Contourlet Transform Based Listless Block Cube Tree Coding For Hyperspectral Images

Shrish Bajpai
Department of Electronics & Communication Engineering, Faculty of Engineering & IT, Integral University, Lucknow, Uttar Pradesh, India
E-Mail : shrishbajpai@gmail.com

Divya Sharma
Department of Electronics & Communication Engineering, Institute of Engineering and Technology, Lucknow, Uttar Pradesh, India
E-Mail : divya.fgiet@gmail.com

Naimur Rahman Kidwai
Department of Electronics & Communication Engineering, Faculty of Engineering & IT, Integral University, Lucknow, Uttar Pradesh, India
E-Mail : naimkidwai@gmail.com

2026 6th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT) | 979-8-3315-8478-8/26/$31.00 ©2026 IEEE | DOI: 10.1109/IMPACT68503.2026.11468267

Abstract— **The performance of compression algorithms at low bit rates is a critical benchmark, particularly for hyperspectral imaging where the fidelity of reconstruction is paramount. Although wavelet-based approaches are prevalent in the literature, they frequently present a trilemma of undesirable trade-offs, suffering from insufficient coding efficiency, exorbitant memory requirements, or high computational overhead.**

The proposed compression algorithm employees advance wavelet transform to leverage both spectral and spatial redundancies found in HS data cubes. Present study explores the utilization of block tree coding algorithm and contourlet transform to compress HS images. The primary goal is to enhance the coding efficiency while minimizing storage and transmission requirements. The proposed compression algorithm is evaluated on four benchmarks against eight state of art other compression algorithms on three performance metrics named coding efficiency, coding memory and coding complexity. In addition, it has low encoding/decoding time than other compression algorithm. From the simulation result, it has been clear that proposed compression algorithm has 2% to 4% increase in coding efficiency compared to the other state of art compression algorithms.

Keywords—Compression Algorithm, Wireless Sensor, Energy Efficient ; Transform Coding ; Set Partitioned

I. INTRODUCTION

Hyperspectral (HS) imaging also known as HSI represents a significant advancement over traditional imaging methodologies by capturing a contiguous spectrum of wavelengths (400 nm to 2500 nm) for each pixel within a scene, extending from the visible to the near-infrared region [1-2]. Unlike conventional techniques that record data in broad primary colors, HSI decomposes the electromagnetic spectrum into numerous narrow bands, thereby generating a detailed spectral signature that enables the identification of materials based on their unique physicochemical properties and the detection of subtle alterations imperceptible to other modalities [3]. This powerful analytical technique has since become indispensable across a diverse array of research and industrial fields, including precision agriculture for monitoring crop health, remote sensing for environmental mapping, biomedical imaging for cancer detection, and security applications for counterfeit detection, underscoring its broad utility and transformative potential [4-5].

A common structural analogy equates hyperspectral data cubes to video sequences, with wavelength serving as a counterpart to time [5]. Nevertheless, the statistical properties of these two data types differ significantly. Video data exhibits high inter-frame correlation due to object motion, whereas hyperspectral bands are correlated through smooth, material-dependent spectral responses, with no spatial displacement between bands [6].

A single HS image has size of ~150 MB or more and it required a lot of memory space to save the multiple HS images [7]. Moreover with the limited onboard memory, transmission of capture HS image data requires data transmission speed and browsing time. Thus, an effective compression algorithm is required to save the coding memory, reduce complexity and lower down data browsing time [4].

Recent time many different type of HyperSpectral Image Compression Algorithms (HSICAs) are proposed which includes predictive coding, vector quantization based, machine learning based, tensor based and transform based compression algorithms [4]. Among them, transform based compression algorithm had better performance than the other type of compression algorithm. Mathematical transform based Set Partition Hyperspectral Image Compression Algorithm has superiority to the other compression algorithms because of different properties named embeddedness (decoding process can be perform in lower bit rates than encoding process), low complexity, high coding efficiency and little coding memory requirements [5].

