

VEŽBA 7 – 2D konvolucija i detekcija ivica u slici

Potrebno predznanje

- Poznavanje programskog jezika C
- 2D signali
- RGB i YUV prostori boja i konverzija između ova dva prostora
- Diskretna konvolucija
- Filtriranje 1D signala u vremenskom domenu

Šta će biti naučeno tokom izrade vežbe

Tokom izrade ove vežbe biće objašnjena implementacija 2D diskretne konvolucije kao jedne od najčešće korišćenih operacija u obradi 2D signala. Kroz primere biće prikazana primena 2D diskretne konvolucije. Upoznaćete se sa osnovama filtriranja 2D signala. Na primeru prikazan je uticaj uklanjanja niskih i visokih frekvencija iz signala slike. Upoznaćete se sa problemom detekcija ivica u slici i jednim načinom njegovog rešavanja upotrebom Sobel operatora.

Motivacija

Kod linearnih vremenski invarijantnih sistema vrednost izlaznih odbiraka signala računa se operacijom diskretne konvolucije ulaznog signala i impulsnog odziva sistema. Za 2D signale definisana je dvodimenzionalna diskretna konvolucija. Ova operacija predstavlja osnov za dalje izučavanje algoritama i sistema za obradu 2D signala. Jedan primer takvog sistema jeste sistema za detekciju ivica unutar slike. Svrha detekcije naglih promena u osvetljaju kao ivica unutar slike jeste prepoznavanje osobina onoga što je na slici prikazano. U idealnom slučaju primena algoritama za detekciju ivica dovodi do prepoznavanja objekata i njihovih svojstava u smislu oblika, dimenzija, orijentacije itd. Detekcija ivica predstavlja jedan od fundamentalnih koraka u obradi slike koja se odnosi na analizu slike, prepoznavanje šablona unutar slike, prepoznavanje pokreta i tehnikama računarskog vida.

Teorijske osnove

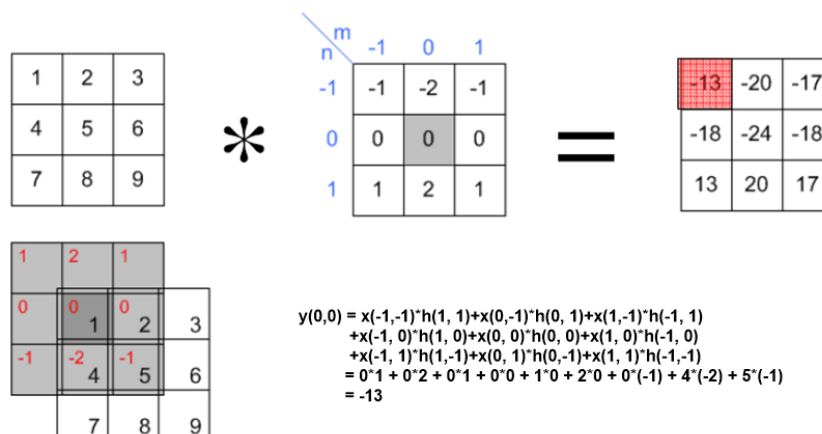
Diskretna 2D konvolucija

Kod 1D signala linearni vremenski invarijantni sistemi bili su predstavljeni impulsnim odzivom sistema $h(t)$. Obrada signala u okviru ovih sistema odgovara operaciji diskretne konvolucije između signala i impulsnog odziva sistema. Analogno tome u domenu 2D signala ovakvi sistemi predstavljeni su dvodimenzionalnim impulsnim odzivom $h(n,k)$ veličine $N \times K$. Obrada signala u okviru ovakvih sistema u prostornom domenu svodi se na operaciju dvodimenzionalne diskretne konvolucije. 2D konvolucija data je jednačinom:

$$y(v,h) = \sum_{n=1}^N \sum_{k=1}^K h(n,k) \cdot x(v-n, h-k)$$

2D diskretna konvolucija je linearna kombinacija $N \times K$ okolnih tačaka pri čemu su težinski koeficijenti definisani sa $N \times K$ vrednosti impulsnog odziva $h(n,k)$.

Prikaz računanja 2D diskretne konvolucije dat je na slici 1.



