A 4D DCT-Based Lenslet Light Field Codec

Eduardo A. B. da Silva

Márcio P. Pereira

Gustavo Alves

Carla Pagliari

Murilo B. de Carvalho

Fernando Pereira

Vanessa Testoni

Signal, Multimedia and Telecommunications Lab, COPPE/Poli/UFRJ

Instituto Militar de Engenharia

Universidade Federal Fluminense

Instituto Superior Técnico, Universidade de Lisboa - Instituto de Telecomunicações

Samsung Research Brazil

Outline

1. Context and Objectives
2. 4D DCT-Based Lenslet Light Field Codec
3. Performance Assessment
4. Conclusions

1. Context and Objectives

Context

- Light field describes light flowing in every direction through every point in space
- 2D camera array and plenoptic photography (lenslets) are capture systems for light fields
- Light field representations need a huge amount of data
- State-of-the art coding approaches:
  - Make use of prediction, using view synthesis and depth maps

Context

- Codec was applied to the lenslets light fields datasets from the JPEG-Pleno Call for Proposals
- Luminance was encoded using transforms of dimensions 13x13x15x15 (13x13 views, each one divided into 15x15 blocks)
- Chrominance was encoded using transforms of dimensions 13x13x8x8 (13x13 views, each one divided into 8x8 blocks)
- Spatial blocks that are smaller than 15x15 for luminance or 8x8 for chrominance were extended using simple pixel repetition.
- This work is a contribution and is in the context of JPEG Pleno

Objectives

- Explore an alternative approach using light fields intrinsic 4D redundancy

2. 4D DCT-Based Lenslet Light Field Codec

Block Diagram

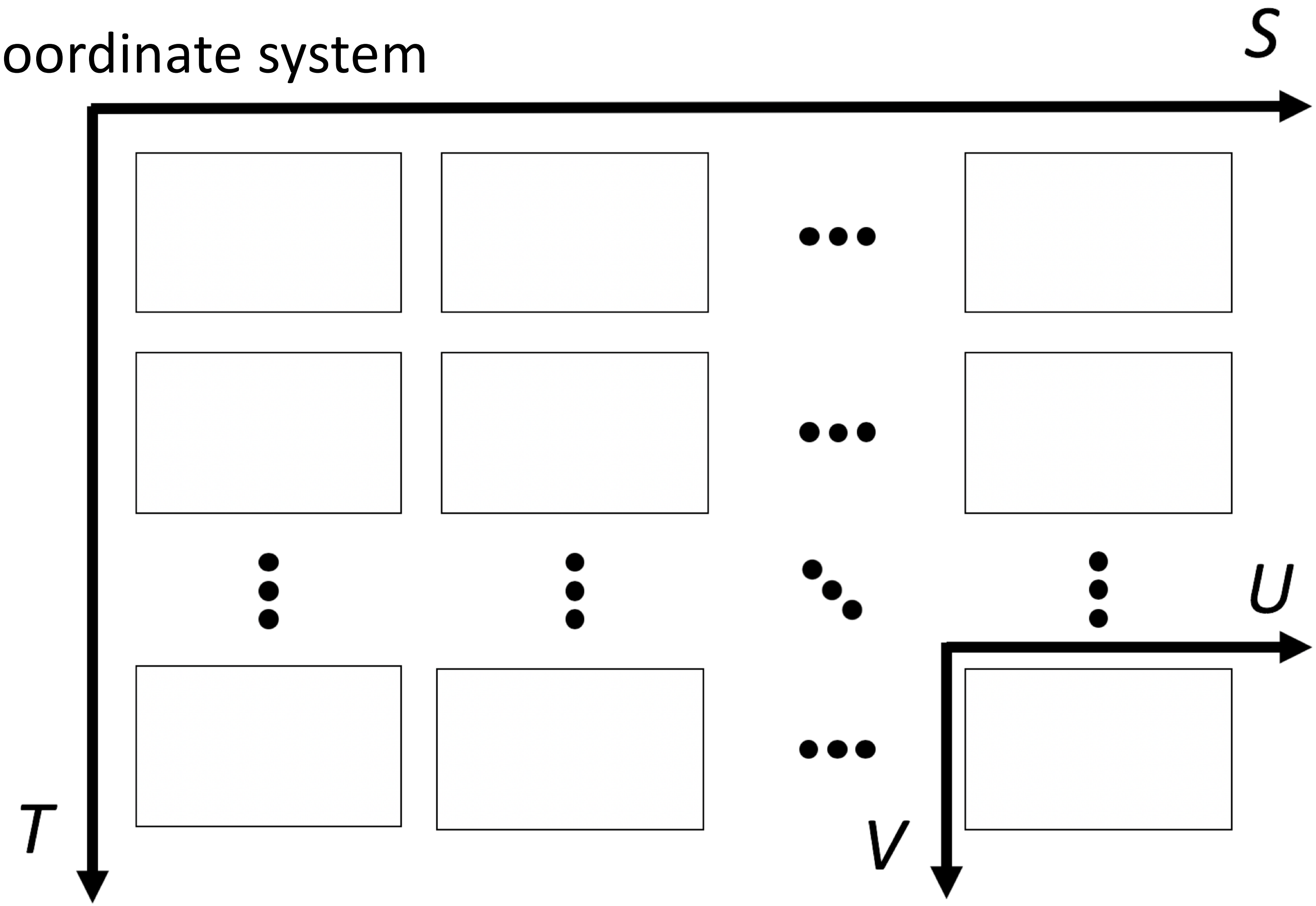
- A simple way to explore light fields 4D redundancy:
  - 4D DCT transform
  - Hexadeca-tree bitplane clustering
  - Entropy coding

  - We refer to it as MuLE-TH (Multidimensional Light Field Encoder based on Hexadeca-Tree Bitplanes)

4D Transform

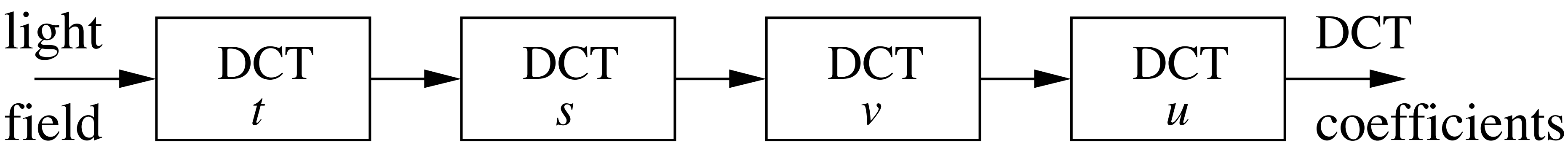
4D Transform

- Coordinate system

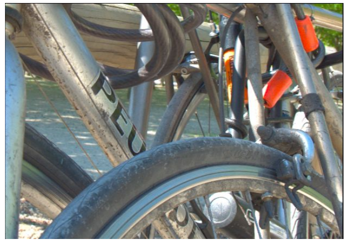


4D Transform

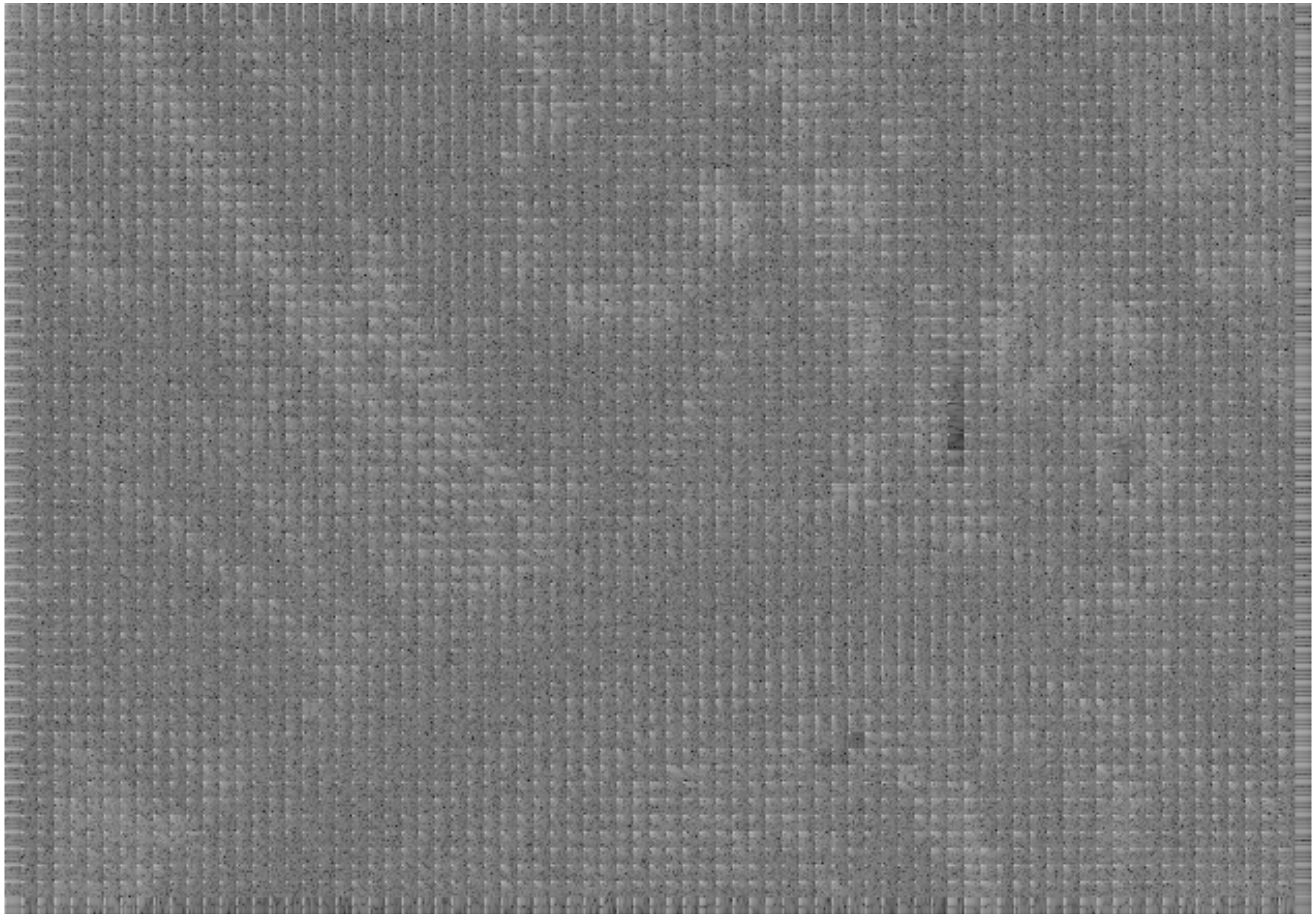
- Subtract from the light field half its dynamic range (in the case of 10-bit light fields, 1024/2).
- Compute a 4D DCT ($t \times s \times v \times u$),
- Represent it as 32-bit integers
- Group them in subbands
  - A 16 × 16 × 512 × 512 light field transformed by an 8 × 8 × 8 × 8 transforms is grouped as 4096 (8^4) subbands, each with 16/8 × 16/8 × 512/8 × 512/8 coefficients.



