**8. Conclusions**

The aim of this work has been to explore the area of still image compression, improve existing techniques and to develop new image compressors. In order to achieve this the basic techniques used in still image compression were highlighted in Chapter 2 and discussed in relation to image compression. Low complexity methods of image compression were examined in Chapter 3 and the parameters used were optimised. Although the methods employed in this chapter did not compare well to JPEG in rate distortion terms, important improvements were made to their design which were carried forward to Chapter 4. Chapter 4 expands on the low complexity methods and investigates the effect of partitioning the image. The result of Chapter 4 was an image compressor that competed with JPEG at higher compressions. In Chapter 5 the topic of optimal quantisation was introduced and its application to still image compression was discussed. In Chapter 6 more complex DCT methods were investigated. The operation of the JPEG standard was briefly explained and its performance was demonstrated. Two other new complex DCT coders were introduced in Chapter 6. The variable coefficient DCT expanded on ideas from Chapter 4 and used the JPEG compression framework. The optimal quantisation DCT used the method described in Chapter 5 with linear quantisation. Both of these methods performed better than JPEG. In Chapter 7 wavelets were examined, and the Shapiro EZW was implemented as a basis for comparison with the other methods. The optimal quantisation method was also applied to wavelets and was found to be better than JPEG but not as efficient as Shapiro.

Chapter 2 discussed the three basic processes in image compression and summarised their important points. Transforms are usually orthogonal and this has been very important throughout this work since they are easier to handle in this form and produce better results. The quantisation can be a simple process and the use of equation (2.24) was a major simplifying factor through this work. Until the error performance under quantisation was examined it was not clear that this simple connection between block size and quantisation could be made. Most methods in the work have used Huffman lossless compression but it is important to remember that Huffman compression is only effective for entropies above 1.5-2 bits/symbol and it is impossible to code symbols whose entropies are less than 1bits/symbol with Huffman compression. In the cases where this has been true adaptive arithmetic coding has been used to replace the Huffman.

Chapter 3 investigated low complexity methods and their enhancement. Both the orthogonalised polynomial and the truncated DCT were contenders for the best basis approximation method. Before they could be compared their parameters were optimised. It was found that the order 6 polynomial (figure 3-8) using quantisation of Q=4 (figure 3-5) were the best parameters for the polynomial. The truncated DCT was also optimal for order 6 (figure 3-13) and Q=4.0 (figure 3-11). This is not a very surprising result but it is important to show it. The truncated DCT was shown to be better than the polynomial in figure 3-15 and this allowed all further work to employ the DCT in favour of the polynomial. The results of applying a fractal enhancement were also investigated in Chapter 3. It was shown that increasing the level of fractal enhancement progressively improved the rate distortion performance of the DCT basis approximation. The addition of a simple orthogonalised centred block fractal produced a small improvement (figure 3-17) and by searching for the best parent block a further improvement was achieved (figure 3-19). The method of limited fractal searching was much more effective, and the gains in rate distortion performance were much more pronounced (figure 3-22). Although increasing fractal complexity did improve the rate distortion performance of the system there was a speed penalty, that increased with fractal complexity. The final enhancement assessed was Vector Quantised (VQ) error clean-up. The method showed improved results compared to the limited fractal method (figure 3-23) at low compression but tended towards the performance of the plain DCT basis approximation at higher compressions.

Chapter 4 investigated using the truncated order 6 DCT in hierarchical structures. The existing methods were introduced and the Quad-tree and Horizontal-Vertical Decomposition (HVD) structures were chosen for closer inspection. The block splitting criterion was initially a fixed threshold. The threshold method was implemented with the quad-tree structure (figure 4-6) and was less effective than the fixed block size method from Chapter 3. A new method was developed to produce an optimal splitting of the image, called the error sorting method. This splits the block with the largest square error and continues to split blocks until the compressor’s bit budget is exhausted. This was shown to have better performance than either the fixed block size method or the threshold method (figure 4-13). Because the DCT can directly calculate the squared error from its coefficients, blocks that are encoded do not have to be rendered (making the method very fast). The quantisation for the error sorting quad-tree DCT method was optimised to ensure that it was not different to the fixed block size method and Q=4 was found to the optimal again (figure 4-11). The HVD was also implemented using an order 6 DCT and error sorting but was shown not to be as effective as the quad-tree on complex images such as Goldhill (figure 4-15). The HVD is effective on images with regions of low detail such as mug shots (figure 4-17) where it out performs the quad-tree. Finally in chapter 4 the effect of applying fractals to the quad-tree was investigated. It was found that fractals do not improve the quad-tree order 6 DCT since the rate distortion improvement is not high enough. It was also noted that although the quad tree order 6 DCT did not perform as well as JPEG it did not suffer from ‘JPEG like’ blocking artefact since it localised the effects of the DCT and only used large blocks where they were appropriate.

