《数字图像处理》
第8讲 彩色图像处理

冯建江

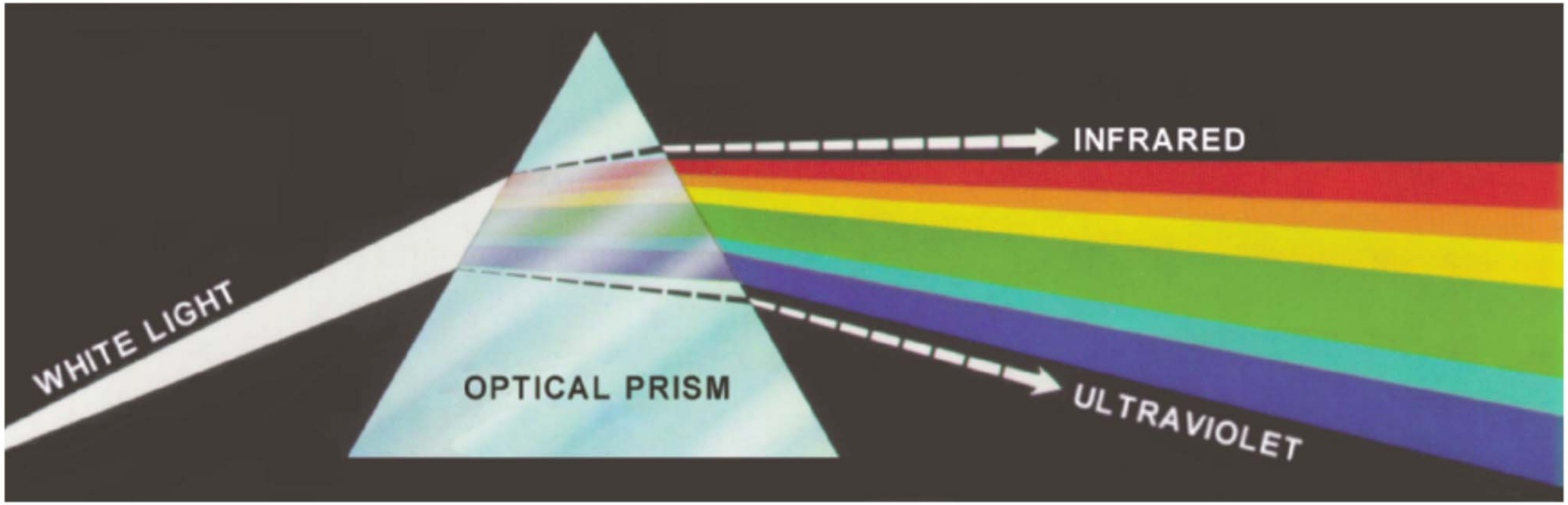
清华大学 自动化系

2017.11.16

内 容

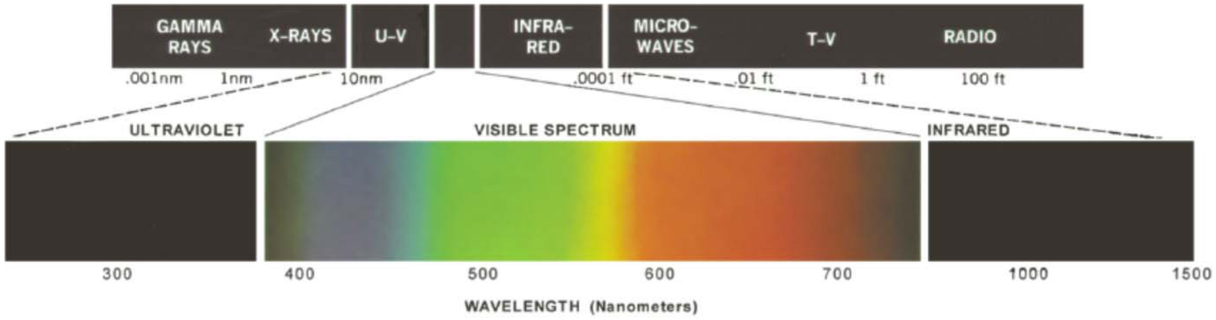
- 颜色模型
- 伪彩色图像处理
- 真彩色图像处理

光谱



- 1666年，牛顿发现一束太阳光通过三棱镜后，阳光被分解成了红、橙、黄、绿、青、蓝、紫七种颜色。
- 如果再放一个倒置的三棱镜，可以把七色光合成白光。
- 这一发现证明，普通的光是由各种颜色的光组成的。

电磁波谱

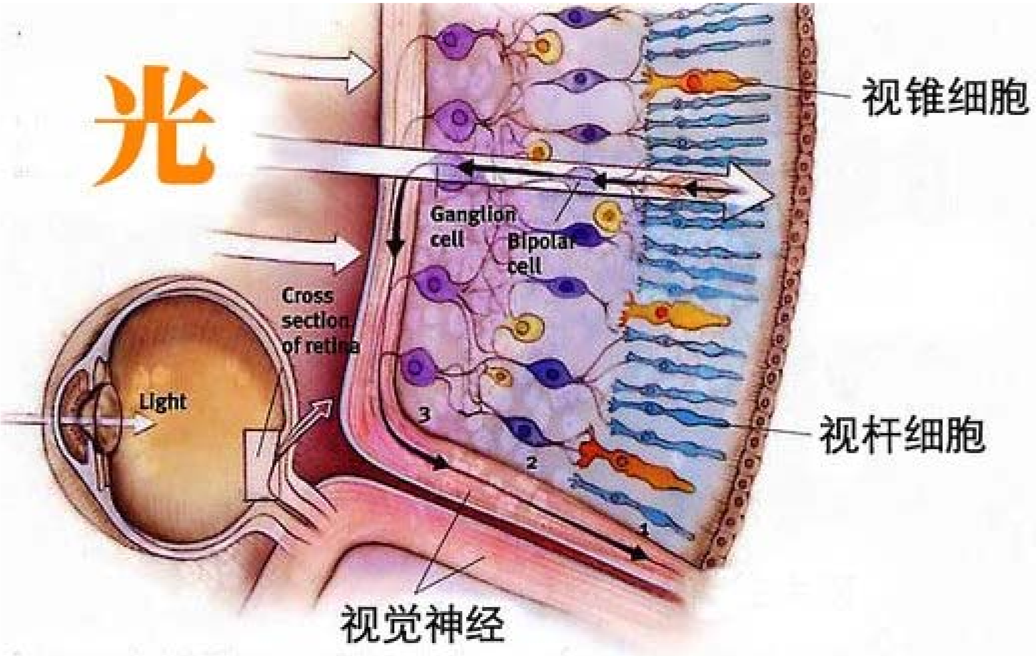


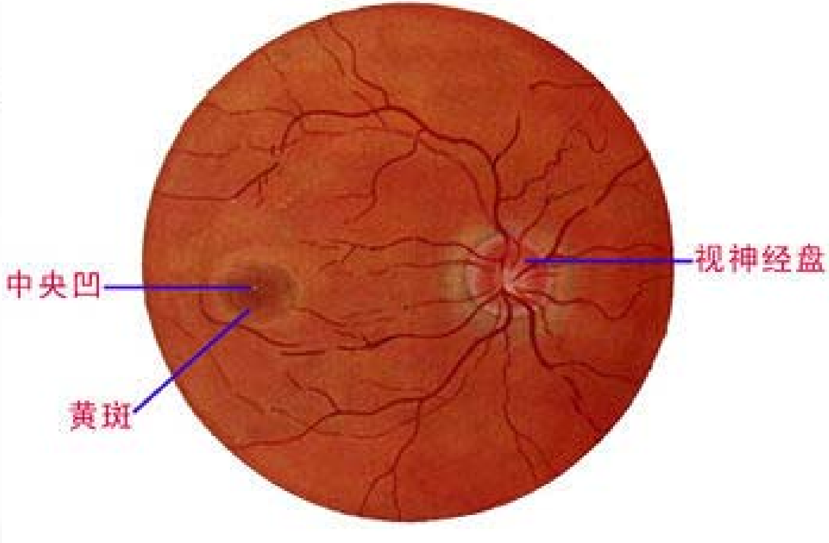
red	~ 625-740 nm	~ 480-405 THz
orange	~ 590-625 nm	~ 510-480 THz
yellow	~ 565-590 nm	~ 530-510 THz
green	~ 500-565 nm	~ 600-530 THz
cyan	~ 485-500 nm	~ 620-600 THz
blue	~ 440-485 nm	~ 680-620 THz
violet	~ 380-440 nm	~ 790-680 THz

所谓的颜色，只是不同频率的电磁波在人眼中的反应. 世上本没有颜色，是人的视觉神经对电磁波频率的感受.

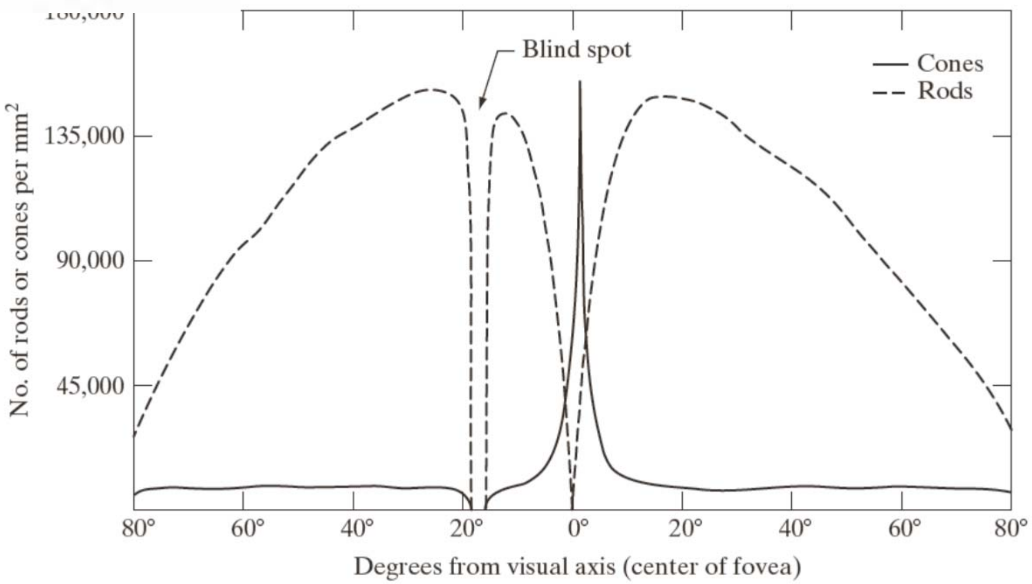
我们所看到物像，由色度和亮度组合而成. 色度反映其色彩构成，即电磁波的频率组合；亮度反应物像的明亮程度，即电磁波的辐射能量.

视锥细胞（Cones）和视杆细胞（Rods）



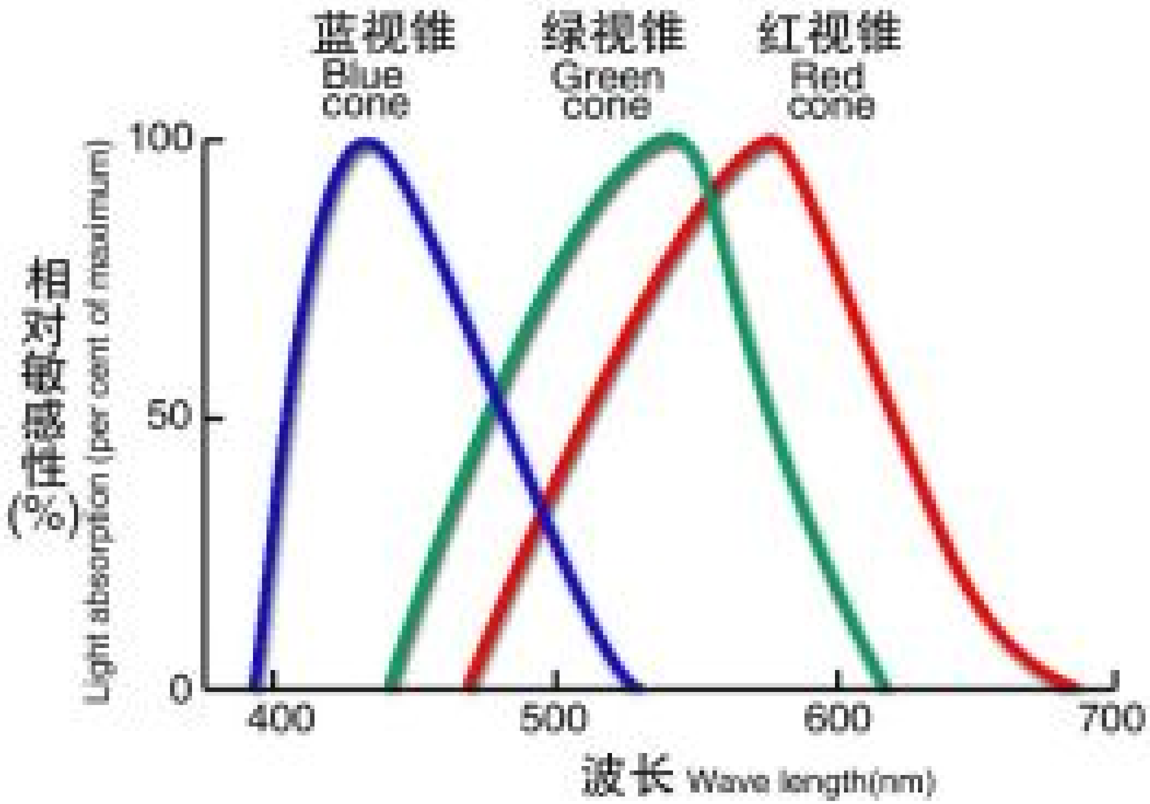


眼底镜所见（右侧）

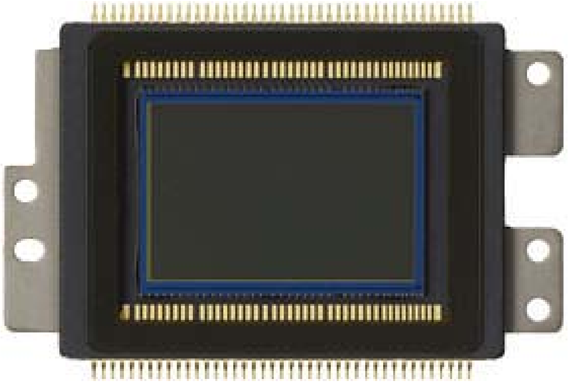


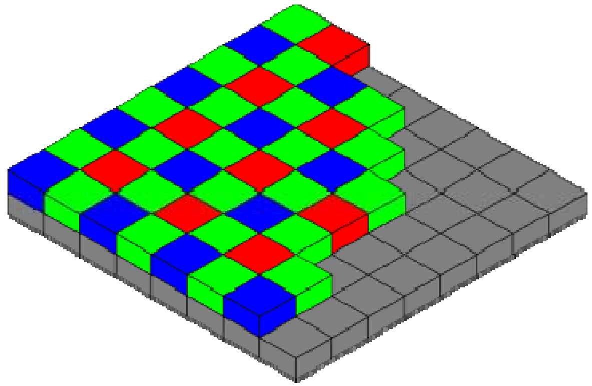
三种视锥细胞对光谱的敏感性

- 三种视锥细胞对入射光具有不同的频谱响应曲线，或者说它们对不同波长的辐射比较敏感。
- 三条曲线分布较宽，且有重叠



图像传感器（Image sensor）





- 图像传感器类似于人类的视网膜。
- 图像传感器中每个感光元件对应一个像素。
- 在接受光照后，感光元件产生对应的电流，电流大小与光强对应，电流是模拟的，经模数转换为数字。
- 由于感光元件只能感应光的强度，无法捕获色彩信息，因此必须在感光元件上方覆盖彩色滤光片。
- 常用做法是覆盖红绿蓝三色滤光片，以1:2:1的比例构成（因人眼对绿色敏感）。

https://en.wikipedia.org/wiki/Bayer_filter
http://www.canon.com.cn/specialsite/ds_abcbook/beginner01.html

