



河南大學

明德新民 止于至善

第5章 彩色图像处理

陈小潘

计算机与信息工程学院



目录

- 彩色图像基础
- 彩色变换
- 彩色图像增强
- 彩色图像平滑
- 彩色图像锐化
- 彩色图像分割

彩色图像基础

两个因素推动彩色的运用。

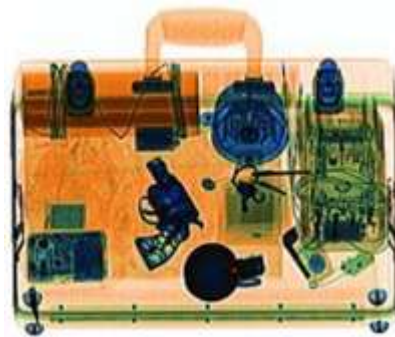
- 彩色是一个强大的描述子，它通常可以简化从场景中提取和识别目标
- 人类可以分辨上千种不同的颜色，但是只能分辨大约几十种灰度。彩色在人工图像分析中十分有用。



陶瓷的真彩色图像



多光谱遥感图像（假彩色图像）



行李箱的X光图像进行伪彩色图像处理后的图像

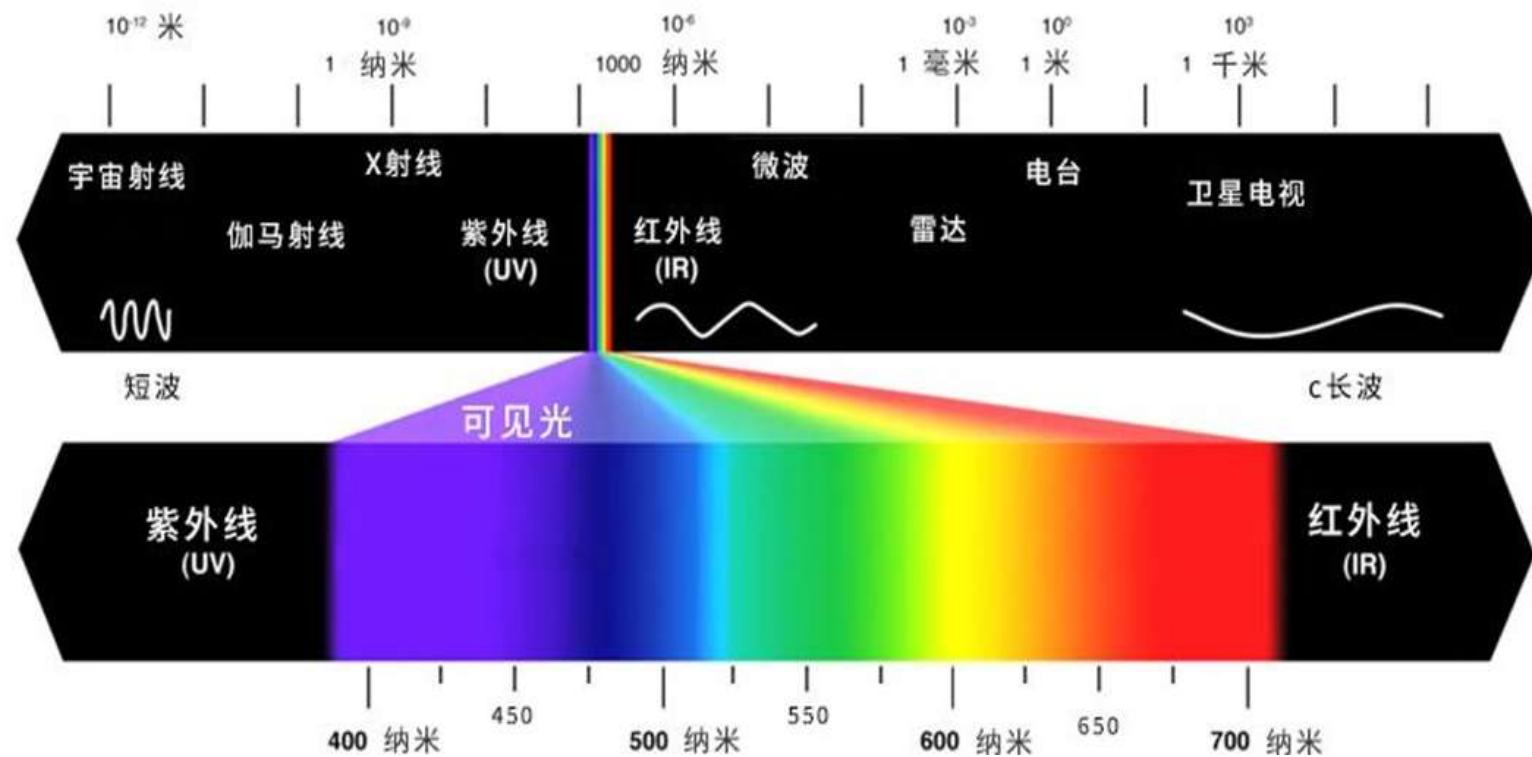
彩色图像基础

1665—1667年间，牛顿独自进行了一系列实验，探究光的各种现象。通过将一块三棱镜置于阳光下，他发现透过三棱镜后，光被分解成了不同的颜色，这一现象后来被称为光谱。牛顿首次将三棱镜转化为光谱仪，揭示了颜色本质的起源。



图 牛顿发现光谱现象

彩色图像基础



彩色图像基础

■ 什么是颜色

颜色是通过眼、脑和我们的生活经验所产生的对光的视觉感受，具有主观性。我们肉眼所见到的光线，是由波长范围很窄的电磁波产生的，不同波长的电磁波表现为不同的颜色，对色彩的辨认是肉眼受到电磁波辐射能刺激后所引起的视觉神经感觉。

彩色图像基础

- 颜色可分为无彩色和有彩色两大类
 - ✓ 无彩色指白色、黑色和各种深浅程度不同的灰色。能够同样吸收所有波长光的表面看起来是灰色的，反射的光多则显浅灰色，反射的光少则显深灰色
 - ✓ 彩色指除去上述黑白系列以外的各种颜色
- 通常所说的颜色一般多指彩色
- 可见光区的波长在400nm-700nm，人类视觉系统对红、绿、蓝三个波段的光特别敏感，对这三个波段的光能量进行采样，就可以得到一幅彩色图像。

彩色图像基础

■ 彩色的定义

彩色是物体的一种属性，一般来说它取决于三个方面的因素：

- ✓ 光源——照射光的光谱特性或光谱能量分布。
- ✓ 物体——被照射物体的反射特性。
- ✓ 成像接收器（眼睛或成像传感器）——光谱能量吸收性质。

其中光特性是彩色科学的核心。如果光没有颜色（比如观察者看到的黑白电视的光），那么它的属性只是亮度或数值。亮度可以用灰度值来描述，灰度值范围从黑色到灰色，最后到白色。

彩色图像基础

■ 彩色的定义

彩色的三种基本特性量

✓ **辉度**（亮度、明度）：是指每单位面积、每单位立体角，在某一方向上，自发光表面发射出的光通量，也就是指眼睛从某一方向所看到光源或**物体反射**光线的明亮强度。**辉度与物体的反射率成正比。**

✓ **色调**：与光谱中光的波长相联系

✓ **饱和度**：色调光的纯度

■ 色调和饱和度合起来称为**色度**。彩色可用辉度和色度共同表示。

■ 三基色原理

色度学中的三基色原理：自然界中的绝大多数的颜色都可看作是由**红、绿、蓝**三种颜色组合而成。三种基本色光的波长：

红（R，red）：700nm

绿（G，green）：546.1nm

蓝（B，blue）：435.8nm

一般将红、绿、蓝这三种颜色称为三基色。

彩色图像基础

■ 三基色原理

相加混色

三基色按不同比例相加进行的混色，称为相加混色。

红色 + 蓝色 = 品红色

红色 + 绿色 = 黄色

绿色 + 蓝色 = 青色

红色 + 绿色 + 蓝色 = 白色

(5.1)



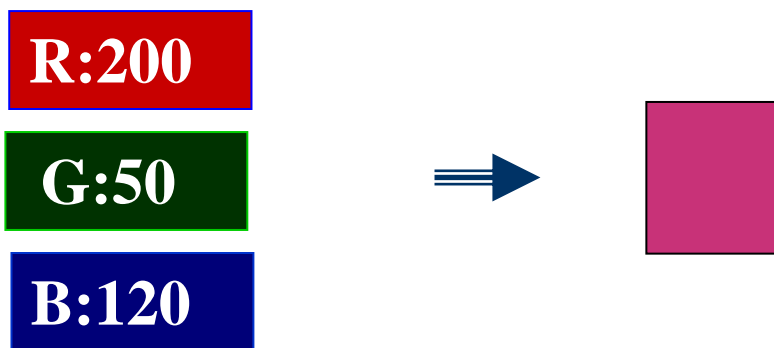
由三基色中的两色相加混色而成的颜色，称为二次色。

彩色图像基础

■ 三基色原理

相加混色

相加混色中涉及到灰色比例问题：



相加混色主要应用在录像、电视和计算机显示器等。

■ 三基色原理

相减混色

利用颜料和染料等的吸色性质可以实现相减混色。

- ✓ 相减混色就是从白光中滤去某种颜色而得到另一种颜色。
- ✓ 相减混色的基色为青、品红色、黄。

白色 - 红色 = 青色

白色 - 绿色 = 品红色 (5.2)

白色 - 蓝色 = 黄色

白色 - 绿色 - 红色 - 蓝色 = 黑色

彩色图像基础

■ 三基色原理

相减混色（续）

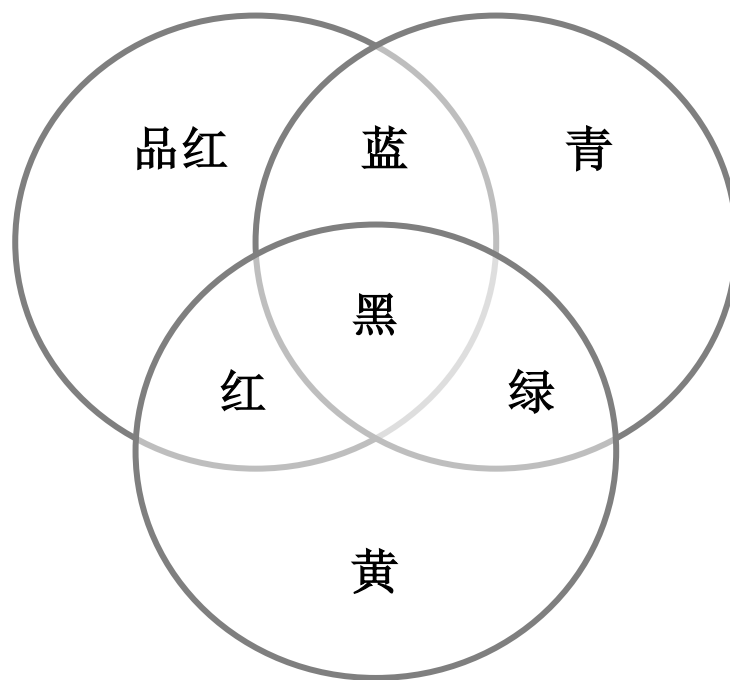


图 相减混色的三基色及其补色的关系

相减混色主要应用：绘画、摄影（包括彩色电影胶片制作）、彩色印刷和彩色印染等。

彩色图像基础

■ CIE色度图

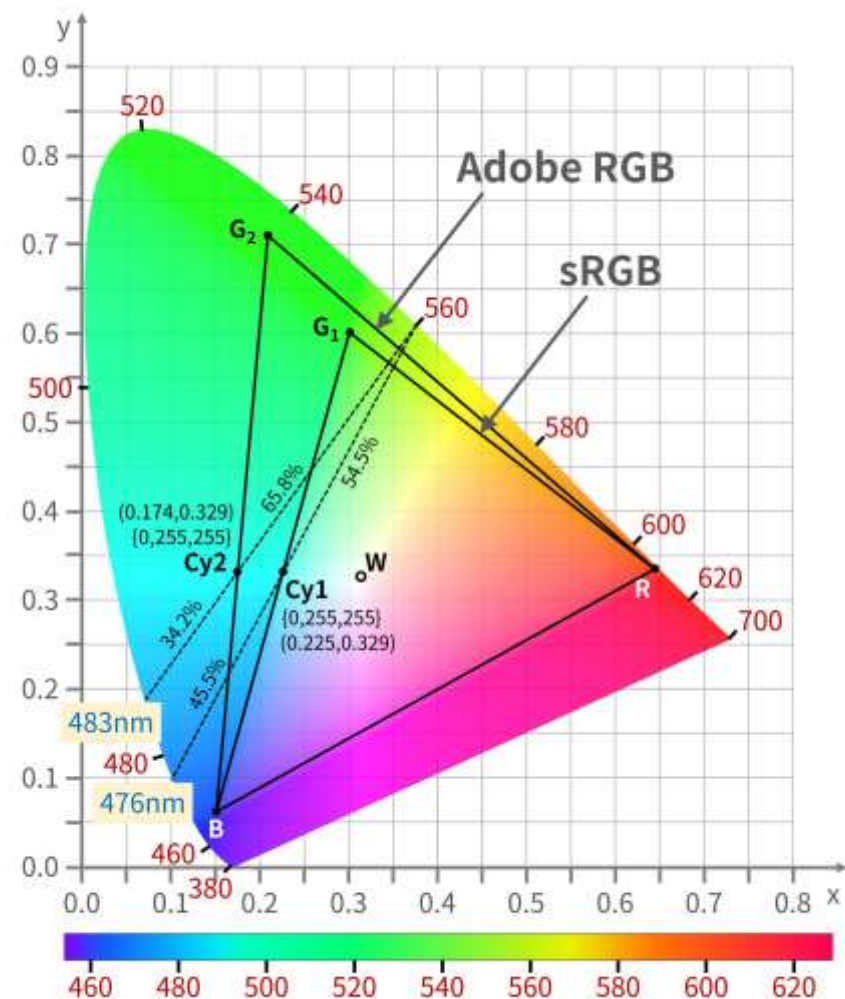
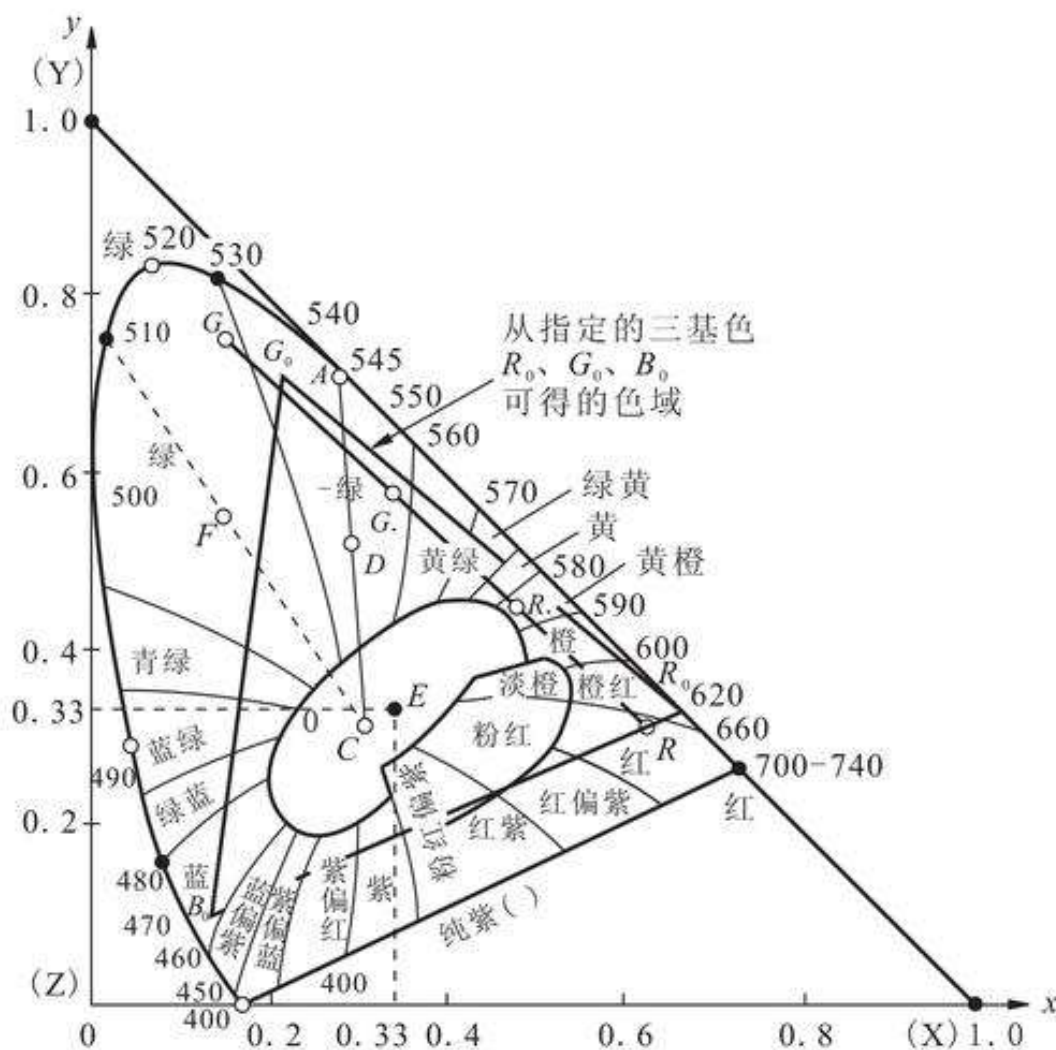
CIE(Commission Internationale de l'Eclairage, 国际照明委员会)。设X、Y和Z分别表示形成某种特殊颜色时需要的红、绿、蓝三基色的量值；x、y和z分别表示形成某种特殊颜色C时红、绿、蓝三基色所占的比例系数，则有：

$$C = xX + yY + zZ \quad (5.5)$$

且 $x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z} \quad z = \frac{Z}{X + Y + Z} \quad (5.6-5.8)$

