The JPEG Pleno Light Field Standard Coding Modes

Carla Pagliari

Instituto Militar de Engenharia

JPEG Pleno Standard

- JPEG Pleno(ptic) is an upcoming standard from the ISO/IEC JTC 1/SC 29/WG 1 (JPEG) Committee.
- It will provide a common framework for coding light field, holography, and point clouds, with each one describing different sampled representations of the plenoptic function.

ISO/IEC

- ISO/IEC: International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).
- ISO/IEC JTC1: Standardization in the field of information technology.
- ISO/IEC JTC 1/SC 29: Coding of audio, picture, multimedia and hypermedia information.
- ISO/IEC JTC 1/SC 29/WG 1: JPEG Coding of digital representations of images.

WG 1 JPEG

- WG 1 Convenor
- WG 1 Webmaster
- JPEG Communication Subgroup
- JPEG Requirements Subgroup
- JPEG Systems & Integration Subgroup
- JPEG Image Coding and Quality Subgroup
- JPEG Plenoptic Coding and Quality Subgroup
- Ad hoc Groups (temporary)

JPEG Ad-hoc Groups (AHGs)

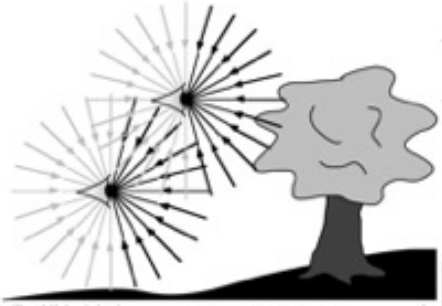
- Ad Hoc Group on JPEG XS
- Ad Hoc Group on JPEG Website, Branding, Sponsorships
- **Ad Hoc Group on JPEG Pleno - Light Field**
- Ad Hoc Group on JPEG Pleno - Point Cloud
- Ad Hoc Group on JPEG Pleno – Holography
- Ad Hoc Group on JPEG 2000
- Ad Hoc Group on JPEG XL
- Ad Hoc Group on software and documentation best practices
- Ad Hoc Group on Fake Media
- Ad Hoc Group on Learning-based Image Coding
- Ad Hoc Group on JPEG Systems
- Ad Hoc Group on JPEG Reference Software
- Ad Hoc Group on Best Practices for JPEG Meetings
- Ad Hoc Group on Digital Media Storage using DNA

https://jpeg.org/participate.html

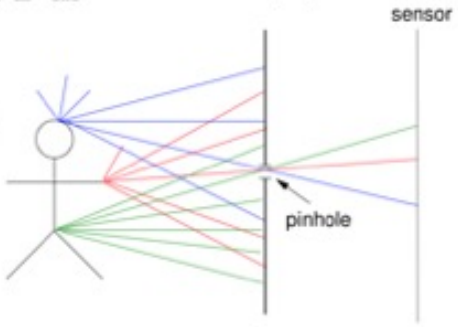
Light Field

The Plenoptic Function

- The classical work on photometry [1] considered the light field as a vector field in 3D space [1]. The light field is also known as the plenoptic function, parameterized by Adelson and Bergen [2].
- A region of space is filled with a dense array of light rays of varying intensities [2].
- If a pinhole camera is positioned at a certain point, it will select the set of rays at that point showing that the rays form an image [2].



Extracted from [2].



1. Lippman G. La photographie integrale. Comptes-Rendus. Academie des Sciences; 1908. p. 446–551.

Light Field

- Any illuminated object in the 3D space fills the space around it with rays of light, that have different wavelengths and vary in time.
- These fields of light (light fields) can be sampled by recording 2D images of the object from many points of view.
- These points of view could be placed at different locations, receiving the light rays at different angles (directions), wavelenghts and time [2].

Light Field

- Adelson and Bergen [2] specified the structure of light by the plenoptic function, that is a 7D structure that describes all visual information in the 3D world.
- For an idealized eye (observer) at every possible location (Vx,Vy,Vz) it associates for each direction (θ,φ) a light ray in a particular wavelength (λ) at a particular point in time (t).
- The radiance L is represented by a 7D function of a light ray:

$$L = L(\mathrm{Vx},\mathrm{Vy},\mathrm{Vz},\ \theta,\ \varphi,\ \lambda,\ t)$$

Light Field

- At a given moment of time, any human observer can acquire two samples from the Vx axis using the binocular vision. But both Vy and Vz axes cannot be sampled by an static viewer at a given instant of time. As humans can only sample, at a given moment, along five of the seven dimensions of the plenoptic function it can be reduced to a 5D structure [2] and parameterized as

$$L(\mathrm{Vx},\mathrm{Vy},\mathrm{Vz},\theta,\varphi,\lambda,t) \longrightarrow L(\mathrm{Vx},\theta,\varphi,\lambda,t)$$

- The works in [3, 4] assume that geometrical optics rays travel along straight lines and remain constant in a bounded empty region (free of occluders) of space, thus reducing the 5D light field structure to a 4D one, as the wavelength (λ) is not considered. So, from a 7D function the radiance can be reduced to a 4D one.
- To represent the direction of the rays entering the eye (or the camera), one can parameterize the plenoptic function in terms of spatial coordinates of an imaginary picture plane erected at a unit distance from the pupil [2,3,4], instead of using the angles (θ and φ).

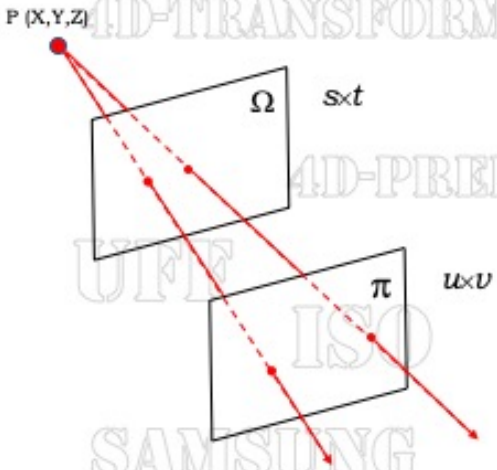
Light Field

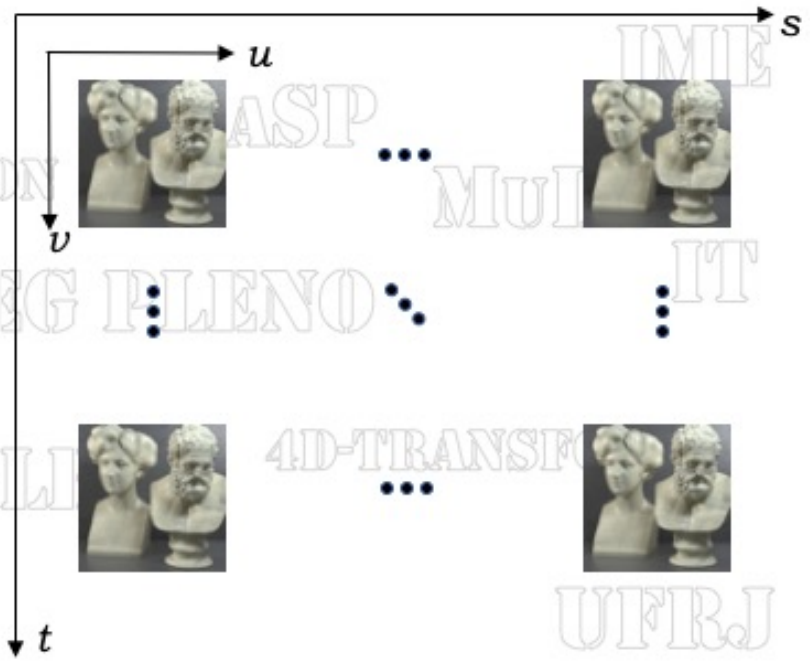
- By representing the spatial axis Vx as s, and the angles as a plane π, the 4D light field structure then becomes:

$$L = L(u, v, s, t)$$

- The parameterization $L(u, v, s, t)$ is a 4D simplification of the plenoptic function that considers the intensity of each light ray constant along its path, parameterizing each light ray by the two 2D coordinates of its intersection with two planes Ω and π:
  - with the (s,t) coordinates indexing the views (angular views) and the (u,v) coordinates indexing the samples within the views

Light Field





4D-TRANSFORM
4D-PREDICTION
JPEG PLENO
SAMSUNG
IME
UFRJ

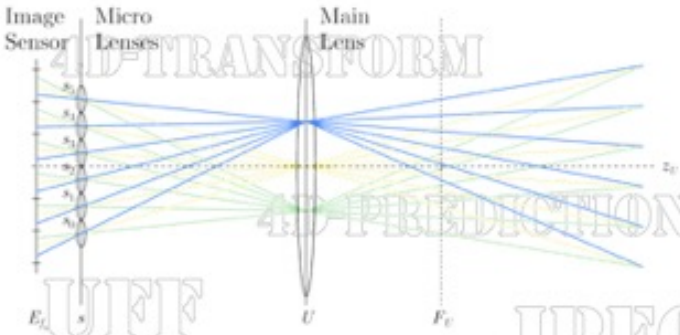
Light Field

Movement along the *s* axis

Movement along the *t* axis

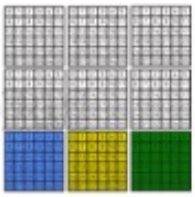
Greek dataset

Light Field Acquisition



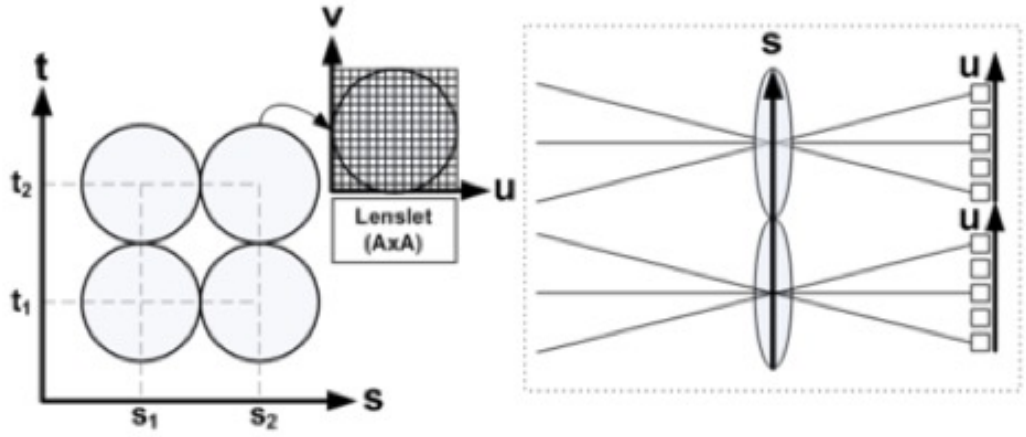
Micro images

Sub-apertures



http://www.plenoptic.info/pages/sub-aperture.html

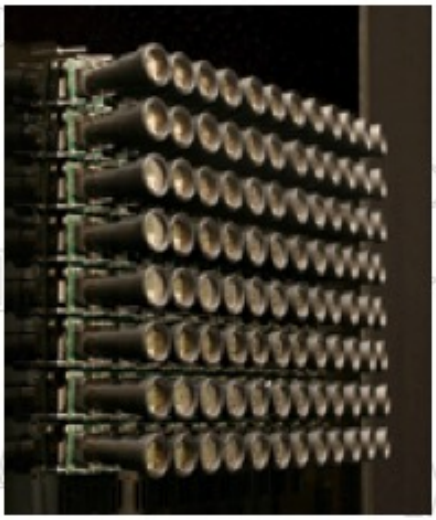
Light Field Acquisition



Each circle corresponds to a single micro-lens and the number of micro-lenses defines the spatial resolution of the light field.
Light field can be parameterized by the lenslet positions (s,t) and the pixel positions (u,v) behind a lenslet.

