

1. 习题

1.1 离散傅里叶变换对

对于离散傅里叶变换对，常数项 $1/MN$ 可以包含在 DFT 里，也可以包含在 IDFT 里，甚至可以分为两个常数项 $1/\sqrt{MN}$ 分别放在正变换和反变换前面，可以通过输入一个单位冲激函数（单位脉冲），得到 DFT 的结果 a。

如果 $a = 1$ ，则说明常数项 $1/MN$ 包含在 IDFTL 里面

如果 $a = 1/MN$ ，则说明常数项 $1/MN$ 包含在 DFT 里面

如果 $a = 1/\sqrt{MN}$ ，则说明分为两个常数项 $1/\sqrt{MN}$ 分别放在正变换和反变换前面。

1.2 傅里叶频谱

图 1(b) 和图 1(c) 对应的傅里叶频谱是一样的，根据平移性质，空间域的平移不改变频域里的傅里叶频谱

1.3 频率域滤波器

- i. 找出一个滤波器 $H(u, v)$ ，使得它在频率域与一下空间滤波器等价：

$$\frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$H(u, v)$ 如下矩阵

$$\begin{bmatrix} 1 + 0.00000000e + 00j & -0.125 - 2.16506351e - 01j & -0.125 + 2.16506351e - 01j \\ -0.125 - 2.16506351e - 01j & 0.25 - 0.4330127025e + 00j & -0.5 - 0.3330669075e - 3.75j \\ -0.125 + 2.16506351e - 01j & -0.5 - 0.3330669075e - 3.75j & 0.25 + 0.4330127025e + 00j \end{bmatrix}$$

- ii. $H(u, v)$ 是高通滤波器还是低通滤波器？请给出相应的数学证明。

是高通滤波器，可以看到 u, v 值较小时，频谱值较小， u, v 值逐渐增大时，频谱值较大。

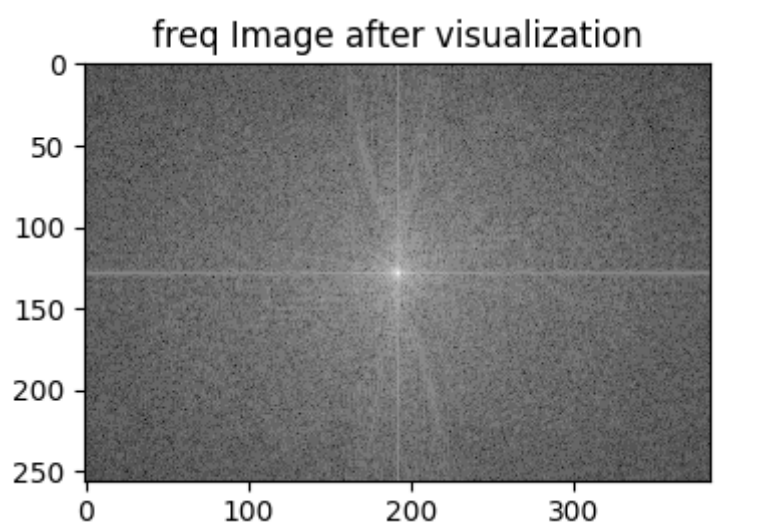
2. 编程题



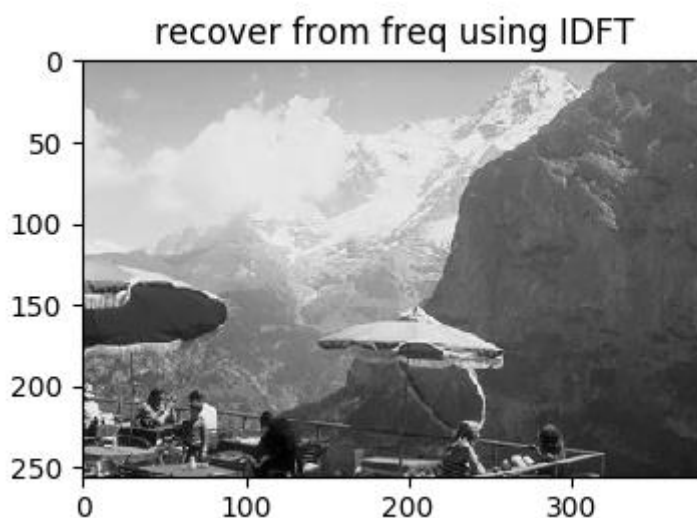
Original Image

2.2 傅里叶变换

1. 将中心化后的傅里叶频谱粘贴到报告里



2. 对 1.的结果做离散傅里叶反变换 (IDFT)，将实部粘贴到报告里。



3. 描述如何实现 DFT/IDFT:

DFT:

- i. 首先将原图像进行 0 延拓，延拓后的高度和宽度均是原来的两倍
- ii. 中心化，图像中的每个像素的灰度值都乘上 $(-1)^{(x+y)}$
- iii. 根据公式 $F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$ ，进行运算，由于语言对矩阵乘法计算有所优化，采用矩阵乘法的形式来进行公式的运算

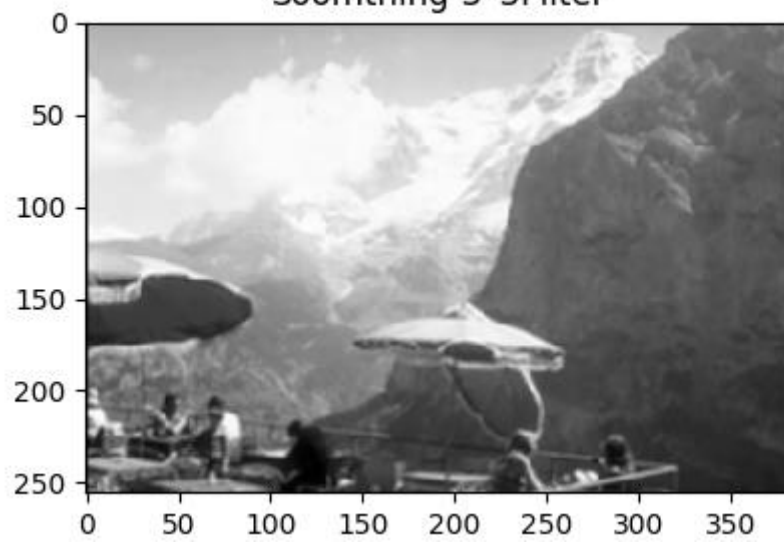
IDFT:

- i. 根据公式 $MNf(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}$ ，先将 DFT 得到的结果，进行计算，使用矩阵乘法的形式。
- ii. 将 I 得到的结果除以 MN ，得到 $f(x, y)$ ，并乘上 $(-1)^{(x+y)}$ 去中心化，得到最后的结果。

