

## 只(挚)爱图像处理

一心无二用, 本人只专注于基础图像算法的实现与优化, 如图像增强、滤镜、分割、解码编码等, 无心恋及图像识别。

[博客园](#) [首页](#) [博问](#) [闪存](#) [新随笔](#) [管理](#) [码等](#) [无心恋及图](#) [posts - 79, comments - 53, trackbacks - 0](#)

### 基于灰度世界、完美反射、动态阈值等图像自动白平衡算法的原理、实现及效果

白平衡是电视摄像领域一个非常重要的概念, 通过它可以解决色彩还原和色调处理的一系列问题。白平衡是随着电子影像再现色彩真实而产生的, 在专业摄像领域白平衡应用的较早, 现在家用电子产品(家用摄像机、数码照相机)中也广泛地使用, 然而技术的发展使得白平衡调整变得越来越简单容易, 但许多使用者还不甚了解白平衡的工作原理, 理解上存在诸多误区。它是实现摄像机图像能精确反映被摄物的色彩状况, 有手动白平衡和自动白平衡等方式, 本文简要的介绍了几种自动白平衡算法。

#### 一、原始的灰度世界算法

灰度世界算法 (Gray World)是以灰度世界假设为基础的,该假设认为对于一幅有着大量色彩变化的图像, R、G、B 三个分量的平均值趋于同一个灰度K。一般有两种方法来确定该灰度。

(1)直接给定为固定值, 取其各通道最大值的一半, 即取为127或128;

(2)令  $K = (R_{aver} + G_{aver} + B_{aver})/3$ , 其中  $R_{aver}$ ,  $G_{aver}$ ,  $B_{aver}$  分别表示红、绿、蓝三个通道的平均值。

算法的第二步是分别计算各通道的增益:

$$K_r = K / R_{aver};$$

$$K_g = K / G_{aver};$$

$$K_b = K / B_{aver};$$

算法第三步为根据Von Kries 对角模型, 对于图像中的每个像素R、G、B, 计算其结果值:

$$R_{new} = R * K_r;$$

$$G_{new} = G * K_g;$$

$$B_{new} = B * K_b;$$

对于上式, 计算中可能会存在溢出 (>255, 不会出现小于0的)现象, 处理方式有两种。

a、直接将像素设置为255, 这可能会造成图像整体偏白。

QQ:33184777

群:30417248/下载博客中文档。

Email: laviewpbt@sina.com

格言: 人一定要靠自己。

开发语言: VB6、C、C++、C#

开发特长: 图像算法实现和优化。

短期目标: 自由状态。

代表软件: [Imageshop](#)

昵称: [Imageshop](#)

园龄: [4年8个月](#)

粉丝: [827](#)

关注: [1](#)

[+加关注](#)

#### 随笔分类(113)

[\[01\] 图像增强算法\(26\)](#)  
[\[02\] 图像特效滤镜\(14\)](#)  
[\[03\] 图像算法优化\(13\)](#)  
[\[04\] 图像识别相关\(14\)](#)  
[\[05\] 图像去雾技术\(10\)](#)  
[\[06\] 图像直方图处理\(5\)](#)  
[\[07\] 双边滤波磨皮\(6\)](#)  
[\[08\] 图像二值相关\(5\)](#)  
[\[09\] 图像颜色空间\(6\)](#)  
[\[10\] 图像编码解码\(7\)](#)  
[\[11\] 图像技术杂谈\(3\)](#)  
[\[12\] 图像算法资源\(3\)](#)  
[\[13\] 图像抠图技术\(1\)](#)

#### 随笔档案(79)

[2015年12月\(1\)](#)  
[2015年11月\(1\)](#)  
[2015年10月\(1\)](#)

b、计算所有 $R_{new}$ 、 $G_{new}$ 、 $B_{new}$ 的最大值，然后利用该最大值将计算后数据重新线性映射到[0,255]内。实践证明这种方式将会使图像整体偏暗，建议采用第一种方案。

一般来说，灰度世界算法的效果还是比较好的呢，并且该算法的执行速度非常快之快，目前也存在了不少对该算法进行改进的效果，有时间我在整理一下。



[2015年8月\(2\)](#)  
[2015年7月\(1\)](#)  
[2015年4月\(1\)](#)  
[2015年2月\(1\)](#)  
[2014年11月\(1\)](#)  
[2014年8月\(3\)](#)  
[2014年7月\(4\)](#)  
[2014年6月\(1\)](#)  
[2014年4月\(2\)](#)  
[2014年3月\(3\)](#)  
[2014年2月\(1\)](#)  
[2014年1月\(1\)](#)  
[2013年12月\(4\)](#)  
[2013年11月\(5\)](#)  
[2013年10月\(3\)](#)  
[2013年9月\(6\)](#)  
[2013年8月\(4\)](#)  
[2013年7月\(7\)](#)  
[2013年6月\(4\)](#)  
[2013年5月\(1\)](#)  
[2013年4月\(6\)](#)  
[2013年2月\(3\)](#)  
[2013年1月\(2\)](#)  
[2012年12月\(1\)](#)  
[2012年3月\(1\)](#)  
[2012年2月\(1\)](#)  
[2011年12月\(2\)](#)  
[2011年11月\(5\)](#)

## 文章分类(12)

[调整\(1\)](#)  
[图像模糊\(4\)](#)  
[图像去噪\(4\)](#)  
[增强函数\(3\)](#)

## 文章档案(13)

[2015年8月\(1\)](#)  
[2014年10月\(12\)](#)

## 积分与排名

积分 - 209515  
排名 - 770

## 最新评论

[1. Re:选择性模糊及其算法的实现。](#)

博主好牛逼

--四车半



[2. Re:颜色空间系列3: RGB和YCbCr颜色空间的转换及优化算法](#)  
@Imageshop赞, 不过排序, 好麻烦的...

