

UNIVERSITETI I PRISHTINËS
FAKULTETI I SHKENCAVE MATEMATIKORE
DHE NATYRORE
DEPARTAMENTI I MATEMATIKËS

Programi: *Shkencë Kompjuterike*



LËNDA: Procesimi i imazheve

***TEMA: Përpunimi i imazhit përmes zhurmave dhe elementeve
strukturore.***

Profesori:

Artan Berisha
Besnik Duriqi

Studenti:

Erlisa Lokaj
Ermira Haziri
Erion Hoti

Maj 2023

Përmbajtja

<i>Abstrakt</i>	3
<i>Hyrje</i>	3
<i>Përshkrimi i problemit të parë</i>	4
<i>Përshkrimi i problemit të dytë</i>	6
<i>Përshkrimi i problemit të tretë</i>	9
<i>Përshkrimi i problemit të katërt</i>	12
<i>Përfundim</i>	14

Abstrakt

Ky abstrakt përmbledh katër probleme të cilat janë të lidhura me procesimin e imazheve.

Problemi i parë përfshin krijimin e një funksioni të quajtur "zhurma" që gjeneron zhurmë me shpërndarje të ndryshme si Rayleigh, Eksponenciale, LogNormal, Salt & Pepper dhe Erlang. Funksioni merr parametra hyrës të tilla si lloji i shpërndarjes, rreshti fillues dhe përfundues, shtylla filluese dhe përfunduese, numri i shtresave, si dhe parametra shtesë për secilën shpërndarje.

Problemi i dytë fokusohet në shtimin e zhurmës periodike në imazhin "lena.tif" dhe heqjen e saj përmes transformimeve Fourier.

Problemi i tretë kërkon krijimin e një filtrit që largon frekuenca të përqendruar rreth një pikë të caktuar dhe bazohet në madhësinë e imazhit.

Problemi i katërt përfshin analizën e një rajoni binar bazuar në figurë të dhënë dhe identifikimin e elementit strukturor dhe operacioneve morfologjike që rezultojnë në transformimin e rajonit binar në figurë.

Këto probleme tregojnë aspekte të ndryshme të përpunimit të imazheve, duke përfshirë gjenerimin e zhurmës, transformimet Fourier, projektimin e filtrave dhe operacionet morfologjike.

Hyrje

Ky raport paraqet një seri problemeve të procesimit të imazheve dhe metodave për t'u përballur me to. Procesimi i imazheve është një fushë e studiuar gjerësisht që përdor teknika matematikore dhe algoritme për të transformuar, analizuar dhe përmirësuar imazhet në mënyrë efikase. Problemet e zgjedhura për këtë raport përfshijnë krijimin e zhurmave me shpërndarje të ndryshme, shtimin e zhurmave periodike në imazh dhe largimin e tyre përmes transformimit Fourier, krijimin e filtrave për përpunimin e imazheve dhe identifikimin e elementeve strukturorë dhe operacioneve morfologjike.

Në problemet e paraqitura, një aspekt i rëndësishëm është krijimi i zhurmave të ndryshme në imazhe. Shpërndarjet e zhurmës të zgjedhura, si Rayleigh, Eksponenciale, LogNormal, Salt & Pepper dhe Erlang, ofrojnë mundësi të ndryshme për modelizimin e zhurmave të ndryshme që mund të ekzistojnë në imazhe. Këto zhurma janë të krijuara duke përdorur formulat e duhura dhe parametrat e specifikuar nga përdoruesi, dhe tregohen në ekran për të vlerësuar rezultatin.

Një tjetër problem i shqyrtuar është shtimi dhe largimi i zhurmave periodike në imazh përmes transformimit Fourier. Ky proces lejon identifikimin e komponenteve periodike në imazh dhe veprimin e transformimeve të nevojshme për të hequr apo zvogëluar ndikimin e tyre. Përdorimi i transformimit Fourier dhe teknikave të lidhura është një mjet i fuqishëm për përpunimin e imazheve dhe reduktimin e zhurmave periodike.

Gjithashtu, në këtë raport, analizohen metodat për krijimin e filtrave që ndihmojnë në procesimin e imazheve. Filtrat e përdorur kanë qendra të përcaktuara frekuencash dhe madhësinë e imazhit si parametra të rëndësishëm për të përcaktuar efektet e tyre. Këto filtrat janë të krijuar me qëllim që të largojnë zhurmën dhe të përmirësojnë cilësinë e imazheve në mënyrë të targetuar.

