Low-Complexity Real-Time Light Field Compression using 4-D Approximate DCT

Namalka Liyanage*, Chamith Wijenayake†, Chamira U. S. Edussooriya‡, Arjuna Madanayake§, Renato Cintra¶, Eliathamby Ambikairajah*

*School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia
†School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, Australia
‡Department of Electronic and Telecommunication Engineering, University of Moratuwa, Moratuwa, Sri Lanka
§ Department of Electrical and Computer Engineering, Florida International University, Miami, USA
§ Department of Electrical Engineering, Federal University of Pernambuco, Brazil

***Abstract*—A low-complexity codec and a hardware architecture are proposed for achieving real-time compression of four-dimensional (4-D) light field (LF) signals captured from camera/lenslet arrays. The proposed system employs the 4-D extension of the two-dimensional (2-D) 8×8 approximate discrete cosine transform (ADCT) that has recently appeared in the literature. Motivated by the partial separability of the multidimensional spectrum of LFs, the proposed 4-D ADCT is obtained by cascading 2-D inter-view and 2-D intra-view transform stages. Software simulations are provided to confirm the performance of the 4-D ADCT based compression and comparisons are made with respect to 2-D inter-view only and 2-D intra-view only compression. Proposed digital hardware designs are validated using stepped hardware co-simulation on a Xilinx Virtex-7 VC-707 FPGA platform verifying 597 MHz maximum frequency of operation, implying 18×10^3 LFs/sec throughput for performing 4-D ADCT on $(8\times\ 8\times432\times624\times3)$ size LFs, with 38 dB average PSNR and 0.95 average SSIM for sub aperture images when 10% of the ADCT coefficients per each $(8\times8\times8\times8)$ hypercube are retained.**

I. Introduction

A four dimensional (4-D) light field (LF), which is a lower-dimensional abstraction of the seven-dimensional (7-D) plenoptic function [1], geometrically represents the light rays emanating from a scene with respect to a pair of reference planes (i.e., camera and image planes). Such LFs, typically captured from either a plenoptic camera (such as Lytro camera [2]), a camera array or a gantry [1] can be considered as a two-dimensional (2-D) array of 2-D sub aperture images (SAIs). Due to additional post-capture processing capabilities such as depth filtering [3]–[10], volumetric focusing [11], [12], imaging in low light [13] and occlusion suppression [5]–[7], LF imaging can provide added benefit in surveillance [14], [15], multimedia and industrial [16], [17] applications. However, the large volume of data captured in LFs makes it quite challenging to exploit the full potential of LF imaging, especially in real-time, mobile and web-based applications, where limited storage and transmission bandwidth requirements are to be met. Thus, fast and low-complexity LF compression techniques are of common interest [15], [18].

In this work, we propose a 4-D LF compression system with low area- and time- very large scale integration (VLSI) complexities, suitable for real-time compression of 4-D LFs using 4-D *approximate discrete cosine transform* (ADCT). The proposed system employs a multiplier-less, 8-point 1-D ADCT, which has recently appeared in literature and require only 14 additions [19], as a building block to obtain higher dimensional (approximate) transform. Motivated by the 2-D partial separability of the 4-D frequency domain corresponding to LFs, 4-D ADCT is obtained as a cascade of two 2-D ADCTs, which are referred to as inter-view and intra-view transform. Here, intra-view and inter-view refer to operations within a SAI and across SAIs, respectively. Thus obtained 4-D ADCT of a LF is subsequently treated as a collection of 4-D hypercubes of size $(8\times8\times8\times8)$ and are subjected to 4-D quantization to obtain an approximated and compressed LF in the 4-D DCT domain; see Fig. 1. We also provide comparisons with commonly used 2-D inter-view only and 2-D intra-view only LF compression approaches, which are all based on ADCT computation. Finally, we propose a hardware architecture that exploits massive parallelism available on field programmable gate arrays (FPGAs) to perform fast computation of the proposed 4-D ADCT of a LF. The proposed hardware designs are implemented on a Xilinx Virtex 7 VC-707 FPGA platform and verified using stepped hardware co-simulation with both lenslet array based and high density camera array (HDCA) based real LF data sets.

