

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 2110

**AUTOMATSKO PREPOZNAVANJE
ZVIJEZDA**

Hrvoje Šalković

Zagreb, lipanj 2011.

Sadržaj

1. Uvod	1
2. Algoritmi za prepoznavanje zvijezda	4
2.1. Vrste algoritama	5
2.1.1. Klasifikacijski algoritmi	5
2.1.2. Regresijski algoritmi	6
2.1.3. Kategorični algoritmi računanja slijeda	6
2.1.4. Analizirajući algoritmi	6
2.2. Odabir algoritma	7
3. Identifikacija zvijezda	9
3.1. Idejno rješenje problema	11
4. Implementacija algoritma	13
4.1. Odabir 100 najsvjetlijih zvijezda	18
4.2. Izoliranje 400 najsvjetlijih zvijezda	23
4.3. Parametrizacija podataka	24
4.4. Prepoznavanje zvijezda	27
Zaključak	28
Literatura	29
Sažetak	30

1. Uvod

Astronomija je prirodna znanost koja se bavi proučavanjem nebeskih objekata (npr. zvijezda, planeta, kometa, maglica, zvjezdanih klastera i galaksija) te pojava u Zemljinj atmosferi (npr. pozadinsko kozmičko zračenje). Bavi se evolucijom, fizikalnim i kemijskim svojstvima, te kretanjem nebeskih objekata, kao i problemom nastanka svemira.

Astronomija je također jedna od najstarijih znanosti poznata ljudima. Još u drevnim kulturama mogu se naći astronomski spomenici (npr. *Stonehenge*), a već i rane civilizacije (poput Babilonaca, Grka, Kineza i sl.) počele su se baviti promatranjem zvjezdanog neba. No, tek izumom teleskopa (1608. godine) se astronomija počela razvijati u modernu znanost kakvu je poznajemo danas. Iako se astronomija povijesno povezivala s mnogo različitih disciplina (poput nebeske navigacije, računanja kalendara, pa čak i astrologija), danas se profesionalna astronomija poistovjećuje s astrofizikom.

Tijekom dvadesetog stoljeća profesionalna astronomija se podijelila na promatračku i teoretsku. Promatračka astronomija se bavi prikupljanjem podataka od promatranja nebeskih tijela, dok se teoretska orijentira prema razvoju matematičkih modela za opisivanje nebeskih tijela i pripadnih fenomena. Obje grane su međusobno povezane, jer se teoretska astronomija trudi objasniti

rezultate promatračke, dok rezultati promatračke astronomije služe za provjeru teorija.

Naravno, u današnje vrijeme, kad je oprema poput teleskopa, digitalnih kamera visoke rezolucije i sl. prilično dostupna široj populaciji, profesionalni astronomi nisu jedini koji proučavaju zvjezdano nebo. Čak štoviše, astronomi amateri su pridonijeli mnogim velikim astronomskim otkrićima, a astronomija je postala jedna od rijetkih znanosti gdje čak i amateri mogu imati aktivnu ulogu u otkrivanju novih fenomena.

U ovom radu se doduše, nećemo baviti svim tim astronomskim fenomenima, već automatskim prepoznavanjem zvijezda na digitalnoj fotografiji. Kako bi neke stvari bile jasnije potrebno je objasniti pojam astrofotografije, pošto je astrofotografija neka vrsta baze od koje sve počinje.

Astrofotografija je u biti samo specijalna vrsta fotografije koja sadrži razna nebeska tijela (nama su zbog teme rada najzanimljivije zvijezde), te velika područja noćnog neba. Prve fotografije nebeskog tijela (u ovom slučaju Mjeseca) nastale su 1840. godine, no tek se u kasnom 19. stoljeću pojavljuju prve detaljne zvjezdane fotografije. Uz mogućnosti detaljnog snimanja nebeskih tijela poput Sunca, Mjeseca i ostalih planeta, astrofotografija također ima sposobnost prikazati slike predmeta nevidljivih ljudskom oku poput zamućenih zvijezda, izmaglica ili galaksija. To se postiže dugotrajnom ekspozicijom, je su digitalne kamere sposobne akumulirati i zbrojiti ukupan doprinos

svjetline svih fotona kroz taj duži vremenski period. U amaterskim primjenama, astrofotografija se više koristi za stvaranje 'oku ugodnih' slika nego za neke prave znanstvene doprinose, a u današnje vrijeme, razvojem tehnologije, taj zadatak postaje sve lakši.

Naravno, nije dovoljno samo imati dobar fotoaparat ili dobru kameru. Za profesionalne primjene potrebno je mnogo više. Prava oprema za promatranje zvijezda, u današnje vrijeme podrazumijeva teleskop koji ima mogućnost automatskog praćenja i raspoznavanja zvijezda. Profesionalni astronomi koriste mnoge složene algoritme za praćenje kretanja zvijezda i ostalih nebeskih tijela, koje su za nas obične smrtnike vjerojatno neshvatljivi. U ovom radu pokušat ćemo napraviti jednostavniji algoritam za automatsko prepoznavanje zvijezda na digitalnoj fotografiji, jer se ne bavimo profesionalnom astronomijom. Taj algoritam će, naravno, biti daleko od savršenog, ali će pokušati najbolje aproksimirati tražene rezultate.

Razvijeni algoritam se možda neće boriti s onim najsloženijima, ali to nije niti moguće tražiti. Ovaj će se algoritam bazirati na tome da se bavimo samo s jednim manjim dijelom neba, jer je stvaran broj zvijezda na nebu prevelik za takvo nešto. Više o samom algoritmu i problematici primjene bit će u nastavku rada. Zasad, recimo samo da, iako ovaj rad neće pridonijeti nekim revolucionarnim otkrićima, ipak ima svoju značajnost.

2. Algoritmi za prepoznavanje zvijezda

Digitalnu sliku neke skupine zvijezda možemo promatrati u grubo kao skup bijelih točaka na crnoj pozadini (u savršenim uvjetima). S takvim shvaćanjem algoritam za automatsko prepoznavanje zvijezda možemo svesti na neki algoritam za prepoznavanje uzoraka (engl. *pattern recognition algorithm*).

