SF2: Image Processing Final Report

Nicholas Capel (nrjc2)

June 2, 2016

1 Introduction

In Interim reports 1 and 2, four distinct image compression schemes were analysed. The Laplacian Pyramid, the Discrete Cosine Transform, the Lapped Bi-orthogonal Transform, and the Discrete Wavelet Transform. Of the four schemes, the LBT and DWT were deemed worthy of further investigation.

In this report, a new measure of image quality, the SSIM, is introduced. The DWT and LBT are also examined in greater detail. In addition, the quantisation and coding schemes were both examined, and improvements made.

2 Measures of Visual Appearance

In previous reports, I have used the MSE as a measure of similarity between two images. However, that might not necessary be the best measure of image quality. Ultimately, the aim of these compression schemes is to create a compressed image that appears the best to the human eye. Hence, a good metric must be able to identify the statistics of the image that are important to human perception.

In this report, 2 separate schemes for measuring the quality of images were examined:

1. Mean Square Error

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$$

2. SSIM (Structural Similarity Index)

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

The MSE scheme is a rather intuitive method to measure the error between two images. It is easy to state, and has a clear physical meaning. However, the MSE scheme has clear limitations.

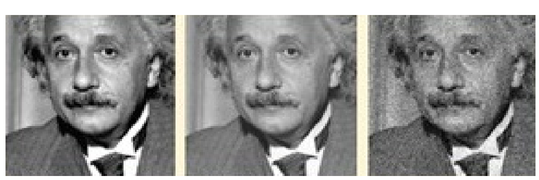


Figure 1: Three Images with equal MSE

In these three images, the MSE is exactly the same. However, from a viewer's perspective, the image quality is vastly different. More specifically, MSE performs poorly when rotations, DC-level variations, scaling, and shifting are involved.

As compared to the MSE, the SSIM is a superior measure of visual appearance, because it can take the similarity of structural features into account. Hence, in the rest of this report, the SSIM will be used in conjunction with the MSE for the purposes of measuring the visual quality of images.

3 Investigations into the DWT

In Interim Report 2, the Discrete Wavelet Transformation scheme was described, and was found to have relatively high Compression Ratios. Furthermore, at equivalent RMS error levels, the artifacts are much less noticeable than any of the other schemes investigated.

It was observed that equal MSE schemes perform better than constant quantisation schemes. Furthermore, as the layer depth increased, the compression ratio increased. Hence, for all experiments performed in this report, a deep wavelet tree is used, with quantisation performed using the equal MSE scheme.

When compressing the Legall 5/3 scheme to 40,960 bits per image, significant artifacts were introduced (Figure 8c).

It was theorised that other wavelets exist that give better images at high compression. Hence, a series of experiments were conducted to determine how the choice of wavelets can affect compression when compressing down to a size of 40,960 bits per image.

Table 1: SSIM and MSE for wavelets

Wavelet	Level	SSIM	MSE	Figure
Legall 5-3	4	0.4926	17.0794	8c
Inverse Legall 5-3	4	0.038	58.7973	8d
CDF 9-7	6	0.4822	16.7329	8e
Haar	8	-0.0038	69.3062	8f
DB 8	6	0.0016	69.7929	8g
JPEG (reference)	-	0.6704	12.8703	8b

Although the Wavelet-compressed images did not possess the blocky artifacts that are inherent in the DCT encoding of the JPEG format, they do not show significant improvements in either SSIM or in MSE as compared to the the JPEG. Moreover, wavelet transforms introduce their own artifacts into the image, and the visual quality of the image tends to vary vastly within the image.

This effect is best illustrated in Figure 9-B. At the bottom-left corner of the image, the fences and the patterns on the wall of the house are reproduced to a high degree of accuracy. However, looking at the sky and the lighthouse itself, they are reproduced in extremely poor detail. Human observers will look at the image and find the vast differences in image quality between different parts of the same image to be highly distracting.

In the DCT, each block is considered separately due to the nature of the algorithm, hence, when quantisation occurs, the errors introduced are localised to specific blocks. However, in wavelets, the quantisation errors seem to be introduced in an unpredictable manner, leading to the phenomena described above.

