Direction-Adaptive Karhunen-Loève Transform for Image Compression

Vinay Raj Hampapur, Wendy Ni

Abstract—Amongst transform coding techniques for image compression, the Karhunen-Loève Transform (KLT) uniquely achieves maximum coding gain by decorrelating the input image signals. However, KLT’s performance relies heavily on the accuracy of the estimated input signal distribution. On the other hand, directional transforms such as the Directional Discrete Transform (DDCT) exploit the similarity in image blocks with directional content, though the transform does not adapt to varying image content. In our studies, we designed and implemented an adaptive KLT transform dubbed Direction-Adaptive Karhunen-Loève Transform (DA-KLT), a novel transform coding technique for lossy image compression. The DA-KLT combined the optimality of the KLT with direction-adaptive training to provide a more accurate estimate of the image signal statistics. Our method achieved a consistent improvement over KLT, but fell short of the DDCT in performance. Nevertheless, the DA-KLT delved into a new direction in image compression research by combining two approaches to further exploit statistical dependencies in images.

Index Terms—DA-KLT, DDCT, KLT, block classification, image compression

I. INTRODUCTION

T RANSFORM coding is a class of lossy image compression techniques that exploits the statistical dependencies between image pixels by converting them to coefficients with little or no dependencies [1]. Image transforms are generally carried out as block-wise decomposition of image pixels using a basis that is often pre-defined and independent of image content. Transforms with fixed bases include Discrete Cosine Transform (DCT) and various Discrete Wavelet Transforms (DWTs). In comparison, the Karhunen-Loève Transform (KLT) improves compression by customising its basis vectors to the image content. For a random vector $X$ representing the set of pixel values in image blocks, the matrix of basis vectors $\Phi$ is defined to be the matrix of eigenvectors of the covariance matrix $C_X$ [1]:

$$C_X = E [(X - E[X])(X - E[X])^T] = \Phi C_Y \Phi^T$$

(1)

Because $C_Y$, the covariance matrix of the output transform coefficients, is a diagonal matrix, the KLT has completely decorrelated its input $X$. As this is a necessary condition for full exploitation of input signal dependencies, the KLT tends to perform very well in general [1]. Furthermore, eigendecomposition using the single value decomposition (SVD) ensures that the matrix $\Phi$ is orthonormal. For a Gaussian input, the output is then Gaussian and independent. Hence, for an image with Gaussian statistics, KLT fully exploits statistical dependencies between pixels and is the optimal transform [1]. However, in order to calculate the covariance matrix $C_X$, it is essential that the image statistics is known a priori. This limits the use of KLT as such information is not usually available.

Transforms may be improved in another manner by taking into account the directionality of the block content. For example, the Directional Discrete Cosine Transform (DDCT) operates in multiple modes, where each block can be vectorized differently for use with pre-defined basis vectors customised for various directions (Fig. 1 and 2) [2]. Then, one set of basis vectors would align most closely with the block content, improving energy compaction and hence compression ratio [2].

In this study, we combined KLT and direction adaptation to design and implement a novel transform coding technique, Direction-Adaptive KLT (DA-KLT). Instead of pre-defining the matrix $\Phi$ is orthonormal. For a Gaussian input, the output is then Gaussian and independent. Hence, for an image with Gaussian statistics, KLT fully exploits statistical dependencies between pixels and is the optimal transform [1]. However, in order to calculate the covariance matrix $C_X$, it is essential that the image statistics is known a priori. This limits the use of KLT as such information is not usually available.

Transforms may be improved in another manner by taking into account the directionality of the block content. For example, the Directional Discrete Cosine Transform (DDCT) operates in multiple modes, where each block can be vectorized differently for use with pre-defined basis vectors customised for various directions (Fig. 1 and 2) [2]. Then, one set of basis vectors would align most closely with the block content, improving energy compaction and hence compression ratio [2].