Takav algoritam može se koristiti za uspoređivanje uzoraka koje tvore zvijezde na digitalnoj fotografiji sa već unaprijed zabilježenim međusobnim odnosima zvijezda u nekom katalogu (za potrebe ovog rada koristi se katalog Tycho2). Naravno, bilo bi preopširno i prekomplicirano takav algoritam primjenjivati na cijeli katalog jer on sadrži zapise o milijunima zvijezda, pa se mora barem otprilike znati koji dio neba je slikan, kako bi se algoritam mogao ograničiti na neko područje u okolini zvijezda koje su na toj slici. Na tako ograničenom području algoritam uspoređuje uzorke (tj. međusobne odnose bliskih zvijezda) na slici sa podacima iz kataloga, te na temelju nekog kriterija (koji će biti objašnjen kasnije) pokušava odrediti o kojim je zvijezdama riječ.

Postoje mnogi algoritmi za prepoznavanje uzoraka, a u daljnjim poglavljima bit će navedena nekolicina tih algoritama i njihove klasifikacije.

2.1.Vrste algoritama

Prepoznavanje uzoraka se definira kao pridruživanje neke vrste izlazne vrijednosti (oznaka; engl. *label*) nekoj unaprijed poznatoj ulaznoj vrijednosti (instanca; engl. *instance*) prema nekom specifičnom algoritmu. Jedan od primjera takvih algoritama je klasifikacija, odnosno klasifikacijski algoritmi, koji pokušavaju ulaznoj vrijednosti dodijeliti neku klasu (najjednostavniji primjer bi bio određivanje je li neka e-mail poruka 'spam'). Naravno osim klasifikacijskih algoritama postoje još brojne druge vrste algoritama.

Neki od njih su: regresijski algoritmi (dodjeljuju realnu izlaznu vrijednost svakoj ulaznoj), algoritmi označavanja slijeda (dodjeljuju klasu svakom članu nekog slijeda vrijednosti), te analizirajući algoritmi (dodjeljuju analizirajuće stablo ulaznoj rečenici).

2.1.1. Klasifikacijski algoritmi

Pod klasifikacijske algoritme možemo uvrstiti: klasifikator maksimalne entropije (engl. *maximal entropy classifier*), 'naivni' Bayesov klasifikator (engl. *naive Bayes classifier*), neuronske mreže (engl. *neural networks*), algoritam k najbližih susjeda (engl. *k-nearest-neighbor*, skraćeno *KNN*) i drugi.

Za zadatak ovog rada najpovoljniji izbor je algoritam k najbližih susjeda (pojednostavljeni oblik, uz dodatne postupke, koji će biti objašnjen u daljnjim poglavljima ovog rada).

2.1.2 Regresijski algoritmi

Iako niti jedan od ovih algoritama neće biti korišten, vrijedi spomenuti neke od njih. Postoje nadzirani (linearna regresija, Gaussova regresija procesa, te neuronske mreže) i nenadzirani (analiza glavnih komponenti, te analiza nezavisnih komponenti). Navedeni algoritmi neće biti posebno objašnjeni, pošto u rješavanju zadatka ovog rada neće biti korišteni, kao što je to već prethodno rečeno.

2.1.3. Kategorični algoritmi označavanja slijeda

Kao što je to bio slučaj i sa regresijskim algoritmima, ovdje također postoje nadzirani i nenadzirani algoritmi. Nadzirani su skriveni Markovljevi modeli (engl. *hidden Markov models*), koji ujedno mogu biti i nenadzirani, Markovljevi modeli maksimalne entropije (engl. *maximum entropy Markov models*), te uvjetna slučajna polja (engl. *conditional random fields*). Ovi se algoritmi također neće koristiti u rješavanju zadatka ovog rada, pa neće biti detaljno objašnjavani.

2.1.4. Analizirajući algoritam

Naziv algoritma je vjerojatnosna kontekstno neovisna gramatika (engl. *probabilistic context free grammar*), a može biti nadzirani ili nenadzirani. Ali, ovaj algoritam nije niti moguće

primijeniti na rješavanje zadatka ovog rada, pa neće biti posebno objašnjavan.

2.2. Odabir algoritma

Kao što je već bilo objašnjeno u prethodnim poglavljima, digitalnu fotografiju skupine zvijezda možemo razmatrati kao skupinu bijelih točaka na crnoj pozadini. Kako bi odredili koje se zvijezde nalaze na fotografiji, moramo barem otprilike znati koji dio neba je slikan, te usporediti oblike koje tvore zvijezde na fotografiji s podacima o tim zvijezdama u katalogu.

Za određivanje tih oblika najbolje odgovara algoritam k najbližih susjeda (KNN), pa će se za rješavanje zadatka ovog rada koristiti pojednostavljena verzija ovog algoritma. Prepoznavanje zvijezda započinje s najsajnijom zvijezdom na slici, na način da se ona uparuje sa po dvije susjedne zvijezde, čineći s njima trokut, i tako za svih k prvih susjeda (k je neki cijeli broj, a za potrebe ovog rada uzet će se da je $k=10$). Ovaj algoritam odabran je upravo zbog razloga što digitalna fotografija ne mora biti u istom mjerilu kao podaci iz kataloga, a omjeri krakova tih trokuta (točnije rečeno kutova) i kut između krakova će biti isti bez obzira na mjerilo.

Dodatno, zvijezde na fotografiji mogu (zbog raznih smetnji) biti manje svjetline nego što je navedeno u katalogu, te mogu biti pomaknute po x i y osima u odnosu na svoje susjede, što također

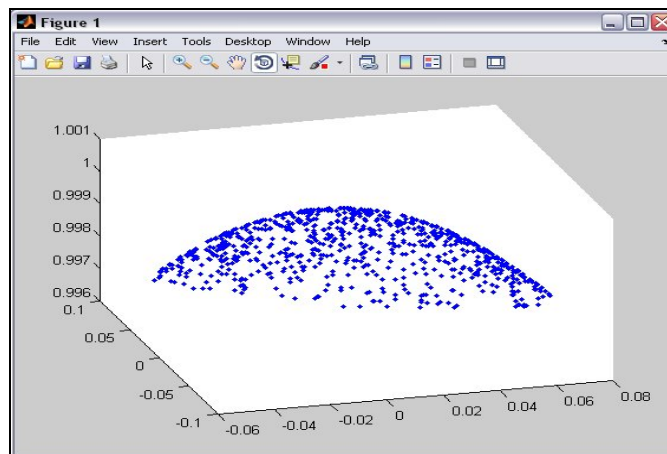
komplicira problem identifikacije zvijezda. Takve se probleme, nažalost, ne može izbjeći, ali ovaj algoritam je sigurno najbolje rješenje jer je neovisan o prostornoj skali, rotaciji slike i pomaku ishodišta fotografije.