显然有： $x + y + z = 1 \quad (5.9)$

彩色图像基础



彩色图像基础

■ CIE色度图

- (1) 从380nm（紫色）到700nm（红色）的所有纯色都位于舌形色度图的边界上；任何不在舌形色度图的边界但在其内部的点都代表一种由纯色混合而成的颜色。
- (2) 离开舌形色度图的边界移向中心，表示加入更多的白光而使该颜色的纯度降低，到中心的等能量点饱和度为零。
- (3) 图中连接任意两点的直线表示了由连线两端的点所代表的颜色按不同比例相加而得到的颜色变化。
- (4) 从等能量点到位于色度图边界上的任意一点画一直线表示了对应边界点纯色的所有色调。
- (5) 过C点的直线端点的两彩色为互补色

彩色模型

彩色模型：也称为彩色空间或彩色系统，是一种简化彩色规范的方式，旨在根据特定标准以常见和可接受的方式表示颜色。

它通过坐标系统和子空间的规范来定义，每种颜色在该系统中通过坐标系的单个点表示。

常见的彩色模型包括RGB、CMY（CMYK）、HSI、YUV、YCbCr、Lab等，这些模型在实际项目和应用中被广泛使用。

彩色模型

■ 彩色模型分类

(1) 面向诸如彩色监视器、彩色视频摄像机和彩色打印机的硬件设备。

主要有RGB模型、CMY（青、品红、黄）模型和CMYK（青、品红、黄、黑）模型。**RGB模型**主要用于彩色监视器和彩色视频摄像机；**CMYK**主要用于彩色打印机。

(2) 面向彩色处理应用的模型

例如，HSI模型（hue-saturation-intensity，即色调、亮度和饱和度）。

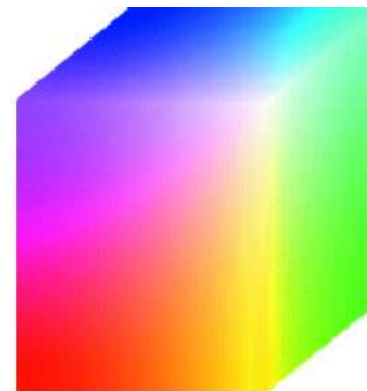
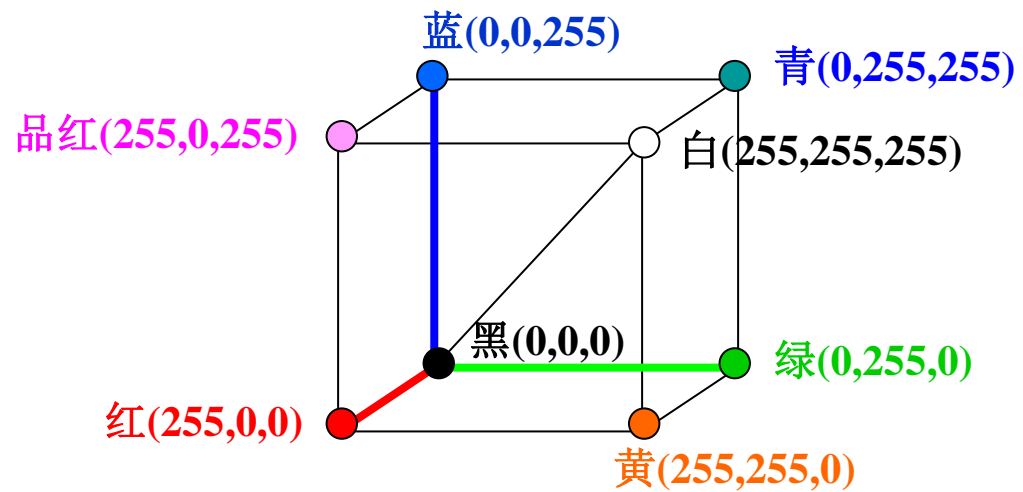
彩色模型

■ RGB彩色模型

- ✓ RGB彩色模型使用红绿蓝三原色的亮度来定量表示颜色，该颜色模型也称为**加色混色模型**，是以RGB三色光相互叠加来实现混色的方法。
- ✓ RGB彩色模型可以看作是三维直角坐标彩色系统中的一个单位正方体。
- ✓ 彩色立方体中的一个点来表示一种颜色。每个点有三个分量，分别代表该点颜色的 R、G、B 值。
- ✓ 原点所对应的颜色为黑色，该点的三个分量值都为零。距离原点最远的顶点对应的颜色为白色。
- ✓ 从黑到白的灰度值分布在这两个点的连线上，该线称为**灰色线**。

彩色模型

■ RGB彩色模型



彩色模型

■ RGB彩色模型

一般为方便起见，总将立方体归一化为单位立方体，这样所有的R、G、B的值都在区间[0, 1]之中。

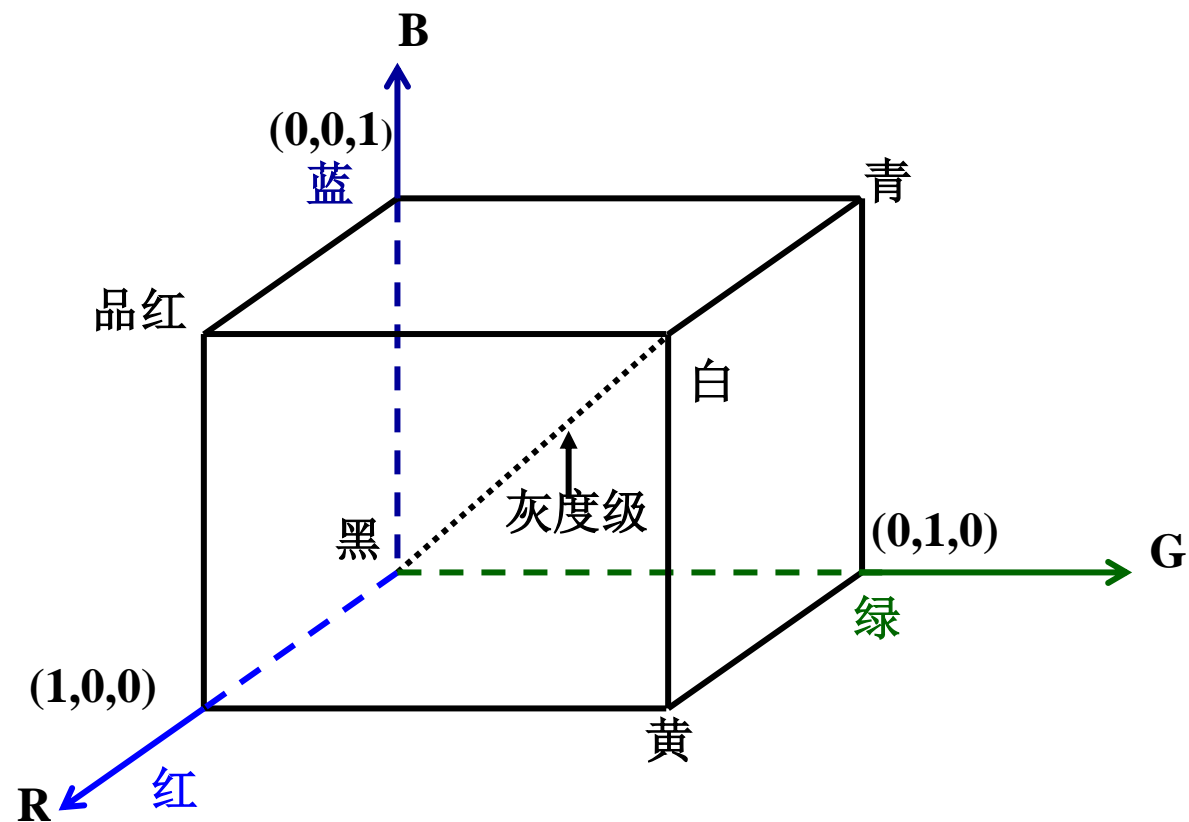


图5.4 RGB彩色立方体示意图

彩色模型

■ RGB彩色模型

- ✓ RGB彩色模型表示的图像由3个图像分量组成，每个图像分量都是其原色图像。当送入RGB监视器时，这三幅图像在荧光屏上混合产生一幅合成的彩色图像。
- ✓ RGB空间中表示每个像素的比特数称为像素深度。RGB图像由红绿蓝三幅图像构成，假设每幅图像都是8bit图像，则每个RGB图像是24bit图像。
- ✓ 24bit的彩色图像也称全（真）彩色图像
- ✓ 全彩色图像颜色总数是 $2^{24}=16777216$ 。

彩色模型

■ RGB彩色模型

RGB颜色空间最常用的用途就是**显示器系统**，彩色阴极射线管、彩色光栅图形的显示器都使用R、G、B数值来驱动R、G、B电子枪发射电子，并分别激发荧光屏上的R、G、B三种颜色的荧光粉发出不同亮度的光线，并通过相加混合产生各种颜色；扫描仪也是通过吸收原稿经反射或透射而发送来的光线中的R、G、B成分，并用它来表示原稿的颜色。

彩色模型

■ RGB彩色模型

RGB色彩空间称为与**设备相关**的色彩空间，因为不同的扫描仪扫描同一幅图像，会得到不同色彩的图像数据；不同型号的显示器显示同一幅图像，也会有不同的色彩显示结果。

显示器和扫描仪使用的RGB空间与CIE 1931 RGB真实三原色表色系统空间是不同的，后者是与设备无关的颜色空间。

彩色模型

■ RGB彩色模型

■ 提取分量图像

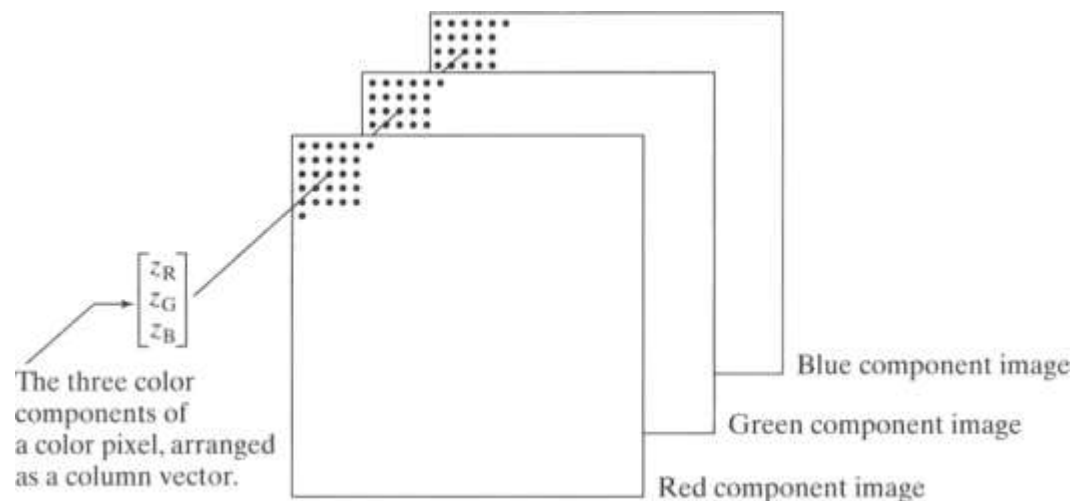
`fR=rgb-image (:,:,1);`

`fG=rgb-image (:,:,2);`

`fB=rgb-image (:,:,3);`

■ 用cat函数将分量图像合成彩色图像

`rgb-image = cat(3, fR, fG, fB)`



彩色模型

■ HSI彩色模型

HSI模型由美国色彩学家孟塞尔(H.A.Munseu)于1915年提出的，它反映了人的视觉系统感知彩色的方式，以**色调**（hue）、**饱和度**（saturation）和**强度**（intensity，也称亮度）三种基本特征量来感知颜色。

- **色调H**：与光波的频率有关，它表示人的感官对不同颜色的感受，如红色、绿色、蓝色等，它也可表示一定范围的颜色，如暖色、冷色等。
- **饱和度S**：表示颜色的纯度，纯光谱色是完全饱和的，加入白光会稀释饱和度。饱和度越大，颜色看起来就会越鲜艳，反之亦然。
- **亮度I**：对应成像亮度和图像灰度，是颜色的明亮程度。

■ HSI彩色模型

HSI模型的建立基于两个重要的事实：

- ① I分量与图像的彩色信息无关；
- ② H和S分量与人感受颜色的方式是紧密相联的。这些特点使得HSI模型非常适合彩色特性检测与分析。

彩色模型

■ HSI彩色模型

HSI色彩空间可以用一个圆锥空间模型来描述，如图所示。色彩空间的圆锥模型相当复杂，但确能把色调、亮度和色饱和度的变化情形表现得很清楚。在HSI色彩空间可以大大简化图像分析和处理的工作量。HSI色彩空间和RGB色彩空间只是同一物理量的不同表示法，因而它们之间存在着转换关系。

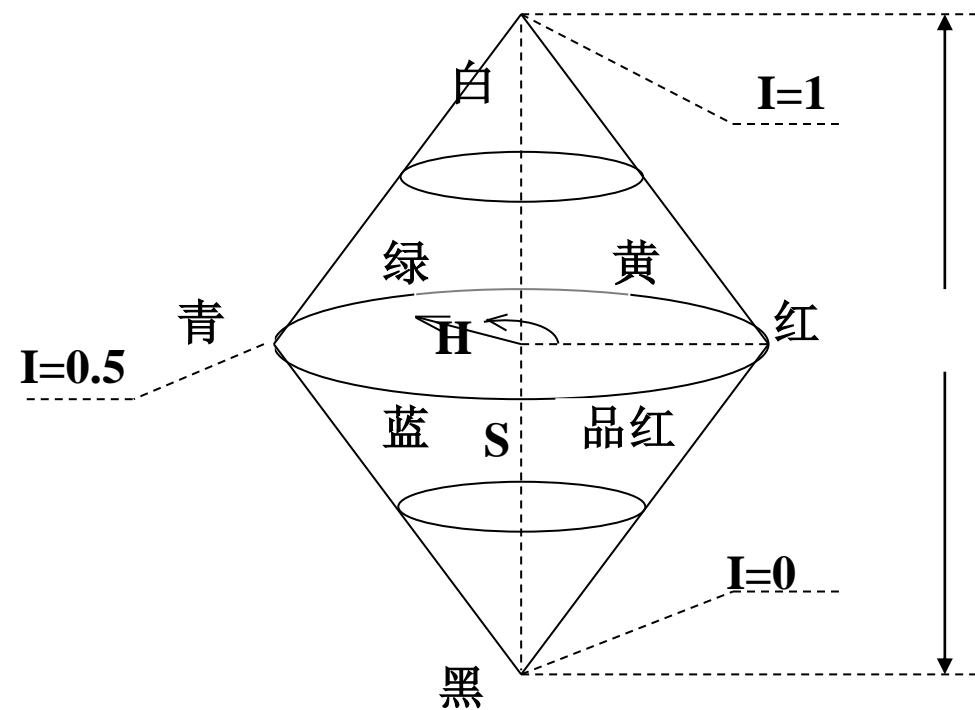


图5.5 基于圆形彩色平面的HSI彩色模型

彩色模型

上图双圆锥模型中下面圆锥的顶点为黑点，上面的圆锥的顶点为白点，连接黑点和白点的双圆锥的轴线成为亮度轴，用于表示亮度分量 I 。黑点的亮度为0，白点的亮度为1，任何位于 $[0,1]$ 内的亮度值都可以由亮度轴与垂直于该亮度轴的圆平面的交点给出。

进一步来讲，对于双圆锥体上的任一个色点 p 来说， p 的亮度由 p 所在的垂直于亮度轴的平面与亮度轴的交点确定。

当强度 $I=0$ 时，色调 H 、饱和度 S 无定义；当 $S=0$ 时，色调 H 无定义。

彩色模型

■ HSI彩色模型

I表示光照强度或称为亮度，它确定了像素的整体亮度，而不管其颜色是什么。亮度**I**大小由物体反射系数来决定。反射系数越大，物体的亮度越大，反之越小。



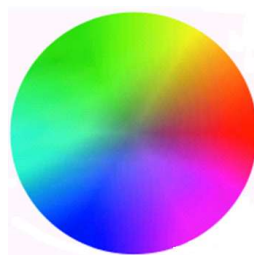
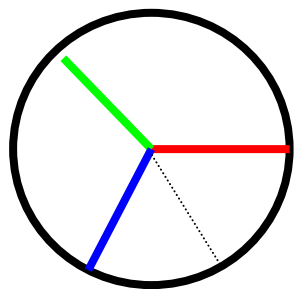
I: 小

大

彩色模型

■ HSI彩色模型

色调H由绕亮度轴I的旋转角度表示，0度的彩色为红色，120度的为绿色，240度的为蓝色。色度从0度~360度覆盖了所有可见光谱的彩色。彩色的色调反映了该彩色最接近什么样的光谱波长。



