


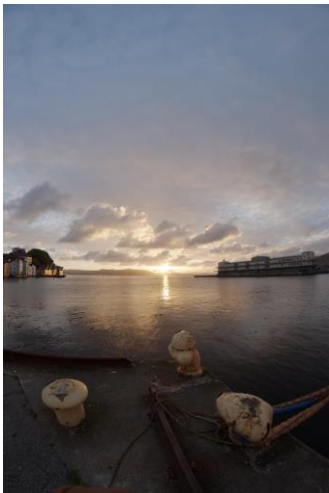

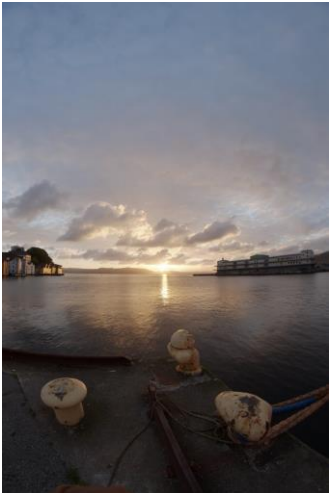





# 基于梯度的多曝光融合分析报告

本次作业采用两种方法对于不同曝光度图片进行融合，方法一是基于梯度的多曝光图像融合，方法二是基于梯度方向的多曝光融合。

**方法一：**对于静态图像可以实现较好的融合效果，可以将不同曝光度的图片的细节充分展示在一张图片中。在实验中采用了三种不同的算子进行梯度计算（sobel，laplacion，canny），其中 laplacion 算子的计算结果最令人满意，而传统的 sobel 算子在图像像素的边界处的处理效果不佳，将一些噪声也进行了放大。Canny 算子更适合对图像的边缘进行提取，用在此处效果不佳。以下是不同算子的结果对比图：

Sobel	Canny	Laplacion
		
		
		

