

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD BR. 25

**ODREĐIVANJE KORESPODENTNIH
TOČAKA NA SLIKAMA DOBIVENIM
STEREO KAMEROM**

JUSUFADIS BAKAMOVIĆ

Zagreb, lipanj 2010.

SADRŽAJ

1	Uvod	- 5 -
2	Epipolarna geometrija	- 6 -
2.1	Fundamentalna matrica F	- 7 -
3	Lokalne značajke	- 8 -
3.1	Idealna značajka	- 8 -
3.2	Vrste značajki	- 9 -
3.3	Detektori značajki (engl. <i>feature detectors</i>)	- 10 -
3.3.1	Kutni detektori (engl. <i>corner detectors</i>)	- 10 -
3.3.2	Grudasti detektori (engl. <i>blob detectors</i>)	- 17 -
3.3.3	Detektori regija (engl. <i>region detectors</i>)	- 20 -
4	Efikasne implementacije ekstrakcije značajki	- 24 -
4.1	SURF (engl. <i>Speeded Up Robust Features</i>)	- 24 -
4.2	FAST (engl. <i>Features from Accelerated Segment Test</i>)	- 26 -
4.3	Pregled karakteristika detektora	- 27 -
5	SIFT (engl. <i>Scale Invariant Feature Transform</i>)	- 28 -
5.1	Detekcija ekstrema kroz niz skala	- 28 -
5.1.1	<i>LoG</i> vs. <i>DoG</i>	- 29 -
5.1.2	Formiranje <i>DoG</i> piramide	- 30 -
5.1.3	Traženje ekstrema	- 31 -
5.1.4	Određivanje parametara	- 32 -
5.2	Lokalizacija značajki	- 34 -
5.2.1	Odbacivanje nestabilnih značajki	- 35 -
5.3	Dodjeljivanje orijentacije	- 37 -
5.4	Izgradnja deskriptora	- 38 -
5.4.1	Reprezentacija deskriptora	- 39 -
5.4.2	Odabir parametara	- 40 -
5.4.3	Afine transformacije	- 41 -

5.5	Primjene	- 42 -
5.5.1	Prepoznavanje objekata (engl. <i>object recognition</i>)	- 42 -
6	Implementacija SIFT algoritma	- 47 -
6.1	<i>Feature</i> klasa	- 47 -
6.2	<i>SIFT</i> klasa	- 48 -
6.2.1	Preprocesiranje ulazne slike	- 48 -
6.2.2	Detekcija ekstrema kroz niz skala	- 49 -
6.2.3	Lokalizacija značajki	- 50 -
6.2.4	Dodjeljivanje orijentacije	- 51 -
6.2.5	Kreiranje deskriptora	- 52 -
6.2.6	Sučelje prema korisniku	- 53 -
6.3	Demo aplikacija	- 54 -
6.3.1	Detekcija SIFT značajki	- 55 -
6.3.2	Traženje korespondentnih točaka	- 57 -
7	Zaključak	- 59 -
8	Literatura	- 60 -
9	Naslov, sažetak i ključne riječi	- 63 -
10	Title, abstract and keywords	- 64 -

1 UVOD

Problem korespodencije u računarstvu svoju motivaciju pronalazi u ljudskom načinu vizualnog poimanja vanjskog svijeta. Svaki čovjek ima dva oka koja su međusobno blago udaljena. Iz tog proizlazi da se ljudskom mozgu zapravo šalju vizualni podražaji slične, pomaknute, scene ali iz dva različita kuta gledanja. U takvim situacijama ljudski je mozak u mogućnosti *automatizirano* spojiti takva dva različita pogleda u jedinstvenu sliku odnosno pronaći njihove korespodentne točke. Takvim problemom se bavi i grana računalnog vida – stereo vid (engl. *stereo vision*).

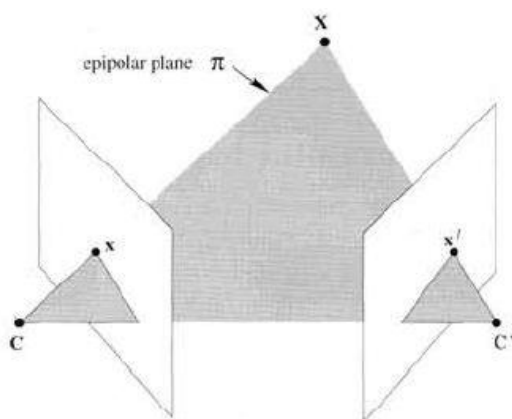
Traženje korespodentnih točaka u računarstvu je problem koji se odnosi na pronalaženje istovjetnih točaka na dvije različite slike dobivene iz različitih pogleda pomoću dvije kamere. Dohvaćanje slika može se odvijati istovremeno ali i u različitim vremenskim trenucima. Scena koja se analizira nije ograničena u kontekstu statičnosti već objekti koji je čine mogu mijenjati svoj položaj u trodimenzionalnom prostoru što pridonosi dodatnoj težini problema. Problem se također može poopćiti i na više slika odnosno više kamera. Korespodentne točke moguće je povezati matematički aparatom – epipolarnom geometrijom.

Pošto se stereo vid temelji na epipolarnoj geometriji drugo poglavlje donosi kratak uvod u istu. Međutim, kako se ovaj rad više fokusira na metode koje mogu na programski način riješiti problem korespodencije, opisuju se neke od danas najpopularnijih. Pošto se većina metoda temelji na ekstrakciji lokalnih značajki treće poglavlje donosi njihovu definiciju te opisuje neke od vrsta detektora značajki.

Četvrto poglavlje donosi kratak pregled nekih od efikasnih implementacija detektora značajki dok peto poglavlje detaljno opisuje SIFT algoritam koji danas slovi kao jedan od najstabilnijih. Iz tog razloga ovaj rad donosi i vlastitu implementaciju SIFT-a koja je opisana u šestom poglavlju.

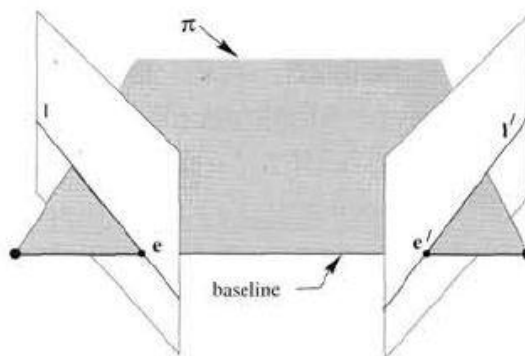
2 EPIPOLARNA GEOMETRIJA

Epipolarna geometrija je dio stereo vida koja opisuje odnose između slika dobivenih pomoću dvije kamere koje snimaju scenu iz dva različita kuta. Primjerice neka se točka X , koja se nalazi u 3D prostoru, projektivnom geometrijom preslikava u 2D prostor (slikovna ravnina s kamere) u točke x i x' . Sve tri točke, X , x i x' , su koplanarne odnosno leže u istoj epipolarnoj ravnini π . Problem koji definira epipolarna geometrija jest upravo određivanje odnosa između točaka x i x' .



Slika 1 Preslikavanje točke iz 3D svijeta u dvije 2D slikovne ravnine pomoću dvije kamere C i C' .

Kada bi samo točka x bila poznata tada bi se točka x' mogla izračunati iz tzv. *epipolarnog ograničenja*. Taj uvjet govori da se točka x' može nalaziti samo na optičkoj osi gdje epipolarna ravnina π siječe ravninu slike dobivene s druge kamere. Linija koja odgovara toj optičkoj osi se naziva *epipolarna linija* (na slici označene kao l i l'). Ovaj uvjet je od izuzetne važnosti pošto smanjuje broj točaka za koje treba ispitati da li su korespondentni par točki x . Linija na kojoj se nalaze obje kamere naziva se *bazna linija* (engl. *baseline*). Mjesto na kojem se sijeku bazna linija i ravnina slike naziva se *epipol* (na slici označeni kao e i e').



Slika 2 Elementi epipolarne geometrije.

2.1 FUNDAMENTALNA MATRICA F

Fundamentalna matrica opisuje epipolarnu geometriju na matematički način. Kao što je već zaključeno točka x' uvijek mora ležati na epipolarnoj liniji, l' , druge slikovne ravnine. Prema tome kako bi se pronašla točka x' , potrebno je samo pronaći preslikavanje točke x na epipolarnu liniju l' kako slijedi.

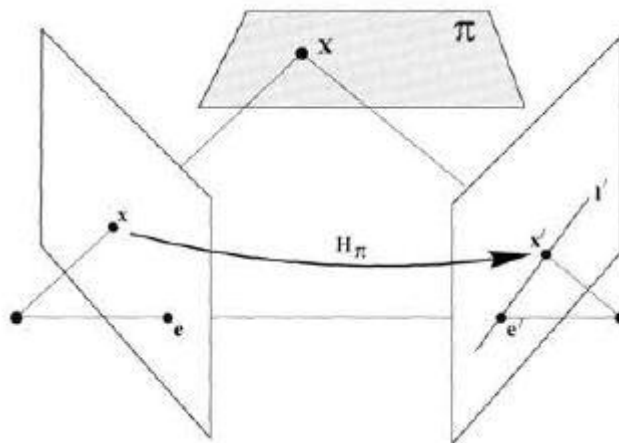
Neka pomoćna ravnina π ne siječe niti jedan od centara kamera. Neka je na mjestu gdje zraka prve kamere siječe pomoćnu ravninu točka X čija projekcija na drugu slikovnu ravninu odgovara točki x' koja se nalazi na epipolarnoj liniji l' . Kako epipol e' druge slikovne ravnine i točka x' leže na epipolarnoj liniji, tada vrijedi:

$$l' = e' \times x' \quad (1.1)$$

Ukoliko bi točku x' prikazali kao točku x pomnoženu s H_π tada bi se prethodni izraz mogao zapisati kao:

$$l' = e' \times x' = e' \times H_\pi x = Fx \quad (1.2)$$

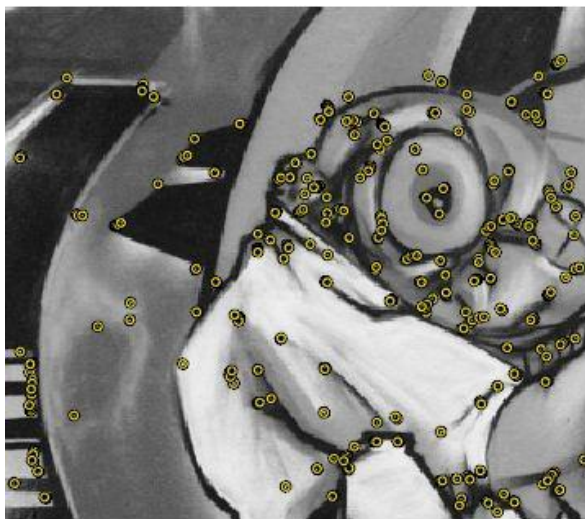
gdje je $e' \times H_\pi$ fundamentalna matrica F a H_π 2D homografska matrica koja se može definirati pošto su preslikavanja svake točke x_i u točku x'_i projektivno ekvivalentna u odnosu na točke X_i .



Slika 3 Ilustracija procesa pronalaženja fundamentalne matrice.

3 LOKALNE ZNAČAJKE

Lokalna značajka slike je uzorak koji se po svojim svojstvima razlikuje od svih ostalih koji se nalaze u neposrednom susjedstvu. Mjerilo različitosti direktno ovisi o problemu koji se rješava pa se prema tome može definirati na različite načine. Često se u raznim kombinacijama koriste svojstva kao što su intenzitet, boja i tekstura. Značajke mogu biti reprezentirane u različitim oblicima. Tako primjerice značajke mogu biti obične točke ali i kutovi, rubovi ili čak regije.



Slika 4 Primjer kutnih značajki.

Detektori značajki se oblikuju ovisno o primjeni pa iz tog razloga postoji niz različitih detektora koji izračunavaju značajke ovisno o kontekstu. Ponekad je kontekst poznat u smislu da značajka ima specifičnu semantičku interpretaciju i unaprijed poznato značenje. Primjer takvih značajki bi mogli biti rubovi detektirani na zračnim slikama koji predstavljaju ceste. Međutim, značenje značajke često nije poznato već se temelji na složenijim matematičkim izračunima čiji je cilj učiniti značajku što neosjetljivijom na neke tipične promjene kao što su recimo varijabilnost u osvjetljenju ili rotacije. Ovakve značajke nalaze svoju primjenu u praćenju i prepoznavanju objekata, kalibraciji kamere, 3D rekonstrukciji i sl.

3.1 IDEALNA ZNAČAJKA

Idealna situacija bi bila kada bi lokalne značajke semantički jednoznačno odgovarale dijelovima slike i time predstavljale određene objekte na slici, no kao što je već spomenuto to najčešće nije slučaj. Ipak, definirajmo svojstva značajki koja bi bila poželjna:

1. Ponovljivost (engl. *repeatability*)

U nizu različitih slika sličnog sadržaja koje su snimljene u različitim uvjetima, idealna značajka bi trebala biti prisutna u svim slikama.

2. Karakterističnost ili prepoznatljivost (engl. *distinctiveness*)

Značajke bi trebale imati što različitija svojstva kako bi se što lakše mogle raspoznavati. Drugim riječima značajka bi trebala sadržavati što više jednoznačnih informacija.

3. Lokalnost

Značajka bi trebala biti lokalna. Obično se definira susjedstvo nad kojim se posmatra lokalnost značajke.

4. Kvantiteta

Broj detektiranih značajki na jednoj slici bi trebao biti dovoljno velik kako bi se detektirale značajke i na manjim objektima. Ipak, optimalan broj značajki ovisi o primjeni.

5. Preciznost

Detektirane značajke bi trebalo biti moguće što preciznije locirati koristeći recimo razne interpolacije.

6. Efikasnost

Detekcija značajki ne bi trebala biti previše računski zahtjevna kako bi se mogle primjeniti i u vremenski kritičnim aplikacijama.

Jasno je da sve nabrojane karakteristike ne mogu biti u potpunosti zadovoljene pa je prema tome ovisno o situaciji nužan kompromis. *Ponovljivost* značajke najbitnija je karakteristika koja zapravo opisuje robusnost i stabilnost značajke. *Karakterističnost* i *lokalnost* značajke su svojstva koja istovremeno ne mogu biti zadovoljena. Što je značajka lokalnija to je informacija koju nosi sve manja. Slično, invarijantnost značajke poboljšava njezinu ponovljivost ali i umanjuje karakterističnost. Zahtjev na *preciznost* značajke ima posebnu važnost pri traženju određenih podudarnosti kao što je recimo slučaj u kalibraciji kamere ili estimaciji epipolarne geometrije. *Broj značajki* također je od velike važnosti i ovisi o specifičnoj primjeni. Ovisno o primjeni je potrebno dobro procijeniti odgovarajući broj jer u suprotnom može rezultirati negativnim utjecajem.

3.2 VRSTE ZNAČAJKI

Jedna od podjela lokalnih značajki bi mogla biti sljedeća:

1. Kutne (engl. *corners*)

Pojam kutne značajke se odnosi na točke u slici koje imaju veliku zakrivljenost (engl. *curvature*). Takve točke nisu nužno i projekcije pravih kuteva iz 3D svijeta pa se one recimo mogu detektirati i na raznim spojevima ili čak teksturiranim površinama. No, za većinu primjena to nije niti bitno pošto je cilj dobiti dobru značajku koja će biti robusna i visoko ponovljiva.

2. Grudaste (engl. *blobs*)

Grudaste značajke se za razliku od kutnih odnose na dijelove slike a ne na same točke. Detektiraju regije slike koje su recimo po osvjetljenju različite od okolnih.

3. Regionalne (engl. *regions*)

Značajke koje su, slično kao i grudaste, fokusirane na izdvajanje određenih regija slike.

3.3 DETEKTORI ZNAČAJKI (ENGL. *FEATURE DETECTORS*)

Ovo poglavlje će predstaviti tri osnovne grupe detektora prijespomenutih vrsta lokalnih značajki.

3.3.1 KUTNI DETEKTORI (ENGL. *CORNER DETECTORS*)

Niz kutnih detektora različitih matematičkih modela je predloženo u raznim radovima. Stoga će ovo poglavlje opisati par reprezentativnih metoda koje se najčešće koriste. Metode osim što se razlikuju u načinu ekstrakcije lokalnih značajki (gradijenti, morfologija, geometrija), razlikuju se i po razini invarijantnosti (translacija, rotacija, skala, afine transformacije).

3.3.1.1 HARRISOV DETEKTOR

Harrisov detektor značajki predložen je u radu Harrisa i Stephensa (Harris, C. et al., 1988). Temelji se na autokorelacijskoj matrici koja se neovisno o vrsti značajke često koristi i u drugim detektorima. Ona opisuje gradijentnu distribuciju u susjedstvu oko promatrane točke (x,y) :

$$M = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} I_x^2(x, \sigma_D) & I_x(x, \sigma_D)I_y(x, \sigma_D) \\ I_x(x, \sigma_D)I_y(x, \sigma_D) & I_y^2(x, \sigma_D) \end{bmatrix} \quad (1.3)$$

$$I_x(x, \sigma_D) = \frac{\partial}{\partial x} g(\sigma_D) * I(x) \quad (1.4)$$

$$g(\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (1.5)$$

U točki (x,y) matrica gradijenata se izračunava koristeći Gaussov filtar diferencijske skale σ_D , nakon čega se gradijenti izgladuju također Gausovim filtrom ali druge, integracijske skale σ_I . Vlastite vrijednosti dobivene matrice M predstavljaju promjene signala u dva ortogonalna smjera

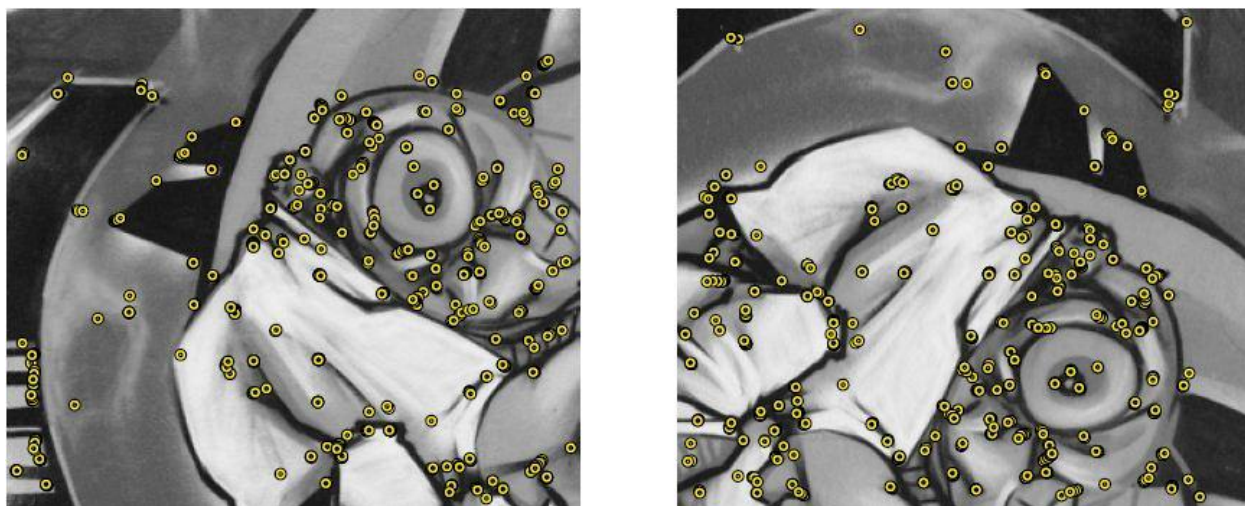
u susjedstvu oko točke koristeći skalu σ_I . Koristeći ovu činjenicu kutne značajke se tada mogu pronaći kao lokacije na slici čiji se smjerovi bitno razlikuju u oba smjera, ili drugim riječima, čije su obje vlastite vrijednosti dovoljno velike. Ovakvo saznanje Harris je iskoristio i predložio mjeru *kutnosti značajke* (engl. *cornerness*):

$$\text{cornerness} = \det(M) - \lambda * \text{trace}(M) \quad (1.6)$$

gdje je $\det(M)$ determinanta a $\text{trace}(M)$ trag matrice M . Tipična vrijednost za λ je 0.04.

