3D-Block Partitioning Embedded Coding for Hyperspectral Image Sensors

2023 International Conference on Power, Instrumentation, Energy and Control (PIECON) | 979-8-3503-9976-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/PIECON56912.2023.10085841

Harshit Chandra
ECE Department, Faculty of Engineering and Information Technology, Integral University, Lucknow, Uttar Pradesh, India
E-Mail : harshitchandra55@gmail.com

Shrish Bajpai
ECE Department, Faculty of Engineering and Information Technology,
Integral University, Lucknow, Uttar Pradesh, India
E-Mail : shrishbajpai@gmail.com

Abstract—**Computational complexity with the coding efficiency of any hyperspectral image sensor is a challenging issue. 3D-SPIHT has reasonable complexity and generates an embedded bit-stream. Due to the linked lists, the processing of the compression through the 3D-SPIHT gets slow at the high bit rates. This manuscript presents a low-complexity version of 3D-SPIHT which uses the array structure instead of linked lists. These arrays are independent of each other. The coding memory required by the arrays is the same as the linked list used in 3D-SPIHT. Through the use of array, the coding complexity is reduced and coding efficiency is increased with the use of parity of the bit plane. Thus, the proposed compression scheme is an optimum solution for the resource-constraint hyperspectral image sensor.**

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

I. INTRODUCTION

Hyperspectral (HS) image represents the target scene in an electromagnetic spectrum of continuous and narrow bands having rich spectral and spatial information at the pixel-level resolution [1]. This high-dimensional data has massive information, used in multiple applications ranging from agriculture, mining, diagnosis and surgery, geology, remote sensing, etc [2]. This huge image data needs to transmit to the earth stations for analysis and it requires an ample amount of data bandwidth, processing time, memory and sensor power. The HS image compression has generated considerable attention due to its importance in saving onboard memory, data transmission bandwidth with reduced coding complexity. But, the compression of the HS images increases the coding process [3]. The HS image is also 3D data and it is different to the video data. The first two dimensions in HS image and video is spatial dimensions while the third dimension in the HS image is related to the wavelength and in the video, it is related to the time (temporal) [4]. An ideal HyperSpectral Image Compression Scheme (HSICS) has properties of low coding complexity, high coding gain, low coding memory requirement and can work for the lossy, near lossless and lossless compression process.

The HSICSs are subdivided into six categories in vector quantization-based compression scheme, prediction-based compression scheme, transform-based compression scheme, compressive sensing-based compression scheme, machine learning-based compression scheme and hybrid compression scheme [4-8]. Among the above transform based compression schemes give an optimum performance than other compression schemes. The HS image is transformed into the frequency domain using the wavelet and cosine mathematical transform. Among above, wavelet transform gives a high performance than any other mathematical transform. The compression scheme uses the properties of the transform to achieve compression [5].

To address the problems of HS image compression (complexity and coding efficiency), a novel compression scheme 3D-Block Partitioning Embedded Coding (3D-BPEC) is proposed, where linked lists are removed by the simple arrays to reduce the multiple read/write operation. Through this, the overall complexity is reduced significantly.

The remaining parts of the manuscript are structured as follows. In Section II overview of the transform approach to hyperspectral image compression and the compression scheme proposed is illustrated; in Section III, the experimental results are presented and a discussion is given in Section IV. Section V summarizes the main concluding remark.

II. RELATED WORK

The wavelet transform-based set partitioned HSICS is a special type of transform HSICS that uses the properties (energy compaction and decorrelation property) of the wavelet transform to achieve the compression of HS image. 3D-EBCOT, 3D-SBHP utilized the above properties of wavelet transform but required large run time and coding memory [4]. The wavelet transform based set partitioned HSICS 3D-SPECK [9] & 3D-SPIHT [10] uses the set structure of the transform image to achieve the compression with the generation of the embedded output bit stream. As it is known that there is lot of insignificant coefficients at the higher bit planes. 3D-SPECK [9] uses the block cube structure and 3D-SPIHT [10] uses the tree structure to arrange the insignificant coefficients. 3D-WBTC shows the high coding gain at the low bit rates as it uses the block cube tree structure [11]. The list based HSICSs have high coding complexity (due to the linked lists). The complexity is reduced by removing the lists and use of markers to track the significance of the sets or coefficients [11]. 3D-LSK [12], 3D-NLS [13] and 3D-LMBTC [14] are the listless version of the 3D-SPECK, 3D-SPIHT and 3D-WBTC [9-11]. The coding efficiency is reduced due to the use of markers [15-16].