Gul, M. Shahzeb Khan and Gunturk, Bahadir K. "Spatial and Angular Resolution Enhancement of Light Fields Using Convolutional Neural Networks", IEEE Transactions on Image Processing, May 2018

Light Field Acquisition



http://graphics.stanford.edu/projects/array/

Light Field

Bikes dataset

Movement along the *s* axis

Set2 dataset

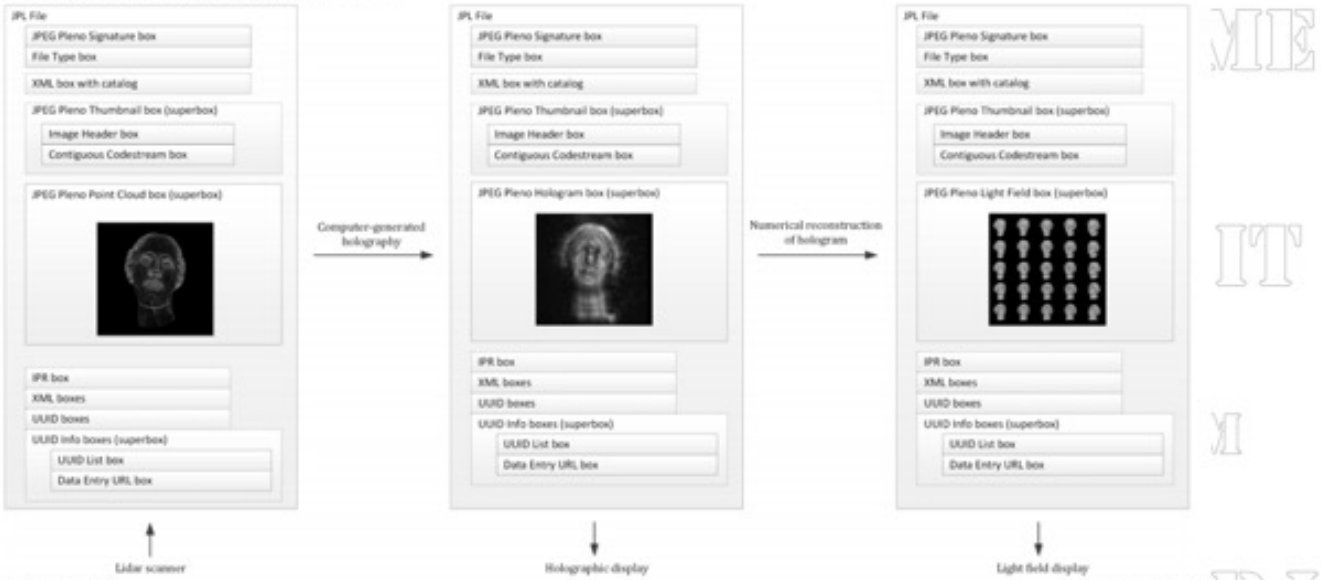
Movement along the *s* axis

JPEG Pleno Standard

JPEG Standards

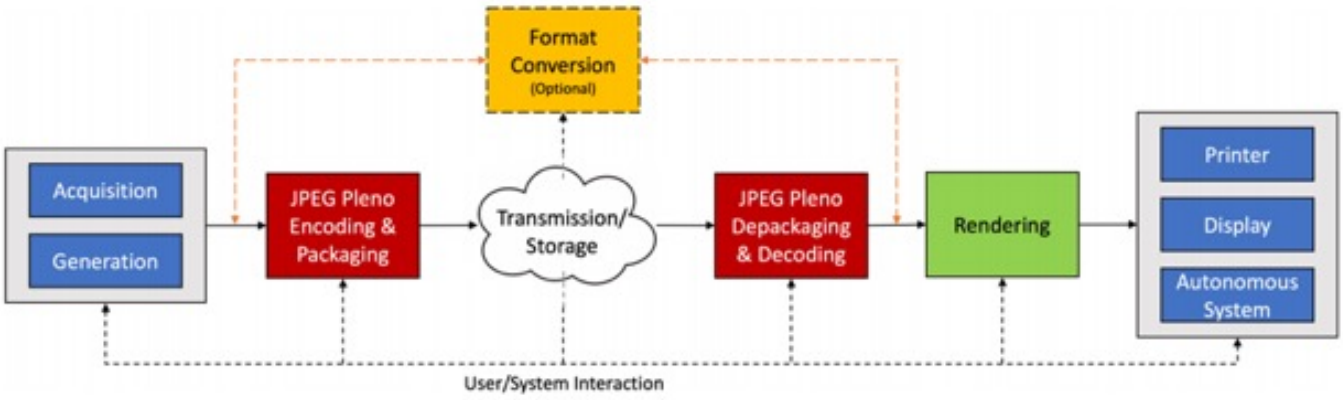
- Call for Evidence (CfE)
- Call for Proposals (CfP)
- JPEG AHG meetings
  - Exploration Studies (ES)
  - Core Experiments (CE)
  - Writing the standard (https://www.iso.org/stage-codes.html)
    - Working Draft (WD)
    - Committee Draft (CD)
    - Draft International Standard (DIS)
    - Final Draft International Standard (FDIS)
      - Proof of a new International Standard (PRF)
    - International Standard (IS)

JPEG Pleno Standard



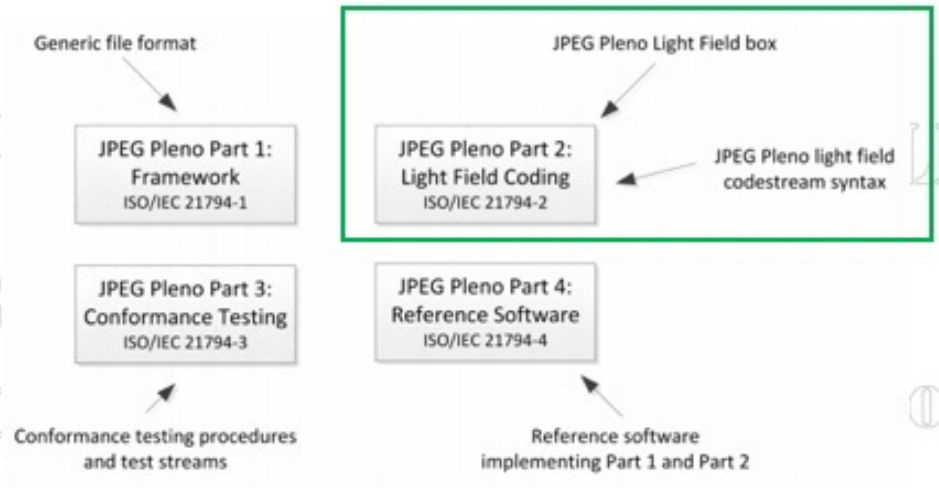
Extracted from [5].

JPEG Pleno Standard



Extracted from [5].

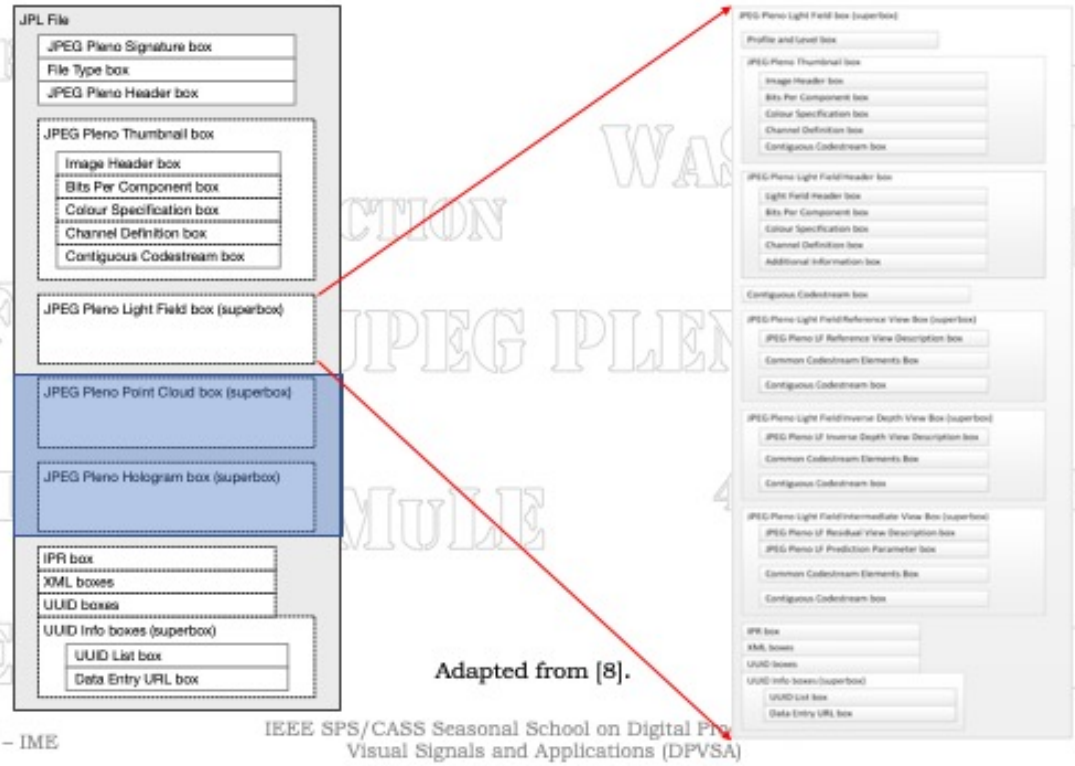
JPEG Pleno Standard



Adapted from [5,6].

The JPEG Pleno Light Field Standard Parts 1 and 2 are in their final stage [5, 6, 7].

JPEG Pleno generic file format



Adapted from [8].

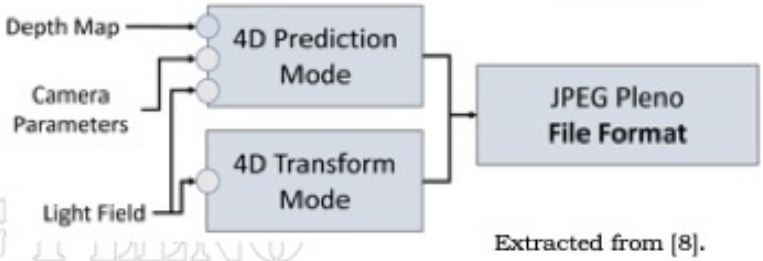
JPEG Pleno generic file format

Adapted from [8].

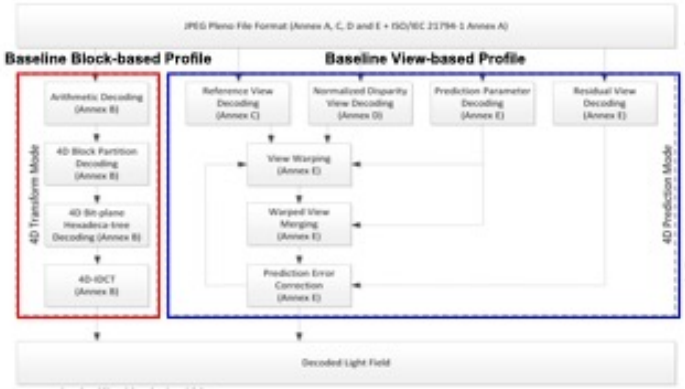
JPEG Pleno LF Standard Coding Modes

JPEG Pleno LF Standard Coding Modes

- Two independent coding modes
  - 4D-Prediction
  - 4D-Transform



Extracted from [8].