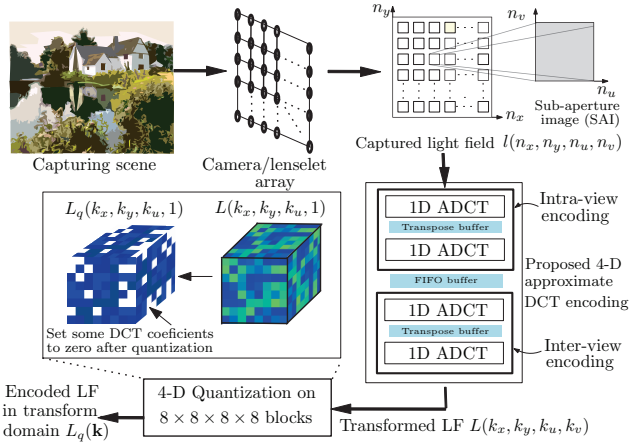

Fig. 1: Overview of the proposed 4-D ADCT based LF compression. Full processing pipeline which typically involves entropy coding of the (approximate) DCT coefficients after quantization is not shown.

II. Related Work in Light Field Compression

Transform-based lossy compression for 4-D LFs range from 2-D image compression [20], [21] exploiting either inter-

view pixel coherence or intra-view pixel coherence [15], 3-D video compression applied on an image stack (i.e., a pseudo video) obtained by ordering the SAIs of a LFs along the typical time dimension in video processing [22]–[24], and true 4-D techniques based on both inter-view and intra-view compression [25]–[29]. Arranging the SAIs of a LF along a third dimension creates a 3-D body which can employ transform-based encoding techniques such as 3D-DCT [30]–[32] and 3-D discrete wavelet transform (3D-DWT) [33], [34]. These compression schemes exploit existing intra-view spatial redundancy and partially exploit inter-view correlation, consequently leading to improved compression performance compared to image based compression. However, the use of a spatial dimension as the time dimension in LF compression limits their performance in 5-D light field video (LFV) compression, where the *actual* time dimension is involved.

LF encoding methods described in [25]–[29] employ both intra- and inter-view pixel coherence by applying 4-D DCT. In [26], 4-D redundancy in lenslet array-based and HDCA-based LF data sets were exploited using 4-D DCT and 2-D DCT to exploit intra-view and inter-view redundancy in LFs. This work also shows that better compression performance is achievable for HDCA LFs by employing larger block size $(8 \times 8 \times 64 \times 64)$ of the input LF for 4-D DCT. In [28] and [29] similar 4-D DCT-based LF encoding approaches were described following a novel JPEG based 4-D quantization technique. Further, a weighted 4-D DCT LF encoding technique was proposed in [25].

Both approximate arithmetic hardware units [35]–[37] and approximate transform algorithms [38]–[41] have been used in image and video compression to obtain drastic reduction in computational complexity within the compression pipeline. Several null multiplicative approximations of 1-D 8-point DCT have been proposed in [19], [38]–[43], which require only binary additions and shift operation with no multiplications. These 8-point DCT approximations include, Bouguezel-Ahmad-Swamy (BAS) series [38]–[40] which require 16 to 24 additions and the rounded DCT (RDCT) [43], which require 22 additions. In particular, the modified rounded DCT (MRDCT) [41] and the improved DCT (IDCT) [19] report the lowest complexity of with 14 additions only. While these algorithms have been used in image and video compression, to the best of our knowledge, they have not been used in the context of 4-D static LF and 5-D dynamic LFV signals.

III. Proposed 4-D ADCT based LF Compression

The input to the proposed 4-D ADCT-based compression system is a LF $l(\mathbf{n})$, where $\mathbf{n} \equiv (n_x, n_y, n_u, n_v) \in \mathbb{N}^4$ denotes the 4-D discrete spatial domain, (n_x, n_y) corresponds to the view-point index on the camera plane and (n_u, n_v) corresponds to the pixel index of a SAI on the image plane. Here, with the standard two plane parameterization [1], the input LF is assumed to have $N_x \times N_y$ SAIs, each of size $N_u \times N_v$ pixels. Further, the three color channels are processed independently, thus, for brevity, we limit the discussion to one color channel.

