



Digitalna obrada slike

Lekcija VIII

FILTRI



Denoising signala

- Jedna od primjena filtara (vjerovatno najvažnija) je u **denoisingu** (uklanjanju aditivnog šuma koji je zahvatio signal).
- Problem denoisinga se može zapisati na sledeći način (ovdje je taj problem zapisan kao 1D problem):
 - Signal od interesa je **$f(n)$** . Signal je zahvaćen bijelim aditivnim šumom **$v(n)$** , **$x(n)=f(n)+v(n)$** .
 - Cilj je dobiti filtrirani signal **$s(n)$** koji je što je moguće bliži signalu **$f(n)$** ali na osnovu opservacija signala koji je zahvaćen šumom **$x(n)$** .



Denoising signala

- Kao mjeru sličnosti **f(n)** i **s(n)** posmatračemo srednju kvadratnu grešku:

$$MSE = E\{[s(n) - f(n)]^2\} = \frac{1}{N} \sum_{n=1}^N [s(n) - f(n)]^2$$

N ovdje označava ukupnu dužinu signala

- Pod određenim pretpostavkama a pod uslovom da znamo funkciju gustine raspodjele šuma koji zahvata signal možemo da odredimo izlaz iz filtra koji minimizuje srednju kvadratnu grešku.
- Takav filter se naziva **ML** filter (**Maximum likelihood** filter – filter maksimalne sličnosti ili vjerodostojnosti).



ML filter

- Minimizacioni problem za definisanje ML filtra glasi:

$$s(n) = \arg \min_{\mu} \sum_{k=n-N}^{n+N} F(x(k) - \mu)$$

- U ovom slučaju je $2N+1$ širina simetričnog lokalnog susjedstva oko tačke n za koju želimo da odredimo izlaz iz filtra. Opšte pravilo: **širi filter (veće N) bolje uklanja šum ali istovremeno oštećuje signal.**
- **argmin** _{μ} oznaka se čita kao: ono μ za koje se izraz minimizuje.
- **$F()$** se naziva **loss funkcija** (funkcija gubitaka, ovaj naziv potiče iz ekonomije) i za ML filter bi trebala biti jednaka **$F(\xi) = -\log p(\xi)$** gdje je **$p(\xi)$** **funkcija gustine raspodjele šuma.**



Loss funkcija za Gausov šum

- Gausov šum (jedan od najčešćih modela šuma) ima funkciju gustine raspodjele:

$$p(\xi) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\xi^2 / 2\sigma^2}$$

σ^2 je varijansa šuma

- Loss funkcija za ovaj šum je:

$$F(\xi) = \underbrace{\log \sqrt{2\pi\sigma}}_{\text{aditivna}} + \underbrace{|\xi|^2 / 2\sigma^2}_{\text{multiplikativna}}$$

**aditivna i
multiplikativna
konstanta koje
ne utiču na
rezultat
filtriranja**

- Stoga se kao loss funkcija za Gausov šum usvaja $F(\xi) = |\xi|^2$.



MA Filter

- Za sliku izlaz iz filtra za slučaj Gausovske loss funkcije dobija se kao:

$$J(\mu) = \sum_{k=n-N}^{n+N} \sum_{l=m-M}^{m+M} [x(k,l) - \mu]^2$$

μ koje minimizuje ovaj izraz se dobija diferenciranjem $J(\mu)$ po μ i izjednačavanjem sa nulom:

$$\frac{\partial J(\mu)}{\partial \mu} = \sum_{k=n-N}^{n+N} \sum_{l=m-M}^{m+M} -2[x(k,l) - \mu] = 0 \longrightarrow \sum_{k=n-N}^{n+N} \sum_{l=m-M}^{m+M} x(k,l) = \mu \sum_{k=n-N}^{n+N} \sum_{l=m-M}^{m+M} 1$$



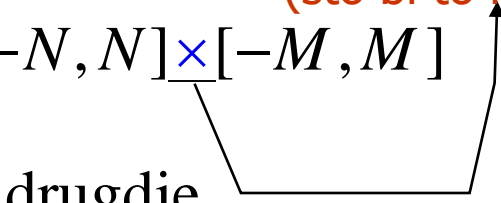
MA filter

$$\begin{aligned} s(n, m) = \mu &= \frac{1}{(2N+1)(2M+1)} \sum_{k=n-N}^{n+N} \sum_{l=m-M}^{m+M} x(k, l) = \\ &= \frac{1}{(2N+1)(2M+1)} \sum_{k=-N}^N \sum_{l=-M}^M x(n+k, m+l) \end{aligned}$$

- Interesantan detalj je da je **MA** filter (**moving average** ili filter sa pokretnom sredinom) konvolucionni filter i da se može dobiti izuzetno jednostavnom konvolucijom slike sa impulsnim odzivom oblika:

$$h(n, m) = \begin{cases} \frac{1}{(2N+1)(2M+1)} & [n, m] = [-N, N] \times [-M, M] \\ 0 & \text{drugdje} \end{cases}$$

Dekartov proizvod
(što bi to moglo biti?)





MA filter – MATLAB realizacija

```
clear
a=imread('Baboon.jpg');
imshow(a)
a=rgb2gray(a);
b3=uint8(conv2(double(a),ones(3)/9,'same'));
b5=uint8(conv2(double(a),ones(5)/25,'same'));
b7=uint8(conv2(double(a),ones(7)/49,'same'));
```