--rainet

[3. Re:优化的对比度增强算法用于有雾图像的清晰化处理\(算法效果是我目前看到最为稳定的, 且对天空具有天然的免疫力, 极力推荐有需要的朋友研究\)。](#)

@飞机上挂热水壶请问你调通之后出来的效果和原文中的效果一样吗? ...

--LearnAndTry

[4. Re:限制对比度自适应直方图均衡化算法原理、实现及效果](#)

关于裁剪限幅, 我阅读源代码时, 发现是区内直方图Bin上的平均值作为基础值与参数进行乘积进行处理, 但这种处理方法的参数不应该是0~1呀?

--会飞的鸟猪

[5. Re:限制对比度自适应直方图均衡化算法原理、实现及效果](#)

请教下彩色图的处理, 我通过HSI先转成亮度图, 然后通过此方法处理, 最后在HSI逆转回去, 发现色彩不佳; 请问楼主在彩色图处理上有什么好办法?

--会飞的鸟猪

[6. Re:优化的对比度增强算法用于有雾图像的清晰化处理\(算法效果是我目前看到最为稳定的, 且对天空具有天然的免疫力, 极力推荐有需要的朋友研究\)。](#)

@ysywzhm您好, 我也在尝试用MATLAB实现这篇文章的代码, 但是调试时遇到了一些问题, 请问您是否能够共享一下代码让我参考一下, 谢谢! ...

--nigoo

[7. Re:优化的对比度增强算法用于有雾图像的清晰化处理\(算法效果是我目前看到最为稳定的, 且对天空具有天然的免疫力, 极力推荐有需要的朋友研究\)。](#)

@x+y您好, 我也在尝试用MATLAB实现这篇文章的代码, 但是调试时遇到了一些问题, 请问您是否能够共享一下代码让我参考一下, 谢谢! ...

--nigoo

[8. Re:一种可实时处理 O\(1\)复杂度](#)





原图

处理后的图

## 二、完美反射算法

当初写这个代码的时候的一些参考文献一下子也找不到了，就从已经写好的代码中描述下该算法的过程吧。

原理：完美全反射理论perfect Reflector假设图像上最亮点就是白点，并以此白点为参考对图像进行自动白平衡,最亮点定义为R+G+B的最大值,具体编码步骤如下：

- (1) 计算每个像素的R\G\B之和，并保存到一临时内存块中。

[+ View Code](#)

- (2) 按R+G+B值的大小计算出其前10%或其他Ratio的白色参考点的的阈值T。

[图像去雾算法的实现。](#)

“另外一种优化方式就是先计算行方向的平均值”，是累积值吧？

--星夜落尘

[9. Re:基于中值滤波或双边滤波方式的图像去雾效果的研讨。](#)

博主你好，你的代码有个地方我没有看懂在注释：“步骤2：然后考虑到有较好对比度的纹理区域可能没有雾，这部分区域就不需要做去雾处理”的下面\*FilterP = (unsigned char)Diff但是.....

--莲叶tianian

[10. Re:基于灰度世界、完美反射、动态阈值等图像自动白平衡算法的原理、实现及效果](#)

@superglass我也遇到这样的问题，不知道这个Db，Dr足够小的阈值设为多少？而且会出现找不到满足条件的白点的情况，博主你怎么处理的？...

--xiaohui20110101

[11. Re:优化的对比度增强算法用于有雾图像的清晰化处理（算法效果是我目前看到最为稳定的，且对天空具有天然的免疫力，极力推荐有需要的朋友研究）。](#)

您好，请问可以加一下我的QQ吗：1075777878，有些问题想请教您。

--海子呢

[12. Re:基于Fast Bilateral Filtering算法的 High-Dynamic Range\(HDR\) 图像显示技术。](#)

楼主，你好，你做出来的效果挺好的，特别是对普通的偏暗的图像处理，请问楼主，你除了按照你文章开始部分给出的执行部分（伪代码描述，这个在论文的原作者网站也有描述）进行代码实现外，有没有加入一些其它的图像处.....

