

VB图像处理之图像的色彩纠正 PDF转换可能丢失图片或格式
，建议阅读原文

https://www.100test.com/kao_ti2020/137/2021_2022_VB_E5_9B_BE_E5_83_8F_E5_A4_c97_137665.htm 先给大家介绍一下一些颜色的小常识。以电脑的颜色来说，颜色的组成是右红、绿、蓝三种颜色组成。以最容易的24位色来说，红色、绿色、蓝色各用1个字节来表示，1个字节有8位，所以加在一起正好是24位。由于电脑无法用连续的模拟值来表示一个自然量，只能将它们分成一段一段来显示，分得越多就越接近自然。1个字节是 $2^8 = 256$ ，所以在24位色中每一个单色都有256种不同的强度，三种颜色按照不同的强度混合，可以得到 2^{24} 种色彩。大约可以表达1677万种色彩，对于人的眼睛来说已经和自然色彩没有区别了。如果我们用PHOTOSHOP来打开一张图片，选择LEVEL工具，就可以察看这张图片的色彩分布了。原图：灰度通道：红色通道：绿色通道：蓝色通道：由上面的四个通道图，我们可以发现，这张图片的色彩都是以低亮度为主，红绿蓝三色在高亮度区域的分布都很少，而灰度通道也表明整张图片的亮度值很低。在前面的文章中，我已经和大家讲过，人眼对于灰度（亮度）的敏感程度最高。因此，如果我们能通过一种方法把这章图片的灰度提高，那么它在我们的视觉中就会有更好的表现。或许有朋友说只要把图片加亮度，不是就可以了吗？不错，通过提高所有色彩的亮度，可以把原来处于第亮区域的色彩转移到中亮度或高亮度区域，但是大家也知道，在算法中，加亮度只是很简单的在R、G、B三个值后面直接加上一个偏移量：

$$\text{NewRed}=\text{OldRed} \text{ Offset}, \text{NewGreen}=\text{OldGreen} \text{ Offset}, \text{NewBlue}$$

Offset,但是这种做法只是“粗暴”地将整个色彩空间搬了一个位置，而没有改变其分布。请看下面，我在这张图片加上了120点亮度：再看看此时的色彩分布你就知道我为什么说它“粗暴”了。灰度通道：红色通道：绿色通道：蓝色通道：可以看到，原图上的高亮度部分的色彩信息全部丢失，而低亮度部分（0 - 120）则是一片空白，如果说原来的图片是 $2^8 \times 2^8 \times 2^8$ 一共1677万色的话，那么现在的图片则是： $(256 - 120)^3 = 251$ 万色，也就是说，通过我们给图片加了120点的亮度，我们丢失了一大半的颜色信息。（选择加120点亮度是为了使图片的整体亮度和直方图均衡后的亮度相当，易于比较。）那么，有什么方法可以在不丢失，或丢失很少的颜色信息的前提下做到是图片显示更好呢？有，这就是今天要说的“灰度直方图均衡”的方法。先给大家看一下效果：下面是将原来的图片通过灰度直方图的方法处理后的效果和前面的比较。原图：提高亮度120：灰度直方图均衡：通过直方图均衡后的颜色分布如下：灰度通道：红色通道：绿色通道：蓝色通道：通过对比，我们可以发现，灰度直方图均衡，是对原图中的色彩分布按照出现的频率作了一个分布。将出现最多的色彩“分”开，将出现很少的色彩“挤”得更紧，这样作的好处就是把我们的眼睛所看到的主体表示得更鲜明了。好了，关于效果，也已经说得很多了，下面我们就来说明一下这个算法是如何实现的。首先，我们需要获取要处理的图片中所有像素的色彩分布统计，也就是上面的几个通道所作那样。假设有一张图（我们直接用灰度来表示）：100 50 20 20 40 50 100 250 200 统计入下：20：2 40：1 50：2 100：2 200：1 250：1 这张图一共有9个像素，我们用比例

来表示每种颜色的出现比例：20 : 2 / 9 40 : 1 / 9 50 : 2 / 9 100 : 2 / 9 200 : 1 / 9 250 : 1 / 9 由于所有的色彩出现的次数不可能超过图片的总像素，因此，将所有色彩的比例相加也不会超过1（大家已经可以看出正好是1）最后我们按照从低到高的顺序，把各个色彩的比例进行加权统计，也就是当前点的“权”等于该点的原有比例加上前一个点的“权”，我们得到一个新的统计表：20 : 2 / 9 40 : 3 / 9 50 : 5 / 9 100 : 7 / 9 200 : 8 / 9 250 : 9 / 9 最后，根据这个新的统计表，我们来把像素的亮度用一个新的亮度来代替，算法为：新亮度 = 该点“权” × 255 20 : 2 / 9 >> 20（第一点不动，依然用20）40 : 3 / 9 × 255 = 85 50 : 5 / 9 × 255 = 141 100 : 7 / 9 × 255 = 198 200 : 8 / 9 × 255 = 226 250 : 9 / 9 × 255 = 255 这时我们得到了新的图：100 50 20 198 141 20 20 40 50 >> 10 85 141 100 250 200 198 255 226 原图中相对出现频率多的部分的宽度变大了。而出现较少的部分则变窄了。所以，灰度直方图均衡的作用就是把一张图片上出现多的色彩拓展，而把出现少的色彩压缩。从而得到了更“均衡”的色彩分布。下面附上我的例程：

```

Private Type
ColorChart ColorCount(255) As Long 统计原来图片中的亮度出现次数
PixelCount As Long 记录图片的像素个数
ColRatio(255) As Single 记录每一个亮度的出现比例
NewVal(255) As Byte 存放新的亮度索引
End Type
Dim ColChart As ColorChart
Public Sub
StatisticsChart()
Dim R As Byte
Dim G As Byte
Dim B As Byte
Dim Gray As Integer
Dim X As Long
Dim Y As Long
Dim I As Long
Dim L As Long
Dim M As Long
Dim C As Double
On Error GoTo
ErrLine
Done = False
TimeFilter = timeGetTime
With ColChart
For X = 0 To 255 先把数组清零
.ColorCount(X) = 0
Next
For X = 0

```

To OutPutWid 这两个循环用来扫描图片数据，记录每个点的灰度和出现次数 For Y = 0 To OutPutHei R = ColVal(2, X, Y) G = ColVal(1, X, Y) B = ColVal(0, X, Y) Gray = R * 3 G * 6 B Gray = Gray \ 10 .ColorCount(Gray) = .ColorCount(Gray) + 1 Next Next .PixelCount = X * Y 获得图片的像素总量 C = 1 / .PixelCount .ColRatio(0) = .ColorCount(M, 0) * C 计算每个亮度的出现比例 .NewVal(0) = 0 色值最小的色彩总是为0，不参与计算 L = 0 For I = 1 To 255 .ColRatio(I) = .ColorCount(I) * C .ColRatio(L) 进行加权 .NewVal(I) = .ColRatio(I) * 255 计算新的颜色索引 L = L + 1 Next For X = 0 To OutPutWid For Y = 0 To OutPutHei R = Colval(2, X, Y) 读取原来点的颜色 G = Colval(1, X, Y) B = Colval(0, X, Y) R = .NewVal(R) 查表得到新的颜色 G = .NewVal(G) B = .NewVal(B) ColOut(2, X, Y) = R 把新的颜色放到输出数组中 ColOut(1, X, Y) = G ColOut(0, X, Y) = B Next Next End With Done = True TimeFilter = timeGetTime - TimeFilter Exit Sub ErrLine: Done = True MsgBox Err.Description End Sub

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com