An attempt was made to correct this by splitting the DWT into different, 8×8 tiles and performing DWT on each tile separately. A DWT scheme was implemented as shown in the Figure below:

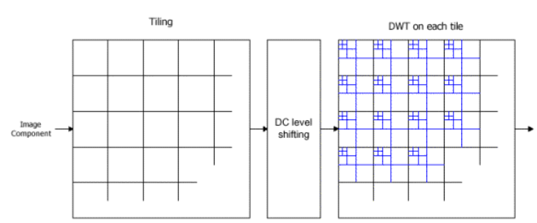


Figure 2: Tiled DWT

The resulting tile-DWT-compressed image is shown in Figure 9-C. As can be observed, the image quality is slightly more uniform. However, the Compression Ratio has been decreased, as each individual tile can only be split into a DWT pyramid size of 2. Furthermore, ugly blocking artifacts have once again been introduced due to the discontinuities between each separate block.

It was decided to abandon the DWT as an encoding scheme, as it simply does not perform well at high compression ratios, and any attempt to increase the image quality has failed horribly. It is probably the case that at small Mean Square Errors, the Wavelet compression performs much better than the LBT in terms of image artifacts. However, past a certain point, the higher compression ratio of the LBT wins out.

Table 2: SSIM and MSE for wavelets

Wavelet	Level	MSE	SSIM	Bits	Figure
Legall 5-3	4	16.0027	0.3505	41033	9-B
Legall 5-3 with Tiling	2	12.4759	0.4243	113120	9-C

4 Investigations into the LBT

An experiment was conducted to augment the JPEG encoding format with the POT to implement a LBT. The LBT selected had a block size of N=8, and a scaling factor of $\sqrt{2}$.

It was observed that,when compressing down to 40,960 bits, the resulting LBT transform (Figure 10a) had a lower MSE and higher SSIM as compared to the JPEG (Figure 8b). The most pronounced benefit of using the LBT was a removal of the blocking artifacts that are inherent in the JPEG format.

Table 3: SSIM and MSE for LBT

Scheme	MSE	SSIM
JPEG	12.8703	0.6704
JPEG + LBT	12.1444	0.682

Hence, the simplest and most basic implementation of the LBT outperformed the JPEG format, which is something that none of the complex wavelet transforms were able to achieve. Thus, the decision was made to explore the LBT in greater detail.

In previous interim reports, the overlap factor was always assumed to be half of the block size. It was theorised that a trade-off exists between blocking artifacts and ringing artifacts. When no overlap is implemented (the naive DCT), there exist blocking artifacts about the edge of DCT blocks. When the default overlap factor of 4 was introduced using the standard POT, this led to a decrease in the blocking artifacts, but an increase in ringing artifacts around sharp discontinuities in the image. Thus, it is hypothesized that a "sweet-spot" exists which provides a good trade-off between ringing and blocking artifacts.

(a) O=2 (b) O=4 (default)

Figure 3: Comparison between overlap sizes

However, further investigation revealed that no such sweet-spot exists (Figure 3). Any attempt to decrease the overlap reintroduces the blocking artifacts, while not significantly reducing the ringing artifacts. As can be seen in Table 4, the MSE and SSIM are both adversely affected by decreasing the overlap factor. Hence, in the final competition image, the default overlap factor was used.

