Efficient Lossy Ultraspectral Data Compression: Hybrid Predictive Transform Coding Algorithm

Deqi Song

August 2013
Abstract

This project proposes a new lossy ultraspectral image compression algorithm, hybrid predictive transform (HPT) coding algorithm. It is based on the predictive coding and the transform coding algorithms. The HPT algorithm consists of four stages: preprocessing, prediction, discrete cosine transformation, and compression protocol. Since the attributes of the ultraspectral data change obviously in the spectral direction, the HPT employs the two predictors, that are used in the partition differential plus-code modulation (PDPCM) and low complexity lossless compression for two-dimensional (LOCO-2D) algorithms, to remove the correlation. When the threshold of spectral correlation is 0.9, the HPT algorithm divides the ultraspectral data into two groups—the data with a low spectral correlation and the data with a high spectral correlation—in the preprocessing stage. The data with a high spectral correlation uses the PDPCM predictor to remove spectral correlation, and then information is converted into transform domain using a two-dimensional discrete cosine transform (2D-DCT). Since 2D-DCT implements two separable cosine transforms in horizontal and vertical directions, it concentrates information on a number of low frequency coefficients. In order to increase the compression ratio, the HPT algorithm designs a compression protocol in each image to discard non-essential information. Each compression protocol, requiring 255 bits, is comprised of three components: removing parameter, essentiality sequence of index sets and corresponding quantization sequence. The parameter regulates the number of information discarded and the essentiality sequence evaluates the essentiality of each index set. Moreover, the HPT algorithm is an adaptive algorithm, where the compression protocol
is adapted to the error of the reconstructed image by using a threshold to control the error generated in compression process. In order to ignore the error accumulation generated in prediction process, the HPT adopts a switching filter to restart the compression model when the error is too large. The HPT algorithm provides an excellent compression performance in lossy compression, that achieves compression ratio between 9:1 and 35:1 with average SNR ranging from 25 dB to 32 dB.
Acknowledgement

First of all, I would like to thank Dr. Vinay K. Ingle, my thesis advisor, for his guidance and support throughout this project. I would like to thank Dr. Bahram Shafai and Dr. Hanoch Lev-Ari for being my thesis committee and spending time to read my thesis. Furthermore, I also want to thank Doctor Rolando Herrero for his help and guidance when I started the project. Finally, I want to thank my parents and friends for their support during the past two years of graduate study at Northeastern University.
Contents

Abstract

Acknowledgement

1 Introduction

1.1 Ultraspectral Imaging

1.2 AIRS Data

1.3 Ultraspectral Data Compression

1.3.1 Vector Quantization

1.3.2 Transform Coding

1.3.3 Predictive Coding

1.4 Summary

2 Review of Compression Techniques

2.1 3D-JPEG

2.1.1 Subcubes Construction

2.1.2 Level Shifting

2.1.3 Three-Dimensional Discrete Cosine Transform

2.1.4 Quantization

2.1.5 Source Encoder

2.1.6 Source Decoder
3 Hybrid Predictive Transform Coding Algorithm

3.1 Comparison of Predictors ............................................ 36
3.2 Preprocessing .......................................................... 39
3.3 Block Transform ....................................................... 41
3.4 Rate Allocation ........................................................ 42
3.5 Implementation of HPT Technique .................................. 47
3.6 Summary ............................................................... 49

4 Results and Conclusions

4.1 Initial data ............................................................. 51
4.2 Hybrid Predictive Transform Coding Technique .................. 52
4.3 3D-JPEG ............................................................... 58
4.4 Optimal 2D-DPCM .................................................... 60
4.5 PDPCM ................................................................. 62
4.6 Results ................................................................. 64
4.7 Conclusions .............................................................. 65

Appendices 67

A Parameters 67

B Optimal Coefficients 69

B.1 Optimal Coefficients in NSHP Model ....................... 69
B.2 Optimal Coefficients in PDPCM ......................... 70

Bibliography 72
List of Figures

1.1 Ultraspectral Images Structure ................................................. 11
1.2 Reflectance Curves ................................................................. 12
1.3 Original Images, bands 150, 350, 750, 1050, 1250, 1550, 1850, 2000 ....... 14
1.4 Spectral Correlations Between Adjacent Bands ............................... 15
1.5 Spatial Correlations in Band 400 ............................................... 16

2.1 Flow Diagram of 3D-JPEG ......................................................... 21
2.2 Zigzag Pattern ................................................................. 23
2.3 Flow Diagram of Quantization Matrix Designer ............................ 25
2.4 Predictive Coding ........................................................................ 27
2.5 Pixels Ordering for NSHP model .................................................. 29
2.6 Flow Diagram of PDPCM .......................................................... 32
2.7 Pixel distribution in the same band (k) and the previous band (k-1) as the pixel to be predicted .................................................. 33

3.1 Mean, Maximum, Minimum of Pixel Intensity in each band .................. 37
3.2 Spectral Correlations in Different Bands ........................................ 37
3.3 Four Predictors PDF of Predictive Error Band from 500 to 600 .............. 38
3.4 Four Predictors PDF of Predictive Error In The Bands from 1600 to 1700 .. 39
3.5 The Pre-processing .................................................................... 40
3.6 Pixel Intensity Distribution VS 1D Coefficient Sequence Distribution ...... 42
3.7 Reshape Coefficients and Generate Index Sets ........................................... 43
3.8 Two Types of Distribution of Index Set .................................................. 44
3.9 Compose Sequence of Index ....................................................................... 45
3.10 Form of Compression Protocol ................................................................. 46
3.11 Flow Diagram of Proposal Compression Method ........................................ 48

4.1 CR VS Average SNR ...................................................................................... 52
4.2 CR VS Average SNR ...................................................................................... 53
4.3 Original VS Reconstructed (LOCO-2D algorithm) Band = 1660, 1710 (Average SNR = 11.8226 dB)) ................................................................. 54
4.4 Original VS Reconstructed (CR = 31.08, Ave SNR = 23.4 dB RC = 0.32)
    Band = 310, 750, 1350, 1950 (SNR = 29.34 dB, 29.85 dB, 20.05 dB, 30.53 dB) 55
4.5 Original VS Reconstructed (CR = 25.8, Ave SNR = 26.9 dB, RC = 0.42)
    Band = 310, 750, 1350, 1950 (SNR = 30.23 dB, 31.54 dB, 23.72 dB, 33.59 dB) 56
4.6 Original VS Reconstructed (CR = 11.71, Ave SNR = 32.02 dB, RC = 0.59))
    Band = 310, 750, 1350, 1950 (SNR = 34.54 dB, 36.4dB, 24.32dB, 36.56dB) 57
4.7 CR VS Average SNR ...................................................................................... 58
4.8 Original and Reconstructed (Band = 310 and 1350) Average SNR 19.4387
    dB, 26.9936 dB, 37.0496 dB ........................................................................ 59
4.9 CR VS Average SNR ...................................................................................... 60
4.10 Original and Reconstructed (Band = 750 and 1350) Average SNR 22.14 dB,
    30.02 dB, 35.22 dB .................................................................................. 61
4.11 CR VS Average SNR ...................................................................................... 62
4.12 Original VS Reconstructed (Band = 500) Average SNR, 17.12 dB, 29.36 dB,
    SNR 36.06 dB .......................................................................................... 63
4.13 Compression Results .................................................................................. 64
Chapter 1

Introduction

In the past decades, the breakthrough of remote sensor promoted the development of ultraspectral image analysis. Ultraspectral images can be found in many fields, such as environmental monitoring, mineral exploration, material identification, and agriculture. The wide applications of ultraspectral images require the improvement of ultraspectral image compression. Recently, it is an active research project to reduce the amount of ultraspectral data needed to contain a given quantity of information. This thesis discusses a new lossy ultraspectral image compression technique.

This chapter consists of three sections: ultraspectral imaging, AIRS Data, and basic techniques used in ultraspectral image compression. Section 1.1 introduces the fundamental concepts of ultraspectral imaging. Section 1.2 presents one type of ultraspectral data used in this thesis. Section 1.3 reviews basic techniques used in ultraspectral image compression.

1.1 Ultraspectral Imaging

With the development of remote sensors, it is possible to collect a huge number of images in narrow adjacent spectral bands. The instruments producing multispectral images are complex remote sensors called spectrometers, that can produce images with a number of broad wavelength bands. Multispectral remote sensors collect images in dozens of spectral bands,
hyperspectral remote sensors collect images in hundreds of spectral bands, and ultraspectral remote sensors collect images in thousands of spectral bands.

Three-dimensional ultraspectral data contains thousands of two-dimensional band images, and each band image contains the radiating information generated by a remote sensor in a given wavelength. The structure of ultraspectral data is shown in Figure 1.1, where \( x(i, j, k) \) is the pixel value at \((i, j, k)\). Ultraspectral images contain a wealth of information.

![Figure 1.1: Ultraspectral Images Structure](image)

By studying the material properties in a continuous spectrum, engineers can interpret the information existing in the ultraspectral images. Recently, the breakthrough of complex sensors improves ultraspectral image analysis, that involves the development of two technologies: spectroscopy and remote imaging [2].

Spectroscopy studies the emitted or reflected light from different materials and its energy at different wavelengths. Since the energy at different wavelengths is reflected or absorbed in different ways, depending on which material properties each wavelength possesses [3], the spectral reflectance has different functions of wavelength in different ground materials. The energy at certain wavelength is reflected or absorbed to different degrees. Figure 1.2 shows spectral reflectance curves for different ground materials. The features of reflectance are
usually called absorption bands, and the strength and shape of absorption bands can be used to identify different materials.

![Figure 1.2: Reflectance Curves](image)

Some inorganic materials, such as minerals, crystalline structures and chemical composition, maintain the shape of the reflectance curve and the presence of absorption bands. For example, one special chemical element, the ion, can govern the shape of the spectral curve. Furthermore, green plants also can control the spectral reflectance curve, due to the absorption effects of leaf pigments (such as chlorophyll). Chlorophyll absorbs blue and red wavelengths more strongly than other visible light [1].

