Scilab Manual for Image Processing by Mr Gautam Pal Computer Engineering Tripura Institute of Technlogy[1]

Solutions provided by
Mr R.Senthilkumar- Assistant Professor
Electronics Engineering
Institute of Road and Transport Technology

June 30, 2026

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Scilab Manual and Scilab codes written in it can be downloaded from the "Migrated Labs" section at the website http://scilab.in

Contents

List of Experiments

List of Figures

Experiment: 1

Distance and Connectivity: To understand the notion of connectivity and neighborhood defined for a point in an image.

Scilab code Solution 1.1 Exp1

```
1 //Prog1.Image Arithmetic - To learn to use
    arithmetic operations to combine images.
2 //Software version
3 //OS Windows7
4 //Scilab5.4.1
5 //Image Processing Design Toolbox 8.3.1-1
6 //Scilab Image and Video Proccessing toolbox
    0.5.3.1-2
7 clc;
8 clear;
9 close;
10 I  = imread('C:\Users\senthilkumar\Desktop\
    Gautam_PAL_Lab\DIP_Lab2\cameraman.jpeg'); //SIVP
    toolbox
11 J = imread('C:\Users\senthilkumar\Desktop\
```

```
    Gautam_PAL_Lab\DIP_Lab2\rice.png');//SIVP toolbox
IMA = imadd(I,J); //SIVP toolbox
figure
ShowImage(IMA,'Image Addition')//IPD toolbox
IMS = imabsdiff(I,J);//SIVP toolbox
figure
ShowImage(IMS,'Image Subtraction');//IPD toolbox
IMD = imdivide(I,J);//SIVP toolbox
IMD = imdivide(IMD,0.01);//SIVP toolbox
figure
ShowImage(uint8(IMD),'Image Division');//IPD toolbox
IMM = immultiply(I,I);//SIVP toolbox
figure
ShowImage(uint8(IMM),'Image Multiply');//IPD toolbox
```

Experiment: 2

Image Arithmetic - To learn to use arithmetic operations to combine images.

Scilab code Solution 2.2 Exp2

```
1 //Prog2.Distance and Connectivity: To understand the
      notion of connectivity
2 //and neighborhood defined for a point in an image.
3 //Software version
4 //OS Windows7
5 //Scilab5.4.1
6 //Image Processing Design Toolbox 8.3.1-1
7 //Scilab Image and Video Proccessing toolbox
     0.5.3.1-2
8 clc;
9 clear;
10 close;
11 //[1].Euclidean Distance between images and their
     histograms
12 I = imread('C:\Users\senthilkumar\Desktop\
     Gautam_PAL_Lab\DIP_Lab2\lenna.jpg');
13 J = imread('C:\Users\senthilkumar\Desktop\
```

```
    Gautam_PAL_Lab\DIP_Lab2\cameraman.jpeg')
h_I = CreateHistogram(I);//IPD toolbox
h_J = CreateHistogram(J);//IPD toolbox
I = double(I);
J = double(J);
E_dist_Hist = sqrt(sum((h_I-h_J).^2));//Euclidean
    Distance between histograms of two images
E_dist_images = sqrt(sum((I(:)-J(:)).^2));//
    Euclidean Distance between two images
disp(E_dist_images,'Euclidean Distance between two
    images');
disp(E_dist_Hist,'Euclidean Distance between
    histograms of two images')
//[2]. Connectivity - 8 connected to the background
exec('C:\Users\senthilkumar\Desktop\Gautam_PAL_Lab\
    gray2bin.sci')
Ibin = gray2bin(I);
Jbin = gray2bin(J);
//conversion of gray image into binary image
conn = [1,1,1;1,1,1;1,1,1];//8-connectivity
exec('C:\Users\senthilkumar\Desktop\Gautam_PAL_Lab\
    numdims.sci')
num_dims = numdims(I);
exec('C:\Users\senthilkumar\Desktop\Gautam_PAL_Lab\
    padarray.sci')
B = padarray(Ibin);
global FILTER_ERODE;
StructureElement = CreateStructureElement('square',
    3);
B_eroded = MorphologicalFilter(B,FILTER_ERODE,
    StructureElement.Data);//IPD toolbox
//note:StructureElement.Data and conn both are same
    values
//except that StructureElement.Data is boolean
    either true or false
p = B&~B_eroded;
[m,n] = size(p);
for i = num_dims:m+num_dims-2
```

```
40      for j = num_dims:n+num_dims-2
41          pout(i-1,j-1) = p(i,j);
42      end
43  end
44  figure
45  ShowImage(uint8(I),'Gray Lenna Image')
46  figure
47  ShowImage(Ibin,'Binary Lenna Image')
48  figure
49  ShowImage(pout,'8 neighbourhood connectiviy in Lenna
        Image')
```

---

check Appendix AP 4 for dependency:

`gray2bin.sci`

check Appendix AP 5 for dependency:

`numdims.sci`

check Appendix AP 6 for dependency:

`padarray.sci`

Experiment: 3

Image Arithmetic –To study the effect of these operations on the dynamic range of the output image.

