

Scilab Manual for
Image Processing and Machine Vision
by Dr Ravindra Chandu Patil
Electronics and Telecommunication
Engineering
M. H. Saboo Siddik College Of Engineering
Mumbai University¹

Solutions provided by
Dr Ravindra Chandu Patil
Electronics and Telecommunication Engineering
M. H. Saboo Siddik College Of Engineering Mumbai University

June 19, 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	4
1 Write a program to perform arithmetic operations on an image .	18
2 Write a program to perform sampling on a given image.	52
3 Write a program to perform quantization on a given image.	59
4 Write a program to perform point processing for image enhancement on a given image.	73
5 Write a program to perform image enhancement using neighborhood processing on a given image.	152
6 Write a program to perform Histogram and Histogram equalization on a given image.	176
7 Write a program to perform image segmentation using edge detection on a given image.	196
8 Write a program to perform image segmentation using local and global thresholding on a given image.	231
9 Write a program to perform Block Truncation coding on a given image.	245
10 Write a program to implement morphological dilation and erosion operation on an image .	251

- 11 Write a program to implement morphological closing and opening operation on a given image. 262
- 12 Write a program to implement boundary extraction using morphological operation on an image. 272
- 13 Write a program to perform Truncated DCT and IDCT on a given image. 280
- 14 Write a program to perform image Wavelet transform on given image. 290

List of Experiments

Solution 1.1	Increasing and decreasing the contrast of color images and grayscale images	18
Solution 1.2	Separation of color planes and display of color image	30
Solution 1.3	Flipping operation on color planes	31
Solution 1.4	Addition and subtraction operation on color images and grayscale images	40
Solution 1.5	Formation of grayscale image from color image using multiplication and division operation	48
Solution 2.0	Sampling of grey scale image	52
Solution 3.1	Quantization of grey scale image using 1 2 3 4 bits per pixels	59
Solution 3.2	Quantization of grey scale image using 5 6 7 and 8 bits per pixels	66
Solution 4.1	To perform a Negation operation on grayscale image and color image	73
Solution 4.2	To implement Thresholding of image using mean as a threshold value and half of the maximum pixel value of image	86
Solution 4.3	To Implement gray level slicing with background and without background	98
Solution 4.4	To implement Bit Plane slicing of gray scale image	110
Solution 4.5	To Perform contrast Stretching using lower mean color value and upper mean color value as a lower threshold and upper threshold value respectively on given image	121
Solution 4.6	Perform Power transform on given image	132
Solution 4.7	To Perform Log transform on given image	137

Solution 5.1	Image Enhancement using neighbourhood processing Average filtering	152
Solution 5.2	Image Enhancement using neighbourhood processing Median filtering	158
Solution 5.3	Image Enhancement using neighbourhood processing high boost filter	159
Solution 5.4	Image Enhancement using neighbourhood processing high pass filter	171
Solution 6.1	To implement Histogram and histogram equalization of grayscale image	176
Solution 6.2	To Implement Histogram and histogram equalization of color image	183
Solution 7.1	To Implement Image segmentation based on discontinuities using edge detection Ordinary operator	196
Solution 7.2	To Implement Image segmentation discontinuities using edge detection Robert Operator	203
Solution 7.3	To Implement Image segmentation discontinuities using edge detection prewitt operator	210
Solution 7.4	To Implement Image segmentation discontinuities using edge detection Sobel operator	216
Solution 7.5	To Implement Image segmentation discontinuities using edge detection Laplacian operator	222
Solution 7.6	To Implement Image segmentation based on discontinuities using edge detection LOG operator	226
Solution 8.1	To implement image segmentation based on similarities using Global thresholding	231
Solution 8.2	To implement image segmentation based on similarities using local thresholding	238
Solution 9.0	To implement Block truncation coding on grayscale image and color image	245
Solution 10.1	To implement Dilation of binary image and grayscale image edges	251
Solution 10.2	To implement Erosion of binary image	253
Solution 11.1	To perform morphological opening operation on binary image	262
Solution 11.2	To perform morphological Closing operation on binary image	267

Solution 12.0	To implement Boundary Extraction using morphological operation of binary image	272
Solution 13.0	To Perform Truncated DCT and IDCT on a given Image	280
Solution 14.0	To Perform Wavelet Transform on grayscale image	290
AP 2	TRUNCTION DCT IDCT	301
AP 3	erosion	302
AP 4	dilation function	302
AP 5	upper and lower mean color values	304
AP 6	colorplanes	304
AP 7	Image thersholding	306
AP 8	Sum of product	306
AP 9	masking 2x2	307
AP 11	histogram function	308
AP 12	masking function	308
AP 13	log function	309
AP 14	gamma function	309
AP 15	upper lower color function	311
AP 17	graylevel slicing function	312
AP 18	Function for Thesholding	313
AP 19	Function for negation	313
AP 20	Quantization function	313
AP 21	Sampling function	314

List of Figures

1.1	Increasing and decreasing the construct of color images and grayscale images	21
1.2	Increasing and decreasing the construct of color images and grayscale images	22
1.3	Increasing and decreasing the construct of color images and grayscale images	23
1.4	Increasing and decreasing the construct of color images and grayscale images	24
1.5	Increasing and decreasing the construct of color images and grayscale images	25
1.6	Increasing and decreasing the construct of color images and grayscale images	26
1.7	Increasing and decreasing the construct of color images and grayscale images	27
1.8	Increasing and decreasing the construct of color images and grayscale images	28
1.9	Increasing and decreasing the construct of color images and grayscale images	29
1.10	Sepration of colorplanes and display of color image	32
1.11	Sepration of colorplanes and display of color image	33
1.12	Sepration of colorplanes and display of color image	34
1.13	Sepration of colorplanes and display of color image	35
1.14	Fliping operation on color planes	37
1.15	Fliping operation on color planes	38
1.16	Fliping operation on color planes	39
1.17	Addition and substraction operation on color images and grayscale images	42

1.18	Addition and subtraction operation on color images and grayscale images	43
1.19	Addition and subtraction operation on color images and grayscale images	44
1.20	Addition and subtraction operation on color images and grayscale images	45
1.21	Addition and subtraction operation on color images and grayscale images	46
1.22	Addition and subtraction operation on color images and grayscale images	47
1.23	Formation of grayscale image from color image using multiplication and division operation	49
1.24	Formation of grayscale image from color image using multiplication and division operation	50
1.25	Formation of grayscale image from color image using multiplication and division operation	51
2.1	Sampling of grey scale image	54
2.2	Sampling of grey scale image	55
2.3	Sampling of grey scale image	56
2.4	Sampling of grey scale image	57
2.5	Sampling of grey scale image	58
3.1	Quantization of grey scale image using 1 2 3 4 bits per pixels	61
3.2	Quantization of grey scale image using 1 2 3 4 bits per pixels	62
3.3	Quantization of grey scale image using 1 2 3 4 bits per pixels	63
3.4	Quantization of grey scale image using 1 2 3 4 bits per pixels	64
3.5	Quantization of grey scale image using 1 2 3 4 bits per pixels	65
3.6	Quantization of grey scale image using 5 6 7 and 8 bits per pixels	68
3.7	Quantization of grey scale image using 5 6 7 and 8 bits per pixels	69
3.8	Quantization of grey scale image using 5 6 7 and 8 bits per pixels	70
3.9	Quantization of grey scale image using 5 6 7 and 8 bits per pixels	71
3.10	Quantization of grey scale image using 5 6 7 and 8 bits per pixels	72

4.1	To perform a Negatiation operation on grayscale image and color image	76
4.2	To perform a Negatiation operation on grayscale image and color image	77
4.3	To perform a Negatiation operation on grayscale image and color image	78
4.4	To perform a Negatiation operation on grayscale image and color image	79
4.5	To perform a Negatiation operation on grayscale image and color image	80
4.6	To perform a Negatiation operation on grayscale image and color image	81
4.7	To perform a Negatiation operation on grayscale image and color image	82
4.8	To perform a Negatiation operation on grayscale image and color image	83
4.9	To perform a Negatiation operation on grayscale image and color image	84
4.10	To perform a Negatiation operation on grayscale image and color image	85
4.11	To implement Thresholding of image using mean as a thshold value and half of the maximum pixel value of image	88
4.12	To implement Thresholding of image using mean as a thshold value and half of the maximum pixel value of image	89
4.13	To implement Thresholding of image using mean as a thshold value and half of the maximum pixel value of image	90
4.14	To implement Thresholding of image using mean as a thshold value and half of the maximum pixel value of image	91
4.15	To implement Thresholding of image using mean as a thshold value and half of the maximum pixel value of image	92
4.16	To implement Thresholding of image using mean as a thshold value and half of the maximum pixel value of image	93
4.17	To implement Thresholding of image using mean as a thshold value and half of the maximum pixel value of image	94
4.18	To implement Thresholding of image using mean as a thshold value and half of the maximum pixel value of image	95
4.19	To implement Thresholding of image using mean as a thshold value and half of the maximum pixel value of image	96

4.20	To implement Thresholding of image using mean as a thshold value and half of the maximum pixel value of image	97
4.21	To Implement gray level slicing with background and without background	100
4.22	To Implement gray level slicing with background and without background	101
4.23	To Implement gray level slicing with background and without background	102
4.24	To Implement gray level slicing with background and without background	103
4.25	To Implement gray level slicing with background and without background	104
4.26	To Implement gray level slicing with background and without background	105
4.27	To Implement gray level slicing with background and without background	106
4.28	To Implement gray level slicing with background and without background	107
4.29	To Implement gray level slicing with background and without background	108
4.30	To Implement gray level slicing with background and without background	109
4.31	To implement Bit Plane slicing of gray scale image	112
4.32	To implement Bit Plane slicing of gray scale image	113
4.33	To implement Bit Plane slicing of gray scale image	114
4.34	To implement Bit Plane slicing of gray scale image	115
4.35	To implement Bit Plane slicing of gray scale image	116
4.36	To implement Bit Plane slicing of gray scale image	117
4.37	To implement Bit Plane slicing of gray scale image	118
4.38	To implement Bit Plane slicing of gray scale image	119
4.39	To implement Bit Plane slicing of gray scale image	120
4.40	To Perform constract Sretching using lower mean color value and upper mean color value as a lower thershold and upper thershold value respectively on given image	123
4.41	To Perform constract Sretching using lower mean color value and upper mean color value as a lower thershold and upper thershold value respectively on given image	124

4.42	To Perform contract Sretching using lower mean color value and upper mean color value as a lower thershold and upper thershold value respectively on given image	125
4.43	To Perform contract Sretching using lower mean color value and upper mean color value as a lower thershold and upper thershold value respectively on given image	126
4.44	To Perform contract Sretching using lower mean color value and upper mean color value as a lower thershold and upper thershold value respectively on given image	127
4.45	To Perform contract Sretching using lower mean color value and upper mean color value as a lower thershold and upper thershold value respectively on given image	128
4.46	To Perform contract Sretching using lower mean color value and upper mean color value as a lower thershold and upper thershold value respectively on given image	129
4.47	To Perform contract Sretching using lower mean color value and upper mean color value as a lower thershold and upper thershold value respectively on given image	130
4.48	To Perform contract Sretching using lower mean color value and upper mean color value as a lower thershold and upper thershold value respectively on given image	131
4.49	Perform Power transform on given image	134
4.50	Perform Power transform on given image	135
4.51	Perform Power transform on given image	136
4.52	Perform Power transform on given image	138
4.53	Perform Power transform on given image	139
4.54	Perform Power transform on given image	140
4.55	Perform Power transform on given image	141
4.56	Perform Power transform on given image	142
4.57	Perform Power transform on given image	143
4.58	To Perform Log transform on given image	145
4.59	To Perform Log transform on given image	146
4.60	To Perform Log transform on given image	147
4.61	To Perform Log transform on given image	148
4.62	To Perform Log transform on given image	149
4.63	To Perform Log transform on given image	150

5.1	Image Enhancement using neighbourhood processing Average filtering	154
5.2	Image Enhancement using neighbourhood processing Average filtering	155
5.3	Image Enhancement using neighbourhood processing Average filtering	156
5.4	Image Enhancement using neighbourhood processing Average filtering	157
5.5	Image Enhancement using neighbourhood processing Median filtering	160
5.6	Image Enhancement using neighbourhood processing Median filtering	161
5.7	Image Enhancement using neighbourhood processing Median filtering	162
5.8	Image Enhancement using neighbourhood processing Median filtering	163
5.9	Image Enhancement using neighbourhood processing high boost filter	165
5.10	Image Enhancement using neighbourhood processing high boost filter	166
5.11	Image Enhancement using neighbourhood processing high boost filter	167
5.12	Image Enhancement using neighbourhood processing high boost filter	168
5.13	Image Enhancement using neighbourhood processing high boost filter	169
5.14	Image Enhancement using neighbourhood processing high boost filter	170
5.15	Image Enhancement using neighbourhood processing high pass filter	172
5.16	Image Enhancement using neighbourhood processing high pass filter	173
5.17	Image Enhancement using neighbourhood processing high pass filter	174
5.18	Image Enhancement using neighbourhood processing high pass filter	175

6.1	To implement Histogram and histogram equalization of grayscale image	178
6.2	To implement Histogram and histogram equalization of grayscale image	179
6.3	To implement Histogram and histogram equalization of grayscale image	180
6.4	To implement Histogram and histogram equalization of grayscale image	181
6.5	To implement Histogram and histogram equalization of grayscale image	182
6.6	To Implement Histogram and histogram equalization of color image	186
6.7	To Implement Histogram and histogram equalization of color image	187
6.8	To Implement Histogram and histogram equalization of color image	188
6.9	To Implement Histogram and histogram equalization of color image	189
6.10	To Implement Histogram and histogram equalization of color image	190
6.11	To Implement Histogram and histogram equalization of color image	191
6.12	To Implement Histogram and histogram equalization of color image	192
6.13	To Implement Histogram and histogram equalization of color image	193
6.14	To Implement Histogram and histogram equalization of color image	194
6.15	To Implement Histogram and histogram equalization of color image	195
7.1	To Implement Image segmentation based on discontinuities using edge detection Ordinary operator	198
7.2	To Implement Image segmentation based on discontinuities using edge detection Ordinary operator	199
7.3	To Implement Image segmentation based on discontinuities using edge detection Ordinary operator	200

7.4	To Implement Image segmentation based on discontinuities using edge detection Ordinary operator	201
7.5	To Implement Image segmentation based on discontinuities using edge detection Ordinary operator	202
7.6	To ImplementImage segmentation discontinuities using edge detection Robert Operator	205
7.7	To ImplementImage segmentation discontinuities using edge detection Robert Operator	206
7.8	To ImplementImage segmentation discontinuities using edge detection Robert Operator	207
7.9	To ImplementImage segmentation discontinuities using edge detection Robert Operator	208
7.10	To ImplementImage segmentation discontinuities using edge detection Robert Operator	209
7.11	To Implement Image segmentation discontinuities using edge detection prewitt operator	211
7.12	To Implement Image segmentation discontinuities using edge detection prewitt operator	212
7.13	To Implement Image segmentation discontinuities using edge detection prewitt operator	213
7.14	To Implement Image segmentation discontinuities using edge detection prewitt operator	214
7.15	To Implement Image segmentation discontinuities using edge detection prewitt operator	215
7.16	To Implement Image segmentation discontinuities using edge detection Sobel operator	217
7.17	To Implement Image segmentation discontinuities using edge detection Sobel operator	218
7.18	To Implement Image segmentation discontinuities using edge detection Sobel operator	219
7.19	To Implement Image segmentation discontinuities using edge detection Sobel operator	220
7.20	To Implement Image segmentation discontinuities using edge detection Sobel operator	221
7.21	To Implement Image segmentation discontinuities using edge detection Laplacian operator	223
7.22	To Implement Image segmentation discontinuities using edge detection Laplacian operator	224

7.23	To Implement Image segmentation discontinuities using edge detection Laplacian operator	225
7.24	To Implement Image segmentation based on discontinuities using edge detection LOG operator	228
7.25	To Implement Image segmentation based on discontinuities using edge detection LOG operator	229
7.26	To Implement Image segmentation based on discontinuities using edge detection LOG operator	230
8.1	To implement image segmentation based on simalarities using Global thresholding	233
8.2	To implement image segmentation based on simalarities using Global thresholding	234
8.3	To implement image segmentation based on simalarities using Global thresholding	235
8.4	To implement image segmentation based on simalarities using Global thresholding	236
8.5	To implement image segmentation based on simalarities using Global thresholding	237
8.6	To implement image segmentation based on simalarities using local thresholding	240
8.7	To implement image segmentation based on simalarities using local thresholding	241
8.8	To implement image segmentation based on simalarities using local thresholding	242
8.9	To implement image segmentation based on simalarities using local thresholding	243
9.1	To implement Block truncation coding on grayscale image and color image	247
9.2	To implement Block truncation coding on grayscale image and color image	248
9.3	To implement Block truncation coding on grayscale image and color image	249
10.1	To implement Dilation of binary image and grayscale image edges	254

10.2	To implement Dilation of binary image and grayscale image edges	255
10.3	To implement Dilation of binary image and grayscale image edges	256
10.4	To implement Dilation of binary image and grayscale image edges	257
10.5	To implement Dilation of binary image and grayscale image edges	258
10.6	To implement Erosion of binary image	260
10.7	To implement Erosion of binary image	261
11.1	To perform morphological opening operation on binary image	264
11.2	To perform morphological opening operation on binary image	265
11.3	To perform morphological opening operation on binary image	266
11.4	To perform morphological Closing operation on binary image	269
11.5	To perform morphological Closing operation on binary image	270
11.6	To perform morphological Closing operation on binary image	271
12.1	To implement Boundary Extraction using morphological operation of binary image	275
12.2	To implement Boundary Extraction using morphological operation of binary image	276
12.3	To implement Boundary Extraction using morphological operation of binary image	277
12.4	To implement Boundary Extraction using morphological operation of binary image	278
12.5	To implement Boundary Extraction using morphological operation of binary image	279
13.1	To Perform Truncated DCT and IDCT on a given Image	282
13.2	To Perform Truncated DCT and IDCT on a given Image	283
13.3	To Perform Truncated DCT and IDCT on a given Image	284
13.4	To Perform Truncated DCT and IDCT on a given Image	285
13.5	To Perform Truncated DCT and IDCT on a given Image	286
13.6	To Perform Truncated DCT and IDCT on a given Image	287
13.7	To Perform Truncated DCT and IDCT on a given Image	288
13.8	To Perform Truncated DCT and IDCT on a given Image	289
14.1	To Perform Wavelet Transform on grayscale image	292

14.2 To Perform Wavelet Transform on grayscale image	293
14.3 To Perform Wavelet Transform on grayscale image	294
14.4 To Perform Wavelet Transform on grayscale image	295
14.5 To Perform Wavelet Transform on grayscale image	296
14.6 To Perform Wavelet Transform on grayscale image	297
14.7 To Perform Wavelet Transform on grayscale image	298
14.8 To Perform Wavelet Transform on grayscale image	299
14.9 To Perform Wavelet Transform on grayscale image	300

Experiment: 1

Write a program to perform arithmetic operations on an image .

Scilab code Solution 1.1 Increasing and decreasing the contrast of color images and grayscale images

```
1 //Ex 1.1 Program Title:Increasing and decreasing the
    construct of color images and grayscale images.
2 //environment: Scilab 5.2.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
    book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
    Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
    plots
```

```

13 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed.getSIVPpath()
    is pre-defined function in SIVP toolbox.
14 ii =imread(SIVP_PATH + 'images/rose.bmp');//read
    first color image from diractory
15 ii2=imread(SIVP_PATH + 'images/rose2.bmp');//read
    second color image from diractory
16 ii3=rgb2gray(ii);
17 ii4=rgb2gray(ii2);
18 f1=ii+50;//addition of constant in color images
19 f2=ii-50;//substraction of constant from color
    images
20 f3=ii2+50;//addition of constant in color images
21 f4=ii2-50;//substraction of constant from color
    images
22 f5=ii3+30;//Addition of constant in grayscale images
23 f6=ii3-10;//Substraction of constant from grayscale
    images
24
25 figure(1);ShowColorImage(ii,' color image1');title('
    color image1','color','red','fontsize',3);//color
    images
26 figure(2);ShowColorImage(ii2,'color image2');title('
    color image2','color','red','fontsize',4);//
    color images
27 figure(3);ShowColorImage(f1,'Addition of constant
    color image');title('Addition of constant in
    color image','color','red','fontsize',4);//show
    addition of constant in color images
28 figure(4);ShowColorImage(f2,'Substraction of
    constant color image');title('Substraction of
    constant color image','color','red','fontsize'
    ,3);//show substraction of constant from color
    images
29 figure(5);ShowColorImage(f3,'Addition of constant
    color image');title('Addition of constant from
    color image','color','red','fontsize',4);//show
    addition of constant in color images

```

```
30 figure(6); ShowColorImage(f4, 'Substraction of
    constant color image'); title('Substraction of
    constant in color image', 'color', 'red', 'fontsize',
    ,4); //show substraction of constant in color
    images
31 figure(7); ShowImage(ii3, 'Grayscale image'); title('
    Grayscale Image', 'color', 'red', 'fontsize', 4); //
    Grayscale image
32 figure(8); ShowImage(f5, 'Addition of constant in
    grayscale image'); title('Addition of constant in
    grayscale image for increasing constract', 'color',
    , 'red', 'fontsize', 4); //show addition of constant
    in grayscale image
33 figure(9); ShowImage(f6, 'Substraction of constant
    from grayscale image'); title('Substraction of
    constant from grayscale image', 'color', 'red', '
    fontsize', 4); //show substraction of constant from
    grayscale image
```

check Appendix ?? for dependency:

color image1



Figure 1.1: Increasing and decreasing the contrast of color images and grayscale images

color image2



Figure 1.2: Increasing and decreasing the contrast of color images and grayscale images

Addition of constant in color image



Figure 1.3: Increasing and decreasing the contrast of color images and grayscale images

Substraction of constant color image

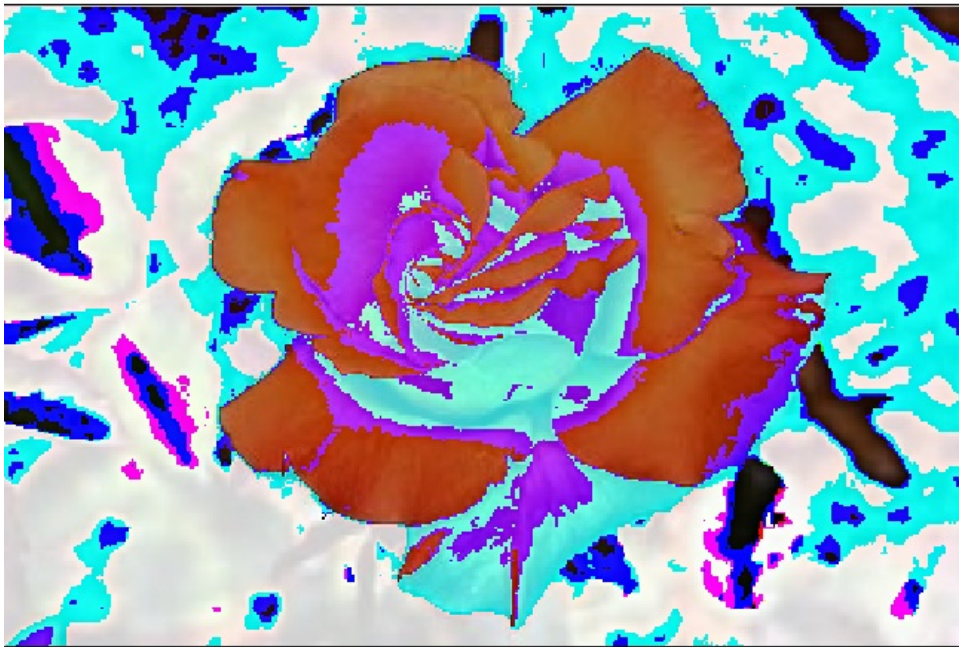


Figure 1.4: Increasing and decreasing the contrast of color images and grayscale images

Addition of constant from color image



Figure 1.5: Increasing and decreasing the contrast of color images and grayscale images

Substraction of constant in color image

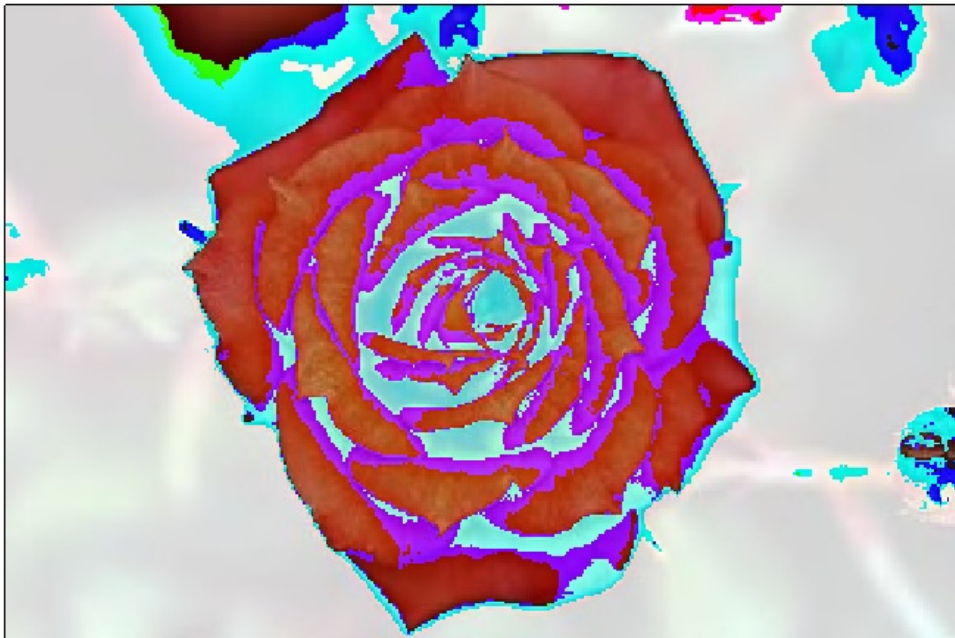


Figure 1.6: Increasing and decreasing the contrast of color images and grayscale images

Grayscale Image



Figure 1.7: Increasing and decreasing the contrast of color images and grayscale images

Addition of constant in grayscale image for increasing contrast



Figure 1.8: Increasing and decreasing the contrast of color images and grayscale images

Substraction of constant from grayscale image



Figure 1.9: Increasing and decreasing the contrast of color images and grayscale images

rose.BMP

check Appendix ?? for dependency:

rose2.BMP

Scilab code Solution 1.2 Separation of colorplanes and display of color image

```
1 //Exp 1.2 Program Title:Separation of colorplanes and
  display of color image
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference book name : 1) Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
  plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
  in which SIVP toolbox is installed.getSIVPpath()
  is pre-defined function in SIVP toolbox.
14 ii =imread(SIVP_PATH + 'images/rose.bmp');///read
  color image from diractory
15 r=ii;//copy color image in other variable to show
  red plane
16 g=ii;//copy color image in other variable to show
  green plane
17 b=ii;//copy color image in other variable to show
  blue plane
```

```

18 r(:,:,2)=0; //make green plane Zero
19 r(:,:,3)=0; //make Blue Plane Zero
20 g(:,:,1)=0; // Make RED PLANE Zero
21 g(:,:,3)=0; //Make Blue Plane Zero
22 b(:,:,1)=0; // Make RED PLANE Zero
23 b(:,:,2)=0; //Make green Plane Zero
24 figure(1); ShowColorImage(ii, 'Color image1'); title('
    Color image', 'color', 'red', 'fontsize', 5); //show
    original color image
25 figure(2); ShowColorImage(r, 'red color plane'); title(
    'Red color plane', 'color', 'red', 'fontsize', 5); //
    show Red plane
26 figure(3); ShowColorImage(g, 'green color plane');
    title('Green color plane', 'color', 'red', 'fontsize
    ', 5); //show Green plane
27 figure(4); ShowColorImage(b, 'blue color plane'); title
    ('Blue color plane', 'color', 'red', 'fontsize', 5);
    //show Blue plane

```

check Appendix ?? for dependency:

rose.BMP

Scilab code Solution 1.3 Flipping operation on color planes

```

1 //Exp 1.3 Program Title: Flipping operation on color
  planes
2 //environment: Scilab 5.5.2

```

Color image 1



Figure 1.10: Separation of colorplanes and display of color image

Red color plane



Figure 1.11: Separation of colorplanes and display of color image

Green color plane



Figure 1.12: Separation of colorplanes and display of color image

Blue color plane

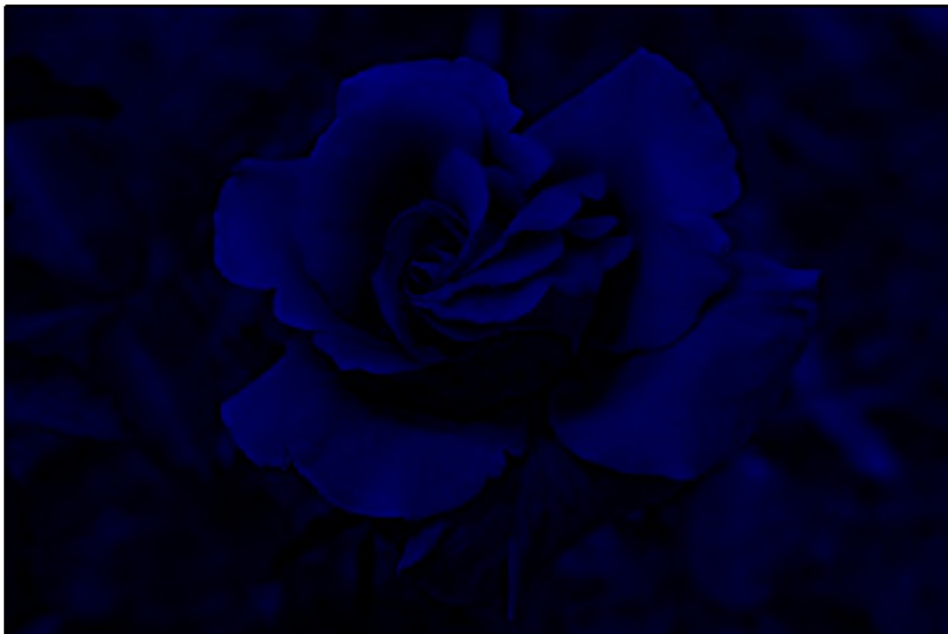


Figure 1.13: Separation of colorplanes and display of color image

```

3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional, 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir,
9 //3)Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed.getSIVPpath()
   is pre-defined function in SIVP toolbox.
14 ii =imread(SIVP_PATH + 'images/rose.bmp');///read
   color image from diractory
15 [Ri1,Gi2,Bi3]=colorplanes(ii);///call function for
   seperation of RGB color planes
16 f1=flipdim(Ri1,1);///flip red plane vertically
17 f2=flipdim(Bi3,2);///flip Blue plane Horizontally
18 figure(1);ShowColorImage(ii,'Color image1');title('
   Color image 1','color','red','fontsize',5);///show
   original color image
19 figure(2);ShowImage(f1,'Fliping vertically');title('
   Fliping vertically','color','red','fontsize',5);///
   show fliped red plane image
20 figure(3);ShowImage(f2,'Fliping Horizontal');title('
   Fliping Horizontal','color','red','fontsize',5);
   //show fliped blue plane image