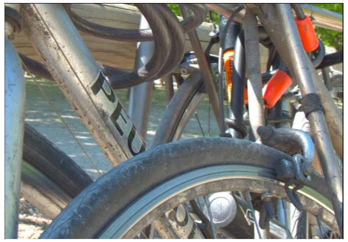
4D DCT as blocks of coefficients



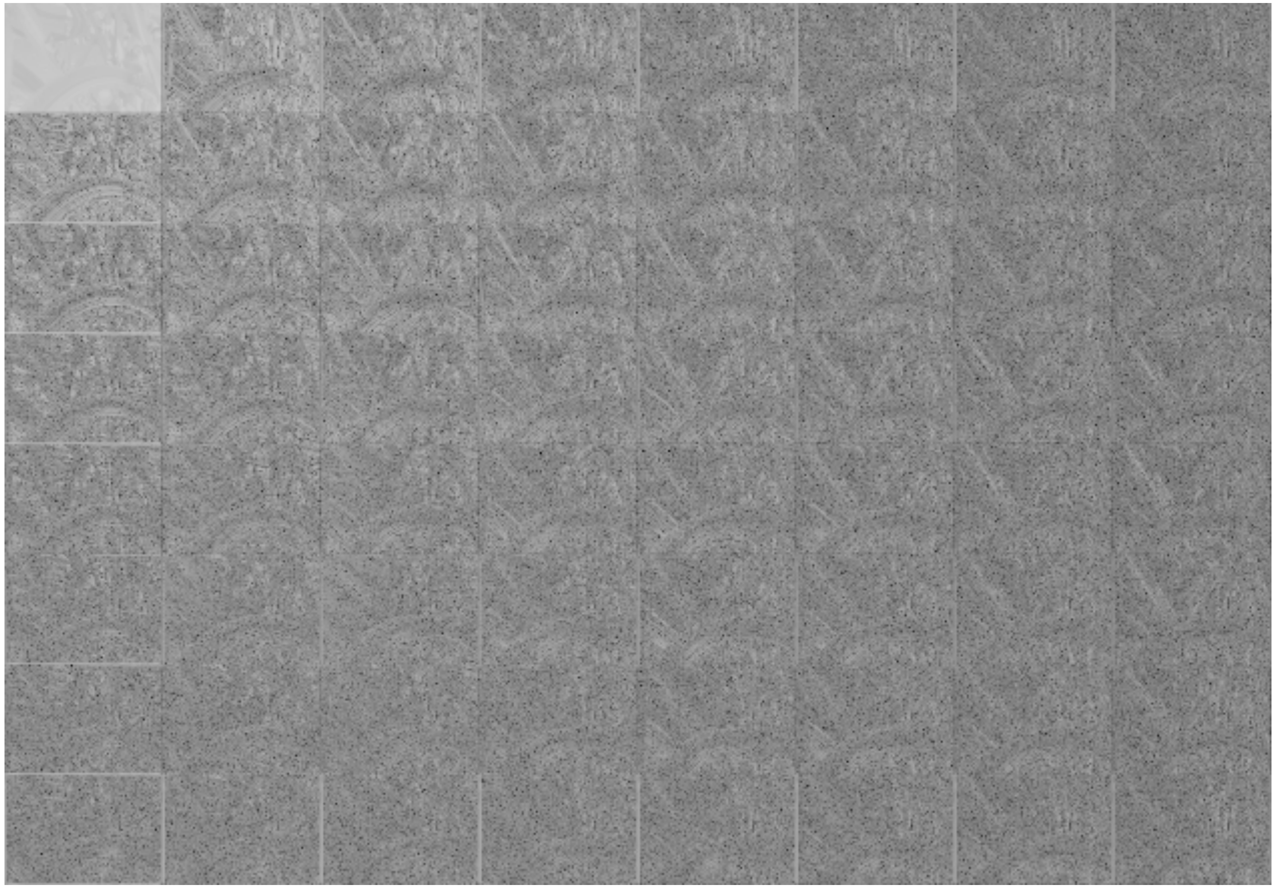
4D DTC 8 × 8 × 8 × 8 ($t \times s \times v \times u$)



4D DCT grouped as subbands

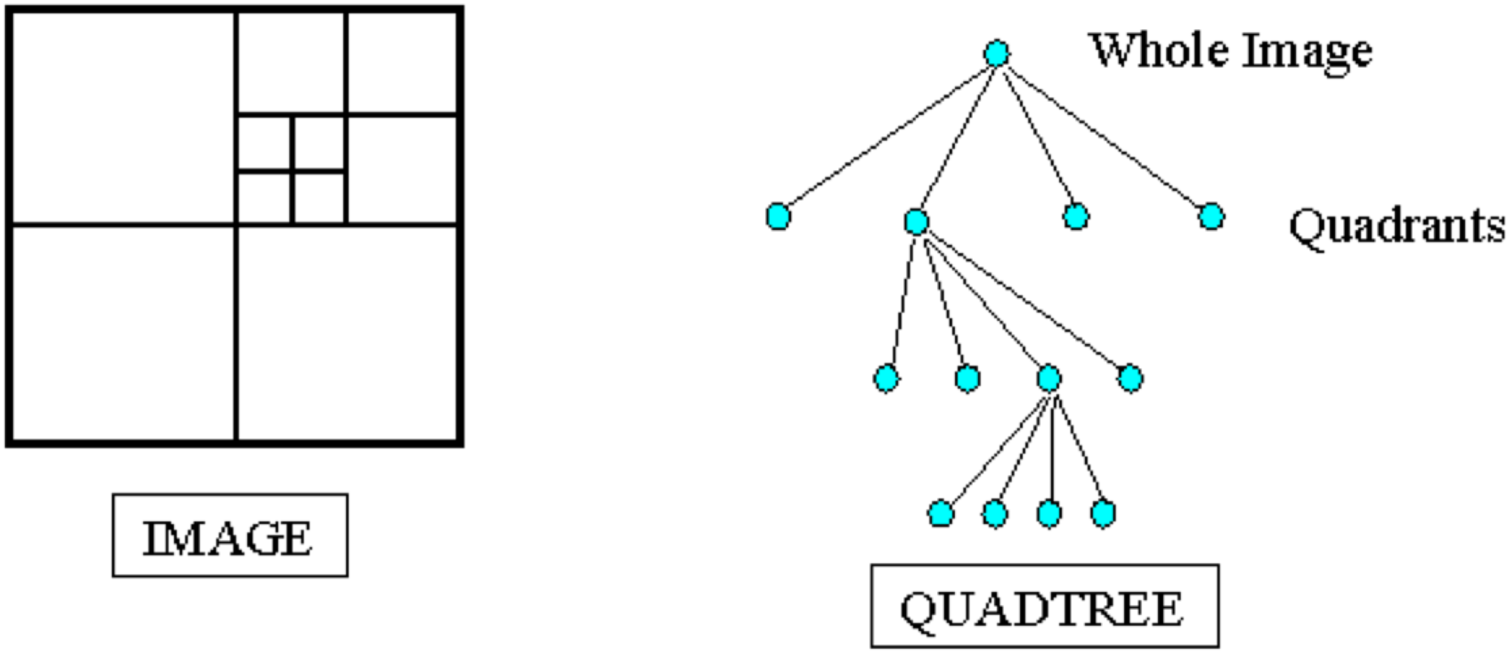


4D DCT 8 × 8 × 8 × 8 ($t \times s \times v \times u$) – grouped into subbands

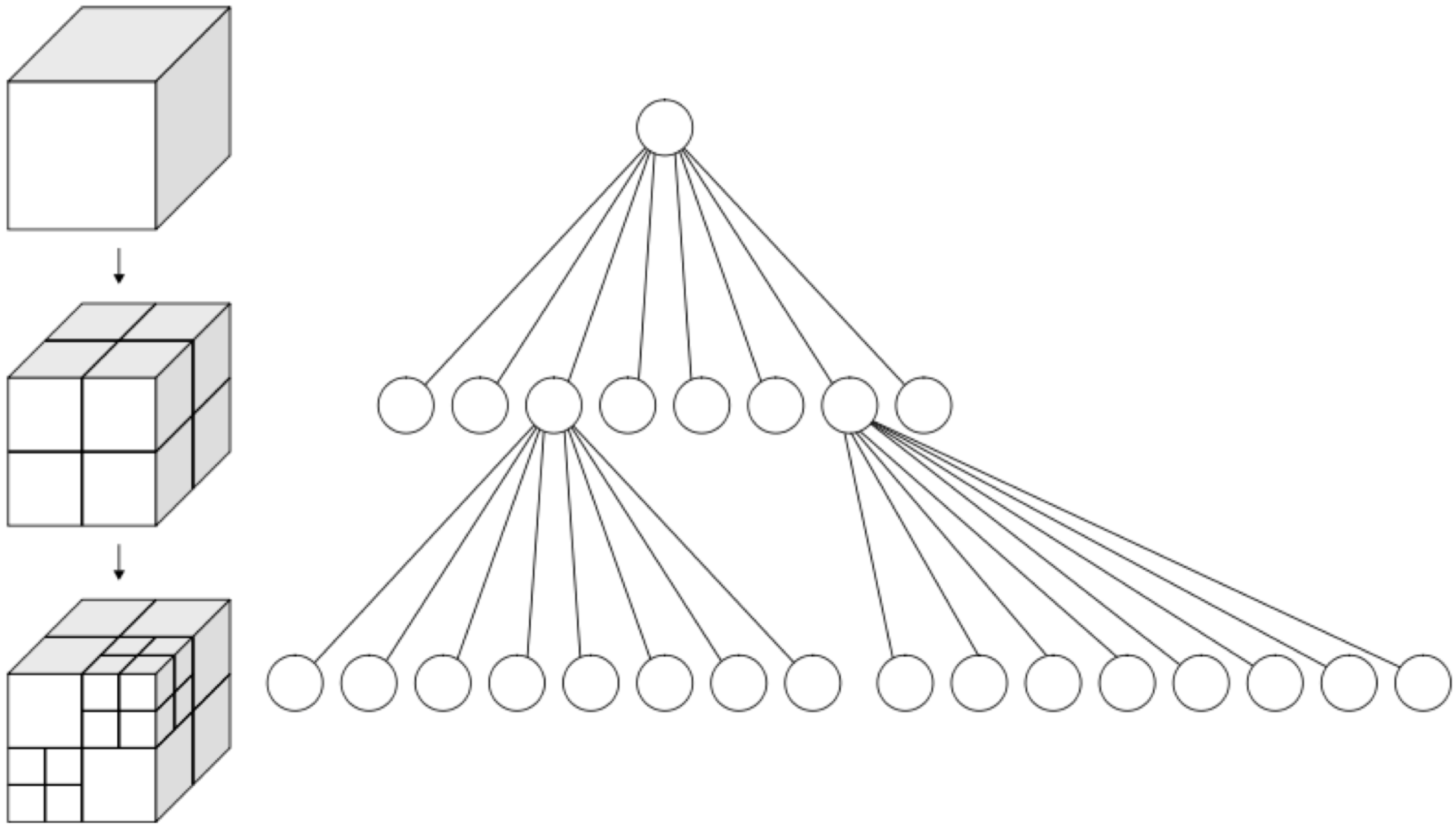


Hexadeca-tree Bitplanes

Quadtrees



Octrees



Hexadeca-trees

- Extension to 4D of quadtrees and octrees.
- Used to locate the non-zero 4D DCT coefficients
- Does this by bitplanes, starting from the highest (most significant).
- How?

Hexadeca-tree bitplane encoding

1. Set the precision to be used (number of bits)
2. Starts from the coefficients that are non-zero at the most significant bitplane
   – It is composed by the coefficients whose magnitude is larger than the threshold, in this case equal to half the dynamic range of the coefficients
3. Partition the light field in hexadeca-trees until the coefficients that are non-zero at the current bitplane are located.
   – This provides joint coding for the leading zeros for each bitplane
   – Very efficient in 4D.
4. Once these are located, send their binary representation at the precision set in step 1, including their sign, to the entropy coder.
5. Go to the next bitplane (less significant)
6. Repeats the process recursively starting from each subimage generated by the hexadeca-tree from step 3.
7. Finish when all the non-zero coefficients at the set precision have been encoded.

