Detekcija ključnih točaka slike invarijantna na skalu

Viljušić, Veronika

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:602489

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-27**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij elektrotehnike

DETEKCIJA KLJUČNIH TOČAKA SLIKE INVARIJANTNA NA SKALU

Završni rad

Veronika Viljušić

Osijek, 2018.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 23.09.2018.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Veronika Viljušić
Studij, smjer:	Preddiplomski sveučilišni studij Elektrotehnika i informacijska tehnologija
Mat. br. studenta, godina upisa:	4134, 28.09.2017.
OIB studenta:	78619746402
Mentor:	Izv. prof. dr. sc. Irena Galić
Sumentor:	Hrvoje Leventić
Sumentor iz tvrtke:	
Naslov završnog rada:	Detekcija ključnih točaka slike invarijantna na skalu
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	23.09.2018.
Datum potvrde ocjene Odbora:	26.09.2018.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 28.09.2018.

Ime i prezime studenta:	Veronika Viljušić
Studij:	Preddiplomski sveučilišni studij Elektrotehnika i informacijska tehnologija
Mat. br. studenta, godina upisa:	4134, 28.09.2017.
Ephorus podudaranje [%]:	2%

Ovom izjavom izjavljujem da je rad pod nazivom**: Detekcija ključnih točaka slike invarijantna na skalu**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Irena Galić

i sumentora Hrvoje Leventić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1
1.1. Zadatak rada 1
2. OPĆENITO O ZNAČAJKAMA 2
2.1. Detekcija značajki 2
2.1.1. Detekcija rubova 2
2.1.2. Detekcija kutova 3
2.1.3. Blob detekcija 4
2.1.4. Ridge detekcija 4
2.2. Ekstrakcija značajki 4
2.3. Scale Invariant Feature Transform (SIFT) 5
3. BINARY ROBUST INVARIANT SCALABLE KEYPOINTS (BRISK) 8
3.1. Scale – space detekcija ključnih točaka 8
3.2. Opis ključnih točaka 9
3.2.1. Uzorkovanje i procjena rotacije 10
3.2.2. Deskriptor 11
3.2.3. Podudaranje 12
4. EKSPERIMENTALNI DIO 13
4.1. OpenCV 13
4.2. Eksperimenti 13
4.3. Analiza rezultata 23
5. ZAKLJUČAK 25
6. LITERATURA 26
7. SAŽETAK 27
8. ABSTRACT 28
9. ŽIVOTOPIS 29

1. UVOD

Kao znanstvena disciplina, računalni vid se bavi teorijom i tehnologijom za izgradnju sustava koji izdvajaju informacije iz slike ili višedimenzionalnih podataka [1]. Područje je u računarstvu koje se bavi omogućavanjem računalima da "vide", identificiraju i obrađuju sliku. Neke od oblasti koje računalni vid obuhvaća su rekonstrukcija scene, praćenje videa, prepoznavanje objekata, procjena 3D položaja itd.

Široko primijenjena tehnika u računalnom vidu je rastavljanje slike u lokalne oblasti odnosno značajke koje se koriste za ublažavanje složenosti slike, ali i za istodobno iskorištavanje svojstava izgleda pojedinih oblasti. Prikaz slike, prepoznavanje objekta na slici i podudaranje, rekonstrukcija 3D prizora i praćenje pokreta se oslanjaju na prisutnost karakterističnih i ustaljenih značajki na slici te tako postoje različiti načini pristupa tom problemu. Idealna detekcija ključnih točaka slike otkriva istaknute značajke slike te se one mogu ponovno otkriti usprkos promjeni gledišta, odnosno jasna je i otporna na sve moguće transformacije slike. Također, idealni deskriptor bilježi najvažniji i karakteristični sadržaj koji se nalazi u istaknutim značajkama slike na način da određeni objekt može biti prepoznat i kada je dijelom prekriven.

Metoda za detekciju ključnih točaka slike korištena u ovom radu je *Binary Robust Invariant Scalable Keypoints (BRISK).* Navedeni algoritam omogućava brzu i kvalitetnu detekciju ključnih točaka slike te je u značajnoj mjeri invarijatan na skalu i rotaciju.

1.1. Zadatak rada

Zadatak rada je pokazati i objasniti način rada algoritma *BRISK* koji pripada bibliotekama OpenCV-a. Treba pokazati kako radi te priložiti i objasniti rezultate dobivene eksperimentima. Ukazati na prednosti i nedostatke ovog algoritma.

2. OPĆENITO O ZNAČAJKAMA

U području obrade slike jako važnu ulogu imaju njena svojstva, odnosno značajke (engl. *feature*). Razlikujemo detekciju i ekstrakciju značajki. Detekcija podrazumijeva pronalaženje ključnih točaka, dok ekstrakcija opisuje točke i izdvaja one najvažnije informacije koje će biti korisne za daljnje usporedbe i obradu slike.

2.1. Detekcija značajki

Detekcija značajki u području obrade slika predstavlja operaciju niže razine budući da se obično izvodi kao prva operacija na slici na način da ispituje svaki piksel slike kako bi se vidjelo postoji li na tom pikselu značajka [2]. Ako je to jedan od koraka nekog algoritma, onda će algoritam ispitivati sliku samo u području značajki. Ponekad se algoritam koristi za pretraživanje samo određenih dijelova slike, npr. ako je pretraživanje skupo ili postoje vremenska ograničenja. Prilikom izvođenja mnogih algoritama računalnog vida, početni korak je upravo detekcija značajki.

2.1.1. Detekcija rubova

Detekcija rubova obuhvaća različite matematičke metode te je cilj prepoznavanje onih točaka na digitalnoj slici u kojima se svjetlina značajno mijenja, odnosno nema kontinuitet [3]. Točke u kojima se svjetlina slike naglo mijenja obično se nalaze kao skup segmenata koji čine zakrivljenu liniju odnosno rubove. Isti problem pronalaženja diskontinuiteta u jednodimenzionalnim signalima je poznat kao *step detection* te je problem pronalaženja prekida signala tijekom vremena poznat kao detekcija promjene. Detekcija rubova temeljni je alat u obradi slika, strojnom vidu i računalnom vidu, posebno u području detekcije i ekstrakcije značajki.

Rezultat primjene detektora rubova na sliku u idealnom slučaju je skup povezanih krivulja koje označavaju granice objekata, granice površinskih oznaka, kao i krivulje koje odgovaraju diskontinuitetima u orijentaciji površine. Dakle, primjenom algoritma za detekciju rubova može se značajno smanjiti količina podataka koja se obrađuje te se stoga mogu filtrirati informacije koje se smatraju manje bitnima, a da se pritom zadržavaju važna strukturna svojstva slike [3]. Ako je prilikom obrade slike korak detekcije rubova uspješan, sljedeći zadatak koji se vrši radi dobivanja informacija iz izvorne slike može biti znatno pojednostavljen. Međutim, takve idealne rubove iz izvornih slike nije uvijek moguće dobiti.

Na netrivijalnim slikama rubovi su često pogrešno detektirani, odnosno rubne krivulje nisu povezane jer nedostaju rubni segmenti. Također, pojavljuju se lažni rubovi koji ne odgovaraju pojavama na slici, te se time sljedećem zadatku komplicira tumačenje podataka slike.

