**PROJECT-8**

**INVERSE GAUSSIAN**

**FILTER**

EE5356 Digital Image Processing

Dr. K. R. Rao

Submitted By:

Ashutosh Desai

UTA ID:1001388602

Email: - ashutoshrajesh.desai@mavs.uta.edu

SPRING-2018

**QUESTION:**



, *m, n=0,1,…..,N-1*

, *k, l=0,1,…..,N-1*

are 2D-DFT coefficients

(5.65), page 147

(5.66), page 148

2D-DFT and 2D-IDFT (unitary) are defined in (5.63) & (5.64), page 146.

is the unitary DFT matrix.

Inverse Gaussian filter (IGF) is defined by  where

0≤k,l≤(N/2)

Otherwise (7.48), page 258

IGF emphasizes high frequency coefficients and restores images blurred by atmospheric turbulence or other phenomenon modeled by Gaussian point spread functions.

 (7.47), page 256

2D-DFT coefficients are filtered in the frequency domain.

Apply IGF to the following images,

Lena (512×512), use 15707

Goldhill (256×256), use 28198

Boat (512×512), use 41694

Girl (512×512), use 34657

1) Display both original and filtered images.

2) Display the IGF (3D) using MATLAB commands,

surf(, ‘EdgeColor’, ‘none’);

colormap(hsv);

shading interp;

alpha(0.7);

grid on;

axis tight;

**MATLAB SCRIPT:**

clc;

clear all;

close all;

Input\_Img1=double(imread('D:\STUDY\DIP\Test img\lena512.bmp'));

Input\_Img2=double(imread('D:\STUDY\DIP\Test img\goldhill256.bmp'));

Input\_Img3=double(imread('D:\STUDY\DIP\Test img\girl512.bmp'));

Input\_Img4=double(imread('D:\STUDY\DIP\Test img\boat512.gif'));

Var\_1=15707;

Var\_2=28198;

Var\_3=34657;

Var\_4=41694;

Img\_1F=fft2(Input\_Img1);

Img\_2F=fft2(Input\_Img2);

Img\_3F=fft2(Input\_Img3);

Img\_4F=fft2(Input\_Img4);

New\_Img2=zeros(1,128);

for a=1:1:128

for b=1:1:128

New\_Img2(a,b)=exp(((a^2)+(b^2))/(2\*Var\_2));

end

end

New\_Img1=zeros(1,256);

New\_Img3=zeros(1,256);

New\_Img4=zeros(1,256);

for a=1:1:256

for b=1:1:256

New\_Img1(a,b)=exp(((a^2)+(b^2))/(2\*Var\_1));

New\_Img3(a,b)=exp(((a^2)+(b^2))/(2\*Var\_3));

New\_Img4(a,b)=exp(((a^2)+(b^2))/(2\*Var\_4));

if a<=128 && b>128

New\_Img2(a,b)=New\_Img2(a,257-b);

elseif a>128 && b<=128

New\_Img2(a,b)=New\_Img2(257-a,b);

elseif a>128 && b>128

New\_Img2(a,b)=New\_Img2(257-a,257-b);

end

end

end

for a=1:1:512

for b=1:1:512

if a<=256 && b>256

New\_Img1(a,b)=New\_Img1(a,513-b);

New\_Img3(a,b)=New\_Img3(a,513-b);

New\_Img4(a,b)=New\_Img4(a,513-b);

elseif a>256 && b<=256

New\_Img1(a,b)=New\_Img1(513-a,b);

New\_Img3(a,b)=New\_Img3(513-a,b);

N\_B(a,b)=New\_Img4(513-a,b);

elseif a>256 && b>256

New\_Img1(a,b)=New\_Img1(513-a,513-b);

New\_Img3(a,b)=New\_Img3(513-a,513-b);

New\_Img4(a,b)=New\_Img4(513-a,513-b);

end

end

end

surf(New\_Img1,'EdgeColor','none');

colormap(hsv);

shading interp;

alpha(0.7);

axis tight;

grid on;

title('512x512 Mask');

figure;

surf(New\_Img2,'EdgeColor','none');

colormap(hsv);

shading interp;

alpha(0.7);

axis tight;

grid on;

title('256x256 Mask');

Img1\_f=Img\_1F.\*New\_Img1;

Img2\_f=Img\_2F.\*New\_Img2;

Img3\_f=Img\_3F.\*New\_Img3;

Img4\_f=Img\_4F.\*New\_Img4;

Img1\_ifft=ifft2(Img1\_f);

Img2\_ifft=ifft2(Img2\_f);

Img3\_ifft=ifft2(Img3\_f);