In this study, we combined KLT and direction adaptation to design and implement a novel transform coding technique, Direction-Adaptive KLT (DA-KLT). Instead of pre-defining the matrix $\Phi$ is orthonormal. For a Gaussian input, the output is then Gaussian and independent. Hence, for an image with Gaussian statistics, KLT fully exploits statistical dependencies between pixels and is the optimal transform [1]. However, in order to calculate the covariance matrix $C_X$, it is essential that the image statistics is known a priori. This limits the use of KLT as such information is not usually available.

Transforms may be improved in another manner by taking into account the directionality of the block content. For example, the Directional Discrete Cosine Transform (DDCT) operates in multiple modes, where each block can be vectorized differently for use with pre-defined basis vectors customised for various directions (Fig. 1 and 2) [2]. Then, one set of basis vectors would align most closely with the block content, improving energy compaction and hence compression ratio [2].

In this study, we combined KLT and direction adaptation to design and implement a novel transform coding technique, Direction-Adaptive KLT (DA-KLT). Instead of pre-defining the matrix $\Phi$ is orthonormal. For a Gaussian input, the output is then Gaussian and independent. Hence, for an image with Gaussian statistics, KLT fully exploits statistical dependencies between pixels and is the optimal transform [1]. However, in order to calculate the covariance matrix $C_X$, it is essential that the image statistics is known a priori. This limits the use of KLT as such information is not usually available.

Transforms may be improved in another manner by taking into account the directionality of the block content. For example, the Directional Discrete Cosine Transform (DDCT) operates in multiple modes, where each block can be vectorized differently for use with pre-defined basis vectors customised for various directions (Fig. 1 and 2) [2]. Then, one set of basis vectors would align most closely with the block content, improving energy compaction and hence compression ratio [2].

In this study, we combined KLT and direction adaptation to design and implement a novel transform coding technique, Direction-Adaptive KLT (DA-KLT). Instead of pre-defining the matrix $\Phi$ is orthonormal. For a Gaussian input, the output is then Gaussian and independent. Hence, for an image with Gaussian statistics, KLT fully exploits statistical dependencies between pixels and is the optimal transform [1]. However, in order to calculate the covariance matrix $C_X$, it is essential that the image statistics is known a priori. This limits the use of KLT as such information is not usually available.

Transforms may be improved in another manner by taking into account the directionality of the block content. For example, the Directional Discrete Cosine Transform (DDCT) operates in multiple modes, where each block can be vectorized differently for use with pre-defined basis vectors customised for various directions (Fig. 1 and 2) [2]. Then, one set of basis vectors would align most closely with the block content, improving energy compaction and hence compression ratio [2].

In this study, we combined KLT and direction adaptation to design and implement a novel transform coding technique, Direction-Adaptive KLT (DA-KLT). Instead of pre-defining
basis vectors, the DA-KLT used training images to optimise the basis vectors. Training images were first partitioned into equal-sized square blocks, which were classified into depending on the presence and direction of dominant edges within the blocks. The classification algorithm we implemented utilised Canny’s edge detection [3] and pixel variance. Block statistics for each class were then used to calculate multiple sets of basis vectors to be used for image compression. To compress a test image, the same partition and classification procedures were carried out. Then, KLT was performed on each block using appropriate basis vectors. The coefficients were finally quantised and encoded using a Huffman coder.

The DA-KLT was implemented in MATLAB and tested on 512 × 512 8-bit grayscale images. The results were benchmarked against conventional KLT and DDCT. We also investigated the effect of varying quantisation step and block size, and implementing the additional features of principal component truncation and DC separation. The outcome we obtained shows that DA-KLT is a feasible transformation coding technique. Hence our work signalled a new direction of exploration in the improvement of transform coding techniques.

In Section II of this report, we first describe the functional and programmatic components of DA-KLT from training to the compression stage. Then, in Section III, we present and discuss the trends, implications and performance assessment of some results. Section IV concludes this report by reviewing the benefits and significance of DA-KLT.