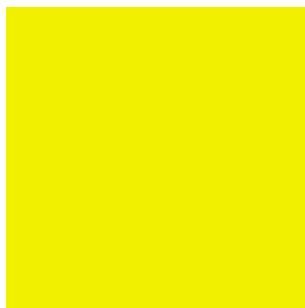
彩色模型

■ HSI彩色模型

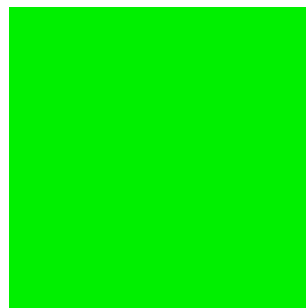
HSI色系 —— 色度(H)效果示意图



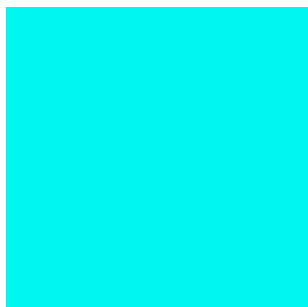
$H=0^\circ$



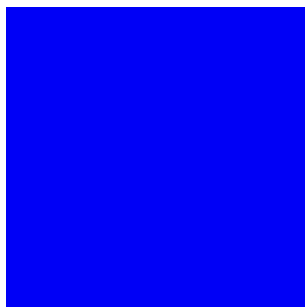
$H=60^\circ$



$H=120^\circ$



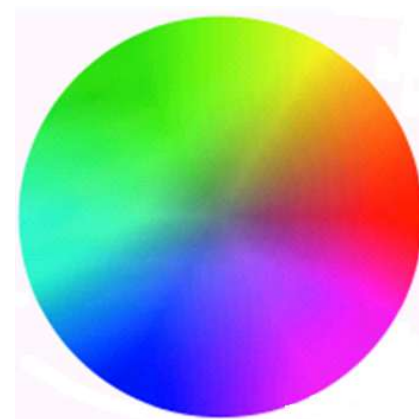
$H=180^\circ$



$H=240^\circ$



$H=300^\circ$



彩色模型

■ HSI彩色模型

S: 饱和度，表示颜色的深浅程度。饱和度越高，颜色越深，如深红色、深绿色等。饱和度参数是色环的原点到彩色点的半径长度。在环的外围圆周是纯的或称饱和的颜色，其饱和度值为1。在中心是中性（灰）色，即饱和度为0。

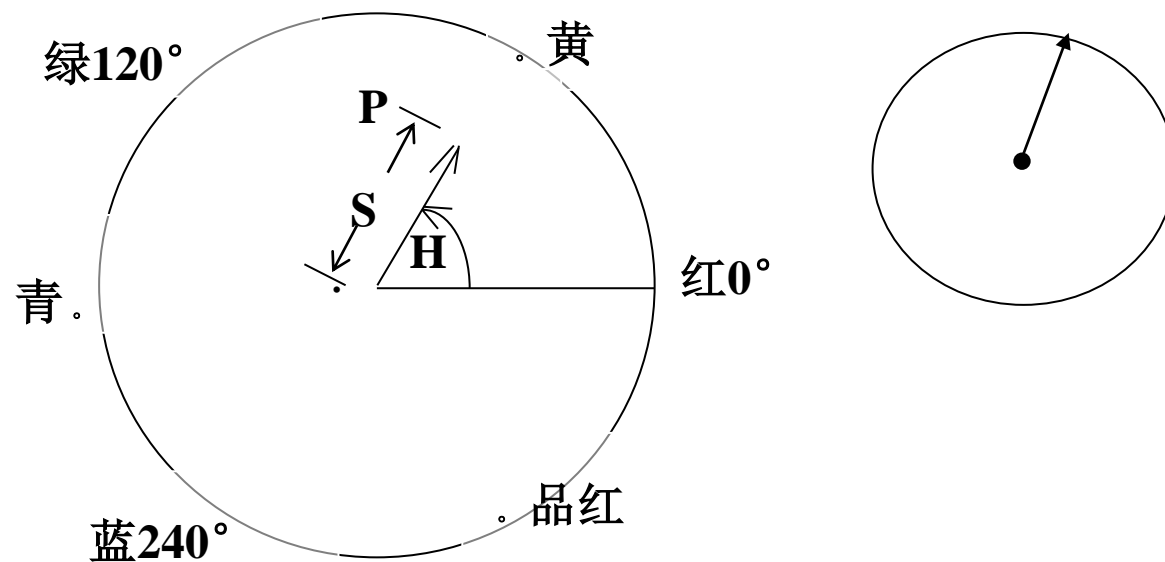


图 HSI彩色模型中的色调和饱和度

彩色模型

■ HSI彩色模型

HSI色系 —— 饱和度(S)效果示意图



不完全饱和

完全饱和

不完全饱和

彩色模型

■ HSI彩色模型

- ✓ 同一种色块，在不同强度的白光照射下，反射的光波波长一样（色调相同），但人眼感觉到的颜色不同。
- ✓ 某一颜色的光，亮度很弱，趋于黑色；反之，亮度很强，趋于白色，其饱和度同样趋于零；只有在亮度适中时，该颜色的饱和度才能达到最大



图 饱和度与亮度的关系示例

彩色模型

■ HSI彩色模型

HIS模型主要应用于彩色动画、图形创作等软件彩色应用处理系统。主要特点：

- (1) H分量、S分量与人感受颜色的方式联系紧密，I分量与图像的彩色信息无关，因而使HSI模型非常适合于彩色特性检测与分析。
- (2) 在彩色图像处理时仅对I分量进行处理不会改变原图像总的彩色种类，使得HSI模型成为开发基于彩色描述的图像处理方法的良好工具，而这种彩色描述对人来说是自然而直观的。

HSI模型的优点：解除了亮度和彩色信息之间的联系，它适合于许多灰度处理技术。

彩色模型

■ RGB彩色模型到HSI彩色模型的转换

$$\theta = \arccos \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - G)(G - B)]^{1/2}} \right\} \quad (5.10)$$

$$I = \frac{1}{3}(R + G + B) \quad (5.11)$$

$$S = 1 - \frac{3}{R + G + B} [\min(R, G, B)] \quad (5.12)$$

$$H = \begin{cases} \theta & G \geq B \\ 360 - \theta & G < B \end{cases} \quad (5.13)$$

彩色模型

■ RGB彩色模型到HSI彩色模型的转换

% 从RGB图像提取HSI模型的H、S、I分量图像的matlab程序

```
clc; clear all; close all;
img0=imread('miroslava.jpg');
rgb = im2double(img0); % 将图像数据类型转换为double类型
r = rgb(:, :, 1); g = rgb(:, :, 2); b = rgb(:, :, 3); % 提取rgb三个分量图像
% 提取H分量, 其中eps是防止除数为0的很小量
theta_num = 0.5*((r - g) + (r - b)); % 计算色调角式 (5.10) 的分子
theta_den = sqrt((r - g).^2 + (r - b).*(g - b)); % 计算式 (5.10) 的分母
theta_num = acos(theta_num./(theta_den + eps)); % 依据式 (5.10)
H = theta_num; % 根据式 (5.13) 计算H分量值
H(b > g) = 2*pi - H(b > g);
```

```
theta_num = min(min(r, g), b);
theta_den = r + g + b;
theta_den(theta_den == 0) = eps; % if theta_den=0, 则theta_den=0
S = 1 - 3.* theta_num./theta_den; % 根据式 (5.12) 提取S分量
H(S == 0) = 0; % if S=0, 则H=0
I = (r + g + b)/3; % 根据式 (5.11) 提取I分量
subplot(2,2,1); imshow(img0); title('原RGB彩色图像'); % 显示原彩色图像
subplot(2,2,2); imshow(H); title('H分量图像'); % 显示H分量图像
subplot(2,2,3); imshow(S); title('S分量图像'); % 显示S分量图像
subplot(2,2,4); imshow(I); title('I分量图像'); % 显示I分量图像
```

彩色模型

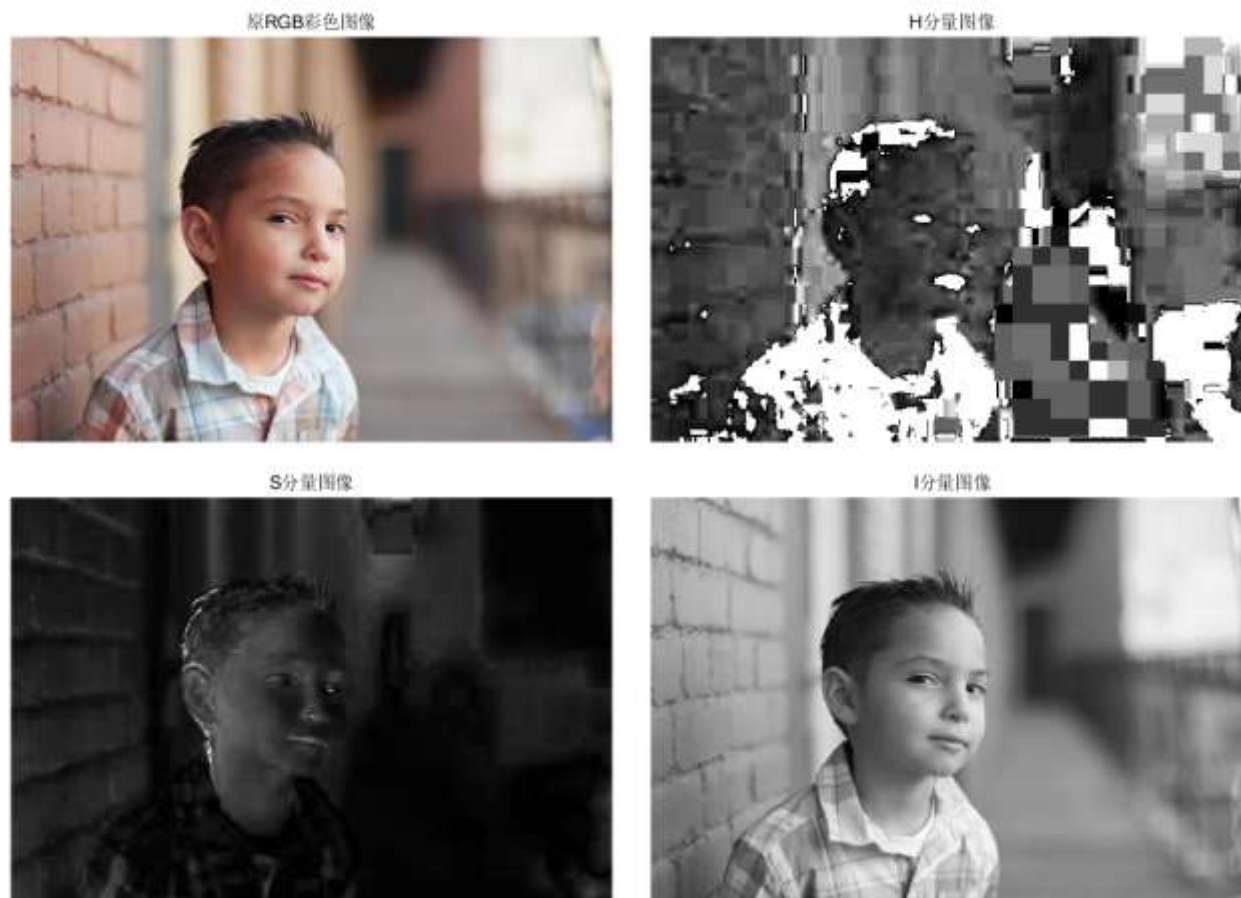
■ RGB彩色模型到HSI彩色模型的转换

```
function hsi = rgb2hsi(rgb)
rgb = im2double(rgb);
R = rgb(:,:,1);%分离出R通道
G = rgb(:,:,2);%分离出G通道
B = rgb(:,:,3);%分离出B通道
%要计算出H的话，根据公式需要先算出 $\theta$ 角度
num = ((R - G) + (R - B))/2.0;%计算公式的上半部分
den = sqrt((R - G).^2 + (R - B).*(G - B));%计算公式的下半部分
theta = acos(num./(den + eps));%将上下两部分相除同时取一个反三角函数 这里加入的eps是防止除数为零
H = theta;%先令H为 $\theta$ 
H(B > G) = 2*pi - H(B > G);%如果B分量大于G分量 则取 $2\pi - \theta$ 
H = H/(2*pi);%得到H通道最终值
num = min(min(R,G),B);%取得RGB三通道中最小的
den = R + G + B;%取得RGB通道之和
S = 1 - 3.*num./(den + eps);%进行S通道的计算 这里的eps同样是防止除数为零
H(S == 0) = 0;
I = (R + G + B)/3;%进行I通道的计算
hsi = cat(3,H,S,I);%将HSI三通道联合成为一个HSI图像
end
```


彩色模型

■ RGB彩色模型到HSI彩色模型的转换

例：利用RGB彩色模型图像提取HSI彩色模型的H（色调）、S（饱和度）和I（亮度）分量图像的Matlab程序。



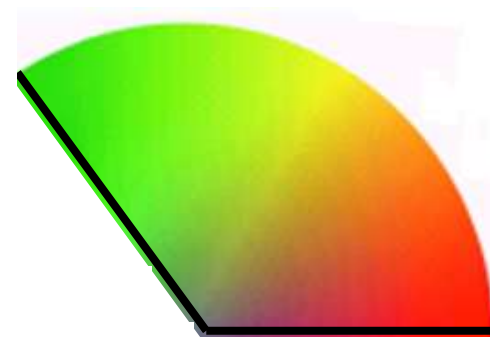
■ HSI彩色模型到RGB彩色模型的转换

(1) 当 $0^\circ \leq H < 120^\circ$:

$$B = I(1 - S) \quad (5.14)$$

$$R = I \left[1 + \frac{S \cos(H)}{\cos(60^\circ - H)} \right] \quad (5.15)$$

$$G = 3I - (B + R) \quad (5.16)$$



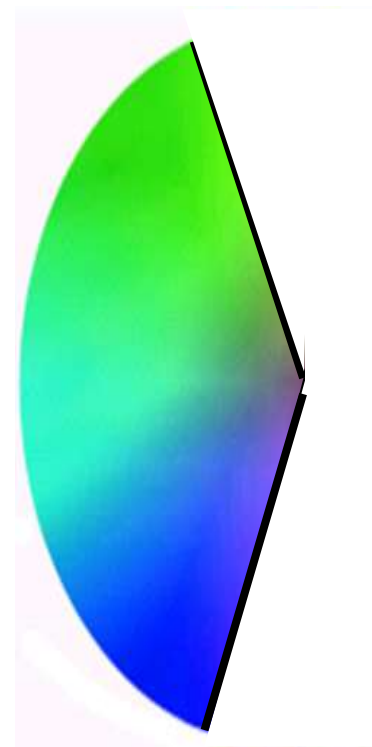
■ HSI彩色模型到RGB彩色模型的转换

(2) 当 $120^\circ \leq H < 240^\circ$:

$$R = I(1 - S) \quad (5.17)$$

$$G = I \left[1 + \frac{S \cos(H - 120^\circ)}{\cos(180^\circ - H)} \right] \quad (5.18)$$

$$B = 3I - (R + G) \quad (5.19)$$



彩色模型

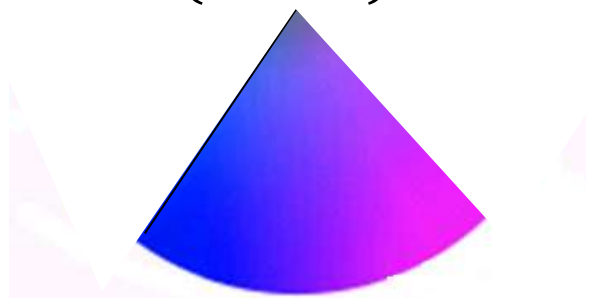
■ HSI彩色模型到RGB彩色模型的转换

(3) 当 $240^\circ \leq H < 300^\circ$:

$$G = I(1 - S) \quad (5.20)$$

$$B = I \left[1 + \frac{S \cos(H - 240^\circ)}{\cos(300^\circ - H)} \right] \quad (5.21)$$

$$R = 3I - (G + B) \quad (5.22)$$



值得注意的是：300~360之间为非可见光谱色，没有定义。

彩色模型

```
function rgb=hsi2rgb(hsi)
```

```
%提取HSI的各个分量
```

```
hsi=im2double(hsi);%将hsi转化为双精度浮点类型
```

```
H=hsi(:,:,1)*2*pi;
```

```
S=hsi(:,:,2);
```

```
I=hsi(:,:,3);
```

```
%执行变换方程
```

```
R=zeros(size(hsi,1),size(hsi,2));
```

```
G=zeros(size(hsi,1),size(hsi,2));
```

```
B=zeros(size(hsi,1),size(hsi,2));
```

```
%RG区( $0 \leq H < 2\pi/3$ )
```

```
idx=find((0<=H) & (H<2*pi/3));%寻找 $0 \leq H < 2\pi/3$ 
```

```
B(idx)=I(idx).*(1-S(idx));
```

```
R(idx)=I(idx).*(1+S(idx).*cos(H(idx))./cos(pi/3-H(idx)));
```

```
G(idx)=3*I(idx)-(R(idx)+B(idx));
```

```
%BG区( $2\pi/3 \leq H < 4\pi/3$ )
```

```
idx=find((2*pi/3<=H) & (H<4*pi/3));%寻找 $2\pi/3 \leq H < 4\pi/3$ 
```

```
R(idx)=I(idx).*(1-S(idx));
```

```
G(idx)=I(idx).*(1+S(idx).*cos(H(idx)-2*pi/3)./cos(pi-H(idx)));
```

```
B(idx)=3*I(idx)-(R(idx)+G(idx));
```

```
%BR区( $4\pi/3 \leq H \leq 2\pi$ )
```

```
idx=find((4*pi/3<=H) & (H<=2*pi));%寻找 $4\pi/3 \leq H \leq 2\pi$ 
```

```
G(idx)=I(idx).*(1-S(idx));
```

```
B(idx)=I(idx).*(1+S(idx).*cos(H(idx)-4*pi/3)./cos(5*pi/3-H(idx)));
```

```
R(idx)=3*I(idx)-(G(idx)+B(idx));
```

```
%将3个分量联合成为一个RGB图像
```

```
rgb=cat(3,R,G,B);
```

```
rgb=max(min(rgb,1),0);
```

```
end
```

