

**SISTEM ZA SKUPLJANJE I POSTAVLJANJE PRSTENJA SA MAŠINSKIM
PREPOZNAVANJEM****RING PICK-AND-PLACE WITH MACHINE VISION**Stefan Mirilović, Milan Vidaković, *Fakultet tehničkih nauka, Novi Sad***Oblast – RAČUNARSTVO I AUTOMATIKA**

Kratak sadržaj – Zadatak rada predstavlja sistem za skupljanje i postavljanje (*Pick-and-Place*) prstenja sa mašinskim prepoznavanjem. Softverski deo predstavljaju jedna WPF aplikacija, *Halcon* i *Codesys*. Hardverski deo predstavlja *Fanuc* robot *CRX-10iA/L* sa specijalno napravljenom hvataljkom i kamerom. WPF aplikacija omogućava korisniku da pušta, pauzira i stopira robota, među ostalim funkcijama.

Ključne reči: WPF, C#, *Codesys*, *Fanuc Robot*, *Halcon*, Mašinsko prepoznavanje

Abstract – This paper deals with a system for pick-and-placing rings using machine vision. Software portion of the system is a WPF application, *Halcon* and *Codesys*. Hardware portion is a *Fanuc* robot *CRX-10iA/L* with a custom-built gripper and a camera. WPF application allows the user to start, pause and stop the robot, among other functions.

Keywords: WPF, C#, *Codesys*, *Fanuc Robot*, *Halcon*, *Machine vision*

1. UVOD

Zadatak rada predstavlja razvoj sistema za *pick-and-place* [1] prstenja koji koristi mašinsko prepoznavanje (eng. *Machine vision*) [2]. Ovakvi slični sistemi koriste se u raznim industrijama, od keksova do nakita, da zamene ljudsku radnu snagu i time, dugoročno gledano, smanjuju troškove i čine proizvodnju mnogo produktivnijom.

Za razvoj sistema se morala koristiti kombinacija raznih tehnologija i značajan deo rada je na njihovoj međusobnoj komunikaciji.

2. OPIS HARDVERSKIH TEHNOLOGIJA

Hardverski deo čine robot korporacije *FANUC* [3], model *CRX-10iA/L*, kamera *Basler acA1920-25gc* i specijalno napravljena hvataljka sa LED osvetljenjem.

2.1. FANUC Robot CRX-10iA/L

FANUC je japanska korporacija koja nudi proizvode i servise u polju automatizacije i najveći je proizvođač industrijskih robota na svetu.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Vidaković, red. prof.

Model *CRX-10iA/L* je jedan od tzv. kolaborativnih robota (eng. Collaborative robot ili Cobot) [4]. Kolaborativni roboti su predviđeni za bliski rad sa ljudima i zbog toga su u odnosu na uobičajene industrijske robote napravljeni od lakših materijala, imaju glađe ivice, dodatne bezbednosne funkcije, i ograničenja na brzini i sili da bi osigurali bezbednost u radu.

Robot ima širok izbor različitih tipova ulaza i izlaza, od kojih su za ovaj rad najznačajniji digitalni ulazi i izlazi. Robot takođe podržava *Ethernet/IP* [5] konekcije, što će biti ključno za komunikaciju sa softverom.

2.2. Basler acA1920-25gc kamera

Kamera koja se koristi za mašinsko prepoznavanje je *Basler acA1920-25gc*, rezolucije 1920px x 1080px, 2MP, i može da snima 25 fps (frejmova u sekondi).

Kamera koristi *GigE Vision* [6] interfejs, što je standard uveden 2006. godine za industrijske kamere visokih performansi. On nudi okvir za prenos video zapisa velike brzine preko *Ethernet* mreže.

2.3. Hvataljka

Na robotu je nameštena specijalno napravljena hvataljka za prstenje. Hvataljka je povezana za dva ventila, ventili za regulator pritiska, a regulator za kompresor za vazduh.

Regulator ograničava maksimalni vazdušni pritisak na zadatu vrednost, koja je nameštena na 0.1 MPa ili 1 bar. Dva ventila kontrolišu otvaranje i zatvaranje hvataljke, zaviseći od toga koji je od njih otvoren.

2.4. Podloga

Za najbolje rezultate mašinskog prepoznavanja mora se odabrati pogodna podloga za objekat koji se prepoznaje. Za prstenje je iznenađujuće najbolji izbor najobičniji blok papir za crtanje. Blok papir pati od bljeska i ne daje homogenu podlogu na slici, ali veliki kontrast bele pozadine i tamnih prstenja je prevladao i davao ubedljivo najbolje rezultate.