sliku prebacimo u double format

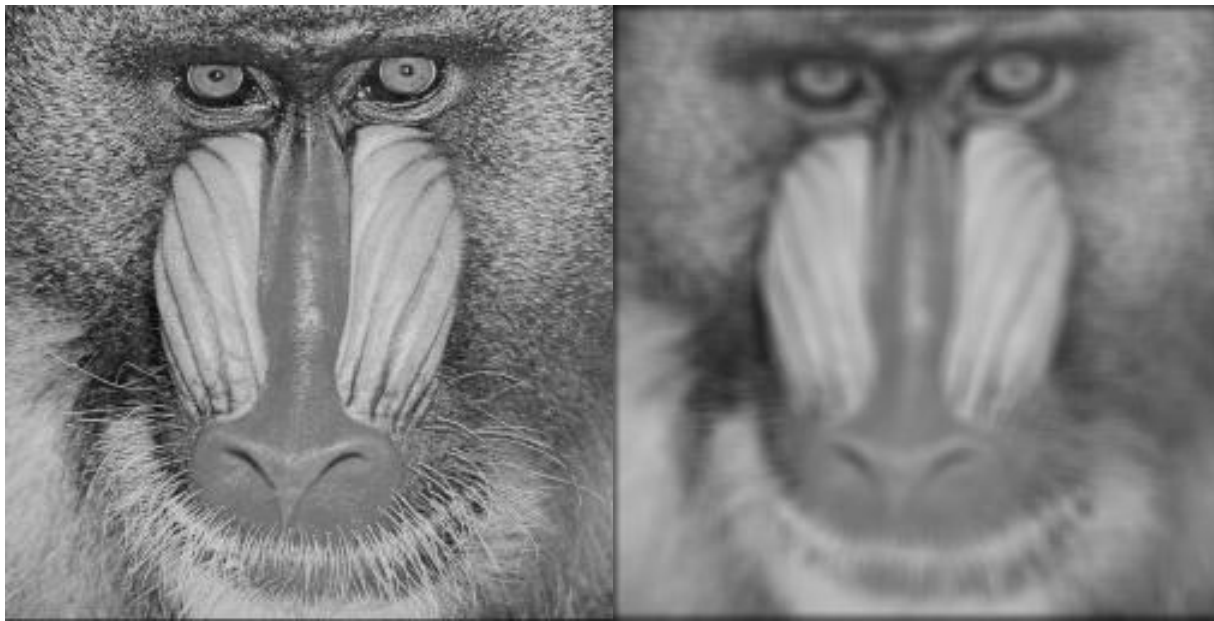
obavimo konvoluciju sa
kvadratnom zonom

vrši zaokruživanja i vraćanje u
format slike.

'same' obezbeđuje da
rezultat konvolucije ima
iste dimenzije kao prvi
argument naredbe conv2.
Kolike bi inače bile
dimenzije?

- Za vježbu odredite frekventni odziv MA filtra. Da li je u pitanju niskopropusni ili visokopropusni filter?

Poređenje filtrirane slike



- Prikazali smo originalnu sliku i sliku filtriranu 7×7 filtrom. Očigledno ovako širok filter zamagljuje sliku kvari ivice, itd. Stoga težimo da kad je slika zahvaćena šumom vršimo filtriranje sa što je moguće užim MA filtrom kako ne bi oštetili informacije sakrivene u ivicama i detaljima slike.

Za vježbu oštetite sliku Gausovim šumom određene varijanse (možete koristiti naredbe `imnoise` ili `randn`) i pogledajte kolika je srednja kvadratna greška prije i poslije MA filtriranja sa filtrima različite širine.



Mane MA filtra

- Već smo vidjeli da MA filter posjeduje jednostavnu poluintuitivnu, formu koja međutim kvari ivice i detalje slike.
- Ovaj filter se izuzetno jednostavno realizuje.
- Slika ne mora biti ekskluzivno oštećena Gausovim šumom već može biti oštećena i šumom koji ima rijetko pojavljivanje ali sa veoma velikom amplitudom.
- Takav šum nazivamo **impulsnim šumom**.
- MA filter je veoma osjetljiv na impulsni šum jer jedan impuls proširi po susjedstvu (istina sa nešto manjom amplitudom).

MA filter za 3×3 susjedstvo

1	1	2	1	3
1	2	1	1	3
1	26	1	1	2
2	1	1	2	1
1	2	2	1	3

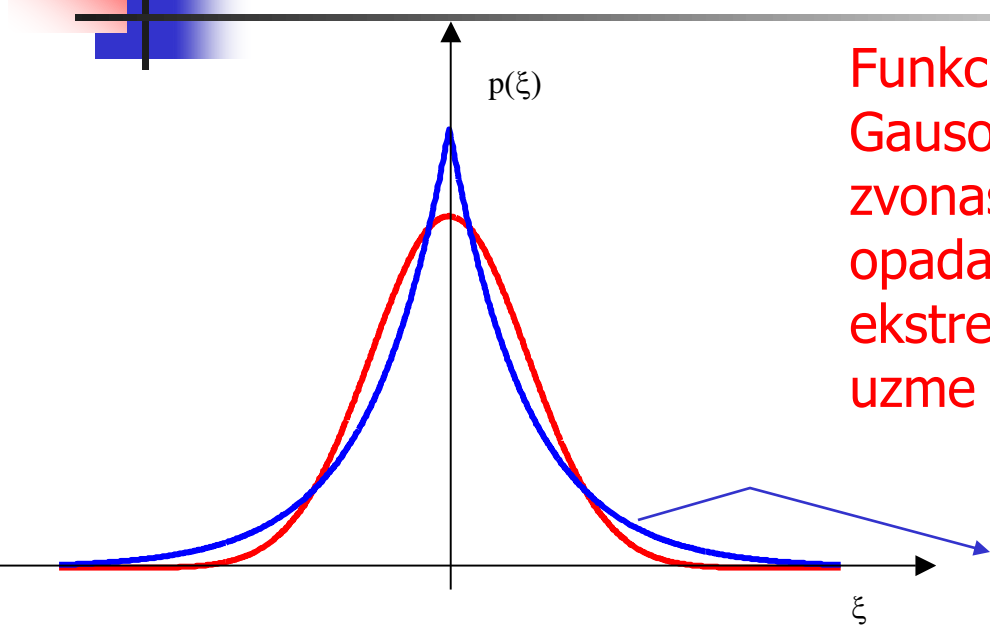
4	4	1.67
4	4	1.44
4.11	4	1.55

Pretpostavimo da je ovo slika i da je filtriramo sa 3×3 MA filtrom.

Crvenom bojom je označen impuls koji očigledno znatno odstupa od vrijednosti iz lokalnog susjedstva. Izvršimo filtriranje unutrašnjeg 3×3 kvadrata.

Svi pikseli koji su susjedi impulsu imaju znatno uvećane vrijednosti odnosno MA filter širi uticaj impulsa po slici (u ovom slučaju zahvaćeni su svi susjedni pikseli).

Poređenje funkcija gust. raspod.



Funkcija gustine raspodjele Gausovog šuma koja je zaobljena zvonasta kriva koja veoma brzo opada ka nuli što znači da je ekstremno mala vjerovatnoća da uzme velike vrijednosti.

Funkcija gustine raspodjele Laplasovog šuma koji je tipičan primjer impulsnih šumova ima "dugačak rep" odnosno sa mjerljivom vjerovatnoćom može uzeti relativno velike vrijednosti.

Funkcija gustine raspodjele Laplasovog šuma je proporcionala **$\exp(-|\xi|)$** pa je odgovarajuća loss funkcija **$F(\xi) = |\xi|$** .



MEDIAN FILTER

- Median filter je ML filter za slučaj signala zahvaćenog Laplasovim šumom.
- Izvođenje polazi od:

$$J(\mu) = \sum_{k=n-N}^{n+N} \sum_{l=m-M}^{m+M} |x(k, l) - \mu|$$

- Pa poslije diferenciranja dobijamo:

$$\frac{\partial J(\mu)}{\partial \mu} = \sum_{k=n-N}^{n+N} \sum_{l=m-M}^{m+M} -\text{sign}[x(k, l) - \mu] = 0$$

Funkcija znaka +1 za
pozitivne vrijednosti, -1 za
negativne i 0 za nulu.



