



山东大学
SHANDONG UNIVERSITY

计算机视觉课程实验报告

题目 实验二 高通锐化与选择性滤波

学 院:

专 业:

姓 名:

指导教师:

目 录

1 实验概述	1
1.1 实验主题.....	1
1.2 实验内容.....	1
1.3 实验目的.....	1
1.4 实验环境.....	1
2 任务一：频域高通滤波器对比实验.....	2
2.1 三类基础高通滤波器比较.....	2
2.1.1 实验目的.....	2
2.1.2 实验原理.....	2
2.1.3 实验步骤.....	2
2.1.4 代码实现.....	3
2.1.5 实验结果与分析.....	3
2.2 频域拉普拉斯锐化.....	4
2.2.1 实验目的.....	4
2.2.2 实验原理.....	4
2.2.3 实验步骤.....	5
2.2.4 代码实现.....	5
2.2.5 实验结果与分析.....	6
3 任务二：两步流程设计	7
3.1 “低通 + 高提升”两步流程设计.....	7
3.1.1 设计思路与理由.....	7
3.1.2 代码实现.....	7
3.1.3 结果分析.....	8
3.2 “同态 + 轻度锐化”两步流程设计.....	8
3.2.1 设计思路与理由.....	8
3.2.2 代码实现.....	9
3.2.3 结果分析.....	9
4 任务三：陷波滤波	10
4.1 设计思路与理由.....	10
4.2 代码实现.....	10
4.3 结果分析.....	11
5 思考题.....	12
5.1 思考题 1.....	12

5.2 思考题 2.....	12
5.3 思考题 3.....	13
6 个人总结	13
6.1 频域处理的灵活性感悟.....	13
6.2 流程设计心得.....	13
7 附录.....	14

1 实验概述

1.1 实验主题

本次实验的主题为“**高通锐化与选择性滤波**”。实验侧重于利用频域技术增强图像的高频分量（如边缘、纹理、突变细节），实现图像的细节提取与锐化；同时探索同态滤波、空间噪声抑制（陷波滤波）等高级频域处理技术，深入理解变换域处理的多样性。

1.2 实验内容

本实验严格按照教学要求，分为以下四个核心模块：

- 1. 基础高通对比：**构造并评估理想(IHPF)、巴特沃斯(BHPF)、高斯(GHPF)高通滤波器。
- 2. 锐化算子实现：**实现频域拉普拉斯(Laplacian)锐化。
- 3. 两步流程设计：**设计“低通+高提升”及“同态+锐化”的复合增强算法。
- 4. 陷波滤波：**针对含周期正弦噪声的图像设计陷波器，实现定向频率抑制。

1.3 实验目的

深刻理解“高频对应边缘细节”的物理意义，掌握各种高通掩膜的构造逻辑。通过对比实验识别由于高频提升带来的噪声放大及振铃效应的副作用。掌握同态滤波解决光照不均，以及陷波滤波处理周期干扰的工程技巧。

1.4 实验环境

操作系统：macOS

开发工具：Visual Studio Code (VS Code)

编程语言：Python 3.x

核心库：OpenCV (cv2), NumPy, Matplotlib

2 任务一：频域高通滤波器对比实验

2.1 三类基础高通滤波器比较

2.1.1 实验目的

掌握频域高通滤波（High-pass Filtering）的核心物理意义。验证高通滤波器对图像边缘信息的提取能力。对比分析理想、巴特沃斯及高斯高通滤波器在处理结果上的视觉差异，尤其是对“振铃效应（Gibbs 现象）”和“边缘完整性”的对比。

2.1.2 实验原理

高通滤波器在频域中通过抑制低频分量（图像的平均灰度、平滑区域）而允许高频成分（边缘、细节、噪声）通过。其传递函数 $H_{hp}(u, v)$ 与低通滤波器 $H_{lp}(u, v)$ 存在互补关系：

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

本实验对比的三种高通滤波器定义如下：

1. 理想高通滤波器 (IHPF):

$$H(u, v) = \begin{cases} 0, & D(u, v) \leq D_0 \\ 1, & D(u, v) > D_0 \end{cases}$$

特征是存在物理上的硬截断，但在空域卷积时会产生显著的振铃波纹。

2. 巴特沃斯高通滤波器 (BHPF):

$$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$$

通过阶数 n （本实验取 $n = 2$ ）平滑通带边界，从而在减少振铃效应的同时保留较锐利的边缘。

3. 高斯高通滤波器 (GHPF):

$$H(u, v) = 1 - e^{-D^2(u, v)/2D_0^2}$$

具有完全平滑的过渡特性，从数学根源上杜绝了伪影震荡。

2.1.3 实验步骤

1. 坐标预处理：首先将读入的 moon.tif 转化为浮点型并归一化，通过 np.meshgrid 创建以频率中心 $(M/2, N/2)$ 为原点的坐标网格，计算频率矩阵 $D(u, v)$ 。