Scilab code Solution 3.3 Exp3

```
1 //Program 3:Image Arithmetic --To study the effect
    of these operations on the dynamic range of the
    output image.
2 //Software version
3 //OS Windows7
4 //Scilab5.4.1
5 //Image Processing Design Toolbox 8.3.1-1
6 //Scilab Image and Video Proccessing toolbox
    0.5.3.1-2
7 clc;
8 clear;
9 close;
10 I = imread('C:\Users\senthilkumar\Desktop\
    Gautam_PAL_Lab\DIP_Lab2\redrose.jpg');
11 J = imread('C:\Users\senthilkumar\Desktop\
```

```
    Gautam_PAL_Lab\DIP_Lab2\mistymorning.jpg');
I = imresize(I,[300,300],'bicubic');
J = imresize(J,[300,300],'bicubic');
K = imadd(I,J)
ShowColorImage(I,'Red Rose Color Image')
figure
ShowColorImage(J,'Misty Morning Color Image')
figure
ShowColorImage(K,'Color Images addition result image')
I_gray = rgb2gray(I);
J_gray = rgb2gray(J);
figure
ShowImage(I_gray,'Red Rose Gray Image')
figure
ShowImage(J_gray,'Misty Morning Gray Image')
Imean = mean2(I_gray);
Jmean = mean2(J_gray);
Ithreshold = double(Imean)/double(max(I_gray(:)));
Jthreshold = double(Jmean)/double(max(J_gray(:)));
I_bw = im2bw(I,Ithreshold);
J_bw = im2bw(J,Jthreshold);
figure
ShowImage(I_bw,'Red Rose Binary Image')
figure
ShowImage(J_bw,'Misty Morning Binary Image')
```