II. RELATED WORK

In the last couple of decades, there are many compression algorithms are proposed. Among of them, mathematical transform-based compression algorithm works with the both type of compression named as lossy and lossless compression. With the reference to the performance of different metrics, set partitioned compression algorithms are preferred due to the it's properties of embeddedness, complexity (easy to implement) and high coding efficiency [8-9]. There are many mathematical transforms such as cosine transform, wavelet transform, curvelet transform, shearlet transform, contourlet transform [10] etc are used in the convert the HS image from time domain to frequency domain. Wavelet transform has high decorrelation property and whole energy is pack in few coefficients [11].

Mathematical transform based set partitioned hyperspectral image compression algorithms are the special type of compression algorithm which utilize the property of wavelet transform [10]. It is done through the set structure of the wavelet transform HS image. These compression algorithm runs from top bit plane to the last bit plane or till the bit budget is available with the compression algorithms. It has been known that at the top bit planes, there are large number of insignificant coefficients. These coefficients are representing as a group through the single bit [4]. Through this way, a lot of coding memory had been saved. Moreover, through this simple process, coding complexity is also low

979-8-3315-8478-8/26/$31.00 ©2026 IEEE

which make these compression algorithms as an optimum choice for resource constraint HS image sensors [11-12].

Set partitioned HSICAs can be classified on the basis of the process of partition rule or use of lists (or requirement of coding memory) [4,13]. On the process of partition rule, compression algorithms can de organized into three groups named as zero block coding, zero tree coding and zero block tree coding [9]. While on the basis of lists, it can be grouped into the three groups named as list based, listless (or marker) and array based. 3D-Set Partitioned Embedded bloCK (3D-SPECK) [14] and 3D-Listless SpecK (3D-LSK) [15] belongs to zero block based compression algorithm in which transform HS image is split into the continuous blocks. Any block is called zero block if it is insignificant to the current bit plane and it is partitioned into the same size of small blocks if there is any significant coefficient. 3D-Set Partitioning in Hierarchical Trees (3D-SPIHT) [16] and 3D-No List Spiht (3D-NLS) [17] belongs to the zerotree based compression algorithms in which transform HS image coefficients are grouped to form the spatial orientation tree. A spatial orientation tree is zerotree if there is no coefficient found to the current threshold. 3D-Wavelet Block Tree Coding (3D-WBTC) [18] and 3D- Low Memory Cube Tree Coding (3D-LMBTC) [19] belongs to the zero block tree based compression algorithms in which transform HS image is partitioned into the continuous blocks and then block trees are created with the roots in the topmost sub-band in a zero tree fashion.

It has been known that listless compression algorithms such as 3D-LSK [15] and 3D-NLS [17] had fixed memory demand depend only on the size of HS image [10] while the demand of coding memory for list based compression algorithms such as 3D-SPECK [12] and 3D-SPIHT [14] is rapidly increased for with the higher coding rate and it also get slow (high coding complexity) due to the multiple read/write operations associated with the compression algorithm [17]. Listless compression algorithms used different type of markers or state table to define the status of the coefficient/sets. The demand of coding memory is also depended upon the efficient use of the markers as multiple type of markers increase the requirement of coding memory and also increase the coding complexity of the algorithm. Array based compression algorithm such as 3D-Block Partitioning Embedded Coding (3D-BPEC) [20] uses the unidirectional array instead of markers or lists to define the state of the sets or coefficients.

In the past many wavelet transform based compression algorithms (listless) are proposed which have low coding memory demand or reduce complexity [19]. But these has low coding efficiency. The issue can be addressed by using the advance wavelet transform. In past curvelet transform and contourlet transform had been applied with the zeroblock coding based HSICA (listless version) and it has been observed that there is a improvement in the coding efficiency of the compression algorithm. It is due to the effective representation of the edges or curves (discontinuities) present in the HS image. Effective representation means lower down requirement of the coefficients to represent the curve/edges which let increase the coding efficiency [10].