MEDIAN FILTER

- Suma sign-ova se anulira ako je polovina argumenata veća 0 a polovina manja od 0.
- Ovo se može shvatiti i na drugi način. Izvršimo sortiranje $x(k,l)$ za $k \in [n-N, n+N]$ i $l \in [m-M, m+M]$ u neopadajući redosljed. Označimo sortirane vrijednosti sa $\mathbf{x}_{(i)}$ gdje je:
$$\mathbf{x}_{(1)} \leq \mathbf{x}_{(2)} \leq \dots \leq \mathbf{x}_{(i-1)} \leq \mathbf{x}_{(i)} \leq \dots \leq \mathbf{x}_{(2N+1)(2M+1)}$$
- Median je tada vrijednost iz sredine sortirane sekvence odnosno $\mathbf{x}_{((2N+1)(2M+1)+1)/2}$.
- To je vrijednost μ za koju je tačno polovine $x(k,l) - \mu$ manja od nule (ima sign -1) a polovina veća od nule (ima sign 1).



Median parne sekvence

- Uzmimo primjer sekvence sa parnim brojem članova

1 3 11 4 2 5

- Sortirana sekvenca je:

1 2 3 4 5 11

- Kao median se može usvojiti svaka vrijednost između dva srednja elementa u sekvenci (u ovom slučaju između 3 i 4) pa će odgovarajuća minimizujuća funkcija dati rezultat 0.
- Po pravilu se međutim usvaja da je to aritmetička sredina dva srednja člana niza (u ovom slučaju je to 3.5).

Median filter za 3×3 susjedstvo

1	1	2	1	3
1	2	1	1	3
1	26	1	1	2
2	1	1	2	1
1	2	2	1	3

1	1	1
1	1	1
1	1	1

Pretpostavimo da je ovo slika i da je filtriramo sa 3×3 median filtrom. **Crvenom bojom** je označen impuls koji očigledno znatno odstupa od vrijednosti iz lokalnog susjedstva. Izvršimo filtriranje unutrašnjeg 3×3 kvadrata.

Očigledno je uticaj impulsa nestao u svim pikselima filtrirane slike

Za filtriranje ovog piksela sekvenca je imala 5 jedinica, 3 dvojke i jednu trojku.



Osobine median filtra

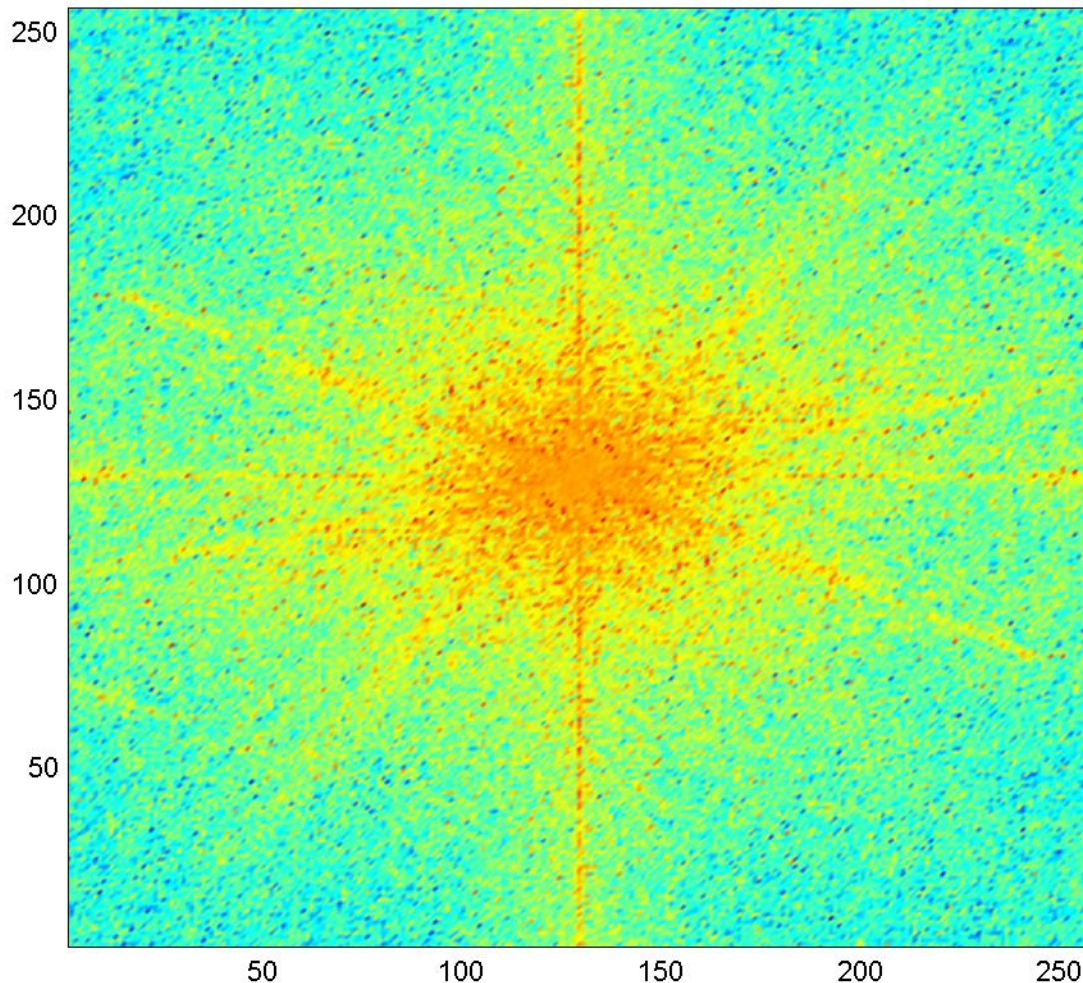
- Median filter je nelinearan!
- To znači da se ni konvolucija ni FFT algoritmi ne mogu koristiti za njegovu evaluaciju već da se za svaku tačku mora vršiti sortiranje susjednih piksela (npr. **quick sort** ili **insertion sort** algoritmom).
- Ovo implicira da je median filter **znatno sporiji** za računanje od filtra sa pokretnom sredinom.
- Postavlja se pitanje kakav je ovaj filter u spektralnom smislu: da li je niskopropusan ili visokopropusan.



