

Image Processing Project # 3

**HOMOMORPHIC PROCESSING AND ITS APPLICATION TO
IMAGE ENHANCEMENT**

Submitted by:

Nithya Sundaram

HOMOMORPHIC PROCESSING AND ITS APPLICATION TO IMAGE ENHANCEMENT

Introduction

The objective of the project is to study and implement the techniques of multiplicative homomorphic systems designed for enhancing the images in spatial domain. The image can be characterized by two components, (1) the amount of source illumination incident and (2) amount of illumination reflected by the object. These are called the illumination and reflectance components of the image. The illumination-reflectance model can be used to develop a frequency domain procedure for improving the appearance of an image by simultaneous gray-level range compression and contrast enhancement. In this application, the key to the approach is the separation of the illumination and the reflectance components.

Homomorphic Filtering

An image as a function can be expressed as the product of illumination and reflectance components as follows:

$$F(x,y) = I(x,y) * R(x,y) \quad (1)$$

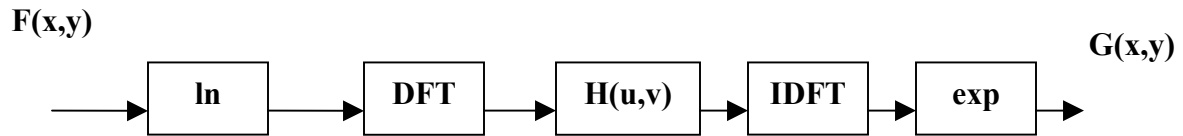
Equation (1) cannot be used directly to operate separately on the frequency components of illumination and reflectance because the Fourier transform of the product of two functions is not separable. Instead the function can be represented as a logarithmic function wherein the product of the Fourier transform can be represented as the sum of the illumination and reflectance components as shown below:

$$\ln(x,y) = \ln(I(x,y)) + \ln(R(x,y)) \quad (2)$$

The Fourier transform of equation (2) is

$$Z(u,v) = Fi(u,v) + Fr(u,v) \quad (3)$$

The fourier transformed signal is processed by means of a filter function $H(u,v)$ and the resulting function is inverse fourier transformed. Finally, inverse exponential operation yields an enhanced image. This enhancement approach is termed as homomorphic filtering. The whole operation is expressed as a block diagram below:



Implementation of Homomorphic filtering

Consider an image with 256 x 256 pixels where the pixels have varying intensity. The pattern goes from dark to light as we go from left to right. This is also called a horizontal intensity and it can be visualized by creating a matrix of size 256 with non-uniform illumination. This is the first illumination pattern under investigation. **Figure1** shows the horizontal illumination pattern of 256 x 256 pixels.

Illumination Pattern



Figure 1: Illumination pattern, Image with varying intensity

The image is then multiplied with the non-uniform illumination. In the real world the corrupted images are expected to be the way the image that is corrupted by this type of non-uniform illumination. The corrupted image under investigation is as shown in **Figure 2**. The figure shows a comparison of the original image and the non-uniformly illuminated image. The aim of the project is to homomorphically filter out the illumination component to retain the original image.

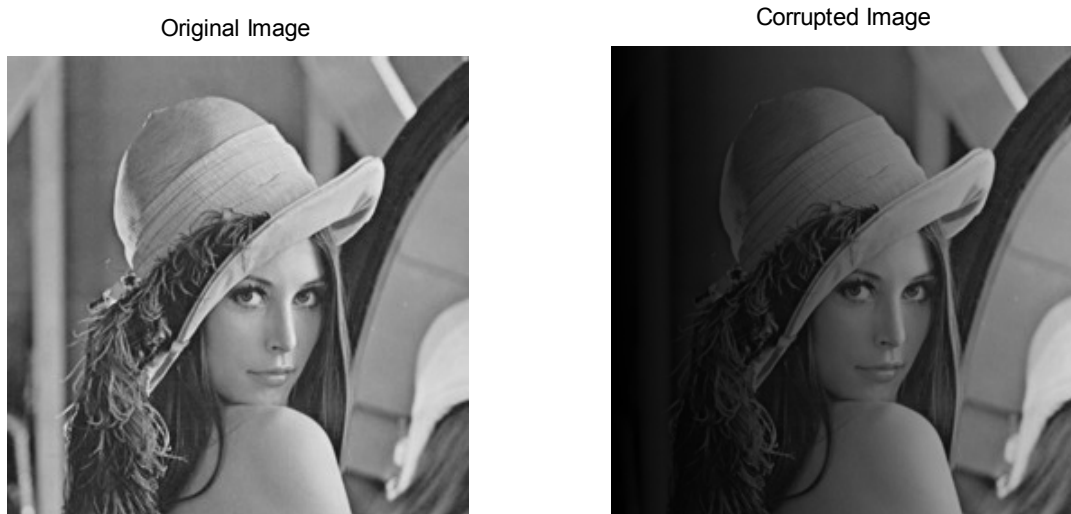


Figure 2: Left panel – Original image, Right panel – Corrupted image

The histogram of the corrupted image is shown in **Figure 3**. As we can see, the histogram is very non-uniform. The areas where it is corrupted densely i.e., with more darkness intensity, the histogram has a greater value than the lesser dense area. This is due to the contribution from the non-uniform illumination component of the image.

