Lossless Compression of Hyperspectral Images Using Adaptive Prediction and Backward Search Schemes

CHENG-CHEN LIN AND YIN-TSUNG HWANG
Department of Electrical Engineering
National Chung Hsing University
Taichung, 402 Taiwan

In this paper, an effective lossless compression scheme for hyperspectral images is presented. The proposed scheme is based on a table look-up approach in prediction and employs two novel measures to improve the compression performance. The first measure takes advantage of the spatial data correlation and formulates the derivation of a spectral domain predictor as a process of Wiener filtering. The derived predictor is considered statistically optimal provided that the data within a small context window are stationary. This property holds in most cases due to spatial data correlation. Under the Wiener filtering framework, the proposed predictor can be extended from one-tap to multi-tap prediction to further improve performance. In the second measure, a backward search scheme is used instead of look-up tables, which reduces the memory storage requirement drastically and achieves performance equivalent to that obtained using multiple look-up tables. The search effort is greatly reduced using the quantization index approach. Simulations on parameter settings and refinements on entropy coding are conducted to fine-tune performance. Experiments on 5 sequences of AVIRIS images show that the proposed scheme can yield an average compression ratio of as high as 3.85.

Keywords: hyperspectral imaging, lossless compression, adaptive prediction, wiener filtering, context-based arithmetic coding

1. INTRODUCTION

Hyperspectral imaging has become an active research topic in recent years due to its wide-spread applications in areas such as resource management, agriculture, mineral exploration, and environmental monitoring. However, the growing scientific and technological demands in spatial and spectral resolution have drastically increased the data volume of hyperspectral images. For example, a scene data cube recorded by the NASA JPL Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [1] contains 224 bands of $512 \times 614$ sized images. Fig. 1 shows some image bands of the Cuprite sequence in the AVIRIS library. Since each hyperspectral pixel is represented using 16-bit precision, the data storage requirement for one scene is about 134 MB. For a hyperspectral image that consists of several scenes, the data storage requirement can easily reach 400-600MB. A large data volume requires a massive amount of storage space and consumes excessive transmission bandwidth. To alleviate these problems, state of the art image/data compression techniques are employed. Existing compression standards such as MPEG-4 and H.264 target mainly video sequences. These schemes are not efficient for the hyperspectral images because they fail to fully exploit the spectral data correlation. Compression schemes dedicated to hyperspectral images can be either lossy or lossless. Lossy schemes [2], which trim off
uninteresting data along with compression, are usually tailored to specific analysis purposes, e.g., detection or classification. Lossless schemes, on the other hand, are meant for general analysis as they preserve all raw data. In this paper, we focus on lossless schemes.

Current hyperspectral image compression schemes can be classified by the techniques used to explore data correlations. These include vector quantization (VQ), transform coding, predictive coding, and look-up table (LUT). VQ based approaches [3, 4] partition hyperspectral bands into blocks and co-located pixels (pixels with the same spatial location in each band) within the block are set to form a vector. VQ indexes and prediction residuals are compressed using entropy coding. The compression ratios of VQ based schemes are limited due to the practical constraint in the VQ code book size. Computation complexity in the code book search negatively affects the processing speed.

Transform based schemes largely follow the framework of 3-D image/video compression and treat spatial and spectral data correlations indiscriminately. The goal is to jointly explore the spectral and the spatial data correlations. In [5, 6], a 3-D integer wavelet transform was applied. The tree structured coefficients were then coded by zero tree based bit-plane coding schemes. In [7], an integer wavelet packet transform (WPT) with unitary scaling was adopted. The coding schemes were 3-D Embedded Zero Block Coding (3-DEZBC) plus context based adaptive arithmetic coding. The scheme reported in [8] conforms to part 2 of the JPEG 2000 standard; i.e., it uses a hybrid 3-D wavelet transform plus context based adaptive arithmetic coding (CBAAC). The discrete cosine transform (DCT) and the Karhunen-Loeve transform (KLT) have also been employed in the literature [9].

Predictive coding based compression techniques employ spatial, spectral, or hybrid predictors to decorrelate the image data. In [10], an inter-band version of CALIC, which adaptively selects inter-band (spectral) and intra-band (spatial) predictions, was presented. The predictor coefficients were calculated adaptively using the data correlation within the prediction context windows. This scheme was further enhanced in [11] (M-CALIC), which used only a multi-band predictor with fixed coefficients. The coefficients, however, needed
to be optimized by off-line linear regression procedures. In [12], a Kalman filter was employed to perform the prediction and the prediction residuals were coded using context-based arithmetic coding. The work reported in [13] focused on a low complexity predictor using simple additions and comparisons. The results showed limited compression efficiency. The performance enhancement measures proposed for predictive coding based approaches include sophisticated prediction schemes such as least square (LS) formulation [14-16], additional decorrelation processes [18, 19], and data pre-processing techniques such as band reordering and clustering [20-22].

