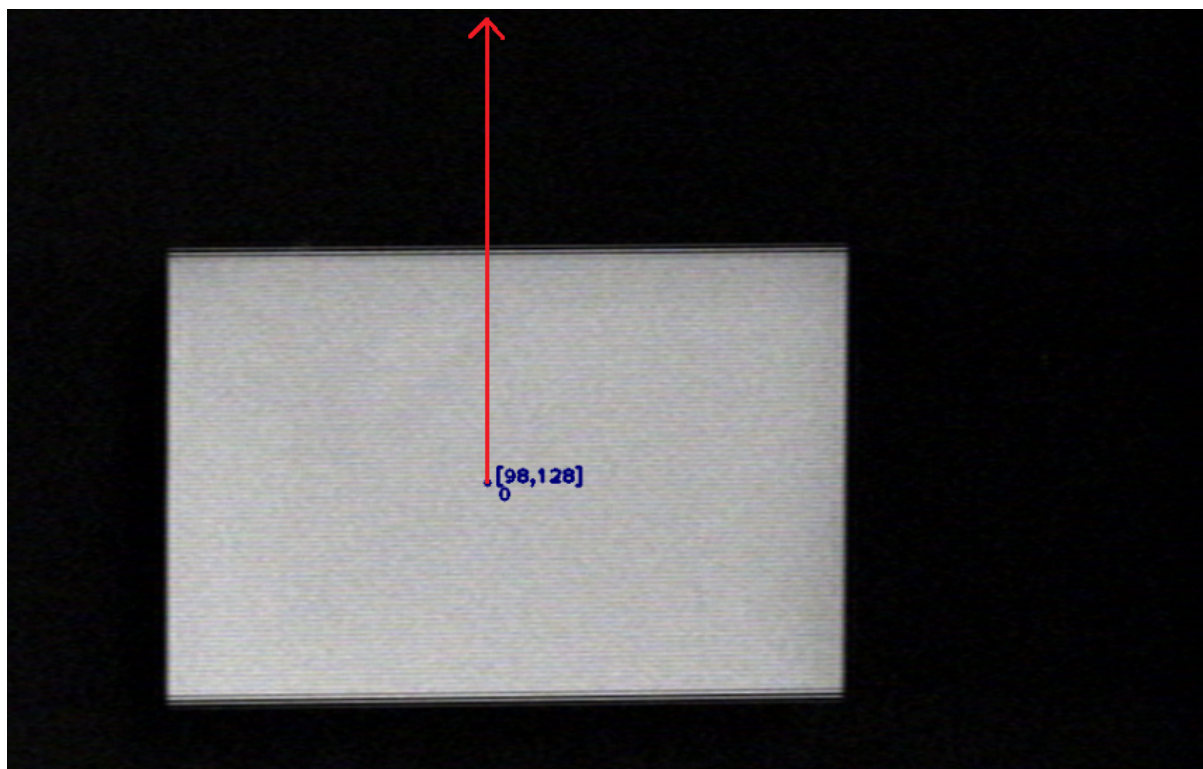
Vrijeme kašnjenja je isto za sve brzine trake, međutim Δx se mijenja sa promjenom brzine trake. Također se može primjetiti da se pri većim brzinama trake javlja efekat zamagljenja ivica eng. *blurring*. Zbog ovoga može doći do nepravilnog određivanja konture objekta, a samim tim i pozicije.



Slika 6.5: Isti objekat pri različitim brzinama trake: a) pri 33%; b) pri 40%; c) pri 50%; d) pri 60%; e) pri 66% ; f) pri 83%; g) pri 100%

6.3 Estimacija trajektorije u budućem vremenu

Ukoliko je poznata pozicija objekta moguće je estimirati njegovu trajektoriju u budućem vremenu. Trajektorija u budućem vremenu predstavlja polupravu sa početkom u određenom centroidu, koja je paralelna x -osom stvarnog koordinatnog sistema i koja se teoretski nastavlja do beskonačnosti (u stvarnosti je to do kraja trake) u smjeru kretanja trake. Na slici 6.6 prikazan se objekat i njegova trajektorija u budućem vremenu. Ukoliko je potrebno naći poziciju u određe-



