Intelligent Systems Group
Universität Duisburg-Essen
Prof. Dr. Josef Pauli / Stefan Lörcks
WS 18/19
2017-11-16

Exercises Computer/Robot Vision - 05

PROGRAMMING (7 points)

In the last exercises we implemented two algorithms ourselves: The shrinking method as an example for active contours and the hough transform for straight lines. This was very useful to gain insights and get a deeper knowledge how these methods actually work. As both methods are well known for several years, they are also implemented in several libraries. This means in future work you do not have to implement them yourself, but can make use of existing implementations. Therefore, in this exercise we will get to know MATLAB's built-in functions for active contours and hough transform (for straight lines).

12. `activecontour` **by MATLAB** (2 points)
    This exercise introduces `activecontour`, a function provided by MATLAB to do segmentation by extracting the boundaries of objects. First you will read the documentation of the function, as this is the natural approach, if you want to use built-in functions. Afterwards you will apply the function on four test images.

    a) Read the documentation of the function `activecontour` [1]. Try out the first three examples:

    - Segment an Image Specifying the Mask
    - Segment Image Overlaying Mask and Contour on Original Image
    - Segment an Image Specifying a Polygonal Mask Created Interactively

    Make sure to have a look at the possible input arguments. The tips might also be useful.

    b) Create a script named

    `CRV_12.m`

    The first eight lines should look like:

```
%% CRV_12_activecontour
% name: John Doe
% student number: 11235813

%% clean up
clear all;
close all;
clc;
```

Of course again, fill in your name and student number!

c) Extract the boundary of the object in the first test image using the built-in function `activecontour`:

   i. Load the edge image ac01-star.png, which is available on moodle in (ActiveContourTestImages.zip).

   ii. If necessary do preprocessing.

   iii. Open a figure, use `roipoly` to let the user choose an initial curve and close the figure afterwards.
   Useful commands: `figure`, `imshow`, `roipoly`, `close`
   Forbidden commands: `ginput`

   iv. Use `activecontour` to extract the boundary of the object. Adjust the parameters.

   v. Create a maximized figure and plot the initial and the final curve on the original image.

d) Extend your code, such that all four test images are processed:

   2) The second test image ac02-SubjectiveContourIllusion.png shows the subjective contour illusion from the original paper regarding snakes [2]. The contour is defined by objects in the outer area. The human brain can imagine the boundary immediately. This is harder to achieve by computer vision techniques. Figure 1 shows the result of the original snake algorithm in the left part and the subjective contour illusion in the right part. Try to achieve a similiar result!

   3) The third test image ac03-prostate.png contains a MRI scan of the hip area. (The bright round areas at the left and right are the femur heads, the large dark area in the bottom part is the bladder.) A organ that is hard to locate is the prostate. Here, you should try to segment the prostate using active contours. The ground truth segmentation, provided by an expert, is shown in figure 2.

   4) The last test image ac04.png is not given. Take a picture yourself (or download something interesting from the web), convert it to gray scale and extract the boundaries of an object!

   Show all the results (initial and final curve on top of image) of all four images in ONE maximized figure. This means you can

   - Create eight subplots and show each image twice: once with the initial curve and once with the final curve.

   or

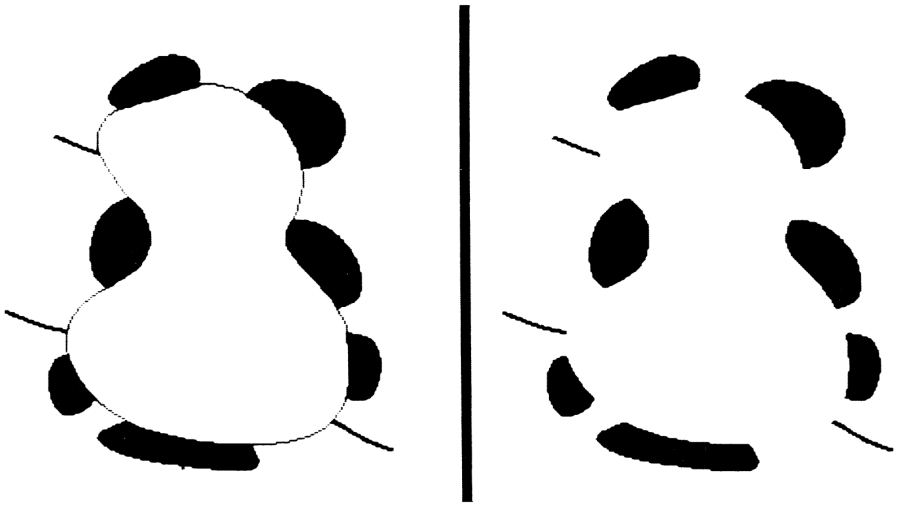

Figure 1: Subjective contour illusion (right) and the result of the snake algorithm. Source: [2].

