Near Real-Time Light field video Compression using 5-D Approximate DCT

Sritharan Braveenan
160073F

Supervisor: Dr. Chamira U. S. Edussooriya

Co-Supervisor: Dr. Chamith Wijenayake

EN4922 Research Project

June 1, 2021

Contents

Contents

- Introduction
- Review of DCT and ADCT

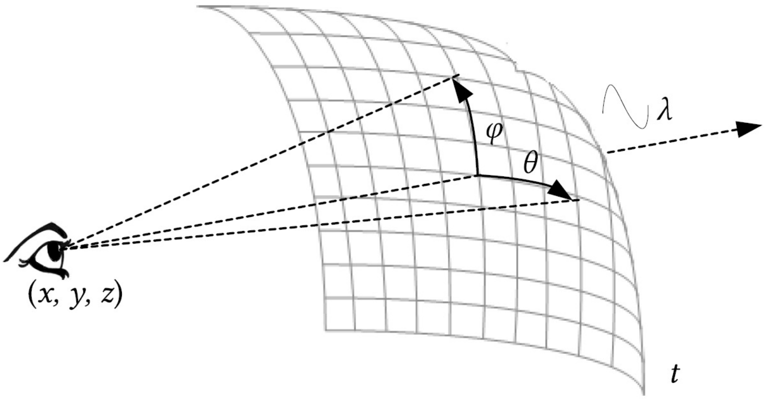
Contents

Contents

Contents

Introduction

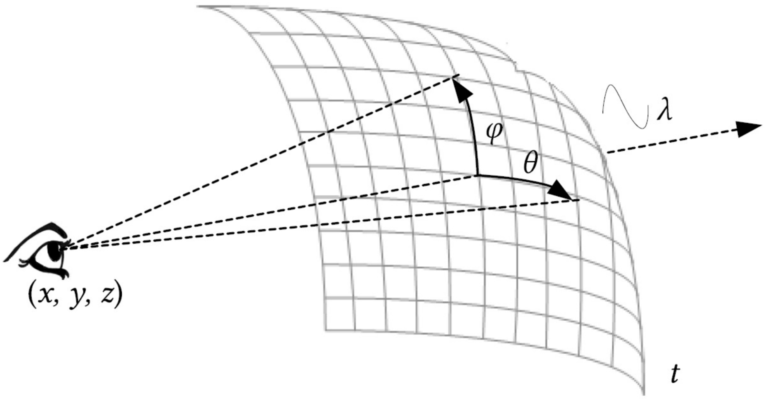
- Light rays emanating from a scene is completely defined by the seven-dimensional plenoptic function [1].



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

Introduction

- Light rays emanating from a scene is completely defined by the seven-dimensional plenoptic function [1].



- Light rays are modelled 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.

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

Introduction

- 5-D light field video (LFV) is a simplified form of the 7-D plenoptic function. It derives with two assumptions [2][3].

[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.

Introduction

- 5-D light field video (LFV) is a simplified form of the 7-D plenoptic function. It derives with two assumptions [2][3].
  - Intensity of a light ray does not change along its direction of propagation (z).

[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.

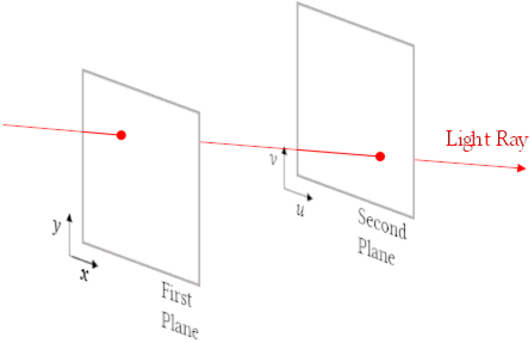
Introduction

- 5-D light field video (LFV) is a simplified form of the 7-D plenoptic function. It derives with two assumptions [2][3].
    - Intensity of a light ray does not change along its direction of propagation (z).
    - Wavelength (λ) is represented by red, green and blue (RGB) color channels. We treat all three channels separately.

[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.

