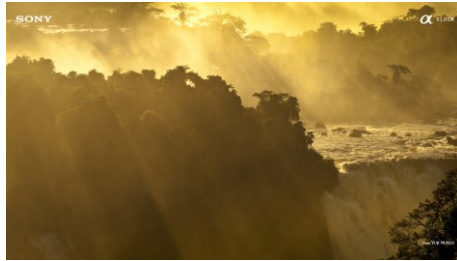


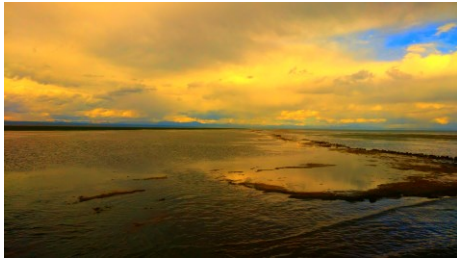
II. 根据课程 PPT 和对应参考文献实现颜色转换。

A. 测试图像集合，共 10 组：

第 1 组：(左边为原图，右边为参考图)



结果图：



第 2 组：(左边为原图，右边为参考图)



结果图：



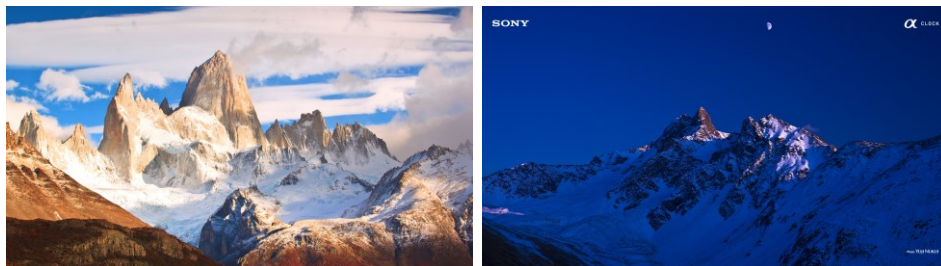
第 3 组: (左边为原图, 右边为参考图)



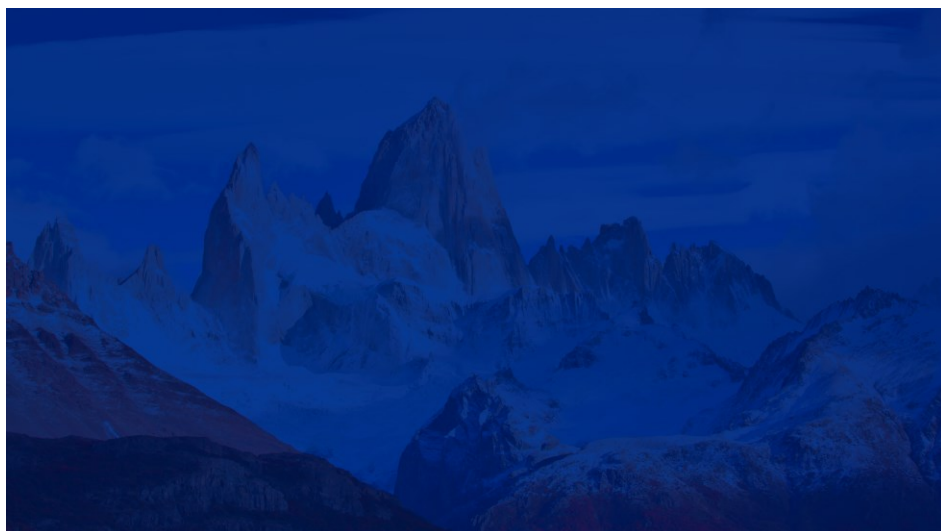
结果图:



第 4 组: (左边为原图, 右边为参考图)

