

Scilab Manual for
Image Processing
by Mr Biren Patel
Others
Dharmsinh Desai University¹

Solutions provided by
Mr Biren Patel
Others
D.D. University

May 14, 2026

¹Funded 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 Scilab Solutions	3
1 Digital Image Fundamentals	4
2 Image Enhancement in the Spatial Domain	8
3 Filtering In the Spatial Domain	16
4 Filtering The Frequency Domain	23
5 Image Restoration	37
6 Color Image Processing	54
7 Image Compression	63
8 Morphological Image Processing	70
9 Image Segmentation	76
10 Wavelets	81

List of Experiments

Solution 1.1	Digital Image Fundamentals	4
Solution 2.1	Image Enhancement in the Spatial Domain	8
Solution 3.1	Filtering In the Spatial Domain	16
Solution 4.1	Filtering In the Frequency Domain	23
Solution 5.1	Image Restoration	37
Solution 5.2	Image Restoration	43
Solution 6.1	Color Image Processing Fundamentals	54
Solution 6.2	Color Image Processing	58
Solution 7.1	Image Compression	63
Solution 7.2	Image Compression	66
Solution 7.3	Image Compression	68
Solution 8.1	Morphological Image Processing	70
Solution 9.1	Image Segmentation	76
Solution 10.1	Wavelets	81
Solution 10.2	Wavelets	87
Solution 10.3	Wavelets	92

Experiment: 1

Digital Image Fundamentals

Scilab code Solution 1.1 Digital Image Fundamentals

```
1
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //
6 //
7 clc //to clear command window.
8 clear all //to kill previously defined variables.
9 xdel(winsid())//to close all currently open figure(s
   ).
10
11
12
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed.getSIVPpath()
   is pre-defined function in SIVP toolbox.
14 i = imread(SIVP_PATH + 'images/lena.png');//Reading
   from sub-directory of images.
15
16 b=rgb2gray(i);//to convert an image into grayscale
   image.
```

```

17 imwrite(b,'3.jpg');//to write an image.
18 info=imfinfo('3.jpg');//to get information like size
    , width ,height etc using imfinfo() function.
19 disp(info.Width);//imfinfo store data into info
    variable and disp() is function to use for
    displaying content of argument in Scilab console.
20 disp(info.FileName);
21 disp(info.FileSize);
22 disp(info.Width);
23 disp(info.Height);
24 disp(info.BitDepth);
25 disp(info.ColorType);
26
27 SIVP_PATH = getSIVPpath();//to locate a directory
    in which SIVP toolbox is installed.getSIVPpath()
    is pre-defined function in SIVP toolbox.
28 i = imread(SIVP_PATH + 'images/lena.png');//Reading
    from sub-directory of images
29 ShowColorImage(i,'Original Image');//To show a color
    image in graphical window using "ShowColorImage
    ()" function .one can use imshow function which
    uses tk image show widget for displaying an image
    .
30 title('Original Image');//title() is used for
    providing a title to an image.
31
32
33 im=imresize(i,1.3);//To resize an image,1.3 is
    scale factor
34 ShowColorImage(im,'Resized Image');//To show an
    image in graphical window using "ShowImage()"
    function .one can use imshow function which use
    tk image show widget for displaying an image.
35 title('Resized Image');// title() is used for
    providing a title to an image.
36
37 im1=imresize(i,1.3,'bilinear');//to resize an image
    .Third parameter is used for selecting one

```

```

    method.
38 ShowColorImage(im1,'Resized Image using bilinear
    interpolation');//To show an image in graphical
    window using "ShowImage()" function .one can use
    imshow function which use tk image show widget
    for displaying an image.
39 title('Resized Image using bilinear interpolation');
    // title() is used for providing a title to an
    image.
40
41
42 im2=imresize(i,1.3,'bicubic');//to resize an image
43 ShowColorImage(im2,'Resized Image using bicubic
    interpolation');//To show an image in graphical
    window using "ShowImage()" function .one can use
    imshow function which use tk image show widget
    for displaying an image.
44 title('Resized Image using bicubic interpolation');
    // title() is used for providing a title to an
    image.
45
46
47 im3=rgb2gray(i);//To convert an image into
    grayscale image.
48 figure,ShowImage(im3,'Grayscale image');//figure is
    used to display images in separate window.
49 title('Grayscale Image');//title() is used for
    providing a title to an image.
50
51 subimage=imcrop(im2,[1,1,256,256]);//subimage is
    image left after cropping at 1,1 top left corner.
    Width of an image is 256 and height of an image
    is 256.
52 figure,ShowColorImage(subimage,'Cropped image');//
    figure is used to display images in separate
    window.
53 title('Cropped Image');//title() is used for
    providing a title to an image.

```


Experiment: 2

Image Enhancement in the Spatial Domain

check Appendix ?? for dependency:

21.tif

check Appendix ?? for dependency:

log.tif

check Appendix ?? for dependency:

pollen.tif

check Appendix ?? for dependency:

pollensmall.tif

Scilab code Solution 2.1 Image Enhancement in the Spatial Domain

```
1
2 //
3 //environment: Scilab 5.4.1
4 //Toolbox: Image Processing Design 8.3.1-1
5 //Toolbox: SIVP 0.5.3.1-2
```

```

6 //Toolbox:Scilab Wavelet Toolbox0.1.19-1
7 //Toolbox:Huffcomp Toolbox 1.1.1
8 //OS:Windows 7
9 //
10 //Reference book name : Digital Image Processing
11 //book author: Rafael C. Gonzalez and Richard E.
    Woods
12 clc //to clear command window.
13 clear all //to kill previously defined variables.
14 xdel(winsid())//to close all currently open figure(s
    ).
15
16
17 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed. getSIVPpath()
    is pre-defined function in SIVP toolbox.
18 rgb = imread(SIVP_PATH + 'images/lena.png');//
    Reading from sub-directory of images.
19 figure, ShowColorImage(rgb, 'Original color image');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
20 title('Original Color Image');//title() is used for
    providing a title to an image.
21
22 grayimage=rgb2gray(rgb);//to convert a RGB image
    into grayscale image.
23
24 figure, ShowImage(grayimage, 'Grayscale image');//
    ShowImage() is used to show image, figure is
    command to view images in separate window.
25 title('Grayscale image');//title() is used for
    providing a title to an image.
26
27 function [negative]=imagenegative(b)//imagenegative
    () is function to find negative of an image.
28 [nr nc]=size(b)//to find dimension of a
    grayscale image.

```

```

29     for i = 1:nr
30         for j = 1:nc
31             negative(i,j)=255-b(i,j);
32         end;
33     end
34 endfunction
35
36 negative=imagenegative(grayimage);
37 figure,ShowImage(negative,'Negative image');//
    ShowImage() is used to show image, figure is
    command to view images in separate window.
38 title('Negative image');//title() is used for
    providing a title to an image.
39
40 aaa=ReadImage('21.tif');//ReadImage() is function
    defined in IPD toolbox.
41 figure,ShowImage(aaa,'Intensity Ramp Image');//
    ShowImage() is used to show image, figure is
    command to view images in separate window.
42 title('Intensity Ramp Image');//title() is used for
    providing a title to an image.
43
44 graydouble=double(aaa);//to convert image into
    double precision.
45
46 function [g1]=imadjust(b,lowin,lowout,highin,highout
    ,gamma1)//imadjust() is used for gamma correction
    .
47     [nr nc]=size(b);
48     for i = 1:nr
49         for j = 1:nc
50             g1(i,j)=lowout+(highout-lowout)*((b(i,j)
                -lowin)/(highin-lowin))^gamma1;
51         end;
52     end
53 endfunction
54
55 s1=imadjust(graydouble,0.0392,0.7843,0.589,1,2);

```

```

56 d=mat2gray(s1); // to convert matrix into grayscale
    image.
57 figure, ShowImage(d, 'Result of enhancing image with
    gamma=2'); // ShowImage() is used to show image,
    figure is command to view images in separate
    window.
58 title('Result of enhancing image with gamma=2'); //
    title() is used for providing a title to an
    image.
59
60
61 aaa=ReadImage('log.tif'); // ReadImage() is function
    defined in IPD toolbox.
62 figure, ShowImage(aaa, 'Image'); // ShowImage() is used
    to show image, figure is command to view images
    in separate window.
63 title('Image'); // title() is used for providing a
    title to an image.
64
65 graydouble=double(aaa); // to convert image into
    double precision.
66 function [log1]=logtransform(b,c); // to perform log
    transformations on an image
67     [nr nc]=size(b);
68     for i = 1:nr
69         for j = 1:nc
70             log1(i,j)=c*log(1+b(i,j));
71         end
72     end
73 endfunction
74
75 s2=logtransform(graydouble,1);
76
77 ans1=255*uint8(s2);
78 figure, ShowImage(ans1, 'Result of applying log
    transformations with c=1'); // ShowImage() is used
    to show image, figure is command to view images
    in separate window.

```

```

79 title('Result of applying log transformations with c
      =1'); //title() is used for providing a title to
      an image.
80
81
82
83
84 aaa=ReadImage('pollen.tif'); //ReadImage() is
      function defiened in IPD toolbox.
85 figure, ShowImage(aaa, 'washed_out_pollen_image'); //
      ShowImage() is used to show image, figure is
      command to view images in separate window.
86 title('washed_out_pollen_image'); //title() is used
      for providing a title to an image.
87 bb=double(aaa);
88
89 function [ans]=contraststretch(b,r1,s1,r2,s2); //to
      perform contrast stretching on a image.
90 [nr nc]=size(b);
91 m1=s1/r1;
92 m2=(r2-r1)/(s2-s1);
93 m3=(255-s2)/(255-r2);
94 for i = 1:nr
95     for j = 1:nc
96         if(b(i,j)<=r1) then
97             ans(i,j)=m1*b(i,j);
98         end
99
100        if(b(i,j)>r1) then
101            if(b(i,j)<=r2) then
102                ans(i,j)=m2*b(i,j)+(r2*s1-r1*s2)/(r2-
                    r1);
103            end
104        end
105        if(b(i,j)>r2) then
106            ans(i,j)=m3*b(i,j)+(255*s2-r2*255)
                    /(255-r2);
107        end

```

```

108         end
109     end
110 endfunction
111 s2=contraststretch(bb,70,40,150,200);
112 ans2=mat2gray(s2);
113 figure,ShowImage(ans2,'Result of contrast stretching
    ');//ShowImage() is used to show image, figure is
    command to view images in separate window.
114 title('Result of contrast stretching');//title() is
    used for providing a title to an image.
115
116
117 aa=ReadImage('pollensmall.tif');//To read an image.
118 a=rgb2gray(aa);//to convert an image into grayscale
119 b=double(a);
120 [nr,nc]=size(a)
121
122
123 [count, cells]=imhist(a);//imhist() is used to
    obtain histogram.
124 scf(10);imhist(a,256,'');//scf() is used to set
    current graphic window.
125
126
127
128 bit=8;           //l is variable to used find
    possible intensity levels of an image.
129 l=2^bit;
130
131 r=zeros(1,l);//rK is used for input intensity levels
    for an image.
132 n=zeros(1,l);//nK is used for number of pixels,
    which have different intensity levels.
133 p=zeros(1,l);//pk is probability of occurrence of
    intensity level rk in a digital image.
134
135 for i = 1:l//for loop is used to define different
    intensity levels.

```

```

136     r(1,i)=i-1;
137 end
138
139
140 for k=1:l//for loop is used to find occurrence of rk
    intensity level
141     for i = 1:nr
142         for j = 1:nc
143             if a(i,j)==r(1,k) then
144                 n(1,k)=n(1,k)+1;
145             end
146         end
147     end
148 end
149
150 p=n/(nr*nc);//to find probability of occurrence of
    intensity level rk in a digital image.
151
152
153 for k=1:l//for loop is used to find transformation's
    result.
154     temp=0;
155     for j=1:k
156         temp=temp+255*p(1,j);
157     end
158     if(ceil(temp)-temp>0.5) then
159         s(1,k)=floor(temp);
160     else
161         s(1,k)=ceil(temp);
162     end
163 end
164
165 for k=1:l//for loop is used to assign new intensity
    levels.
166     for i = 1:nr
167         for j = 1:nc
168             if (b(i,j)==r(1,k)) then
169                 dd(i,j)= s(1,k);

```

```
170             end
171         end
172     end
173 end
174 ddd=uint8(dd);
175 [count, cells]=imhist(ddd,256);//imhist() is used to
    get 256 bins of histogram.
176 clf;
177 scf(11);imhist(ddd,256);//To show histogram
178 scf(12);ShowImage(ddd,' Result of histogram
    equalization ');//ShowImage() is used to show
    image, figure is command to view images in
    separate window.
179 title('Result of histogram equalization');//title()
    is used for providing a title to an image.
```

Experiment: 3

Filtering In the Spatial Domain

Scilab code Solution 3.1 Filtering In the Spatial Domain

```
1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox:Huffcomp Toolbox 1.1.1
7 //OS:Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11 clc //to clear command window.
12 clear all //to kill previously defined variables.
13 xdel(winsid())//to close all currently open figure(s
    ).
14
15
16 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed. getSIVPpath()
    is pre-defined function in SIVP toolbox.
17 rgb = imread(SIVP_PATH + 'images/lena.png');//
```

```

    Reading from sub-directory of images.
18 figure,ShowColorImage(rgb,'Original color image');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
19 title('Original Color Image');//title() is used for
    providing a title to an image.
20
21 im=rgb2gray(rgb);//to convert a RGB image into
    grayscale image.
22
23 figure,ShowImage(im, 'Grayscale image');//ShowImage()
    is used to show image, figure is command to view
    images in separate window.
24 title('Grayscale image');//title() is used for
    providing a title to an image.
25
26 filter=fspecial('average',3);//fspecial() is used to
    create special filters.
27 imf = imfilter(im, filter);
28 figure,ShowImage(imf, 'Filtered Grayscale image using
    3*3 average mask');//ShowImage() is used to show
    image, figure is command to view images in
    separate window.
29 title('Filterd Grayscale image using 3*3 average
    mask');//title() is used for providing a title
    to an image.
30
31
32 filter=fspecial('average',9);//fspecial() is used to
    create special filters.
33 imf = imfilter(im, filter);
34 figure,ShowImage(imf, 'Filterd Grayscale image using
    9*9 average mask');//ShowImage() is used to show
    image, figure is command to view images in
    separate window.
35 title('Filterd Grayscale image using 9*9 average
    mask');//title() is used for providing a title

```

```

    to an image.
36
37
38 filter=fspecial('average',16);//fspecial() is used
    to create special filters.
39 imf = imfilter(im, filter);
40 figure,ShowImage(imf, 'Filterd Grayscale image using
    16*16 average mask');//ShowImage() is used to
    show image, figure is command to view images in
    separate window.
41 title('Filterd Grayscale image using 16*16 average
    mask');//title() is used for providing a title
    to an image.
42
43
44 imn = imnoise(im, 'gaussian',0.01,0.02);//imnoise()
    is used to add noise in an image.'gaussian' is
    used as a second argument to function for adding
    Gasussian noise
45 figure,ShowImage(imn, 'Image corrupted by Gaussian
    noise');//ShowImage() is used to show image,
    figure is command to view images in separate
    window.
46 title('Image corrupted by Gaussian noise');//title
    () is used for providing a title to an image.
47
48
49 imn = imnoise(im, 'speckle');//imnoise() is used to
    add noise in an image.'speckle' is used as a
    second argument to function for adding Speckle
    noise.Third argument is used for various.
50 figure,ShowImage(imn, 'Image corrupted by Speckle
    noise');//ShowImage() is used to show image,
    figure is command to view images in separate
    window.
51 title('Image corrupted by speckle noise');//title()
    is used for providing a title to an image.
52

