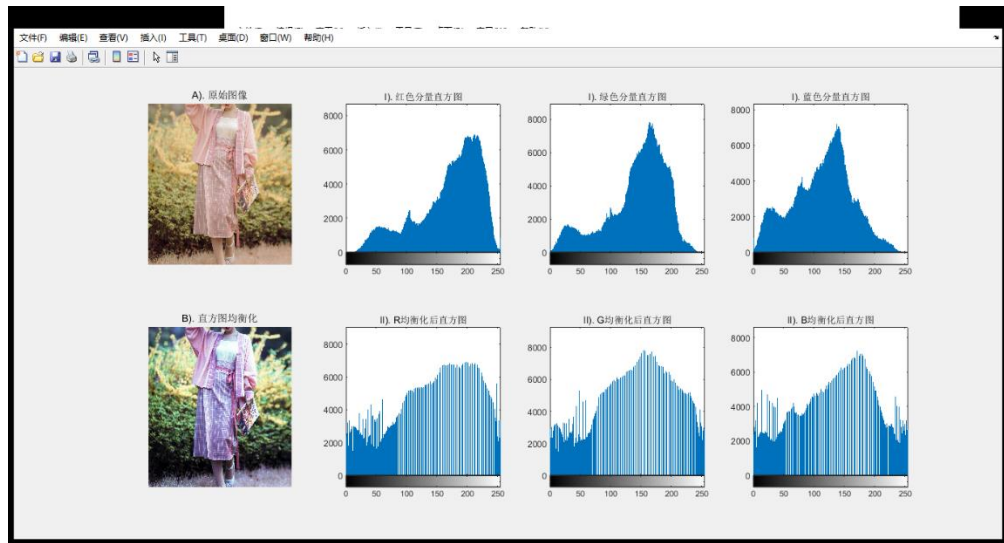


1. 彩色直方图均衡化

对 R、G、B 三个通道进行直方图均衡化，然后将处理后的三通道重构成一张 RGB 图。

(1) 结果



(2) 源程序

```
function Histogramequalization()
[f,p]=uigetfile('*.tif','TIFF.tif');%YF.jpg
if f
I=imread(strcat(p,f));
end

h=figure();
set(h,'position',[300 100 1000 600]);

%提取RGB分量
Ir=I(:,:,1);
Ig=I(:,:,2);
Ib=I(:,:,3);
I1=histogram(Ir);
I2=histogram(Ig);
I3=histogram(Ib);
c=cat(3,I1,I2,I3);
subplot(2,4,1);imshow(I);
title('A. 原始图像');
subplot(2,4,5);imshow(c);
title('B. 直方图均衡化');

subplot(2,4,2);
imhist(Ir);
title('I. 红色分量直方图');
subplot(2,4,6);
imhist(I1);
title('II. R均衡化后直方图');
subplot(2,4,3);
imhist(Ig);
title('I. 绿色分量直方图');
subplot(2,4,7);
imhist(I2);
title('II. G均衡化后直方图');
```

```

subplot(2,4,4);
imhist(Ib);
title('I). 蓝色分量直方图');
subplot(2,4,8);
imhist(I3);
title('II). B均衡化后直方图');

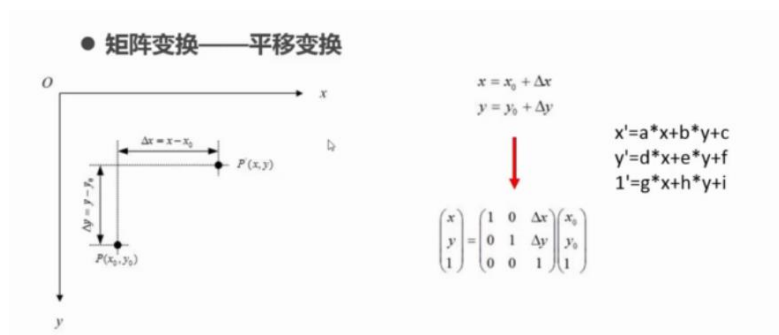
function d=histogram(I)
J=I;
[m,n]=size(I);      %确定矩阵大小
area=m*n;
a=zeros(1,256);
b=zeros(1,256);
for i=1:m
    for j=1:n
        d=I(i,j)+1;
        a(1,d)=a(1,d)+1;
    end
end
for i=1:256
    sum=0;
    for j=1:i
        sum=sum+a(1,j);
    end
    b(1,i)=sum*255/area;
end
for i=1:m      %使用均衡化数据代替原始数据
    for j=1:n
        d=J(i,j)+1;
        J(i,j)=b(1,d);
    end
end
end
d=J;