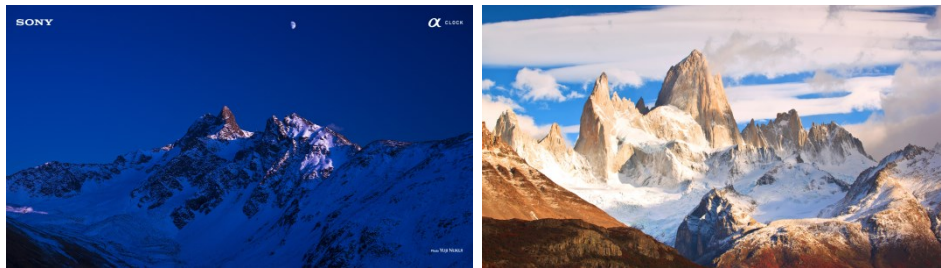


结果图:





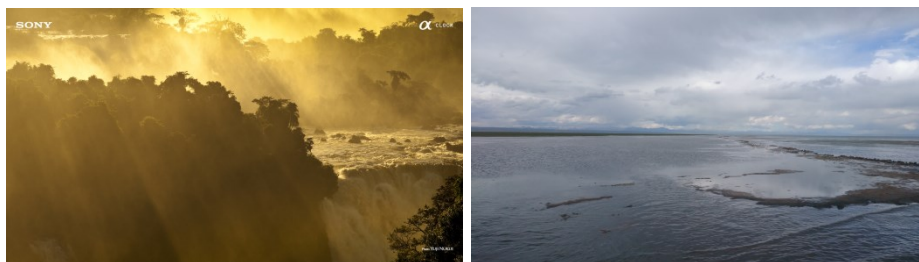
第 5 组: (左边为原图, 右边为参考图)



结果图:



第 6 组: (左边为原图, 右边为参考图)

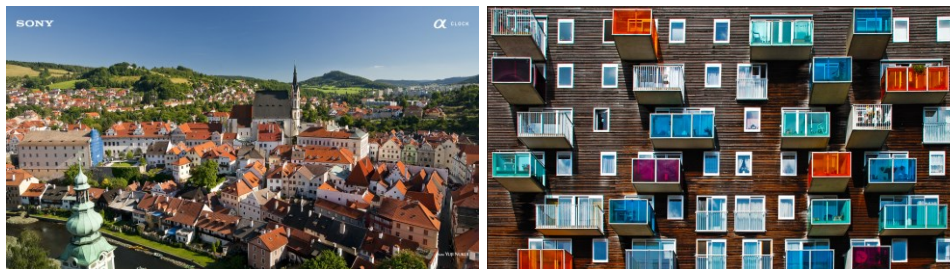


结果图:





第 7 组: (左边为原图, 右边为参考图)



结果图:



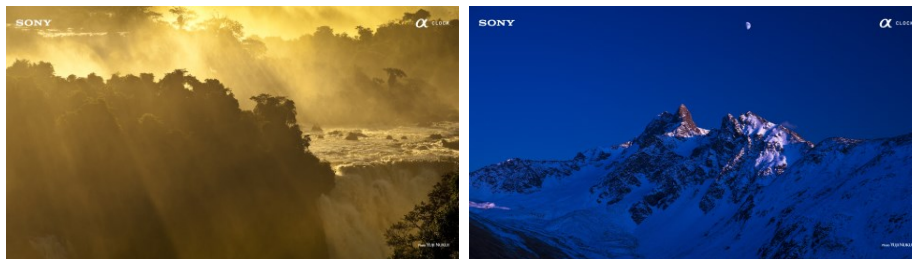
第 8 组: (左边为原图, 右边为参考图)