In the first step of the system, intra-views of the LF are transformed into DCT coefficients by partitioning all SAIs into blocks of 8×8 and applying 2-D DCT. Let X be the extracted 2-D block of size 8×8, whose entries are given by $x[n_1, n_2]$, for $n_1, n_2 = 1, 2, \ldots, 8$. The entries of the 2-D transformed block Y with ideal DCT are given by [44]:

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

where

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

for $k_1, k_2 = 1, 2, \ldots, N-1$ and $N = 8$. This ideal 2-D DCT can also be expressed in matrix form $Y = \mathbf{C}_N \cdot X \cdot \mathbf{C}_N^{\intercal}$, where the elements of the ideal DCT matrix is $\mathbf{C}_N$ are $C_N[k, n] = a_N[k] \cdot \cos\left(\frac{\pi(2n+1)k}{2N}\right)$ for $k, n = 0, 1, \ldots, N-1$. To achieve null multiplier complexity and reduced adder complexity, we employ the ADCT algorithm in [19], which corresponds to the ADCT matrix $\widehat{\mathbf{C}}_\mathbf{N}$, expressed as the product of low complexity $N \times N$ integer matrix $\mathbf{T}_N$ and $N \times N$ diagonal matrix $\mathbf{D}_N$; $\widehat{\mathbf{C}}_N = \mathbf{D_N T_N}$ [19], [38]–[43]. Although, the diagonal matrix $\mathbf{D}_N$ typically contains irrational numbers leading to a computational overhead, in the context of image compression, this diagonal matrix is absorbed into the the quantization step of the image compression algorithm [20]. This 2-D blocked-ADCT operation is repeated for all the SAIs within the LF. Thus, $N_x \times N_y \times \lceil N_u/8 \rceil \times \lceil N_v/8 \rceil$ many such 2-D ADCT operations are performed in order to complete the intra-view ADCT transform on the entire input LF, leading to the mixed-domain 4-D signal $L_1(n_x, n_y, k_u, k_v)$. Next, using a similar approach, inter-view 2-D ADCT operation is performed by considering 8×8 blocks of view points extracted with respect to a given pixel point across the SAIs. This step also repeats for $\lceil N_x/8 \rceil \times \lceil N_y/8 \rceil \times N_u \times N_v$ times to fully transform the LF to obtain $L(\mathbf{k})$, where $(k_x, k_y, k_u, k_v) \in \mathbb{N}^4$.

Next, 4-D quantization is performed on $L(\mathbf{k})$ to discard high frequency components. As a proof-of-the-concept, we employ an $8 \times 8 \times 8 \times 8$ constant valued matrix $Q(\mathbf{k})$ with a pre-defined threshold to quantize each $8 \times 8 \times 8 \times 8$ hypercube of $L(\mathbf{k})$ as $L_q(\mathbf{k}) = \text{round}\left[\left(\frac{L(\mathbf{k})}{Q(\mathbf{k})}\right)\right] \cdot Q(\mathbf{k})$, where both the division and multiplication of the matrices are element-wise. Alternatively, frequency-dependent 4-D quantization approaches can also be used such as the 4-D extension of JPEG quantization [20], [28], [29]. After such quantization, the 4-D ADCT coefficients higher than the threshold are retained, leading to a lossy compression of the LF. In order to reduce the arithmetic complexity of the 4-D ADCT computation, the four diagonal matrices corresponding to four individual ADCT operations can be absorbed into the 4-D quantization matrix $Q(\mathbf{k})$. In [45], a method is proposed to embed the diagonal matrices into a 3-D quantization volume. We extend this method in [45] for the 4-D hyper-volume case to create the modified quantization matrix $Q^*(\mathbf{k})$ and employ for quantization.

IV. Performance Analysis

Quality of the compressed LF is assessed by using peak signal to noise ratio (PSNR) and structural similarity (SSIM)

