Near Real-Time Light Field Video Compression using 5-D Approximate DCT

Sritharan Braveenan, Chamira U. S. Edussooriya *Member, IEEE,* Chamith Wijenayake *Member, IEEE,* and Arjuna Madanayake *Member, IEEE*

***Abstract*—Five-dimensional (5-D) light field videos (LFVs) capture spatial, angular and temporal variations of light rays emanating from scenes. This leads to significantly large amount of data compared to conventional three-dimensional videos (3-D) , which capture only the spatial and temporal variations of light rays. In this paper, we propose an LFV compression technique using low-complexity 5-D approximate discrete-cosine transforms (ADCTs). In order to further reduce the computational complexity, our technique exploits the partial separability of LFV representations and apply two-dimensional (2-D) ADCT for sub-aperture images of LFV frames with intra-view and inter-view configurations. Furthermore, we apply one-dimensional ADCT with respect to the temporal dimension. We evaluate the performance of the proposed LFV compression technique using several 5-D ADCT algorithms, and the exact 5-D DCT. The experimental results obtained with LFVs confirm that the proposed LFV compression technique provides more than 150 times reduction in the data volume with near lossless fidelity with peak-signal-to-noise ratio greater than 40 dB and structural similarity index greater than 0.9. Furthermore, the proposed LFV compression technique achieves compression in 150 seconds for an LFV of size $8 \times 8 \times 160 \times 240 \times 24$, i.e., 6.25 s per LFV frame, with an ADCT requiring only 14 additions for a 8-point ADCT, confirming near real-time processing.**

***Index Terms*—Light field videos, compression, approximate discrete cosine transform, low complexity.**

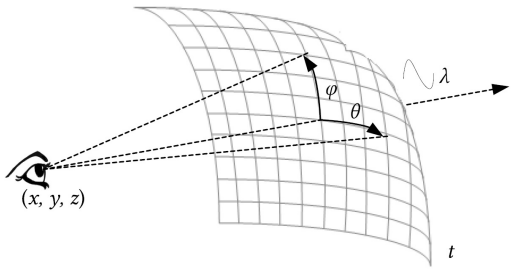

Fig. 1. The 7D plenoptic function

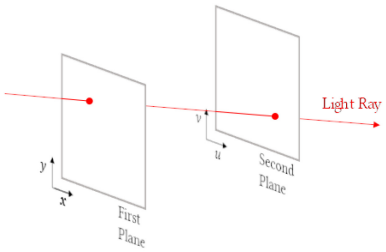

Fig. 2. Two-plane parameterization for LFV frame

I. Introduction

Light rays emanating from a scene is completely defined by the seven-dimensional plenoptic function [1], as shown in Fig 1, which models light rays at every possible location in the three-dimensional (3-D) space (x, y, z), from every possible direction (θ, ϕ), at every wavelengths λ, and at every time t. Five-dimensional (5-D) light field video (LFV) is a simplified form of the seven-dimensional plenoptic function derived with two assumptions: intensity of a light ray does not change along its direction of propagation and the wavelength is represented by red, green and blue (RGB) color channels [2], [3]. With the first assumption, we can eliminate one space dimension, typically z, and with the second assumption, we can eliminate the wavelength. Therefore, an LFV corresponding to one color channel is a 5-D function of x, y, θ, ϕ and t. In general, the spatial and angular dimensions x, y, θ, and ϕ are parameterized using two parallel planes [4], as shown in Fig 2, where (x, y) plane is called the camera plane and (u, v) plane is called the image plane. We call a two-dimensional (2-D) image and 3-D video corresponding to a given spatial sample (x_0, y_0) as a sub-aperture image (SAI) and a sub-aperture video (SAV), respectively.

An LFV captures spatial, angular and temporal variation of light rays in contrast to a conventional 3-D video, which captures only the spatial and temporal variations of light rays. This richness of information of LFVs lead to novel applications such as post-capture refocusing [5]–[7], depth estimation [8] and depth-velocity filtering [9], [10] which are not possible 3-D videos. Furthermore, LFVs are especially useful in augmented reality and virtual reality applications [11]. On the other hand, the data associated with an LFV is significantly larger compared to 3-D videos, e.g., one color channel of an LFv of size $8 \times 8 \times 434 \times 625 \times 16\times$ requires 2.17 GB, with 8-bits per pixel. This limits the potential of real-time processing of LFVs especially with mobile, edge or web applications where storage is limited or data rate is not sufficient for real-time communication. Therefore, in order

S. Braveenan and C. U. S. Edussooriya are with the Department of Electronic and Telecommunication Engineering, University of Moratuwa, Moratuwa 10400, Sri Lanka (e-mail: {160073F, chamira}@uom.lk).

