EE604A - Digital Image Processing Assignment

Satya Prakash Panuganti, 14610 1 September, 2017

Image Sources

Sunset (high.jpg) : https://www.flickr.com/photos/mediaflex/4190084346
Carving (low.jpg) : https://www.flickr.com/photos/30440933@N06/2847993403
Dog (small.jpg) : https://res.cloudinary.com/rover-com/image/upload/a_exif,c_fill,f_jpg,fl_progressive g_face:center,h_100,q_80,w_100/remote/images/pets/4NpPzO8N/50e4a023d9/original.jpg
divyat.jpg : Own work. Taken with permission.

Solution 1

(a)

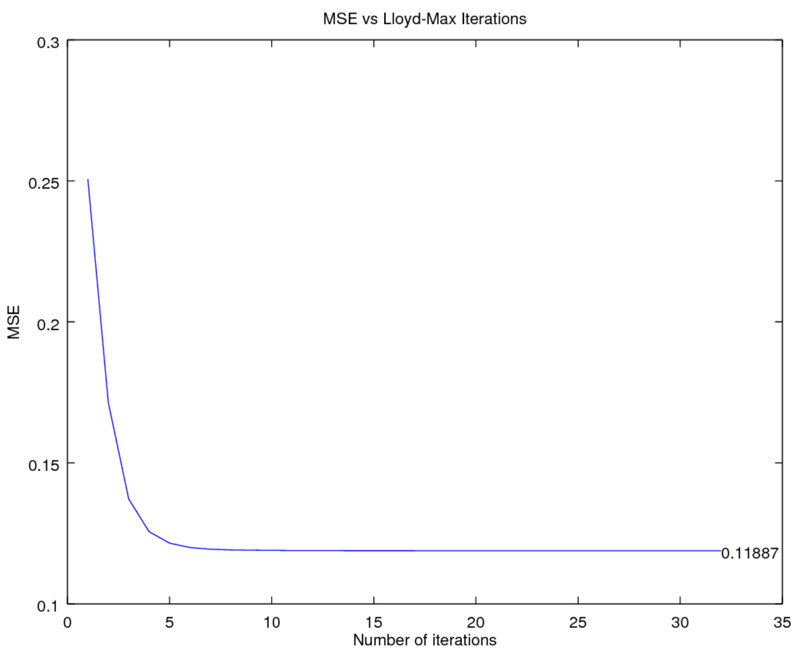
Code has been written in two MATLAB® \GNU Octave files :

- Q1\lloyd_max_quantizer.m : The file contating the function.
- Q1\Q1.m : The script to perform the required tasks.

(b)

The Lloyd-Max Quantizer was run over a signal sampled from a gaussian distribution of zero mean and unit variance.
The 4 representation levels are : -1.51440 -0.43642 0.50108 1.57091
The corresponding transition levels are : $-\infty$ -0.975410 0.032332 1.035995 ∞

(c)



(d)

Experiments

μ	σ	l_1	l_2	l_3	l_4	m_1	m_2	m_3	m_4	m_5	MSE
0	1	-1.51440	-0.43642	0.50108	1.57091	$-\infty$	-0.975410	0.031332	1.035995	∞	0.11887
0	0.5	-1.08358	-0.34192	0.29386	1.04099	$-\infty$	-0.712752	-0.024028	0.667426	∞	0.05901
0	2	-2.20816	-0.71628	0.56213	2.04503	$-\infty$	-1.46220	-0.77074	-1.30352	∞	0.23625
2.3	1	0.81634	1.86325	2.77444	3.84569	$-\infty$	1.3398	2.3188	3.3101	∞	0.11695
2.3	0.5	1.2784	2.0036	2.6423	3.3939	$-\infty$	1.6410	2.3229	3.0181	∞	0.059066
2.3	2	0.14523	1.65523	2.92182	4.4294	$-\infty$	0.90023	2.28853	3.67338	∞	0.23724

Observations

(e)

Let the signal be given by S and the pdf of S be $p_S(s)$.

$$Now,\ p_S(s) = \begin{cases} \dfrac{1}{b-a} & a \leq s < b \\ 0 & \text{otherwise} \end{cases}$$