For LUT schemes, the basic idea is to use existing pixel values instead of calculated values from a predictor function for prediction. These schemes are particularly effective for calibrated hyperspectral images using a 16-bit per pixel format. In [23], a table indexed by the distinct pixel values in the previous band was first established. The table provided an updated mapping of pixel values between two co-located pixels in the previous and current bands. To predict a current pixel, the pixel value of the co-located pixel in the previous band is used as a key to search a look-up table. This scheme was enhanced in [24], which employed a second look-up table that recorded the replaced entries from the first look-up table. Two values were returned from the look-up tables and the one closest to the LAIS (Locally Average Inter-band Scaling) estimate was chosen as the prediction of the current pixel. To reduce the look-up table size, a quantization technique with band adaptive quantization factors was introduced in [25].

In this paper, a lossless compression scheme, which is based on a LUT approach in prediction and employs two novel measures to improve the compression performance, is presented. The first measure takes advantage of the spatial data correlation and adaptively derives the prediction coefficients of the spectral domain predictor as a process of Wiener filtering. In the second measure, a backward search scheme is applied instead of using look-up tables, which reduces the memory storage requirement drastically and achieves performance equivalent to that obtained using multiple look-up tables.

2. PROPOSED COMPRESSION SCHEME

The block diagram of the proposed compression scheme is shown in Fig. 2. It consists of 5 major modules, i.e., intra prediction, inter-/multi-band prediction, backward search index (BSI), quantized index, and entropy coding. This scheme supports both intra- and inter-band predictions.

Fig. 2. Block diagram of the proposed lossless hyperspectral image compression scheme.
However, only the first band of the hyperspectral images is coded by intra-band prediction. The remaining images are all coded by an inter-band prediction scheme. Since the proposed scheme is based on LUT, the prediction scheme obtains a prediction reference value $P_{\text{ref}}$ instead of a direct prediction value. Let $\phi$ be the pixel value of the pixel in the previous band co-located with the current pixel (under prediction). The BSI module searches the previous band to identify the pixels with the same pixel value as $\phi$. The corresponding pixel values in the current band of those identified pixels are compared with the $P_{\text{ref}}$ value. The one with the closest value is selected as the inter-band prediction value $P_{\text{inter}}$. Unlike $P_{\text{ref}}$ which is a calculated value, $P_{\text{inter}}$ holds an existing pixel value in the current band available during the decoding process. The quantized index module corresponds to a relaxation measure of the BSI operations. Instead of searching for pixels with pixel values identical to $\phi$, the difference between the two pixel values is quantized to reduce the search effort. Finally, the entropy coding module performs adaptive arithmetic coding on prediction residuals.

### 2.1 Intra Prediction

Intra prediction is applied only to the first image along the spectral line. The median predictor defined in JPEG-LS [26] is employed here due to its simplicity and efficiency for still images. Fig. 3 shows the context model of JPEG-LS, where $y$ is the current pixel and $NW$, $N$, and $W$ denotes three neighboring pixels, respectively. The estimate of pixel $y$ is as follows:

$$\hat{y}_{\text{intra}} = \begin{cases} 
\min(N, W), & \text{if } NW \geq \max(N, W) \\
\max(N, W), & \text{if } NW \leq \min(N, W) \\
N + W - NW, & \text{otherwise.} 
\end{cases}$$

(1)

Fig. 3. Context window for intra-prediction.

### 2.2 Inter-/Multi-band Prediction

In the original LUT scheme [23], only one table is used to record the mapping between the two co-located pixels in adjacent bands, i.e. previous versus current bands. The table is updated constantly such that each table entry always has the latest mapping. This scheme was extended in the LAIS-LUT scheme [24], where two look-up tables were used to record the two most recent mappings. The prediction reference value $P_{\text{ref}}$ is obtained by multiplying the value $\phi$ with a locally averaged inter-band scaling (LAIS) factor. The $P_{\text{ref}}$ value is then compared with the two corresponding entries in the two look-up tables and the one with the closer value is selected as the prediction value $P_{\text{inter}}$. To enhance the LUT
scheme, we observed that a more accurate prediction reference value $P_{ref}$ leads to better compression result. A simple experiment was conducted to verify this observation. As shown in Table 1, if the prediction reference value $P_{ref}$ always ideally matches the value of the current pixel, the compression ratio of using two look-up tables can be improved to 3.75 compared with 3.47 and 3.59 for the LAIS-LUT and LAIS-QLUT-OPT [25] schemes, respectively. This experiment was also used to evaluate the effectiveness of increasing the number of look-up tables. For the LAIS-LUT scheme with ideally matched $P_{ref}$ values, the compression efficiency improves consistently until the number of look-up tables reaches 9. Beyond that point, the compression efficiency then starts to decline. In practice, such ideally matched $P_{ref}$ values are not available. For the $P_{ref}$ calculation method used in LAIS-LUT scheme, the best result was obtained when the number of look-up tables is 5. The compression ratio improvement, however, is only 0.08. For the more sophisticated LAIS-QLUT-OPT scheme, the best result was obtained when the number of look-up tables is 3. The compression ratio is enhanced by only 0.03. This shows that the effectiveness of increasing the number of LUTs is diminished when less efficient $P_{ref}$ calculation schemes are used.

