Deskriptor ključnih točaka inspiriran ljudskim vizualnim sustavom

Strišković, Barbara

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij elektrotehnike

DESKRIPTOR KLJUČNIH TOČAKA INSPIRIRAN LJUDSKIM VIZUALNIM SUSTAVOM

Završni rad

Barbara Strišković

Osijek, 2018.

SADRŽAJ

1. UVOD

Računalni vid je interdisciplinarno područje koje se bavi razumijevanjem digitalnih slika i videa kako bi računala mogla prikupiti što više korisnih informacija iz jedne slike ili dijela slike koje služe za analizu i obradu iste. Slika se može definirati kao funkcija dvije varijable *f(x,y)* pri čemu su x i y prostorne koordinate, a vrijednost funkcije svakog para koordinata se naziva intenzitet ili nivo sive boje slike [1]. Ako su x,y i f konačne, diskretne veličine takvu sliku nazivamo digitalnom slikom. Digitalna slika je sastavljena od konačnog broja elemenata, svaki element ima određenu lokaciju i vrijednost, takvi elementi se nazivaju pikseli.

Jedan od zadataka računalnog vida je prepoznavanje dvodimenzionalnih i trodimenzionalnih objekata te upravo to radi Fast Retina Keypoint (FREAK) algoritam. FREAK algoritam pronalazi zadani objekt na slici koja mu je dana, no problem ovakvih algoritama je da pronađu objekt ako je on u drugom položaju ili druge veličine stoga je neophodno da algoritam bude invarijantan na skalu i rotaciju. Vizualna korespondencija, podudaranje objekata i druge primjene računalnog vida oslanjaju se na slike s malim brojem ključnih točaka. Najvažniji zadatak je učinkovito opisati ključne točke sa stabilnim, kompaktnim i robusnim prikazom invarijantnim na skalu, rotaciju, transformacije i šumove. Prednost FREAK-a naspram drugih algoritama za detekciju i opisivanje značajki je njegova brza izvedba. FREAK je inspiriran ljudskim vizualnim sustavom što znači da algoritam uzorkovanja točaka napravljen po uzoru na ljudski vizualni sustav.

1.1. Zadatak rada

Zadatak ovog rada je pojasniti način rada algoritama *FREAK* koji pripada bibliotekama OpenCV-a. Potrebno je pokazati i objasniti kako algoritam radi te priložiti i pojasniti rezultate eksperimenta. Izložiti prednosti i nedostatke ovog algoritma.

2. DETEKCIJA I OPIS ZNAČAJKI

Prepoznavanje i povezivanje ključnih točaka odnosno značajki (eng. *feature)* slike je važan dio računalnog vida. Značajke su svojstva slike iz kojih izvlačimo informacije potrebne za daljnje obrade i analize. Ako želimo povezati dvije slike odnosno prepoznati određene značajke jedne slike na drugoj slici, potrebno je odrediti značajke koje će biti relevantne. Razlikujemo detekciju i ekstrakciju značajki. Detekcija predstavlja pronalaženje ključnih točaka, dok ekstrakcija objašnjava kako opisati te točke kako bi one bile korisne za daljnje analize i usporedbe.

2.1. Detekcija značajki

Prije same detekcije značajki potrebno je odrediti koja vrsta značajki je najbolja za povezivanje značajki. Značajke mogu biti dijelovi slike s promjenama u kontrastu, promjenama u strukturi, promjenama u orijentaciji ili slično. Potrebno je pronaći svojstvo koje je jedinstveno kako bi pretraga bila što djelotvornija. Ako je odabrano svojstvo koje rotacijom ili nekom drugom promjenom može odgovarati na nekoliko mjesta slike to se smatra loše odabranom značajkom, dakle potrebna je jedinstvenost.

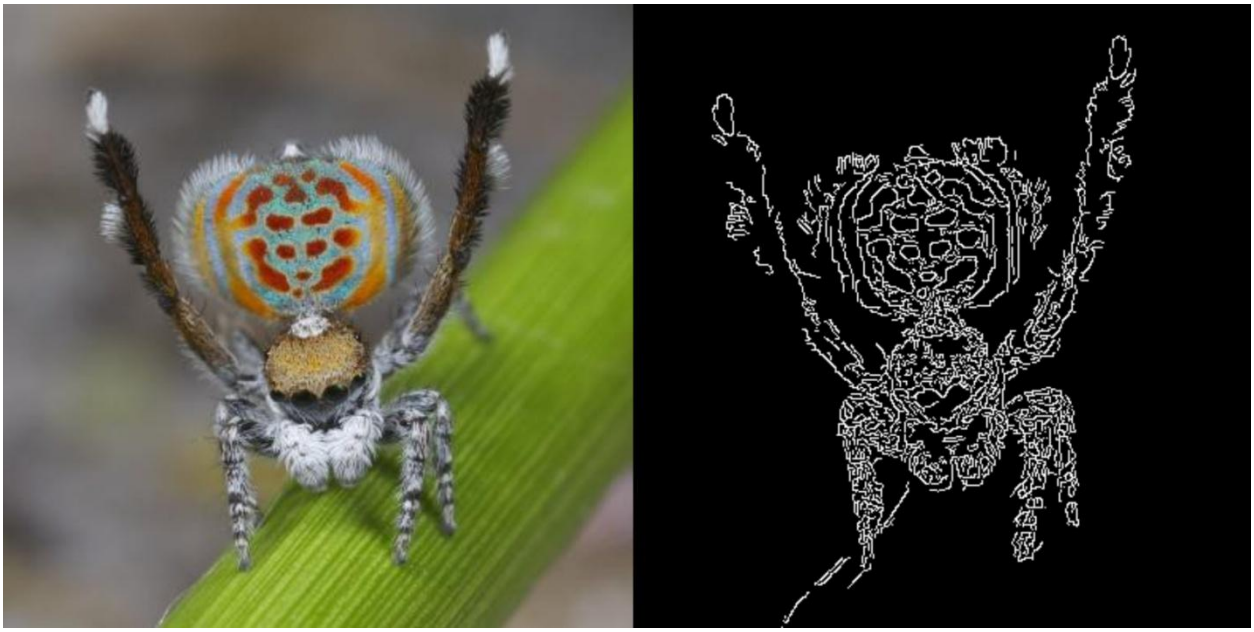
2.1.1. Detekcija rubova

Detekcija rubova ostvaruje se pomoću različitih matematičkih metoda koje ciljaju na pronalaženje točaka digitalne slike gdje se osvjetljenje slike mijenja značajno i diskontinuirano [2]. Mjesta odnosno točke gdje se osvjetljenje slike mijenja značajno su organizirani u skupinu segmenata zakrivljenih linija i to se naziva rubovima. Detekcija rubova je osnovni alat obrade slika, strojnog vida i računalnog vida, posebno u području detekcije značajki i ekstrakcije značajki.

Rubovi izdvojeni iz dvodimenzionalnih slika trodimenzionalne scene mogu se klasificirati kao ovisni o gledištu ili neovisni o gledištu [2]. Rubovi neovisni o gledištu zadržavaju svojstva trodimenzionalne scene ili objekta, kao što su oznake površine ili oblik površine. Rubovi koji su ovisni o gledištu mogu se mijenjati s obzirom na promjenu gledišta i odražavaju geometriju scene. Rub može biti granica između dvije boje no isto tako granica može biti mali broj piksela druge boje naspram okoline . Rubovi dobiveni iz realnih, stvarnih slika uobičajeno nisu savršeni (eng. *ideal step edge)* i najčešće ih prati jedan ili više sljedećih efekata: fokalno zamućenje uzorkovano konačnom dubinom polja i funkcijom širenja konačnih točaka, penumbralna zamagljenost uzrokovana sjenama stvorenim iz izvora svjetlosti nenultog radijusa i zasjenjivanje.

Postoji nekoliko pristupa detekciji rubova, a najpoznatiji su pristup temeljen na pretraživanju i pristup temeljen na nultom križanju (eng. *zero-crossing*). Poznati algoritmi su Canny i Sobel.

Slika 2.1. *Ulaz(lijevo) i izlaz(desno) Sobel algoritma[3]*



Slika 2.2. *Ulaz(lijevo) i izlaz(desno) Canny algoritma[3]*

2.1.2. Detekcija kutova

Kut se definira kao križanje dva ruba, iako može se definirati i kao točka za koju vrijedi da postoje dva ruba različitog smjera u okolini točke [4, 5].

Detektori kutova se dijele u 3 grupe: detektori temeljeni na predlošku, detektori temeljeni na obrisima i direktni detektori kuteva [4]. Detektori temeljeni na predlošku koriste različite reprezentativne predloške kako bi pronašli poklapanje na slici, koristi se korelacija između predloška i slike kako bi se otkrio kut. Nedostatak ovog pristupa je nemogućnost pokrivanja svih mogućih položaja kutova s reprezentativnim predlošcima. Detektori temeljeni na obrisima prvo pronalaze rubove, a zatim na osnovu obrisa pronalaze kutove. Direktni detektori kutova koriste metode matematičkog izračunavanja, no prije toga je potrebno na sliku primijeniti statističke operacije, nakon toga kutevi se pronalaze na osnovu tih statističkih informacija.

