

# 目 录 基于频域的图像恢复算法设计与实现

---

## 摘 要：

随着现代科学技术的发展，各领域对于图像质量的要求越来越高，然而在实际过程中图像容易受到噪声干扰。基于频率域去噪方法可以将图像从空间域转化为频率域，能直观分析信息成分是图像去噪的重要手段。

本文围绕图像的含噪声频谱的图像恢复问题，通过构建滤波器实现图像去噪，并分析讨论了过采样率以及先验信息对图像恢复概率的影响。针对问题 1，设计了低通滤波算法，该算法模型实现对含噪图像的有效恢复，将测试图像原信噪比 5.7602 提升为 22.7686，并分析了滤波器参数等对图像恢复的影响。针对问题 2，建立了多种噪声模型并设计了基于频域的混合噪声去除算法，该算法相较于其他滤波算法降噪效果明显，经过滤波处理后信噪比都有显著提高。针对问题 3，分析了过采样率和相关先验信息对图像恢复概率的影响，大过采样率并不利于提高图像恢复概率，掌握先验信息有利于针对性的采取有去噪措施，提高图像恢复概率。

本文设计的去噪算法相较于传统方法更加高质量实现图像恢复，大量对比实验验证了方法的有效性与优越性并将所提算法用于附件数据处理。实际上影响图像恢复的因素很多，问题更为复杂，为更高效实现图像恢复，本文提出自适应自主参数寻优的模型改进方向。

**关键字：** 图像恢复 频域滤波 去噪 傅里叶变换

# 目录

<b>1. 问题重述</b> .....	<b>4</b>
1.1 问题背景 .....	4
1.2 问题重述 .....	4
1.3 资料条件 .....	4
1.4 模型基本假设 .....	4
<b>2. 问题 1 的建模与求解</b> .....	<b>5</b>
2.1 问题 1 分析 .....	5
2.2 巴特沃斯低通滤波算法 .....	5
2.2.1 傅里叶变换 .....	5
2.2.2 巴特沃斯低通滤波器 .....	6
2.3 实验与分析 .....	7
2.4 去噪效果分析 .....	7
2.5 模型评价 .....	9
<b>3. 问题 2 的建模与求解</b> .....	<b>10</b>
3.1 问题分析 .....	10
3.2 典型噪声建模 .....	10
3.2.1 高斯噪声 .....	10
3.2.2 泊松噪声 .....	10
3.2.3 脉冲噪声 .....	11
3.2.4 瑞利噪声 .....	13
3.3 混合噪声识别与去噪算法 .....	13
3.3.1 高-低频分离 .....	13
3.3.2 高频去噪 .....	15
3.4 实验对比分析 .....	18
3.5 模型评价 .....	18
<b>4. 问题 3 的建模与求解</b> .....	<b>22</b>
4.1 问题分析 .....	22
4.2 过采样率与图像恢复概率 .....	22
4.3 相关先验信息与图像恢复概率 .....	22
4.4 模型评价 .....	23

<b>5. 模型评价与改进</b> .....	<b>25</b>
<b>参考文献</b> .....	<b>26</b>
<b>附录 A 过采样率和先验信息的图像恢复影响对比图</b> .....	<b>27</b>
<b>附录 B MATLAB 源程序</b> .....	<b>33</b>
2.1 问题 1 主要程序 .....	33
2.2 问题 2 主要程序 .....	34
2.3 问题 3 主要程序 .....	37
2.4 信噪比计算函数 .....	40

# 1. 问题重述

## 1.1 问题背景

随着现代科学技术的发展，各领域对于图像质量的要求越来越高，然而，在实际过程中，图像的获取不可避免地会受到外界的干扰，例如环境亮度，光线强度或者硬件运算精度等等。图像去噪技术程度很大程度地影响着信息利用的水平。图像去噪方法主要可以分为空间域和频率域处理方法，其中频率域处理方法最早可以追溯到傅立叶变换的使用，它可以将信息从空间域转化为频率域进行分析，是图像去噪的重要手段，在天文学、遥感成像、医疗图像等领域获得了重要应用。

## 1.2 问题重述

基于以上研究背景，题目给出了三幅  $256 \times 256$  图像的含噪声频谱数据信息，本文在此基础上需要解决以下三个问题：

问题 1：基于图像含噪声频谱对单幅图像进行去噪。对含单一噪声的图像频谱进行处理，建立相关数学模型，给出去噪算法，恢复图像信息，并分析相应图像去噪效果的因素。

问题 2：基于图像含噪声频谱对其进行混合去噪。对于含义混合噪声的图像，基于频域法，建立相关数学模型，给出去噪算法，恢复图像信息。

问题 3：基于问题 1 和问题 2，分析过采样和先验信息如何影响图像的恢复概率，并给出以上图像的恢复概率。

## 1.3 资料条件

附件 1-6 提供了图像恢复技术问题的基本数据信息，各文件的详细说明如下：附件 1 提供了问题 1 的图像噪声频谱数据信息，附件 2 和附件 3 提供了问题 2 的图像噪声频谱数据信息，附件 4 为问题 1 图像的恢复结果，待写入；附件 5 和附件 6 为问题 2 中两幅图像的恢复结果，待写入。附件 1-3 为已知题给信息，附件 4-6 为待解信息。

## 1.4 模型基本假设

- (1) 噪声是随机的，与图像中的物体无关。
- (2) 文中设计的噪声是加性的，即噪声是图像中已有信息的简单叠加。
- (3) 图像中信息都是平稳的，即它们的统计特性不随时间变化。

## 2. 问题 1 的建模与求解

### 2.1 问题 1 分析

要求基于已给的图像含噪频谱数据，对给定一幅  $256 \times 256$  的图像频谱噪声数据进行去噪。本文主要基于傅里叶变换进行图像去噪处理。从数学意义上看，傅里叶变换将一个图像转换为一系列周期函数来进行处理。从物理效果上看，傅里叶变换将图像信息从空间域转换到频率域，即将图像的灰度分布函数转换为图像的频率分布函数<sup>[1]</sup>。基本原理如下所示：图像中的噪声往往对应着图像上灰



图 1 任务流程图

度值的突变，因此属于高频部分，图像中的其他大部分内容则主要集中在低频部分<sup>[2]</sup>。原始图像进行二维离散傅里叶变换可以得到这个图像每个像素的相位和幅度值。相位代表图像中的位置形状信息，一般保留并不对其做处理，以保证可以恢复出原始图像的形状；幅度值主要表示能量的大小，幅值越高能量越大，因此只要将高频部分的能量值置为 0，就可以去除噪声。在进行傅里叶变换（Matlab 对应“fft2”函数）后，低频部分集中分布在四个角落，通过移位（Matlab 对应“fftshift”函数）处理之后，将其集中到图像的中心，以便于进行滤波等处理。当完成对幅值的处理后，再进行反移位（Matlab 对应“ifftshift”函数）恢复到原始的分布情况，之后结合相位计算出每个像素对应的值，再将其进行傅里叶反变换（Matlab 对应“ifft”函数），此时图像中的噪声便得以去除，相应的时域图像也就得以恢复<sup>[3]</sup>。

### 2.2 巴特沃斯低通滤波算法

#### 2.2.1 傅里叶变换

代数形式的傅里叶变换<sup>[4]</sup>

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

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

其中

$$\begin{pmatrix} u, x = 0, 1, 2, \dots, M - 1; \\ v, y = 0, 1, 2, \dots, N - 1 \end{pmatrix}$$

$f(x, y)$  : 表示灰度图像,  $F(u, v)$  为  $f(x, y)$  的频谱,  $M$  表示  $f(x, y)$  的行数,  $N$  为  $f(x, y)$  的列数。

矩阵形式的二维离散傅里叶变换

$$\begin{cases} F = G_1 f G_2 \\ f = G_3 F G_4 \end{cases} \quad (3)$$

其中

$$G_1 = \begin{bmatrix} e^{-j2\pi \frac{0 \cdot 0}{M}} & e^{-j2\pi \frac{0 \cdot 1}{M}} & \dots & e^{-j2\pi \frac{0 \cdot (M-1)}{M}} \\ e^{-j2\pi \frac{1 \cdot 0}{M}} & e^{-j2\pi \frac{1 \cdot 1}{M}} & \dots & e^{-j2\pi \frac{1 \cdot (M-1)}{M}} \\ \dots & \dots & \dots & \dots \\ e^{-j2\pi \frac{(M-1) \cdot 0}{M}} & e^{-j2\pi \frac{(M-1) \cdot 1}{M}} & \dots & e^{-j2\pi \frac{(M-1) \cdot (M-1)}{M}} \end{bmatrix}$$

$$G_2 = \begin{bmatrix} e^{-j2\pi \frac{0 \cdot 0}{N}} & e^{-j2\pi \frac{0 \cdot 1}{N}} & \dots & e^{-j2\pi \frac{0 \cdot (N-1)}{N}} \\ e^{-j2\pi \frac{1 \cdot 0}{N}} & e^{-j2\pi \frac{1 \cdot 1}{N}} & \dots & e^{-j2\pi \frac{1 \cdot (N-1)}{N}} \\ \dots & \dots & \dots & \dots \\ e^{-j2\pi \frac{(N-1) \cdot 0}{N}} & e^{-j2\pi \frac{(N-1) \cdot 1}{N}} & \dots & e^{-j2\pi \frac{(N-1) \cdot (N-1)}{N}} \end{bmatrix}$$

$$G_3 = G_1^{-1} = \frac{1}{M} \begin{bmatrix} e^{-j2\pi \frac{0 \cdot 0}{M}} & e^{-j2\pi \frac{0 \cdot 1}{M}} & \dots & e^{-j2\pi \frac{0 \cdot (M-1)}{M}} \\ e^{-j2\pi \frac{1 \cdot 0}{M}} & e^{-j2\pi \frac{1 \cdot 1}{M}} & \dots & e^{-j2\pi \frac{1 \cdot (M-1)}{M}} \\ \dots & \dots & \dots & \dots \\ e^{-j2\pi \frac{(M-1) \cdot 0}{M}} & e^{-j2\pi \frac{(M-1) \cdot 1}{M}} & \dots & e^{-j2\pi \frac{(M-1) \cdot (M-1)}{M}} \end{bmatrix}$$

$$G_4 = G_2^{-1} = \frac{1}{N} \begin{bmatrix} e^{-j2\pi \frac{0 \cdot 0}{N}} & e^{-j2\pi \frac{0 \cdot 1}{N}} & \dots & e^{-j2\pi \frac{0 \cdot (N-1)}{N}} \\ e^{-j2\pi \frac{1 \cdot 0}{N}} & e^{-j2\pi \frac{1 \cdot 1}{N}} & \dots & e^{-j2\pi \frac{1 \cdot (N-1)}{N}} \\ \dots & \dots & \dots & \dots \\ e^{-j2\pi \frac{(N-1) \cdot 0}{N}} & e^{-j2\pi \frac{(N-1) \cdot 1}{N}} & \dots & e^{-j2\pi \frac{(N-1) \cdot (N-1)}{N}} \end{bmatrix}$$

### 2.2.2 巴特沃斯低通滤波器

本文采用巴特沃斯低通滤波器<sup>[5]</sup>, 滤波器数学模型可用如下振幅的平方对频率的公式表示:

$$|H(\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}} = \frac{1}{1 + \varepsilon^2 \left(\frac{\omega}{\omega_p}\right)^{2n}} \quad (4)$$

