**任课教师：黄文清**

**《智能信息处理及应用》**

**（ 2020 -- 2021 学年 第一学期）**

**实**

**验**

**报**

**告**

**学号：2018329621262**

**2018329621277**

**姓名：刘恒玮**

**朱若晨**

**班级：智能科学与技术（2）班**

**实验：基于拉普拉斯金字塔的图像融合算法**

**一、实验目的**

1. 学习并掌握拉普拉斯金字塔的构建和重构原理。
2. 学习并掌握基于拉普拉斯金字塔的图像融合方法的基本原理。
3. 学习并掌握图像融合的统计特性评价指标的计算方法。
4. 能利用基于拉普拉斯金字塔的图像融合方法，融合同一静态场景中曝光程度不同的2张图像，以得到信息含量较高的图像。
5. 复习python语言，能熟练使用python语言编写程序。

**二、实验设备**

本次实验以小组形式进行。

实验环境：python语言。

实验设备：win10 Intel i7 2.21GHz。

1. **实验原理**

**高斯金字塔**(Gaussianpyramid): 用来向下采样，主要的图像金字塔。高斯金字塔是通过高斯平滑和亚采样获得一些列下采样图像，也就是说第K层高斯金字塔通过平滑、亚采样就可以获得K+1层高斯图像，高斯金字塔包含了一系列低通滤波器，其截至频率从上一层到下一层是以因子2逐渐增加，所以高斯金字塔可以跨越很大的频率范围。

**拉普拉斯金字塔**(Laplacianpyramid): 用来从金字塔低层图像重建上层未采样图像，在数字图像处理中也即是预测残差，可以对图像进行最大程度的还原，配合高斯金字塔一起使用。两者的简要区别：高斯金字塔用来向下降采样图像，5而拉普拉斯金字塔则用来从金字塔底层图像中向上采样重建一个图像。要从金字塔第i层生成第i+1层（我们表示第i+1层为G\_i+1），我们先要用高斯核对G\_1进行卷积，然后删除所有偶数行和偶数列。当然的是，新得到图像面积会变为源图像的四分之一。按上述过程对输入图像G\_0执行操作就可产生出整个金字塔。当图像向金字塔的上层移动时，尺寸和分辨率就降低。OpenCV中，从金字塔中上一级图像生成下一级图像的可以用PryDown。而通过PryUp将现有的图像在每个维度都放大两遍。图像金字塔中的向上和向下采样分别通过OpenCV函数 pyrUp 和 pyrDown 实现。

**对图像向上采样**：pyrUp函数  
**对图像向下采样**：pyrDown函数  
 这里的向下与向上采样，是对图像的尺寸而言的（和金字塔的方向相反），向上就是图像尺寸加倍，向下就是图像尺寸减半。而如果我们按上图中演示的金字塔方向来理解，金字塔向上图像其实在缩小，这样刚好是反过来了。但是，PryUp和PryDown不是互逆的，即PryUp不是降采样的逆操作。这种情况下，图像首先在每个维度上扩大为原来的两倍，新增的行以0填充。然后给指定的滤波器进行卷积（实际上是一个在每个维度都扩大为原来两倍的过滤器）去估计“丢失”像素的近似值。

PryDown是一个会丢失信息的函数。为了恢复原来更高的分辨率的图像，我们要获得由降采样操作丢失的信息，这些数据就和拉普拉斯金字塔有关系了。

图像融合的研究目的是将同一静态场景的多张图像加以综合，以得到信息含量较高的图像。

拉普拉斯金字塔构建的基本步骤是：

1）建立图像高斯金字塔

设源图像为*G*0，把*G*0当做高斯金字塔底层。构造高斯金字塔第*L*层图像*GL*的方法为：首先将第*L*-1层的图像*GL*-1和具有低通特性的5×5高斯核函数*ω*(*m*, *n*)进行卷积，再将卷积出来的结果进行隔行隔列降采样，即：

 (1)

