

## 9.3 实验指导



### 9.3.1 灰度级到彩色转换

#### 1. 实验内容

对一幅灰度图像实行到彩色的转换。

#### 2. 实验原理

对任何输入像素的灰度级执行 3 个独立的变换，接着把 3 个变换结果分别送入彩色电视监视器的红、绿、蓝通道。这种方法产生一幅合成图像，其彩色内容受变换函数特性的调制。3 个变换都是基于平滑的非线性函数，这种方法在技术上有相当大的灵活性。

#### 3. 实验方法及程序

实验方法主要由如下两步组成。

(1) 确定红、绿、蓝 3 个变换器的变换函数。

(2) 合成彩色。

其参考程序设计如下。

```
clear all;
a=imread('i_boat_gray.bmp');
c=zeros(size(a));
pos=find((a>=60)&(a<105));
c(pos)=a(pos);
b(:,:,3)=c;
c=zeros(size(a));
pos=find((a>=105)&(a<150));
c(pos)=a(pos);
b(:,:,2)=c;
c=zeros(size(a));
pos=find(a>=150);
c(pos)=a(pos);
b(:,:,1)=c;
b=uint8(b);
figure;
subplot(1,2,1), imshow(a,[]), title('原始图像');
subplot(1,2,2), imshow(b,[]), title('灰度彩色变换后图像');
```

#### 4. 实验结果与分析

实验结果如图 9.7 所示。

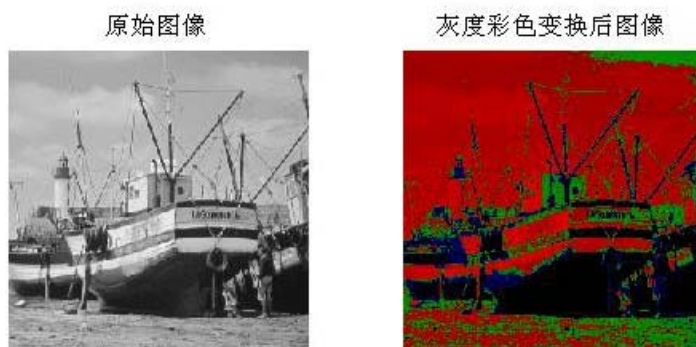


图 9.7 灰度到彩色变换处理的结果

从实验结果可以看出：灰度图像变换成伪彩色图像，图像内容的层次变得较清晰，改善了图像的视觉效果。

### 5. 思考题

- (1) 对参考程序给出功能注释。
- (2) 写出变换的变换函数。
- (3) 选择另一个变换函数进行实验。



## 9.3.2 彩色图像的两种平滑滤波

### 1. 实验内容

对一幅彩色图像采用如下两种方法实现平滑滤波，并对结果进行比较。

(1) 彩色图像用传统的邻域平均单独平滑  $R$ 、 $G$ 、 $B$  3 个分量图像，再对平滑后的分量图像合成。

(2) 将  $RGB$  图像转换到  $HSI$  模型，仅对强度分量  $I$  进行平滑处理，而保持色度和饱和度分量不变，再将其转换到  $RGB$  空间进行显示。