II. DA-KLT

A. Training

To overcome the lack of a priori knowledge of the input image statistics, we processed a set of training images to approximate the image statistics and derive basis vectors for the transform. In order to exploit directionality, we first partitioned the images into blocks of size \( N \) where \( N \in \{4, 8, 16, 32\} \). These blocks were then classified into one of 10 classes using an algorithm utilising Canny’s Edge Detection Algorithm [3]. In our algorithm, we first calculated the gradients \( \delta x \) and \( \delta y \) of the block in the horizontal and vertical directions respectively. From these gradients, the angle and magnitude for each pixel were calculated. Then angle \( \theta = \tan^{-1} \left( \frac{\delta y}{\delta x} \right) \) was quantized into 8 different directions: 0°, ±22.5°, ±45°, ±67.5°, and 90°. In order to determine the dominant direction, we summed the magnitudes of the gradient vectors in each direction. If the maximum sum exceeded threshold \( X_{th} \), we defined the block as directional and assigned a class ID in the range 1, 2, · · · 8 corresponding to one of the directions. The choice of \( X_{th} \) is critical to the success of the DA-KLT. If the threshold is set too high, many blocks will be misclassified as belonging to the remaining two classes, thereby degrading the performance of DA-KLT. Through visual assessment, we set the threshold for a block size of \( N \) with variance \( \sigma^2 \) as:

\[
X_{th} = 10 \cdot (2N\sqrt{2} - 1) + \frac{\sigma^2}{N} \tag{2}
\]

The first term in the numerator accounts for the minimum number of pixels that should contribute to the directionality of the block, while the block variance provides some adjustment. Experimentation showed that this threshold always identified strongly directional classes, but missed some weakly directional blocks. A class ID of 9 corresponded to a “flat” block with variance of the block less than a threshold of 15, which was again determined using visual criteria. All remaining blocks were classified as class 10, which corresponded to “textured” blocks. This block classifier is demonstrated in Fig. 4.

Following block classification for the entire test image set, we calculated the statistics for each class. For block size \( N \), all blocks with the same class ID are vectorized column-wise and collated to calculate a covariance matrix of size \( N \times N \). Thus, \( 10 \) covariance matrices are produced. Then the matrix of basis vectors is calculated as in (1). For a good block classification scheme, the basis vectors should demonstrate strong directionality (Fig. 5). For a block classification scheme with good energy compaction, truncation of least dominant principal components in the training data could be carried out to reduce the number of coefficients. Here, dominance of a principal component was measured by the relative size of the corresponding eigenvalue, which translated to the energy carried by the component. With the basis vectors
arranging in order of decreasing dominance, the number of truncated principal components was chosen for class \(m\) so that the first \(L(m)\) (preserved) components carried energy at or just exceeding a given threshold. If the training image set were a good representative of the test images to be compressed, the preserved components would also be the most dominant components in the test images, hence preserving energy at or above the threshold.

Following transformation of the images, the quantised transformation coefficients and class IDs must be transmitted to reconstruct the images. Hence we needed to calculate the minimum average variable-length code (VLC) codeword length as an approximation of entropy, the theoretical lower bound of the average number of bits required to transmit each pixel [1]. VLC tables were trained for each quantised coefficient of each class, and the class IDs. We used a uniform mid-tread quantizer with quantisation step \(\Delta\). The minimum codeword length for a quantisation index \(q\) is calculated, using the VLC table for the \(i^{th}\) coefficient from class \(m\) with quantisation step \(\Delta\) and probability mass function (PMF) of the set of all quantisation coefficients \(p_{m,i,\Delta}(q)\) using the aforementioned VLC table as:

\[
r(q) = -\log_2 (p_{m,i,\Delta}(q))
\]  

Similarly, the minimum codeword length for a block classification ID \(m\) with the PMF of the set of all block classification IDs in the training image set \(p_{\text{class}}(m)\) is calculated as:

\[
r(m) = -\log_2 (p_{\text{class}}(m))
\]  

To account for differences in distributions across coefficients and classes, we implemented a separate VLC for each coefficient from each class by estimating PMFs individually for each quantisation step. This required that we perform block-wise DA-KLT on the training set. As all blocks have already been classified, the transformation coefficients were calculated by applying the KLT using corresponding basis vectors in \(\Phi\) to \(x\) the vector of pixels values from the block (with vectorizing done in a manner consistent with that during the training of the basis vectors) as follows:

\[
y = T \cdot x = \Phi^T x
\]  

