Coloring Grayscale Image using Reference Image

Eginata Kasan / 13517030
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): eginatakasan@gmail.com

***Abstract*—This paper explores the method of coloring grayscale images using a reference image by comparing and matching their luminance. This is done by converting both images into YCbCr color space. This method is proven to be effective yet results in imperfect coloring for some images. The issue lies in choosing the right pair of target and reference image to get the best results. Another recommended improvement is the use of swatches for selective color transfers.**

Keywords—image processing; luminance comparison; color mapping;

I. Introduction

Grayscale images are images that consists of colors between black or white. It is a representation of amount of light. The first images captured were in grayscale, before in 1907, autochrome, a method of capturing colored images, was introduced.

Coloring grayscale images can also be done manually. One would first group objects based on their darkness and then pick their suitable colors. The colors are chosen using colors either based on references on what it should look like or simply based on their preference. The manual method of coloring the images can be slow, although the principle can be used as a base for a faster automatic method. In this paper, automatic coloring of grayscale images is explored based on luminance comparison with a reference image.

This paper is meant as an exploration towards a method for coloring grayscale images. It should be useful for anyone who wants a quick and simple coloring using colors from similar images.

II. Theory

A. YCbCr Color Space

A YCbCr color space is a color space, like RGB, consisting of luminance (Y) and blue and red component compared to the chroma component (Cb and Cr) [2]. The YCbCr color space is commonly used in video or digital photography system. The Y component (luma) of YCbCr is essentially a grayscale version of a colored image when separated from Cb and Cr component.

YCbCr color space is used for coloring grayscale images because it separates luminance from the colors. RGB color space does not have luminance in their components, when a color appears contrast the three components also have high value. Bright purple in RGB means that there is high value of blue, but also high value of red and green as well.

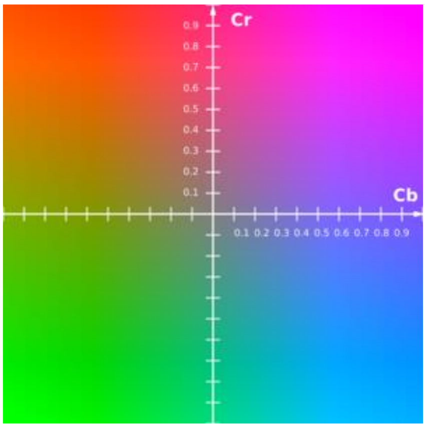

Fig. 1. Example of YCbCr color space at Y = 0.5

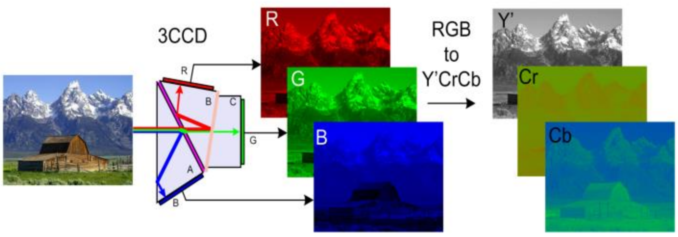

Fig. 2. An example of image to RGB to YCbCr conversion

B. Luminance

Luminance is a photometric measure of the luminous intensity per unit area of light travelling in a given direction [3]. Luminance can also be described as brightness, how much light is travelling through, reflected by, or emitted by an object. An image with only luminance displayed will be a grayscale copy of the image.

III. Related Works

There are a lot of methods proposed for coloring grayscale images. Reinhard, Ashikmin, Goosh and Shirley proposed a color transfer using matching of YCbCr color between two colored images for correction of colors in 2001 [1]. Tomihisa

Welsh, Michael Ashikhmin, and Klaus Mueller proposed using the same method to color grayscale images in [3]. Jeny Rajan uploaded a matlab program using the same idea to the mathworks file exchange [2]. There are also methods not including reference images. Huiwen Chang, et al. (2015) used a color pallete instead of an image reference and image segmentation method [6].

IV. METHODOLOGY

The idea of coloring grayscale image is to find which pixel in the reference image is most similar to the pixel in target image.

