安徽大学《数字图像处理（双语）》实验报告（6）

学号 WA2224013 专业 机器人工程 姓名 郭义月

实验日期 **2024.12.31** 教师签字 成绩

**【实验名称】:**

**彩色图像处理**

**【实验目的】：**

**1. 熟悉和掌握彩色图像空间**

**2. 通过MATLAB编程实现彩色图像的反色**

**3. 通过MATLAB编程实现两种常见的彩色图像分层**

**4. 通过MATLAB编程实现不同空间的彩色图像直方图均衡**

**【实验内容】**

**Intensity Slicing**

(a) Implement intensity slicing, with the characteristic that you can specify different ranges of gray-level values for the input image and your program will output an RGB image whose pixels have a specified color. You can set the colors in color palette.

(b) Process the above images with your program with different ranges (8 for (1), 3 for (2), 20 for (3)).

|  |  |
| --- | --- |
|  |  |
| picker\_phantom.tif | (2)weld-original.tif |



(3) tropical\_rain\_grayscale.tif

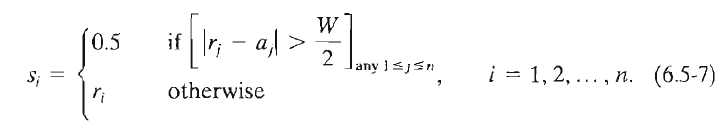
**Color complement**

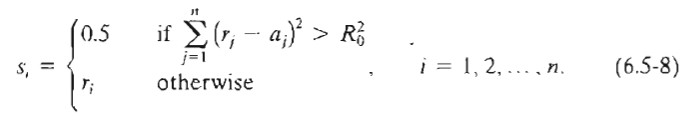
Use the complement function (as show in following figure 6.33(b)) of R,G,B components and H, S, I components respectively to implement the color image complement of image ‘strawberries’ .



**Color slicing**

Using the following two equation to implement the color slicing on image ‘strawberries’ where n=3 denotes the Red, Green and Blue component. Set W= 0.2549, R=0.1765, a = (0.6863, 0.1608, 0.1922). Pixels outside the cube and sphere were replaced by color (0.5, 0.5, 0.5)





**Histogram Processing on Color Images**

The following image “Fig0637(a)(caster\_stand\_original)” is a color image of a caster stand containing cruets and shakers.

(a) Demonstrate the histogram of its intensity component

(b) Implement the Histogram Equalizing on the intensity component, without altering the hue and saturation. Comparing the result to the original image.

(c) Further increase the image's saturation component and compare the result to (b)

(d) Demonstrate the histogram of its intensity component of the processed image after (c) and compare the distribution to (a).

**【实验代码和结果】**

**Task 1：Intensity Slicing**

(a) Implement intensity slicing, with the characteristic that you can specify different ranges of gray-level values for the input image and your program will output an RGB image whose pixels have a specified color. You can set the colors in color palette.

clc;clear;close all;

image1 = im2double(imread('picker\_phantom.tif'));

image2 = im2double(imread('weld-original.tif'));

image3 = im2double(imread('tropical\_rain\_grayscale.tif'));

intensity\_slicing(image1,8)

intensity\_slicing(image2,3)

intensity\_slicing(image3,20)

function intensity\_slicing(image1,r)

sp1 = (max(image1(:)) - min(image1(:)))/r;

range1 = zeros(r,2);

for i = 1:r

range1(i,1) = sp1\*(i-1);

range1(i,2) = sp1\*i;

end

colorPalette1 = lines(r); %可以生成r个颜色

numColors = 64; % 比需要的数量多一些

colorPaletteFull = lines(numColors);

% 等距采样 8 个颜色

% r = 8;

indices = round(linspace(1, numColors, r));

colorPalette2 = colorPaletteFull(indices, :);

% r = 8;

linesColors = lines(r);

parulaColors = parula(r);

% 平均两种调色板

colorPalette3 = (linesColors + parulaColors) / 2;

[x1,y1] = size(image1);

image11 = zeros(x1,y1,3);

image22 = zeros(x1,y1,3);

image33 = zeros(x1,y1,3);

