拓展资源 9.2　补 充 内 容

9.2.1　颜色科学发展简史

最近 100 多年来，为满足各种不同用途的需求，人们已经开发了许多不同名称的颜色空间，尽管几乎所有的颜色空间都是从 RGB 颜色空间导出的，现有的颜色空间也还没有一个完全符合人的视觉感知特性、颜色本身的物理特性或发光物体和光反射物体的特性，但随着科学和技术的进步，人们还在继续开发新的颜色空间。

1．Isaac Newton（1642—1727 年）的色圆

Newton 于 1666 年开始研究颜色，提出了白光包含所有可见光谱的波长，并用棱镜演示了这个事实。同时把红色和紫色首尾相接形成色圆（Color Circle），也称牛顿色圆（Newton color circle），如图 9.1 所示。

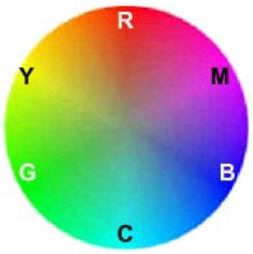

图 9.1　牛顿色圆

在色圆中，圆周表示色调，圆的半径表示饱和度，R 表示红色，G 表示绿色，B 表示蓝色，C 表示青色，M 表示深红，Y 表示黄色，为揭示三原色原理奠定了基础。

2．Thomas Young（1773—1829 年）的假设

在 1802 年，Young 认为人的眼睛有 3 种不同类型的颜色感知接收器，大体上相当于红、绿和蓝 3 种基色的接收器。Hermann von Helmholtz（1821—1894 年）认为 Young 的看法非常重要，使用 3 种基色相加可产生范围很宽的颜色，并把这个想法用于定量研究，因此有时把他们的想法称为 Young-Helmholtz 理论。直到 1965 年前后进行的详细的生理学实验证明，在眼睛中的确存在 3 种不同类型的锥体，证实了 Thomas Young 的假设是正确的。

3．James Clerk Maxwell（1831—1879 年）的色度学

19 世纪 60 年代，Maxwell 探索了 3 种基色的关系。认为 3 种基色相加产生的色调不能覆盖整个感知色调的色域，而使用相减混色产生的色调却可以，并且认为彩色表面的色调和饱和度对眼睛的敏感度比亮度低。Maxwell 的工作被认为是现代色度学的基础。

4．颜色度量方法的确定

20 世纪 20 年代对科学家们提出的理论进行了实验，实验结果表明：红、绿和蓝相加混色的确能产生某个色域里的所有可见颜色，但不能产生所有的光谱色（单一波长的颜色），

尤其是在绿色范围内的光谱色；如果加入一定量的红光，所有颜色都可呈现，并用三色激励值表示 R，G，B 基色，但必须允许红色激励值为负值（即用补色）。1931 年国际照明委员会（CIE）定义了标准颜色体系，规定所有的激励值应该为正值，并用 x 和 y 两个坐标表示所有可见的颜色，并且绘制了 CIE 色度图（CIE chromaticity diagram），该图是用 x, y 平面表示的马蹄形曲线，为大多数定量的颜色度量方法奠定了基础。

9.2.2 颜色的特性

颜色的 3 个特性：色调（Hue）、饱和度（Saturation）、亮度（Intensity）。

色调是视觉系统对一个区域呈现的颜色的感觉，即对可见物体辐射或发射的光波波长的感觉。色调是最容易把颜色区分开的属性，色调用红（Red）、橙（Orange）、黄（Yellow）、绿（Green）、青（Cyan）、蓝（Blue）和紫（Violet）等术语来刻画。色调数目多于 1000 万种，然而普通人只能区分 200 种，颜色专业人士可辨认的色调数目为 300～400 种。在色圆中，圆周表示色调。圆周上的颜色具有相同的饱和度和亮度，但它们的色调不同，如图 9.2 所示。