其他动物的颜色感知能力

斗牛士为什么用红色斗篷？

其他动物的颜色感知能力

- 灵长类有3种视锥细胞：红、绿、蓝。
- 其他哺乳动物有2种：黄和蓝。
- 鱼类、两栖、爬行、鸟类有4种：红、绿、蓝、紫外线。
- 无脊椎动物比较复杂，有的有12个颜色感受器，如虾蛄；有的只有1种感受器，只能看到黄色。
- 有些夜行动物没有视锥细胞，为全色盲。

防鸟撞的新型玻璃

在美国，建筑物的玻璃每年会造成5. 5亿只鸟类死亡，玻璃外墙建筑是造成鸟类死亡的首要威胁

直接添加图案影响美观

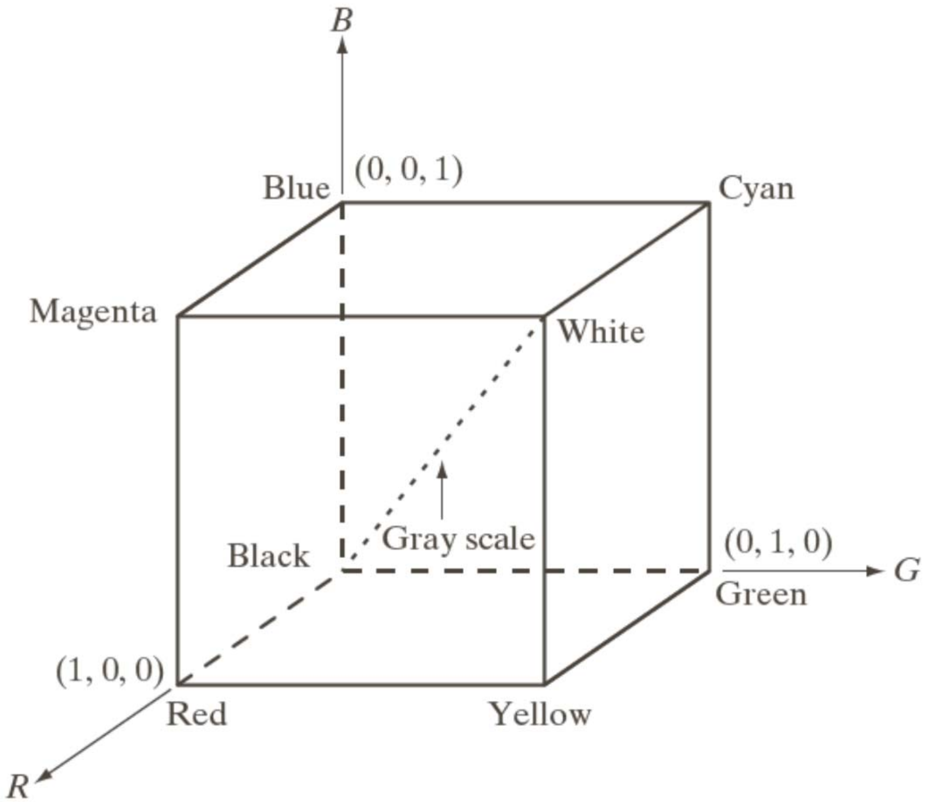
受蜘蛛网启发，研发新型玻璃，紫外传感器能看到网格，对人眼为透明

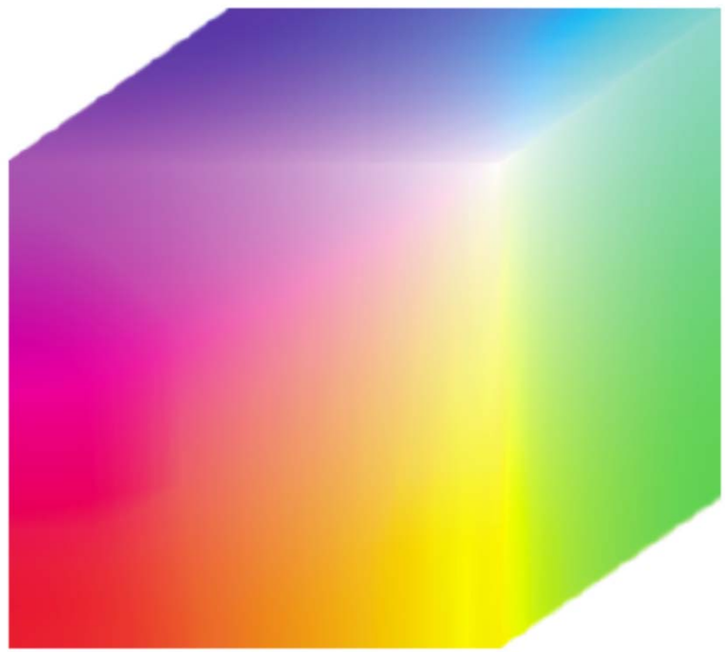
初步实验表明，66%的鸟类能避免撞新型玻璃

http://news.ifeng.com/a/20140729/41355150_0.shtml

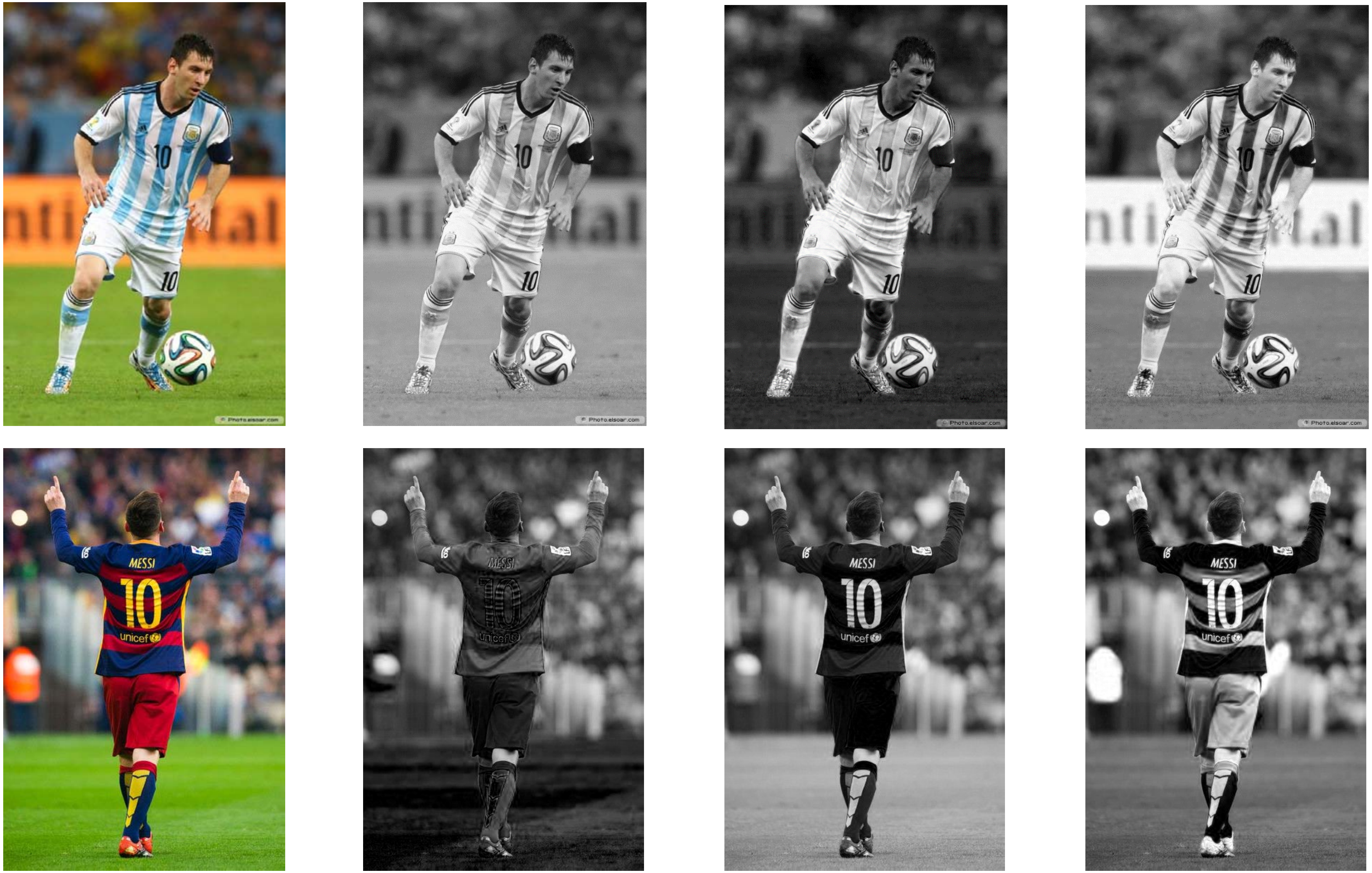
RGB颜色模型

- RGB颜色模型是图像采集、显示设备（CRT、LED显示器，投影仪等）采用的模型
- 通过对红(R)、绿(G)、蓝(B)三个颜色通道的变化以及它们相互之间的叠加来得到各种颜色





R G B ?

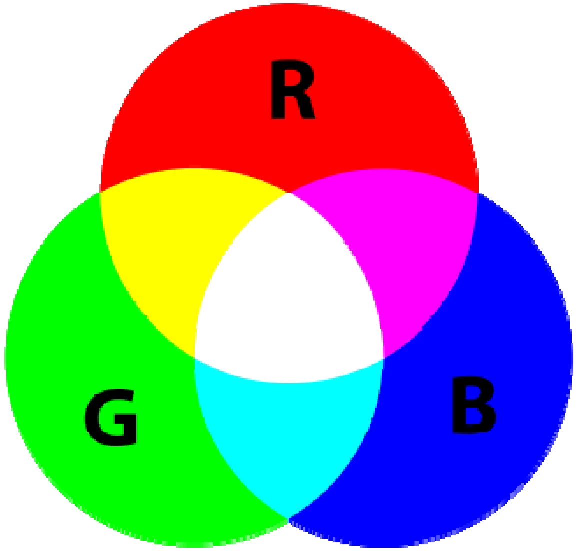


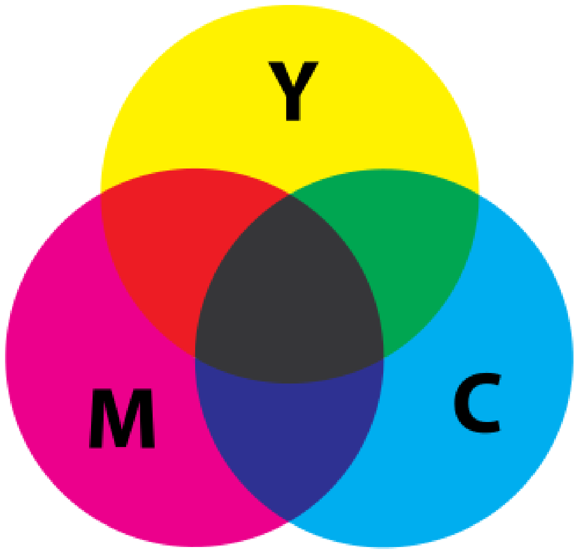
R G B

```matlab
function showRGB()
I = imread('..\data\player2.jpg');
figure(1),clf
ax(1)=subplot(2,4,1);imshow(I);
ax(2)=subplot(2,4,2);imshow(I(:,:,1));
ax(3)=subplot(2,4,3);imshow(I(:,:,2));
ax(4)=subplot(2,4,4);imshow(I(:,:,3));
linkaxes(ax);

I = imread('..\data\player3.jpg');
ax2(1)=subplot(2,4,5);imshow(I);
ax2(2)=subplot(2,4,6);imshow(I(:,:,1));
ax2(3)=subplot(2,4,7);imshow(I(:,:,2));
ax2(4)=subplot(2,4,8);imshow(I(:,:,3));
linkaxes(ax2);
```

