

Academic year

2025-2026

Faculty of Applied Engineering

Digital Image Processing

Solutions to the Exercises

Walter Daems

Bachelor of Science in de industriële wetenschappen - elektronica-ICT

1509FTIABV 5-Digital Image Processing

1517FTIDSB 5-Digitale beeldverwerking

This document has been typeset using \LaTeX and the `uantwerpen` package.
Calculations have been performed using Matlab/Octave and generic programming and script languages (C/C++/Perl/Python).
Graphics have been composed using PGF, TiKZ and InkScape.
All this material has been prepared on a GNU/Linux workstation.

All trademarks are copyright of their respective owners.

Typesetting of this document was enabled by:



This document is under copyright. However, if you want to obtain a free license to use and distribute it (whether it as a lecturer or as a student), send an e-mail with your request to the author (walter.daems@uantwerpen.be).

DIP-2024-3.11

CONFIDENTIAL AND PROPRIETARY.

© 2025 University of Antwerp, All rights reserved.

Contents

2	Discrete Calculus	1
4	Sampling and Resampling	5
5	Quantization and Requantization	9
6	Light and Color Modeling	21
7	Mathematical Morphology	25
8	Image Segmentation	35

Discrete Calculus

Solution 2.2.2-1: The solution is:

$$\frac{df}{dx}[x] = [7, -5, -5, 3, -2, \textcircled{-7}, 3, -2, 1, 1, 6]$$

Solution 2.2.2-2: The solution is:

$$\frac{df}{dx}[x] = [3.5, 1, -5, -1, 0.5, \textcircled{-4.5}, -2, 0.5, -0.5, 1, 3.5, 3]$$

Solution 2.3.2-1: The solution is:

$$\frac{d^2f}{dx^2}[x] = [7, -12, 0, 8, -5, -5, \textcircled{10}, -5, 3, 0, 5, -6, 0, 0]$$

Solution 2.3.2-2: The solution is:

$$\frac{d^2f}{dx^2}[x] = [7, -12, 0, 8, -5, \textcircled{-5}, 10, -5, 3, 0, 5, -6]$$

Solution 2.4.2-1:

$$(f \star g)[x] = \begin{bmatrix} 6.8, & 6.66, & 42.85, & -47.13, & \textcircled{56.21}, & -40.94, & 60.86, & -60.97, \\ -37.46, & -11.29, & 29.33, & -10.2, & 1.5 \end{bmatrix}$$

Solution 2.4.2-2:

$$(f \star g)[x] = \begin{bmatrix} 72.21, & -98.83, & 87.81, & \textcircled{-4.37}, & -48.9, & 34.02, & -90.73, & -11.13, \\ 58.34, & 36.92, & -35.26, & 7.48, & -3.36 \end{bmatrix}$$

Solution 2.8.3-1:

Sampling and Resampling

Solution 4.4.4.1-1: The matrix A is composed by filling the columns with the coordinates of the new basis vectors (in terms of the old basis vectors). Therefore:

$$A = \begin{bmatrix} 0.3 & -2 \\ -0.8 & 3 \end{bmatrix}$$

The point with coordinates $(3, 2)$ is transformed to new coordinates (u, v) according to:

$$\begin{bmatrix} u \\ v \end{bmatrix} = A \cdot \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 0.3 & -2 \\ -0.8 & 3 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} -3.1 \\ 3.6 \end{bmatrix}$$

The transformation matrix related to the inverse transformation is A^{-1} :

$$A^{-1} = \begin{bmatrix} -4.286 & -2.857 \\ -1.143 & -0.429 \end{bmatrix}$$

And indeed:

$$\begin{bmatrix} x \\ y \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} -3.1 \\ 3.6 \end{bmatrix} = \begin{bmatrix} -4.286 & -2.857 \\ -1.143 & -0.429 \end{bmatrix} \cdot \begin{bmatrix} -3.1 \\ 3.6 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

Solution 4.4.4.1-2: We start by writing the coordinates of the new basis vectors as columns of the matrix A below:

$$A = \begin{bmatrix} 1 & -0.5 \\ 1 & 2 \end{bmatrix}$$

Using this matrix, we can determine the coordinates of $T(-0.5, -1)$:

$$\begin{bmatrix} u \\ v \end{bmatrix} = A \cdot \begin{bmatrix} -0.5 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2.5 \end{bmatrix}$$

This has been illustrated in the picture below:

We need to execute $T_t \circ T_s \circ T_r$ and therefore the total transformation matrix becomes:

$$\begin{aligned} T = T_t \cdot T_s \cdot T_r &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\left(-\frac{\pi}{3}\right) & -\sin\left(-\frac{\pi}{3}\right) & 0 \\ \sin\left(-\frac{\pi}{3}\right) & \cos\left(-\frac{\pi}{3}\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} -1.23205 & 1.86603 & 1 \\ -0.86603 & 0.5 & 2 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Applying this transformation to the points $(0,0)$ and $(-1,4)$ yields:

$$\begin{aligned} (0,0) &\mapsto \begin{bmatrix} -1.23205 & 1.86603 & 1 \\ -0.86603 & 0.5 & 2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \\ (-1,4) &\mapsto \begin{bmatrix} -1.23205 & 1.86603 & 1 \\ -0.86603 & 0.5 & 2 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 9.6962 \\ 4.8660 \\ 1 \end{bmatrix} \end{aligned}$$

Quantization and Requantization

Solution 5.2.3-1: Let's start by calculating the interval size:

$$\Delta x = \frac{x_{\max} - x_{\min}}{2^N} = \frac{1.2 \text{ V} - (-0.2 \text{ V})}{2^8} = \frac{1.4 \text{ V}}{256} = 5.4687 \text{ mV}$$

Then, let's determine in which interval 0.6 V is located:

$$v = \frac{x - x_{\min}}{\Delta x} = \frac{0.6 \text{ V} - (-0.2 \text{ V})}{5.4687 \text{ mV}} = 146.29$$

Therefore x is located in I_{146} . The corresponding binary code is 0b10010010 or 0x98 in hexadecimal notation.

The voltage corresponding to interval number 100 is calculated as:

$$\begin{aligned} x &= (-0.2 \text{ V} + 100.5\Delta x) \pm 0.5\Delta x \\ &= (-0.2 \text{ V} + 0.54961 \text{ V}) \pm 0.00274 \text{ V} \\ &= 0.34961 \text{ V} \pm 0.00274 \text{ V} \end{aligned}$$

Solution 5.2.3-2: Let's start by calculating the interval size:

$$\Delta x = \frac{x_{\max} - x_{\min}}{2^N} = \frac{1.5 \text{ V} - (-1.5 \text{ V})}{2^{10}} = \frac{3 \text{ V}}{1024} = 2.9297 \text{ mV}$$

Let's now map 0 V to its specific interval:

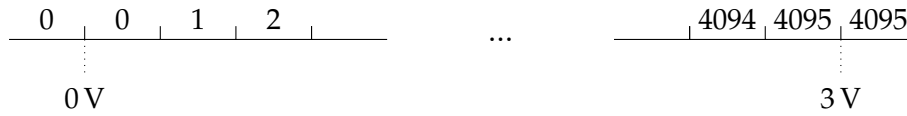
$$v = \frac{x - x_{\min}}{\Delta x} = \frac{0 \text{ V} - (-1.5 \text{ V})}{2.9297 \text{ mV}} = 512$$

This means that 0 V lies spot on the boundary of two intervals, i.e. 511 and 512. With equal probability the converter will judge it to be in one of those two intervals. A proper reconstruction value will therefore be $+\Delta x$ or $-\Delta x$ (depending on how your DSP rolls its dice).

The voltage corresponding to interval number 314 can be found as:

$$\begin{aligned} x &= (-1.5 \text{ V} + 314.5\Delta x) \pm 0.5\Delta x \\ &= (-1.5 \text{ V} + 0.92139 \text{ V}) \pm 0.00147 \text{ V} \\ &= -0.57861 \text{ V} \pm 0.00147 \text{ V} \end{aligned}$$

Solution 5.2.3-3: The mapping is as follows:



From this drawing, the interval size can be determined, noting that the interval from 0 to 3 V is divided in 4096 intervals:

$$\Delta x = \frac{3\text{ V} - 0\text{ V}}{4096} = 732.42\ \mu\text{V}$$

A value of 2000 corresponds to an input voltage x in the following range:

$$2000\Delta x \leq x < 2001\Delta x$$

Therefore:

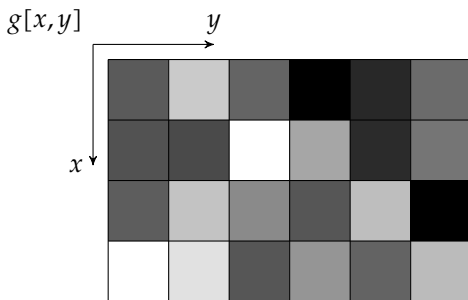
$$\begin{aligned} x &= 2000.5\Delta x \pm 0.5\Delta x \\ &= 1.4652\text{ V} \pm 0.0004\text{ V} \end{aligned}$$

Solution 5.2.3-4: Submit your solution to your lecturer. He will be happy to check it for you.

The image negative:

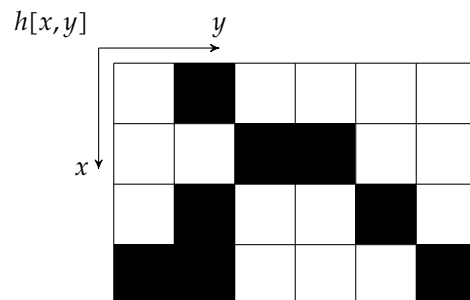
Solution 5.4.2.1-1:

$g[x,y]$	y					
	90	202	100	0	40	107
x	82	74	255	166	43	117
	93	195	138	86	190	0
	255	225	86	149	99	187



The threshold transformation:

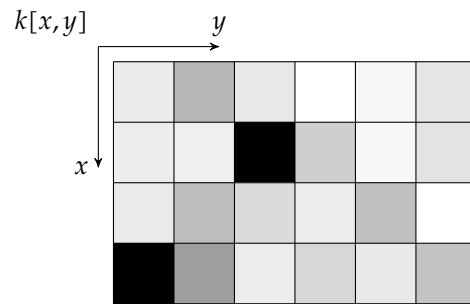
$h[x,y]$	y					
	255	0	255	255	255	255
x	255	255	0	0	255	255
	255	0	255	255	0	255
	0	0	255	255	255	0



The log transformation:

$$k[x,y]$$

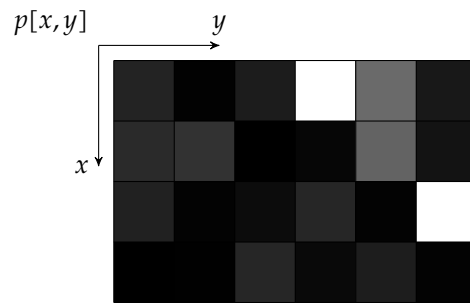
	y					
x	235	183	232	255	247	230
	237	239	0	207	247	227
	234	189	219	236	193	255
	0	158	236	215	233	195



The inverse log transformation:

$$p[x,y]$$

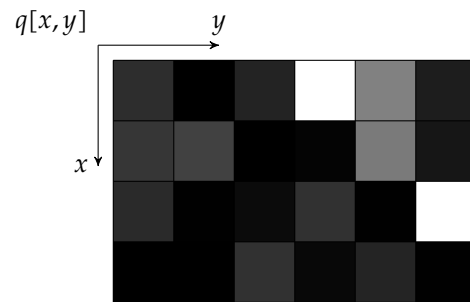
	y					
x	35	2	28	255	106	24
	42	50	0	6	99	19
	33	3	12	38	3	255
	0	1	38	9	29	3



The gamma transformation with $\gamma = 4$:

$$q[x,y]$$

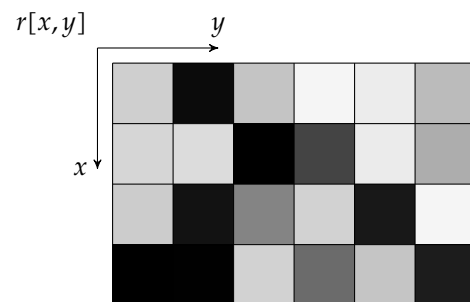
	y					
x	45	0	35	255	129	29
	54	65	0	4	122	22
	42	1	11	49	1	255
	0	0	49	8	36	1



The contrast stretching transformation:

$$r[x,y]$$

	y					
x	207	11	196	245	236	187
	214	220	0	68	235	173
	204	18	132	210	24	245
	0	1	210	107	197	28

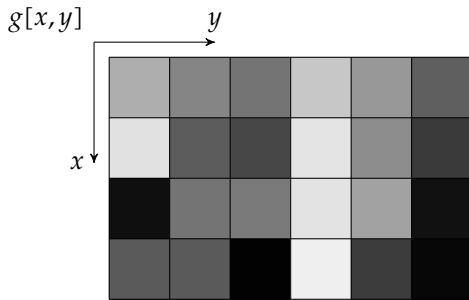


The image negative:

Solution 5.4.2.1-2:

$$g[x,y]$$

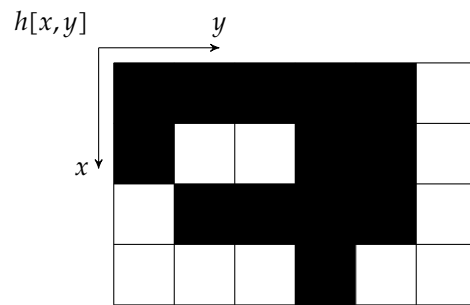
174	133	116	199	152	95
225	92	71	228	141	58
15	116	122	227	162	17
90	90	2	239	60	7



The threshold transformation:

$$h[x,y]$$

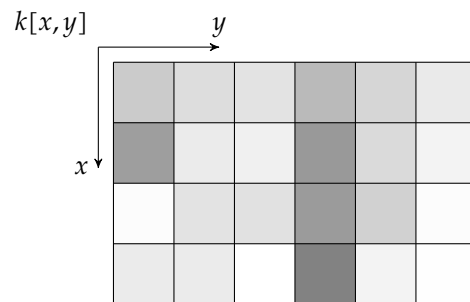
0	0	0	0	0	255
0	255	255	0	0	255
255	0	0	0	0	255
255	255	255	0	255	255



The log transformation:

$$k[x,y]$$

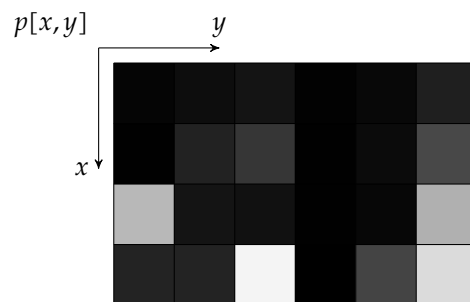
203	221	227	186	214	234
158	235	240	153	218	243
252	227	225	155	209	252
235	235	255	130	243	254



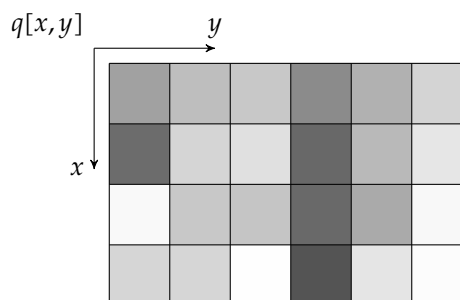
The inverse log transformation:

$$p[x,y]$$

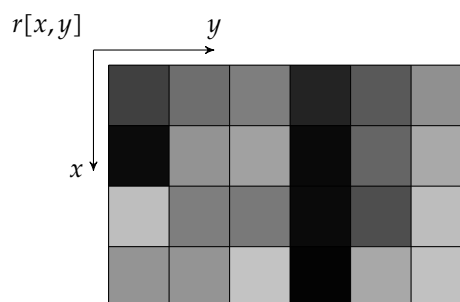
5	13	20	2	8	31
1	34	54	1	11	72
184	20	17	1	7	176
35	35	244	0	68	219



The gamma transformation with $\gamma = 4$:



The contrast stretching transformation:



Solution 5.4.2.1-3: The standard intensity transformations' formulae have been composed such that they generate a range from 0 to $L - 1$. Scaling them to output a range from 0 to $M - 1$ is simple. Just multiply them with a factor $(M - 1)/(L - 1)$ before quantization. Let's demonstrate this for the image negative. I'm sure you will be able to work out the other ones yourself.

