智能医学数字图像处理实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 19084129 | 班级 | 190841班 |
| 姓名 | 李奕澄 | 专业 | 生物医学工程 |
| 实验  名称 | 实验5：图像频域增强 | | |
| 实验  目的  和  内容 | **实验目的和要求：**  掌握频域滤波的常⽤滤波器原理和⽅法  **实验内容：**  利⽤Python程序实现频域滤波的常⽤滤波器，并分析不同截⽌频率对不同滤波器的频域增强效果 | | |
| 实验  结果  与  分析 | 低通滤波器滤除高频噪声，频域高通滤波器滤除低频噪声，相同类型的滤波器的截止频率不同，对图像的滤除效果也会不同图像经过低通滤波器，图像的高频分量滤掉了，图像变得平滑，而且不同的截止频率，出来的图像也不同，系数小的效果强 | | |
| 实验  代码 | import cv2  import numpy as np  from matplotlib import pyplot as plt  def LowPassFilter(img):  # 傅里叶变换  dft = cv2.dft(np.float32(img), flags=cv2.DFT\_COMPLEX\_OUTPUT)  fshift = np.fft.fftshift(dft)  # 设置低通滤波器  rows, cols = img.shape  crow, ccol = int(rows / 2), int(cols / 2) # 中心位置  # 设置掩膜  mask = np.zeros((rows, cols, 2), np.uint8)  mask[crow - 20:crow + 20, ccol - 20:ccol + 20] = 1  # 掩膜图像和频谱图像乘积  f = fshift \* mask    # 傅里叶逆变换  ishift = np.fft.ifftshift(f)  iimg = cv2.idft(ishift)  res = cv2.magnitude(iimg[:, :, 0], iimg[:, :, 1])  return res  # 理想高通滤波器  def HighPassFilter(img):  # 傅里叶变换  f = np.fft.fft2(img)  fshift = np.fft.fftshift(f)  # 设置高通滤波器  rows, cols = img.shape  crow, ccol = int(rows / 2), int(cols / 2)  fshift[crow - 2:crow + 2, ccol - 2:ccol + 2] = 0  # 傅里叶逆变换  ishift = np.fft.ifftshift(fshift)  iimg = np.fft.ifft2(ishift)  iimg = np.abs(iimg)  return iimg  # 巴特沃斯低通滤波器  def ButterworthLowPassFilter(image, d, n, s1):  f = np.fft.fft2(image)  fshift = np.fft.fftshift(f)  def make\_transform\_matrix(d):  transform\_matrix = np.zeros(image.shape)  center\_point = tuple(map(lambda x: (x - 1) / 2, s1.shape))  for i in range(transform\_matrix.shape[0]):  for j in range(transform\_matrix.shape[1]):  def cal\_distance(pa, pb):  from math import sqrt  dis = sqrt((pa[0] - pb[0]) \*\* 2 + (pa[1] - pb[1]) \*\* 2)  return dis  dis = cal\_distance(center\_point, (i, j))  transform\_matrix[i, j] = 1 / (1 + (dis / d) \*\* (2 \* n))  return transform\_matrix  d\_matrix = make\_transform\_matrix(d)  new\_img = np.abs(np.fft.ifft2(np.fft.ifftshift(fshift \* d\_matrix)))  return new\_img  # 巴特沃斯高通滤波器  def ButterworthHighPassFilter(image, d, n, s1):  f = np.fft.fft2(image)  fshift = np.fft.fftshift(f)  def make\_transform\_matrix(d):  transform\_matrix = np.zeros(image.shape)  center\_point = tuple(map(lambda x: (x - 1) / 2, s1.shape))  for i in range(transform\_matrix.shape[0]):  for j in range(transform\_matrix.shape[1]):  def cal\_distance(pa, pb):  from math import sqrt  dis = sqrt((pa[0] - pb[0]) \*\* 2 + (pa[1] - pb[1]) \*\* 2)  return dis  dis = cal\_distance(center\_point, (i, j))  transform\_matrix[i, j] = 1 / (1 + (d / dis) \*\* (2 \* n))  return transform\_matrix  d\_matrix = make\_transform\_matrix(d)  new\_img = np.abs(np.fft.ifft2(np.fft.ifftshift(fshift \* d\_matrix)))  return new\_img  #用来正常显示中文标签  plt.rcParams['font.sans-serif']=['SimHei']  if \_\_name\_\_ == '\_\_main\_\_':  img1 = cv2.imread('scenery.png', 0)  img2 = cv2.imread('scenery.png')  source = cv2.cvtColor(img2,cv2.COLOR\_BGR2RGB)  f = np.fft.fft2(img1)  fshift = np.fft.fftshift(f)  # 取绝对值后将复数变化为实数 # 取对数的目的是将数据变换到0~255  s1 = np.log(np.abs(fshift))  lfilter = LowPassFilter(img1)  hfilter = HighPassFilter(img1)  blfilter = ButterworthLowPassFilter(img1,20,1,s1)  bhfilter = ButterworthHighPassFilter(img1,2,1,s1)    plt.subplot(231), plt.imshow(source, 'gray'), plt.title('原始图像')  plt.axis('off')  plt.subplot(232), plt.imshow(img1, 'gray'), plt.title('灰度图像')  plt.axis('off')  plt.subplot(233), plt.imshow(lfilter, 'gray'), plt.title('低通滤波图像')  plt.axis('off')  plt.subplot(234), plt.imshow(hfilter, 'gray'), plt.title('高通滤波图像')  plt.axis('off')  plt.subplot(235), plt.imshow(blfilter, 'gray'), plt.title('巴特沃斯低通滤波图像')  plt.axis('off')  plt.subplot(236), plt.imshow(bhfilter, 'gray'), plt.title('巴特沃斯高通滤波图像')  plt.axis('off') | | |
| 成绩  评定 | 教师签名：  年 月 日 | | |