Postoji nekoliko algoritama za pronalaženje kutova, jedan od prvih je Moravec detektor, a poznati su još Harris i Stephens detektor , Susan detektor i Förstner detektor. Moravec detektor definira kut kao točku s niskom samosličnosti [4]. Algoritam ispituje svaki piksel slike kako bi vidio postoji li prisutnost kutova uzimajući u obzir sličnost ulaznih uzoraka. Sličnost se mjeri uzimanjem sume kvadratnih razlika između dva *patches*. Niža vrijednost ukazuje na veću sličnost. Ako je piksel u području jednoličnog intenziteta onda će okolni uzorci izgledati slično. Ako je piksel na rubu, susjedni uzorci u smjeru okomito na rub će izgledati poprilično drugačije ali susjedni uzorci u smjeru paralelno s rubom će rezultirati malim promjenama. Cilj je da suma kvadratnih razlika između uzoraka i njegove okoline bude najmanja moguća. Ako je ta suma lokalno najveća onda je pronađena interesna točka. Problem ovog algoritma je činjenica da ako je kut prezentiran kao da nije u smjeru okoline, najmanja suma kvadratnih razlika će biti velika i rub će biti netočno izabran kao interesna točka [5].

Harrisov detektor kuteva se zasniva na autokorelacijskoj funkciji signala. Osnovna ideja detektora je otkriti pokazuje li točka značajne promjene u svim smjerovima okoline, ako pokazuje znači da je ta točka kut [4].

Slika 2.3. *Usporedba rezultata Susan (lijevo) i Harris (desno) detektora[6]*

2.1.3. Blob detekcija

Blob se definira kao područje svjetlije (ili tamnije) od okoline okružen blago zakrivljenom linijom odnosno područje slike u kojem su određena svojstva konstantna ili približno konstantna [7]. Točke koje se nalaze u blob području se smatraju sličnima. Metode koje koriste blob detekciju ciljaju na detektiranje područja digitalne slike koja se razlikuju u svojstvima kao što su osvjetljenje ili boja, s obzirom na svojstva okoline [8]. Najučestalija metoda za blob detekciju je konvolucija. Interesno područje se izražava kao funkcija položaja i s obzirom na tu funkciju se koriste dva načina pretrage za blob detekciju, diferencijalne metode koje koriste derivacije funkcije s obzirom na poziciju i metode temeljene na lokalnim ekstremima koji pronalaze lokalni minimum ili lokalni maksimum funkcije.

2.1.4. Ridge detekcija

Najčešće korištene značajke u algoritmima računalnog vida su rubovi, no to ne bi trebalo isključivati uporabu ostalih značajki. Blob detektori daju informacije o postojanju objekata, signalizirajući da drugi modul treba pobliže pogledati [8]. *Ridge* značajke se mogu promatrati kao poboljšanja verzija takvog deskriptora, koje dodatno omogućuju otkrivanje približne osi simetrije kandidat objekta [9,10]. Za definiranje ridges iz podataka o intenzitetu, postoji nekoliko mogućih pristupa.

2.1.5. Scale Invariant Feature Transform

Scale Invariant Feature Transform (SIFT) je jedan od najpoznatijih algoritama za pronalaženje i opisivanje ključnih točaka u području računalnog vida. Razvijen je 1999. godine od strane Davida Lowea, a poboljšana verzija objavljena je 2004. Godine [11].

SIFT je invarijantan na skalu i rotaciju, također otporan je na promjene osvjetljenja. Algoritam transformira podatke slike u koordinate invarijantne na promjenu veličine povezane s lokalnim značajkama. Ovaj pristup generira velik broj značajki koji gusto prekriva sliku preko svih razina veličina i lokacija. Slika veličine 500×500 piksela se može prikazati s 2000 stabilnih značajki. Količina značajki je djelomično važna za prepoznavanje objekata, mogućnost za detekciju malog objekata u zatrpanoj pozadini zahtjeva točno podudaranje najmanje 3 značajke.

Za podudaranje i prepoznavanje slika, SIFT značajke su prvo izdvojene iz seta referentnih slika i spremljene u bazu podataka. Nova slika se uspoređuje tako što se uspoređuju individualne značajke slike sa značajkama iz baze podataka i pronalazi se kandidat odgovarajućih značajki te se računa Euklidova udaljenost njihovih vektora [12]. Deskriptor ključnih točaka je vrlo prepoznatljiv što omogućava veliku vjerojatnost da *feature* pronađe svoj odgovarajući par u velikoj bazi podataka.

Kako bi se generirao set značajki SIFT odrađuje sljedeće faze:

1. Scale - space detekcija ekstrema: prva faza je pretraga svih lokacija u svim veličinama, ona se ostvaruje korištenjem *Difference of Gaussian* funkcije kako bi se pronašle potencijalne interesne točke invarijantne na skalu i rotaciju [12].
2. Lokalizacija ključnih točaka: rezultat prve faze algoritma su kandidatne točke, no budući da ih ima puno, neke su nestabilne i podložne šumovima stoga druga faza algoritma odabire najboljih točaka [12].
3. Dodjela orijentacije: jedna ili više orijentacija se dodjeljuje svakoj točki na temelju lokalnog gradijenta smjera slike. Sve buduće operacije se izvode na podacima slike koji su transformirani u odnosu na dodijeljenu orijentaciju, veličinu i lokaciju za svaku značajku, čime se osigurava invarijantnost na transformacije [12].
4. Deskriptor ključnih točaka: lokalni gradijent slike mjeri se na odabranoj skali u području oko svake ključne točke. Lokalni gradijenti se transformiraju u opisni vektor koji omogućava promjene veličine i osvjetljenja [12].

Scale – space detekcija ekstrema se koristi u nekoliko poznatih algoritama poput FREAK-a, SIFT-a, BRISK-a i slično. Cilj ostvarivanja invarijantnosti na skalu je ključan za ostvarivanje ključnih točaka visoke kvalitete, traži se maksimum ne samo u ravnini slike već i u prostoru mjerila koristeći rezultat FAST algoritma kao mjere ispupčenosti. Ključne točke se detektiraju korištenjem kaskadnog filtriranja koje koristi efikasne algoritme za identificiranje lokacije kandidata koji su onda ispitani u detalje [12]. Prvi korak je identificirati lokaciju i veličinu koja

se može ponovljivo dodijeliti pod različitim pogledima na objekt. Detekcija lokacija koje su invarijantne na promjenu veličine slike može se ostvariti traženjem stabilnih značajki kroz sve moguće veličine koristeći kontinuiranu funkciju veličina.

2.2. Deskriptori značajki

Nakon što se detekcijom pronađe određena značajka, primjenjuju se tehnike ekstrakcije kako bi se dobile korisne informacije iz tih značajki. Ekstrakcija značajki opisuje relevantni oblik informacije sadržan u uzorku kako bi proces klasifikacije uzoraka bio jednostavniji [13]. U prepoznavanju uzoraka i obradi slika, ekstrakcija značajki je specijalni oblik redukcije dimenzija. Glavni cilj ekstrakcije je zadržati relevantne informacije iz originalne slike i predstaviti te informacije u manjim dimenzijama .

Ako su ulazni podaci u algoritam preveliki da bi se obradili i ako se pretpostavlja da postoji zalihost tada će se ulazni podaci transformirati u reduciran komplet reprezentativnih značajki.[13] Transformacija ulaznih podataka u komplet značajki se naziva ekstrakcijom značajki. Ako su ekstraktirane značajke pomno odabrane očekuje se da će komplet značajki izvući relevantne informacije iz ulaznih podataka kako bi se ostvario željeni zadatak, koristeći reducirane reprezentativne podatke umjesto onih originalne veličine.

Ekstrakcija značajki je gotova nakon pretprocesorske faze prepoznavanja znakova. Primarni zadatak prepoznavanja uzoraka je uzeti ulazni uzorak i točno ga dodijeliti kao jedan od mogućih izlaza klase.

Ovaj proces se može podijeliti u dva dijela: selekcija značajki i klasifikacija. Selekcija je važna za cijeli proces jer klasifikator neće moći prepoznati ako je selekcija loše odrađena [13]. Lippmanov kriteriji za odabir značajki: značajke trebaju sadržavati informacije koje se razlikuju između klasa, trebaju biti neosjetljive na irelevantne varijabilnosti na ulazu i trebaju biti ograničeni brojem kako bi se osiguralo efikasno računanje diskriminantnih funkcije i ograničila količina podataka .

