



Sveučilište u Zagrebu  
Fakultet elektrotehnike i računarstva  
Zavod za elektroničke sustave i obradbu informacija

# 2D DFT, poboljšanje i obnavljanje slike, ekstrakcija značajki i segmentacija

## Obrada informacija (FER-2) - Laboratorijska vježba 3.

### 1. Uvod

Laboratorijske vježbe iz Obrade informacija se izvode na računalu, a koristi se programski sustav MATLAB. Osim što posjeduje mogućnost izvođenja raznih jednostavnih i izuzetno složenih matematičkih operacija, MATLAB ima i modul Simulink koji je zamišljen kao alat za brzo i jednostavno simuliranje raznih sustava. S MATLAB-om i Simulink-om ste se upoznali na [LiV-u MATLAB](#).

### 2. Priprema

Prisjetite se kako se pišu MATLAB m-funkcije te kako se koristi Simulink (vježbe [2.](#) i [4.](#) sa [LiV-a MATLAB](#)). Kao podsjetnik vam osim materijala korištenih na [LiV-u MATLAB](#) može poslužiti i priručnik [Kratke upute za korištenje MATLAB-a](#) koji je dostupan na [stranicama predmeta](#).

Na vježbama će se koristiti slike iz USC-SIPI baza slika. Te se slike uobičajeno koriste za evaluaciju algoritama digitalne obradbe slike. Bazu može pregledati na adresi <http://sipi.usc.edu/services/database/>. Slike iz baze su pohranjene u TIFF formatu pa prije dolaska na vježbe ponovite kako se TIFF slike učitavaju u MATLAB radnu okolinu.

Prije dolaska na laboratorijske vježbe korisno je pročitati sve zadatke koje ćemo raditi. Primijetite da uz neke od zadatake piše **(PRIPREMA)**. To su zadaci koje morate riješiti prije dolaska na vježbu. Rješenja uredno napišite **rukom** na papiru. U zaglavlje svakog papira s pripremom napišete vaše ime i prezime te JMBAG.

Pripremni zadaci su: 3.1-1a, 3.1-2a, 3.2-3a

### 3. Rad u laboratoriju

Prije početka rada uključite dnevnik (naredba diary). Bez obzira na dnevnik također vam preporučamo da rješenje svakog zadatka spremite kao MATLAB m-skriptu ili Simulink model čije ime je redni broj zadatka.

Bodove iz laboratorija stječete tijekom vježbi. Treća laboratorijska vježba nosi četiri boda. Točno napisana priprema vam donosi jedan bod. U zadnjem satu laboratorijskih vježbi (ili prije ako ste prije gotovi s vježbom) dežurni asistent će obići studente te svakog od vas zamoliti da demonstrira rješenja zadanih zadataka. Ako

uspješno demonstrirate zadatke<sup>1</sup> (jedan bod) i ako točno odgovorite na postavljena pitanja vezana uz vježbu (još dva boda) dobivate preostala tri boda.

Svaki student može samo jednom demonstrirati rješenje zadatka i odgovarati. Dakle ako krivo odgovorite ili ako je demonstrirano rješenje zadatka krivo ne dobivate bodove. NEMA popravkih zadataka!

Dio zadataka označen je kao (PRIPREMA), dok je dio označen kao (ZA ONE KOJI ŽELE ZNATI VIŠE). Zadatke za pripremu ste riješili prije vježbi, dok zadatke za one koji žele znati više ne trebate rješavati. Predlažemo da ih preskočite te se vratite na njih ako obavezne zadatke završite prije predviđenog vremena.

## 3.1. Transformacije slike

Prvi dio prve laboratorijske vježbe se bavi transformacijama slike. Kao i za ostale dijelove vježbe sve predviđene operacije se izvršavaju na **sivim slikama**, dakle ne na slikama u boji. Dosta slika iz USC-SIPI baze slika su slike u boji te ih je prije korištenja potrebno pretvoriti u sive slike. U MATLAB-u se isto radi naredbom `rgb2gray` (za slike s paletom boja se koristi naredba `ind2gray`):

```
» [img,map] = imread('4.2.04.tiff'); % učitamo sliku
» img = double(rgb2gray(img));      % pretvaramo je u sivu sliku
» whos
```

Name	Size	Bytes	Class	Attributes
img	512x512	2097152	double	
map	0x0	0	double	

### 25 minuta **Zadatak 3.1-1 Dvodimenzionalna diskretna Fourierova transformacija**

Diskretna Fourierova transformacija (DFT) najčešće se računa korištenjem algoritma za brzu Fourierovu transformaciju (FFT). Kako je 2D DFT separabilna transformacija, obično računamo prvo 1D FFT po redcima te 1D FFT po stupcima. Stoga je složenost 2D FFT algoritma  $O(N^2 \log_2 N)$ . U MATLAB-u 2D FFT računamo korištenjem funkcije `fft2` na slijedeći način:

```
» [img,map] = imread('moon.tif'); % učitamo sliku
» imgFT = fft2(double(img));      % primijenimo MATLAB fft2 funkciju
» imgFT = fftshift(imgFT);        % centriramo spektar
» whos
```

Name	Size	Bytes	Class
img	537x358	192246	uint8 array
imgFT	537x358	3075936	double array (complex)
map	0x0	0	double array

Grand total is 384492 elements using 3268182 bytes

MATLAB funkcija `fft2` računa spektar čiji centar nije u sredini dobivene matrice te je na rezultat potrebno primijeniti funkciju `fftshift`. Funkcija `fftshift` pomiče spektar tako da se istosmjerna komponenta nađe u središtu slike što je uobičajeni prikaz spektra.

---

<sup>1</sup> Demonstracija će vam biti jednostavnija ako ste napisali m-skriptu - jednostavno pozovete tu skriptu. Ako niste pisali m-skripte iskoristite dnevnik. Dnevnik otvorite u ugrađenom editoru i odaberete dio koji odgovara rješenju zadatka kojeg tražimo. Pritiskom na desnu tipku miša dobivate izbornik iz kojega je potrebno odabrati Evaluate Selection. MATLAB tada izvršava označeni dio te crta sve slike i prikazuje sve rezultate.

