AN ASYMMETRIC LOSSLESS IMAGE COMPRESSION TECHNIQUE

Nasir Memon
Computer Science Department
Northern Illinois University
Dekalb, IL 60155
memon@cs.niu.edu

Khalid Sayood
Department of Electrical Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0551
ksayood@eecomm.unl.edu

1. MOTIVATION

Lossless image compression is often required in situations where compression is done once and decompression is to be performed a multiple number of times. Since compression is performed only once, the time required for compression is not a critical factor while selecting an appropriate compression scheme for such applications. What is more important is the amount of time and memory needed for decompression and the compression ratio obtained. For example, in applications like archiving of image data or distribution of image data on a CD-ROM or other suitable media, a vast amount of resources are usually available for the compression step but not for the decompression step. While the images are archived or prepared for distribution on high-performance computing platforms, end-users who may need to use the data for scientific or entertainment purposes are often equipped with low-end workstations or even personal computers. Hence, while selecting an appropriate compression scheme care needs to be taken to ensure that the end-user will be able to decompress the data quickly and with a minimum of resources, both in terms of memory and computation.

Compression schemes that require differing levels of complexity and computational power in the encoder and the decoder are referred as asymmetric techniques. A classic example of an asymmetric lossy compression technique is full search vector quantization. Here, in the compression step, the codebook is exhaustively searched for the best matching code vector. However, once this best match is located, only its index needs to known during decompression and an approximation of the original vector can quickly be obtained by a simple table look-up operation.

While there exist many asymmetric techniques for the lossy compression of image data, most techniques reported for lossless compression of image data have been symmetric. In this paper we present a new lossless compression technique that is well suited for asymmetric applications. It gives superior performance compared to other lossless compression techniques reported in the literature. Hence, it can also potentially be adapted for use in symmetric applications that require high compression ratios.

2. PREDICTION PATTERNS

Lossless data compression techniques generally consist of two distinct and independent components - modeling and coding. In lossless image compression, the task of modeling is often split into two stages. In the first stage, a prediction model is used to predict pixel values and replace them by the error in prediction. In order to encode the prediction residuals efficiently, an error model is constructed in the second stage to capture any structure that remains after prediction. The first step is also called decorrelation and the second step is referred to as error modelling. For a recent survey on lossless image compression techniques, see [3].

Linear prediction techniques are often used in the prediction step for lossless image compression. Such techniques use a linear combination of neighboring pixels as the prediction for the current pixel. When using linear predictive techniques, one first decides a model order and a neighborhood set based on which prediction is to be performed. Then, given an image, coefficients that minimize the mean squared prediction error can be computed. These coefficients can be computed either for the entire image or on a block-by-block basis. However, stationarity assumptions required by such techniques are often invalid even for image blocks. Besides, minimizing the mean square prediction error does not necessarily translate to a minimization of the entropy of the prediction error. Therefore, predictors which are optimal in the sense of minimizing the mean squared prediction error have not proven to be very successful in lossless image compression applications. In the rest of this section we describe a new approach for adapting the prediction process, that is especially suitable for asymmetric lossless compression applications.

Given a set of prediction schemes $\mathcal{F} = \{f_1, f_2, \ldots, f_r\}$, we would like to select for each pixel the prediction scheme that gives the smallest prediction error. However, the cost
of specifying the prediction scheme for each pixel can be prohibitive. One way to reduce the cost is to use the same prediction scheme for a \( k \times k \) block of pixels. The overhead required to encode the best prediction scheme from \( \mathcal{F} \) for a specific block is \( \log_2 N \) bits per pixel. For \( r = 8 \) and \( k = 8 \) this is 0.05 bits per pixel. Note that such a technique is asymmetric. During compression all possible prediction schemes have to be tested for each block. For decompression only the best scheme, as specified by the side-information, is used.