Entropy Coder

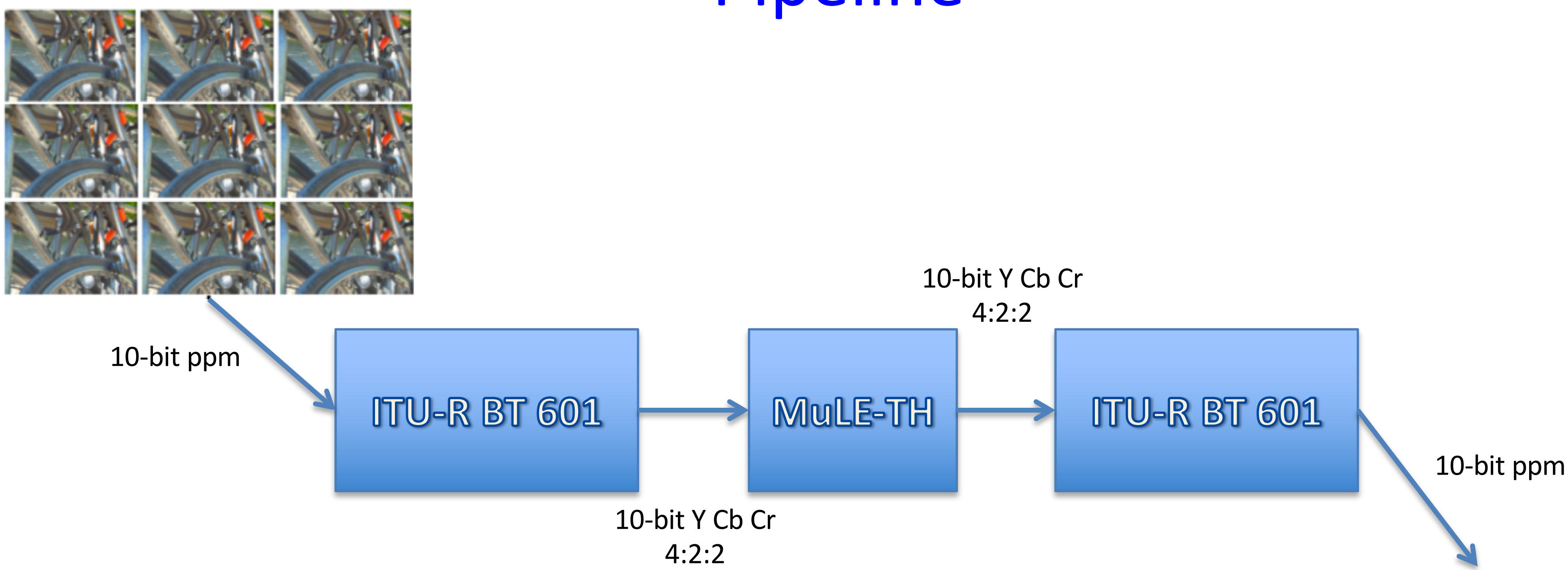
Entropy Coder

Uses a context-based adaptive arithmetic coder

- One binary context for significance flags (indicating hexadeca-tree partitions) by bit-plane.
- One non-binary context for DC coefficients per bit plane
  - With as many symbols as DC coefficients up to 512 symbols. For more than 512 symbols, the remaining least significant bits are encoded using an equally likely binary context.
- One non-binary context for AC coefficients per bit plane
  - A similar strategy as for encoding the DC coefficients is used.

3. Performance Assessment

Pipeline



- Assessment:

  PSNRyuv = (6*PSNRy+PSNRu+PSNRv)/8

  SSIMyuv = (6*SSIMy +SSIMu +SSIMv)/8

Dataset

- JPEG Pleno lenslet dataset
- 15x15 views of 626x434 pixels
- From left to right: Bikes, Danger, Pillars, Fountain and Friends

MuLE-TH Performance Assessment

- MuLE-TH codec was applied to the lenslets light fields datasets from the JPEG-Pleno Call for Proposals
- Luminance was encoded using transforms of dimensions 13x13x15x15 (13x13 views, each one divided into 15x15 blocks)
- Chrominance was encoded using transforms of dimensions 13x13x8x8 (13x13 views, each one divided into 8x8 blocks)
- Spatial blocks that are smaller than 15x15 for luminance or 8x8 for chrominance were extended using simple pixel repetition.

MuLE-TH Performance Assessment

- PSNR-Y, PSNR-YUV, SSIM-Y and SSIM-YUV have been computed for rates approximately between 0.005 bpp and 0.75 bpp

Bjontegaard BD-rate relative to HEVC

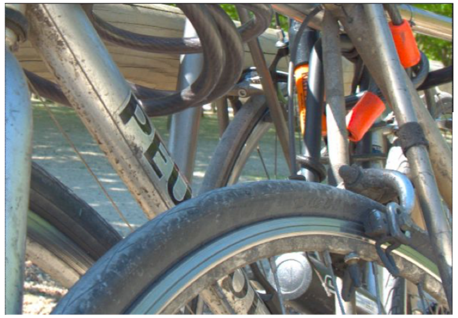
- Comparison only with methods with available 4:2:2 results
- HEVC and VP9
  - Pseudo temporal serpentine scan order of views
  - HEVC codec: x265 v2.3

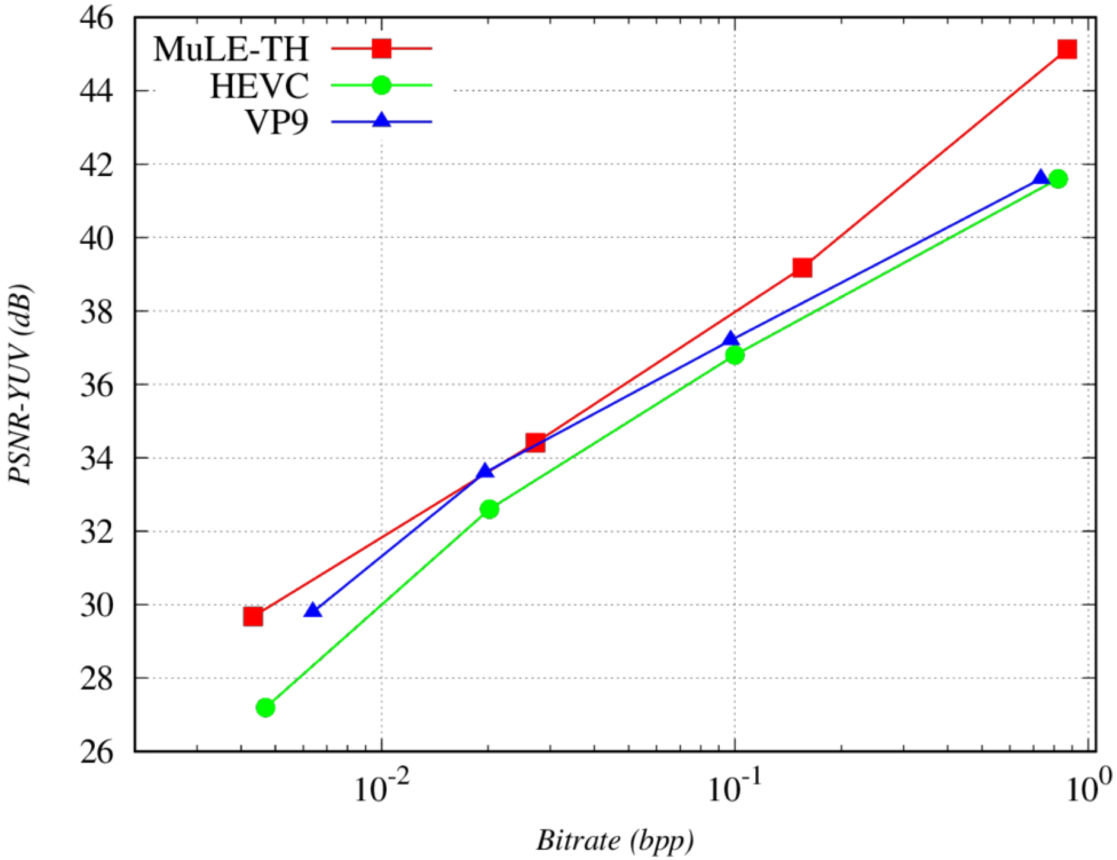
	Bikes	Danger	Pillars	Fountain	Friends
VP9	-22.52	-18.51	-21.71	-18.07	-21.92
USTC codec[1]	-21.00	NA	NA	NA	NA
TUT codec[2]	-26.74	-84.52	-51.76	-9.05	-10.06
MuLE-TH	**-39.11**	**-53.19**	**-38.74**	**-24.73**	**-34.83**

(1) – Shegyang Zhao, Zhibo Chen, "Light field image coding via linear approximation prior," in IEEE International Conference on Image Processing 2017 – Light Field Coding Grand Challenge, Beijing-China, September 2017
(2) – Ioan Tabus, Petri Helin, Pekka Astola, "Lossy compression of lenslet images from plenoptic cameras combining sparse predictive coding and JPEG 2000," in IEEE International Conference on Image Processing 2017 – Light Field Coding Grand Challenge, Beijing-China, September 2017