Izdvajanje značajki je važan korak u konstrukciji klasifikacije svih uzoraka i cilja na ekstrakciju relevantnih informacija koje karakteriziraju svaku klasu. U tom procesu relevantne značajke su ekstraktirane iz objekata u oblik vektora [13]. Takve vektore koriste klasifikatori kako bi dodijelili ulaznoj jedinici izlaznu jedinicu. Klasifikatoru je jednostavnije klasificirati između različiti klasa tako što gleda te značajke kao da dozvoljava da se pomalo razlikuju. Ekstrakcija

značajki je proces dobivanja najvažnijih podataka iz *sirovih podataka*. Ekstrakcija značajki pronalazi komplet parametara koji definiraju oblik simbola precizno i unikatno. U ekstrakciji značajki, svaki simbol je predstavljen s vektorom koji postaje njegov identitet [13]. Glavni cilj ekstrakcije je izvući komplet značajki koji maksimiziraju nivo prepoznavanja s najmanjom količinom elemenata i generiraju sličan komplet za različit primjer istog simbola.

2.3. Difference of Gaussian

Difference of Gaussian (DoG) je funkcija koja oduzima dvije verzije izvorne slike, s tim da je jedna verzija više zamagljena odnosno zamućena (eng. *blurred*), a druga manje. Zamućene slike se dobivaju konvolucijom izvornih slika sivih tonova s Gaussovim *kernelima* koji imaju različita standardna odstupanja σ [14]. S vrijednošću σ se kontrolira razina zamućivanja slika. Oduzimanjem jedne slike od druge čuvaju se prostorne informacije koje se nalaze između raspona frekvencija i sadržane su u dvije mutne slike. DoG je filtar koji odbacuje sve osim prostornih frekvencija koje su prisutne u izvornoj slici sivih tonova [14]. *Difference of Gaussian* slike *f* možemo zapisati na sljedeći način:

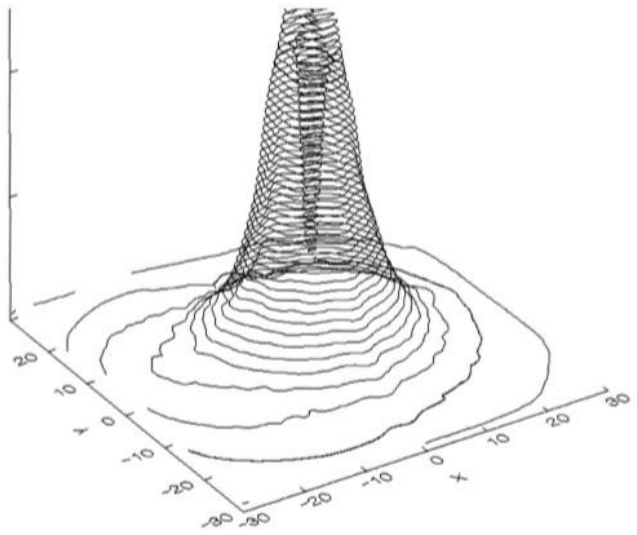
$$f * g(1) - f * g(2) = f * [g(1) - g(2)] \qquad (2-1)$$

3. FAST RETINA KEYPOINT (FREAK)

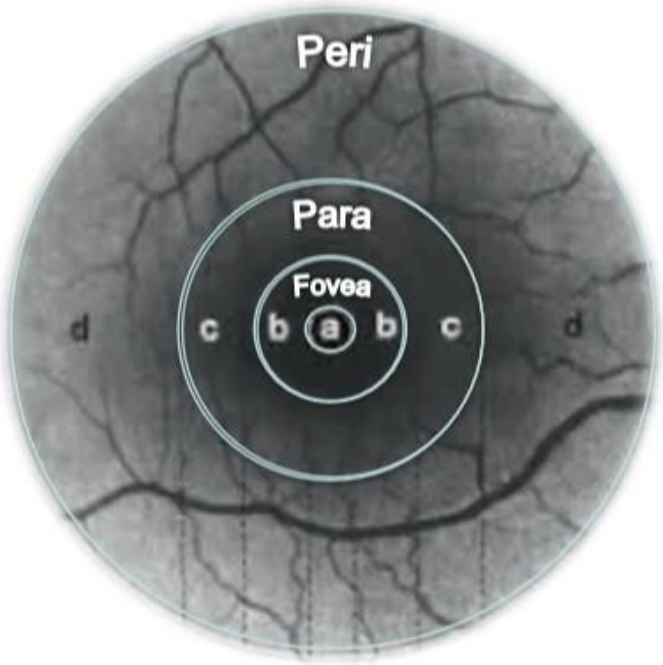
FREAK je jedan od algoritama za prepoznavanje i opisivanje ključnih točaka, algoritam predlaže deskriptor ključnih točaka inspiriran ljudskim vizualnim sustavom [15]. Algoritam određuje ključne točke nekog objekta, te pronalazi te točke na drugoj slici čime se ostvaruje proces povezivanja (eng. *matching*).

3.1. Povezanost ljudskog vizualnog sustava i algoritma

Neuroznanstvenici su napredovali u razumijevanju ljudskog vizualnog sustava te su vrlo dobro objasnili kako se slika prenosi do mozga.[15] Mrežnica oka izvlači detalje iz slika pomoću *Difference of Gaussian* (DoG) te ovaj algoritam predlaže isti princip rada za stvaranje deskriptora. Ganglijske stanice mrežnice oka imaju okrugla receptivna područja koja omogućavaju uočavanje slabih kontrasta i brzih promjena u vidnom prizoru. Područje gdje svjetlost utječe na odgovor ganglijske stanice naziva se receptivno područje. Veličina i osjetljivost receptivnog područja ovisi o radijalnoj udaljenosti od foveole. Prostorna distribucija ganglijskih stanica mijenja se eksponencijalno s promjenom udaljenosti od foveole (slika 2.1.). Područje mrežnice u kojem se nalaze ganglijske stanice se dijeli u četiri područja: foveola, fovea, parafoveal i perifoveal (slika 2.2.) [15]. Svako područje utječe na otkrivanje i prepoznavanje objekata, slika stvorena u fovei ima veću razlučivost od slike stvorene u perifovealnom području.



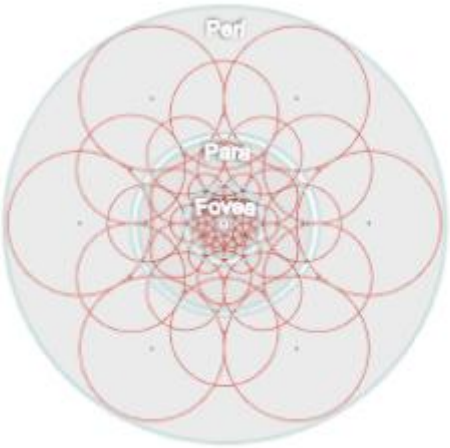
Slika 3.1. *Raspodjela ganglijskih stanica mrežnice oka [15]*



Slika 3.2. *Dijelovi mrežnice [15]*

3.2. Uzorkovanje

FREAK algoritam koristi rešetku za uzorkovanje koja je kružna i ima veliku gustoću uzimanja uzoraka blizu središta kružnice (slika 2.3.) [15]. Ova rešetka je odabrana po uzoru na raspodjelu ganglijskih stanica u mrežnici oka. Gustoća uzoraka opada eksponencijalno jer se i gustoća ganglijskih stanica mijenja na taj način . Svaki uzorak treba biti „izglađen“ s *kernelom* kako ne bi bio osjetljiv na šum. Da bi ovaj koncept odgovarao modelu mrežnice oka koriste se različiti *kerneli* za svaki uzorak budući da slike stvorene u različitim dijelovima mrežnice imaju različitu razlučivost [15]. Primjena različitih *kernela* za svaki uzorak dovodi do boljih performansi postupka. Svaka kružnica predstavlja standardnu devijaciju (odstupanje) Gaussovih raspodjela primijenjenih na odgovarajuće točke uzorkovanja. Budući da se kružnice preklapaju, više informacija je obuhvaćeno što stvara zalihost koja omogućava korištenje manje receptivnih područja. Takva zalihost postoji u receptivnim područjima mrežnice oka.