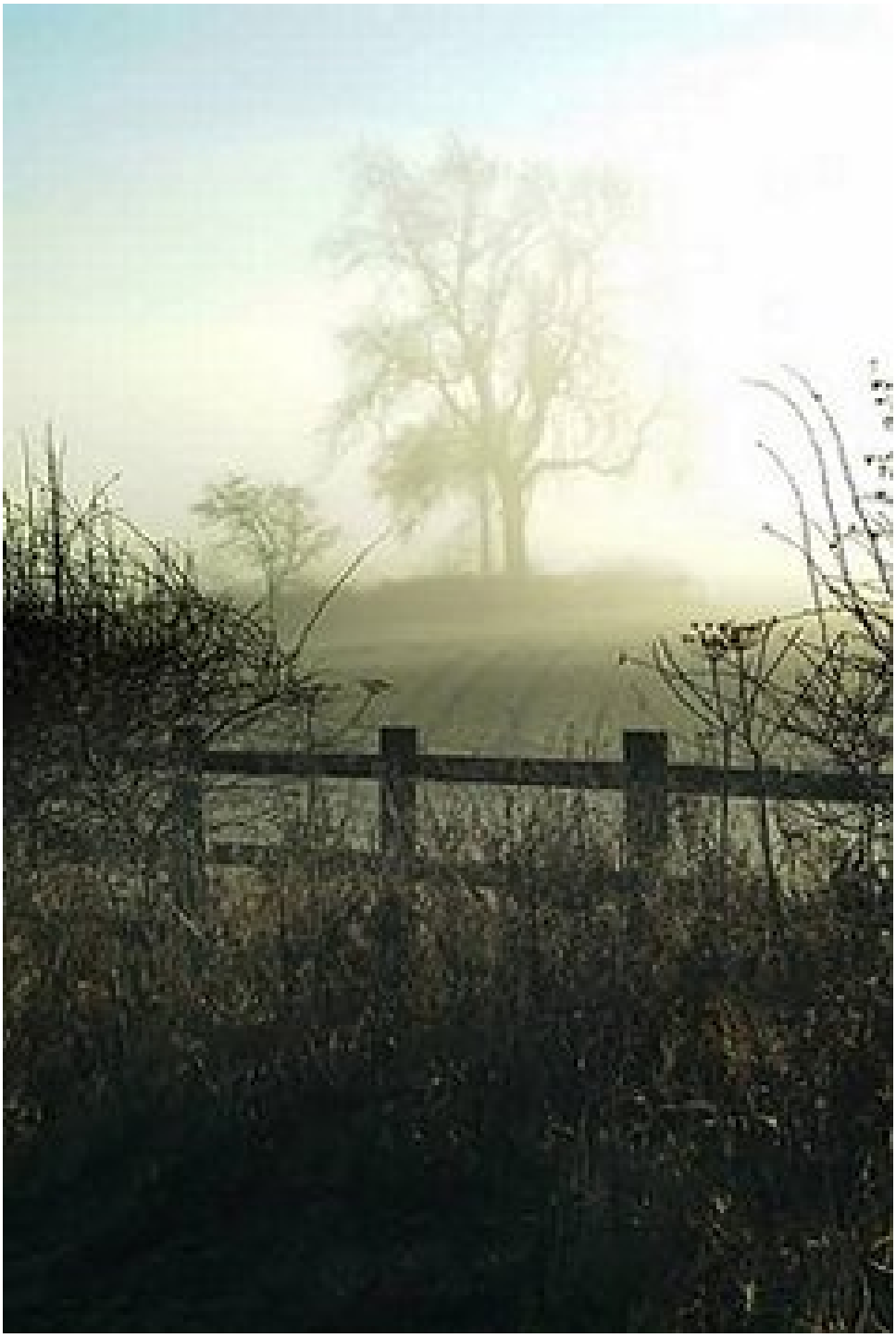

Figure 3.1: Exp3

Figure 3.2: Exp3

Experiment: 4

Image Arithmetic –To study methods to enforce closure forces the output image to also be an 8 bit image.

Scilab code Solution 4.4 Exp4

```
1 //Prog4.Image Arithmetic --To study methods to
    enforce closure forces the output image to also
    be an 8 bit image.
2 //Software version
3 //OS Windows7
4 //Scilab5.4.1
5 //Image Processing Design Toolbox 8.3.1-1
6 //Scilab Image and Video Proccessing toolbox
    0.5.3.1-2
7 clc;
8 clear;
9 close;
10 I = imread('C:\Users\senthilkumar\Desktop\
    Gautam_PAL_Lab\DIP_Lab2\cameraman.jpeg');
11 J = imread('C:\Users\senthilkumar\Desktop\
```

```
    Gautam_PAL_Lab\DIP_Lab2\lenna.jpg');
K = imabsdiff(I,J);
ShowImage(I,'Cameraman Image')
figure
ShowImage(J,'Lenna Image')
figure
ShowImage(K,'Absolute Difference Between cameraman
    and Lenna Image')
L = imcomplement(K);
figure
ShowImage(L,'Complement of difference Image K ')
rect = [20,30,200,200];
I_subimage = imcrop(I,rect);
J_subimage = imcrop(J,rect);
figure
ShowImage(I_subimage,'Sub Image of Cameraman Image')
figure
ShowImage(J_subimage,'Sub Image of Lenna Image')
a=2;
b =0.5;
M =  imlincomb(a,I,b,J);
figure
ShowImage(M,'Linear Combination of cameraman and
    Lenna Image')
N=  imlincomb(b,I,a,J);
figure
ShowImage(N,'Linear Combination of cameraman and
    Lenna Image')
```

Figure 4.1: Exp4

Figure 4.2: Exp4

Experiment: 5

Affine Transformation - To learn basic image transformation i) Translation ii) Rotation iii) Scaling

Scilab code Solution 5.5 Exp5

```
1 //Prog5. Affine Transformation - To learn basic
    image transformation
2 // i) Translation ii) Rotation iii) Scaling
3 //Software version
4 //OS Windows7
5 //Scilab5.4.1
6 //Image Processing Design Toolbox 8.3.1-1
7 //Scilab Image and Video Proccessing toolbox
    0.5.3.1-2
8 clc;
9 clear;
10 clc;
11 I  = imread('C:\Users\senthilkumar\Desktop\
    Gautam_PAL_Lab\DIP_Lab2\lenna.jpg');//size 256
    x256
```

```
[m,n] = size(I);
for i = 1:m
    for j =1:n
        //Scaling
        J(2*i,2*j) = I(i,j);
        //Rotation
        p = i*cos(%pi/2)+j*sin(%pi/2);
        q = -i*sin(%pi/2)+j*cos(%pi/2);
        p = ceil(abs(p)+0.0001);
        q = ceil(abs(q)+0.0001);
        K(p,q)= I(i,j);
        //shear transformation
        u = i+0.2*j;
        v = j;
        L(u,v)= I(i,j);
     end
end
figure
ShowImage(I,'original Image');
figure
ShowImage(J,'Scaled Image');
figure
ShowImage(K,'Rotated Image');
figure
ShowImage(L,'Shear transformed (x direction)Image');
```

Experiment: 6

Affine Transformation –To learn the role of interpolation operation i) Bi-linear ii) Bi-cubic iii) nearest neighbor

