

低光照增强算法：基于 Retinex

Retinex 理论的不严谨概括如下，其中 I 是观测到的图像， R 是要预测的图像， L 是环境光：

$$I = R * L$$

问题的关键在于 L 是什么，怎么求。

SSR

这个最简单的方法，认为 L 可以用 I 的高斯模糊图像进行近似。即：

$$\log R = \log(I) - \log(F(I))$$

其中 F 表示高斯核，大小由用户自己定义。求完之后对 $\log(R)$ 归一化到 $0-\ln 255$ ，然后做一个 e 的指数。为什么对 $\log(R)$ 进行 scale，而不是 e 完后再做？因为 e 会后放大了差异。

这里的实现是有注意点的：第一， I 是转为 $0-1$ 的浮点数；

MSR

在 SSR 中有一个变量，那就是高斯核的大小。不同大小有不同效果，小的时候，细节多但光晕现象严重；大的时候色彩表现好，但对比度低，细节丢失多。下图是小到大的结果：





MCSR 就是采取几个核进行恢复之后，融合起来（其实就是加权求和）：

$$\sum_{n=1}^N w_n [\log I_i(x, y) - \log(F_n(x, y) * I_i(x, y))]$$

一般而言，取 15、80、200 的 sigma，w 平均分（即各自为 1/3），效果比较好。

MSRCR

上面的方法有这样的问题：当图片中不满足灰度世界理论（即 RGB 三个通道的值不太平衡，比如红色比较多），得出的结果往往会减小饱和度，使得图片看起来暗淡无光。

MSRCR 就是解决这一问题，实际上在上一个方法得出结果之后，根据 RGB 的值做一些不同权重的调整，其中 α 和 β 是用户的参数，下标 i 表示是红绿蓝哪个通道：

$$I'_i(x, y) = \frac{I_i(x, y)}{\sum_{j=1}^S I_j(x, y)} \quad \Rightarrow \quad C_i(x, y) = \beta \log[\alpha I'_i(x, y)]$$

$$\text{MCRCR}_i(x, y) = C_i(x, y) * \text{MSR}_i(x, y)$$

对光照分量进一步处理

看到一个博客，里面提到用【二维 Gamma】来对提取出的光照分量处理一下，之后再进行图像校正。之所以调整，还是如上面图片所展示的，有的时候会有光晕，说明光照分量不要那么极端。

均匀图像整体质量的目的. 对于输入的图像 $F(x,y)$, 假设提取出的光照分量为 $I(x,y)$, 在参考文献[12]的基础上, 构造了一种新的二维伽马函数, 其表达式如下 $O(x,y) = 255 \left(\frac{F(x,y)}{255} \right)^{\gamma}, \gamma = \left(\frac{1}{2} \right)^{\frac{I(x,y)-m}{m}}. \quad (5)$ 式中: $O(x,y)$ 为校正后的输出图像的亮度值; γ 为用于亮度增强的指数值, 其中包含了图像的光照分量特性; m 为光照分量的亮度均值. 如果光照分量	注意这里指数应该是 $m - I(x,y)$
--	--

LIME

Low-light Image Enhancement via Illumination Map Estimation, TIP 2017, Xiaojie Guo, CAS (中科院)

这篇文章认为光照 L 应该可以用这样的方式来模拟：

$$L^c(x) = \max_{c \in \{R,G,B\}} I^c(x)$$