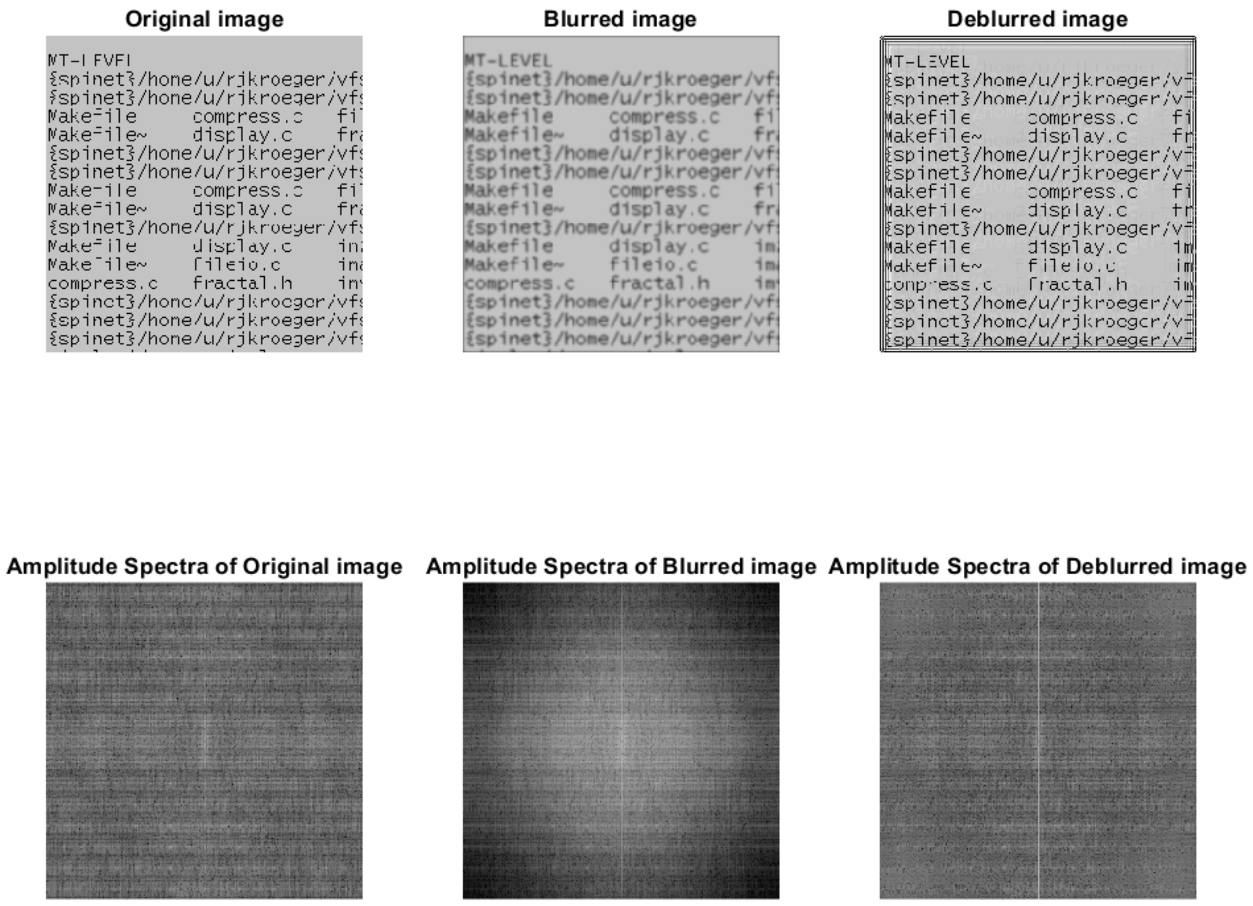
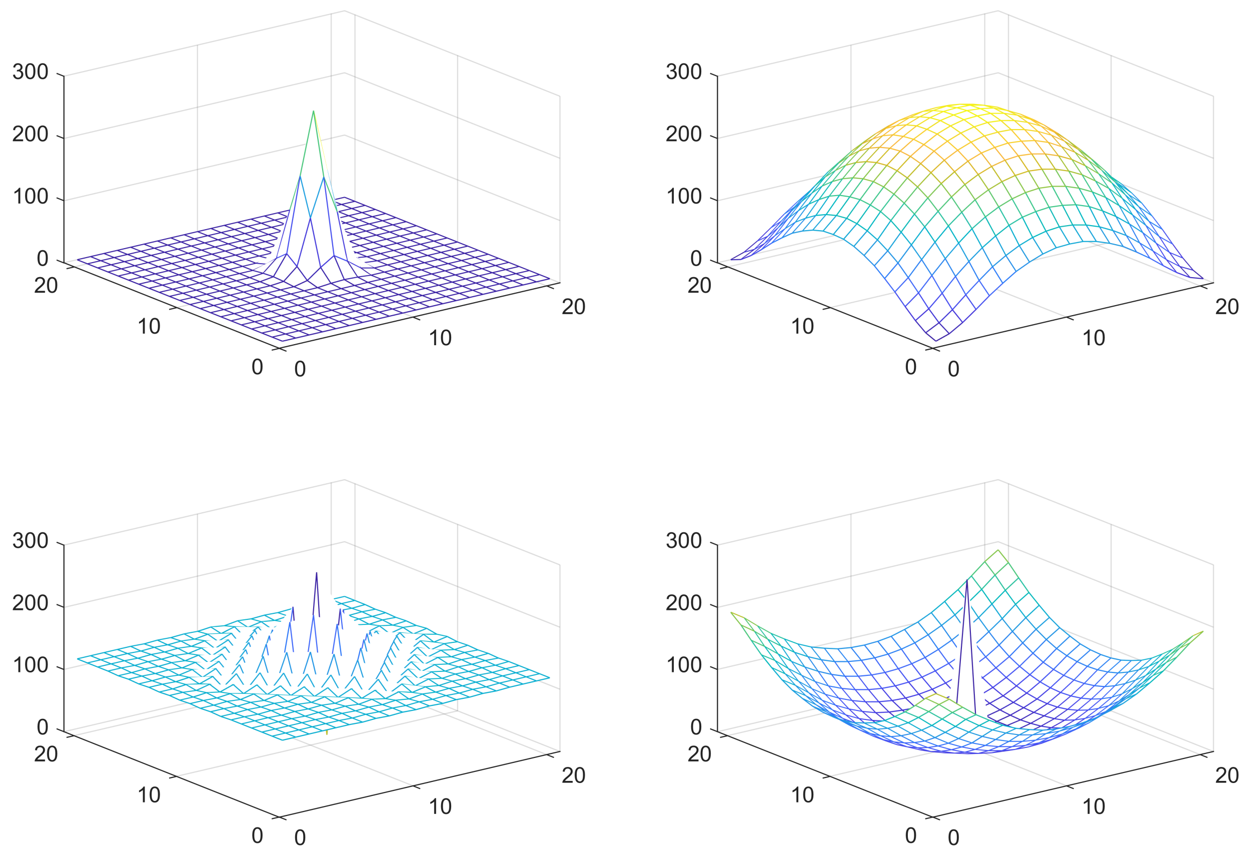
4. Image Deblurring by Frequency Domain Inverse Filter Design

```
% Loading the image from current directory
current_dir = pwd;
file_name = '\text.tif';
im = imread(strcat(current_dir,file_name));
img = double(im);
% 2D Gaussian Filter
stand_deviation=1;
gaussian_filter = zeros(21,21);
[m n] = size(gaussian_filter);
for i=1:21
    for j=1:21
        neumerator = exp(-1*((i-(m+1)/2)^2+(j-(n+1)/2)^2)/(2*stand_deviation^2)) ;
        denominator = 1/(2*pi*stand_deviation^2);
        gaussian_filter(i,j)=(denominator*neumerator);
    end
end

% sum of the coefficients
sum(gaussian_filter, 'all');

%2D convolution between the original image and the Gaussian filter
blurred_image=conv2(img,gaussian_filter,'same');

%2D-DFT to the spatial domain Gaussian filter
%Frequency domain inverse Gaussian filter
kernfreq_dm_iGF = real(ifft2(1./fft2(gaussian_filter)));

%2D convolution between the spatial domain Gaussian filtered image and the spatial domain inver
spacial_dm_dImg=conv2(blurred_image,kernfreq_dm_iGF','same');


% Normalization of the images by intensity transformation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Original Image
image_temp=log(1+ abs(fftshift(fft2(img))));
%Min-Max normalization
original_normalized_image=((image_temp-min(min(image_temp)))./...