Let us assume that there 'L' representation levels, a_i be the representation levels and m_i be the transition levels.
We begin by initializing m_i uniformly :

$$m_i = a + \frac{b-a}{L} \times (i-1)\ \forall\, i \in [1, L+1]$$

We next set a_i :

$$\begin{aligned} a_i &= \frac{\int_{m_i}^{m_{i+1}} s p_S(s) ds}{\int_{m_i}^{m_{i+1}} p_S(s) ds}\ \forall\, i \in [1, L] \\ &= \frac{\frac{m_{i+1}^2 - {m_i}^2}{2(b-a)}}{\frac{(m_{i+1}-m_i}{b-a}}\ \forall\, i \in [1, L] \\ &= \frac{m_i + m_{i+1}}{2}\ \forall\, i \in [1, L] \\ &= a + \frac{1}{b-a} \times \frac{i+i-1}{2}\ \forall\, i \in [1, L] \\ &= a + \frac{2i-1}{2(b-a)}\ \forall\, i \in [1, L] \end{aligned}$$

We now set m_i :

$$\begin{aligned} m_i &= \frac{a_{i-1} + a_i}{2}\ \forall\, i \in [2, L] \\ &= a + \frac{1}{4(b-a)} \times (2(i+i-1) - 2)\ \forall\, i \in [2, L] \\ &= a + \frac{b-a}{L} \times (i-1)\ \forall\, i \in [2, L] \end{aligned}$$

m_i do not change in the above step. Hence, the Lloyd-Max Quantizer has converged to a final solution.

Solution 2

A C++ function, EE604A::histogram_matching(), for histogram matching of cv::Mat images has been developed. The code required for the matching function is present in :

- Q2\src\histogram_matching.cxx
- Q2\src\histogram_matching.h

A small program which uses the histogram matching function is present in Q2\src\Q2.cxx

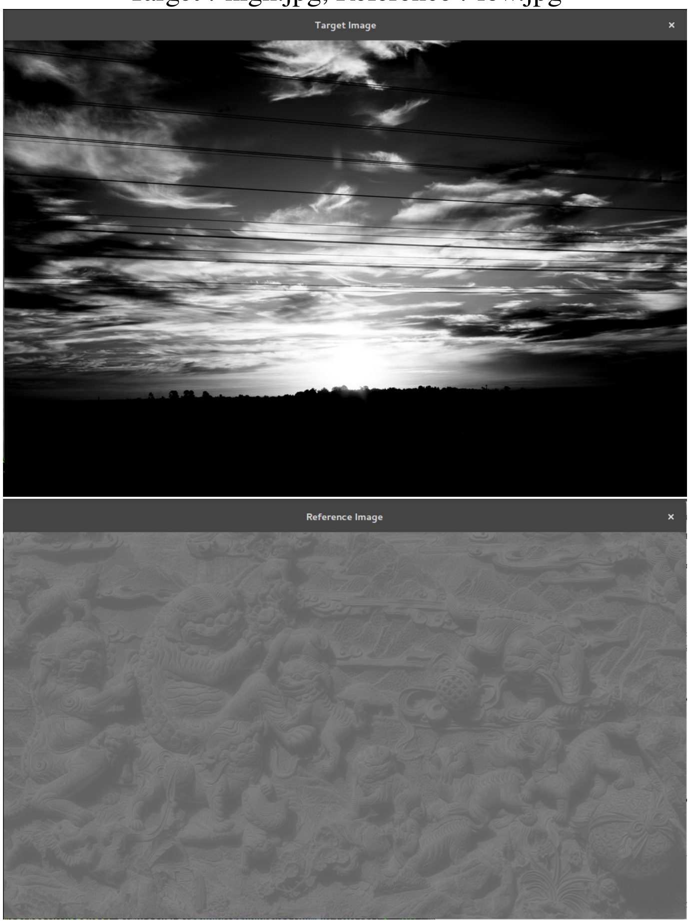
In order to build and run the code, the following steps need to be followed :

1. Enter Q2\src through terminal.
2. run build.sh (OpenCV needs to installed and a version of g++ supporting c++14 needs to be present)
3. Execute Q2 (the binary file) with the reference and target relative/absolute image paths. Eg. [./Q2 ../../images/high.jpg ../../images/low.jpg] or [./Q2 ../../images/low.jpg ../../images/high.jpg] (On Ubuntu)
4. The program can be closed by Ctrl+C on the terminal or by pressing 'q' when one of the image windows is open.

