

DETYRA 3 – ZHURMA DHE MORFOLOGJIA MATEMATIKE

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

Kodi i dhënë përcakton një funksion të quajtur Zhurma që gjeneron lloje të ndryshme zhurmash bazuar në shpërndarjen e specifikuar. Funksioni merr disa parametra hyrës, duke përfshirë llojin e shpërndarjes (shpërndarja), koordinatat e rajonit ku duhet të krijohet zhurma (x, y, z, k) dhe parametrat specifike për secilën shpërndarje (s, a, b).

Brenda funksionit, një matricë zhurme inicializohet me zero bazuar në dimensionet e rajonit të specifikuar. Sythet e mbivendosur përsëriten mbi çdo piksel brenda rajonit, dhe bazuar në shpërndarjen e zgjedhur, vlera korresponduese e zhurmës gjenerohet dhe caktohet në pozicionin përkatës në matricën e zhurmës.

Shpërndarjet janë: Rayleigh, Exponential, LogNormal, Salt&Pepper dhe Erlang. Nëse ofrohet një lloj shpërndarjeje i pavlefshëm, ipet një gabim për të njoftuar përdoruesin për zgjedhjen e pavlefshme.

Së fundi, matrica e zhurmës së gjeneruar kthehet si rezultati i funksionit.

```

1. function retval = Zhurma(shperndarja, x, y, z, k, s, a, b)
2.     % Generate noise based on the specified distribution
3.     noise = zeros(z-x+1, k-y+1);
4.     for i = x:z
5.         for j = y:k
6.             if strcmp(shperndarja, 'Rayleigh')
7.                 noise(i-x+1, j-y+1) = raylrnd(a);
8.             elseif strcmp(shperndarja, 'Eksponenciale')
9.                 noise(i-x+1, j-y+1) = exprnd(a);
10.            elseif strcmp(shperndarja, 'LogNormal')
11.                noise(i-x+1, j-y+1) = lognrnd(a, b);
12.            elseif strcmp(shperndarja, 'Salt&Pepper')
13.                noise(i-x+1, j-y+1) = imnoise(0, 'salt & pepper', a);
14.            elseif strcmp(shperndarja, 'Erlang')
15.                noise(i-x+1, j-y+1) = gamrnd(a, b);
16.            else
17.                error(['Shperndarja ' shperndarja ' e zgjedhur nuk eshte valide!']);
18.            end
19.        end
20.    end
21. end
22. retval = noise;
23. endfunction

```

Pastaj për secilën shpërndarje, shënohet kodi në command view si vijon. Po paraqesim rastin për zhurmën Erlang. Kodi i dhënë demonstroi përdorimin e funksionit Zhurma për të gjeneruar zhurmë Erlang dhe për ta zbatuar atë në një imazh. Imazhi i përdorur në këtë shembull ngarkohet duke përdorur funksionin `imread`, ku imazhi 'lena.tif' lexohet dhe ruhet në variablin `img`.

Funksioni Zhurma thirret me parametrat që specifikojnë zhurmën Erlang, një rajon që mbulon të gjithë imazhin (1, 1, madhësia(`img`, 1), madhësia(`img`, 2)) dhe vlerat 0.8 për `a` dhe `b`. Matrica e zhurmës së gjeneruar normalizohet më pas duke e pjesëtuar me 255, duke siguruar që vlerat të jenë brenda intervalit të vlefshëm prej [0, 1].

Më pas, imazhi i zhurmshëm merret nga shumëzimi sipas elementeve të matricës së normalizuar të zhurmës me imazhin origjinal. Ky hap aplikon zhurmën Erlang në imazh, pasi intensiteti i çdo piksel shumëzohet me vlerën përkatëse të zhurmës.

