International Journal of Image Processing and Vision Science

Volume 1 | Issue 3 Article 8

January 2013

GRAY COLOR CONVERSION USING CLASSIFICATION METHOD

ASHOK KUMAR PATEL
Dept. of Electrical Engineering NIT Raipur, India, ashokbecse2006@yahoo.com

SANJAY KUMAR
Dept. of Electrical Engineering NIT Raipur, India, skumar.it@nitrr.ac.in

GRAY COLOR CONVERSION USING CLASSIFICATION METHOD

ASHOK KUMAR PATEL[1] & SANJAY KUMAR[2]

[1,2]Dept. of Electrical Engineering NIT Raipur, India
E-mail : ashokbecse2006@yahoo.com, skumar.it@nitrr.ac.in

Abstract - There are various methods of gray conversion of RGB color images. They are calculated by three functions, Grayscale function, the Desaturate function, and the value component of the HSV option of the Decompose function. By converting this method we found the gray color image but when trying the reverse it is not same as actual images. That is due to the function that produces same value for different color and when reversing there is conflict between choosing one of them. Here I want to introduce a function using Classification method that produces different values of gray color for a range of RGB color images so the nearly reverse is possible.

Keywords - *Grayscale function, Desaturate function, Decompose function, Classification Method.*

I. INTRODUCTION

When any RGB image is converted into gray color using the Grayscale function then they are calculated with the formula of some multiplication then it may possible that two of them with different color have the same value.

Luminance = 0.30*r+0.59*g+0.11*b

The result is denoted in integer only between 0 to 255 then 0.1 to 1 may considered as 1 similarly 254.1 to 255 may considered as 255. RGB(40,200,0) is equal to RGB(60,160,160) is equal to 130.

Similarly the removal of the saturation from an image should also produces a grayscale. One way to desaturate an image is to replace the RGB value for each pixel with that of the closest point on the neutral axis.

Lightness = {max(R,G,B)+min (R,G,B)}/2

The result is denoted in integer only between 0 to 255. So the result of two RGB may be same for example RGB(255,0,0), RGB(0,255,0), RGB(0,0,255) have same value equal to 128.

Finally, it is possible to get a slightly different grayscale conversion by using the HSV option of the Decompose function. This decomposes the original image into three new images, each an 8-bit grayscale image representing the hue, saturation, and value components of the image. The value component is a conversion to grayscale that is based on moving to the neutral axis by selecting the maximum RGB component.

V = max(R,G,B)

The result is already in integer and between 0 to 255. The result may be same for more than two different color for example RGB(0,255,0) and RGB(0,0, 255) and RGB(0,255,255) have same value 255.

So grayscale images produce by these function is not reversible. Here I introduce a new algorithm that converts the RGB color images into 256 colors and there corresponding value is used for grayscale images. They are reversible but with some variation.

II. PROPOSED ALGORITHM:

Here I classified the RGB color on 256 classes on the basis of their effect in producing color so we choose 64 values of Blue color and 32 values of Red and Green in one class. So they produce 8*8*4=256 classes.

The algorithm is based on the following statement

```
If Red is between 0 to 31
    If Green is between 0 to 31
        If Blue is between 0 to 63
            Val=0;
        Else if Blue is between 64 to 128
            Val=1;
            :
            :
            Endif
        Else if Green is between 32 to 63
            If Blue is between 0 to 63
                Val=4;
                :
                :
            Endif
        Endif
    Endif
```

The conversion is implemented using following function

```
        x=((a-rem(a,32))/32);
        y=((b-rem(b,32))/32);
        z=((c-rem(c,64))/64);
        val=x*32+y*4+z;
where   a=red(image);
        b=green(image);
        c=blue(image);
```

and val is a matrix with same size of image and represent the proposed gray converted image.

The inverse conversion is implemented using following function

```
k=(val-rem(val,32))/32;
J1=k*32;
n=rem(val,4);
J3=n*64;
 m=(val-k*32-n)/4;
 J2=m*32;
```

Where val is a matrix of grayscale image

```
J1=red(image);
J2=green(image);
J3=blue(image);
```

III. EXPERIMENT AND RESULT:

I did this conversion on some images of car and inverse them while the result is some satisfactory. Fig1(a,b) show the original image and Fig2(a,b) show converted gray images while Fig3(a,b) show the inverse conversion to RGB. From this result we can say that we can achieve nearly similar image using this function.

.

Fig. 1(a,b) Original

Fig. 2(a,b) Gray converted

Fig. 3(a,b) Inverse into RGB

Here we can also compare gray converted images using different function. Fig 4 shows the image converted using grayscale and Fig 5 shows the image converted using desaturation function. They are little similar images. Fig 6 shows the image converted using decompose function and Fig 7 shows the image converted using our proposed method and they are little similar.

Fig. 4 : Grayscale function

Fig. 5 : Desaturation function

Fig. 6 : Decompose function

Fig. 7 : Proposed method

IV. CONCLUSION:

Here we see that this image is nearly similar to another method that is decomposition function of HSV value. Using this method we can compress the images like other gray scale conversion and can be used for transmission of images and can achieve nearly color images. Here we use the first value of range but we can also use random on that range.

REFERENCES:

[1] "Grokking the GIMP – Learning Advance Image Editing Techniques" by Carey Bunks http://drpeterjones.com/colorcalc/

◈◈◈