--superglass

[13. Re:《Single Image Haze Removal Using Dark Channel Prior》一文中图像去雾算法的原理、实现、效果（速度可实时）](#)

五体投地

--xherofly

[14. Re:使用局部标准差实现图像的局部对比度增强算法。](#)

看完博主文章真是 牛逼吊炸天

--ccdoube

[15. Re:十三种基于直方图的图像全](#)

[+ View Code](#)

(3) 遍历图像中的每个点，计算其中R+G+B值大于T的所有点的R\G\B分量的累积和的平均值。

[+ View Code](#)

(4) 对每个点将像素量化到[0,255]之间。

[+ View Code](#)

[局二值化算法原理、实现、代码及效果。](#)

第十个算法，此句：for (MaxValue = 255; MaxValue > MinValue && HistGram[MinValue] == 0; MaxValue--); 其中数组索引是否.....

--hcx0713

[16. Re:十三种基于直方图的图像全局二值化算法原理、实现、代码及效果。](#)

“五、迭代最佳阈值”和“十一：ISODATA(也叫做intermeans法)”实际上是一个算法，请查证！

--quarryman

[17. Re:基于局部均方差相关信息的图像去噪及其在实时磨皮美容算法中的应用。](#)

楼主，我最近在做相机美颜的app，能详细讲一下怎么磨皮美白吗，

--小辉辉+

[18. Re:基于局部均方差相关信息的图像去噪及其在实时磨皮美容算法中的应用。](#)

mac上面能跑不

--刀刀daodao

[19. Re:解析opencv中Box Filter的实现并提出进一步加速的方案（源码共享）。](#)

之前在做ace的时候，切身地体验过boxfilter和gaussfilter之前的速度差别。感觉现在opencv的实现已经不错了，楼主能够更进一步，有没有考虑直接提交到opencv里面去？这样岂不是给.....

--jsxyhelu

[20. Re:解析opencv中Box Filter的实现并提出进一步加速的方案（源码共享）。](#)

boxfilter的运用场合的确很多，楼主能够对算法精细地分析非常厉害。研究以后会谈一谈自己的想法。

--jsxyhelu

[21. Re:解析opencv中Box Filter的实现并提出进一步加速的方案（源码共享）。](#)

好图

--Ender.Lu

[22. Re:基于局部均方差相关信息的图像去噪及其在实时磨皮美容算法中的应用。](#)





博主，你好。我看了你的文章，根据内容自己编码运行了，但是结果好像没有效果，使用均方差处理后磨皮程度要6以上才有磨皮效果。

--zs0817

[23. Re: 《Single Image Haze Removal Using Dark Channel Prior》一文中图像去雾算法的原理、实现、效果（速度可实时）](#)

楼主，速度上的问题可以采取一些弥补措施，比如gpu加速什么的，做应用什么的已经足够了把

--hantics

[24. Re: 《Single Image Haze Removal Using Dark Channel Prior》一文中图像去雾算法的原理、实现、效果（速度可实时）](#)

膜拜膜拜！

--chwise

[25. Re: 颜色空间系列2: RGB和CIELAB颜色空间的转换及优化算法](#)

感谢您提供的分享，现在才发现原来简单的转换算法可以优化的这么好

--jjdbear

## 阅读排行榜

- [1. 《Single Image Haze Removal Using Dark Channel Prior》一文中图像去雾算法的原理、实现、效果（速度可实时）\(25751\)](#)
- [2. 限制对比度自适应直方图均衡化算法原理、实现及效果\(20306\)](#)
- [3. 带色彩恢复的多尺度视网膜增强算法（MSRCR）的原理、实现及应用。\(11449\)](#)
- [4. 一年去雾算法研究的总结。\(10213\)](#)
- [5. 一种具有细节保留功能的磨皮算法。\(9338\)](#)
- [6. 收集的网络上大型的开源图像处理软件代码（提供下载链接\)\(8271\)](#)
- [7. 优化的对比度增强算法用于有雾图像的清晰化处理（算法效果是我目前看到最为稳定的，且对天空具有天然的免疫力，极力推荐有需要的朋友研究）。\(7377\)](#)
- [8. 调整图像- 自动对比度、自动色阶算法\(6814\)](#)
- [9. 局部自适应自动色阶/对比度算法在图像增强上的应用。\(6667\)](#)



原图

Ratio=10%

Ratio=2%

从效果上看,该算法应该比灰度世界的效果要好些,但是也还是受到Ratio这个参数的影像。特别是第二个图片,过高的Ratio导致图片过于泛白。这个问题可以还是最后量化的哪一步引起的,我会抽空再研究一下其他的量化方式,尽量降低Ratio的影响。

针对上述的第二步,看到很多matlab和VC的代码,有很多人居然先用快速排序对累加后的数据进行排序,然后再取其10%的最大值,对图像的数据进行排序,可能就是再快速的排序都快不起来吧,看到这,也许全国人民都笑了。

### 三、动态阈值算法

参考论文: [A Novel Automatic White Balance Method For Digital Still Cameras](#)

同经典的一些算法相同,算法分为两个步骤:白点检测和白点调整。

白点检测:

(1) 为了增强算法的鲁棒性,原文将图像分成12部分,其中宽高比为4:3,关于这一点,我认为不合理,对图像不是通用的,后文再说。

(2) 计算每个区域的Cb\Cr分量的平均值 $M_b/M_r$ 。

(3) 按下式计算每个区域的Cb\Cr分量的绝对差的累积值 $D_b/D_r$ :

$$D_b = \sum_{i,j} (|C_b(i,j) - M_b|) / N$$

[10. 一种可实时处理 O\(1\)复杂度图像去雾算法的实现。\(6598\)](#)