The contourlet transform, an extension of the wavelet transform, has multiresolution, localization, directionality, critical sampling, and anisotropy properties [21]. The contourlet transform extends the curvelet transform by incorporating principles of human visual perception. This synthesis enables the effective representation of image contours possessing diverse elongated shapes and a multitude of orientations [22-23].

III. 3D-Contourlet Transform Based LOW MEMORY Block Tree Coding

3D Contourlet Transform Based Low Memory Cube Tree Coding (3D-CT-LMBTC) is a listless compression algorithm which uses markers to define the significant/insignificant sets/coefficients. The 3D-CT-LMBTC algorithm is a listless variant of the 3D-WBTC [18] technique, which supersedes its predecessor's data-dependent linked lists with a fixed-size Static memory for Insignificant Block Cube Tree (SIBCT), while retaining the same set partitioning rules.

The proposed compression scheme employs a fixed-size static memory to track its partition rules, which enhances processing speed and reduces memory requirements. This approach maintains embedded bit-stream preservation and coding efficiency while utilizing designated markers within a two-bit per coefficient state table to explicitly track the set partitions. The state table utilizes a four-value marker to delineate the status of block cube trees within the partitioning hierarchy: a value of '0' indicates a tree not yet subjected to the significance test named as '1' and '2' categorize trees as type 'A' or type 'B', respectively; and a value of '3' identifies a node, which can be either type, that was found to be significant during early passes of the algorithm.

The 3D-CT-LMBTC compression algorithm is a bit-plane encoder that processes data from the most significant bit-plane downwards, terminating either upon reaching the least significant bit-plane or when a predefined bit budget is exhausted. The algorithm commences with the block cubes in the LLL band, testing each against the current threshold and encoding their significance. Block cubes that become newly significant or are slated for refinement are then iteratively partitioned into eight adjacent sub-blocks via octa-partitioning. This recursive partitioning continues until no further subdivision is required or a minimum block cube size of '2 x 2 x 2' (comprising eight coefficients) is attained.

The proposed 3D-LMBTC scheme is a bit-plane encoder that processes data from the most significant bit-plane downwards, terminating either upon reaching the least significant bit-plane or when a predefined bit budget is exhausted. The algorithm commences with the block cubes in the LLL band, testing each against the current threshold and encoding their significance. Block cubes that become newly significant or are slated for refinement are then iteratively partitioned into eight adjacent sub-blocks via octa-partitioning. This recursive partitioning continues until no further subdivision is required or a minimum block cube size of ‘2 x 2 x 2’ (comprising eight coefficients) is attained.

The 3D-CT-LMBTC algorithm processes significant block cube trees hierarchically. Upon identifying a significant type 'A' tree, its offspring are encoded immediately. If its grand-descendant set is also significant, eight new type 'A' trees are spawned by updating their state to '1' in the SIBCT, while the parent's state is marked '3' for testing in the current bit-plane. Conversely, a significant type 'A' tree with an insignificant grand-descendant set is reclassified as a type 'B' tree (state '2') for significance testing in the subsequent bit-plane. For a significant type 'B' tree, the grand-descendant set's significance is encoded, and if found significant, eight new type 'A' trees are generated, with their states updated from '0' to '1'. Decoding is performed by a symmetric decoder that mirrors the encoder's logic, with the additional step of identifying coefficients for refinement. The decoder uses the input bit-stream to set coefficient bits and signs, performs mid-tread dequantization for partially decoded coefficients, and finally reconstructs the 3D dataset via a linear index and an inverse 3D contourlet transform.

IV. SIMULATION RESULT

To validate the compression capabilities of the proposed HSICA, we conducted the simulation experiment on the two HS image datasets named as Washington DC Mall and Cuprite. Coding efficiency (Peak Signal to Noise Ratio as PSNR in dB), coding memory (in KB) and coding complexity (in execution time for encoding and decoding process as second) are employed as the evaluation metrics [10,14-19,24,28,29]. The HS image data was preprocessed by extracting a fixed-size '128 x 128 x 128' cube from the top-left corner of each image, employing zero-padding to meet the target size for any smaller images. To guarantee fairness in comparison, all experiments are performed under identical simulation experimental conditions.