Odabrani je algoritam također i relativno jednostavno za implementirati, jer je prilično intuitivan. Naravno, postoje i problemi koje ovaj algoritam ne može riješiti, ali oni i nemaju preveliki utjecaj na konačno rješenje zadatka. Jedini negativni utjecaj takvih problema je taj da će rezultati automatskog prepoznavanja zvijezda biti aproksimativni, te će postojati i mogućnost pogrešne identifikacije pojedinih zvijezda na fotografiji.

3. Identifikacija zvijezda

Za početak, korištenje algoritma ćemo ograničiti na jedan određeni dio neba i određeni broj zvijezda, jer bi inače izvođenje algoritma bilo previše komplicirano. Za rješavanje ovog dijela ćemo se prvo referencirati na zvezdani katalog Tycho2 dostupan na internetu preko sustava Vizier. Na početnom izborniku kataloga dana je mogućnost prikaza velikog broja parametara (uključujući redni broj zvijezde, magnituda, desno uzdizanje, deklinacija...) od kojih će za rješanje zadatka ovog rada biti potrebni samo neki.

Ovaj problem ćemo pojednostaviti tako da će se odabrati područje od 240 kutnih minuta oko zvijezde *Polaris* (Sjevernjača), sa otprilike njenih 700 najbližih susjeda, do jedanaeste magnitude¹.



Slika 1- 3-D prikaz odabranog dijela neba oko zvijezde Polaris

¹ Svjetlina objekta na noćnom nebu. Svjetle zvijezde su opisane malenim ili negativnim brojevima, dok su slabo vidljivi objekti opisani velikim vrijednostima. Skala je logaritamska, što znači da je zvijezda magnitude 1 točno 100 puta svjetlija od zvijezde magnitude 6.

Iako se ovakav prikaz dobije pomoću podataka direktno izvučenih iz kataloga, taj kadar se, zbog jednostavnosti rješavanja problema, neće koristiti kao trodimenzionalni, već će biti sveden na dvije dimenzije, kao da je riječ o fotografiji. Takvom restrikcijom dolazi i do mogućnosti javljanja novih problema, zbog odbacivanja treće dimenzije, ali se uvelike olakšava rješenje zadatka, odnosno prepoznavanje zvijezda na fotografiji.

U samoj implementaciji će se samo simulirati fotografija neba (opet, zbog pojednostavljenja problema) na način da će se iz odabranog kadra izdvojiti manja slika (više o tome kako u nastavku rada) koju će se zatim rotirati, translirati i uvesti manje perturbacije na pozicijama, te magnitudama svih zvijezda u tom kadru kako bi se dobio dojam da je to stvarno prava fotografija neba. Te će se promjene na kadru (perturbacije) morati ponavljati za svako pokretanje programa, i svaki put će morati biti slučajnog iznosa, dodatno naglašavajući dojam da se radi o stvarnim fotografijama zvjezdanog neba.

Uz takvu pretpostavku, da je simulirana slika ustvari prava fotografija zvjezdanog neba, može se jednostavnim matematičkim algoritmima izvući opisne značajke svih zvijezda na slici, te se zatim iste pohraniti u odgovarajuće strukture za daljnju obradu. Nakon pohrane ti će se podaci o zvijezdama na simuliranoj fotografiji uspoređivati sa podacima iz kataloga zvijezda u svrhu identificiranja zvijezda, pronalaženjem traženog indeksa i oznake pojedine zvijezde u strukturi.

3.1. Idejno rješenje problema

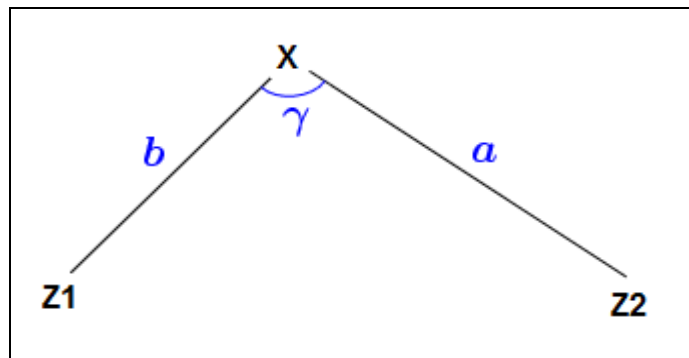
Uz sve navedeno u prethodnom dijelu poglavlja, nameće se ideja za rješenje zadatka. Ta ideja je uzeti manji dio neba u pravokutnom obliku sa kadra oko zvijezde *Polaris* (koja se koristi kao primjer u ovom radu; naravno može se uzeti područje oko bilo koje druge zvijezde), zatim na tom pravokutniku izolirati 100 najsjajnijih zvijezda, te nakon svega taj pravokutnik translacijom i rotacijom centrirati kako bi se dobio potpun dojam stvarne digitalne fotografije zvjezdanog neba.

Nakon toga uzima se jedan veći komad neba koji će nekim dijelom ili u cijelosti prekrivati manji komad neba koji predstavlja simulaciju digitalne fotografije zvjezdanog neba. Na tom većem dijelu bit će potrebno izolirati 400 najsjajnijih zvijezda, te nakon toga izvršiti parametrizaciju na obje slike. Manja slika mora nekim dijelom biti prekrivena većom slikom iz jednostavnog razloga, kako bi usporedbe koje će se vršiti između manje slike i veće, odnosno uparivanje zvijezda na manjoj s onom u katalogu imalo sa sigurnošću nekakve rezultate, ne nužno sa 100% točnošću. U najboljem slučaju veća slika će u cijelosti prekrivati manju, tako da bi i rezultati u takvom slučaju bili puno točniji. Takav slučaj će biti siguran jedino ako za područje veće slike odaberemo cjelokupno područje oko zvijezde *Polaris* (koju ćemo razmatrati za izvedbu ovog algoritma).