### 2. 实验原理

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

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

式中，

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

可以看出，如标量图像那样，该向量分量可以用传统的灰度邻域处理单独的平滑 RGB 图像的每一平面得到。

### 3. 实验方法及程序

实验方法如下。

(1) 彩色图像用传统的邻域平均单独平滑  $R$ 、 $G$ 、 $B$  3 个分量图像，再对平滑后的分量图像合成。

(2) 将 RGB 图像转换到 HSI 模型，仅对强度分量  $I$  进行平滑处理，而保持色度和饱和度分量不变，再将其转换到 RGB 空间进行显示。

(3) 计算两种平滑滤波结果的差。

其参考程序设计如下。

```
rgb1=imread('i_flower673.jpg');
rgb=im2double(rgb1);
fR=rgb(:,:,1);
fG=rgb(:,:,2);
fB=rgb(:,:,3);
w=fspecial('average');
fR_filtered=imfilter(fR,w);
fG_filtered=imfilter(fG,w);
fB_filtered=imfilter(fB,w);
rgb_filtered=cat(3,fR_filtered,fG_filtered,fB_filtered);
I1=(fR+fG+fB)/3;
tmp1=min(min(fR,fG),fB);
tmp2=fR+fG+fB;
tmp2(tmp2==0)=eps;
S=1-3.*tmp1./tmp2;
tmp1=0.5*((fR-fG)+(fR-fB));
tmp2=sqrt((fR-fG).^2+(fR-fB).*(fG-fB));
theta=acos(tmp1./(tmp2+eps));
H1=theta;
H1(fB>fG)=2*pi-H1(fB>fG);
H1=H1/(2*pi);
H1(S==0)=0;
w=fspecial('average');
I=imfilter(I1,w);
H=H1*2*pi;
R=zeros(size(rgb1,1),size(rgb1,2));
G=zeros(size(rgb1,1),size(rgb1,2));
B=zeros(size(rgb1,1),size(rgb1,2));
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)=1.0-(R(ind)+B(ind));
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)=1.0-(R(ind)+G(ind));
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)=1.0-(G(ind)+B(ind));
hsi_rgb=cat(3,R,G,B);
diff=(abs(hsi_rgb-rgb_filtered));
    
```