Scilab code Solution 6.6 Exp6

```
1 //Prog6. Affine Transformation --To learn the role
    of interpolation operation
2 // i) Bi-linear ii) Bi-cubic iii) nearest neighbor
3 //Software version
4 //OS Windows7
5 //Scilab5.4.1
6 //Image Processing Design Toolbox 8.3.1-1
7 //Scilab Image and Video Proccessing toolbox
    0.5.3.1-2
8 clc;
9 clear;
10 close;
11 I  = imread('C:\Users\senthilkumar\Desktop\
    Gautam_PAL_Lab\DIP_Lab2\lenna.jpg');//size 256
    x256
```

```
12 [m,n] = size(I);
13 for i = 1:m
14     for j =1:n
15         //Scaling
16         J(1.5*i,1.5*j) = I(i,j); //512x512 Image
17     end
18 end
19 I_nearest = imresize(J,[256,256]); //'nearest' -
      nearest-neigbor interpolation
20 I_bilinear = imresize(J,[256,256],'bilinear');// '
      bilinear' - bilinear interpolation
21 I_bicubic = imresize(J,[256,256],'bicubic');//'
      bicubic' - bicubic interpolation
22 figure
23 ShowImage(uint8(I_nearest),'nearest-neigbor
      interpolation');
24 figure
25 ShowImage(uint8(I_bilinear),'bilinear - bilinear
      interpolation');
26 figure
27 ShowImage(uint8(I_bicubic),'bicubic - bicubic
      interpolation');
```

Experiment: 7

Affine Transformation –To learn the effect of multiple transformations i) Significance of order in which one carried out

Scilab code Solution 7.7 Exp7

```
1  //Prog7.Affine Transformation --To learn the effect
       of multiple transformations i) Significance of
       order in which one carried out
2  //Software version
3  //OS Windows7
4  //Scilab5.4.1
5  //Image Processing Design Toolbox 8.3.1-1
6  //Scilab Image and Video Proccessing toolbox
       0.5.3.1-2
7  clc;
8  clear;
9  close
10 I  = imread('C:\Users\senthilkumar\Desktop\
```

```
    Gautam_PAL_Lab\DIP_Lab2\lenna.jpg');
[m,n] = size(I);
for i = 1:m
    for j =1:n
        //shear transformation and rotation
        u = i+0.2*j;
        v = 0.3*i+j;
        M(u,v) = I(i,j);
        //shear transformation, rotation and scaling
        N(u*1.5,v*1.5)= I(i,j);
    end
end
figure
ShowImage(I,'original Lenna Image')
figure
ShowImage(M,'Shear transformed+rotated Lenna Image')
figure
ShowImage(N,'Shear Transformed+rotated+scaled Lenna
    Image')
```

Experiment: 8

Point Operations - To learn image enhancement through point transformation-i)Linear transformation ii) Non-linear transformation

Scilab code Solution 8.8 Exp8

```
1 //Prog8.Point Operations - To learn image
    enhancement through point transformation
2 //i)Linear transformation ii) Non-linear
    transformation
3 //Software version
4 //OS Windows7
5 //Scilab5.4.1
6 //Image Processing Design Toolbox 8.3.1-1
7 //Scilab Image and Video Proccessing toolbox
    0.5.3.1-2
8 clc;
9 clear;
10 close;
```

```
11 I = imread('C:\Users\senthilkumar\Desktop\
      Gautam_PAL_Lab\DIP_Lab2\rice.png');
12 //(i). Linear Transformation
13 //IMAGE NEGATIVE
14 I = double(I);
15 J = 255-I;
16 figure
17 ShowImage(I,'Original Image')
18 figure
19 ShowImage(J,'Linear Transformation-IMAGE NEGATIVE')
20 //(ii) Non-linear transformation
21 //GAMMA TRANSFORMATION
22 GAMMA = 0.9;
23 K = I.^GAMMA;
24 figure
25 ShowImage(K,'Non-linear transformation-GAMMA
      TRANSFORMATION')
```