饱和度表示颜色的纯洁性，用来区别颜色明暗的程度。当一种颜色掺入其他光成分越多时，就说该颜色越不饱和；完全饱和的颜色是指没有渗入白光所呈现的颜色；单一波长的光谱色是完全饱和的颜色。在色圆中，圆的半径表示饱和度。沿半径方向上的颜色具有相同的色调和亮度，但它们的饱和度不同，距离圆心越远，饱和度越深，如图 9.2 所示。

亮度可以认为是光的强度，其含义是单位面积上反射或发射的光的强度，亮度常用垂直轴表示。沿垂直轴上的颜色具有相同的色调和饱和度，但它们的亮度不同：底部的亮度值小，最小值为 0，表示黑色；顶部的亮度值大，最大值为 1，表示白色；中间部分表示灰色，由下向上，灰度值增大，如图 9.2 所示。

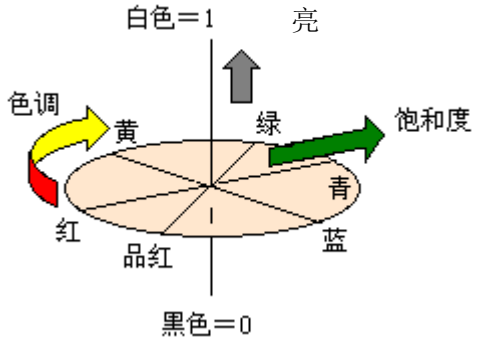

图 9.2　色调的表示

9.2.3 颜色模型的分类

颜色模型（Color Model）和颜色空间（Color Space）表示相同的含义，互为同义词。

1. 从与设备是否相关的角度分类

设备相关是指指定生成的颜色与生成颜色的设备有关。例如，RGB 颜色模型是与显示系

统相关的颜色模型，计算机显示器使用 RGB 来显示颜色，用像素值（例如，*R*=250，*G*=123，*B*=23）生成的颜色将随显示器的亮度和对比度的改变而改变。CMY（Cyan，Magenta，Yellow）或 CMYK（Cyan，Magenta，Yellow，Black）是与彩色打印系统相关的颜色模型。

设备无关是指指定生成的颜色与生成颜色的设备无关。例如，CIE La*b*颜色空间是与设备无关的颜色空间，它建立在 HSV（Hue，Saturation，Value）颜色模型的基础上，用该空间指定的颜色无论在什么设备上生成的颜色都相同。

2. 从颜色感知的角度分类

可以分为混合型颜色空间、非线性亮度—色度型颜色空间和色调—饱和度—强度型颜色空间 3 类。

混合型颜色空间是指按 3 种基色的比例合成颜色，如 RGB，CMY(K)和 XYZ。

非线性亮度—色度型颜色空间是指用一个分量亮度表示非色彩的感知，用两个独立的分量表示色彩的感知。例如，La*b*，Lu*v*，XYZ，YUV 和 YIQ 等就是这类颜色模型。当需要黑白图像时，使用这样的系统就非常方便。其中，La*b*，Lu*v*和 XYZ 等是由国际照明委员会（CIE）定义的颜色空间，是与设备无关的颜色模型。用做颜色的基本度量方法，在科学计算中得到广泛应用。对不能直接相互转换的两个颜色空间，可利用这类颜色空间作为过渡性的颜色空间。YUV 和 YIQ 是由于广播电视需求的推动而开发的颜色空间，是与设备相关的颜色模型。其主要目的是通过压缩色度信息以有效地播送彩色电视图像。

色调—饱和度—强度型颜色空间是指用色调和饱和度描述色彩的感知，可使颜色的解释更直观，而且对消除光亮度的影响很有用，如 HSI 颜色空间和 HSV 颜色空间等。

9.2.4 常用的颜色模型

1. RGB，CMY 和 CMYK 颜色模型

RGB 颜色模型是使用不同数量的红、绿和蓝 3 种基色相加产生颜色。在数字图像处理中，实际中最通用的面向硬件的模型就是 RGB 模型，该模型用于彩色监视器和一大类彩色视频摄像机。

