

Seam Carving 压缩图片实验报告

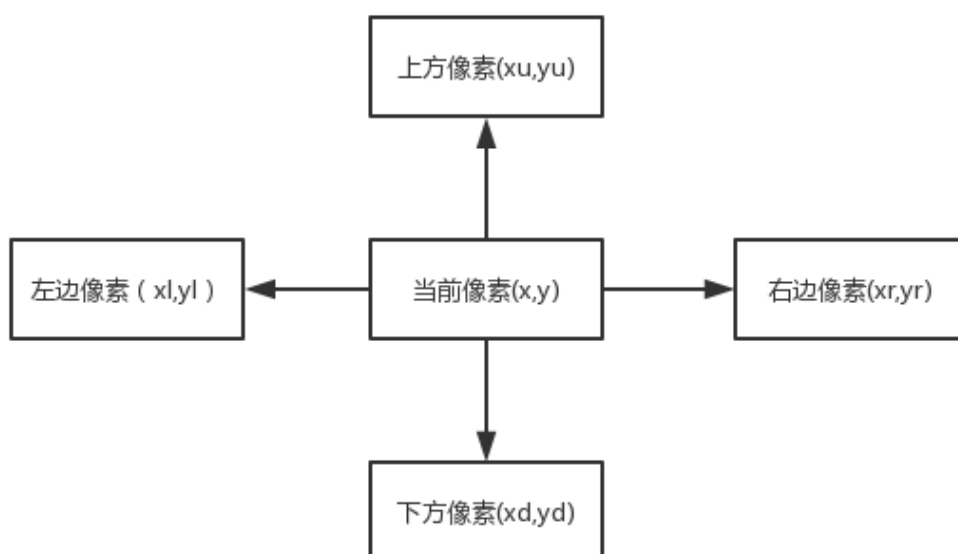
一、实验环境

操作系统: Windows10 专业版 64 位
处理器: Intel(R) Core(TM) i7-8700 @3.2GHZ
内存: 16G
编程语言: Java 1.8

二、算法描述

1. 计算破坏度的方法

计算每一个像素的上下和左右像素的 RGB 差值的平方和为破坏度。



左右平方和

$$E_{rl} = (x_{l_R} - x_{r_R})^2 + (x_{l_G} - x_{r_G})^2 + (x_{l_B} - x_{r_B})^2$$

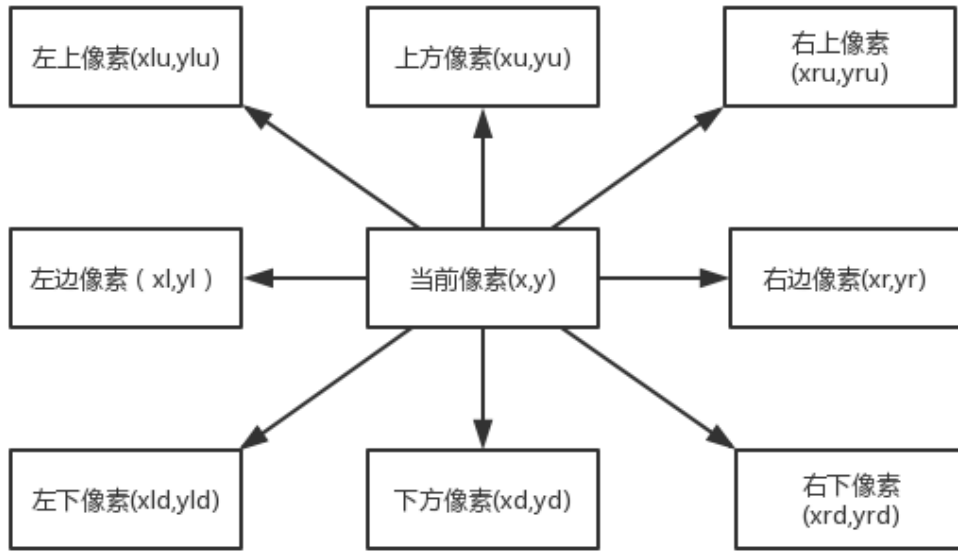
上下平方和

$$E_{ud} = (x_{u_R} - x_{d_R})^2 + (x_{u_G} - x_{d_G})^2 + (x_{u_B} - x_{d_B})^2$$

破坏度计算公式为:

$$E = E_{rl} + E_{ud}$$

另一种方法是考虑四周的 RGB 差值的平方和为破坏度。



左右平方和

$$E_{rl} = (x_{l_R} - x_{r_R})^2 + (x_{l_G} - x_{r_G})^2 + (x_{l_B} - x_{r_B})^2$$