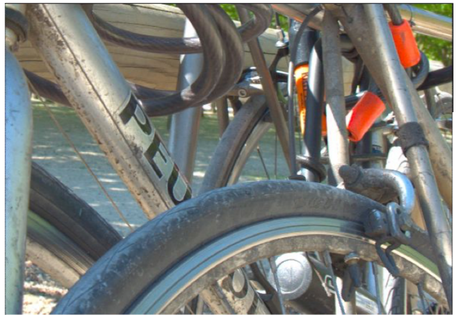
PSNR and SSIM Results

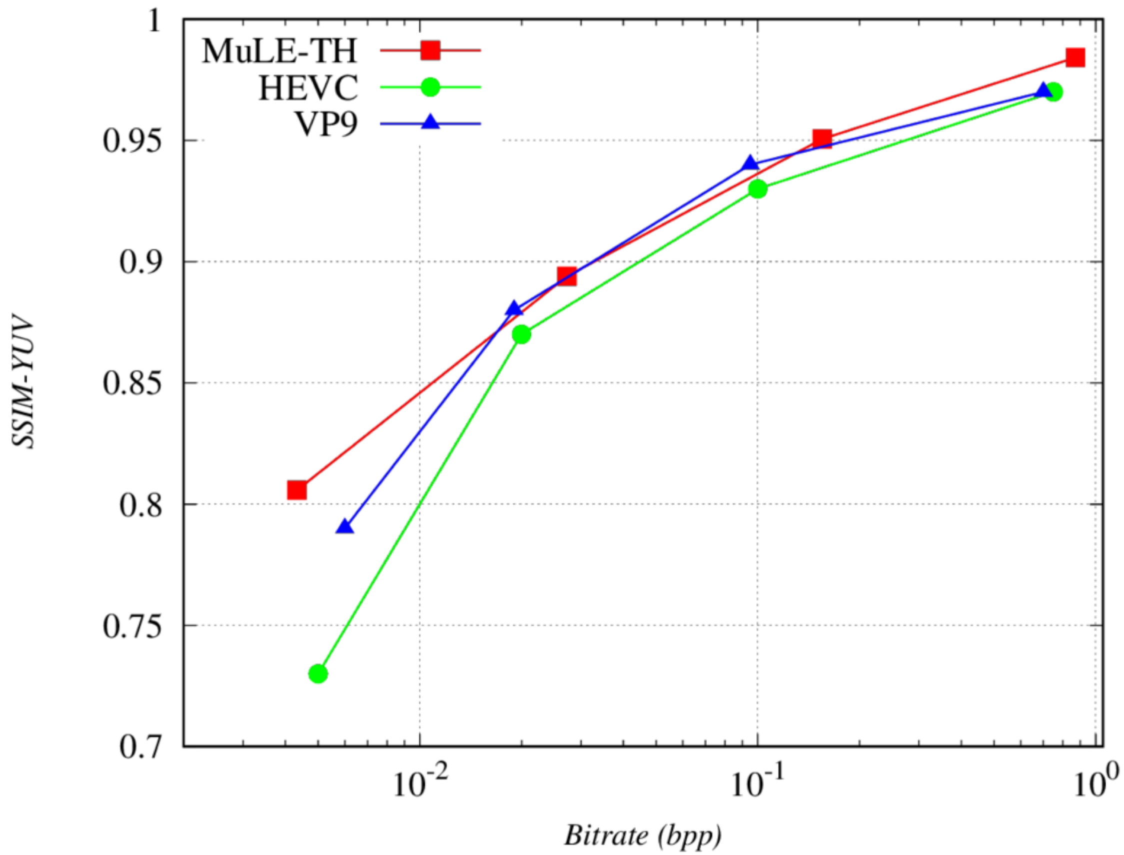
Bikes PSNR YUV



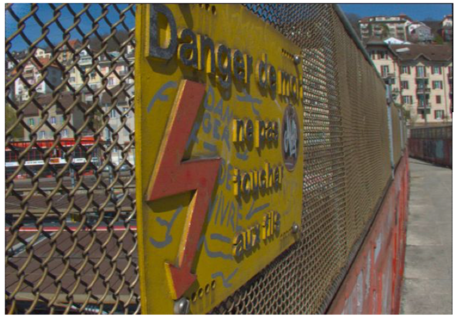


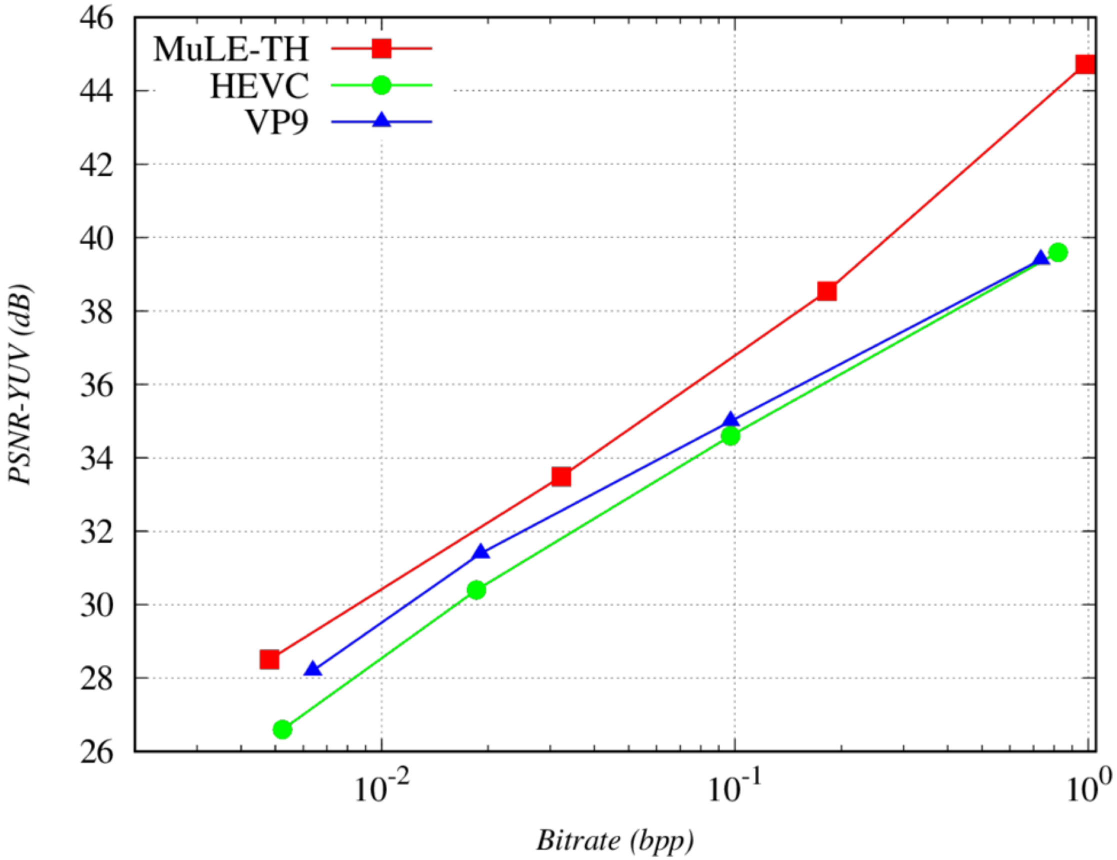
Bikes SSIM YUV



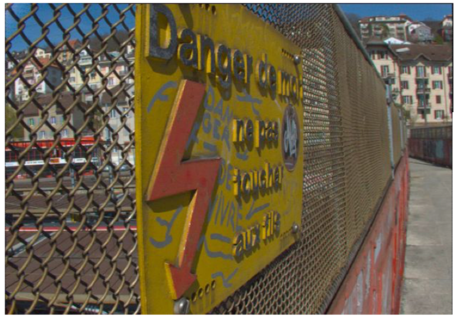


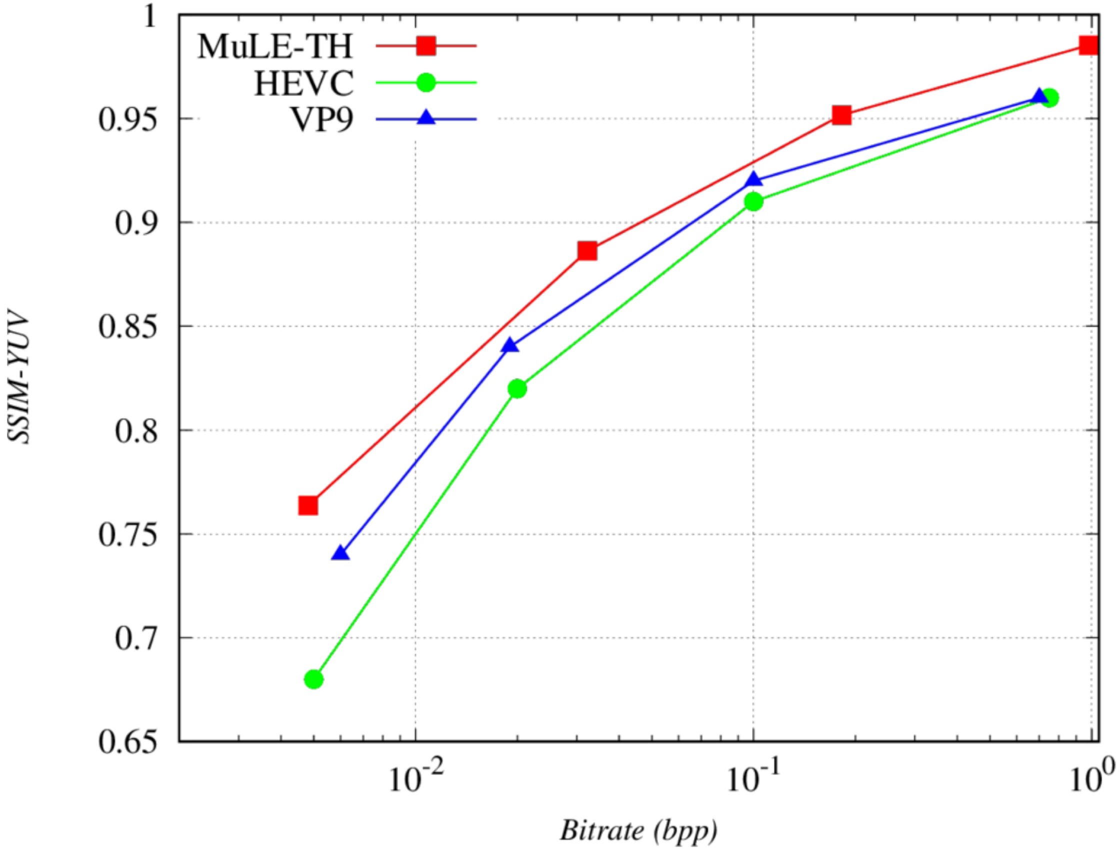
Danger PSNR YUV



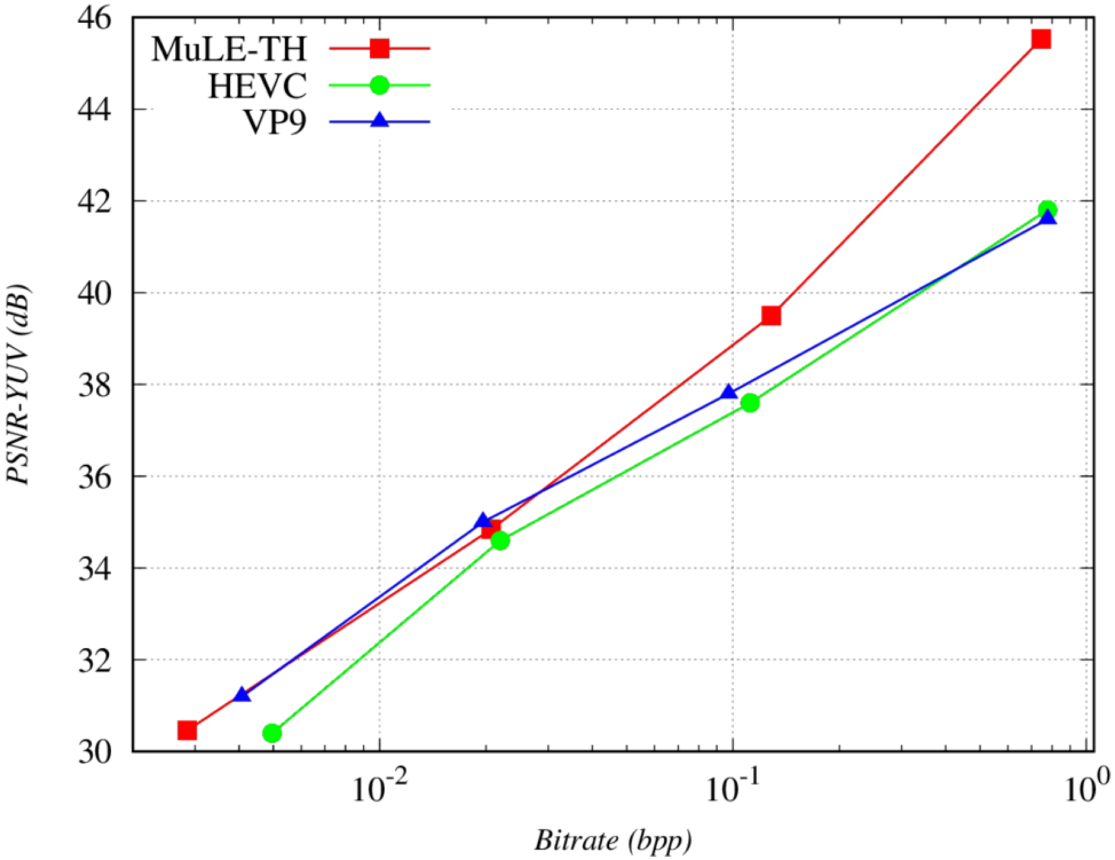


Danger SSIM YUV

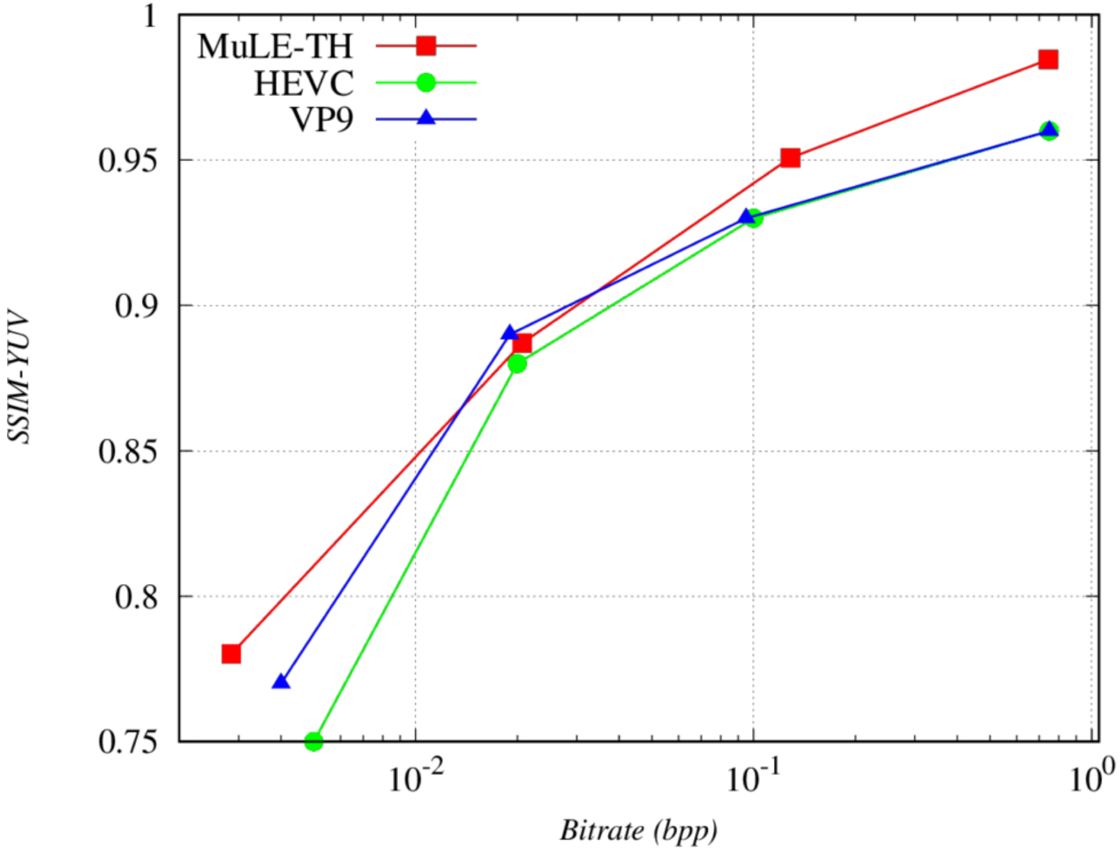