Spektar koji smo dobili i koji se nalazi u varijabli `imgFT` je kompleksan. Kod prikazivanja spektra moramo prikazati i amplitudu i fazu. Budući da su razlike u iznosu koeficijenata velike te dosežu i do nekoliko redova veličine, amplitudu uobičajeno prikazujemo u decibelima (logaritmiramo je radi kompresije dinamičkog opsega):

```
» amplituda = 20 * log10(abs(imgFT)); % računamo amplitudu u dB
» faza = angle(imgFT); % računamo fazu
» figure, imagesc(amplituda)
» figure, imagesc(faza)
```

- a) (PRIPREMA) Napišite izraz za 2D DFT transformaciju te pokažite da je transformacija separabilna.
- b) Iz USC-SIPI baze slika odaberite jednu sliku. Korištenjem naredbe `fft2` izračunajte 2D DFT transformaciju odabrane slike te nacrtajte njen amplitudni i fazni spektar. Korištenjem naredbe `fft` izračunajte 1D DFT transformaciju po stupcima pa po retcima. Opet nacrtajte amplitudni i fazni spektar. Jesu li rezultati isti?

```
» [img,map] = imread('moon.tif'); % učitamo sliku
» stupci = fft(double(img)); % računamo transformaciju po stupcima
» retci = fft(stupci, [], 2); % računamo transformaciju po retcima
» amplituda = abs(retci); % računamo amplitudu itd.
```

- c) Iz USC-SIPI baze slika odaberite jednu fotografiju i jednu sliku tekture (npr. Brodatz D9 ili D112). Prikažite amplitudu i fazu spektra. U kojem se dijelu spektra nalazi većina energije?
- d) Iz USC-SIPI baze slika odaberite dvije različite slike istih dimenzija. Označimo spektar prve slike s  $X(\omega_1, \omega_2)$  i spektar druge slike s  $Y(\omega_1, \omega_2)$ . Izračunajte i nacrtajte slike koje se dobivaju primjenom inverzne 2D DFT transformacije na spektre koji su sastavljeni od amplitudnog spektra jedne slike i faznog spektra druge slike, odnosno izračunajte

$$\text{IDFT} \left[ |X(\omega_1, \omega_2)| e^{j \arg Y(\omega_1, \omega_2)} \right]$$

i

$$\text{IDFT} \left[ |Y(\omega_1, \omega_2)| e^{j \arg X(\omega_1, \omega_2)} \right].$$

Komentirajte rezultate! Je li dominantan utjecaj faznog ili amplitudnog spektra? Objasnite!

```
» lena = rgb2gray(imread('4.2.04.tiff')); % učitavamo slike
» tiffany = rgb2gray(imread('4.2.02.tiff'));
» LENA = fft2(double(lena)); % računamo 2D DFT
» TIFFANY = fft2(double(tiffany));
» prva = ifft2( abs(LENA) .* exp(j*angle(TIFFANY)) );
» max(imag(prva(:))) % primijetite da zbog numeričke pogreške
% postoji mala imaginarna komponenta

ans =

3.7593e-014

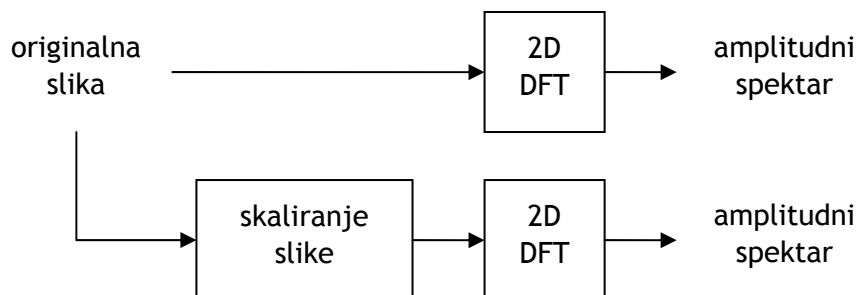
» imagesc(real(prva)) % uklanjamo imaginarni dio i crtamo sliku
» colormap(gray) % je li na slici Lena ili Tiffany? OBJASNITE!
```

## 20 minuta Zadatak 3.1-2 Svojstva 2D DFT transformacije

U ovom zadatku pozabaviti ćemo se raznim svojstvima 2D DFT transformacije.

- a) (PRIPREMA) Ponovite i navedite svojstva 2D DFT transformacije (posebno prostorno skaliranje i rotaciju).

- b) Iz USC-SIPI baze slika odaberite jednu fotografiju. Izračunajte i prikažite amplitudne spektre originalne i skalirane slike. U čemu je razlika?

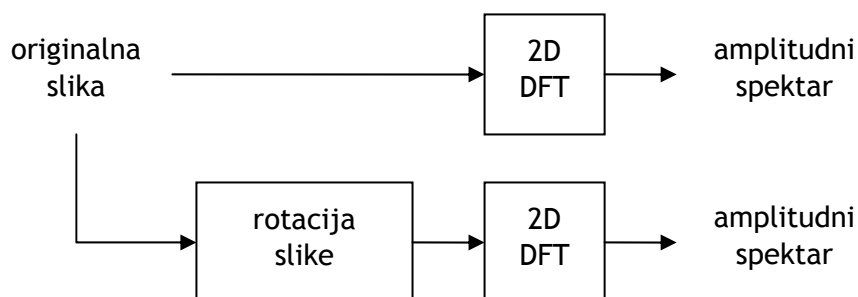


U MATLAB-u slici mijenjate dimenzije naredbom `imresize`:

```

» [img,map] = imread('moon.tif'); % učitamo sliku
» imgV = imresize(img,2);         % povećamo sliku dva puta
» imgM = imresize(img,.25);       % smanjujemo sliku četiri puta
  
```

- c) Iz USC-SIPI baze slika odaberite jednu fotografiju. Izračunajte i prikažite amplitudne spektre originalne i rotirane slike. U čemu je razlika?