可以明显看到拉普拉斯算子无论是在强曝光的位置还是在阴影处都有着更优的处理能力。

本次作业采用的高斯滤波器和中值滤波器均为自己实现，速度比 opencv 自带的函数慢 1~2 倍。但是处理效果要略优于 opencv 自带的函数模板



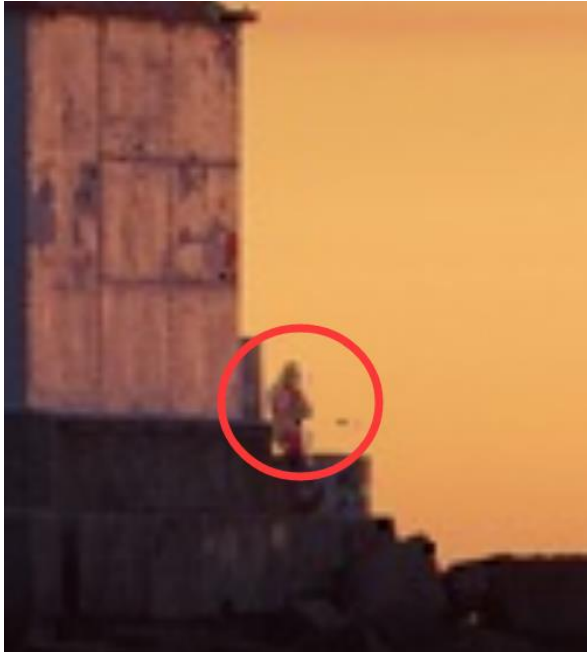
自己实现的高斯滤波



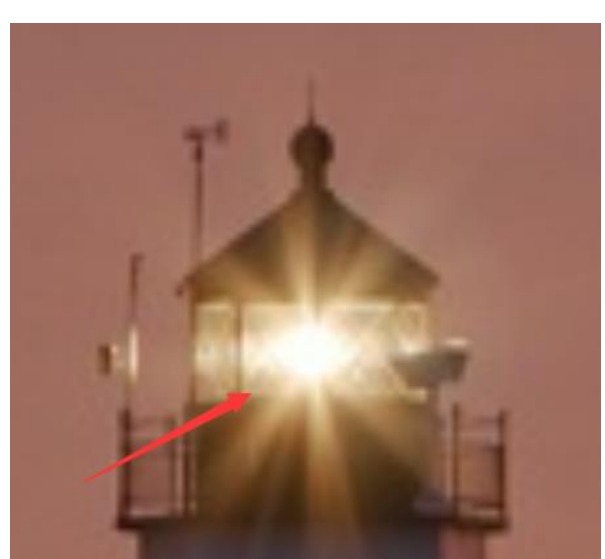
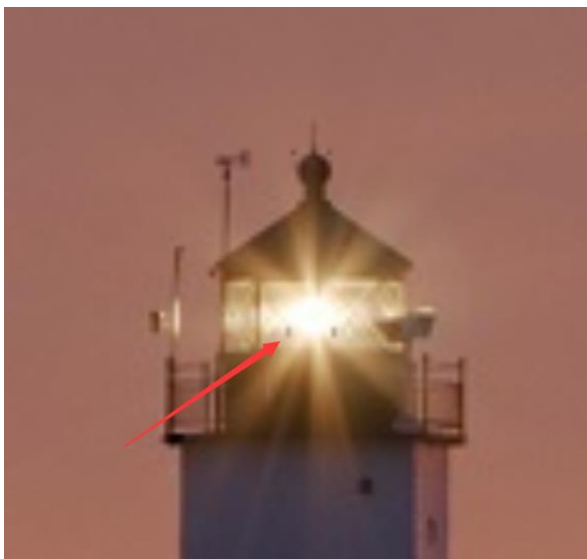
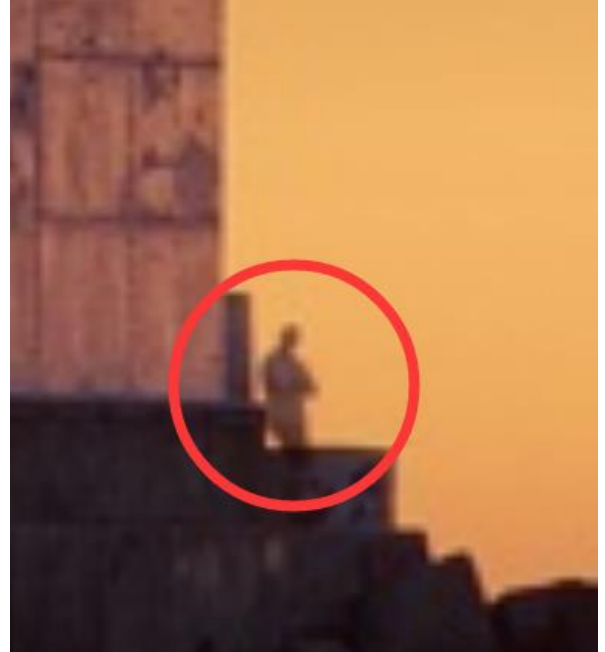
模板实现的高斯滤波