其中,  $n$  = 为滤波器的阶数,  $\omega_c$  = 为截止频率, 即振幅下降为-3 分贝时的频率,  $\omega_p$  = 为通频带边缘频率,  $\frac{1}{1+\varepsilon^2} = |H(\omega)|^2$  是在通频带边缘的数值。

### 2.3 实验与分析

为验证低频滤波算法模型的可行性，需构造测试图像，噪声，以及评价指标。采用某  $256 \times 256$  的图像，首先添加高斯噪声，然后按照题目中描述的去采样流程，对测试集图片进行过采样，最后通过傅里叶变换和低通滤波器进行降噪恢复。评价指标采用信噪比（SNR），信噪比是在输入图像和输出图像之间的均方根误差（MSE）基础上定义的

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left( \hat{f}(i, j) - f(i, j) \right)^2 \quad (5)$$

$$SNR = 10 \lg \left( \frac{\sigma_f^2}{MSE} \right) \quad (6)$$

其中  $\hat{f}(i, j)$  为去噪后的灰度图，

$$\sigma_f^2 = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \hat{f}(i, j)^2$$

得到如图2所示，可以看 100Hz 低通滤波效果较好，能够比较完整地恢复原图信息。经计算，未去噪图像、经过 50Hz 低通滤波去噪、经过 100Hz 低通滤波去噪的信噪比分为 5.7602、15.7840、22.7686，这与肉眼直观感受一致。结果表明测试集图片能够较好地得以恢复，从而证明了所建立去噪算法的可行性。将所用模型方法应用于题给图像频谱数据信息集，对图像信息进行恢复，如下图所示：

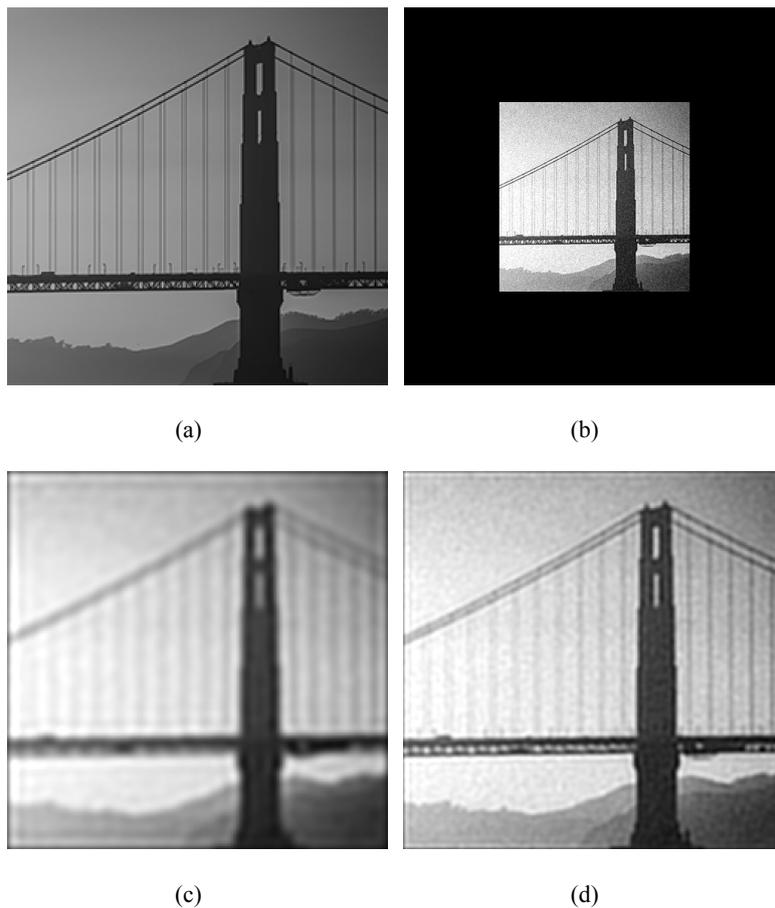
### 2.4 去噪效果分析

去噪效果的评价标准主要可以分为主观评价和客观评价两方面。作为自然界中具有高度智慧的人类，在经过漫长历史进化后，获得了对视觉图像的感受能力。因此，通过主观评价可以初步判断去噪效果的好坏，该方法简单有效。一般经验丰富的专家，往往比一般人具有更高的感受能力，评价去噪效果会更加权威。然而图像去噪效果难以肉眼区分，或者需要有大量图像需要判断，单纯依靠主管经验评价不切实际，因此需要客观评价指标。从客观角度来分析图像的恢复效果，可以考虑以下几个方面

(1) 信噪比（SNR）：信噪比越高，表示图像中 useful 信息与无用信息的比例越大，图像质量越好。

(2) 峰值信噪比（PSNR）：PSNR 是一种常用的衡量图像质量的指标，它表示原始图像与恢复后图像之间的差异程度。

(3) 结构相似性指数（SSIM）：SSIM 是一种综合考虑了亮度、对比度和结构信息的图像质量评价指标。



**图 2 低通滤波算法实验: (a) 原始灰度图; (b) 带有高斯噪声的过采样图; (c) 经过 50Hz 低通滤波后的灰度图; (d) 经过 100Hz 低通滤波后的灰度图**

本文中采用了信噪比进行去噪效果判断，所得结论与主观评价一致。

影响图像恢复的因素有很多

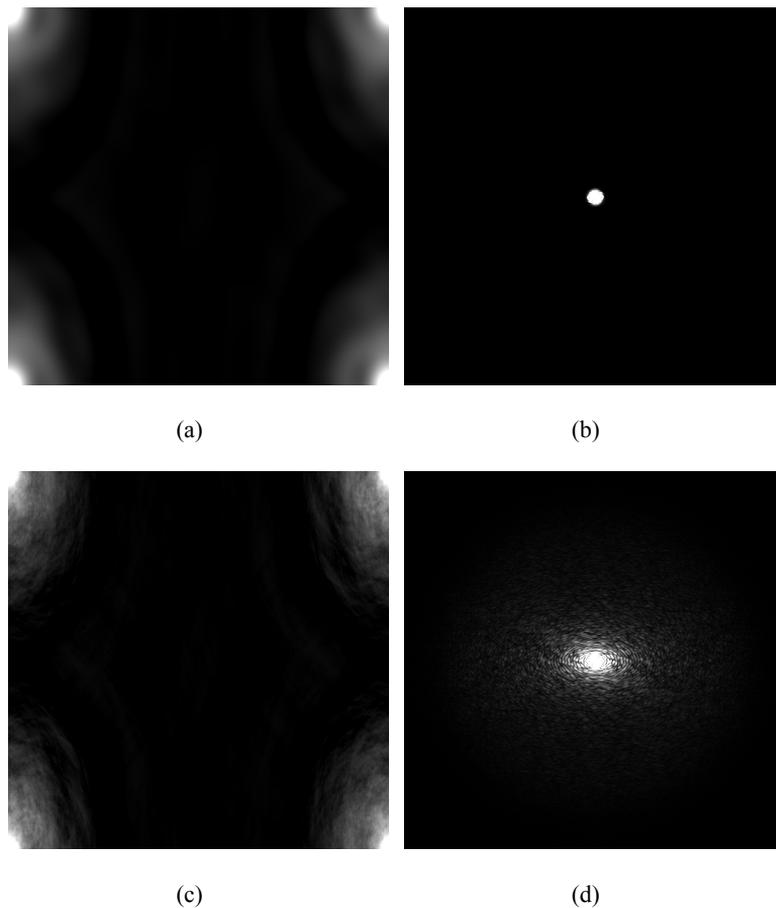
(1) 噪声类型：不同类型的噪声需要采用不同的去噪算法。例如，高斯噪声可以使用 Wiener 滤波器或 Kalman 滤波器进行去噪，而椒盐噪声则需要使用中值滤波器或双边滤波器等。

(2) 噪声水平：噪声水平越高，恢复图像的难度就越大。在实际应用中，通常会对图像进行预处理，以降低噪声水平或减小其对恢复结果的影响。

(3) 原始图像质量：原始图像的质量越高，恢复出的图像质量也越好。因此，在进行图像恢复之前，需要对原始图像进行适当的预处理和增强。

(4) 去噪算法：不同的去噪算法适用于不同类型的噪声和图像质量。目前常用的去噪算法包括均值滤波、中值滤波、双边滤波、Wiener 滤波、小波变换等。

(5) 参数设置：不同的去噪算法需要不同的参数设置才能取得最佳效果。因此，在进行图像恢复时，需要根据具体情况调整参数设置。



**图3 附件1数据的低通滤波处理效果: (a) 50Hz 低通滤波图; (b) 50Hz 低通滤波频谱图; (c) 100Hz 低通滤波图; (d) 100Hz 低通滤波图**

(6) 硬件设备：硬件设备的性能和质量也会影响图像恢复的效果。例如，高性能的计算机可以处理更大的图像数据集，从而提高恢复效果。

本文中比较了低通滤波算法的参数设置，一个为 50Hz，一个为 100Hz，两个滤波器所实现的去噪效果明显不同，同样的去噪算法也需要不同的参数设置才能取得最佳效果。

## 2.5 模型评价

本节针对图像含噪声频谱数据，设计了低通滤波算法，该算法模型可以实现对含噪图像的去噪。实验结果表明通过低通滤波去噪算法，图像的信噪比得到提升，验证了算法的有效性。然而，不同滤波器参数的设计对滤波效果有明显影响，如何选取最佳滤波参数是一个值得进一步研究的问题，可以考虑启发式优化方法进一步确定滤波器参数。另外，高频信号中除噪声外仍有图像原始信息，低通滤波器在去噪的同时也去除了部分有效信息，针对该问题，接下来我们结合问题 2 进行了分析和建模。

## 3. 问题 2 的建模与求解

### 3.1 问题分析

问题 2 与问题 1 相比, 图片中包含多种类型噪声的混合。通过对两幅  $256 \times 256$  的图像的频谱进行过采样, 得到各自相应的  $512 \times 512$  含噪声频谱信息, 需设计混合噪声的去噪算法, 并恢复原图像。首先, 需要分析掌握典型噪声类别和特性, 以便生成混合噪声, 研究混合噪声去噪算法。然后设计基于傅里叶变换的频域滤波算法。巴特沃斯滤波器将图像的高频信息与低频信息分离, 其中高频信息中往往对应着噪声信息, 即灰度值突变的部分, 但是图像边缘等也包含在内, 因此考虑将高频信息进一步处理, 剔除噪声保留图像信息, 然后再将处理后的高频信息与低频信息融合以生成原始高质量图像。最后进行实验仿真和对比分析, 并将算法用于题目附件中所提供的数据。

### 3.2 典型噪声建模

#### 3.2.1 高斯噪声

概率密度函数服从高斯分布的噪声称为高斯噪声<sup>[6]</sup>, 若其幅度分布服从高斯分布且功率谱密度服从均匀分布则称为高斯白噪声。数字图像处理领域, 高斯噪声是最为常见的一种噪声, 其来源包括摄像时光线强度与分布、拍摄设备的电路元件性能和工作状态等<sup>[7, 8, 9]</sup>。大多数噪声近似符合高斯分布, 这一特征降低了对噪声进行数学分析的难度。高斯噪声的概率密度函数为

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[ -(z - \mu)^2 / 2\sigma^2 \right] \quad (7)$$