Adapted from [5].

JPEG Pleno generic file format

4D-Prediction

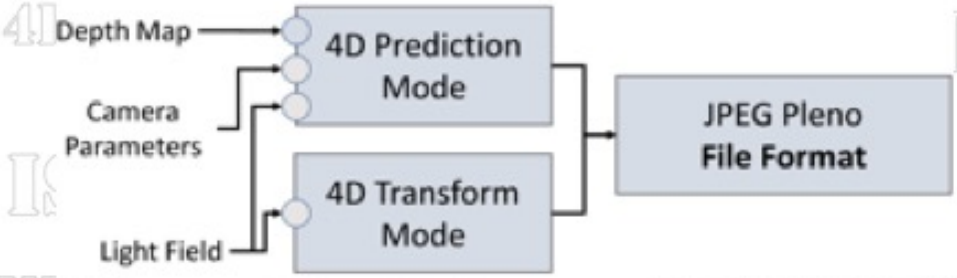
4D-Transform and 4D-Prediction

Adapted from [8].

JPEG Pleno LF Standard

- ISO/IEC JTC1/SC29/WG1, "Information technology: JPEG Pleno Part 2 – Light field coding ISO/IEC FDIS 21794-2," Apr. 2020, WG1N87033, 87th JPEG Meeting, Erlangen (online), Germany
  - Prof. Eduardo Antônio Barros da Silva (UFRJ) is the co-editor of the JPEG Pleno Part 2, co-chair of the JPEG Pleno Light Field Ad hoc group (up to the 89th meeting) and co-chair of the JPEG Requirements Subgroup
  - Prof. Carla Pagliari (IME) is the co-chair of the JPEG Pleno Light Field Ad hoc group (starting after the 89th meeting)
- ISO/IEC JTC1/SC29/WG1, "Information technology: JPEG Pleno Part 3 – Conformance testing ISO/IEC CD 21794-3", Oct. 2020, WG1N89103, 89th, JPEG Meeting, Rennes (online), France
- ISO/IEC JTC1/SC29/WG1, "Information technology: JPEG Pleno Part 4 – Reference software ISO/IEC CD 21794-4", Oct. 2020, WG1N89104, 89th, JPEG Meeting, Rennes (online), France
  - Dr. Pedro Garcia and Dr. Ismael Seidel (Samsung R&D Brazil) are the co-editors of the JPEG Pleno Parts 3 and 4

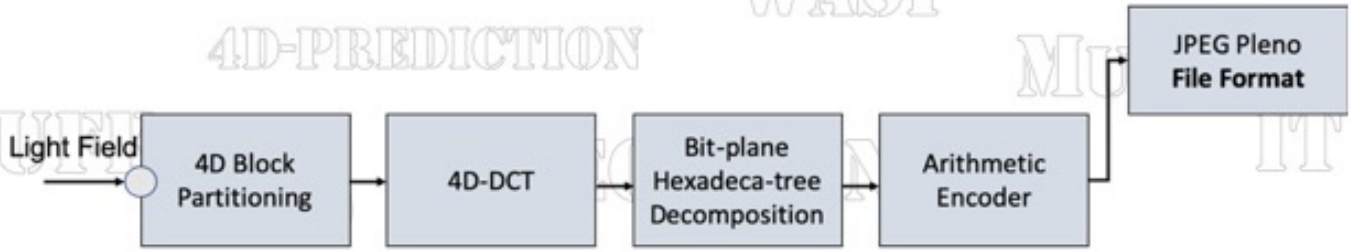
JPEG Pleno LF Standard Coding Modes



Extracted from [8].

JPEG Pleno LF Standard Coding Modes

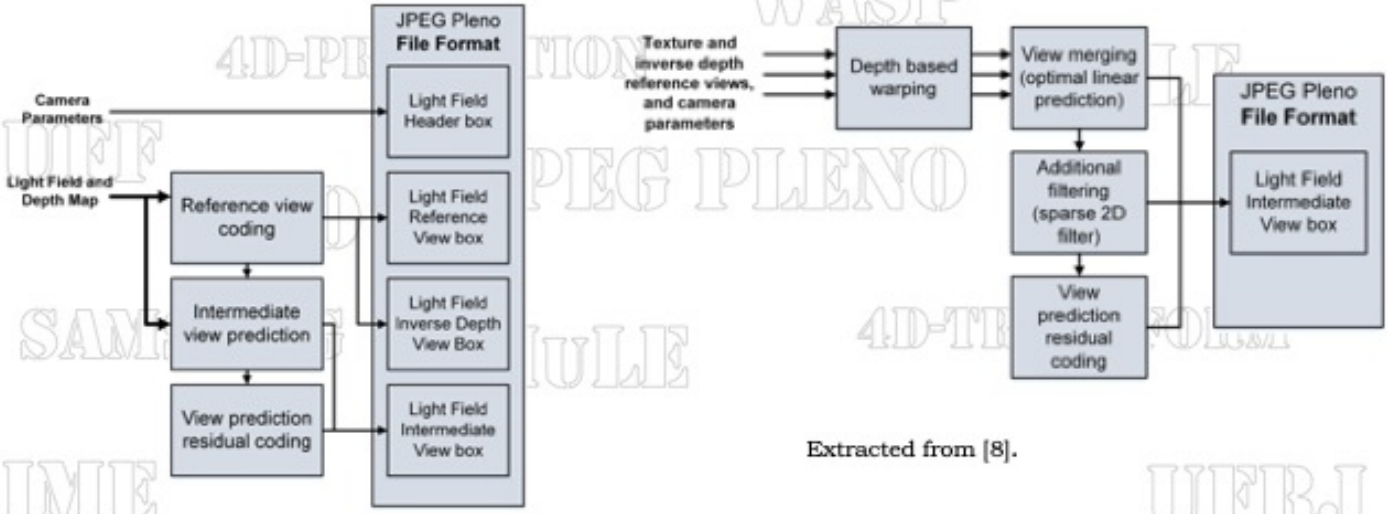
- 4D-Transform block diagram



Extracted from [8].

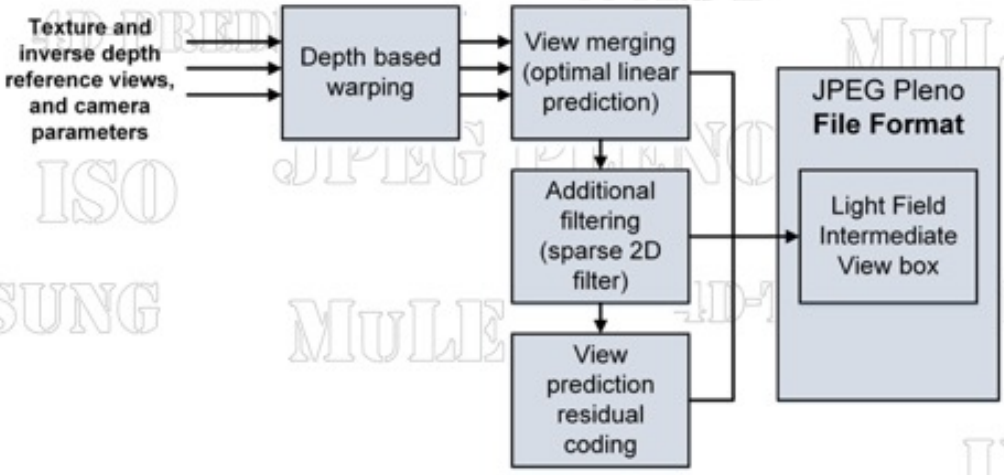
JPEG Pleno LF Standard Coding Modes

- 4D-Prediction block diagram



Extracted from [8].

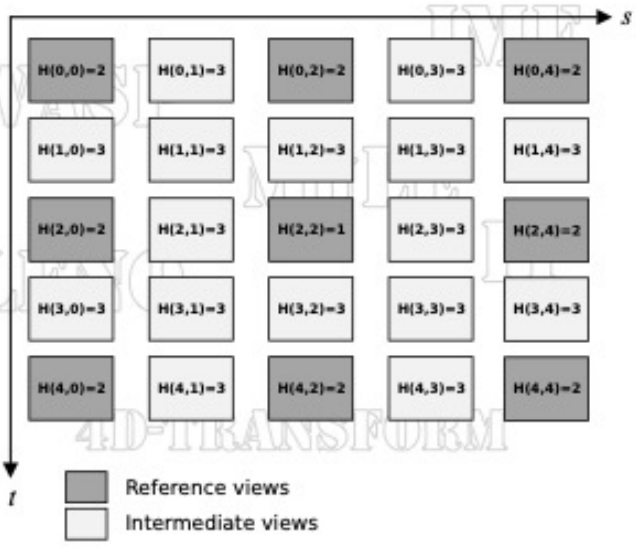
4D-Prediction mode



Extracted from [8].

4D-Prediction

- In the 4D Prediction Mode (4DPM) a subset of views is selected as reference views.
- The rest of the views are referred to as intermediate views.
- The texture and depth of the reference views are encoded using the JPEG2000 standard.
- The 4DPM is the Warping and Sparse Prediction (WaSP) codec, initially presented in [9]. A novel version is proposed in [10] using HEVC instead of JPEG 2000.



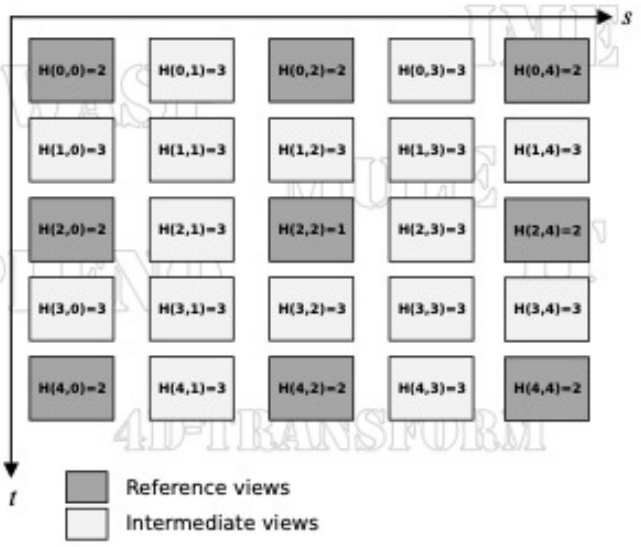
Extracted from [5].

Hierarchical Encoding

- The WaSP coding is organized on hierarchical levels, where all views belonging to a particular level are encoded using views encoded in the previous hierarchical levels (1 to 6).
- The encoder/user can decide to encode the whole LF using only the reference view of Level 1 that is encoded by JPEG 2000, whereas all the remaining views are synthesized using the reference view and the depth maps.
- If one uses two levels (1 and 2), the reference views of Levels 1 and 2 are encoded by JPEG 2000, whereas all the remaining views are synthesized using the references views and the depth maps.
- If one uses three levels...

Hierarchical Encoding

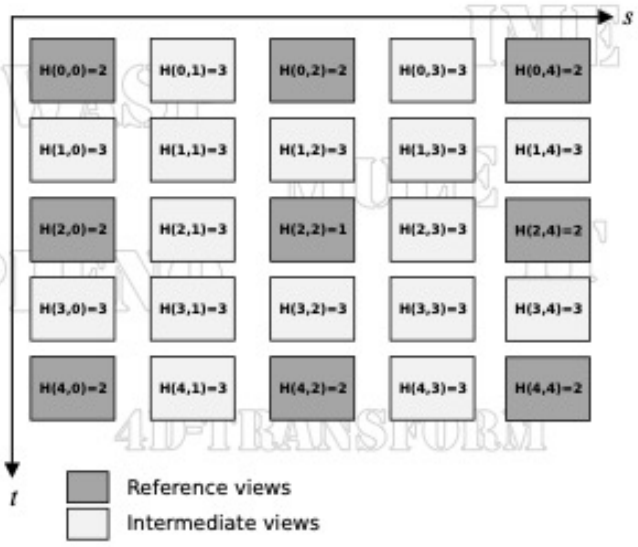
- The hierarchical partitioning of views used is similar to the frame referencing structures used in video codecs such as HEVC.
- Reference view, depth and view prediction information is encoded by default with JPEG 2000 (other codecs can be used [10]).