In this paper, two $P_{ref}$ calculation schemes are proposed to improve the accuracy of $P_{ref}$ values. The first one is for an inter-band predictor; it uses information from only one previous band to calculate the $P_{ref}$ values. The second one is for a multi-band predictor; it extends the first scheme by using multiple previous bands. Let $y$ be the current pixel value. An inter-band predictor of $P_{ref}$ has the following prediction formula:

\[
\hat{y} = ax + b, \tag{2}
\]

where $\hat{y} (= P_{ref})$ is the prediction reference value and $x (= \varphi)$ is the pixel value of the co-located pixel in the previous band. Under the least mean square error criterion shown in Eq. (2), statistically optimal prediction coefficients $a$ and $b$ can be obtained using the expressions given in Eqs. (3) and (4).

\[
a = \frac{E\{xy\} - m_{x}m_{y}}{\sigma_{x}^{2}}, \tag{3}
\]
Fig. 4. Definitions of context windows (a) The next to the previous band; (b) The previous band; (c) The current band.

\[ b = m_y - am_x, \]  
where \( m_x = E\{x\} \) and \( m_y = E\{y\} \) are the means of the random variables \( x \) and \( y \), respectively. Given a context window containing \( M \) neighboring pixels, coefficient \( a \) can be approximated by:

\[
a = \frac{M \sum_{i=1}^{M} x_i y_i - \sum_{i=1}^{M} x_i \sum_{i=1}^{M} y_i}{M \sum_{i=1}^{M} x_i^2 - \left( \sum_{i=1}^{M} x_i \right)^2},
\]

where \( x_i \) and \( y_i \) are pixels within the context windows in the previous and the current band, respectively. Fig. 4 shows the definitions of context windows in the current, the previous and the next to the previous bands. “w” denotes pixels co-located with the current pixel in the second previous band. Since the inter-band data correlation is very high in hyperspectral images, we can further extend the inter-band predictor to a multi-band predictor, which utilizes more than one previous band in the prediction. A linear 2-tap multi-band predictor has the following prediction formula:

\[
y^* = \alpha w + \beta x + \gamma
\]

and \( \beta \) are prediction coefficients and \( \gamma \) is a prediction offset. Using the least mean square error criterion, \( \alpha \) and \( \beta \) can be derived as the solutions of the following Wiener-Hopf equation:

\[
\begin{bmatrix}
\sigma^2_w & \sigma_{wx} \\
\sigma_{wx} & \sigma^2_x
\end{bmatrix}
\begin{bmatrix}
\alpha \\
\beta
\end{bmatrix}
=
\begin{bmatrix}
\sigma_{wy} \\
\sigma_{xy}
\end{bmatrix}.
\]
The prediction is now equivalent to the process of Wiener filtering. For the context windows defined in Fig. 4, the following statistical parameters can be approximated:

\[
\sigma_i^2 = \mathbb{E}\{x_i^2\} - m_i^2 = M \sum_{i=1}^{M} x_i^2 - \left(\sum_{i=1}^{M} x_i\right)^2,
\]

(8)

\[
\sigma_w^2 = \mathbb{E}\{w_i^2\} - m_w^2 = M \sum_{i=1}^{M} w_i^2 - \left(\sum_{i=1}^{M} w_i\right)^2,
\]

(9)

\[
\sigma_{wy} = \mathbb{E}\{w_y\} - m_w m_y = M \sum_{i=1}^{M} w_i y_i - \sum_{i=1}^{M} w_i \sum_{i=1}^{M} y_i,
\]

(10)

\[
\sigma_{wx} = \mathbb{E}\{w_x\} - m_w m_x = M \sum_{i=1}^{M} w_i x_i - \sum_{i=1}^{M} w_i \sum_{i=1}^{M} x_i,
\]

(11)

\[
\sigma_{xy} = \mathbb{E}\{xy\} - m_x m_y = M \sum_{i=1}^{M} x_i y_i - \sum_{i=1}^{M} x_i \sum_{i=1}^{M} y_i.
\]

(12)

This leads to the solutions:

\[
\alpha = \frac{\sigma_x^2 \sigma_{wy} - \sigma_w \sigma_{xy}}{\sigma_w^2 \sigma_x^2 - (\sigma_{wx})^2}, \quad \beta = \frac{\sigma_x \sigma_{xy} - \sigma_w \sigma_{wy}}{\sigma_w^2 \sigma_x^2 - (\sigma_{wx})^2}.
\]

(13)

The predictor equation in Eq. (6) now becomes:

\[
y^\prime = \alpha (w - m_w) + \beta (x - m_x) + m_w.
\]

(14)

The multi-band predictor is expected to have better prediction efficiency at the cost of computation complexity.