上面两幅图是在求得三幅图的梯度图并取反再融合的梯度图。虽然模板的轮廓更加清晰但是也产生了很多杂质，在后期计算过程中被不断放大，导致效果不好。

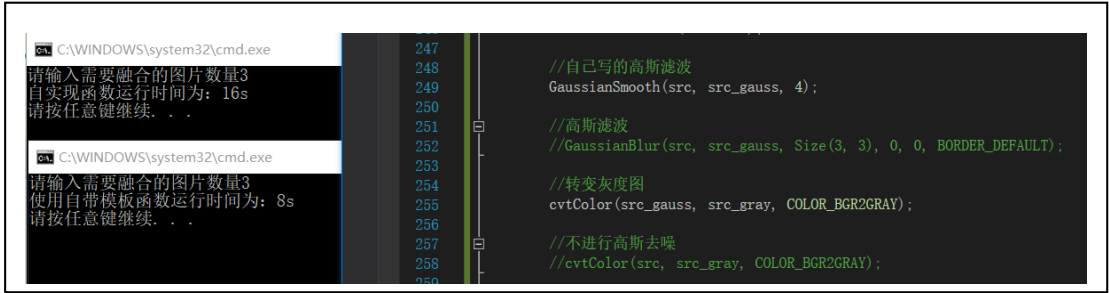
模板高斯滤波函数



自己实现高斯滤波



运行时间对比如下：

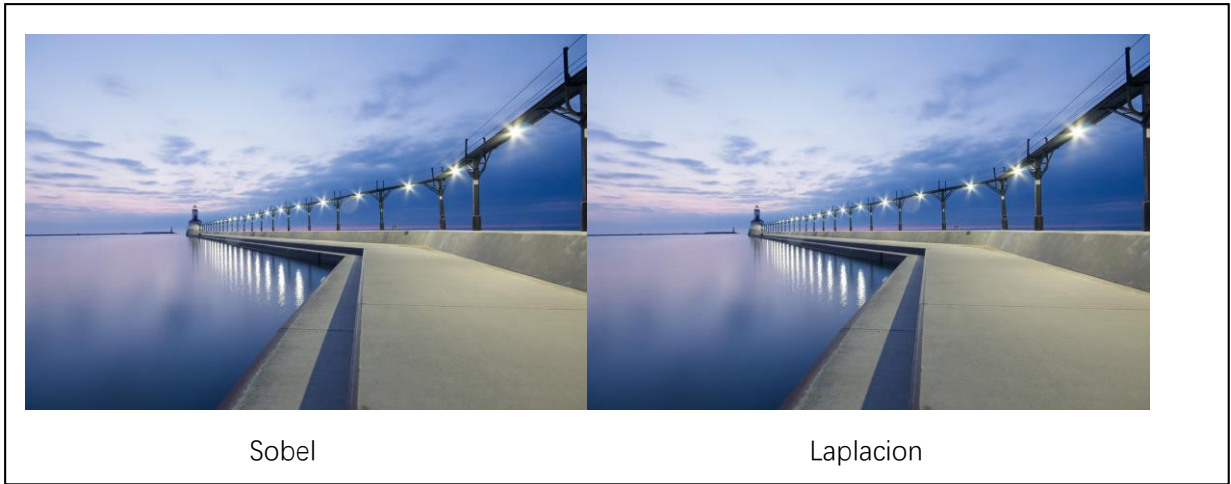


```
C:\WINDOWS\system32\cmd.exe
请输入需要融合的图片数量3
自实现函数运行时间为: 16s
请按任意键继续. . .

C:\WINDOWS\system32\cmd.exe
请输入需要融合的图片数量3
使用自带模板函数运行时间为: 8s
请按任意键继续. . .

247
248 //自己写的高斯滤波
249 GaussianSmooth(src, src_gauss, 4);
250
251 //高斯滤波
252 //GaussianBlur(src, src_gauss, Size(3, 3), 0, 0, BORDER_DEFAULT);
253
254 //转变灰度图
255 cvtColor(src_gauss, src_gray, COLOR_BGR2GRAY);
256
257 //不进行高斯去噪
258 //cvtColor(src, src_gray, COLOR_BGR2GRAY);
259
```

方法二：方法二利用梯度方向不会随曝光度改变，而且移动物体所在区域纹理走向和其他图像明显不同，使得可以利用梯度方向来判断运动物体所在的区域。利用梯度方向作为权值的算法也可以有效改善 sobel 算子对图像处理的能力，与 laplacian 算子的结果几乎没有差异



本人按照老师给的论文，奋战 12 小时，计算出梯度，梯度方向，权值。融合后的效果极差。

在网上翻到您指导发表的论文《基于子带结构和梯度的多曝光图像融合技术研究\_赵颖》中的投票思想，没有完全领会。仅仅利用将梯度方向计算出的权值进行融合得到的效果与方法一的效果相差不多。仅在一些局部实现了消除鬼影，效果差强人意。





遗憾没能完全整个作业。