[11. 图像抠图算法学习 - Shared Sampling for Real-Time Alpha Matting\(6423\)](#)

[12. 十三种基于直方图的图像全局二值化算法原理、实现、代码及效果。\(6351\)](#)

[13. 基于灰度世界、完美反射、动态阈值等图像自动白平衡算法的原理、实现及效果\(6043\)](#)

[14. 颜色空间系列1: RGB和CIEXYZ颜色空间的转换及相关优化\(5965\)](#)

[15. 双指数边缘平滑滤波器用于磨皮算法的尝试。\(5549\)](#)

[16. Unsharp Mask\(USM\)锐化算法的原理及其实现。\(5433\)](#)

[17. 颜色空间系列2: RGB和CIELAB颜色空间的转换及优化算法\(5315\)](#)

[18. 避免图像去雾算法中让天空部分出现过增强的一种简易方法。\(4884\)](#)

[19. Wellner 自适应阈值二值化算法\(4754\)](#)

[20. 基于局部均方差相关信息的图像去噪及其在实时磨皮美容算法中的应用。\(4669\)](#)

### 评论排行榜

[1. 《Single Image Haze Removal Using Dark Channel Prior》一文中国图像去雾算法的原理、实现、效果\(速度可实时\)\(53\)](#)

[2. 限制对比度自适应直方图均衡化算法原理、实现及效果\(38\)](#)

[3. 一种具有细节保留功能的磨皮算法。\(33\)](#)

[4. 基于灰度世界、完美反射、动态阈值等图像自动白平衡算法的原理、实现及效果\(31\)](#)

[5. 优化的对比度增强算法用于有雾图像的清晰化处理\(算法效果是我目前看到最为稳定的,且对天空具有天然的免疫力,极力推荐有需要的朋友研究\)。\(21\)](#)

[6. 一种可实时处理 O\(1\)复杂度图像去雾算法的实现。\(21\)](#)

[7. 带色彩恢复的多尺度视网膜增强算法\(MSRCR\)的原理、实现及应用。\(20\)](#)



$$D_r = \sum_{i,j} (|C_r(i,j) - M_r|) / N$$

上式中N为每个区域的像素数。

(4) 如果 $D_b/D_r$ 的值偏小, 则我们忽略这一块, 因为这表明这一块的颜色分布比较均匀, 而这样的局部对于白平衡不好。这个偏小的准则我们稍微再谈。

(5) 统计对于除了符合第四条的的其他区域的 $M_b/M_r/D_b/D_r$ 的平均值作为整幅图像的 $M_b/M_r/D_b/D_r$ 值。

关于这一条, 原文的话是: The final  $M_b$ 、 $M_r$ 、 $D_b$ 、 $D_r$  are obtained by taking the average of those regions that pass this additional step.

我在实际中做的时候就是分别对每块进行的, 似乎效果也还不错。

(6) 按下述规则初步确定哪些点是属于白色参考点:

$$|C_b(i,j) - (M_b + D_b \times \text{sign}(M_b))| < 1.5 \times D_b$$

$$|C_r(i,j) - (1.5 \times M_r + D_r \times \text{sign}(M_r))| < 1.5 \times D_r$$

(7) 对于初步判断已经属于白色参考点的像素, 按大小取其亮度值为前10%的位最终确定的白色参考点。

白点调整:

(1) 计算白色参考点亮度值的平均值 $R_{\text{aver}}, G_{\text{aver}}, B_{\text{aver}}$  (各通道分开计算)。

(2) 按照以下各式计算每个通道的增益:

$$R_{\text{gain}} = Y_{\text{max}} / R_{\text{avew}}$$

$$G_{\text{gain}} = Y_{\text{max}} / G_{\text{avew}}$$

$$B_{\text{gain}} = Y_{\text{max}} / B_{\text{avew}}$$

式中,  $Y_{\text{max}}$ 就是YCbCr颜色空间中Y分量的在整幅图像中的最大值。

(3) 按照以下各式计算最终每个通道的颜色值:

$$R' = R \times R_{\text{gain}}$$

$$G' = G \times G_{\text{gain}}$$

8. Crimm Imageshop 2.3。(18)
9. 一年去雾算法研究的总结。(15)
10. 对比度保留之彩色图像去色算法--基础算法也可以上档次。(13)
11. 使用局部标准差实现图像的局部对比度增强算法。(13)
12. 对皮肤美白算法的一些研究。(12)
13. 利用联合双边滤波或引导滤波进行升采样(Upsampling)技术提高一些耗时算法的速度。(12)
14. 收集的网络上大型的开源图像处理软件代码 (提供下载链接)(11)
15. 基于中值滤波或双边滤波方式的图像去雾效果的研讨。(10)
16. 基于Fast Bilateral Filtering 算法的 High-Dynamic Range(HDR) 图像显示技术。(9)
17. 十三种基于直方图的图像全局二值化算法原理、实现、代码及效果。(9)
18. 肤色检测算法 - 基于不同颜色空间简单区域划分的皮肤检测算法(9)
19. 图像偏色检测算法, 速度快, 效果好, 分享给大家。(8)
20. 二值图像中封闭孔洞的高效填充算法 (附源码)。(8)