Extracted from [5].

Hierarchical Encoding

- The intermediate (synthesized) views are predicted using the depth maps and the reference views of the current hierarchical level and the previously encoded views of the lower hierarchical levels.



Extracted from [5].

000_000	001_000	002_000	003_000	004_000	005_000	006_000	007_000	008_000
000_001	001_001	002_001	003_001	004_001	005_001	006_001	007_001	008_001
000_002	001_002	002_002	003_002	004_002	005_002	006_002	007_002	008_002
000_003	001_003	002_003	003_003	004_003	005_003	006_003	007_003	008_003
000_004	001_004	002_004	003_004	004_004	005_004	006_004	007_004	008_004
000_005	001_005	002_005	003_005	004_005	005_005	006_005	007_005	008_005
000_006	001_006	002_006	003_006	004_006	005_006	006_006	007_006	008_006
000_007	001_007	002_007	003_007	004_007	005_007	006_007	007_007	008_007
000_008	001_008	002_008	003_008	004_008	005_008	006_008	007_008	008_008

4D-TRANSFORM IME WASP 4D-PREDICTION MULE UFF ISO JPEG PLENO IT SAMSUNG MULE 4D-TRANSFORM IME UFRJ

Hierarchical Level 1
R_1 (bpp)

4D-TRANSFORM IME WASP 4D-PRODUCTION MULE UFF ISO JPEG PLENO SAMSUNG MULE 4D-TRANSFORM IME UFRJ

000_000	001_000	002_000	003_000	004_000	005_000	006_000	007_000	008_000
000_001	001_001	002_001	003_001	004_001	005_001	006_001	007_001	008_001
000_002	001_002	002_002	003_002	004_002	005_002	006_002	007_002	008_002
000_003	001_003	002_003	003_003	004_003	005_003	006_003	007_003	008_003
000_004	001_004	002_004	003_004	**004_004**	005_004	006_004	007_004	008_004
000_005	001_005	002_005	003_005	004_005	005_005	006_005	007_005	008_005
000_006	001_006	002_006	003_006	004_006	005_006	006_006	007_006	008_006
000_007	001_007	002_007	003_007	004_007	005_007	006_007	007_007	008_007
000_008	001_008	002_008	003_008	004_008	005_008	006_008	007_008	008_008

Hierarchical Level 2
R_2 (bpp)

000_000	001_000	002_000	003_000	004_000	005_000	006_000	007_000	008_000
000_001	001_001	002_001	003_001	004_001	005_001	006_001	007_001	008_001
000_002	001_002	002_002	003_002	004_002	005_002	006_002	007_002	008_002
000_003	001_003	002_003	003_003	004_003	005_003	006_003	007_003	008_003
000_004	001_004	002_004	003_004	004_004	005_004	006_004	007_004	008_004
000_005	001_005	002_005	003_005	004_005	005_005	006_005	007_005	008_005
000_006	001_006	002_006	003_006	004_006	005_006	006_006	007_006	008_006
000_007	001_007	002_007	003_007	004_007	005_007	006_007	007_007	008_007
000_008	001_008	002_008	003_008	004_008	005_008	006_008	007_008	008_008

Hierarchical Level 3
R_3 (bpp)

000_000	001_000	002_000	003_000	004_000	005_000	006_000	007_000	008_000
000_001	001_001	002_001	003_001	004_001	005_001	006_001	007_001	008_001
000_002	001_002	002_002	003_002	004_002	005_002	006_002	007_002	008_002
000_003	001_003	002_003	003_003	004_003	005_003	006_003	007_003	008_003
000_004	001_004	002_004	003_004	**004_004**	005_004	006_004	007_004	008_004
000_005	001_005	002_005	003_005	004_005	005_005	006_005	007_005	008_005
000_006	001_006	002_006	003_006	004_006	005_006	006_006	007_006	008_006
000_007	001_007	002_007	003_007	004_007	005_007	006_007	007_007	008_007
000_008	001_008	002_008	003_008	004_008	005_008	006_008	007_008	008_008

Hierarchical Level 4
R_4 (bpp)

000_000	001_000	002_000	003_000	004_000	005_000	006_000	007_000	008_000
000_001	001_001	002_001	003_001	004_001	005_001	006_001	007_001	008_001
000_002	001_002	002_002	003_002	004_002	005_002	006_002	007_002	008_002
000_003	001_003	002_003	003_003	004_003	005_003	006_003	007_003	008_003
000_004	001_004	002_004	003_004	004_004	005_004	006_004	007_004	008_004
000_005	001_005	002_005	003_005	004_005	005_005	006_005	007_005	008_005
000_006	001_006	002_006	003_006	004_006	005_006	006_006	007_006	008_006
000_007	001_007	002_007	003_007	004_007	005_007	006_007	007_007	008_007
000_008	001_008	002_008	003_008	004_008	005_008	006_008	007_008	008_008

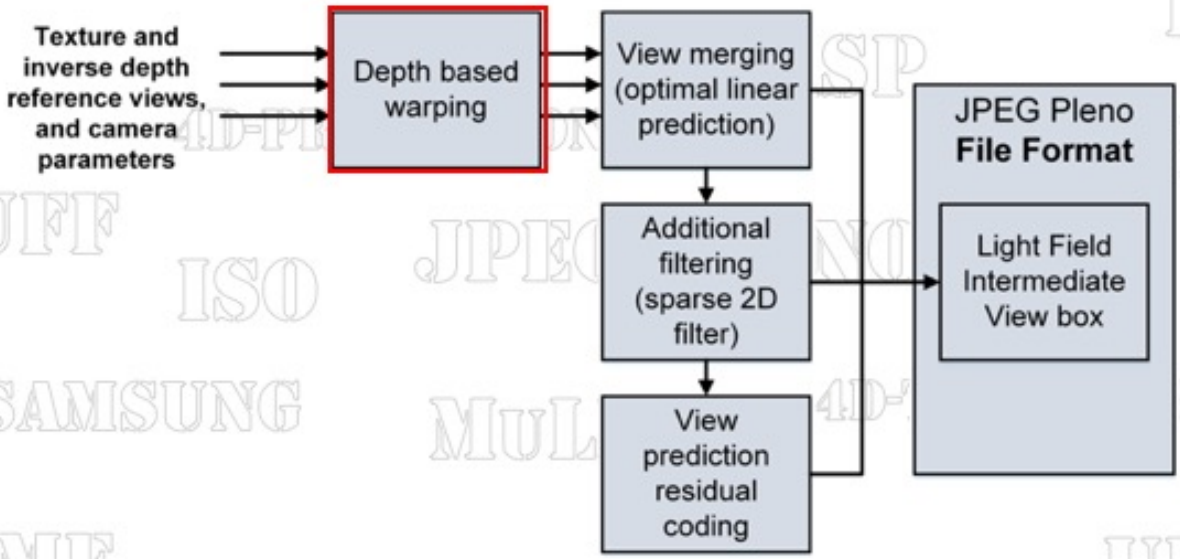
Hierarchical Level 5
R_5 (bpp)

000_000	001_000	002_000	003_000	004_000	005_000	006_000	007_000	008_000
000_001	001_001	002_001	003_001	004_001	005_001	006_001	007_001	008_001
000_002	001_002	002_002	003_002	004_002	005_002	006_002	007_002	008_002
000_003	001_003	002_003	003_003	004_003	005_003	006_003	007_003	008_003
000_004	001_004	002_004	003_004	004_004	005_004	006_004	007_004	008_004
000_005	001_005	002_005	003_005	004_005	005_005	006_005	007_005	008_005
000_006	001_006	002_006	003_006	004_006	005_006	006_006	007_006	008_006
000_007	001_007	002_007	003_007	004_007	005_007	006_007	007_007	008_007
000_008	001_008	002_008	003_008	004_008	005_008	006_008	007_008	008_008

Hierarchical Level 6

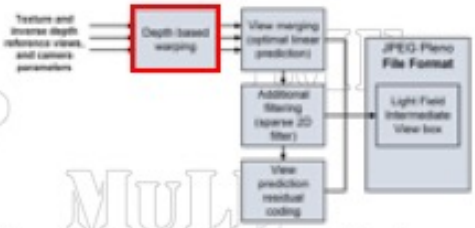
R_6 (bpp)

Depth-based warping



Extracted from [8].

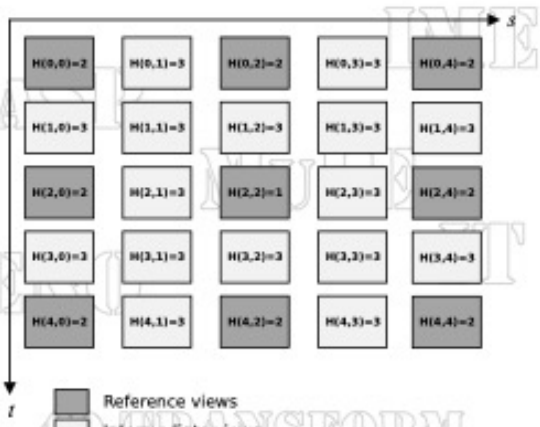
Depth-based warping



- Depth information is used to warp the pixels of the reference views to their corresponding locations in the camera position of the target intermediate view [9,10].
- Each warped reference view produces a distinct pattern of disoccluded pixels, used to assist the segmentation of the predicted intermediate view.
- The segmentation used in 4DPM is generated by the split of the set of pixels into various occlusion classes. Each occlusion class defines a specific subset of warped reference views. These subsets are used to obtain the optimal linear predictor the least-squares sense involving only the relevant warped reference views at each pixel location of the predicted view.
- The occlusion classes are specially designed to handle the depth-based prediction of wide baseline light fields where disocclusions in warped reference views need to be considered during view merging [9,10].

Depth-based warping

Warping the reference view X(0,2) to the camera position of the target view X(4,2)



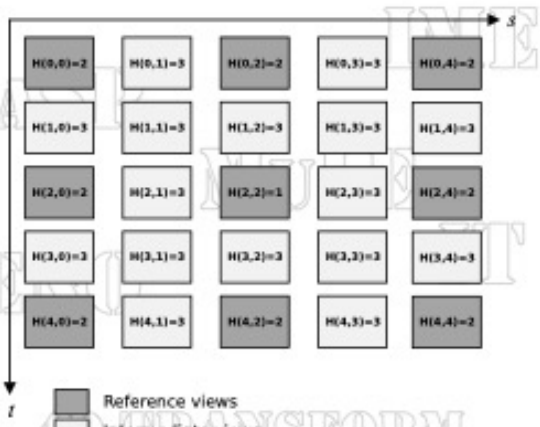
Extracted from [5].

Extracted from [10].

