



Digitalna obrada slike

Lekcija X

DETEKCIJA IVICA



Važnost ivica

- Od početka kursa naučeni smo da kod obrade slike postoji zakonomjernost: niskopropusni dio frekvencijskog područja slike nosi značajan dio energije ali je relativno malo upotrebljiv sa stanovišta vizuelnog kvaliteta slike kao i za mašinsku viziju; visokopropusni dio spektra slike međutim je male energije ali nosi značajan dio informacija kako za ljudsku tako i za mašinsku viziju. Osnovni problem visokofrekventnog dijela je mala energija tako da ovaj dio spektra brzo “utone” u šum.
- Najvažniji dio visokofrekventnog dijela spektra su **ivice**: **objekat i slika su potpuno prepoznatljivi samo na osnovu ivica; ivice se mogu zapisati u binarnoj formi i kao takve efikasno obrađivati; na osnovu detektovanih ivica može se podešavati veličina lokalnog susjedstva kod algoritama za filtriranje itd.**



Što je ivica?

- Nema precizne definicije.
- Približna definicija je da je ivica zona slike u kojoj dolazi do nagle promjene osvjetljaja.
- Kako smo u dosadašnjoj praksi detektovali nagle promjene neke funkcije.
- Jedan od načina je na osnovu velikih vrijednosti izvoda koje sugerišu da je u posmatranoj tački došlo do promjene veličine funkcije.
- **Da li možemo ovakvu strategiju primjeniti kod ivica?**
- Uz malo prilagođavanja možemo.



Magnituda izvoda

- Izvod se definiše kao:

$$f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

- Za 2D funkcije izvod se definiše kao:

$$\nabla f(x, y) = [f_x(x, y), f_y(x, y)] = \left[\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right]$$

↘ **Parcijalni izvodi po x i y**

Nagli prelaz sa jednog na drugi osvjetljaj se može detektovati preko magnitude (amplitude ivice) koja se može računati sa:

$$e(x, y) = \sqrt{f_x^2(x, y) + f_y^2(x, y)}$$



Izvod slike i ugao ivice

- Ponekad se $e(x,y)$ računa alternativno kao:

$$e(x, y) = |f_x(x, y)| + |f_y(x, y)|$$

- Lokalni pravac izvoda (ivice) se može računati kao:

$$\varphi(x, y) = \arctan\left(\frac{f_y(x, y)}{f_x(x, y)}\right)$$

- Očigledno da ćemo veličinu $e(x,y)$ (ili neku alternativu) koristiti kao detektor ivice tako što ćemo je porediti sa nekim pragom a na osnovu $\varphi(x,y)$ ćemo pokušati da dođemo do još neke dodatne informacije o ivici.



Detektor ivice i difference

- Osnovna ideja kod detektora ivica je da na osnovu poređenja odziva detektora $e(x,y)$ sa nekim **pragom** koji može biti zavisian od pozicije $T(x,y)$ donese binarna odluka: jeste ili nije ivica. Stoga je sama slika koja predstavlja ivice binarna:

$$iv(x, y) = \begin{cases} 1 & e(x, y) \geq T(x, y) \longrightarrow (x,y) \text{ pripada ivici} \\ 0 & e(x, y) < T(x, y) \longrightarrow (x,y) \text{ ne pripada ivici} \end{cases}$$

Prvi problem koji je lako uočljiv je činjenica da mi ne raspoložemo sa kontinualnom slikom i da umjesto izvoda moramo koristiti **difference** (**konačne razlike**) kao procjene izvoda.



Problem diferenci

- Kako su kod nas pikseli razmaknuti za 1 to možemo na osnovu relacije za izvod uvesti sledeću aproksimaciju izvoda:

$$\Delta f_x(n, m) = f(n+1, m) - f(n, m)$$

$$\Delta f_y(n, m) = f(n, m+1) - f(n, m)$$

Drugi krupan problem je činjenica što konačne razlike pojačavaju šum.

sa (\mathbf{x}, \mathbf{y}) prešli smo na piksele (\mathbf{m}, \mathbf{n}) .

Pretpostavimo da u slici nema ivice i da je u pitanju samo Gausov bijeli šum sa varijansom σ^2 i da je šum nezavisan od piksela do piksela (stacionaran). Tada je varijansa konačne razlike po x-koordinati:

$$\begin{aligned}
 E\{(\Delta f_x(n, m))^2\} &= E\{[v(n+1, m) - v(n, m)]^2\} \xrightarrow{=0} \\
 &= E\{v^2(n+1, m)\} - 2E\{v(n+1, m)v(n, m)\} + E\{v^2(n, m)\} = 2\sigma^2
 \end{aligned}$$

uvećana varijansa



Kako prevazići problem šuma?

- Osnovna ideja kod prevazilaženja problema asociranih sa šumom je usrednjiti jedan broj piksela normalno u odnosu na pravac ivice. Dakle, za računanje ivice u pravcu x-koordinate u tački (n,m) ne koristiti samo osvjetljaje u tačkama $(n,m+1)$ i (n,m) već i osvjetljaje u pikselima $(n+1,m+1)$ i $(n+1,m)$ te $(n-1,m+1)$ i $(n-1,m)$.
- Čitava klasa detektora ivica u koju spadaju i popularni **Sobelov** i **Prewittov** detektori je zasnovana na ovom principu.
- Prije nego objasnimo detektore ove grupe prikazaćemo Robertsov detektor. Ovaj detektor koristi 4 susjedna piksela za računanje odziva detektora.



