

# HW3: 频率域滤波

数字图像处理课程助教，中山大学

**欢**迎进入数字图像处理第三次作业！疯狂的傅里叶变换来了！你准备好接受挑战了吗？第三次作业的难度比前两次作业有所增加，但是我们相信你们会克服重重困难。提交作业的时候需要提交一份报告（PDF 格式）和所有与作业相关的代码。再次声明：抄袭=挂科。除此之外，迟交作业将会被扣除本次作业至少 30% 以上的成绩。

## 1 习题

请回答以下问题，并将你的答案写入报告中。

### 1.1 离散傅里叶变换对（10 分）

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，甚至可以分成两个常数项  $1/\sqrt{MN}$  分别放在正变换和反变换前面。假如你想调用一个用于计算二维 DFT 对的函数接口，但是你看不到源码，文档也没提及相关信息，请问如何才能找到包含该项（或这些项）的位置？

### 1.2 傅里叶频谱（15 分）

图 1(b)和图 1(c)是对图 1(a)进行延拓(padding)操作得到的，它们填充的 0 值的数量相同但位置不同。请问，图 1(b)和图 1(c)对应的傅里叶频谱(Fourier spectrum)是否一样呢？请说明理由。

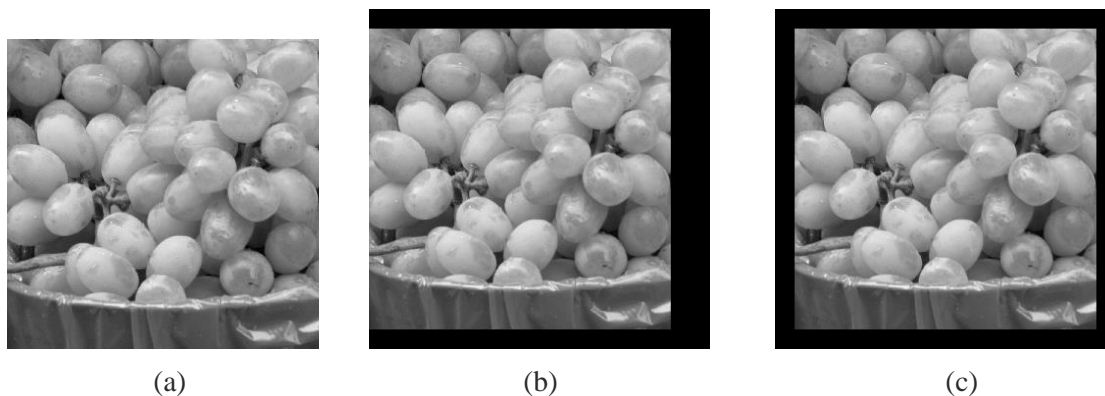


图 1：傅里叶频谱图

## 1.3 频率域滤波器（15 分）

- 1.（5 分）找出一个滤波器  $H(u,v)$ ，使得它在频率域与以下空间滤波器等价：

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

- 2.（10 分） $H(u,v)$  是高通滤波器还是低通滤波器？请给出相应的数学证明。

## 2 编程题

编程完成以下三个任务，并在报告里回答相应的问题。提交作业的时候，请不要忘了提交相关的代码。

### 2.1 要求

**输入** 请下载附件“hw3.zip”（与 hw2.zip 一样），解压并根据学号的后两位数选择相应的图片，这张图将是你的编程题目的初始输入。例如，如果你的学号是“15110349”，那么你应该选择“49.png”作为你的输入。如有需要，你可以通过 Photoshop 将图片的格式转换成 BMP，JPEG 等。

**编程语言** 允许使用任何编程语言，鼓励使用 Python 和 Matlab。

**其他** 你还需要注意以下几点：

1. 你可以调用第三方库来操作图像，但是对于以下的编程题目要求实现的功能，你必须自己实现。例如，你能够调用 Matlab 的“imread”函数来读取图像，但是你不能直接使用 Matlab 的“fft2”函数来进行傅里叶变换。
2. 良好的用户体验是值得鼓励的，但是我们不会因此给你太多的加分。所以不要花太多时间在用户体验上，毕竟这不是一门人机交互课程。
3. 请保持代码工整。糟糕的代码风格将会让你丢掉至多 20% 的分数。
4. 提交作业时，请将所有相关的代码、依赖、可执行文件、输入、输出和报告都打包成一个 zip 压缩包交上来，命名格式为“编程语言\_学号\_姓名\_第几次作业”，如“matlab\_12345678\_张三\_hw3.zip”。报告需要注明运行的平台和使用的库的版本号，如 OpenCV3.2。