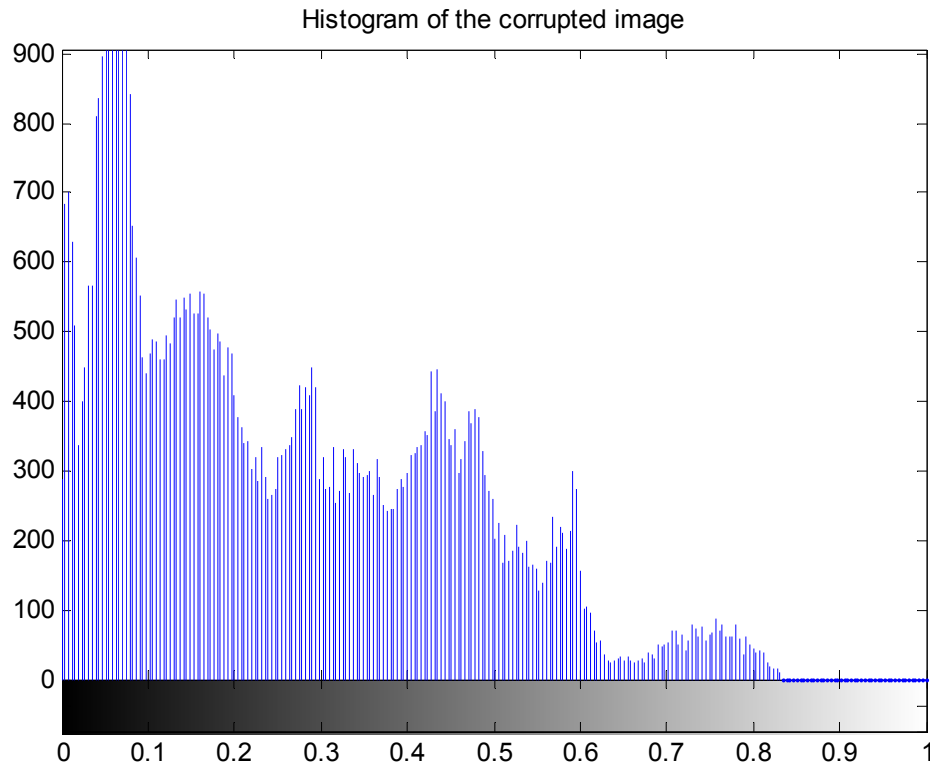


Figure 3: Histogram of the non-uniformly illuminated image

Histogram equalization is a technique that divides the bins of the histogram and spreads the image uniformly to some extent. The restored image as a result of the histogram equalization is shown in **Figure 4**. The overall intensity is much lighter so that the figure appears better than the corrupted image i.e., the darker portions of the corrupted images are visible clearly. However the results are not perfect due to the continued presence of the non-uniform illumination.



**Figure 4: Result of histogram equalization, Left pane – Original image,
Right panel - Restored image**

The illumination component of an image generally is characterized by slow spatial variations, while reflectance component tends to vary abruptly. These characteristics lead to associating the low frequencies of the Fourier transform of the logarithm of an image with illumination and high frequencies with reflectance, which are used as an advantage in image enhancement.

Using the technique of homomorphic filtering, assuming that we know the image, we can filter out all kinds of illumination and restore the image. The frequency characteristics of the illumination component are shown in **Figure 5**. The frequency component of the log of the illumination is then subtracted from the frequency characteristics of the corrupted image.

A good deal of control can be gained over the illumination and reflectance components with a homomorphic filter assuming that we know the image. This control requires specification of a filter function $H(u,v)$ that affects the low and high frequency components of the fourier transform in different ways.