其中，*N*是高斯金字塔的最顶层层号，*CL*为第*L*层图像的列数，*RL*是第*L*层图像的行数。

*G*0、*G*1、……、*GN*构成了一个图像高斯金字塔。*G*0为高斯金字塔的最底层，依次往上，*GN*为高斯金字塔的最顶层，金字塔的总层数为*N*+1。

2）由高斯金字塔建立图像的拉普拉斯金字塔

首先将*GL*利用*Expand*算子内插放大，使放大后的图像的尺寸与*GL*-1的尺寸大小相同。内插的灰度值可以用原来像素的灰度值加权平均得到的。由于*GL*是通过对*GL*-1卷积降采样得到的，所以所包含的信息是少于*GL*-1的，它们虽然尺寸相同，但是并不相等。

令，

 (2)

则，由*LP*0，*LP*1，...，*LPN*构成的金字塔即为拉普拉斯金字塔。

3）由拉普拉斯金字塔重建源图像

从拉普拉斯金字塔最顶层开始逐层由上至下，按照公式(3)恢复对应的高斯金字塔，最终得到源图像。

 (3)

**五、实验内容及步骤**

1. **读入图片，并对图片统一格式**

img\_left = cv2.imread('left.png',0)

img\_right = cv2.imread('right.png',0)

img\_size = (1024,1024)  #统一图片大小

1. **利用高斯金字塔向下采样。（左右同理）**

G = img\_left.copy()

guass\_pyramid\_left = [img\_left.copy()] #保存左聚焦的高斯金字塔图片

for i in np.arange(Series):     #将左聚焦进行高斯金字塔处理，总共六级处理

    G = cv2.pyrDown(G)

    guass\_pyramid\_left.append(G)

G = img\_right.copy()

guass\_pyramid\_right = [img\_right.copy()] #保存右聚焦的高斯金字塔图片

for i in np.arange(Series):  #将右聚焦进行高斯金字塔处理，总共六级处理

    G = cv2.pyrDown(G)

    guass\_pyramid\_right.append(G)

1. **利用拉普拉斯金字塔向上采样。（左右同理）**

laplace\_pyramid\_left = [guass\_pyramid\_left[Series-1]] #保存左聚焦的拉普拉斯金字塔图片

for i in np.arange(Series-1,0,-1):    #将左聚焦进行拉普拉斯金字塔处理，总共5级处理

    GE = cv2.pyrUp(guass\_pyramid\_left[i])

    L = cv2.subtract(guass\_pyramid\_left[i-1],GE)

    laplace\_pyramid\_left.append(L)

laplace\_pyramid\_right = [guass\_pyramid\_right[Series-1]]#保存右聚焦的拉普拉斯金字塔图片

for i in np.arange(Series-1,0,-1):    #将右聚焦进行拉普拉斯金字塔处理，总共5级处理

    GE = cv2.pyrUp(guass\_pyramid\_right[i])

    L = cv2.subtract(guass\_pyramid\_right[i-1],GE)#两张图片相减

    laplace\_pyramid\_right.append(L)

1. **最后进行重建实现图像融合**

LS = []

for la,lb in zip(laplace\_pyramid\_left,laplace\_pyramid\_right):

    rows,cols = la.shape

    la[:,:] = la[:,:]\*0.5

    lb[:,:] = lb[:,:]\*0.5

    ls = la+lb    #将两个图像的矩阵0.5 0.5的比例融合在一起

    LS.append(ls)

#重建过程

ls\_reconstruct = LS[0]   #这里LS[0]为高斯金字塔的最小图片

for i in range(1,Series):                        #第一次循环的图像为高斯金字塔的最小图片，依次通过拉普拉斯金字塔恢复到大图像

    ls\_reconstruct = cv2.pyrUp(ls\_reconstruct)

    ls\_reconstruct = cv2.add(ls\_reconstruct, LS[i]) #采用金字塔拼接方法的图像

real = cv2.addWeighted(img\_left,0.5,img\_right,0.5,1) #直接融合

1. **计算融合后图像的平均灰度梯度的函数**

def avgGradient(image):#计算平均灰度梯度

    width = image.shape[1]

    width = width - 1

    heigt = image.shape[0]

    heigt = heigt - 1

    tmp = 0.0

    for i in range(width):

        for j in range(heigt):

            dx = float(image[i, j + 1]) - float(image[i, j])