% imshow(image11)

for i = 1:x1

for j = 1:y1

for k = 1:r

if image1(i,j)>range1(k,1)&&image1(i,j)<range1(k,2)

image11(i,j,:) = colorPalette1(k,:,:);

image22(i,j,:) = colorPalette2(k,:,:);

image33(i,j,:) = colorPalette3(k,:,:);

break;

end

end

end

end

figure;

subplot(2,2,1);

imshow(image1);

title('original image');

subplot(2,2,2);

imshow(image11)

title('intensity slicing1');

subplot(2,2,3);

imshow(image22)

title('intensity slicing2');

subplot(2,2,4);

imshow(image33)

title('intensity slicing3');

end

实验结果：为达到ppt上较合理的伪彩色颜色分配，分别用三种不同的方法输出不同的彩色颜色，实验结果如下：

8 for picker\_phantom.tif

|  |  |
| --- | --- |
|  |  |
|  |  |

3 for weld-original.tif

|  |  |
| --- | --- |
|  |  |
|  |  |

20 for tropical\_rain\_grayscale.tif

|  |  |
| --- | --- |
|  |  |
|  |  |

**Task 2：Color complement**

RGB实验代码：

clc;clear;close all;

image = im2double(imread('Fig\_strawberries.tif'));

image1 = 1 - image;

subplot(1,2,1);

imshow(image);

title('original image');

subplot(1,2,2);

imshow(image1);

title('RGB complement')

实验结果：

|  |  |
| --- | --- |
|  |  |

HSI实验代码：

RGB2HSI函数：

function HSI = myRGB2HSI(RGB)

% 从RGB颜色空间向HSI颜色空间的转换

% RGB(uint8):输入的RGB彩色图像

% HSI(double):转换后的HSI彩色图像

img1\_double = im2double(RGB); % 转成double并作归一化处理

[r, c, k] = size(img1\_double);

H = zeros(r, c);

S = zeros(r, c);

I = zeros(r, c);

for i = 1 : r

for j = 1 : c

% 分别获取R,G,B分量

R = img1\_double(i,j,1);

G = img1\_double(i,j,2);

B = img1\_double(i,j,3);

fenzi = 0.5 \* ( (R-G)+(R-B) );

fenmu = sqrt( (R-G)^2 + (R-B)\*(G-B) );

% 易错点：分母需加上eps防止为0

xita = acos( fenzi/(fenmu+eps) );

if ( B<=G )

HSI(i,j,1) = xita;

else

HSI(i,j,1) = 2\*pi-xita;

end

HSI(i,j,1) = HSI(i,j,1) / (2\*pi); % H分量需要除以2\*pi进行归一化

min\_value = min(min(R,G),B);

% 易错点：分母需加上eps防止为0

HSI(i,j,2) = 1 - ( 3/(R+G+B+eps) ) \* min\_value;

HSI(i,j,3) = (R+G+B)/3;

end

end

end

HSI2RGB函数：

function RGB = myHSI2RGB(HSI)

% 从HSI颜色空间向RGB颜色空间的转换

% HSI(double):输入的HSI彩色图像

% RGB(uint8): 转换后的RGB彩色图像

HSI = im2double(HSI);

[r,c,k] = size(HSI);

RGB = zeros(r,c,k);

for i = 1 : r

for j = 1 : c

H = HSI(i,j,1)\*2\*pi;

S = HSI(i,j,2);

I = HSI(i,j,3);

if ( H>=0 && H<2/3\*pi)

expression = S\*cos( H )/(cos( pi/3-H ) + eps);

RGB(i,j,1) = I \* ( 1+expression );

RGB(i,j,3) = I \* (1-S);

RGB(i,j,2) = 3\*I - ( RGB(i,j,1)+RGB(i,j,3) );

elseif ( H>=2/3\*pi && H<4/3\*pi)

H = H-2\*pi/3;

RGB(i,j,1) = I \* (1-S);

expression = S\*cos( H )/(cos( pi/3-H ) + eps);

RGB(i,j,2) = I \* ( 1+ expression );

RGB(i,j,3) = 3\*I - ( RGB(i,j,1)+RGB(i,j,2) );

elseif (H>=4/3\*pi && H<=2\*pi)

H = H-4\*pi/3;

RGB(i,j,2) = I \* (1-S);

expression = S\*cos( H )/(cos( pi/3-H ) + eps);

RGB(i,j,3) = I \* ( 1+ expression );

RGB(i,j,1) = 3\*I - ( RGB(i,j,2)+RGB(i,j,3) );

end

end

end

RGB = RGB \* 255;

RGB = uint8(RGB);

end

主函数：

clc;clear;close all;

image = imread('Fig\_strawberries.tif');

image1 = myRGB2HSI(image);

image2 = zeros(size(image1));