其中,  $z$  表示像素灰度值,  $\mu$  为  $z$  的期望,  $\sigma$  为  $z$  的标准差。

以常值灰度图像为例, 分析高斯噪声特性。首先任意建立一个常值灰度图像, 如图4所示, 可以看到常值灰度图对应的灰度值为常值。在此基础上, 添加高斯噪声, 得到的灰度图与直方图如图5所示。可以看到, 带有高斯噪声灰度图像的直方图明显呈现高斯分布, 此时说明作用在常值灰度图上的噪声为高斯分布噪声。

#### 3.2.2 泊松噪声

概率密度函数服从泊松分布的一类噪声成为泊松噪声, 泊松分布一般用于描述单位时间内随机事件发生的次数的概率分布。当拍摄环境亮度较小时出现泊松噪声, 设  $z$  取值为 0、1、2、... 的离散随机变量, 则其概率密度函数为

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[ -(z - \mu)^2 / 2\sigma^2 \right] \quad (8)$$

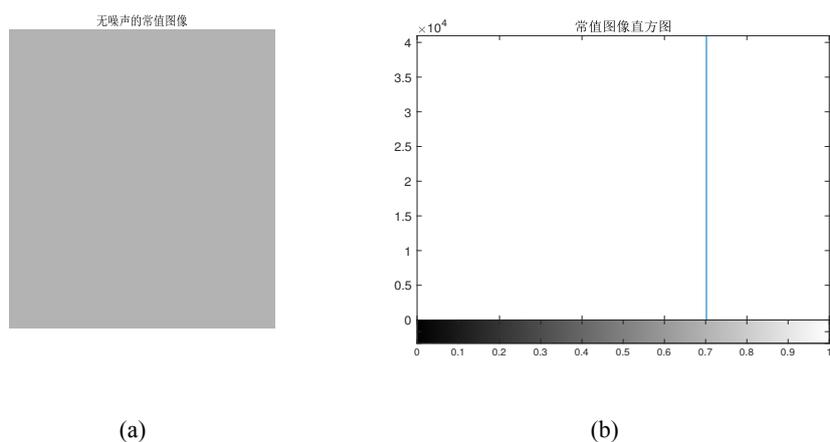


图4 常值灰度图像的像素图与直方图: (a) 常值灰度图像的像素图; (b) 常值灰度图像的直方图

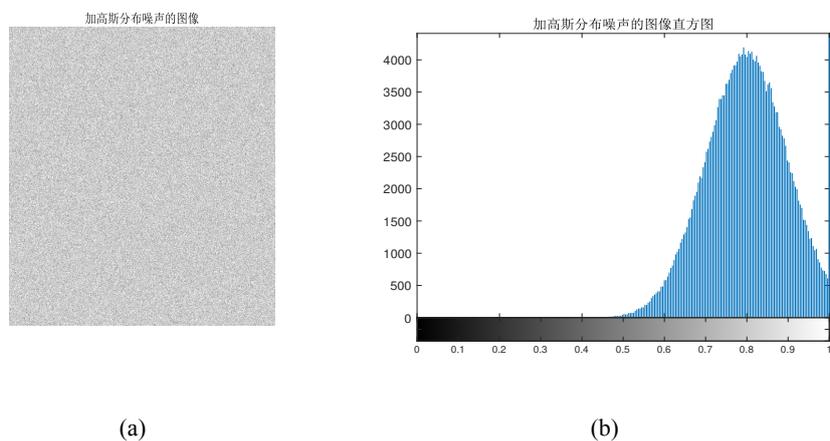


图5 带有高斯噪声的灰度图像及其直方图: (a) 带有高斯噪声的灰度图像; (b) 带有高斯噪声灰度图像的直方图

其中  $\lambda$  为  $z$  均值和方差. 在常值灰度图基础上, 添加泊松分布噪声, 得到的灰度图与直方图如图6所示. 可以看到, 带有泊松分布噪声灰度图像的直方图明显呈现泊松分布, 此时说明作用在常值灰度图上的噪声为泊松分布噪声。

### 3.2.3 脉冲噪声

脉冲噪声随机的改变图像中某些像素值, 其变化幅度大且不规则, 呈现尖峰脉冲状态<sup>[10]</sup>. 在二值图像上, 脉冲噪声表现为将一些黑像素点变白或将白像素点变黑. 脉冲噪声一般由图形分割造成, 如黑白电视机常出现的“雪花”即为脉

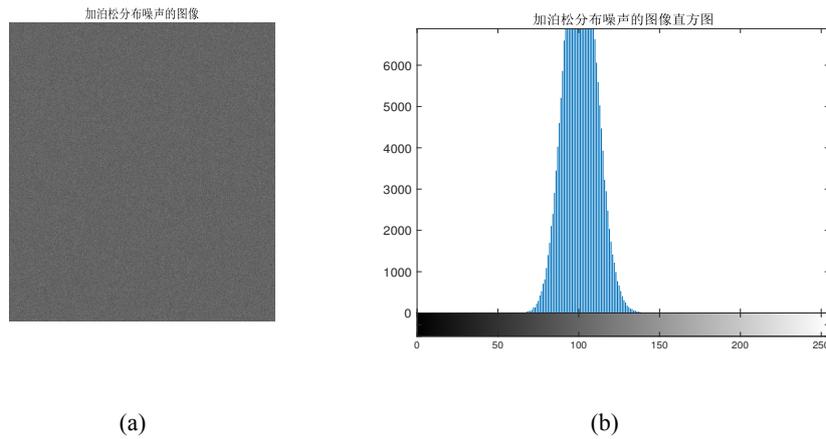


图 6 带有泊松分布噪声的灰度图像及其直方图: (a) 带有泊松分布噪声的灰度图像; (b) 带有泊松噪声灰度图像的直方图

冲噪声。脉冲噪声的概率密度函数为

$$p(z) = \begin{cases} p_a, z = a \\ p_b, z = b \\ 0, \text{others} \end{cases} \quad (9)$$

当  $p_a$  与  $p_b$  均非 0 则称为双极脉冲噪声或椒盐噪声，本文中主要考虑椒盐噪声。在常值灰度图基础上，添加椒盐噪声，得到的灰度图与直方图如图7所示。可以看到，带有椒盐噪声灰度图像的直方图明显脉冲特点，集中在灰度值的最大值与最小值，此时说明作用在常值灰度图上的噪声为椒盐噪声。

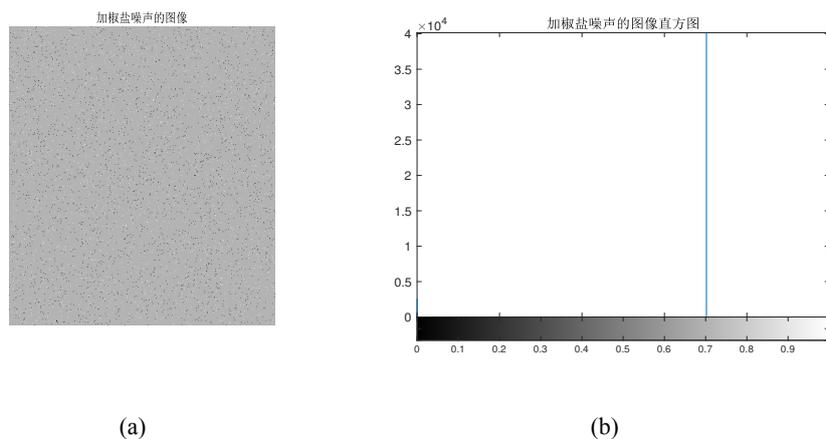


图 7 带有椒盐噪声的灰度图像及其直方图: (a) 带有椒盐噪声的灰度图像; (b) 带有椒盐噪声灰度图像的直方图

### 3.2.4 瑞利噪声

概率密度函数服从瑞利分布的一类噪声成为瑞利噪声，常见于对独立多径分量接受包络统计时变特性或平坦衰落信号接收包络的描述，若连续随机变量  $z$  服从瑞利分布，则其概率密度函数为

$$p(z) = \begin{cases} 2(z-a)/b \exp[-(z-a)^2/b], & z \geq a \\ 0, & z < a \end{cases} \quad (10)$$

其中， $z$  的均值为  $\mu = a + \sqrt{\pi b/4}$  方差为  $\sigma^2 = b(4 - \pi)/4$ 。在常值灰度图基础上，添加瑞利噪声，得到的灰度图与直方图如图8所示。可以看到，带有瑞利噪声灰度图像的直方图呈现瑞利分布特性，此时说明作用在常值灰度图上的噪声为瑞利噪声。

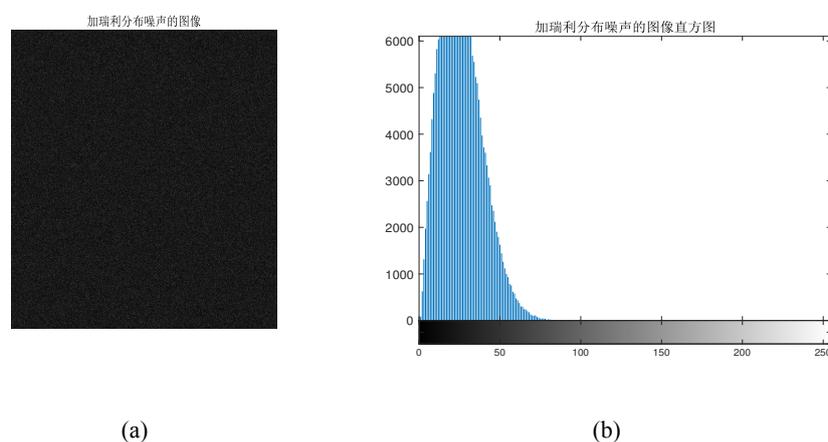


图8 带有瑞利噪声的灰度图像及其直方图: (a) 带有瑞利噪声的灰度图像; (b) 带有瑞利噪声灰度图像的直方图

以某  $256 \times 256$  像素的图像为例，如图9所示，依次对其添加高斯分布噪声，泊松分布噪声，椒盐噪声，瑞利分布噪声，如图10所示。在噪声密度方面，高斯分布噪声、泊松分布噪声、瑞利分布的密度分布相似均属于密集分布型，噪点之间存在混叠，而椒盐噪声分布稀疏，噪点之间独立。

## 3.3 混合噪声识别与去噪算法

### 3.3.1 高-低频分离

图像经过傅里叶变换后得到频谱图，其中低频部分对应着图形中呈现平滑特性的灰度分量而高频部分对应着图形中物体轮廓和灰度值突变的部分。滤波器可以对频域的图像进行处理，其中低通滤波器可以截断高频信息，高通滤波器可以阻断低频信息。噪声往往是高频的，问题1中使用低通滤波器将高频信息过