III. 3D BLOCK PARTITIONING EMBEDDED CODING

The proposed compression scheme 3D-Block Partitioning Embedded Coding uses the simple 1D array structure instead of linked lists or markers or state table tables for the tracking of the significance of the sets or coefficients. It also utilized the property of parity during the bit plane pass. It employs the same tree structure as 3D-SPIHT [10] to achieve the compression of the HS image. Due to the embeddedness property of the 3D-BPEC, it works with lossy and lossless compression. All block cubes are

979-8-3503-9976-9/23/$31.00 ©2023 IEEE

coded according to the HS image bit plane where the top most bit plane coded first and rest bit planes coded according to their priority. The range of resolution levels, from the lowest to the highest bit plane, each bit plane is scanned. For any wavelet transform level $1 \leq \lambda \leq L$ The lowest sub-band LLL_λ is scanned first followed by LLH_λ, LHL_λ, LHH_λ, HLL_λ, HLH_λ, HHL_λ and HHH_λ (L is the level for the wavelet transform). Each sub band is split into the continuous smaller block cubes of size 'η x η x η' in each sub-band and 'η' should be in the power of 2. These smaller block cubes should be confine in the sub-bands. Though these smaller block cubes, the coding scheme produces a rate scalable output bit stream and individual bit planes are arrange in the rising order of the resolution levels.

The 3D-BPEC computes the maximum bit plane for each smaller block cubes define as P_{max}^{z} as in Eq 1 with $1 < z < Z$ where 'z' is define as the block cube sequence while 'Z' is the total number of block cubes present in the transform HS image. The P_{max}^{z} is store in array named as 'maxbp' for each block cube. P_{max} is computed for entire transform HS image as in Eq 2 and it is available to the encoder and decoder end.

$$P_{max}^{z} = \lfloor \log_2[\max(\max(\max(T)))] \rfloor \qquad 1$$

$$P_{max} = \max(P_{max}^{z}) \qquad 2$$

For any coefficient 'C_{ijk}' or block cube 'B_z' of transform HS image found significant (SIG) for against bit plane 'n' when

$$2^n < |C_{ijk}| < 2^{n+1} \qquad 3$$

$$2^n < |B_z| < 2^{n+1} \qquad 4$$

Otherwise it will be consider as insignificant (ISIG). The unique features that distinguish the proposed 3D-BPEC from all state of art HSICS are that it takes into account the parity of the bit-plane coding passes and the use of 1D array instead of random accessed linked lists. Like 3D-SPIHT [10], 3D-BPEC uses the same list but in different fashion. The list of significant pixel (LSP) is replaced by the 1D array named as Significant Coefficient Array (SCA). This is because coefficients which are present in the SCA generate the refinement bit for the next bit planes and bit generated in one sequence. The list of insignificant set (LIS) had been replaced by the two 1D array named as Odd Parity Insignificant Set Array (OPISA) and Even Parity Insignificant Set Array (EPISA). OPISA and EPISA stores the ISIG sets which are generated in the last bit plane and will be tested for the next bit plane. In the same way list of insignificant pixel (LIP) had been replaced by the two 1D array named as Odd Parity Insignificant Coefficient Array (OPICA) and Even Parity Insignificant Coefficient Array (EPICA). OPICA stores coefficients that are found ISIG during an odd pass to be tested during the next even pass while EPICA stores coefficients that are found ISIG during an even pass to be tested during the next odd pass. The OPISA, EPISA, OPICA and EPICA work as first in first out (FIFO) fashion.

The encoding process of the proposed HSICS is divided into two passes: Initialization Pass & Coding pass. The coding pass is sub divided into four sub pass as Test Insignificant Coefficient Pass (TICP), Test Insignificant Set Pass (TISP), Code Significant Block Cube Pass (CSBCP) & Refinement Pass (RP).

Initialization Pass : The compression scheme starts with the calculation of the highest bit plane for each block cube, saving them in the 1D array 'maxbp' while the top bit plane is calculated as in Eq 2. Five more 1D arrays (SCA, OPISA, EPISA, OPICA and EPICA) are also initialized as empty arrays. This pass runs only one time at the start of the encoding process.

