Digital Image Processing HW2
Parth Pujari and Aayushi Barve

1 Question 1

Consider the image vector f. Let us represent it as

$$f = \begin{bmatrix} f_1 \\ f_2 \\ . \\ . \\ f_n \end{bmatrix}$$

Then,

$$f * w = \begin{bmatrix} w_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 \\ w_2 & w_1 & 0 & 0 & 0 & 0 & 0 & 0 & ... & 0 \\ w_3 & w_2 & w_1 & 0 & 0 & 0 & 0 & 0 & ... & 0 \\ w_4 & w_3 & w_2 & w_1 & 0 & 0 & 0 & 0 & ... & 0 \\ ... \\ 0 & ... & w_6 & w_5 & w_4 & w_3 & w_2 & w_1 & ... & 0 \\ ... \\ 0 & 0 & 0 & 0 & 0 & ... & w_6 & w_5 & w_4 & w_3 \\ 0 & 0 & 0 & 0 & 0 & ... & 0 & w_6 & w_5 & w_4 \\ 0 & 0 & 0 & 0 & 0 & ... & 0 & 0 & w_6 & w_5 \\ 0 & 0 & 0 & 0 & 0 & ... & 0 & 0 & 0 & w_6 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ f_{n-1} \\ f_n \end{bmatrix} \tag{1}$$

The reversed convolution mask slides across the rows as shown. This is an example of a **circular matrix** which is a special case of a **Toeplitz matrix**. These matrices are useful for fast matrix multiplications. An nxn toeplitz can be multiplied by a vector in $\mathcal{O}(nlog(n))$ time using **fast discrete fourier transforms**. and hence can be much more useful than brute force multiplication of the mask with the image if masks are larger. For example a mask of size 20 convolved with an image of size 500 will take $\propto (20 * 500)$ time while the toeplitz multiplcation will take $\propto (500 * log(500))$ which is $\propto (500 * 9)$.

2 Question 2

Consider the matrix form of the given equation.

$$\begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} A \begin{bmatrix} 1 \\ y \\ y^2 \\ y^3 \end{bmatrix} \tag{2}$$

Where $A_{i,j}$ is $a_{i-1,j-1}$. Given 16 points (x_k, y_k) for k : 1 to 16 with intensities I_k. Then the matrix equation for each point becomes:

$$\begin{bmatrix} & & & & .... & & & & \\ & & & & .... & & & & \\ & & & & .... & & & & \\ 1 & y_k & y_k^2 & y_k^3 & x_k & x_k y_k & .... & x_k^3 y_k^2 & x_k^3 y_k^3 \\ & & & & .... & & & & \\ & & & & .... & & & & \\ & & & & .... & & & & \end{bmatrix} \begin{bmatrix} a_{0,0} \\ a_{0,1} \\ .. \\ a_{i,j} \\ .. \\ a_{3,2} \\ a_{3,3} \end{bmatrix} = \begin{bmatrix} .. \\ .. \\ .. \\ I_k \\ .. \\ .. \\ .. \end{bmatrix} \tag{3}$$

Represent this as $Ba = v$, then a can be found by inverting B.

$$a = B^{-1}v \tag{4}$$

We require 16 neighbours because there are 16 coefficients (unknowns) and we need a 16 rowed matrix with non 0 nullity to find them.

3 Question 3

We're given a probability distribution P which is the PDF of the intensities of the first image and G which is the standard gaussian distribution. We want to find the PDF of their sum. Since the gaussian noise is independent of P, the PDF Q of the new image is as computed in the previous assignment as the continuous convolution of P and G (assumption, PDFs of images are continuous PDFs, otherwise integrals become summations)

$$Q(I_2 = z) = \int_{-\infty}^{\infty} P(I_1 = i) \cdot G(z - i)\, di \tag{5}$$

4 Question 4

(a) The Laplacian mask with a -4 at its center is not separable. Separable matrices have rank 1, the Laplacian has rank 2.
(b) The Laplacian mask with a -4 at its center cannot be implemented entirely as 1D convolutions. If it were possible, it would have to be an application of both row and column convolutions. This is because the application of only a row convolution utilizes just the pixels in the same row of the target pixel, similarly for the column convolution, which is not the case in the Laplacian. Therefore, it must be an application of a row and a column convolution (not multiple as this would bring more than 9 surrounding pixels into the picture). But since we know the Laplacian is not separable, this cannot be possible.