3. OPIS SOFTVERSKIH TEHNOLOGIJA

Za softverski deo rada su korišćeni *CodeSys* [7] i WPF (*Windows Presentation Foundation*) [8] aplikacija. Za mašinsko prepoznavanje je korišćen *Halcon* [9].

Za komunikaciju sa robotom je korišćen *Ethernet/IP* protokol.

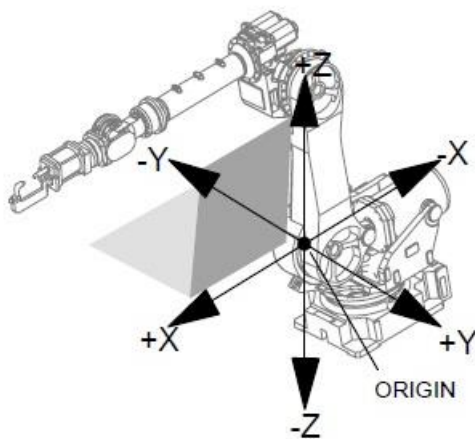
3.1. Robotski softver

Robot ima dva suštinski različita koordinatna sistema po kojim može da se pomera.

Jedan koordinatni sistem je Kartezijev (Dekartov) koordinatni sistem od šest osa:

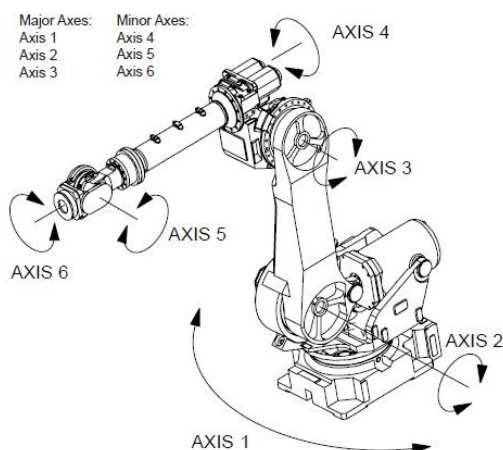
- x – dužina,
- y – širina,
- z – visina,
- w – ugao rotacije oko x-ose,
- p – ugao rotacije oko y-ose i
- r – ugao rotacije oko z-ose.

Slikovno objašnjenje kartezijevog koordinatnog sistema na robotu može se videti na Slici 1.



Slika 1. Kartezijev koordinatni sistem

Drugi koordinatni sistem se zove zglobni (eng. Joint). Kao što naziv sugerise, ose ovog koordinatnog sistema su zglobovi robota i pošto model CRX-10iA/L ima šest zglobova, tako i zglobni koordinatni sistem ima šest osa: J1, J2, J3, J4, J5 i J6. Slikovno objašnjenje može se videti na Slici 2.



Slika 2. Zglobni koordinatni sistem

3.2. Codesys

Codesys je razvojno okruženje za programiranje kontrolerskih aplikacija po međunarodnom industrijskom standardu IEC 61131-3 [10]. Ove aplikacije rade na tzv. PLC-ovima (eng. *Programmable logic controller*) [11].

PLC je industrijski kompjuter koji je robustan i prilagođen za kontrolu proizvodnih procesa, kao što su montažne trake, mašine i roboti.

Codesys ima mogućnost da pretvori redovan računar u PLC sa njegovim *Codesys Control Win V3*. Ovaj alat je takođe Codesys-ov sistem izvršavanja ili rantajm sistem (eng. *Runtime system*) [12]. Rantajm sistem je neophodan za pokretanje koda napisanog u Codesys-u, jer Codesys ne generiše izvršne programe, kao što je .exe fajl, već zavisi od rantajm sistema da pokrene rantajm okruženje (eng. *Runtime environment*) u kojem se onda izvršava kod.

Po IEC 61131-3 standardu, Codesys nudi pet programskih jezika u svom razvojnom okruženju:

- IL (eng. *instruction list*) je programski jezik sličan asemblerskom, zastareo,
- ST (eng. *structured text*) je slično programiranju u C-u ili *Pascal-u*,
- LD (eng. *ladder diagram*) je dijagram koji omogućava programeru da virtualno spoji relejne kontakte i zavojnice,
- FBD (eng. *function block diagram*) je dijagram koji omogućava korisniku da brzinski programira boolean i analogne izraze i
- SFC (eng. *sequential function chart*) je dijagram pogodan za programiranje sekvencijalnih procesa i tokova.