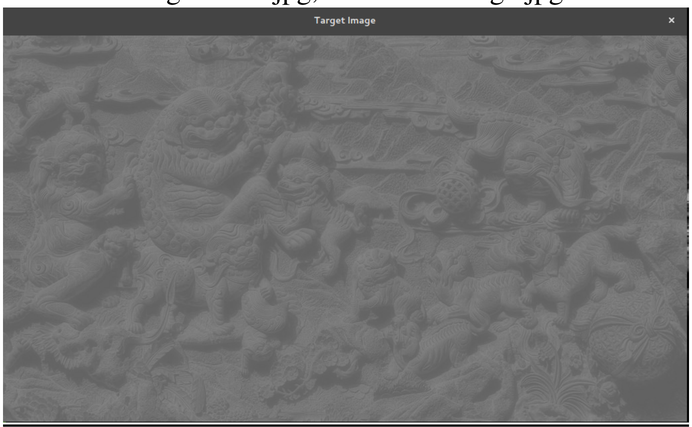
REMARK : A simple plot of the the histograms can be obtained by toggling the third argument of the function EE604A::histogram_matching() to true.

Results

Target : high.jpg, Reference : low.jpg



Target : low.jpg, Reference : high.jpg



Solution 3

Let the 8-connect neighborhood of a point be given by

v_{11}	v_{12}	v_{13}
v_{21}	v_{22}	v_{23}
v_{31}	v_{32}	v_{33}

.

In blinear interpolation, we fit a function $f(x,y)$ using weights a_{ij} in the following manner :

$$f(x,y) = \Sigma_{i=0}^{1}\Sigma_{j=0}^{1} a_{ij}x^i y^j \tag{1}$$

Let x_1, x_2, x_3 and y_1, y_2, y_3 be the x and y coordinates of the points in the neighborhood. It is clear that $x_1 \neq x_2$, $x_2 \neq x_3$, $x_3 \neq x_1$, $y_1 \neq y_2$, $y_2 \neq y_3$ and $y_3 \neq y_1$. Also, let e be the error vector.

$$We\ have, \begin{bmatrix} f(x_1,y_1) \\ f(x_1,y_2) \\ f(x_1,y_3) \\ f(x_2,y_1) \\ f(x_2,y_2) \\ f(x_2,y_3) \\ f(x_3,y_1) \\ f(x_3,y_2) \\ f(x_3,y_3) \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_1 & y_2 & x_1y_2 \\ 1 & x_1 & y_3 & x_1y_3 \\ 1 & x_2 & y_1 & x_2y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_2 & y_3 & x_2y_3 \\ 1 & x_3 & y_1 & x_3y_1 \\ 1 & x_3 & y_2 & x_3y_2 \\ 1 & x_3 & y_3 & x_3y_3 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} + e \tag{2}$$

$$\implies \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{21} \\ v_{22} \\ v_{23} \\ v_{31} \\ v_{32} \\ v_{33} \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_1 & y_2 & x_1y_2 \\ 1 & x_1 & y_3 & x_1y_3 \\ 1 & x_2 & y_1 & x_2y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_2 & y_3 & x_2y_3 \\ 1 & x_3 & y_1 & x_3y_1 \\ 1 & x_3 & y_2 & x_3y_2 \\ 1 & x_3 & y_3 & x_3y_3 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix} + e \tag{3}$$

Our aim is to determine $\begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix}$ while minimizing the the MSE, $e^T e$.

Let $f = \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \\ v_{21} \\ v_{22} \\ v_{23} \\ v_{31} \\ v_{32} \\ v_{33} \end{bmatrix}$, $X = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_1 & y_2 & x_1y_2 \\ 1 & x_1 & y_3 & x_1y_3 \\ 1 & x_2 & y_1 & x_2y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_2 & y_3 & x_2y_3 \\ 1 & x_3 & y_1 & x_3y_1 \\ 1 & x_3 & y_2 & x_3y_2 \\ 1 & x_3 & y_3 & x_3y_3 \end{bmatrix}$ and $W = \begin{bmatrix} a_{00} \\ a_{01} \\ a_{10} \\ a_{11} \end{bmatrix}$

$$\therefore\ e = f - XW \tag{4}$$

The best estimate for W is given by :

$$\hat{W} = (X^T X)^{-1} X^T f \qquad on\ minimizing\ the\ MSE,\ e^T e. \tag{5}$$

