

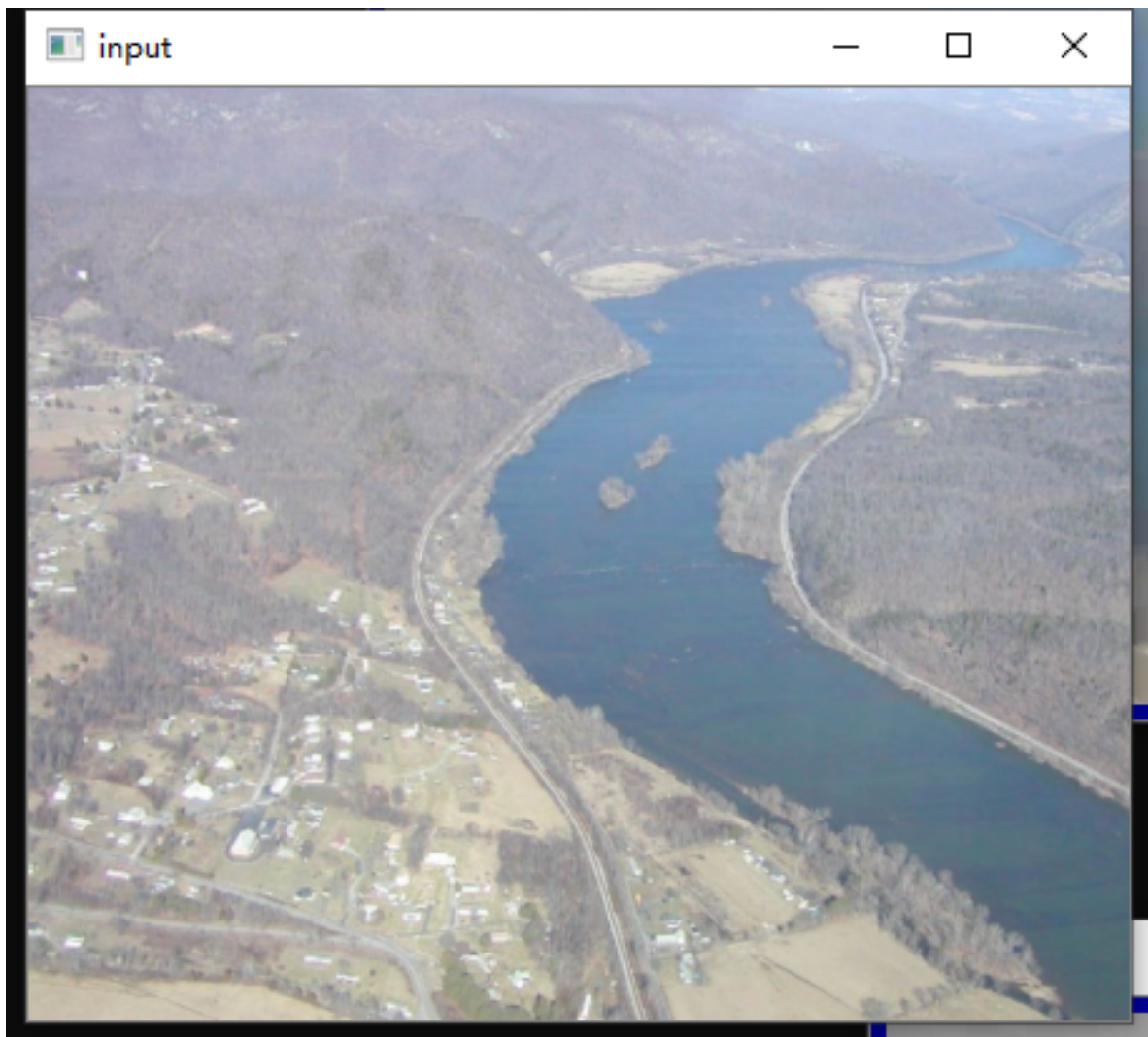
# 学习目的

---

- 1.暗通道先验去雾
- 2.初步学会OpenCv的使用
- 3.配置OpenCv环境

## 实验数据(1)

Input: 即输入的目标是带有雾气的图片



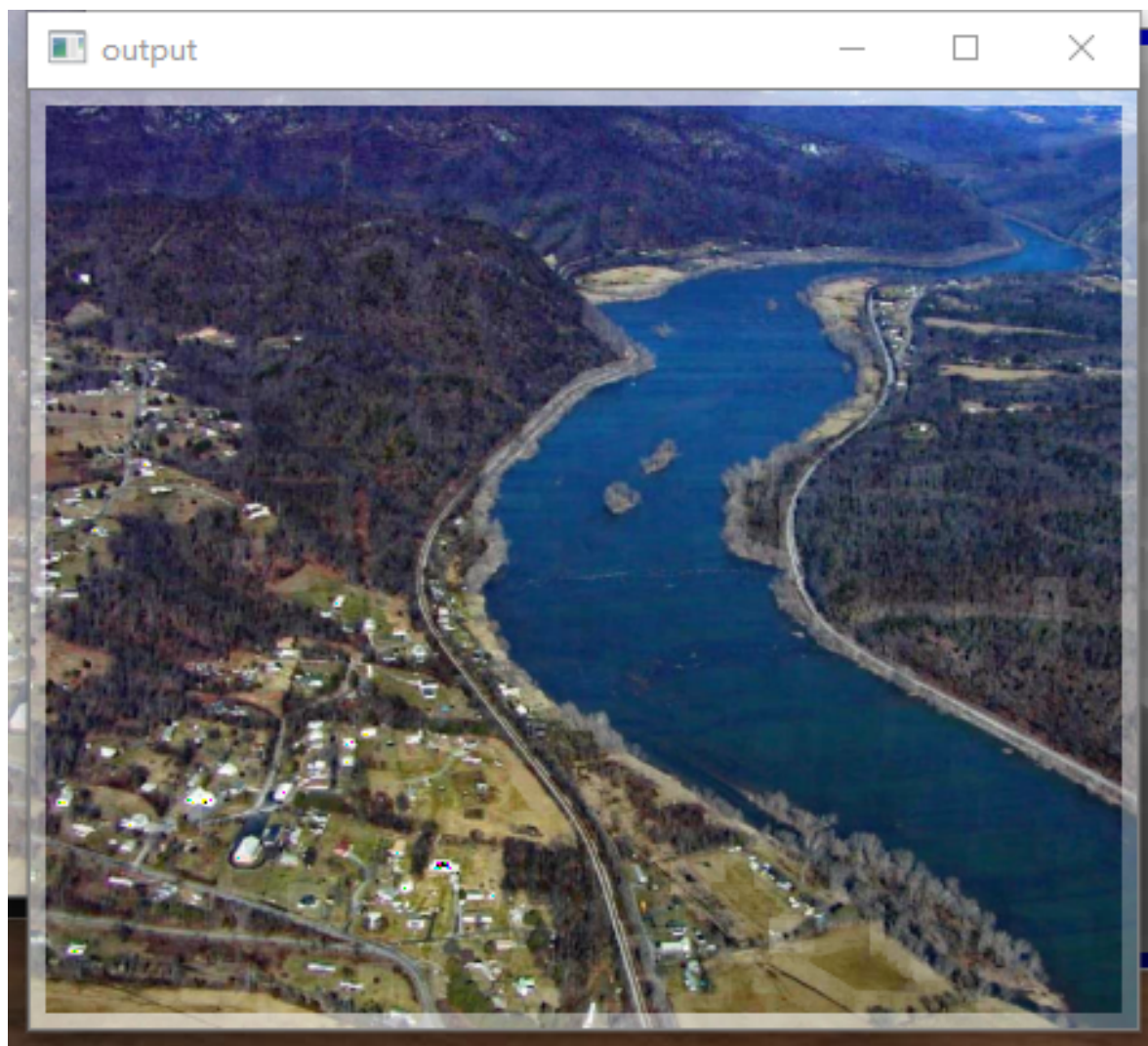
output2:用最小值滤波之后产生的暗通道



output12:直接对图片进行最小值滤波（方便求解 $t(x)$ ）

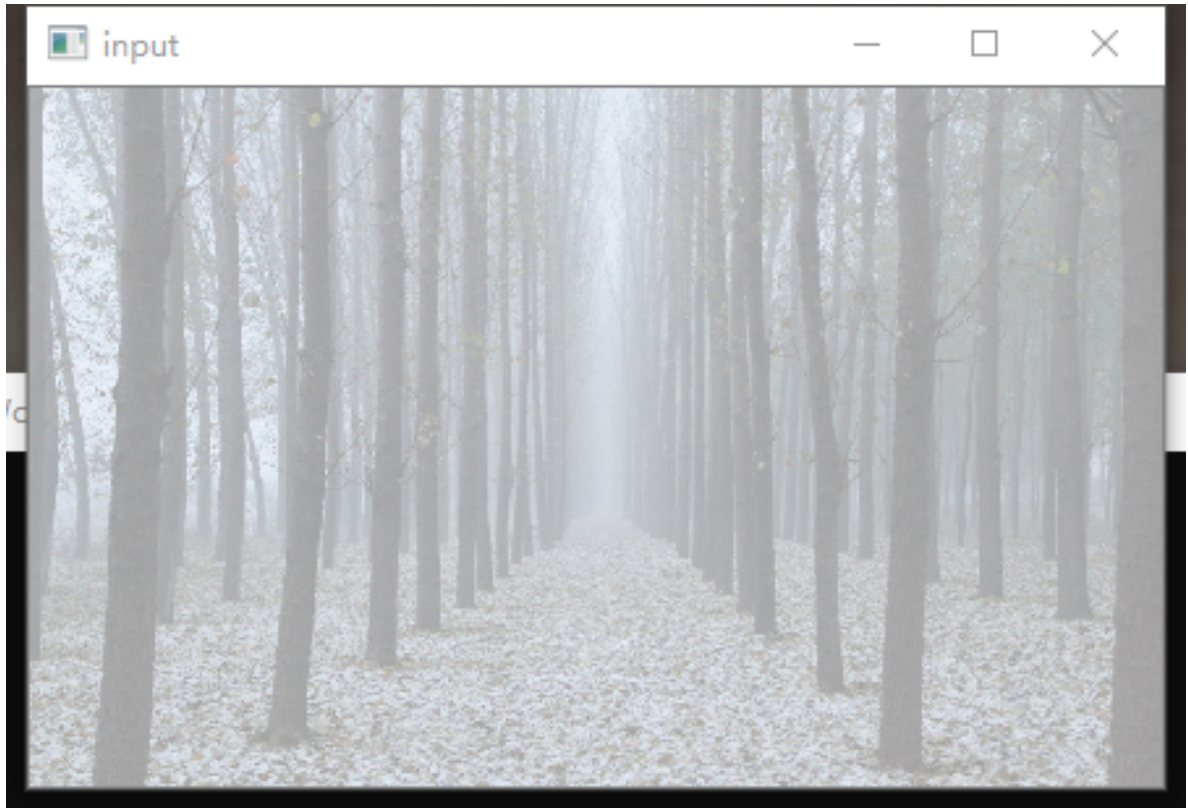


output:使用暗通道先验去雾之后得到的去雾图



## 实验数据(2)

Input: 即输入的目标是带有雾气的图片

