Image Analysis
Exercise 01 - Digital Images

2019

Introduction

This exercise introduces Matlab as a tool for image analysis. Basic knowledge of Matlab is assumed.

It is important to use the Matlab `help` and `doc` command to understand Matlab functions and so on. Try looking at the help for `imread` for example:

```
doc imread
```

Data material

The data and material needed for this exercise can be downloaded from

`http://courses.compute.dtu.dk/02502/`.

Start by creating an exercise folder where you keep your Matlab scripts. Download the data and place them in this folder.

Basic Image Handling

Before starting a new exercise, clear the MATLAB workspace and close figure windows by:

```
close all;
clear;
```

In this exercise we will examine a part of an X-ray image of a hand. X-ray examinations are extremely common and are used for example for bone fracture detection. However, X-rays can also be used for diagnosis of osteoporosis (age related loss of bone mass). Initially, we will look at the four metacarpal bones.

Exercise 1

Start by reading the image into your Matlab workspace:

```
mc = imread('metacarpals.png');
```

(always remember the ";" after the line when reading images)

then check the image dimensions by writing:

```
size(mc)
```

Also check the contents of the workspace and see how the image is represented:

```
whos
```

Exercise 2

Display the image by writing:

```
imshow(mc);
```

Use the zoom tool to inspect the metacarpals. Try to explain why the color of the bone is more light at the edges of the metacarpals.

Exercise 3

Compute and visualise the histogram of the image:

```
imhist(mc);
```

The bin values of the histogram can also be stored by writing:

```
[counts,x] =  imhist(mc);
```

What is the most common pixel value in the image? (hint use plot and zoom in the plot)

It can also be found using:

```
find(counts == max(counts))
```

Exercise 4

The value of a pixel can be examined by:

```
mc(r,c)
```

where r and c are the row and column of the pixel. What is the pixel value at (r,c) = (100,90) ?

Note that pixels can be indexed both by (row, column) and by (x,y) and that (x,y) = (column, row). (1,1) is placed at the upper left corner in both systems.

Exercise 5

Matlab has a simple image analysis tool that can be used to explore images. Start the `Image Tool` by writing:

```
imtool(mc);
```

Try to zoom on a part of the bone and use the `Inspect Pixel Values` tool to look at the pixel values.

Which region of the image (background, bone, edge of bone) corresponds to the highest peak in the histogram?

Working with your own image

It is now time to work with one of your own images. It is assumed that you know how to either save an image from a digital camera on the disk or download an image.

First copy your image to your Matlab folder. If you do not have an image you can use the `horns.jpg` example image.

Exercise 6

Start by reading it into a Matlab variable called I and examine the size of it. If it is wider than 1000 pixels use `imresize` to resize it to less than 1000 pixels in either width or height.

```
I  = imread('horns.jpg');
I2 = imresize(I, 0.25);
imshow(I2);
```

Here we selected a scale factor of 0.25. Explain how this factor can be calculated automatically? (hint use `size` and perhaps `max`).

Exercise 7

The image we are now working with is a color image. That means that each pixel is defined using three values R (red), G (green), and B (blue). Let us examine a single pixel using the `impixel` command. `impixel` uses (x,y) coordinates.

```
impixel(I2, 500, 400)
```

The pixel values can also be inspected by (in row,column coordinates)

```
I2(400,500,:)
```

To be able to work with the image, it can be transformed into a gray-level image:

```
Igr = rgb2gray(I2);
figure;
imshow(Igr);
```

Inspect the gray value of the same pixel as before. Try to explain how it was computed.

Exercise 8

Compute and show the histogram of you own image.

Exercise 9

Take an image that is very dark (for example by disabling the blitz) and another very light image. Compute and visualise the histograms for the two images. Explain the difference between the two.

DICOM images

Exercise 10

Typical images from the hospital are stored in the DICOM format. An example image from a CT angiographic examination of the liver is found in the training image directory. It is named `CTangio.dcm`.

