Problem 1

a) Use a skeletonization algorithm to shrink the object below. You should indicate the intermediate results.

(4p)

```
xxxxxxxxxx
xxxxx
xxxxxxxxxx
xxxxx
xxxxxxxxxx
```

1b) Determine a suitable binary operation, or sequence of operations (algorithm), for splitting up the two objects below (i.e. remove the pixel at the fifth line) without removing the top pixel. Describe the algorithm by words, name(s) of operation(s), and appearance(s) of structure elements.

(3p)

```
  1
1 1 1
1 1 1
1 1 1
  1
1 1 1
1 1 1
1 1 1
```

1c) Why does the skeletonization algorithm work better for long and thin objects?

(1p)

1d) Why may conventional shape factors based on measuring the object perimeter cause problems?

(1p)

1e) Determine a suitable hit-miss operator for finding corners like the underlined pixel below.

(1p)

```
111
111
111
```

Problem 2

a) The filters h_1 and h_2 are given below

h_1:

$1\ \underline{0}\ 1$

h_2:
1
$\underline{0}$
1

Convolve the two filters so as to create the filter h_3.
(2p)

2b) Compute the Fourier transform of h_3.
(2p)

2c) What type of filter is this (low-, band- or highpass)?
(1p)

2d) Why should the sum of the coefficients for a low-pass filter be one?
(1p)

2e) Why should the sum of the coefficients for a high-pass filter be zero?
(1p)

2f) What will happen to an image if it is subject to the filter h_4 specified below
(1p)

$0\ \underline{0}\ 1$

2g) Explain this phenomena by an analysis of the equivalent filtering in the Fourier domain. Hint: Compute the Fourier transform of h_4 and try to understand the relation of the filtering operation in the frequency and spatial domains.

Problem 3

Motion Correction

Two images are recorded of the same object (for example stem cells in a microscopy). Between the two sampling occasions the cells have moved only a little (time delay 1 min), and the images can be regarded as identical. However, due to vibrations the camera has moved relative to the object causing a horizontal shift. The algorithm for estimating this shift makes use of correlation. This is defined (for the one-dimensional case) as the convolution of

$$c_{f_1 f_2}(x) = f_1(x) * f_2(-x) = \int f_1(x-u) f_2(-u) du$$

(Notice the minus sign in $f_2(-x)$)

a) What is the Fourier transform, $C_{f_1 f_2}(\omega)$, of $c_{f_1 f_2}(x)$,?

(1p)

b) If the functions are identical except for a translation ($f_1(x) = f(x),\ f_2(x) = f(x - x_0)$), what is then $C_{f_1 f_2}(\omega)$?

(1p)

c) From the result in a2, where is the information about the translation: in $\left|C_{f_1 f_2}(\omega)\right|$, $\arg\left(C_{f_1 f_2}(\omega)\right)$, or in both?

(1p)

d) Write a formula for x_0 using the previous results and

$$x_0 = \frac{\sum x\delta(x - x_0)}{\sum \delta(x - x_0)} \quad \delta(x - x_0) = ...?$$

(2p)

Estimation of velocity field

Again, two cell images are recorded at different occasions but this time with very long time in-between (1 hour), During this time, cells have moved slightly but the camera has not moved. The so called Optical Flow algorithm makes use of two constraints, C_1 and C_2.

e) What are the names of these constraints, what are the mathematical definitions, and what is the meaning of each of them (describe by words)?

(3p)

f) The second constraint, C_2, makes use of a Lagrangian multiplier, λ. Explain the importance of λ?

(1p)

g) Which are the limitations of the Optical Flow algorithm? What may cause the algorithm to fail?

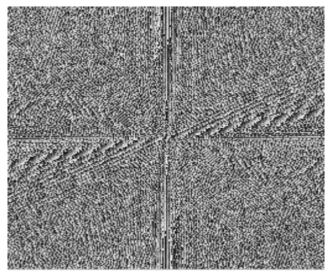
Problem 4

Below you find one input image and five output images. There is also a list of ten image analysis procedures. The task is to link each output image to a specific procedure (1-a, 2-b, etc.). Each correct link will give one point. For each link, an additional point will be given if you can clearly motivate your choice. Any incorrect motivation will lead to subtraction by one point. (10p)


Input image


1


2


3


4


5

a) Canny edge detection b) Sobel edge detection c) Histogram equalization

d) Ideal low-pass filtering e) Butterworth low-pass-filtering f) Fourier phase
g) Fourier magnitude h) Reconstruction using Fourier magnitude
i) Reconstruction using Fourier phase j) Ideal high-pass filtering