[x,y] = size(image1(:,:,1));

for i = 1:x

for j = 1:y

image2(i,j,3) = 1 - image1(i,j,3);

image2(i,j,2) = image1(i,j,2);

if image1(i,j,1)<0.5

image2(i,j,1) = image1(i,j,1) + 0.5;

else

image2(i,j,1) = image1(i,j,1) - 0.5;

end

end

end

image3 = myHSI2RGB(image2);

figure;

subplot(1,2,1);

imshow(image);

title('original image');

subplot(1,2,2);

imshow(image3);

title('HSI complement')

实验结果：

|  |  |
| --- | --- |
|  |  |

比较RGB与HIS 的补色结果，发现HIS总是有一些偏绿，具体原因在在小结与讨论中。

**Task 3：Color slicing**

Using the following two equation to implement the color slicing on image ‘strawberries’ where n=3 denotes the Red, Green and Blue component. Set W= 0.2549, R=0.1765, a = (0.6863, 0.1608, 0.1922). Pixels outside the cube and sphere were replaced by color (0.5, 0.5, 0.5)

实验代码：

clc;clear;close all;

image = im2double(imread("Fig\_strawberries.tif"));

subplot(1,3,1);

imshow(image);

title('original image');

[x,y,z] = size(image);

W = 0.2549;

R = 0.1765;

a = [0.6863, 0.1608, 0.1922]';

image1 = zeros(size(image));

image2 = zeros(size(image));

for i = 1:x

for j = 1:y

b = image(i,j,:);

c = b(:);

if all(abs(c - a) < W/2)

image1(i,j,1) = image(i,j,1);

image1(i,j,2) = image(i,j,2);

image1(i,j,3) = image(i,j,3);

else

image1(i,j,1) = 0.5;

image1(i,j,2) = 0.5;

image1(i,j,3) = 0.5;

end

end

end

subplot(1,3,2);

imshow(image1);

title('cube slicing');

for i = 1:x

for j = 1:y

b = image(i,j,:);

c = b(:);

if dist(c',a) < R

image2(i,j,1) = image(i,j,1);

image2(i,j,2) = image(i,j,2);

image2(i,j,3) = image(i,j,3);

else

image2(i,j,1) = 0.5;

image2(i,j,2) = 0.5;

image2(i,j,3) = 0.5;

end

end

end

subplot(1,3,3);

imshow(image2);

title('sphere slicing');

实验结果：

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Task 4：Histogram Processing on Color Images**

(a) Demonstrate the histogram of its intensity component

实验代码：

clc;clear;close all;

image = imread('Fig0637(a)(caster\_stand\_original).tif');

image1 = myRGB2HSI(image);

H = image1(:,:,1);

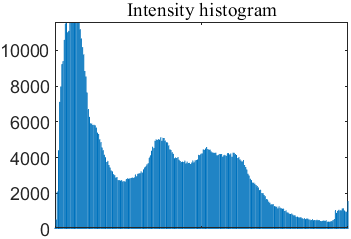
S = image1(:,:,2);

I = image1(:,:,3);

subplot(2,2,1), imshow(image), title('original image',FontName='Times New Roman');

subplot(2,2,2), imhist(I(:)), title('Intensity histogram',FontName='Times New Roman');

实验结果：



(b) Implement the Histogram Equalizing on the intensity component, without altering the hue and saturation. Comparing the result to the original image.

实验代码：

clc;clear;close all;

image = imread('Fig0637(a)(caster\_stand\_original).tif');

image1 = myRGB2HSI(image);

H = image1(:,:,1);

S = image1(:,:,2);

I = image1(:,:,3);

subplot(2,2,1), imshow(image), title('original image',FontName='Times New Roman');

subplot(2,2,2), imhist(I(:)), title('Intensity histogram',FontName='Times New Roman');

[row,col] = size(I);

R = round(I\*255);

% figure;

% imshow(R)

PMF = zeros(1, 256);

for i = 1:row

for j = 1:col

PMF(R(i,j) + 1) = PMF(R(i,j) + 1) + 1;

end

end

PMF = PMF / (row \* col);

CDF = zeros(1,256);