Table 4: SSIM and MSE for Different overlap sizes

Scheme	MSE	SSIM
O=2	13.1663	0.65661
O=4 (default)	12.9005	0.67162

5 Quantisation Investigations

5.1 Centre-Clipped Quantisers

5.1.1 DWT

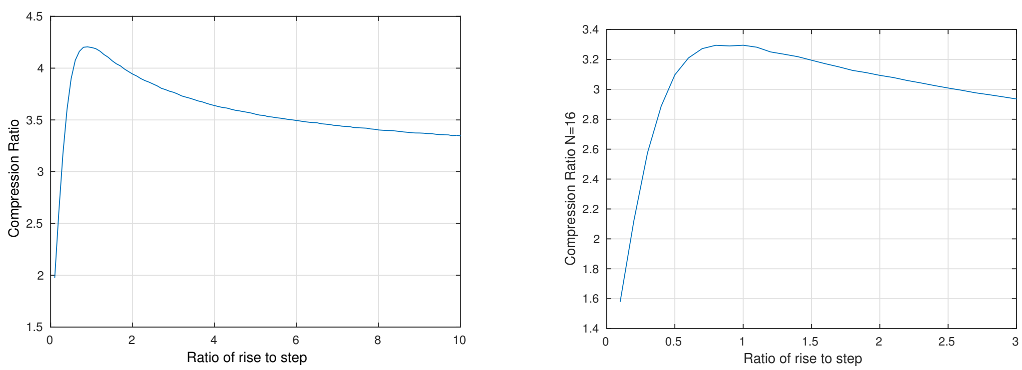
DWT was conducted on the lighthouse image at a pyramid level of 6, using the 5/3 Legall Wavelet. The RMS error was enforced to be equal to a direct quantisation at step size 17, and the value of the rise size was varied.

The value of Compression Ratio is maximized when the rise size is 0.9 times the step size (Figure 4a).

5.1.2 LBT

LBT was conducted on the lighthouse image, using a POT coefficient of $\sqrt{2}$ and a block size of 8. Once again, the RMS Error was enforced to be equal to a direct quantisation at step size 17, and the value of the rise size was then varied.

The Compression Ratio was measured at a block size of N=16. The value of the Compression Ratio is maximized when the rise size is exactly equal to the step size (Figure 4b).



(a) DWT (b) LBT

Figure 4: Compression Ratio vs Step Size Ratio

As the rise size increases, the compression ratio initially increases, as we obtain savings from quantising coefficients to zero. However, once the rise size increases past a certain point, the compression ratio decreases, as more and more of the image is being set to zero. Overall, it was found that a double-width centre step quantiser is an excellent compromise, and works well with all the front-end systems tested.

5.2 Suppression of Coefficients

In order to investigate the effect of suppressing coefficients, an experiment was conducted to measure the energy of any of the individual sub-images. All the subimage coefficients were then zeroed if the sub-image energy fell below a certain percentage of the total image energy. The MSE was then optimized to be the same as a direct quantisation at step 17, and the compression ratio was measured. A figure of the results can be seen below:

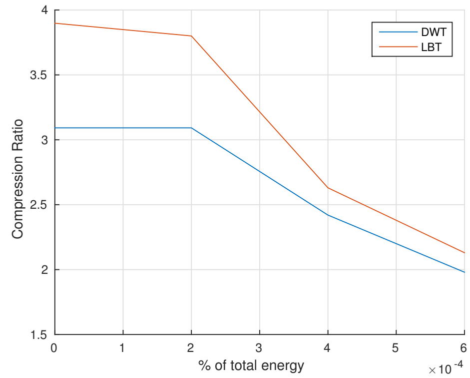


Figure 5: Compression Ratio vs Threshold energy level

As can be observed, the Compression Ratio falls as the threshold energy increases.

However, it might be the case that the calculation of the compression ratio underestimates the suppression of coefficients. The first order entropy calculations do not take savings from runs of zeros in the run-length encoded scheme into account. When the entire sub-image is zeroed, the length of the runs of zeros might increase as compared to the base case, adding extra bit savings.

However, at high compression, the variable quantisation step size combined with the zero-rise width of the quantiser would have caused the coefficients of the high-pass sub-images to be suppressed anyway. Hence, the savings introduced are unlikely to be large, and no further investigations were conducted into the suppressing of coefficients.

5.3 Using the IJG Standard Quantisation Table

The JPEG Matlab code implements a uniform quantisation scheme, with each coefficient quantised to the same level. However, this is sub-optimal, as the human eye places a very different weightage on coefficients corresponding to different frequencies. As a result, a non-uniform quantisation scheme was implemented based on the IJG Standard Table:

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

This quantisation table was not derived from the equal MSE scheme, as was the case with all quantisation tables in previous reports, but by psycho-physical experiments that examine the subjective quality of images as perceived by humans. Hence, this quantisation matrix was chosen because it preferentially quantises the frequencies that humans deem as important.