彩色模型

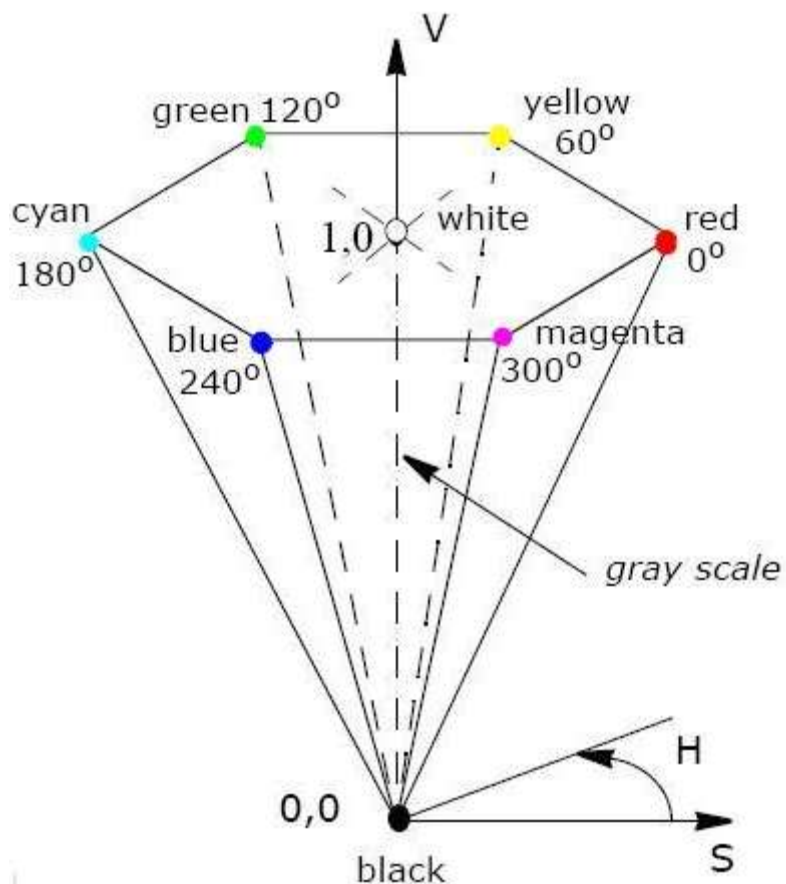
■ HSV彩色模型

HSV(Hue, Saturation, Value)是根据颜色的直观特性由A. R. Smith在1978年创建的一种颜色空间，也称六角锥体模型(Hexcone Model)。

HSV颜色模型是指H、S、V三维颜色空间中的一个可见光子集，它包含某个颜色域的所有颜色。

彩色模型

■ HSV彩色模型



- ✓ **色调H**参数表示色彩信息，即所处的光谱颜色的位置。该参数用一角度量来表示，取值范围为 $0^{\circ} \sim 360^{\circ}$ 。若从红色开始按逆时针方向计算，红色为 0° ，绿色为 120° ，蓝色为 240° 。它们的补色是：黄色为 60° ，青色为 180° ，紫色为 300° ；
- ✓ **饱和度S**：取值范围为 $0.0 \sim 1.0$ ；
- ✓ **亮度V**：取值范围为 0.0 (黑色) ~ 1.0 (白色)。

彩色模型

■ HSV彩色模型

HSV模型在颜色分割中的应用具有重要意义，特别是在处理颜色时，H（色调）和S（饱和度）分量起到了关键作用。H分量代表颜色的种类或色调，而S分量则表示颜色的纯度或饱和度。

在颜色分割中，利用H和S两个分量，可以有效地进行颜色分割。H和S分量通常用于表示颜色之间的距离，称为“颜色距离”。颜色距离是衡量两种颜色之间差异的数值，通过计算色调和饱和度的差异，可以精确地分辨不同的颜色区域。

利用HSV模型进行颜色分割，不仅能够提高分割精度，还能够使分割结果更加直观，因为H和S能直接反映出颜色的本质特征。

彩色模型

■ HSV彩色模型

RGB转HSV

(1) 对于图像中任意坐标点，其RGB颜色空间为(R,G,B)，HSV颜色空间为(H,S,V)，首先需要将R、G、B值转换到0~1之间：

$$R = R/255$$

$$G = G/255$$

$$B = B/255$$

彩色模型

■ HSV彩色模型

RGB转HSV

(2) 计算H、S、V值:

$$V = \max(R, G, B)$$

$$S = \begin{cases} \frac{V - \min(R, G, B)}{V} & , if V \neq 0 \\ 0 & , else \end{cases}$$

$$H = \begin{cases} 60 * (G - B) / (V - \min(R, G, B)) & , if V = R \\ 120 + 60 * (B - R) / (V - \min(R, G, B)) & , if V = G \\ 240 + 60 * (R - G) / (V - \min(R, G, B)) & , if V = B \end{cases}$$

如果计算得到的H值小于0, 将该值再加上360, 得到最终的H值:

彩色模型

■ HSV彩色模型

RGB转HSV

$$V \leftarrow \max(R, G, B)$$

$$S \leftarrow \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$H \leftarrow \begin{cases} 60(G - B)/(V - \min(R, G, B)) & \text{if } V = R \\ 120 + 60(B - R)/(V - \min(R, G, B)) & \text{if } V = G \\ 240 + 60(R - G)/(V - \min(R, G, B)) & \text{if } V = B \end{cases}$$

If $H < 0$ then $H \leftarrow H + 360$. On output $0 \leq V \leq 1, 0 \leq S \leq 1, 0 \leq H \leq 360$.

彩色模型

■ HSV彩色模型

HSV转RGB原理

对于图像中任意坐标点，其RGB颜色空间为(R,G,B)，HSV颜色空间为(H,S,V)。首先将可视化图像的H、S、V值分别转换到 $0^\circ \sim 360^\circ$ 、 $0 \sim 1$ 、 $0 \sim 1$ 的范围：

$$H = H * 2$$

$$S = S/255$$

$$V = V/255$$

彩色模型

■ HSV彩色模型

HSV转RGB原理

那么R、G、B的计算公式如右所示，其中floor表示向下取整运算。

Matlab中提供了HSV和RGB的转换程序

```
hsv_image = rgb2hsv(input_image);
```

```
enhanced_image = hsv2rgb(hsv_image);
```

$$h_i = \text{floor}\left(\frac{H}{60}\right)$$

$$f = \frac{H}{60} - h_i$$

$$p = V * (1 - S)$$

$$q = V * (1 - f * S)$$

$$t = V * (1 - (1 - f) * S)$$

$$(R, G, B) = \begin{cases} (V, t, p) & , \text{if } h_i = 0 \\ (q, V, p) & , \text{if } h_i = 1 \\ (p, V, t) & , \text{if } h_i = 2 \\ (p, q, V) & , \text{if } h_i = 3 \\ (t, p, V) & , \text{if } h_i = 4 \\ (V, p, q) & , \text{if } h_i = 5 \end{cases}$$

彩色模型

■ 其它彩色模型简介

1、CMYK色系——基本概念

这种色系用于印刷行业，是一种**减色系统**，将从白光中滤出三种原色之后获得的颜色，作为其色系的三原色CMY。K为黑色，为了印刷时，用黑色墨进行印刷。

- C: Cyan, 青色，从白色中滤去红色。
- M: Magenta, 品红，从白色中滤去绿色。
- Y: Yellow, 黄色，从白色中滤去蓝色。
- K: black=黑色，



图 佳能彩色喷墨打印机墨水

■ 其它彩色模型简介

1、CMYK色系——基本概念

在印刷过程中要经过一个分色的过程。所谓分色就是将计算机中使用的RGB颜色转换成印刷使用的CMYK颜色。

在转换过程中存在着两个复杂的问题：（1）这两个颜色空间在表现颜色的范围上不完全一样，RGB的色域较大而CMYK则较小，因此就要进行色域压缩；（2）是这两个颜色都是和具体的设备相关，颜色本身没有绝对性。因此就需要通过一个与设备无关的颜色空间来进行转换，即可以通过XYZ或LAB颜色空间来进行转换。

■ 其它彩色模型简介

1、CMYK色系——着色原理

着色原理是基于光吸收的。

C与M叠加：同时吸收了R与G，则为蓝色；

C与Y叠加：同时吸收了R与B，则为绿色；

M与Y叠加：同时吸收了G与B，则为红色。

彩色模型

■ 其它彩色模型简介

2、YUV色系——基本概念

- “Y” 表示明亮度（Luminance或Luma），即灰阶值；
- “U” 和 “V” 表示色度（Chrominance或Chroma），用于描述影像的色彩及饱和度。
- **目的：**为了使电视节目可以同时被黑白电视机及彩色电视机接收。电视信号在发射时，转换成YUV形式；接收时再还原成RGB三基色信号，由显像管显示。

目录

- 彩色图像基础
- 彩色变换
- 彩色图像增强
- 彩色图像平滑
- 彩色图像锐化
- 彩色图像分割

一、反色变换

反色（补色）是指与某种色调互补的另一种色调。

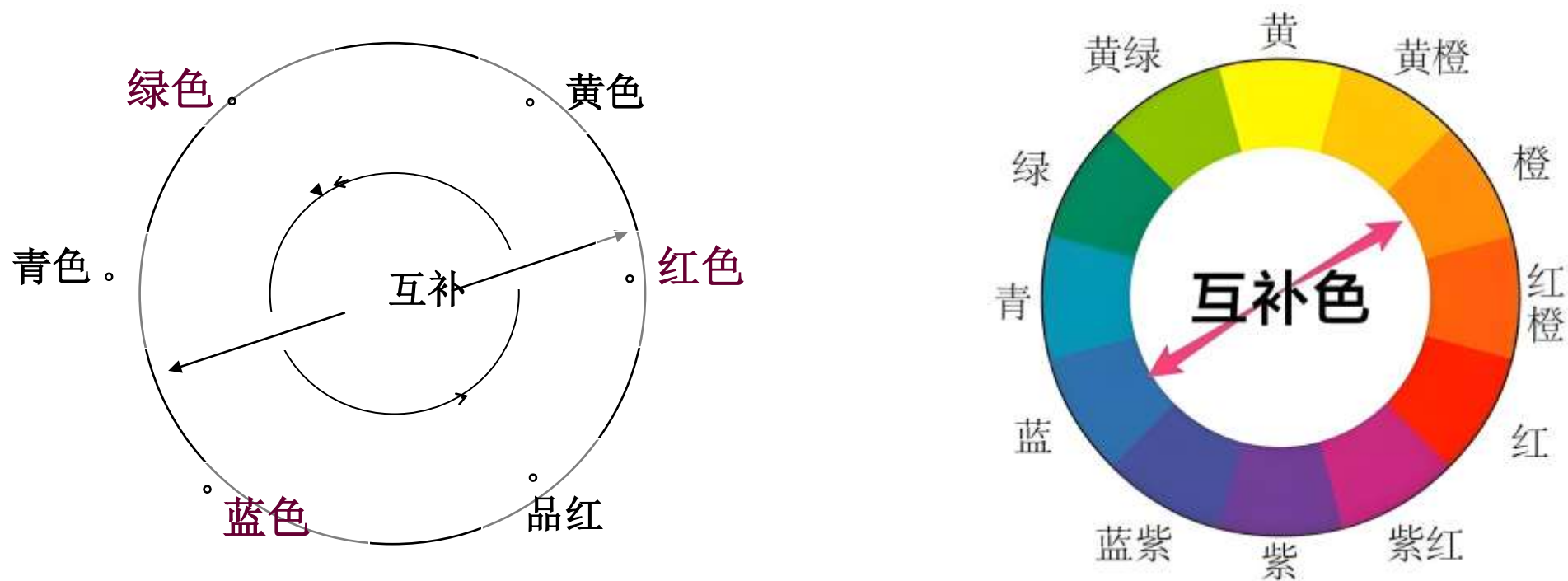


图 颜色间的互补关系

一、反色变换



原彩色图像



RGB补色图像

原图像的黄色区域被蓝色代替，原图像背景是白的，补色是黑色。

一、反色变换

设 $f(x,y)$ 为输入彩色图像，彩色分量的量化级别为256，则反色图像 $g(x,y)$ 与输入图像 $f(x,y)$ 的R、G、B分量之间的关系可表示为：

$$\begin{bmatrix} g_R(x,y) \\ g_G(x,y) \\ g_B(x,y) \end{bmatrix} = \begin{bmatrix} 255 - f_R(x,y) \\ 255 - f_G(x,y) \\ 255 - f_B(x,y) \end{bmatrix}$$

一、反色变换

```
clc;
im = imread('yellowlily.jpg');
% 获取图像的行数和列数
[rows, cols, ~] = size(im);
im1=im;
% 将每个像素的RGB值取反
for row = 1:rows
    for col = 1:cols
        im(row, col, 1) = 255 - im(row, col, 1);
        im(row, col, 2) = 255 - im(row, col, 2);
        im(row, col, 3) = 255 - im(row, col, 3);
    end
end
```

```
subplot(1,2,1);
imshow(im1);title('原彩色图像','fontsize',14)
subplot(1,2,2);
imshow(im);title('负片图像','fontsize',14)
```

逐个像素处理，效率低

一、反色变换



图 彩色图像的反色变换

一、反色变换

```
clc;  
im = imread('yellowlily.jpg');  
% 获取图像的行数和列数  
[rows, cols, ~] = size(im);  
im1=im;  
% 将每个像素的RGB值取反  
  
im1(:, :, 1) = 255 - im(:, :, 1);  
im1(:, :, 2) = 255 - im(:, :, 2);  
im1(:, :, 3) = 255 - im(:, :, 3);  
subplot(1,2,1);  
imshow(im);title('原彩色图像','fontsize',14)  
subplot(1,2,2);  
imshow(im1);title('负片图像','fontsize',14)
```

利用矩阵处理，效率高

二、彩色图像的灰度化

- 将彩色图像转变为灰度图像的处理称为彩色图像的灰度化处理。
- 将彩色图像转换为灰度图像的实质，就是通过对图像R、G、B分量的变换，使得每个像素点的R、G、B分量值相等。
- 彩色图像的灰度化方法主要包括：最大值法、平均值法和加权平均值法。

二、彩色图像的灰度化

(1) 最大值法

将输入图像中的每个像素的R、G、B分量值的最大者赋给输出图像中对应像素的R、G、B分量的方法。用公式可表示为：

$$g_R(x, y) = g_G(x, y) = g_B(x, y) = \max[f_R(x, y), f_G(x, y), f_B(x, y)] \quad (5.24)$$

注：因为灰度图图像像素值的特点是R、G、B分量的亮度值相同。

二、彩色图像的灰度化

(2) 平均值法

将输入图像中的每个像素的R、G、B分量的算术平均值赋给输出图像中对应像素的R、G、B分量的方法。用公式可表示为：

$$g_R(x, y) = g_G(x, y) = g_B(x, y) = (f_R(x, y) + f_G(x, y) + f_B(x, y))/3 \quad (5.25)$$

二、彩色图像的灰度化

(3) 加权平均法

将输入图像中的每个像素的R、G、B分量的加权平均值赋给输出图像中对应像素的R、G、B分量的方法。用公式可表示为：

$$g_R(x, y) = g_G(x, y) = g_B(x, y) = \omega_R f_R(x, y) + \omega_G f_G(x, y) + \omega_B f_B(x, y) \quad (5.26)$$

其中： $\omega_R + \omega_G + \omega_B = 1$

二、彩色图像的灰度化

(3) 加权平均法

人眼对绿光的亮度感觉仅次于白光，是三基色中最亮的，红光次之，蓝光最低。如果权值 ω_G 、 ω_R 、 ω_B 满足条件 $\omega_G > \omega_R > \omega_B$ ，将会得到比较合理的灰度化结果。

相关研究表明，当 $\omega_G=0.587$ 、 $\omega_R=0.299$ 、 $\omega_B=0.114$ 时，得到的灰度化图像较合理，此时公式（5.26）就变为：

$$f_R(x, y) = f_G(x, y) = f_B(x, y) = 0.299 \cdot f_R(x', y) + 0.587 \cdot f_G(x', y) + 0.114 \cdot f_B(x', y) \quad (5.27)$$

二、彩色图像的灰度化

```
% 将彩色图像转换成灰度图像matlab程序 (rgb_to_gray10_2.m)
clc; clear all; close all;
f=imread('miroslava.jpg');
[h, w, color] = size(f); % 求原彩色图像的矩阵参数
max_f = zeros(h, w); % 创建全零矩阵, 存储最大值灰度图像
for i = 1:h
    for j = 1:w
        max_f(i, j) = max(f(i, j, :)); % 最大值转换公式
    end
end
max_f = uint8(max_f); % 将转换结果转换为8位图像格式
mean_f = zeros(h, w); % 创建全零矩阵, 存储平均值灰度图像
for i = 1:h
    for j = 1:w
        sum = 0;
        for k = 1:color
            sum = sum + f(i, j, k) / 3.0; % 平均值转换公式
            mean_f(i, j) = sum;
        end
    end
end
end
```

二、彩色图像的灰度化

```
mean_f = uint8(mean_f);
```

```
weight_f = zeros(h , w); % 创建全零矩阵，存储加权平均灰度图像
```

```
for i = 1:h
```

```
    for j = 1:w
```

```
        weight_f(i,j)=f(i,j,1)*0.299 + f(i,j,2)*0.587 + f(i,j,3)*0.114;
```

```
    end
```

```
end
```

```
weight_f = uint8(weight_f);
```

```
subplot(2,2,1); imshow(f); title('原彩色图像');
```

```
subplot(2,2,2); imshow(max_f); title('最大值法图像');
```

```
subplot(2,2,3); imshow(mean_f); title('平均值法图像');
```

```
subplot(2,2,4); imshow(weight_f); title('加权平均法图像');
```

二、彩色图像的灰度化

原彩色图像



最大值法图像



平均值法图像



加权平均法图像



三、真彩色转变为256色

真彩色（True Color）是指在组成一幅彩色图像的每个像素值中，有R、G、B三个基色分量，每个基色分量直接决定其基色强度，这样产生的色彩称为真彩色。

发色总数指显示设备或图形系统能够呈现的不同颜色数量。16位色的发色总数是65536色，也就是2的16次方；24位色被称为**真彩色**，它可以达到人眼分辨的极限，发色数是1677万多色，也就是2的24次方。

三、真彩色转变为256色

将真（全）彩色图像转化为256色图像会有大量的颜色信息损失掉。因此，在转换过程中要找到合适的映射关系，使得**变换后的256种颜色在原图像中最具代表性或出现的频率最高**。