C. Wijenayake is with the School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, QLD 4072, Australia (e-mails: c.wijenayake@uq.edu.au).

A. Madanayake is with the Department of Electrical and Computer Engineering, Florida International University, Miami, FL, USA (e-mail: amadanay@fiu.edu).

to fully exploit the capabilities enabled by LFVs, efficient LFV compression techniques are required to be developed. Even though, a number of compression techniques have been developed for four-dimensional light fields [12], [13], their straight- forward extension to 5-D LFVs may not results the most efficient compression techniques for LFVs.

In this paper, we propose a low-complexity LFV lossy compression technique using 5-D approximate discrete-cosine transforms (ADCTs). To the best of author's knowledge, our techniques is the *first compression technique proposed for LFVs*. We employed low-complexity ADCTs developed for type-2 discrete cosine transform (DCT), which has excellent energy-compaction property and is widely used in data compression applications [14]. These ADCTs have significantly lower computational complexity compared to the exact DCT [15], [16]. In order to further reduce the computational complexity, our technique exploits the partial separability of LFV representations. In particular, we consider blocks of size $8 \times 8 \times 8 \times 8 \times 8$ and apply 2-D 8×8-point ADCT for SAIs of LFV frames with intra-view and inter-view configurations separately. Denoting a discrete-domain LFV as $l(n_x, n_y, n_u, n_v, n_t)$, where n_x, n_y, n_u, n_v, and n_t are the discrete domains corresponding to continuous domains x, y, u, v and t, respectively (with uniform sampling), we apply 2-D ADCT with respect to (n_u, n_v) in the intra-view configuration (i.e., within SAIs) whereas we apply 2-D ADCT with respect to (n_x, n_y) in the inter-view configuration (i.e., across SAIs). Finally, we apply one-dimensional 8-point ADCT with respect to the temporal dimension. We evaluate the performance of the proposed LFV compression technique using several ADCT algorithms, and the exact 5-D DCT. The experimental results obtained with LFVs confirm that the proposed LFV compression technique provides more than 150 times reduction in the data volume with near lossless fidelity with peak-signal-to-noise ratio greater than 40 dB and structural similarity index greater than 0.9. Furthermore, the proposed LFV compression technique, with an unoptimzed MATLAB implementation, achieves compression in 150 seconds for an LFV of size $8 \times 8 \times 160 \times 240 \times 24$ with an ADCT requiring only 14 additions for a 8-point ADCT [16]. That is 6.25 s per LFV frame, which confirms near real-time processing.

The rest of the paper is organized as follows. In Sec. II, we discuss the related work. We present a review of DCT and ADCT in Sec. III. In Sec. IV, we present the proposed LFV compression technique in detail. In Sec. V, we present experimental results, and finally, in Sec. VI, we present conclusion and future work.

II. Related Work

A. 4-D Light Field Compression

2-D Discrete Cosine Transform (2-D DCT) is widely used transform for image and video compression notably in JPEG [17], MPEG-1 [18], H.264 [19], which has good energy compaction for images. 4-D Light Field compression is inspired from 2-D image compression, instead of 2-D blocks LF is partitioned into small size 4-D blocks and applies 4-D DCT directly on it. This approach recently used in the JPEG Pleno Verification Model for lenslet Light Field compression [20]. But to handle 4-D block, high computational complexity is needed, which is not suitable for real-time applications. To reduce complexity, using partial separability, apply cascade of two 2-D DCT on 4-D blocks [13]. Another approach is arranging SAIs of a LF along a third dimension, create a 3-D body and apply 3-D DCT on it [21]. This gives better compression than image-based compression but not suitable for 5-D LFV, where actual temporal dimension involves, using spatial dimension as temporal dimension limits the performance. Blocking artifacts is major drawback of DCT and 2D-DWT which is applied in JPEG2000 [22] works efficiently in reducing the blocking artifacts, same as 4D-DWT uses in LF compression. Karhunen Loève Transform (KLT) is another transform for Light Field compression where Vector Quantization (VQ) scheme uses to cluster different Micro Images (MI) into a representative set of vectors which are used to coded with KLT. This gives better PSNR than DCT based for the lower bit rates [23]. Graph Fourier Transform (GFT) is recently proposed transform for Light Field compression [24] where graph based representation uses to model color, disparity or other geometry information from Light Field. GFT achieves reduction of up to 21.92% in number of transform coefficients when compared to DCT-based compression, while providing better or equal mean squared reconstruction error. However complexity wise DCT is better than other transforms mentioned.

