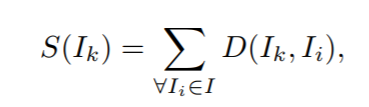
**The English version of this will be written in the future, too busy now…**

作为一个新手，反正我学得很慢。为了把代码研究透。不过好在新的代码都看得懂了，除了一小部分，如vector<CostfIdx> &similari = similar[i];，没太弄懂通过，为啥加个&。

另外，HC的代码主要是从程老师的github里弄来的，不是我自己写的，我只是为了学习的目的，也希望程老师实验室不要追究……当然，非常感谢程老师实验室为saliency领域的蓬勃发展而贡献代码，这也感染了我，我以后有所研究，也会贡献代码！

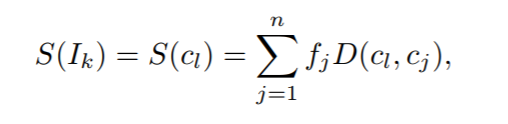
本文的主要目的是，介绍下HC。（写着，写着，发现自己写的好烂……太乱了。）

首先，有这么一种计算显著性的方法：



此方法是基于颜色差异来评估显著性的。就是一个像素，与整个图片所有像素共N方减1个的值作为显著性，当然要进行归一化才叫显著性。

那么问题来了，一个像素与其他像素都做一次diff，也就是计算欧式距离，这复杂度也太大了吧，O(N方)。这要算到猴年马月去？用一个trick，先统计颜色直方图，也就是给相同的颜色聚类，显著性变成了这样：



它的复杂度还有，。如果O(N)<O(n方)，也就是减少颜色的数量，就可以达到线性时间了。24位真彩图颜色有256的立方个。在n方一下，这肯定比像素多多了。

Visual attention detection in video sequences using spatiotemporal cues原文作者，用过Lab来进行降维，首先我没看，但根据程老师的论文里说，Zhai和Shah，他们把L给忽略 ，也就是亮度。这样以来颜色的维度下降了。颜色种类还剩256的平方个。

写的太烂了，说的太细，不如不说。我写的粗一点吧。

减少颜色可以加速。

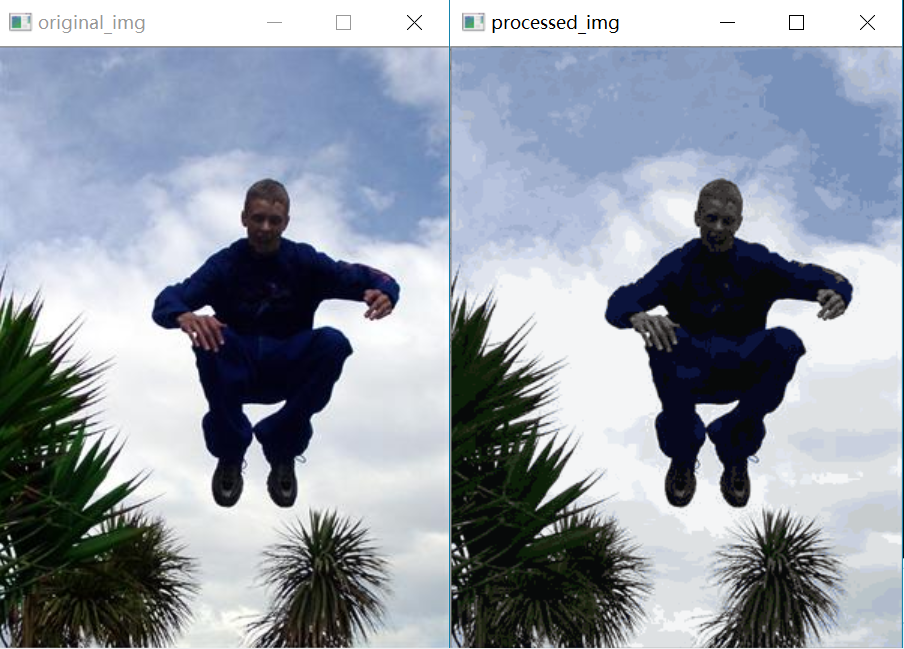
真彩色空间保留颜色信息多。

那就在真彩色空间降维，用12\*12\*12替代256\*256\*256。

有些颜色占得比重小，那就把比重小的都找出来，加在一起总量不超过5%的总像素数。

给这些像素分配到邻近的大比重像素中。

处理后的图像进行了降维，颜色信息还在。具体啥样？



直觉地说，计算显著性，应该变化不大。

但这也有问题，可能一些很近的像素被分配到两个颜色块（bin）中。

就如人脸和手变成了灰色，这是因为人脸和手的颜色被加到天空颜色中并加权做了平均。

怎么处理？局部平均！

用相邻的n/4个颜色的平均差异来计算 。



平均滑处理前 平滑处理后

以上的每步的图片都和程老师的论文结果进行了比较，具有极高的一致性。