A. *Coding Efficiency*

Coding Efficiency of any compression algorithm is measure by the PSNR in decibel (dB). For the effective compression (total reconstruction of image after compression process), the value of PSNR (lossless) should be '∞' and for the lossy compression, value of PSNR has numerical numbered [25-26].

A comparative analysis of coding efficiency at higher bit rates, detailed in Table 1, reveals that the PSNR performance of various listless algorithms, including the proposed 3D-CT-LMBTC which utilizes markers in lieu of lists, is nearly equivalent. This convergence in performance is attributed to the high-fidelity reconstruction of the hyperspectral image that is achievable at elevated bit rates [10,27].

B. *Coding Memory*

The coding memory footprint of listless compression algorithms is fixed and determined solely by the HS image size, whereas list-based algorithms exhibit variable memory demands that fluctuate with the target bit rate [30]. As illustrated in Table 1, the proposed algorithm requires marginally higher coding memory than proposed compression algorithm, 3D-LMBTC [19] and 3D-ZM-SPECK [28]. This is consistent with the characteristic of list-based compression algorithms, which generally maintain lower memory usage at low bit rates due to the smaller number of significant coefficients being processed.

C. *Coding Complexity*

Coding efficiency is quantified by the computational time required for compression, which encompasses both encoding and decoding phases [31]. The encoding process, which generates an embedded bit stream from the transformed hyperspectral image, is inherently more computationally intensive than decoding [32]. This disparity arises because encoding must repeatedly assess the significance of sets or coefficients across each bit plane, a step unnecessary during decoding. The proposed algorithm improves upon this metric by employing fewer markers than comparable methods like 3D-LSK [15] and 3D-NLS [17], thereby reducing the time spent on read/write operations. A comparative analysis of encoding and decoding times is provided in Table 1.

V. CONCLUSION

Present manuscript proposes a novel listless compression algorithm based on the contourlet transform. Evaluation on four hyperspectral images demonstrates that the proposed method significantly enhances coding efficiency and reduces computational complexity, achieving a higher PSNR than benchmark algorithms. Furthermore, memory requirements can be optimized by integrating the contourlet transform with frameworks like 3D-ZM-SPECK [28]. Beyond the wavelet and contourlet transforms, future work could explore the radon and shearlet transforms as promising alternatives for further advancing compression performance.

REFERENCES

[1] Vijay Joshi, J. Sheeba Rani "An On-Board Satellite Multispectral and Hyperspectral Compressor (MHyC): An Efficient Architecture of a Simple Lossless Algorithm", IEEE Transactions on Circuits and Systems I: Regular Papers, Volume 72, Issue 5, pp. 2167 - 2177, 2025. doi : 10.1109/TCSI.2025.3545127.

[2] Debaleena Datta, Pradeep Kumar Mallick, Akash Kumar Bhoi, Muhammad Fazal Ijaz, Jana Shafi, and Jaeyoung Choi "Hyperspectral image classification: Potentials, challenges, and future directions." Computational intelligence and neuroscience, Volume 2022, pp. 1-36, 2022. doi : 10.1155/2022/3854635.

[3] Yaman Dua, Vinod Kumar, and Ravi Shankar Singh. "Comprehensive review of hyperspectral image compression algorithms", Optical Engineering Volume 59, Issue 9, pp. 090902-090902, 2020. Doi : 10.1117/1.OE.59.9.090902.

[4] Shrish Bajpai, "3D-listless block cube set-partitioning coding for resource constraint hyperspectral image sensors." Signal, Image and Video Processing, Volume 18, Issue 4, pp. 3163-3178, 2024. doi : 10.1007/s11760-023-02979-0.

[5] Vinod Kumar Tripathi, Shrish Bajpai. "3D Single List Set Partitioning in Hierarchical Trees For Onboard Hyperspectral Image Sensors." SSRG International Journal of Electronics and Communication Engineering, Volume 12, Issue 4, pp. 265-281, 2025.