B. Approximate Discrete Cosine Transform

Floating point operation reduction is an important aspect for real time compression [16] than accuracy, which increases circuit complexity and power consumption. To omit this in compression, approximate transforms are introduced in literature, notably 8-point approximate DCT [15], [16], [25]–[27] and 16-point approximate DCT [28]. Generally approximate DCT matrices can be written as matrix multiplication of diagonal matrix which contain only irrational numbers and low complexity matrix which contains only powers of two [16]. Bouguezel [25] first introduces low complexity 8 point approximate DCT in 2008 which matrix does matrix multiplication with 8×8 block in 18 additions and 2 shifts but for exact 8 point DCT, 64 multiplications and 56 additions are needed. After this, several approximate transform algorithms are developed and arithmetic complexity of those are summarized in TableI. Low complexity matrix of BAS2011 [26] does not have $1/2$ to omit right shift but CB2011 [27] only has 0 and 1 to omit both shifts. Both Modified CB2011 [15] and PMC2014 [16] ACDT algorithm arithmetic complexities are same but PMC2014 gives better SAI quality.

III. Review of DCT and ADCT

A. 1-D DCT

Let $\mathbf{x}$ be $N \times 1$ vector, whose entries are given by $x[n]$ for $n = 1, 2, ...N$ and $\mathbf{y}$ is 1-D DCT transformation of $\mathbf{x}$, whose entries are given by [14]:

$$y[k] = a_N[k] \cdot \sum_{n=0}^{N-1} x[n] \cdot \cos\left(\frac{\pi k(2n+1)}{2N}\right), \qquad (1)$$

TABLE I
ARITHMETIC COMPLEXITY

DCT method	Mult	Add	Shifts	Total
Exact DCT	64	56	0	120
BAS2008 [25]	0	18	2	20
BAS2011 [26] with a = 0	0	16	0	16
BAS2011 [26] with a = 1	0	18	0	18
BAS2011 [26] with a = 1	0	18	2	20
CB2011 [27]	0	22	0	22
Modified CB2011 [15]	0	14	0	14
PMC2014 [16]	0	14	0	14

where

$$a_N[k] = \frac{1}{\sqrt{N}} \begin{cases} 1, & k_i = 0 \\ \sqrt{2}, & k_i = 1, 2, .., N-1 \end{cases}$$

To solve equations efficiently, equation (3) can be written as:

$$Y = C_N \cdot X, \tag{2}$$

C_N entries are given by:

$$C_N[k, n] = a_N[k] \cdot \cos\left(\frac{\pi k(2n+1)}{2N}\right)$$

To reduce arithmetic complexity, instead of ideal DCT matrix C_N, approximate DCT matrix $\hat{C}_N$ is used, which can be written as [15], [16], [25]–[27]:

$$\hat{C}_N = D_N \cdot T_N, \tag{3}$$

where D_N is diagonal matrix which consists only irrational numbers and T_N is low complexity matrix which consists powers of 2.

B. 2-D DCT

Let X be $N \times N$ matrix, whose entries are given by $x[n_1, n_2]$ for $n_1, n_2 = 1, 2, ...N$ and Y is 2-D DCT transformation of X, whose entries are given by [29]:

$$y[k_1, k_2] = a_N[k_1] \cdot a_N[k_2] \cdot \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} x[n_1, n_2] \cdot \cos\left(\frac{\pi k_1(2n_1+1)}{2N}\right) \cdot \cos\left(\frac{\pi k_2(2n_2+1)}{2N}\right), \tag{4}$$

where

$$a_N[k_i] = \frac{1}{\sqrt{N}} \begin{cases} 1, & k_i = 0 \\ \sqrt{2}, & k_i = 1, 2, .., N-1 \end{cases}$$

To solve equations efficiently, equation (4) can be written as [16]:

$$Y = C_N \cdot X \cdot C_N^T, \tag{5}$$

where C_N entries are given by:

$$C_N[k, n] = a_N[k] \cdot \cos\left(\frac{\pi k(2n+1)}{2N}\right)$$

In approximate DCT matrix $\hat{C}_N$ used instead of ideal DCT matrix C_N same as above.

