

Šolski center Celje

Srednja šola za strojništvo, mehatroniko in medije

UPORABA STROJNEGA VIDA PRI IGRANJU ŠAHA

RAZISKOVALNA NALOGA

Avtor:

Denis FURMAN, M-4. c

Mentor:

Robert Ojsteršek, dipl. inž. mehatronike UN

Mestna občina Celje, Mladi za Celje

Celje, 2015

POVZETEK

Človeški vid je eno izmed najpomembnejših čutil za interakcijo z okoljem. Ljudje smo skozi zgodovino za zgled jemali mater naravo. Ta nas je navdihovala s svojimi rešitvami za nekatere probleme, mi pa smo na podlagi tega iskali načine, kako jih uporabiti v svoj prid. V ta namen sem se odločil, da raziščem in zgradim sistem, pri katerem bi človeški vid uporabili drugače, in sicer bi ga nadomestili z »umetnim očesom.« Strojni vid bi uporabil pri šahu za pozicioniranje figur na igralni plošči.

KAZALO

POVZETEK.....	2
1 UVOD	5
2 HIPOTEZE.....	6
3 ZGODOVINA STROJNEGA VIDA.....	7
4 MEHANSKI DEL.....	8
5.1 OGRODJE.....	8
5.2 PRIJEMALO	9
5 ELEKTRIČNI DEL.....	11
5.1 POGON.....	11
5.2 KRMILJENJE SERVOMOTORJEV	13
5.2.1 PULZNO-ŠIRINSKA MODULACIJA – PŠM.....	13
5.3 NAPAJANJE	15
6 STROJNI VID.....	19
6.1 OSNOVE.....	19
6.2 KAMERA	21
7. MATLAB.....	22
7.1 PRIPRAVA ZA PROCESIRANJE SLIK	24
7.2 ZAJEMANJE SLIK	25
8 ZAKLJUČEK.....	33
9 VIRI.....	34

KAZALO SLIK

SLIKA 1: OGRODJE Z VODILI	8
SLIKA 2: KROGLIČNI LEŽAJ V PRIMERJAVI S KOVANCEM ZA 2 EVRA	9
SLIKA 3: PRIJEMALO.....	10
SLIKA 4: SERVOMOTOR	11
SLIKA 5: RAZSTAVLJEN SERVO.....	12
SLIKA 6: POTENCIOMETER.....	12
SLIKA 7: ARDUINO UNO.....	13
SLIKA 8: PULZNO-ŠIRINSKA MODULACIJA	14
SLIKA 9: SHEMA NAPAJALNIKA	15
SLIKA 10: POTEK POVEZAVE KOMPONENT.....	16
SLIKA 11: KEMIKALIJE, UPORABLJENE PRI POSTOPKU JEDKANJA, OD LEVE PROTI DESNI: Klorovodikova kislina, vodikov peroksid, natrijev hidroksid	17
SLIKA 12: UV-SIJALKA	17
SLIKA 13: PRIMERJAVA GOSTOTE PIKSLOV	19
SLIKA 14: SESTAVA RGB-PIKSLA	20
SLIKA 15: SPLETNA KAMERA LOGITECH C270.....	21
SLIKA 16: OKOLJE MATLAB	22
SLIKA 17: IZBIRA MAPE	23
SLIKA 18: PRIMER UKAZOV V UKAZNEM OKNU IN SHRANJEVANJE SPREMENLJIVK	23
SLIKA 19: IZBIRANJE KAMERE	25
SLIKA 20: IZDELOVANJE GRAFIČNEGA VMESNIKA	26
SLIKA 21: ZAJEMANJE SLIKE	27

1 UVOD

Cilj raziskovalne naloge je zgraditi sistem, ki bo prepoznal postavitev figur in jih prestavljal po določenem načinu. Sistem bo deloval na sledeč način: igralec šaha bo na začetku naredil potezo in spletni kameri izpolnil pogoj za zajemanje slike. Ta bo preko USB-vodila povezana na prenosni računalnik, ki bo s programsko opremo primerjal prejšnjo sliko (začetno) in sedanjo. Razlika slik bo rezultat poteze nasprotnika. Ta podatek bomo preko RS232 UART-komunikacije poslali razvojni plošči Arduino, ki se bo na podlagi razlike slik odločil za nadaljnjo potezo.

Veliko mladih in starejših ustvarjalcev se rado poigra z mikrokontrolerji, nanje priklopijo LCD, elektromotor ali senzor in že imajo izdelek, na katerega so lahko ponosni.

Problem nastopi, ko želijo narediti nekaj kompleksnejšega oz. zahtevnejšega, saj so nekatere komponente precejšnji denarni zalogaj. Dandanes stremimo k temu, da bi dobili kvaliteten material za nizko ceno. Majhne elektronske komponente lahko ugodno naročimo na spletu, največkrat pri kitajskih ponudnikih.

Moja naloga je raziskati načine, kako premikati figure po plošči in pri tem uporabiti ideje, ki bi bile stroškovno najmanj pogojene.

2 HIPOTEZE

Jedro raziskovalne naloge bo strojni vid. Ponudniki kamer za strojni vid poizkušajo za svoj izdelek iztržiti od 100 pa vse tja do 2000 evrov. Ne smemo zanemariti dejstva, da se te kamere uporabljajo za industrijske procese, kjer prihaja do velikih natančnosti in malih pogrškov. Sam sem zato uporabil spletno kamero, ki bo pravšnja za zadano nalogo.

Glede na to sem si postavil naslednje hipoteze:

- Zgraditi sistem za prestavljanje figur.
- Uspešno zaznavanje barv z nizkocenovno spletno kamero.
- Uporabiti čim manj že narejenih polizdelkov.
- Za sistem uporabiti čim manj denarja.
- Robot se sam odloča, kakšna poteza je najboljša.

3 ZGODOVINA STROJNEGA VIDA

Strojni vid je računalniška veja, ki se je v največji meri izoblikovala v zadnjih dvajsetih letih.

V petdesetih letih dvajsetega stoletja so s fotografije uspešno zajemali obliko elementa. V šestdesetih letih so z 2D-slike ekstraktali 3D-geometrični predmet. Zatem so sledila raziskovanja na tem področju, kar je napredek za razvoj umetne inteligence. V sedemdesetih letih so na MIT ustanovili novo smer z imenom Strojni vid. Namen je bil zaznavanje in prepoznavanje preprostih predmetov. V osemdesetih letih so začeli izdelovati sisteme za optično zaznavanje znakov. Uporabljeni so bili v industriji za zaznavanje znakov, števil ali črk. Izdelovati so začeli prve »pametne kamere«, kar bi lahko povečalo uporabo strojnega vida v industriji. V devetdesetih letih se pojavi preko 100 podjetij, ki prodajajo sisteme in LED-osvetlitve. Cene sistemov strojnega vida se počasi začnejo spuščati.

4 MEHANSKI DEL

5.1 OGRODJE

Ogrodje oz. konstrukcija XYZ-table je narejena iz lesa. Na to sem paralelno pritržil dve vodili, ki se sicer uporabljata za predale, kar mi bo omogočalo pomik v X-osi. V kotih sem privijačil U-profile. Na vsaki strani sta izvrtani po dve luknji, v kateri sem vstavil kroglične ležaje. Skoznje pa pride vstavljena gred, na katero so v U-profilih pritrjeni zobniki. Okoli zobnikov pride nameščen zobati jermen.