Detekcija rubova jedan je od temeljnih koraka u raznim tehnikama računalnog vida, obradi slike, analize slike te uzorkovanju slike. Neki od poznatih algoritama su Sobel i Canny.

2.1.2. Detekcija kutova

Detekcija kutova je metoda koja se koristi u računalnim sustavima za izdvajanje određenih vrsta značajki te da bi se mogao odrediti sadržaj slike [4]. Koristi se u otkrivanju pokreta, praćenju videozapisa, panoramama, 3D modeliranju i prepoznavanju objekata.

Kut se može definirati kao sjecište dvaju rubova te kao točka za koju vrijedi da u njenoj okolini postoje dva rubna pravca različitih smjerova [4]. Točka interesa je ona točka na slici koja ima dobro definiran položaj te se može jednoznačno odrediti. To znači da ta točka može biti kut, ali može biti i izolirana točka maksimalnog ili minimalnog lokalnog intenziteta, završetak linije ili točka na krivulji gdje je zakrivljenost lokalno maksimalna. Kao posljedica toga, ako treba otkriti samo kutove, potrebno je napraviti lokalnu analizu detektiranih točaka kako bi se utvrdilo koje su od njih pravi kutovi.

Jedna od stavki koja određuje kvalitetu detektora kuta je njegova sposobnost otkrivanja istog kuta na više sličnih slika koje se razlikuju u svjetlini, rotaciji kuta i sl. Neki od poznatih detektora kutova su Harris detektor [4], SUSAN [5] i Moravec algoritam [5], i dr.

Moravec algoritam je jedan od prvih algoritama za detekciju kutova i definira kut kao točku koja ima malu samosličnost [5]. Algoritam ispituje svaki piksel na slici kako bi vidio postoji li kut, ispitujući susjedstvo (engl. *patch*) piksela i uspoređujući koliko je susjedstvo piksela slično drugim susjedstvima u blizini. Sličnost se dobiva zbrajanjem kvadratnih razlika između dva susjedstva. Što je broj niži, sličnost je veća. Ako se piksel nalazi u području slike jednolikog intenziteta, onda će susjedstvo izgledati slično. U slučaju da se piksel nalazi na rubu slike, onda će susjedstvo koji se nalazi u smjeru okomito na rub slike izgledat sasvim drugačije, dok će oni paralelnom smjeru u odnosu na rub rezultirati malom promjenom. Ako se piksel nalazi na značajci s promjenama u svim smjerovima, onda niti jedno susjedstvo neće izgledati slično. Jačina kuta je definirana kao najmanji zbroj kvadratnih suma između susjedstva i okoline. Ako je taj broj lokalno maksimalan, onda postoji točka interesa.

2.1.3. Blob detekcija

Metode detekcije *bloba* u računalnom vidu su usmjerene da otkrivanje područja slike koja se razlikuju u svojstvima, kao što je svjetlina slike ili boja, u usporedbi s okolnim područjima [6]. *Blob* se može definirati kao područje slike u kojem su određena svojstva konstanta ili približno konstanta, odnosno sve se točke *bloba* mogu smatrati sličnima. Najčešća metoda detekcije *bloba* je konvolucija.

Postoje dvije glavne metode detekcije *bloba* s obzirom na područje koje će se ispitivati, a to su: diferencijalne metode koje se temelje na izvodima funkcije s obzirom na poziciju i metode koje se temelje na pronalaženju lokalnih maksimuma i minimuma funkcije. [6]

Razvoj *blob* detektora je bitan jer pruža dodatne informacije o područjima slike koje nisu dobivene korištenjem detektora rubova ili detektora kutova. Blob detekcija je korištena i za otkrivanje regija interesa (engl. *Region of interest - ROI*) koje se zatim dalje obrađuju. Ta područja mogu signalizirati prisutnost objekata ili dijelova objekata na slici pomoću aplikacije za detekciju istih. Također, *blob* detektori se koriste za analizu i prepoznavanje teksture slike.

2.1.4. Ridge detekcija

Blob detekcija signalizira postojanje predmeta na određenom području slike, a budući da *ridge* detekcija osim toga omogućuje otkrivanje približne osi simetrije promatranog objekta, ista se smatra poboljšanom verzijom *blob* detekcije [7]. Jedan od razloga za razvijanje *ridge* detektora u računalnom vidu je mogućnost otkrivanja unutrašnjosti izduženih objekata na slici. *Ridge* se može promatrati kao dopuna interesnim točkama ili lokalnim ekstremnim točkama. *Ridge* detektori se koriste za otkrivanje cesta na zračnim slikama, za otkrivanje krvnih žila ili za trodimenzionalne slike magnetske rezonance [8].

2.2. Ekstrakcija značajki

Prije dobivanja svojstava, na uzorkovanu sliku se primjenjuju razne tehnike obrade kao npr. binarizacija, normalizacija, promjena veličine slike i sl. Nakon toga, primjenjuju se tehnike ekstrakcije značajki kako bi se dobila svojstva koja su korisna u razvrstavanju i raspoznavanju slika [9]. Tehnike ekstrakcije značajki su korisne u različitim aplikacijama za obradu slika, kao npr. prepoznavanje znakova, odnosno likova.

Ekstrakcija značajki opisuje bitne oblike informacija koje su sadržane u uzorku tako da se zadatak razvrstavanja uzorka jednostavno obavlja određenim postupkom. U prepoznavanju uzoraka i obradi slika, ekstrakcija značajki je poseban oblik redukcije dimenzija slike [9]. Glavni cilj je dobiti najbitnije informacije iz izvornih podataka i te informacije predstaviti u manjim

dimenzijama. Kada su ulazni podaci preveliki da bi ih algoritam obradio te se smatra da postoji višak (previše podataka, nedovoljno informacija), onda se ulazni podaci preoblikuju u smanjen prikaz skupa značajki također zvan vektor značajki. Pretvaranje ulaznih podataka u skup značajki se naziva ekstrakcija značajki. Ako su izdvojene značajke odabrane pažljivo, očekuje se da će skup značajki izdvojiti važne informacije iz ulaznih podataka kako bi se željeni zadatak izvršio koristeći reducirani prikaz podataka. Prepoznavanje uzoraka je nova i rastuća oblast u području obrade slike. Koristi se na mnogo načina, kao što su prepoznavanje znakova, provjera dokumenata, čitanje bankovnih i kreditnih kartica, čitanje poštanske adrese, prepoznavanje teksta itd. Također, prepoznavanje znakova se primjenjuje u novonastalim oblastima kao što su razvoj elektroničkih knjižnica, multimedijskih baza podataka, te sustava koji zahtijevaju ručni unos podataka. Prepoznavanje optičkih znakova (OCR) je postupak pretvaranja skeniranih slika strojno otisnutog ili rukom pisanog teksta u format koji se može obraditi računalom. Proces prepoznavanja optičkih znakova ima tri faze: predobrada, ekstrakcija značajki, razvrstavanje.