Robertsov detektor

- Ovo je izuzetno jednostavan detektor čiji se odziv računa kao:

$$e(n, m) = \sqrt{[\sqrt{f(n, m)} - \sqrt{f(n+1, m+1)}]^2 + [\sqrt{f(n+1, m)} - \sqrt{f(n, m+1)}]^2}$$

Vidimo da koristimo samo 4 susjedna piksela za određivanje detektora ivica koji ne daje informacije o pravcu ivice.

Prilikom dizajna ovog detektora vođeno je računa o ljudskom vizuelnom sistemu i po nalazima Roberta ovo su operacije koje ljudi koriste da bi odredili ivice. Ne smatra se među najboljim detektorima.



Maska

- Najšire korištena grupa filtara je zasnovana na **maski**. Maska je 3×3 konvoluciona matrica ($e(n,m)$ se može računati preko konvolucije).
- Dizajniraju se dvije matrice da bi se izvršila detekcija ivica duž pravca x- i y-ose.
- Pravila prilikom selekcije matrica.
 - Matrica koja se koristi da za detekciju ivice duž pravca x ose mora biti ista kao ona koja se koristi za detekciju ivica duž y ose ali rotirana za $\pi/2$ ili transponovana. Na ovaj način se isti značaj daje ivicama u ova dva normalna pravca. Mi ćemo posmatrati detekciju u pravcu x-ose i neka je maska:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$



Dizajn maske

- Centralni piksel je piksel od interesa. Mi želimo oduzeti piksele koji odgovaraju pozicijama a_{12} i a_{32} . Međutim, koristimo i dva susjedna piksela da pripomognu u cilju umanjivanja uticaja šuma. Jasno je da lijevi i desni piksel treba uzeti sa istom jačinom pa na osnovu ovoga slijedi da $a_{11}=a_{31}$, $a_{21}=a_{23}$ i $a_{31}=a_{33}$:

$$\begin{bmatrix} a_{11} & a_{12} & a_{11} \\ a_{21} & a_{22} & a_{21} \\ a_{31} & a_{32} & a_{31} \end{bmatrix}$$

- Pikseli u donjem redu se oduzimaju od piksela u gornjem redu stoga treba postaviti te koeficijente da imaju različit predznak:

$$\begin{bmatrix} a_{11} & a_{12} & a_{11} \\ a_{21} & a_{22} & a_{21} \\ -a_{11} & -a_{12} & -a_{11} \end{bmatrix}$$



Dizajn maske

- Ako je slika uniformna u lokalnom susjedstvu to znači da nema ivice i da odziv detektora ivice treba da bude 0. Stoga treba da važi da je suma koeficijenata maske 0:

$$\begin{bmatrix} a_{11} & a_{12} & a_{11} \\ a_{21} & -2a_{21} & a_{21} \\ -a_{11} & -a_{12} & -a_{11} \end{bmatrix}$$

- Obično se želi da je odziv detektora koji traži ivicu u jednom pravcu na ivicu koja je u drugom pravcu jednak 0. Da bi se ovo dobilo postavljaju se centralni pikseli ove matrice na 0:

$$\begin{bmatrix} a_{11} & a_{12} & a_{11} \\ 0 & 0 & 0 \\ -a_{11} & -a_{12} & -a_{11} \end{bmatrix}$$



Dizajn maske

- Konačno detekciona matrica se može podijeliti sa a_{11} jer se ovo može tretirati kao "nevažna" multiplikativna konstanta:

$$\begin{bmatrix} 1 & K & 1 \\ 0 & 0 & 0 \\ -1 & -K & -1 \end{bmatrix} \quad \begin{array}{l} \text{Za ivice u} \\ \text{x-pravcu} \end{array}$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -K & 0 & K \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{array}{l} \text{Za ivice u} \\ \text{y-pravcu} \end{array}$$

$K=1 \Rightarrow$ Prewittova matrica

$K=2 \Rightarrow$ Sobelova matrica

Najpoznatiji detektor
zasnovan na maski.

Detektor - primjer

Lena, ivice detektovane na osnovu Sobelovog detektora po oba pravca (čita se Zobel), te u pravcu horizontale odnosno vertikale.



ILI operacija
primjenjena na
odzive detektora
po horizontali i
vertikali.



MATLAB - Naredba

- Matlabova naredba za detekciju ivica je **edge**.
- Naredba je jednostavna za korišćenje u nekom osnovnom obliku: **E=edge(G,'tip','parametri');** gdje je G sivoskalirana slika čije se ivice traže, 'tip' je dati tip detektora ivica (do sada učeni su 'roberts', 'prewitt' i 'sobel') a 'parametri' su određeni parametri datog detektora. Izlazna slika **E** je binarna. Obično je treći parametar (koji se može izostaviti i prepustiti MATLAB-u za podešavanje) **prag** koji se zadaje kao broj od **0** do **1** (može se postaviti i da je **[]**), dok je četvrti parametar kod 'sobel' i 'prewitt', pravac ivice koji može biti: 'horizontal', 'vertical' ili 'both'.



