# 作业二

## 一、sobel滤波

#### 1.****图像预处理****：

图像首先被cv2.cvtColor()转换为灰度图，这是因为灰度图像上的颜色信息通常是在灰度图上进行的。彩色图像中的颜色信息对于边缘检测并不重要，因此我们将图像转换为单通道（灰度）图像。

#### 2.sobel****滤波****：

使用cv2.Sobel()函数计算在x和y方向上的半径。Sobel算子是常用的边缘检测算子，它过宽幅操作对图像图像进行辅助，计算每个像素点在水平方向和垂直方向上的变化。

* sobel\_x：在水平上进行边缘检测。
* sobel\_y：在垂直方向上进行边缘检测。

#### 3.****面积宽度计算****：

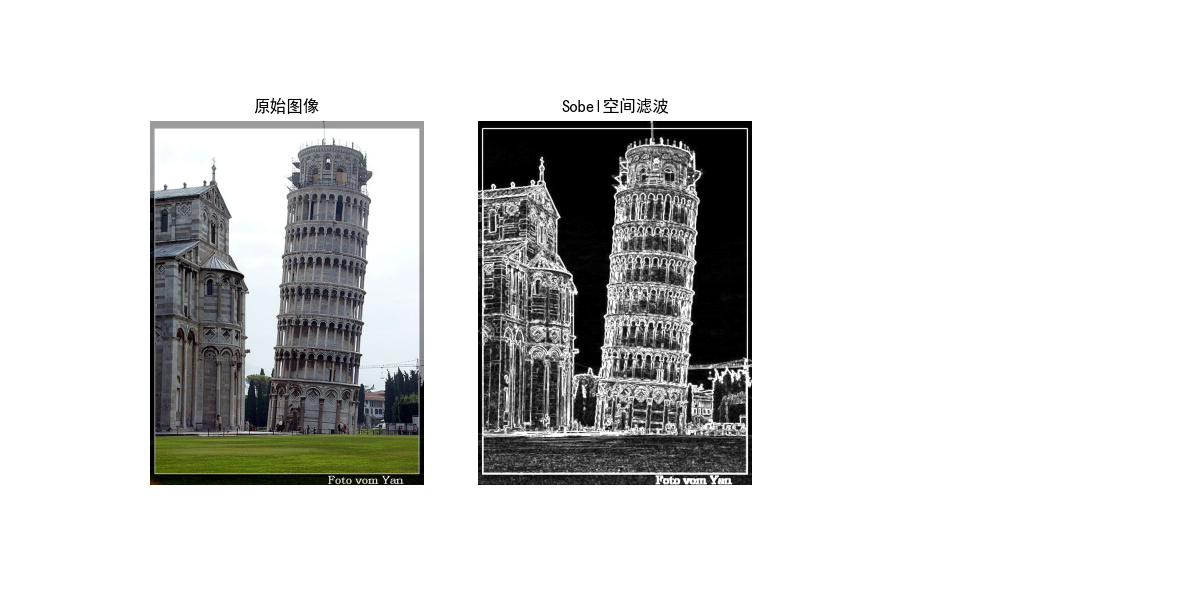
通过cv2.magnitude()计算每个像素点的梯度宽度值，它结合了x方向和y方向的梯度信息，表示该像素点的梯度信息边缘强度。

#### 4.****结果处理****：

将计算得到的最小幅值结果限制在0到255的范围内，将其转换为8位无符号整数，适合显示和保存。

该方法通过Sobel促进提取图像中的边缘信息，适用于很多计算机任务，如图像分割、特征检测等。在实际应用中，您可以进一步调整滤波器的大小或方向，以适应不同的需求。

#### 5.结果



#### 6.源代码：

import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
  
# Sobel空间域滤波函数  
def sobel\_space\_filter(image):  
 # 将图像转换为灰度图（如果图像已经是灰度图则跳过）  
 gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  
  