Ekstrakcija značajki se obavlja nakon faze predobrade u sustavu prepoznavanja znakova. Osnovni zadatak prepoznavanja uzorka je uzeti ulazni uzorak i ispravno ga dodijeliti kao jedan od mogućih izlaznih klasa [9]. Proces se može podijeliti u dvije opće faze, a to su odabir značajki i klasifikacija. Odabir značajki je ključan za cijeli proces, inače klasifikator neće moći prepoznavati iz loše odabranih značajki. Ekstrakcija značajki je važan korak u konstrukciji bilo kojeg klasifikacijskog uzorka te je cilj izdvojiti bitne informacije koje karakteriziraju svaku klasu. U tom su procesu bitna svojstva izdvojena iz objekata kako bi se formirali vektori značajki koje koristi klasifikator kako bi povezao ulaznu jedinicu s ciljnom izlaznom jedinicom. Klasifikatoru je najlakše takvo prepoznavanje jer mu omogućuje istovremeno da vidi i usporedi obje jedinice i njihova svojstva. Ekstrakcija značajki podrazumijeva pronalaženje skupa parametara koji definiraju oblik znaka precizno i jedinstveno. Tijekom faze ekstrakcije značajki, svaki znak je predstavljen vektorom značajki koji postoje njegov identitet.

2.3. Scale Invariant Feature Transform (SIFT)

SIFT je jedan od algoritama koji se koristi u području računalnog vida, a služi za pronalaženje i opisivanje lokalnih značajki neke slike [10]. Karakteristika SIFTa je što transformira podatke slike u koordinate koje su invarijantne na promjenu veličine povezane s lokalnim značajkama. Također, otporan je i na promjene osvjetljenja na slici, promjene točke gledišta i zaklonjenost objekta. Ovakav način pristupa generira veliki broj značajki koje gusto prekrivaju sve razine veličine i lokacije slike. Slika veličine 500x500 piksela će imati oko 2000 stabilnih značajki (ovaj broj ovisi o sadržaju slike i odabranim parametrima). Količina značajki je naročito važna

za prepoznavanje objekata. Naime, za detekciju malih objekata koji se nalaze na zatrpanoj podlozi potrebno je da se najmanje tri značajke točno podudaraju [11].

Za podudaranje i prepoznavanje slike, SIFT značajke se prvo izvlače iz skupa referentnih slika te su zatim spremljene u bazu podataka. Nova slika se podudara s osnovnom na način da se svaka njena značajka pojedinačno uspoređuje s podacima u bazi te se pronalazi kandidat uspoređivanjem zajedničkih značajki na temelju Euklidove udaljenosti njihovih vektora značajki.

Deskriptori ključnih točaka su vrlo prepoznatljivi, što daje veliku vjerojatnost pojedinačnim značajkama da pronađu odgovarajući par u velikoj bazi podataka [11]. Međutim, na pretrpanoj slici mnoge značajke neće pronaći odgovarajući par u bazi podataka, što dovodi do mnogih lažnih podudaranja uz one ispravne. Ispravna podudaranja se mogu izdvojiti iz cijelog skupa tako što se identificiraju podskupine ključnih točaka koje odgovaraju objektu i njegovom položaju, skali (engl. *scale*) i orijentaciji objekta na slici. Vjerojatnost da će neke značajke slučajno odgovarati ovim parametrima je puno manja od vjerojatnosti da podudaranje pojedinačne značajke bude pogrešno. Određivanje tih dosljednih grupa se može vrlo brzo izvršiti korištenjem učinkovite tablice generalizirane Houghove transformacije.

Svaki skup od tri ili više značajki koje se podudaraju s objektom i njegovom položajem su predmet za daljnju detaljniju provjeru. Najprije je napravljena *least - squared* procjena za srodnu aproksimaciju položaja objekta [11]. Identificiraju se i sve druge značajke slike u skladu s položajem objekta, a one koje ne odgovaraju su odbačene. Naposljetku, detaljno se izračunava vjerojatnost da određeni skup značajki označava postojanje objekta, uzevši u obzir stvarna i moguća lažna podudaranja. Objekti koje se podudaraju, odnosno prolaze sve korake, s visokom sigurnošću se identificiraju se kao točna podudaranja.

Glavne faze koje se koriste za generiranje skupa značajki:

1. *Scale - space* detekcija ekstrema: Prva faza pretraživanja za sva mjerila i položaje na slici. [11]

2. Lokalizacija ključnih točki: Na mjestu položaja svake moguće značajke je korišten detaljan model kako bi se odredila lokacija i skala. Ključne točke su odabrane na temelju mjere njihove stabilnosti. [11]

3. Dodjela orijentacije: Jedna ili više orijentacija dodjeljuje se svakoj lokaciji ključne točke na temelju smjera lokalnog gradijenta. Sve buduće obrade slike se izvode na podacima koji su

transformirani u odnosu na dodijeljenu orijentaciju, skalu i lokaciju svake značajke, čime se osigurava invarijantnost na te transformacije. [11]

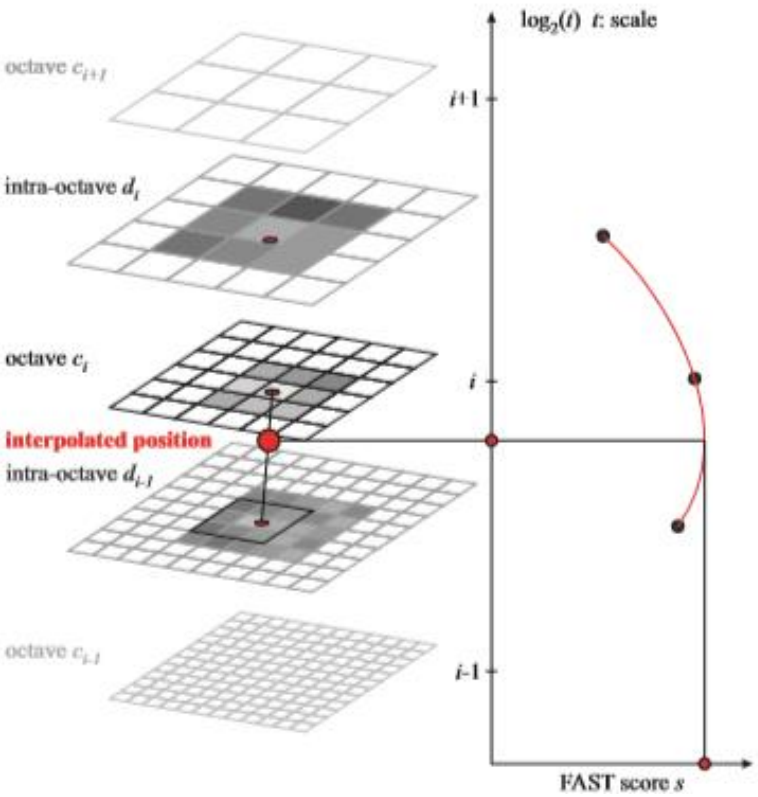
4. Deskriptor ključnih točaka: U okolini svake ključne točke mjere se lokalni gradijenti po odabranoj skali te se transformiraju u prikaz koji dopušta značaja iskrivljenja lokalnog oblika i promjene osvjetljenja. [11]

3. BINARY ROBUST INVARIANT SCALABLE KEYPOINTS (BRISK)

BRISK je jedan od algoritama koji se koristi za detekciju ključnih točaka slike te omogućuje kvalitetan opis istih. Zahvaljujući binarnoj prirodi deskriptora, ključne točke generirane ovim algoritmom se mogu vrlo uspješno podudarati. Omogućuje brzu detekciju iskorištavajući uštede brzine.

