EECE-7315 **Digital Image Processing** **SPRING-2015**

Color Image enhancement using non-linear transfer fucntions

Yuhao Huang,MSEE

April 28, 2015

Department of Electrical and Computer Engineering

Northeastern University

ABSTRACT

In this paper, we implement an image enhancement method proposed by Deepak Ghimire and Joonwhoan Lee[1]. In this method, the enhancement is divided into two processes: 1. Luminance enhancement, and 2. Contrast enhancement. The enhancement is applied on the V component. The luminance enhancement is achieved using a specifically designed non-linear transfer function. The contrast enhancement transforms the pixel's using a Gaussian kernel. We use histogram equalization and a method called INDANE to compare with the method we use in the paper. The result shows that the method in this paper produces visually better image than the other two methods.

Contents

I. Introduction

Image clarity is very easily affected by lighting, weather, or equipment that has been used to capture the image. These conditions lead to image may suffer from loss of information. Therefore, some image enhancement techniques are developed for recovering the information in an image. Color images provide more and richer information for visual perception than that of the gray images. Color image enhancement plays an important role in Digital Image Processing. The principle objectives of image enhancement techniques is to process an image so that the result is more suitable than the original image for a specific application. The enhancement methods can broadly be divided in to the following two categories:1. Spatial Domain Methods 2. Frequency Domain Methods. In spatial domain techniques, we directly deal with the image pixels. The pixel values are manipulated to achieve desired enhancement. It can be denoted by the expression, $g(x, y) = T[f(x, y)]$. In frequency domain methods, the image is first transferred in to frequency domain. It means that, the DFT, DCT or other transforms of the image is computed first. All the enhancement operations are performed on the transform of the image and then the inverse transform is performed to get the resultant image.

Many techniques have been developed to enhance images,such as histogram equalization, homomorphic filtering[2],retinex[3],INDANE(Integrated Neighborhood Dependent Approach for Nonlinear Enhancement)[4],etc. Histogram Equalization (HE), is a technique that made contrast adjustment using images' histogram. This technique is based on the idea of remapping the histogram of the scene to a histogram that has a near-uniform probability density function. Homomorphic Filtering, is sometimes used for image enhancement. It simultaneously normalized the brightness across an image and increases the contrast. Here, Homomorphic Filtering is used to remove multiplicative noise.Since illumination and reflectance are combined multiplicatively, the component are made additive by taking the logarithm of the image intensity, so that these linearly in the frequency domain. Illumination variations can be thought of as a multiplicative noise, and can be reduced by filtering in the log domain.

Retinex theory was first proposed by Edwin Land in 1964. There are many algorithms based on Retinex theory such as Single Scale Retinex, Multi Scale Retinex (MSR), Multi Scale Retinex with modified color restoration (MSRCR)[5], Fast Multi Scale Retinex (FMSR)[6], etc. INDANE is an image enhancement method proposed by Li Tao and V.K.Asari [4]. They applied it to the gray component of the color image and used a nonlinear function for luminance enhancement.

The general process of the method we're implementing is as follows:

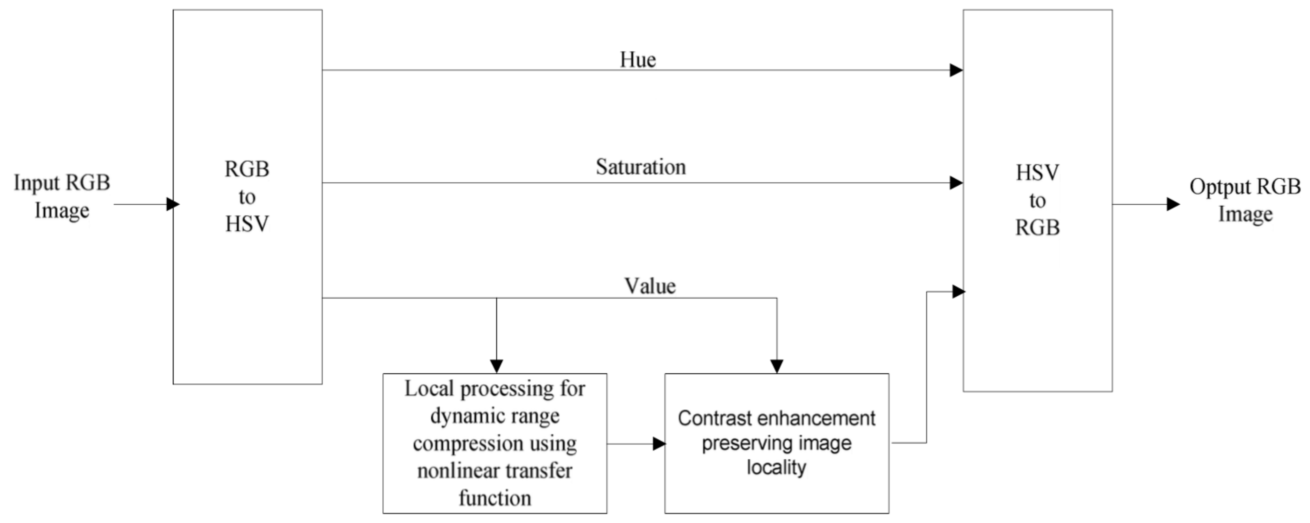

Input RGB Image
RGB to HSV
Hue
Saturation
Value
HSV to RGB
Optput RGB Image
Local processing for dynamic range compression using nonlinear transfer function
Contrast enhancement preserving image locality

Figure 1: Flowchart of the method to be implemented

The rest of the paper will be organized as follows: In section II, the method we implement will be described. In section III, the experiment result will be shown. Also, the result of histogram equalization and INDANE will also be shown for comparing. In section IV, the result of comparing will be discussed and the conclusion will be drawn.

II. Image enhancement using nonlinear transfer function

In general, images are represented in RGB color model. However, HSV is closer to human perception, where H refers to hue which is the spectral composition of the color, S the saturation or the purity of the color and V the luminance value of the color. The RGB values can be converted to HSV values with the following equations.

$$H = \begin{cases} H_1 & B \leq G \\ 360 - H_1 & B > G \end{cases} \tag{1}$$

where,

$$H_1 = \cos^{-1} \frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}} \tag{2}$$

$$S = \frac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)} \tag{3}$$

$$V = \frac{\max(R,G,B)}{255} \tag{4}$$

A. Luminance Enhancement

Luminance enhancement is our first step to enhance the image. It is also known as dynamic range compression, which is used for low or nonuniform illumination. This process is applied to the V component obtained by (4). We use a specifically designed nonlinear transfer function to do the luminance enhancement. The transfer function is as follows:

$$V_{LE} = \frac{V^{0.75x+0.25} + 0.4(1-x)(1-V) + V(1-x)}{2} \tag{5}$$

Here the x is the parameter describing the local luminance information of the image. It is calculated by the function below.

$$x = \begin{cases} 0 & L \leqslant 50 \\ \frac{L-50}{100} & 50 < L \leqslant 150 \\ 1 & L > 150 \end{cases} \tag{6}$$

where L is the intensity level corresponding to a cumulative distribution function(CDF) of 0.1. It means when more than 90% of all the pixels have intensity higher than 150, x is 1. If 10% or more of all pixels have intensity lower than 50, x is zero. Otherwise the $x = \frac{L-50}{100}$. For example, for $L = 80$,then $x = 0.3$, we have the following curve in Figure 2 corresponding to (5), where the x axis is V and y axis is V_{LE}.

The parameter x takes into account the local intensity of different regions of the image and thus preserves the details. If x is a global parameter, for different parts of the image, we won't achieve a good enough enhancement result. For different x, we will have different shape of the transfer function for different regions of the image, which produces better luminance enhancement for the whole image.

To achieve the goal of setting different values for x according to the local luminance information, we have the following steps. First, we would divide the whole image into blocks