Zapravo ovakav način izračuna bez direktnog pronalaženja vlastitih vrijednosti je računski manje zahtjevan a također daje jednaku informaciju o kutnosti značajke.

Značajke pronađene Harrisovim detektorom su invarijantne na translaciju i rotaciju a i stabilne su pod različitim svjetlosnim uvjetima. U studijama (Schmid, C. et al., 1998) i (Schmid, C. et al., 2000) pokazano je da ovakve značajke imaju najvišu ponovljivost i nose najviše informacija (karakterističnost).

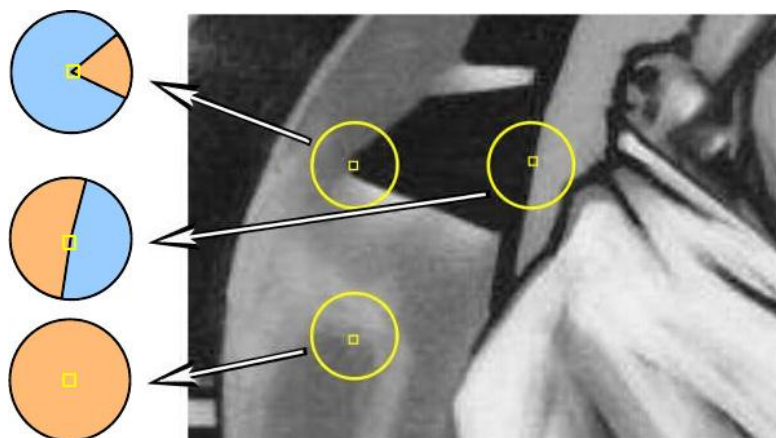


Slika 5 Harris kutne značajke na dvjema rotiranim slikama.

3.3.1.2 SUSAN DETEKTOR

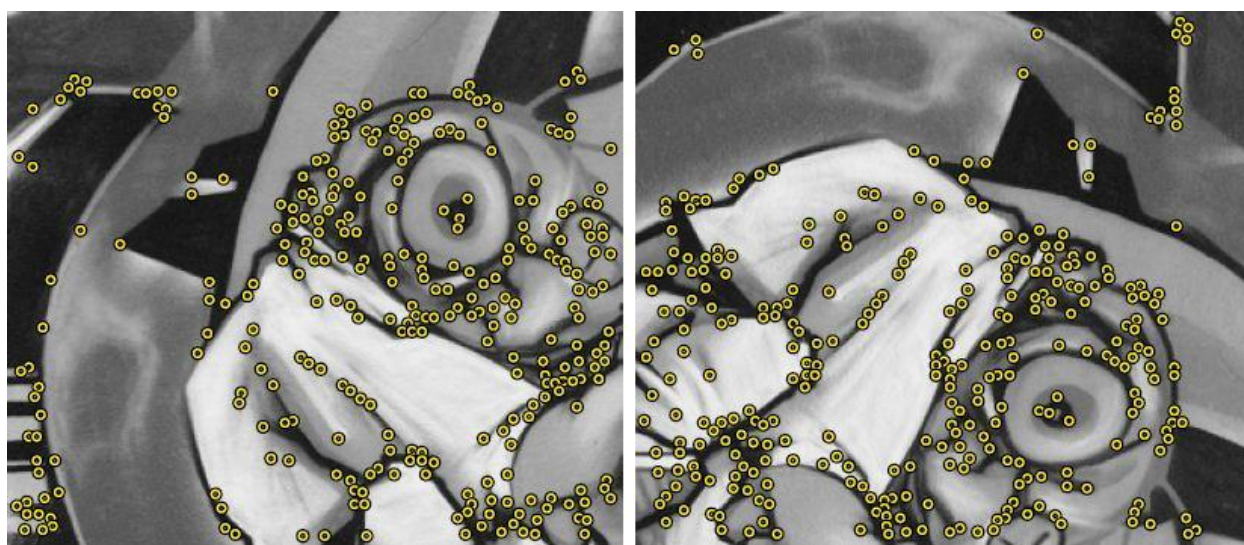
SUSAN detektor značajki predložen je u radu Smitha i Bradya (Smith, S.M. et al., 1997) i temelji se na morfološkom pristupu za razliku od prethodnog detektora koji je koristio lokalne gradijente. SUSAN zapravo znači *Smallest Univalued Segment Assimilating Nucleus* i metoda je koja se osim u detekciji kuteva, koristi i u rubnim detektorima te potiskivanju šuma. Ovaj detektor za svaki slikovni element definira kružno susjedstvo fiksnog radijusa. Takav slikovni element koji je centralan u svom susjedstvu se naziva *nucleus*, a njegov intenzitet se koristi kao referentni. Nakon toga se svi preostali slikovni elementi u susjedstvu dijele u dvije kategorije ovisno o tome da li je vrijednost njihovog intenziteta dovoljno ili nedovoljno slična referentnom. Na ovakav način svaki slikovni element sadrži informaciju o homogenosti njegovog kružnog

susjedstva. Primjerice područja slične svjetline će sadržavati gotovo sve točke iz kružnog susjedstva, dok će blizu rubova taj odnos pasti na 50%. Blizu kuteva taj odnos će biti oko 25%. Sve tri situacije su prikazane na sljedećoj slici.



Slika 6 SUSAN kutevi detektira segmentacijom kružnog susjedstva u slične (narančaste) i različite (plave) regije. Kutevi se nalaze na mjestima gdje relativna površina slične regije doseže lokalni minimum a pritom zadovoljava i predefimirani prag.

Kako bi se metoda učinila robusnijom, slikovni elementi koji su bliži centralnom elementu dobivaju veće težinske koeficijente. Također, kao i kod Harrisovog detektora dobivaju se značajke invarijantne na translaciju i rotaciju.

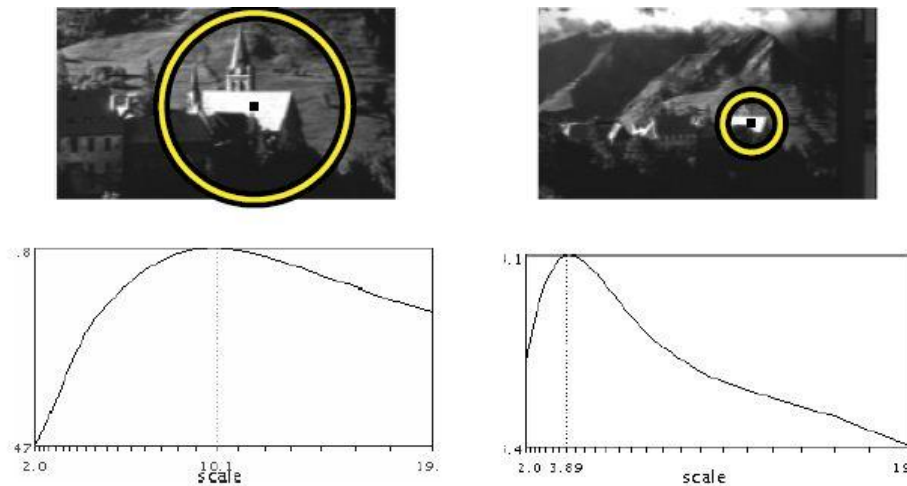


Slika 7 SUSAN kutne značajke na dvjema rotiranim slikama.

3.3.1.3 HARRIS-LAPLACE DETEKTOR

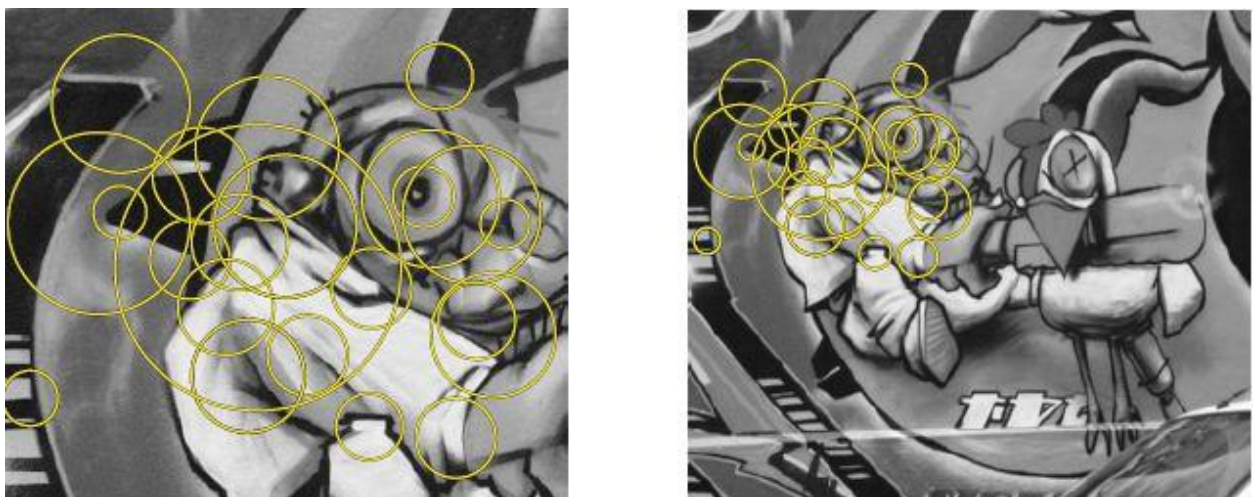
Kako su dosadašnji detektori bili invarijantni samo na translaciju i rotaciju, Mikolajczyk i Schmid su u svom radu (Mikolajczyk, K. et al., 2004) predložili Harrisov detektor invarijantan na skalu (*Harris-Laplace*) te na afine transformacije (*Harris-Affine*).

Harris-Laplace detektor temelji se na Harrisovom detektoru značajki koji pri detekciji koristi više skala (engl. *multiscale*). Karakteristična skala se određuje procesom kojeg je predložio Lindeberg u svojim radovima (Lindeberg, T. et al., 1993) i (Lindeberg, T. et al., 1998). Cilj je pronaći karakterističnu skalu za koju Laplaceov operator postiže ekstrem.



Slika 8 Primjer dviju karakterističnih skala na slikama snimljenim različitim povećanjem te odziv Laplaceovog operatora. Karakteristične skale su 10.1 i 3.9. Odnos tih dviju skala odgovara odnosu skala (povećanja) korištenih pri snimanju.

Na sljedećoj slici prikazan je rad ovog detektora. Vidljivo je da su pronađene iste značajke na slikama pri različitim skalama.



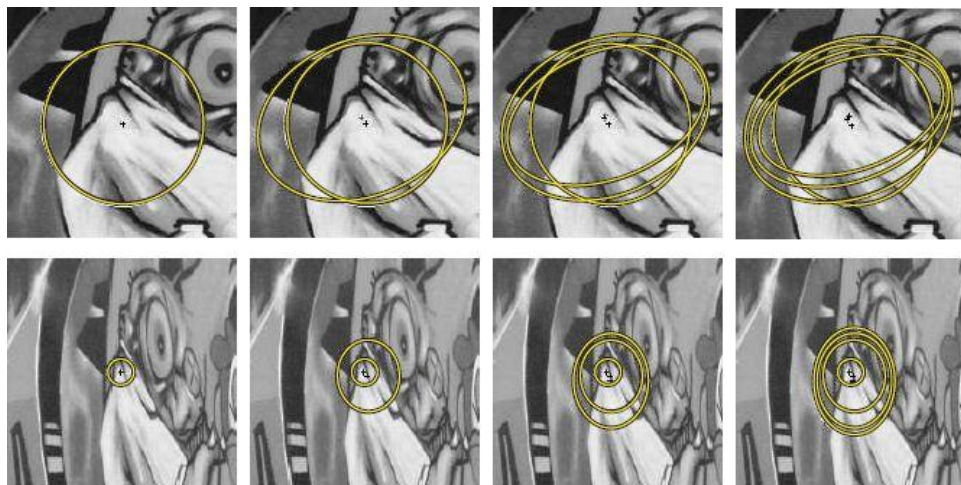
Slika 9 Iste značajke pronađene na slikama s različitim skalama. Radijus kružnice odgovara skali pri kojoj je značajka pronađena.

3.3.1.4 HARRIS-AFFINE DETEKTOR

Ovaj detektor se temelji na Harris-Laplaceovom detektoru ali koji pomoću iterativnog algoritma estimira eliptične affine regije (Lindeberg, T. et al., 1995) i (Lindeberg, T. et al., 1997) koje

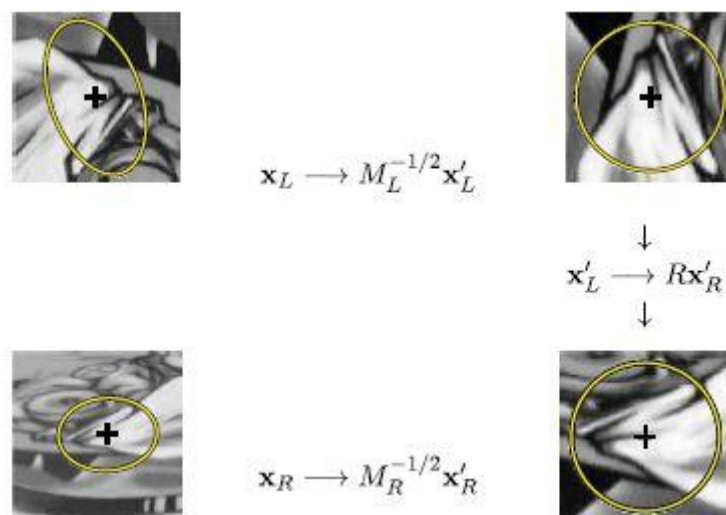
omogućavaju pronalazak kutnih značajki koje su invarijantne na affine transformacije. Proces se sastoji od sljedećih koraka:

1. Detektiraj inicijalnu regiju pomoću Harris-Laplace detektora.
2. Estimiraj afini oblik koristeći autokorelacijsku matricu.
3. Normaliziraj afinu regiju tako da bude kružna.
4. Ponovno detektiraj regiju na novoj normaliziranoj lokaciji i skali.
5. Vрати se na korak 2 ukoliko vlastite vrijednosti autokorelacijske matrice za novu značajku nisu jednake.



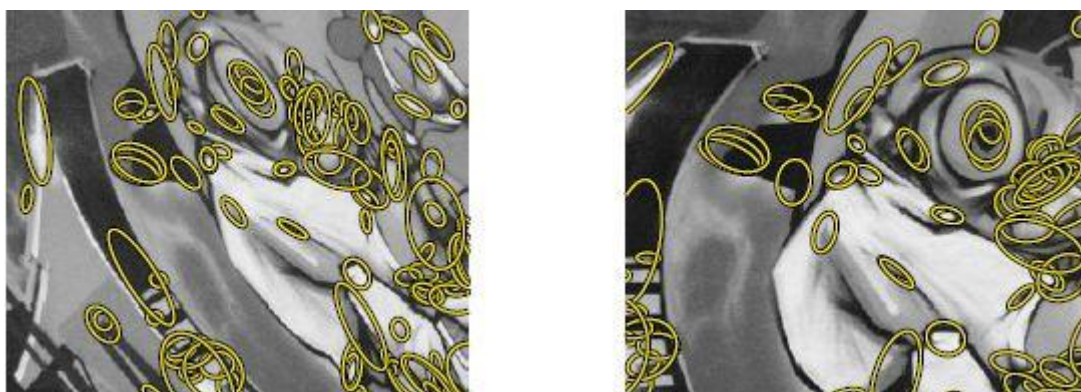
Slika 10 Iterativni postupak detekcije značajki invarijantnih na affine transformacije (gornji i donji red).

Vlastite vrijednosti autokorelacijske matrice M koriste se kao mjera afinosti oblika i to pomoću transformacije koja projicira uzorak intenziteta susjedstva na onaj s jednakim vlastitim vrijednostima. Transformacija koja se traži je zapravo drugi korijen od autokorelacijske matrice, $M^{0.5}$. Kao što je prikazano na sljedećoj slici, može se pokazati da ukoliko su dvije točke, X'_L i X'_R , povezane afinom transformacijom, tada će njihove normalizirane verzije, $X_L = M_L^{-0.5}X'_L$ i $X_R = M_R^{-0.5}X'_R$, biti povezane jednostavnom rotacijom $X'_L = RX'_R$.



Slika 11 Dijagram koji ilustrira normalizaciju koristeći autokorelacijsku matricu.

Sljedeća slika prikazuje rezultat Harris-Affine detektora značajki. Ove značajke su invarijantne na translaciju, rotaciju, skalnu i affine transformacije.

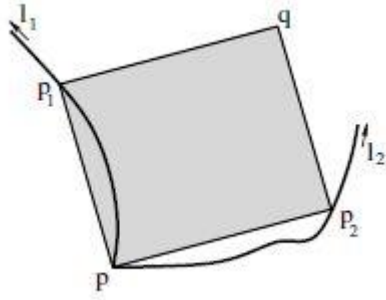


Slika 12 Dobro detektirane značajke unatoč afinoj transformaciji.

3.3.1.5 REGIJE TEMELJENE NA RUBOVIMA

Tuytelaars i Van Gool su u svojim radovima (Tuytelaars, T. et al., 1999) i (Tuytelaars, T. et al., 2004) predložili heurističku tehniku koja iskorištava geometriju rubova koji se obično nalaze u neposrednoj blizini Harrisove kutne značajke. Na taj način se također postiže afina invarijantnost. Motivacija u korištenju rubova leži u činjenici što su rubovi bez obzira na promjene skale, točke gledišta ili iluminacije i daje vrlo stabilne značajke. Nadalje, koristeći geometriju rubova, dimenzionalnost problema se može značajno smanjiti.

Detektor započinje od Harrisove značajke \mathbf{p} i obližnjeg ruba koji se pronade pomoću Cannyevog rubnog detektora (Harris, C. et al., 1988). Kako bi se povećala robusnost i na promjene u skali, značajke se ekstrahiraju na višestrukim skalama. Dvije takve točke $\mathbf{p1}$ i $\mathbf{p2}$ kreću se od kutne značajke \mathbf{p} u oba smjera duž ruba, kao što je i prikazano na sljedećoj slici. Na taj način detektor dodatno iskorištava i informaciju o geometriji ruba.



Slika 13 Kretanje točaka \mathbf{p}_1 i \mathbf{p}_2 od centralne točke \mathbf{p} .

Njihova relativna brzina u sprezi je s invarijantnim afinim parametrima l_1 i l_2 :

$$l_i = \int abs(|\mathbf{p}_i^{(1)}(s_i)\mathbf{p} - \mathbf{p}_i(s_i)|) ds_i \quad (1.7)$$

gdje je s_i proizvoljan parametar krivulje, $\mathbf{p}_i^{(1)}(s_i)$ prva derivacija $\mathbf{p}_i(s_i)$ a $|\dots|$ determinanta. Ovaj uvjet propisuje da površine između spoja $\langle \mathbf{p}, \mathbf{p}_1 \rangle$ i ruba te spoja $\langle \mathbf{p}, \mathbf{p}_2 \rangle$ i ruba ostaju jednake. Prema tome, $l = l_1 = l_2$.