CMY颜色模型

- CMY（CMYK）是印刷中采用的色彩模式
- 它和RGB的不同：
  - ➢RGB模式是一种发光的色彩模式。例如，在一间黑暗的房间内，我们仍然可以看见电脑屏幕上的内容；
  - ➢CMYK是一种依靠反光的色彩模式。例如，阳光或灯光照射到报纸上，再反射到我们的眼中，才看到报纸的内容。
  - ➢RGB叫加法三原色，CMY叫减法三原色
  - ➢显示中，R+G+B为白色
  - ➢印刷中，C+M+Y为黑色
- 在印刷中，出于成本和质量考虑，黑色是专门的颜料
- CMY与RGB模型的转换：$CMY = 1 - RGB$





猜RGB值

3

115,222,67

1

201,58,148

2

189,52,80

4

38,41,138

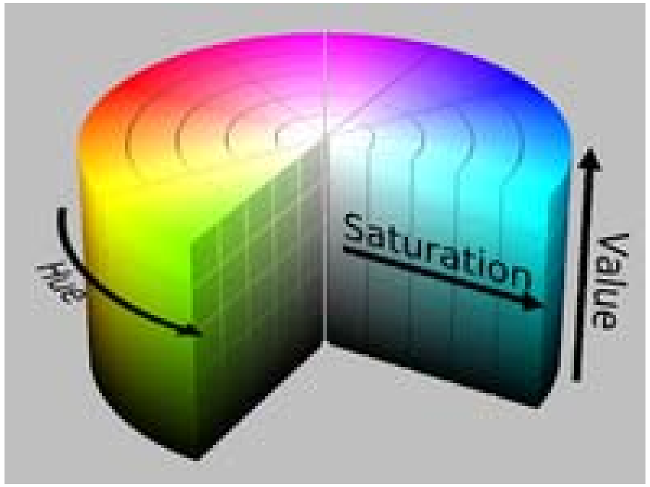
说出颜色名字

HSI颜色模型

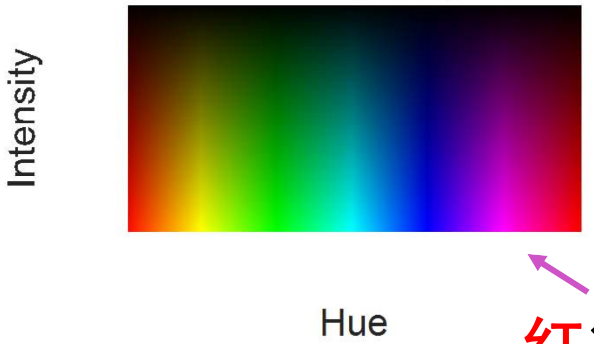
- RGB、CMY是面向设备（采集、显示、印刷）的颜色模型，与人的视觉感知不同，对人的使用不方便
- 例如，随便指定一个图像像素，人很难说出其RGB值
- HSI模型是美国色彩学家孟塞尔（Munseu）于1915年提出的，它反映了人的视觉系统感知颜色的方式，以色调(Hue)、饱和度(Saturation)和亮度(Intensity)三种基本特征量来感知颜色。同MATLAB的HSV模型。
- I分量与图像的彩色信息无关；H和S分量与人感受彩色的方式是紧密相联的，统称为色度
- HSI也是人类用语言表达颜色的方式（亮红、暗红、鲜红、淡红）

HSI颜色模型

- HSI圆柱坐标系中，0度为红色，随着度数增加，颜色为橙黄绿青蓝紫，最后又回到红色
- 平面图显示方式适合调色板应用，分别用鼠标选择



饱和度为1



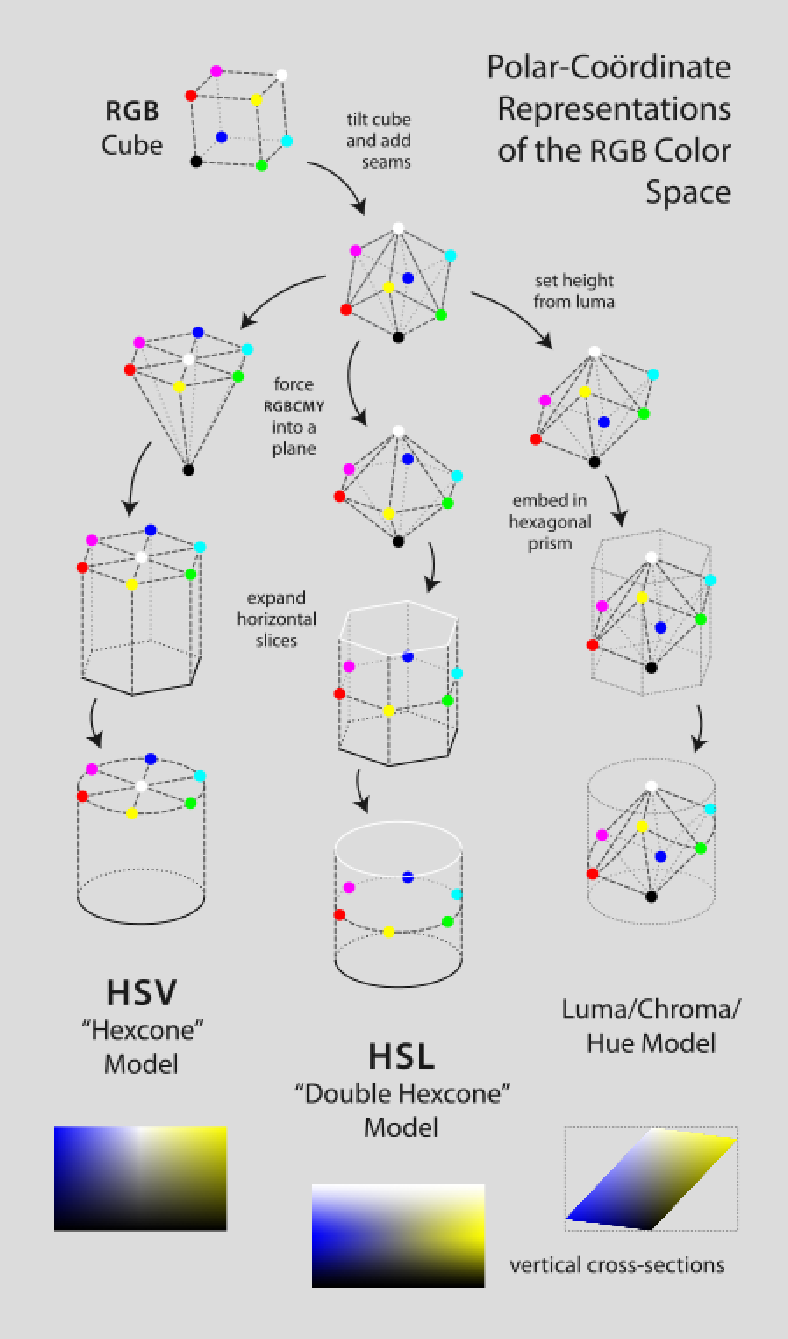
红得发紫

亮度为1

色调为0

1、圆柱最外面的圆筒展开
2、最上面一层像扇子一样展开
3、沿着圆柱中轴的切面

```matlab
function CheckHSV()
% 2016-11-09

M = 200; N = 400;
[DX, DY] = meshgrid(1:N,1:M);
hsv = zeros(M, N, 3);

% S = 1
hsv(:,:,1) = DX/N;
hsv(:,:,2) = ones(M,N);
hsv(:,:,3) = DY/M;
I1 = hsv2rgb(hsv);

% V = 1
hsv(:,:,1) = DX/N;
hsv(:,:,2) = DY/M;
hsv(:,:,3) = ones(M,N);
I2 = hsv2rgb(hsv);

% H = 0
hsv(:,:,1) = 0*ones(M,N);
hsv(:,:,2) = DY/M;
hsv(:,:,3) = DX/N;
I3 = hsv2rgb(hsv);

figure(1), clf
subplot(1,3,1),imshow(I1),xlabel('Hue'),ylabel('Intensity');
subplot(1,3,2),imshow(I2),xlabel('Hue'),ylabel('Saturation');
subplot(1,3,3),imshow(I3),xlabel('Intensity'),ylabel('Saturation');
```



RGB to HSI

$$H = \begin{cases} \theta & \text{if } (B \le G) \\ 360 - \theta & \text{if } (B > G) \end{cases}$$

$$\theta = \cos^{-1}\left[\frac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}} \right]$$

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

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

HSI to RGB

$0 \le H \le 120^o$

$120^o \le H \le 240^o$

$240^o \le H < 360^o$

$$R = I(1 + \frac{S \cos H}{\cos(60^o - H)})$$

$$R = I(1 - S)$$

$$R = 3I - (G + B)$$

$$G = 3I - (R + B)$$

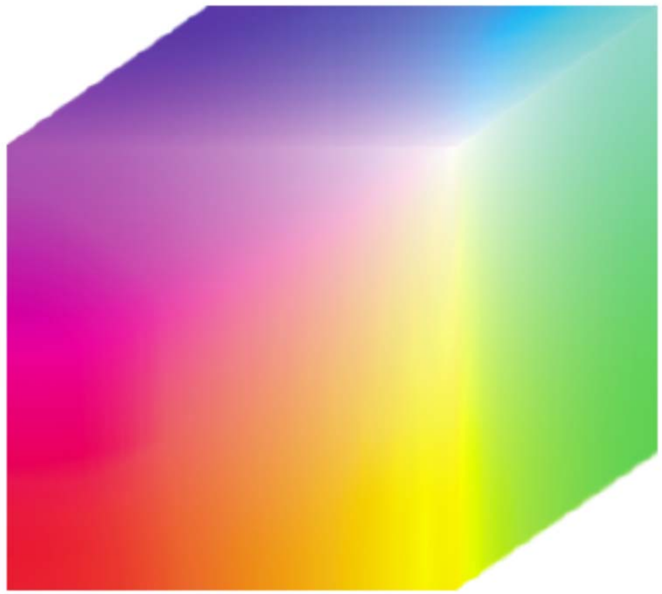
$$G = I(1 + \frac{S \cos H}{\cos(60^o - H)})$$

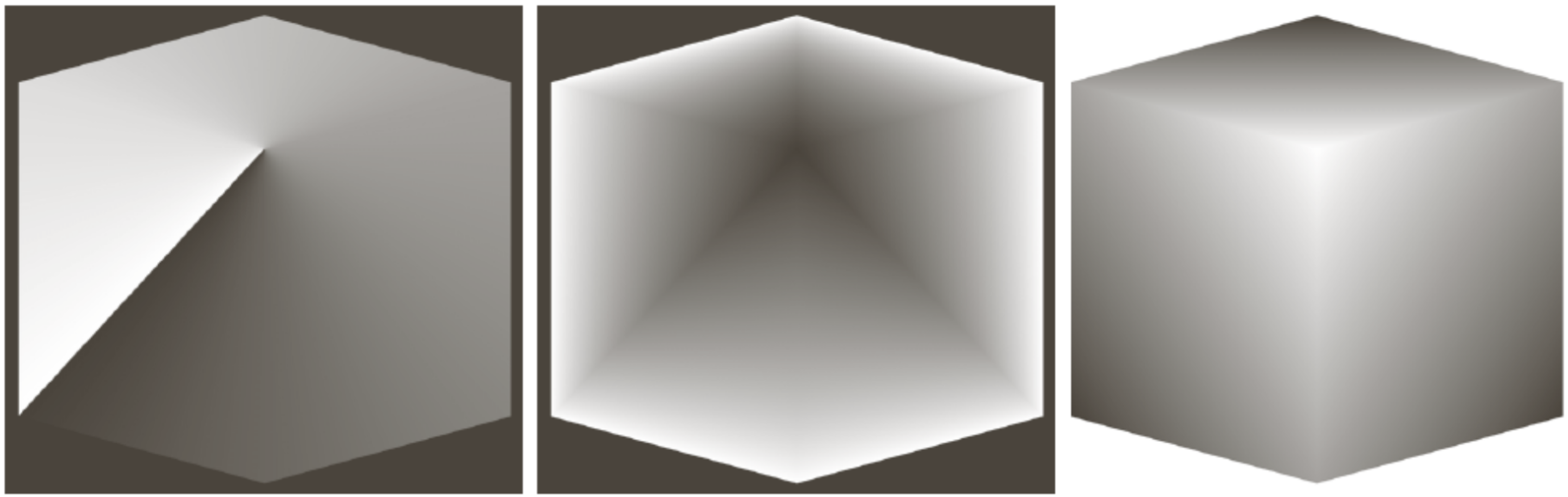
$$G = I(1 - S)$$

$$B = I(1 - S)$$

$$B = 3I - (R + G)$$

$$B = I(1 + \frac{S \cos H}{\cos(60^o - H)})$$





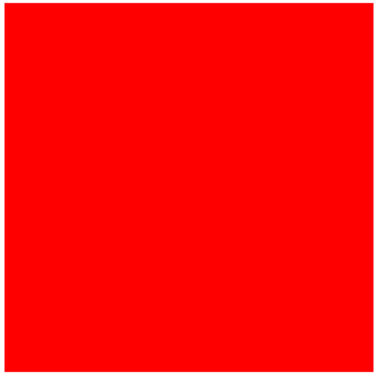
a b c

FIGURE 6.15 HSI components of the image in Fig. 6.8. (a) Hue, (b) saturation, and (c) intensity images.