### 2.3 Backward Search Index

As shown in Table 1, prediction efficiency of the LUT approach can be enhanced by adopting more look-up tables to record more mappings. However, this dramatically increases memory usage. For 16-bit resolution pixels, the number of distinct pixel values is 65536, which defines the number of table entries. Each table entry is 16 bits wide. The total memory size for one look-up table is 128 KB. The memory usage for 224 bands amounts to 28 MB. This requirement increases further if multiple look-up tables are adopted. To mitigate the memory usage issue, an on-the-fly search algorithm called the backward search index (BSI) is employed. The pseudo code of the BSI algorithm is shown in Fig. 5.

The outer loop indexed by \( v \) is in charge of \( L \) pixel lines under search. For each search line, the starting point \( sp \) is set first. The inner loop indexed by \( h \) corresponds to the search within a pixel line. The searched pixel value closest to the prediction reference value \( P_{ref} \) is always tracked. If the difference \( P_{err} \) is no greater than the threshold value, early termination occurs. Due to spatial data correlation, the search can be confined to a small region surrounding the current pixel. The multi-LUT approach can be easily implemented in this scheme by searching for multiple pixels in the previous band with an identical pixel value \( \phi \). A maximum match count is set and the pixel that leads to the best match of the calculated \( P_{ref} \)
// \(b\): current band, \((i, j)\) coordinate of current pixel
// \(L\): number of line buffers for search
// \(W\): width of the image
min_error = 65536;
count = 0;
For \(v = i\) to max\((i - L), 0\)  // \(v\): line index
  If \((v == i)\) 
    \(sp = j - 1;\)  // \(sp\): starting point in a line
  Else 
    \(sp = W - 1;\)
For \(h = sp\) to 1  // \(h\): pixel position in a line
  If (LineBuffer\((b - 1, v, h) == LineBuffer\((b - 1, i, j)\))
    \(P_{err} = \text{abs} (\text{LineBuffer} (b, v, h) - P_{ref});\)  // with min. predict error
    \(P_{inter} = \text{LineBuffer} (b, v, h);\)
  If \((P_{err} <= \text{Threshold})\)
    return \(P_{inter};\)  // early termination
    count++;
  If (count == MaxCount)
    return \(P_{inter};\)
}

Fig. 5. Backward search index algorithm.

Fig. 6. Example of the BSI algorithm.

value is selected. The maximum match count is equivalent to the number of look-up tables used. To further decrease the search effort, a threshold for early termination of the search process is defined. When the difference between the selected pixel value and the \(P_{ref}\) value falls below the threshold, the search will stop.

An example of the BSI algorithm is shown in Fig. 6. Assume that the calculated \(P_{ref}\) value is 631 and that the maximum match count is set to 7 \((\text{MaxCount})\). If no relaxation is applied to the search process, \(i.e\). the threshold value is 0, 7 pixels in the previous band with pixel values equal to 620 (marked in gray) are identified. Their corresponding pixels in the current band have 7 distinct pixel values, \(i.e\)., 627, 635, 632, 626, 634, 631, and 625. The pixel value of 631 is closest to the calculated \(P_{ref}\) value and is thus selected as the prediction reference value \(P_{inter}\). This requires a total of 31 pixel searches. If the search threshold is set to 2, the search process stops on the third encounter of matched pixels in
the previous band whose corresponding pixel value in the current band is 632. Only 12 pixel searches are needed in this case, which greatly reduces computation complexity.