            dy = float(image[i + 1, j]) - float(image[i, j])

            ds = math.sqrt((dx \* dx + dy \* dy) / 2)

            tmp += ds

    imageAG = tmp / (width \* heigt)

    return imageAG

1. **输出各个融合结果的统计特性**

real\_mean = np.mean(real)#求均值

real\_var = np.var(real)#求方差

real\_std = np.std(real,ddof=1)#求标准差

real\_AG = avgGradient(real)#平均灰度梯度

real\_MI = MI(img\_left,img\_right,real)#互信息

ls\_reconstruct\_mean = np.mean(ls\_reconstruct)#求均值

ls\_reconstruct\_var = np.var(ls\_reconstruct)#求方差

ls\_reconstruct\_std = np.std(ls\_reconstruct,ddof=1)#求标准差

ls\_reconstruct\_AG = avgGradient(ls\_reconstruct)#平均灰度梯度

ls\_reconstruct\_MI = MI(img\_left,img\_right,ls\_reconstruct)#互信息

print("直接拼接均值为：%f" % real\_mean,"\n拉普拉斯金字塔融合均值为：%f\n" % ls\_reconstruct\_mean)

print("直接拼接方差为：%f" % real\_var,"\n拉普拉斯金字塔融合方差为：%f\n" % ls\_reconstruct\_var)

print("直接拼接标准差为:%f" % real\_std,"\n拉普拉斯金字塔融合标准差为:%f\n" % ls\_reconstruct\_std)

print("直接拼接标平均灰度梯度为:%f" % real\_AG,"\n拉普拉斯金字塔融合平均灰度梯度为:%f\n" % ls\_reconstruct\_AG)

print("直接拼接标互信息为:%f" % real\_MI,"\n拉普拉斯金字塔融合互信息为:%f\n" % ls\_reconstruct\_MI)

1. **展示直接拼接和拉普拉斯金字塔融合法的融合结果**

plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签

plt.subplot(221), plt.imshow(cv2.cvtColor(img\_left,cv2.COLOR\_BGR2RGB))

plt.title("左聚焦")

plt.subplot(222), plt.imshow(cv2.cvtColor(img\_right,cv2.COLOR\_BGR2RGB))

plt.title("右聚焦")

plt.subplot(223), plt.imshow(cv2.cvtColor(real,cv2.COLOR\_BGR2RGB))

plt.title("直接拼接")

plt.subplot(224), plt.imshow(cv2.cvtColor(ls\_reconstruct,cv2.COLOR\_BGR2RGB))

plt.title("laplace\_pyramid")

plt.show()