```

```

53 imns = imnoise(im, 'salt & pepper');//imnoise() is
    used to add noise in an image.'gaussian' is used
    as a second argument to function for adding Salt
    & Pepper noise.Third argument is used for noise
    density.
54 figure,ShowImage(imns,'Image corrupted by salt and
    pepper noise');//ShowImage() is used to show
    image, figure is command to view images in
    separate window.
55 title('Image corrupted by salt and pepper noise');//
    title() is used for providing a title to an
    image.
56
57 filter=fspecial('average',3);//fspecial() is used to
    create special filters.
58 imf = imfilter(imns, filter);
59 figure,ShowImage(imf,'Filterd Grayscale image using
    3*3 average mask');//ShowImage() is used to show
    image, figure is command to view images in
    separate window.
60 title('Filterd Grayscale image using 16*16 average
    mask');//title() is used for providing a title
    to an image.
61
62 subim = imcrop(imns, [1, 1, 300, 300]);//imcrop() is
    used to crop an image.
63 function [resimg]=median2(image,filtersize,type1)//
    median2() is function to filter an image.
64     size1=filtersize;
65     [nr,nc]=size(image);
66     if type1=="zero" then
67         temp=zeros(nr+2*floor(size1/2),nc+2*floor(
            size1/2));
68     end
69     temp=zeros(nr+2*floor(size1/2),nc+2*floor(size1
        /2));
70
71     temp(ceil(size1/2):nr+ceil(size1/2)-1,ceil(size1

```

```

        /2):nc+ceil(size1/2)-1)=subim(1:$,1:$)
72     for i=ceil(size1/2):nr+ceil(size1/2)-1
73         for j=ceil(size1/2):nc+ceil(size1/2)-1
74             t=temp(i-floor(size1/2):1:i+floor(size1
                /2),j-floor(size1/2):1:j+floor(size1
                /2))
75             y=gsort(t);
76             temp(i,j)=median(y);
77         end
78     end
79
80     nn=temp(ceil(size1/2):nr+ceil(size1/2)-1,ceil(
        size1/2):nc+ceil(size1/2)-1)
81     resimg=mat2gray(nn)
82
83 endfunction
84
85 nnn=median2(subim,3,"zero");//median2() is a 2-D
    median filter , second argument is used for filter
    size. It should be odd integer. Third argument
    is used for zero padding.
86 figure,ShowImage(nnn,'Filtered image using 3*3
    median filter');//ShowImage() is used to show
    image, figure is command to view images in
    separate window.
87 title('Filtered image using 3*3 median filter ');//
    title() is used for providing a title to an
    image.
88
89
90
91 F=fspecial('sobel');//fspecial('sobel') returns a 3x3
    horizontal edges sobel filter.
92 imf = imfilter(im, F);
93 figure,ShowImage(imf,'Filterd Grayscale image using
    3*3 average mask');//ShowImage() is used to show
    image, figure is command to view images in
    separate window.

```

```

94 title('Filterd Grayscale image using 3*3 average
    mask');//title() is used for providing a title
    to an image.
95
96 F=fspecial('prewitt')//fspecial('prewitt') returns a
    3x3 horizontal edges prewitt filter.
97 imf = imfilter(im, F);
98 figure,ShowImage(imf,'Filtering Grayscale image
    using 3x3 horizontal edges prewitt filter.');//
    ShowImage() is used to show image, figure is
    command to view images in separate window.
99 title('Filtering Grayscale image using 3x3
    horizontal edges prewitt filter.');//title() is
    used for providing a title to an image.
100
101 F=fspecial('gaussian')//fspecial('gaussian', hsize,
    sigma) returns a Gaussian lowpass filter.
102 imf = imfilter(im, F);
103 figure,ShowImage(imf,'Filtering Grayscale image
    using Gaussian lowpass filter.');//ShowImage()
    is used to show image, figure is command to view
    images in separate window.
104 title('Filtering Grayscale image using Gaussian
    lowpass filter. ');//title() is used for
    providing a title to an image.
105
106 F=fspecial('laplacian')//fspecial('laplacian', alpha)
    returns a 3-by-3 Laplacian filter. The returned
    filter is [alpha, 1-alpha, alpha; 1-alpha, -4, 1-
    alpha; alpha, 1-alpha, alpha]/(alpha+1). The
    default value for alpha is 0.2.
107 imf = imfilter(im, F);
108 figure,ShowImage(imf,'Filtering Grayscale image
    using 3x3 Laplacian filter.');//ShowImage() is
    used to show image, figure is command to view
    images in separate window.
109 title('Filtering Grayscale image using 3x3
    Laplacian filter.');//title() is used for

```

```
    providing a title to an image.
110
111 F=fspecial('log')//fspecial('log',hsize,sigma)
    returns a Laplacian of Gaussian filter. The size
    of returned filter is determined by parameter
    hsize. hsize can be a 1x2 vector which indicate
    the rows and columns of F. If hsize is a scalar,
    F is a square matrix. The default value for hsize
    is [5, 5]; the default value for sigma is 0.5.
112 imf = imfilter(im, F);
113 figure,ShowImage(imf,'Filterd Grayscale image using
    Laplacian of Gaussian filter.');//ShowImage() is
    used to show image, figure is command to view
    images in separate window.
114 title('Filterd Grayscale image using Laplacian of
    Gaussian filter.');//title() is used for
    providing a title to an image.
```

Experiment: 4

Filtering The Frequency Domain

Scilab code Solution 4.1 Filtering In the Frequency Domain

```
1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox: Huffcomp Toolbox 1.1.1
7 //OS: Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11 clc //to clear command window.
12 clear all //to kill previously defined variables.
13 xdel(winsid()) //to close all currently open figure(s
    ).
14
15 function [H]=lowpassfilter(type1,M,N,D0,n) //
    lowpassfilter is used to filter an image .
16     u=0:(M-1);
```

```

17     v=0:(N-1);
18     idx=find(u>M/2);
19     u(idx)=u(idx)-M;
20     idy=find(v>N/2);
21     v(idy)=v(idy)-N;
22     [U,V]=meshgrid(v,u);
23     D=sqrt(U.^2+V.^2);
24     select type1
25     case 'ideal' then
26         H=double(D<=D0);
27     case 'butterworth' then
28         if argn(2)==4 then
29             n=1;
30         end
31         t1=D./D0
32         t2=t1.^(2*n)
33         t3=1+t2;
34         [nr1,nc1]=size(t3);
35         a=ones(nr1,nc1);
36         H=a./t3;
37     case 'gaussian' then
38         H=exp(-(D.^2)./(2*(D0^2)));
39     else
40         disp('Unknown filter type.')
41     end
42 endfunction
43
44 function [H1]=highpassfilter(type2,M,N,D0,n) //
    highpassfilter() is used to filter an image.
45     if argn(2)==4 then
46         n=1;
47     end
48     h=lowpassfilter(type2,M,N,D0,n);
49     H1=1-h;
50 endfunction
51
52 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed. getSIVPpath()

```

```

    is pre-defined function in SIVP toolbox.
53  rgb = imread(SIVP_PATH + 'images/lena.png'); //
    Reading from sub-directory of images.
54  figure, ShowColorImage(rgb, 'Original Color image'); //
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
55  title('Original Color Image'); // title() is used for
    providing a title to an image.
56
57  im=rgb2gray(rgb); //to convert a RGB image into
    grayscale image.
58
59
60  f=double(im);
61  [M,N]=size(f);
62
63  h=fft2(f); //fft2() is used to find 2-Dimensional
    Fast Fourier Transform of an matrix
64  i=log(1+abs(h));
65  in=fftshift(i); //fftshift() is used to rearrange the
    fft output, moving the zero frequency to the
    center of the spectrum.
66  inm=mat2gray(in)
67  figure, ShowImage(inm, 'Frequency Spectrum'); //
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
68  title('Frequency Spectrum'); //title() is used for
    providing a title to an image.
69
70
71  filt=lowpassfilter('ideal',M,N,5);
72  n=filt.*h; //convolving an image with a two
    dimensional filter.
73  n1=real(ifft(n)); //ifft() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
74  filt1=fftshift(filt); //fftshift() is used to

```

```

    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
75 filt2=mat2gray(filt1);
76 figure,ShowImage(filt2,'Frequency Spectrum');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
77 title('Frequency Spectrum');//title() is used for
    providing a title to an image.
78
79 mm=mat2gray(n1);
80 figure,ShowImage(mm,'Ideal Lowpass filteredimage [
    cutofffreq=5]');//ShowColorImage() is used to
    show color image, figure is command to view
    images in separate window.
81 title('Ideal Lowpassfiltered image[cutofffreq=5]');//
    //title() is used for providing a title to an
    image.
82
83
84
85 filt=lowpassfilter('ideal',M,N,50);
86 n=filt.*h;//convolving an image with a two
    dimensional filter.
87 n1=real(iffn(n));//iffn() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
88 filt1=fftshift(filt);//fftshift() is used to
    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
89 filt2=mat2gray(filt1);
90 figure,ShowImage(filt2,'Frequency Spectrum');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
91 title('Frequency Spectrum');//title() is used for
    providing a title to an image.
92
93 mm=mat2gray(n1);

```

```

94 figure,ShowImage(mm,'Ideal Lowpass filtered image[
    cutofffreq=5]');//ShowColorImage() is used to
    show color image, figure is command to view
    images in separate window.
95 title('Ideal Lowpass filtered image[cutofffreq=50]')
    ;//title() is used for providing a title to an
    image.
96
97
98 filt=lowpassfilter('ideal',M,N,200);
99 n=filt.*h;//convolving an image with a two
    dimensional filter.
100 n1=real(ifft(n));//ifft() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
101 filt1=fftshift(filt);//fftshift() is used to
    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
102 filt2=mat2gray(filt1);
103 figure,ShowImage(filt2,'Frequency Spectrum');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
104 title('Frequency Spectrum');//title() is used for
    providing a title to an image.
105
106 mm=mat2gray(n1);
107 figure,ShowImage(mm,'Ideal Lowpass filtered image[
    cutofffreq=200]');//ShowColorImage() is used to
    show color image, figure is command to view
    images in separate window.
108 title('Ideal Lowpass filtered image[cutofffreq=200]')
    );//title() is used for providing a title to an
    image.
109
110
111
112
113 filt=lowpassfilter('butterworth',M,N,5);

```

```

114 n=filt.*h;//convolving an image with a two
    dimensional filter.
115 n1=real(iffn(n));//iffn() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
116 filt1=fftshift(filt);//fftshift() is used to
    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
117 filt2=mat2gray(filt1);
118 figure,ShowImage(filt2,'Frequency Spectrum');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
119 title('Frequency Spectrum');//title() is used for
    providing a title to an image.
120
121 mm=mat2gray(n1);
122 figure,ShowImage(mm,'butterworth Lowpass filtered
    image[cutofffreq=5]');//ShowColorImage() is used
    to show color image, figure is command to view
    images in separate window.
123 title('butterworth Lowpass filtered image[cutofffreq
    =5]');//title() is used for providing a title to
    an image.
124
125
126
127 filt=lowpassfilter('butterworth',M,N,50);
128 n=filt.*h;//convolving an image with a two
    dimensional filter.
129 n1=real(iffn(n));//iffn() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
130 filt1=fftshift(filt);//fftshift() is used to
    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
131 filt2=mat2gray(filt1);
132 figure,ShowImage(filt2,'Frequency Spectrum');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate

```

```

    window.
133 title('Frequency Spectrum'); //title() is used for
    providing a title to an image.
134
135 mm=mat2gray(n1);
136 figure,ShowImage(mm,'butterworth Lowpass
    filteredimage [cutofffreq=5]'); //ShowColorImage()
    is used to show color image, figure is command to
    view images in separate window.
137 title('butterworth Lowpass filteredimage [cutofffreq
    =50]'); //title() is used for providing a title
    to an image.
138
139
140 filt=lowpassfilter('butterworth',M,N,200);
141 n=filt.*h; //convolving an image with a two
    dimensional filter.
142 n1=real(ifft(n)); //ifft() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
143 filt1=fftshift(filt); //fftshift() is used to
    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
144 filt2=mat2gray(filt1);
145 figure,ShowImage(filt2,'Frequency Spectrum'); //
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
146 title('Frequency Spectrum'); //title() is used for
    providing a title to an image.
147
148 mm=mat2gray(n1);
149 figure,ShowImage(mm,'butterworth Lowpass
    filteredimage [cutofffreq=200]'); //ShowColorImage
    () is used to show color image, figure is command
    to view images in separate window.
150 title('butterworth Lowpass filteredimage [cutofffreq
    =200]'); //title() is used for providing a title
    to an image.

```

```

151
152
153 filt=lowpassfilter('butterworth',M,N,5);
154 n=filt.*h;//convolving an image with a two
    dimensional filter.
155 n1=real(ifft(n));//ifft() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
156 filt1=fftshift(filt);//fftshift() is used to
    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
157 filt2=mat2gray(filt1);
158 figure,ShowImage(filt2,'Frequency Spectrum');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
159 title('Frequency Spectrum');//title() is used for
    providing a title to an image.
160
161 mm=mat2gray(n1);
162 figure,ShowImage(mm,'gaussian Lowpass filtered image
    [cutofffreq=5]');//ShowColorImage() is used to
    show color image, figure is command to view
    images in separate window.
163 title('gaussian Lowpass filtered image[cutofffreq=5]
    ');//title() is used for providing a title to an
    image.
164
165
166
167 filt=lowpassfilter('gaussian',M,N,50);
168 n=filt.*h;//convolving an image with a two
    dimensional filter.
169 n1=real(ifft(n));//ifft() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
170 filt1=fftshift(filt);//fftshift() is used to
    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
171 filt2=mat2gray(filt1);

```

```

172 figure,ShowImage(filt2,'Frequency Spectrum');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
173 title('Frequency Spectrum');//title() is used for
    providing a title to an image.
174
175 mm=mat2gray(n1);
176 figure,ShowImage(mm,'gaussian Lowpass filtered image
    [cutofffreq=5]');//ShowColorImage() is used to
    show color image, figure is command to view
    images in separate window.
177 title('gaussian Lowpass filtered image[cutofffreq
    =50]');//title() is used for providing a title
    to an image.
178
179
180 filt=lowpassfilter('gaussian',M,N,200);
181 n=filt.*h;//convolving an image with a two
    dimensional filter.
182 n1=real(ifft(n));//ifft() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
183 filt1=fftshift(filt);//fftshift() is used to
    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
184 filt2=mat2gray(filt1);
185 figure,ShowImage(filt2,'Frequency Spectrum');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
186 title('Frequency Spectrum');//title() is used for
    providing a title to an image.
187
188 mm=mat2gray(n1);
189 figure,ShowImage(mm,'gaussian Lowpass filtered image
    [cutofffreq=200]');//ShowColorImage() is used to
    show color image, figure is command to view
    images in separate window.