Slika 1: Ogrodje z vodili



Slika 2: Kroglični ležaj v primerjavi s kovancem za 2 evra

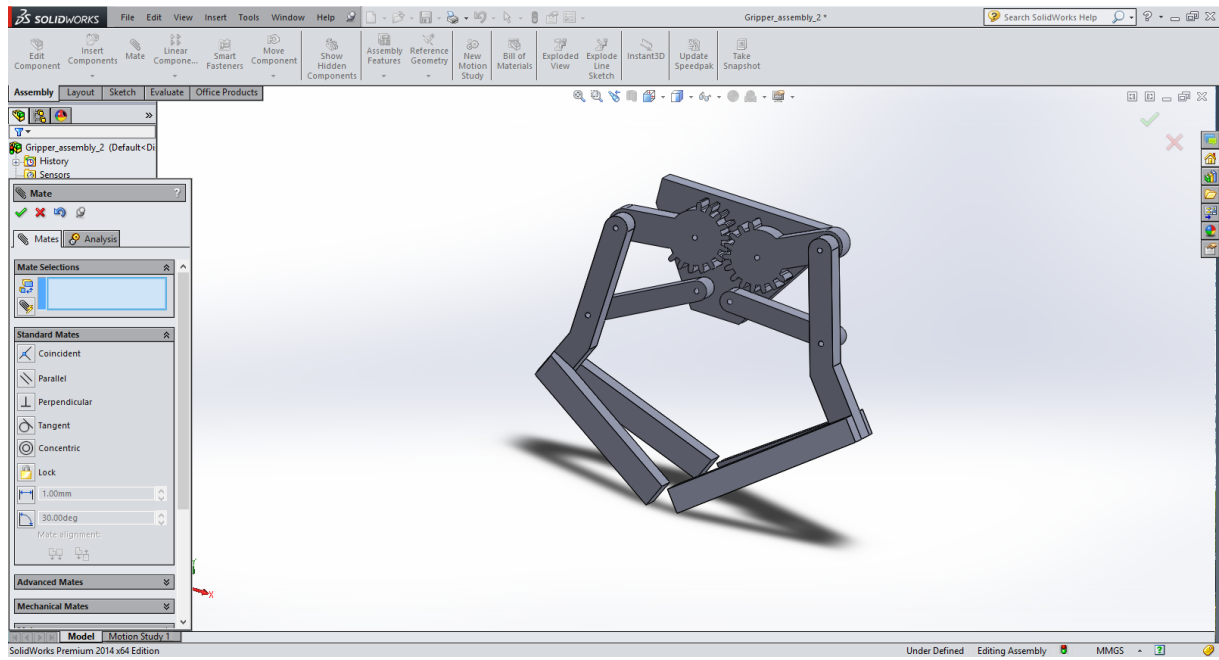
5.2 PRIJEMALO

Prijemalo je poleg ogrodja ključni element za delovanje. Njegova naloga je čvrsto zgrabiti figuro, da jo lahko premaknemo. Vsaka figura pa se razlikuje po velikosti, torej je naloga skonstruirati prijemalni člen, ki bo zagrabil ne glede na obliko figure.

Solidworks spada na področje CAD-tehnologije. Je orodje, s katerim lahko hitro izdelujemo modele, kar je ključnega pomena pri inženirjih, ki snujejo nove izdelke. Preden se lotijo fizične izdelave, zmodelirajo, da dobijo predstavo o videzu in morebitnih tehničnih težavah.

Izdelava modela prijemala je najprej potekala na papirju. Izhajal sem iz velikosti figur, potem sem se moral spopasti s težavo različnih oblik. Nato sem skiciral celoten griper v naravni velikosti, da sem dobil približne mere. Te sem zaokrožil in pričel izdelavo CAD-modela.

Raziskovalna naloga - Uporaba strojnega vida pri igranju šaha



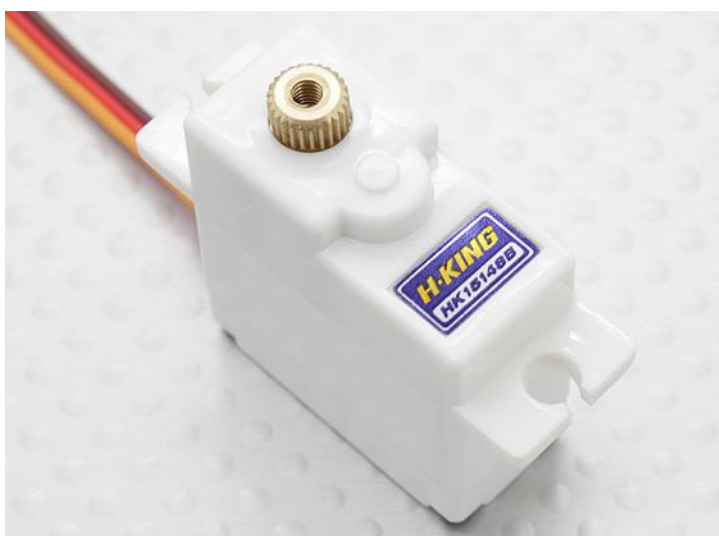
Slika 3: Prijemalo

Najpreprostejši način fizične izdelave modelov je s 3D-tiskalnikom. Trdnost je zadostna za prijemanje figur in prav tako funkcionalnost.

5 ELEKTRIČNI DEL

5.1 POGON

Za pogon sem uporabil servomotorje. Zanje je tipično, da omogočajo 90- ali 180-stopinjski kotni zasuk, kar je premalo za dolžino, ki jo potrebujem. Servomotor bi lahko enostavno modificiral na zmerno vrtenje, vendar bi izgubil možnost njegove natančne pozicije.



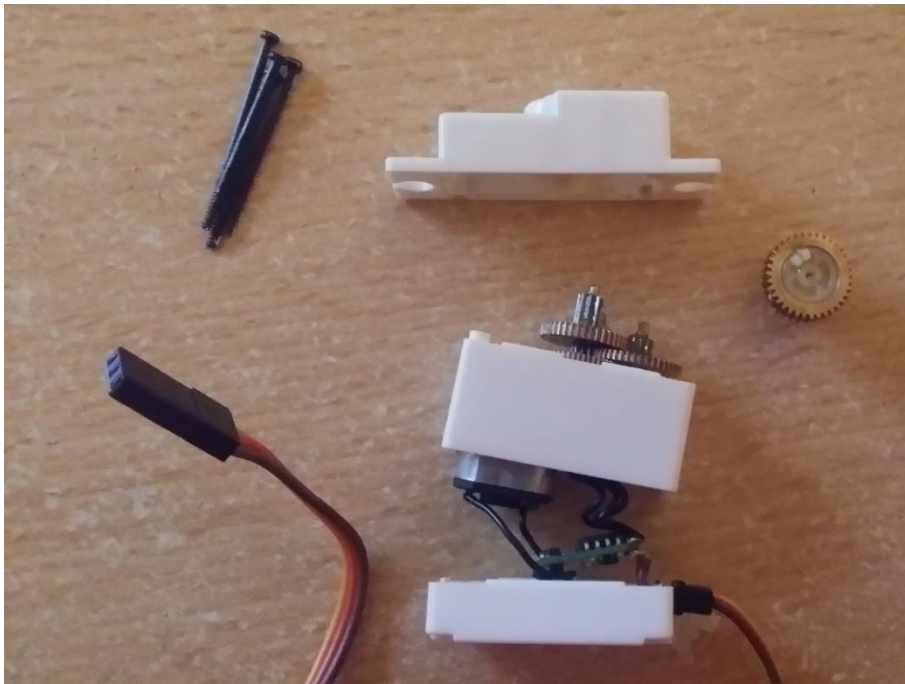
Slika 4: Servomotor

19-gramski digitalni servomotor premore 0.25 Nm navora pri 4.8 V napajalni napetosti ali 0.28 Nm pri 6 V napajalni napetosti. V 0.17 sekunde se pomakne za 60 stopinj, če ga napajamo z 4.8 V ali v 0.14, če ga vzbujamo s 6 V. Servo predstavlja pravšnji pogonski aktuator po zelo ugodni ceni.

Servomotor deluje v zaprto-zančnem sistemu. Informacijo o položaju osi dobi iz potenciometra, ki je priključen na regulacijsko vezje. To pa daje ukaz DC-motorju za premik levo ali desno. Problem v prvi vrsti predstavlja potenciometer, ki je namenjen samo za zasuke, manjše od 360 stopinj. Torej z zamenjavo obstoječega potenciometra s potenciometrom, ki premore 10 zasukov, lahko pridemo do želene rešitve. Paziti moramo, da ima v obeh primerih enako vrednost upornosti in lahko se lotimo zamenjave. Obstoječi potenciometer je s tremi kabli priključen na regulacijsko vezje. Za pravilno nadaljnje delovanje je dobro vedeti, kako

so kabli priključeni na potenciometer. Najlažje je to narediti z multimetrom, pri katerem merimo upornost med priključki, obenem pa vrtimo potenciometer. Enako storimo na zunanjem potenciometru. Sedaj lahko odrežemo in podaljšamo kable do zunanjega potenciometra.