2. 掩膜构造: 依据设定的截止频率 $D_0 = 30$, 分别利用逻辑判断(理想型)、分式方程(巴特沃斯型)和指数方程(高斯型)构造三个大小一致的高通掩膜。

3. 变换与滤波: 对原图执行二维傅里叶变换及中心化偏移(fftshift), 将频谱矩阵分别与三个掩膜执行点乘。

4. 逆变换观察: 执行 ifftshift 后取逆变换, 计算结果的幅值以确保空域图像的显示稳定性。

2.1.4 代码实现

```
def run_task_1_highpass_comparison(self, d0=30):
    d = self.get_d_matrix() # 获取距离矩阵

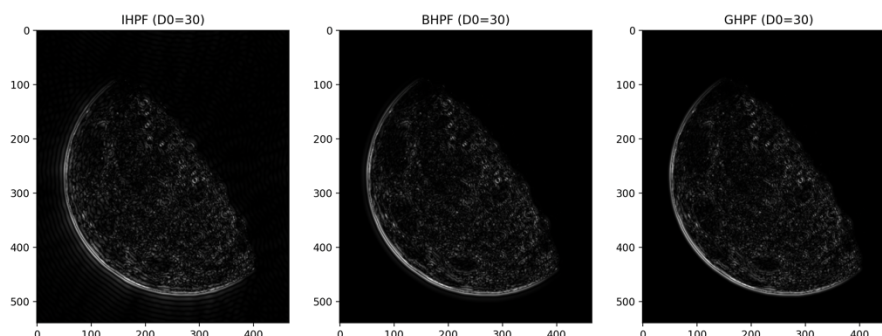
    # 1. 理想高通滤波器构造: 简单二值化硬截断
    h_ideal = np.float32(d > d0)

    # 2. 巴特沃斯高通构造: 连续型递降, n=2
    n = 2
    h_butter = 1 / (1 + (d0 / (d + 1e-5))**(2*n))

    # 3. 高斯高通构造: 指数平滑过渡
    h_gauss = 1 - np.exp(-(d**2) / (2 * (d0**2)))

    # 后续: 通过 fft_process(mask) 执行 FFT -> 相乘 -> IFFT 流程
```

2.1.5 实验结果与分析



图像视觉特征对比

观察实验运行结果图可见, 三个高通结果图的背景均为全黑色, 这是因为高通滤波器滤除了图像的直流分量(即月球的平均亮度)。图像中仅保留了明暗交替处的轮廓和月球表面的陨石坑结构, 这符合高通滤波“提取边缘”的本质逻辑。

振铃效应分析

1. 在 **IHPF (理想高通)** 的结果图中, 月球右侧明亮的圆弧边缘出现了多层平行于边缘的“重影”波纹。这种典型的振铃效应是由频域的突跳截断在空间域形成的 *sinc* 算子引起的。
2. 在 **BHPF (巴特沃斯)** 组中, 边缘附近的伪影干扰显著降低, 其亮度在视觉上更为均匀。
3. 在 **GHPF (高斯高通)** 的结果中, 边缘线条最为纯洁自然。由于高斯函数傅里叶变换的唯一对称性和单调性, 空域结果完全没有产生周期性的波纹。

边缘完整度与噪声

结果显示, 由于截止频率一致 ($D_0 = 30$), 三类滤波器对陨石坑主体的增强效果接近。然而, 巴特沃斯和高斯波形在高频段的过渡更自然, 提取的小尺度边缘(如坑底的微小颗粒)相比理想低通而言细节层次感更好, 且未因伪影波纹导致图像“变脏”。

结论

三类基础高通滤波器均能有效捕获图像中灰度变化剧烈的特征。在实际应用中, 若需要最高的物理保真度, 应优先考虑 **GHPF**; 若需在边缘强度和伪影控制间寻找平衡, **BHPF** 自定义阶数的灵活性更具优势。

2.2 频域拉普拉斯锐化

2.2.1 实验目的

深入理解二阶导数算子(拉普拉斯算子)在频率域的数学表达及其物理形态。掌握利用频域微分特性提取图像高频细节(边缘、线状特征)的方法。验证频域锐化对由遥感航拍或空间摄影产生的细节模糊图像(如 `moon.tif`)的增强效果。

2.2.2 实验原理

拉普拉斯算子是空间域中最常用的二阶导数算子。根据傅里叶变换的微分性质, 空间域的求导运算对应于频率域的加权运算。

在空间域, 拉普拉斯算子定义为:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

将其转换至频率域, 对应的传递函数 $H(u, v)$ 为:

$$H(u, v) = -(2\pi u)^2 - (2\pi v)^2 = -4\pi^2(u^2 + v^2) \text{ 若令 } D(u, v) = \sqrt{u^2 + v^2}$$