Matrice većih dimenzija

- Za dimenzije susjedstva 5×5 Prewittove matrice se definišu na sledeći način:

$$\begin{bmatrix} -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$



Prag detekcije ivice

- Prag je prilično teško odrediti na neki pošten način. Naime, u tamnom dijelu slike odziv detektora ivica će biti mali a to ne znači da u tom dijelu nemamo ivica dok u svijetlim oblastima čak i mali prelaz (u procentima) sa jednog na drugi osvjetljaj nam daje veliki odziv detektora ivice.
- Još veći problem predstavlja činjenica da mi vjerovatno za datu tačku možemo imati liniju koja ne prolazi ni horizontalno ni vertikalno pa nam mogu oba detektora i po x i po y osi imati istovremeno mali ili veliki odziv.
- Konačno može da se dogodi da više uzastopnih piksela predstavlja postepeni prelaz sa jednog na drugi osvjetljaj.



Prag detekcije ivice

- Ozbiljne knjige rijetko ulaze u razmatranje dizajna praga ivice jer je teško uspostaviti konzistentan pristup. Radi potpunosti ovdje ćemo demonstrirati kako se to obično radi kod klase detektora kojoj pripadaju Sobelov i Prewittov.

- Odziv detektora se postavi kao:

$$e(n, m) = \frac{e_x^2(n, m) + e_y^2(n, m)}{2}$$

→ Odzivi detektora računati za pravac x-ose odnosno y-ose

- Globalni prag se obično računa kao:

$$T = \sqrt{\frac{4}{MN} \sum_m \sum_n e(n, m)}$$



Prag detekcije ivice **PLUS**

- Ako se želi detekcija ivica u pravcu x-ose onda se zahtjeva da je $|e_x(n,m)| > |e_y(n,m)|$ odnosno na sličan način ako se želi detektovati ivica u pravcu y-ose mora da važi:
 $|e_y(n,m)| > |e_x(n,m)|$.
- Konačno se od odziva detektora traži da bude i lokalni maksimum odnosno da bude veći od odziva u susjednim tačkama po odgovarajućim pravcima:
 $|e_x(n,m)| > |e_x(n-1,m)|$ i $|e_x(n,m)| > |e_x(n+1,m)|$ ako se detektuje ivica u pravcu x-ose odnosno
 $|e_y(n,m)| > |e_y(n,m-1)|$ i $|e_y(n,m)| > |e_y(n,m+1)|$ ako se detektuje ivica u pravcu y-ose.



Pravac ivice

- Očigledno određivanje praga detekcije za ivicu je složen posao i što je još gore od detektora do detektora se mijenja određivanje dobrog praga.
- Postoji i problem kako estimirati pravac ivice. Kod klase detektora zasnovanih na maski kao što su Sobelov i Prewittov imamo podatke samo o ivicama u pravcu x i y koordinate. Postoji mogućnost da se estimacija pravca ivice obavi na osnovu $e_x(n,m)$ i $e_y(n,m)$ kao

$$\hat{\phi}(x, y) = \arctan\left(\frac{e_y(x, y)}{e_x(x, y)}\right)$$



Kirchovi detektori

- Kirch je predložio da se uvedu dodatne maske sa kojim bi se vršila estimacija ivica u dodatnim pravcima. Na primjer detektor koji bi bio osjetljiv na ivice u pravcima x , y -osa te osa koje sa x -osom zaklapaju uglove $\pi/4$ i $3\pi/4$ bi imao sledeće maske:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}$$

Pravac ivice se procijenjuje na osnovu najvećeg odziva detektora.



Kirchovi detektori

- Kirch se baš dobro potrudio pa je dizajnirao i detektore koji mogu da otkriju osam pravaca ivice.

$$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

$$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

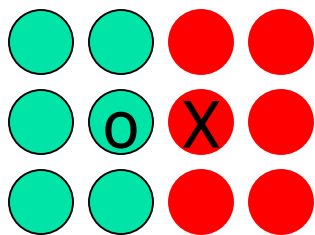


Laplasov detektor

- Logika Laplasovog detektora je bitno drugačija od detektora definisanih preko maske.
- Laplasov diferencijalni operator se definiše kao:

$$\nabla^2 f(n, m) \approx f(n, m) - \frac{1}{4} [f(n+1, m) + f(n-1, m) + f(n, m+1) + f(n, m-1)]$$

Ivica (nagla promjena osvjetljaja) se detektuje tamo gdje dobijamo nulu Laplasovog operatora (obično gdje imamo prolazak kroz nulu).



Ako je crveno 100 a zeleno 50 odziv Laplasovog detektora za piksel X je: $100 - 350/4 > 0$ dok je za piksel 0: $50 - 250/4 < 0$ što sugeriše da je prolazak kroz nulu (ivica) između ova dva piksela.



