UNIVERSITETI I PRISHTINËS

FAKULTETI I SHKENCAVE MATEMATIKE-NATYRORE

DEPARTAMENTI I MATEMATIKËS

SHKENCA KOMPJUTERIKE



Lënda: Procesimi i imazheve

Asistent: Besnik Duriqi Studentët: Grupi 7

## Hyrje

Në këto detyra, do të merremi me operacione të ndryshme për procesimin e imazheve. Detyra e parë përfshin gjenerimin e llojeve të ndryshme të zhurmës duke përdorur shpërndarje specifike dhe aplikimin e tyre në një imazh. Detyra e dytë përqendrohet në shtimin e zhurmës periodike në imazhin "lena.tif" dhe largimin e saj duke përdorur transformimet e Fourier-it. Detyra e tretë prezanton filtrin e refuzimit të bandës Gaussian për largimin e zhurmës periodike. Së fundi, detyra e katërt kërkon identifikimin e elementit strukturor dhe operacioneve matematike morfologjike që përdoren për të arritur Figurën 3 nga regjioni binar origjinal (Figura 2).

### 1).

Krijoni funksionin ‘zhurma’ i cili do të gjenerojë zhurmën me shpërndarje të Rayleigh, Eksponenciale, LogNormal, Salt&Pepper dhe Erlang. Funksioni duhet të merr këta parametra hyrës:

**Zhurma(‘Shpërndarja’, x, y, z, k, s, a, b)**

ku ‘Shpërndarja’ paraqet shpërndarjen me të cilën do të gjenerohet zhurma, x paraqet

rreshtin prej ku do te filloj zhurma, y paraqet shtyllën prej ku do të filloj zhurma, z paraqet

rreshtin prej ku do te përfundojë zhurma, k paraqet shtyllën prej ku do të përfundojë

zhurma, numri i shtresave s ku s=1,2 ose 3, parametri a, parametri b (për parametrat a dhe

b shikoni sllajdet).

### Zgjidhja:

*function Y = Zhurma\_Dt1(tipi, x, y, z, k, s, a, b)*

*Y=-1; %% Vlere random qe perdoret si koefiecient shumezimi per shperndarjen e caktuar*

*pkg load image;*

*img = imread('baboon.jpg');*

*m = k-x;*

*n = z-y;*

*switch(tipi)*

*case 'rayleigh' %% Zhurma\_Dt1('rayleigh',100,100,400,420,3,0.2,0.4);*

*Y = a+sqrt(-b\*log(1-rand(m,n)));*

*for i = 1:1:(m)*

*for j = 1:1:(n)*

*for l = 1:1:s*

*img(x+i,y+j,l) = img(x+i,y+j,l) + 60\*Y(i,j);*

*endfor*

*endfor*

*endfor*

*case 'exponential' %% Zhurma\_Dt1('exponential',100,100,400,420,3,0.2,0.4);*

*Y = -log(1-rand(m,n))/a;*

*for i = 1:1:(m)*

*for j = 1:1:(n)*

*for l = 1:1:s*

*img(x+i,y+j,l) = img(x+i,y+j,l) + 15\*Y(i,j);*

*endfor*

*endfor*

*endfor*

*case 'lognormal' %% Zhurma\_Dt1('lognormal',100,100,400,420,3,0.2,0.4);*

*Y = a\*exp(b\*randn(m,n));*

*for i = 1:1:(m)*

*for j = 1:1:(n)*

*for l = 1:1:s*

*img(x+i,y+j,l) = img(x+i,y+j,l) + 100\*Y(i,j);*

*endfor*

*endfor*

*endfor*

*case 'salt & pepper' %% Zhurma\_Dt1('salt & pepper',100,100,400,420,3,0.2,0.4);*