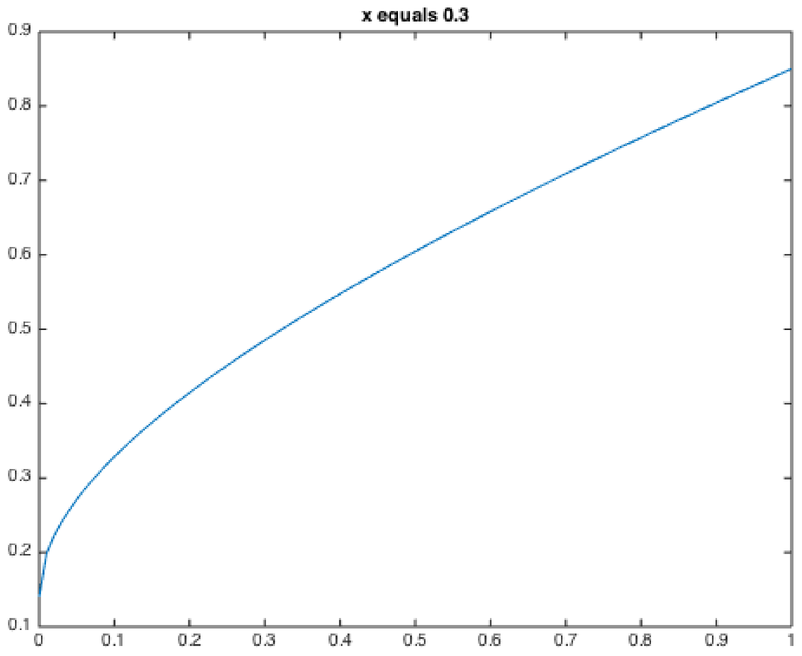


Figure 2: Nonlinear transfer function shape

of equal size N. The histogram of each block will be utilized for calculating x. Second, after obtaining different values of x, we again divide each block into sub-blocks of size $n \times n$. For the previously calculated x, we treat them as the center parameters of the blocks and then for every four center parameters, we use bi-linear interpolation to calculate the parameters for each pixel inside the squares formed by the four centers. The interpolation is used to eliminate the blocking artifacts. In this way, we have the x corresponding to each pixel of the image. Using the x we get, we can finish the luminance enhancement process and get V_{LE}.

B. Contrast Enhancement

After the luminance enhancement, contrast enhancement will be performed. Unlike the conventional techniques, the contrast enhancement used here is an adaptive process based on the intensity information of both the center(processed) pixel and its surrounding pixels. The neighboring luminance information of the center pixel is obtained by convolving with a Gaussian kernel having a form like

$$G(x,y) = K \cdot e^{\frac{-(x^2+y^2)}{c^2}} \tag{7}$$

where K is determined by

$$\iint K \cdot e^{\frac{-(x^2+y^2)}{c^2}} \, \mathrm{d}x\mathrm{d}y = 1 \tag{8}$$

and c is the scale or Gaussian surround space constant. The convolution can be expressed as

$$V_c(x,y) = V_F(x,y) \otimes G(x,y) \tag{9}$$

The convolution result contains the luminance information of neighboring pixels. It is compared with the center pixel for contrast enhancement process. The process is described by the following two equations.

$$V_{CE}(x,y) = 255 V_{LE}(x,y)^{E(x,y)} \tag{10}$$

where $E(x,y)$ is defined by

$$E(x,y) = R(x,y)^p = [V_C(x,y)/V_F(x,y)]^p \tag{11}$$

$V_{CE}(x,y)$ is the V component after contrast enhancement, and $R(x,y)$ is the ratio of Gaussian filtered and original V component image. p is an image dependent parameter determined by using global standard deviation of the original V component imageV_F. If the center pixel is brighter than the surrounding pixels, the ratio $R(x,y)$ will be smaller than 1 and thus the luminance is decreased, in which way the contrast is pulled up. On the other hand if the center pixel is darker than the surrounding pixels, the ratio $R(x,y)$ will be larger than 1 and hence the luminance is increased. The contrast will be lowered. To get the optimal result, different scales of the Gaussian kernel can be used and the linear combination of multiple convolution results will be the final output, which can be expressed like

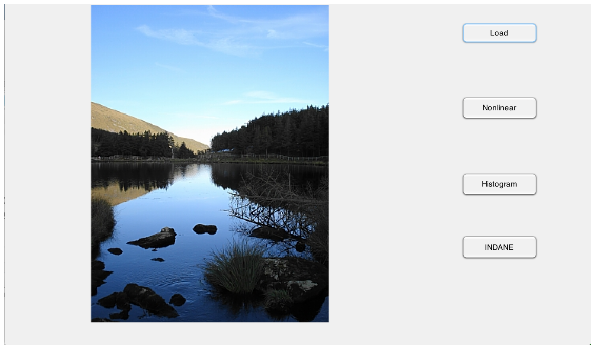
$$V_{CE}(x,y) = \sum_i w_i V_{CE}^i(x,y), i = 1,2,3... \tag{12}$$

where i represents different scales and w_i is the weight for each output $V_{CE}^i(x,y)$. It may require much computation time for a weighted output, so in this paper we just use a single scale for contrast enhancement. Through this process, the contrast of the image will be improved significantly. After the contrast enhancement, the H, S, and V_{CE} are combined together to form a color image. Finally the RGB image is obtained by converting HSV image back to RGB color space.

