

# Identification of Application-Suitability of Point Operations on Digital Images using MATLAB

R. Swaminathan

Asst.Professor

Ururu Dhanalakshmi College, Kattur,  
Tiruchirapalli -19,  
Tamilnadu, India

## ABSTRACT

Spatial Domain and Frequency Domain operations play an important role in Digital Image Processing. Spatial Domain operations directly manipulate the pixels of a given image and contain a different category of operations, which include point, local and global operations. This paper mainly deals with point operations in order to identify their application-suitability on digital images using MATLAB and proposes a new function on contrast-stretching.

## Keywords

Spatial Domain, Image Negative, Square root, Logarithm, Thresholding, Contrast Stretching.

## 1. INTRODUCTION

Digital images are more popular with computer-users rather than analog images. The reason is that digital images can be stored in a digital computer, can be transmitted over networks, and some digital processing can be applied to them using digital computers to extract useful information. Spatial Domain Operations that can be applied to digital images include point, neighbourhood and global operations. Among these operations this paper mainly deals with point operations and proposes a new function that is based on the existing one given below

$$g = 1./(1 + (m ./ f) .^E) \quad [1]$$

Where  $f$  denotes the input image,  $g$  denotes the output image. In this function the two elements  $m$  and  $E$  play an important role in determining the contrast of the image ( $g$ ). The new function which has been derived from the above function is given below

$$g = (1./ (1 + (m ./ f) .^E)) + c$$

Where  $c$  denotes the constant value, which plays an important role in determining the contrast of the image ( $g$ ) and simplifies the operation of contrast stretching that is being carried out by the existing function mentioned above.

## 2. SPATIAL DOMAIN

The operations studied by this paper can be represented by the following expression.

$$o_i(x,y) = T_f[i_i(x,y)]$$

$i_i(x,y)$  is the input image,  $o_i(x,y)$  is the output image and  $T_f$  is the transformation function which can be applied either to a single image or to a group of images. The transformation function that operates on a single image is illustrated with the following figure.

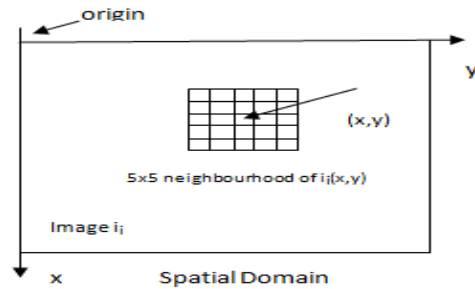


Fig 1 : Neighbourhood pixels of  $(x, y)$

The above picture has been drawn based on the information given in the reference [1] & [3].

## 3. IMAGE NEGATIVE

This is one of the image processing techniques, which is used to enhance white or gray details embedded in dark regions of an image. It can be expressed using mathematical notations as follows

$$g(x,y) = 255 - f(x,y) \quad [2]$$

Where  $f(x,y)$  is the input image and  $g(x,y)$  is the output image.

### 3.1 Implementation

```
img = imread('e:\leaf.jpg');  
img=rgb2gray(img);  
[rows,cols]=size(img);  
neg_img=zeros(rows,cols);  
for row = 1:rows  
for col = 1:cols  
neg_img(row,col)=255-img(row,col);  
end  
end  
subplot(1,2,1);  
imshow(img);  
title('a');  
subplot(1,2,2);  
imshow(neg_img,[]);  
title('b');
```

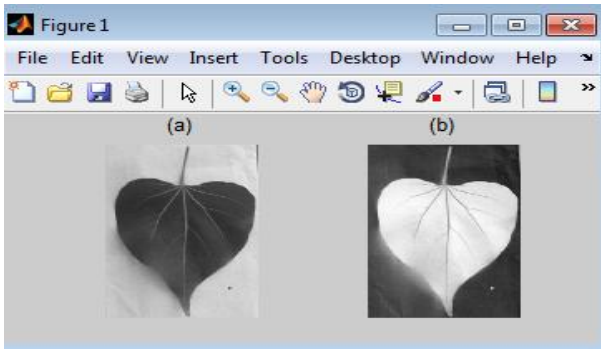


Fig 2: (a) Gray scale image (b) Negative image

In the above figure, the negative image clearly shows the vein of a Portia – leaf than its gray image