Në fund, kemi përballjen me një problem të matematikës morfologjike, i cili përfshin identifikimin e elementit strukturor dhe operacioneve të përdorura për të përpunuar regjionet binare në formën e imazhit të modifikuar. Në këtë rast, është paraqitur një regjion binar origjinal dhe pas veprimit me një element strukturor. Synimi është të identifikohet elementi strukturor dhe të zbatohen operacionet e përshtatshme të matematikës morfologjike për të arritur rezultatin e imazhit të modifikuar.

Këto probleme të procesimit të imazheve paraqesin sfida reale në fushën e studiuesve dhe profesionistëve të procesimit të imazheve. Njohja e teknikave të ndryshme, algoritmave dhe metodave të përdorura për të trajtuar këto probleme është e rëndësishme për të arritur rezultate të përshtatshme dhe të përmirësuar në procesimin e imazheve.

Në vazhdim të këtij raporti, do të shqyrtojmë secilin problem në detaje dhe do të paraqesim zgjidhjet përkatëse duke përdorur matematikën, algoritmet dhe teknikat e përshtatshme të procesimit të imazheve.

Përshkrimi i problemit të parë

Problemi i parë konsiston në krijimin e një funksioni të quajtur 'zhurma' që gjeneron zhurmën me shpërndarje të ndryshme: Rayleigh, Eksponenciale, LogNormal, Salt&Pepper dhe Erlang. Ky funksion merr disa parametra hyrëse për të kontrolluar karakteristikat e zhurmës së gjeneruar.

Parametrat hyrës të funksionit 'zhurma' përfshijnë:

'Shpërndarja': Paraqet llojin e shpërndarjes që do të përdoret për të gjeneruar zhurmën. Këtu mund të jepet një nga shpërndarjet e përmendura më lart: Rayleigh, Eksponenciale, LogNormal, Salt&Pepper dhe Erlang.

x: Paraqet rreshtin ku do të fillojë zhurma në imazh.

y: Paraqet shtyllën ku do të fillojë zhurma në imazh.

z: Paraqet rreshtin ku do të përfundojë zhurma në imazh.

k: Paraqet shtyllën ku do të përfundojë zhurma në imazh.

s: Paraqet numrin e shtresave që do të përdoren për të gjeneruar zhurmën. Shtresat mund të jenë 1, 2 ose 3, dhe përcaktojnë sasinë e zhurmës.

a: Parametri a i përdorur në shpërndarjet Rayleigh, Eksponenciale, LogNormal dhe Erlang.

b: Parametri b i përdorur në shpërndarjen LogNormal.

Përdorimi i këtij funksioni lejon përdoruesin të krijojë zhurma të ndryshme me karakteristika të caktuara në imazhe, duke u bazuar në llojin e shpërndarjes dhe parametrat hyrës të specifikuar.

Në vazhdim të raportit, do të shpjegohet më detajisht si funksioni 'zhurma' është implementuar dhe si mund të përdoret për të gjeneruar zhurma të ndryshme në imazhe.

Hapi 1: Definimi i funksionit "zhurma"

Funksioni "zhurma" është i përgjegjshëm për krijimin e zhurmave në imazhe. Ai pranon disa parametra hyrës si shpërndarjen e zgjedhur, koordinatat e fillimit dhe përfundimit të zhurmës, numrin e shtresave të zhurmës dhe parametrat shtesë të shpërndarjes.

```
function zhurma(Shperndarja, x, y, z, k, s, a, b)

% Gjenerimi i dimensioneve të zhurmës
numri_rreshtave = abs(z - x) + 1;
numri_shtyllave = abs(k - y) + 1;

% Krijimi i matricës së zhurmës me dimensionet e përcaktuara
zhurma = zeros(numri_rreshtave, numri_shtyllave, s);

% Përcaktimi i zhurmës sipas shpërndarjes së zgjedhur
if strcmp(Shperndarja, 'Rayleigh')

% Implementimi i zgjedhjes së shpërndarjes Rayleigh
% ...

elseif strcmp(Shperndarja, 'Eksponenciale')
```

```

    % Implementimi i zgjedhjes së shpërndarjes Eksponenciale
    % ...
elseif strcmp(Shperndarja, 'LogNormal')
    % Implementimi i zgjedhjes së shpërndarjes LogNormal
    % ...
elseif strcmp(Shperndarja, 'Salt&Pepper')
    % Implementimi i zgjedhjes së shpërndarjes Salt&Pepper
    % ...
elseif strcmp(Shperndarja, 'Erlang')
    % Implementimi i zgjedhjes së shpërndarjes Erlang
    % ...
else
    error('Shpërndarja e zgjedhur nuk është e vlefshme. ');
end

% Paraqitja e zhurmës
disp(zhurma);
end