*Y = img(x:x+m,y:y+n,1:3);*

*Y = imnoise(img(x:x+m,y:y+n,1:3),'salt & pepper',0.1);*

*for i = 1:1:(m)*

*for j = 1:1:(n)*

*for l = 1:1:s*

*img(x+i,y+j,l) =Y(i,j,l);*

*endfor*

*endfor*

*endfor*

*imshow(img,[]);*

*case 'erlang' %% Zhurma\_Dt1('erlang',100,100,400,420,3,0.2,0.4);*

*Y = -log(1-rand(m,n))/a;*

*for i = 2:b*

*Y = Y -log(1-rand(m,n))/a;*

*endfor*

*for i = 1:1:(m)*

*for j = 1:1:(n)*

*for l = 1:1:s*

*img(x+i,y+j,l) = img(x+i,y+j,l) + 15\*Y(i,j);*

*endfor*

*endfor*

*endfor*

*end*

*imshow(img,[]);*

*endfunction*

Funksioni ekzekutohet nga Command Window duke thirrur funksioni Zhurma\_Dt1 me parametrat perkates varesisht nga shperndarja. [1]

Ne vazhdim shohim rezultatet e nxjerra nga funksioni.

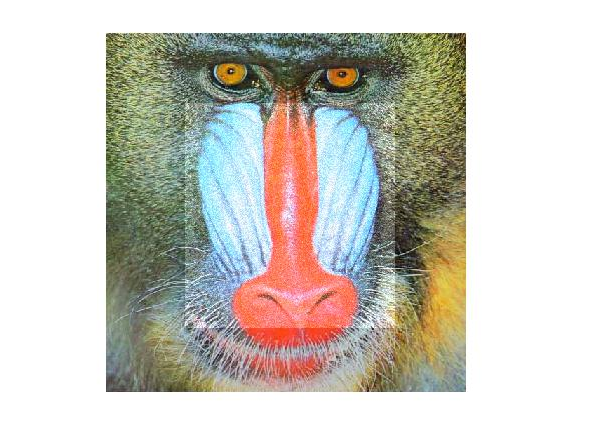


Figura 1: Shperndarja e Rayleigh-it

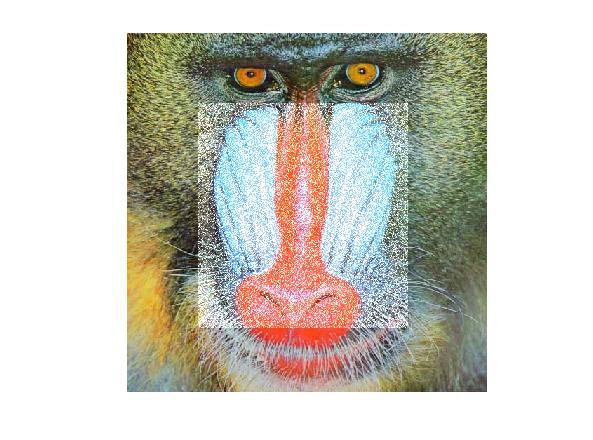


Figura 2: Shperndarja eksponenciale

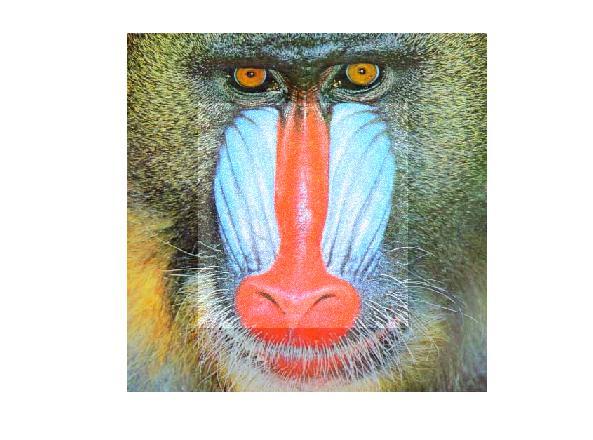


Figura 3: Shperndarja salt & pepper

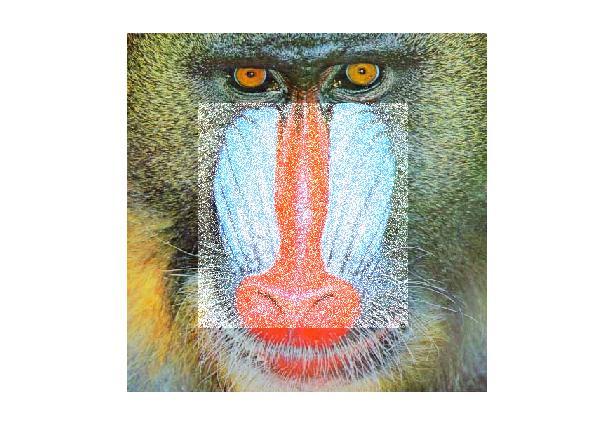


Figura 4: Shperndarja e Erlang-ut