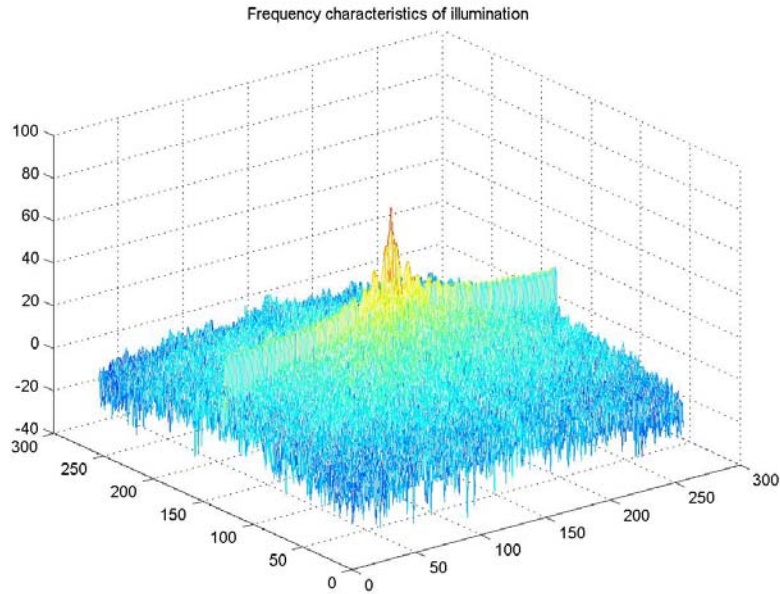


Figure 5: Frequency characteristics of the illumination pattern

The parameters of the filter are chosen in such a way so that the filter function tends to decrease the contribution made by the low frequencies (illumination) and amplify the contribution made by high frequencies (reflectance). The net result is simultaneous dynamic range compression and contrast enhancement. One such filter designed for this type of non-uniform illumination is shown in **Figure 6**.

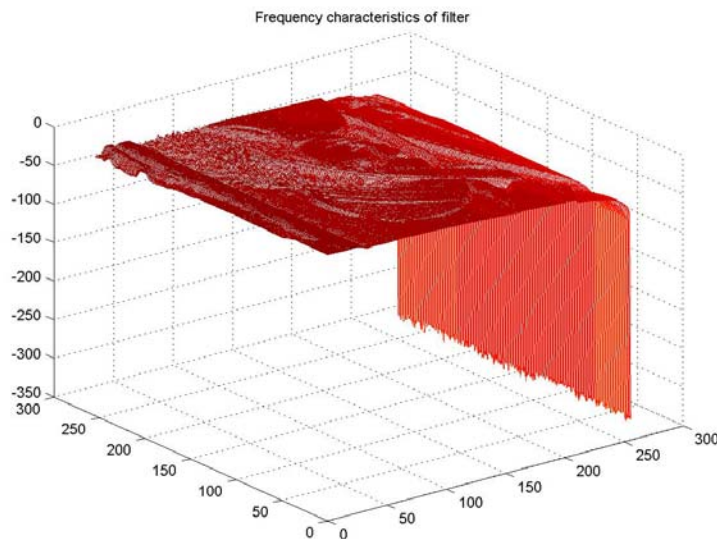


Figure 6: Frequency characteristics of Homomorphic filter

Metrics for deciding the performance of Homomorphic filtering

The comparison of frequency characteristics of the original image and the image restored with the help of the homomorphic filtering is chosen as a metric. The resulting image displays frequency characteristics similar to that of the original filter. **Figure 7** shows the frequency characteristics of the original image in the left end compared to the frequency characteristics of the restored image in the right end of the plot and frequency characteristics of the corrupted image in the bottom.

