Digital Image Processing
Prof. Kasaei
Due 17 Khordad

Homework 5
Image Analysis

Notes:

- Problem 1 and 2 are theoretical and the rest of the problems are practical.
- If a question asks you to run the code on images, please save your results in a folder. Or you can use jupyter notebooks and leave the results in the notebook.

If you have any questions, please contact me at msdkhairi@gmail.com

Problem 1: Morphological Analysis (25 points)

In this problem, different aspects of the morphological analysis will be investigated.

(a) Let A be a structuring element that contains one point valued 1. What happens if any set of points B is dilated by A?

(b) Opening and closing operations are need in what scenarios?

(c) Let A be a set of points. define complement of A as $A^c = \{w|w \notin A\}$. Show that the following equation holds: $(A \ominus B)^c = A^c \oplus \hat{B}$

(d) Prove that duality holds for closing and opening that is: $(A \bullet B)^c = A^c \circ \hat{B}$

Problem 2: Hough Transform (25 points)

Hough Transform is used to detect shapes such as lines and circles. Answer the following questions.

(a) What is the hough transform of point origin $(0,0)$? (It should be a straight line)

(b) What set of points have the hough transform same as (a)?

(c) Develop a general algorithm using hough transform to obtain normal representation of a line $y = ax + b$.

(d) Consider the coordinate system (θ, ρ). For a point (x_i, y_i), we can plot a curve in the (θ, ρ) plane using the equation: $x_i cos(\theta) + y_i sin(\theta) = \rho$. Find the *min* and *max* values of θ and ρ for all the visible lines on an image.

(e) Show that for any two points (x_i, y_i) and (x_j, y_j) where $i \neq j$, the corresponding curves in the (θ, ρ) plane of these points intersect at exactly one point.

(f) Show that for any set of points S that are on the same line in an image plane, their corresponding curves in the (θ, ρ) plane will intersect at exactly one point.

Problem 3: Morphological Analysis [practical] (30 points)

In this problem, we will perform different morphological operations on different images. you are free to use any library you want.

(a) Load and show the `hole.jpg` image.

- Perform dilation and erosion on this image using two different structuring elements and report the results. The size of the structuring element is of your choice.
- Perform open and close operations on this image again using the same structuring elements. The size of the structuring element is of your choice. Report and explain the results.
- Using morphological operation, try to fill the hole as best as possible. explain and report your result.

(b) Load and show the `noisycameraman.jpg` image and `cameraman.jpg` image side-by-side. As you can see, we have added salt noise to the `cameraman.jpg` image. now using morphological operations try to reduce the noise as much as possible without distorting the image. report and explain your result.

(c) Load and show the `circles.jpg` image.

- plot the histogram of this image.
- in this image, you see a number of dark circles in a light background. first do a thresholding and remove the background, leaving only the circles as white and background as black in a binary image. show this result.
- now use a morphological operation to separate circles from each other such that they won't touch each other anymore. explain and report your result.

(d) Load and show the `lines_circle.jpg` image. In this image, you see some lines and circles together. separate the lines from circles as best as possible using morphological operations. explain what operations you used and report two images, one that only contains circles and another that only contains lines. show these images side-by-side.

Problem 4: Hough Transform [practical] (20 points)

In this problem, we want to implement Hough Transform and then compare our results with OpenCV implementation.

(a) Load and show the `coins.jpg` image.

(b) Implement Hough Transform and show the detected lines using it on the `coins.jpg` image.

(c) Do the same as in (b) only using OpenCV built-in function for Hough Transform. and finally compare your results with this part side-by-side.