Lab 3:
Histogram and gray scale transformation. Operations on binary images. Distance maps.

Maria Magnusson, 2018, 2019, 2021, 2023
Computer Vision Laboratory, Department of Electrical Engineering,
Linköping University, Sweden
Based on an older lab developed at the Department of Electrical Engineering.

1 Introduction

Read all instructions before the lab-exercise. Exercises marked with a pointing hand are supposed to be solved as preparation before the lab-exercise. Suggested answers for all questions, as well as a multiple choise table is in the end of this lab booklet. In case of any wrong answer, the problem should be discussed with the teacher.

A computer symbol indicates that a MATLAB -script has to be written. Save all your MATLAB -scripts and demonstrate them to the teacher.

Extra

An exercise marked with 'Extra' can be performed in terms of interest.

Start by extracting the zip-archive containing the files `baboon.tif`, `circles.tif`, `distObject.tif`, `labyrinth1.tif`, `labyrinth2.tif`, `blod256.mat`, `clic.mat`, `nuf0a.mat`, `nuf4b.mat`, `nuf5.mat`, `track10.m`, `trackDist.m` into a suitable folder.

2 Histogram and gray scale transformation

From a histogram, we get information about the gray scale distribution in the image. Compute the histogram of the light image **clic** by creating a file `ShowHistograms.m` with the subsequent contents and execute it.

```
binvect = [0:1:255];
load clic.mat
histo = hist(clic(:), binvect);
figure(1)
colormap(gray(256))
subplot(2,1,1), imagesc(clic, [0 255]);
axis image; title('light image'); colorbar
subplot(2,1,2), plot(binvect, histo, '.-b');
axis tight; title('histogram')
```

The vector `binvect` gives the central positions for the "containers" in the histogram and `clic(:)` flattens the image into a long vector.

QUESTION 1: How are the lighting conditions reflected in the histogram for the light image **clic**?

Relatively high contrast, many light pixels, spike at 231 (background color)

Extend the MATLAB script `ShowHistograms.m` so that it shows the dark image **blod256** and its histogram in `figure(2)`.

QUESTION 2: How are the lighting conditions reflected in the histogram for the dark image **blod256**?

Very dark lighting conditions, not the best contrast

Extend the MATLAB script `ShowHistograms.m` so that it shows the image **baboon** and its histogram in `figure(3)`. Baboon is loaded with:
`Im = double(imread('baboon.tif'));`.

QUESTION 3: How is the relatively low contrast in the image **baboon** reflected in the histogram?

The histogram is squeezed into a small range

Extend the MATLAB script `ShowHistograms.m` with a gray scale transformation of the image **baboon** so that the contrast increases:
`ImT = A * Im - B;`
Let `Im` $= 50 \Rightarrow$ `ImT` $= 0$ and `Im` $= 200 \Rightarrow$ `ImT` $= 255$.

QUESTION 4: Which values do you give A and B?

A = 1.7 B = 85

QUESTION 5: Look at the histogram of `ImT`, the transformed image. Note that the new histogram is more spread out than the previous histogram of `Im`. Why is there a high peak for pixel value 0 in the histogram?

All negative values goes to zero

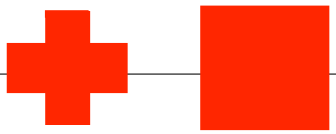
3 Binary operators

3.1 Non-connectivity preserving expansion and contraction

QUESTION 6: The operations dilation and erosion are the basic morphological operations for binary images. Describe the principles of dilation and erosion in terms of what happens with the neighbors!

Dilation: value 0, neighbour 1 => value 1, Erosion: value 1, neighbour 0 => value 0

EXERCISE I: Sketch the structuring elements for $d^{(4)}$, $d^{(8)}$ and octagonal metric!



Create a file `DilatePoints.m` with the subsequent contents and execute it.

```
Im = zeros(64,64);
Im(20,20) = 1; Im(50,30) = 1; Im(25,50) = 1;
SE4 = [0 1 0;
       1 1 1;
       0 1 0];
Im_d4 = imdilate(Im, SE4);
figure(1)
colormap(gray(256))
subplot(2,3,1), imagesc(Im);
axis image; title('Three points');
subplot(2,3,2), imagesc(Im_d4);
axis image; title('1 iter d4');
```

The original image with three points (top-left in figure 1) is dilated by $d^{(4)}$ (top-middle).