Depth-based warping

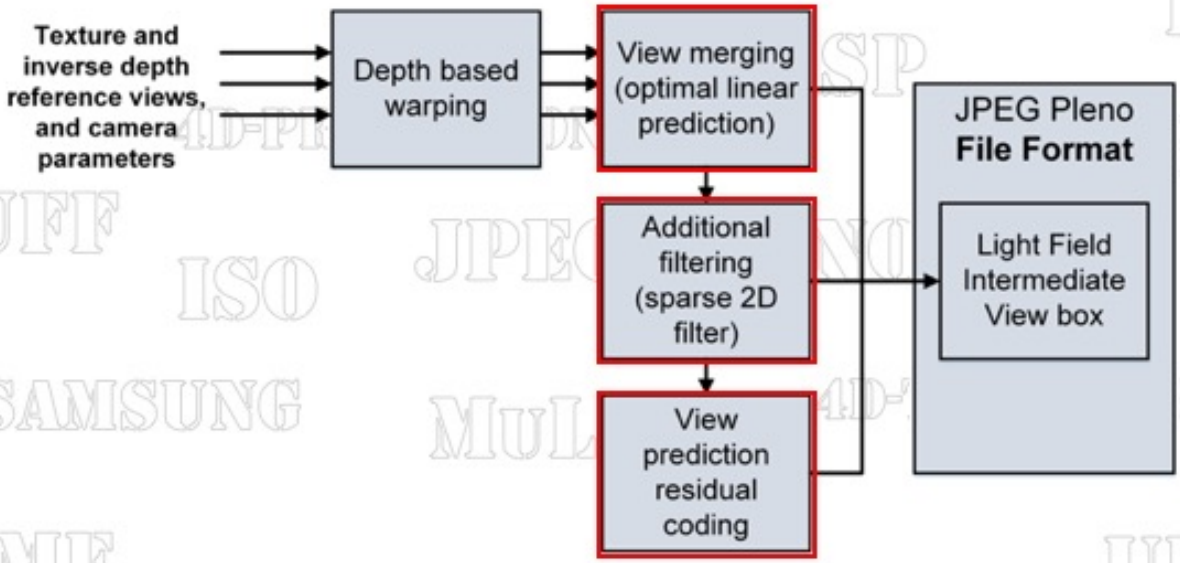
Warping the reference view X(20,98) to the camera position of the target view X(4,2)

Extracted from [10].



Extracted from [5].

View merging



Extracted from [8].

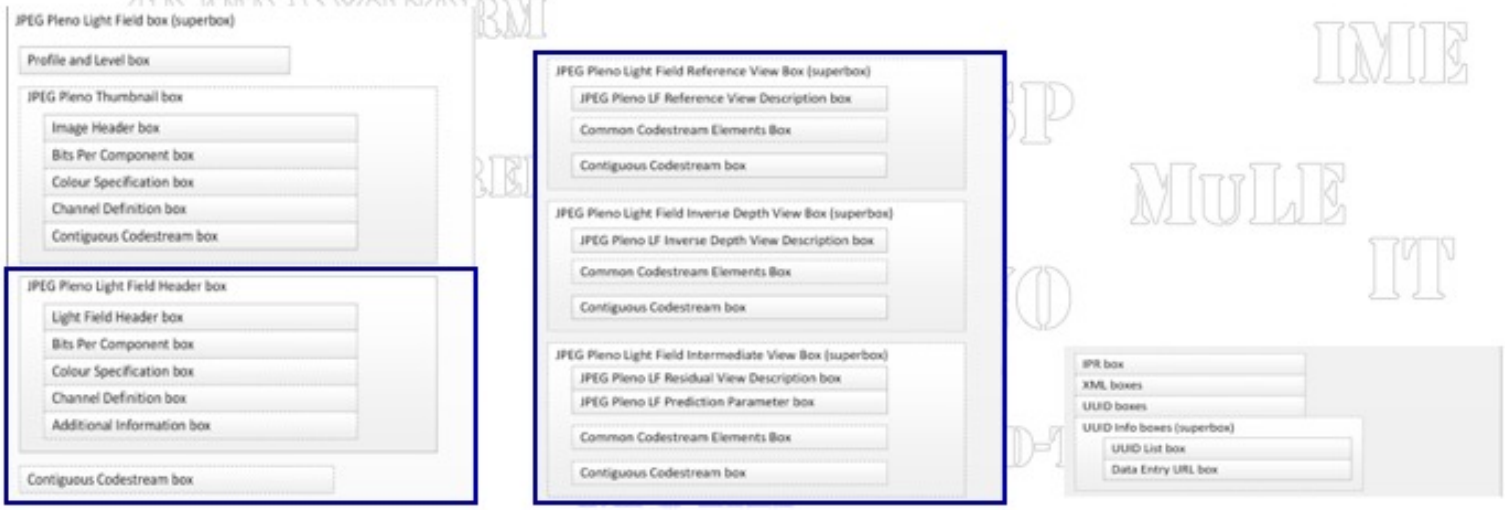
View merging

- The view merging stage requires inpainting.
- Then a second sparse prediction stage is applied, involving a 2D convolution with a spatially sparse filter.
- The intermediate-depth views are synthesized using a depth-based view warping algorithm, allowing each already processed hierarchical level to act as a source of reference views for subsequent hierarchical levels.
- The entire texture and depth components of any intermediate view are reconstructed using the prediction stages.
- The view prediction stage of the texture component is optionally followed by coding of the view prediction error (residue).

4D-TRANSFORM IME WASP 4D-PREDICTION MULE UFF IT ISO JPEG PLENO SAMSUNG MULE 4D-TRANSFORM IME UFRJ

Decoder

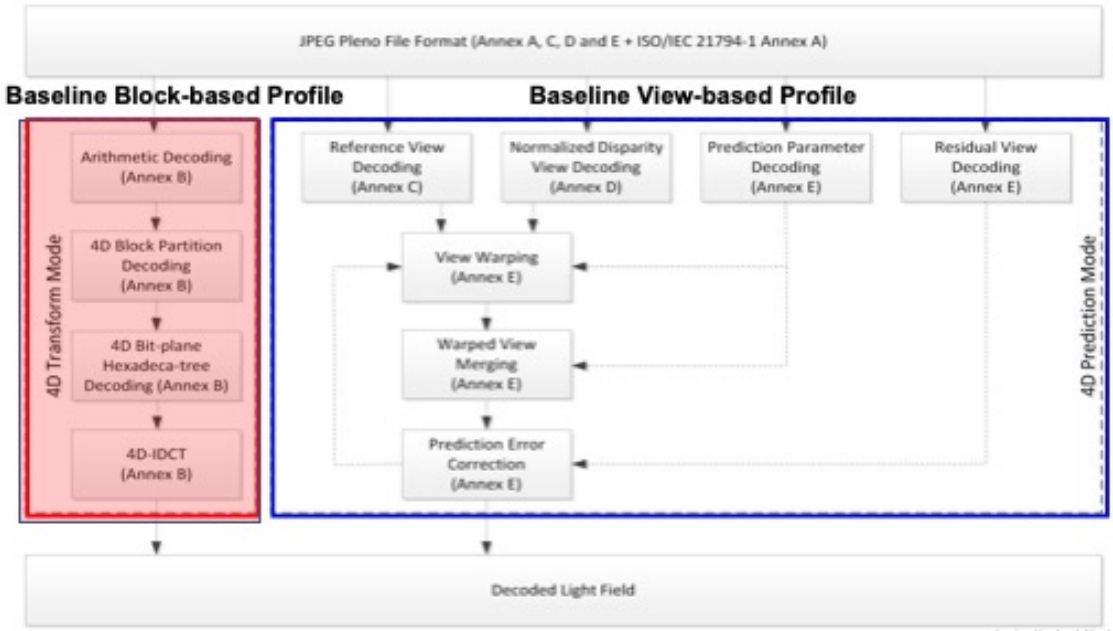
JPEG Pleno generic file format



4D-Prediction

Adapted from [8].

Decoder

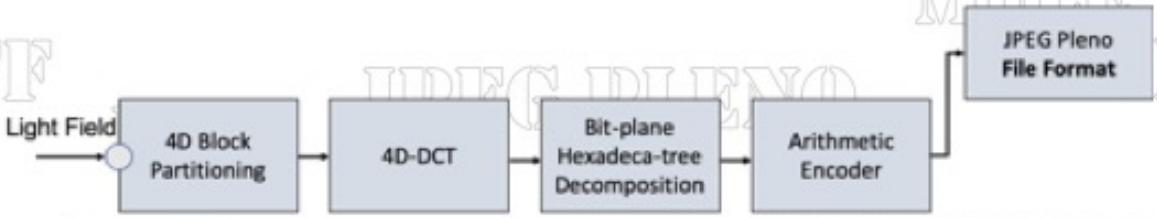


Adapted from [5].

Decoder

- Decodes the individual codestreams of the reference views.
- Decodes the individual codestreams of the residual views.
- Decodes the depth maps.
- Decodes the signaling of the parameters of the intermediate view prediction.
- Decodes the signaling of the configuration of the residual view encoding.
- Decodes the signaling of redundant header information from individual codestreams of the residual views.
- Reconstruct the intermediate views.

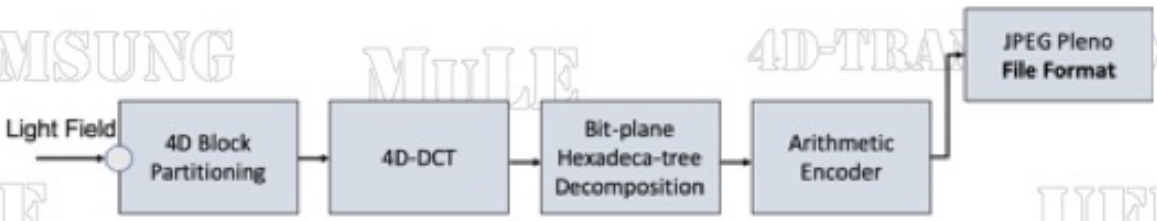
4D-Transform mode



Extracted from [8].

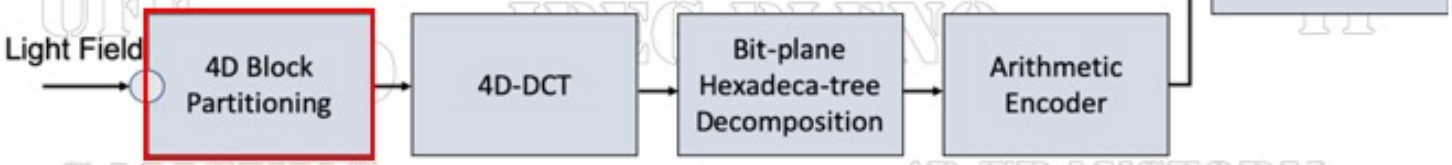
4D-Transform mode

- It is a fully 4D-native codec.
- The 4D-Transform (4DTM) coding mode is based on 4D block-based coding of the 4D LF data.
- It does not rely on any depth information.
- It does not need any camera parameter information.
- It is an evolution on the Multidimensional Light Field Encoder (MuLE) coding solution [11, 12].



Extracted from [8].

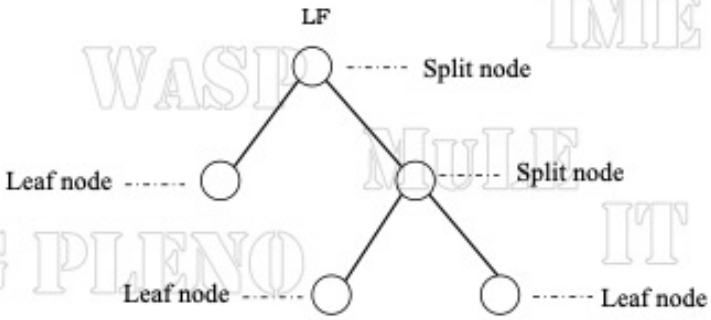
4D Block partitioning



Extracted from [8].