### 2).

Shtoni zhurmën periodike (mëposhtë) te imazhi ‘lena.tif’ dhe largojeni me anë të

transformimeve Furie?

27\*cos((pi\*x)/2+(pi\*y)/3)+ 20\*sin((pi\*x)/5+(pi\*y)/3)

### Zgjidhje:

*C = ones(256:256);*

*for I = 1:256*

*for j = 1:256*

*A(I,j) = 27\*cos(((pi\*I)/2) + ((pi\*j)/3))+20\*sin(((pi\*I)/5) + ((pi\*j)/3));*

*endfor*

*endfor*

*I = imread('lena.tif');*

*I = im2double(I);*

*IM = I + A;*

*imshow(IM);*

*figure,*

*FI = fft2(IM);*

*SFI = fftshift(FI);*

*mesh(log(abs(SFI)));*

*figure*

*imshow(log(abs(SFI)),[]);*

*for I = 1:256*

*for j = 1:256*

*if (sqrt((I-64)^2 + (j-86)^2) <= 9)*

*C(I,j) = 0;*

*endif*

*if (sqrt((I-194)^2 + (j-170)^2) <= 9)*

*C(I,j) = 0;*

*endif*

*if (sqrt((I-151)^2 + (j-170)^2) <= 25)*

*C(I,j) = 0;*

*endif*

*if (sqrt((I-100)^2 + (j-84)^2) <= 25)*

*C(I,j) = 0;*

*endif*

*endfor*

*endfor*

*C(63:65,:) = 0;*

*C(100:104, :) = 0;*

*C(192:194, :) = 0;*

*C(152:155, :) = 0;*

*C(:, 84:88) = 0;*

*C(:,168:173) = 0;*

*figure,*

*imshow(C,[]);*

*figure,*

*IMM = SFI.\*C;*

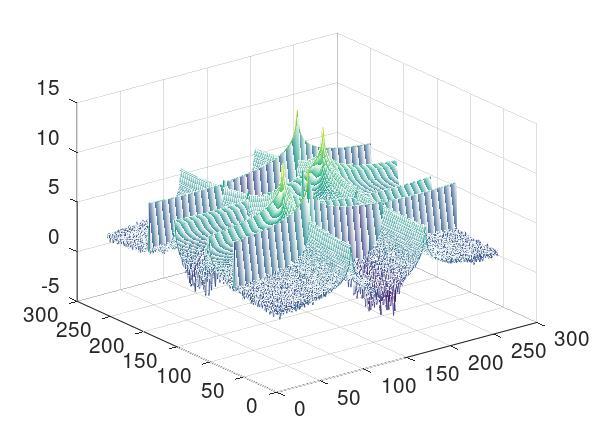
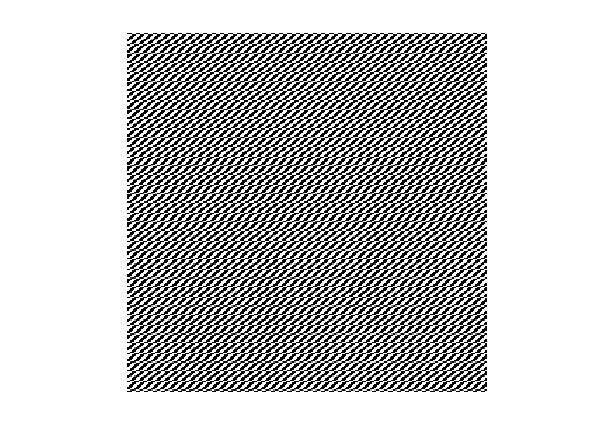
*imshow(IMM);*