Problem 5

Apply Dynamic Programming for computing the optimal path from the left to the right border of the image, f, below. The brightness cost $C_1(p_i) = \max(f) - f(p_i)$ and the smoothness cost $C_2(p_{i-1}, p_i) = \Delta y$. The total cost $C(p_i) = w_1 * C_1(p_i) + w_2 * C_2(p_{i-1}, p_i)$, where the weights $w_1 = 2$ and $w_2 = 3$. The answer should include the cost accumulation matrix with back tracing pointers and the coordinates of the optimal path. (10p)

f(x, y):

5	1	1	4	4
2	3	4	2	3
3	2	1	3	6
2	2	3	3	4

Problem 6

a) The image shown in Fig. X contains two rectangular objects. Use the **watershed algorithm** to separate the objects from each other. Assume that the background is already segmented as one object.

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

Figure X.

5 p

b) Segment the image shown in Fig. Y using the **watershed algorithm**. The two seed pixels are underlined.

1	1	1	1	1	1	1	1
1	11	13	10	10	12	13	10
1	11	13	**9**	10	13	14	10
1	11	13	12	12	13	15	12
1	11	13	14	14	15	15	11
1	12	12	12	12	13	3	11
0	4	3	3	3	2	2	2
1	2	2	2	2	2	2	2

Figure Y.

2 p

c) Segment the image shown in Fig. Y using **region growing**. The two seed pixels are underlined. Use $\Delta=3$. *2p*

d) Comment on the result of b) and c). *1p*

Solution 1

a)
0000xxxxxx
0000x
00xxxxxxxx
000x0
000xxxxxxx

b)
Apply the hit-miss operation using the 3 x 3 structure element below. Remove pixels at a perfect fit.

111
010
111

c) Because it is sensitive to shape noise

d) Because of the fractal effect

e)
0 0 0
1 1 0
1 1 0

Solution 2

a)
1 0 1
0 0 0
1 0 1

b) 2cos(u+v) + 2cos(u-v)

c) Low-pass

d) To preserve the mean intensity of the image

e) To prevent the filter from giving a non-zero response in a homogenous region

f) It will be shifted one step to the left

g) The Fourier transform of the filter is an exponential. To multiplicate in the Fourier domain with an exponential means shifting in the image domain.

Solution 3

a) $C_{f_1 f_2}(\omega) = F[f_1] \cdot \overline{F[f_2]}$

b) $C_{f_1 f_2}(\omega) = F[f_1] \cdot \overline{F[f_2]} = F[f] \cdot \overline{F[f]} \cdot e^{i2\pi x_0} = \left|F[f]\right|^2 e^{i2\pi x_0}$

c) $\ln \arg\left(C_{f_1 f_2}(\omega)\right)$

d) $x_0 = \frac{\sum x\delta(x-x_0)}{\sum \delta(x-x_0)} \quad \delta(x-x_0) = F^{-1}\left[e^{i2\pi x_0}\right] = \left[\frac{F[f_1]\cdot\overline{F[f_2]}}{\left|F[f_1]\cdot\overline{F[f_2]}\right|}\right]$

e)

Brightness (or spatio-temporal) constraint:

$$\frac{\partial I}{\partial x}u(x_1)+\frac{\partial I}{\partial y}v(y_1)+\frac{\partial I}{\partial t}=0$$

This constraint means that the brightness of a particular point in a pattern is constant

Velocity smoothness constraint

$$\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2+\left(\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2=0$$

This constraint means that neighboring points in a pattern have similar velocities

f) A larger value of λ results in a more smooth velocity field.

g) If every point of the brightness pattern move independently, there is little hope of recovering the velocities.

Solution 4

1 – d (Ideal low-pass filtering).
The image is smoothed but the ringing effect typical for ideal lp-filtering is clearly demonstrated.

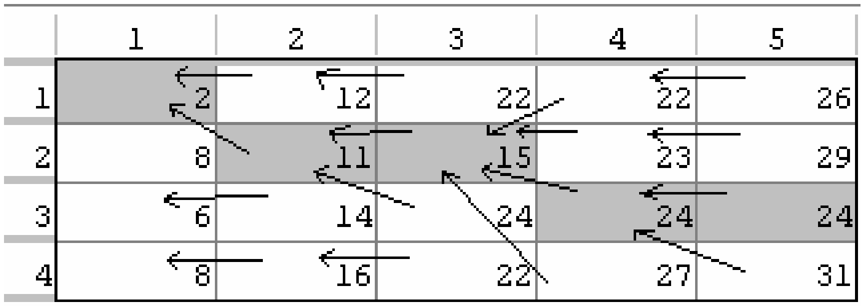
2 – c (Histogram equalization)
The image contrast is improved, especially for the darker parts.

