INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Department of Electronics & Electrical Communication

M.Tech First year

Visual Information and Embedded System Engineering (VIPES)

EC69505 - IMAGE PROCESSING LABORATORY

A report on Mini Project

Color Selection and Editing for Palette-Based Photo Recoloring

Submitted By:
Arnab Biswas (21EC65R01)
Mohit Gupta (21EC65R17)

CONTENTS

INTRODUCTION

Color manipulation is a key process in photo enhancement, and professional image editing suites incorporate an array of tools to support it. A palette, also known as a colour lookup table (CLUT), is a correspondence table in computer graphics that assigns an index to selected colours from a colour space's colour reproduction spectrum. The word 'colour palette,' sometimes known as 'palette,' refers to the whole range of colours that a digital device, such as a camera LCD or a computer monitor, can render or the range of colours contained inside a given image. A computer monitor's colour palette can be as large as 16 million colours, but an image usually has a considerably smaller amount, usually 256.

Depending on the demands of the user, a palette, or colour palette, may display the complete range of RGB colours, or shades of a certain colour, and anything in between. When colour matching during post-processing, palettes can be built automatically to satisfy user needs. The color of an image can be easily modified with the help of a color palette.

Palette-based photo recoloring is used to extract palette colors from an image and edit the image colors. It provides an easy way to change the appearance of an image by simply modifying the color palette. The basic steps involved in this process are that first the palette of the image is obtained and then the particular color is modified by looking at its color palette value and the modified color combination is transferred to the original image.

(a) (b)

Figure 1: In the above images (a) part is the original image (b) part is the modified image after recolorisation process

ALGORITHM

1. Colour Transfer:

Colour transfer is done by using the following formula:

$$\sum_j^k w_{ij}\mathbf{e}_j / \sum_j^k w_{ij}$$

where, $\mathbf{e}_j = \mathbf{c}'_j - \mathbf{c}_j$ and $\mathbf{c}'_j$ & $\mathbf{c}_j$ are the desired value of the colour and previous value of the colour or $\mathbf{c}_j$ is the colour which we want to change and also, $w_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{c}_j\|}{2\sigma_r^2}\right)$,

where $\mathbf{x}_i$ is the input of the colour and σ_r is chosen as the mean distance between all pairs of the colour in the original pallet.

So, to get the value of σ_r, we use the following algorithm

START:
Set a map//this is used to get the pallet colour value
For i=1 to image rows
For j=1 to image columns
Get pixel_value
If pixel_value is not stored in map add a new data in map
Else update the frequency of pixel value by one
Endfor
Set pair of vector to 0
For i=1 to size of map
Store the data from map to vector
Endfor
Sort the vector by frequency
Set vector to 0
For i=1 to size of vector
Only store the hightest frequency value from old vector to new vector
Endfor
For i=1 to 5
Find distance using the formula sqrt(x_1*x_1-y_1*y_1)
sum=sum+distance
Endfor
mean=sum/5
Return mean
END

Next to find the value of $\sum_j^k w_{ij}\mathbf{e}_j / \sum_j^k w_{ij}$, we use the following algorithm,
START:
Define d//this variable takes the input so that we can change the colour of image by user-defined colour
Define c_j_next//to store the next pixel value
Store the value of c_j_next according to d
For i=1 to image rows

For j=1 to image columns
Get corresponding pixel value of image
If pixel value is in the range of desired pixel value
Set e_j=desired pixel value-initial pixel value
Typecast the distance value from integer to float
For k=1 to 5
Set exp_numa=pixel value-previous pallet value
Set exp_numa=(-1)*exp_numa
Set w_i_j=exponential of exp_numa/2*sigma
Find sum of expression of numerator
Find sum of expression of denominator
Endfor
Set the new pixel value=previous pixel value+(numerator sum/denominator sum)
Endfor
END

After transferring the colour I have done the gamma conversion of the shading part. The gamma conversion is done using the following formula:

$$O = \left(\frac{I}{255}\right)^{\frac{1}{\gamma}} \cdot 255$$

Where, O is the output pixel value, I is the input pixel value and γ is the gamma factor. Gamma factor controls image brightness. If gamma < 1 then the image will be darker if gamma > 1 then the image will be lighter. A gamma = 1 has no effect. An image pixel values are converted from the range [0, 255] to [0, 1.0]. After calculations, values are converted back to the range [0, 255].

