

Vježba 7.

Određivanje ivica objekata

Cilj ove vježbe je upoznavanje sa algoritmima za određivanje ivica objekata na slici zasnovanim na korištenju gradijentnih filtara. Postupak se, generalno, sastoji iz dva koraka. U prvom se izračunava aproksimacija derivacije slike korištenjem odabranog gradijentnog filtra, a u drugom se formira binarna slika ivica, najčešće poređenjem sa fiksnim pragom.

U IPT se korištenjem funkcije `fspecial` mogu generisati maske za Prewittov, Sobelov i LoG filter. U slučaju Prewittovog i Sobelovog filtra dobijene maske se mogu iskoristiti za izračunavanje gradijenta u smjeru vertikalne ose, tj. određivanje horizontalnih ivica. Ukoliko se želi izračunavanje gradijenta u smjeru horizontalne ose može se upotrebiti transponovana maska. Takođe, može se primijetiti da su ove maske rotirane za 180 stepeni u odnosu na one date na predavanjima tako da će se smjer vektora gradijenta izračunatog njihovim korištenjem razlikovati od smjera vektora gradijenta izračunatog korištenjem filtara datih na predavanjima za 180 stepeni. Međutim, ova razlika nema većeg praktičnog značaja.

U IPT postoji i funkcija `edge` koja integriše pomenuta dva koraka za određivanje ivica objekata na slici. Sintaksa ove funkcije je:

```
[ivice, prag] = edge(a, metod, prag, parametri);
```

gdje je `a` slika na kojoj se određuju ivice, `metod` je string koji označava izabrani metod određivanja ivica, `prag` je vrijednost praga koji se koristi pri formiranju binarne slike, a `parametri` su parametri algoritma i razlikuju se za pojedine metode. Vrijednost praga se ne mora zadati u kom slučaju je funkcija izračunava automatski. Izlazni argument `ivice` predstavlja binarnu sliku ivica, a `prag` je upotrebljena vrijednost praga. Automatski izračunata vrijednost praga nije uvijek optimalna i često je potrebno njeno ručno podešavanje. Zbog toga se obično prvo funkcija pozove bez zadate vrijednosti praga, a zatim se polazeći od te vrijednosti ručno zadaju modifikovane vrijednosti praga sve dok se ne dobije prihvatljivo rješenje.

Nakon određivanja ivica pikseli na rezultujućoj binarnoj slici bi trebalo da određuju ivice objekata. Međutim, zbog poremećaja kao što su šum i sjenčenje ovo nije uvijek slučaj. Zbog toga je sledeći korak u analizi slike primjena neke procedure za povezivanje ivica. Jedna od efikasnih metoda koja se može koristiti u ovu svrhu je Houghova transformacija. U svom standardnom obliku Houghova transformacija se koristi za detekciju linija na slici na osnovu piksela ivica. U IPT postoji funkcija `hough` čija je sintaksa:

```
[H, theta, rho] = hough(BW, param1, val1, param2, val2);
```

Ova funkcija izračunava standardnu Houghovu transformaciju binarne slike BW i vraća njene vrijednosti u matrici H . Preostala dva izlazna argumenta θ (u stepenima) i ρ su vrijednosti θ i ρ za koju se izračunava transformacija. Parametri Houghove transformacije mogu biti:

`ThetaResolution` – korak diskretizacije ugla θ ,

`RhoResolution` – korak diskretizacije ρ ose.

Ovi parametri su opcioni i redosljed njihovog zadavanja nije bitan. Predefinisana vrijednost oba parametra je 1.

Sledećim fragmentom koda je dat primjer kako se može izračunati i vizuelizovati Hougova transformacija binarne slike f .

```
[H, theta, rho] = hough(f);  
imshow(theta, rho, H, [], 'notruesize')  
axis on, axis normal  
xlabel('\theta'), ylabel('\rho')
```

Prvi korak u upotrebi Houghove transformacije za detekciju i povezivanje linija je pronalaženje lokalnih maksimuma (pikova) Houghove transformacije. Ovo nije jednostavan zadatak zbog kvantizacije vrijednosti piksela, kvantizacije parametarskog prostora, te činjenice da ivice na slici nisu uvijek savršeno ravne. U IPT je nalaženje lokalnih maksimuma implementirano funkcijom `houghpeaks` čija je osnovna sintaksa:

```
peaks = houghpeaks(H, numpeaks);
```

Ulazni argumenti ove funkcije su matrica Houghove transformacije H i željeni broj maksimuma `numpeaks`. Izlazni argument `peaks` je matrica sa dvije kolone u kojima se nalaze brojevi redova i kolona u matrici Houghove transformacije u kojima se nalaze maksimumi. Parametri ove funkcije su opisani u dokumentaciji.