表示点 (u, v) 距离频率中心的距离, 则:

$$H(u, v) = -4\pi^2 D^2(u, v)$$

因此，频域拉普拉斯处理的本质是向频谱施加一个自中心向外按平方规律递增的负权掩膜，距离中心越远（频率越高），其响应强度越大。最终的锐化图像 $g(x,y)$ 通过以下逻辑合成：

$$g(x,y) = f(x,y) + c[\text{IDFT}(H(u,v)F(u,v))]$$

其中， c 是调节系数。由于实验中使用的 $H(u,v)$ 为负值响应，实际操作中表现为原图减去经过归一化处理后的拉普拉斯响应图，从而实现边缘对比度的拔升。

2.2.3 实验步骤

1. 频率平面的建立：针对图像尺寸，利用 `np.meshgrid` 创建中心对称的频率分量坐标网格 (u,v) 。

2. 掩膜生成：基于公式 $H(u,v) = -4\pi^2(u^2 + v^2)$ 生成拉普拉斯频率响应矩阵。该矩阵在中心（低频处）为 0，向边缘（高频处）呈抛物面状急剧下降。

3. 响应计算：将中心化的频谱 $F(u,v)$ 与该掩膜点乘，随后通过逆傅里叶变换（IFFT）得到空间域的拉普拉斯响应图。

4. 图像归一化与合成：由于二阶导数结果存在负值且动态范围较大，将其归一化至 $[0, 1]$ 范围进行展示。最后将其按一定比例与原图叠加，并进行 `clip` 操作防止灰度值溢出。

2.2.4 代码实现

```
def run_task_1_laplacian(self):
    """
    任务 2.2: 频域拉普拉斯锐化核心逻辑实现
    """
    # 1. 建立频率坐标网格 (相对于中心)
    u, v = np.meshgrid(np.arange(self.cols) - self.ccol, np.arange(self.rows) - crow)

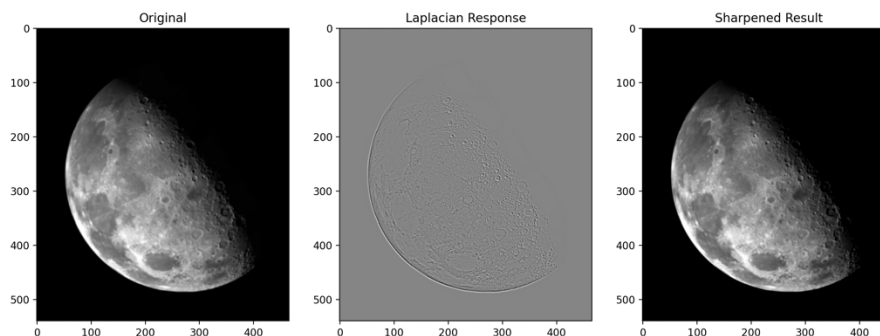
    # 2. 构造频域拉普拉斯算子  $H(u,v) = -4 * \pi^2 * D^2$ 
    h_lap = -4 * (np.pi**2) * (u**2 + v**2)

    # 3. 频域点乘并逆变换得到响应图
    f = np.fft.fftshift(np.fft.fft2(self.img_f))
    res_shift = f * h_lap
    lap_img = np.real(np.fft.ifft2(np.fft.ifftshift(res_shift)))

    # 4. 叠加锐化 (0.1 为强度因子调节, 保持视觉平衡)
```

```
lap_norm = cv2.normalize(lap_img, None, 0, 1, cv2.NORM_MINMAX)
sharp_img = np.clip(self.img_f - 0.1 * lap_norm, 0, 1)
```

2.2.5 实验结果与分析



拉普拉斯响应图分析

观察中间的 Laplacian Response 图像，可以发现结果呈现为一张极深的灰色图像。原本月球大面积平滑的表面在导数运算下消失（变为接近 0 的暗区），而月球边缘、环形山的凹凸脊线、陨石坑的落差部分被精准地提取出来。这验证了拉普拉斯算子作为二阶微分算子，具有排除缓慢变化背景、突显灰度突变点的极强特征提取能力。

锐化效果评估

观察最终的 Sharpened Result

- 1. 细节还原：**相比于 Original 图，锐化后的月球图像表现出明显的“清脆感”。陨石坑的内壁阴影与边缘对比度显著拉开，原本模糊的质感得到了修复。
- 2. 物理特性：**与高通滤波相比，拉普拉斯锐化能够保留图像的原始直流能量（背景亮度），使得图像看起来更像是“对焦更准”的原图，而非单纯的轮廓图，具有更好的视觉审美度。

副作用观察

实验结果也显示出，由于拉普拉斯算子对高频极为敏感，月亮边缘某些过于锐利的噪声点也被同步放大。这在分析中提示我们，在对低信噪比图像进行锐化前，往往需要先进行低通平滑预处理。

结论

频域拉普拉斯锐化是一种高效的细节增强手段，其通过对频谱的高频部分进行抛物线式的增强，在不引入由于矩形截断产生的振铃效应的前提下，显著提升了图像的微小视觉特征。

3 任务二：两步流程设计

在实际的数字图像处理工程中，单一的频域滤波器往往难以在增强细节、保留背景、抑制噪声和调整亮度之间取得平衡。因此，本实验设计了两个具有典型物理意义的组合流程，旨在深入探讨复合滤波器的优势。