Table 5: SSIM and MSE for constant and IJG quantisation

Scheme	MSE	SSIM
JPEG + LBT	12.1444	0.682
JPEG + LBT + IJG quantisation	13.1088	0.6551

Although the IJG Standard Quantisation leads to a worse MSE and SSIM as compared to the baseline scheme, from a subjective POV, it is noted that the IJG quantisation scheme suppresses the LBT artifacts (Figure 10a), and looks more pleasing to the human eye as compared to the default LBT (Figure 10b). This is likely because the IJG Standard Table is specifically designed to give artifacts at the threshold of visibility. Thus, it was selected as a superior quantisation scheme.

6 Coding Investigations

6.1 Discussion on choice of Huffman coding

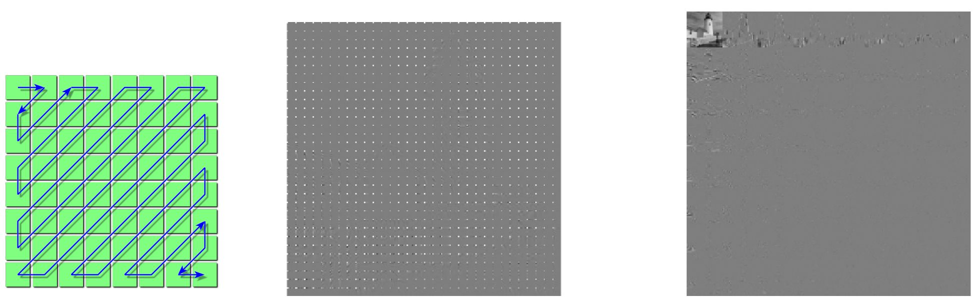
Previous reports were mainly concerned with reducing the entropy of natural images using clever decomposition schemes. However, after an image has been quantised, it is necessary to convert the quantised image into a string of bits that can then be stored.

In this report, the Huffman code is used for compression. It is chosen mainly because it is simple to use. However, Huffman codes have a weakness. When the probability of a single event is greater than 0.5, Huffman codes are only able to allocate a minimum of one bit to this event, and hence, the Huffman code becomes much less ideal when the probability of a single event, $p_i > 0.5$.

Hence, in order to improve the naive Huffman code, we use run length encoding. In Image Compression, the event with the greatest probability is a pixel having an intensity of zero. This is because on quantisation at high step sizes, small coefficients in the high frequencies will be quantised to a value of zero. Run length encoding essentially codes a run of zeroes as a single event in the Huffman code. Hence, in the new Huffman table, the probability of any single event will now be less than half, and the efficiency of the code will be restored.

However, in order to exploit these runs of zeroes, the order of sampling (or scanning) of the coefficients becomes important. In the JPEG format implemented in the MATLAB code provided, when encoding the coefficients of the DCT, the image is scanned in a zigzag fashion, with the top-left coefficient of a block being encoded as a DC coefficient. A diagram is shown in Figure 6a.

This is a reasonable scanning strategy because it exploits the structure of the DCT coefficients Take the lighthouse image, for example:



(a) Zigzag encoding sequence

(b) Lighthouse post-DCT

(c) Lighthouse post-DCT rearranged

Figure 6: Huffman coding illustrations

At high quantisations step sizes, the high-pass coefficients will all be suppressed. Hence, when encoding the image using the zig-zag sequence, there will be extremely long runs of zeros along the diagonals that correspond to the high-pass coefficients. This can be exploited by the run-length-encoding scheme to increase the compression.

Although the augmented Huffman code works well, it is not the ideal solution. The Huffman code works well only when the probability histogram of the data being coded has probabilities close to 2^{-k}, where k is an integer. A possible workaround is to group existing symbols together and enlarge the alphabet. However, as the Huffman code tree enlarges, it quickly becomes extremely difficult to construct, and hence, there is a computationally-bound upper limit on maximum size of the Huffman coding tables.