Za svaku vrijednost l , točke $\mathbf{p}_1(l)$ i $\mathbf{p}_2(l)$ zajedno s kutom \mathbf{p} definiraju paralelogram $\Omega(l)$. Paralelogram, kao što prikazuje sljedeća slika, razapet je s vektorima $\mathbf{p}_1(l) - \mathbf{p}$ i $\mathbf{p}_2(l) - \mathbf{p}$. Kako postoji više od jedne vrijednosti l , dobiva se 1D familija regija koje su oblika paralelograma. Iz te familije regija odabire se jedan (ili više) paralelogram(a) za koje sljedeće fotometričke¹ veličine dostižu ekstrem:

$$Inv_1 = abs\left(\frac{|p_1 - p_g \quad p_2 - p_g|}{|p - p_1 \quad p - p_2|}\right) \frac{M_{00}^1}{\sqrt{M_{00}^2 M_{00}^0 - (M_{00}^1)^2}} \quad (1.8)$$

$$Inv_2 = abs\left(\frac{|p - p_g \quad q - p_g|}{|p - p_1 \quad p - p_2|}\right) \frac{M_{00}^1}{\sqrt{M_{00}^2 M_{00}^0 - (M_{00}^1)^2}} \quad (1.9)$$

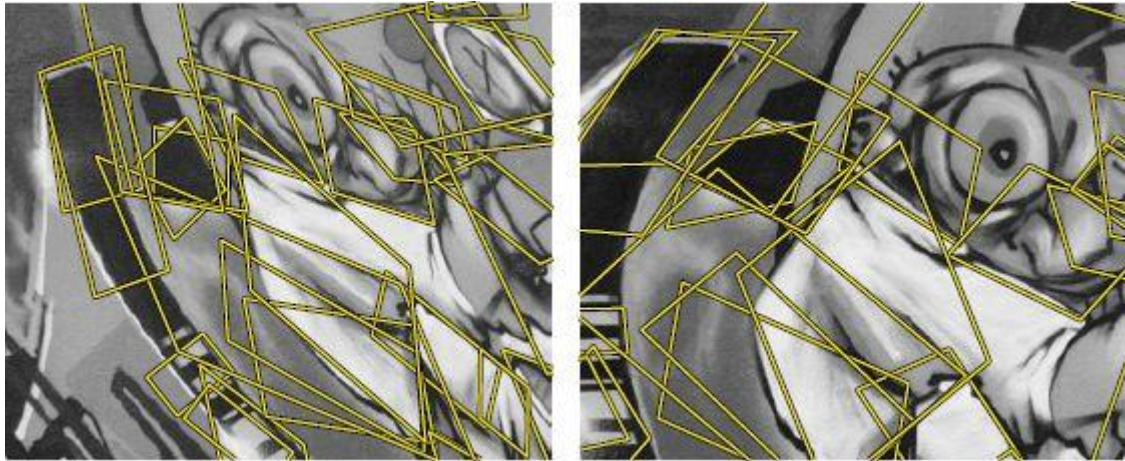
$$M_{pq}^n = \int_{\Omega} I^n(x,y) x^p y^q dx dy \quad (1.10)$$

$$p_g = \left(\frac{M_{10}^1}{M_{00}^1}, \frac{M_{01}^1}{M_{00}^1}\right) \quad (1.11)$$

gdje je M_{pq}^n moment n -tog reda i $(p+q)$ -tog stupnja, p_g centar gravitacije regije težinske vrijednosti $I(x, y)$, i q kut paralelograma suprotnog od kuta p .

Rezultat ovakvog detektora se može vidjeti na sljedećoj slici.

¹ Fotometrija – znanost koja se bavi proučavanjem mjerenja količine opažene svjetlosti koje prima ljudsko oko.



Slika 14 Detektirane regije u obliku paralelograma na dvjema slikama.

3.3.2 GRUDASTI DETEKTORI (ENGL. *BLOB DETECTORS*)

Nakon traženja kutnih točaka sljedeća najintuitivnija lokalna značajka su grudice (engl. *blobs*). U ovom poglavlju opisano je par takvih detektora koje po pristupu detekciji možemo podijeliti u dvije grupe. Prvi i uobičajeni pristup su opet metode zasnovane na derivacijama te njihove ekstenzije u vidu razine invarijantnosti. Drugi pristup se zasniva na entropiji vjerojatnosne distribucije intenziteta. Takvi detektori pronalaze ispučene regije (engl. *salient regions*).

3.3.2.1 HESSIAN DETEKTOR

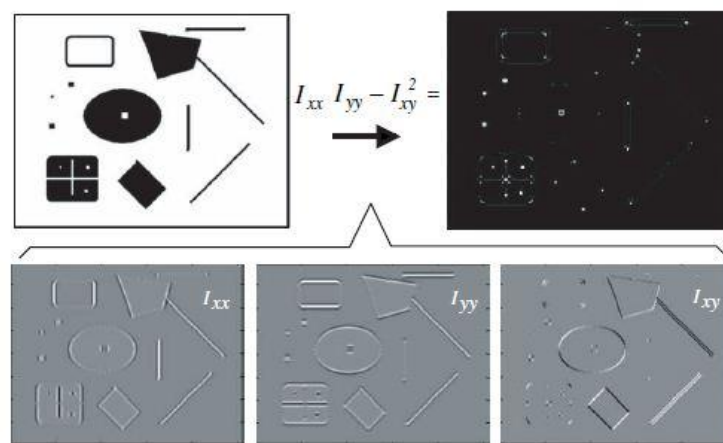
Ovaj detektor koristi 2×2 Hessianovu matricu dobivenu pomoću slikovnih intenziteta $I(x, y)$:

$$H = \begin{bmatrix} I_{xx}(x, \sigma_D) & I_{xy}(x, \sigma_D) \\ I_{xy}(x, \sigma_D) & I_{yy}(x, \sigma_D) \end{bmatrix} \quad (1.12)$$

gdje je svaki član matrice derivacija drugog reda zaglađena Gausovim filtrom veličine σ_D .

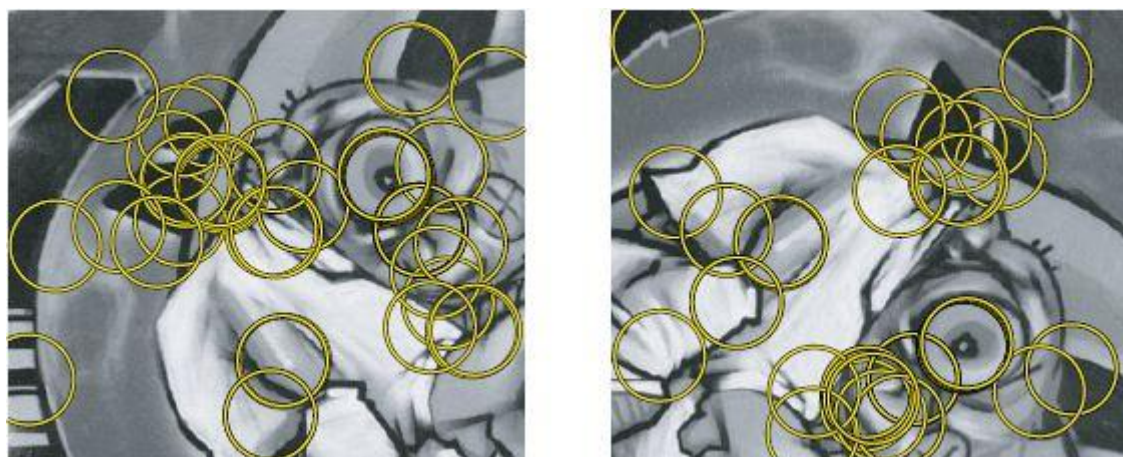
Determinanta i trag ovakve matrice često se koriste u raznoraznim filtrima. Vjerojatno najpoznatiji je Laplaceov filter koji se koristi u detekciji grudastih značajki. Međutim, nedostatak je da detektira i značajke koje nisu dovoljno stabilne. Primjerice lokalni maksimum se često nalazi u blizini kontura ili ravnih linija, što znači da je promjena signala samo u jednom smjeru a to maksimum čini više osjetljivim na šum te na manje promjene susjednih slikovnih elemenata. Jedno od rješenja je odabrati lokaciju i skalu značajke za koju trag i determinanta Hessianove matrice također daje lokalni maksimum.

Primjer detekcije značajki dan je na sljedećoj slici. Prvo se izračunaju derivacije drugog reda zaglađene originalne slike (gore lijevo) pomoću Gausovog filtra (donji red). Nakon toga se pomoću njih izračuna determinanta Hessianove matrice (gore desno).



Slika 15 Ilustracija komponenti Hessianove matrice.

Pošto su derivacije drugog reda simetrični filtri, odziv će biti slab točno u točkama gdje je promjena signala najveća. Prema tome, maksimumi će biti pronađeni na grudicama čija se veličina podudara s veličinom Gaussove jezgre σ_D . Primjer detekcije grudastih značajki pomoću Hessian detektora predložen je na sljedećoj slici.

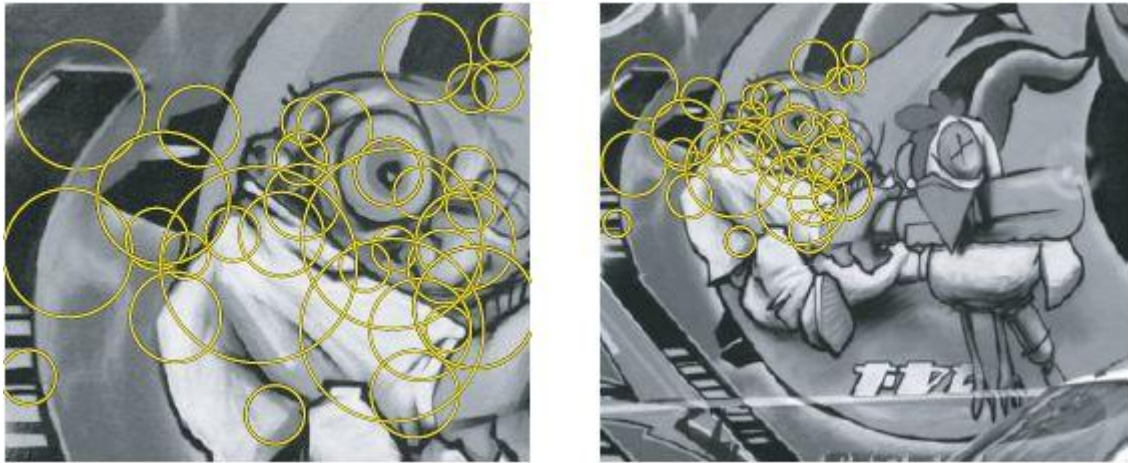


Slika 16 Rezultat Hessian detekcije na dvjema rotiranim slikama.

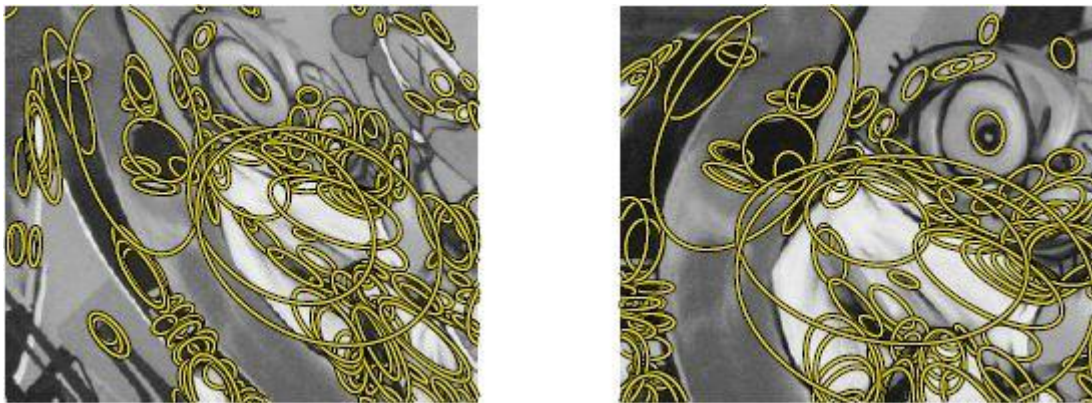
Također je bitno napomenuti da Laplaceov filtar ima svojstvo separabilnosti te se zbog bolje efikasnosti može aproksimirati pomoću *DoG* filtra (engl. *Difference of Gaussians*).

3.3.2.2 HESSIAN-LAPLACE I HESSIAN-AFFINE DETEKTOR

Također kao i kod Harrisovih inačica detektora invarijantnih na skalu i na affine transformacije, Mikolajczyk i Schmid su u svom radu (Mikolajczyk, K. et al., 2004) predložili iste metode za ovu vrstu značajki. Prema tome broj detektiranih regija se može također kontrolirati postavljanjem praga na determinantu Hessianove matrice kao i na odziv Laplaceovog filtra. U pravilu, ovakvi detektori pronalaze velik broj značajki. Rezultat detektora dan je na sljedeće dvije slike.



Slika 17 Hessian-Laplace detektor daje značajke invarijantne na promjenu skale.



Slika 18 Hessian-Affine detektor daje značajke invarijantne i na affine transformacije.

3.3.2.3 DETEKTOR ISPUČENIH REGIJA (ENGL. SALIENT REGIONS)

Kadir i Brady su u svom radu (Kadir, T. et al., 2001) predložili novu metodu detekcije grudastih značajki koja za razliku od prethodnih motivaciju pronalazi u teoriji informacija. Glavna ideja je pronaći ispučene regije, gdje se ispučenost mjeri pomoću entropije vjerojatnosne distribucije intenziteta. Međutim, sama entropija ne daje dovoljno informacija kako bi se precizno locirale značajke pa se prema tome koristi dodatni kriterij – *vlastita raznolikost* (engl. *self-dissimilarity*).

Detekcija značajki izvodi se u dva koraka:

1. Za svaki slikovni element x kroz niz skala s izračunaj entropiju H vjerojatnosne distribucije $p(I)$:

$$H = - \sum_I p(I) \log p(I) \quad (1.13)$$

2. Estimacija vjerojatnosne distribucije se temelji na distribuciji intenziteta u kružnom susjedstvu radijusa s oko slikovnog elementa x .

Nakon toga se zabilježe lokalni maksimumi entropija i te regije postaju kandidati za značajku.

3. Za svakog kandidata izračunaj magnitudu od derivacije vjerojatnosne distribucije $p(I)$ koristeći skalu s :

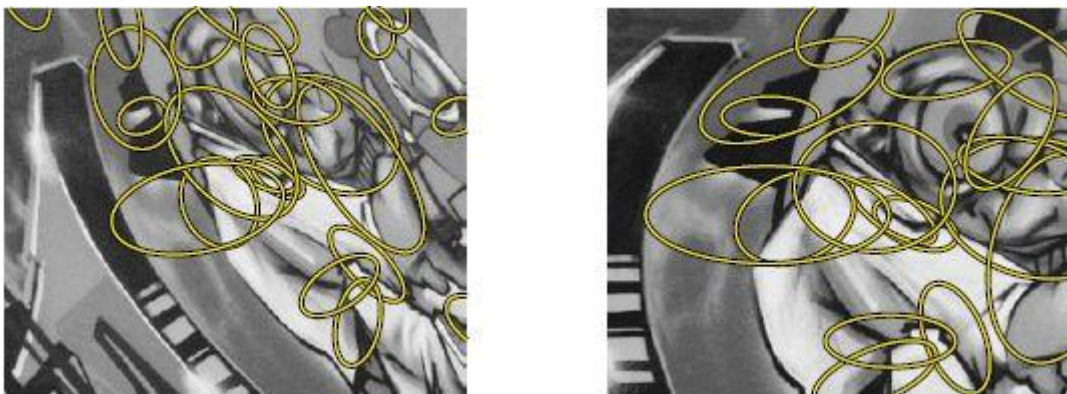
$$W = \frac{s^2}{2s-1} \sum_I \left| \frac{\partial p(I;s)}{\partial s} \right| \quad (1.14)$$

4. Ispupčenost se sada može proračunati koristeći sljedeći izraz:

$$Y = WH \quad (1.15)$$

Broj značajki ekstrahiran ovom metodom je relativno nizak ali pronalazi svoju primjenu u prepoznavanju objekata zbog njihove stabilnosti.

Također u radu (Kadir, T. et al., 2004) je predložena verzija koja je invarijantna na affine transformacije.



Slika 19 Pronađene ispupčene regije invarijantne na affine transformacije.

3.3.3 DETEKTORI REGIJA (ENGL. *REGION DETECTORS*)

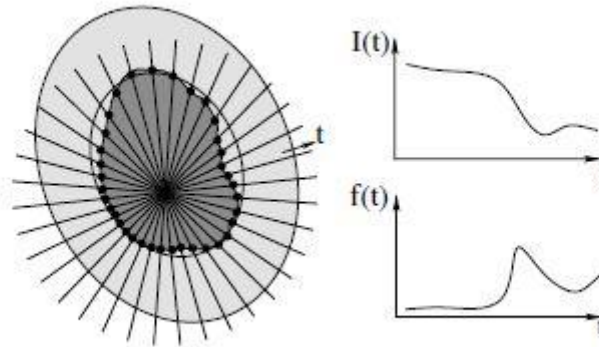
Ovo poglavlje opisuje detektore značajki koji za razliku od prethodnih ekstrahiraju regije slike kao značajke. Ovakvi detektori pronalaze regije na temelju intenziteta ili tražeći regije na temelju ekstrema koji moraju zadovoljavati određen uvjet stabilnosti.

3.3.3.1 DETEKTOR REGIJA NA TEMELJU INTENZITETA

Tuytelaars i Van Gool su u svojim radovima (Tuytelaars, T. et al., 2000) i (Tuytelaars, T. et al., 2004) predložili detektor koji pronalazi regije koje su invarijantne na affine transformacije. Detektor započinje tražeći ekstreme kroz višestruke skale te oko njih radijalno ispituje funkciju intenziteta:

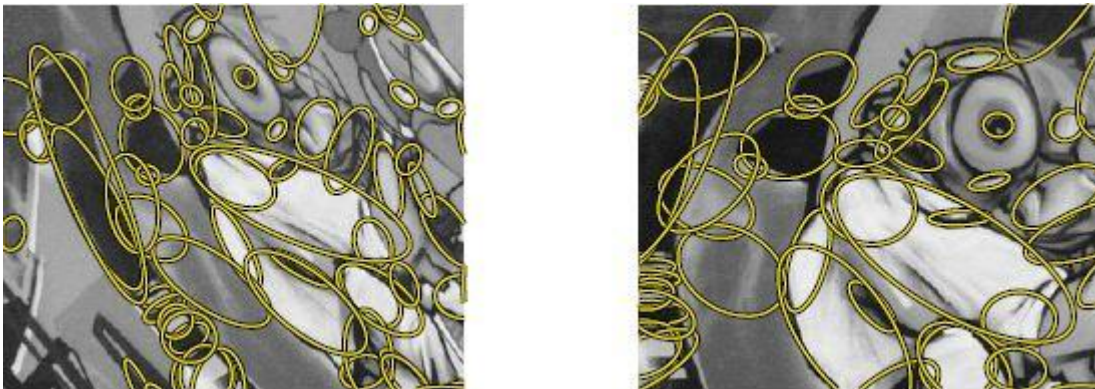
$$f(t) = \frac{\text{abs}(I(t) - I_0)}{\max\left(\frac{\int_0^t \text{abs}(I(t) - I_0) dt}{t}, d\right)} \quad (1.16)$$

gdje je t proizvoljni parametar duž radijalne zrake, $I(t)$ intenzitet na mjestu t , I_0 vrijednost intenziteta na mjestu ekstrema te d mali broj koji se dodaje kako bi se spriječilo dijeljenje s nulom.



Slika 20 Ilustracija detektora regija na temelju intenziteta

Točka u kojoj ova funkcija postiže ekstrem je invarijantna na afine i linearne fotometričke transformacije. Prema tome detektor duž zraka traži sve točke ekstreme koje pripadaju istom lokalnom ekstremu I_0 . Kao što je prikazano i na slici nakon toga sve pronađene točke povezuju se međusobno kako bi se dobila regija invarijantna na afine transformacije. Pošto se ovim načinom uglavnom dobivaju nepravilne regije, one se zamjenjuju s elipsastim regijama.



Slika 21 Rezultat detektora regija temeljenog na intenzitetima.