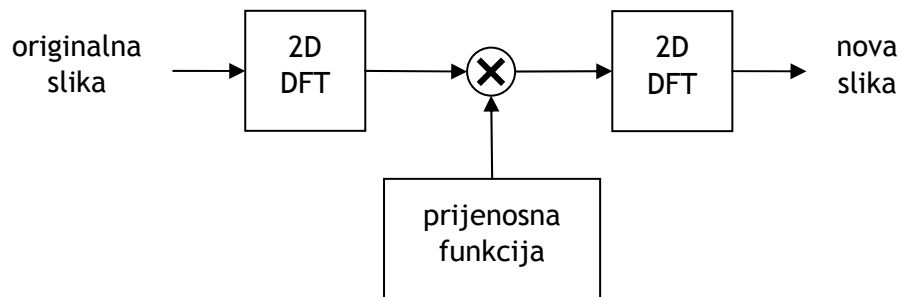
U MATLAB-u sliku rotiramo pomoću naredbe `imrotate`:

```

» [img,map] = imread('moon.tif'); % učitamo sliku
» imgR=imrotate(img,66,'crop');   % rotiramo sliku oko centra za kut 66°
  
```

### 25 minuta **Zadatak 3.1-3 Crikularna konvolucija i filtriranje u domeni transformacije**

Pod filtriranjem slike obično podrazumijevamo propuštanje slike kroz neki LSI sustav. Takvu operaciju možemo računati korištenjem 2D konvolucije ili u domeni transformacije. Filtriranje u domeni transformacije izvodi se množenjem Fourierove transformacije slike s Fourierovom transformacijom impulsnog odziva filtra:



Prijenosna funkcija LSI sustava je transformacija impulsnog odziva sustava. Za dobivanje raznih impulsnih odziva (a time i prijenosnih funkcija) možete koristiti funkciju `fspecial`:

```
» [img,map] = imread('cameraman.tif');    % učitamo sliku
» img = double(img);
» imgFT = fft2(img);                      % računamo 2D DFT slike
» flt = fspecial('motion',10,35);         % odabiremo filter
» fltFT = fft2(flt,size(img,1),size(img,2));
» imgMBFT = imgFT .* fltFT;               % da bi mogli množiti spektre
                                           % dimenzije DFT transformacija
                                           % moraju biti iste
» imgMB = ifft2(imgMBFT);                 % računamo inverz
» imagesc(imgMB);                         % prikazujemo rezultat
```

Prikazani postupak množenja u domeni dvodimenzionalne diskretne Fourierove transformacije odgovara cirkularnoj konvoluciji<sup>2</sup> definiranoj s

$$g(x, y) = \sum_{i=0}^{M_x-1} \sum_{j=0}^{N_x-1} f(i, j) h((x-i) \bmod M_f, (y-j) \bmod N_f),$$

gdje su  $M_x$  i  $N_x$  dimenzije slike  $f(x, y)$  uz  $M_f \geq M_h$  i  $N_f \geq N_h$ . Ako želimo pomoću diskretne Fourierove transformacije izračunati linearnu konvoluciju potrebno je proširiti signale  $f(x, y)$  i  $h(x, y)$  s nulama tako da dimenzije oba signala budu barem  $M_f + M_h - 1 \times N_f + N_h - 1$ . Za tako proširene signale množenje u domeni diskretne Fourierove transformacije odgovara linearnoj konvoluciji signala.

Uobičajeno se signal proširuje s nulama do prve veće potencije broja dva zbog ubrzanja postupka (korištenje FFT algoritma). Sada je nakon množenja signala u domeni transformacije rezultat inverzne transformacije dimenzija većih od rezultata linearne konvolucije (veći od  $M_f + M_h - 1 \times N_f + N_h - 1$ ). Rezultat linearne konvolucije koji tražimo se nalazi unutar izračunate inverzne Fourierove transformacije te je samo potrebno odbaciti višak koji ne nosi informaciju. Točan položaj rezultata će ovisiti o načinu na koji smo signal  $h(x, y)$  proširili s nulama.

- Iz USC-SIPI baze slika odaberite jednu fotografiju. Korištenjem naredbe `fspecial` konstruirajte impulsni odziv  $h(x, y)$  2D sustava (sami odaberite sustav). Ako su  $M_f$  i  $N_f$  dimenzije slike  $f(x, y)$  i ako su  $M_h$  i  $N_h$  dimenzije impulsnog odziva  $h(x, y)$  neka bude  $M_f \approx 10M_h$  i  $N_f \approx 10N_h$  (odaberite neparan broj uzoraka po  $x$  i  $y$  osi za impulsni odziv).
- Korištenjem naredbe `conv2` izračunajte linearnu konvoluciju koja je ujedno i odziv odabranog LSI sustava. Prikažite rezultat!

<sup>2</sup> Primijetite da za diskretnu Fourierovu transformaciju pretpostavljamo periodičnost signala, tj. signal je diskretni periodični signal. Stoga izravno množenje u DFT domeni odgovara upravo cirkularnoj konvoluciji.

- c) Množenjem u domeni 2D DFT transformacije izračunajte odziv sustava sa i bez proširenja matrica na dimenzije  $M_f + M_h - 1 \times N_f + N_h - 1$ . Kako se periodičnost cirkularne konvolucije manifestira u rezultatu za slučaj kada ne proširimo slike?

Matrica se može jednostavno proširiti s nulama dodavanjem još jednog elementa:

```
» h = [0 -1 0; -1 4 -1; 0 -1 0] % definiramo impulsni odziv dimenzija 3x3

h =

     0     -1     0
    -1     4    -1
     0     -1     0

» h(4,7) = 0 % dodavanjem nule u četvrtom retku i
              % sedmom stupcu proširujemo odziv s nulama

h =

     0     -1     0     0     0     0     0
    -1     4    -1     0     0     0     0
     0     -1     0     0     0     0     0
     0     0     0     0     0     0     0
```

## 3.2. Poboljšanje slike

Drugi dio treće laboratorijske vježbe bavi se metodama poboljšanja slike. Kako se poboljšanje ne može uvijek objektivno mjeriti postoji jako puno postupaka. U ovoj vježbi ćemo razmotriti par primjera, dva vezana uz bezmemorijske sustave za koje se problem poboljšanja svodi na određivanje funkcije koja transformira vrijednosti slike i jedan vezan uz memorijske sustave.