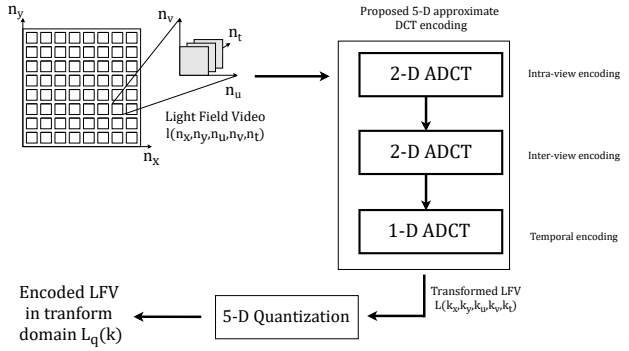

Fig. 3. Overview of the proposed 5-D ADCT based LFV compression

IV. PROPOSED 5-D ADCT BASED LFV COMPRESSION

Overview of Proposed 5-D ADCT Based LFV Compression is explained in Fig. 3, where input to our system is Light Field Video $l(n_x, n_y, n_u, n_v, n_t)$, which size is $(N_x, N_y, N_u, N_v, N_t)$, indicates there are N_t number of frames in Light Field Video and each LFV frame has $N_x \times N_y$ Sub Aperture Images (SAI) where size of each Sub Aperture Image (SAI) is $N_u \times N_v$ pixels. Compression has been done in each channel separately and YUV format is preferable format.

Our first step of the system is intra-view encoding, where intra-views of the LF are transformed into DCT coefficients by partitioning all SAIs of each LFV frame into blocks of 8×8 and applying 2-D ADCT. This intra-view ADCT operation is repeated for all 8×8 blocks in all SAIs of each LFV frame and $N_x \times N_y \times N_u/8 \times N_v/8 \times N_t$ no of operations needed to complete intra-view ADCT transformation on entire Light Field Video. This leads to create mixed domain 5-D signal $L_1(n_x, n_y, k_u, k_v, n_t)$.

Inter-view encoding is second step of system, where we create $N_x \times N_y$ view points blocks from $L_1(n_x, n_y, k_u, k_v, n_t)$ by changing pixel point across SAIs of each LFV frame, partition that into 8×8 blocks and apply 2-D ADCT on it. This inter-view ADCT operation is repeated for all 8×8 blocks across all SAIs of each LFV frame and $N_x/8 \times N_y/8 \times N_u \times N_v \times N_t$ no of operations needed to complete inter-view ADCT transformation on entire Light Field Video. This leads to create mixed domain 5-D signal $L_2(k_x, k_y, k_u, k_v, n_t)$.

Next step of the system is named as temporal encoding, where we take points along time axis from $L_2(k_x, k_y, k_u, k_v, n_t)$ for each LFV view point and pixel point, partition that into dimension of 8×1 vectors and apply 1-D ADCT on it. This temporal ADCT operation is repeated for all 8×1 vectors along time axis for particular view point and pixel point and $N_x \times N_y \times N_u \times N_v \times N_t/8$ no of operations are needed to complete temporal ADCT transformation on entire Light Field Video. This creates fully transformed 5-D signal $L(k_x, k_y, k_u, k_v, k_t)$.