DEMO A: Extend the file `DilatePoints.m` with:
one dilation of $d^{(8)}$ (top-right),
10 iterations of $d^{(4)}$ (bottom-left),
10 iterations of $d^{(8)}$ (bottom-middle),
Finally, to the bottom-right, show 5 iterations of $d^{(oct)}$, which is equal to 5 iterations of $d^{(4)}$ and 5 iterations of $d^{(8)}$.
Tip: You will need to write a for-loop.

QUESTION 7: Which structuring element allows for the most uniform dilation?

octagon?

Create a MATLAB script `nuf4bHistogram.m`, where you load the image **nuf4b** and compute its histogram.

QUESTION 8: How do you use the histogram to find out an appropriate threshold for the image? Which threshold `T` is appropriate for **nuf4b**?

aroundd 133

DEMO B: Threshold nuf4b by using: `nuf4bT = (nuf4b<T);`, where `T` is your chosen threshold. Then all pixels that fullfil (nuf4b<T) are set to 1. The others are set to 0. Note that the object is white and contains ones after thresholding and the background is black and contains zeros after thresholding.

QUESTION 9: How can we fill in cracks and holes in the object? How can we remove spurs and unwanted branches along the outline of the object? The thickness of the object should not change.

dilate and erode (close)

DEMO C: The thresholded image that you get from **nuf4b** has already a smooth outline, right? However, the thresholded images that you will get from **nuf5** and **nuf0a** have some problems with the outline. Make experiments to find out a combination of dilations (**imdilate**) and erosions (**imerode**) in $d^{(4)}$ and $d^{(8)}$ metrics to improve the outline of these images.

• The problem with **nuf5** is a crack between the "roof" and the "body".
• The problem with **nuf0a** is an unwanted spur and a small hole. The text "NOL" is not a problem.
• The thickness of the object should not change.

QUESTION 10: Which threshold `T` is appropriate for **nuf5**?

179

QUESTION 11: Which threshold `T` is appropriate for **nuf0a**?

138

3.2 Connectivity preserving shrinking

Unlimited use of the previously described operations for erosion may divide objects into parts. Also, small objects risk to completely disappear. This can be prevented by connectivity preserving shrinking.

QUESTION 12: Give a reference to the two lecture slides that show the structuring elements for connectivity preserving shrinking for $d^{(4)}$ and $d^{(8)}$ connectivity.

MATLAB has a slightly different variant in `bwmorph` called with the `'shrink'` parameter. It seems to give a slightly jagged contour of the objects.

Create a file `ShrinkObjects.m` with the subsequent contents and execute it.

```
ImC = imread('circles.tif');
SE4 = [0 1 0;
       1 1 1;
       0 1 0];
SE8 = [1 1 1;
       1 1 1;
       1 1 1];
ImE = ImC;
for k=1:4
  ImE = imerode(ImE, SE4);
  ImE = imerode(ImE, SE8);
```

```
end
ImS = bwmorph(ImC,'shrink',4);
figure(1)
colormap(gray(256))
subplot(2,3,1), imagesc(ImC);
axis image; title('Circles');
subplot(2,3,2), imagesc(ImE);
axis image; title('ImE');
subplot(2,3,3), imagesc(ImS);
axis image; title('ImS');
```

QUESTION 13: In the top-middle, an erosion of four iterations with $d^{(oct)}$ is shown and in the top-right, four iterations of connectivity preserving shrinking is shown. Which operation may divide one object into two?

erosion

Consult the documentation of `bwmorph` and change the third parameter so that the operation is repeated until the image no longer changes. Then all objects are supposed to shrink to a point.

QUESTION 14: However, there exist some objects that will not be shrunk to a point, but rather to connected curves. What is the special property of such an object?

objects with holes

DEMO D: Extend the file `ShrinkObjects.m` so that it shrinks all objects to a point and then calculates the number of objects in the original image. You will need the commands `sum` and `imfill` with parameter `'holes'`. The command `imfill` fills holes in a binary image. One way to do this is explained in a lecture.

3.3 Thinning to a skeleton

If we want to shrink the object, but still keep the overall shape of it, we can perform thinning and end up with a skeleton.

QUESTION 15: Give a reference to the two lecture slides that show the structuring elements for thinning to skeleton for $d^{(4)}$ and $d^{(8)}$ connectivity.