```

color image1



Figure 1.14: Flipping operation on color planes

Fliping vertically



Figure 1.15: Fliping operation on color planes

Fliping Horizontal



Figure 1.16: Fliping operation on color planes

check Appendix [AP 6](#) for dependency:

colorplanes.sci

check Appendix ?? for dependency:

rose.BMP

Scilab code Solution 1.4 Addition and subtraction operation on color images and grayscale images

```
1 //Exp 1.4 Program Title:Addition and subtraction
  operation on color images and grayscale images//
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1) Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
  plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
  in which SIVP toolbox is installed.getSIVPpath()
  is pre-defined function in SIVP toolbox.
14 ii =imread(SIVP_PATH + 'images/rose.bmp');//read
  first color image from diractory
15 ii1=imread(SIVP_PATH + 'images/rose1.bmp');//read
  second color image from diractory
16 ii2=imread(SIVP_PATH + 'images/rose2.bmp');//read
  third color image from diractory
```

```

17 [Ri1,Gi2,Bi3]=colorplanes(ii);//call function for
    seperation of RGB color planes
18 f1=flipdim(Ri1,1);//flip red plane vertically
19 f2=flipdim(Bi3,2);//flip Blue plane Horizontally
20 f3=f1+f2;//Addition of flipped red plane and blue
    plane image
21 f4=ii+ii1;//addition of color images
22 f5=ii-ii2;//substraction of color images
23 figure(1);ShowColorImage(ii,'Color image1');title('
    Color image 1','color','red','fontsize',5);//show
    color image
24 figure(2);ShowColorImage(ii1,'original image2');
    title('Color image 2','color','red','fontsize',5)
    ;//show color image
25 figure(3);ShowColorImage(ii2,'original image3');
    title('Color image 3','color','red','fontsize',5)
    ;//show color image
26 figure(4);ShowImage(f3,'Addition of two flipped
    color planes');title('Addition of two flipped
    color planes','color','red','fontsize',4);//show
    addition of two flipped color planes
27 figure(5);ShowColorImage(f4,'Addition of color
    images');title('Addition of color images','color',
    'red','fontsize',5);//show addition of two color
    images
28 figure(6);ShowColorImage(f5,'Substract images');
    title('Substraction of color images','color','red',
    'fontsize',5);//show addition of two color
    images

```

Color image 1



Figure 1.17: Addition and subtraction operation on color images and grayscale images

Color image 2



Figure 1.18: Addition and subtraction operation on color images and grayscale images

Color image 3



Figure 1.19: Addition and subtraction operation on color images and grayscale images

Addition of two flipped color planes

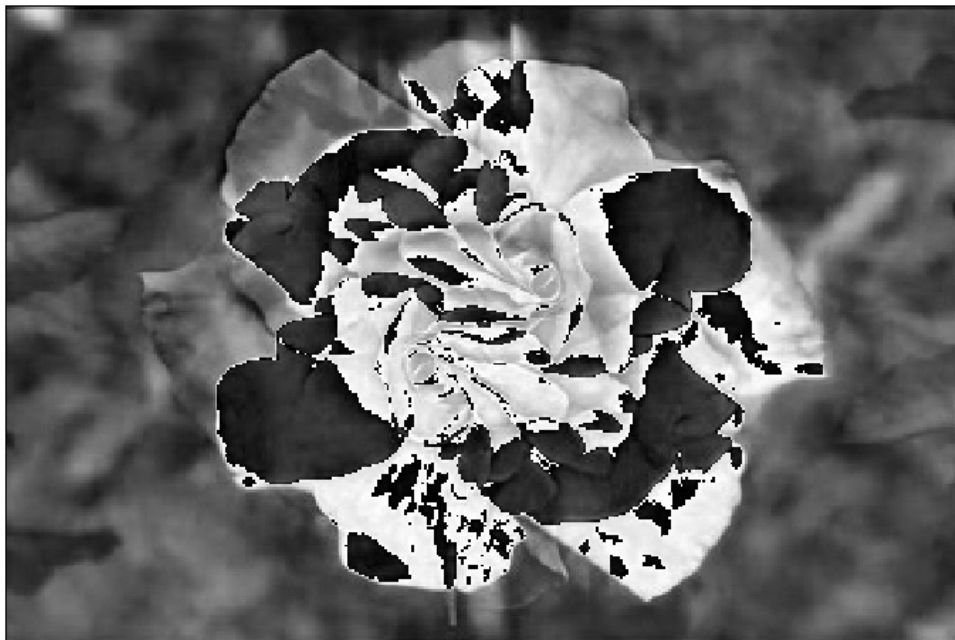


Figure 1.20: Addition and subtraction operation on color images and grayscale images

Addition of color images



Figure 1.21: Addition and subtraction operation on color images and grayscale images

check Appendix [AP 6](#) for dependency:

`colorplanes.sci`

check Appendix ?? for dependency:

`rose.BMP`

check Appendix ?? for dependency:

`rose1.bmp`

check Appendix ?? for dependency:

Substraction of color images

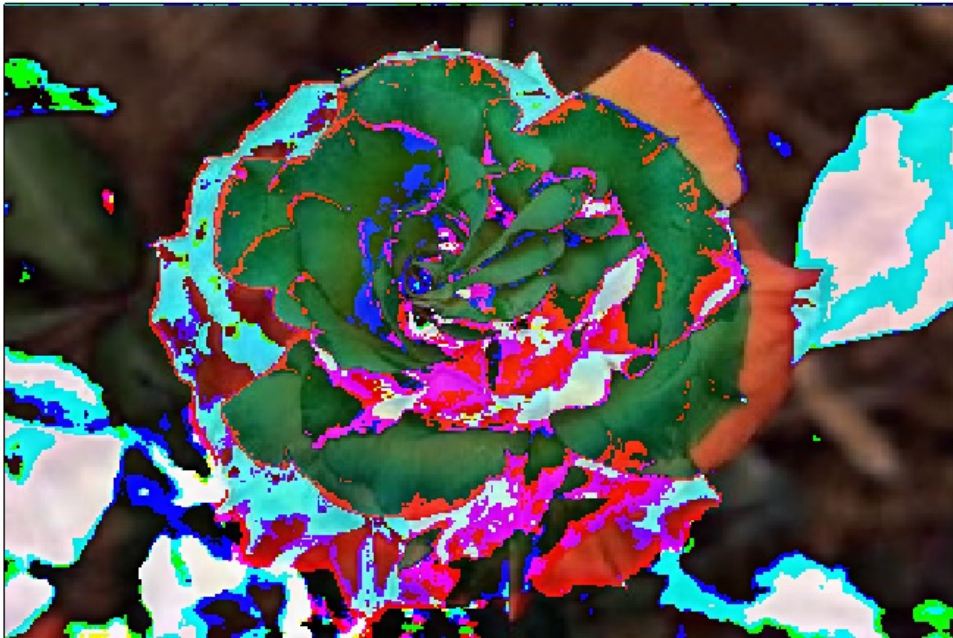


Figure 1.22: Addition and subtraction operation on color images and grayscale images

rose2.BMP

Scilab code Solution 1.5 Foramation of grayscale image from color image using multiplication and division operation

```
1 //Exp 1.5 Program Title:Foramation of grayscale
  image from color image using multiplication and
  division operation
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1) Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
  plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
  in which SIVP toolbox is installed.getSIVPpath()
  is pre-defined function in SIVP toolbox.
14 ii =imread(SIVP_PATH + 'images/rose.bmp');//read
  first color image from diractory
15 [Ri1,Gi2,Bi3]=colorplanes(ii);//call function for
  seperation of RGB color planes
16 //Gray scal image Formation
17 gr=(immultiply(Ri1,0.30)+immultiply(Gi2,0.59)+
  immultiply(Bi3,0.10));//Multiplication and
  Additon Operation On an image
18 ga=(Ri1+Gi2+Bi3)/3//average of given image by addiig
  and dividing by 3
```

Color image 1



Figure 1.23: Foramation of grayscale image from color image using multiplication and division operation

```
19
20 figure(1);ShowColorImage(ii,'original imagel');title
    ('Color image 1','color','red','fontsize',5);//
    show original color image
21 figure(2);ShowImage(gr,'Greyscale image');title('
    Grayscale image','color','red','fontsize',5);//
    show greyscale image
22 figure(3);ShowImage(ga,'Addition of color images');
    title('Averaging of color planes','color','red','
    fontsize',5);//show addition of color color
    planes
```

Grayscale image



Figure 1.24: Foramation of grayscale image from color image using multiplication and division operation

check Appendix [AP 6](#) for dependency:

`colorplanes.sci`

check Appendix ?? for dependency:

`rose.BMP`

Averaging of color planes



Figure 1.25: Foramation of grayscale image from color image using multipli-
cation and division operation

Experiment: 2

Write a program to perform sampling on a given image.

Scilab code Solution 2.0 Sampling of grey scale image

```
1 //Exp 2 Program Title: Sampling of grey scale image
2 ///environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional, 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir,
9 //3)Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed
14 ii1=imread(SIVP_PATH + 'images/rose.bmp');//read
   second color image from diractory
```

```
15     ii2=rgb2gray(ii1); //color image into gray scale
        image
16 //main program command line
17 t1=sam(ii1,2); //calling function sampling by 2
18 t2=sam(ii1,4); //calling function sampling by 4
19 t3=sam(ii1,6); //calling function sampling by 6
20 figure(1); ShowColorImage(ii1, 'Color image having
        size 256 x 384'); title('Color image having size
        256 x 384', 'color', 'red', 'fontsize', 5); //IPD tool
        box
21 figure(2); ShowImage(ii2, 'Gray scale image having
        size 256 x 384'); title('Gray scale image having
        size 256 x 384', 'color', 'red', 'fontsize', 5); //IPD
        tool box
22 figure(3); ShowImage(t1, 'Down sampling by 2'); title('
        Down sampling by 2', 'color', 'red', 'fontsize', 5);
        //IPD tool box
23 figure(4); ShowImage(t2, 'Down sampling by 4'); title('
        Down sampling by 4', 'color', 'red', 'fontsize', 5);
        //IPD tool box
24 figure(5); ShowImage(t3, 'Down sampling by 6'); title('
        Down sampling by 6', 'color', 'red', 'fontsize', 2);
        //IPD tool box
```

check Appendix ?? for dependency:

rose.BMP

check Appendix [AP 21](#) for dependency:

Color image having size 256 x 384



Figure 2.1: Sampling of grey scale image

Gray scale image having size 256 x 384



Figure 2.2: Sampling of grey scale image

Down sampling by 2



Figure 2.3: Sampling of grey scale image

Down sampling by 4



Figure 2.4: Sampling of grey scale image

`sam.sci`

Down sampling by 6



Figure 2.5: Sampling of grey scale image

Experiment: 3

Write a program to perform quantization on a given image.

Scilab code Solution 3.1 Quantization of grey scale image using 1 2 3 4 bits per pixels

```
1 //Exp 3.1 Program Title:Quantization of grey scale
   image using 1,2,3 and 4 bits per pixels
2 ///environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed
```

```
14 ii1=imread(SIVP_PATH + 'images/rose.bmp');//read
    second color image from directory
15 ii=rgb2gray(ii1);//to convert an image into
    grayscale image.
16 t5=quag(ii,4);//quantization by using 4 bits i.e 16
    gray level
17 t6=quag(ii,3);//quantization by using 3 bits i.e 8
    gray level
18 t7=quag(ii,2);//quantization by using 2 bits i.e 4
    gray level
19 t8=quag(ii,1);//quantization by using 1 bit i.e 2
    gray level
20 figure(1);ShowImage(ii,'original gray image having
    size 256 x 384');title('original grey image','
    color','red','fontsize',5);//IPD tool box
21 figure(2);ShowImage(t5,'Quantized using 16 grey
    level');title('Quantized using 16 grey level','
    color','red','fontsize',5);//IPD tool box
22 figure(3);ShowImage(t6,'Quantized using 8 grey level
    ');title('Quantized using 8 grey level','color','
    red','fontsize',5);//IPD tool box
23 figure(4);ShowImage(t7,'Quantized using 4 grey level
    ');title('Quantized using 4 grey level','color','
    red','fontsize',5);//IPD tool box
24 figure(5);ShowImage(t8,'Quantized using 2 grey level
    ');title('Quantized using 2 grey level','color','
    red','fontsize',5);//IPD tool box
```

original grey image



Figure 3.1: Quantization of grey scale image using 1 2 3 4 bits per pixels

Quantized using 16 grey level



Figure 3.2: Quantization of grey scale image using 1 2 3 4 bits per pixels

Quantized using 8 grey level



Figure 3.3: Quantization of grey scale image using 1 2 3 4 bits per pixels

Quantized using 4 grey level



Figure 3.4: Quantization of grey scale image using 1 2 3 4 bits per pixels

Quantized using 2 grey level



Figure 3.5: Quantization of grey scale image using 1 2 3 4 bits per pixels

check Appendix [AP 20](#) for dependency:

quag.sci

check Appendix ?? for dependency:

rose.BMP

Scilab code Solution 3.2 Quantization of grey scale image using 5 6 7 and 8 bits per pixels

```
1 //Exp 3.2 Program Title:Quantization of grey scale
   image using 5,6,7 and 8 bits per pixels
2 ///environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed
14 ii1=imread(SIVP_PATH + 'images/rose.bmp');//read
   second color image from diractory
15 ii=rgb2gray(ii1);//to convert an image into
   grayscale image.
16 t1=quag(ii,8);//quantization by using 8 bits i.e 256
   gray level
17 t2=quag(ii,7);//quantization by using 7 bits i.e 128
   gray level
```

```
18 t3=quag(ii,6); //quantization by using 6 bits i.e 64
    gray level
19 t4=quag(ii,5); //quantization by using 5 bits i.e 32
    gray level
20 figure(1);ShowImage(ii,'original gray image having
    size 256 x 384');title('original grey image','
    color','red','fontsize',5); //IPD tool box
21 figure(2);ShowImage(t1,'Quantized using 256 grey
    level');title('Quantized using 256 grey level','
    color','red','fontsize',5); //IPD tool box
22 figure(3);ShowImage(t2,'Quantized using 128 grey
    level');title('Quantized using 128 grey level','
    color','red','fontsize',5); //IPD tool box
23 figure(4);ShowImage(t3,'Quantized using 64 grey
    level');title('Quantized using 64 grey level','
    color','red','fontsize',5); //IPD tool box
24 figure(5);ShowImage(t4,'Quantized using 32 grey
    level');title('Quantized using 32 grey level','
    color','red','fontsize',5); //IPD tool box
```

check Appendix [AP 20](#) for dependency:

quag.sci

check Appendix ?? for dependency:

rose.BMP

original grey image



Figure 3.6: Quantization of grey scale image using 5 6 7 and 8 bits per pixels

Quantized using 256 grey level



Figure 3.7: Quantization of grey scale image using 5 6 7 and 8 bits per pixels

Quantized using 128 grey level



Figure 3.8: Quantization of grey scale image using 5 6 7 and 8 bits per pixels

Quantized using 64 grey level



Figure 3.9: Quantization of grey scale image using 5 6 7 and 8 bits per pixels

Quantized using 32 grey level



Figure 3.10: Quantization of grey scale image using 5 6 7 and 8 bits per pixels

Experiment: 4

Write a program to perform point processing for image enhancement on a given image.

Scilab code Solution 4.1 To perform a Negation operation on grayscale image and color image

```
1 //Ex 4.1 Program Title:To perform a Negation
   operation on grayscale image and color image
2 ///environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional, 64 bit
6 //Reference books names : 1) Digital Image Processing
   book author :
7 // Rafael C. Gonzalez and Richard E. Woods,
8 //2) Digital Image Processing book author :S. Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
   Thekedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
```

```

13 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed
14 i1=imread(SIVP_PATH + 'images/rose.bmp'); //read
    color image from diractory
15 i1=imresize(i1,[128,128])
16 im1=rgb2gray(i1);
17 [im2,im3,im4]=colorplanes(i1); //call function for
    seperation of RGB color planes
18 //Gray scal image Formation
19 im5=negimg(im1); //negative of grayscale image
20 im6=negimg(im2); //negative of red color plane
21 im7=negimg(im3); //negative of green color plane
22 im8=negimg(im4); //negative of blue color plane
23 im9=cat(3,im5,im6,im7); //concatenation
24 figure(1), ShowColorImage(i1, 'Color image'), title('
    Color image', 'color', 'red', 'fontsize', 5); //IPD
    Toolbox
25 figure(2); ShowImage(im1, 'Grayscale Image'); title('
    Grayscale Image', 'color', 'red', 'fontsize', 5); //
    IPD Toolbox
26 figure(3); ShowImage(im5, 'Grayscale Image'); title('
    Negative of Grayscale Image', 'color', 'red', '
    fontsize', 5); //IPD Toolbox
27 //Negation Of Red Color Plane ////////////////
28 figure(4); ShowImage(im2, 'Red color Plane '); title('
    Red color Plane', 'color', 'red', 'fontsize', 5); //
    IPD Toolbox
29 figure(5); ShowImage(im7, 'Negative Red color Plane ')
    ; title('Negative Red color Plane', 'color', 'red', '
    fontsize', 5); //IPD Toolbox
30 //Negation Of Green Color Plane ////////////////
31 figure(6); ShowImage(im3, 'Green color Plane'); title('
    Green color Plane', 'color', 'red', 'fontsize', 5); //
    IPD Toolbox
32 figure(7); ShowImage(im7, 'Negative of Green color
    Plane'); title('Negative of Green color Plane', '
    color', 'red', 'fontsize', 5); //IPD Toolbox
33 //Negation Of Blue Color Plane ////////////////

```

```
34 figure(8);ShowImage(im4,'Blue color Plane');title('
    Blue color Plane','color','red','fontsize',5);//
    IPD Toolbox
35 figure(9);ShowImage(im8,'Negative of Blue color
    Plane');title('Negative of Blue color Plane','
    color','red','fontsize',5);//IPD Toolbox
36 ////Negative of Color Image/////////////////////////////////
37 figure(10);ShowColorImage(im9,'Negative of Color
    Image');title('Negative of Color Image','color','
    red','fontsize',5);//IPD Toolbox
```

check Appendix [AP 6](#) for dependency:

colorplanes.sci

check Appendix [AP 19](#) for dependency:

negimg.sci

check Appendix ?? for dependency:

Negative of Color Image



Figure 4.1: To perform a Negation operation on grayscale image and color image

Color image



Figure 4.2: To perform a Negation operation on grayscale image and color image

Grayscale Image



Figure 4.3: To perform a Negation operation on grayscale image and color image

Negative of Grayscale image

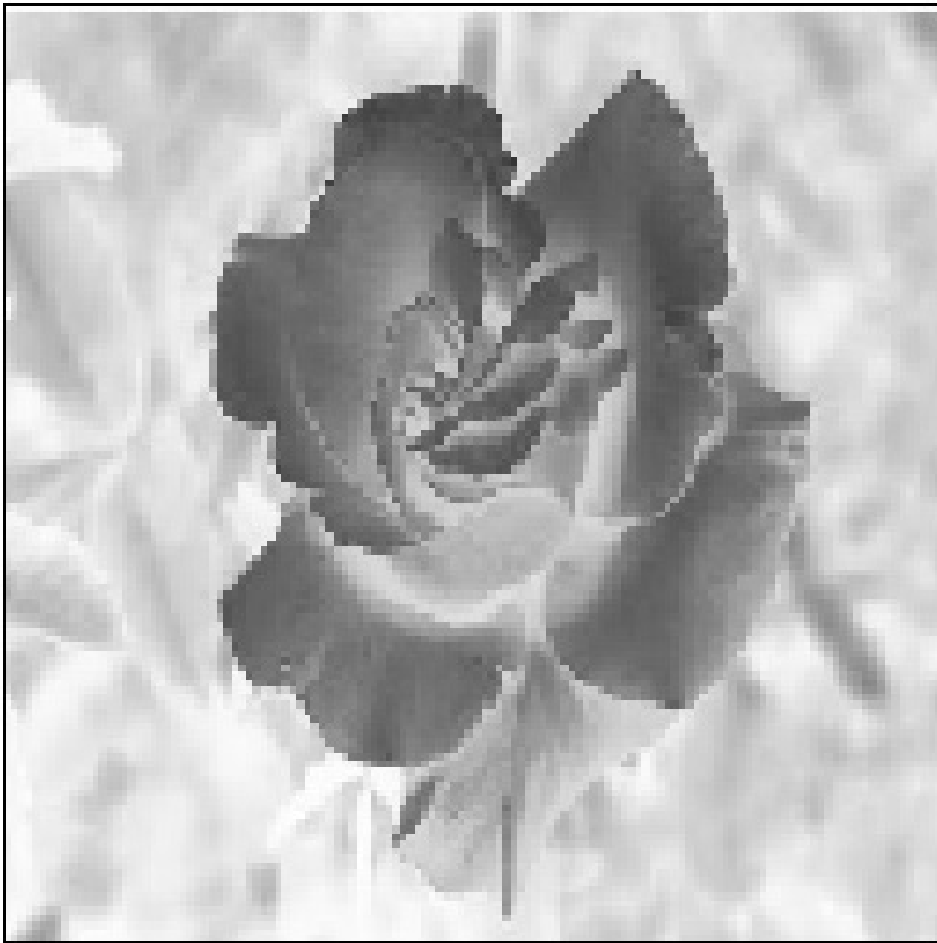


Figure 4.4: To perform a Negation operation on grayscale image and color image

Red color Plane



Figure 4.5: To perform a Negation operation on grayscale image and color image

Negative of Red color Plane

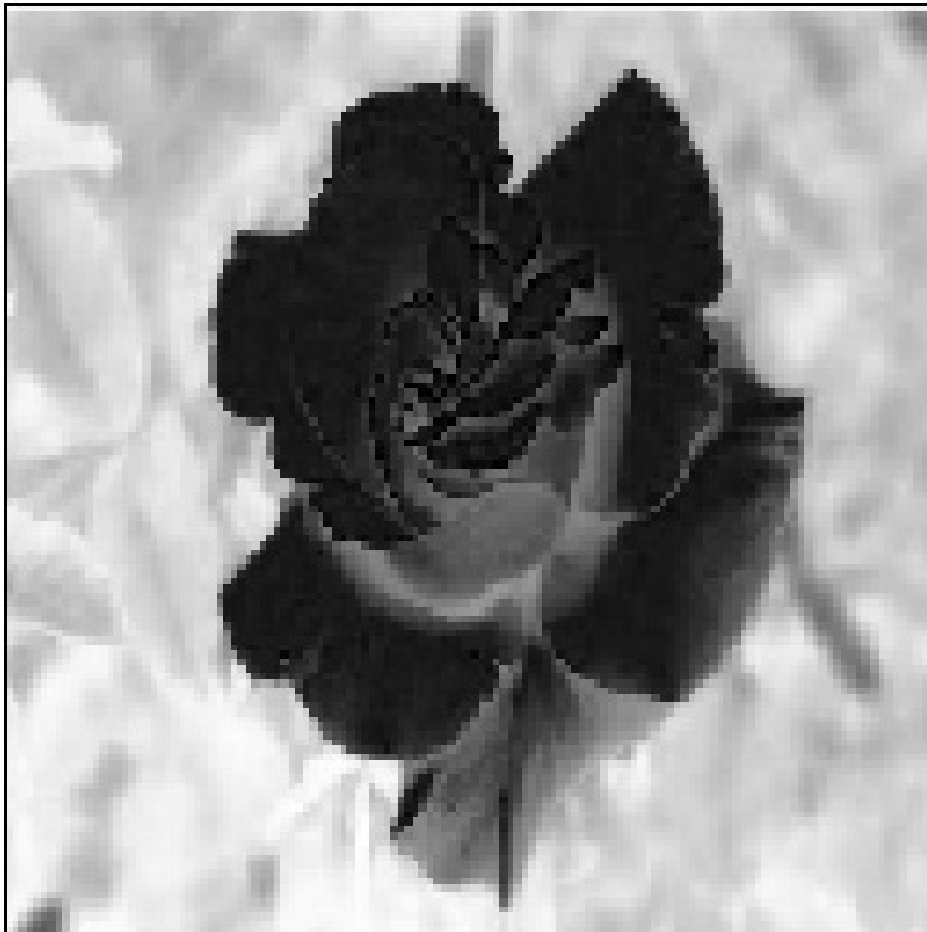


Figure 4.6: To perform a Negation operation on grayscale image and color image

Green color Plane



Figure 4.7: To perform a Negation operation on grayscale image and color image

Negative of Green color Plane

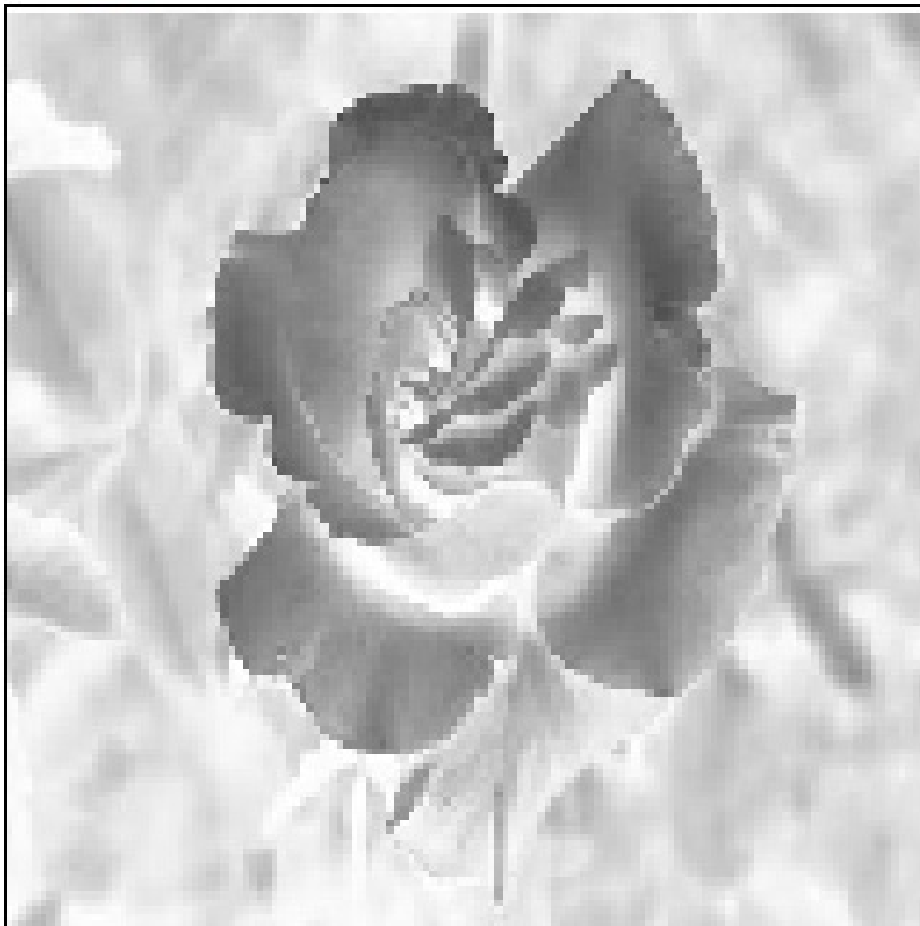


Figure 4.8: To perform a Negation operation on grayscale image and color image

Blue color Plane



Figure 4.9: To perform a Negation operation on grayscale image and color image

Negative of Blue color Plane

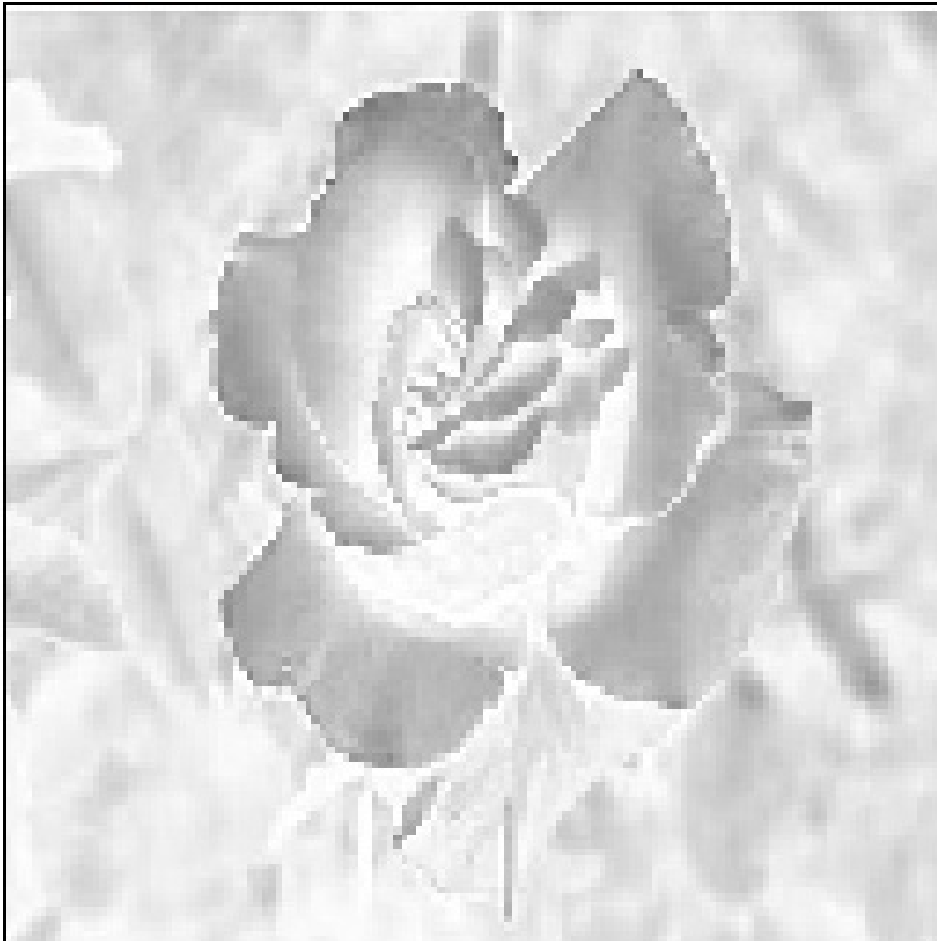


Figure 4.10: To perform a Negation operation on grayscale image and color image

rose.BMP

Scilab code Solution 4.2 To implement Thresholding of image using mean as a thshold value and half of the maximum pixel value of image

```
1 //Ex 4.2 Program Title:To implement Thresholding of
  image using mean as a thshold value and half of
  the maximum pixel value of image
2 ///environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1) Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
  plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
  in which SIVP toolbox is installed
14 ii1=imread(SIVP_PATH + 'images/rose.bmp');//read
  color image from diractory
15 im=rgb2gray(ii1);
16 [r,g,b]=colorplanes(ii1)//color planes separation
17 [im3,im2]=thres(im);//thresholding on grey scale
  image
18 [rim3,rim2]=thres(r);//thresholding on red plane
19 [gim3,gim2]=thres(g);// thresholding on green plane
20 [bim3,bim2]=thres(b);// thresholding on blue plane
21 figure(1);ShowColorImage(ii1,'color image');title('
  Color image ','color ','red ','fontsize ',6);//IPD
  toolbox
```

```

22 figure(2);ShowImage(im,'Grayscale image');title('
    Grayscale image','color','red','fontsize',6);//
    IPD toolbox
23 ///////////////////////////////////////////////////////////////////Thresholding Using mean value as a
    theshold value/////////////////////////////////////////////////////////////////
24 figure(3);ShowImage(im2,'Threshold of grayscale
    image mean is Threshold');title('Threshold of
    grayscale image when mean is Threshold','color',
    'red','fontsize',4);//IPD toolbox
25 figure(4);ShowImage(rim2,'Threshold Image mean is
    Threshold');title('Threshold of red plane when
    mean is Threshold','color','red','fontsize',4);//
    IPD toolbox
26 figure(5);ShowImage(gim2,'Threshold of green plane
    mean is Threshold');title('Thresholding of green
    plane when mean is Threshold','color','red',
    'fontsize',4);//IPD toolbox
27 figure(6);ShowImage(bim2,'Threshold of blue plane
    mean is Threshold');title('Threshold of blue
    plane when mean is Threshold','color','red',
    'fontsize',4);//IPD toolbox
28 ///////////////////////////////////////////////////////////////////Thresholding Using half of maximum
    value as a theshold value/////////////////////////////////////////////////////////////////
29 figure(7);ShowImage(im3,'Threshold of grayscale
    image thershold is max/2');title('Thresholding of
    grayscale image when threshold is max/2','color',
    'red','fontsize',4);//IPD toolbox
30 figure(8);ShowImage(rim3,'Threshold of red plane
    Image thershold is max/2');title('Thresholding of
    red plane when threshold is max/2','color','red',
    'fontsize',4);//IPD toolbox
31 figure(9);ShowImage(gim3,'Threshold Image thershold
    is max/2');title('Thresholding of green plane
    when threshold is max/2','color','red','fontsize',
    4);//IPD toolbox
32 figure(10);ShowImage(bim3,'Threshold of blue plane
    thershold is max/2');title('Thresholding of blue
    plane when threshold is max/2','color','red',

```

Thresholding of blue plane when threshold is max/2



Figure 4.11: To implement Thresholding of image using mean as a threshold value and half of the maximum pixel value of image

```
fontsize ',4); //IPD toolbox
```

Color image



Figure 4.12: To implement Thresholding of image using mean as a threshold value and half of the maximum pixel value of image

Grayscale image



Figure 4.13: To implement Thresholding of image using mean as a threshold value and half of the maximum pixel value of image

Threshold of grayscale image when mean is Threshold



Figure 4.14: To implement Thresholding of image using mean as a threshold value and half of the maximum pixel value of image

Threshold of red plane when mean is Threshold



Figure 4.15: To implement Thresholding of image using mean as a threshold value and half of the maximum pixel value of image

Thresholding of green plane when mean is Threshold



Figure 4.16: To implement Thresholding of image using mean as a threshold value and half of the maximum pixel value of image

Threshold of blue plane when mean is Threshold



Figure 4.17: To implement Thresholding of image using mean as a threshold value and half of the maximum pixel value of image

check Appendix [AP 6](#) for dependency:

`colorplanes.sci`

check Appendix ?? for dependency:

`rose.BMP`

check Appendix [AP 18](#) for dependency:

Thresholding of grayscale image when threshold is $\text{max}/2$



Figure 4.18: To implement Thresholding of image using mean as a threshold value and half of the maximum pixel value of image

Thresholding of red plane when threshold is $\text{max}/2$



Figure 4.19: To implement Thresholding of image using mean as a threshold value and half of the maximum pixel value of image

Thresholding of green plane when threshold is $\text{max}/2$



Figure 4.20: To implement Thresholding of image using mean as a threshold value and half of the maximum pixel value of image

thres.sci

Scilab code Solution 4.3 To Implement gray level slicing with background and without background

```
1 //Ex 4.3 Program Title:To Implement gray level
   slicing with background and without background
2 ///environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed
14 ii1=imread(SIVP_PATH + 'images/rose2.bmp');//read
   color image from diractory
15 im1=rgb2gray(ii1);
16 [im2,im3]=grayslic(im1);//graylevel slicing
17 [r,g,b]=colorplanes(ii1)//color planes seperation
18 [rim2,rim3]=grayslic(r);//graylevel slicing red
   plane
19 [gim2,gim3]=grayslic(g);//graylevel slicing green
   plane
20 [bim2,bim3]=grayslic(b);//graylevel slicing blue
   plane
21 figure(1); ShowColorImage(ii1, 'Color image');title('
```

```

        Color image','color','red','fontsize',6);//IPD
        tool box
22 figure(2);ShowImage(im1,'Grayscale image');title('
    Grayscale image','color','red','fontsize',6);//
    IPD tool box
23 ///////////////Gray level slicing with background method
    ///////////////
24 figure(3);ShowImage(im3,'Gray level slicing with
    background');title('Gray level slicing with
    background for grayscale image','color','red','
    fontsize',4);//IPD tool box
25 figure(4);ShowImage(rim3,'Gray level slicing with
    background');title('Gray level slicing with
    background for red plane','color','red','
    fontsize',4);//IPD tool box
26 figure(5);ShowImage(gim3,'Gray level slicing with
    background');title('Gray level slicing with
    background for green plane','color ','red','
    fontsize',4);//IPD tool box
27 figure(6);ShowImage(bim3,'Gray level slicing with
    background');title('Gray level slicing with
    background for blue plane','color ','red','
    fontsize',4);//IPD tool box
28 ///////////////Gray Level Slicing without Background
    method////////////////
29 figure(7);ShowImage(im2,'Gray level slicing without
    background');title('Gray level slicing without
    background for grayscale image','color','red','
    fontsize',4);//IPD tool box
30 figure(8);ShowImage(rim2,'Gray level slicing without
    background');title('Gray level slicing without
    background for red plane','color','red','
    fontsize',4);//IPD tool box
31 figure(9);ShowImage(gim2,'Gray level slicing without
    background');title('Gray level slicing without
    background for green plane','color ','red','
    fontsize',4);//IPD tool box
32 figure(10);ShowImage(bim2,'Gray level slicing

```

Gray level slicing without background for blue plane



Figure 4.21: To Implement gray level slicing with background and without background

```
without background');title('Gray level slicing  
without background for blue plane','color ','red'  
, 'fontsize ',4);//IPD tool box
```

Color image



Figure 4.22: To Implement gray level slicing with background and without background

Grayscale image



Figure 4.23: To Implement gray level slicing with background and without background

Gray level slicing with background

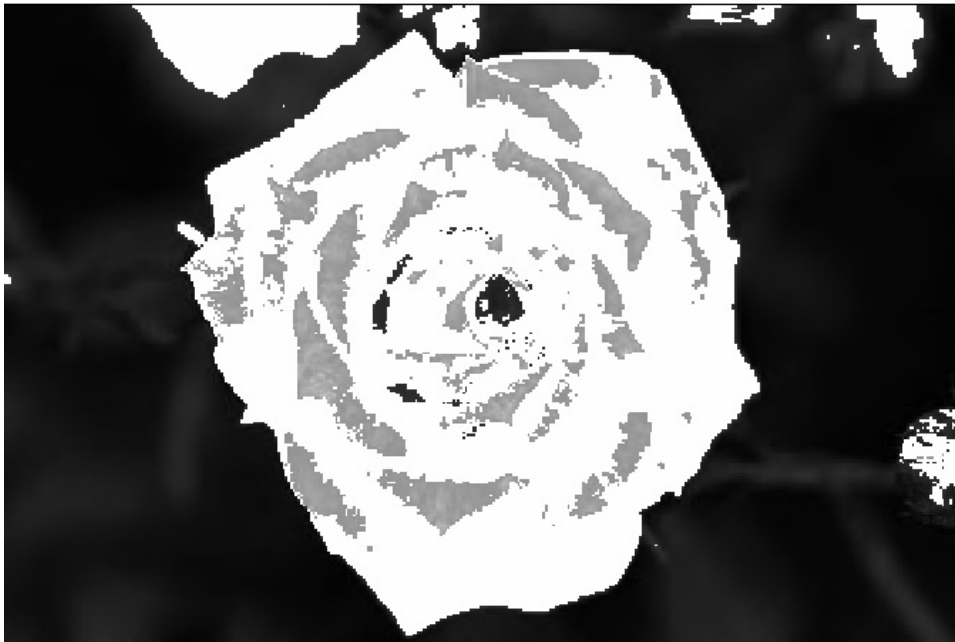


Figure 4.24: To Implement gray level slicing with background and without background

Gray level slicing with background for red plane

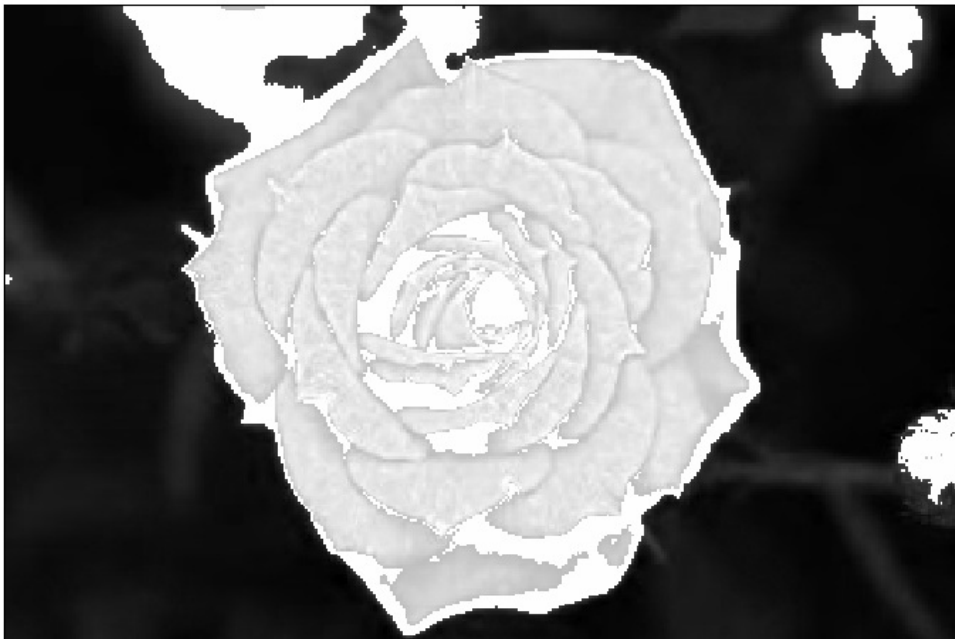


Figure 4.25: To Implement gray level slicing with background and without background

Gray level slicing with background for green plane



Figure 4.26: To Implement gray level slicing with background and without background

Gray level slicing with background for blue plane



Figure 4.27: To Implement gray level slicing with background and without background

check Appendix [AP 6](#) for dependency:

`colorplanes.sci`

check Appendix [AP 17](#) for dependency:

`grayslic.sci`

check Appendix ?? for dependency:

Gray level slicing without background



Figure 4.28: To Implement gray level slicing with background and without background

Gray level slicing without background for red plane



Figure 4.29: To Implement gray level slicing with background and without background

Gray level slicing without background for green plane



Figure 4.30: To Implement gray level slicing with background and without background

rose2.BMP

Scilab code Solution 4.4 To implement Bit Plane slicing of gray scale image

```
1 //Ex 4.4 Program Title:To implement Bit Plane
   slicing of gray scale image
2 ///environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed
14 i1=imread(SIVP_PATH + 'images/rose.bmp');//read
   color image from diractory
15 i2=rgb2gray(i1);//Color to gray scale image
16 i3=double(i2);
17 b1=modulo(i3,2);//remainder of image pixels at 1st
   division
18 b2=modulo(floor(i3/2),2);
19 b3=modulo(floor(i3/4),2);
20 b4=modulo(floor(i3/8),2);
21 b5=modulo(floor(i3/16),2);
22 b6=modulo(floor(i3/32),2);
23 b7=modulo(floor(i3/64),2);
```

```
24 b8=modulo(floor(i3/128),2);
25 figure(1);ShowImage(i2,'Greyscale image');title('
    Greyscale image','color','red','fontsize',6);//
    IPD tool box
26 figure(2);ShowImage(b1,'Bit plane 1 LSB');title('Bit
    plane 1 LSB','color','red','fontsize',6);//IPD
    tool box
27 figure(3);ShowImage(b2,'Bit plane 2');title('Bit
    plane 2','color','red','fontsize',6);//IPD tool
    box
28 figure(4);ShowImage(b3,'Bit plane 3');title('Bit
    plane 3','color','red','fontsize',6);//IPD tool
    box
29 figure(5);ShowImage(b4,'Bit plane 4');title('Bit
    plane 4','color','red','fontsize',6);//IPD tool
    box
30 figure(6);ShowImage(b5,'Bit plane 5');title('Bit
    plane 5','color','red','fontsize',6);//IPD tool
    box
31 figure(7);ShowImage(b6,'Bit plane 6');title('Bit
    plane 6','color','red','fontsize',6);//IPD tool
    box
32 figure(8);ShowImage(b7,'Bit plane 7');title('Bit
    plane 7','color','red','fontsize',6);//IPD tool
    box
33 figure(9);ShowImage(b8,'Bit plane 8 MSB');title('Bit
    plane 8 MSB','color','red','fontsize',6);//IPD
    tool box
```

Greyscale image



Figure 4.31: To implement Bit Plane slicing of gray scale image

Bit plane 1 LSB

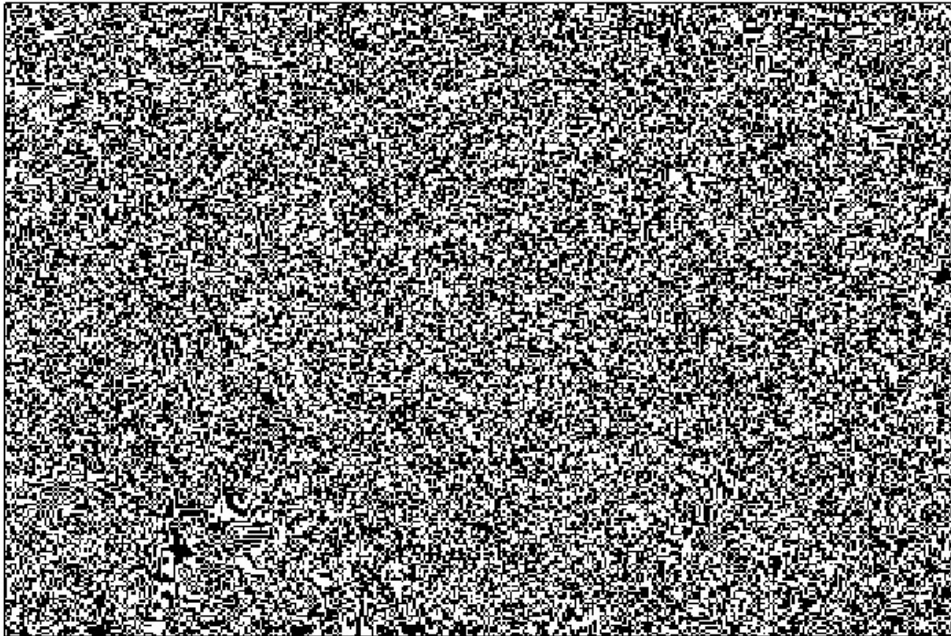


Figure 4.32: To implement Bit Plane slicing of gray scale image

Bit plane 2

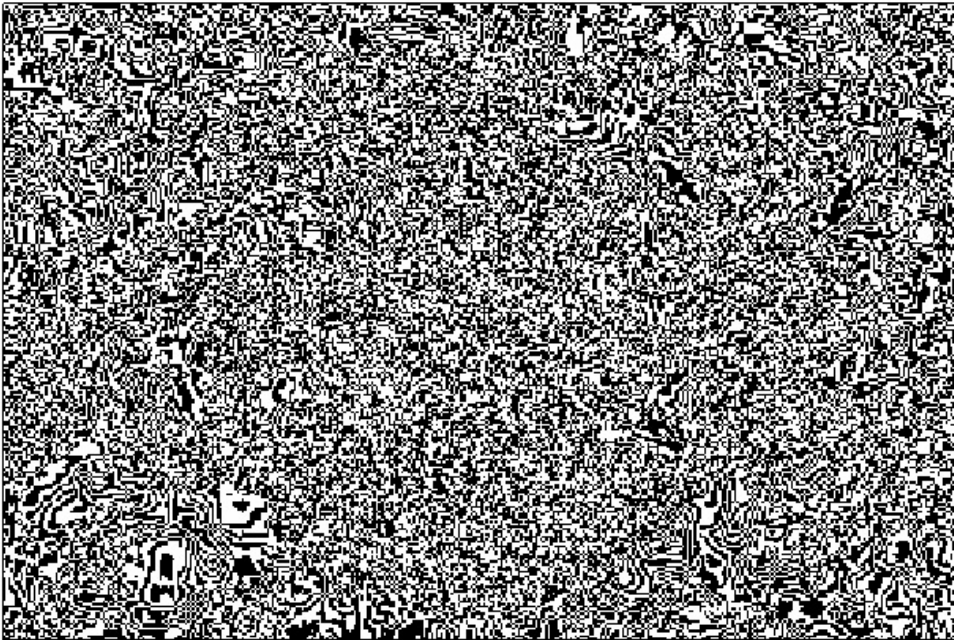


Figure 4.33: To implement Bit Plane slicing of gray scale image

Bit plane 3

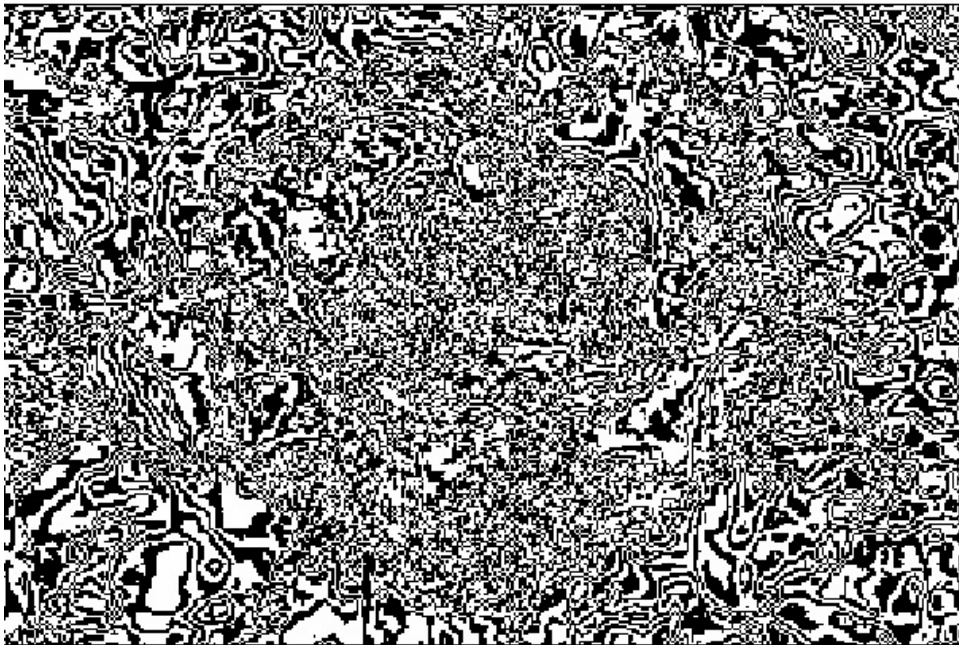


Figure 4.34: To implement Bit Plane slicing of gray scale image

Bit plane 4



Figure 4.35: To implement Bit Plane slicing of gray scale image

check Appendix ?? for dependency:

rose.BMP

Bit plane 5



Figure 4.36: To implement Bit Plane slicing of gray scale image

Bit plane 6



Figure 4.37: To implement Bit Plane slicing of gray scale image

Bit plane 7



Figure 4.38: To implement Bit Plane slicing of gray scale image

Bit plane 8 MSB



Figure 4.39: To implement Bit Plane slicing of gray scale image

Scilab code Solution 4.5 To Perform contrast Stretching using lower mean color value and upper mean color value as a lower threshold and upper threshold value respectively on given image

```
1 //Ex 4.5 Program Title:To Perform contrast
  Stretching using lower mean color value and upper
  mean color value as a lower threshold and upper
  threshold value respectively on given image
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
  Thekkedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
  plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
  in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose2.bmp'); //read
  color image from directory
15 im2=rgb2gray(im1);
16
17 [umr,lmr,im21]=upper_lower_color(im2); //contrast
  stretching of grayscale image
18 [R,G,B]=colorplanes(im1);
19 [umr1,lmr1,imr]=upper_lower_color(R); //contrast
  stretching of Red plane
20 [umr2,lmr2,img]=upper_lower_color(G); //contrast
  stretching of Green plane
21 [umr3,lmr3,imb]=upper_lower_color(B); //contrast
  stretching of blue plane
22 figure(1); ShowColorImage(im1, 'Color image');title('
```

```

    Color image','color','red','fontsize',4);//IPD
    TOOLBOX
23 figure(2);ShowImage(im2,'Grayscale image');title('
    Grayscale image','color','red','fontsize',4);//
    IPD TOOLBOX
24 figure(3);ShowImage(im21,'Grayscale image Contrast
    Stretching');title('Grayscale image Contrast
    Stretching low Bound=lower color mean, higher
    bound=upper color mean','color','red','fontsize'
    ,4);
25 figure(4);ShowImage(R,'Red plane');title('Red Plane'
    , 'color','red','fontsize',3);//IPD TOOLBOX
26 figure(5);ShowImage(imr,'Red plane Contrast
    Stretching');title('Red plane Contrast Stretching
    low Bound=lower color mean higher bound=upper
    color mean','color','red','fontsize',4);
27 figure(6);ShowImage(G,'Green plane');title('Green
    Plane','color','red','fontsize',3);//IPD TOOLBOX
28 figure(7);ShowImage(img,'Green plane Contrast
    Stretching');title('Green plane Contrast
    Stretching low Bound=lower color mean higher
    bound=upper color mean','color','red','fontsize'
    ,4);
29 figure(8);ShowImage(B,'Blue plane');title('Blue
    Plane','color','red','fontsize',3);//IPD TOOLBOX
30 figure(9);ShowImage(imb,'Blue plane Contrast
    Stretching');title('Blue plane Contrast
    Stretching low Bound=lower color mean higher
    bound=upper color mean','color','red','fontsize'
    ,4);//IPD TOOLBOX

```

Color image



Figure 4.40: To Perform contrast Stretching using lower mean color value and upper mean color value as a lower threshold and upper threshold value respectively on given image

Grayscale image



Figure 4.41: To Perform contrast Stretching using lower mean color value and upper mean color value as a lower threshold and upper threshold value respectively on given image

Grayscale image Contrast Stretching low Bound=lower color mean, higher bound=upper color mean



Figure 4.42: To Perform contrast Stretching using lower mean color value and upper mean color value as a lower threshold and upper threshold value respectively on given image

Red Plane



Figure 4.43: To Perform contrast Stretching using lower mean color value and upper mean color value as a lower threshold and upper threshold value respectively on given image

check Appendix [AP 6](#) for dependency:

`colorplanes.sci`

**Red plane Contrast Stretching low Bound=lower color mean
higher bound=upper color mean**



Figure 4.44: To Perform contrast Stretching using lower mean color value and upper mean color value as a lower threshold and upper threshold value respectively on given image

Green Plane



Figure 4.45: To Perform contrast Stretching using lower mean color value and upper mean color value as a lower threshold and upper threshold value respectively on given image

Green plane Contrast Stretching low Bound=lower color mean higher bound=upper color mean



Figure 4.46: To Perform contrast Stretching using lower mean color value and upper mean color value as a lower threshold and upper threshold value respectively on given image

Blue Plane



Figure 4.47: To Perform contrast Stretching using lower mean color value and upper mean color value as a lower threshold and upper threshold value respectively on given image

Blue plane Contrast Stretching low Bound=lower color mean higher bound=upper color mean



Figure 4.48: To Perform contrast Stretching using lower mean color value and upper mean color value as a lower threshold and upper threshold value respectively on given image

check Appendix ?? for dependency:

rose2.BMP

check Appendix [AP 15](#) for dependency:

upper_lower_color.sci

Scilab code Solution 4.6 Perform Power transform on given image

```
1 //Ex 4.6 Program Title:Perform Power transform on
  given image
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
  plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
  in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH +'images/rose1.bmp');//read
  color image from diractory
15 im1=imresize(im1,[128,128]);
16 im3=rgb2gray(im1);
17 iii1=gamm1(im3,0.5);
18 iii2=gamm1(im3,0.9);
19 iii3=gamm1(im3,1);
20 iii4=gamm1(im3,1.02);
```

```

21 iii5=gamm1(im3,1.05);
22 iii6=gamm1(im3,1.1);
23 iii7=gamm1(im3,1.2);
24 figure(1);ShowColorImage(im1,'Color image');title('
    Color image','color','red','fontsize',5);//IPD
    tool box;
25 figure(2);ShowImage(im3,'Grayscale image');title('
    original grey image','color','red','fontsize',5);
    //IPD tool box;
26 figure(3);ShowImage(iii1,'Power transform image');
    title('Power Transform Image =0.5','color','red',
    ', 'fontsize',4');//IPD toolbox
27 figure(4);ShowImage(iii2,'Power transform image');
    title('Power Transform Image =0.9','color','red',
    ', 'fontsize',4');//IPD toolbox
28 figure(5);ShowImage(iii3,'Power transform image');
    title('Power Transform Image =1','color','red',
    ', 'fontsize',4');//IPD toolbox
29 figure(6);ShowImage(iii4,'Power transform image');
    title('Power Transform Image =1.02','color','red',
    ', 'fontsize',4');//IPD toolbox
30 figure(7);ShowImage(iii5,'Power transform image');
    title('Power Transform Image =1.05','color','red',
    ', 'fontsize',4');//IPD toolbox
31 figure(8);ShowImage(iii6,'Power transform image');
    title('Power Transform Image =1.1','color','red',
    ', 'fontsize',4');
32 figure(9);ShowImage(iii7,'Power transform image');
    title('Power Transform Image =1.2','color','red',
    ', 'fontsize',4');//IPD toolbox

```

Color image



Figure 4.49: Perform Power transform on given image

original grey image



Figure 4.50: Perform Power transform on given image

Power Transform Image $\gamma=0.5$



Figure 4.51: Perform Power transform on given image

check Appendix [AP 6](#) for dependency:

`colorplanes.sci`

check Appendix [AP 14](#) for dependency:

`gamm1.sci`

check Appendix ?? for dependency:

`rose1.bmp`

Scilab code Solution 4.7 To Perform Log transform on given image

```
1 //Ex 4.7 Program Title:To Perform Log transform on
   given image
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
```

Power Transform Image $\gamma=0.9$



Figure 4.52: Perform Power transform on given image

Power Transform Image $\gamma=1$



Figure 4.53: Perform Power transform on given image

Power Transform Image $\gamma=1.02$



Figure 4.54: Perform Power transform on given image

Power Transform Image $\gamma=1.05$



Figure 4.55: Perform Power transform on given image

Power Transform Image $\gamma=1.1$



Figure 4.56: Perform Power transform on given image

Power Transform Image $\gamma=1.2$



Figure 4.57: Perform Power transform on given image

```

9 //3) Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose1.bmp');//read
   color image from diractory
15 //im11=imresize(im1,[128,128]);
16 im3=rgb2gray(im1);
17 [im4,im5,im6]=colorplanes(im1);//call function for
   seperation of RGB color planes
18 [ii]=log_im(im3);//Log Transform of Grayscale image
19 [iir]=log_im(im4);//Log Transform of red color plane
20 [iig]=log_im(im5);//Log Transform of green color
   plane
21 [iib]=log_im(im6);//Log Transform of blue color
   plane
22 figure(1);ShowColorImage(im1,'Color image');title('
   Color image','color','red','fontsize',4);//IPD
   tool box
23 figure(2);ShowImage(im3,'Grayscale image');title('
   Grayscale image','color','red','fontsize',4);//
   IPD tool box
24 figure(3);ShowImage(ii,'Log Transform of Grayscale
   image');title('Log Transform of Grayscale image',
   'color','red','fontsize',4);//IPD tool box
25 figure(4);ShowImage(iir,'Log Transform of Red color
   plane image');title('Log Transform of Red color
   plane image','color','red','fontsize',4);//IPD
   tool box
26 figure(5);ShowImage(iig,'Log Transform of Green
   color plane image');title('Log Transform of Green
   color plane image','color','red','fontsize',4);
   //IPD tool box
27 figure(6);ShowImage(iib,'Log Transform of Blue color

```

Color image



Figure 4.58: To Perform Log transform on given image

```
plane image');title('Log Transform of Blue color  
plane image','color','red','fontsize',4);//IPD  
tool box
```

check Appendix [AP 6](#) for dependency:

Grayscale image



Figure 4.59: To Perform Log transform on given image

LOG Transform of Grayscale image



Figure 4.60: To Perform Log transform on given image

LOG Transform of Red color plane image



Figure 4.61: To Perform Log transform on given image

LOG Transform of Green color plane image



Figure 4.62: To Perform Log transform on given image

LOG Transform of Blue color plane image



Figure 4.63: To Perform Log transform on given image

colorplanes.sci

check Appendix [AP 13](#) for dependency:

log_im.sci

check Appendix ?? for dependency:

rose1.bmp

Experiment: 5

Write a program to perform image enhancement using neighborhood processing on a given image.