Final step in our proposed 5-D LFV compression system is quantization, where Light Field Video is partitioned into

TABLE II
DATA SET SPECIFICATION

Dataset	Channel	Size	Min	Max
Car	U	$8 \times 8 \times 176 \times 256 \times 24$	25	181
	V	$8 \times 8 \times 176 \times 256 \times 24$	80	193
	Y	$8 \times 8 \times 352 \times 512 \times 24$	3	255
David	U	$8 \times 8 \times 160 \times 240 \times 24$	49	122
	V	$8 \times 8 \times 160 \times 240 \times 24$	107	151
	Y	$8 \times 8 \times 320 \times 480 \times 24$	43	229
Toy	U	$8 \times 8 \times 160 \times 240 \times 24$	3	160
	V	$8 \times 8 \times 160 \times 240 \times 24$	50	224
	Y	$8 \times 8 \times 320 \times 480 \times 24$	12	255

$8 \times 8 \times 8 \times 8 \times 8$ blocks and apply $8 \times 8 \times 8 \times 8 \times 8$ constant value matrix on it, which is expressed as [17]:

$$L_q(k) = round\left(\frac{L(k)}{Q(k)}\right) \cdot Q(k), \tag{6}$$

where $Q(k)$ is 5-D constant value matrix and both division and multiplication are done in element wise. After quantization, coefficients higher than threshold value (value in $Q(k)$) are only retained in $L_q(k)$ and this leads to lossy Light Field Video compression.

V. EXPERIMENTAL RESULTS

To measure performance of Light Field Video compression, three Light Field Videos were used : Car, David, Toy. All three Light Field Videos were made from 15×15 multiple camera array. For our processing 8×8 views were taken and all SAIs are in YUV420 format. Table. II summarizes size, minimum and maximum value of each channel data set.

Compression quality is measured by using Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index for Measuring image quality (SSIM) of each SAI of decompressed Light Field Video with original Light Field Video [30]. The PSNR between the original SAI A and the reconstructed SAI A' is computed as follows [30]:

$$PSNR = 10\log_{10}\left(\frac{2^n - 1}{MSE}\right), \tag{7}$$

Where n is no of bits in SAI and MSE between the two M×N SAIs A and A' is given by:

$$MSE = \left(\frac{1}{MN}\right)\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(A(i,j) - A'(i,j))^2$$

The SSIM between the original SAI A and the reconstructed SAI A' is computed as follows [31]:

$$SSIM = \frac{(2\mu_A\mu_{A'} + C_1)(2\sigma_{AA'} + C_2)}{(\mu_A^2 + \mu_{A'}^2 + C_1)(\sigma_A^2 + \sigma_{A'}^2 + C_2)}, \tag{8}$$

Where $\mu_A, \mu_{A'}, \sigma_A, \sigma_{A'}, , \sigma_{AA'}$ are local means, standard deviations, and cross covariance for SAIs A, A'. The PSNR and SSIM for Light Field Video are computed by averaging the PSNRs and SSIMs of SAIs individually. The final $SSIM_{YCbCr}$ of entire Light Field Video is computed using only the luminance component (Y) [30] and final $PSNR_{YCbCr}$ of entire Light Field Video is computed by using $PSNR_Y, PSNR_{Cb}$ and $PSNR_{Cr}$ as follows [30]:

$$PSNR_{YCbCr} = \frac{6PSNR_Y + PSNR_{Cb} + PSNR_{Cr}}{8} \tag{9}$$

TABLE III
QUANTIZATION

Quantization value	Compression rate [bpp]	PSNR [dB]	SSIM	Energy retained
1	1.33	62.32	1	1
3	0.47	55.26	0.99	1
5	0.28	53.53	0.99	1
7	0.2	52.41	0.99	1
10	0.14	51.26	0.99	0.99
15	0.09	50.06	0.98	0.99
25	0.06	48.76	0.98	0.98
35	0.04	48.06	0.97	0.97
50	0.04	47.44	0.97	0.96
80	0.03	46.67	0.97	0.95
100	0.03	46.31	0.97	0.95
120	0.03	45.97	0.96	0.95
150	0.03	45.65	0.96	0.94
180	0.03	45.25	0.96	0.94
200	0.03	44.99	0.96	0.94

The proposed 5-D ADCT LFV compression attempts to completely exploit the redundancy in both intra-view and inter-view of any type of LFV. In Fig.4(a)-(b) we demonstrate this by comparing the proposed 5-D ADCT based compression with intra-view and temporal only 3-D ADCT-based compression and inter-view and temporal only 3-D ADCT-based compression. As expected, the 5-D ADCT based compression shows best performance in both PSNR and SSIM as it completely utilizes redundancy in LFVs. In Fig. 5(a)-(c), PSNR is plotted against compression rate in bits per pixel for 6 different ADCT matrices and exact DCT. When decreasing compression rate, PSNR is also decreasing and at one point it is converging. All 7 PSNR vs compression rate graphs are nearly same for all 3 data sets. For data set Car and Toy, PSNR converges near to 41dB and compression rate starts at near to 0.6 out of 1. For data set David, PSNR converges near to 45dB and compression rate starts at near to 0.3 out of 1. As mentioned in Table II, data set David has small range values than other data sets. This can be reason for that particular data set getting better PSNR values for small compression rate values.