Sledeći fragment koda ilustruje korištenje funkcije `houghpeaks` za pronalaženje maksimuma i njihovo označavanje na dijagramu Houghove transformacije.

```
P = houghpeaks(H, 5, 'threshold', ceil(0.3*max(H(:))));  
x = T(P(:,2)); y = R(P(:,1));  
% Houghova transformacija je već prikazana i uključen je hold  
plot(x, y, 's', 'color', 'white');
```

Konačno, kada se pronađu kandidati za maksimume preostaje da se odredi da li postoje linijski segmenti koji im odgovaraju i gdje oni počinju i završavaju se. Ovo je u IPT implementirano funkcijom `houghlines` čija je sintaksa:

```
lines = houghlines(BW, T, R, P, param1, val1, param2, val2);
```

Ulazni argumenti su binarna slika BW , vektori vrijednosti θ i ρ koje je vratila funkcija `hough`, T i R , respektivno. P je matrica koju je vratila funkcija `houghpeaks`. `lines` je struktura u kojoj se nalaze početne i krajnje tačke linijskih segmenata, te njihov nagib i udaljenost od ishodišta. Parametri su:

FillGap – udaljenost između linijskih segmenata koji se spajaju u jedan segment,

MinLength – minimalna dužina linijskog segmenta koji se zadržava.

Detekcija linijskih segmenata se može izvesti na sledeći način:

```
lines = houghlines(BW,T,R,P,'FillGap',30,'MinLength',10);
figure, imshow(I), hold on
for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

    % označavanje početaka i krajeva linija
    plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
    plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');
end
```

Zadaci

1. Učitati test-sliku `test.tif` (data na web stranici predmeta) i prikazati je. Korištenjem `im2double` promijeniti klasu kojoj pripadaju vrijednosti piksela.
2. Korištenjem Robertsovog gradijentnog filtra (dat na predavanjima) odrediti derivaciju test-slike u smjeru x i y -ose. Prikazati rezultate. Vodite računa o činjenici da rezultat izračunavanja gradijentnog operatora može izići iz opsega vrijednosti intenziteta.
3. Odrediti amplitudu gradijenta izračunatog Robertsovim filtrom. Prikazati rezultat. Odrediti i smjer gradijenta. Prikazati rezultat. Komentarisati. Na osnovu dobijenih rezultata formulisati postupak za određivanje ivica na slici. Formirati binarnu sliku na kojoj su ivice predstavljene vrijednošću 1, a ostatak slike vrijednošću 0.
4. Narušiti test-sliku Gausovim aditivnim šumom nulte srednje vrijednosti i varijanse 0.001. Odrediti i prikazati amplitude gradijenata izračunatih Robertsovim nesimetričnim i simetričnim filtrima, te Prewittovim i Sobelovim filtrom. Komentarisati. Za svaki od ovih slučajeva pokušajte formirati najbolju binarnu sliku ivica
5. Na test-sliku iz tačke 1. primijeniti LoG filtar. Prikazati i komentarisati rezultat. Na koji način standardna devijacija Gausovog filtra utiče na rezultat? Kako se ovaj rezultat može iskoristiti za određivanje položaja ivica?
6. Proučiti parametre funkcije `edge` za određivanje ivica korištenjem LoG filtra. Korištenjem ove funkcije pokušajte formirati najbolju binarnu sliku ivica. Rezultat uporedite sa rezultatima iz tačke 4. Komentarisati.
7. Učitati sliku `kuca.tif`. Odrediti ivice na ovoj slici korištenjem Robertsovog, Sobelovog i LoG detektora ivica implementiranih u funkciji `edge` sa predefinisanim vrijednostima parametara.

8. Interaktivno podesiti vrijednosti parametara funkcije `edge` kako bi se dobila što bolja segmentacija. Uporediti kvalitet dobijenih rezultata.
9. **Houghova transformacija.** Formirati jednostavnu binarnu test-sliku koja će poslužiti za ilustraciju Houghove transformacije.

```
f = zeros(101);
```

```
f(1, 1) = 1; f(101, 1) = 1; f(1, 101) = 1;
```

```
f(101, 101) = 1; f(51, 51) = 1;
```

Izračunati Houghovu transformaciju ove slike i prikazati rezultat. Komentarisati korištenjem sledećih naredbi:

10. Izračunati Houghovu transformaciju za najbolju sliku sa ivicama dobijenu u tački 7. Prikazati rezultat. Odrediti 5 najvećih vrijednosti Houghove transformacije i prikazati linije koje im odgovaraju.