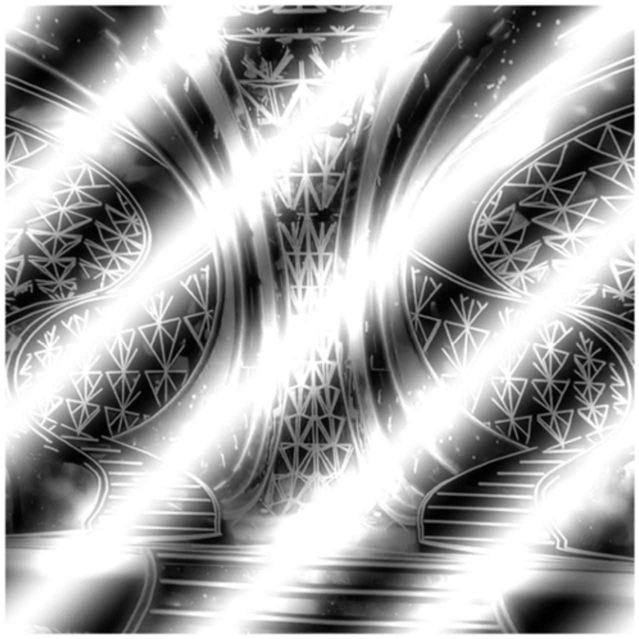
Video Signals

Date: 12 February 2024

Ex.1.[11 pts] The following image (size: 800x800) has a specific periodic noise to be removed working in the frequency domain.
Describe the noise in the frequency domain addressing its 2D Discrete Fourier Transform.
Provide a method, working in the frequencies that can remove the noise.
Describe the whole pipeline to denoise the image and to provide the restored one.



Ex.2.[11 pts]
A display has a luminance contrast ratio of 2000:1; assuming a Weber's constant of 0.02 what will be the minimum number of non-uniform quantization levels to avoid any visible intensity transition? If the quantization has to be linear what would be the minimum number of levels to satisfy the same constraint?
Detail every mathematical step.

Ex. 3 is overleaf

Es.3. [11 pts to be solved writing on the paper a suitable MATLAB code]
You want to develop an application able to fix underexposed figure based on a family of methods called Center Surround Retinex.

List of possible Matlab functions

```
figure
im2doubl
e
imread
rgb2gray
imcrop
imfilter
imhist
imopen
rgb2ind
histeq
rgb2hsv
imshow
fspecial
fft2
strel
imdilate
hsv2rgb
```

Write a MATLAB script able to perform the following steps:

a) Read the input image (stored in *'input.png'* file), convert it to a double representation and visualize it.
b) Convert the image to grayscale, obtaining *I*.
c) Apply a gaussian filter to *I* in the frequency domain following these steps:
   I. Apply the Fourier Transform to *I.*
   II. Define a matrix with the same size as *I*, containing normalized values of a gaussian function centered in the middle and having standard deviation equal to 80 (*hint: use fspecial*).
   III. Apply the Fourier Transform to the matrix obtained in the previous step.
   IV. Compute the element wise multiplication between the two FFT outputs and move back to the spatial domain.
d) Compute the difference between the logarithm of I and the logarithm of the image obtained in c). (*hint: add 1 to avoid calculating the logarithm of 0*).
e) Linearly rescale what you obtained from the previous point in order to have values in the [0,1] range.
f) Calculate the HSV representation of the color input image. Substitute the value channel with the matrix obtained in e).
g) Convert back the new HSV image into the RGB representation and visualize the result.

Solutions

Es.1

The noise is represented by a planar sinusoid that performs 3 periods in the vertical direction and 3 periods in the horizontal one.
Since the maximum is in the top-left corner (assumed as the point with x=0 and y=0) we can write the noise as:

$$noise(n,m) = \cos\left(2\pi \cdot \frac{3}{800} n\right) \cos\left(2\pi \cdot \frac{3}{800} m\right)$$

Es.2

For the non uniform case:

$$\frac{I_{max}}{I_{min}} = (1+0.02)^n = 2000 \rightarrow n \log 1.02 = \log 2000 \rightarrow n \cong 384$$

For the uniform case we would have: $n = \frac{I_{max} - I_{min}}{0.02} = 99950$

Es.3

```
clc
close all
clear all

%a)
Im = im2double(imread('input.png'));
figure
imshow(Im)

%b)
l = rgb2gray(Im);

%c I)
L = fft2(l);

%c II)
gauss = fspecial('gaussian',size(l),80);

%c III)
fgauss = fft2(gauss);

%c IV)
ls = ifft2(fgauss.*L);

%d)
l_log = log(l+1);
ls_log = log(ls+1);
ret = l_log - ls_log;

%e)
ret = (ret - min(ret(:))) / (max(ret(:)) - min(ret(:)));

%f)
Ihsv = rgb2hsv(Im);
Ihsv(:, :, 3) = ret;

%g)
```

```
out = hsv2rgb(Ihsv);
figure
imshow(out)
```