However, with an arithmetic code, an arbitrarily long code can be constructed trivially, which leads to compression that better approaches the theoretical minimum bound. However, due to time constraints, arithmetic coding was outside the scope of this investigation.

6.2 Investigating alternative scanning strategies

On closer examination, the diagonal scanning method is also sub-optimal. This is because in our new scheme, the zero rise size is set to be exactly equal to the quantisation size. Hence, the ordering of the scanning should start at

Table 6: SSIM and MSE for diagonal vs optimized scanning(LBT with IJG quantisation)

Scheme	MSE	SSIM
Diagonal	13.1088	0.6551
Optimized	13.3240	0.6425

the smallest value of the IJG standard table and proceed to the largest. A modification was made to the scanning strategy, and the images were once again compressed to the same value of 40,960 bits.

It is seen that the "optimized" scanning method based on the IJG Standard Table (Figure 10c) does not improve the image compression performance as compared to the default diagonal scanning scheme (Figure 10b). This might be due to the fact that the IJG tables represent quantisation step sizes that are tuned to work on the average image. The bridge image used in this case, on the other hand, might have significantly different statistics, such that scanning based on the IJG table is no longer optimal.

However, a decision was made to abandon scanning methods as an area of exploration, as preliminary results suggest that additional gains in this area are not likely to lead to significant improvements in compression.

6.3 Improving the coding of DC coefficients

Another possible area of improvement in the default JPEG encoding is in the encoding of the DC components. Firstly, in the JPEG encoding scheme provided, the number of DC bits is fixed to a constant value for all sub-images. However, this is a waste of bits. Small DC coefficients do not require as many bits to encode as large DC coefficients. Utilizing the formula:

$$bits = \log_2 |x| + 2$$

where x is the DC coefficient, we may dynamically allocate DC bit sizes to each of the DC values, and decrease the total number of bits used to encode the image. When this scheme was implemented on the Bridge Image, significant improvements were made in both SSIM and MSE (Table 7).

Furthermore, the DC component in each block is large and varies across the blocks, but is often close to that of the adjacent blocks. Hence, it is possible to use a Differential Modulation scheme to encode the difference between adjacent DC values instead of the absolute value. Theoretically, this should be able to decrease the number of bits used to encode an image that has the same quantisation step size, as we can use less DC bits. The results of an experiment on the bridge image are shown below. Note that adding differential modulation to the DC coefficients further improves both the MSE and the SSIM (Table 7, Figure 10e).

Table 7: SSIM and MSE for dynamic bit-size allocation

Scheme	MSE	SSIM
Base Scheme (JPEG + LBT + IJG)	13.1088	0.6551
Dynamic Sizing	12.8089	0.6716
Dynamic Sizing + Differential Modulation	12.7022	0.6769

6.4 Investigating the benefits of default vs customised Huffman tables

Originally, it was theorised that, since all natural images share the same properties and the proposed image transform is fundamentally similar to the default JPEG encoding, it might be possible to save 1424 bits by using the default JPEG Huffman tables.

An experiment was conducted to test this hypothesis. The best possible scheme was used, and the bridge image compressed down to 40,960 bits with and without the custom Huffman table.

It was found that using a custom Huffman table improved compression performance significantly. It is hypothesized that the default JPEG table does not apply well to the new scheme, as the Photo Overlap Transform modifies the pre-DCT image such that the natural statistics are significantly different from the original natural image. Hence, a custom Huffman table needs to be used, as the default Huffman table values are no longer as relevant.