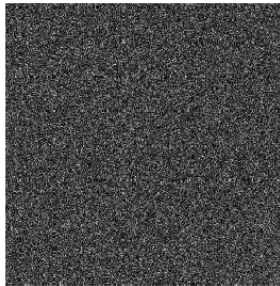
Imazhet që rezultojnë, duke përfshirë imazhin origjinal, zhurmën e krijuar Erlang dhe imazhin e zhurmshëm, shfaqen duke përdorur nënplotesa. Funksioni `imshow` përdoret për të shfaqur imazhet, ndërsa funksioni i titullit përdoret për të siguruar titujt e duhur për çdo nënplot.

```
1. img = imread('lena.tif');
2. zhurma = 'Erlang';
3. noise = Zhurma(zhurma, 1, 1, size(img, 1), size(img, 2), 1, .8, .8);
4. noise = noise/255;
5. noisyImage = noise.*img;
6.
7. subplot(1, 3, 1);
8. imshow(img, []);
10. title('Fotoja origjinale');
11.
12. subplot(1, 3, 2);
13. imshow(noise, []);
14. title('Zhurma Erlang');
15.
16. subplot(1, 3, 3);
17. imshow(noisyImage, []);
18. title('Imazhi me zhurme');
19. title('Imazhi me zhurme', 'Parent');
```

Fotoja origjinale



Zhurma Rayleigh



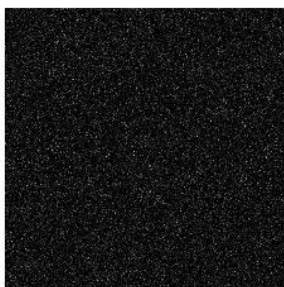
Imazhi me zhurme



Fotoja originale



Zhurma Eksponencial



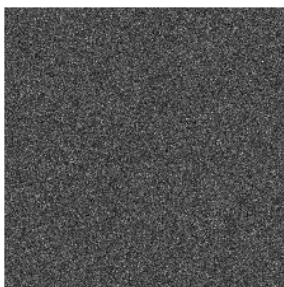
Imazhi me zhurme



Fotoja originale

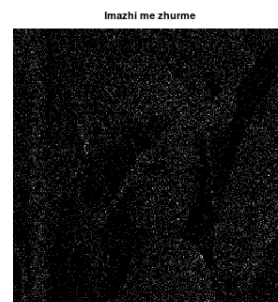
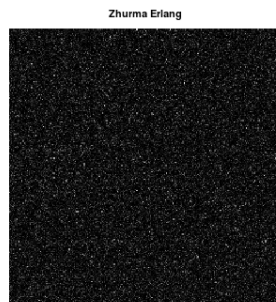
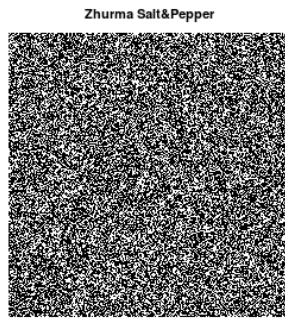
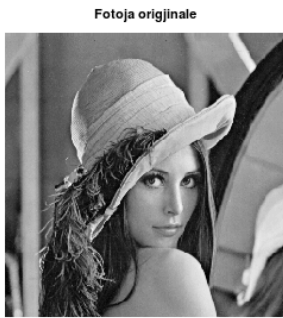


Zhurma LogNormal



Imazhi me zhurme





DETYRA 2

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

$$20 \cdot \cos((\pi \cdot x)/3 + (\pi \cdot y)/7) + 23 \cdot \sin((\pi \cdot x)/3 + (\pi \cdot y)/8)$$

Në fillim llogaritet matrica e zhurmës periodike e madhësisë të figurës origjinale, duke e kalkuluar me formulën e japur nga detyra. Pastaj shtohet zhurma në imazh. Pastaj bëhen transformimet Furie, pas asaj bëhet analiza e imazhit pas shiftit me transformimin Furie dhe gjeihen pikat e zhurmave. E bëjmë një maskë që heqë zhurmat dhe pastaj bëhet shumëzim një me një të maskës dhe figurës të transformuar. Pastaj bëhen transformimet inverse Furie. Dhe shfaqim imazhet relevante.