[6] R. Nagendran, Sudhir Ramadass, K. Thilagvathi, and Ananda Ravuri. "Neural Reinforcement-Oriented Hyperspectral Image Compression: Adaptive Approaches for Enhanced Quality." Chemometrics and Intelligent Laboratory Systems, Volume 266, pp. 105495. doi : 10.1016/j.chemolab.2025.105495.

[7] Huang, Xin, and Liangpei Zhang. "An adaptive mean-shift analysis approach for object extraction and classification from urban hyperspectral imagery." IEEE Transactions on Geoscience and Remote Sensing, Volume 46, Issue 12, pp. 4173-4185, 2008. doi : 10.1109/TGRS.2008.2002577.

[8] Dusselaar, Rui, and Manoranjan Paul. "Hyperspectral image compression approaches: opportunities, challenges, and future directions: discussion." Journal of the Optical Society of America A, Volume 34, Issue 12, pp. 2170-2180, 2017. doi : 10.1364/JOSAA.34.002170.

[9] Christophe, Emmanuel, Corinne Mailhes, and Pierre Duhamel. "Hyperspectral image compression: adapting SPIHT and EZW to anisotropic 3-D wavelet coding." IEEE Transactions on Image processing Volume 17, Issue 12, pp. 2334-2346, 2008. doi : 10.1109/TIP.2008.2005824.

[10] Shrish Bajpai, Divya Sharma, Monauwer Alam, Vishal Singh Chandel, Amit Kumar Pandey, and Suman Lata Tripathi. "Curvelet transform based compression algorithm for low resource hyperspectral image sensors." Journal of Electrical and Computer Engineering, Volume 2023, pp. 8961271, 2003. doi : 10.1155/2023/8961271.

[11] Emmanuel Christophe, Corinne Mailhes, and Pierre Duhamel. "Hyperspectral image compression: adapting SPIHT and EZW to anisotropic 3-D wavelet coding." IEEE Transactions on Image

processing Volume 17, Issue 12, pp. 2334-2346, 2008. doi : 10.1109/TIP.2008.2005824.

[12] Divya Sharma, Y. K. Prajapati, and R. Tripathi. "Success journey of coherent PM-QPSK technique with its variants: a survey." IETE Technical Review, Volume 37, Issue 1, pp. 36-55, 2020. doi : 10.1080/02564602.2018.1557569.

[13] Nadia Zikiou, Mourad Lahdir, and David Helbert. "Support vector regression-based 3D-wavelet texture learning for hyperspectral image compression." The Visual Computer, Volume 36, Issue 7 pp. 1473-1490, 2020. doi : 10.1007/s00371-019-01753-z.

[14] Xiaoli Tang, William A. Pearlman. "Lossy-to-lossless block-based compression of hyperspectral volumetric data." 2004 International Conference on Image Processing, 2004. ICIP'04. Vol. 5. IEEE, 2004. doi : 10.1109/ICIP.2004.1421815.

[15] Ruzelita Ngadiran, Said Boussakta, Bayan Sharif, and Ahmed Bouridane "Efficient implementation of 3D listless SPECK." International Conference on Computer and Communication Engineering (ICCCE'10). IEEE, pp. 1-5, 2010. doi : 10.1109/ICCCE.2010.5556843.

[16] Xiaoli Tang, William A. Pearlman. "Three-dimensional wavelet-based compression of hyperspectral images." Hyperspectral data compression. Boston, MA: Springer US, pp. 273-308, 2006. doi : 10.1007/0-387-28600-4_10.

[17] V. K. Sudha, R. Sudhakar, "3D listless embedded block coding algorithm for compression of volumetric medical images", J Sci Ind Res, Volume 72, pp. 735–748, 2013.

[18] Shrish Bajpai, Naimur Rahman Kidwai, Harsh Vikram Singh. "3D wavelet block tree coding for hyperspectral images." International Journal of Innovative Technology and Exploring Engineering, Volume 8, Issue 6C, pp. 64-68, 2019.