The original transform is:

$$Q_{neg} : m = L - 1 - l.$$

After scaling, it becomes:

$$Q_{neg} : m = \frac{M - 1}{L - 1}(L - 1 - l)$$

Solution 5.4.2.1-4: To avoid spoiling your search, no solution is given here. Send yours to the lecturer. He will be happy to comment on it.

Solution 5.4.2.2-1: The table below shows the result. Using a spreadsheet makes the calculations most easy.

l	n	$f_i(l)$	$F_i(l)$	$15 \cdot F_i(l)$	m
0	36	0.09	0.09	1.35	1
1	32	0.08	0.17	2.55	3
2	25	0.06	0.23	3.49	3
3	20	0.05	0.28	4.24	4
4	17	0.04	0.33	4.88	5
5	14	0.04	0.36	5.40	5
6	16	0.04	0.40	6.00	6
7	17	0.04	0.44	6.64	7
8	18	0.05	0.49	7.31	7
9	19	0.05	0.54	8.03	8
10	23	0.06	0.59	8.89	9
11	57	0.14	0.74	11.03	11
12	37	0.09	0.83	12.41	12
13	30	0.08	0.90	13.54	14
14	25	0.06	0.97	14.48	14
15	14	0.04	1.00	15.00	15

You might also want to try the same in MATLAB/OCTAVE.

```
l = [ 36 32 25 20 17 14 16 17 18 19 23 57 37 30 25 14 ];
f = l / sum(l);
F = cumsum(f);
m = round( 15 * F );
```

Alternatively, one might use the `histeq` function.

Solution 5.4.2.2-2: The table below shows the result. Using a spreadsheet makes the calculations most easy.

l	n	$f_i(l)$	$F_i(l)$	$15 \cdot F_i(l)$	m
0	1	0.00	0.00	0.04	0
1	11	0.03	0.03	0.45	0
2	21	0.05	0.08	1.24	1
3	31	0.08	0.16	2.40	2
4	41	0.10	0.26	3.94	4
5	35	0.09	0.35	5.25	5
6	31	0.08	0.43	6.41	6
7	21	0.05	0.48	7.20	7
8	11	0.03	0.51	7.61	8
9	1	0.00	0.51	7.65	8
10	11	0.03	0.54	8.06	8
11	21	0.05	0.59	8.85	9
12	31	0.08	0.67	10.01	10
13	41	0.10	0.77	11.55	12
14	51	0.13	0.90	13.46	13
15	41	0.10	1.00	15.00	15

You might also want to try the same in MATLAB/OCTAVE.

```
l = [ 1 11 21 31 41 35 31 21 11 1 11 21 31 41 51 41 ];
f = l / sum(l);
F = cumsum(f);
m = round( 15 * F );
```

Alternatively, one might use the `histeq` function.

Solution 5.4.2.2-3: The mapping of both histograms onto the unrounded uniform intensities can be found in the solution of the previous exercises. Based on these results we create a distance table.

l	m																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0.04	0.45	1.24	2.40	3.94	5.25	6.41	7.20	7.61	7.65	8.06	8.85	10.01	11.55	13.46	15.00	
1	1.35	1.31	0.90	0.11	1.05	2.59	3.90	5.06	5.85	6.26	6.30	6.71	7.50	8.66	10.20	12.11	13.65
2	2.55	2.51	2.10	1.31	0.15	1.39	2.70	3.86	4.65	5.06	5.10	5.51	6.30	7.46	9.00	10.91	12.45
3	3.49	3.45	3.04	2.25	1.09	0.45	1.76	2.93	3.71	4.13	4.16	4.58	5.36	6.53	8.06	9.98	11.51
4	4.24	4.20	3.79	3.00	1.84	0.30	1.01	2.18	2.96	3.38	3.41	3.83	4.61	5.78	7.31	9.23	10.76
5	4.88	4.84	4.43	3.64	2.48	0.94	0.38	1.54	2.33	2.74	2.78	3.19	3.98	5.14	6.68	8.59	10.13
6	5.40	5.36	4.95	4.16	3.00	1.46	0.15	1.01	1.80	2.21	2.25	2.66	3.45	4.61	6.15	8.06	9.60
7	6.00	5.96	5.55	4.76	3.60	2.06	0.75	0.41	1.20	1.61	1.65	2.06	2.85	4.01	5.55	7.46	9.00
8	6.64	6.60	6.19	5.40	4.24	2.70	1.39	0.23	0.56	0.97	1.01	1.43	2.21	3.38	4.91	6.83	8.36
9	7.31	7.28	6.86	6.08	4.91	3.38	2.06	0.90	0.11	0.30	0.34	0.75	1.54	2.70	4.24	6.15	7.69
10	8.03	7.99	7.58	6.79	5.63	4.09	2.78	1.61	0.83	0.41	0.38	0.04	0.82	1.99	3.53	5.44	6.98
11	8.89	8.85	8.44	7.65	6.49	4.95	3.64	2.48	1.69	1.28	1.24	0.83	0.04	1.13	2.66	4.58	6.11
12	11.03	10.99	10.58	9.79	8.63	7.09	5.78	4.61	3.83	3.41	3.38	2.96	2.18	1.01	0.52	2.44	3.98
13	12.41	12.38	11.96	11.18	10.01	8.48	7.16	6.00	5.21	4.80	4.76	4.35	3.56	2.40	0.86	1.05	2.59
14	13.54	13.50	13.09	12.30	11.14	9.60	8.29	7.13	6.34	5.93	5.89	5.48	4.69	3.53	1.99	0.08	1.46
15	14.48	14.44	14.03	13.24	12.08	10.54	9.23	8.06	7.28	6.86	6.83	6.41	5.63	4.46	2.93	1.01	0.53
15	15.00	14.96	14.55	13.76	12.60	11.06	9.75	8.59	7.80	7.39	7.35	6.94	6.15	4.99	3.45	1.54	0.00

The mapping from l to m can now be found by finding the minimum in every row. The mapping from m to l can be found bin finding the minimum in every column. This leads to the following table:

From l	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
To m	2	3	4	4	5	5	6	6	7	10	11	13	13	14	15	15

And vice versa:

From m	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
To l	0	0	0	1	3	5	7	8	8	8	9	10	11	11	13	15

In MATLAB/OCTAVE it would not be wise to compose the large distance matrix. For a 4-bit quantization this is still feasible, but the matrix grows with the square of the number of quantization levels. We can exploit the nature of the matrix. Fact is that every row and every column of the matrix is monotonically increasing. Therefore, we can use the `histc` function. The solution contains quite some fiddling to make it robust. Try building your own solution without the fiddling and see where it goes wrong. Based on that result, you will understand the fiddling.

```
% calculate the mapping of l to a uniform histogram
l = [ 36 32 25 20 17 14 16 17 18 19 23 57 37 30 25 14 ];
fl = l / sum(l);
F1 = cumsum(fl);
l1 = 15 * F1;

% calculate the mapping of m to a uniform histogram
m = [ 1 11 21 31 41 35 31 21 11 1 11 21 31 41 51 41 ];
fm = m / sum(m);
Fm = cumsum(fm);
mm = 15 * Fm;

% determine the mapping from l to lm
[ N, idx ] = histc( l1*(1-eps), mm); % factor 1-eps to avoid 16th bin
idx = idx + (idx == 0);           % avoid 0th bin
enclosing(1,:) = mm(idx) - l1;
enclosing(2,:) = mm(idx+1) - l1;
idx = idx + ( sum(enclosing) < 0 );
lm = idx - 1; % bin1 corresponds to 0, bin2 to 1, a.s.o.

% determine the mapping from m to ml
[ N, idx ] = histc( mm*(1-eps), l1); % factor 1-eps to avoid 16th bin
idx = idx + (idx == 0);           % avoid 0th bin
enclosing(1,:) = l1(idx) - mm;
enclosing(2,:) = l1(idx+1) - mm;
idx = idx + ( sum(enclosing) < 0 );
ml = idx - 1; % bin1 corresponds to 0, bin2 to 1, a.s.o.
```

Alternatively, one might use the `histmatch` function.

Solution 5.4.2.2-4: Send the picture and the mapping table (e.g. in a spreadsheet), or your MATLAB/OCTAVE code to your lecturer. He will be happy to check it for you.

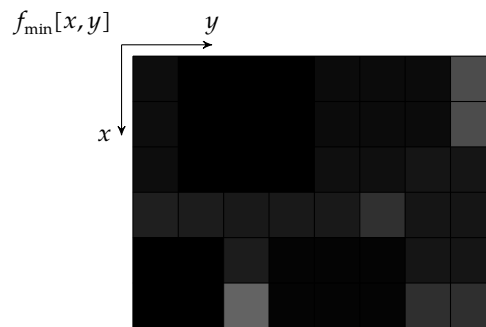
Solution 5.4.2.2-5: Send the picture and the mapping table (e.g. in a spreadsheet), or your MATLAB/OCTAVE code to your lecturer. He will be happy to check it for you.

Solution 5.4.2.2-6: Send the pictures and the mapping table (e.g. in a spreadsheet), or your MATLAB/OCTAVE code to your lecturer. He will be happy to check it for you.

Solution 5.4.3.2-1: The result of applying the Min-filter:

$$f_{\min}[x, y]$$

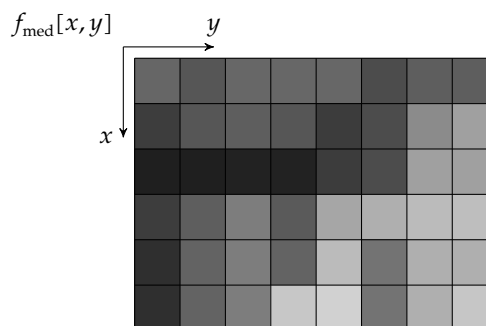
	y								
x		13	0	0	0	11	11	11	76
		13	0	0	0	11	11	11	76
		13	0	0	0	15	15	21	21
		31	28	25	25	25	48	21	21
		0	0	28	4	4	4	21	21
		0	0	99	4	4	4	47	47



The result of applying the Median-filter:

$$f_{\text{med}}[x, y]$$

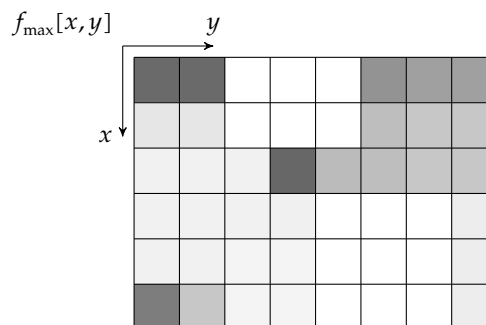
	y								
x		102	86	103	103	103	76	94	94
		61	86	94	86	60	76	139	160
		31	31	34	34	60	76	160	160
		61	94	125	90	166	175	187	190
		47	99	125	99	187	115	175	175
		47	99	125	199	209	115	175	198



The result of applying the Max-filter:

$$f_{\max}[x, y]$$

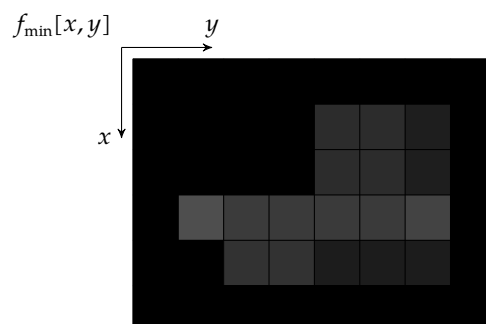
	y								
x		106	106	255	255	255	147	160	160
		230	230	255	255	255	190	199	199
		241	241	241	103	187	190	199	199
		241	241	241	240	255	255	255	237
		241	241	244	244	255	255	255	237
		125	199	244	244	255	255	255	237



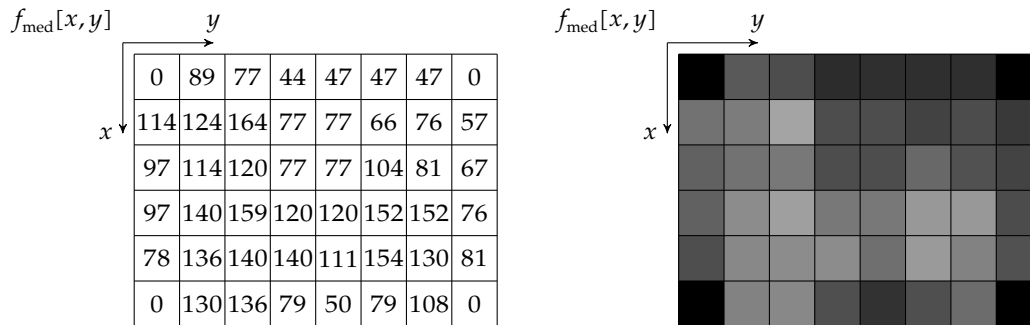
Solution 5.4.3.2-2: The result of applying the Min-filter:

$$f_{\min}[x, y]$$

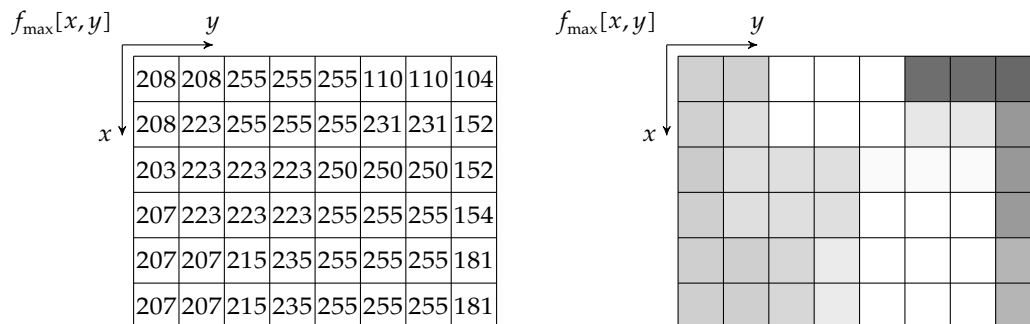
	y								
x		0	0	0	0	0	0	0	
		0	0	0	0	44	44	30	0
		0	0	0	0	44	44	30	0
		0	78	59	58	58	58	67	0
		0	0	50	50	28	28	28	0
		0	0	0	0	0	0	0	0



The result of applying the Median-filter:



The result of applying the Max-filter:



Solution 5.4.3.2-3: The following code will do the trick:

```

a = [ 102  106   86  255  147   11  139   94;
      19   13    0  103   15   43   76  160;
      61  230   94   25   60  166  190  199;
      31  241   28   34   90  187   48   21;
      47  125  199  240  209  255  175  237;
      0   125   99  244    4  115   47  198 ];

subplot(2,2,1);
imshow( mat2gray( a ) );
title( 'original' );

mina = ordfilt2( a, 1, ones(3,3), 'symmetric' )
subplot(2,2,2);
imshow( mat2gray( mina ) );
title( 'result_of_Min-filter' );

meda = ordfilt2( a, 5, ones(3,3), 'symmetric' )
subplot(2,2,3);
imshow( mat2gray( meda ) );
title( 'result_of_Median-filter' );

maxa = ordfilt2( a, 9, ones(3,3), 'symmetric' )
subplot(2,2,4);
imshow( mat2gray( maxa ) );
title( 'result_of_Max-filter' );

```

Adapt the code above for the second exercise yourself.