In Fig. 6(a)-(c), SSIM is plotted against compression rate in bits per pixel for 6 different ADCT matrices and exact DCT. All 7 SSIM vs compression rate graphs are not exactly same for all 3 data sets. From these, assumption can be taken as final outputs will be differed when using different ADCT matrices. Here none of the values are less than 0.88 out of 1. From these, we can confirm that more than 88% percentage of information can be retained while reconstructing compressed LFV. Table III explains when changing value in constant quantization matrix how compression rate, PSNR, SSIM and energy retained are changing. For this experiment Y channel of data set David is used. When you are increasing quantization

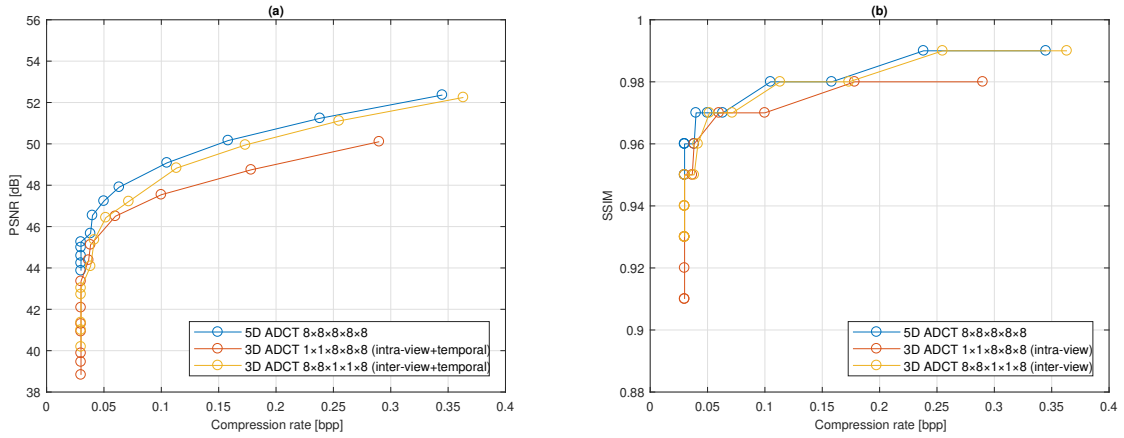

Fig. 4. (a) PSNR vs Compression rate (b) SSIM vs Compression rate for 5-D ADCT, intra-view + temporal only and inter-view + temporal only

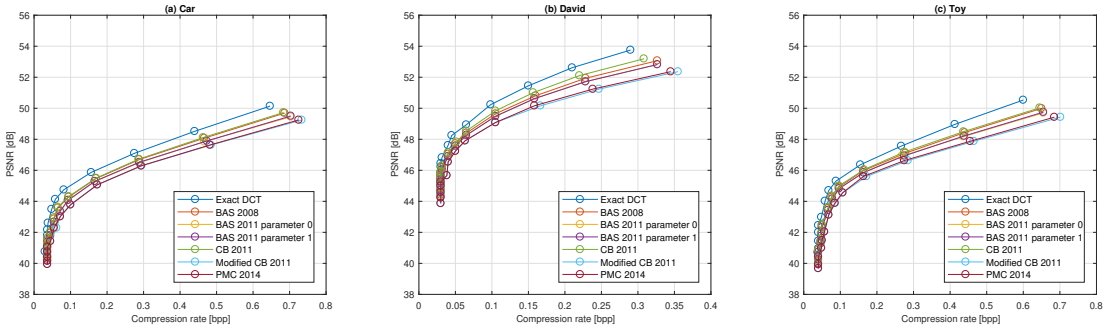

Fig. 5. PSNR vs Compression rate on different ADCT algorithms Exact DCT, BAS-2008 [25], BAS-2011 for parameter 0 [26], BAS-2011 for parameter 1 [26], CB-2011 [27], Modified CB-2011 [15] and PMC 2014 [16] in 5-D domain for data set (a) Car (b) David (c) Toy.

value, at one point retained energy rate starts to decrease but compression rate, PSNR and SSIM converge.