Slika 3.3*. Ilustracija razdiobe uzoraka [15]*

3.3. Deskriptor

Potrebno je konstruirati binarni deskriptor *F* određivanjem razlike između parova receptivnih područja s odgovarajućim Gaussovim *kernelom*. Deskriptor *F* je binarni niz koji se sastoji od sekvence jednobitne razlike *Difference of Gaussian* (DoG) [15]. *Difference of Gaussian* radi tako da od dvije verzije iste slike no različite razlučivosti, oduzimanjem dobije treću verziju koja pokazuje detalje, a to se u našem slučaju zapisuje nizom nula i jedinica. Deskriptor je prikazan sljedećom formulom:

$$F = \sum_{0 \le a < N} 2^{\alpha} T(P_a) \qquad (3-1)$$

Gdje je P_a par receptivnih područja, *N* željena veličina deskriptora, a *T*

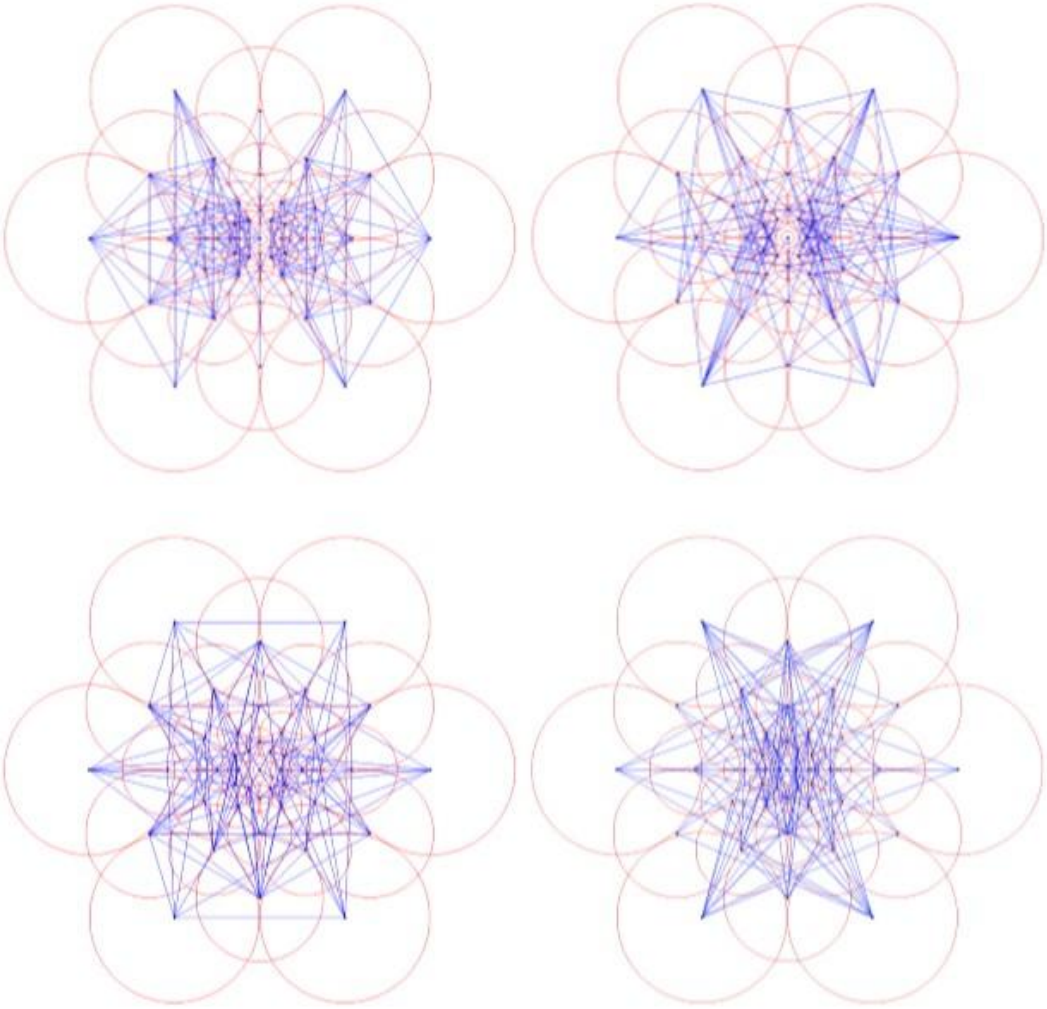
$$T = \begin{cases} 1 \; ako \; [(I(P_a^{r1}) \; - I(P_a^{r2}) \;] > 0 \\ 0 \; za \; sve \; ostalo \end{cases} \qquad (3-2)$$

$I(P_a^{r1})$ je intenzitet prvog receptivnog područja para P_a odnosno intenzitet prve slike na određenim koordinatama dok je a $I(P_a^{r2})$ intenzitet druge slike na istim koordinatama [15]. Ukoliko je razlika oduzimanja ova dva inteziteta veća od nule T poprima vrijednost 1, a za sve ostalo 0, sumiranjem ovih razlika dobijemo binarni zapis.

S nekoliko desetaka receptivnih područja (kružnica) moguće je pronaći tisuće parova za deskriptor no moguće je da neki parovi ne budu korisni kako bi se efikasno opisala slika. Kako

bi se odabrali najbolji parovi uzoraka iz receptivnih područja potrebno je napraviti idući postupak:

1. Kreira se matrica *D* od pedeset tisuća izdvojenih ključnih točaka. Svaki red odgovara ključnoj točki predstavljenoj sa svojim deskriptorom napravljenim od svih mogućih parova iz prikupljenih uzorka [15].
2. Izračunava se srednja vrijednost svakog stupca. Da bi se dobila diskriminantna osobina, željena je velika varijanca. Srednja vrijednost od 0,5 dovodi do najviše varijance binarne distribucije [15].
3. Nadalje, potrebno je poredati stupce u odnosu na najveću varijancu [15].
4. Zadržava se najbolji stupac (prosjek od 0,5) i iterativno se dodaju preostali stupci koji imaju nisku povezanost s odabranim stupcima [15].



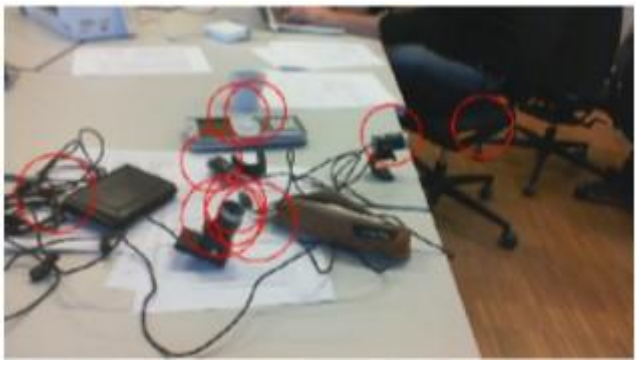
Slika 3.4. *Ilustracija odabranih parova [15]*

3.4. Saccadic pretraga

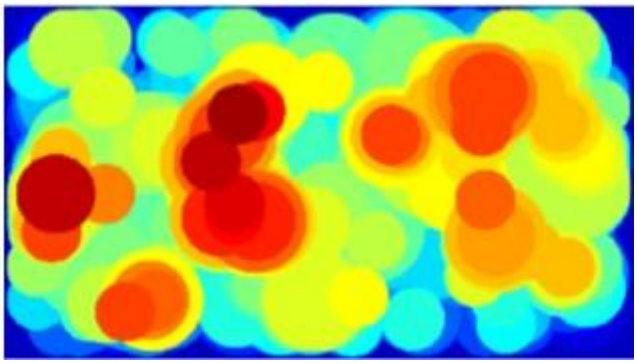
Ljudske oči ne mogu nepomično gledati u jednu točku, one se kreću s diskontinuiranim individualnim pokretima koji se nazivaju *saccades* (brz pokret očima između dvije fiksirane točke) [15]. Topologija ćelija mrežnice je razlog ovim pokretima, fovea oka bilježi informacije visoke rezolucije zahvaljujući svojim receptorima velike gustoće stoga je vrlo važna u procesu prepoznavanja objekata upravo zato se pokušava imitirati prirodni *saccades* oka rastavljanjem deskriptora u nekoliko koraka. Započinje se pretraga s prvih 16 bajtova kojima se opisuje FREAK deskriptor i predstavljaju grube informacije. Ako je udaljenost manja od određenog praga, nastavlja se usporedba sa sljedećim bajtovima koji trebaju omogućiti finiju analizu. Kao rezultat toga, vrši se kaskadna usporedba koja ubrzava korake podudaranja .

Na sljedećim slikama ćemo prikazati korake *saccadic* pretrage. Prvo je potrebno opisati objekt koji želimo tražiti, FREAK deskriptorom, što se vidi na slici 3.3. Nadalje, taj objekt se traži na idućoj slici (slika 3.4.), gdje se nalazi više sličnih objekata pri čemu se objekti na novoj slici također opisuju FREAK deskriptorom. Prva kaskada pretraživanja isključuje određen broj objekata odnosno pronalazi se malo objekata s velikim podudaranjem, što se može vidjeti na slici 3.5., samo nekoliko objekata je označeno crvenom bojom što znači veći stupanj podudaranja. Slika 3.6. prikazuje udaljenost sličnih objekata. Na zadnjoj slici (slika 3.7.) može se vidjeti da je pretraga dala zadovoljavajuće rezultate odnosno pronašla naš objekt iako se on nalazi u različitim pozicijama (na slici na kojoj se tražio naš objekt, postoji više takvih objekata u različitim pozama) .

