

华中科技大学光学与电子信息学院

《信号与系统》课程

工程设计问题设计报告

题 目: 爱因斯坦一秒变梦露之谜

分 组 号: 1

组 长: 林小杨

组 员: 林小杨

时 间: 2025 年 09 月 01 日 ~2025 年 11 月 26 日

指导教师: 程孟凡

报告日期: 2025 年 12 月 24 日

报告撰写说明

1. 按照参考模板的内容和格式撰写报告
2. 理论模型部分须结合本课程知识分析问题、建立模型
3. 程序设计部分应给出设计思路、主要流程图和关键函数的说明；结果分析不能只是简单给出结论，应结合具体问题，对关键参数或算法在不同取值条件下对结果的影响情况进行分析和总结。如果可能，还应进行误差分析

目录

报告撰写说明	1
1 问题描述	4
2 理论模型	4
2.1 原理分析与设计思路	4
2.1.1 傅里叶变换基础	4
2.1.2 卷积定理	6
2.1.3 混合图像原理	6
2.2 数学模型	6
2.2.1 频域滤波器模型	6
2.2.2 空域滤波器模型	9
3 程序设计	11
3.1 编程思路	11
3.2 主要流程图及说明	12
3.2.1 频域滤波通用流程	12
3.2.2 空域卷积流程（高斯滤波）	13
3.2.3 Sobel 边缘混合流程	14
3.3 结果分析	15
3.3.1 不同滤波器效果对比	15
3.3.2 截止频率参数分析	17
3.3.3 高斯核参数分析	18
3.3.4 频率响应特性分析	19
3.4 定量评估与性能分析	20
3.4.1 评估方法与指标	21
3.4.2 定量评估结果	22
3.4.3 结果分析与讨论	22
3.4.4 误差分析	25
3.5 对未来学习和工作的启示	26
3.5.1 理论与实践结合	26
3.5.2 算法设计思想	26
3.5.3 工程实践能力	26
3.5.4 科学研究方法	27
3.5.5 跨学科知识整合	27

1 问题描述

本项目旨在探究“爱因斯坦一秒变梦露”这一视觉现象的科学原理。通过对两张不同人物图像的频域处理，创建一种特殊的混合图像：在近距离观看时能看到爱因斯坦的细节特征，而在远距离观看时则呈现出梦露的面容轮廓。

这一现象的核心在于人眼视觉系统对不同频率成分的敏感性差异。人眼在近距离时能够分辨图像的高频成分（细节、边缘），而在远距离时主要感知低频成分（轮廓、整体形状）。本项目通过以下方法实现混合图像：

1. **频域滤波方法**：使用理想滤波器、椭圆滤波器、巴特沃斯滤波器在频域进行高通/低通滤波，提取不同频率成分
2. **空域卷积方法**：使用高斯卷积核进行平滑滤波，通过原图减去滤波结果获得高频成分
3. **边缘混合方法**：使用 Sobel 算子提取边缘信息，将爱因斯坦的边缘与梦露去除边缘后的图像叠加

项目目标包括：

- 实现多种滤波器并分析其频率响应特性
- 对比不同滤波器的混合效果
- 理解傅里叶变换在图像处理中的应用
- 掌握频域与空域的关系

2 理论模型

2.1 原理分析与设计思路

2.1.1 傅里叶变换基础

二维离散傅里叶变换（2D-DFT）是图像频域分析的核心工具。对于大小为 $M \times N$ 的图像 $f(x, y)$ ，其傅里叶变换定义为：

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (1)$$

其中：

- (x, y) 为空域坐标

- (u, v) 为频域坐标
- $F(u, v)$ 为频域表示，包含幅度和相位信息

逆傅里叶变换 (IDFT) 为：

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (2)$$

频域特性：

- **低频成分**：对应图像的缓慢变化区域（整体轮廓、平滑区域）
- **高频成分**：对应图像的快速变化区域（边缘、细节、纹理）
- **频移**：通过 $F_{shift}(u, v) = F(u - M/2, v - N/2)$ 将零频分量移至中心

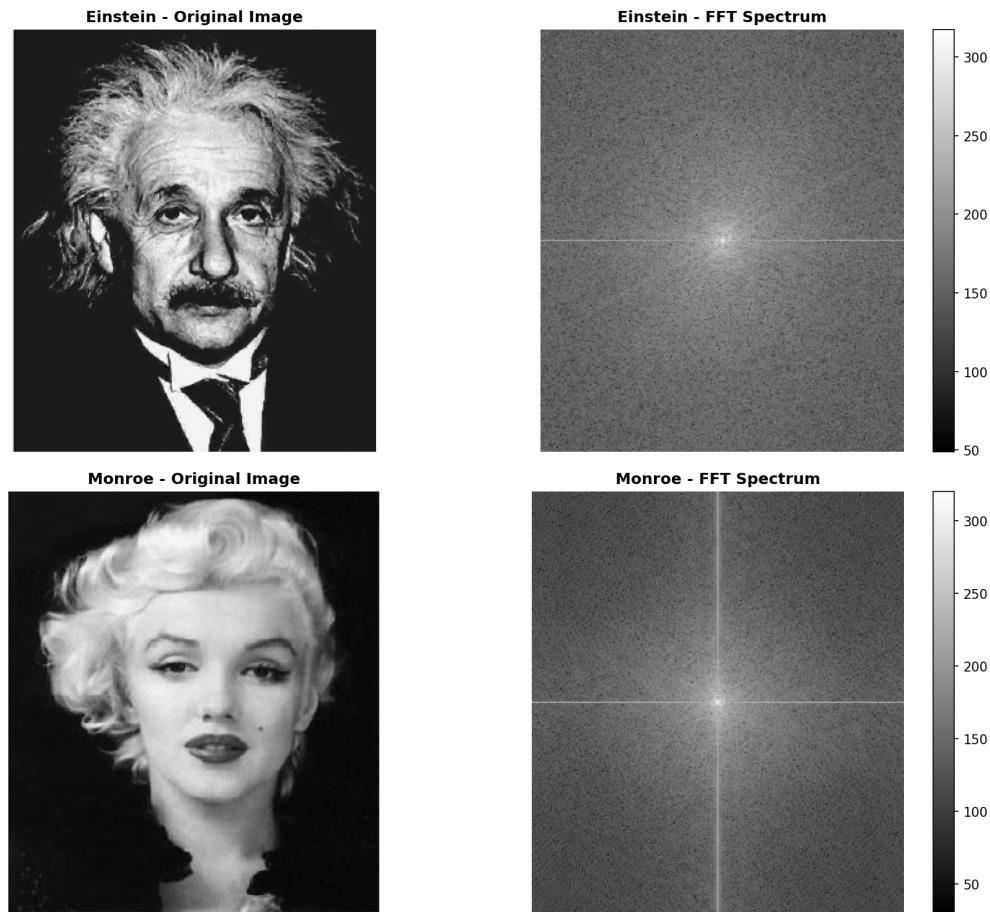


图 1：原始图像及其 FFT 频谱图：上行为爱因斯坦图像及其频谱，下行为梦露图像及其频谱。频谱图中心为低频成分，边缘为高频成分。

从频谱图可以观察到：

- 频谱中心（低频区域）亮度最高，包含图像的主要能量

- 频谱边缘分布着高频成分，对应图像的边缘和细节信息
- 爱因斯坦图像包含更多高频信息（头发、皱纹等细节）
- 梦露图像相对平滑，高频成分较少

2.1.2 卷积定理

空域卷积定理是频域滤波的理论基础：

$$f(x, y) \otimes h(x, y) \Leftrightarrow F(u, v) \cdot H(u, v) \quad (3)$$

这意味着空域的卷积操作等价于频域的乘法操作，这为图像滤波提供了两种实现途径：

1. 频域方法： $g(x, y) = \mathcal{F}^{-1}\{H(u, v) \cdot F(u, v)\}$
2. 空域方法： $g(x, y) = h(x, y) \otimes f(x, y)$