KODI

```
1. function y = Detyra3b (I)
2. % Ne fillim marrim gjatesine dhe gjeresine e I-s
3. [M,N]=size(I);
4. % Krijojme matricen A e cila do te mbaje zhurmen periodike
5. A = zeros(M,N);
6. for i=1:M
7. for j=1:N
```

```

8.    % Mbushja e A-se
10.   A(i,j)=(20*cos((pi*i)/3 + (pi*j)/7))+(23*sin((pi*i)/3 + (pi*j)/8));
11.   end
12.   end
13.   % Mbledhim I dhe A, ashtu qe te krijojme imazh me zhurme
14.   B = I + A;
15.   % Fillojme transformimet furie
16.   FTB = fft2(B);
17.   FTBSH=fftshift(FTB);
18.   % Pasi qe identifikojme zhurmat krijojme masken qe ben eliminimin e saj
19.   % Ne fillim krijohet maska e mbushur me 1
20.   F = ones(M,N);
21.   % pastaj heqim (bejme 0) pjeset ku jane zhurmat
22.   F(85:88, 110:114) = 0;
23.   F(171:173, 144:148) = 0;
24.   % Konbinojme masken me imazhin e modifikuar me tf
25.   C1 = FTBSH .* F;
26.   % Bejme transformimet inverse
27.   C2 = ifftshift(C1);
28.   C3 = ifft2(C2);
29.   % Marrim pjesen reale te imazhit final
30.   Final = uint8(real(C3));
31.   % shfaqim te gjitha imazhet qe nevojitet per spjegimin e funksionit
32.   % Sipas rradhes : imazhi origjinal, imazhi me zhurme, imazhi ku duhet te shihen
33.   zhurmat, imazhi final
34.   imshow(I), figure, imshow(B), figure, imshow(log(abs(FTBSH)),[]), figure, imshow(F),
35.   figure, imshow(Final);
36.   endfunction

```

REZULTATI

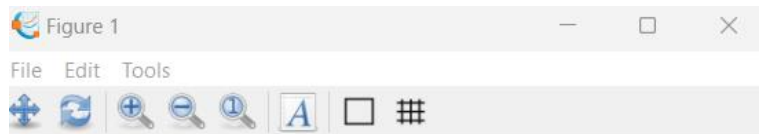
Kodin e mëposhtëm e shkruajmë në command window.

```

% Load the image
img = imread('lena.tif');
% Kalkulo dhe shfaq rezultatin
Detyra3b(I);

```

Disa figura do të shfaqen. Imazhi origjinal :



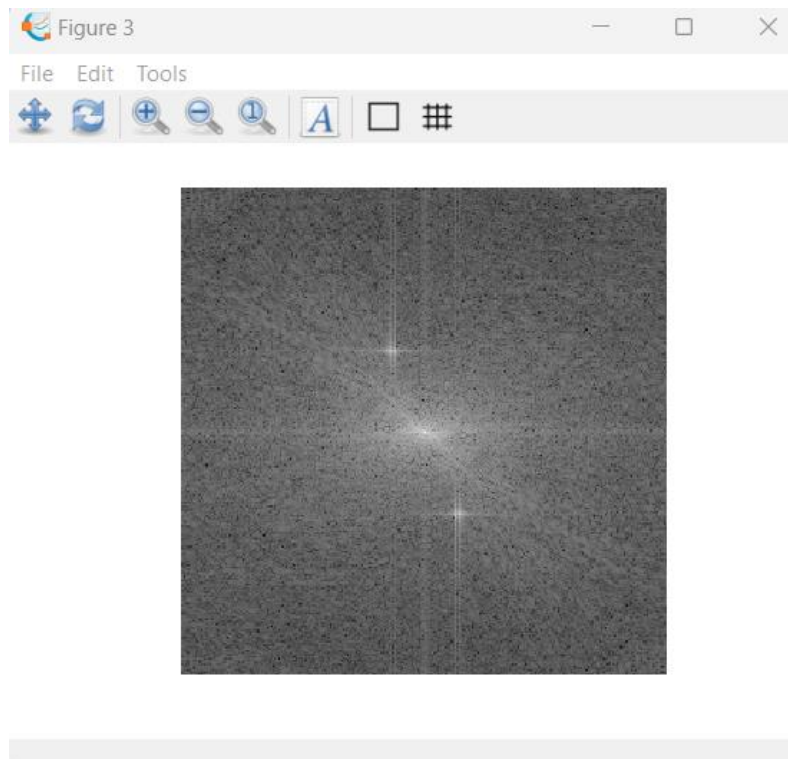
(13.125, 16.206)