Remote sensors such as airborne or satellite ultraspectral sensors can be used to accurately measure the spectral reflectance. However, the sensor measures reflected light for an altitude of about 20 kilometers, so there are some factors affecting the measured values. In
this situation, surface materials cannot completely determine the properties of spectral reflectance. Many factors, that include atmospheric effects, sensor effects, illumination factors and input solar energy, also can influence the spectral reflectance [2].

In order to identify the material, the radiance value in the image needs to be converted into reflectance, that is used to compare with reference reflectance spectra. The effective conversion needs to account for several effects such as atmospheric transmission, solar energy, or sensor gain. Recently, there are some commonly used reflectance conversions, such as flat field conversion and average relative reflectance conversion. These conversions are based on different theories of statistics, and these conversions provide a mathematical analysis method to understand the information in the ultraspectral images.

In conclusion, ultraspectral imaging is an area of active research studying, and ultraspectral images are widely used in the fields of mineral mapping, and detection of soil properties, vegetation species, vegetation stress and military vehicles [1]. However, interpreting ultraspectral images requires the transmitted ultraspectral data contain a given amount of information. (In order to ensure transmission channel utilization, the transmitted data are compressed in the remote sensors.) With the wide applications of ultraspectral images, there is an active research development in ultraspectral image compression.

1.2 AIRS Data

The development of remote sensing applications promotes ultraspectral image analysis. There are two significant sensors developed by NASA, namely ultraspectral AVIRIS sensors and ultraspectral AIRS sensors. The three-dimensional data used in this thesis is obtained from AIRS sensors [4].

AIRS stands for Atmospheric Infrared Sounder, that is the standard dataset that is used in studying compression of ultra-spectral image data. It is comprised of thousands of band images generated in the infrared spectrum, that covers the infrared wavelengths from 4 µm to 15.4 µm. The AIRS data is widely applied by weather forecasting centers around the
world. Its major application is the acquisition of atmospheric parameters such as air quality, temperature, and clouds moisture [5].

The AIRS data obtained throughout the day is divided into 240 granules in such a way that a single granule is captured every 6 minutes. Granules consist of 135 scan lines with 90 pixels per scan line over 2,378 bands, so there are a total of $135 \times 9 \times 2378 = 28,892,700$ pixels per granule. Notice that there are 347 bands normally ignored because they are incompatible with other data. AIRS normally distributes 12-bits or 14-bits per pixel data (This study uses 12-bits per pixel data) [5]. Figure 1.3 points out some representative original bands.

![Figure 1.3: Original Images, bands 150, 350, 750, 1050, 1250, 1550, 1850, 2000](image-url)
Furthermore, in the AIRS standard, the near-allocated bands always have a very high correlation. The ultraspectral images have two types of correlations: spatial correlation between neighboring pixels in the same band and the spectral correlation between adjacent bands. Since the AIRS data is operated by the continuous spectrum near the infrared region, it always has a high spectral correlation. Figure 1.4 points out the spectral correlation between adjacent bands. The AIRS data roughly keeps a high spectral correlation except a few special bands, where the spectral correlation rapidly decreases.

![Figure 1.4: Spectral Correlations Between Adjacent Bands](image)

The AIRS data also has a high spatial correlation. For example, Figure 1.5 shows the spatial correlation in the band 400. This thesis exploits the high correlation for efficient compression. As the above discussion, most of the bands have a high spectral correlation, so this thesis generally uses the spectral correlation for compression. However, when the spectral correlation is obviously lower than spatial correlation, the spatial correlation is exploited to effectively compress data.
1.3 Ultraspectral Data Compression

Based on whether to reconstruct images with all of the compressed information, ultraspectral compression techniques can be classified into two types: lossless compression and lossy compression. Lossless compression, also called error-free coding, uses all of the compressed information to regenerate images, and its reconstructed images are exactly same as the original data. Lossless compression has zero error in the compression and decompression process. On the contrary, lossy compression discards non-essential information, and regenerates images using partial coefficients. This results in a loss of information but when done properly, the regenerated image has an excellent fidelity. This thesis discusses the lossy ultraspectral image compression technique.

Lossy ultraspectral image compression technique generally consists of three components: decorrelation, quantization, and source encoding. The lossy compression includes decorrelation and quantization. There are three fundamental compression techniques applied in lossy ultraspectral image compression. These are vector quantization, transform coding, and predictive coding.
1.3.1 Vector Quantization

Vector quantization (VQ) is used in the compression stage, and most quantization techniques are not fully reversible. It mainly consists of four stages: vector formation, set generation, codebook generation, and quantization. It treats a set of scalars as vectors, determining the best quantization vector based on which one can best minimize the average distortion between the input and output in the sense of mean squared error (MSE) [3]. Therefore, vector quantization is more efficient than scalar quantization, but more difficult to implement.

Vector quantization was initially used to solve the problem due to the need of multidimensional integration. In the beginning of 1970s, Lindo, Buzo, and Gray provided a VQ algorithm called Lindo-Buzo-Grey (LBG) algorithm [17]. It used a training sequence to bypass the need for multidimensional integration. Then, researchers applied splitting codebook in the LBG algorithm to reduce the quantizer search complexity by replacing full research sequence with a sequence of tree decisions. The idea of cluster was proposed to design VQ algorithms in 1980s [16].

1.3.2 Transform Coding

Ultraspectral compression mainly uses three types of transform coding algorithms, namely, principal component analysis (PCA), discrete cosine transform (DCT), and discrete wavelet transform (DWT). The PCA is the most effective transform to concentrate energy on a small number of coefficients. It is widely used in ultraspectral data decorrelation. When all principal components are retained, PCA is commonly known as the Karhunen-Loève transform (KLT), where the ultraspectral data with N spectral bands produces an \( N \times N \) unitary KLT transform matrix [9]. KLT performs an excellent decorrelation in practice. However, because a fast algorithm of KLT is not possible, it is not widely used in ultraspectral image compression.

The discrete cosine transform is a block transform, that implements separable 1D cosine transforms in horizontal, vertical, and spectral directions. It has a capability to concentrate
energy on the low frequency coefficients for correlated data. A still image compression algorithm in Joint Photographic Expert Group (JPEG) is based on this block transform. Furthermore, because the transform direction does not take image orientation features, a directional DCT, whose transform matrix is dependent on directional angle and interpolation, has been provided to more effectively concentrate energy on a small number of coefficients in recent years[10].

The discrete wavelet transform is a subband filter, that implements separable 1D wavelet transforms in horizontal, vertical, and spectral directions. It is mainly used in the Joint Photographic Expert Group 2000 (JPEG2000), Set Partitioning in Hierarchical Trees (SPIHT), and Set Partitioned Embedded bloCK (SPECK) algorithms.

### 1.3.3 Predictive Coding

Predictive coding is to reduce the amount of transmitted data required to contain a given quality of information. By eliminating redundancies in the neighboring pixels in the same or adjacent bands, the predictive algorithm accomplishes the initial decorrelation. The results are quantized and then encoded with a known codebook. By whether or not to quantize the information, predictive coding can be divided into lossless or lossy predictive coding. Lossless predictive coding is used in lossless compression, that is preformed with two steps: decorrelation and source encoding. On the other hand, the lossy predictive coding quantizes the information before encoding, and the quantization step generates irreversible error. By the characteristics of the predictive function, the predictor can be classified into the linear predictor and the non-linear predictor. The linear predictor uses a linear combination of the previous pixel values to estimate the current pixel value. The non-linear predictor always follows a known context or a nonlinear equation to predict the current pixel value.

The vector quantization is applied to the lossy compression to solve the problem in the coding of multispectral imagery. The transform coding uses one type of transform (eg. discrete cosine transform, discrete wavelet transform, etc) to convert data into transform
domain. The algorithm concentrates information on a few transform coefficients. The predictive coding algorithm eliminates correlation using a predictive combination function of the previous pixel values. By the features of prediction (eg. characteristics of predictive function, components) the technique can be classified into different types. Lossy compression is comprised of decorrelation and quantization. Transform coding and predictive coding algorithms are usually used in decorrelation step, and vector quantization is sometimes exploited in quantization step.

1.4 Summary

This chapter introduced active development and wide applications of ultraspectral imaging, and specifically presented one type of the ultraspectral data generated by AIRS remote sensors. It also provided a quick summary of basic techniques in ultraspectral image compression.

The following chapters are organized as follows: Chapter 2 reviews some basic techniques used in ultraspectral image compression. Chapter 3 presents the methodology and procedure of the hybrid predictive transform (HPT) coding technique. Chapter 4 compares the compression results of the HPT method with the other three compression methods (presented in Chapter 2).
Lossy image compression is a process to reduce the amount of data required to convey a given amount of information. In this process, the data that represents repeated or non-essential information is discarded. To remove the data representing repeated or non-essential information, lossy compression technique usually employs the transform coding and predictive coding before quantization.

This chapter reviews some lossy compression algorithms used in ultraspectral compression. 3D-JPEG adopts a discrete cosine transform to concentrate information on the low frequency coefficients, and then reduces the quantity of coefficients by discarding the non-essential ones. Optimal 2D-DPCM and PDPCM use a linear combination of previous pixels to remove the spatial or spectral redundancies. LOCO-2D, and LOCO-3D algorithms follow a known context mapper to remove spatial or spectral redundancies. Furthermore, these compression algorithms are the basis of the HPT technique in Chapter 3.

2.1 3D-JPEG

Joint Photographic Experts Group (JPEG) developed a lossy compression algorithm that is based on discrete cosine transform (DCT). It is comprised of two components: compression and decompression. The compression part consists of six stages: data preprocessing, block
generation, DCT, level shifting, quantization, and source encoder. The decompression part is an inverse process, that is comprised of five stages: source decoder, de-quantization, inverse DCT, level shifting, and image reconstruction. JPEG is usually used to compress two-dimensional or three-dimensional still images. Their fundamental theory is same. This section presents the fundamental theory and procedure of the 3D-JPEG. Figure 2.1 shows the flow diagram of the 3D-JPEG compression.

