International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 2, February 2015

HYPERSPECTRAL IMAGE COMPRESSION USING DUAL-TREE BEZW ALGORITHM

Ms. Gayatri Kallat and Mr. A.Yogeshwaran, M.E.,

Abstract—**This paper proposes a lossless to lossy compression scheme for hyper spectral images based on dual-tree Binary Embedded Zero tree Wavelet (BEZW) algorithm. The algorithm adapts Karhunen– Loève Transform and Discrete Wavelet Transform to achieve 3-D integer reversible hybrid transform and decor relate spectral and spatial data. Since statistics of the hyper spectral image are not symmetrical, the asymmetrical dual-tree structure is introduced. The 3-D BEZW algorithm compresses hyper spectral images by implementing progressive bit plane coding. The lossless and lossy compression performance is compared with other state-of-the-art predictive coding and transform-based coding algorithms on Airborne Visible/Infrared Imaging Spectrometer images. Results show that the 3-D-BEZW lossless compression performance is comparable with the best predictive algorithms, while its computational cost is comparable with those of transform based algorithms.**

Index Terms— **Binary Embedded zerotree wavelet (BEZW), hyperspectral images, Karhunen–Loève transform (KLT), Predictive Coding.**

I.INTRODUCTION

The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) was developed by the NASA Jet Propulsion Laboratory in 1987 and provides spectral images with 224 contiguous bands covering the spectral ranges from 400 to 2500 nm. The unique high spectral resolution has been used in a broad range of scientific research such as terrain classification, agricultural monitoring, and military surveillance. The AVIRIS instrument yields files of several gigabytes of data, which are recoded and stored onboard. Therefore, compression of the hyper spectral image is necessary to facilitate storage and transmission. Compression techniques for hyper spectral images fall into two groups: predictive and transform-based coding. Roger and Cavenor studied Adaptive Differential Pulse-Code Modulation (ADPCM) for the lossless compression of AVIRIS hyper spectral images. They experimented with five linear spatial, spectral, or spatial–spectral predictors optimized by least square (LS) and encoded residuals by a Variable-Length Coding algorithm. Aiazzi *et al.* proposed the Fuzzy DPCM which groups the causal neighbourhoods of each pixel by Fuzzy C-Mean algorithm and then computes the optimized coefficient predictor for each cluster. The final estimate is computed as the weighted sum of all the outputs of predictors, where the weights are the similarity degrees. Mielikainen and Tovainen presented the Cluster DPCM (C-DPCM). The spectral data are clustered into 16 groups by means of the Linde–Buzo–Gray method. Coefficients of the linear predictors of each cluster are optimized by minimizing the mean square error (MSE). Wu and Memo extended their 2-D context-based adaptive lossless image coding (2-D-CALIC) to 3-D-CALIC [5] for hyper spectral images. In 3-D-CALIC,the predictor is switched between intra - and inter band predictor according to the dependence among causal neighbourhoods. Multiband-CALIC (M-CALIC) uses only the inter band prediction, but parameters are optimized for hyper spectral images. M-CALIC outperforms the 3-D-CALIC. Another widely used and important compression technique is transform-based compression, which maps signal samples from the spatial/spectral domain into another space to produce useful statistical properties such as energy compaction and decor related components. Some examples of transforms include the Karhunen–Loève Transform (KLT), Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT). In general, the standard image compression, Joint Photographic Experts Group (JPEG), uses 8 × 8 DCT and the later JPEG2000 uses 2-D DWT. Penna *et al.* compressed hyper spectral images using JPEG2000 and investigated the performance under different transform techniques, including WT, DCT, KLT, and various combinations. Shapiro introduced the embedded zero tree wavelet (EZW). It is a progressive and embedded image coding scheme because it successively encodes values of coefficients in order of importance based on his novel tree structure. Zero tree refers to a tree of insignificant coefficients. Since the integer DWT can be implemented using a lifting scheme, Bilgin *et al.* compressed the hyper spectral image by 3-D isotropic integer DWT decomposition and 3-D-EZW. Said and Pearlman extended Shapiro's EZW and introduced an improved algorithm, Set Partitioning in Hierarchical Trees (SPIHT). Cho

ISSN: 2278 – 909X

and Pearlman showed that SPIHT has better performance than the EZW because it utilizes high-degree zerotrees and generates more compact binary results. Lim *et al.* applied the 3-D-SPIHT algorithm based on symmetrical 3-D-DWT to hyper spectral images and showed that 3-D-SPIHT can be successfully applied to compress hyper spectral images.

