Računalni vid
Ključne točke u prostoru mjerila

Josip Šarić

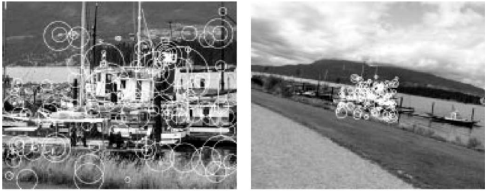
Fakultet elektrotehnike i računarstva
Sveučilište u Zagrebu

03. studenog 2023.

Pregled

Invarijantnost na mjerilo

- Cilj: Odrediti korespondenciju međusobno razmaknutih pogleda
- različiti smjerovi gledanja (oblik!)
- različita udaljenost kamera od scene (mjerilo!)



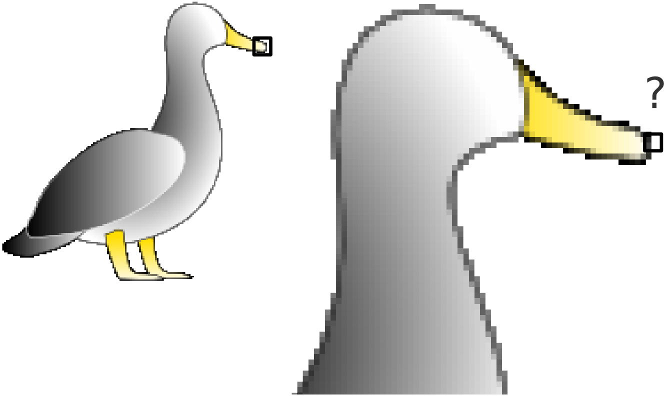
(a) Scale change of 3.9 and rotation of 17°.

(b) Scale change of 1.8 and viewpoint change of 30°

[Mikolajczyk04]

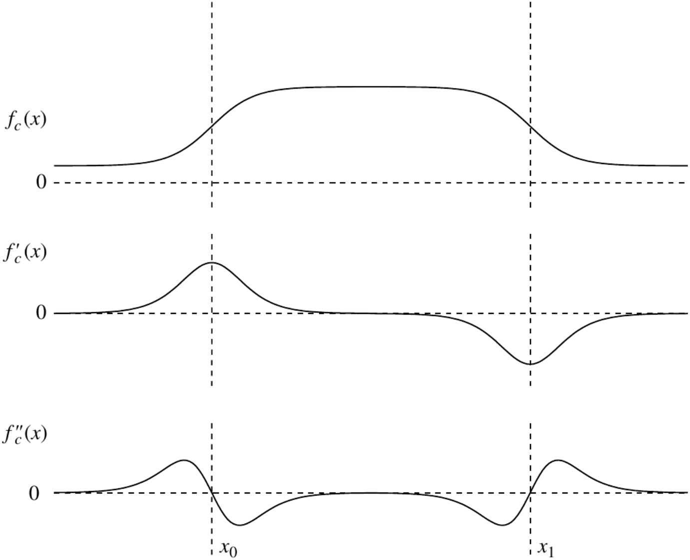
Invarijantnost na mjerilo

- Što fali Harrisu?
- rotacijski invarijantan, ali ne i invarijantan na mjerilo
- nema deskriptor značajki



Pregled

Druga derivacija u rubovima



[Bovik09]

Laplaceov operator

- Definicija: $\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$
- diskretizacija uz pomoć metode konačnih razlika:

$$\frac{\partial^2 I}{\partial x^2} \approx \frac{I(x+h,y) - 2I(x,y) + I(x-h,y)}{h^2} \tag{1}$$

$$\frac{\partial^2 I}{\partial y^2} \approx \frac{I(x,y+h) - 2I(x,y) + I(x,y-h)}{h^2} \tag{2}$$

- konvolucijska jezgra:

$$\begin{array}{ccc} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{array} \tag{3}$$

Laplaceov operator kao detektor rubova

- detektira rubove metodom provjere prolaska kroz nulu (eng. zero-crossing)
- ne određuje orijentaciju i smjer ruba
- osjetljiv na šum