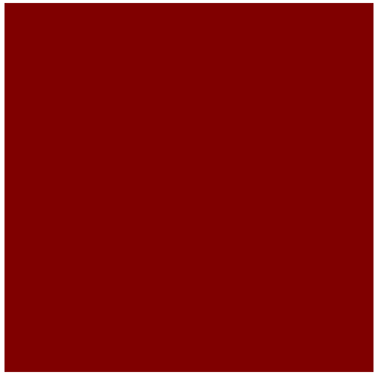
给定颜色文字描述，合成该颜色图像

```
%colors
h = 100; w = 100;
figure(1),clf

hsv_image = ones(h, w, 3);
rgb_image = hsv2rgb(hsv_image);
subplot(1,3,1)
imshow(rgb_image);

hsv_image = ones(h, w, 3);
hsv_image(:,:,3) = 0.5*ones(h, w);
rgb_image = hsv2rgb(hsv_image);
subplot(1,3,2)
imshow(rgb_image);

hsv_image = ones(h, w, 3);
hsv_image(:,:,2) = 0.5*ones(h, w);
rgb_image = hsv2rgb(hsv_image);
subplot(1,3,3)
imshow(rgb_image);
```

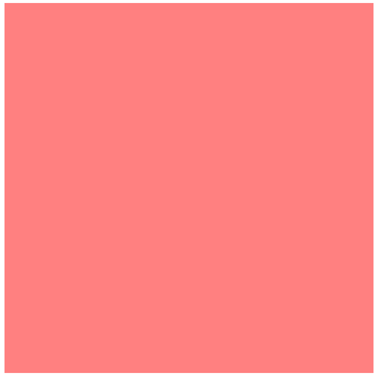
鲜红



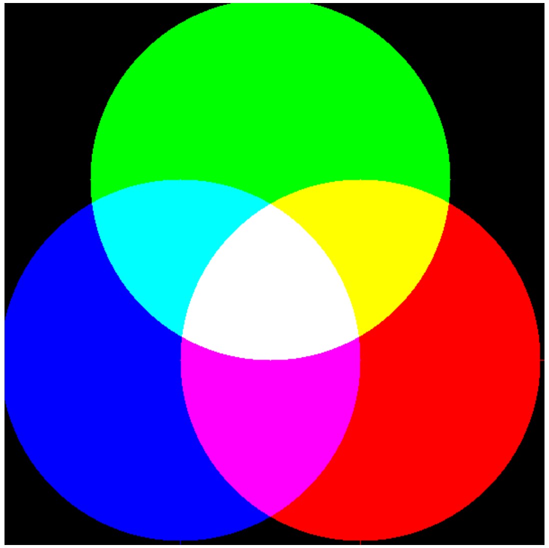
暗红

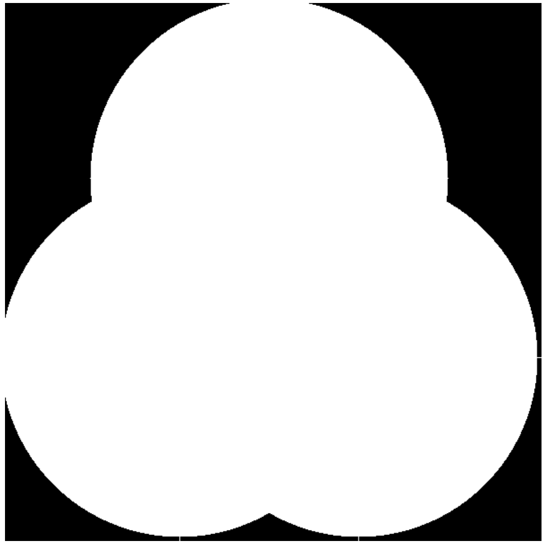


淡红

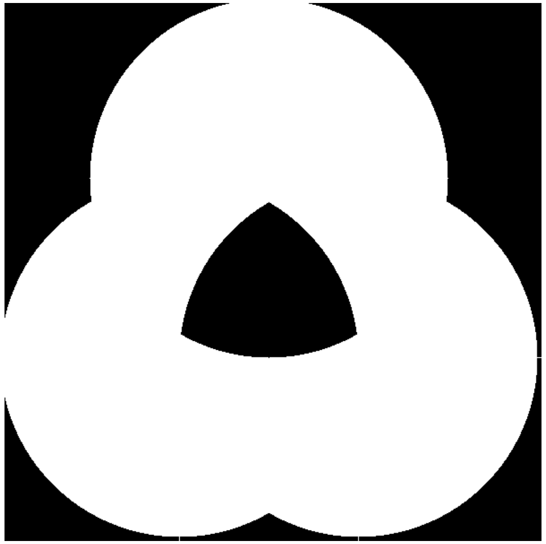


编程合成淡绿、深蓝

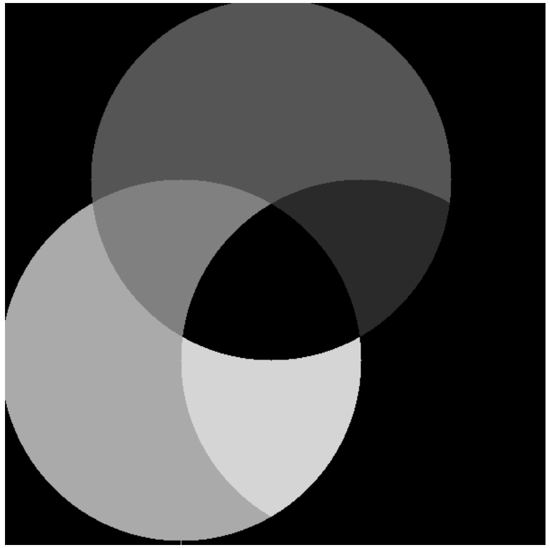




I



S

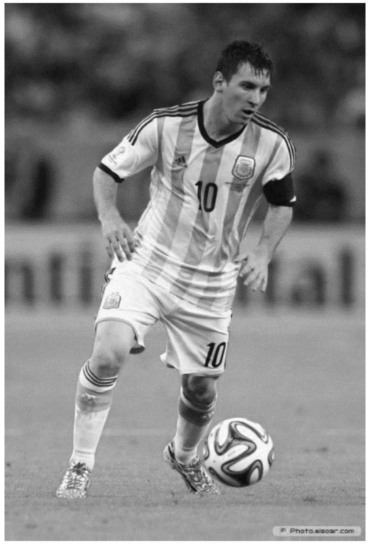


H

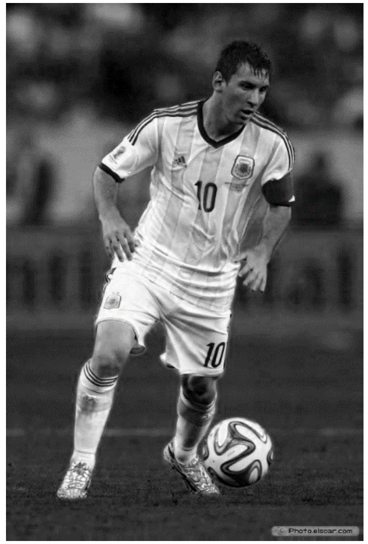
```matlab
% rgbcircles
% 2016-11-10

r = 200;
I = zeros(3*r,3*r,3);
[DX,DY] = meshgrid(1:3*r,1:3*r);
R = sqrt((DX-2*r).^2+(DY-2*r).^2) <= r;
I1 = I;
I1(:,:,1) = R;

G = sqrt((DX-1.5*r).^2+(DY-r).^2) <= r;
I2 = I;
I2(:,:,2) = G;

B = sqrt((DX-r).^2+(DY-2*r).^2) <= r;
I3 = I;
I3(:,:,3) = B;
I = I1+I2+I3;

close all
figure(1),
imshow(I);

hsv = rgb2hsv(I);
figure(2),
subplot(2,2,1),imshow(I)
subplot(2,2,2),imshow(hsv(:,:,1))
subplot(2,2,3),imshow(hsv(:,:,2))
subplot(2,2,4),imshow(hsv(:,:,3))
```

比较RGB和HIS分量图像

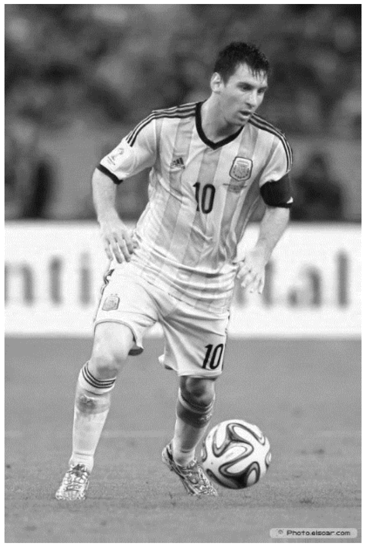
R



G



B



I

H

S

```matlab
function showRGBandHSV()
% 2016-11-10
I = imread('..\data\player2.jpg');
hsv = rgb2hsv(I);
figure(1),clf
ax(1)=subplot(2,4,1);imshow(I);
ax(2)=subplot(2,4,2);imshow(I(:,:,1));
ax(3)=subplot(2,4,3);imshow(I(:,:,2));
ax(4)=subplot(2,4,4);imshow(I(:,:,3));
ax(5)=subplot(2,4,6);imshow(hsv(:,:,1));
ax(6)=subplot(2,4,7);imshow(hsv(:,:,2));
ax(7)=subplot(2,4,8);imshow(hsv(:,:,3));
linkaxes(ax);
```

内 容

- 颜色模型
- 伪彩色图像处理
- 真彩色图像处理

伪彩色图像处理

- 伪彩色处理是指将灰度图像转换成彩色图象。
- 因为人眼对于彩色的分辨能力远高于对灰度图像的分辨能力，所以将灰度图像转换成彩色可以提高人眼对图像细节的辨别能力。
- 伪彩色并不能真实的反映图像的彩色情况。

亮度切割（intensity slicing）

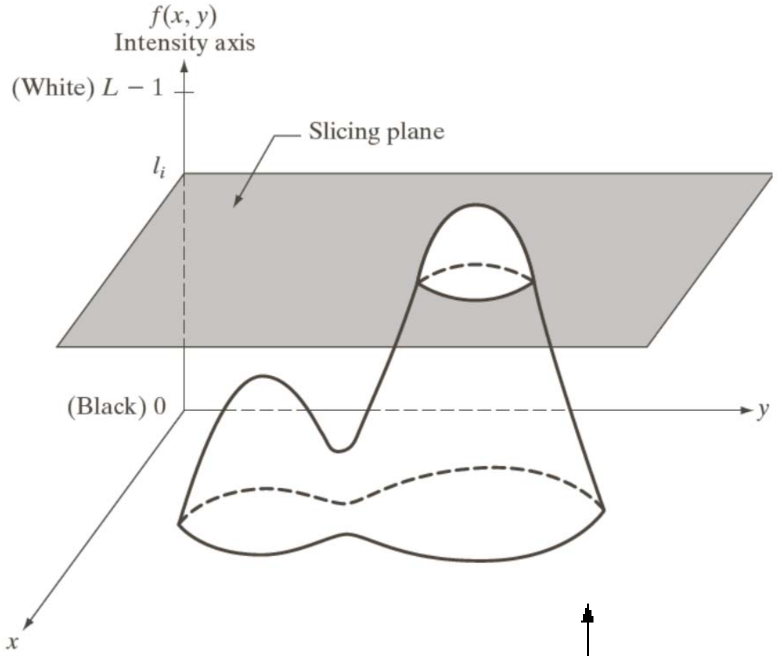


FIGURE 6.18 Geometric interpretation of the intensity-slicing technique.

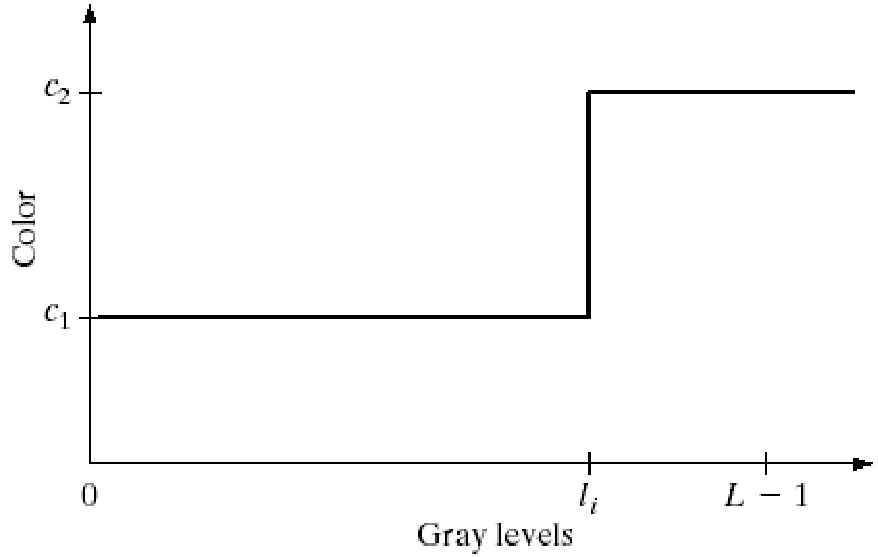
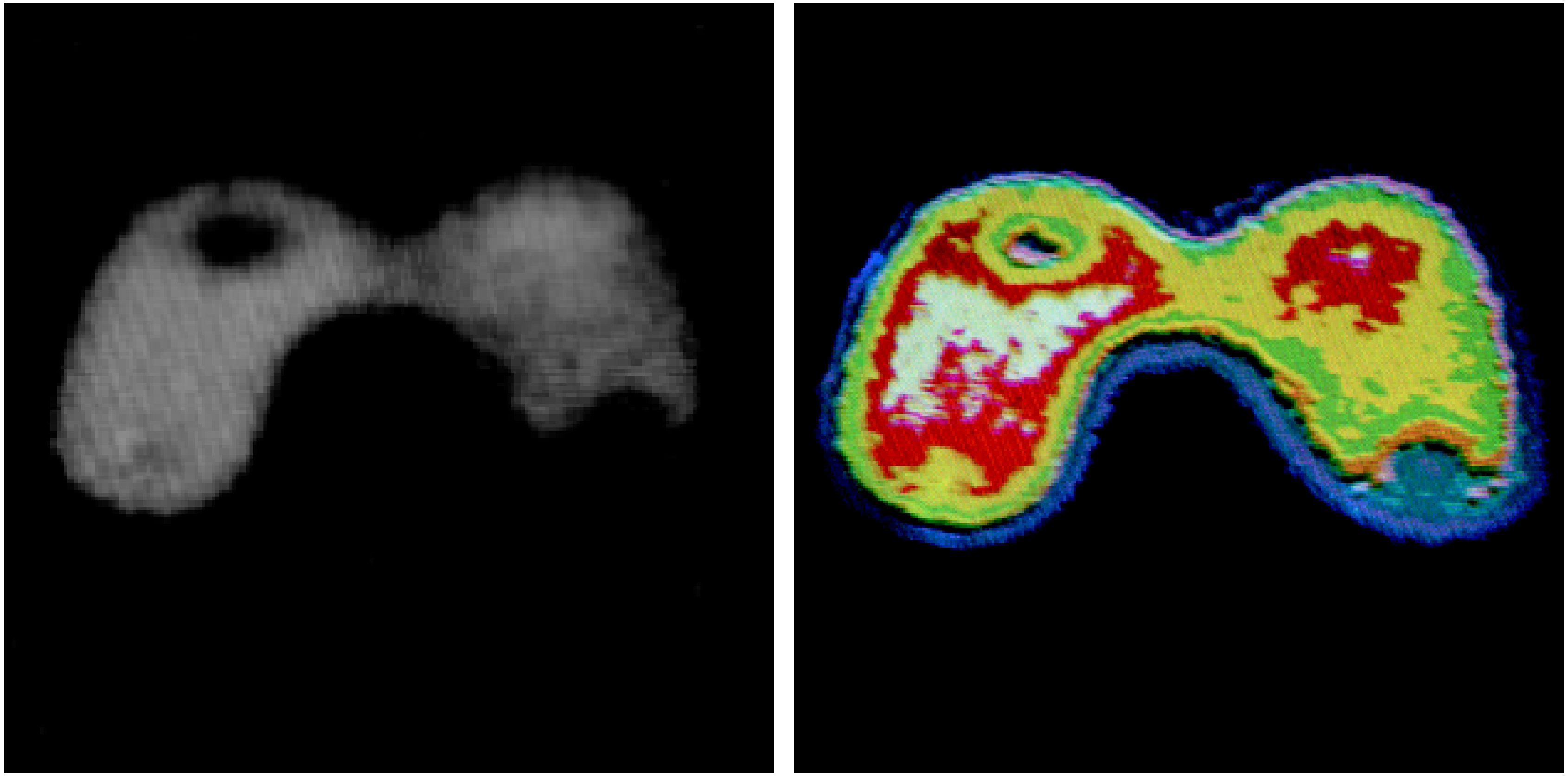


FIGURE 6.19 An alternative representation of the intensity-slicing technique.