Solution 5.4.3.2-4: Send your results to the lecturer. He will be happy to check it.

Solution 5.4.3.3-1: The result of applying the moving average filter with symmetric extension (left) and zero padding (right):

	y								
x		87	98	121	186	162	111	57	56
		117	109	104	137	139	135	109	113
		166	130	85	94	106	148	127	125
		143	136	116	97	123	169	156	120
		98	95	84	97	120	145	102	58
		34	44	54	91	127	132	82	41

	y								
x		38	62	64	108	98	72	37	23
		75	109	104	137	139	135	109	73
		106	130	85	94	106	148	127	91
		100	136	116	97	123	169	156	95
		67	95	84	97	120	145	102	52
		25	38	46	62	88	93	64	24

Solution 5.4.3.3-2: The result of applying the Laplacian filter with symmetric extension (left) and zero padding (right):

	y								
x		187	548	-876	-617	-612	155	285	-36
		-782	666	934	957	-919	964	673	207
		507	-827	-37	427	-169	-456	-1013	-757
		-802	-112	-133	323	-22	629	-788	1043
		467	-405	-161	-81	304	-991	-402	423
		306	67	315	701	-931	257	600	-16

	y								
x		-252	227	-1386	1320	-1191	-194	106	-331
		-1162	666	934	957	-919	964	673	-152
		-39	-827	-37	427	-169	-456	-1013	-1060
		-1190	-112	-133	323	-22	629	-788	819
		189	-405	-161	-81	304	-991	-402	365
		223	11	246	439	-1277	-91	439	-171

Solution 5.4.3.3-3: Correlation yields:

	y								
x		70.40	-143.62	-36.33	19.23	421.92	75.06	336.61	47.35
		287.84	250.70	172.98	-121.51	37.27	-84.20	98.42	165.36
		-112.76	216.63	-27.40	467.19	-29.59	411.26	51.21	-80.16
		126.04	156.18	-142.25	101.59	96.31	0.73	319.58	19.61
		90.34	161.21	483.99	-75.73	149.98	49.89	259.90	131.65
		-121.31	203.65	343.54	84.41	-162.95	85.57	178.83	346.05

Convolution yields:

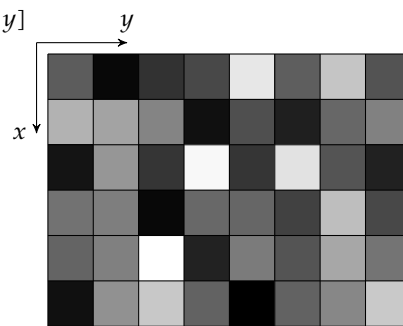
$$(f \star h)[x, y]$$

		y							
x		-78.95	-71.87	-64.79	492.84	188.48	488.99	-8.95	-10.75
		289.22	28.34	-272.50	88.28	70.72	241.76	203.06	136.31
		144.43	120.73	497.79	-5.23	255.52	-149.96	139.52	-64.67
		83.88	-153.53	123.58	86.51	36.62	436.63	98.28	163.88
		274.28	337.79	-111.87	201.64	-78.12	260.64	124.95	-86.47
		-14.28	324.76	62.84	92.71	78.76	73.18	352.77	177.66

These intensities extend beyond the range 0 to 255. In many cases it makes sense to remap the range of the resulting image to the valid range of 0 to 255. Doing so yields:

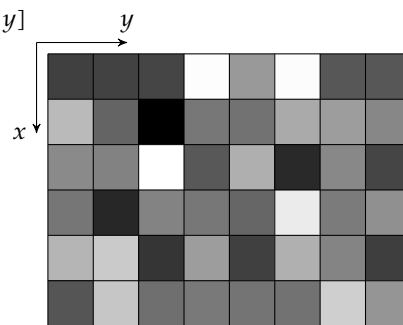
$$(f \star h)_{scaled}[x, y]$$

		y							
x		92	8	50	72	231	94	197	83
		178	163	132	16	79	31	103	129
		20	150	53	248	53	226	84	33
		114	126	8	104	102	65	190	72
		100	128	255	34	123	84	167	116
		16	145	200	98	0	98	135	201



$$(f \star h)_{scaled}[x, y]$$

		y							
x		64	66	69	253	153	252	87	87
		186	100	0	119	114	170	157	135
		138	130	255	88	175	41	136	69
		118	39	131	119	102	235	123	144
		181	202	53	157	64	176	132	62
		85	198	111	121	116	114	207	149



Light and Color Modeling

Solution 6.1.4-1: The rays that are targeted on point F leave the reflector under an angle of 30° . Under that angle, the luminous intensity I_v amounts

$$I_v = 1\,500 \text{ cd.}$$

The distance between L and F can be calculated using basic trigonometry:

$$|LF| = \frac{|LL'|}{\cos 30^\circ} = 2.309\,4 \text{ m}$$

The radial flux density at F amounts to:

$$\begin{aligned} E_{v,R} &= \frac{\partial \Phi_v}{\partial A} \\ &= \frac{\partial \Phi_v}{\partial \Omega} \cdot \frac{\partial \Omega}{\partial A} \\ &= I_v \cdot \frac{1}{|LF|^2} \\ &= 281 \text{ lx} \end{aligned}$$

However, we still need to correct for the fact that the same flux needs to cover more area, i.e. the correction for the *observation angle*. This is the angle between the normal on the surface and the incident light waves. This angle can be calculated to be 30° (by coincidence equal to the emission angle of the light waves that are considered), i.e.

$$E_{v,S} = E_{v,R} \cdot \cos 30^\circ = 243.57 \text{ lx}$$

Solution 6.1.4-2: The rays that are targeted on point F leave the reflector under an angle of 30° . Under that angle, the luminous intensity I_v amounts

$$I_v = 2\,500 \text{ cd.}$$

The distance between L and F can be calculated using the generalized sine rule for arbitrary triangles:

$$\frac{\sin \hat{F}}{|LL'|} = \frac{\sin \hat{L}}{|FL'|} = \frac{\sin \hat{L}'}{|LF|}$$

Therefore:

$$\begin{aligned} |LF| &= |LL'| \frac{\sin \hat{L}'}{\sin \hat{F}} \\ &= 1 \text{ m} \frac{\sin 105^\circ}{\sin 45^\circ} \\ &= 1.366\,0 \text{ m} \end{aligned}$$

The radial flux density at F amounts to:

$$\begin{aligned} E_{v,R} &= \frac{\partial \Phi_v}{\partial A} \\ &= \frac{\partial \Phi_v}{\partial \Omega} \cdot \frac{\partial \Omega}{\partial A} \\ &= I_v \cdot \frac{1}{|LF|^2} \\ &= 1339.8 \text{ lx} \end{aligned}$$

However, the incident flux of light needs to cover more area because there is an angle between the normal of the surface that is to be lit and the incident light waves. We need to take this into account. This is the so-called *observation angle* correction. The angle can be determined to be 45° . Therefore:

$$E_{v,S} = E_{v,R} \cdot \cos 45^\circ = 947.38 \text{ lx}$$

Solution 6.1.4-3: The luminous exitance of the surface equals:

$$M_v = \rho \cdot E_v = 0.23 \cdot 250 \text{ lx} = 57.7 \text{ lm/m}^2$$

Assuming perfect diffuse reflection corresponds to assuming the surface is a Lambertian radiator. Therefore:

$$L_v = \frac{M_v}{\pi} = 18.303 \text{ cd/m}^2$$

Solution 6.1.4-4: The size of the screen when compared to the viewing distance, allows assuming that the viewing angle is almost constant for the entire screen.

The luminous intensity under the given viewing angle amounts to:

$$I_v = \cos \theta \cdot L_v \cdot A = 0.54464 \cdot 350 \text{ cd/m}^2 \cdot 4 \cdot 6.4516 \times 10^{-4} \text{ m}^2 = 0.49193 \text{ cd}$$

The flux density can be calculated as:

$$\begin{aligned} E_{v,R} &= \frac{\partial \Phi}{\partial A} \\ &= \frac{\partial \Phi}{\partial \Omega} \cdot \frac{\partial \Omega}{\partial A} \\ &= I_v \cdot \frac{1}{R^2} \\ &= 218.64 \text{ mlx} \end{aligned}$$

This value confirms what we might have expected: a cellphone screen is not a good lighting device.

Solution 6.3.9.5-1: We start by calculating C_{CMY} :

$$C_{CMY} = 1 - C_{RGB} = \begin{bmatrix} 0.7 \\ 0.3 \\ 0.8 \end{bmatrix}$$

Next, we derive C_{CMYK} :

$$C_{CMYK} = \begin{bmatrix} C - \min(C_{CMY}) \\ M - \min(C_{CMY}) \\ Y - \min(C_{CMY}) \\ \min(C_{CMY}) \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.0 \\ 0.5 \\ 0.3 \end{bmatrix}$$

Finally, we convert to C_{HSI} :

$$I = \frac{R + G + B}{3} = 0.4$$

$$S = 1 - \frac{\min(R, G, B)}{I} = 1 - \frac{0.2}{0.4} = 0.5$$

$$H = \arccos\left(\frac{R - 0.5G - 0.5B}{\sqrt{(R - G)^2 + (R - B)(G - B)}}\right) = 2\pi - 1.2373 = 1.9043$$

Solution 6.3.9.5-2: They all result in the same RGB color vector, i.e.:

$$C_{RGB} = \begin{bmatrix} 0.3 \\ 0.7 \\ 0.2 \end{bmatrix}.$$

Solution 6.3.9.5-3: Using MATLAB you should be able to check the results yourself.

Mathematical Morphology

Solution 7.2.3-1: The $N_4(p)$ (left), $N_8(p)$ (middle) and $N_D(p)$ (right) neighborhoods have been indicated in gray below. Note that the neighborhood concept is not dependent on the pixels' intensity value.

0	0	0	0	0	0	1
1	0	1	1	0	0	1
0	0	1	1	0	1	0
0	0	0	0	1	0	0
0	1	1	0	0	1	1

0	0	0	0	0	0	1
1	0	1	1	0	0	1
0	0	1	1	0	1	0
0	0	0	0	1	0	0
0	1	1	0	0	1	1

0	0	0	0	0	0	1
1	0	1	1	0	0	1
0	0	1	1	0	1	0
0	0	0	0	1	0	0
0	1	1	0	0	1	1

Solution 7.2.3-2: The 4-V-adjacency (left), 8-V-adjacency (middle) and m-V-adjacency have been indicated in gray below. Note that the adjacency concept *is* dependent on the pixels' intensity value.

0	0	0	0	0	0	1
1	0	1	1	0	0	1
0	0	1	1	0	1	0
0	0	0	0	1	0	0
0	1	1	0	0	1	1

0	0	0	0	0	0	1
1	0	1	1	0	0	1
0	0	1	1	0	1	0
0	0	0	0	1	0	0
0	1	1	0	0	1	1

0	0	0	0	0	0	1
1	0	1	1	0	0	1
0	0	1	1	0	1	0
0	0	0	0	1	0	0
0	1	1	0	0	1	1

Solution 7.2.3-3: The 4-V-adjacency (left), 8-V-adjacency (middle) and m-V-adjacency have been indicated in gray below. Note that the adjacency concept *is* dependent on the pixels' intensity value.

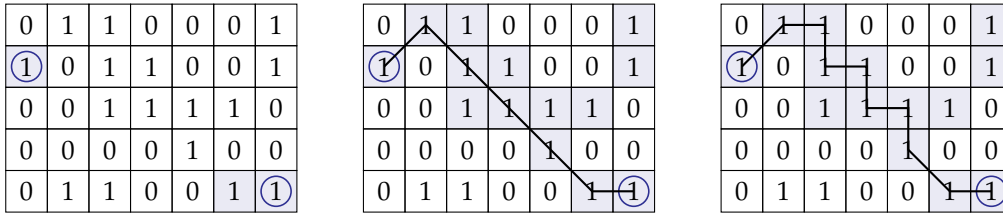
1	2	7	0	6	5	6
3	0	3	4	7	0	4
0	1	2	6	1	5	1
5	2	4	0	6	3	2
0	0	3	7	4	3	1

1	2	7	0	6	5	6
3	0	3	4	7	0	4
0	1	2	6	1	5	1
5	2	4	0	6	3	2
0	0	3	7	4	3	1

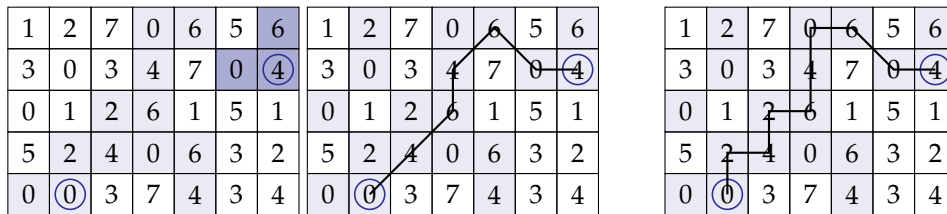
1	2	7	0	6	5	6
3	0	3	4	7	0	4
0	1	2	6	1	5	1
5	2	4	0	6	3	2
0	0	3	7	4	3	1

Solution 7.2.3-4: On the left, you can find the result for the 4-V-path. There is no path between the given pixels, therefore also no shortest path. The pixels do not reside in the same component. The separate components have been indicated.

