# Homework #7

## Due Date: 11:59pm Tuesday 13 May 2014

In this homework you will write a C++ program that can compress images using the Haar Wavelet Transform. Your program will read the image on `stdin` and outputs the resulting image on `stdout`, and get in the required operation on the command line.

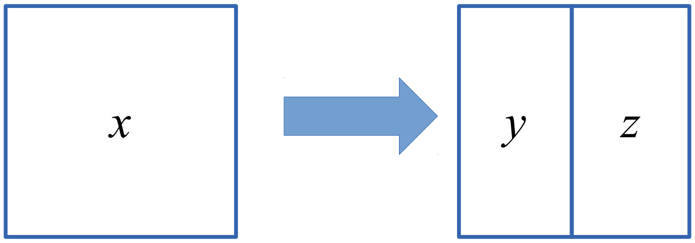
Please present a report containing your answers as well as a zip file containing all your code.

1. [4 points] Implement the functions `haar_forward_horizontal()`, `haar_inverse_horizontal()`, `haar_forward_vertical()`, and `haar_inverse_vertical()` to compute 1D forward and inverse Haar Wavelet Transforms on the rows/columns of an image. For an input $n \times n$ image $x$:

   Try the functions on the small images `msg1.pgm` and `msg2.pgm` and make sure they produce the correct output and that the inverse does produce the original input.

   - The Haar forward horizontal transform works on each row and produces two components:
     - The approximation $y[i,j] = \frac{x[i,2j]+x[i,2j+1]}{\sqrt{2}}$
     - The detail $z[i,j] = \frac{x[i,2j]-x[i,2j+1]}{\sqrt{2}}$

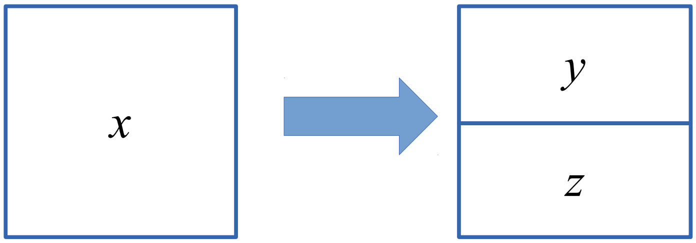
     The output coefficients are then the concatenation of $y$ and $z$. Note that $y$ and $z$ each has the same number of rows as $x$ but half the number of columns of $x$.



   - The Haar inverse horizontal transform works on each row and reconstructs the odd and even columns of $x$ as follows:
     - The *even* columns $x[i,2j] = \frac{y[i,j]+z[i,j]}{\sqrt{2}}$
     - The *odd* columns $x[i,2j+1] = \frac{y[i,j]-z[i,j]}{\sqrt{2}}$
   - The Haar forward vertical transform works on each column and produces two components:

- The approximation $y[i,j]=\frac{x[2i,j]+x[2i+1,j]}{\sqrt{2}}$
- The detail $z[i,j]=\frac{x[2i,j]-x[2i+1,j]}{\sqrt{2}}$

The output coefficients are then the concatenation of *y* and *z*. Note that *y* and *z* each has the same number of columns as *x* but half the number of rows of *x*.



- The Haar inverse vertical transform works on each column and reconstructs the odd and even rows of *x* as follows:
  - The *even* rows $x[2i,j]=\frac{y[i,j]+z[i,j]}{\sqrt{2}}$
  - The *odd* rows $x[2i+1,j]=\frac{y[i,j]-z[i,j]}{\sqrt{2}}$

2. [2 points] Implement the functions `haar_forward()` and `haar_inverse()` to implement 2D forward and inverse Haar Wavelet Transforms on an image. You could implement that by first transforming in the horizontal then in the vertical, or the opposite. But make sure in the inverse you do the operations in the right sequence.

   Try the functions on the small images `msg1.pgm` and `msg2.pgm` and make sure they produce the correct output and that the inverse does produce the original input.

3. [3 points] Implement the function `compress_haar()` to compress and reconstruct an image using 2D Haar Transform. Try it on the four images (sena, sensin, earth, omaha) using different threshold values (*threshold* = 0, 0.5, 1, 10), and compute the SNR of the reconstructed image. What do you notice? Include sample output quantized images in your report.

## *Command Line*

You need to modify the main file hw07.cpp to include the required functionality. Your program should be named hw06, and should be called as follows:

- To compute horizontal and vertical forward Haar transforms on an image and output the coefficients on the stdout:

  ```
  ./hw07 -haar1_for dir < input.ppm
  ```

  where *dir* is either "*h*" or "*v*" for horizontal or vertical, the input image is called input.ppm and the output is written to stdout. For example, to compute the forward horizontal Haar of msg1.ppm, you could run:

  ```
  ./hw07 -haar1_for h < msg1.pgm
  ```

- To compute horizontal and vertical inverse Haar transforms on the coefficients and output the reconstructions on the stdout:

  ```
  ./hw07 -haar1_inv dir < input.ppm
  ```

  where *dir* is either "*h*" or "*v*" for horizontal or vertical, the input image is called input.ppm and the output is written to stdout. For example, to compute the inverse vertical Haar of msg1.ppm, you could run:

  ```
  ./hw07 -haar1_inv v < msg1.pgm
  ```

- To compute the forward 2D Haar transform and output the result to the stdout:

  ```
   ./hw07 -haar2_for < input.ppm
  ```

- To compute the inverse 2D Haar transform and output the result to the stdout:

  ```
  ./hw07 -haar2_inv < input.ppm
  ```

- To compress an input image using 2D Haar Wavelets and output the SNR and the reconstructed image on stdout:

  ```
  ./hw07 -haar_compress t < input.ppm
  ```

  where *t* is the threshold value, the input image is called input.ppm and the output is written to stdout where the first line contains the SNR and the rest contain the output PPM file. For example, to compress the image sena.ppm with threshold *threshold* = 0.5 you could run:

  ```
  ./hw07 -haar_compress 0.5 < sena.pgm
  ```

## Instructions

- All code should be implemented in C++ under Linux.
- Please submit your homework in one zip file named as follows: *CMPN206.HW##.FirstName.LastName.zip*, so for example if your name is Mohamed Aly and this is homework #1, then the file name should be *CMPN206.HW01.Mohamed.Aly.zip*.
- Please include all your code and sample output in the zip file, with a README file to explain what you did. Failure to follow these instructions will cause deductions from your grade.
- You are allowed to discuss the problems among yourselves. However, **copying** any part of the code will result a grade of **ZERO**. No exceptions.

## Grading

- 9 points: requirements above
- 1 point: submission instructions