*skk = ifftshift(IMM);*

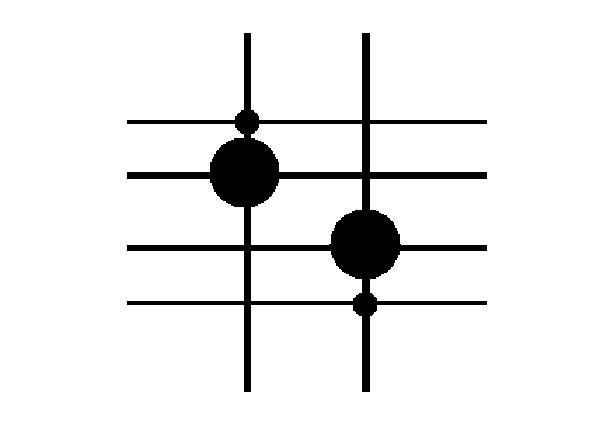
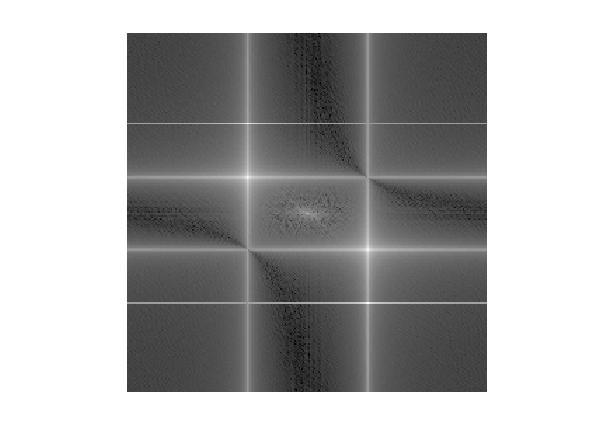
*imazhi = ifft2(skk);*

*imshow(imazhi);*

Fillimisht imazhit fillestar i shtojmë zhurmën ne baze te funksionit te dhënë, dhe pas shtimit te zhurmës dhe grafikut 3D kemi pamjen ne vijim:

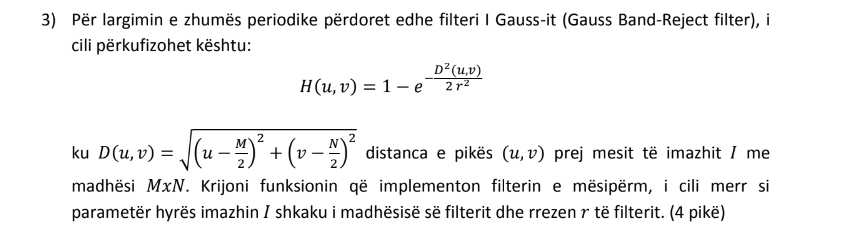


Pastaj duke gjetur me saktësi koordinatat e zhurmës krijojmë filterat ne forme rrethi me një rreze te caktuar dhe largojmë zhurmën. [2]



Si dhe rezultati përfundimtar qe fitohet pas largimit te zhurmës është si ne vijim:





Këtu është implementimi i ekuivalentë i filtrit Gauss Band-Reject në Octave:

*function filtered\_image = gauss\_band\_reject\_filter(image, size, radius)*

*% Llogaritja e qendrës së imazhit*

*center = floor(size(image) / 2) + 1;*

*% Llogaritja e distancave të pikave prej qendrës së imazhit*

*distances = sqrt(sum((ndgrid(1:size(image, 1), 1:size(image, 2)) - center).^2, 3));*

*% Krijimi i maskës së filtrit*

*mask = ones(size(image));*

*mask(distances >= center(1) - size/2 & distances <= center(1) + size/2 & distances <= radius) = 0;*

*% Aplikimi i filtrit Gauss përmes maskës*

*filtered\_image = imgaussfilt(image, size/2) .\* mask;*

*end*

Kur thirret ky funksion, do të kthehet imazhi i filtruar duke përdorur filtrin Gauss Band-Reject për largimin e zhumës periodike.

Mund ta përdorni këtë funksion duke kaluar një imazh si argument hyrës, së bashku me madhësinë e filtrit dhe rrezen e filtrit. [3] Për shembull:

*pkg load image;*

*% Ngarkimi i imazhit*

*image = imread("imazhi.jpg");*

*% Konvertimi i imazhit në gri nëse është e nevojshme*

*if size(image, 3) == 3*

*image = rgb2gray(image);*

*end*

*% Aplikimi i filtrit Gauss Band-Reject për largimin e zhumës periodike*

*filtered\_image = gauss\_band\_reject\_filter(image, 20, 50);*

*% Paraqitja e imazhit origjinal dhe të filtruar*

*subplot(1, 2, 1);*

*imshow(image);*

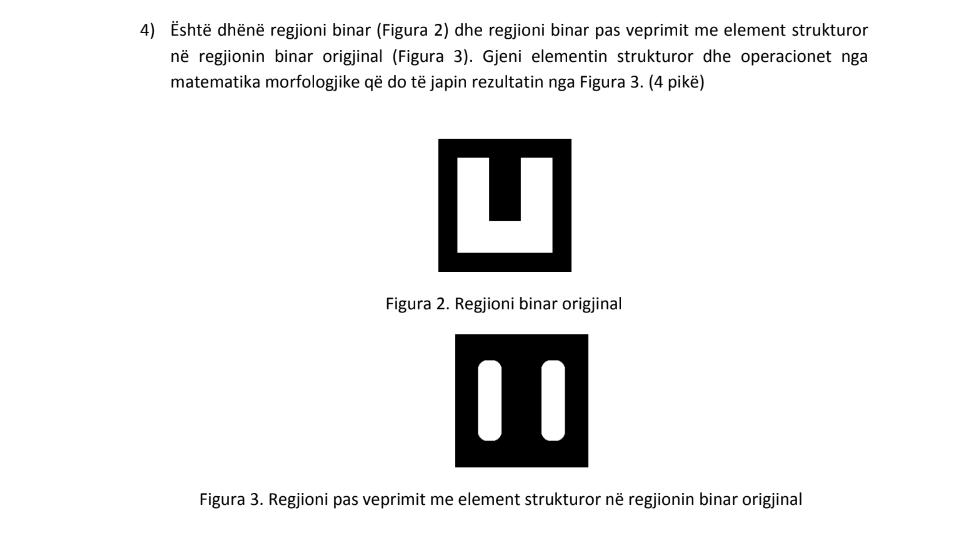
*title("Imazhi origjinal");*

*subplot(1, 2, 2);*

*imshow(filtered\_image);*

*title("Imazhi i filtruar");*

Mund ta ndryshojmë madhësinë e filtrit dhe rrezes sipas nevojës.