```

```

190 title('gaussian Lowpass filtered image[cutofffreq
      =200]'); //title() is used for providing a title
      to an image.
191
192
193
194 clc //to clear command window.
195 clear all //to kill previously defined variables.
196 xdel(winsid()) //to close all currently open figure(s
      ).
197
198 SIVP_PATH = getSIVPpath(); //to locate a directory
      in which SIVP toolbox is installed. getSIVPpath()
      is pre-defined function in SIVP toolbox.
199 rgb = imread(SIVP_PATH + 'images/lena.png'); //
      Reading from sub-directory of images.
200 figure, ShowColorImage(rgb, 'Original color image'); //
      ShowColorImage() is used to show color image,
      figure is command to view images in separate
      window.
201 title('Original Color Image'); //title() is used for
      providing a title to an image.
202
203 im=rgb2gray(rgb); //to convert a RGB image into
      grayscale image.
204
205
206 f=double(im);
207 [M,N]=size(f);
208
209 h=fft2(f); //fft2() is used to find 2-Dimensional
      Fast Fourier Transform of an matrix
210 i=log(1+abs(h));
211 in=fftshift(i);
212 inm=mat2gray(in)
213 figure, ShowImage(inm, 'Frequency Spectrum'); //
      ShowColorImage() is used to show color image,
      figure is command to view images in separate

```

```

    window.
214 title('Frequency Spectrum'); //title() is used for
    providing a title to an image.
215
216
217 filt=highpassfilter('ideal',M,N,5);
218 n=filt.*h; //convolving an image with a two
    dimensional filter.
219 n1=real(iffth(n)); //iffth() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
220 filt1=fftshift(filt); //fftshift() is used to
    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
221 filt2=mat2gray(filt1);
222 figure, ShowImage(filt2, 'Frequency Spectrum'); //
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
223 title('Frequency Spectrum'); //title() is used for
    providing a title to an image.
224
225 mm=mat2gray(n1);
226 figure, ShowImage(mm, 'Ideal highpass filteredimage [
    cutofffreq=5]'); //ShowColorImage() is used to
    show color image, figure is command to view
    images in separate window.
227 title('Ideal highpass filteredimage [cutofffreq=5]');
    //title() is used for providing a title to an
    image.
228
229
230
231 filt=highpassfilter('ideal',M,N,50);
232 n=filt.*h; //convolving an image with a two
    dimensional filter.
233 n1=real(iffth(n)); //iffth() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
234 filt1=fftshift(filt); //fftshift() is used to

```

```

    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
235 filt2=mat2gray(filt1);
236 figure,ShowImage(filt2,'Frequency Spectrum');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
237 title('Frequency Spectrum');//title() is used for
    providing a title to an image.
238
239 mm=mat2gray(n1);
240 figure,ShowImage(mm,'Ideal highpass filteredimage [
    cutofffreq=50]');//ShowColorImage() is used to
    show color image, figure is command to view
    images in separate window.
241 title('Ideal highpass filteredimage [cutofffreq=50]')
    ;//title() is used for providing a title to an
    image.
242
243
244 filt=highpassfilter('ideal',M,N,200);
245 n=filt.*h;//convolving an image with a two
    dimensional filter.
246 n1=real(ifft(n));//ifft() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
247 filt1=fftshift(filt);//fftshift() is used to
    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
248 filt2=mat2gray(filt1);
249 figure,ShowImage(filt2,'Frequency Spectrum');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
250 title('Frequency Spectrum');//title() is used for
    providing a title to an image.
251
252 mm=mat2gray(n1);
253 figure,ShowImage(mm,'Ideal highpass filteredimage [

```

```

        cutofffreq=200]'); // ShowColorImage() is used to
        show color image, figure is command to view
        images in separate window.
254 title('Ideal highpass filtered image [cutofffreq=200]')
        ); // title() is used for providing a title to an
        image.
255
256
257
258 filt=highpassfilter('butterworth',M,N,200);
259 n=filt.*h; //convolving an image with a two
        dimensional filter.
260 n1=real(ifft(n)); //ifft() is used to find inverse 2-
        Dimensional Fast fourier transform of an matrix
261 filt1=fftshift(filt); //fftshift() is used to
        rearrange the fft output, moving the zero
        frequency to the center of the spectrum.
262 filt2=mat2gray(filt1);
263 figure, ShowImage(filt2, 'Frequency Spectrum'); //
        ShowColorImage() is used to show color image,
        figure is command to view images in separate
        window.
264 title('Frequency Spectrum'); //title() is used for
        providing a title to an image.
265
266 mm=mat2gray(n1);
267 figure, ShowImage(mm, 'butterworth highpass filtered
        image [cutofffreq=200]'); // ShowColorImage() is
        used to show color image, figure is command to
        view images in separate window.
268 title('butterworth highpass filtered image [
        cutofffreq=200]'); //title() is used for providing
        a title to an image.
269
270
271
272 filt=highpassfilter('gaussian',M,N,200);
273

```

```

274 n=filt.*h;//convolving an image with a two
    dimensional filter.
275 n1=real(iffn(n));//iffn() is used to find inverse 2-
    Dimensional Fast fourier transform of an matrix
276 filt1=fftshift(filt);//fftshift() is used to
    rearrange the fft output, moving the zero
    frequency to the center of the spectrum.
277 filt2=mat2gray(filt1);
278 figure,ShowImage(filt2,'Frequency Spectrum');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
279 title('Frequency Spectrum');//title() is used for
    providing a title to an image.
280
281 mm=mat2gray(n1);
282 figure,ShowImage(mm,'gaussian highpass filtered
    image[cutofffreq=200]');//ShowColorImage() is
    used to show color image, figure is command to
    view images in separate window.
283 title('gaussian highpass filtered image[cutofffreq
    =200]');//title() is used for providing a title
    to an image.

```

Experiment: 5

Image Restoration

Scilab code Solution 5.1 Image Restoration

```
1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox:Huffcomp Toolbox 1.1.1
7 //OS:Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11 clc //to clear command window.
12 clear all //to kill previously defined variables.
13 xdel(winsid())//to close all currently open figure(s
    ).
14
15 function imn = imnoise22(im, noise_type, param1,
    param2)
16     imtype = typeof(im(1));
17     im=im2double(im);
18     im=matrix(im(:), size(im));
```

```

19 //Gaussian noise
20 if (noise_type == 'gaussian' | noise_type == '
    Gaussian') then
21     if ~exists('param1','local')
22         m=0
23     else
24         m=param1
25     end
26     if ~exists('param2','local')
27         v=0.01
28     else
29         v=param2
30     end
31
32     old_rand_gen=rand('info');
33     rand('normal');
34
35     imn = im + sqrt(v)*rand(im) + m;
36
37     rand(old_rand_gen);
38
39
40 elseif noise_type == 'localvar'
41     if argn(2) < 3 then
42         error('Too few arguments for noise type ''
            localvar''.');
43     elseif argn(2) == 3 then
44         if( or(size(im)<>size(param1))) then
45             error("The first parameter for ''localvar''
                should have the same"+...
                " size with the input image.");
46         end
47     end
48
49     old_rand_gen=rand('info');
50     rand('normal');
51     imn = matrix(im(:), size(im)) + sqrt(param1).*
        rand(im);
52     rand(old_rand_gen);

```

```

53
54     elseif argn(2) == 4 then
55         if( or(size(param1)<>size(param2))) then
56             error("The two parameters for ''localvar''
57                 should have the same size.");
58         end
59
60         minp1 = min(param1);
61         maxp1 = max(param1);
62         imn = min(max(im(:),minp1),maxp1); //max(im,
63                                         //because
64                                         //im is a
65                                         //hypermat
66                                         //
67                                         //
68                                         //
69                                         //
70                                         //
71                                         //
72                                         //
73                                         //
74                                         //
75                                         //
76                                         //
77                                         //
78                                         //
79                                         //
80                                         //
81                                         //
82                                         //
83                                         //
84                                         //
85                                         //
86                                         //
87                                         //
88                                         //
89                                         //
90                                         //
91                                         //
92                                         //
93                                         //
94                                         //
95                                         //
96                                         //
97                                         //
98                                         //
99                                         //
100                                        //
101                                        //
102                                        //
103                                        //
104                                        //
105                                        //
106                                        //
107                                        //
108                                        //
109                                        //
110                                        //
111                                        //
112                                        //
113                                        //
114                                        //
115                                        //
116                                        //
117                                        //
118                                        //
119                                        //
120                                        //
121                                        //
122                                        //
123                                        //
124                                        //
125                                        //
126                                        //
127                                        //
128                                        //
129                                        //
130                                        //
131                                        //
132                                        //
133                                        //
134                                        //
135                                        //
136                                        //
137                                        //
138                                        //
139                                        //
140                                        //
141                                        //
142                                        //
143                                        //
144                                        //
145                                        //
146                                        //
147                                        //
148                                        //
149                                        //
150                                        //
151                                        //
152                                        //
153                                        //
154                                        //
155                                        //
156                                        //
157                                        //
158                                        //
159                                        //
160                                        //
161                                        //
162                                        //
163                                        //
164                                        //
165                                        //
166                                        //
167                                        //
168                                        //
169                                        //
170                                        //
171                                        //
172                                        //
173                                        //
174                                        //
175                                        //
176                                        //
177                                        //
178                                        //
179                                        //
180                                        //
181                                        //
182                                        //
183                                        //
184                                        //
185                                        //
186                                        //
187                                        //
188                                        //
189                                        //
190                                        //
191                                        //
192                                        //
193                                        //
194                                        //
195                                        //
196                                        //
197                                        //
198                                        //
199                                        //
200                                        //
201                                        //
202                                        //
203                                        //
204                                        //
205                                        //
206                                        //
207                                        //
208                                        //
209                                        //
210                                        //
211                                        //
212                                        //
213                                        //
214                                        //
215                                        //
216                                        //
217                                        //
218                                        //
219                                        //
220                                        //
221                                        //
222                                        //
223                                        //
224                                        //
225                                        //
226                                        //
227                                        //
228                                        //
229                                        //
230                                        //
231                                        //
232                                        //
233                                        //
234                                        //
235                                        //
236                                        //
237                                        //
238                                        //
239                                        //
240                                        //
241                                        //
242                                        //
243                                        //
244                                        //
245                                        //
246                                        //
247                                        //
248                                        //
249                                        //
250                                        //
251                                        //
252                                        //
253                                        //
254                                        //
255                                        //
256                                        //
257                                        //
258                                        //
259                                        //
260                                        //
261                                        //
262                                        //
263                                        //
264                                        //
265                                        //
266                                        //
267                                        //
268                                        //
269                                        //
270                                        //
271                                        //
272                                        //
273                                        //
274                                        //
275                                        //
276                                        //
277                                        //
278                                        //
279                                        //
280                                        //
281                                        //
282                                        //
283                                        //
284                                        //
285                                        //
286                                        //
287                                        //
288                                        //
289                                        //
290                                        //
291                                        //
292                                        //
293                                        //
294                                        //
295                                        //
296                                        //
297                                        //
298                                        //
299                                        //
300                                        //
301                                        //
302                                        //
303                                        //
304                                        //
305                                        //
306                                        //
307                                        //
308                                        //
309                                        //
310                                        //
311                                        //
312                                        //
313                                        //
314                                        //
315                                        //
316                                        //
317                                        //
318                                        //
319                                        //
320                                        //
321                                        //
322                                        //
323                                        //
324                                        //
325                                        //
326                                        //
327                                        //
328                                        //
329                                        //
330                                        //
331                                        //
332                                        //
333                                        //
334                                        //
335                                        //
336                                        //
337                                        //
338                                        //
339                                        //
340                                        //
341                                        //
342                                        //
343                                        //
344                                        //
345                                        //
346                                        //
347                                        //
348                                        //
349                                        //
350                                        //
351                                        //
352                                        //
353                                        //
354                                        //
355                                        //
356                                        //
357                                        //
358                                        //
359                                        //
360                                        //
361                                        //
362                                        //
363                                        //
364                                        //
365                                        //
366                                        //
367                                        //
368                                        //
369                                        //
370                                        //
371                                        //
372                                        //
373                                        //
374                                        //
375                                        //
376                                        //
377                                        //
378                                        //
379                                        //
380                                        //
381                                        //
382                                        //
383                                        //
384                                        //
385                                        //
386                                        //
387                                        //
388                                        //
389                                        //
390                                        //
391                                        //
392                                        //
393                                        //
394                                        //
395                                        //
396                                        //
397                                        //
398                                        //
399                                        //
400                                        //
401                                        //
402                                        //
403                                        //
404                                        //
405                                        //
406                                        //
407                                        //
408                                        //
409                                        //
410                                        //
411                                        //
412                                        //
413                                        //
414                                        //
415                                        //
416                                        //
417                                        //
418                                        //
419                                        //
420                                        //
421                                        //
422                                        //
423                                        //
424                                        //
425                                        //
426                                        //
427                                        //
428                                        //
429                                        //
430                                        //
431                                        //
432                                        //
433                                        //
434                                        //
435                                        //
436                                        //
437                                        //
438                                        //
439                                        //
440                                        //
441                                        //
442                                        //
443                                        //
444                                        //
445                                        //
446                                        //
447                                        //
448                                        //
449                                        //
450                                        //
451                                        //
452                                        //
453                                        //
454                                        //
455                                        //
456                                        //
457                                        //
458                                        //
459                                        //
460                                        //
461                                        //
462                                        //
463                                        //
464                                        //
465                                        //
466                                        //
467                                        //
468                                        //
469                                        //
470                                        //
471                                        //
472                                        //
473                                        //
474                                        //
475                                        //
476                                        //
477                                        //
478                                        //
479                                        //
480                                        //
481                                        //
482                                        //
483                                        //
484                                        //
485                                        //
486                                        //
487                                        //
488                                        //
489                                        //
490                                        //
491                                        //
492                                        //
493                                        //
494                                        //
495                                        //
496                                        //
497                                        //
498                                        //
499                                        //
500                                        //
501                                        //
502                                        //
503                                        //
504                                        //
505                                        //
506                                        //
507                                        //
508                                        //
509                                        //
510                                        //
511                                        //
512                                        //
513                                        //
514                                        //
515                                        //
516                                        //
517                                        //
518                                        //
519                                        //
520                                        //
521                                        //
522                                        //
523                                        //
524                                        //
525                                        //
526                                        //
527                                        //
528                                        //
529                                        //
530                                        //
531                                        //
532                                        //
533                                        //
534                                        //
535                                        //
536                                        //
537                                        //
538                                        //
539                                        //
540                                        //
541                                        //
542                                        //
543                                        //
544                                        //
545                                        //
546                                        //
547                                        //
548                                        //
549                                        //
550                                        //
551                                        //
552                                        //
553                                        //
554                                        //
555                                        //
556                                        //
557                                        //
558                                        //
559                                        //
560                                        //
561                                        //
562                                        //
563                                        //
564                                        //
565                                        //
566                                        //
567                                        //
568                                        //
569                                        //
570                                        //
571                                        //
572                                        //
573                                        //
574                                        //
575                                        //
576                                        //
577                                        //
578                                        //
579                                        //
580                                        //
581                                        //
582                                        //
583                                        //
584                                        //
585                                        //
586                                        //
587                                        //
588                                        //
589                                        //
590                                        //
591                                        //
592                                        //
593                                        //
594                                        //
595                                        //
596                                        //
597                                        //
598                                        //
599                                        //
600                                        //
601                                        //
602                                        //
603                                        //
604                                        //
605                                        //
606                                        //
607                                        //
608                                        //
609                                        //
610                                        //
611                                        //
612                                        //
613                                        //
614                                        //
615                                        //
616                                        //
617                                        //
618                                        //
619                                        //
620                                        //
621                                        //
622                                        //
623                                        //
624                                        //
625                                        //
626                                        //
627                                        //
628                                        //
629                                        //
630                                        //
631                                        //
632                                        //
633                                        //
634                                        //
635                                        //
636                                        //
637                                        //
638                                        //
639                                        //
640                                        //
641                                        //
642                                        //
643                                        //
644                                        //
645                                        //
646                                        //
647                                        //
648                                        //
649                                        //
650                                        //
651                                        //
652                                        //
653                                        //
654                                        //
655                                        //
656                                        //
657                                        //
658                                        //
659                                        //
660                                        //
661                                        //
662                                        //
663                                        //
664                                        //
665                                        //
666                                        //
667                                        //
668                                        //
669                                        //
670                                        //
671                                        //
672                                        //
673                                        //
674                                        //
675                                        //
676                                        //
677                                        //
678                                        //
679                                        //
680                                        //
681                                        //
682                                        //
683                                        //
684                                        //
685                                        //
686                                        //
687                                        //
688                                        //
689                                        //
690                                        //
691                                        //
692                                        //
693                                        //
694                                        //
695                                        //
696                                        //
697                                        //
698                                        //
699                                        //
700                                        //
701                                        //
702                                        //
703                                        //
704                                        //
705                                        //
706                                        //
707                                        //
708                                        //
709                                        //
710                                        //
711                                        //
712                                        //
713                                        //
714                                        //
715                                        //
716                                        //
717                                        //
718                                        //
719                                        //
720                                        //
721                                        //
722                                        //
723                                        //
724                                        //
725                                        //
726                                        //
727                                        //
728                                        //
729                                        //
730                                        //
731                                        //
732                                        //
733                                        //
734                                        //
735                                        //
736                                        //
737                                        //
738                                        //
739                                        //
740                                        //
741                                        //
742                                        //
743                                        //
744                                        //
745                                        //
746                                        //
747                                        //
748                                        //
749                                        //
750                                        //
751                                        //
752                                        //
753                                        //
754                                        //
755                                        //
756                                        //
757                                        //
758                                        //
759                                        //
760                                        //
761                                        //
762                                        //
763                                        //
764                                        //
765                                        //
766                                        //
767                                        //
768                                        //
769                                        //
770                                        //
771                                        //
772                                        //
773                                        //
774                                        //
775                                        //
776                                        //
777                                        //
778                                        //
779                                        //
780                                        //
781                                        //
782                                        //
783                                        //
784                                        //
785                                        //
786                                        //
787                                        //
788                                        //
789                                        //
790                                        //
791                                        //
792                                        //
793                                        //
794                                        //
795                                        //
796                                        //
797                                        //
798                                        //
799                                        //
800                                        //
801                                        //
802                                        //
803                                        //
804                                        //
805                                        //
806                                        //
807                                        //
808                                        //
809                                        //
810                                        //
811                                        //
812                                        //
813                                        //
814                                        //
815                                        //
816                                        //
817                                        //
818                                        //
819                                        //
820                                        //
821                                        //
822                                        //
823                                        //
824                                        //
825                                        //
826                                        //
827                                        //
828                                        //
829                                        //
830                                        //
831                                        //
832                                        //
833                                        //
834                                        //
835                                        //
836                                        //
837                                        //
838                                        //
839                                        //
840                                        //
841                                        //
842                                        //
843                                        //
844                                        //
845                                        //
846                                        //
847                                        //
848                                        //
849                                        //
850                                        //
851                                        //
852                                        //
853                                        //
854                                        //
855                                        //
856                                        //
857                                        //
858                                        //
859                                        //
860                                        //
861                                        //
862                                        //
863                                        //
864                                        //
865                                        //
866                                        //
867                                        //
868                                        //
869                                        //
870                                        //
871                                        //
872                                        //
873                                        //
874                                        //
875                                        //
876                                        //
877                                        //
878                                        //
879                                        //
880                                        //
881                                        //
882                                        //
883                                        //
884                                        //
885                                        //
886                                        //
887                                        //
888                                        //
889                                        //
890                                        //
891                                        //
892                                        //
893                                        //
894                                        //
895                                        //
896                                        //
897                                        //
898                                        //
899                                        //
900                                        //
901                                        //
902                                        //
903                                        //
904                                        //
905                                        //
906                                        //
907                                        //
908                                        //
909                                        //
910                                        //
911                                        //
912                                        //
913                                        //
914                                        //
915                                        //
916                                        //
917                                        //
918                                        //
919                                        //
920                                        //
921                                        //
922                                        //
923                                        //
924                                        //
925                                        //
926                                        //
927                                        //
928                                        //
929                                        //
930                                        //
931                                        //
932                                        //
933                                        //
934                                        //
935                                        //
936                                        //
937                                        //
938                                        //
939                                        //
940                                        //
941                                        //
942                                        //
943                                        //
944                                        //
945                                        //
946                                        //
947                                        //
948                                        //
949                                        //
950                                        //
951                                        //
952                                        //
953                                        //
954                                        //
955                                        //
956                                        //
957                                        //
958                                        //
959                                        //
960                                        //
961                                        //
962                                        //
963                                        //
964                                        //
965                                        //
966                                        //
967                                        //
968                                        //
969                                        //
970                                        //
971                                        //
972                                        //
973                                        //
974                                        //
975                                        //
976                                        //
977                                        //
978                                        //
979                                        //
980                                        //
981                                        //
982                                        //
983                                        //
984                                        //
985                                        //
986                                        //
987                                        //
988                                        //
989                                        //
990                                        //
991                                        //
992                                        //
993                                        //
994                                        //
995                                        //
996                                        //
997                                        //
998                                        //
999                                        //
1000                                       //