Slika 6.6: Crvenom bojom naznačena je trajektorija u budućem vremenu za dati objekat

noj tački potrebno je poznavati još jedan parametar: vrijeme koje je potrebno da objekat dođe iz početne u željenu poziciju ili udaljenost željene tačke od početne. Tada se na jednostavan način može odrediti pozicija u željenoj tački. Tabela 6.3 prikazuje estimiranu poziciju za objekte na slici 6.3 kod drugog senzora detekcije. Treba napomenuti da su pozicije određene nakon što cijeli objekat "prođe" senzor budući da se okidanje kamere, a samim tim i određivanje pozicije dešava na opadajuću ivicu.

Broj mjerenja	Pozicija kod prvog senzora (poznata)	Pozicija kod drugog senzora (estimirana)
10	(133, 150)	(695, 150)
12	(137, 86)	(699, 86)
15	(132, 188)	(694, 188)
18	(108, 150)	(670, 150)

Tabela 6.3: Trenutna i estimirana pozicija objekta

6.4 Problemi određivanja pozicije

Odstupanja u nalaženju pozicije uglavnom su rezultat nesavršenog određivanja konture. Problemi pri određivanju konture su razni: sjena objekta, refleksija, osvjetljenje i dr. Također do pogrešnog određivanja dolazi ukoliko objekat nije u cjelosti na slici. U ovoj aplikaciji senzor okida na opadajuću ivicu, odnosno kamera se trigeruje nakon što objekat prođe senzor, te su ovakvi problemi u većini slučajeva izbjegnuti. Do problema će doći ukoliko je objekat preveliki da stane u frejm (6.7) ili ukoliko se traka kreće velikim brzinama.



Slika 6.7: Problem se javlja ukoliko je objekat preveliki

Zaključak

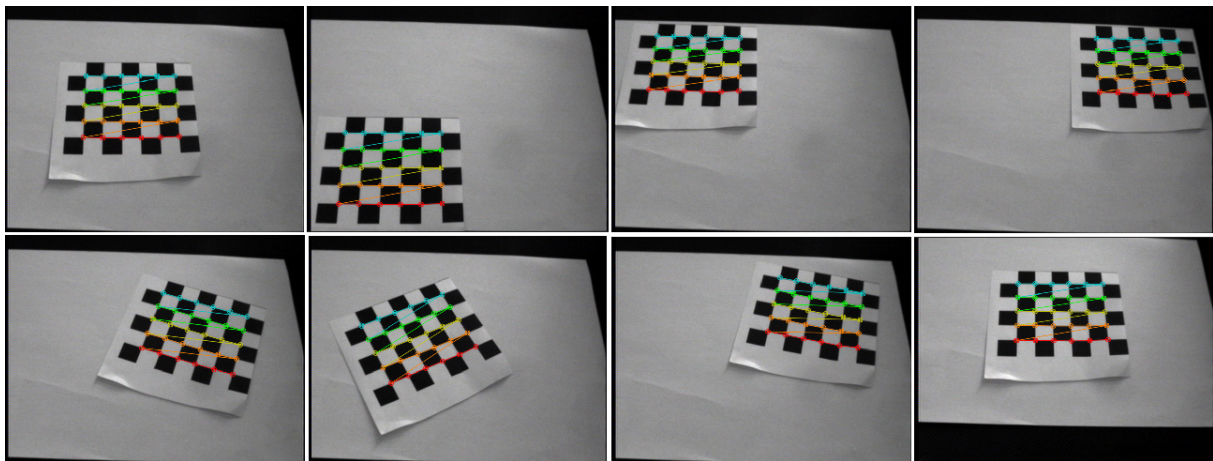
Na osnovu eksperimentalnih rezultata prikazanih u prethodnom poglavlju može se zaključiti da razvijena aplikacija određuje poziciju i orijentaciju sa relativnom greškom mjerenja koja ne prelazi 5%, što je izuzetno dobro budući da se mjerenje vrši u milimetrima. Pogrešno određivanje pozicije i orijentacije javlja se ukoliko je objekat reflektirajući, ukoliko je pored objekta na slici vidljiva njegova sjena te ukoliko je objekat slikan kretajući se velikim brzinama. Pomenuta greška bi se mogla umanjiti korištenjem kamere veće rezolucije, korištenjem preciznijih metoda određivanja pozicije, podešavanjem osvjetljenosti objekta. Greške u određivanju pozicije koje se javljaju u sistemu su zanemariva pri malim brzinama trake. Za dalji razvoj, potrebno je sa sistemom povezati manipulator koji se nalazi iznad drugog senzora, tako da se on može postaviti na estimiranu vrijednost pozicije.

