

Univerzitet u Nišu
Elektronski fakultet

Obrada slike korišćenjem Guided filtra

Autor: Predrag Nikolić **Mentor:** Dejan Rančić

8. oktobar 2017

Sadržaj

1	Obrada slike	1
1.1	Uvod	1
1.2	Osnovni procesi u obradi digitalne slike	2
1.3	Pribavljanje slike	3
1.4	Bitne relacije između piksela	4
1.5	Osnovne operacije nad slikom	5
2	Osnovni tipovi filtara	9
2.1	Osnovni pojmovi	9
2.2	Smoothing filtri(filtri za uklanjanje šuma)	10
2.3	Sharpening filtri(filtri za određivanje ivica)	13
2.4	Kombinovanje filtara	18
3	Guided filter	18
3.1	Uvod	18
3.2	Slične ideje	19
3.3	Definicija filtra	20
3.4	Očuvanje ivica	22
3.5	Kernel	22
3.6	Osobine filtra	22
3.7	Izračunavanje i efikasnost	22
3.8	Poboljšanje	22
3.9	Eksperimenti	22
4	Primena	22
	Literatura	23

1 Obrada slike

1.1 Uvod

Računarska grafika je oblast računarstva koja se bavi obradom, stvaranjem i modeliranjem slika i video materijala, korišćenjem kompjutera. Ovaj izraz se obično odnosi na kompjuterski generisane podatke u vidu slike, koji su kreirani uz pomoć specijalizovanog softvera ili hardvera. Ova oblast se odnosi na sve iz sveta kompjutera što nije prikazano tekstom ili zvukom. Naučnici su počeli da se bave problemima iz računarske grafike još 60-ih, a danas ona obuhvata mnoge podoblasti i prožima mnoge druge oblasti kako računarskih tako i drugih nauka. Neke od tema kojima se bavi računarska grafika su dizajn korisničkog interfejsa, spote grafika, vektorska grafika, 3D modeling itd. Možemo da zaključimo da se proučavanje računarske grafike, kao deo računarskih nauka, bavi metodima za digitalnu obradu i manipulaciju vizuelnim sadržajem. Iako se računarska grafika obično odnosi na tro-dimenzionalnu kompjutersku grafiku takođe obuhvata i dvo-dimenzionalnu grafiku i obradu slike.

Obrada slike je proces kod kojeg se slika ili video (koji je ustvari niz slika) obrađuju da bi se dobila izmenjena slika (video) ili skup karakteristika ili nekih bitnih parametara. Slike se u obradi tretiraju kao signali pa se za njihovu obradu koriste metode iz oblasti obrade signala i razne matematičke operacije. Ova oblast je takođe usko povezana sa kompjuterskom vizijom i naravno kompjuterskom grafikom. Kada se kaže obrada slike obično se misli na digitalne slike, ali je takođe moguća i obrada optičkih i analognih slika.

Digitalna slika može da se definiše kao funkcija $f(x, y)$, gde su x i y koordinate ravni. x i y predstavljaju koordinate piksela na slici. Pikseli su elementi od kojih je sastavljena slika. Obično x predstavlja vrednost u odnosu na širinu slike, a y u odnosu na visinu slike. Rezolucija slike se obično obeležava kao širina puta visina, izraženo u pikselima. Vrednost funkcije f je intenzitet boje piksela na poziciji (x, y) . Intenzitet može da bude izražen kao jedna vrednost, ako je u pitanju crno bela slika, ili preko tri vrednosti ako je slika u boji. Postoje različiti načini za predstavljanje slike u boji (Npr. RGB, CMYK, HSV itd.). Vrednosti funkcije f kao i x i y su diskretne, jer je reč o digitalnoj slici.

Čulo vida je nasloženije čulo čoveka i ima možda i najveću ulogu u čovekovoj precepciji okoline. Međutim ono nije savršeno. Za razliku od ljudskog vida koji je limitiran vidljivim elektromagnetnim spektrom, u računarstvu imamo mogućnost da predstavimo ceo spektar i da se bavimo talasnim dužinama koje naše oko ne može da vidi. Obrada slike je složena oblast i prožima mnoge druge oblasti. Ne postoje jasne granice između obrade slike i oblasti kompjuterskog vida. Sama obrada slike može da bude deo sistema veštačke inteligencije za prepoznavanje lica ili nešto slično. Za obradu slike možemo da je porces koji na ulazu uzima sliku, vrši obradu i na izlazu se nalazi ponovo slika, ali u toku obrade takođe računa atribute te slike.

Potreba za korišćenjem digitalnih slika javila se još 1920-tih godina u novinarstvu. Tada je prvi put slika poslana prekookenaski putem analognog kabla. Naravno prvi problemi koji su se javili tada imali su veze sa kvalitetom slike. Prvi sistemi su mogli da prenose crno bele slike, do 15 nijansi sive. Prvi kompjuteri koji su bili dovoljno moći da izvršavaju neku značajniju obradu slika javili su se 60-ih godina. Neke od prvih primena bile su vezane za poboljšanje slika iz svemira. Zatim je obrada slika počela da se koristi u medicini i astronomiji 70-ih godina, a i dan danas je bitna u tim oblastima.

U današnje vreme skoro da ne postoji oblast u kojoj se u nekoj meri ne koristi obrada slika. Jedna od najpoznatijih primena obrade slike je kreiranje i obrada rendgenskih slika



Slika 1: Digitalna slika koja je kreirana na osnovu kodirane trake 1921 godine

(radiografija). Ona se najviše koristi u medicini ali postoje i druge primene. Primena postoji i za slike koje predstavljaju ultravioletnu boju. Ove slike se koriste u mikroskopiji. Postoje mnoge primene u svetu tehnologije kao što su izoštravanje slika, restauracija slika, prepoznavanje lica i objekata, prepoznavanje oblika. Takođe obrada slike se koristi u filmovima i medijima.

Bitne komponente sistema za obradu slike su specijalizovan hardver za obradu slike, kao što su grafičke kartice, zatim kompjuter koji izvršava procese obrade slike i prikazuje rezultat. Takođe je bitan softver koji se koristi za obradu slike kao i komponente i metode za čuvanje velike količine podataka. Naravno možda i najbitniji komponenta je displej na kojem možemo da vidimo rezultat obrade slika.

1.2 Osnovni procesi u obradi digitalne slike

Neki od osnovnih procesa i primena u obradi digitalne slike su sledeći. *Pribavljanje slike* (eng. *Image acquisition*) se odnosi na to kako je slika nastala. Slika može da bude data u digitalnom obliku, a može da se zada i u analognom. Akvizicija slike uključuje predprocesiranje kao što je skaliranje i drugo. *Poboljšanje slike* (eng. *Image enhancement*) je najprostiji ali i najkorišćeniji postupak u obradi slika. Cilj ovog procesa je da se na slici izraze bitni aspekti radi izučavanja parametra neke slike. Primene su različite (npr. izoštravanje ivica radi uočavanja detalja). *Restauracija slike* (eng. *Image restoration*) je oblast koja se bavi poboljšanjem izgleda slike. Razlika između restauracije i poboljšanja je u tome što se tehnike za restauraciju oslanjaju na matematičke i probabilističke modele degradacije slike, dok se poboljšanje zasniva na subjektivnom osećaju čoveka. *Obrada slika u boji* (eng. *Color image processing*) je oblast koja se bavi obradom slike u boji i artifaktima koje slike u boji donose u odnosu na obradu crno belih slika. *Kompresija* (eng. *Compression*) se bavi tehnikama za kompresiju slike, radi smanjenja veličine slika, što je od velikog značaja za prenos i čuvanje podataka. *Morfološka obrada slike* (eng. *Morphological processing*) se bavi alatima za ekstrakciju komponenti slika koji su pogodni za određenu reprezentaciju podataka koje slika nosi. *Segmentacija* (eng. *Segmentation*) se bavi podelom slike na kohezivne delove ili objekte. *Reprezentacija i deskripcija* (eng. *Representation and description*) kao što samo ime kaže se bavi predstavljanjem slike kako na ulazu tako i na izlazu nekog procesa obrade slike. I na kraju *Prepoznavanje* (eng. *Recognition*) je proces u kojem se objektima dodeljuju labelle odnosno imena na osnovu deskriptora, koji se mogu dobiti na osnovu slike. Svi ovi procesi mogu da koriste bazu znanja u kojoj se čuvaju bitne činjenice za svaki od procesa prilikom obrade slika.