2.1.1 Subcubes Construction

In order to divide the ultraspectral images into a sequence of subcubes $x_s$, the size of original data needs to be adjusted. All subcubes have the same size of $N \times N \times N$. When the size of the data cannot be all divisible by $N$ separately, it is expanded with zero padding.

2.1.2 Level Shifting

Each subcube’s $N^3$ pixels is level-shifted by subtracting the mean across the N bands. Then the new subcubes $\bar{x}_s$ are calculated. (The mean is also used in the step of inverse level shifting in the decompression process.)
2.1.3 Three-Dimensional Discrete Cosine Transform

Each subcube is converted into transform domain using a three-dimensional discrete cosine transform (3D-DCT), and the transform coefficients are calculated. An \(N \times N \times N\) three-dimensional discrete cosine transform is defined as:

\[
C(t_1, t_2, t_3) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \bar{x}_s(i, j, k) \cos \left( \frac{\pi}{2N} t_1 (2i + 1) \right) \cdot \cos \left( \frac{\pi}{2N} t_2 (2j + 1) \right) \cdot \cos \left( \frac{\pi}{2N} t_3 (2k + 1) \right)
\]

where \(0 \leq t_1, t_2, t_3 < N\), \(\bar{x}_s(i, j, k)\) is the pixel value at the point of \((i, j, k)\) in each subcube, and \(C(t_1, t_2, t_3)\) is the transform coefficient at \((t_1, t_2, t_3)\) in each subcube. The 3D discrete cosine transform implements three separable 1D cosine transforms in horizontal, vertical and spectral directions. Therefore, compared with the 2D-DCT, the 3D-DCT has a better performance in concentrating energy on the low frequency coefficients.

2.1.4 Quantization

The quantization is obtained by dividing each subcubes coefficient matrix \(C\) by a known quantization matrix \(Q\). In order to decrease the bits required to store per coefficient in the quotient matrix, the quotients are rounded. The quantization step is expressed as:

\[
C_q(t_1, t_2, t_3) = \text{round} \left( \frac{C(t_1, t_2, t_3)}{Q(t_1, t_2, t_3)} \right), 1 \leq i, j, k \leq N
\]

where \(0 \leq t_1, t_2, t_3 < N\), \(C(t_1, t_2, t_3)\) is the transform coefficient at \((t_1, t_2, t_3)\), and \(Q(t_1, t_2, t_3)\) is quantization coefficient at \((t_1, t_2, t_3)\). Notice that the compression ratio is determined by the number of coefficients discarded, and the quantization matrix directly affects the number of discarded coefficients.

2.1.5 Source Encoder

The rounded quotient matrix contains a large number of zero-valued coefficients. In order to decrease the number of coefficients required to store, a zigzag pattern is used to reshape the rounded quotient subcubes \(C_q\). The size of each quotient subcube is \(N \times N \times N\), and each layer
of the subcube is reordered with a zigzag pattern one by one (shown in Fig 2.2). Then each subcube is represented by $N$ unidimensional sequences. Using a special symbol to express the remainder of coefficients with zero-value in each unidimensional sequence, parts of zero-valued coefficients can be discarded. The $N$ unidimensional sequences is combined into one unidimensional sequence, and the result is encoded by a known variable-length coder, such as a template Huffman code.

![Figure 2.2: Zigzag Pattern](image)

2.1.6 Source Decoder

The source decoder is an inverse source encoder process. According to the previously known codebook (template Huffman codebook), the encoded data is decoded, and the result is an unidimensional sequence. The unidimensional sequence separates in each special symbol, and hence the unidimensional sequence can be divided into a sequence of unidimensional sequences. Reshaping each unidimensional sequence with an inverse zigzag pattern, it is reordered into an $N \times N$ subimage. Every $N$ subimages construct an $N \times N \times N$ subcube $C_q$. 

23
2.1.7 De-quantization

The de-quantization is an inverse quantization process to multiply each reconstructed sub-cube \( C_q \) by a previously known quantization matrix \( Q \). The inverse process is expressed as

\[
C'(t_1, t_2, t_3) = C_q(t_1, t_2, t_3)Q(t_1, t_2, t_3)
\]  

(2.3)

where \( 1 \leq t_1, t_2, t_3 \leq N \), and \( C'(t_1, t_2, t_3) \) is the de-quantized coefficient at \((t_1, t_2, t_3)\).

2.1.8 Inverse Three-Dimensional Discrete Cosine Transform

Each de-quantized coefficient subcube \( C' \) is inversely converted into original domain using an inverse three-dimensional discrete cosine transform (inverse 3D-DCT). An \( N \times N \times N \) inverse three-dimensional discrete cosine transform is defined as:

\[
\bar{x}_s'(i, j, k) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \frac{1}{N^3} w_1(t_1)w_2(t_2)w_3(t_3)C''(t_1, t_2, t_3)
\]

\[
\times \cos \left( \frac{\pi}{2N} t_1 (2i + 1) \right) \cos \left( \frac{\pi}{2N} t_2 (2j + 1) \right) \cos \left( \frac{\pi}{2N} t_3 (2k + 1) \right)
\]

(2.4)

where \( 0 \leq i, j, k < N \), and

\[
w_1(t_1) = \begin{cases} 
0.5, & t_1 = 0 \\
1, & 1 \leq t_1 < N 
\end{cases}
\]

(2.5)

\[
w_2(t_2) = \begin{cases} 
0.5, & t_2 = 0 \\
1, & 1 \leq t_2 < N 
\end{cases}
\]

\[
w_3(t_3) = \begin{cases} 
0.5, & t_3 = 0 \\
1, & 1 \leq t_3 < N 
\end{cases}
\]

Each reconstructed pixel subcube \( \bar{x}_s' \) is calculated. (Notice that there is no error generated in the processing of inverse 3D-DCT.)
2.1.9 Level Shifting

Each subcube’s $N^3$ pixels is level-shifted by adding the mean across the $N$ bands. The new subcubes $\bar{x}_s'$ are calculated. Notice that the mean is the one stored in the compression process.

2.1.10 Data Reconstruction

Every $N$ bands are reconstructed by combining a number of subcubes. When the size of reconstructed data is bigger than original data, we remove the padding, that is used to expand the original data in the step of subcube construction.

2.1.11 Rate Allocation

The design of quantization matrix affects the compression performance of the 3D-JPEG. The aim of the quantization is to allocate more bits to store the essential information and fewer bits to store the non-essential information. Therefore, the quantization matrix distributes higher coefficient value into the non-essential points and lower coefficient value in the essential points into the quantization matrix. Figure 2.3 shows its overall flow diagram.

![Figure 2.3: Flow Diagram of Quantization Matrix Designer](image)

The algorithm consists of six steps: expansion, block transformation, level shifting, DCT, essentiality calculation, and redistribution. The size of the input data is adjusted to ensure it can be all divisible by $N$, and then we construct a sequence of subcubes $x_s$ (the size is set at $N\times N\times N$). All pixels in each subcube is level-shifted by subtracting the mean of each subcube. A three-dimensional discrete cosine transform is used to convert each subcube’s pixels into transform domain, and then a number of transform coefficient cubes $C$ are computed. The essentiality calculation is to compute the signification of each point in the quantization cube.
Since one point’s essentiality depends on the probability of high coefficients appearing in the point, the essentiality distribution matrix $M$ is proportional to the mean of the absolute value of all transform coefficients cubes. The essentiality matrix is defined as:

$$M(t_1, t_2, t_3) = \frac{\sum_{n=0}^{P-1} |C(t_1, t_2, t_3)|}{P}$$

(2.6)

where $0 \leq t_1, t_2, t_3 < N$, $C(t_1, t_2, t_3)$ is the transform coefficients (calculated with equation 2.1) at the point of $(t_1, t_2, t_3)$, and $P$ is the number of blocks. In the matrix $M$, the higher coefficient value means the higher essentiality in the point. Furthermore, because the aims of quantization matrix are to remove the non-essential information and to allocate more bits to store the essential information, we reorder the matrix $M$ by distributing lower coefficient value into the points with higher coefficient value. Then a quantization matrix is designed.

### 2.2 Predictive Coding Algorithms

Predictive coding consists of compression and decompression. The compression part includes three components: predictor, quantizer, and source encoder. The decompression part is composed of two components: predictor and source decoder (shown in Fig 2.4). A discrete sequence of samples $\{x(n), n = 0, 1, 2, \ldots\}$ are inputted to the compression part. Through the predictive function, the current pixel $x(n)$ is predicted from the previous pixel values. The predictive expression is given by

$$\hat{x}(n) = f(x(n - 1), x(n - 2), \ldots)$$

(2.7)

where the $\hat{x}(n)$ is the predictive value of $x(n)$, and the $f(\cdot)$ is predictive function. The $W$ and $H$ are the height and width of the input image. The predictive error is the difference between the predictive pixel value and the current pixel value, that is given by

$$e(n) = x(n) - \hat{x}(n)$$

(2.8)

In the next step, the predictive error is quantized. The quantization function is given by

$$e_q(n) = Q(e(n))$$

(2.9)
where $Q(\cdot)$ is the quantization function and $e_q(n)$ is the quantized predictive error. The quantization is an irreversible process that generates error. In fact, the predictive process does not generate error. The error is generated in the quantization process. The source encoder is to encode the quantized predictive error using a known codebook. The encoded information is stored and transmitted in the channel.

