2022 5th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT) | 978-1-6654-7647-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/IMPACT55510.2022.10029076

Listless Block Cube Tree Coding for Low Resource Hyperspectral Image Compression Sensors

Harshit Chandra[1], Shrish Bajpai[2]
[1,2] *ECE Department, Faculty of Engineering and Information Technology, Integral University, Lucknow, Uttar Pradesh, India*
[1] harshitchandra55@gmail.com
[2] shrishbajpai@gmail.com

Abstract— **Hyperspectral (HS) image has rich spectral information content, which facilitates multiple applications including remote sensing. Due to the big data size of the HS image, compression is a required process for the efficiency of image storage and transmission. However, the complexity of the compression algorithms turns real-time compression into a very challenging task. A novel listless set partitioned hyperspectral image compression algorithm is proposed. The proposed compression algorithm uses zero block cube tree structure to exploit the inter and intra sub-band correlation to achieve the compression. From the result, it has been clear that the proposed compression algorithm has low coding complexity with at-par coding efficiency. Thus, it can be a suitable contender for low-resource hyperspectral image sensors.**

***Keywords*— Coding Complexity * Compression * Hyperspectral Image * Set Partitioned Hyperspectral Image Compression * Wavelet Transform**

I. Introduction

Hyperspectral image has abundant spatial and spectral information which are gathered in spectral range from 400 nm to 2500 nm having a spectral resolution of 10 nm [1]. The HS images are widely used in remote sensing, precision agriculture, military target detection, mineral exploration, health care etc [2]. Due to the large data size and a lot of unwanted redundancy of HS images, image compression becomes a necessary step before transmission of the image from the sender end to the receiver end. Besides saving the onboard sensor memory, the hyperspectral image compression algorithm (HSICA) also reduces computational complexity (time), saves transmission bandwidth and reduces the sensor power computation [3]. The HS image is 3D data but it is different from the video data as the third dimension (spectral) is related to the 'wavelength' while for video, the third dimension is time (temporal) [4].

The HSICA are broadly divided into five different classes named as predictive coding (PC), vector quantization (VQ), machine learning (ML) based algorithm, transform-based coding (TC) and hybrid compression algorithm. The PC compression algorithms are the least complex compression algorithm but they have very low coding gain. The VQ compression algorithms use the dictionary to achieve compression. The same dictionary is available in the encoder and decoder end. The dictionary generates the coded symbol and the same pattern is defined as a unique symbol. The ML compression algorithms have high coding gain but it has high coding complexity. The PC, VQ and ML compression algorithms work only for lossless compression and if any data loss happens during the transmission then the coding gain reduces significantly. The TC compression algorithms apply the mathematical transform (wavelet, cosine, fourier) to the HS image and convert the same to the frequency domain. Among these mathematical transform, wavelet transform is widely used as it gives excellent energy clustering in space and frequency due to its pyramid structure. The hybrid compression algorithm is the combination of any two of the mentioned algorithms to achieve compression [5-6].

The aim of the manuscript is to introduce a listless implementation of the block cube tree coding technique which has low computation complexity with high coding gain and should work at any coding rate. The remaining manuscript is arranged as follows. Section 2 covers related work associated to the proposed compression algorithm. Section 3 gives detailed explanation of the proposed HSICA followed by outlines simulation results and detail analysis. Conclusion is covered in the last section of the paper.