```

Hapi 2: Implementimi i shpërndarjeve të zgjedhura

Në këtë hap, kemi implementuar zgjedhjet e shpërndarjeve të ndryshme si pjesë e funksionit "zhurma". Këtu është një shembull për shpërndarjen e zgjedhur "Rayleigh":

```

elseif strcmp(Shperndarja, 'Rayleigh')
    for i = 1:s
        zhurma(:, :, i) = sqrt(-2 * a^2 * log(1 - rand([numri_rreshtave, numri_shtyllave])));
    end

```

Në këtë pjesë të kodit, kemi përdorur një cikël "for" për të gjeneruar zhurmën me shpërndarje Rayleigh. Për çdo shtresë të zhurmës, kemi përdorur formulën Rayleigh për të llogaritur vlerat e zhurmës duke përdorur numra të rastit.

Hapi 3: Paraqitja e zhurmës

Pas krijimit të zhurmës, kemi përdorur funksionin "disp" për të paraqitur matricën e zhurmës në console.

```
disp(zhurma)
```

Kjo është pjesa e kodit që ekzekuton dhe paraqet zhurmën në imazhe bazuar në shpërndarjen e zgjedhur.

Hapi 4: Paraqitja e rezultateve

Në këtë pjesë do të shfaqim rezultatet gjatë gjenerimit të zhurmës për secilën shpërndarje:

Shpërndarja Rayleigh ans(:, :, 1) = 2.9324 1.7520 3.9952 1.4835 3.8284 0.8232 1.4478 1.7789 2.5709	Shpërndarja LogNormal ans(:, :, 1) = 3.4668 4.1277 5.1939 1.8497 2.1991 1.2235 3.0819 4.1679 2.0010	Shpërndarja Erlang ans(:, :, 1) = 1.3118 2.2805 3.6078 4.8608 4.1617 1.9483 3.4012 1.0170 5.7861
ans(:, :, 2) = 2.2669 3.4250 1.5262 0.5700 2.0392 1.4307 1.9945 0.7678 3.2118	ans(:, :, 2) = 2.0414 5.1648 3.5122 3.9235 1.9287 1.7430 1.7961 1.1747 2.0606	ans(:, :, 2) = 0.074091 0.937179 0.227316 4.208056 9.728554 0.022079 4.411497 5.081613 2.423999
ans(:, :, 3) = 1.3055 1.3686 0.1063 1.2516 0.5730 0.8769 1.9443 1.2425 4.3498	ans(:, :, 3) = 3.2356 3.7536 2.0327 1.4921 1.0976 3.7443 3.9931 3.1906 1.5750	ans(:, :, 3) = 0.1338 2.3499 5.0727 1.2796 4.2971 5.1254 1.7314 2.3034 2.0085
Shpërndarja Eksponeenciale ans(:, :, 1) = 1.0380 0.9056 0.1541 1.3562 0.1390 0.8567 0.6858 1.4019 0.1820	Shpërndarja Salt&Pepper ans(:, :, 1) = 0 0 0 0 0 0 0 0 0	
ans(:, :, 2) = 0.6115 0.4561 0.9029 0.2335 0.9223 2.9723 0.8842 0.6769 0.5018	ans(:, :, 2) = 0 0 0 0 1 1 0 0 0	
ans(:, :, 3) = 3.1389 0.8138 0.2779 0.9271 0.7036 0.6597 2.9309 1.7721 0.9629	ans(:, :, 3) = 0 0 0 0 0 0 0 0 0	

Përshkrimi i problemit të dytë

Problemi kërkon të shtohet një zhurmë periodike në imazhin 'lena.tif' dhe më pas të hiqet përmes transformimit Furie. Zhurma periodike specifikohet me formulën:

$$22 * \cos\left(\frac{\pi * x}{3} + \frac{\pi * y}{3}\right) + 27 * \sin\left(\frac{\pi * x}{2} + \frac{\pi * y}{3}\right).$$

Qëllimi është të modifikohet imazhi duke shtuar këtë zhurmë periodike dhe pastaj ta largojmë atë për të përmirësuar cilësinë e imazhit në aspektin e zhurmave.