Prilozi

Prilog A

Kod za kalibraciju kamere

Za kalibraciju kamere korišten je uzorak šahovske ploče (*eng. chessboard pattern*). Uzorak je slikan kamerom u različitim pozicijama kao što je prikazano na slici A.1.



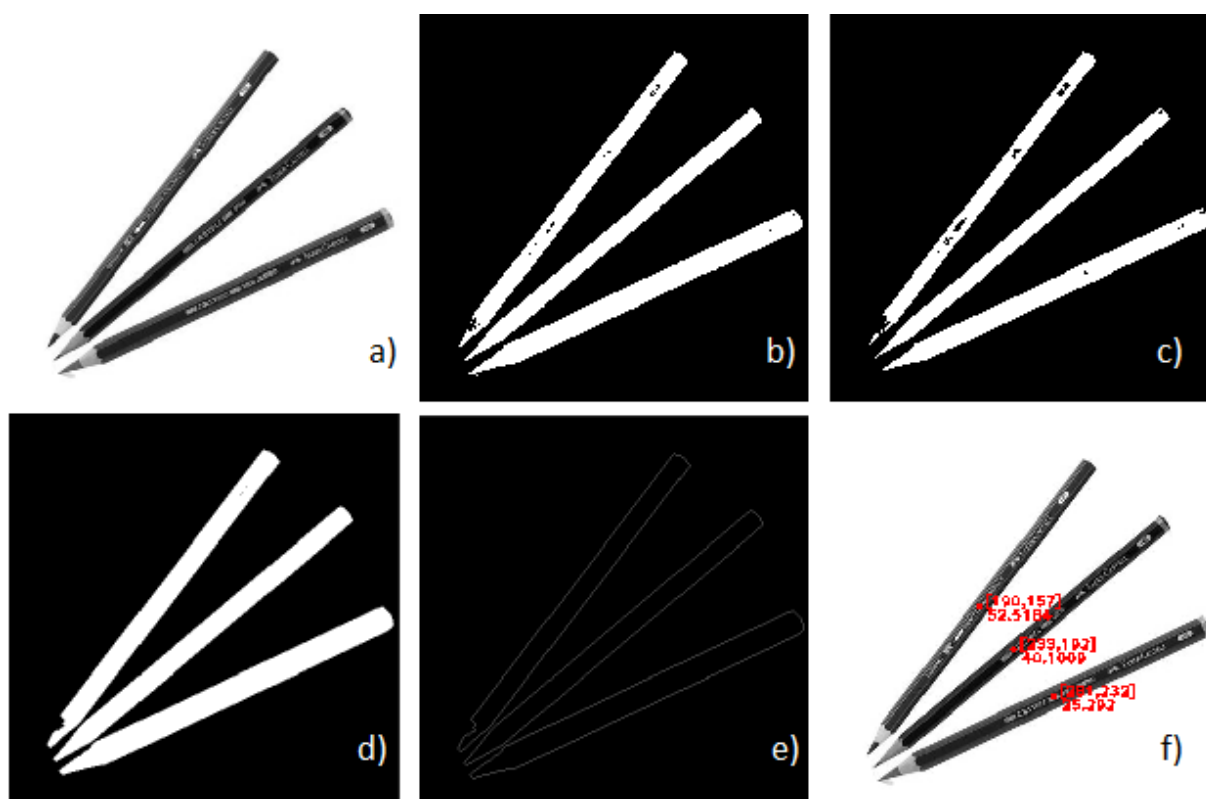
Slika A.1: Određivanje unutrašnjih kornera uzorka u OpenCV-u

Prilog B

Određivanje pozicije i orijentacije kroz *OpenCV*

Nakon što kamera uslika sliku vrši se segmentacija pragom, čiji je rezultat binarna slika. Na nju je primjenjena operacija erozije, te dilatacija četiri puta kako bi se "popunile" šupljine na objektu. Sa *cv::findContours* sa slike se izdvajaju konture. One koje su previše male površine ("mala površina" je definirana u kodu) se zanemaruju.