```

2.

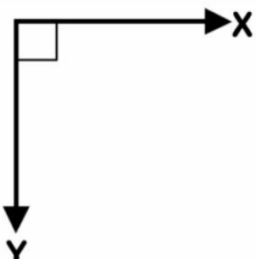
(1) 像素间的坐标变换原理

1) 平移变换



2) 尺度变换（伸缩）

● 矩阵变换——缩放变换



$$X = K1 * x0$$

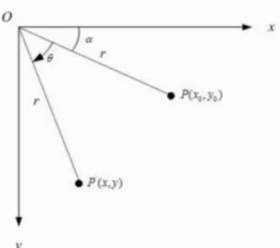
$$Y = K2 * y0$$

$$\begin{matrix} x' = a * x + b * y + c \\ y' = d * x + e * y + f \\ 1' = g * x + h * y + i \end{matrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1' \end{pmatrix} = \begin{pmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix}$$

3) 旋转变换

● 矩阵变换——旋转变换



$$x_0 = r \cos \alpha$$

$$y_0 = r \sin \alpha$$

$$x = r \cos(\alpha + \theta) = r \cos \alpha \cos \theta - r \sin \alpha \sin \theta = x_0 \cos \theta - y_0 \sin \theta$$

$$y = r \sin(\alpha + \theta) = r \sin \alpha \cos \theta + r \cos \alpha \sin \theta = y_0 \cos \theta + x_0 \sin \theta$$

$$\begin{matrix} x' = a * x + b * y + c \\ y' = d * x + e * y + f \\ 1' = g * x + h * y + i \end{matrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix}$$

4) 镜像变换

假设原图像的高度为 h ，宽度为 w ，变换后，图像的尺寸不变。那么原图像中 (x_0, y_0) 经过水平镜像后坐标变为 $(w-1-x_0, y_0)$ 。用矩阵表示为：

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & w-1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

逆变换为：

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & w-1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