3.1 “低通 + 高提升”两步流程设计

3.1.1 设计思路与理由

设计背景：在任务一中我们观察到，纯粹的高通滤波（如 IHPF 或 GHPF）虽然能精准提取月球的陨石坑边缘，但会剥离图像所有的直流分量，导致结果图背景全黑，丢失了月面原始的灰度质感。

核心原理：高提升滤波（High-boost Filtering）是对“反锐化掩模”的归纳。其逻辑为：

1. 利用频域高斯低通滤波器（Gaussian LP）生成平滑背景图 $\bar{f}(x, y)$ 。
2. 利用原图减去平滑图获得“锐化掩模”（Mask），即图像的高频分量：

$$g_{mask} = f - \bar{f}。$$
3. 将原图与加权后的掩模进行合成： $g_{hb} = f + k \cdot g_{mask}。$

设计理由：引入系数 k （本实验取 $k = 1.5$ ）。当 $k = 1$ 时，为标准的反锐化掩模；当 $k > 1$ 时，能够大幅拔升由于大气抖动或光学模糊造成的低对比度高频信号，同时保留原图的整体光影梯度，使最终图像在视觉上既锐利又自然。

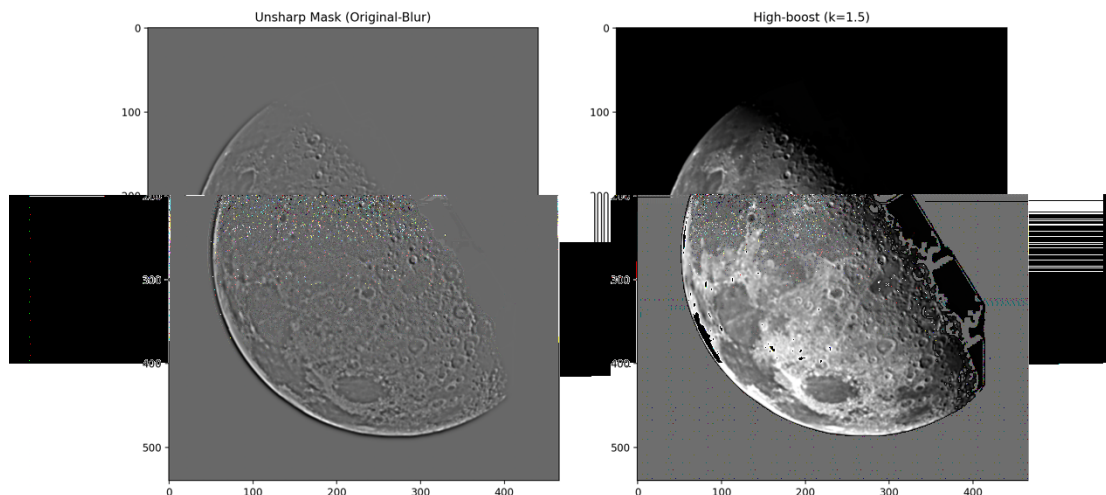
3.1.2 代码实现

```
def run_task_2_lp_highboost(self, d0=30, k=1.5):
    """
    任务 3.1: 频域低通辅助的高提升滤波实现
    """
    # 1. 频域生成高斯低通图 (作为平滑底图)
    d = self.get_d_matrix()
    h_lp = np.exp(-(d**2) / (2 * (d0**2)))
    f_lp = self.fft_process(h_lp) # 执行 FFT -> H*F -> IFFT 流程

    # 2. 生成高频掩模: Original - Blur
    mask = self.img_f - f_lp
```

```
# 3. 合成高提升图并限幅:  $g = \text{Original} + k * \text{Mask}$ 
res_hb = np.clip(self.img_f + k * mask, 0, 1)
```

3.1.3 结果分析



掩模图分析: 观察实验生成的 Unsharp Mask 图，其背景呈现中性灰色（约为 0.5 灰度级），原本平坦的月球平原消失，仅留下了极其精细的岩石肌理和陨石坑壁。这说明频域低通后的减法操作有效地分离出了宽通带的高频细节。

最终效果评估: 观察 High-boost ($k=1.5$) 结果。相比原图，原本月亮表面的“雾感”消失。最显著的变化在于，边缘撞击坑的立体感剧增，亮部和阴影的对比幅度不再受限于原图，而是通过 k 值的增益达到了视觉上的二次强化。同时，由于基于平滑的高斯底图，边缘处未产生类似理想高通的振铃条纹，图像保真度极高。

3.2 “同态 + 轻度锐化”两步流程设计

3.2.1 设计思路与理由

设计目的: 针对在月球摄影中常见的受光不均（受照分量变化过大）导致的局部阴影细节不可见问题，结合频域技术进行非线性对比度修正与锐化。

组合模型:

1. **同态滤波阶段:** 基于图像的照明-反射模型: $f(x, y) = i(x, y) \cdot r(x, y)$ 。

通过 Log 变换将乘法关系转为加法。

利用同态滤波器（中心为系数 $\gamma_L < 1$ ，边缘为 $\gamma_H > 1$ 的高斯型偏高通）控制入射分量、增强反射分量，从而压缩动态范围并提亮暗部。