    (max(max(image_temp))-min(min(image_temp)))).*(size(image_temp,1)-1);

% Blurred Image
image_temp_1=log(1+ abs(fftshift(fft2(blurred_image))));
%Min-Max normalization
blur_normalized_image=((image_temp_1-min(min(image_temp_1)))./...
    (max(max(image_temp_1))-min(min(image_temp_1)))).*(size(image_temp_1,1)-1);

% Deblurred Image
image_temp_2=log(1+ abs(fftshift(fft2(spacial_dm_dImg))));
%Min-Max normalization
deblur_normalized_image=((image_temp_2-min(min(image_temp_2)))./...
    (max(max(image_temp_2))-min(min(image_temp_2)))).*(size(image_temp_2,1)-1);
```

```
if true
    figure('Renderer', 'painters', 'Position', [10 10 900 700])
    subplot(2,3,1), imshow(uint8(img)), title('Original image')
    subplot(2,3,2), imshow(uint8(blurred_image)), title('Blurred image')
    subplot(2,3,3), imshow(uint8(spacial_dm_dImg)), title('Deblurred image')
    subplot(2,3,4), imshow(uint8(original_normalized_image)),title('Amplitude Spectra of Origin
    subplot(2,3,5),imshow(uint8(blur_normalized_image)),title('Amplitude Spectra of Blurred ima
    subplot(2,3,6),imshow(uint8(deblur_normalized_image)),title('Amplitude Spectra of Deblurred

end
```



```
% Min-Max normalizing Gaussian blur filter
blur_GF=((gaussian_filter-min(min(gaussian_filter)))./...
    (max(max(gaussian_filter))-min(min(gaussian_filter)))).*255;

%2D DFT on Normalized Gaussian blur filter
blur_GF_fft_temp = log(1+ abs(fftshift(fft2(blur_GF))));
blur_GF_fft=((blur_GF_fft_temp-min(min(blur_GF_fft_temp)))./...
    (max(max(blur_GF_fft_temp))-min(min(blur_GF_fft_temp)))).*255;

% Min-Max normalizing Gaussian de-blur filter
deblur_GF=((kernfreq_dm_iGF-min(min(kernfreq_dm_iGF))./...
    (max(max(kernfreq_dm_iGF))-min(min(kernfreq_dm_iGF)))).*255;

%2D DFT on Normalized Gaussian de-blur filter
deblur_GF_fft_temp = log(1+ abs(fftshift(fft2(deblur_GF))));
deblur_GF_fft=((deblur_GF_fft_temp-min(min(deblur_GF_fft_temp))./...
    (max(max(deblur_GF_fft_temp))-min(min(deblur_GF_fft_temp)))).*255;

if true
    figure('Renderer', 'painters', 'Position', [10 10 900 600])
    subplot(2,2,1); mesh(uint8(blur_GF));
    subplot(2,2,2); mesh(uint8(blur_GF_fft));
    subplot(2,2,3); mesh(uint8(deblur_GF));
    subplot(2,2,4); mesh(uint8(deblur_GF_fft));
end
```



300
200
100
0
20
10
0