Scilab code Solution 5.1 Image Enhancement using neighbourhood processing Average filtering

```
1 //Exp 5.1 Program Title:Image Enhancement using
   neighbourhood processing Average filtering
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
```

```

12 xdel(winsid());///clear all the previous figures
    plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');///read
    color image from directory
15 im2=rgb2gray(im1);///grayscale image
16 im3=imnoise(im2, 'gaussian',0.01);///adding Gaussian
    noise in image
17 im4=double(im3);
18 w=1/9*[1,1,1;1,1,1;1,1,1]//average filtering mask
19 [av1]=masking(im4,w);///convolution process
20 figure(1);ShowColorImage(im1, 'Color image');title('
    Color image', 'color', 'red', 'fontsize',4);///IPD
    tool box
21 figure(2);ShowImage(im2, 'Grayscale image');title('
    Grayscale image', 'color', 'red', 'fontsize',4);///
    IPD tool box
22 figure(3);ShowImage(im3, 'Image after adding gaussian
    noise');title('Image after adding gaussian noise
    ', 'color', 'red', 'fontsize',4);///IPD tool box
23 figure(4);ShowImage(av1, 'Image after average
    filtering');title('Image after average filtering'
    , 'color', 'red', 'fontsize',4);///IPD tool box

```

check Appendix [AP 12](#) for dependency:

masking.sci

check Appendix ?? for dependency:

rose.BMP

Color image



Figure 5.1: Image Enhancement using neighbourhood processing Average filtering

Grayscale image



Figure 5.2: Image Enhancement using neighbourhood processing Average filtering

Image after adding gaussian noise



Figure 5.3: Image Enhancement using neighbourhood processing Average filtering

Image after average filtering



Figure 5.4: Image Enhancement using neighbourhood processing Average filtering

Scilab code Solution 5.2 Image Enhancement using neighbourhood processing Median filtering

```
1 //Exp 5.2 Program Title:Image Enhancement using
   neighbourhood processing Median filtering
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1) Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose1.bmp');//read
   color image from directory
15 im2=rgb2gray(im1);//grayscale image
16 im3=imnoise(im2,'salt & pepper',0.2);//adding salt &
   pepper noise in image
17 [r,c]=size(im3);
18 for i=2:1:r-1
19     for j=2:1:c-1
20         a=[im3(i-1,j-1),im3(i-1,j),im3(i-1,j+1),...
21             im3(i,j-1),im3(i,j),im3(i,j+1),...
22             im3(i+1,j-1),im3(i+1,j),im3(i+1,j+1)];
23         m=gsort(a); //sorting the elements of a in
           ascending order ;
```

```

24             av1(i,j)=m(5); //select median value
25         end
26 end
27 figure(1); ShowColorImage(im1, 'Color Image'); title('
    Color Image', 'color', 'red', 'fontsize', 6); //IPD
    Toolbox
28 figure(2); ShowImage(im2, 'Grayscale Image'); title('
    Grayscale Image', 'color', 'red', 'fontsize', 6); //
    IPD Toolbox
29 figure(3); ShowImage(im3, 'Salt and Pepper noise Image
    '); title('Salt and Pepper noise Image', 'color', '
    red', 'fontsize', 6); //IPD Toolbox
30 figure(4); ShowImage(av1, 'Median Filtered Image');
    title('Median Filtered Image', 'color', 'red', '
    fontsize', 6); //IPD Toolbox

```

check Appendix ?? for dependency:

rose1.bmp

Scilab code Solution 5.3 mage Enhancement using neighbourhood processing high boost filter

```

1 //Exp 5.3 Program Title:Image Enhancement using
    neighbourhood processing high boost filter
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2

```

Color Image



Figure 5.5: Image Enhancement using neighbourhood processing Median filtering

Grayscale Image



Figure 5.6: Image Enhancement using neighbourhood processing Median filtering

Salt and Pepper noise Image

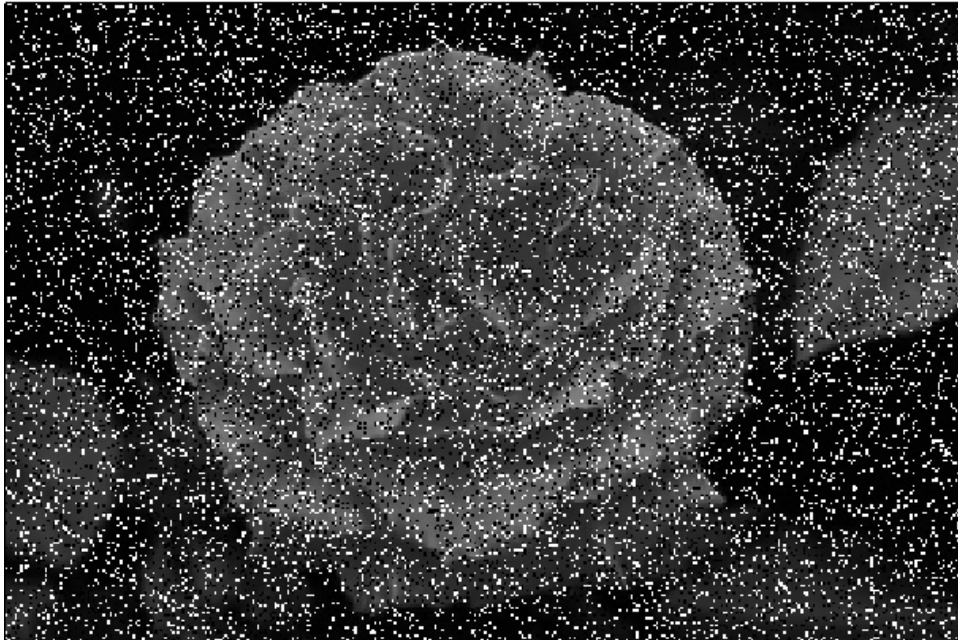


Figure 5.7: Image Enhancement using neighbourhood processing Median filtering

Median Filtered Image



Figure 5.8: Image Enhancement using neighbourhood processing Median filtering

```

5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');//read
   color image from diractory
15 im2=rgb2gray(im1);//grayscale image
16 im3=double(im2);
17 w=[-1 -1 -1; -1 8.9 -1; -1 -1 -1];//high boost
   filter mask
18 w1=[-1 -1 -1; -1 9.1 -1; -1 -1 -1];//high boost
   filter mask
19 w2=[-1 -1 -1; -1 9.3 -1; -1 -1 -1];//high boost
   filter mask
20 w3=[-1 -1 -1; -1 10 -1; -1 -1 -1];//high boost
   filter mask
21 [av1]=masking(im3,w);//convolution process
22 [av2]=masking(im3,w1);//convolution process
23 [av3]=masking(im3,w2);//convolution process
24 [av4]=masking(im3,w3);//convolution process
25 figure(1);ShowColorImage(im1,'Color image');title('
   Color image','color','red','fontsize',6);//IPD
   Toolbox
26 figure(2);ShowImage(im2,'Grayscale image');title('
   Grayscale image','color','red','fontsize',6);//
   IPD Toolbox
27 figure(3);ShowImage(av1,'High boost filtered Image')
   ;title('High boost filtered Image','color','red',
   'fontsize',6);//IPD Toolbox

```

Color image



Figure 5.9: mage Enhancement using neighbourhood processing high boost filter

```
28 figure(4); ShowImage(av2, 'High boost filtered Image  
    for 9.1 ');title('High boost filtered Image for  
    9.1 ', 'color ', 'red ', 'fontsize ',6);//IPD Toolbox  
29 figure(5); ShowImage(av3, 'High boost filtered Image  
    for 9.3 ');title('High boost filtered Image for  
    9.3 ', 'color ', 'red ', 'fontsize ',6);//IPD Toolbox  
30 figure(6); ShowImage(av4, 'High boost filtered Image  
    for 10 ');title('High boost filtered Image for 10 '  
    , 'color ', 'red ', 'fontsize ',6);//IPD Toolbox
```

Grayscale image



Figure 5.10: mage Enhancement using neighbourhood processing high boost filter

High boost filtered Image



Figure 5.11: Image Enhancement using neighbourhood processing high boost filter

check Appendix [AP 12](#) for dependency:

`masking.sci`

check Appendix ?? for dependency:

`rose.BMP`

High boost filtered Image for 9.1



Figure 5.12: Image Enhancement using neighbourhood processing high boost filter

High boost filtered Image for 9.3



Figure 5.13: mage Enhancement using neighbourhood processing high boost filter

High boost filtered Image for 10



Figure 5.14: Image Enhancement using neighbourhood processing high boost filter

Scilab code Solution 5.4 Image Enhancement using neighbourhood processing high pass filter

```
1 //Exp 5.4 Program Title:Image Enhancement using
   neighbourhood processing high pass filter
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1) Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');//read
   color image from diractory
15 im2=rgb2gray(im1);//grayscale image
16 im4=double(im2);
17 w=[-1 -1 -1; -1 8 -1; -1 -1 -1];//high pass filter
   mask
18 w1=[-2 -2 -2; -2 16 -2; -2 -2 -2];//high pass filter
   mask
19 [av1]=masking(im4,w);//convolution process
20 [av2]=masking(im4,w1);//convolution process
21 figure(1);ShowColorImage(im1,'Color image');title('
   Color image','color','red','fontsize',6);//IPD
   Toolbox
```

Color image



Figure 5.15: Image Enhancement using neighbourhood processing high pass filter

```
22 figure(2); ShowImage(im2, 'greyscale image'); title('
    greyscale image', 'color', 'red', 'fontsize', 6); //
    IPD Toolbox
23 figure(3); ShowImage(av1, 'High Pass filtered Image');
    title('High Pass filtered Image', 'color', 'red', '
    fontsize', 6); //IPD Toolbox
24 figure(4); ShowImage(av2, 'High Pass filtered Image2')
    ;title('High Pass filtered Image2', 'color', 'red',
    'fontsize', 6); //IPD Toolbox
```

Grayscale image



Figure 5.16: Image Enhancement using neighbourhood processing high pass filter

High Pass filtered Image



Figure 5.17: Image Enhancement using neighbourhood processing high pass filter

check Appendix [AP 12](#) for dependency:

`masking.sci`

check Appendix ?? for dependency:

`rose.BMP`

High Pass filtered Image2

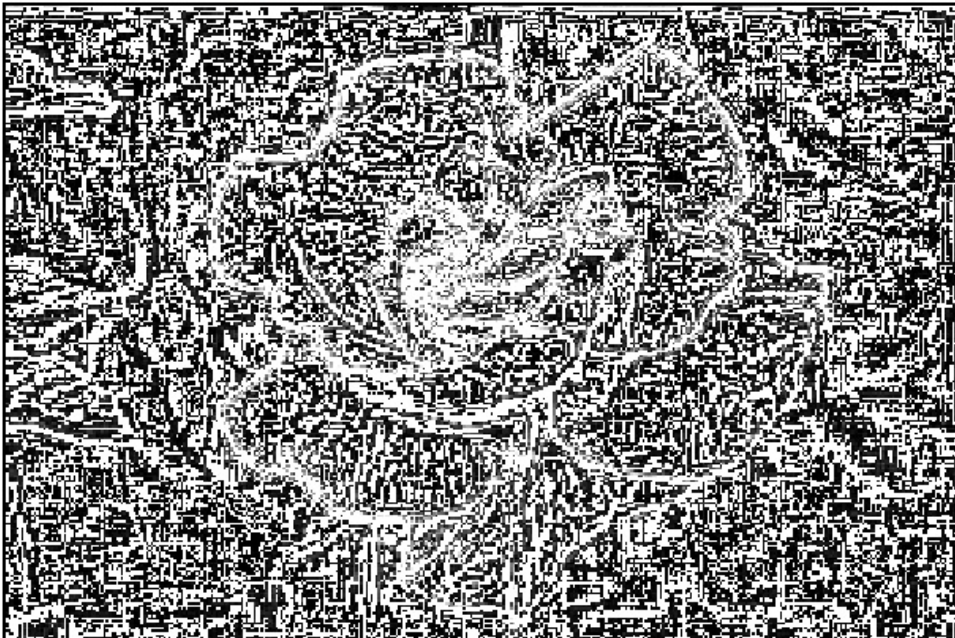


Figure 5.18: Image Enhancement using neighbourhood processing high pass filter

Experiment: 6

Write a program to perform Histogram and Histogram equalization on a given image.

Scilab code Solution 6.1 To implement Histogram and histogram equalization of grayscale image

```
1 //Exp 6.1 Program Title:To implement Histogram and
   histogram eqlization of grayscale image
2 ///environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
```

```

13 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');//read
    color image from diractory//converting RGB to
    grayscale
15 im2=rgb2gray(im1);
16 [h,eq,q]=stogram(im2);//Histogram and equalized
    histogram of gray scale image
17 figure(1);ShowColorImage(im1,'Color Image');title('
    Color image','color','red','fontsize',6);
18 figure(2);ShowImage(im2,'grayscale Image');title('
    grayscale image','color','red','fontsize',6);
19 figure(3);bar(h,'1','red');title('Histogram of
    grayscale image','color','blue','fontsize',6);
    xlabel('Intensity Levels','color','green','
    fontsize',3);ylabel('Frequency','color','red','
    fontsize',3);
20 figure(4);bar(q,'1','red');title('Equalized
    Histogram of grayscale image','color','blue','
    fontsize',6); xlabel('Intensity Levels','color','
    red','fontsize',3);ylabel('Frequency','color','
    red','fontsize',3);
21 figure(5);ShowImage(eq,'Equalized Image');title('
    Equalized grayscale image','color','red','
    fontsize',4);

```

check Appendix ?? for dependency:

rose.BMP

Color image



Figure 6.1: To implement Histogram and histogram equalization of grayscale image

Grayscale image



Figure 6.2: To implement Histogram and histogram equalization of grayscale image

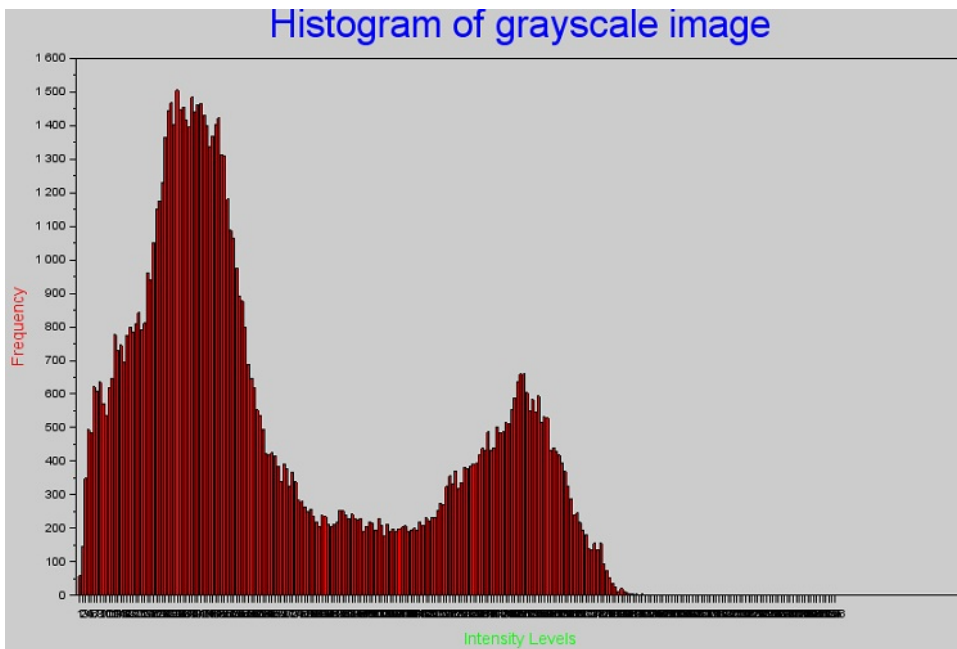


Figure 6.3: To implement Histogram and histogram equalization of grayscale image

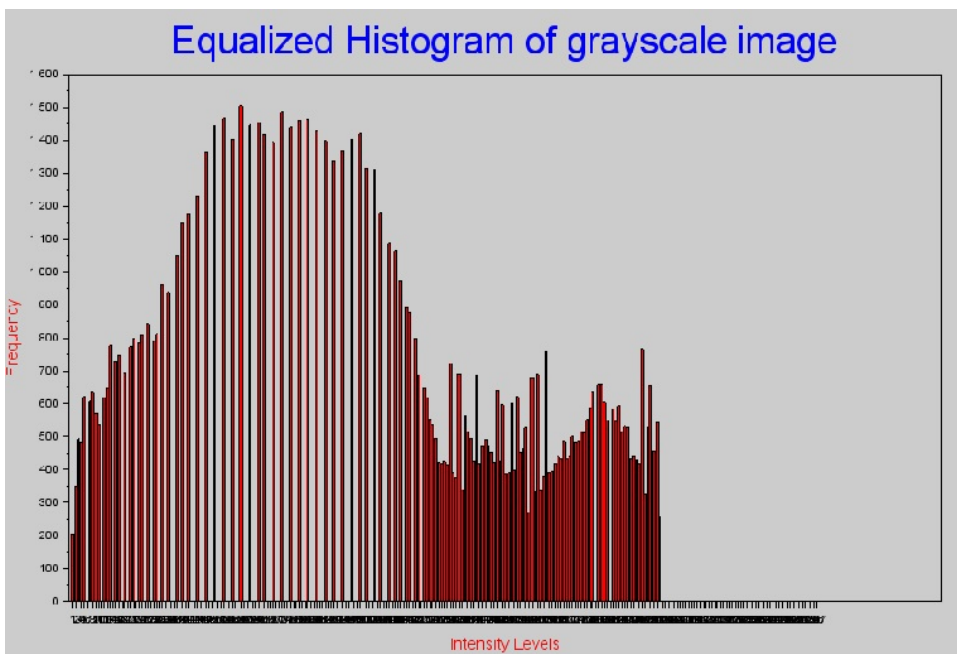


Figure 6.4: To implement Histogram and histogram equalization of grayscale image

Equalized grayscale image



Figure 6.5: To implement Histogram and histogram equalization of grayscale image

check Appendix [AP 11](#) for dependency:

stogram.sci

Scilab code Solution 6.2 To Implement Histogram and histogram equalization of color image

```
1 //Exp 6.2 Program Title: To Implement Histogram and
  histogram equalization of color image
2 ///environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1) Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
  plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
  in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');//read
  color image from diractory
15 im1=imresize(im1,[128,128]);
16 [r,g,b]=colorplanes(im1);//seperation of colorplanes
17 [hr,eqr,qr]=stogram(r);//Histogram and equalized
  histogram of red plane image
18 [hg,eqg,qg]=stogram(g);//Histogram and equalized
  histogram of green plane image
19 [hb,eqb,qb]=stogram(b);//Histogram and equalized
  histogram of blue plane image
20 aa=cat(3,eqr,eqg,eqb);
```

```

21 figure(1);bar(hr,'1','red');title('Red plane
    Histogram','color','blue','fontsize',2); xlabel('
    Intensity Levels','color','red','fontsize',2);
    ylabel('Frequency','color','red','fontsize',2);//
    IPD Toolbox
22 figure(2);bar(qr,'1','red');title('Equalized
    Histogram Red plane','color','blue','fontsize',2)
    ; xlabel('Intensity Levels','color','red','
    fontsize',2);ylabel('Frequency','color','red','
    fontsize',2);//IPD Toolbox
23 figure(3);ShowImage(eqr,'Red plane Equalized Image')
    ;title('Red plane Equalized image','color','red'
    , 'fontsize',2)//IPD Toolbox
24 figure(4);bar(hg,'1','red');title('Green plane
    Histogram','color','blue','fontsize',2); xlabel('
    Intensity Levels','color','red','fontsize',3);
    ylabel('Frequency','color','red','fontsize',2);//
    IPD Toolbox
25 figure(5);bar(qg,'1','red');title('Equalized
    Histogram Green plane','color','blue','fontsize'
    ,2); xlabel('Intensity Levels','color','red','
    fontsize',2);ylabel('Frequency','color','red','
    fontsize',2);//IPD Toolbox
26 figure(6);ShowImage(eqg,'Green plane Equalized Image
    ');title('Green plane Equalized image','color','
    red','fontsize',2)//IPD Toolbox
27 figure(7);bar(hb,'1','red');title('Blue plane
    Histogram','color','blue','fontsize',2); xlabel('
    Intensity Levels','color','red','fontsize',3);
    ylabel('Frequency','color','red','fontsize',2);//
    IPD Toolbox
28 figure(8);bar(qb,'1','green');title('Equalized
    Histogram Blue plane','color','blue','fontsize'
    ,2); xlabel('Intensity Levels','color','red','
    fontsize',2);ylabel('Frequency','color','red','
    fontsize',2);//IPD Toolbox
29 figure(9);ShowImage(eqb,'Blue plane Equalized Image'
    );title('Blue plane Equalized image','color','

```

```
    red','fontsize',2)//IPD Toolbox  
30 figure(10);ShowColorImage(aa,'Equalized color image  
    Image');title('Equalized color image Image','  
    color','red','fontsize',2)//IPD Toolbox
```

check Appendix [AP 6](#) for dependency:

colorplanes.sci

check Appendix ?? for dependency:

rose.BMP

check Appendix [AP 11](#) for dependency:

stogram.sci

Equalized color image Image



Figure 6.6: To Implement Histogram and histogram equalization of color image

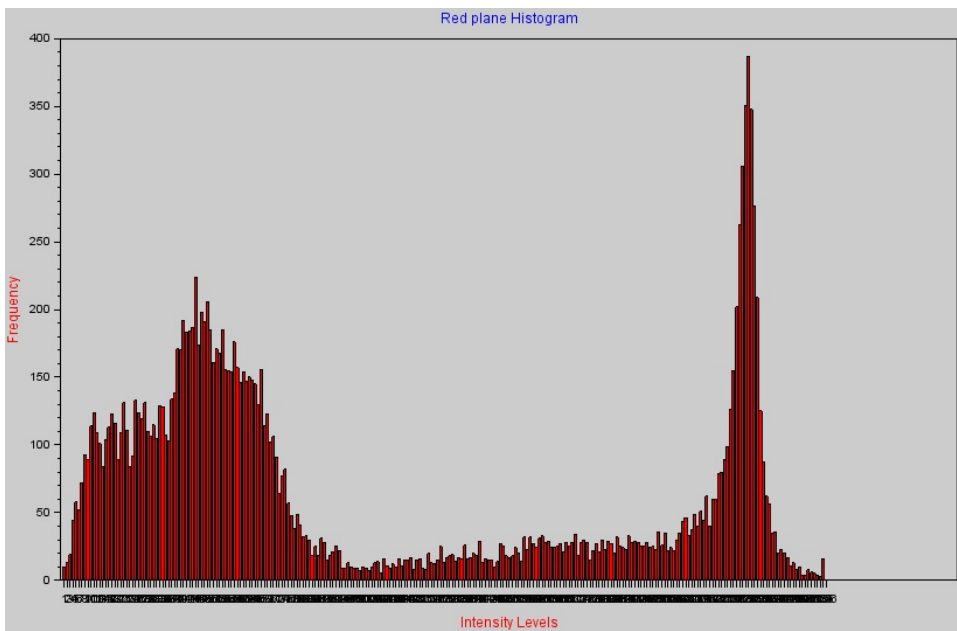


Figure 6.7: To Implement Histogram and histogram equalization of color image

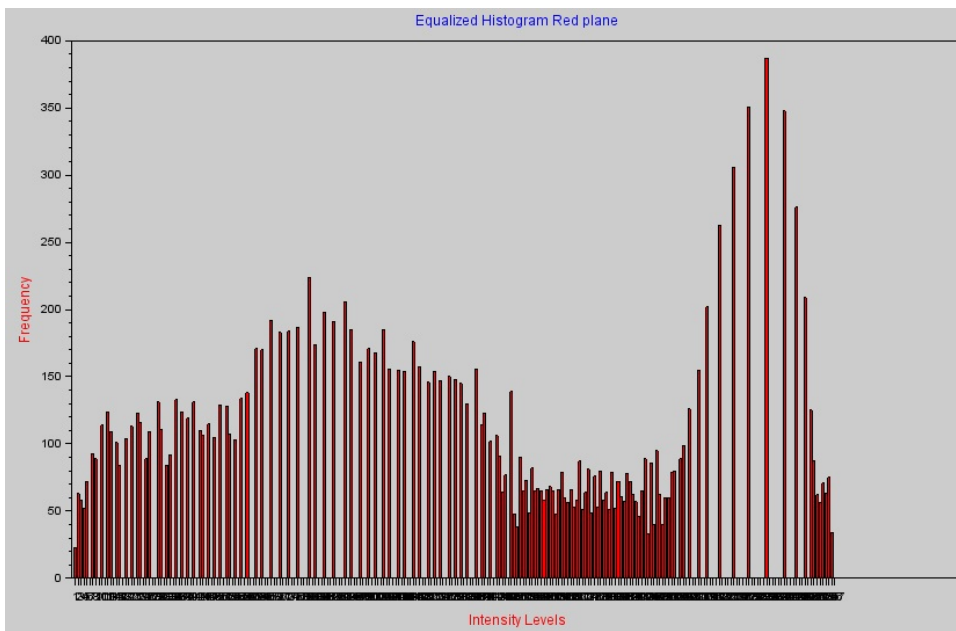


Figure 6.8: To Implement Histogram and histogram equalization of color image

Red plane Equalized image



Figure 6.9: To Implement Histogram and histogram equalization of color image

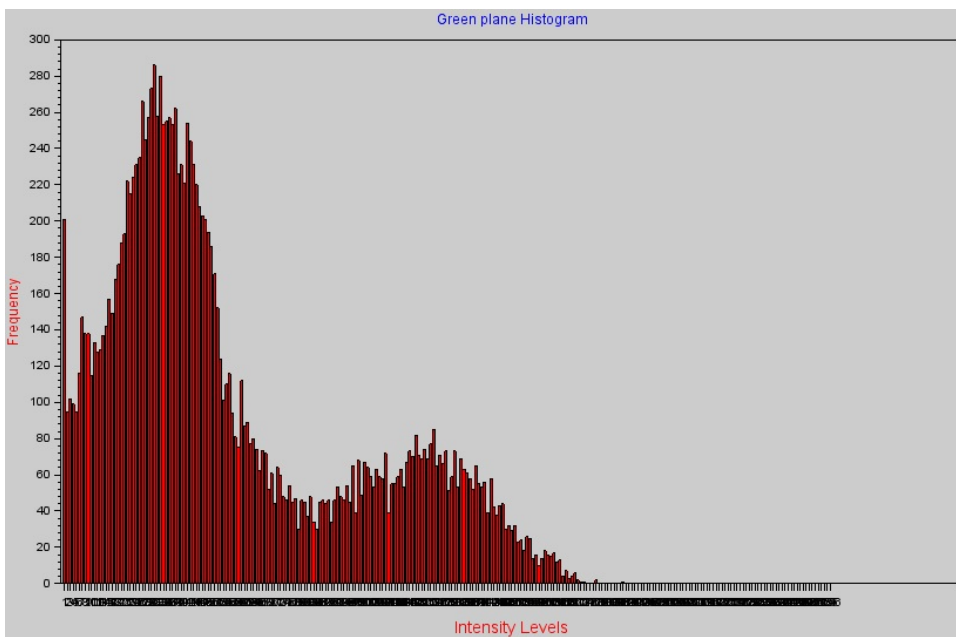


Figure 6.10: To Implement Histogram and histogram equalization of color image

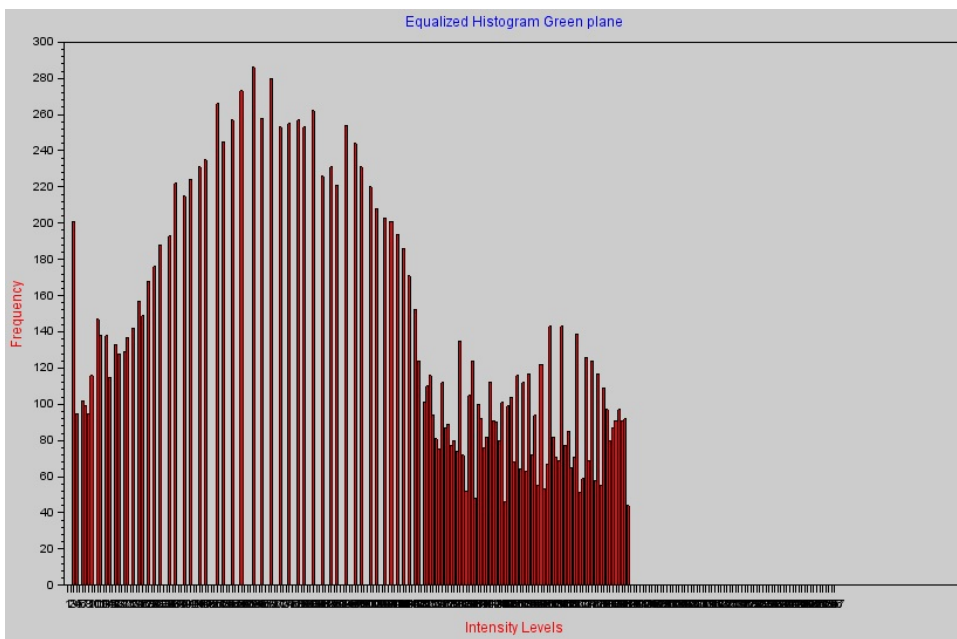


Figure 6.11: To Implement Histogram and histogram eqlization of color image

Green plane Equalized image



Figure 6.12: To Implement Histogram and histogram equalization of color image

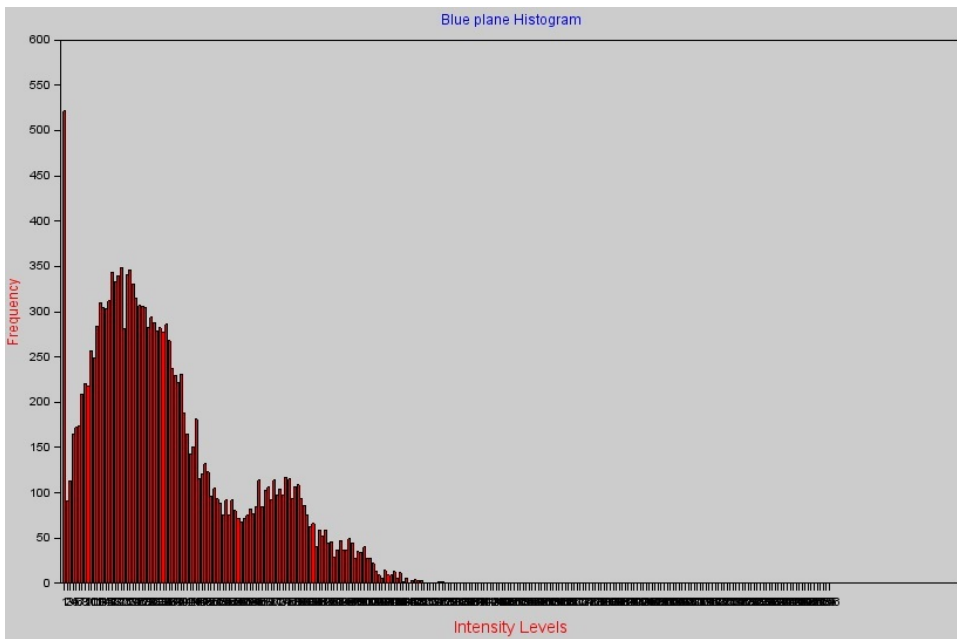


Figure 6.13: To Implement Histogram and histogram eqlization of color image

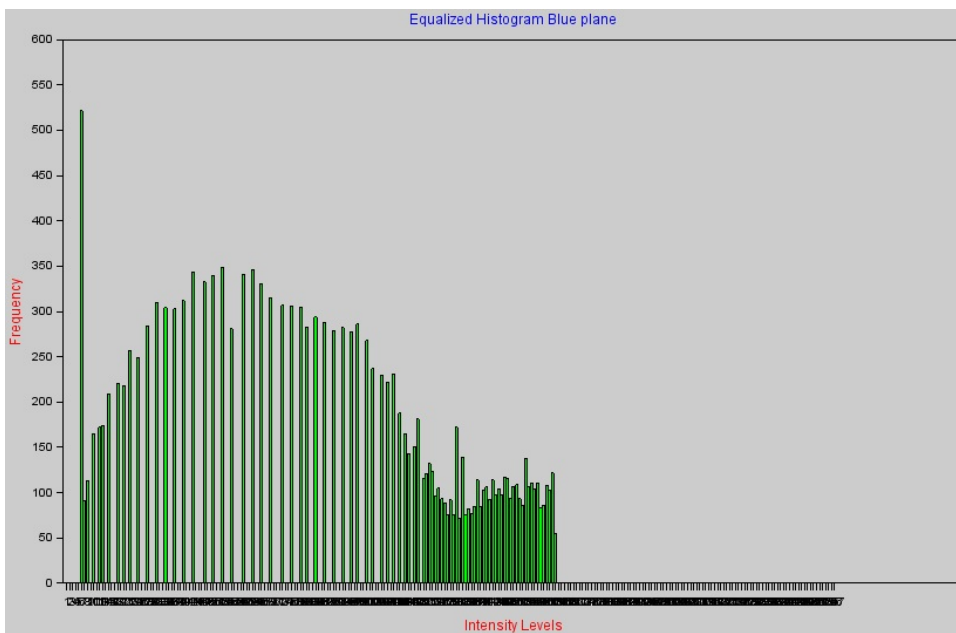


Figure 6.14: To Implement Histogram and histogram equalization of color image

Blue plane Equalized image



Figure 6.15: To Implement Histogram and histogram equalization of color image

Experiment: 7

Write a program to perform image segmentation using edge detection on a given image.

Scilab code Solution 7.1 To Implement Image segmentation based on discontinuities using edge detection Ordinary operator

```
1 //Exp 7.1 Program Title:To Implement Image
   segmentation based on discontinuities using edge
   detection Ordinary operator
2 ///environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
```

```

plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose1.bmp');//read
    color image from directory
15 im2=rgb2gray(im1);
16 //ordinary mask
17 M1=[1,0;-1,0]; //X-Gradientmask
18 M2=[1 -1;0 0]; //Y-Gradientmask
19 [im3,im4]=masking2x2(im2,M1,M2) //convolution
20 im5=im3+im4; //total edges
21 im5=uint8(im5);
22 figure(1); ShowColorImage(im1, 'Original Color image')
    ;title('original color image','color','red','
    fontsize',6'); //IPDtoolbox
23 figure(2); ShowImage(im2, ' Grayscale image');title('
    Grayscale image','color','red','fontsize',6'); //
    IPDtoolbox
24 figure(3); ShowImage(im3, ' X Gradient Edges');title('
    X Gradient Edges','color','red','fontsize',6'); //
    IPDtoolbox
25 figure(4); ShowImage(im4, 'Y Gradient Edges');title('Y
    Gradient Edges','color','red','fontsize',6'); //
    IPDtoolbox
26 figure(5); ShowImage(im5, 'Total Edges');title('Total
    Edges of given image','color','red','fontsize'
    ,6'); //IPDtoolbox

```

Color Image



Figure 7.1: To Implement Image segmentation based on discontinuities using edge detection Ordinary operator

Grayscale Image



Figure 7.2: To Implement Image segmentation based on discontinuties using edge detection Ordinary operator

X Gradient Edges

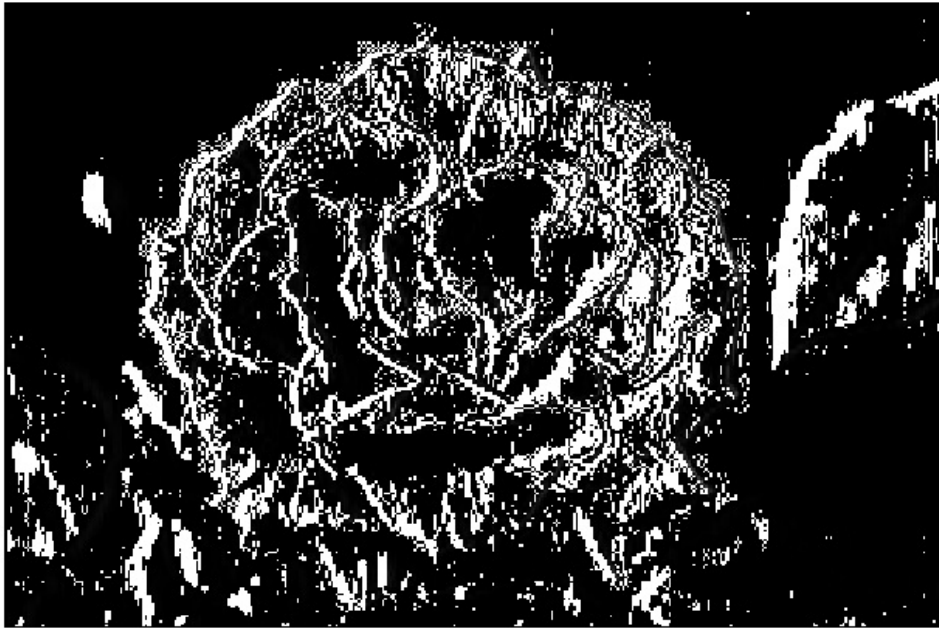


Figure 7.3: To Implement Image segmentation based on discontinuities using edge detection Ordinary operator

Y Gradient Edges



Figure 7.4: To Implement Image segmentation based on discontinuities using edge detection Ordinary operator

Total Edges of given image



Figure 7.5: To Implement Image segmentation based on discontinuities using edge detection Ordinary operator

check Appendix [AP 9](#) for dependency:

masking2x2.sci

check Appendix ?? for dependency:

rose1.bmp

Scilab code Solution 7.2 To Implement Image segmentation discontinuities using edge detection Robert Operator

```
1 //Exp 7.2 Program Title:To Implement Image
   segmentation discontinuities using edge detection
   Robert Operator
2 ///environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1) Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');//read
   color image from directory
15 im2=rgb2gray(im1);
16 //robert mask
17 M1=[1 0;0 -1];//X-Gradientmask
18 M2=[0 1;-1 0];//Y-Gradientmask
```

```
19 [im3,im4]=masking2x2(im2,M1,M2); //convolution
20 im5=im3+im4; //Totaledges
21 im5=uint8(im5);
22 figure(1);ShowColorImage(im1,'Original Color image')
    ;title('original color image','color','red','
    fontsize','6'); //IPDtoolbox
23 figure(2);ShowImage(im2,' Grayscale image');title('
    Grayscale image','color','red','fontsize','6'); //
    IPDtoolbox
24 figure(3);ShowImage(im3,' X Gradient Edges');title('
    X Gradient Edges','color','red','fontsize','6'); //
    IPDtoolbox
25 figure(4);ShowImage(im4,'Y Gradient Edges');title('Y
    Gradient Edges','color','red','fontsize','6'); //
    IPDtoolbox
26 figure(5);ShowImage(im5,'Total Edges');title('Total
    Robert mask Edges of given image','color','red','
    fontsize','6'); //IPDtoolbox
```

check Appendix [AP 9](#) for dependency:

masking2x2.sci

check Appendix ?? for dependency:

rose.BMP

Color image



Figure 7.6: To ImplementImage segmentation discontinuties using edge de-
tection Robert Operator

Grayscale image



Figure 7.7: To ImplementImage segmentation discontinuities using edge detection Robert Operator

X Gradient Edges



Figure 7.8: To ImplementImage segmentation discontinuities using edge detection Robert Operator

Y Gradient Edges



Figure 7.9: To ImplementImage segmentation discontinuities using edge detection Robert Operator

Total Robert mask Edges of given image



Figure 7.10: To Implement Image segmentation discontinuities using edge detection Robert Operator

Scilab code Solution 7.3 To Implement Image segmentation discontinuities using edge detection prewitt operator

```

1 //Exp 7.3 Program Title:To Implement Image
  segmentation discontinuities using edge detection
  prewitt operator
2 ////environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
  plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
  in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');//read
  color image from directory
15 im2=rgb2gray(im1);
16 //prewitt mask
17 M1=[-1,-1,-1;0,0,0;1,1,1];//X-Gradientmask
18 M2=[-1,0,1;-1,0,1;-1,0,1];//Y-Gradientmask
19 im3=sum_of_product(im2,M1);//convolution
20 im4=sum_of_product(im2,M2);//convolution
21 im5=im3+im4;//totaledges
22 im5=uint8(im5);
23 figure(1);ShowColorImage(im1,'Original Color image')
  ;title('original color image','color','red','
  fontsize',6');//IPDtoolbox
24 figure(2);ShowImage(im2,' Grayscale image');title('
  Grayscale image','color','red','fontsize',6');//
  IPDtoolbox

```

Color image



Figure 7.11: To Implement Image segmentation discontinuties using edge detection prewitt operator

```
25 figure(3); ShowImage(im3, ' X Gradient prewitt Edges')  
    ;title('X Gradient prewitt Edges', 'color', 'red', '  
    fontsize', 6'); //IPDtoolbox  
26 figure(4); ShowImage(im4, 'Y Gradient prewitt Edges');  
    title('Y Gradient prewitt Edges', 'color', 'red', '  
    fontsize', 6'); //IPDtoolbox  
27 figure(5); ShowImage(im5, 'Total prewitt Edges'); title  
    ('Total prewitt Edges of given image', 'color', '  
    red', 'fontsize', 6'); //IPDtoolbox
```

Grayscale image



Figure 7.12: To Implement Image segmentation discontinuities using edge detection prewitt operator

X Gradient prewitt Edges



Figure 7.13: To Implement Image segmentation discontinuities using edge detection prewitt operator

check Appendix ?? for dependency:

`rose.BMP`

check Appendix [AP 8](#) for dependency:

`sum_of_product.sci`

Y Gradient prewitt Edges

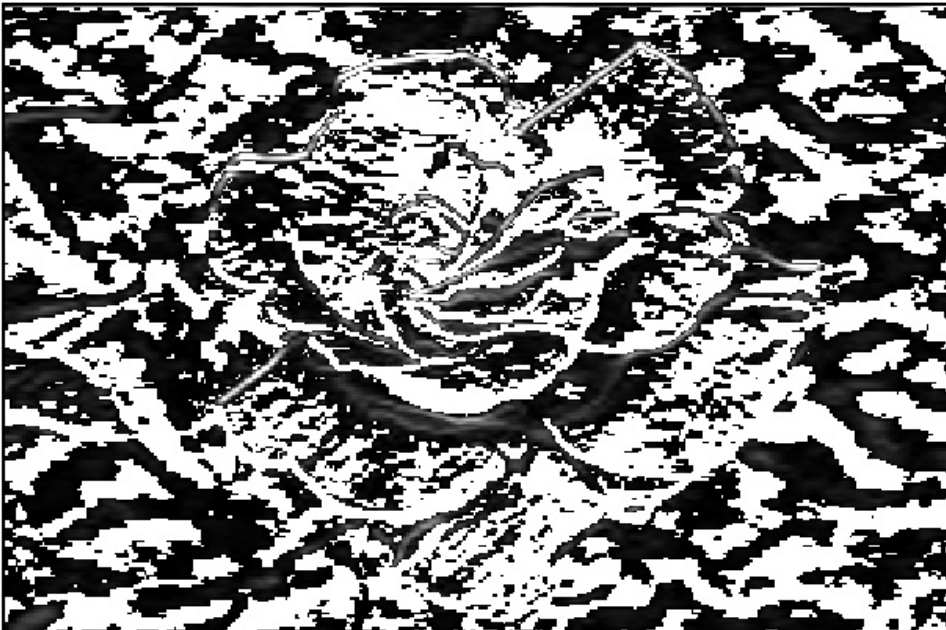


Figure 7.14: To Implement Image segmentation discontinuities using edge detection prewitt operator

Total prewitt Edges of given image



Figure 7.15: To Implement Image segmentation discontinuities using edge detection prewitt operator

Scilab code Solution 7.4 To Implement Image segmentation discontinuities using edge detection Sobel operator

```

1 //Exp 7.4 Program Title:To Implement Image
  segmentation discontinuities using edge detection
  Sobel operator
2 ////environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
  plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
  in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');//read
  color image from directory
15 im2=rgb2gray(im1);
16 //Sobel mask
17 M1=[-1 , -2, -1;0,0,0;1,2,1]; //X-Gradientmask
18 M2=[-1 ,0,1;-2,0,2;-1,0,1]; //Y-Gradientmask
19 im3=sum_of_product(im2,M1); //Convolution
20 im4=sum_of_product(im2,M2); //Convolution
21 im5=im3+im4; //Total Edges
22 im5=uint8(im5);
23 figure(1); ShowColorImage(im1, 'Original Color image')
  ;title('original color image', 'color', 'red', '
  fontsize',6'); //IPD tool box
24 figure(2); ShowImage(im2, ' Grayscale image');title('
  Grayscale image', 'color', 'red', 'fontsize',6'); //
  IPD tool box

```

Color image



Figure 7.16: To Implement Image segmentation discontinuties using edge detection Sobel operator

```
25 figure(3); ShowImage(im3, 'X Gradient Sobel Edges');  
    title('X Gradient Sobel Edges', 'color', 'red', '  
        fontsize', 6'); //IPD tool box  
26 figure(4); ShowImage(im4, 'Y Gradient Sobel Edges');  
    title('Y Gradient Sobel Edges', 'color', 'red', '  
        fontsize', 6'); //IPD tool box  
27 figure(5); ShowImage(im5, 'Total Sobel Edges'); title(''  
    Total Sobel Edges of given image', 'color', 'red', '  
    'fontsize', 6'); //IPD tool box
```

Grayscale image



Figure 7.17: To Implement Image segmentation discontinuities using edge detection Sobel operator

X Gradient Sobel Edges



Figure 7.18: To Implement Image segmentation discontinuities using edge detection Sobel operator

check Appendix ?? for dependency:

`rose.BMP`

check Appendix [AP 8](#) for dependency:

`sum_of_product.sci`

Y Gradient Sobel Edges

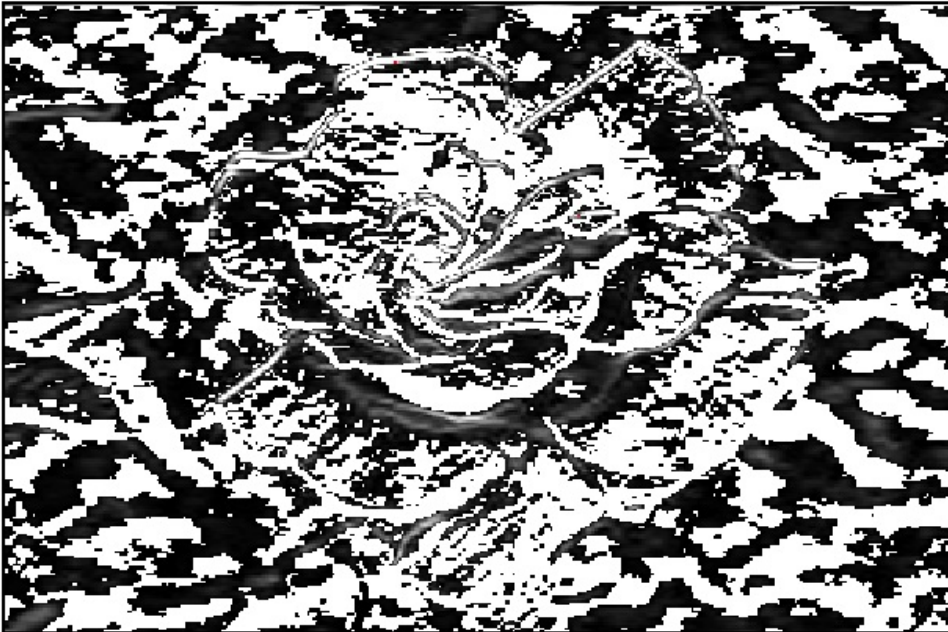


Figure 7.19: To Implement Image segmentation discontinuities using edge detection Sobel operator

Total Sobel Edges of given image



Figure 7.20: To Implement Image segmentation discontinuities using edge detection Sobel operator

Scilab code Solution 7.5 To Implement Image segmentation discontinuities using edge detection Laplacian operator

```

1 //Exp 7.5 Program Title:To Implement Image
  segmentation discontinuities using edge detection
  Laplacian operator
2 ////environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
  plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
  in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');//read
  color image from directory
15 im2=rgb2gray(im1);
16 //Laplacian mask
17 M1=[0 , -1, 0; -1, 4, -1; 0, -1, 0];
18 im3=sum_of_product(im2,M1);//convolution
19 im5=uint8(im3);
20 figure(1);ShowColorImage(im1, 'Original Color image')
  ;title('original color image', 'color', 'red', '
  fontsize', 6');//IPD tool box
21 figure(2);ShowImage(im2, ' Grayscale image');title('
  Grayscale image', 'color', 'red', 'fontsize', 6');//
  IPD tool box
22 figure(3);ShowImage(im3, ' Laplacian mask Edges');
  title('Laplacian mask Edges', 'color', 'red', '
  fontsize', 6');//IPD tool box

```

Color image



Figure 7.21: To Implement Image segmentation discontinuities using edge detection Laplacian operator

check Appendix ?? for dependency:

`rose.BMP`

check Appendix [AP 8](#) for dependency:

`sum_of_product.sci`

Grayscale image



Figure 7.22: To Implement Image segmentation discontinuities using edge detection Laplacian operator

Laplacian mask Edges

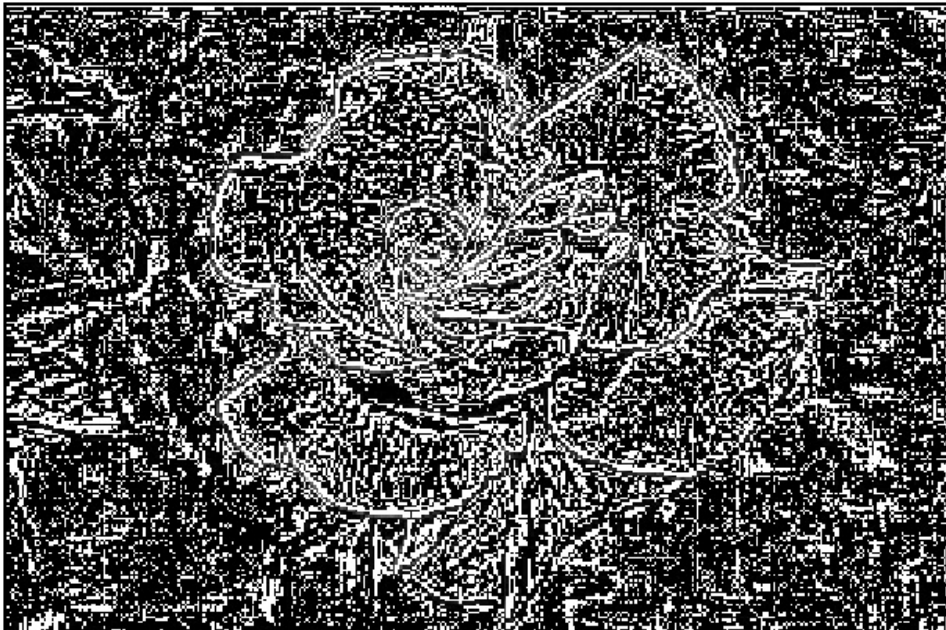


Figure 7.23: To Implement Image segmentation discontinuities using edge detection Laplacian operator

Scilab code Solution 7.6 To Implement Image segmentation based on discontinuities using edge detection LOG operator

```

1 //Exp 7.6 Program Title:To Implement Image
  segmentation discontinuities using edge detection
  LOG operator
2 ////environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
  plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
  in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');//read
  color image from directory
15 im2=rgb2gray(im1);
16 img3=double(im2);
17 [r,c]=size(img3);
18 w1=[0 0 -1 0 0;0 -1 -2 -1 0;-1 -2 16 -2 -1 ;0 -1 -2
  -1 0;0 0 -1 0 0];//LOGmask
19 for i=3:r-2
20   for j=3:c-2
21     im4(i,j)=[w1(1)*img3(i-2,j-2)+w1(2)*img3(i-2,
  j-1)+w1(3)*img3(i-2,j)+w1(4)*img3(i-2,j+1)
  +w1(5)*img3(i-2,j+2)+...
22     w1(6)*img3(i-1,j-2)+w1(7)*img3(i-1,j-1)+w1
  (8)*img3(i-1,j)+w1(9)*img3(i-1,j+1)+w1
  (10)*img3(i-1,j+2)+...
23     w1(11)*img3(i,j-2)+w1(12)*img3(i,j-1)+w1(13)

```

```

                *img3(i,j)+w1(14)*img3(i,j+1)+w1(15)*img3
                (i,j+2)+...
24         w1(16)*img3(i+1,j-2)+w1(17)*img3(i+1,j-1)+w1
                (18)*img3(i+1,j)+w1(19)*img3(i+1,j+1)+w1
                (20)*img3(i+1,j+2)+...
25         w1(21)*img3(i+2,j-2)+w1(22)*img3(i+2,j-1)+w1
                (23)*img3(i+2,j)+w1(24)*img3(i+2,j+1)+w1
                (25)*img3(i+2,j+2)];
26     end
27 end
28 figure(1);ShowColorImage(im1,'Original Color image')
    ;title('original color image','color','red','
    fontsize',6');//IPD tool box
29 figure(2);ShowImage(im2,' Grayscale image');title('
    Grayscale image','color','red','fontsize',6');//
    IPD tool box
30 figure(3);ShowImage(im4,' LOG operators Edges');
    title('LOG operators Edges','color','red','
    fontsize',6');//IPD tool box

```

check Appendix ?? for dependency:

rose.BMP

Color image



Figure 7.24: To Implement Image segmentation based on discontinuities using edge detection LOG operator

Grayscale image



Figure 7.25: To Implement Image segmentation based on discontinuities using edge detection LOG operator

LOG operators Edges

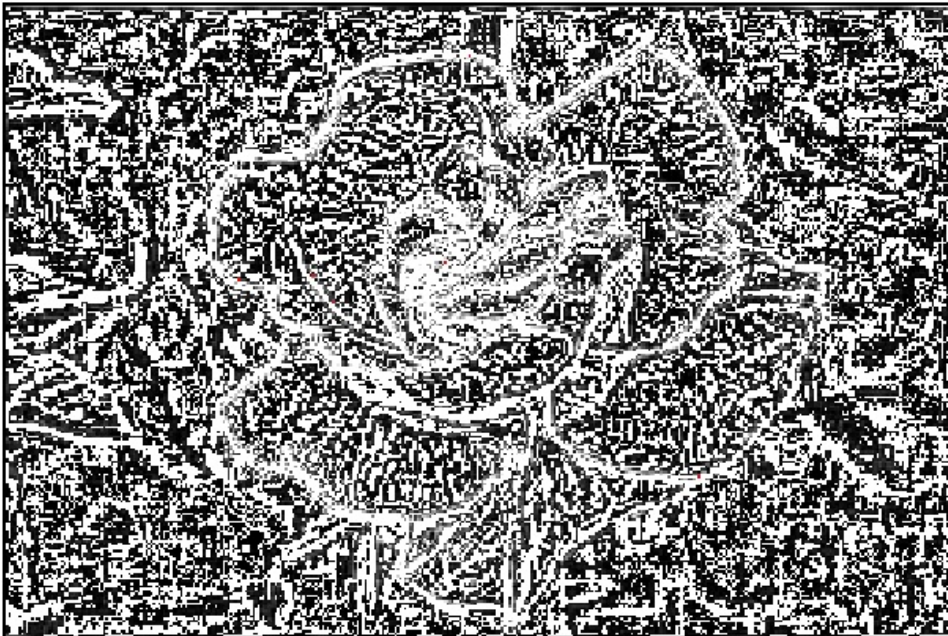


Figure 7.26: To Implement Image segmentation based on discontinuities using edge detection LOG operator

Experiment: 8

Write a program to perform image segmentation using local and global thresholding on a given image.