Median filter u spektralnom domenu

- Pošto median nije linearan njegov spektralni odziv se samo može procijeniti.
- Mi smo to uradili na sledeći način. Uzeta je slika kojoj je dodat šum a zatim je primjenjen median filter. Sračunali smo 2D DFT ulazne slike i 2D DFT filtrirane slike. Pod spektralnim odzivom smo podrazumjevali odnos 2D DFT izlaza i 2D DFT ulaza. Radi preciznosti proceduru smo ponovili desetak puta za različite šumove i usrednjili ove rezultate koji su prikazani na narednom slajdu.

Median filter u spektr. domenu



Prikazan je logaritam
dobijene aproksimativne
spektralne karakteristike
median filtra. Crvene i
žute boje odgovaraju
velikim vrijednostima a
plave i zelene malim.

Zaključak. Median je
niskopropusan filter
(kao i MA filter) ali ipak
dobro čuva ivice kod
slike.

Za vježbu sami
rekonstruirajte ovaj
eksperimentat.



Dobre osobine mediana

- Median filter odlično uklanja impulsni šum ali je za Gausov šum samo neznatno gori od filtra sa pokretnom sredinom (ovo ćemo pokazati na eksperimentu).
 - Da bi ovo ilustrovali daćemo **asimptotsku formulu** za srednju kvadratnu grešku filtriranja slike kod koje je primijenjena optimizaciona tehnika sa funkcijom gubitaka $F(\xi)$ a funkcija gustine raspodjele šuma je $p(\xi)$ i gdje ne mora da važi $F(\xi) = -\log p(\xi)$:

$$\text{MSE} = E\{|\underbrace{s(n,m)}_{\text{Zašumljena slika}} - \underbrace{f(n,m)}_{\text{Originalna slika bez šuma}}|^2\} \approx \frac{1}{(2N+1)(2M+1)} \frac{\int_{-\infty}^{\infty} (F'(\xi))^2 p(\xi) d\xi}{\left(\int_{-\infty}^{\infty} F''(\xi) p(\xi) d\xi \right)^2}$$



Dobre osobine mediana

- Prethodna relacija je veoma gruba aproksimacija koja se ne izvodi baš na prost način a pretpostavlja da slika u lokalnom susjedstvu ima konstantan osvjetljaj što naravno nije tačno u realnim aplikacijama.
 - Za vježbu obavezno uradite sledeće eksperimente: provjeriti aproksimativnu srednju kvadratnu grešku (Mean Squared Error) za Gausov šum kod mediana i MA filtra; zatim ponovite iste operacije za Laplasov šum kao i za Cauchyjev šum. Cauchy-jev šum je takođe model impulsnog šuma sa f-jom gustine raspodjele: $p(\xi) = a/\pi(\xi^2 + a^2)$.
 - **Kakav zaključak iz ovoga možete da izvedete?**
 - Sugerisani zaključak: median je nešto lošiji od MA filtra za Gausov šum, nešto bolji za Laplasov a mnogo bolji za Cauchyjev.
 - Median radi dobro za gotovo sve impulsne šumove.

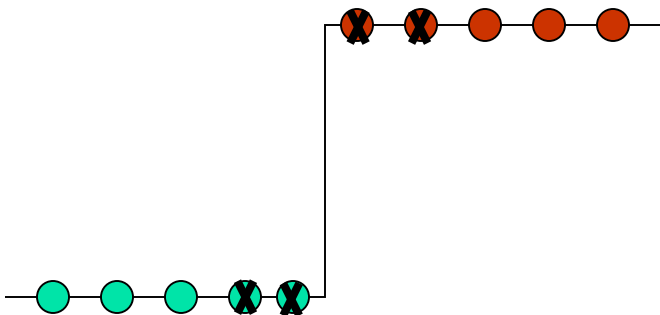


Median – ivice i detalji

- Izuzetno dobra osobina median filtra je činjenica da za razliku od filtra sa pokretnom sredinom čuva ivice objekata u slici (ivice su nagli prelazi sa jednog na drugi osvjetljaj i veoma su važne za ljudsku viziju).
 - Kreirajte sledeći eksperiment. Posmatrajte signal kod kojeg imate K uzastopnih nula praćen sa K uzastopnih jedinica. Primijenite 1D filter sa pokretnom sredinom i median filter sa $2N+1$ odbiraka u okolini ivice i pogledajte što ćete dobiti.
- Jedina bitna mana median filtra uz računsku složenost je to što se mali detalji slike tretiraju kao impulsi i bivaju eliminisani.
- U realnim slikama da bi se izbjeglo uništavanje detalja median se primjenjuje obično u susjedstvu ne većem od 9×9 a često samo 3×3 .



Median i ivice



Na simplifikovanom primjeru pokažimo kako median filter čuva ivice. Neka je susjedstvo širine 3 i neka je zeleni piksel 0 a crveni piksel 1. Na ivici želim da sačuvanom navedene vrijednosti. Za četiri piksela označena znakom **x** smo sračunali izlaz iz median filtra i MA filtra. Kod mediana zadržavamo skokovit prelaz dok kod MA imamo neprijatni blagi prelaz kojim se slika zamagljuje.

Prvi piksel, susjeds. {0, 0, 0}, MA=0, MED=0

Drugi piksel, susjeds. {0, 0, 1}, MA=1/3, MED=0

Treći piksel, susjeds. {0, 1, 1}, MA=2/3, MED=1

Četvrti piksel, susjeds. {1, 1, 1}, MA=1, MED=1



Realizacija median filtra

```
function y=med_filt(x,p)
mM=p(1); mM=round((mM-1)/2);
nN=p(2); nN=round((nN-1)/2);
if(mM<0) mM=0; end
if(nN<0) nN=0; end
if(mM==0&nN==0)
    y=x;
else
    [M,N]=size(x);
    x=double(x);
    y=x;
```

Realizacija je data u obliku MATLAB funkcije: **x** je slika od interesa a **p** je vektor od dva parametra koji predstavljaju dimenzije susjedstva u kojima se vrši filtriranje.

Ako je **p=[9 9]** dobićemo **mM=4** i **nN=4** a ovo dalje znači da ćemo vršiti filtriranje oko svake tačke u intervalu $[m-mM, m+mM] \times [n-nN, n+nN]$. Obezbjedili smo da **mM** i **nN** ne mogu biti negativni.

U ovom slučaju ne treba filtrirati.

Priprema za filtriranje uz pretvaranje slike u double format. **y** je izlazna slika (na ovaj način samo inicijalizovana).



Realizacija median filtra

```
for m=1:M  
    for n=1:N
```

 } → **Prolazimo kroz sve piksele slike.**

```
        z=x(max(1,m-mM):min(M,m+mM),max(1,n-nN):min(N,n+nN));
```

```
        y(m,n)=median(z(:));
```

```
    end
```

```
end
```

```
end
```

z(:) matricu pretvara u vektor a naredba **median** određuje median ove sekvence.