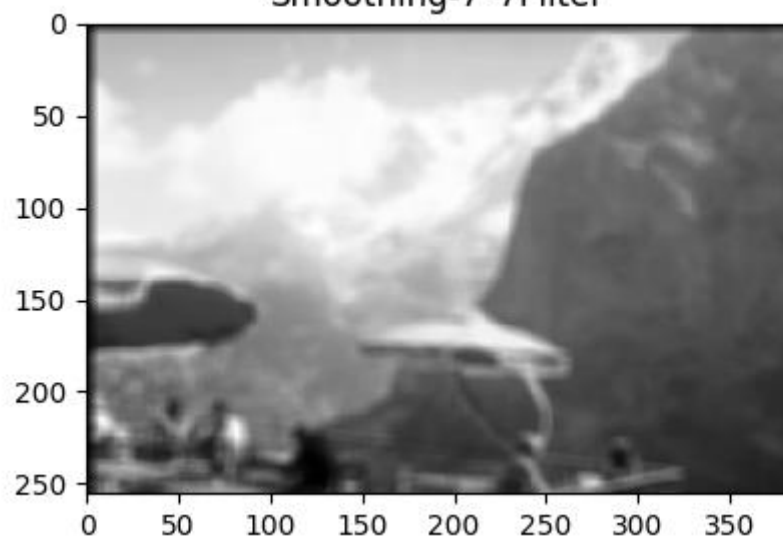
2.3 频率域滤波

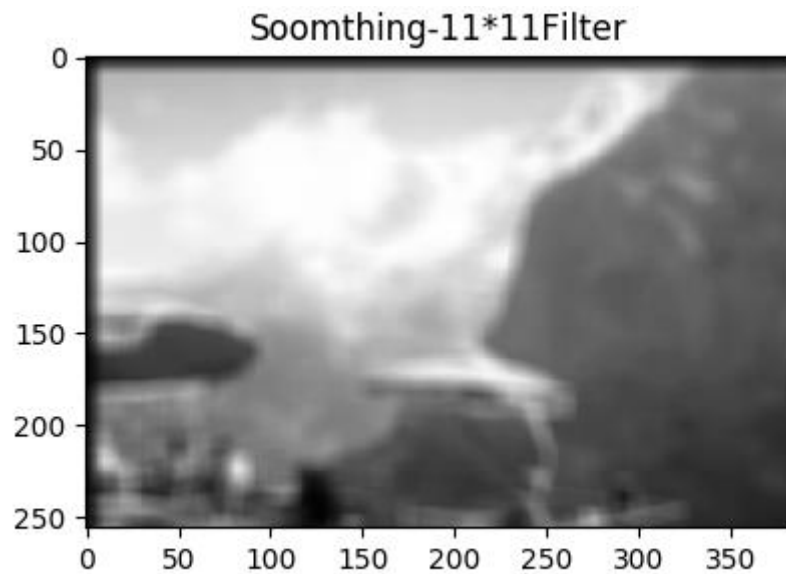
1. 分别用 3×3 , 7×7 , 11×11 的均值滤波器来平滑输入

Soomthing-3*3Filter

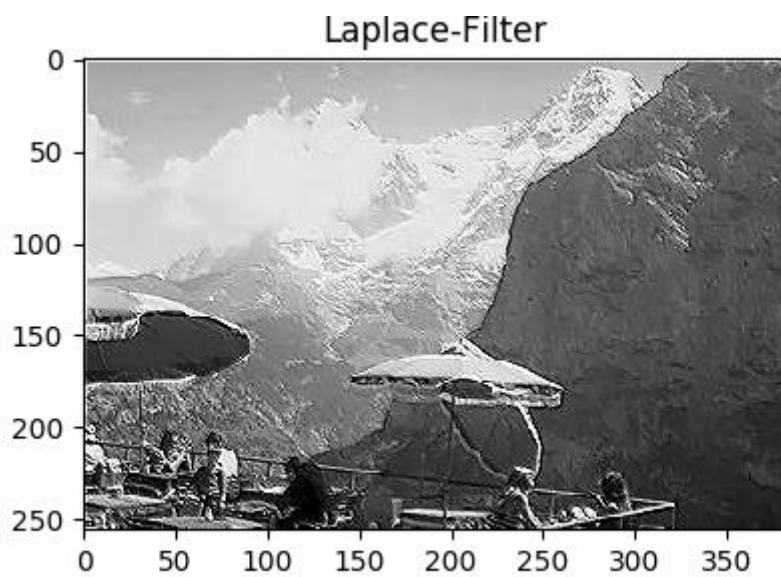


Smoothing-7*7Filter





2. 用 3×3 的拉普拉斯滤波器锐化图像



3. 详细描述如何实现滤波操作：

- i. 对输入的图像做 0 延拓，延拓后的图像的高度和宽度都为原来图像的两倍。并将图像置于延拓后的图像左上角（其它像素点灰度值均置为 0），每个像素点乘以 $(-1)^{(x+y)}$ ，做中心化操作
- ii. 对输入的滤波器做 0 延拓，延拓后的滤波器的高度和宽度都与 I 中延拓后的图像一致。并将原滤波器置于延拓后的

图像左上角（其它像素点灰度值均置为 0），每个像素点乘以 $(-1)^{(x+y)}$ ，也做中心化操作

- iii. 调用 `dft2d(with parameter 0)` 将 I 和 II 延拓后的矩阵做二维傅里叶变换，得到相应的傅里叶变换矩阵 $F(u, v)$ 和 $H(u, v)$ 。并通过公式 $G(u, v) = H(u, v)F(u, v)$ （阵列乘法）
- iv. 调用 `idft`（`dft2d with parameter 1`），将 $G(u, v)$ 做反傅里叶变换，得到结果 $g(x, y)$ （去中心化已在此步完成）

最后，本次实验使用 python3 作为编程语言。使用到了 numpy, PIL, matplotlib 和 cv2 等库，最终在 Windows10 的操作系统环境下运行通过。