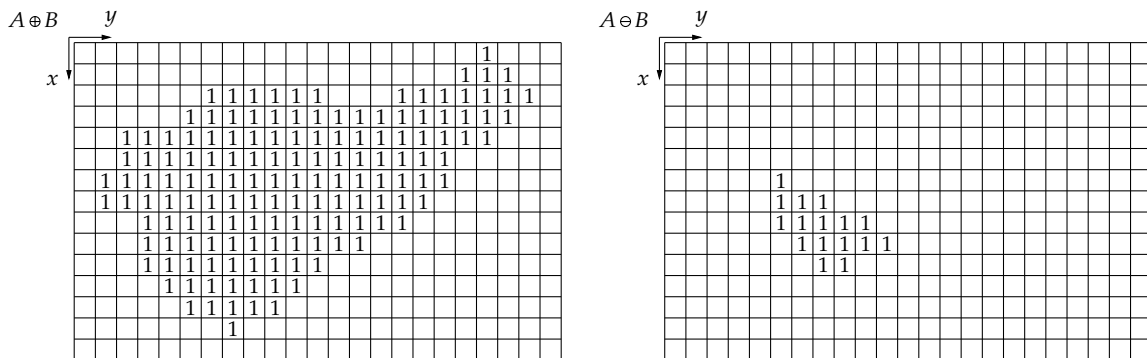
In the middle, you can find the result for the 8-V-path. The shortest 8-V-path has been indicated. Its length is 6. The component has been indicated in gray.
 On the right, you can find the result for the m-V-path. A shortest m-V-path has been indicated. The path has length 9. The component has been indicated in gray.



Solution 7.2.3-5: On the left, you can find the result for the 4-V-path. There is no path between the given pixels, therefore also no shortest path. The pixels do not reside in the same component. The separate components have been indicated.
 In the middle, you can find the result for the 8-V-path. The shortest 8-V-path has been indicated. Its length is 6. The component has been indicated in gray.
 On the right, you can find the result for the m-V-path. A shortest m-V-path has been indicated. The path has length 9. The component has been indicated in gray.

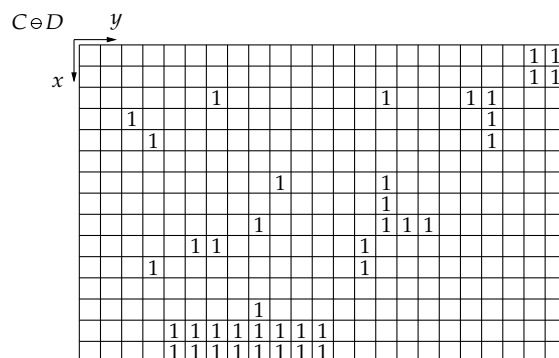
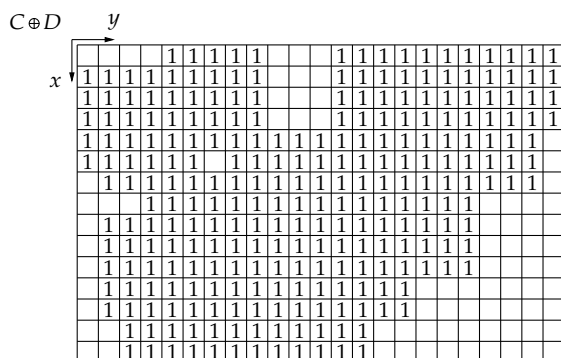


Solution 7.3.1.3-1: The solution can be found below:



Solution 7.3.1.3-2: The answer is in both cases: no. The sequence of erosion followed by dilation or vice versa gives rise to two new derived operations: opening and closing.

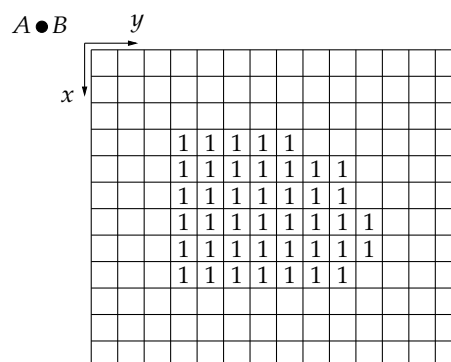
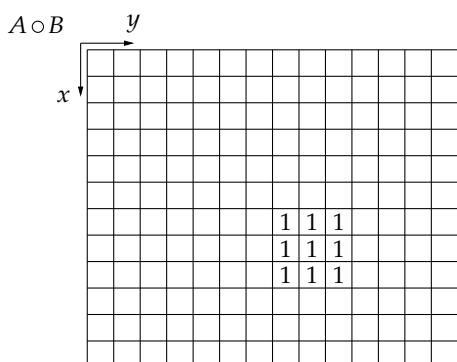
Solution 7.3.1.3-3: The solution can be found below:



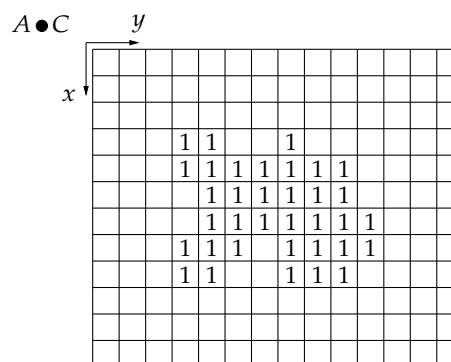
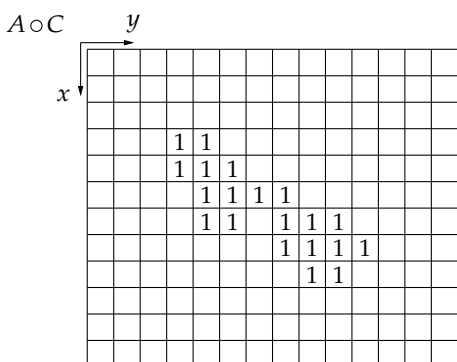
Solution 7.3.1.3-4: The actual solutions can be found in the solutions of the previous exercises. The use of the functions will be along the lines of

```
B = [ 0 0 0 0 0 0 0 1;
      0 0 0 0 0 1 0;
      0 0 1 1 1 0 0;
      0 0 1 0 0 0 0;
      0 0 0 0 0 0 0 ];
se = strel('arbitrary', B);
A = zeros(15,23); % fill out your specific data
C = imdilate(A, se);
D = imerode(A, se);
```

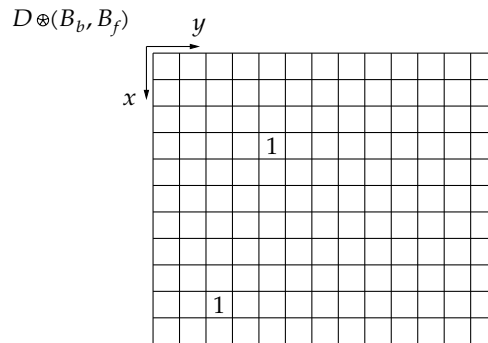
Solution 7.3.2.5-1: The solutions can be found below:



Solution 7.3.2.5-2: The solutions can be found below:



Solution 7.3.2.5-3: The solution can be found below:

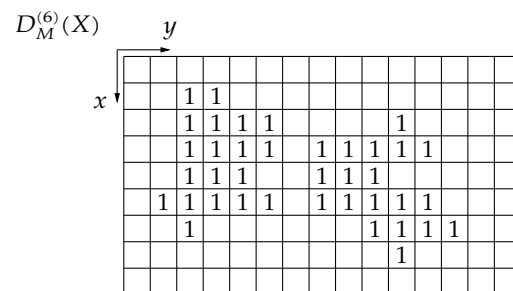
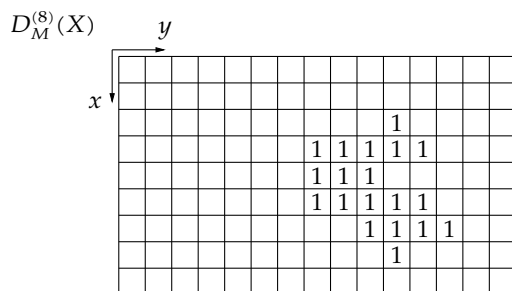


Solution 7.3.2.5-4: Show your solution to the lecturer. He will be happy to comment on it.

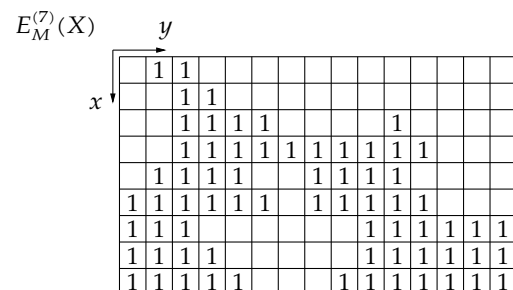
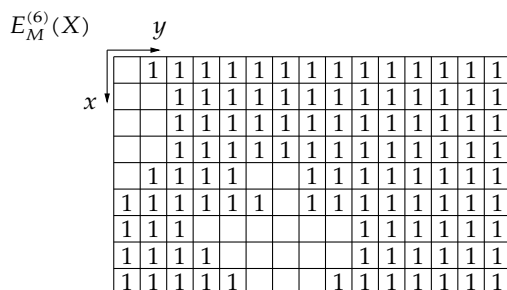
Solution 7.3.2.5-5: Show your solution to the lecturer. He will be happy to comment on it.

Solution 7.3.2.5-6: Show your solution to the lecturer. He will be happy to comment on it.

Solution 7.3.3.4-1: The result with a disc-shaped structuring element can be found on the left, with a square-shaped structuring element on the right.

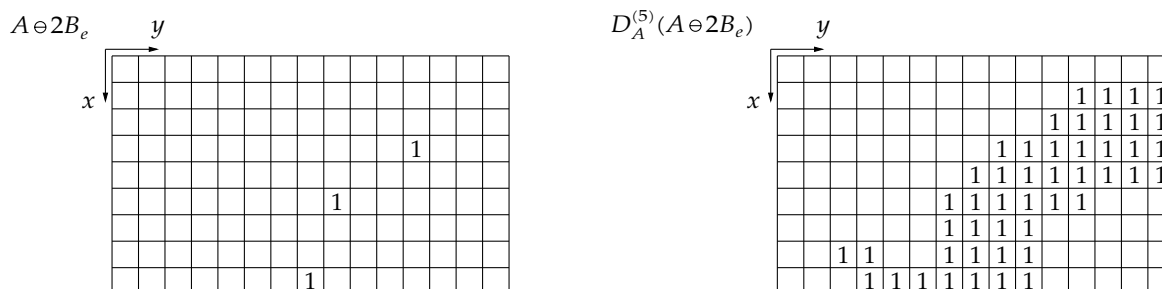


Solution 7.3.3.4-2: The result with a disc-shaped structuring element can be found on the left, with a square-shaped structuring element on the right.

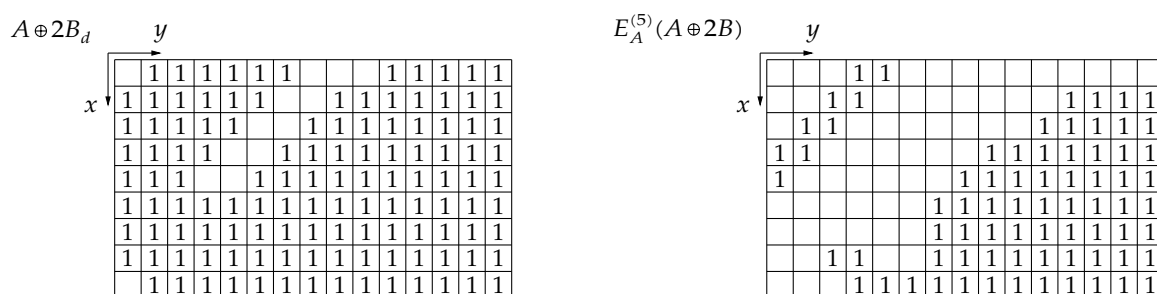


It's easier to understand this exercise if you focus on the background pixels rather than on the foreground pixels.

Solution 7.3.3.4-3: The intermediate result after two erosions can be found below on the left. The final result after reconstruction by dilation can be found below on the right. Note how the fine features have been removed, while the fat features are fully intact.



Solution 7.3.3.4-4: The intermediate result after two dilations can be found below on the left. The final result after reconstruction by erosion can be found below on the right. Note how the fine background features have been removed, while the fat background features are fully intact.



Again, it's easier to understand this exercise if you focus on the background pixels rather than on the foreground pixels.

Solution 7.3.3.4-5: If you doubt your result, send it to the lecturer. He will be happy to check it and comment on it.

Solution 7.3.3.4-6: Send your solution to the lecturer. He will be happy to check your solution and comment on it.

Solution 7.3.3.4-7: Send your solution to the lecturer. He will be happy to check your solution and comment on it.

Solution 7.3.3.4-8: Send your solution to the lecturer. He will be happy to check your solution and comment on it.

Solution 7.3.3.4-9: Send your solution to the lecturer. He will be happy to check your solution and comment on it.

Solution 7.4.1.4-1: The solution can be found below. Common mistakes: (a) you have forgotten to reflect the structuring element before applying the dilation; (b) you continued to work with a reflected structuring element when calculating the erosion.

	y									
x	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.9	0.9	0.9
	0.8	0.8	0.8	0.7	0.8	0.8	0.8	0.8	0.9	0.9
	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.9	0.9	0.9
	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
	0.8	0.8	0.8	0.8	0.9	0.9	0.9	0.9	0.9	0.8
	0.7	0.8	0.8	0.8	0.9	1.0	1.0	1.0	1.0	0.8
	1.0	1.0	1.0	0.9	0.9	1.0	1.0	1.0	1.0	0.9
	1.0	1.0	1.0	0.9	0.9	1.0	1.0	1.0	1.0	0.9
	1.0	1.0	1.0	0.9	0.9	1.0	1.0	1.0	1.0	0.9
	1.0	1.0	1.0	0.9	0.9	0.9	0.9	0.9	0.9	0.9

	y									
x	0.3	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.1
	0.3	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.1
	0.2	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.1
	0.2	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.1
	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.3
	0.2	0.0	0.0	0.0	0.0	0.1	0.1	0.2	0.2	0.3
	0.4	0.0	0.0	0.0	0.0	0.1	0.1	0.2	0.2	0.2
	0.4	0.0	0.0	0.0	0.0	0.1	0.1	0.1	0.2	0.2
	0.4	0.0	0.0	0.0	0.0	0.1	0.1	0.1	0.2	0.2
	0.6	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2

Solution 7.4.1.4-2: The answer is in both cases: no. The sequence of erosion followed by dilation or vice versa gives rise to two new derived operations: opening and closing.

