Lossless Compression of Hyperspectral Images Using Hybrid Based Clustering DPCM

G. Mahalakshmi, A. Sivasankar

Abstract— This project explores the use of hybrid clustering technique for lossless compression method for Hyperspectral images. It is based on the joint use of fuzzy c means and nearest neighbor algorithms. In this method, linear prediction is performed using coefficients optimized for each spectral cluster separately. The difference between the prediction and original values is entropy coded using an adaptive range coder for each cluster. The result shows that this method has lower bit-per-pixel value. It is an extension to the existing lossless compression algorithm. Better partitioning of data is achieved. The technique starts with the fuzzy c means algorithm, performed as the first stage for an adequately high number of centroids and continues with the nearest neighbour algorithm executed for the clusters obtained in the first stage, as the set of initial objects to be merged for relatively complex shapes.

Index Terms— Hyperspectral images, image compression, lossless compression.

I. Introduction

LOSSLESS compression of hyperspectral images is an important part of storing spectral data for long-term storage. Due to the enormous data volumes produced by hyperspectral imagers, a lot of effort has been spent to research more efficient ways to compress hyperspectral images. Three different types of compression modalities for hyperspectral images can be defined. Lossy compression achieves the lowest bit rate among the three modalities. It does not bind the difference between each reconstructed pixel and the original pixel. Instead, the reconstructed image is required to be similar to the original image in a mean-square error sense. Near-lossless compression bounds the absolute difference between each reconstructed pixel and the original pixel by a predefined constant. Lossless compression requires the exact original image to be reconstructed from the compressed data. In this letter, we present an extension to an existing lossless compression algorithm. The novel algorithm has been designed with the single goal of optimizing it for maximum coding efficiency. Other considerations such as low encoder complexity and error resiliency were not part of the design goals of the proposed method.

Previous approaches to lossless compression of Hyperspectral images include both onboard and offline compression methods. One of the methods focusing on onboard compression is A1, which is one of the three distributed source coding algorithms proposed in [1]. It has been designed for onboard compression, and it focuses on coding efficiency. The other two algorithms proposed in [1] are more focused on error resiliency.

Since the focus of this letter is on coding efficiency, we will ignore the other algorithms, which have lower coding efficiency. The A1 algorithm independently encodes nonoverlapped blocks of 16 × 16 samples in each band. This independence makes it easy to parallelize the algorithm. The first block of each band is transmitted uncompressed. The pixel values are predicted by a linear prediction that utilizes pixel values in previous bands, the average pixel values of both the current block and the colocated block in the previous band. Instead of sending prediction parameters to the decoder, they are guessed by the decoder. For each guess, the pixels of the block are reconstructed, and the cyclic redundancy check (CRC) is computed. Once CRC matches the one included in the compressed file, the process terminates. Another method focusing on onboard compression is fast lossless (FL) algorithm [2], which employs the previous band for prediction and adapts the predictor coefficients using recursive estimation.

1.1 The Search for a More Flexible Technique

Clustering, i.e. *placing the similar together and the dissimilar apart*, is not only a model of the basic intellectual activity, and not only a fundamental problem in multivariate analysis, but, first and foremost, a tool used in multiplicity of domains.

Although the general clustering problem can be considered to have found an ultimate search for more powerful (in terms of finding “true solutions”), more adapted (to numerous specific situations) and more efficient (in terms of computational effort) techniques is still on. Due to this, and due to the varied properties of the techniques proposed, the domain of cluster analysis is still developing, also in its theoretical aspect, as witnessed, e.g., by books such as . The dozens of existing approaches have each some merits in one or more of the fields mentioned, but also display, inevitably, poor performance with respect to some other ones. This, again, propels the development, especially of the narrowly designed techniques, like those specializing in definite tasks of pattern recognition, document retrieval etc.

1.2 The Reasons for the New Development

The reasons for the development of the new technique can be summarized as follows: (1) to develop a method of clustering that providing effective means for visualization; then (2) to thereby develop a method that can be effectively used in pattern recognition; and (3) to improve on the existing techniques in more general terms. These reasons acted in the sequence as here provided: from a more modest goal to a much more ambitious one. It was, namely, hoped that effective visualization would constitute a good starting point

Manuscript received on June, 2013.

PG Scholar G. Mahalakshmi, VLSI Design, Anna University Of Technology Madurai, Madurai, India.

Asst.prof A.Sivasankar, ECE, Anna University Of Technology Madurai, Madurai, India.

to the other two goals, but even if it were to stop there, the exercise would be worth the effort. It must be added that while aiming at the technique aiding in visualisation of the data sets we planned to have, at this stage of work, a simple instrument that would be
tested on several cases through human verification. Actually, this is also what visualisation is about.