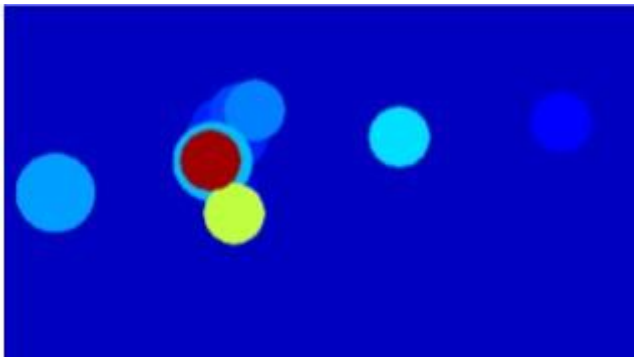
Slika 3.5. *objekt koji želimo tražiti[15]*



***Slika 3.6**. najviše slični objekti nakon prve kaskade[15]*



***Slika 3.7**. prikaz udaljenosti (u boji) nakon prve kaskade[15]*

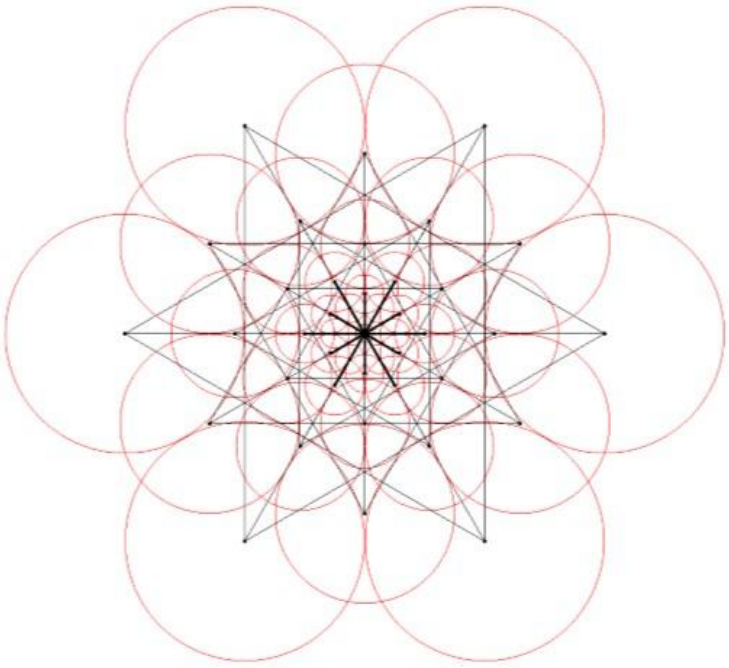


***Slika 3.8**. prikaz udaljenosti (u boji) nakon zadnje kaskade[15]*

***Slika 3.9**. objekti koji se podudaraju, nakon zadnje kaskade[15]*

3.5. Orijentacija

Kako bi se procijenila rotacija ključne točke, zbrajaju se procijenjeni lokalni gradijenti nad odabranim parovima [15]. Koriste se dugački parovi kako bi se izračunala globalna orijentacija. Parove koje smo odabrali imaju simetrična receptivna područja u odnosnu na središte što prikazuje sljedeća ilustracija.



Slika 3.10. *Ilustracija odabranih parova za izračunavanje orijentacije [15]*

$$O = \frac{1}{M} \sum_{P_o \in G} \left(I(P_o^{r1}) - I(P_o^{r2}) \right) \frac{P_o^{r1} - P_o^{r2}}{\| P_o^{r1} - P_o^{r2} \|} \qquad (2\text{-}3)$$

Formula za računanje orijentacije: neka je *G* skup svih parova korištenih za računanje lokalnog gradijenta. Neka je *M* broj parova pri čemu parovi pripadaju *G*, P_o^{ri} predstavlja 2D vektor prostornih koordinata središta receptivnog područja [15].

4. EKSPERIMENTALNI DIO

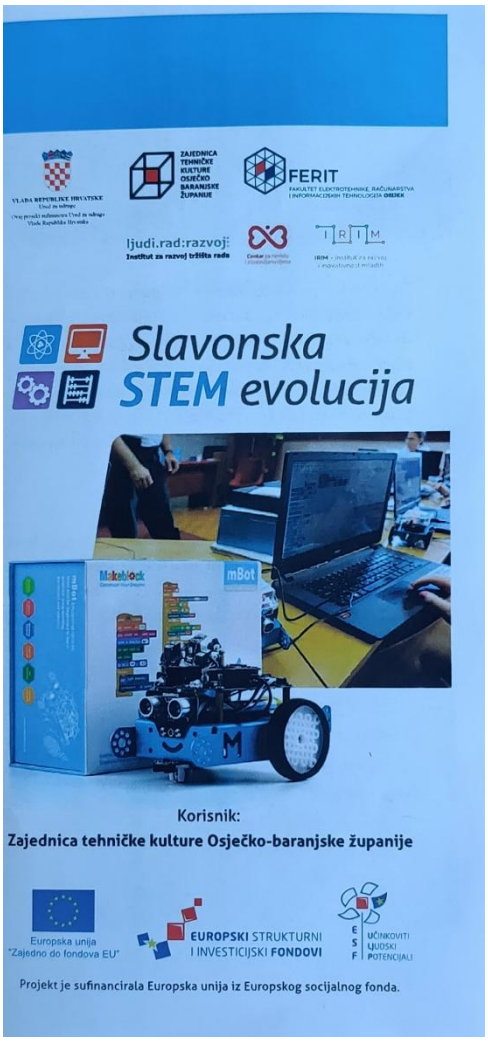
4.1. OpenCV

OpenCV (*Open Source Computer Vision*) je skup biblioteka otvorenog koda namijenjenih radu s računalnim vidom (eng. *computer vision*) te strojnom učenju [16]. Poseban naglasak je na aplikacijama koje rade s podacima u stvarnom vremenu (eng. *real-time application*). Glavni cilj OpenCV-a je pružanje jednostavne infrastrukture za rad s računalnim vidom te omogućuje jednostavan i brz razvoj jednostavnih ali i složenih aplikacija. Veliki broj operacija i kompliciranih algoritama nad podacima u stvarnom vremenu zahtijeva optimiziran kod pisan na nižoj razini, upravo zato je OpenCV izvorno pisan u programskom jeziku C/C++. Iako je pisan u C jeziku, razvijena su sučelja prema drugim jezicima više razine odnosno sadrži sučelja za rad u programskim okruženjima Java, Python, C++ i MATLAB, također podržana je na gotovo svim operacijskim sustavima.

OpenCV sadrži više od 2500 optimiziranih algoritama koji uključuju opsežan skup klasičnih i najsuvremenijih algoritama za rad s računalnim vidom. Ti algoritmi se mogu koristiti za otkrivanje i prepoznavanje lica, prepoznavanje objekata, praćenje pokretnih objekata, pronalaženje sličnih slika u bazama podataka, itd.