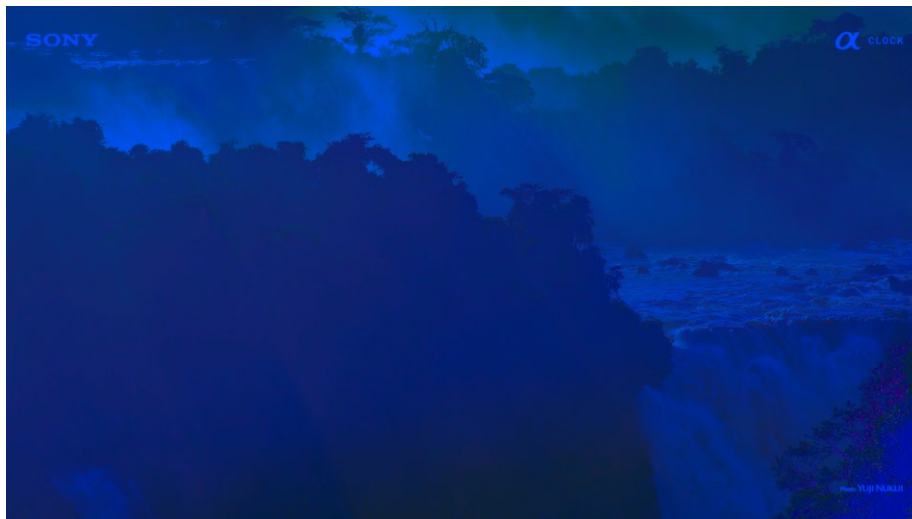
结果图:



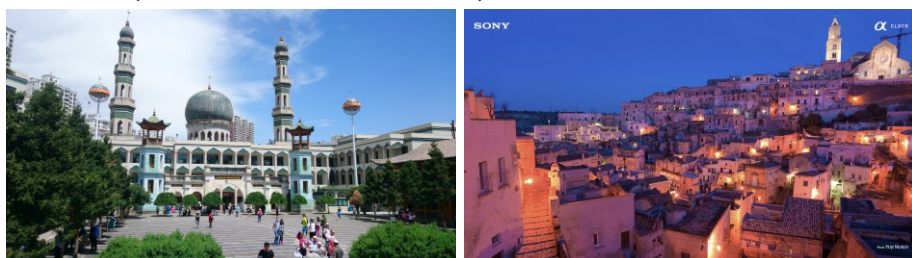
第 9 组: (左边为原图, 右边为参考图)



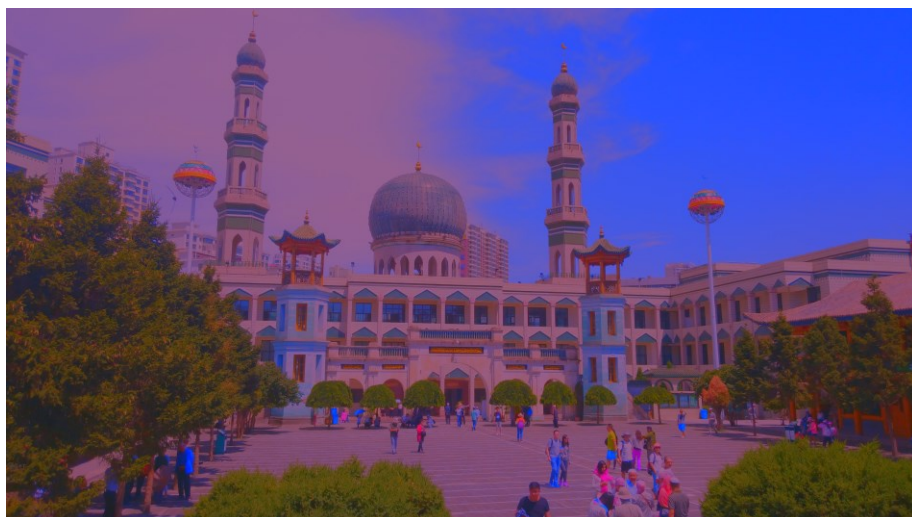
结果图:



第 10 组: (左边为原图, 右边为参考图)



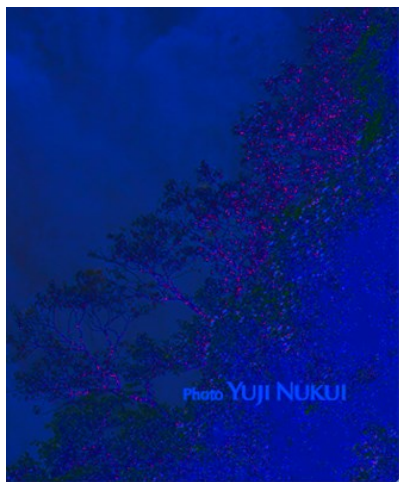
结果图:



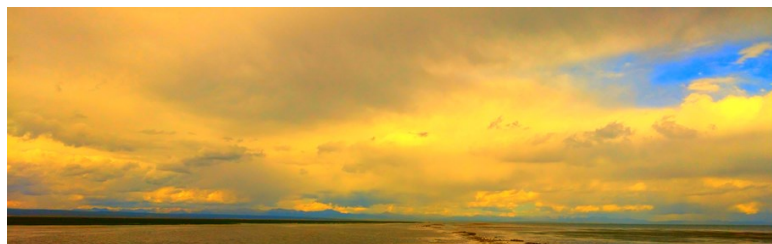


B. 对实验结果进行分析

答：在上面 10 组的结果中，我们可以看出，这个算法得出的结果，有一些还是有缺陷的，例如第 9 组结果图的右下角：



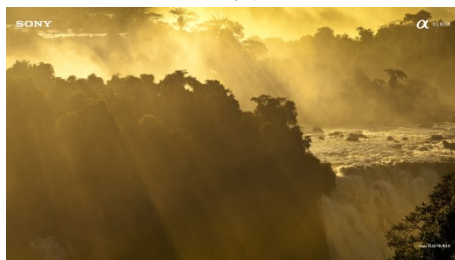
可以看到颜色上出现了一些缺陷，还有第 1 组的结果图：



和第 10 组的结果图的天空，我们可以看出在这个算法对于颜色的处理比较差。

明显，根据论文的算法，核心的转换过程是将 L、a、b 三个通道的直方图都大约看成高斯分布来进行的，于是我查看 L、a、b 三个通道的实际情况：

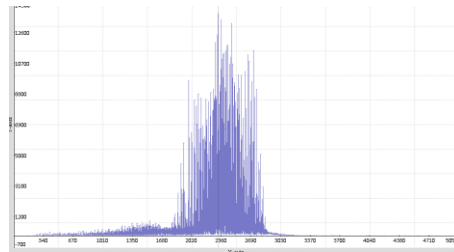
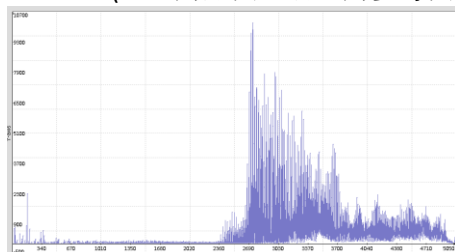
原图



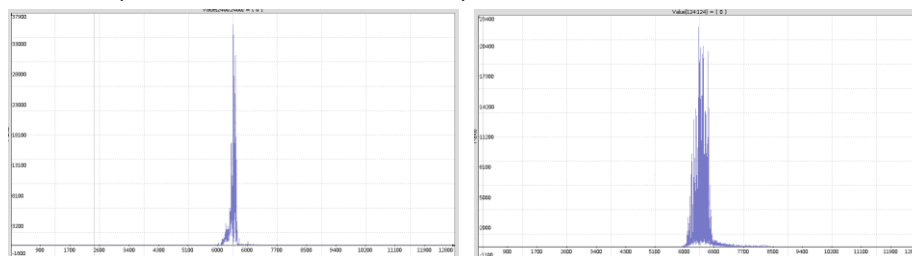
参考图



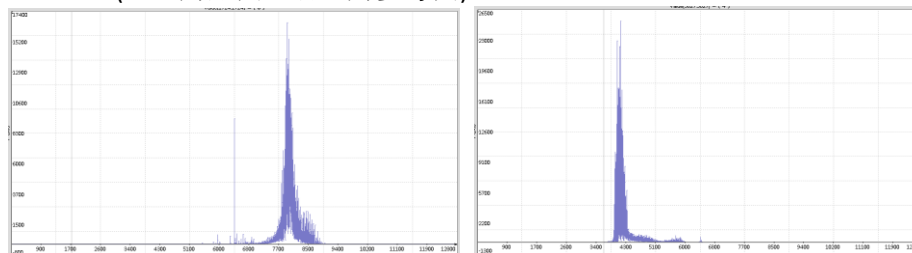
L 通道: (左边为原图, 右边为参考图)