Imazhi me zhurmë:

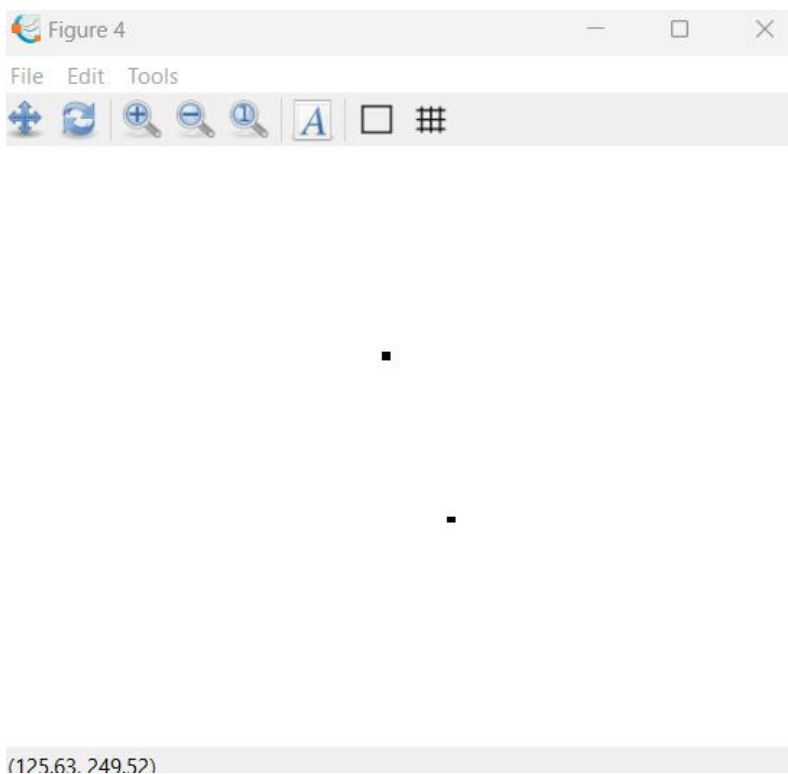


(0.062673, 0.24102)

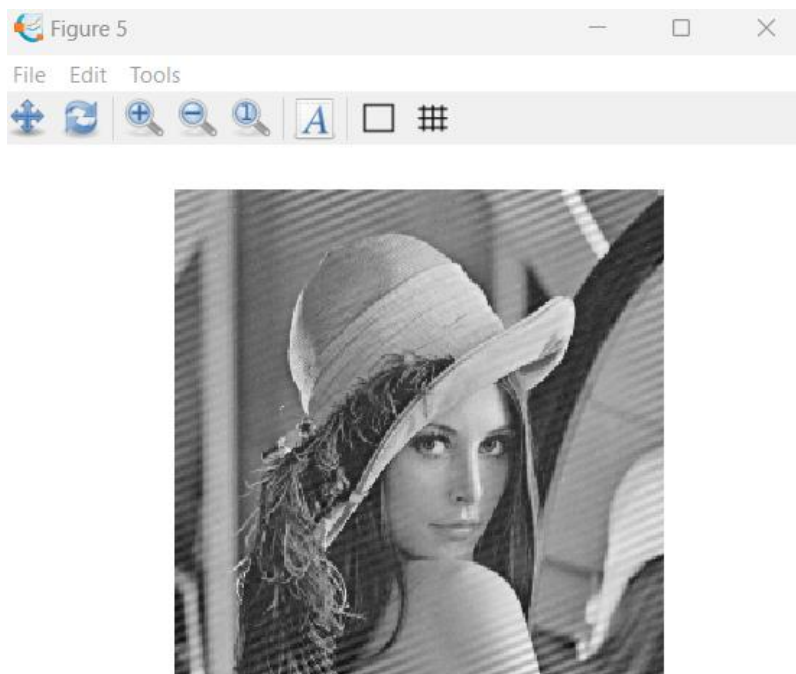
Imazhi pas transformimeve Furie:



Imazhi maskë:



Imazhi pas hekjes së zhurmës:



(81.081. 4.1244)

DETYRA 3

3) Krijoni filterin

$$H_g(u, v) = 1 - \exp \left\{ -\frac{1}{2} \left(\frac{D_1(u, v) D_2(u, v)}{D_0^2} \right) \right\}$$

ku $D_0 = (u_0, v_0)$ është qendra e njërës nga frekuencat që duhet të largohet (me gjithë të është edhe pika simetrike), $D_i(u, v) = \sqrt{\left(u - \frac{M}{2} + (-1)^i u_0\right)^2 + \left(v - \frac{N}{2} + (-1)^i v_0\right)^2}$, $i = 1, 2$ dhe M, N madhësia e imazhit. (4 pikë)

Zgjidhje

Në fillim merren parametrat e imazhit hyrës, pastaj kalkulohet formula e dhënë për secilin pixell të tij. Shembulli për rezultat është marrur për lena.tif (pa zhurmë, vetëm për demonstrim) dhe një piksell i rastësishëm është zgjedhur për parametrat d_0 , u_0 dhe v_0 .

KODI

- | | |
|----|--------------------------------------------------------------------------------------|
| 1. | function y = Detyra3c (I, d0, u0, v0) |
| 2. | % Njehere marrim madhesine dhe gjeretin e imazhit origjinal dhe krijojme nje matrice |
| 3. | boshe |

```

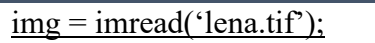
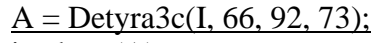
4. [M,N] = size(I);
5. for i=1:M
6.   for j=1:N
7.     A(i,j) = 1 - exp((-1/2)*((sqrt((i - (M/2) - u0)^2+(j - (N/2) - v0)^2)*sqrt((i - (M/2) +
8.     u0)^2+(j - (N/2) + v0)^2))/(d0^2)));
9.   end
10. end
11. end
12. y=A;
13. endfunction

```

REZULTATI

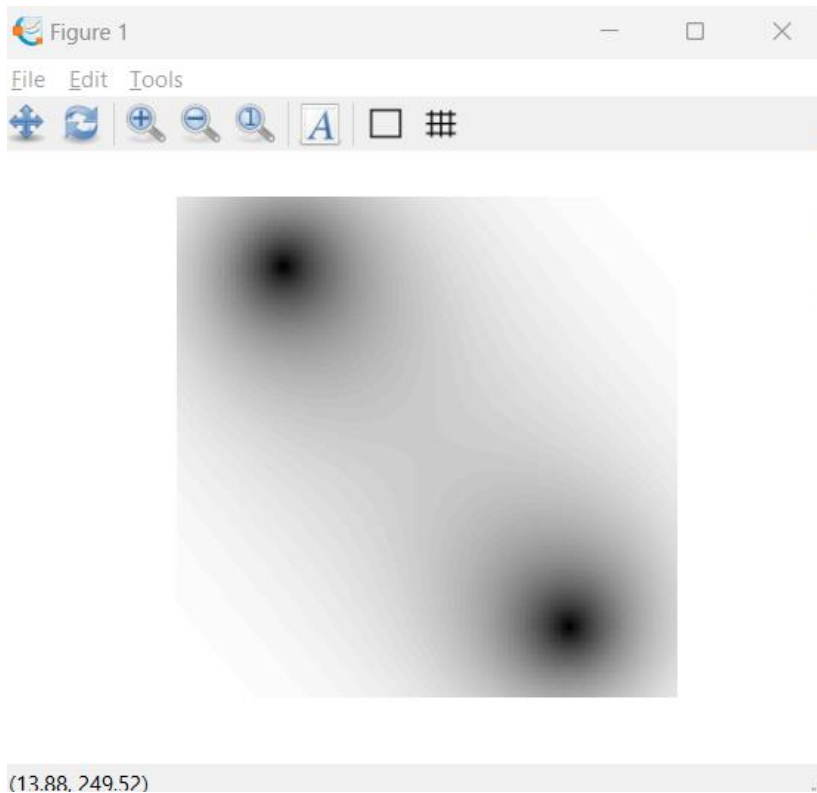
Kodin e mëposhtëm e shkruajmë në command window.