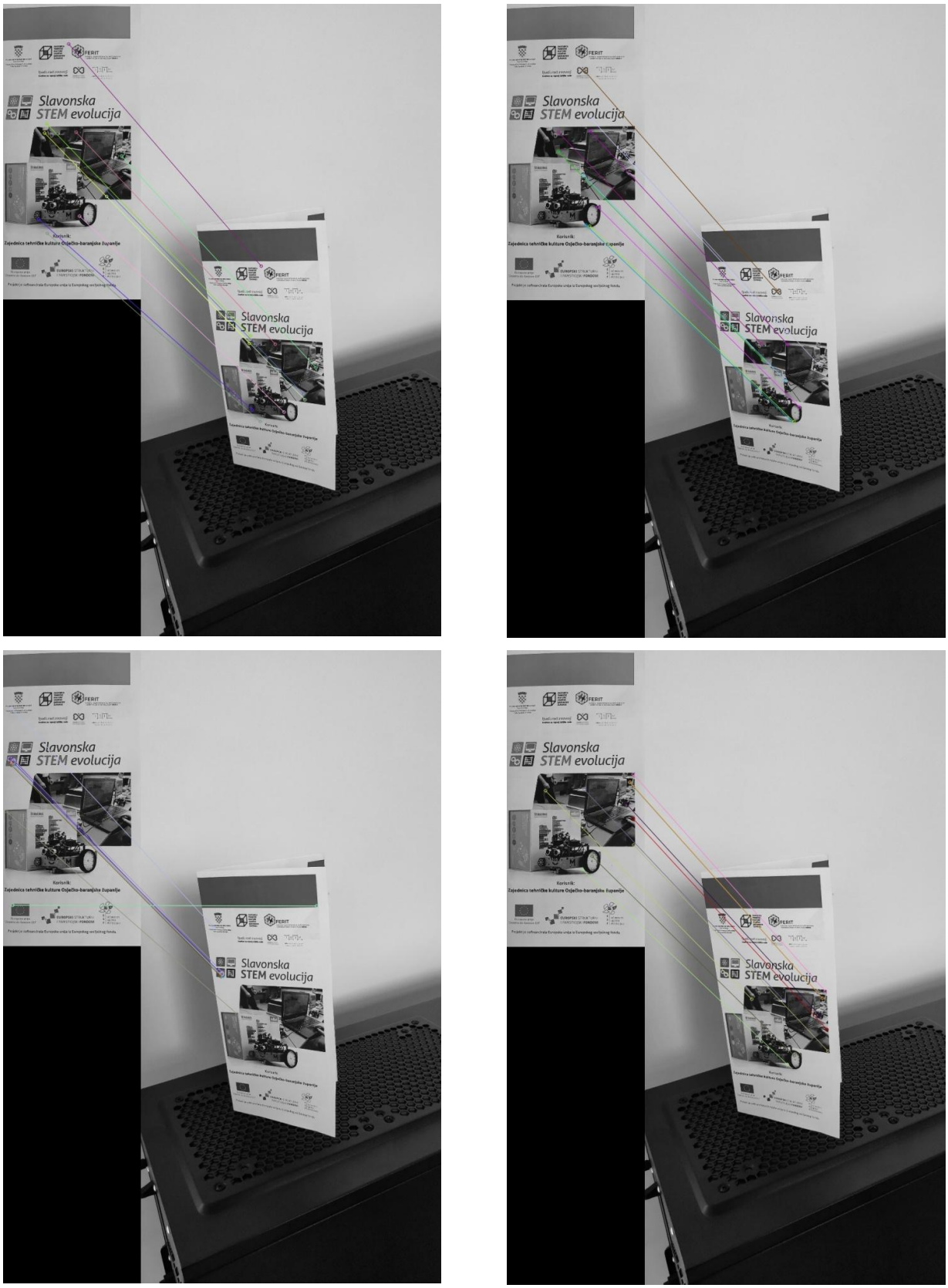
4.2. Eksperiment

Eksperiment se provodi tako da se definira jedna slika koja predstavlja uzorak koji se traži na svakoj od ostalih slika. Implementiralo se četiri algoritma FREAK, BRISK, SIFT i SURF, te će eksperiment prikazati situacije u kojima je FREAK bolji i one situacije u kojima je lošiji te je bolje koristiti druge algoritme. Svaki algoritam pronalazi 10 značajki koje povezuje sa slikom, ispravno povezivanje se definira kao povezivanje u kojem se ostvarilo da se točke s objekta pronađu na slici unutar objekta, ako povezane točke nisu ispravne ali su unutar traženog objekta, takvo povezivanje smatramo ispravnim jer tražimo objekt, dakle neispravno povezivanje je samo ono gdje je značajka objekta povezana s točkom koja nije unutar objekta. Eksperiment je napravljen na ukupno dvanaest te su sve uključene u analizu, no u radu je priloženo šest slika.



Slika 4.1. *Uzorak (letak)*

Slika 4.2. *Prva slika na kojoj tražimo odabrani uzorak*



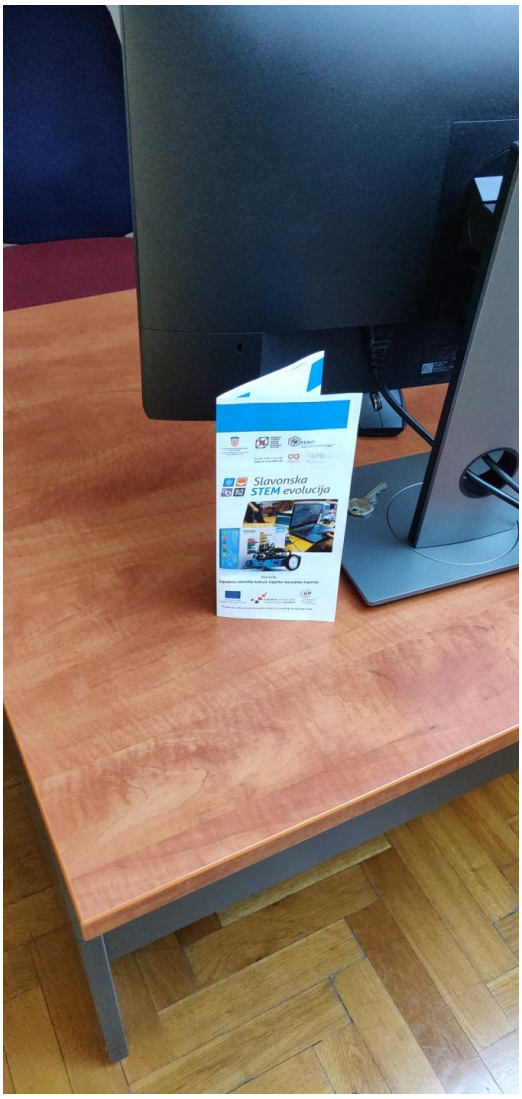
Slika 4.3. *Rezultati na slici: gore lijevo FREAK, gore desno BRISK, dolje lijevo SIFT, dolje desno SURF*

Na slici 4.3. vidimo rezultate povezivanja točka. FREAK algoritam je izabrao 10 značajki koje je potrebno povezati sa slike uzorka na slici na kojoj tražimo objekt. Slika na kojoj tražimo objekt nije previše kompliciranа i natrpana, vidimo čistu pozadinu i podlogu zanimljive teksture, no to nije omelo algoritam u točnom povezivanju. Sva povezivanja se nalaze unutar traženog objekta i točna su. Rezultat BRISK algoritma je također izvrstan, sve značajke su točno povezane, razlika je samo u izboru značajki. SIFT algoritam griješi na dva mjesta, oba mjesta su kutovi. Pogrešno je spojena točka koja je pronađena kod zastave EU te je spojena s plavim pravokutnikom letka, a druga pogrešno spojena točka je na kutiji koja je nacrtana na letku iako su navedene točke pogrešno spojene ne smatramo ih greškama jer su i dalje unutar našeg objekta kojeg tražimo odnosno unutar definicije točnog povezivanja. SURF pokazuje također izvrsne rezultate, sve odabrane značajke su točno povezane.

Slika 4.4. *Druga slika na kojoj tražimo uzorak*

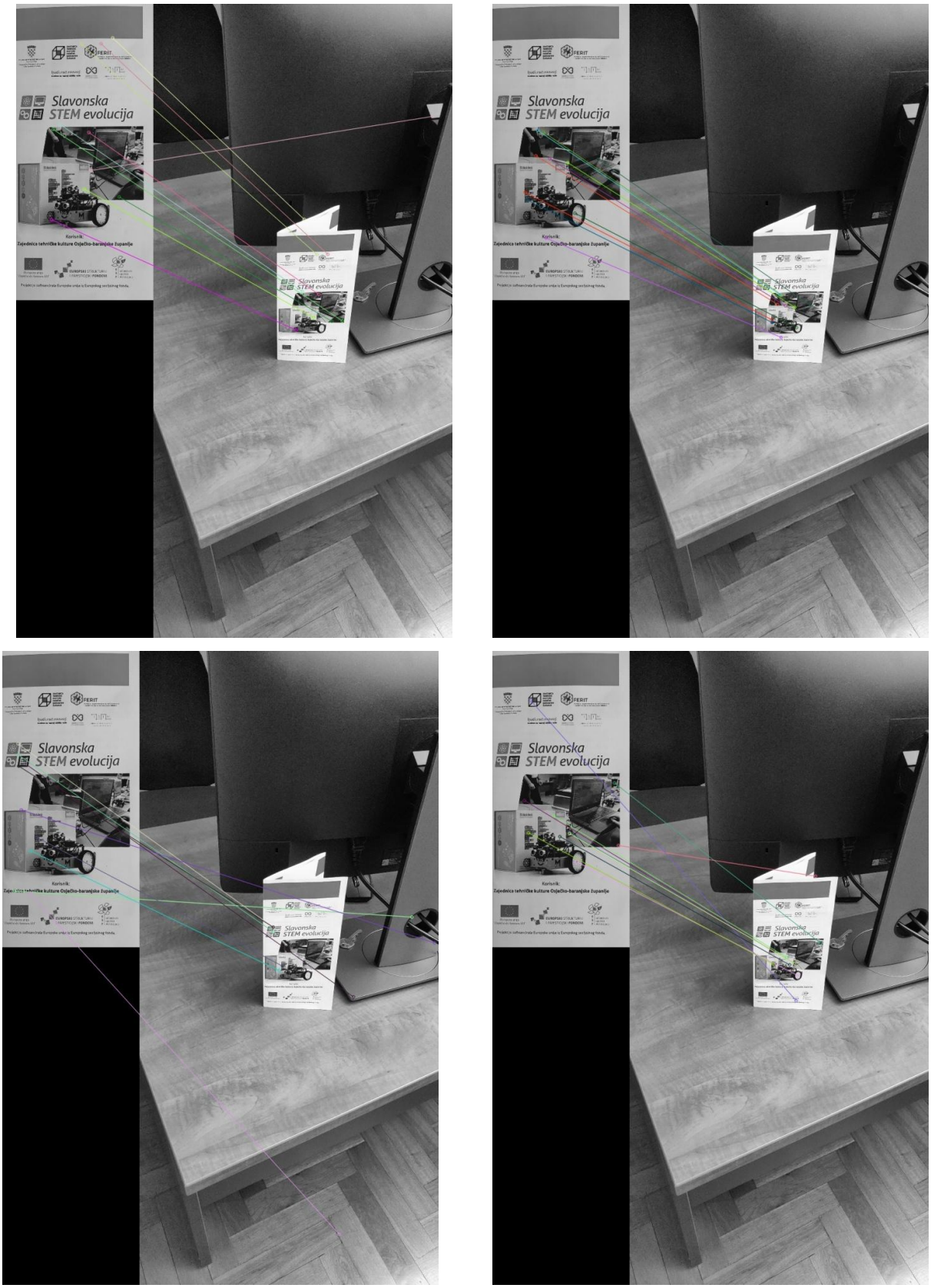
***Slika 4.5**. Rezultati na slici: gore lijevo FREAK, gore desno BRISK, dolje lijevo SIFT, dolje desno SURF*

Na slici 4.5 FREAK algoritam ponovno točno povezuje sve značajke. BRISK također sve značajke točno povezuje. SIFT je pogriješio sa spajanjem jedne točke no i dalje je ostao unutar objekta. Algoritam SURF sve značajke točno povezuje.