V zobniškem prenosu ima servomotor zatič, ki je viden na spodnji sliki. Ta preprečuje večje zasuke in ga lahko enostavno odstranimo.



Slika 5: Razstavljen servo

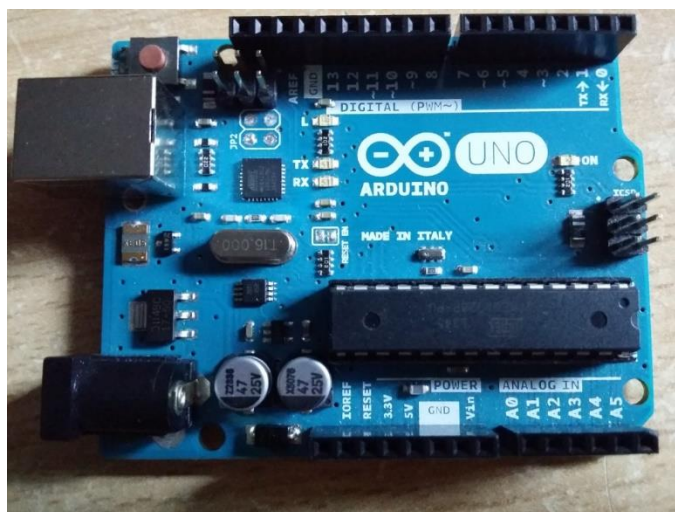


Slika 6: Potenciometer

5.2 KRMILJENJE SERVOMOTORJEV

Krmiljenje servomotorjev bo prevzel Arduino UNO. Gre za odprtokodno vezje, katerega jedro je Atmelov mikrokrmilnik Atmel 328p. Je zelo razširjena in popularna platforma, saj s programsko opremo Arduino IDE enostavno pišemo programe. Kar je najpomembneje, na spletu najdemo veliko literature in že narejenih projektov. V mojem primeru bom krmilil servomotorje, za katere pa je že v sami programski opremi napisana knjižnica. Z ukazom pa jo pokličemo v delovno okolje.

Arduino UNO sprejema napajalno napetost od 5 V do 12 VDC. Nanj lahko shranimo do 32 kB podatkov in deluje na frekvenci 16 MHz. Na plošči lahko uporabimo 6 analognih vhodov in 14 digitalnih vhodov ali izhodov. 6 digitalnih pinov vključuje PŠM, ki je ključna za upravljanje s servomotorji.

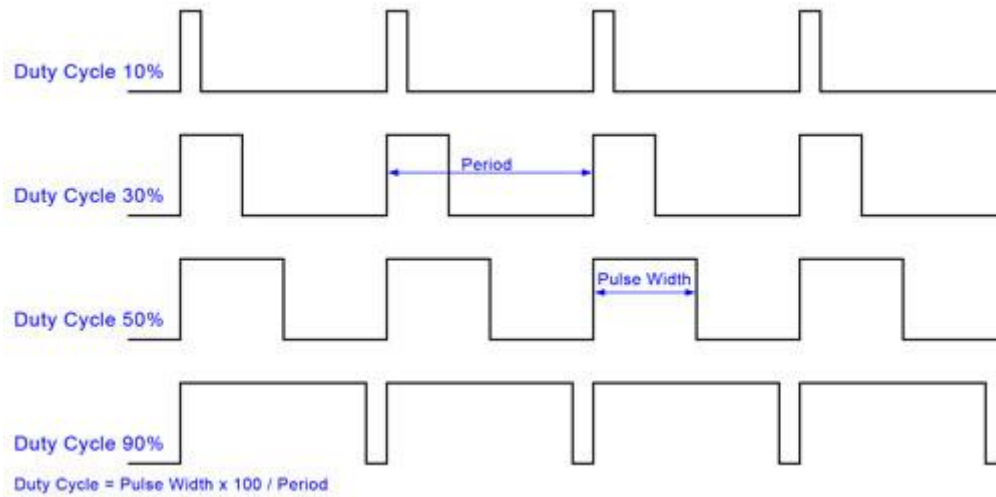


Slika 7: Arduino UNO

5.2.1 PULZNO-ŠIRINSKA MODULACIJA - PŠM

Pulzno-širinska modulacija (PWM) je tehnika nadzora oziroma krmiljenja energije, ki jo pošiljamo el. porabnikom z uporabo on-off digitalnega signala. Delu periode, ko je digitalni signal na on, pravimo duty cycle (čas trajanja periode). Povprečna vrednost enosmernega signala se lahko spreminja s spreminjanjem duty cikla. Duty cikel se lahko spreminja od 0 (signal je na off) do 1 (signal je konstantno na on). Če ima signal vrednost 5 V, kadar je na on, in vrednost 0 V, kadar je na off, potem lahko s spreminjanjem duty cikla dosežemo katero

koli vrednost med 0 V in 5 V. S pomočjo te metode lahko krmilimo npr. hitrost vrtenja DC-motorjev ali svetilnost svetlobnih teles.



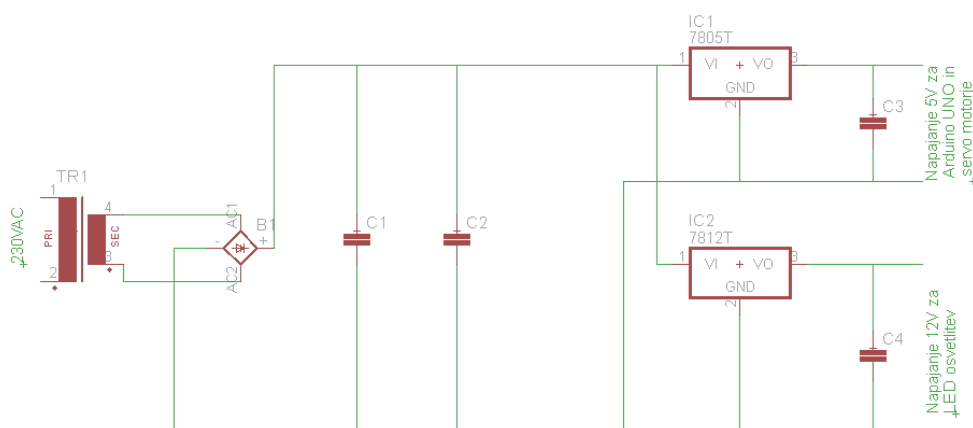
Slika 8: Pulzno-širinska modulacija

5.3 NAPAJanJE

Arduino UNO se napaja kar preko USB-priključka prenosnega računalnika. Vendar pa je ta omejen na 500 mA, kar je za poganjanje treh servomotorjev premalo, poleg tega pa je potrebno priskrbeti 12 V napetosti za LED-trak.

V spletnih trgovinah lahko kupimo cenene adapterje, ker pa je bil moj namen izdelovati iz nič, sem to vezje naredil sam. Prednost takega vezja je v tem, da bo omogočalo napajanje tako servomotorjev, kot tudi LED-traku.

Za izdelovanje shem sem uporabil programsko opremo Eagle. Ponaša se s široko bazo električnih komponent in jo lahko brezplačno snamemo s spleta (<http://www.cadsoftusa.com/>).