Odsjecamo zonu slike od **m-mM** do **m+mM** i od **n-nN** do **n+nN** ali vodimo računa da zona ne izađe van granica **[1,M]×[1,N]**. Ovo nije zasigurno najbrži način da se ovo odsjecanje ostvari ali je relativno pogodan za nivo objašnjenja na kursu.

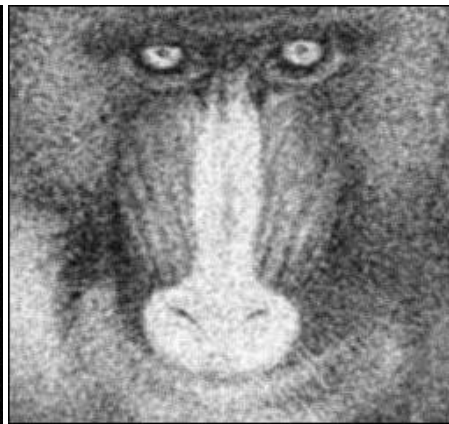
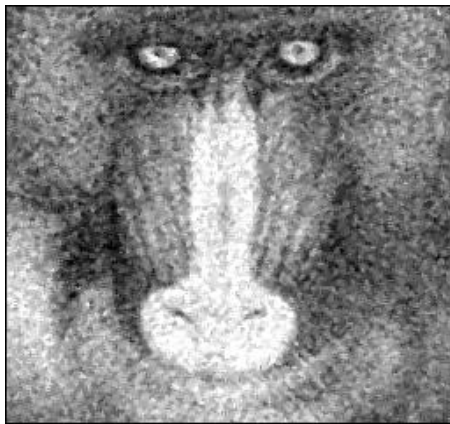
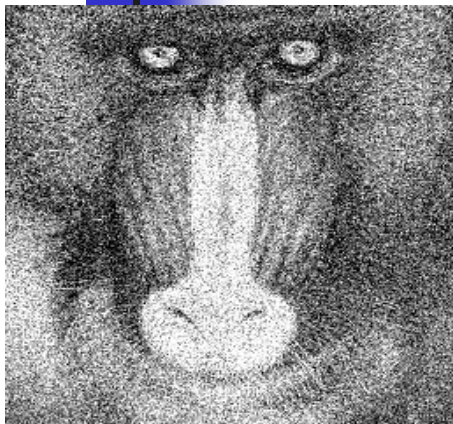
```
y=uint8(y);
```

 → **Dobijeni rezultat se transformiše u format slike.**

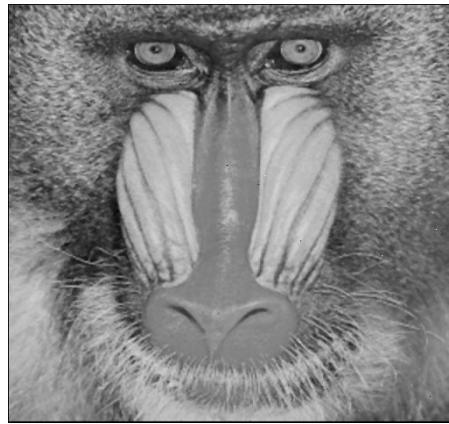
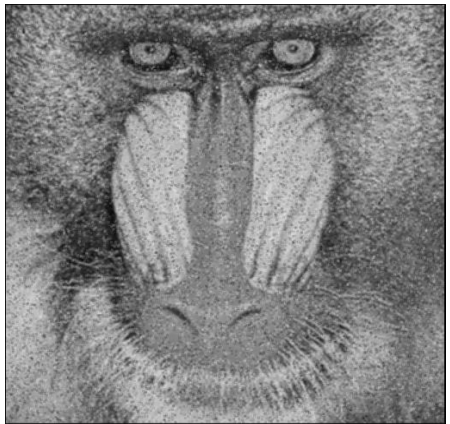
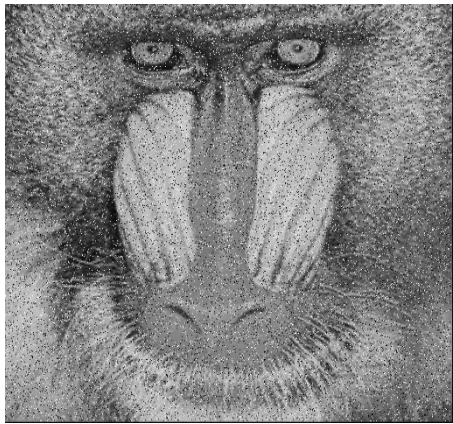
Funkcija je relativno dobra (neprofesionalna) aplikacije. Primjer poziva:

```
B=med_filt(A,[5 5]);
```

Poređenje filtara



Slika zahvaćena Gausovim šumom i filtrirana filtrom sa pokretnom sredinom i median filtrom. Kod filtriranih slika ne uočava se bitna razlika u kvalitetu.



Slika zahvaćena Laplasovim šumom i filtrirana filtrom sa pokretnom sredinom i median filtrom. Median ima znatno bolji kvalitet.

Poređenje kvaliteta se može vršiti samo numerički.



Test šumovi

- Test šumovi se mogu kreirati na više načina. Npr. Gausov šum se može dobiti kada se slika sabere sa $s * \text{randn}(N, M)$ gdje je $N \times M$ dimenzija slike dok je s standardna devijacija šuma.
- Sliku koja je na ovaj način zahvaćena šumom prije prikazivanja treba vratiti u format slike to jest odsjeći (ili zaokružiti) necjelobrojni dio i vratiti u granice od 0 do 255 ako je u pitanju slika koja je prikazana sa 8 bita.
- Alternativna naredba za dodavanje šuma je oblika:
 $B = \text{imnoise}(A, 'tip', \text{parametri});$
- A je originalna slika bez šuma, tip je tip šuma recimo `'salt & pepper'` za tipičan impulsni šum, dok su $parametri$ parametri šuma koji se dodaje (kod `'salt & pepper'` je procenat impulsa).



Kreiranje šumova

- Konsultujte knjigu da bi vidjeli kako se recimo Laplasov šum može kreirati na osnovu šuma sa uniformnom raspodjelom (ovaj šum je dostupan u MATLAB-u preko naredbe **rand**).
- U novijim verzijama MATLAB-a postoji mnoštvo tipova šuma koji su već realizovani. Npr. **poissrnd** daje multiplikativni Poissonov šum koji ima srednju vrijednost i standardnu varijansu koja je jednaka vrijednosti osvjetljaja nezašumljene slike u datoj tački. Ovakav šum je karakteristika slika dobijenih ultrazvučnim snimanjima.
- Druge generatore slučajnog šuma možete dobiti u MATLAB-ovom helpu preko opcije **search** i unesite **Random numbers generator**.
- Za testiranje različitih algoritama u praksi je neophodno znati realizovati šumove a često MATLAB funkcije nijesu dostupne ili nijesu realizovane na mašinama sa kojima treba raditi (recimo medicinski uređaji) pa ih treba znati realizovati od početka.