```

```

83     old_rand_gen=rand('info');
84     rand('uniform');
85     prob=rand(im);
86     rand(old_rand_gen);
87
88     imn=im;
89     imn(prob < d/2) = 0;
90     imn(prob >=d/2 & prob < d) = 1;
91
92     elseif noise_type=='speckle'
93         if ~exists('param1','local')
94             v=0.04
95         else
96             v=param1
97             if( v < 0) then
98                 error("The parameter for ''speckle'' noise
99                     should >=0.");
100             end
101         end
102
103         old_rand_gen=rand('info');
104         rand('uniform');
105         imn = im + im .* (sqrt(v) * (rand(im)-0.5) );
106         rand(old_rand_gen);
107
108     elseif noise_type == 'poisson'
109         error('Not yet implemented');
110     else
111         error('Invalid noise type. ');
112     end
113
114     //conver the output image to the same type as the
115     input image
116
117     select imtype
118     case 'uint8' then
119         imn = im2uint8(imn);
120     case 'int8' then

```

```

119     imn = im2int8(imn);
120     case 'uint16' then
121         imn = im2uint16(imn);
122     case 'int16' then
123         imn = im2int16(imn);
124     case 'int32' then
125         imn = im2int32(imn);
126     case 'constant' then
127         imn = im2double(imn);
128     else
129         error("Data type " + imtype + " is not supported
130             .");
131     end
132 endfunction
133
134
135 //
136 SIVP_PATH = getSIVPpath(); //to locate a directory
137     in which SIVP toolbox is installed. getSIVPpath()
138     is pre-defined function in SIVP toolbox.
139 rgb = imread(SIVP_PATH + 'images/lena.png'); //
140     Reading from sub-directory of images.
141 figure, ShowColorImage(rgb, 'Original color image'); //
142     ShowColorImage() is used to show color image,
143     figure is command to view images in separate
144     window.
145 title('Original Color Image'); //title() is used for
146     providing a title to an image.
147 im=rgb2gray(rgb); //to convert a RGB image into
148     grayscale image.
149
150
151
152 k = imnoise(im, 'salt & pepper', 0.02);
153 figure, ShowImage(k, 'Salt & pepper noise corrupted
154     image'); //ShowColorImage() is used to show color
155     image, figure is command to view images in
156     separate window.
157 title('Salt & pepper noise corrupted image'); //title

```

```

    () is used for providing a title to an image.
145
146
147 [count, cells]=imhist(k,256);//imhist() is used to
    find histogram of an image.
148 figure;imhist(k,255, '');
149 title('Histogram of Salt & pepper noise corrupted
    image');//title() is used for providing a title
    to an image.
150
151 k = imnoise(im, 'gaussian', 0.02, 0.02);
152 figure, ShowImage(k, 'Gaussian noise corrupted image'
    );//ShowColorImage() is used to show color image
    , figure is command to view images in separate
    window.
153 title('Gaussian noise corrupted image');//title() is
    used for providing a title to an image.
154
155 [count, cells]=imhist(k,256);//imhist() is used to
    find histogram of an image.
156 figure;imhist(k,255, '');
157 title('Histogram of Gaussian noise corrupted image'
    );//title() is used for providing a title to an
    image.
158
159
160 k = imnoise22(im, 'speckle', 0.8);//minor error in
    original function
161 figure, ShowImage(k, 'Speckle noise corrupted image')
    ;//ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
162 title('Speckle noise corrupted image');//title() is
    used for providing a title to an image.
163 [count, cells]=imhist(k,256);//imhist() is used to
    find histogram of an image.
164 figure;imhist(k,255, '');
165 title('Histogram of Speckle noise corrupted image')

```

```
    ;//title() is used for providing a title to an
    image.
```

Scilab code Solution 5.2 Image Restoration

```
1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox:Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox:Huffcomp Toolbox 1.1.1
7 //OS:Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11 clc //to clear command window.
12 clear all //to kill previously defined variables.
13 xdel(winsid())//to close all currently open figure(s
    ).
14
15 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed. getSIVPpath()
    is pre-defined function in SIVP toolbox.
16 rgb = imread(SIVP_PATH + 'images/lena.png');//
    Reading from sub-directory of images.
17 figure, ShowColorImage(rgb, 'Original color image');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
18 title('Original Color Image');//title() is used for
    providing a title to an image.
19 im=rgb2gray(rgb);//to convert a RGB image into
    grayscale image.
20
```

```

21 function [f]=alphatrim(g,m,n,d)//alphatrim() is used
    to filter an image using alpha-trimmed mean
    filter
22     size1=m;
23     [nr,nc]=size(g);
24     temp=zeros(nr+2*floor(size1/2),nc+2*floor(size1
        /2));
25     temp(ceil(size1/2):nr+ceil(size1/2)-1,ceil(size1
        /2):nc+ceil(size1/2)-1)=g(1:$,1:$)
26
27     for i=ceil(size1/2):nr+ceil(size1/2)-1
28         for j=ceil(size1/2):nc+ceil(size1/2)-1
29             t=temp(i-floor(size1/2):1:i+floor(size1
                /2),j-floor(size1/2):1:j+floor(size1
                /2))
30             y=gsort(t);
31             a=y(:)
32             b=a';
33             t1=b(1+d/2:$-d/2);
34             temp2(i,j)=mean(t1);
35         end
36     end
37     nn=temp2(ceil(size1/2):nr+ceil(size1/2)-1,ceil(
        size1/2):nc+ceil(size1/2)-1)
38     f=mat2gray(nn)
39 endfunction
40
41
42 function [f]=gmean1(g,m,n);//gmean1() is used to
    filter an image using Geometric mean filter
43     size1=m;
44     q=m*n;
45     g=double(g);
46     [nr,nc]=size(g);
47     temp=zeros(nr+2*floor(size1/2),nc+2*floor(size1
        /2));
48     temp(ceil(size1/2):nr+ceil(size1/2)-1,ceil(size1
        /2):nc+ceil(size1/2)-1)=g(1:$,1:$)

```

```

49     temp=temp+1;
50     for i=ceil(size1/2):nr+ceil(size1/2)-1
51         for j=ceil(size1/2):nc+ceil(size1/2)-1
52             t=temp(i-floor(size1/2):1:i+floor(size1
                    /2),j-floor(size1/2):1:j+floor(size1
                    /2)) ;
53             temp2(i,j)=prod(t);
54         end
55     end
56     temp3=temp2.^(1/q);
57     nn=temp3(ceil(size1/2):nr+ceil(size1/2)-1,ceil(
        size1/2):nc+ceil(size1/2)-1)
58     f1=nn-1;
59     f=mat2gray(f1)
60
61 endfunction
62
63
64
65 function [f]=gmean2(g,m,n); //gmean2() is used to
    filter an image using Geometric mean filter
66     size1=m;
67     q=m*n;
68     [nr,nc]=size(g);
69     temp=zeros(nr+2*floor(size1/2),nc+2*floor(size1
        /2));
70
71     temp(ceil(size1/2):nr+ceil(size1/2)-1,ceil(size1
        /2):nc+ceil(size1/2)-1)=g(1:$,1:$)
72
73     for i=ceil(size1/2):nr+ceil(size1/2)-1
74         for j=ceil(size1/2):nc+ceil(size1/2)-1
75             t=temp(i-floor(size1/2):1:i+floor(size1
                    /2),j-floor(size1/2):1:j+floor(size1
                    /2)) ;
76             temp2(i,j)=geomean(t);
77         end
78     end

```

```

79     nn=temp2(ceil(size1/2):nr+ceil(size1/2)-1,ceil(
        size1/2):nc+ceil(size1/2)-1)
80     f=mat2gray(nn)
81 endfunction
82
83 function [f]=harmean1(g,m,n) //harmean1() is used to
    filter an image using Harmonic mean filter.
84     size1=m;
85     d=m*n;
86     g=double(g);
87     [nr,nc]=size(g);
88     temp=zeros(nr+2*floor(size1/2),nc+2*floor(size1
        /2));
89     temp(ceil(size1/2):nr+ceil(size1/2)-1,ceil(size1
        /2):nc+ceil(size1/2)-1)=g(1:$,1:$)
90
91     for i=ceil(size1/2):nr+ceil(size1/2)-1
92         for j=ceil(size1/2):nc+ceil(size1/2)-1
93             t=temp(i-floor(size1/2):1:i+floor(size1
                /2),j-floor(size1/2):1:j+floor(size1
                /2)) ;
94             t1=ones(m,n)./(t+%eps)
95             t2=sum(t1);
96             temp2(i,j)=d/t2;
97             // temp2(i,j)=harmean(t);
98         end
99     end
100    nn=temp2(ceil(size1/2):nr+ceil(size1/2)-1,ceil(
        size1/2):nc+ceil(size1/2)-1)
101    f=mat2gray(nn);
102 endfunction
103
104 function [f]=charmmean1(g,m,n,q) //charmmean1() is use
    to filter an image using Contra Harmonic mean
    filter
105     size1=m;
106     d=m*n;
107     g=double(g);

```

```

108     [nr,nc]=size(g);
109     temp=zeros(nr+2*floor(size1/2),nc+2*floor(size1
110             /2));
111     temp(ceil(size1/2):nr+ceil(size1/2)-1,ceil(size1
112             /2):nc+ceil(size1/2)-1)=g(1:$,1:$)
113     disp(q)
114     for i=ceil(size1/2):nr+ceil(size1/2)-1
115         for j=ceil(size1/2):nc+ceil(size1/2)-1
116             t=temp(i-floor(size1/2):1:i+floor(size1
117                     /2),j-floor(size1/2):1:j+floor(size1
118                             /2)) ;
119             d1=(t+%eps).^q
120             n1=(t+%eps).^(q+1)
121             d2=sum(d1);
122             n2=sum(n1);
123             temp2(i,j)=n2/(d2);
124         end
125     end
126     nn=temp2(ceil(size1/2):nr+ceil(size1/2)-1,ceil(
127             size1/2):nc+ceil(size1/2)-1)
128     f=nn;
129 endfunction
130
131
132
133 function [resimg]=ordfilt2(image,m,n,type1)//
134     ordfilt2() is used to filter an image using min
135     or max filter.
136     size1=m;
137     [nr,nc]=size(image);
138     temp=zeros(nr+2*floor(size1/2),nc+2*floor(size1
139             /2));
140     temp(ceil(size1/2):nr+ceil(size1/2)-1,ceil(size1
141             /2):nc+ceil(size1/2)-1)=image(1:$,1:$)
142     select type1
143     case 'min' then
144         for i=ceil(size1/2):nr+ceil(size1/2)-1
145             for j=ceil(size1/2):nc+ceil(size1/2)-1