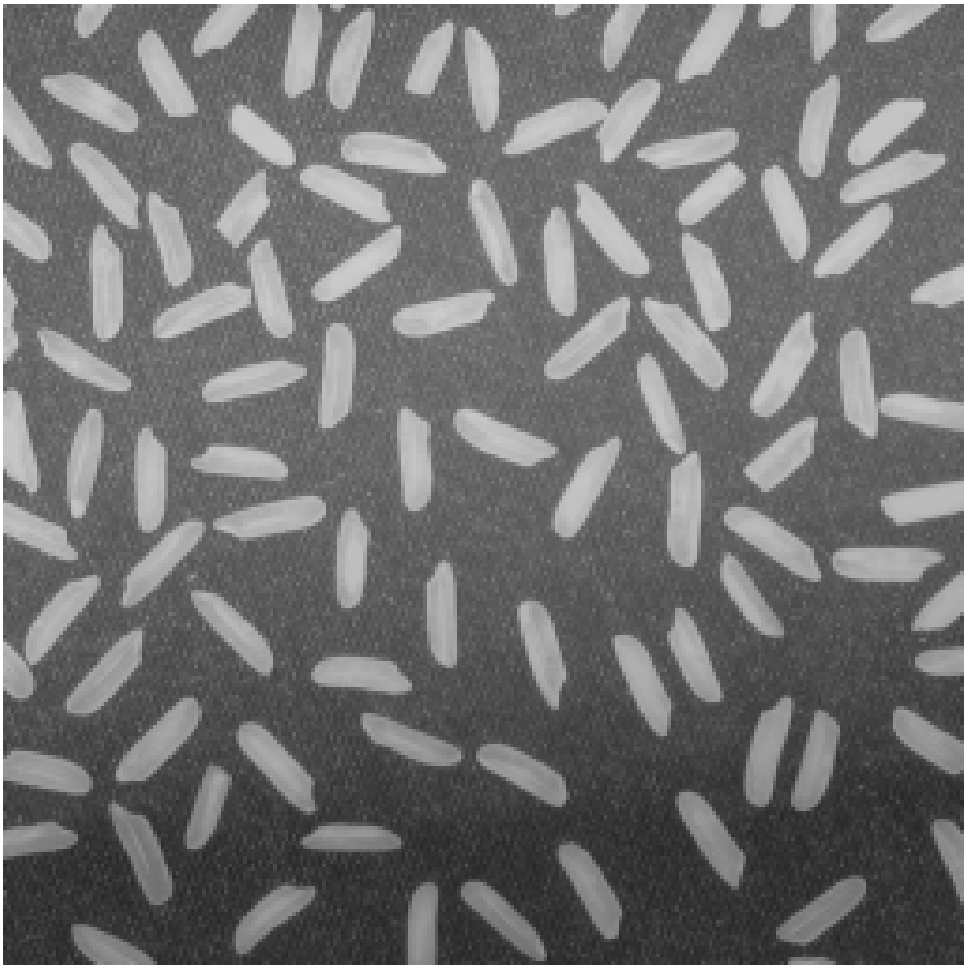

Figure 8.1: Exp8

Experiment: 9

Neighborhood Operations - To learn about neighborhood operations and use them for i) Linear filtering ii) Non-linear filtering

Scilab code Solution 9.9 Exp9

```
1 //Prog9.Neighborhood Operations - To learn about
     neighborhood operations and use them for
2 //i) Linear filtering ii) Non-linear filtering
3 //Software version
4 //OS Windows7
5 //Scilab5.4.1
6 //Image Processing Design Toolbox 8.3.1-1
7 //Scilab Image and Video Proccessing toolbox
     0.5.3.1-2
8 clc;
9 clear;
10 close;
11 I = imread('C:\Users\senthilkumar\Desktop\
```

```
    Gautam_PAL_Lab\DIP_Lab2\lenna.jpg');
I_noise = imnoise(I,'salt & pepper');
figure
ShowImage(I,'Original Lenna Image')
figure
ShowImage(I_noise,'Noisy Lenna Image')
//Case 1: Linear Filtering
//(i).Linear Filtering -Example 1: Average Filter
F_linear1 = 1/25*ones(5,5);//5x5 mask
I_linear1 = imfilter(I_noise,F_linear1);//linear
    filtering -Average Filter
figure
ShowImage(I_linear1,'Linear Average Filtered Noisy
    Lenna Image')
//(ii).Linear Filtering -  Example 2: Gaussing
    filter
hsize = [5,5];
sigma = 1;
F_linear2 = fspecial('gaussian', hsize, sigma); //
    Linear filtering- gaussian Filter
I_linear2 = imfilter(I_noise,F_linear2);
figure
ShowImage(I_linear2,'Linear Gaussian Filtered Noisy
    Lenna Image')
//Case 2: Non-Linear Filtering
//(i).Median Filtering
F_NonLinear = [3,3];
I_NonLinear =  MedianFilter(I_noise,F_NonLinear);//
    Median Filter 3x3
figure
ShowImage(I_NonLinear,'Median Filtered(Non-Linear)
    Noisy Lenna Image')
```

Experiment: 10

Neighborhood Operations –To study the effect of the size of neighborhood on the result of processing

Scilab code Solution 10.10 Exp10

```
1 //Prog10.Neighborhood Operations --To study the
    effect of the size of neighborhood on the result
    of processing
2 //Software version
3 //OS Windows7
4 //Scilab5.4.1
5 //Image Processing Design Toolbox 8.3.1-1
6 //Scilab Image and Video Proccessing toolbox
    0.5.3.1-2
7 clc;
8 clear;
9 close;
10 I = imread('C:\Users\senthilkumar\Desktop\
    Gautam_PAL_Lab\DIP_Lab2\lenna.jpg');
11 I_noise = imnoise(I,'salt & pepper');
```

```
12 FilterSize = [3 3]; //filter size 3x3
13 I_3x3 = MedianFilter(I_noise,FilterSize);
14 I_5x5 = MedianFilter(I_noise,[5 5]);
15 I_7x7 = MedianFilter(I_noise,[7 7]);
16 I_9x9 = MedianFilter(I_noise,[9 9]);
17 figure
18 ShowImage(I,'Original Lenna Image')
19 figure
20 ShowImage(I_noise,'Original Lenna Image')
21 figure
22 ShowImage(I_3x3,'Filtered Lenna Image-Filter size 3
      x3')
23 figure
24 ShowImage(I_5x5,'Filtered Lenna Image-Filter size 5
      x5')
25 figure
26 ShowImage(I_7x7,'Filtered Lenna Image-Filter size 7
      x7')
27 figure
28 ShowImage(I_9x9,'Filtered Lenna Image-Filter size 9
      x9')
```

Experiment: 11

Image Histogram - To understand how frequency distribution can be used to represent an image.