两种常用的转化算法

- 中位切分法
- 流行色法

三、真彩色转变为256色

1. 中位切分法

中位切分法的基本过程是：

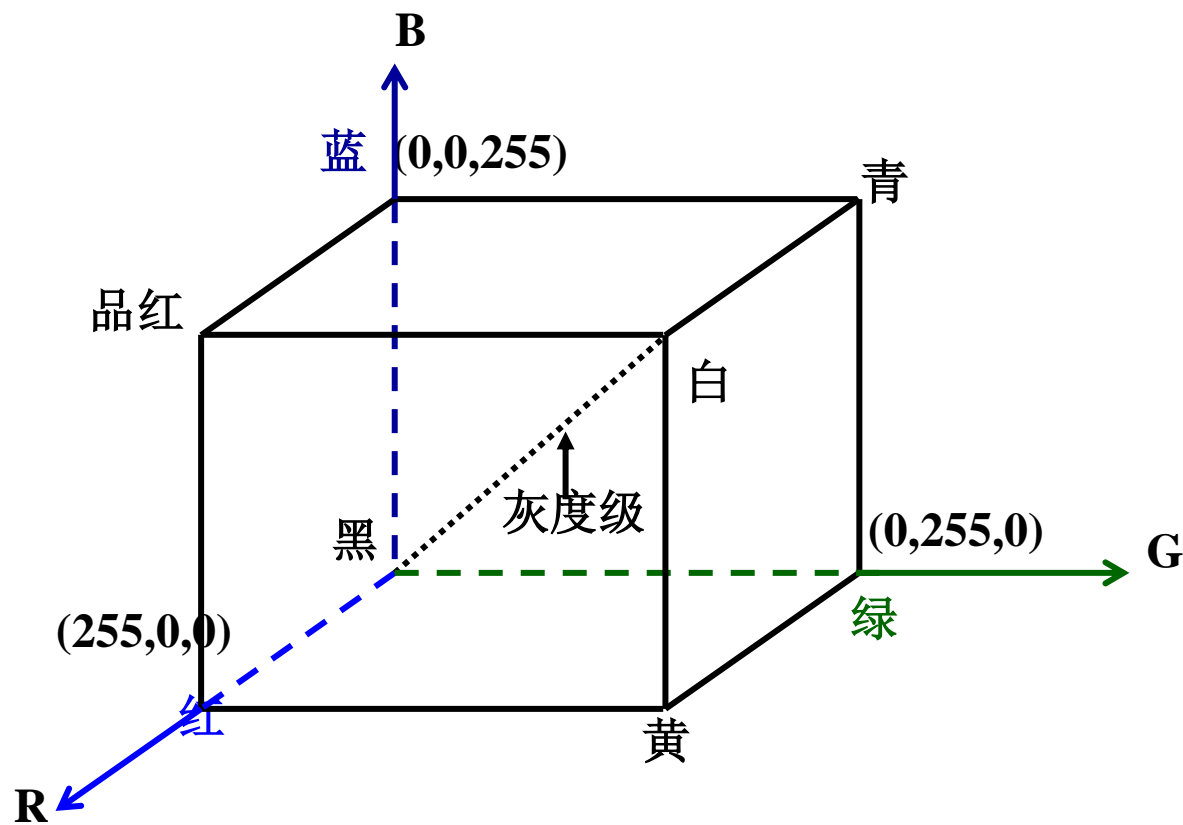
首先，将RGB彩色空间中的3个坐标轴进行均匀量化，将每个坐标轴分为256个级别，0为最暗，255为最亮，这样真彩色图像的各种颜色就可以用坐标空间的各个量化点来表示。

然后，将彩色立方体划分为256个小立方体，使各立方体包含相同的颜色数。

最后，求出这256个小立方体的中心点的颜色。

三、真彩色转变为256色

1. 中位切分法



三、真彩色转变为256色

1. 中位切分法(median-cut algorithm)



(a) 原真彩色图像



(b) 转变成的256色图像

图 利用中位切分法转换真彩色图像成256色图像的实例

三、真彩色转变为256色

2. 流行色法(popularity algorithm)

流行色算法的基本过程：

首先，对原彩色图像中各颜色出现的概率进行统计。

然后，按照由大到小的顺序选择出前256种颜色。

最后，将其它颜色按照与这256种颜色就近的原则进行转换，用这256种颜色代替原真彩色图像中的颜色。

三、真彩色转变为256色

2. 流行色法



(a) 原真彩色图像



(b) 转变成的256色图像

图 利用流行色法转换真彩色图像成256色图像的实例

三、真彩色转变为256色

转换效果比较:



原图



中位切分法转换成的256彩色结果



流行色法转换成的256彩色结果

四、彩色平衡

当颜色通道中不同的敏感度、增光因子、偏移量等使图像的3个分量发生了不同的线性变换时，就会导致图像的三基色“不平衡”。

彩色平衡就是通过对色彩偏移的图像进行色彩校正，也即通过调整图像的R、G、B三个分量的强度，恢复图像场景原始颜色特征的技术和过程。

常用的彩色平衡方法有：

- 白平衡法
- 颜色平均值最小法

四、彩色平衡

1. 白平衡法

所谓白平衡法，就是将景物中的白色物体在图像中还原为白色。

简单白平衡法的基本过程是：

首先，依据式（5.28）计算出色偏图像的亮度分量，求出图像的最大亮度 I_{\max} 和平均亮度 \bar{I} 。

$$I(x, y) = 0.299 \cdot f_R(x, y) + 0.587 \cdot f_G(x, y) + 0.114 \cdot f_B(x, y) \quad (5.28)$$

其次，设定一个较大的阈值 T （如0.95），求出图像中亮度值大于 $0.95I_{\max}$ 的点的集合。

四、彩色平衡

1. 白平衡法

然后，认为这些点对应实际场景中白色的点，计算出这些点对应的R、G、B分量的均值 \bar{R} 、 \bar{G} 、 \bar{B} ，依据公式（5.29）确定出白平衡法的调整参数 K_R 、 K_G 、 K_B ；最后，利用调整公式（5.30）对色偏图像进行调整。

$$k_R = \frac{\bar{I}}{\bar{R}} \quad k_G = \frac{\bar{I}}{\bar{G}} \quad k_B = \frac{\bar{I}}{\bar{B}} \quad (5.29)$$

$$\begin{bmatrix} G_R(x, y) \\ G_G(x, y) \\ G_B(x, y) \end{bmatrix} = \begin{bmatrix} k_R & 0 & 0 \\ k_G & 0 & 0 \\ k_B & 0 & 0 \end{bmatrix} \begin{bmatrix} f_R(x, y) \\ f_G(x, y) \\ f_B(x, y) \end{bmatrix} \quad (5.30)$$

四、彩色平衡

1. 白平衡法



(a) 原色偏图像



(b) 白平衡法结果图像

图 白平衡法对图像的彩色平衡

四、彩色平衡

```
% 白平衡法进行彩色平衡的MATLAB程序
clc;clear all;close all;
f0=imread('strawberries-RGB.tif');
[h,w,color]= size(f0);
I=f0;%色偏图像亮度分量图像初值
sum=0.0;
I_max=0.0;
for i = 1:h
    for j= 1:w
        I(i,j)=f0(i,j,1)*0.299 + f0(i,j,2)*0.587 + f0(i,j,3)*0.114;
        I1=double(I(i,j));
        sum= sum + I1;
        if I1>I_max
            I_max = I1;
        end
    end
end
I_ave=sum/(h* w);
I95=0.95 * I_max;
[I_r,I_g,I_b]=deal(0.0,0.0,0.0);
num=0;
```

四、彩色平衡

```
for i= 1:h
    %求色偏图像的平均亮度
    %设置阈值
    %给色偏分量和及个数赋初值
    for j= 1:w
        I1 = double(I(i,j));
        if I1>I95
            %求亮度值大于0.95*I_max的像素点的r、 g、
            b分量值的和
            I_r=I_r+ double(I(i,j,1));
            I_g=I_g + double(I(i,j,2));
            I_b=I_b + double(I(i,j,3));
            num =num+1;
        end
    end
end
end
```

```
R_ave=I_r/num;
G_ave=I_g/num;
B_ave= I_b/num;
kR=I_ave/R_ave;
kG= I_ave/G_ave;
kB= I_ave/B_ave;
f1(:,:,1)= f0(:,:,1)* kR;
f1(:,:,2)= f0(:,:,2)* kG;
f1(:,:,3)= f0(:,:,3)* kB;
%计算分量和的平均值
%计算调整系数
%按式(10.30)对色偏图像进行调整
%显示原图像
subplot(1,2,1);imshow(f0);title('原色偏图像', 'fontsize',14);
subplot(1,2,2);imshow(f1,[]);title('白平衡法彩色平衡图像', 'fontsize',14);
```

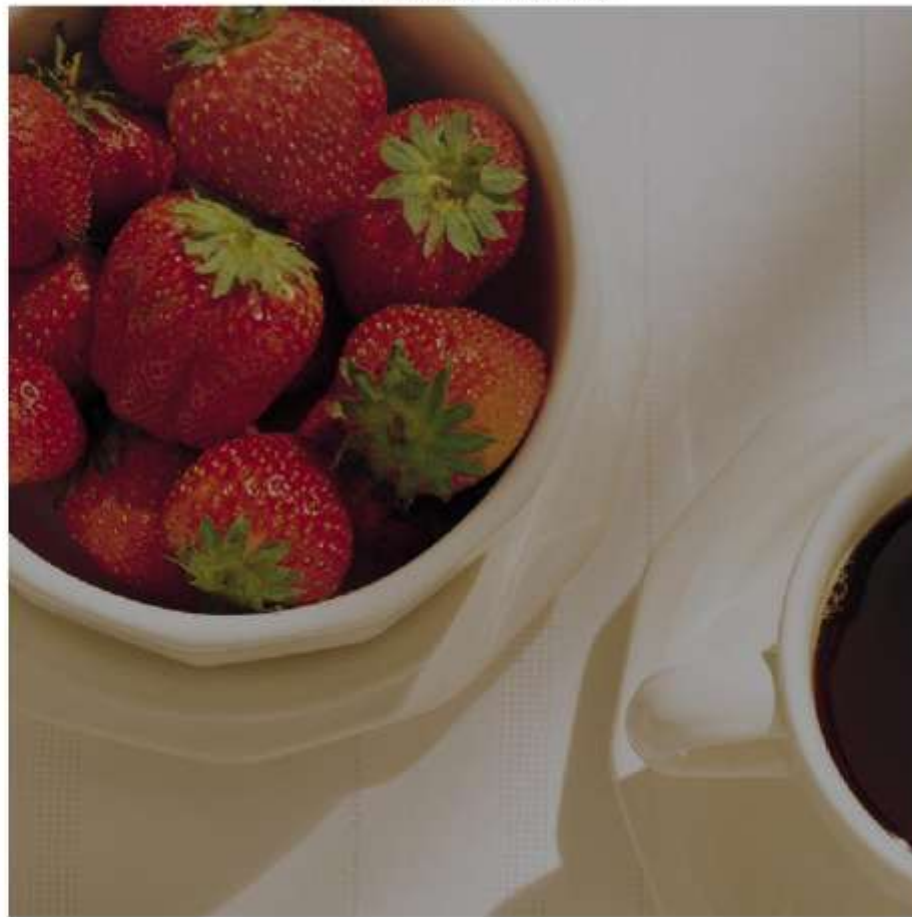

四、彩色平衡

1. 白平衡法

原色偏图像



白平衡法彩色平衡图像



四、彩色平衡

2. 颜色平均值最小法

基本思想：

寻找彩色图像中较强的颜色通道（彩色图像中，所有像素的某个彩色分量，构成了该分量的彩色通道），通过对较强的颜色通道进行抑制和对较弱的颜色通道进行增强来达到彩色平衡的目的。

四、彩色平衡

2. 颜色平均值最小法

颜色平均值最小法的基本过程是：

首先，首先计算色偏图像各彩色通道的像素平均值 R_{mean} 、 G_{mean} 、 B_{mean} ，并求它们中的最小值 $S_{RGB} = \min(R_{\text{mean}}, G_{\text{mean}}, B_{\text{mean}})$ 。

其次，统计各颜色通道中像素值大于 S_{RGB} 的像素点的个数 N_R 、 N_G 、 N_B ，并求它们中的最大值 $N_{\text{max}} = \max(N_R, N_G, N_B)$ ，该值对应的颜色通道即为颜色信息较强的通道。

四、彩色平衡

2. 颜色平均值最小法

颜色平均值最小法的基本过程是：

然后，将3个颜色通道的像素值按照从大到小的顺序排列直到它们的个数为 N_{\max} 时为止，从而形成3个颜色通道像素值倒排序向量，并将红、绿、蓝3个颜色通道向量中第 N_{\max} 个元素的值 T_R 、 T_G 、 T_B 作为阈值（一般情况下，信息最弱的颜色通道的阈值较小）。

四、彩色平衡

2. 颜色平均值最小法

颜色平均值最小法的基本过程是：

接着，依据式（5.31）确定出颜色平均值最小法的调整参数 k_R 、 k_G 、 k_B 。

最后，利用调整公式（5.32）对色偏图像进行调整。

$$k_R = \frac{S_{RGB}}{T_R} \quad k_G = \frac{S_{RGB}}{T_G} \quad k_B = \frac{S_{RGB}}{T_B} \quad (5.31)$$

$$\begin{bmatrix} G_R(x, y) \\ G_G(x, y) \\ G_B(x, y) \end{bmatrix} = \begin{bmatrix} k_R & 0 & 0 \\ 0 & k_G & 0 \\ 0 & 0 & k_B \end{bmatrix} \begin{bmatrix} f_R(x, y) \\ f_G(x, y) \\ f_B(x, y) \end{bmatrix} \quad (5.32)$$

四、彩色平衡

2. 颜色平均值最小法



(a) 原色偏图像



(b) 颜色平均值最小法结果图像

图 颜色平均值最小法对色偏图像彩色平衡的结果

四、彩色平衡

%颜色平均值最小法对色片彩色图像进行彩色平衡

clc;

clear all;

close all;

img0= imread('strawberries-RGB.tif');

[h, w, color]= size(img0);

I= img0;

[R_sum,G_sum,B_sum]=deal(0.0,0.0,0.0);

for i= 1:h

for j= 1:w

%色偏图像亮度分量图像初值

R_sum = R_sum + double(I(i,j,1));

G_sum=G_sum + double(I(i,j,2));

B_sum=B_sum+ double(I(i,j,3));

end

end

S_rgb=min(min(R_sum,G_sum),B_sum);

S_rgb=S_rgb/(h* w);

[N_r,N_g,N_b]= deal(0,0,0);%给色偏分量和及个数赋初值

for i = 1:h%统计各颜色通道中大于S_rgb的像素

for j= 1:w

%求各颜色通道像素值和最小者

%求各颜色通道像素值和平均值最小

if double(I(i,j,1))>S_rgb

N_r=N_r+1;

end

if double(I(i,j,2))> S_rgb

N_g=N_g+ 1;

end

if double(I(i,j,3))> S_rgb

N_b=N_b + 1;