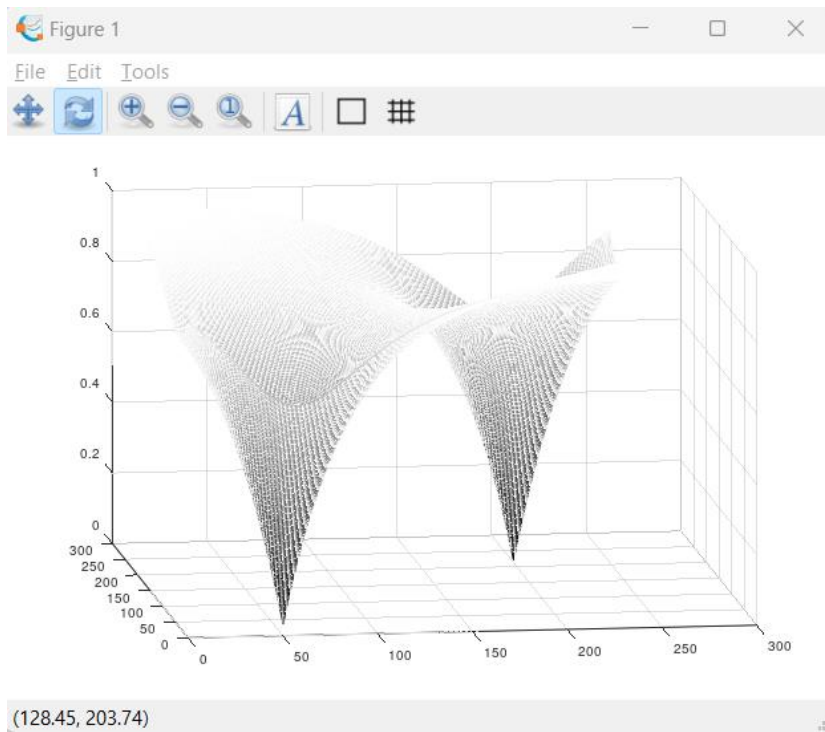
```


A = Detyra3c(I, 66, 92, 73);

mesh(A);

```

Dy figura do të shfaqen. Filteri 2D dhe 3D:





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 Figura 3. (4 pikë)



Figura 2. Regjioni binar origjinal



Figura 3. Regjioni pas veprimit me element strukturor në regjionin binar origjinal

Zgjidhje

KODI

```
1. SE=strel('line',90,90);
2. A =imopen(I,SE);
3. imshow(A,[]);
4.
5. SE = strel('line',133,180);
6. B =imopen(I,SE);
7. imshow(B,[]);
8.
10. C=A&B;
11. imshow(C,[]);
12.
13. %zvogelon width
14. se = strel('rectangle', [1, 17]);
15. eroded = imerode(C, se);
16. imshow(eroded,[]);
17.
18. %zvogelon height
19. se = strel('rectangle', [42, 1]);
20. eroded = imerode(eroded, se);
21. imshow(eroded,[]);
```

REZULTATI