We need $X^T X$ to be invertable for a solution to exist for the given interpolation problem. Let us first show that $dim(nullspace\ of\ X) = 0$. We're going to prove that the column vectors of X are linearly independent, i.e. $c_1 X_1 + c_2 X_2 + c_3 X_3 + c_4 X_4 = 0 \implies c_1 = c_2 = c_3 = c_4 = 0$ where $X = [X_1\ X_2\ X_3\ X_4]$
On taking $c_1 X_1 + c_2 X_2 + c_3 X_3 + c_4 X_4 = 0$,

$$We\ have,\ c_1 + c_2 x_1 + c_3 y_1 + c_4 x_1 y_1 = 0 \tag{6}$$
$$c_1 + c_2 x_1 + c_3 y_2 + c_4 x_1 y_2 = 0 \tag{7}$$
$$\implies c_3(y_1 - y_2) + c_4 x_1 (y_1 - y_2) = 0 \qquad from\ (6) - (7) \tag{8}$$
$$\implies (y_1 - y_2)(c_3 + c_4 x_1) = 0 \tag{9}$$
$$\implies c_3 + c_4 x_1 = 0 \qquad \because\ y_1 \neq y_2 \tag{10}$$
$$Similiarly,\ using\ c_1 + c_2 x_2 + c_3 y_1 + c_4 x_2 y_1 = 0 \tag{11}$$
$$c_1 + c_2 x_2 + c_3 y_2 + c_4 x_2 y_2 = 0 \tag{12}$$
$$we\ get,\ c_3 + c_4 x_2 = 0 \tag{13}$$
$$\implies c_3 = c_4 = 0 \qquad from\ (10),\ (13)\ and\ \because\ x_1 \neq x_2 \tag{14}$$
$$Also,\ c_1 = c_2 = 0\ from\ (6),\ (11),\ (14)\ and\ \because\ x_1 \neq x_2 \tag{15}$$

$\therefore\ c_1 X_1 + c_2 X_2 + c_3 X_3 + c_4 X_4 = 0 \implies c_1 = c_2 = c_3 = c_4 = 0$
$\therefore\ dim(columnspace\ of\ X) = 4$ as the column vectors are linearly independent.

$$Hence,\ dim(nullspace\ of\ X) = 4 - 4 = 0 \tag{16}$$

We're now going to prove that $nullspace\ of\ X = nullspace\ of\ X^T X$.

$$Let\ y \in nullspace\ of\ X$$
$$\therefore Xy = 0$$
$$\implies X^T X y = 0$$
$$\implies y \in nullspace\ of\ X^T X$$
$$\therefore nullspace\ of\ X \subseteq\ nullspace\ of\ X^T X$$

$$Let\ y \in nullspace\ of\ X^T X$$
$$\therefore X^T X y = 0$$
$$\implies y^T X^T X y = 0$$
$$\implies (Xy)^T X y = 0$$
$$\implies Xy = 0$$
$$\implies y \in nullspace\ of\ X$$
$$\therefore nullspace\ of\ X^T X \subseteq\ nullspace\ of\ X$$

$$\therefore\ nullspace\ of\ X^T X = nullspace\ of\ X \tag{17}$$

Hence, $dim(nullspace\ of\ X^TX) = dim(nullspace\ of\ X) = 0$ using (17) and (18).
X^TX is a 4×4 square matrix. It has full rank since its nullspace is 0. $\therefore$ X^TX is invertible and hence bilinear interpolation in an 8-neighborhood can always be used to get a solution for W and (1) can be constructed.

Solution 4

We are assuming that $\eta_i(x_1, y_1)\ and\ \eta_j(x_2, y_2)$ are independent if i $\neq j\ or\ x_1 \neq x_2\ or\ y_1 \neq y_w$.

$$\therefore\ if\ i \neq j,\ E[\eta_i(x,y)\eta_j(x,y)] = 0 \tag{18}$$

$$\begin{aligned}
We\ have,\ g_i(x,y) &= f_i(x,y) + \eta_i(x,y) \\
Also,\ f_i(x,y) &= f(x,y) \\
Now,\ \hat{g}(x,y) &= \frac{1}{K}\Sigma_{i=1}^{K} g_i(x,y) \\
&= \frac{1}{K}Kf(x,y) + \frac{1}{K}\Sigma_{i=1}^{K}\eta_i(x,y) \\
&= f(x,y) + \frac{1}{K}\Sigma_{i=1}^{K}\eta_i(x,y) \\
Now,\ noise\ variance,\ E[(g(\hat{x,}y) - f(x,y))^2] &= E[(\frac{1}{K}\Sigma_{i=1}^{K}\eta_i(x,y))^2] \\
&= \frac{1}{K^2}\Sigma_{i=1}^{K}\Sigma_{j=1}^{K}E[\eta_i(x,y)\eta_j(x,y)] \\
&= \frac{1}{K^2}\Sigma_{i=1}^{K}E[\eta_i(x,y)^2] && from\ (18) \\
&= \frac{1}{K^2}\Sigma_{i=1}^{K}\sigma^2 && \because\ E[\eta_i(x,y)^2] = \sigma^2 \\
&= \frac{\sigma^2}{K} \\
q.e.d
\end{aligned}$$