II. DISCRETE COSINE TRANSFORM (DCT)

A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering,from lossycompression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations.The use of cosine rather than sine functions is critical for compression, since it turns out (as described below) that fewer cosine functions are needed to approximate a typical signal, whereas for differential equations the cosines express a particular choice of boundary conditions In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common.The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT", its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT". Two related transforms are the discrete sine transform (DST), which is equivalent to a DFT of real and odd functions, and the modified discrete cosine transform (MDCT), which is based on a DCT of overlapping data.

III. THRESHOLDING METHOD

The simplest method of image segmentation is called the thresholding method. This method is based on a clip-level (or a threshold value) to turn a gray-scale image into a binary image. There is also a balanced histogram thresholding. The key of this method is to select the threshold value (or values when multiple-levels are selected). Several popular methods are used in industry including the maximum entropy method, Otsu's method (maximum variance), and k-means clustering. Recently, methods have been developed for thresholding computed tomography (CT) images. The key idea is that, unlike Otsu's method, the thresholds are derived from the radiographs instead of the (reconstructed) image.

IV. DRAWBACKS

Low accuracy

Pixel difference expansion
Performance is very low
Noises occur In the image

V. BINARY EMBEDDED ZERO-TREE WAVELET TRANSFORMS (BEZW)

Binary Embedded Zerotrees of **Wavelet transforms** (B**EZW**) is a lossy image compression algorithm. At low bit rates, i.e. high compression ratios, most of the coefficients produced by a subband transform (such as the wavelet transform) will be zero, or very close to zero. This occurs because "real world" images tend to contain mostly low frequency information (highly correlated). However where high frequency information does occur (such as edges in the image) this is particularly important in terms of human perception of the image quality, and thus must be represented accurately in any high quality coding scheme. By considering the transformed coefficients as a tree (or trees) with the lowest frequency coefficients at the root node and with the children of each tree node being the spatially related coefficients in the next higher frequency subband, there is a high probability that one or more subtrees will consist entirely of coefficients which are zero or nearly zero, such subtrees are called zerotrees. Due to this, we use the terms node and coefficient interchangeably, and when we refer to the children of a coefficient, we mean the child coefficients of the node in the tree where that coefficient is located. We use children to refer to directly connected nodes lower in the tree and descendants to refer to all nodes which are below a particular node in the tree, even if not directly connected. In zerotree based image compression scheme such as EZW and SPIHT, the intent is to use the statistical properties of the trees in order to efficiently code the locations of the significant coefficients. Since most of the coefficients will be zero or close to zero, the spatial locations of the significant coefficients make up a large portion of the total size of a typical compressed image. A coefficient (likewise a tree) is considered significant if its magnitude (or magnitudes of a node and all its descendants in the case of a tree) is above a particular threshold. By starting with a threshold which is close to the maximum coefficient magnitudes and iteratively decreasing the threshold, it is possible to create a compressed representation of an image which progressively adds finer detail. Due to the structure of the trees, it is very likely that if a coefficient in a

ISSN: 2278 – 909X

particular frequency band is insignificant, then all its descendants (the spatially related higher frequency band coefficients) will also be insignificant.EZW uses four symbols to represent (a) a zerotree root, (b) an isolated zero (a coefficient which is insignificant, but which has significant descendants), (c) a significant positive coefficient and (d) a significant negative coefficient. The symbols may be thus represented by two binary bits. The compression algorithm consists of a number of iterations through a dominant pass and a subordinate pass, the threshold is updated (reduced by a factor of two) after each iteration. The dominant pass encodes the significance of the coefficients which have not yet been found significant in earlier iterations, by scanning the trees and emitting one of the four symbols. The children of a coefficient are only scanned if the coefficient was found to be significant, or if the coefficient was an isolated zero. The subordinate pass emits one bit (the most significant bit of each coefficient not so far emitted) for each coefficient which has been found significant in the previous significance passes. The subordinate pass is therefore similar to bit-plane coding.There are several important features to note. Firstly, it is possible to stop the compression algorithm at any time and obtain an approximation of the original image, the greater the number of bits received, the better the image. Secondly, due to the way in which the compression algorithm is structured as a series of decisions, the same algorithm can be run at the decoder to reconstruct the coefficients, but with the decisions being taken according to the incoming bitstream. In practical implementations, it would be usual to use an entropy code such as arithmetic code to further improve the performance of the dominant pass. Bits from the subordinate pass are usually random enough that entropy coding provides no further coding gain.