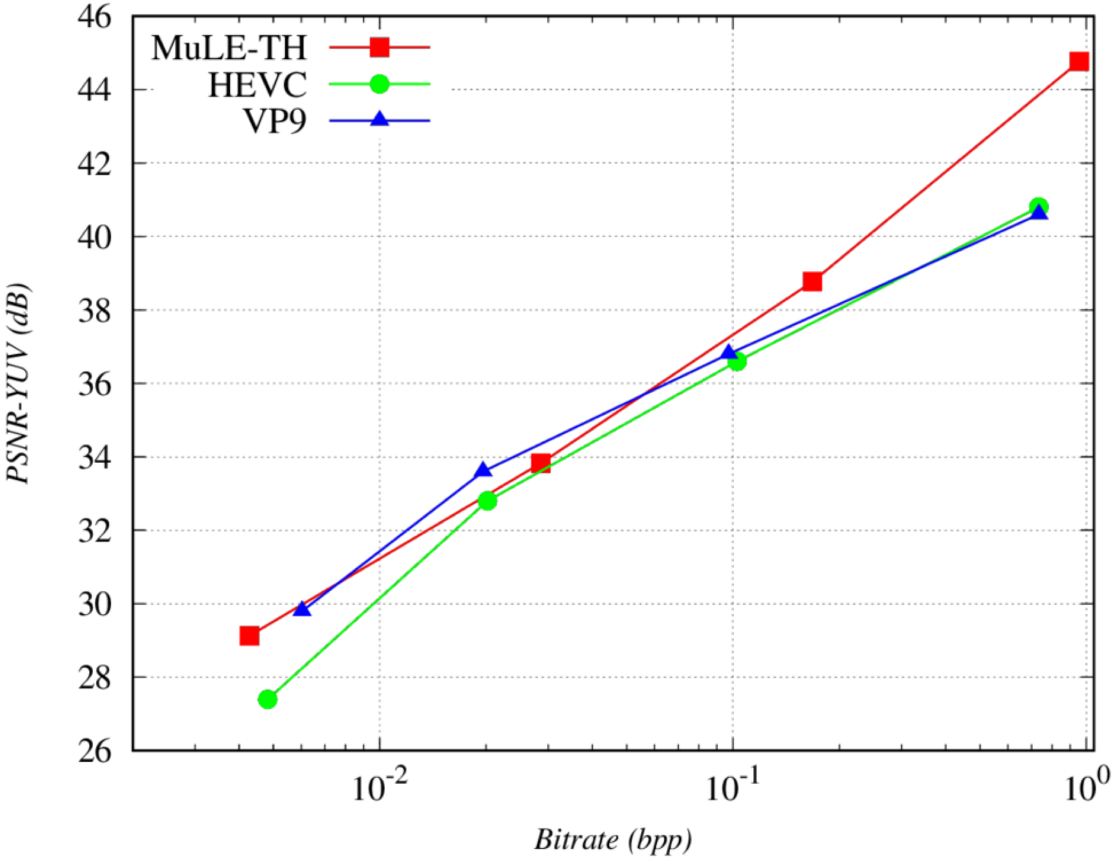
Pillars PSNR YUV



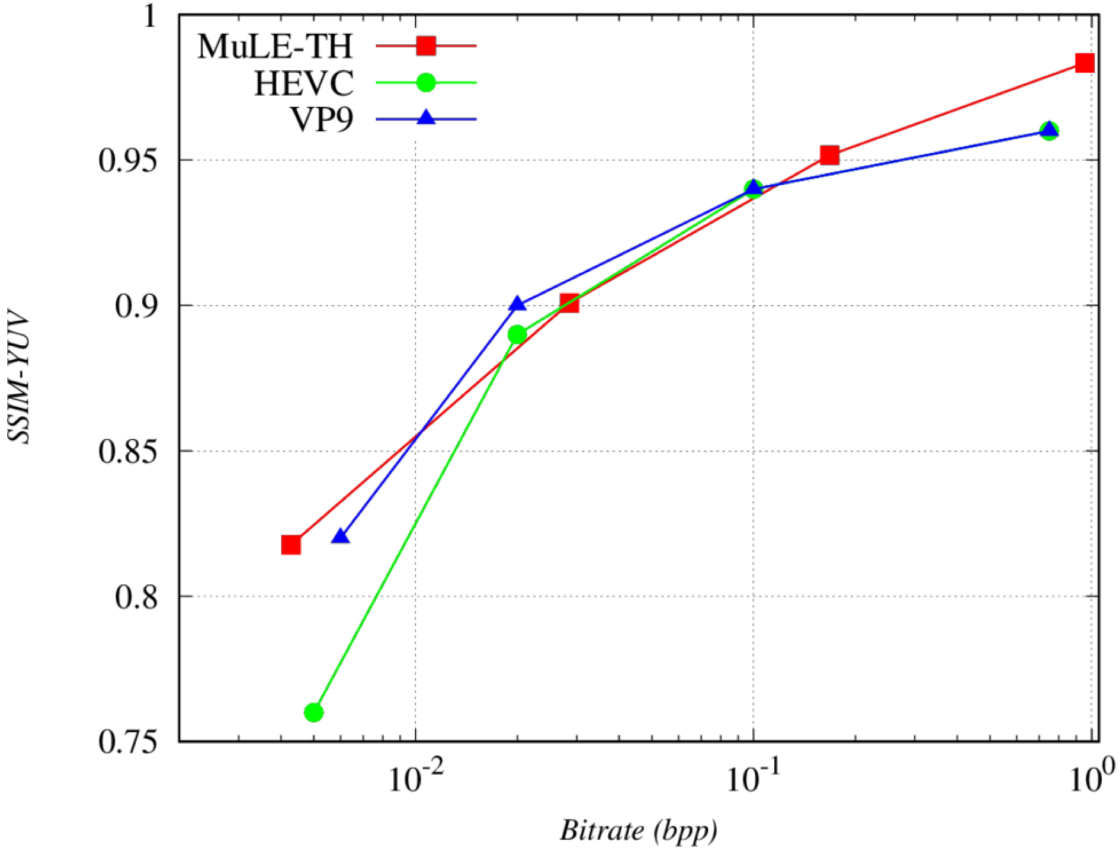
Pillars SSIM YUV



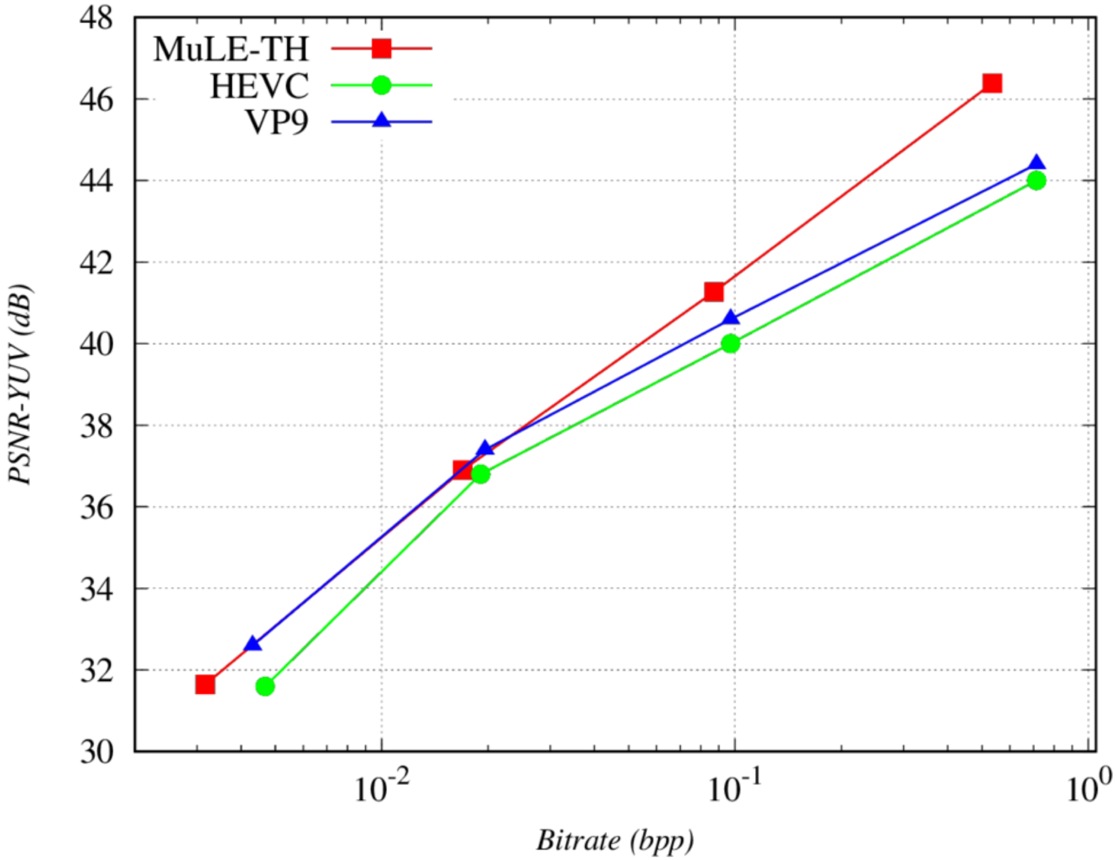
Fountain PSNR YUV



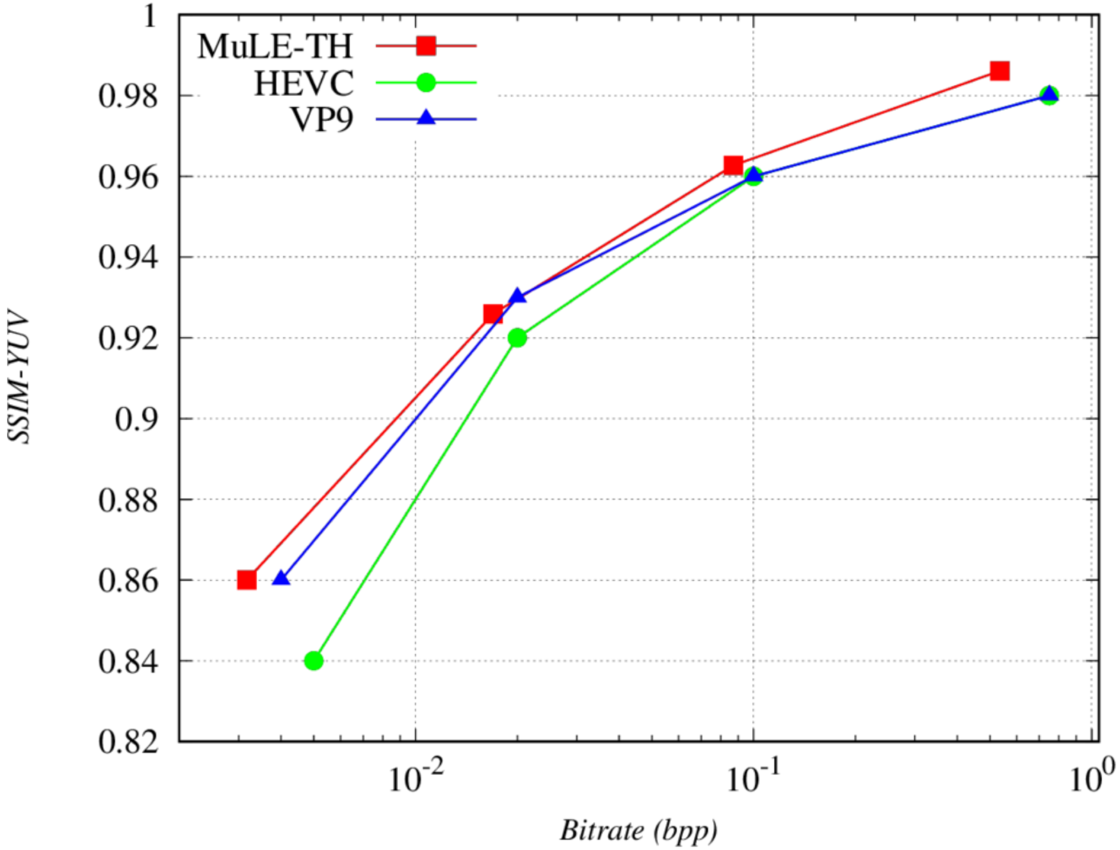
Fountain SSIM YUV



Friends PSNR YUV



Friends SSIM YUV

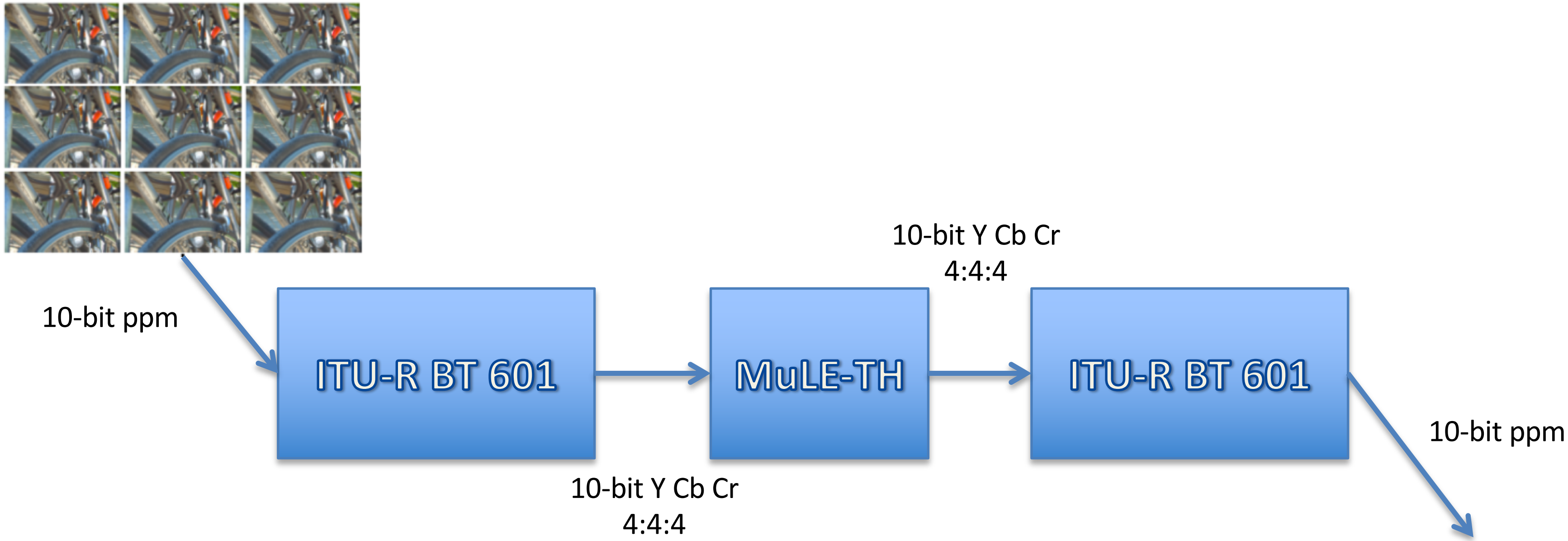


Mule-TH comparison with JPEG Pleno Verification Model