Coding Pass : After the processing of the initialization pass, the coding pass process as the sequence of TICP, TISP, CSBCP & RP. This pass runs till the last bit plane or bit budget is available.

a. **Test Insignificant Coefficient Pass** : This pass scan all the coefficients present in the OPICA and EPICA array according to the parity. If the bit plane is odd, then OPICA will be reinitialized again as an empty array. The coefficient present in the EPICA is tested for significance. If it found SIG, then it is coded with the sign bit and the coefficient is transferred to the end of the SCA. If the coefficient is ISIG, it is added to the end of the OPICA. The same procedure is repeated for the even bit plane but the arrays are interchanged.

b. **Test Insignificant Set Pass** : This pass scan sets for all sets present in the OPISA and EPISA. This sub pass work in the same way as TICP works. For the odd bit plane pass, OPISA is reinitialized again as an empty array. The set present in the EPISA is tested for the SIG against the current bit plane. If set is found SIG then it is octa tree partitioning (OTP) and new sets or coefficients are generated. These new sets or coefficients are tested for the SIG. If set is ISIG then it will added in the end of the OPISA. The same procedure is repeated for the even bit plane but the arrays are interchanged.

c. **Code Significant Block Cube Pass** : This sub pass tested the significance for the small block cube. Each block cube B_z is tested for significance. If block cube is significant, bit '1' generates and it will be octa partitioned into the eight new block cubes having the half dimension to the parent block cube.

d. **Refinement Pass** : A refinement bit is generated to all coefficients who are significant in the previous bit planes. The coefficient is refined to more bit precision by outputting it's n^{th} MSB to the bit stream against the n^{th} bit plane. After that the current threshold is reduced by the half and coding bit plane run till the bit budget exhausts.

IV. EXPERIMENT RESULT & ANALYSIS

The effectiveness of the suggested compression strategy is demonstrated by comparison with other cutting-edge HSICSs 3D-SPECK (HSICS 1), 3D-SPIHT (HSICS 2), 3D-WBTC (HSICS 3), 3D-LSK (HSICS 4), 3D-NLS (HSICS 5), 3D-LMBTC (HSICS 6), 3D-ZM-SPECK (HSICS 7) and 3D-LCBTC (HSICS 8) [9-16]. The two standard HS images Washington DC (HS Image I), and Cuprite (HS Image II) are taken for the performance evaluation. The HSICAs are run on the same hardware (11th generation i5 processor of 2.4

GHz with 20 GB RAM) and software platform (operating system as Windows 11).

The 5 level wavelet transform is applied on the HS image cube. The transform HS image is quantized to the integer and converted to the 1D array through the linear indexing scheme [16]. The computational complexity is calculated through the encoding and decoding time while the coding efficiency is calculated in the terms of Peak Signal-to-Noise Ratio (PSNR) [17-19].

Coding Complexity : The coding complexity is measured as the time required for the computation of mathematical and logical operations by the proposed compression scheme. The time needed by the encoder is known as encoding time while the time require by the decoder to decoder the receive bit stream is known as the decoding time. The encoding time is always greater than the decoding time because encoder needs to check the significance of each set or coefficient which was insignificant in last bit plane. Table 1 gives the detailed description of the encoding time while Table 2 covers the decoding time required by the different HSICSs. It has been clear from Table 1 and Table that the proposed compression scheme performs better to the other list based HSICS [9-11]. Due to the use of array saves the multiple read/write operation. But, it performance degraded when it get to compare with the listless HSICSs [12-18].

Coding Efficiency : The amount of bits necessary to obtain the desired quality of the reconstructed HS picture is used to calculate the coding efficiency. It is measured mathematically as in the terms of PSNR as in Eq 5 while the associated mean square error (MSE) is calculated as in Eq 6.

$$PSNR = 20\ \log_{10}\left[\frac{Signal\ Amplitude_{\ max}}{MSE}\right] \qquad 5$$

$$\text{MSE} = \frac{1}{\lambda}\sum_{a}\sum_{b}\sum_{c}[A(\alpha,\beta,\gamma) - B(\alpha,\beta,\gamma)]^2 \qquad 6$$

The original HS image is defined as A(α,β,γ) while the reconstructed HS image after the compression process is defined as B(α,β,γ) . The total number of pixels present in HS image is defined as λ.

