Računalni vid

Elementarne slikovne značajke

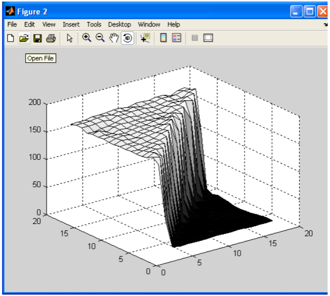
Siniša Šegvić
UniZg-FER D307
Sveučilište u Zagrebu

Plan

1. Gradijenti i glađenje
2. Rubovi
3. Ključne točke

Gradijent: slika kao funkcija

Slike često tretiramo kao kontinuirane funkcije



[collins07cse486]

Prednost: analiza primjenom alata matematičke analize

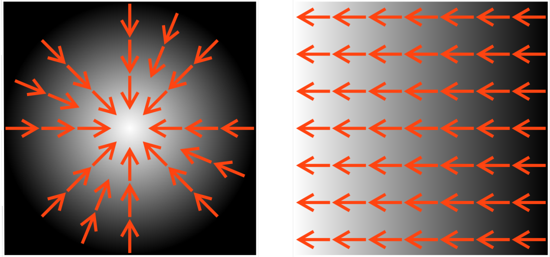
- rubovi objekata imaju veliki gradijent
- okna s "nejednolikim" gradijentom pogodna za **korespondenciju**
- u drugoj slici iste scene, lakat ćemo lakše prepoznati od nadlaktice

Gradijent: formulacija

Gradijent kontinuirane skalarne funkcije dvije varijable:

$$\nabla I(\mathbf{q}) = \partial I / \partial \mathbf{q} = [I_x(\mathbf{q}), I_y(\mathbf{q})]$$

Gradijent možemo interpretirati kao smjer najbržeg rasta sive razine:

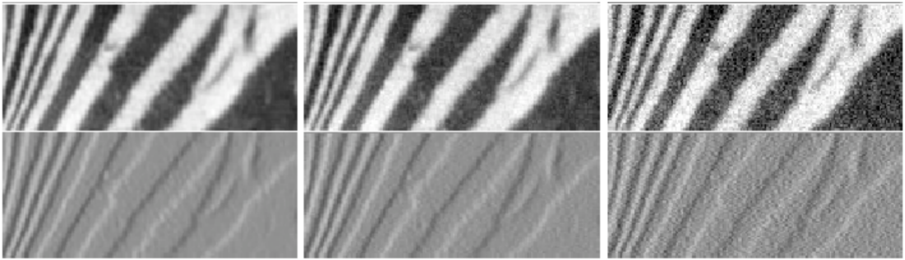


[Wikipedia]

Gradijent: šum

Međutim, deriviranje se loše ponaša u prisustvu šuma

- prvi redak: slike s rastućim udjelom šuma
- drugi redak: aproksimacija negativnog horizontalnog gradijenta konačnim razlikama



[ponce12book]

Vidimo da deriviranje naglašava šum

- zbog toga prije deriviranja obično primjenjujemo glađenje

Gradijent: glađenje

Jače glađenje tolerira više šuma ali i gubi fine detalje

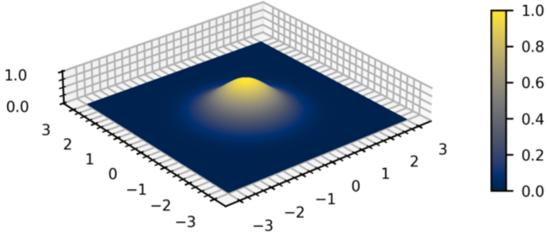
[lindeberg08wiley]

Gradijent: Gauss

Optimalnu detekciju rubova osigurava glađenje koje možemo aproksimirati konvolucijom s dekoreliranim Gaussom [canny86pami]

- svaki piksel mijenjamo s prosjekom njegovog susjedstva
- bliži susjedi su važniji
- važno svojstvo: separabilnost

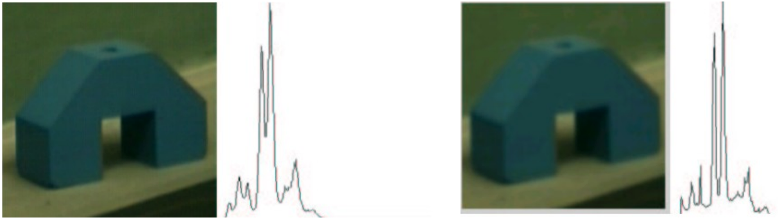
$$f(x, y) = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2 + y^2}{2\sigma^2})$$