2.4 Quantized Index

The Quantized index is a measure adopted in the look-up table approach to reduce the table size. Instead of using the original pixel value as the table index, a quantized index is used. This measure can be applied to the proposed BSI scheme to reduce the search effort. Optimal quantization factors, however, vary from band to band. Although an exhaustive search method can be employed to determine the quantization factors, it is too costly to implement. Therefore, heuristic quantization factors were adopted in LAIS-QLUT [25]. In this paper, experiments were first conducted to evaluate the performance difference between schemes using on-line calculated quantization factors and those using fixed numbers. For on-line calculation, pixels in the same band are sampled to obtain an estimate of the mean of the pixel values. The value is then classified into one of the six categories using threshold values of 512, 768, 2048, 3072, and 4096. The corresponding quantization factors used in each category are 4, 8, 16, 32, 64, and 128, respectively. For the fixed quantization factor approach, only 7 quantization factors are explored in each band to determine the most efficient one. The 7 quantization factor candidates are 4, 8, 16, 32, 64, 128, and 256. Four hyperspectral sequences, i.e., Cuprite, Jasper Ridge, Lunnar Lake, and Moffett Field were used in the experiments. The experiment results indicate a strong correlation in optimal quantization factors among the 4 test sequences. For the fixed quantization factor approach, 70% of the 224 bands in all 4 sequences share the same optimal quantization factors. 22% of them have identical quantization factors in 3 sequences. The remaining 8% have equal quantization factors in 2 sequences. The resultant quantization factors for the 224 bands are illustrated in Fig. 7. The experimental results also show that the fixed quantization factor approach outperforms the on-line calculated scheme by a small margin (0.06 bpp) in terms of compression efficiency. The fixed quantization factor approach is thus adopted in our scheme. Compared with the heuristic quantization factors reported in [25], the selected quantization factors are all powers of two, which avoids division. Using the quantized index, the search effort in the BSI scheme can be further reduced.

Fig. 7. Quantization factors used in each band.
2.5 Entropy Coding

In our scheme, a simple first-order entropy coder is adopted instead of a context-based (high order) entropy coder. A first-order entropy coder examines only the prediction error of the current pixel. We follow the framework presented in [23] and use separate entropy models for the sign bit and the magnitude. The sign bit model has 3 distinct symbols: “0”, “1”, and “2”. The symbol “2” stands for the case of zero prediction error while symbols “0” and “1” represent the positive and the negative prediction errors, respectively. The magnitude model contains 130 symbols (“1” to “129” plus an “ESC”). If the magnitude of the prediction error is not larger than the threshold value (129), an AAC scheme is applied to perform the coding. However, if the magnitude is greater than the threshold value, an escape symbol “ESC” is coded first.

After subtracting the threshold from the magnitude, the resultant 16-bit data is divided into sub-words for separate coding. In [23], the 16-bit data is split into a high byte and a low byte. The design concern here is once again the memory usage of the entropy models. If the hyperspectral images are scanned in a band-interleaved-by-line (BIL) format, the entropy models of all 224 bands must be kept in memory during the coding and the decoding processes. The memory requirement for one band is 645 bytes (3 + 130 + 256 + 256). The total memory requirement is 141 KB. In our scheme, the 16-bit data is sliced into one 2-bit long and two 7-bit long sub-words. Since the probability of a prediction error magnitude greater than 128 is under 2% in our experiments, the coding efficiency of these over-sized values is not critical. Each sub-word has a value of less than 128 and may share one entropy model to save memory. The new memory requirement is only 57KB. A flow-chart of the proposed entropy coding scheme is shown in Fig. 8.

![Fig. 8. Flowchart of the entropy coding.](image)

3. EXPERIMENTAL RESULTS

The hyperspectral images for the experiments are the 1997 Airborne Visible InfraRed
Imaging Spectrometer (AVIRIS) radiance images [1]. The images were taken from five areas: Cuprite, Jasper Ridge, Lunar Lake, Moffet Field, and Low Altitude. All images have a width of 614 pixels. The heights of the 5 sequences of hyperspectral image are 2206, 2586, 1431, 2031, and 3689 pixels, respectively. Each sequence contains 224 image bands and each pixel is represented by a 16-bit signed integer.

3.1 Parameter Settings

The parameter settings for the experiments are as follows: The first parameter is the context window size $M$. The selection range is from 8 to 21. Our initial simulation results indicate a 0.05 bpp (bits/pixel) performance difference between $M = 21$ and $M = 8$. A further increase in the context window size (beyond 21) does not show any meaningful improvements. Therefore, $M$ is set to 21 in our scheme. The second parameter is the maximum match count (MaxCount). According to the experimental results listed in Table 1, the count is set to 8. The third parameter is the line buffer size, which defines the search area in each previous band. This is basically a tradeoff between the memory size and the compression performance. A line buffer size of 5 (lines) was chosen. The performance degradation compared to the case using an unlimited buffer size is only 0.006 bpp. Finally, the fixed quantization approach is adopted instead of the on-line training approach due to its performance advantage, as explained in section 2.4.