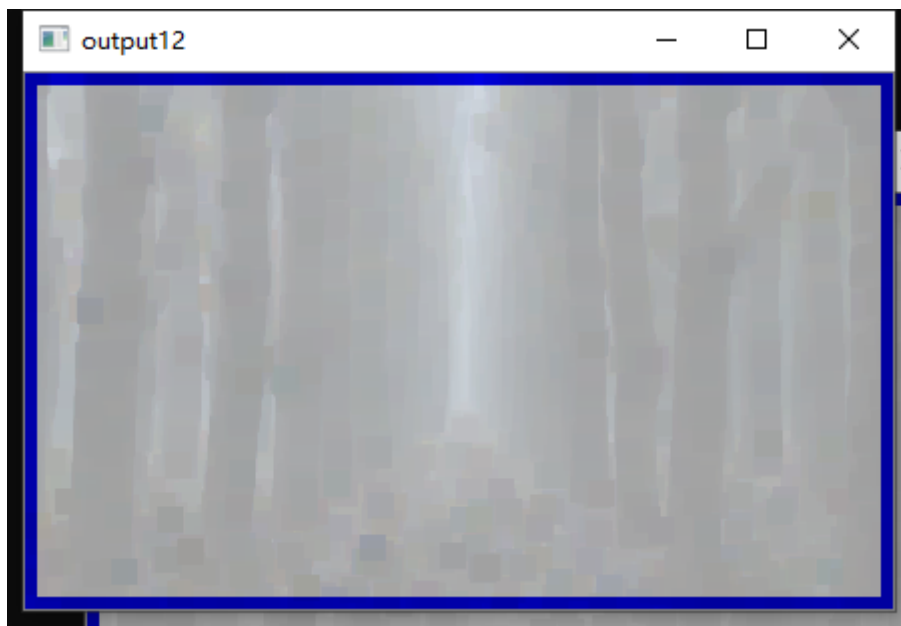


output2:用最小值滤波之后产生的暗通道

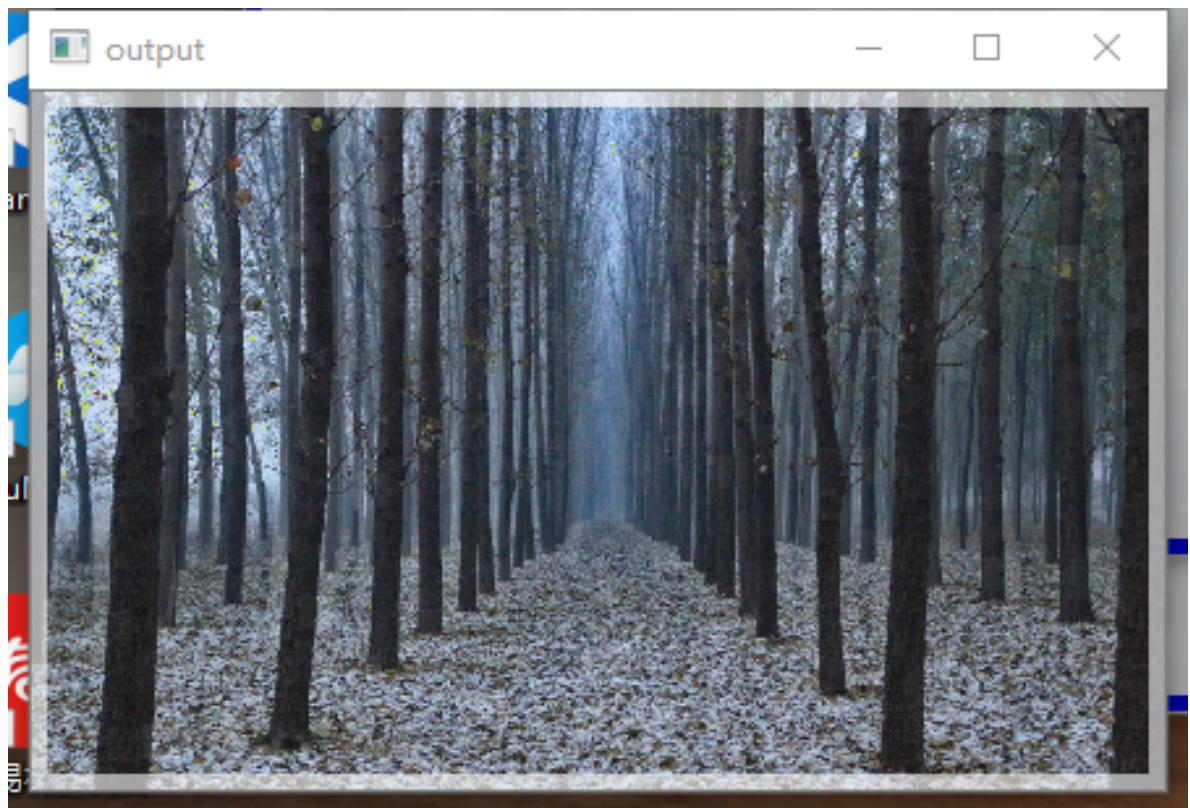




output12:直接对图片进行最小值滤波（方便求解 $t(x)$ ）

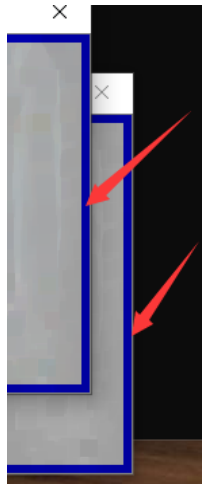


output:使用暗通道先验去雾之后得到的去雾图

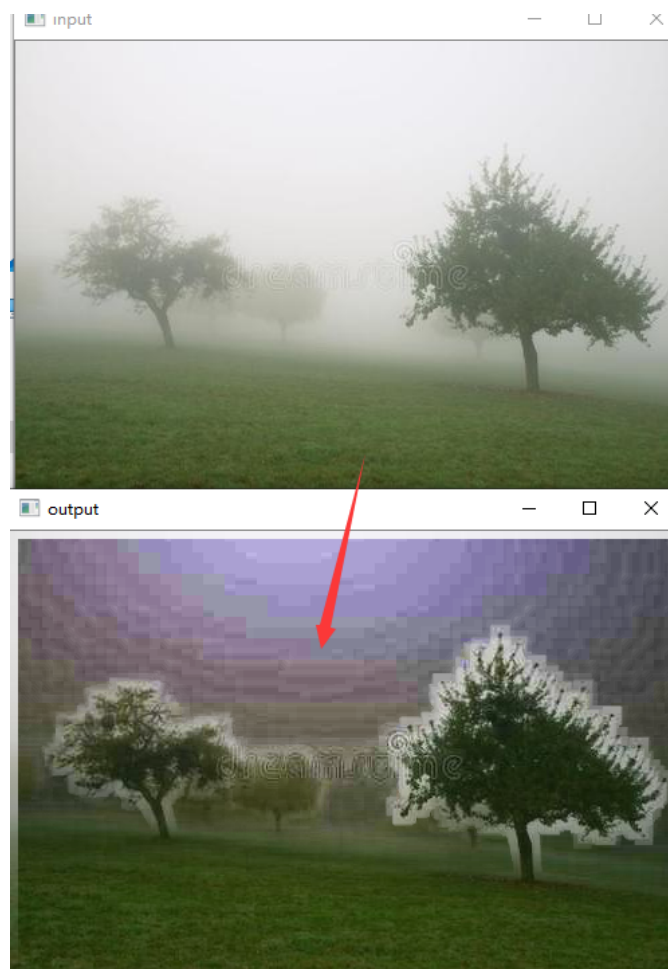


## 遗留问题

1.当我使用最小值滤波的时候会留下一个蓝色的边框再图片上，我怀疑是我的最小值滤波实现的逻辑有问题，并且我写的滤波时间复杂度也略高。



2.当使用的图片清晰度非常低的时候容易出现半圈很严重的光晕  