36 38

QUESTION 16: What is the difference between a 4- and an 8-connective skeleton?

8 connective have diagonal but 4-connective does not

QUESTION 17: Do you think that it is possible to reconstruct the object from its skeleton?

not generally

DEMO E: Previously we loaded and thresholded the image **nuf4b**. Do that again in a script called `nuf4bskeleton.m`. Then compute the skeleton by using `bwmorph` called with the `'skel'` and `inf` parameters.

QUESTION 18: Did you recieve a 4- or an 8-connective skeleton?

8con

Note that skeleton have distracting spurs. Delete the spurs with a couple of iterations of connectivity preserving shrinking. Use `bwmorph` with `'shrink'`. This operation is called "pruning".

QUESTION 19: How many iterations did you need to prune the skeleton?

3

---

4 Shape matching

Connectivity preserving shrinking actually uses shape matching. Nevertheless, we will now study shape matching with the Hit-or-Miss transformation in detail.

QUESTION 20: Give a reference to the lecture slide that shows the structuring elements for $d^{(4)}$ endpoints.

EXERCISE II: Try to figure out the structuring elements for $d^{(8)}$ endpoints. There are eigth different!

Note that one of the three structuring element below should match the upper-left stroke of your pruned skeleton.

-	0	0
1	1	0
-	0	0

,

0	0	0
0	1	0
1	0	0

or

0	0	0
0	1	0
-	1	-

.

The equation for the Hit-or-Miss transformation is given in the lecture slides. Consequently, it can be implemented using the MATLAB commands `imerode`, and `.*`.

DEMO F: Extend `nuf4bskeleton.m` so that the endpoint of the upper-left stroke is detected. (Depending of the detailed shape of your skeleton, you may also detect another endpoint.) Show an image of the endpoint(s), possibly overlaid on the pruned skeleton of **nuf4b**.

Then call the function `track10.m`, which tracks 10 pixels along the skeleton, starting from the given endpoint `[y,x]`. (`track10.m` is listed in the appendix.) It outputs the reached coordinates `[yout,xout]`.
From `[yout,xout]` and `[y,x]` and using the function `atan2`, the angle of the stroke can be calculated.

Endpoint cord: 57,23

QUESTION 21: What is the angle of the upper-left stroke?

angle is 2.4150 radians

It is also possible to recieve a skeleton with a different algorithm called the medial axis transform (MAT), which is used by the `bwskel` function in MATLAB .

Extra **QUESTION 22**: If you run a MATLAB version of 2018 or later, you can try `bwskel` on the thresholded image **nuf4b**. Compare this skeleton with the skeleton obtained by using `bwmorph` with the `'skel'` and `inf` parameters. What is the main difference between the two skeletons?

---

5 Segmentation and labeling

Thresholding segments the image into objects and background. The process to identify individual objects and to give them names is called labeling. When the labeling algorithm is finished, all objects are marked with unique labels, and we will be able to see how many objects there are in the image.

We will now label the **circles** image. Create a file `LabelObjects.m` with the subsequent contents and execute it.

```
ImC = imread('circles.tif');
CC = bwconncomp(ImC);
ImLabel = labelmatrix(CC);
figure(1)
colormap(gray(256))
subplot(2,2,1), imagesc(ImC);
axis image; title('Circles'); colorbar
subplot(2,2,2), imagesc(ImLabel);
axis image; title('Label image'); colorbar
```

Apparently, `bwconncomp` in combination with `labelmatrix` give a labelled image. According to the MATLAB documentation, the basic steps in finding the connected components and label them are:

1) Search for the next unlabeled pixel, p.

2) Use a flood-fill algorithm to label all the pixels in the connected component containing p.

3) Repeat steps 1 and 2 until all the pixels are labelled.

Change the colortable to `colormap(jet(256))` for a nicer look!

QUESTION 23: Which label values do the objects in your image receive?

1 to 10

QUESTION 24: Which label value is given to the background?

0

QUESTION 25: In which order brings the algorithm out the labels to the various objects?

top bot left right

QUESTION 26: It is easy to get a binary image of only one object. How do you get a binary image with only the object labelled 5? Give a MATLAB command.

