

图像处理 HW5：图像提亮

李天宇 2200013188 信息科学技术学院

1 基本实现

1.1 直方图均衡化

OpenCV 对于直方图均衡化的实现接口如下：`cv.equalizeHist(src[, dst]) -> dst`。

1.2 CLAHE

`cv.createCLAHE([, clipLimit[, tileGridSize]]) -> retval` 创建一个指向 `cv::CLAHE` 类的智能指针并将其初始化。接着，使用 `cv.CLAHE.apply(src[, dst]) -> dst` 即可处理图像。

1.3 一个问题

这两个调用针对的都是灰度图像，而我们要处理的是彩色图像。课件中给出了一种解决办法：转为 HSL/HSV 空间，只对 L/V 均衡化。不过还应注意，OpenCV 提供的是 $RGB \rightarrow HLS$ 的接口，要注意 L 参数的位置。

2 手动实现直方图均衡化

算法实现思路如下：

- 计算 `src` 的直方图 H
- 归一化直方图（总和为 255）
- 计算直方图积分

$$H'_i = \sum_{0 \leq j < i} H(j)$$

- 使用 H' 作为查找表转化图像：

$$\text{dst}(x, y) = H'(\text{src}(x, y))$$

在实际算法中，是通过计算图像的 *cdf* 而实现的。

3 进阶功能实现

3.1 手动实现 CLAHE

实现 CLAHE 分为如下几步：

1. 分割图像为小块，tileGridSize 是分成的块数
2. 计算每个小块的灰度直方图
3. 限制对比度，计算 CDF 并进行均衡化
4. 插值和边界处理，合并子块

在实际的代码编写中，这一部分是最麻烦的，因为它涉及到分区域处理，我也没有什么简洁的处理方法。我按照课件的方法，将图像分为三种块，共九块，并分别编写了处理函数。另外，在限制对比度上，我只分配了一次，也就是分配后允许超出阈值。以下是三张图的对比：



图 1: CLAHE

可以看到，在未做插值处理时，图像分块明显，各部分明暗不一，经过插值处理，图像更加自然。

3.2 手动实现直方图匹配

直方图匹配需要通过 cdf_{src} 和 cdf_{ref} 来计算像素值的映射关系。记

$$dst(x, y) = f(src(x, y))$$

则有

$$f(x) = \arg \min_y |cdf_{src}(x) - cdf_{ref}(y)|$$

代码实现不难。

4 后记

结果到最后，我还是没有去更新前面的函数，能跑就行。也许以后有动力了会去更新一下吧 ()
但是无论如何，这门课的五个小作业确实很有趣：原理并不算特别难，同时也能实际写一些代码，实现一些实用的功能。我自认为，在这门课收获颇丰，同时，也很有成就感。
以上。