The resulting vector of coefficients \(y\) has length \(N^2\) when there is no principal component truncation, and \(L(m)\) when there is. The coefficients were then quantised. PMFs were estimated by calculating the frequency of occurrence for each value of the quantisation index. Furthermore, we also implemented a Huffman coder to demonstrate an achievable bitrate. We used MATLAB’s implementation of the Huffman encoder to simulate our binary lossless entropy encoding system and hence generate the Huffman table (i.e. the codebook). We also made a minor adjustment and encoded coefficients with a single value in their PMF with a 0, i.e. deterministic coefficients are encoded with the codeword 0.

Because the performance of DA-KLT depends strongly on the trained PMF, the choice of training images is vital. This requires not only that all block classes be sufficiently represented in the training set, but also that the range of training coefficient values be representative of the expected coefficient values for test images. At the same time, to optimise the system for general photographic images, we limited the testing images to 50 photos of mostly natural scenes and objects, man-made objects in natural environments, human faces, and aerial photos. While this led to a non-uniform distribution of blocks amongst the classes (with the "flat" and "textured" classes contain more samples than directional classes), we felt this is a reasonable representation of most photographs that have more surface pixels than edge pixels.

\[\text{B. Compression}\]

Image compression in DA-KLT consisted of the same steps of block classification, KLT and quantisation of transform coefficients (Fig. 6). They were implemented using the same algorithm and code as during training. In order to gauge the performance of DA-KLT and gain insight into the effect of changing various parameters, we observed the relationship between peak-signal-to-noise ratio (PSNR) and minimum average bit rates (i.e. average word lengths). The PSNR is a measure of distortion, which arises due to the quantisation of the transform coefficients in the process of compression. The distortion \(d\) is measured as the mean-square-error (MSE) of
the reconstructed image $\tilde{X}$ and original image $X$:

$$d = E[(\tilde{X} - X)^2] = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (\tilde{x}_{i,j} - x_{i,j})^2 / N^2$$

which is clearly the inverse transformation of (5). Then the distortion was calculated using (6). For 8-bit grayscale images, PSNR is calculated as:

$$PSNR = 10 \cdot \log_{10} \left( \frac{255^2}{d} \right)$$

The minimum average codeword lengths for individual transform coefficients’ quantisation indices were calculated using (4) and averaged across all coefficients to produce an overall minimum average codeword length. To complete the encoding half of the DA-KLT image compression system, we also examined the Huffman rates produced by the Huffman encoder. Our Huffman encoder used the Huffman codebook generated in the training stage to encode the quantized transform coefficients. However, since the codebook used approximate, trained PMFs, it may be incomplete, i.e. some DA-KLT coefficients from the test images may not be present in the codebook as they had zero probabilities in the trained PMFs. As this occurred relatively rarely (about 20 per test image on average), we found the closest values represented in the codebook and used their codewords instead. These gaps in the PMF will disappear as we increase the number and diversity of the training images.

### C. Measure of Performance

The performance of DA-KLT was predominantly measured by observing the PSNR-rate curve. The minimum average word length calculated was intended to be an estimation of the entropy of the image. However, as it used trained PMFs, it was not the actual entropy, which is used in the following theoretical calculations. For an image with pixels that can be can be modelled by a Gaussian distribution with variance $\sigma^2$, the SNR is expected to approach the theoretical bound [1]:

$$SNR_{d(R)} = 10 \cdot \log_{10} \left( \frac{\sigma^2}{d} \right) = 20 \cdot R \cdot \log_{10}(2) \approx 6R$$

From (8) and (9), the PSNR is found to have the following bound:

$$PSNR \approx 6R + 10 \log_{10} \left( \frac{255^2}{\sigma^2} \right)$$

Hence, we expect the PSNR-rate curve to approach an asymptote of gradient 6 dB per bit if both the test image has Gaussian statistics or can be modelled as a Gaussian mixture [1], and the PMFs are estimated accurately.