end

end

end

四、彩色平衡

`N_max= max(max(N_r,N_g),N_b);% 求大于S_rgb的像素个数的最大者`

`Aary=I(1:h,1:w,1);Vect= Aary(:);% 阵列ary中的全部像素赋给向量Vect`

`Ord=fliplr(Vect);% 向量中元素按降序排列`

`T_r= Ord(N_max);% 将向量中第N_max个元素赋给T_r`

`Aary=I(1:h,1:w,2); % 阵列ary中的全部像素赋给向量Vect`

`Vect = Aary(:);% 向量中元素按降序排列`

`Ord=fliplr(Vect);T_g = Ord(N_max);% 将向量中第N_max个元素赋给T_g`

`Aary=I(1:h,1:w,3);`

`Vect = Aary(:);% 阵列Aary中的全部像素赋给向量vect`

`Ord= fliplr(Vect);% 向量中元素按降序排列`

`T_b= Ord(N_max);% 将向量中第N_max个元素赋给T_b`

`kR=S_rgb/double(T_r);`

`kG=S_rgb/double(T_g);`

`kB=S_rgb/double(T_b);`

`% 按式(10.31)计算调整系数`

`f1(:, :,1)= img0(:, :,1)*kR;% 按式(10.32)对色偏图像进行调整`

`f1(:, :,2)= img0(:, :,2)* kG;`

`f1(:, :,3)= img0(:, :,3)* kB;`

`subplot(1,2,1);imshow(img0);title('原色偏彩色图像`

`','fontsize',14);% 显示原色偏彩色图像`

`subplot(1,2,2);imshow(f1,[]);title('颜色平均值最小法彩色平衡图像','fontsize',14);`

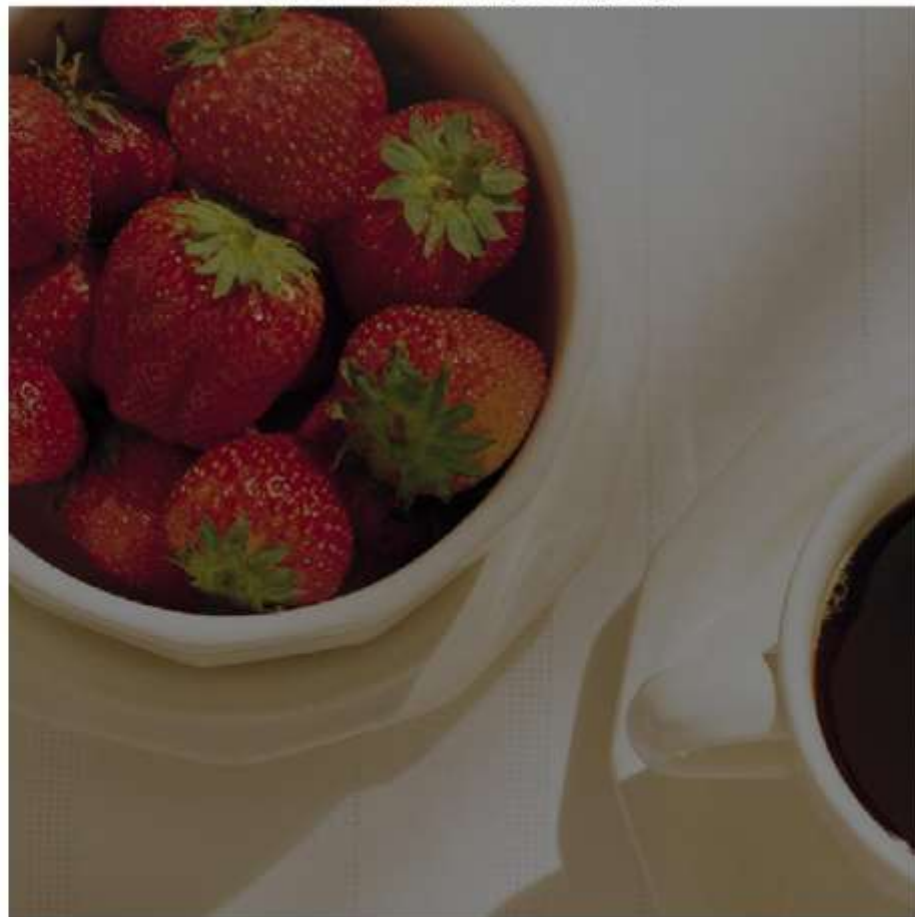
四、彩色平衡

2. 颜色平均值最小法

原色偏彩色图像



颜色平均值最小法彩色平衡图像



目录

- 彩色图像基础
- 彩色变换
- 彩色图像增强
- 彩色图像平滑
- 彩色图像锐化
- 彩色图像分割

彩色图像增强

在获取的彩色图像中，可能会存在对比度低、颜色偏暗、局部细节不明显等问题，为了改善图像的视觉效果、突出图像的特征，利于进一步的处理，需要对图像进行增强处理。

彩色图像增强依据处理对象的不同可分为：

- 真彩色增强
- 伪彩色增强
- 假彩色增强

一、真彩色增强

真彩色增强的处理对象是具有 2^{24} 种颜色的彩色图像（又称全彩色图像）。为了避免破坏图像的彩色平衡，真彩色的增强通常选择在HSI模型下进行。

依据选择增强分量和增强目的的不同，可将真彩色增强分为：

- 亮度增强
- 色调增强
- 饱和度增强

一、真彩色增强

1. 亮度增强

彩色图像的亮度增强是仅对彩色图像的**亮度分量**进行处理的增强方法，它的目的是通过对图像亮度分量的调整使得图像在合适的亮度上提供最大的细节。

亮度增强可以在其亮度分量上使用灰度图像的增强算法，如灰度变换法、直方图增强法等。

一、真彩色增强

```
clear;close all;clc;
f=imread('miroslava.jpg'); % 读取图像
f=im2double(f);
R=f(:,:,1); % R基色
G=f(:,:,2); % G基色
B=f(:,:,3); % B基色
R_equ=histeq(R); % 均衡化
G_equ=histeq(G);
B_equ=histeq(B);
f_equ=cat(3,R_equ,G_equ,B_equ); % 整合
```

```
hsi_f=rgb2hsi(f); % 转换成HSI彩色空间
H=hsi_f(:,:,1); % 色度分量
S=hsi_f(:,:,2); % 饱和度分量
I=hsi_f(:,:,3); % 亮度分量
I_equ=histeq(I); % 均衡化
hsi_f_equ=cat(3,H,S,I_equ); % 整合
fI_equ=hsi2rgb(hsi_f_equ); % 转换回RGB空间

subplot(131),imshow(f),title('原图')
subplot(132),imshow(f_equ),title('RGB均衡化')
subplot(133),imshow(fI_equ),title('RGB转HSI后I均衡化')
```

一、真彩色增强

原图



RGB均衡化



RGB均衡化后I均衡化

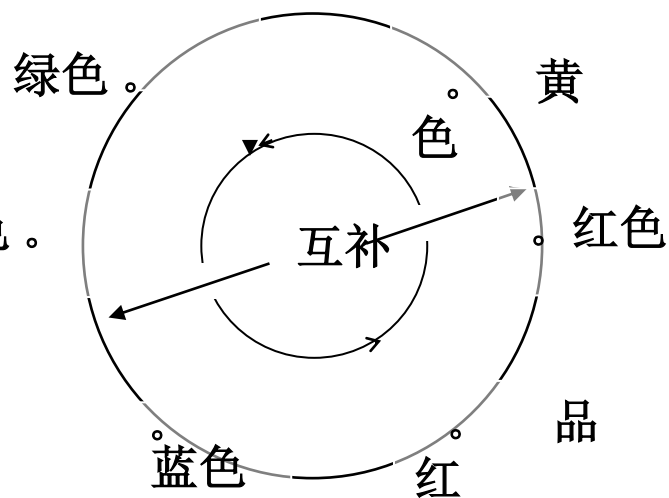


一、真彩色增强

2. 色调增强

色调增强是通过**增加颜色间的差异**来达到图像增强的目的，一般可以通过对彩色图像每个点的色度值加上或减去一个常数来实现。

由于彩色图像的色度分量是一个**角度值**，因此对色度分量加上或减去一个常数，相当于图像上所有点的颜色都沿着下图的彩色环逆时针或顺时针旋转一定的角度。由于彩色处理色相分量图像的操作必须考虑灰度级的“**周期性**”，即对色调值加上 120° 和加上 480° 是相同的。



一、真彩色增强

2. 色调增强



原图像



(a) 色度值加上120的图像



(b) 色度值减去120的图像

图 真彩色图像的色度增强实例

一、真彩色增强

3. 饱和度增强、

饱和度增强可以使彩色图像的颜色更为鲜明。饱和度增强可以通过对彩色图像每个点的饱和度值乘以一个大于1的常数来实现；反之，如果对彩色图像每个点的饱和度值乘以小于1的常数，则会减弱原图像颜色的鲜明程度。



原图像



(a) 饱和度值乘以3的图像



(b) 饱和度值乘以0.3的图像

图 真彩色图像的饱和度增强实例

二、伪彩色增强

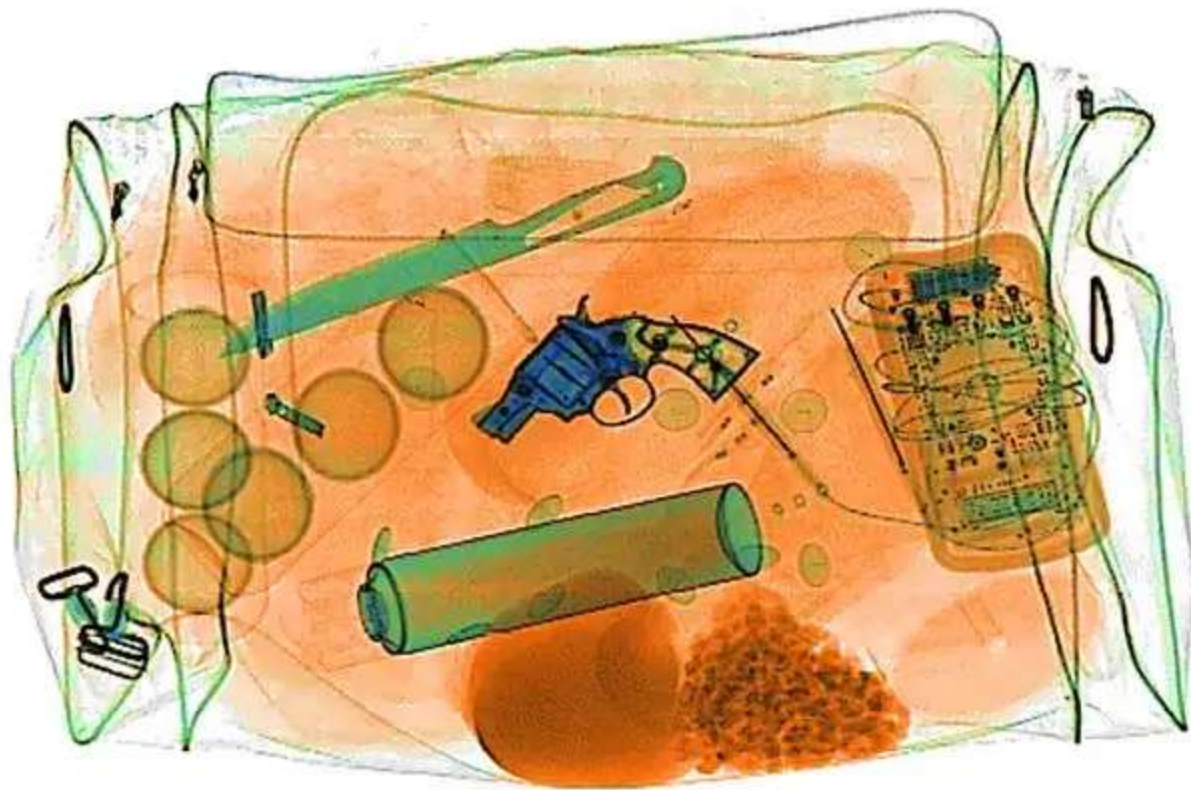
- ✓ 人类只能识别几十级灰度值，但是能分辨上千种颜色，通过使用彩色可以增加人类所能辨识的目标种类，
- ✓ 对彩色图像的**细节**能了解得更清楚
- ✓ 彩色还能增强图像的**活泼性**，减少厌倦感并增加安检人员的关注度

二、伪彩色增强

在遥感、医学、安全检查等图像处理中，为了直观地观察和分析图像数据，常采用将灰度图像映射到彩色空间的方法，突出兴趣区域或待分析的数据段。

- ✓ 不改变像素的几何位置，而仅改变其显示的颜色。
- ✓ 伪彩色增强作为一种实用的图像增强技术，主要用于提高人眼对图像的分辨能力。
- ✓ 可以用计算机来完成，也可以用专用硬件设备来实现。

二、伪彩色增强






X光安检机成像

二、伪彩色增强

X射线穿过物体时，一部分X射线被物体吸收，另一部分则通过物体继续传播。探测器将剩余的X射线能量转变为信号，这些信号被放大，并送到信号处理（计算机）做进一步处理，再经过图像处理系统后，**根据不同物质对X射线吸收程度的不同，采用颜色分析法是基于不同的物质材料在X射线图像下呈现出不同颜色，从而进行物质类别的辨识**。监视器上出现的各种颜色，是物体密度、质量和数量的反映，因此，可根据图像颜色的深浅来对物体的品质进行评估。

二、伪彩色增强

类别	颜色	典型物质
有机物		含氢、碳、氮、氧的物质，如糖
混合物和轻金属		含钠、硅、氯的物质，如盐， <u>轻金属</u> ，如铝等
无机物		如铁，铜，银等

一般情况下，食品、塑料等有机物是橙色。书本、陶瓷等无机物显示为绿色，混合物也多呈现绿色，比如不锈钢制品。金属则显示为蓝色，电击枪、刀具则是呈现深浅不一的蓝色。

二、伪彩色增强

基本原理： 将灰度图像或者单色图像的各个灰度级匹配到彩色空间中的一点，从而使单色图像映射成彩色图像。

- ✓ 设 $f(x,y)$ 为一幅灰度图像， $R(x,y)$ ， $G(x,y)$ ， $B(x,y)$ 为 $f(x,y)$ 映射到RGB空间的三个颜色分量： $R(x,y)=f_R(f(x,y))$ ， $G(x,y)=f_G(f(x,y))$ ， $B(x,y)=f_B(f(x,y))$ ，其中 f_R ， f_G ， f_B 为某种映射函数。
- ✓ 给定**不同的映射函数**就能将灰度图像转化为不同的伪彩色图像。

二、伪彩色增强

- 伪彩色增强的处理对象是灰度图像，将一幅具有不同灰度级的图像通过一定的映射转变为彩色图像，来达到增强人对图像的分辨能力。
- 伪彩色增强可分为空域增强和频域增强两种，在这两种算法中，密度分层法、灰度级-彩色变换法和频率滤波法是三种较为常用的算法。

二、伪彩色增强

1. 密度分层法

密度分层法（又称强度分层法）是将灰度图像中任意一点的灰度值看作该点的密度函数。

密度分层法的基本过程是：

首先，用平行于坐标平面的平面序列 L_1, L_2, \dots, L_N 将密度函数分割为多个互相分隔的灰度区间。然后，给每一区域分配一种颜色。这样就能够将一幅灰度图像映射为彩色图像。

二、伪彩色增强

1. 密度分层法

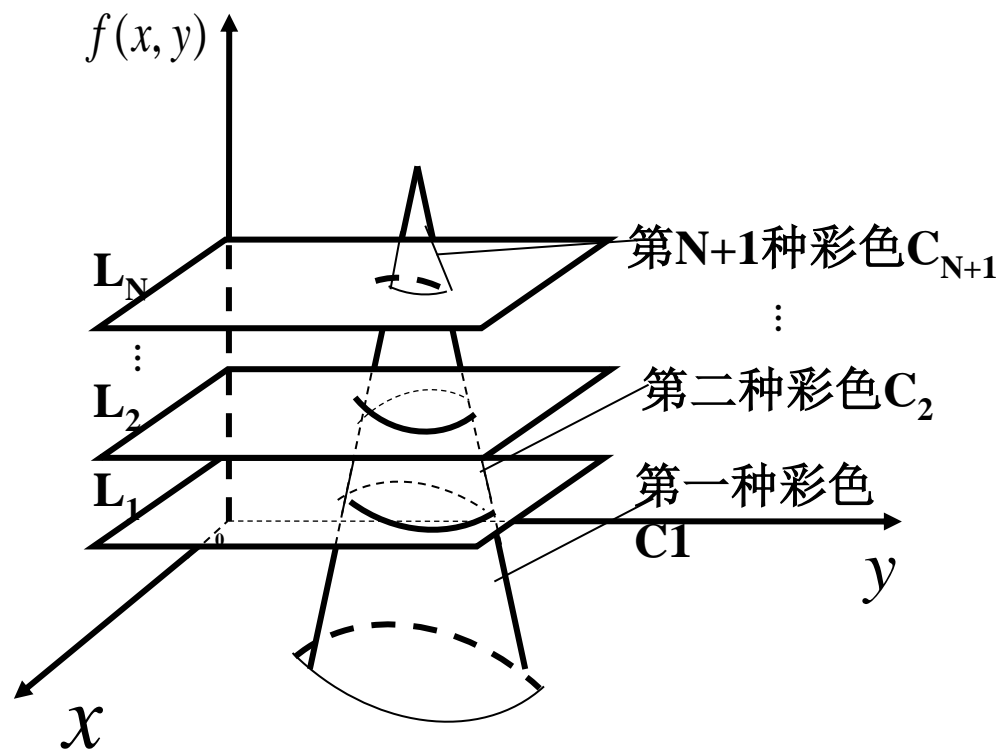


图 密度分层法空间示意

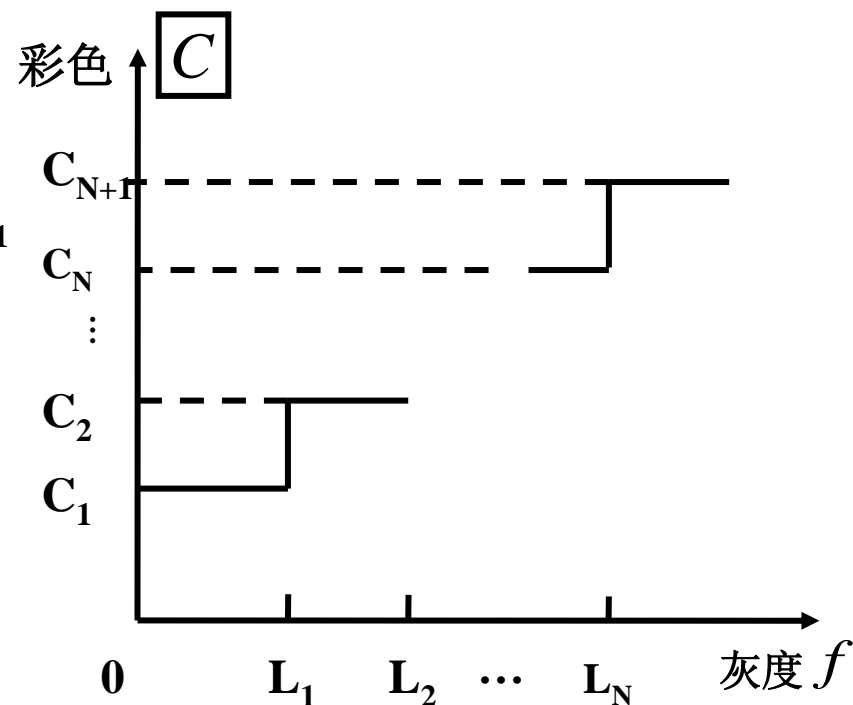
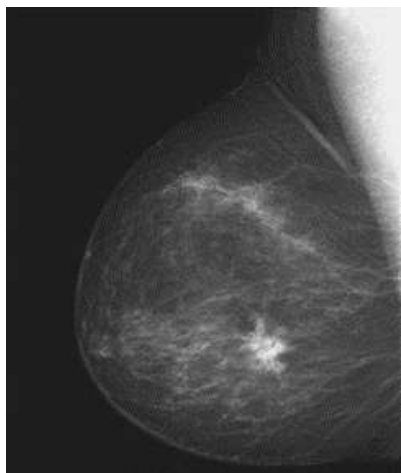
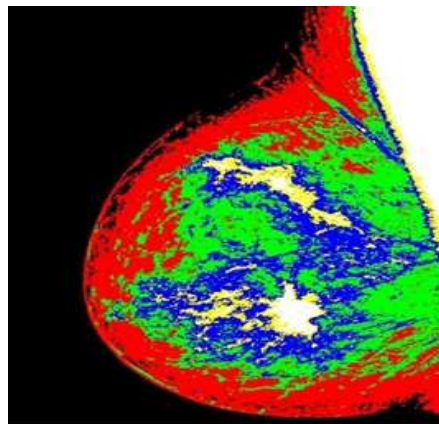