图9 原始图像灰度图

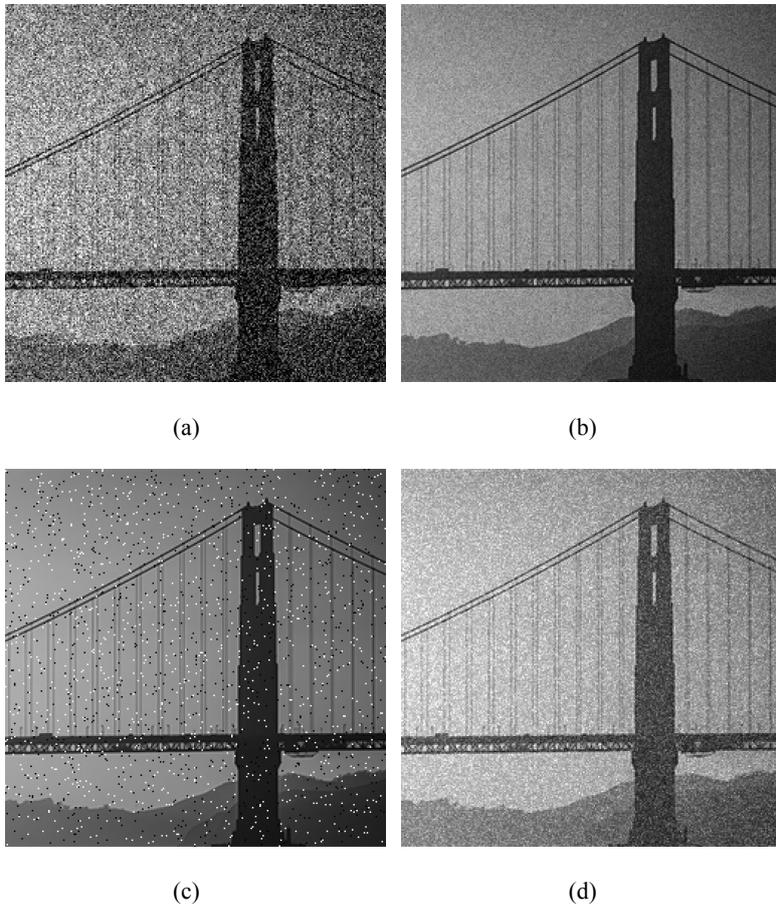


图10 四种噪声对原图像的影响: (a) 高斯噪声对原始图像的影响; (b) 泊松噪声对原始图像的影响; (c) 椒盐噪声对原始图像的影响; (d) 瑞利噪声对原始图像的影响

滤，得到了去噪图像。其中高频信息中往往对应着噪声信息，即灰度值突变的部分，但是图像边缘等也包含在内，因此考虑将高频信息进一步处理。本文采用巴特沃斯滤波器将图像的高频信息与低频信息分离，低通巴特沃斯滤波器幅度平

方公式为

$$|H(\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}} = \frac{1}{1 + \varepsilon^2 \left(\frac{\omega}{\omega_p}\right)^{2n}} \quad (11)$$

高通巴特沃斯滤波器幅度平方公式为

$$|H(\omega)|^2 = 1 - \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}} = 1 - \frac{1}{1 + \varepsilon^2 \left(\frac{\omega}{\omega_p}\right)^{2n}} \quad (12)$$

选择相同的截断频率  $\omega_c$ ，可以将图形频谱信息分离成互补的高频与低频部分。选择截止频率  $\omega_c = 40\text{Hz}$ ，对图像添加高斯噪声并进行过采样，如图11所示，左边一列为像素图，右边一列为频谱图。(a) 为原图像，(b) 为原图像对应的频谱图，对其进行高频滤波之后，得到的频谱图如图 (d) 所示，可以看到其保留了中心处的低频信息，四周高频信息被过滤掉，高频滤波所对应的像素图如图 (c) 所示。对原图进行低频滤波，得到了高频频谱图如图 (f) 所示，中心处的低频信息被过滤掉，对应的像素图如图 (e) 所示，可以看出，高频部分保留了图像更多的边缘信息。

### 3.3.2 高频去噪

获得图像高频信息后可以通过傅里叶反变换得到像素图像，传统针对像素图像进行滤波的方法有均值滤波算法和中值滤波算法。然而，传统均值算法与中值算法在处理时将窗口内所有像素点同时进行处理，原始像素点大部分发生了改变，同时噪声的灰度值也被计算在内。针对传统去噪方法的问题，本文采用一种噪声识别去噪算法。第一，需要判断某像素点是否大概率为噪声，不是噪声的像素点可不进行处理。第二，对于判断为噪声的像素点用其周边 8 个邻域像素进行表示，噪声灰度值本身不参与运算，如图12所示。具体地，设定一个阈值  $\delta$  用于判断某像素点的邻域内是否存在与其近似的灰度像素，若无近似像素点则判断该像素点为孤立噪声点，然后通过使用周边 8 个邻域像素的平均值替换该点灰度值。取阈值  $\delta = 60$  对高频图像进行改进中值滤波处理，如图13所示。可以看到，对高频部分对应的图像进行中值滤波后去掉图中突变的一些像素点，在与低频信息融合后又得到了完整的频谱图 (d)，经过傅里叶反变换之后得到滤波后的图像如图 (c) 所示，与原图像相比，图像变得更加清晰。

综上所述，整个算法整个流程可以总结为：

(1) 通过巴特沃斯滤波器获得图像互补高低频频谱图，然后将高频部分（包含有的边缘与噪声）通过傅里叶逆变换转化为空间域，得到像素图像。

(2) 设  $f(x, y)$  为高频部分对于像素图像中某点灰度值，其中  $x, y$  表示像

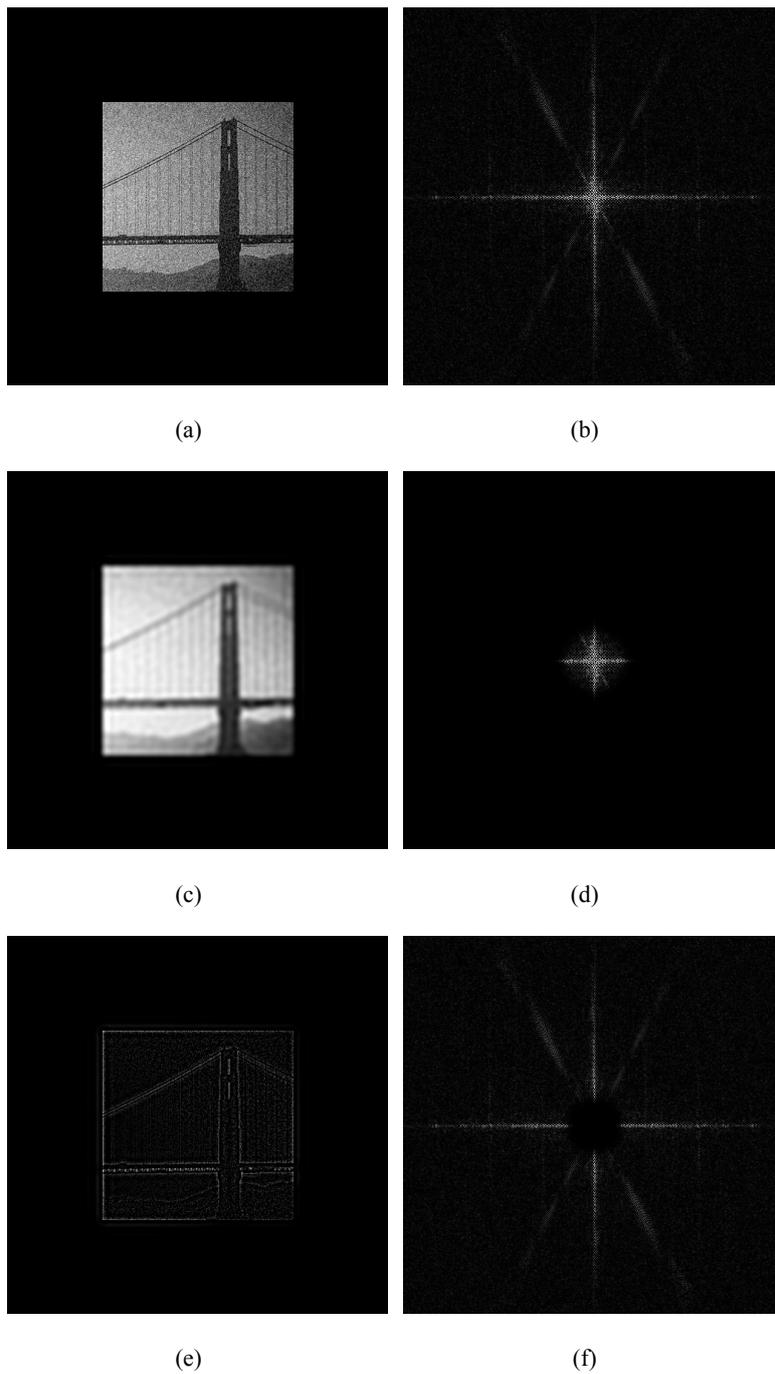


图 11 高频与低频对应的图像信息: (a) 加高斯噪声的原图像; (b) 原图像频谱图; (c) 低通滤波后的图像; (d) 低通滤波后的频谱图; (e) 高通滤波后的图像; (f) 高通滤波后的频谱图

素位置，选取阈值  $\delta$ ，判断公式

$$|f(x, y) - f(x + a, y + b)| \geq \delta$$

是否成立，若成立转 (3)，若不成立窗口转到下一个像素点。式中,  $a, b = -1, 0, 1$

$$\frac{1}{8} \times$$

1	1	1
1	0	1
1	1	1

图 12 改进的 3×3 均值窗口

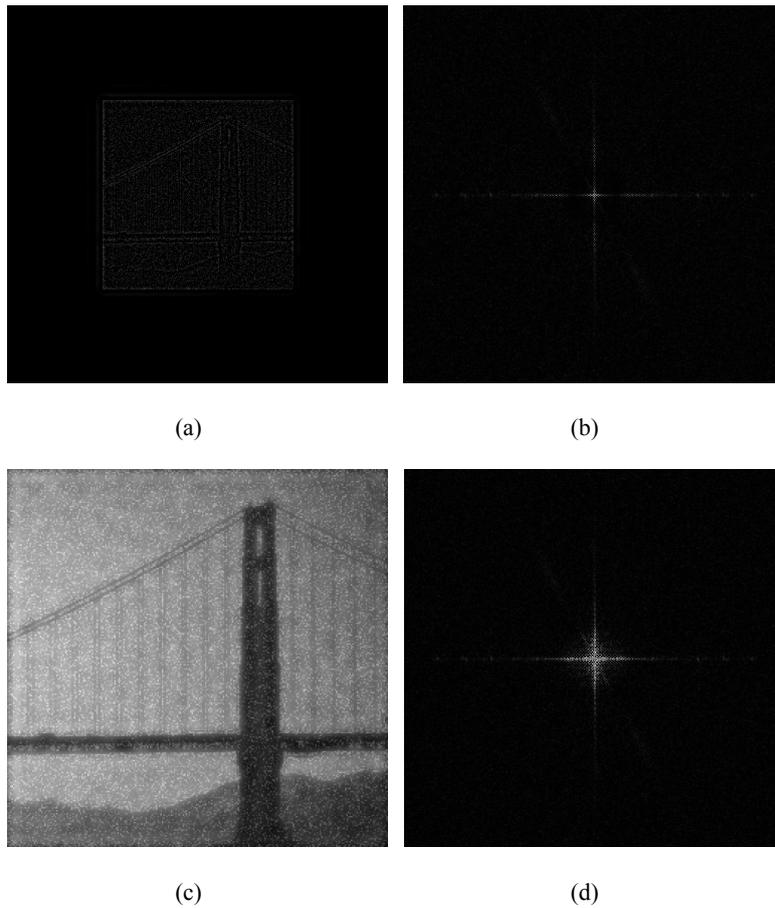


图 13 高频中值滤波处理与高低频重新融合: (a) 高频去噪后的图像; (b) 高频去噪后的频谱图; (c) 高频低频融合降噪后的图像; (d) 高频低频融合后的频谱图