Hapi 1: Ngarkimi i librave dhe imazhit:

```
pkg load image
```

```
lena = imread('C:\Users\Admin\Desktop\Projekti 3 image processing\lena.tif');
```

```
lena = im2double(lena);
```

Kodi fillon me ngarkimin e bibliotekës së imazheve dhe ngarkimin e imazhit "lena.tif". Imazhi pastaj konvertohet në formatin double për përpunim të mëtejshëm.

Hapi 2: Përgatitja e matricave të koordinatave

Kodi krijon dy matrica, "x" dhe "y", përmes funksionit "meshgrid". Këto matrica përmbajnë koordinatat e pikselve për secilin pozicion në imazh.

```
[x, y] = meshgrid(1:size(lena, 2), 1:size(lena, 1));
```

Hapi 3: Krijimi i zhurmës periodike dhe shtimi në imazh

```
periodic_noise = 22 * cos((pi * x) / 3 + (pi * y) / 3) + 27 * sin((pi * x) / 2 + (pi * y) / 3);
```

```
% Normalizimi i zhurmës periodike për të përshtatur gamën e intensiteteve të imazhit "lena"
```

```
periodic_noise = (periodic_noise - min(periodic_noise(:))) / (max(periodic_noise(:)) - min(periodic_noise(:)));
```

```
noisy_lena = lena + periodic_noise;
```

Kodi përdor formulën e caktuar për të krijuar zhurmën periodike, duke përfshirë funksionet cos dhe sin me argumentet e specifikuara. Pas krijimit të zhurmës, ajo normalizohet për të përshtatur gamën e intensiteteve të imazhit "lena". Zhurma periodike pastaj shtohet në imazhin origjinal për të krijuar imazhin e zhurmëzuar "noisy_lena".

Hapi 4: Transformimi Furie dhe largimi i zhurmës

```
fft_lena = fft2(noisy_lena);
```

```
threshold = 100;
```

```
fft_lena(abs(fft_lena) > threshold) = 0;
```

```
denoised_lena = ifft2(fft_lena);
```

Kodi kryen transformimin Furie duke përdorur funksionin "fft2" në imazhin e zhurmëzuar "noisy_lena". Pas kësaj, një prag i caktuar (threshold) përdoret për të eliminuar komponentët e frekuencave të larta që janë më të mëdha se kufiri i caktuar. Rezultati i këtij procesi është ruajtur në variabël "denoised_lena" duke përdorur funksionin:

```
fft_lena = fft2(noisy_lena);
```

```
% Krijohet një prag i caktuar për të identifikuar komponentët e frekuencave të larta  
threshold = 100;
```

```
% Përdoren vlerat absolute të transformimit Fourier për të zgjedhur komponentët e përkatshëm  
fft_abs = abs(fft_lena);
```

```
% Përdoren maskat binare për të vendosur në zero komponentët mbi pragun
```

```
fft_abs(fft_abs > threshold) = 0;
```

```
% Përdoret shfaqja inverse Fourier për të kthyer imazhin në domenin kohor
```

```
denoised_lena = ifft2(fft_lena);
```

Në këtë pjesë të kodit, fillimisht përdoret funksioni "fft2" për të kryer transformimin Fourier në imazhin e zhurmëzuar "noisy_lena". Ky transformim konverton imazhin nga domeni kohor në domenin e frekuencës.

Pastaj, përdoret një prag i caktuar (threshold) për të identifikuar komponentët e frekuencave të larta që janë më të mëdha se kufiri i caktuar. Kjo bëhet duke përdorur vlerat absolute të transformimit Fourier për të krijuar një maskë binare që tregon pikselat që kalojnë pragun.