### 10 minuta Zadatak 3.2-1 Gama korekcija

Monitori s katodnom cijevi ulazne napone koji predstavljaju intenzitet ne preslikavaju linearno na zaslon, već je dobiveni intenzitet svjetla kojeg vidimo proporcionalan s potencijom ulaznog napona. Tipična vrijednost potencije je oko 2, što znači da sliku koju želimo prikazati na katodnoj cijevi monitora moramo prilagoditi za prikaz. To činimo tako da intenzitete ne preslikavamo linearno u napon, već vrijednosti najprije potenciramo s potencijom od otprilike 1/2. Taj postupak se zove gama korekcija i uvijek se vrši prije prikaza na zaslonu s katodnom cijevi. Kako različite katodne cijevi imaju različite karakteristike, za određivanje točnih vrijednosti gama faktora potrebna je kalibracija.

Gama korekcija dakle odgovara potenciranju pri čemu smo potenciju označili slovom gama. Za računanje gama korekcije u MATLAB-u se koristi funkcija `imadjust`<sup>3</sup>:

```
» [img,map] = imread('forest.tif'); % učitamo sliku
» imshow(img, map) % prikažimo je
» img = ind2gray(img,map);
» imgG = imadjust(img,[],[],0.5); % računamo gamma korekciju s gama 0,5
```

<sup>3</sup> MATLAB nije idealan alat za određivanje gama korekcije jer nije moguće odmah interaktivno mijenjati iznose gama faktora. Većina boljih aplikacija namijenjenih obradbi slike ili pripremi materijala za tisak dolazi s modulom za određivanje gama korekcije. Kvalitetniji upravljački programi grafičkih kartica također posjeduju mogućnost kalibracije i određivanja ispravnog iznosa gama korekcije. Stoga umjesto određivanja gama korekcije unutar MATLAB-a možete iskoristiti bilo koju od aplikacija koja ima modul za kalibraciju monitora.

- a) (ZA ONE KOJI ŽELE ZNATI VIŠE) Učitajte sliku `psgamma.tif` koja sadrži testni uzorak za određivanje gama korekcije. Uzorak se sastoji od četiri pruge u bijeloj, plavoj, zelenoj i crvenoj boji. Svaka od pruga ima popunjeni pravokutnih u sredini te uzorak sa strane. Sliku je potrebno prikazati na zaslonu uz normalno osvjetljenje. Tada se podešava gama korekcija za svaku od boja na taj način da s nešto veće udaljenosti nije moguće razlikovati popunjeni pravokutnih od odgovarajuće pozadine. Jednom kada su određene vrijednosti gama korekcije za crvenu, zelenu i plavu boju četvrti bijeli pravokutnik trebao bi biti stopljen s pozadinom. Pokušajte odrediti iznose korekcije za vaš zaslon (kalibracija ima smisla samo ako koristite zaslon s katodnom cijevi ili LCD koji nije spojen putem DVI sučelja).

### 15 minuta **Zadatak 3.2-2 Histogram prvog reda**

Za sive slike histogram prvog reda predstavlja relativnu frekvenciju pojave različitih svjetlina sive boje u slici. Histogram prvog reda u MATLAB-u dobivamo naredbom `imhist`:

```
» [img,map] = imread('canoe.tif'); % učitavamo indeksiranu sliku
» imhist(img,map)                 % prikazujemo histogram uz
                                   % odgovarajuću paletu boja
» [img,map] = imread('moon.tif'); % učitavamo sivu sliku
» imhist(img)                     % paleta sada nije potrebna
```

Izjednačavanje histograma je operacija pri kojoj se vrijednosti točaka slike mijenjaju tako da histogram nove slike postane jednolik, odnosno tako da broj točaka u slici bude približno jednak za sve svjetline. Histogram slike izjednačava se pozivom funkcije `histeq`:

```
» [img,map] = imread('forest.tif'); % učitamo sliku
» img = ind2gray(img,map);          % pretvaramo sliku u sivu
» imgEQ = histeq(img);              % izjednačavamo histogram
» figure, imshow(img)
» figure, imshow(imgEQ)
```

- a) Učitajte sliku `forest.tif` te iz USC-SIPI baze slika učitajte sliku `4.2.02.tiff`. Prikažite histograme prvog reda tih slika prije i nakon izjednačavanja histograma. Vidite li više detalja prije ili nakon izjednačavanja?
- b) Učitajte sliku `auto.tif` te modelirajte histogram<sup>4</sup> tako da broj na registarskim tablicama auta sa slike postane čitljiv.

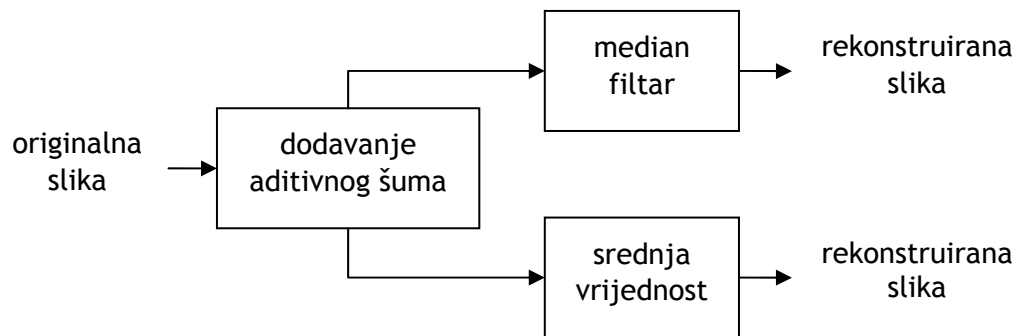
Ako je  $F$  neka funkcija distribucije i neka je  $F^{-1}$  njezina inverzna distribucija. Ako je  $U$  skup brojeva jednolike distribucije na intervalu  $[0,1]$  (u ovom slučaju skup vrijednosti točaka slike), onda skup brojeva  $F^{-1}(U)$  ima distribuciju  $F$ . Ovu činjenicu možemo koristiti za modeliranje histograma:

```
» [img,map] = imread('forest.tif'); % učitamo sliku
» img = ind2gray(img,map);          % pretvaramo sliku u sivu
» imgEQ = histeq(img);              % izjednačavamo histogram
» imgMEQ = imscale(img,[0 1]);      % sada slika imgMEQ ima približno
                                   % jednoliku distribuciju na [0 1]
» imgMEQ = norminv(imgMEQ,0,10);    % slika sada ima normalnu razdiobu
                                   % s očekivanjem 0 i devijacijom 10
```

<sup>4</sup> Pogledajte u kojem intervalu brojeva se nalaze vrijednosti točaka registarske tablice te odredite preslikavanje koje rasteže taj interval na veći. Opisani postupak se može promatrati kao modeliranje histograma ili kao konstrukcija nelinearnog preslikavanja koje poboljšava sliku.

