The Haar Transform & Image Compression [1]

Dr. François Pitié

Electronic and Electrical Engineering Dept.

pitief@tcd.ie www.mee.tcd.ie/~sigmedia

THIS LABORATORY SESSION COUNTS FOR 10% OF THE TOTAL 4C8 MARK

Lab Information

Submission Requirements

In this lab, you will be required to compile a report that answers specific instructions included in the report. The report should be typed up and submitted electronically using the **PDF file format** via the module's blackboard page by the specified deadline. Remember to put your name and student number on the top of your reports.

Penalties for Late Submission

- less than 15 minutes past the deadline - no penalty.
- greater than 15 minutes and less than 2 hours past the deadline - loss of 25% of the final mark.
- greater than 2 hours past the deadline - loss of 50% of the final mark.

Plagiarism Policy

Any submitted code or answers that can be seen to be plagiarised from other sources will result in 0 marks being awarded for that section of the lab and may result in you being awarded 0 marks for the entire assignment. Anti-plagiarism software will be used on submitted materials.

[1] This lab was originally written by Prof. Anil Kokaram and ammended by Dr. David Corrigan.

Test image

Please use the following image as the test image in this assignment. The image can be downloaded from the following link: `https://www.hlevkin.com/TestImages/girlface.bmp`.

1 Recalling your knowledge of Variable Length Coding

In an attempt to compress an image by coding the intensity of each pixel in that image, the intensity was first quantised to 6 levels. For that image, the values and probabilities of these levels are given in the table below.

Level	Intensity Value	Probability
1	0	$\frac{5}{8}$
2	43	$\frac{3}{32}$
3	86	$\frac{3}{32}$
4	129	$\frac{1}{32}$
5	172	$\frac{1}{8}$
6	215	$\frac{1}{32}$

Q1a Calculate the Entropy of this quantised image.

Q1b Using the Huffman Coding method, design a set of variable length codewords for this image. Include your Huffman Coding Tree in your report.

Q1c Calculate the average codeword length using your designed codewords. Does it agree with the Entropy of the events in bits/pel i.e. $-\sum_i p_i \log_2(p_i)$? Explain any discrepancy.

2 Calculating Image Entropy

Q2a Write a MATLAB function `calcEntropy` that calculates the Entropy of an image. The function definition is below:

```
function entropy = calcEntropy(Y)
%This function takes as input a 2D array Y containing
%the image intensities and returns the entropy.
```

Save this function in a m-script called `calcEntropy.m` to allow you to call the function by the name `calcEntropy`. With this function in your m-file you can now calculate entropies simply by doing `e = calcEntropy(Y)` for instance.

Your first step will be to calculate the histogram of the greyscale image `Y`. The histogram should contain bins ranging from the minimum value of the input image to the maximum value of the input image in increments of 1 (assuming values of `Y` range from 0 to 255). Then use this histogram to compute the entropy in bits/pel.

Remember that in Matlab `log2(x)` returns the logarithm to the base 2 of x. If x is a vector it returns a vector containing the $\log_2$ result for each element. Also, by definition $0 \times \log(0) = 0$, so make sure your code does the same.

Q2b Estimate the entropy, H_o, for the unquantised image that has been given to you.

Q2c Now edit your m-file to quantise the image using your assigned Q_{step} value. e.g.

```
pic = Q_step*round(pic/Q_step);
```

Estimate the entropy (H_{qi}) of the quantised image using the given `Q_step` value.

Q2d Explain any difference between H_{qi} and H_o.

Q2e Write a MATLAB function to calculate the MSE between 2 images. The function definition should be as below:

```
function MSE = calcMSE(Y1, Y2)
% This function takes as input two 2D array Y1 and Y2 containing
% the image intensities of two pictures and returns the mean square error
% between both Y1 and Y2.
```

Q2f What is the MSE between the quantised and unquantised images?

Q2g Comment on the visual quality of the quantised and unquantised images.

3 The 2D Haar Transform

The basic idea in Transform image coding is to provide a compact representation for the image data. You will verify this with the 2D Haar Transform in this section.

Q3a write a MATLAB function that implements the 1-level Haar Transform and outputs a image of its subbands. The function definition should be as follows:

```
function H = calcHaarLevel1(Y)
% This function takes as input a 2D array Y containing
% the image intensities of a picture and returns the 1-level
% Haar Transform
```

Q3b Using the Matlab function that you have written to implement the 1-level Haar Transform, and using your assigned quantisation step size, calculate the resulting Entropy H_{qhaar} of the transformed image after quantisation has been applied.

Q3c Is $H_{qhaar} < H_{qi}$? Why / Why not?

Q3d To check the quality of the image after compression, the only thing to do is to reconstruct the picture using the `calcInvHaar` function. Compress and reconstruct your image using stepsizes of $Q_{step}/2$, Q_{step} and $2 \times Q_{step}$ and comment on the differences you see in the pictures compared to the original image **and** the quantised image from Section 2.

Q3e Calculate the entropy of the transformed and quantised images and the mean squared error between each reconstructed image and the original image. Comment on the relationship between the entropy and the objective quality metric for the 3 quantisation step sizes.

Q3f Using the same quantisation step sizes, calculate the MSE for the case where quantisation is performed on the image directly (ie. no haar transform is applied). Does the MSE metric correctly rank the perceived quality of the 6 compressed images (3 quantisation levels with/without transformation into the Haar domain)? Explain your answer.

4 The multi-level 2D Haar Transform

Q4a write a MATLAB function that implements the n-level Haar Transform and outputs a image of its subbands. The function definition should be as follows:

```
function H = calcHaar(Y, n)
% This function takes as input a 2D array Y containing
% the image intensities of a picture and returns the 1-level
% Haar Transform
% n is the number of levels used.
```

To implement this function you can adapt your level 1 implementation and recursively call the Haar Transform on the LoLo subband. You can test that your function works by checking the results on the provided test image.

Q4b Calculate the multi-level Haar Transform of your image and the reconstructed image for level numbers from 1 to 5 using your quantisation step size. Calculate the entropy of the quantised transformed image. Comment on the how the number of levels chosen affects the entropy and image quality.