3 – f (Fourier phase)
There are no other plausible alternatives.

4 - i (Reconstruction using Fourier phase)
The image shows the position of image intensity shifts but has not got the characteristics of neither Canny nor Sobel filtering.

5 – b (Sobel edge detection)
The image presents thick and signed edges in both directions. The image has clearly been subject to an additive offset so pixels with negative sign becomes zero or higher. Hence, homogenous regions with zero-response become gray because of this offset.

Solution 5



Solution 6:

a) First use any of the distance transforms:

Euclidean:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	1	2	2	2	2	1	0	0	0	0	0	0	0	0
0	1	2	3	3	2	1	0	0	0	0	0	0	0	0
0	1	2	3	3	2	1	0	0	0	0	0	0	0	0
0	1	2	3	3	2	1	0	0	0	0	0	0	0	0
0	1	2	2	2	2	1	0	0	0	0	0	0	0	0
0	1	1	1	1	$\sqrt{2}$	$\sqrt{2}$	1	1	1	1	1	1	0	0
0	0	0	0	0	1	2	2	2	2	2	2	1	0	0
0	0	0	0	0	1	2	3	3	3	3	2	1	0	0
0	0	0	0	0	1	2	2	2	2	2	2	1	0	0
0	0	0	0	0	1	1	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	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

Cityblock:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	1	2	2	2	2	1	0	0	0	0	0	0	0	0
0	1	2	3	3	2	1	0	0	0	0	0	0	0	0
0	1	2	3	3	2	1	0	0	0	0	0	0	0	0
0	1	2	3	3	2	1	0	0	0	0	0	0	0	0
0	1	2	2	2	2	1	0	0	0	0	0	0	0	0
0	1	1	1	1	2	2	1	1	1	1	1	1	0	0
0	0	0	0	0	1	2	2	2	2	2	2	1	0	0
0	0	0	0	0	1	2	3	3	3	3	2	1	0	0
0	0	0	0	0	1	2	2	2	2	2	2	1	0	0
0	0	0	0	0	1	1	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	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

Chessboard:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	1	2	2	2	2	1	0	0	0	0	0	0	0	0
0	1	2	3	3	2	1	0	0	0	0	0	0	0	0
0	1	2	3	3	2	1	0	0	0	0	0	0	0	0
0	1	2	3	3	2	1	0	0	0	0	0	0	0	0
0	1	2	2	2	2	1	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	0	0	1	2	2	2	2	2	2	1	0	0
0	0	0	0	0	1	2	3	3	3	3	2	1	0	0
0	0	0	0	0	1	2	2	2	2	2	2	1	0	0
0	0	0	0	0	1	1	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	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

Either find local max and let intensity decrease, or use 1/(dist. im.). Background already segmented.

Image after watershed transform:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	2	2	2	2	0	1	1	1	1	1	1	1	1
1	0	2	2	2	2	0	1	1	1	1	1	1	1	1
1	0	2	2	2	2	0	1	1	1	1	1	1	1	1
1	0	2	2	2	2	0	1	1	1	1	1	1	1	1
1	0	2	2	2	2	0	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
1	1	1	1	1	0	3	3	3	3	3	3	0	1	1
1	1	1	1	1	0	3	3	3	3	3	3	0	1	1
1	1	1	1	1	0	3	3	3	3	3	3	0	1	1
1	1	1	1	1	0	0	0	0	0	0	0	0	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

1=background, 0=watershed lines, 2=object 1, 3=object 2

b)

1	1	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	1	1	1
1	1	0	2	2	2	0	1	1	1
1	1	0	2	2	2	0	1	1	1
1	1	0	0	0	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

1=background (or object 1), 2= object (or object 2), 0=watershed line c)

1	1	1	1	1	1	1	1
1	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2
1	2	2	2	2	2	2	2
1	2	2	2	2	2	1	2
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

1= background (or object 1), 2= object (object 2)

mean of regions for each iteration:

it:	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
b	0	1	1.8	1.8	1.9	1.8	1.8	1.8	1.8	1.7	1.7	1.7
o	9	10.5	11.8	12	12.2	12.1	12.1	12.1	12.1	12.1	12.1	12.1

d) Watershed transform segments based on where the “water level” gets first, not based on intensity on neighboring pixels as region growing. Therefore object area becomes larger when doing region growing since the region with high values is larger than found at watershed segmentation.