

Image Processing Matlab tutorial 4 Histogram Manipulation

1. Objectives:

We will investigate the properties of the histogram and its various uses for image pre-processing, visualisation and segmentation.

During this session, you will learn & practice:

- 1- More programming in matlab
- 2- More examples of functions and scripts
- 3- Histogram manipulation in matlab
- 4- Simple segmentation techniques

2. Resources required:

In order to carry out this session, you will need to download images and matlab files from the Image Processing module on vision:

<http://vision.hw.ac.uk>

Please download the following images:

- Image of Lena
- Landsat image

We can also use other images – for example circuit image. (circuit.tif). Note that a full list of all the images supplied as part of matlab can be accessed through `help images/indemos`.

You are of course encouraged to try these programs out on images of your choice.

3. Histogram manipulation:

As already seen in class, the histogram of an image is a simple way to see how the gray level (i.e. the information) is spread over the available range of grey levels. This in turn can be used to highlight problems during the acquisition (clipping due to over exposure, low contrast). The histogram can also be modified to make better use of the available range for further processing and display.

There is another way to see the histogram in the light of probability theory. An image can be seen as a random field (array) of pixels, each pixel being a random variable. Assuming these pixels follow the same statistical law, then the histogram can be interpreted as an estimator of the probability density function (pdf) of this law. This fact has been used for texture

classification using local histograms for example (here, the texture are modelled locally by the histogram).

Matlab has several functions available to calculate and process histograms:

- `imhist`: Display the histogram of an image.
Syntax: `hist = imhist(ima,nb_boxes)`.
`imhist` works with `uint8` types images.
- `histeq`: histogram equalisation and specification.
Syntax: `ima1 = histeq(ima,hgram)` where `hgram` is a specified histogram. If `hgram` is not present, histogram equalisation is performed.

To Try:-

Please load an image (for example `landsat.tif`).

Display the image in a figure;

Now use the `imhist` and `histeq` functions

Comments?

Use the `histeq` function to specify histograms for various images and compare the result with the desired histogram.

Can you explain the difference?

Can you improve the results?

What is the influence of the number of boxes?

Now have a look at the `imadjust` function and see how it is related to `histeq`.

4. Matlab Demos

Matlab has a number of demos for Image Enhancement. Investigate the use of Histogram Equalisation and Simple transformation functions using the demo “**Intensity Adjustment and Histogram Equalisation**”. (At the prompt type `demodemos` and select the Image Processing Toolbox)

Also investigate the demo “**Contrast Enhancement Techniques**”. Work through the steps provided to see the use of the `imadjust`, `histeq` and `adapthisteq` functions.