It has been observed from Table 3 the coding efficiency of the proposed HSICS and 3D-SPECK [9] varies from -0.05 dB to 0.07 dB for HS Image I & -0.71 dB to -0.06 dB for HS Image II. In the same way, the coding efficiency of the proposed HSICS and 3D-SPIHT [10] varies from 0.15 dB to 0.28 dB for HS Image I & 0.37 dB to 0.97 dB for HS Image II. The coding efficiency variation between the proposed HSICS and 3D-WBTC [11] varies from -0.04 dB to 0.09 dB for HS Image I & - 0.16 dB to - 0.74 dB for HS Image II.

V. CONCLUSION

The proposed HSICS is a special type of HS image compression scheme which uses 1D simple array for the tracking of the significance of the coefficient/sets. The use of array (SCA, OPISA, EPISA, OPICA, EPICA and maxbp) reduces the number of read/write operation (multiple access to the associated linked lists) through which complexity reduces significantly. The use of the parity process increase the coding gain but the coding gain can also be increase through the use of curvelet or countourlet transform. These transform gives the optimum performance in the edges or corners of objects present in the HS image. The coding complexity further reduces through use of the block tree coding structure as it gives the superior performance than the block tree or block cube structure. The coding complexity can be reduced further by the use of the parallel processing (dividing the HS image into the equal size cube sets) of the image compression process.

ACKNOWLEDGEMENT

The IU/R&D/2022-MCN0001733 is the manuscript communication number for this article.

REFERENCES

[1] Swalpa Kumar Roy, Ankur Deria, Danfeng Hong, Muhammad Ahmad, Antonio Plaza, and Jocelyn Chanussot, "Hyperspectral and LiDAR Data Classification Using Joint CNNs and Morphological Feature Learning." IEEE Transactions on Geoscience and Remote Sensing, 2022. doi : 10.1109/TGRS.2022.3177633.

[2] Shrish Bajpai, Harsh Vikram Singh and Naimur Rahman Kidwai, "Feature extraction & classification of hyperspectral images using singular spectrum analysis & multinomial logistic regression classifiers", IEEE International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), 97-100, 2017. doi : 0.1109/MSPCT.2017.8363982.

[3] Reaya Grewal, Singara Singh Kasana, Geeta Kasana, "Hyperspectral image segmentation: a comprehensive survey" Multimedia Tools and Applications, 2022. doi : 10.1007/s11042-022-13959-w.

[4] Y Dua, V Kumar and RS Singh, "Comprehensive review of hyperspectral image compression algorithms", Optical Engineering, Vol. 59, Issue 9, 090902, 2020. doi : 10.1117/1.OE.59.9.090902.

[5] Y Dua, V Kumar and RS Singh, "Parallel lossless HSI compression based on RLS filter", Journal of Parallel and Distributed Computing, Vol. 150, Issue 9, 60-68, 2021. doi : 10.1016/j.jpdc.2020.12.004.

[6] V K Bairagi, A M Sapkal, M S Gaikwad, "The role of transforms in image compression", Journal of the institution of engineers (INDIA): Series B, 94(2), 135-140, 2013. doi : 10.1007/s40031-013-0049-9.

[7] Al-Janabi AK, Efficient and simple scalable image compression algorithms. Ain Shams Engineering Journal, 10(3), 463-470, 2019. doi : 10.1016/j.asej.2019.01.008.

[8] S Das, "Hyperspectral image, video compression using sparse tucker tensor decomposition", IET Image Process 15 (4), 964–973, 2021. doi : 10.1049/ipr2.12077.

[9] Tang Xiaoli, William A Pearlman, James W Modestino. "Hyperspectral image compression using three-dimensional wavelet coding" In Image and Video Communications and Processing, vol. 5022, 1037-1047, 2003.

[10] Tang Xiaoli, William A Pearlman, "Three-dimensional wavelet-based compression of hyperspectral images" In Hyperspectral data compression, 273-308, 2006, Springer, Boston, MA.

[11] S Bajpai, N R Kidwai, H V Singh, "3D wavelet block tree coding for hyperspectral images", International Journal of Innovative Technology and Exploring Engineering 8(6C):64–68, 2019.

[12] Ngadiran R, Boussakta S, Sharif B, Bouridane A, "Efficient implementation of 3D listless SPECK.", IEEE International Conference on Computer and Communication Engineering, 1-4, 2010. doi : 10.1109/ICCCE.2010.5556843.

[13] V K Sudha, R Sudhakar, "3D listless embedded block coding algorithm for compression of volumetric medical images", Journal of Scientific & Industrial Research, 72, 735-738, 2013.