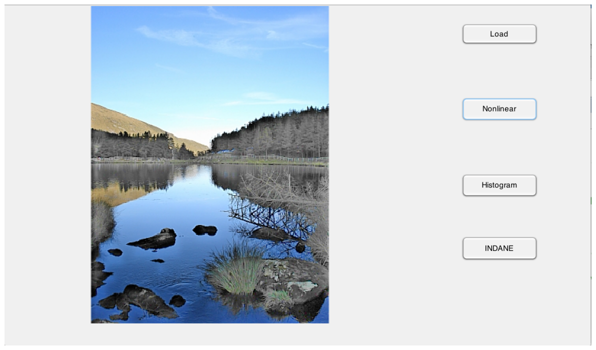
III. Experiment result and comparison of different methods

The method we are focusing on is applied to a number of images under dark illumination conditions and is used to compare with histogram equalization and INDANE. In experiment, we choose the N and n according to the size of the images.

We use three images for comparing different methods. For convenience, We build a GUI for the comparison. Each method can be implemented using a button. The images are as follows.

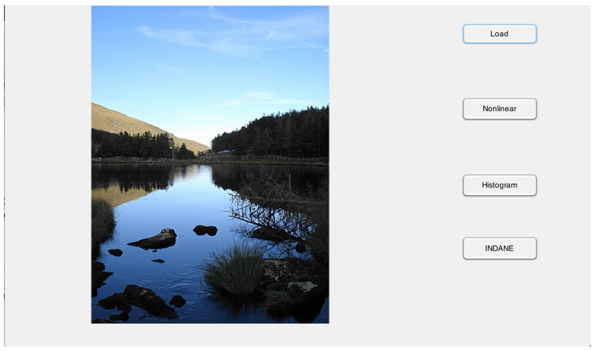


(a) original image

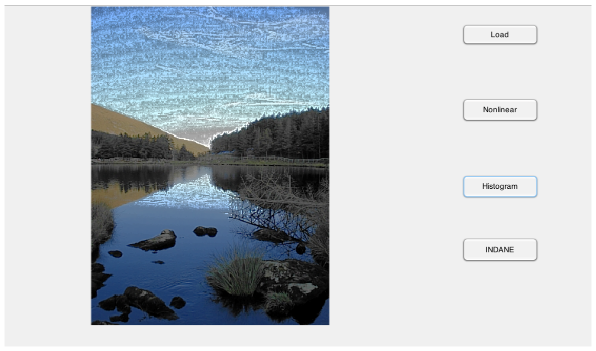


(b) image after processing

Figure 3: Proposed method

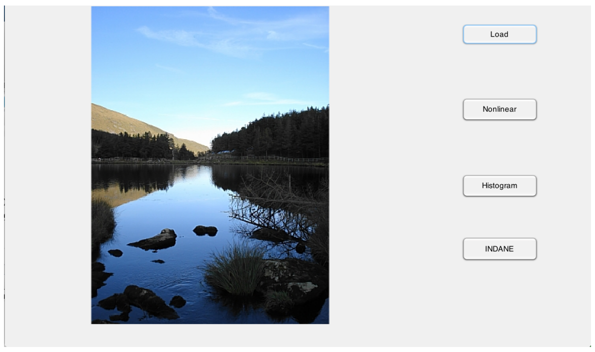


(a) original image

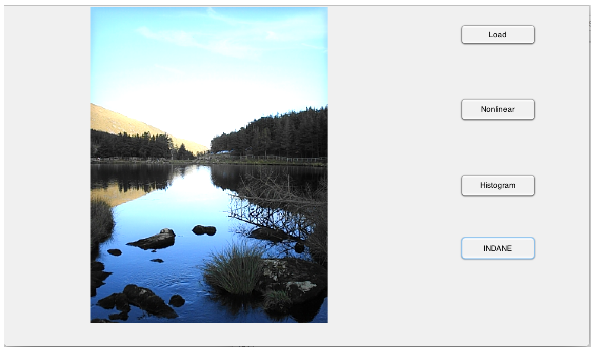


(b) image after processing

Figure 4: HE method

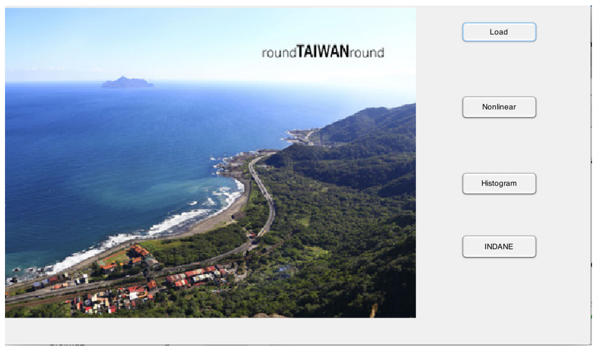


(a) original image

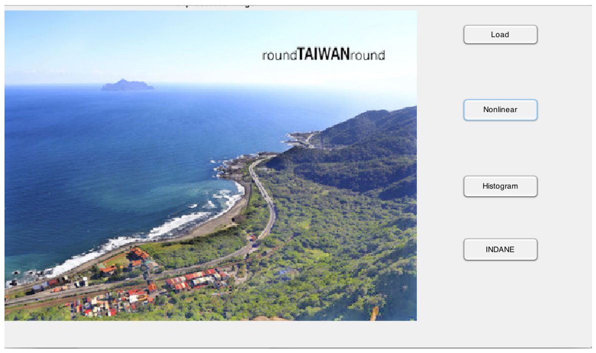


(b) image after processing

Figure 5: INDANE

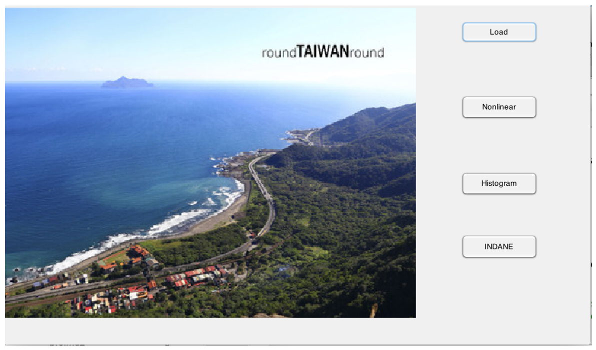