Start by examinating the header information using:

```
ctInf = dicominfo('CTangio.dcm');
ctInf
```

who has made the used scanner?

This image has been *anonymised* so patient information has been removed. Else the patients name and diagnosis is sometimes also available.

Now read the image using the DICOM image reader.

```
ct = dicomread('CTangio.dcm');
```

Use `whos` to see how the pixels are stored.

Try showing the image using `imshow`. As seen the contrast is very bad. This is due to the scaling of the image pixel values and because they are not (as usual) 8-bit values.

Use `imtool` to open the image

Use the menu option Adjust Contrast to play around with the colors. See if you can make a nice looking image by adjusting the sliders.

Color Maps

Exercise 11

When working with graylevel images they are viewed using 256 levels of gray. There is another method of viewing them using a *Color map* where each gray level is shown as a color. Matlab has several predefined color maps. Try

```
mc = imread('metacarpals.png');
imshow(mc);
colormap(gca, jet);
```

Experiment with different colormaps. For example `cool`, `hot`, `pink`, `copper`, and `hsv`. Use

```
doc colormap
```

to get a list of choices.

Colour channels

We are now going to look at the intensity values of the different channels of an colour (RGB) image taken at DTU compute.

Start by reading and showing the `DTUSign1.jpg` image.

You can visualise for example the R component of the image using:

```
Rcomp = im1(:,:,1);
figure;
imshow(Rcomp);
colormap(gca, gray);
```

Do this with the R, G, and B components.

Why does the DTU Compute sign look bright on the R channel image and dark on the G and B channels? Why do the walls of the building look bright in all channels?

Simple Image Manipulation

In this exercise we simply change the pixel values in an rectangle of the image.

Start by reading and showing the `DTUSign1.jpg` image. Now create a black rectangle in the image, by setting all RGB channels to zero:

```
im1(500:1000,800:1500,:)=0;
```

Show the image again and save it to disk as `DTUSign1-marked.jpg` using:

```
imwrite(im1,'DTUSign1-marked.jpg');
```

Try to save the image using different image formats like for example PNG.

Try to create a blue rectangle around the DTU Compute sign and save it.

Advanced Image Visualisation

Before doing image analysis it is very important to get an intuitive understanding on how the image *looks as seen from the computer*. The next set of tools can help to gain a better understanding. In this example we will work with an x-ray image of the human metacarpal. It is a hollow bone and we want to understand how a hollow structure looks on an image. Start by reading the image

```
fing = imread('finger.png');
```

To get an understanding of the grey levels in the image, a contour plot can be used that visualise the *iso-lines* of the grey levels:

```
imcontour(fing, 5);
```

It can clearly be seen that the borders of the bone is on an iso-line - but what about the background. Why are there so many iso-lines in the background?

To investigate the properties of the hollow bone a grey-level profile can be sampled across the bone. Use the tool `improfile` to sample profile across the bone:

```
imshow(fing);
improfile;
```

Start the profile with a single mouse-click and end it with a double click. What do you see - can you recognise the inner and outer borders of the bone-shell in the profile?

Finally, the image can be plotted as a landscape, where the height is equal to the grey level:

```
mesh(double(fing));
```

Use the mouse to rotate the mesh and find a good viewpoint. Notice how the bone looks like a valley seperated by two ridges. We will use this fact later in the course to measure bone thickness.

Matlab commands and help

The documentation for the *Matlab image processing toolbox* can be found at `http://www.mathworks.se/help/images/index.html`.

The following commands are useful:

Command	Description
imread	Reads an image from disk
imshow	Show the image
size	Size of image
impixel	Show the pixel value at (x,y)
imfinfo	Shows information about an image file
imhist	Computes and shows image histogram
rgb2gray	Converts RGB color image to a gray level image
type	Shows the implementation of a function
figure	opens a new figure window
imresize	resize an image
dicomread	read DICOM images
dicominfo	get header information from DICOM image
colormap	change the colormap of the currently shown image

An overview of all image functions can be obtained by writing:

```
help images
```

use

```
doc imread
```

to get a window with help.