上下平方和

$$E_{ud} = (x_{u_R} - x_{d_R})^2 + (x_{u_G} - x_{d_G})^2 + (x_{u_B} - x_{d_B})^2$$

左上右下对角线平方和为

$$E_l = (x_{lu_R} - x_{rd_R})^2 + (x_{lu_G} - x_{rd_G})^2 + (x_{lu_B} - x_{rd_B})^2$$

左下右上对角线平方和为

$$E_r = (x_{ld_R} - x_{ru_R})^2 + (x_{ld_G} - x_{ru_G})^2 + (x_{ld_B} - x_{ru_B})^2$$

破坏度计算公式为：

$$E = E_{rl} + E_{ud} + E_l + E_r$$

2. seam carving 算法描述

以按行进行 seam carving 为例：

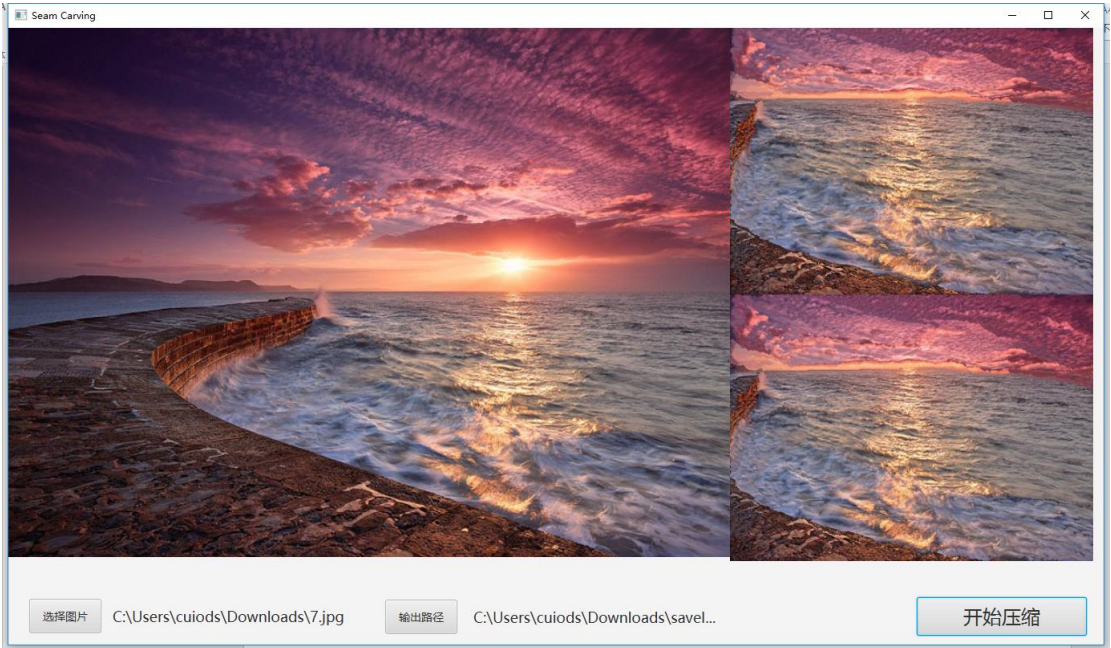
- (1) 计算每一个像素点的破坏度
- (2) 计算第一行每个像素的破坏度和，也就是每个像素的破坏度本身，标记第一行像素选取的都是自身。
- (3) 从第二行开始，计算每个像素的破坏度和的最小值，每个像素的破坏度和最小值是像素本身的破坏度与上一行同一列或相邻列破坏度之和的最小值。记录标记选取上一行哪一列的破坏和。
- (4) 找出最后一行破坏值之和最小的一个像素，根据标记向上查找删除的一条接缝。