3.1. Scale – space detekcija ključnih točaka

Scale – space teorija služi za rukovanje slikovnim strukturama koje su na različitim skalama. Govoreći o BRISK-u, slojevi *scale – space* piramide se sastoje od n oktava c_i i n unutarnjih oktava d_i za $i = \{0, 1, \ldots, n-1\}$ i obično $n = 4$ [12]. Oktave se oblikuju progresivno poluuzorkovanjem izvorne slike (odgovara c_0). Svaka unutarnja oktava d_i se nalazi između slojeva c_i i c_{i+1} kao što je prikazano na slici 3.1. Ključna točka je identificirana na oktavi c_i.



Slika 3.1. *Scale – space detekcija točke* [12]

Prva unutarnja oktava d_0 dobiva se smanjenjem originalne slike c_0 faktorom od 1.5, dok su ostali slojevi unutarnje oktave izvedeni uzastopnim poluuzorkovanjem [12]. Stoga, ako t označava skalu onda $t(c_i) = 2^i$ i $t(d_i) = 2^i \cdot 1.5$.

Važno je napomenuti da i FAST i AGAST pružaju različite alternative oblika maske za otkrivanje ključnih točaka. U BRISK-u se uglavnom upotrebljava maska 9-16, koja u biti zahtijeva najmanje 9 uzastopnih piksela u krugu od 16 piksela kako bi bila dosta svjetlija ili tamnija od središnjeg piksela za ispunjavanje FAST kriterija. U početku, detektor FAST 9-16 primjenjuje se zasebno na svaku oktavu i unutrašnju oktavu koristeći isti prag T kako bi se identificirale moguće regije od interesa. Zatim, točke koje pripadaju tim područjima podvrgavaju se supresiji ne-maksima u *scale-space*: prvo, upitna točka mora ispuniti maksimalno stanje u odnosu na 8 susjednih FAST rezultata *s* u istom sloju. Rezultat *s* definiran je kao maksimalna vrijednost praga gdje se promatrana točka u slici i dalje smatra kutom. Drugo, rezultati u sloju iznad i ispod trebaju biti niži. Provjerava se unutar kvadratnih dijelova jednake veličine: bočna dužina je odabrana da bude 2 piksela u sloju s pretpostavljenim maksimumom. Budući da su susjedni slojevi (a time i njihovi FAST rezultati) predstavljeni različitom diskretizacijom, primjenjuju se neke interpolacije na granicama slojeva. Slika 3.1. prikazuje primjer ovog uzorkovanja i pretraživanja maksima.

Otkrivanje maksima preko osi skale kod oktave c_0 (sl. 3.1.) je poseban slučaj: da bi se dobili FAST rezultati za virtualni unutrašnju oktavu d_{-1} ispod c_0, primjenjujemo masku FAST 5-8 na c_0. međutim, u ovom slučaju nije potrebno da rezultati u sloju d_{-1} budu niži od rezultata ispitivane točke u oktavi c_0. [12]

Uzevši u obzir *image saliency* kao kontinuiranu veličinu ne samo duž slike nego i duž dimenzije ljestvice, izvršava se subpikselno i kontinuirano usavršavanje ljestvice za svaki otkriveni maksimum. Kompleksnost procesa usavršavanja ograničava se *fitanjem* 2D kvadratne funkcije (funkcija najmanjih kvadrata) na svaki od 3 kombinacije rezultat-susjedstvo (engl. *score-patch*). Rezultat su tri sub-pikselna *saliency* maksimuma [12]. Kako bi se izbjeglo ponovno uzorkovanje, smatra se da je na svakom sloju rezultat 3 po 3. Zatim, ovi usavršeni rezultati koriste se na 1D paraboli duž osi ljestvice koja daje konačnu procjenu rezultata i ljestvice na njenom maksimumu. Za završni korak koordinati slike se ponovno interpoliraju između dijelova u slojevima do određenje ljestvice.

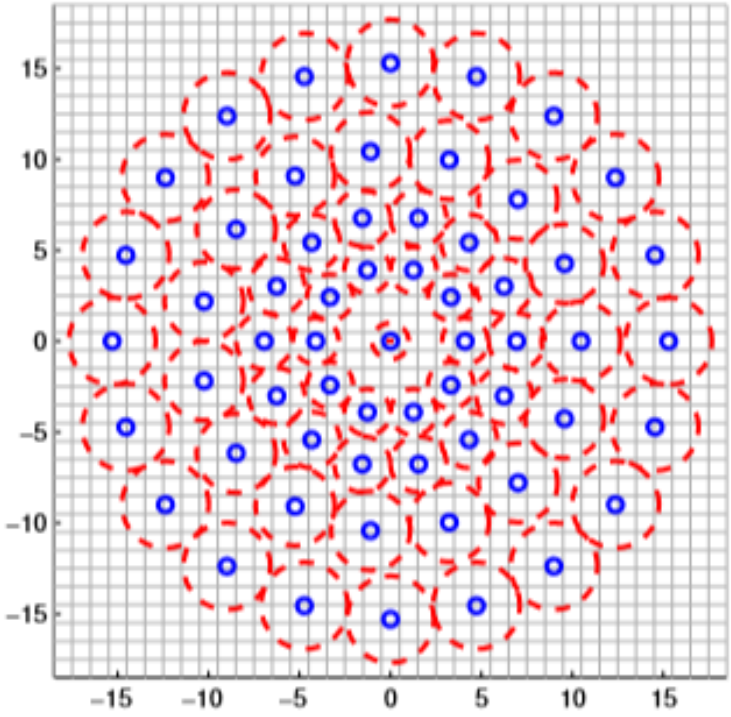
3.2. Opis ključnih točaka

S obzirom na skup ključnih točaka, BRISK deskriptor je sastavljen kao binarni niz koji je nastao nadovezivanjem rezultata jednostavnih testova usporedbe svjetline [12]. Ova ideja se pokazala kao vrlo učinkovita. Kod BRISK algoritma se prepoznaje karakterističan smjer svake ključne točke kako bi se omogućili orijentacijski normalizirani deskriptori i tako se postiže invarijantnost

na rotaciju koja je ključna za robusnost. Također, pažljivo se odabiru usporedbe svjetline s naglaskom na maskimiziranje deskriptivnosti.

3.2.1. Uzorkovanje i procjena rotacije

Ključni koncept BRISK deskriptora je uzorkovanje okoline ključne točke. Uzorkovanje, prikazano na slici 3.2., definira N jednako razmaknutih mjesta na krugovima koji su koncentrični s ključnom točkom [12]. Plavi krugovi označavaju mjesta uzorkovanja, a crveni isprekidani krugovi su polumjera σ koji odgovara standardnoj devijaciji Gaussovog *karnela*.