[14] S Bajpai, N R Kidwai, H V Singh, A K Singh, "Low memory block tree coding for hyperspectral images" Multimedia Tools and Applications, 78(19), 27193-27209, 2019. doi : 10.1007/s11042-019-07797-6.

[15] S Bajpai, N R Kidwai, H V Singh, A K Singh, "A low complexity hyperspectral image compression through 3D set partitioned embedded zero block coding", Multimedia Tools and Applications, 81(1), 841-872, 2022. doi : 10.1007/s11042-021-11456-0.

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

[17] Dibyendu Barman, Abul Hasnat, Bandana Barman, "A quantization based codebook formation method of vector quantization algorithm to improve the compression ratio while preserving the visual quality of the decompressed image", Multidimensional Systems and Signal Processing, 1-19, 2022. doi : 10.1007/s11045-022-00856-6.

[18] D. Sharma, Y. K. Prajapati, and R. Tripathi. "Success journey of coherent PM-QPSK technique with its variants: a survey", IETE Technical Review, vol. 37, no. 1, pp. 36-55, 2020. doi : 10.1080/02564602.2018.1557569.

[19] D Venu, A V R Mayuri, S Neelakandan, G L N Murthy, N Arulkumar, Nilesh Shelke, "An efficient low complexity compression based optimal homomorphic encryption for secure fiber optic communication", Optik, 252, 168545, 2022. doi : 10.1016/j.ijleo.2021.168545.

[20] Andrew Campbell, Jaime Zabalza, Paul Murray, Stephen Marshall, Simon Malone, Hyperspectral imaging through partially transparent media. In Hyperspectral Imaging and Applications II, Vol. 12338, pp. 61-66, 2023. doi : 10.1117/12.2653629.

Table 1 : Analysis of the computational complexity of the different HSICSs with the 3D-BCPEC (Encoding Time)

Bit Rate	HSICS 1 [9]	HSICS 2 [10]	HSICS 3 [11]	HSICS 4 [12]	HSICS 5 [13]	HSICS 6 [14]	HSICS 7 [15]	HSICS 8 [16]	3D-BPEC	HSICS 1 [9]	HSICS 2 [10]	HSICS 3 [11]	HSICS 4 [12]	HSICS 5 [13]	HSICS 6 [14]	HSICS 7 [15]	HSICS 8 [16]	3D-BPEC
	HS Image I									HS Image II								
0.1	25	7.5	6.50	0.80	0.91	3.90	1.78	0.76	6.08	17.3	6.3	4.7	0.9	1.12	3.2	1.78	0.86	5.27
0.2	57.9	25.8	24.8	1.10	1.21	5.10	2.81	1.04	21.51	55.8	26	16.6	1.2	1.54	6.8	3.01	1.09	22.34
0.3	92.1	37.5	32	1.50	1.65	7.70	3.68	1.41	33.8	107.9	45.5	39.1	2	2.27	7.1	4.08	1.92	38.41
0.4	269.7	117.9	195.5	2.00	2.12	9.70	5.69	1.93	97.21	182.3	75.6	68.2	2.1	2.41	9.2	5.21	2.07	72.31
0.5	414.8	140.1	211.2	2.50	2.64	11.30	7.41	2.44	117.2	276.1	95.4	93.3	2.2	2.58	11.1	6.32	2.11	91.08
0.6	576	166.4	247.9	2.90	3.02	13.30	7.99	2.82	159.1	298.4	161.7	155.7	3.4	3.61	12.9	7.55	3.24	148.1
0.7	887.5	405.7	625	3.20	3.37	18.10	9.66	3.13	318.7	438.8	179.2	202.2	3.9	4.21	15	8.76	3.79	175.9
0.8	1130.5	474.2	710.2	3.80	3.96	20	9.91	3.85	548.2	558.7	198.5	358.5	4.2	4.48	16.5	9.66	4.02	187.8
0.9	1334.6	555.7	746	4	4.14	20.60	12.53	4.04	704.1	656.1	282.8	371	4.4	4.69	18.1	11.20	4.12	247.3
1	1497.5	575	804	4.41	4.57	21.10	13.21	4.38	774.9	905.1	364	652.5	5	5.23	20.3	15.89	5.02	487.2

Table 2 : Analysis of the computational complexity of the different HSICSs with the 3D-3D-BPEC (Decoding Time)