青（Cyan）、深红（Magenta）、黄（Yellow）是光的二次色，它们是颜料的原色。CMY 颜色模型是使用白光中减去不同数量的青、深红和黄 3 种颜色产生的颜色。大多数在纸上沉积彩色颜料的设备，如彩色打印机和复印机，要求输入 CMY 数据或在内部做 RGB 到 CMY 的转换。转换就是执行如下的一个简单操作。

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{9.1}$$

这里假定所有的彩色值都归一化为[0,1]范围。式（9.1）显示了从涂覆青色颜料的表面反射的光不包含红色（即公式中 $C=1-R$）。与此相似，纯深红色不反射绿色，纯黄色不反射蓝色。在图像处理中，这一模型主要用于产生硬复制输出。

等量的颜料原色（青、深红、黄）可以产生黑色。实际上，为打印组合这些颜色产生的黑色是不纯的，因此，为了产生真正的在打印中起主要作用的颜色——黑色，需要加入第 4 种颜色——黑色，于是就提出了 CMYK 彩色模型。CMYK 是将黑色分量加到 CMY 空间形

成的。这样当出版商说到“四色打印”时，是指 CMY 彩色模型的 3 种原色再加上黑色。

2. HSI 和 HSV 颜色模型

在这类颜色模型中，指定颜色方式非常直观，很容易选择所需要的色调，并且已经把亮度从颜色信息中分离出来了。这些特点使得这类模型非常适合于借助人的视觉系统来感知彩色特性的图像处理算法。

在 HIS（Hue，Saturation，Intensity）模型中，色调 H 用角度值（0°～360°）表示，如红、黄、绿、青、蓝、品红的角度值分别为 0°、60°、120°、180°、240° 和 300°。饱和度 S，分成低（0%～20%），产生灰色而不管色调；中（40%～60%），产生柔和的色调；高（80%～100%），产生鲜艳的颜色。强度 I 取值范围从 0%（黑）～100%（最亮）。

HSV（Hue，Saturation，Value）模型是 A. R. Smith 根据颜色的直观特性于 1978 年创建的，也称六角锥体模型（Hexcone Model），如图 9.3 所示。在该模型下，色调用角度度量，变化的范围是 0°～360°。红色为 0°，按逆时针方向计算，绿色为 120°，蓝色为 240°。饱和度的取值范围为 0.0～1.0。亮度值的取值范围为 0.0（黑色）～1.0（白色）。

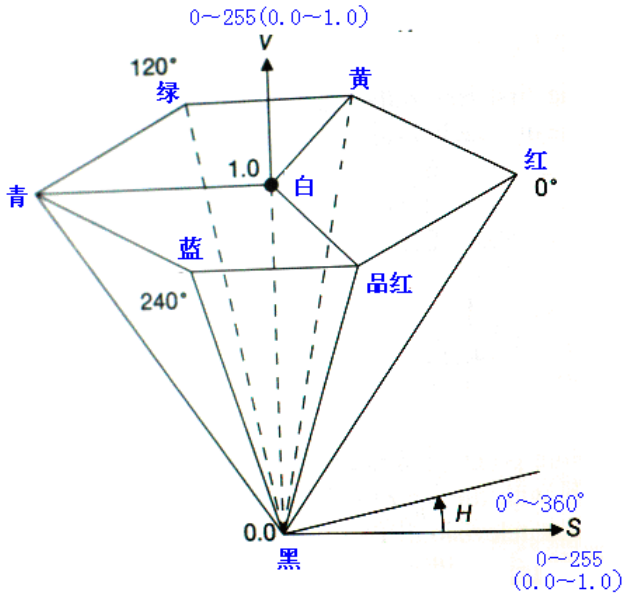

图 9.3　HSV 颜色模型

RGB 空间转换成 HSV 空间的公式如下。

$$H=\begin{cases}\arccos\dfrac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}} & B\leqslant G\\ 2\pi-\arccos\dfrac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}} & B>G\end{cases}$$

$$S=\frac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)}$$

$$V=\frac{\max(R,G,B)}{255}$$

对于 HSV 空间中的两个像素点 $p(h_p,s_p,v_p)$ 和 $c(h_c,s_c,v_c)$，它们之间的距离按照如下公式计算：

$$d(p(h_p,s_p,v_p),c(h_c,s_c,v_c))=\sqrt{(v_p-v_c)^2+(s_p\cos h_p-s_c\cos h_c)^2+(s_p\sin h_p-s_c\sin h_c)^2}$$