Pozicija je određena sabiranjem koordinata po x i y , te dijeljenjem dobijenih suma sa veličinom ivice konture. Orijentacija je određena pomoću najmanjeg mogućeg pravougaonika koji se može postaviti oko objekta. Primjer određivanja pozicije i orijentacije prikazan je na slici B.1.



Slika B.1: a) Originalna slika; b) Segmentacija pragom; c) Erozijska; d) Dilatacijska; e) Konture; f) Određene su pozicija i orijentacija

Ukoliko je objekat pravilnog oblika (krug, pravilni mnogougao), orijentacija se neće moći odrediti jer će najmanji pravougaonik koji obuhvata čitav objekat biti kvadrat (ili približno kvadrat s obzirom da se dešava izobličenje objekta tokom preprocesiranja). U ovom slučaju, na upit o orijentaciji ovakvog objekta, aplikacija će dati odgovor da on nema orijentacije, kao što je prikazano na slici B.2.



Slika B.2: Novčići nemaju orijentaciju

Bitno je napomenuti da ovaj kod određuje poziciju u koordinatnog sistemu slike. Pozicija je piksel i određena je njegovim koordinatama po x i y osi, gdje je $(0,0)$ gornji lijevi čošak. Ori-jentacija se računa u stepenima i njena vrijednost je u opsegu $[-90^\circ, 90^\circ]$.

Lista publikacija

Literatura

- [1] Fontoura, L. D., Cezar, R. M., Shape Classification and Analysis. CRC Press, 2009.
- [2] Sony, Sony XC-55 datasheet, (02. marta 2018.), dostupno na: <http://www.alldatasheet.com>
- [3] Prince, J. D., Computer Vision: Models, Learning, and Inference. Cambridge, 2012.
- [4] Hartley, R., Zisserman, A., Multiple View Geometry in Computer Vision. Cambridge, 2004.
- [5] Russ, J. C., The Image Processing Handbook. CRC Press, 2011.
- [6] Electric, S., Modicon M241 Logic Controller, (22. septembra 2018.), dostupno na: http://www.eschneider.pl/download/04%20Automatyka%20przemyslowa/PLC%20sterowniki/M241_PLC_Modicon-Hardware_Guide_2014ENG.pdf
- [7] Electronics, L., ET 318B Energetic reflection light scanner, (23. septembra 2018.).
- [8] Electronic, T., Incremental Rotary Encoders Overview, (23. septembra 2018.), dostupno na: <http://trelectronic.com/sites/default/files/brochures/Brochure-PDF/Incremental%20Encoders.pdf>
- [9] Peterson, C., How it Works: The Charged-Coupled Device, or CCD. Journal of Young Investigators, 2001.
- [10] Curtin, D. P., The Textbook of Digital Photography, 2007.
- [11] Bazargani, H., Laganière, R., “Camera calibration and pose estimation from planes”, IEEE Instrumentation & Measurement Magazine, Vol. 18, No. 6, 2015, str. 20-27.
- [12] The OpenCV Reference Manual, Release 3.0.0-dev, (22. septembra 2018.), dostupno na: <https://docs.opencv.org/3.0-beta/opencv2refman.pdf>
- [13] M. Sonka, V. H., Boyle, R., Image processing, Analysis, and Machine Vision. Thompson, 2008.
- [14] R. C. Gonzalez, S. L. E., R. E. Woods, Digital Image Processing Using MATLAB. Pearson Education, 2004.
- [15] Jähne, B., Digital Image Processing. Springer, 2005.
- [16] C. Demant, B. S.-A., Waszkewitz, P., Industrial Image Processing: Visual Quality Control in Manufacturing. Springer, 1999.
- [17] MODICON, I. A. S., Inc., Modicon Modbus Protocol Reference Guide, (22. septembra 2018.), dostupno na: http://modbus.org/docs/PI_MBUS_300.pdf