### 15 minuta **Zadatak 3.2-3 Usrednjavanje i median filter**

U ovom dijelu vježbe usporediti ćemo statistički median filter i obični filter za usrednjavanje u primjeni uklanjanja aditivnog šuma. Blok shema eksperimenta prikazana je na slici.



Aditivni šum slici dodajemo pozivom funkcije `imnoise`:

```
» [img,map] = imread('saturn.tif'); % učitamo sliku  
» imgGN = imnoise(img,'gaussian'); % dodajemo bijeli šum  
» imgSP = imnoise(img,'salt & pepper'); % dodajemo binarni šum
```

Median filter je statistički filter koji za svaki ulazni blok zadane veličine na izlazu daje median tog bloka. Postoje i drugi statistički filteri koji računaju određenu statističku veličinu za pojedini ulazni blok. U MATLAB-u se median filtriranje ostvaruje funkcijom `medfilt2`:

```
» imgMF=medfilt2(imgSP,[5 5]); % filtriramo šum s median filtrom  
% uz blok veličine 5x5
```

Usrednjavanje koje također možemo promatrati kao statistički filter (rezultat je procjena srednje vrijednosti ulaznog bloka) u MATLAB-u realiziramo računanjem konvolucije:

```
» maska = ones(5); % kreiramo masku željenih dimenzija (5x5)  
» maska = maska / sum(maska(:)); % usrednjavamo masku  
» maska % filter je određen s maskom u kojoj su svi  
% elementi jednaki
```

maska =

0.0400	0.0400	0.0400	0.0400	0.0400
0.0400	0.0400	0.0400	0.0400	0.0400
0.0400	0.0400	0.0400	0.0400	0.0400
0.0400	0.0400	0.0400	0.0400	0.0400
0.0400	0.0400	0.0400	0.0400	0.0400

```
» imgSF = conv2(imgSP, maska, 'same');
```

- a) (PRIPREMA) Razmotrite najjednostavniji filter za usrednjavanje dimenzija  $N \times N$  s impulsnim odzivom

$$h(x,y) = \begin{cases} \frac{1}{N^2}, & \left\lceil -\frac{N-1}{2} \right\rceil \leq x \leq \left\lceil \frac{N-1}{2} \right\rceil \text{ i } \left\lceil -\frac{N-1}{2} \right\rceil \leq y \leq \left\lceil \frac{N-1}{2} \right\rceil \\ 0, & \text{inače} \end{cases}$$

Odredite faznu karakteristiku filtra u ovisnosti o dimenziji  $N$ . Je li karakteristika ista za parni i neparni  $N$ ? Objasnite zašto se uobičajeno odabiru LSI sustavi s konačnim impulsnim odzivom koji ima neparan broj uzoraka po  $x$  i  $y$  osi.



- b) Iz USC-SIPI baze slika odaberite neku fotografiju te joj dodajte Gaussov i impulsni šum (*salt&pepper noise*). Usporedite učinak oba filtra na uklanjanje Gaussovog i na uklanjanje impulsnog šuma. Komentirajte rezultate. Koji filter je primjereniji za uklanjanje pojedinih vrsta šuma?

### 3.3. Ekstrakcija značajki i segmentacija slike

Treći dio treće laboratorijske vježbe bavi se ekstrakcijom značajki i segmentacijom slike. U velikom broju aplikacija potrebno je izdvojiti prostorne značajke slike. Na temelju tih značajki računalo može opisati i interpretirati sadržaj slike. Takvo izdvajanje prostornih značajki predstavlja početnu kariku u postupku analize slike. Prostorne značajke se izravno određuju iz vrijednosti intenziteta slike.

#### 15 minuta **Zadatak 3.3-1 Amplitudne značajke slike**

Amplitudne značajke slike za svaku točku slike određujemo kao neke parametre izračunate iz vrijednosti intenziteta točaka u okolini promatrane točke. U MATLAB-u postoji nekoliko funkcija koje primjenjuju zadanu operaciju na definiranu okolinu za svaku točku slike (`blkproc`, `colfilt` i `nlfilt`), no za potrebe vježbi je napisana funkcija `imfeat2` koju ćemo koristiti:

```
» help imfeat2 <ENT>
```

```
IMFEAT2 - Određuje značajke slike.  
Z = IMFEAT2(I, [M N], značajka) određuje zadanu značajku  
slike I na blokovima veličine M sa N. Značajka može biti  
'mean' za aritmetičku sredinu bloka MxN  
'stddev' za standardnu devijaciju bloka MxN  
'min' za minimum bloka MxN  
'max' za maksimum bloka MxN  
'entropy' za entropiju bloka MxN  
'energy' za energiju bloka MxN
```

Rezultat je matrica u kojoj pojedini element odgovara traženoj značajki bloka veličine M sa N u ulaznoj slici.

Od značajki koje `imfeat2` određuje 'max', 'min', 'mean' i 'stddev' su amplitudne značajke, dok su 'entropy' i 'energy' značajke histograma prvog reda.

- a) Iz USC-SIPI baze slika odaberite neku sliku manjih dimenzija, do 256×256 (možete i smanjiti neku veću sliku). Za odabranu sliku odredite statističke značajke 'max', 'min', 'mean' i 'stddev' na prozoru veličine 5×5 koristeći funkciju `imfeat2`. Komentirajte rezultate.