Im5 = (ImLabel==5)

According to the MATLAB documentation, it is easy to get a binary image of the largest object. Extend the MATLAB script `LabelObjects.m` with the following commands:

```
ImBig = 0*ImLabel;
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
ImBig(CC.PixelIdxList{idx}) = 1;
subplot(2,2,4), imagesc(ImBig);
axis image; title('ImBig'); colorbar
```

QUESTION 27: Which was the largest object, i.e. which label did it have?

6

---

6 Distance measures in binary images

A distance map in the backgrond shows the shortest distance from each pixel in the background to the outline of the object. Similarly, a distance map in the object shows the shortest distance from each pixel in the object to the outline of the object.

6.1 Distance maps in the background

Create a file `DistanceMaps.m` with the subsequent contents and execute it.

```
Im = zeros(64,64);
Im(20,20) = 1;
Im(50,30) = 1;
Im(25,50) = 1;
Im_eq = bwdist(Im);
figure(1)
colormap(colorcube(51))
subplot(2,2,1), imagesc(Im);
axis image; title('Three points'); colorbar
Im_eq = bwdist(Im);
subplot(2,2,2), imagesc(Im_eq, [0 50]);
axis image; title('Euclidian'); colorbar
```

The colormap **colorcube** is used for better visualization.
The default and shown distance map is the 'Euclidian'. Extend the MATLAB script `DistanceMaps.m` so that it also shows the `'cityblock'` and `'chessboard'` distance maps.

QUESTION 28: Click in the images and check which distance maps contains real values and which contains integers.

euclidian is only one with real values

QUESTION 29: Which distance map can be produced by using a $d^{(4)}$ structuring element?

city

6.2 Distance map in the object

Extend the file `DistanceMaps.m` with the subsequent contents and execute it.

```
ImD = imread('distObject.tif');
figure(2)
colormap(colorcube(51))
subplot(2,2,1), imagesc(ImD);
axis image; title('distObject'); colorbar
```

QUESTION 30: How can you get a distance map inside the object? Give a MATLAB command.

imd_eq = bwdist(~ImD)

QUESTION 31: Show the distance map, click in the image and search for maximum values. How is the maximum value of the distance map **maxDist** related to the maximum thickness **maxThick** of the object? Give an equation.

maxThich = 2 maxDist

6.3 Distance map with obstacles. Application: Labyrinth

Here we will apply the distance map to a binary image with the object pixels regarded as obstacles. A labyrinth will be solved by finding the "shortest path", which is obtained with a distance map followed by a tracking algorithm.

If you like, you can try to solve one of the labyrinths in the appendix at home.

Create a file `Labyrinth.m` with the subsequent contents and execute it.

```
Im = imread('labyrinth1.tif');
maxiter = 500; % Number of iterations
maxval = 5000;
DistIm = maxval .* Im; % Set the obstacles to a high value
temp1  = 0 .* Im;
temp1(20,160) = 1; % Starting point
SE8 = [1 1 1;
       1 1 1;
       1 1 1];
for k=1:maxiter
  temp2 = imdilate(temp1, SE8);
  temp2 = temp2 .* ~Im;
  DistIm = DistIm + k*(temp2-temp1);
  temp1 = temp2;
end
figure(1)
colormap(gray(256))
imagesc(Im);
axis image; title('labyrinth'); colorbar
```

```
figure(2)
colormap(colorcube(5000+1))
imagesc(DistIm, [0 maxval]);
axis image; title('DistIm'); colorbar
```

QUESTION 32: Notice that the distance map starts from one point and spreads into the background and in the labyrinth. The distance map is not allowed to proceed into the obstacles. Which position has the starting point?

20,160

DEMO G: Change the number of iterations so that the distance map spreads into the whole labyrinth. Do not take too many iterations, because it is a slow implementation. Then call the function `trackDist.m` with `Im`, `distIm` and a suitable starting point. The function performs tracking in the distance map towards lower values and returns an image with the tracking result. (`trackDist.m` is given and also listed in the appendix.)