Scilab code Solution 8.1 To implement image segmentation based on similarities using Global thresholding

```
1 //Exp 8.1 Program Title:To implement image
  segmentation based on simalarities using Global
  thresholding //
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
```

```

11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
    plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');//read
    second color image from diractory
15 im2=rgb2gray(im1);// grey scale conversion
16 [im441,im442,im443]=imthresholding(im2);//global
    thresholding
17 figure(1);ShowColorImage(im1,'Color image');title('
    Color Image','color','red','fontsize',4');//
    displaying result
18 figure(2);ShowImage(im2,'Grayscale image');title('
    Grey scale image','color','red','fontsize',4');//
    displaying result
19 figure(3);ShowImage(im441,'Global Thersholding 1
    grayscale Image');title('Global Thersholding 1(
    mean) of Grey scale Image1','color','red','
    fontsize',4');// displaying result
20 figure(4);ShowImage(im442,'Global Thersholding 2(
    lowermean) grayscale Image');title('Global
    Thersholding 2(lowermean) grayscale Image','color
    ','red','fontsize',4');// displaying result
21 figure(5);ShowImage(im443,'Global Thersholding 3(
    uppermean) Gray scale Image Image');title('Global
    Thersholding 3(uppermean) Gray scale Image Image
    ','color','red','fontsize',4');// displaying
    result

```

Color Image



Figure 8.1: To implement image segmentation based on similarities using Global thresholding

Grey scale image



Figure 8.2: To implement image segmentation based on similarities using Global thresholding

Global Thersholding 1(mean) of Grey scale Image1



Figure 8.3: To implement image segmentation based on simalarities using Global threholding

Global Thersholding 2(lowermean) grayscale Image

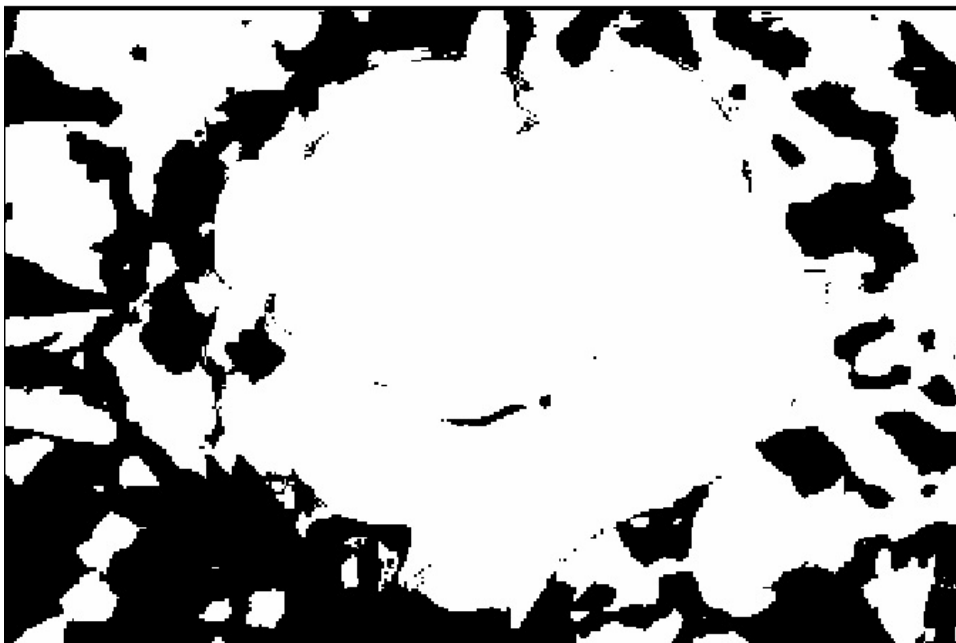


Figure 8.4: To implement image segmentation based on simalarities using Global threholding

Global Thersholding 3(uppermean) Gray scale Image Image



Figure 8.5: To implement image segmentation based on simalarities using Global thresholding

check Appendix [AP 7](#) for dependency:

`imthresholding.sci`

check Appendix ?? for dependency:

`rose.BMP`

check Appendix [AP 5](#) for dependency:

`uplmean.sci`

Scilab code Solution 8.2 To implement image segmentation based on similarities using local thresholding

```
1 //Exp 8.2 Program Title:To implement image
   segmentation based on simalarities using local
   thresholding /
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
13 SIVP_PATH = getSIVPpath(); //to locate a directory
   in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');//read
   color image from diractory
15 im2=rgb2gray(im1);// grey scale conversion
16 im2=imresize(im2,[256,256]);// resizeig of the image
17 im11=im2(1:128,1:128);// 128x128 first block
18 im12=im2(1:128,129:256);// 128x128 second block
19 im13=im2(129:256,1:128);// 128x128 third block
20 im14=im2(129:256,129:256);// 128x128 fourth block
21 [b11,b12,b13]=imthresholding(im11);//local
   thresholding on block 1
22 [b21,b22,b23]=imthresholding(im12);//local
   thresholding on block 2
23 [b31,b32,b33]=imthresholding(im13);//local
   thresholding on block 3
24 [b41,b42,b43]=imthresholding(im14);//local
   thresholding on block 4
```

```
25 im15=[b11,b21;b31,b41]; //Concnationation of all the
    blocks mean threshold
26 im16=[b12,b22;b32,b42]; //Concnationation of all the
    blocks lower mean threshold
27 im17=[b13,b23;b33,b43]; //Concnationation of all the
    blocks lower mean threshold
28 figure(1); ShowImage(im2, 'Grayscale image'); title('
    Grey scale image', 'color', 'red', 'fontsize', 4); //
    displaying result
29 figure(2); ShowImage(im15, 'Local thresholding mean of
    each block gray scale image'); title('Local
    thresholding mean of each block gray scale image',
    'color', 'red', 'fontsize', 4); // displaying
    result
30 figure(3); ShowImage(im16, 'Local thresholding
    lowermean of each block gray scale image'); title(
    'Local thresholding lowermean of each block gray
    scale image', 'color', 'red', 'fontsize', 4); //
    displaying result
31 figure(4); ShowImage(im17, 'Local thresholding
    uppermean of each block gray scale image'); title(
    'Local thresholding uppermean of each block gray
    scale image', 'color', 'red', 'fontsize', 4); //
    displaying result
```

check Appendix [AP 7](#) for dependency:

imthresholding.sci

check Appendix ?? for dependency:

rose.BMP

Grey scale image



Figure 8.6: To implement image segmentation based on similarities using local thresholding

Local thresholding mean of each block gray scale image



Figure 8.7: To implement image segmentation based on similarities using local thresholding

Local thresholding lowermean of each block gray scale image



Figure 8.8: To implement image segmentation based on similarities using local thresholding

Local thresholding uppermean of each block gray scale image



Figure 8.9: To implement image segmentation based on similarities using local thresholding

check Appendix [AP 5](#) for dependency:

`uplmean.sci`

Experiment: 9

Write a program to perform Block Truncation coding on a given image.

Scilab code Solution 9.0 To implement Block truncation coding on grayscale image and color image

```
1 //Exp 9 Program Title:To implement Block truncation
   coding on grayscale image and color image
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional, 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
```

```

13 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed
14 im1=imread(SIVP_PATH + 'images/rose.bmp');//read
    second color image from directory
15 im2=rgb2gray(im1);
16 im11=imresize(im1,[256,256]);// image resize
17 imd=double(im11);
18 im2=double(im2);
19 [red,green,blue]=colorplanes(imd);//Seperation of
    color planes
20 [gr,lmrgr,umrgr]=uplmean(im2);//BTC for
    graysacleimage
21 [r1,lmrr,umrr]=uplmean(red);//BTC for red plane
22 [g1,lmrg,umrg]=uplmean(green);//BTC for green plane
23 [b1,lmrb,umrb]=uplmean(blue);//BTC for blue plane
24 btcim=cat(3,r1,g1,b1);// formation of BTC image
25 btcim=uint8(btcim);
26 figure(1);ShowColorImage(im1,'Color image image');
    title('Color image','color','red','fontsize',6);
27 figure(2);ShowImage(gr,'Truncated grayscale image in
    upper mean and upper mean color values');title('
    Truncated grayscale image in upper mean and upper
    mean color values','color','red','fontsize',3);
28 figure(3);ShowColorImage(btcim,'Truncated image in
    upper mean and upper mean color values');title('
    Truncated image in upper mean and upper mean
    color values','color','red','fontsize',3);

```

check Appendix [AP 6](#) for dependency:

colorplanes.sci

check Appendix ?? for dependency:

Color image



Figure 9.1: To implement Block truncation coding on grayscale image and color image

Truncated grayscale image in upper mean and upper mean color values



Figure 9.2: To implement Block truncation coding on grayscale image and color image

Truncated image in upper mean and upper mean color values



Figure 9.3: To implement Block truncation coding on grayscale image and color image

rose.BMP

check Appendix [AP 5](#) for dependency:

uplmean.sci

Experiment: 10

Write a program to implement morphological dilation and erosion operation on an image .

Scilab code Solution 10.1 To implement Dilation of binary image and grayscale image edges

```
1 //Exp 10.1 Program Title:To implement Dilation of
   binary image and grayscale image edges
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
   book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
   Theckedath
10 clc //to clear command window.
11 clear all //to disable previously defined variables.
12 xdel(winsid());///clear all the previous figures
   plots
```

```

13 // Creation of binary image using folloing program
14 im1=zeros(100,100);// Create zeros matrix for image
    formation
15 im2=zeros(100,100);// Create zeros matrix for image
    formation
16 im3=zeros(100,100);// Create zeros matrix for image
    formation
17 im22=zeros(100,100);// Create zeros matrix for image
    formation
18 im1(30:80,30:80)=1;// Make a square object in im1
    matrix
19 im2(2:30,82:90)=1;// Make a vertical rectangale in
    im2 matrix
20 im3(30:30,81:81)=1;// Create a dot which connect
    both above objects
21 SIVP_PATH = getSIVPpath();//to locate a directory
    in which SIVP toolbox is installed
22 im6=imread(SIVP_PATH + 'images/rose.bmp');//read
    second color image from directory
23 im7=rgb2gray(im6);
24 im8=edge(im7,'canny');// Edge detection using canny
    edge detection
25 im4=im1+im2+im3;//Addition of im1 and im2 for
    creation of two objects
26 im10=dial(im8);// dialtion of edges
27 im5=dial(im4);// dialtion of edges
28 figure(1);ShowImage(im4,'Binary Image');title('
    Binary Image','color','red','fontsize',4');//IPD
    Toolbox
29 figure(2);ShowImage(im7,' Grey scale image');title('
    Gray scale image','color','red','fontsize',4');//
    IPD Toolbox
30 figure(3);ShowImage(im8,'Edges of the grey scale
    image');title('Edges of the gray scale image','
    color','red','fontsize',4');
31 figure(4);ShowImage(im10,'Dialted Edges');title('
    Dilated Edges','color','red','fontsize',4');//IPD
    Toolbox

```

```
32 figure(5); ShowImage(im5, 'Dialted Binary image');
    title('Dialted Binary image', 'color', 'red', '
    fontsize', 4'); //IPD Toolbox
```

check Appendix [AP 4](#) for dependency:

dial.sci

check Appendix ?? for dependency:

rose.BMP

Scilab code Solution 10.2 To implement Erosion of binary image

```
1 //Exp 10.2 Program Title:To implement Erosion of
    binary image
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional, 64 bit
6 //Reference books names : 1) Digital Image Processing
    book author :
7 // Rafael C. Gonzalez and Richard E. Woods,
8 //2) Digital Image Processing book author :S. Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
    Theckedath
10 clc //to clear command window.
```

Binary Image

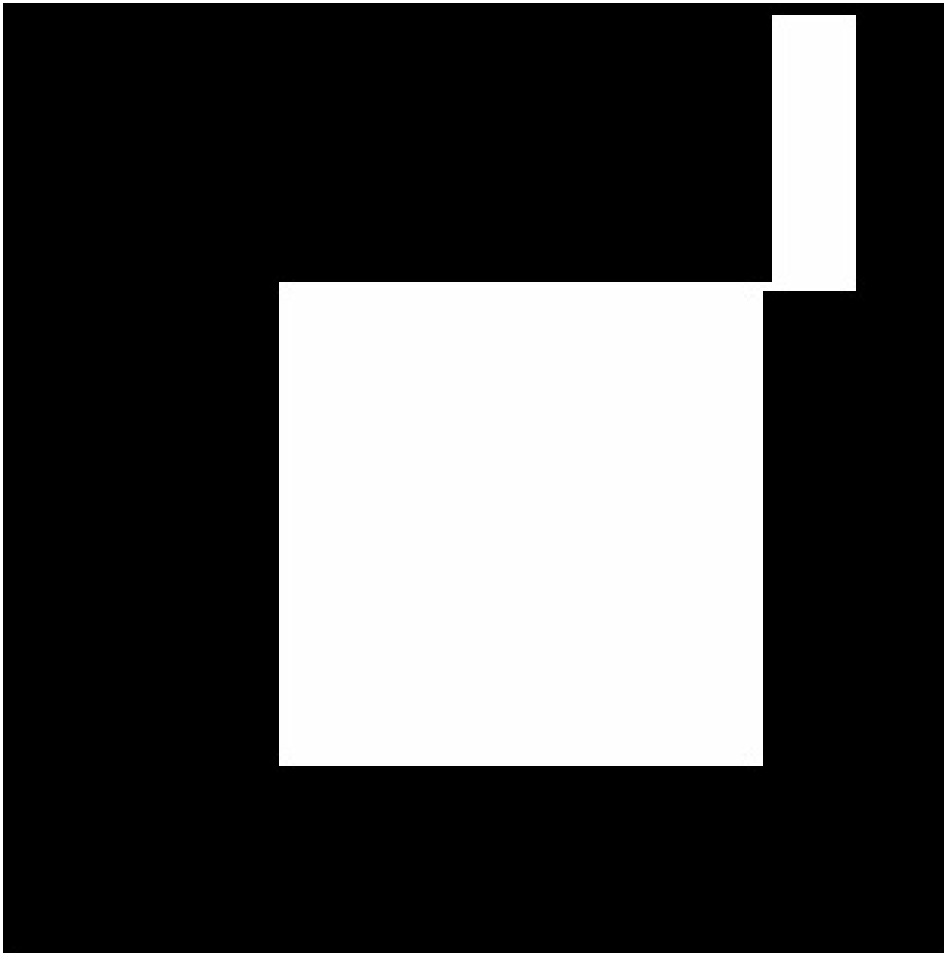


Figure 10.1: To implement Dilation of binary image and grayscale image edges

Gray scale image



Figure 10.2: To implement Dilation of binary image and grayscale image edges

Edges of the gray scale image



Figure 10.3: To implement Dilation of binary image and grayscale image edges

Dilated Edges



Figure 10.4: To implement Dilation of binary image and grayscale image edges

Dialted Binary image

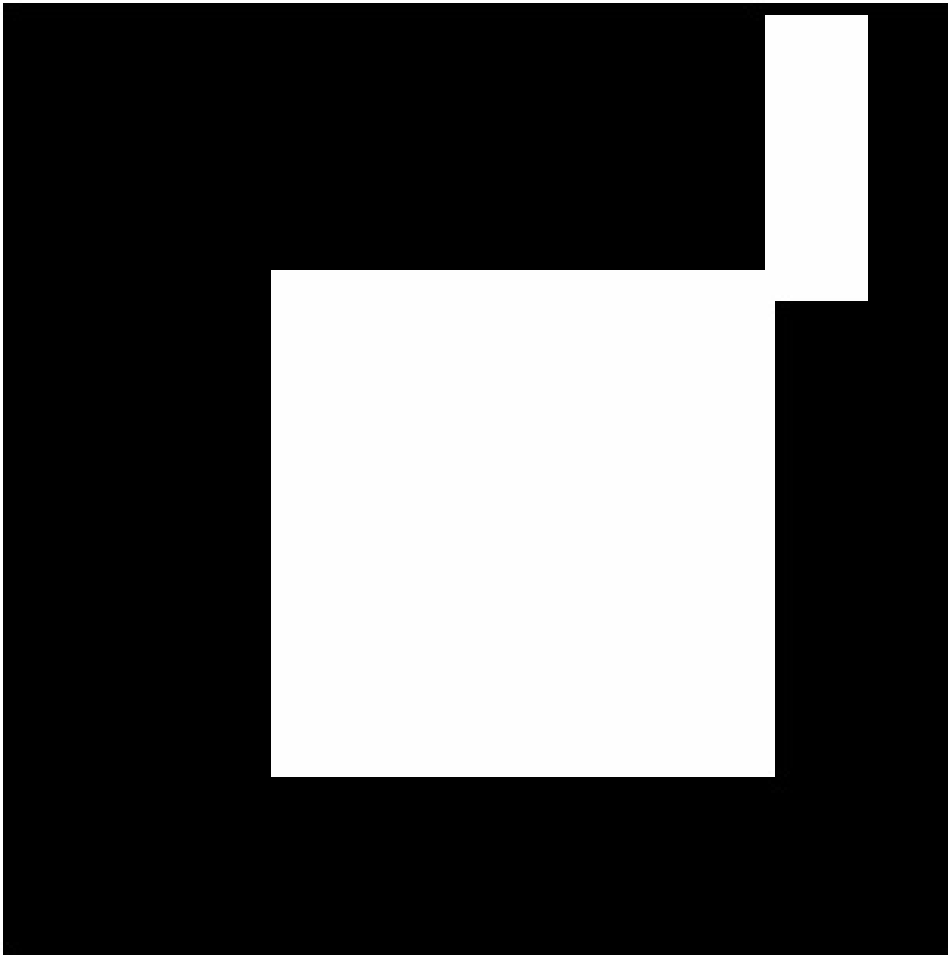


Figure 10.5: To implement Dilation of binary image and grayscale image edges

```

11 clear all;
12 xdel(winsid());
13 // Creation of binary image using folloing program
14 im1=zeros(100,100);// Create zeros matrix for image
    formation
15 im2=zeros(100,100);// Create zeros matrix for image
    formation
16 im3=zeros(100,100);// Create zeros matrix for image
    formation
17 im33=zeros(100,100);// Create zeros matrix for image
    formation
18 im1(30:80,30:80)=1;// Make a square object in im1
    matrix
19 im2(2:30,82:90)=1;// Make a vertical rectangale in
    im2 matrix
20 im3(30:30,81:81)=1;// Create a dot which connect
    both above objects
21 im33(2:30,18:28)=1;// Create a objects
22 i5=im1+im2+im3+im33;//Addition of im1 im2 im3 and
    im33 for creation of three objects
23 [im6]=erod(i5);
24 figure(1);ShowImage(i5,' Binary Image');title('
    Binary Image','color','red','fontsize',4');//IPD
    TOOLBOX
25 figure(2);ShowImage(im6,'Binary Image after erosion'
    );title('Binary Image After erosion','color','red
    ','fontsize',4');//IPD TOOLBOX

```

check Appendix [AP 3](#) for dependency:

erod.sci

Binary Image

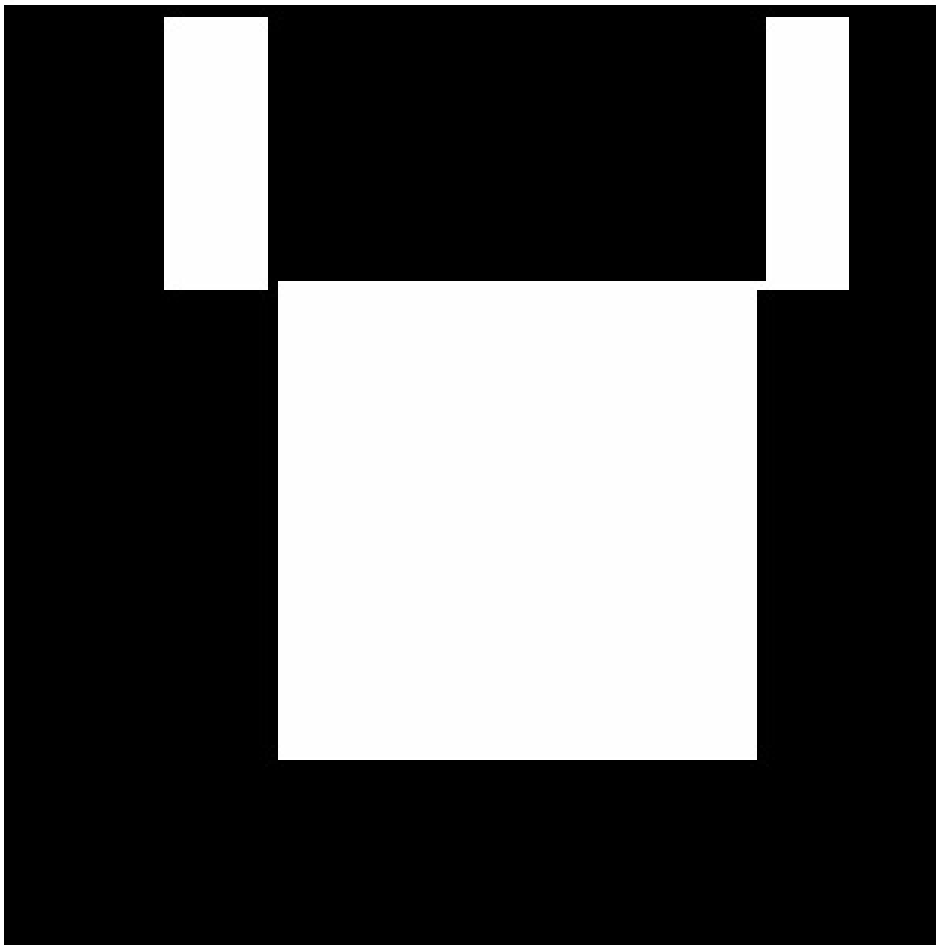


Figure 10.6: To implement Erosion of binary image

Binary Image After erosion

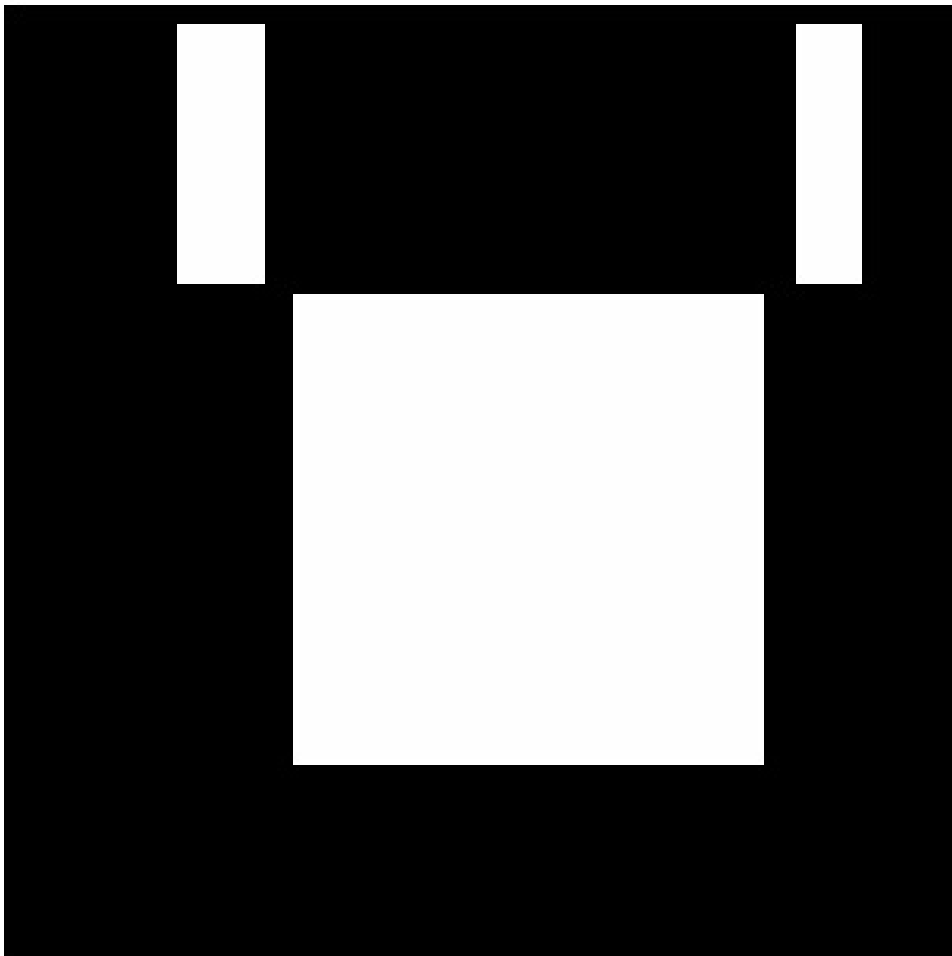


Figure 10.7: To implement Erosion of binary image

Experiment: 11

Write a program to implement morphological closing and opening operation on a given image.

Scilab code Solution 11.1 To perform morphological opening operation on binary image

```
1 //Exp 11.1 Program Title:To perform morphological
  opening operation on binary image
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
11 clear all;
```

```

12 xdel(winsid());
13 // Creation of binary image using folloing program
14 im1=zeros(100,100);// Create zeros matrix for image
    formation
15 im2=zeros(100,100);// Create zeros matrix for image
    formation
16 im3=zeros(100,100);// Create zeros matrix for image
    formation
17 im33=zeros(100,100);// Create zeros matrix for image
    formation
18 im1(30:80,30:80)=1;// Make a square object in im1
    matrix
19 im2(2:30,82:90)=1;// Make a vertical rectangale in
    im2 matrix
20 im3(30:30,81:81)=1;// Create a dot which connect
    both above objects
21 im33(2:30,18:28)=1;// Create a objects
22 i4=im1+im2+im33+im3;//Addition of im1,im33 and im2 ,
    im3for creation of three objects
23 im10=erod(i4);// Erosion of edges
24 im11=dial(im10);// dialtion of edges
25 figure(1);ShowImage(i4, 'Binary Image');title('Binary
    Image', 'color', 'red', 'fontsize',4');
26 figure(2);ShowImage(im10, 'Erodeded Image');title('
    Eroded Image', 'color', 'red', 'fontsize',4');
27 figure(3);ShowImage(im11, 'Opening the object');title
    ('opening the object', 'color', 'red', 'fontsize'
    ,4');

```

check Appendix [AP 4](#) for dependency:

`dial.sci`

Binary Image

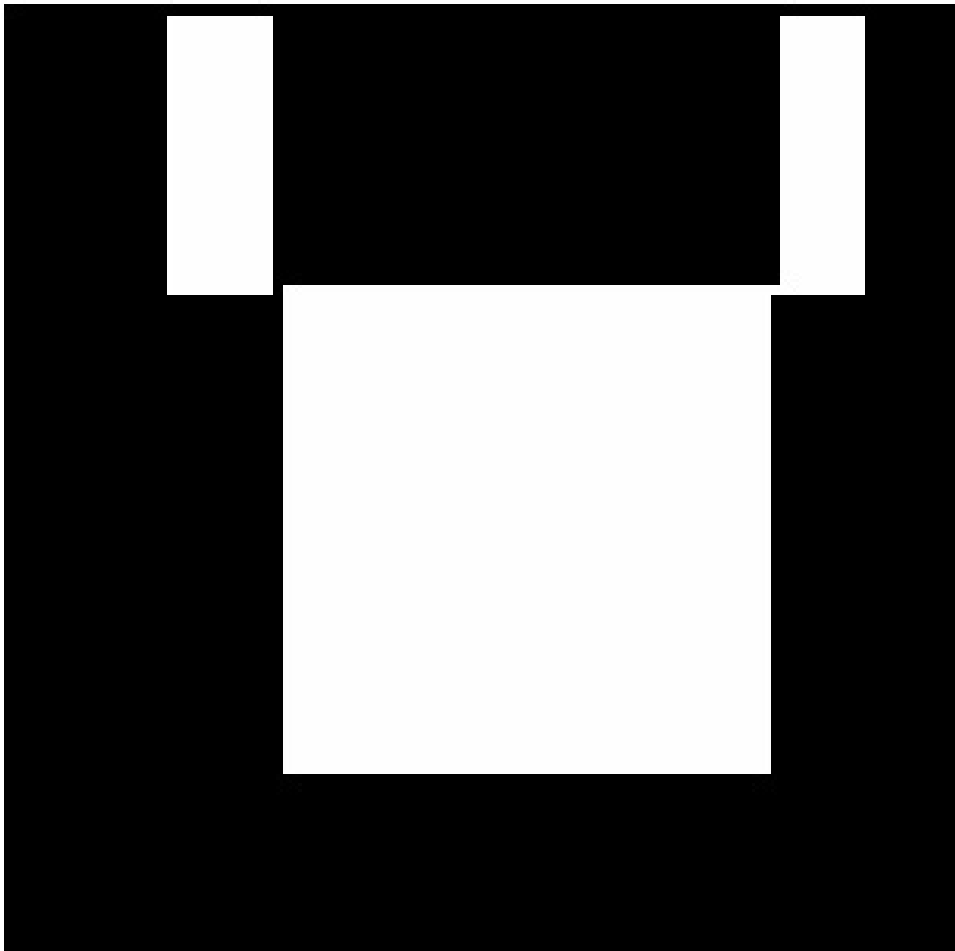


Figure 11.1: To perform morphological opening operation on binary image

Eroded Image

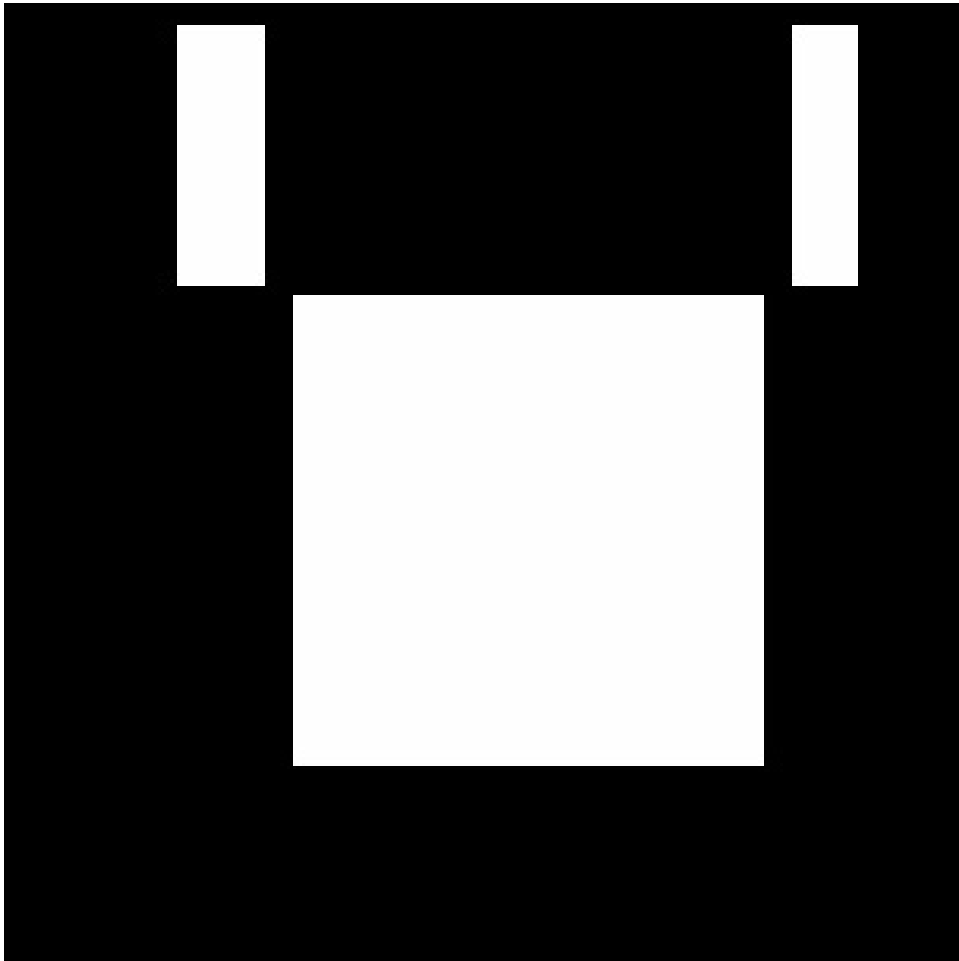


Figure 11.2: To perform morphological opening operation on binary image

opening the object

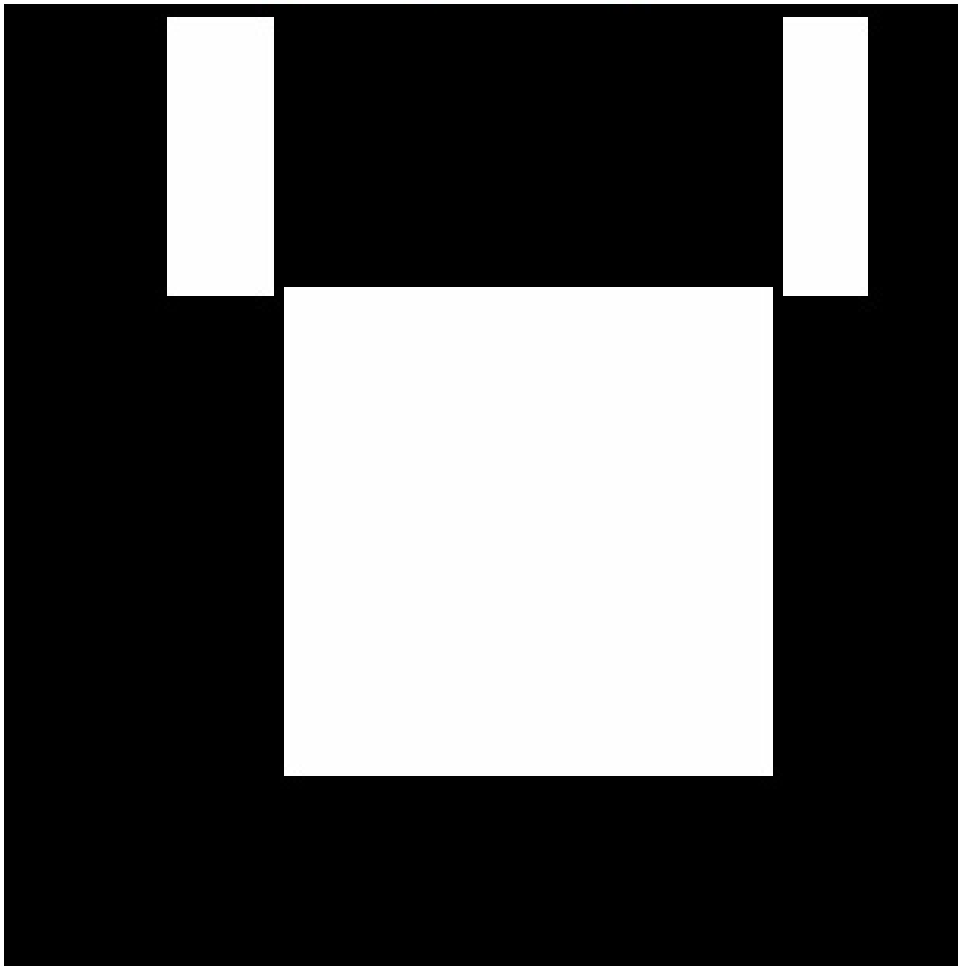


Figure 11.3: To perform morphological opening operation on binary image

check Appendix [AP 3](#) for dependency:

`erod.sci`

Scilab code Solution 11.2 To perform morphological Closing operation on binary image

```
1 //Exp 11.2 Program Title:To perform morphological
  Closing operation on binary image
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1) Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2) Digital Image Processing book author :S.Sudhir ,
9 //3) Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
11 clear all;
12 xdel(winsid());
13 // Creation of binary image using folloing program
14 im1=zeros(100,100); // Create zeros matrix for image
  formation
15 im2=zeros(100,100); // Create zeros matrix for image
  formation
16 im3=zeros(100,100); // Create zeros matrix for image
  formation
17 im33=zeros(100,100); // Create zeros matrix for image
  formation
18 im1(30:80,30:80)=1; // Make a square object in im1
  matrix
19 im2(2:30,82:90)=1; // Make a vertical rectangale in
  im2 matrix
```

```
20 im3(30:30,81:81)=1; // Create a dot which connect
    both above objects
21 im33(2:30,18:28)=1; // Create a objects
22 i4=im1+im2+im33; // Addition of im1,im33 and im2 for
    creation of three objects
23 im10=dial(i4); // Erosion of edges
24 im11=erod(im10); // dialtion of edges
25 figure(1); ShowImage(i4, 'Binary Image'); title('Binary
    Image', 'color', 'red', 'fontsize', 4');
26 figure(2); ShowImage(im10, 'Dilated Image'); title('
    Dilated Image', 'color', 'red', 'fontsize', 4');
27 figure(3); ShowImage(im11, 'Closing the object'); title
    ('Closing the object', 'color', 'red', 'fontsize',
    ,4');
```

check Appendix [AP 4](#) for dependency:

dial.sci

check Appendix [AP 3](#) for dependency:

erod.sci

Binary Image

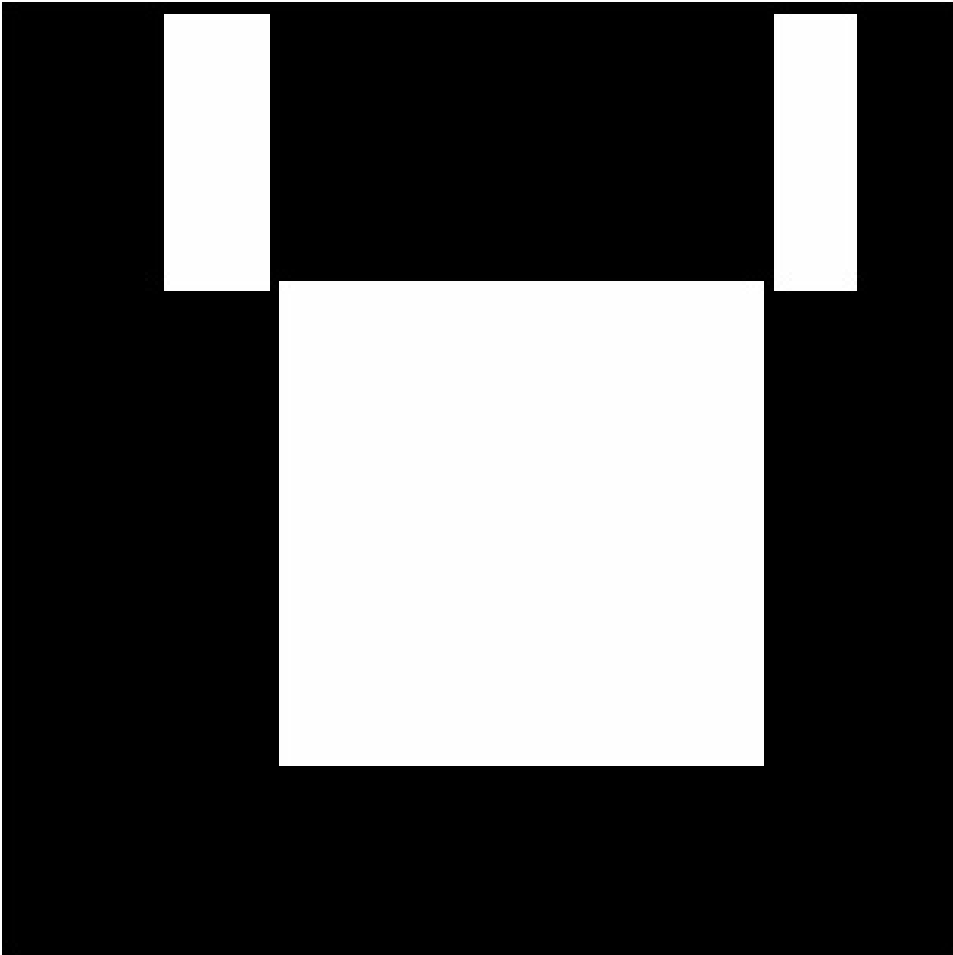


Figure 11.4: To perform morphological Closing operation on binary image

Dilated Image

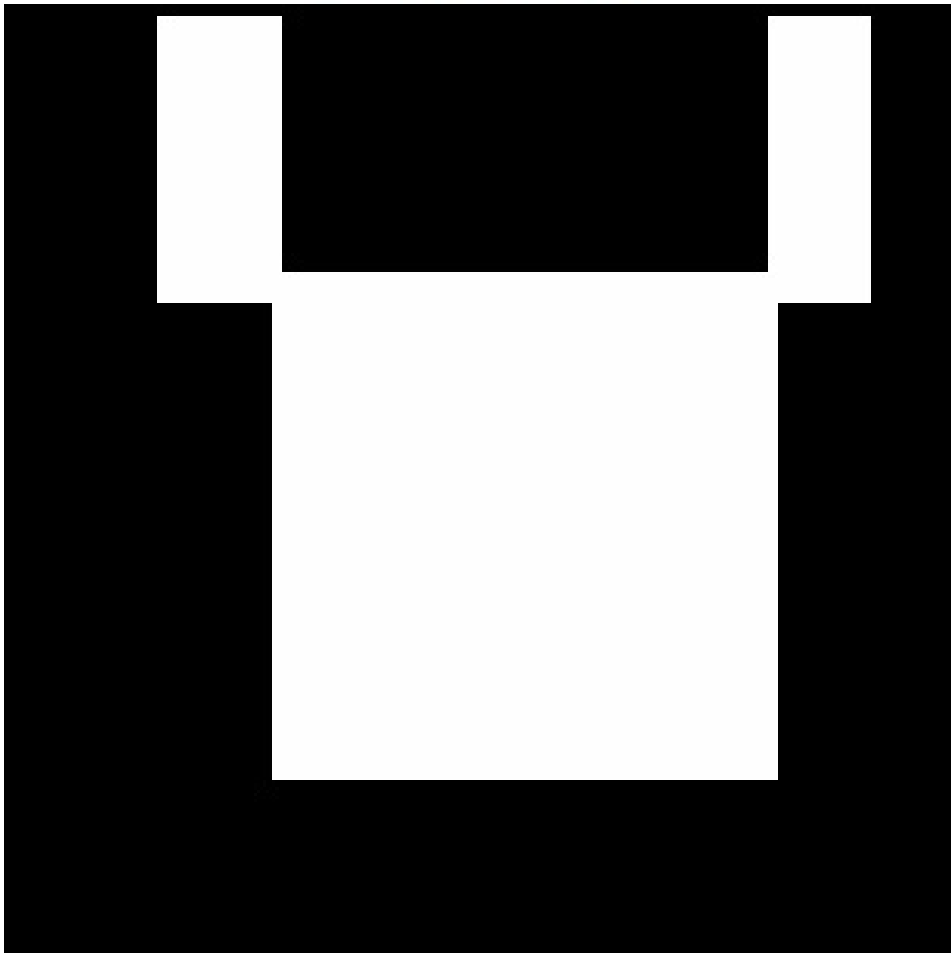


Figure 11.5: To perform morphological Closing operation on binary image

Closing the object

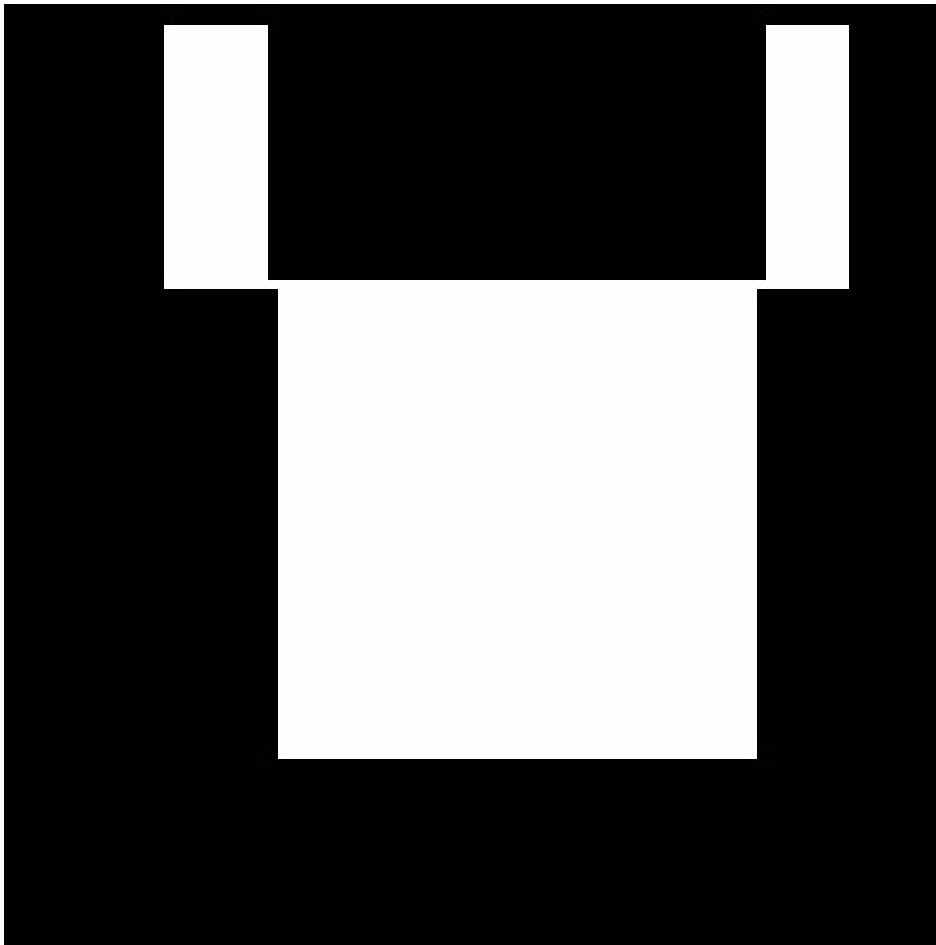


Figure 11.6: To perform morphological Closing operation on binary image

Experiment: 12

Write a program to implement boundary extraction using morphological operation on an image.

Scilab code Solution 12.0 To implement Boundary Extraction using morphological operation of binary image

```
1 //Exp 12 Program Title:To implement Boundary
  Extraction using morphological operation of
  binary image
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Tested on OS: Windows 7 Professional , 64 bit
6 //Reference books names : 1)Digital Image Processing
  book author :
7 // Rafael C.Gonzalez and Richard E.Woods,
8 //2)Digital Image Processing book author :S.Sudhir ,
9 //3)Digital Image Processing book author :Dhanajay
  Theckedath
10 clc //to clear command window.
```

```

11 clear all;
12 xdel(winsid());
13 // Creation of binary image using folloing program
14 im1=zeros(100,100);// Create zeros matrix for image
    formation
15 im2=zeros(100,100);// Create zeros matrix for image
    formation
16 im3=zeros(100,100);// Create zeros matrix for image
    formation
17 im1(30:80,30:80)=1;// Make a square object in im1
    matrix
18 im2(2:30,82:90)=1;// Make a vertical rectangle in
    im2 matrix
19 im3(30:30,81:81)=1;// Create a dot which connect
    both above objects
20 im4=im1+im2;//Addition of im1 and im2 for creation
    of two objects
21 im8=erod(im4);//Erosion function
22 im10=im4-im8;//boundary extraction using image-
    eroded image
23 im11=dial(im4);// dilation
24 im12=im11-im4;// boundary extration using dilated
    image - image
25 figure(1),ShowImage(im4,"binary image"),title("
    Binary Image",'color','red','fontsize',4');//
    Show binary image whose boundary is extracted
26 figure(2),ShowImage(im8,"Erosion of image"),title("
    Image After Erosion",'color','red','fontsize',4')
    ;// Show the eroded image
27 figure(3),ShowImage(im10,"Boundary of Objects in the
    Binary Image"),title("Boundary of Objects in the
    Binary Image (Image - Eroded image)",'color','
    red','fontsize',4');// Boundary of objects
28 figure(4),ShowImage(im11,"Dilation of Binary Image")
    ,title("Dilation of Binary Image",'color','red','
    fontsize',4');// Dilation of image
29 figure(5),ShowImage(im12,"Boundary of Objects in the
    Binary Image"),title("Boundary of Objects in the

```

```
Binary Image (Dilated image - Image)", 'color', 'red', 'fontsize', 4'); // Boundary of objects using dilated image - original image
```

check Appendix [AP 4](#) for dependency:

```
dial.sci
```

check Appendix [AP 3](#) for dependency:

```
erod.sci
```

Binary Image

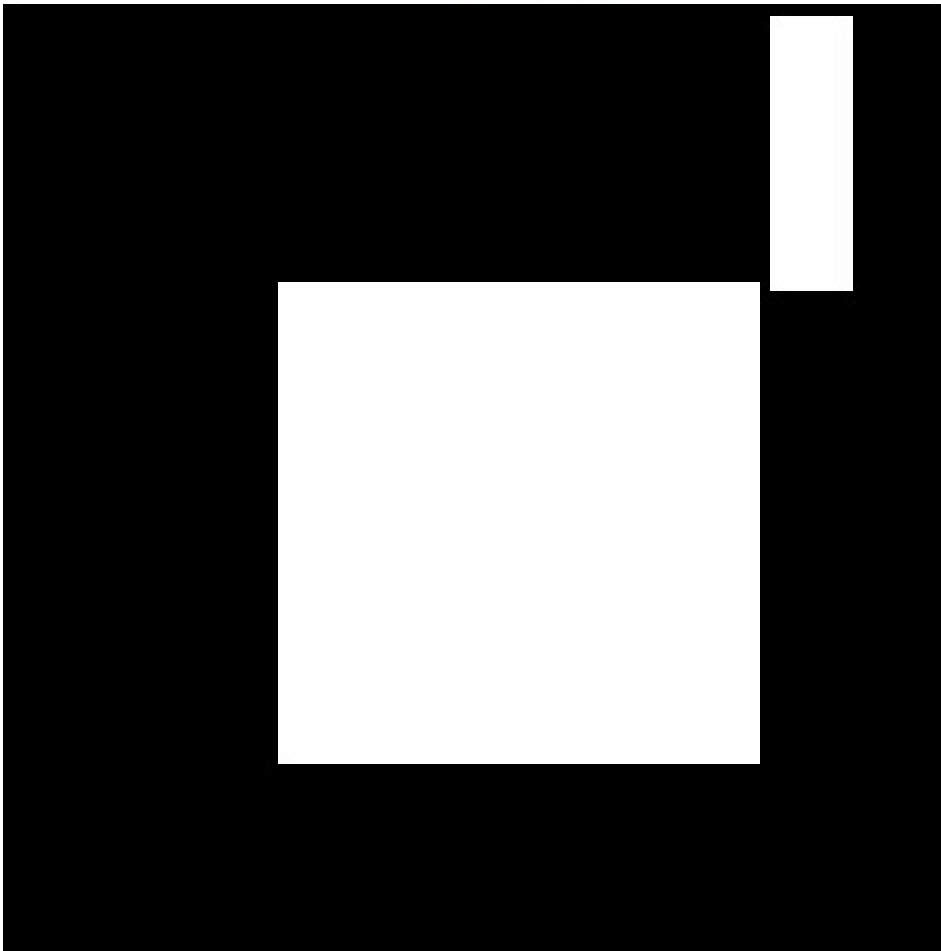


Figure 12.1: To implement Boundary Extraction using morphological operation of binary image

Image After Erosion

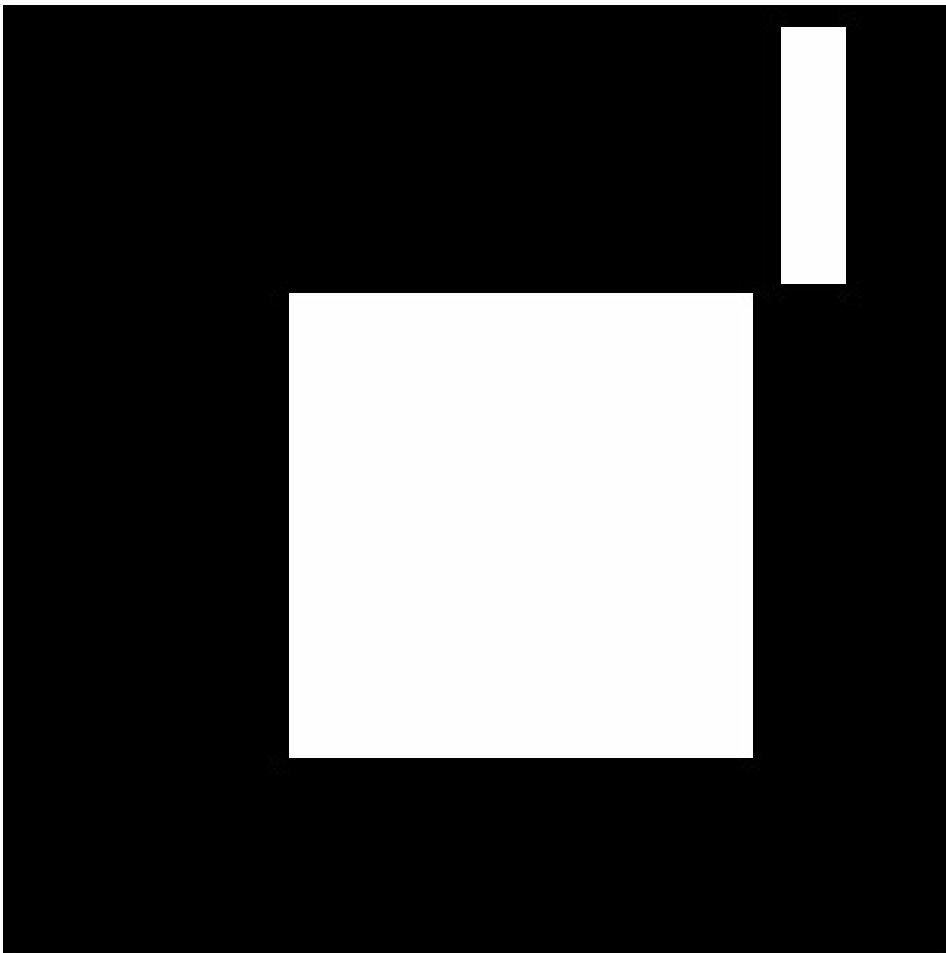


Figure 12.2: To implement Boundary Extraction using morphological operation of binary image

Boundary of Objects in the Binary Image (Image - Eroded image)

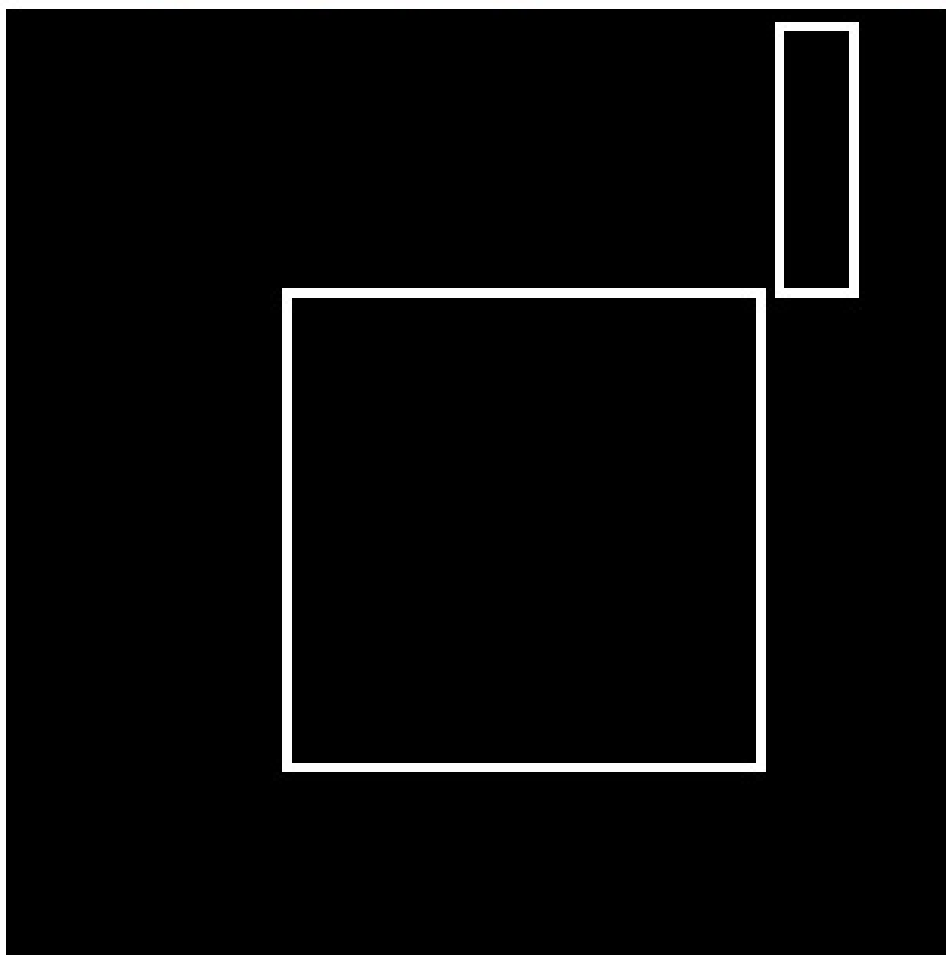


Figure 12.3: To implement Boundary Extraction using morphological operation of binary image

Dilation of Binary Image

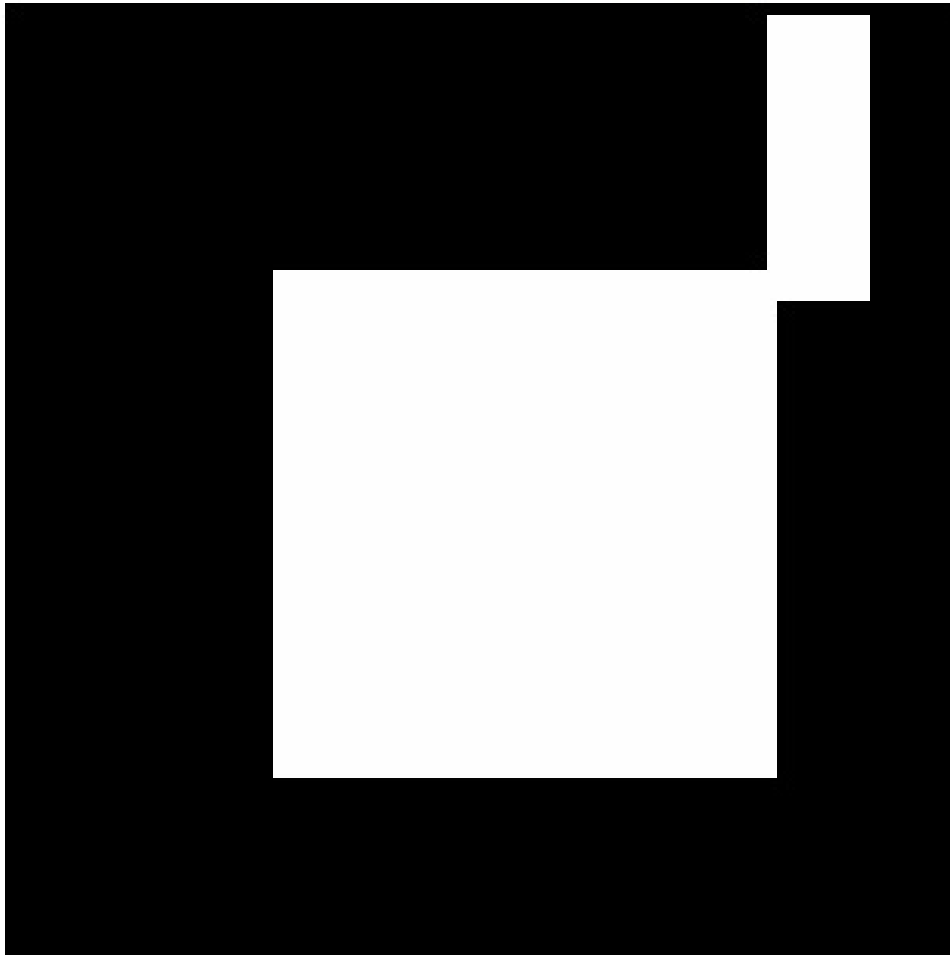


Figure 12.4: To implement Boundary Extraction using morphological operation of binary image

Boundary of Objects in the Binary Image (Dilated image - Image)

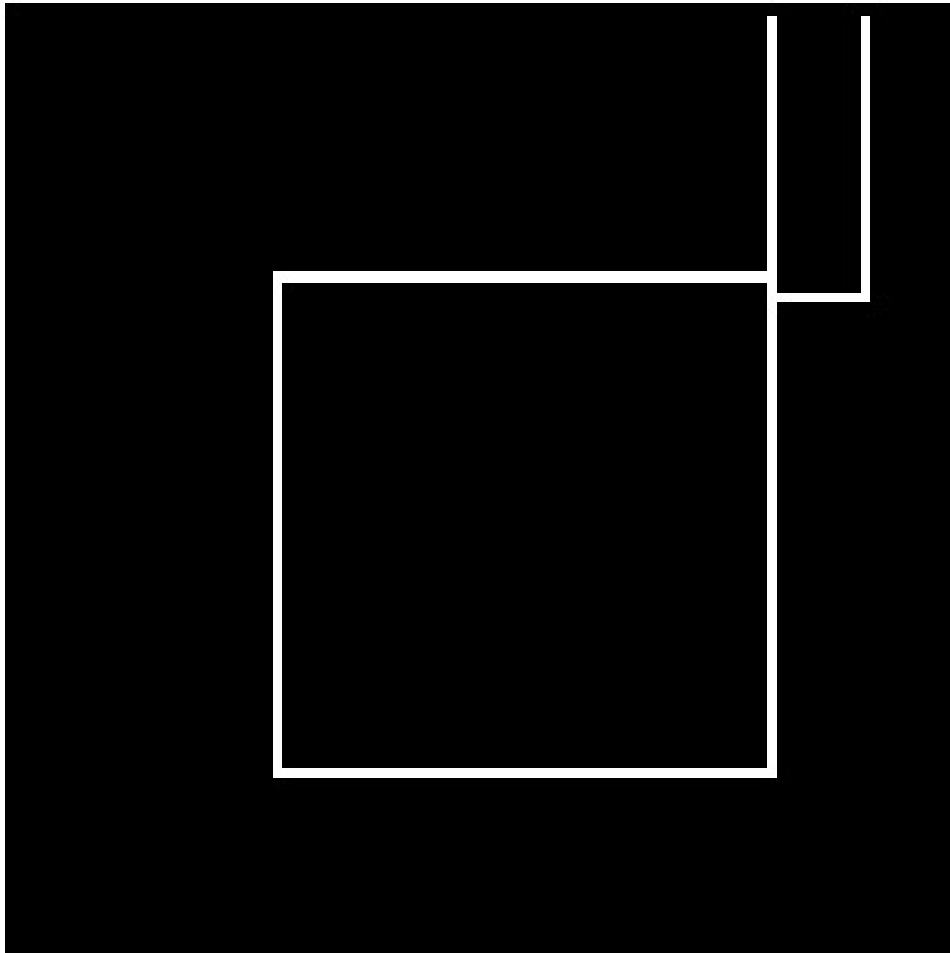


Figure 12.5: To implement Boundary Extraction using morphological operation of binary image

Experiment: 13

Write a program to perform Truncated DCT and IDCT on a given image.

