

# 同态滤波实验报告

## 一、实验目的

- 掌握图像乘性噪声的数学模型及分离原理
- 理解对数变换在噪声处理中的作用，实现乘性噪声到广义加性噪声的转换
- 学习频域分析（FFT/IFFT）在图像处理中的应用
- 掌握频域增益曲线的设计与调整方法，抑制噪声并恢复图像

## 二、实验原理

### 1. 乘性噪声模型

图像中的乘性噪声可以表示为：

$$g(x, y) = f(x, y) \cdot \eta(x, y) + \varepsilon(x, y)$$

其中：

- $f(x, y)$  为原始图像
- $\eta(x, y)$  为乘性噪声
- $\varepsilon(x, y)$  为加性噪声

本实验重点处理乘性噪声。

### 2. 同态滤波基本原理

同态滤波是一种非线性图像处理技术，特别适用于处理具有不均匀照明或乘性噪声的图像。其基本步骤如下：

- 对数变换**：将乘性噪声转换为加性噪声  
 $\ln(g(x, y)) \approx \ln(f(x, y)) + \ln(\eta(x, y))$
- 傅里叶变换**：将信号转换到频域  
 $F\{\ln(g(x, y))\} = F\{\ln(f(x, y))\} + F\{\ln(\eta(x, y))\}$
- 频域滤波**：应用增益曲线抑制噪声频率  
 $H(u, v) \cdot F\{\ln(g(x, y))\}$

其中高斯同态滤波器的传递函数为：

$$H(u, v) = (\gamma_h - \gamma_l) \cdot (1 - e^{-D^2(u, v)/2D_0^2}) + \gamma_l$$

- $D(u, v)$  是频域中点  $(u, v)$  到原点的距离
- $D_0$  是截止频率
- $\gamma_l$  是低频增益
- $\gamma_h$  是高频增益

- 逆傅里叶变换**：将信号转回空域

- 指数变换**：恢复原始的动态范围

$$\hat{f}(x, y) = \exp(F^{-1}\{H(u, v) \cdot F\{\ln(g(x, y))\}\})$$

## 三、实验步骤

### 1. 对数变换

首先对输入图像进行对数变换，将乘性噪声转换为加性噪声：

```
def log_transform(self, epsilon=1e-6):  
    """对数变换，将乘性噪声转换为加性噪声"""  
    if self.img is None:  
        raise ValueError("未加载图像")  
  
    return np.log(self.img + epsilon)
```

注意：对数变换时添加了一个小常数  $\epsilon$  ( $1e-6$ )，以避免对数运算中出现零值导致的无穷大问题。

### 2. 傅里叶变换与频域分析

对对数变换后的图像执行快速傅里叶变换(FFT)：

```
def fft_transform(self, log_img):  
    """对对数变换后的图像执行快速傅里叶变换(FFT)"""  
    f_transform = np.fft.fft2(log_img)  
    f_shift = np.fft.fftshift(f_transform)  
    self.spectrum = np.abs(f_shift)  
  
    return f_shift, self.spectrum
```

### 3. 设计增益曲线

创建高斯同态滤波器：

```
def create_filter(self, shape, cutoff, gamma_l=0.6, gamma_h=2.0):  
    """创建高斯同态滤波器"""  
    rows, cols = shape  
    crow, ccol = rows // 2, cols // 2  
  
    u = np.arange(0, rows)  
    v = np.arange(0, cols)  
    u, v = np.meshgrid(u - crow, v - ccol, indexing='ij')  
  
    d_squared = u**2 + v**2  
  
    mask = (gamma_h - gamma_l) * (1 - np.exp(-d_squared / (2 * (cutoff**2)))) + gamma_l  
  
    self.filter_mask = mask  
    return mask
```