We generalize the above approach in a natural manner. Since there are many segments within an image which possess similar patterns of inter-pixel relationships, if we partition the image into \( k \times k \) blocks, it would be reasonable to expect to find several blocks in which a similar pattern of prediction schemes gives optimal performance. Hence, given a set of linear prediction schemes, we define a prediction pattern \( T = (t_{ij}) \) to be a \( k \times k \) array with each element of the array representing an index of a prediction scheme from the given set of \( \mathcal{F} \) of predictors. Given an image, a set of prediction schemes \( \mathcal{F} \), and a codebook of prediction patterns, we decorrelate each block in the image by identifying the best prediction pattern from within the codebook. Where by best we mean the prediction pattern that yields the minimum absolute sum of prediction errors over all the entries in the codebook. We then transmit the index of this prediction pattern to the receiver followed by an encoding of the prediction residuals. It is clear that such a decorrelation technique is asymmetric.

The problem is to design an optimal set of prediction schemes and an optimal codebook of prediction patterns, given an image (or class of images). The problem stated in this manner seems formidable. In order to make it more tractable, we impose certain structure on the set of predictors by initializing it in a suitable manner. Having initialized \( \mathcal{F} \), we iteratively descend to a locally optimal predictor/codebook design by repeating the following two steps. In the first step we design an optimal codebook for current choice of \( \mathcal{F} \). Then in the second step we derive coefficients for prediction schemes in \( \mathcal{F} \) based on the current optimal codebook that minimize a given cost criteria. The codebook now is no longer optimal with respect to the set of predictors \( \mathcal{F} \), so we repeat step 1 again, followed by step 2 and so on. It is easy to see that the above procedure converges and we stop when the reduction in prediction errors is less than a pre-decided threshold.

An optimal codebook for the current choice of \( \mathcal{F} \) can be constructed by using the generalized Lloyd algorithm (GLA) [2]. Let \( B = \{ B_1, B_2, \ldots, B_t \} \) be a training set of image blocks coming from the image or family of images for which we are constructing a codebook of prediction patterns \( C = \{ T_1, T_2, \ldots, T_m \} \). We define the distortion incurred for using the prediction pattern \( T \) on block \( B \) to be

\[
W(T, B) \text{ given by}
W(T, B) = \sum_{i=1}^{k} \sum_{j=1}^{k} |E_x(B, i, j, t_{ij})| \tag{1}
\]

Where \( E_x(P, i, j, k) \) denotes the prediction error when the prediction scheme \( f_k \) is used on the pixel \( P[i, j] \). Let \( T_{(C, B)} \) denote the prediction pattern from \( C \) which yields the minimum sum of absolute weights when used on block \( B \). Our goal is to construct a codebook \( C \) such that \( \sum_{i=1}^{m} W(T_{(C, B)}, B) \) is minimized. It is easy to show that given a cluster \( B = \{ B_1, B_2, \ldots, B_m \} \) of image blocks each of size \( k \times k \), the prediction pattern \( T \) that gives the minimum distortion \( \sum_{i=1}^{m} W(T, B_i) \) is given by

\[
T[i, j] = \min_{s} \sum_{k=1}^{m} E_x(B_k, i, j, s) \quad 1 \leq s \leq |\mathcal{F}| \tag{2}
\]

This gives us the notion of a centroid prediction pattern, given a cluster of image blocks and we can now use the GLA algorithm.

For a given codebook, coefficients for the prediction schemes in \( \mathcal{F} \) that minimize a given cost criteria can be computed by collecting all pixels that get mapped to a particular prediction scheme and then computing optimal coefficients. If the mean square error is chosen as the cost criteria to minimize then it is easy to compute optimal coefficients using standard linear regression techniques. However, as mentioned before, in lossless image compression applications, coefficients that minimize the absolute sum of prediction errors results in a lower order entropy of prediction errors as compared to coefficients that minimize \( \text{mse} \) [4]. This is because, if we assume that prediction errors are Laplacian (which they usually are) then minimizing the absolute sum of prediction errors leads to minimization of zero-order entropy.