LoG - Laplacian of Gaussian

- primijenom Gaussovog glađenja prije Laplaceovog operatora dobivamo tkz. LoG operator (Laplacian of Gaussian)
- zbog asocijativnosti konvolucije prvo možemo konvoluirati Laplaceovu i Gaussovu jezgru, a zatim novu jezgru sa slikom:

$$\nabla^2[G_\sigma(x,y) * I(x,y)] = \nabla^2 G_\sigma(x,y) * I(x,y) = LoG_\sigma * I(x,y) \tag{4}$$

- algebarski podsjetnik:

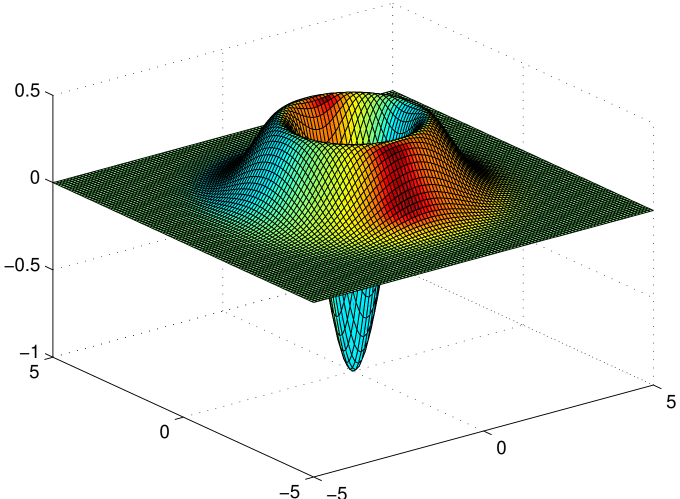
$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5}$$

$$LoG = \frac{\partial^2}{\partial x^2} G_\sigma(x,y) + \frac{\partial^2}{\partial y^2} G_\sigma(x,y) \tag{6}$$

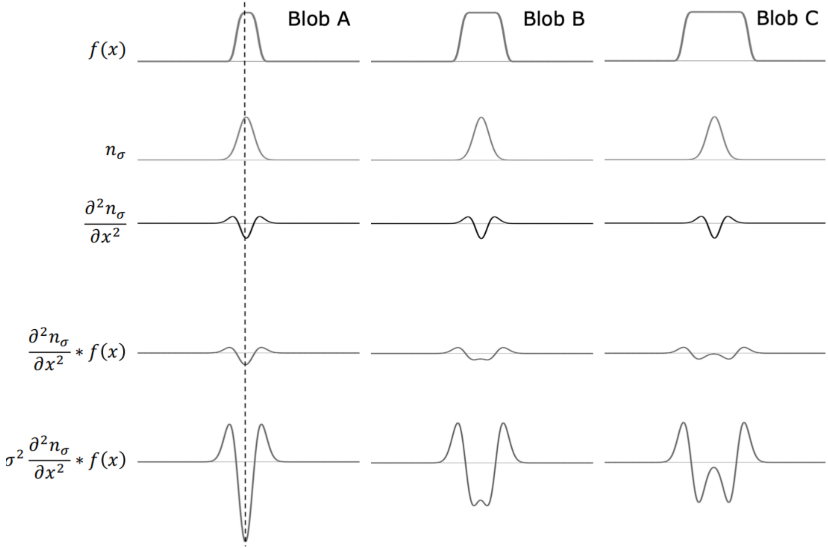
$$= \frac{x^2+y^2-2\sigma^2}{\sigma^4} e^{-(x^2+y^2)/2\sigma^2} \tag{7}$$

LoG - Laplacian of Gaussian

- izgled jezgre LoG operatora podsjeća na sombrero, pa je poznat i kao "mexican hat operator"



LoG kao detektor blobova - 1D



[Nayar22]

