

TNM087 - Image Processing and Analysis

Task 1 - Basic Image Info

Background:

In Matlab gray value images are matrices and color images are three-dimensional arrays. In the following we will use the terms ‘column’ and ‘row’ in connection with images to mean a line in y-direction (column) and a line in x-direction (‘row’). Note that the first index in a matrix is the row index and the second the column index. Therefore: image(m,n) is the pixel at position (n,m) in the gray value image if we use the (x,y) = (width, height) coordinates for images and the (row, column) convention for matrices.

Basic commands handling image files are imread, imwrite and imfinfo

Ordinary jpg image have 8bits/pixel/channel, tiff images can have 8bits/pixel/channel or 16bits/pixel/channel

Typical Matlab commands to display images are imshow, image, imagesc, imshowpair and for more advanced operations imtool. Read the Matlab documentation for details

The coordinates of the cursor can be captured with the ginput command.

Task:

Write a Matlab function that extract basic information about an image. Details are in the template file

Syntax:

`function [ImSize, ImType, BitPerPixel, MaxMin, RGBpts, figh] = ...`

`BasicImageInfo(filename, nopts)`

Hints:

Have a look at the Matlab introduction in the lecture

Read the documentation of some basic tools in the Image Processing Toolbox

The template file is Task1BasicImageInfo.m

Save your solution as BasicImageInfo.m and submit only the m-file

Your code must work for all filenames that point to legal image files (and all image formats understood by Matlab, such as jpg, tif, etc.)

Details:

- `%% Collect image information with imfinfo`
The input variable `filename` is the path to the image file. You have to use `imfinfo` to extract the necessary information in the next code lines
- `%% Compute minimum and maximum values`
Read in the image and compute the minimum/maximum values in `OImage`. If `OImage` is a color image with $M \times N$ pixels then you have $M \times N \times 3$ numbers and the minimum and the maximum value are minimum and maximum values of all these $M \times N \times 3$ numbers. Use only one line of code for the minimum (and one for the maximum) computation.
Note: You can compute both, the minimum and the maximum value with a single command. Optional find out how!
- `%% Pick the pixel positions and collect the RGBvectors`
If `nopts = 2` and the image is an RGB image then `RGBpts` should be of size 2×3 if it is a gray value image then it could be of size 2×1 or 2×3 depending on your choice. Add a comment motivating your decision
- `%% Generate the white squares and display the result`
Select the size of the square (say 5×5 or 7×7 pixels) for every point that you picked with `ginput` you should select all points in the square (centered at those points) and change their color to white. Max 3 lines of code per point should be enough.