## 推荐排行榜

1. 一种可实时处理 O(1)复杂度图像去雾算法的实现。(31)
2. 一种具有细节保留功能的磨皮算法。(28)
3. 《Single Image Haze Removal Using Dark Channel Prior》一文中图像去雾算法的原理、实现、效果 (速度可实时) (22)
4. 优化的对比度增强算法用于有雾图像的清晰化处理 (算法效果是我目前看到最为稳定的, 且对天空具有天然的免疫力, 极力推荐有需要的朋友研究)。(17)
5. 带色彩恢复的多尺度视网膜增强算法 (MSRCR) 的原理、实现及应用。(17)
6. 对比度保留之彩色图像去色算法--基础算法也可以上档次。(16)
7. 图像抠图算法学习 - Shared Sampling for Real-Time Alpha Matting(12)
8. 收集的网络上大型的开源图像处



$$B' = B \times B_{gain}$$

其中R/G/B为在原始的颜色空间中的值，注意这里要进行溢出检测的。

简单的谈下白点检测的分块操作吧，原文把图像分成4\*3的12块，这样做事针对于我们很多数码照片是这个比例的，如果通用，我觉得应该用每个块的大小来控制，比如每块为 100\*100个像素。

这个算法的效果如下：



[理软件代码（提供下载链接）\(12\)](#)

[9. 自己编码使用去色、曲线、色阶算法实现照片怀旧特效。\(11\)](#)

[10. 图像偏色检测算法，速度快，效果好，分享给大家。\(10\)](#)

[11. 二值图像中封闭孔洞的高效填充算法（附源码）。\(9\)](#)

[12. 图像纹理合成及纹理传输算法学习（附源码）。\(9\)](#)

[13. 基于局部均方差相关信息的图像去噪及其在实时磨皮美容算法中的应用。\(9\)](#)

[14. 基于中值滤波或双边滤波方式的图像去雾效果的研讨。\(9\)](#)

[15. 24位真彩色图像转换为16位高彩色图像的实现方法及效果改进\(9\)](#)

[16. 一年去雾算法研究的总结。\(9\)](#)

[17. 对皮肤美白算法的一些研究。\(8\)](#)

[18. 基于Fast Bilateral Filtering 算法的 High-Dynamic Range\(HDR\) 图像显示技术。\(8\)](#)

[19. o\(1\)复杂度之双边滤波算法的原理、流程、实现及效果。\(8\)](#)

[20. 双指数边缘平滑滤波器用于磨皮算法的尝试。\(7\)](#)







原图



块大小50\*50

块大小100\*100

上三图表明：1、该算法效果非常好；2、对块大小不太敏感，因此非常适合于自动化操作。

关于RGB到YCbCr的快速转换，可以参考：[颜色空间系列3: RGB和YCbCr颜色空间的转换及优化算法](#)

由于在上述链接的文章中，YCbCr颜色空间已经被转换到[0,255]，因此白色检测部分的第（6）条中的sign函数就不需要了。

同样，提供个编译好的文件给有兴趣研究该算法的朋友看看效果：

<http://files.cnblogs.com/Imageshop/AutoWhiteBalance.zip>

后记：

针对动态阈值法，很多朋友反映如果YCbCr的值量化在0到255之间的话会出现所有的像素都会被式（6）初步判断为白色参考点。这样前期的工作就失去了意义，算法就变成了类似于完美反射算法那了，稍微有点不同的地方就是两者选择两点的准则有所不同。虽然这样做的最终结果还算不错，但确实和论文的本意像违背了，后面经过实践，如果把YCbCr的值量化在-127到128之间，式6中的Sgn同样适用，则初步判断为白点的数会大量的减少，对于同一个图片，同一个参数两个算法的最终的效果比较如下：

更新后的下载链接依旧如下，以增加了修正后的功能。







\*\*\*\*\*作者: laviewpbt 时间: 2013.4.20

联系QQ: 33184777 转载请保留本行信息

\*\*\*\*\*

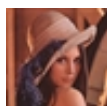
分类: [\[01\] 图像增强算法](#)

标签: [自动白平衡](#), [灰度世界模型](#), [完美反射理论](#), [YCbCr空间动态阈值](#), [white balance](#)

好文要顶

关注我

收藏该文



[Imageshop](#)

[关注 - 1](#)

[粉丝 - 827](#)

[+ 加关注](#)

1

0

(请您对文章做出评价)

« 上一篇: [基于色彩恒常 \( color constancy \) 特性的Frankle-McCann Retinex图像增强。](#)

» 下一篇: [Wellner 自适应阈值二值化算法](#)

posted on 2013-04-20 11:03 [Imageshop](#) 阅读(6043) 评论(31) [编辑](#) [收藏](#)