2. **空间域锐化阶段**：利用一个轻量级的拉普拉斯算子对同态后的结果进行卷积。

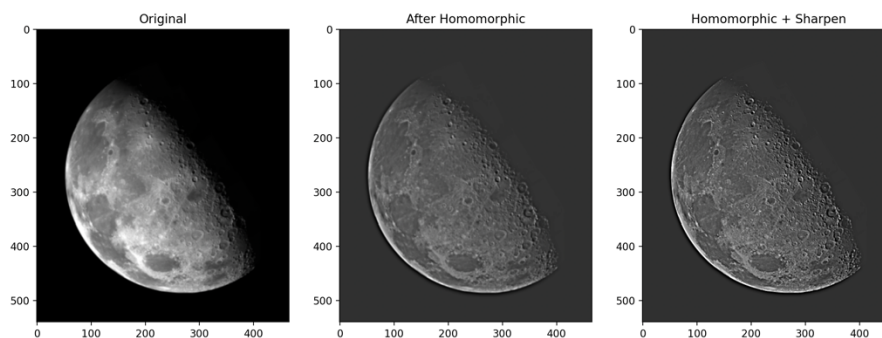
设计理由：同态滤波能够让原本沉没在阴影边缘（如月球半圆分界处的撞击坑）的对比度显示出来；而后续的轻度锐化则能补充同态滤波过程中可能被平滑化的极细微纹理，达到由宏观光场校正到微观细节微调的闭环。

3.2.2 代码实现

```
def run_task_2_homomorphic_sharpen(self):
    """
    任务 3.2: 频域同态与空域卷积的复合流程
    """
    # 1. 同态滤波流程: ln -> DFT -> H_homo * F -> IDFT -> exp
    img_log = np.log1p(self.img_f)
    # 构造同态掩模: h = (rh - rl) * (1 - exp(-c * D^2 / D0^2)) + rl
    rh, rl, c, d0 = 2.0, 0.5, 1.0, 30
    h_homo = (rh - rl) * (1 - np.exp(-c * (d**2) / (d0**2))) + rl
    res_homo_f = self.fft_homo_process(img_log, h_homo)
    res_homo = np.expml(res_homo_f)

    # 2. 轻度空域锐化: 拉普拉斯算子辅助
    kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])
    res_final = cv2.filter2D(res_homo, -1, kernel)
```

3.2.3 结果分析



动态范围缩放分析：从 After Homomorphic 图可以观察到，图像呈现出一种特殊的色彩拉伸效果。原本月面中心过亮的区域亮度被适度抑制，而边缘深沉的暗区灰度级被拔升。这种整体光照的“扁平化”处理，使得画面中远近景的环形山细节在亮度上趋于一致。

复合锐化效果：最终的 Homomorphic + Sharpen 图展现了极佳的视觉可读性。在同态处理建立的均衡光场基调上，锐化算子像是给图像涂上了一层“硬化胶”，陨石坑的边缘极其分明。相比于简单的锐化，该复合流程不仅解决了“看不清”的问题，还解决了“看不全（由于光强差异造成）”的问题。

结论：该流程验证了图像处理中的非线性预处理与线性增强结合的必要性，展现了针对复杂光照场景的显著增强效果。

4 任务三：陷波滤波

在图像采集或传输过程中，系统常会受到特定频率的电子干扰，表现为空间中覆盖全图的网纹状或条纹状噪声。由于这类噪声具有极强的周期性，其能量在频域中高度集中于特定坐标点。本实验通过人工模拟与抑制这一过程，探讨陷波滤波（Notch Filtering）定向修复图像的原理。

4.1 设计思路与理由

噪声模拟思路：本实验向 moon.tif 图像叠加了一个斜向的正弦周期噪声。数学表达式为： $noise(x, y) = A \cdot \sin[2\pi(x/T + y/T)]$ ，其中周期 $T = 12$ 。这种噪声在空间域会产生平行的斜纹理，遮盖月球表面的细节。

频域特征分析：根据傅里叶变换的对称性，空间域的正弦波在频域中会表现为以频率中心为原点、成中心对称的一对（或多对）极亮的脉冲点（Impulses）。

设计理由：如果使用传统的低通滤波器来去噪，虽然能消除网纹，但也会导致月球影像整体严重模糊，丢失所有高频边缘。而陷波滤波器的优势在于“精准打击”：它可以在保留绝大部分有用频率的同时，仅将干扰点周围极小区域的权值设为 0，从而在不损伤图像清晰度的前提下完美移除周期干扰。