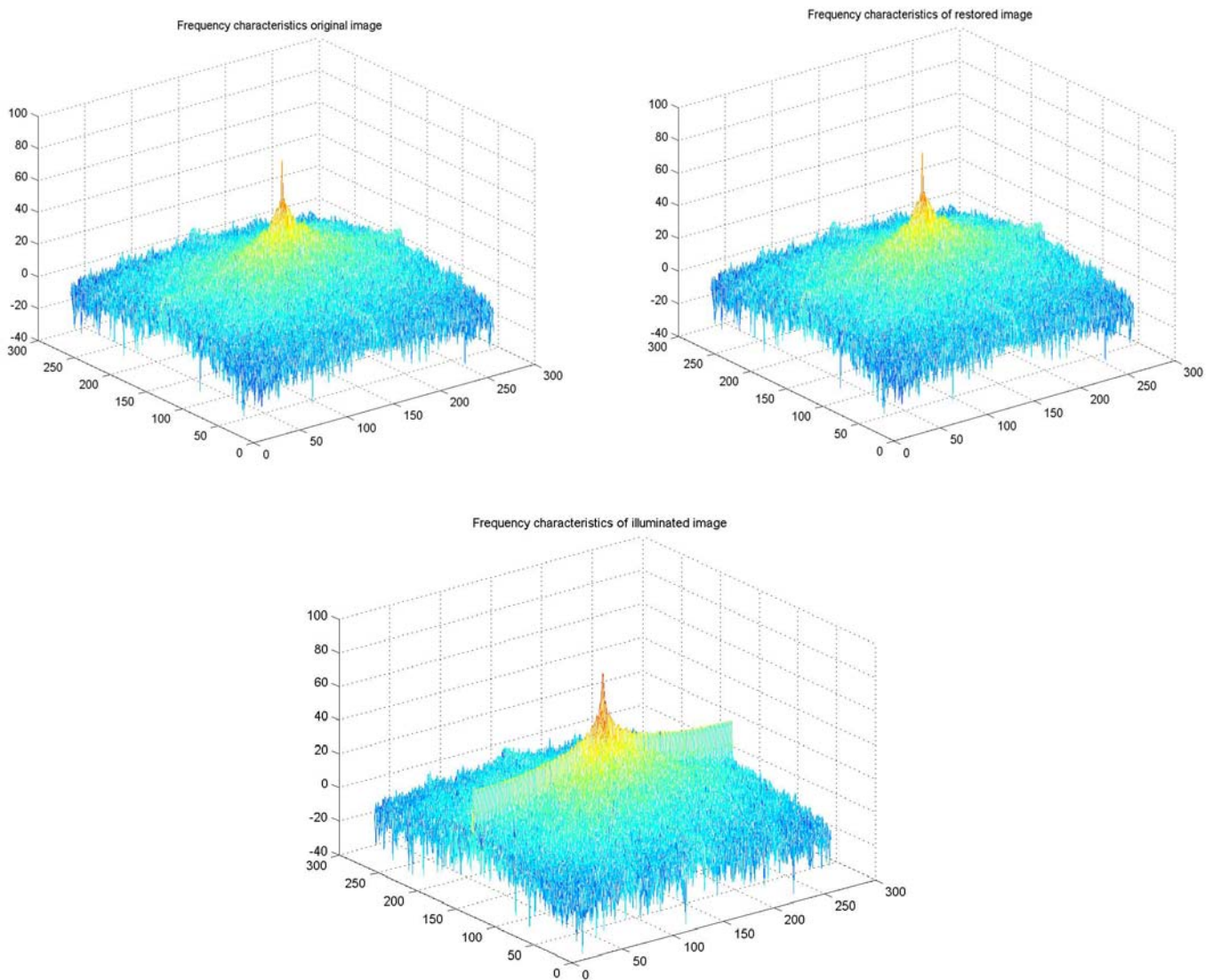


Figure 7: Original image frequency characteristics (left) with restored image frequency characteristics (right) in comparison with the frequency characteristics of the illuminated image (bottom).

The restored images are also compared in the spatial domain and the images are displayed in the **Figure 8**. The restored image looks very close to the original image i.e., the illumination component is completely filtered out so that the non-uniform pattern is no more visible to the naked eye. This metric proves the method of homomorphic filtering to be a very efficient technique.



Figure 8: Original image (left panel), Restored image (right panel)

Another metric in this case is the error performance matrix or the error plot of the images, which decides the performance of the homomorphic filtering. The error plot is shown in **Figure 9** below.

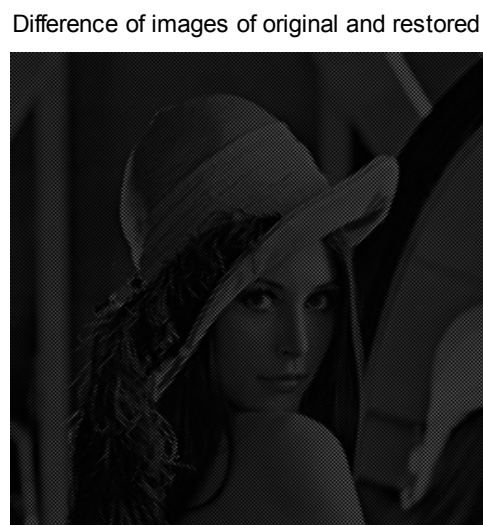


Figure 9: Difference of original and restored images

Similar experiments were done with different types of illumination patterns. The previous illumination pattern was rotated 90 degrees to obtain a vertical illumination and also 45 degrees tilt and a sine wave pattern was tried as illumination. The following **Figure 10** shows the different types of illumination patterns tried with the technique of homomorphic filtering.