VI. SPIHIT

In computer science and information theory, datacompression, source coding, or bit-rate reduction involves encoding information using fewer bits than the original representation. Compression can be either lossyor lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression. Lossy compression reduces bits by identifying unnecessary information and removing it. The process of reducing the size of a data file is popularly referred to as data compression, although its formal name is source coding (coding done at the source of the data before it is stored or transmitted). Compression is useful because it helps reduce resource usage, such as data storage space or transmission capacity. Because compressed data must be decompressed to use, this extra processing imposes computational or other costs through decompression; this situation is far from being a free lunch. Data compression is subject to a space–time complexity trade-off. For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed, and the option to decompress the video in full before watching it may be inconvenient or require additional storage. The design of data compression schemes involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced (*e.g.*, when using lossy data compression), and the computational resources required to compress and uncompress the data.

VII. LOSSLESS

Lossless data compression algorithms usually exploit statistical redundancy to represent data more concisely without losing information, so that the process is reversible. Lossless compression is possible because most real-world data has statistical redundancy. For example, an image may have areas of colour that do not change over several pixels; instead of coding "red pixel, red pixel, ..." the data may be encoded as "279 red pixels". This is a basic example of run-length encoding; there are many schemes to reduce file size by eliminating redundancy.The Lempel– Ziv (LZ) compression methods are among the most popular algorithms for lossless storage.DEFLATE is a variation on LZ optimized for decompression speed and compression ratio, but compression can be slow. DEFLATE is used in PKZIP, Gzip and PNG. LZW (Lempel–Ziv– Welch) is used in GIF images. Also noteworthy is the LZR (Lempel-Ziv–Renau) algorithm, which serves as the basis for the Zip method. LZ methods use a table-based compression model where table entries are substituted for repeated strings of data. For most LZ methods, this table is generated dynamically from earlier data in the input. The table itself is often Huffman encoded (e.g. SHRI, LZX). A current LZ-based coding scheme that performs well is LZX, used in Microsoft's CAB format.The best modern lossless compressors use probabilistic models, such as prediction by partial matching. The Burrows–Wheeler transform can also be viewed as an indirect form of statistical modelling.The class of grammar-based codes are gaining popularity because they can compress highly repetitive text, extremely effectively, for instance, biological data collection of same or related species, huge versioned document collection, internet archives, etc. The basic task of grammar-based codes is constructing a context-free grammar deriving a

ISSN: 2278 – 909X

single string. Sequitur and Re-Pair are practical grammar compression algorithms for which public codes are availableIn a further refinement of these techniques, statistical predictions can be coupled to an algorithm called arithmetic coding. Arithmetic coding, invented by Jorma Rissanen, and turned into a practical method by Witten, Neal, and Cleary, achieves superior compression to the better-known Huffman algorithm and lends itself especially well to adaptive data compression tasks where the predictions are strongly context-dependent. Arithmetic coding is used in the bi-level image compression standard JBIG, and the document compression standard DjVu. The text entry system Dasher is an inverse arithmetic coder.

VIII. LOSSY

Lossy data compression is the converse of lossless data compression . In these schemes, some loss of information is acceptable. Dropping nonessential detail from the data source can save storage space. Lossy data compression schemes are informed by research on how people perceive the data in question. For example, the human eye is more sensitive to subtle variations in luminance than it is to variations in color. JPEG image compression works in part by rounding off nonessential bits of information.There is a corresponding trade-off between preserving information and reducing size. A number of popular compression formats exploit these perceptual differences, including those used in music files, images,andvideo.Lossy image compression can be used in digital cameras, to increase storage capacities with minimal degradation of picture quality. Similarly, DVDs use the lossy MPEG-2 Video codec for video compression.In lossy audio compression, methods of psychoacoustics are used to remove non-audible (or less audible) components of the audio signal. Compression of human speech is often performed with even more specialized techniques; speech coding, or voice coding, is sometimes distinguished as a separate discipline from audio compression. Different audio and speech compression standards are listed under audio codecs. Voice compression is used in Internet telephony, for example audio compression is used for CD ripping and is decoded by audio players.