978-1-6654-7647-8/22/$31.00 ©2022 IEEE

II. RELATED WORK

The wavelet transform-based set partitioned HSICA is a special type of compression algorithm which have high coding efficiency, embedded output and low coding complexity. These algorithms use the set structure (pyramid structure of the transform HS image) to accumulate massive number of insignificant coefficients of the transform HS image either by zero block cube or zero tree or zero block cube tree [7]. The most significant bit plane is given high priority and encoded first till the bit budget is available. The 3D-SPECK [8] uses zero block cube to group the insignificant coefficients. It utilizes the two linked lists to track the significance of the sets or coefficients. In the same way, 3D-SPIHT [9] uses zero tree cube to group the insignificant coefficients and 3D-WBTC [10] uses zero block cube tree to group the insignificant coefficients. The 3D-SPIHT [9] and 3D-WBTC [10] employ the three linked lists to find the significance of the sets or coefficients, while 3D-NLS and 3D-LSK are listless HSICAs, use specific markers to trace the significance of particular sets or coefficients. [11]. The 3D-LMBTC and 3D-ZM-SPECK have low coding memory requirements but this comes with the cost of coding gain. The 3D-LCBTC utilizes two types of markers and two small lists for the tracking of the sets [12].

III. 3D - Listless Block Cube Tree Coding

The 3D-Listless Block Cube Tree Coding (3D-LBCTC) is the low-weight listless implementation of the 3D-WBTC [10]. The 3D-WBTC utilizes the linked list (three) to trace the significance of the block cube trees or coefficients. At a very low bit rate, the list-based HSICAs have low computation complexity but at the medium and high bit rates, these algorithms suffer from high computation complexity which reduces the performance of the HS image sensor. The proposed HSICA 3D-LBCTC does not use any linked list but it uses eight different types of markers for the tracking (coefficients or sets). Details of markers employ 3D-LBCTC are presented in Table 1.

Table 1 : Details of the markers used in the proposed HSICA

Marker	Detail of marker
IC	This coefficient is insignificant or not checked to the current bit plane
NC	This coefficient is newly significant to the current bit plane and no refinement is needed
SC	This coefficient is already significant in the previous bit plane and the refinement bit is generated
DC	This coefficient is the first child of the tree having all descendants of its parents.
CC	This coefficient is checked for significance during the insignificant pass
SD	This coefficient is the first child in the block cube tree having all the descendants of the parent block cube.
SG	This coefficient is the first grandchild having all grand descendants of its grandparent block cube
SN*	These markers are used at the leading nodes of the bit plane
SN2	This coefficient is the first child of the SD set. This coefficient and its sixty-four neighbors (4 x 4 x 4) can be skipped
SN3	This coefficient is the first grandchild of the SD set. This coefficient and its five hundred twelve neighbors (8 x 8 x 8) can be skipped
SN4	This coefficient is the first great-grandchild of the SD set. This coefficient and its four thousand ninety-six neighbors (16 x 16 x 16) can be skipped

The encoding process of the proposed HSICA is divided into two pass named as Initialization Pass (IP) and Bit Plane Pass (BPP). Further, BPP is divided into three sub-passes named as Insignificant Coefficient Pass (ICP), Insignificant Set Pass (ISP) and Refinement Pass (RP).

Initialization Pass : This compression algorithm is initialized by calculating the top bit plane with the help of Eq 1. The transform HS image is converted to the 1D array Υ_i through linear indexing.

$$n = \lfloor \log_2[\max\{|\Upsilon_i|\}] \rfloor \quad (1)$$
$$\daleth = 2^n \quad (2)$$

The 'n' is the top most bit plane while $\daleth$ is the maximum threshold of the transform HS image as shown in Eq 2. All block cubes present in the LLL band are tested for significance first. This pass runs only one-time throughput the encoding process.

Bit Plane Pass : After the IP, the BPP is initiated. This pass runs for all bit planes until the bit budget is available. It has the following three sub-passes.

a. **Insignificant Coefficient Pass (ICP) :** This pass is used to test the coefficients which are insignificant to the last bit plane. The coefficient having the 'IC marker' will be tested against current threshold. If the coefficient is significant against the current threshold then the marker is changed to the 'CC marker'.
b. **Insignificant Set Pass (ISP)** : This pass performs the significant testing of the sets (zero tree) against the current threshold. If set is significant against the threshold, then it is partitioned as per zero block cube tree partitioned rule. This process is replicated till it reaches to the coefficient level or partitioned sets are insignificant.
c. **Refinement Pass (RP)** : All previous significant bit (significant in last bit plane) has to go through the refinement pass. The refinement bit generates and sends to the output bit stream. The coefficients having 'SC markers' or 'CC markers' will generate the refinement bit against the current threshold.