且不同时为 0。

(3) 该像素为噪声点，则计算其邻域像素的均值  $\bar{f}(x, y)$  并用其替换原点的

像素值

$$\bar{f}(x, y) = \frac{1}{8} \begin{bmatrix} f(x-1, y-1) + f(x-1, y) + f(x-1, y+1) \\ +f(x, y-1) + f(x, y+1) + f(x+1, y-1) + \\ f(x+1, y) + f(x+1, y-1) \end{bmatrix}$$

(4) 将去噪后的高频部分图像通过傅里叶变换转化为频谱图，然后与低频频谱图进行融合得到完整频谱图，最后再通过傅里叶逆变换得到去噪后的完整图像。

### 3.4 实验对比分析

考虑到脉冲噪声在能量分布上与其他几种噪声有明显不同，设计三种混合噪声模式，分别为：脉冲-高斯噪声、脉冲-瑞利噪声、高斯-泊松-瑞利噪声。图像的降噪效果采用信噪比衡量，实验效果对比图如图14所示。图中(a)、(f)、(k)分别为脉冲噪声+高斯噪声、脉冲噪声+瑞利噪声、高斯噪声+泊松噪声+瑞利噪声污染的图像，对应的每一行分别为经过均值滤波、中值滤波、低通滤波以及本文算法处理后的效果图。计算降噪前后图像的信噪比值，如表1所示。脉冲-高斯混合噪声对图像质量影响较大，在进行滤波处理前信噪比较低，脉冲-瑞利混合噪声对图像质量影响较小，在进行滤波处理前信噪比较高。本文算法相较于其他滤波算法降噪效果明显，经过滤波处理后信噪比都有显著提升。

表 1：不同降噪算法降噪前后信噪比值

噪声类别	均值滤波	中值滤波	低通滤波	本文算法
脉冲-高斯噪声	1.9406/4.9324	1.9581/4.6585	1.9751/5.7446	1.9948/10.1760
脉冲-瑞利噪声	16.7135/16.442	16.9587/22.5789	16.8910/13.6137	16.7012/31.5859
三种混合噪声	9.3911/16.6195	9.4492/20.054	9.4396/14.1309	9.5548/30.8604

将本文算法用于附件2与附件3中数据，可以得到滤波前后灰度图与频谱图，分别如图15与图16所示。可以主观看出，其灰度图像清晰度明显提高。

### 3.5 模型评价

文中基于频域滤波将图像的高频部分与低频部分分离处理，然后对高频部分进行噪声识别，并采用改进滑动中值滤波窗口对高频图像中噪声部分进行去除，保留有用信息，最后再将处理后的高频部分与低频部分融合，并恢复出原图像。通过对比实验表明该算法相对于其他算法具有明显优势，将其用于附件中

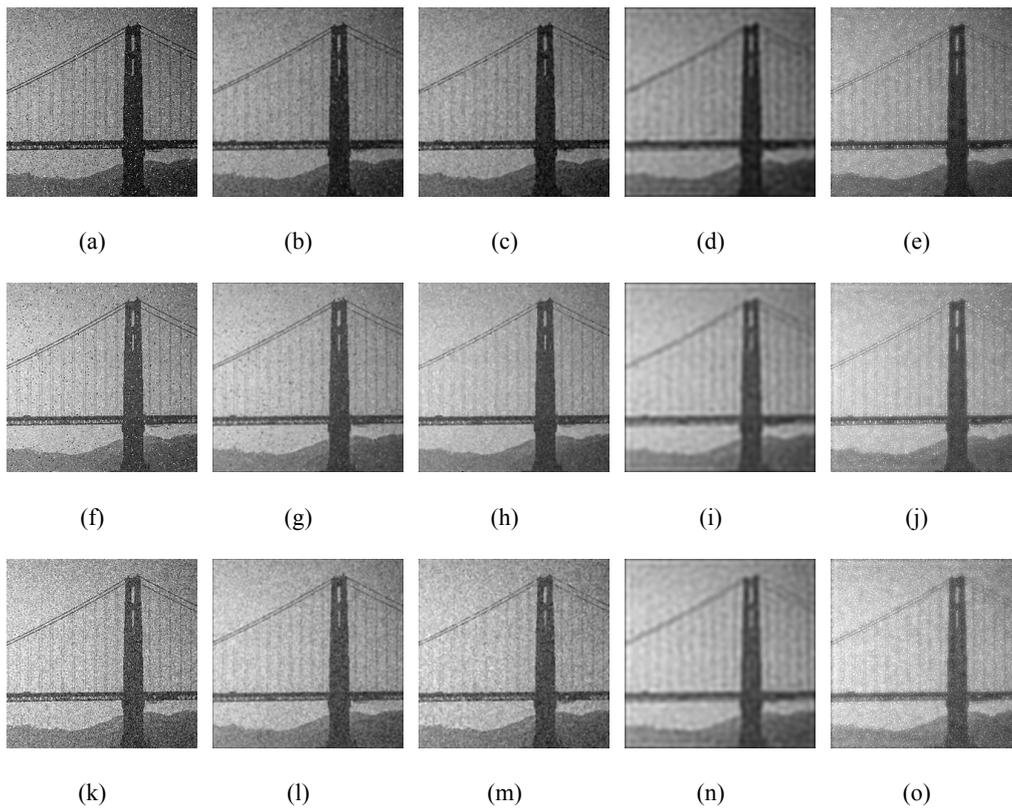


图 14 实验效果对比: (a) 脉冲-高斯噪声混合污染图像; (b) 脉冲-高斯噪声混合污染图像均值滤波后图像; (c) 脉冲-高斯噪声混合污染图像中值滤波后图像; (d) 脉冲-高斯噪声混合污染图像低通滤波后图像; (e) 脉冲-高斯噪声混合污染图像本文方法滤波后图像; (f) 脉冲-瑞利噪声混合污染图像; (g) 脉冲-瑞利噪声混合污染图像均值滤波后图像; (h) 脉冲-瑞利噪声混合污染图像中值滤波后图像; (i) 脉冲-瑞利噪声混合污染图像低通滤波后图像; (j) 脉冲-瑞利噪声混合污染图像本文方法滤波后图像; (k) 三种噪声混合污染图像; (l) 三种噪声混合污染图像均值滤波后图像; (m) 三种噪声混合污染图像中值滤波后图像; (n) 三种噪声混合污染图像低通滤波后图像; (o) 三种噪声混合污染图像本文方法滤波后图像;

所提供数据去噪，所得图像清晰度明显提高。然而，在滤波参数选择时多凭借经验，未能设计自适应参数寻优算法，模型仍有改进空间。

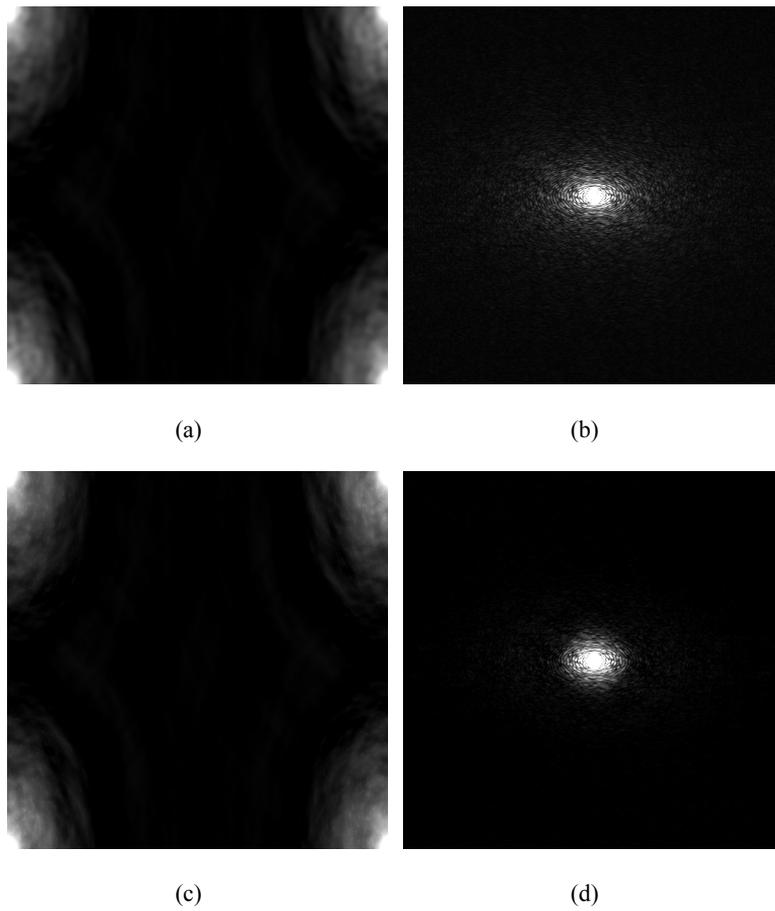


图 15 附件 2 的图像滤波: (a) 滤波前灰度图; (b) 滤波前频谱图; (c) 滤波后灰度图; (d) 滤波后频谱图

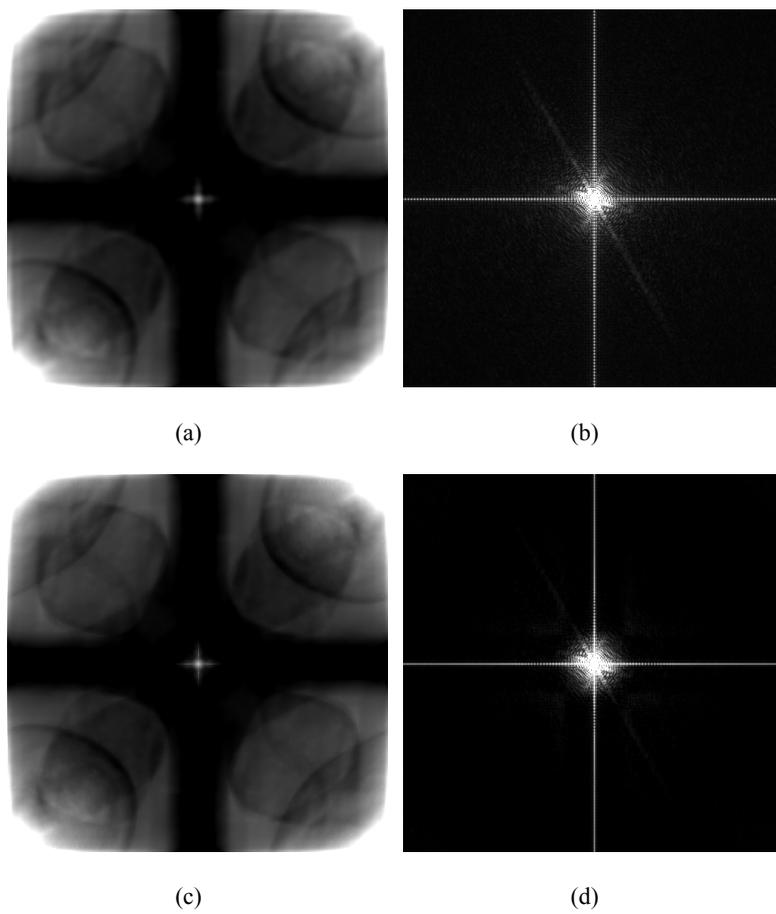


图 16 附件 3 的图像滤波: (a) 滤波前灰度图; (b) 滤波前频谱图; (c) 滤波后灰度图; (d) 滤波后频谱图

## 4. 问题3的建模与求解

### 4.1 问题分析

本题是在基于频谱信息来恢复图像的过程中，考虑频谱过采样率与图像相关先验信息如何影响图像的恢复概率，目标是提高图像恢复概率。图像恢复概率的量化与分析基于问题1、问题2已完成的模型。首先需要明晰过采样率以及先验信息的概念，然后利用统计学方法分析恢复概率问题，最后给出图像的恢复概率。