(a) original image

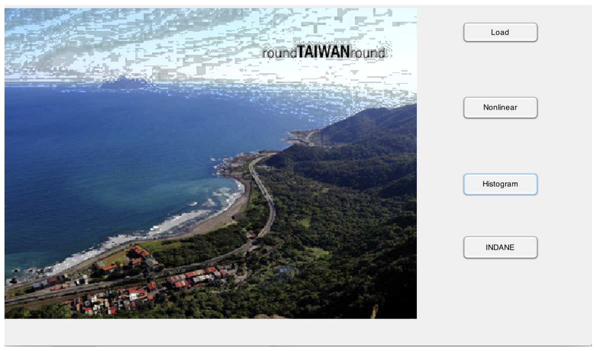


(b) image after processing

Figure 6: Proposed method

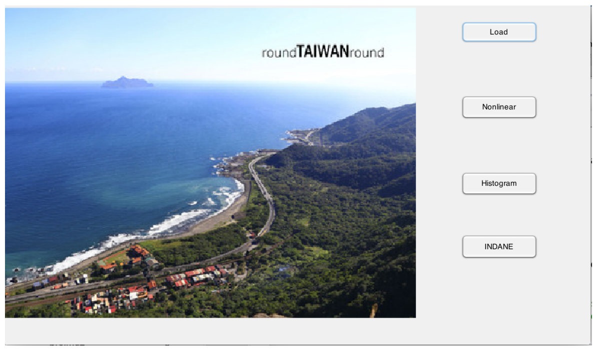


(a) original image

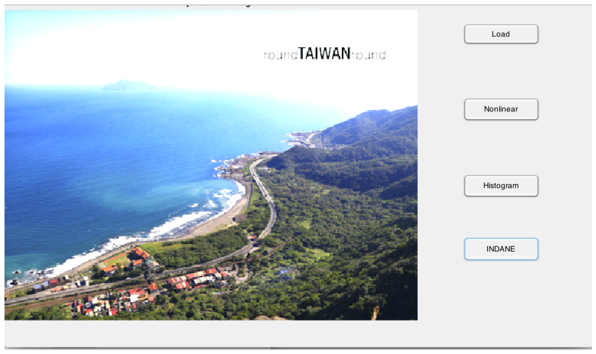


(b) image after processing

Figure 7: Histogram Equalization

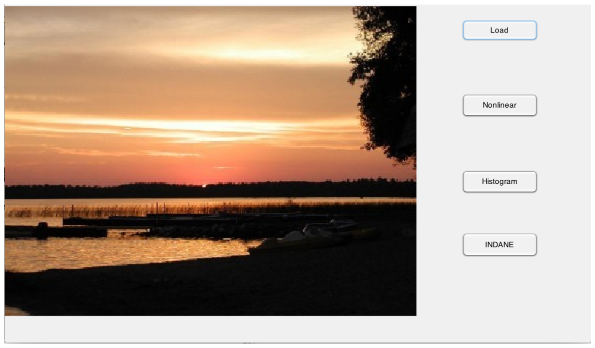


(a) original image

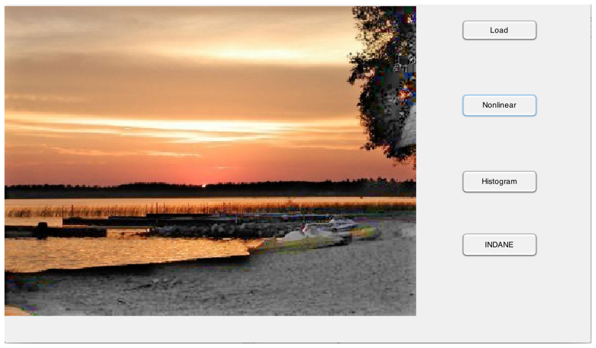


(b) image after processing

Figure 8: INDANE

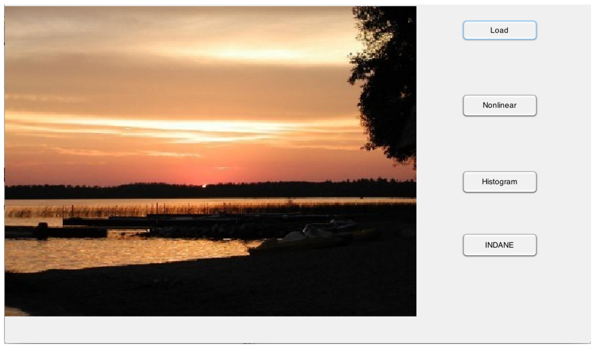


(a) original image

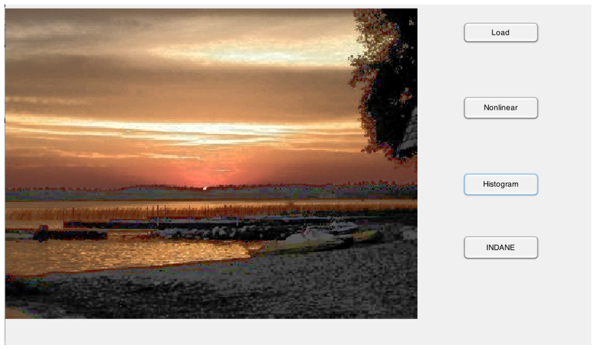


(b) image after processing

Figure 9: Proposed method

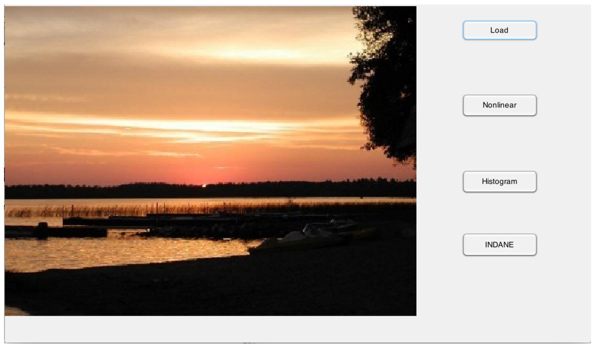


(a) original image

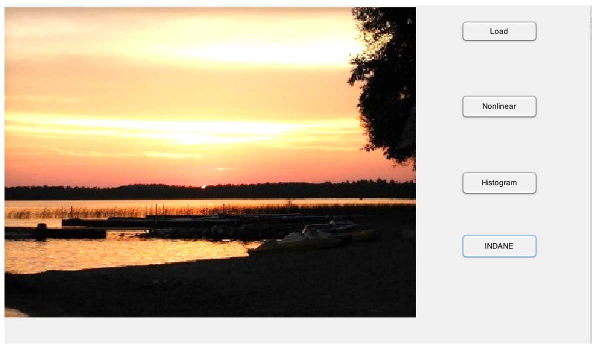


(b) image after processing

Figure 10: Histogram Equalization



(a) original image



(b) image after processing

Figure 11: INDANE

IV. Conclusions

Here we discuss the results of different methods. For these images, the processed images using the proposed method are strongly enhanced. The colors are preserved while enhancing the details. For example, in the first image, we can see the details of the grass and the trees and the colors are preserved. The images after histogram equalization are also enhanced because we can see the details of the grass and trees too. However, there is a big problem with HE, which is the artifacts appear in the first image. The images after using INDANE are brighter and also enhanced a little. But the enhancement is not obvious and the colors are changed a little.

In conclusion, the proposed method achieves a better performance in color image enhancement. It can enhance the details and preserve the original color at the same time.

V. References

[1]Deepak Ghimire, Joonwhoan Lee, "Nonlinear Transfer Function-Based Local Approach for Color Image Enhancement," *IEEE Transactions on Consumer Electronics*, Vol. 57,No.2, May 2011.

[2] Rafael, C. Gonzalez and Richard E. Woods. *Digital Image Processing*, 2nd edition, Prentice Hall, 2002.