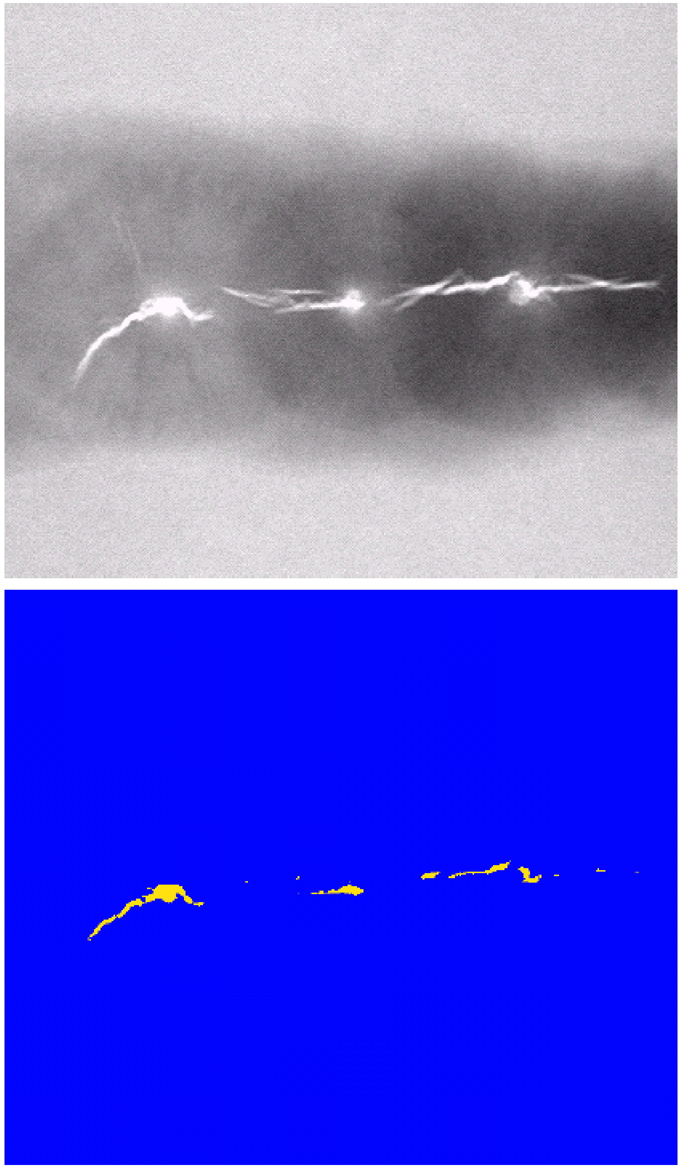
亮度切割（多个颜色）



a b

FIGURE 6.20 (a) Monochrome image of the Picker Thyroid Phantom. (b) Result of density slicing into eight colors. (Courtesy of Dr. J. L. Blankenship, Instrumentation and Controls Division, Oak Ridge National Laboratory.)

亮度切割（有明确物理意义）

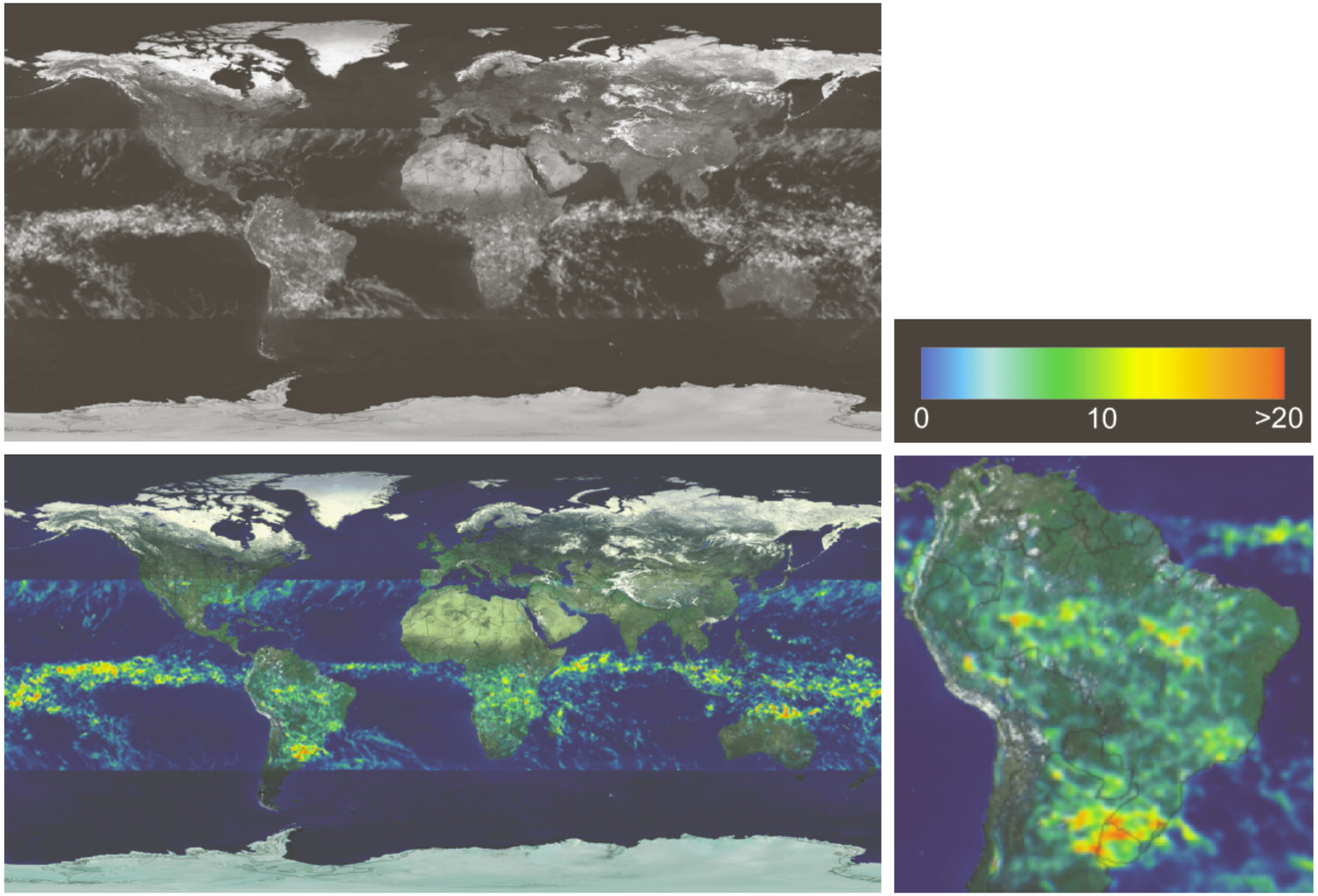


a
b

FIGURE 6.21 (a) Monochrome X-ray image of a weld. (b) Result of color coding. (Original image courtesy of X-TEK Systems, Ltd.)

用x光发现焊锡的空洞、裂缝

亮度切割（更多颜色）



a b
c d

FIGURE 6.22 (a) Gray-scale image in which intensity (in the lighter horizontal band shown) corresponds to average monthly rainfall. (b) Colors assigned to intensity values. (c) Color-coded image. (d) Zoom of the South American region. (Courtesy of NASA.)

论文中尽量用伪彩色

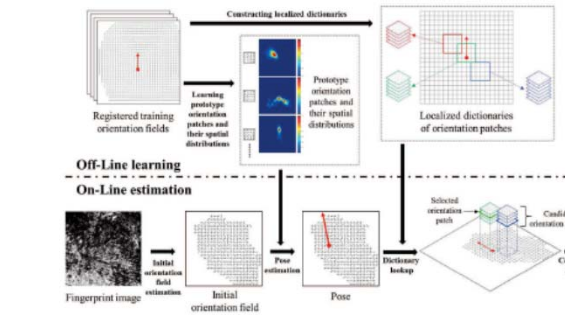


Fig. 3. Flowchart of the proposed fingerprint pose estimation and orientation field estimation approach. The ap
independent parts. The first part is the off-line learning algorithm, while the second part is the on-line estimati

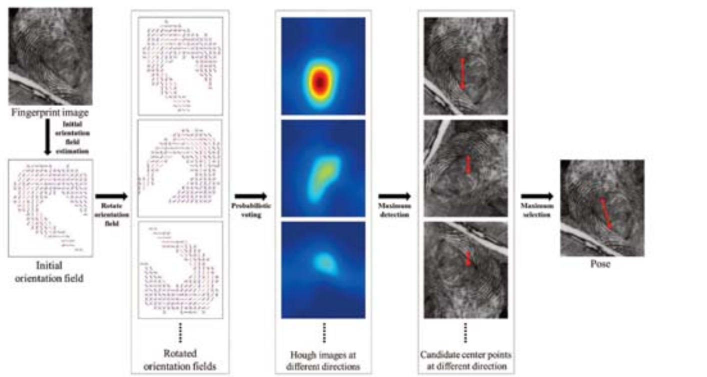
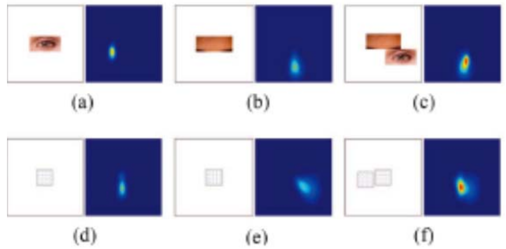


Fig. 11. Pose estimation of a fingerprint with unknown direction. The initial orientation field is rotated by a set of possible angles; the center estimation procedure is then applied to detect the center at each angle; and finally the one with the largest value is chosen.



g. 6. Predicting the center of a face based on a face part is analogous
h. In
bility
that
. (b)
. (d)
ation
of

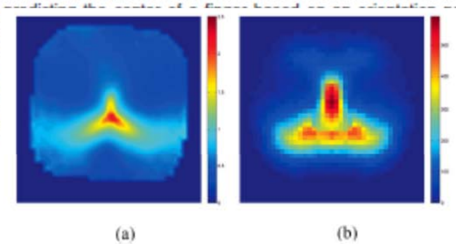


Fig. 13. Correlation between the variation of orientation and the size (the number of orientation patches) of localized dictionaries. (a) Image of circular standard deviation at each location of registered training orientation fields. (b) Image of size of localized dictionaries. Red color

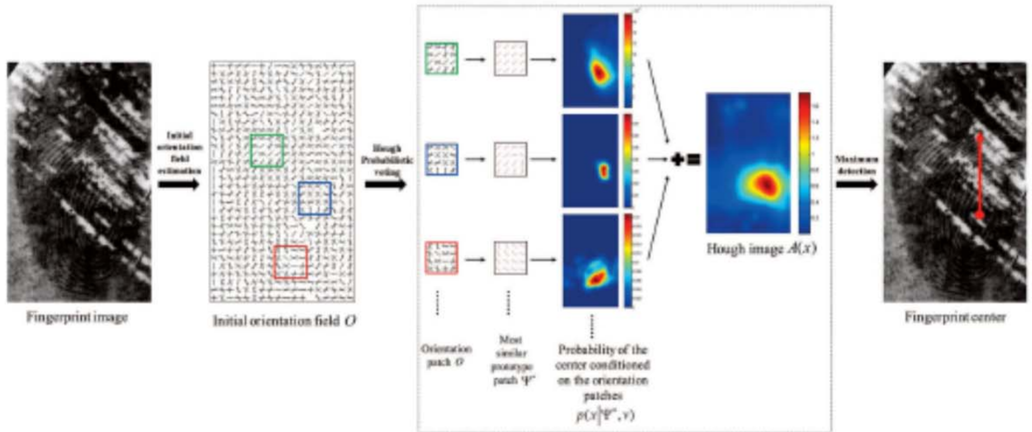


Fig. 10. Flowchart of estimating the finger center of an upright latent fingerprint.

orientation fields

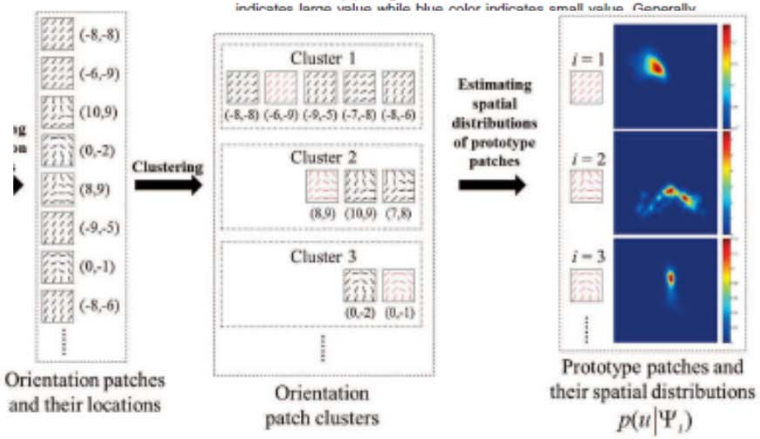


Fig. 8. Flowchart of learning a set of prototype orientation patches and their spatial distributions.