Laplasov detektor - mane

- Osnovni problem kod Laplasovog detektora je prevelika osjetljivost na šum. Naime, ako je slika uniformna i ako je zahvaćena makar malim šumom Laplasov detektor će detektovati mnoštvo “ivica” kao promjene znaka odziva detektora.
- Postoji mnoštvo (više ili manje prihvatljivih tehnika) kako ukloniti šum iz detektora. Jedna od tehnika je sračunati lokalnu varijansu piksela slike u nekom susjedstvu:

$$\sigma^2(n, m) = \frac{1}{(2M+1)^2} \sum_{k=n-M}^{n+M} \sum_{l=n-M}^{n+M} [f(k, l) - \bar{f}(n, m)]^2$$



lokalno susjedstvo



Laplasov detektor - mane

- U prethodnoj relaciji važi:

$$\bar{f}(n, m) = \frac{1}{(2M + 1)^2} \sum_{k=n-M}^{n+M} \sum_{l=m-M}^{m+M} f(k, l)$$

- Laplasov detektor uvodi prag varijanse. Ako je lokalna varijansa mala onda ivica nije detektovana (slika je približno uniformna) a ako je varijansa velika i postoji prolazak kroz nulu ivica je detektovana.
- Postoji i alternativa da se zona u kojoj može biti detektovana ivica prepozna ne na osnovu varijanse već na osnovu razlike:

$$w(n, m) = \frac{\max_A \{f(n, m)\} - \min_A \{f(n, m)\}}{1}$$

↓
**maksimalan i minimalan osvjetljaj u
lokalnom susjedstvu**

Ako je $w(n, m)$ veće od praga
ta tačka može biti dio ivice.



Laplas-Gausov detektor

- Jedan od načina za borbu protiv šuma je da se izlaz iz Laplasovog detektora filtrira Gausovim filtrom (Gausovim zamagljenjem) ili obrnuto da se prvo slika zamagli Gausovim filtrom pa da se onda primjeni Laplasijan (obje operacije su linearne i njihov redosljed se može zamijeniti bez razlike u rezultatu).
- Dobijeni detektor se u literaturi ponekad naziva **LoG** (Laplacian of Gaussian).
- Napominjemo da je matrica Gaussovog filtra data u obliku:

$$h(n, m) = \exp(-(n^2 + m^2) / 2\sigma^2)$$



Laplas-Gausov detektor

- Najčešće korišćena matrica za LoG detektor je:

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Naravno za različito σ slijede različite matrice za LoG ali se matrica data lijevo najčešće koristi u praksi.

Varijanta LoG detektora je Canny-jev detektor koji se smatra jednim od najboljih (mada ne baš najjednostavnijih).

Canny-jev operator je jedini koji je praktično uveden na osnovu odgovarajućeg optimizacionog problema. Mi ćemo ga ipak radi jednostavnosti uvesti poluintuitivno.



Canny detektor

- Kod ovog detektora se vrši filtriranje slike sa tri različita filtra. Gausovim filtrom sa impulsnim odzivom:

$$h(n, m) = \exp(-(n^2 + m^2) / 2\sigma^2) / (2\pi\sigma^2) \longrightarrow \text{konstanta koja se usvaja}$$

te sa filtrima čiji je impulsni odziv jednak parcijalnim izvodu impulsnog odziva Gausovog filtra:

$$h_x(n, m) = \partial h(n, m) / \partial n = -n \exp(-(n^2 + m^2) / 2\sigma^2) / (\pi\sigma^2)$$

$$h_y(n, m) = \partial h(n, m) / \partial m = -m \exp(-(n^2 + m^2) / 2\sigma^2) / (\pi\sigma^2)$$

Sračuna se odziv slike na Gausov filter:

$$g(n, m) = f(n, m) *_n *_m h(n, m)$$

Ova operacija se koristi da redukuje uticaj šuma.



Canny detektor

- Nakon ove operacije izlaz iz Gausovog filtra se filtrira sa filterima koji imaju impulsne odzive u obliku izvoda Gausove funkcije:

$$e_x(n, m) = g(n, m) *_{n,m} h_x(n, m)$$

$$e_y(n, m) = g(n, m) *_{n,m} h_y(n, m)$$

- Odziv detektora se računa kao:

$$e(n, m) = \sqrt{e_x^2(n, m) + e_y^2(n, m)}$$

Ova veličina se najčešće normalizuje na 1 djeljenjem sa maksimalnom vrijednošću $e(n, m)$.

Procedura koja se provodi u praksi i koja iz odziva $e(n, m)$ formira binarnu sliku sa ivicama je posebna priča.



Canny detektor - Pragovi

- Neki od koraka za dobijanje ivica kod Canny detektora na osnovu $e(n,m)$:
 - Odredite procenat slike u kojem mogu biti ivice. Tom procentu odgovaraju najveće vrijednosti odziva $e(n,m)$.
 - Eliminišu se sve tačke koje nijesu lokalni maksimumi po odgovarajućim pravcima s time što se sada posmatraju i dijagonalni pravci.
 - Kandidat za ivicu su one tačke koje su veće od odabranog praga (prag 1), imaju jedan od najvećih odziva detektora, predstavljaju lokalni maksimum.
 - Još se ponekad vrši određivanje piksela slike koji imaju odziv veći od praga 2 (koji je veći od praga 1) i konačni odziv određuje kao neka kombinacija dvije dobijene slike (umanjen broj piksela u odnosu na slučaj sa manjim pragom i uvećan u odnosu na slučaj sa većim pragom).
- HUHUUH!!!