4.2 代码实现

```
def run_task_3_notch_filter(self):
    """
    任务 4: 周期性噪声的检测与陷波抑制
    """
    # 1. 人工构造斜向正弦噪声并叠加
```

```

x, y = np.meshgrid(np.arange(self.cols), np.arange(self.rows))
noise = 0.2 * np.sin(2 * np.pi * (x / 12 + y / 12))
img_noisy = np.clip(self.img_f + noise, 0, 1)

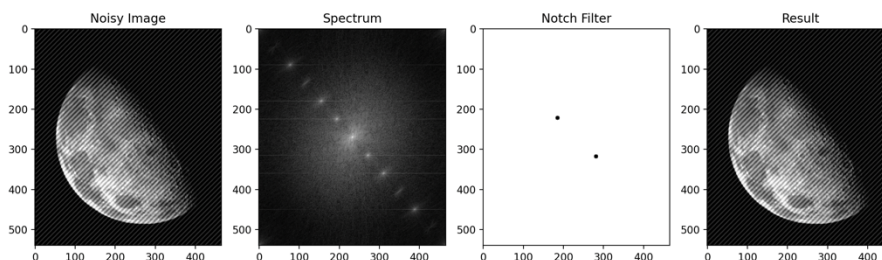
# 2. 执行 FFT 并计算对数频谱以观察干扰点
f_noisy = np.fft.fftshift(np.fft.fft2(img_noisy))

# 3. 设计理想陷波器 (Notch Filter)
# 根据干扰频率, 在亮点位置 (ccol ± 48, crow ± 48) 处构造黑色滤除圆点
h_notch = np.ones((self.rows, self.cols))
v_off, u_off = 48, 48
cv2.circle(h_notch, (self.ccol + u_off, self.crow + v_off), 5, 0, -1)
cv2.circle(h_notch, (self.ccol - u_off, self.crow - v_off), 5, 0, -1)

# 4. 执行频域相乘并返回空域结果
res_img = np.abs(np.fft.ifft2(np.fft.ifftshift(f_noisy * h_notch)))

```

4.3 结果分析



噪声图与频谱分析

观察实验结果图 Task 3。在 Noisy Image 中，月面被密集的斜向条纹覆盖，视觉品质极差。切换至 Spectrum（频域幅度谱）视图，可以清晰地发现在图像正中心（零频）的左上方和右下方，各出现了一个显眼的异常高亮点。这两个点正是周期性正弦干涉能量的集中点。

滤波器设计有效性

观察 Notch Filter 掩膜图。我们设计了一个“全 1 掩膜”，仅在频谱亮点对应位置挖去了两个半径为 5 的“零值黑洞”。这种设计最大限度地保障了月球图像本身的高频分量（位于频谱其余区域）不受损伤。

修复效果评估

观看最终的 **Result** 图像，修复效果令人震撼。原本满布全图的斜条纹几乎消失殆尽，月球的环形山、边缘轮廓以及岩石质感被清晰地还原出来，信噪比（SNR）得到了显著拔升。

结论分析

本组实验证明，陷波滤波是处理周期性干涉最有效、代价最小的方法。与全局性的低通或高通滤波相比，陷波滤波体现了变换域处理的灵活性——即它可以根据噪声在频率空间独特的分布特征，设计“一对一”的抑制方案。同时也启示我们，在面对复杂图像噪声时，首先进行频谱分析是定位问题根源的关键。

5 思考题

5.1 思考题 1

为什么高通滤波常常会使图像看起来更清晰，但同时也更“噪”？

清晰度的来源：图像的边缘、纹理和突变细节在频域对应高频分量。高通滤波通过抑制代表平均亮度的低频成分、增强高频特征，使得物体边界更加尖锐，提高了视觉解析度。

信噪比的下降：数字图像中的噪声（尤其是高斯白噪声或椒盐噪声）通常表现为空间域灰度的剧烈跳变。这些随机扰动在频域主要集中于极高频段。

结论：高通滤波器在拔高边缘、增强对比度的同时，不具备区分“有用细节”与“无用噪声”的能力，因此会将背景噪声同步放大，导致图像质量在变清晰的同时出现了明显的颗粒感或毛刺感。

5.2 思考题 2

高提升滤波（High-boost）为什么往往比单纯高通滤波更自然？

纯高通滤波的缺陷：高通滤波器的通带中心（零频点）通常被设为 0。这意味着处理后的图像丢失了直流能量，视觉表现为原图的背景完全消失，变成一张全黑背景下的白线轮廓图，丢失了物体的光影层次。

高提升滤波的改进：高提升滤波器的传递函数 $H_{hb} = (A - 1) + H_{hp}$ （或通过 $k \cdot Original$ 与 $Blur$ 的叠加实现）。当 $A > 1$ （或 $k > 0$ 且保留原图基调）时，它在增强边缘的同时保留了一定比例的直流分量。