Slika 3.2. *BRISK uzorkovanje gdje je N = 60* [12]

Da bi se izbjeglo preklapanje pri uzorkovanju intenziteta točke p_i, primjenjuje se Gaussovo izjednačavanje korištenjem standardne devijacije σ_i proporcionalne udaljenosti između točaka na odgovarajućem krugu. Pozicioniranje i skaliranje uzorka za ključnu točku k na slici nam omogućuje razmatranje jednog od $N \cdot (N-1)/2$ parova uzorkovanih točaka (p_i, p_j). Izmjerene vrijednosti intenziteta ovih točaka koje se označavaju kao $I(p_i, \sigma_i)$ i $I(p_j, \sigma_j)$ se koriste za izračunavanje lokalnog gradijenta $g(p_i, p_j)$ formulom

$$g(p_i, p_j) = (p_j - p_i) \cdot \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2} \quad (3-1)$$

Uzimajući u obzir skup svih parova uzorkovanja $\mathcal{A}$:

$$\mathcal{A} = \{(p_i, p_j) \in \mathbb{R}^2 \times \mathbb{R}^2 \mid i < N \wedge j < i \wedge i, j \in \mathbb{N}\} \quad (3-2)$$

definira se podskup parova na kratkoj udaljenosti $\mathcal{S}$ i drugi podskup parova na dugoj udaljenosti $\mathcal{L}$

$$\mathcal{S} = \{(p_i, p_j) \in \mathcal{A} \mid \|p_j - p_i\| < \delta_{max}\} \subseteq \mathcal{A} \quad (3-3)$$

$$\mathcal{L} = \{(p_i, p_j) \in \mathcal{A} \mid \|p_j - p_i\| > \delta_{max}\} \subseteq \mathcal{A} \quad (3-4)$$

Prag udaljenosti je postavljen na δ_{max} = 9.75t i δ_{min} = 13.67t (t je skala od k). Procjenjuje se cjelokupni karakteristični smjer uzorkovanja ključne točke k:

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \cdot \sum_{(p_i, p_j) \in \mathcal{L}} g(p_i, p_j) \quad (3-5)$$

3.2.2. Deskriptor

Za formiranje rotacijski i skala – normaliziranog deskriptora, u BRISK metodi se primjenjuje uzorak rotiran za α = arctan2(g_y, g_z) oko ključne točke k. Izvršavajući usporedbe intenziteta kratke udaljenosti parova točaka (p_i^α, p_j^α) formira se deskriptor bit – vektora d_k , tako da svaki bit b odgovara:

$$b = \begin{cases} 1, & I(p_j^\alpha, \sigma_j) > I(p_i^\alpha, \sigma_i) \\ 0, & za\ sve\ ostalo \end{cases} \quad (3-6)$$

$$\forall (p_i^\alpha, p_j^\alpha) \in \mathcal{S}$$

BRISK koristi deterministički način uzorkovanja koji rezultira jednolikom gustoćom uzorkovanja u određenom polumjeru oko ključne točke. U usporedbi s BRIEF deskriptorom, BRISK koristi znatno manje točaka za uzorkovanje, ograničavajući složenost vrijednosti traženih intenziteta. Nadalje, usporedbe su prostorno ograničene tako da se varijacije svjetline zahtijevaju

samo da bi bile lokalno konzistentne. Uz uzorkovanje i prag udaljenosti dobivamo niz od 512 bita. [12]

3.2.3. Podudaranje

Podudaranje dva BRISK deskriptora je jednostavno izračunavanje njihove Hammingove udaljenosti [12]. Broj bitova koji su različiti u dva deskriptora označava mjeru njihove različitosti.

4. EKSPERIMENTALNI DIO

4.1. OpenCV

OpenCV (*Open Source Computer Vision*) je biblioteka otvorenog koda koja se koristi za rad računanim vidom, a odnosi se na aplikacije koje rade s podacima u stvarnom vremenu [13]. Bitan cilj OpenCV-a je pružanje jednostavne infrastrukture za rad s računalnim vidom, te omogućavanje jednostavnog i brzog razvoja jednostavnih, ali i složenijih aplikacija. Biblioteka sadrži preko 2500 optimiziranih algoritama koji obuhvaćaju gotovo sva područja računalne vizije i strojnog učenja. Ti se algoritmi mogu koristiti za otkrivanje i prepoznavanje lica, prepoznavanje objekata, praćenje pokretnih objekata itd. OpenCV je originalno pisan u programskom jeziku C budući da obavljanje velikog broja operacija i algoritama u stvarnom vremenu zahtjeva optimiziran kod koji je pisan na nižoj razini. Također, razvijena su sučelja i prema drugom jezicima više razine kao štu su Java, Python, Ruby itd.

4.2. Eksperimenti

U nastavku su prikazane neke od slika koje su korištene za eksperimente te rezultati eksperimenata za svaku od četiri metode, a to su BRISK, FREAK, SIFT i SURF. Kao ulaz u svaku metodu korištene su dvije slike. Metode trebaju odrediti podudaranje ključnih točaka na dvije ulazne slike. Bitna stavka je da se svaka točka jedne slike mora točno podudarati s točkom druge slike, te će se na osnovi tog kriterija uspoređivati postotak točnosti algoritama. Za svaku metodu detektirane su ključne točke i izračunati deskriptori. Izračunati deskriptori za svaku sliku su međusobno uspoređeni korištenjem *brute-force matchera,* te je odabrano 10 točaka koje se najbolje podudaraju. Rezultati podudaranja predstavljeni su na sljedećim slikama.

Primjer 1.

Slika 4.1. *Ulazne slike za prvi primjer*

Slika 4.2. *Rezultat BRISK metode*

Slika 4.3. *Rezultat FREAK metode*

Slika 4.4. *Rezultat SIFT metode*

Slika 4.5. *Rezultat SURF metode*

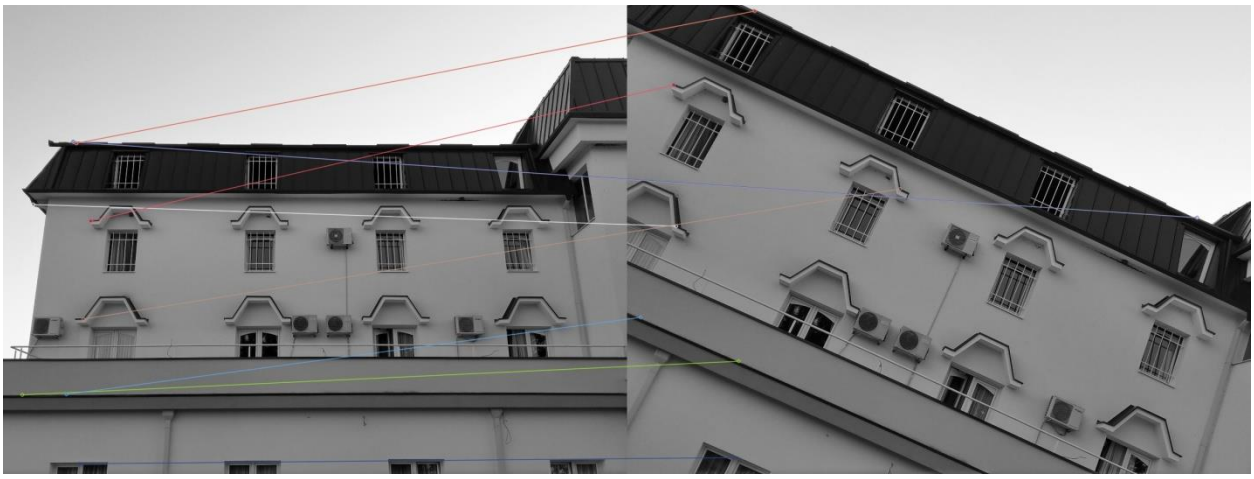
BRISK i SIFT u ovom slučaju svih deset ključnih točaka povezuju točno (sl. 4.2. i 4.4.)., dok FREAK i SURF u odnosu na BRISK po jednu točku povezuju pogrešno. FREAK pogrešno povezuje točke prozora zgrade (sl. 4.3.), dok SURF pogrešno povezuje dio zgrade koji nije u sjeni u pozadini stabala (sl. 4.5.).