Matrica koja predstavlja impulsni odziv sistema se napiše unazad po X i Y osi, i pozicionira tako da se sredina matrice nalazi na koordinatama piksela čija se vrednost računa. Potom se vrši množenje svakog elementa matrice sa elementom slike koji se nalazi ispod njega i vrednosti se akumuliraju.

Filtriranje slike

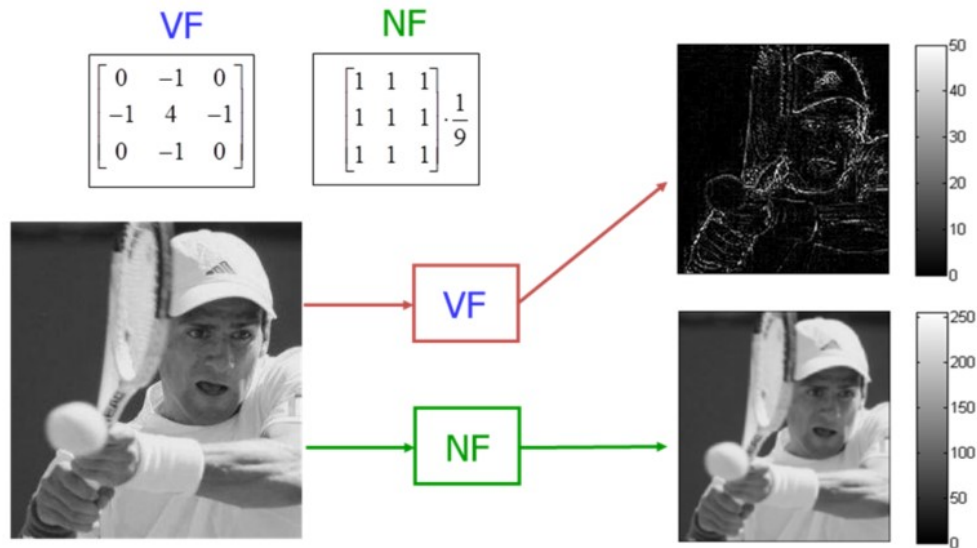
Filtriranje slike svodi se na operaciju diskretne konvolucije. Kernel filtra ujedno predstavlja i impulsni odziv. Veličina kernela mora biti neparna, odnosno kernel mora imati središnji element. Ukoliko je suma svih koeficijenata filtra jednaka 1, nakon filtriranja slika će sačuvati osvetljaj. Ukoliko je suma manja od 1, rezultujuća slika je tamnija od ulazne, a ukoliko je veća od 1 rezultujuća slika je svetlija.

S obzirom da slika ima konačne dimenzije, potrebno je pre filtriranja proširiti ivice slike za polovinu veličine kernela filtra, kako ne bi došlo do čitanja van opsega kada se vrši filtriranje ivičnih tačaka. Dodati piksela se popunjavaju nulama ili se vrši kopiranje ivičnih vrednosti originalne slike. Drugi način rešenja

ovog problema jeste da se u samoj implementaciji operacije konvolucije vodi računa o pristupanju vrednostima tačaka van slike. Kao i kod kopiranja vrednosti koje su logički van slike mogu da se zanemare (podrazumeva se da im je vrednost 0) ili da se vrši računanje sa ivičnim vrednostima.

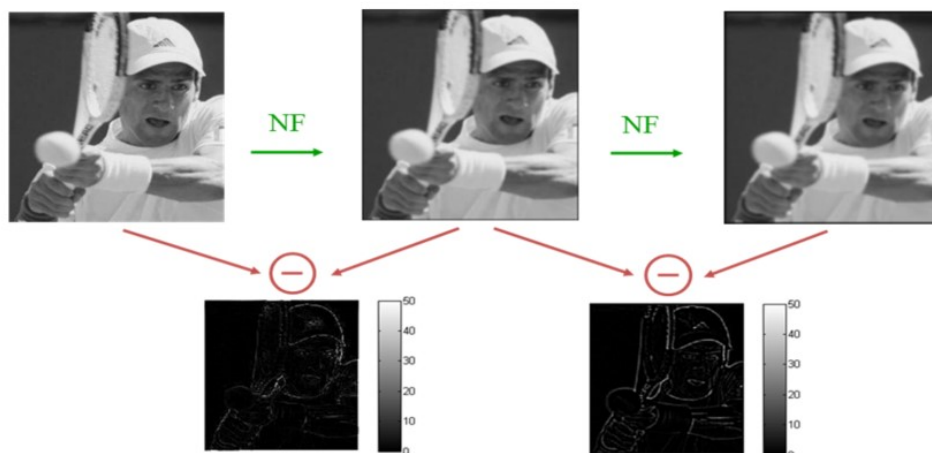
Prilikom filtriranja potrebno je voditi računa prekoračenju opsega. Ukoliko rezultat konvolucije prelazi maksimalnu vrednost (npr. 255 za Y kod YUV formata) potrebno ga je ograničiti na tu vrednost, isti slučaj je i sa minimalnom vrednošću (u slučaju vrednosti manje od 0 ograničiti na 0).