## FeedBack:

### #1楼

2013-04-20 14:10 | [恶盈好谦](#)

够专业，有点不够通俗。自己查了白平衡才知道一些来龙去脉，以及应用的场景。

<http://baike.baidu.com/view/860.htm>

支持(0) 反对(0)

### #2楼

2013-05-30 15:13 | [superglass](#)

楼主，您好，文章很精彩！

(4) 如果 $D_b/D_r$ 的值偏小，则我们忽略这一块，因为这表明这一块的颜色分布比较均匀，而这样的局部对于白平衡不好。这个偏小的准则我们稍微再谈

如图，动态阈值法中的第（4）小点， $D_b/D_r$ 的偏小准则是怎么样的呢？

支持(1) 反对(0)

### #3楼[楼主]

2013-05-30 15:28 | [Imageshop](#)

@ [superglass](#)

第四条对结果影响不大，我在实际编码中删除了这一条。

支持(0) 反对(0)

### #4楼

2013-05-31 11:00 | [superglass](#)

@ [Imageshop](#)

您好！我现在也在实现动态阈值的算法，但发现符合第（6）小点的像素点个数为0？另外，RGB转YCbCr的公式不是很确定，得到的Cb和Cr的值基本在128附近希望能得到你的一些帮助，谢谢！

支持(0) 反对(0)



#5楼[楼主]

2013-05-31 20:18 | [Imageshop](#)

@ superglass

关于RGB转YCbCr的公式我用的

是<http://www.cnblogs.com/Imageshop/archive/2013/02/14/2911309.html>

我这篇文章的转换算法吧，转换后的YCbCr各分量的范围还是在0到255之间。

你说满足你第6条的像素点数为0 应该还是你代码有问题。

支持(0) 反对(0)

#6楼

2013-07-15 14:26 | [飞飞飞鱼](#)

第二种方法中的maxvalue分别对应rgb三个颜色的最大值么？

支持(0) 反对(0)

#7楼[楼主]

2013-07-15 17:26 | [Imageshop](#)

@ 飞飞飞鱼

我实际上就是定义的MaxValue =255.

支持(0) 反对(0)

#8楼

2013-07-26 00:14 | [yyyy123](#)

@superglass

这算法我也在实现中，第六步所检测出来的候补白点个数一样是零

我是用openCV2实现的（摘自ICVS2013： Will drop C API support soon.  
C++ is much better）

这段程式只做到第五步，为了方便除错我简化了原本的代码  
改为直接对整张图操作

代码 放在pastebin上

<http://pastebin.com/46GMfqbk>

执行结果是

```
cr criteria = 1335.03, 16.8539
```

```
cb criteria = 2317.13, 20.6839
```

cb, cr的值介于0~255之间, 根本不可能满足第六步的判断条件

```
|cb(i, j) - 2317.13| < 20.6839
```

```
|cr(i, j) - 1335.03| < 16.8539
```

前两个我已经实现了, 就剩这一个无法做到, 问题不知道是出在哪里?

cb\_left\_criteria和cr\_left\_criteria的数值实在是大得很诡异

希望楼主可以不吝赐教, 感激不尽

支持(0) 反对(0)

#9楼[楼主]

2013-07-26 09:01 | [Imageshop](#)

@ yyy123

<http://pastebin.com/46GMfqbk> 打不开。

支持(0) 反对(0)

#10楼

2013-07-26 09:03 | [yyy123](#)

可能是被挡了? 我这里能打开

```
1  #include <cmath>
2  #include <iostream>
3
4  #include <opencv2/core/core.hpp>
5  #include <opencv2/imgproc/imgproc.hpp>
6  #include <opencv2/highgui/highgui.hpp>
7
8  int main()
9  {
10     cv::Mat input = cv::imread("/Users/yyy/Downloads/phot
11
12     if(input.empty()){
13         std::cerr<<"can't read image"<<std::endl;
14         return -1;
15     }
16
17     if(input.channels() == 4){
```

```
18         cv::cvtColor(input, input, CV_BGRA2BGR); //去掉a1
19     }
20     cv::cvtColor(input, input, CV_BGR2YCrCb); //转换到YCr
21                                     //转换后的范
22     cv::Scalar const global_mean = cv::mean(input);
23     double dr = 0;
24     double db = 0;
25     for(int i = 0; i != input.rows; ++i){
26         uchar const *ptr = input.ptr<uchar>(i) + 1; //to
27         for(int j = 0; j != input.cols; ++j){
28             dr += std::abs(*ptr - global_mean[1]); ++ptr
29             db += std::abs(*ptr - global_mean[2]); ptr +
30         }
31     }
32     dr /= input.total();
33     db /= input.total();
34
35     //公式 (3) 与 (4) 的判断条件
36     double const cr_left_criteria = 1.5 * global_mean[1]
37     double const cr_right_criteria = 1.5 * dr;
38     double const cb_left_criteria = global_mean[2] + db
39     double const cb_right_criteria = 1.5 * db;
40
41     std::cout<<"cr criteria = "<<cr_left_criteria<<"<
42     std::cout<<"cb criteria = "<<cb_left_criteria<<"<
43
44     return 0;
45 }
46
47
```

