

一种新型的快速色彩平衡算法

王朝辉, 尚 华

(武汉科技大学 计算机科学与技术学院, 湖北 武汉 430065)

摘要: 在图像处理中, 为了提高传统色彩平衡算法的计算速度、降低算法的复杂度, 提出了一种全新的快速色彩平衡算法, 即过滤少量极端像素颜色值并按比例提高剩余的非极端像素颜色值的方法。通过对多幅图片的处理, 相对于传统的色彩平衡方法, 该算法得到了更好的效果、具有更好的性能。

关键词: 色彩平衡; 排序算法; 直方图算法; 白平衡

中图分类号: TP391.41

文献标识码: A

文章编号: 1674-7720(2010)19-0037-02

A novel fast color balance algorithm

WANG Zhao Hui, SHANG Hua

(College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China)

Abstract: This paper proposed a novel fast color balance algorithm in order to enhance the computation rate of traditional color balance algorithm and reduce the complexity of algorithm. The algorithm could filter a small amount of pixels with extreme color values and then according to the scales increased the remaining pixels with non-extreme color values. After processing the multiple images, comparing to the traditional color balance algorithms, the proposed algorithm obtains a better result and performance.

Key words: color balance; sorting algorithm; histogram algorithm; white balance

在图像采集的过程中, 由于不同光照下获取的图片颜色值差异较大, 对图片的显示及图片的分析产生困难。因此, 在摄影和图像处理中, 不少学者提出了通过色彩平衡来解决这个难点。色彩平衡是一种通过全局调整图像的 RGB 颜色信息使得图片中出现色偏的颜色正常化的方法。学术界中提出的较普遍的算法有灰平衡、中性平衡或白平衡算法^[1-3]。图像处理过程中, 假设原来图像中某些像素点是白色的, 但由于图像获取过程中由于光线照射等原因产生色偏现象, 使得这些像素点的 R、G、B 值不再保持相同。WANG^[4]提出了白平衡算法, 即通过调整 R、G、B 三个分量的值使之达到平衡, 然后按照这个调整比例对整幅图像进行调整, 达到色彩平衡的效果。LEE Hsien Che^[5]提出的算法原理是在 CCD 或 CMOS 矩阵表示的原 RGB 颜色值下将其转换为更加标准的感性颜色。

本文提出了一种新型的快速色彩平衡算法, 首先取一个最大值 V_{\max} 和一个最小值 V_{\min} , 把小于 V_{\min} 的值全

部改成 V_{\min} , 把大于 V_{\max} 的值全部改成 V_{\max} 。执行过程中, 首先需要优化一些极端值, 优化的极端值数量由输入的优化率 s 决定; 再根据 s 为整幅图像分别确定一个合适的最大值 V_{\max} 和最小值 V_{\min} , 颜色值在 $[V_{\min}, V_{\max}]$ 范围外的像素即为极端像素, 将极端像素的值改在 $[V_{\min}, V_{\max}]$ 范围内, 最后再根据 V_{\min} 、 V_{\max} 和 s 按比例来提高非极端像素的颜色值。

1 新型色彩平衡算法

本文中算法的理论基础是在 RGB 模式中颜色值越高越接近白色, 颜色值越低越接近黑色。如果图像是在黑暗中拍摄的, 其像素最高的颜色值一定小于 255, 则通过扩大颜色值的范围, 图像就会变亮。例如, 闪电的主要颜色是 R 和 G, 那么通过色彩平衡就要增强 B, 这样会减少一些微黄色色调, 图像会更亮一些, 不但增加了画面的和谐性, 而且使图像看起来更自然。

综上所述, 本算法的目的是增加 R、G、B 三个颜色通道的颜色值, 并使它们的范围保持在 $[0, 255]$ 。最简单

的实现方式就是给每个通道 $ax+b$ 的颜色值。其中, a, b 是比例因子, 需要通过计算得到, x 是初始颜色值, $ax+b$ 必须在 $[0, 255]$ 的范围内。

在实际图像中经常会遇到颜色值为 0 或 255 的像素。在本算法中, 需要将这些点稍微优化一下, 因为运用本算法会产生全白或全黑的区域, 但不必担心优化这些点会产生很大的影响。因为同一幅图上颜色相近的点有很多, 只需要将这些点的值改为与它们颜色最接近的点的值就可以了。实际需要优化的点或许很多, 或许只有一个, 不好确定确切的数目, 所以必须用优化率来决定究竟要改变多少个点的值。假设一幅图像上有 N 个像素点, 优化率为 s , 则需要优化 $N \times s$ 个点, 并使 $0 < N \times s < 255$ 。优化率的选取不能太大, 因为需要优化的点通常只占很少一部分。

2 新型色彩平衡算法实现过程

本文介绍了排序和直方图两种改进的算法来实现色彩平衡的过程, 下面阐述两种算法的实现和实用性。

2.1 排序算法

算法流程如下:

(1) 排序

将 N 个像素存入矩阵, 并将它们按颜色值大小从低到高进行排序(排序之前要先将矩阵备份, 因为以后还要用到)。

(2) 确定优化数量和最大最小值

输入优化率 s , 则优化数量为 $N \times s$, 需要优化的点是排序后的矩阵的前 $N \times s / 2$ 和后 $N \times s / 2$ 。因此, V_{\min}, V_{\max} 分别产生在 $N \times s / 2$ 和 $N \times (1 - s / 2) - 1$ 的点。

(3) 优化像素

将颜色值小于 V_{\min} 的全部改为 V_{\min} , 将颜色值大于 V_{\max} 的全部改为 V_{\max} 。

(4) 生成颜色值范围为 $[\min, \max]$ 的新图像

假设新颜色值为 $f(x) = ax + b$ (x 为旧颜色值), 则有:

$$a \times V_{\min} + b = \min \quad (1)$$

$$a \times V_{\max} + b = \max \quad (2)$$

可由式(1)、式(2)推导出:

$$f(x) = (x - V_{\min}) \times (\max - \min) / (V_{\max} - V_{\min}) + \min$$

2.2 直方图算法

N 个像素颜色值排序的时间复杂度是 $O(N \log(N))$ 。还有一种更高效的方法, 即以直方图为基础的改编版本, 最坏情况下其时间复杂度也仅为 $O(N)$ 。

算法流程如下:

(1) 建立一个关于像素颜色值的直方图

横向 i 表示颜色值, 纵向表示 $\leq i$ 的像素数量 q 。

(2) 确定优化数量和最大最小值

根据直方图表示的意义, 柱的高度 $> N \times s / 2$ 的最小颜色值 i , 即为 V_{\min} ; 同理, 柱的高度 $\leq N \times (1 - s / 2)$ 的最大

颜色值 i , 即为 V_{\max} 。如果 $s=0$, 那么 i 的最小值即为 V_{\min} , i 的最大值即为 V_{\max} 。

(3) 优化像素

将颜色值小于 V_{\min} 的全部改为 V_{\min} , 将颜色值 $> V_{\max}$ 的全部改为 V_{\max} 。

(4) 生成颜色值范围为 $[\min, \max]$ 的新图像

假设新颜色值为 $f(x) = ax + b$ (x 为旧颜色值), 则有:

$$a \times V_{\min} + b = \min \quad (3)$$

$$a \times V_{\max} + b = \max \quad (4)$$

可由式(3)、式(4)推导出:

$$f(x) = (x - V_{\min}) \times (\max - \min) / (V_{\max} - V_{\min}) + \min$$

对于 16 位像素值, 直方图算法是非常好的算法, 只需要大约 256 KB 的内存, 得到 V_{\min} 和 V_{\max} 的速度非常快, 其速度可与二分法媲美。

但对于 32 位像素值, 直方图算法大约需要耗费 18 MB 内存。这种情况下, 最好采用排序算法。

3 实验结果

将本文提出的算法用于改善曝光不足或采光条件不好或特殊光线下(如日落)生成的图像的一种算法。该算法可以运用于大部分图像, 即使原图是在正常光线下拍摄的也同样适用。但有一种情况例外: 如果图像是纯色的, 即所有像素含有相同的值 v , 则比 v 大和比 v 小的像素数量都是 0, 不论 s 取何值, $V_{\min} = V_{\max}$, 这种情况下不能使用本算法。

图 1 分别展现了原图分别经优化率为 0%、1%、2% 和 3% 的色彩平衡算法优化后的结果。由图可以看出, 其图像的优化率不同, 优化结果也不同。



图 1 经色彩平衡算法优化结果

与基于 CCD 或 CMOS 矩阵转换原 RGB 颜色为更加标准的感性颜色及白平衡算法等传统的色彩平衡算法不同, 本文提出的新型快速色彩平衡算法, 通过获取一个由优化率 s 决定的区间 $[V_{\min}, V_{\max}]$, 然后将这个区间外的像素点改在 $[V_{\min}, V_{\max}]$ 内, 最后再根据 V_{\min} 、 V_{\max} 和 s 按比例来提高区间 $[V_{\min}, V_{\max}]$ 内像素的颜色值。实验结果表明, 本算法能得到较好的效果, 与其他算法相比更为简明、快速。

参考文献

- [1] MARC E. Color constancy [M]. 111 River Street Hoboken, NJ. John Wiley & Sons, 2007.
- [2] Wikipedia Contributors. Color balance [OL]. Wikipedia, The

Free Encyclopedia.[http://en.wikipedia.org/w/index.php?title = Color_balance&oldid=336334367](http://en.wikipedia.org/w/index.php?title=Color_balance&oldid=336334367).2010.

- [3] JONATHAN S. Color balancing techniques [J]. Digital Light & Color, 2007(1):1-19.
- [4] WANG Y G,GAO N Y. Color balance in color image[J]. Computer and Information Technology, 2009(1):101-102.
- [5] HSIEN C L. Introduction to color imaging science[J]. Cam-

bridge University Press, 2005(1):450.

(收稿日期:2010-06-02)

作者简介:

王朝辉,男,1967年生,博士,教授,主要研究方向:计算机先进过程控制理论与应用等。

尚华,女,1986年生,硕士研究生,主要研究方向:图像处理。