Na slici ispod dat je primer kernela filtera propusnika niskih frekvencija (NF) i visokih frekvencija (VF) i rezultat filtriranja slike ovim filterima.



Slika 1 - Primer filtriranja slike

Moguće je izvršiti i sukcesivno filtriranje koristeći isti filter. Dat je primer sukcesivnog filtriranja NF filtrom sa prethodne slike. Nakon svake iteracije dobija se zamućenija slika, odnosno više komponenti visokih frekvencija je potisnuto iz slike.



Slika 2 - Sukcesivno filtriranje NF filterom

Detekcija ivica u slici

Jedna od primena filtriranja slike jeste detekcija ivica u slici. Detekcija ivica je jedna od važnih metoda obrade slike, jer je to neophodni korak u segmentaciji objekata u slici. Pod detekcijom ivica se podrazumeva generisanje slike na osnovu ulazne, takve da su vidljive samo ivice objekata (vrednost 1 označava tačku na ivici a vrednost 0 tačku koja nije na ivici).

Osnovna ideja je korišćenje gradijenta (prvog izvoda) za meru da li neka tačka pripada ivici ili ne. U tome konceptu prvi izvod po x-osi predstavlja meru za vertikalnu ivicu a izvod po y-osi predstavlja meru za horizontalnu ivicu.

Najpoznatiji operatori za detekciju ivica su poznati kao Sobel operatori:

$$h_v(v,h) = \frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad h_h(v,h) = \frac{1}{4} \begin{bmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

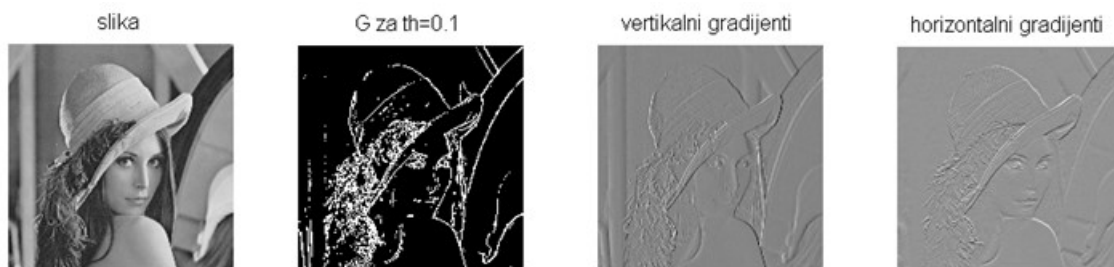
Detekcija ivica upotrebom Sobel operatora vrši se tako što se ulazna slika filtrira horizontalnim i vertikalnim operatorom odvojeno. Rezultat filtriranjem horizontalnim Sobel-operatorom jeste horizontalni gradijent, a vertikalnim vertikalni gradijent. Zato se za detekciju ivica u bilo kom pravcu koristi Euklidska mera gradijenta kao globalni gradijent, koji se izračunava na osnovu horizontalnog i vertikalnog gradijenta upotrebom sledeće jednačine:

$$G(v,h) = \sqrt{G_v^2(v,h) + G_h^2(v,h)}.$$

Ako vrednosti normalizovane ulazne slike leže u opsegu vrednosti od 0 do 1, vrednosti vertikalnih i horizontalnih gradijenata leže u opsegu od -1 do 1. Vrednosti globalnog gradijenta su u opsegu od 0 do $\sqrt{2}$. Na kraju, za svaku tačku ispituje se da li binarni globalni gradijent prelazi određenu graničnu vrednost. Ukoliko $G(v,h)$ prelazi granicu smatra se da tačka sa koordinatama (v,h) pripada ivici:

$$G(v,h) = \begin{cases} 0 & G(v,h) < th \\ 1 & G(v,h) \geq th \end{cases},$$

Menjanjem praga odlučivanja povećava se ili smanjuje osetljivost na ivice. Na slici ispod je prikazan primer detekcije ivica u jednoj slici.