图 密度分层法平面示意图

二、伪彩色增强



乳房X光的灰度图像



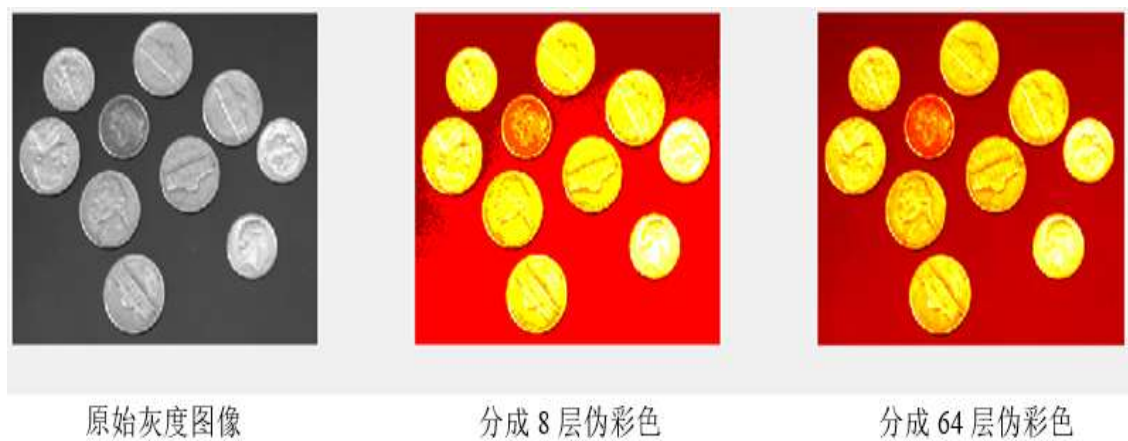
使用6种颜色灰度分层后的结果

原本很难看出轮廓变化的灰度图像，经过灰度分层法变换为伪彩色图像后，轮廓被显示出来。通过灰度分层法，改变颜色的数量和灰度区间大小，可以快速了解灰度图像中目标物体的灰度变换特性和轮廓。

二、伪彩色增强

1. 密度分层法

```
I=imread('coins.png');  
GS8=grayscale(I,8);  
GS64=grayscale(I,64);  
subplot(1,3,1),imshow(I),title('原始灰度图像');  
subplot(1,3,2),imshow(GS8, hot(8)),title('分成8层伪彩色');  
subplot(1,3,3),imshow(GS64, hot(64)),title('分成64层伪彩色');
```



二、伪彩色增强

2. 灰度级-彩色变换法

灰度级-彩色变换伪彩色增强法基本思想：对图像中每个像素点的灰度值采用不同的变换函数进行3个独立的变换，并将结果映射为彩色图像的R、G、B分量值，然后对3个通道合并就可以得到一幅RGB空间上的彩色图像。

由于灰度级-彩色变换法在变换过程中用到了三基色原理，与密度分层法相比，该算法可有效地拓宽结果图像的颜色范围。

二、伪彩色增强

2. 灰度级-彩色变换法

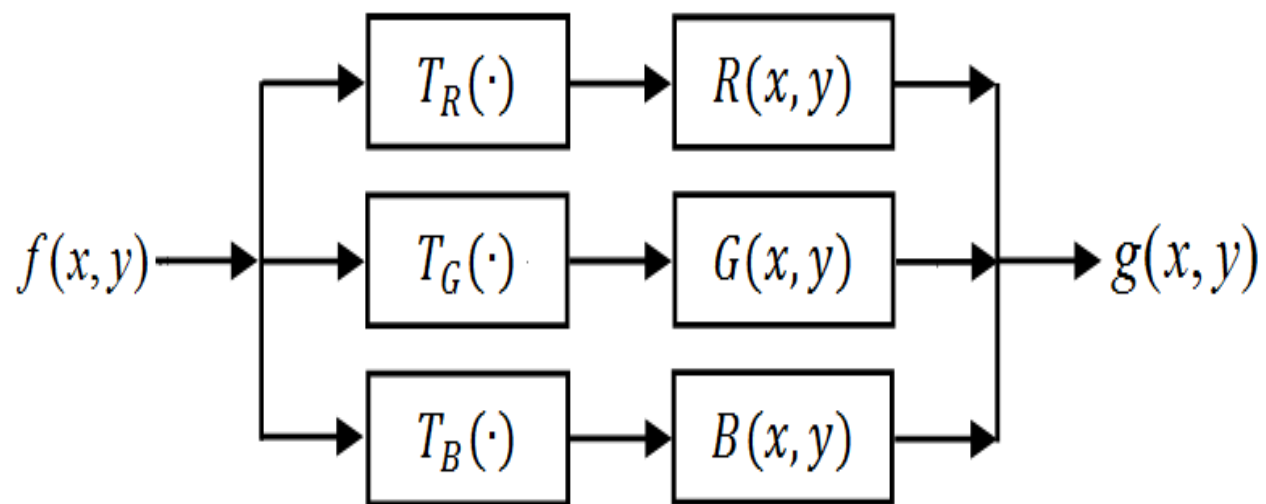


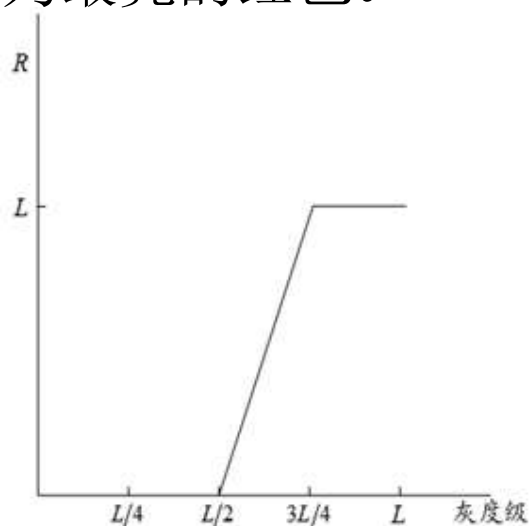
图 灰度级-彩色变换法示意图

二、伪彩色增强

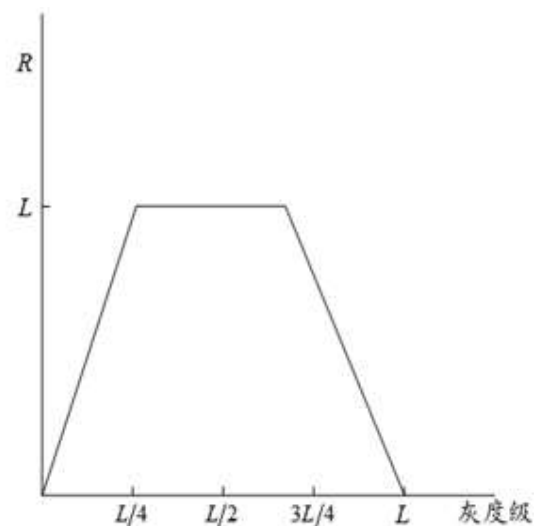
2. 灰度级-彩色变换法

下图中的三幅图形依次表示了红色分量、绿色分量和蓝色分量的传递函数。从

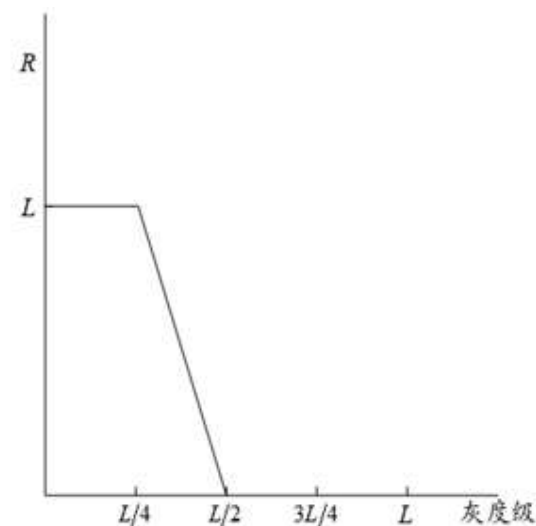
(a) 图中可以看出，任何灰度小于 $L/2$ 的像素都会被转换为尽量暗的红色；而灰度在 $L/2$ 和 $3L/4$ 之间的像素是红色由暗到亮的线性变换；所有灰度大于 $3L/4$ 的像素都被转变为最亮的红色。



(a) 红色分量的传递函数



(b) 绿色分量的传递函数



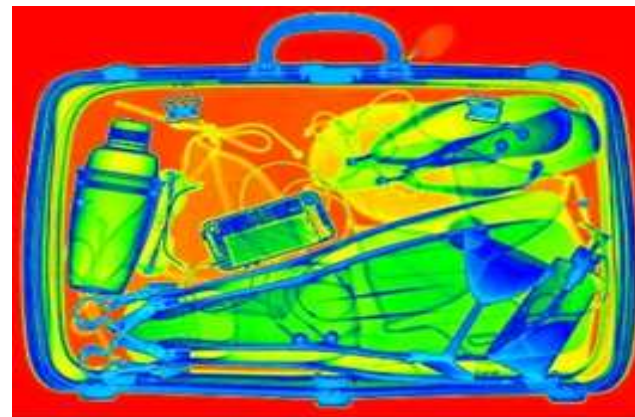
(c) 蓝色分量的传递

图 典型的传递函数

二、伪彩色增强



行李箱的X光扫描灰度图像



灰度级到彩色变换后的结果图像

原灰度图像中可以发现物品之间重叠导致难以分辨，通过对原图像分别在红、绿、蓝三个通道上进行对应的预设函数变换，再对三个输出合成一幅伪彩色图像。

二、伪彩色增强

2. 灰度级-彩色变换法

```
I= imread ( 'moon.tif');           %读取图像
I=double(I);                       %转换成double型，便于计算
[m,n]=size(I);                     %得到图像的高度和宽度
L=256;                             %灰度的最大等级数为L=256
for i=1:m
    for j=1:n                       %二重循环，对每个点的值进行变换
        if I(i,j)<L/4               %当灰度值I<L/4时
            R(i,j)=0;               %R=0
            G(i,j)=4*I(i,j);        %G=4*I
            B(i,j)=L;               %B=L
        else if I(i,j)<=L/2         %当(L/4) =<I<=(L/2)时
            R(i,j)=0;               %R=0
            G(i,j)=L;               %G=L
            B(i,j)=-4*I(i,j)+2*L;   %B=(-4)*I+2*L
        end
    end
end
```


二、伪彩色增强

2. 灰度级-彩色变换法

```
else if I(i,j)<=3*L/4      % 当(L/2)<I<=(3L/4)时
    R(i,j)=4*I(i,j)-2*L;    % R=4*I-2L
    G(i,j)=L;              % G=L
    B(i,j)=0;              % B=0
else                        % 当I>(3L/4)时
    R(i,j)=L;              % R=L
    G(i,j)= -4*I(i,j)+4*L; % G=(-4)*I+4L
    B(i,j)=0;              % B=0
end
end
end
end
end
```

二、伪彩色增强

2. 灰度级-彩色变换法

```
for i=1:m
    for j=1:n                %二重循环，对每个点赋值
        G2C(i,j,1)=R(i,j); %将R的值作为G2C的第一个分量
        G2C(i,j,2)=G(i,j); %将G的值作为G2C的第二个分量
        G2C(i,j,3)=B(i,j); %将B的值作为G2C的第三个分量
    end
end
G2C=G2C/256;                %值归一化
figure,imshow(G2C);         %显示变换得到的伪彩色图像
```

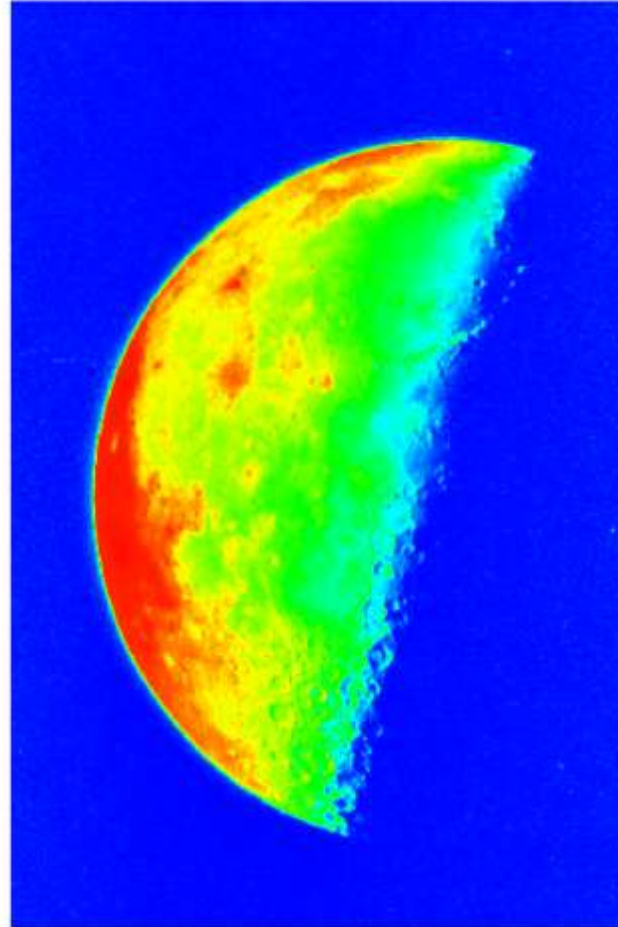
二、伪彩色增强

2. 灰度级-彩色变换法

原始图像



灰度值到彩色变换后图像



二、伪彩色增强

3. 频率滤波法

频率滤波伪彩色增强法基本思想：首先对原灰度图像进行傅立叶变换，然后用3种不同的滤波器分别对得到的频率（谱）图像进行独立的滤波处理，处理完后再用傅立叶逆变换将得到三种不同频率的图像映射为单色图像，经过一定的后处理，最后将这三幅灰度图像分别映射为彩色图像的R、G、B分量，这样就可以得到一幅RGB空间上的彩色图像。

与密度分层法和灰度级-彩色变换法两种算法相比，频率滤波法输出的伪彩色与灰度图像的灰度级无关，仅与灰度图像不同空间频率成分有关。

二、伪彩色增强

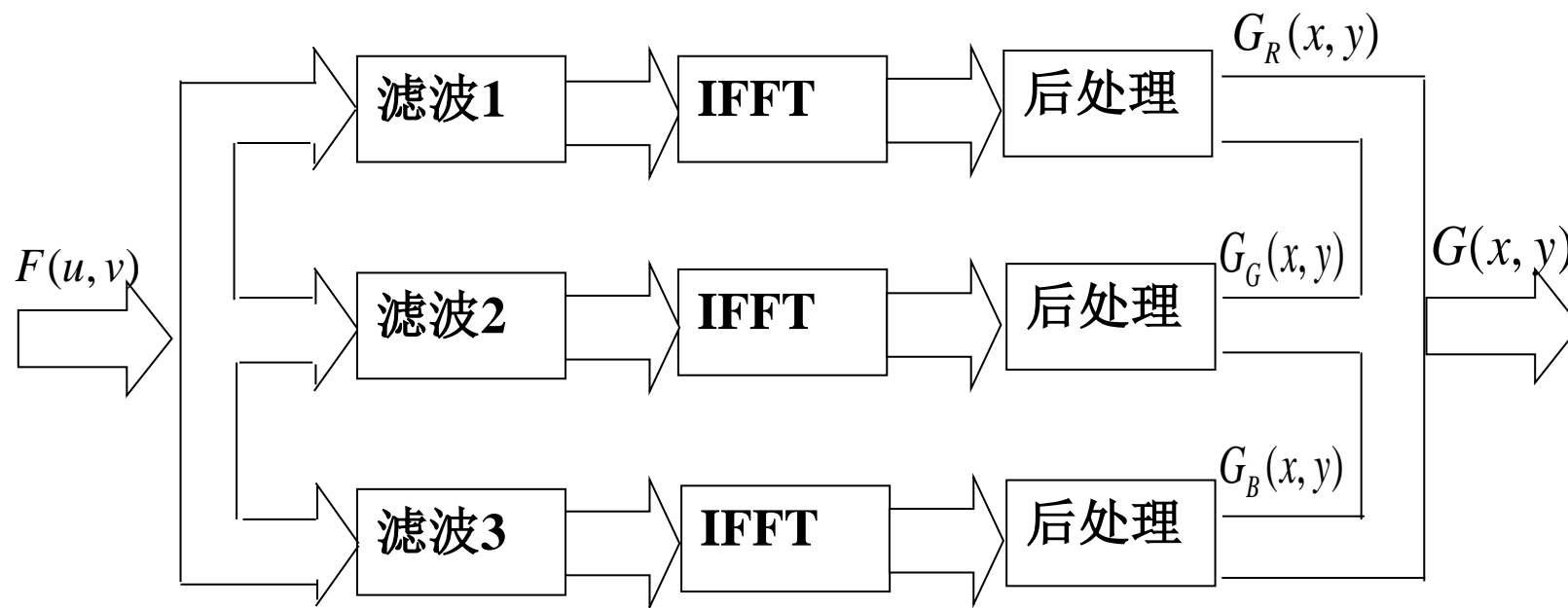


图 频率滤波法变换示意图

二、伪彩色增强

3. 频率滤波法



(a) 原灰度图像



(b) 得到的伪彩色图像

图 频率滤波法增强实例

三、假彩色增强

假彩色增强:

从一幅初始的彩色图像或者从多谱图像的波段中生成增强的彩色图像的一种方法。假彩色图像处理允许可见光以外的光谱也转换为彩色图像的向量分量，例如红外图像。

假彩色增强的实质是从一幅彩色图像映射到另一幅彩色图像，由于得到的彩色图像不再能反映原图像的真实色彩，因此称为假彩色增强。

三、假彩色增强

假彩色处理

处理对象是三基色描绘的**自然图像或同一景物的多光谱图像**。对自然图像，假彩色的处理方法之一是将人们所关注的目标物映射为与原色不同的假彩色，即原有的彩色图像变换成给定彩色分布的图像。

伪彩色增强

是将**灰度或单一波段的图像**变换为彩色图像，从而将人眼不能区分的微小的灰度差别显示为明显的色彩差异，更便于解译和提取有用信息。彩色图像中的彩色根据黑白图像的灰度级或其他图像特征（如空间频率成分）人为给定。

三、假彩色增强

假彩色增强的意义在于：

- ✓ 画家通常将图像中的景物赋以与**现实不同的颜色**，以达到引人注目的目的。
- ✓ 对于一些细节特征不明显的彩色图像，可以利用假彩色增强将这些细节赋以人眼敏感的颜色，以达到辨别图像细节的目的。
- ✓ 在遥感技术中，利用假彩色图像可以将多光谱图像合成彩色图像，使图像看起来逼真、自然，有利于对图像进行后续的分析与解译。

三、假彩色增强

假彩色增强可以看作是一个从原图像到新图像的线性坐标变换。

例如，真彩色图像可以按照下面的公式转换为假彩色图像。

$$\begin{bmatrix} R_g \\ G_g \\ B_g \end{bmatrix} = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 \\ \alpha_2 & \beta_2 & \gamma_2 \\ \alpha_3 & \beta_3 & \gamma_3 \end{bmatrix} \begin{bmatrix} R_f \\ G_f \\ B_f \end{bmatrix} \quad (5.33)$$

目录

- 彩色图像基础
- 彩色变换
- 彩色图像增强
- 彩色图像平滑
- 彩色图像锐化
- 彩色图像分割

彩色图像平滑

- 彩色图像的平滑处理相对比较复杂
- 处理的对象是向量，并且图像所用的彩色空间不同，向量表示的含义也不同。

基于RGB彩色模型的彩色图像平滑

令 S_{xy} 表示在RGB彩色图像中定义一个中心在 (x,y) 的邻域的坐标集，在该邻域中R、G、B分量的平均值为：

$$\bar{c}(x,y) = \frac{1}{K} \sum_{(x,y) \in S_{xy}} c(x,y)$$

令 c 代表RGB彩色空间中的任意向量， $c(x,y)$ 的分量是一幅彩色图像在一点上的RGB分量。彩色分量是坐标 (x,y) 的函数，表示为：

$$c(x,y) = \begin{pmatrix} c_R(x,y) \\ c_G(x,y) \\ c_B(x,y) \end{pmatrix} = \begin{pmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{pmatrix}$$

基于RGB彩色模型的彩色图像平滑

按照上式可得：

$$\bar{c}(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \frac{1}{K} \sum_{(\mathbf{x}, \mathbf{y}) \in S_{xy}} R(x, y) \\ \frac{1}{K} \sum_{(\mathbf{x}, \mathbf{y}) \in S_{xy}} G(x, y) \\ \frac{1}{K} \sum_{(\mathbf{x}, \mathbf{y}) \in S_{xy}} B(x, y) \end{pmatrix} \quad (5.11)$$

可以看出，如标量图像一样，可以通过传统的灰度邻域分别对RGB图像的每个平面单独进行平滑处理得到该向量分量。

结论：可以在每个彩色平面的基础上进行邻域平均值平滑，其结果与用RGB彩色向量执行平均是相同。平滑滤波可以使图像模糊化，从而减少图像中的噪声。

基于RGB彩色模型的彩色图像平滑

```
rgb=imread('yellowlily.jpg');    % 读取图像
fR=rgb(:,:,1);                  % 图像红色分量
fG=rgb(:,:,2);                  % 图像绿色分量
fB=rgb(:,:,3);                  % 图像蓝色分量
w=fspecial('average');          % 均值滤波模板
fR_filtered=imfilter(fR,w);      % 对图像红色分量滤波
fG_filtered=imfilter(fG,w);      % 对图像绿色分量滤波
fB_filtered=imfilter(fB,w);      % 对图像蓝色分量滤波
rgb_filtered=cat(3,fR_filtered,fG_filtered,fB_filtered);
```

```
% 将滤波后的三个分量组合得到新的彩色图像
subplot(3,3,1),imshow(fR),title('(a)红色分量')
subplot(3,3,2),imshow(fG),title('(b)绿色分量')
subplot(3,3,3),imshow(fB),title('(c)蓝色分量')
subplot(3,3,4),imshow(fR_filtered),title('(d)红色分量滤波')
subplot(3,3,5),imshow(fG_filtered),title('(e)绿色分量滤波')
subplot(3,3,6),imshow(fB_filtered),title('(f)蓝色分量滤波')
subplot(3,3,7),imshow(rgb),title('(g)原始图像')
subplot(3,3,8),imshow(rgb_filtered),title('(h)平滑后图像')
```

基于RGB彩色模型的彩色图像平滑



(a) R 分量



(b) G 分量



(c) B 分量



(d) R 分量滤波结果



(e) G 分量滤波结果



(f) B 分量滤波结果



(g) 原始彩色图像



(h) 彩色图像平滑结果

目录

- 彩色图像基础
- 彩色变换
- 彩色图像增强
- 彩色图像平滑
- 彩色图像锐化
- 彩色图像分割

彩色图像的锐化

■ 彩色图像锐化

锐化的主要目的是突出图像的细节。对于采用RGB模型的图像，利用拉普拉斯图像锐化算子对输入向量各分量进行拉普拉斯微分，就形成点(x,y)处的颜色向量的拉普拉斯微分：

$$\nabla^2[\mathbf{c}(\mathbf{x},\mathbf{y})]=\begin{Bmatrix} \nabla^2[\mathbf{R}(\mathbf{x},\mathbf{y})] \\ \nabla^2[\mathbf{G}(\mathbf{x},\mathbf{y})] \\ \nabla^2[\mathbf{B}(\mathbf{x},\mathbf{y})] \end{Bmatrix}$$

通过分别计算每一分量图像的Laplacian去计算全彩色图像的Laplacian。

彩色图像的锐化

使用经典的Laplacian滤波模板分别对每个分量图像进行锐化。MATLAB程序实现如下。

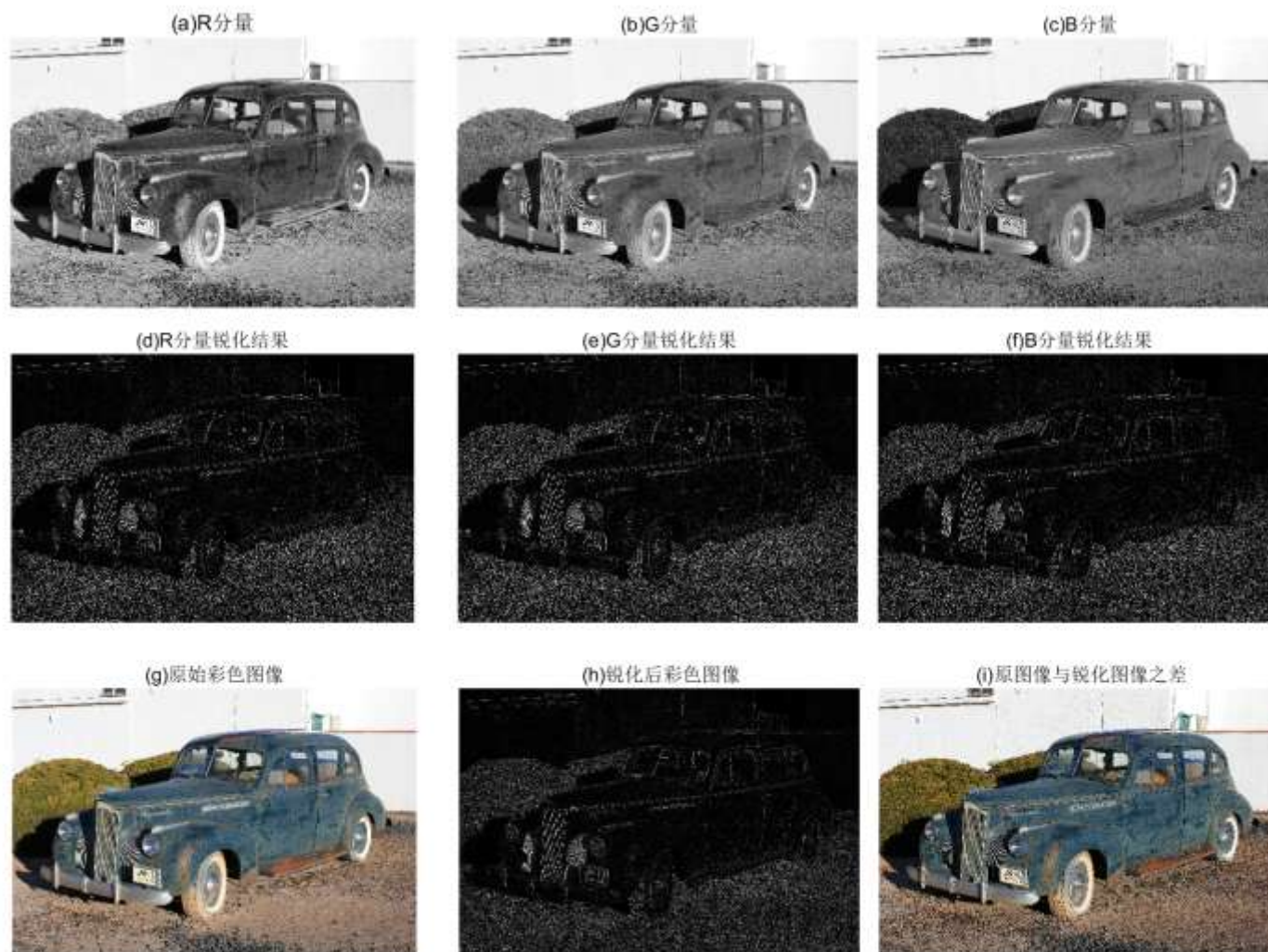
```
rgb= imread('car1.jpg');    % 读取图像
fR=rgb(:,:,1);              % 图像的红色分量
fG=rgb(:,:,2);              % 图像的绿色分量
fB=rgb(:,:,3);              % 图像的蓝色分量
lapMatrix=[1 1 1;1 -8 1;1 1 1];    %Laplacian滤波模板
fR_tmp= imfilter(fR,lapMatrix,'replicate'); % 行号对图像红色分量锐化滤波
fG_tmp= imfilter(fG,lapMatrix,'replicate'); % 行号对图像绿色分量锐化滤波
fB_tmp= imfilter(fB,lapMatrix,'replicate'); % 行号对图像蓝色分量锐化滤波
rgb_tmp=cat(3,fR_tmp,fG_tmp,fB_tmp); % 滤波后三分量组合
rgb_sharped=imsubtract(rgb,rgb_tmp); % 原图像与锐化图像之差
```

彩色图像的锐化

使用经典的**Laplacian**滤波模板分别对每个分量图像进行锐化。

```
subplot(3,3,1),imshow(fR),title('(a)R分量', 'fontsize',12)
subplot(3,3,2),imshow(fG),title('(b)G分量')
subplot(3,3,3),imshow(fB),title('(c)B分量')
subplot(3,3,4),imshow(fR_tmp),title('(d)R分量锐化结果')
subplot(3,3,5),imshow(fG_tmp),title('(e)G分量锐化结果')
subplot(3,3,6),imshow(fB_tmp),title('(f)B分量锐化结果')
subplot(3,3,7),imshow(rgb),title('(g)原始彩色图像')
subplot(3,3,8),imshow(rgb_tmp),title('(h)锐化后彩色图像')
subplot(3,3,9),imshow(rgb_sharped),title('(i)原图像与锐化图像之差')
```

彩色图像的锐化



目录

- 彩色图像基础
- 彩色变换
- 彩色图像增强
- 彩色图像平滑
- 彩色图像锐化
- 彩色图像分割

彩色图像分割

■ 彩色图像分割

分割是一种将图像分成若干区域的处理方法。彩色图像分割描述了从图像中提取一个或多个连续的、满足均匀性（同质性）标准的连接区域的过程。这里的均匀性标准基于从图像光谱成分中提取的特征，这些特征是在给定的颜色空间中定义的。例如，分割可以基于几何特性和光学特性。

1、HSI彩色空间分割

HSI彩色空间分割，根据颜色对图像进行分割，在单独的平面上进行处理。以饱和度作为模板图像，将感兴趣的特征区域从色调图像中分离出来。由于强度不携带颜色信息，所以彩色图像分割一般不使用强度图像。

彩色图像分割

1. HSI彩色空间分割

```
clear;close all;clc;
f1=imread('baihe.jpg'); %读取图像
f=im2double(f1);
R=f(:,:,1); %R基色
G=f(:,:,2); %G基色
B=f(:,:,3); %B基色
hsi_f=rgb2hsi(f); %转换成HSI彩色空间
H=hsi_f(:,:,1); %色度分量
S=hsi_f(:,:,2); %饱和度分量
I=hsi_f(:,:,3); %亮度分量
S1=(S>0.3*(max(S(:))));
F=S1.*H;
```

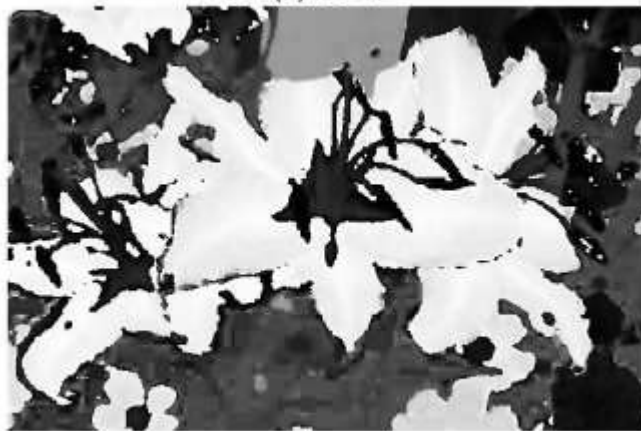
```
subplot(2,3,1),imshow(f1),title('(a)RGB原始分量','fontsize',12)
subplot(2,3,2),imshow(H),title('(b)H分量','fontsize',12)
subplot(2,3,3),imshow(S),title('(c)S分量','fontsize',12)
subplot(2,3,4),imshow(I),title('(d)I分量','fontsize',12)
subplot(2,3,5),imshow(S1),title('(e)二值饱和模板图像',
'fontsize',12)
subplot(2,3,6),imshow(F),title('(f)花朵分割结果','fontsize',12)
```


彩色图像分割

(a)RGB原始分量



(b)H分量



(c)S分量



(d)I分量



(e)二值饱和模板图像



(f)花朵分割结果



彩色图像分割

2.RGB彩色空间分割

在彩色图像分割方面，使用RGB色彩空间要优于HSI色彩空间在图像分割。假设RGB图像中某一感兴趣区域内彩色的平均用向量 \vec{a} 表示。分割的**目标**是判断该图像中每一点的彩色向量与 \vec{a} 的相似程度，如果该点的彩色分量与 \vec{a} 相似，那么该点属于分割结果；否则该点不在分割结果内。对于相似性的度量，最简单的度量之一是**欧氏距离**。让 z 代表RGB空间中的任意一点。如果 \vec{a} 和 z 之间的距离小于某个特定阈值 D_0 ，就可以说 \vec{a} 和 z 相似， \vec{a} 和 z 之间的欧氏距离由下式给出：

$$D(z, \vec{a}) = \|z - \vec{a}\| = [(z - \vec{a})^T (z - \vec{a})]^{\frac{1}{2}} = [(z_R - a_R)^2 + (z_G - a_G)^2 + (z_B - a_B)^2]^{\frac{1}{2}}$$

彩色图像分割

下标R、G、B表示向量 \vec{a} 和 z 的R、G、B分量。 $D(z, \vec{a}) \leq D_0$ 表示的点的轨迹为一个以 \vec{a} 的顶点为圆心，以 D_0 为半径的实心球体，此时球表面和内部的点与 \vec{a} 相似，球外的点与 \vec{a} 不相似，将相似和不相似的点分别赋以白、黑两种颜色，就可以得到一幅二值分割图像。

彩色图像分割

在RGB空间的彩色分割

```
clear,clc,close all;
rgb=imread('yellowlily.jpg'); %读取图像
rgb1= im2double(rgb); %转换成double类型
r=rgb1(:,:,1); %图像的红色分量
g=rgb1(:,:,2); %图像的蓝色分量
b=rgb1(:,:,3); %图像的蓝色分量
r1=r(129:256,86:170); %在花的红分量中选择一块矩形区域
r1_u=mean(mean(r1(:))); %计算该矩形区域的均值
[m,n]=size(r1); %得到该矩形区域的高度和宽度
sd1=0.0;
for i=1:m
    for j=1:n %两重循环对差值的平均进行累加
        sd1=sd1+(r1(i,j)-r1_u)*(r1(i,j)-r1_u);
    end
end
```

```
r1_d=sqrt(sd1/(m*n)); %计算该区域的标准偏差
r2=zeros(size(rgb1,1),size(rgb1,2));
ind=find((r>r1_u-1.25*r1_d)&(r<r1_u+1.25*r1_d));
%找到符合条件的点
r2(ind)=1; %将符合条件的点的灰度值赋为1
subplot(2,3,1),imshow(rgb),title('(a)原始图像')
subplot(2,3,2),imshow(r),title('(b)红色分量')
subplot(2,3,3),imshow(g),title('(c)蓝色分量')
subplot(2,3,4),imshow(b),title('(d)蓝色分量')
subplot(2,3,5),imshow(r2),title('(e)分割结果')
```


彩色图像分割

(a)原始图像



(b)红色分量



(c)绿色分量



(d)蓝色分量



(e)分割结果



THE END
Thank You.

