

**UNIVERSITETI I PRISHTINËS**  
**FAKULTETI I SHKENCAVE MATEMATIKE – NATYRORE**  
**DEPARTAMENTI I MATEMATIKËS**  
**PROGRAMI: Shkencë kompjuterike**



**Detyra e tretë**  
**Lënda: Procesim i imazheve**

**Studentët:**

*Altin Duraku*

*Bleron Sylmetaj*

*Erdin Osmani*

*Semestri i gjashtë*

*Maj 2023, Prishtinë*

## Përshkrimi i detyrave

Kemi zgjidhur një seri ushtrimesh që janë projektuar për të praktikuar aftësitë e tyre në procesimin e imazheve. Ushtrimet mbulojnë një gamë temash, duke përfshirë gjenerimin e llojeve të ndryshme të zhurmave me shpërndarje të ndryshme, shtimin e zhurmave periodike në një imazh dhe largimin e tyre duke përdorur transformimet Fourier dhe filterin Gauss Band-Reject. Përveç kësaj, ushtrimet kërkojnë përdorimin e operacioneve matematike morfologjike për të manipuluar rajone binare në imazhe. Çdo ushtrim ofron udhëzime të qarta mbi ato që duhet bërë dhe cilat funksione ose teknika duhet të përdoren për të plotësuar detyrën. Nga ana tjetër, kjo seri ushtrimesh është një mjet i shkëlqyeshëm për përdoruesit për të përmirësuar aftsinë e tyre në procesimin e imazheve duke fituar eksperiencën praktike me probleme reale.

## Detyra 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 të filloj zhurma, y paraqet shtyllën prej ku do të filloj zhurma, z paraqet rreshtin prej ku do të 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).

### Zgjidhje

```
function Y = Zhurma_Dt1(tipi, x, y, z, k, s, a, b)
```

```
    Y=-1;      %% Vlere random qe perdoret si koeficient 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

```

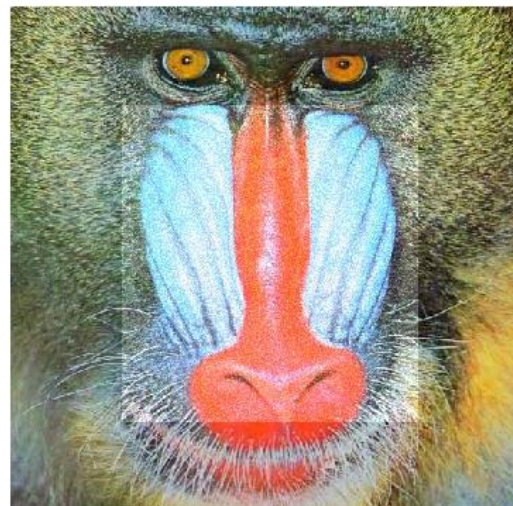
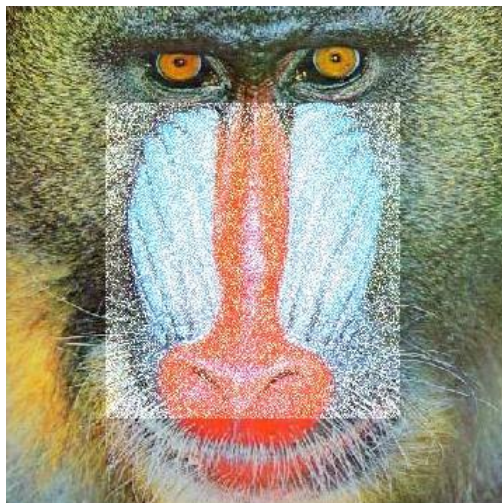
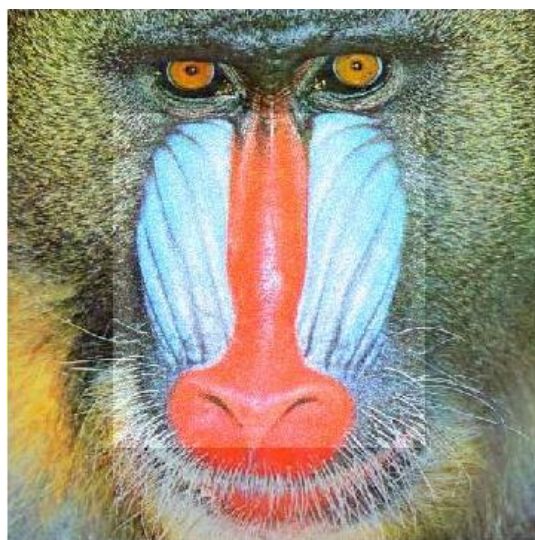