### 4.2 过采样率与图像恢复概率

二维带噪图像可以表示为

$$\hat{f}(x, y) = f(x, y) + n(x, y) \quad (13)$$

其中， $\hat{f}(x, y)$  为原始图像信号在  $(x, y)$  位置上的灰度值， $n(x, y)$  表示干扰噪声，且噪声为加性噪声。采用信噪比评价算法的图像去噪效果。

首先明确图像频谱的过采样率，对于一幅  $2^m \times 2^m$  的图像，对其频谱过采样，放在  $2^n \times 2^n$  大小的网格中心，剩余的像素值为 0，然后进行离散傅里叶变换，得到相应的  $2^n \times 2^n$  频谱信息，其过采样率  $R = 2^{n-m}$ 。由此可知，对于一幅  $256 \times 256$  的图像，对其频谱过采样，放在  $512 \times 512$  的网格中心，其过采样率  $R$  为 2。

为了能够更好的观察过采样率对图像恢复概率的影响，我们首先分析相关先验信息一定的情况下，即不改变噪声类型和参数，图像的恢复质量随过采样率从 2 至 7 的变化场景见附录 A 中图19至图24。信噪比随过采样率变化曲线如图17所示，从图中可以看出，每次实验滤波前的图像信噪比都近似，这是因为没有改变噪声的先验信息。然而，随着过采样率的提高，信噪比逐渐降低，并趋于稳定。我们将信噪比大于 20 设置为判定图像是否恢复的标准，对于同一类型和参数噪声，在不同过采样率下进行 200 次滤波实验，统计滤波后信噪比发现，对于过采样率为 2 的去噪方法可以 100% 实现图像恢复，而过采样率大于 2 情况下无法实现图像恢复。

### 4.3 相关先验信息与图像恢复概率

可以利用噪声相关先验信息来估计噪声的类型和水平，然后有针对性的采取有去噪措施，提高图像恢复概率。例如：

(1) 滤波方法。对于椒盐噪声，中值滤波具有较好的效果；对于高斯噪声，维纳滤波效果较好；对于乘性噪声，维纳滤波和高斯平滑滤波效果较好，而中值

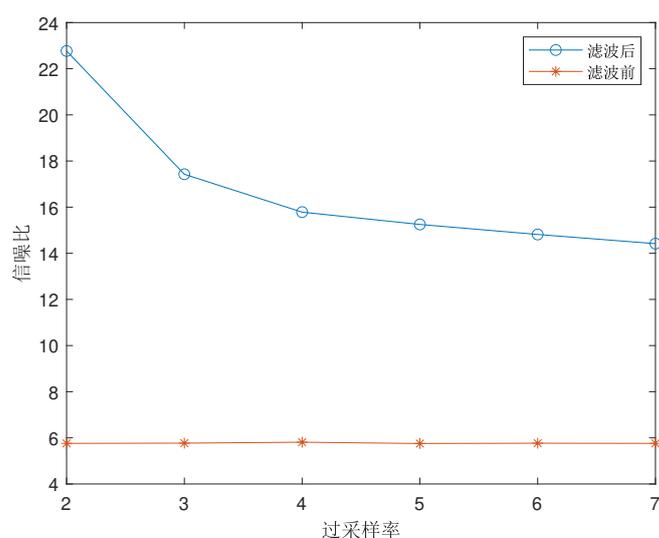


图 17 信噪比随过采样率变化曲线

滤波会稍差一些。

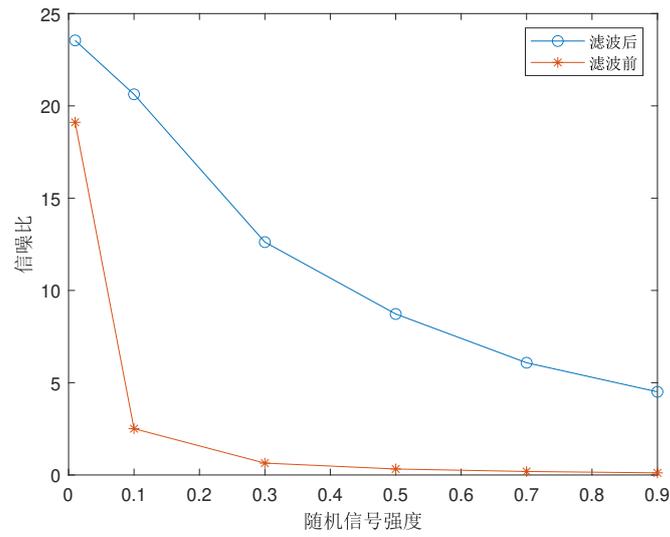
(2) 滤波参数。调节不同滤波方法中的模型参数能够有效调整滤波属性，从而影响图像去噪效果。

(3) 滤波频带。不同的频率范围也会对去噪效果产生影响，如上述实验过程中高通滤波的效果要好于低通滤波；又例如在逆滤波方法中靠近原点区域的滤波效果更好。

本文结合构造含噪声的频谱信息数据，分析噪声参数对滤波效果的影响。考虑同一类型噪声不同噪声强度，研究图像恢复效果。针对随机噪声，通过改变其幅值模拟噪声强度的改变，滤波器使用过采样率为 2 的低通滤波器。图像的恢复质量随噪声从 0.01 至 0.9 的变化场景见附录 A 中图25至图30。信噪比噪声强度的变化曲线如图18所示，从图中可以看出，每次实验滤波前后的图像信噪比都在下降，这是因为噪声强度增加导致信噪比降低，同时滤波器的性能也在下降。因此，为了提高图像恢复效果，需要改变滤波器参数或更换滤波器类型。我们将信噪比大于 20 设置为判定图像是否恢复的标准，对于同一类型和参数滤波器，在不同强度的随机噪声下分别进行 200 次滤波实验，统计滤波后信噪比发现，对于噪声强度在 0.1 及其以下的情况，91% 的概率可以实现图像恢复，而噪声强度在 0.2 及其以下情况下则无法实现图像恢复。

#### 4.4 模型评价

文中分析讨论了过采样率和相关先验信息如何影响图像的恢复概率，并通过控制变量法进行了大量仿真实验。本文中的实验结果表明，过采样率越大，信噪比降低，当过采样率为 2 时，对于某些噪声可以完全实现图像恢复；以随机噪



**图 18 信噪比随噪声强度变化曲线**

声强度分析了先验信息对恢复概率的影响，当噪声强度小于 0.1 情况下，存在滤波器可以大概率恢复图像。为便于分析，文中考虑因素比较简单，然而实际情况十分复杂，本文所作分析具有参考意义，但仍存在局限性，值得进一步研究。

## 5. 模型评价与改进

文中针对图像含单一噪声频谱数据，设计了低通滤波算法，该算法模型可以实现对含噪图像的去噪。实验结果表示通过低通滤波去噪算法，图像的信噪比得到提升，验证了算法的有效性。然而，不同滤波器参数的设计对滤波效果有明显影响，如何选取最佳滤波参数是一个值得进一步研究的问题，可以考虑启发式优化方法进一步确定滤波器参数。另外，高频信号中除噪声外仍有图像原始信息，低通滤波器在去噪的同时也去除了部分有效信息。针对该问题，进一步研究了高频低频融合降噪方法。基于频域滤波将图像的高频部分与低频部分分离处理，然后对高频部分进行噪声识别，并采用改进滑动中值滤波窗口对高频图像中噪声部分进行去除，保留有用信息，最后再将处理后的高频部分与低频部分融合，并恢复出原图像。通过对比实验表明该算法相对于其他算法具有明显优势，将其用于附件中所提供数据去噪，所得图像清晰度明显提高。然而，在滤波参数选择时多凭借经验，未能设计自适应参数寻优算法，模型仍有改进空间。

本文分析讨论了过采样率和相关先验信息如何影响图像的恢复概率，并通过控制变量法进行了大量仿真实验。本文中的实验结果表明，过采样率越大，信噪比降低，当过采样率为 2 时，对于某些噪声可以完全实现图像恢复；以随机噪声强度分析了先验信息对恢复概率的影响，当噪声强度小于 0.1 情况下，存在滤波器可以大概率恢复图像。为便于分析，文中考虑因素比较简单，然而实际情况十分复杂，本文所作分析具有参考意义，但仍存在局限性，值得进一步研究。

接下来，可以针对滤波器参数优化问题进一步对模型进行改进。

## 参考文献

- [1] 刘珩, SAR 图像去噪方法研究, 苏州大学.
- [2] 陈佳音, 改进的傅里叶变换在红外图像降噪中应用研究, 科技通报, 28(10):3, 2012.
- [3] 刘莉, 谭吉春, 分数域图像混合噪声盲复原方法, 国防科技大学学报, 33(3):5, 2011.
- [4] 侯艳丽, 一种有效的去除图像混合噪声的算法, 河南大学学报: 自然科学版, 41(2):4, 2011.
- [5] 管美静, 珠杰, 吴燕如, 图像去噪方法研究进展, 计算机时代(2), 2020.
- [6] 武英, 一种自适应图像去噪混合滤波方法, 计算机工程与应用, 46(7):4, 2010.
- [7] 杨辉, 唐建锋, 陆世专, 一种去除图像混合噪声的新方法, 衡阳师范学院学报, 31(6):5, 2010.
- [8] 袁文成, 杨德兴, 陈超, 图像混合噪声的一种组合滤波消除方法, 微处理机, 28(4):4, 2007.
- [9] 张芬, 基于多小波变换的图像去噪方法研究, 成都理工大学, 2011.
- [10] 邹洁, 谱域光学相干层析成像技术中信号和图像处理的研究, 南京航空航天大学.