Bit Rate	HSICS 1 [9]	HSICS 2 [10]	HSICS 3 [11]	HSICS 4 [12]	HSICS 5 [13]	HSICS 6 [14]	HSICS 7 [15]	HSICS 8 [16]	3D-BPEC	HSICS 1 [9]	HSICS 2 [10]	HSICS 3 [11]	HSICS 4 [12]	HSICS 5 [13]	HSICS 6 [14]	HSICS 7 [15]	HSICS 8 [16]	3D-BPEC
	HS Image I									HS Image II								
0.1	17.40	6.10	5	0.70	0.79	2.30	1.71	0.64	5.51	13.40	5	3.1	0.7	0.94	2.2	1.70	0.63	4.74
0.2	48.80	24.80	22.50	1.07	1.07	3.30	2.71	1.01	19.08	46.70	22.10	14.6	1	1.32	4.8	2.92	0.91	19.81
0.3	75.40	34.80	28.50	1.45	1.43	4.90	3.59	1.33	30.45	93.70	40.20	35.4	1.8	2.04	5.5	3.98	1.69	33.18
0.4	264.2	106.3	180.4	1.70	1.94	8.10	5.41	1.62	88.59	162.5	70.10	65.8	1.9	2.31	7	5.07	1.81	63.91
0.5	339.1	135.4	191.7	2.20	2.31	7.70	6.82	2.07	102.8	236.1	88.30	91.5	2	2.45	8.5	6.01	1.92	84.25
0.6	532.4	149.6	244.6	2.60	2.79	9.80	7.95	2.48	138.4	281.2	160.9	149	2.9	3.38	10.2	7.17	2.79	139.2
0.7	807.6	327.1	558	2.70	3.04	11.60	8.80	2.53	287.2	435	175.8	196.8	3.1	4.03	11.9	8.28	2.97	161.8
0.8	1058.1	448.9	675.3	3.10	3.67	13.40	9.38	3.09	507.3	525.9	195.3	316	3.8	4.24	13.1	9.21	2.61	177.2
0.9	1142.3	486.2	725	3.20	3.93	13.60	11.82	3.33	679.1	599.2	273.5	366.9	4	4.47	15	10.34	3.83	231.9
1	1289.7	504	774	3.70	4.24	15.50	12.31	3.87	719.4	884.4	346.6	596	4.5	5.07	15.90	14.03	4.38	445.2

Table 3 : Analysis of the coding efficiency of the different HSICSs with the 3D-BCPEC (PSNR)

Bit Rate	HSICS 1 [9]	HSICS 2 [10]	HSICS 3 [11]	HSICS 4 [12]	HSICS 5 [13]	HSICS 6 [14]	HSICS 7 [15]	HSICS 8 [16]	3D-BPEC	HSICS 1 [9]	HSICS 2 [10]	HSICS 3 [11]	HSICS 4 [12]	HSICS 5 [13]	HSICS 6 [14]	HSICS 7 [15]	HSICS 8 [16]	3D-BPEC
	HS Image I									HS Image II								
0.1	38.53	38.28	38.50	38.35	38.12	38.29	38.33	38.31	38.54	25.64	24.67	25.77	25.65	24.61	25.60	25.79	25.49	25.04
0.2	41.54	41.34	41.52	41.49	41.27	41.19	41.42	41.59	41.52	30.92	29.44	31.03	30.88	29.33	30.77	30.87	30.84	30.41

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	33.84
0.4	45.26	45.11	45.25	45.09	45.09	44.59	45.24	45.28	45.31	38.05	37.04	38.15	38.05	36.97	37.50	38.16	38.18	37.79
0.5	46.81	46.60	46.81	46.76	46.41	46.09	46.73	46.83	46.88	41.27	40.51	41.37	41.32	40.45	41.17	41.26	41.39	41.21
0.6	48.45	48.24	48.43	48.42	48.21	48.38	48.39	48.49	48.42	43.46	42.58	43.57	43.47	42.50	43.36	43.43	43.52	43.01
0.7	49.76	49.53	49.74	49.73	49.50	49.17	49.69	49.78	49.81	45.55	45	45.81	45.78	44.89	45.60	45.68	45.83	45.44
0.8	51.12	50.84	51.09	51.07	50.76	50.28	50.97	51.17	51.09	47.12	46.43	47.26	47.07	46.38	47.03	47.11	47.16	46.89
0.9	52.24	52.06	52.22	52.24	52.06	51.67	52.12	52.26	52.28	48.74	47.95	48.85	48.75	47.91	48.66	48.78	48.91	48.51
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	49.64