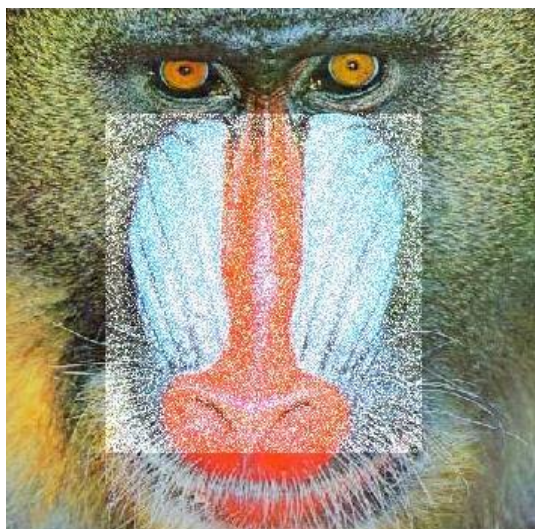
Figura 1: Shpërndarja e Rayleigh



*Figura 2: Shpërndarja eksponenciale*



*Figura 3: Shpërndarja salt & pepper*



*Figura 4: Shpërndarja Erlang*

## Detyra 2

Shtoni zhurmën periodike (mëposhtë) te imazhi 'lena.tif' dhe largojeni me anë të transformimeve Furie?

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

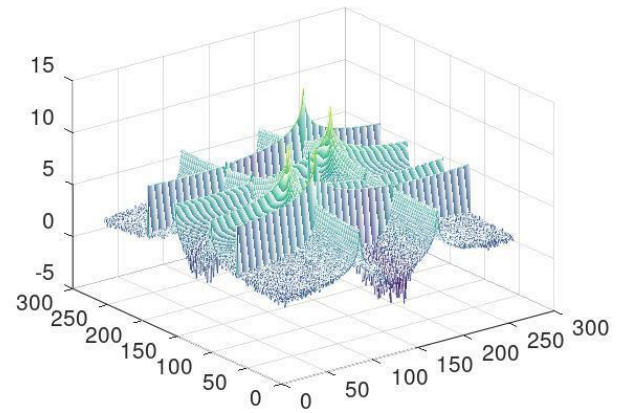
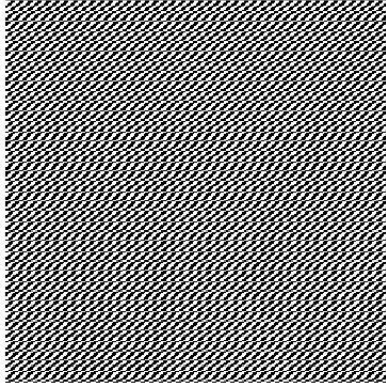
Zgjidhje

```
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);
```

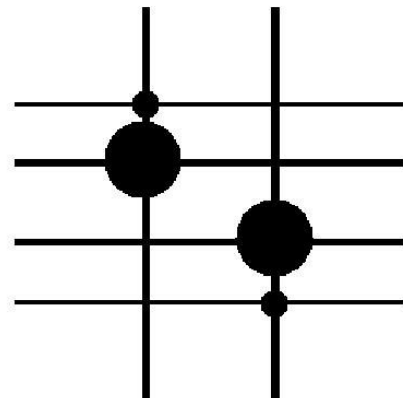
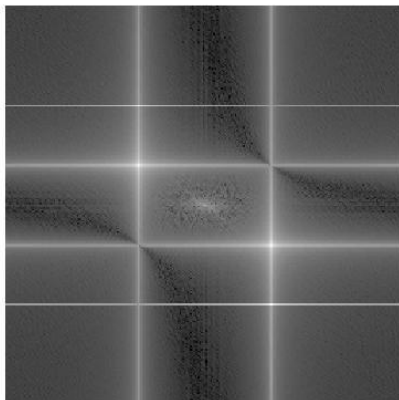
```

skk = ifftshift(IMM);
imazhi = ifft2(skk);
imshow(imazhi);

```



Pastaj duke gjetur koordinatat e zhurmës krijojmë filterat në formë rrethi me nje rreze të caktuar dhe largojmë zhurmën.



Rezultati përfundimtar që fitohet pas largimit të zhurmës është:





### Detyra 3

Për largimin e zhurmës periodike përdoret edhe filteri i Gauss-it (Gauss Band-Reject filter), i cili përkufizohet kështu:

$$H(u, v) = 1 - e^{-\frac{D^2(u, v)}{2r^2}}$$

Ku  $D(u, v) = \sqrt{\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2}$  distanca e pikës (u, v) prej mesit të imazhit me madhësi Krijoni funksionin që implementon filterin e mësipërm, i cili merr si parametër hyrës imazhin shkaku i madhësisë së filterit dhe rrezën të filterit.

#### Zgjidhje

Këtu është implementimi i ekuivalent 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 rrezën e filtrit. 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.

## Detyra 4

Është dhënë regjioni binar (Figura 2) dhe regjioni binar pas veprimit me element strukturor në regjionin binar origjinal (Figura 3). Gjeni elementin strukturor dhe operacionet nga matematika morfologjike që do të japin rezultatin nga Figura 3.







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');
```

Sigurohuni që të keni ngarkuar figurën 2 (regjionin binar origjinal) me emrin 'figura2.png' në direktorinë tuaj të punës përpara se të ekzekutoni kodin. Ndryshoni madhësinë e elementit strukturor sipas nevojës tuaj duke ndryshuar dimensionet e vektorit `vertical\_line` në linjën 9.

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