[Wikipedia]

Gradijent: Gauss (2)

Glađenje je elementarna operacija predobrade koju ima smisla koristiti i ako ne ciljamo na rubove

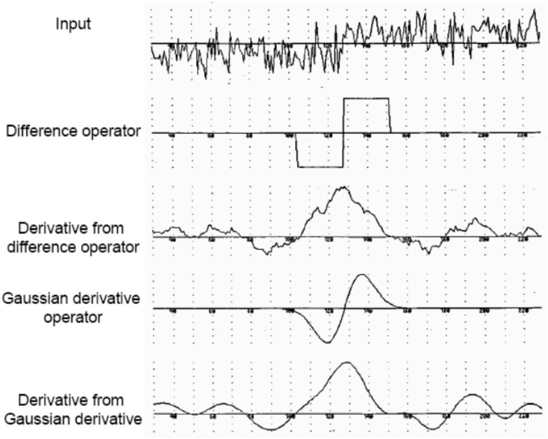


[segvic00ms]

GRADIJENT: RECEPT

Aproksimirati gradijente konvolucijom s derivacijom Gaussovog filtra

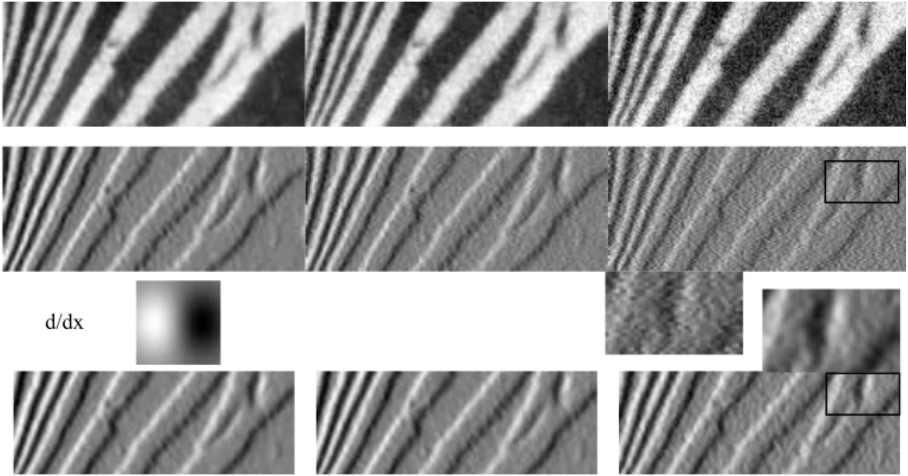
- dva koraka mogu se stopiti u jedan
- konvolucije i derivacije su asocijativne



[canny86pami]

Gradijent: recept (2)

Gradijente je bolje aproksimirati konvolucijom s derivacijom Gaussa nego konačnim diferencijama



[ponce12book]

Gradijent: separabilna izvedba

I gauss i derivirani Gauss su separabilni 2D filtri: 2D konvolucija s takvim filtrima može se izvesti uz pomoć dvije 1D konvolucije.

Ako nam se žuri, gradijente možemo aproksimirati i konačnim razlikama u zaglađenoj slici.

Jako važno za primjene u stvarnom vremenu, posebno ako postupak koristi slikovne piramide.

Gaussov filtar može se aproksimirati rekurzivnim binomnim filtrom (dodatno ubrzanje).

Rubovi

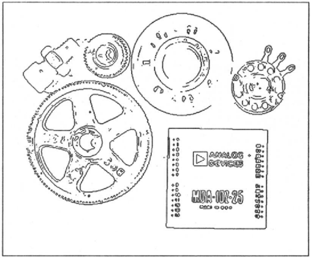
Rubovi upućuju na granice objekata:

- **oblik**: inherentno svojstvo objekta
- teže ga je sakriti od **teksture**
- organizmi koji vide oblik imaju evolucijsku prednost

[biologyonline,flickr]

Rubovi: motivacija

Rane metode iz romantičarske faze računalnog vida imale su ambiciju detektirati objekte analizom rubova:



[canny86pami]