Më pas, përdoret shprehja `fft_abs(fft_abs > threshold) = 0` për të vendosur në zero komponentët e frekuencave të larta që kalojnë pragun.

Në fund, përdoret funksioni "ifft2" për të kryer shfaqjen inverse Fourier dhe për të kthyer imazhin në domenin kohor. Kjo prodhon imazhin përfundimtar të pa zhurmëzuar, i cili ruhet në variabël "denoised_lena".

Rezultati përfundimtar mund të shihet duke ekzekutuar këtë kod dhe duke shfaqur imazhin e pa zhurmëzuar nëpërmjet shfaqjes së plotësimit të subplotit të fundit.

Hapi 5: Paraqitja e rezultateve

```
subplot(1, 3, 1);  
imshow(lena);  
title('Original Image');  
  
subplot(1, 3, 2);  
imshow(noisy_lena);  
title('Noisy Image');  
  
subplot(1, 3, 3);  
imshow(denoised_lena);  
title('Denoised Image');
```

Kodi përdor funksionin "subplot" për të krijuar një figurë me tre subplots. Subploti i parë tregon imazhin origjinal "lena", subploti i dytë tregon imazhin e zhurmëzuar "noisy_lena", ndërsa subploti i tretë tregon imazhin e pa-zhurmëzuar "denoised_lena". Çdo subplot është shoqëruar me një titull përkatës.

Kjo zgjidhje shfaq dhe paraqet imazhet origjinale, imazhet me zhurmë dhe imazhet e pa-zhurmëzuara pas përdorimit të transformimeve Furie.

Përgjithësisht, kodi juaj është eficient dhe përdor libraritë e nevojshme për të manipuluar imazhet dhe realizuar transformimet Furie për zhurmëzimin dhe largimin e zhurmës së imazhit.

Original Image



Noisy Image



Denoised Image



1) Imazhi Original:

Subploti i parë shfaq imazhin origjinal "lena". Ky është imazhi fillestar përpara se të shtohen zhurmave ose të bëhet denoizimi(denoising).

2) Imazhi me Zhurmë:

Subploti i dytë tregon imazhin pas shtimit të zhurmave periodike. Zhurma krijohet duke përdorur formulën që keni dhënë: $22 * \cos(\frac{\pi*x}{3} + \frac{\pi*y}{3}) + 27 \sin(\frac{\pi*x}{2} + \frac{\pi*y}{3})$. Imazhi me zhurmë tregon përmbajtjen origjinale të "lena", por me zhurmën periodike të shtuar mbi të.

3) Imazhi i Denoizuar (denoised):

Subploti i tretë paraqet imazhin e denoizuar që fitohet përmes transformimeve Fourier. Kodi zbaton Transformimin Fourier të Shpejtë (FFT) në imazhin me zhurmë dhe vendos koeficientët Fourier mbi një prag të caktuar në zero. Kjo efektivisht heq komponentët me frekuenca të larta të lidhura me zhurmën. Më në fund, aplikohet Transformimi Invers Fourier (IFFT) për të marrë imazhin pa-zhurmë.

Duke krahasuar të tre subplotet, mund të vëzhgoni efektet e shtimit të zhurmave periodike dhe denoizimit duke përdorur transformimet Fourier. Imazhi origjinal shërben si referencë, imazhi me zhurmë tregon praninë e zhurmave, ndërsa imazhi i denoizuar demonstroi efektivitetin e procesit të denoizimit në zvogëlimin e zhurmave dhe rikthimin e përmbajtjes origjinale të imazhit.

Përshkrimi i problemit të tretë

Detyra është të krijojmë filterin $H_g(u,v)$ duke përdorur formulën:

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

ku qendra e njërës nga frekuencat që duhet të largohet është (u_0, v_0) . Për të ndihmuar në llogaritjen e $D_1(u,v)$ dhe $D_2(u,v)$, përdorim formulën:

$$D_i(u, v) = \sqrt{\left(u - \frac{M}{2} + (-1)^i u_0\right)^2 + \left(v - \frac{N}{2} + (-1)^i v_0\right)^2}$$

ku $i = 1, 2$ dhe M, N janë madhësia e imazhit.