Below are the steps that are used to color images using a reference image:

1. Load 2 input images: target image and reference image
2. If target image is not yet a grayscale image, convert target image into grayscale.
3. Convert both target image and reference image into YCbCr color space.
4. For each of the pixels in the target image, find the most similar pixel in the reference image (minimum of difference) to the target pixel.
5. Color the target pixel using the color of the most similar pixel in the reference image.
6. Convert target image back to RGB.
7. Result is a colored RGB target image.

A. *Specifications*

The source code below is a program written in MATLAB. It requires the Image Processing Toolbox that can be installed when prompted by the MATLAB IDE.

The program requires 2 input strings, the path to the target image and the path to the reference image. It will output 3 figures: the original grayscale image, the reference image, and colored image as a result.

B. *Source Code*

Below are the source codes for transforming grayscale images into colored images using luminance matching.

The code below is for loading the input images and getting their size.

```
% load the target image and the reference
image
imt=imread(img1);
imref=imread(img2);
```

Convert target image to grayscale if it is not grayscale and throw an error if the reference image is not colored.

```
% get width, height, and number of colors
in the images
[sx sy sz]=size(imt);
[tx ty tz]=size(imref);

% if target image not grayscale, convert
to grayscale first
if sz~=1
    A = zeros(sx,sy);
    for x = 1:sx
        for y = 1:sy
            A = imt(:,:,1) * 0.299 +
imt(:,:,2) * 0.587 + imt(:,:,3) * 0.144;
        end
    end
    imt = A;
end
% error if reference image is not colored
if tz~=3
    disp ('img2 must be a color image
(not indexed)');
```

Turn the grayscale image into a 3x3 matrix for later conversion to YCbCr color space.

```
% turn imt into 3x3 matrix
imt(:,:,2) = imt(:,:,1);
imt(:,:,3) = imt(:,:,1);
```

Convert the images into YCbCr color space.

```
% Convert to ycbcr color space
nspace1=rgb2ycbcr(imref);
nspace2= rgb2ycbcr(imt);
```

Normalize the reference and target image. Map the pixel values from (min,max) to (0,255).

```
% Normalize the matrix
ms=double(nspace1);
mt=double(nspace2);

max_val_ref=max(max(ms));
min_val_ref=min(min(ms));
max_val_t=max(max(mt));
min_val_t=min(min(mt));

d_ref=max_val_ref-min_val_ref;
d_target=max_val_t-min_val_t;

dx_ref=ms;
dx_target=mt;
dx_ref=(dx_ref*255)./(255-d_ref);
dx_target=(dx_target*255)./(255 -
d_target);

[mx,my,~]=size(dx_target);
```

For each pixel in the target image, find the value of a pixel in the reference image which has the smallest difference of between them. Put the value to the result image matrix.

```
% Find most similar pixel in reference
image
for i=1:mx
  for j=1:my
    iy=dx_target(i,j);
    tmp=abs(dx_ref-iy);
    ck=min(min(tmp));
    [r,c] = find(tmp==ck);
    ck=isempty(r);

     if (ck~=1)
      result(i,j,2)=nspace1(r(1),c(1),2);
      result(i,j,3)=nspace1(r(1),c(1),3);
      result(i,j,1)=nspace2(i,j,1);
     end
  end
end
```

Convert the result matrix, whixh is in YCbCr color space to RGB color space for display.

```
% convert result to rgb for display
rslt=ycbcr2rgb(result);
```

Display 3 figures at the end of the program: the initial grayscale image, the reference image, and the result image

```
% Show input images and RGB result
figure,imshow(uint8(imt));
figure,imshow(uint8(imref));
figure,imshow(uint8(rslt));
```

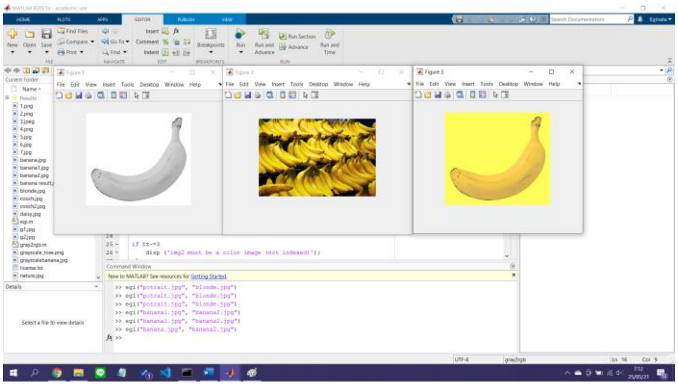

Fig. 3. Screenshot of the Matlab Program in action on Windows

V. RESULTS

The results section provides some colored images compared to the original or grayscale images as result. It is also completed with analysis of the result.