3.3.3.2 DETEKTOR REGIJA MAKSIMALNO STABILNIH EKSTREMA (ENGL. MAXIMALLY STABLE EXTREMAL REGIONS)

Matas je u svom radu (Matas, J. et al., 2002) predložio detektor MSER koji je analogan radu algoritma povezanih komponenti (engl. *connected components*) nad slikom na koju je primjenjen

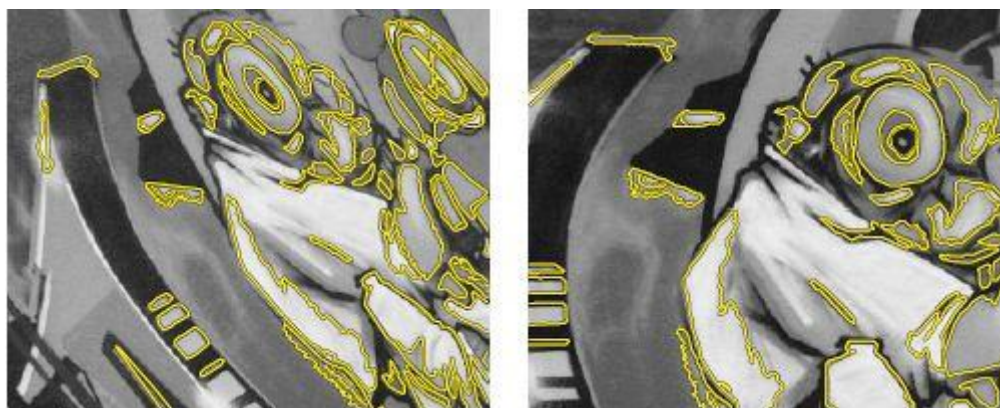
određeni prag. Detektor pronalazi regije ekstrema koje su maksimalno stabilne u kontekstu praga (engl. *threshold*).

Skup regija ekstrema ε se pronalazi gotovo linearno ovisno o broju slikovnih elemenata. Koraci su sljedeći:

1. Sortiraj slikovne elemente po intenzitetima.
2. Sortirane slikovne elemente označi na slici.
3. Pomoću *Union-Find* algoritma pronađi povezane komponente te spremi njihove površine kao funkcije intenziteta.

Kako bi se među pronađenim regijama, tj. povezanim komponentama, pronašle one koje zadržavaju stabilnost ovisno o pragu, detektor prolazi kroz niz pragova i zaključuje koje od njih ima to svojstvo. Prema tome, ne pronalazi se globalni ili optimalni prag već se stabilnost regija testira koristeći niz pragova. Ovakve regije su invarijantne na afine transformacije.

Ova metoda uglavnom najbolje rezultate daje na strukturiranim slikama koje se daju dobro segmentirati. Također je relativno brza te je trenutno od svih detektora invarijantnih na afine transformacije najbrža. Nedostatak je osjetljivost na zamučivanje slike što je i razumljivo pošto narušava kriterij stabilnosti. Uglavnom svoju primjenu nalazi u prepoznavanju objekata.



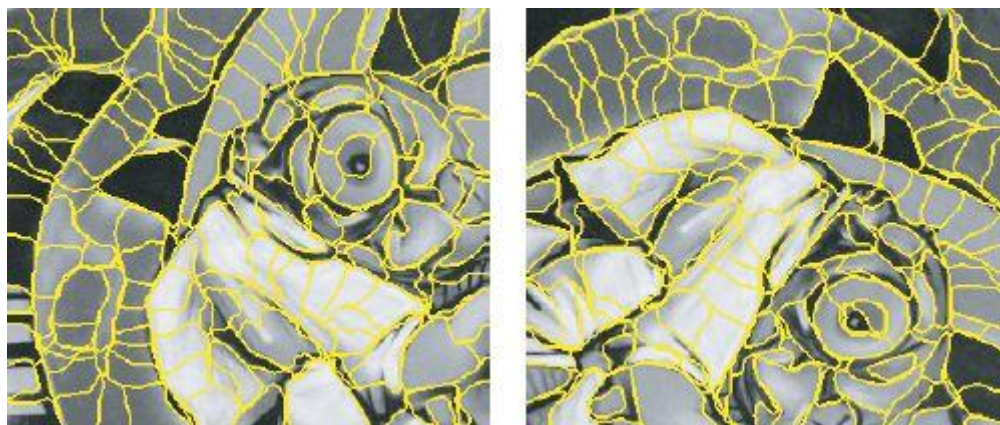
Slika 22 Regije pronađene MSER detektorom.

3.3.3.3 SUPERPIXELS DETEKTOR

Imajući na umu činjenicu da se tradicionalnim tehnikama segmentacije dobivaju regije koje su jednostavno prevelike da bi se mogle koristiti kao značajke, Mori (Mori, G. et al., 2004) te Ren i Malik (Ren, X. et al., 2005) u svojim radovima predlažu metodu koja slikovne elemente grupira u manje atomarne regije – superpiksele (engl. *superpixels*).

Algoritam pronalazi superpiksele koristeći metodu normaliziranih rezova (Shi, J. et al., 2000) koja pomoću informacija o svjetlini, boji, teksturi pronalazi regije koje su dovoljne *dobre* (engl. *goodness criterion*).

U suprotnosti s ostalim metodama, regije pronađene ovim detektorom preklapaju cijelu sliku i nikada se ne preklapaju. Sve pronađene regije također imaju slične skale pa prema tome metoda nije invarijantna na skalu. Isto tako zbog uniformnosti regija značajke se ne mogu dobro razlikovati a i njihova ponovljivost je niska. Prema tome primjenu ne pronalazi u prepoznavanju objekata već u semantičkoj segmentaciji slike.



Slika 23 Superpiksel regije.

4 EFIKASNE IMPLEMENTACIJE EKSTRAKCIJE ZNAČAJKI

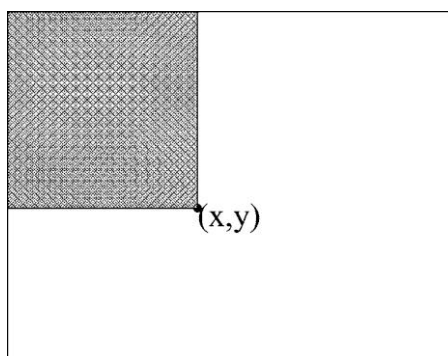
Većina opisanih detektora značajki uključuju izračune gradijenata, derivacija, autokorelacijskih matrica, entropija i sl. Pošto detektor takav izračun ponavlja za svaki slikovni element a većinom i na raznim skalama, tada takav proračun postaje računalno skup odnosno neprigodan za većinu primjena. Ovo poglavlje donosi par detektora značajki koji u obzir uzimaju i efikasnost izvršavanja.

4.1 SURF (ENGL. *SPEEDED UP ROBUST FEATURES*)

Bay je u svojim radovima (Bay, H. et al., 2006) i (Bay, H. et al., 2008) predložio detektor značajki temeljen na Hessian-Laplaceovom detektoru invarijantnim na skale. Međutim, kako bi izbjegao direktno računanje Hessianove matrice, Bay je iskoristio aproksimaciju koristeći *integralne slike* (engl. *integral images*). Integralne slike prvi su koristili Viola i Jones u svom radu koji se bavio detekcijom lica u realnom vremenu. Općenito, integralne slike omogućuju vrlo brz izračun Haarovih valića (engl. *Haar wavelets*) ali i bilo kojeg konvolucijskog filtra koji se temelji na usrednjavanju susjeda (engl. *box-type convolution filter*).

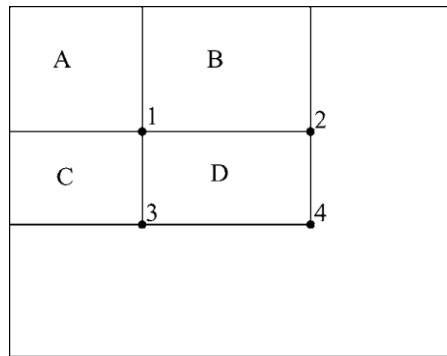
Integralna slika $I_{\Sigma}(x, y)$ predstavlja zbroj intenziteta svih slikovnih elemenata koji se nalaze gore lijevo od $I(x, y)$:

$$I_{\Sigma}(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(x, y) \quad (1.17)$$



Slika 24 Vrijednost integralne slike u točki (x, y) je zbroj svih intenziteta na gore lijevo.

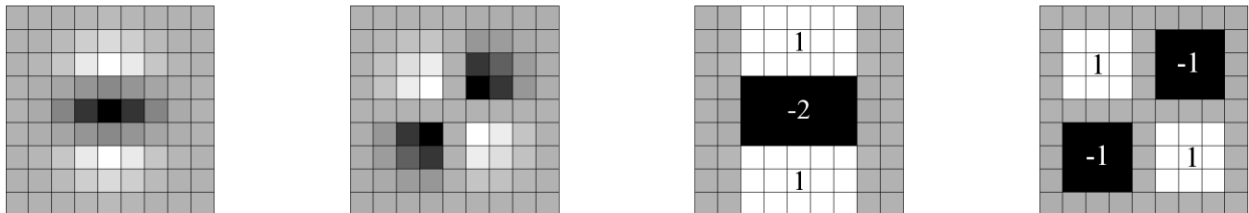
Imajući integralnu sliku koristeći četiri operacije zbrajanja moguće je izračunati zbroj intenziteta unutar bilo kojeg pravokutnika, kao što je i prikazano na sljedećoj slici.



Slika 25 Suma intenziteta unutar pravokutnika D može se izračunati prema točkama:

$$4 + 1 - (2 + 3)$$

Druga aproksimacija koja se koristi u ovom detektoru su filtri s maskama za usrednjavanje koji se koriste zbog artefakata koji se javljaju uslijed diskretizacije Gaussovih parcijalnih derivacija drugog reda. Brz izračun omogućuju integralne slike.



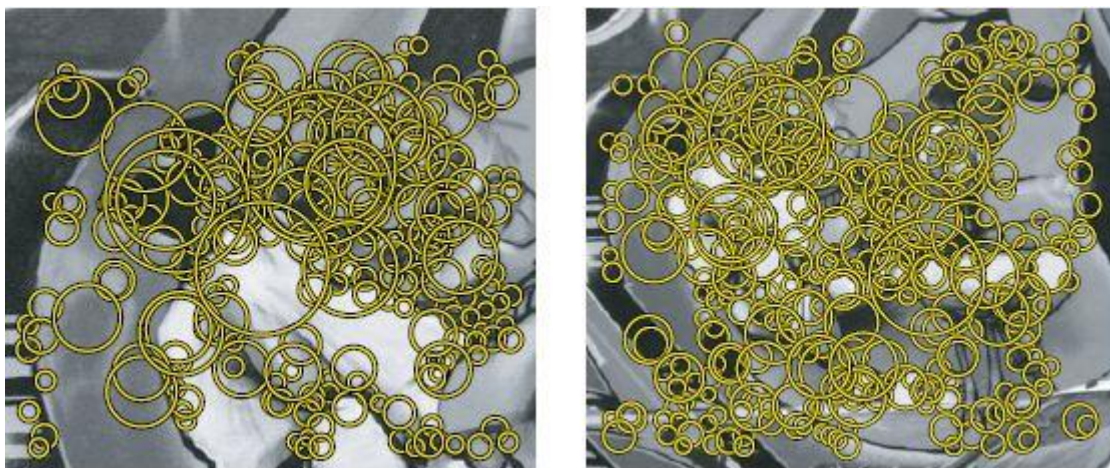
Slika 26 Lijevo: Diskretizirane Gaussove parcijalne derivacije drugog reda po y i po xy smjeru. Desno: Njihove aproksimacije pomoću filtara.

Koriste se filtri veličine 9×9 koji aproksimiraju Gausa s parametrom $\sigma = 1.2$ koji predstavlja najvišu prostornu rezoluciju.

Aproksimacija determinante SURF detektor čini prema sljedećoj formuli:

$$\det(H_{approx}) = D_{xx}D_{yy} + (0.6D_{xy})^2 \quad (1.18)$$

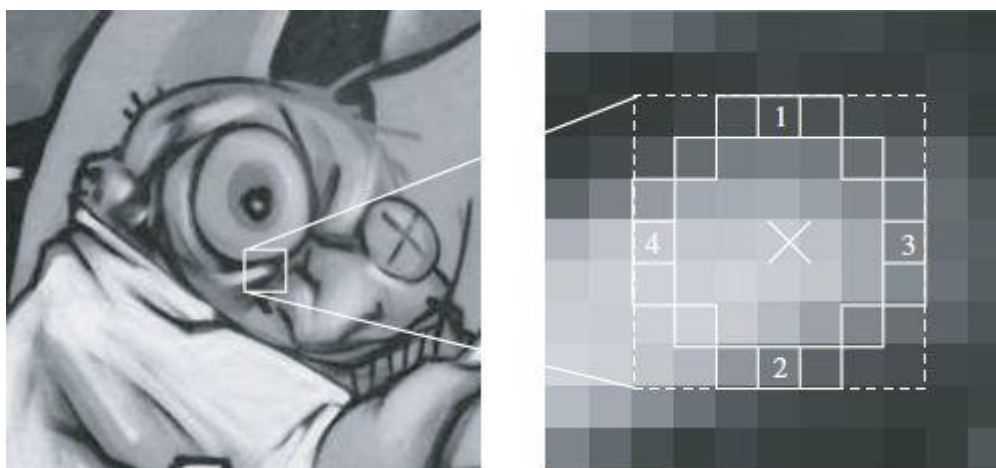
gdje su D_{xx} , D_{xy} i D_{yy} aproksimacijski filtri a 0.6 težinska vrijednost. Vrijednost ove determinante predstavlja odziv grudaste značajke na lokaciji x . Odzivi se spremaju a pri traženju lokalnih maksimuma koristi i kvadratna interpolacija.



Slika 27 SURF grudaste značajke.

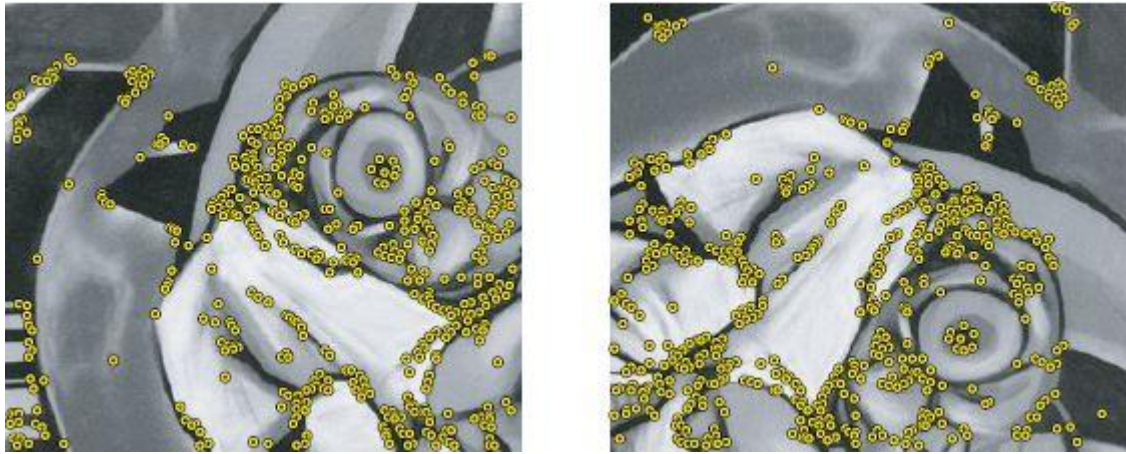
4.2 FAST (ENGL. FEATURES FROM ACCELERATED SEGMENT TEST)

Rosten i Drummond su u svojim radovima (Rosten, E. et al., 2005) i (Rosten, E. et al., 2006) predložili novi detektor koji se temelji na već poznatom SUSAN detektoru značajki. FAST koristi sličnu ideju pa svaki slikovni element uspoređuje sa svojim 16-susjedstvom. Prvo elemente na pozicijama 1 i 2 uspoređuje sa zadanim pragom a zatim i elemente 3 i 4. Slikovni elementi se nakon toga, kao i kod SUSAN detektora, grupiraju u dvije skupine ovisno o sličnosti s referentnim intenzitetom. Tada se pomoću ID3 stabala (Quinlan, J.R. et al., 1986) donosi odluka o tome da li se radi o elementu koji je kut. Na kraju se primjenjuje potiskivanje ne-maksimuma (engl. *non-maxima suppression*) na sumu apsolutnih razlika elemenata unutar 16-susjedstva i centralnog elementa.



Slika 28 Usporedba sa 16-susjedstvom.

FAST detektor je brži od SURF detektora oko 6 puta iako nije invarijantan na skalu. Invarijantnost na skalu predložena je u radu (Lepetit, V. et al., 2006) gdje se koristi aproksimacija Laplaceove funkcije.



Slika 29 Značajke detektirane FAST detektorom.

4.3 PREGLED KARAKTERISTIKA DETEKTORA

Sljedeća tablica daje pregled najvažnijih karakteristika spomenutih detektora značajki.

Tabela 1 Pregled karakteristika detektora značajki.

	Harris	Hessian	SUSAN	Harris-Laplace	Hessian-Laplace	SURF	Harris-Affine	Hessian-Affine	Salient regions	Edge-based	MSER	Intensity-based	Superpixels
<i>Kutne z.</i>	✓		✓	✓	(✓)	(✓)	✓	(✓)	(✓)	✓			
<i>Grudaste z.</i>		✓		(✓)	✓	✓	✓	(✓)	✓	✓			
<i>Regionalne z.</i>											✓	✓	✓
<i>Invarijantnost na translaciju</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Invarijantnost na rotaciju</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Invarijantnost na skalu</i>				✓	✓	✓	✓	✓	✓	✓	✓	✓	(✓)
<i>Invarijantnost na afine t.</i>							✓	✓	(✓)	✓	✓	✓	(✓)
<i>Ponovljivost</i>	+++	++	++	+++	+++	++	+++	+++	+	+++	+++	++	+
<i>Lokalizacija</i>	+++	++	++	+++	+++	++	+++	+++	+	+++	+++	++	+
<i>Robusnost</i>	+++	++	++	++	+++	++	++	+++	++	+	++	++	+
<i>Efikasnost</i>	++	+	+++	+	+	+++	++	++	+	+	+++	++	+

5 SIFT (ENGL. SCALE INVARIANT FEATURE TRANSFORM)

SIFT je metoda za ekstrakciju invarijantnih značajki koja daje odlične rezultate pri traženju korespondentnih točaka između dva različita pogleda na objekt ili scenu. Značajke koje generira su invarijantne na skalu i rotaciju a djelomično i na afine transformacije te varijacije u osvjetljenju i promjene položaja kamere. Njihova karakterističnost doprinosi još većoj vjerojatnosti da će se korespondentne točke uspješno detektirati. Metoda je predložena je 1999. godine a autor je David G. Lowe.

Prednost ove metode je u tome što generira velik broj značajki. Primjerice na slici veličine 500x500 detektirat će se oko 2000 stabilnih značajki, iako to naravno ovisi i o sadržaju slike te raznim podešenjima samog algoritma. U traženju korespondentnih točaka i prepoznavanju objekata kvantiteta značajki je vrlo bitna. Iako je broj značajki velik, trošak procesa detekcije je minimiziran zahvaljujući kaskadnom pristupu u kojem se računalno zahtjevnije operacije primjenjuju samo na lokacijama koje su prošle inicijalne korake.

Koraci algoritma su sljedeći:

- 1. Detekcija ekstrema kroz niz skala (engl. *Scale-space extrema detection*)**

Prvi korak algoritma koristeći razlike Gausovim filtrom zamućenih slika kao aproksimaciju Laplaceovog rubnog operatora pronalazi ekstreme koji postaju točke od interesa (engl. *interest keypoints*).

- 2. Lokalizacija značajki**

Za svaki ekstrem interpolacijom se izračunava još točnija lokacija i skala na kojem je pronađen te se provjerava uvjet stabilnosti.