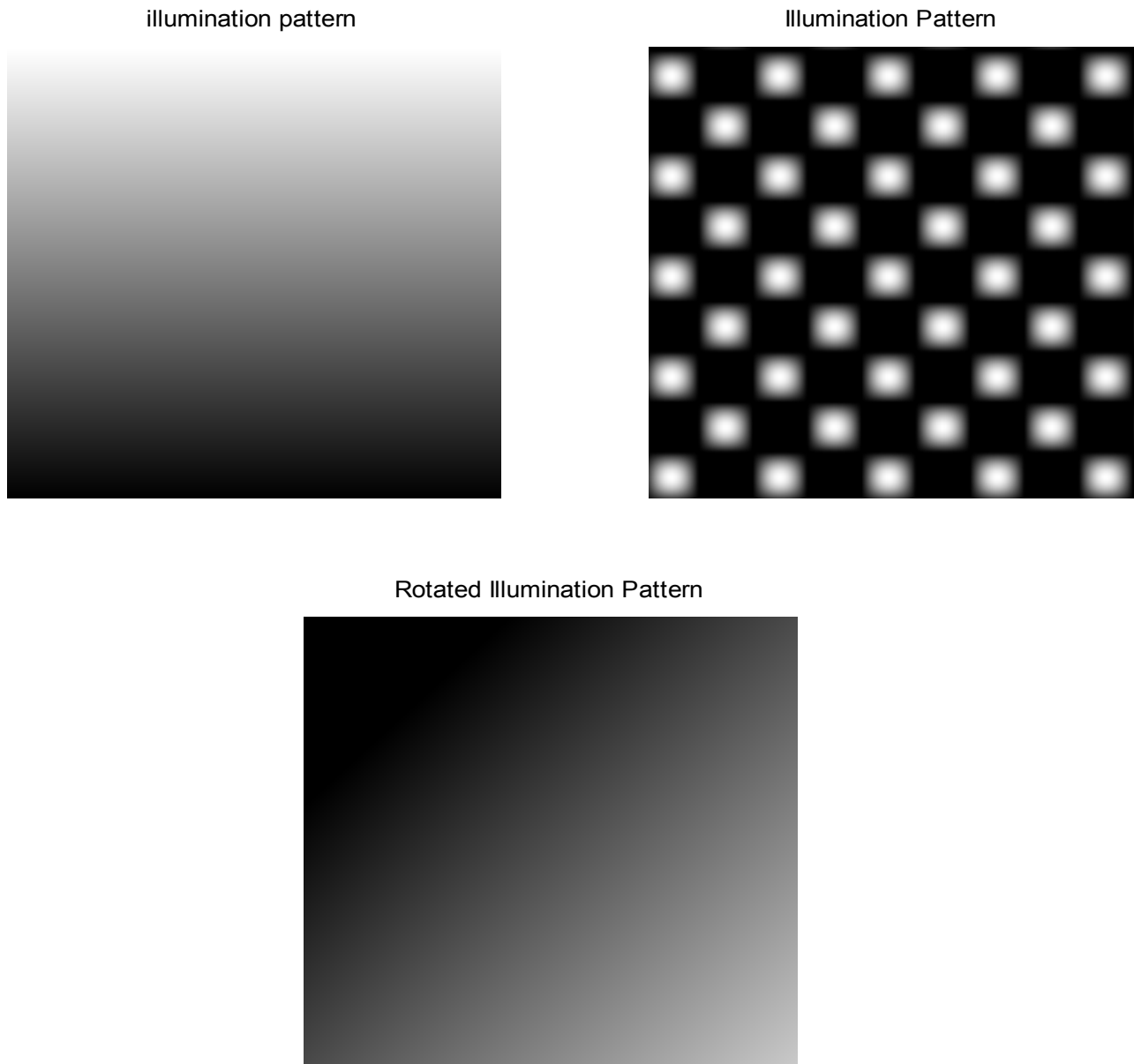