Xiao Yang, Jianjiang Feng, Jie Zhou: Localized Dictionaries Based Orientation Field Estimation for Latent Fingerprints. IEEE T-PAMI 2014

伪彩色变换

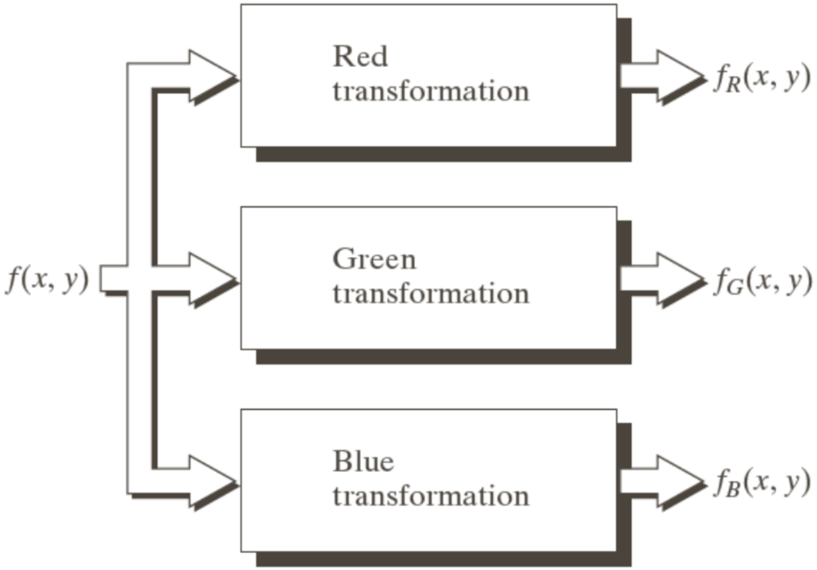
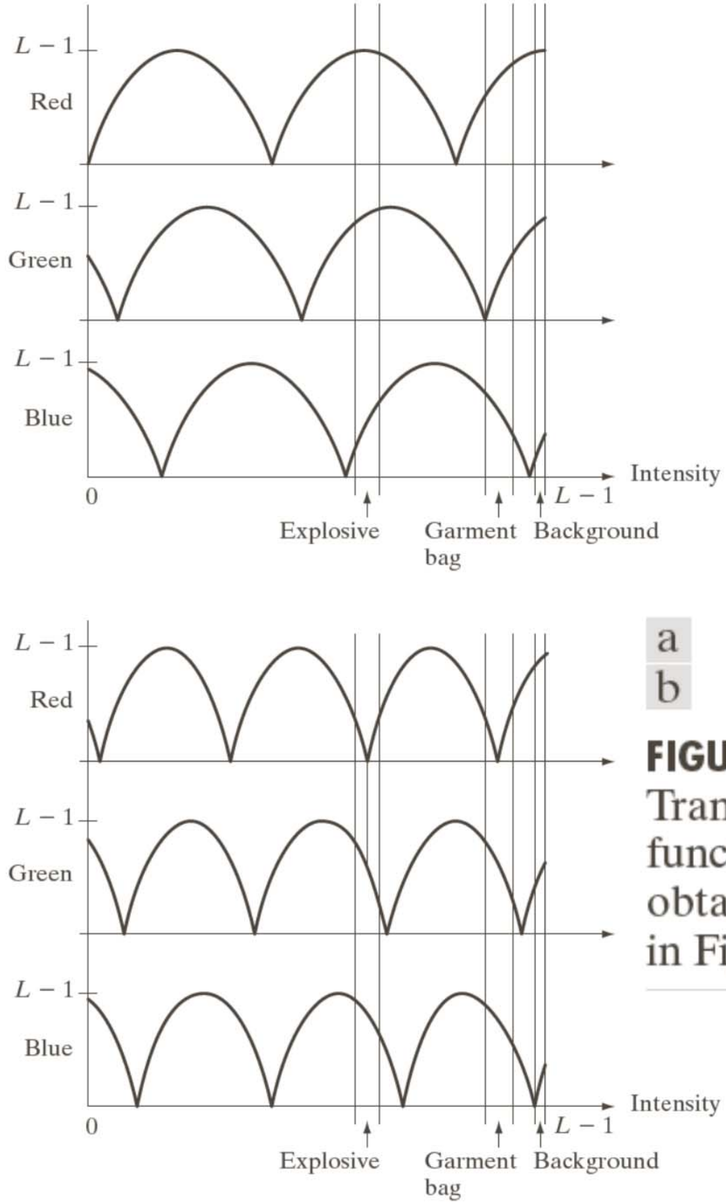


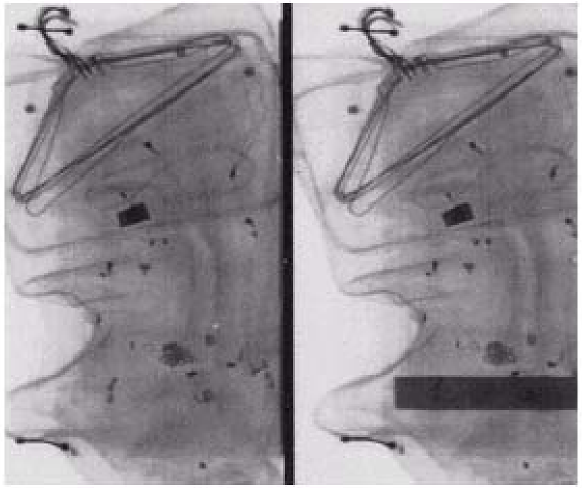
FIGURE 6.23 Functional block diagram for pseudocolor image processing. f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

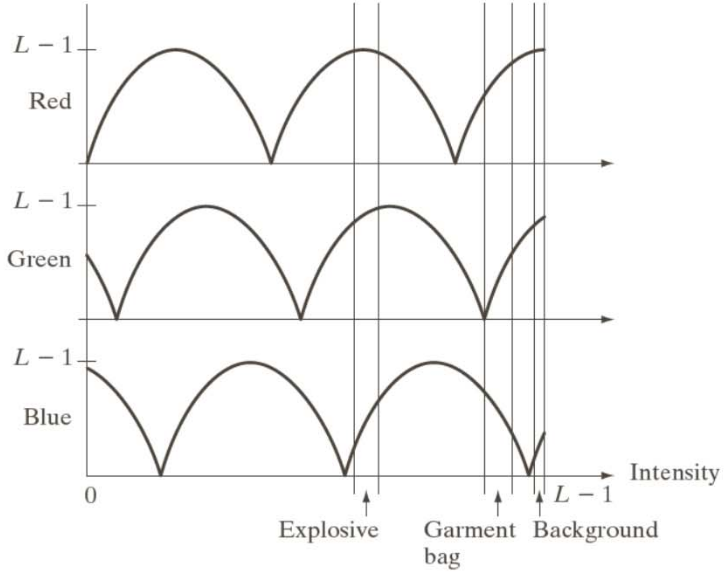


a
b

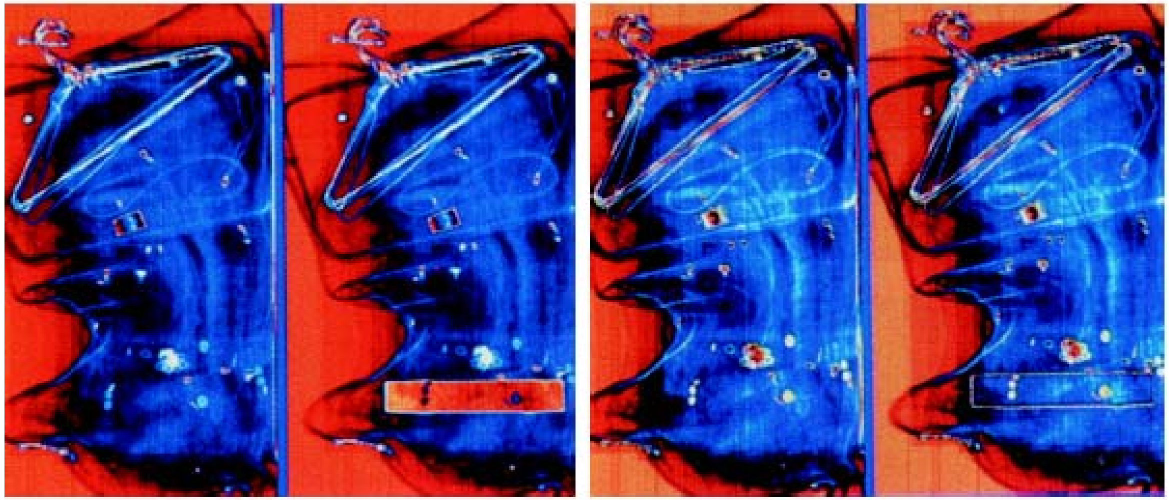
FIGURE 6.25 Transformation functions used to obtain the images in Fig. 6.24.

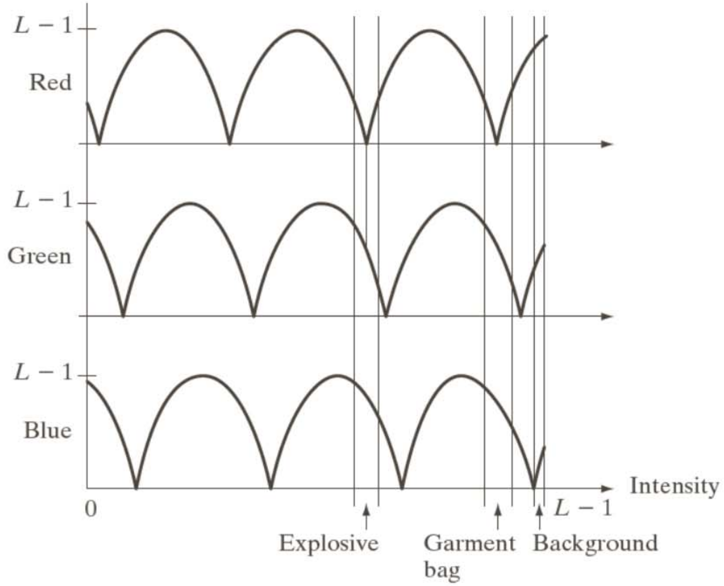
伪彩色变换





L − 1
Red
L − 1
Green
L − 1
Blue
Intensity
0
L − 1
Explosive
Garment bag
Background





L − 1
Red
L − 1
Green
L − 1
Blue
Intensity
0
L − 1
Explosive
Garment bag
Background

内 容

- 颜色模型
- 伪彩色图像处理
- 真彩色图像处理

真彩图像处理

- 输入和输出图像均为彩色的
- 例如，24bit的RGB图像，一个像素的三个分量值均占8bit
- 彩色图像可视为三幅标量图像，也可以视为矢量图像
- 2种处理策略：
  - 对每幅分量图像单独处理，最后合并起来
  - 将每个像素视为矢量，按照矢量定义进行处理
- 2种策略的结果是否相同，取决于具体的运算
- 对于均值滤波、锐化等线性滤波，结果相同

单分量变换增强

- 在HSI颜色空间中，人眼对于不同分量的感受是比较独立的
- 基于HSI空间，可以只对单分量进行处理，得到许多种想要的效果
- 算法分三步：
  1. 将RGB分量图转化为HSI分量图
  2. 利用灰度图像增强方法对某分量图做增强
  3. 将HSI分量图转化为RGB分量图
- 下面分别介绍三个分量的处理

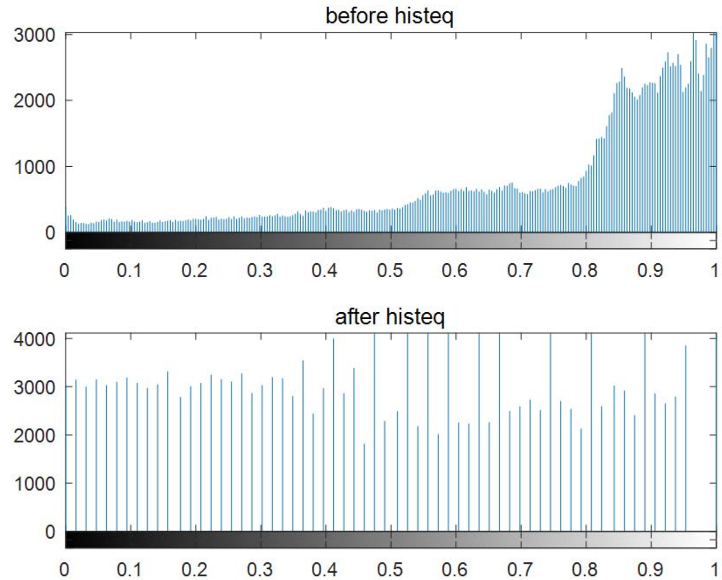
亮度增强：伽马校正

$$s = r^{\gamma}, \quad r \in [0,1], \quad \gamma > 0$$

r输入像素值，s输出像素值

```matlab
function EnhanceIntensity()
% 2016-11-09

I = imread('..\data\exposure1.jpg');
gamma = 3;

% I = imread('..\data\exposure2.jpg');
% gamma = 0.5;

I = im2double(I);
hsv = rgb2hsv(I);
hsv(:,:,3) = hsv(:,:,3).^gamma;
J = hsv2rgb(hsv);

figure(1), clf
subplot(1,2,1),imshow(I);
subplot(1,2,2),imshow(J);
```