NLoG kao detektor blobova u slici

- blob - regija s pikselima međusobno sličnog intenziteta, ali različitog od njihove okoline
- blobovi često odgovaraju istaknutim dijelovima slike koje želimo detektirati
- za detekciju blobova možemo koristiti normalizirani LoG

$$NLoG = \sigma^2(\nabla^2 G_\sigma(x, y) * I(x, y)) \tag{8}$$

- ekstremni odziv u blobovima veličine prilagođene skali σ
- kako bismo pronašli blobove različitih veličina moramo koristiti NLoG s različitim vrijednostima $\sigma \rightarrow$ prostor mjerila

Pregled

Prostor mjerila

- prostor mjerila možemo definirati kao

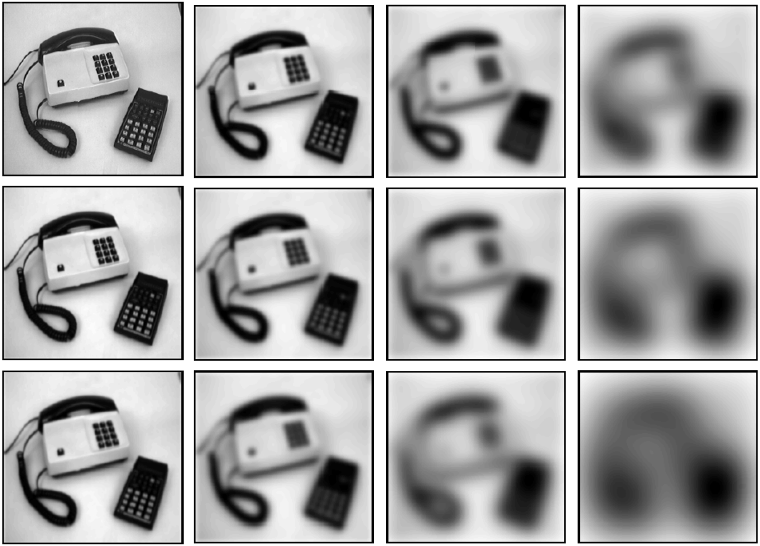
$$S(x, y, \sigma) = G_\sigma(x, y) * I(x, y) \tag{9}$$

- prostor mjerila odgovara funkciji slike proširene za parametar σ koji opisuje mjerilo
- pronalaskom ekstrema u prostoru mjerila pronalazimo lokacije blobova, ali i njihovo mjerilo σ koja je proporcionalna veličini bloba
- u praksi izaberemo neki diskretan skup mjerila
- zgodno svojstvo Gaussiana: $G_{\sigma_1}(x, y) * G_{\sigma_2}(x, y) = G_{\sqrt{{\sigma_1}^2 + {\sigma_2}^2}}(x, y)$

Zadatak - konstrukcija prostora mjerila

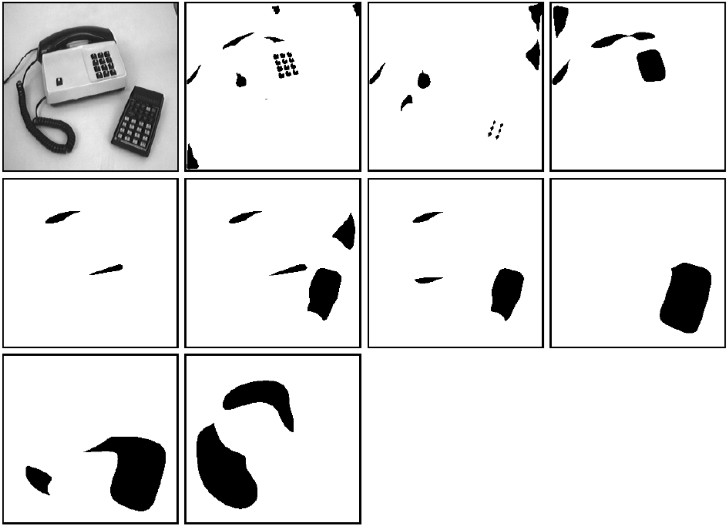
Razmatramo konstrukciju prostora mjerila prema pravilu $\sigma_k = \sigma_0 \cdot s^k$, gdje je σ_0 početno mjerilo, a s konstanta. Odredite s ako želimo stvoriti prostor mjerila s ukupno pet razina, gdje vrijedi $\sigma_4 = 2\sigma_0$. Odredite mjerilo za sve razine prostora mjerila, te mjerila Gaussovih jezgri ako prostor mjerila stvaramo uzastopnim konvolucijama, a $\sigma_0 = 1$.