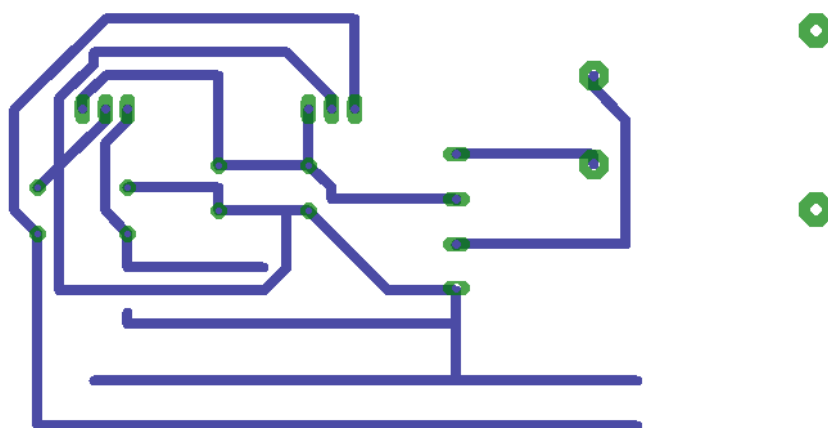


Slika 9: Shema napajalnika

Najpomembnejša komponenta napajalnika je transformator, saj opravi največ dela. Iz omrežne napetosti 230 V izmenične napetosti pretvori v 15 V izmenične napetosti. Naslednji element je usmernik. Poznamo dva tipa usmernikov: polvalnega in polnovalnega. Razlikujeta se po načinu usmerjanja izmenične veličine. Izmenična veličina je sestavljena iz period. Prva polperioda prevaja v pozitivno smer, medtem ko druga prevaja v negativno. Polvalni usmernik torej prepušča samo prvo polperiodo, druge pa ne. Dejansko tako izgubimo polovico moči, zato se v praksi ne uporablja. Uporablja pa se polnovalni usmernik, znan pod imenom Graetzov mostič, ki sem ga uporabil tudi sam. Ta pa drugo, negativno polperiodo

preusmeri v pozitivno. Rezultat so torej pozitivno usmerjene polperiode. Naloga kondenzatorjev C1 in C2 je grobo glajenje napetosti. Zaradi velikih napetostnih nihanj sta lahko kondenzatorja večje kapacitivnosti. Sledita napetostna regulatorja LM7805 in LM 7812. Regulatorja iz družine LM78xx sta namenjena za fino glajenje napetosti. Zadnji dve števili povesta, za kakšno napetost generirata na izhodnih priključkih. Kondenzatorja C3 in C4 pa služita za glajenje napetosti ob morebitnih zunanjih vplivih in sta zelo majhnih vrednosti.

Sedaj je potrebno postaviti komponente na ploščo in jih povezati. Eagle ima možnost samodejnega povezovanja.



Slika 10: Potek povezave komponent

Potek povezav komponent natisnemo na paus papir. Natiskano vezje postavimo na ploščico, prekrito s fotolakom, in osvetlimo z UV-lučjo približno 4 minute in pol na razdalji 50 cm. S tem dosežemo projiciranje vezja na ploščico. Na ploščici je še vedno sloj fotolaka, ki se ga znebimo tako, da v 1 dl vode raztopimo 30 zrn natrijevega hidroksida (NaOH). Sedaj lahko dobro vidimo potek povezav. Sledi postopek jedkanja. Pripravimo mešanico vodikovega peroksida (H_2O_2), klorovodikove kisline (HCl), znana tudi pod imenom solna kislina, in vode. Vodikov peroksid je 30 %, solna kislina pa je 20 %. Razmerje mešanice naj bo 1 : 1 : 1. Nastala tekočina reagira z bakrom, ki se bo počasi začel razgrajevati. Ostale bodo samo sledi, kjer se mešanica bakra ni razgradila.

Raziskovalna naloga - Uporaba strojnega vida pri igranju šaha



Slika 11: Kemikalije, uporabljene pri postopku jedkanja, od leve proti desni: klorovodikova kislina, vodikov peroksid, natrijev hidroksid



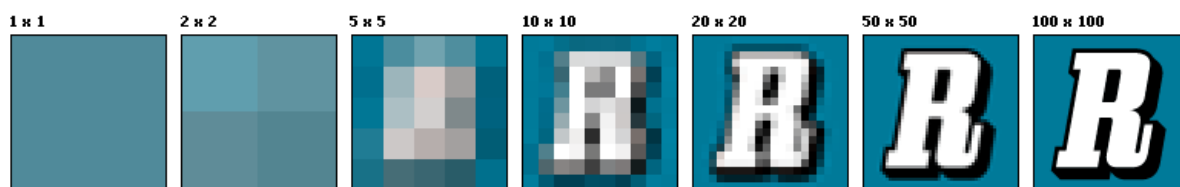
Slika 12: UV-sijalka

Uporabljena je bila živosrebrova luč, moči 150 W. Takšne lahko ponekod srečamo pri javni razsvetljavi. Za delovanje moramo zagotoviti ustrezno predstikalno napravo. V tem primeru za to služi dušilka. To je navitje, ki ob vklopu onemogoča prekomeren tok skozi žarnico, saj bi jo v nasprotnem primeru uničilo. Žarnico sem predelal, in sicer je ob nakupu v zunanjem zaščitnem plašču bel prah (fosfor). Njegov namen je, da UV-svetlobo, ki se proizvaja v notranjem visokotlačnem plašču, filtrira. Tega prahu se moramo znebiti. Prva možnost je, da celoten zunanji plašč stremo in nam ostane samo notranji plašč. Drugi način, ki sem ga izbral sam, je, da v plašč z diamantnim orodjem zvrtno luknjo in vanjo nasujemo nekaj kremenčevega peska. Beli prah se lepo odstrani in na koncu žarnico še operemo od znotraj. **UV-svetloba je zdravju škodljiva, zato se ne zadržujemo v prostoru, kadar takšna sijalka sveti.**

6 STROJNI VID

6.1 OSNOVE

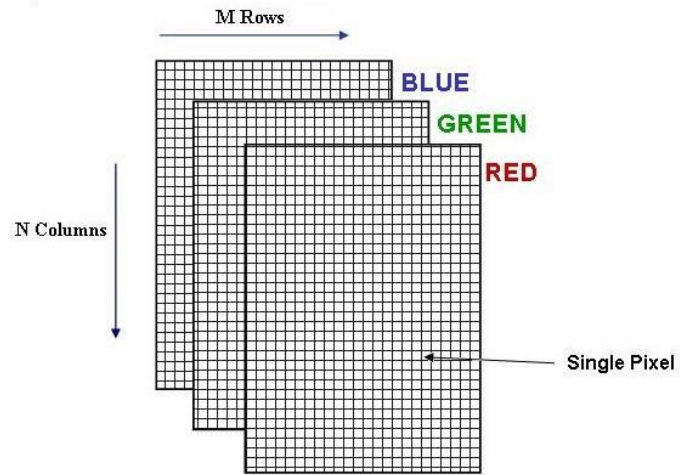
Preden se lotimo procesiranja slik, moramo vedeti, kaj je računalniška slika. Kadar kupujemo fotoaparati ali pametni telefon in je naš namen zajemati fotografije, se vprašamo, kakšno ločljivost ima senzor. Ko govorimo o ločljivosti senzorja, mislimo na število pik, ki jih ima senzor za zajemanje slik. Te pike lahko poimenujemo tudi piksli. Torej »1 M« oz. 1 megapiksel pomeni, da lahko zajemamo slike ločljivosti 1280*720.



Slika 13: Primerjava gostote pikslov

Piksel je torej najmanjši gradnik digitalne slike ali zaslona. Najprej se osredotočimo na črno-belo sliko. Barvo piksela določimo s številom od 0 do 255. Pri tem je 0 črna in 255 bela, števila vmes pa predstavljajo različne odtenke sive. Lahko rečemo, da je slika sestavljena iz 2D-matrice števil.

Barvne slike pa dobimo z mešanjem osnovnih barv. Rdeča, zelena in modra so osnovne barve (angl. RGB red-green-blue). Vsaka osnovna barva ima vrednosti od 0 do 255. Kar pomeni 256 odtenkov rdeče, 256 odtenkov zelene in 256 odtenkov modre. Če ta števila pomnožimo, dobimo vse možne kombinacije barv. Tukaj pa imamo 3 2D-matrice števil za posamezno osnovno barvo, zložene »ena na drugo«.



Slika 14: Sestava RGB-piksla

6.2 KAMERA

Za zajemanje slik sem uporabil spletno kamero Logitech C270. Ima možnost zajemanja slik v HD-ločljivosti, 720 x 480 pikselov, vendar bo zadostovalo 640 x 320. To pomeni, da bo matrica vsebovala 205800 števil.