4D Block partitioning

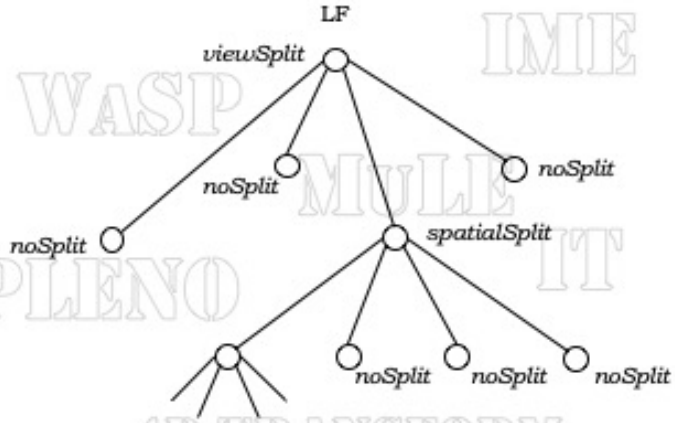
- The LF is partitioned into 4D blocks (hyperparallelepipeds) prior to the transform.
- The 4D Block Partitioning can be performed using variable-sized or multi-scale blocks.
- The nodes represent (and correspond to) some portion of the LF.
- The image may be reconstructed in a conventional form by traversing the tree by using recursive decomposition.



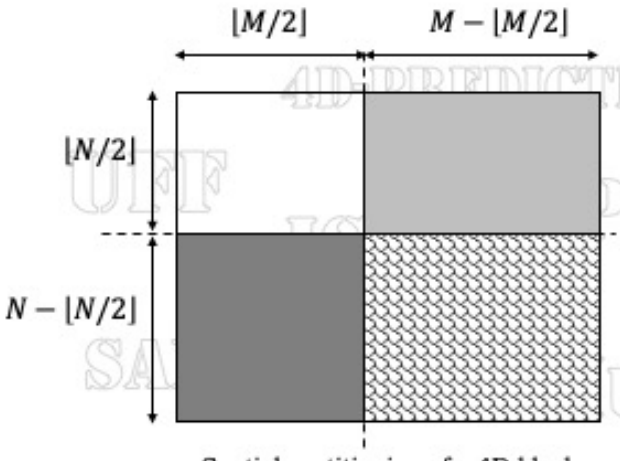
Extracted from [11]

4D Block partitioning

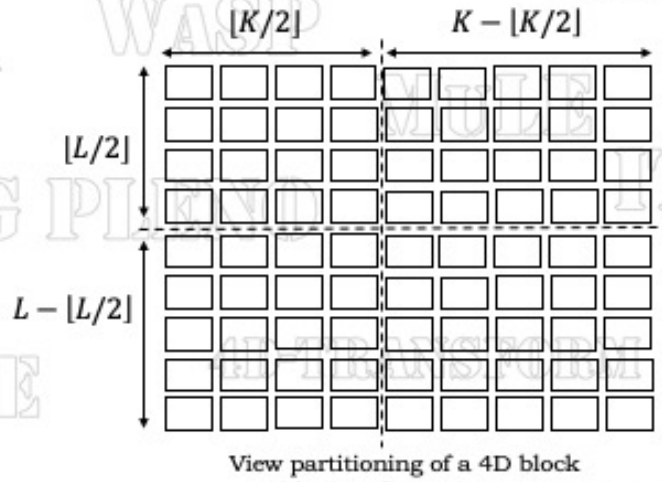
- Initially, the LF is pre-partitioned into non-overlapping 4D blocks of pre-determined maximum dimensions.
- Each block can be further partitioned into a set of non-overlapping 4D sub-blocks according to an RD criterion.
- The partition process aims at minimizing a Lagrangian coding cost.
- The optimal partition is signaled using a tree structure defined by ternary flags:
  - *noSplit* (transform)
  - *spatialSplit*
  - *viewSplit*



4D Block partitioning



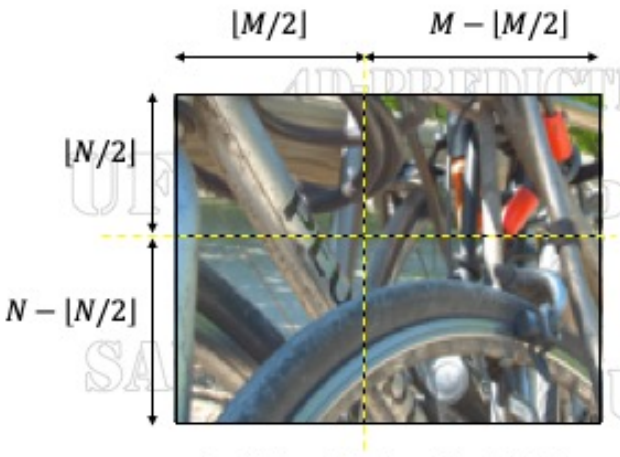
Spatial partitioning of a 4D block with spatial dimensions $M \times N$



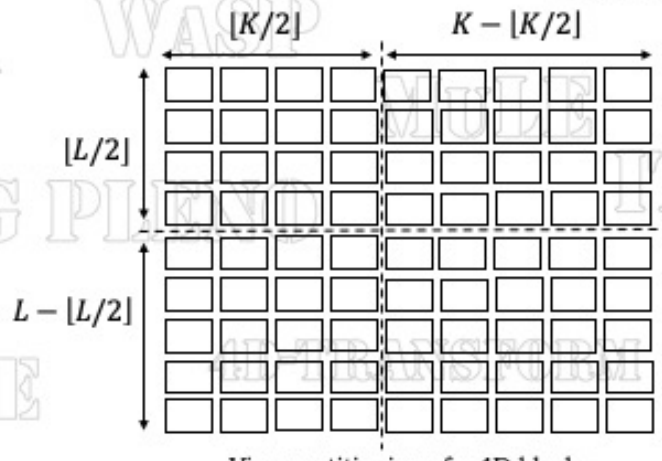
View partitioning of a 4D block with view dimensions $K \times L$

Extracted from [12]

4D Block partitioning



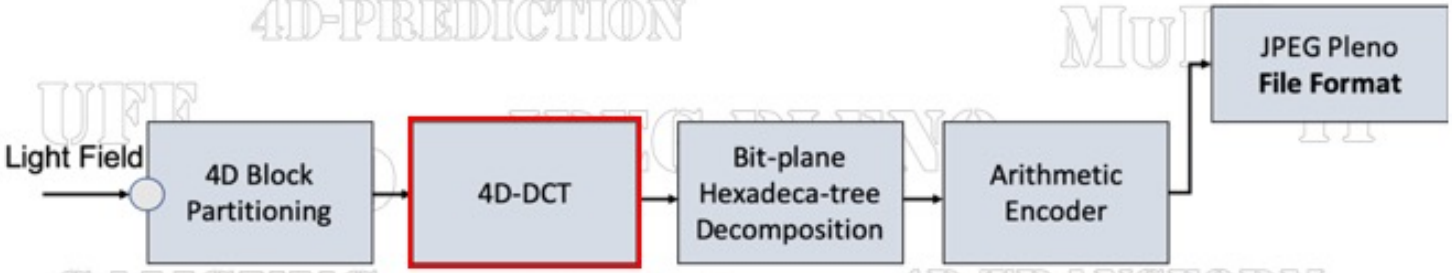
Spatial partitioning of a 4D block with spatial dimensions $M \times N$



View partitioning of a 4D block with view dimensions $K \times L$

Extracted from [12]

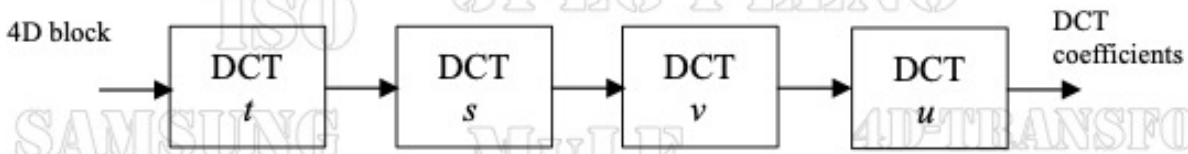
4D-DCT



Extracted from [8].

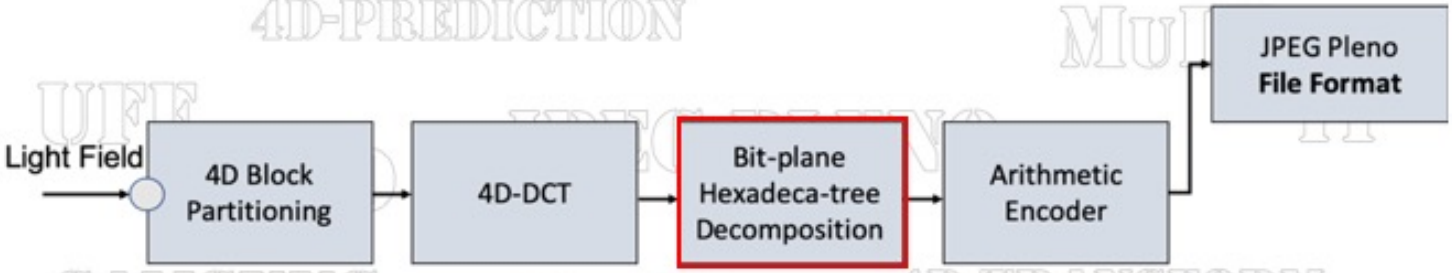
4D-DCT

- The 4D-DCT can be computed is a very efficient way using a separable transform.



Extracted from [12]

Bit-plane Hexadeca-tree Decomposition



Extracted from [8].

Bit-plane Hexadeca-tree Decomposition

- After the 4D-DCT is performed, the set of transform coefficients is sliced into 4D bit-planes.
- A transform coefficient is considered non-significant on a 4D bit-plane if its bits belonging to higher 4D bit-planes are all zero.
- Otherwise, the transform coefficient is considered to be significant.

Bit-plane Hexadeca-tree Decomposition

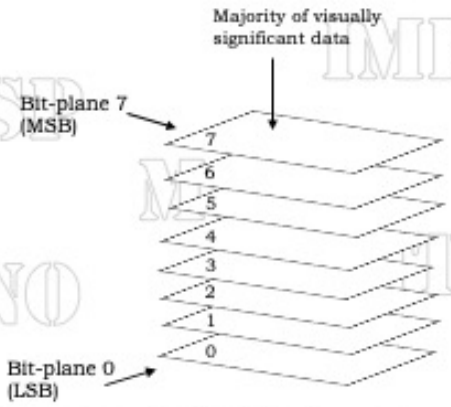
- A hexadeca-tree with ternary flags is used to group the non-significant transform coefficients and thus localize the significant transform coefficients.
  - either a block of transform coefficients containing a significant coefficient at the current bit-plane is split into 16 blocks in the four *s*, *t*, *u*, *v* dimensions;
  - or a block of transform coefficients not containing any significant coefficient at the current bit-plane is not split (indicates that for all pixels of the block that the binary representation of their magnitudes at the current bit-plane and above are zero);
  - or a block of transform coefficients containing a significant coefficient at the current bit-plane is discarded (the node has no descendants and is represented by an all-zeros block)

Bit-plane Hexadeca-tree Decomposition

- The 4D bit-planes are scanned from the most significant bit-plane to the least significant one, where the least significant 4D bit-plane being determined by the desired quantization level.
- The hexadeca-tree oriented clustering procedure returns the lowest Lagrangian cost and the resulting associated coding string, which defines the optimal tree.
- Both the hexadeca-tree bits and the bits from the transform coefficients are encoded using an adaptive arithmetic coder.