Lindeberg - detekcija značajki u prostoru mjerila



[Lindeberg93]

Lindeberg - 50 najvažnijih blobova

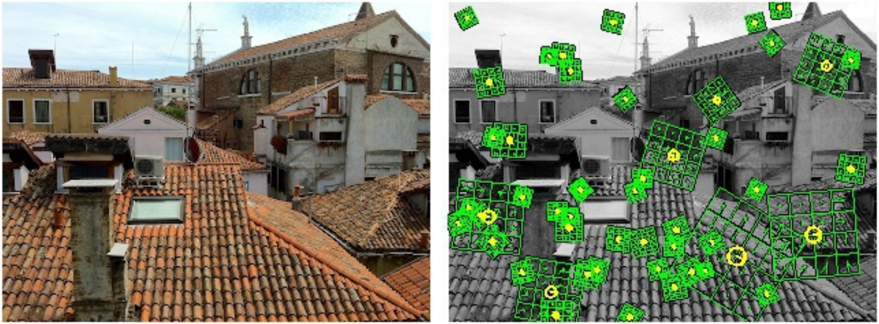


[Lindeberg93]

Pregled

Detekcija značajki u prostoru mjerila

- SIFT - scale invariant feature transform (Lowe, 1999 i 2004)
- detektira značajke invarijantne na skalu i rotaciju
- svakoj značajki pridružuje deskriptor s 128 elemenata



[vlfeat.org]

SIFT

- Koraci algoritma SIFT:
  1. Konstrukcija prostora mjerila i DoG aproksimacija LoG-a
  2. Pronalazak ekstrema u prostoru mjerila
  3. Lokalizacija ključnih točaka
  4. Izračun orijentacije ključnih točaka
  5. Izračun deskriptora ključnih točaka

SIFT - aproksimacija operatora NLoG

- algoritam SIFT ne primjenjuje izravno operator NLoG na sliku, nego ga aproksimira razlikom slika različitih mjerila (DoG - Difference of Gaussians)
- ova aproksimacija slijedi iz jednadžbe za difuziju topline:

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \tag{10}$$

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \tag{11}$$

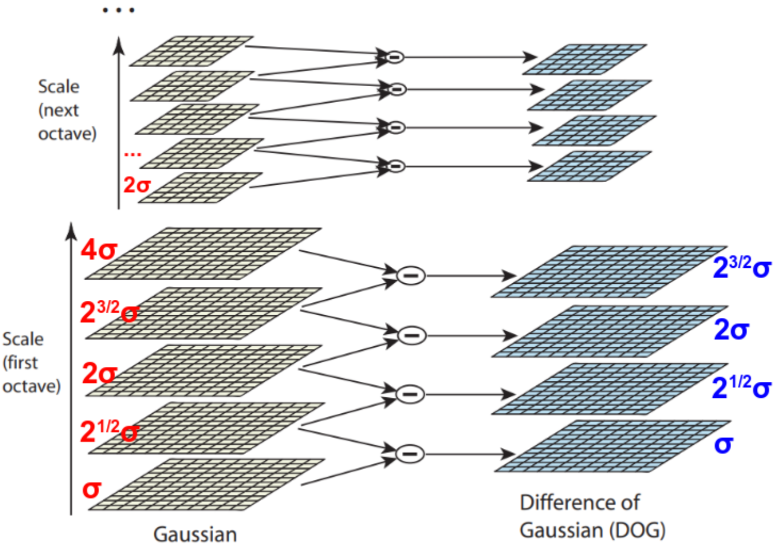
$$(k-1)\sigma^2 \nabla^2 G \approx G(x, y, k\sigma) - G(x, y, \sigma) \tag{12}$$