QUESTION 33: Compare the labyrinths in the appendix pages with the labyrinths on the computer. There is a slight difference in the form of two white spots. Why was it necessary to add these white spots?

shortcut otherwise

---

7 Appendix

7.1 track10.m

```
function [yout,xout] = track10(Im, y, x)
% This function tracks 10 pixels along a skeleton in image Im.
% It starts from the given end-point [y,x].
% It outputs the reached coordinates [yout,xout].

for k=1:10
  text = sprintf('y = %d, x = %d.', y, x); disp(text);
  ROI = Im(y-1:y+1,x-1:x+1); % grab ROI
  ROI(2,2) = 0;                % set current pos to 0
  if k>1
    ROI(4-ypos,4-xpos) = 0;    % set old pos to 0
  end
  [ypos,xpos] = find(ROI==1);  % look for the one
  y = y+ypos-2;                % update y-position
  x = x+xpos-2;                % update x-position
end
```

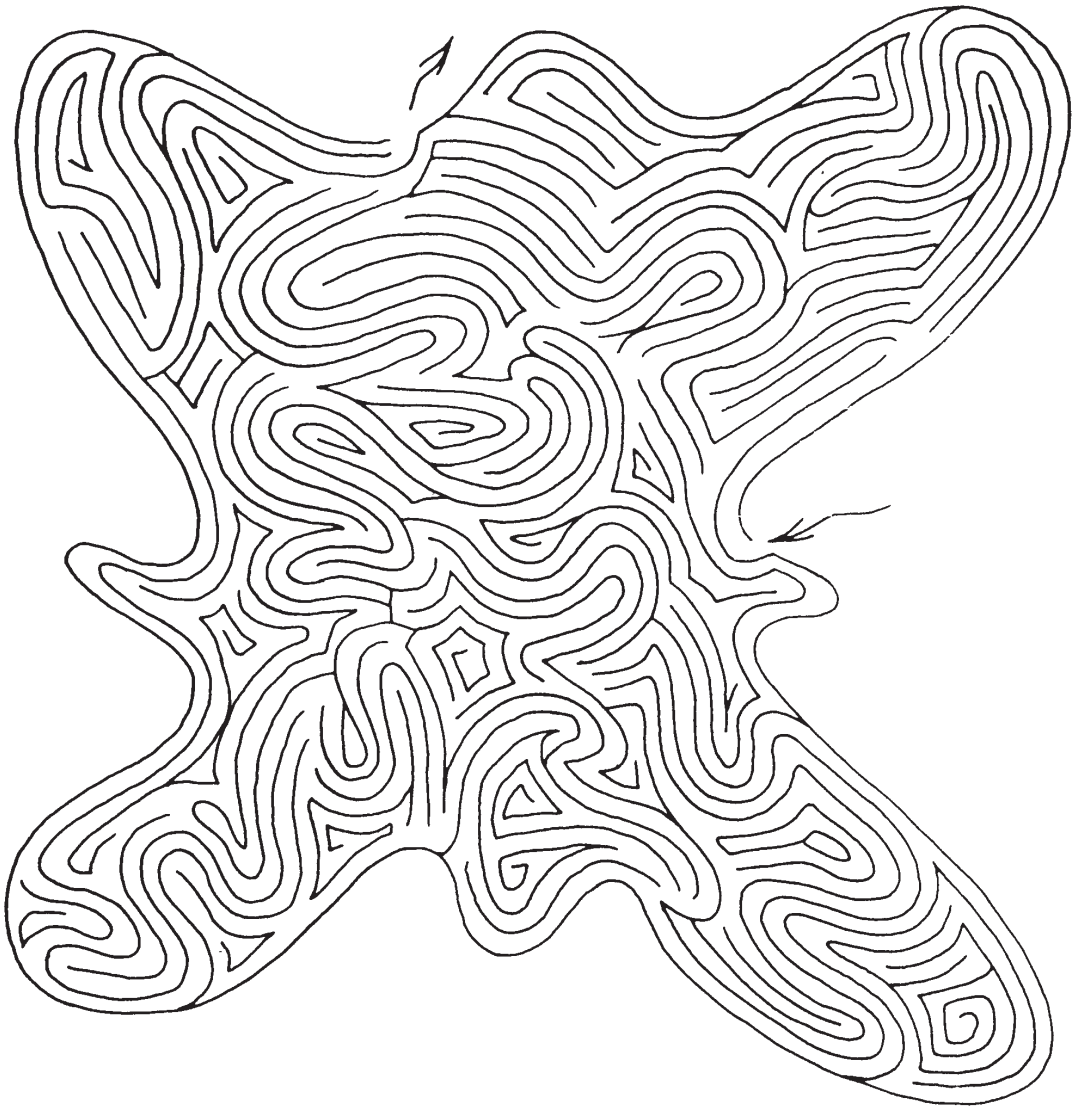
```
yout = y;
xout = x;
```

7.2 trackDist.m

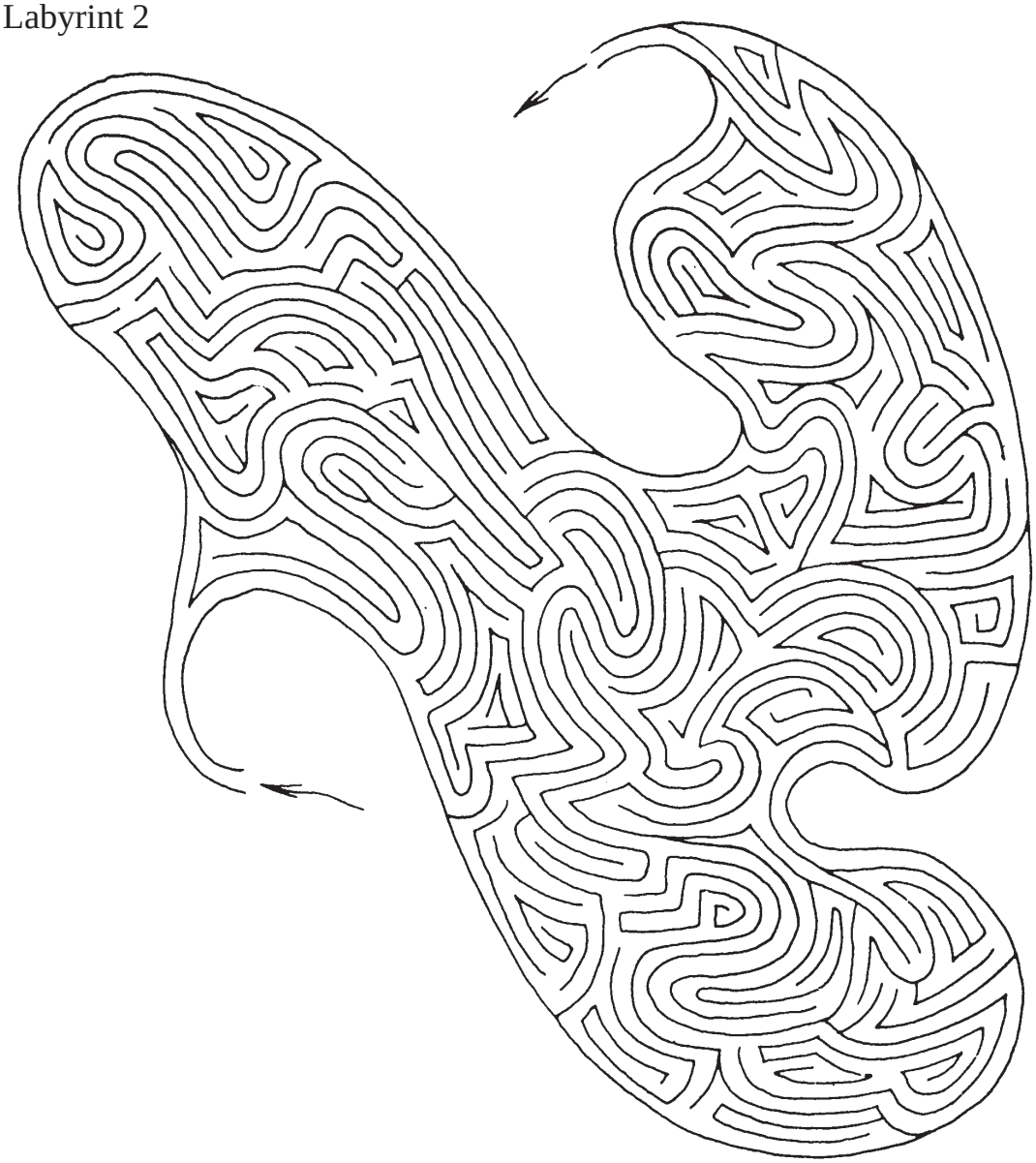
```
function TrackIm = trackDist(Im, DistIm, starty, startx);
% This function performs tracking in the distance map DistIm
% towards lower values.
% Im is the original image with obstacles.
% starty is the starting position in y (row).