Scilab code Solution 11.11 Exp11

```
1 //Prog11.Image Histogram - To understand how
    frequency distribution can be used to represent
    an image
2 //Software version
3 //OS Windows7
4 //Scilab5.4.1
5 //Image Processing Design Toolbox 8.3.1-1
6 //Scilab Image and Video Proccessing toolbox
    0.5.3.1-2
7 clc;
8 clear;
9 close;
10 I = imread('C:\Users\senthilkumar\Desktop\
    Gautam_PAL_Lab\DIP_Lab2\pout.png');
11 [count, cells]= imhist(I);
```

```
12 scf(0)
13 ShowImage(I,'Original Image pout.png')
14 scf(1);
15 plot2d3('gnn',cells,count)
16 title('Histogram Plot of Original Image')
17 exec('C:\Users\senthilkumar\Desktop\Gautam_PAL_Lab\
     histeq.sci');
18 Iheq = histeq(I);
19 [count, cells]= imhist(Iheq);
20 scf(2)
21 ShowImage(Iheq,'Histogram Equalized Image pout.png')
22 scf(3)
23 plot2d3('gnn',cells,count)
24 title('Histogram of Histogram Equalized Image')
```

---

check Appendix AP 3 for dependency:

`histeq.sci`

Figure 11.1: Exp11

Experiment: 12

Image Histogram –To study the correlation between the visual quality of an image with its histogram.

Scilab code Solution 12.12 Exp12

```
1 //Prog12.Image Histogram --To study the correlation
    between the visual quality of an image with its
    histogram.
2 //Software version
3 //OS Windows7
4 //Scilab5.4.1
5 //Image Processing Design Toolbox 8.3.1-1
6 //Scilab Image and Video Proccessing toolbox
    0.5.3.1-2
7 clc;
8 clear;
9 close;
10 I = imread('C:\Users\senthilkumar\Desktop\
    Gautam_PAL_Lab\DIP_Lab2\pout.png');
11 I = imresize(I,[256,256]);
```

```
12 [count, cells]= imhist(I);
13 exec('C:\Users\senthilkumar\Desktop\Gautam_PAL_Lab\
     DIP_Lab2\histeq.sci');
14 Iheq = histeq(I);
15 [count1, cells1]= imhist(Iheq);
16 Corr_Bet_Same_Images = corr2(I,Iheq);
17 disp(Corr_Bet_Same_Images,'Correlation between
     original Image and Its Histogram equalized Image'
     )
18 J = imread('C:\Users\senthilkumar\Desktop\
     Gautam_PAL_Lab\cameraman.jpeg');
19 Corr_Bet_Diff_Images = corr2(Iheq,J);
20 disp(Corr_Bet_Diff_Images,'Correlation between pout.
     png and cameraman.jpeg images')
21 x = xcorr(count,count); //correlation of histogram
     of the same
22 x1 = xcorr(count,count1);//correlation of histogram
     of original image and its histogram equalized
     image
23 scf(0)
24 plot2d3('gnn',1:length(x),x,5)
25 title('correlation between histograms of original
     image')
26 scf(1)
27 plot2d3('gnn',1:length(x1),x1,5)
28 title('correlation between histograms of original
     image and its histogram equalized image')
29 //RESULT
30 //Correlation between original Image and Its
     Histogram equalized Image
31 //
32 //    0.9784662
33 //
34 // Correlation between pout.png and cameraman.jpeg
     images
35 //
36 //  - 0.3204259
37 //
```

check Appendix AP 3 for dependency:

`histeq.sci`

Experiment: 13

Fourier Transform: To understand some of the fundamental properties of the Fourier transform.

Scilab code Solution 13.13 Exp13

```
1 //Prog13.Fourier Transform: To understand some of
    the fundamental properties of the Fourier
    transform.
2 //Software version
3 //OS Windows7
4 //Scilab5.4.1
5 //Image Processing Design Toolbox 8.3.1-1
6 //Scilab Image and Video Proccessing toolbox
    0.5.3.1-2
7 clc;
8 clear;
9 close;
10 I  = imread('C:\Users\senthilkumar\Desktop\
    Gautam_PAL_Lab\DIP_Lab2\lenna.jpg');
11 exec('C:\Users\senthilkumar\Desktop\Gautam_PAL_Lab\
```

```
    DIP_Lab2\fft2d.sci');
exec('C:\Users\senthilkumar\Desktop\Gautam_PAL_Lab\
    DIP_Lab2\ifft2d.sci');
//[1].2D-DFT and its Inverse 2D-DFT
I = double(I);
J = fft2d(I);
K = real(ifft2d(J));
figure
ShowImage(I,'Original Lenna Image')
figure
ShowImage(abs(J),'2D DFT (spectrum) of Lenna Image')
figure
ShowImage(K,'2d IDFT of Lenna Image')
//[2]. Two times fftshift results in original
    spectrum
L = fftshift(J);
M = fftshift(L);
figure
ShowImage(abs(L),'fftshited spectrum of Lenna Image'
    )
figure
ShowImage(abs(M),'two times fftshifted')
```