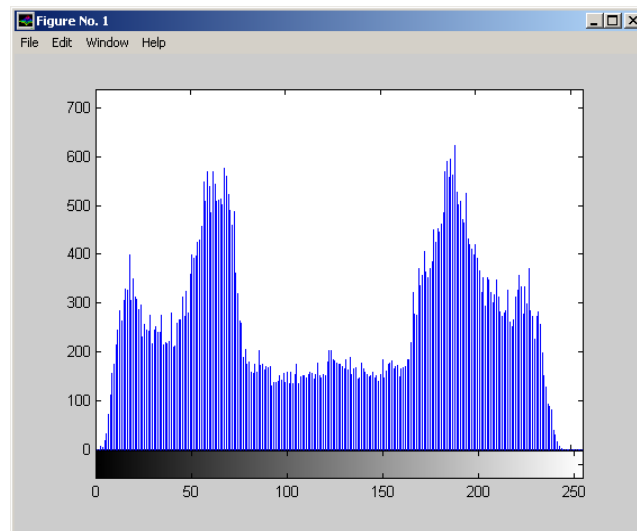
#### 15 minuta **Zadatak 3.3-2 Amplitudna segmentacija**

Amplitudna segmentacija je najjednostavnija metoda segmentacije slike. Korisna je kad amplitudne značajke dovoljno precizno definiraju regije scene. Ako je  $t$  neka vrijednost praga onda točki slike pridružujemo vrijednost 0 ili 1 ovisno o usporedbi s pragom, dakle

$$g(x, y) = \begin{cases} 1, & f(x, y) > t \\ 0, & \text{inače} \end{cases}.$$

Najčešće se za amplitudnu segmentaciju koristi histogram prvog reda kako bi odabrali vrijednost praga.

U ovom dijelu vježbe potrebno je na osnovu histograma slike odrediti prag segmentacije. Ukoliko na slici postoji više različitih regija takvih da je svjetlina točaka unutar regije otprilike jednaka, očekujemo bimodalni ili multimodalni histogram. Jedan takav histogram prikazan je na slici 1.. Na tom histogramu vidimo nekoliko skupina, te bi u slučaju segmentacije na dvije regije za prag odabrali vrijednost oko 130.



Slika 1. Multimodalni histogram

Za amplitudnu segmentaciju u MATLAB-u koristimo funkciju `graystlice` koja ulazne slike segmentira s zadanim pragovima. Pragovi se navode u jednom vektoru i moraju biti između 0 i 1. Ako izlaz funkcije `graystlice` nismo pridružili nekoj varijabli funkcija odmah crta rezultat segmentacije. Postupak u MATLAB-u bi bio sljedeći:

```
» [img,map] = imread('testpat1.tif');      % učitavamo sliku
» imhist(img)                             % prikazujemo histogram
» graystlice(imscale(img),130/255)         % segmentiramo sliku s pragom
                                           % kojeg smo odabrali
```

Prikazani histogram smo mogli shvatiti kao višemodalni. U tom slučaju bi sliku segmentirali na sljedeći način:

```
» graystlice(imscale(img),[45 130]/255)   % segmentiramo sliku s
                                           % pragovima 45 i 130
```

- Učitajte sliku `blood1.tif`. Na toj slici prikazana su krvna zrnca. Kao rezultat segmentacije potrebno je dobiti pojedina krvna zrnca uz što manje šuma. Na tako dobivenoj binarnoj slici krvna zrnca su objekti kružnog oblika sa šupljinom u sredini. Prikažite histogram te odredite prag. Dobivate li segmentirana krvna zrnca?
- Za sliku `blood1.tif` prikažite rezultat segmentacije s odabranim pragom te rezultate koji se dobiju za prag 0,2 veći i 0,2 manji od optimalnog. Utječe li pomak praga značajno na rezultate?
- Ukoliko histogram slike `blood1.tif` interpretiramo kao trimodalni gdje bi se nalazio drugi prag? Što dobivamo u ovom slučaju kao rezultat segmentacije?

### 20 minuta **Zadatak 3.3-3 Određivanje rubova**

Za određivanje rubova u MATLAB-u koristi se funkcija `edge`. Podržane je nekoliko različitih metoda određivanja rubova. Funkcija kao rezultat vraća binarnu sliku veličine ulazne slike na kojoj su označeni detektirani rubovi.

Podržane metode su `'sobel'`, `'prewitt'`, `'roberts'`, `'zerocross'`, `'log'` i `'canny'`. Za metode koje se oslanjaju na procjenu prve derivacije estimacija derivacije se računa za svaku točku te se na tako dobivenoj slici vrši amplitudna segmentacija. Prag za amplitudnu segmentaciju moguće je zadati unaprijed. Za metode koje se oslanjaju na procjenu druge derivacije detektira se prolaz kroz nulu. I za te metode je moguće zadati prag, no prag tada određuje strminu ruba. U MATLAB-u se naredba koristi na sljedeći način:

```
» [img,map] = imread('saturn.tif'); % učitavamo sliku
» rub = edge(img,'sobel');          % primjenimo Sobelovu metodu
» imshow(rub)                      % prikazimo detektirane rubove
» rub = edge(img,'sobel',0.02);     % zadali smo osjetljivost 0,02
» imshow(rub)
» rub=edge(img,'log',[ ],4);        % primjenjujemo LoG metodu
                                    % uz devijaciju 4
» imshow(rub)
```

- Iz USC-SIPI baze slika učitajte sliku `4.2.07.tiff`. Primijenite na sliku funkciju `edge` te prikazite rezultate za Sobelov operator uz razne vrijednosti praga. Za koju vrijednost se dobije najbolji rezultat?
- Na slici `4.2.07.tiff` primijenite funkciju `edge` uz automatski odabir praga za Sobelov operator. Razlikuje li se ta vrijednost od vrijednosti koju ste dobili u prethodnom zadatku? Za koji slučaj dobivamo bolji rezultat?
- Na slici `4.2.07.tiff` primijenite `'zerocross'` ili `'log'` metodu. Dodajte šum slici pomoću naredbe `imnoise` te usporedite rezultate `'zerocross'` ili `'log'` metode s rezultatima za Sobelov operator.
- (ZA ONE KOJI ŽELE ZNATI VIŠE) Na slici `4.2.07.tiff` primijenite Cannyjevu metodu. Usporedite rezultate s rezultatima detekcije za ostale metode.