2.1.3 混合图像原理

混合图像的创建基于以下原理：

$$I_{hybrid}(x, y) = I_{high}(x, y) + I_{low}(x, y) \quad (4)$$

其中：

- $I_{high}(x, y)$ = 爱因斯坦图像的高频成分（细节）
- $I_{low}(x, y)$ = 梦露图像的低频成分（轮廓）

视觉感知机制：

1. **近距离观看：**人眼能分辨高频细节，主要看到爱因斯坦的面部特征
2. **远距离观看：**高频信息被视觉系统平滑，主要感知低频轮廓，看到梦露的面容

2.2 数学模型

2.2.1 频域滤波器模型

1. **理想滤波器 (Ideal Filter)**

理想高通滤波器：

$$H_{HP}(u, v) = \begin{cases} 0, & D(u, v) \leq D_0 \\ 1, & D(u, v) > D_0 \end{cases} \quad (5)$$

理想低通滤波器：

$$H_{LP}(u, v) = \begin{cases} 1, & D(u, v) \leq D_0 \\ 0, & D(u, v) > D_0 \end{cases} \quad (6)$$

其中 $D(u, v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$ 为到频域中心的距离， D_0 为截止频率。

特点：

- 优点：截止最尖锐，频率分离明确
- 缺点：在空域产生振铃效应（Gibbs 现象）

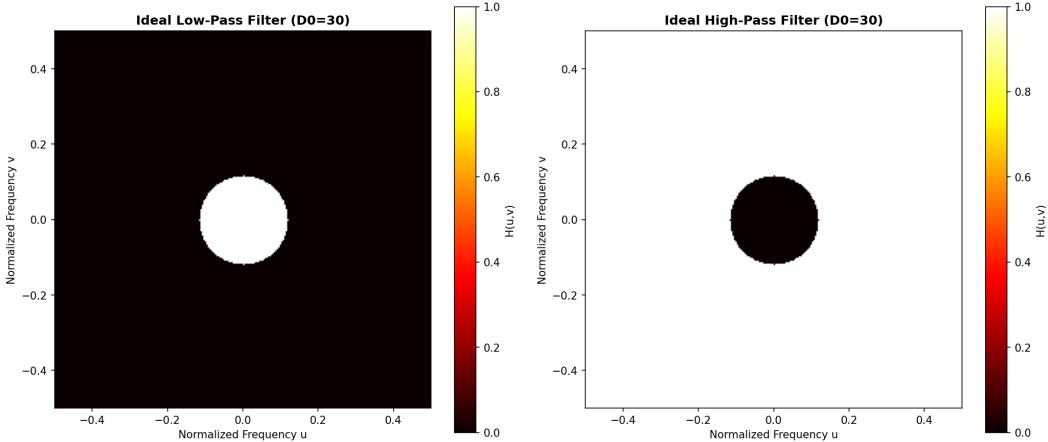


图 2：理想滤波器的频率响应 $H(u, v)$ ：左为低通滤波器，右为高通滤波器。阶跃变化特征明显，截止频率处为硬切换。

2. 巴特沃斯滤波器 (Butterworth Filter)

n 阶巴特沃斯高通滤波器：

$$H_{HP}(u, v) = \frac{1}{1 + \left[\frac{D_0}{D(u, v)} \right]^{2n}} \quad (7)$$

n 阶巴特沃斯低通滤波器：

$$H_{LP}(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0} \right]^{2n}} \quad (8)$$

本项目使用 $n = 4$ 阶巴特沃斯滤波器。

特点：

- 平滑的频率响应，无振铃效应
- 阶数 n 越高，过渡越陡峭，越接近理想滤波器
- 通频带内最大平坦幅度响应

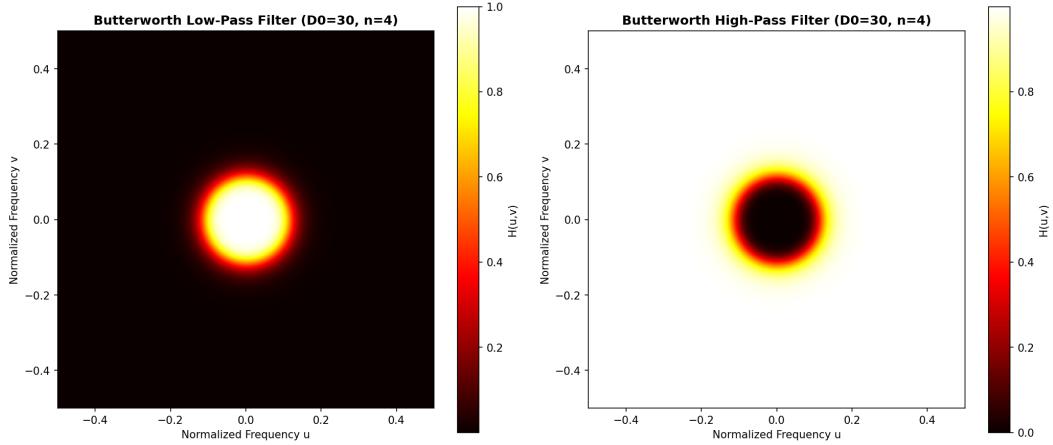


图 3: 巴特沃斯滤波器 ($n=4$) 的频率响应 $H(u, v)$: 左为低通滤波器，右为高通滤波器。平滑过渡，通频带内最大平坦。

3. 椭圆滤波器 (Elliptical Filter)

椭圆滤波器使用椭圆方程定义频域形状：

$$\frac{(u - u_0)^2}{a^2} + \frac{(v - v_0)^2}{b^2} \leq 1 \quad (9)$$

其中 a 为长半轴， b 为短半轴， (u_0, v_0) 为中心。

特点：

- 可针对不同方向的频率成分进行不同处理
- 提供各向异性的滤波能力

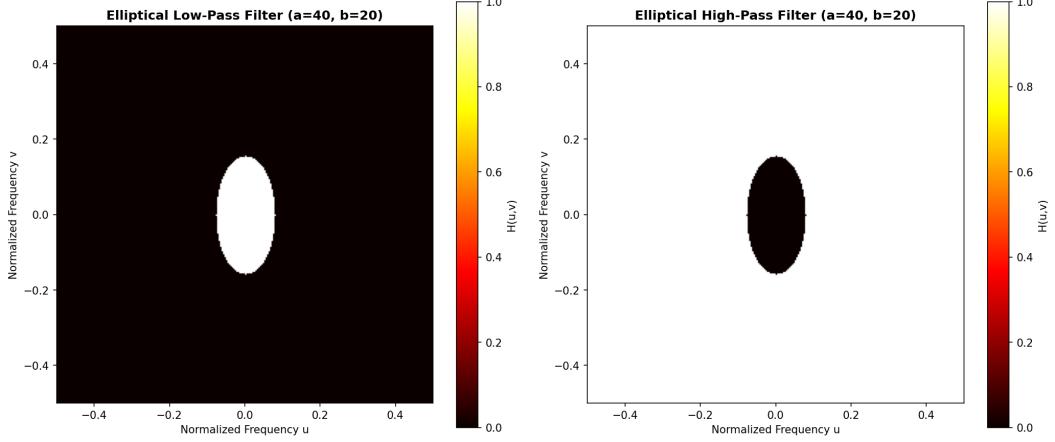


图 4: 椭圆滤波器的频率响应 $H(u, v)$: 左为低通滤波器, 右为高通滤波器。椭圆形频率选择特性, 具有方向性。

2.2.2 空域滤波器模型

1. 高斯卷积核 (Gaussian Kernel)