Detekcija izolovane tačke

- Izolovana tačka se može detektovati sa detektorom koji ima oblik:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

- Alternativno može se koristiti matrica većih dimenzija:

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 2 & -1 & -1 \\ -1 & 2 & 4 & 2 & -1 \\ -1 & -1 & 2 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

Detekcija linije na kontrastnoj površi

- Ako ivica nije pozicija na kojoj se dvije bitno različite boje dodiruju već je recimo samo bijela linija preko crne pozadine dobar način da se izvrši detekcija je preko matrica oblika:

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$



Detekcija ivica u boji

- Najjednostavniji način da se izvrši detekcija ivica kod slike u boji je da se to obavi za svaki kanal pojedinačno pa da se dobijene slike spoje:

```
A=imread('flowers.tif');
```

```
er=edge(A(:,:,1));
```

```
eg=edge(A(:,:,2));
```

```
eb=edge(A(:,:,3));
```

```
e=er|eg|eb;
```

```
imshow(e)
```

Detektor po raznim kanalima računati po podrazumijevanim parametrima funkcije edge iz MATLAB-a.

Tri detektora spojena u jedan korišćenjem logičke ili operacije.

Ovo je veoma jednostavno i daje efektne rezultate ali u posljednje vrijeme ima dosta napora da se kreiraju složeniji detektori ivica slike u boji zbog nekih značajnih aplikacija. Zainteresovani studenti mogu ovu temu odabrati za projektni zadatak.



Ostatak priče o ivicama

- Gotovo svi poznatiji detektori ivica su pomenuti u okviru ove naše lekcije.
- U posljednje vrijeme se pokušava (i ima dosta uspjeha) u dizajnu detektora ivica koji mogu da imaju adaptivnu širinu maske koja se koristi kod detekcije. Teorijski se pokazuje da ovakav način vodi ka “gotovo” tačnim vrijednostima izvoda slike za razliku od diferenci koje se ovdje koriste.
- Vidjeli smo da je određivanje praga veoma izazovan i težak problem koji se najčešće prevazilazi heurističkim (ad-hok) tehnikama.



Houghova transformacija

- Često se slike transformišu u binarne koristeći detektore ivica pa se zatim na osnovu nekih jednostavnih geometrijskih objekata – **primitiva** vrši prepoznavanje oblika i objekata u slici.
- Jedan od tih primitiva je prava linija.
- Prava linija se može zapisati preko linearne zavisnosti $y = ax + b$ (ili $m = an + b$). Osnovni cilj kod linije je odrediti da li neki piksel pripada liniji i zatim odrediti parametre linije **(a,b)**.
- Procedure nije jednostavna. Posmatrajmo tačku (x_1, y_1) koja pripada nekoj ivici ali ne znamo da li pripada pravoj liniji niti kojoj ako pripada.



Houghova transformacija

Algoritam

- Usvaja se domen mogućih vrijednosti **a** i **b** kao Dekartov proizvod **A×B** gdje su $a_i \in A$ i $b_j \in B$ moguće vrijednosti iz ovih skupova.
- Možemo označiti $a_i = a_{\min} + i\Delta a$, $b_j = b_{\min} + j\Delta b$ gdje su a_{\min} i b_{\min} minimalne vrijednosti iz odgovarajućih skupova.
- Neka su a_{\max} i b_{\max} maksimalne vrijednosti ovih skupova i neka je kardinalnost (broj članova u) skupova N_A i N_B .
- Tada važi: $\Delta a = (a_{\max} - a_{\min}) / (N_A - 1)$ i $\Delta b = (b_{\max} - b_{\min}) / (N_B - 1)$.
- Kardinalnost skupova ne smije da bude ni prevelika ni premala. Prevelika usložnjava računanje a ponekad i otežava detekciju dok premala je neprecizna u selekciji parametara.



Algoritam za Hough transform.

- Formira se matrica $P(i,j)=P(a_i,b_j)$ gdje se svakom elementu iz domena pridružuje jedna vrijednost $P(i,j)$.
- Na početku se svako $P(i,j)$ $i=1,\dots, N_a$ i $j=1,\dots,N_b$ postavi na nulu.
- Za svaku tačku slike (x_k,y_k) za koju je vrijednost detektora ivica jednaka 1 (detektovana ivica) $e(x_k,y_k)=1$ odrede se sve prave sa parametrima iz domena kojima ova tačka može da pripada $\mathbf{b}_r=\mathbf{y}_k-\mathbf{a}_q\mathbf{x}_k$ i za sve te vrijednosti uveća za 1 odgovarajući elementi matrice P : $P(q,r)=P(q,r)+1$.