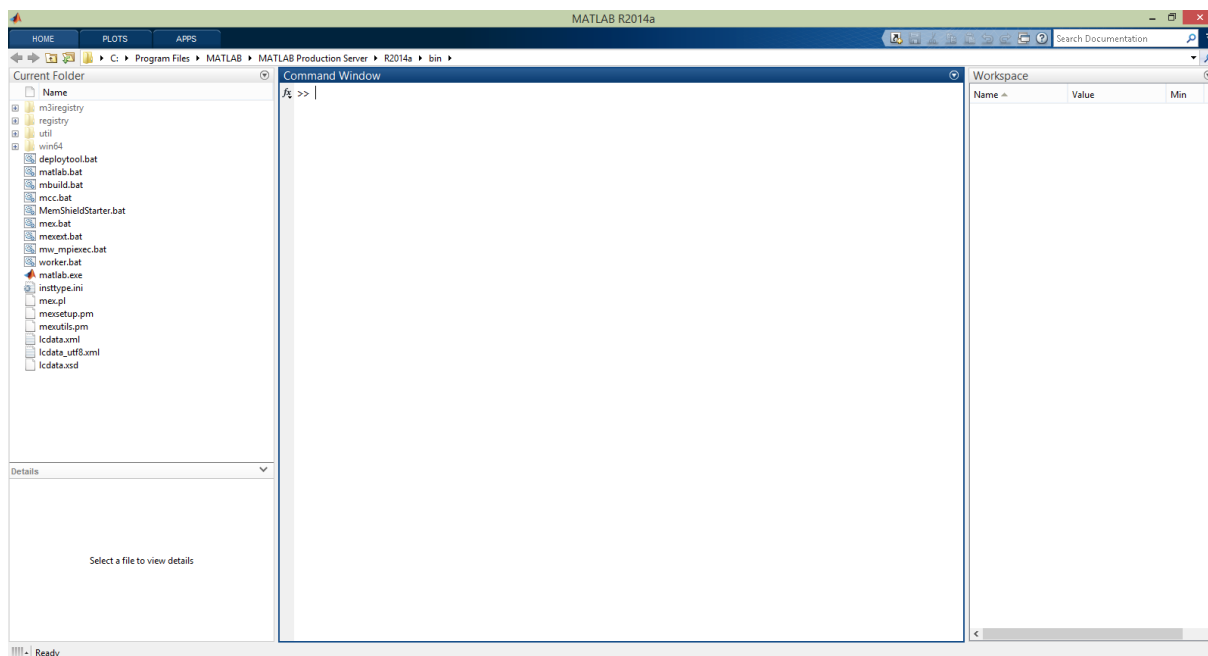


Slika 15: Spletna kamera Logitech C270

7. MATLAB

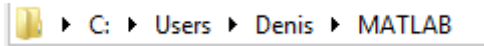
MATLAB (matrix laboratory, kar pomeni laboratorij za matrike) je programski paket za numerično analizo in programski jezik četrte generacije. Razvija ga podjetje MathWorks. Med drugim je primeren za operacije z matrikami, risanje funkcij, implementacijo algoritmov, analizo in razvoj vodenja sistemov, katere omogočimo z dodatkom Simulink, digitalno obdelavo signalov in načrtovanjem filtrov. Omogoča tudi razvoj uporabniških vmesnikov. Programi, pisani v MATLAB-u, pa lahko uporabljajo tudi knjižnice, pisane v C-jeziku, C++, Javi in Fortranu.

Ko zaženemo MATLAB, se pojavi okno v prednastavljeni postavitvi.



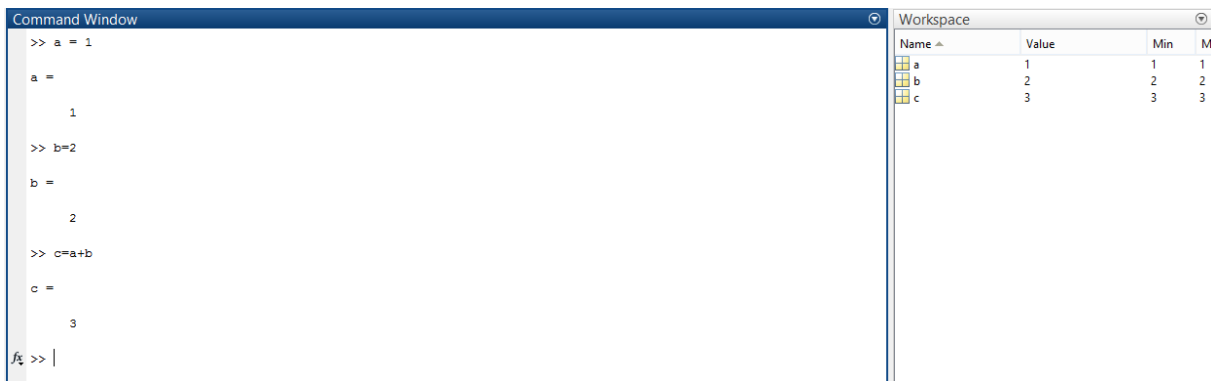
Slika 16: Okolje MATLAB

Da lahko pričnemo delati, si moramo postaviti delovno mesto. To enostavno naredimo z izbiro mape na računalniku. Vanjo se shranjujejo vsi podatki, ki jih obdelujemo. Če hočemo v sistem klicati npr. sliko ali graf, se moramo prepričati, da je ta slika v dani mapi. Na levi imamo možnost sprehajanja skozi mape za lažji pregled.



Slika 17: Izbira mape

Osrednji del nam predstavlja ukazno okno. Tukaj vpisujemo ukaze, ki jih MATLAB izračuna oz. jih procesira. Vpisujemo lahko tudi preproste osnovne matematične račune. Po pritisku na enter dobimo rezultat. Vsako številko lahko zapišemo kot spremenljivko. Ta se nam shrani v ločeno okno desno od ukaznega za boljši pregled.



Slika 18: Primer ukazov v ukaznem oknu in shranjevanje spremenljivk

7.1 PRIPRAVA ZA PROCESIRANJE SLIK

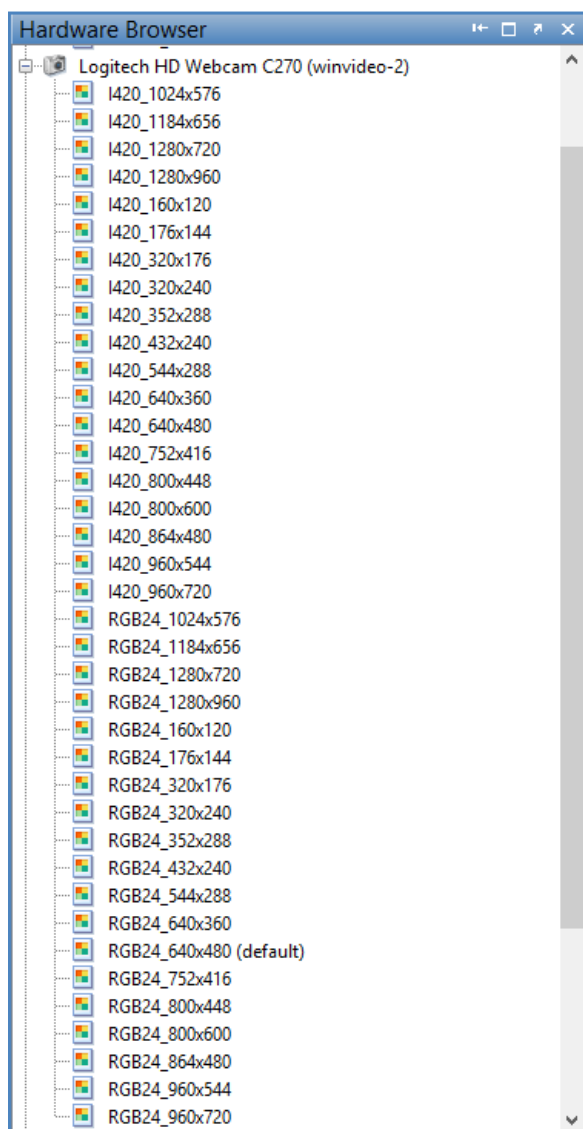
Da lahko pričnemo z obdelavo slik, moramo namestiti orodje za obdelavo slik (angl. Image processing toolbox). To je skupek funkcij, ki povečajo zmožnost MATLAB-ovega procesiranja matrik. Orodje podpira veliko število funkcij za obdelavo matrik.