MuLE-TH x JPEG Pleno VM

- JPEG Pleno VM and MuLE-TH codec were applied to the lenslets light fields datasets from JPEG Pleno Common Test Conditions.
- Luminance and chrominance were encoded using transforms of dimensions 13x13x31x25 (13x13 views, each one divided into 14x25 blocks)

Pipeline – MuLE-TH x JPEG Pleno VM

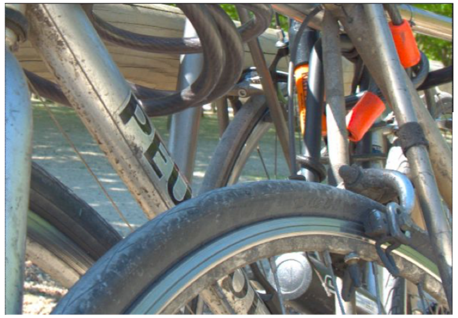


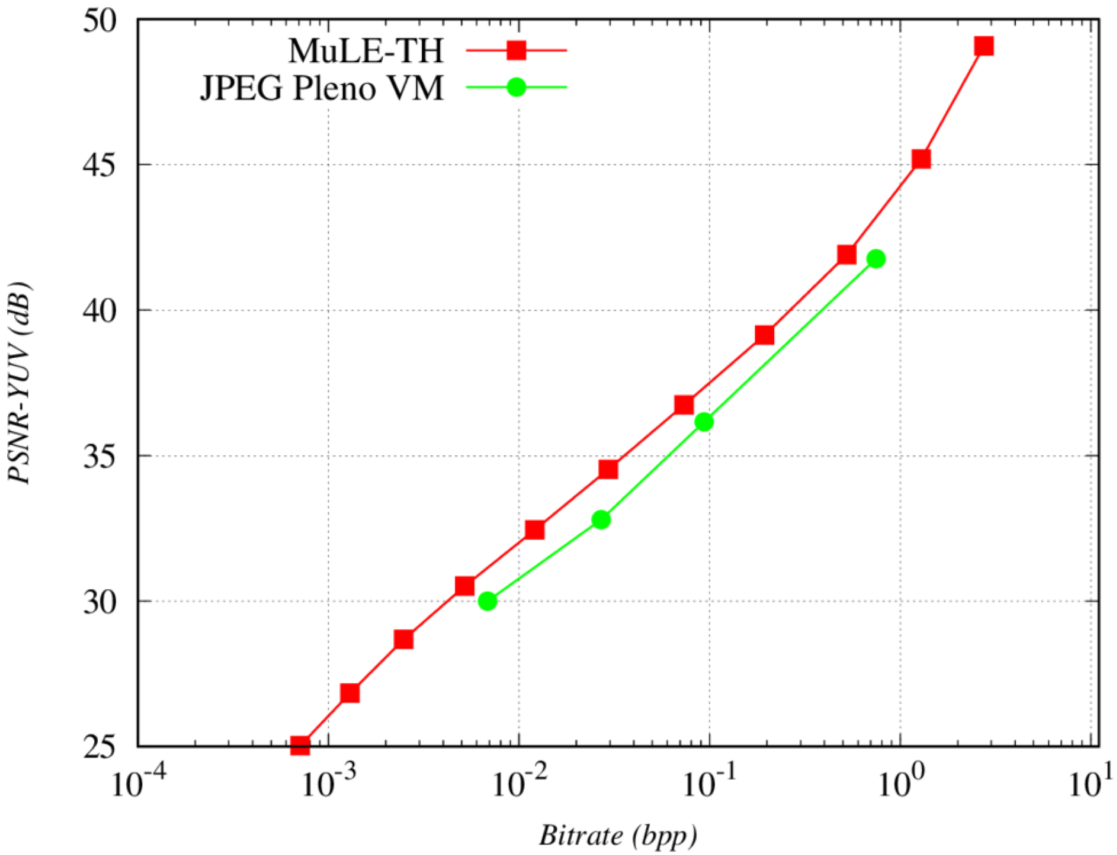
- Assessment:

  PSNRyuv = (6*PSNRy+PSNRu+PSNRv)/8

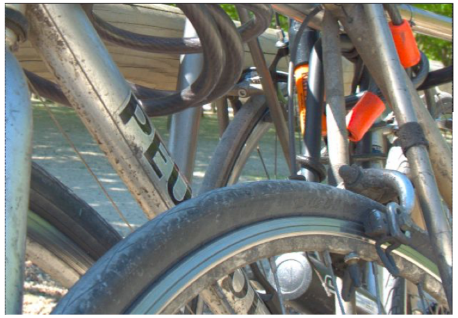
  SSIMyuv = (6*SSIMy +SSIMu +SSIMv)/8

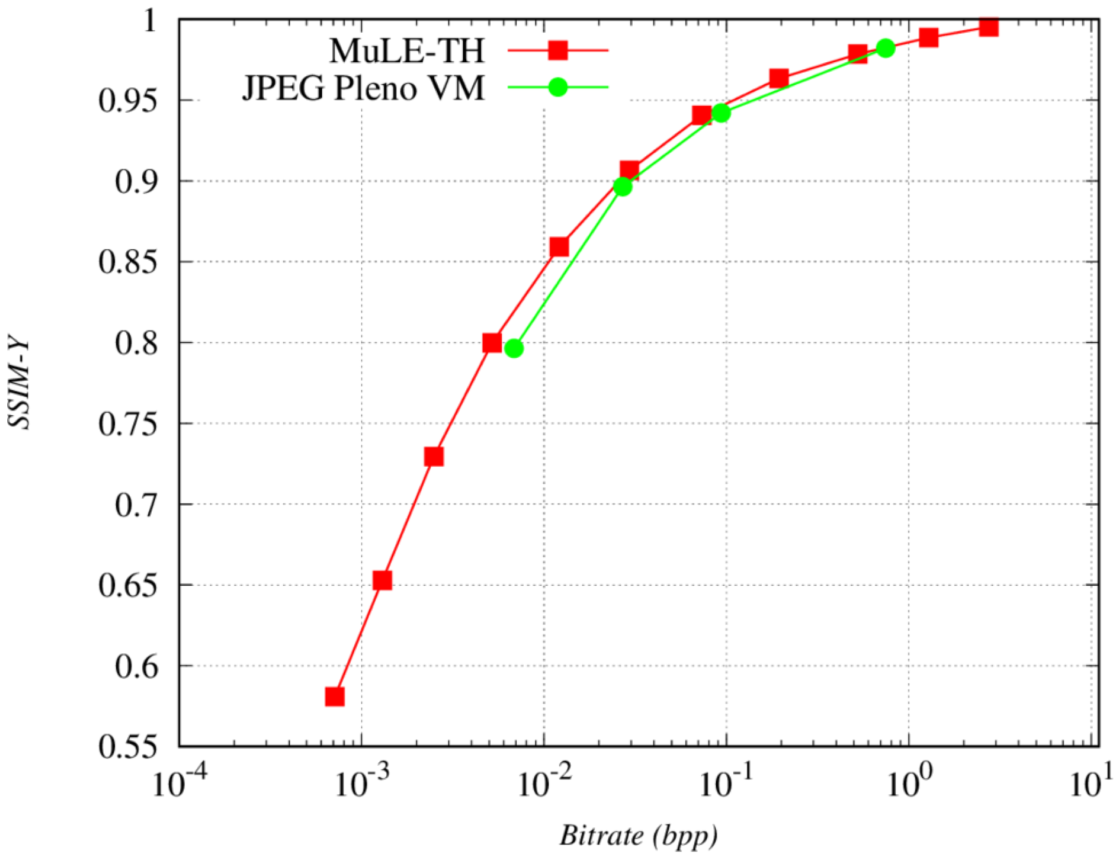
Bikes PSNR YUV





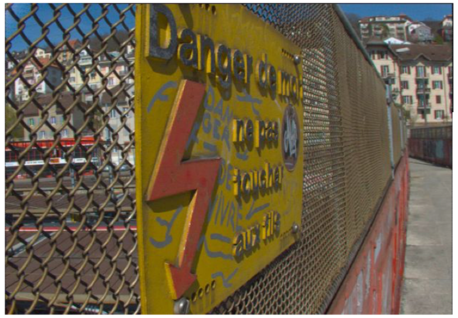
Bikes SSIM Y

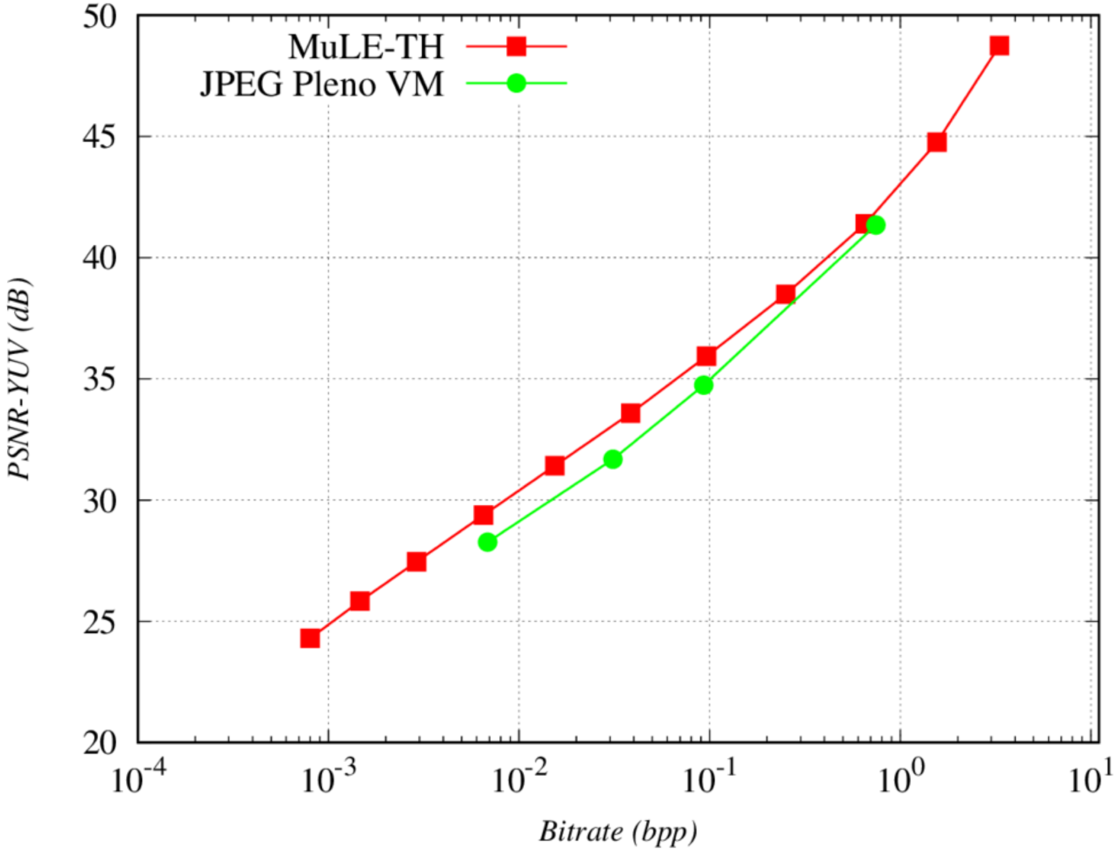




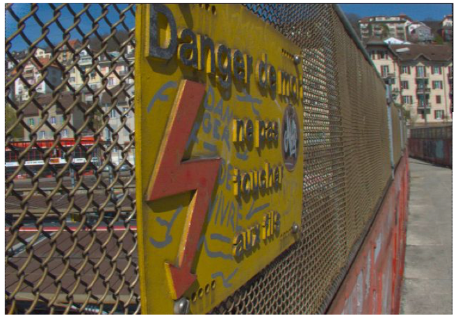
MuLE-TH
JPEG Pleno VM
SSIM-Y
Bitrate (bpp)
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
10^-4
10^-3
10^-2
10^-1
10^0
10^1