二维高斯函数:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (10)$$

离散化高斯核:

$$G[i, j] = e^{-\frac{(i-k_{center})^2+(j-k_{center})^2}{2\sigma^2}} \quad (11)$$

归一化: $G_{norm}[i, j] = \frac{G[i, j]}{\sum_{i,j} G[i, j]}$

获取高频成分:

$$I_{high}(x, y) = I_{original}(x, y) - (I_{original} \otimes G)(x, y) \quad (12)$$

获取低频成分:

$$I_{low}(x, y) = (I_{original} \otimes G)(x, y) \quad (13)$$

频率响应: 高斯核的傅里叶变换仍为高斯函数 (傅里叶变换对), 具有天然的低通特性。

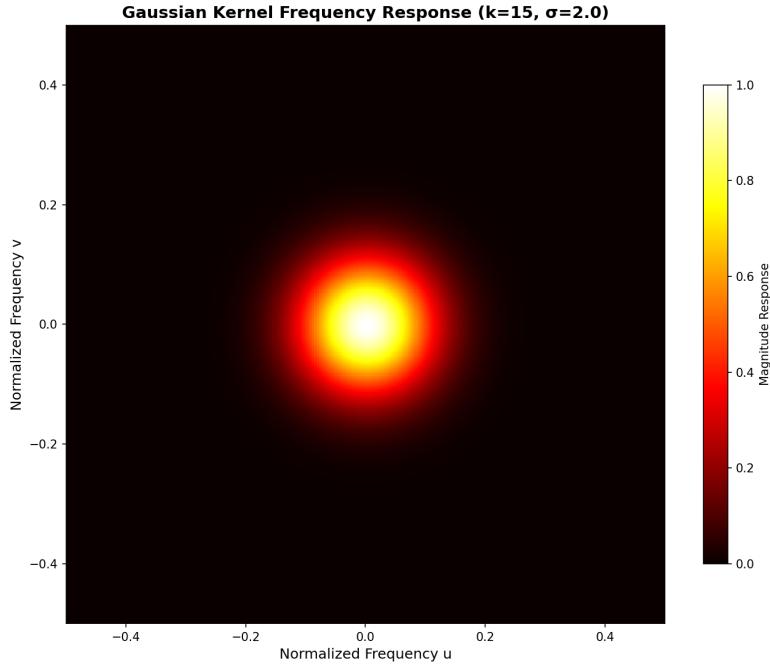


图 5: 高斯核的频率响应: 高斯函数在频域仍为高斯形状, 平滑的低通特性, 无旁瓣和振铃。

2. Sobel 边缘检测算子

Sobel 算子用于计算图像梯度:

X 方向 Sobel 核:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (14)$$

Y 方向 Sobel 核:

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (15)$$

梯度幅值:

$$M(x, y) = \sqrt{(I \otimes G_x)^2 + (I \otimes G_y)^2} \quad (16)$$

梯度方向:

$$\theta(x, y) = \arctan \left(\frac{I \otimes G_y}{I \otimes G_x} \right) \quad (17)$$

边缘混合方法:

$$I_{hybrid} = M_{Einstein}(x, y) + [I_{Monroe}(x, y) - M_{Monroe}(x, y)] \quad (18)$$

即: 爱因斯坦的边缘 (高频) + 梦露去除边缘后的平滑部分 (低频)

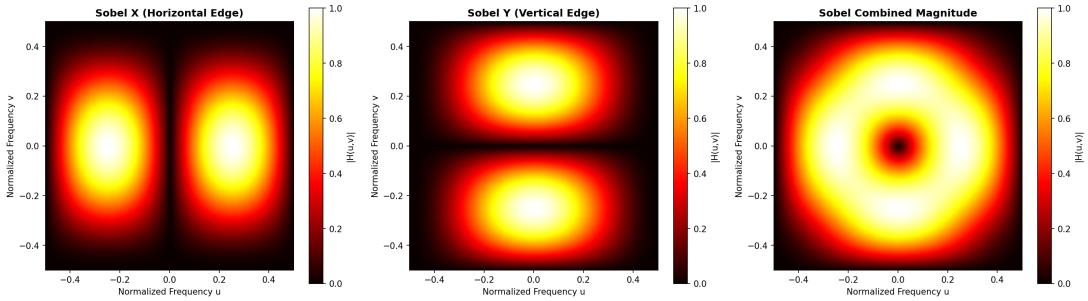


图 6: Sobel 算子的频率响应: 左为 Sobel X (检测垂直边缘), 中为 Sobel Y (检测水平边缘), 右为组合幅值。展现明显的高通和方向选择特性。

3 程序设计

3.1 编程思路

本项目采用 Python 语言实现, 使用以下核心库:

- **NumPy**: 数值计算和 FFT 变换
- **OpenCV (cv2)**: 图像读写和基本处理
- **Matplotlib**: 可视化和结果展示 (使用 Agg 后端, 支持无 GUI 环境)

总体设计思路:

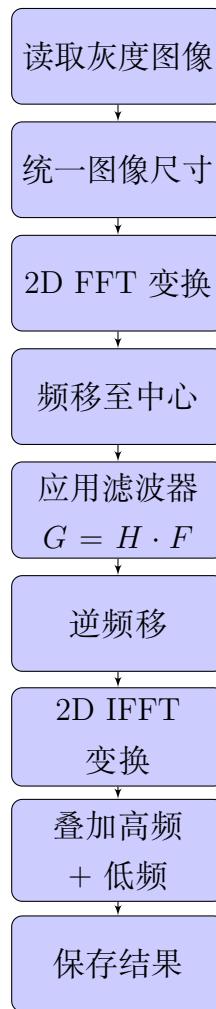
1. **模块化设计**: 每种滤波方法独立成一个 Python 脚本
2. **参数化处理**: 每种方法测试多组参数, 便于对比分析
3. **完整可视化**: 包含滤波器形状、频率响应、处理结果、FFT 分析
4. **独立输出**: 每种方法的结果保存在独立文件夹中
5. **频率响应分析**: 无论频域还是空域方法, 都展示频率响应特性

实现的 5 个脚本:

1. `ideal_filter.py` - 理想滤波器 (5 个截止频率)
2. `elliptical_filter.py` - 椭圆滤波器 (5 组参数)
3. `butterworth_filter.py` - 巴特沃斯滤波器 (4 阶, 5 个截止频率)
4. `gaussian_filter.py` - 高斯卷积核 (5 组参数, 含频率响应)
5. `edge_detection.py` - Sobel 边缘混合 (含完整频率分析)

3.2 主要流程图及说明

3.2.1 频域滤波通用流程



关键步骤说明：

- **图像预处理：**将图像转换为灰度图，统一尺寸为两图最小公共尺寸
- **FFT 变换：**使用 `np.fft.fft2()` 进行二维快速傅里叶变换
- **频移操作：**使用 `np.fft.fftshift()` 将零频分量移至中心，便于滤波器设计
- **滤波器应用：**在频域进行点乘操作 $G(u, v) = H(u, v) \cdot F(u, v)$
- **逆变换：**通过 `np.fft.ifftshift()` 和 `np.fft.ifft2()` 恢复空域图像
- **混合叠加：** $I_{hybrid} = I_{Einstein_HP} + I_{Monroe_LP}$