**六、实验结果**

**实验参数设置：**

img\_size = (1024,1024) #统一图片大小

Series = 6 #高斯金字塔处理级数

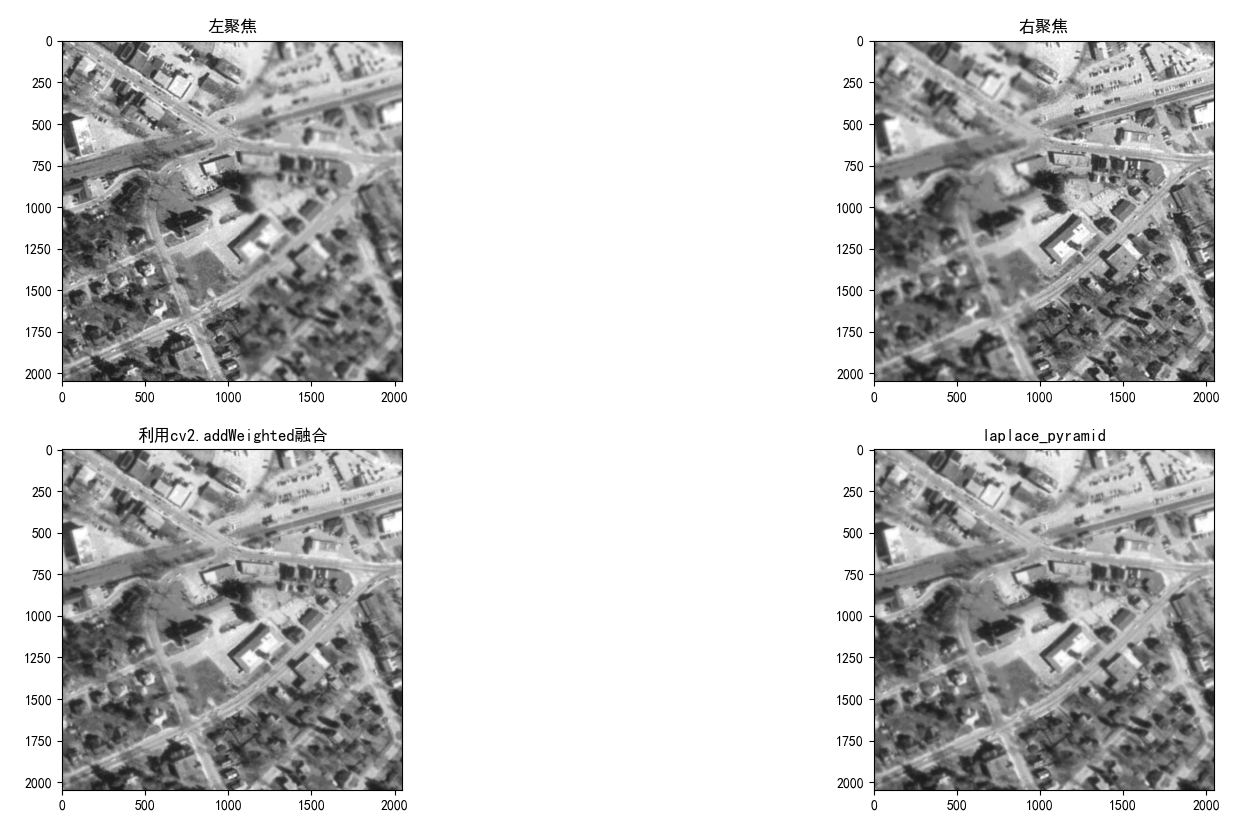
guass\_pyramid\_left = [img\_left.copy()] #保存左聚焦的高斯金字塔图片

guass\_pyramid\_right = [img\_right.copy()] #保存右聚焦的高斯金字塔图片

laplace\_pyramid\_left = [guass\_pyramid\_left[Series-1]] #保存左聚焦的拉普拉斯金字塔图片

laplace\_pyramid\_right = [guass\_pyramid\_right[Series-1]]#保存右聚焦的拉普拉斯金字塔图片

**实验结果截图：**



图一：直接拼接和拉普拉斯金字塔融合法结果



图二：融合结果的评价指标

**实验结果分析：**

拉普拉斯金字塔融合法的方差、标准差小于直接拼接法，反映出来拉普拉斯金字塔融合法融合效果较好，但是平均灰度值却也是小于直接拼接法，通常认为平均梯度越大，图像清晰度越好，融合质量越好，此实现结果与理论不符合，可能是拉普拉斯金字塔融合过程中的信息损失。拉普拉斯金字塔融合法的互信息值MI大于直接拼接法，反应了图像信息间的联系较大，说明拉普拉斯金字塔融合法融合效果较好。

**七、实验总结**

通过本次实验，我理解了拉普拉斯金字塔融合法算法的具体过程以及如何运用拉普拉斯金字塔融合法算法去实现图像融合的问题。本次实验也不是一帆风顺的，实验过程中出现了一些问题，通过思考、查阅资料和上机实践，有些问题得以解决，有些问题还存有疑问。

**问题1**：如果图像大小不是2的次方，则无法进行拉普拉斯金字塔融合法算法。

**解决方案：**通过运用opencv的resize函数重新设置图像大小，比如说实验过程中将图片大小重新设置为1024\*1024，即可解决此问题。

**问题2**：经过多层的拉普拉斯金字塔融合之后的图片，出现了失真模糊的情况，且金字塔的层数越大，失真和模糊的情况越严重。

**解决方案**：尚未解决

**参考资料：**

1. 《图像融合算法》实验报告指导书。
2. 《智能信息处理技术原理与应用》
3. 《人工智能》（第3版）