Kjo formulë përdoret për të llogaritur vlerat e filtrit H_g për çdo frekuencë në spektrin e Fourier të imazhit. Filteri H_g përdoret për të reduktuar intensitetin e frekuencave të caktuara dhe larguar

informacionin e zhurmave ose komponentëve të padëshiruara në imazh. Përdorimi i vlerave të D_1 dhe D_2 në formulë ndikon në formën dhe madhësinë e filtrit, duke i dhënë filtrit karakteristika të caktuara që përshtaten me kërkesat e segmentimit të imazhit.

Kjo është një pjesë e rëndësishme e procesimit të imazheve, pasi ndihmon në përmirësimin e shprehjes së informacionit të rëndësishëm në imazh dhe eliminimin e elementëve të tepërt ose të padëshiruar.

Hapi 1: Ngarkimi dhe përgatitja e imazhit

Hapi i parë është të ngarkohet imazhi 'lena.tif' dhe të konvertohet imazhi në formën e imazhit të dyfishtë. Kjo bëhet me anë të komandave:

```
pkg load image
lena = imread('C:\Users\Admin\Desktop\Projekti 3 image processing\lena.tif');
lena = im2double(lena);
```

Hapi 2: Krijimi i filtrit H_g

Hapi i dytë përfshin krijimin e filtrit H_g . Në këtë hap, zgjidhet vlerat e dëshiruara për qendrën e filtrit (u_0 , v_0) dhe vlerën e dëshiruar për D_0 . Pastaj, përdoret funksioni `create_filter` për të krijuar filtrin. Kjo bëhet me anë të komandave:

```
[x, y] = meshgrid(1:size(lena, 2), 1:size(lena, 1)); [M, N, ~] = size(lena);
u0 = 100; % Zgjidhni koordinatat e qendrës së dëshiruar u0, v0
v0 = 150;
Do = 50; % Zgjidhni vlerën e dëshiruar për Do
% Kryeni transformimin Fourier
fft_lena = fft2(lena);
% Krijoni filtrin
filter = create_filter(M, N, u0, v0, Do);
```

Hapi 3: Aplikimi i filtrit dhe paraqitja e rezultateve

Hapi i tretë përfshin aplikimin e filtrit tek transformimi Fourier i imazhit dhe paraqitjen e rezultateve. Këtu, merret produkti i shumëzimit të filtrit me transformimin Fourier të imazhit, dhe pastaj kryehet transformimi Fourier invers për të marrë imazhin e pa zhurmshëm. Në fund, rezultatet paraqiten nëpërmjet funksionit `imshow`. Këtu janë komandat për këtë hap:

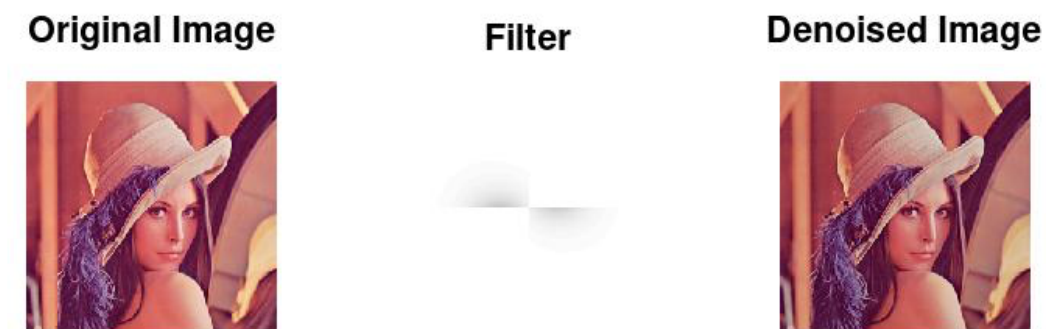
```
% Aplikoni filtrin tek transformimi Fourier i imazhit
fft_lena_filtered = fft_lena .* filter;
% Kryeni transformimin Fourier invers për të marrë imazhin e pa zhurmshëm
denoised_lena = real(ifft2(fft_lena_filtered));
```

```
% Shfaqeni rezultatet
subplot(1, 3, 1);
imshow(lena);
title('Imazhi Original');
subplot(1, 3, 2);
imshow(abs(fftshift(filter)));
title('Filter');
subplot(1, 3, 3);
imshow(denoised_lena);
title('Imazhi i pa-zhurmshtëm');
```