Fotografije sem zajemal z že prej omenjeno kamero. MATLAB okolje v osnovi ni namenjeno zajemanju slik, zato nanj namestimo orodje, ki nam to omogoča (angl. Image acquisition toolbox). Sledi postopek konfiguracije kamere.

V ukazno okno napišemo »fx>>imaqwinfo« za pridobitev informacij o nameščenih adapterjih priključenih kamer. Prikaže se nam odgovor, iz katerega razberemo adapterje. V primeru, da se nam ne pokaže, kamero pa imamo priključeno, pomeni, da MATLAB ne »vidi« strojne opreme. To rešimo tako, da s spleta namestimo adapter. Pozorni moramo biti, da je kompatibilen z operacijskim sistemom na računalniku.

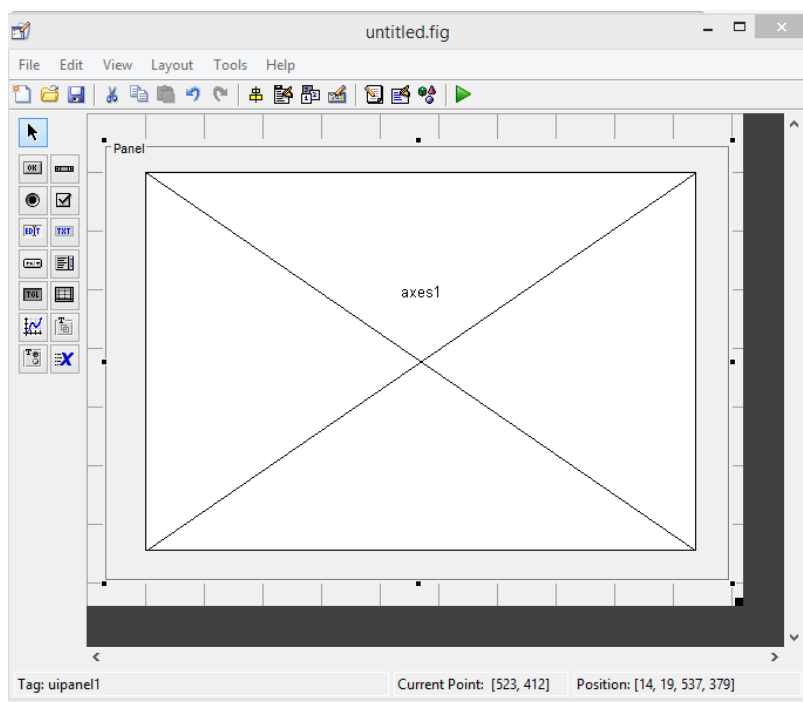
7.2 ZAJEMANJE SLIK

Ukaz »fx>>imaqttool« nam odpre okno za zajemanje slik. Na levi strani je spisek kamer ter ločljivosti, ki jih je kamera pripravljena zajemati. Orodje označi vsako strojno opremo s številko npr. winvideo-2. To si moramo zapomniti poleg resolucije, ki je nastavljena po privzetih nastavitvah (640 x 480).



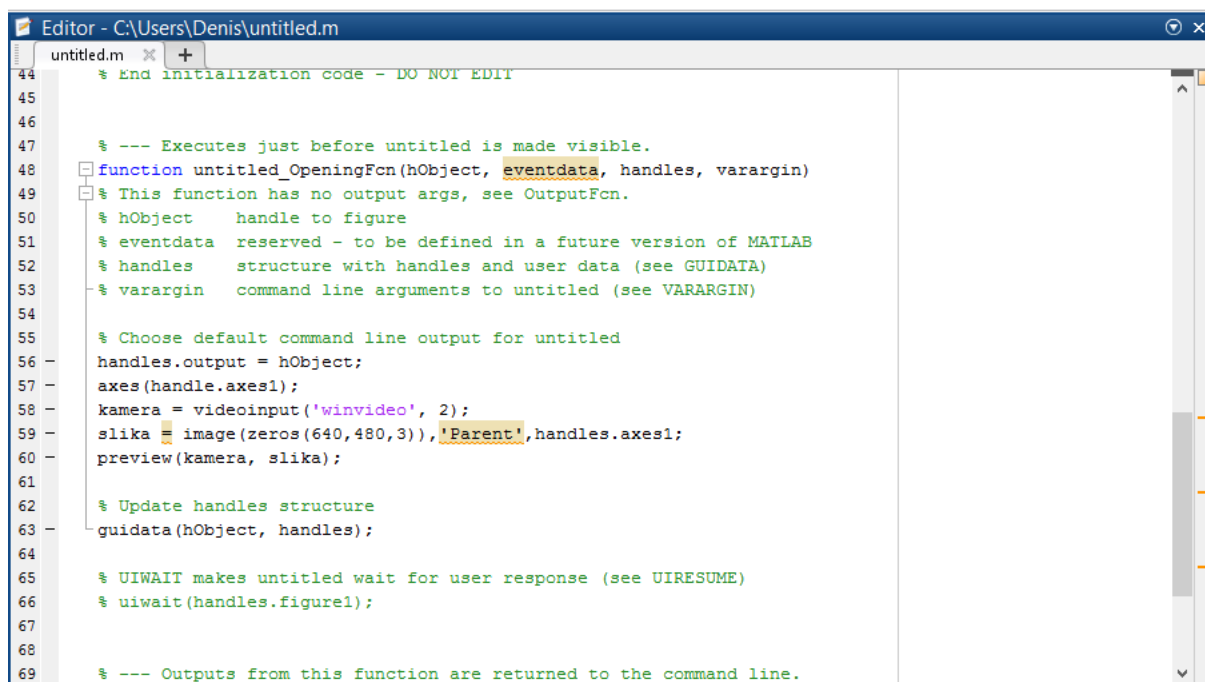
Slika 19: Izbiranje kamere

Kamera je pripravljena na zajemanje slik. Sledi urejevanje praznega prostora za prikaz. Lahko rečemo, da pripravljamo GUI (graphical user interface). To storimo z ukazom »fx>>guide«. Odpre se nam okno in izberemo prazni grafični vmesnik. Izberemo ploščo (panel) in nakažemo prostor, v katerem se bo prikazala slika. Sedaj pa kliknemo na »axes« ter zopet nakažemo območje. V tem prostoru se bo generirala slika, kadar bomo poklicali ukaz za zajemanje slike in se označi z »axes1«, kar predstavlja ime tega prostora.