Numeričke ocjene kvaliteta

- Najpoznatije numerička ocjena kvaliteta filtriranja (kao i nekih drugih algoritama u obradi slike) je SNR (Signal to Noise Ratio).

- SNR se definiše kao:

$$SNR = 10 \log_{10} \frac{\frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N f^2(n, m)}{\frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N [\tilde{f}(n, m) - f(n, m)]^2} =$$

Izražava se u
decibelima dB.

Srednja kvadratna
vrijednost slike
("srednja snaga
slike").

Originalna slika.

Filtrirana slika.

$= 10 \log_{10} \frac{\frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N f^2(n, m)}{\text{MSE}}$ Srednja kvadratna greške filtriranja (Mean Squared Error). Takođe, može biti procjena kvaliteta.

Numeričke ocjene kvaliteta

- Danas najpopularnija mjera kvaliteta je odnos Pseudo signal – šum (Pseudo Signal to Noise Ratio – PSNR) koja se definiše kao:

$$PSNR = 10 \log_{10} \frac{\mathbf{f}_{\max}^2}{\frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N [\tilde{f}(n, m) - f(n, m)]^2} = 10 \log_{10} \frac{\mathbf{f}_{\max}^2}{MSE}$$

Maksimalna vrijednost signal (maksimalni osvjetljaj).

- Isključujući neke relativno nerealne situacija (npr. tamna slika ima samo jednu svijetlu tačku) PSNR odnos je veoma dobro procjena kvaliteta filtriranja slike.



Numeričke ocjene kvaliteta

- Zna se da za $\text{PSNR} > 60\text{dB}$ razlike dvije slike se obično ne mogu uočiti čak ni u uslovima direktnog poređenja.
- Za $\text{PSNR} > 40\text{dB}$ (uglavnom i za $\text{PSNR} > 35\text{dB}$) razlike se mogu uočiti samo pažljivim upoređivanjem originala i filtrirane slike.
- Za $15\text{dB} < \text{PSNR} < 35\text{dB}$ slika se može prepoznati ali je lošijeg kvaliteta (pri donjoj granici jako lošeg a pri gornjoj relativno dobrog).
- Za $\text{PSNR} < 15\text{dB}$ (ili $\text{PSNR} < 20\text{dB}$) slika je neupotrebljiva.
- Ovakve granice su i razlog zbog kojeg se PSNR najčešće i koristi kao numerička ocjena kvaliteta.



Ostale ocjene kvaliteta

- **Numeričke:** Jedna relativno često korišćena ocjena kvaliteta je maksimalna apsolutna vrijednost greške; Pored uvedenih danas je sve popularnija mjera **ISNR** (**I**mprovement **p**oboljšanje **S**NR):

$$ISNR = 10 \log_{10} \frac{\frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N [x(n, m) - f(n, m)]^2}{\frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N [\tilde{f}(n, m) - f(n, m)]^2}$$

Ovim numeričke
ocjene kvaliteta
nijesu i
iscrpljene!

Kao i sve mjere koje se računaju preko logaritma i ova se mjera izražava u dB. Predstavlja odnos MSE za ulazni signal i za izlazni signal i pokazuje za koliko dB smo popravili SNR našim filtriranjem. Ako je ISNR=0dB nijesmo popravili ništa.



Ostale ocjene kvaliteta

- **Nenumeričke.** Sve numeričke mjere kvaliteta u praksi imaju neke mane. Npr. PSNR koji je kod većine realnih slika veoma dobar ne daje realne rezultate ako je slika veoma tamna sa svega nekoliko svijetlih tačaka. Stoga se još uvijek koriste i mjere kvaliteta koje se ne izražavaju strogo formulom već recimo ocjenom “vrlo dobro”. Najčešće se i nenumeričke mjere na neki način kvantifikuju pa se većem broju korisnika da ocjeni rezultat filtriranja sa opisnim ocjenama a te se opisne ocjene kvantifikuju numeričkim (npr. od 1 do 5) i usrednje i recimo se kaže prosječna ocjena filtriranja je **3.62**.
- Možda izgleda čudno ali ovakva “mjerjenja” nijesu rijetka u praksi i najčešće je problem samo u pronalaženju “dobrovoljaca” koji će vršiti ocjenjivanje.



Varijante median filtra

- U praksi su realizovane brojne varijante median filtra.
- Pored filtra sa pravougaonom oblašću susjedstva mogu se koristiti i filtri sa drugim oblicima susjedstva: obliku plusa, romba, kruga itd. Pokušajte da modifikujete funkciju koja vam je predložena da bi radila za neki od navedenih odnosno za proizvoljan oblik susjedstva.
- Do prije nekoliko godina značajna tema je bilo pojednostavljivanje računanja median filtra sa stanovišta računske kompleksnosti (sortiranje je složenije od konvolucije) odnosno u pogledu memorijskih zahtjeva (pored polazne slike, memorišemo i sliku rezultat). Mi ćemo ovdje definisati dva “pojednostavljena” median filtra:
 - Separabilni median filter (smanjuje računsku složenost)
 - Rekurzivni median filter (umanjuje memorijske zahtjeve).



Separabilni median filter

- Kod separabilne varijante median filter se primjenjuje za svaku vrstu (ili kolonu) kao 1D filter pa se zatim primjenjuje na svaku kolonu (odnosno vrstu) kao 1D filter. Dajmo realizaciju ove funkcije samo u dijelu dvostruke **for** petlje (ostatak je sličan kao kod mediana):

```
function y=sep_med_filt(x,p)
```

```
%%%%%%%%Naredbe kao kod median filtra
```

```
for m=1:M
```

```
    for n=1:N
```

```
        t(m,n)=median(x(m,max(1,n-nN):min(N,n+nN)));
```

```
    end
```

```
end
```

```
for m=1:M
```

```
    for n=1:N
```

```
        y(m,n)=median(t(max(1,m-mM):min(M,m+mM),n));
```

```
    end
```

```
end
```

```
y=uint8(y);
```

**Filtriranje po
vrstama.**

**Filtriranje po
kolonama slike
međurezultata.**

Procjena složenosti sep. mediana

- Kod median filtra se za svaku tačku u slici vrši sortiranje niza dimenzija $(2K+1) \times (2K+1)$.
- Najbrži algoritmi sortiranja (recimo quick sort) pod dobrim uslovima zahtjevaju $N \log_2 N$ poređenja za sortiranje odnosno u našem slučaju je to $2(2K+1)^2 \log_2(2K+1)$.
- Kod separabilnog median filtra vršimo sortiranje sekvence od $2K+1$ duž jedne koordinate pa zatim vršimo 1 sortiranje duž druge koordinate pa je složenost: $(2K+2)(2K+1) \log_2(2K+1)$ pa ovaj filter ima procjenjenu manju složenost za $(2K+1)/(K+1)$ puta u odnosu na median filter.
- Napomenimo da su sve dobijene relacije aproksimacije i da nijesu previše tačne za malo K ali daju dobar osjećaj za složenost.
- Napomenimo a to i provjerite da su rezultati dobijeni separabilnim median filtrom slični onima dobijenim kod median filtra.