Introduction

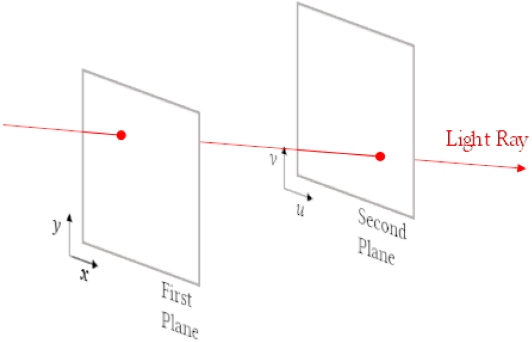
- In general, the spatial and angular dimensions x, y, θ, and ϕ are parameterized using two parallel planes. [4]



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

Introduction

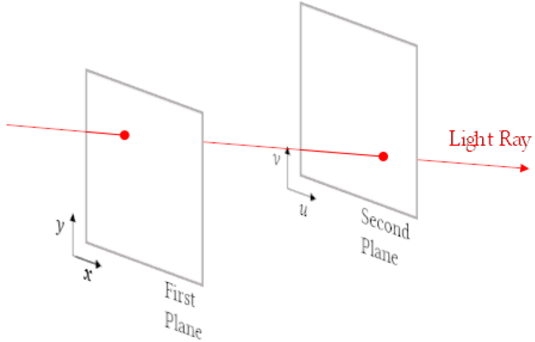
- In general, the spatial and angular dimensions x, y, θ, and ϕ are parameterized using two parallel planes. [4]



- Here light ray intersects the first plane at coordinates (x, y) which defines the spatial location of the ray.

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

- In general, the spatial and angular dimensions x, y, θ, and ϕ are parameterized using two parallel planes. [4]



- Here light ray intersects the first plane at coordinates (x, y) which defines the spatial location of the ray.
- Then it is propagated in free-space until it intersects the second plane at coordinates (u, v) which specifies the propagation direction.

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

Introduction

- A LFV captures spatial, angular and temporal variation of light rays, leading to use in following applications

Introduction

- A LFV captures spatial, angular and temporal variation of light rays, leading to use in following applications
  - Post-capture refocusing [5]
  - Depth estimation [6]
  - Depth-velocity filtering [7]

[5] S. S. Jayaweera, C. Edussooriya, C. Wijenayake, P. Agathoklis, and L. Bruton, "Multi-volumetric refocusing of light fields," TechRxiv, 2020.
[6] 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.
[7] 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.

Introduction

- A LFV captures spatial, angular and temporal variation of light rays, leading to use in following applications
  - Post-capture refocusing [5]
  - Depth estimation [6]
  - Depth-velocity filtering [7]
- But challenge is the data associated with an LFV is significantly larger. e.g., one color channel of an LFV of size $8 \times 8 \times 434 \times 625 \times 16$ requires 2.17 GB, with 8-bits per pixel.

[5] S. S. Jayaweera, C. Edussooriya, C. Wijenayake, P. Agathoklis, and L. Bruton, "Multi-volumetric refocusing of light fields," TechRxiv, 2020.
[6] 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.
[7] 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.

Introduction

- A LFV captures spatial, angular and temporal variation of light rays, leading to use in following applications
  - Post-capture refocusing [5]
  - Depth estimation [6]
  - Depth-velocity filtering [7]
- But challenge is the data associated with an LFV is significantly larger. e.g., one color channel of an LFV of size $8 \times 8 \times 434 \times 625 \times 16$ 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 limited storage and transmission bandwidth requirements are to be met.

[5] S. S. Jayaweera, C. Edussooriya, C. Wijenayake, P. Agathoklis, and L. Bruton, "Multi-volumetric refocusing of light fields," TechRxiv, 2020.
[6] 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.
[7] 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.

Introduction

- Our objective is to propose a low-complexity LFV lossy compression technique using 5-D approximate discrete-cosine transforms (ADCTs)

Introduction

- Our objective is to propose a low-complexity LFV lossy compression technique using 5-D approximate discrete-cosine transforms (ADCTs)
- Here we are using type-2 discrete cosine transform (DCT), which has excellent energy-compaction property and is widely used in data compression applications

Introduction

- Our objective is to propose a low-complexity LFV lossy compression technique using 5-D approximate discrete-cosine transforms (ADCTs)
- Here we are using type-2 discrete cosine transform (DCT), which has excellent energy-compaction property and is widely used in data compression applications
- To the best of my knowledge, this is the *first compression technique proposed for LFVs.*

Review of DCT and ADCT

- 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

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

where

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

Review of DCT and ADCT

- 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

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

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 efficiently, 1-D DCT can be written as:

$$\mathbf{y} = C_N \cdot \mathbf{x}, \tag{2}$$

C_N entries are given by:

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

Review of DCT and ADCT

- 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

$$y[k_1, k_2] = a_N[k_1] \cdot a_N[k_2] \cdot \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} x[n_1, n_2] \cdot \cos\left(\frac{\pi k_1(2n_1+1)}{2N}\right) \cdot \cos\left(\frac{\pi k_2(2n_2+1)}{2N}\right), \tag{3}$$

where

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

Review of DCT and ADCT

- To solve efficiently, 2-D DCT can be written as

$$Y = C_N \cdot X \cdot C_N^T, \tag{4}$$

where C_N entries are given by:

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

Review of DCT and ADCT

- To solve efficiently, 2-D DCT can be written as

$$Y = C_N \cdot X \cdot C_N^T, \tag{4}$$

where 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

$$\hat{C}_N = D_N \cdot T_N, \tag{5}$$

where D_N is diagonal matrix which consists only irrational numbers and T_N is low complexity matrix which consists only zeros and powers of 2 $\{\frac{1}{2}, 0, 1, 2\}$.

Review of DCT and ADCT

E.g. Bouguezel-Ahmad-Swamy Approximate DCT (BAS 2008) [8]

$$\hat{C}_1 = D_1 \cdot T_1, \tag{6}$$

where $T_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -1 & \frac{1}{2} \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}$

$D_1 = diag(1/\sqrt{8}, 1/\sqrt{4}, 1/\sqrt{5}, 1/\sqrt{2}, 1/\sqrt{8}, 1/\sqrt{4}, 1/\sqrt{5}, 1/\sqrt{2})$

[8] 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.

Review of DCT and ADCT

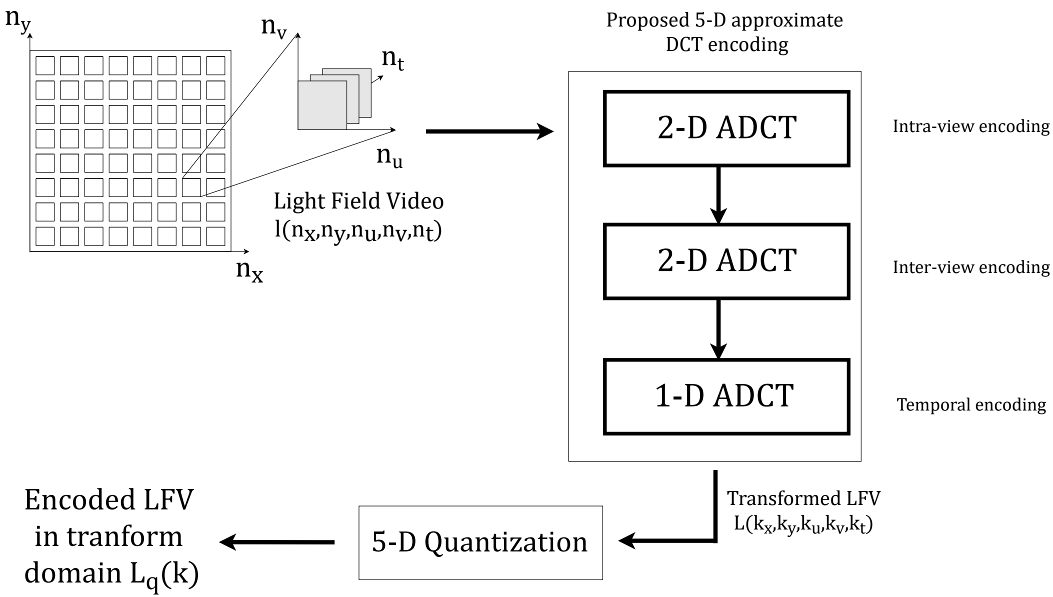
Table 1: Arithmetic Complexity

DCT method	Mult	Add	Shifts	Total
Exact DCT	64	56	0	120
BAS2008 [8]	0	18	2	20
BAS2011 with a = 0 [9]	0	16	0	16
BAS2011 with a = 1 [9]	0	18	0	18
CB2011 [10]	0	22	0	22
Modified CB2011 [11]	0	14	0	14
PMC2014 [12]	0	14	0	14

[9] "A low-complexity parametric transform for image compression." IEEE, 2011, pp. 2145–2148.
[10] R. J. Cintra and F. M. Bayer, "A dct approximation for image compression," IEEE Signal Processing Letters, vol. 18, pp. 579–582, 2011.
[11] 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.
[12] 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.