#### 4. SQUARE ROOT

This function expands the gray scale range in dark areas of an image. The following mathematical notation represents a square root function.

$$g = \sqrt{255f(x,y)} \quad [2]$$

Here  $f(x,y)$  is the pixel value of the input image and  $g$  is the output image. In this paper the above function has been implemented as the following

$$g(x,y) = \sqrt{f(x,y)}$$

Where  $g(x,y)$  is the output image and  $f(x,y)$  is the input image.

#### 4.1 Implementation

```
clear all; close all;
img = imread('e:\tv.jpg');
img = rgb2gray(img);
img2 = im2double(img).^(1/2);
figure;
subplot(1,2,1);
imshow(img);
title('a');
subplot(1,2,2);
imshow(img2); title('b');
```

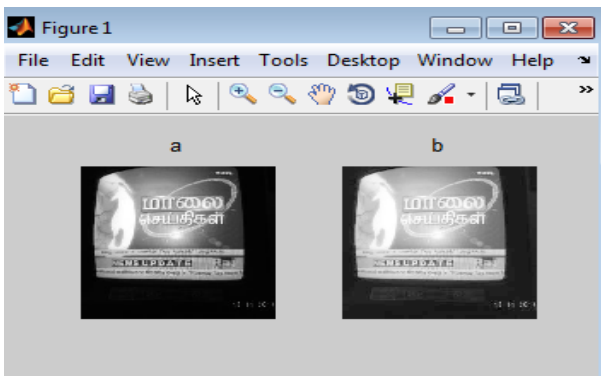


Fig 3: (a) Original image (b) Its square root

In the above figure, the dark portions of the original image have been expanded and the result of this expansion process is shown in the next image. i.e., the darkness of the second image is lesser than its input image.

#### 5. LOGARITHM

It is one of the point operations, which can be used to reduce the contrast of brighter regions of an image and is implemented using the following expression.

$$g = c * \log_c(1 + f) \quad [1]$$

Where  $g$  is the output,  $f$  is the input and  $c$  is the constant.

#### 5.1 Implementation

```
clear all
inputim=imread('e:\spider.jpg');
inputim=imresize(inputim,[256,256]);
imdoube = im2double(inputim);
cim = imdoube;
[row,col] = size(imdoube);
con = 2;
for i = 1 : row
    for j = 1 :col
        cim(i,j)= con*log(1+imdoube(i,j));
    end end
subplot(1,2,1);
imshow(imdoube); title('a');
subplot(1,2,2);
imshow(cim);title('b');
```

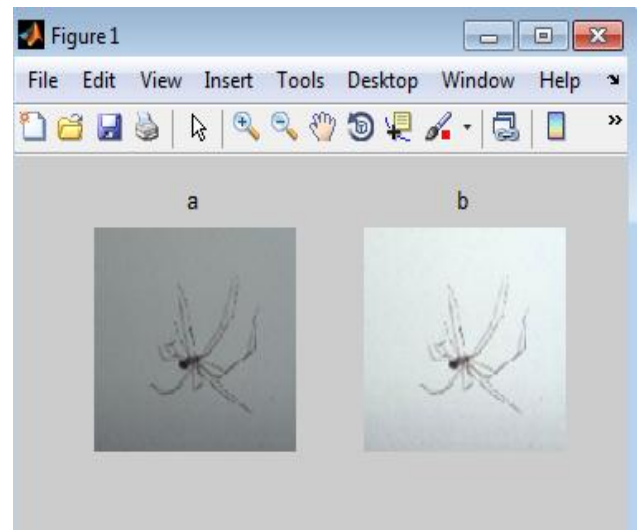


Fig 4: (a) Original image comprising a spider (b) its log form

In the above figure the second image shows the result of the logarithm applied to the original image (first image). It is obvious that the contrast of the second image has been reduced.