Slika 20: Izdelovanje grafičnega vmesnika

Grafični vmesnik shranimo in odprla se nam bo MATLAB.m datoteka. Pod »handle.output = hObject« lahko začnemo pisati program za predogled zajemanja. Definirati moramo polje, ki smo ga naredili (axes1), katero kamero uporabljamo in karakteristiko kamere s katero zajemamo.



```

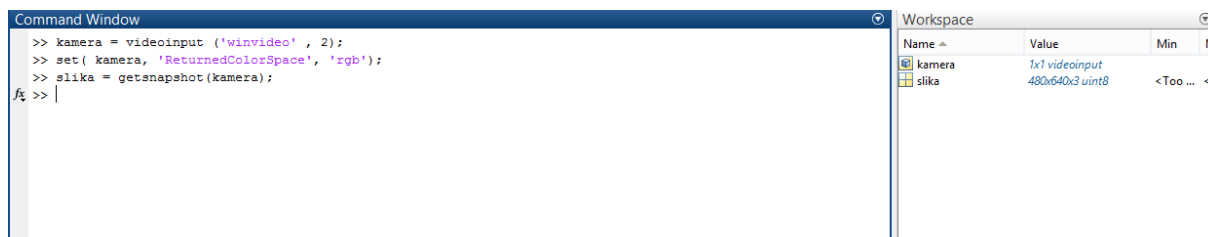
Editor - C:\Users\Denis\untitled.m
untitled.m x +
44 % End initialization code - DO NOT EDIT
45
46
47 % --- Executes just before untitled is made visible.
48 function untitled_OpeningFcn(hObject, eventdata, handles, varargin)
49 % This function has no output args, see OutputFcn.
50 % hObject    handle to figure
51 % eventdata reserved - to be defined in a future version of MATLAB
52 % handles    structure with handles and user data (see GUIDATA)
53 % varargin   command line arguments to untitled (see VARARGIN)
54
55 % Choose default command line output for untitled
56 handles.output = hObject;
57 axes(handles.axes1);
58 kamera = videoinput('winvideo', 2);
59 slika = image(zeros(640,480,3),'Parent',handles.axes1);
60 preview(kamera, slika);
61
62 % Update handles structure
63 guidata(hObject, handles);
64
65 % UIWAIT makes untitled wait for user response (see UIRESUME)
66 % uiwait(handles.figure1);
67
68
69 % --- Outputs from this function are returned to the command line.

```

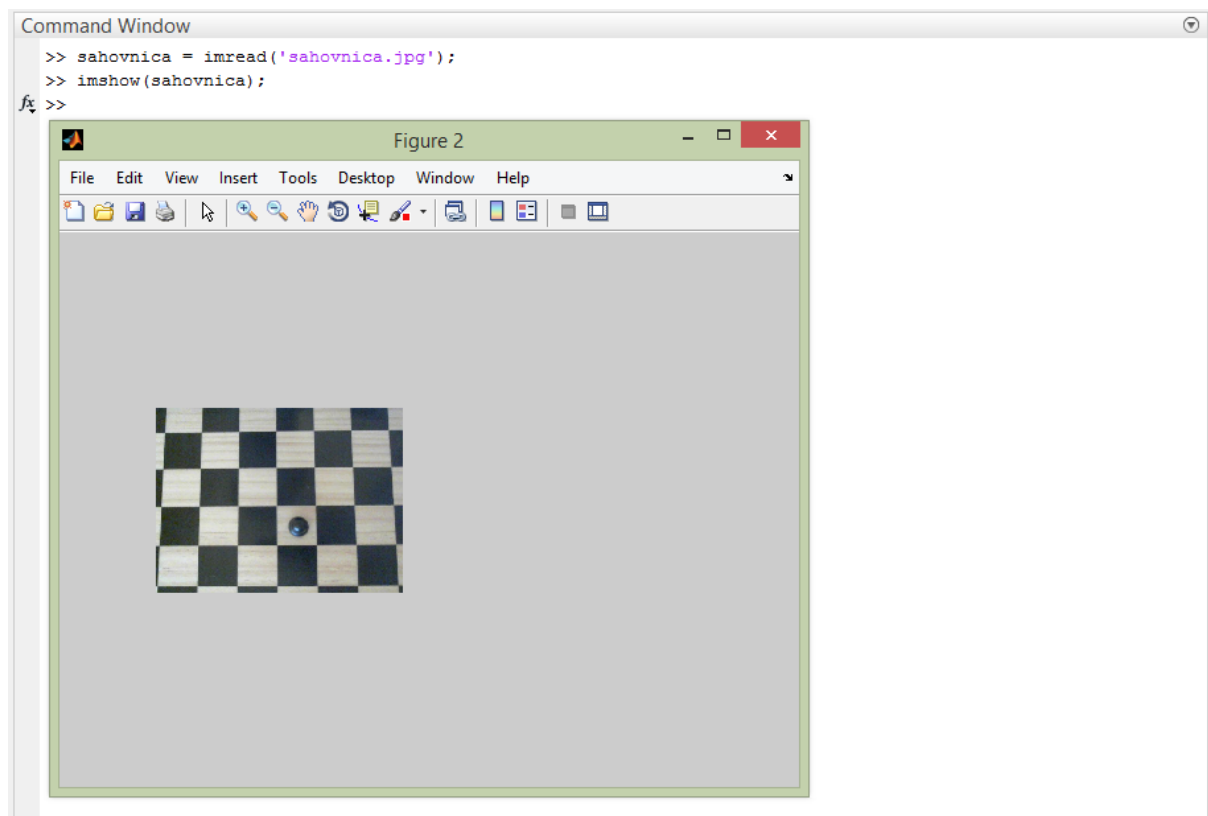
Slika 21: Predogled zajemanja slike

Za zajemanje slik se vrnemo v ukazno okno, kjer zopet definiramo kamero »kamera = videoinput('winvideo', 2);«. Sledi ukaz s katerim opišemo vrsto slike. V mojem primeru bo to RGB slika. Naslednja vrstica poda ukaz, da zajamemo slika. Slika se bo shranila v spremenljivko »slika«. Sedaj jo lahko prikočemo z ukazom »imshow = (slika)«.

Raziskovalna naloga - Uporaba strojnega vida pri igranju šaha



Slika 22: Zajemanje slike v komandnem oknu



Slika 23: Zajeta šahovnica

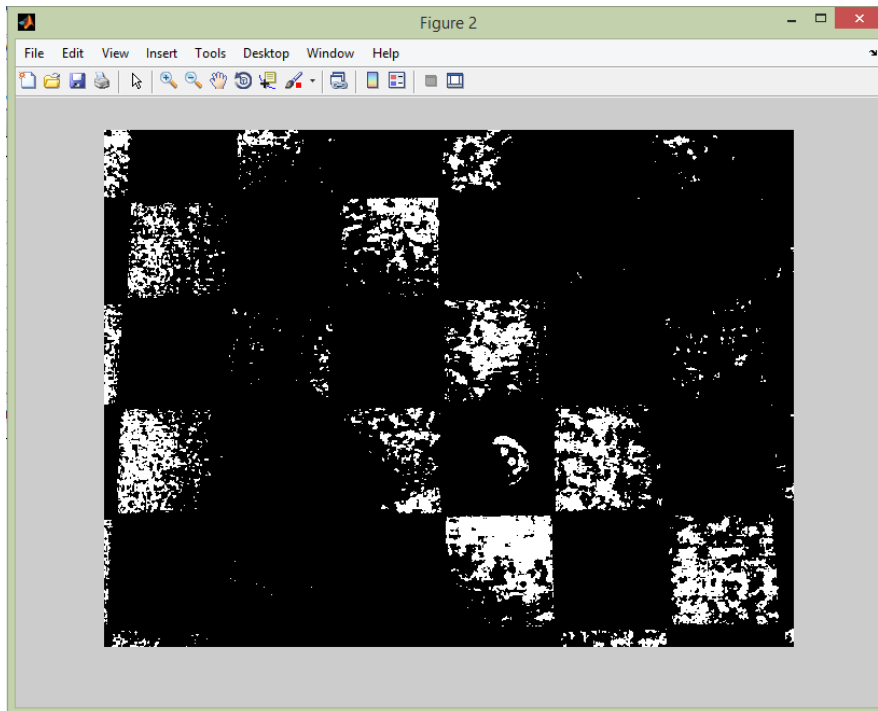
Sedaj, ko smo zajeli sliko, se lahko lotimo prepoznavanja. Najlažji način je, da prepoznamo barve. Z ukazom »matrix = impixel(sahovnica);« se nam odpre novo okno. Tukaj označimo točke. Točke, katere bomo izbrali se bodo pokazale v obliki matrice. V njej bodo števila od 0 do 255 v treh vrstah. To so RGB števila, ki tvorijo dane piksele. Ne smemo pa pozabiti ukaza, ki naroči MATLAB-u, da naj naredi spremenljivke za vsako barvo posebej, »red = sahovnica (:,:,1); green = sahovnica (:,:,2); blue = sahovnica (:,:,3);«. Z matematičnimi neenakostmi lahko opišemo izbrane piksele in za rezultat dobimo samo želen objekt.