(也曾经怀疑是使用了jpg格式导致的，但是修改格式并不能解决问题)



### 3.滤波执行时间非常的长，时间复杂度远超想象

(因为网上通常是最大最小均值的开源代码一起给，但我只需要最小值，所以学习之后自我实现，可能在实现过程中导致了代码的冗杂)

```
Mat MinFilter(Mat src, int kernel) {
    int r = (kernel - 1) ;
    Mat dst_ex;
    Mat dst;
    dst.create(src.size(), src.type());
    copyMakeBorder(src, dst_ex, r, r, r, r, BORDER_CONSTANT, Scalar(255));
    for(int k=0;k<3;k++)
    {
        for (int i = r; i < dst_ex.rows - r; i++)
        {
            for (int j = r; j < (dst_ex.cols - r); j++)
            {
                int minVal = dst_ex.at<Vec3b>(i, j)[k];
                for (int s = -r; s < r + 1; s++)
                {
                    for (int t = -r; t < r + 1; t++)
                    {
                        if (dst_ex.at<Vec3b>(i + s, j + t)[k] < minVal)
                        {
                            minVal = dst_ex.at<Vec3b>(i + s, j + t)[k];
                        }
                    }
                }
                dst.at<Vec3b>(i - r, j - r)[k] = minVal;
            }
        }
    }
    return dst;
}
```