## 附录 A 过采样率和先验信息的图像恢复影响对比图

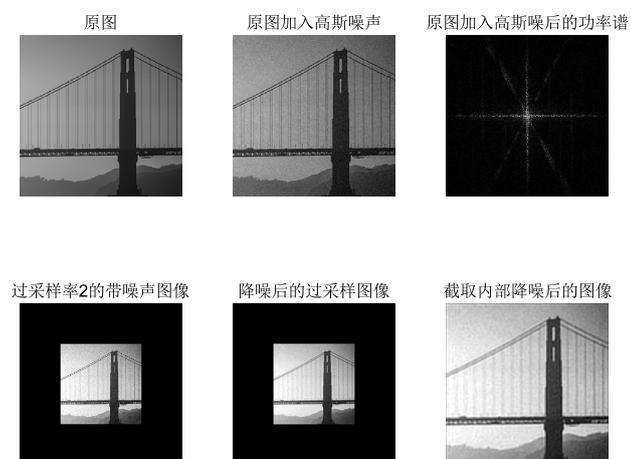


图 19 过采样率 2 的滤波效果

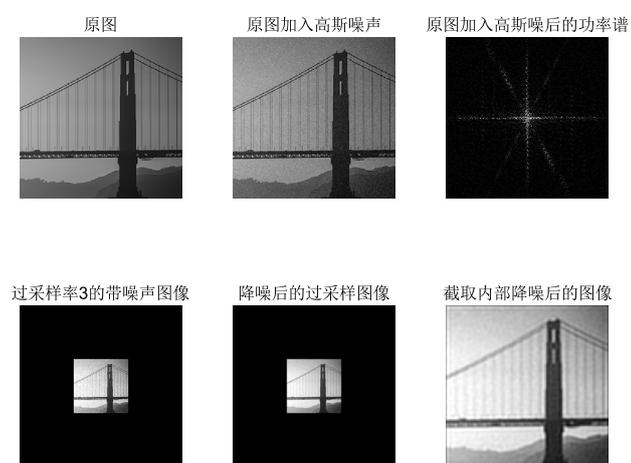


图 20 过采样率 3 的滤波效果

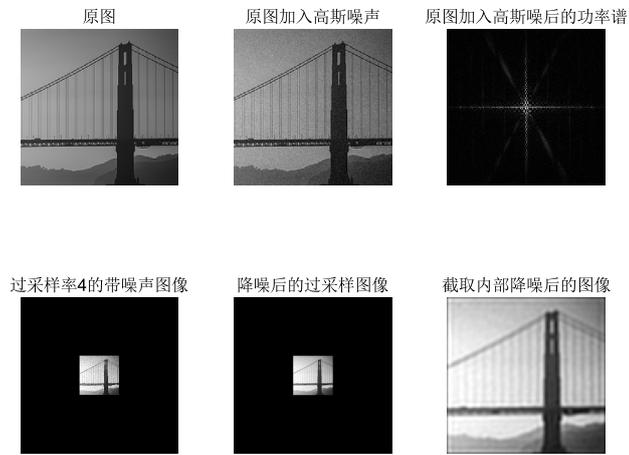


图 21 过采样率 4 的滤波效果

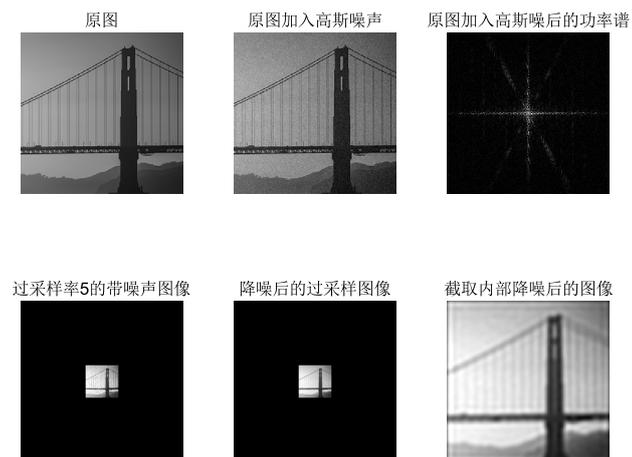


图 22 过采样率 5 的滤波效果

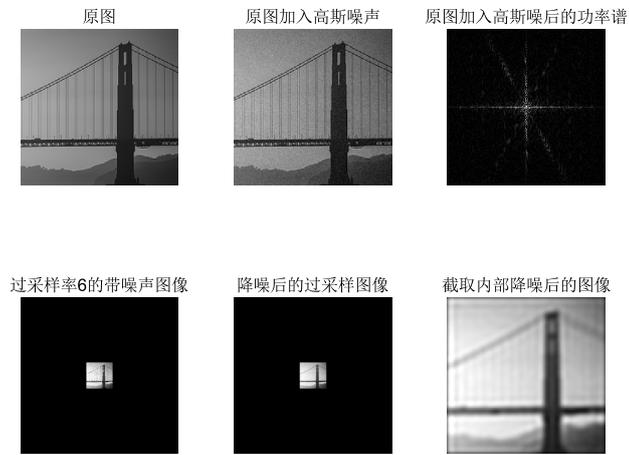


图 23 过采样率 6 的滤波效果

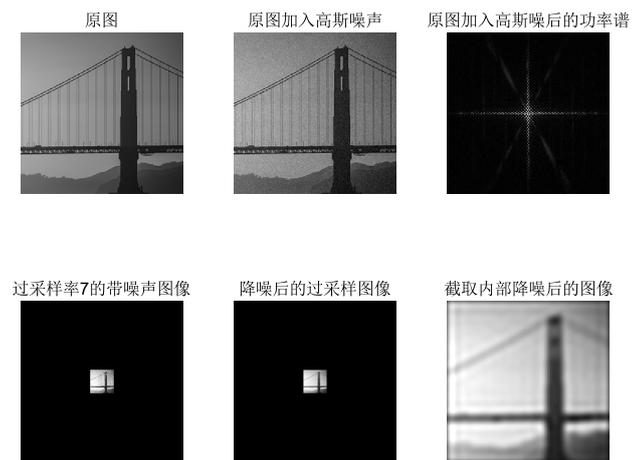


图 24 过采样率 7 的滤波效果

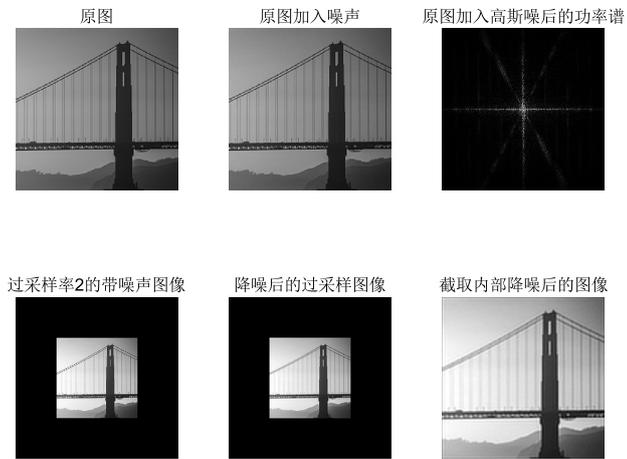


图 25 噪声强度 0.01 的滤波效果

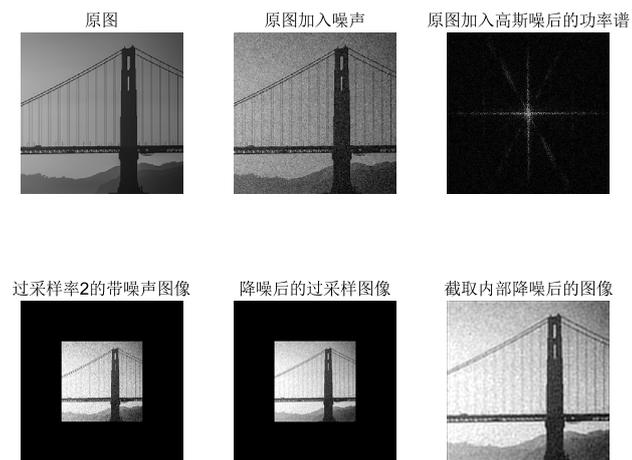


图 26 噪声强度 0.1 的滤波效果

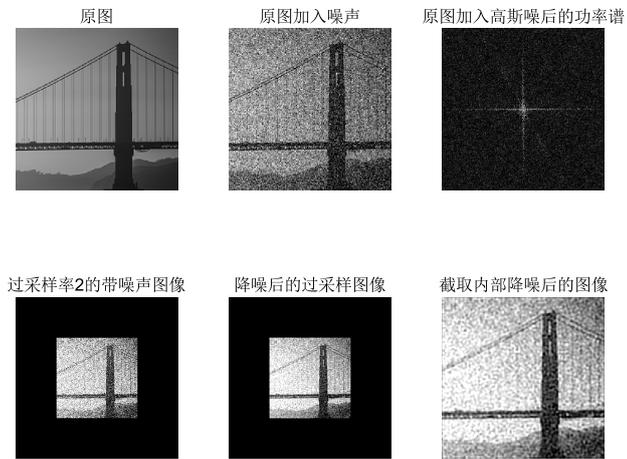


图 27 噪声强度 0.3 的滤波效果

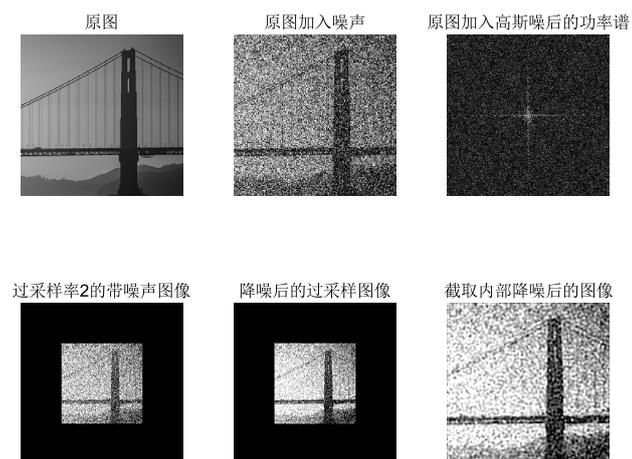


图 28 噪声强度 0.5 的滤波效果

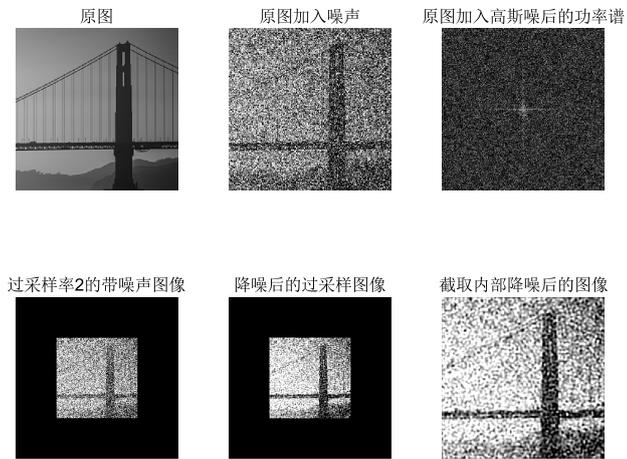


图 29 噪声强度 0.7 的滤波效果

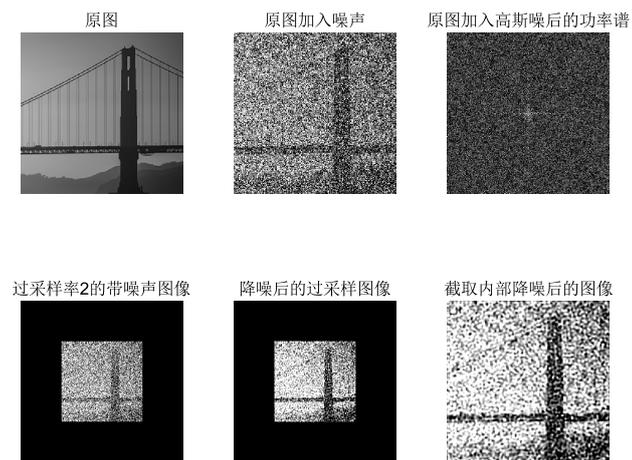


图 30 噪声强度 0.9 的滤波效果

## 附录 B MATLAB 源程序

### 2.1 问题 1 主要程序

```
clear;
clc;

sigma = 0.05; % noise variance
ir=imread('test256.jpg');
xiangsu=im2double(ir);
randn('seed', 0)
xiangsu = xiangsu + sigma*randn(size(xiangsu));

expand=zeros(512,512);% 放置像素图像中心
for i=1:256
    for j=1:256
        expand(128+i,128+j)=xiangsu(i,j);
    end
end

j=fftshift(fft2(expand));
[x,y]=meshgrid(-256:255,-256:255);%产生离散数据
z=sqrt(x.^2+y.^2);
d1=50;d2=100;%滤波器截止频率
n=6;%滤波器的阶数
h1=1./(1+(z/d1).^(2*n));
h2=1./(1+(z/d2).^(2*n));
k1=j.*h1;
k2=j.*h2;
l1=ifft2(ifftshift(k1));
l2=ifft2(ifftshift(k2));
figure(1)
subplot(222);imshow(expand); title('Noise Image')
subplot(221);imshow(rgb2gray(ir));title('original')
subplot(223);imshow(real(l1));title('10Hz')
subplot(224);imshow(real(l2));title('60Hz')
figure(2)
```

```

imshow(uint8(j));title('功率谱')
figure(11)
imshow(expand,'border','tight'); title('Noise Image')
figure(12)
imshow(rgb2gray(ir),'border','tight');title('original')
figure(13)
Iold = im2uint8(11);
for i=1:256
    for j=1:256
        I1(i,j)=Iold(128+i,128+j);%滤波后的图并且裁剪了的图
    end
end
imshow(I1,'border','tight');title('10Hz')
figure(14)
Iold = im2uint8(12);
for i=1:256
    for j=1:256
        I2(i,j)=Iold(128+i,128+j);%滤波后的图并且裁剪了的图
    end
end
imshow(I2,'border','tight');title('100Hz')

xiangsu=rgb2gray(xiangsu);
In=im2uint8(xiangsu);%带高斯噪声的216X216的图

ir=imread('test256.jpg');
xiangsu=im2double(ir);
xiangsu=rgb2gray(xiangsu);
Ir=im2uint8(xiangsu);%高清图

snr1=SNR1(Ir,In)%信噪比函数,Ir不用动,高斯噪声下信噪比
snr2=SNR1(Ir,I1)%信噪比函数,Ir不用动,滤波后下信噪比
snr2=SNR1(Ir,I2)%信噪比函数,Ir不用动,滤波后下信噪比

```

## 2.2 问题 2 主要程序

```

clear;
clc;

```

```

sigma = 0.05; % noise variance
ir=imread('test256.jpg');
ir=rgb2gray(ir);
xiangsu=im2double(ir);
randn('seed', 0)

% xiangsu = imnoise(xiangsu,'salt & pepper',0.02);%椒盐噪声
xiangsu = imnoise(xiangsu,'gaussian',0.01); %添加高斯噪声
% xiangsu = imnoise(xiangsu,'poisson');%加入泊松噪声
expand=zeros(512,512);% 放置像素图像中心
for i=1:256
    for j=1:256
        expand(128+i,128+j)=xiangsu(i,j);
    end
end

j=fftshift(fft2(expand));

[x,y]=meshgrid(-256:255,-256:255);%产生离散数据
z=sqrt(x.^2+y.^2);
d1=40;d2=40;%滤波器截止频率
n=6;%滤波器的阶数
h1=1./(1+(z/d1).^(2*n));
h2=1./(1+(z/d2).^(2*n));
k1=j.*h1;
k2=j.*(1-h2);
% S=sum(sum((real(j)))) %截断频率选择
% Sh=sum(sum((real(k2))))
% rate=abs(Sh)/abs(S)
l1=ifft2(ifftshift(k1));
l2=ifft2(ifftshift(k2));
% figure(1)
% subplot(322);imshow(expand); title('Noise Image')
% subplot(321);imshow(ir);title('original')
% subplot(323);imshow(uint8(k1));title('10HZ滤波后频谱图');
% subplot(324);imshow(uint8(k2));title('200HZ滤波后频谱图');
% subplot(325);imshow(real(l1));title('10Hz')

```

```

% subplot(326);imshow(real(l2));title('60Hz')
figure(11)
imshow(expand,'border','tight'); title('加高斯分布噪声图像')
figure(12)
imshow(uint8(k1),'border','tight');title('40Hz低通滤波后频谱图');
figure(13)
imshow(uint8(k2),'border','tight');title('40Hz高通滤波后频谱图');
figure(14)
imshow(real(l1),'border','tight');title('40Hz低通滤波后图像')
figure(15)
imshow(real(l2),'border','tight');title('40Hz低通滤波后图像')
figure(2)
imshow(uint8(j),'border','tight');title('完整频谱图')

figure(3)
subplot(121);imshow(l2);
title('高频原图');
figure(31)
imshow(l2,'border','tight');
title('高频原图');
[m, n] = size(l2); %m表示行数(即高度); n表示列数(即宽度)
x1 = im2uint8(l2);

%% -----%%

