Image Analysis
Exercise 11 - Line and Path Tracing

2019

Introduction

The topic of this exercise is line and path tracing in images. In the first part, the goal is to identify straight lines using the Hough transformation. The second part is focused on finding the outline of the lower part of the liver as seen on computed tomography images.

As usual start by downloading the exercise data and start by creating a new m-file to contain the exercise scripts.

Part 1: Lines and the Hough space

A line can be represented in the so-called normal representation:

$$x \cos\theta + y \sin\theta = \rho \tag{1}$$

Figure 1 illustrates the geometric interpretation of the parameters θ and ρ. A vertical line has for example $\theta = 0^o$ and ρ being the x-intercept of the line.

A line is therefore a point in *Hough space* that has θ on the x-axis and ρ on the y-axis. A point can belong to many lines and is therefore seen as sinusoids in the *Hough space*.

Start by creating a little image with a single point:

```
a = zeros(50,50);
a(25,25) = 1;
imshow(a);
```

The Hough transform can be computed using the Matlab function `hough`.

Exercise 1 *Compute the hough transform of the image* `a` *and show the Hough space:*

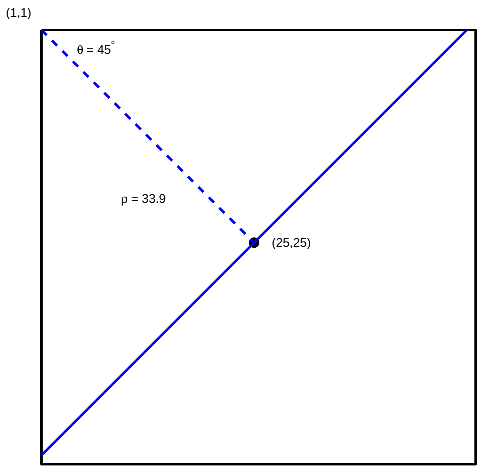

Figure 1: The line representation used in the Hough transform.

```
[H, T, R] = hough(a);
figure;
imshow(H, [], 'XData', T, 'YData', R, 'InitialMag', 'fit');
axis on
title('Hough space');
xlabel('\theta')
ylabel('\rho')
```

What do you see?

Exercise 2 *Compute and visualise the Hough space when the image* `a` *contains three points:*

```
a = zeros(50,50);
a(25,25) = 1;
a(5,5) = 1;
a(20,45) = 1;
```

Exercise 3 *Compute and visualise the Hough space when the image* `a` *contains another set of three points:*

```
a = zeros(50,50);
a(25,25) = 1;
a(15,15) = 1;
a(35,34) = 1;
```

Compare the result with the Hough space from the previous exercise. What is the significant difference?

The function `houghpeaks` can locate maxima in the Hough space.

Exercise 4 *Use* `houghpeaks` *to find the maximum in the Hough space from the previous exercise:*

```
P  = houghpeaks(H);
```

The maximum can be plotted in the Hough space:

```
hold on;
x = T(P(:,2));
y = R(P(:,1));
plot(x,y,'o','color','red');
```

Exercise 5 *Compute the slope-intercept equation of the line that corresponds to the found maximum in the previous exercise. Plot the line together with the three input points.*

Line detection in a real image

In this part of the exercise we will examine if the Hough transform can be used to locate the edges of signs in the photos from the DTU sign challenge.

You can start by using a photo with a very visible sign and even cut out a region of the photo where the sign is very visible. You can for example call the photo you will be working with for `DTUphoto.png`.

Start by converting your image into a gray-scale image.

As an initial step we apply edge detection on the image.

Exercise 6 *Read the image, filter, detect edges and display the filtered version:*

```
I  = imread('DTUphoto.png');
imshow(I,[]);

h = fspecial('gaussian', 3, 1);
I2 = imfilter(I,h,'replicate');

figure;
imshow(I2,[]);

figure;
BW = edge(I2,'prewitt');
imshow(BW);
```

Why do we apply a Gaussian filter before edge detection?

Exercise 7 *Compute the Hough transform of the edge image* `BW` *and show the Hough space.*

Exercise 8 *Find the maxima in the Hough space of* `BW` *using* `houghpeaks` *and mark it as in Exercise 4.*

The maximum can now be extracted using `houghlines` and shown as a line in the original image.