Slika 3 - Detekcija ivica upotrebom Sobel operatora

Zadaci

U okviru ove vežbe vršiće se obrada nad Y komponentom slike. Na samom početku funkcije obrade izvršena je konverzija iz RGB prostora u YUV, a nakon obrade iz YUV u RGB upotrebom funkcija realizovanih u prošloj vežbi.

Zadatak 1

Implementirati Funkciju:

• `void convolve2D (uchar Y_buff[], int xSize, int ySize, double filterCoeff, int N)` koja vrši operaciju diskretne konvolucije nad ulaznom slikom koristeći prosleđeni filter. Parametri funkcije su:

- `Y_buff` – ulazna slika
- `xSize` – horizontalna dimenzija slike u pikselima
- `ySize` – vertikalna dimenzija slike u pikselima
- `filterCoeff` – koeficijenti filtra
- `N` – veličina filtra (koeficijenti filtra su veličine $N \times N$)

Pre računanja konvolucije potrebno je voditi računa o pristupanju vrednostima koje su logički van slike. Prilikom računanja vrednosti ivičnih piksela čije su horizontalne koordinate u opsegu $[0, N/2)$ i $[xSize - N/2 - 1, xSize)$, ili vertikalne u opsegu $[0, N/2)$ ili $[ySize - N/2 - 1, ySize)$ za piksele čije su koordinate manje od nula i/ili veće od veličine slike podrazumevati vrednost 0.

Zadatak 2

Implementirati funkciju:

- `void performNFFilter (uchar input[], int xSize, int ySize)`

koja vrši filtriranje slike upotrebom datog kernela NF filtra na slici 1 koristeći diskretnu konvoluciju realizovanu u prvom zadatku.

Zadatak 3

Implementirati funkciju:

- `void performVFFilter (uchar input[], int xSize, int ySize)`

koja vrši filtriranje slike upotrebom datog kernela VF filtra na slici 1 koristeći diskretnu konvoluciju realizovanu u prvom zadatku.

Zadatak 4

Implementirati funkciju:

- `void performSuccessiveNFFilter (uchar input[], int xSize, int ySize, int stages)`

koja vrši sukcesivno filtriranje slike upotrebom datog kernela NF filtra na slici 1 koristeći diskretnu konvoluciju realizovanu u prvom zadatku. Broj filtriranja zadat je parametrom *stages*. Prilikom poziva funkcije kao parametar *stages* proslediti vrednost grafičke kontrole *params[0]*.

Zadatak 5

Implementirati funkciju:

- `void performSobelEdgeDetection(uchar input[], int xSize, int ySize, uchar threshold)`

koja vrši detekciju ivica upotrebom Sobel operatora. Granična vrednost data je parametrom *threshold*. Prilikom poziva funkcije kao parametar *threshold* proslediti vrednost grafičke kontrole *params[0]*. U okviru funkcije potrebno je implementirati sledeće korake:

1. Napraviti kopiju ulazne slike
2. Izvršiti filtriranje upotrebom horizontalnog Sobel operatora nad jednom slikom. Rezultat je horizontalni gradijent.
3. Izvršiti filtriranje upotrebom vertikalnog Sobel operatora nad drugom slikom. Rezultat je vertikalni gradijent.
4. Izračunati ukupni gradijent. Proveriti da li je vrednost veća od granice. Ukoliko jeste, postaviti vrednost piksela na 255 u suprotnom postaviti na 0.

Zadatak 6

Implementirati funkciju:

- `void performNFplusSobelEdgeDetection(uchar input[], int xSize, int ySize, int stages, uchar threshold)`

koja prvo vrši sukcesivno filtriranje upotrebom NF filtra kao u drugom zadatku, a potom detekciju ivica kao u zadatku 5. Komentarisati uticaj filtriranja na NF filtrom na rezultat detekcije ivica.