The HSICA initiates from the topmost bit plane and runs till the bit budget is available. All block cubes in the LLL band are tested at the beginning of the encoding process. The size of the block cube is '2 x 2 x 2'. The block cube present in the top of left corner has no descendant and the other seven corresponding block cubes have eight offspring each in the high-frequency sub-bands. This creates the block cube tree and they are marked as 'SN* marker' according to the wavelet orientation level. If any block cube tree is tested significant against present threshold, it will be partitioned into the block cubes and the change of the marker also happens. The process is repeated till last bit plane gets processed or the bit budget is available.

IV. Experiment Result & Analysis

To measure the performance of the proposed HSICA, it is compared with the state of art HSICAs 3D-SPECK (HSICA 1), 3D-SPIHT (HSICA 2), 3D-WBTC (HSICA 3), 3D-LSK (HSICA 4), 3D-NLS (HSICA 5), 3D-LMBTC (HSICA 6), 3D-ZM-SPECK (HSICA 7) and 3D-LCBTC (HSICA 8). The three standard HS images Washington DC (HS Image I), and Cuprite (HS Image II) are used for performance evaluation. All algorithms are run on the same hardware and software platform. The coding complexity (computation time) and coding efficiency (Peak Signal to Noise Ratio) is used as performance measuring parameter for the 3D-LBCTC with the other state of art HSICA [8-16].

The coding efficiency is determined in decibels (dB) for Peak Signal to Noise Ratio (PSNR) [17]. The five-level 3D-DWT is applied to the HS image. The transform coefficients are quantized to the nearest integer and covert in 1D array through linear indexing [15].

4.1 Coding Complexity : The coding complexity of any HSICA is calculated as time required by the hardware resources to run (encode and decode) it. The decoding time is lower than the encoding time as there is no comparison operations are required in the decoding process. The high complex nature of any HSICA requires high processing time for generation of the output bit stream. From Table 2 and Table 3, it is clear that the proposed 3D-LBCTC requires less time than the other HSICA except for 3D-LSK, 3D-NLS and 3D-LCBTC. It is due to the listless nature of the compression algorithm. The list-based algorithms required multiple read or write operations (several memory access) which makes the algorithm complex in nature. Thus, it requires more time for the encoding/decoding of the coefficients [17].

4.2 Coding Efficiency : The coding efficiency is calculated in terms of the number of bits required to achieve the desired quality of the reconstructed HS image. It is measured mathematically as in the terms of PSNR as in Eq 3

$$PSNR = 20\ \log_{10}\left[\frac{Signal\ Amplitude_{\ max}}{MSE}\right] \qquad 3$$

$$\text{MSE} = \frac{1}{\lambda}\sum_{a}\sum_{b}\sum_{c}[A(\alpha,\beta,\gamma) - B(\alpha,\beta,\gamma)]^2 \qquad 4$$

The mean square error (MSE) is defined in Eq 4 The total number of pixels in the HS image is defined by the λ while the original HS image and reconstructed HS image are defined as $A(\alpha,\beta,\gamma)$ & $B(\alpha,\beta,\gamma)$.

It is observed from Table 4 that the coding efficiency of the 3D-LBCTC varies from -0.26 dB to 1.7 dB. The loss of the coding gain is due to the bit budget exhaust in between the bit plane. When the bit budget is exhausted in end of the bit plane, it gives a slightly higher coding gain. The variation of the coding gain between 3D-LBCTC and 3D-SPECK is -0.2 dB to +0.37 dB for HS image I and -0.13 dB to 0.27 dB for HS image II. The dissimilarity of the coding gain between 3D-LBCTC and 3D-SPIHT is 0.01 dB to 0.58 dB for HS image I, and 0.65 dB to 1.7 dB for HS image II. In the same way, the difference between the 3D-LBCTC and 3D-WBTC is -0.18 dB to 0.37 dB for HS image I, and -0.24 dB to 0.17 dB for HS image II.