A 通道: (左边为原图, 右边为参考图)

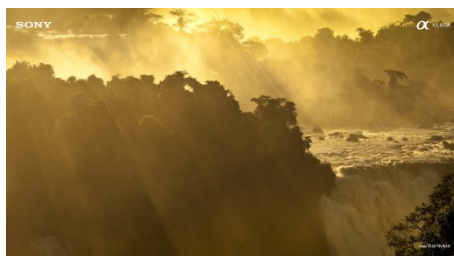


B 通道: (左边为原图, 右边为参考图)



从上面的直方图可以看出, L、a、b 三个通道都只是和高斯分布勉强符合, 应用论文中的公式, 明显是会有缺陷的。于是我根据论文中的思想, 即把参考图的直方图的分布当作原图直方图均衡化的目标, 然后分别对三个通道做均衡化, 最后转换回 RGB 空间。下面列出我的方式与论文方式的比较:

原图

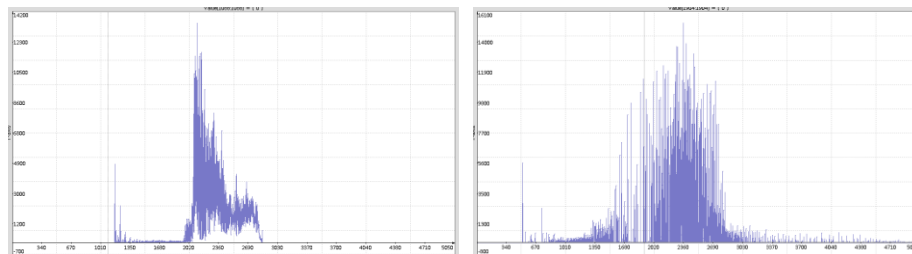


参考图

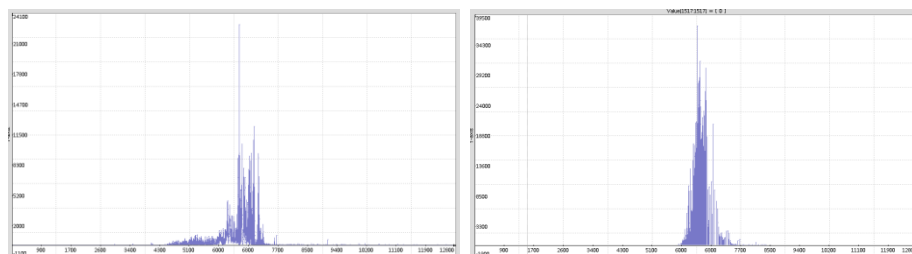


得到的结果图的三个通道的直方图:

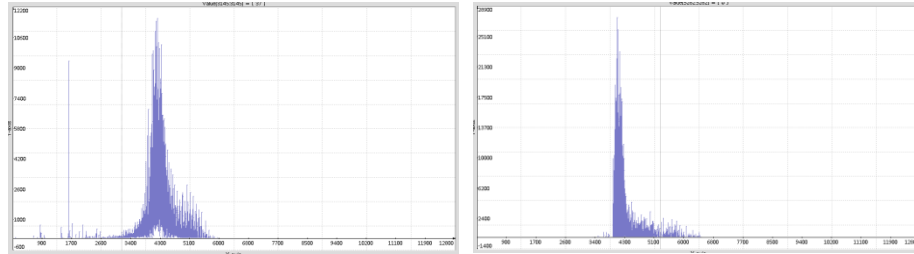
L 通道: (左边为论文法, 右边为我的方法)