[19] Shrish Bajpai, Naimur Rahman Kidwai, Harsh Vikram Singh, and Amit Kumar Singh "Low memory block tree coding for hyperspectral images." Multimedia Tools and Applications, Volume 78, Issue 19 pp. 27193-27209, 2019. doi : 10.1007/s11042-019-07797-6.

[20] Harshit Chandra, and Shrish Bajpai. "3D-Block Partitioning Embedded Coding for Hyperspectral Image Sensors." 2023 International Conference on Power, Instrumentation, Energy and Control (PIECON). IEEE, 2023. doi : 10.1109/PIECON56912.2023.10085841.

[21] M.N. Do, and M. Vetterli, "The Contourlet Transform: An Efficient Directional Multiresolution Image Representation," IEEE Transactions on Image Processing, vol. 14, no. 12, pp. 2091-2106, 2005. doi : 10.1109/TIP.2005.859376.

[22] Zainab N. Abdulhameed Al-Rawi, Haraa R. Hatem, and Israa H. Ali, "Image Compression Using Contourlet Transform," 2018 1st Annual International Conference on Information and Sciences (AiCIS), Fallujah, Iraq, pp. 254-258, 2018. doi : 10.1109/AiCIS.2018.00053.

[23] R. Eslami, and H. Radha, "Wavelet-Based Contourlet Transform and its Application to Image Coding," 2004 International Conference on Image Processing, Singapore, vol. 5, pp. 3189-3192, 2004. doi : 10.1109/ICIP.2004.1421791.

[24] Divya Sharma, "Image Quality Assessment Metrics for Hyperspectral Image Compression Algorithms." 2024 Second International Conference Computational and Characterization Techniques in Engineering & Sciences, pp. 1-5, 2024. doi : 10.1109/IC3TES62412.2024.10877577.

[25] Mohd Tausif, Ekram Khan, and Antonio Pinheiro. "Computationally efficient wavelet-based low memory image coder for WMSNs/IoT." Multidimensional Systems and Signal Processing, Volume 34, Issue 3, pp. 657-680, 2023. doi : 10.1007/s11045-023-00878-8.

[26] Mohd Tausif and Ekram Khan. "Image Coding of Natural and Light Field Images: A Tutorial." IETE Journal of Education, Volume 66, Issue 1, pp. 25-34, 2025. doi : 10.1080/09747338.2025.2473577.

[27] Mohd Tausif, and Ekram Khan. "Integration of Artificial Intelligence and Machine Learning for wearable electronics." Sustainable Electronics. CRC Press, 2025. 260-283. doi : 10.1201/9781003635789-11.

[28] Shrish Bajpai, Naimur Rahman Kidwai, Harsh Vikram Singh, and Amit Kumar Singh, "A low complexity hyperspectral image compression through 3D set partitioned embedded zero block coding." Multimedia Tools and Applications, Volume 81, Issue 1, pp. 841-872, 2022. doi : 10.1007/s11042-021-11456-0.

[29] Shrish Bajpai, "Low complexity block tree coding for hyperspectral image sensors." Multimedia Tools and Applications, Volume 81, Issue 23, pp. 33205-33232, 2022. doi : 10.1007/s11042-022-13057-x.

[30] Xiaoli Tang, Sungdae Cho, and William A. Pearlman. "3D set partitioning coding methods in hyperspectral image compression." Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429). Vol. 2. IEEE, 2003. doi : 10.1109/ICIP.2003.1246661.

[31] R. Nagendran, and A. Vasuki. "Hyperspectral image compression using hybrid transform with different wavelet-based transform coding." International Journal of Wavelets, Multiresolution and Information Processing, Volume 18, Issue 01, pp. 1941008, 2020. doi : 10.1142/S021969131941008X.

[32] Vijay Joshi, "An Efficient FPGA Implementation of a Simple Lossless Algorithm (SLA) for On-board Satellite Hyperspectral Data Compression." 2024 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2024. doi : 10.1109/ISCAS58744.2024.10558039.