```

```

137         t=temp(i-floor(size1/2):1:i+floor(
                size1/2),j-floor(size1/2):1:j+
                floor(size1/2))
138         y=gsort(t);
139         temp2(i,j)=min(y);
140     end
141 end
142 case 'max' then
143     for i=ceil(size1/2):nr+ceil(size1/2)-1
144         for j=ceil(size1/2):nc+ceil(size1/2)-1
145             t=temp(i-floor(size1/2):1:i+floor(
                    size1/2),j-floor(size1/2):1:j+
                    floor(size1/2))
146             y=gsort(t);
147             temp2(i,j)=max(y);
148         end
149     end
150 end
151 nn=temp2(ceil(size1/2):nr+ceil(size1/2)-1,ceil(
        size1/2):nc+ceil(size1/2)-1)
152 resimg=mat2gray(nn)
153 endfunction
154
155
156
157 function [f] = spfilt(g,type1, m,n,parameter)
158     if argn(2) ==2 then
159         m=3;n=3;Q=1.5;d=2;
160     elseif argn(2)==5 then
161         Q=parameter;d=parameter;
162     elseif argn(2)==4 then
163         Q=1.5;d=2;
164     else
165         disp('wrong number of inputs');
166     end
167     select type1
168     case 'amean' then
169         filtersize=m

```

```

170         w=fspecial('average',filtersize);
171         f=imfilter(g,w);
172     case 'gmean1' then
173         f=gmean1(g,m,n);
174     case 'gmean2' then
175         f=gmean2(g,m,n);
176     case 'hmean' then
177         f=harmean1(g,m,n);
178
179     case 'charmean' then
180         f=charmean1(g,m,n,Q);
181     case 'median' then
182         filtersize = [m n];
183         f=MedianFilter(g,filtersize);
184         // f=medfilt2(g,[m n],'symmetric');
185
186     case 'max' then
187         f=ordfilt2(g,m,n,'max');
188
189     case 'min' then
190         f=ordfilt2(g,m,n,'min');
191
192     case 'midpoint' then
193         f1=ordfilt2(g,m,n,'max');
194         f2=ordfilt2(g,m,n,'min');
195         f=imlincomb(0.5,f1,0.5,f2);
196
197     case 'atrimmed' then
198         if (d<0) |(d/2 ~=round(d/2)) then
199             disp('d must be a nonnegative, even
200                 integer');
201             end
202             disp(d)
203             d=2;
204             f=alphatrim(g,m,n,d);
205     else
206         disp('Unknown filter type.');
```

```

207 endfunction
208
209
210 v=imnoise(im,'salt & pepper',0.02)
211 figure,ShowImage(v,'Salt & pepper noise corrupted
    image');//ShowColorImage() is used to show color
    image, figure is command to view images in
    separate window.
212 title('Salt & pepper noise corrupted image') ;//
    title() is used for providing a title to an
    image.
213
214
215 h=spfilt(v,'median',3,3)
216 figure,ShowImage(h,'Result of filtering with a
    median filter of size 3*3');//ShowColorImage() is
    used to show color image, figure is command to
    view images in separate window.
217 title('Result of filtering with a median filter of
    size 3*3') ;//title() is used for providing a
    title to an image.
218
219
220
221 h=spfilt(v,'amean',3)
222 figure,ShowImage(h,'Result of filtering with an
    arithmetic mean filter of size 3*3');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
223 title('Result of filtering with an arithmetic mean
    filter of size 3*3') ;//title() is used for
    providing a title to an image.
224
225
226
227 i=spfilt(v,'max',3,3)
228 figure,ShowImage(i,'Result of filtering with a max

```

```

    filter of size 3*3'); // ShowColorImage() is used
    to show color image, figure is command to view
    images in separate window.
229 title('Result of filtering with a median filter of
    size 3*3'); // title() is used for providing a
    title to an image.
230
231 // Filtering the corrupted image with midpoint filter
232 j=spfilt(v, 'midpoint')
233 figure, ShowImage(j, 'Result of filtering with a
    Midpoint filter of size 3*3'); // ShowColorImage()
    is used to show color image, figure is command to
    view images in separate window.
234 title('Result of filtering with a Midpoint filter of
    size 3*3') ; // title() is used for providing a
    title to an image.
235
236 j=spfilt(v, 'min', 3, 3)
237 figure, ShowImage(j, 'Result of filtering with a Min
    filter of size 3*3'); // ShowColorImage() is used
    to show color image, figure is command to view
    images in separate window.
238 title('Result of filtering with a Min filter of size
    3*3'); // title() is used for providing a title
    to an image.
239
240
241 km=spfilt(v, 'atrimmed')
242 figure, ShowImage(km, 'Result of filtering with a
    Alpha-trimmed mean filter of size 3*3'); //
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
243 title('Result of filtering with a Alpha-trimmed mean
    filter of size 3*3') ; // title() is used for
    providing a title to an image.
244
245 s = imcrop(v, [1, 1, 300, 300]);

```

```

246 s = v;
247
248 l=spfilt(s, 'gmean1',3,3)
249 figure,ShowImage(l,'Result of filtering with a
    Geometric mean filter of size 3*3');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
250 title('Result of filtering with a Geometric mean
    filter of size 3*3') ;//title() is used for
    providing a title to an image.;
251
252 l=spfilt(s, 'gmean2',3,3)
253 figure,ShowImage(l,'Result of filtering with a
    Geometric mean filter of size 3*3');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
254 title('Result of filtering with a Geometric mean
    filter of size 3*3') ;//title() is used for
    providing a title to an image.;
255
256 m=spfilt(s, 'hmean',3,3)
257 figure,ShowImage(m,'Result of filtering with a
    Harmonic mean filter of size 3*3');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
258 title('Result of filtering with a Harmonic mean
    filter of size 3*3');//title() is used for
    providing a title to an image.;
259
260
261
262 n=spfilt(v, 'charmearn',3,3,1)
263 figure,ShowImage(n,'Result of filtering with a
    Contra Harmonic mean filter of size 3*3 and order
    of filter is 1');//ShowColorImage() is used to

```

```
    show color image, figure is command to view
    images in separate window.
264 title('Result of filtering with a Contra Harmonic
    mean filter of size 3*3 and order of filter is 1'
    );//title() is used for providing a title to an
    image.
```

Experiment: 6

Color Image Processing

Scilab code Solution 6.1 Color Image Processing Fundamentals

```
1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox:Huffcomp Toolbox 1.1.1
7 //OS:Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11 clc //to clear command window
12 clear all //to kill previously defined variables
13 xdel(winsid())//to close all currently open figure(s
    ).
14
15
16
17 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed. getSIVPpath()
    is pre-defined function in SIVP toolbox.
```

```

18 rgb = imread(SIVP_PATH + 'images/lena.png'); //
    Reading from sub-directory of images.
19 figure, ShowColorImage(rgb, 'Original color image'); //
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
20 title('Original Color Image'); // title() is used for
    providing a title to an image.
21
22
23 R=rgb(:,:,1); // Separation of red component from
    image
24 figure, ShowImage(R, 'Red component separation from
    original image'); // ShowColorImage() is used to
    show color image, figure is command to view
    images in separate window.
25 title('Red component separation from original image'
    ); // title() is used for providing a title to an
    image.
26
27
28
29 G=rgb(:,:,2); // Separation of green component from
    image
30 figure, ShowImage(R, 'Green comonent separation from
    original image'); // ShowColorImage() is used to
    show color image, figure is command to view
    images in separate window.
31 title('Green component separation from original
    image'); // title() is used for providing a title
    to an image.
32
33
34 B=rgb(:,:,3); // Separation of blue component from
    image
35 figure, ShowImage(R, 'Blue component separation from
    original image'); // ShowColorImage() is used to
    show color image, figure is command to view

```

```

    images in separate window.
36 title('Blue component separation from original image
    ');//title() is used for providing a title to an
    image.
37
38 [k,map]=RGB2Ind(rgb);//RGB image is converted to an
    indexed image
39 figure,ShowImage(k,'Indexed image',map);//map matrix
    is colormap matrix of an RGB image
40
41 rg=Ind2RGB(k,map);//to convert indexed image to true
    color image.
42 figure,ShowColorImage(rg,'RGB image is converted
    from an indexed image');//ShowColorImage() is
    used to show color image, figure is command to
    view images in separate window.
43 title('RGB image is converted from an indexed image'
    );//title() is used for providing a title to an
    image.
44
45 function [y]=ind2gray(x,map)
46 rgb=Ind2RGB(x,map);//to convert an indexed image
    into RGB image
47 yiq=rgb2ntsc(rgb);//RGB image is converted into NTSC
    color system.
48 y=yiq(:,:,1);
49 endfunction
50
51 t=ind2gray(k,map);
52 figure,ShowImage(t,'Indexed image is converted to
    grayscale');//ShowColorImage() is used to show
    color image, figure is command to view images in
    separate window.
53 title('Indexed image is converted to grayscale');//
    title() is used for providing a title to an
    image.
54
55

```

```

56 yiq=rgb2ntsc(rgb); //YIQ components are obtained from
    RGB components of an image.
57 figure, ShowColorImage(yiq, 'Luminance(Y), Hue(I) and
    Saturation(Q) components of a RGB image'); //
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
58 title('Luminance(Y), Hue(I) and Saturation(Q)
    components of a RGB image'); //title() is used for
    providing a title to an image.
59
60 rgb1=ntsc2rgb(yiq); //RGB components are obtained
    from YIQ components of an image.
61 figure, ShowColorImage(rgb1, 'RGB components are
    obtained from NTSC format'); //ShowColorImage() is
    used to show color image, figure is command to
    view images in separate window.
62 title('RGB components are obtained from NTSC format'
    ); //title() is used for providing a title to an
    image.
63
64 ycbcr=rgb2ycbcr(rgb); //YCbCr components are obtained
    from RGB components of an image.
65 figure, ShowColorImage(ycbcr, 'Luminance(Y) ,two color
    difference components Cb and Cr components of a
    RGB image'); //ShowColorImage() is used to show
    color image, figure is command to view images in
    separate window.
66 title('Luminance(Y) ,two color difference components
    Cb and Cr components of a RGB image'); //title()
    is used for providing a title to an image.
67
68 rgb2=ycbcr2rgb(ycbcr);
69 figure, ShowColorImage(rgb2, 'RGB components are
    obtained from YCbCR color space'); //
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.

```

```
70 title ('RGB components are obtained from YCbCr color
    space'); //title() is used for providing a title
    to an image.
```

Scilab code Solution 6.2 Color Image Processing

```
1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox: Huffcomp Toolbox 1.1.1
7 //OS: Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11 clc //to clear command window.
12 clear all //to kill previously defined variables.
13 xdel(winsid()) //to close all currently open figure(s
    ).
14 //
15
16 //stacksize() is used to increase stack size to
    achieve maximum performance. Restart Scilab if
    error no 10001(cannot allocate memory) is occurred
    .
17 stacksize('max')
18
19
20 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed. getSIVPpath()
    is pre-defined function in SIVP toolbox.
21 rgb = imread(SIVP_PATH + 'images/lena.png'); //
    Reading from sub-directory of images.
```

```

22 figure, ShowColorImage(rgb, 'Original color image'); //
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
23 title('Original Color Image'); //title() is used for
    providing a title to an image.
24
25
26 hsv=rgb2hsv(rgb); //HSV components are obtained from
    RGB components of an image using rgb2hsv().
27 figure, ShowColorImage(hsv, 'HSV components of a RGB
    image'); //ShowColorImage() is used to show color
    image, figure is command to view images in
    separate window.
28 title('HSV components of a RGB image'); //title() is
    used for providing a title to an image.
29
30 rgb3=hsv2rgb(hsv); //RGB components are obtained from
    HSV color system using hsv2rgb().
31 figure, ShowColorImage(rgb3, 'RGB componenta are
    obtained from HSV color system'); //ShowColorImage
    () is used to show color image, figure is command
    to view images in separate window.
32 title('RGB componenta are obtained from HSV color
    system'); //title() is used for providing a title
    to an image.
33
34 cmy=imcomplement(rgb); //cyan magenta and yellow
    colors are obtained from an image using
    imcomplement().
35 figure, ShowColorImage(cmy, 'cyan magenta and yellow
    are obtained from the RGB image'); //
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
36 title('cyan magenta and yellow are obtained from the
    RGB image'); //title() is used for providing a
    title to an image.

```

```

37
38 rgb4=imcomplement(cmy); //RGB components are obtained
    from cyan magenta and yellow components if an
    image.
39 figure, ShowColorImage(rgb4, 'RGB components are
    obtained from cyan magenta and yellow components'
    ); //ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
40 title('RGB components are obtained from cyan magenta
    and yellow components'); //title() is used for
    providing a title to an image.
41
42
43
44 rgb=im2double(rgb)
45 function [hsi]=rgb2hsi(rgb) //rgb2hsi() is user
    defined function, which is used for converting
    RGB format to HSI color model.
46
47     r = rgb(:, :, 1); //separating Red, Green and
        Blue components from the RGB image.
48     g = rgb(:, :, 2);
49     b = rgb(:, :, 3);
50     num = 0.5*((r - g) + (r - b));
51     den1=(r - g).^2 + (r - b).*(g - b)
52     den = den1.^0.5;
53     theta = acos(num./(den+%eps)); // %eps is minimum
        number to avoid divide by zero exception.
54     H = theta;
55     H(b > g) = 2*pi - H(b > g);
56     H = H/(2*pi);
57
58     num = min(min(r, g), b);
59     den = r + g + b;
60     den(den == 0) = %eps;
61     S = 1 - 3.* num./den;
62

```

```

63     H(S == 0) = 0;
64
65     I = (r + g + b)/3;
66
67     hsi = rgb;
68     hsi(:,:,1)=H;
69     hsi(:,:,2)=S;
70     hsi(:,:,3)=I;
71
72     endfunction
73     hsi=rgb2hsi(rgb);
74
75     figure, ShowColorImage(hsi, 'HSI_image');//
        ShowColorImage() is used to show color image,
        figure is command to view images in separate
        window.
76     title('HSI_image');//title() is used for providing
        a title to an image.
77
78     function [rgb]=hsi2rgb(hsi)//hsi2rgb() is user
        defined function, which is used for converting HSI
        format to RGB color model.
79
80
81     H = hsi(:, :, 1) * 2 * %pi;
82     S = hsi(:, :, 2);
83     I = hsi(:, :, 3);
84
85     [nr nc]=size(H)//to find dimension of an image.
86
87
88     R = zeros(nr,nc);
89     G = zeros(nr,nc);
90     B = zeros(nr,nc);
91
92
93     idx = find( (0 <= H) & (H < 2*%pi/3));
94     B(idx) = I(idx) .* (1 - S(idx));

```

```

95     R(idx) = I(idx) .* (1 + S(idx) .* cos(H(idx)) ./
        cos(%pi/3 - H(idx)));
96     G(idx) = 3*I(idx) - (R(idx) + B(idx));
97
98
99     idx = find( (2*%pi/3 <= H) & (H < 4*%pi/3) );
100    R(idx) = I(idx) .* (1 - S(idx));
101    G(idx) = I(idx) .* (1 + S(idx) .* cos(H(idx) -
        2*%pi/3) ./cos(%pi - H(idx)));
102    B(idx) = 3*I(idx) - (R(idx) + G(idx));
103
104
105    idx = find( (4*%pi/3 <= H) & (H <= 2*%pi));
106    G(idx) = I(idx) .* (1 - S(idx));
107    B(idx) = I(idx) .* (1 + S(idx) .* cos(H(idx) -
        4*%pi/3) ./cos(5*%pi/3 - H(idx)));
108    R(idx) = 3*I(idx) - (G(idx) + B(idx));
109
110
111    rgb = hsi;
112    rgb(:,:,1)=R;
113    rgb(:,:,2)=G;
114    rgb(:,:,3)=B;
115    rgb = max(min(rgb, 1), 0);
116
117    endfunction
118
119
120    rgb=hsi2rgb(hsi);
121
122    figure,ShowColorImage(rgb,'RGB components are
        obtained from hsi color system');//ShowColorImage
        () is used to show color image, figure is command
        to view images in separate window.
123    title('RGB components are obtained from HSI color
        system');//title() is used for providing a title
        to an image.