![Diagram of Predictive Coding](image)

**Figure 2.4:** Predictive Coding

In the decompression, the predictive error $e_q$ is obtained by decoding received information using a previously known codebook. The reconstructed pixel is the sum of predictive pixel value and quantized predictive error. The reconstructed expression is given by

$$\tilde{x}(n) = \hat{x}(n) + e_q(n)$$  \hspace{1cm} (2.10)

where $\tilde{x}(n)$ is the reconstructed pixel. The predictive pixel depends on the previously reconstructed pixel values

$$\hat{x}(n) = f(\tilde{x}(n-1), \tilde{x}(n-2), ...)$$  \hspace{1cm} (2.11)
The reconstructed error is totally generated in the quantization process. The prediction usually can be classified into linear prediction and non-linear prediction. When the predictive function \( f(\cdot) \) is a linear combination of previous pixels, the prediction is linear. When the predictive function \( f(\cdot) \) cannot be expressed as a linear combination, the prediction is non-linear.

### 2.3 Linear Predictive Coding Algorithms

Similar with the general predictive coding algorithm described in Section 2.2, linear prediction consists of compression and decompression. The compression part is comprised of three steps: linear predictor, quantizer, and source encoder. Decompression part includes source decoder and linear predictor. Especially, the predictive function of the linear prediction can be represented by a linear combination of previous pixel values.

#### 2.3.1 Optimal 2D-DPCM

Optimal two-dimensional differential pulse-code modulation (optimal 2D-DPCM) is a simple linear predictor. It uses non-symmetric half plane (NSHP) model to estimate the current pixel (shown in Fig 2.5). In the optimal 2D-DPCM, the predictive expression can be written as a weighted sum of the neighboring pixel values [4].

\[
\hat{x}(i, j) = \alpha_1 x(i, j - 1) + \alpha_2 x(i - 1, j) + \alpha_3 x(i - 1, j - 1) + \alpha_4 x(i - 1, j + 1) \tag{2.12}
\]

where \( 1 \leq i \leq H - 1, 1 \leq j \leq W - 1 \), and \( x(i, j) \) is the pixel value at \((i, j)\). \( \alpha_1, \alpha_2, \alpha_3 \) and \( \alpha_4 \) are optimal coefficients. The mean square error (MSE) is given by

\[
E\{e^2\} = E \left\{ \left( x(i, j) - \alpha_1 x(i, j - 1) - \alpha_2 x(i - 1, j) \right)^2 \right\} - \left( \alpha_3 x(i - 1, j - 1) - \alpha_4 x(i - 1, j + 1) \right) \tag{2.13}
\]

Since the predictive pixel value is the multiple of previous pixel values and the error is the difference between original pixel values and predictive pixel values, the error and previous pixel are orthogonal (see Appendix B.1). The optimal coefficients are calculated in the case
of minimizing the mean of square error (MSE). That is

\[
\begin{bmatrix}
    r_x(0, 0) & r_x(1, 1) & r_x(1, 0) & r_x(1, -2) \\
    r_x(1, 1) & r_x(0, 0) & r_x(0, 1) & r_x(0, -1) \\
    r_x(1, 0) & r_x(0, 1) & r_x(0, 0) & r_x(0, 2) \\
    r_x(1, 2) & r_x(0, -1) & r_x(0, 2) & r_x(0, 0)
\end{bmatrix}
\begin{bmatrix}
    \alpha_1 \\
    \alpha_2 \\
    \alpha_3 \\
    \alpha_4
\end{bmatrix}
= \begin{bmatrix}
    r_x(0, 1) \\
    r_x(1, 0) \\
    r_x(1, 1) \\
    r_x(1, -1)
\end{bmatrix}
\] (2.14)

where

\[r_x(m, n) = E\{x(i, j)x(j - m, j - n)\}, 0 \leq m \leq 1, 0 \leq n \leq 2\] (2.15)

is a covariance function of image. Then the predictor model is established. The predictive error (calculated with equation (2.8)) is quantized by a quantization function \(Q(\cdot)\), where \(Q(\cdot)\) can be an uniform or non-uniform quantization sequence. The quantized error is encoded with a known codebook (e.g., Template Huffman Code, Run-Length Code, Golomb Code, etc.). Moreover, when the input image has a high spatial correlation, the distribution of the predictive error is highly concentrated, because high spatial correlation means that more redundant information can be discarded in the prediction process. Therefore, NSHP model has a better compression performance on the data with a high spatial correlation.

![Figure 2.5: Pixels Ordering for NSHP model](image)

**2.3.2 PDPCM**

Unlike Optimal 2D-DPCM, the partition differential pluse-code modulation (PDPCM) is based on the first order DPCM that estimates the predictive pixel values in the current band
by the pixel values in the previous band. Let the predictive expression be expressed as

\[ \hat{x}(i, j, k) = \alpha_1(k)x(i, j, k - 1) + \alpha_0(k) \]  

(2.16)

where \( 0 \leq i \leq H - 1, 0 \leq j \leq W - 1, 2 \leq k \leq B \), and the sample \( x(i, j, k) \) represents the pixel at \((i, j)\) in the band \( k \). \( \alpha_0 \) and \( \alpha_1 \) are optimal coefficients. Its mean square error (MSE) is given by

\[ E\{e_k^2\} = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (x(i, j, k) - \alpha_1(k)x(i, j, k - 1) - \alpha_0(k))^2 \]  

(2.17)

When the mean square error is minimum, the optimal coefficients can be calculated by the following equations (see Appendix B.2)

\[ \alpha_0(k) = \left(1 - \frac{r_k(1) - \mu_k^2}{r_k(0) - \mu_k^2}\right) \mu_k \]  

(2.18)

\[ \alpha_1 = \frac{r_k(1) - \mu_k^2}{r_k(0) - \mu_k^2} \]  

(2.19)

where \( r_k \) and \( \mu_k \) are

\[ r_k(t) = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x(i, j, k)x(i, j, k - t) \]  

(2.20)

\[ \mu_k = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x(i, j, k) \]  

(2.21)

where \( \mu_k \) is the mean of band \( k \), \( r_k \) is the covariance between the band \( k - t \) and the band \( k \), and \( t \) is the lag in the spectral dimension. Moreover, in order to reduce the number of optimal coefficients in the predictive function, the PDPCM normalizes each bands pixels. That is

\[ x'(i, j, k) = (x(i, j, k) - \mu_k) / \sigma_k \]  

(2.22)

where \( \mu_k \) and \( \sigma_k \) are the mean and variance in the band \( k \). The normal images’ \( r'_k(0) \) and \( \mu'_k \) are given by

\[ r'_k(0) = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x'(i, j, k)^2 = 1 \]  

(2.23)
\[ \mu'_k = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x'(i, j, k) = 0 \] (2.24)

Through the equations (2.18) and (2.19), the optimal coefficients are calculated. They are

\[ \alpha'_0(k) = 0 \] (2.25)

\[ \alpha'_1(k) = r'_k(1) \] (2.26)

In this situation, the predictive equation is given by

\[ \hat{x}(i, j, k) = \alpha'_1(k)x'(i, j, k - 1) = r'_k(1)x'(i, j, k - 1) \] (2.27)

where

\[ r'_k(1) = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x'(i, j, k)x'(i, j, k - 1) \] (2.28)

is the covariance between the band \( k - 1 \) and the band \( k \). Furthermore, the spectral correlation \( \text{Cor}(k) \) between adjacent images is defined as

\[ \text{Cor}(k) = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x'(i, j, k)x'(i, j, k - 1) = r'_k(1) \] (2.29)

where the \( H \) and \( W \) is the size of each band. When the spectral correlation is low, the distribution of the predictive error is not highly concentrated, because the fewer spectral redundancies are discarded. The PDPCM has a better compression performance on the data with a high spectral correlation. In order to keep a good compression performance, the PDPCM employs a threshold \( T_c \) to limit the spectral correlation between adjacent bands. Using a threshold \( (T_c) \), the PDPCM separates the whole bands into a series of partitioning data and employs the optimal DPCM predictor to compress each partitioning data. Threshold \( (T_c) \) is compared with the spectral correlation \( (\text{Cor}(k)) \). If \( T_c > \text{Cor}(k) \), the present predictive model is stopped, the next initial image is inputted, and the process is restarted. If \( T_c < \text{Cor}(k) \), the adjacent band is predicted with the present predictive model.

The flow diagram of PDPCM is showed in Figure 2.6. The input images are normalized to get the normal images are \( x'_1 \) and \( x'_2 \). Using the equation (2.28), we compute the optimal
coefficient. Then we can estimate predictive image $\hat{x}_2'$ using the above mentioned PDPCM algorithm. The predictive error $e$ is calculated by subtracting the predictive image $\hat{x}_2'$ from the normal image $x_2$. We quantize the predictive error $e$, and the result $e_q$ is encoded by a template Huffman code. Finally, the reconstruct the image $\hat{x}_2$ is the sum of the quantized error predictive $e_q$ and predictive normal image $\hat{x}_2'$.

### 2.4 Nonlinear Predictive Coding Algorithms

When the predictive function cannot be represented by a linear combination of previous pixel values, the predictive algorithm is viewed as a non-linear prediction. The non-linear predictor usually estimates the current pixel value using a known context mapper. Similar with the general predictive coding algorithm, in the non-linear prediction compression also consists of three stages: context predictor, quantization, and source encoder (shown in Figure 2.4).
2.4.1 LOCO Algorithms

The LOCO-2D algorithm, stands for Low Complexity Lossless Compression for two-dimensional images, is a low-complex context predictor [11]. The predictive context is given by

\[
\hat{x}(i, j, k) = \begin{cases} 
\min(x(i - 1, j, k), x(i, j - 1, k)), & x(i - 1, j - 1, k) \geq \\
\max(x(i - 1, j, k), x(i, j - 1, k)), & x(i - 1, j - 1, k) \geq \\
x(i - 1, j, k) + x(i, j - 1, k) - x(i - 1, j - 1, k), & \text{otherwise}
\end{cases}
\]

where \(0 \leq i \leq H - 1, 0 \leq j \leq W - 1, 1 \leq k \leq B\), and the \(x(i, j, k)\) is the pixel value at \((i, j)\) in the band \(k\). The \(H, W,\) and \(B\) are the size of the input data. The pixel distribution is shown in Figure 2.7. The context mapper is also called median predictor.