## 2.2 傅里叶变换（30 分）

实现一个函数来完成 2 维离散傅里叶变换和反变换(DFT 和 IDFT)。函数格式为 “dft2d (input\_img, flags) → output\_img”, 该函数返回 DFT / IDFT 的结果。“flags” 是一个参数, 用于指定 DFT 或 IDFT。如有需要, 你可以修改该函数的格式。

请载入你的输入图像, 用你的程序来完成:

1. (10 分) 对输入图像做离散傅里叶变换 (DFT), 并且把中心化后的傅里叶频谱粘贴到报告里。
2. (5 分) 对上一个问题的结果做离散傅里叶反变换 (IDFT), 并把实部粘贴到报告里。注意: 实部与输入图像非常相似。(请思考其中的原因。)
3. (15 分) 详细描述你是如何实现 DFT / IDFT 的, 也就是说, 针对 “dft2d” 函数进行算法说明, 字数不能超过两页。请集中在算法方面, **不要简单地复制/粘贴代码**。

## 2.3 附加题: 快速傅里叶变换（50 分）

这是一个可选的任务。这里你需要自己编程实现快速傅里叶变换(FFT)。函数格式是 “fft2d(input\_img, flags) → output\_img”, 该函数返回 FFT/IFFT 的结果。“flags” 是一个参数, 用于指定 FFT 或 IFFT。

“fft2d” 的输出结果与 “dft2d” 非常相似。然而, 你实现的函数可能只能处理尺寸大小为 2 的幂次方的图片, 因此我们建议你通过缩放图像到合适大小来解决这个问题。

请载入你的输入图像, 用你的程序来完成:

1. (15 分) 对输入图像做快速傅里叶变换 (FFT), 并且把中心化后的傅里叶频谱粘贴到报告里。
2. (10 分) 对上一个问题的结果做快速傅里叶反变换 (IFFT), 并把实部粘贴到报告里。
3. (25 分) 详细描述你是如何实现 FFT / IFFT 的, 也就是说, 针对 “fft2d” 函数进行算法说明, 字数不能超过三页。

你可能注意到 “fft2d” 的运行速度比 “dft2d” 有明显的提升。但是, 假如你用纯 Python(或者 Matlab,...), 你的 “fft2d” 有可能非常慢, 请不要在意。

## 2.4 频率域滤波（30 分）

实现一个函数来完成频率域滤波。函数格式是 “filter2d\_freq(input\_img, filter) → output\_img”, 这里的 “filter” 是给定的滤波器。如有需要, 可以修改函数格式。根据卷积定理, 你需要对给定的图片和滤

波器做离散傅里叶变换/快速傅里叶变换 (DFT / FFT)，然后将他们乘起来，最后做离散傅里叶反变换/快速傅里叶反变换来获得滤波结果。因此，基于“dft2d” (或者“fft2d”)来实现“filter2d\_freq”应该还是比较简单的。当然，你需要注意一些细节，比如缠绕误差(wraparound error) (参加书本的第 4.7 节)

请载入你的输入图像，用你的程序"filter2d\_freq"来完成：

1. (12 分) 分别用  $3 \times 3$ ,  $7 \times 7$  和  $11 \times 11$  的均值滤波器来平滑你输入的图像，将相应的三个输出结果粘贴到报告里。
2. (4 分) 用  $3 \times 3$  的拉普拉斯滤波器来锐化你输入的图像 (本上有 4 种拉普拉斯滤波器，参见图 3.37，你可以使用其中任意一种)，并将输出结果放在报告中。
3. (14 分) 详细描述你是如何实现滤波操作的，也就是说，针对“filter2d\_freq”函数进行算法说明，字数不能超过两页。

注意：

1. 前两个问题的结果理应与第二次作业的结果非常相似(假如第二次作业的实现正确)。

### 3 参考

0	1	0	1	1	1
1	-4	1	1	-8	1
0	1	0	1	1	1
0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

a b  
c d

**图 3.37** (a) 执行式(3.6-6)定义的离散拉普拉斯变换所用的滤波器掩模，(b)用于执行该公式的扩展掩模，它包括对角线邻域，(c)和(d)其他两种拉普拉斯的实现