A 通道:



B 通道:

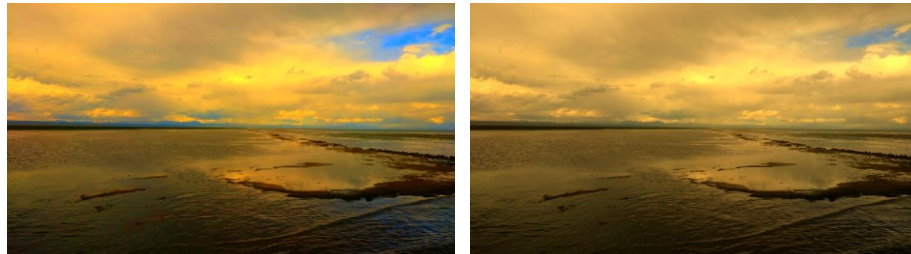


根据上面的结果图，对比上一页的参考图的 L、a、b 的三个通道图，可以发现我的方法得到的结果更加符合。

然后产生了一个新的问题，就是直方图均衡化有一个离散的过程，由于 L、a、b 三个通道是 float 类型的，而要做均衡化就必须把通道的值量化成整数。此处就存在一个取舍，例如，对于 L 通道，取值范围[0, 100]，如果直接就做精确到个位数的均衡化，就会把小数点后的数字（也就是会失去细节信息）丢掉。考虑到这样，我增加了一个常量 c，将取值范围放大 c 倍，这样就可以保留更多的信息，但同时会使程序的运行时间增加。同时考虑到了 Lab 空间的特性，只有 L 通道保存的信息会影响像素块空间上的连续性，最后决定对 L 通道执行论文方法，对 a、b 通道执行我的方法。最终形成我的结果。

下面列出一些结果图：(左边为论文法，右边为我的方法，其中 c 取 100)

第 1 组:



第 2 组:

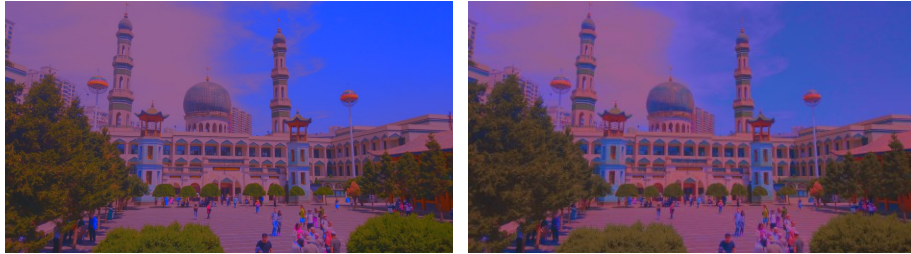


第 3 组:

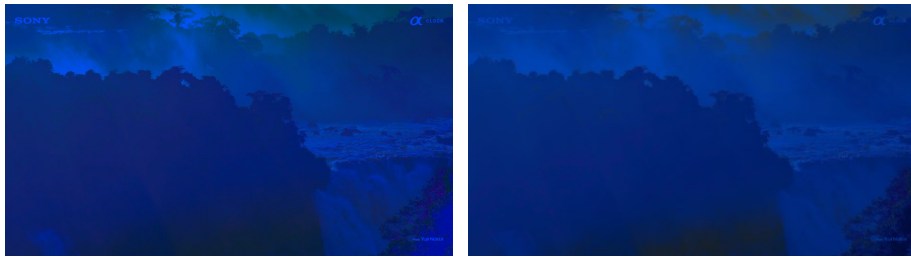




第 4 组:



第 5 组: (可以留意右下角的颜色没有干扰了, 但是总体的对比度低了)



总结地说, 我的方法并不是没有缺陷, 我也发现了不少。对比论文中的方法, 在某些图片上表现得更好, 但又在某些图片上会表现得更差。仍然需要更深入地去研究, 例如可以寻找更好的颜色空间, 或者有更好的实现方法。