International Journal of Advance Research in Engineering, Science & Technology(IJAREST), ISSN(O):2393-9877, ISSN(P): 2394-2444,
Volume 2, Issue 6, June - 2015, Impact Factor: 2.125

Implementation of Image Enhancement by Various Methods

Sonal B Prajapati[1], Priyanka N Dadwani[2], Prashant R Polawala[3]
[1]Department of Electronics & Communication,VIE ,VIER
sonalprajapati14@gmail.com
[2]Department of Electrical,VIE ,VIER
Dadwani.priyanka@gmail.com
[3]Department of Electrical,VIE ,VIER
Prp.Rfetr@gmail.com

Abstract

Image Enhancement is one of the most important and difficult techniques in image research. The aim of image enhancement is to improve the visual appearance of an image, or to provide a "better transform representation for future automated image processing. Many images like medical images ,satellite images, aerial images and even real life photographs suffer from poor contrast and noise. It is necessary to enhance the contrast and remove the noise to increase image quality. One of the most important stages in medical images detection and analysis is Image Enhancement techniques which improves the quality (clarity) of images for human viewing, removing blurring and noise, increasing contrast, and revealing details are examples of enhancement operations. The enhancement technique differs from one field to another according to its objective. In this paper, "Bit Plane Slicing" method is investigated using MATLAB/Simulink to understand the importance of each bit plane in the image.

Key words —*Image Enhancement, MATLAB/Simulink, Bit Plane Slicing.*

Introduction

The existing techniques of image enhancement can be classified into two categories: Spatial Domain and Frequency domain enhancement. Image enhancement is basically improving the interpretability or perception of information in images for human viewers and providing `better' input for other automated image processing techniques. The principal objective of image enhancement is to modify attributes of an image to make it more suitable for a given task and a specific observer. During this process, one or more attributes of the image are modified. The choice of attributes and the way they are modified are specific to a given task. Moreover, observer-specific factors, such as the human visual system and the observer's experience, will introduce a great deal of subjectivity into the choice of image enhancement methods. There exist many techniques that can enhance a digital image without spoiling it. 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 [1], we directly deal with the image pixels. The pixel values are manipulated to achieve desired enhancement. In frequency domain methods, the image is first transferred in to frequency domain. It means that, the Fourier Transform of the image is computed first. All the enhancement operations are performed on the Fourier transform of the image and then the Inverse Fourier transform is performed to get the resultant image. These enhancement operations are performed in order to modify the image brightness, contrast or the distribution of the grey levels. As a consequence the pixel value (intensities) of the output image will be modified according to the transformation function applied on the input values. Image enhancement is applied in every field where images are ought to be understood and analyzed. For example, medical image analysis, analysis of images from satellites etc. Image enhancement simply means, transforming an image f into image g using T. (Where T is the transformation. The values of pixels in images f and g are denoted by r and s, respectively. As said, the pixel values r and s are related by the expression,

$$s = T(r) \qquad (1)$$

Where T is a transformation that maps a pixel value r into a pixel value s. The results of this transformation are mapped into the grey scale range as we are dealing here only with grey scale digital images. So, the results are mapped back into the range [0, L-1], where L=2k, k being the number of bits in the image being considered. So, for instance, for an 8-bit image the range of pixel values will be [0, 255].

Bit-plane slicing

Instead of highlighting gray-level ranges, highlighting the contribution made to total image appearance by specific bits might be desired. Suppose that eachpixel in an image is represented by 8 bits. Imagine that the image is composed of eight 1-bit planes, ranging from bit-plane 0 for the least significant bit to bitplane 7 for the most significant bit. In terms of 8-bit bytes, plane 0 contains all the lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all the high-order bits.Figure 1 shows the various bit planes for the image shown in Fig. 3.13. Note that the higher-order bits (especially the top four) contain the majority of the visually significant data.The other bit planes contribute to more subtle details in the image. Separating a digital image into its bit planes is useful for analyzing the relative importance played by each bit of the image, a process that aids in determining the adequacy of the number of bits used to quantize each pixel. Also, this type of decomposition is useful for image compression.

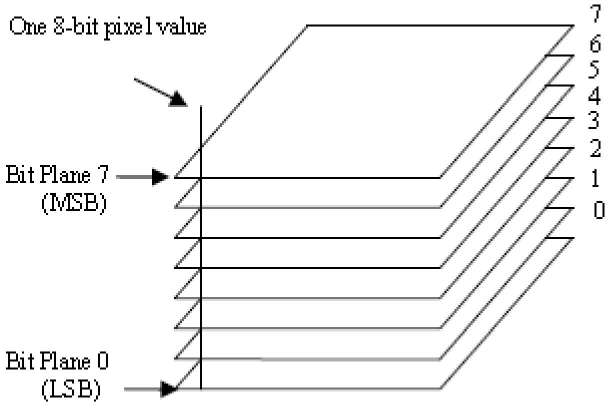

Figure 1

III. SIMULATION AND RESULTS

(a) Instructions

imshow(I) displays the image I in a Handle Graphics® figure, where I is a grayscale, RGB (truecolor), or binary image. For binary images, imshow displays pixels with the value 0 (zero) as black and 1 as white.

double(x) returns the double-precision value for X. If X is already a double-precision array, double has no effect.

bitand(integ1,integ2) returns the bit-wise AND of values integ1 and integ2.