## 6. THRESHOLDING

Gray level transformations that are based on point operations can be represented as

$$o_i(x,y) = T_f [i_i(x,y)]$$

The above function implies  $[v = T_f(u)]$ .  $i_i(x,y)$  is the input image,  $o_i(x,y)$  is the output image and  $T_f$  is the transformation function,  $v$  is the gray level of the output image and  $u$  is the gray level of the input image. The following figure illustrates that the pixel whose value is greater than the threshold value is transformed into white and the pixel whose value is lesser than the threshold value is transformed into black.

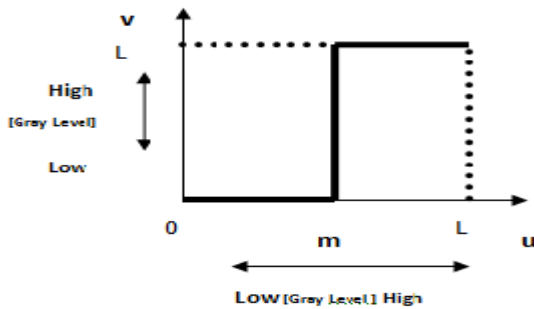


Fig 5: Operation of thresholding

In the above figure  $m$  represents the threshold value and the value of  $v$  is set to 0(zero) if  $u \leq m$  or set to 255 if  $u > m$ .

### 6.1 Implementation

```
myimage=imread('e:\nature.jpg');
Tvalue=80;
myimage=rgb2gray(myimage);
[rows,cols]=size(myimage);
subplot(1,2,1);
imshow(myimage);
title('a');
cimage = myimage;
for i=1:rows
for j=1:cols
if(cimage(i,j)>Tvalue)
cimage(i,j)=255;
else
cimage(i,j)=0;
end
end
end
subplot(1,2,2);
imshow(cimage);
title('b');
```

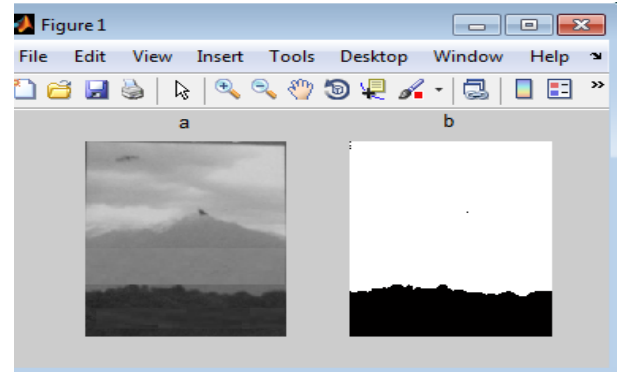


Fig 6: (a) Original image (b) Binary image

## 7. CONTRAST STRETCHING

It is a point operation which darkens the dark portion and brightens the bright portion of an image based on the threshold value. A mathematical notation for this operation given below

$$s = T(r) = \frac{1}{1+(m/r)^E} \quad [1]$$

Where  $r$  denotes the intensities of the input image,  $s$  denotes the corresponding intensity values in the output image, and  $E$  controls the slope of the function. This equation is implemented in MATLAB for a floating point image as

$$g = 1./(1 + (m./f).^E) \quad [1]$$

### 7.1 Implementation

```
Iimage=imread('e:\nature.jpg');
f =double(Iimage);
E=3.2;
m=78;
Oimage = 1 ./ (1 + (m./f).^E);
g =im2uint8(mat2gray(Oimage));
subplot(1,2,1),imshow(Iimage);
title('a');
subplot(1,2,2),imshow(g);
title('b');
```