![Pixel distribution in the same band (k) and the previous band (k-1) as the pixel to be predicted.](#)

Furthermore, based on the estimation that adjacent band’s differences between two pixel points should be similar, LOCO-3D algorithm assumes that \(x(i, j, k) - x(i - 1, j, k) \approx x(i, j, k - 1) - x(i - 1, j, k - 1)\). Therefore, the predictive context of LOCO-3D algorithm
can be given by

\[
\hat{x}(i, j, k) = \begin{cases} 
\min(x(i - 1, j, k), x(i, j - 1, k)), & x(i - 1, j - 1, k) \geq \\
\max(x(i - 1, j, k), x(i, j - 1, k)), & x(i - 1, j - 1, k) > \\
x(i - 1, j, k) - x(i, j, k - 1) + x(i - 1, j, k - 1), & \text{otherwise}
\end{cases}
\]

(2.31)

where \(0 \leq i \leq H - 1, 0 \leq j \leq W - 1, \) and \(2 \leq k \leq B\). The new context mapper does not only consider the spatial correlation among the neighboring pixels, but also considers the spectral correlation between the adjacent bands.

Furthermore, in the quantizer, the predictive error, generated by above context predictors, is quantized by a known quantization function \(Q(\cdot)\). The result is encoded with a variable-length code, such as a template Huffman code.

### 2.5 Summary

Image compression is the process to reduce the amount of data required to reconstruct original images. There are many different techniques applied in compress ultraspectral image. This chapter presented five compression algorithms, namely, 3D-JPEG, optimal 2D-DPCM, PDPCM, LOCO-2D, and LOCO-3D algorithms. Based on the fundamental theory of the above compression techniques, Chapter 3 provides a new lossy compression technique, hybrid predictive transform coding technique.
Chapter 3

Hybrid Predictive Transform Coding Algorithm

This chapter presents a new ultraspectral image compression technique, namely hybrid predictive transform (HPT) coding technique. It is based on the predictive coding and the transform coding. It consists of four stages: preprocessing, prediction, block transformation, and compression protocol. In preprocessing step, the ultraspectral data is divided into two groups: a group of data with a high spectral correlation and a group of data with a low spectral correlation. The data with a low spectral correlation uses the LOCO-2D algorithm to remove spatial redundancies. The data with a high spectral correlation uses the PDPCM (non-quantization) to remove the spectral redundancies, and then a discrete cosine transform is used to convert the information into transform domain. The HPT method increases the compression ratio by discarding part of transform coefficients. A compression protocol is provided to regulate the compression ratio. Moreover, the HPT method is also an adaptive compression method, that uses a threshold to adjust the amount of removed information. To control the error accumulation, a switching filter is used to restart the model when error is too large.

Section 3.1 compares the compression performance of four different predictors (described
in Section 2.3 and Section 2.4), explaining why we need to divide ultraspectral data and compress them separately. Section 3.2 presents the preprocessing step. Section 3.3 explains why a discrete cosine transform is adopted to discard the non-essential information. Section 3.4 presents the design of compression protocol. Section 3.5 presents the implementation of the HPT method.

3.1 Comparison of Predictors

Predictor can be classified into two types: linear predictor or nonlinear predictor. Section 2.3 presented two linear predictors: optimal 2D-DPCM and PDPCM. The optimal 2D-DPCM estimates predictive pixel values with the neighboring pixel values in the same band, that removes the data’s spatial redundancies. The PDPCM estimates predictive pixel values in the current band with the whole pixel values in the previous band. As discussed in Section 2.3, when the data has a high spectral correlation, the PDPCM provides a better compression performance, due to that a high spectral correlation means more spectral redundancies can be discarded. Then its distribution of the predictive error is highly concentrated. Section 2.4 indicated two non-linear predictors: LOCO-2D context predictor and LOCO-3D context predictor. They are all context predictors, that employ a known context book to predict the current pixel value. However, the LOCO-2D algorithm uses the neighboring pixel values in the same band to estimate predictive pixel values, and the LOCO-3D algorithm uses the neighboring pixel values in the same and previous bands to estimate the current pixel values.

The AIRS data, used in this thesis, contains 2017 bands. Its attributes, such as spectral correlation, intensity range and mean, change in the spectral direction. For the data with different attributes, the compression preformation of predictors varies widely. In order to effectively remove redundancies, the AIRS data is divided into a series of data packets. The spectral correlation between adjacent bands can be used to divide the original data into different groups. Section 1.2 described that the ultraspectral data generally has a high spectral correlation. Figure 1.3 shows that the AIRS data roughly keeps a high spectral
correlation except on a few special bands. Furthermore, sometimes intensity range also can be used to separate the original data. Figure 3.1 points out that the intensity of each band generally decreases in the higher bands. In HPT method, the AIRS data is divided into two groups: one group of data with a high spectral correlation and higher pixel intensity and another group of data with a low spectral correlation and low pixel intensity.

Comparing the compression efficiency of the previous four predictors on the data with a
high spectral correlation, PDPCM has the best performance. For example, the bands from 500 to 600 have a high spectral correlation (see in Fig 3.2 (a)). Their spectral correlation are mostly higher than 0.94, and their pixel intensity ranges from −20 to 180 (see in Fig 3.1). In this situation, the previous four predictors all provide a good compression performance, but the PDPCM has the best compression performance. Its probability density function of predictive error distribution has the narrowest range, ranging from −1.5 to 1.5. Figure 3.3 shows the probability density function of predictive error generated by the four predictors. Therefore, for the data with a high spectral correlation, PDPCM has a better compression performance.

![Figure 3.3: Four Predictors PDF of Predictive Error Band from 500 to 600](image)

On the other hand, LOCO-2D and LOCO-3D context predictors have a better compression performance on the bands with a low spectral correlation. For example, the bands from 1600 to 1700 cannot keep a high spectral correlation (see in Fig 3.2 (b)). Their spectral correlation changes rapidly and their pixel intensity ranges from −1 to 1. In this case, the two context predictors, LOCO-2D and LOCO-3D context predictors, have a better compression performance. Figure 3.4 shows the probability density function of the predictive error generated by the four predictors. The two context predictors are more effective in removing redundancies. They concentrate the predictive error on a narrow range.

38
In conclusion, the compression performance of the four predictors depends on the attributes of the data. PDPCM has the best compression performance on the bands with a high spectral correlation. On the other hand, the non-linear predictors, 2D-LOCO and 3D-LOCO context predictors, provide a better compression performance on the bands with a low spectral correlation and low intensity. Therefore, the HPT method employs two predictors, PDPCM and LOCO-2D context predictor, to compress the bands with different attributes in an effective manner.

### 3.2 Preprocessing

Ultraspectral data contains a large number of bands, so that its attributes, such as spectral correlation and pixel intensity range, change frequently. As discussed in Section 3.1, the PDPCM has the best compression performance in compressing the data with a high spectral correlation, and the LOCO-2D context predictor has an excellent performance for the data with a low spectral correlation (Especially for the data with a narrow intensity range). So it is more effective to use two different predictors to separably compress the data with different attributes.

Figure 3.5 shows the procedure of the preprocessing. The preprocessing step is to divide

---

**Figure 3.4:** Four Predictors PDF of Predictive Error In The Bands from 1600 to 1700
Figure 3.5: The Pre-processing

thousands of bands into two groups of data (with a high spectral correlation or a low spectral correlation). By way of comparing the spectral correlation with a threshold $T_c$, the data is divided into two groups. If the spectral correlation is lower than $T_c$, the adjacent two bands are extracted from the original data. The extracted bands are reorganized into a new group of data with a low spectral correlation. If the spectral correlation is higher than $T_c$, bands are not extracted from the original data. The remainder of the original data is organized into another group of data with a high spectral correlation. Furthermore, since ultraspectral data generally has a high spectral correlation, the group of data with a high spectral correlation contains a large number of bands and the group of data with a low spectral correlation just contains a few bands.

The two groups of data are separably applied to two different predictors. LOCO-2D algorithm (described in Section 2.4.1) is used to compress the first group of data with a low spectral correlation. The median context is used to estimate the current pixel values. The predictive error is computed by subtracting predicted pixel values from the original pixel values. The predictive error is quantized, and the result is encoded by a template Huffman code.
On the other hand, the group of data with a high spectral correlation is applied to a linear predictor, similar to the PDPCM (described in Section 2.3.2). Two adjacent bands \( n \) and \( n - 1 \) are inputted, and then we compute their spectral correlation \( \text{Cor}(n) \) with the equation (2.29). If the spectral correlation \( \text{Cor}(n) \) is smaller than threshold \( T_c \), the previous band \( n - 1 \) is stored as the initial image and the next band \( n + 1 \) is inputted to repeat the former step. If the spectral correlation \( \text{Cor}(n) \) is bigger than a threshold \( T_c \), the two bands are initially normalized. Using the equation (2.28), the optimal coefficient can be calculated. The PDPCM predictor estimates the predictive pixel value in the current band using the predictive equation (2.27). Then predictor error is calculated by subtracting the predictive pixel values from the original pixel values. Then the predictive error is applied to the next stage.

### 3.3 Block Transform

In order to increase the compression ratio, the HPT method removes the non-essential information stored in the predictive error (generated by PDPCM). Principal component analysis (PCA) is the most effective method to focus the information on a small number of coefficients, but its fast calculation is computationally intensive. So the HPT method adopts the discrete cosine transform, having the nearest efficiency with PCA, to redistribute the information. A two-dimensional discrete cosine transform implements two separable 1D discrete cosine transforms in horizontal and vertical directions so it effectively concentrates information on the low frequency coefficients.