V. CONCLUSION

In this paper, we have proposed a novel compression algorithm that exploits intra sub-band and inter-sub-band correlation to achieve a high coding gain. Due to its listless nature, it has low coding complexity and fixed coding memory, which makes it as an optimum choice for the onboard HS image sensors. The coding complexity can be reduced further by reducing the number of markers. The coding gain can also increase with the use of curvelet or ripplet or contourlet transform as edges in the HS images are define properly through these advance mathematical transform. The coding memory can reduce through minimize the number of markers.

ACKNOWLEDGEMENT

The Manuscript Communication Number for the article is IU/R&D/2022-MCN0001692.

REFERENCES

1. Chutia, D., Bhattacharyya, D. K., Sarma, K. K., Kalita, R., & Sudhakar, S. (2016). Hyperspectral remote sensing classifications: a perspective survey. Transactions in GIS, 20(4), 463-490.
2. Kumar, V., Singh, R. S., & Dua, Y. (2022). Morphologically dilated convolutional neural network for hyperspectral image classification. Signal Processing: Image Communication, 101, 116549.
3. Dua, Y., Kumar, V., & Singh, R. S. (2021). Parallel lossless HSI compression based on RLS filter. Journal of Parallel and Distributed Computing, 150, 60-68.
4. Das, S. (2021). Hyperspectral image, video compression using sparse tucker tensor decomposition. IET Image Processing, 15(4), 964-973.
5. Dua, Y., Kumar, V., & Singh, R. S. (2020). Comprehensive review of hyperspectral image compression algorithms. Optical Engineering, 59(9), 090902.
6. Dua, Y., Singh, R. S., Parwani, K., Lunagariya, S., & Kumar, V. (2021). Convolution neural network based lossy compression of hyperspectral images. Signal Processing: Image Communication, 95, 116255.
7. Bairagi, V. K., Sapkal, A. M., & Gaikwad, M. S. (2013). The role of transforms in image compression. Journal of the institution of engineers (INDIA): Series B, 94(2), 135-140.

Table 2 : Analysis of the computational complexity of the different HSICAs with the 3D-LBCTC (Encoding Time)

Bit Rate	HSICA 1 [8]	HSICA 2 [9]	HSICA 3 [10]	HSICA 4 [11]	HSICA 5 [12]	HSICA 6 [13]	HSICA 7 [14]	HSICA 8 [15]	3D-LBCTC	HSICA 1 [8]	HSICA 2 [9]	HSICA 3 [10]	HSICA 4 [11]	HSICA 5 [12]	HSICA 6 [13]	HSICA 7 [14]	HSICA 8 [15]	3D-LBCTC
	HS Image I									HS Image II								
0.1	25	7.5	6.50	0.80	0.91	3.90	1.78	0.76	0.98	17.3	6.3	4.7	0.9	1.12	3.2	1.78	0.86	1.21
0.2	57.9	25.8	24.8	1.10	1.21	5.10	2.81	1.04	1.21	55.8	26	16.6	1.2	1.54	6.8	3.01	1.09	1.75
0.3	92.1	37.5	32	1.50	1.65	7.70	3.68	1.41	1.71	107.9	45.5	39.1	2	2.27	7.1	4.08	1.92	2.61
0.4	269.7	117.9	195.5	2.00	2.12	9.70	5.69	1.93	2.29	182.3	75.6	68.2	2.1	2.41	9.2	5.21	2.07	2.93
0.5	414.8	140.1	211.2	2.50	2.64	11.30	7.41	2.44	2.78	276.1	95.4	93.3	2.2	2.58	11.1	6.32	2.11	3.01
0.6	576	166.4	247.9	2.90	3.02	13.30	7.99	2.82	3.22	298.4	161.7	155.7	3.4	3.61	12.9	7.55	3.24	3.97
0.7	887.5	405.7	625	3.20	3.37	18.10	9.66	3.13	3.61	438.8	179.2	202.2	3.9	4.21	15	8.76	3.79	4.71
0.8	1130.5	474.2	710.2	3.80	3.96	20	9.91	3.85	4.11	558.7	198.5	358.5	4.2	4.48	16.5	9.66	4.02	5.02
0.9	1334.6	555.7	746	4	4.14	20.60	12.53	4.04	4.34	656.1	282.8	371	4.4	4.69	18.1	11.20	4.12	5.54
1	1497.5	575	804	4.41	4.57	21.10	13.21	4.38	4.92	905.1	364	652.5	5	5.23	20.3	15.89	5.02	6.07

