

去紫边算法记录

由于项目上对资源的要求还是比较苛刻的，而且又是线性扫描相机，所以参考的主要是三四篇 10 年左右的文章，复现后发现效果也还行，特此记录。参考文章如下：

1. Removing chromatic aberration by digital image processing
2. Automatic detection and correction of purple fringing using the gradient information and desaturation
3. 数字成像系统图像消紫边方法研究，浙大硕士生毕业论文

这里记录的是 Depurple 方法，即后期处理。去紫边包括前期配合镜头的 CAC 校正、后期的 ISP 算法。

论文记录如下：

1. chung2009 和 chung2010：这两篇是一样的，在文档后面有记录（检查方法一+消除方法二）
2. kim2008：这篇也是在 docx 文档有记录（检查三，消除未记录），反正很简单，配合着看十几分钟完事
3. kang2007：这是最早的文章了，引用也多，但是感觉没必要看了，作者喜欢唬人，文章无谓公式太多
4. kang2010：直接看几个公式就好，但是感觉实用价值不大
5. chang2013：这个感觉有点道理，就是太复杂而且效果不明显，最后弃用了。paper-2 是复现论文，github 也有代码。写给我自己：本地跑通过，但感觉没多大用。

去紫边主要分为两个部分：检查紫边和消除紫边。

检查紫边

方法一

现在处理第 i 行，对于某个像素 (i, j) ，根据周边的像素，RGB 三个颜色分别做计算，计算的值叫做 E ，其中绿色点就是像素 (i, j) ：

1	0	-1
2	0	-2
1	0	-1

设：计算好的值分别为 E_R 、 E_G 、 E_B ，有一个用户输入的参数：阈值 T

在第 i 行往右遍历找到 $E_G > T$ 的像素点，假设位置为 p ，即 $\text{abs}(E_G[i, p]) > T$ ，记录 $E_G[i, p]$ 的符

号位, 设为 sign:

1. 从 p 往左遍历, 假设当前遍历的位置为 m, 判断 $\text{sign} * \max(\text{ER}[i, m], \text{EG}[i, m], \text{EB}[i, m]) > T$ 是否成立。如果成立, 则继续往左遍历; 如果不成立, 则终止。
2. 从 p 往右遍历, 假设当前遍历的位置为 n, 判断 $\text{sign} * \max(\text{ER}[i, n], \text{EG}[i, n], \text{EB}[i, n]) > T$ 是否成立。如果成立, 则继续往左遍历; 如果不成立, 则终止。
3. 找到 m 和 n 后, [m,n] 就是之后要处理的紫边。

方法二

同方法一, 计算出 ER、EG、EB, 有用户输入的参数: 阈值 T。

对第 i 行进行处理, 如果位置 p 满足 $\max(\text{abs}(\text{ER}[i, m], \text{EG}[i, m], \text{EB}[i, m])) > T$, 那么就记录该位置为紫边像素点, 之后会对这个位置的像素进行处理。注意需要先取绝对值, 再选择最大值。

方法三

有用户输入的参数: 阈值 T

1. 每个像素判断 R-B 的值是否小于 T, B-G 的值是否大于 T。如果都满足, 则记录为 1, 否则记录为 0, 设计算好的值叫做 C1
2. 每个像素 RGB 分别做处理: 当前处理的颜色值为 I, 根据下述公式判断 I 在哪个区间, 设最终的结果分别为 QR、QG、QB

$$Q = \begin{cases} 3 & \text{if } 223 < I \leq 255 \\ 2 & \text{if } 191 < I \leq 223 \\ 1 & \text{if } 127 < I \leq 191 \\ 0 & \text{if } I \leq 127 \end{cases}$$

3. 每个像素每个颜色分别做处理: 第二步计算好了当前要处理颜色的 Q 值, 计算该点的 Q 值减去右边像素对应的 Q 值的绝对值。即假设像素点 (i, j), 那么计算 $|\text{QR}(i, j) - \text{QR}(i, j + 1)|$, 绿色蓝色同理。判断计算好的三个颜色的值是否都为 0; 如果是则记录为 0, 否则记录为 1, 设该步骤计算的结果为 C2
4. 如果某个像素的 C1 和 C2 都是 1, 那么就记录该点为待处理的紫边像素点。

消除紫边

消除紫边就是对检查紫边步骤检测出的紫边像素点进行处理, 每次处理一段。假设对第 i 行进行

处理，要处理的紫边像素点从位置 lo 到位置 hi。

方法一

记录位置 lo 和位置 hi 的绿色值，结果为 G1 和 G2，现在处理 lo 到 hi 之间的某个像素点 p：

如果位置 p 的绿色值在 min(G1,G2)和 max(G1,G2)之间，修改红色和蓝色值，以红色为例：

$$\text{如果 } G1 = G2: R[p] = \frac{G1 + G2}{2}; \text{ 否则: } R[p] = \frac{G[p] - G2}{G1 - G2} (R[lo] - R[hi]) + R[hi]$$

方法二

现在处理 lo 到 hi 之间的某个像素点 p，处理红色和蓝色的像素值，以红色为例：

1. 计算位置 lo、位置 hi 的 R-B 值，分别为 X1 和 X2；位置 p 也计算这样的值，设为 Xp
2. 如果 Xp 不在 X1 和 X2 之间，那么修改红色：R[p]=G[p]+(X1+X2)/2

方法三

现在处理 lo 到 hi 之间的某个像素点 p，处理红色和蓝色的像素值，以红色为例：

1. 计算位置 lo、位置 hi 的 R-B 值，分别为 X1 和 X2
2. 修改红色：如果 |X1| < |X2|，那么 R[p] = G[p] + X1；否则 R[p] = G[p] + X2