- ova aproksimacija nam omogućuje da prvo napravimo prostor mjerila, a zatim jednostavnim oduzimanjem slika susjednih mjerila dobijemo odzive NLoG operatora

SIFT - konstrukcija prostora mjerila

- SIFT-ov prostor mjerila sastoji se od niza oktava
- jednu oktavu čini niz slika čije se mjerilo postupno povećava
- sve slike unutar jedne oktave imaju istu prostornu rezoluciju, koja se polovi s prelaskom na sljedeću oktavu $\rightarrow$ piramida slika
- dizajn SIFT-ovog prostora mjerila kontroliramo sa sljedećim parametrima:
  - broj oktava
  - broj intervala S
  - početno mjerilo σ_0
- mjerilo susjednih slika unutar oktave tada se razlikuje s faktorom $k = 2^{1/S}$
- broj intervala odgovara broju mjerila na kojima tražimo lokalni ekstrem odziva DoG

SIFT - prostor mjerila za S=2



[Lowe04]

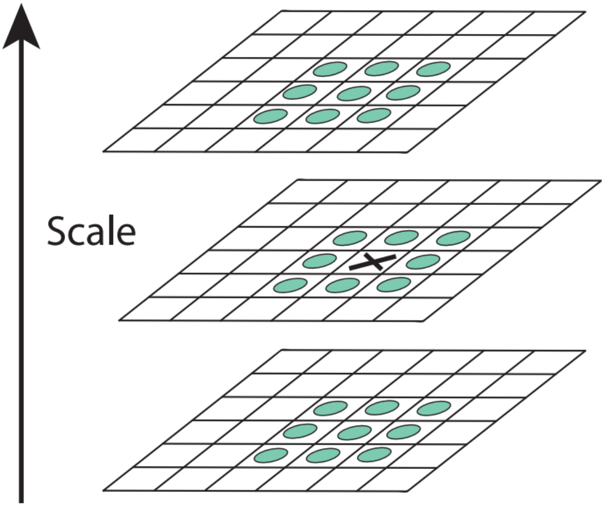
SIFT - prostor mjerila

- u praksi se pokazalo da je S=3 optimalan broj intervala
- članak također predlaže ulaznu sliku uvećati dva puta bilinearnom interpolacijom te započeti konstrukciju piramidalnog prostora mjerila od te rezolucije

SIFT - pronalazak ekstrema u prostoru mjerila

- lokalne ekstreme u prostoru mjerila pronalazimo provjerom maksimuma odziva DoG unutar pomične kocke dimenzija $3 \times 3 \times 3$
- kocka se dakle pomiče po prostornim koordinatama, ali i po dimenziji mjerila
- detekciju ekstrema provodimo zasebno za svaku oktavu

SIFT - pronalazak ekstrema u prostoru mjerila



[Lowe04]

SIFT - precizna lokalizacija ključnih točaka

- SIFT nakon detekcije lokalnih ekstrema na diskretnoj rešetci provodi preciznu lokalizaciju ključnih točaka
- to za rezultat ima subpikselsku preciznost
- postupak se temelji na aproksimaciji 3D kvadratne funkcije kroz 27 točaka unutar odgovarajuće pomične kocke

SIFT - precizna lokalizacija ključnih točaka

- pretpostavimo koordinatni sustav čije ishodište odgovara točki ekstrema na diskretnoj rešetci
- tada aproksimiramo funkciju $D(\mathbf{x})$ gdje $\mathbf{x} = (x, y, \sigma)$ odgovara odmaku od prostornih koordinata te mjerila razmatranog ekstrema
- funkciju možemo razviti u Taylorov red do kvadratnog člana

$$D(\mathbf{x}) = D(\mathbf{0}) + \nabla D(\mathbf{0}) \cdot \mathbf{x} + \frac{1}{2}\mathbf{x}^T \mathbf{H}(\mathbf{0})\mathbf{x} \tag{13}$$