Img4\_ifft=ifft2(Img4\_f);

figure;

subplot(1,2,1);imshow(uint8(Input\_Img1));title('Actual Image');

subplot(1,2,2);imshow(uint8(Img1\_ifft));title('Filtered Image ');

figure;

subplot(1,2,1);imshow(uint8(Input\_Img3));title('Actual Image');

subplot(1,2,2);imshow(uint8(Img3\_ifft));title('Filtered Image');

figure;

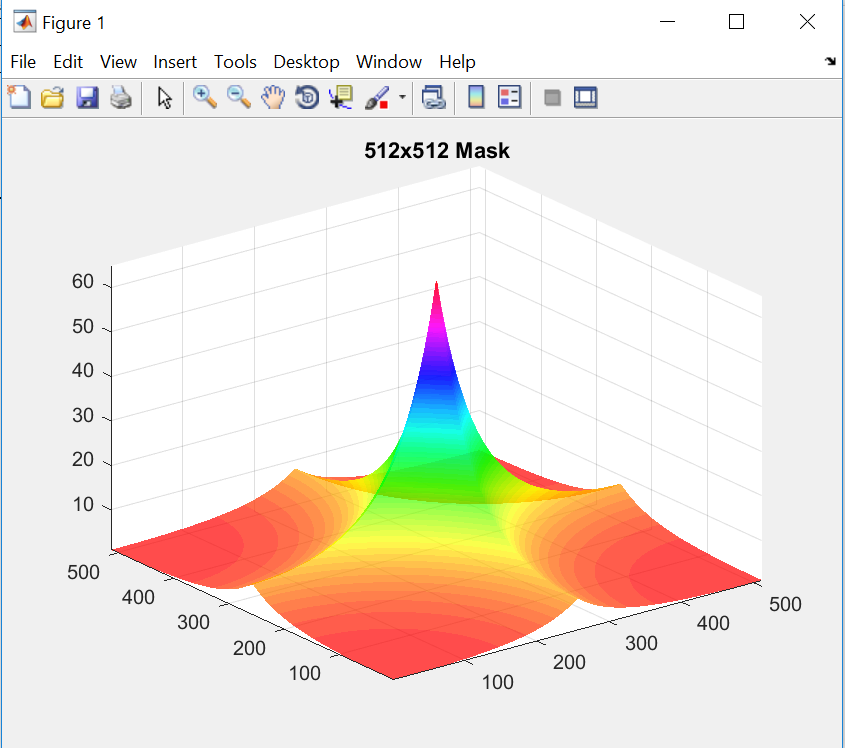
subplot(1,2,1);imshow(uint8(Input\_Img2));title('Actual Image');