Exercise 9 *Compute the slope-intercept equation of the line that corresponds to the found maximum and plot the line on the input image.*

Exercise 10 *Extract the line and show it in the original image:*

```
line = houghlines(BW,T,R,P,'FillGap',1000);
figure;
imshow(I,[]);
hold on
```

```
% Plot line
xy = [line.point1; line.point2];
plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

% Plot beginnings and ends of line
plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');

hold off;
```

Several lines can be located simultaneous using `houghpeaks`.

Exercise 11 *Use* `houghpeaks` *to locate 6 lines:*

```
P  = houghpeaks(H,6);
x = T(P(:,2)); y = R(P(:,1));

imshow(H,[],'XData',T,'YData',R,...
           'InitialMagnification','fit');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;

plot(x,y,'o','color','red');
```

The function `houghlines` can also compute several lines based on the output of `houghpeaks`.

```
lines = houghlines(BW,T,R,P,'FillGap',1000);
```

Exercise 12 *Create a Matlab* `for` *loop that plots all found lines in the original image.*

The loop can start like:

```
figure, imshow(I), hold on
for k = 1:length(lines)
```

and should of course end with `end`*.*

To get the points of line number k the following syntax is used:

```
  xy = [lines(k).point1; lines(k).point2];
```

Did you find the lines you expected?

Exercise 13 *Experiment with different numbers of lines in* `houghpeaks`*. You can also try with an alternative edge detector or size of the Gaussian filter.*

You know that the signs looks red - so perhaps that can be used to pre-process the image before the Hough transform? Perhaps only looking at the red channel of the image?

Part 2: Finding the liver outline

The goal of this exercise is to locate the outline of the lower part of the liver.

We will work with computed tomography (CT) images of the liver region. They are `LiverTraining1-3.dcm` and `LiverValidation1-3.dcm`.

Exercise 14 *Start by examining the images and show them together. The images are stored in DICOM format and the pixel values are in Hounsfield units. You can use*

```
imshow(ct, [-1000,400]);
```

to scale the grayscale of the image. You can try to change the parameter (-1000 and 400) to make the anatomical structures more visible.

A set of scripts have been provided for the following exercises. We will try to use path finding by dynamic programming to locate the border. The method starts with a resampling of the image using a spoke based method.

Initially, we manually need to point out where the center of the liver is located.

Exercise 15 *Use the Matlab function* `ginput` *to manually mark the center of the liver. Put the found center in a variable called* `center`.

A function has been provided that can resample the image using the spoke resampling method.

Exercise 16 *Use the following to resample the image (ct is your original image):*

```
lSpoke = 110;
nAng = 360;
IM=double(ct);
[linesI, lCoords] = SampleSpokes(IM, nAng, lSpoke, center);
```

The arguments are the image, the number of spokes, the pixel length of the spokes, and the center of the organ.

The resampled spoke image is `linesI`. To be able to do path tracing the potential path should go from the top of the resampled image to the bottom of the resampled image.

Exercise 17 *Show the resampled image and comment on it. Can it be used to detect the border of the organ?*

Path tracing using dynamic programming computes the path with the lowest cumulated cost. The task is now to create a suitable cost image from the resampled image. A cost image should have a low value where the border of the organ of interest is.

Exercise 18 *Use image filtering to turn the resampled image into a suitable cost image. Describe the filter or the filters you are using and why you are using them. Also show the cost image.*

Exercise 19 *Find the path using*

```
Brd = doDP(costI);
```

Exercise 20 *The found path should be plotted on top of the resampled image. The path can be plotted using:*

```
plot(Brd,1:nAng,'-r','linewidth',2);
```

use `hold on` *and* `hold off` *to plot the path on top of the resampled image.*

A function is provided that can convert the found path into the original image coordinate system.

Exercise 21 *Use*

```
[ix,iy] = dpConvertPolar(nAng, lCoords, Brd);
```

to convert the found path into the image coordinate system. Plot it on the original image.

Exercise 22 *Comment on your found result. Did you successfully locate the liver boundary? Perhaps you should change the filters that were used to create the cost image.*

Exercise 23 *Train you algorithm using* `LiverTraining1-3.dcm` *and validate the performance on* `LiverValidation1-3.dcm`*. Did you manage to create a semi-automatic algorithm that can find the liver outline?*