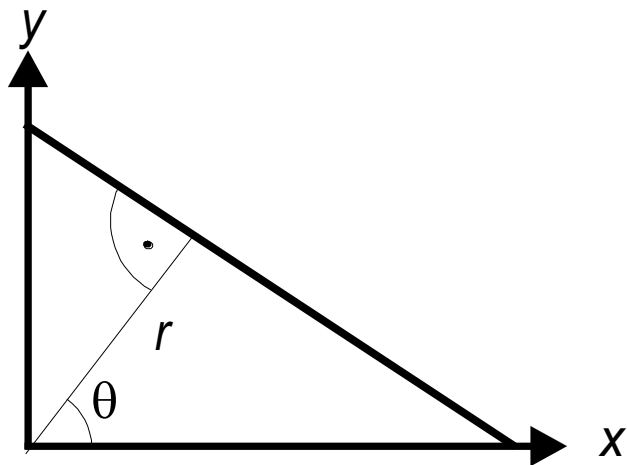
Algoritam Hough transformacije

- Kada se postupak ponovi za svaku tačku onda se detektuje maksimum matrice **P** i ta vrijednost odgovara pravoj sa odgovarajućim parametrima.
- Poništi se dio elemenata matrice **P** oko detektovanog maksimuma i postupak ponovi za traženje narednog.
- Postupak se nastavlja sve do traženog broja detektovanih linija ili dok više nema jasno izraženog maksimuma matrice **P**.
- Premda veoma jednostavna, Houghova transformacija je iznenađujuće kvalitetan detektor pravih linija.
- Osnovni problem Houghove transformacije su teškoće u detekciji linija približno paralelnih x osi oblika $x+b=0$ za koje je koeficijent pravca blizu beskonačno.

Problem kod Hougha

- Ovaj problem se prevazilazi tako što se prava linija parametrizuje kao:

$$r = y \cos \theta + x \sin \theta$$



$$R \geq r \geq 0$$

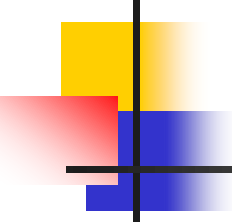
$$\pi/2 \geq \theta > -\pi/2$$

Granica domena koja se određuje na osnovu dimenzija slike.

Na ovaj način su oba parametra koja su iskorišćena za parametrizaciju prave ograničenog domena.

Houghova transformacija

Test primjer



```
clear
%Tacke pravougaonika
N=256;
B=zeros(N);
%Linije
for k=60:150,B(k,k)=1;end
for k=20:110,B(k,k+80)=1;end
for k=60:-1:20,B(k,120-k)=1;end
for k=150:-1:110,B(k,300-k)=1;end
M=256;
P=zeros(M);
tacnost=0.2;
R=linspace(-M,M,M);
th=linspace(-pi/2,pi/2,M);
```

Realizacija je čisto ilustrativna bez ugrađenih određenih elemenata koji je mogu učiniti efikasnijom ili robusnijom na uticaj šuma.

Nacrtan pravougaonik (četiri prave linije).

Matrica P, inicijalizacija i prostor parametara.

Dodatni parametar korišćen u algoritmu.



MATLAB Realizacija - Primjer

```
for y=1:N
```

```
for x=1:N
```

```
    if(B(x,y)==1)
```

```
        for r=1:M
```

```
            for p=1:M
```

```
                if(abs(R(r)-x*cos(th(p))-
```

```
                    y*sin(th(p)))<tacnost),  
                    P(r,p)=P(r,p)+1;end
```

```
            end
```

```
        end
```

```
    end
```

```
end
```

```
end
```

```
figure(1),imshow(B)
```

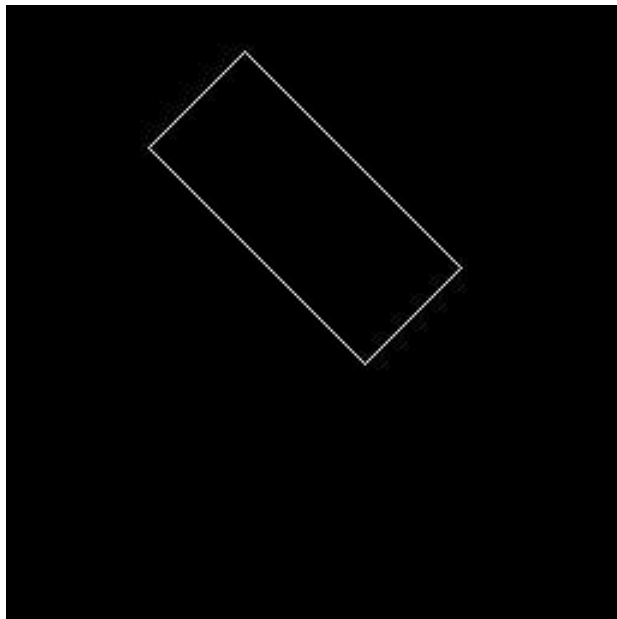
```
figure(2),pcolor(th,R,P),shading interp
```

Ako tačka pripada ivici...