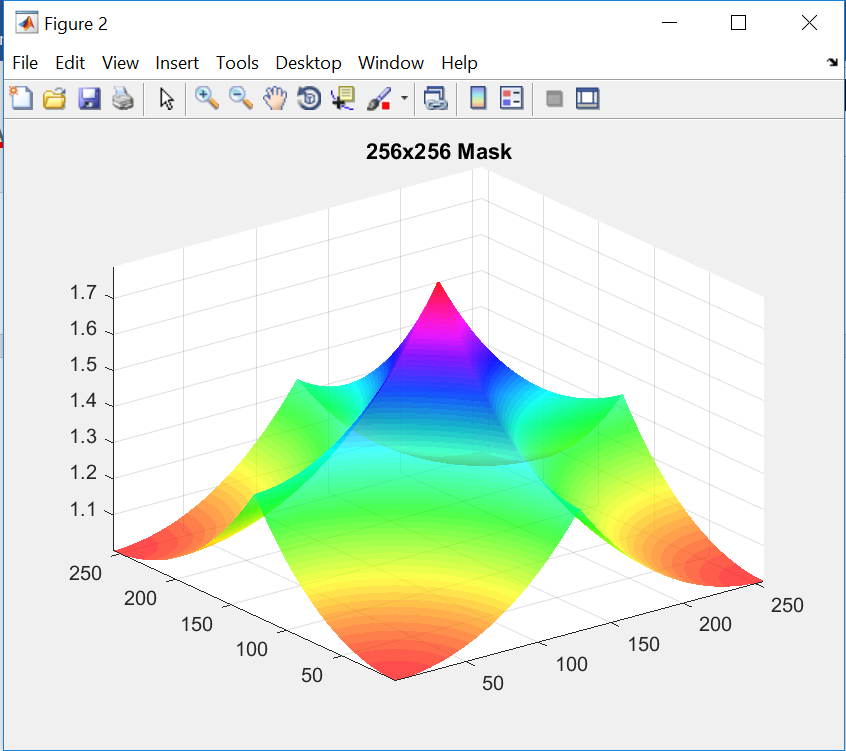
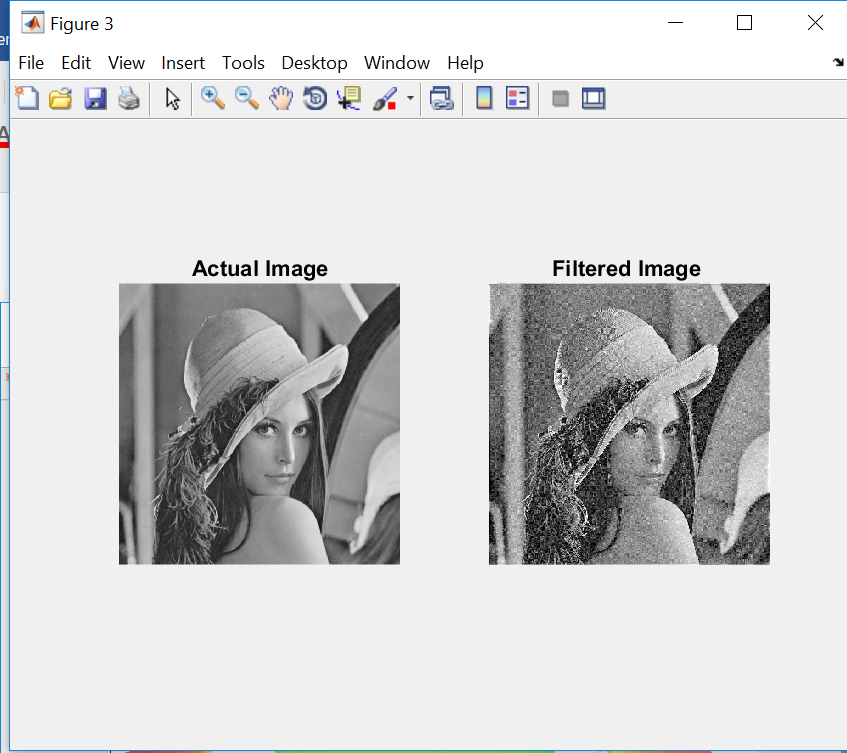
subplot(1,2,2);imshow(uint8(Img2\_ifft));title('Filtered Image');

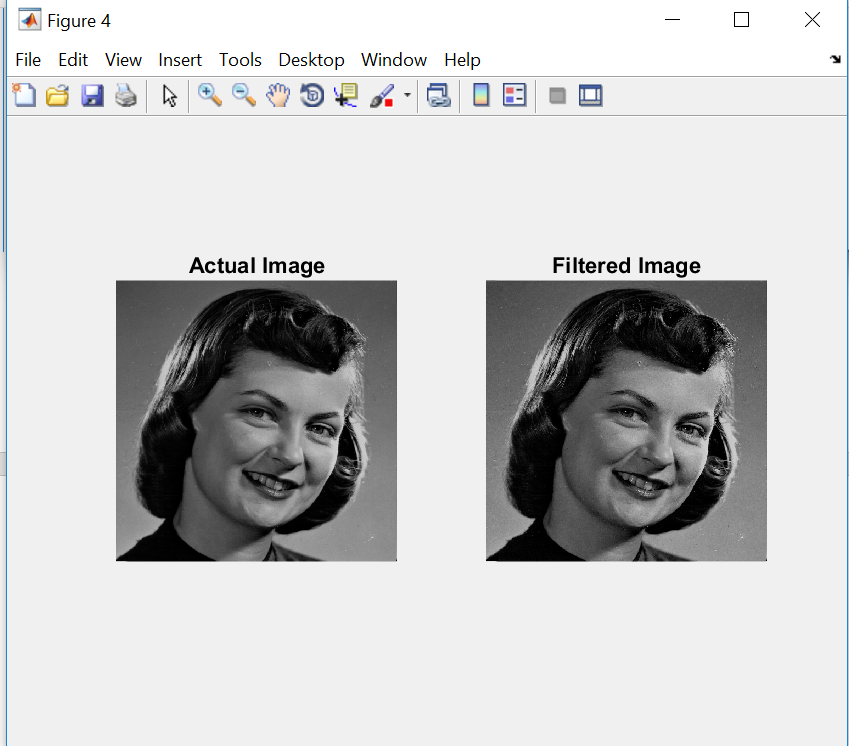
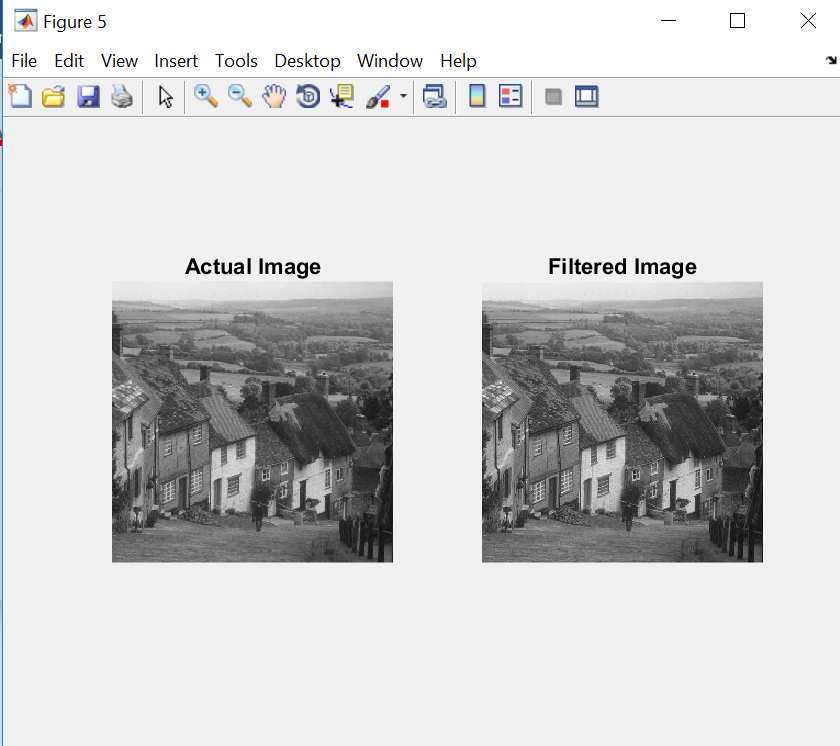
figure;