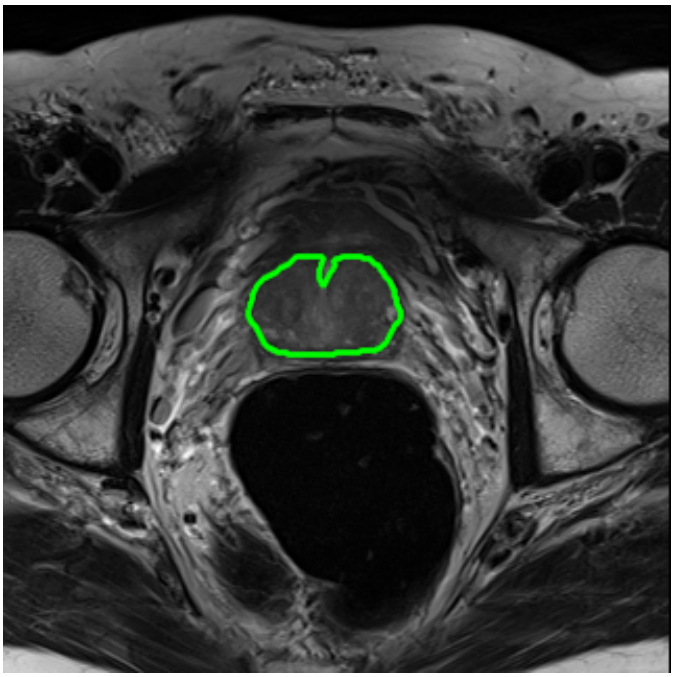

Figure 2: MRI scan of prostate with segmentation

- Create four subplots and show each image once with both curves (initial and final) visualized on it.

Save this figure as a png-file named 'contours.png'. For example it could look like figure 3.

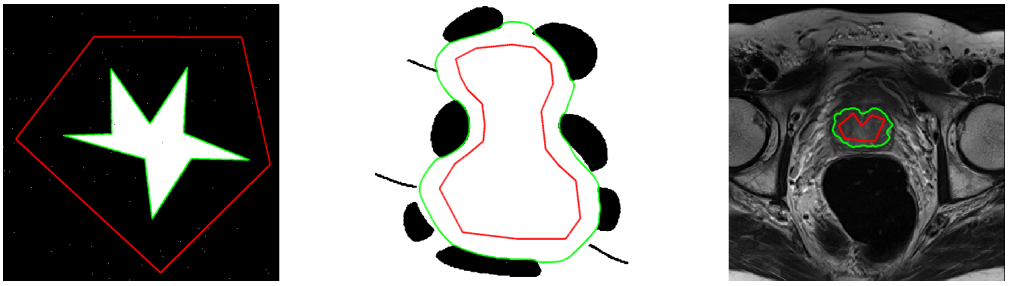

Figure 3: Possible result for 'contours.png' for first three test images. In your submission all four test images should be part of 'contours.png'!

Hint:
The following code snippet creates a maximized figure, loads some random content and saves the figure as 'MaxFig.png' with a resolution of 300 DPI:

```
f = figure('units','normalized','outerposition',[0 0 1 1]);
subplot(1,2,1);
imshow(imread('coins.png')); title('title1');
subplot(1,2,2);
imshow(imread('cameraman.tif')); title('title2');
print(f, 'MaxFig.png' , '-dpng', '-r300');
```

e) OPTIONAL: If you implemented the shrinking method in exercise 10, you can also compare the results: Apply the shrinking method on the test images of this exercise and compare the results.

13. `hough`, `houghpeaks` **and** `houghlines` **by MATLAB** (2 points)
This exercise introduces the functions `hough`, `houghpeaks` and `houghlines`, which are provided by MATLAB to do detect infinite and finite lines within images. We will use it to extract the finite lines in the images from last exercise sheet.

a) Read the documentation of the function `houghlines` [3]. Try out the example 'Find Line Segments and Highlight longest segment'. Make sure to have a look at the documentation of `hough` and `houghpeaks`.

b) Create a script named

`CRV_13.m`

The first eight lines should look like:

```
%% CRV_13_houghlines
% name: John Doe
% student number: 11235813

%% clean up
clear all;
close all;
clc;
```

Of course again, fill in your name and student number!

c) Use the two images edge01.png and edge02.png from exercise 11 and try to extract the finite lines (line segments). Create a maximized figure and visualize the detected line segments for both images.

References

[1] MathWorks Matlab Help. *activecontour: Segment image into foreground and background using active contour*. URL: `https://de.mathworks.com/help/images/ref/activecontour.html`.

[2] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. "Snakes: Active contour models". In: *International Journal of Computer Vision* 1.4 (1988), pp. 321–331. ISSN: 0920-5691. DOI: `10.1007/BF00133570`.

[3] MathWorks Matlab Help. *houghlines: Extract line segments based on Hough transform*. URL: `https://de.mathworks.com/help/images/ref/houghlines.html`.