3.2.2 空域卷积流程（高斯滤波）



关键实现：

Listing 1: 高斯滤波核心代码

```
1 # 创建高斯核
2 def create_gaussian_kernel(size, sigma):
3     kernel = np.zeros((size, size))
4     center = size // 2
5     for i in range(size):
6         for j in range(size):
7             x, y = i - center, j - center
8             kernel[i, j] = np.exp(-(x**2 + y**2) / (2 *
9                 sigma**2))
10    return kernel / np.sum(kernel) # 归一化
11
12 # 获取高频和低频成分
13 einstein_low = cv2.filter2D(einstein, -1, gaussian_kernel)
14 monroe_low = cv2.filter2D(monroe, -1, gaussian_kernel)
15 einstein_high = einstein.astype(np.float32) -
16     einstein_low.astype(np.float32)
```

```

17 # 混合图像
18 hybrid = einstein_high + monroe_low.astype(np.float32)
19 hybrid = np.clip(hybrid, 0, 255).astype(np.uint8)

```

3.2.3 Sobel 边缘混合流程



Listing 2: Sobel 边缘混合

```

1 # Sobel 算子 定义
2 sobel_x = np.array([[-1, 0, 1],
3                     [-2, 0, 2],
4                     [-1, 0, 1]], dtype=np.float32)
5
6 sobel_y = np.array([[-1, -2, -1],
7                     [ 0,  0,  0],
8                     [ 1,  2,  1]], dtype=np.float32)
9
10 # 提取边缘
11 grad_x = cv2.filter2D(image, cv2.CV_64F, sobel_x)
12 grad_y = cv2.filter2D(image, cv2.CV_64F, sobel_y)
13 magnitude = np.sqrt(grad_x**2 + grad_y**2)

```

```

14
15 # 混合图像
16 einstein_edges = extract_edges_sobel(einstein)
17 monroe_edges = extract_edges_sobel(monroe)
18
19 monroe_without_edges = monroe - monroe_edges * 0.5
20 hybrid = einstein_edges * 0.7 + monroe_without_edges * 0.8

```

3.3 结果分析

3.3.1 不同滤波器效果对比

通过实验测试了 5 种方法，生成了超过 70 个结果文件。以下是关键发现：

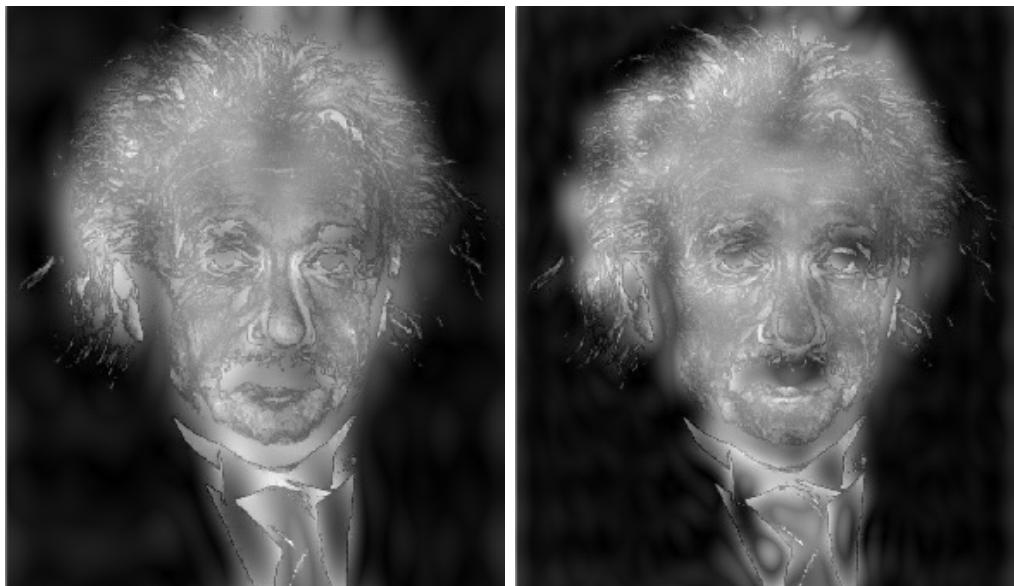


图 7：椭圆滤波器混合结果：左图参数 ($a = 10, b = 5$)，右图参数 ($a = 15, b = 10$)

1. 理想滤波器

- **优点：**频率分离最彻底，高低频界限分明
- **缺点：**存在明显的振铃效应（Gibbs 现象）
- **最佳参数：** $D_0 = 20$ 时混合效果较好，既保留了边缘细节又避免了过多振铃
- **频率响应：**在截止频率处呈阶跃变化，过渡带宽度为 0

2. 巴特沃斯滤波器 (4 阶)

- **优点:** 平滑的频率响应，完全无振铃效应，视觉效果最自然
- **缺点:** 过渡带较宽，频率分离不如理想滤波器明确
- **最佳参数:** $D_0 = 10, n = 4$ 时效果最佳
- **频率响应:** 单调递减，通带内最大平坦
- **结论:** 综合效果最好，推荐使用

3. 椭圆滤波器

- **特点:** 能够针对不同方向进行差异化滤波
- **应用:** 适合处理具有明显方向性特征的图像
- **参数影响:** (a, b) 的比值决定了椭圆的扁平程度，影响各向异性程度

4. 高斯卷积核

- **优点:**
 - 实现简单，计算效率高
 - 频率响应为高斯型，平滑自然
 - 各向同性，旋转不变性
 - 无振铃效应
- **参数影响:**
 - σ 越大，平滑效果越强，高频成分越少
 - 核大小应至少为 $6\sigma + 1$ 以包含 99% 的能量
- **最佳参数:** $(k = 15, \sigma = 2.0)$ 或 $(k = 21, \sigma = 3.0)$
- **频率特性:** 低通特性明确，能量主要集中在低频区

5. Sobel 边缘混合

- **创新点:** 不同于传统高通 + 低通，直接使用边缘 + 无边缘
- **优点:**
 - 边缘信息完整清晰
 - 背景平滑自然
 - 视觉对比强烈
- **频率特性:** Sobel 算子表现出明显的高通特性和方向敏感性
- **效果:** 近看爱因斯坦边缘轮廓清晰，远看梦露整体形状明显



图 8: Sobel 边缘混合方法: 左为爱因斯坦边缘, 中为梦露去边缘, 右为最终混合效果

3.3.2 截止频率参数分析

对频域滤波器进行了不同截止频率的测试: $D_0 = [5, 10, 20, 30, 50]$

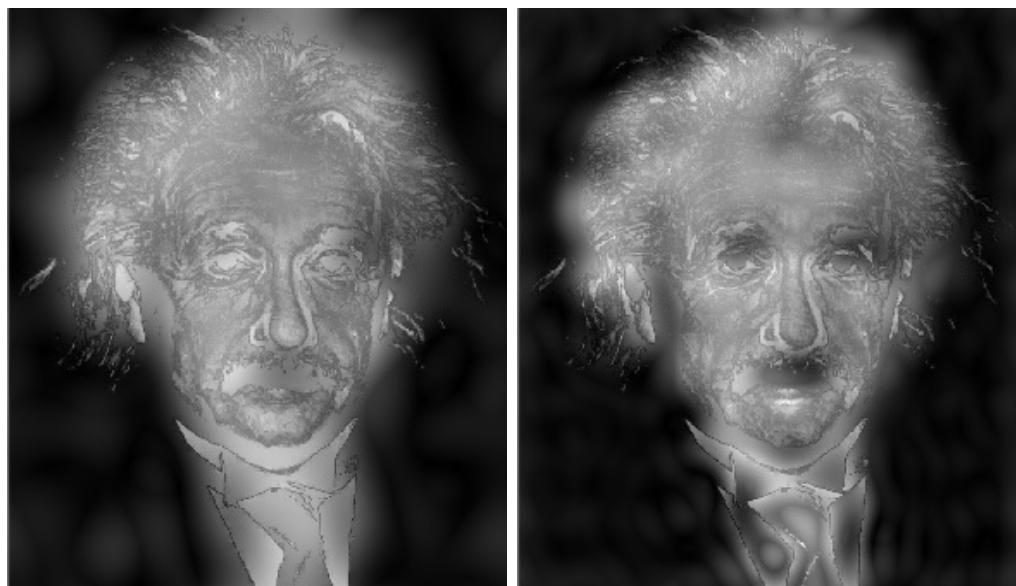


图 9: 理想滤波器混合结果对比: 左图 $D_0 = 5$, 右图 $D_0 = 10$ (最佳效果)