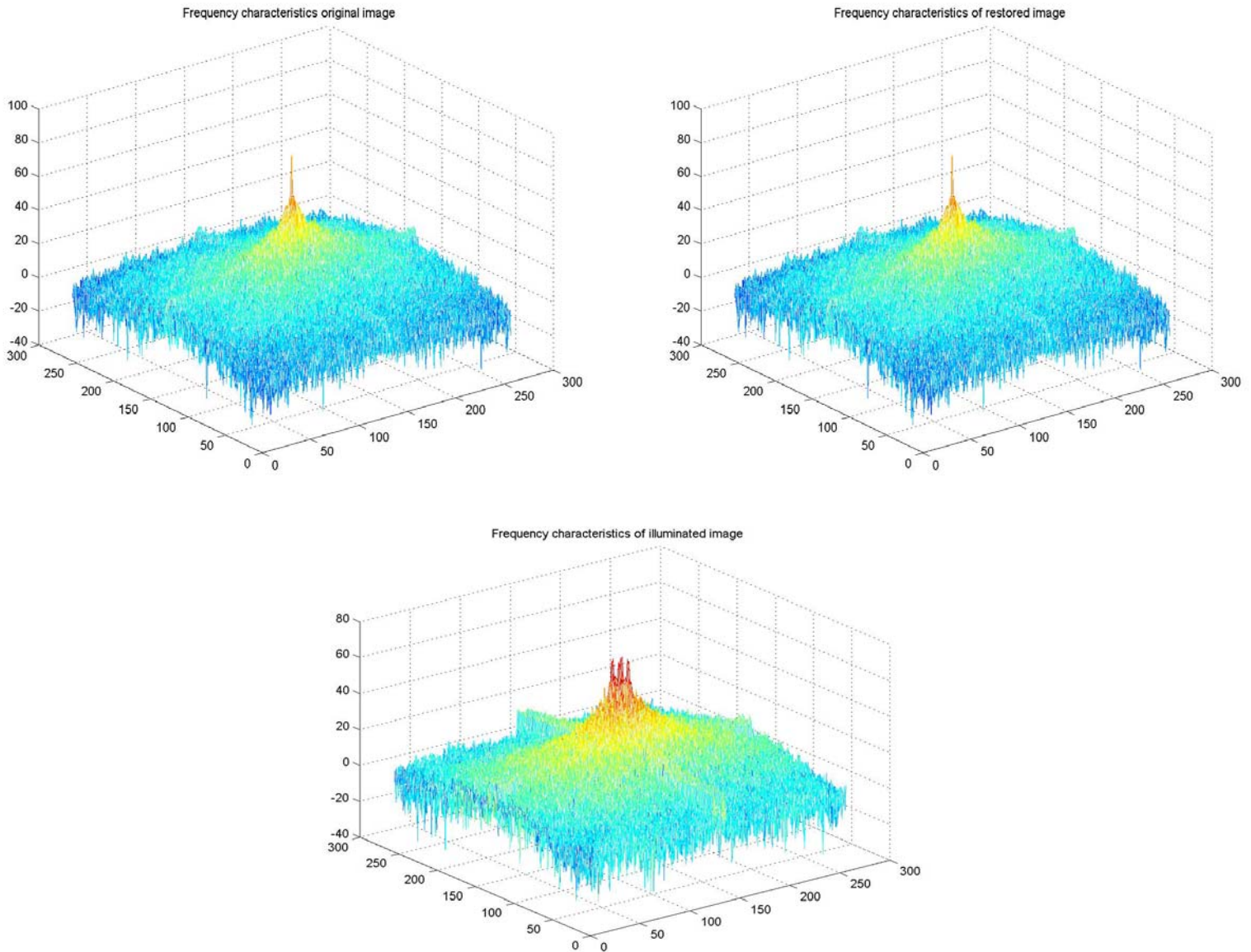