#### 4. 实验结果与分析

实验结果如图 9.8 所示。

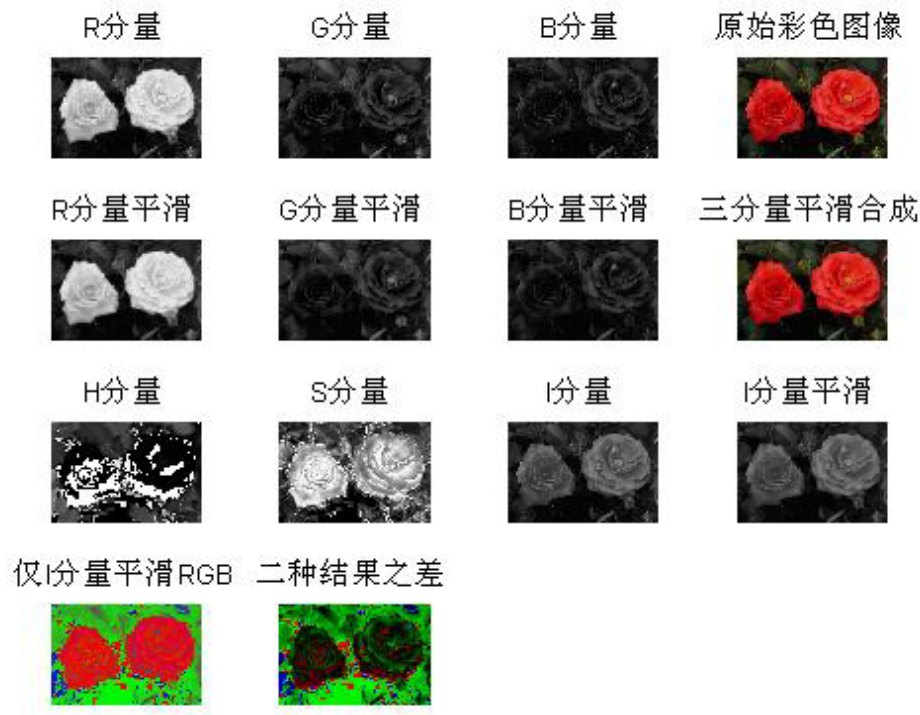


图 9.8 彩色图像的平滑滤波

从实验结果可以看出：用两种方法对图像进行平滑所得的结果不完全相同。

#### 5. 思考题

- (1) 对参考程序给出功能注释。
- (2) 对实验结果进行详细分析。



### 9.3.3 彩色图像的两种锐化滤波

#### 1. 实验内容

对一幅彩色图像采用如下两种方法实现锐化滤波，并对结果进行比较。

(1) 彩色图像用传统的 Laplacian 分别锐化  $R$ 、 $G$ 、 $B$  3 个分量图像，再对锐化后的分量图像合成。

(2) 将 RGB 图像转换到 HSI 模型，仅对强度分量  $I$  进行锐化处理，而保持色度和饱和度分量不变，再将其转换到 RGB 空间进行显示。

#### 2. 实验原理

在 RGB 彩色系统中，向量  $c$  的 Laplacian 变换为

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

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

#### 3. 实验方法及程序

实验方法如下。

(1) 彩色图像用传统的 Laplacian 分别锐化  $R$ 、 $G$ 、 $B$  3 个分量图像，再对锐化后的分量图像合成。

(2) 将 RGB 图像转换到 HSI 模型，仅对强度分量  $I$  进行锐化处理，而保持色度和饱和度分量不变，再将其转换到 RGB 空间进行显示。

(3) 计算两种锐化滤波结果的差。

其参考程序设计如下。

```
rgb1=imread('i_flower673.jpg');
rgb=im2double(rgb1);
fR=rgb(:,:,1);
fG=rgb(:,:,2);
fB=rgb(:,:,3);
lapMatrix=[1 1 1;1 -8 1;1 1 1];
fR_filtered=imfilter(fR,lapMatrix,'replicate');
fG_filtered=imfilter(fG,lapMatrix,'replicate');
fB_filtered=imfilter(fB,lapMatrix,'replicate');
rgb_tmp=cat(3,fR_filtered,fG_filtered,fB_filtered);
rgb_filtered=imsubtract(rgb,rgb_tmp);
I1=(fR+fG+fB)/3;
tmp1=min(min(fR,fG),fB);
tmp2=fR+fG+fB;
```

```

tmp2(tmp2==0)=eps;
S=1-3.*tmp1./tmp2;
tmp1=0.5*((fR-fG)+(fR-fB));
tmp2=sqrt((fR-fG).^2+(fR-fB).*(fG-fB));
theta=acos(tmp1./(tmp2+eps));
H1=theta;
H1(fB>fG)=2*pi-H1(fB>fG);
H1=H1/(2*pi);
H1(S==0)=0;
lapMatrix=[1 1 1;1 -8 1;1 1 1];
I=imfilter(I1,lapMatrix,'replicate');
    
```