II. LITERATURE SURVEY

[1] Lossless Compression of Hyperspectral Images Using Clustered Linear Prediction With Adaptive Prediction Length (mar 2012) proposed the use of adaptive prediction length in clustered differential pulse code modulation (C-DPCM) lossless compression method for hyperspectral images.
[2] Lossless Hyperspectral-Image Compression Using Context-Based Conditional Average(dec2009) proposed a new algorithm for lossless compression of hyperspectral images is proposed. The spectral redundancy in hyperspectral images is exploited using a context-match method driven by the correlation between adjacent bands.
[3] Hyperspectral Image Compression Using Three-Dimensional Wavelet Coding (Nov 2009) proposed the Three-Dimensional Set Partitioned Embedded block (3DSPECK) algorithm based on the observation that hyperspectral images are contiguous in the spectrum axis(this implies large inter-band correlations) and there is no motion between bands.
[4] A Block-based Inter-band Lossless Hyperspectral Image Compressor (sep 2009) proposed a hyperspectral image compressor called BH which considers its input image as being partitioned into square blocks, each lying entirely within a particular band, and compresses one such block at a time by using the following steps: first predict the block from the corresponding block in the previous band, then select a predesigned code based on the prediction errors, and finally encode the predictor coefficient and errors.

III. THE TECHNIQUE

We shall now give the complete description of the approach, with just a short consideration of the technical details, which are left to ampler future publications on the subject, linked with further developments of the method.

3.1 Notation

Assume we deal with n objects (observations, items), indexed i, $i\in I = \{1,\ldots,n\}$. Each object is described with m variables (attributes, features), of any character, and such description is denoted x_i, $x_i\in X_I$. We can postulate that these variables form the space of all potential objects, denoted E_X, $X_I\subseteq E_X$. Assume, further, that we can define a distance in E_X, with distance between objects considered denoted $d(x_i,x_j) = d_{ij}$. Distances d_{ij} form a symmetric matrix $D = \{d_{ij}\}_{ij}$. The set I is divided (partitioned) in the clustering problem into subsets (clusters)
denoted A_q, $q = 1,\ldots,p$, where p is the number of clusters, forming a partition P. Whenever applicable, the representative object of a cluster, whether belonging to X_I, or to E_X-X_I, will be denoted x_q (it is assumed that there is only one such object per cluster).

3.2 General Scheme of the Algorithm

The algorithm is divided into two stages: In the first stage the k-means algorithm is performed with predefined number of clusters, p_1 (user's choice) at a relatively "high" level, anyway – much higher than the expected "ultimate" ("objective"?) number of clusters. Just as a hint, for a wide range of values of n one can use p_1=$n_{1/2}$. In this manner, clusters $A_{1\ q}$ are obtained, q=1,…,p_1. Once the first stage terminated, the matrix of distances between clusters $A_{1\ q}$ is calculated, D_1. On the basis of this matrix the classical progressive merger procedure of single link (nearest neighbour) is performed.
Just like p_1, the number of clusters ultimately determined might be an explicit choice of the user, p_2, or the merger procedure can be carried out to the very end, with p_2 determined on the basis of additional information.

3.3 The Rationale

The use of the two known algorithms in the way here proposed is justified by the following reasoning: It is well known that k-means form spherical or ellipsoidal clusters and converge very quickly. If the clusters formed are "small", and "dense" in the set of objects considered, the convergence is even quicker, and there is little hazard of finding a local minimum, so that either only few repetitions are needed, or they can be given up at all. By applying k-means in this way we obtain an effective breakdown of the data set into small, compact subsets, even though these subsets may have very little to do with the actual "shape" of the proper clusters sought. In the second stage single link is used, which agglomerates the subsets according to minimum distances, so that if these subsets form any linear and complex shape, it should get uncovered, with the single link algorithm not so much penalised by the necessity of maintaining and re-calculating the distance matrix, owing to the shrinking of the dimension of the problem in the first stage.

