

第一次作业：直方图均衡化

姓名：丁保荣 学号：171860509

2019 年 9 月 19 日

1 全局直方图均衡化

全局直方图均衡化的代码是 `hist_equal()` 函数 (在 `hist_equal.m` 中).

1.1 主要实现思路

代码中有详细注释。

设定最大灰度 ($L-1$) 为 255, 提取图像的长宽, 然后计算 H . $H(i)$ 是对应灰度的像素点个数。然后根据 H 运用公式计算出转换函数 $T: r \rightarrow s$. 然后对原图像运用转换函数 T , 并返回新的图像。在实现的过程中出现的问题主要是一开始几乎整张图都是白色的, 后来发现是类型的问题, 需要把类型转换成 `uint8`. MATLAB 的类型转换不能那么写 `(uint8)(s)`, 得这样写 `uint8(s)`. 一开始不知道以为不能这样做类型转换。

1.2 实现效果



图 1: 原图像



图 2: 全局直方图均衡后



图 3: 原图像



图 4: 全局直方图均衡后

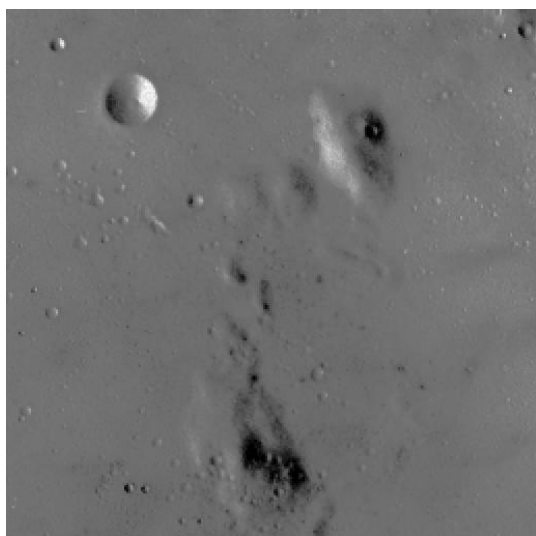


图 5: 原图像



图 6: 全局直方图均衡后

2 局部直方图均衡化

局部直方图均衡化的代码是 `local_hist_equal()` 函数 (在 `local_hist_equal.m` 中)。

2.1 主要实现思路

代码中有详细注释。

B 是设置邻域的边长, 经过多次的实验, 发现 B 设置成 31 的效果还行, 也就是邻域的大小是 31×31 . 然后统计了一系列关于图像的信息, 例如均值, 方差。本来想着给局部直方图统计增强用的, 但发现效果并不好, 感觉可能是自己系数 (k_0, k_1, k_2) 取得不太好, 而且可能系数跟具体的图像也有关。然后就是将区块在这个图像中逐个移动, 然后对每个区域进行直方图均衡化, 即算出对中心点的转换函数。我运用排序来确定中心点的转换函数, 并没有算出所有灰度的。排序的复杂度是 $O(2B^2 \log B)$, 因为遍历邻域本来就要 $O(B^2)$, 所以在 B 很大的时候, 复杂度也还行, 当 B 比较小的时候, 这样实现的复杂度就比算出所有灰度值转换函数的复杂度低了。

2.2 实现效果

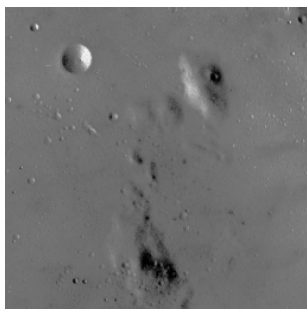


图 7: 原图像



图 8: 全局直方图均衡后

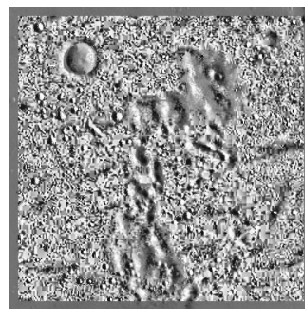


图 9: 局部直方图均衡后

参考效果:https://scikit-image.org/docs/dev/auto_examples/color_exposure/plot_local_equalize.html#sphx-glr-auto-examples-color-exposure-plot-local-equalize-py

3 彩色图像均衡化

彩色图像均衡化的代码在 ByHSV.m 和 ByHSL.m 中。

3.1 主要实现思路

代码中有详细注释。

彩色图像的处理，主要是将 RGB 色彩空间转换到 HSV 或者 HSL 色彩空间上去，并对饱和度或亮度（通过控制比特来调节）做全局直方图均衡化或者局部直方图均衡化。然后均衡化之后再转换回 rgb 色彩空间。在这过程中要注意一些类型的转换。

MATLAB 自带了 hsv 转换的函数，但没有 hsl 的转换函数。我在网上找到了一个 hsl 的转换函数，并对它进行了一些修改。

3.2 实现效果



图 10: 原图像



图 11: HSV 转换 + 对饱和度全局直方图均衡后



图 12: 原图像



图 13: HSL 转换 + 对饱和度全局直方图均衡后

发现通过 HSV 转换的，饱和度更高。



图 14: 原图像

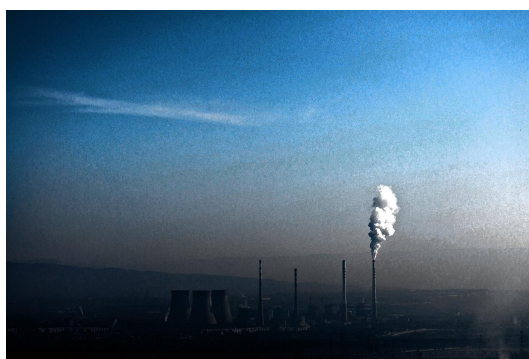


图 15: HSV 转换 + 对亮度全局直方图均衡后



图 16: 原图像



图 17: HSL 转换 + 对亮度全局直方图均衡后



图 18: 原图像



图 19: HSV 转换 + 对饱和度和亮度全局直方图均衡后



图 20: 原图像



图 21: HSL 转换 + 对饱和度和亮度全局直方图均衡后

4 对数变换

对数变换的代码在 `log_conversion.m` 中

变换公式: $s = c \log(1 + r)$

4.1 实现效果



图 22: 原图像



图 23: 对数变换后的图像

5 幂等变换

幂等变换的代码在 `power_conversion.m` 中

变换公式: $s = cr^\gamma$

5.1 实现效果



图 24: 原图像



图 25: 幂等变换后的图像, $\gamma = 2$



图 26: 原图像



图 27: 幂等变换后的图像, $\gamma = 3$