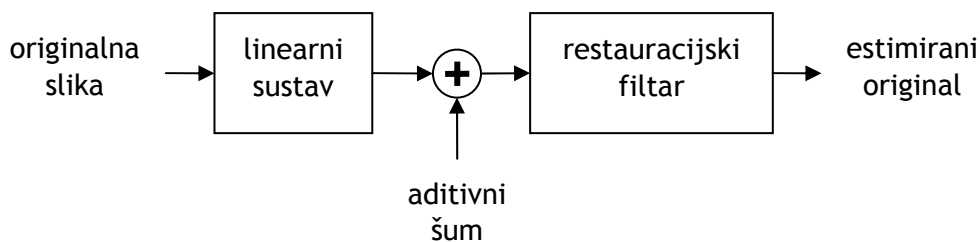
## 3.4. Obnavljanje slike

Zadnji dio treće laboratorijske vježbe bavi se problemom obnavljanja slike. Obnavljanje slike (eng. *image restoration*) ima za svrhu dobivanje slike što sličnije izvornoj slici i to samo temeljem poznavanja degradirane slike i načina ili modela degradacije. Za razliku od problema poboljšanja slike kojeg je teško strogo matematički formulirati, problem obnavljanja slike je jasno definiran.

Degradiranje slike obično modeliramo kao složeni sustav koji se sastoji od linearnog memorijskog prostorno nepromjenjivog sustava i bezmemorijske nelinearnosti. Osim toga realni sustavi uvijek unose šum koji se najčešće modelira kao aditivni šum. Uz zanemarenu bezmemorijsku nelinearnost<sup>5</sup> problem degradacije i obnavljanja slike se može predstaviti slijedećom blok shemom prikazanom na slici 2..

---

<sup>5</sup> Kako nelinearnost modeliramo kao bezmemorijski sustav, određivanje postupka restauracije svodi se na problem određivanja inverzne funkcije.



Slika 2. Model degradacije slike

### 20 minuta **Zadatak 3.4-1 PSF i OTF**

Pretpostavimo da je slika propuštena kroz linearni prostorno nepromjenjiv sustav koji je opisan s funkcijom razmazivanja točke<sup>6</sup> (*Point Spread Function*). U tom slučaju nam PSF predstavlja upravo impulsni odziv sustava. Uz poznatu PSF možemo odrediti i prijenosnu funkciju sustava (*Optical Transfer Function*). Taj linearni sustav koji predstavlja funkciju degradacije ćemo modelirati kao dvodimenzionalni FIR filter. Za konstruiranje različitih degradacija možete koristiti MATLAB funkciju `fspecial`.

Ako želite primijeniti više različitih degradacija (npr. i atmosfersko zamućenje i pomak kamere) morate računati više uzastopnih konvolucija. Odziv takvog složenog sustava možete odrediti računajući linearnu konvoluciju odziva pojedinih sustava (kaskadni spoj).

- a) Odredite PSF i OTF za vertikalne i horizontalne pomake za 5 točaka. Nacrtajte impulsni odziv (PSF) te amplitudni i fazni spektar. Za crtanje možete koristiti naredbu `surf`:
- ```

» PSF = fspecial('gaussian',31,5);    % računamo PSF za Gaussovo zamućenje
» surf(PSF)                          % crtamo PSF
» OTF = psf2otf(PSF, [64 64]);       % računamo OTF
» surf(fftshift(OTF))                % crtamo OTF
» [H,Fx,Fy] = freqz2(PSF);           % računamo 2D DTFT transformaciju
» surf(Fx,Fy,abs(H))
» surf(Fx,Fy,angle(H))

```
- b) Odredite i skicirajte PSF i OTF za kombinaciju zamućenja, i to za slabo Gaussovo zamućenje (mala vrijednost devijacije sigma) i pomak od desetak točaka pod nekim kutem. Primijenite dobivenu PSF na neku od aero-fotografija iz USC-SIPI baze (npr. 2.2.01.tiff ili 2.1.08.tiff).

### 25 minuta **Zadatak 3.4-2 Inverzno filtriranje**

Poznajemo li PSF možemo odrediti njenu 2D DFT transformaciju u odabranom broju točaka. Inverzni filter je tada definiran s

$$H_i[k,l] = \frac{1}{H[k,l]},$$

gdje je  $H[k,l]$  2D DFT transformacija PSF funkcije u odabranom broju točaka. Pseudoinverzni filter je definiran izrazom

<sup>6</sup> Dakle impulsnim odzivom za kojeg vrijedi da suma svih uzoraka impulsnog odziva nije jednaka nuli.

$$H_{pi}[k,l] = \begin{cases} \frac{1}{H[k,l]}, & \forall H[k,l] > K \\ 0, & \forall H[k,l] < K \end{cases},$$

gdje je  $K$  proizvoljno odabrana konstanta. Ponekad se prijenosna funkcija pseudoinverznog filtra dodatno ograničava množenjem s prijenosnom funkcijom nekog niskopropusnog filtra.

- Za odabranu PSF iz 3.4-1b zadatka odredite rezultat inverznog i pseudoinverznog filtriranja za slučaj degradacije s i bez aditivnog šuma za neke od slika iz USC-SIPI baze (proizvoljan odabir).
- Za odabrane slike iz a) dijela zadatka odredite srednje kvadratno odstupanje obnovljene slike od izvorne slike i SNR. Poklapa li se kvadratno odstupanje (manje je bolje) i SNR (više je bolje) s kvalitetom obnovljene slike?

Ako se izvorna slika nalazi u varijabli `img`, a obnovljena u varijabli `imgO`, srednje kvadratno odstupanje određujemo na slijedeći način:

```
» mean2((img-imgO).^2) % računamo srednje kvadratno odstupanje
ans =
    0.0094
```

Ponekad nas osim srednjeg kvadratnog odstupanja zanima i omjer snaga signal/šum ili općenito signal/signal:

```
» imgP = abs(fft2(img)).^2; % spektralna gustoća snage izvorne slike
» imgPsr = mean2(imgP) % srednja snaga izvorne slike

imgPsr =
    1.1467e+005

» imgOP = abs(fft2(imgO)).^2; % spektralna gustoća snage obnovljene slike
» imgOPsr = mean2(imgOP) % srednja snaga obnovljene slike

imgOPsr =
    1.2677e+005

» imgPsr/imgOPsr % odnos izvorna/obnovljena slika
% primijetite da ste ga također mogli
% izračunati u dB
ans =
    0.9045
```

- (ZA ONE KOJI ŽELE ZNATI VIŠE) Isprobajte inverzno i pseudoinverzno filtriranje kao u prethodnom zadatku, ali uz kvantizaciju razina na 8 bita (vrijednosti od 0 do 255, za kvantizaciju možete koristiti naredbu `round`). Kako kvantizacija utječe na rezultat filtriranja?