  
  
Këtu është një shembull i kodit në Octave duke përdorur paketën `image` për të zbatuar operacionin e dilatimit në regjionin binar origjinal dhe për të krijuar figurën '| |':

*pkg load image;*

*% Ngarkimi i Figurës 2 (regjionit binar origjinal)*

*image = imread('figura2.png');*

*image = rgb2gray(image);*

*% Krijimi i elementit strukturor (vijës vertikale)*

*vertical\_line = ones(5, 1);*

*% Zbatimi i operacionit të dilatimit*

*dilated\_image = imdilate(image, vertical\_line);*

*% Shfaqja e rezultatit (Figura 3)*

*imshow(dilated\_image);*

*title('Figura 3');*

Pasi të ekzekutohet kodi, rezultati do të shfaqet në një dritare të re me titullin 'Figura 3'. Mund të ndryshojmë titullin e dritares nëse dëshirojmë.

## Përfundimi

Në këto detyra, eksploruam teknika dhe koncepte të ndryshme në procesimin e imazheve. Mësuam si të gjenerojmë dhe aplikojmë lloje të ndryshme të zhurmës në imazhe, si të largojmë zhurmën periodike duke përdorur transformimet e Fourier-it, dhe si të përdorim filtrin e refuzimit të bandës Gaussian për largimin e zhurmës. Gjithashtu, zbuluam rëndësinë e elementeve strukturorë dhe operacioneve matematike morfologjike në modifikimin e regjioneve binare. Duke kuptuar këto teknika, mund të përmirësojmë dhe analizojmë imazhe për aplikime të ndryshme, si përmirësimi i cilësisë së imazhit ose nxjerrja e informacionit të rëndësishëm nga regjionet binare.  
  
**Aplikime në jetën reale:**

Detyra 1: Gjenerimi i llojeve të ndryshme të zhurmës

Shembull: Një praktik i gjenerimit të zhurmës me ndryshime të ndryshme është në algoritmet e mëdha të zhurmës nga imazhet. Duke disa zhurmë të llojeve Rayleigh, Ekponenciale, LogNormale, Salt&Pepper ose Erlang në një imazh, hulumtuesit të mund të zhvillojnë dhe të testojnë metodat e mëdha që kanë të bëjnë me të tjerët.

Detyra 2: Shtimi i zhurmës periodike dhe largimi i saj me anë të transformimeve të Fourier-it

Shembull: Në markatim të imazheve, një model i zhurmës periodike mund të shtohet në një imazh për të fshehur një shenjë të fshehur. Më pas, Transformimi i Fourier-it mund të përdoret për të izoluar dhe të mëdhenj të zhurmshëm periodike, duke informacion informacionin e shënjuar të fshehur në imazh.

Detyra 3: Implementimi i filtrit Gaussian Band-Reject

Shembull: Një aplikacion i filtrit Gaussian Band-Reject është në astronomi. Imazhet astronomike shpeshherë nga zhurma periodike e padëshiruar e ndjenjave nga turbulenca atmosferike ose artefaktet e sensorëve. Duke u filtruar nga Gaussian Band-Reject me një aplikim dhe rreze të ndikimit, astronomët mund të zvogëlojnë frekuencat e veçanta të zhurmës duke përdorur në të gjitha karakteristikat e astronomisë së dëshiruar.

Detyra 4: Operacionet morfologjike në imazhe binare

Shembull: Operacionet morfologjike në imazhet binare kanë aplikime në njohjen e karaktereve. Duke kryer operacione si erozioni dhe dilatimi, është e mundur të kompjuterikohet ose të përsoset forma e karakterit të një imazhi binar, duke rritur saktësinë e sistemeve optike të karakterit (OCR). [4]

## Referencat

|  |  |
| --- | --- |
| [1] | R. C. G. &. R. E. Woods, «"Digital Image Processing",» Pearson Education, 2017. |
| [2] | R. E. W. &. S. L. E. Rafael C. Gonzalez, «"Digital Image Processing Using MATLAB",» Pearson Education, 2004. |
| [3] | T. F. C. &. J. (. Shen, «"Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods",» Society for Industrial and Applied Mathematics (SIAM), 2005. |
| [4] | J. G. P. &. D. G. Manolakis, «"Digital Signal Processing: Principles, Algorithms, and Applications",» Pearson Education, 2006. |