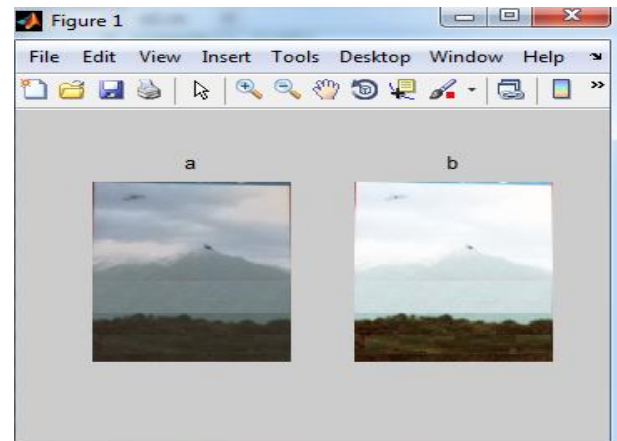


Fig 7: (a) Original image (b) Contrast stretched image

The contrast of the image produced by the function

$$g = 1./ (1 + (m./f) .^E)$$

can be further increased or decreased by adding a constant value to it. The changed function and its implementation are given below

$$g = (1./ (1 + (m./f) .^E)) + c$$

## 7.2 Implementation

```
Iimage=imread('e:\nature.jpg');
f =double(Iimage);
E=3.2; m=78;
g = 1 ./ (1 + (m./f) .^E);
sub_g=im2uint8((mat2gray(1./(1+(m./f)
.^E))))+ -60;
add_g=im2uint8((mat2gray( 1./(1 +
(m./f) .^E))))+ 60;
subplot(1,2,1),imshow(sub_g);
title('b1');
subplot(1,2,2),
imshow(add_g);
title('b2');
```

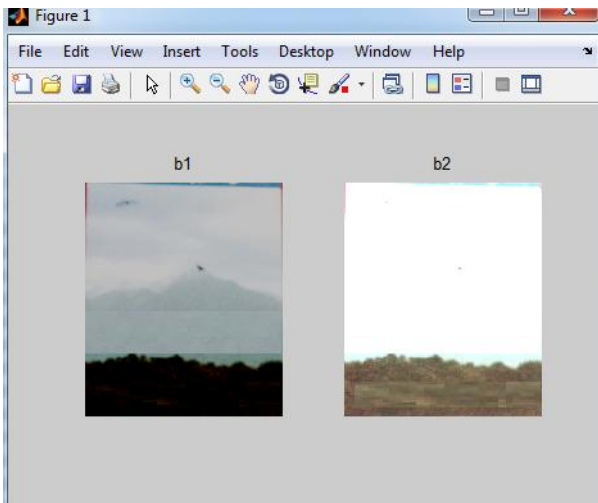


Fig 8: (b1) Low-Contrast (b2) High-Contrast

The above figure shows the operation of increasing and decreasing the contrast of the image (g) can be made very easily by adding a constant value to the existing function.

Table 1. Details of the function applied to the Fig 7:(a)

Function applied: $g = (1./ (1 + (m./f) .^E)) + c$			
m	E	c	Result
78	3.2	-60	Fig 8: (b1)
78	3.2	60	Fig 8: (b2)

## 8. CONCLUSION

This paper has made a study of point operations by applying them to digital images taken by a SONY Cyber-Shot camera and identified a new function  $g = (1./ (1 + (m./f) .^E)) + c$  based on the existing function  $g = 1./ (1 + (m./f) .^E)$  [1] for the purpose of simplifying the operation of contrast stretching that is being carried out by  $g = 1./ (1 + (m./f) .^E)$  [1] and removing the need for changing the values of m and E every time the contrast of the image (g) needs to be changed.

## 9. REFERENCES

- [1] Digital Image Processing using MATLAB, Second Edition, Rafael C. Gonzales, Richard E. Woods, Steven L. Eddins, McGraw Hill Education (India) Private Limited, New Delhi.
- [2] Digital Image Processing, S. Sridhar, Oxford University Press.
- [3] N. Mohanapriya, B. Kalaavathi, PhD, Comparative Study of Different Enhancement Techniques for Medical Images, International Journal of Computer Applications (0975 – 8887) Volume 61– No.20, January 2013
- [4] Digital Image Processing, Third Edition, Rafael C. Gonzalez, Richard E. Woods, Pearson.
- [5] Fundamentals of Digital Image Processing, ANIL K. JAIN, PHI Learning Private Limited, 2013.