IMAGE PROCESSING (RRY025)

STUDIO EXERCISES

Lecture 14, Image Compression -II

On the windows system to use m-files written for the course you must in windows after starting MATLAB type

```
addpath \\mfil.me.chalmers.se\course-err041
```

If you are using the linux operating system after starting MATLAB you should type

```
addpath /chalmers/groups/course-err041
```

EX 1. Run length Coding

In matlab load the binary image 'text' **text = imread('text.tif')**. Display this using **imshow(text, [])**.

Run **rlc1(text)**, which calculates the distribution of run lengths of black pixels in the binary image (a white pixel is represented as a run length of zero). **Questions** Using the information given calculate the degree of lossless compression possible by just applying run length coding to this image.

If in addition source encoding (using short codes to represent common run-lengths and long codes to represent uncommon ones) is used how much can the runlengths code be compressed? What is the overall compression achieved if we combine run-length and such source/entropy coding?

EX 2. Predictor-Error Compression

Load a version of the cameraman image with **cam=imread('cameraman.tif')**. Type **dcam = difim(cam);** to calculate a version of the input image where each pixel is differenced with with the previous pixel on the same row. i.e. it uses a predictive 'Mapper' and a prediction rule which is the difference between pixels. Calculate the entropy of this difference image using **calcent(dcam);** Given that the input image has 256 gray levels (8 bits) what degree of lossless compression is possible using single pixel entropy coding?

Now type **gcam = gradim(cam);** which predicts each pixel based on the local gradient, calculated from the two previous pixels on the line. Type **type gradim** to see how it works. Plotted is the error image (true pixel value minus prediction). Using **calcent(gcam);** find the entropy of the error image and calculate the degree of lossless compression using this predictor rule.