- podsjetimo se da radimo razvoj oko točke ekstrema, pa konstante D, ∇D i $\mathbf{H}$ odgovaraju vrijednostima funkcije, prve derivacije odnosno Hesseove matrice u točki ekstrema na diskretnoj rešetci
- ako deriviramo razvijenu funkciju D i izjednačimo s nulom, dobivamo nove subpikselske koordinate ekstrema:

$$\mathbf{x}^* = -\mathbf{H}^{-1}(\mathbf{0}) \cdot \nabla D(\mathbf{0}) \tag{14}$$

SIFT - precizna lokalizacija ključnih točaka

- funkciju D ćemo još dodatno iskoristiti kako bismo odbacili nestabilne ekstreme s niskim kontrastom
- to radimo na način da evaluiramo funkciju D u novom ekstremu $\mathbf{x}^*$, te odbacimo sve ključne točke koje imaju $|D(\mathbf{x}^*)| < 0.03$

SIFT - precizna lokalizacija ključnih točaka

- dodatno u ovom koraku ćemo odbaciti ključne točke detektirane zbog visokog odziva DoG jezgre na rubovima
- razmatra se omjer veće i manje svojstvene vrijednosti Hesseove matrice na mjerilu ključne točke
- kod rubova taj omjer ima visoku vrijednost
- po uzoru na Harrisa umjesto izračuna svojstvenih vrijednosti, računa se njihov omjer preko determinante i traga matrice
- podsjetnik (α i β su svojstvene vrijednosti):

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{15}$$

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta \tag{16}$$

$$Det(\mathbf{H}) = D_{xx} \cdot D_{yy} - D_{xy}^2 = \alpha \cdot \beta \tag{17}$$

SIFT - precizna lokalizacija ključnih točaka

- pretpostavimo da je α veća svojstvena vrijednost i uvedemo konstantu r za koju vrijedi $\alpha = r\beta$

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{18}$$

- kako bismo provjerili omjer svojstvenih vrijednosti dovoljno je napraviti sljedeće

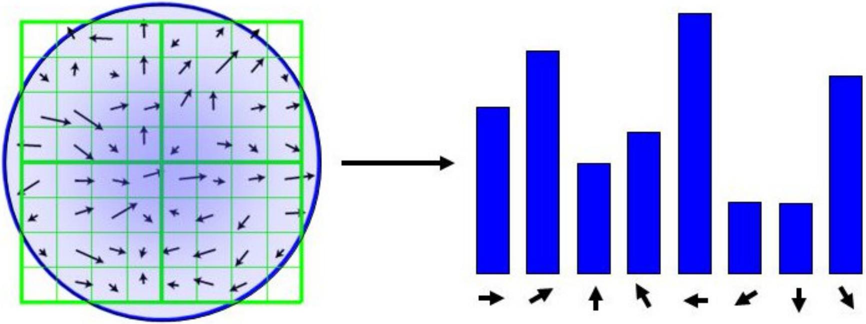
$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r_{\mathrm{prag}}+1)^2}{r_{\mathrm{prag}}} \tag{19}$$

- u praksi r_{prag} postavljamo na 10, te odbacujemo sve ključne točke koje ne ispunjavaju gornji uvjet

SIFT - orijentacije ključnih točaka

- određivanje orijentacije ključne točke je važno za postupak izlučivanja deskriptora invarijantnih na rotaciju
- u SIFT-u orijentaciju ključne točke određujemo na temelju histograma orijentacija gradijenata u njoj susjednim pikselima na slici najbližeg mjerila
- u ovome koraku histogram razlikuje 36 orijentacija što znači da svaki bin pokriva 10°
- svaka točka unutar regije susjedstva doprinosi histogramu s vrijednošću umnoška njene magnitude gradijenta i iznosa Gaussove funkcije odgovarajućeg mjerila centrirane u ključnoj točki
- dakle najveći utjecaj na orijentaciju ključne točke imaju bliski susjedi s velikom magnitudom