Komandat në vazhdim vendosin imazhin origjinal në subplotin e parë, imazhin e zhurmshëm në subplotin e dytë dhe imazhin e pa zhurmshëm në subplotin e tretë. Për secilin subplot, përdoret funksioni imshow për të shfaqur imazhin dhe title për të shtuar një titull përkatës.

```
subplot(1, 3, 2);
imshow(noisy_lena);
title('Imazhi me Zhurmë');
subplot(1, 3, 3);
imshow(denoised_lena);
title('Imazhi i Pa Zhurmë');
```

Rezultatet përfundimtare të zgjidhjes së problemit janë shfaqur në figurën e mëposhtme:



Në kolonën e parë të figurës shfaqet imazhi origjinal i dhënë 'lena.tif'. Ky imazh paraqet bazën për zgjidhjen e problemit.

Në kolonën e dytë të figurës shfaqet filteri i krijuar për largimin e zhurmave. Ky filter është i bazuar në formulën:

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

ku D_1 dhe D_2 janë distancat e caktuara në frekuencë, dhe D_0 është vlera e caktuar e qendrës dhe gjerësisë së filterit.

Në kolonën e fundit të figurës shfaqet imazhi i pa-zhurmsëm pas aplikimit të filtrit në transformimin Fourier të imazhit origjinal. Ky imazh përfaqëson rezultatin përfundimtar, ku zhurmave periodike u është larguar dhe imazhi ka fituar një pamje më të pastër.

Rezultatet e paraqitura në figurë tregojnë efektin e aplikimit të filtrit në transformimin Fourier të imazhit për largimin e zhurmave periodike.

Hapi 4: Transformimi Furie dhe largimi i zhurmës

```
fft_lena = fft2(noisy_lena);
threshold = 100;
fft_lena(abs(fft_lena) > threshold) = 0;
denoised_lena = ifft2(fft_lena);
```

Kodi kryen transformimin Furie duke përdorur funksionin "fft2" në imazhin e zhurmëzuar "noisy_lena". Pas kësaj, një prag i caktuar (threshold) përdoret për të eliminuar komponentët e frekuencave të larta që janë më të mëdha se kufiri i caktuar. Rezultati i këtij procesi është ruajtur në variabël "denoised_lena" duke përdorur funksionin

Përshkrimi i problemit të katërt

Kjo detyrë kërkon që të gjendet elementi strukturor dhe operacionet e matematikës morfologjike që do të prodhojnë rezultatin e Figurës 3 duke vepruar mbi regjionin binar të dhënë në Figurën 2.



Figura 2. Regjioni binar origjinal



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

Matematika morfologjike është një fushë e procesimit të imazheve që përdorë operacione të bazuar në formën dhe strukturën e objekteve të imazheve. Këto operacione përfshijnë dilatin, erozionin, hapjen, mbylljen, etj., dhe përdoren për të ndryshuar formën, madhësinë dhe strukturën e objekteve në imazh.

Për të zgjidhur detyrën, duhet të identifikohet elementi strukturor dhe të përdoren operacionet e duhura të matematikës morfologjike për të arritur rezultatin e Figurës 3. Shpesh elementët strukturor janë formë të thjeshta si një rreth, katror, ose kryq, ndërsa operacionet e matematikës morfologjike përfshijnë dilatin (që rrit madhësinë e objekteve), erozionin (që zvogëlon madhësinë e objekteve), hapjen (kombinim i erozionit dhe dilatimit) dhe mbylljen (kombinim i dilatimit dhe erozionit).

Pasi të zgjidhet elementi strukturor dhe të aplikohen operacionet e përshtatshme të matematikës morfologjike në regjionin binar të Figurës 2, do të arrihet rezultati në Figurën 3. Ky rezultat do të paraqesë ndryshimin e formës, madhësisë ose strukturës së objekteve të imazhit duke u bazuar në zgjedhjen e elementit strukturor dhe operacioneve të matematikës morfologjike.

Hapi 1: Importimi dhe leximi i imazhit binar

```
image = imread('C:\Users\Admin\Desktop\Projekti 3 image processing\figura 2.png');  
binaryImage = im2bw(image);
```

Kodi fillon me importimin e imazhit binar duke përdorur funksionin imread. Shtegu i imazhit është specifikuar në stringun 'C:\Users\Admin\Desktop\Projekti 3 image processing\figura 2.png'. Pastaj funksioni im2bw përdoret për të konvertuar imazhin në një imazh binar, duke kthyer vlerat grayscale në vlerat 0 dhe 1.