5 Question 5

We've learnt that convolution is both commutative and associative. We will call the $(2a+1) \times (2a+1)$ convolution as A and the image I. Then we have the following:

$$
\begin{aligned}
I' =& (A * (A * (A * (.....(A * I)..))) \\
=& (A * A * A * .....A * I)....(\text{Associativity}) \\
=& (A * A * A * ....A) * I
\end{aligned}
$$

This can be calculated recursively as $A * A$ then A on this and so on.

6 Question 6

The first part is a convolution of the gaussian with the 1D image. This can be written as:

$$
\begin{aligned}
I'(z) &= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} e^{-\frac{(x-z)^2}{2\sigma^2}} (cx+d)\, dx \\
&= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} e^{-\frac{(x-z)^2}{2\sigma^2}} (cx)\, dx + d \\
&= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} e^{-\frac{(t)^2}{2\sigma^2}} (ct)\, dx + cz + d \\
&= cz + d
\end{aligned}
$$

For the bilateral filter, we have a similar convolution integral and use the fact that the product of two gaussians is a scaled gaussian

$$
\begin{aligned}
I'(z) &= \frac{\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_r} e^{-\frac{c^2(x-z)^2}{2\sigma_r^2}} \frac{1}{\sqrt{2\pi}\sigma_s} e^{-\frac{(x-z)^2}{2\sigma_s^2}} (cx+d)\, dx}{\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_r} e^{-\frac{c^2(x-z)^2}{2\sigma_r^2}} \frac{1}{\sqrt{2\pi}\sigma_s} e^{-\frac{(x-z)^2}{2\sigma_s^2}}\, dx} \\
&= \frac{\int_{-\infty}^{\infty} \frac{s}{\sqrt{2\pi}\sigma} e^{-\frac{(x-z)^2}{2\sigma^2}} (cx+d)\, dx}{\int_{-\infty}^{\infty} \frac{s}{\sqrt{2\pi}\sigma} e^{-\frac{(x-z)^2}{2\sigma^2}}\, dx} \\
&= cz + d
\end{aligned}
$$

7 Question 7

To prove that the Laplacian is invariant under the transformation $u = x\cos\theta - y\sin\theta$ and $v = x\sin\theta + y\cos\theta$:
We note that $x = u\cos\theta + v\sin\theta$ and $y = -u\sin\theta + v\cos\theta$
Therefore, $\frac{\partial x}{\partial u} = \cos\theta$, $\frac{\partial y}{\partial u} = -\sin\theta$, $\frac{\partial x}{\partial v} = \sin\theta$, $\frac{\partial y}{\partial v} = \cos\theta$
We have,

$$\begin{aligned}
f_{uu} =& \frac{\partial^2 f}{\partial u^2} \\
=& \frac{\partial}{\partial u}\left(\frac{\partial f}{\partial x}\cdot\frac{\partial x}{\partial u} + \frac{\partial f}{\partial y}\cdot\frac{\partial y}{\partial u}\right) \\
=& \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\cdot\frac{\partial x}{\partial u} + \frac{\partial f}{\partial y}\cdot\frac{\partial y}{\partial u}\right)\cdot\frac{\partial x}{\partial u} + \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\cdot\frac{\partial x}{\partial u} + \frac{\partial f}{\partial y}\cdot\frac{\partial y}{\partial u}\right)\cdot\frac{\partial y}{\partial u} \\
=& \cos\theta\cdot\left(\frac{\partial^2 f}{\partial x^2}\cdot\cos\theta + \frac{\partial f}{\partial x}\cdot\frac{\partial}{\partial x}(\cos\theta) + \frac{\partial f}{\partial xy}\cdot(-\sin\theta) + \frac{\partial f}{\partial y}\cdot\frac{\partial}{\partial x}(-\sin\theta)\right) \\
& - sin\theta\cdot\left(\frac{\partial^2 f}{\partial y^2}\cdot(-\sin\theta) + \frac{\partial f}{\partial y}\cdot\frac{\partial}{\partial y}(-\sin\theta) + \frac{\partial f}{\partial xy}\cdot(-\sin\theta) + \frac{\partial f}{\partial y}\cdot\frac{\partial}{\partial x}(-\sin\theta)\right) \\
=& \cos^2\theta\cdot\frac{\partial^2 f}{\partial x^2} + \sin^2\theta\cdot\frac{\partial^2 f}{\partial y^2}
\end{aligned}$$