## 25 minuta **Zadatak 3.4-3 Wienerovo filtriranje**

Uz pretpostavku stacionarnosti u širem smislu slučajnog procesa koji predstavlja model slike te uz prisutan nekorelirani aditivni šum srednje vrijednosti nula Wienerov filter je određen izrazom

$$H_w[k, l] = \frac{H^*[k, l]}{H[k, l]H^*[k, l] + \frac{S_{NN}[k, l]}{S_{XX}[k, l]}}$$

gdje je  $H[k, l]$  optička prijenosna funkcija, dok su  $S_{NN}[k, l]$  i  $S_{XX}[k, l]$  spektralne gustoće snage signala i šuma. Izraz se može zapisati i u obliku

$$H_w[k, l] = \frac{1}{H[k, l]} \frac{|H[k, l]|^2 S_{XX}[k, l]}{|H[k, l]|^2 S_{XX}[k, l] + S_{NN}[k, l]} = \frac{1}{H[k, l]} \frac{S_{YY}[k, l]}{S_{YY}[k, l] + S_{NN}[k, l]}$$

iz kojeg se vidi rastav Wienerovog filtra u kaskadu inverznog filtra i običnog Wienerovog filtra za uklanjanje šuma. Problem pri određivanju prijenosne funkcije Wienerovog filtra je upravo estimacija spektralnih gustoća snage signala i šuma (odnosno autokorelacijskih funkcija) te optičke prijenosne funkcije jer najčešće raspolažemo samo s degradiranom slikom. Stoga se u prvoj aproksimaciji pretpostavlja konstantni omjer gustoća spektara snage  $K$ , te je prijenosna funkcija Wienerovog filtra

$$H_w[k, l] = \frac{H^*[k, l]}{H[k, l]H^*[k, l] + K}$$

Pri tome se parametar  $K$  može procijeniti iz degradirane slike. Primijetite da za male vrijednosti parametra  $K$  Wienerov filter postaje blizak inverznom filtru. Wienerovo filtriranje u MATLAB-u radi naredba `deconvwnr`.

- Za odabranu PSF iz 3.4-1b odredite rezultat Wienerovog filtriranja za slučaj degradacije s i bez aditivnog šuma za neke od slika iz USC-SIPI baze. Koristite šum srednje vrijednosti nula. Pretpostavite da ne poznajete vrijednosti gustoće spektra snage slike i šuma, odnosno da odnos spektara modelirate samo konstantom  $K$  (treći parametar koji se prosljeđuje naredbi `deconvwnr`). Odredite srednje kvadratno odstupanje obnovljene slike od izvorne slike. Odredite odnos signal/šum prije i nakon filtracije.
- Ponovite prethodni zadatak uz poznatu autokorelacijsku funkciju šuma te uz autokorelacijsku funkciju slike koju ste procijenili iz degradirane slike (dakle, uz poznate gustoće spektra snage šuma i slike).

Aproksimaciju autokorelacijske funkcije slike možete brzo odrediti korištenjem teorema o cirkularnoj korelaciji

$$r_{ff}(x, y) = f(x, y) \otimes f(-x, -y) \quad \text{O} \bullet \quad F[k, l]F^*[k, l] = |F[k, l]|^2$$

i korištenjem naredbi `fft2` i `ifft2`:

```
» [img, map] = imread('4.2.04.tiff'); % učitavamo sliku
» img = double(rgb2gray(img)); % slika mora biti siva slika
» rii = ifft2(abs(fft2(img)).^2); % koristimo teorem o korelaciji
» imagesc(fftshift(rii)) % maksimum mora biti u centru slike
```

Ponekad će zbog numeričkih pogrešaka biti potrebno ukloniti zanemarivu no ipak prisutnu imaginarnu komponentu korištenjem naredbe `real`.

- Ponovite prethodni zadatak uz pretpostavku poznatih autokorelacijskih funkcija signala  $R_{XX}$  i šuma  $R_{NN}$ .

## 4. Literatura

1. R. C. Gonzales i R. E. Woods, *Digital Image Processing 3<sup>rd</sup> Edition (DIP/3e)*, Prentice Hall, 2008., <http://www.imageprocessingplace.com/>
2. *Image Processing Toolbox User's Guide*, The Mathworks, [http://www.mathworks.com/access/helpdesk/help/pdf\\_doc/images/images\\_tb.pdf](http://www.mathworks.com/access/helpdesk/help/pdf_doc/images/images_tb.pdf)
3. H. Babić, *Signali i sustavi (zavodska skripta)*, FER, Zagreb 1996., [http://sis.zesoi.fer.hr/predavanja/pdf/sis\\_2001\\_skripta.pdf](http://sis.zesoi.fer.hr/predavanja/pdf/sis_2001_skripta.pdf)
4. Ž. Ban, *Osnove MATLAB-a*, FER, Zagreb, 2006. [http://www.fer.hr/\\_download/repository/Osnove\\_matlaba.pdf](http://www.fer.hr/_download/repository/Osnove_matlaba.pdf)
5. *Laboratorij i vještine - MATLAB: Uvod u Simulink*, FER, Zagreb, 2006. [http://www.fer.hr/\\_download/repository/simulinkskripta.pdf](http://www.fer.hr/_download/repository/simulinkskripta.pdf)
6. T. Petković, *Kratke upute za korištenje MATLAB-a*, FER, Zagreb, travanj 2005. [http://www.fer.hr/\\_download/repository/matlab\\_upute.pdf](http://www.fer.hr/_download/repository/matlab_upute.pdf)