Table 3 : Analysis of the computational complexity of the different HSICAs with the 3D-LBCTC (Decoding Time)

Bit Rate	HSICA 1 [8]	HSICA 2 [9]	HSICA 3 [10]	HSICA 4 [11]	HSICA 5 [12]	HSICA 6 [13]	HSICA 7 [14]	HSICA 8 [15]	3D-LBCTC	HSICA 1 [8]	HSICA 2 [9]	HSICA 3 [10]	HSICA 4 [11]	HSICA 5 [12]	HSICA 6 [13]	HSICA 7 [14]	HSICA 8 [15]	3D-LBCTC
	HS Image I									HS Image II								
0.1	17.40	6.10	5	0.70	0.79	2.30	1.71	0.64	0.82	13.40	5	3.1	0.7	0.94	2.2	1.70	0.63	1.12
0.2	48.80	24.80	22.50	1.07	1.07	3.30	2.71	1.01	1.14	46.70	22.10	14.6	1	1.32	4.8	2.92	0.91	1.64
0.3	75.40	34.80	28.50	1.45	1.43	4.90	3.59	1.33	1.53	93.70	40.20	35.4	1.8	2.04	5.5	3.98	1.69	2.44
0.4	264.2	106.3	180.4	1.70	1.94	8.10	5.41	1.62	2.01	162.5	70.10	65.8	1.9	2.31	7	5.07	1.81	2.74
0.5	339.1	135.4	191.7	2.20	2.31	7.70	6.82	2.07	2.52	236.1	88.30	91.5	2	2.45	8.5	6.01	1.92	2.91
0.6	532.4	149.6	244.6	2.60	2.79	9.80	7.95	2.48	2.94	281.2	160.9	149	2.9	3.38	10.2	7.17	2.79	3.57
0.7	807.6	327.1	558	2.70	3.04	11.60	8.80	2.53	3.21	435	175.8	196.8	3.1	4.03	11.9	8.28	2.97	4.47
0.8	1058.1	448.9	675.3	3.10	3.67	13.40	9.38	3.09	3.84	525.9	195.3	316	3.8	4.24	13.1	9.21	2.61	4.86
0.9	1142.3	486.2	725	3.20	3.93	13.60	11.82	3.33	4.02	599.2	273.5	366.9	4	4.47	15	10.34	3.83	5.32
1	1289.7	504	774	3.70	4.24	15.50	12.31	3.87	4.51	884.4	346.6	596	4.5	5.07	15.90	14.03	4.38	5.88

Table 4 : Analysis of the coding efficiency of the different HSICAs with the 3D-LBCTC (PSNR)