While inspecting the PSNR-rate curves, our region of interest is $30 - 40$ dB of PSNR. A realistic image compression scheme would generally not operate in the region with lower PSNR due to the high distortion, or the region with high rate, which translates to a low compression ratio.

The effectiveness of an orthonormal transform is measured by the amount of energy compaction it is able to achieve [1]. The concentration of energy in a few transformation coefficients also leads to smaller variances for at least some of the non-energy-carrying coefficients. The coding gain $G_T$ for an orthonormal transform is calculated as:

$$G_T = \frac{1}{N^2} \sum_{n=0}^{N^2-1} \frac{\sigma_x^2}{\sigma_y^2}$$

where the numerator is the arithmetic of the variances of the transformation coefficients in a block of size N, and the denominator is the geometric mean of the same quantities. It is to be noted that even though different blocks have different basis functions applied on them based on their class ID, we essentially treat the blocks and consequently the coefficients to be the same after the transformation. Ideally, we would weigh the coefficients from each block in the image based on the occurrence(s) of similar such blocks in the image. We did not pursue this because we defined our transform method which doubly depends on the block classification for both the training and testing purposes.

A small variance in at least one of the transformation coefficients significantly reduces the geometric mean, resulting in a higher coding gain. Hence, a high coding gain is an indication of good energy compaction and consequently effective coding.

While numerical measures such as the PSNR-rate curve and coding gain provide an idea of the performance of compression
schemes, they do not always reflect the visual quality of the reconstructed images.

III. RESULTS & DISCUSSION

A. Overview of performance

To examine the PSNR-rate behaviour of the DA-KLT, we began with a block size of 8 and quantisation steps $\Delta \in \{2^1, 2^{1.5}, \ldots, 2^9\}$. As the quantisation step decreased and the rate increased, the PSNR curves of the test images "Mandrill", "Lena", "Peppers" and "Goldhill" approached slopes of 6 $\text{dB/} \text{bit}$ (Fig. 7) as predicted by theory (10). This suggests that the images could be modelled well as either Gaussian or a Gaussian mixture.

The relative displacement of the PSNR-rate curves was largely due to the variance $\sigma^2$ as shown in (10). The image "Mandrill", which consists of significant variation in pixel values for each block, had a greater variance and hence a lower vertical position in comparison to "Lena" and "Peppers", both of which have large areas of relatively little pixel variation (Fig. 8).

Visual inspection of the reconstructed images showed few discernable artefacts for quantisation step $\Delta < 32$. For higher $\Delta$, blocking artefacts became increasingly noticeable. However, an exception was the highly textured furry regions on "Mandrill". Despite its relatively high distortion, those regions remained visually satisfactory at quantisation steps of 64 and 128 (Fig. 9). This clearly illustrated that the visual redundancy in some images.

We then investigated the performance of DA-KLT by varying the block sizes and sweeping the PSNR-rate curve using the same set of quantisation steps. Changing block size produced progressive improvement in the performance (Fig. 10). For example, for "Mandrill", increasing the block size from 4 to 8 produced a gain of about 1 $\text{dB}$. A further increase from 8 to 16 produced an additional gain of about 0.1 $\text{dB}$. This improvement was partly due to the decrease in overhead bits required to transmit class IDs. More importantly, large blocks were able to capture more joint information and allowed it to be exploited by the compression technique. The PSNR-rate curves for "Mandrill" remained roughly parallel, demonstrating that the class statistics (Gaussian or Gaussian mixture in this case) for various block sizes remained roughly constant. For "Lena", the statistics were not as invariant to block size, so the PSNR-rate curve at larger block sizes had a shallower slope (Fig. 10). Visually, larger blocks produced better reconstruction within the block boundaries (Fig. 11).

B. Effectiveness of Direction Adaptation

To isolate the improvement offered by exploiting the directionality present in the image blocks, and adapting basis vectors to each class, we compared the PSNR-rate curve of DA-KLT to that of a simple trained KLT that uses only one set of basis vectors across all blocks. We found that some improvements are made by DA-KLT comparing to KLT. The amount of improvement varied between test images, corresponding to the number of strongly directional blocks that can take advantage of direction adaptation. For example, direction adaptation improved the high-rate performance for "Mandrill" by about 1 $\text{dB}$, for "Peppers" by 0.85 $\text{dB}$ and for "Lena" by 1.15 $\text{dB}$ (Fig. 12). Coding gain values for the images also confirmed that the DA-KLT was a more effective orthonormal transform coding scheme than KLT (Table I).