Scilab code Solution 13.0 To Perform Truncated DCT and IDCT on a given Image

```
1 //Exp 13 Program Title:To Perform Truncated DCT and
   IDCT on a given Image
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
   //
   Tested on OS: Windows 7 Professional , 64 bit
5 //Reference books names : 1)Digital Image Processing
   book author :
6 // Rafael C.Gonzalez and Richard E.Woods,
7 //2)Digital Image Processing book author :S.Sudhir ,
8 //3)Digital Image Processing book author :Dhanajay
   Theckedath
9 clc //to clear command window.
10 clear all; //to disable previously defined variables
   .
```

```

11 xdel(winsid());///clear all the previous figures
    plots
12 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed
13 im1=imread(SIVP_PATH + 'images/rose.bmp');///
    readcolor image from diractory
14 im2=rgb2gray(im1);
15 [imm1,imm2,imm3]=im_dct(im2);//DCT and IDCT
16 [r1,r2,r3]=im_dct(im2);//DCT and IDCT
17 [g1,g2,g3]=im_dct(im2);//DCT and IDCT
18 [b1,b2,b3]=im_dct(im2);//DCT and IDCT
19 figure(1),ShowImage(r1,'Red plane Truncated DCT
    coeffi.IDCT at first level');title('Red plane
    Truncated DCT coeffi IDCT at first level','color'
    ,'red','fontsize',3');//IPDtoolbox
20 figure(2),ShowImage(r2,'Red plane Truncated DCT
    coeffi.IDCT at second level');title('Red plane
    Truncated DCT coeffi.IDCT at second level','color
    ','red','fontsize',3');//IPDtoolbox
21 figure(3),ShowImage(g1,'Green plane IDCT at first
    level');title('Green plane Truncated DCT coeffi.
    IDCT at first level','color','red','fontsize',3')
    ;//IPDtoolbox
22 figure(4),ShowImage(g2,'Green plane IDCT at second
    level');title('Green plane Truncated DCT coeffi.
    IDCT at second level','color','red','fontsize'
    ,3');//IPDtoolbox
23 figure(5),ShowImage(b1,'Blue plane IDCT at first
    level');title('Blue plane Truncated DCT coeffi.
    IDCT at first level','color','red','fontsize',3')
    ;//IPDtoolbox
24 figure(6),ShowImage(b2,'Blue plane IDCT at second
    level');title('Blue plane Truncated DCT coeffi.
    IDCT at second level','color','red','fontsize'
    ,3');//IPDtoolbox
25 figure(7),ShowImage(imm1,'Grey scale image IDCT at
    first level');title('Grey scale image Truncated
    DCT coeffi.IDCT at first level','color','red','

```

Red plane Truncated DCT coeffi IDCT at first level



Figure 13.1: To Perform Truncated DCT and IDCT on a given Image

```
    fontsize',3'); //IPDtoolbox  
26 figure(8), ShowImage(imm2, 'Grey scale image Truncated  
    DCT coeffi.IDCT at second level'); title('Grey  
    scale image Truncated DCT coeffi.IDCT at second  
    level', 'color', 'red', 'fontsize',3'); //IPDtoolbox
```

Red plane Truncated DCT coeffi.IDCT at second level



Figure 13.2: To Perform Truncated DCT and IDCT on a given Image

Green plane Truncated DCT coeff.IDCT at first level



Figure 13.3: To Perform Truncated DCT and IDCT on a given Image

Green plane Truncated DCT coeffi.IDCT at second level



Figure 13.4: To Perform Truncated DCT and IDCT on a given Image

Blue plane Truncated DCT coeff.IDCT at first level



Figure 13.5: To Perform Truncated DCT and IDCT on a given Image

Blue plane Truncated DCT coeffi.IDCT at second level



Figure 13.6: To Perform Truncated DCT and IDCT on a given Image

check Appendix [AP 2](#) for dependency:

`im_dct.sci`

check Appendix ?? for dependency:

`rose.BMP`

Grey scale image Truncated DCT coeffi.IDCT at first level



Figure 13.7: To Perform Truncated DCT and IDCT on a given Image

Grey scale image Truncated DCT coeffi.IDCT at second level



Figure 13.8: To Perform Truncated DCT and IDCT on a given Image

Experiment: 14

Write a program to perform image Wavelet transform on given image.

Scilab code Solution 14.0 To Perform Wavelet Transform on grayscale image

```
1 //Exp 14 Program Title:To Perform Wavelet Transform
   on grayscale image
2 //environment: Scilab 5.5.2
3 //Toolbox: Image Processing Design 8.3.1-1
4 //Toolbox: SIVP 0.5.3.2
5 //Toolbox:wavlab 1.0.1
6 //Tested on OS: Windows 7 Professional , 64 bit
7 //Reference books names : 1)Digital Image Processing
   book author :
8 // Rafael C.Gonzalez and Richard E.Woods,
9 //2)Digital Image Processing book author :S.Sudhir ,
10 //3)Digital Image Processing book author :Dhanajay
   Theckedath
11 clc //to clear command window.
12 clear all; //to disable previously defined variables
```

```

13 xdel(winsid());///clear all the previous figures
    plots
14 SIVP_PATH = getSIVPpath(); //to locate a directory
    in which SIVP toolbox is installed
15 ii1=imread(SIVP_PATH + 'images/rose.bmp');///read
    color image from diractory
16 r= rgb2gray(ii1);
17 im=double(r);
18 [a,b,c,d]=dwt2(im, 'db2');//WavelabToolbox
19 [a1,b1,c1,d1]=dwt2(a, 'db2');//WavelabToolbox
20 figure(1);ShowImage(im, 'Grayscale image');title('
    Grayscale image', 'color', 'red', 'fontsize', 4);//
    IPDtoolbox
21 figure(2);ShowImage(a, 'LL Component at 1st Level');
    title('LL Component at 1st Level', 'color', 'red', '
    fontsize', 4);//IPDtoolbox
22 figure(3);ShowImage(b, 'LH Component at 1st Level');
    title('LH Component at 1st Level', 'color', 'red', '
    fontsize', 4);//IPDtoolbox
23 figure(4);ShowImage(c, 'HL Component at 1st Level');
    title('HL Component at 1st Level', 'color', 'red', '
    fontsize', 4);//IPDtoolbox
24 figure(5);ShowImage(d, 'HH Component at 1st Level');
    title('HH Component at 1st Level', 'color', 'red', '
    fontsize', 4);//IPDtoolbox
25 figure(6);ShowImage(a1, ' 2nd Level Decomposition');
    title('2nd Level Decomposition', 'color', 'red', '
    fontsize', 4);//IPDtoolbox
26 figure(7);ShowImage(b1, ' 2nd Level Decomposition');
    title('2nd Level Decomposition', 'color', 'red', '
    fontsize', 4);//IPDtoolbox
27 figure(8);ShowImage(c1, ' 2nd Level Decomposition');
    title('2nd Level Decomposition', 'color', 'red', '
    fontsize', 4);//IPDtoolbox
28 figure(9);ShowImage(d1, ' 2nd Level Decomposition');
    title('2nd Level Decomposition', 'color', 'red', '
    fontsize', 4);//IPDtoolbox

```

Grayscale image



Figure 14.1: To Perform Wavelet Transform on grayscale image

LH Component at 1st Level

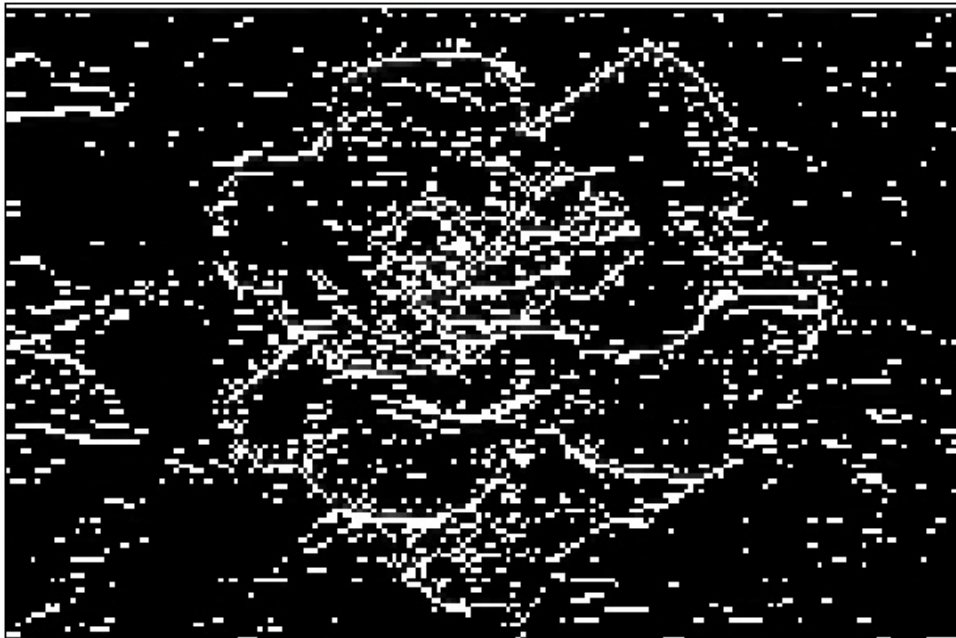


Figure 14.2: To Perform Wavelet Transform on grayscale image

LL Component at 1st Level



Figure 14.3: To Perform Wavelet Transform on grayscale image

HL Component at 1st Level



Figure 14.4: To Perform Wavelet Transform on grayscale image

HH Component at 1st Level



Figure 14.5: To Perform Wavelet Transform on grayscale image

2nd Level Decomposition



Figure 14.6: To Perform Wavelet Transform on grayscale image

2nd Level Decomposition

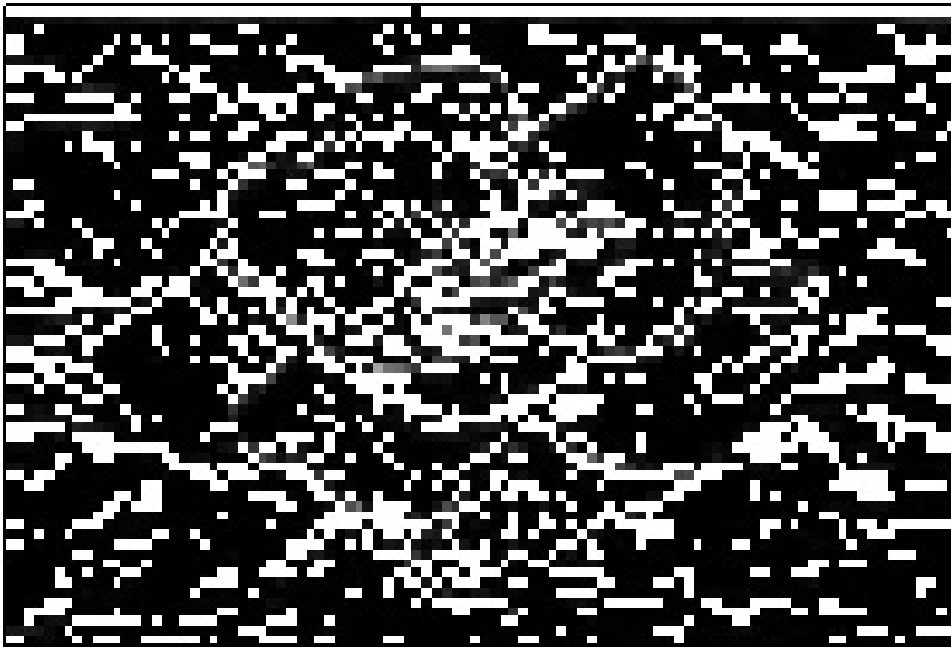


Figure 14.7: To Perform Wavelet Transform on grayscale image

2nd Level Decomposition



Figure 14.8: To Perform Wavelet Transform on grayscale image

2nd Level Decomposition

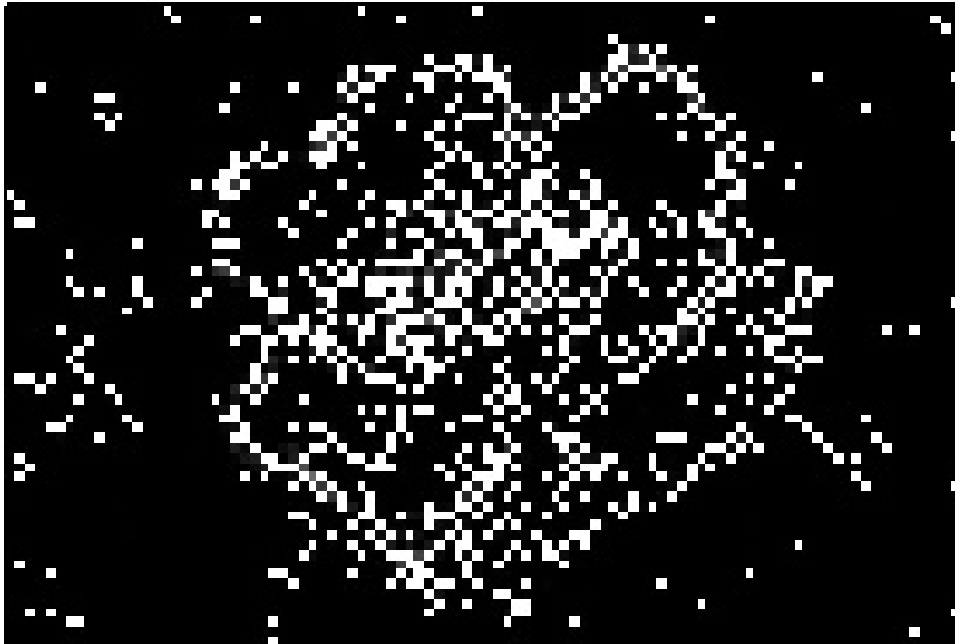


Figure 14.9: To Perform Wavelet Transform on grayscale image

check Appendix ?? for dependency:

rose.BMP

Appendix

```
Scilab code AP 12 function [i1,i2,i3]=im_dct(im2)
2 [r,c]=size(im2);
3 im3=double(im2);
4 im4=dct(im3); //dct of given image
5 im44=im4(1:r/2,1:c/2); //truncated DCT coefficients 1
   st level
6 im45=im4(1:r/4,1:c/4); //truncated DCT coefficients 2
   nd level
7 im46=im4(1:r/8,1:c/8); //truncated DCT coefficients 3
   rd level
8 i1=idct(im44); //inverse dct of given image DCT
   coeffi.
9 i2=idct(im45); //inverse dct of given image DCT
   coeffi.
10 i3=idct(im46); //inverse dct of given image DCT
   coeffi.
11 endfunction
```

TRUNCTION DCT IDCT

```
Scilab code AP 13 function i6=erod(i5)
2 [r1,c1]=size(i5);
3 i6=zeros(r1,c1);
4 w=[1 1 1;1 1 1;1 1 1];
5 for i=2:r1-1
6     for j=2:c1-1
7
```

```

8      aaa=[w(1)*i5(i-1,j-1),w(2)*i5(i-1,j),w(3)*i5(i-1,j+1),...
9          w(4)*i5(i,j-1),w(5)*i5(i,j),w(6)*i5(i,j+1),...
10         w(7)*i5(i+1,j-1),w(8)*i5(i+1,j),w(9)*i5(i+1,j+1)];
11     i6(i,j)=min(aaa); // erosion
12     end
13 end
14
15
16 endfunction

```

erosion

Scilab code AP 14 `function i6=dial(i5)`

```

2  [r1,c1]=size(i5);
3  i6=zeros(r1,c1);
4  w=[1 1 1;1 1 1;1 1 1];
5  for i=2:r1-1
6      for j=2:c1-1
7
8          aaa=[w(1)*i5(i-1,j-1),w(2)*i5(i-1,j),w(3)*i5(i-1,j+1),...
9              w(4)*i5(i,j-1),w(5)*i5(i,j),w(6)*i5(i,j+1),...
10             w(7)*i5(i+1,j-1),w(8)*i5(i+1,j),w(9)*i5(i+1,j+1)];
11         i6(i,j)=max(aaa); // Dilation
12     end
13 end
14 endfunction

```

dilation function

Scilab code AP 15 `function [r1,lmr,umr]=uplmean(red) //`
`function for calculation of upper mean and lower`

```

    mean color and formation of image in terms of
    upper mean and lower mean color
2  [r,c]=size(red); // row and columns
3  sm=0;
4  bmr=zeros(r,c);
5  for i=1:1:r
6      for j=1:1:c
7          sm=sm+red(i,j); // sum of all the
            intensity
8      end
9  end
10 mn=sm/(r*c); //mean threshold value
11 for i=1:1:r
12     for j=1:1:c
13         if red(i,j)>=mn //image binarization
14             bmr(i,j)=1;
15         else
16             bmr(i,j)=0;
17         end
18     end
19 end
20 sm=0;
21 for i=1:1:r
22     for j=1:1:c
23         sm=sm+bmr(i,j); // summation of binary
            image
24     end
25 end
26 m=0;
27 for i=1:1:r
28     for j=1:1:c
29         m=m+(bmr(i,j)*red(i,j));
30     end
31 end
32 umr=(1/sm)*m; // upper mean color value
33 f=0;
34 for i=1:1:r
35     for j=1:1:c

```

```

36         f=f+((1-bmr(i,j))*red(i,j));
37     end
38 end
39 lmr=(1/((r*c)-sm))*f; // lower mean color value
40 for i=1:r
41     for j=1:c
42         if bmr(i,j)==1
43             r1(i,j)=umr; // Replacing 1 with upper
                mean color value
44         else
45             r1(i,j)=lmr; // Replacing 0 with
                lower mean color value
46         end
47     end
48 end
49 endfunction

```

upper and lower mean color values

Scilab code AP 6

```

1 //FUNCTION FOR SEPERATION OF COLOR PLANE
2 function [i1,i2,i3]=colorplanes(img2) //creating
    function
3 i1=img2(:,:,1); //red color plane
4 i2=img2(:,:,2); //green color plane
5 i3=img2(:,:,3); //blue color plane
6 endfunction

```

colorplanes

Scilab code AP 7

```

1 function [im41,im42,im43]=imthresholding(im3); //
    function for thresholding im41 is thresholded
    image in terms of 0 and 1; //im42 is thresholded
    image using threshold 0 and 255;
2 im3=double(im3); //change data types
3 [row,col]=size(im3); // size of image row and column
4 sum1=0; // initialization of variable

```

```

5  for i=1:1:row
6      for j=1:1:col
7          sum1=sum1+im3(i,j); // addition of all the
              pixels
8      end
9  end
10 m=sum1/(row*col); // calculation of mean as threshold
11 [r1,lmr,umr]=uplmean(im3)
12 for i=1:1:row
13     for j=1:1:col
14         if im3(i,j)>=m; // comparison if pixel is greater
              than threshold
15             im41(i,j)=255; //make 1 in resulting image
16
17         else
18             im41(i,j)=0; //make 0 in resulting image
19
20         end
21     end
22 end
23
24 for i=1:1:row
25     for j=1:1:col
26         if im3(i,j)>=lmr; // comparison if pixel is
              greater than threshold
27             im42(i,j)=255; //make 1 in resulting image
28
29         else
30             im42(i,j)=0; //make 0 in resulting image
31
32         end
33     end
34 end
35
36 for i=1:1:row
37     for j=1:1:col
38         if im3(i,j)>=umr; // comparison if pixel is
              greater than threshold

```

```

39         im43(i,j)=255; //make 1 in resulting image
40
41     else
42         im43(i,j)=0; //make 0 in resulting image
43
44     end
45 end
46 end
47
48 endfunction

```

Image thersholding

Scilab code AP 8 `function a1=sum_of_product(im4,w1)`

```

2     im4=double(im4);
3     [r,c]=size(im4);
4     for i=2:1:r-1
5         for j=2:1:c-1
6             a1(i,j)=[w1(1)*im4(i-1,j-1)+...
7                     w1(2)*im4(i-1,j)+...
8                     w1(3)*im4(i-1,j+1)+...
9                     w1(4)*im4(i,j-1)+...
10                    w1(5)*im4(i,j)+...
11                    w1(6)*im4(i,j+1)+...
12                    w1(7)*im4(i+1,j-1)+...
13                    w1(8)*im4(i+1,j)+...
14                    w1(9)*im4(i+1,j+1)];
15         end
16     end
17 endfunction

```

Sum of product

Scilab code AP 9 `function [a1,a2]=masking2x2(im4,w1,w2)`

```

2     i2=double(im4);
3     [r c]=size(i2);
4
5     for i=2:r-1

```

```

6     for j=2:c-1
7         a1(i,j)=[w1(1)*i2(i,j)+...
8                 w1(2)*i2(i,j+1)+...
9                 w1(3)*i2(i+1,j)+...
10                w1(4)*i2(i+1,j+1)];
11        a2(i,j)=[w2(1)*i2(i,j)+...
12                w2(2)*i2(i,j+1)+...
13                w2(3)*i2(i+1,j)+...
14                w2(4)*i2(i+1,j+1)];
15    end
16 end
17 endfunction

```

masking 2x2

Scilab code AP 111 **function** [hh,eqq,q]=stogram(red)

```

2     img4=double(red);
3     [r,c]=size(img4);
4     hh=zeros(1,256);
5     for i=1:1:r
6         for j=1:1:c
7             if img4(i,j)==0
8                 img4(i,j)=1;
9             end
10        end
11    end
12
13    for i=1:1:r
14        for j=1:1:c
15            tt=img4(i,j);
16            hh(tt)=hh(tt)+1;
17        end
18    end
19    pd=hh/(r*c); //PDF
20    cdf(1)=pd(1);
21    m=max(max(img4));
22    for xx=2:1:m
23        cdf(xx)=pd(xx)+cdf(xx-1); //CDF

```

```

24 end
25 new=round(cdf*m); //new rounded pixel values
26 new=new+1;
27 q=zeros(1,257);
28 for ii=1:1:r
29     for jj=1:1:c
30         tt=img4(ii,jj);
31         eq(ii,jj)=new(tt);
32         t1=eq(ii,jj);
33         q(t1)=q(t1)+1;
34     end
35 end
36 eq=eq-1; // equalized intensity levels
37 eqq=uint8(eq); //
38 endfunction

```

histogram function

```

Scilab code AP 12 function [av1]=masking(im4,w)
2     [r,c]=size(im4); //getting the rows and columns
      of image im4;
3 for i=2:1:r-1
4     for j=2:1:c-1
5         av1(i,j)=[w(1)*im4(i-1,j-1)+...
6                 w(2)*im4(i-1,j)+...
7                 w(3)*im4(i-1,j+1)+...
8                 w(4)*im4(i,j-1)+...
9                 w(5)*im4(i,j)+...
10                w(6)*im4(i,j+1)+...
11                w(7)*im4(i+1,j-1)+...
12                w(8)*im4(i+1,j)+...
13                w(9)*im4(i+1,j+1)];
14     end
15 end
16 endfunction

```

masking function

Scilab code AP 13 `function [ii]=log_im(im3)`

```
2   im5=double(im3);
3   [r,c]=size(im5);
4   cc=255/log10(1+max(max(im5)));
5   for i=1:1:r
6       for j=1:1:c
7           im=im5(i,j);
8           t=cc.*log10(1+im);
9           ii(i,j)=t;
10      end
11  end
12  ii=uint8(ii);
13  endfunction
```

log function

Scilab code AP 14 `function iii=gamm1(im3,cc)`

```
2   im5=double(im3);
3   [r,c]=size(im1);
4   g=1.2;
5   for i=1:1:r
6       for j=1:1:c
7           im=im5(i,j);
8           t=cc.*im.^g;
9           iii(i,j)=t;
10      end
11  end
12  endfunction
```

gamma function

Scilab code AP 15

```
1   function [umr,lmr,im2]=upper_lower_color(red)
2   red=double(red);
3   [r,c]=size(red);
4   sm1=0;
5   bmr=zeros(r,c);
6   for i=1:1:r
```

```

7         for j=1:1:c
8             sm1=sm1+red(i,j);
9         end
10    end
11    mn=sm1/(r*c); //mean calculation
12    for i=1:1:r
13        for j=1:1:c
14            if red(i,j)>=mn
15
16                bmr(i,j)=1; //binary image formation
17            else
18                bmr(i,j)=0;
19            end
20        end
21    end
22    sm=0;
23    for i=1:1:r
24        for j=1:1:c
25            sm=sm+bmr(i,j);
26        end
27    end
28    m=0;
29    for i=1:1:r
30        for j=1:1:c
31            m=m+(bmr(i,j)*red(i,j));
32        end
33    end
34    umr=(1/sm)*m; //upper mean color value
35    f=0;
36    for i=1:1:r
37        for j=1:1:c
38            f=f+((1-bmr(i,j))*red(i,j));
39        end
40    end
41    lmr=(1/((r*c)-sm))*f; //lower mean color value
42    for i=1:r;
43        for j=1:c;
44            if red(i,j)<=lmr then

```

```

45         im2(i,j)=0.5*red(i,j);
46     else if red(i,j)<=umr then
47         im2(i,j)=2.5*(red(i,j)-lmr)+0.5*lmr;
48     else im2(i,j)=0.5*(red(i,j)-umr)+0.5*lmr
         +2.5*(umr-lmr);
49     end
50 end
51 end
52 end
53
54 endfunction

```

upper lower color function

Scilab code AP 117 `function [im2,im3]=grayslic(im1)`

```

2     mm=max(im1); //calculate maximum pixel value
3 m1=mm/2; //half of the maxi pixel value
4 m11=m1/2; //lower bound
5 m12=m1+m11; // upper bound
6 im1=double(im1);
7 [r,c]=size(im1); //row and column
8 //gray level slicing without background
9 for i=1:r;
10    for j=1:c;
11        if (im1(i,j)>=m11)&(im1(i,j)<=m12) then
12            im2(i,j)=255;
13        else
14            im2(i,j)=0;
15        end
16    end
17 end
18 //gray level slicing with background
19 for i=1:r;
20    for j=1:c;
21        if (im1(i,j)>=m11)&(im1(i,j)<=m12) then
22            im3(i,j)=255;
23        else
24            im3(i,j)=im1(i,j);

```

```
25
26     end
27 end
28 end
29 endfunction

graylevel slicing function
```

```
Scilab code AP 18 function [im3,im2]=thres(im1)
2 im1=double(im1);
3 [r,c]=size(im1);
4 m=0;
5 for i=1:r;
6     for j=1:c;
7         m=m+im1(i,j);
8     end
9 end
10 m=m/(r*c); //mean threshold
11 m1=(max(im1))/2; //threshold 2
12 for i=1:r;
13     for j=1:c;
14         if im1(i,j)>=m then
15             im2(i,j)=255;
16         else
17             im2(i,j)=0;
18         end
19     end
20 end
21 for i=1:r;
22     for j=1:c;
23         if im1(i,j)>=m1 then
24             im3(i,j)=255;
25         else
26             im3(i,j)=0;
27         end
28     end
29 end
30 endfunction
```

Function for Thesholding

Scilab code AP 19 `function im4=negimg(im)`

```
2 im1=double(im);
3 im4=255-im1; //maximum pixel value and image pixel
  subtraction
4 endfunction
```

Function for negation

Scilab code AP 20

```
1 function F1=quag(ii,bb); //function for
  quantization at different(no.of bits)gray level
2 cc=double(ii); //change data type
3 b=cc+1 //make zero pixel value to 1
4 c=max(max(b)); //find out maxium pixel value in
5 t=c/(2^bb); //Maximum pixel gray level
6 F=floor(b/(t+1)); // each pixel in the matrix
7 F1=(F*255)/max(max(F)); //normalization
8 endfunction
```

Quantization function

Scilab code AP 21 // function for sampling

```
2 function t=sam(ii1,d) //d is the input for number of
  samples
3   ii=rgb2gray(ii1);
4   [row,col]=size(ii);
5   b1=1;b2=1; //initilaizaion of row and column
6   for i=1:d:row
7       for j=1:d:col
8           t(b1,b2)=ii(i,j); //storing d sample
9           b2=b2+1; //increament col
10      end
11  b2=1;
12  b1=b1+1; //increament row
13  end
```

14 `endfunction`

Sampling function