SIFT - orijentacije ključnih točaka



SIFT - orijentacije ključnih točaka

- nakon prikupljanja histograma pronalazimo njegov maksimum, te također sve orijentacije čija je vrijednost (frekvencija) barem 80% maksimalne
- za svaku orijentaciju koja zadovoljava taj uvjet stvaramo po jednu ključnu točku
- možemo dakle imati više ključnih točaka na istoj lokaciji, ali različitih orijentacija
- kako bismo preciznije odredili orijentaciju ključne točke pronalazimo parabolu koja najbolje opisuje odgovarajući ekstrem i njegove susjedne vrijednosti u histogramu
- konačno orijentacija ključne točke odgovara tjemenu te parabole

SIFT - izračun deskriptora

- deskriptor ključne točke odgovara konkatenaciji orijentacijskih histograma izračunatih unutar disjunktnih 4×4 prozora u susjedstvu veličine 16×16
- u ovome koraku histogram razlikuje 8 mogućih orijentacija
- zbog toga je dimenzionalnost SIFT deskriptora značajki 128
  - $128 = 16 \cdot 8$
  - $16 =$ broj različitih prozora dimenzija 4×4 u susjedstvu dimenzija 16×16
  - $8 =$ dimenzija histograma kojeg računamo unutar svakoga prozora

SIFT - postupak izračuna deskriptora

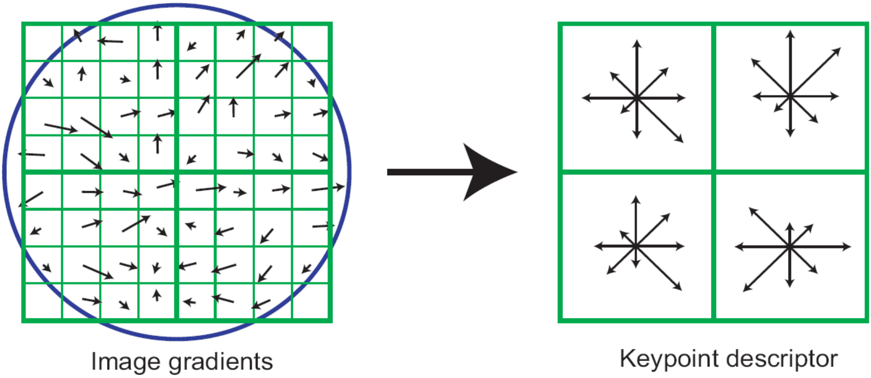
- na slici koja odgovara mjerilu razmatrane ključne točke izračunamo gradijente, te magnitude i orijentacije svih piksela
- lokaciju i orijentaciju svakog piksela iz susjedstva rotiramo prema orijentaciji razmatrane ključne točke $\rightarrow$ invarijantnost na rotaciju!
- prema toj rotiranoj lokaciji određujemo pripadnost te točke jednom od disjunktnih prozora odnosno histogramu

SIFT - postupak izračuna deskriptora

- slično kao i ranije doprinos svakog piksela iz susjedstva određen je umnoškom njegove magnitude i težine određene prema Gaussu
- centar ovog Gaussa postavi se u razmatranu ključnu točku, a njegova standardna devijacija na mjerilo ključne točke uvećano faktorom 1.5
- kako bi se poboljšala invarijantnost na lokalne pomake, a posebice u graničnim područjima, doprinos svakog piksela se prema trilinearnoj interpolaciji također dijeli na susjedne histograme te orijentacije
- na kraju se svi histogrami konkateniraju u jedan vektor koji se zatim:
  1. normalizira na jediničnu udaljenost
  2. vrijednost svakog elementa vektora se ograniči na maksimalno 0.2
  3. ponovno normalizira na jediničnu udaljenost

SIFT - ilustracija izračuna deskriptora

- ilustracija izračuna deskriptora za susjedstvo veličine 8×8 podijeljeno na 4 prozora veličine 4×4
- dimenzija deskriptora u ovom slučaju bila bi $4 \times 8 = 32$