- kasnije smo spoznali da to ipak ne bi išlo samo tako :-)
- da bi znao iscrtati rub objekta - moraš razumjeti gdje je objekt
- gradijent je presirova informacija za tako zahtjevan zadatak
- ipak, stečena iskustva mogu dati korisnu intuiciju pri razvoju modernih metoda

Rubovi: Cannyjev algoritam

Jedan od najstarijih i najdosljednijih detektora rubova

Glavni koraci algoritma:

1. računanje gradijenta konvolucijom s deriviranim Gaussom
   - možemo i aproksimirati konačnim razlikama u zaglađenoj slici
   - ne puno lošije, do 2× brže
2. stanjivanje rubova potiskivanjem ne-maksimalnih odziva u smjeru pružanja gradijenta
   - ideja je rubove smjestiti u točkama infleksije profila sive razine uzduž smjera gradijenta
   - dobro za lokalizaciju rubova
3. (opcionalno) potisnuti rubove s malenim gradijentom usporedbom s unaprijed zadanim histereznim pragom

Rubovi: Cannyjev algoritam (2)

1. Računanje gradijenta konvolucijom s deriviranim Gaussom
   - tipično - Sobelov operator

X – Direction Kernel

-1	0	1
-2	0	2
-1	0	1

Y – Direction Kernel

-1	-2	-1
0	0	0
1	2	1

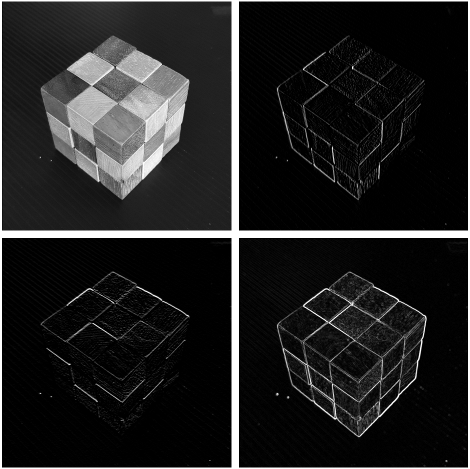
   - rezultat: dvije slike gradijenata, u x i y smjeru (G_x, G_y)
   - iznos i smjer gradijenta:

$$|G| = \sqrt{G_x^2 + G_y^s}$$

$$\theta = arctan(G_y/G_x)$$

Rubovi: Cannyjev algoritam (3)

1. Računanje gradijenta konvolucijom s deriviranim Gaussom (2)
   - Sobelov operator - primjer: gornji red: izvorna slika, G_x; donji red: G_y, $|G|$

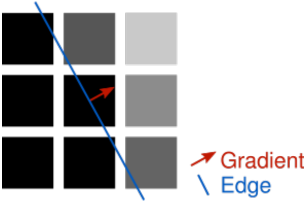


Napomena: G_x i G_y mogu biti i pozitivni i negativni - na slici su prikazane apsolutne vrijednosti

RUBOVI: CANNYJEV ALGORITAM (4)

2. Stanjivanje rubova potiskivanjem ne-maksimalnih odziva u smjeru pružanja gradijenta

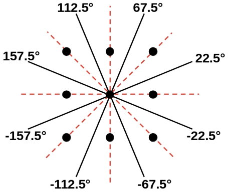
- rezultat računanja gradijenata je slika rubova, ali često širokih i nekoliko piksela (neprecizna lokalizacija)
- težimo preciznijoj lokalizaciji značajki
- ideja: svaki piksel u slici $|G|$ usporediti s njegova dva susjeda u smjeru gradijenta (okomit na smjer pružanja ruba) i zadržati samo one koji imaju veći iznos gradnijenta od svojih susjeda (ostale postavljamo na 0)

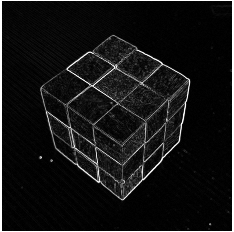


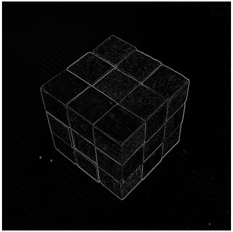
RUBOVI: CANNYJEV ALGORITAM (5)

