

DSP2 SS2018 – Exercise 7: Circle detection

(Application of segmentation and edge detection)

Note: The file `main.cpp` shows the working principle of a circle detection using the Hough transformation. Please have a look at the file and read the comments.

PART A: Circle Hough Transformation

- The input images show circles in black and white. We can use the images for the Hough transformation without any preparation (brightness, contrast, blur).
- Hough transformation for circles:

1. Start with empty 3D accumulator $A[a,b,r] = 0$
2. Use the image with detected edges and do for each (not black) pixel
For a radius r do
For each θ with $\theta = [0^\circ, 360^\circ]$ do
 $a = x - r * \cos(\theta * \pi / 180^\circ)$
 $b = y - r * \sin(\theta * \pi / 180^\circ)$
 $A[a,b,r] += 1$

- We will use step sizes less than 1 for the accumulator cells and θ (see `main.cpp`) to increase the accuracy. Feel free to create your function without step size parameters (that means step sizes = 1) at first. If that works, implement the step sizes.
- Instead of one 3D accumulator, we will create many 2D accumulators (one accumulator for every radius).
- The goal of PART A is to show the Hough transformed images. You do not have to find circles yet.

Exercise: Implement the function “`void Segmentation::houghCircle(...)`” that calculates the accumulator (output image) from the edge image (input) for a **single, fixed** radius.

PART B: Find circles

- Different images with different number of circles and radii are given.
- To find the circles we have to create the accumulator with the function from PART A for different radii $r = [r_{\min}, r_{\max}]$. Local maxima in the accumulator represent the circle center.
- In `segmentation.h` a “CircleItem” item is defined. With the help of `std::vector` a list of these items can be created: `std::vector<CircleItem> circles;`. You will find examples in the code for how to use this list.
- Use the function `findAndRemoveMaximum(...)` to find circle centers in the accumulator.

Exercise: Implement the function “`void Segmentation::findCircles(...)`” that uses the function `houghCircle()` from PART A to create the accumulators, find circle centers and add the circles to the list.

PART C: Preparation of a picture for the use of the Circle Hough transformation

This is only a demonstration how a picture has to be prepared (brightness and contrast adjustment, applying Gaussian blur and find edges) to find the circles.