Each band’s predictive error, generated in PDPCM, is divided into a sequence of blocks with the same \( N \times N \) block size. Each block is level-shifted by subtracting the mean of each band, and the result is converted into transform domain using a 2D-DCT. Each transform coefficient block is reshaped with a zigzag pattern (described in Section 2.1), and then it is reordered as a 1D coefficient sequence. Figures 3.6 (a) and (c) show the intensity distribution.
of the predictive error in two typical bands. Their intensity distributions are random. Figures 3.6 (b) and (d) show the intensity distribution of all unidimensional coefficient sequences, where the information is highly concentrated on a number of low frequency coefficients. In fact, the transform coefficient distribution of the arbitrary band’s predictive error can be classified into two types (shown in Figure 3.6 (b) and (d)).

3.4 Rate Allocation

After converting each pixel block into transform domain, the generated transform coefficients are reshaped into a sequence of 1D coefficient sequences. Then the HPT method quantizes
each 1D coefficient sequence by dividing it with a known 1D quantization sequence. Rounding the quotient sequence results in a number of zero-valued coefficients generated in the rounded quotients. A special symbol is used to express remainder of coefficients with zero-value. The process decreases the number of coefficients required to store. Since the aim of lossy compression is to compress images while keeping the reconstructed image quality at a higher level, a regulation to control the removing process is necessary. The HPT method provides a compression protocol to regulate the process by controlling the error generated in compression process. The new developed compression protocol effectively allocates more bits to store essential coefficients. In HPT method, the compression protocol consists of three components: removing parameter $C_o$, essentiality sequence of each index set, and corresponding quantization sequence. Removing parameter $C_o$ is used to regulate the number of the discarded coefficients in each unidimensional coefficient sequence. When the block size is $N \times N$, the removing parameter $C_o$ ranges from 1 to $N^2$. If we want to remove about $1 - C_o$ coefficients per 1D coefficient sequence, the removing parameter is equal to $C_o$.

![Figure 3.7: Reshape Coefficients and Generate Index Sets](image)

Figure 3.7: Reshape Coefficients and Generate Index Sets
Essentiality sequence is used to evaluate the importance of each index set in one band image, where index set is the set of coefficients in the index $i$ and $1 \leq i \leq N^2$. Reshaping all coefficient matrices with the zigzag pattern, coefficient matrices are reordered into 1D coefficient sequences (shown in Fig 3.7). In the case of $N = 5$, there are 25 index sets in the 1D coefficients sequences, and each index set contains 486 elements of coefficients. Figure 3.8 points out the intensity distribution of transform coefficients in the 25 index sets, where the red points are the means of each index set, the green points are the standard deviations of each index set, and the yellow points are the maximum and minimum values of each index set. The index set, that contains more high-value coefficients and changes more dramatically, has higher essentiality. In order to evaluate the essentiality of each index set, a essentiality function $I(i)$ is defined as

$$I(i) = (1 - \alpha_1 - \alpha_2)\mu(i) + \alpha_1\sigma(i) + \alpha_2\max(i), i = 1, 2, \ldots, 25$$

(3.1)

where $i$ is the index number, $\mu(i)$ is the mean of index set $i$, $\sigma(i)$ is the standard deviation of index set $i$, and $\max(i)$ is the maximum of index set $i$. The higher the value of $I(i)$ is,

**Figure 3.8:** Two Types of Distribution of Index Set

the more important the index set is. Computing the essentiality sequence with the equation

44
(3.1), the HPT method ranks index sets with essentiality from high to low. The more important index set is arranged into the forward position and less important index set is arranged into the backward position. The essentiality sequence is stored in the compression protocol. The HPT method reorders the 1D coefficient sequences using the essentiality sequence, and the index set contains more essential information is distributed into the forward positions of 1D coefficient sequences. The HPT method mainly discards the coefficients in the backward positions of 1D coefficient sequences in quantization step. Therefore, reordering 1D coefficient sequences with the essentiality sequence cannot only discard more non-essential coefficients, but also decreases the possibility to discard the essential coefficients. Figure 3.9 points out the reordering process, where the non-essential index set is allocated to the backward points and essential index set is allocated to the forward, while the more non-essential coefficients and fewer essential coefficients are discarded.

![Figure 3.9: Compose Sequence of Index](image)

Furthermore, the HPT method provides an approach to design the quantization sequence. There are two aims in developing the quantization sequence: remove more non-essential coefficients in less essential index set and limit the maximum value existing in quotients. In order to satisfying these two aims, the HPT method provides two quantization expressions.
They are defined as:

\[
Q(i) = \begin{cases} 
\frac{\min(i)}{2}, & i = 1 \\
\frac{a \cdot \max(i) + b \cdot \mu(i)}{a + b}, & 1 < i \leq C_o \\
\frac{c \cdot \max(i) + d \cdot \mu(i)}{c + d}, & i > C_o
\end{cases}
\]  

(3.2)

and

\[
Q(i) = \begin{cases} 
\frac{\mu(i) + k \cdot \max(i) + k \cdot \min(i)}{1 + 2k}, & i = 1 \\
\frac{a \cdot \max(i) + b \cdot \mu(i)}{a + b}, & 1 < i \leq C_o \\
\frac{c \cdot \max(i) + d \cdot \mu(i)}{c + d}, & i > C_o
\end{cases}
\]  

(3.3)

where \(C_o\) is removing parameter, \(\max(i)\) is the maximum abstract value in the index set \(i\), \(\min(i)\) is the minimum abstract value in the index set \(i\), and \(\mu(i)\) is the mean of abstract value in the index set \(i\). The two groups of equations use five scalar parameters \((a, b, c, d, \text{ and } k)\) to adjust the quantization sequence. When the coefficients distribution is similar with the type shown in Figure 3.8 (b), the first group of equations is used to generate quantization sequence. When the coefficients distribution is similar with the type shown in Figure 3.8 (a), the second group of equations is used to generate quantization sequence. There are two advantages to design the quantization sequence in the former equations. The generated quantization sequence not only limits the value of quotients, but also effectively controls the amount of information discarded.

Figure 3.10: Form of Compression Protocol

Figure 3.10 points out the form of compression protocol. The compression protocol
contains 51 points and each point requires 5 bits to store. The first point is a parameter $C_o$, that regulates the number of coefficients required to store in each 1D sequence. The later 25 points are used to store the essentiality sequence of index sets. The last 25 points are used to store the corresponding quantization sequence, that allocates 2 bits to integer and 3 bits to fraction.

3.5 Implementation of HPT Technique

In the HPT method, the ultraspectral data is initially divided into two groups in the pre-processing step. The group of data with a low spectral correlation uses LOCO-2D predictor to remove spatial redundancies. The predictive error is quantized and the result is encoded by a template Huffman code. Another group of data with a high spectral correlation uses the predictor of PDPCM (non-quantization) to remove the spectral redundancies. In order to increase compression ratio, the HPT method discards part of non-essential information. Since the discrete cosine transform effectively concentrates information on a small number of low frequency coefficients, it is used to redistribute information existing in the predictive error. Then a compression protocol is generated to regulate the process of removing non-essential coefficients. Furthermore, the HPT method uses two thresholds $T_s$ and $T_k$ to control the error generated by the compression process. The threshold $T_s$ is a switch filter to restart compression model when the error is too large. The threshold $T_k$ is used to control the generation of compression protocol. If the standard deviation of error is bigger than threshold $T_k$, we increase the parameter $C_o$ to regenerate a new compression protocol and repeat the quantization step. Therefore, the HPT method is an adaptive compression method, where the design of compression protocol is adapt to each band. Figure 3.11 shows the flow diagram of the HPT technique.

The compression process contains the following ten steps:

Step 1: Read the initial band $B_1$ and the adjacent band $B_2$.

Step 2: Normalize the two images $B_1$ and $B_2$, and then get the normal images $B_1'$
and B2′. Compute the spectral correlation between B1′ and B2′. Compare the spectral correlation with threshold $T_c$. If the spectral correlation is bigger than $T_c$, go to the next step. If not, re-read the initial band B1 and its adjacent band B2.

Step 3: Compute the corresponding optimal coefficients and get the optimal predictor. Estimates the next normal band B2″ using the pixels in the previous normal band B1′. Compute the predictive error by subtracting predictive pixel values in band B2″ from original pixel values in band B2′.

Step 4: Divide the predictive error into a sequence of blocks with the same 5×5 block size.

Step 5: Subtract the mean of the predictive error from each block. Converted the result into transform domain using a discrete cosine transform. Calculate the transform coefficient matrix.

**Figure 3.11:** Flow Diagram of Proposal Compression Method
Step 6: Reshape each coefficient matrix using a zigzag pattern to generate a unidimensional coefficient sequence. Generate the bands compression protocol (described in Section 3.4).

Step 7: Reorder each 1D coefficient sequence using the essentiality sequence regulated in compression protocol. Divide the reordered 1D sequence with the corresponding 1D quantization sequence regulated in the compression protocol. Round the quotients and store the result.

Step 8: Reorder all rounded quotient sequences into 5×5 blocks. Convert blocks into original domain using an inverse 2D-DCT.

Step 9: Reconstruct the predictive error and calculate the standard deviation of error $\sigma_e$ in the compression process. Compare $\sigma_e$ with threshold $T_k$. If it is larger than $T_k$ and $C_o$ is smaller than 25 (the maximum value of $C_o$), increase the parameter $C_o$ and return to step to 6 to recreate a new compression protocol. If $\sigma_e$ is either smaller than $T_k$ or parameter $C_o$ is equal to 25, use a special symbol to discard the remainder of zero-valued quotients generated in step 7. Input the new unidimensional sequence into entropy encoder to encode with a template Huffman coding. Output the result.

Step 10: Input the reconstructed band $B_2'''$ into switching filter. Compare the standard deviation of error $\sigma_e$ with threshold $T_s$. If it is bigger than $T_s$, restart the the HPT model and input the next band as the initial band $B_1$. If it is smaller than $T_s$, use the reconstructed predictive error to reconstruct $B_2'''$. Repeat the former steps from 2 to 9. Simulation results of HPT are discussed in Chapter 4.

