

低光照增强算法：CLAHE

直方图均衡 PLUS 版本，随便快速写写。逻辑就是针对直方图均衡的缺点进行改进。CLAHE 算法处理 RGB 时转换为 HSV，然后只对 V 分量进行处理。

1. 通过设置阈值防止直方图均衡的结果过于平均带来的问题 (如对于像素集中的图片, 如晚霞, 均衡可能就会让原来相近的像素差异增大, 导致看起来很假)。

具体思路: 设置阈值, 超过阈值的部分才进行均衡, 也就是平摊到各个值之中, 看下图即可:

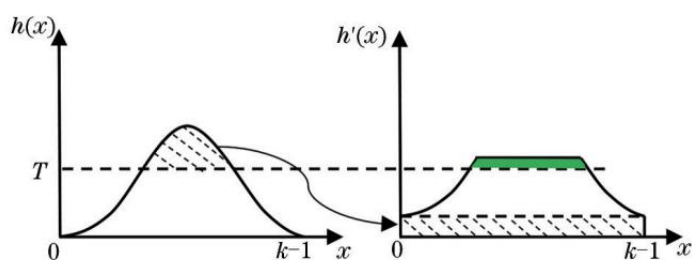
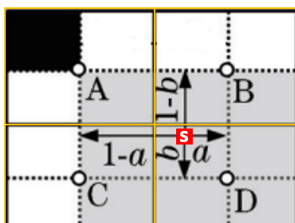


图1 直方图裁剪

2. 局部均衡化。一张图本来是亮的地方亮, 暗的地方暗, 整体均衡会带来一些问题。

具体思路: 很显然, 图片分成许多块, 认为每个块里面的光照比较一样, 有亮块有暗块, 每个块进行直方图均衡, 这个方法叫做 Adaptive Histogram Equalization。但他的问题是亮暗交界处, 亮的小块和暗的小块, 做均衡化的结果大不一样, 输出的图片有明显的突变。

CLAHE 算法的改进是采用了双线性插值。对于任意一个小块, 会考虑附近几个小块的信息, 这样看上去就平滑了好多了:



6) 插值运算, 如图 2 所示, A、B、C、D 分别为 4 个相邻子图像的中心, $P_A(s)$ 、 $P_B(s)$ 、 $P_C(s)$ 、 $P_D(s)$ 分别为它们所在子图像区域的映射函数。设 s 为区域内某点上的像素值, 则它的新像素值 s' 可通过与周围相邻区域进行插值运算获得, 即

$$s' = (1-b)[(1-a)P_C(s) + aP_D(s)] + b[(1-a)P_A(s) + aP_B(s)] \quad (5)$$

在图像的边缘和角点处, 插值方法有所不同, 获得 s' 的公式为

$$s' = (1-a)P_A(s) + aP_B(s), \quad (6)$$

$$s' = P_A(s). \quad (7)$$