2. Stanjivanje rubova potiskivanjem ne-maksimalnih odziva u smjeru pružanja gradijenta (2)

- pri tome, smjerove gradijenta diskretiziramo u 8 pravaca



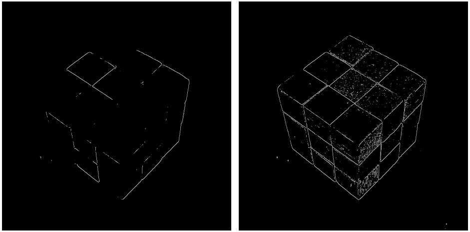




Rubovi: Cannyjev algoritam (6)

3. Potiskivanje rubova s malenim gradijentom usporedbom s unaprijed zadanim histereznim pragom

- binarizacija rubova usporedbom s pragovima
- $|G| > T_1$ - sigurno rub ("jaki rubovi")
- $T_1 \geq |G| \geq T_2$ - potencijalno rub (slabi rubovi)
- $|G| < T_2$ - sigurno nije rub



lijevo - jaki rubovi; desno - svi rubovi

Rubovi: Cannyjev algoritam (7)

Literatura: https://ieeexplore.ieee.org/document/4767851

Detalji: https://cvunizg.github.io/lab/lab2/

Ključne točke

Značajka: istaknuti dio slike

- izgled i položaj **prepoznatljivi** iz različitih dijelova scene (kut, blob, ...)
- zbog efikasnije implementacije, najčešći kvadratni oblik (koriste se i pravokutnici, paralelogrami, krugovi, ...)

veličina značajke: kompromis između **otpornosti na zaklanjanje** i neprepoznatljivosti

svojstveno mjerilo i rotacija: noviji koncept, problemi u stvarnom vremenu (GPGPU!)

Pronalaženje kuteva

Značajka: istaknuti dio slike

- izgled i položaj **prepoznatljivi** iz različitih dijelova scene (kut, blob, ...)
- zbog efikasnije implementacije, najčešći kvadratni oblik (koriste se i pravokutnici, paralelogrami, krugovi, ...)

veličina značajke: kompromis između **otpornosti na zaklanjanje** i neprepoznatljivosti

svojstveno mjerilo i rotacija: noviji koncept, problemi u stvarnom vremenu (GPGPU!)

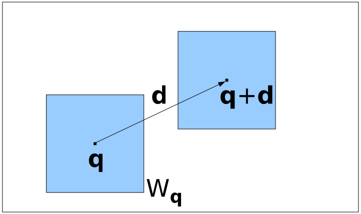
Pronalaženje kuteva

Harrisovi kutevi: unatoč poodmakloj dobi (1988), jedan od najefikasnijih pristupa s fiksnim mjerilom

Ideja: pronaći točke $\mathbf{q}_{zn}$ koje maksimiraju lokalnu prepoznatljivost susjedstva zadane veličine

Promotrimo ``autokorelacijsku funkciju'' (zapravo **SSD**):

$$E_{\mathbf{q}}(\mathbf{d}) = \sum_{x,y \in W(\mathbf{q})} [\mathrm{I}(x+d_x, y+d_y) - \mathrm{I}(x,y)]^2$$



$E_{\mathbf{q}}(\mathbf{d})$: različitost susjedstva točke $\mathbf{q}$, u ovisnosti o pomaku $\mathbf{d}$

Pronalaženje kuteva

Ideja: značajke su u točkama gdje $E_{\mathbf{q}}$ raste u svim smjerovima!

Tražimo kriterij koji će otkriti točke $\mathbf{q}(x_{zn}, y_{zn})$ gdje to vrijedi.

Rezultat: U svakoj točki slike računa se matrica **G**

- **G** sadrži lokalne iznose momenata drugog reda gradijenta slike

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} \sum_W \mathrm{I}_x^2 & \sum_W \mathrm{I}_x \mathrm{I}_y \\ \sum_W \mathrm{I}_x \mathrm{I}_y & \sum_W \mathrm{I}_y^2 \end{bmatrix} = \begin{bmatrix} a & c \\ c & b \end{bmatrix} \tag{1}$$

Harrisov odziv r sada je: $r(\mathbf{q}) = ab - c^2 - k(a+b)^2$

Konačno, značajke su u lokalnim maksimumima odziva:

$\mathbf{q}_{zn} = max_{\mathbf{q}} r(\mathbf{q})$

Pronalaženje kuteva

Izvod Harrisove formule

Zapišimo E u vektorskom obliku:

$$\begin{aligned} E_{\mathbf{q}}(\mathbf{d}) &= \sum_{x,y \in W(\mathbf{q})} [\mathrm{I}(x + d_x, y + d_y) - \mathrm{I}(x, y)]^2 \\ &= \sum_{\mathbf{q}_i \in W(\mathbf{q})} [\mathrm{I}(\mathbf{q}_i + \mathbf{d}) - \mathrm{I}(\mathbf{q}_i)]^2 \end{aligned}$$

Pronalaženje kuteva

Izvod Harrisove formule (2)

Razvijmo $\mathrm{I}(\mathbf{q}_i + \mathbf{d})$ po Tayloru oko $\mathbf{q}_i$:

$$\begin{aligned} E_{\mathbf{q}}(\mathbf{d}) &= \sum_{\mathbf{q}_i \in W_{\mathbf{q}}} [\mathrm{I}(\mathbf{q}_i + \mathbf{d}) - \mathrm{I}(\mathbf{q}_i)]^2 \\ &\approx \sum_{\mathbf{q}_i \in W} [\mathrm{I}(\mathbf{q}_i) + \nabla \mathrm{I}(\mathbf{q}_i) \cdot \mathbf{d} - \mathrm{I}(\mathbf{q}_i)]^2 \end{aligned}$$

$$\begin{aligned} E_{\mathbf{q}}(\mathbf{d}) &\approx \sum_{\mathbf{q}_i \in W} [g^\top \mathbf{d}]^2 = \sum_{\mathbf{q}_i \in W} \mathbf{d}^\top (g g^\top) \mathbf{d} \\ &= \mathbf{d}^\top \cdot \mathbf{G} \cdot \mathbf{d}, \quad \text{gdje je} \quad \mathbf{G} = \sum_W g g^\top \end{aligned}$$

Podsjetnik: $\nabla \mathrm{I}(\mathbf{q}_i) = g^\top = [\mathrm{I}_x(\mathbf{q}_i), \mathrm{I}_y(\mathbf{q}_i)]$

Pronalaženje kuteva

Izvod Harrisove formule (3)

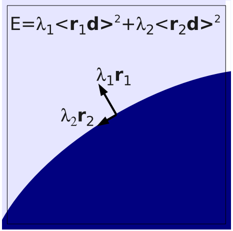
Matrica G je realna i simetrična pa se može dijagonalizirati:

$$\mathbf{G} = \begin{bmatrix} \sum_W \mathrm{I}_x^2 & \sum_W \mathrm{I}_x\mathrm{I}_y \\ \sum_W \mathrm{I}_x\mathrm{I}_y & \sum_W \mathrm{I}_y^2 \end{bmatrix} = \mathbf{R}^\top\mathbf{D}\mathbf{R},\ \text{gdje je}\ \ \mathbf{R} = \begin{bmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \end{bmatrix}$$

Sada kriterijsku funkciju možemo izraziti u koordinatama koje su poravnate s lokalnom strukturom slike:

$$E_{\mathbf{q}}(\mathbf{d}) = \mathbf{d}^\top \cdot \mathrm{G} \cdot \mathbf{d} = \sum_{\mathbf{q}_i \in W} \mathbf{d}_{\mathrm{R}}^\top \cdot \mathrm{D} \cdot \mathbf{d}_{\mathrm{R}},$$

$$\mathbf{d}_{\mathrm{R}} = \mathrm{R} \cdot \mathbf{d} = [\mathbf{r}_1^\top \mathrm{d}, \mathbf{r}_2^\top \mathrm{d}]^\top$$

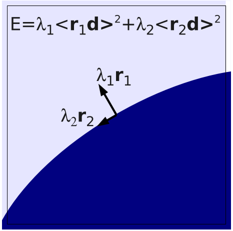


Pronalaženje kuteva

Izvod Harrisove formule (4)

U novim koordinatama, doprinosi komponenti pomaka **nezavisno** utječu na kriterijsku funkciju E!

$$
\begin{aligned}
E_{\mathbf{q}}(\mathbf{d}) &= \mathbf{d}^\top \cdot \mathbf{G} \cdot \mathbf{d} = \mathbf{d_R}^\top \cdot \mathbf{D} \cdot \mathbf{d_R} \\
&= \mathbf{d_R}^\top \cdot \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \cdot \mathbf{d_R} \\
&= \lambda_1 d_{R1}^2 + \lambda_2 d_{R2}^2 \\
&= \lambda_1 (\mathbf{r}_1^\top \mathbf{d})^2 + \lambda_2 (\mathbf{r}_2^\top \mathbf{d})^2
\end{aligned}
$$



Slika se lokalno brzo mijenja u svim smjerovima $\Leftrightarrow$ λ_1, λ_2 veliki!