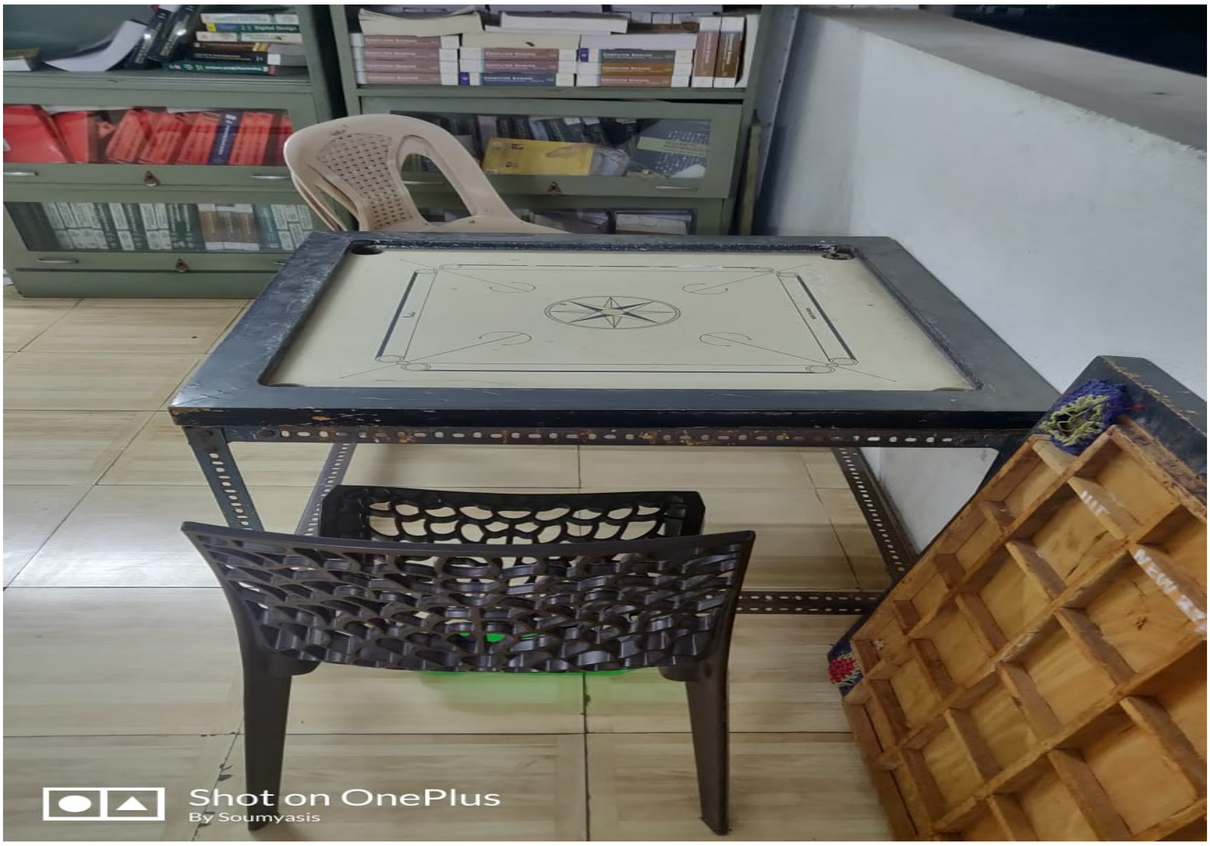
Gamma correction can be implemented using a lookup table (LUT). It maps the input pixel values to the output values. For each pixel value in the range [0, 255] is calculated corresponding gamma-corrected value. We have done the gamma conversion by the following algorithm:

```
START
Set invgamma
invgamma=1/gamma
Store mat table
Set a pointer p to iterate
For i=1 to 256
p[i]=(pow(i / 255.0, invgamma) * 255)
endfor
Use lookup table to change input pixel to output pixel
END
```