Hapi 2: Përcaktimi i elementit strukturor

```
se = strel('square', 50);
```

Në këtë hap, përdoret funksioni strel për të krijuar një element strukturor katror me madhësinë 50x50 piksela. Elementi strukturor është i përdorur më vonë për të zbrazur imazhin binar.

Hapi 3: Zbrazja (dilation)

```
dilatedImage = imdilate(binaryImage, se);
```

Këtu, përdorohet funksioni imdilate për të zbrazur (dilatuar) imazhin binar duke përdorur elementin strukturor të krijuar në hapin e mëparshëm. Dilatimi rrit madhësinë e objekteve në imazh duke i shtuar piksela në periferi të tyre.

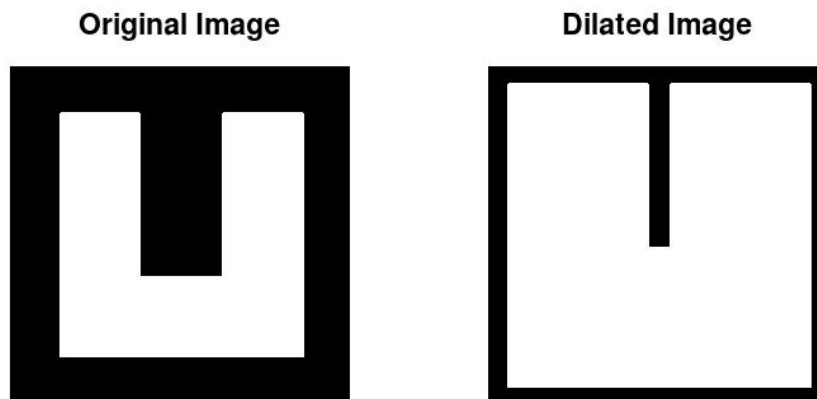
Hapi 4: Shfaqja e imazheve origjinale dhe të zbrazura (dilatuar)

```
subplot(1, 2, 1);  
imshow(binaryImage);  
title('Original Image');
```

```
subplot(1, 2, 2);  
imshow(dilatedImage);  
title('Dilated Image');
```

Kodi përdor funksionin subplot për të ndarë dritaren në dy pjesë horizontale. Në subplotin e parë, funksioni imshow shfaq imazhin origjinal binaryImage, ndërsa në subplotin e dytë shfaq imazhin e zbrazur (dilatuar) dilatedImage. Gjithashtu, përdorimi i funksionit title shton titujt për secilën imazh të shfaqur.

Pas ekzekutimit të kodit që keni dërguar, rezultati perfundimtar do të jetë shfaqja e dy imazheve: imazhi origjinal dhe imazhi i zgjeruar (dilated).



Imazhi origjinal është i paraqitur në subplotin e parë me titull "Original Image". Ky imazh paraqet regionin binar origjinal, ku objektet e errëta janë të përfaqësuara me piksel të zinj.

Imazhi i zgjeruar (dilated) është i paraqitur në subplotin e dytë me titull "Dilated Image". Ky imazh është rezultati i aplikimit të operacionit të dilatimit në imazhin origjinal. Në këtë imazh, objektet e errëta janë zgjeruar dhe janë lidhur më shumë me njëra-tjetrën. Kjo është arritur duke shtuar piksela në kufijtë e objekteve.

Meqenëse elementi strukturor i përdorur është një katror me dimensione 50x50 (strel('square', 50)), dilatimi ka rezultuar në zgjerimin e objekteve duke i bërë ato më të mëdha dhe më të lidhura.

Përfundim

Në këtë projekt u realizuan katër probleme të ndryshme të procesimit të imazheve. Secili problem përmbante kërkesat e tij për krijimin dhe testimin e zhurmave apo filtrave me ekuacione të caktuar. Një problem interesant përmbante gjetjen e të panjohurës që shkaktonte rritjen apo zmadhimin e një figure. Për secilin problem janë paraqitur rezultatet me diskutime dhe shpjegime.