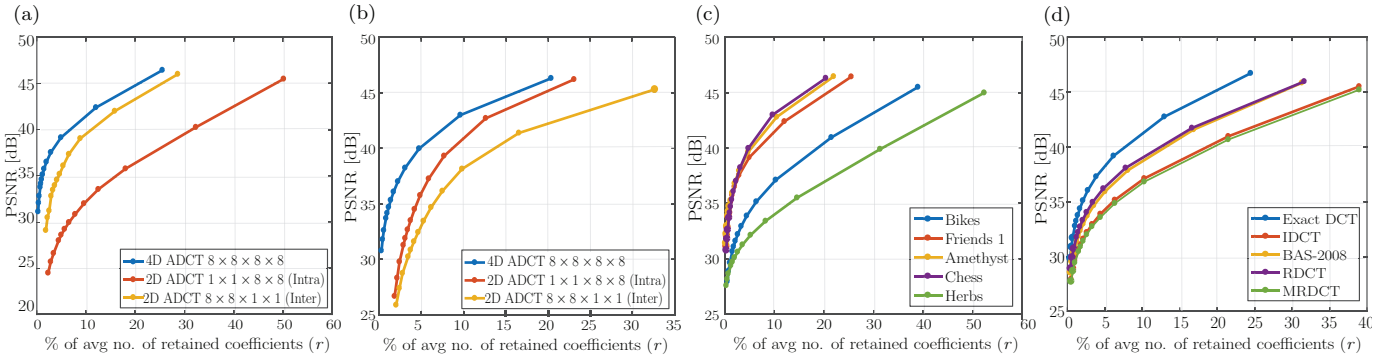

Fig. 2: PSNR vs. the percentage of retained coefficients (r) for 4-D ADCT, 2-D ADCT for intra-view only and 2-D ADCT for inter-view only, on (a) lenslet LF Friends1 and (b) HDCA LF Chess. (c) The proposed 4-D ADCT based compression on different LFs. (c) Comparison with different ADCT algorithms IDCT [19], BAS-2008 [38], RDCT [43], MRDCT [41] and exact DCT in the 4-D domain.

metrics of SAIs of the decompressed LF. For both PSNR and SSIM, we report minimum, maximum and average values across the SAIs. Encoding efficiency is assessed by the percentage of average number of retained coefficients (r) in an $8 \times 8 \times 8 \times 8$ block in the transform domain, bits-per-pixel (bpp), and compression ratio, which is the ratio between the file size of the input and output LFs. The data set used for the performance analysis includes five LFs: 'Bikes' and 'Friends1' from the EPFL LF data set [46], which have been captured by a Lytro camera [2]; 'Amethyst' and 'Chess' HDCA LFs from the Stanford LF data set, which have been acquired using a Lego Gantry [47]; and the 'Herbs' LF, which is a synthetically generated LF from the HCI database [48]. Although these LFs are of different resolution in all four dimensions, to ease our process of encoding $8\times8\times8\times8$ sized hypercubes, we selected the central 8×8 SAIs and truncate the resolution in n_u, n_v dimensions to the nearest multiple of 8.

It has been shown that lenslet-based LFs with low disparity among view points have higher inter-view data correlation and HDCA-based LFs with high disparity have higher contribution on 4-D redundancy from intra-view of the LF [26]. Thus, the proposed 4-D ADCT LF encoding attempts to completely exploit the 4-D redundancy in both intra- and inter-view of any type of LF. In Fig. 2(a)-(b) we demonstrate this by comparing the proposed 4-D ADCT based compression with intra-view only 2-D ADCT-based and inter-view only 2-D ADCT-based compression on lenslet and HDCA LFs, respectively, showing the average PSNR vs. the percentage of retained coefficients (r). As expected, the 4-D ADCT encoding shows best performance as it completely utilizes 4-D redundancy in LFs.

Fig. 2(c) shows the resulting average PSNR vs. r for the five different LFs considered, with the 4-D ADCT encoding. Further, Table. I shows the 4-D ADCT based compression results for these LFs for different values of r. Elapsed encoding time is measured for each LF using the MATLAB implementation of 4-D ADCT on a Windows 10 machine with an Intel Core i7 processor with 8 GB RAM and encoding time increases in proportion to the size of the input LF. For $r = 10\%$, all these LFs have about 0.5 bpp bitrate and $11:1$ compression ratio. Also, we present the minimum, maximum, and average values of PSNR and SSIM metrics to demonstrate that all SAIs are adequately encoded following the proposed method.

TABLE I: 4-D ADCT ENCODING ON LFS FOR $r = 10, 5, .5\%$

r	LF	Elapsed time (s)	Comp. Rate (bpp)	Comp.-ratio	PSNR (dB)			SSIM		
					min	avg	max	min	avg	max
10	Bikes	10.5	.52	11 : 1	36	37	38	.95	.96	.98
	Friends1	10.4	.56	10 : 1	41	42	43	.98	.99	.99
	Amethyst	28.2	.46	10 : 1	42	43	43	.97	.98	.98
	Chess	39.9	.44	12 : 1	43	43	43	.97	.97	.97
	Herbs	10.2	.41	13 : 1	33	33	34	.95	.96	.97
5	Bikes	10.6	.30	21 : 1	32	34	36	.92	.94	.96
	Friends1	10.5	.31	16 : 1	37	39	40	.97	.98	.99
	Amethyst	28.3	.27	19 : 1	39	40	40	.95	.96	.96
	Chess	39.9	.27	21 : 1	39	40	41	.95	.95	.96
	Herbs	10.5	.39	15 : 1	31	32	33	.95	.96	.96
.5	Bikes	9.2	.07	71 : 1	26	27	28	.82	.85	.88
	Friends1	9.3	.07	60 : 1	30	32	34	.93	.94	.95
	Amethyst	27.4	.06	83 : 1	31	32	34	.89	.91	.93
	Chess	38.7	.07	86 : 1	30	31	33	.84	.87	.89
	Herbs	9.1	.07	65 : 1	27	28	29	.88	.90	.91