VI. Conclusion and Future Work

Light Field Videos widely used in different applications but due to large amount of data, limits the potential of real-time processing of LFVs. A reduced complexity compression approach is proposed for five dimensional LFVs using multiplier-less 5-D ADCT. The proposed system employs a multiplier-less 8-point ADCT block which has recently appeared in literature and requires only 14 additions. To reduce complexity, by using partial separability, 5-D compression is done in following order: 2-D intra view compression, 2-D inter view compression and temporal compression. Then do quantization on it by dividing Light Field Video into $8 \times 8 \times 8 \times 8 \times 8$ size blocks and obtain compressed Light Field Video. Here Light Field Video compression done for all three channels separately. Effectiveness of the proposed 5-D ADCT based compression is verified with software simulation comparisons against 2-D inter-view + 1-D temporal only and 2-D intra-view + 1-D temporal only ADCT compression achieved using PSNR and SSIM metrics. To check compression quality, SSIM and PSNR is calculated between original Light Field Video and reconstructed Light Field Video for different approximate 8 point 2-D DCT algorithms which are already published in literature. Based on our results we could get minimum 43 dB average PSNR and minimum 0.89 average SSIM. These indicate our compression algorithm is efficient and we do not loose that much information while compressing. Future work includes 5-D Light Field Video compression implementation in FPGA device and 5-D Light Field Video reconstruction algorithm development using prediction modelling.

References

[1] E. H. Adelson and J. R. Bergen, *The plenoptic function and the elements of early vision*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of ..., 1991, vol. 2.

[2] C. Zhang and T. Chen, "A survey on image-based rendering—representation, sampling and compression," *Signal Process.: Image Commun.*, vol. 19, no. 1, pp. 1–28, Jan. 2004.

[3] H.-Y. Shum, S. B. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1020–1037, Nov. 2003.

[4] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. Annu. Conf. Comput. Graph.*, 1996, pp. 31–42.

[5] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," Stanford Univ., Computer Science Technical Report, 2005.

[6] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Linear volumetric focus for light field cameras," *ACM Trans. Graph.*, vol. 34, no. 2, pp. 15:1–15:20, Feb. 2015.

[7] S. S. Jayaweera, C. Edussooriya, C. Wijenayake, P. Agathoklis, and L. Bruton, "Multi-volumetric refocusing of light fields," *TechRxiv*, 2020.

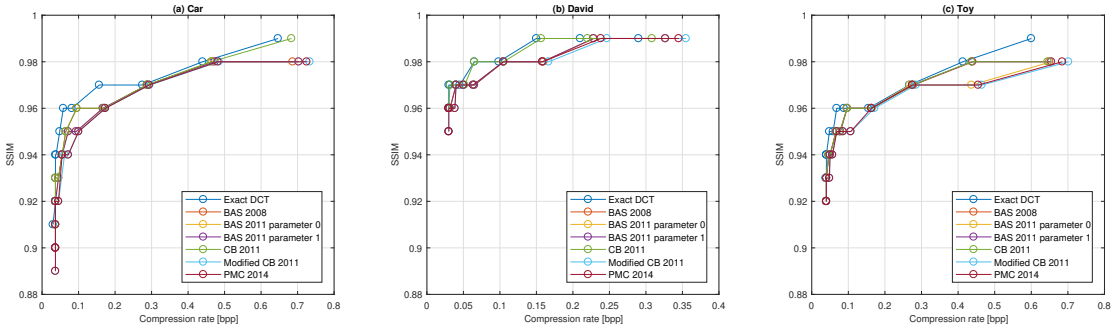

Fig. 6. SSIM vs Compression rate on different ADCT algorithms Exact DCT, BAS-2008 [25], BAS-2011 for parameter 0 [26], BAS-2011 for parameter 1 [26], CB-2011 [27], Modified CB-2011 [15] and PMC 2014 [16] in 5-D domain for data set (a) Car (b) David (c) Toy.

[8] T. Kinoshita and S. Ono, "Depth estimation from 4D light field videos," in *International Workshop on Advanced Imaging Technology*, vol. 11766, 2021, pp. 1–6.