Solution 7.4.1.4-3: The solution can be found below. Common mistakes: (a) you have forgotten to reflect the structuring element before applying the dilation; (b) you continued to work with a reflected structuring element when calculating the erosion.

	y									
x	0.8	0.8	0.8	0.7	0.7	0.7	0.7	0.7	0.8	0.8
	0.8	0.8	0.6	0.6	0.9	0.9	0.9	0.5	0.7	0.8
	0.4	0.9	0.9	0.9	0.9	0.9	0.7	0.7	0.7	0.7
	0.4	0.9	1.0	1.0	1.0	0.9	1.0	1.0	1.0	0.7
	0.4	0.9	1.0	1.0	0.8	0.9	1.0	1.0	0.9	0.7
	0.4	0.5	1.0	1.0	1.0	0.9	0.9	1.0	1.0	0.6
	0.4	0.5	1.0	1.0	0.9	0.9	0.8	0.9	0.9	0.5
	0.5	0.5	0.8	1.0	1.0	0.9	0.9	0.8	0.8	0.5
	0.7	0.7	0.7	0.9	0.9	0.9	0.6	0.8	0.8	0.8
	0.7	0.5	0.7	0.7	0.9	0.9	0.9	0.8	0.8	0.5

	y									
x	0.1	0.1	0.1	0.1	0.1	0.3	0.2	0.0	0.0	0.2
	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.0	0.0
	0.1	0.1	0.0	0.0	0.2	0.3	0.2	0.0	0.0	0.0
	0.0	0.0	0.1	0.0	0.0	0.2	0.2	0.2	0.1	0.1
	0.2	0.0	0.0	0.0	0.0	0.2	0.2	0.2	0.2	0.1
	0.0	0.0	0.0	0.2	0.2	0.2	0.2	0.2	0.1	0.1
	0.1	0.0	0.0	0.2	0.2	0.2	0.3	0.3	0.1	0.1
	0.0	0.0	0.0	0.0	0.4	0.0	0.0	0.2	0.1	0.1
	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.1
	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1

Solution 7.4.1.4-4: The actual solutions can be found in the solutions of the previous exercises. The use of the functions will be along the lines of

```
f = [ 0.1 0.3 0.2 0.4 0.7 0.8 1.0;
      0.3 0.4 0.3 0.2 0.1 0.9 0.0;
      0.7 0.3 1.0 0.4 0.1 0.7 0.1
      0.2 0.1 0.3 1.0 0.9 0.7 0.2;
      0.0 0.4 0.6 0.9 0.1 0.3 0.0 ];
g = [ 1 1 1; 0 1 0; 1 1 1 ];
seg = strel( 'arbitrary', g );
fdilated = imdilate( f, seg );
feroded = imerode( f, seg );
```

Solution 7.4.2.4-1: The solution can be found below.

$$(f \circ g)[x, y]$$

	y										
x		0.9	0.9	0.8	0.3	0.1	0.3	0.4	0.5	0.3	0.1
		0.9	0.8	0.7	0.0	0.3	0.5	0.8	0.7	0.5	0.2
		0.8	0.7	0.5	0.1	0.5	0.8	0.8	0.8	0.7	0.2
		0.7	0.5	0.1	0.0	0.4	0.7	0.8	0.7	0.2	0.0
		0.5	0.1	0.0	0.1	0.3	0.4	0.7	0.1	0.0	0.1
		0.1	0.1	0.0	0.1	0.1	0.3	0.2	0.0	0.1	0.1
		0.1	0.0	0.0	0.0	0.1	0.2	0.6	0.2	0.1	0.1
		0.0	0.1	0.0	0.1	0.2	0.6	0.9	0.6	0.1	0.1
		0.1	0.1	0.1	0.0	0.3	0.9	0.9	0.9	0.4	0.1
		0.1	0.1	0.0	0.3	0.8	0.9	0.9	0.9	0.8	0.1

$$\hat{T}_g(f)$$

	y										
x		0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.1	0.1	0.0	0.1	0.5	0.1	0.3	0.3	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.2	0.1	0.0
		0.0	0.0	0.2	0.0	0.0	0.3	0.0	0.0	0.3	0.0
		0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.4
		0.4	0.0	0.0	0.1	0.0	0.5	0.4	0.0	0.0	0.6
		0.2	0.0	0.1	0.0	0.0	0.7	0.2	0.0	0.0	0.2
		0.0	0.0	0.0	0.0	0.0	0.3	0.1	0.0	0.0	0.1
		0.2	0.1	0.2	0.0	0.0	0.0	0.1	0.0	0.0	0.2
		0.0	0.0	0.0	0.2	0.0	0.0	0.1	0.0	0.0	0.0

$$(f \bullet g)[x, y]$$

	y										
x		1.0	0.9	0.8	0.4	0.4	0.4	0.9	0.8	0.3	0.3
		0.9	0.9	0.8	0.5	0.4	1.0	0.9	1.0	0.8	0.3
		0.8	0.7	0.5	0.4	0.5	0.8	1.0	1.0	0.8	0.5
		0.7	0.5	0.3	0.3	0.4	1.0	0.9	0.7	0.5	0.5
		0.5	0.3	0.2	0.2	0.3	0.9	0.7	0.5	0.5	0.5
		0.5	0.2	0.1	0.2	0.2	0.8	0.6	0.6	0.3	0.7
		0.3	0.1	0.1	0.1	0.2	0.9	0.8	0.3	0.3	0.3
		0.2	0.2	0.1	0.2	0.2	0.9	1.0	0.6	0.3	0.3
		0.3	0.2	0.3	0.2	0.5	0.9	1.0	0.9	0.4	0.3
		0.2	0.2	0.2	0.5	0.8	0.9	1.0	0.9	0.8	0.4

$$\hat{B}_g(f)$$

	y										
x		0.0	0.0	0.0	0.1	0.3	0.1	0.5	0.3	0.0	0.2
		0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.1
		0.0	0.0	0.0	0.3	0.0	0.0	0.1	0.0	0.0	0.3
		0.0	0.0	0.0	0.3	0.0	0.0	0.1	0.0	0.0	0.5
		0.0	0.2	0.2	0.1	0.0	0.0	0.0	0.4	0.5	0.0
		0.0	0.1	0.1	0.0	0.1	0.0	0.0	0.6	0.2	0.0
		0.0	0.1	0.0	0.1	0.1	0.0	0.0	0.1	0.2	0.0
		0.2	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.2	0.1
		0.0	0.0	0.0	0.2	0.2	0.0	0.0	0.0	0.0	0.0
		0.1	0.1	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.3

Solution 7.4.2.4-2: The solution can be found below.

$$(f \circ g)[x, y]$$

	y										
x		0.0	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
		0.0	0.1	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
		0.0	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
		0.0	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
		0.0	0.0	0.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0
		0.1	0.1	0.0	0.0	0.0	0.0	0.2	0.2	0.2	0.2
		0.1	0.1	0.0	0.0	0.0	0.0	0.2	0.2	0.2	0.2
		0.1	0.1	0.0	0.0	0.0	0.0	0.2	0.2	0.2	0.2
		0.1	0.1	0.0	0.1	0.1	0.1	0.0	0.0	0.2	0.2
		0.0	0.0	0.0	0.1	0.1	0.1	0.0	0.0	0.0	0.0

$$\hat{T}_g(f)$$

	y										
x		0.2	0.9	0.1	0.0	0.1	0.9	0.0	0.1	0.0	0.1
		0.0	0.1	0.9	0.1	0.9	0.0	0.0	0.1	0.0	0.0
		0.1	0.0	0.1	0.9	0.9	0.1	0.0	0.0	0.1	0.0
		0.1	0.1	0.9	0.0	0.1	0.9	0.2	0.0	0.2	0.1
		0.0	0.1	0.9	0.0	0.0	0.1	1.0	0.2	0.0	0.1
		0.0	0.9	0.1	0.0	0.0	0.0	0.0	0.8	0.0	0.8
		0.1	0.9	0.0	0.1	0.1	0.0	0.0	0.8	0.8	0.0
		0.9	0.1	0.1	0.2	0.0	0.2	0.8	0.0	0.8	0.0
		0.0	0.0	0.0	0.1	0.7	0.9	0.1	0.0	0.0	0.8
		0.0	0.1	0.2	0.9	0.1	0.0	0.0	0.1	0.0	0.1

$$(f \bullet g)[x, y]$$

	y										
x		1.0	1.0	1.0	1.0	1.0	1.0	0.1	0.1	0.1	0.1
		0.2	0.2	1.0	1.0	1.0	1.0	0.1	0.1	0.1	0.1
		0.1	0.1	1.0	1.0	1.0	1.0	0.1	0.1	0.1	0.1
		0.1	0.1	1.0	1.0	1.0	1.0	0.2	0.2	0.2	0.2
		0.1	0.1	1.0	0.2	0.2	0.2	1.0	0.2	0.2	0.2
		1.0	1.0	0.2	0.2	0.2	0.2	1.0	1.0	1.0	1.0
		1.0	1.0	0.2	0.2	0.2	0.2	1.0	1.0	1.0	1.0
		1.0	1.0	0.2	0.2	0.2	0.2	1.0	1.0	1.0	1.0
		0.1	0.1	0.2	0.8	0.8	1.0	0.2	0.2	0.2	1.0
		0.1	0.1	0.2	1.0	1.0	1.0	0.2	0.2	0.2	1.0

$$\hat{B}_g(f)$$

	y										
x		0.8	0.0	0.8	0.9	0.8	0.0	0.1	0.0	0.1	0.0
		0.2	0.0	0.0	0.8	0.0	0.9	0.1	0.0	0.1	0.1
		0.0	0.1	0.8	0.0	0.0	0.8	0.1	0.1	0.0	0.1
		0.0	0.0	0.0	0.9	0.8	0.0	0.0	0.2	0.0	0.1
		0.1	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.2	0.1
		0.9	0.0	0.1	0.2	0.2	0.2	0.8	0.0	0.8	0.0
		0.8	0.0	0.2	0.1	0.1	0.2	0.8	0.0	0.0	0.8
		0.0	0.8	0.1	0.0	0.2	0.0	0.0	0.8	0.0	0.8
		0.0	0.0	0.2	0.6	0.0	0.0	0.1	0.2	0.0	0.0
		0.1	0.0	0.0	0.0	0.8	0.9	0.2	0.1	0.2	0.9

Solution 7.4.2.4-3: The actual solutions can be found in the solutions of the previous exercises.

Solution 7.4.3.5-1: The solution can be found below. The reconstruction converges after 9 iterations.

$$D_f^{(9)}(h)$$

0.0	0.1	0.3	0.1	0.0	0.0	0.0	0.0	0.0	0.0
0.3	0.7	0.9	0.7	0.2	0.0	0.0	0.1	0.0	0.2
0.2	0.9	1.0	0.9	0.0	0.0	0.2	0.2	0.2	0.2
0.1	0.7	0.8	0.8	0.3	0.0	0.2	0.2	0.2	0.2
0.0	0.0	0.1	0.2	0.0	0.0	0.2	0.2	0.1	0.2
0.1	0.0	0.0	0.0	0.2	0.2	0.2	0.1	0.0	0.1
0.0	0.2	0.2	0.1	0.0	0.2	0.2	0.2	0.1	0.2
0.2	0.2	0.2	0.2	0.2	0.1	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2	0.0	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2	0.0	0.0	0.1	0.2	0.2

Solution 7.4.3.5-2: The solution can be found below. The reconstruction converges after 13 iterations.

$$E_f^{(13)}(h)$$

0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.7	0.9	0.7	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.9	1.0	0.9	0.5	0.5	0.5	0.5	0.5	1.0
0.5	0.7	0.8	0.8	0.5	0.5	0.7	0.6	0.5	0.9
0.5	0.5	0.5	0.5	0.5	0.5	0.8	0.5	0.1	0.4
0.5	0.5	0.5	0.5	0.5	0.5	0.7	0.1	0.0	0.1
0.5	0.5	0.5	0.5	0.5	0.5	0.8	0.3	0.1	0.3
0.6	1.0	0.7	0.5	0.5	0.5	0.9	0.7	0.5	0.8
0.9	1.0	0.9	0.7	0.5	0.5	0.5	0.9	1.0	0.7
1.0	1.0	0.9	0.8	0.5	0.5	0.5	0.5	0.5	0.5

Solution 7.4.3.5-3: The solution can be found below. The intermediate result after two erosions can be found on the left. The final result (after 9 geodesic dilations) can be found on the right.

$$(f \ominus 2g_e)[x, y]$$

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.0
0.0	0.3	0.2	0.0	0.0	0.0	0.0	0.1	0.2	0.0
0.6	0.8	0.3	0.0	0.0	0.0	0.0	0.0	0.1	0.0

$$O_R^{(2)}(f)$$

0.0	0.1	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0
0.2	0.2	0.2	0.2	0.2	0.0	0.0	0.1	0.0	0.2
0.2	0.2	0.2	0.2	0.0	0.0	0.2	0.2	0.2	0.2
0.1	0.2	0.2	0.2	0.2	0.0	0.2	0.2	0.2	0.2
0.0	0.0	0.1	0.2	0.0	0.0	0.2	0.2	0.1	0.2
0.1	0.0	0.0	0.0	0.2	0.2	0.2	0.1	0.0	0.1
0.0	0.3	0.2	0.1	0.0	0.2	0.2	0.2	0.1	0.2
0.6	0.8	0.7	0.4	0.2	0.1	0.2	0.2	0.2	0.2
0.8	0.8	0.8	0.7	0.4	0.0	0.2	0.2	0.2	0.2
0.8	0.8	0.8	0.8	0.3	0.0	0.0	0.1	0.2	0.2

Solution 7.4.3.5-4: The solution can be found below. The intermediate result after two dilations can be found on the left. The final result (after 13 geodesic erosions) can be found on the right.

		y									
	x	0.7	0.9	1.0	0.9	0.7	0.2	0.2	0.3	0.5	1.0
		0.9	1.0	1.0	1.0	0.9	0.7	0.7	0.6	1.0	1.0
		1.0	1.0	1.0	1.0	1.0	0.9	0.8	1.0	1.0	1.0
		0.9	1.0	1.0	1.0	0.9	0.8	0.8	0.9	1.0	1.0
		0.7	0.9	1.0	0.9	0.8	0.8	0.8	0.8	0.9	1.0
		0.6	1.0	0.8	0.8	0.7	0.8	0.9	0.8	0.7	0.9
		1.0	1.0	1.0	0.7	0.8	0.9	0.9	0.9	1.0	0.8
		1.0	1.0	1.0	1.0	0.9	0.9	0.9	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	0.9	0.9	1.0	1.0	1.0	1.0
		1.0	1.0	1.0	1.0	0.9	0.8	0.9	1.0	1.0	1.0

		y									
	x	0.3	0.3	0.3	0.2	0.2	0.2	0.2	0.2	0.2	0.2
		0.3	0.7	0.9	0.7	0.2	0.2	0.2	0.2	0.2	0.3
		0.2	0.9	1.0	0.9	0.2	0.2	0.2	0.3	0.5	1.0
		0.2	0.7	0.8	0.8	0.3	0.2	0.7	0.6	0.5	0.9
		0.2	0.2	0.2	0.2	0.2	0.2	0.8	0.5	0.5	0.5
		0.2	0.2	0.2	0.2	0.2	0.2	0.3	0.7	0.5	0.5
		0.2	0.3	0.2	0.2	0.2	0.2	0.8	0.5	0.5	0.5
		0.6	1.0	0.7	0.4	0.2	0.2	0.9	0.7	0.5	0.8
		0.9	1.0	0.9	0.7	0.4	0.2	0.5	0.9	1.0	0.7
		1.0	1.0	0.9	0.8	0.3	0.2	0.2	0.2	0.2	0.3

Solution 7.4.3.5-5: The solution can be found below.

		y									
	x	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.1	0.5	0.7	0.5	0.0	0.0	0.0	0.0	0.0	0.1
		0.0	0.7	0.8	0.7	0.0	0.0	0.0	0.1	0.3	0.8
		0.0	0.5	0.6	0.6	0.1	0.0	0.5	0.4	0.3	0.7
		0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.3	0.0	0.2
		0.0	0.0	0.0	0.0	0.0	0.1	0.5	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.1	0.0	0.1
		0.0	0.2	0.0	0.0	0.0	0.0	0.7	0.5	0.3	0.6
		0.1	0.2	0.1	0.0	0.0	0.0	0.3	0.7	0.8	0.5
		0.2	0.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1

Solution 7.4.3.5-6: The solution can be found below.

		y									
	x	0.3	0.2	0.0	0.1	0.2	0.2	0.2	0.2	0.2	0.2
		0.0	0.0	0.0	0.0	0.0	0.2	0.2	0.1	0.2	0.0
		0.0	0.0	0.0	0.0	0.2	0.2	0.0	0.0	0.0	0.0
		0.1	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0
		0.2	0.2	0.1	0.0	0.2	0.2	0.0	0.0	0.4	0.1
		0.1	0.2	0.2	0.2	0.0	0.0	0.0	0.4	0.5	0.4
		0.2	0.0	0.0	0.1	0.2	0.0	0.0	0.2	0.4	0.2
		0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.2	0.2	0.1	0.0	0.0

Solution 7.4.3.5-7: The actual solutions can be found in the solutions of the previous exercises.