(a)

(b)

(c)

Fig. 4. (a) Grayscale image of a waterfall (b) Reference image of a different waterfall (d) Colored result

Figure 4 above shows the input and results of coloring a grayscale waterfall image. The resulting image appears to consist of mostly green, including the body of water, even though blue colored water appears in the reference image. However, the resulting waterfall image does not appear out of place, as some natural body of water can appear green.

Figure 5 shows a result of coloring an image of a green scenery. The result in figure 5(c) shows an almost accurate coloring, with exception of a blueish tint at the grass caused by the blue color of the lake in the reference image. However, the trees and the sky coloring are accurate recreations of how they are in reality.


(a)


(b)


(c)

Fig. 5. (a) Grayscale image of a green plains (b) Reference image of a lake surrounded by greeneries (c) Colored result

The results on the right (figure 6) are from using images of the sea. In this example, the input used is a colored image of the sea. This is done to compare the result and how the result is supposed to look like.

As both sample and reference image has only one hue of color, the resulting image seems normal although has a different tint of blue compared to the original image. The colored result however does not appear strange or different from reality.


(a)


(b)


(c)


(d)

Fig. 6. (a) Original colored image of the sea (b) Image transformed into grayscale as target image (c) Reference image of the sea (d) Colored result

Images below here are the images in process using a picture of a colored rose as a sample. The colored rose is then turned into grayscale before processing. Comparison between the original image in figure 7(a) and the resulting image in 7(d) shows that the coloring is not perfect as the stem of the rose has a hue of red instead of green.


(a)


(b)


(c)


(d)

Fig. 7. (a) Original colored image of rose (b) Rose transformed into grayscale as target image (c) Reference image (d) Colored result


(a)


(b)


(c)


(d)

Fig. 8. The same rose images from figure 7 in reversed position (a) Original rose (b) Grayscale version of rose (c) Reference rose (d) Colored result

Figure 8 shows the images of rose in figure 7 in reverse position. The result is much less accurate to the intended original image than in figure 8(a).

This is caused by the reference image not providing diverse enough colors for the target image to take from. For example, the bright green background of the target image is not provided in the reference 8(b).

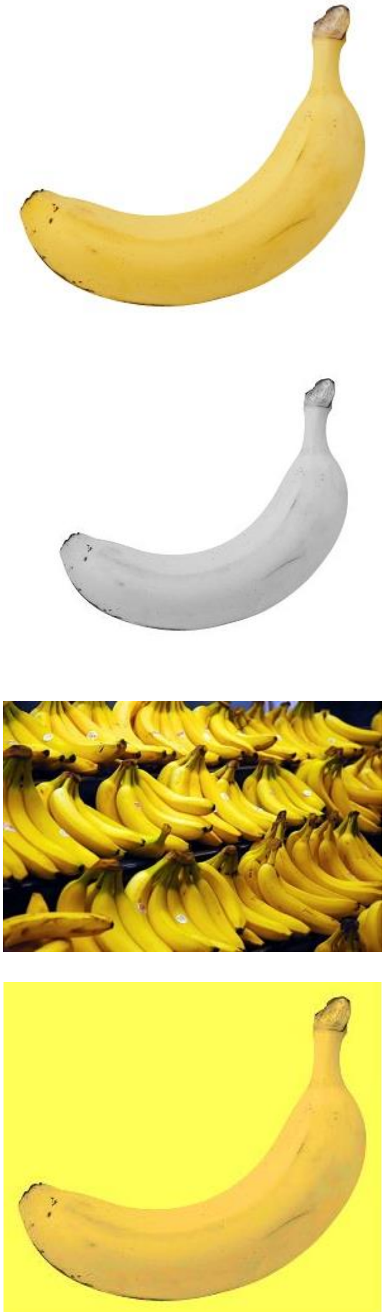

Fig. 9. Another result of the program using bananas

Figure 9 shows a result of coloring a grayscale banana. The resulting image appears the same except for the background. This is caused by the normalization before matching. Thus, normalization makes white (255) equals to the highest value at the reference image.