支持(0) 反对(0)

#11楼[楼主]

2013-07-26 09:18 | [Imageshop](#)

@ yyy123

我没用过OPENCV，不知道你里面用的那些函数比如之类的具体的input.total是否正确或者说是否真是我们想要的。

对那副散步图，请你贴出你代码中的db,dr,global\_mean[1],global\_mean[2]的具体值。

支持(0) 反对(0)



[#12楼](#)2013-07-26 09:36 | [yyyy123](#) $\text{total} = 161210$  $\text{dr} = 11.236, \text{db} = 13.7893$  $\text{global\_mean}[1] = 104.824, \text{global\_mean}[2] = 156.676$ 

total : 散步图的像素总数

dr, db : 论文中的average absolute difference (dr对应到cr,db对应到cb)

global\_mean[1],global\_mean[2] : cr和cb通道的平均值([1]对应到cr,[2]对应到cb)

支持(0) 反对(0)

[#13楼](#) [楼主]2013-07-26 09:58 | [Imageshop](#)

式6中的绝对值应该去掉。

详见这里<http://www.cnblogs.com/haar/articles/1392227.html>。

支持(0) 反对(0)

[#14楼](#)2013-07-26 11:52 | [yyyy123](#)

谢谢楼主的帮助,我仔细的观

看<http://www.cnblogs.com/haar/articles/1392227.html>后

发现了几点疑问

(1) : 论文 (A Novel Automatic White Balance Method For Digital Still Cameras) 中的Db,Dr应该不是方差吧?

(2) : 第六步去掉绝对值,根据我之前的代码给的数据

Dr,Db为average absolute difference的情况

 $\text{satisfy number} = 161210$  $\text{total} = 161210$  $\text{dr} = 11.236, \text{db} = 13.7893$  $\text{global\_mean}[1] = 104.824, \text{global\_mean}[2] = 156.676$  $1.5 * \text{Mr} + \text{Dr} * \text{sign}(\text{Mr}) = 1335.03, 1.5 * \text{Dr} = 16.8539$  $\text{Mb} + \text{Db} * \text{sign}(\text{Mb}) = 2317.13, 1.5 * \text{Db} = 20.6839$ 

若是Dr,Db取方差,数据则为

 $\text{dr} = 143.875, \text{db} = 235.285$

```
global_mean[1] = 104.824, global_mean[2] = 156.676
1.5 * Mr + Dr * sign(Mr) = 15238.8, 1.5 * Dr = 215.813
Mb + Db * sign(Mb) = 37020.2, 1.5 * Db = 352.928
```

这一来，整张图的pixels都会符合条件

事后的取10%的动作，其实就跟perfect reflector相差无几了

代碼(以容易閱讀，除錯為設計原則)

[+ View Code](#)

支持(0) 反对(0)

#15楼[楼主]

2013-07-26 15:12 | [Imageshop](#)

@ yyy123

你说的是，但是效果还是不同的，主要是一个是利用了Y分量前10%的，一个是R+G+B分量前 10%的。

支持(0) 反对(0)

#16楼

2013-07-26 18:50 | [yyy123](#)

谢谢，这是第二篇我读过，实作后发现效果与数据不相符的论文，巧的是这两篇都是台湾人发出来的论文，都能在IEEE上找到。第一篇是我的期末作业，尝试实现一篇论文上的演算法，该演算法的假设前提很怪（教授亦同意这篇论文的理论很怪），实作出来的结果果然不如预期，教授只是笑笑，这种事情他大概看多了。

这第二篇是基于个人兴趣而实作的，反正演算法不会很复杂，测试的代码在半小时内就能写出来。这篇论文读起来头头是道，实际执行时却发现根本无法满足第六步。第六步是怎么推理出来的我们不清楚，根据论文所述，应该是类似gray world的经验法则。

支持(0) 反对(0)

#17楼[楼主]

2013-07-26 21:39 | [Imageshop](#)

@ yyy123

你说的问题我已经修正了。见文章尾部的后记。

支持(0) 反对(0)

#18楼

2013-07-26 23:02 | [yyyy123](#)

引用

如果把YCbCr的值量化在-127到128之间，式6中的Sgn同样适用

楼主是否能提供一下你使用的转换公式？

这是我使用的

```
1  input.convertTo(input, CV_16SC3);
2      typedef short TYPE;
3      //a simple way to process all of the three channels
4      OCV::process_three_channels<TYPE>(input, [] (TYPE &b,
5      {
6          TYPE y = r * 0.299 - g * 0.168935 + 0.499813 * b
7          TYPE cr = r * 0.587 - g * 0.331665 - 0.418531 *
8          TYPE cb = r * 0.114 - g * 0.50059 - 0.081282 * b
9          b = y;
10         g = cr;
11         r = cb;
12     });
13 }
```

执行结果候补白点确实少了很多

我之前的代码有错，没有把sign算进去

```
1  template<typename T>
2  inline T sign(T const &input){
3      return input >= 0 ? 1 : -1;
4  }
5
6  double const cr_left_criteria = 1.5 * global_mean[1] + dr
7      double const cr_right_criteria = 1.5 * dr;
8      double const cb_left_criteria = global_mean[2] + db *
9      double const cb_right_criteria = 1.5 * db;
```

感谢楼主的热心帮助

支持(0) 反对(0)



#19楼

2013-08-22 16:27 | [alicopy](#)

楼主,你好。

对于动态阈值算法第七点的描述有点理解不了。

引用

(7) 对于初步判断已经属于白色参考点的像素, 按大小取其亮度值为前10%的位最终确定的白色参考点。

是不是这样 if ( $Y[cur] > Y_{max} * 10\%$ ) //cur is final white point?