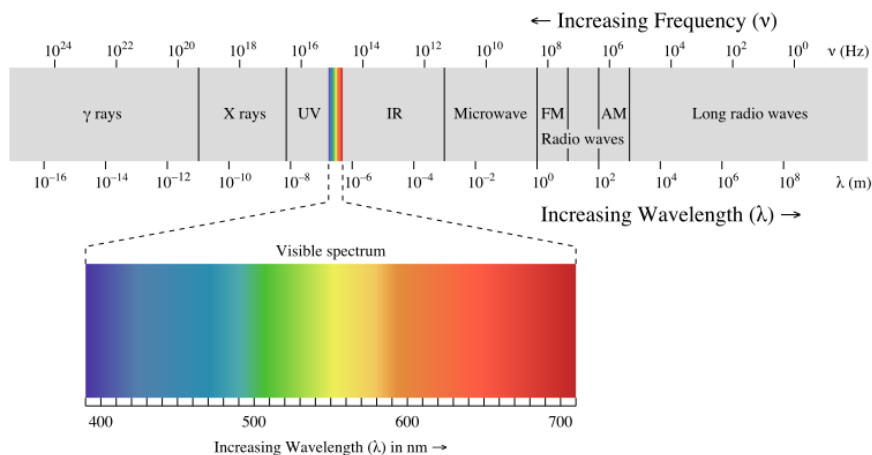
Postupci i osobine koje se javljaju na određenoj slici mogu da se jave ponovo. pa je čuvanje podataka o obradi veoma bitno.

U ovom radu ćemo se fokusirati na pribavljanje slike i poboljšanje slike.

1.3 Pribavljanje slike

Iako je oblast digitalne obrade slika izgrađena na onovu matematičkih i probablističkih formaulacija, pribavljanje slike je proces koji je određen na osnovu ljudskog čula vida. Slika se u digitalnom svetu pribavlja naslićan nači na koji čovek vidi, odnosno kreira sliku u mozgu. Ljudski preceptor za vid je oko. Prozirni prednji delovi oka lome zrake svetlosti projektujući umanjenu i obrnutu sliku na fotosenzitivnu mrežnjaču gde se u specijalizovanim nervnim ćelijama obavlja pretvaranje slike u električne nervne impulse. Zrak se prelama u očnom sočivu, a deo oka odgovoran za prikupljanje slike je žuta mrlja, gde su nervne ćelije najgušće raspoređene. Pored žute mrlje se nalazi početak vidnog živca koji je neosetljiv na svetlo, pa se njegova projekcija u vidom polju naziva crna mrlja. Nervni impulsi kreirani u ćelijama se prenosu u mozak koji spaja slike iz oba oka i okreće obrnutu sliku čime se stvara osećaj vida u ljudskom mozgu.

Svetslost koju mi opažamo ima određeni intenzitet i boju. Ljudsko oko je sposobno da vidi samo deo elektromagnetnog spektra. Ono može da detektuje svetlost koja se u sprektru nalazi izmedju ultraviolettne i infracrvene boje. Posto se svetlost tretira kao talas, boju odnosno deo spektra određuje talasna dužina. Talasna dužina se predstavalja kao odnos brzine svetlosti i frekvencije svetla $\lambda = c/\nu$. Energija koju nosi talas se računa se kao $E = h\nu$, gde je h Plankova konstanta, a ν je frekvencija talasa. Ljudsko oko vidi svetlost talasne dužine između $0.4 \cdot 10^{-6}m$ i $0.7 \cdot 10^{-6}m$.



Slika 2: Elektromagnetni spektar

Slika se u digitalnom svetu generiše kao kombinacija osvetljenja prostora i reflektivne ili apsorbovane energije iz prostora scene za koju se pravi slika. Postoje raličiti tipovi senzora za akviziciju slika, ovde će biti opisan sistem sa senzorom u obliku 2-D niza (matrice). Senzor je sastavljen od elementata koji su raspoređeni u matrici. Svaki element je zadužen za prikupljanje svetlosti iz scene. U svakom elementu se beleži osveteljenje scene i tako se formira signal slike koji se kasnije dodatno obrađuje.

Pošto svetlost dolazi i kontinulanog prostora, vrednost signala koji nosi sliku će biti kontinualna. Da bi takav signal mogao da se obradi u računaru potrebno je da se taj signal

digitalizuje. Digitalizaciju signala omogućuju smplovanje i kvantizacija. Smplovanje se odnosi na digitalizaciju koordinata piksela, odnosno određivanja mesta u prostoru. Kvantizacija se odnosi na digitalizaciju vrednosti amplitude signala. Pošto je slika skup određenog broja piksela od kojih svaki ima tačno određene koordinate u prostoru (x, y) , signal treba da se sampluje tako da svaki piksel bude popunjen jedinstvenom vrednošću. Vrednost amplitude signala je kontinualna, pa je kao takvu nemoguće sačuvati u računaru, zbog toga se uvodi kvantizacija gde se ta vrednost prevodi u diskretan prostor. Taj proces se naziva kvantizacija. Vrednost intenziteta piksela se obično predstavlja kao ceo broj između 0 i 255. Postupak smplovanja i kvantizacije je bitan, jer od njega zavisi kvalitet i veličina podataka koje nosi signal slike. Ako je broj smplova ili kvantizacionih nivoa mali doći će do smanjenja kvaliteta slike, tj. slika neće biti verno prikazana. Međutim ako je broj smplova ili kvantizacionih nivoa veliki broj podataka za čuvanje u memoriji će biti preveliki. U ovom postupku je bitno da se pronađe balans između ove dve stvari. U zavisnosti od potrebe će biti definisan broj smplova i kvantizacionih nivoa. Npr. nekada je bitnije da se sačuva što više slika, nego da slika bude precizna.

Slika se u digitalnom svetu predstavlja kao $f(x, y)$, gde su x i y koordinate piksela na slici, a $f(x, y)$ je vrednost piksela u koordinatama (x, y) .

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & f(0, 2) & \dots & f(0, N-1) \\ f(1, 0) & f(1, 1) & f(1, 2) & \dots & f(1, N-1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f(M-1, 0) & f(M-1, 1) & f(M-1, 2) & \dots & f(M-1, N-1) \end{bmatrix}$$

gde su N i M respektivno širina i visina slike.

Broj piksela slike se računa kao $b = M * N$, ako je slika crno bela, odnosno intenzitet piksela je prikazan kao jedna vrednost. Ako je u pitanju slika u boji onda se vrednost funkcije $f(x, y)$ predstavlja kao vektor vrednosti $[i_1 i_2 \dots i_k]$, gde je k broj kanala, odnosno broj boja kojima je prikazana slika (najčešće 3 ako je u pitanju slika u boji). Tada se broj piksela računa kao $b = M * N * k$.

1.4 Bitne relacije između piksela

Piksel p sa koordinatama (x, y) ima horizontalne i vertikalne susede $(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$. Ovo se naziva 4-suseda piksela. Ako se dodaju dijagonalni susedi $(x+1, y+1)$, $(x-1, y-1)$, $(x+1, y-1)$, $(x-1, y+1)$ onda dobijamo 8-suseda piksela. Povezanost je osobina koja se dosta koristi u obradi slike. Ako su pikseli susedi i ako imaju istu vrednost ili vrednost iz istog podopsega mogućih vrednosti piksela onda su oni povezani. Putanja niz piksela koji su povezani na slici.

Postoje nekoliko načina za računanje distance između piksela. Ako imamo piksele p , q i z sa koordinatama (x, y) , (s, t) i (v, w) , D će biti mera distance ako važi:

- a) $D(p, q) \geq 0 (D(p, q) = 0 \text{ ako } p = q)$,
- b) $D(p, q) = D(q, p)$,
- c) $D(p, z) \leq D(p, q) + D(q, z)$.

Euklidska distanca:

$$D_e(p, q) = [(x-s)^2 + (y-t)^2]^{1/2}.$$

Euklidska distanca nije baš pogodna za sliku koja ima diskretne vrednosti pa se za distancu koriste:

D_4 distanca 4-suseda:

$$D_4(p, q) = |x - s| + |y + t|.$$

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

D_8 distanca 8-suseda:

$$D_8(p, q) = \max(|x - s|, |y + t|).$$

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

1.5 Osnovne operacije nad slikom

Glavni cilj poboljšanja slike je da se slika obradi na takav način, da krajnji rezultat zadovoljava odgovarajuću primenu. tehnika koja će da se koristi za obradu slike zavisi od konkretne primene i uslova koje rezultujuća slika treba da ispuni. Obrada slike se deli na dve kategorije: Obrada slike u prostornom domenu i obrada slike u frekvencijskom domenu. Prostorni domen se odnosi na ravan u kome se slika nalazi i metode koje izvršavaju operacije nad slikom u prostoru, dok se kod drugog tipa operacije izvršavaju nad slikom koja je predstavljena u obliku signala. U ovom radu će biti objašnjena obrada slike u prostornom domenu.