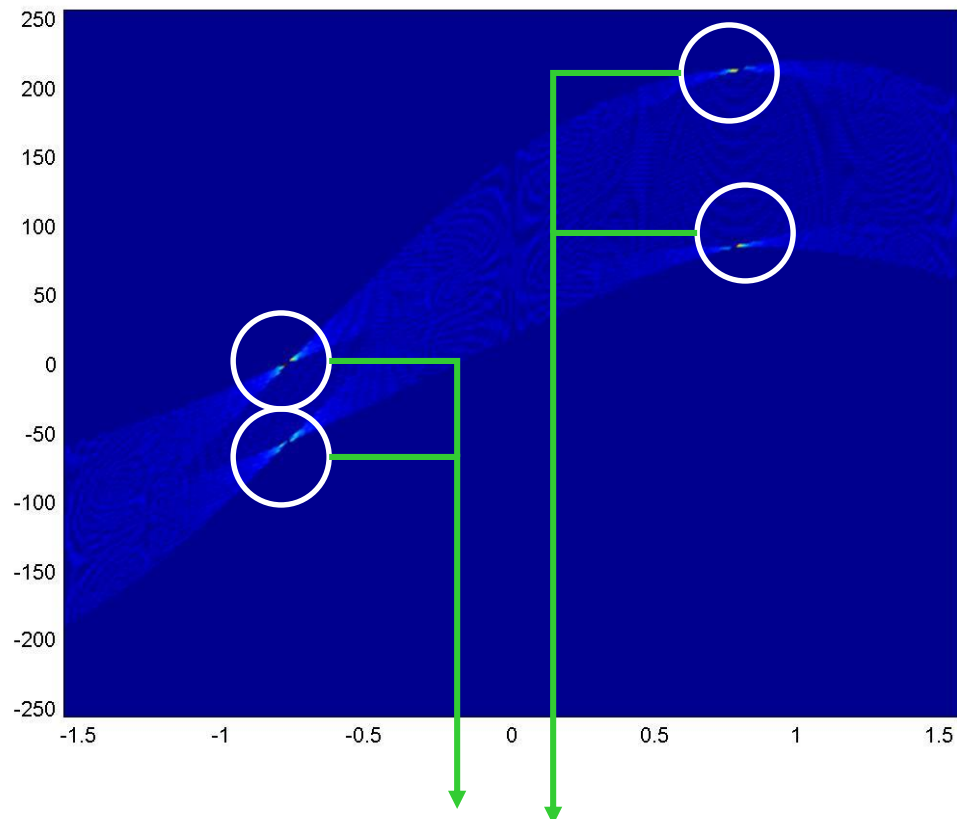
... uvećati za 1 one elemente
matrice P koji mogu predstavljati
prave koje prolaze kroz tu tačku.

Grafički prikaz.

Rezultati



“Pravougaonik”



**Houghova transformacija sa
detektovanim linijama**



Houghova transformacija - Zaključak

- Krajnje jednostavna zamisao, poluintuitivna, podložna raznim unapređenjima (poboljšanjima tačnosti, otpornosti na uticaj šuma, otkrivanja kratkih pravih linija) i danas je jedan od najpopularnijih algoritama za detekciju pravih linija.
- Uporedne analize pokazuje da po gotovo svim kriterijuma prevazilaze druge algoritme.
- Može se lako povezati sa Radonovom transformacijom.
- Koristi se i kod drugih oblika koji se mogu parametrizovati preko 2 parametra ali i kod onih sa više parametara koji se na neki prihvatljiv način mogu pojednostaviti.



Houghova transformacija - Krug

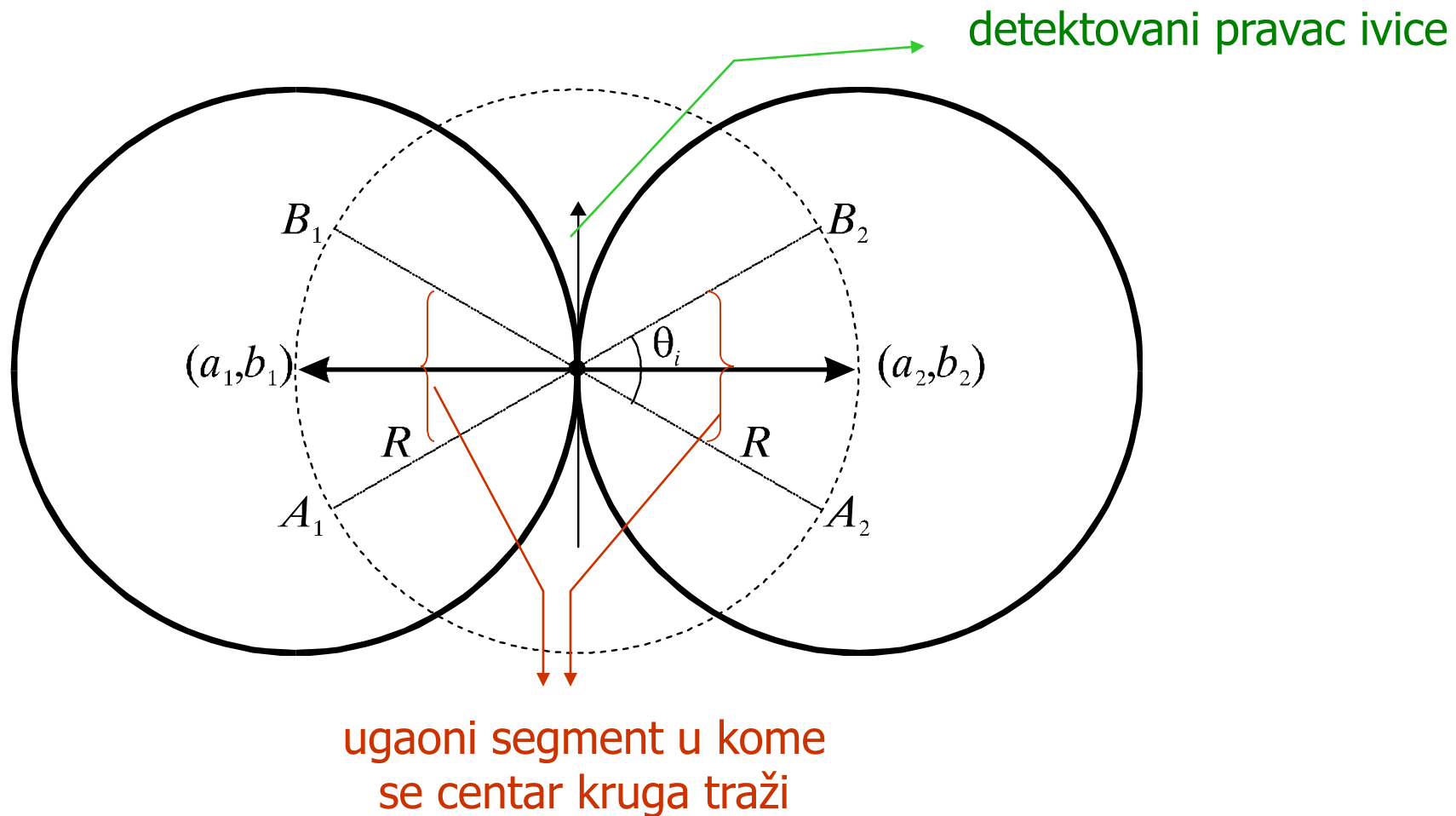
- Krug se može zadati preko koordinata centra i poluprečnika na sledeći način:
$$(x-a)^2+(y-b)^2=r^2$$
- Parametrizacija se u cilju provođenja Houghovog algoritma može obaviti preko tri parametra: (a,b,r) .
- Tačke koje pripadaju krugu se mogu zapisati i na sledeći način:
$$x=a+r\cos\theta \qquad y=b+r\sin\theta$$
- Strategije pojednostavljivanja se koriste da bi se 3D pretraživanje koje može biti računski neprihvatljivo svelo na prihvatljivu mjeru.