Primjer 2.

Slika 4.6. *Ulazne slike za drugi primjer*

Slika 4.7. *Rezultat BRISK algoritma*

Slika 4.8. *Rezultat FREAK algoritma*



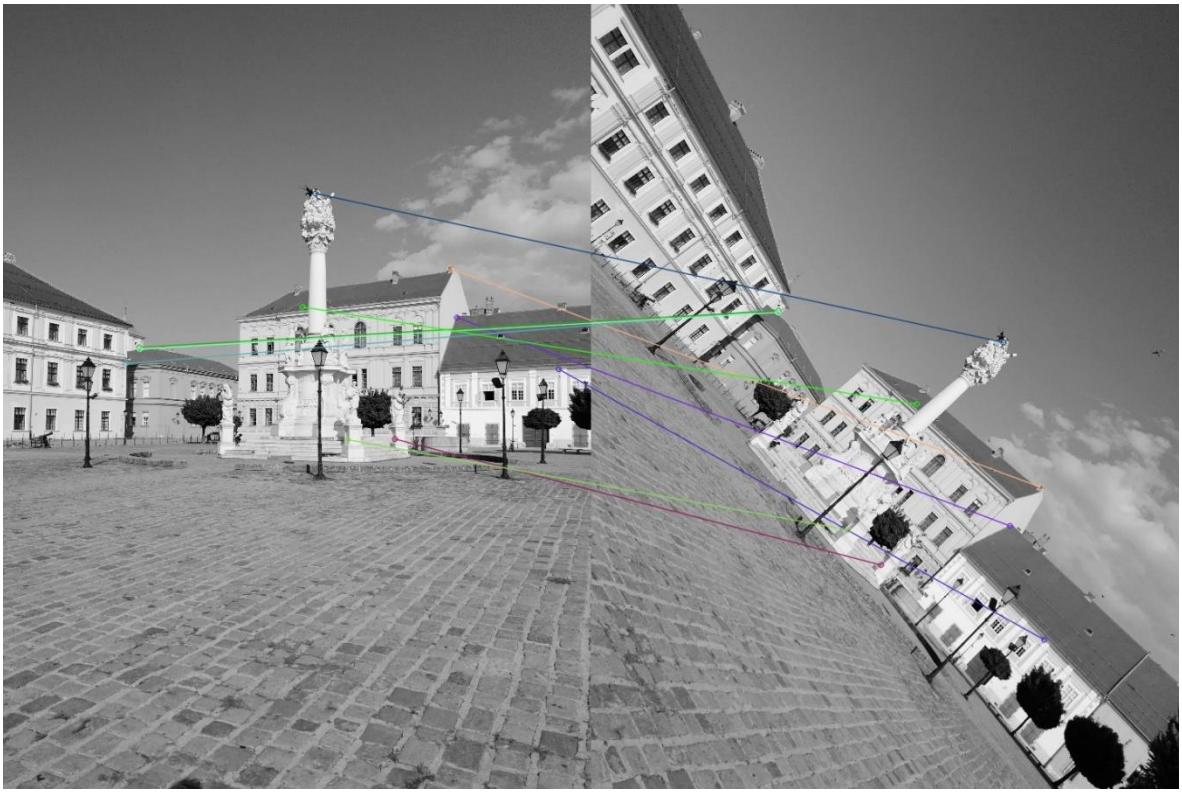
Slika 4.9. *Rezultat SIFT algoritma*

Slika 4.10*. Rezultat SURF algoritma*

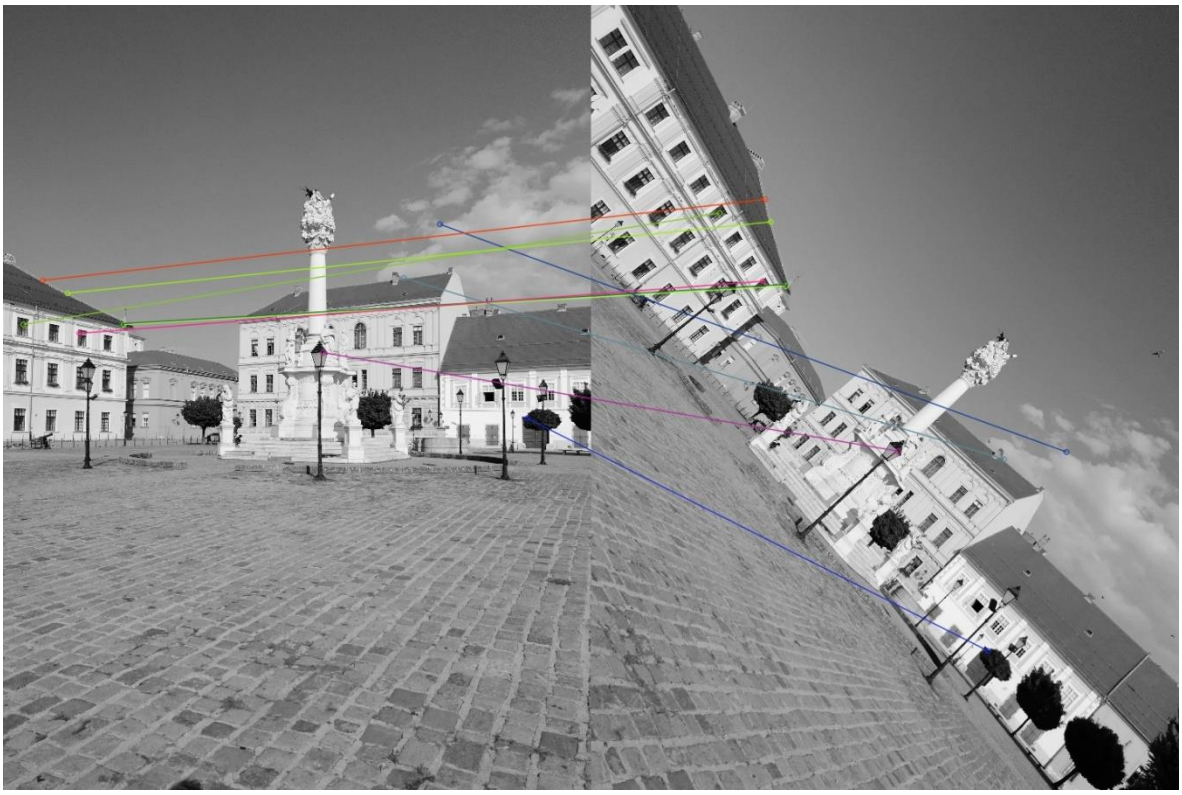
U ovom slučaju objekt druge slike je uz to što je rotiran, ujedno i uvećan, te BRISK algoritam svakako sve ključne točke slike povezuje točno (sl. 4.7.). FREAK u ovom slučaju griješi te samo dvije točke slike povezuje točno (sl. 4.8.). SIFT, kao i FREAK, griješi te samo jednu točku povezuje točno (sl. 4.9.). SURF točno povezuje osam točaka što je puno bliže rezultatu BRISK-a, u odnosu na SIFT i FREAK (sl. 4.10.).