Proposed 5-D ADCT LFV compression

- Overview of Proposed 5-D ADCT Based LFV Compression



Proposed 5-D ADCT LFV compression

- Input to our system is $l(n_x, n_y, n_u, n_v, n_t)$, which size is $(N_x, N_y, N_u, N_v, N_t)$, indicates there are N_t number of frames in LFV and each 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.

Proposed 5-D ADCT LFV compression

- Input to our system is $l(n_x, n_y, n_u, n_v, n_t)$, which size is $(N_x, N_y, N_u, N_v, N_t)$, indicates there are N_t number of frames in LFV and each 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.
- In order to further reduce the computational complexity, our technique exploits the partial separability of LFV representations [13].

[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.

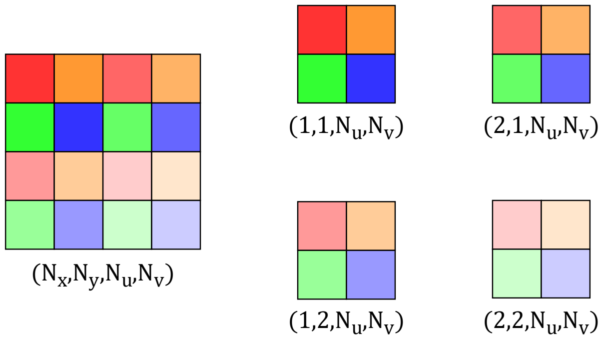
Proposed 5-D ADCT LFV compression

- Input to our system is $l(n_x, n_y, n_u, n_v, n_t)$, which size is $(N_x, N_y, N_u, N_v, N_t)$, indicates there are N_t number of frames in LFV and each 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.
- In order to further reduce the computational complexity, our technique exploits the partial separability of LFV representations [13].
- LFV compression has been done for each channel separately.

[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.

Proposed 5-D ADCT LFV compression

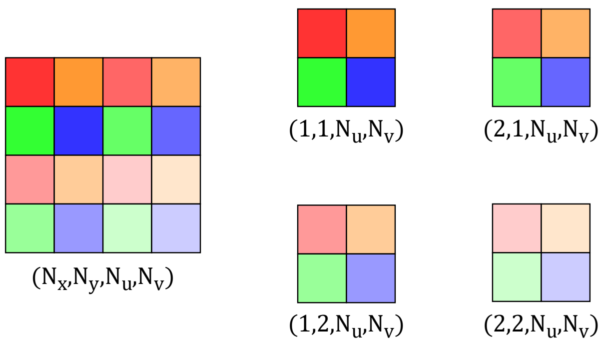
Step 1 – Intra-view encoding



- 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.

Proposed 5-D ADCT LFV compression

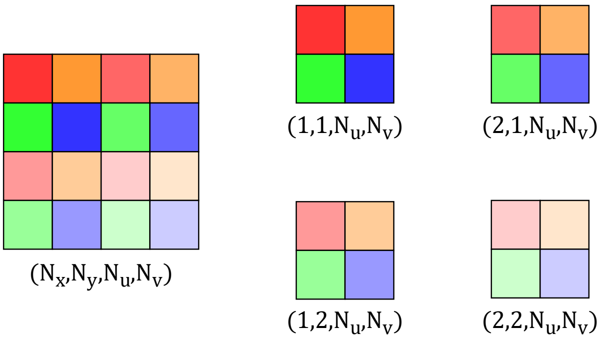
Step 1 – Intra-view encoding



- 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.
- $N_x \times N_y \times N_u/8 \times N_v/8 \times N_t$ no of operations needed to complete intra-view ADCT transformation on entire LFV.

Proposed 5-D ADCT LFV compression

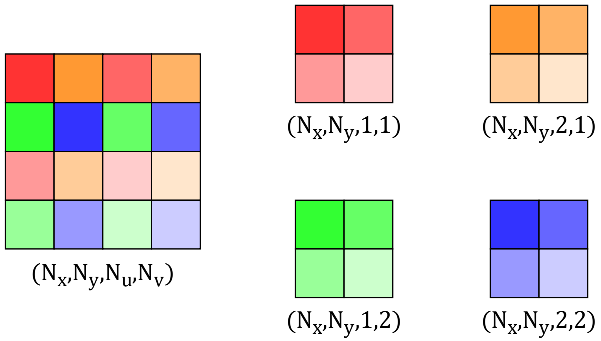
Step 1 – Intra-view encoding



- 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.
- $N_x \times N_y \times N_u/8 \times N_v/8 \times N_t$ no of operations needed to complete intra-view ADCT transformation on entire LFV.
- This leads to create mixed domain 5-D signal $L_1(n_x, n_y, k_u, k_v, n_t)$.