Similarly,

$$f_{vv} = \sin^2\theta\cdot\frac{\partial^2 f}{\partial x^2} + \cos^2\theta\cdot\frac{\partial^2 f}{\partial y^2}$$

Therefore,

$$f_{uu} + f_{vv} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

This gives,

$$f_{xx} + f_{yy} = f_{uu} + f_{vv} \tag{6}$$

8 Question 8

The images for both sets of noises (std=5 and 10) are shown below. We used varying kernel sizes for the bilateral filter. We observe that as σ_s increases the noise reduces while also removing edges. Larger σ_r blurs out broad features but does retain edges.
For the barbara image, ($\sigma_s = 3$, $\sigma_r = 15$) gives staircase effects near the forehead and left cheekbone. For the kodak image, we see a clear removal of the noise from the white wall, the retention of most of the drawings on the wall and a majority of the trees in the background as well as the pattern on the ceiling for $\sigma_r = 15$.

Figure 1: Noisy Image 1, Sigma_Noise = 5

Figure 2: Bilateral 2,2

Figure 3: Bilateral 0.1, 0.1

Figure 4: Bilateral 3, 15

Figure 5: Noisy Image 1, Sigma_Noise = 10

Figure 6: Bilateral 2,2

Figure 7: Bilateral 0.1, 0.1

Figure 8: Bilateral 3, 15

Figure 9: Noisy Image 1, Sigma_Noise = 5

Figure 10: Bilateral 2,2

Figure 11: Bilateral 0.1, 0.1

Figure 12: Bilateral 3, 15

Figure 13: Noisy Image 1, Sigma_Noise = 10

Figure 14: Bilateral 2,2

Figure 15: Bilateral 0.1, 0.1

Figure 16: Bilateral 3, 15

9 Question 9

We use local histogram equalizers as follows:

$$I'(r) = (L-1)\sum_{i=0}^{I(r)} P(I=i) \tag{7}$$

Where r is the pixel, $I(r)$ is the intensity at pixel r and P is the local histogram, $L-1=255$
It is worth noting that our local histogram is not continuous, nor does it take every discrete integral value of intensities, but is divided into 64 bins. Therefore, each bin accounts for $255/64 \approx 4$ intensity values.

Due to this, upper limit of the summation is not necessarily $I(r)$ but could be a value upto 4 units of intensity larger than $I(r)$. Due to this, the image may be whiter than it should and hence, we sum up the bins as usual until the bin corresponding to $I(r)$ which contributes to either half or third or a fourth of its value (a heuristic).

$$I'(r) = (L-1)\left(\sum_{bin=1}^{bin(I(r))-1} P(I \in bin) + \frac{1}{K}\cdot P(I \in bin(I(r)))\right) \tag{8}$$

The value for K we have chosen is 4 (variable K in the local_hist_eq function)

A clear first observation is that local histograms work very well for small features like leaves and twigs but not over large continuous ones like floor tiles or the sky. The local filter performs much better on the second image.

For the first image we see false patterns on the sky that arise due to local distributions that do not match the distribution of intensities of the sky. We see the same in the top left of the forest image.

Further, since all images are zero padded, the local histograms at the corners and the edges are dominated by zeros giving false light spots at these places because the excess zeros (which will lie in the first bin) will always contribute to the summation above. However, if the images were 1 (or 255) padded then the extra 255s would only contribute to the pixels with already high intensities (as only their summation would reach the last bin), while this padding would leave the low intensity pixels with lower intensities. Zero padding makes more sense for histogram equalization.