支持(0) 反对(0)

#20楼[楼主]

2013-08-23 11:37 | [Imageshop](#)

@ alicopy

不知道你的Y数组里面是什么数据。

支持(0) 反对(0)

#21楼

2013-08-23 11:43 | [alicopy](#)

@ Imageshop

我的Y数组存储的是满足条件(6)式的像素点的亮度值(Y分量)。

支持(0) 反对(0)

#22楼[楼主]

2013-08-23 13:15 | [Imageshop](#)

@ alicopy

对Y数组进行直方图统计, 然后从255向0方向计算直方图的累计值, 当累加值第一次大于初步符合白色参考点总数的10%时, 停止累计, 以此值为临界点, 亮度大于其值的则为最终的白色参考点。

支持(0) 反对(0)

#23楼

2013-08-23 14:27 | [alicopy](#)

@ Imageshop

理解了。

感谢楼主的耐心解答~

支持(0) 反对(0)

[#24楼](#)

2014-02-23 21:33 | [hust\\_hunter](#)

楼主，你好，能让我看看动态阈值白平衡的代码嘛，谢谢啦！

我也是csu毕业的，加你qq加不了~~呜呜

支持(0) 反对(0)

[#25楼](#)[楼主]

2014-02-23 21:44 | [Imageshop](#)

@ [hust\\_hunter](#)

基本上我不提供源代码，我QQ设置了要回答问题的。

支持(0) 反对(0)

[#26楼](#)

2014-02-24 15:30 | [hust\\_hunter](#)

@ [Imageshop](#)

请教一下这个算法真的如论文说的那样，动态阈值法的效果比较好吗？用matlab程序试了之后图像变暗了，是什么原因啊？

支持(0) 反对(0)

[#27楼](#)[楼主]

2014-02-24 15:41 | [Imageshop](#)

@ [hust\\_hunter](#)

论文的效果基本没作假，至于你的程序效果不对，别人无从分析其原因。

支持(0) 反对(0)

[#28楼](#)

2014-02-24 16:24 | [hust\\_hunter](#)

@ [Imageshop](#)

嗯嗯，谢谢楼主，我再研究一下那个matlab程序，楼主是中南大学信息院的吗？看了你自己编的imageshop软件好厉害啊，编程能力强！

支持(0) 反对(0)

#29楼

2014-02-24 21:15 | [hust\\_hunter](#)

@ Imageshop

楼主，你做的时候 没有将图像进行分块吗？是直接计算整个图像的Mb，Db值吗？

[支持\(0\)](#) [反对\(0\)](#)

#30楼

2014-05-23 15:32 | [Hello8](#)

@imageshop，看到楼主对图像算法的精度，感到折服！借此机会想请教下大神，就是对于那种日常色温淡蓝色的场景（如天空）和高色温的白色场景，如何来进行区分，或者说有没有稍微好一点的方法？期待楼主的回复。

[支持\(0\)](#) [反对\(0\)](#)

#31楼

2016-04-07 17:39 | [xiaohui20110101](#)

@ superglass

我也遇到这样的问题，不知道这个Db，Dr足够小的阈值设为多少？而且会出现找不到满足条件的白点的情况，博主你怎么处理的？

[支持\(0\)](#) [反对\(0\)](#)[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

【推荐】[50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库](#)

【推荐】[融云即时通讯云 - 豆果美食、Faceu等亿级APP都在用](#)

【推荐】[报表开发有捷径：快速设计轻松集成，数据可视化和交互](#)

【推荐】[一个月仅用630元赚取15000元，学会投资](#)





#### 最新IT新闻：

- [Edge浏览器重磅升级：支持广告屏蔽插件](#)
  - [1499元！红米Pro 8月6日首发开卖：足量现货](#)
  - [美国专利局本周批准近3900项专利 这5项最有趣](#)
  - [小扎进攻性酷似盖茨：无法收购的初创公司直接碾碎](#)
  - [滴滴收购优步中国后 老外觉得Uber在中国打了一场败仗](#)
- » [更多新闻...](#)



#### 最新知识库文章：

- [可是姑娘，你为什么要编程呢？](#)
  - [知其所以然（以算法学习为例）](#)
  - [如何给变量取个简短且无歧义的名字](#)
  - [编程的智慧](#)
  - [写给初学前端工程师的一封信](#)
- » [更多知识库文章...](#)

Copyright ©2016 Imageshop Powered By [博客园](#) 模板提供： [沪江博客](#)