[9] C. U. S. Edussooriya, D. G. Dansereau, L. T. Bruton, and P. Agathoklis, "Five-dimensional depth-velocity filtering for enhancing moving objects in light field videos," *IEEE Trans. Signal Process.*, vol. 63, no. 8, pp. 2151–2163, Apr. 2015.

[10] C. U. S. Edussooriya, L. T. Bruton, and P. Agathoklis, "Velocity filtering for attenuating moving artifacts in videos using an ultra-low complexity 3-D linear-phase IIR filter," *Multidim. Syst. Signal Process.*, vol. 28, no. 2, pp. 597–616, Apr. 2017.

[11] J. Wang, X. Xiao, H. Hua, and B. Javidi, "Augmented reality 3d displays with micro integral imaging," *Journal of Display Technology*, vol. 11, pp. 889–893, 2014.

[12] C. Conti, L. D. Soares, and P. Nunes, "Dense light field coding: A survey," *IEEE Access*, vol. 8, pp. 49 244–49 284, 2020.

[13] N. Liyanage, C. Wijenayake, C. U. S. Edussooriya, A. Madanayake, R. Cintra, and E. Ambikairajah, "Low-complexity real-time light field compression using 4-d approximate dct." IEEE, 2020, pp. 1–5.

[14] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Prentice Hall Press, 2009.

[15] F. M. Bayer and R. J. Cintra, "Dct-like transform for image compression requires 14 additions only," *Electronics Letters*, vol. 48, pp. 919–921, 2012.

[16] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, S. Kulasekera, and A. Edirisuriya, "Improved 8-point approximate dct for image and video compression requiring only 14 additions," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, pp. 1727–1740, 2014.

[17] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still image data compression standard*. Springer Science Business Media, 1992.

[18] N. Roma and L. Sousa, "Efficient hybrid dct-domain algorithm for video spatial downscaling," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–16, 2007.

[19] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, pp. 560–576, 2003.

[20] G. de Oliveira Alves, M. B. de Carvalho, C. L. Pagliari, P. G. Freitas, I. Seidel, M. P. Pereira, C. F. S. Vieira, V. Testoni, F. Pereira, and E. A. da Silva, "The jpeg pleno light field coding standard 4d-transform mode: How to design an efficient 4d-native codec," *IEEE Access*, vol. 8, pp. 170 807–170 829, 2020.

[21] A. Mehanna, A. Aggoun, O. Abdulfatah, M. R. Swash, and E. Tsekleves, "Adaptive 3d-dct based compression algorithms for integral images." IEEE, 2013, pp. 1–5.

[22] M. Rabbani, "Jpeg2000: Image compression fundamentals, standards and practice," *Journal of Electronic Imaging*, vol. 11, p. 286, 2002.

[23] J.-S. Jang, S. Yeom, and B. Javidi, "Compression of ray information in three-dimensional integral imaging," *Optical Engineering*, vol. 44, p. 127001, 2005.

[24] V. R. M. Elias and W. A. Martins, "On the use of graph fourier transform for light-field compression," *Journal of Communication and Information Systems*, vol. 33, 2018.

[25] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "Low-complexity 8× 8 transform for image compression," *Electronics Letters*, vol. 44, pp. 1249–1250, 2008.

[26] ——, "A low-complexity parametric transform for image compression." IEEE, 2011, pp. 2145–2148.

[27] R. J. Cintra and F. M. Bayer, "A dct approximation for image compression," *IEEE Signal Processing Letters*, vol. 18, pp. 579–582, 2011.

[28] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "A novel transform for image compression." IEEE, 2010, pp. 509–512.

[29] N. I. Cho and S. U. Lee, "Fast algorithm and implementation of 2-d discrete cosine transform," *IEEE transactions on circuits and systems*, vol. 38, pp. 297–305, 1991.

[30] J. Jpeg, F. Pereira, C. Pagliari, E. D. Silva, I. Tabus, H. Amirpour, M. Bernardo, and A. Pinheiro, "Coding of still pictures title: Jpeg pleno light field coding common test conditions v3.3 source: Wg1 contributors: International organisation for standardisation international electrotechnical commission jpeg pleno-light field coding common test conditions 2 contents," 2019.

[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, pp. 600–612, 2004.