Proposed 5-D ADCT LFV compression

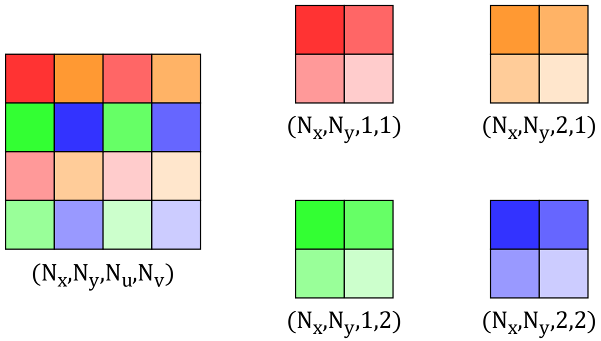
Step 2 – Inter-view encoding



- 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.

Proposed 5-D ADCT LFV compression

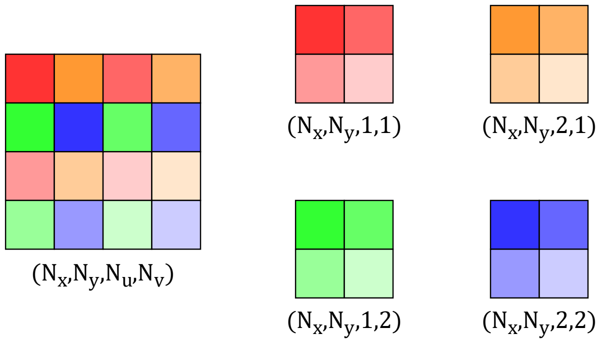
Step 2 – Inter-view encoding



- 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.
- $N_x/8 \times N_y/8 \times N_u \times N_v \times N_t$ no of operations needed to complete inter-view ADCT transformation on entire LFV.

Proposed 5-D ADCT LFV compression

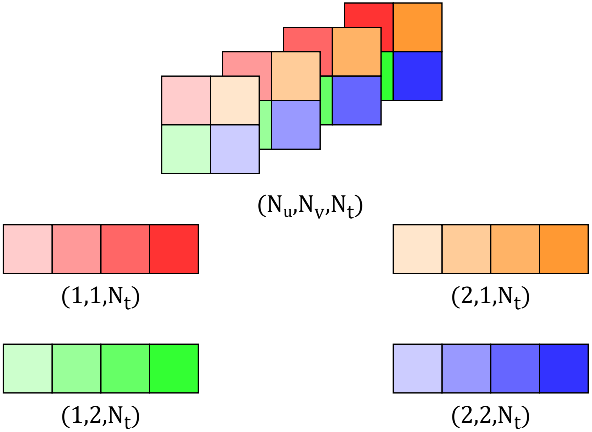
Step 2 – Inter-view encoding



- 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.
- $N_x/8 \times N_y/8 \times N_u \times N_v \times N_t$ no of operations needed to complete inter-view ADCT transformation on entire LFV.
- This leads to create mixed domain 5-D signal $L_2(k_x, k_y, k_u, k_v, n_t)$.

Proposed 5-D ADCT LFV compression

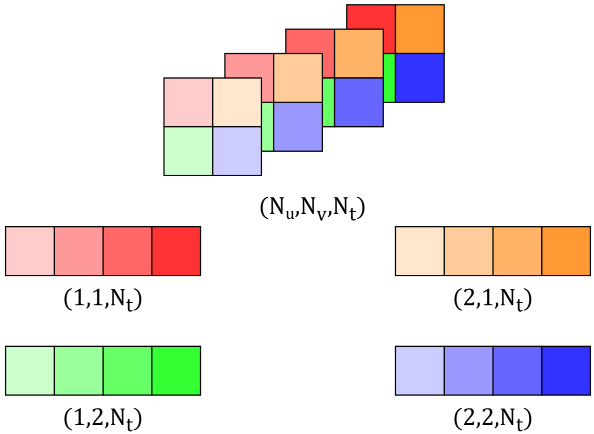
Step 3 – Temporal encoding



- 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.

