Exercise 2: Basics (2) - Discrete Geometry and Image Operations

2.1 Discrete geometry

Exercises

2.1 Create a binary image with Matlab and perform the following operations

1. *shrink*
2. *expand*
3. *labelling*.

2.2 Connectivity: In an orthogonal grid, sketch binary image objects that are

1. 4-adjacent,
2. 8-adjacent and
3. m-adjacent.

Specify the path lengths of the image objects.

2.3 Sketch a binary image object and state the following geometrical properties:

1. Area
2. Contour length (different dimensions).

2.4 Euler number: Draw examples of binary pictures whose topological properties are indicated by the corresponding Euler number E:
a) E = 2, b) E = 1, c) E = 0, d) E = -1, e) E = -2.

1. What is the Euler number of the letters A, B, C and D ?
2. Draw the corresponding adjacency trees.

2.5 Convex objects (discrete plane): A discrete object O can be said to be convex, for example, if there is at least one digital straight chord segment $S(A,B)$ between any two pixels A and B ($A, B \in O$) that is entirely contained in O.

1. Draw examples of discrete and analogue objects that are convex and non-convex (concave) respectively.
2. Investigate a method for parallel determination of con vexed image objects by examining D_8 neighbourhoods in sufficient form.

2.2 Elementary image operations

Exercises

2.6 Binarisation
Load an image (e.g. `coins.png`) and binarise the image so that the coins characters appear as (black objects) on a white background in the binary image.
Solve the binarization using "own" coding and using build in functions.

Matlab hint: `im2bw()` (deprecated) or `imbinarize()`; alternatively determine a threshold value automatically with `graythresh()` (see 2.7) .

2.7 Threshold determination
Determine a suitable threshold value. You can use the histogram for this purpose.

You can carry out the threshold determination automatically using build in functions. Check the help (`doc`) which algorithm is used by the build in function and how this algorithm works.
Think about how an own algorithm do to so could work? What are statistical / formal arguments supporting that your algorithm is reasonable?

Matlab hint: `imhist(), histogram(), graythresh()`

2.8 Binarisation using precomputed Threshold value
Perform binarisation usni the threshold computed in 2.7 and create a result image. Check its technical properties.

2.9 Multi-Level Thresholding
Load the image `circlesBrightDark.png` and theshold it into three classes. How is this possible?
Matlab hint: `multithresh()`

2.10 Carry out simple and elementary gray value manipulations (unary point operations). Consider the corresponding transformation functions for the performed operations, outline the transformation functions and briefly explain the effect and possible practical applications:

1. Image inversion
2. gray value spreading or compression
3. Histogram linearisation

Describe these operations formally and then realize them in code using Matlab.

2.11 Load the image `hawkes_bay_raw.jpg` provided in moodle. It has low contrast. Implement an histogram equalization manually coding each needed step without using build in high level functions. Compare your result with `histeq()`.

Hint:

- Code a function to get the image statistics (histogram)
- Derive a suited LUT from the image statistics encoding a transfer function that carries out a histogram equalization (hint: cf. definition of CDF and PDF)
- Apply this LUT to the image