Figure 17: 7x7

Figure 18: 31x31

Figure 19: 51x51

Figure 20: 71x71

Figure 21: 7x7

Figure 22: 31x31

Figure 23: 51x51

Figure 24: 71x71

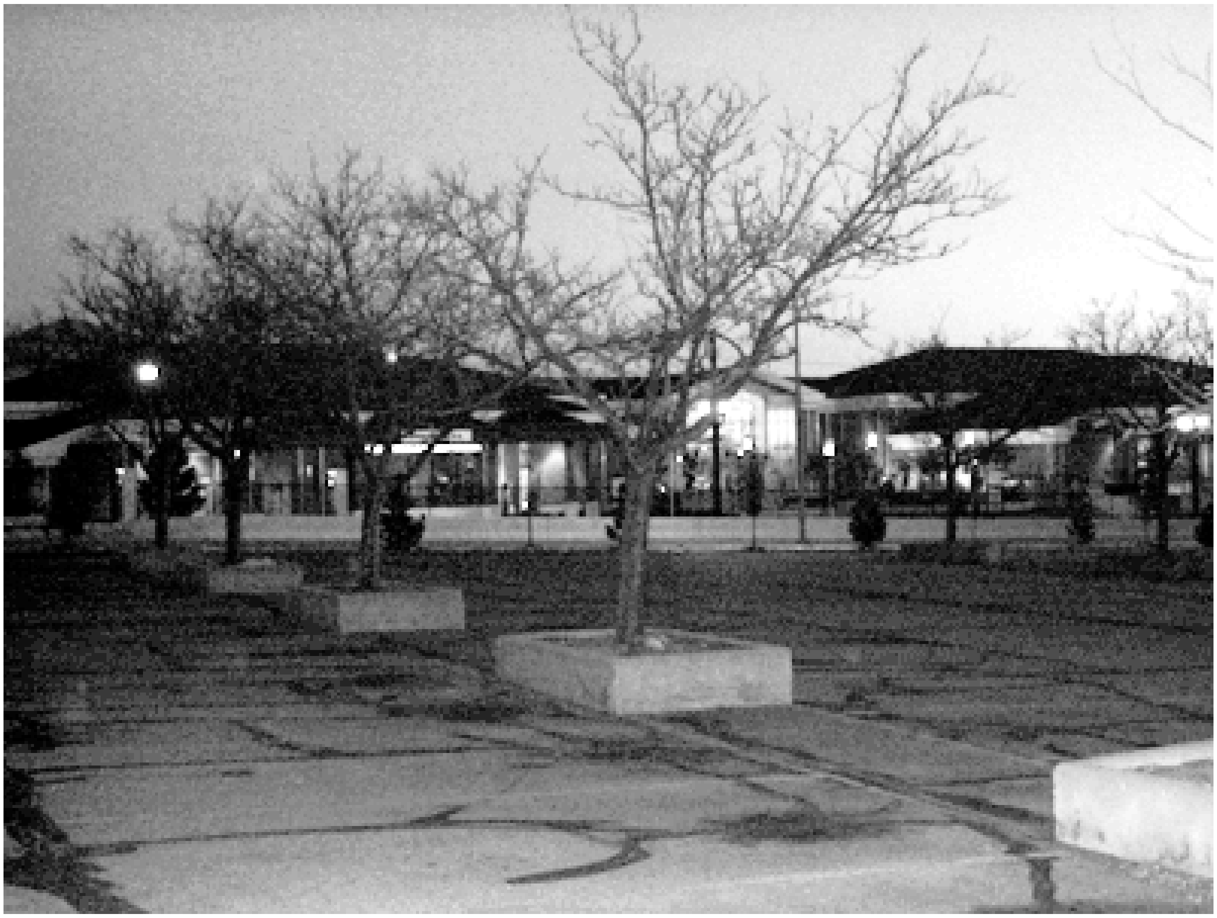

Figure 25: Global Histogram Eq

Figure 26: Global Histogram Eq