Proposed 5-D ADCT LFV compression

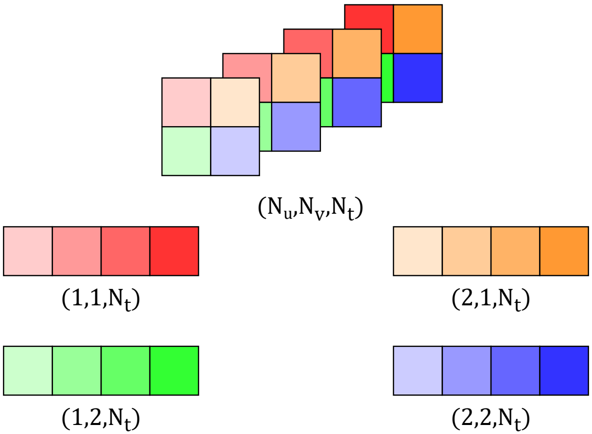
Step 3 – Temporal encoding



- 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.
- $N_x \times N_y \times N_u \times N_v \times N_t/8$ no of operations are needed to complete temporal ADCT transformation on entire LFV.

Proposed 5-D ADCT LFV compression

Step 3 – Temporal encoding



- 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.
- $N_x \times N_y \times N_u \times N_v \times N_t/8$ no of operations are needed to complete temporal ADCT transformation on entire LFV.
- This creates fully transformed 5-D signal $L(k_x, k_y, k_u, k_v, k_t)$.

Proposed 5-D ADCT LFV compression

Step 4 - Quantization

- 5-D signal $L(k_x, k_y, k_u, k_v, k_t)$ is partitioned into $8 \times 8 \times 8 \times 8 \times 8$ blocks and apply $8 \times 8 \times 8 \times 8 \times 8$ constant value matrix on it.

$$L_q(k) = round\left(\frac{L(k)}{Q(k)}\right) \cdot Q(k) \tag{7}$$

Proposed 5-D ADCT LFV compression

Step 4 - Quantization

- 5-D signal $L(k_x, k_y, k_u, k_v, k_t)$ is partitioned into $8 \times 8 \times 8 \times 8 \times 8$ blocks and apply $8 \times 8 \times 8 \times 8 \times 8$ constant value matrix on it.

$$L_q(k) = round\left(\frac{L(k)}{Q(k)}\right) \cdot Q(k) \tag{7}$$

- Here $Q(k)$ is 5-D constant value matrix and both division and multiplication are done in element wise.

Proposed 5-D ADCT LFV compression

Step 4 - Quantization

- 5-D signal $L(k_x, k_y, k_u, k_v, k_t)$ is partitioned into $8 \times 8 \times 8 \times 8 \times 8$ blocks and apply $8 \times 8 \times 8 \times 8 \times 8$ constant value matrix on it.

$$L_q(k) = round\left(\frac{L(k)}{Q(k)}\right) \cdot Q(k) \tag{7}$$

- Here $Q(k)$ 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.

Experimental Results

Compression quality

- Compression quality is measured by using
  - PSNR - Peak Signal to Noise Ratio
  - SSIM - Structural Similarity Index for Measuring image quality

Experimental Results

Compression quality

- Compression quality is measured by using
  - PSNR - Peak Signal to Noise Ratio
  - SSIM - Structural Similarity Index for Measuring image quality
- The PSNR between the original SAI A and the reconstructed SAI A' is computed as follows

$$PSNR = 10\log_{10}\left(\frac{2^n - 1}{MSE}\right), \tag{8}$$

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

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

Experimental Results

- The SSIM between the original SAI A and the reconstructed SAI A' is computed as follows

$$SSIM = \frac{(2\mu_A \mu_{A'} + C_1)(2\sigma_{AA'} + C_2)}{(\mu_A^2 + \mu_{A'}^2 + C_1)(\sigma_A^2 + \sigma_{A'}^2 + C_2)}, \tag{9}$$

  Where $\mu_A, \mu_{A'}, \sigma_A, \sigma_{A'}, , \sigma_{AA'}$ are local means, standard deviations, and cross covariance for SAIs A, A'.