Pod parametrizaciju se smatra zapisivanje ključnih podataka o zvijezdama koji će se koristiti pri usporedbi manje i veće slike,

odnosno pri identifikaciji zvijezda na simuliranoj fotografiji (manjoj slici).

Ti ključno podaci za svaku zvijezdu će biti njezini međusobni odnosi s njenih 10 najbližih susjeda, indeks te zvijezde, njena magnituda, te njezine x i y koordinate koje će ona zauzimati u danom sustavu. Sama struktura koja će opisivati odnos zvijezde s njezinim susjedima sastojat će se od parova γ i d , gdje će γ predstavljati kut kojeg zatvaraju a i b , dok će d predstavljati omjer a i b ($d = \frac{a}{b}$). U ovoj notaciji a i b će predstavljati euklidske udaljenosti trenutno razmatrane zvijezde od dviju njezinih najbližih susjeda, te pri tome još treba paziti da je a uvijek veće od b kako bi vrijednost d uvijek bila veća od 1. Za svaku zvijezdu bit će 10 takvih trojki (za 10 najbližih susjeda te zvijezde), što znači da će se struktura svake zvijezde sastojati od 10 matrica koje će u sebi sadržati 2×9 podataka (dva stupca, od kojih jedan predstavlja d , a drugi γ).



Slika 2 - pojednostavljeni prikaz strukture za zvijezdu

4. Implementacija algoritma

Sama implementacija razvijenog algoritma bit će napravljena u programskoj okolini Matlab, kako je bilo i navedeno u zadatku. Rješenje će biti izvedeno pomoću glavne skripte, u kojoj će dodatno biti implementirane funkcije radi jednostavnijeg i logičnijeg (odnosno preglednijeg) izvođenja samog algoritma.

Prvo što se izvodi u skripti je učitavanje podataka o zvijezdama u odgovarajuće strukture pomoću funkcije `ucitaj_bazu`, pomoću koje će biti iscrtan i 3D prostor oko zvijezde *Polaris* s njezinih približno 800 susjeda. Ta će funkcija spremiti podatke o zvijezdama iz kataloga Tycho2 u matricu u kojoj će se nalaziti podaci o rednom broju zvijezde u katalogu (**ind**; u prvom stupcu), indeksu zvijezde unutar Tycho2 kataloga (**TYC1**, **TYC2**, **TYC3** koji će biti spremljeni u strukturu **TYC**; u drugom stupcu matrice), podaci o desnom uzdizanju zvijezde uz "proper motion" (**RAmdeg**; u petom stupcu), deklinaciji (**DEmdeg**; u šestom stupcu), te podatke o magnitudi zvijezde (**BTmag** i **VTmag**, pomoću kojih se izračunava **Vmag**; u sedmom i osmom stupcu). Ova će funkcija dodatno, pomoći desnog uzdizanja i deklinacije, kojima se prikazuje pozicija zvijezde u kutnim stupnjevima, izračunati poziciju zvijezde u 3D prostoru (odnosno x, y i z koordinate).


```

ind=ma_tycho2(:,1);
TYC=ma_tycho2(:,2:4);
ang=ma_tycho2(:,5:6)/180*pi;

BTmag=ma_tycho2(:,7);
VTmag=ma_tycho2(:,8);
Vmag=VTmag-0.09*(BTmag-VTmag);

x=cos(ang(:,2)).*cos(ang(:,1));
y=cos(ang(:,2)).*sin(ang(:,1));
z=sin(ang(:,2));

pos=[x y z];

```

Slika 3 - dio koda funkcije `ucitaj_bazu` za izračunavanje pozicije zvijezda

Na gornjoj slici vidljiv je način izračuna magnitude zvijezde ($V_{\text{mag}} = V_{\text{Tmag}} - 0.09 * (B_{\text{Tmag}} - V_{\text{Tmag}})$), te pozicije zvijezde u 3D prostoru izračunom njenih prostornih koordinata x , y i z iz desnog uzdizanja i deklinacije zvijezde.

No, za ovaj zadatak nije nam potrebna treća dimenzija, jer se ona i ne može prikazati na digitalnoj fotografiji, već se koriste samo parametri x i y , te redni broj zvijezde (ind) u katalogu, te magnituda zvijezde koje se učitavaju u matricu `Atlas`. Na taj se način dobiva za rješenje ovog zadatka potrebna projekcija na 2D prostor. U toj projekciji se zatim odabire pravokutnik koji će predstavljati simuliranu fotografiju pomoću Matlab naredbe `imrect`. Na toj manjoj slici (odnosno odabranom pravokutniku) označavamo 100 najsjajnijih zvijezda sortiranjem po magnitudi u padajućem slijedu. Jedini

problem je što naredba `imrect` izbacuje pravokutnik unaprijed određene visine i širine, te bez nagiba, pa se korisniku daje mogućnost podešavanja širine i visine pravokutnika. Nagib pravokutnika se mora uzeti kao ulazni podatak iz radnog prostora Matlaba, jer naredba `imrect` u sebi ne podržava mogućnost zakrivljenja zadanog pravokutnika. Taj nagib će u implementaciji predstavljati nakrivljenost kamere u trenutku snimanja tražene fotografije na kojoj će se vršiti daljnja obrada.