---

check Appendix AP 1 for dependency:

`fft2d.sci`

check Appendix AP 2 for dependency:

`ifft2d.sci`

Experiment: 14

Colour Image Processing: To learn colour images are handled and processed i)Models for representing colour ii) Methods of proces

Scilab code Solution 14.14 Exp14

```
1 //Prog14. Colour Image Processing: To learn colour
    images are handled and processed
2 //i)Models for representing colour ii) Methods of
    proces
3 //Software version
4 //OS Windows7
5 //Scilab5.4.1
6 //Image Processing Design Toolbox 8.3.1-1
7 //Scilab Image and Video Proccessing toolbox
    0.5.3.1-2
8 clc;
9 clear;
10 close;
```

```
RGB = imread('C:\Users\senthilkumar\Desktop\
   Gautam_PAL_Lab\DIP_lab2\football.jpg');
figure
ShowColorImage(RGB,'RGB Color Image')
YIQ = rgb2ntsc(RGB);
figure
ShowColorImage(YIQ,'NTSC image YIQ')
RGB = ntsc2rgb(YIQ);
YCC = rgb2ycbcr(RGB);
figure
ShowColorImage(YCC,'equivalent HSV image YCbCr')
RGB = ycbcr2rgb(YCC);
HSV = rgb2hsv(RGB);
figure
ShowColorImage(HSV,'equivalent HSV image')
RGB = hsv2rgb(HSV);
R = RGB(:,:,1);
G = RGB(:,:,2);
B = RGB(:,:,3);
figure
ShowImage(R,'Red Matrix')
figure
ShowImage(G,'Green Matrix')
figure
ShowImage(B,'Blue Matrix')
```

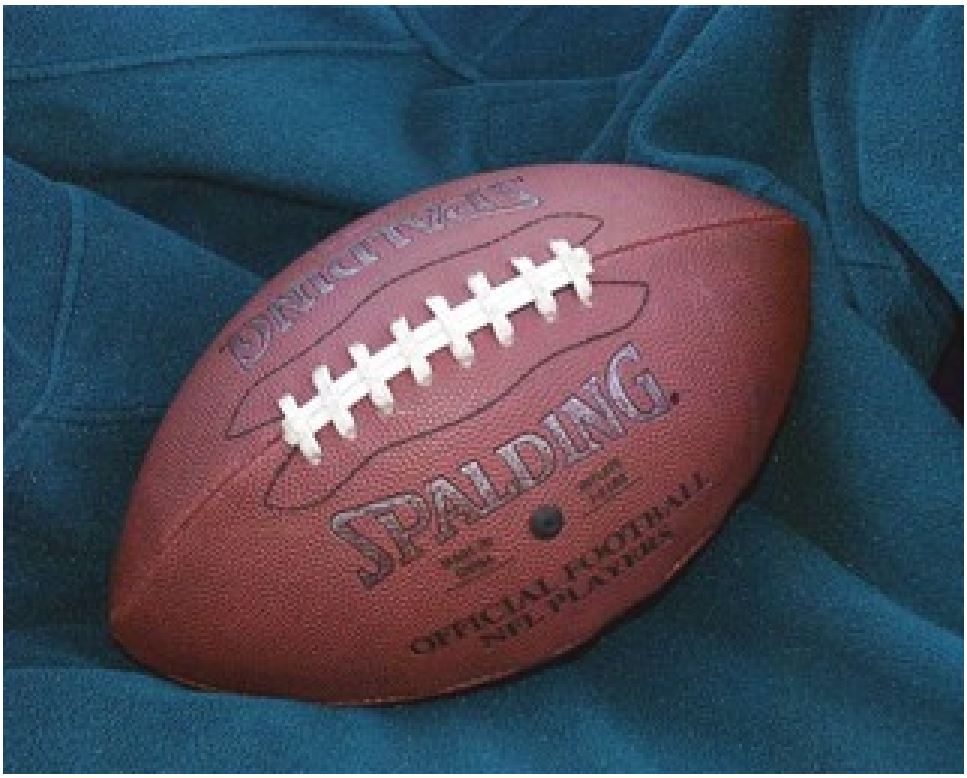

Figure 14.1: Exp14

Experiment: 15

Morphological Operations: To understand the basics of morphological operations which are used in analyzing the form and shape de