```

Experiment: 7

Image Compression

Scilab code Solution 7.1 Image Compression

```
1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox:Huffcomp Toolbox 1.1.1
7 //OS:Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11 clc //to clear command window.
12 clear all //to kill previously defined variables.
13 xdel(winsid())//to close all currently open figure(s
    ).
14
15 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed. getSIVPpath()
    is pre-defined function in SIVP toolbox.
16 rgb = imread(SIVP_PATH + 'images/lena.png');//
    Reading from sub-directory of images.
```

```

17 figure, ShowColorImage(rgb, 'Original color image'); //
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
18 title('Original Color Image'); //title() is used for
    providing a title to an image.
19
20 im=rgb2gray(rgb); //to convert a RGB image into
    grayscale image.
21
22 M=8;
23 N=8;
24 [nr,nc]=size(im);
25
26
27 p=imcrop(im,[1,1,8,8]); //image is divided into 8*8
    blocks for computation of DCT.
28 p=double(p);
29 p1=p-128;
30 f=dct(p1); //dct() is used to find DCT
    transformations.
31 //q is Luminance quantization matrix.
32 q=[16 11 10 16 24 40 51 61
33 12 12 14 19 26 58 60 55
34 14 13 16 24 40 57 69 56
35 14 17 22 29 51 87 80 62
36 18 22 37 56 68 109 103 77
37 24 35 55 64 81 104 113 92
38 49 64 78 87 103 121 120 101
39 72 92 95 98 112 100 103 99];
40 //Each DCT coefficient f(u,v) is divided by the
    corresponding quantizer step-size parameter q(u,v
    ) in the quantization matrix and rounded to the
    nearest integer
41 t=f./q;
42 g=floor(t)
43
44

```

```

45 function [out]=zigzag(in)//zigzag() is function to
    find a vector sorted by the criteria of the
    spatial frequency
46     [num_rows num_cols]=size(in);
47
48     // Initialise the output vector
49     out=zeros(1,num_rows*num_cols);
50
51     cur_row=1;  cur_col=1;  cur_index=1;
52
53     // First element
54     //out(1)=in(1,1);
55
56     while cur_row<=num_rows & cur_col<=num_cols
57         if cur_row==1 & modulo(cur_row+cur_col,2)
           ==0 & cur_col~=num_cols then
58             out(cur_index)=in(cur_row,cur_col);
59             cur_col=cur_col+1;
           //move right at the top
60             cur_index=cur_index+1;
61
62         elseif cur_row==num_rows & modulo(cur_row+
           cur_col,2)~=0 & cur_col~=num_cols then
63             out(cur_index)=in(cur_row,cur_col);
64             cur_col=cur_col+1;
           //move right at the bottom
65             cur_index=cur_index+1;
66
67         elseif cur_col==1 & modulo(cur_row+cur_col
           ,2)~=0 & cur_row~=num_rows then
68             out(cur_index)=in(cur_row,cur_col);
69             cur_row=cur_row+1;
           //move down at the left
70             cur_index=cur_index+1;
71
72         elseif cur_col==num_cols & modulo(cur_row+
           cur_col,2)==0 & cur_row~=num_rows then
73             out(cur_index)=in(cur_row,cur_col);

```

```

74         cur_row=cur_row+1;
           //move down at the right
75         cur_index=cur_index+1;
76
77     elseif cur_col~=1 & cur_row~=num_rows &
           pmodulo(cur_row+cur_col,2)~=0 then
78         out(cur_index)=in(cur_row,cur_col);
79         cur_row=cur_row+1;      cur_col=cur_col
           -1; //move diagonally left down
80         cur_index=cur_index+1;
81
82     elseif cur_row~=1 & cur_col~=num_cols &
           pmodulo(cur_row+cur_col,2)==0 then
83         out(cur_index)=in(cur_row,cur_col);
84         cur_row=cur_row-1;      cur_col=cur_col
           +1; //move diagonally right up
85         cur_index=cur_index+1;
86
87     elseif cur_row==num_rows & cur_col==num_cols
           //obtain the bottom right element
88         out(M*N)=in(8,8);
           //end of the operation
89         break
           //terminate the operation
90     else
91     end
92 end
93
94 endfunction
95
96 out=zigzag(g);
97 disp(out)

```

Scilab code Solution 7.2 Image Compression

```

1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox: Huffcomp Toolbox 1.1.1
7 //OS: Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11 clc //to clear command window.
12 clear all //to kill previously defined variables.
13 xdel(winsid())//to close all currently open figure(s
    ).
14
15
16 nn=[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0
    0 3 0 4 4 4 5 0 0 0 0 5 0 0 0];
17 function [rle]=relencoder(nn)//run length encoder
18
19     k=1;
20     n= [nn 1]; // dummy ending
21
22     t=0;
23     for i = 1:length(n)-1
24         if n(i)==0 then
25             t=t+1;
26             if t==15 then
27                 r(k,1) =t;
28                 r(k,2)=0;
29                 k=k+1;
30                 t=0;
31             end
32         else
33             valuecode = n(i);
34             lengthcode = t;
35             r(k,1) =lengthcode;

```

```

36             r(k,2)=valuecode;
37             k=k+1;
38             t=0;
39         end
40     end
41     rle=r;
42     rle($+1,:)=0;
43     disp(rle)
44     disp(size(rle))
45 endfunction
46
47 rr=relencoder(nn);

```

Scilab code Solution 7.3 Image Compression

```

1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox: Huffcomp Toolbox 1.1.1
7 //OS: Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11
12 //this code is for huffman encoding using toolbox.
13 clc //to clear command window.
14 clear all //to kill previously defined variables.
15 xdel(winsid())//to close all currently open figure(s
    ).// Generate a Testmatrix
16
17 sp=sparse
    ([1,1;1,2;1,3;1,4;1,5;1,6],[5,7,10,15,20,45])

```

```
18 [SB,h,L,QM]=huffman(sp);
19 //SB contains the symbols
20 disp(SB);
21 // h is the normalized histogram
22 disp(h);
23 // L contains the number of bits used for the
    symbols
24 disp(L);
25 // QM is the complete code table ,
26 // containing symbol, bits and no. of bits
27 disp(QM);
28 disp(sp)
```

Experiment: 8

Morphological Image Processing

check Appendix ?? for dependency:

82.tif

check Appendix ?? for dependency:

wirebondmask.tif

Scilab code Solution 8.1 Morphological Image Processing

```
1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox: Huffcomp Toolbox 1.1.1
7 //OS: Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11 clc //to clear command window.
```

```

12 clear all //to kill previously defined variables.
13 xdel(winsid())//to close all currently open figure(s
    ).
14
15
16 a=[0 0;0 1]
17 b=[0 1;1 1]
18
19
20
21 Image=imread('wirebondmask.tif');
22
23 StructureElement = CreateStructureElement('square',
    9);//CreateStructureElement() is used to create
    structuring element.First parameter is used to
    create square structuring elemnt of size 9
24
25 ResultImage = ErodeImage(Image, StructureElement);//
    ErodeImage() is used to perform erosion of an
    image by structuring element.
26 figure,ShowImage(ResultImage,'Erosion of an image by
    square structuring element.');//ShowColorImage()
    is used to show color image, figure is command
    to view images in separate window.
27 title('Erosion of an image by square structuring
    element.');//title() is used for providing a
    title
28
29
30 Image=imread('82.tif');
31 figure,ShowImage(Image,'Text with broken characters.
    ');//ShowImage() is used to show color image,
    figure is command to view images in separate
    window.
32 title('Text with broken characters.');//title() is
    used for providing a title
33
34 StructureElement1 = CreateStructureElement('square',

```

```

    3); //CreateStructureElement() is used to create
    structuring element. First parameter is used to
    create square structuring element of size 3.
35 ResultImage1 = DilateImage(Image, StructureElement1)
    ; //DilateImage() is used to perform erosion of an
    image by structuring element.
36 figure, ShowImage(ResultImage1, 'Dilation of an image
    by square structuring element. Broken segments
    were joined. '); //ShowImage() is used to show
    color image, figure is command to view images in
    separate window.
37 title('Dilation of an image by square structuring
    element. Broken segments were joined. '); //title()
    is used for providing a title
38
39 StructureElement2= CreateStructureElement('square',
    3); //CreateStructureElement() is used to create
    structuring element. First parameter is used to
    create square structuring element of size 9
40 StructureElement2.Data=[%f %t %f;%t %t %t;%f %t %f]
41
42 ResultImage2 = DilateImage(Image, StructureElement2);
    //DilateImage() is used to perform erosion of an
    image by structuring element.
43 figure, ShowImage(ResultImage2, 'Dilation of an image
    by square structuring element. Broken segments
    were joined. '); //ShowImage() is used to show
    color image, figure is command to view images in
    separate window.
44 title('Dilation of an image by square structuring
    element. Broken segments were joined. '); //title()
    is used for providing a title
45
46
47 ResultImage3 = OpenImage(Image, StructureElement1); //
    //OpenImage() is used to open an image by
    structuring element.
48 figure, ShowImage(ResultImage3, 'Opening of an image.'

```

```

        );//ShowColorImage() is used to show color image,
        figure is command to view images in separate
        window.
49 title('Opening of an image.');//title() is used for
        providing a title
50
51 ResultImage4 = CloseImage(Image,StructureElement1);
52 figure,ShowImage(ResultImage4,'Closing of an image.'
        );//ShowColorImage() is used to show color image,
        figure is command to view images in separate
        window.
53 title('Closing of an image.');//title() is used for
        providing a title
54
55
56 image1=imread('83.tif');
57 image2=imcomplement(image1);//imcomplement() is used
        to find complement of an image.
58
59 function [result]=hitmiss(i,se1,se2)//hitmiss() is
        used to perform hit and miss transform,which is
        used for shape detection.
60         e=imcomplement(i);//imcomplement() is used to
        find complement of an image.
61 c=ErodeImage(e,se2);//ErodeImage() is used to erode
        an image by structuring element.
62 b=ErodeImage(i,se1)////ErodeImage() is used to
        erode an image by structuring element.
63 result1=b&c;
64 disp(result1)
65 [nr,nc]=size(i);
66 idx=find(result1==%t);
67 result=zeros(nr,nc)
68 result(idx)=255;
69 endfunction
70
71 i=[ 0.    0.    0.    0.    0.    0.    0.    0.
        0.

```

```

72     0.    0.    1.    0.    0.    0.    1.    0.
       0.
73     0.    1.    1.    1.    0.    1.    1.    1.
       0.
74     0.    0.    1.    0.    0.    0.    1.    1.
       0.
75     0.    0.    0.    0.    0.    0.    0.    0.
       0.
76     0.    0.    0.    0.    0.    0.    0.    0.
       0.]
77
78 se1= CreateStructureElement('square', 3);
79 se1.Data=[%f %t %f;%t %t %t;%f %t %f]
80
81 se2= CreateStructureElement('square', 3);
82 se2.Data=[%t %f %t;%f %f %f;%t %f %t];
83
84 figure,ShowImage(i,'Original binary image');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
85 title('Original binary image');//title() is used for
    providing a title
86
87 Result=hitmiss(i,se1,se2); //hitmiss() is used to
    perform hit and miss transform,which is used for
    shape detection.
88 figure,ShowImage(Result,'Result of Hit and Miss
    Transform.');//ShowColorImage() is used to show
    color image, figure is command to view images in
    separate window.
89 title('Result of Hit and Miss Transform.');//title()
    is used for providing a title
90
91 //boundary extraction
92
93 se1= CreateStructureElement('square', 3);
94 Image=imread('wirebondmask.tif');

```

```
95 result1=ErodeImage(Image,se1);
96 result=Image-result1;
97 figure,ShowImage(Image,'Original binary image');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
98 title('Original binary image.');//title() is used
    for providing a title
99
100 figure,ShowImage(result,'Extracted boundary of an
    image.');//ShowColorImage() is used to show color
    image, figure is command to view images in
    separate window.
101 title('Extracted boundary of an image.');//title()
    is used for providing a title
```

Experiment: 9

Image Segmentation

check Appendix ?? for dependency:

`building.tif`

check Appendix ?? for dependency:

`turbineblade.tif`

check Appendix ?? for dependency:

`wirebondmask.tif`

Scilab code Solution 9.1 Image Segmentation

```
1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox:Huffcomp Toolbox 1.1.1
7 //OS:Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
```

```

11 clc //to clear command window.
12 clear all //to kill previously defined variables.
13 xdel(winsid())//to close all currently open figure(s
    ).
14
15 Image=ReadImage('turbineblade.tif');
16 figure,ShowImage(Image,'Gray scale image with a
    isolated black point.');//ShowColorImage() is
    used to show color image, figure is command to
    view images in separate window.
17 title('Gray scale image with a isolated black point.
    ');//title() is used for providing a title to an
    image.
18
19 image=double(Image);
20 mask =[-1 -1 -1;-1 8 -1;-1 -1 -1];
21
22 res = imfilter(image,mask);
23 g=abs(res);
24 t=max(g);
25 g=g>=t;
26 figure,ShowImage(g,'Detection of point');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
27 title('Detection of point.');//title() is used for
    providing a title to an image.
28
29
30 hmask=[-1 -1 -1;2 2 2;-1 -1 -1];//mask for
    horizontal line detection.
31 vmask=[-1 2 -1;-1 2 -1;-1 2 -1];//mask for vertical
    line detection.
32 dlmask=[-1 -1 2;-1 2 -1;2 -1 -1];//mask for +45
    degree line detection.
33 dlmask2=[2 -1 -1;-1 2 -1;-1 -1 2];//mask for -45
    degree line detection.
34