- 3. Dodjeljivanje orijentacija**

Računajući gradijente jedna ili više orijentacija se dodjeljuje svakoj značajki.

- 4. Izgradnja deskriptora**

Računajući gradijente oko svake značajke formira se deskriptor koji opisuje svaku značajku.

5.1 DETEKCIJA EKSTREMA KROZ NIZ SKALA

SIFT započinje tražeći točke u slikama koje bi zadovoljavale uvjete ponovljivosti, stabilnosti i invarijantnosti. Invarijantnost točaka na različite skale, kao što je već korišteno u dosada spomenutim detektorima značajki, se može postići koristeći detekciju na različitim skalama.

Slika na različitim skalama je zapravo slika konvoluirana Gausovim filtrom parametra σ . Tada je slika zapravo reprezentirana u 3D prostoru kao $L(x, y, \sigma)$:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1.19)$$

gdje je Gaussov filtar definiran kao

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1.20)$$



Slika 30 Slika s parametrom $\sigma = 0$.



Slika 31 Slika s parametrom $\sigma = 1$.



Slika 32 Slika s parametrom $\sigma = 4$.



Slika 33 Slika s parametrom $\sigma = 16$.

5.1.1 LoG vs. DoG

Kako se zapravo potencijalne značajke traže na slikama rubova intuitivno bi bilo upotrijebiti recimo Laplaceov rubni operator (engl. *Laplacian of Gaussian*; LoG). Međutim, on zahtjeva proračun drugih parcijalnih derivacija što bi moglo nepogodno djelovati na brzinu cjelokupnog SIFT algoritma pa iz tog razloga Lowe kao zamjenu za Laplaceov operator koristi njegovu aproksimaciju – razliku slika zamućenih s Gausovim filtrom (engl. *Difference of Gaussian*;

DoG). DoG je zapravo konvolucija originalne slike s razlikom dviju susjednih slika zamućenih Gaussovima s parametrima σ koji se razlikuju za neku konstantu k :

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (1.21)$$

Ovakva metoda je efikasna iz razloga što se na ovakav način računanje parcijalnih derivacija svodi na jednu operaciju oduzimanja dviju zamućenih slika koje se zbog ranijeg zahtjeva na invarijantnost na skalu ionako moraju izračunati.

Kako bi se pokazalo da je DoG zaista bliska aproksimacija LoG operatora koji je normaliziran po skali može se iskoristiti difuzna toplinska jednadžba koja je parametrizirana po σ :

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \quad (1.22)$$

Pošto se $\frac{\partial G}{\partial \sigma}$ može aproksimirati pomoću diferencija, tada jednadžba poprima sljedeći oblik:

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (1.23)$$

iz čega slijedi:

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G \quad (1.24)$$

Ovaj izraz dokazuje aproksimaciju pošto je član $(k - 1)$ konstantan kroz sve skale pa prema tome i ne utječe na rezultat dok je normaliziranost LoG operatora ($\sigma^2 \nabla^2 G$) ionako potrebna zbog postizanja invarijantnosti na skalu.

5.1.2 FORMIRANJE DoG PIRAMIDE

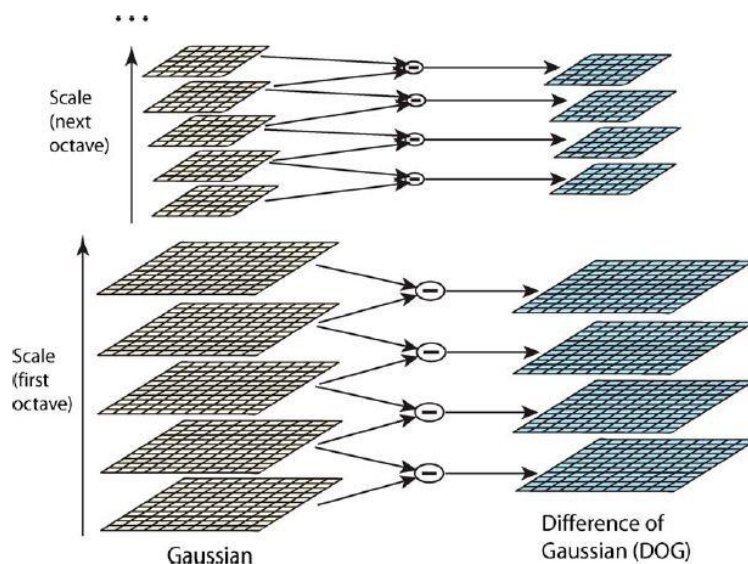
Kako se DoG računa preko razlike dviju slika zamućenih Gaussovima filtrom, potrebno je prvo napraviti piramidu takvih slika. Lowe svaku razinu piramide naziva *oktava* koju ujedno i dijeli na cjelobrojni broj intervala s . Svaka oktava piramide se formira na način da se ulazna slika oktave određen broj puta konvoluirala s Gaussovima filtrom parametra $k\sigma$ gdje pritom mora vrijediti $k = 2^{\frac{1}{s}}$. Prema tome svaka oktava zapravo predstavlja familiju skala (engl. *scale-space*). Broj slika zamućenih Gaussovima filtrom odnosno broj konvolucija unutar oktave mora biti najmanje $s + 3$ i to zbog detekcije značajki što će biti obrazloženo u sljedećem poglavlju.

Svakim prelaskom u novu oktavu, nova razina piramide zamućenih slika se kreira na način da se ponavlja postupak konvolucije s Gaussovima filtrom ali s novom ulaznom slikom. Nova ulazna slika je ona čija je skala σ dva puta veća od one inicijalne na toj razini piramide. Ta slika će

uvijek biti jedna od zadnje dvije slike iz prethodne oktave. Također, rezolucija novih ulaznih slika se uvijek smanjuje dva puta.

Sad, kad je izgrađena piramida zamućenih slika moguće je formirati *DoG* piramidu. Izgradnja je vrlo jednostavna i sastoji se samo od oduzimanja susjednih zamućenih slika.

Cijeli proces ilustriran je na sljedećoj slici.

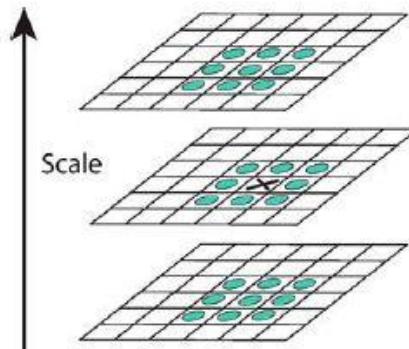


Slika 34 Izgradnja DoG piramide. Za svaku oktavu kreira se grupa slika zamućenih Gausovim filtrom (lijevo). Iz njih se oduzimanjem susjednih kreira *DoG* piramida (desno). Svakim prelaskom u novu oktavu nova ulazna slika postaje dva puta manja nego ona prethodna.

5.1.3 TRAŽENJE EKSTREMA

Mikolajczyk je u svom radu (Mikolajczyk, K. et al., 2002) došao do zaključka kako ekstremi u *DoG* slikama daju najstabilnije značajke od svih ostalih sličnih funkcija kao što su recimo gradijenti, Hessianove ili Harrisove matrice pa se prema tome oni i koriste u SIFT algoritmu.

Detekcija ekstrema koristi *26-susjedstvo*. Promatrani slikovni element osim što u svom 8-susjedstvu mora biti ekstrem, također mora biti i ekstrem u 9-susjedstvu *DoG* slika na susjednim razinama u oktavi. Ovaj korak objašnjava prethodno definiran uvjet od minimalno $s + 3$ slika po oktavi. Za $s = 0$ postojale bi točno 3 slike po oktavi što je dostatno za traženje ekstrema. Kada bi imali manje od 3 slike po oktavi onda uvjet *26-susjedstva* ne bi bio zadovoljen.



Slika 35 Ekstremi se detektiraju usporedbom slikovnog elementa X i njegovih 26 susjeda.

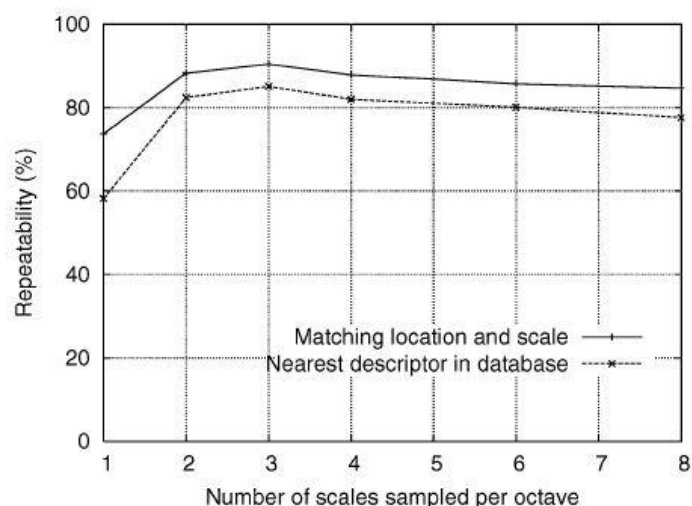
Iako na prvi pogled tako ne izgleda, trošak ovog koraka nije velik pošto se većina promatranih točaka eliminira vrlo brzo nakon samo par provjera.

5.1.4 ODREĐIVANJE PARAMETARA

5.1.4.1 BROJ SKALA PO OKTAVI

Lowe je koristeći 32 slike eksperimentalno utvrdio broj skala koji maksimizira stabilnost ekstrema. Slike su sadržavale realne scene različitog sadržaja uključujući rotacije, skaliranje, affine transformacije, promjene u kontrastu i osvjetljenju te dodavanje šuma.

Sljedeća slika pokazuje kako se ponovljivost odnosno stabilnost značajki mijenja ovisno o broju skala po oktavi.

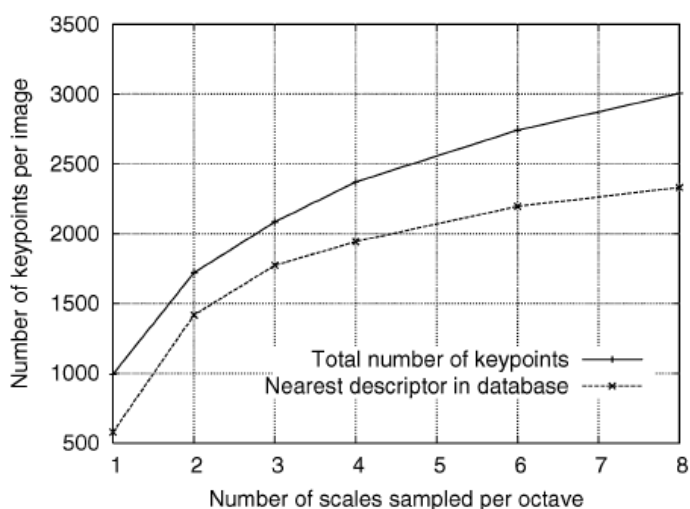


Slika 36 Utjecaj broja skala na ponovljivost značajke.

Gornja linija pokazuje postotak značajki koje su ponovno detektirane u transformiranoj slici, dok donja linija pokazuje postotak značajki za koje je pronađena korespondentna značajka u bazi podataka od 40 000 unosa koristeći algoritam najbližeg susjeda. Ovakvo svojstvo je itekako

poželjno u algoritmima prepoznavanja. Kako graf pokazuje najbolja ponovljivost se ostvaruje koristeći $s = 3$ skale po oktavi.

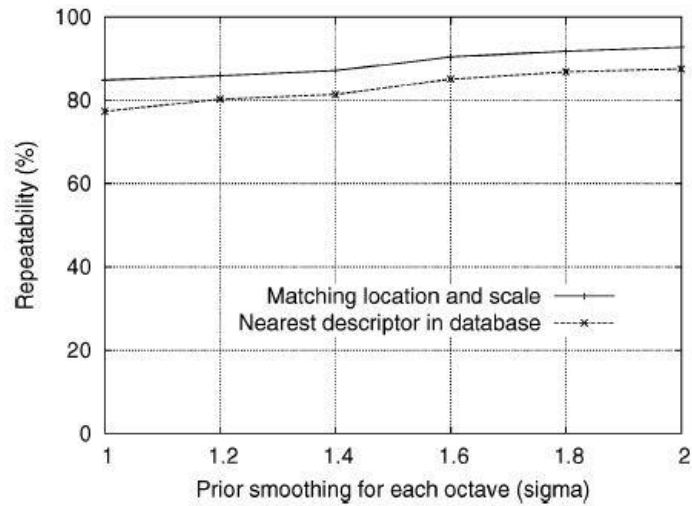
Kako sljedeća slika pokazuje, povećanje broja skala po oktavi utječe na povećanje broja značajki te broja korespondentnih značajki pronađenih u bazi. Iako se korištenjem više skala po oktavi detektira veći broj značajki, one su u prosjeku manje stabilne pa je prema tome i njihova ponovljivost niža. Na taj način se također i efikasnost algoritma dovodi u pitanje pa je prema tome najoptimalnije odabrati broj skala pri kojem se detektiraju najstabilnije značajke koje su dostatne za većinu primjena.



Slika 37 Broj značajki ovisno o broju skala po oktavi.

5.1.4.2 PRETHODNO ZAMUĆIVANJE ULAZNIH SLIKA

Lowe je također ustvrdio da se zamućivanjem ulaznih slika ponovljivost značajki bitno mijenja. Kao što sljedeća slika pokazuje odnos ponovljivosti i inicijalnog faktora zamućivanja σ je proporcionalan.



Slika 38 Ponovljivost raste proporcionalno sa zamućivanjem.

Međutim kako σ zapravo određuje veličinu jezgre Gaussovog filtra, zbog efikasnosti Lowe je odabrao $\sigma = 1.6$ koja daje ponovljivost dovoljno blisku optimalnoj.

5.1.4.3 PROCESIRANJE ULAZNE SLIKE

Ukoliko bi se ulazna slika zamutila, kao što je već rečeno, prije detekcije ekstrema, tada bi se visoke prostorne frekvencije izgubile. Prema tome, kako bi se one očuvale Lowe predlaže povećanje veličine **prve** ulazne slike (prva oktava piramide) za dva puta. Pretpostavka je da ulazna slika ima zamućenje od $\sigma = 0.5$ pa iz tog proizlazi da će nova, dva puta veća, ulazna slika imati zamućenje od $\sigma = 1.0$. Ovakvo procesiranje ulazne slike povećava broj stabilnih točaka za čak 4 puta, dok daljnja povećanja ne pokazuju neka veća poboljšanja.

5.2 LOKALIZACIJA ZNAČAJKI

Nakon detekcije ekstrema, sljedeći korak algoritma preračunava njihove cjelobrojne lokacije i skale u realne. Takav pristup pridonosi preciznijoj lokalizaciji i većoj stabilnosti značajki. Izračun se temelji na 3D kvadratnoj funkciji koja koristi Taylorov razvoj reda (na do kvadrat):

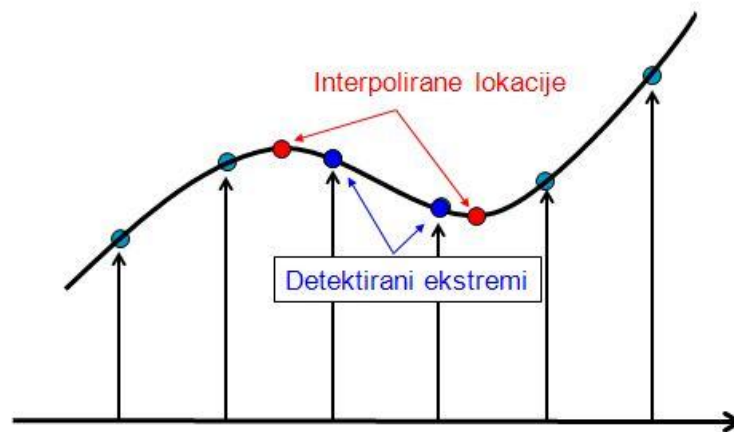
$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x} \quad (1.25)$$

Interpolirani parametri ekstrema dobivaju se derivacijom dane funkcije po parametrima ekstrema $\mathbf{x} = (x, y, \sigma)^T$ i izjednačavanjem s nulom:

$$\hat{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}} \quad (1.26)$$

gdje je $\frac{\partial^2 D^{-1}}{\partial x^2}$ Hessiana matrica koja se lako aproksimira koristeći diferencije a D prva derivacija oko točke ekstrema.

Ukoliko je vektor $\hat{x} = (\hat{x}, \hat{y}, \hat{\sigma})^T$ po bilo kojoj od svojih dimenzija po vrijednosti veći od **0.5**, tada to znači da se lokacija ekstrema nalazi bliže nekoj drugoj interpoliranoj točki. U tom slučaju, interpolacija se ponavlja ali oko nove točke: $\hat{x} = x + \hat{x}$. Ukoliko interpolacija konvergira nakon određenog broja iteracija, izračunati pomak \hat{x} se dodaje lokaciji i skali na kojoj je ekstrem originalno pronađen: $x = x + \hat{x}$.



Slika 39 Ilustracija problema.

5.2.1 ODBACIVANJE NESTABILNIH ZNAČAJKI

Kako bi se eliminirale nestabilne značajke SIFT koristi dvije metode.

5.2.1.1 ODBACIVANJE ZNAČAJKI NA TEMELJU NISKOGR KONTRASTA

Prethodno izračunat pomak \hat{x} iskorištava se za odbacivanje značajki s lošim kontrastom koje su vrlo osjetljive na šum. Kombinirajući prethodno dvije navedene formule dobivamo:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (1.27)$$

Značajke za koje vrijedi $D(\hat{x}) < 0.3$ su značajke s lošim kontrastom pa se prema tome one odbacuju.

5.2.1.2 ODBACIVANJE ZNAČAJKI NA TEMELJU SLABIH RUBOVA

Kako DoG pronalazi između ostalog i rubove koji su zapravo slabo definirani pa su prema tome vrlo osjetljivi na šum. Ukoliko se značajka detektira upravo na takvim mjestima tada je ona nestabilna i treba se eliminirati. Takvi loše definirani rubovi imaju veliku glavnu zakrivljenost

(engl. *principal curvature*) duž ruba a malu u ortogonalnom smjeru. Glavna zakrivljenost se može izračunati pomoću 2x2 Hessianove matrice, \mathbf{H} , u točki značajke:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (1.28)$$

Vlastite vrijednosti Hessianove matrice \mathbf{H} su proporcionalne glavnim zakrivljenostima parcijalnih derivacija u točki značajke. Međutim, kako je bitan samo njihov međusoban odnos, izračun vlastitih vrijednosti se može izbjeći:

Neka je α vlastita vrijednost najveće magnitude, a β manje magnitude.

Tada sumu vlastitih vrijednosti možemo izračunati preko traga matrice \mathbf{H} a umnožak preko determinante:

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta \quad (1.29)$$

$$Det(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (1.30)$$

Ukoliko determinanta, iako vrlo rijetko, ispadne negativna tada se značajka također odbacuje.