Solution 5

Let $f(x,y,z) : \Re^3 \rightarrow \Re$. Let $[u\ v\ w]^T$ be the position of $[x\ y\ z]^T$ in a rotated cooridnate frame. The relationship between $[u\ v\ w]^T$ and $[x\ y\ z]^T$ is given by :

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{19}$$

$$where,\ R = [R_1\ R_2\ R_3] \tag{20}$$

$$= \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{21}$$

$$Known\ properties\ of\ R,\ |R_1| = 1 \tag{22}$$

$$|R_2| = 1 \tag{23}$$

$$R_1^T R_2 = 0 \tag{24}$$

$$R_3 = R_1 \times R_2 \tag{25}$$

$$\implies |R_3| = 1 \tag{26}$$

$$R_1^T R_3 = R_2^T R_3 = 0 \tag{27}$$

$$R^{-1} = R^T \tag{28}$$

$$We\ can\ also\ write,\ \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R^T \begin{bmatrix} u \\ v \\ w \end{bmatrix} \qquad from\ (19),\ (28) \tag{29}$$

$$Now,\ \nabla^2 f(u,v,w) = \frac{\partial^2 f}{\partial^2 u} + \frac{\partial^2 f}{\partial^2 v} + \frac{\partial^2 f}{\partial^2 w} \tag{30}$$

$$\frac{\partial f}{\partial u} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial u} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial u} + \frac{\partial f}{\partial z}\frac{\partial z}{\partial u} \tag{31}$$

$$= R_{11}\frac{\partial f}{\partial x} + R_{12}\frac{\partial f}{\partial y} + R_{13}\frac{\partial f}{\partial z} \qquad from\ (21), (29) \tag{32}$$

$$Similarily,\ \frac{\partial f}{\partial v} = R_{21}\frac{\partial f}{\partial x} + R_{22}\frac{\partial f}{\partial y} + R_{23}\frac{\partial f}{\partial z} \tag{33}$$

$$\frac{\partial f}{\partial w} = R_{31}\frac{\partial f}{\partial x} + R_{32}\frac{\partial f}{\partial y} + R_{33}\frac{\partial f}{\partial z} \tag{34}$$

$$
\begin{aligned}
\therefore\ \frac{\partial^2 f}{\partial^2 u} &= R_{11}(\frac{\partial^2 f}{\partial^2 x}\frac{\partial x}{\partial u} + \frac{\partial^2 f}{\partial y \partial x}\frac{\partial y}{\partial u} + \frac{\partial^2 f}{\partial z \partial x}\frac{\partial z}{\partial u}) \\
&+ R_{12}(\frac{\partial^2 f}{\partial x \partial y}\frac{\partial x}{\partial u} + \frac{\partial^2 f}{\partial^2 y}\frac{\partial y}{\partial u} + \frac{\partial^2 f}{\partial z \partial x}\frac{\partial z}{\partial u}) \\
&+ R_{13}(\frac{\partial^2 f}{\partial x \partial z}\frac{\partial x}{\partial u} + \frac{\partial^2 f}{\partial y \partial z}\frac{\partial y}{\partial u} + \frac{\partial^2 f}{\partial^2 z}\frac{\partial z}{\partial u}) && from\ (32)\ and\ chain\ rule \quad (35) \\
&= R_{11}^2\frac{\partial^2 f}{\partial^2 x} + R_{11}R_{12}\frac{\partial^2 f}{\partial y \partial x} + R_{11}R_{13}\frac{\partial^2 f}{\partial z \partial x} \\
&+ R_{12}R_{11}\frac{\partial^2 f}{\partial x \partial y} + R_{12}^2\frac{\partial^2 f}{\partial^2 y} + R_{12}R_{13}\frac{\partial^2 f}{\partial z \partial y} \\
&+ R_{13}R_{11}\frac{\partial^2 f}{\partial x \partial z} + R_{13}R_{12}\frac{\partial^2 f}{\partial y \partial z} + R_{13}^2\frac{\partial^2 f}{\partial^2 z} && from\ (21), (29) \quad (36) \\
Similarily,\ \frac{\partial^2 f}{\partial^2 v} &= R_{21}^2\frac{\partial^2 f}{\partial^2 x} + R_{21}R_{22}\frac{\partial^2 f}{\partial y \partial x} + R_{21}R_{23}\frac{\partial^2 f}{\partial z \partial x} \\
&+ R_{22}R_{21}\frac{\partial^2 f}{\partial x \partial y} + R_{22}^2\frac{\partial^2 f}{\partial^2 y} + R_{22}R_{23}\frac{\partial^2 f}{\partial z \partial y} \\
&+ R_{23}R_{21}\frac{\partial^2 f}{\partial x \partial z} + R_{23}R_{22}\frac{\partial^2 f}{\partial y \partial z} + R_{23}^2\frac{\partial^2 f}{\partial^2 z} && (37) \\
\frac{\partial^2 f}{\partial^2 w} &= R_{31}^2\frac{\partial^2 f}{\partial^2 x} + R_{31}R_{32}\frac{\partial^2 f}{\partial y \partial x} + R_{31}R_{33}\frac{\partial^2 f}{\partial z \partial x} \\
&+ R_{32}R_{31}\frac{\partial^2 f}{\partial x \partial y} + R_{32}^2\frac{\partial^2 f}{\partial^2 y} + R_{32}R_{33}\frac{\partial^2 f}{\partial z \partial y} \\
&+ R_{33}R_{31}\frac{\partial^2 f}{\partial x \partial z} + R_{33}R_{32}\frac{\partial^2 f}{\partial y \partial z} + R_{33}^2\frac{\partial^2 f}{\partial^2 z} && (38)
\end{aligned}
$$