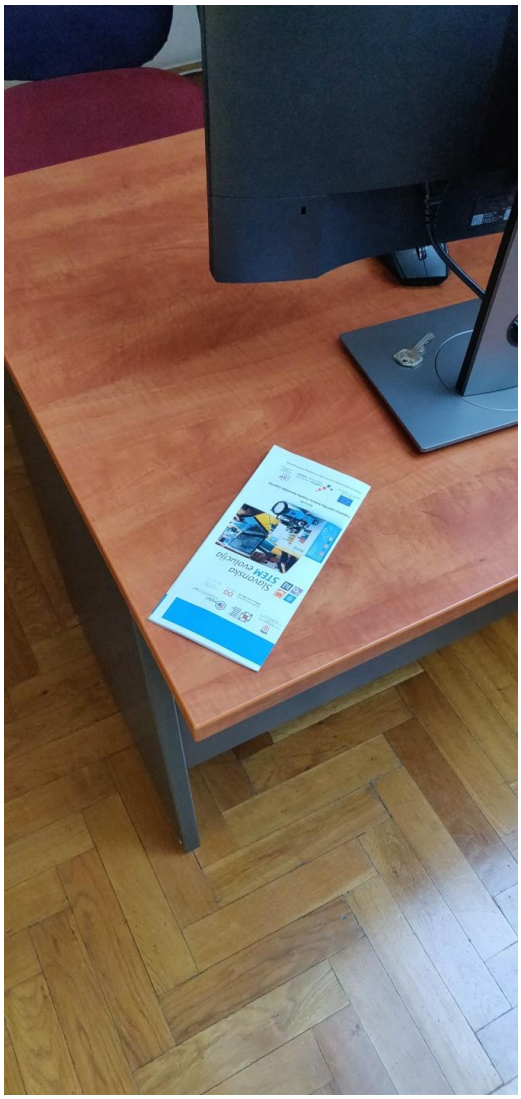
Slika 4.6. *Treća slika na kojoj tražimo odabrani uzorak*

Slika 4.6. za razliku od prethodne dvije slike, ima više stvari osim uzorka kojeg algoritam traži što automatski ukazuje na mogućnost pogreške.



***Slika 4.7**. Rezultati na slici: gore lijevo FREAK, gore desno BRISK, dolje lijevo SIFT, dolje desno SURF*

Na slici 4.7. se vidi da je FREAK pogriješio, jedna od deset mogućih značajki je krivo povezana. Značajka pronađena na letku je spojena s rubom monitora. BRISK algoritam nije zbunjen zbog dodatnih predmeta u pozadini i ponovno je sve značajke točno povezao. SIFT algoritam pokazuje lošije rezultate naspram ostalih i naspram prve dvije slike, s obzirom na prve dvije slike možemo reći da je u ovom slučaju zbunjen pozadinom. SIFT je četiri značajke spojio krivo. SURF pokazuje zadovoljavajuće rezultate, ima samo jednu grešku.

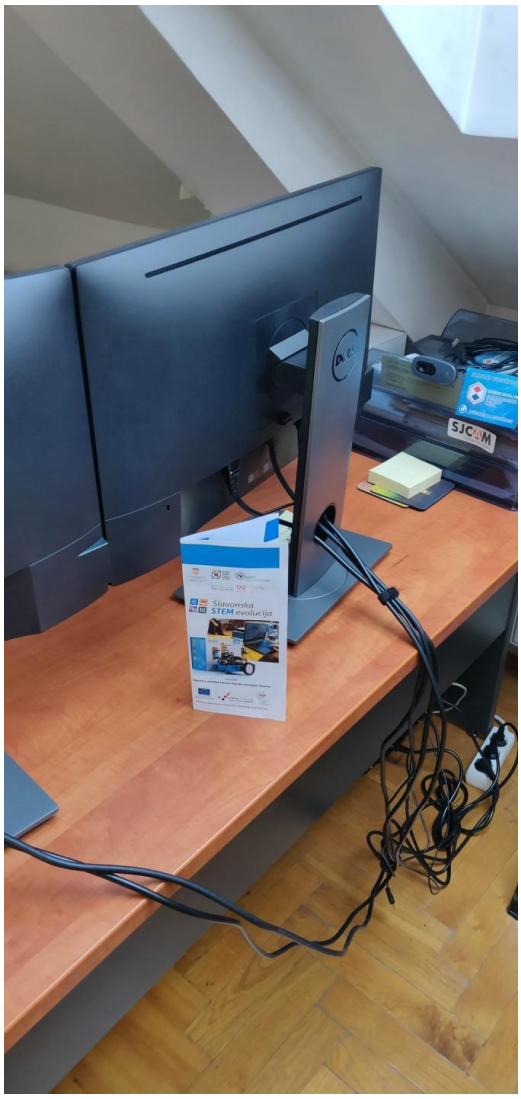


Slika 4.8. *Četvrta slika na kojoj se traži odabrani uzorak*

Za razliku od prve tri slike, na četvrtoj se može vidjeti da je traženi objekt podosta zarotiran, što je također jedan od kriterija pod kojim algoritam mora ispravno raditi (invarijantnost na rotaciju). Pozadina je vrlo slična prethodnoj.

***Slika 4.9**. Rezultati na slici: gore lijevo FREAK, gore desno BRISK, dolje lijevo SIFT, dolje desno SURF*

Na slici 4.9. FREAK pokazuje tri neispravna podudaranja. BRISK prikazuje ispravno povezivanje svi značajki. SIFT je ispravno povezao sve odabrane značajke. SURF je krivo povezao samo jednu značajku.

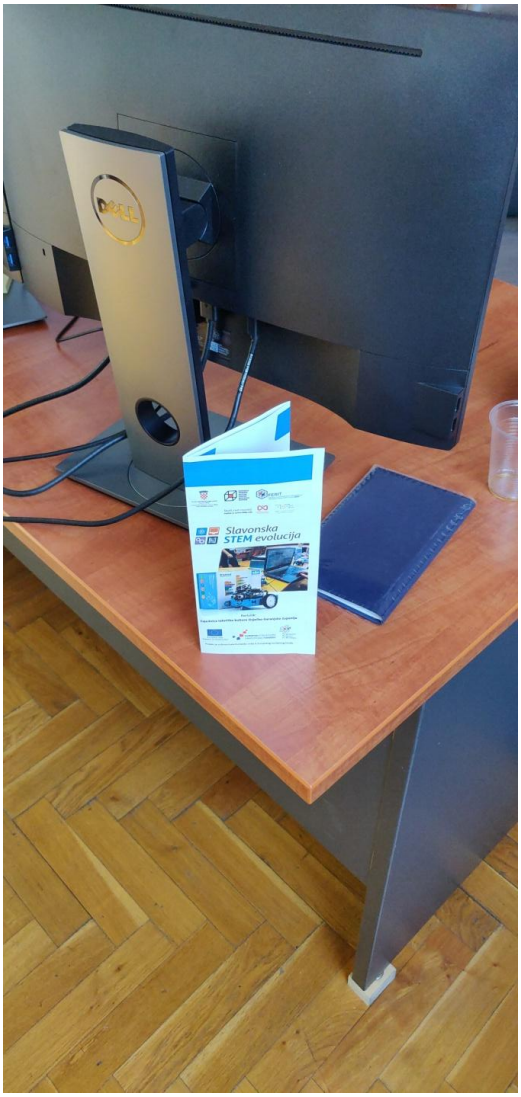


Slika 4.10. *Peta slika na kojoj se traži odabrani uzorak*

Slika 4.10. ima najviše različitih stvari koje mogu zbuniti algoritme odnosno izborom raznovrsnih stvari u okolini traženog predmeta povećava se mogućnost pogreške.