Sada neka je r odnos između tih vlastitih vrijednosti takav da vrijedi: $\alpha = r\beta$. Iz toga slijedi:

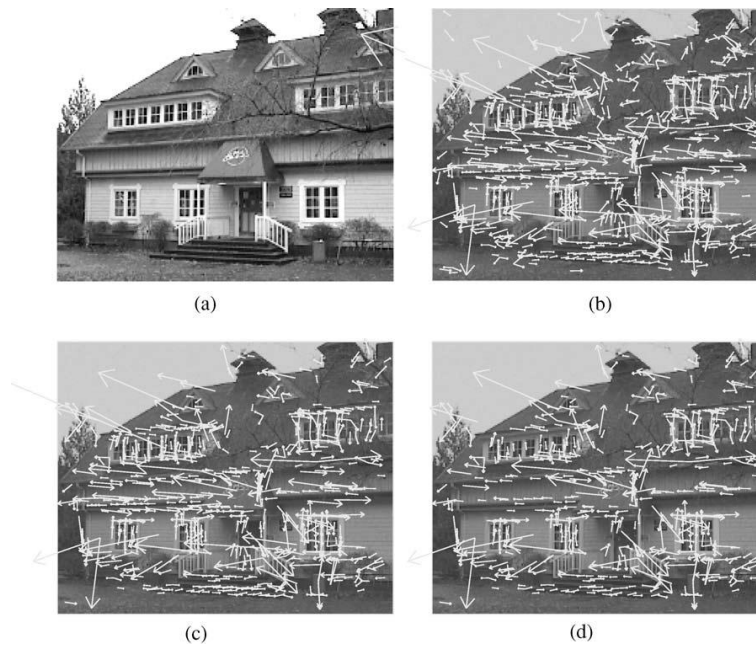
$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r} \quad (1.31)$$

Prema tome provjera da li je glavna zakrivljenost ispod nekog praga, r , vrši se jednostavno pomoću izraza:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r + 1)^2}{r} \quad (1.32)$$

Lowe u svom radu koristi prag $r = 10$.

Sljedeća slika prikazuje korake odbacivanja značajki.



Slika 40 **(a)** Originalna 233x189 slika. **(b)** Inicijalno je nakon DoG pronađeno ukupno 832 značajke. **(c)** Nakon eliminacije točaka s niskim kontrastom preostalo je 729 značajki. **(d)** Konačno je nakon eliminacije na temelju glavnih zakrivljenosti preostalo 536 značajki.

5.3 DODJELJIVANJE ORIJENTACIJE

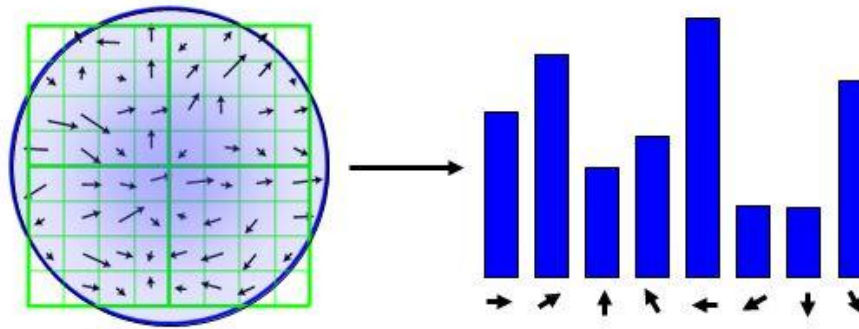
Nakon procesa lokalizacije i eliminacije SIFT značajkama pridjeljuje orijentaciju. Lowe je zaključio kako sljedeći pristup daje najstabilnije rezultate.

Skala značajke određuje koju zamućenu sliku, L , odabrati za daljnje izračune. Za svaki slikovni element odabrane slike $L(x, y)$ izračunaj magnitudu i orijentaciju gradijenta prema sljedećim formulama:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (1.33)$$

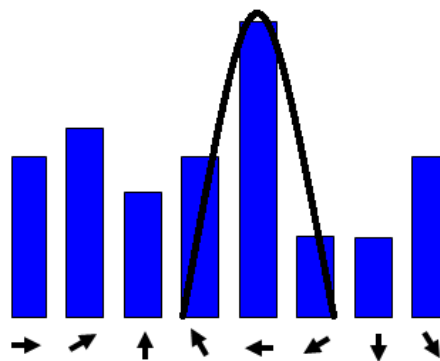
$$\theta(x, y) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (1.34)$$

Nakon toga se za svaku značajku oblikuje orijentacijski histogram pomoću izračunatih gradijenata. Histogram čini 36 blokova što bi značilo da svaki blok pokriva 10° . Svaki element odnosno izračunata orijentacija dodana u histogram se množi s njezinom izračunatom magnitudom te težinskim faktorom kojeg čini cirkularni Gaussov prozor širine σ koji je 1.5 puta veći nego skala značajke.



Slika 41 Ilustracija oblikovanja orijentacijskog histograma.

Maksimumi u orijentacijskom histogramu odgovaraju dominantnim smjerovima gradijenata oko značajke. Ukoliko u histogramu postoje vrhovi koji su najmanje 80% najvišeg vrha, tada se kreiraju nove značajke s istim lokacijama i skalama ali različitim orijentacijama. Otprilike je 15% takvih točaka ali eksperimentalno je utvrđeno da pridonose uspješnosti traženja korespondentnih točaka. Kao orijentacija značajke uzima se rezultat kvadratne interpolacije (engl. *parabola fitting*) tri susjedna bloka histograma od kojih je jedan onaj koji je odabran kao maksimum. Time se postiže finija raspodjela orijentacije pošto svaki blok histograma ima raspon od 10° .



Slika 42 Interpolacija oko maksimuma.

5.4 IZGRADNJA DESKRIPTORA

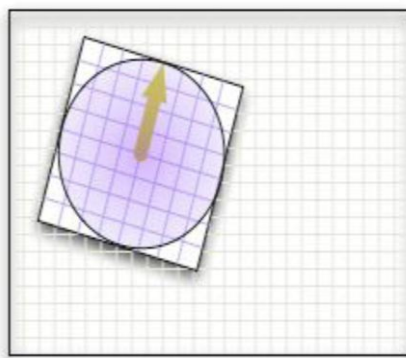
Do ovog posljednjeg koraka algoritma procesirane su sve značajke – pridjeljene su im lokacije, skale i orijentacije pomoću kojih je postignuta invarijantnost na skalu i rotaciju. U ovom koraku će se postići i invarijantnost na promjene u osvjetljenju te na promjene položaja kamere.

Jedan od pristupa bi mogao uključivati korištenje normalizirane korelacije slikovnih elemenata u regiji oko značajke. Međutim, ovakva jednostavna korelacija je vrlo osjetljiva na promjene koje uzrokuju pomake slikovnih elemenata kao što su affine transformacije ili promjene položaja

kamere. Ideja na kojoj Lowe zasniva svoje rješenje temelji se na radu Edelmana (Edelman, S. et al., 1997) čiji pristup uključuje područje biološkog vida odnosno simulaciju kompleksnih neurona iz primarne vizualne moždane opne (engl. *primary visual cortex*). Ti neuroni odgovaraju gradijentu točno određene orijentacije i prostorne frekvencije ali lokacije koja je podložna malim pomacima. Upravo ti mali pomaci su se pokazali kao puno bolje sredstvo za ostvarivanje invarijantnosti obzirom na promjene položaja kamere. Rezultati Edelmanovih testiranja su dosegla čak 94% ispravnih raspoznavanja u odnosu na pristup korelacije čiji je rezultat bio tek 35%.

5.4.1 REPREZENTACIJA DESKRIPTORA

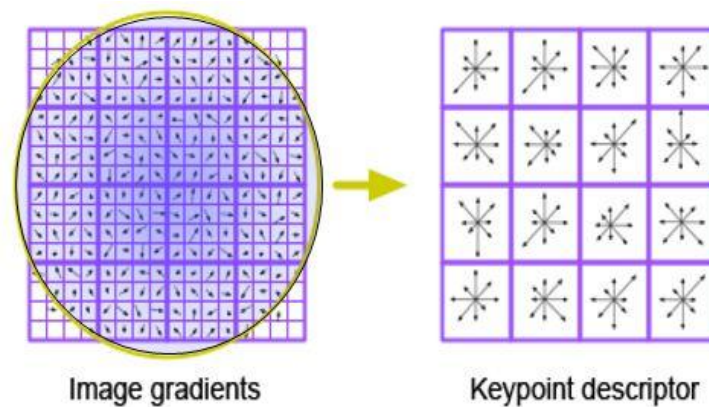
Proces izgradnje deskriptora započinje skupljanjem gradijenata oko svake značajke. Slika iz koje se odabiru gradijenti odgovara skali značajke koja se trenutno promatra. Kako bi se postigla invarijantnost na orijentaciju, koordinate deskriptora te orijentacije gradijenata su rotirane relativno u odnosu na orijentaciju značajke.



Slika 43 Rotacija u odnosu na orijentaciju značajke.

Na magnitude gradijenata primjenjuje se Gaussov prozor širine σ koja je jednaka jednoj polovici širine prozora deskriptora. Cilj je izbjeći iznenadne promjene u deskriptoru koje su uzrokovane manjim promjenama u poziciji prozora te dati manji značaj gradijentima koji su dalje od značajke pošto su oni najosjetljiviji na promjene koje uzrokuju pomake slikovnih elemenata (engl. *misregistration*).

Sljedeći korak u izgradnji deskriptora je izračun orijentacijskih histograma od 8 blokova u 4×4 regijama oko značajke. Na takav način se, po uzoru na rad Edelmana, postiže da pomak gradijenata do ukupno 4 pozicije još uvijek pridonosi istom histogramu.



Slika 44 Ilustracija izgradnje deskriptora.

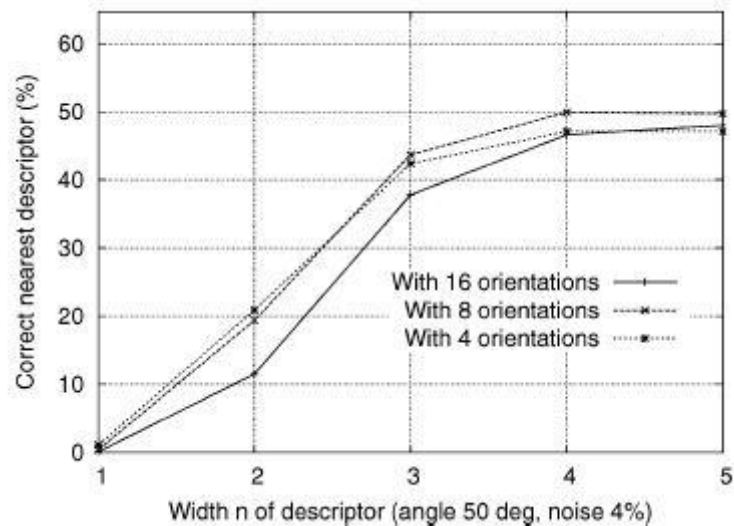
Kako bi se izbjegle nagle promjene deskriptora uzrokovane pomacima gradijenata, Lowe koristi trilinearnu interpolaciju kako bi svaki uzorak histograma rasporedio u odgovarajući blok. Drugim riječima, svaki uzorak histograma se množi s težinskim faktorom $1 - d$ za svaku dimenziju, gdje je d udaljenost uzorka od centralne vrijednosti bloka.

Konačno, deskriptor se formira koristeći izračunate orijentacijske histograme. Lowe u svom radu koristi polje od 4×4 histograma gdje svaki od njih ima po 8 blokova. Prema tome, vektor značajki je veličine $4 \times 4 \times 8 = 128$.

Kako bi se postigla i invarijantnost na promjene u osvjetljenju, vrši se normalizacija formiranog vektora. Primjerice linearne promjene kao što su promjene u kontrastu ili promjene u osvjetljenju biti će poništene normalizacijom. Međutim, nelinearne promjene kao što su saturacija ili nelinearne promjene u osvjetljenju pridonose velikim promjenama u magnitudama gradijenata. Da bi se smanjio utjecaj takvih potencijalnih promjena, postavlja se prag na velike magnitude i on iznosi 0.2. Nakon toga vektor se ponovno treba normalizirati.

5.4.2 ODABIR PARAMETARA

Dva su parametra koja određuju kompleksnost deskriptora: broj blokova po histogramu i broj orijentacijskih histograma. Sljedeća slika pokazuje eksperimentalne rezultate koji pokazuju kako se mijenja postotak ispravno pronađenih korespondentnih točaka s promjenom broja blokova po histogramu. Korištenim slikama dodano je 4% šuma te je položaj kamere postavljen pod nagibom od 50% u odnosu na ravninu kako bi se ispitala i invarijantnost na affine transformacije.

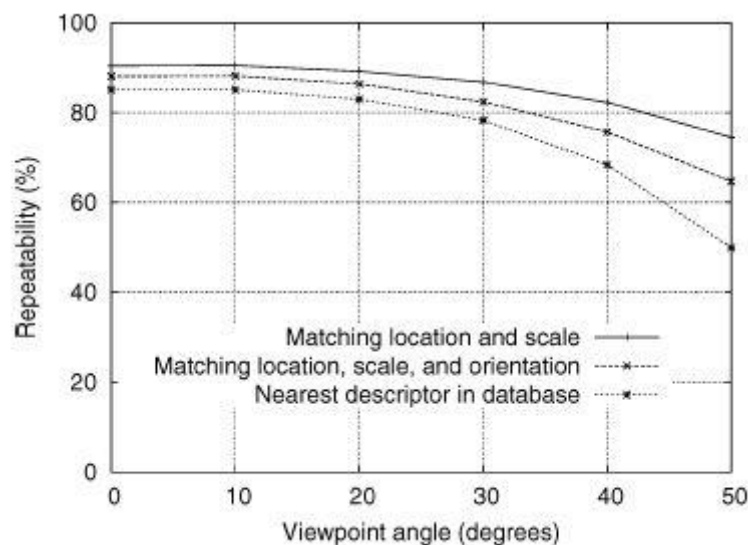


Slika 45 Postotak ispravno pronađenih korespondentnih točaka u bazi podataka od 40 000 značajki u ovisnosti o parametrima deskriptora.

Kao što graf pokazuje, rezultati pri širini deskriptora $n = 1$ i nisu zadovoljavajući ali napreduju rastom parametra n sve do $n = 4$. Daljnje povećanje širine deskriptora ne pridonosi većim napretkom pa se Lowe prema tome odlučio za $n = 4$ te 8 blokova po histogramu. Primjerice 16 blokova po histogramu daje nešto bolje rezultate ali i negativno utječe na vremensku efikasnost cjelokupnog algoritma.

5.4.3 AFINE TRANSFORMACIJE

Sljedeća slika pokazuje osjetljivost deskriptora na afine transformacije, tj. poglede na scene pod različitim nagibima.



Slika 46 Osjetljivost deskriptora na afine transformacije.

Kao što je i bilo za očekivati, stabilnost značajki opada povećanjem afinih distorzija, iako nikada ispod 50% za promjene do 50°. Za postizanje invarijantnosti na veće promjene, trebali bi se iskoristiti neki od spomenutih detektora koji su neosjetljivi na afine transformacije. Mikolajczyk je u svom radu (Mikolajczyk, K. et al., 2002) predložio afinu verziju Harrisovog detektora koja je za promjene od 50° davala čak lošije rezultate ali je za promjene od 70° zadržavala stabilnost značajki na 40%. Još jedan pristup bi bio koristiti standardni SIFT detektor i stvoriti bazu dodatnih značajki na slikama na kojima su primjenjene afine transformacije. Pritchard i Heidrich su u svom radu (Pritchard, D. et al., 2003) na ovaj način uspjeli poboljšati rezultate za čak 3 puta.

5.5 PRIMJENE

SIFT se zbog svoje robusnosti pronalazi u različitim primjenama računalnog vida. Neki od primjera su:

- Problem lokalizacije robota
 - o Robot pomoću stereo vizije (2 kamere) izgrađuje 3D model nepoznate okoline u kojoj se nalazi i tako omogućava proračun svoje trenutne pozicije.
- Stvaranje panoramskih snimaka
 - o SIFT značajke omogućuju automatizaciju stvaranja panoramskih snimaka.
- Proširena stvarnost (engl. *augmented reality*)
 - o Pojam se odnosi na računalno generiranje imaginarnih objekata unutar živih scena, bilo da su informativnog ili zabavnog sadržaja.
- Filmska industrija
 - o SIFT se koristi u tehnici koja u filmske scene ubacuje efekte računalne grafike.
- Interakcija čovjeka i računala
 - o Primjena u kontekstu prepoznavanja ljudskih akcija od strane računala.
- Medicina
 - o Primjena prilikom analize ljudskog mozga pomoću MRI slika.

Inače, Lowe se u svojim radovima detaljnije bavio primjenom prepoznavanja objekata (engl. *object recognition*).

5.5.1 PREPOZNAVANJE OBJEKATA (ENGL. *OBJECT RECOGNITION*)

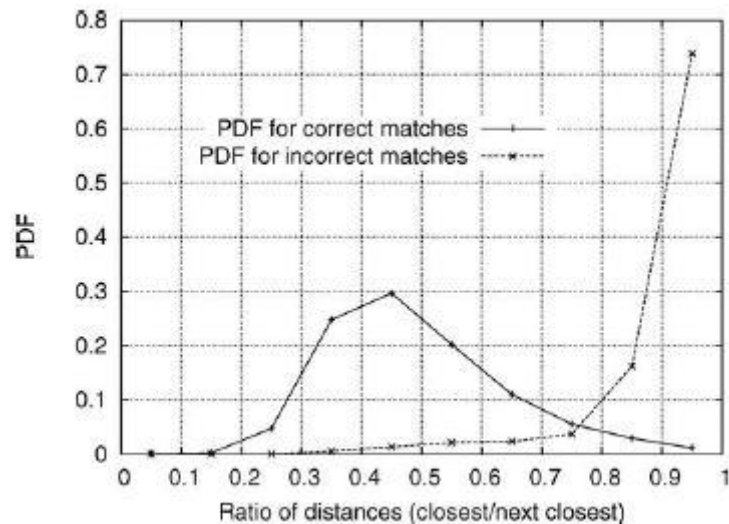
Proces prepoznavanja započinje ekstrakcijom značajki za koje se dalje pokušavaju naći odgovarajuće značajke iz baze podataka koje su dobivene pomoću testnih slika. Međutim, mnoge odgovarajuće značajke neće biti ispravno detektirane zbog određene količine šuma

prisutnog u slikama te nejednoznačnosti nekih značajki. Iz tog razloga se značajke ne promatraju individualno već u skupinama od najmanje 3 značajke koje odgovaraju nekom objektu na određenoj poziciji u slici. Takve skupine podliježu daljnjim provjerama koje onda daju konačan sud u procesu prepoznavanja. Cijeli proces detaljnije je objašnjen u sljedećim poglavljima.

5.5.1.1 NALAZENJE ODGOVARAJUĆIH ZNAČAJKI

Kako bi se u nekoj složenoj sceni prepoznao određeni objekt, očito je da bi se trebale usporediti značajke te scene sa značajkama traženog objekta. Prema tome, potrebno je imati bazu značajki unaprijed izračunatih na testnim slikama traženog objekta. Sama usporedba značajki vrši se pomoću algoritma najbližeg susjeda koji se temelji na Euklidskoj udaljenosti. Podudarnost značajki bi trebala biti upravo na mjestu gdje se nalazi objekt, međutim zbog prisutnosti šuma i sličnih smetnji za neke značajke neće biti pronađen niti jedan odgovarajući par u bazi. Da bi se riješio ovakav problem potrebno je razviti metodu koja će uspješno detektirati takve značajke te ih odbaciti iz daljnjeg proračuna. Primjerice, Lowe predlaže metodu koja pronalazi najbližu i drugu najbližu susjednu značajku. Ovakva kombinacija dvije najbliže udaljenosti daje dobre rezultate pošto će za ispravnu detekciju udaljenost u većini slučajeva biti manja nego što bi to bio slučaj kod neispravne detekcije. Primjerice, krivo detektirane značajke se mogu pojaviti zbog visoke dimenzionalnosti prostora značajki pa je samim time i veća vjerojatnost da će neka od udaljenosti biti dovoljno bliska za *ispravnu* detekciju.

Sljedeća slika pokazuje rezultate ovakve mjere na realnim slikama gdje je prikazan odnos najbližeg i drugog najbližeg susjeda i funkcije gustoće vjerojatnosti (*PDF*). Graf pokazuje da je funkcija gustoće vjerojatnosti za ispravno detektirane značajke više centrirana na mjestima gdje su omjeri među udaljenostima niži. Prema tome, Lowe odlučuje odbaciti značajke čiji su omjeri veći od 0.8 čime eliminira barem 90% lažno detektiranih značajki a samo oko 5% onih ispravnih.



