CS 675 – Computer Vision – Spring 2018

Instructor: Marc Pomplun

Assignment #3

Posted on April 3 – due by April 12

Question 1: Finding Circular Things with the Hough Transform

We already discussed this topic in class: The Hough transform can be used to detect contours of any shape. An easy shape to deal with in terms of programming is the circle. Your task is to modify the netpbm_hough.c code so that it detects circles of any (reasonable) size and location in a given input image. As we figured out in class, this requires the estimation of three parameters: the vertical and horizontal positions of the circle center and the radius of the circle. Therefore, you will have to use a three-dimensional output space for the modified Hough transform. You should define your own data type for this purpose; basically, it will be a three-dimensional array.

The output space should be of reasonable size, otherwise computation could take a long time. Please find a nice test image that contains some near-perfect circles. The desk.ppm image on the course homepage may work but is probably not ideal. Let your algorithm detect as many circles as there are in the image, and submit your code, the input image, the Hough-transformed image with marked maxima, and the input image with the detected circles being marked. To visualize the Hough-transformed image, you can simply show a 2D projection. The easiest and most illustrative way to do this may be to simply ignore the radius parameter and just show the two-dimensional space spanned by the vertical and horizontal circle center positions.

Question 2: Noise Removal à la Fourier

On the course homepage, you will find an image file named “umb_noisy.pgm.” It shows our very own university in very poor quality. It seems that there is some rather regular noise pattern that would be very difficult to remove in the spatial domain, but maybe there is a chance to do so in the frequency domain? Use the “fourierDemo” and “invFourierDemo” functions in the netpbm_fourier.c file (also available on the course homepage) to remove the noise as much as possible. Describe the steps that you took. Do not submit any code but submit the cleaned image and images documenting the steps you took, such as Fourier spectra.

Question 3: Split and Merge

Apply the Split-and-Merge algorithm to find homogeneous regions in the following 8×8 image, where numbers indicate the brightness of the pixels. Your homogeneity criterion is that intensities must not vary by more than one brightness unit. For example, values 8 and 9 could occur in the same region, whereas 8 and 10 could not. Use as many of the instances below as you like to show how the algorithm works. Splitting should be shown in detail, merging could be shown in a single step.

9	9	9	9	9	9	9	8
9	9	9	9	9	9	3	3
9	9	9	9	9	1	3	3
9	9	9	9	9	1	4	4
2	8	8	4	4	4	4	4
7	7	7	7	2	2	4	4
8	8	4	4	3	6	6	6
0	0	0	0	3	6	6	6

(a)

9	9	9	9	9	9	9	8
9	9	9	9	9	9	3	3
9	9	9	9	9	1	3	3
9	9	9	9	9	1	4	4
2	8	8	4	4	4	4	4
7	7	7	7	2	2	4	4
8	8	4	4	3	6	6	6
0	0	0	0	3	6	6	6

(b)

9	9	9	9	9	9	9	8
9	9	9	9	9	9	3	3
9	9	9	9	9	1	3	3
9	9	9	9	9	1	4	4
2	8	8	4	4	4	4	4
7	7	7	7	2	2	4	4
8	8	4	4	3	6	6	6
0	0	0	0	3	6	6	6

(c)

9	9	9	9	9	9	9	8
9	9	9	9	9	9	3	3
9	9	9	9	9	1	3	3
9	9	9	9	9	1	4	4
2	8	8	4	4	4	4	4
7	7	7	7	2	2	4	4
8	8	4	4	3	6	6	6
0	0	0	0	3	6	6	6

(d)

9	9	9	9	9	9	9	8
9	9	9	9	9	9	3	3
9	9	9	9	9	1	3	3
9	9	9	9	9	1	4	4
2	8	8	4	4	4	4	4
7	7	7	7	2	2	4	4
8	8	4	4	3	6	6	6
0	0	0	0	3	6	6	6

(e)

9	9	9	9	9	9	9	8
9	9	9	9	9	9	3	3
9	9	9	9	9	1	3	3
9	9	9	9	9	1	4	4
2	8	8	4	4	4	4	4
7	7	7	7	2	2	4	4
8	8	4	4	3	6	6	6
0	0	0	0	3	6	6	6

(f)

Question 4: Growing Quad Trees

Write down the quad tree for the following image:

	■	■			■	■	
	■				■		
			■	■			
	■					■	
		■	■	■	■		