Kao što je navedeno obrada slike u prostornom domenu znači da se operacije izvršavaju nad pikselima. Ove procese možemo da označimo kao $g(x, y) = T[f(x, y)]$, gde je $f(x, y)$ početna slika, $g(x, y)$ obrađena slika, a T je operator koji je definisan nad pikselima i njihovim susedima na slici f . Piksela (x, y) i njegovi susedi prave pravougaonik, čiji je centar (x, y) , nad kojim se izvršava operacija T . Vrednost piksela $g(x, y)$ se računa kao vrednost piksela $f(x, y)$ i njegovih suseda, nad kojima se izvršavaju neke operacije. Ovaj proces se obično naziva konvolucija. Osnovni postupak obrade slike se izvršava tako što se za svaki piksel slike izvrši operacija nad tim pikselom i njegovim susedima i dobijena vrednost se dodeljuje tom pikselu. Postoje dva tipa dodele: dodela na mestu računanja kada se ista slika koristi za računanje i dodelu novih vrednosti i drugi tip dodele kod kojeg se dodela vrši na drugoj slici u piksel koji ima iste koordinate kao piksel kome se pristupa na slici koja se obrađuje.

Ako vrednosti funkcija $f(x, y)$ i $g(x, y)$ predstavimo kao r i s respektivno, onda obradu slike možemo da predstavimo kao $s = T(r)$. U nastavku je da pregled nekih osnovnih transformacija nad slikom u prostornom domenu. Ove transformacije će biti predstavljene nad jednokanalnim slikama (slike čiji pikseli uzimaju samo jednu vrednost - nijansu sive boje). Što se tiče slika u boji postupak za ove transformacije je sličan. Transformacije se ili vrše nad svakim kanalom nezavisno, ili se slika prevodi u sliku sa jednim kanalom (postoji više načina), pa se primenjuje određena transformacija.

Najosnovnija operacija koja može da se izvede nad slikom je negativ. Svaki piksel slike dobija inverznu vrednost u odnosu na svoju. Ako se vrednosti piksela kreću u granicama $[0, L - 1]$, rezultujuća vrednost se dobija kao $s = L - 1 - r$. Ova operacija se koristi na crno belim slikama koje su dominantno crne, a belom bojom su prikazani bitni podaci na slici, jer je lakse uočiti neke detalje.



Slika 3: 1)slika 2)negativ

Nad slikom može da se primeni logaritamska operacija. $s = c * \log 1 + r$. Ova operacija se koristi za izračunavanje spektra slike, što može da bude korisno za operacije koje se izvršavaju u kompleksnom domenu. Takođe nad slikom možemo da primenimo i stepene funkcije $s = c * r^\nu$ ili $s = c * (r + \epsilon)^\nu$, gde su c i ν pozitivne konstante i ϵ je konstanta. Ove operacije se koriste za gama korekciju. Jedna od najpoznatijih primena je u tv uređajima koji su koriste katodne cevi. Takođe se koristi i za magnetnu rezonancu kako bi se pojačale boje i time izrazili detalji.

Nasuprot predhodno navedena dva pristupa posoje i linearne transformacije. Nad slikom mogu da se izvode razne linearne funkcije. Neke od najkorićenijih su povećavanje kontrasta ili povećanje osvetljenja. Sve ove operacije imaju isti oblik $s = T(r)$, gde je T linearna transformacija, npr. menjanje osvetljenja slike može da se izrazi kao $s = r + c$ odnosno $s = r - c$, gde je c konstanta u opsegu vrednosti koje uzimaju s i r .

Jedna od specifičnih obrada je obrada uz pomoć histograma. Histogram digitalne slike čiji pikseli uzimaju vrednosti iz $[0, L - 1]$ je diskretna funkcija $h(r_k) = n_k$, gde je r_k k -ta vrednost u opsegu vrednosti koje uzimaju pikseli, a n_k je broj piksela koji imaju tu vrednost. Obično se koristi noramlizovana f-ja $h(r_k) = n_k/n$, gde je $n = \max(n_0, n_1, \dots, n_{L-1})$, pa će sve vrednosti funkcije h biti u opsegu $[0, 1]$. Histogrami mogu da se prikažu i grafički i oni nam ukazuju na to koje su nijanse zastupljene i kojoj meri na slici.

Obrada slike uz pomoć histograma se vrši tako što se izračuna histogram slike, nad njim se izvrši transformacija, a zatim pikseli na slici dobiju odgovarajuće vrednosti na osnovu transformisanog histograma. Nad histogramom mogu da se izvrše razne operacije

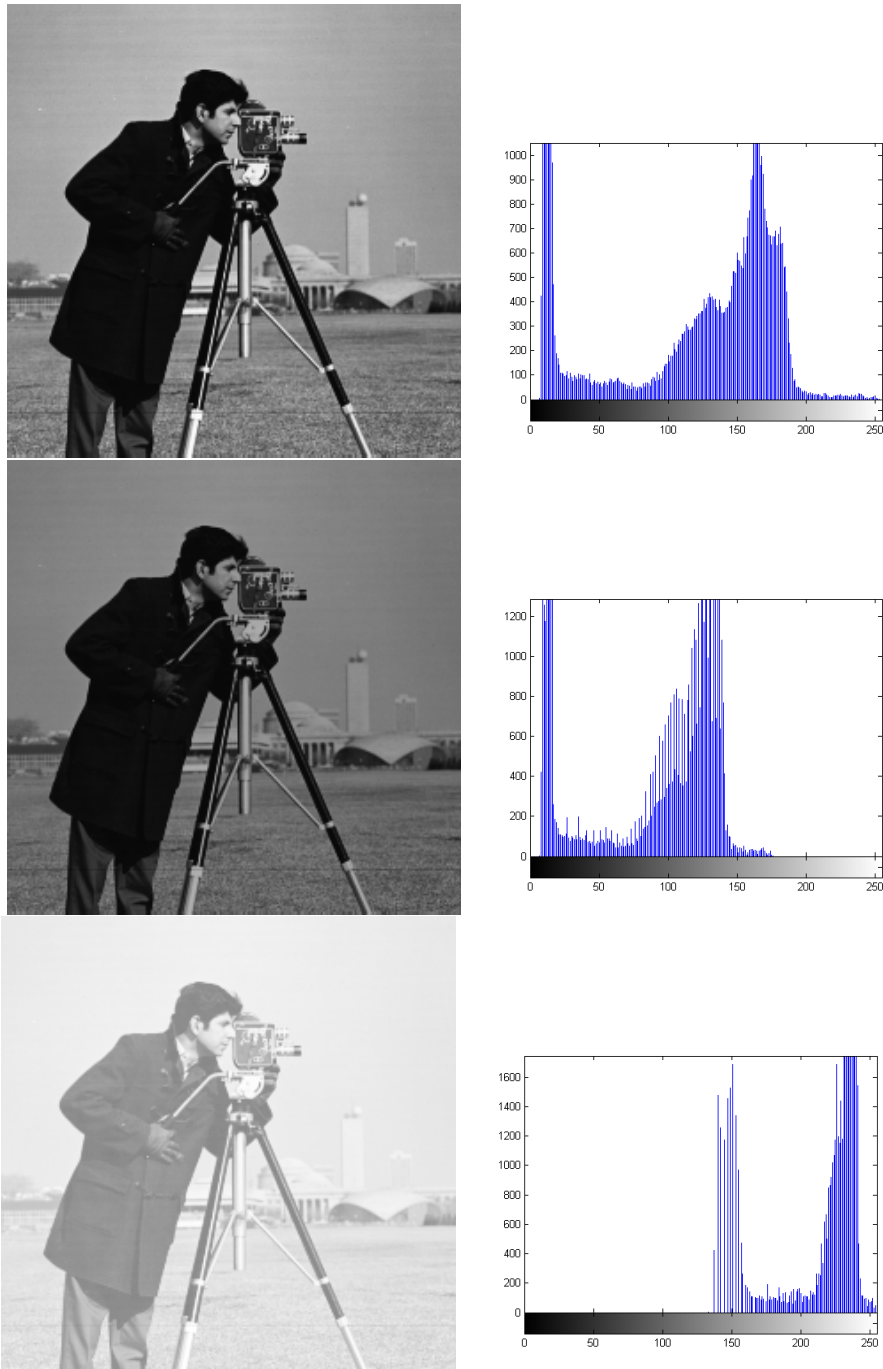


Slika 4: 1) slika 2) logaritamska transformacija 3) stepena transformacija $c = 1/\nu = 0.6$ 4) stepena transformacija $c = 1/\nu = 0.3$

u cilju poboljšanja slike. Jedna od operacija koja može da se izvrši uz pomoć histograma je histogramsko izjednačavanje. Cilj ovog postupka je da napravi uniformi histogram, tj. da na slici sve nijanse budu jednako zastupljene. Ovo je korisna transformacija ako je na slici dominiraju iste nijanse. Ako na slici dominiraju određeni broj nijansi slika može da bude nejasna, izjednačavanjem odnosno uniformisanjem histograma dobijamo sliku koja ima zastupljene sve nijanse, što će dovesti do vizuelnog poboljšanja slike. Ovde ne zalazimo detaljno u aspekte obrade slike pomoću histograma, jer ona nije od važnosti za ovaj rad.