	1	2	3
1	48	50	63
2	43	55	51
3	60	65	71
4	52	59	69
5	47	53	67
6	48	63	56
7	32	45	54
8	47	53	67
9	53	61	63
10	52	61	68
11	44	55	57

Slika 24: Matrika RGB števil

```
>> red = sahovnica (:,:,1); green = sahovnica (:,:,2); blue = sahovnica (:,:,3);  
>> matrix = impixel(sahovnica);  
>> rezultat = red<40 & blue<50 & green<65;  
>> imshow(rezultat)
```

Slika 25: Postopek ukazov za pridobitev slike.

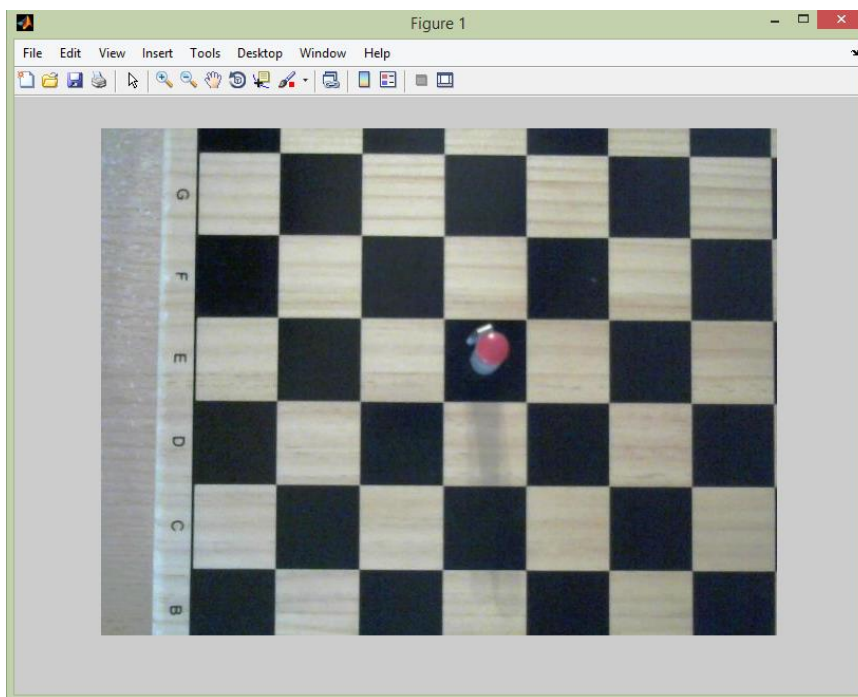


Slika 26: Pridobljena slika

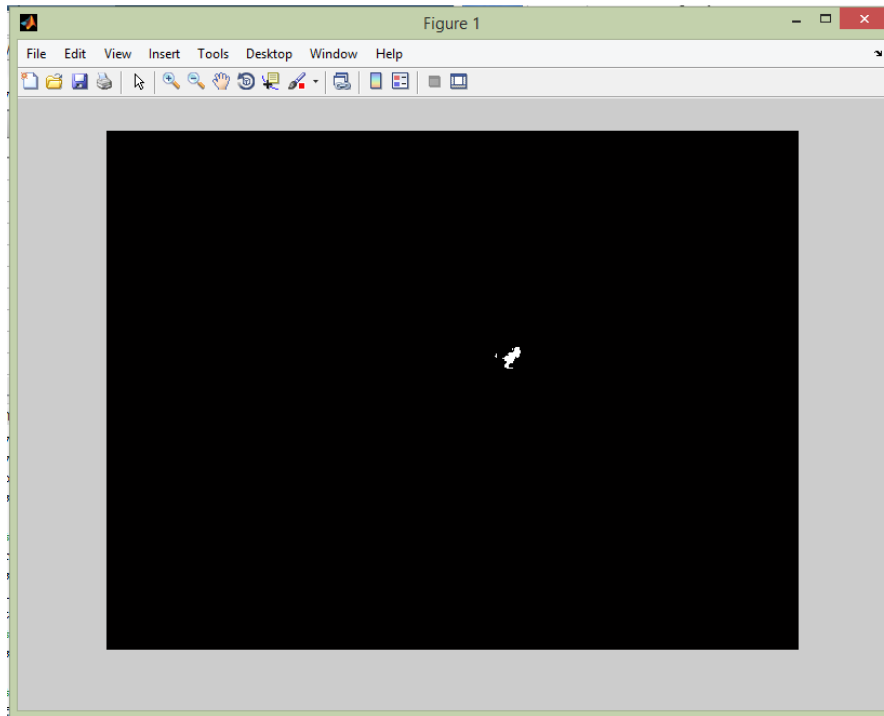
Z rezultatom nisem bil zadovoljen. Bela barva pomeni ujemanje pikslov, ki sem jih pred tem napisal. Torej del barve figure je enaka kot barva črnih polj in je za to nemogoče razločiti. Tudi na sami figuri je vrednost pikslov različna. Če se vrnemo na zajeto sliko je viden odsev svetlobe in ta vpliva na spremembo. Najprej je potrebno zagotoviti konstanten tok svetlobe na ploščo. Druga možnost je, da spremenim barvo figure. S tem bi dosegel večjo razliko pikslov in se tem boljše prepoznavanje.

Na ploščo sem dal rdeč zamašek ter ponovil postopek.

Raziskovalna naloga - Uporaba strojnega vida pri igranju šaha



Slika 27: Šahovnica z rdečim predmetom



Slika 28: Rezultat zaznavanja barve

Na sliki je dobro razviden predmet, čeprav ni celoten, ga lahko prepoznamo na plošči. Dejavniki, ki zraven nekonstantve svetlobe vplivajo na rezultat je število izbranih točk na sliki. Izbral sem samo nekaj točk, kar je omejilo velikost pikslov. Potem je tukaj izbira pikslov, katere naj MATLAB izbere. Z natančnim določanjem lahko prav tako pridemo do boljšega rezultata.

Torej spremembo barv vsake figure lahko določimo za kakšno figuro gre.

8 ZAKLJUČEK

Naloga je zahtevala veliko dela in predvsem znanja v zvezi z uporabo MATLAB-a. Zastavljene hipoteze na začetku sem potrdil ali zavrgel.

Tabla za prestavljanje figur opravlja svoje delo, prav tako tudi prijemalo. Čeprav potrebuje nekaj nadgradnje, hipotezo potrjujem. Zaznavanje barv s spletno kamero ni povzročalo večjih težav. Pri celotni nalogi sem poizkušal čim več narediti sam. Napajalnik bi sicer bil zanesljivejši, če bi ga kupil, a sem za to potrdil tudi to hipotezo. Glede na to, da sem za pogon uporabil servomotorje in ne koračnih, da sem uporabil nizkocenovno spletno kamero ter da je ogrodje iz lesa, je potrjena hipoteza.

Da bi se robot sam odločal o najboljši potezi, bi moral uporabiti že narejen program (chess engine), ki bi najboljšo potezo poslal MATLAB-u. Tukaj pa se stvar oteži, zato to hipotezo ovržem.

9 VIRI

[1] *Arduino UNO* (online). (citirano 9. 3. 2015). Dostopno na naslovu:

<http://gadgetgangster.com/news/53-reviews/316-review-arduino-uno-part-i.html>

[2] *MATLAB procesiranje slik* (online). (citirano 9. 3. 2015). Dostopno na naslovu:

<http://www.mathworks.com/products/computer-vision/features.html#object-detection-and-recognition>

[3] *Pulzno-širinska modulacija* (online). (citirano 9. 3. 2015). Dostopno na naslovu:

<https://sites.google.com/site/programiranjejpjic/home/vaja-12>

[4] *Zgodovina strojnega vida* (online). (citirano 21. 2. 2015). Dostopno na naslovu:

<https://www.epicsysinc.com/blog/machine-vision-history>

IZJAVA

Mentor, Robert Ojsteršek dipl. inž. mehatronike UN, v skladu z 2. in 17. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi naslovom

Uporaba strojnega vida pri igranju šaha,

katere avtor je Denis Furman:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje

Celje, 12.3.2015

žig šole

Podpis mentorja

Podpis odgovorne osebe