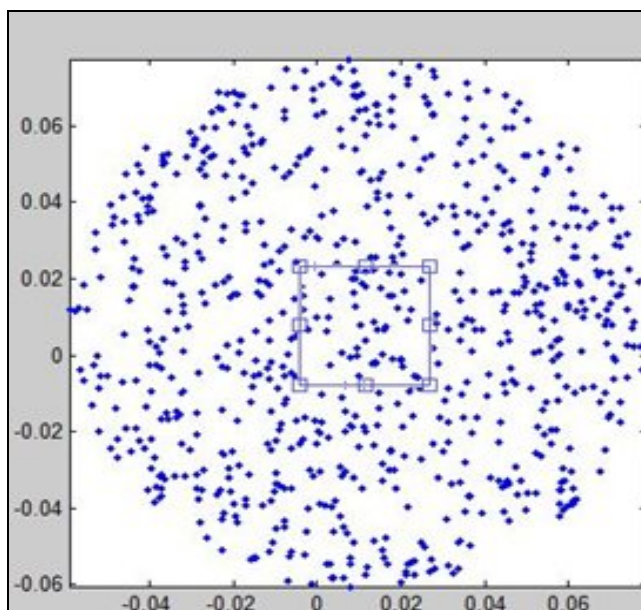
```
axis equal
axis([Xmin Xmax Ymin Ymax])

%[korX korY]=ginput(4);
h = imrect(gca, [Xmean-sirina/2 Ymean-visina/2 sirina visina]);
accepted_pos = wait(h);
delete(h);

w=accepted_pos(3); %width
h=accepted_pos(4); %height
x=accepted_pos(1);y=accepted_pos(2); %corner position
xv=[x x+w x+w x x];yv=[y y+h y+h y];
R(1,:)=xv;R(2,:)=yv;
kut=input('Unesite kut rotacije pravokutnika:');
kutZaTocku=-kut;
alpha=kut*2*pi/360;
XY=[cos(alpha) -sin(alpha);sin(alpha) cos(alpha)]*R;%matrična transformacija za rotaciju
hold on
% plot(xv,yv);
plot(XY(1,:),XY(2,:),'r');
```

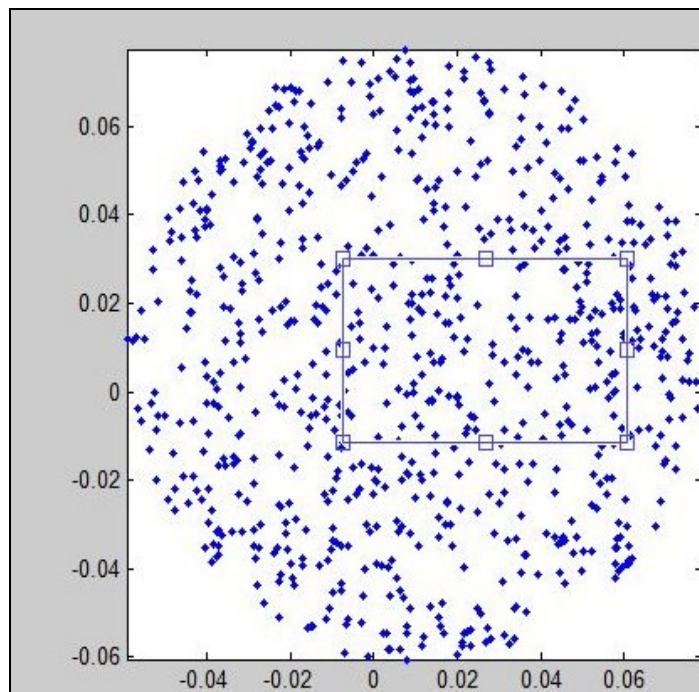
Slika 4 - dio koda koji prikazuje podešavanje pravokutnika od strane korisnika

U gornjem izvodu iz koda vidi se način na koji korisnik može odabrati visinu i širinu pravokutnika koji će simulirati digitalnu fotografiju, te zatim dodatno definirati nagib kamere u trenutku snimanja željenog kadra.



Slika 5 - prikaz početnog pravokutnika dobivenog pomoću `imrect` naredbe

Gornja slika prikazuje kako izgleda početni pravokutnik koji se dobiva nakon izvođenja naredbe `imrect` koji se zatim pomoću miša može razvući na veličinu koja odgovara korisniku, te ga se može i translirati u prostoru kako bi se odabrao željeni kadar za simuliranu fotografiju. Naravno, ovim se korisniku daje mogućnost da odabere mnogo različitih simulacija, te se dobiva određena vjerodostojnost simulirane fotografije.

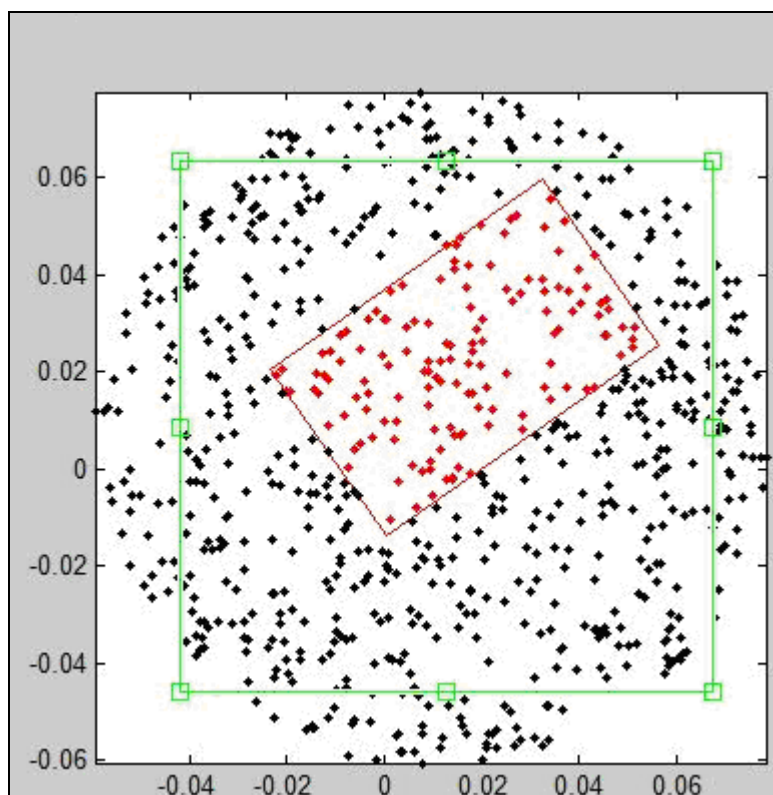


Slika 6 - jedan od mogućih odabira početnog kadra

Nakon što korisnik odabere početni kadar (npr. kao na gornjoj slici), dvostrukim stiskom na tipku miša u radnom prozoru se korisniku daje mogućnost odabira nagiba simulirane fotografije, kako bi se dobio što vjerniji prikaz stvarno uzete digitalne fotografije zvjezdanog neba.