 # Sobel算子分别在x和y方向上进行卷积  
 sobel\_x = cv2.Sobel(gray, cv2.CV\_64F, 1, 0, ksize=3) # x方向梯度  
 sobel\_y = cv2.Sobel(gray, cv2.CV\_64F, 0, 1, ksize=3) # y方向梯度  
  
 # 计算梯度幅值（即边缘强度）  
 sobel\_edges = cv2.magnitude(sobel\_x, sobel\_y)  
  
 # 将结果限制在0到255之间，并转换为8位无符号整数  
 sobel\_edges = np.uint8(np.clip(sobel\_edges, 0, 255))  
  
 return sobel\_edges  
  
  
def main():  
 # 读取图像（请替换为您图像的路径）  
 img = cv2.imread('./建筑物.jpg')  
  
 # 使用Sobel空间域滤波进行边缘检测  
 sobel\_result = sobel\_space\_filter(img)  
  
 # 显示原始图像和处理后的图像  
 plt.figure(figsize=(12, 6))  
 # 设置支持中文的字体  
 plt.rcParams['font.sans-serif'] = ['SimHei'] # 使用黑体  
 plt.rcParams['axes.unicode\_minus'] = False # 解决负号显示问题  
 # 原始图像  
 plt.subplot(1, 3, 1)  
 plt.imshow(cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)) # 将BGR转为RGB以便Matplotlib显示  
 plt.title('原始图像')  
 plt.axis('off') # 关闭坐标轴  
  
 # Sobel空间域滤波结果  
 plt.subplot(1, 3, 2)  
 plt.imshow(sobel\_result, cmap='gray') # 使用灰度图显示  
 plt.title('Sobel空间滤波')  
 plt.axis('off') # 关闭坐标轴  
  
 # 显示所有图片  
 plt.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

## 高通滤波

#### 1.傅里叶变换

使用 np.fft.fft2() 将原图转换为频域。傅里叶变换能够将图像的空间信息转换为频率信息，其中低频部分代表图像的平滑区域，而高频部分则代表图像的边缘和细节。

#### ****2.频域高通滤波****：

* np.fft.fftshift()：将傅里叶变换结果的低频部分移动到频域图像的中心，以便查看。
* 通过将频域图像中心的小区域（低频部分）置为0，达到高通滤波的效果，即过滤掉图像的低频部分，只保留高频部分。这有助于去除图像的平滑区域，保留边缘和细节。

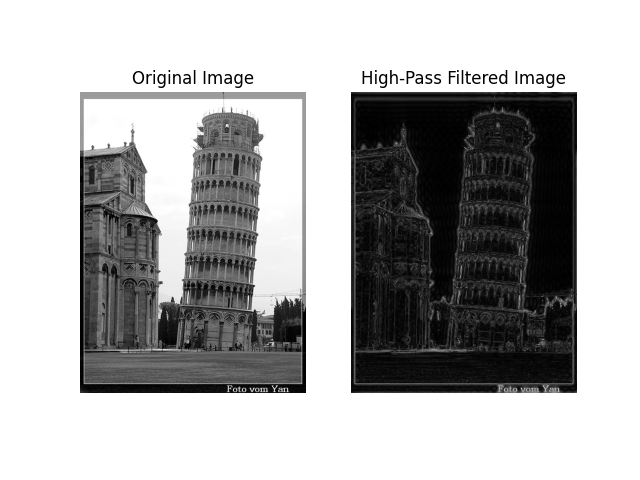
#### ****3.逆傅里叶变换****：

* np.fft.ifftshift()：将频域图像的低频部分移回左上角。
* np.fft.ifft2()：对处理后的频域图像进行逆傅里叶变换，得到过滤后的空间域图像。