Experimental Results

- The SSIM between the original SAI A and the reconstructed SAI A' is computed as follows

$$SSIM = \frac{(2\mu_A\mu_{A'} + C_1)(2\sigma_{AA'} + C_2)}{(\mu_A^2 + \mu_{A'}^2 + C_1)(\sigma_A^2 + \sigma_{A'}^2 + C_2)}, \tag{9}$$

  Where $\mu_A, \mu_{A'}, \sigma_A, \sigma_{A'}, , \sigma_{AA'}$ are local means, standard deviations, and cross covariance for SAIs A, A'.
- Final PSNR and SSIM are calcualted as [14]

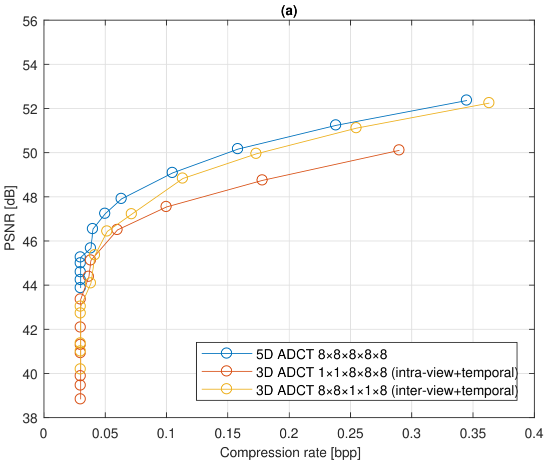
$$PSNR_{YCbCr} = \frac{6PSNR_Y + PSNR_{Cb} + PSNR_{Cr}}{8} \tag{10}$$

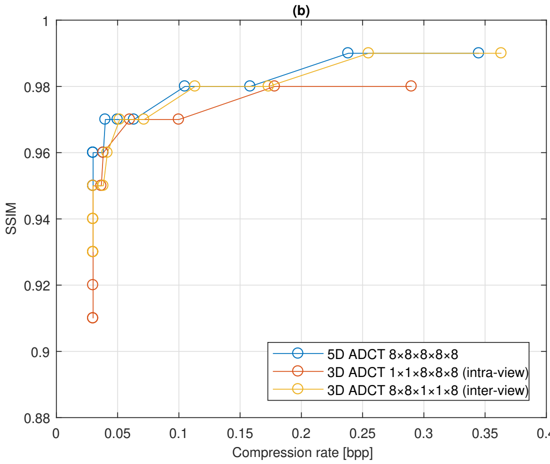
$$SSIM_{YCbCr} = SSIM_Y \tag{11}$$

[14] 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.

Experimental Results

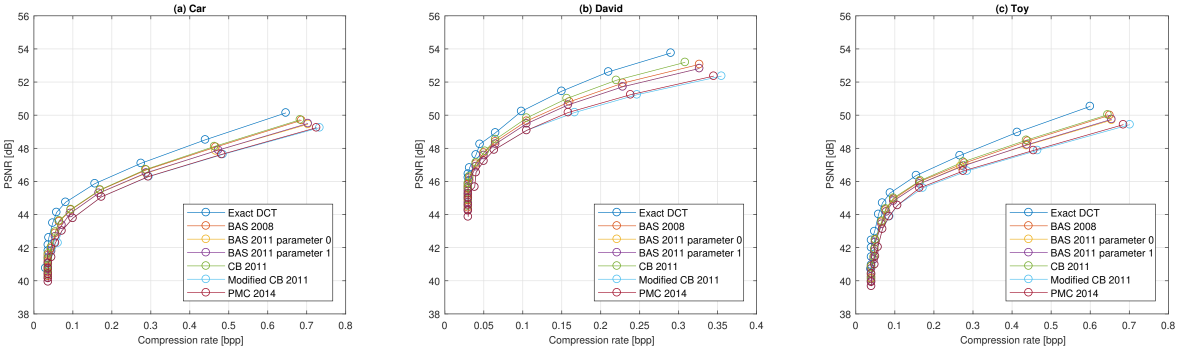
(a) PSNR vs Compression rate (b) SSIM vs Compression rate for 5-D ADCT, intra-view + temporal only and inter-view + temporal only