In table 1 we give the zero order entropy of prediction errors obtained by a preliminary implementation of our

<table>
<thead>
<tr>
<th></th>
<th>8 × 8 × 256</th>
<th>Block JPEG</th>
<th>JPEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balloon</td>
<td>2.88</td>
<td>3.01</td>
<td>3.17</td>
</tr>
<tr>
<td>Barb 1</td>
<td>4.79</td>
<td>5.01</td>
<td>5.29</td>
</tr>
<tr>
<td>Zelda</td>
<td>3.86</td>
<td>3.97</td>
<td>4.15</td>
</tr>
<tr>
<td>Hotel</td>
<td>4.48</td>
<td>4.63</td>
<td>4.95</td>
</tr>
<tr>
<td>Barb 2</td>
<td>4.90</td>
<td>5.06</td>
<td>5.23</td>
</tr>
<tr>
<td>Board</td>
<td>3.67</td>
<td>3.83</td>
<td>4.18</td>
</tr>
<tr>
<td>Girl</td>
<td>3.88</td>
<td>4.07</td>
<td>4.26</td>
</tr>
<tr>
<td>Gold</td>
<td>4.51</td>
<td>4.66</td>
<td>4.85</td>
</tr>
<tr>
<td>Boats</td>
<td>4.04</td>
<td>4.21</td>
<td>4.46</td>
</tr>
</tbody>
</table>

Table 1: Entropy of error image using local codebooks.
technique on some 576 × 720 images from the JPEG test set. The original images are color images with three different color planes - Y, U and V. We took the Y image as our test image. Column 1 shows the entropy of the error image, in bits per pixel, achieved for the test set for a codebook of size 256 with 8×8 blocks. The overhead for transmitting the codebook and also the index of the best prediction pattern for each block are included in the figures. Codebook initialization for each pixel in the block, which was then placed in the codebook. The set of eight lossless JPEG predictors [5] were used as the initial set of predictors and coefficients were optimized in each iteration by minimizing mse. The second column in table 1 gives results obtained with the same block size by using the best of the eight JPEG predictors on each image block. The overhead of specifying the prediction scheme for each block is also included in the figures. Figures with the best of the eight JPEG predictors used on the entire image are listed in the last column.

From the above we see that even for single frame images we get significant improvement over the JPEG predictors. More improvement can surely be obtained when codebooks are constructed for specific types of images. For example, when compressing multi-spectral images a single codebook can be constructed for multiple bands thereby significantly reducing the overhead needed in transmitting the codebook. Assuming we use a set of eight different predictors and a codebook of size n is used for m successive bands, the cost overhead per pixel becomes \( \frac{2m+nk^2}{mN} \) bits per pixel for an \( M \times N \) image with \( k \times k \) blocks. With \( M = N = 512 \), \( k = 8 \), \( m = 8 \) and even with codebook as large as 1024, the overhead for a local codebook is only 0.12 bits per pixel. With hyper-spectral images, which have more than 200 bands, the overhead becomes negligible. Results with such images will be the subject of future work.

3. ERROR MODELLING

If we treat the prediction residuals as iid, they can be encoded at rates close to the zero order entropy by using some standard variable length coding technique. However, structure in the prediction residuals can be further exploited to obtain lower bit rates by using an error model. Many sophisticated error modelling techniques have been reported in the literature. For an asymmetric application, we choose to model the prediction errors by a composite source model. That is, we model the data as an interleaved sequence generated by multiple sub-sources. Each sub-source has its own model and associated probability distribution. Each subsequence is treated as an iid sequence generated by the corresponding sub-source and encoded by some well known technique like adaptive arithmetic coding. We construct the composite source model by partitioning the error image into blocks (not necessarily of the same size used in the prediction step) and classifying each block as emanating from a particular sub-source. The label indicating the sub-source to which the block belongs is transmitted along with an encoding of the block. The receiver then uses the appropriate variable length code to decode the block. Clearly, this technique is asymmetric.

In order to model the prediction residuals as a composite source, we need to develop an efficient technique to classify the blocks. Stated more precisely, given a set of blocks
\[ B = \{B_1, B_2, \ldots, B_m\} \]
from the error image, we wish to select a switching function that maps a given block \( B_i \) to the sub-source \( s_i \), where \( 1 \leq s_i \leq k \) such that the k-source entropy \( H_k(B) \) of \( B \) is minimized. Where
\[ H_k(B) = \sum_{s=1}^{k} P_s \cdot H(B^s) \]
with \( P_s = \frac{|B^s|}{|B|} \) and \( H(B^s) \) being the single source entropy of the set of blocks from \( B \) that get mapped to sub-source \( s \). We suspect that the problem stated above is intractable. However, we can design heuristic solutions. One simple heuristic to construct a composite source model consisting of \( k \) sub-sources is to classify the blocks based on some measure of activity within the block. Different measures of activity can be used. Having decided a measure, we compare it to a set of \( k \) predetermined thresholds \( T_1 < T_2, \ldots, < T_k \). A given block is classified to sub-source \( i \) if the activity measure for the block was less than or equal to \( T_i \) but greater than \( T_{i-1} \). Given a residual image, optimal thresholds that minimize the k-source entropy of prediction errors can be computed efficiently by using standard dynamic programming techniques. A number of different activity measures were tried. The absolute sum of prediction errors within a block was observed to be quite robust and gave superior performance over the test set of images.