Primjer 3.

Slika 4.11. *Ulazne slike za treći primjer*

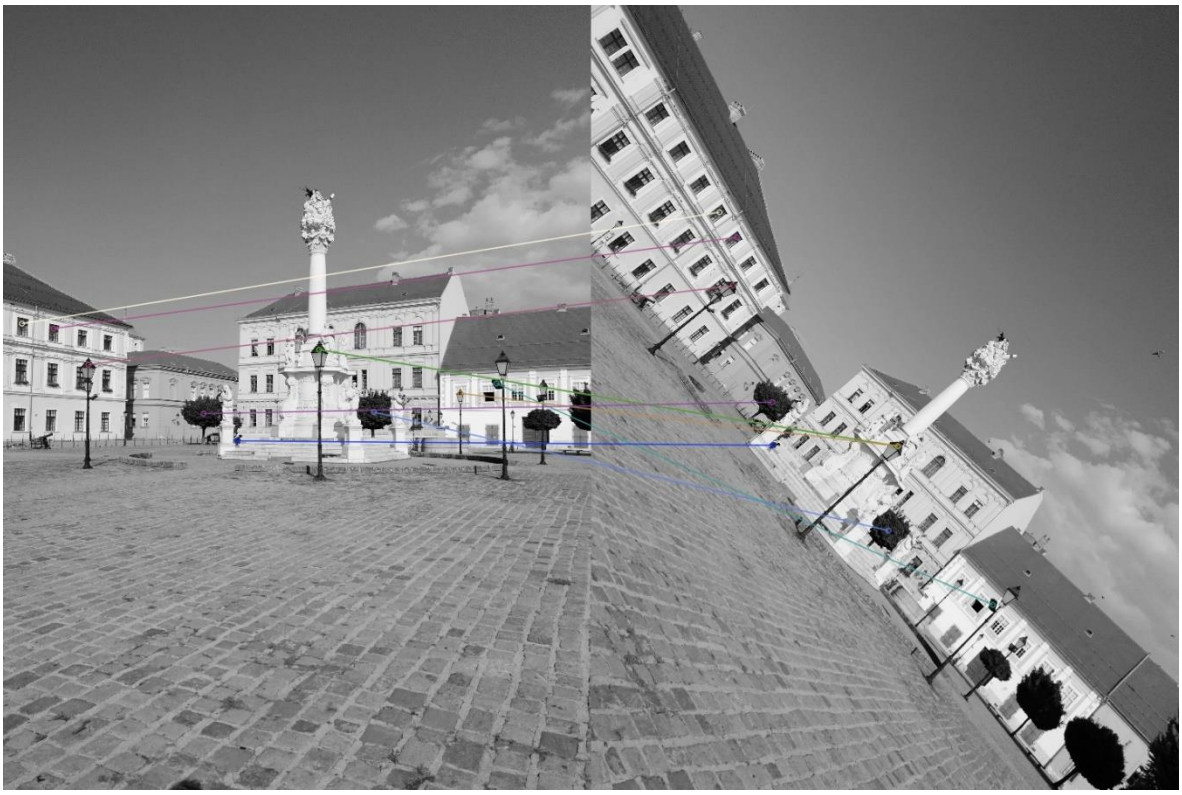


Slika 4.12. *Rezultat BRISK algoritma*



Slika 4.13. *Rezultat FREAK algoritma*

Slika 4.14. *Rezultat SIFT algoritma*



Slika 4.15. *Rezultat SURF algoritma*

U trećem primjeru FREAK, SIFT i SURF algoritmi povezuju točke s podjednakom točnošću, odnosno jednu ili dvije ključne točke povezuju krivo (sl. 4.13., 4.14., 4.15.), dok BRISK i dalje sve točke povezuje točno (sl. 4.12.).

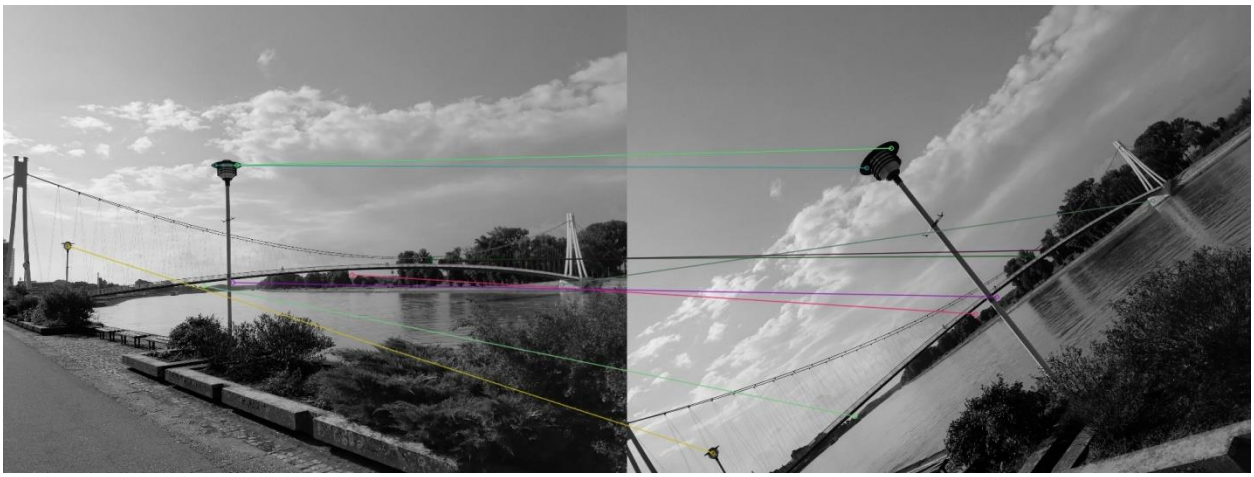
Primjer 4.

Slika 4.16. *Ulazne slike za četvrti primjer*

Slika 4.17. *Rezultat BRISK algoritma*

Slika 4.18. *Rezultat FREAK algoritma*

Slika 4.19. *Rezultat SIFT algoritma*



Slika 4.20. *Rezultat SURF algoritma*

U četvrtom primjeru se vidi kako BRISK i FREAK algoritmi sve točke povezuju točno. (sl. 4.17. i 4.18.). SIFT krivo povezuje dvije točke, odnosno grm lijeve slike i krošnju stabla na desnoj slici, te svjetiljku i rub mosta (sl. 4.19.). SURF pogrešno povezuje samo jednu točku, a to je okolina prve svjetiljke (sl. 4.20.). Pogrešno detektirane točke su mogle nastati jer se svjetiljka u odnosu na most na drugoj slici nalazi u drugačijem položaju, odnosno "siječe" most na potpuno drugom mjestu.