#### 4. 实验结果与分析

实验结果如图 9.9 所示。

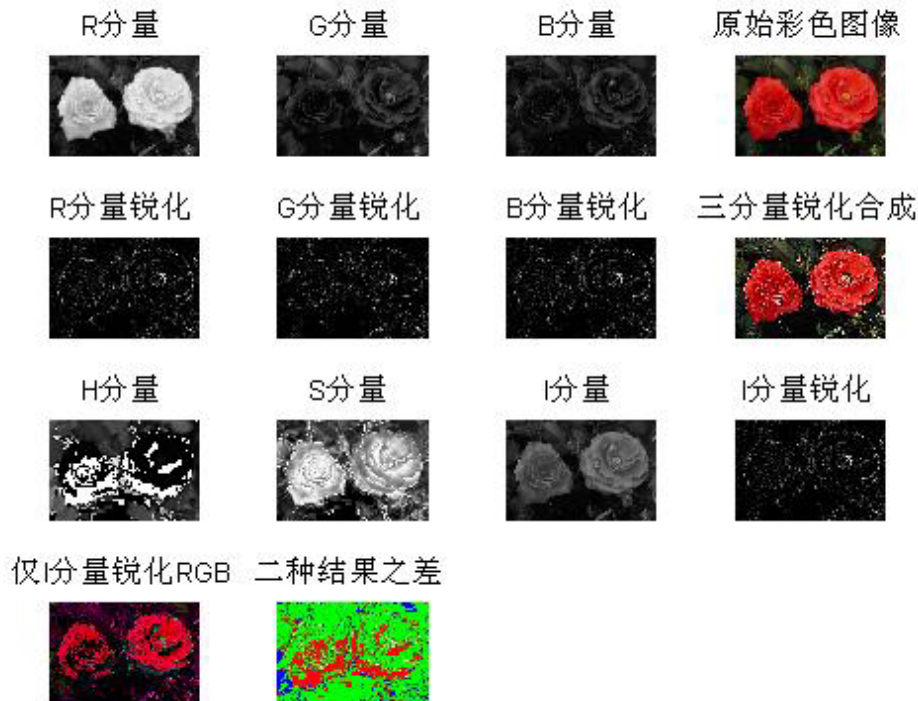


图 9.9 彩色图像的锐化

从实验结果可以看出：用两种方法对图像进行锐化所得的结果不完全相同。

#### 5. 思考题

- (1) 对参考程序给出功能注释。
- (2) 对实验结果进行详细分析。



### 9.3.4 彩色图像的边缘检测

#### 1. 实验内容

对一幅彩色图像，通过计算向量梯度的幅值进行边缘检测。

#### 2. 实验原理

令  $r, g, b$  是 RGB 彩色空间沿  $R, G, B$  轴的单位向量，可定义向量为

$$u = \frac{\partial R}{\partial x} r + \frac{\partial G}{\partial x} g + \frac{\partial B}{\partial x} b$$

$$v = \frac{\partial R}{\partial y} r + \frac{\partial G}{\partial y} g + \frac{\partial B}{\partial y} b$$

数量  $g_{xx}, g_{yy}, g_{xy}$  定义为这些向量的点乘，如下所示：

$$g_{xx} = u \bullet u = u^T u = \left| \frac{\partial R}{\partial x} \right|^2 + \left| \frac{\partial G}{\partial x} \right|^2 + \left| \frac{\partial B}{\partial x} \right|^2$$

$$g_{yy} = v \bullet v = v^T v = \left| \frac{\partial R}{\partial y} \right|^2 + \left| \frac{\partial G}{\partial y} \right|^2 + \left| \frac{\partial B}{\partial y} \right|^2$$

$$g_{xy} = u \bullet v = u^T v = \frac{\partial R}{\partial x} \frac{\partial R}{\partial y} + \frac{\partial G}{\partial x} \frac{\partial G}{\partial y} + \frac{\partial B}{\partial x} \frac{\partial B}{\partial y}$$

利用该表示法， $c(x, y)$  的最大变化率方向可以由角度给出：

$$\theta = \frac{1}{2} \arctan \left[ \frac{2g_{xy}}{(g_{xx} - g_{yy})} \right]$$

$(x, y)$  点在  $\theta$  方向上变化率的幅值由下式给出：

$$F(\theta) = \left\{ \frac{1}{2} [(g_{xx} + g_{yy}) + (g_{xx} - g_{yy}) \cos 2\theta + 2g_{xy} \sin 2\theta] \right\}^{\frac{1}{2}}$$

#### 3. 实验方法及程序

实验方法如下。

(1) 彩色图像向量梯度的计算。

(2) 根据彩色图像向量梯度的幅值检测边缘。

其参考程序设计如下。

```
rgb=imread('fly01.jpg');
sob=fspecial('sobel');
Rx=imfilter(double(rgb(:,:,1)),sob,'replicate');
Ry=imfilter(double(rgb(:,:,1)),sob,'replicate');
Gx=imfilter(double(rgb(:,:,2)),sob,'replicate');
Gy=imfilter(double(rgb(:,:,2)),sob,'replicate');
Bx=imfilter(double(rgb(:,:,3)),sob,'replicate');
By=imfilter(double(rgb(:,:,3)),sob,'replicate');
gxx=Rx.^2+Gx.^2+Bx.^2;
gyy=Ry.^2+Gy.^2+By.^2;
```

```
gxy=Rx.*Ry+Gx.*Gy+Bx.*By;
theta=0.5*(atan(2*gxy./(gxx-gyy+eps)));
G1=0.5*((gxx+gyy)+(gxx-gyy).*cos(2*theta)+2*gxy.*sin(2*theta));
theta=theta+pi/2;
G2=0.5*((gxx+gyy)+(gxx-gyy).*cos(2*theta)+2*gxy.*sin(2*theta));
G1=G1.^0.5;
G2=G2.^0.5;
rgb_gradiant=mat2gray(max(G1,G2));
```

#### 4. 实验结果与分析

实验结果如图 9.10 所示。



图 9.10 利用梯度进行彩色图像的边缘检测

从实验得到的边缘图像可以看出，飞机的边缘细节都能够较好地提取出来。

#### 5. 思考题

- (1) 对参考程序给出功能注释。
- (2) 对实验结果进行详细分析。