***Slika 4.11**. Rezultati na slici: gore lijevo FREAK, gore desno BRISK, dolje lijevo SIFT, dolje desno SURF*

Na slici 4.11. FREAK pokazuje prilično loše rezultate, od deset mogućih, pet značajki je krivo povezao. Mogućnost pogreške koju smo pretpostavili na početku provjere ove slike je prisutna. Rezultati BRISK algoritma pokazuju da su sve značajke točno povezane. SIFT algoritam je dvije značajke krivo povezao. SURF je jednu značajku krivo povezao.



***Slika 4.12**. Šesta slika na kojoj tražimo odabrani uzorak.*

***Slika 4.13**. Rezultati na slici: gore lijevo FREAK, gore desno BRISK, dolje lijevo SIFT, dolje desno SURF*

FREAK pokazuje iznimno loše rezultate, šest značajki je krivo povezao. BRISK pokazuje lošije rezultate nego u prethodnim slučajevima, četiri značajke su krivo povezane. Od deset mogućih značajki, SIFT je četiri značajke krivo povezao, također pojavljuje se precrtavanje linija koje povezuju značajke pa su linije teško uočljive. SURF jednu značajku krivo povezuje.

4.3. Analiza

Slike su fotografirane u različitim uvjetima, pojavljivale su se slike s malo predmeta, s više predmeta i slike gdje je traženi objekt okrenut. Eksperiment je pokazao da svi algoritmi griješe ali isto tako da mogu pokazati izvrsne rezultate. Sljedeća tablica prikazuje broj točno detektiranih značajki, pri čemi prvi broj označava broj točno detektiranih, a drugi broj je ukupan broj značajki, te je na kraju izračunat postotak točno detektiranih značajki. Prvih šest slika u tablici predstavlja slike prikazane u radu.

	FREAK	BRISK	SIFT	SURF
1. Slika	10/10	10/10	10/10	10/10
2. Slika	10/10	10/10	10/10	10/10
3. Slika	9/10	10/10	6/10	9/10
4. Slika	7/10	10/10	10/10	9/10
5. Slika	5/10	10/10	8/10	9/10
6. Slika	4/10	6/10	6/10	9/10
7. Slika	8/10	10/10	10/10	10/10
8. Slika	5/10	9/10	10/10	9/10
9. Slika	6/10	9/10	3/10	4/10
10. Slika	10/10	10/10	10/10	10/10
11. Slika	9/10	10/10	10/10	9/10
12. Slika	8/10	10/10	8/10	9/10
Ukupan broj točno detektiranih značajki	91/120	114/120	101/120	107/120
Postotak	75,83%	95%	84,16%	89,16%

Tablica 4.1. *Rezultati točno detektiranih značajki*

5. ZAKLJUČAK

Iako ljudi bez poteškoća prepoznaju određene predmete koje žele pronaći, računala još uvijek uče kako to postići te se to područje istraživanja naziva računalni vid. Jedan od algoritama koji se bavi navedenom tematikom je algoritam FREAK predstavljen u ovom radu. Razvoj ovoga te sličnih algoritama je od iznimne važnosti za unaprjeđenje računalnog vida

Teorijskim pregledom su objašnjene metode najznačajnijih detekcija te načini rada deskriptora koji se koriste u računalnom vidu. Detekcija kutova i rubova ključna je za sve ostale algoritme. Opisan je princip rada FREAK algoritma te njegova teorijska i matematička pozadina.

Eksperiment se temeljio na usporedbi s drugim sličnim algoritmima te pokazao da FREAK nije konstantan odnosno ne daje stalno očekivane rezultate. FREAK dobro radi u nezahtjevnim uvjetima poput slika s jednostavnom okolinom i dobrom preglednošću na objekt koji se traži. Kao i područje istraživanja, ovaj algoritam je relativno mlad te su potrebna dodatna unaprjeđenja istog kako bi se poboljšala precizna izvedivost.

6. LITERATURA

[1] R.C.Gonzales; R.E.Woods; Digital Image Processing; Prentice Hall; New Yersey; dostupno na: http://web.ipac.caltech.edu/staff/fmasci/home/astro_refs/Digital_Image_Processing_2ndEd.pdf; [zadnji pristup: 14.9.2018.]

[2] Edge detection; dostupno na https://en.wikipedia.org/wiki/Edge_detection [zdanji pristup: 14.9.2018.]

[3] N.Tsankashvili; Comparing Edge Detection Methods; dostupno na: https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e; [zadnji pristup: 14.9.2018.]

[4] T.P. Patel; S.R. Panchal; Corner Detection Techniques: An Introductory Survey; IJERD; dostupno na: https://www.ijedr.org/papers/IJEDR1404047.pdf ; [zadnji pristup 14.9.2018.]

[5] Corner Detection; dostupno na: https://en.wikipedia.org/wiki/Corner_detection; [zadnji pristup 14.9.2018.]

[6] J.Chen; L. Zou; J. Zhang; L. Dou; The Comparision and Application of Corner Detection Algorithms; Journal of Multimedia; prosinac 2009; dostupno na: https://pdfs.semanticscholar.org/2369/7150910fb796b6e2a5b5e786e96622098d86.pdf; [zadnji pristup 14.9.2018]

[7] A. Danker; A.Rosenfeld; Blob detection by relaxation; University of Maryland, College Park; srpanj 1979; dostupno na: http://www.dtic.mil/dtic/tr/fulltext/u2/a086190.pdf ; [zadnji pristup 14.9.2018.]

[8] Blob detection; dostupno na: https://en.wikipedia.org/wiki/Blob_detection; [zadnji pristup 14.9.2018.]

[9] Ridge detection; dostupno na: https://en.wikipedia.org/wiki/Ridge_detection [zadnji pristup 14.9.2018.]

[10] T. Lindberg; Edge detection and ridge detection with automatic scale selection; Royal Institute of Technology; Stockholm,Sweden; dostupno na: http://www.diva-portal.org/smash/get/diva2:452310/FULLTEXT01.pdf ; [zadnji pristup 14.9.2018.]

[11] Scale-invariant feature transform; dostupno na: https://en.wikipedia.org/wiki/Scale-invariant_feature_transform ; [zadnji pristup 14.9.2018.]

[12] D.G. Lowe; Distinctive Image Features from Scale-Invariant Keypoints; University of British Columbia; Vancouver, Canada; 5. Siječnja 2004.

[13] G. Kumar; P.K. Bhatia; A Detailed Review of Feature Extraction in Image Processing Systems

[14] Difference of Gaussian; dostupno na https://en.wikipedia.org/wiki/Difference_of_Gaussians [zadnji pristup 14.9.2018.]

[15] A. Alahi, R. Ortiz; P. Vandergheynst; FREAK: Fast Retina Keypoint; Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

[16] OpenCV; dostupno na https://opencv.org/about.html [zadnji pristup 14.9.2018.]

7. SAŽETAK

Rad se bavi područjem računalnog vida, odnosno algoritmom koji pripada tom području. Navedeno područje istražuje digitalne slike odnosno bavi se razumijevanjem slika u svrhu pronalaska značajki po kojima je moguće prepoznati i povezati dvije slike.

Kako bi se pojednostavila analiza slike iz iste se izdvajaju bitne značajke koje će koristiti u daljnjoj obradi. Algoritam FREAK se koristi upravo za to određivanje ključnih točaka te povezivanje istih, inspiriran je ljudskim vizualnim sustavom što pretpostavlja bolje rezultate izvođenja algoritma. Cilj ovog rada je pojasniti FREAK, izložiti prednosti i nedostatke te izvesti eksperiment koji treba potkrijepiti navedeno.

Ključne riječi: računalni vid, značajke, analiza, ljudski vizualni sustav

8. ABSTRACT

The paper deals with the field of computer vision, that is, the algorithm that belongs to this field. This area explores digital images and deals with the understanding of the image in order to discover features that make it possible to recognize and link two images. Understanding the picture can be defined as extracting useful information from the image, needed for further analysis.

In order to simplify the image analysis, the essential features that will be used in further processing are extracted. One of the tasks involved in computing is to detect and connect these features. The FREAK algorithm is used to precisely define and correlate key points, inspired by the human visual system, which presupposes better performance of the algorithm. The aim of this paper is to clarify FREAK, expose advantages and disadvantages, and carry out the experiment that needs to be supported.

Keywords: computer vision, features, analysis, human visual system

9. ŽIVOTOPIS

Barbara Strišković je rođena 31. prosinca 1996. godine u Osijeku. Odrasla je u Donjem Miholjcu gdje je pohađala Osnovnu školu Augusta Harambašića. Nakon završetka osnovne škole upisuje Srednju školu Donji Miholjac, smjer Opća gimnazija. Nakon srednje škole upisuje preddiplomski studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, te na drugoj godini odabire smjer Komunikacije i informatika.

_____

Barbara Strišković