3.3. Windows Presentation Foundation

Windows Presentation Foundation (WPF) je besplatan i open-source grafički podsistem za pravljenje desktop aplikacija na Windows-u. U WPF-u se koristi *Extensible Application Markup Language (XAML)* za definisanje korisničkog interfejsa (frontend), i C# za programiranje funkcionalnosti (backend).

3.4. Halcon

MVTec Halcon je obiman softver za mašinsko prepoznavanje sa integrisanim razvojnim okruženjem (*HDevelop*). U smislu funkcionalnosti se može porediti sa *OpenCV* [13] bibliotekom. Verovatno najveća razlika između njih je to što je *OpenCV* besplatan za korišćenje i open-source, dok *Halcon* nije ni jedno ni drugo.

Halcon može da se koristi iz njegovog razvojnog okruženja *HDevelop*, ali može da se poziva njegov API iz C-a, C++-a, *Visual Basic-a* i C#-a.

4. OPIS IMPLEMENTACIJE

Sistem čine WPF aplikacija i Codesys kontrolerska aplikacija, koji međusobno komuniciraju preko deljene memorije.

4.1. Komunikacija sa Codesys-om

Za komunikaciju sa Codesys-om, WPF aplikacija ima dve promenljive: `inputVariables` i `outputVariables`. `InputVariables` je struktura koja sadrži sve ulazne

parametre, tj. parametre koje idu iz pravca Codesys u WPF. OutputVariables je struktura koja sadrži sve izlazne parametre, iz pravca WPF u Codesys.

Komunikacija između WPF aplikacije i Codesys-a je uspostavljena preko deljene memorije, koja se kontinualno čita i piše u beskonačnoj petlji, koja se poziva preko BackgroundWorker klase. BackgroundWorker klasa izvršava operacije na zasebnoj niti, što dozvoljava spomenutoj petlji da se vrti kontinualno i asinhrono bez ometanja glavne niti.

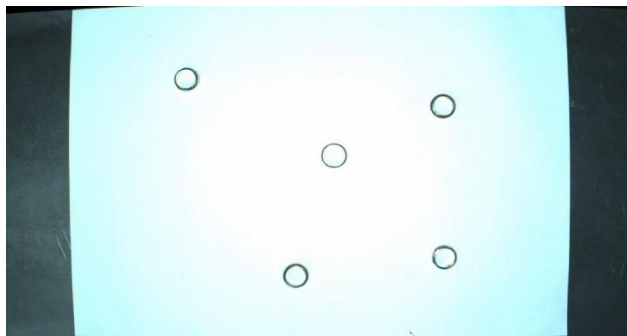
Radi komunikacije se prave dva *Memory-mapped* fajla. Oni su neperzistentni fajlovi koji nisu povezani sa datotekom na disku.

Kada poslednji proces završi rad sa datotekom, podaci se gube. Ovaj vid komunikacije je odabran zbog njegove odlične brzine.

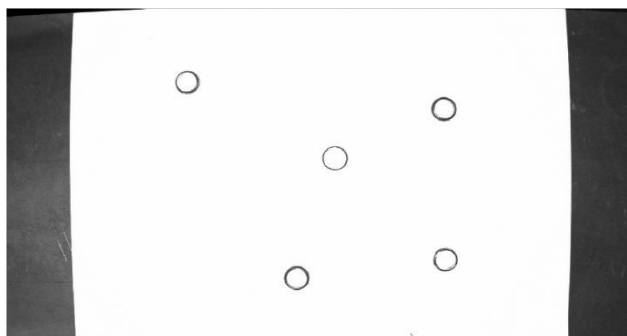
4.2. Prepoznavanje prstenja

Kao što je već rečeno, za prepoznavanje se koristi Halcon-ov API. Sa operacijom Decompose3 trokanalna (RGB, eng. *Red, Green, Blue*) slika koju kamera uslika se pretvori u tri jednokanalne slike, gde svaka odgovara jednoj boji od tri koje zajedno čine RGB. Originalna slika i jedna generisana se vide na Slikama 3. i 4.

Najveći kontrast između pozadine i prstenja davala je slika koja odgovara plavoj boji, pa se ona koristila dalje u programu.