[3]Youhei Terai, Tomio Goto, Satoshi Hirano, and Masaru Sakuari "Color Image Contrast Enhanncement by Retinex Model",13th *IEEE International Symposium on Consumer Electronics* 2009, pp. 392-393.

[4]Li Tao, and V. K. Asari, "Adaptive and integrated neighborhood- dependent approach for nonlinear enhancement of color images", *Journal of Electron Imaging*, vol. 14, no. 4, pp.

043006-1-043006-14, Dec. 2005.

[5]Hanumantharaju M.C., Ravishankar M., Rameshbabu D.R., and Ramchandran S., "Color Image Enhancement using Multiscale Retinex with Modified Color Restoration Technique,"Second *International Conference on Emerging Applications of Information Tecnology* 2011, pp. 93-97.

[6]Xiong Jie, Han Li-na, Geng Guo-hua, and Zhou Ming-quan, "Real Color Image Enhancement by Illumination-Reflectance Model and Wavelet Transform," *International Forum on Information Technology and Applications* 2009, pp. 691-695.

VI. Vita

Yuhao Huang was born in China on February 21,1992. He received his B.E. degree in communications engineering from Beijing University of Posts and Telecommunications, Beijing, China in 2014. He is now a Master student in Northeastern University, Boston, U.S. His research interests include image processing and machine learning.

VII. Appendix

Code of implementing the proposed method:

```
clc;

clear;

f_rgb=imread('img15.jpg'); %input image in rgb format

f_hsv=rgb2hsv(f_rgb); %input image in hsv format

row_num=size(f_rgb,1);

col_num=size(f_rgb,2);
```

```
%----decomposing H,S,V components----%

h_comp=f_hsv(:,:,1);

s_comp=f_hsv(:,:,2);

v_comp=f_hsv(:,:,3);

% v_comp=(v_comp.^0.24+(1-v_comp)*.5+v_comp.^2)/2;

%%----luminance enhancement process----%%

%----setting the window -----%

blk_size=10;   %block size

[v_lum,x_matrix]=lumi_enh(v_comp,blk_size);

img_lum=rgbconv(h_comp,s_comp,v_lum);

%%----contrast enhancement----%%

%----using Gaussian filter to obtain surrounding luminance information---%

sig=10;

gauss_filt=fspecial('gaussian', [row_num,col_num], sig);

% gauss_filt=gauss_init(v_comp,sig);

v_filt=imfilter(v_comp,gauss_filt,'conv');

%-----calculating the exponent E(x,y) and get the final enhanced V component------%

v_final=cont_enh(blk_size,v_filt,v_comp,v_lum);

% s_enh=s_comp.^(1);

%v_final=v_conv(cA_lum_enh,cH_enh,cV_enh,cD_enh);

f_final=rgbconv(h_comp,s_comp,v_final);

%-----plotting figures-----%

figure
```

```
imshow(f_final);

title('final processed image');


figure

imshow(f_rgb);

title('original image');


figure

imshow(img_lum);

title('image after luminance enhancement');
```

Code of INDANE:

```
clc;

clear;

f_rgb=imread('img2.jpg'); %input image in rgb format

f_gray=rgb2gray(f_rgb); %input image in hsv format

f_g_ori=f_gray;

f_gray=im2double(f_gray);

row_num=size(f_rgb,1);

col_num=size(f_rgb,2);

%----decomposing R,G,B components----%

r_comp=im2double(f_rgb(:,:,1));

g_comp=im2double(f_rgb(:,:,2));
```

```
b_comp=im2double(f_rgb(:,:,3));

%%----luminance enhancement process----%%

f_gray=(f_gray.^0.24+(1-f_gray)*.5+f_gray.^2)/2;

% img_lum=rgbconv(h_comp,s_comp,v_lum);

%%----contrast enhancement----%%

%----using Gaussian filter to obtain surrounding luminance information---%

sig=[5,20,240];

wgt=[1/3 1/3 1/3];

gauss_filt=zeros(row_num,col_num,3);

for i=1:3

    gauss_filt(:,:,i)=fspecial('gaussian', [row_num,col_num], sig(i));

    f_filt(:,:,i)=imfilter(f_gray,gauss_filt(:,:,i),'conv');

end


%-----calculating the exponent r(x,y)------%

for i=1:3

    r(:,:,i)=r_calcul(f_filt(:,:,i),f_gray);

    f_R(:,:,i) = f_filt(:,:,i).^r(:,:,i);

end

% f_fingr=cont_enh(blk_size,f_filt,f_gray,v_lum);

f_R=wgt(1)*f_R(:,:,1)+wgt(2)*f_R(:,:,2)+wgt(3)*f_R(:,:,3);

% f_R=uint8(f_R);

lambda=1.5;
```

```
% f_gray=im2uint8(f_gray);

f_R_r=f_R.*(r_comp./f_gray)*lambda;

f_R_g=f_R.*(g_comp./f_gray)*lambda;

f_R_b=f_R.*(b_comp./f_gray)*lambda;

f_rgb_enh=cat(3,f_R_r,f_R_g,f_R_b);

f_rgb_enh=im2uint8(f_rgb_enh);

% f_R=rgbconv(h_comp,s_comp,f_R);

%-----plotting figures-----%

figure

imshow(f_rgb_enh);

title('final processed image');

figure

imshow(f_rgb);

title('original image');
```