Table 8: SSIM and MSE with and without custom Huffman table

Scheme	MSE	SSIM
Without Custom Huffman	12.7022	0.6769
With Custom Huffman	12.6021	0.7304

7 Final Scheme

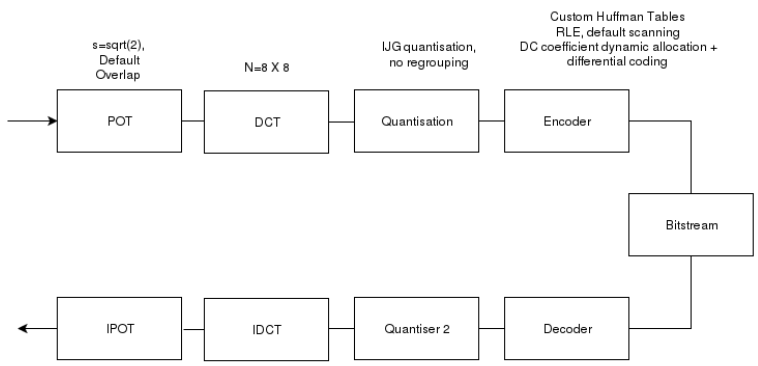


Figure 7: Final Competition Encoding and Decoding Flow

The above figure describes the final encoding and decoding sequence used in the competition. The final compression scheme implemented was a combination of the best schemes that were discovered in the process of investigation. These include:

1. Using the LBT with **default overlap size** =4, **s** = $\sqrt{2}$, and **block size** of 8×8 **.**
2. Utilising **IJG default tables** to perform quantisation
3. Utilising **custom Huffman tables** with **RLE.**
4. Using **differential coding** of DC coefficients, with **dynamic allocation** of bit sizes.

I was in a group with Lewis Jones, and he focused on doing experiments on the Naive LBT scheme, attempting to change the values of the regrouping coefficient, M, as well as the size of the overlap in the POT to try to improve the results. I spent a significant amount of time examining the DWT schemes, as well as experimenting with various coding strategies.

Overall, we achieved third place in the competition, which is a rather good result. The competition images can be found in Figure 11, and the final comparison metrics are in Table 9.

8 Conclusion

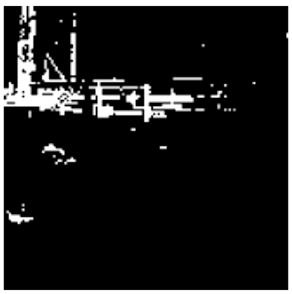
In conclusion, over the course of three reports, four different schemes were examined (DWT, DCT, LBT, Laplacian). The decision was made to focus on two of those schemes: the DWT and the LBT. In the final report, further investigations were performed into the LBT, the DWT, quantisation strategies, and coding strategies. The best schemes were them combined into the final scheme that was implemented in the competition.

A Appendix

(a) Original Image

(b) JPEG encoded

(c) Legall 5/3 Wavelets



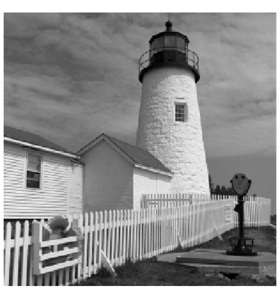
(d) Inverse Legall 5/3 Wavelets

(e) CDF 9-7 Wavelets

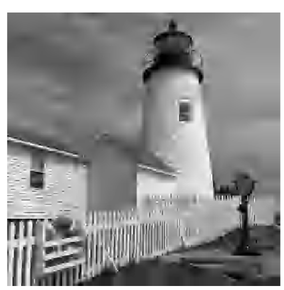
(f) Haar Wavelets

(g) DB8 Wavelets

Figure 8: Bridge Compressed With Wavelets



(a) Original Image



(b) Legall 5/3 Wavelet

(c) Legall 5/3 Wavelet with Tiling

Figure 9: Lighthouse Compressed with Wavelets

(a) LBT direct implementation

(b) LBT + IJG quantisation

(c) LBT + IJG quantisation + IJG scanning

(d) LBT + IJG quantisation + dynamic bit allocation

(e) LBT + IJG quantisation + dynamic bit allocation + Differential coding

(f) LBT + Custom Huffman Table

Figure 10: LBT and Quantisation investigation images

(a) Uncompressed Images

(b) Compressed Images

Figure 11: Competition Images

Table 9: SSIM and MSE of compressed images

Image	MSE	SSIM
Parrot	5.6331	0.7278
Bridge	12.5054	0.6960
Flamingo	12.0971	0.7413