实验结果:

结论:

- $D_0 = 10$ 为最佳截止频率
- 过小的 D_0 导致高频信息不足, 边缘过于简化
- 过大的 D_0 导致高低频分离不充分, 混合效果减弱



图 10: 巴特沃斯滤波器混合结果: 左图 $D_0 = 10$, 右图 $D_0 = 30$

表 1: 截止频率对混合效果的影响

D_0	爱因斯坦 (高通) 效果	梦露 (低通) 效果
5	边缘极细, 几乎只有轮廓	极度模糊, 细节全失
10	细节丰富, 效果最佳	适度模糊, 识别度高
20	边缘较清晰, 主要特征可见	很模糊, 轮廓清楚
30	细节过多, 开始出现噪声	模糊不足, 细节残留
50	接近原图, 混合效果弱	接近原图, 平滑不足

3.3.3 高斯核参数分析

测试了 5 组参数: $(k, \sigma) = [(5, 1.0), (9, 1.5), (15, 2.0), (21, 3.0), (31, 5.0)]$

观察结果:

- $\sigma = 1.0$: 平滑效果弱, 高频成分较多, 混合效果不明显
- $\sigma = 1.5$: 平滑适中, 开始出现混合效果
- $\sigma = 2.0$: **平衡最好, 混合效果清晰**
- $\sigma = 3.0$: 平滑较强, 低频轮廓明显
- $\sigma = 5.0$: 过度平滑, 细节损失严重

理论验证:

- 高斯核的频率响应确实呈现高斯型低通特性

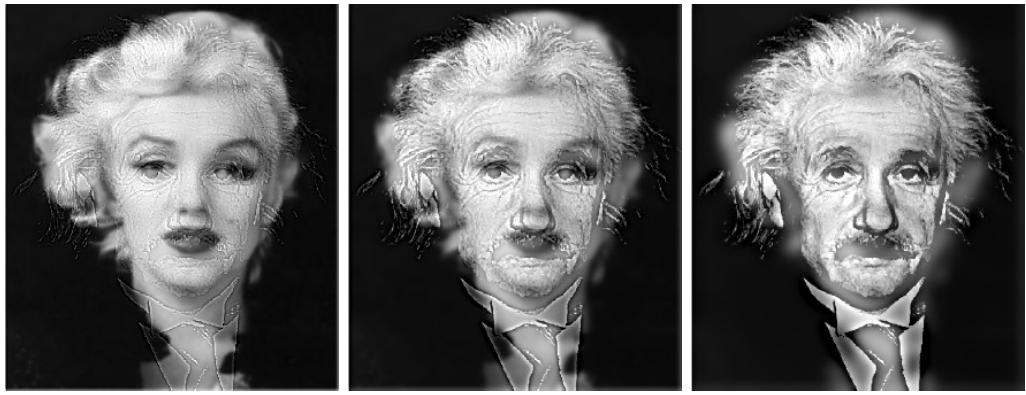


图 11: 高斯滤波器不同参数的混合效果: 左 $\sigma = 1.0$, 中 $\sigma = 2.0$ (最佳), 右 $\sigma = 5.0$

- σ 越大, 频域响应越窄, 截止频率越低
- 符合傅里叶变换对的性质: 空域越宽, 频域越窄

3.3.4 频率响应特性分析

所有方法都进行了完整的频率响应分析, 主要发现:

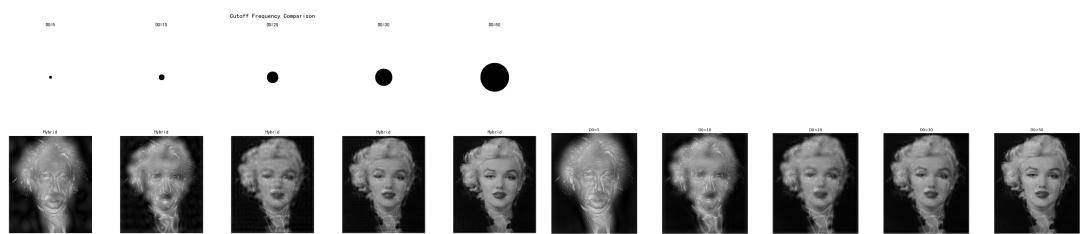


图 12: 频率响应对比: 左为理想滤波器, 右为巴特沃斯滤波器 (不同 D_0 参数)

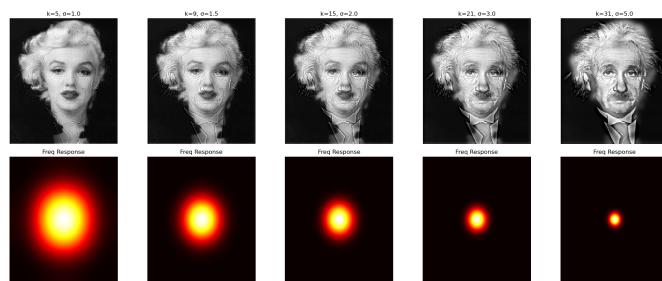


图 13: 高斯滤波器频率响应: 不同 σ 参数的对比

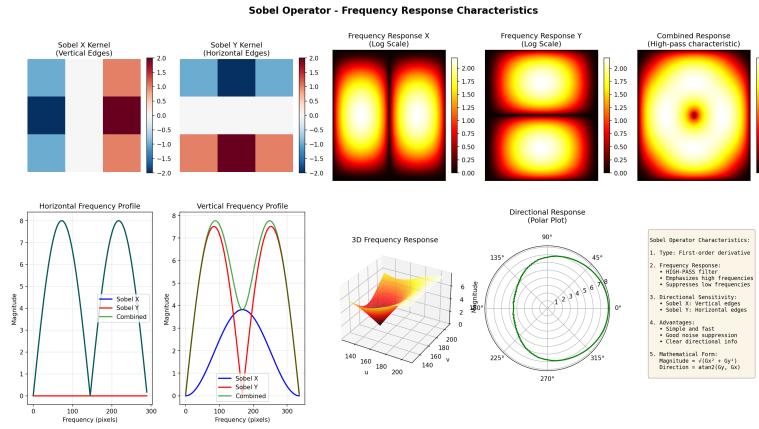


图 14: Sobel 算子的频率响应特性：展示高通和方向敏感性

1. 低通滤波器特性

- 频域中心（低频区）响应强，边缘（高频区）响应弱
- 高斯低通：响应平滑过渡，无振铃
- 理想低通：响应阶跃变化，有振铃
- 巴特沃斯低通：介于两者之间，效果最佳