<table>
<thead>
<tr>
<th>Codebook Size</th>
<th>8 × 8 × 256</th>
<th>UCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balloon</td>
<td>2.78</td>
<td>2.81</td>
</tr>
<tr>
<td>Barb 1</td>
<td>4.43</td>
<td>4.44</td>
</tr>
<tr>
<td>Zelda</td>
<td>3.74</td>
<td>3.80</td>
</tr>
<tr>
<td>Hotel</td>
<td>4.26</td>
<td>4.28</td>
</tr>
<tr>
<td>Barb 2</td>
<td>4.66</td>
<td>4.57</td>
</tr>
<tr>
<td>Board</td>
<td>3.53</td>
<td>3.57</td>
</tr>
<tr>
<td>Girl</td>
<td>3.74</td>
<td>3.81</td>
</tr>
<tr>
<td>Gold</td>
<td>4.41</td>
<td>4.45</td>
</tr>
<tr>
<td>Boats</td>
<td>3.86</td>
<td>3.85</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>3.93</strong></td>
<td><strong>3.95</strong></td>
</tr>
</tbody>
</table>

Table 2: Entropy of error image using local codebooks.
resulted in poorer bit rates as compared to the first heuristic. In

table 2 we list the k-source entropy of prediction errors obtained by using the first heuristic on the residual images obtained from the prediction step described in section 2 and listed in table 1. Column one lists entropies obtained with 8 sub-sources and a general set of thresholds for all images. The overhead required to transmit sub-source label for each block is also included in the figures. In column 2 we list the entropy rates reported in [6], with UCM, a highly complex symmetric technique designed not as a practical alternative but as an ideal that provides performance bounds that can be expected from more practical lossless image compression techniques. We see that the asymmetric technique we report here already gives us slightly better performance than what they report.

4. CONCLUSIONS AND FUTURE WORK

Our preliminary implementations have resulted in bit-rates that are competitive with symmetric techniques that have comparable complexity. These implementations use a local codebook constructed for each image and the cost of transmitting the codebook is counted as overhead. We expect to get further improvements for codebooks designed for a family of correlated images like multi-spectral images or 3D medical images. However, the most important aspect of our technique is the fact that it is asymmetric, making it useful in suitable applications.

Nevertheless there are some aspects of our technique that are ad-hoc and need improvement. First, we have initialized the set of prediction schemes \( \mathcal{F} \) in an ad-hoc manner. The question of selecting the best set of predictors for a given image or family of images needs to be addressed in a more formal setting. Next, the size of the codebook was also selected in an ad-hoc manner. Perhaps alternative clustering techniques like PNN [1] can be used to jointly optimize the size of the codebook along with the cost of encoding the image with respect to the codebook. There are other improvements that can be made on the basic technique we have reported. The performance of the GLA algorithm can be improved by periodically splitting large clusters and also by stochastic relaxation techniques to avoid local minima. Many alternate distortion measures are also possible while selecting the prediction pattern for a specific block. Another way to improve the codebook is to separate image blocks into different categories based on the texture and edges present within the block. A different codebook can then be designed for each such type of block. The error modelling scheme needs to be improved and better heuristics need to be designed for the problem described in section 3. It is possible to formulate the problem as a combinatorial optimization problem and this can perhaps lead to better heuristics. Finally, the codebooks can be constructed and maintained in an adaptive manner. This would lead to a symmetric algorithm. The performance of such an algorithm needs to be compared to other well known symmetric techniques.

5. REFERENCES


