

# 数字图像处理 作业报告 第2周

2011011309 邓佳迪 计 15

## 一、 Point Processing 的各种操作

算法描述：对图片的每个像素进行依据公式进行处理，并把处理的结果输出。可得图片如下图所示：



图1 左上：原图 右上：增加亮度 左下：增加对比度 右下：gama修正 $\gamma < 1.0$





图2 左上：原图 右上：减少亮度 左下：减少对比度 右下：gama修正 $\gamma > 1.0$

实现代码：见 ‘point\_processing.m’

## 二、 Histogram Matching 的实现

算法描述：先原图像和目标图像的颜色进行概率分析，通过对比两者的概率(使得出现在原图像和目标图像的两种颜色的概率一致的颜色相对应)，制作 look up map，最后进行图像处理。



图3 左上：原图 右上：目标图片 左下：生成图片

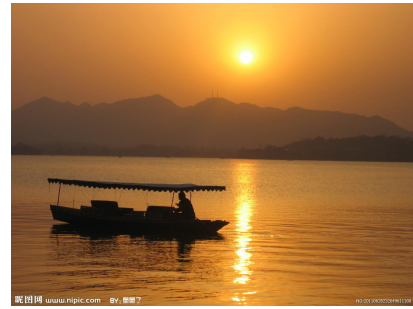


图 4 左上：原图 右上：目标图片 左下：生成图片

代码实现：见 ‘point\_processing.m’和 ‘remap.m’

### 三、 去雾算法的实现

算法描述：

1. 获取目标图像的暗通道
2. 计算大气亮度
3. 对图片进行 guided filter
4. 通过雾霾的计算公式求出复原图像





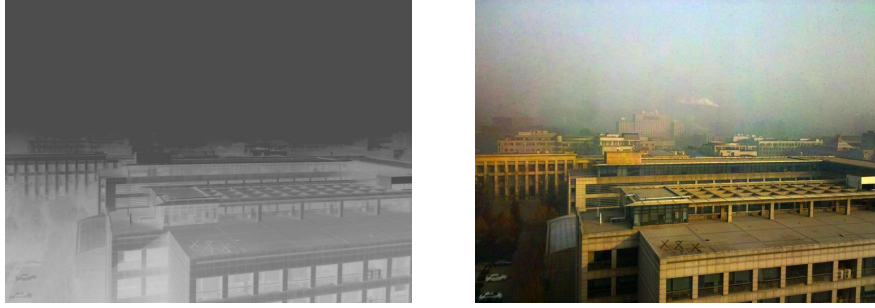


图 5 左上：原图 右上：暗通道图 左下：导向图 右下：最终复原图



图 6 左上：原图 右上：暗通道图 左下：导向图 右下：最终复原图

#### 四、 讨论题：

1. 你觉得要是做 Point Processing 的时候不使用 Look up table 而是对每个像素直接计算，有何利弊？

利：节省程序运行时间开销（访问 cache 比做浮点数运算快），让程序编写模块化（对于每种操作只要定义一张 look up table 就可以了）

弊：空间开销增加（尽管可以忽略）

2. 课上提到的 Histogram Matching 的算法复杂度是多少？为什么？

（假设每像素的比特数为  $n$ ，请估计算法最坏情况下执行操作数与  $n$  的关系）。

每像素比特数为  $n$ ，则共有  $2^n$  个可用的颜色，在使用 Histogram Matching 算法时，考虑到每个颜色本身是有序的，所以进行概率比较的复杂度为  $O(2^n)$