Code of HE:

```
clc;

clear;

f=imread('img2.jpg');

f=rgb2hsv(f);

f_H=f(:,:,1);

f_S=f(:,:,2);

f_V=f(:,:,3);
```

```
f_he_R= histeq(f_H);

f_he_G= histeq(f_S);

f_he_B= histeq(f_V);

% f_he=cat(3,f_he_R,f_he_G,f_he_B);

f_he=cat(3,f_H,f_S,f_he_B);

f=hsv2rgb(f);

f_he=hsv2rgb(f_he);

figure

imshow(f_he);

figure

imshow(f);
```

Functions:

```
function [enh_image,x_matrix]=lumi_enh(image,blk_size)

xblkstart=1;

yblkstart=1;

n_ovlap=blk_size;

numblk_row=(size(image,1)-blk_size)/n_ovlap+1;

numblk_col=(size(image,2)-blk_size)/n_ovlap+1;


for i=1:numblk_row

    for j=1:numblk_col
```

```
        temp=image( xblkstart : xblkstart + blk_size-1,yblkstart : yblkstart+blk_size-1)

        temp_hist = imhist(temp);

        blkcdf = cumsum(temp_hist); %transformation function

        blkcdf_norm = blkcdf / max(blkcdf);

        L=find(blkcdf_norm>=0.1, 1 );

        x=findz(double(L));

        x_matrix(i,j)=x;

        %enh_block=enhapprox(z,temp,b,b);

        %enh_a(x:x+b-1,y:y+b-1)=enh_block;

        yblkstart=yblkstart+n_ovlap;

    end

    yblkstart=1;

    xblkstart=xblkstart+n_ovlap;

end


x_temp=padarray(x_matrix,[1,1],1,'both');

%--------interpolating x for each pixel inside the block(window)--------%

x_interp=resizem(x_temp,blk_size,'bilinear');

z_stpt=floor(blk_size/2);

x_modif=x_interp(z_stpt+1:size(image,1)+z_stpt,z_stpt+1:size(image,2)+z_stpt);


enh_image=LE_vcomp(x_modif,image,size(image,1),size(image,2));
```

```
function V_lumenh=LE_vcomp(x,image,size_row,size_col)

V_lumenh=zeros(size(image,1),size(image,2));

for i=1:size_row

    for j=1:size_col

        V_lumenh(i,j)=(image(i,j)^(0.75*x(i,j)+0.25)+0.4*(1-x(i,j))*(1-image(i,j))+(imag

    end

end


function E=E_calcul(V_filt,image,blk_size)



num_row=size(image,1)/blk_size;

num_col=size(image,2)/blk_size;

x=1;

y=1;

E=zeros(size(image,1),size(image,2));

for i=1:num_row

    for j=1:num_col

        temp=image(x:x+blk_size-1,y:y+blk_size-1);

        temp_filt=V_filt(x:x+blk_size-1,y:y+blk_size-1);

        g=find_g(temp);

        %E_new=updateE(g,x,y,b,temp,temp_filt);

        E(x:x+blk_size-1,y:y+blk_size-1)=(temp_filt./temp).^g;

        y=y+blk_size;
    end

    y=1;

    x=x+blk_size;

end


function V_ce=cont_enh(blk_size,v_filt,v_comp,v_lum_enh)


E = E_calcul(v_filt,v_comp,blk_size);

V_ce=(v_lum_enh).^E;


function img = rgbconv(h,s,v)

hsvimg=cat(3,h,s,v);

img = hsv2rgb(hsvimg);


function z=findz(L)

if L<=50

    z=0;

elseif (L>50) && (L<=150)

    z=(L-50)/100;

else

    z=1;

end
```

```
function g=find_g(temp)

temp_std=std(temp(:));

if temp_std<=2

    g=0.45;

elseif temp_std>2 && temp_std<10

    g=(27-2*temp_std)/13;

else

    g=0.25;

end
```