On performing (36) + (37) + (38), we get using (22), (23), (23), (26) and (27)

$$
\begin{aligned}
\frac{\partial^2 f}{\partial^2 u} + \frac{\partial^2 f}{\partial^2 v} + \frac{\partial^2 f}{\partial^2 w} &= \frac{\partial^2 f}{\partial^2 u} + \frac{\partial^2 f}{\partial^2 v} + \frac{\partial^2 f}{\partial^2 w} && (39) \\
\implies \nabla^2 f(u, v, w) &= \frac{\partial^2 f}{\partial^2 u} + \frac{\partial^2 f}{\partial^2 v} + \frac{\partial^2 f}{\partial^2 w} && (40) \\
&= \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y} + \frac{\partial^2 f}{\partial^2 z} && from\ (39) \quad (41) \\
&= \nabla^2 f(x, y, z) && (42)
\end{aligned}
$$

$q.e.d$

Solution 6

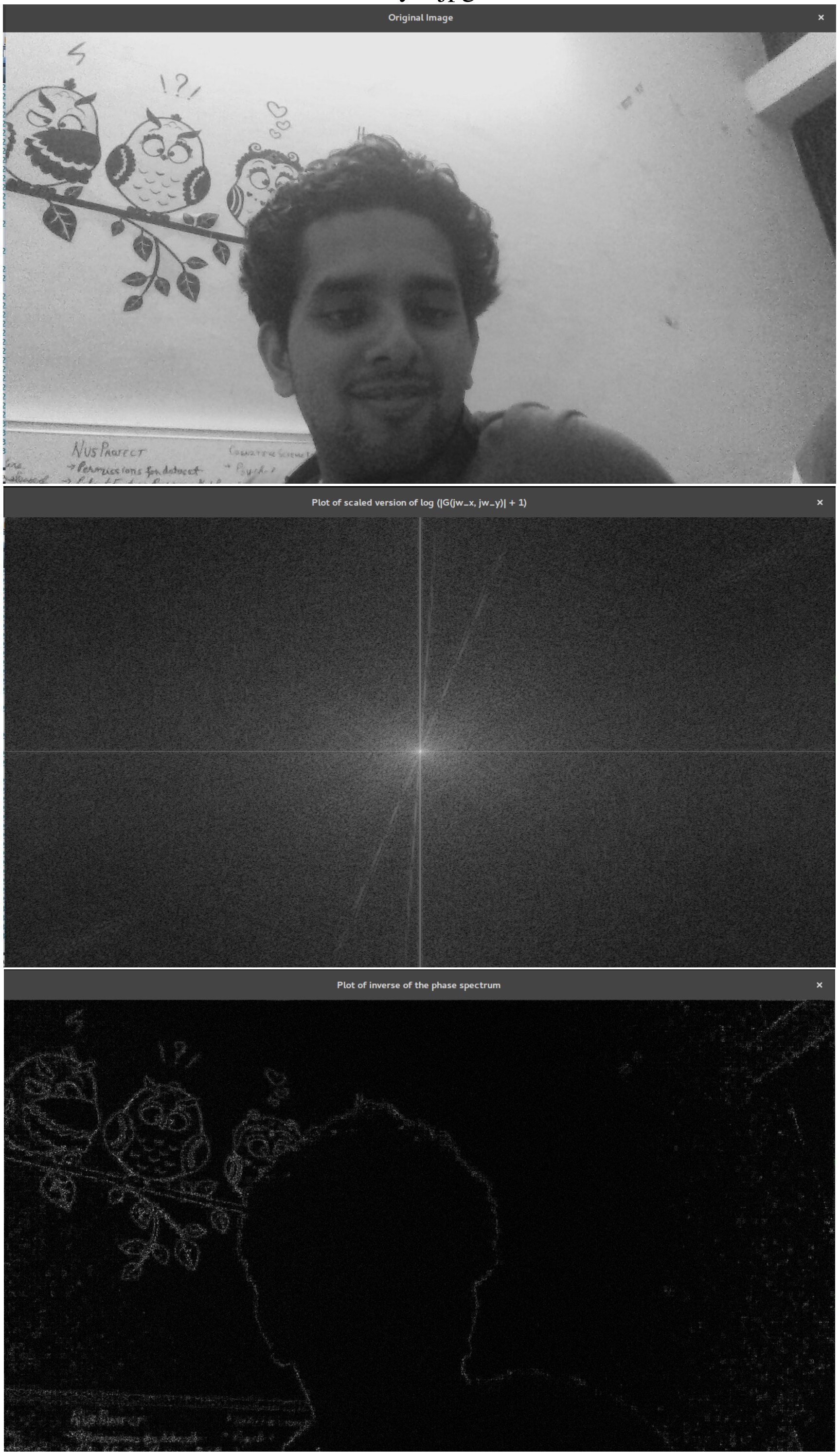
A C++ file, Q6\src \Q6.cxx, contains the code required to perform the required task.

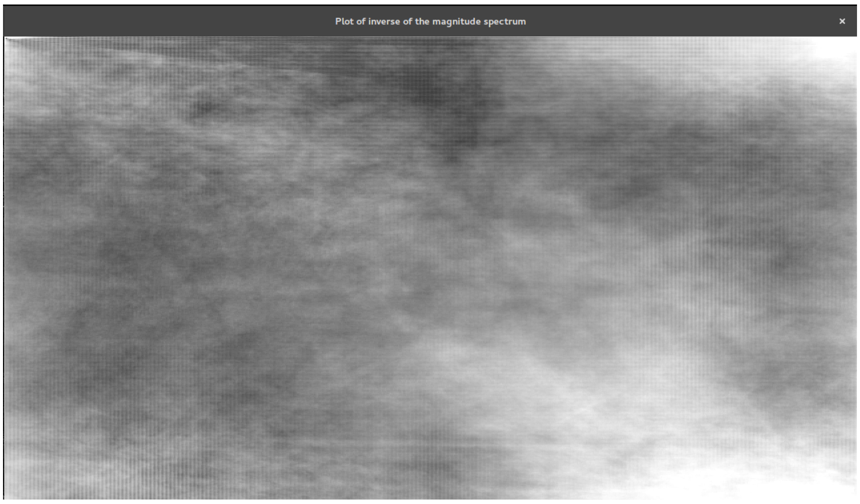
In order to build and run the code, the following steps need to be followed :

1. Enter Q6\src through terminal.
2. run build.sh (OpenCV needs to installed and a version of g++ supporting c++14 needs to be present)
3. Execute Q6 (the binary file) with the relative/absolute image path. Eg. [./Q6 ../../images/divyat.jpg] or [./Q6 ../../images/low.jpg] (On Ubuntu)
4. The program can be closed by Ctrl+C on the terminal or by pressing ‘q’ when one of the image windows is open.