3.4 Some Technical Remarks

The primary issues of technical nature, which arise in the implementation and running of the algorithm are quite obvious: **i.** determination of p_1: besides the hint provided above, caution must be made of the maintenance of the reasonable proportions between n, p_1 and the envisaged p_2; one might also use a constant divisor, bringing n down to p_2; this issue is, of course, closely associated with the fact that neither k-means nor single link by themselves provide a way to determine the "proper" p_2; **ii.** generation of the initial centroid candidates for the k-means stage: given that we start with a much bigger number of centroids than the sought number of final clusters, the initial centroid candidates can be determined by a method different from random choice in E_X (or X_I); **iii.** calculation of the distance matrix D_1; this is the key issue in the computational efficiency of the algorithm; in the application developed to implement the method, a user is offered three options at this point: (**1**) complete enumeration (i.e. $d(A_{1q},A_{1q'})$ = min $\{d_{ij}$: $x_i\in A_{1q}$, $x_j\in A_{1q'}\}$ is obtained on the basis of all pairs i, j such that $x_i\in A_{1q}$, $x_j\in A_{1q'}$;(**2**) the value of $d(A_{1q},A_{1q'})$ is calculated as the d_{ij} between $x_i\in A_{1q}$ that is the closest to $x_{q'}$ and $x_j\in A_{1q'}$ that is the closest to x_q; (**3**) a predefined proportion (user's choice) of objects in both clusters is compared conform to the scheme (2) above. The fact that these choices are offered in determination of D_1 comes from the contribution of this phase of functioning of the algorithm to the overall computational burden, both in terms of time and memory requirements.

Prediction is performed using a linear predictor. All the pixels

used to make the prediction have the same spatial location as the current pixel, and the coefficients are optimized to minimize the mean-square error inside each cluster, i.e., optimization is performed for each cluster separately.Prediction coefficients are encoded using an adaptive 12-bit range coder. Prediction length N_z is selected for each band z from 10 to 200 in steps of 10 by trying experimenting with all possibilities and selecting the one that minimizes the sum of entropy-coded residual and prediction coefficient data.

The resulting differences between the predicted and actualpixel values, i.e., residual, are entropy coded using a range coder, with a separate model for each cluster at each band. Hence, the cluster and band are used as context for the static entropy coder. First, frequency tables are estimated using a normal distribution fit. Next, the mean and variance of a normal distribution are saved as side information. Before actual residual encoding, the residual between the actual frequency table and the normal distribution is encoded using an adaptive 8-bit range coder. For decoding, class indices are stored as 16-bit integers, which are encoded using an adaptive 16-bit range coder.

IV. RESULTS

We have presented a modification to the C-DPCM lossless compression method for hyperspectral images. The modification to the C-DPCM method showed 2%, 1%, and 3% average improvements on the 16-bit calibrated images, the 16-bit uncalibrated images, and the 12-bit uncalibrated images respectively.

Comparison chart

Existing	Area usage	Power in mW	Delay in ns
FCM	146316	29.95	9.56
ENODER CONTROL	146386	61.12	3.85
POST PROCESS	133704	44.09	3.85

Proposed	Area usage	Power in mW	Delay in ns
HYBRID	131020	4.6	9.56
ENCODER CONTROL	132684	23.61	3.86
POST PROCESSCONTROL	132684	19.07	3.86

V. CONCLUSION

Hybrid based clustering for lossless compression of Hyperspectral images using VHDL shows that it consumes less power, less area and less delay for SPARTAN devices rather implementing lossless compression algorithm by fuzzy based clustering DPCM. Hybrid clustering also provides good compression ratio.

REFERENCES

[1] J. Mielikainen, "Lossless compression of hyperspectral images using clustered linear prediction with adaptive prediction length," *IEEE Signal Process. Lett.*, vol. 9, no. 6, pp. 157–160, mar. 2012.

[2] J.Mielikainen and P. Toivanen, "Lossless Hyperspectral-image compression using context based conditional average," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 12, pp. 2943–2946, Dec. 2009.

[3] C.-C. Lin and Y.-T.Hwang, "Hyperspectral image compression using three dimensional wavelet coding," *IEEE Geosci.RemoteSens.Lett.*, vol. 7, no. 3, pp. 558 562, Nov. 2009.

[4] M. Slyz and L. Zhang, "A block-based inter-band lossless Hyperspectral image compressor," in *Proc. IEEE Data Comp. Conf.*, 2005, pp. 427–436,Sep.2009.

[5] A. Abrardo, M. Barni, E. Magli, and F. Nencini, "Error-resilient and low-complexity onboard lossless compression of hyperspectral images by means of distributed source coding," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 4, pp. 1892–1904, Mar. 2009.

[6] B. Huang and Y. Sriraja, "Lossless compression of hyperspectral imagery via lookup tables with predictor selection," in *Proc. SPIE*, 2006, pp. 63 650L.1–63 650L.8.

[7] B. Aiazzi, L. Alparone, and S. Baronti, "Quality issues for compression of Hyperspectral imagery through spectrally adaptive DPCM," in *SatelliteData Compression*. New York: Springer-Verlag.

[8] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[9] A. B. Kiely and M. A. Klimesh, "Exploiting calibration-induced artifacts in lossless compression of hyperspectral imagery," *IEEE Trans. Geosci.Remote Sens.*, vol. 47, no. 8, pp. 2672–2678, Aug. 2009.