3.2 Memory Requirement

Since the proposed BSI scheme aims to reduce the memory overhead of conventional LUT approaches, a comparison of memory requirements is provided. Table 2 summarizes the memory requirements for the LUT [23], the LAIS-LUT [24], the LAIS-QLUT [25], and the proposed M-QBSI schemes for BIL (band interleaved by line) and BSQ (band sequential) input data formats, respectively. Each table entry contains two items, one for the mathematical expression and one for the actual value. $N (= 614)$ and $B (= 224)$ indicate the width of the image and the number of bands, respectively. $L (= 5)$ represents the

<table>
<thead>
<tr>
<th>Table 2. Memory requirement for BIL and BSQ methods.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input type</strong></td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Input line buffer (pixels)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Reference band line buffer (pixels)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Look-up table (pixels)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Entropy model (bytes)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Total (bytes)</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
maximum number of lines searched in the BSI scheme. \( P (=16) \) is the pixel word length and \( Q_i \) is the quantization factor used in band \( i \). For a fair comparison, all schemes were assumed to have identical buffering and access strategies for data from the previous band. The BIL input format, which requires all 224 bands being processed concurrently line by line, demands a much larger memory requirement than the BSQ input format. The proposed scheme needs to buffer \( L \) lines of data in the previous band during the search. Since \( L \) is only 5, the incurred memory overhead is insignificant. The major difference in memory requirements is due to look-up tables. The LAIS-LUT scheme requires 2 look-up tables per band as opposed to 1 look-up table per band in the LUT scheme. Each table is indexed by a 16-bit pixel value and has 65,536 entries. The memory requirement is further magnified by the factor \( B \), leading to an impractically large memory size. Using the quantized index approach, the LAIS-QLUT scheme can effectively reduce the table memory size by a factor of 14. The reduced size is about 4MB. The proposed scheme, on the contrary, requires no look-up tables. The LUT, the LAIS-LUT, and the LAIS-QLUT schemes adopt the same entropy coding scheme. Four models (sign, threshold, MSB, and LSB) are needed, with corresponding model sizes of 3, 130, 256, and 256, respectively. 645 bytes of memory are required per band. In our scheme, three models (sign, magnitude, and subword) are used, with a total memory requirement per band of 261 \((3 + 130 + 128)\) bytes.

### 3.3 Compression Results

We compared the proposed scheme with various schemes published in the literature. These include JPEG_LS [26], JPEG_2000 [27], 3D-CALIC [10], SLSQ [15], M-CALIC [11], KSP [12], NPHI/EPHI [17], S-FPM/S-RLP [16], C-DPCM [22], CCAP (AAC) [19], LUT [23], LAIS-LUT [24], and LAIS-QLUT [25]. Two versions of the proposed scheme were employed. They are I-QBSI and M-QBSI, where “M”, “I”, and “Q” stand for multi-band prediction, inter-band prediction, and quantized index options, respectively. Both band-interleaved-by-line (BIL) and band-sequential (BSQ) input data formats were evaluated. The compression ratios corresponding to each input data format are summarized in Tables 3 and 4. The compression ratios of schemes other than the proposed one are the numbers reported in [19, 25]. Five sequences of hyperspectral images were used in the performance evaluation. Since the compression results of the Low Altitude sequences are not available for all schemes, they are not counted in calculating the average compression ratios. From the results in Tables 3 and 4, the two versions of the proposed scheme significantly outperform others in both BIL and BSQ formatted sequences. M-QBSI has the best performance since it employs both multi-band prediction and quantized index options. Notably, the compression ratio of the M-QBSI scheme on the Cuprite sequence is 4.01. The compression ratio enhancements range from 7% to 14% when compared with three other LUT-based schemes, *i.e.* LUT [23], LAIS-LUT [24], and LAIS-QLUT [25]. This is mainly attributed to two major factors. The first factor is the more accurate prediction scheme, which explores spatial data correlation and is considered statistically optimized by assuming that the data in the context window are stationary.