Additionally, we employ selected 8-point DCT approximation algorithms, namely IDCT [19], BAS-2008 [38], RDCT [43] and MRDCT [41] along with exact DCT albeit in their 4-D extensions to compare the performance. Experimental results shown in Fig. 2(d) indicate that the average PSNR vs. r values for the 'Bikes' LF, and as expected exact DCT shows the best encoding performance. BAS-2008 and RDCT which are computationally expensive with 18 and 22 additions respectively, show better encoding performance compared to MRDCT and IDCT which only require 14 additions to implement 1-D 8-point ADCT.

V. HARDWARE IMPLEMENTATION OF PROPOSED LF COMPRESSION ALGORITHM

Proposed digital hardware architecture to perform the 4-D ADCT is shown in Fig. 3. In the first step of the process 2-D ADCT is applied to intra-views of the LF using an array of M parallelly connected 8×8 2D_ADCT modules (Fig. 3(a)). Each 2D_ADCT module is a cascade of 1D_ADCT modules to perform row-wise transform, transpose buffer and another 1D_ADCT module to perform column-wise transform. This Intra-view encoding takes $(N_xN_y/8M)N_uN_v$ cycles to complete transform and until that partially transformed output LF is stored in a FIFO buffer of size $N_xN_yN_uN_v$. Next step applies the 2-D ADCT inter-view transformation to the partially transformed LF, following a similar cascaded architecture with M parallel connected 8×8 2D_ADCT modules (Fig. 3(c)). Notice that, 1D_ADCT module can implement any