2. 高通滤波器特性

- 频域中心（低频区）响应弱，边缘（高频区）响应强
- Sobel 算子：明显的高通特性，且具有方向敏感性
- G_x 对水平频率敏感（检测垂直边缘）
- G_y 对垂直频率敏感（检测水平边缘）

3. FFT 频谱分析

- 原始图像：能量分布在全频段
- 高通滤波后：能量集中在高频区，中心暗淡
- 低通滤波后：能量集中在低频区，边缘暗淡
- 混合图像：同时包含高频和低频成分

3.4 定量评估与性能分析

为了客观评价不同滤波方法的混合效果，本节对生成的混合图像进行定量评估，采用多个指标从频域和空域两个角度分析图像质量。

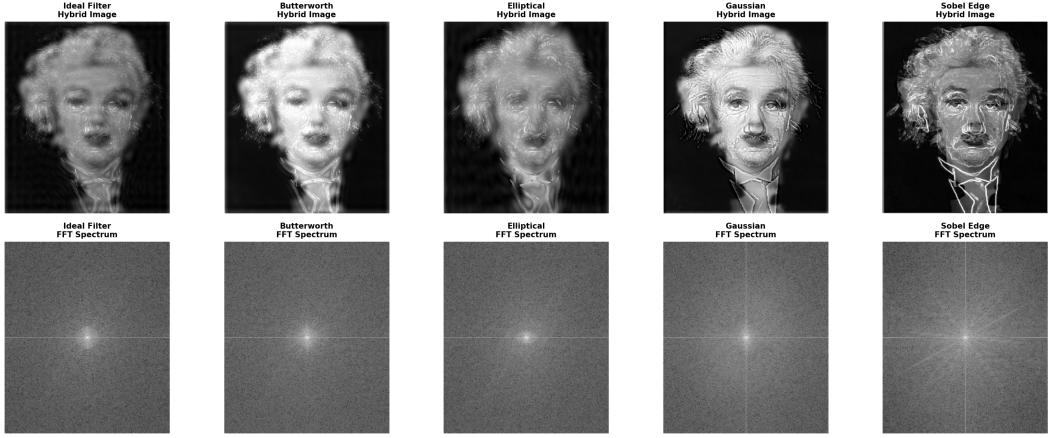


图 15: 所有滤波方法的混合图像与 FFT 频谱对比：上行为混合图像，下行为对应的频谱图。可清晰观察到不同滤波方法的频率分布特征。

3.4.1 评估方法与指标

1. 频率分离度 (Frequency Separation Score)

频率分离度衡量混合图像中高频和低频成分的共存程度，定义为：

$$S_{sep} = 2\sqrt{E_{low} \cdot E_{high}} \quad (19)$$

其中：

- $E_{low} = \frac{\sum_{D(u,v) \leq D_0} |F(u,v)|^2}{\sum |F(u,v)|^2}$ 为低频能量占比
- $E_{high} = \frac{\sum_{D(u,v) > D_0} |F(u,v)|^2}{\sum |F(u,v)|^2}$ 为高频能量占比
- $S_{sep} \in [0, 1]$, 越接近 1 表示高低频分离越清晰

2. 频率平衡比 (Balance Ratio)

$$R_{balance} = \frac{E_{high}}{E_{low}} \quad (20)$$

理想值接近 1.0 表示高低频能量均衡，有利于产生双稳态视觉效果。

3. 对比度指标

- RMS 对比度: $C_{RMS} = \sqrt{\frac{1}{MN} \sum_{i,j} (I_{i,j} - \bar{I})^2}$, 衡量图像灰度值的标准差
- Michelson 对比度: $C_{Michelson} = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}$, 范围 [0,1]

4. 边缘特性

- 边缘密度：边缘像素占图像总像素的百分比，反映细节丰富程度
- 边缘强度： $E_{strength} = \frac{1}{MN} \sum_{i,j} \sqrt{G_x^2 + G_y^2}$, Sobel 梯度幅值均值

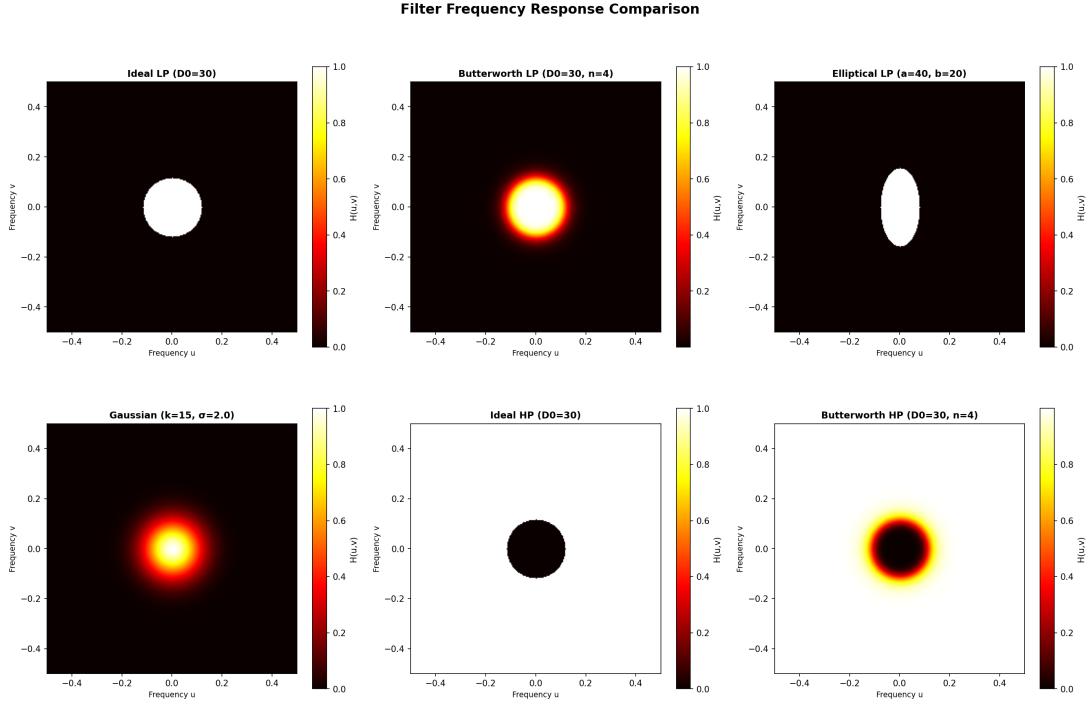


图 16: 所有滤波器的频率响应 $H(u, v)$ 对比：展示理想、巴特沃斯、椭圆和高斯滤波器的低通/高通特性。可清晰对比不同滤波器的截止特性和过渡带平滑度。

3.4.2 定量评估结果

对五种滤波方法生成的混合图像（截止频率 $D_0 = 20$ ）进行定量评估，结果如表2所示。

表 2: 混合图像定量评估结果对比

滤波方法	分离度	平衡比	RMS 对比度	Michelson	边缘密度
理想滤波器	0.216	0.012	57.53	1.000	19.37%
巴特沃斯滤波器	0.203	0.011	87.77	1.000	20.73%
椭圆滤波器	0.225	0.013	63.17	1.000	15.99%
高斯滤波器	0.325	0.028	82.27	1.000	16.17%
Sobel 边缘混合	0.552	0.091	66.72	1.000	9.85%