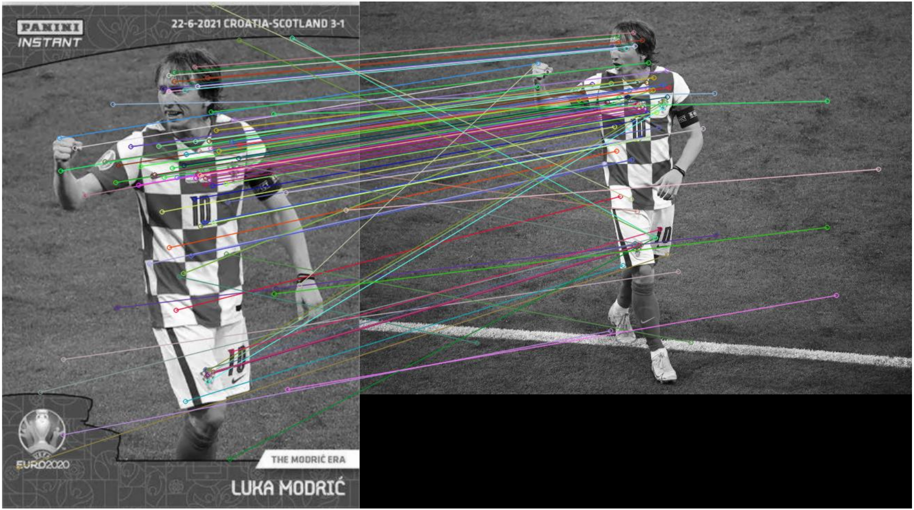


[Lowe04]

SIFT - primjena na uparivanje ključnih točaka

- deskriptore možemo uspoređivati pomoću L2 udaljenosti

SIFT - primjena na uparivanje ključnih točaka



SIFT - primjena na panoramsko spajanje slika

(a) Image 1 (b) Image 2

(c) SIFT matches 1 (d) SIFT matches 2

(e) RANSAC inliers 1 (f) RANSAC inliers 2

(g) Images aligned according to a homography

[Brown07]

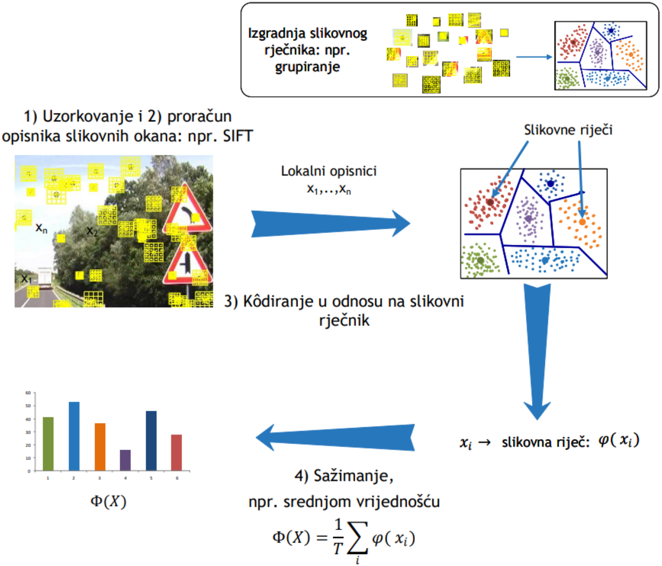
SIFT - primjena na pronalazak primjeraka

[Lowe04]

SIFT - primjena na pronalazak primjeraka

[Lowe04]

SIFT - primjena na klasifikaciju slika



Materijali

1. Članak: `https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf`
2. Neslužbena implementacija u pythonu:
`https://medium.com/@russmislam/`
`implementing-sift-in-python-a-complete-guide-part-1-306a9`
3. Video predavanja 4 i 12-16: `https://www.youtube.com/playlist?`
`list=PL2zRqk16wsdqXEMpHrc4Qnb5rA1Cylrhx`