9.2.5 色彩平衡

1．色彩平衡原理

色彩平衡算法，一般用来修正曝光不足的图像以及在人造光及特殊自然光下采集的图像。在很多图像中，都可能出现一些像素值接近于 255 或接近于 0 的像素点，这些像素点所在的比例很小，但实际只包含较少的信息量。因此，我们可以去除这些最暗及最亮的灰度级。然后再对余下的灰度级进行重新分配，这样做既有对图像白平衡的作用，同时也可以对图像对比度进行增强。对于 HSV 空间来讲，仅对 V 通道进行处理后，将处理前与处理后的比值分别于 R、G、B 三通道相乘得到平衡结果。图 9.4 是算法的实现步骤流程图（s1 和 s2 是我们希望消除的灰度级的比例，s1 为左边灰度级比例，s2 为右边灰度级比例）。

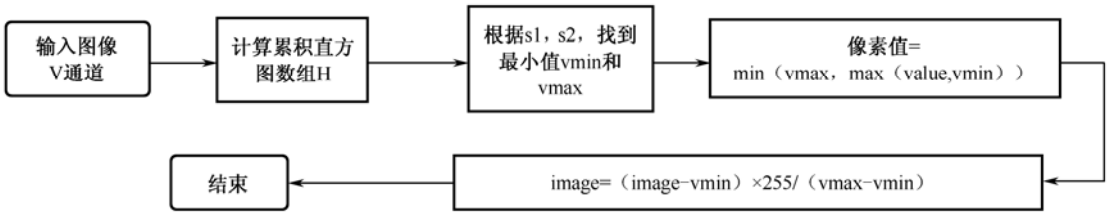

图 9.4　白平衡算法流程图

其中累积直方图存储的是小于等于当前灰度值的像素的个数。若设总像素个数为 N，则 vmin 指的是满足 $H(X)>N\times s1/100$ 的最小值。vmax 指的是满足 $H(X)<N\times(1-s2/100)$ 的最大值。

2．白平衡算法的实现

```
function img = simplestColorBalance(I,s1,s2)
%对图像进行白平衡
%s1:直方图左边的阈值百分比
%s2:直方图右边的阈值百分比
[m,n] = size(I);
Max = max(I(:));
N = m*n;
images = round(reshape(I,[1 N]));%将二维模板变为一维,并对其量化到整数域 0~255
%初始化累计直方图矩阵
histo = zeros(1,Max+1);
```

```
for i=1:N
   histo(images(i)+1) = histo(images(i)+1) + 1;
end
for i=2:Max+1
   histo(i) = histo(i) + histo(i - 1);
end

%找到左边最小值和右边最大值
vmin = 1;
while histo(vmin) < N * s1 / 100
   vmin = vmin + 1;
end
vmin = vmin - 1;
vmax = Max+1;
while histo(vmax) > N * (1 - s2 / 100)
   vmax = vmax - 1;
end
vmax = vmax - 1;
%将像素值量化到[vmin,vmax]之间
images = double(max(vmin,min(vmax,images)));
images = round((images - vmin)*255/(vmax - vmin));
img = reshape(images,[m n]);
End
```

3．实验结果

如果是对 HSV 中的 V 通道进行处理，则在处理后需计算 A=V’/V，处理前后的比值，之后在转换到 RGB 空间中，作 R=A.*R、G=A.*G、B=A.*B 处理。实验结果如图 9.5 所示。