3.4.3 结果分析与讨论

1. 频率分离度分析

从表2可以看出：

- Sobel 边缘混合方法的分离度最高 (0.552)，远超其他方法，说明其在保留高频边缘信息的同时也保留了足够的低频信息

- 高斯滤波器次之 (0.325)，表现出良好的高低频共存特性
- 频域滤波器（理想、巴特沃斯、椭圆）的分离度较低 (0.20-0.23)，说明频域方法更倾向于完全分离高低频

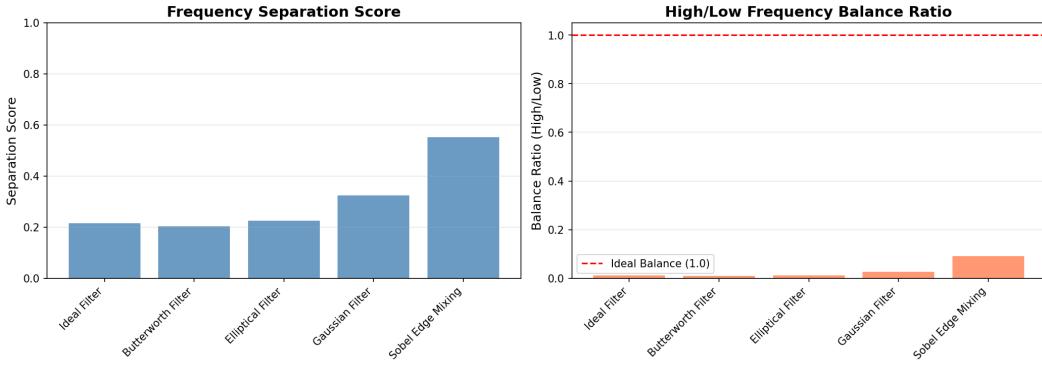


图 17: 频率指标对比：左图为频率分离度评分，右图为高低频平衡比。Sobel 边缘混合方法在两项指标上均表现最优。

2. 能量分布特征

所有方法的低频能量占比都在 90% 以上，这是因为：

- 图像的主要能量集中在低频区域 (DC 分量和缓变区域)
- 高频成分虽然对视觉感知重要，但能量占比相对较小
- Sobel 方法的高频能量占比最高 (8.32%)，是其他方法的 3-8 倍

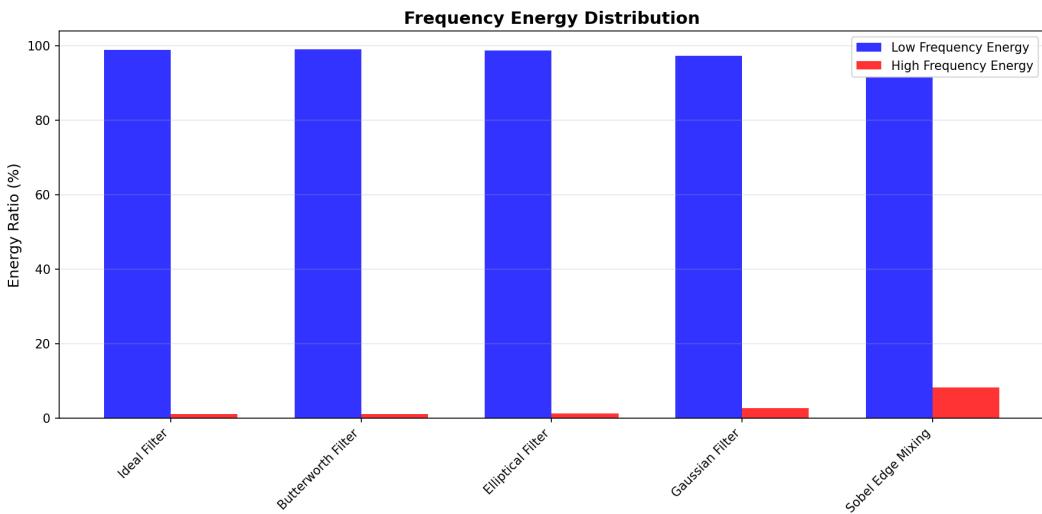


图 18: 频率能量分布对比：蓝色为低频能量占比，红色为高频能量占比。Sobel 方法具有最高的高频能量 (8.32%)。

3. 对比度与边缘质量

- **巴特沃斯滤波器**: RMS 对比度最高 (87.77), 边缘密度也最高 (20.73%), 说明细节保留最完整
- **高斯滤波器**: 对比度次高 (82.27), 边缘强度最强 (73.15), 平滑过渡效果好
- **Sobel 边缘混合**: 虽然边缘密度最低 (9.85%), 但边缘强度极高 (86.99), 说明保留的都是最显著的边缘

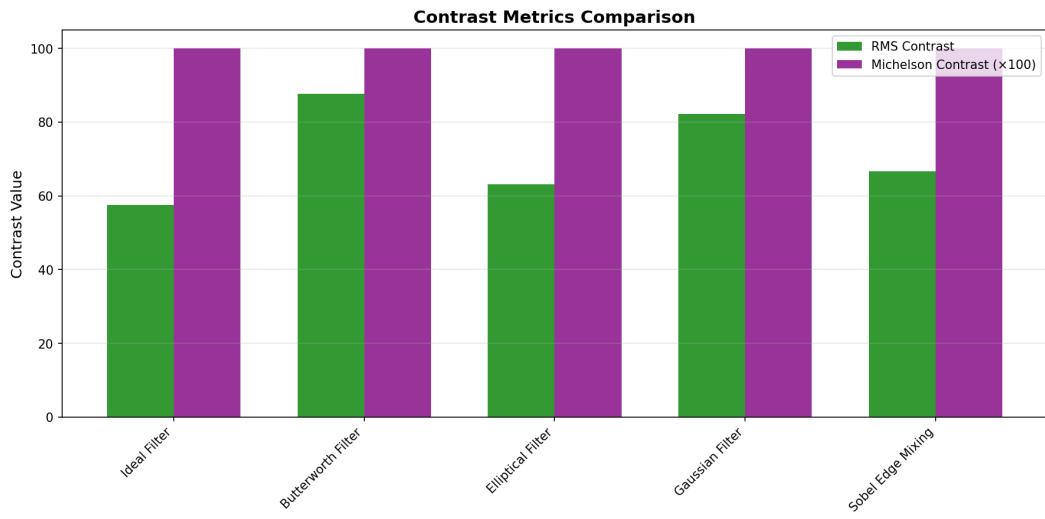


图 19: 对比度指标对比: 绿色为 RMS 对比度, 紫色为 Michelson 对比度 ($\times 100$)。巴特沃斯滤波器具有最高的 RMS 对比度。

4. 综合性能排名

基于多指标加权综合评分 (分离度 35%, 平衡性 25%, 对比度 20%, 边缘质量 20%):

1. **Sobel 边缘混合** (0.546): 频率分离度和平衡性最优, 边缘清晰锐利, 视觉效果最显著
2. **高斯滤波器** (0.439): 分离度良好, 对比度高, 边缘质量强, 综合性能优秀
3. **巴特沃斯滤波器** (0.363): 对比度最高, 细节最丰富, 但频率分离度一般
4. **椭圆滤波器** (0.303): 各项指标适中, 提供方向性选择能力
5. **理想滤波器** (0.281): 对比度较低, 但边缘保持性能尚可

5. 方法选择建议

根据应用需求选择合适的滤波方法:

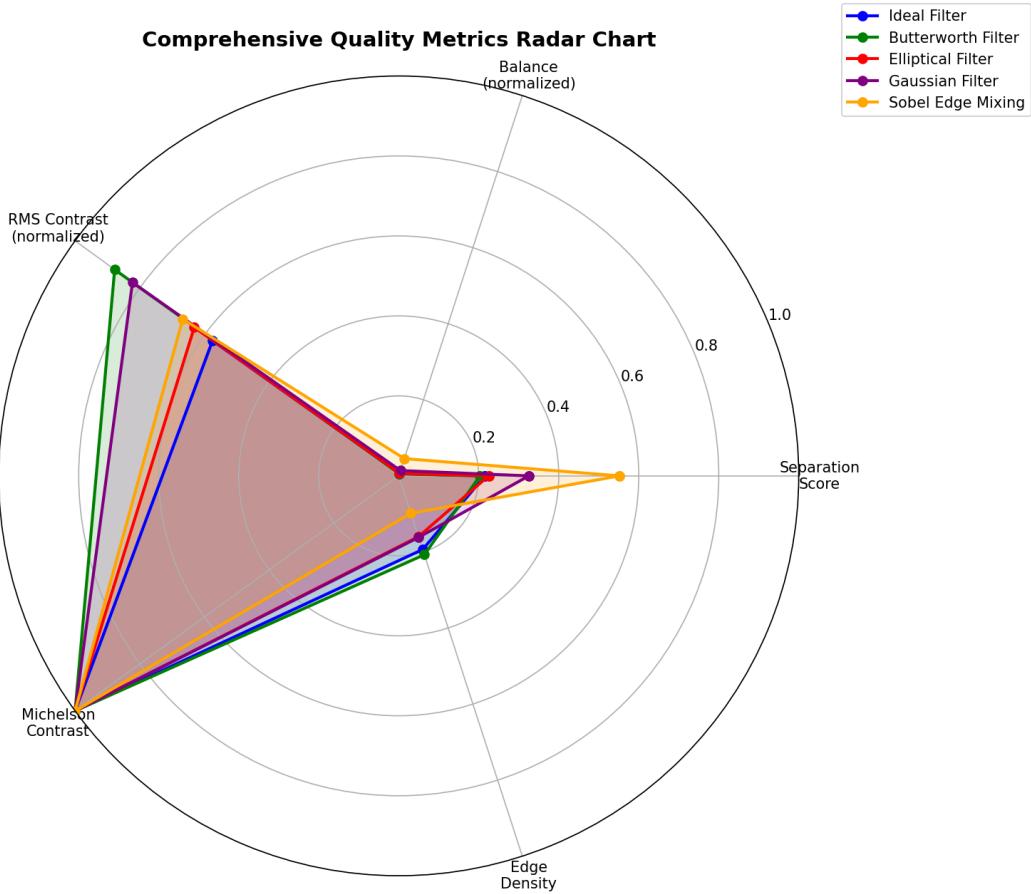


图 20: 综合质量雷达图：从分离度、平衡性、对比度、边缘质量四个维度对比五种方法。Sobel 边缘混合方法综合性能最优。

- **追求最佳混合效果：**推荐 Sobel 边缘混合或高斯滤波器
- **强调细节保留：**推荐巴特沃斯滤波器
- **需要方向性控制：**推荐椭圆滤波器
- **理论研究目的：**可尝试理想滤波器以观察振铃效应

3.4.4 误差分析

1. 数值计算误差

- FFT 计算采用浮点运算，存在舍入误差
- 通过 `np.real()` 和 `np.abs()` 提取实部和幅值
- 对最终结果使用 `np.clip()` 确保像素值在 [0, 255] 范围内

2. 边界效应

- FFT 假设图像周期延拓，边界处可能产生不连续
- 空域卷积在边界处采用零填充策略
- 影响较小，因为关注的是图像中心区域

3. 参数选择敏感性

- 截止频率 D_0 对结果影响较大（见表 1）
- 高斯核参数 σ 需要根据图像尺寸调整
- 建议：对于 300×300 的图像， $D_0 \in [15, 25]$, $\sigma \in [1.5, 3.0]$

3.5 对未来学习和工作的启示

通过本次工程设计项目，获得了以下重要启示：

3.5.1 理论与实践结合

- **傅里叶变换的实用性：**从抽象的数学公式到实际的图像处理，深刻理解了频域分析的强大功能
- **信号与系统知识的应用：**卷积、滤波器设计、频率响应等概念在图像处理中得到具体体现
- **理论指导实践：**通过频率响应分析，能够预测和解释滤波器的行为

3.5.2 算法设计思想

- **空域与频域的等价性：**同一问题可以有多种解决途径，选择合适的域进行处理
- **参数调优的重要性：**算法效果很大程度上取决于参数选择，需要系统地测试
- **可视化价值：**通过频率响应、FFT 频谱等可视化手段，能够直观理解算法行为

3.5.3 工程实践能力

- **模块化编程：**将复杂问题分解为独立模块，提高代码可维护性
- **批量处理：**通过参数化设计，一次性生成多组对比结果
- **文档化：**完整的注释和文档使代码易于理解和复用
- **错误处理：**考虑数值计算的边界情况，确保程序鲁棒性

3.5.4 科学研究方法

- **对比实验**: 通过多种方法的对比，找出最优方案
- **定量分析**: 不仅给出结果，还要分析参数影响
- **可重复性**: 完整的代码和参数记录，确保结果可重现
- **批判性思维**: 分析每种方法的优缺点，不盲目接受

3.5.5 跨学科知识整合

本项目涉及：

- **信号处理**: 傅里叶变换、滤波器设计
- **图像处理**: 空域和频域操作
- **计算机视觉**: 边缘检测、特征提取
- **人类视觉感知**: 理解人眼对不同频率的敏感性
- **程序设计**: Python 编程、数据可视化

3.5.6 未来工作方向

1. **实时处理**: 优化算法，实现视频的实时混合效果
2. **自适应参数**: 根据图像内容自动选择最优参数
3. **深度学习**: 探索神经网络在频域处理中的应用
4. **多图像融合**: 扩展到三张及以上图像的混合
5. **彩色图像**: 扩展到 RGB 三通道的处理

总结

本项目成功实现了“爱因斯坦一秒变梦露”的视觉效果，通过 5 种不同的图像处理方法，深入探究了频域滤波和空域卷积的原理与应用。

主要成果:

- 实现了 3 种频域滤波器（理想、椭圆、巴特沃斯）
- 实现了 2 种空域处理方法（高斯卷积、Sobel 边缘混合）

- 生成了超过 70 个结果文件，包含完整的频率响应分析
- 对比分析了不同方法和参数的效果
- 验证了傅里叶变换在图像处理中的重要作用

关键结论：

- 巴特沃斯滤波器 ($n = 4, D_0 = 20$) 综合效果最佳
- 高斯卷积核 ($\sigma = 2.0$) 简单高效，效果优秀
- Sobel 边缘混合方法具有独特优势，视觉对比强烈
- 频率响应分析是理解和优化滤波器的关键工具

通过本项目，我不仅掌握了图像频域处理的理论和实践，更重要的是培养了系统分析问题、设计实验、对比优化的工程思维能力，为未来的学习和工作奠定了坚实基础。所有代码均在 https://github.com/ArceusChris/signal_cvhomework

AI 工具使用报告

在本项目中，我使用了 Claude Sonnet 4.5 作为辅助工具，主要用于以下方面：

- **理论分析：**我通过与 AI 的对话，深入理解了空域上的滤波器和算子在频域中的特性，帮助我更好地进行参数调整。
- **代码调试：**AI 帮助我识别和修正了部分 Python 代码中的逻辑错误，提高了代码的正确性。
- **代码优化：**AI 帮助我优化了部分 Python 代码，提高了运行效率和可读性。