CDF(1) = PMF(1);

for i = 2:256

CDF(i) = CDF(i - 1) + PMF(i);

end

Sk = zeros(1,256);

for i = 1:256

Sk(i) = CDF(i) \* 255;

end

Sk = round(Sk);

for i = 1:row

for j = 1:col

R(i,j) = Sk(R(i,j) + 1);

end

end

R = R/255;

subplot(2,2,3), imshow(R), title('after Histogram Equalization',FontName='Times New Roman');

subplot(2,2,4), imhist(R(:)), title('intensity Histogram Equalization',FontName='Times New Roman');

figure;

image2 = ones(size(image));

image2(:,:,1) = H;

image2(:,:,2) = S;

image2(:,:,3) = R;

image3 = myHSI2RGB(image2);

imshow(image3);

title('Intensity Histogram Equalization',FontName='Times New Roman')

image4 = image3 - image;

figure;

imshow(image4)

title('diffrerence after Histogram Equalization',FontName='Times New Roman')

figure;

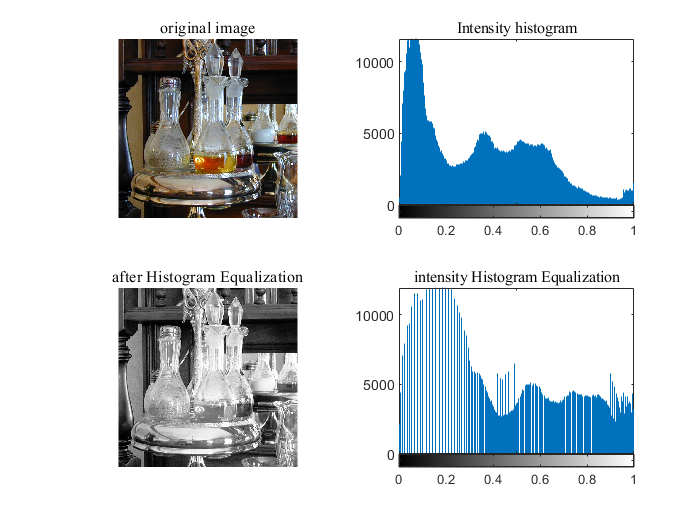
II = image4(:,:,3);

histogram(II);

title('diffrerence after Histogram Equalization',FontName='Times New Roman')

实验结果：

对其进行Intensity 直方图均衡化：



直方图均衡化的结果：



与原来图像的比较：

|  |  |
| --- | --- |
|  |  |

(c) Further increase the image's saturation component and compare the result to (b)

实验代码：在(b)的基础上加上对S加0.2提高其饱和度

image1 = myRGB2HSI(image);

H = image1(:,:,1);

S = image1(:,:,2);

I = image1(:,:,3);

S = S + 0.2;

实验结果：提高饱和度后的结果+和原来的的图像的差异对比

|  |  |
| --- | --- |
|  |  |

(d) Demonstrate the histogram of its intensity component of the processed image after (c) and compare the distribution to (a).

实验代码：在(c)的基础上加上以下部分

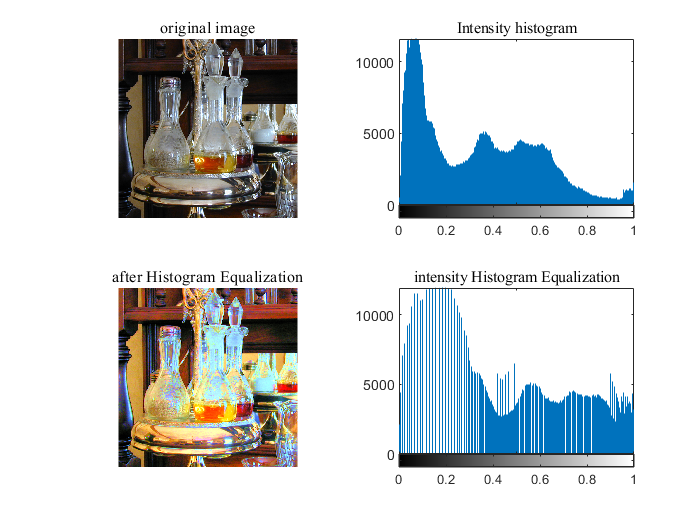
subplot(2,2,1), imshow(image), title('original image',FontName='Times New Roman');

subplot(2,2,2), imhist(I(:)), title('Intensity histogram',FontName='Times New Roman');