for i = 2:m-2+1
    for j = 2:n-2+1
        d1=abs(x1(i,j)-x1(i-1,j+1));
        d2=abs(x1(i,j)-x1(i,j+1));
        d3=abs(x1(i,j)-x1(i+1,j+1));
        d4=abs(x1(i,j)-x1(i-1,j));
        d5=abs(x1(i,j)-x1(i+1,j));
        d6=abs(x1(i,j)-x1(i-1,j-1));
        d7=abs(x1(i,j)-x1(i,j-1));
        d8=abs(x1(i,j)-x1(i+1,j-1));
        if max([d1 d2 d3 d4 d5 d6 d7 d8])>=60
            x1(i,j) =
                (x1(i-1,j+1)+x1(i,j+1)+x1(i+1,j+1)+x1(i-1,j)+x1(i+1,j)+x1(i-1,j-1)+x
        end
    end
end

```

```

    end
end
dstImage = uint8(x1);
figure(3)
subplot(122);imshow(dstImage);
title('高频中值去噪效果图');
figure(32)
imshow(dstImage,'border','tight');
title('高频中值去噪效果图');

tk2=fftshift(fft2(im2double(x1)));%高频的转频率图
    可以将x1改为12测试，还能回归到空洞
figure(4)
subplot(121);imshow(uint8(tk2));title('高频中值去噪后的功率谱')
subplot(122);imshow(uint8(k1+tk2));title('高低频融合后的功率谱')
figure(41)
imshow(uint8(tk2),'border','tight');title('功率谱')
figure(42)
imshow(uint8(k1+tk2),'border','tight');title('功率谱')
%再进行一次复原
j2=ifft2(ifftshift(k1+tk2));
figure(5)
%subplot(322);
Iold = im2uint8(j2);
for i=1:256
    for j=1:256
        I(i,j)=Iold(128+i,128+j);%滤波后的图并且裁剪了的图
    end
end
end
imshow(I,'border','tight'); %title('高频去噪后复原')

```

### 2.3 问题 3 主要程序

```

clear;
clc;

sigma = 0.01; % noise variance 0.05
ir=imread('test256.jpg');
xiangsu=im2double(ir);

```

```

%xiangsu=rgb2gray(ir);% 获得灰度图
M=size(xiangsu,1);% 原图分辨率 (大小256)
R=2;% 过采样率
N=R*M;% 过采样后图像大小, 如512
start_change=N/2-M/2;
%% 加入噪声
randn('seed', 0)
xiangsu = xiangsu + sigma*randn(size(xiangsu));
%% 过采样
expand=zeros(N,N);% 放置像素图像中心
for i=1:M
    for j=1:M
        expand(start_change+i,start_change+j)=xiangsu(i,j);
    end
end

j=fftshift(fft2(expand));
[x,y]=meshgrid(-N/2:N/2-1,-N/2:N/2-1);%产生离散数据
z=sqrt(x.^2+y.^2);
d1=10;d2=100;%滤波器截止频率
n=6;%滤波器的阶数
h1=1./(1+(z/d1).^(2*n));
h2=1./(1+(z/d2).^(2*n));
k1=j.*h1;
k2=j.*h2;
l1=ifft2(ifftshift(k1));
l2=ifft2(ifftshift(k2));
figure(1)
subplot(222);imshow(expand); title('Noise Image')
subplot(221);imshow(ir);title('original')
subplot(223);imshow(real(l1));title('10Hz')
subplot(224);imshow(real(l2));title('60Hz')
figure(2)
imshow(uint8(j));title('功率谱')
figure(11)
subplot(233);imshow(uint8(j));title('原图加入高斯噪后的功率谱')

Iold = im2uint8(l2);

```

```

for i=1:256
    for j=1:256
        I(i,j)=Iold(start_change+i,start_change+j);%滤波后的图并且裁剪了的图
    end
end

xiangsu=rgb2gray(xiangsu);
In=im2uint8(xiangsu);%带高斯噪声的216X216的图

ir=imread('test256.jpg');
xiangsu=im2double(ir);
xiangsu=rgb2gray(xiangsu);
Ir=im2uint8(xiangsu);%高清图

snr1=SNR1(Ir,In)%信噪比函数,Ir不用动,高斯噪声下信噪比
snr2=SNR1(Ir,I)%信噪比函数,Ir不用动,滤波后下信噪比
figure(3)
subplot(131);imshow(Ir,'border','tight');title('原图')
subplot(132);imshow(In,'border','tight');title('高斯噪声')
subplot(133);imshow(I,'border','tight');title('降噪后')
figure(11)
subplot(231);imshow(Ir,'border','tight');title('原图')
subplot(232);imshow(In,'border','tight');title('原图加入噪声')
subplot(234);imshow(expand,'border','tight');
    title('过采样率2的带噪声图像')
subplot(235);imshow(real(l2),'border','tight');title('降噪后的过采样图像')
subplot(236);imshow(I,'border','tight');title('截取内部降噪后的图像')
figure(12)
plot([2 3 4 5 6 7],[22.7686 17.4257 15.7840 15.2504 14.8166
    14.4174],'-o')
hold on
plot([2 3 4 5 6 7],[5.7602 5.7704 5.8113 5.7541 5.7675
    5.7598],'-*')
legend('滤波后','滤波前')
xlabel('过采样率')
ylabel('信噪比')

figure(13)
plot(0.1*[0.1 1 3 5 7 9],[23.5460 20.6274 12.6182 8.7230 6.0814

```

```

    4.5062], '-o')
hold on
plot(0.1*[0.1 1 3 5 7 9],[19.1105 2.5097 0.6430 0.3241 0.1893
    0.1147], '-*')
legend('滤波后','滤波前')
xlabel('随机信号强度')
ylabel('信噪比')

```

## 2.4 信噪比计算函数

```

function snr=SNR1(I,In)

[row,col,nchannel]=size(I);
snr=0;
if nchannel==1%gray image
Ps=sum(sum((I-mean(mean(I))).^2));%signal power
Pn=sum(sum((I-In).^2));%noise power
snr=10*log10(Ps/Pn);
elseif nchannel==3%color image
for i=1:3
    Ps=sum(sum((I(:, :, i)-mean(mean(I(:, :, i))))).^2));%signal
        power
    Pn=sum(sum((I(:, :, i)-In(:, :, i)).^2));%noise power
    snr=snr+10*log10(Ps/Pn);
end
snr=snr/3;
end
end
end

```