% startx is the starting position in x (column).
% The function returns an image with the tracking result.

y = starty;
x = startx;
val = DistIm(y,x);
TrackIm = double(Im);
iter = max(DistIm(:))+1;
k=0;
while val>0 & iter>0
  k=k+1;
  if k<=10
    text = sprintf('iter = %d: y = %d, x = %d, val = %d.', k, y, x, val);
    disp(text);
  end
  if k==10
    disp('I will only display the result of 10 iterations.');
  end
  val = DistIm(y,x);
  if (DistIm(y-1,x-1)<val) ypos=y-1; xpos=x-1; val=DistIm(y-1,x-1); end
  if (DistIm(y+1,x-1)<val) ypos=y+1; xpos=x-1; val=DistIm(y+1,x-1); end
  if (DistIm(y-1,x+1)<val) ypos=y-1; xpos=x+1; val=DistIm(y-1,x+1); end
  if (DistIm(y+1,x+1)<val) ypos=y+1; xpos=x+1; val=DistIm(y+1,x+1); end
  if (DistIm(y-1,x)<=val)  ypos=y-1; xpos=x;   val=DistIm(y-1,x); end
  if (DistIm(y+1,x)<=val)  ypos=y+1; xpos=x;   val=DistIm(y+1,x); end
  if (DistIm(y,x-1)<=val)  ypos=y;   xpos=x-1; val=DistIm(y,x-1); end
  if (DistIm(y,x+1)<=val)  ypos=y;   xpos=x+1; val=DistIm(y,x+1); end
  y = ypos; % update y-position
  x = xpos; % update x-position
  TrackIm(y,x) = 2;
end

if k == 1 disp('Failed to track, only one iteration'); end
if iter == 0 disp('Failed to track, too many iterations'); end
```