Slika 47 Vjerojatnost da je pronađena odgovarajuća značajka ispravna može se ocijeniti promatrajući odnos udaljenosti najbliže i druge najbliže susjedne značajke. Isprekidana linija pokazuje neispravno detektirane značajke dok puna one ispravno detektirane.

5.5.1.2 EFIKASNO PRONALAZENJE NAJBLIŽEG SUSJEDA

Ne postoje poznati algoritmi koji u visoko dimenzionalnom prostoru mogu pronaći najbližeg susjeda na efikasniji način od tradicionalnog iscrpnog pretraživanja. Čak ni algoritam koji koristi *kd-stablo* ne daje znatnija ubrzanja već za više od 10 dimenzija. Zbog toga je Lowe predložio BBF algoritam (engl. *Best-Bin-First*) koji aproksimativno pronalazi najbližeg susjeda u s velikom vjerojatnošću.

BBF algoritam koristi prilagođeni način pretraživanja *kd-stabla* u kojem se red pretrage temelji na najbližoj udaljenosti u odnosu na onu referentnu. Ovakav prioritetni način zahtjeva implementaciju prioritetnog reda na hrpi (engl. *heap based priority queue*). Prednost u odnosu na tradicionalni način pretraživanja je u tome što se pretraga može prekinuti već nakon određenog broja koraka ispitivanja najbližih susjeda. Lowe u svojoj implementaciji pretražuje najbližih 200 značajki. U bazi podataka od 100 000 značajki, ovakva metoda donosi ubrzanje od čak 2 reda veličine a samo oko 5% gubitka ispravnih detekcija.

5.5.1.3 FORMIRANJE SKUPINA ZNAČAJKI POMOĆU HOUGHOVE TRANSFORMACIJE

Kako bi prepoznavanje objekta bilo pouzdano Lowe je zaključio da su već dovoljne samo 3 značajke koje upućuju na objekt. Iako su u prethodnim koracima zbog prisutnosti šuma odbačene većina lažno detektiranih parova značajki, još uvijek nisu procesirane značajke koje su pronađene na mjestima objekata. Mnoge poznate metode kao što su RANSAC (engl. *RANdom SAMple Consensus*) ili LMS (engl. *Least Median of Squares*) ne daju dobre rezultate kada se većina značajki treba odbaciti kao što je slučaj ovdje pa se prema tome Lowe odlučuje za

generaliziranu Houghovu transformaciju pomoću koje se formiraju skupine od najmanje 3 značajke u prostoru položaja (engl. *pose space*). Na ovaj način se vjerojatnost krive interpretacije *ispravnosti* značajki smanjuje.

5.5.1.4 ODREĐIVANJE AFINIH PARAMETARA

Nakon Houghove transformacije svaka grupa značajki je predmet geometrijske verifikacije koristeći metodu najmanjih kvadrata (engl. *least squares*). Na taj način se dolazi do najboljih afinih projekcijskih parametara u odnosu na testne slike.

Afina transformacija kod 3D rotacija ne-planarnih površina za razliku od rotacije planarnih površina ne daje dovoljno dobru aproksimaciju. Općenitije rješenje bi bila fundamentalna matrica, no ona zahtjeva najmanje 7 točaka za izračun za razliku od samo 3 točke kod afine transformacije. Prema tome Lowe se odlučuje ipak za korištenje afnog rješenja.

Ukoliko projekcija značajke leži unutar definiranog opsega pogreške te ukoliko je unutar grupe ostalo više od 3 značajke tada se ona zadržava. Postupak najmanjih kvadrata se iterativno ponavlja sve dok se ne provjere sve točke.

5.5.1.5 PRIMJERI

Lowe je u svom radu (Lowe, D.G. et al., 2004) dao par primjera prepoznavanja objekata vlastite implementacije pa prema tome slijedi par slika koje prikazuju rezultate procesa.

Na sljedećoj slici lijevo prikazani su objekti koji se trebaju prepoznati – žaba i vlakić. U sredini se nalazi vrlo složena scena koja sadrži te objekte ali **sakrivene** iza raznih drugih objekata. Na slici desno su prikazani rezultati prepoznavanja. Unutar paralelograma koji identificira objekt prikazane su samo značajke koje su iskorištene pri prepoznavanju gdje su značajke prikazane u obliku malih kvadrata. Kao što se vidi rezultati su odlični i upravo tu se vidi snaga razvijene metode koja zaista ima moć prepoznavanja čak i kad su objekti prikriveni.



Slika 48 Rezultat metode za prepoznavanje objekata koju je predložio Lowe.

Slijedi još jedan primjer prepoznavanja koji pokazuje moć ove metode. Kao što slika i prikazuje prepoznavaju su razni drveni uzorci te čak i grane.



Slika 49 Primjer prepoznavanja čak i ljudskom oku teško diferencirajućih uzoraka.

6 IMPLEMENTACIJA SIFT ALGORITMA

Ovo poglavlje ukratko će opisati implementaciju SIFT algoritma baziranu na podacima iz članka D.Lowea (Lowe, D.G. et al., 2004). Algoritam je implementiran u programskom jeziku C++ koristeći jednu od danas najpopularnijih biblioteka računalnog vida – *OpenCV*. Razvojno okruženje je *Codegear C++ Builder 2009*.

Razvijene su dvije klase od kojih je jedna pomoćna dok je druga sučelje prema krajnjem korisniku. Tijekom razvoja, kao dodatna pomoć pri razumijevanju algoritma, korištene su SIFT implementacije otvorenog tipa A.Vedaldija (Vedaldi, A. et al., 2005) i R.Hessa (Hess, R. et al., 2010).

Ime klase	Funkcionalnost
<i>SIFT</i>	Implementacija SIFT algoritma koja korisniku nudi sučelje za korištenje.
<i>Feature</i>	Pomoćna klasa koja oponaša lokalnu značajku.

6.1 FEATURE KLASA

Ova klasa zapravo služi kao struktura podataka koja reprezentira lokalnu značajku. Za svaku lokalnu značajku spremaju se dvije vrste informacija:

- lokacija, oktava i skala na kojima je značajka zaista pronađena
- interpolirana lokacija i skala te orijentacija i deskriptor značajke

Klasa pored dva konstruktora ne sadrži nikakve dodatne metode. Prvi konstruktor rezervira potrebnu memoriju i vrši inicijalizaciju članskih varijabli, dok drugi stvara novu značajku na temelju postojeće. Takav scenarij je recimo pogodan u fazi algoritma gdje se kreiraju nove značajke istih lokacija, oktava i skala ali različitih orijentacija.

```
struct DetectionData
{
    int    row;        // feature detected row
    int    col;        // feature detected column
    int    octave;    // feature detected octave
    int    scale;     // feature detected scale
};

class Feature
{
    friend class SIFT;

private:
```

```

double x;          // interpolated feature row
double y;          // interpolated feature column
double scale;     // interpolated feature scale
double orientation; // feature orientation
vector<double> descriptor; // feature descriptor

// original data where feature was detected
DetectionData detectionData;

public:
    Feature();
    Feature(const Feature &f);
};

```

6.2 SIFT KLASA

Ovo je klasa koja sadrži implementaciju SIFT detektora značajki. Klasa je strukturirana slično po koracima algoritma pa će prema tome tim redoslijedom biti i opisana. Za svaki će se korak navesti odgovarajuće metode te parametri i njihove inicijalne vrijednosti kako je Lowe predložio u svom radu. Također će, ukoliko je to potrebno, biti komentirani pojedini parametri čija se vrijednost iz rada ne može olako iščitati.

6.2.1 PREPROCESIRANJE ULAZNE SLIKE

Algoritam zahtjeva par koraka koji bi se zbog boljih rezultata trebali izvršiti nad ulaznim slikama:

- zamučivanje Gausovim filtrom širine $\sigma = 0.5$
- povećanje veličine slike za dva puta
- prebacivanje skale slike iz standardne $[0, 255]$ u $[0, 1]$ skalu
- prebacivanje slike u sliku sivih tonova (engl. *grayscale*)

Dvije su metode koje obavljaju tražene korake:

```

IplImage*
preprocess(IplImage* img);

IplImage*
convertToGray32(IplImage* img);

```


Ime parametra	Opis	Vrijednost
<code>initSigma</code>	Širina prozora Gaussovog filtra koja se koristi pri preprocesiranju ulazne slike.	0.5
<code>resampleFactor</code>	Faktor povećanja ulazne slike.	2

6.2.2 DETEKCIJA EKSTREMA KROZ NIZ SKALA

Ovaj korak algoritma pronalazi ekstreme u slikama na različitim skalama kroz određen broj oktava. Prvo se stvaraju slike zamučene Gausovim filtrom (`BuildGaussianScaleSpace()`) a onda iz njih slike rubova pomoću *DoG* aproksimacije (`BuildDoGScaleSpace()`). Na slikama rubova se traže ekstremi (`isExtrema()`).

Pripadne metode su:

```
void
  BuildGaussianScaleSpace();

void
  BuildDoGScaleSpace();

bool
  isExtrema(int octave, int level, int row, int col);
```

Odgovarajući parametri su:

Ime parametra	Opis	Vrijednost
baseSigma	Osnovna širina prozora σ Gaussovog filtra koja se koristi pri zamućivanju.	1.6
kSigma	Frakcijski dio širine prozora Gaussovog filtra koji se koristi pri zamućivanju.	$\frac{1}{2^{nSubLevels}}$
gaussImgsPerOctave	Broj zamućenih slika po jednoj oktavi.	$nSubLevels + 3$
diffGaussImgsPerOctave	Broj slika po oktavi dobivenih razlikom susjednih zamućenih slika (engl. <i>DoG space</i>).	$gaussImgsPerOctave - 1$
nSubLevels	Broj intervala po jednoj oktavi.	3
nOctaves	Ukupan broj oktava.	$\log_2(MIN(image\ width, image\ height)) - 2$

Pošto Lowe u svom radu ne navodi na koji način se izračunava ukupan broj oktava, formula za izračun je preuzeta iz Vedaldijeve i Hessove implementacije. Lowe jedino u svom patentu (Lowe, D.G. et al., 2004), koji je stariji od posljednje verzije članka, navodi podatak kako slike iz posljednje oktave ne bi trebale biti manje od $30px$.

Također je i širina Gaussovog filtra pomoću kojeg se zamućuju slike u radu pomalo nejasan pa prema tome cjelovita formula glasi:

$$\sigma = baseSigma * 2^{octave + \frac{kSigma}{subLevel}} \quad (1.35)$$

gdje je *octave* oktava a *subLevel* razina na kojima se nalazi slika koja se trenutno procesira.

6.2.3 LOKALIZACIJA ZNAČAJKI

Nakon što su ekstremi detektirani potrebno je interpolacijom finije lokalizirati značajke te odbaciti one koje ne zadovoljavaju kriterije stabilnosti – značajke niskog kontrasta te slabih rubova.

```

Feature*
  keypointLocalization (int octave, int level,
                       int row, int col);

void
  keypointInterpolation(int octave, int level,
                       int row, int col,
                       double *r, double *c, double *l);

bool
  isLowContrastKeypoint(int octave, int level,
                       int row, int col,
                       double r, double c, double s);

bool
  isPoorEdgeKeypoint(int octave, int level,
                    int row, int col);

```

Metode `keypointLocalization()` i `keypointInterpolation()` zadužene su za interpolaciju parametara potencijalnih značajki kako bi se dobili precizniji rezultati, dok su metode `isLowContrastKeypoint()` i `isPoorEdgeKeypoint()` odgovorne za odbacivanje nestabilnih značajki.

Parametri koji se koriste u ovoj fazi algoritma su:

Ime parametra	Opis	Vrijednost
<code>lowContrastThreshold</code>	Prag koji određuje da li značajka ima nizak kontrast.	0.04
<code>curvatureThreshold</code>	Prag koji određuje da li se značajka nalazi na slabom rubu.	10
<code>maxInterpSteps</code>	Maksimalni broj koraka interpolacije.	5

6.2.4 DODJELJIVANJE ORIJENTACIJE

Nakon lokalizacije značajki te odbacivanja onih nestabilnih potrebno je značajkama pridjeliti orijentacije.

```

double*
  computeOrientationHist (Feature &feature);

void
  smoothHistogram (double *histogram);

```

```

int
  histogramPeak(double *histogram, double *max);

void
  addMoreKeypoints(Feature &feature,
                   vector<Feature> &newFeatures,
                   double *histogram,
                   double peak, int peakBin);

```

Metodom `computeOrientationHist()` izračunava se orijentacijski histogram u okolini značajke na temelju kojeg se dodjeljuje određena orijentacija. Nakon toga se histogram eventualno može izgladiti pomoću metode `smoothHistogram()`. Zaglađivanje histograma Lowe u svom radu ne spominje ali njegovim korištenjem u prosjeku se dobivaju stabilniji rezultati.

Na temelju izračunatog orijentacijskog histograma značajke provjerava se da li je moguće generirati iste značajke ali s novim orijentacijama. Za taj korak odgovorne su metode `histogramPeak()` i `addMoreKeypoints()`.

Parametri specifični za ovaj korak algoritma su:

Ime parametra	Opis	Vrijednost
<code>oriHistRegion</code>	Širina regije unutar koje se proračunava orijentacijski histogram značajke.	4
<code>oriHistBins</code>	Broj blokova (engl. <i>bin</i>) koji sačinjavaju orijentacijski histogram.	36
<code>peakRatio</code>	Prag koji određuje da li će se iz promatrane značajke generirati nove značajke.	0.8
<code>histSmoothSteps</code>	Broj koraka izgladivanja orijentacijskog histograma.	2

6.2.5 KREIRANJE DESKRIPTORA

Posljednji korak SIFT algoritma na temelju pronađenih značajki kreira odgovarajuće deskriptore te ih normalizira. Pri izračunu se koriste metode `computeDescriptor()` i `interpolateHistEntry()` dok se za normalizaciju koristi metoda `descriptorNormalization()`.

```

void
  computeDescriptor(Feature &feature);

```

```

void
  interpolateHistEntry(double ***hist,
                      double rowBin, double colBin,
                      double oriBin,
                      double magnitude);

void
  descriptorNormalization(vector<double> &descriptor);

```

Parametri su sljedeći:

Ime parametra	Opis	Vrijednost
magnification	Povećanje prozora deskriptora.	3
descriptorWidth	Širina regije u kojoj se izračunava deskriptor.	4
descriptorOriHistBins	Broj blokova orijentacijskog histograma u deskriptoru.	8
featureVectorSize	Veličina deskriptora.	128
descrMagnitudeThreshold	Prag koji određuje koje će se magnitude deskriptora odbaciti.	0.2

6.2.6 SUČELJE PREMA KORISNIKU

Korisniku je pruženo vrlo jednostavno sučelje za rad s klasom. Ukoliko želi koristiti inicijalne vrijednosti parametara algoritma, kao što su i predložene u članku, tada će sav posao za njega odraditi konstruktor `SIFT()`. Međutim podešavanje parametara na željene vrijednosti moguće je kroz `setXX()` metode, gdje je `XX` ime parametra. Slično, vrijednosti parametara moguće je dohvatiti kroz `getXX()` metode. Odabir slike nad kojom se žele izračunati SIFT značajke vrši se pomoću metode `loadInputImage()`. Nakon uspješno učitane slike, detekcija značajki započinje pozivom `run()` metode. Broj detektiranih značajki dohvaća se metodom `getNumberOfFeatures()`. Nakon detekcije, značajke je moguće pozivom metode `drawFeatures()` nacrtati na ulaznu sliku te prikazati pozivom metode `showFeatures()`. Rezultat metode `drawFeatures()` može se spremiti pozivom metode `saveFeatureImage()`. Također je moguće detektirane značajke spremiti na disk u tekstualnu datoteku pozivom `exportFeatures()`.

```

void

```

```

    loadImage(char* filename);

void
    run();

int
    getNumberOfFeatures();

void
    drawFeatures();

void
    showFeatures();

int
    exportFeatures(char* filename);

void
    saveFeatureImage(char* filename);

void
    setXX();

typeXX
    getXX();

```

6.3 DEMO APLIKACIJA

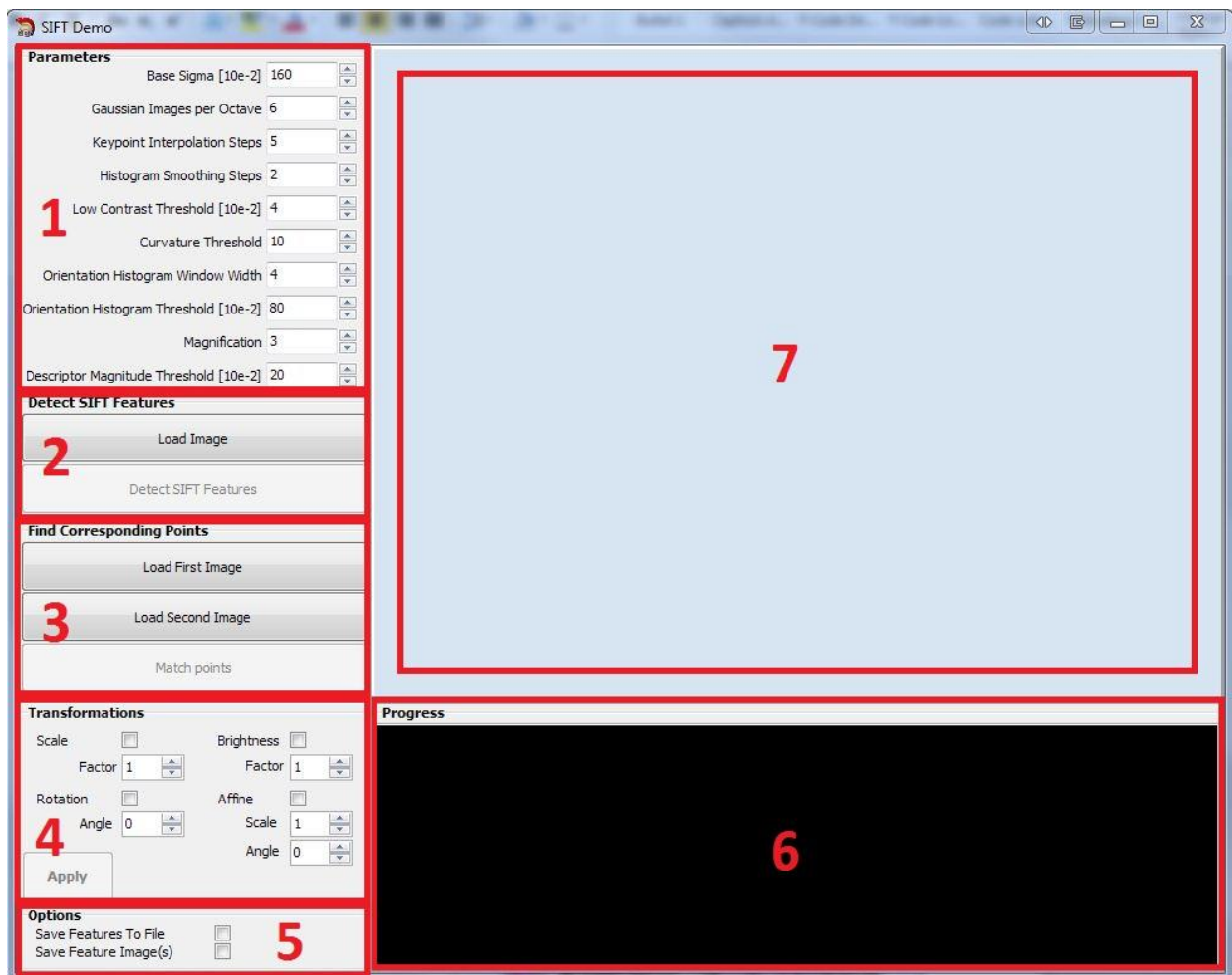
Kako bi se pokazala funkcionalnost implementiranog SIFT algoritma, u sklopu rada razvijena je demo aplikacija. Aplikacija omogućuje sljedeće korake:

1. Transformacije nad odabranom slikom (skaliranje, rotacija, afina transformacija, podešavanje svjetline).
2. Variranje parametara SIFT algoritma.
3. Detekcija SIFT značajki nad odabranom slikom.
4. Detekcija korespondentnih točaka na dvama odabranim slikama.
5. Spremanje rezultata u tekstualnom obliku (prema formatu kojeg je predložio Lowe) ili u obliku *.png* slike.