Figure 10: Different illumination patterns, Top panel left – vertical illumination, Top right panel – sine wave pattern, Bottom panel – 45 degrees rotated illumination pattern

Experiments with sine wave illumination pattern

The image with the illumination pattern in the form of sine wave was investigated and is shown in the **Figure 11**. Here also, the results of homomorphic filtering are found to be much effective than other conventional methods. **Figure 11** displays the frequency characteristics of the original image in the left end and frequency characteristics of the restored image in the right end along with the frequency characteristics of the corrupted image in the bottom.



A homomorphic filter was designed using the multiplicative and logarithmic techniques. The net result is simultaneous dynamic range compression and contrast enhancement. One such filter designed for sinusoidal type of non-uniform illumination is shown in **Figure 12**.

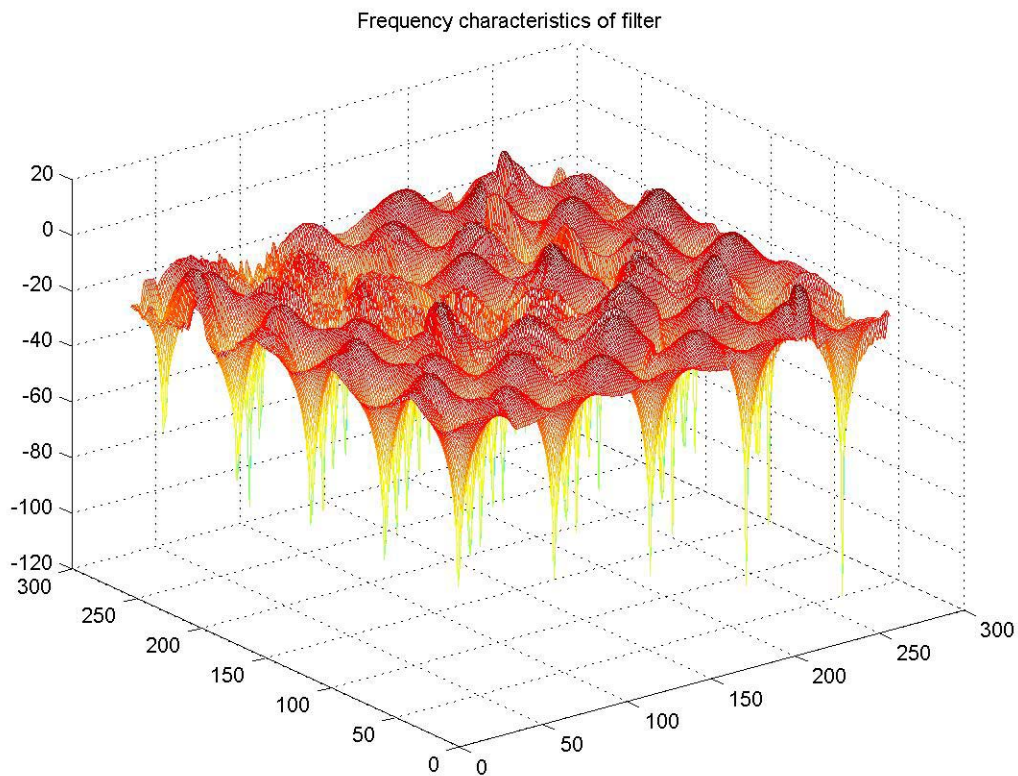


Figure 12: Frequency characteristics of homomorphic filter designed for a sinusoidal non-uniform illumination

The original image and the image restored using the homomorphic filtering is also compared in the spatial domain. The result shows that the homomorphic filtering gets rid of the whole non-uniform illumination and the restored image looks much closer to the original image. The results are shown in **Figure 13**.

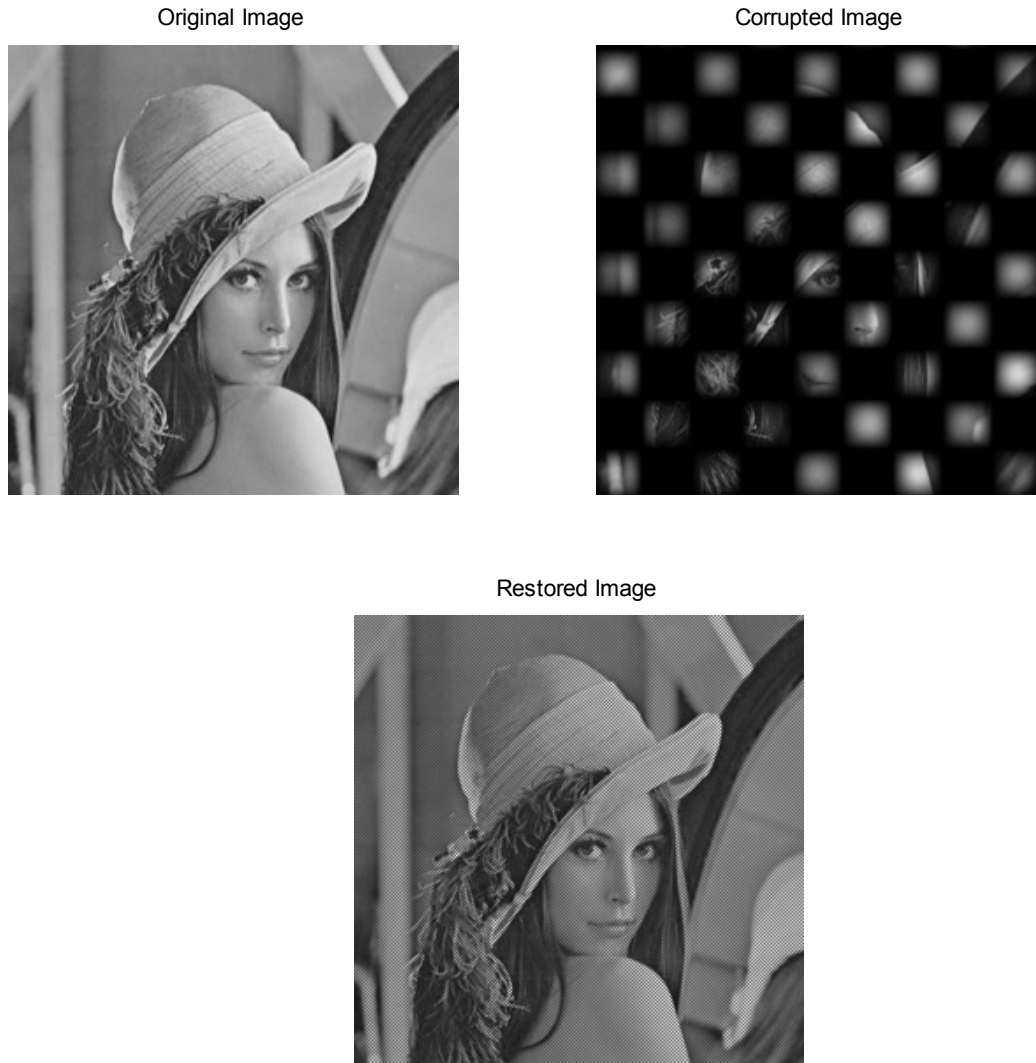


Figure 13: Results of homomorphic filtering, Top left – Original, Top right – Corrupted, Bottom – Restored image