At very low rates, the KLT outperformed the DA-KLT in terms of PSNR-rate relationship. For such high distortion, no information was effectively exploited by either transform method, while DA-KLT had the additional overhead of transmitting class IDs. Of course, since realistic compression routines would not be operating in this region, this point was mostly of academic interest.
Fig. 10. Performance of DA-KLT on "Mandrill" and "Lena" for block sizes of 4, 8 and 16. Larger block sizes improved performance, though with decreasing benefit.

Fig. 11. Reconstructed images "Mandrill" and "Lena" with quantisation step of 128 and block size 16. Reconstructed image quality is improved using the larger block size.

Fig. 12. PSNR-rate curve of DA-KLT with comparison to a trained, non-directional-adaptive conventional KLT. Over the region of interest (30 – 40 dB PSNR), DA-KLT consistently outperforms KLT by exploiting the directionality of image blocks to estimate the pixel statistics for each block more accurately.

C. Comparison to DDCT

In the region of interest (30 – 40 dB PSNR), the DA-KLT consistently underperformed in comparison to the DDCT implemented by Chuo-ling Chang of Stanford University (Fig. 13). The main reason may be that the estimated minimum average word length was only an estimation of the entropy rate. Hence the theoretical bound prescribed by the PSNR-entropy curve should be located further to the left of both the DDCT and DA-KLT curves. Then, poor estimation of the PMFs led to a "poorer" performance by the DA-KLT. Given that the DDCT algorithm proposed by Zeng and Fu [2] utilised a brute-force block classification technique that choose the encoding mode (i.e. direction) with the lowest Lagrangian cost, it is also likely our gradient and variance-based classification algorithm was sub-optimal. The overhead in transmitting the class information of the blocks also contributed to the discrepancy. Hence the gap between DA-KLT and DDCT may be reduced through the use of more training images with more representative statistics, and the implementation of a better set of measures for block classification.

One advantage of DA-KLT over DDCT is that the gradient and variance-based block classification method was much faster than the brute-force classification, making the former much more suitable to applications where the encoder has limited computing resources, e.g. mobile devices.

<table>
<thead>
<tr>
<th>Coding Gain</th>
<th>Mandrill</th>
<th>Lena</th>
<th>Peppers</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA-KLT</td>
<td>4.79</td>
<td>54.35</td>
<td>52.35</td>
</tr>
<tr>
<td>KLT</td>
<td>4.46</td>
<td>47.63</td>
<td>41.82</td>
</tr>
</tbody>
</table>

Fig. 13. PSNR-rate curves of DA-KLT and DDCT. DDCT consistently perform better than DA-KLT.
D. Huffman Encoder Performance

The performance of the Huffman encoder was as expected from theory. At high rates (low quantisation step), the Huffman rate approached the estimated entropy (Fig. 14). It is important to note, however, that both the VLC tables and Huffman tables were calculated using the same set of estimated PMFs. Depending on the quality of the PMF estimation, both the Huffman rate and the estimated entropy may still have ample room for improvement. At low rates (high quantisation step), because our implementation used at least one bit to represent each symbol (transformation coefficient or class ID), the Huffman rate was lower-bounded by 1 (Fig. 14).

E. Principal Component Truncation

Because truncation of principal components irreversibly discarded some energy-carrying coefficients from the test images, it placed an upper limit on the PSNR that could be achieved (Fig. 15). The actual values of the limit for different test images depended on the actual image statistics, as the truncation threshold applies exactly only to the training image set.