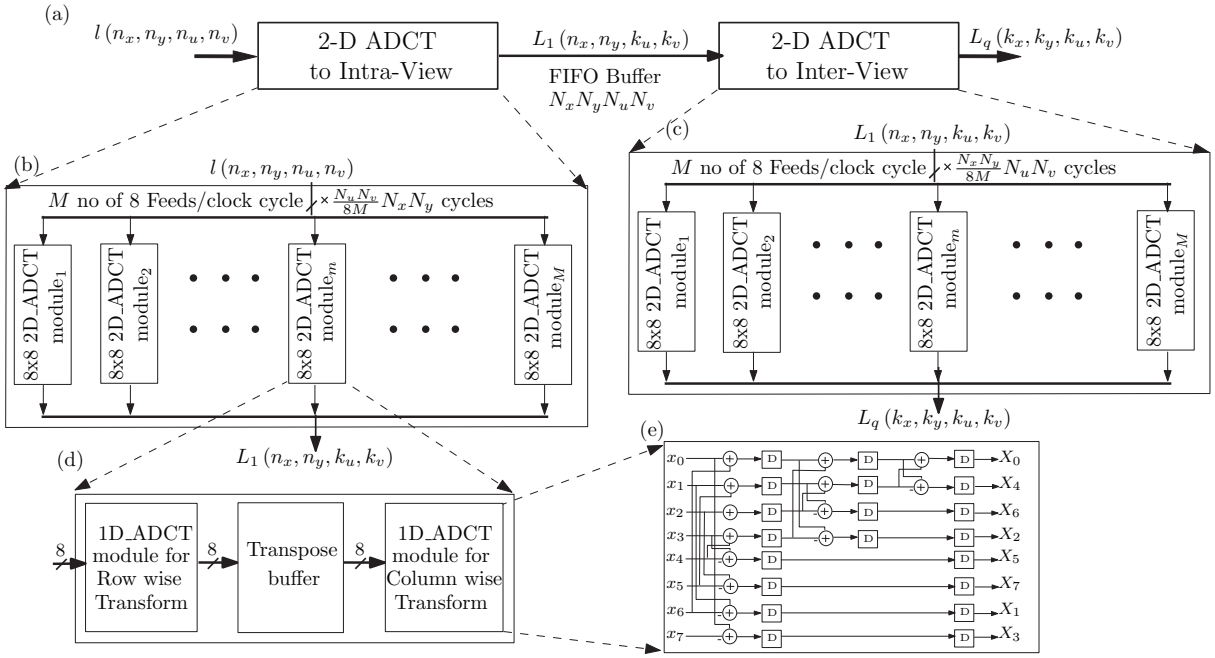

Fig. 3: Hardware architecture for the proposed 4-D ADCT based LF encoding algorithm; (a) two subsystems of the architecture with; (b) M number of 2D_ADCT modules to apply 2-D ADCT to intra-view of the LF; (c) M number of 2D_ADCT modules to apply 2-D ADCT to Inter-view of the LF.(d) 2D-ADCT module created by cascading a 1D-ADCT module for row-wise 1-D ADCT, transposition buffer and 1D-ADCT module for column-wise 1-D ADCT. (e) Digital architecture of 1D-ADCT module for IDCT [19]

TABLE II: FPGA Resources and Performance of Proposed Encoding Scheme for Different Approximate DCT

Parameters	BAS-2008	RDCT	MRDCT	IDCT
M	175	175	200	200
flip-flop pairs (%)	421050(69)	446600(74)	420800(69)	401600(66)
Slice LUTs (%)	284900(94)	295050(97)	280400(92)	280400(92)
LUTRAM(%)	41650(32)	41300(31)	53600(41)	72800(55)
CPD (ns)	1.88	1.67	1.67	1.67
Frequency (MHz)	531	597	597	597
Throughput (LF/s)	14×10^3	16×10^3	18×10^3	18×10^3

TABLE III: Performance of LFs encoded on FPGA, $r = 10\%$

LF	PSNR (dB)			SSIM		
	min	avg.	max	min	avg.	max
Bikes	34	36	36	.93	.94	.97
Friends1	39	40	42	.95	.96	.96

particular 8-point ADCT algorithm available and accordingly M should be selected to effectively utilize the available hardware resources. The architecture operates at speed $F_{clk} = 1/T_{CPD}$ where T_{CPD} denotes the critical path delay (CPD) of the design. As systolic data processing happens for n_u, n_v dimensions in intra-view subsystem and n_x, n_y dimensions in inter-view subsystem, the total time taken to encode a LF with three color channels is $T_{overall} = \frac{3}{8 \times M} N_x N_y N_u N_v T_{CPD}$ (Theoretical throughput).

A prototype FPGA implementation of the proposed architecture was functionally verified on a Xilinx Virtex-7 xc7vx485tffg1761-2 FPGA device by using on-chip hardware co-simulation facility in Xilinx Vivado System generator platform with inputs provided through MATLAB workspace. Table. II reports hardware resource consumption and time taken to perform encoding on 'Bikes' LF of size ($8\times 8\times 432\times 624 \times 3$), for hardware architectures implemented following IDCT, BAS-2008, RDCT and MRDCT DCT approximations for the 1D_ADCT module. These designs are compared in terms of CPD, throughput and resource consumption. Table. III provides PSNR and SSIM values for 'Bikes' and 'Friends1' LFs encoded on FPGA for fixed-point word length 8 and then quantized on MATLAB workspace. These values for $r = 10\%$ show 2 dB and 0.2 degradation in PSNR and SSIM values, respectively, compared to the respective software implementation results given in Table. I.

VI. Conclusion and Future Work

A reduced complexity compression approach is proposed for high-dimensional LFs using multiplier-less 4-D ADCT. The proposed system employs a multiplier-less 8-point ADCT block which has recently appeared in literature and requires only 14 additions, as the building block to obtain a 4-D ADCT of a LF and subsequently employs 4-D hard-thresholding within each $8 \times 8 \times 8 \times 8$ hypercube of such transformed data. Effectiveness of the proposed 4-D ADCT based compression is verified with software simulation comparisons against 2-D inter-view only and 2-D intra-view only ADCT compression achieved using PSNR and SSIM metrics. A digital hardware architecture is proposed which exploits massive parallelism on a Xilinx Virtex-7 FPGA device verifying 597 MHz of maximum frequency of operation correspondingly implying a real-time throughput of 18×10^3 LFs/sec. FPGA designs are verified using stepped hardware co-simulation facility available in Xilinx Vivado System Generator tool flow and reports average PSNR of 38 dB and average SSIM of 0.95, computed across the SAIs of an example LF, when 10% of the ADCT coefficients are retained per each $8\times8\times8\times8$ hypercube of transformed data. Future work includes experimentation with more sophisticated 4-D quantization approaches and extending the proposed design for 5-D LFVs.