Image Segmentation

Solution 8.2.1-1: The histogram corresponds to a 20×20 image, therefore it must contain a total pixel count of 400. Because of this, the missing value is equal to 12.

In a spreadsheet the class averages can be conveniently calculated as:

$\text{sumproduct}(\langle l \rangle, \langle n \rangle) / \text{sum}(\langle n \rangle)$ where $\langle l \rangle$ and $\langle n \rangle$ denote the proper intensity and count ranges.

In MATLAB/OCTAVE the following script will mimic the calculations you made in the spreadsheet, realizing the ranges using a mask vector (containing ones below the threshold and zeros above the threshold).

```
l = 0:15
n = [ 4 12 20 21 20 19 13 12 13 8 26 38 46 65 49 34 ]
lt = 8
mask = l < lt
mu1 = ( ( n .* mask ) * l' ) / sum( n .* mask )
mu2 = ( ( n .* (1-mask) ) * l' ) / sum( n .* (1-mask) )
lt = 0.5 * ( mu1 + mu2 )
%% => lt = 8.05

mask = l < lt
mu1 = ( ( n .* mask ) * l' ) / sum( n .* mask )
mu2 = ( ( n .* (1-mask) ) * l' ) / sum( n .* (1-mask) )
lt = 0.5 * ( mu1 + mu2 )
%% => lt = 8.36
```

Conclusion: $l_T = 8.36$ (or equally good: $l_T = 9$)

Solution 8.2.1-2: A viable algorithm call could be:

```
n = [ 139 104 94 79 33 20 13 6 18 27 27 11 14 10 3 2 ];
lt = avgmeanthreshold( n );
```

using as algorithm:

```
function lt = avgmeanthreshold( n )
N = length(n);
l = 0:(N-1);
lo = -1;
lt = N/2;
while ( abs( lt - lo ) >= 1 )
    mask = l < lt;
    mu1 = ( ( n .* mask ) * l' ) / sum( n .* mask );
    mu2 = ( ( n .* (1-mask) ) * l' ) / sum( n .* (1-mask) );
```

```

    lo = lt;
    lt = ceil( (mu1+mu2) / 2 );
end
end

```

While debugging you may want to remove some of the trailing semicolons to be able to inspect intermediate values.

Note: the stopping criterion is not limit-cycle proof (i.e. the obtained value of l_T may go one up every odd time the while loop is called and go one unit down every even call). However, we will not elaborate on that any further.

Solution 8.2.1-3: Using the algorithm composed for the previous exercise, we can very simply treat this problem as:

```

n = (0:31) * 5;
lt = avgmeanthreshold( n );

```

After three iterations, the algorithm finds: $l_T = 19$.

Solution 8.2.2-1: The frequency table together with the key algorithmic values P_1 , M_1 and σ_{12}^2 can be found to be:

l	n_l	$f(l)$	P_1	M_1	σ_{12}^2
0	4	0.010000	0.010000	0.000000	0.959735
1	12	0.030000	0.040000	0.030000	3.373125
2	20	0.050000	0.090000	0.130000	6.818314
3	21	0.052500	0.142500	0.287500	9.929670
4	20	0.050000	0.192500	0.487500	12.409800
5	19	0.047500	0.240000	0.725000	14.288856
6	13	0.032500	0.272500	0.920000	15.205366
7	12	0.030000	0.302500	1.130000	15.675223
8	13	0.032500	0.335000	1.390000	15.788002
9	8	0.020000	0.355000	1.570000	15.606378
10	26	0.065000	0.420000	2.220000	14.415799
11	38	0.095000	0.515000	3.265000	12.330671
12	46	0.115000	0.630000	4.645000	9.600136
13	65	0.162500	0.792500	6.757500	5.691008
14	49	0.122500	0.915000	8.472500	2.562890
15	34	0.085000	1.000000	9.747500	

Note that the last value in the column of M_1 equals μ_G . The largest interclass variance occurs at $l = 8$, therefore $l_T = 9$. The obtained separability equals

$$\eta(l_T) = \frac{\sigma_*^2}{\sigma_G^2} = 0.82$$

given that $\sigma_*^2 = \sigma_{12}^2(l = 8) = 15.788002$, and

$$\sigma_G^2 = \sum_{i=0}^{15} f(l)(l - \mu_G)^2 = 19.168744.$$

The obtained value of $\eta(l_T) = 0.82$ is judged to be good (> 0.6).

Solution 8.2.2-2: For your convenience the full spreadsheet table has been printed below:

l	n_l	$f(l)$	P_1	M_1	σ_{12}^2
0	139	0.231667	0.231667	0.000000	3.448065
1	104	0.173333	0.405000	0.173333	5.938351
2	94	0.156667	0.561667	0.486667	8.106221
3	79	0.131667	0.693333	0.881667	10.065923
4	33	0.055000	0.748333	1.101667	10.842055
5	20	0.033333	0.781667	1.268333	11.078106
6	13	0.021667	0.803333	1.398333	10.999771
7	6	0.010000	0.813333	1.468333	10.826820
8	18	0.030000	0.843333	1.708333	9.897511
9	27	0.045000	0.888333	2.113333	7.997918
10	27	0.045000	0.933333	2.563333	5.649384
11	11	0.018333	0.951667	2.765000	4.465658
12	14	0.023333	0.975000	3.045000	2.607878
13	10	0.016667	0.991667	3.261667	1.020199
14	3	0.005000	0.996667	3.331667	0.451457
15	2	0.003333	1.000000	3.381667	

Otsu's algorithm has been implemented in to a MATLAB/OCTAVE algorithm below:

```
function [ lt, eta ] = otsuThresholding( n )
    %% perform Otsu Thresholding on a histogram vector n

    %% 1. Calculate the normalized histogram of the image
    L = length(n);
    l = 0 : L-1;
    N = sum(n);
    n = n / N;

    %% 2. Calculate _G and _G^2
    muG = n * l';
    sigGsqr = n * (l-muG).^2';

    %% 3. Let
    lt = 0;
    sigstarsqr = 0,
    M1 = 0;
    P1 = 0;

    %% 4. For all l = 0, 1, 2, . . . , L - 1:
    for i = 1:L-1 %% because of 1-indexed vectors! normally: l = 0:L-2
        %% 4.1. Calculate P1
        P1 = P1 + n(i);

        %% 4.2. Calculate M1
        M1 = M1 + l(i)*n(i);
    end
end
```

```

%% 4.3. Calculate sigsq
sigsq = P1 / ( 1 - P1 ) * (M1 / P1 - muG )^2;

%% 4.4. test
if ( sigsq > sigstarsq )
    lt = i; %% no + 1 because of zero-indexed vectors
    sigstarsq = sigsq;
end
end

%% 5. lt has been determined and:
eta = sigstarsq / sigGsqr;
end

```

This algorithm can be called as:

```

n = [ 139 104 94 79 33 20 13 6 18 27 27 11 14 10 3 2 ];
[ lt, eta ] = otsuThresholding( n );

```

This yields: $l_T = 6$ and $\eta(l_T) = 0.81$.

Solution 8.2.2-3: We can reuse the algorithm implemented for last exercise, to obtain a value for l_T and η .

```

n = (0:31)*5;
[ lt, eta ] = otsuThresholding( n );

```

this results in $l_T = 20$ and $\eta(l_T) = 0.72252$.

You'll notice an error occurs when executing this algorithm. Try to overcome this error by adding some extra measures in the algorithm.

Solution 8.3.2-1: For N_4 - left:

$$f(p) < 0 \Rightarrow S(p) = \max \left(\underset{q=p_N}{0}, \underset{q=p_E}{\frac{1 - (-2)}{2}}, \underset{q=p_S}{\frac{2 - (-2)}{2}}, \underset{q=p_W}{0} \right) = 4$$

For N_4 - middle:

$$f(p) > 0 \Rightarrow S(p) = 0$$

For N_4 - right:

$$f(p) = 0 \Rightarrow S(p) = \max \left(\underset{q=p_N}{0}, \underset{q=p_E}{\left| \frac{-3 - 1}{2} \right|}, \underset{q=p_S}{0}, \underset{q=p_W}{\left| \frac{1 - (-3)}{2} \right|} \right) = 2$$

For N_8 - left:

$$f(p) < 0 \Rightarrow S(p) = \max \left(\underset{q=p_N}{0}, \underset{q=p_{NE}}{\frac{2 - (-2)}{\sqrt{2}}}, \underset{q=p_E}{\frac{1 - (-2)}{2}}, \underset{q=p_{SE}}{\frac{3 - (-2)}{\sqrt{2}}}, \underset{q=p_S}{\frac{2 - (-2)}{2}}, \underset{q=p_{SW}}{\frac{2 - (-2)}{\sqrt{2}}}, \underset{q=p_W}{0}, \underset{q=p_{NW}}{0} \right) = 4$$

For N_8 - middle:

$$f(p) > 0 \Rightarrow S(p) = 0$$

For N_8 - right:

$$f(p) = 0 \Rightarrow S(p) = \max \left(\underbrace{0}_{q=\underline{p}_N}, \underbrace{0}_{q=\underline{p}_{NE}}, \underbrace{\left| \frac{-3-1}{2} \right|}_{q=\underline{p}_E}, \underbrace{\left| \frac{-2-2}{2\sqrt{2}} \right|}_{q=\underline{p}_{SE}}, \underbrace{0}_{q=\underline{p}_S}, \underbrace{0}_{q=\underline{p}_{SW}}, \underbrace{\left| \frac{1-(-3)}{2} \right|}_{q=\underline{p}_W}, \underbrace{\left| \frac{2-(-2)}{2\sqrt{2}} \right|}_{q=\underline{p}_{NW}} \right) = 2$$

Solution 8.3.2-2: For N_4 - left:

$$f(p) = 0 \Rightarrow S(p) = \max \left(\underbrace{\left| \frac{1-(-1)}{2} \right|}_{q=\underline{p}_N}, \underbrace{0}_{q=\underline{p}_E}, \underbrace{\left| \frac{-1-1}{2} \right|}_{q=\underline{p}_S}, \underbrace{0}_{q=\underline{p}_W} \right) = 1$$

For N_4 - middle:

$$f(p) > 0 \Rightarrow S(p) = 0$$

For N_4 - right:

$$f(p) < 0 \Rightarrow S(p) = \max \left(\underbrace{0}_{q=\underline{p}_N}, \underbrace{0}_{q=\underline{p}_E}, \underbrace{4-(-1)}_{q=\underline{p}_S}, \underbrace{1-(-1)}_{q=\underline{p}_W} \right) = 5$$

For N_8 - left:

$$f(p) = 0 \Rightarrow S(p) = \max \left(\underbrace{\left| \frac{1-(-1)}{2} \right|}_{q=\underline{p}_N}, \underbrace{\left| \frac{-3-4}{2\sqrt{2}} \right|}_{q=\underline{p}_{NE}}, \underbrace{0}_{q=\underline{p}_E}, \underbrace{\left| \frac{2-(-2)}{2\sqrt{2}} \right|}_{q=\underline{p}_{SE}}, \underbrace{\left| \frac{-1-1}{2} \right|}_{q=\underline{p}_S}, \underbrace{\left| \frac{4-(-3)}{2\sqrt{2}} \right|}_{q=\underline{p}_{SW}}, \underbrace{0}_{q=\underline{p}_W}, \underbrace{\left| \frac{-2-2}{2\sqrt{2}} \right|}_{q=\underline{p}_{NW}} \right) = \frac{7}{2\sqrt{2}}$$

For N_8 - middle:

$$f(p) > 0 \Rightarrow S(p) = 0$$

For N_8 - right:

$$f(p) < 0 \Rightarrow S(p) = \max \left(\underbrace{0}_{q=\underline{p}_N}, \underbrace{\frac{2-(-1)}{\sqrt{2}}}_{q=\underline{p}_{NE}}, \underbrace{0}_{q=\underline{p}_E}, \underbrace{0}_{q=\underline{p}_{SE}}, \underbrace{4-(-1)}_{q=\underline{p}_S}, \underbrace{\frac{2-(-1)}{\sqrt{2}}}_{q=\underline{p}_{SW}}, \underbrace{1-(-1)}_{q=\underline{p}_W}, \underbrace{0}_{q=\underline{p}_{NW}} \right) = 5$$

Solution 8.3.2-3:

		y				
	x	0	3	0	0	0
		0	3	0	0	0
		3	4	0	2	0
		0	0	0	0	6

Solution 8.3.2-4:

		y					
x		0	3	0	0	0	0
		0	3	0	0	$\frac{7}{2\sqrt{2}}$	0
		3	4	0	2	$\frac{7}{\sqrt{2}}$	0
		0	0	0	0	6	0

Solution 8.4.2.1-1: We need to assume border replication, i.e. we need to consider the original image to be embedded in a larger frame:

		y							
x	f_{ext}	1	1	2	8	14	13	12	12
		1	1	2	8	14	13	12	12
		3	3	7	15	3	2	0	0
		6	6	15	3	4	7	3	3
		15	15	4	1	2	8	11	11
		15	15	4	1	2	8	11	11

This allows for calculating the x-component of the gradient f^*g_x :

		y					
x	f^*g_y	9	14	1	-15	-34	-35
		23	13	-2	-21	-25	-24
		21	-5	-18	-9	16	28
		7	-4	-15	-3	7	17

and the y-component of the gradient f^*g_y :

		y					
x	f^*g_y	6	26	20	-3	-7	-4
		14	16	-3	-4	-6	-7
		2	-5	-17	-2	5	-3
		-13	-31	-15	18	17	2