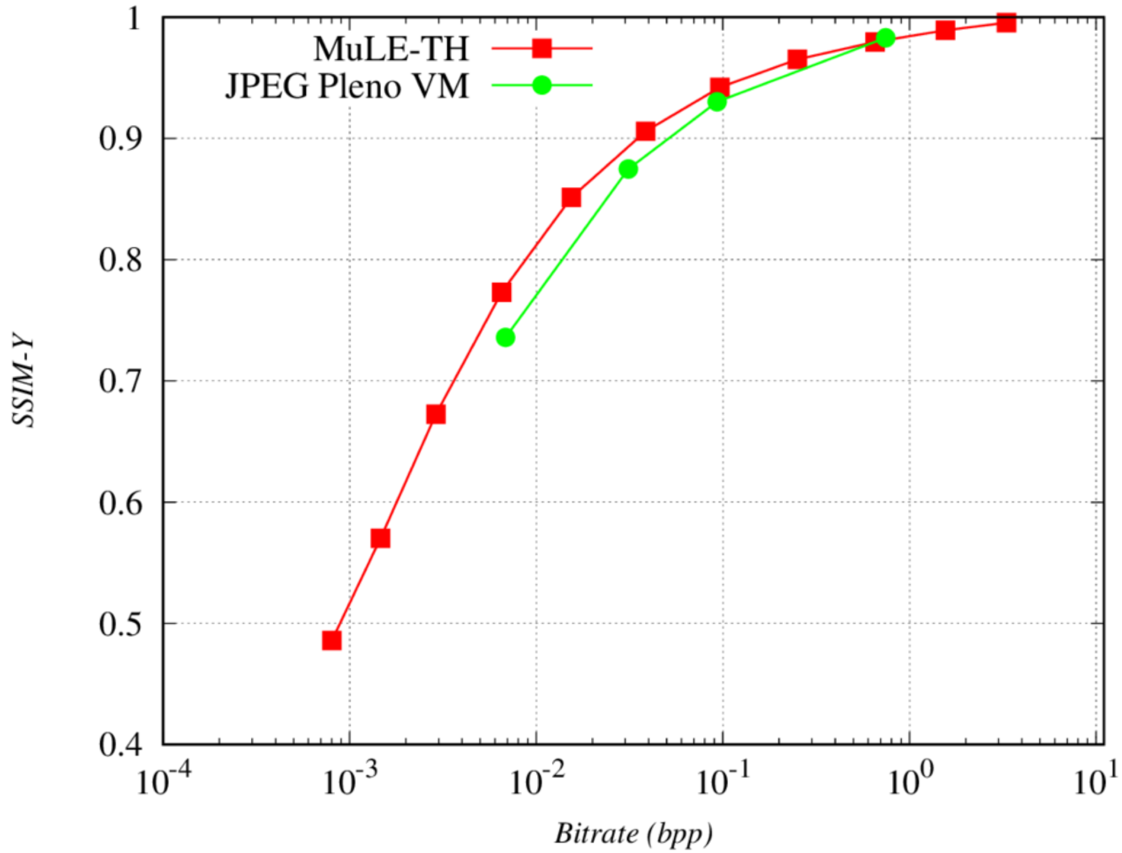
Danger PSNR YUV



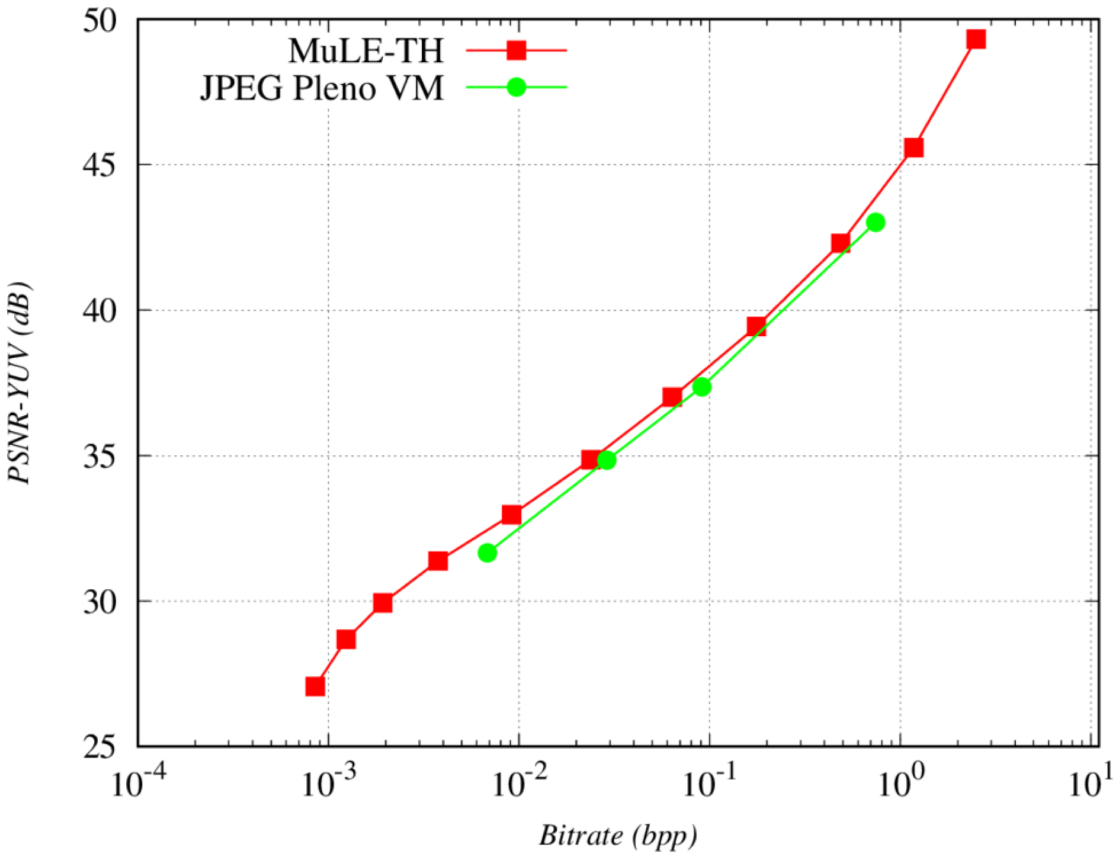


Danger SSIM Y

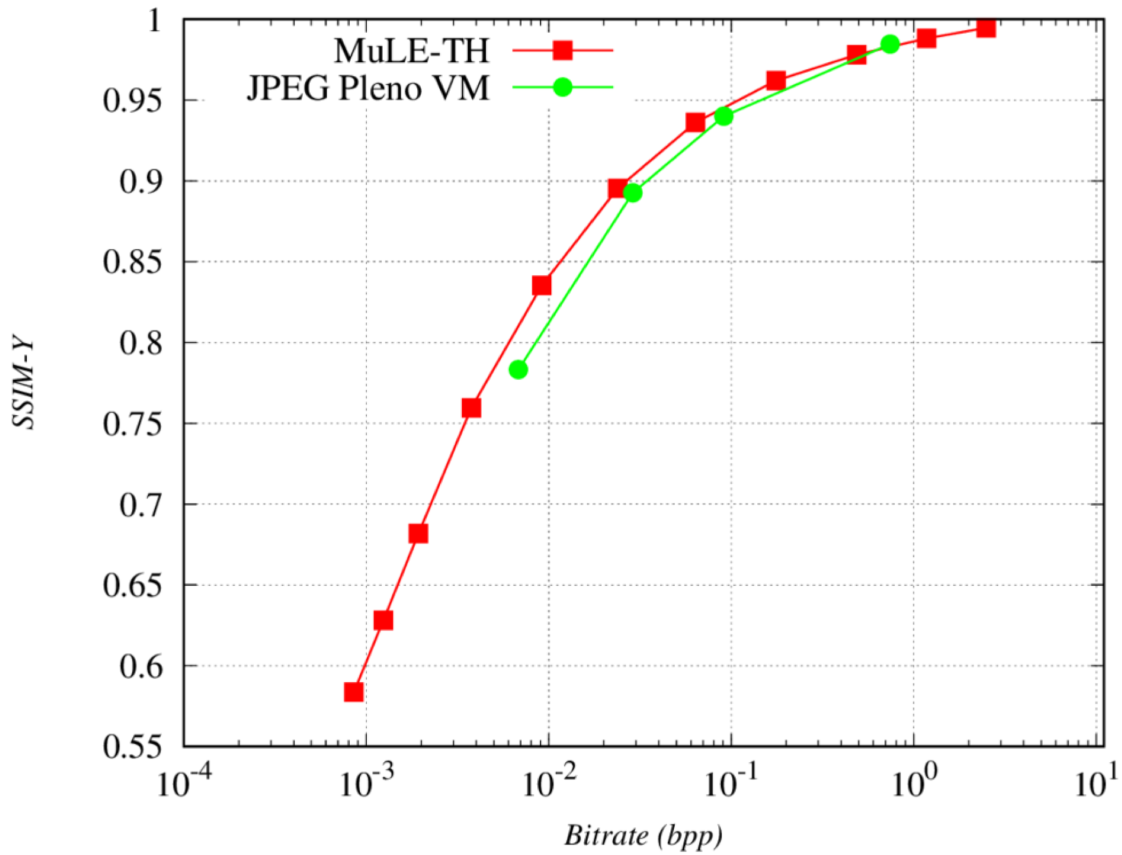




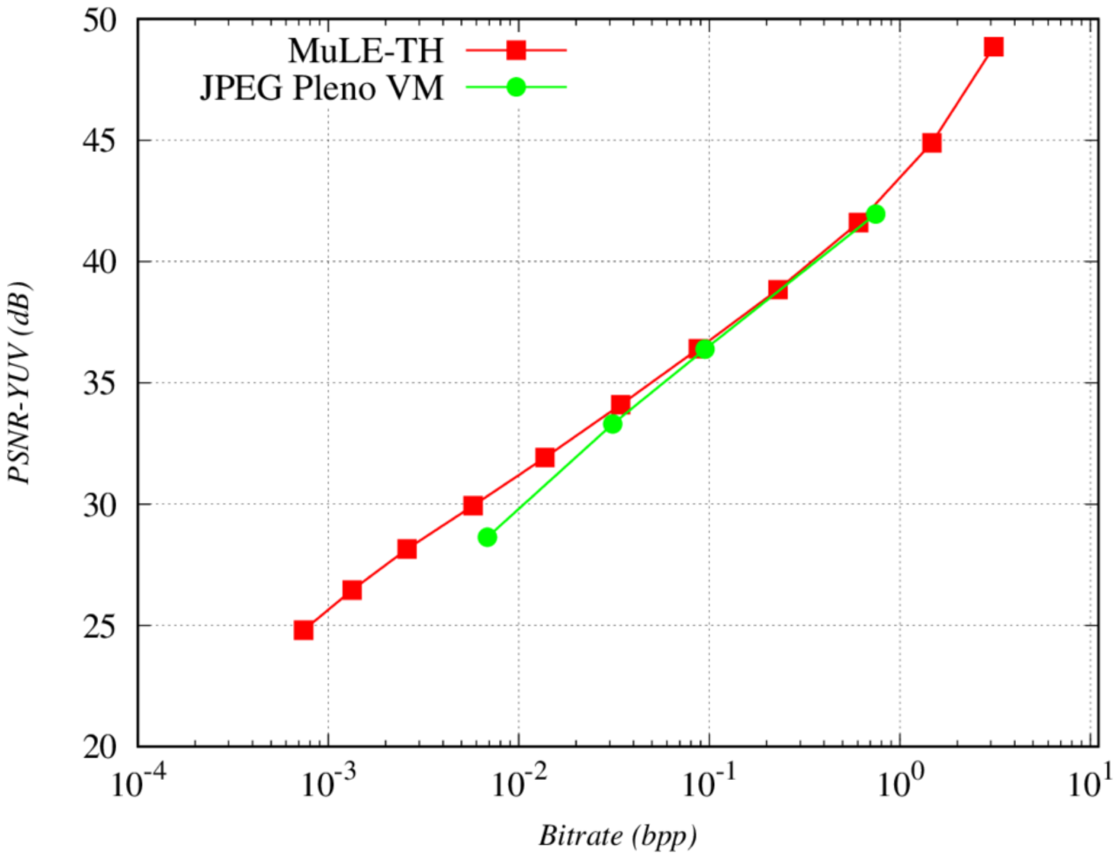
Pillars PSNR YUV



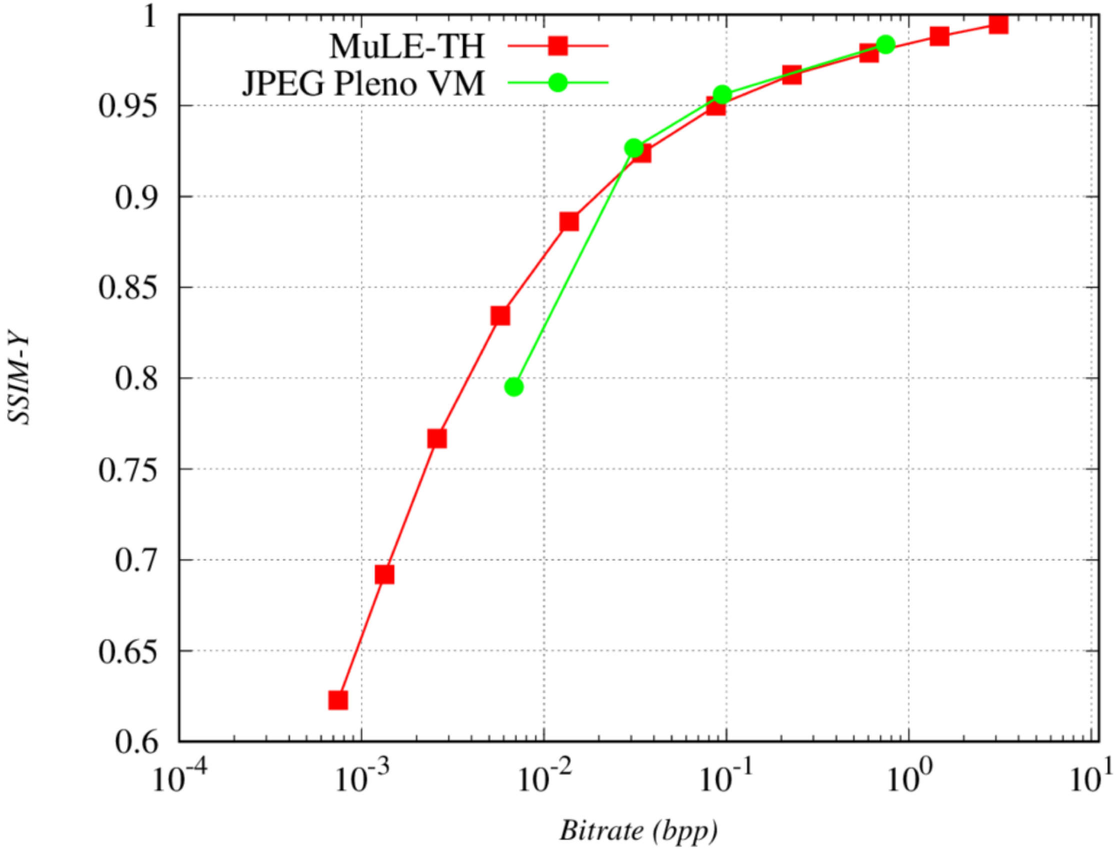
Pillars SSIM Y



Fountain PSNR YUV



Fountain SSIM Y



Analysis

- The results for the lenslets are essentially competitive with benchmarks and other sophisticated methods
- This has been achieved with just a simple, naive encoder that tries to exploit the light fields 4D redundancy as a whole.

4. Conclusions and Future Work

Conclusions

- Proposed coding approach is natively able to exploit intrinsic 4D redundancy of a light field.
- Proposed method does not rely on depth maps or use view synthesis
  - It is therefore immune to the common problems derived from low accuracy of depth maps estimation.
- Results for the lenslet dataset suggest that exploitation of 4D redundancy as a whole has good potential.
- Remark: This work is a contribution to JPEG Pleno

Future Work

- Research and discover other ways to explore 4D redundancy
  - For instance: prediction

Thank you!