(b)Program 1

```
I=imread('cameraman.tif');
I1=double(I)
imshow(I)

MSB=128*ones(256,256);
MSB_1=bitand(MSB,I1);
subplot(2,4,1);imshow(MSB_1)

plane1=64*ones(256,256);
plane_1=bitand(plane1,I1)
subplot(2,4,2); imshow(plane_1)

plane2=32*ones(256,256);
plane_2=bitand(plane2,I1)
subplot(2,4,3); imshow(plane_2)

plane3=16*ones(256,256);
plane_3=bitand(plane3,I1)
subplot(2,4,4); imshow(plane_3)

plane4=8*ones(256,256);
plane_4=bitand(plane4,I1)
subplot(2,4,5); imshow(plane_4)

plane5=4*ones(256,256);
plane_5=bitand(plane5,I1)
subplot(2,4,6); imshow(plane_5)

plane6=2*ones(256,256);
plane_6=bitand(plane5,I1)
subplot(2,4,7); imshow(plane_6)

plane7=1*ones(256,256);
plane_7=bitand(plane7,I1)
subplot(2,4,8); imshow(plane_7)
```

Fig 3.Bit 0 to Bit 7

(c)Program 2 Gray level slicing

Programme

```
I1= imread('cameraman.tif');
 subplot(1,2,1)
imshow(I1)
A=I1;
[a,b]=size(I1);
for i=1:1:a
  for j=1:1:b
     if (A(i,j)>=100 && A(i,j)<=150)
        B(i,j)=100;
     else
        B(i,j)=A(i,j);
     end
  end
 end
```

```
D=uint8(B)
subplot(1,2,2)
 imshow(D)
```

Fig. 4 Gray level slicing

(D)Program 3 Power law transformation

```
I1= imread('cameraman.tif');
 subplot(1,2,1)
 imshow(I1)
title('original image')
u=double(I1)
s=50*power(u,0.25)
subplot(1,2,2)
imshow(uint8(s))
title('power law tran. 50*(u^.25)')
```

Fig. 5 Power law transformation

(E)Program 4 Negative image

```
 I1= imread('cameraman.tif');
 subplot(1,2,1)
imshow(I1)
title('original image')
A=I1;
[a,b]=size(I1);
for i=1:1:a
   for j=1:1:b
       B(i,j)=(255-A(i,j));
   end
 end
C=uint8(B)
subplot(1,2,2)
imshow(C)
title('negative image')
```

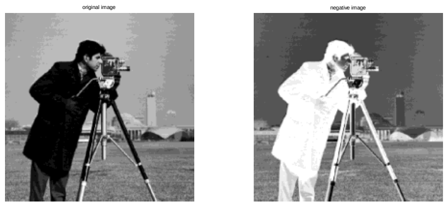

Fig. 6 Negative image

(F)Program 4 Log transformation

```
I1= imread('cameraman.tif');
subplot(1,2,1)
imshow(I1)
title('original image')
A=I1;
[a,b]=size(I1);
for i=1:1:a
for j=1:1:b
B11(i,j)=(A(i,j));
C11(i,j)=30*(log(B11(i,j)));
end
end
D=uint8(C11)
subplot(1,2,2)
imshow(D)
title('LOG TRANSFRMATION')
```

Fig. 7 Log transformation

III.Conclusion

In this paper we have simulated different methods like Gray level slicing , Power law transformation , Negative image , Log transformation for different applications. Here we have also simulated images in different planes from Bit0 to Bit7. The methods for which we have received results are spatial domain method.

REFERENCES

[1] Bhabatosh Chanda and Dwijest Dutta Majumder, 2002, Digital Image Processing and Analysis.
[2] R.W.Jr. Weeks,(1996). Fundamental of Electronic Image Processing . Bellingham: SPIE Press
[3] A. K. Jain, Fundamentals of Digital Image Processing. Englewood Cliffs, NJ: Prentice Hall, 1989.
[4] R.M. Haralick, and L.G. Shapiro, Computer and Robot Vision, Vol-1, Addison Wesley, Reading, MA
[5] R. Jain, R. Kasturi and B.G. Schunck, Machine Vision, McGraw-Hill International Edition, 1995.
[6] W. K. Pratt, Digital image processing, Prentice Hall,

1989.

[7] A.C. Bovik, Digital Image Processing Course Notes, Dept. of Electrical Engineering, U. of Texas at Austin, 1995.

[8] J.C. Russ, The Image Processing Handbook, CRC Press, Boca Raton, FL., 1992.

[9] R Hummel, “Histogram modification techniques“, Computer Graphics and Image Processing, Vol. 4, pp. 209-224, 1975.

[10] S. E. Umbaugh, “Computer Vision & Image Processing,”Prentice Hall PTR, 1998

[11] S. M. Pizer, et al., “Adaptive Histogram Equalization and its Variations,” Comput. Vision, Graphics and Image