Table 1 : Simulation results for two HS images on four performance metrics applied on eight state of art HSICAs and proposed HSICA

Bit Rate	3D-SPECK [14]	3D-SPIHT [16]	3D-WBTC [18]	3D-LSK [15]	3D-NLS [17]	3D-LMBTC [19]	3D-ZM-SPECK [28]	3D-LCBTC [29]	3D-CT-LMBTC (Proposed HSICA)	3D-SPECK [14]	3D-SPIHT [16]	3D-WBTC [18]	3D-LSK [15]	3D-NLS [17]	3D-LMBTC [19]	3D-ZM-SPECK [28]	3D-LCBTC [29]	3D-CT-LMBTC (Proposed HSICA)
	Coding Efficiency (Peak Signal to Noise Ratio)																	
	Washington DC Mall									Cuprite								
0.1	38.53	38.28	38.50	38.35	38.12	38.29	38.33	38.31	32.37	25.64	24.67	25.77	25.65	24.61	25.60	25.79	25.49	25.78
0.2	41.54	41.34	41.52	41.49	41.27	41.19	41.42	41.59	41.3	30.92	29.44	31.03	30.88	29.33	30.77	30.87	30.84	31.02
0.3	43.51	43.30	43.49	43.55	43.30	43.48	43.57	43.58	43.58	34.55	33.36	34.58	34.55	33.27	34.42	34.59	34.61	34.95
0.4	45.26	45.11	45.25	45.09	45.09	44.59	45.24	45.28	44.74	38.05	37.04	38.15	38.05	36.97	37.50	38.16	38.18	38.41
0.5	46.81	46.60	46.81	46.76	46.41	46.09	46.73	46.83	46.47	41.27	40.51	41.37	41.32	40.45	41.17	41.26	41.39	41.55
0.6	48.45	48.24	48.43	48.42	48.21	48.38	48.39	48.49	48.61	43.46	42.58	43.57	43.47	42.50	43.36	43.43	43.52	43.79
0.7	49.76	49.53	49.74	49.73	49.50	49.17	49.69	49.78	49.99	45.55	45.00	45.81	45.78	44.89	45.60	45.68	45.83	46.07
0.8	51.12	50.84	51.09	51.07	50.76	50.28	50.97	51.17	51.38	47.12	46.43	47.26	47.07	46.38	47.03	47.11	47.16	47.38
0.9	52.24	52.06	52.22	52.24	52.06	51.67	52.12	52.26	52.48	48.74	47.95	48.85	48.75	47.91	48.66	48.78	48.91	49.12
1	53.52	53.32	53.51	53.49	53.33	53.46	53.47	53.59	53.79	49.83	49.24	49.98	49.86	49.22	49.68	49.71	50.01	50.33
	Coding Memory																	
	Washington DC Mall									Cuprite								
0.1	243.8	263.3	250.1	512	1024	12	0	300.59	12	277.7	277.6	282.8	512	1024	12	0	300.59	12
0.2	416.3	438	416	512	1024	12	0	300.59	12	414.5	434.3	417.2	512	1024	12	0	300.59	12
0.3	701.1	628.6	704	512	1024	12	0	300.59	12	544.3	514.7	546.3	512	1024	12	0	300.59	12
0.4	733.8	723.6	733	512	1024	12	0	300.59	12	601.9	576.5	594.5	512	1024	12	0	300.59	12
0.5	1048.8	1060.5	1049	512	1024	12	0	300.59	12	671.2	701.9	674.5	512	1024	12	0	300.59	12