4.1. Odabir 100 najsvjetlijih zvijezda

Nakon što je korisnik uspješno odabrao manji pravokutnik, odnosno simuliranu fotografiju (koja mora sadržati barem 100 zvijezda), pomoću dodatnih se funkcija provjerava koje su točke unutar pravokutnika, te ih se posebno označava na novom prikazu neba.



Slika 7 - prikaz neba s posebno označenim slikama na manjoj slici

```

% Provjera nalazi li se neka točka iz atlasa unutar
% odabranog pravokutnika
brojac=1;
noviIND=[];
noviX=[];
noviY=[];
noviBTmag=[];

brojacNOT=1;
noviINDNOT=[];
noviXNOT=[];
noviYNOT=[];
noviBTmagNOT=[];

N=length(stariX);
kutOrg=kut;
kut=pozitivanKut(kut);
for k=1:N
    logic=isInRectangle(XY',[stariX(k) stariY(k)],
    kut);
    if (logic==1)
        noviIND(brojac)=ind(k);
        noviX(brojac)=stariX(k);
        noviY(brojac)=stariY(k);
        noviBTmag(brojac)=BTmag(brojac);
        brojac=brojac+1;
    else
        noviINDNOT(brojacNOT)=ind(k);
        noviXNOT(brojacNOT)=stariX(k);
        noviYNOT(brojacNOT)=stariY(k);
        noviBTmagNOT(brojacNOT)=BTmag(brojacNOT);
        brojacNOT=brojacNOT+1;
    end
end
end

```

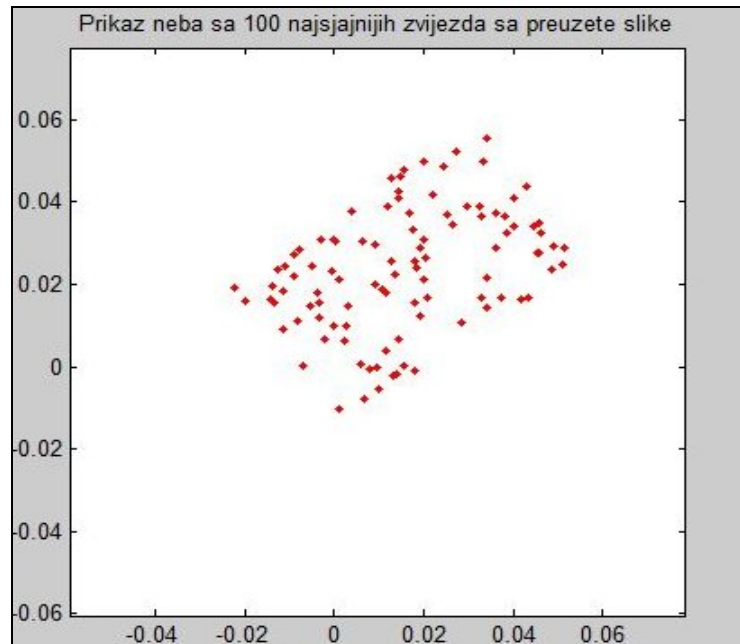
Slika 8 - dio koda kojim se provjerava je li neka točka unutar zadanog pravokutnika

Na gornjoj slici nalazi se samo dio koda kojim se provjerava nalazi li se točka unutar traženog pravokutnika, jer se još dodatno mora paziti ako je nagib pravokutnika 0° ili 90° , pošto se oni tretiraju kao specijalan oblik te funkcije (nagib pravaca kojima je omeđen prostor unutar pravokutnika je 0 za horizontalne, odnosno beskonačno za vertikalne stranice pravokutnika). Nakon što se razriješi pitanje koje se točke nalaze unutar pravokutnika prikazuje se Atlas prikazan na slici 7. Veći, zeleni, pravokutnik na toj slici predstavlja područje (odnosno jedno od mogućih područja) na kojem će se dodatno izolirati 400 najsjajnijih zvijezda potrebnih za usporedbu.

U manjem pravokutniku je zatim potrebno izolirati 100 najsvjetlijih zvijezda, a to se postiže na vrlo jednostavan način, sortiranjem zvijezda po magnitudi od najsvjetlije prema najmanje svijetloj te pohranom u novu matricu (u programu se to izvodi sortiranjem po stupcu u kojem se u matrici nalaze vrijednosti magnituda svih zvijezda unutar tog pravokutnika).

```
if (size(novaMatrica,1)<100)
    error('Odabrana širina slike je
premal a i ne sadrži 100 zvijezda ponovite
odabir');
end
novaMatricaSort4=sortrows(novaMatrica,4);
slika100=novaMatricaSort4(1:100,:);
```

