MMAT5390: Mathematical Image Processing

Assignment 4 Solutions

1. (a) The given information implies

$$H(1,1) = \frac{1}{65} \text{ and } H(2,2) = \frac{1}{17}.$$

Hence

$$\begin{cases} \dfrac{2^n}{D_0^{2n}+2^n} &= \dfrac{1}{65}, \\ \dfrac{8^n}{D_0^{2n}+8^n} &= \dfrac{1}{17}, \end{cases} \text{ and thus } \begin{cases} D_0^{2n}+2^n &= 65 * 2^n, \\ D_0^{2n}+8^n &= 17 * 8^n. \end{cases}$$

Therefore, we have $n = 1$ and $D_0^2 = 128$.

(b) Increasing the order n of the Butterworth high-pass filter sharpens its transition at the cutoff D_0. This more aggressively attenuates low frequencies while preserving high frequencies, enhancing edges and fine details. However, excessively large n can introduce ringing artifacts. Thus, n trades off edge sharpness for potential over-enhancement.

2. (a) Note that $g = h * f$, where

$$h(m,n) = \begin{cases} \frac{1}{\mu} & \text{if } n \le \mu - 1, m = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Let $0 \le u, v \le N-1$. Then

$$\begin{aligned} DFT(h)(u,v) &= \frac{1}{\mu N^2} \sum_{k=0}^{\mu-1} e^{-2\pi j \frac{kv}{N}} \\ &= \begin{cases} \frac{1}{\mu N^2} \frac{1-e^{-2\pi j \frac{\mu v}{N}}}{1-e^{-2\pi j \frac{v}{N}}} & \text{if } e^{-2\pi j \frac{v}{N}} \neq 1 \\ \frac{1}{N^2} & \text{if } e^{-2\pi j \frac{v}{N}} = 1 \end{cases} \\ &= \begin{cases} \frac{1}{\mu N^2} \frac{e^{-\pi j \frac{\mu v}{N}} (e^{\pi j \frac{\mu v}{N}} - e^{-\pi j \frac{\mu v}{N}})}{e^{-\pi j \frac{v}{N}} (e^{\pi j \frac{v}{N}} - e^{-\pi j \frac{v}{N}})} & \text{if } u \notin N\mathbb{Z} \\ \frac{1}{N^2} & \text{if } v \in N\mathbb{Z} \end{cases} \\ &= \begin{cases} \frac{1}{\mu N^2} e^{-\pi j \frac{(\mu-1)v}{N}} \frac{\sin \frac{\mu \pi v}{N}}{\sin \frac{\pi v}{N}} & \text{if } v \neq 0 \\ \frac{1}{N^2} & \text{if } v = 0. \end{cases} \end{aligned}$$

Since $DFT(h * f) = N^2 DFT(h) \odot DFT(f)$,

$$H(u,v) = N^2 DFT(h)(u,v) = \begin{cases} \frac{1}{\mu} e^{-\pi j \frac{(\mu-1)v}{N}} \frac{\sin \frac{\mu \pi v}{N}}{\sin \frac{\pi v}{N}} & \text{if } v \neq 0 \\ 1 & \text{if } v = 0 \end{cases}.$$

(b) The parameter μ controls the blur severity in the degradation function $H(u,v)$. As μ increases, the magnitude $H(u,v)$ decays more rapidly for $v \neq 0$, suppressing high frequencies more aggressively and worsening blur, while the oscillatory term $sin(\frac{\mu \pi v}{N})$ introduces more nulls where frequencies are entirely lost. Thus, larger μ intensifies blur by attenuating finer details and introducing periodic frequency nulls.

3. Note that $I_1(2,2) = e^{-1} DFT(I)(2,2)$ and $H_1(2,2) = e^{-\frac{2^2+2^2}{ab}} = e^{-1}$.
We have $ab = 8$

Also

$$
\begin{aligned}
I_{Haar} &= \tilde{H} I \tilde{H}^T \\
&= \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{pmatrix} \begin{pmatrix} a & b & b & b \\ b & a & b & b \\ b & b & a & b \\ b & b & b & a \end{pmatrix} \begin{pmatrix} 1 & 1 & \sqrt{2} & 0 \\ 1 & 1 & -\sqrt{2} & 0 \\ 1 & -1 & 0 & \sqrt{2} \\ 1 & -1 & 0 & -\sqrt{2} \end{pmatrix} \\
&= \begin{pmatrix} a+3b & 0 & 0 & 0 \\ 0 & a-b & 0 & 0 \\ 0 & 0 & a-b & 0 \\ 0 & 0 & 0 & a-b \end{pmatrix}
\end{aligned}
$$

Hence, $I_{Haar}(1,1) = a - b = 2$. By solving the above equations, we have $a = 4$ and $b = 2$.

4. (a) Since

$$
\begin{aligned}
\frac{\partial f}{\partial x} &\approx \frac{1}{2h}(f(x+h,y) - f(x-h,y)) \\
\frac{\partial f}{\partial y} &\approx \frac{1}{2h}(f(x,y+h) - f(x,y-h))
\end{aligned}
$$

We can approximate the followings

$$
\begin{aligned}
\frac{\partial^2 f}{\partial x^2} &\approx \frac{1}{2h}(f_x(x+h,y) - f_x(x-h,y)) \\
&\approx \frac{1}{4h^2}(f(x+2h,y) - 2f(x,y) + f(x-2h,y)) \\
\frac{\partial^2 f}{\partial y^2} &\approx \frac{1}{2h}(f_y(x,y+h) - f_y(x,y-h)) \\
&\approx \frac{1}{4h^2}(f(x,y+2h) - 2f(x,y) + f(x,y-2h))
\end{aligned}
$$

Hence, by letting $h = 1$, the discrete Laplacian operator

$$
\Delta f = \frac{1}{4}(f(x+2,y) + f(x-2,y) + f(x,y+2) + f(x,y-2) - 4f(x,y))
$$

(b) Note that we can write the discrete Laplacian operator as a convolution $\Delta f = h * f$, where

$$
h(x,y) = \begin{cases} -1 & \text{if } (x,y) = (0,0) \\ \frac{1}{4} & \text{if } (x,y) = (\pm 2, 0) \text{ or } (0, \pm 2) \\ 0 & \text{otherwise.} \end{cases}
$$

Then $DFT(\Delta f) = N^2 DFT(h) \odot DFT(f)$, and thus

$$
\begin{aligned}
H(u,v) &= N^2 DFT(h)(u,v) \\
&= \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} h(x,y) e^{-2\pi j \frac{ux+vy}{N}} \\
&= -1 + \frac{1}{4}(e^{2\pi j \frac{2u}{N}} + e^{-2\pi j \frac{2u}{N}} + e^{2\pi j \frac{2v}{N}} + e^{-2\pi j \frac{2v}{N}}) \\
&= -1 + \frac{1}{2}\cos\frac{4\pi u}{N} + \frac{1}{2}\cos\frac{4\pi v}{N} \\
&= -\sin^2\frac{2\pi u}{N} - \sin^2\frac{2\pi v}{N} \\
&= -4\cos^2\frac{\pi u}{N}\sin^2\frac{\pi u}{N} - 4\cos^2\frac{\pi v}{N}\sin^2\frac{\pi v}{N}
\end{aligned}
$$

5. (optional)
MATLAB code:

```
recon = (h * U') * freq * (h * U');
```

Python code:

```
U_H = np.conj(U.T)
recon = U_H @ freq @ U_H
```

6. (optional)
MATLAB code:

```
filter = exp(-dist/(2*sigma^2));
```

Python code:

```
filter = np.exp(-dist / (2 * sigma**2))
```