Conclusion

This project is an illustration of a non-linear signal processing. The algorithm is based on a very simple concept i.e., assuming that we know the image; we can filter out all kinds of illumination and restore the image. This technique is simple conceptually but requires a lot of tuning towards the algorithm in all stages to get the restored image close to the original image. Important information to be noted in this project is that all the filtering was done in frequency domain and the improvements were noted in the spatial domain.

Matlab Source Code

```
clear all;
close all;

I = imread('C:\MATLAB6p1\work\lena.jpg');

G = rgb2gray(I);
GM = im2double(G);

GN = 1-GM;
GN = im2uint8(GN);

t = 256;
ill = illumination_v1(t);

product = GM.*ill;

figure;
imshow(GM);
title('Original Image');

figure;
imshow(product);
title('Corrupted Image');

figure;
kk = imhist(product);
imhist(product);
title('Histogram of the corrupted image');

figure;
imhist(kk);
title('Histogram after equalization');

figure;
jj = histeq(product);
imshow(jj);
title('Image restored after histogram equalization');

log_GM = log(GM + 1);
log_product = log(product + 1);

fft_GM = (fftshift(fft2(log_GM)));
figure;
original_value = (20*log10(abs(fft_GM)));
mesh(20*log10(abs(fft_GM)));
title('Frequency characteristics original image');

fft_product = (fftshift(fft2(log_product)));
figure;
```

```
mesh(20*log10(abs(fft_product)));
title('Frequency characteristics of illuminated image');
```

```
diff_ill = (fft_product - fft_GM);
figure;
mesh(20*log10(abs(diff_ill)));
title('Frequency characteristics of illumination');
```

```
filt_filt = ifft2(diff_ill);
figure;
mesh(20*log10(abs(filt_filt)));
title('Frequency characteristics of filter');
```

```
restored = (fft_product - diff_ill);
figure;
desired_value = (20*log10(abs(restored)));
mesh(20*log10(abs(restored)));
title('Frequency characteristics of restored image');
```

```
restored_im = ifft2(restored);
```

```
restored_image = (exp(restored_im)) - 1;
```

```
figure;
abs_restored_image = abs(restored_image);
imshow(abs_restored_image);
title('Restored Image');
```

```
%metrics for testing the performance
final = original_value - desired_value;
sum_ave = det(final);
```

```
%error image
error = restored_image - GM;
error1 = im2uint8(error);
figure;
imshow(error1);
title('Error image');
```

Function file

```
function[ill] = illumination_v1(t);
```

```
p = 1/t:1/t:1;
```

```
for k = 1:t,
    ill(k,:) = p(:);
end
figure;
imshow(ill);
title('Illumination Pattern');
```

```

p = 1/t:1/t:1;

for k = 1:t,
    ill(k,:) = p(:);
end
ill = rot90(ill);
figure;
imshow(ill);
title('illumination pattern');

```

```

p = 1/t:1/t:1;

for k = 1:t,
    il(k,:) = sin(9*p*pi + 0.2);
end

for k = 1:t,
    il_1(k,:) = sin(9*p*pi + 0.2);
end

```

```

il_1 = rot90(il_1);

ill = il.*il_1;

figure;
imshow(ill);
title('Illumination Pattern');

```

```

p = 1/t:1/t:1;

for k = 1:t,
    il(k,:) = p(:);
end
figure;
imshow(il);
title('Illumination Pattern');

ill = imrotate(il,-45,'bilinear','crop');
imshow(ill);
title('Rotated Illumination Pattern');

```

```

dia = sqrt(2*(t^2));

p = 1/dia:1/dia:1;

for k = 1:dia,
    il(k,:) = p(:);
end
size_il = size(il)
figure;
imshow(il);

```



```
title('Illumination Pattern');

J = imrotate(il,-45,'bilinear','crop');
ill = imcrop(J,[0,0,t,t]);
size_ill = size(ill)
figure;
imshow(ill);
title('Rotated Illumination Pattern');

p = 1/t:1/t:1;

for k = 1:t,
    il(k,:) = exp(p*pi/10);
end

for k = 1:t,
    il_1(k,:) = exp(p*pi/10);
end

il_1 = rot90(il_1);

ill = il.*il_1;

figure;
imshow(ill);
title('Illumination Pattern');
```