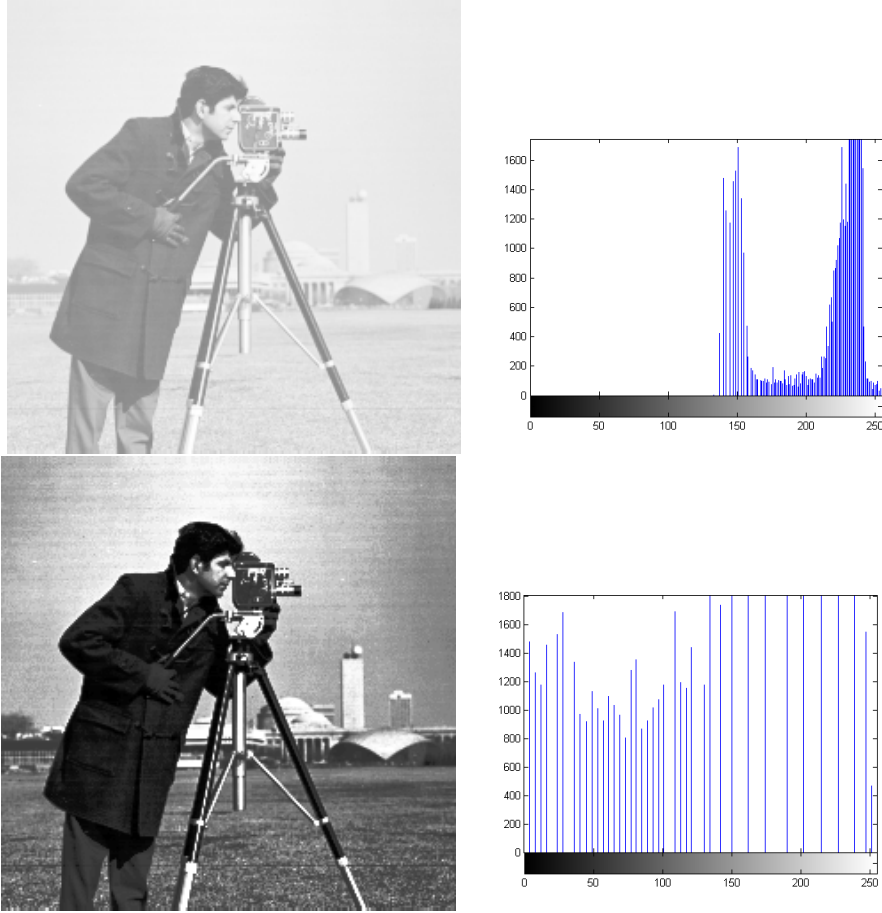
Nad slikama možemo da primenimo i aritmetičke i logičke operacije. Logičke operacije AND, OR i NOT mogu da se primene nad slikama. AND operacija može da se koristi da bi se sa slike izvukao određeni region od značaja. Region može da se izvuče tako što će se primeniti AND operacija nad slikom i maskom. Maska je slika koja sadrži informaciju koji deo slike želimo da ostavimo. Maska se formira tako što se na mestima koje želimo da sačuvamo stavljamo maksimalnu vrednost koju uzimaju pikseli. Ako je opseg nijansi $[0, 1]$ onda je to 1, na svim ostalim mestima stavlja se 0. Rezultat ove operacije biće deo slike određen maskom. Na isti način se primenjuju i OR i NOT operacije. Logičke operacije su značajne za morfološku obradu slike koja je spomenuta kao jedna od oblasti kojima se bavi obrada slike.

Nad slikama se često primenjuje operacija oduzimanja. Ako želimo da vidimo da li se dve slike razlikuju možemo da ih oduzmemo jednu od druge, što će da nam ukaze



Slika 5: levo: slika desno: njen histogram

na razlike, jer će mesta gde su pikseli isti biti 0 tj crna a mesta gde ima razlike će biti predstavljena određenom nijansom. U ovu svrhu može da se koristi i deljenje. Sabiranje i množenje može da se koristi u svrhu dodavanja detalja na slici.



Slika 6: 1)slika 2)histogram 3)slika i 4)histogram nakon histogramskog izjednačavanja

2 Osnovni tipovi filtara

2.1 Osnovni pojmovi

Kao što je ranije spomenuto prilikom obrade slike operacije se izvršavaju nad pikselom i njegovim susedima sa jedne strane i ogovarajućom podslikom sa druge. Ta pod slika nosi informacije o koje se koriste za filtriranje slike. Ta podslika (odnosno pravougao-nih piksela) se naziva kernel ili prozor. Koncept filtriranja slike je uzet iz matematike odnosno analize i matematički predstavljen taj proces je ustvari konvolucija $f * w$ gde je f predstavlja sliku koja se filtrira, w predstavlja kernel za filtriranje. Za svaki piksel slike koja se filtrira računa se nova vrednost uz pomoć kernela w i odgovarajućeg piksela i njegovih suseda, iz slike. Ako imamo kernel oblika 3×3 računanje vrednosti piksela (x, y) izgleda ovako

$$R = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots + w(0, 0)f(x, y) + \dots + w(1, 0)f(x + 1, y) + w(1, 1)f(x + 1, y + 1).$$

$$R = \begin{bmatrix} f(x - 1, y - 1) & f(x - 1, y) & f(x - 1, y + 1) \\ f(x, y - 1) & f(x, y) & f(x, y + 1) \\ f(x + 1, y - 1) & f(x + 1, y) & f(x + 1, y + 1) \end{bmatrix} .* \begin{bmatrix} w(-1, -1) & w(-1, 0) & w(-1, 1) \\ w(0, -1) & w(0, 0) & w(0, 1) \\ w(1, -1) & w(1, 0) & w(1, 1) \end{bmatrix}$$

Koeficijet kernela $w(0,0)$ odgovara pikselu $f(x,y)$, što ukazuje na to da je maska centrirana oko piksela (x,y) . Obzirom da kernel mora da bude centriran oko jednog piksela, za kernel dimenzije $m \times n$, važi da je $m = 2a + 1$ i $n = 2b + 1$, gde su a i b pozitivni celi brojevi. Najmanji kernel koji ima smisla je 3×3 , a trivijalni slučaj predstavlja kernel 1×1 , koji se primenjuje samo kod osnovnih operacija. Generalno filtriranje slike f dimenzija $M \times N$, korišćenjem kernela w veličine $m \times n$ izrazom:

$$g(x,y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s,t)f(x+s,y+t),$$

gde je $a = (m-1)/2, b = (n-1)/2$. Da bi se filtrirala cela slika potrebno je da se i zračuna vrednost $g(x,y)$ za svako $x \in [0, M-1]$ i za svako $y \in [0, N-1]$.

Matrica kernela može da se obelži i ovako za primer kernela 3×3 :

$$R = \begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix} \cdot * \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix}$$

$$R = w_1 z_1 + w_2 z_2 + \dots + w_{mn} f_{mn} = \sum_{i=1}^{mn} w_i z_i.$$

w_i predstavlja vrednosti kernela, z_i predstavlja vrednosti odgovarajućih piksela slike, a m i n predstavljaju dimenzije kernela.

Možemo da kažemo da se filtriranje obavlja tako što kernel prolazi kroz celu sliku. Ovaj pristup se koristi kako za linearne tako i za nelinearne filtre. Bitno je obratiti pažnju na piksele slike koji se nalaze uz ivicu slike. Ti pikseli nemaju sve susede pa oni moraju na neki način da se dodele. To može da se reši na više različitih načina, a neki od uobičajnih su postavljanje vrednosti tih suseda na 0 ili neku drugu konstantu ili uzimanje vrednosti simetričnih piksela u odnosu na piksel za koji se vrši konvolucija.

Dva osnovna tipa filtera su filteri za uklanjanje šuma i zamućivanje slike koji se nazivaju smoothing filteri i filteri za izoštravanje slike, odnosno filteri za određivanje ivica ivica koji se nazivaju sharpening filteri. Pošto ne postoji tačan prevod za ove nazive dalje u radu se koriste imena smoothing i sharpening.