## 4. 频域滤波与逆变换

应用滤波器到频域图像并进行逆傅里叶变换：

```
def apply_filter(self, f_shift, cutoff=30.0, gamma_l=0.5, gamma_h=1.5):
    """应用滤波器到频域图像"""
    mask = self.create_filter(f_shift.shape, cutoff, gamma_l, gamma_h)
    self.filtered_spectrum = f_shift * mask
    return self.filtered_spectrum

def inverse_fft(self, filtered_spectrum):
    """进行逆傅里叶变换恢复图像"""
    f_ishift = np.fft.ifftshift(filtered_spectrum)
    img_back = np.fft.ifft2(f_ishift)
    return np.real(img_back)
```

## 5. 指数变换与图像恢复

执行指数变换，复原对数变换：

```
def exp_transform(self, ifft_img):
    """执行指数变换，复原对数变换"""
    img_exp = np.exp(ifft_img)

    # 改进归一化方法，避免过曝
    img_normalized = (img_exp - np.min(img_exp)) / (np.max(img_exp) -
    np.min(img_exp))

    # 适度调整亮度，使用合适的gamma校正
    img_adjusted = np.power(img_normalized, 0.1) # 调整gamma值，平衡亮度

    # 增强对比度
    img_contrast = np.clip((img_adjusted - 0.2) * 1.5, 0, 1)

    self.filtered_img = img_contrast
    return img_contrast
```

## 四、实验结果与分析

### 实验参数设置

在本实验中，我们使用了以下参数：

- 截止频率(cutoff): 40.0
- 低频增益(gamma\_l): 0.5
- 高频增益(gamma\_h): 2.0

## 效果图



## 结果分析

1. **对数变换效果**：对数变换成功地将图像中的乘性噪声转换为加性噪声，使后续的频域处理成为可能。
2. **频谱分析**：通过观察频谱图，可以看到噪声和图像信号在频域中的分布情况。乘性噪声通常表现为频谱中的特定模式。
3. **滤波效果**：高斯同态滤波器通过抑制低频成分（ $\gamma_l=0.5$ ）和增强高频成分（ $\gamma_h=2.0$ ），有效地改善了图像的对比度和细节表现。
4. **图像恢复**：经过处理后的图像显示出明显改善的视觉效果，包括：
  - 更均匀的亮度分布
  - 增强的边缘和细节
  - 改善的对比度

## 参数影响分析

1. **截止频率(cutoff)**：
  - 较大的截止频率会保留更多的高频成分，增强图像细节但可能放大噪声
  - 较小的截止频率会抑制更多的高频成分，减少噪声但可能模糊图像细节
2. **低频增益( $\gamma_l$ )**：
  - 较小的值会更强烈地抑制低频成分，减少照明不均问题
  - 较大的值会保留更多的低频成分，保持图像的整体亮度特性
3. **高频增益( $\gamma_h$ )**：
  - 较大的值会更强烈地增强高频成分，提高边缘和细节的清晰度
  - 较小的值会减弱高频增强效果，产生更平滑的结果

## 五、思考题解答

### 1. 理论分析

**为什么对数变换能将乘性噪声转换为加性噪声？是否所有乘性噪声都适用此方法？**

对数变换能将乘性噪声转换为加性噪声是因为对数函数的性质： $\log(a \times b) = \log(a) + \log(b)$ 。当图像模型为  $g(x, y) = f(x, y) \cdot \eta(x, y)$  时，对两边取对数得到：

$\log(g(x, y)) = \log(f(x, y)) + \log(\eta(x, y))$ ，这样乘性噪声就转换为了加性噪声。

并非所有乘性噪声都适用此方法。当噪声或信号包含零值或负值时，对数变换会出现问题。此外，当噪声不满足乘性模型（如混合噪声）时，单纯的对数变换效果会受到限制。

**频域处理相比空域处理有何优势？列举至少两种频域滤波器的设计方法。**

频域处理相比空域处理的优势：

1. 能够分离不同频率的信号成分，更有针对性地处理特定频率的噪声
2. 某些复杂的滤波操作在频域中计算更为高效
3. 可以直观地观察信号的频率分布特性，便于分析噪声特征
4. 对于全局性质的处理（如去除周期性噪声）更为有效