同理图像经过垂直镜像变换后的逆运算为

$$\begin{cases} x_0 = x_1 \\ y_0 = h - y_1 - 1 \end{cases}$$

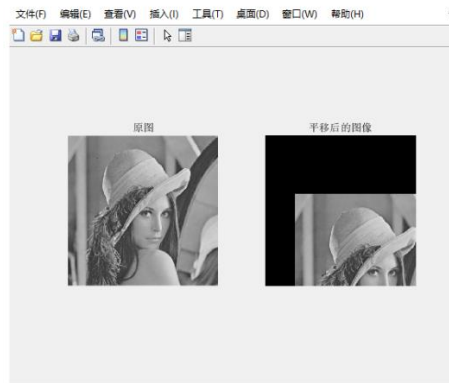
对角镜像的逆运算为：

$$\begin{cases} x_0 = w - x_1 - 1 \\ y_0 = h - y_1 - 1 \end{cases}$$

(2) 编程实现灰度 lena 图像的平移变换、尺度变换（伸缩）、旋转变换和镜像变换

1) 平移变换

a) 结果



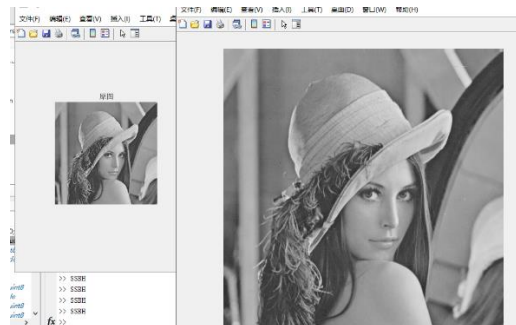
b) 源程序

```
x0=100;y0=200;
img=imread('lena.tiff');
subplot(1,2,1);imshow(img);title('原图');

A=[1 0 0 ; 0 1 0 ; x0 y0 1 ];
tform =maketform('affine',A);
X=imtransform(img,tform,'XData',[1 size(img,2)],'YData',[1 size(img,1)]);
figure;
subplot(1,2,2);imshow(X);title('平移后的图像');
```

2) 尺度变换（伸缩）

a) 结果



b) 源程序

```
img1=imread('lena.tiff');
imshow(img1);
subplot(1,2,1);imshow(img1);title('原图')
img4=imresize(img1,2);
figure;
subplot(1,2,1);imshow(img4);
```

3) 尺度变换（伸缩）

a) 结果



b) 源程序

```
img1=imread('lena.tiff');
%figure;
imshow(img1);
%figure;
subplot(1,2,1);imshow(img1);title('原图')
img3=imrotate(img1,180);
subplot(1,2,2);imshow(img3);title('旋转后的图像');
```

4) 尺度变换 (伸缩)

a) 结果



b) 源程序

```
img = imread('lena.tiff');
img = im2double(img);
subplot(1,2,1);
imshow(img);
title('原图');

[M,N,Z]=size(img);g=zeros(M,N,Z);
for i=1:M
    for j=1:N
        for s=1:Z
            g(i,j,s) = img(i,N-j+1,s);
        end
    end
end
subplot(1,2,2);
imshow(g);
title('镜像后的图像');
```

3.

(1) 图像的彩色空间表示及其转换公式

1) RGB 转 HSI

$$H = \begin{cases} \theta, & G \geq B \\ 2\pi - \theta, & G < B \end{cases}$$

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

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

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

2) RGB 转 HSV

$$R' = R / 255$$

$$G' = G / 255$$

$$B' = B / 255$$

$$C_{\max} = \max(R', G', B')$$

$$C_{\min} = \min(R', G', B')$$

$$\Delta = C_{\max} - C_{\min}$$

H 计算:

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} + 0 \right) & , C_{\max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right) & , C_{\max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right) & , C_{\max} = B' \end{cases}$$

S 计算:

$$S = \begin{cases} 0 & C_{\max} = 0 \\ \frac{\Delta}{C_{\max}} & C_{\max} \neq 0 \end{cases}$$

V 计算:

$$V = C_{\max}$$

3) RGB 转 YCbCr

ITU^① 在 BT. 601[50] 中规定: $w_R = 0.299, w_B = 0.114 (w_G = 1 - w_B - w_R = 0.587)$ 。有了这些值, 变换可以写为:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (12.34)$$

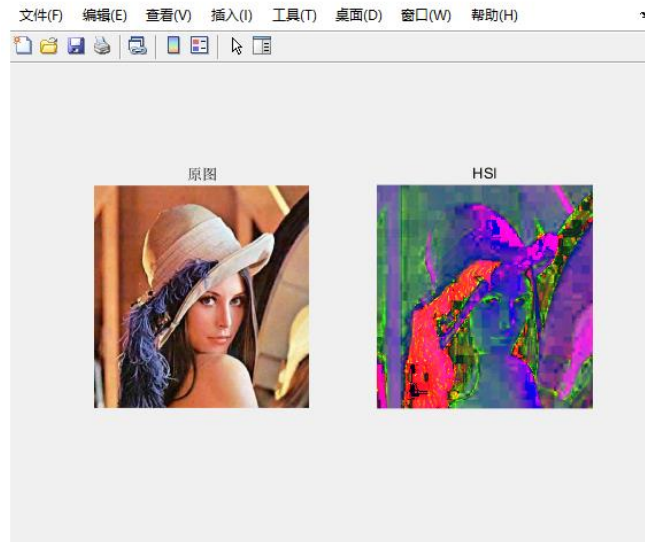
反变换可写为:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.403 \\ 1.000 & -0.344 & -0.714 \\ 1.000 & 1.773 & 0.000 \end{bmatrix} \cdot \begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} \quad (12.35)$$