4.3. Analiza rezultata

Eksperimenti su rađeni za sedam parova slika, a u radu su prikazana četiri primjera. Na svaki par slika su primijenjena četiri algoritma, a to su BRISK, FREAK, SIFT i SURF. U svakom primjeru je pojedini algoritam povezivao ključne točke jedne slike s točkama druge slike te je cilj bio da su točke povezane točno, odnosno da su povezane točke identične. Slike su se razlikovale u tome što su objekti na prvoj slici uglavnom vodoravno posloženi, dok je druga slika rotirana, ili su objekti slike uvećani i sl. Tablica 4.1. prikazuje rezultate točnosti za svaku metodu za pojedini primjer. Prvi broj se odnosi na broj točno povezanih točaka, dok se drugi odnosi na broj ukupno povezanih točaka, kojih je za svakih par slika bilo deset. Prva četiri primjera se odnose na slike koje su prikazane u radu, dok se posljednja tri odnose na slike na kojima su također provođeni u eksperimenti. U tablici 4.2. su prikazani ukupni rezultati točno povezanih točaka za svaku metodu, kao i prosječna točnost svake metode.

Slika	1.	2.	3.	4.	5.	6.	7.
BRISK	10/10	10/10	10/10	10/10	10/10	10/10	10/10
FREAK	9/10	2/10	9/10	10/10	10/10	10/10	10/10
SIFT	10/10	1/10	8/10	8/10	10/10	9/10	10/10
SURF	9/10	8/10	9/10	9/10	9/10	8/10	8/10

Tablica 4.1. *Rezultati točno povezanih točaka za svaki primjer*

	Ukupan broj povezanih točaka	Broj točno povezanih točaka	Prosjek
BRISK	70	70	10
FREAK	70	60	8,57
SIFT	70	56	8
SURF	70	60	8,57

Tablica 4.2. *Ukupni rezultati svake metode*

Iz dobivenih rezultata se može vidjeti kako je BRISK metoda na svakom primjeru točno povezala svih deset ključnih točaka slike, dok ostale tri metode imaju podjednako dobre rezultate. Važno je primijetiti kako su u slučaju kada je slika uz to što je rotirana ujedno i uvećana algoritmi FREAK i SIFT napravili veliki broj grešaka, dok je SURF točno povezao gotovo sve točke. Iako FREAK i SURF imaju prosječno jednaku točnost, SURF pojedinačno za svaku sliku ima bolje rezultate budući da su svi bili poprilično blizu rezultatima BRISKA.

5. ZAKLJUČAK

U računalnom vidu vrlo važnu uloga imaju tehnike koje se koriste za rastavljanje slike na lokalne značajke te se pomoću njih slika kasnije obrađuje. Postoje razne tehnike koje omogućavaju detekciju značajki. U ovom radu je objašnjen način rada algoritma BRISK koji omogućava detekciju značajki slike te njihovo povezivanje.

Provedeni su eksperimenti kako bi se pokazala točnost i učinkovitost algoritma u odnosu na ostale metode. Iz prikazanih rezultata se jasno vidi kako je spomenuti algoritam točno povezivao točke jedne slike s točkama druge slike, unatoč tome što je slika rotirana, uvećana, ili što je svjetlina slika jača, odnosno slabija u odnosu na prvu.

Upravo na osnovi provedenih eksperimenata je dokazano kako je BRISK invarijantan na skalu i rotaciju, što čini prednost u odnosu na ostale algoritme budući da su algoritmi pogrešno povezivali točke ako je slika znatno uvećana. Time se zaključuje kako se BRISK pokazao znatno bolji, točniji i brži te pokazuje izvrsne rezultate.

6. LITERATURA

[1] Science Daily: Computer Vision, https://www.sciencedaily.com/terms/computer_vision.htm, pristupljeno: 17.9.2018.

[2] Feauture detection (computer vision), https://en.wikipedia.org/wiki/Feature_detection_(computer_vision), pristupljeno: 14.9.2018.

[3] Edge detection, https://en.wikipedia.org/wiki/Edge_detection, pristupljeno: 14.9.2018.

[4] Corner detection, https://en.wikipedia.org/wiki/Corner_detection, pristupljeno: 14.9.2018.

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

[6] Blob detection, https://en.wikipedia.org/wiki/Blob_detection, pristupljeno: 15.9.2018.

[7] T. Lindberg; Edge detection and ridge detection with automatic scale selection; Royal Institute of Technology; Stockholm,Sweden, http://www.diva-portal.org/smash/get/diva2:452310/FULLTEXT01.pdf, pristupljeno: 15.9.2018.

[8] Ridge detection, https://en.wikipedia.org/wiki/Ridge_detection, pristupljeno: 15.9.2018.

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

[10] Scale-invariant feature transgorm, https://en.wikipedia.org/wiki/Scale-invariant_feature_transform, pristupljeno 13.9.2018.

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

[12] Stefan Leutenegger, Margarita Chli and Roland Y . Siegwart; BRISK: Binary Robust Invariant Scalable Keypoints; Autonomous Systems Lab, Switzerland

[13] OpenCV, https://opencv.org/about.html, pristupljeno: 15.9.2018.

7. SAŽETAK

Računalni vid je područje u računarstvu koje omogućava računalima da identificiraju i obrađuju slike. Tehnika koja se koristi u računalnom vidu je rastavljanje slike u lokalne značajke te se na taj način ublažava složenost slike. Te značajke imaju jako važnu ulogu u području obrade slike te se pomoću njih vrši obrada iste. Da bi se slika mogla obrađivati, potrebno je te značajke točno detektirati.

Cilj ovog rada je predstavljanje algoritma BRISK koji se koristi upravo za detekciju i opis ključnih točaka. U radu su prikazani rezultati algoritma BRISK u odnosu na druge algoritme koji se također koriste za detekciju ključnih točaka. BRISK se pokazao kao dosta točniji te brži.

Može se zaključiti kako je na osnovi provedenih eksperimenata potvrđeno kako je BRISK invarijantan i skalu i na rotaciju, što je veoma bitna stavka, budući da su ostali algoritmi većinom pogrešno povezivali ključne točke jedne slike s rotiranom drugom slikom.

Ključne riječi: računalni vid, značajke, ključne točke, BRISK

8. ABSTRACT

Computer vision is an area in computing that enables computers to identify and process images. The technique used in computer vision is disassembling the image into local features and thus reducing the complexity of the image. These features play a very important role in the image processing field, which is processing the image. In order for the image to be processed, these features need to be accurately detected.

The aim of this paper is to present the BRISK algorithm, which is used for detection and description of key points. The paper presents the results of the BRISK algorithm compared to other algorithms that are also used to detect key points. BRISK proved to be much more accurate and faster.

Thus, it can be concluded, based on the experiments carried out, that BRISK is invariant both on the scale and rotation, which is a very important thing since the other algorithms mostly misconnected the key points of a single rotated image.

Keywords: computer vision, features, key points, BRISK

9. ŽIVOTOPIS

Veronika Viljušić je rođena 19.12.1996. u Mannheimu, Savezna Republika Njemačka. Pohađala je Osnovnu školu Orašje te nakon završene osnovne škole upisuje Opću gimnaziju u Orašju. Poslije srednje škole upisuje preddiplomski studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, te na drugoj godini upisuje smjer Komunikacije i informatika.

_____

Veronika Viljušić