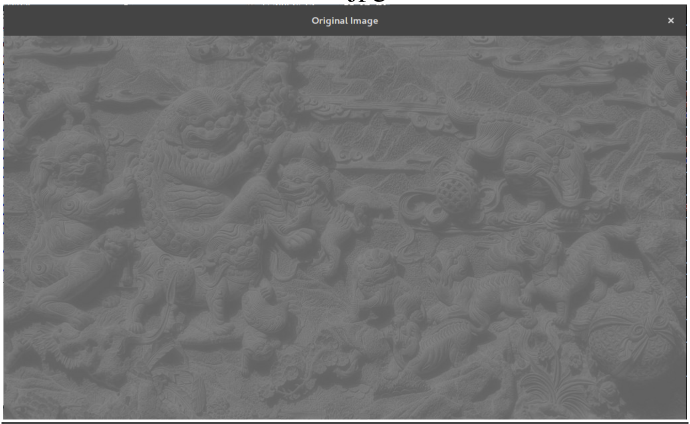
Results

divyat.jpg




Plot of inverse of the magnitude spectrum

low.jpg


Original Image


Plot of scaled version of log (|G(jw_x, jw_y)| + 1)


Plot of inverse of the phase spectrum

Observations

Reconstruction using only phase spectrum

From the above the tests, it can be inferred that on performing reconstruction using only phase information, edges are being preserved and we are losing all grey-levels of homogenous regions. From the spectrum plots, we can infer that the lower frequency components tend to have greater values than the higher frequency ones in general images. Thus, when we remove the magnitude information, the lower frequency components are getting more suppressed than the higer frequency components, i.e. the operation is similar to a high-pass filter. This premise explains the above images obatined using only the phase spectrum.

Reconstruction using only magnitude spectrum

The reconstruction using only magnitude spectrum appears to only contain information regarding the prominent greylevels present in the image.

Solution 7

Code has been written in two .cxx files :

1. Q7\src \pixel_nlm.cxx : Contains the function, EE604A::pixel_nlm(), used to perform neighborhood filter using non-local pixel values.

2. Q7\src \Q7.cxx : Calls the contained function, task(), which adds noise as needed and evaluated the neighborhood filtering method.

There is also a header file present for the EE604A::pixel_nlm() function :

1. Q7\include \pixel_nlm.cxx

In order to build and run the code, the following steps need to be followed :

1. Enter Q7\src through terminal.

2. run build.sh (OpenCV needs to installed and a version of g++ supporting c++14 needs to be present)

3. Execute Q7 (the binary file) with the relative/absolute image path. Eg. [./Q7 ../../images/small.jpg] (On Ubuntu)

4. The program can be closed by Ctrl+C on the terminal or by pressing ‘q’ when one of the image windows is open.

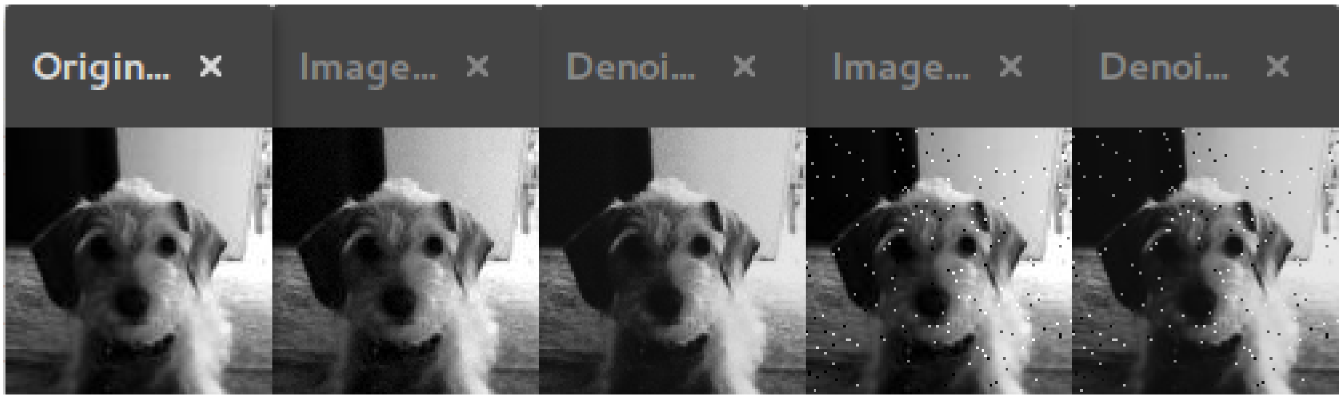


Figure 1: The images from left to right are as follows : (a) The original image (small.jpg), (b) The images with added zero-mean gaussian noise with $\sigma^2 = 0.1$, (c) The reconstucted image [b], (d) The image with impulse noise, (e) The reconstructed image [d]