REFERENCES

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

[2] *Lytro ILLUM User Manual*. [Online]. Available: http://manuals.lytro.com/illum/

[3] D. Dansereau and L. Bruton, "A 4D frequency-planar IIR filter and its application to light field processing," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4, 2003, pp. IV–476–IV–479.

[4] D. Dansereau and L. T. Bruton, "A 4-D dual-fan filter bank for depth filtering in light fields," *IEEE Trans. Signal Process.*, vol. 55, no. 2, pp. 542–549, 2007.

[5] N. Liyanage, C. Wijenayake, C. Edussooriya, and A. Madanayake, "Low-complexity 4-D IIR filters for Multi-depth filtering and Occlusion suppressing in Light fields," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2018.

[6] N. Liyanage, C. Wijenayake, C. U. S. Edussooriya, and E. Ambikairajah, "Reduced-complexity depth filtering and occlusion suppression using modulated-sparse light fields," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.

[7] N. Liyanage, C. Wijenayake, C. Edussooriya, A. Madanayake, P. Agathoklis, L. T. Bruton, and E. Ambikairajah, "Multi-depth filtering and occlusion suppression in 4-D light fields: Algorithms and architectures," *Signal Processing*, vol. 167, p. 107294, 2020.

[8] 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. on Sig. Proc.*, vol. 63, no. 8, pp. 2151–2163, 2015.

[9] C. U. S. Edussooriya, L. T. Bruton, and P. Agathoklis, "A 5-D IIR depth-velocity filter for enhancing objects moving on linear-trajectories in light field videos," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2015, pp. 2381–2384.

[10] C. Wijenayake, N. Liyanage, C. U. S. Edussooriya, H. Seatang, P. Agathoklis, and L. T. Bruton, "Design and implementation of 5-D IIR depth velocity filters for light field video processing," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, pp. 1–5, 2018.

[11] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Linear Volumetric Focus for Light Field Cameras," *ACM Trans. Graphics*, vol. 34, no. 2, pp. 1–20, 2015.

[12] S. U. Premaratne, C. U. S. Edussooriya, C. Wijenayake, L. T. Bruton, and P. Agathoklis, "A 4-D sparse FIR hyperfan filter for volumetric refocusing of light fields by hard thresholding," in *Proc. Int. Conf. Digital Signal Process.*, 2018, pp. 1–5.

[13] A. Isaksen, L. McMillan, and S. Gortler, "Dynamically reparameterized light fields," in *Proc. Annu. Conf. Comput. Graph.*, 2000, pp. 297–306.

[14] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy, "Using plane + parallax for calibrating dense camera arrays," in *Proc. IEEE Conf. Comput. Vision and Pattern Recogn.*, vol. 1, 2004, pp. I–2–I–9.

[15] R. S. Higa, Y. Iano, R. B. Leite, R. F. Chavez, and R. Arthur, "Employing light field cameras in surveillance: An analysis of light field cameras in a surveillance scenario," *3D Res.*, vol. 5, no. 1, pp. 5:1–5:11, Mar. 2014.

[16] J. Chen, J. Hou, Y. Ni, and L. Chau, "Accurate light field depth estimation with superpixel regularization over partially occluded regions," *IEEE Trans. on Image Proc.*, vol. 27, no. 10, pp. 4889–4900, Oct 2018.

[17] Y. Ni, J. Chen, and L. Chau, "Reflection removal based on single light field capture," in *2017 IEEE Int. Symp. on Circuits Systems*, May 2017, pp. 1–4.

[18] *Overview of JPEG Pleno*. [Online]. Available: https://jpeg.org/jpegpleno/index.html

[19] 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 Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. June, pp. 1727–1740, jun 2014.

[20] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York, NY: Van Nostrand Reinhold, 1992.

[21] D. Taubman and M. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Springer Publishing Company, Incorporated, 2013.

[22] D. Liu, L. Wang, L. Li, Zhiwei Xiong, Feng Wu, and Wenjun Zeng, "Pseudo-sequence-based light field image compression," in *IEEE Int. Conf. Multimedia & Expo Workshops (ICMEW)*, jul 2016, pp. 1–4.

[23] J. Chen, J. Hou, and L. Chau, "Light field compression with disparity-guided sparse coding based on structural key views," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 314–324, Jan 2018.

[24] F. Murgia and D. Giusto, "A comparison of raw light field lossless data compression algorithms," pp. 1–4, nov 2016.

[25] Y. Miyagi, K. Takahashi, M. P. Tehrani, and T. Fujii, "Reconstruction of compressively sampled light fields using a weighted 4D-DCT basis," in *IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 502–506.

