

图像融合

摘要

利用拉普拉斯金字塔混合图像

关键词：拉普拉斯金字塔 掩模 图像合成

涉及概念

RGB图像转化为灰度图像

灰度转换 $gray = R * 0.30 + G * 0.59 + B * 0.11$

注意到 `cv2.imread` 是以 *BGR* 的顺序存储

cv2 库函数

```
import cv2
cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
#src为转换对象
```

自行实现

```
def convertGray(self, img_rgb):
    #灰度转换的自定义实现
    img_gray = img_rgb[:, :, 0] * 0.11 + img_rgb[:, :, 1] * 0.59 +
img_rgb[:, :, 2] * 0.3
    ##只有当数组类型为uint8时，opencv才会认为这是图片
    img_gray = img_gray.astype(np.uint8)
    return img_gray
```

高斯金字塔

原图作为第0层高斯金字塔，第n+1层图像由第n层降采样得到

拉普拉斯金字塔

构造拉普拉斯金字塔的目的就是为了恢复高分辨率的图像。

第n层拉普拉斯金字塔由第n层高斯金字塔和第n+1层高斯金字塔的上采样做差得到。

$$\begin{aligned}L_i &= G_i - \text{PyrUp}(G_{i+1}) \\ G_i &= L_i + \text{PyrUp}(G_{i+1})\end{aligned}$$

所以对于一个 *n* 层的高斯金字塔，可以用 *n* - 1 层的拉普拉斯金字塔和第 *n* 层高斯金字塔来表示

降采样和上采样

降采样和上采样分别是得到高斯金字塔和拉普拉斯金字塔的关键

降采样

- 低通滤波器卷积处理(如高斯低通滤波器)
- 缩小(例如长宽缩小到原来的 $\frac{1}{2}$,可以去掉所有偶数行/列)

上采样

- 放大图像(例如长宽放大两倍,任意 4 个相邻像素中, 填一个原来的值, 其余为 0)
- 低通滤波器处理,卷积核乘以放大倍数的平方倍

实验过程

- 输入图像并转化为灰度图
- 生成图像和掩模的高斯金字塔和拉普拉斯金字塔
- 图像的拉普拉斯金字塔与对应掩模相乘, 得到的结果相加
- 用合成得到的拉普拉斯还原图像的高斯金字塔

代码概览

输入

```
'''输入图像'''
im1=imread('1.jpg')#掩模
im2=imread('2.png')
mask=imread('3.jpg')

'''将RGB转化为灰度图像'''
gray1=cvtColor(im1,cv2.COLOR_BGR2GRAY)
gray2=cvtColor(im2,cv2.COLOR_BGR2GRAY)
mask=cvtColor(mask,cv2.COLOR_BGR2GRAY)
opmask=bitwise_not(mask)#非运算
```

生成金字塔

```
def gengp(im,layer):
    '''给定图像和层数,生成高斯'''
    g=im.copy()
    gp=[g]
    for i in range(layer-1):
        g=pyrDown(g)
        gp.append(g)
    return gp

def genlp(gp):
    layer=len(gp)
    lp=[gp[-1]]
    for i in range(layer-1,0,-1):
        up=pyrUp(gp[i])
        src1,src2=sameSize(gp[i-1],up)#上采样可能shape有差异,sameSize函数负责统一大小
        l=subtract(src1,src2)
```

```
lp.append(l)  
return lp
```

混合金字塔

```
def masked(src1,src2):  
    '''图像金字塔与掩模金字塔相乘'''  
    layer=len(src1)  
    dst=[]  
    for i in range(layer):  
        r=bitwise_and(src1[i],src2[i])#与掩模做与运算  
        dst.append(r)  
    return dst  
def addingae(src1,src2):  
    '''图像金字塔相加'''  
    layer=len(src1)  
    dst=[]  
    for i in range(layer):  
        r=add(src1[i],src2[i])  
        dst.append(r)  
    return dst
```

还原图像

```
def lp2gp(lp):  
    '''由拉普拉斯金字塔得到高斯金字塔'''  
    gp=[lp[0]]  
    for i in range(len(lp)-1):  
        up=pyrUp(gp[i])  
        #print(pyrUp(lp[i]).shape,lp[i+1].shape)  
        src1,src2=sameSize(up,lp[i+1])  
        g=add(src1,src2)  
        gp.append(g)  
    return gp
```

实验结果

获得基于高斯金字塔层数的合成图像，但边缘融合效果并不理想。

总结与展望