Početni prozor aplikacije prikazan je na sljedećoj slici. Dijelovi prozora su sljedeći:

1. Promjenjivi parametri algoritma opisani u poglavlju 6.2.
2. Detekcija SIFT značajki.
3. Detekcija korespondentnih točaka.

4. Transformacije nad slikom.
5. Opcionalno spremanje rezultata.
6. Tijek izvršavanja.
7. Prikaz rezultata.



Slika 50 Demo aplikacija.

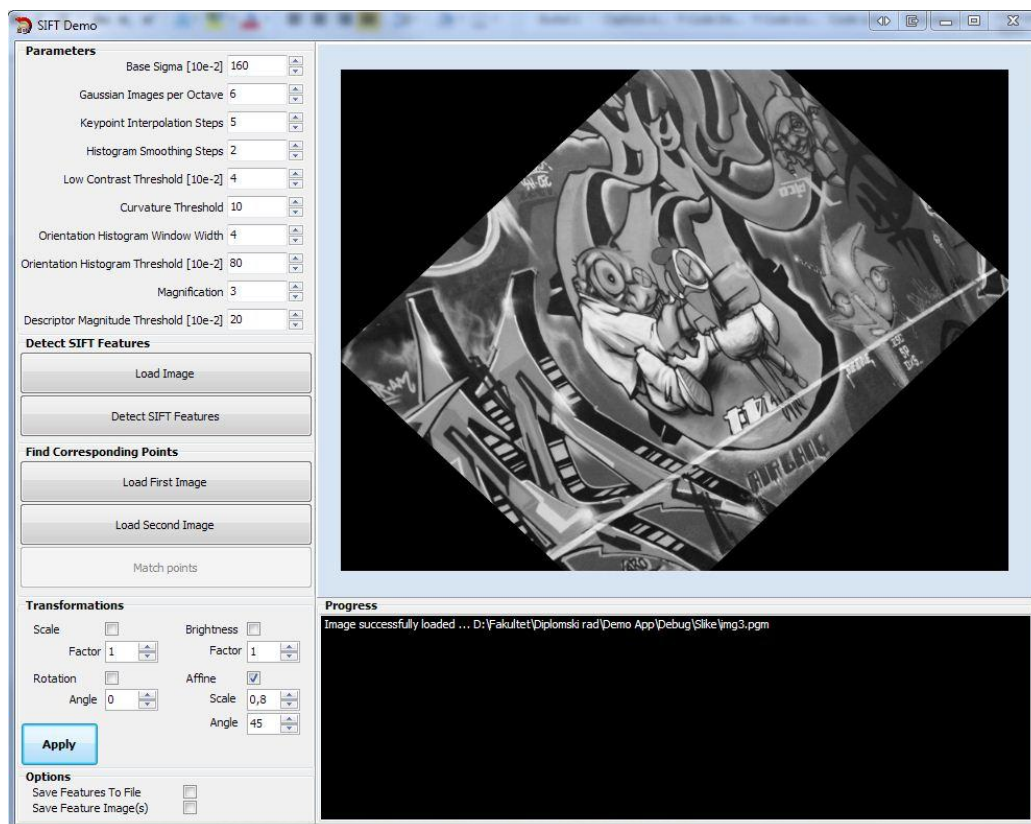
Slijedi kratki opis korištenja aplikacije.

6.3.1 TRANSFORMACIJE NAD SLIKAMA

Aplikacija omogućuje korisniku primjenjivanje raznih transformacija nad odabranom slikom. Na taj način moguće je generirati set sličnih slika i tako pratiti utjecaj transformacija na rezultate SIFT algoritma. Korisniku su ponuđene transformacije na koje je SIFT algoritam invarijantan:

1. Skaliranje
2. Rotacija
3. Afina transformacija
4. Svjetlina

Svaka od transformacija naravno ima i varijabilne parametre koje korisnik po želji može mijenjati. Nakon učitane slike pomoću gumba *Load Image*, određena transformacija (ili niz transformacija) se primjenjuje na sliku pritiskom na gumb *Apply*.

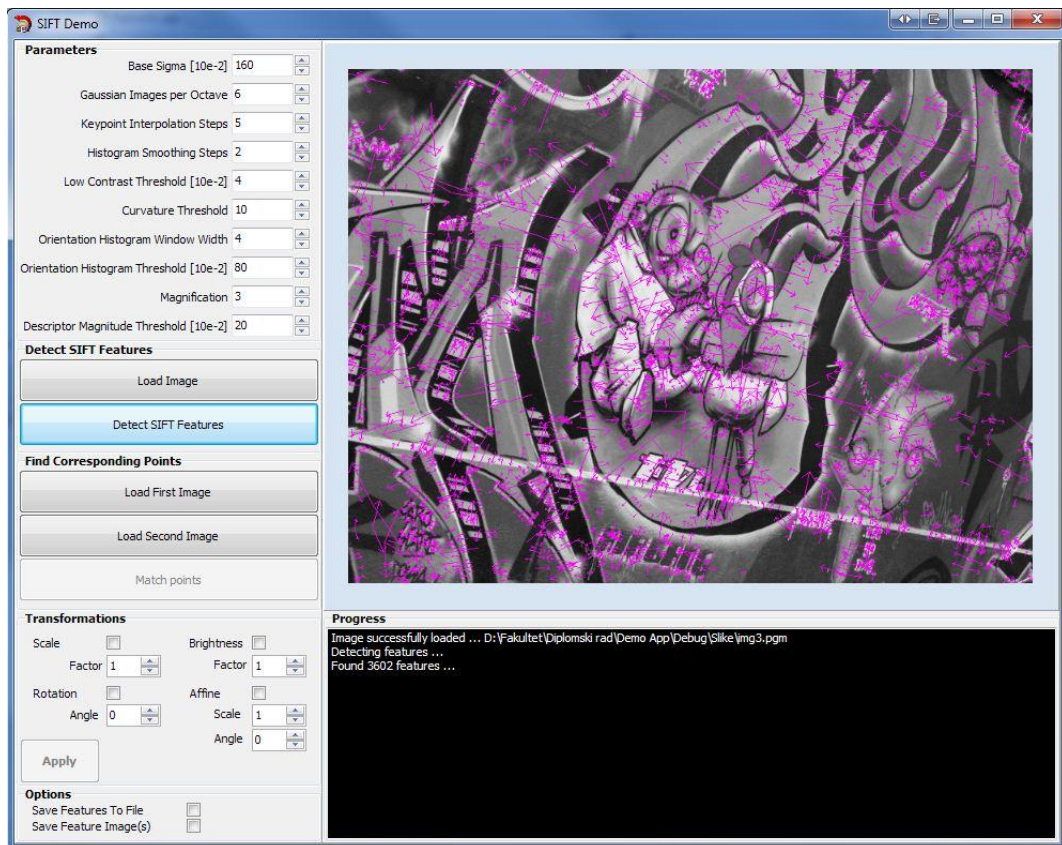


Slika 51 Afina transformacija nad slikom s parametrima: skala=0.8; kut=45°

6.3.2 DETEKCIJA SIFT ZNAČAJKI

Ovaj način rada prezentira i prikazuje rezultate detekcije SIFT značajki nad odabranom slikom. Željena slika se odabire pritiskom na gumb *Load Image* dok se algoritam pokreće na pritisak gumba *Detect SIFT Features*. Korisniku je također omogućeno mijenjanje parametara algoritma te spremanje rezultata na disk računala.

Primjer izvršavanja je dan na sljedećoj slici. Značajke su nacrtane u obliku strelica određene duljine.

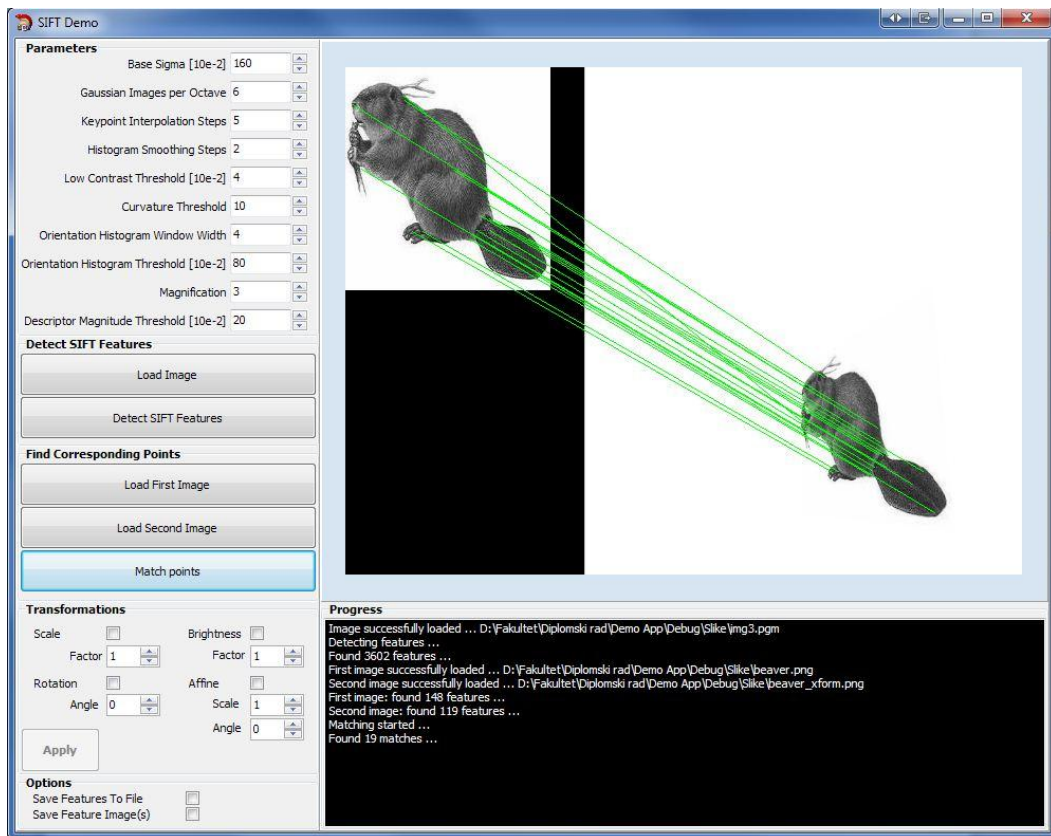


Slika 52 Detektirane SIFT značajke.

6.3.3 TRAŽENJE KORESPONENTNIH TOČAKA

U ovom načinu rada prezentira se rješenje problema kojim se bavio ovaj rad – traženje korespondentnih točaka. Korisnik prvo pritiskom na gumb *Load First Image* te *Load Second Image* treba odabrati dvije slike nad kojim želi naći korespondentne točke. Nakon što su slike uspješno učitane, korespondentne točke se pronalaze pritiskom na gumb *Match Points*. Korisniku je također omogućeno mijenjanje parametara algoritma te spremanje rezultata na disk računala.

Primjer izvršavanja je dan na sljedećoj slici. Korespondentne točke su povezane linijama.



Slika 53 Pronađene korespondentne točke.

7 ZAKLJUČAK

Rješenje problema pronalaženja korespondentnih točaka između dvije (ili više) slike zasniva se na uparivanju karakterističnih točaka tih dviju slika. Problemom detekcije karakterističnih točaka bavi se grana računalnog vida koja proučava lokalne značajke. Lokalne značajke standardni su alat koji se koristi u prepoznavanju i traženju objekata. Ovaj rad dao je pregled danas najpopularnijih detektora lokalnih značajki. Detektore razlikujemo po razini invarijantnosti koju podržavaju. Četiri su tipa invarijantnosti proučena u ovom radu: translacija, rotacija, skaliranje i afinost. Ugrubo možemo podijeliti detektore na tri razine invarijantnosti – one koji su invarijantni samo na translaciju i rotaciju, one koji su dodatno invarijantni i na skaliranje te na one koji su invarijantni na sva četiri tipa invarijantnosti. Koji detektor odabrati, direktno ovisi o primjeni odnosno potrebnoj razini invarijantnosti. U radu su predstavljeni mehanizmi rada svakog od detektora te njihove popratne karakteristike. Detaljnije od ostalih, opisan je SIFT algoritam koji se prema mnogim ispitivanjima pokazao kao najstabilniji. Algoritam je u detalje opisan korak po korak te naposljetku i implementiran. Prema tome, problem korespondencije riješen je upravo SIFT algoritmom.

8 LITERATURA

- Bay, H., Tuytelaars, T. i Van Gool, L. SURF: Speeded up robust features. Proceedings of the European Conference on Computer Vision. 2006, 404–417
- Bay, H., Ess, A., Tuytelaars, T. i Van Gool, L. Speeded-up robust features (SURF). International Journal on Computer Vision and Image Understanding. vol. 110, 3 (2008), 346–359
- Bradski G. i Kaehler A. Learning OpenCV, prvo izdanje, Sebastopol US, O'Reilly Media, 2008
- Edelman, S., Intrator, N., i Poggio, T. 1997. Complex cells and object recognition. <http://kybele.psych.cornell.edu/~edelman/archive.html>
- Harris, C. i Stephens, M. A Combined corner and edge detector. Alvey Vision Conference, 1988, 311-318
- Hartley R. i Zisserman A. Multiple View Geometry in Computer Vision, drugo izdanje, Cambridge UK, Cambridge University Press, 2003
- Hess R. SIFT Feature Detector, 2010., *A C implementation of a SIFT image feature detector*, <http://web.engr.oregonstate.edu/~hess/>, 20.4.2010.
- Kadir, T. i Brady, M. Scale, saliency and image description. International Journal of Computer Vision. vol. 45, 2 (2001), 83–105
- Kadir, T. i Brady, M. i Zisserman, A. An affine invariant method for selecting salient regions in images. Proceedings of the European Conference on Computer Vision. 2004, 345–457
- Lepetit, V. i Fua, P. Keypoint recognition using randomized trees. IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 28, 9 (2006), 1465–1479
- Lindeberg, T. Scale-space theory: A basic tool for analysing structures at different scales. Journal of Applied Statistics. 2 (1994), 225-270
- Lindeberg, T. Detecting salient blob-like image structures and their scales with a scale-space primal sketch – a method for focus-of-attention. International Journal of Computer Vision. vol 11, 3 (1993), 283-318
- Lindeberg, T. Feature detection with automatic scale selection. International Journal of Computer Vision. vol 30, 2 (1998), 79-116

Lindeberg, T. Direct estimation of affine image deformation using visual front-end operations with automatic scale selection. Proceedings of the International Conference on Computer Vision. 1995, 134-141

Lindeberg, T. i Garding, J. Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. Image and Vision Computing. vol. 15, 6 (1997), 415-434

Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision. 60 (2004), 91–110

Lowe, D.G. Method and Apparatus for Identifying Scale Invariant Features in an Image and use of same for locating an Object in an Image. United States Patent, US 6,711,293 B1, 2004

Matas, J., Chum, O., Urban, M. i Pajdla, T. Robust wide-baseline stereo from maximally stable extremal regions. Proceedings of the British Machine Vision Conference. 2002, 384–393

Mikolajczyk, K. i Schmid, C. Scale and affine invariant interest point detectors. International Journal of Computer Vision. vol 1, 60 (2004), 63-86

Mikolajczyk, K., 2002, Detection of local features invariant to affine transformations, Ph.D. thesis, Institut National Polytechnique de Grenoble, France

Mikolajczyk, K. i Schmid, C. An affine invariant interest point detector. European Conference on Computer Vision, 2002, 128–142

Mori, G., Ren, X., Efros, A. i Malik, J. Recovering human body configurations: Combining segmentation and recognition. Proceedings of the Conference on Computer Vision and Pattern Recognition. vol. 2, 2004, 326–333

Pritchard, D. i Heidrich, W. Cloth motion capture. Computer Graphics Forum (Eurographics 2003), vol 3, 22 (2003), 263–271

Ren, X., Fowlkes, C. i Malik, J. Scale-invariant contour completion using conditional random fields. Proceedings of the International Conference on Computer Vision. vol. 2, 2005, 1214–1221

Rosten, E. i Drummond, T. Fusing points and lines for high performance tracking. Proceedings of the International Conference on Computer Vision. 2005, 1508–1511

Rosten, E. i Drummond, T. Machine learning for high-speed corner detection. Proceedings of the European Conference on Computer Vision. 2006, 430–443

Quinlan, J. R. Induction of decision trees. Machine Learning. vol. 1, 1986, 81–106

- Schmid, C., Mohr, R. i Bauckhage, C. Comparing and evaluating interest points. Proceedings of the International Conference on Computer Vision. 1998, 230-235
- Schmid, C., Mohr, R. i Bauckhage, C. Comparing and evaluating interest point detectors. International Journal of Computer Vision. vol. 37, 2 (2000), 151-172
- Shi, J. i Malik, J. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 22, 8 (2000), 888–905
- Smith, S.M. i Brady, J.M. A new approach to low level image processing. International Journal of Computer Vision. vol 23, 34 (1997), 45-78
- Tuytelaars, T. i Van Gool, L. Content-based image retrieval based on local affinity invariant regions. International Conference on Visual Information Systems. 1999, 493-500
- Tuytelaars, T. i Mikolajczyk, K. Local Invariant Feature Detectors: A Survey. Foundations and Trends in Computer Graphics and Vision, 3 (2007), 177-280
- Tuytelaars, T. i Van Gool, L. Matching widely separated views based on affine invariant regions. International Journal of Computer Vision, vol. 1, 59 (2004), 61-85
- Tuytelaars, T. i Van Gool, L. Wide baseline stereo matching based on local, affinity invariant regions. Proceedings of the British Machine Vision Conference. 2000, 412–425
- Vedaldi A. SIFT Tutorial, *VLFeat: An open source computer vision in C*, 2005, <http://www.vlfeat.org/overview/sift.html>, 25.04.2010.

9 NASLOV, SAŽETAK I KLJUČNE RIJEČI

Naslov rada: „*Određivanje korespodentnih točaka na slikama dobivenim stereo kamerom*“

Ključne riječi: problem korespodencije, lokalne značajke, detektori značajki, SIFT

Ovaj rad iznosi problem traženja korespodentnih točaka na slikama dobivenim stereo kamerom. Kao uvod u problematiku, opisuje se epipolarna geometrija koja predstavlja matematički model problema korespodencije. Pristup rješavanju problema je u pronalaženju karakterističnih odnosno zajedničkih točaka u slikama. Karakteristične točke u kontekstu računalnog vida se nazivaju lokalne značajke. Rad u nastavku opisuje čemu služe lokalne značajke, koje sve vrste postoje te razne metode koje ih detektiraju. Metode, tj. detektori značajki međusobno su uspoređeni i dane su njihove glavne karakteristike. Kao jedna od najstabilnijih metoda danas znanih ali rješenje problema korespodencije, detaljno je opisan i implementiran SIFT algoritam.

10 TITLE, ABSTRACT AND KEYWORDS

Title: “*Finding point correspondences using stereo camera images*”

Keywords: correspondence problem, local features, feature detectors, SIFT

This thesis addresses a correspondence problem. As an intro, epipolar geometry is described as it presents a mathematical model of correspondence problem. Finding characteristic points in images is a way to solve the problem. These points in computer vision context are called local features. So furthermore, it is described what local features are, and also some of the popular methods for detecting them. Those methods or feature detectors are individually described and there was also given a table comparing all important characteristics for all of them. As one of the most stable methods nowadays and a solution to correspondence problem, SIFT algorithm is described in detail and given an implementation as well.