Scilab code Solution 15.15 Exp15

```
1 //Prog15.Morphological Operations: To understand the
      basics of morphological operations
2 //which are used in analyzing the form and shape
3 //Software version
4 //OS Windows7
5 //Scilab5.4.1
6 //Image Processing Design Toolbox 8.3.1-1
7 //Scilab Image and Video Proccessing toolbox
     0.5.3.1-2
8 clc;
9 clear;
10 close;
11 Image = imread('C:\Users\senthilkumar\Desktop\
```

```
    Gautam_PAL_Lab\DIP_Lab2\tire.jpeg');
12 StructureElement = CreateStructureElement('square'
    ,3); // generate structuring element IPD atom
13 ResultImage1 = ErodeImage(Image,StructureElement);
    //IPD Atom
14 ResultImage2 = DilateImage(Image,StructureElement);
    //IPD Atom
15 ResultImage3 =  BottomHat(Image,StructureElement);
    //IPD Atom
16 ResultImage4 =  TopHat(Image,StructureElement); //
    IPD Atom
17 figure
18 ShowImage(Image,'Original Image')
19 figure
20 ShowImage(ResultImage1,'Eroded Image')
21 figure
22 ShowImage(ResultImage2,'Dilated Image')
23 figure
24 ShowImage(ResultImage3,'bottom hat filtered image')
25 figure
26 ShowImage(ResultImage4,'top hat filtered image')
27
28 ResultImage5 = imadd(ResultImage3,ResultImage4);
29 figure
30 ShowImage(ResultImage4,'top hat filtered image+
    bottom hat filtered image')
```

Figure 15.1: Exp15

Appendix

Scilab code AP 1

```
function [a2] = fft2d(a)
//a = any real or complex 2D matrix
//a2 = 2D-DFT of 2D matrix  'a'
m=size(a,1)
n=size(a,2)
// fourier transform along the rows
for i=1:n
a1(:,i)=exp(-2*%i*%pi*(0:m-1)'.*.(0:m-1)/m)*a(:,i)
end
// fourier transform along the columns
for j=1:m
a2temp=exp(-2*%i*%pi*(0:n-1)'.*.(0:n-1)/n)*(a1(j,:))
   .'
a2(j,:)=a2temp.'
end
for i = 1:m
    for j = 1:n
        if((abs(real(a2(i,j)))<0.0001)&(abs(imag(a2(
           i,j)))<0.0001))
            a2(i,j)=0;
        elseif(abs(real(a2(i,j)))<0.0001)
            a2(i,j)= 0+%i*imag(a2(i,j));
        elseif(abs(imag(a2(i,j)))<0.0001)
            a2(i,j)= real(a2(i,j))+0;
        end
    end
end
```

2D Fast Fourier Trasnform

Scilab code AP 12

```
function [a] =ifft2d(a2)
//a2 = 2D-DFT of any real or complex 2D matrix
//a = 2D-IDFT of a2
m=size(a2,1)
n=size(a2,2)
//Inverse Fourier transform along the rows
for i=1:n
a1(:,i)=exp(2*%i*%pi*(0:m-1)'.*.(0:m-1)/m)*a2(:,i)
end
//Inverse fourier transform along the columns
for j=1:m
atemp=exp(2*%i*%pi*(0:n-1)'.*.(0:n-1)/n)*(a1(j,:)).'
a(j,:)=atemp.'
end
a = a/(m*n)
a = real(a)
endfunction
```

2D Inverse Fast Fourier Transform

Scilab code AP 13

```
function [hea,b]=histeq(a)
    //a- original image
    //b- histogram
    //hea- histogram equalized image
    [m n]=size(a);
    for i=1:256
        b(i)=length(find(a==(i-1)));
    end
    pbb=b/(m*n);
    pb(1)=pbb(1);
    for i=2:256
        pb(i)=pb(i-1)+pbb(i);
    end

    s=pb*255;
```

```
16      sb=uint8(round(s));
17      index =0;
18      for i=1:m
19          for j=1:n
20              index = double(a(i,j))+1;//convert it to
                    double
21              //otherwise index = 255+1 =0
22              hea(i,j)= sb(index);//histogram
                   equalization
23          end
24      end
25  endfunction
```

Histogram Equalization

Scilab code AP 14

```
1   function X = gray2bin(x)
2       xmean = mean2(x);
3       [m,n]= size(x);
4       X = zeros(m,n);
5       for i = 1:m
6           for j = 1:n
7               if x(i,j)> xmean then
8                   X(i,j) = 1;
9               end
10          end
11      end
12  endfunction
```

Gray Pixel value to Binary value

Scilab code AP 15

```
1   function n = numdims(X)
2       n = length(size(X));
3   endfunction
```

Total number of pixels in an image

Scilab code AP 16

```
1   function B = padarray(b)
```

```
2       //pad zeros in columns and rows at both ends of
            an binary image
3   [m,n] = size(b);
4   num_dims = length(size(b));
5   B = zeros(m+num_dims,n+num_dims);
6   for i = num_dims:m+num_dims-1
7       for j = num_dims:m+num_dims-1
8           B(i,j) = b(i-1,j-1);
9       end
10  end
11  endfunction
```

Pad Array