伪代码描述为：

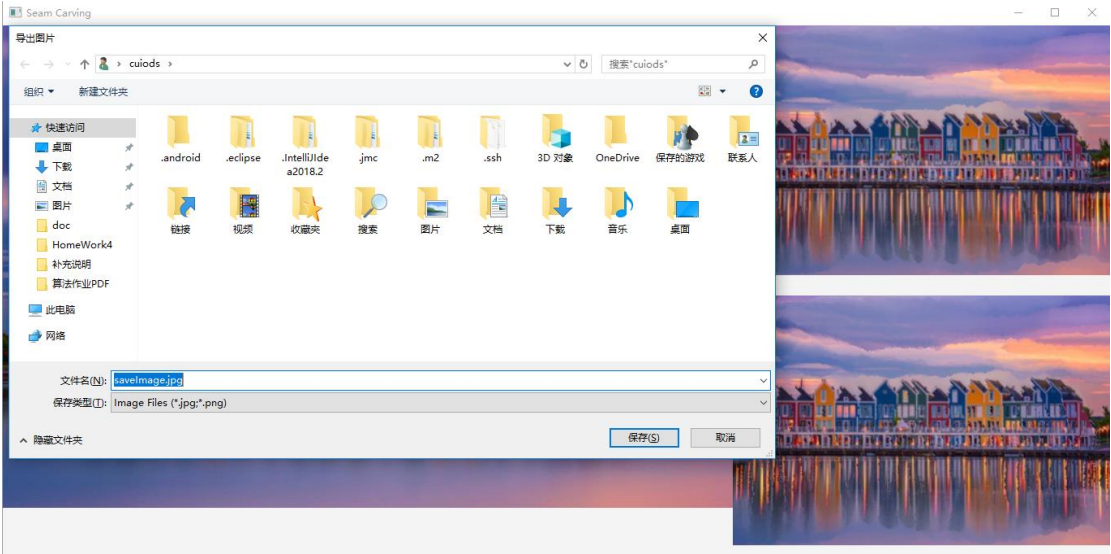
```
SEAM_CARVING (Pic, Energy, Result)
    for i = 0 to width
        last[i] = Energy[0][i]
        tags[0][i] = 0
    for i = 1 to height
        for j = 0 to width
            if j-1 >= 0 && j+1 <= width-1
                if last[j-1] < last[j]
                    current[j] = Energy[i][j] + last[j-1]
                    tags[i][j] = -1
                else if last[j+1] < last[j]
                    current[j] = Energy[i][j] + last[j+1]
                    tags[i][j] = 1
                else
                    current[j] = Energy[i][j] + last[j]
                    tags[i][j] = 0
            //讨论边界情况，有边界则不比较该边
            else if j-1 < 0
                .....
            else
                .....
        last = current

    minIndex = 0
    minValue = INFINITE
    for i = 0 to width
        if last[i] < minValue
            minValue = last[i]
            minIndex = i
    for i = 0 to height
        Result[i] = minIndex
        minIndex = minIndex + tags[i][minIndex]
    return Result//返回每一行需要删除的列坐标
```