Pronalaženje kuteva

Harrisov odziv ``nagrađuje'' točke u kojima su veliki i λ_1 i λ_2:

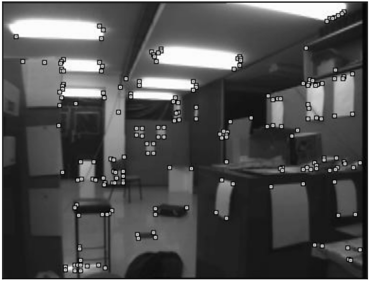
$$\mathbf{G} = \begin{bmatrix} \sum_W \mathrm{I}_x^2 & \sum_W \mathrm{I}_x \mathrm{I}_y \\ \sum_W \mathrm{I}_x \mathrm{I}_y & \sum_W \mathrm{I}_y^2 \end{bmatrix} = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$$

$$\begin{aligned} r &= ab - c^2 - k \cdot (a+b)^2 \qquad k \in [0.04, 0.07] \\ &= \det(\mathbf{G}) - k \cdot (tr^2(\mathbf{G}) \\ &= \lambda_1 \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2 \end{aligned}$$

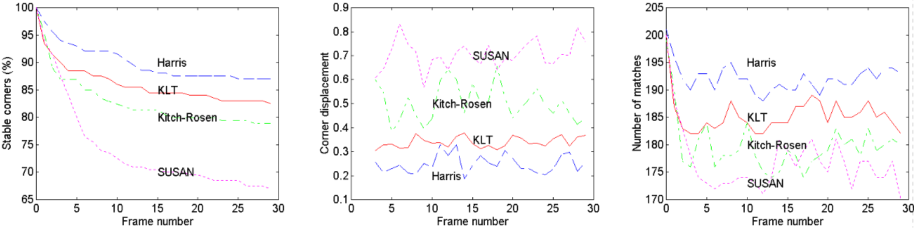
Konačni rezultat su lokalni maksimumi odziva

Potiskivanje ne-maksimalnih odziva dovodi do dobre lokalizacije značajki

Pronalaženje kuteva



Broj značajki (a), lokalizacija (b), te uspješnost uparivanja (c):

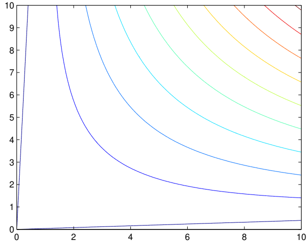


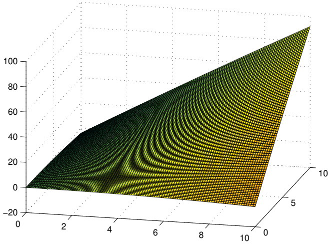
[tissainayagam04imavis]

Pronalaženje kuteva

Oblik funkcije odziva:

$$r = \lambda_1 \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2$$

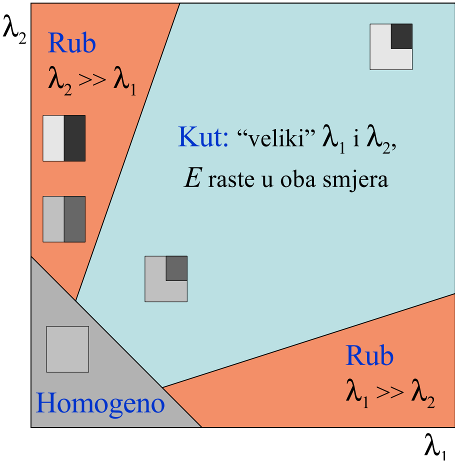




Pronalaženje kuteva

Oblik funkcije odziva (2):

$$r = \lambda_1 \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2$$



Pronalaženje kuteva

Detalji:

- težinska funkcija? $\mathbf{G} = \sum_W w(x, y) \begin{bmatrix} \mathrm{I}_x^2 & \mathrm{I}_x\mathrm{I}_y \\ \mathrm{I}_x\mathrm{I}_y & \mathrm{I}_y^2 \end{bmatrix}$
- derivacijsko i integracijsko mjerilo?