亮度增强：直方图均衡


before histeq
3000
2000
1000
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
after histeq
4000
3000
2000
1000
0
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

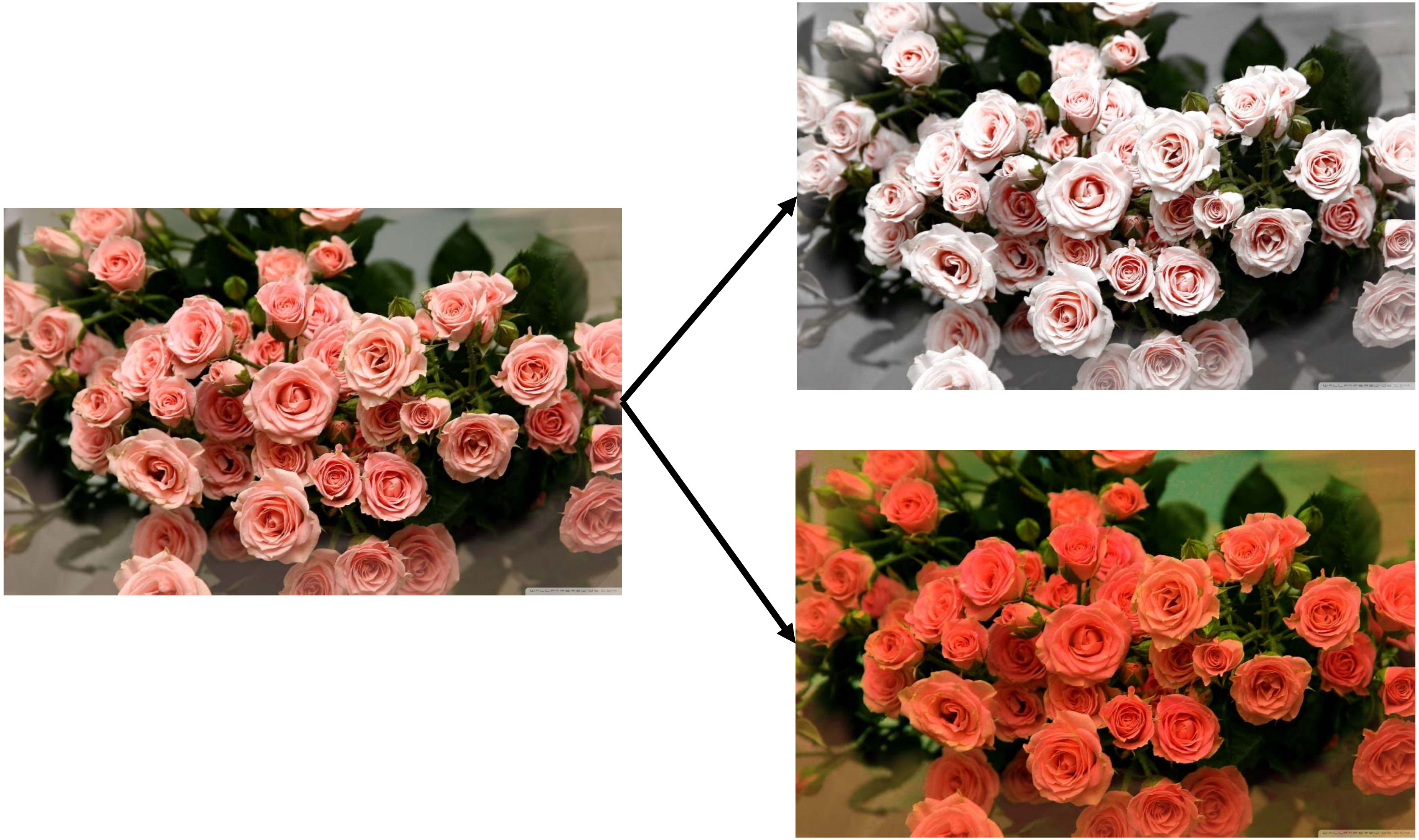

RGB


HSI

```matlab
function ColorHistEq()
% ColorHistEq
% 2016-11-05

I = imread('..\data\color_histeq2.jpg');
I = im2double(I);

J1 = I;
for k = 1:3
    J1(:,:,k) = histeq(I(:,:,k));
end

I_hsv = rgb2hsv(I);
J2_hsv = I_hsv;
J2_hsv(:,:,3) = histeq(I_hsv(:,:,3));
J2 = hsv2rgb(J2_hsv);

close all
figure(1),imshow(I)
figure(2),imshow(J1)
figure(3),imshow(J2)
figure(4),subplot(2,1,1),imhist(I_hsv(:,:,3)),title('before histeq')
figure(4),subplot(2,1,2),imhist(J2_hsv(:,:,3)),title('after histeq')

figure(5),clf
for k = 1:3
    ax(1)=subplot(3,3,k);imhist(I(:,:,k));
    ax(2)=subplot(3,3,k+3);imhist(J1(:,:,k));
    ax(3)=subplot(3,3,k+6);imhist(J2(:,:,k));
end
```

饱和度增强

- 减小饱和度，图像颜色变淡，原来比较淡的区域已经变为灰色
- 增加饱和度，图像颜色变艳



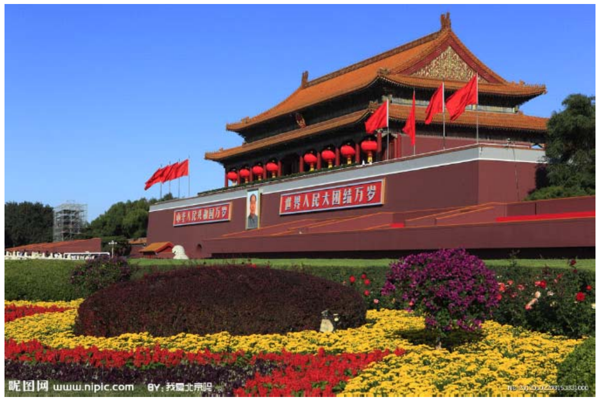
```matlab
function EnhanceSaturation()
% 2016-11-09

I = imread('..\data\saturation2.jpeg');
% I = imread('..\data\saturation3.jpg');
I = im2double(I);
hsv = rgb2hsv(I);

hsv2 = hsv;
gamma = 3;
hsv2(:,:,2) = hsv(:,:,2).^gamma;
J1 = hsv2rgb(hsv2);

gamma = 0.3;
hsv2(:,:,2) = hsv(:,:,2).^gamma;
J2 = hsv2rgb(hsv2);

figure(1), clf
subplot(1,3,1),imshow(I);
subplot(1,3,2),imshow(J1);
subplot(1,3,3),imshow(J2);
```

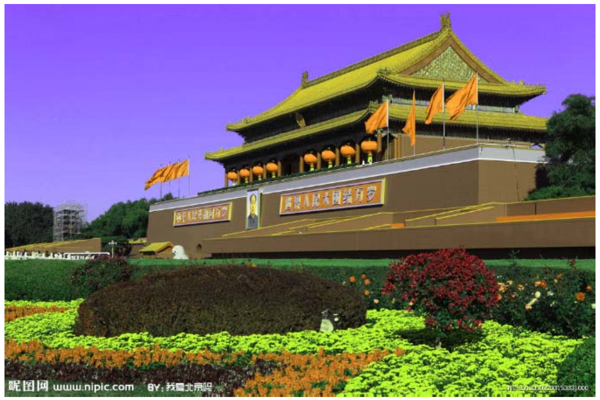
色调增强

- 与亮度、饱和度的增强相比，色调增强比较特殊。
- 在HSI空间中，色调对应的是角度，而且是循环的。
- 如果色调整体加或减一个数，颜色在色谱上移动。当数字比较小时，色调变冷或暖；当数字较大时，色调会出现剧烈变化。

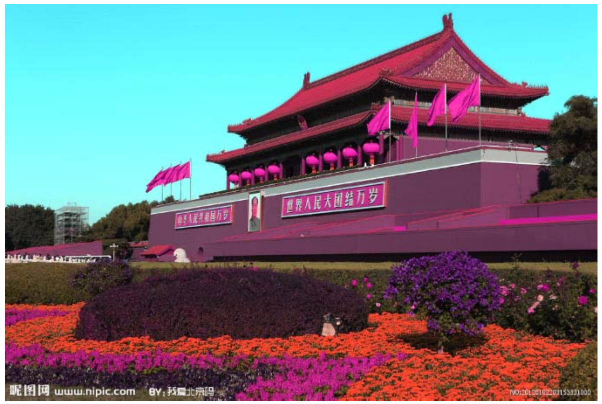
+0.05

-0.05





+0.1



-0.1

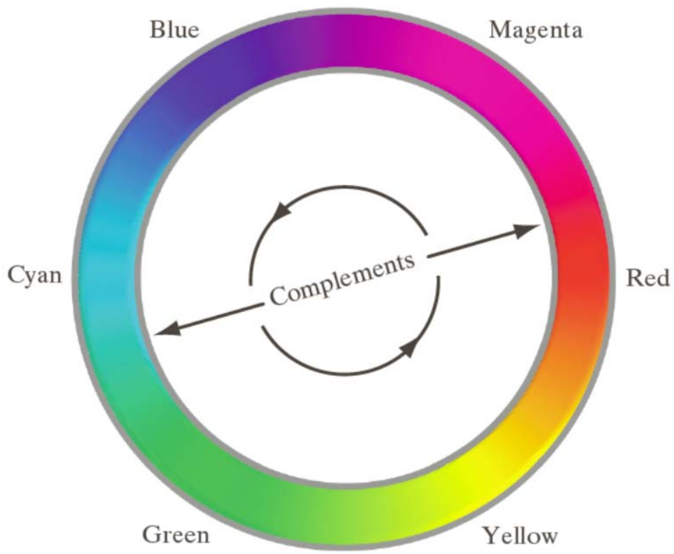
```matlab
function EnhanceHue()
% 2016-11-09

I = imread('..\data\Fig0636(woman_baby_original).tif');
% I = imread('..\data\saturation3.jpg');
I = im2double(I);
hsv = rgb2hsv(I);

hsv2 = hsv;
hsv2(:,:,1) = mod(255*(hsv(:,:,1)+0.05),256)/255;
J1 = hsv2rgb(hsv2);

hsv2(:,:,1) = mod(255*(hsv(:,:,1)-0.05),256)/255;
J2 = hsv2rgb(hsv2);

close all
figure(1)
subplot(1,3,1),imshow(I);
subplot(1,3,2),imshow(J1);
subplot(1,3,3),imshow(J2);
```

反色

RGB空间

HSI空间

```
function ComplementColor()
% 2016-11-09

I = imread('..\data\hue1.jpg');
I = im2double(I);
hsv = rgb2hsv(I);

J1 = 1-I;

hsv2 = hsv;
hsv2(:,:,1) = mod(255*(hsv(:,:,1)+0.5),256)/255;
hsv2(:,:,3) = 1-hsv2(:,:,3);
J2 = hsv2rgb(hsv2);

close all
figure(1)
subplot(1,3,1),imshow(I);
subplot(1,3,2),imshow(J1);
subplot(1,3,3),imshow(J2);
```



彩色图像分割

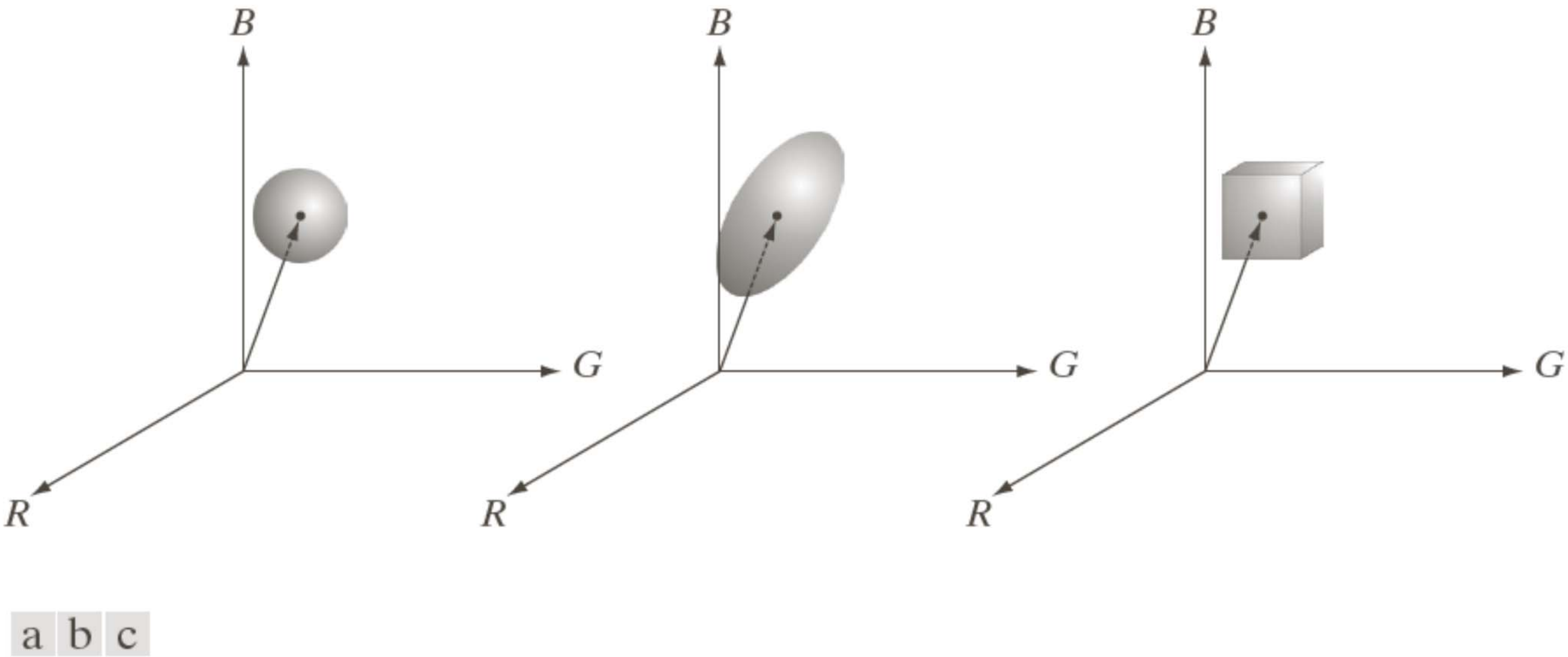


FIGURE 6.43
Three approaches for enclosing data regions for RGB vector segmentation.

```matlab
function ColorSlicing()

I=imread('..\data\red1.jpg');
I = im2double(I);
M = size(I,1);
N = size(I,2);
figure(1),imshow(I);

% choose point of interest
h = impoint;
pos = wait(h);
pos = round(pos);
a = [I(pos(2),pos(1),1) I(pos(2),pos(1),2) I(pos(2),pos(1),3)];
R = 70/255;% radius

D = (I(:,:,1)-a(1)).^2+(I(:,:,2)-a(2)).^2+(I(:,:,3)-a(3)).^2;
mask = D<=R*R;

% choose ROI
figure(2),imshow(mask);
h = imrect;
pos = wait(h);
pos = round(pos);
roi = false(size(mask));
roi(pos(2):pos(2)+pos(4),pos(1):pos(1)+pos(3)) = 1;
mask = mask & roi;

g = rgb2gray(I);
J = I;
J(:,:,1) = g;
J(:,:,2) = g;
J(:,:,3) = g;
idx = find(mask);
J(idx) = I(idx);
J(idx+M*N) = I(idx+M*N);
J(idx+M*N*2) = I(idx+M*N*2);
figure(3),imshow(J,[]);
```