(2) 编程实现彩色 lena 图像的 RGB 空间与 HSI、HSV、YCbCr 等颜色空间的转换。

1) HSI 转换

a) 结果

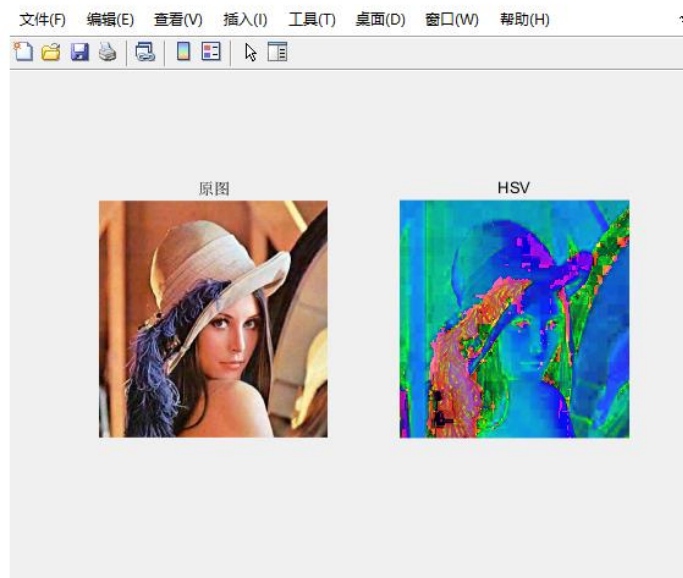


b) 源程序

```
f = imread('lena.jpg');
g = rgb2hsi(f);
subplot(1, 2, 1), imshow(f), title('原图');
subplot(1, 2, 2), imshow(g), title('HSI');
```

2) HSV 转换

a) 结果

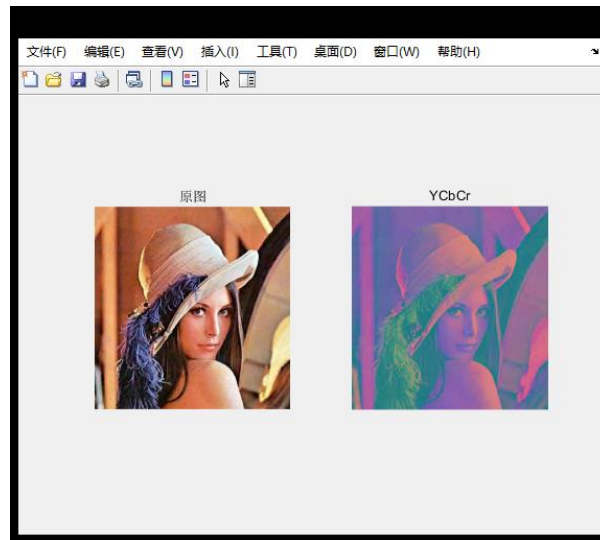


b) 源程序

```
f = imread('lena.jpg');
g = rgb2hsv(f);
subplot(1, 2, 1), imshow(f), title('原图');
subplot(1, 2, 2), imshow(g), title('HSV');
```

3) YCbCr 转换

a) 结果



b) 源程序

```
P = imread('lena.jpg');  
g = rgb2ycbcr(P);  
subplot(1, 2, 1), imshow(P), title('原图');  
subplot(1, 2, 2), imshow(g), title('YCbCr');
```