3.6 Summary

This chapter proposed the hybrid predictive transform coding technique, that consists of four stages: preprocessing, prediction, block transformation, and compression protocol. In
preprocessing step, the ultraspectral data is divided into two groups, and they are separately applied to two different predictor: PDPCM (non-quantization) and LOCO-2D context predictor. Then a 2D-DCT is used to remove the non-essential information existing in the predictive error generated by PDPCM. To remove more non-essential information with a higher quality of reconstructed image, a new developed compression protocol is used to control compression ratio. Furthermore, the HPT method is also an adaptive compression method, using two thresholds $T_s$ and $T_k$ to control the reconstructed error and to adjust compression protocol, respectively. Chapter 4 compares the compression performances of the above mentioned compression methods, namely HPT method, 3D-JPEG, Optimal DPCM, and PDPCM.
Chapter 4

Results and Conclusions

This chapter compares the hybrid predictive transform coding technique’s compression performance with three compression techniques, namely 3D-JPEG (described in Section 2.1), optimal 2D-DPCM (described in Section 2.2), and PDPCM (described in Section 2.2). The goal is to evaluate the advantages and disadvantages of the HPT technique with the above mentioned compression techniques.

4.1 Initial data

The input AIRS data is obtained from National Archives and Records Administration, that records the change of the atmospheric layer in different spectrums. It consists of 135 scan lines with 90 pixels per scan line over 2031 bands, so there are $135 \times 90 \times 2031 = 24,676,650$ pixels. AIRS data allocates 12-bits to per pixel, where 1-bit is allocated to sign, 8-bits are allocated to integer, and 3-bits are allocated to decimal representations. Furthermore, it is important to mention that the AIRS data has been used in other studies about ultraspectral image compression [19]. Figure 1.2 points out eight representative bands. In the following sections, some parameters (defined in Appendix A.1), such as compress ratio (CR), signal-to-noise ratio (SNR), peak signal-to-noise ratio (PSNR), are used to evaluate the compression performance of the above mentioned compression methods.
4.2 Hybrid Predictive Transform Coding Technique

The HPT method, described in Chapter 3, is based on the PDPCM, LOCO-2D, and 2D-JPEG algorithms. It firstly extracts a few bands from the whole data, and then we input them into the LOCO-2D context predictor. When the threshold $T_c$ is 0.9, there are 90 bands that are extracted. The rest of 1941 bands uses a compression technique that is combination of the PDPCM and 2D-JPEG algorithm. The PDPCM is exploited to remove spectral redundancies and the block transformation is used to remove non-essential information. By way of generating a compression protocol, the compression method appears to provide a better performance on discarding non-essential information. In order to control the memory required to store the compression protocol, the block size should not be too large. It is set at value $5 \times 5$. The higher value of parameter $C_0$ is assumed in compression protocol generation, the more information is removed.

![Hybrid Predictive Transform](image)

**Figure 4.1:** CR VS Average SNR

The HPT method applies an adaptive parameter $C_0$ in design of the compression protocol. Figure 4.1 shows the relationship between compression ratio and average SNR, where
$C_o$ is equal to 5, 8, 15, 20, or 25, and relative quantization sequence is equal to $0.5Q$, $0.75Q$ or $Q$ ($Q$ is quantization matrix calculated by the equation (3.2) and (3.2) when $a = 0.5$, $b = 0.3$, $c = 0.2$, $d = 0.2$, $k = 0.2$). $T_s$ is equal to 0.02, 0.04, or 0.05 and $T_k$ is equal to 0.01 or 0.015, or 0.025. Increasing the parameter $C_o$, the number of the discarded coefficients decrease. In this situation, the average SNR generally increases and the compression ratio decreases. Changing the multiple of quantization sequence, the HPT method exploits different quantization sequences including $0.5Q$, $0.75Q$, and $Q$. The higher multiple brings about more error and higher compression ratio. The threshold $T_s$ determines whether to restart the compression model. The threshold $T_k$ determines whether to increase the parameter $C_o$ and regenerate another quantization sequence. Therefore, the threshold $T_s$ determines the error generated in the whole compression process, and the threshold $T_k$ is used to control the fluctuations in compression ratio and average SNR.

![Figure 4.2: CR VS Average SNR](image)

Figure 4.2 depicts the compression result from the LOCO-2D algorithm that shows the relationship between compression ratio and average SNR. The low average SNR does not
mean there is huge error generated in compression process. In fact, the error is not large. The reason of the low average SNR is due to the fact that the input data’s intensity ranges from -1.5 to 1.5 and its standard deviation is too low. Therefore, even if the standard deviation of the error is low, the average SNR is still very low. The low compression ratio is due to the fact that the context predictor requires 2-bits per pixel to record the context number at each point. Figure 4.3 shows bands 1660 and 1720 comparing the original image against the reconstructed one in average SNR is equal to 11.8226 dB.

![Image](image-url)

**Figure 4.3:** Original VS Reconstructed (LOCO-2D algorithm) Band = 1660, 1710 (Average SNR = 11.8226 dB)

On the other hand, the compression result from the HPT algorithm has a high compression ratio when the average SNR is between 25 and 32 dB. In this compression process, the band’s SNR fluctuates in the spectral direction, due to the accumulation of error in the prediction step. Four representative images (with low error and large error) are used to compare the original image with reconstructed one. Figures 4.4, 4.5, and 4.6 show bands 310, 750, 1350, and 1950 comparing the original image against the reconstructed one for different compression ratio (CR = 31.08, CR = 25.08, and CR = 11.7), specifically average SNR = 23.4 dB, average SNR = 26.9 dB, average SNR = 32.02 dB, respectively.
Figure 4.4: Original VS Reconstructed (CR = 31.08, Ave SNR = 23.4 dB RC = 0.32) Band = 310, 750, 1350, 1950 (SNR = 29.34 dB, 29.85 dB, 20.05 dB, 30.53 dB)
Figure 4.5: Original VS Reconstructed (CR = 25.8, Ave SNR = 26.9 dB, RC = 0.42) Band = 310, 750, 1350, 1950 (SNR = 30.23 dB, 31.54 dB, 23.72 dB, 33.59 dB)
Figure 4.6: Original VS Reconstructed (CR = 11.71, Ave SNR = 32.02 dB, RC = 0.59)) Band = 310, 750, 1350, 1950 (SNR = 34.54 dB, 36.4dB, 24.32dB, 36.56dB)
4.3 3D-JPEG

The 3D-JPEG, described in Section 2.1, is based on a block transform coding algorithm. It separates the three-dimensional multispectral data into a number of cubes with the same size of $N \times N \times N$, and then each cube is converted into transform domain using a three-dimensional discrete cosine transform. The transform coefficients are quantized, and then the results are encoded with a predetermined Huffman code. Figure 4.7 shows the relationship between compression ratio and SNR. The size of cube is set at $8 \times 8 \times 8$. Compared with the hybrid predictive-transform algorithm, 3D-JPEG appears to have a lower compression ratio. Figure 4.8 shows two representative bands 750 and 1350 comparing the original image against the reconstructed one for different quantization matrix of $2Q$, and $Q$, $0.25Q$, specifically average SNR = 19.44 dB, average SNR = 26.99 dB, average SNR = 37.05 dB, respectively.

![3D-JPEG](image)

**Figure 4.7:** CR VS Average SNR
Figure 4.8: Original and Reconstructed (Band = 310 and 1350) Average SNR 19.4387 dB, 26.9936 dB, 37.0496 dB
## 4.4 Optimal 2D-DPCM

The optimal 2D-DPCM, described in Section 2.3, employs a linear predictor in the spatial dimension. It uses a linear combination of the previous pixels in the same band to estimate the current pixel. It is quantized by a non-uniform quantization sequence and encoded with a predetermined Huffman code. Figure 4.9 shows the relationship between compression ratio and average SNR. Increasing the quantization level, the compression ratio decreases and the average SNR increases. Figure 4.10 show two representative bands 750 and 1350 comparing the original image against the the reconstructed one for different quantization levels (16, 32, 64) specifically average SNR = 22.14 dB, average SNR = 30.02 dB, average SNR = 35.22 dB, respectively.

![Optimal 2D–DPCM](image)

**Figure 4.9:** CR VS Average SNR
Figure 4.10: Original and Reconstructed (Band = 750 and 1350) Average SNR 22.14 dB, 30.02 dB, 35.22 dB
4.5 PDPCM

The PDPCM, described in Section 2.3, applies the first order optimal DPCM to compress partitioning ultraspectral data, that uses a threshold $T_c$ to limit the spectral correlation between adjacent bands and encodes the quantized information with a predetermined Huffman code. Figure 4.11 shows the relationship between average SNR and compression ratio. The blue stars show the result when the threshold $T_c$ is equal to 0.8, the red stars show the result when the threshold $T_c$ is equal to 0.87, and the green stars show the result when the threshold $T_c$ is equal to 0.93. In PDPCM, the compression ratio decreases with the increase of the average SNR, and the predictive compression method has a good compression performance in the case of the $T_c$ is equal to 0.87. However, this method does not perform a high compression ratio. Figure 4.12 shows two representative bands 750 and 1350 comparing the original image against the reconstructed one for different quantization level (16 32 64) specifically average SNR = 17.12 dB, average SNR = 29.36 dB, average SNR = 36.06 dB, respectively.

\[\text{Figure 4.11: CR VS Average SNR}\]
Figure 4.12: Original VS Reconstructed (Band = 500) Average SNR, 17.12 dB, 29.36 dB, SNR 36.06 dB
4.6 Results

When the average SNR is higher than 32 dB, human eyes cannot distinguish the difference between the original image and the reconstructed image. In this situation, the AIRS data compression is viewed as a low-error or almost loss-less compression. When average SNR is between 25 and 32 dB, the difference between the original image and the reconstructed image can be distinguished. The AIRS data compression is viewed as a lossy compression. Figure 4.13 shows the relationship between compression ratio and average SNR in four different compression techniques (indicated in Section 4.2, 4.3, 4.4, and 4.5). In the low-error compression, the HPT algorithm does not have a higher compression ratio than the PDPCM and the optimal 2D-DPCM, that is due to the fact that

![Compression Results](image)  

Figure 4.13: Compression Results

it needs to allocate the memory required to store initial bands and compression protocol.
However, in the lossy compression, the HPT algorithm has an excellent compression performance to increase compression ratio. It generally provides a better compression performance than 3D-JPEG. Another important advantage of the HPT method is that it is an adaptive method and its compression ratio and average SNR can be adjusted by changing thresholds. (Notice that the average SNR is to calculate the mean of each band’s SNR, which is 10 dB lower than the whole SNR. The calculation expressions of the average SNR and the whole SNR is shown in Appendix A.1 )

4.7 Conclusions

In this thesis, a new lossy ultraspectral image compression algorithm, called hybrid predictive transform (HPT) coding algorithm, is proposed. The HPT algorithm is based on the predictive coding and block transform coding. The predictor is used to remove the spatial or spectral correlation, and extracts the decorrelated information from the data. A discrete cosine transform is used to convert decorrelated information into transform domain, that concentrates information on the low frequency coefficients. Then a compression protocol is generated to regulate how to discard the coefficients containing less information (described in Section 3.4). Furthermore, the HPT method is an adaptive compression method, using two thresholds to control the compression performance.