Rekurzivni median filter

- Druga (nekada veoma česta) varijanta median filtra je **rekurzivni median filter**.
- Primjenljiv je kod sistema koji su oskudijevali sa memorijom.
- Radi na sledeći način. Pretpostavimo da počinjemo filtriranje slike od pozicije (n,m) . Primjenimo median filter i dobijeni rezultat smjestimo u originalnu sliku na poziciju (n,m) . Zatim vršimo filtriranje na poziciji $(n+1,m)$. Filtriranje sada uključuje i već filtrirani piksel (n,m) !!!
- Dakle, kod rekurzivnog filtriranja ne formiramo dvije slike – početnu sliku i sliku rezultat već sve operacije obavljamo na jednoj slici i na taj način štedimo memoriju.

Realizacija rekurzivnog mediana

%Početak kao kod mediana

[M,N]=size(x);

x=double(x);

%%%y=x;!!!!

Ovog koraka više nema
(stavljen pod komentarom).

for m=1:M

for n=1:N

z=x(max(1,m-mM):min(M,m+mM),max(1,n-nN):min(N,n+nN));

x(m,n)=median(z(:));

end

end

Rezultat filtriranja se stavlja
u početnu sliku!



Mane rekurzivnog mediana

- Rekurzivni median ima osnovnu prednost u smanjivanju memorijskih zahtjeva.
- Danas memorija osim u veoma specifičnim uređajima nije skupa ni nedostupna tako da se smanjuje potreba za ovakvom uštedom.
- Rekurzivni median, međutim, može da poništi više impulsa nego median filter (filtriranje se obavlja nad pikselima koji su već filtrirani).
- Međutim, rekurzivni median filter sa relativno širokim susjedstvom u kojem vrši filtriranje razmazuje sliku u pravcu u kojem se vrši filtriranje.
- Za inženjere ovo je veoma loša osobina ali je ponekad koriste “umjetnički” računarski alati za postizanje specifičnih efekata.



Mixed noise slučaj

- Već smo uveli pojam ML filtara i rekli da se oni dizajniraju prema šumu koji se očekuje u slici.
- Rekli smo da je MA filter dobar za Gausov šum dok da se za Laplasov treba koristiti median filter.
- Ako ne znamo koji je tip šuma ali imamo saznanja da je impulsne prirode dobro je započeti sa median filtrom jer on daje dobre rezultate i za druge klase impulsnih šumova.
- Postavlja se pitanje jedne veoma praktične situacije a to je da imamo šum koji je uglavnom Gausov a ponekad se pojavljuju impulsi. Ovakav šum se naziva mješovitim ili **mixed noise**.
- Što u toj prilici raditi?



L - filtri

- Jedna moguća ideja kod mixed šumova je primjeniti median filter koji će vjerovatno dati dosta solidne rezultate.
- Ipak ovi rezultati će dosta odstupati od najboljih koji bi se mogli postići ML filtrom za dati šum.
- Obično se ML filter za miješane šumove ne može izraziti u zatvorenom obliku već se vrši njegovo iterativno izračunavanje.
- Umjesto dizajna ML filtra koji je obično i nemoguć precizno koriste se **L-filtri** koji **kombinuju osobine MA i median filtera**.



L-filtri - definicija

- L-filtar je skraćeno od nečega što se može definisati kao **linearna kombinacija statistika reda**.
- Na primjer pretpostavimo da imamo sliku i da posmatramo njen piksel (n,m) i lokalno susjedstvo dimenzija $(2N+1) \times (2M+1)$.
- Kod median filtra smo vršili sortiranje ovog susjedstva od najmanje do najveće vrijednosti:
$$X_{(1)} \quad X_{(2)} \quad X_{(3)} \quad \dots \quad X_{[(2N+1)(2M+1)+1]/2} \quad \dots \quad X_{[(2N+1)(2M+1)-1]} \quad X_{(2N+1)(2M+1)}$$
- Vrijednosti $\mathbf{x_{(i)}}$ su iz lokalnog susjedstva sortirane tako da je $\mathbf{x_{(i)} \leq x_{(i+1)}}$. Tako je sredina niza odnosno median:
$$\mathbf{X_{[(2N+1)(2M+1)+1]/2}}$$



L-filtri - Definicija

- L-filtar se može definisati:

$$y(n, m) = \sum_{i=1}^{(2N+1)(2M+1)} a_i x_{(i)}^{(n, m)}$$

Koeficijenti L-filtra. ←

→ **Sortirane vrijednosti van lokalnog susjedstva oko tačke (n,m).**

- Karakteristike koeficijenata a_i su uobičajeno selektovane da zadovoljavaju:

$$\sum_{i=1}^{(2N+1)(2M+1)} a_i = 1$$

Uslov koji omogućuje da je energija izlaznog signala jednaka energiji (osvjetljaju) izlaznog signala.

$$a_i = a_{(2N+1)(2M+1)+1-i}$$

Uslov nepristrasnosti koji forsira da se na isti način tretiraju vrijednosti koje su na istoj poziciji u odnosu na median na strani većih i manjih osvjetljaja.



L-filtri – Specijalni slučajevi

- Dva do sada uvedena filtra: MA i median filter su specijalni slučajevi L-filtara:
 - MA filter slijedi za $a_i = 1/(2N+1)(2M+1)$ za svako i .
 - Kako je pretpostavljeno da je broj odbiraka u lokalnom susjedstvu neparan to median filter slijedi $a_{[(2N+1)(2M+1)+1]/2} = 1$ i $a_i = 0$ za $i \neq [(2N+1)(2M+1)+1]/2$.
 - Dva ponekad korišćena specijalna slučaja L-filtara su MAX filter koji daje maksimum osvjetljaja u lokalnom susjedstvu: $a_{(2N+1)(2M+1)} = 1$ i $a_i = 0$ za $i \neq (2N+1)(2M+1)$; kao i MIN filter koji daje minimum osvjetljaja u lokalnom susjedstvu tačke: $a_1 = 1$ i $a_i = 0$ za $i \neq 1$.