Parametri kruga

- Kod parametrizacije kružnica putem Houghove transformacije koristi se pravac ivice. Za svaku tačku koja pripada ivici $\mathbf{e}(\mathbf{x}, \mathbf{y})$ detektuje se i pravac ivice $\phi(\mathbf{x}, \mathbf{y})$.
- Sada se koristi činjenica da je pravac ivice, ako ivica pripada kružnoj liniji, normalan na radijus koji spaja centar i datu tačku. Ovo bi svelo pretraživanje za centar i poluprečnik na 2D pretraživanje odnosno na 1D jer tražimo tačku duž određenog pravca.
- Nažalost, zbog ograničene tačnosti kod detekcije pravca ivice ovo se ne može odraditi na predstavljeni način.
- Umjesto toga se centar kruga traži u određenom segmentu oko normale na detektovani pravac.

Parametri kruga - ilustracija





Za samostalni rad

- Realizujte sve predstavljene detektore ivica: Robertsov, Sobelov, Prewittov, Laplasov, Gaus-Laplasov, Canny-jev detektor.
- Realizujte opisane (u okviru prezentacije te u knjizi) metode za mapiranje odziva detektora u binarnu sliku.
- Proučite metodologiju koja je korišćena za determinisanje praga kod svih a posebno kod Canny-jevog operatora u MATLAB-ovoj funkciji edge.
- **Za miniprojekat:** Razmotrite detekciju ivica u boji pomoću Canny i Kumani operatora. U ovom slučaju se ne vrši detekcija nezavisno po pojedinim kanalima već se rezultati po kanalima kombinuju pri određivanju odziva detektora.
- **Za miniprojekat:** Primjena detektora ivica pri dizajnu adaptivnih filtara.



Za samostalan rad

- **Za miniprojekat:** Primjena detektora ivica u boji za kreiranje adaptivnih filtara u boji.
- **Za miniprojekat:** Realizacija detektora izolovane tačke i detektora kontrasnih linija sa načinom podešavanja praga.
- **Za miniprojekat:** Realizacija Kirchovih detektora i mogućnost da se na osnovu njih odredi pravac ivice preciznije nego kao multipl ugla 45° .
- **Za miniprojekat:** Savremene adaptivne tehnike za detekciju ivica na osnovu precizne estimacije izvoda.
- Izvršite ubrzavanje predstavljenog algoritma za računanje Houghove transformacije. **Pomoć:** Kada detektujete tačku koja pripada ivici ići duž detektovanog pravca a ne za svako (x,y) .



Za samostalan rad

- Povezati Houghovu transformaciju i Radonovu transformaciju.
- **Za miniprojekat:** SLIDE algoritam za detekciju ivica.
- **Za miniprojekat:** Poboljšanja Houghove transformacije.
- **Za miniprojekat:** Poređenje poboljšanja Houghove transformacije i SLIDE algoritma. **Napomena:** Literatura o SLIDE algoritmu se može preuzeti od predmetnog nastavnika.
- Realizujte algoritam za prepoznavanje krugova i kružnih lukova.
- **Za miniprojekat:** Algoritam za prepoznavanje elipsi.
- Kako se pomoću Houghove transformacije može prepoznati početak i kraj linije?
- **Za miniprojekat:** Realizacija algoritama za praćenje ivica. Pogledati skriptu.
- **Za miniprojekat:** RANSAC algoritam je takođe jedna relativno nova tehnika koja se koristi za detekciju linija, krugova i drugih primitivnih oblika. Realizujte je i poredite sa SLIDE-om i Houghovom transformacijom.