2.2 Smoothing filtri(filtri za uklanjanje šuma)

Ovaj tip filtra se koristi za takozvano zamućivanje (*eng. blurring*) i za redukciju šuma. Blurring se koristi za uklanjanje malih detalja sa slike, u procesu ekstrakcije objekata sa slike. Takođe može da se koristi za popunjavanje prekida u linijama na slici, koje mogu da se jave zbog šuma na slici. Rezultat ovog filtra je u suštini srednja vrednost piksela koji su sadržani u okolini kernela koji se koristi za filtriranje. Zbog toga se ovi filtri nazivaju i filteri srednje vrednosti (*eng. averaging filters*).

Ideja ovog filtriranja je da se vrednost konkretnog piksela zameni sa srednjom vrednošću piksela iz njegove okoline, koji su definisani kernelom. Rezultat ovog filtra biće slika koja ima ublažene prelaze između vrednosti susednih piksela. Dva susedna piksela koja se dosta razlikuju biće sličnija po vrednosti. Pošto se pikseli, koji predstavljaju šum, dosta razlikuju po vrednost od susednih piksela, na ovaj način će se oni uklopiti u okolinu. Loša strana ovakvog filtriranja se ogleda u tome što će mesta koja predstavljaju ivice ili linije na slici početi da gube oštrinu. Naime na mestima na kojima se nalaze ivice imamo veliku razliku u susednim pikselima, pa će primenom ovog filtra ta razlika postati manja što će dovesti do gubljenja oštine tih ivica.

$$1/9 * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad 1/16 * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Ovo su tipični primeri smoothing filtra dimenzija 3×3 . Levo je najprostiji primer ovog tipa filtara. Primenom ovog kernela svaki piksel će dobiti kao rezultat prosečnu vrednost susednih piksela koji odgovaraju obliku ovog kernela. Koeficijent $1/9$ služi da bi se dobila strdnja vrednost. Rezultat koji će biti upisan u piksel je $R = 1/9 \sum_{i=1}^9 z + i$. Kao što vidimo na primeru kernela sa desne strane, ne moraju svi susedni pikseli da učestvuju podjednako u sumi. Na ovom primeru vidimo da pikseli različito učestvuju u sumi sa različitim koeficijentima koji su definisani u kernelu. Te vrednosti se nazivaju težine. Vidimo da se i koeficijent koji stoi uz kernel razlikuje u odnosu na primer levo. On je $1/16$, jer je suma brojeva iz kernela 16. U većini slučajeva se ovaj koeficijent određuje na ovaj način. Na ovom primeru vidimo da se prilikom filtriranja daje prednost samom pikselu, jer uz njega stoji najveći koeficijent 4, zatim se daje prednost horizontalnim i vertikalnim susedima, oni imaju koeficijente 2 i tek na kraju su dijagonalni susedi sa koeficijentom 1. Ovi koeficijenti u nekim slučajevima mogu da budu i 0.



Slika 7: 1)slika 2)slika filtrirana kernelom 3×3 3) 5×5 i 4) 9×9

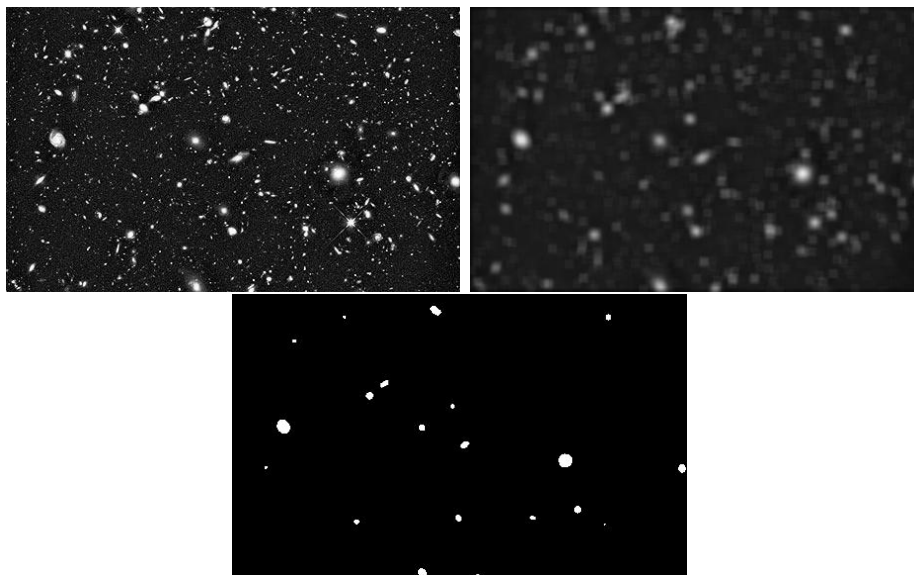
Naravno ova priča važi i za filtre koji koriste kernel bilo koje dimenzije $m \times n$ (m i n su neparni). Ako imamo kernel ovih dimenzija opšta formula za izračunavanje vrednosti rezultujućeg piksela igledaće ovako:

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)},$$

gde su g i f ulazna i rezultujuća slika, (x, y) koordinate piksela za koji se trenutno računa vrednost, $a = (m - 1)/2$, $b = (n - 1)/2$, w težine u kernelu. Da bi cela slika bila filtrirana ova funkcija treba da se izvrši za svako $x \in [0, M - 1]$ i za svako $y \in [0, N - 1]$.

Kao što je navedeno kerneli ovih filtra su proizvoljnih veličina. Za ovaj tip filtra se obično koriste $n \times n$ kerneli, jer oni uzimaju simetrične piksele i najbolje određuju koliko se neki piksel razlikuje u odnosu na njegovu okolinu. Težine definisane u kernelu određuju kojim susedima se daje prednost prilikom računanja srednje vrednosti. Npr. ako neka slika ima dosta vertikalnih ivica, težina koja će stojati uz piksele, koji su vertikalni susedni u odnosu na piksel za koji se računa vrednost, može da bude 0 da bi se izbeglo gubljenje oštine. Veličina dimenzija kernela utiče na to koliko se brojačano uzima suseda, što će na krajnjem rezultatu uticati na to u kojoj je količini slika zamućena. Što su veće dimenzije kernela, veća će biti zamućenost na slici. Pošto je svaka slika drugačija, ona će zahtevati određenu kombinaciju težina i dimenzije kernela. Ovo se obično određuje probom različitih kernela da bi se utvrdilo koji je najbolji za konkretnu sliku.

Kao što je spomenuto ranije jedna od primena ovog tipa filtara je predprocesiranje slike zarad uklanjanja sitnih detalja koji nisu od važnosti. Ovo se primenjuje prilikom ekstrakcije objekata sa slike mogu da se u određenoj meri uklone sitni objekti da bi se olakšala ekstrakcija bitnih objekata. Veličina kernela će odrediti koja koji će objekti biti uklonjeni sa slike. Ako se koristi kernel veličine 3×3 sa slike će biti uklonjene samo tačke veličine 1 piksela. Povećanjem dimenzija kernela biće uklonjene i veći objekti. Odabir veličine kernela zavisiće od toga šta želimo da odstranimo sa slike.



Slika 8: 1)slika 2)slika filtrirana kernelom 9×9 3)slika pretvorena u binarnu sliku(samo crna i bela)

Pored navedenih linearnih filtra postoje i nelinearni filtri bazirani na sortiranju vrednosti piksela koji se nalaze u oblasti koja je obuhvaćena filtriranjem, odnosno određena kernelom. Najpoznatiji primer je mediana filter. Vrednosti u kernelu odnosno težine koje se koriste u ovom filtru su 1. Pikseli iz željene oblasti se sortiraju i bira se vrednost koja

se u tako sortiranom nizu piksela nalazi na sredini. Ona se dodeljuje pikselu koji se trenutno obrađuje. Ako koristimo kernel veličine 3×3 nakon što sortiramo njihove vrednosti za rezultat ćemo odabrati 5-ti piksel po veličini (nije bitno da li se sortiraju rastuće ili opadajuće). Dakle vrednost piksela se zamenjuje sa pikselom iz njegove okoline koji ima statistički srednju vrednost. Median filter je efikasan za uklanjanje slučajnog šuma, a posebno je efikasan za uklanjanje šuma koji se naziva biber i so, a specifičan je po tome što se iskazuje preko crnih i belih tačaka, koje su minimalna i maksimalna vrednost koju piksel može da uzme, pa se zbog toga naziva i impulsni šum. Za ove primene mediana filter je efikasniji od predhodno navedenih filtra, jer bira vrednost koja se najviše javlja u okolini određenog piksela.



Slika 9: 1)slika 2)slika filtrirana median filterom

2.3 Sharpening filtri(filtri za određivanje ivica)

Glavni cilj ovog tipa filtara je da na slici obeleže fine detalje, kao što su linije i prelazi, ili da poboljša izgled tih detalja, koji mogu biti mutni ili da imaju grešku, koja je posledica određenog metoda pribavljanja slike. Primena ovih filtara je veoma široka i varira od elektronskog štampanja preko medicine do industrijske inspekcije i autonomnih vojnih i sistema nadzora.