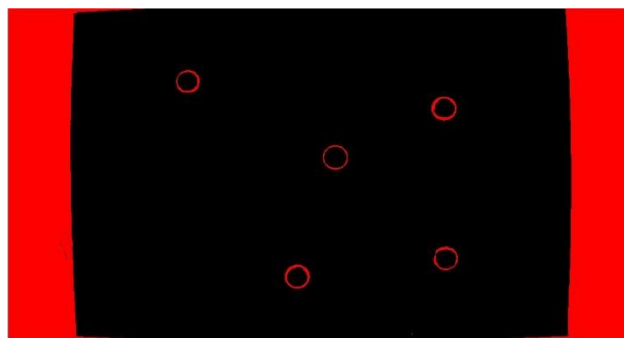
Slika 3. Originalna slika



Slika 4. Izdvojena plava boja iz slike

Sledeća operacija je *Threshold*. Njen zadatak je izdvojiti sve piksele čija je vrednost sive boje između 0 i 150 (0 je crna boja, a 255 bela).

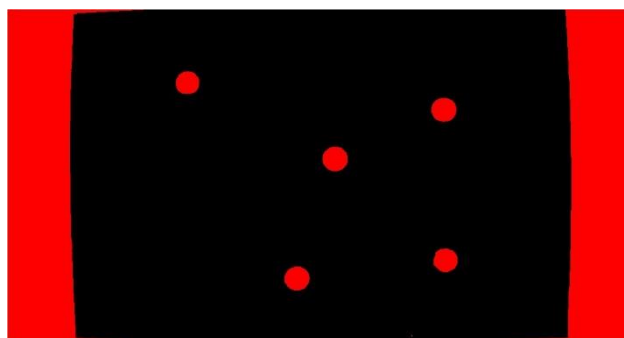
Praktično, ovaj korak je uklonio belu podlogu, a ostatak označio, kao što se vidi na Slici 5. Programski, sve boje slike su se pretvorile u nule i jedinice, gde su nule predstavljene crnom, a jedinice crvenom na Slici 5.



Slika 5. Izdvojene tamne površine

Sledeća operacija *Connection* čini da svaki region koji nije spojen sa drugim bude odvojena komponenta.

Ovo će kasnije omogućiti brojanje komponenti, tj. prstenja. Sledeća operacija *FillUp* popuni sve komponente koji su šuplje. Time se dobije Slika 6.



Slika 6. Popunjene šupljine

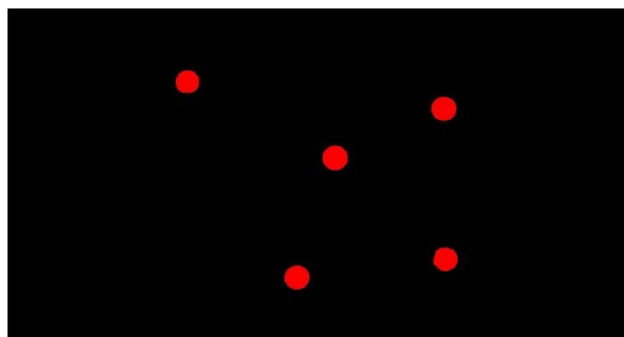
Sledeća operacija *SelectShape* čini više stvari; za svaku komponentu izračuna dve vrednosti:

- *circularity* – procenat koji predstavlja koliko se površina komponente poklapa sa krugom i
- *area* – vrednost površine komponente.

Kad ima ove dve vrednosti za sve komponente, izdvoji sve komponente koji zadovoljavaju sledeće uslove:

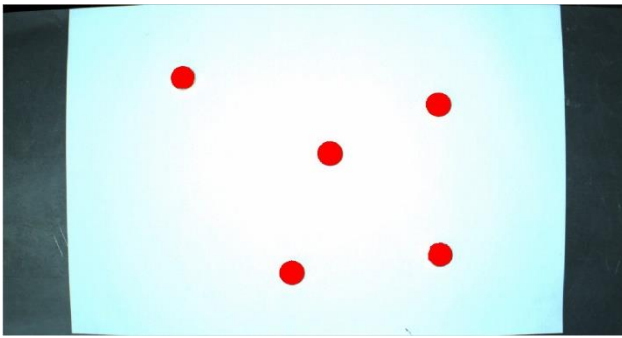
- $0.85 < circularity < 1$ i
- $400 < area < 9999$.

Rezultat ove operacije vidi se na Slici 7.



Slika 7. Rezultat SelectShape operacije

Potom iscrta na prozoru dobijeni rezultat sa Slike 7, što onda izgleda kao na Slici 8.