（a）特殊光照下的图像　　（b）白平衡后的图像

图 9.5　特殊光照下与白平衡光照下对比结果图

将第 5 章讨论的模糊增强算法、动态量化与本章讨论的白平衡方法相结合，得到的图像增强实验结果如图 9.6 所示。

（a）原图　　（b）原 FuzzyHE 增强后　　（c）改进型 FuzzyHE 增强后

图 9.6　模糊增强算法和动态量化与白平衡

```
function[original_image,enhance_image]=Fuzzy_HSI(I,alpha,s1,s2)
%I：输入图像文件地址
%alpha: GIMP 量化系数，一般取为 2，处理整体低光照图像时，可增大该值到
%s1,s2 为作白平衡处理的参数，一般都取为 1
IMG = imread(I);
info1=entropy(rgb2gray(IMG));
m=size(IMG,1);
n=size(IMG,2);
%矩阵归一化
R = mat2gray(im2double(IMG(:,:,1)));
G = mat2gray(im2double(IMG(:,:,2)));
B = mat2gray(im2double(IMG(:,:,3)));
%%%%%%%%%%%%转换到 HSV 空间%%%%%%%%%%%%%%%%%
temp1 = min(min(R,G),B);
temp2 = R+G+B;
temp2(temp2==0)=eps;%避免除数为 0
I = (R+G+B)/3;%计算亮度分量

S = 1-3.*temp1./temp2;
temp1 = 0.5*((R-G)+(R-B));
temp2 = sqrt((R-G).^2+(R-B).*(G-B));
theta = acos(temp1./(temp2+eps));
H=theta;
H(B>G)=2*pi-H(B>G);
H=H/(2*pi);
H(S==0)=0;
%%%%%%%%%%%%%%%%%%%%end%%%%%%%%%%%%%%
%%%%%%%%%%进行模糊估计增强 begin%%%%%
```

```
I2 = I.*255;%V 通道的值在[0,1]内，转换到[0,255]
[I4,k] = FuzzyHE(I2);
%%%%%%%%%%end%%%%%%%%%%%%%%%%

%%%%%%%GIMP 量化和 SCR 白平衡 begin%%%%%%%%
I6 = conMV(I4,alpha);
I6 = simplestColorBalance(I6,s1,s2);
lamda = double(I6)./(I2+eps);
A = min(1,lamda);
I7 = I6/max(max(I6));
%%%%%%%%%%%%%%%end%%%%%%%%%%%%%%%%
HSI = cat(3,H,S,I7);
%%%%%%%%转换 RGB 空间%%%%%%%%%%%%%%
h=HSI(:,:,1)*2*pi;
s=HSI(:,:,2);
i=HSI(:,:,3);
r=zeros(size(HSI,1),size(HSI,2));
g=zeros(size(HSI,1),size(HSI,2));
b=zeros(size(HSI,1),size(HSI,2));
%当 H 在[0,2pi/3]间
ind=find((h>=0)&(h<2*pi/3));
b(ind)=i(ind).*(1.0-s(ind));
r(ind)=i(ind).*(1.0+s(ind).*cos(h(ind))./cos(pi/3.0-h(ind)));
g(ind)=3.0*i(ind)-(r(ind)+b(ind));
%当 H 在[2pi/3,4pi/3]间
ind=find((h>2*pi/3)&(h<4*pi/3));
h(ind)=h(ind)-pi*2/3;
r(ind)=i(ind).*(1.0-s(ind));
g(ind)=i(ind).*(1.0+s(ind).*cos(h(ind))./cos(pi/3.0-h(ind)));
b(ind)=3.0*i(ind)-(r(ind)+g(ind));
%当 H 在[4pi/3,2pi]间
ind=find((h>=4*pi/3)&(h<2*pi));
h(ind)=h(ind)-pi*4/3;
g(ind)=i(ind).*(1.0-s(ind));
b(ind)=i(ind).*(1.0+s(ind).*cos(h(ind))./cos(pi/3.0-h(ind)));
r(ind)=3.0*i(ind)-(g(ind)+b(ind));
%变换前后的比例系数 A
r = im2uint8(r.*A);
```

```
g = im2uint8(g.*A);
b = im2uint8(b.*A);
rgb=cat(3,r,g,b);
info2=entropy(rgb2gray(rgb));
original_image = IMG;
enhance_image = rgb;
End
```