The HPT method is an excellent lossy compression method. In the lossy compression (average SNR is between 25 and 32 dB), its compression ratio ranges from 9 to 35, that is obviously higher than the other three compression methods’. Moreover, the HPT method is also an adaptive compression method. By changing the value of two thresholds $T_s$ and $T_k$, the compression ratio and average SNR can be adjusted in a zone. However in the low-error compression (average SNR is higher than 32 dB and human eyes cannot distinguish the error), the HPT algorithm does not provide a better compression performance than the predictive coding algorithm. Why does the HPT method provide a worse compression performance in the low-error situation? One reason is that the HPT method needs to allocate
memory required to store initial bands and compression protocols. Another reason is that the possibility density function of transform coefficients distribution is not Gaussian, it requires more memory to store encoded data. After converting the decorrelated information into transform domain, the transform coefficients’ distribution is not as concentrated as Gaussian distribution function. These result in the low compression ratio in low-error compression.

In future research, we can try to solve the above problems that decreases the compression ratio. Therefore, we can study the how to increase the efficiency of transform, making the transform coefficients being highly concentrated. Or, it is possible to decrease the amount of initial bands by adopting a feedback to adjust the error generate in compression process.
Appendix A

Parameters

In this thesis, there are three parameters used to evaluate the compression performance of
the above compression methods, namely average signal-to-noise ratio (average SNR), the
whole signal-to-noise ratio (whole SNR), and compression ratio. Each band’s signal-to-noise
ratio (SNR) is defined as:

\[ \text{SNR}_k = 10 \log_{10} \frac{\text{std}(x_k)^2}{\text{std}(x_k - \tilde{x}_k)^2} \]  \hspace{1cm} (A.1)

where \( x_k \) is the band image \( n \), \( \tilde{x}_k \) is the reconstructed band image \( k \), and

\[ \text{std}(x_k) = \left( \frac{1}{n} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (x_k(i, j) - \mu_k)^2 \right)^{\frac{1}{2}} \]  \hspace{1cm} (A.2)

\[ \text{std}(e_k) = \left( \frac{1}{n} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (e_k(i, j) - \bar{e}_k)^2 \right)^{\frac{1}{2}} \]  \hspace{1cm} (A.3)

\[ e_k = x_k - \tilde{x}_k \]  \hspace{1cm} (A.4)

e_\text{k} is the error in the band image \text{k}, \mu_\text{k} is the mean of band image \text{k}, and \bar{e}_\text{k} is the mean of
error in band image \text{k}. Average SNR is defined as:

\[ \text{SNR} = \frac{1}{B} \sum_{k=1}^{B} \text{SNR}_k \]  \hspace{1cm} (A.5)
where $B$ is the number of band. The whole SNR is defined as

$$\text{SNR} = 10 \log_{10} \frac{\text{std}(d)^2}{\text{std}(e)^2} \quad (A.6)$$

where $d$ is the three-dimensional ultraspectral data, error $e$ is the difference between original data and reconstructed data, and

$$\text{std}(x_k) = \left( \frac{1}{n} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \sum_{k=1}^{B} (x(i, j, k) - \mu_d)^2 \right)^{\frac{1}{2}} \quad (A.7)$$

$\mu_d$ is the mean of the whole ultraspectral data.

Compression ratio is defined as:

$$\text{CR} = \frac{\sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \sum_{k=1}^{B} \text{bits}(i, j, k)}{H \times W \times B \times 12} \quad (A.8)$$

where $\text{bits}(i, j, k)$ is the bits number required to store at $(i, j, k)$, and $H$, $W$, and $B$ is the size of the ultraspectral data.
Appendix B

Optimal Coefficients

In the predictive coding algorithm, we use an predictive combination function to estimate
the predictive pixel values. Section 2.3 presented two predictive models, namely optimal
2D-DPCM, and PDPCM. Appendix B presents the mathematical process to compute the
two models’ optimal coefficients in the predictive combination function.

B.1 Optimal Coefficients in NSHP Model

The optimal 2D-DPCM is the one that has the minimum mean square error (MSE) between
the original pixel values and predictive pixel values. We know that the MSE is defined as:

\[ E\{e^2\} = E \left\{ \left( x(i, j) - \alpha_1 x(i, j - 1) - \alpha_2 x(i - 1, j) \right)^2 \right\} \]  \hspace{1cm} (B.1)

The minimum MSE can be achieved by setting the partial derivation of the MSE to zero.
Then we get

\[ \frac{\partial E\{e^2\}}{\partial \alpha_1} = -E \{e(i, j)x(i, j - 1)\} = 0 \]  \hspace{1cm} (B.2)

\[ \frac{\partial E\{e^2\}}{\partial \alpha_2} = -E \{e(i, j)x(i - 1, j)\} = 0 \]  \hspace{1cm} (B.3)
\[ \frac{\partial E\{e^2\}}{\partial \alpha_3} = -E\{e(i, j)x(i - 1, j - 1)\} = 0 \quad (B.4) \]
\[ \frac{\partial E\{e^2\}}{\partial \alpha_4} = -E\{e(i, j)x(i, j + 1)\} = 0 \quad (B.5) \]

where
\[ e(i, j) = x(i, j) - \alpha_1x(i, j - 1) - \alpha_2x(i - 1, j) - \alpha_3x(i - 1, j - 1) - \alpha_4x(i - 1, j + 1) \quad (B.6) \]

Therefore we know the the error \( e(i, j) \) is orthogonal to \( x(i - m, j - n) \), where \( 0 \leq m \leq 1 \), and \(-1 \leq n \leq 1 \). Through the above four equations (B.2-5), we can compute the optimal coefficients by the following equation
\[ \begin{bmatrix} r_x(0, 0) & r_x(1, 1) & r_x(1, 0) & r_x(1, -2) \\ r_x(1, 1) & r_x(0, 0) & r_x(0, 1) & r_x(0, -1) \\ r_x(1, 0) & r_x(0, 1) & r_x(0, 0) & r_x(0, 2) \\ r_x(1, 2) & r_x(0, -1) & r_x(0, 2) & r_x(0, 0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} r_x(0, 1) \\ r_x(1, 0) \\ r_x(1, 1) \\ r_x(1, -1) \end{bmatrix} \quad (B.7) \]

where
\[ r_x(m, n) = E\{x(i, j)x(j - a, j - b)\}, 0 \leq a \leq 1, 0 \leq b \leq 2 \quad (B.8) \]

is a covariance function of image. The equation to calculate the optimal coefficients in optimal 2D-DPCM model is established.

### B.2 Optimal Coefficients in PDPCM

The optimal coefficient in PDPCM model is determined when the mean square error (MSE) is minimum. In PDPCM, the MSE is defined as:
\[ E\{e_k^2\} = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (x_k(i, j) - \alpha_1(k)x_{k-1}(i, j) - \alpha_0(k))^2 \quad (B.9) \]
where \( x_k(i, j) \) is the pixel value at the point of \((i, j)\) in the band \( k \). The minimum MSE can be achieved by setting the partial derivation of the MSE to zero. Then we get

\[
\frac{\partial E\{e_k^2(i, j)\}}{\partial \alpha_0(k)} = -E\{e_k(i, j)x_{k-1}(i, j)\} = 0 \tag{B.10}
\]

\[
\frac{\partial E\{e_k^2(i, j)\}}{\partial \alpha_1(k)} = -E\{e_k(i, j)\} = 0 \tag{B.11}
\]

where

\[
e_k(i, j) = x_k(i, j) - \alpha_1(k)x_{k-1}(i, j) - \alpha_0(k) \tag{B.12}
\]

Therefore we know the the error \( e_k(i, j) \) is orthogonal to \( x_{k-1}(i, j) \). Through the above four equations (B.10-11), we can compute the optimal coefficients by the following equations

\[
\alpha_0(k) = \left(1 - \frac{r_k(1) - \mu_k^2}{r_k(0) - \mu_k^2}\right) \mu_k \tag{B.13}
\]

\[
\alpha_1 = \frac{r_k(1) - \mu_k^2}{r_k(0) - \mu_k^2} \tag{B.14}
\]

where \( r_k \) and \( \mu_k \) are

\[
r_k(t) = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_k(i, j)x_{k-t}(i, j) \tag{B.15}
\]

\[
\mu_k = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_k(i, j) \tag{B.16}
\]

where \( \mu_k \) is the mean of band \( k \), \( r_k \) is the covariance between the band \( k-t \) and the band \( k \), and \( t \) is the lag in the spectral dimension. The equation to calculate the optimal coefficients in PDPCM model is established.
Bibliography