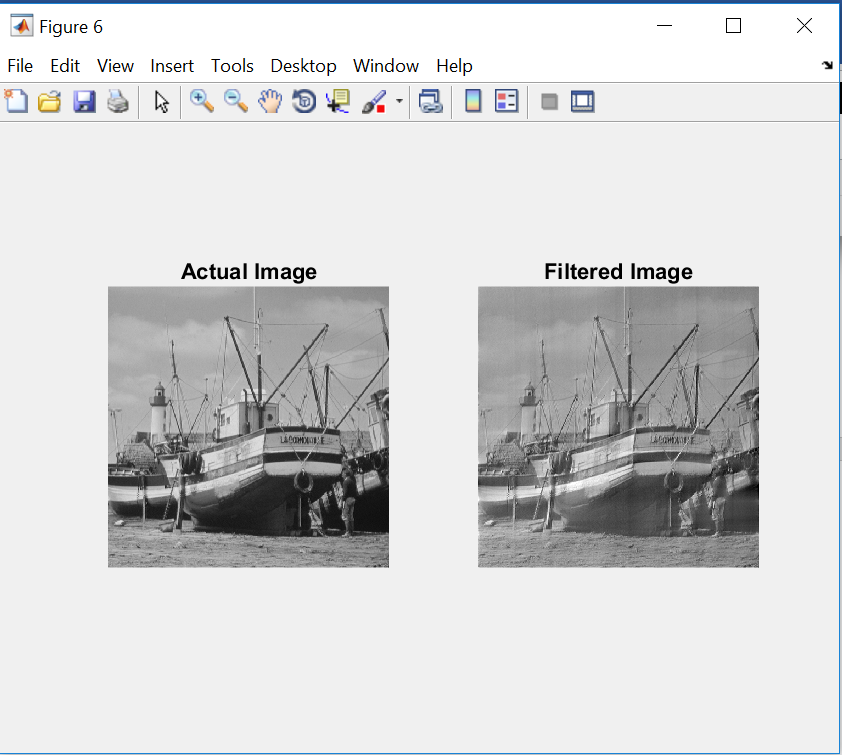
subplot(1,2,1);imshow(uint8(Input\_Img4));title('Actual Image');

subplot(1,2,2);imshow(uint8(Img4\_ifft));title('Filtered Image');

**OUTPUT:**







**CONCLUSION:**

Thus, in this project, we performed Fourier transform on different images and then passed them through IGF (“Inverse Gaussian Filter”). Then we obtained filtered image using Inverse Fourier Transform. The basic use of IGF is blurring and unblurring. So it removes blurring effects from any image.