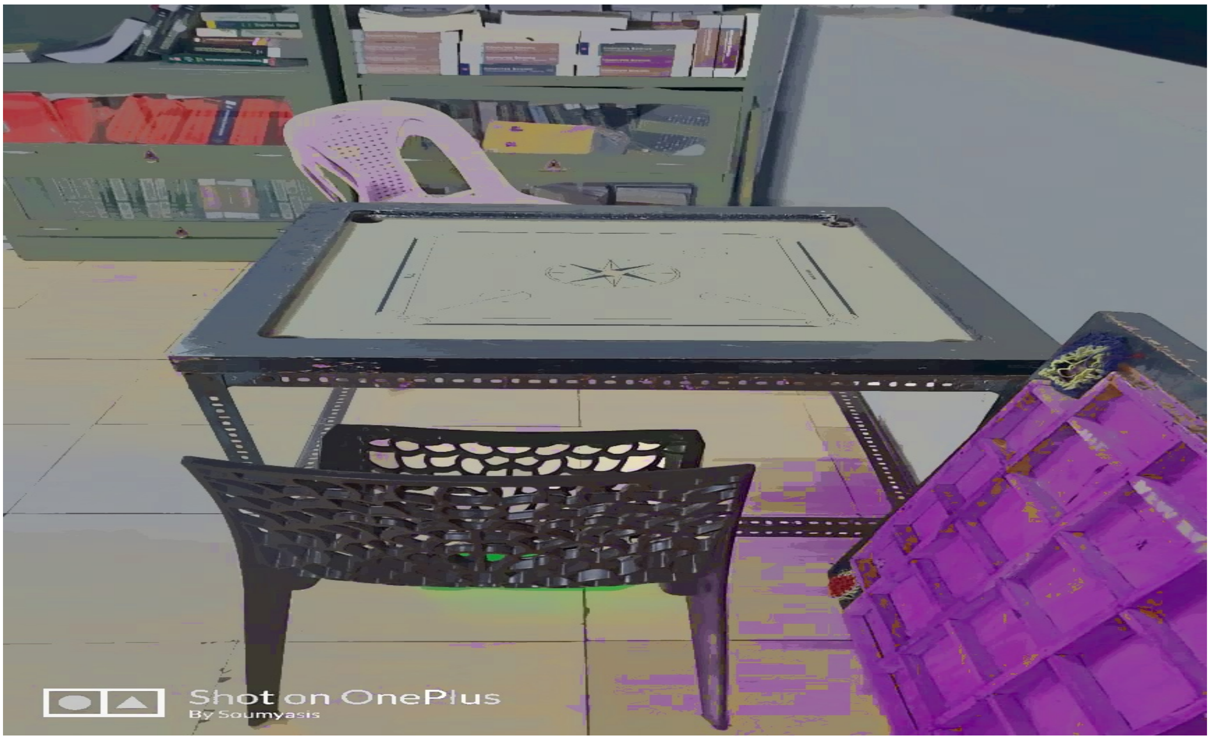
After changing the reflectance part and the shading part we have done the multiplication to get the original image by the following algorithm:

```
START:
Set vector to 0//for storing the shading pixel value
For i=1 to image rows
For j=1 to image columns
Store (shading pixel value/255) in vector
Endfor
For i=1 to image rows
For j=1 to image columns
Output image pixel=reflectance image pixel*value of vector
Endfor
END.
```