Slika 9 - dio koda koji služi za izdvajanje 100 najsvjetlijih zvijezda na slici



Slika 10 - prikaz samo 100 najsvjetlijih zvijezda u manjem pravokutniku

Nakon što dobijemo sliku sa izdvojenih 100 najsvjetlijih zvijezda, potrebno je tu sliku modificirati kako bi se dobio uvjerljiviji prikaz simulirane fotografije. To se radi na način da se sliku centrira pomoću njezine srednje vrijednosti kako bi se dobio dojam da je slikana pod nulnim kutom (odnosno da izgleda kao fotografija uzeta digitalnim fotoaparatom ili kamerom). Dodatno se još uvode i perturbacije po magnitudi i pozicijama zvijezda kako bi se još više dobilo na autentičnosti simulirane fotografije. Taj dio posla obavlja se kodom koji će biti prikazan na sljedećoj slici.


```

% Ispis na novoj slici samo 100 najsjajnijih zvijezda
hold on
figure, plot(slika100(:,2),slika100(:,3),'.','Color','red');
axis equal
axis([Xmin Xmax Ymin Ymax])
title('Prikaz neba sa 100 najsjajnijih zvijezda sa preuzete slike');

% slika koja je centrirana i rotirana kao da je upravo uzeta sa
% fotoaparatom (100 najsjajnijih samo je ukinut kut sa kojih su točke
% preuzete sa originalnog kataloga)
slikaRC(:,1)=slika100(:,2);%x
slikaRC(:,2)=slika100(:,3);%y
slikaRC=slikaRC';
kutOrg=-kutOrg;
alpha=kutOrg*2*pi/360;
slikaRC=[cos(alpha) -sin(alpha);sin(alpha) cos(alpha)]*slikaRC;
slikaRC=slikaRC';
slikaRC(:,1)=slikaRC(:,1)-Xmean;
slikaRC(:,2)=slikaRC(:,2)-Ymean;
hold on
figure, plot(slikaRC(:,1),slikaRC(:,2),'.','Color','red');
axis equal
axis([Xmin Xmax Ymin Ymax])
R=R';
R(:,1)=R(:,1)-Xmean;
R(:,2)=R(:,2)-Ymean;
hold on
plot(R(:,1),R(:,2),'Color','red');
title('Simulacija: Prikaz slike kao da je uzeta fotoaparatom');

```

Slika 11 - dio koda kojim se rotira i centrira preuzeta slika

Nakon što se izvršavanja gornjeg koda dobiva se slika koja će izgledati približno kao da je slikana fotoaparatom. Postupak za uzimanje veće slike koja nekim dijelom pokriva manju bit će isti kao i postupak uzimanja male slike, ali se izoliranje 400 najsvjetlijih zvijezda na tom području vrši na malo drugačiji način, koji će biti opisan u sljedećem podpoglavlju.

4.2. Izoliranje 400 najsvjetlijih zvijezda

Kao što je već rečeno u prethodnom podpoglavlju, izoliranje 400 najsvjetlijih zvijezda na većoj slici vrši se na malo drugačiji način od izoliranja 100 najsvjetlijih na maloj. Tih 400 zvijezda treba se odabrati na način da uključuju i one u manjoj slici, jer bez takvog odabira algoritam ne bi davao rezultate velike točnosti.

```
function [ outM ] = filter400(
MmalaSVE,MvelikaSVE,Mmala100 )

velikaBezMale=[];
brojac=1;
N=size(MvelikaSVE,1);
num=MvelikaSVE(:,1);

for k=1:N
    inx=find(MmalaSVE==num(k));
    if(isempty(inx)==1)
        velikaBezMale(brojac,:)=MvelikaSVE
(k,:);
        brojac=brojac+1;
    end
end

velikaBezMaleSort=sortrows(velikaBezMale,4);
velikaBezMale300=velikaBezMaleSort(1:300,:);

out=Mmala100;
out(101:400,:)=velikaBezMale300;

outM=sortrows(out,-4);

end
```

Slika 12 - dio koda za izlučivanje 400 najsjajnijih zvijezda na većoj slici

4.3. Parametrizacija podataka

Nakon što smo dobili simuliranu digitalnu fotografiju (odnosno manju sliku), te sliku s kojom ćemo je uspoređivati potrebno je provesti parametrizaciju dobivenih podataka. Ulazni parametri za ovaj dio su matrice koje sadrže obilježja zvijezda koje ćemo promatrati. Te se matrice sastoje od 4 vrste podataka, a to su redni broj zvijezde u katalogu, x i y koordinata zvijezde, te njezina magnituda. Uzima se 10 najbližih susjeda referentne zvijezde i njima se dodjeljuju parametri oblika [Z_br, {d=a/b, γ}], gdje je opisnih parametara, odnosno uređenih parova oblika [d=a/b, γ] ima 10x9, jer je tu zvijezdu potrebno opisati u odnosu na svih 10 njenih najbližih susjeda. U prethodnoj strukturi a i b označavaju duljine krakova koje, zajedno s referentom zvijezdom čine dvije od njezinih 10 najbližih susjeda, dok γ označava kut koji ti krakovi zatvaraju.

```
Mkordinata=[M(:,2) M(:,3)];  
Veuklid=pdist(Mkordinata,'euclidean');%poslozi  
euklidske udaljenosti u vektor  
Meuklid=squareform(Veuklid);%prebaci vektor u matricu  
tako da element u retku zapravo točka, a stupac  
udaljenost od druge točke  
maxelem=size(Meuklid,2);%br redaka=br zvijezda
```

Slika 13 - dio koda za mjerenje udaljenosti među zvijezdama

Nakon što se dijelom koda prikazanim na prethodnoj slici izračunaju udaljenosti između zvijezda, potrebno je odrediti i kutove