Fig. 9 shows the prediction efficiency for various schemes using the Cuprite sequence as an example. The prediction efficiency is evaluated as the entropy of the prediction residuals. In Fig. 9 (a), the proposed M-QBSI scheme has a performance edge over three other schemes (LUT, LAIS-LUT, and LAIS-QLUT-OPT). Fig. 9 (b) further highlights the
Table 3. Compression ratio results for BIL methods.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cuprite</th>
<th>Jasper Ridge</th>
<th>Lunnar Lake</th>
<th>Moffett Field</th>
<th>Low Altitude</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D-CALIC</td>
<td>2.97</td>
<td>2.98</td>
<td>3.01</td>
<td>3.17</td>
<td>–</td>
<td>3.04</td>
</tr>
<tr>
<td>SLSQ</td>
<td>3.15</td>
<td>3.15</td>
<td>3.15</td>
<td>3.14</td>
<td>2.98</td>
<td>3.15</td>
</tr>
<tr>
<td>M-CALIC</td>
<td>3.14</td>
<td>3.06</td>
<td>3.19</td>
<td>3.27</td>
<td>–</td>
<td>3.16</td>
</tr>
<tr>
<td>SLSQ-HEU</td>
<td>3.23</td>
<td>3.22</td>
<td>3.23</td>
<td>3.20</td>
<td>3.02</td>
<td>3.22</td>
</tr>
<tr>
<td>NPHI</td>
<td>3.34</td>
<td>3.27</td>
<td>3.22</td>
<td>3.34</td>
<td>–</td>
<td>3.29</td>
</tr>
<tr>
<td>LUT</td>
<td>3.44</td>
<td>3.23</td>
<td>3.40</td>
<td>3.17</td>
<td>3.09</td>
<td>3.31</td>
</tr>
<tr>
<td>S-RLP</td>
<td>3.41</td>
<td>3.44</td>
<td>3.41</td>
<td>3.43</td>
<td>–</td>
<td>3.43</td>
</tr>
<tr>
<td>S-FPM</td>
<td>3.43</td>
<td>3.46</td>
<td>3.43</td>
<td>3.46</td>
<td>–</td>
<td>3.45</td>
</tr>
<tr>
<td>LAIS-LUT</td>
<td>3.58</td>
<td>3.42</td>
<td>3.53</td>
<td>3.36</td>
<td>3.23</td>
<td>3.47</td>
</tr>
<tr>
<td>LAIS-QLUT-HEU</td>
<td>3.73</td>
<td>3.46</td>
<td>3.67</td>
<td>3.45</td>
<td>3.31</td>
<td>3.58</td>
</tr>
<tr>
<td>LAIS-QLUT-OPT</td>
<td>3.73</td>
<td>3.47</td>
<td>3.69</td>
<td>3.46</td>
<td>–</td>
<td>3.59</td>
</tr>
<tr>
<td>Proposed (I-QBSI)</td>
<td>3.95</td>
<td>3.64</td>
<td>3.90</td>
<td>3.62</td>
<td>3.44</td>
<td>3.78</td>
</tr>
<tr>
<td>Proposed (M-QBSI)</td>
<td>4.01</td>
<td>3.73</td>
<td>3.96</td>
<td>3.70</td>
<td>3.53</td>
<td>3.85</td>
</tr>
</tbody>
</table>

Table 4. Compression ratio results for BSQ methods.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cuprite</th>
<th>Jasper Ridge</th>
<th>Lunnar Lake</th>
<th>Moffett Field</th>
<th>Low Altitude</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diff. JPEG-LS</td>
<td>2.91</td>
<td>2.81</td>
<td>2.93</td>
<td>2.84</td>
<td>2.70</td>
<td>2.87</td>
</tr>
<tr>
<td>Diff. JPEG-2k</td>
<td>2.92</td>
<td>2.82</td>
<td>2.94</td>
<td>2.83</td>
<td>2.69</td>
<td>2.88</td>
</tr>
<tr>
<td>3D-CALIC</td>
<td>3.06</td>
<td>3.08</td>
<td>3.09</td>
<td>3.25</td>
<td>–</td>
<td>3.12</td>
</tr>
<tr>
<td>SLSQ-OPT</td>
<td>3.24</td>
<td>3.23</td>
<td>3.23</td>
<td>3.21</td>
<td>3.04</td>
<td>3.23</td>
</tr>
<tr>
<td>CCAP (AAC)</td>
<td>3.22</td>
<td>3.17</td>
<td>3.28</td>
<td>3.39</td>
<td>–</td>
<td>3.23</td>
</tr>
<tr>
<td>M-CALIC</td>
<td>3.22</td>
<td>3.17</td>
<td>3.28</td>
<td>3.39</td>
<td>–</td>
<td>3.26</td>
</tr>
<tr>
<td>KSP</td>
<td>3.28</td>
<td>3.23</td>
<td>3.27</td>
<td>3.25</td>
<td>–</td>
<td>3.26</td>
</tr>
<tr>
<td>LUT</td>
<td>3.44</td>
<td>3.23</td>
<td>3.40</td>
<td>3.17</td>
<td>3.09</td>
<td>3.31</td>
</tr>
<tr>
<td>C-HPHI</td>
<td>3.36</td>
<td>3.29</td>
<td>3.28</td>
<td>3.36</td>
<td>–</td>
<td>3.32</td>
</tr>
<tr>
<td>C-DPCM</td>
<td>3.42</td>
<td>3.46</td>
<td>3.37</td>
<td>3.46</td>
<td>–</td>
<td>3.43</td>
</tr>
<tr>
<td>S-RLP</td>
<td>3.41</td>
<td>3.44</td>
<td>3.41</td>
<td>3.43</td>
<td>–</td>
<td>3.43</td>
</tr>
<tr>
<td>S-FPM</td>
<td>3.43</td>
<td>3.46</td>
<td>3.43</td>
<td>3.46</td>
<td>–</td>
<td>3.45</td>
</tr>
<tr>
<td>LAIS-LUT</td>
<td>3.58</td>
<td>3.42</td>
<td>3.53</td>
<td>3.36</td>
<td>3.23</td>
<td>3.47</td>
</tr>
<tr>
<td>LAIS-QLUT-HEU</td>
<td>3.72</td>
<td>3.46</td>
<td>3.67</td>
<td>3.45</td>
<td>3.31</td>
<td>3.58</td>
</tr>
<tr>
<td>LAIS-QLUT-OPT</td>
<td>3.73</td>
<td>3.47</td>
<td>3.69</td>
<td>3.46</td>
<td>–</td>
<td>3.59</td>
</tr>
<tr>
<td>Proposed (I-QBSI)</td>
<td>3.95</td>
<td>3.64</td>
<td>3.90</td>
<td>3.62</td>
<td>3.44</td>
<td>3.78</td>
</tr>
<tr>
<td>Proposed (M-QBSI)</td>
<td>4.01</td>
<td>3.73</td>
<td>3.96</td>
<td>3.70</td>
<td>3.53</td>
<td>3.85</td>
</tr>
</tbody>
</table>