Svojstva:

- odziv (pa onda i maksimum) invarijantan na rotaciju slike!
- dobra ponovljivost i lokalizacija rezultata (barem za jednostavne značajke)
- problem: odziv nije invarijantan na mjerilo!

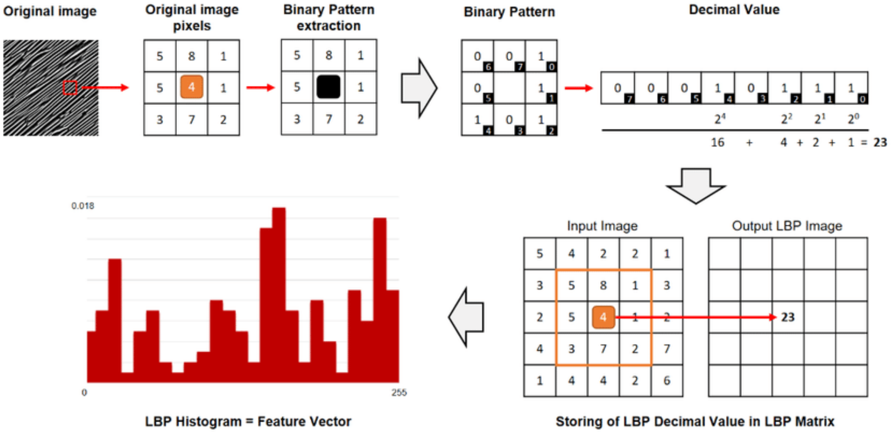
Alternative:

- FAST (brži); DoG, MSER, SURF (invarijantni na mjerilo, sporiji)

Lokalni binarni uzroci

- **LBP** (engl. local binary patterns)
- jednostavan operator za opis teksture
- neosjetljiv na promjene u osvjetljenju
- slična ideja: **Census** transformacija
- primjene
  - opis tekstura
  - podudaranje značajki
  - prisustvo rubova i kuteva

Lokalni binarni uzroci



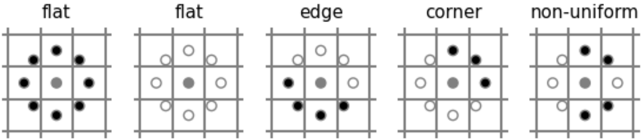
Lokalni binarni uzroci

Postupak: za svaki piksel u izvornoj (sivoj) slici:

- usporedi vrijednost piksela sa svim pikselima u nekom njegovom susjedstvu (tipično 3×3)
- ako je središnji piksel manji od piskela okoline, to označavamo s 0
- ako je središnji piksel veći od piksela okoline, to označavamo s 1
- rezultat: bitovni niz koji očitavamo nekim predefiniranim redoslijedom i pretvaramo u dekadski broj
- (kod analize tekstura slika se dijeli na ćeije i za svaku ćeliju konstruira se LBP histogram - opis lokalne teksture)

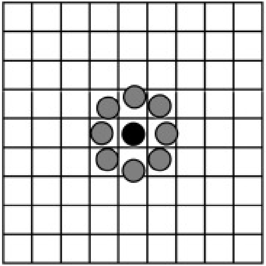
Lokalni binarni uzroci

- LBP enkodira prisustvo tipičnih značajki (rubova, kuteva, ...) u promatranom lokalnom području

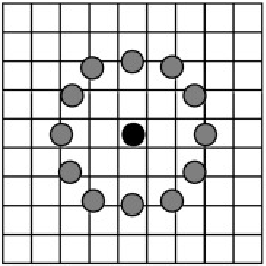


Lokalni binarni uzroci

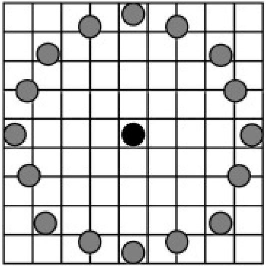
- Moguće su i drukčije veličine okoline



P=8, R=1



P=12, R=2



P=16, R=4

ZAHVALA

Ova predavanja proizišla su iz istraživanja koje je financirala Hrvatska zaklada za znanost projektom ADEPT.

`http://adept.zemris.fer.hr`