Fig. 10. Result of the program using images of apples

Figure 10 shows a result of coloring a grayscale apple. The result is similar to the banana in figure 9, which has a tint of red in place of the white in the background.

Fig. 11. Example of less ideal pairing of target and reference image

Figure 10 shows the process of coloring using a lesser ideal pair of target and reference image. The resulting image is made of a singular hue. This happens when the reference image does not have enough variety of colors for the target image to choose from or the target image has a narrow distribution of luminance.

VI. Conclusion

The coloring of grayscale image using only luminance comparison and pixel mapping proves to be simple yet not able to result perfectly colored images. This is true especially for images consisting of multiple colors as different colors may have the same luminance. Figure 7(d) acts the most apparent example of an imperfect result. Using swatches on top of luminance matching can be used to improve the coloring of grayscale images as stated by Welsh, et al [3]. Swatches are regions that are specified by the user which limits the colors that are available as a reference. This means users can selectively transfer colors between the target and reference swatches. Figure 10 below shows the example use of swatches.

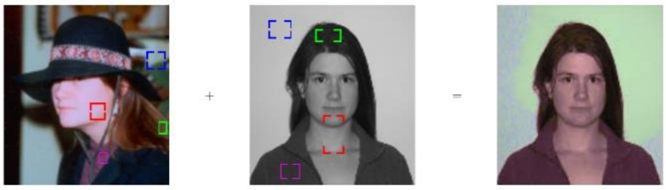

Fig. 12. Example of coloring selectively using swatches (taken from Welsh, et al.'s Transferring Color to Greyscale Images[3])

Another issue is finding perfect reference images which need to have similar distribution of luminance. If not enough variety of brightness and colors are given in the reference image, then the colors in the result would appear too similar. This is demonstrated and can be seen in figure 8 and 10.

Overall, the proposed method can effectively color grayscale images given the right reference image. The idea behind the method is simple and easy to understand.

Video Link at YouTube

Below is a link to a short demonstration video (1 minute) of coloring a grayscale rose. The video is in English, showing the process and results of coloring the grayscale image.

https://youtu.be/iV1pJbyvbBc

The images of roses are picked for their low image resolutions which means faster processing time.

Acknowledgment

First of all, I'd like to thank God for bringing me hope and leading me to where I am now. For his grace and compassion makes me able to stand through every obstacle in life.

Thank you to Mr. Rinaldi Munir for his passionate teaching. Thank you for bringing us theories and skills on Image Processing. Your apparent love for images and videos shines which makes me appreciate details of them more.

Thank you for Jeny Rajan, the author of my reference source code, for sharing your simple yet effective program for the public to learn from.

I'd like to thank my mother, my sisters, and my beloved cousins for loving and supporting me even before and after writing this paper. No matter if I am far or close, I can feel the family warmth which makes me feel like I am never alone.

REFERENCES

[1] Reinhard, E., Ashikmin, M.,Gooch, B. and Shirley, P. 2001. Color Transfer between Images, IEEE Computer Graphics and Applications, September/October2001, 34-40

[2] Jeny Rajan (2021). Gray image to Color image conversion (https://www.mathworks.com/matlabcentral/fileexchange/8214-gray-image-to-color-image-conversion), MATLAB Central File Exchange. Retrieved May 23, 2021.

[3] Welsh, Tomihisa & Ashikhmin, Michael & Mueller, Klaus. (2002). Transferring Color to Greyscale Images. ACM Trans. Graph.. 21. 277-280. 10.1145/566570.566576.

[4] Danoja Dias. 2017. What is YCbCr ? (Color Spaces). Retrieved from: https://medium.com/breaktheloop/what-is-ycbcr-964fde85eeb3 at 23 May 2021 16:55

[5] Wikipedia. (n.a.). Luminance. Retreived from https://en.wikipedia.org/wiki/Luminance at 23 May 2021 17:00

[6] Huiwen Chang, Ohad Fried, Yiming Liu, Stephen DiVerdi, and Adam Finkelstein. 2015. Palette-based Photo Recoloring.

[7] Rinaldi Munir. 2021. Slide Presentasi Kuliah IF4073 Interpretasi dan Pengolahan Citra.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 April 2021

Eginata Kasan / 13517030