Filteri opisani u predhodnom poglavlju koriste sume i srednje vrednosti da bi filtrirali sliku. Ovaj proces je analogan integraciji, pa možemo da zaključimo da sharpening filtri, koji imaju suprotni cilj u odnosu na smoothing filtre, koriste diferenciranje da bi postigli željeni rezultat. Delovanje funkcije izvoda je proporcionalno stepenu nepovezanosti slike u tački, gde je izvod primenjen. Diferenciranje slika će naglasiti ivice i prelaze, kao i šum i sve delove gde se vrednosti piksela naglo menjaju (prelaz između dve boje na slici koje se dosta razlikuju u nijansi npr crna i bela).

Da bi objasnili uticaj izvoda na sliku, objasnimo osnovne koncepte preko jedno dimenzionalnih izvoda. Izvod diskretne funkcije se definiše kao razlika između dva uzastopna elementa. Postoji više načina za definisanje izvoda nad slikom, ali svi oni zahtevaju da funkcija izvoda ispunjava sledeće: 1) jednaka je nuli u ravnim segmentima (segmenti koji imaju konstantnu vrednost); 2) različita je od nule u segmentima gde postoji nagla ili konstantna promena; 3) različita je od nule duž segmenata gde se javlja konstantna promena; Slično i za funkciju drugog izvoda važi da je: 1) jednaka je nuli

u ravnim segmentima; 2) različita je od nule na mestima gde se završava nagla ili konstantna promena; 3) jednaka je nuli duž segmenata gde se javlja konstantna promena; S obzirom da pikseli slike uzimaju diskretne vrednosti vrednost izvoda se kreće u granicama tih vrednosti. Definicija jednodimenzionalnog izvoda je:

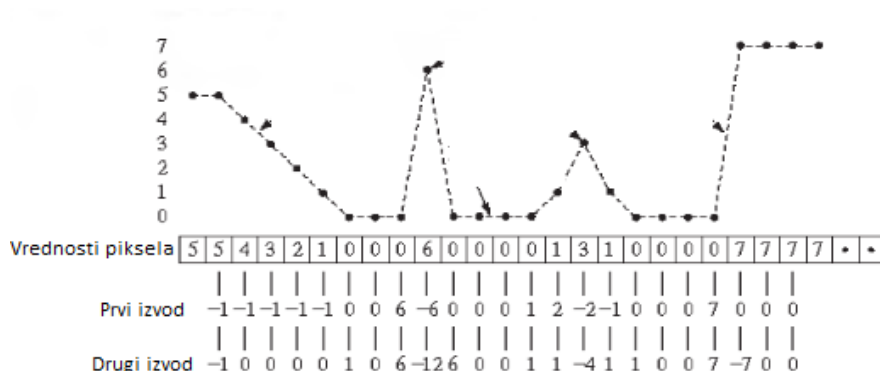
$$\frac{\delta f}{\delta x} = f(x+1) - f(x), \text{ parcijalni izvod po x osi.}$$

$$\frac{\delta f}{\delta y} = f(y+1) - f(y), \text{ parcijalni izvod po y osi.}$$

Slično se za drugi izvod definiše:

$$\frac{\delta^2 f}{\delta x} = f(x+1) + f(x-1) - 2f(x), \text{ drugi parcijalni izvod po x osi.}$$

$$\frac{\delta^2 f}{\delta y} = f(y+1) + f(y-1) - 2f(y), \text{ drugi parcijalni izvod po y osi.}$$



Slika 10: Prvi i drugi izvod nad nizom piksela

Ako pogledamo sliku 10 možemo da uočimo da je prvi izvod različit od nule u segmentu gde je pad konstantan (svi susedi se razlikuju za 1), dok je drugi izvod različit od nule na početku i kraju tog segmenta. Pošto se ivice na slici iskazuju u ovom tranzicijom, možemo da zaključimo da će prvi izvod da proizvede debele ivice, dok će drugi izvod proizvesti tanke ivice. Npr. ako imamo belu liniju na crnoj pozadini, koja ima određenu debljinu, prvi izvod će izdvojiti celu liniju. Drugi izvod će, sa druge strane, izdvojiti dve finije linije koje ukazuju na prelaz sa crne na belu boju i sa bele na crnu liniju. Ako pogledamo deo na kome imamo izolovani skok u vrednosti piksela, možemo da primetimo da je drugi izvod izraženiji nego prvi. Ovo je normalno jer je drugi izvod bolji u pronalaženju oštih promena. Na osnovu ovoga možemo da očekujemo da će drugi izvod bolje da pronalazi finije detalje i šum. Ako je u pitanju manja promena prvi i drugi izvod će se ponašati slično. Na osnovu svega rečenog možemo da zaključimo da primenom prvog izvoda dobijamo debele lvice, dok primenom drugog izvoda dobijamo finije linije i izolirane tačke. Prvi izvod nam ukazuje na mesta gde je promena konstantna. Drugi izvod nam ukazuje na mesta gde je promena velika i izolovana. Ugalvnom se drugi izvod više koristi u konkrentim primenama zbog toga što naglašava finije detalje. Zbog toga ćemo se ovde fokusirati na filtre koje koriste drugi izvod. Bitno je naglasiti da se filteri koji koriste prvi izvod analogno koriste i imaju iste osobine kada se koriste.

Nad slikom se primenjuju dvo-dimenzionalni izvodi, jer slika ima dve dimenzije. Na osnovu ovoga se formiraju kerneli koji se koriste u primeni ovih filtara. Nas interesuju izotropski filteri, tj. filteri koji su invarijantni u odnosu na pravac promena, koje se javljaju na slici. Drugim rečima ovi filteri su nezavisni od rotacije, što znači da će pronaći sve ivice bez obzira na njihov ugao u ravni. Ako sliku rotiramo ovakav filter će pronaći iste ivice kao na originalnoj slici. Najprostija funkcija izvoda koja se koristi je *Laplacian*:

$$\Delta^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}.$$

Pošto su izvodi linerane funkcije možemo da zaključimo da je i *Laplacian* linearna funkcija. Da bi smo ovu funkciju mogli da koristimo za obradu slike moramo da je prevedemo u diskretnu formu. Iskoristićemo funkciju koju smo koristili da definišemo jedno-dimenzionalni izvod. Drugi parcijalni izvod po x-osi će izgledati ovako:

$$\frac{\delta^2 f}{\delta x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y).$$

Slično, drugi parcijalni izvod po y-osi će izgledati ovako:

$$\frac{\delta^2 f}{\delta y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y).$$

Na osnovu ovoga možemo da zaključimo da će laplacian funkcija izgledati ovako:

$$\Delta^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y).$$

Ovako implementirana funkcija može da se predstavi sledećim kernelom.

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

Da bi se u obzir uzele i promene koje se javljaju dijagonalno u odnosu na tačku koju posmatramo koristi se modifikovan kernel koji izgleda ovako.

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

Ovde se i dijagonalni susedi uzimaju obzir. Ovo je kao da imamo definisana još dva parcijalna izvoda nad dijagonalama, pa se za obe dijagonale dodaju $-2f(x, y)$. Zbog toga je koeficijent koji odgovara pikselu (x, y) jednak -8. Takođe možgu da se koriste i kerneli koji imaju težine koje su obrnutog znaka u odnosu na predhodno navedene. Oni će dati iste rezultate, ali njihov znak treba da se uzme u obzir kada se ovaj pristup kombinuje sa drugim operacijama, jer će rezultati biti drugačijeg znaka.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Pošto je laplacian funkcija izvoda, ona će na slici otkriti promene u nijansama, kao što su tačke, ivice i linije. Možemo da žaključimo da će rezultat primene laplacin-a biti slika gde će promene biti označene nekim nijansama bele i sive, a delovi u kojima nema promena biće crni. Ako želimo da na originalnoj slici naglasimo ivice mi možemo da dodamo vrednosti rezultata primene laplacian-a odgovarajućim pikselima. Pošto laplacian može da da i negativne, moramo da obratimo pažnju na to koji oblik kernela koristimo pozitivni ili negativni. Ako je središnji element kernela pozitivan onda ćemo rezultat dodati, a u suprotom oduzeti od slike.