Chapter 5 explained the method of optimal quantisation/source coding. It started by explaining the general problem faced by all image compressors and then proposed a solution, with optimal quantisation. The equal gradient method used in optimal quantisation is well known. This chapter clearly explained the principle of its operation and dealt with the practical problems that were faced when implementing the method with quantisation. The chapter also discussed the problem caused by discrete rate distortion data points and introduces a filter to ensures that the data fits the necessary model. The use of templates was discussed and this allows the optimal quantisation method to be used in systems where time constraints make it impossible to fully construct the rate distortion characteristics of the image.

Chapter 6 investigated three variations of the full Discrete Cosine Transform (DCT). The JPEG standard was briefly explained as an introduction to the DCT methods and a rate distortion characteristic was generated for Goldhill (figure 6-4). The variable coefficient DCT was then introduced which extended the ideas in chapter 4 and allowed the compressor to decide the complexity of the DCT transform for itself. Both the numbers of DCT coefficients and the sizes of the blocks were allowed to vary supported by a quad-tree structure. The coefficients were still compressed using a JPEG encoding scheme but the quantisation was varied to produce the best results (figure 6-8). The optimised variable coefficient DCT performed better than the existing JPEG (figure 6-9). The optimally quantised DCT used the ideas developed in chapter 5 to linearly quantise the coefficients of a DCT with a fixed block size across the image. The best block size was chosen to be 16x16 (figure 6-13) and the method performed much better than JPEG baseline (figure 6-15). Finally the limited coefficient quad-tree DCT (LQT-DCT) and the three methods discussed in this chapter were compared in figure 6-18 which showed their relative performance. The optimally quantised DCT was best, followed by the variable coefficient DCT, JPEG and the LQT-DCT. It was also shown that the LQT-DCT was more efficient than JPEG at the higher compressions.

Chapter 7 investigated the wavelet transform. The ‘unofficial standard’ of wavelet compression, the Shapiro EZW, was introduced and implemented to give a comparison to the other compression methods (figure 7-7). It also highlighted the fact that although wavelets do not have blocking artefact they do suffer from a different type of ‘shot noise’ artefact (figure 7-9). The optimal quantisation method was then applied to the wavelet transform method at different scales. The best number of scales to apply the wavelet transform to was found to be 4 (figure 7-10). It was found that although this method compared favourably with JPEG it was not as efficient as the EZW. This was accounted for by the fact that wavelets have highly correlated sets of coefficients which the EZW exploits but the optimal quantisation method only works when the coefficients are unrelated.

Figure 8-1 shows a comparison between all the methods that have shown promise in this work. There are several important points to notice in this comparison:

* The best rate distortion performance is achieved by the optimally quantised DCT, with the EZW being slightly worse, in rate distortion terms.
* The worst rate distortion performance occurs with the low complexity order 6 DCT but this is also the fastest method.
* The limited searching and fixed block size fractal methods have roughly the same performance as the quad tree order 6 DCT. Unfortunately the fractal requires much more processing time that the quad-tree to achieve this result.
* All the high complexity methods are better than JPEG (which is a high complexity method) and all the low complexity methods are less effective than JPEG but only at low compressions. At high compressions all the methods are better than JPEG.



Figure 8-1. Comparison of rate distortion characteristics of the image compressors from this work.

In general these comparisons show that high complexity methods are the only way to obtain competitive rate distortion performances and as a result there will always be a speed/performance balance in image compression, which is particularly relevant in selecting codecs for use in video compression.

This work has looked at most types of still image compression. Its major contributions have been optimising the parameters of low-complexity transforms, applying these results to hierarchical structures and using optimal source coding techniques on high complexity transforms. This work shows that increasing computational complexity yields better rate distortion performance, and hence a balance has to be reached between the speed of a system and its performance. This work also shows that the transform stage of a compressor is not as important as the compression methods used on the transformed data. The wavelet methods do not perform any better than the DCTs in terms of MSE when the compression is done correctly.

There are some areas in this work that could be continued:

* Application of the vector quantisation error correction or limited searching fractal methods to quad-trees.
* Application of the optimal quantisation methods to hierarchical truncated DCT.
* Use of the rate distortion switch for self extending the truncated quad-tree DCT.
* Application of fractal transform on a different quad-tree to the LQT-DCT, in order to make fractals work better with quad-trees. This is difficult, since the method of orthogonalisation is undefined.
* Extending the template work (chapter 5) to test in a video system.