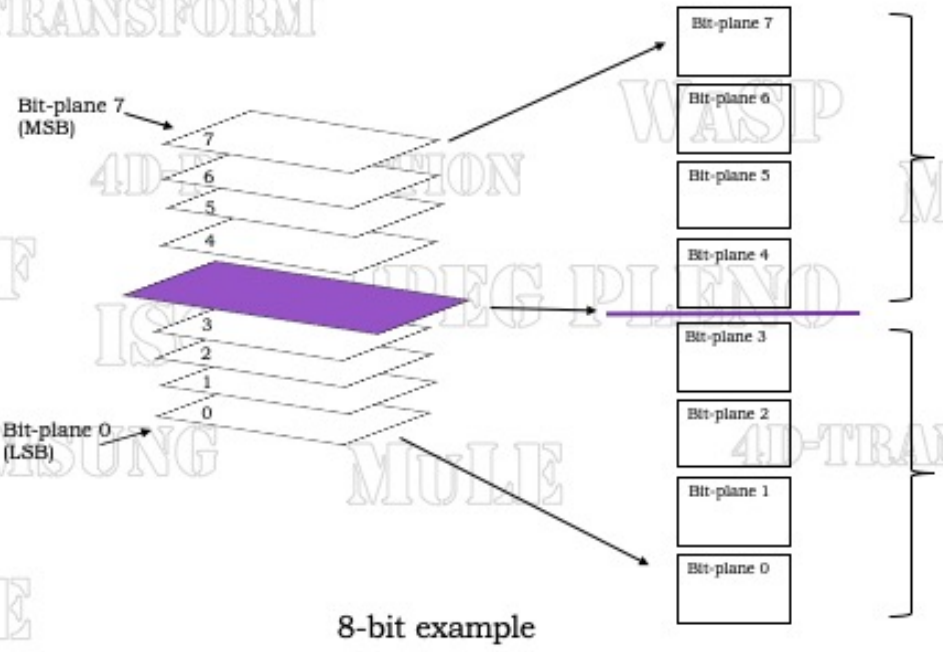
Bit-plane decomposition

- The bit-plane slicing is a technique that slices the image (data) into different planes (with each layer containing sequences of only binary digits).
- The first bit-plane (bit-plane 0) contains the least significant bit (LSB) and the last bit-plane contains the most significant bit (MSB),where the number of planes (layers) usually depends on the bit-depth of the image.



8-bit example

Bit-plane decomposition



8-bit example

Bit-plane decompostion

- The goal of the hexadeca-tree oriented bit-plane clustering is to quantize and segment a block of coefficients.
- The idea of this stage is to cluster the quantized 4D-DCT coefficients using a tree partitioning process combined with bit-plane clustering, as the quantized 4D-DCT coefficients in the 4D hyperparallelepipeds have to be represented as a sequence of symbols for posterior entropy coding.
- This process starts with the binary representation of the quantized 4D-DCT coefficients using a bit-depth corresponding to the used quantizer, which is determined using an RD criterion.

Bit-plane decomposition

- The idea is to generate coefficients clustering-based symbol to efficiently represent the large groups of coefficients quantized to zero.
- The central idea is to segment/split the 4D-DCT coefficients into segments/nodes that either have all their coefficients quantized as zero or have length 1, that is, contain just one coefficient.
- The hexadeca-tree structures associated with the bit-plane representation, are used for the efficient signaling of such segmentation as well as for the efficient encoding of the coefficients quantized as non-zero.
- A coefficient is considered non-significant on a bit-plane if its bits belonging to the bit-planes that are more significant than the current one are all zero. Otherwise, the coefficient is considered as significant.

Bit-plane decomposition

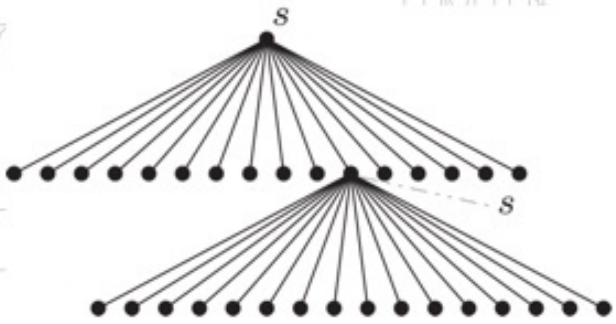
- The highest bit-plane in the coefficient array ,bpMAX, is set so that the largest of the 4D-DCT coefficient magnitudes is smaller than $2^{bpMAX+1}$ and larger than or equal to 2^{bpMAX}.
- The lowest bit-plane, bpMIN, corresponds to the desired quantization level for the 4D-DCT coefficients
  - to represent a coefficient up to a bit-plane bpMIN is equivalent to quantize it with a step size equal to 2^{bpMIN}.
- The bit-planes are scanned from the most significant to the least significant one, with the least significant bit-plane used being determined by the desired quantization level.

Hexadeca-tree

- If one applies an hierarchical data structure to represent a 4D data (LF), a hypervolume of four-space is divided into 16 equal sized regions (hexadeca).
- The four dimensions are given by the two view coordinates (s, t) and their corresponding 2D spatial coordinates (u, v), within each view.
- A hexadeca-tree is used to group the non-significant coefficients and to localize the significant coefficients.
- Both the hexadeca-tree bits and the bits from the coefficients are encoded using an adaptive arithmetic coder.

Hexadeca-tree

- To fully exploit the 4D structure and redundancy among the 4D-DCT coefficients the 4D-Transform mode employs a tree-oriented bit-plane clustering.
- At each node of the tree, the 4D block of transform coefficients is partitioned into 16 4D sub-blocks.
- The quantized 4D-DCT coefficients in the 4D hyperparallelepipeds have to be represented as a sequence of symbols for posterior entropy coding.
- The coefficients clustering-based symbol to efficiently represent the large groups of coefficients quantized to zero is accomplished by the hexadeca-tree clustering.



Extracted from [12]

Hexadeca-tree

- The encoder performs subdivisions of 4D regions of coefficients in 16 4D subregions.
- The decisions of splitting or not a node containing significant coefficients are RD optimized. The hexadeca-tree that orients the segmentation of the 4D-DCT coefficients is constructed by minimizing the Lagrangian cost $J = D + \lambda R$.
- The hexadeca-tree-oriented bit-plane clustering of the block of 4D-DCT coefficients is obtained by recursively subdividing the coefficients until all coefficients quantized as non-zero belong to a 1×1×1×1 4D block-size.

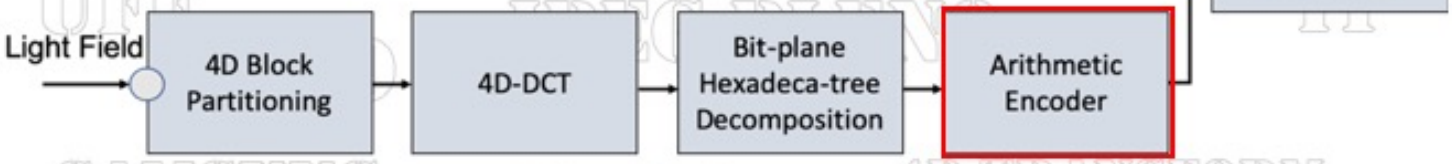
Hexadeca-tree

- If a subregion contains only zero coefficients or a single non-zero coefficient, it is not further subdivided and a symbol '0' is encoded. Otherwise, the subregion is further subdivided into 16 subregions and a symbol '1' is encoded, and so on.
- At the end of this process, the positions of all the non-zero coefficients are encoded by this '0's and '1's bitstream. The hexadeca-tree is the data structure that represents the whole subdivision process.
- Thus, a "0" indicating that there is no further subdivision of a 4D region is an extremely efficient way to represent many zeros using just one symbol (a $N \times N \times N \times N$ region contains N^4 zeros).
- It is a recursive procedure.

Hexadeca-tree

- The hexadeca-tree coding of this segmentation is uniquely specified by a sequence of ternary flags:
  - *codeBlock*
    - the node has no descendants and is represented by an all-zeros block
  - *splitBlock*
    - the node will have up to 16 children, each one associated to a sub-block with approximately half the length of the original block in all four dimensions
  - *lowerBitPlane* (the initial bit-plane is set to its maximum value)
    - indicates that for all pixels of the block that the binary representation of their magnitudes at the current bit-plane and above are zero
- The hexadeca-tree-oriented bit-plane clustering procedure returns the lowest Lagrangian cost and the resulting associated coding string, which defines the optimal tree.

Arithmetic encoder



Extracted from [8].

Arithmetic encoder

- The clustered coefficients and the signaling are fed into to an Adaptive Binary Arithmetic Coder (ABAC).
- It generates the coded representation for the input 4D block.
- The arithmetic encoder contexts are reset for every 4D block encoded so that the blocks are independently encoded, providing 4D block-level random access.

Arithmetic encoder

- The 4D coefficients, ternary flags, and probability context information generated during the encoding process, are input to an Adaptive Binary Arithmetic Coder (ABAC), which generates the coded representation for the input 4D block.
- The data is coded in the codestream as follows:

Symbol	Context range
DCT coefficient sign	0
DCT coefficients bits	1-32
Hexadeca-tree flags	33-98
4D-Block Partition flags	0

Arithmetic encoder

- DCT coefficient sign (DCT coefficient sign if ≠ 0):
  - A fixed probability model with two symbols is used for the signs
- DCT coefficients bits (the magnitude of a DCT coefficient):
  - Adaptive probability models are used to assign a context from the existing 32 (one for each bit-plane)
  - Coefficients dynamic range $[0, 2^{32})$
- Hexadeca-tree flags (the ternary flags: *lowerBitPlane, splitBlock, zeroBlock*)
  - Binary words, with the context associated with the current bit-plane (64 contexts)
  - Adaptive probability models are used
- 4D-Block Partition flags (the ternary partition flags: s*patialSplit, viewSplit, noSplit*)

4D-TRANSFORM IME WASP 4D-PREDICTION MULE UFF IT ISO JPEG PLENO SAMSUNG MULE 4D-TRANSFORM IME UFRJ

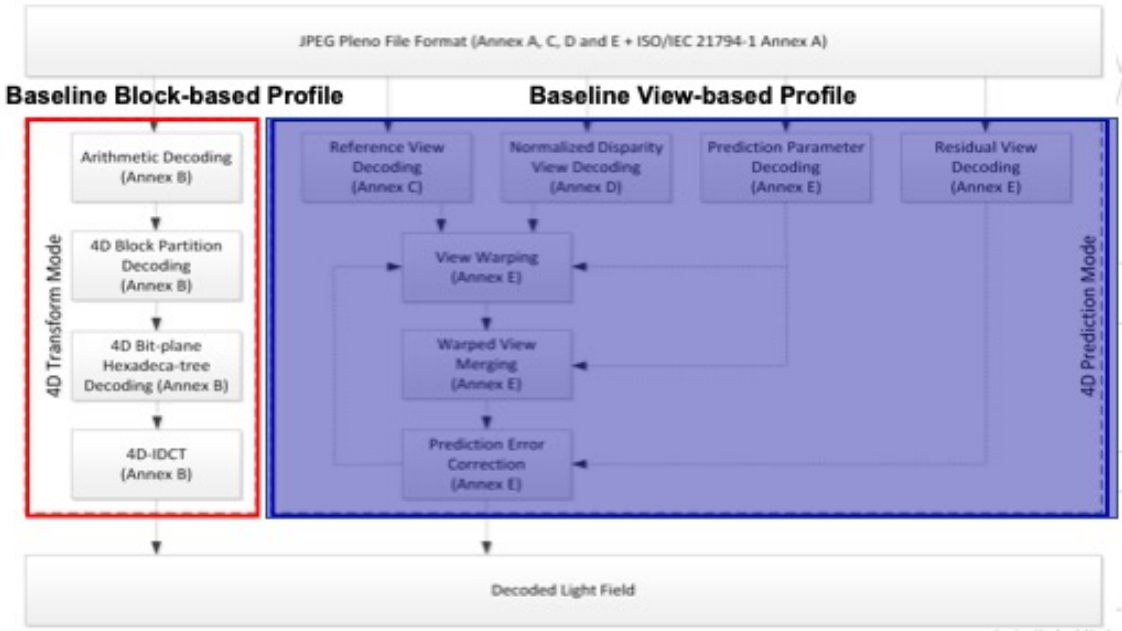
Decoder

JPEG Pleno generic file format

4D-Transform

Adapted from [8].