Slika 11: 1)slika 2)laplacian 3)poboljšana slika dodavanjem laplacian-a

Ova operacija može da se implementira u jednom prolazu tako što će da se modifikuje kernel. Operacija koja je izvršena na predhodnoj slici može da se predstavi ovako:

$$g(x, y) = f(x, y) - \Delta^2 f(x, y) = 5f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y-1) + f(x, y+1)].$$

Na osnovu ove funkcije možemo da zaključimo da će kernel da izgleda ovako:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Postoji još jedan način za postizanje poboljšanja slike. Naime ivice slike mogu da se odrede tako što će od originalne slike oduzeti slika nad kojom je primenjen smoothing filter. Pošto smoothing filter ima suprotno dejstvo od sharpening filtera oduzimanje će dati isti rezultat kao primena sharpening filtera.

Kao što je rečeno ranije za filtriranje slike može da se koristi i prvi izvod. Prvi izvod se za filtriranje koristi tako što se računa dužina gradijenata po x i y osi. Za funkciju

$f(x, y)$ gradijent u koordinatama (x, y) se definiše kao vektor:

$$\Delta \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix}$$

Dužina vektora se računa kao:

$$\Delta f = \text{mag}(\Delta \mathbf{f}) = [G_x^2 + G_y^2]^{1/2} = [(\frac{\delta f}{\delta x})^2 + (\frac{\delta f}{\delta y})^2]^{1/2}.$$

Gradijenti su linearne funkcije, ali njihova funkcija dužine nije jer sadrži stepene operacije. Sa druge strane gradijenti nisu invarijantni u odnosu na rotaciju dok funkcija njihove dužine jeste, pa se zbog toga ona koristi. Iako nije tačno dužina vektora gradijenta se često naziva gradijent. Pošto nije lako primeniti ovu funkciju nad slikom koja ima diskretne vrednosti za potrebe obrade slike se obično koristi aproksimacija dužine gradijenta $\Delta \approx |G_x| + |G_y|$. Ova funkcija je lakša za izračunavanje i prepoznaje promene, ali generalno gubi osobinu invarijacije u odnosu na rotaciju. Kao što je slučaj kod laplacian-a izotropna osobina očuvana je zahvaljujući odabiru maske i dodavanju dijagonalnih elemenata.

Ako je region na slici za kernel veličine 3 x 3 definisan kao

$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}$$

gde z_5 predstavlja $f(x, y)$, a z_1 predstavlja $f(x - 1, y - 1)$ onda je $G_x = (z_9 - z_5)$ i $G_y = (z_8 - z_6)$. Ovo je prva predložena definicija za računanje prvih izvoda. Sledi da je gradijent:

$$\Delta f \approx |z_9 - z_5| + |z_8 - z_6|.$$

Kernel za ovako definisan gradient se naziva Robertsov kros-gradijent operator i izgleda ovako:

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

Pošto za kernele nije uobičajno da imaju parne dimenzije uvodi se ovakva definicija gradijenta. Ako imamo kernel veličine 3x3 funkcija gradijenta izgleda ovako:

$$\Delta f \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|.$$

Kerneli za ovu funkciju izgledaju ovako:

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

Ovi kerneli su poznatiji kao Sobelovi operatori.



Slika 12: 1) slika 2) sobel

2.4 Kombinovanje filtara

Predhodno prikazani filtri za uklanjanje šuma i određivanje ivica imaju veliku primenu u obradi slike. U većini slučajeva oni se koriste kao deo nekog većeg procesa. U predhodnim poglavljima je navedeno nekoliko slučajeva. Smoothing filtri se koriste da bi se uklonili objekti koji nisu od važnosti za ekstrakciju objekata sa slike. Takođe ovi filtri mogu da se koriste za određivanje ivica i poboljšanje slike, tako što će se na sliku dodati detalji dobijeni oduzimanjem slike i njene zamućene verzije, koja se dobija primenom smoothing filtra. Detalji dobijeni primenom filtara za određivanje ivica i detalja mogu da se dodaju na sliku i cilju poboljšanja i naglašavanja detalja.

Jasno je da se ovi filtri koriste u kombinaciji sa drugim filtrima i kao deo nekog većeg procesa. Oni mogu međusobno da se kombinuju. Postoje i složeni filtri koji su napravljivi koristeći koncepte ova dva tipa filtara. To je najmoćniji tip filtara koji kao rezultat daje slike kod kojih se uklanja šum, što je odlika smoothing filtara, ali se pri tome slika ne zamućuje, čuvaju se ivice, što je odlika sharpening filtara. Dva najpoznatija ovakva filtra su *Bilateral filter* i *Guided filter* kojim se ovaj rad detaljnije bavi u nastavku.

3 Guided filter

3.1 Uvod

Veliki broj aplikacija u oblastima kompjuterske vizije i računarske grafike sadrže proces zadužen za filtriranje slika, u cilju uklanjanja i/ili izvlačenja podataka sa slike. Linearni filtri invarijantni na tranziciju, kao što su mean, Laplacian, Gausov i Sobelovi filtri, se široko koriste prilikom obrade slike, za restauraciju, zamućivanje i izoštravanje, detekciju ivica, izvlačenje podataka, dinamičku kompresiju itd. Ovi filteri su prostorno invarijantni i nezavisni od sadržaja slike.

Pa ipak oni nekad nemogu da zadovolje očekivani odgovor neke obrade. Zbog toga se ovde uvodi pojam slike vodiča (*eng. guide*), koja sadrži dodatne informacije koje mogu da se iskoriste pri filtriranju i poboljšaju rezultat. Ovaj koncept je prvi put upotrebljen u procesu difuzije (*eng. anisotropic diffusion*), u kojem gradijenti filtrirane slike služe kao vodič za ovaj proces i tako ovaj proces izbegava da uništi oštrinu ivica. Takođe WLS filter (*eng. weighted least squares*) koristi ulaznu sliku kao vodič, optimizuje kvadratnu funkciju,

koja je ekvivalentna procesu difuzije, koji je malopre pomenut. U mnogim primenama kao slika vodič može da se koristi različita slika u odnosu na onu koja se filtrira. Npr. kod procesa uklanjanja magle sa slike, kao slika vodič se koristi procena količine smetnji na slici, koja treba da se filtrira. Većina ovih procesa je implementirana, optimizacijom kvaratne funkcije, čiji su parametri uzeti iz slike vodiča. Rešenje je predstavljeno rešenjem velikih retko popunjenih matrica. Ovakve nehomogene matrice predstavljaju filtere koji su invarijantni u odnosu na tranziciju. Iako su ovakvi filteri efikasni, oni mogu da budu zahtevni za izvršavanje.

Drugi način da se iskoristi slika vodič je da se njene osobine eksplicitno iskažu u kernelu. *Bilateral filter* je najpoznatiji primer takvog filtera. Rezultat koji piksel dobija, kod ovog filtriranja, je težinska prosečna vrednost piksela i njegovih suseda, gde težine zavise od sličnosti piksela u vodič slici. Vodič slika može i sama da bude ulazna slika u ovom filtru. *Bilateral filter* može da izgladila siten fluktuacije, a da pri tome zadrži ivice. Problem ovog filtera je što ima loše artefakte koji se javljaju blizu ivica. Dolazi do promene boje blizu ivica. Takođe problem je implementirati efikasan algoritam, metodi koji ubrzavaju ovaj algoritam sa druge strane utiču na preciznost njegove obrade.

U radovima [3] i [4] se predlaže *Guided filter* ili vođeni filter. Rezultat filtriranja je lokalna linearna transformacija slike vodiča. Sa jedne strane ovaj filter ima dobre osobine, pošto čuva ivice slike, a sa druge strane nema loše artefakte kao *Bilateral filter*. *Guided filter* može da se koristi kao smoothing filter, ali njegova namena prevazilazi samo to. Uz pomoć slike vodiča rezultat ovog filtriranja će biti slika koja je manje zamućena od originalne slike i bolje struktuirana od nje. Ovaj filter se pokazao dobro u raznim primenama kao što su poboljšanje slike, uklanjanje šuma, HDR kompresija, matting itd. Još jedan prednost *Guided filter*-a je to što ima složenost $O(n)$, gde je n broj piksela slike. Može da se koristi kako na jednokanalnim slikama tako i na višekanalnim slikama. Prema radovima [3] i [4] ovo je najbrži filter ove namene, koji na CPU implementaciji obrađuje 1 mega piksel (1024 x 1024) za 40ms.

3.2 Slične ideje

Bilateral filter je tipičan primer ovog tipa filtera i jedan od prvih koji je zamišljen. To su filteri koji služe za smoothing ali pri tome zadržavaju ivice slike. Rezultat filtriranja svakog piksela slike je težinska srednja vrednost piksela i njegovih suseda. Ima široku primenu. Koristi se za redukciju šuma, HDR kompresiju, dekompoziciju detalja. Generalizovan je u implementaciji *joint bilateral filter*-a, gde se težine računaju u odnosu na sliku vodiča, za koju se koristi posebna slika. Ovaj filter se koristi kada se od filtriranja ne očekuje da pruži informacije o ivicama, npr. kada slika ima dosta šuma ili kada je rezultat koji treba dalje da se iskoristi. Iako je *Bilateral filter* veoma popularan i intuitivan za shvatanje, on ima limitacije.