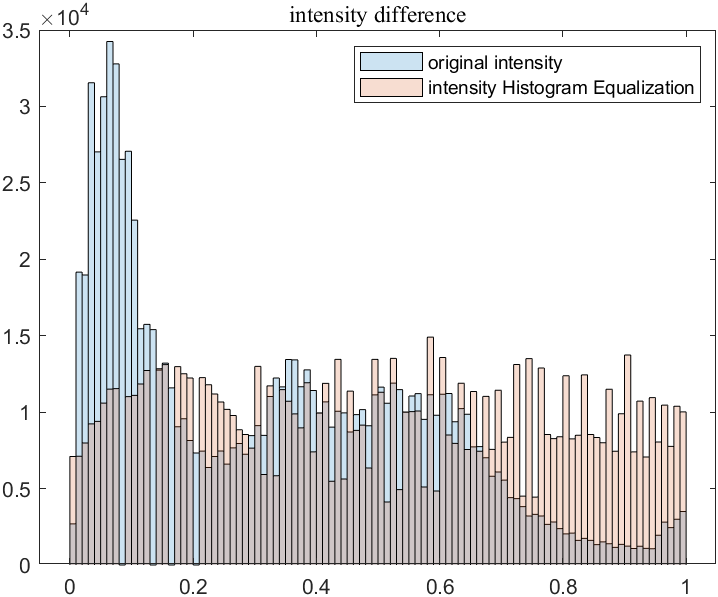
subplot(2,2,3), imshow(image3), title('after Histogram Equalization',FontName='Times New Roman');

subplot(2,2,4), imhist(R(:)), title('intensity Histogram Equalization',FontName='Times New Roman');

实验结果：



Intensity直方图均衡化后与原来的对比图：



**【小结或讨论】**

1. 在任务三中要注意，在cube slicing中，判断一个点是否在a为中心，W为边长的正方形中时，如果用一下规则判断，则得不到想要的效果，但是用sum和all的逻辑是相同的，这是比较奇怪的地方

d = abs(c - a) < W/2;

if sum(d(:)) == 0

image1(i,j,1) = image(i,j,1);

image1(i,j,2) = image(i,j,2);

image1(i,j,3) = image(i,j,3);

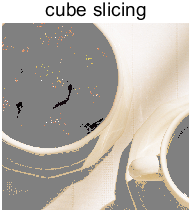
else

image1(i,j,1) = 0.5;

image1(i,j,2) = 0.5;

image1(i,j,3) = 0.5;

end

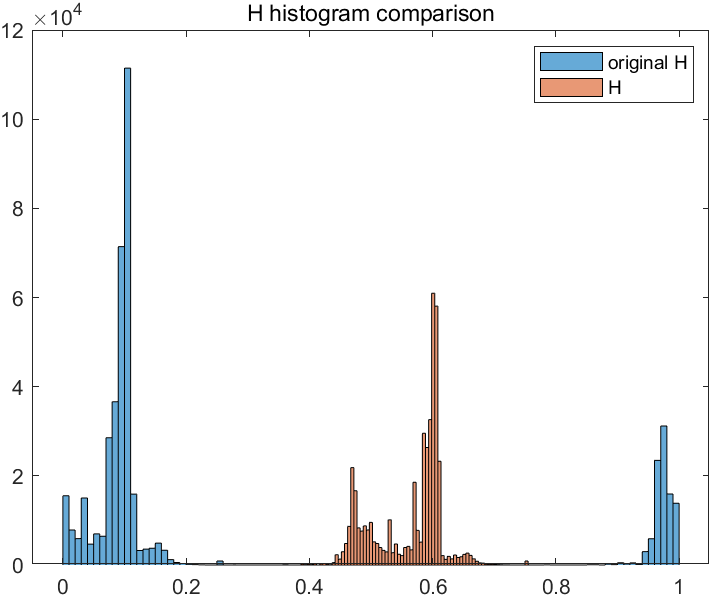


1. 在任务3中sphere slicing中要注意，算欧式距离时，第一个参数是行向量，第二个参数是列向量，才会得到正确的结果。

if dist(c',a) < R

1. 关于在color complement中，HIS模式下总是图像比较绿的原因：

显示前后H的直方图分布：



观察发现，直方图显示H可以实现补色的功能，类似的输出S与I，都可以实现

|  |  |
| --- | --- |
|  |  |

检查RGB2HSI与HSI2RGB的转换函数，输入image对其进行RGB2HSI与HSI2RGB后，观察可知，转换函数能无差的实现转换

figure;

subplot(1,2,1);

imshow(image);

title('original image')

image1 = myRGB2HSI(image);

image6 = myHSI2RGB(image1);

subplot(1,2,2);

imshow(image6);

title('transformed image')

image2 = zeros(size(image1));

|  |  |
| --- | --- |
|  |  |

且多次调试无误，但是依然无法输出正常补色后的图像