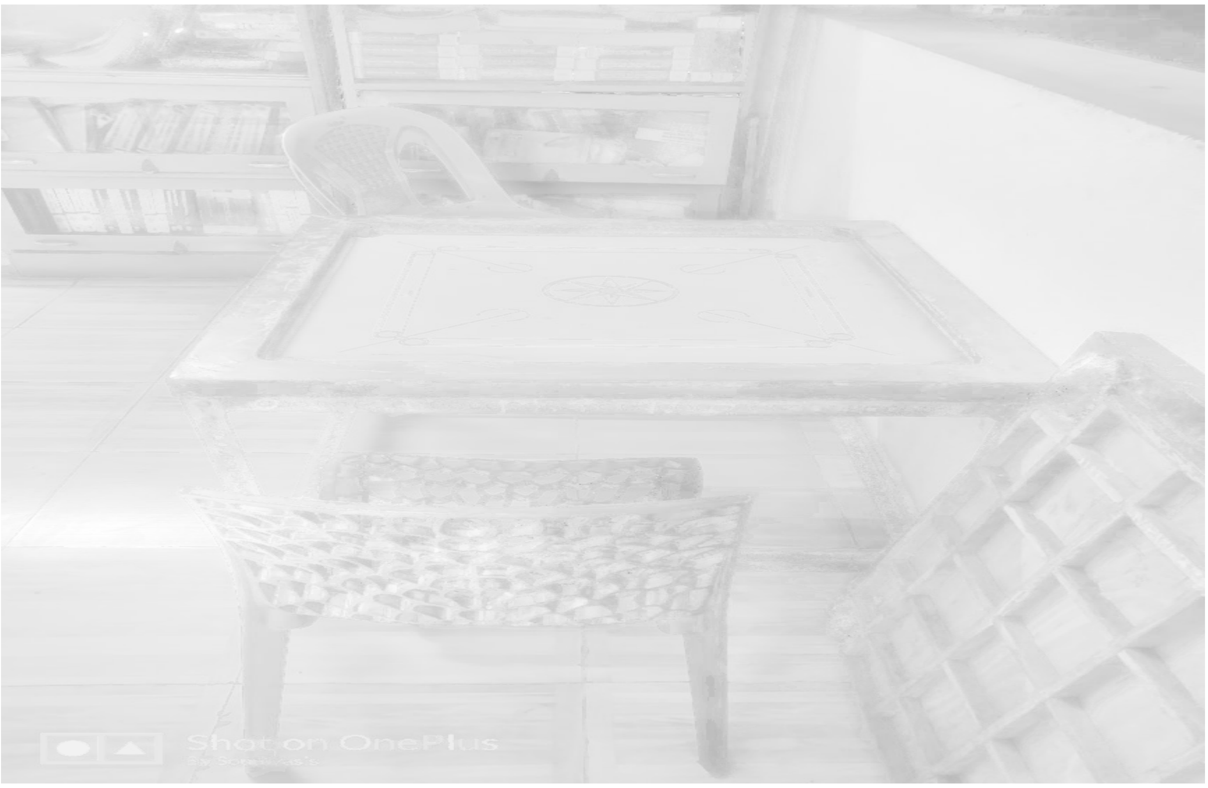
Output Results:



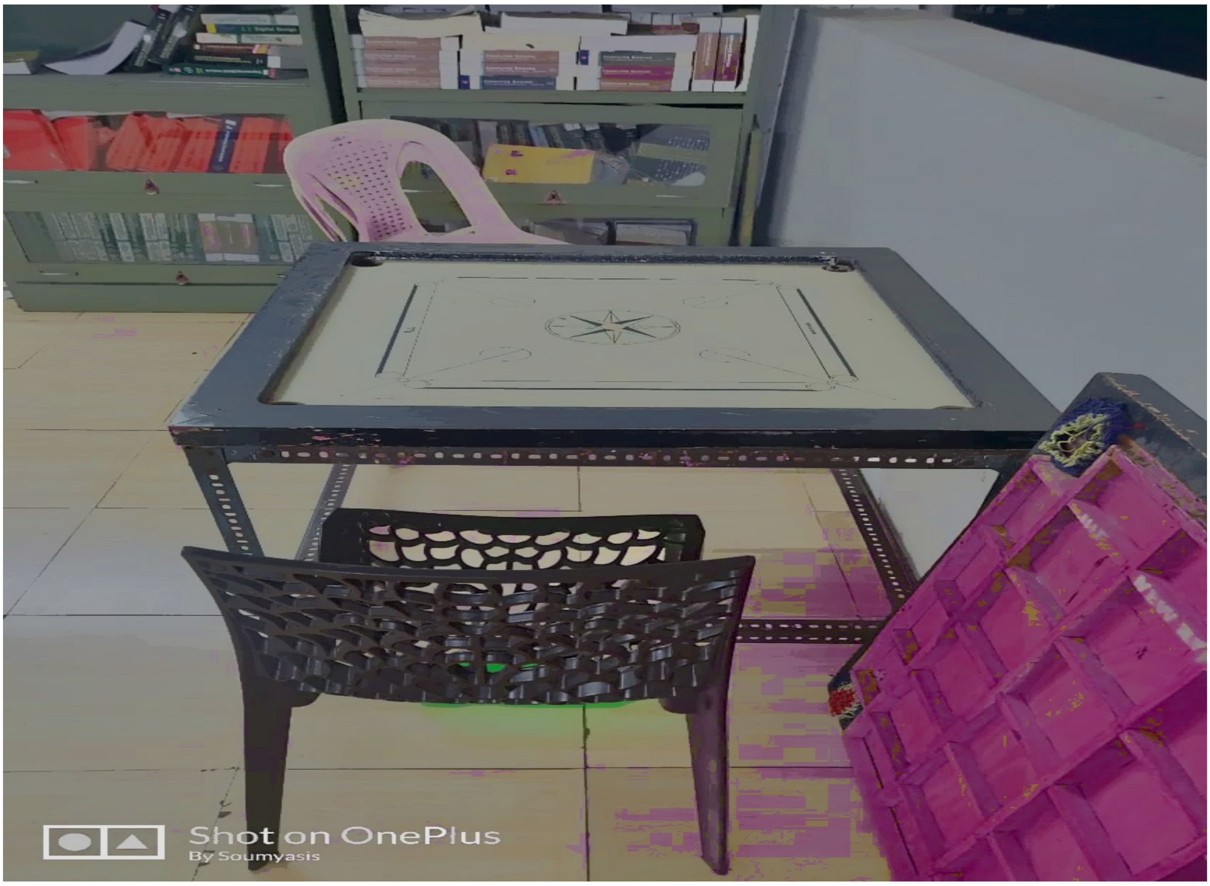
Original input image



Reflectance Image after Colour Transfer



Shading image after Gamma Conversion



Original Image after Colour Transfer and Gamma Conversion

DISCUSSION:

- The problem of luminance present in Cheng's paper has been eliminated by using the concept of Intrinsic Decomposition and the Gamma Correction. In intrinsic decomposition, we only modify the reflectance part.

- With the help of the intrinsic decomposition process, the given image is divided into two parts i.e. Reflectance and Shading images.

$$I = R * S$$

R = Reflectance Image
S = Shading Image

- The overall brightness of an image is controlled via gamma correction. Images that haven't been properly adjusted may appear bleached out or excessively dark. In gamma correction, we modify the shading image.

$$V_{out} = A\, V^{\gamma}$$

- To change any particular color from the palette which is generated with the help of the K-means algorithm we need to change it manually in the code or else done by creating a Graphical User Interface (GUI).

- Finally, to get the modified image we simply multiply the updated reflectance and shading images. First, we normalize the pixel value of the shading image and then perform the multiplication to get back the modified image.

- Similar work can also be done by changing the image to HSI and then modifying the hue values and again converting the given image to RGB to get the modified image.

REFERENCES:

- Chang, H., Fried, O., Liu, Y., DiVerdi, S., & Finkelstein, A. (2015). Palette-based photo recoloring. *ACM Transactions on Graphics*, *34*(4), 1–11. https://doi.org/10.1145/2766978

- Sean Bell, Kavita Bala, and Noah Snavely, "Intrinsic images in the wild," ACM Trans. on Graphics (SIGGRAPH), vol. 33, no. 4, 2014.

- Chang, H., Fried, O., Liu, Y., DiVerdi, S., & Finkelstein, A. (2015b). Palette-based photo recoloring. *ACM Transactions on Graphics*, *34*(4), 1–11. https://doi.org/10.1145/2766978

- Zhang, Q., Xiao, C., Sun, H., & Tang, F. (2017). Palette-Based Image Recoloring Using Color Decomposition Optimization. *IEEE Transactions on Image Processing*, *26*(4), 1952–1964. https://doi.org/10.1109/tip.2017.2671779