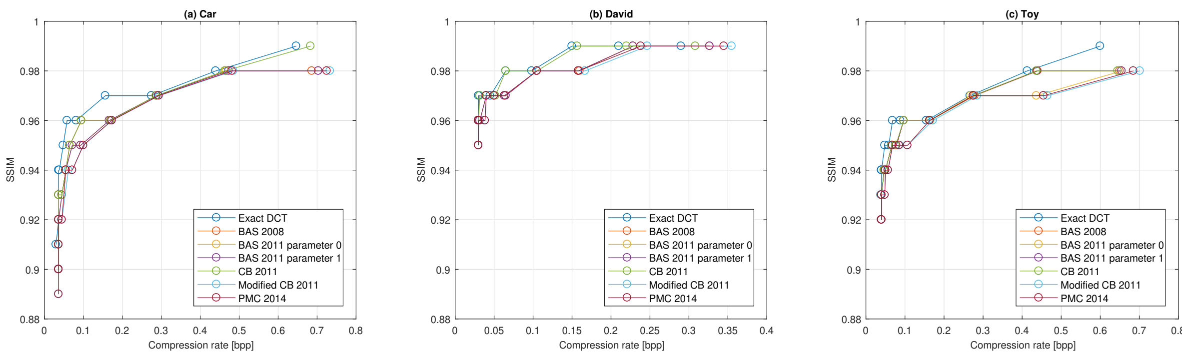
Experimental Results

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



Experimental Results

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

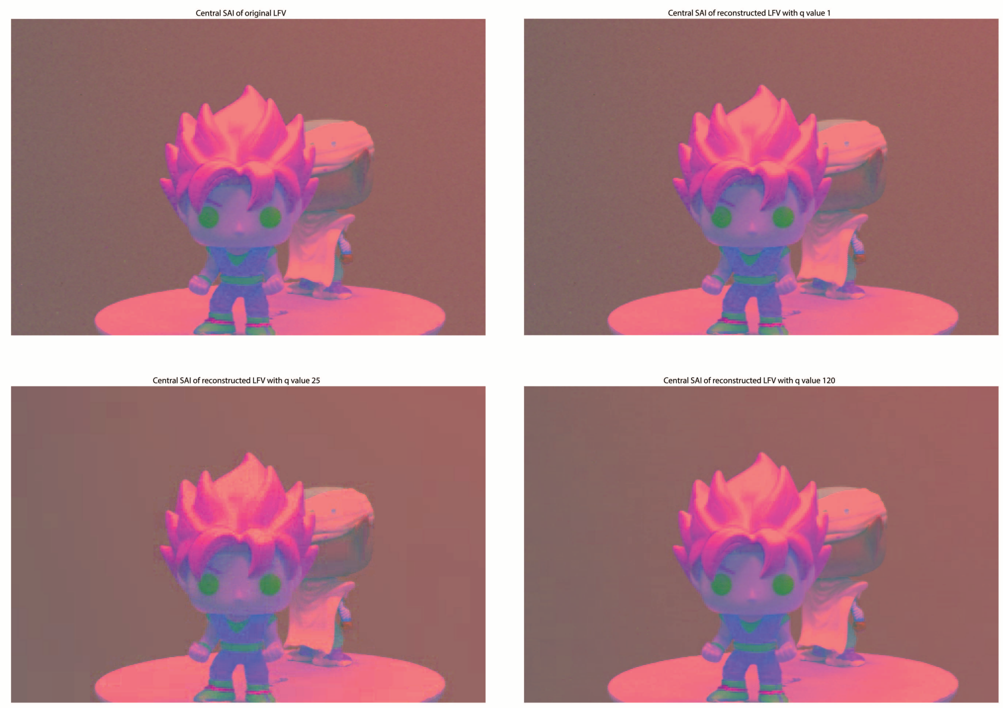


Experimental Results

Table 2: Quantization

Quantization value	Compression rate [bpp]	PSNR	SSIM	Energy retained
1	1.33	62.32	1	1
5	0.28	53.53	0.99	1
10	0.14	51.26	0.99	0.99
25	0.06	48.76	0.98	0.98
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

Central Sub Aperture Image of (a) Original LFV frame (b) Reconstructed LFV frame with quantization value 1 (c) quantization value 25 (d) quantization value 120



Conclusion and Future work

- LFV compression technique using low-complexity 5-D approximate discrete-cosine transforms (ADCTs) is proposed to do compression in near-real time.

Conclusion and Future work

- LFV compression technique using low-complexity 5-D approximate discrete-cosine transforms (ADCTs) is proposed to do compression in near-real time.
- We evaluate the performance of the proposed LFV compression technique using several 5-D ADCT algorithms, and the exact 5-D DCT.

Conclusion and Future work

- LFV compression technique using low-complexity 5-D approximate discrete-cosine transforms (ADCTs) is proposed to do compression in near-real time.
- 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 0.89.

Conclusion and Future work

- 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.

Conclusion and Future work

- 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.
- Future work includes 5-D light field video compression implementation in FPGA device and 5-D light field video compression using prediction.

THANK YOU