```

```

35 Image=ReadImage('wirebondmask.tif')
36 figure,ShowImage(Image,'Image of a wire-bond mask');
    //ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
37 title('Image of a wire-bond mask.');//title() is
    used for providing a title to an image.
38 image=double(Image);
39 g1=imfilter(image,hmask);
40 figure,ShowImage(g1,'Result of processing with
    horizontal line detector mask.');//ShowColorImage
    () is used to show color image, figure is command
    to view images in separate window.
41 title('Result of processing with horizontal line
    detector mask.');//title() is used for providing
    a title to an image.
42
43 g2=imfilter(image,vmask);
44 figure,ShowImage(g2,'Result of processing with
    vertical degree line detector mask.');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
45 title('Result of processing with vertical degree
    line detector mask.');//title() is used for
    providing a title to an image.
46
47 g3=imfilter(image,dmask);
48 figure,ShowImage(g3,'Result of processing with +45
    degree line detector mask.');//ShowColorImage()
    is used to show color image, figure is command to
    view images in separate window.
49 title('Result of processing with +45 degree line
    detector mask.');//title() is used for providing
    a title to an image.
50
51 g4=imfilter(image,dmask2)
52 figure,ShowImage(g4,'Result of processing with -45

```

```

    degree line detector mask.');//ShowColorImage()
    is used to show color image, figure is command to
    view images in separate window.
53 title('Result of processing with -45 degree line
    detector mask.');//title() is used for providing
    a title to an image.
54
55
56 Image=ReadImage('building.tif')
57 figure,ShowImage(Image,'Image of a Building');//
    ShowColorImage() is used to show color image,
    figure is command to view images in separate
    window.
58 title('Image of a building.');//title() is used for
    providing a title to an image.
59
60 e=edge(image,'sobel');//edge() is used to detect an
    edge in grayscale image.Second argument'sobel' is
    used for Sobel edge detector.
61 figure,ShowImage(e,'Edge detection using Sobel
    approximation.');//ShowColorImage() is used to
    show color image, figure is command to view
    images in separate window.
62 title('Edge detection using Sobel approximation.');//
    //title() is used for providing a title to an
    image.
63
64 e=edge(image,'prewitt');//edge() is used to detect
    an edge in grayscale image.Second argument'
    prewitt' is used for prewitt edge detector.
65 figure,ShowImage(e,'Edge detection using Prewitt
    approximation.');//ShowColorImage() is used to
    show color image, figure is command to view
    images in separate window.
66 title('Edge detection using Prewitt approximation.');//
    ;//title() is used for providing a title to an
    image.
67

```

```
68 e=edge(image,'canny');//edge() is used to detect an
    edge in grayscale image.Second argument 'canny' is
    used for Canny edge detector.
69 figure,ShowImage(e,'Edge detection using Canny edge
    detector.');//ShowColorImage() is used to show
    color image, figure is command to view images in
    separate window.
70 title('Edge detection using Canny approximation.');//
    //title() is used for providing a title to an
    image.
```

Experiment: 10

Wavelets

check Appendix ?? for dependency:

woman.bmp

Scilab code Solution 10.1 Wavelets

```
1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox:Huffcomp Toolbox 1.1.1
7 //OS:Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11 clc //to clear command window.
12 clear all //to kill previously defined variables.
13 xdel(winsid())//to close all currently open figure(s
    ).
14
```

```

15 //stacksize() is used to increase stack size to
    achieve maximum performance.Restart Scilab if
    error no 10001(cannot allocate memory) is occured
    .
16 stacksize('max')
17
18
19
20 r=ReadImage('woman.bmp');//ReadImage() is used to
    read an image
21 X=rgb2gray(r);//rgb2gray() is used to convert an
    image into gray scale.
22 figure,ShowImage(X,'A simple test image');//
    ShowImage() is used to show gray scale image,
    figure is command to view images in separate
    window.
23 title('A simple test image');//title() is used for
    providing a title to an image.
24
25 wname = 'haar'//wname is used in two dimension
    multiple level discrete fast wavelet transform. '
    haar' is used for Haar wavelet.
26 [nr,nc]=size(r);
27 x=zeros(nr,nc);
28 x=X(1:$,1:$);
29 x=double(x);
30
31 [wc,s] = wavedec2(x,2,wname);// Compute a 2-level
    decomposition of the image using the Haar filters
    .Second argument is used for level.
32
33 // Extract the level 1 coefficients. Fourth argument
    must be 1 to for extraction the level 1
    coefficient in appcoef2() and detcoef2() function
    .
34 a1 = appcoef2(wc,s,wname,1);//appcoef2() is used to
    extract 2-D approximation coefficients.

```

```

35 h1 = detcoef2('h',wc,s,1); //detcoef2() is used to
    extract 2-D detail coefficient extraction. First
    argument 'h' is used for horizontal detail
    coefficeint.
36 v1 = detcoef2('v',wc,s,1); //detcoef2() is used to
    extract 2-D detail coefficient extraction. First
    argument 'v' is used for vertical detail
    coefficeint.
37 d1 = detcoef2('d',wc,s,1); //detcoef2() is used to
    extract 2-D detail coefficient extraction. First
    argument d is used for diagonal detail
    coefficient.

38
39 // Extract the level 2 coefficients. Fourth argument
    must be 2 to for extraction the level 2
    coefficient in appcoef2() and detcoef2() function
    .

40
41 a2 = appcoef2(wc,s,wname,2);
42 h2 = detcoef2('h',wc,s,2);
43 v2 = detcoef2('v',wc,s,2);
44 d2 = detcoef2('d',wc,s,2);
45
46 // Display the decomposition up to level 1 only.
47 a1=double(a1);
48 cod_a1 = wcodemat(a1,256);
49 cod_a1=double(cod_a1);
50 cod_a1 = wkeep(cod_a1,[256 256]);
51 cod_h1 = wcodemat(h1,260);
52 cod_h1=double(cod_h1);
53 cod_h1 = wkeep(cod_h1, [256 256]);
54 cod_v1 = wcodemat(v1,260);
55 cod_v1=double(cod_v1);
56 cod_v1 = wkeep(cod_v1, [256 256]);
57 cod_d1 = wcodemat(d1,260);
58 cod_d1=double(cod_d1);
59 cod_d1 = wkeep(cod_d1, [256 256]);
60 ans1=[cod_a1 ,cod_h1;cod_v1 ,cod_d1];

```

```

61
62 figure,ShowImage(ans1,'Wavelet Transform of an
    image');//ShowImage() is used to show gray scale
    image, figure is command to view images in
    separate window.
63 title('Wavelet Transform of an image');//title() is
    used for providing a title to an image.
64
65
66
67
68 // Display the entire decomposition upto level 2.
69 cod_a2 = wcodemat(a2,260);
70 cod_a2=double(cod_a2);
71 cod_a2 = wkeep(cod_a2, [128 128]);
72 cod_h2 = wcodemat(h2,260);
73 cod_h2=double(cod_h2);
74 cod_h2 = wkeep(cod_h2, [128 128]);
75 cod_v2 = wcodemat(v2,260);
76 cod_v2=double(cod_v2);
77 cod_v2 = wkeep(cod_v2, [128 128]);
78 cod_d2 = wcodemat(d2,260);
79 cod_d2=double(cod_d2);
80 cod_d2 = wkeep(cod_d2, [128 128]);
81 bb=[[cod_a2,cod_h2;cod_v2,cod_d2],cod_h1;cod_v1,
    cod_d1];
82
83
84 figure,ShowImage(bb,'Two scale Wavelet transform an
    image');//ShowImage() is used to show gray scale
    image, figure is command to view images in
    separate window.
85 title('Two scale Wavelet transform an image');//
    title() is used for providing a title to an
    image.
86
87

```

```

88 // Reconstruction of an image using wrcoef2()
    function. Last argument is used for level of
    reconstruction.
89 ra2 = wrcoef2('a',wc,s,wname,2); // 'a' is used for
    approximation coefficients.
90 rh2 = wrcoef2('h',wc,s,wname,2); // 'h' is used for
    horizontal detail coefficient.
91 rv2 = wrcoef2('v',wc,s,wname,2); // 'v' is used for
    vertical detail coefficient.
92 rd2 = wrcoef2('d',wc,s,wname,2); // 'd' is used for
    detail coefficient.
93
94 ra1 = wrcoef2('a',wc,s,wname,1);
95 rh1 = wrcoef2('h',wc,s,wname,1);
96 rv1 = wrcoef2('v',wc,s,wname,1);
97 rd1 = wrcoef2('d',wc,s,wname,1);
98
99 cod_ra2 = wcodemat(ra2,260);
100 cod_rh2 = wcodemat(rh2,260);
101 cod_rv2 = wcodemat(rv2,260);
102 cod_rd2 = wcodemat(rd2,260);
103 cod_ra1 = wcodemat(ra1,260);
104 cod_rh1 = wcodemat(rh1,260);
105 cod_rv1 = wcodemat(rv1,260);
106 cod_rd1 = wcodemat(rd1,260);
107
108
109
110 // Adding together the reconstructed average at
    level 2 and all of
111 // the reconstructed details gives the full
    reconstructed image.
112 Xhat = ra2 + rh2 + rv2 + rd2 + rh1 + rv1 + rd1;
113 X1=double(X);
114 X2=double(Xhat);
115 X3=max(max(abs(X1-X2))); //
116 disp(X3);
117 disp('Reconstruction error')

```

```

118
119 // Another way to reconstruct the image.
120
121 XXhat = waverec2(wc,s,wname); //waverec2() is used
    for two dimension multiple level inverse discrete
    transform.
122 X1=double(X);
123 X2=double(XXhat);
124 X3=max(max(abs(X1-X2)));
125 disp(X3);
126 disp('Reconstruction error (using waverec2)');
127 // Compression can be accomplished by applying a
    threshold to the wavelet coefficients.
128 thr = 20;
129 [X_comp,wc_comp,s_comp,perf0,perfL2] = wdencomp('gbl',
    wc,s,wname,2,thr,'h',1); //wdencomp() is used for
    de-noising or compression using wavelets. h
    means use hard thresholding. perfL2 is used to
    find energy recovery. perf0 is used to measure
    compression performance.
130
131 cod_X_comp = wcodemat(X_comp,260);
132
133 figure,ShowImage(cod_X_comp,'Compressed using
    global hard threshold'); //ShowImage() is used to
    show gray scale image, figure is command to view
    images in separate window.
134 title('Compressed using global hard threshold') //
    title() is used for providing a title to an
    image.
135 disp('Energy retained');
136 disp(perfL2);
137
138 disp('Null coefficients');
139 disp(perf0);

```

check Appendix ?? for dependency:

101.tif

Scilab code Solution 10.2 Wavelets

```
1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox: Huffcomp Toolbox 1.1.1
7 //OS: Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11 //edge detection
12 clc //to clear command window.
13 clear all //to kill previously defined variables.
14 xdel(winsid())//to close all currently open figure(s
    ).
15
16 //stacksize() is used to increase stack size to
    achieve maximum performance.Restart Scilab if
    error no 10001(cannot allocate memory) is occurred
    .
17 stacksize('max')
18
19
20 X=ReadImage('101.tif');//ReadImage() is used to read
    an image
21 figure,ShowImage(X,'Original Grayscale image');//
    ShowImage() is used to show gray scale image,
    figure is command to view images in separate
    window.
22 title('Original Grayscale Image');//title() is used
    for providing a title to an image.
```

```

23
24 [nr,nc]=size(X);
25 x=zeros(nr,nc);
26 x=X(1:$,1:$);
27 x=double(x);
28
29 [CA,CH,CV,CD]=dwt2(x,'db2');// one level
    decomposition
30 a=(CA-min(CA))/max(CA-min(CA));//normalize the
    approximation coefficients matrix for displaying.
31 h=(CH-min(CH))/max(CH-min(CH));//normalize the
    horizontal detail coefficients matrix for
    displaying.
32 v=(CV-min(CV))/max(CV-min(CV));//normalize the
    vertical detail coefficients matrix for
    displaying.
33 d=(CD-min(CD))/max(CD-min(CD));//normalize the
    diagonal detail coefficients matrix for
    displaying.
34 c1=[a';h']';
35 c2=[v';d']';
36 co=[c1;c2];
37 figure,ShowImage(co,'Original Grayscale image');//
    ShowImage() is used to show gray scale image,
    figure is command to view images in separate
    window.
38 title('Original Grayscale Image');//title() is used
    for providing a title to an image.
39
40 CA(1:$,1:$)=0;//zeroing approximation coefficients
41 a=CA;
42 c1=[a';h']';
43 c2=[v';d']';
44 co=[c1;c2];
45 figure,ShowImage(co,'Deleted approximation
    coefficients ');//ShowImage() is used to show
    gray scale image, figure is command to view
    images in separate window.

```

```

46 title('Deleted approximation coefficients');//title
    () is used for providing a title to an image.
47
48 x1=idwt2(CA,CH,CV,CD,'db2',[nr nc]);//idwt2() is
    used to find Two Dimension Inverse Discrete Fast
    Wavelet Transform.
49
50 figure,ShowImage(x1,' Reconstructed image after
    deleting approximation coefficients');//ShowImage
    () is used to show gray scale image, figure is
    command to view images in separate window.
51 title('Reconstred image after deleting approximation
    coefficients');//title() is used for providing
    a title to an image.
52
53
54 //horizontal line detection
55 [CA,CH,CV,CD]=dwt2(x,'db2');// one level
    decomposition
56 a=(CA-min(CA))/max(CA-min(CA));//normalize the
    approximation coefficients matrix for displaying.
57 h=(CH-min(CH))/max(CH-min(CH));//normalize the
    horizontal detail coefficients matrix for
    displaying.
58 v=(CV-min(CV))/max(CV-min(CV));//normalize the
    vertical detail coefficients matrix for
    displaying.
59 d=(CD-min(CD))/max(CD-min(CD));//normalize the
    diagonal detail coefficients matrix for
    displaying.
60 CA(1:$,1:$)=0;//zeroing the approximation
    coefficients matrix.
61 a=CA;
62 CH(1:$,1:$)=0;//zeroing the horizontal detail
    coefficients matrix.
63 h=CH;
64 c1=[a';h']';
65 c2=[v';d']';

```

```

66 co=[c1;c2];
67 figure,ShowImage(co,' Deleted approximation
    coefficients and horizontal detail coefficients '
    );//ShowImage() is used to show gray scale image,
    figure is command to view images in separate
    window.
68 title('deleted approximation coefficients and
    horizontal detail coefficients ');//title() is
    used for providing a title to an image.
69
70 x1=idwt2(CA,CH,CV,CD,'db2',[nr nc]);//idwt2() is
    used to find Two Dimension Inverse Discrete Fast
    Wavelet Transform.
71 figure,ShowImage(x1,' Reconstructed image after
    deleting approximation coefficients and
    horizontal detail coefficients ');//ShowImage()
    is used to show gray scale image, figure is
    command to view images in separate window.
72 title('Reconstructed image after deleting
    approximation coefficients and horizontal detail
    coefficients ');//title() is used for providing
    a title to an image.
73
74 //vertical line detection
75
76 [CA,CH,CV,CD]=dwt2(x,'db2');// one level
    decomposition
77 a=(CA-min(CA))/max(CA-min(CA));//normalize the
    approximation coefficients matrix for displaying.
78 h=(CH-min(CH))/max(CH-min(CH));//normalize the
    horizontal detail coefficients matrix for
    displaying.
79 v=(CV-min(CV))/max(CV-min(CV));//normalize the
    vertical detail coefficients matrix for
    displaying.
80 d=(CD-min(CD))/max(CD-min(CD));//normalize the
    diagonal detail coefficients matrix for
    displaying.