0.6	1191.1	1222.6	1195.4	512	1024	12	0	300.59	12	854	783.7	857.6	512	1024	12	0	300.59	12
0.7	1277.3	1302.1	1281.7	512	1024	12	0	300.59	12	947.5	865.5	971.7	512	1024	12	0	300.59	12
0.8	1407.7	1415.3	1404.4	512	1024	12	0	300.59	12	1065.3	964.6	1057	512	1024	12	0	300.59	12
0.9	1702.5	1725.5	1704.6	512	1024	12	0	300.59	12	1158	1182	1159	512	1024	12	0	300.59	12
1	1802.5	1826.7	1724.6	512	1024	12	0	300.59	12	1286	1308	1292	512	1024	12	0	300.59	12
	Coding Complexity (Encoding Time)																	
	Washington DC Mall									Cuprite								
0.1	25	7.5	6.50	0.80	0.91	3.9	1.78	0.76	3.77	17.3	6.3	4.7	0.9	1.12	3.2	1.78	0.86	3.02
0.2	57.9	25.8	24.8	1.10	1.21	5.1	2.81	1.04	4.97	55.8	26	16.6	1.2	1.54	6.8	3.01	1.09	6.66
0.3	92.1	37.5	32	1.50	1.65	7.7	3.68	1.41	7.58	107.9	45.5	39.1	2	2.27	7.1	4.08	1.92	6.89
0.4	269.7	117.9	195.5	2.00	2.12	9.7	5.69	1.93	9.52	182.3	75.6	68.2	2.1	2.41	9.2	5.21	2.07	8.94
0.5	414.8	140.1	211.2	2.50	2.64	11.3	7.41	2.44	11.05	276.1	95.4	93.3	2.2	2.58	11.1	6.32	2.11	10.87
0.6	576	166.4	247.9	2.90	3.02	13.3	7.99	2.82	13.08	298.4	161.7	155.7	3.4	3.61	12.9	7.55	3.24	12.71
0.7	887.5	405.7	625	3.20	3.37	18.1	9.66	3.13	17.92	438.8	179.2	202.2	3.9	4.21	15	8.76	3.79	14.74
0.8	1130.5	474.2	710.2	3.80	3.96	20	9.91	3.85	19.74	558.7	198.5	358.5	4.2	4.48	16.5	9.66	4.02	16.35
0.9	1334.6	555.7	746	4	4.14	20.6	12.53	4.04	20.36	656.1	282.8	371	4.4	4.69	18.1	11.20	4.12	17.88
1	1497.5	575	804	4.41	4.57	21.1	13.21	4.38	20.75	905.1	364	652.5	5	5.23	20.3	15.89	5.02	20.06
	Coding Complexity (Decoding Time)																	
	Washington DC Mall									Cuprite								
0.1	17.40	6.10	5.00	0.70	0.79	2.3	1.71	0.64	2.14	13.40	5	3.10	0.70	0.94	2.2	1.70	0.63	2.01
0.2	48.80	24.80	22.50	1.07	1.07	3.3	2.71	1.01	3.18	46.70	22.10	14.60	1.00	1.32	4.8	2.92	0.91	4.54
0.3	75.40	34.80	28.50	1.45	1.43	4.9	3.59	1.33	4.69	93.70	40.20	35.40	1.80	2.04	5.5	3.98	1.69	5.32
0.4	264.2	106.3	180.4	1.70	1.94	8.1	5.41	1.62	5.22	162.5	70.10	65.80	1.90	2.31	7	5.07	1.81	6.71
0.5	339.1	135.4	191.7	2.20	2.31	7.7	6.82	2.07	7.54	236.1	88.30	91.50	2.00	2.45	8.5	6.01	1.92	8.12
0.6	532.4	149.6	244.6	2.60	2.79	9.8	7.95	2.48	9.57	281.2	160.9	149	2.90	3.38	10.2	7.17	2.79	9.94
0.7	807.6	327.1	558	2.70	3.04	11.6	8.80	2.53	11.42	435	175.8	196.8	3.10	4.03	11.9	8.28	2.97	11.65
0.8	1058.1	448.9	675.3	3.10	3.67	13.4	9.38	3.09	13.19	525.9	195.3	316	3.80	4.24	13.1	9.21	2.61	12.87
0.9	1142.3	486.2	725	3.20	3.93	13.6	11.82	3.33	13.31	599.2	273.5	366.9	4.00	4.47	15	10.34	3.83	14.82
1	1289.7	504	774	3.70	4.24	15.5	12.31	3.87	15.25	884.4	346.6	596	4.50	5.07	15.9	14.03	4.38	15.69