difference between the M-QBSI scheme and the runner up scheme LAIS-QLUT-OPT. The average difference is 0.3 bpp (bits/pixel). The second factor is the BSI scheme, which virtually increases the number of look-up tables to 8. For conventional LUT-based approaches, this would require a prohibitively large amount of memory. As for the computation time for the proposed schemes, the computing platform was a PC platform powered by an Intel Core2 Quad CPU Q6600 running at 2.40 GHz. The average computation times (over the
entire 224 bands on the indicated PC platform) for the two schemes are 145 seconds (I-QBSI) and 165 seconds (M-QBSI), respectively. These times can be further reduced by applying more code optimizations.

Based on the results on the Cuprite scene-2 sequence, Table 5 summarizes the contributions of the baseline compressor and each enhancement measure adopted in this paper on the overall compression performance and on the computation time. The baseline compressor means the combination of the simple inter/intra-band predictor, multi-LUT approach, and an adaptive arithmetic coder (AAC). Enhancement measure 1 advantage of the spatial data correlation and adaptively derives the prediction coefficients of the spectral domain predictor as a process of Weiner filtering. Enhancement measure 2 corresponds to the employment of an entropy model selection prior to the AAC stage. Enhancement measure 3 indicates the BSI scheme used in the second prediction stage. Enhancement measure 4 is the quantized index approach applied to the BSI scheme. From Table 5, the compression ratio (CR) enhancement by employing an entropy model selection mechanism is moderate and the induced computation overhead is minimal. For the Cuprite sequence, the BSI scheme alone does not lead to any significant compression ratio improvement. The combination of BSI plus quantized index measures not only improves the compression efficiency by almost 0.13 compression ratio but also reduces the computation time drastically.

4. CONCLUSION

We proposed an effective lossless compression scheme for hyperspectral images. This scheme improves the performance of conventional look-up table approaches by adopting
new prediction measures. Due to the properties of hyperspectral images, the prediction is performed in the spectral domain. One of the new measures takes the spatial data correlation into account and formulates the prediction as Wiener filtering. The Wiener filter is considered statistically optimal if input data are stationary. In our case, the data within the context window are highly spatially correlated and mostly stationary. We only explored the 1-tap (inter-band) and the 2-tap (multi-band) filtering schemes due to the constraint of computation complexity. The scheme, however, can be extended to multiple taps using any previous bands. A backward search scheme was adopted in place of look-up tables. This reduces the memory requirement and achieves performance equivalent to that obtained using multiple look-up tables. The incurred search effort is greatly reduced using the quantization index approach. Other module refinements, e.g., entropy coding, was also investigated. The proposed lossless compression schemes produce an average compression ratio of as high as 3.85, the best result among the compared schemes.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Prof. Jarno Mielikainen at Lappeenranta University of Technology for generously providing his source code. The authors would also like to thank anonymous reviewers for their valuable comments in revising this paper.

REFERENCES

9. L. Galli and S. Salzo, “Lossless hyperspectral compression using KLT,” in Proceed-


Cheng-Chen Lin (林振誠) was born in Taiwan, R.O.C., in 1981. He received the B.S. and M.S. degrees from the Department of Electronic Engineering, National Yunlin University of Science and Technology, Taiwan, in 2003 and 2005. He is currently working toward the Ph.D. degree at the Department of Electrical Engineering, National Chung Hsing University, Taiwan. His research interests include image and video compression, digital signal processing.

Yin-Tsung Hwang (黃穎聰) obtained his B.S. and M.S. degrees, both in Electronic Engineering, from National Chiao Tung University, Hsinchu, Taiwan, in 1983 and 1985, respectively. He received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Wisconsin, Madison in 1993. He then became a faculty member of the Department of Electronic Engineering, National Yunlin University of Science and Technology in Taiwan. In 2004, he joined the Department of Electrical Engineering, National Chung Hsing University and is now an Associate Professor. His research interests include VLSI designs for video compression and for wireless base band processing, automatic HW/SW interfacing in SoC integration, and low power VLSI circuit designs. Dr. Hwang is a member of the IEEE and IEICE.