To this end, the following gradient kernels were used:

		y		
x	g_x	-1	-1	-1
		0	0	0
		1	1	1

		y		
x	g_x	-1	0	1
		-1	0	1
		-1	0	1

This allows for the calculation of the magnitude of the gradient:

$$|\nabla f|$$

		y					
x		11	30	20	15	35	35
		27	21	4	21	26	25
		21	7	25	9	17	28
		15	31	21	18	18	17

A proper threshold T could be obtained using Otsu's method and was found to be 21. This leads to

$$|\nabla f| \geq 21$$

		y					
x		0	1	0	0	1	1
		1	0	0	1	1	1
		1	0	1	0	0	1
		0	1	1	0	0	0

Solution 8.4.2.1-2: We need to assume border replication, i.e. we need to consider the original image to be embedded in a larger frame:

$$f_{ext}$$

		y							
x		15	15	8	5	4	3	2	2
		15	15	8	5	4	3	2	2
		7	7	14	7	3	0	0	0
		1	1	4	13	14	12	15	15
		0	0	0	7	6	2	1	1
		0	0	0	7	6	2	1	1

This allows for calculating the x-component of the gradient f^*g_x :

$$f^*g_x$$

		y					
x		-10	0	7	-2	-6	-7
		-32	-10	14	27	32	35
		-28	-21	-11	5	6	4
		-6	-11	-18	-24	-32	-38

and the y-component of the gradient f^*g_y :

$$f * g_y$$

		y					
	$x \downarrow$	-7	-20	-19	-11	-7	-2
		3	2	-5	-10	-4	2
		10	19	5	-13	-7	2
		3	26	22	-11	-9	1

To this end, the following gradient kernels were used:

		y		
	$x \downarrow$	-1	-1	-1
		0	0	0
		1	1	1

		y		
	$x \downarrow$	-1	0	1
		-1	0	1
		-1	0	1

This allows for the calculation of the magnitude of the gradient:

$$|\vec{\nabla}f|$$

		y					
	$x \downarrow$	12	20	20	11	9	7
		32	10	15	29	32	35
		30	28	12	14	9	4
		7	28	28	26	33	38

A proper threshold T could be obtained using Otsu's method and was found to be 23. This leads to

$$|\vec{\nabla}f| > 23$$

		y					
	$x \downarrow$	0	0	0	0	0	0
		1	0	0	1	1	1
		1	1	0	0	0	0
		0	1	1	1	1	1

Solution 8.4.2.1-3: The following code will do.

```

%% uncomment following line to load image package in octave manually
%% pkg load image;

%% compose image
x = [ 1 2 8 14 13 12;
      3 7 15 3 2 0;
      6 15 3 4 7 3;
      15 4 1 2 8 11 ];

f = mat2gray( x, [0, 15] );

%% calculate gradients
gx = [ -1 -1 -1; 0 0 0; 1 1 1 ];

```

```

gy = [ -1 0 1; -1 0 1; -1 0 1 ];
fgx = imfilter( x, gx, 'corr', 'replicate' );
fgy = imfilter( x, gy, 'corr', 'replicate' );

%% calculate magnitude
m = (fgx.^2 + fgy.^2).^(1/2);

%% perform Otsu's method
Im = mat2gray( m, [-45,45] );
[ thresh, eta ] = graythresh( Im );

result = (m > ( 90*thresh - 45 ) );

```

Solution 8.4.2.2-1: Border replication has been explained in the corresponding exercise on gradient-based edge detection.

Assuming

l	→	y		
	1	1	1	
x	↓	1	-8	1
		1	1	1

we can calculate the laplacian of this image as $\nabla^2 f = f \star l$:

$\nabla^2 f$	→	y					
	12	29	1	-36	-34	-32	
x	↓	17	-3	-64	42	40	52
		20	-66	27	9	-23	18
		-40	28	27	18	-16	-26

Applying Otsu's method on $|\nabla^2 f|$, leads to $T = 33$. This results in a resulting binary edge map:

$ \nabla^2 f > 33$	→	y				
	0	0	0	1	1	0
x	↓	0	0	1	1	1
		0	1	0	0	0
		1	0	0	0	0

Solution 8.4.2.2-2: Border replication has been explained in the corresponding exercise on gradient-based edge detection.

Assuming

$$l = \begin{array}{c} \begin{array}{cc} & y \\ \begin{array}{c} \downarrow x \\ \end{array} & \begin{array}{ccc} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{array} \end{array} \end{array}$$

we can calculate the laplacian of this image as $\nabla^2 f = f \star l$:

$$\nabla^2 f = \begin{array}{c} \begin{array}{cc} & y \\ \begin{array}{c} \downarrow x \\ \end{array} & \begin{array}{cccccc} -31 & 12 & 13 & -2 & -6 & -4 \\ 9 & -52 & 9 & 34 & 53 & 49 \\ 25 & 17 & -49 & -62 & -55 & -89 \\ 6 & 32 & -6 & 15 & 41 & 41 \end{array} \end{array} \end{array}$$

Applying Otsu's method on $|\nabla^2 f|$, leads to $T = 33$. This results in a resulting binary edge map:

$$|\nabla^2 f| > 33 = \begin{array}{c} \begin{array}{cc} & y \\ \begin{array}{c} \downarrow x \\ \end{array} & \begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{array} \end{array} \end{array}$$

Solution 8.4.2.2-3: The following code will do.

```

%% uncomment following line to load image package in octave manually
%% pkg load image;

%% compose image
x = [ 1 2 8 14 13 12;
      3 7 15 3 2 0;
      6 15 3 4 7 3;
      15 4 1 2 8 11 ];

f = mat2gray( x, [0, 15] );

%% calculate laplacian
l = [ 1 1 1; 1 -8 1; 1 1 1 ];
f1 = imfilter( x, l, 'corr', 'replicate' );

%% perform Otsu's method
Im = mat2gray( abs(f1), [0,120] );
[ thresh, eta ] = graythresh( Im );

flm = ( abs(f1) > (120*thresh) );

```

Solution 8.4.3.2-1: Let's first calculate the extent of the filter kernel. We know that the Gaussian has dropped to zero at a distance of 3σ from the kernel's center. Therefore, we will approximate the Gaussian in a box with dimensions $n \times n$ with $n = 2\lceil 3\sigma \rceil + 1$. Given $\sigma = 1$, we obtain $n = 7$.

$$G_y[x, y] = G \star g_y$$

		y						
		0.0198	0.0872	0.1266	-0.0000	-0.1266	-0.0872	-0.0198
		0.1019	0.4483	0.6511	-0.0000	-0.6511	-0.4483	-0.1019
		0.2357	1.0371	1.5061	-0.0000	-1.5061	-1.0371	-0.2357
		0.2995	1.3177	1.9136	0.0000	-1.9136	-1.3177	-0.2995
		0.2357	1.0371	1.5061	-0.0000	-1.5061	-1.0371	-0.2357
		0.1019	0.4483	0.6511	0.0000	-0.6511	-0.4483	-0.1019
		0.0198	0.0872	0.1266	0.0000	-0.1266	-0.0872	-0.0198
	x							

We assumed zeros beyond the border of the Gaussian kernel.

Solution 8.4.3.2-2: Let's first calculate the extent of the filter kernel. We know that the Gaussian has dropped to zero at a distance of 3σ from the kernel's center. Therefore, we will approximate the Gaussian in a box with dimensions $n \times n$ with $n = 2[3\sigma] + 1$. Given $\sigma = 0.6$, we obtain $n = 5$.

Determining the Gaussian's kernel is as easy as evaluating a function:

$$G[x, y] = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Therefore, knowing that $[x, y] = [0, 0]$ is located at the center of the kernel, we obtain:

$$G[x, y]$$

		y				
		0.0000	0.0010	0.0039	0.0010	0.0000
		0.0010	0.0622	0.2494	0.0622	0.0010
		0.0039	0.2494	1.0000	0.2494	0.0039
		0.0010	0.0622	0.2494	0.0622	0.0010
		0.0000	0.0010	0.0039	0.0010	0.0000
	x					

Correlating it with the 4-Laplacian gradient kernel l_4 yields:

$$G_{L4}[x, y] = G \star l_4$$

	$y \rightarrow$				
	0.0019	0.0622	0.2358	0.0622	0.0019
	0.0622	0.2519	0.1308	0.2519	0.0622
	0.2358	0.1308	-3.0026	0.1308	0.2358
	0.0622	0.2519	0.1308	0.2519	0.0622
	0.0019	0.0622	0.2358	0.0622	0.0019
$x \downarrow$					

We assumed zeros beyond the border of the Gaussian kernel.

Solution 8.4.3.2-3: Let's first calculate the extent of the filter kernel. We know that the Gaussian has dropped to zero at a distance of 3σ from the kernel's center. Therefore, we will approximate the Gaussian in a box with dimensions $n \times n$ with $n = 2\lceil 3\sigma \rceil + 1$. Given $\sigma = 0.75$, we obtain $n = 7$.

Determining the Gaussian's kernel is as easy as evaluating a function:

$$G[x, y] = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Therefore, knowing that $[x, y] = [0, 0]$ is located at the center of the kernel, we obtain:

$$G[x, y]$$

	$y \rightarrow$						
	0.0000	0.0000	0.0001	0.0003	0.0001	0.0000	0.0000
	0.0000	0.0008	0.0117	0.0286	0.0117	0.0008	0.0000
	0.0001	0.0117	0.1690	0.4111	0.1690	0.0117	0.0001
	0.0003	0.0286	0.4111	1.0000	0.4111	0.0286	0.0003
	0.0001	0.0117	0.1690	0.4111	0.1690	0.0117	0.0001
	0.0000	0.0008	0.0117	0.0286	0.0117	0.0008	0.0000
	0.0000	0.0000	0.0001	0.0003	0.0001	0.0000	0.0000
$x \downarrow$							

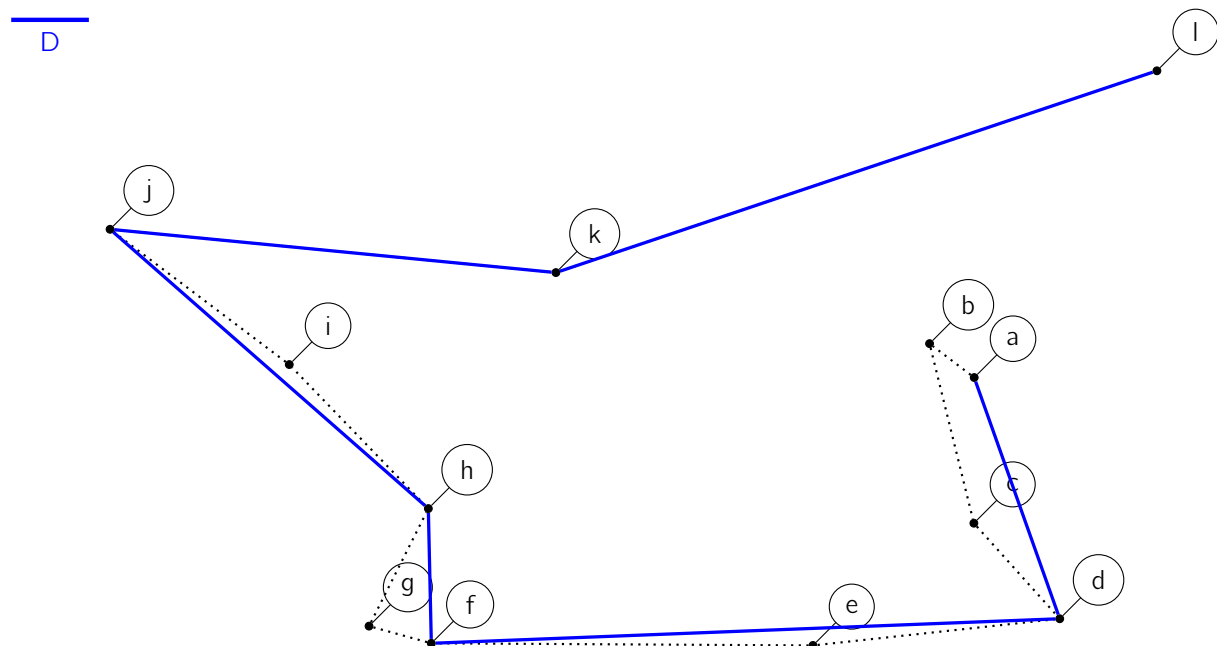
Correlating it with the 8-Laplacian gradient kernel l_8 yields:

$G_{L8}[x, y] = G * I_8$

	y						
	0.0008	0.0126	0.0404	0.0496	0.0404	0.0126	0.0008
x	0.0126	0.1863	0.5278	0.5447	0.5278	0.1863	0.0126
	0.0404	0.5278	0.5516	-1.0766	0.5516	0.5278	0.0404
	0.0496	0.5447	-1.0766	-5.6795	-1.0766	0.5447	0.0496
	0.0404	0.5278	0.5516	-1.0766	0.5516	0.5278	0.0404
	0.0126	0.1863	0.5278	0.5447	0.5278	0.1863	0.0126
	0.0008	0.0126	0.0404	0.0496	0.0404	0.0126	0.0008

We assumed zeros beyond the border of the Gaussian kernel.