7.3 Labyrinth images

Labyrint 1



Labyrint 2



8 Suggested answers

ANSWER 1:
a) The histogram is displaced to the right.
b) The histogram is displaced to the left.
c) The histogram is squeezed into a smaller range.

ANSWER 2:
a) The histogram is displaced to the right.
b) The histogram is displaced to the left.
c) The histogram is squeezed into a smaller range.

ANSWER 3:
a) The histogram is displaced to the right.
b) The histogram is displaced to the left.
c) The histogram is squeezed into a smaller range.

ANSWER 4:
a) $A = 1.8$, $B = 90$
b) $A = 1.9$, $B = 95$
c) $A = 1.7$, $B = 85$

ANSWER 5:
a) There is an error in the MATLAB plot function.
b) Since `binvect` starts at 0, all values below 0 also go into the 0 "container".
c) The image is much darker after the transformation than before.

ANSWER 6:
a) Dilation: If the value is 1 and one neighbour is 1, set the new value to 1.
Erosion: If the value is 0 and one neighbour is 0, set the new value to 0.
b) Dilation: If the value is 0 and one neighbour is 1, set the new value to 1.
Erosion: If the value is 1 and one neighbour is 0, set the new value to 0.
c) Dilation: If the value is 0 and one neighbour is 1, set the new value to 0.
Erosion: If the value is 1 and one neighbour is 0, set the new value to 1.

ANSWER 7:
a) d^4 gives the most uniform result.
b) $d^{(oct)}$ gives the most uniform result.
c) d^8 gives the most uniform result.

ANSWER 8:
a) Set the threshold between the two humps in the histogram, `T`$\approx$`155`.
b) Set the threshold between the two humps in the histogram, `T`$\approx$`145`.
c) Set the threshold between the two humps in the histogram, `T`$\approx$`135`.

ANSWER 9:
a) Cracks and holes: Dilate - Erode. Unwanted branches: Erode - Dilate.
b) Cracks and holes: Dilate - Dilate. Unwanted branches: Erode - Erode.
c) Cracks and holes: Erode - Dilate. Unwanted branches: Dilate - Erode.

ANSWER 10:
a) $\texttt{T}\approx\texttt{202}$.
b) $\texttt{T}\approx\texttt{192}$.
c) $\texttt{T}\approx\texttt{182}$.

ANSWER 11:
a) $\texttt{T}\approx\texttt{140}$.
b) $\texttt{T}\approx\texttt{130}$.
c) $\texttt{T}\approx\texttt{120}$.