koje zatvaraju ti krakovi. Ideja za određivanje je mjerenje tangensa među krakovima, ali problem je što je tangens funkcija koja je zrcalna u prvom i trećem, odnosno drugom i četvrtom kvadrantu, pa se umjesto toga koristi funkcija atan2, koja je ograničena na kutove između -180° i 180° . Mjerenje se radi tako da se postavi jedan referentni krak, a drugi krak se redom spaja sa ostalih 9 susjeda referentne zvijezde. Nakon toga se uzima sljedeća od 10 susjeda i s njom se spoji referentni krak, te se postupak ponavlja dok ne prođemo svih 10 susjeda. Na taj način će za svaku zvijezdu postojati 10 početnih krakova, a samim time i $10 \times 2 \times 9$ opisnih parametara. Ti podaci se zatim spremaju u strukturu S, gdje su zapisani redni broj zvijezde i matrica M sa svim parametrima (dimenzija $10 \times 2 \times 9$). Dio koda kojim se obavlja taj dio posla bit će prikazan na sljedećoj slici.

```

% pronaći koeficijente smjera sa atan2 funkcijom te vektor sortirati od
% najmanjeg kuta prema većemu
vectRBZvijezda=sortVeuklid10(:,1:2);%apsolutno ista matrica
% prolazak kroz sve parove točaka i određivanje kuta
for k2=1:1:10
dvijeTocke=[Mkordinata(k,:);Mkordinata(vectRBZvijezda(k2,:))];
nagib=atan2(dvijeTocke(2,2)-dvijeTocke(1,2),dvijeTocke(2,1)-dvijeTocke(1,1));
vectRBZvijezda(k2,3)=rad2deg(nagib);
end
% sortiranje 10 točaka prema kutu između početne točke(zvijezde) i
% promatrane točke(zvijezde- jedne od 10 okolnih najbližih)
% sortiranje od najmanjeg kuta prema najvećem pri čemu je najmanji mogući kut
% -180 stupnjeva a najveći 180 stupnjeva (definicija atan2 funkcije)
vectRBZvijezdaSort=sortrows(vectRBZvijezda,3);%sad sortiramo po nagibu
%prije je bilo sortirano po euklidu
kutIr=[];
for k4=1:1:10
    for k3=1:1:9 %kruzim s fiksne linije i mijenjam kut
        kutIr(k3,2,k4)=abs(vectRBZvijezdaSort(k4,3)-vectRBZvijezdaSort(mod(k3+(k4-1),10)+1,3));
        if(vectRBZvijezdaSort(k4,2)>=vectRBZvijezdaSort(mod(k3+(k4-1),10)+1,2))
            kutIr(k3,1,k4)=vectRBZvijezdaSort(k4,2)/vectRBZvijezdaSort(mod(k3+(k4-1),10)+1,2);
        else
            kutIr(k3,1,k4)=vectRBZvijezdaSort(mod(k3+(k4-1),10)+1,2)/vectRBZvijezdaSort(k4,2);
        end
    end
end
% kutIr;
S(k,1).brZ=M(k,1);
S(k,1).M3D=kutIr;
end

```

Slika 14 - dio koda za računanje kuta i zapis podataka u strukturu

4.4. Prepoznavanje zvijezde

Nakon izvršene parametrizacije dolazi se do zadnjeg dijela ovog problema, a to je identificiranje zvijezde. To se radi na način da korisnik odabere neku zvijezdu na simuliranoj fotografiji (odnosno manjoj slici koja je centrirana) pomoću pokazivača. Za tu odabranu zvijezdu pokreće se proces identifikacije usporedbama podataka te zvijezde koje se nalaze u opisnoj strukturi simulirane fotografije sa podacima iz opisnim strukturama u velikoj slici. Nakon izvršene usporedbe program vraća svih 90 opisnih vrijednosti iz simulirane fotografije s pronađenim podacima na velikoj slici, i rednim brojem te zvijezde. Pomoću rednog broja zvijezde se može iz kataloga saznati o kojoj je točno zvijezdi riječ.

Zaključak

Automatsko prepoznavanje zvijezda sa digitalne fotografije nije nimalo lagan problem, te iziskuje vrlo složene algoritme ako se žele postići rezultati velike točnosti. Za potrebe ovog rada osmišljen je relativno jednostavan algoritam koji se može koristiti za simulacije prepoznavanja zvijezda. Iako se u radu koncentriralo na područje od oko 800 zvijezda oko jedne od najsvjetlijih zvijezda na noćnom nebu, *Polaris*, odnosno zvijezde sjevernjače, taj algoritam nije ograničen samo na to područje.

Naravno, ovdje razmatrani algoritam se ne bi mogao koristiti u profesionalnom primjenama, ali to nije niti bio njegov cilj. Za algoritam koji bi se mogao koristiti u profesionalnoj astronomiji bilo bi potrebno mnogo više vremena za njegovo osmišljavanje, a i mnogo veća razina znanja. Osmišljavanje takvih algoritama prepustit ćemo profesionalnim astronomima, dok će za našu primjenu i ovakav vrlo pojednostavljeni algoritam biti dovoljan.

Razvijeni algoritam bio je primijenjen na relativno malom broju ulaznih podataka, te je davao rezultate prilično velike točnosti, uz dosta redundancije, tako da se može reći da je na razini primjene za koju je namijenjen prilično uspješan.

Literatura

1. Scharringhausen, B. "Curious About Astronomy: What is the difference between astronomy and astrophysics?"
2. David Malin, Dennis Di Cicco, *Astrophotography - The Amateur Connection, The Roles of Photography in Professional Astronomy, Challenges and Changes*
3. Vizier catalogs, katalog *Tycho2*, <http://vizier.u-strasbg.fr/viz-bin/VizieR-3>
4. <http://en.wikipedia.org/wiki/K-nearest-neighbor>
5. Richard O. Duda, Peter E. Hart, David G. Stork (2001) *Pattern classification* (2nd edition), Wiley, New York, ISBN 0-471-05669-3
6. Hornegger, Joachim; Paulus, Dietrich W. R. (1999). *Applied Pattern Recognition: A Practical Introduction to Image and Speech Processing in C++* (2nd ed.). San Francisco: Morgan Kaufmann Publishers. ISBN 3-528-15558-2.

Sažetak

U radu su razmatrani algoritmi za automatsko prepoznavanje zvijezda na digitalnoj fotografiji. Na početku je dana analogija prepoznavanja zvijezda sa prepoznavanjem uzoraka koje te zvijezde čine.

Odabrani algoritam svodi se na prepoznavanje sličnosti između trokuta koje čine bliske zvijezde približno jednakih magnituda na slici i njihovo uspoređivanje s podacima iz kataloga Tycho2.

Algoritam se pokazao prilično uspješnim u simulacijama prepoznavanja zvijezda na digitalnim fotografijama, te je davao rezultate gotovo savršenih točnosti.

Ključne riječi:

algoritam prepoznavanja, Polaris, magnituda, katalog zvijezda, astrofotografija

Summary

This thesis discussed the possible algorithms for automatic star recognition. An analogy of star recognition with recognizing the patterns the stars made up was given in the introduction.

The selected algorithm is based on recognizing the similarities of the triangles comprised of near stars approximately of same magnitudes with the data from the star catalogue Tycho2.

The algorithm was pretty successful in simulating star recognition on digital photographs, as it gave results of near perfect accuracy.

Keywords:

recognition algorithm, Polaris, magnitude, star catalogue, astrophotography