[26] G. Alves, M. P. Pereira, M. B. de Carvalho, F. Pereira, C. L. Pagliari, V. Testoni, and E. A. B. da Silva, "A study on the 4-D sparsity of jpeg pleno light fields using the discrete cosine transform," in *IEEE Int. Conf. Image Process. (ICIP)*, Oct 2018, pp. 1148–1152.

[27] M. B. De Carvalho, M. P. Pereira, G. Alves, E. A. Silva, C. L. Pagliari, F. Pereira, and V. Testoni, "A 4-D DCT-Based lenslet light field codec," in *IEEE Int. Conf. Image Process.(ICIP)*, aug 2018, pp. 435–439.

[28] D. Barina, T. Chlubna, M. Solony, D. Dlabaja, and P. Zemcik, "Evaluation of 4D Light Field Compression Methods," may 2019. [Online]. Available: http://arxiv.org/abs/1905.07432

[29] "4D-DCT Based Light Field Image Compression," *Excel@FIT, 2019*.

[30] A. Mieee, "A 3D DCT Compression Algorithm For Omnidirectional Integral Images," in *2006 IEEE Int. Conf. Acoustics Speed and Signal Process.*, vol. 2, pp. II–517–II–520.

[31] M. C. Forman and A. Aggoun, "Quantisation strategies for 3D-DCT-based compression of full parallax 3D images," in *Int. Conf. Image Process. and Its Applications*, vol. 1, July 1997, pp. 32–35 vol.1.

[32] A. Mehanna, A. Aggoun, O. Abdulfatah, M. R. Swash, and E. Tsekleves, "Adaptive 3D-DCT based compression algorithms for integral images," in *IEEE Int. Symp. on Broadband Multimedia Systems and Broadcasting (BMSB)*, June 2013, pp. 1–5.

[33] A. Aggoun and M. Mazri, "Wavelet-based compression algorithm for still omnidirectional 3D integral images," *Signal Image and Video Processing*, vol. 2, pp. 141–153, 06 2008.

[34] A. Aggoun, "Compression of 3D integral images using 3D wavelet transform," *Journal of Display Technology*, vol. 7, no. 11, pp. 586–592, Nov 2011.

[35] K. Chen, F. Lombardi, and J. Han, "Matrix multiplication by an inexact systolic array," in *Proc. IEEE/ACM Int. Symp. Nanoscale Architectures (NANOARCH)*, July 2015, pp. 151–156.

[36] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 984–994, April 2015.

[37] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2014, pp. 1–4.

[38] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "Low-complexity 8x8 transform for image compression," *Electronics Letters*, vol. 44, no. 21, pp. 1249–1250, October 2008.

[39] ——, "A low-complexity parametric transform for image compression," in *IEEE Int. Symp. Circuits Sys. (ISCAS)*, May 2011, pp. 2145–2148.

[40] ——, "Binary discrete cosine and hartley transforms," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 4, pp. 989–1002, April 2013.

[41] F. M. Bayer and R. J. Cintra, "DCT-like transform for image compression requires 14 additions only," *Electronics Letters*, vol. 48, no. 15, pp. 919–921, July 2012.

[42] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, and N. Rajapaksha, "Multiplier-free DCT approximations for RF multi-beam digital aperture-array space imaging and directional sensing," *Measurement Science and Technology*, vol. 23, no. 11, p. 114003, oct 2012.

[43] R. J. Cintra and F. M. Bayer, "A dct approximation for image compression," *IEEE Signal Process. Lett.*, vol. 18, no. 10, pp. 579–582, Oct 2011.

[44] Nam Ik Cho and San Uk Lee, "Fast algorithm and implementation of 2-D discrete cosine transform," *IEEE Trans. Circuits Syst.*, vol. 38, no. 3, pp. 297–305, March 1991.

[45] V. D. A. Coutinho, R. J. Cintra, and F. M. Bayer, "Low-complexity multidimensional DCT approximations for high-order tensor data decorrelation," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2296–2310, 2017.

[46] M. Řeřábek and T. Ebrahimi, "New Light Field Image Dataset," vol. 1, pp. 470–473, 2016. [Online]. Available: http://mmspg.epfl.ch/EPFL-light-field-image-dataset

[47] A. Andrew, "The (new) stanford light field archive." [Online]. Available: http://lightfield.stanford.edu/lfs.html

[48] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4D light fields," in *Asian Conference on Computer Vision*. Springer, 2016.