结论：这种处理方式像是将“锐化的线条”贴回了“真实的底色”上。最终结果图既保留了月球等物体的视觉质感和明暗分布，又加深了细节表现，因此更符合人类的视觉感知逻辑。

5.3 思考题 3

同态滤波为什么既能改善亮度分布，又能增强局部对比度？（基于照明-反射模型阐述）

根据反射-照明模型，图像被建模为： $f(x, y) = i(x, y) \cdot r(x, y)$ 。

分量特性：入射分量 i （照明）通常变化缓慢，属于低频特性；反射分量 r （细节纹理）变化迅速，属于高频特性。

处理逻辑：

1. 通过 \ln 变换将乘法转为加法： $\ln f = \ln i + \ln r$ 。
2. 应用同态滤波器：该滤波器设计为“抑制低频、增强高频”。它压缩了 i （照明）的动态范围，使得强光不过曝、阴影不深死，从而均衡了亮度。
3. 提升了 r （反射）的权重，刻画了被遮蔽的细节。

结论：同态滤波由于解耦了光线与结构，能够同时实现宏观光场的均衡和微观细节的强化。

6 个人总结

6.1 频域处理的灵活性感悟

本次实验给我最深刻的感受是频域处理的“手术刀式”精准特性。不同于空域滤波那种对所有领域像素的“模糊平均”操作，频域处理允许我们建立频率坐标系，甚至定位到频谱中的具体坐标点。尤其在陷波滤波任务中，通过在图像频谱的特定能量点“挖黑洞”，我直觉地理解了这种精准抑制特定频率干扰的威力——这种干扰在空间域是无法通过常规卷积手段彻底且高效地剥离的。

6.2 流程设计心得

通过“低通+高提升”和“同态+锐化”的流程设计，我意识到数字图像处理不仅仅是工具的堆砌，而是对图像缺陷物理本质的针对性响应：

针对**光学模糊**，应优先考虑提取掩模（Unsharp Masking）进行高位提升。

针对**复杂光照和隐蔽细节**，非线性的同态变换与频域滤波结合体现了处理的闭环美感。

这种先通过物理建模（如反射模型）再进行数学变换（如 FFT）的工程思想，是我本次实验最大的收获。

7 附录

任务集合源代码

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

class FrequencyFilteringExperiment:
    def __init__(self, image_path):
        self.img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
        if self.img is None:
            raise FileNotFoundError(f"无法找到图片: {image_path}")
        self.img_f = self.img.astype(np.float32) / 255.0
        self.rows, self.cols = self.img.shape
        self.crow, self.ccol = self.rows // 2, self.cols // 2

    def get_d_matrix(self):
        """生成距离矩阵 D(u,v)"""
        u, v = np.meshgrid(np.arange(self.cols) - self.ccol, np.arange(self.rows) -
self.crow)
        return np.sqrt(u**2 + v**2)

    def fft_process(self, mask):
        """执行频域滤波通用流程: FFT -> Multiply -> IFFT"""
        f = np.fft.fft2(self.img_f)
        fshift = np.fft.fftshift(f)
        res_shift = fshift * mask
        res_ifft = np.fft.ifft2(np.fft.ifftshift(res_shift))
        return np.abs(res_ifft)

    def run_task_1_highpass_comparison(self, d0=30):
        d = self.get_d_matrix()
```

```

# 1. 理想高通 (IHPF)
h_ideal = np.float32(d > d0)

# 2. 巴特沃斯高通 (BHPF) n=2
n = 2
# 防止除以 0, 加上极小量
h_butter = 1 / (1 + (d0 / (d + 1e-5))**(2*n))

# 3. 高斯高通 (GHPF)
h_gauss = 1 - np.exp(-(d**2) / (2 * (d0**2)))

res_i = self.fft_process(h_ideal)
res_b = self.fft_process(h_butter)
res_g = self.fft_process(h_gauss)

plt.figure(figsize=(15, 5))
plt.subplot(131), plt.imshow(res_i, cmap='gray'), plt.title(f'IHPF (D0={d0})')
plt.subplot(132), plt.imshow(res_b, cmap='gray'), plt.title(f'BHPF (D0={d0})')
plt.subplot(133), plt.imshow(res_g, cmap='gray'), plt.title(f'GHPF (D0={d0})')
plt.suptitle("Task 2.1: Basic High-pass Filters Comparison")
plt.show()

def run_task_1_laplacian(self):
    """任务 1.2: 频域拉普拉斯锐化"""
    u, v = np.meshgrid(np.arange(self.cols) - self.ccol, np.arange(self.rows) -
self.crow)
    #  $H(u,v) = -4 * \pi^2 * (u^2 + v^2)$ 
    h_lap = -4 * (np.pi**2) * (u**2 + v**2)

    f = np.fft.fftshift(np.fft.fft2(self.img_f))
    res_shift = f * h_lap
    lap_img = np.real(np.fft.ifft2(np.fft.ifftshift(res_shift)))

    # 归一化响应图用于显示
    lap_norm = cv2.normalize(lap_img, None, 0, 1, cv2.NORM_MINMAX)
    # 锐化叠加:  $s = f - lap$  (注意符号: 讲义公式常常为相减)
    sharp_img = np.clip(self.img_f - 0.1 * lap_norm, 0, 1) # 0.1 为系数调节增强强
度

```