Solution 8.4.4.2-1: The final result looks like:

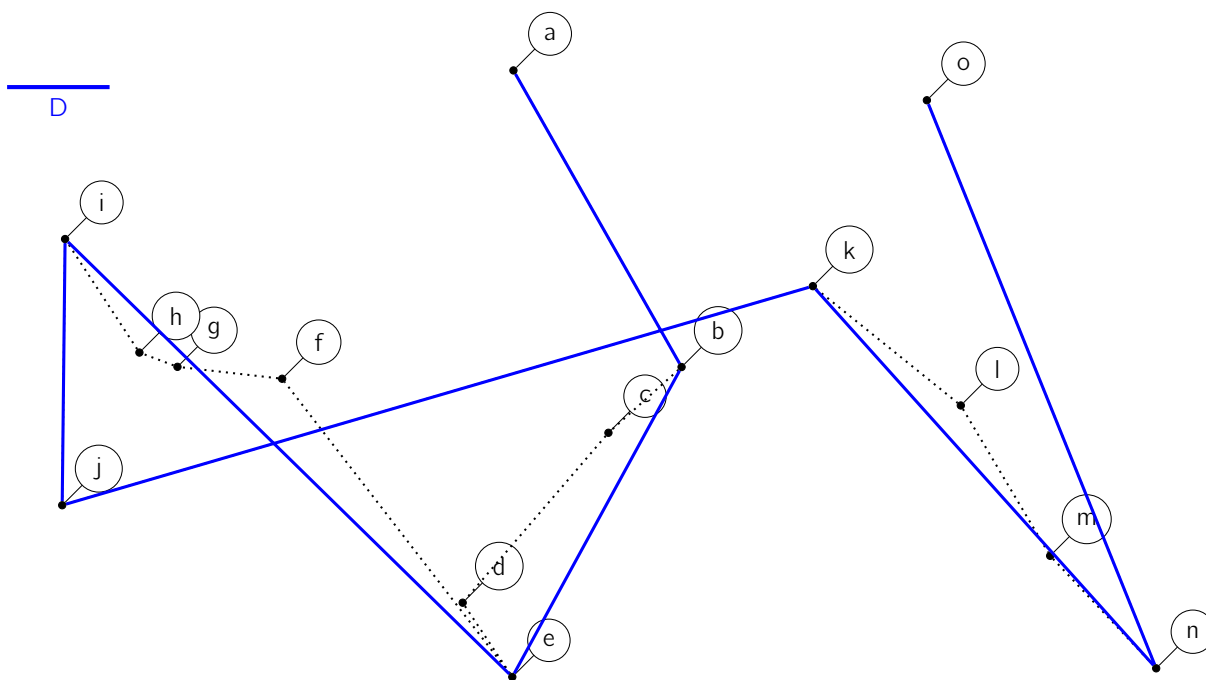


One can find all intermediate stack images below (read from left to right, then from top to bottom). The last stack image is the final one, and allows reading the entire path on the PD stack.

	j	k	j	j	k	j	f	j		
a	a	a	a	a	a	a	a	a		
TP	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP
	l	l	l	l	l	l	l	l		
	PD	PD	PD	PD	PD	PD	PD	PD	PD	PD

										a	
					f		f		d	d	
		h		h	h		h		f	f	
h	j	j		j	j		j		h	h	
f	k	f	k	k	d	k	k		j	j	
a	l	a	l	a	a	l	a	l	k	k	
TP	PD	TP	PD	TP	PD	TP	PD	TP	PD	TP	PD

Solution 8.4.4.2-2: The final result looks like:

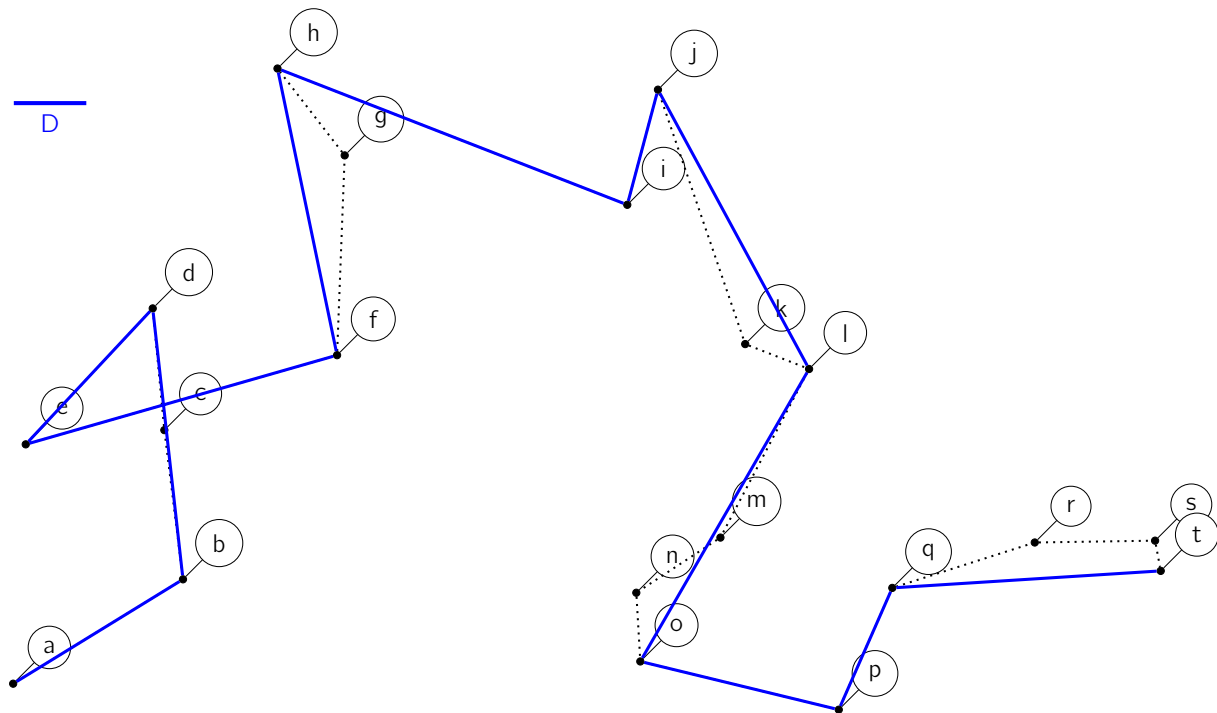


One can find all intermediate stack images below (read from left to right, then from top to bottom). The last stack image is the final one, and allows reading the entire path on the PD stack.

						n				j	
		e		i		i		i		i	
		a	o	a	o	a	o	e	n	e	
TP	PD	TP	PD	TP	PD	TP	PD	TP	PD	TP	PD
								e		e	
k						i		i		i	
j		j		j		j		j		j	
i		i	k	i	k	k		k		k	
e	n	e	n	e	n	e	n	n		b	n
a	o	a	o	a	o	a	o	a	o	a	o
TP	PD	TP	PD	TP	PD	TP	PD	TP	PD	TP	PD

			a
	b		b
	e		e
	i		i
	j		j
	k		k
	n		n
a	o		o
TP	PD	TP	PD

Solution 8.4.4.2-3: The final result looks like:



One can find all intermediate stack images below (read from left to right, then from top to bottom). The last stack image is the final one, and allows reading the entire path on the PD stack.

							q			
							p		p	
				o			o		o	
		h		h			h		h	
a	t	a	t	a	t	a	a	t	a	
TP	PD	TP	PD	TP	PD	TP	TP	PD	TP	PD
										j
										l
			o		o		l		o	o
o	p		p	j	p	j	j	p		p
h	q	h	q	h	q	h	h	q	h	q
a	t	a	t	a	t	a	a	t	a	t
TP	PD	TP	PD	TP	PD	TP	TP	PD	TP	PD

θ	ρ	$\rho_{rounded}$
$-\pi/2$	-1.0000	-1
$-3\pi/8$	-1.3066	-1
$-\pi/4$	-1.4142	-1
$-\pi/8$	-1.3066	-1
0	-1.0000	-1
$\pi/8$	-0.5412	-1
$\pi/4$	0.0000	0
$3\pi/8$	0.5412	1
$\pi/2$	1.0000	1

These $(\theta, \rho_{rounded})$ pairs can be used to increment the accumulator-cell matrix. This results in:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & \textcircled{0} & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Repeating the same procedure for p_2 results in:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & \textcircled{0} & 0 & 1 & 0 & 1 \\ 1 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Likewise for p_3 :

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 2 & 1 \\ 2 & 0 & 0 & 0 & \textcircled{0} & 0 & 1 & 0 & 2 \\ 1 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Continuing for point p_4 :

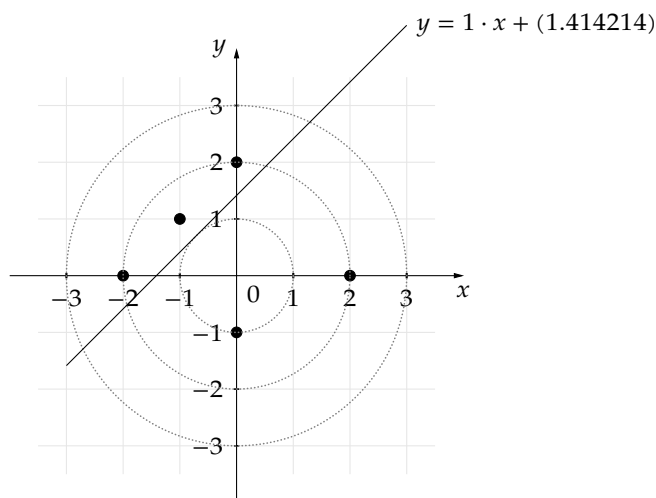
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 2 & 2 & 0 & 0 & 0 & 1 & 2 & 1 \\ 2 & 0 & 0 & 1 & \textcircled{1} & 1 & 1 & 0 & 2 \\ 1 & 2 & 2 & 1 & 1 & 1 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

And finally for point p_5 :

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 2 & 2 & 0 & 0 & 1 & 2 & 2 & 1 \\ 2 & 0 & 0 & 1 & \textcircled{2} & 1 & 1 & 0 & 2 \\ 1 & 2 & \boxed{3} & 2 & 1 & 1 & 2 & 2 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The maximum of 3 colinear points occurs for $(\theta, \rho) = (-\pi/4, -1)$. This corresponds to a line $y = ax + b$ (in Cartesian coordinates x and y with $a = 1$ and $b = \sqrt{2}$).

Drawing the given points and the approximate line we found, results in:



Note that the points don't coincide with the line found. This is due to the quantization of the θ and ρ values introduced by the accumulator-cell algorithm.

Solution 8.4.4.3-2: Let's start with an empty accumulator-cell matrix.

It counts 9 rows, one for every value of ρ , starting with $\rho = 3$ at the top and ending with $\rho = -3$ at the bottom.

It counts 11 columns, one for every value of θ , starting with $\theta = -\pi/2$ on the left and $\theta = \pi/2$

on the right. The center point (for $(\theta, \rho) = (0, 0)$) has been circled.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \textcircled{0} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Let's take the first point p_1 and apply the Hough transformation to it:

$$-1 \cdot \cos \theta - 1 \cdot \sin \theta = \rho$$

Let's fill out every value of θ in this equation, such that we obtain the corresponding values of ρ , that can be rounded to the given grid:

θ	ρ	$\rho_{rounded}$
$-\pi/2$	1.0000	0.75
$-2\pi/5$	0.6420	0.75
$-3\pi/10$	0.2212	0
$-\pi/5$	-0.2212	0
$-\pi/10$	-0.6420	-0.75
0	-1.0000	-0.75
$\pi/10$	-1.2601	-1.5
$\pi/5$	-1.3968	-1.5
$3\pi/10$	-1.3968	-1.5
$2\pi/5$	-1.2601	-1.5
$\pi/2$	-1.0000	-0.75

These $(\theta, \rho_{rounded})$ pairs can be used to increment the accumulator-cell matrix. This results in:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & \textcircled{0} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Repeating the same procedure for p_2 results in:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & \textcircled{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Likewise for p_3 :

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & \textcircled{1} & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 3 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Continuing for point p_4 :

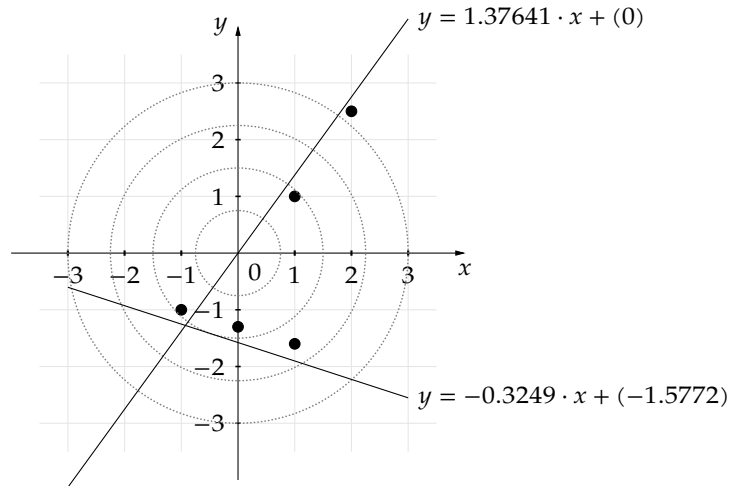
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 2 & 0 & \textcircled{1} & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 3 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

And finally for point p_5 :

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 & 2 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 3 & 0 & \textcircled{1} & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 3 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The maximum of 3 colinear points occurs for $(\theta, \rho) = (-\pi/5, -0)$ and $(\theta, \rho) = (-4\pi/10, -1.5)$. This corresponds to lines $y = ax + b$ (in Cartesian coordinates x and y with $(a, b) = (1.3764, 0)$ and $(a, b) = (-0.3249, -1.5772)$).

Drawing the given points and the approximate lines we found, results in:



Note that the points don't coincide with the line found. This is due to the quantization of the θ and ρ values introduced by the accumulator-cell algorithm.

Solution 8.4.4.3-3: The list of values for θ is straightforward. We can use 7 values and spread them over the range $[-\pi/2, \pi/2]$. This leads to $\theta = -\pi/2, -\pi/3, -\pi/6, 0, \pi/6, \pi/3, \pi/2$. The list of values for ρ is based on the rectangular bounding box, centered at the origin, containing all points. This the box $[-2, 2] \times [-2.5, 2.5]$. Half the diagonal of that rectangle equals 4.2426. It makes sense to pick a maximal ρ that is near that value. E.g., let's take 4. This allows for taking as values for $\rho = -4, -3, -2, -1, 0, 1, 2, 3, 4$. Of course, in real-world applications it does not make sense to use rounded values, but in these type of back-of-the-envelope exercises, it does.

Then, let's continue with the empty accumulator-cell matrix.

It counts 9 rows, one for every value of ρ , starting with $\rho = 4$ at the top and ending with $\rho = -4$ at the bottom.

It counts 7 columns, one for every value of θ , starting with $\theta = -\pi/2$ on the left and $\theta = \pi/2$ on the right. The center point (for $(\theta, \rho) = (0, 0)$) has been circled.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \textcircled{0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Let's take the first point p_1 and apply the Hough transformation to it, to update the accumulator matrix. This leads to:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \textcircled{0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Repeating the same procedure for p_2 results in:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & \textcircled{0} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Likewise for p_3 :

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & \textcircled{0} & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Continuing for point p_4 :

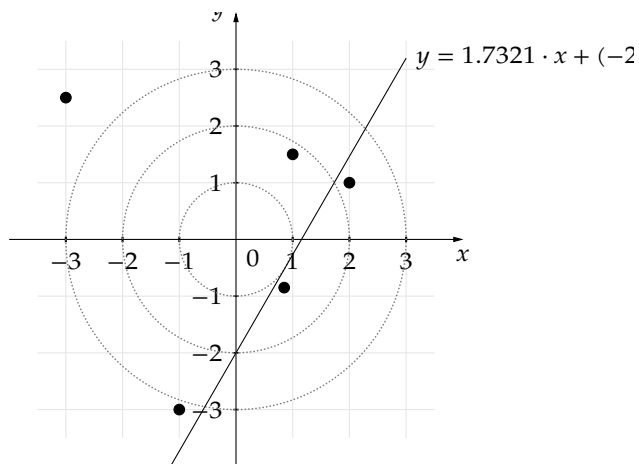
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 & 0 & 1 & 1 \\ 0 & 0 & 1 & \textcircled{0} & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

And finally for point p_5 :

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2 & 2 & 1 \\ 1 & 1 & 3 & 2 & 0 & 1 & 2 \\ 0 & 1 & 1 & \textcircled{0} & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The maximum of 3 colinear points occurs for $(\theta, \rho) = (-\pi/6, 1)$. This corresponds to line $y = ax + b$ (in Cartesian coordinates x and y with $(a, b) = (1.7321, -2)$).

Drawing the given points and the approximate line we found, results in:



Note that the obvious line that can be drawn almost interpolating the three top points is not found by the algorithm. This is due to the (to) coarse quantization of the values used for θ .