```

```

81 CA(1:$,1:$)=0;//zeroing the approximation
    coefficients matrix.
82 a=CA;
83 CV(1:$,1:$)=0;//zeroing the vertical detail
    coefficients matrix.
84 v=CV;
85 c1=[a';h']';
86 c2=[v';d']';
87 co=[c1;c2];
88 figure,ShowImage(co,' Deleted approximation
    coefficients and vertical detail coefficients ');
    //ShowImage() is used to show gray scale image,
    figure is command to view images in separate
    window.
89 title('Deleted approximation coefficients and
    vertical detail coefficients ');//title() is used
    for providing a title to an image.
90
91 x1=idwt2(CA,CH,CV,CD,'db2',[nr nc]);//idwt2() is
    used to find Two Dimension Inverse Discrete Fast
    Wavelet Transform.
92
93 figure,ShowImage(x1,' Reconstructed image after
    deleting approximation coefficients and vertical
    detail coefficients ');//ShowImage() is used to
    show gray scale image, figure is command to view
    images in separate window.
94 title('Reconstructed image after deleting
    approximation coefficients and vertical detail
    coefficients ');//title() is used for providing
    a title to an image.

```

check Appendix ?? for dependency:

pattern.tif

Scilab code Solution 10.3 Wavelets

```
1 //
2 //environment: Scilab 5.4.1
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.1-2
5 //Toolbox: Scilab Wavelet Toolbox0.1.19-1
6 //Toolbox: Huffcomp Toolbox 1.1.1
7 //OS: Windows 7
8 //
9 //Reference book name : Digital Image Processing
10 //book author: Rafael C. Gonzalez and Richard E.
    Woods
11
12 //wavelet based image smoothing.
13 clc //to clear command window.
14 clear all //to kill previously defined variables.
15 xdel(winsid())//to close all currently open figure(s
    ).
16
17 //stacksize() is used to increase stack size to
    achieve maximum performance.Restart Scilab if
    error no 10001(cannot allocate memory) is occurred
    .
18 stacksize('max')
19
20
21
22 X=ReadImage('pattern.tif');//ReadImage() is used to
    read an image
23 figure,ShowImage(X,'Original Grayscale image');//
    ShowImage() is used to show gray scale image,
    figure is command to view images in separate
    window.
24 title('Original Grayscale Image');//title() is used
    for providing a title to an image.
25
26
```

```

27 dwtmode('status')// 'status' is used to display
    current DWT Extension mode.
28 dwtmode('sym')// 'sym' is used for changing DWT
    Extension mode to half symmetrisation
29 wname = 'haar'//wname is used in two dimension
    multiple level discrete fast wavelet transform. '
    haar' is used for Haar wavelet.
30 [nr,nc]=size(X);
31 x=zeros(nr,nc);
32 x=X(1:$,1:$);
33 x=double(x);
34
35 [wc,s] = wavedec2(x,4,wname);// Compute a 2-level
    decomposition of the image using the Haar filters
    .Second argument is used for level.
36
37 // Extract the level 1 coefficients. Fourth argument
    must be 1 to for extraction the level 1
    coefficient in appcoef2() and detcoef2() function
    .
38 a1 = appcoef2(wc,s,wname,1);//appcoef2() is used to
    extract 2-D approximation coefficients.
39 h1 = detcoef2('h',wc,s,1);//detcoef2() is used to
    extract 2-D detail coefficient extraction. First
    argument 'h' is used for horizontal detail
    coefficients.
40 v1 = detcoef2('v',wc,s,1); //detcoef2() is used to
    extract 2-D detail coefficient extraction. First
    argument 'v' is used for vertical detail
    coefficients.
41 d1 = detcoef2('d',wc,s,1); //detcoef2() is used to
    extract 2-D detial coefficient extraction. First
    argument d is used for diagonal detail
    coefficients.
42
43 // Extract the level 2 coefficients. Fourth argument
    must be 2 for extraction the level 2
    coefficients in appcoef2() and detcoef2()

```

```

function .
44
45 a2 = appcoef2(wc,s,wname,2);
46 h2 = detcoef2('h',wc,s,2);
47 v2 = detcoef2('v',wc,s,2);
48 d2 = detcoef2('d',wc,s,2);
49
50 // Extract the level 2 coefficients. Fourth argument
    must be 3 for extraction the level 2
    coefficients in appcoef2() and detcoef2()
    function.
51
52 a3 = appcoef2(wc,s,wname,3);
53 h3 = detcoef2('h',wc,s,3);
54 v3 = detcoef2('v',wc,s,3);
55 d3 = detcoef2('d',wc,s,3);
56
57
58 // Extract the level 2 coefficients. Fourth argument
    must be 4 for extraction the level 2
    coefficients in appcoef2() and detcoef2()
    function.
59
60 a4 = appcoef2(wc,s,wname,4);
61 h4 = detcoef2('h',wc,s,4);
62 v4 = detcoef2('v',wc,s,4);
63 d4 = detcoef2('d',wc,s,4);
64
65
66
67
68 // Display the decomposition up to level 1 only.
69 a1=double(a1);
70 cod_a1 = wcodemat(a1,256);
71 cod_a1=double(cod_a1);
72 cod_a1 = wkeep(cod_a1,[344 344]);
73 cod_h1 = wcodemat(h1,260);
74 cod_h1=double(cod_h1);

```

```

75 cod_h1 = wkeep(cod_h1, [344 344]);
76 cod_v1 = wcodemat(v1,260);
77 cod_v1=double(cod_v1);
78 cod_v1 = wkeep(cod_v1, [344 344]);
79 cod_d1 = wcodemat(d1,260);
80 cod_d1=double(cod_d1);
81 cod_d1 = wkeep(cod_d1, [344 344]);
82 ans1=[cod_a1 ,cod_h1;cod_v1 ,cod_d1];
83
84 figure,ShowImage(ans1,' Wavelet Transform of an
    image');//ShowImage() is used to show gray scale
    image, figure is command to view images in
    separate window.
85 title(' Wavelet Transform of an image');//title() is
    used for providing a title to an image.
86
87
88
89
90 // Display the entire decomposition up to level 2.
91 cod_a2 = wcodemat(a2,260);
92 cod_a2=double(cod_a2);
93 cod_a2 = wkeep(cod_a2, [172 172]);
94 cod_h2 = wcodemat(h2,260);
95 cod_h2=double(cod_h2);
96 cod_h2 = wkeep(cod_h2, [172 172]);
97 cod_v2 = wcodemat(v2,260);
98 cod_v2=double(cod_v2);
99 cod_v2 = wkeep(cod_v2, [172 172]);
100 cod_d2 = wcodemat(d2,260);
101 cod_d2=double(cod_d2);
102 cod_d2 = wkeep(cod_d2, [172 172]);
103 bb=[[cod_a2 ,cod_h2;cod_v2 ,cod_d2],cod_h1;cod_v1 ,
    cod_d1];
104
105
106 figure,ShowImage(bb,' Second level Wavelet transform
    an image');//ShowImage() is used to show gray

```

```

    scale image, figure is command to view images in
    separate window.
107 title(' Second level Wavelet transform an image');//
    title() is used for providing a title to an
    image.
108
109
110 // Display the entire decomposition upto level 3.
111 cod_a3 = wcodemat(a3,260);
112 cod_a3=double(cod_a3);
113 cod_a3 = wkeep(cod_a3,[86 86]);
114 cod_h3 = wcodemat(h3,260);
115 cod_h3=double(cod_h3);
116 cod_h3 = wkeep(cod_h3,[86 86]);
117 cod_v3 = wcodemat(v3,260);
118 cod_v3=double(cod_v3);
119 cod_v3 = wkeep(cod_v3,[86 86]);
120 cod_d3 = wcodemat(d3,260);
121 cod_d3=double(cod_d3);
122 cod_d3 = wkeep(cod_d3,[86 86]);
123 bbb=[[cod_a3, cod_h3;cod_v3, cod_d3], cod_h2;cod_v2,
    cod_d2],cod_h1;cod_v1, cod_d1];
124
125
126 figure,ShowImage(bbb,' Third level Wavelet transform
    an image');//ShowImage() is used to show gray
    scale image, figure is command to view images in
    separate window.
127 title(' Third level Wavelet transform an image');//
    title() is used for providing a title to an
    image.
128
129
130
131 // Display the entire decomposition upto level 2.
132 cod_a4 = wcodemat(a4,260);
133 cod_a4=double(cod_a4);
134 cod_a4 = wkeep(cod_a4, [43 43]);

```

```

135 cod_h4 = wcodemat(h4,260);
136 cod_h4=double(cod_h4);
137 cod_h4 = wkeep(cod_h4, [43 43]);
138 cod_v4 = wcodemat(v4,260);
139 cod_v4=double(cod_v4);
140 cod_v4 = wkeep(cod_v4, [43 43]);
141 cod_d4 = wcodemat(d4,260);
142 cod_d4=double(cod_d4);
143 cod_d4 = wkeep(cod_d4, [43 43]);
144 bbbb=[[ [ [ [cod_a4, cod_h4; cod_v4, cod_d4], cod_h3; cod_v3,
          cod_d3], cod_h2; cod_v2, cod_d2], cod_h1; cod_v1,
          cod_d1]];
145
146
147 figure, ShowImage(bbbb, ' Fourth level Wavelet
    transform an image'); // ShowImage() is used to
    show gray scale image, figure is command to view
    images in separate window.
148 title('Fourth level Wavelet transform an image'); //
    title() is used for providing a title to an
    image.
149
150 // Reconstruction of an image using wrcoef2()
    function. Last argument is used for level of
    reconstruction.
151
152 ra4 = wrcoef2('a',wc,s,wname,4);
153 rh4 = wrcoef2('h',wc,s,wname,4);
154 rv4 = wrcoef2('v',wc,s,wname,4);
155 rd4 = wrcoef2('d',wc,s,wname,4);
156
157 ra3 = wrcoef2('a',wc,s,wname,3);
158 rh3 = wrcoef2('h',wc,s,wname,3);
159 rv3 = wrcoef2('v',wc,s,wname,3);
160 rd3 = wrcoef2('d',wc,s,wname,3);
161
162 ra2 = wrcoef2('a',wc,s,wname,2); // 'a' is used for
    approximation coefficients.

```

```

163 rh2 = wrcoef2('h',wc,s,wname,2); // 'h' is used for
    horizontal detail coefficients.
164 rv2 = wrcoef2('v',wc,s,wname,2); // 'v' is used for
    vertical detail coefficients.
165 rd2 = wrcoef2('d',wc,s,wname,2); // 'd' is used for
    detail coefficients.
166
167 ra1 = wrcoef2('a',wc,s,wname,1);
168 rh1 = wrcoef2('h',wc,s,wname,1);
169 rv1 = wrcoef2('v',wc,s,wname,1);
170 rd1 = wrcoef2('d',wc,s,wname,1);
171
172
173 cod_ra4 = wcodemat(ra4,260);
174 cod_rh4 = wcodemat(rh4,260);
175 cod_rv4 = wcodemat(rv4,260);
176 cod_rd4 = wcodemat(rd4,260);
177
178 cod_ra3 = wcodemat(ra3,260);
179 cod_rh3 = wcodemat(rh3,260);
180 cod_rv3 = wcodemat(rv3,260);
181 cod_rd3 = wcodemat(rd3,260);
182
183 cod_ra2 = wcodemat(ra2,260);
184 cod_rh2 = wcodemat(rh2,260);
185 cod_rv2 = wcodemat(rv2,260);
186 cod_rd2 = wcodemat(rd2,260);
187
188 cod_ra1 = wcodemat(ra1,260);
189 cod_rh1 = wcodemat(rh1,260);
190 cod_rv1 = wcodemat(rv1,260);
191 cod_rd1 = wcodemat(rd1,260);
192
193 //zeroing first level detail coefficients.
194 rh1(1:$,1:$)=0;
195 rd1(1:$,1:$)=0;
196 rv1(1:$,1:$)=0;
197

```

```

198 Xhat = ra4+rh4 + rv4 + rd4+rh3 + rv3 + rd3 + rh2 +
      rv2 + rd2 + rh1 + rv1 + rd1;
199 figure,ShowImage(Xhat,' Reconstruction of an image
      after zeroing first level detail coefficients. ');
      //ShowImage() is used to show gray scale image,
      figure is command to view images in separate
      window.
200 title(' Reconstruction of an image after zeroing
      first and second level detail coefficients. ');//
      title() is used for providing a title to an
      image.
201
202 // zeroing first and second level detail
      coefficients.
203 rh2(1:$,1:$)=0;
204 rv2(1:$,1:$)=0;
205 rd2(1:$,1:$)=0;
206
207 rh1(1:$,1:$)=0;
208 rd1(1:$,1:$)=0;
209 rv1(1:$,1:$)=0;
210 Xhat = ra4+rh4 + rv4 + rd4+rh3 + rv3 + rd3 + rh2 +
      rv2 + rd2 + rh1 + rv1 + rd1;
211 figure,ShowImage(Xhat,'reconstion of an image after
      zeroing first and second level detail
      coefficients. ');//ShowImage() is used to show
      gray scale image, figure is command to view
      images in separate window.
212 title('Reconstruction of an image after zeroing
      first and second level detail coefficients. ');//
      title() is used for providing a title to an
      image.
213
214 //zeroing first , second and third level detail
      coefficients.
215 rh3(1:$,1:$)=0;
216 rv3(1:$,1:$)=0;
217 rd3(1:$,1:$)=0;

```

```

218
219 rh2(1:$,1:$)=0;
220 rv2(1:$,1:$)=0;
221 rd2(1:$,1:$)=0;
222
223 rh1(1:$,1:$)=0;
224 rd1(1:$,1:$)=0;
225 rv1(1:$,1:$)=0;
226 Xhat = ra4+rh4 + rv4 + rd4+rh3 + rv3 + rd3 + rh2 +
      rv2 + rd2 + rh1 + rv1 + rd1;
227 figure,ShowImage(Xhat,' Reconstruction of an image
      after zeroing first , second and third level
      detail coefficients.');//ShowImage() is used to
      show gray scale image, figure is command to view
      images in separate window.
228 title('Reconstruction of an image after zeroing
      first , second and third level detail coefficients
      .');//title() is used for providing a title to
      an image.

229
230
231 //zeroing all level detail coefficients.
232 rh4(1:$,1:$)=0;
233 rv4(1:$,1:$)=0;
234 rd4(1:$,1:$)=0;
235
236
237 rh3(1:$,1:$)=0;
238 rv3(1:$,1:$)=0;
239 rd3(1:$,1:$)=0;
240
241
242 rh2(1:$,1:$)=0;
243 rv2(1:$,1:$)=0;
244 rd2(1:$,1:$)=0;
245
246 rh1(1:$,1:$)=0;
247 rd1(1:$,1:$)=0;

```

```
248 rv1(1:$,1:$)=0;
249 Xhat = ra4+rh4 + rv4 + rd4+rh3 + rv3 + rd3 + rh2 +
      rv2 + rd2 + rh1 + rv1 + rd1;
250 figure,ShowImage(Xhat,' Reconstruction of an image
      after zeroing all level detail coefficients.');//
      ShowImage() is used to show gray scale image,
      figure is command to view images in separate
      window.
251 title(' Reconstruction of an image after zeroing all
      level detail coefficients.');//title() is used
      for providing a title to an image.
```

Appendix