频域滤波器设计方法：

1. 理想滤波器：在特定截止频率处呈现突变的二值滤波器
2. 巴特沃斯滤波器：具有平滑过渡带的滤波器，避免振铃效应
3. 高斯滤波器：基于高斯函数的滤波器，具有良好的时频特性
4. 同态滤波器：针对不同频率范围应用不同增益的非线性滤波器

## 2. 实验分析

**若增益曲线设计不合理（如过度抑制高频），恢复图像会出现何种失真？**

如果增益曲线设计不合理：

1. 过度抑制高频：图像会变得模糊，细节和边缘信息丢失，出现平滑过度的效果
2. 过度增强高频：会放大噪声，产生过度锐化的伪影，图像可能出现"颗粒感"
3. 过度抑制低频：图像整体对比度降低，可能出现"空洞"或"漂白"效果
4. 过度增强低频：图像亮度分布不均，可能导致局部过曝或欠曝

**实验中对数变换后为何需要添加微小量（如  $1e-6$ ）？若省略此操作会引发什么问题？**

在对数变换中添加微小量（如  $1e-6$ ）是为了避免对零值或接近零值的像素点取对数时出现的数学错误（因为  $\log(0)$  是无穷小）。如果省略此操作：

1. 对于值为零的像素，会产生负无穷大的结果，导致计算错误
2. 对于接近零的像素，可能产生极大的负值，扭曲图像的频谱特性
3. 在后续处理中可能导致数值不稳定，影响最终结果的质量

## 3. 拓展思考

**若图像中同时存在加性噪声和乘性噪声，如何改进实验方法？**

当图像同时存在加性噪声和乘性噪声时，可以采用以下改进方法：

1. 两阶段处理：先使用同态滤波处理乘性噪声，再使用维纳滤波等方法处理剩余的加性噪声
2. 混合模型：建立同时考虑加性和乘性噪声的数学模型，如
$$g(x, y) = f(x, y) \cdot \eta_m(x, y) + \eta_a(x, y)$$
3. 小波域处理：在VW小波域中分别处理不同尺度下的噪声，小波变换对混合噪声有较好的分离能力
4. 自适应滤波：根据局部图像特性自适应调整滤波参数，针对不同区域的噪声特性采用不同的处理策略

**频域增益VW曲线是否可以通过自适应方法（如维纳滤波）优化？试说明原理。**

是的，频域增益曲线可以通过自适应方法优化。维纳滤波就是一种典型的自适应频域滤波方法，其原理是：

1. 维纳滤波基于信号和噪声的功率谱密度比例自适应地调整滤波器增益
2. 其传递函数为： $H(u, v) = \frac{S_f(u, v)}{S_f(u, v) + S_n(u, v)}$ ，其中  $S_f$  是信号的功率谱密度， $S_n$  是噪声的功率谱密度
3. 在信噪比高的频率区域，滤波器趋向于保留信号；在信噪比低的区域，滤波器趋向于抑制信号

将这一思想应用到同态滤波中，可以：

1. 估计图像中不同频率成分的信噪比
2. 根据信噪比动态调整  $\gamma_l$  和  $\gamma_h$  参数
3. 设计自适应的截止频率  $D_0$ ，使其根据图像的频谱特性自动调整

这种自适应方法可以针对不同图像的特性提供更优的处理效果，避免人工参数选择的主观性和局限性。

## 六、结论

本实验成功实现了同态滤波器，并应用于处理含有乘性噪声的图像。通过对数变换、频域滤波和指数变换的组合，有效地改善了图像的视觉质量，增强了图像细节和对比度。

实验结果表明，同态滤波是处理不均匀照明和乘性噪声的有效方法。通过调整截止频率、低频增益和高频增益等参数，可以针对不同类型的图像获得最佳处理效果。

此外，实验过程中的可视化分析帮助我们更好地理解图像在不同处理阶段的变化，以及频域处理在图像增强中的作用机制。

未来可以进一步探索自适应参数调整方法，以及将同态滤波与其他图像处理技术结合，以应对更复杂的图像退化问题。