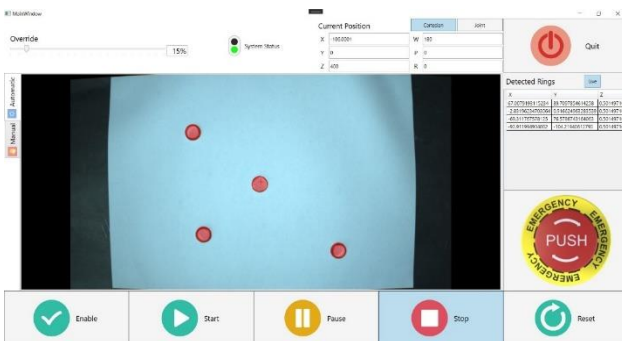
Slika 8. Konačni rezultat

Ovim je trebalo biti prepoznato svo prstenje, gde svaki prsten odgovara jednoj komponenti. Sledeći korak je onda prebrojati komponente sa funkcijom CountObj. Potom se traži row i column svake komponente, što predstavlja njegovu poziciju na slici. U gornjem levom uglu slike je row 0 i column 0, a u donjem desnom je row 1080, a column 1920, što odgovara rezoluciji kamere (1920x1080 piksela). U ovakvom formatu, row i column su robotu neupotrebljivi, zato ih treba pretvoriti u x i y koordinate po robotskom koordinatnom sistemu.

Nad svim koordinatama se onda vrši operacija *ImagePointsToWorldPlane* koja vrši potrebnu konverziju i daje x i y koordinate u milimetrima. Od ovih koordinata se napravi pozicija koja se onda dodaje na listu pozicija koja se šalje Codesys-u.

4.3. Korisnički interfejs

Korisnički interfejs se sastoji od samo jednog prozora u dva režima. Režim menja mnogo elemenata interfejsa, automatski režim može se videti na Slici 9, a ručni na Slici 10.



Slika 9. Korisnički interfejs



Slika 10. Korisnički interfejs u ručnom režimu

5. ZAKLJUČAK

Sistem je razvijen u aktuelnim tehnologijama i sa modelom robota iz najnovije generacije kolaborativnih robota, koji su tek ušli u proizvodnju. Sistem u trenutnom stanju nema neku veliku praktičnu korist, ali daljim razvojem bi se to moglo lako promeniti. Jedan vid daljeg razvoja bi bila zamena statične podloge sa pokretnom trakom. Još jedna ideja za dalji razvoj bi bilo uključivanje i nekakve obrade prstenja nakon skupljanja, poput poliranja.

6. LITERATURA

- [1] Pick-and-place
https://en.wikipedia.org/wiki/Pick-and-place_machine
- [2] Machine vision
https://en.wikipedia.org/wiki/Machine_vision
- [3] FANUC
<https://en.wikipedia.org/wiki/FANUC>
- [4] Collaborative robot
<https://en.wikipedia.org/wiki/Cobot>
- [5] Ethernet/IP
<https://en.wikipedia.org/wiki/EtherNet/IP>
- [6] GigE Vision
https://en.wikipedia.org/wiki/GigE_Vision
- [7] Codesys
<https://en.wikipedia.org/wiki/CODESYS>
- [8] Windows Presentation Foundation
https://en.wikipedia.org/wiki/Windows_Presentation_Foundation
- [9] Halcon <https://www.mvtec.com/products/halcon>
- [10] IEC 61131-3
https://en.wikipedia.org/wiki/IEC_61131-3
- [11] Programmable logic controller
https://en.wikipedia.org/wiki/Programmable_logic_controller
- [12] Runtime system
https://en.wikipedia.org/wiki/Runtime_system
- [13] OpenCV
<https://en.wikipedia.org/wiki/OpenCV>

Kratka biografija:



Stefan Mirilović rođen je 04.05.1997. u Bečeju. Završio osnovnu školu „Zdravko Gložanski“ u Bečeju 2012. godine i Gimnaziju Bečej 2016. godine. Završio osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu, smer Računarstvo i automatika 2020. godine. Nakon završenih osnovnih studija upisao je master akademske studije na istom fakultetu, smer Računarstvo i automatika, modul Elektronsko poslovanje.
kontakt: stefan.mirilovic@gmail.com



Milan Vidaković završio je doktorske studije 2003. godine na Fakultetu tehničkih nauka u Novom Sadu. Na istom fakultetu je 2014. godine izabran za redovnog profesora iz oblasti *Primenjene računarske nauke i informatika*.