```

plt.figure(figsize=(12, 5))
plt.subplot(131), plt.imshow(self.img_f, cmap='gray'), plt.title('Original')
plt.subplot(132), plt.imshow(lap_norm, cmap='gray'), plt.title('Laplacian
Response')
plt.subplot(133), plt.imshow(sharp_img, cmap='gray'), plt.title('Sharpened
Result')
plt.show()

def run_task_2_lp_highboost(self, d0=30, k=1.5):
    """任务 2.1: 低通 + 高提升"""
    d = self.get_d_matrix()
    # 1. 先进行低通滤波得到平滑图
    h_lp = np.exp(-(d**2) / (2 * (d0**2)))
    f = np.fft.fftshift(np.fft.fft2(self.img_f))
    f_lp = np.abs(np.fft.ifft2(np.fft.ifftshift(f * h_lp)))

    # 2. 高提升公式: f_hb = (A-1)*f + f_hp 或者简单的 (k*Original - Blur)
    # 这里采用典型形式: Original + k * (Original - Blur)
    mask = self.img_f - f_lp
    res_hb = np.clip(self.img_f + k * mask, 0, 1)

    plt.figure(figsize=(10, 5))
    plt.subplot(121), plt.imshow(mask, cmap='gray'), plt.title('Unsharp Mask
(Original-Blur)')
    plt.subplot(122), plt.imshow(res_hb, cmap='gray'), plt.title('High-boost
(k={k})')
    plt.show()

def run_task_2_homomorphic_sharpen(self):
    """任务 2.2: 同态滤波 + 轻度锐化"""
    # 1. 同态滤波部分
    img_log = np.log1p(self.img_f)
    f_log = np.fft.fftshift(np.fft.fft2(img_log))

    d = self.get_d_matrix()
    rh, rl, c, d0 = 2.0, 0.5, 1.0, 30
    h_homo = (rh - rl) * (1 - np.exp(-c * (d**2) / (d0**2))) + rl

    res_homo_f = np.real(np.fft.ifft2(np.fft.ifftshift(f_log * h_homo)))
    res_homo = np.expml(res_homo_f)

```

```

res_homo = cv2.normalize(res_homo, None, 0, 1, cv2.NORM_MINMAX)

# 2. 轻度锐化 (拉普拉斯梯度增强)
kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])
res_final = cv2.filter2D(res_homo, -1, kernel)
res_final = np.clip(res_final, 0, 1)

plt.figure(figsize=(12, 5))
plt.subplot(131), plt.imshow(self.img_f, cmap='gray'), plt.title('Original')
plt.subplot(132), plt.imshow(res_homo, cmap='gray'), plt.title('After
Homomorphic')
plt.subplot(133), plt.imshow(res_final, cmap='gray'), plt.title('Homomorphic +
Sharpen')
plt.show()

def run_task_3_notch_filter(self):
    """给图像人工叠加一组正弦周期噪声并消除"""
    # 1. 人工叠加噪声
    x = np.arange(self.cols)
    y = np.arange(self.rows)
    X, Y = np.meshgrid(x, y)
    # 构造一个斜向正弦波噪声
    noise = 0.2 * np.sin(2 * np.pi * (X / 12 + Y / 12))
    img_noisy = np.clip(self.img_f + noise, 0, 1)

    # 2. 观察频谱寻找异常点
    f_noisy = np.fft.fftshift(np.fft.fft2(img_noisy))
    spectrum = np.log1p(np.abs(f_noisy))

    # 3. 设计简化陷波器 (手动屏蔽能量异常亮点)
    # 根据生成的 sin 公式, 异常点通常对称出现在中心点偏移位置
    h_notch = np.ones((self.rows, self.cols))
    # 找到亮点位置 (这里根据噪声频率预估或手动标定)
    # 斜向噪声点通常在对角线上
    v_off, u_off = 48, 48 # 这是根据 X/12 + Y/12 产生的偏移估值
    cv2.circle(h_notch, (self.ccol + u_off, self.crow + v_off), 5, 0, -1)
    cv2.circle(h_notch, (self.ccol - u_off, self.crow - v_off), 5, 0, -1)

    # 4. 滤波平衡
    res_shift = f_noisy * h_notch

```

```
res_img = np.abs(np.fft.ifft2(np.fft.ifftshift(res_shift)))

plt.figure(figsize=(15, 5))
plt.subplot(141), plt.imshow(img_noisy, cmap='gray'), plt.title('Noisy Image')
plt.subplot(142), plt.imshow(spectrum, cmap='gray'), plt.title('Spectrum')
plt.subplot(143), plt.imshow(h_notch, cmap='gray'), plt.title('Notch Filter')
plt.subplot(144), plt.imshow(res_img, cmap='gray'), plt.title('Result')
plt.show()

if __name__ == "__main__":
    exp = FrequencyFilteringExperiment('fig/moon.tif')

    exp.run_task_1_highpass_comparison()
    exp.run_task_1_laplacian()
    exp.run_task_2_lp_highboost()
    exp.run_task_2_homomorphic_sharpen()
    exp.run_task_3_notch_filter()
```