L-filtri – Praktične forme

- Specijalni slučajevi ipak nijesu ono zbog čega su L-filtri predloženi.
- Najčešća praktično korišćena forma L-filtara je **α -trimovana sredina** koja iz sortirane sekvence bira nekoliko vrijednosti oko mediana simetrično sa obje strane mediana i vrši usrednjavanje tih vrijednosti. Formalno se koeficijenti a_i kod α -trimovane sredine mogu definisati kao:

$$a_i = \begin{cases} \frac{1}{2\alpha K + 1} & i \in [(K+1)/2 - \alpha K, (K+1)/2 + \alpha K] \\ 0 & \text{drugdje} \end{cases}$$

$K = (2N+1)(2M+1)$
 $\alpha \in [0, 0.5]$



L-filtri praktične forme

- Za $\alpha=0$ α -trimovana sredina se svodi na median dok se za $\alpha=0.5$ svodi na MA filter. Praktično su ipak interesantnije forme za α između ove dvije ekstremne vrijednosti $\alpha \in [0, 0.5]$ pomoću kojih se može izbjeći uticaj impulsnog šuma ali u isto vrijeme bolji rezultati nego kod mediana za Gausov šum.
- Često korišćeni slučaj L-filtra je varijanta gdje se za lokalno susjedstvo sračuna median i za dobijanje izlaza iz filtra usrednje oni pikseli iz susjedstva kod kojih je apsolutna vrijednost razlike osvijetljaja posmatranog piksela i mediana manja od nekog parametra Δ koji se unaprijed usvaja.



Za samostalni rad

- Kreirajte funkciju koja računa pokretnu sredinu i median za nepravougaono susjedstvo.
- Kreirajte funkciju koja realizuje α -trimovani filter gdje je α ulazni argument funkcije.
 - **Napomena.** Realizacija je slična medianu. U petlji gdje se inače računao median vrši se sortiranje niza naredbom **sort** i usrednjavaju oni članovi sortiranog niza oko mediana čija je udaljenost manja od veličine koja je određena sa α i dimenzijama lokalnog susjedstva.
- Realizujte varijante L-filtara opisane u lekciji.
- Realizujte myriad filter u lokalnom susjedstvu datih granica. Myriad filter se dobija kao ML filter za Cauchyjev šum čija je funkcija gustine raspodjele proporcionalna $s/(\xi^2 + K^2)$ gdje je K linearizacioni parametar.
 - **Napomena.** Nije moguće dobiti izlaz iz ovog filtra u zatvorenom obliku pa se za njegovu realizaciju mora koristiti iterativni algoritam. Početna iteracija može biti izlaz iz MA ili median filtra. Student koji realizuje ovaj filter može biti posebno nagrađen.



Za samostalni rad

- Realna slika je zahvaćena Gausovim šumom. Da li se može dogoditi da srednja kvadratna greška dobijena na izlazu iz MA filtra bude veća od srednje kvadratne greške dobijene na izlazu iz median filtra?
Obrazložiti! Ponoviti isti misaoni eksperiment ako je slika zahvaćena sa Laplasovim šumom i filtrirana sa ova dva filtra. Može li u ovom slučaju ML filter (median filter) biti gori od MA filtra i ako može zbog čega?
- Realizujte separabilni i rekurzivni median filter.
- Uporedite separabilni median, median, α -trimovani filter i MA filter za filtriranje salt&pepper šuma, Gausovog šuma, Laplasovog šuma.
 - **Napomena.** Eksperimenti se obavljaju na sledeći način: posmatra se više slika, posmatra se više nivoa šuma, za svaki nivo šuma ponavlja se eksperiment sa više realizacija šumnog procesa (Monte-Karlo simulacija), posmatra se različita veličina lokalnog susjedstva, različiti parametri šuma i tek na osnovu ovakve statistike donosi odluka.



Za samostalni rad

- Provjerite spektralne karakteristike median filtra.
- Odrediti spektralnu karakteristiku MA filtra (MA filter u frekventnom domenu).
- Da li L-filtri definisani u lekciji zadovoljavaju dva uobičajena uslova za L-filtre?
- Posmatrajte šum sa uniformnom raspodjelom na intervalu $-\Delta$ do Δ . Definirati ML filter za ovaj šum. Da li je ovaj filter pripadnik klase L-filtara i sa kojim koeficijentima? Na osnovu koja dva uvedena specijalna slučaja L-filtara se može definisati ovaj filter?
- Šum uzima sa vjerovatnoćom $(1-p)$ Gausovu raspodjelu sa srednjom vrijednošću 0 i varijansom σ^2 (ovo se ponekad zapisuje šum $N(0, \sigma^2)$ gdje N potiče od normalne raspodjele) i sa vjerovatnoćom p (p je relativno mala vjerovatnoća u praksi obično manja od 10%) Gausovu raspodjelu sa srednjom vrijednošću 0 i varijansom $K^2\sigma^2$ gdje je K veće od 1 a po pravilu 3 ili 5. Odrediti ML filter za ovaj šum. Da li se može realizovati u zatvorenom obliku? Intuitivno odgovorite da li je za ovaj šum bolji MA ili median filter?



Za samostalni rad

- Realizovati ML filter za “miješani Gausov šum” definisan na prethodnom slajdu (vjerovatno iterativnom procedurom). Uporediti dati filter sa MA i medianom. Da li se rezultati poklapaju sa očekivanim i ako ne poklapaju zbog čega?
- Median filter “nasumično” uzima jedan piksel iz lokalnog susjedstva. Mi bi željeli da zadržimo centralni piksel kao najbolji ako nije zahvaćen impulsnim šumom. Postoji varijanta težinskog median filtra kod koje se centralni piksel ponavlja nekoliko puta i median se traži od sekvence koju predstavljaju osvjetljaji piksela iz lokalnog susjedstva i više puta ponavljeni centralni piksel. Realizovati ovaj filter za različite brojeve ponavljanja centralnog piksela. Kakve karakteristike očekujete u filtriranju u zavisnosti od broja ponavljenih piksela i kakve dobijate?
- Rayleighev šum je kvadratni korjen zbira kvadrata dva nezavisna Gausova šuma. Često se koristi u praksi. Da li median, MA i drugi uvedeni filteri mogu da filtriraju ovaj šum? Koja je ML estimacija ovog šuma? Kako modifikovati L-filtre ili median filter da daje neke rezultate u ovom slučaju?



Za samostalni rad

- Kreirati funkcije koje na osnovu zadate polazne (bez šuma) i odredišne slike računaju: PSNR, SNR i druge mjere kvaliteta.
- Da li se ISNR može sračunati sa ova dva parametra? Ako može realizovati funkciju koja računa ISNR a ako ne može dodati još parametara i realizovati funkciju za ISNR.
- Da li se može dogoditi da ISNR bude negativan i što znači negativna vrijednost ISNR-a?