

Experiment (4)

RGB

1. Objective:

- Learn how to display colour channels of RGB (or true colour) images using for loop
- Explore image conversion using for loop.

2. Equipment: pc and MATLAB software.

3. Background:

RGB (or true colour) images are 3-D arrays that we may consider conceptually as three distinct 2-D planes, one corresponding to each of the three red (R), green (G) and blue (B) colour channels. RGB is the most common colour space used for digital image representation as it conveniently corresponds to the three primary colours which are mixed for display on a monitor or similar device.

We can easily separate and view the red, green and blue components of a true-colour image. It is important to note that the colours typically present in a real image are nearly always a blend of colour components from all three channels.

RGB to grey-scale image conversion we can convert from an RGB colour space to a grey-scale image using a simple transform. Grey-scale conversion is the initial step in many image analysis algorithms, as it essentially simplifies (i.e. reduces) the amount of information in the image. Although a grey-scale image contains less information than a colour image, the majority of important, feature related information is maintained, such as edges, regions, blobs, junctions and so on.

4. Procedure:

- Create a script file (m-file) and type the following code to separate and display each of the three red (R), green (G) and blue (B) colour channels.

```
clear;clc; %clear workspace and screen
m=imread('image.jpg'); % read image
figure(1),imshow(m); % display image
c=imfinfo('image.jpg') %store image information in c
y=c.Width
x=c.Height
m=double(m); %convert image data class into double

for i=1:1:x-1 %loop for red channel
    for j=1:1:y-1
        m1(i,j,1)=m(i,j,1);
        m1(i,j,2)=0;
        m1(i,j,3)=0;
    end
end
m1=uint8(m1); %convert image data class into uint8
figure(2),imshow(m1) % display red image

for i=1:1:x-1 %loop for green channel
    for j=1:1:y-1
        m2(i,j,1)=0;
        m2(i,j,2)=m(i,j,2);
        m2(i,j,3)=0;
    end
end
m2=uint8(m2);
figure(3),imshow(m2)
```

```

for i=1:1:x-1                %loop for blue channel
    for j=1:1:y-1
        m3(i,j,1)=0;
        m3(i,j,2)=0;
        m3(i,j,3)=m(i,j,3);
    end
end
m3=uint8(m3);
figure(4),imshow(m3)

%display all images in one figure using subplot function.

```

- Create a script file (m-file) and type the following code to convert RGB to grey-scale image

```

clear;clc;                    %clear workspace and screen
m=imread('image.jpg');       % read image
figure(1),imshow(m);         % display image
c=iminfo('image.jpg')        %store image information in c
y=c.Width
x=c.Height
m=double(m);                 %convert image data class into double

for i=1:1:x-1                %loop for red channel
    for j=1:1:y-1
        r(i,j)=(m(i,j,1)+ m(I,j,2)+ m(I,j,3))/3;
    end
end
r= uint8(r);                  %convert image class to uint8 before display
figure(2),imshow(r)

% use subplot function to display both image before and after conversion.

```

5. Discussion:

1. Why using double command?
2. What the difference between uint8 and double command?