彩色切割增强（Color slicing）

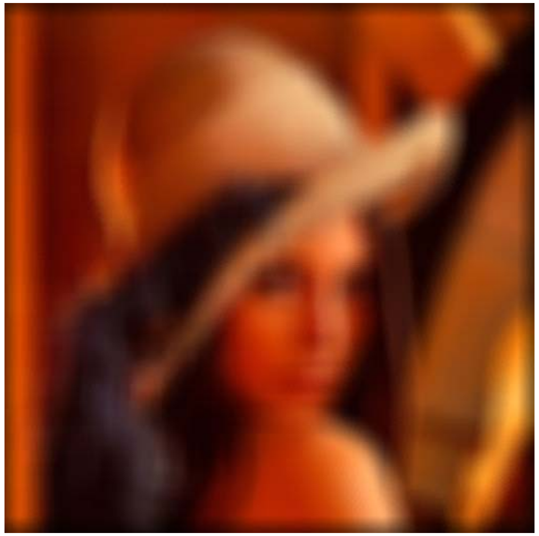
```matlab
function SelectiveColor()

I=imread('..\data\team.jpg');
M = size(I,1);
N = size(I,2);
figure(1),imshow(I);

if 0% edit mask
    h = imfreehand;
    position = wait(h);
    BW = poly2mask(position(:,1), position(:,2), M, N);
    figure(2),imshow(BW);
    imwrite(BW,'..\data\team_mask2.bmp');
    return
end

BW1 = imread('..\data\team_mask1.bmp');
BW2 = imread('..\data\team_mask2.bmp');
BW = BW1 | BW2;
figure(2),imshow(BW);

g = rgb2gray(I);
J = I;
J(:,:,1) = g;
J(:,:,2) = g;
J(:,:,3) = g;
idx = find(BW);
J(idx) = I(idx);
J(idx+M*N) = I(idx+M*N);
J(idx+M*N*2) = I(idx+M*N*2);
figure(3),imshow(J)
```

Selective Coloring

彩色图像平滑

I

J1: smooth rgb channels



J2: smooth intensity only

```matlab
function ColorSmooth()
% ColorSmooth
% 2016-11-05

I = imread('..\data\Fig0638(a)(lenna_RGB).tif');
I = im2double(I);

hsize = 31;% size of averaging filter
h = ones(hsize,hsize)/(hsize*hsize);

J1 = I;
for k = 1:3
    J1(:,:,k) = imfilter(I(:,:,k),h);
end

I_hsv = rgb2hsv(I);
J2_hsv = I_hsv;
J2_hsv(:,:,3) = imfilter(I_hsv(:,:,3),h);
J2 = hsv2rgb(J2_hsv);

close all
for k = 1:3
    figure(1),subplot(1,3,k),imshow(I(:,:,k))
    figure(2),subplot(1,3,k),imshow(I_hsv(:,:,k))
    figure(3),subplot(1,3,k),imshow(J1(:,:,k))
    figure(4),subplot(1,3,k),imshow(J2(:,:,k))
    figure(5),subplot(1,3,k),imshow(J2_hsv(:,:,k))
end
set(1,'name','I')
set(2,'name','I_hsv')
set(3,'name','J1')
set(4,'name','J2')
set(5,'name','J2_hsv')
figure(6),subplot(1,3,1),imshow(I),title('I')
subplot(1,3,2),imshow(J1),title('J1: smooth RGB channels')
subplot(1,3,3),imshow(J2),title('J2: smooth intensity of HSI only')
```

锐化

I

J

```matlab
function ColorSharp()
% ColorSharp
% 2016-11-08

I = imread('..\data\Fig0638(a)(lenna_RGB).tif');
I = im2double(I);

h = [1 1 1; 1 -8 1; 1 1 1];

J1 = I;
for k = 1:3
    temp = imfilter(I(:,:,k),h);
    J1(:,:,k) = I(:,:,k) - temp;
end

close all
figure(1),
for k = 1:3
    subplot(2,3,k),imshow(I(:,:,k))
    subplot(2,3,3+k),imshow(J1(:,:,k))
end

figure(6),ax(1)=subplot(1,2,1);imshow(I,[]),title('I')
ax(2)=subplot(1,2,2);imshow(J1,[]),title('J')
linkaxes(ax);
```

噪声

```matlab
function GaussianNoise()
% 2016-11-08

close all
Ir = imread('..\data\Fig0648(a)(lenna-noise-R-gauss-mean0-var800).tif');
Ig = imread('..\data\Fig0648(b)(lenna-noise-G-gauss-mean0-var800).tif');
Ib = imread('..\data\Fig0648(c)(lenna-noise-B-gauss-mean0-var800).tif');

I = zeros(size(Ir,1),size(Ir,2),3);
I(:,:,1) = im2double(Ir(:,:,1));
I(:,:,2) = im2double(Ig(:,:,1));
I(:,:,3) = im2double(Ib(:,:,1));

figure(1),
subplot(2,2,1),imshow(I)
subplot(2,2,2),imshow(I(:,:,1))
subplot(2,2,3),imshow(I(:,:,2))
subplot(2,2,4),imshow(I(:,:,3))

hsv_image = rgb2hsv(I);
figure(2),
subplot(2,3,1),imshow(I(:,:,1)),title('R');
subplot(2,3,2),imshow(I(:,:,2)),title('G');
subplot(2,3,3),imshow(I(:,:,3)),title('B');
subplot(2,3,4),imshow(hsv_image(:,:,1)),title('H');
subplot(2,3,5),imshow(hsv_image(:,:,2)),title('S');
subplot(2,3,6),imshow(hsv_image(:,:,3)),title('I');
```

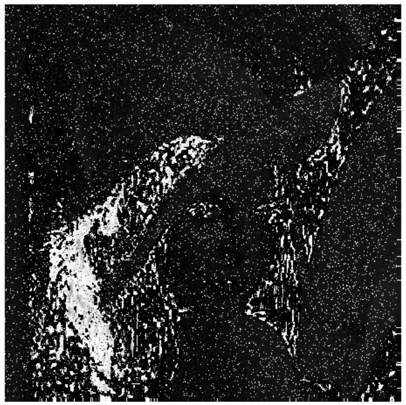
单通道噪声

R

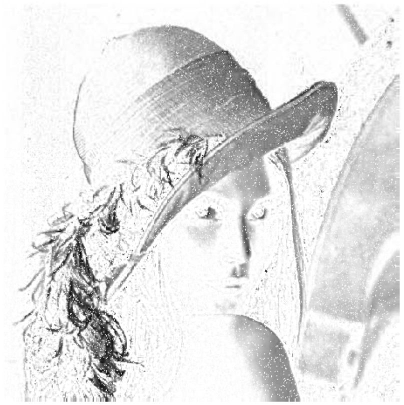
G

B

H



S

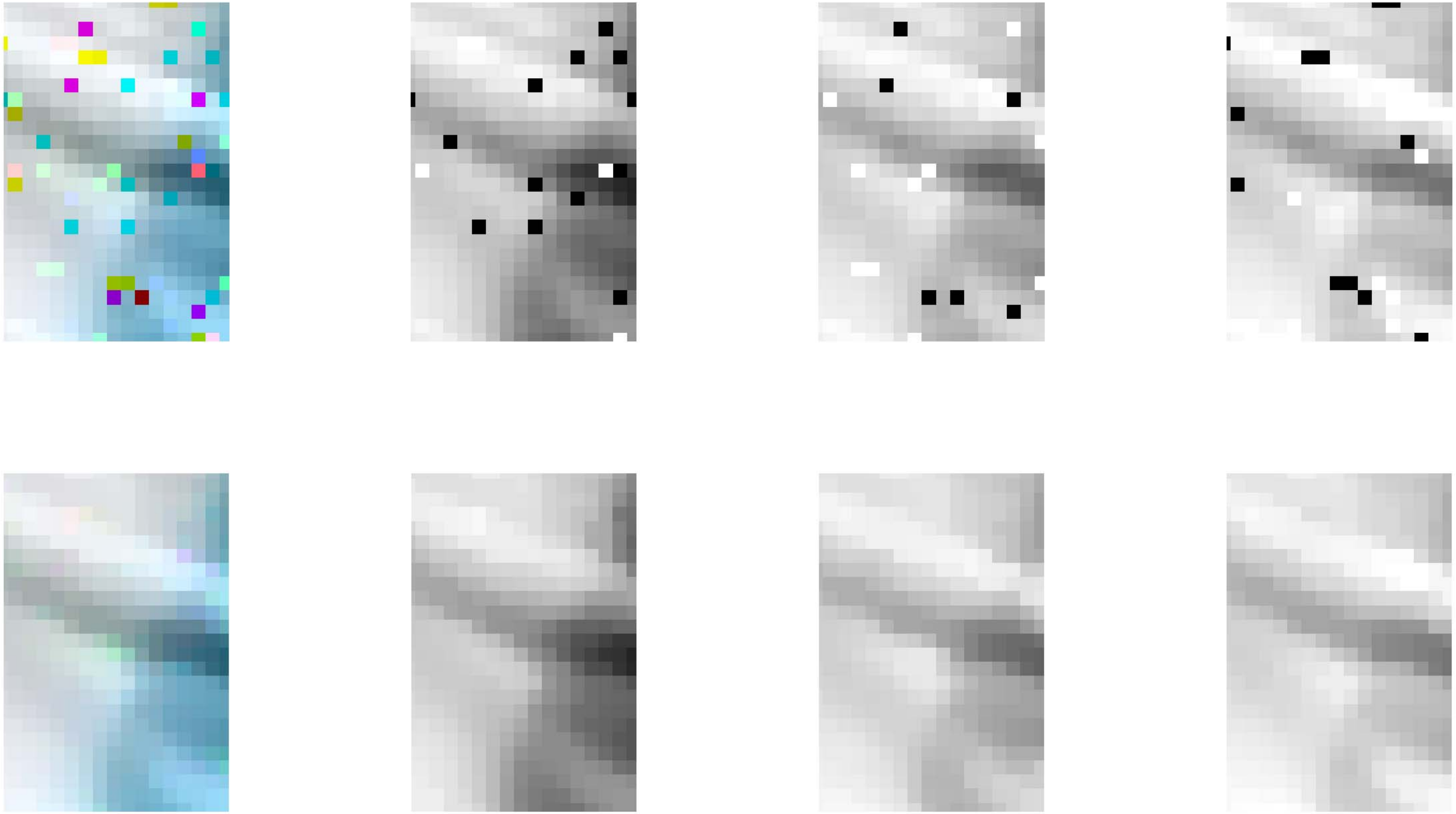


I

```matlab
function GaussianNoiseGreen()
% 2016-11-08

close all
I = imread('..\data\Fig0650(a)(rgb_image_G_saltpep_pt05).tif');

figure(1),
subplot(2,2,1),imshow(I)
subplot(2,2,2),imshow(I(:,:,1))
subplot(2,2,3),imshow(I(:,:,2))
subplot(2,2,4),imshow(I(:,:,3))

hsv_image = rgb2hsv(I);
figure(2),
subplot(2,3,1),imshow(I(:,:,1)),title('R');
subplot(2,3,2),imshow(I(:,:,2)),title('G');
subplot(2,3,3),imshow(I(:,:,3)),title('B');
subplot(2,3,4),imshow(hsv_image(:,:,1)),title('H');
subplot(2,3,5),imshow(hsv_image(:,:,2)),title('S');
subplot(2,3,6),imshow(hsv_image(:,:,3)),title('I');
```

中值滤波

排序滤波器直接用于各个通道，会出问题。



```matlab
function ColorMidFilter()
% ColorMidFilter
% 2016-11-08

I = imread('..\data\player2.jpg');
I = im2double(I);

I2 = I;
for k = 1:3
    I2(:,:,k) = imnoise(I(:,:,k),'salt & pepper',0.05);
end

J = I2;
for k = 1:3
    J(:,:,k) = medfilt2(I2(:,:,k),[3 3]);
end

close all
figure(1),
ax1(1)=subplot(1,3,1);imshow(I),title('I')
ax1(2)=subplot(1,3,2);imshow(I2),title('I2')
ax1(3)=subplot(1,3,3);imshow(J),title('J')
linkaxes(ax1);

figure(2),
ax2(1) = subplot(2,4,1); imshow(I2)
ax2(5) = subplot(2,4,5); imshow(J)
for k = 1:3
    ax2(1+k)=subplot(2,4,1+k);imshow(I2(:,:,k))
    ax2(1+k+4)=subplot(2,4,1+k+4);imshow(J(:,:,k))
end
linkaxes(ax2);
```