COMPRESSION FORMULAS

1. **Low intensity pixels:**

$$Pl = bl + wl(bl - ml)$$

2. **High intensity pixels:**

$$Ph = bh - wh(bh - mh)$$

3. **Middle intensity pixels:**

$$Pml = bl - wm(bml - mm) + (Pl - Ph)$$

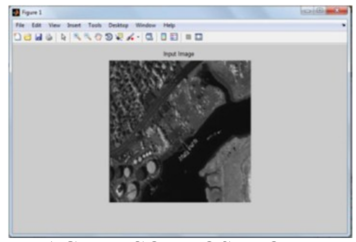
$$Pmh = bh + wm(bmh - mm) + (Pl - Ph)$$

IX. ADVANTAGES

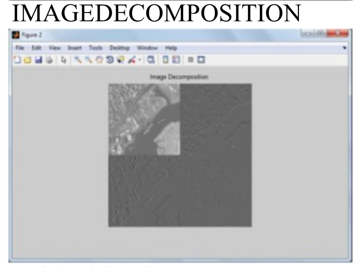
Enhance the Satellite Image Resolution
Spiht is used to compress the image
High resolution image
Storage and transnission

X. SIMULATION RESULTS

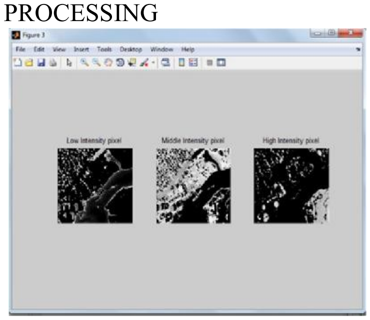
INPUT IMAGE



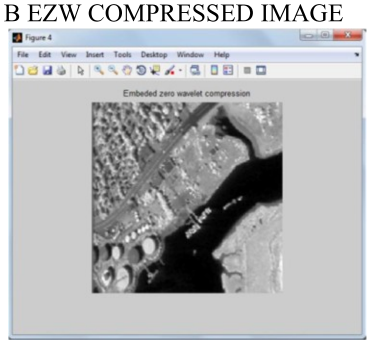
IMAGEDECOMPOSITION



PROCESSING



B EZW COMPRESSED IMAGE



ISSN: 2278 – 909X

XI. CONCLUSION

In this paper, a hybrid reversible integer transform has been proposed in conjunction with an asymmetrical tree structure for lossless and lossy compression of hyperspectral images. The proposed asymmetrical dual-tree structures optimize the performance of the BEZW results. The results demonstrate that the compression performance of BEZW is competitive with the best predictive coding algorithms for lossless compression applications, with significant improvement in computational cost, which is similar to that of other transform based algorithms. The lossy rate-distortion performance of BEZW also consistently exceeds that of other transform based algorithms.

REFERENCES

[1] Kai-jen Cheng and Jeffrey Dill "Lossless to Lossy Dual-Tree BEZW compression for hyper spectral Images" IEEE transactions on geoscience and remote sensing, vol 49, no.16. September 2014

[2] Aaron B. Kiely and Matthew A. Klimesh, " Exploiting Calibration-Induced Artifacts in Lossless Compression of Hyperspectral Imagery,"IEEE TRANSACTION", Issue 1,vol. 47 ,2009**.**

[3] Bruno Aiazzi, Pasquale Alba, Luciano Alparone, and Stefano Baronti, "Lossless Compression of Multi/Hyper-Spectral Imagery Based on a 3-D Fuzzy Prediction," IEEE Transactions on Geoscience and Remote Sensing, VOL. 37, NO. 5, September 2003.

[4] Jarno Mielikainen and Pekka Toivanen, " Clustered DPCM for the Lossless Compression of Hyperspectral Images," IEEE Transactions on Geoscience and Remote Sensing, VOL. 41, No. 12,December 2003

[5] G. A. Triantafyllidis and M. G. Strintzis, "A Context Based Adaptive Arithmetic Coding Technique for Lossless Image Compression," IEEE Signal Processing Letters, VOL. 6, No. 7, July 2000.

ISSN: 2278 – 909X