F. DC Separation and ∆DC Correction

The technique of DC separation and ∆DC correction was implemented in the forward and inverse transforms of DDCT [2] to avoid the "mean weighting defect" associated with using variable-length DCT [4] as shown in the Introduction. While it was proposed that this addition may be able to improve the performance of the DA-KLT, we hypothesised that it would in fact reduce the performance. Because the KLT is not a separable transform, encoding the mean value of the block is equivalent to assigning the constant vector as a basis vector. While the remaining data, now with $N^2 - 1$ degrees of freedom, was encoded using an optimal basis, the overall set of $N^2$ basis vectors was sub-optimal. Hence the DA-KLT performed significantly worse with DC correction (Fig. 16).

IV. Conclusion

In this report, we present DA-KLT as a novel transform coding technique for image compression. DA-KLT was designed to exploit the directionality in images as well as optimise the transformation basis for image content. PSNR-rate curves and coding gain showed that it consistently outperformed the KLT. However, DA-KLT underperformed comparing to the DDCT, possibly due to insufficient training and a sub-optimal classification algorithm. We further explored the effects of principal component truncation and DC separation, and found both modifications to be disadvantageous to the performance of DA-KLT. Our work has combined two approaches that have hitherto remained separate. Hopefully this will become a feasible direction for future improvements in image compression.
techniques.
Suggested future work may involve improving the block classification algorithm and increasing the size and diversity of the training set, so that coefficient distributions may be better estimated. We propose a Lagrangian cost-based framework for choosing the best trained basis from the trained set of bases. Performance of the DA-KLT may also be improved using adaptive block sizes.

APPENDIX A
WORK DISTRIBUTION

TABLE II
TABLE SHOWING TASK DISTRIBUTION BETWEEN TEAM MEMBERS FOR RESULTS THAT MADE IT INTO REPORT

<table>
<thead>
<tr>
<th>Task</th>
<th>Vinay</th>
<th>Wendy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning and code architecture</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>Block classification</td>
<td>90%</td>
<td>10%</td>
</tr>
<tr>
<td>KLT basis calculation (including class statistics calculation and principal component truncation)</td>
<td>10%</td>
<td>90%</td>
</tr>
<tr>
<td>KLT implementation</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>Quantisation</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>VLC training</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>Huffman coder training</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Image reconstruction and distortion calculation</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>VLC rate calculation</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>Coding gain calculation</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Huffman encoder</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>System integration and debugging</td>
<td>20%</td>
<td>80%</td>
</tr>
<tr>
<td>Presentation slides</td>
<td>30%</td>
<td>70%</td>
</tr>
<tr>
<td>Report</td>
<td>45%</td>
<td>55%</td>
</tr>
</tbody>
</table>

TABLE III
TABLE SHOWING TASK DISTRIBUTION BETWEEN TEAM MEMBERS FOR INVESTIGATIVE WORK AND RESULTS THAT COULD NOT MAKE IT INTO REPORT OWING TO TIME CONSTRAINTS

<table>
<thead>
<tr>
<th>Task</th>
<th>Vinay</th>
<th>Wendy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive block size</td>
<td>40%</td>
<td>60%</td>
</tr>
<tr>
<td>Brute force classification</td>
<td>90%</td>
<td>10%</td>
</tr>
<tr>
<td>Synthetic directional block generation</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>DDCT implementation (rendered unnecessary by Chuo-ling’s code)</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>

ACKNOWLEDGMENT

The authors would like to thank Prof. Bernd Girod and Mina Makar for their invaluable guidance throughout this project.

REFERENCES


[5] The training and test images come from the following sources:
  - Fontaine des Terreaux. Copyright photo courtesy of Éric Labouré
  - Pills. Copyright photo courtesy of Karel de Gendre
  - Brandy rose. Copyright photo courtesy of Toni Lankerd, 18347 Woodland Ridge Dr. Apt #7, Spring Lake, MI 49456, U.S.A. (tlankerd@charter.net)
  - Fourvière Cathedral, north wall. F. A. P. Petitcolas
  - Paper machine. Copyright photo courtesy of Karel de Gendre
  - Opera House of Lyon
  - New-York. Copyright photo courtesy of Patrick Luo, University of Cambridge (patrickl@autonomy.com)
  - Arctic Hare. Copyright photos courtesy of Robert E. Barber, Barber Nature Photography (REBarber@msn.com)