Predhodno spomenuti loši artefakti koji se javljaju blizu ivica, jer je gausova funkcija nestabilna blizu ivica zbog velikih razlika u vrednostima piksela. Drugi problem se odnosi na složenost algoritma. Brute-force implementacija ima složenost $O(Nr^2)$, gde je N broj piksela na slici, a r je radius kernela (dimenzija kernela je $a = 2r + 1$). Jasno je da složenost zavisi od veličine kernela što može da bude loše ako se koriste veliki kerneli. Postoje rešenja koja koriste razne metode kao što su distribuirani histogrami koji složenost smanjuju na $O(n \log r)$. Postoji i metod za redukciju složenosti na $O(n)$ korišćenjem integralnih histograma (kao lookup tabela). Međutim konstrukcija histograma je primena filtera u prostornom domenu, pa možemo da zaključimo da ovaj proces zahteva dodatne

korake. Postoji još jedan pristup čija je složenost $O(n)$, a koji se oslanja na agresivno smanjenje slike *subsampling*, čime se gube detalji. Važno je naglasiti da sva spomenuta rešenja koja imaju lineranu složenost, sa druge strane utiču na degradaciju preciznosti. Pošto *Bilateral filter* ima svoje limitacije ljudi su krenuli da istražuju nove metode za filtre koji uglašavaju sliku, a pri tome čuvaju ivice.

Nasuprot lineranim filterima postoje implicitni filteri koji se dobijaju optimizacijom funkcije greške i rešavanjem linearnog sistema jednačina, što je ekvivalentno implicitnom filtriranju slike korišćenjem inverzne matrice. Ovi filteri takođe imaju široku primenu u segmentaciji slike, kolorizaciji slike, itd. Iako su ovakvi filteri efikasni i daju kvalitetan rezultat, međutim računanje linearnih sistema može da bude veoma složeno. Direktna rešenja kao što je gausova eliminacija, nisu praktični, zbog velikog zauzeća memorije. Iterativna rešenja sporo konvergiraju. Uočeno je da su implicitni filteri slični kao eksplisiti.

Očuvanje ivica može da se postigne i filterima koji ne koriste srednje vrednosti susednih piksela. Najpoznatiji je median filter koji je objašnjen u poglavlju 2.2.

3.3 Definicija filtra

Prvo ćemo definisati uopšteni proces linearnog invariantnog filtriranja, koji koristi sliku vodič I , sliku koja se filtrira p i sliku koja je rezultat filtriranja q . I i p su zadati na početku procesa i oni mogu da budu identični. Rezultat filtriranja piksela i će biti:

$$q_i = \sum_j W_{ij}(I)p_j,$$

gde su i i j indeksi piksela, kernel filtera W_{ij} je funkcija slike vodiča I i nezavisna je od p . Ovaj filter je linearan u odnosu na p .

Sada ćemo definisati guided filter. Predpostavka je da je guided filter loklano linearan model između slike vodiča I i rezultata filtriranja q . Za q predpostavljamo da je linearna transformacija slike I u prozoru w_k centriranom u pikselu k :

$$q_i = a_k I_i + b_k, \forall i \in w_k,$$

gde su (a_k, b_k) linearni koeficijenti za koje ćemo predpostaviti da su konstantni u w_k . Za w_k koristimo kvadratni prozor radiusa r . Ovaj lokalni linearni model garantuje da će q imati ivicu samo ako I ima tu istu ivicu, jer je $\Delta q = a \Delta I$. Da bi odredili linearne koeficijente (a_k, b_k) , trebaju nam ograničenja slike koja se filtrira p . Rezultat filtriranja određenog piksela q_i se dobija tako što se od p_i oduzimaju neke neželjene komponente kao što su razni detalji i šum: $q_i = p_i - n_i$. Mi tražimo rešenje koje će minimizovati razliku između p i q , pod sulovom da je ta funkcija linearna. Konkretno za svaki prozor w_k treba da minimizujemo sledeću funkciju greške:

$$E(a_k, b_k) = \sum_{i \in w_k} ((a_k I_i + b_k - p_i)^2 + \varepsilon a_k^2).$$

ε je regularizacioni parametar koji služi da smanji uticaj a_k , ako je njena vrednost velika. Minimizacija ove jednačine predstavlja problem koji se rešava linearnom regresijom. Ovde ne ulazimo u detalje ovog rešenja već dajemo rešenja direktno:

$$a_k = \frac{\frac{1}{|w|} \sum_{i \in w_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \varepsilon},$$

$$b_k = \overline{p_k} - a_k \mu_k.$$

μ_k i σ_k^2 su srednja vrednost i varijansa piksela slike I u prozoru w_k , a $|w|$ je broj piksela u w_k , $\overline{p_k} = \frac{1}{|w|} \sum_{i \in w_k} p_i$ je srednja vrednost piksela slike u w_k . Kada smo sračunali linearne koeficijente (a_k, b_k) , možemo da nastavimo sa postupkom za izračunavanje rezultata q_i . Pošto je će piksel i biti deo različitih prozora w_k , prilikom filtriranja, ne možemo da koristimo jedinstvene koeficijente (a_k, b_k) za jedan piksel. Zbog toga ćemo u obzir uzeti sve moguće vrednosti koeficijenata za q_i . Kada izračunamo vrednosti (a_k, b_k) za svaki piksel ulazne slike, rezultat filtriranja će biti:

$$q_i = \frac{1}{|w|} \sum_{k|i \in w_k} (a_k I_i + b_k).$$

Pošto su prozori w_k kvadratnog oblika i simetrični su, možemo da primetimo da je $\sum_{k|i \in w_k} a_k = \sum_{k \in w_i} a_k$, pa formula može da se napiše ovako:

$$q_i = \overline{a_i} I_i + \overline{b_i},$$

gde su $\overline{a_i} = \frac{1}{|w|} \sum_{k \in w_i} a_k$ i $\overline{b_i} = \frac{1}{|w|} \sum_{k \in w_i} b_k$ srednje vrednosti i y svih prozora u kojima je q_i sadržan. Sa promenom koja je uvedena u ovoj jednačini gubimo sličnost između Δq i ΔI na mestima gde su ivice, jer linearni koeficijenti $(\overline{a_i}, \overline{b_i})$ variraju u zavisnosti od prostora. Međutim pošto su oni rezultat filtra srednje vrednosti, možemo da očekujemo da će njihovi graijenti biti mnogo manji nego gradijenti I blizu jakih ivica. U ovoj situaciji možemo da kažemo da je $\Delta q \approx \overline{a} \Delta I$, što znači da će nagle promene intenziteta na I biti sačuvane i na q .

U nastavku je dat pseudo kod algoritma, koji je definisan na osnovu predhodnih jednačina. f_{mean} je filter srednje vrednosti sa radiusom r . *corr*, *var* i *cov* označavaju redom korelaciju, varijansu i kovarijansu.

Algoritam Guided filter

Ulaz: slika koja se filtrira p , slika vodič I , radius r , parametar regularizacije ε .

Izlaz: filtrirana slika q .

- 1: $mean_I = f_{mean}(I)$
 $mean_p = f_{mean}(p)$
 $corr_I = f_{mean}(I * I)$
 $corr_{Ip} = f_{mean}(I * p)$
 - 2: $var_I = corr_I - mean_I * mean_I$
 $cov_{Ip} = corr_{Ip} - mean_I * mean_p$
 - 3: $a = cov_{Ip} / (var_I + \varepsilon)$
 $b = mean_p - a * mean_I$
 - 4: $mean_a = f_{mean}(a)$
 $mean_b = f_{mean}(b)$
 - 5: $q = mean_a * I + mean_b$
- $* i ./$ su podkompoentno množenje i deljenje matrica.

- 3.4 Očuvanje ivica
- 3.5 Kernel
- 3.6 Osobine filtra
- 3.7 Izračunavanje i efikasnost
- 3.8 Poboljšanje
- 3.9 Eksperimenti
- 4 Primena

Literatura

- [1] *What is Computer Graphics?*, Cornell University Program of Computer Graphics, 1998.
- [2] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Prentice Hall, 2008.
- [3] Kaiming He, Jian Sun, Xiaoou Tang, *Guided Image Filtering*, ECCV, 2010.
- [4] Kaiming He, Jian Sun, Xiaoou Tang, *Guided Image Filtering*, TPAMI, 2013.
- [5] Kaiming He, Jian Sun *Fast Guided Filter*, arXiv, 2015.