ANSWER 12:
a) $d^{(4)}$: p. 33 and $d^{(8)}$: p. 38.
b) $d^{(4)}$: p. 29 and $d^{(8)}$: p. 38.
c) $d^{(4)}$: p. 29 and $d^{(8)}$: p. 31.

ANSWER 13:
a) Shrink.
b) Erosion.
c) Both erosion and shrink.

ANSWER 14:
a) Non-convex objects.
b) Large objects.
c) Objects with holes.

ANSWER 15:
a) $d^{(4)}$: p. 36 and $d^{(8)}$: p. 38.
b) $d^{(4)}$: p. 31 and $d^{(8)}$: p. 38.
c) $d^{(4)}$: p. 29 and $d^{(8)}$: p. 38.

ANSWER 16:
a) 4-connective skeletons has horizontal, vertical and diagonal connections.
8-connective skeletons has horizontal and vertical connections.
b) 4-connective skeletons has horizontal and vertical connections.
8-connective skeletons has horizontal, vertical and diagonal connections.
c) 4-connective skeletons has horizontal connections.
8-connective skeletons has horizontal and vertical connections.

ANSWER 17:
a) Yes, exactly. It is easy to perform the inverse operation.
b) No, not at all. The skeleton have nothing in common with the original object.
c) No, not exactly, thick as well as thin lines are shrinked to the width of 1 pixel.

ANSWER 18:
a) An 8-connective skeleton.
b) A 4-connective skeleton.
c) I did not get a skeleton.

ANSWER 19:
a) 7 iterations using `'shrink'`.
b) 3 iterations using `'shrink'`.
c) 1 iteration using `'shrink'`.

ANSWER 20:
a) p. 18
b) p. 28
c) p. 29

ANSWER 21:
a) Around 60 deg.
b) Around 50 deg.
c) Around 40 deg.

ANSWER 22: **(EXTRA)**
a) The MAT skeleton contains more spurs.
b) The MAT skeleton contains less spurs.
c) The MAT skeleton contains no spurs.

ANSWER 23:
a) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
b) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
c) 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

ANSWER 24:
a) 0
b) 1
c) 2

ANSWER 25:
a) From top to bottom and from left to right.
b) From bottom to top and from right to left.
c) From top to bottom and from right to left.

ANSWER 26:
a) `Im5 = (ImLabel==5);`
b) `Im5 = (ImLabel=5);`
c) `Im5 = ImLabel(5);`

ANSWER 27:
a) 7
b) 6
c) 10

ANSWER 28:
a) The `'cityblock'` distance map contains real values.
The others contain integers.
b) The `'Euclidian'` distance map contains real values.
The others contain integers.
c) The `'chessboard'` distance map contains real values.
The others contain integers.

ANSWER 29:
a) `'chessboard'` can be produced by dilations of the $d^{(4)}$
structuring element.
b) `'cityblock'` can be produced by dilations of the $d^{(4)}$
structuring element.
c) `'Euclidian'` can be produced by dilations of the $d^{(4)}$
structuring element.

ANSWER 30:

```
a) ImD_eq = bwdist(ImD');
b) ImD_eq = bwdist(~ImD);
c) ImD_eq = 1-bwdist(ImD);
```

ANSWER 31:
a) $\mathbf{maxThick} \approx 3 \cdot \mathbf{maxDist}$
b) $\mathbf{maxThick} \approx \mathbf{maxDist}$
c) $\mathbf{maxThick} \approx 2 \cdot \mathbf{maxDist}$

ANSWER 32:
a) The distance map starts from (40,360).
b) The distance map starts from (30,260).
c) The distance map starts from (20,160).

ANSWER 33:
a) Otherwise the labyrinth solver gets stuck in the starting point.
b) Otherwise the labyrinth solver stops in the middle of the labyrinth.
c) Otherwise the labyrinth solver takes a shortcut.

9 Examination

EXERCISES I and II

Home exercises I and II approved by teacher?

DEMONSTRATIONS

Demonstrations approved by teacher?

QUESTIONS

	a	b	c
1:			
2:			
3:			
4:			
5:			
6:			
7:			
8:			
9:			
10:			
11:			

	a	b	c
12:			
13:			
14:			
15:			
16:			
17:			
18:			
19:			
20:			
21:			
(EXTRA) 22:			

	a	b	c
23:			
24:			
25:			
26:			
27:			
28:			
29:			
30:			
31:			
32:			
33:			

Questions approved by teacher?