#### 4.结果分析：

* **原始图像**：原始建筑物图像应该包含了图像的所有频率信息，包括低频和高频。
* **高通滤波后图像**：经过高通滤波后，图像中的低频部分（如平滑区域）被去除，剩下的主要是图像的边缘、纹理和细节。这通常会让图像看起来更加“锐利”，高频成分的突变更加明显。

#### 5.图像



#### 6.源代码

import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
  
def main():  
 # 读取图像，直接以灰度模式读取（0表示灰度模式）  
 img = cv2.imread("./建筑物.jpg", 0)  
  
 # 执行傅里叶变换，将图像从空间域转换到频域  
 f = np.fft.fft2(img)  
  
 # 为了便于观察频域，低频部分默认位于左上角，使用fftshift将低频信息移动到图像中心  
 fshift = np.fft.fftshift(f)  
  
 # 获取图像的行数和列数  
 rows, cols = img.shape  
 crows, ccols = int(rows / 2), int(cols / 2)  
  
 # 创建高通滤波器，通过将频域的低频部分设为0，达到去除低频信息的效果  
 fshift[crows - 30:crows + 30, ccols - 30:ccols + 30] = 0 # 设置中心区域为0，即去除低频  
  
 # 将频域的低频部分移回左上角，准备做逆傅里叶变换  
 ifshift = np.fft.ifftshift(fshift)  
  
 # 执行逆傅里叶变换，将频域信号转换回空间域  
 iimg = np.fft.ifft2(ifshift)  
  
 # 获取逆傅里叶变换的绝对值结果，因为傅里叶变换返回的是复数，取其幅值  
 iimg = np.abs(iimg)  
  
 # 绘制原始图像和经过高通滤波后的结果  
 plt.subplot(121)  
 plt.imshow(img, cmap='gray') # 显示原始图像  
 plt.title('Original Image') # 标题  
 plt.axis('off') # 关闭坐标轴  
  
 # 显示高通滤波后的图像  
 plt.subplot(122)  
 plt.imshow(iimg, cmap='gray') # 显示处理后的图像  
 plt.title('High-Pass Filtered Image') # 标题  
 plt.axis('off') # 关闭坐标轴  
  
 plt.show() # 显示图片  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main() # 调用主函数，开始执行

## 结果比较

|  | **Sobel空间域滤波** | **频域高通滤波** |
| --- | --- | --- |
| **工作原理** | 在空间域进行卷积，基于局部梯度计算图像的边缘和变化 | 在频域进行操作，通过傅里叶变换去除低频信息，保留高频信息 |
| **处理对象** | 主要处理图像的边缘和细节，突出梯度变化大的部分 | 去除图像中的低频成分，保留细节和高频部分 |
| **结果** | 增强边缘，图像中平滑区域变得模糊，突出局部变化 | 背景变模糊或消失，细节和边缘部分变得更加突出 |
| **适用场景** | 边缘检测，提取图像中强烈变化的区域 | 图像锐化，特征提取，突出细节和边缘部分 |
| **优点** | 简单高效，容易实现，适用于边缘检测 | 可以处理大尺度变化，适合图像锐化和细节增强 |
| **缺点** | 对噪声敏感，可能会增强噪声 | 背景区域被去除或模糊，可能导致图像的失真 |

#### ****Sobel滤波****

应用Sobel滤波时，图像的边缘更加明显，且对于建筑物这样的结构化图像，边缘部分如建筑轮廓等变得更加突出。

该方法比较适用于对边缘进行检测和提取，但对于大面积的背景部分，可能会出现较大的模糊区域。图像中的边缘和纹理更加明显，但背景部分依然保留。

#### ****频域高通滤波****

经过高通滤波后，图像的低频部分（即背景）被去除，图像的细节和边缘被强化。在建筑物图像中，建筑物的轮廓和细节将变得更加明显，背景的平滑部分会被去除，导致图像看起来更“锐化”或“增强”。

整体效果会让图像中的背景部分消失，边缘部分更加突出，细节和噪声得以增强，适合做进一步的细节分析。