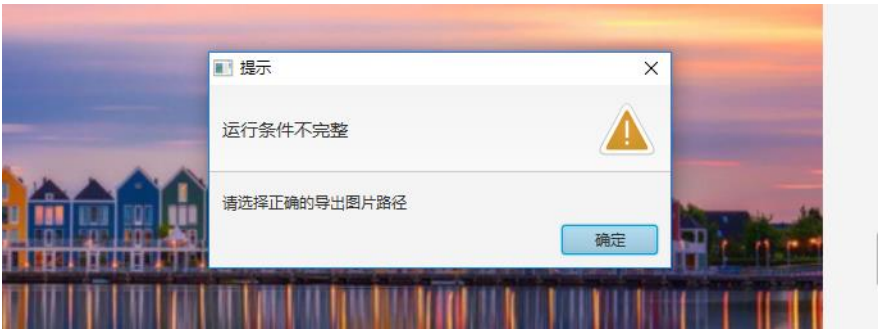
三、实验界面



在选择图片和设置输出路径后可以进行 Seam Carving 压缩。



设置路径不完整会产生错误提示。



四、实验结果

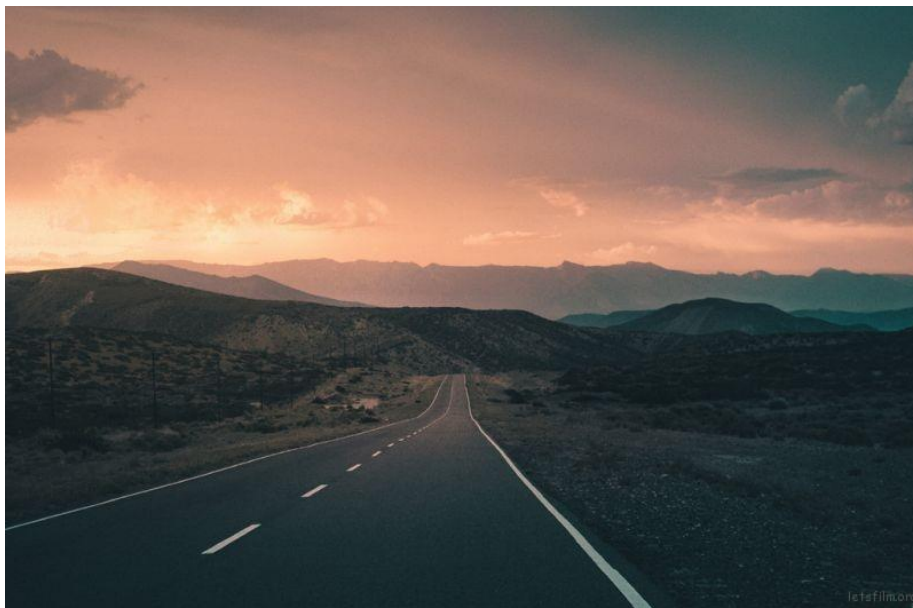
原图：



两种方法的比较：



原图

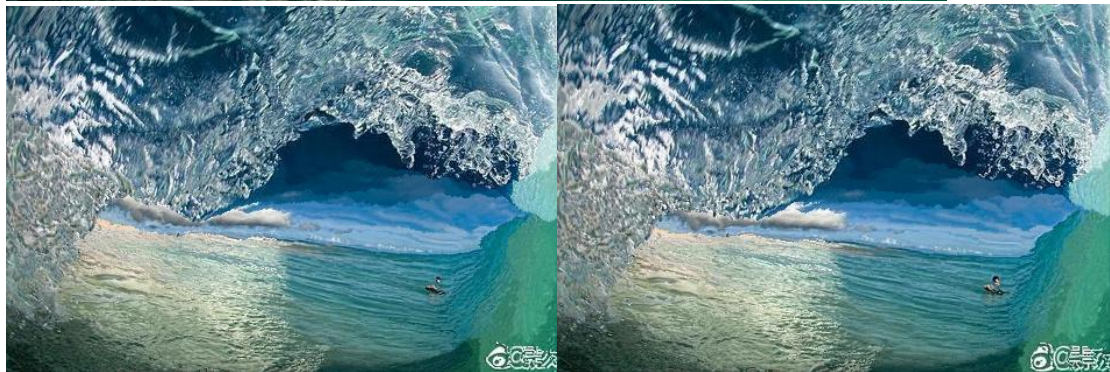




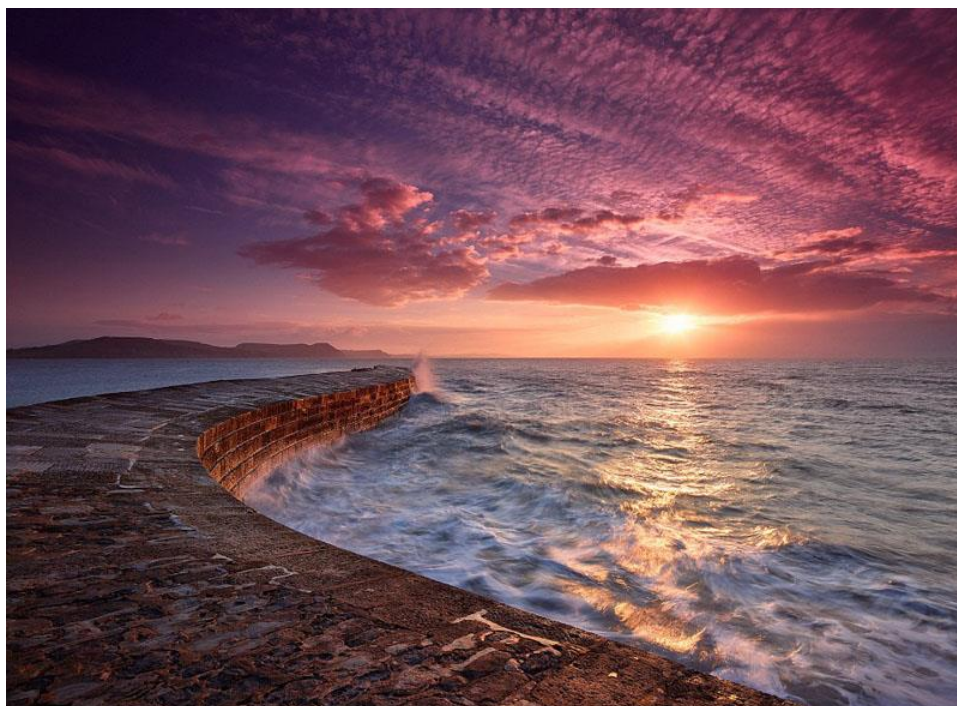
原图：



原图：



原图：





从多组实验的对比可以看出，Seam Carving 算法可以突出原图中的重点事物，但在细节的处理上还有很多问题。通过稍微改进的破坏度计算方式可以使转换后图像的线条更加平滑（右图），但需要耗费更多的时间。因此可以寻找其他更有效的计算破坏度的方式改进这样的算法。