Bit Rate	HSICA 1 [8]	HSICA 2 [9]	HSICA 3 [10]	HSICA 4 [11]	HSICA 5 [12]	HSICA 6 [13]	HSICA 7 [14]	HSICA 8 [15]	3D-LBCTC	HSICA 1 [8]	HSICA 2 [9]	HSICA 3 [10]	HSICA 4 [11]	HSICA 5 [12]	HSICA 6 [13]	HSICA 7 [14]	HSICA 8 [15]	3D-LBCTC
	HS Image I									HS Image II								
0.1	38.53	38.28	38.50	38.35	38.12	38.29	38.33	38.31	38.71	25.64	24.67	25.77	25.65	24.61	25.60	25.79	25.49	25.69
0.2	41.54	41.34	41.52	41.49	41.27	41.19	41.42	41.59	41.77	30.92	29.44	31.03	30.88	29.33	30.77	30.87	30.84	31.14
0.3	43.51	43.30	43.49	43.55	43.30	43.48	43.57	43.58	43.75	34.55	33.36	34.58	34.55	33.27	34.42	34.59	34.61	34.49
0.4	45.26	45.11	45.25	45.09	45.09	44.59	45.24	45.28	45.59	38.05	37.04	38.15	38.05	36.97	37.50	38.16	38.18	38.03
0.5	46.81	46.60	46.81	46.76	46.41	46.09	46.73	46.83	47.18	41.27	40.51	41.37	41.32	40.45	41.17	41.26	41.39	41.57
0.6	48.45	48.24	48.43	48.42	48.21	48.38	48.39	48.49	48.25	43.46	42.58	43.57	43.47	42.50	43.36	43.43	43.52	43.61
0.7	49.76	49.53	49.74	49.73	49.50	49.17	49.69	49.78	50.08	45.55	45.00	45.81	45.78	44.89	45.60	45.68	45.83	45.65
0.8	51.12	50.84	51.09	51.07	50.76	50.28	50.97	51.17	51.27	47.12	46.43	47.26	47.07	46.38	47.03	47.11	47.16	47.33
0.9	52.24	52.06	52.22	52.24	52.06	51.67	52.12	52.26	52.19	48.74	47.95	48.85	48.75	47.91	48.66	48.78	48.91	48.61
1	53.52	53.32	53.51	53.49	53.33	53.46	53.47	53.59	53.47	49.83	49.24	49.98	49.86	49.22	49.68	49.71	50.01	50.08

8. Tang, Xiaoli, William A. Pearlman, and James W. Modestino. "Hyperspectral image compression using three-dimensional wavelet coding." In Image and Video Communications and Processing 2003, vol. 5022, pp. 1037-1047.
9. Tang, X., & Pearlman, W. A. (2006). Three-dimensional wavelet-based compression of hyperspectral images. In Hyperspectral data compression (pp. 273-308). Springer, Boston, MA.
10. Bajpai S, Kidwai NR, Singh HV (2019) 3D wavelet block tree coding for hyperspectral images. International Journal of Innovative Technology and Exploring Engineering 8(6C):64–68.
11. Ngadiran, R., Boussakta, S., Sharif, B., & Bouridane, A. (2010, May). Efficient implementation of 3D listless SPECK. In International Conference on Computer and Communication Engineering (ICCCE'10) (pp. 1-4). IEEE.
12. Sudha, V. K., & Sudhakar, R. (2013). 3D listless embedded block coding algorithm for compression of volumetric medical images.
13. Bajpai, S., Kidwai, N. R., Singh, H. V., & Singh, A. K. (2019). Low memory block tree coding for hyperspectral images. Multimedia Tools and Applications, 78(19), 27193-27209.
14. Bajpai, S., Kidwai, N. R., Singh, H. V., & Singh, A. K. (2022). A low complexity hyperspectral image compression through 3D set partitioned embedded zero block coding. Multimedia Tools and Applications, 81(1), 841-872.
15. Bajpai, S. (2022). Low complexity block tree coding for hyperspectral image sensors. Multimedia Tools and Applications, 2022, 81(23), 33205–33232.
16. Sharma, D., Prajapati, Y. K., & Tripathi, R. (2018). Success journey of coherent PM-QPSK technique with its variants: a survey. IETE Technical Review.
17. Setiadi, D. R. I. M. (2021). PSNR vs SSIM: imperceptibility quality assessment for image steganography. Multimedia Tools and Applications, 80(6), 8423-8444.
18. Kidwai, N. R., Khan, E., & Reisslein, M. (2016). ZM-SPECK: A fast and memoryless image coder for multimedia sensor networks. IEEE Sensors Journal, 16(8), 2575-2587.