Decoder



Adapted from [5].

Decoder

- The decoding process is a straightforward one, where the input bits from the encoder output are processed by an entropy decoder, followed by an hexadeca-tree bit-plane decoder and an inverse transform block.
- The decoder makes no decisions.
- It simply decodes the codestream by:
  - reading the contexts of the ABAC
  - reading the flags to traverse the binary-tree
  - reading the flags to traverse the hexadeca-tree
  - reading the coefficients to perform the inverse DCT

JPEG Pleno LF
Common Test Conditions (CTC)

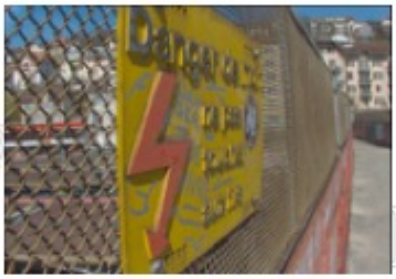
CTC

Test Materials

- JPEG Pleno Light Field Images Test Set
  - Acquisition/creation technology
    - Lenslet Lytro Illum camera
    - High Density Camera Array (HDCA)
    - Synthetic creation
  - Scene geometry
  - Spatial resolution
  - Number of views/perspectives
  - Bit depth
  - Texture

Datasets

Bikes



Danger de Mort

$13 \times 13 \times 625 \times 434$ (10-bit)

Stone Pillars Outside

Fountain&Vincent2

Datasets

Set2 2K sub
33 × 11 × 1920 × 1080 (10-bit)

Datasets

Laboratory 1
31 × 31 × 1936 × 1288
(8-bit, available as 10-bit)

Tarot Cards
17 × 17 × 1024 × 1024
(8-bit, available as 10-bit)

IME
UFRJ

Datasets

Greek
$9 \times 9 \times 512 \times 512$
(8-bit, available as 10-bit)

Sideboard
$9 \times 9 \times 512 \times 512$
(8-bit, available as 10-bit)

CTC

- JPEG Pleno Light Field Images Test Set
  - Bikes, Danger de Mort, Stone Pillars Outside and Fountain&Vincent 2
    - $13 \times 13 \times 625 \times 434$ (10-bit)
  - Set2 2K sub
    - $33 \times 11 \times 1920 \times 1080$ (10-bit)
  - Laboratory1 (Lab1)
    - $31 \times 31 \times 1936 \times 1288$ (8-bit, available as 10-bit)
  - Tarot Cards
    - $17 \times 17 \times 1024 \times 1024$ (8-bit, available as 10-bit)
  - Greek and Sideboard
    - $9 \times 9 \times 512 \times 512$ (8-bit, available as 10-bit)

CTC

Performance Metrics

- Quality
  - PNSR (Matlab© script)
  - SSIM (Matlab© script)
  - Bjøntegaard (Matlab© script)
- Coding conditions
  - Target rates
  - Rate metrics

HEVC (x265) Anchor (single GOP inter IPPP…)

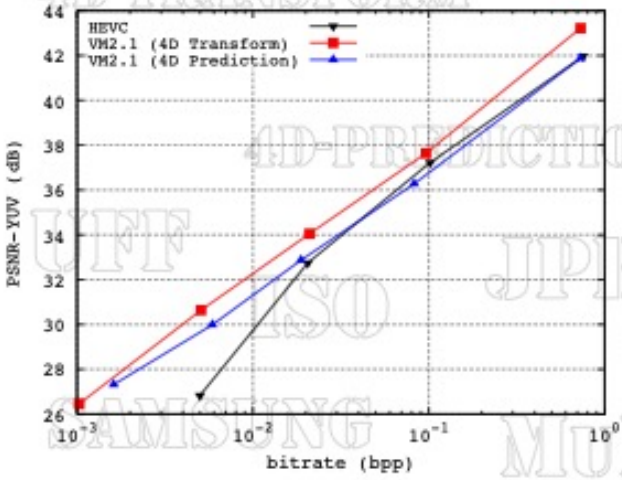
- Encoding
- Decoding

Comments

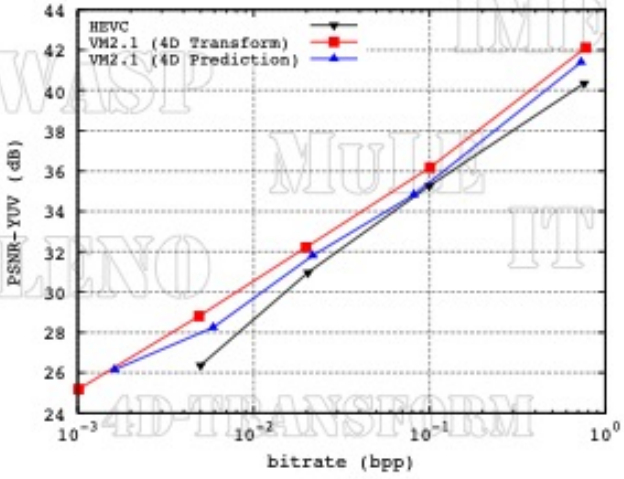
- Why two LF coding modes?
  - One for densely angular sampled datasets (4D-Transform mode)
  - One for sparsely angular sampled datasets (4D-Prediction mode)
- The 4D-Prediction mode relies on depth data to efficiently encode information provided by plenoptic cameras, outperforming the 4D-Transform mode when coding non densely angular sampled LF data.
- The availability or not of reliable depth data in a specific application scenario may be a critical factor to decide which JPEG Pleno LF coding mode to use.
- The 4D-Transform mode codec is conceptually simple and royalty-free.
- It does not rely on depth maps and offers block-based random access, specifically targeting densely angular sampled LFs such as lenslet LFs.

Performance assessment

Performance assessment

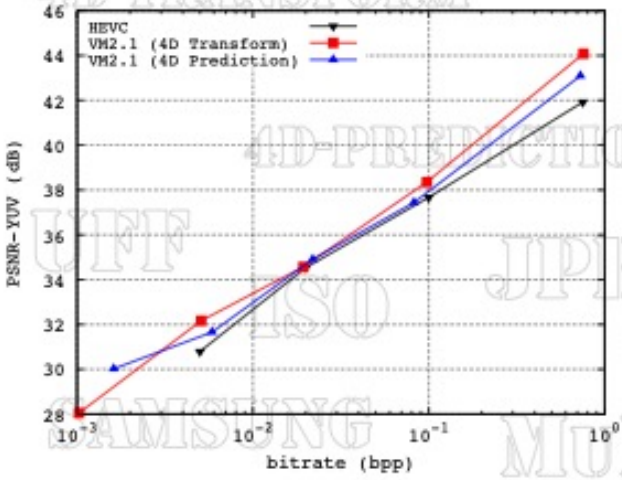


Bikes

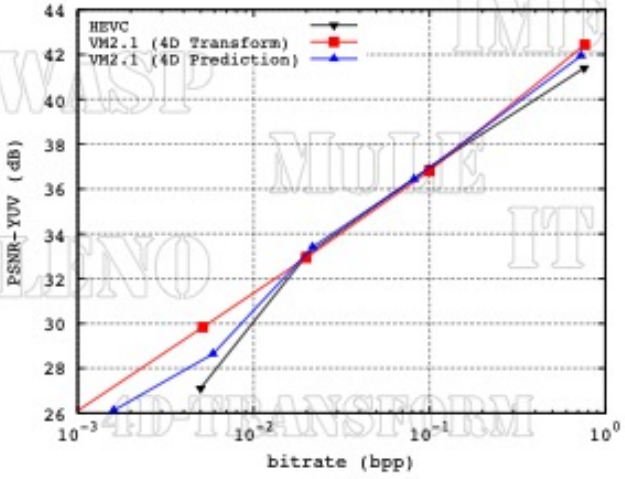


Danger (de Mort)

Performance assessment

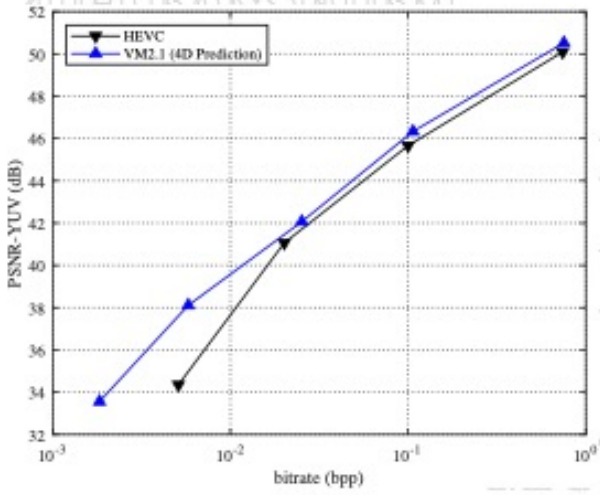


Stone Pillars Outside

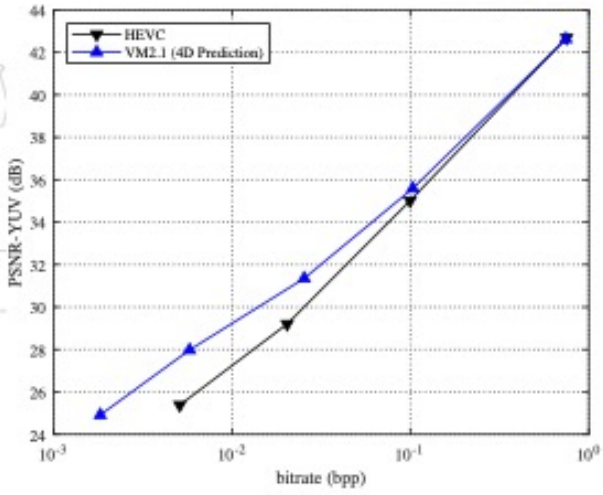


Fountain&Vincent2

Performance assessment

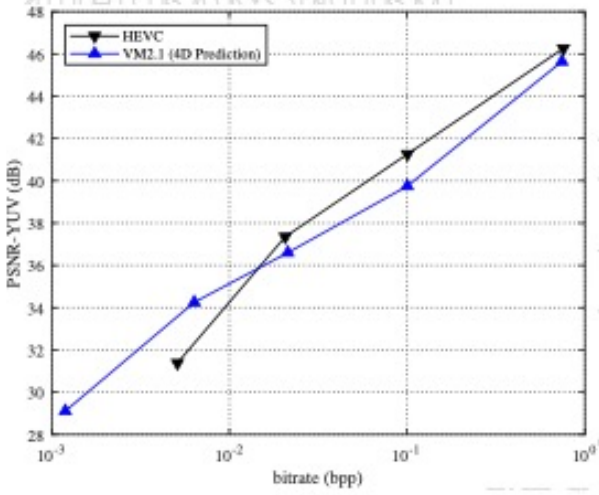


Greek

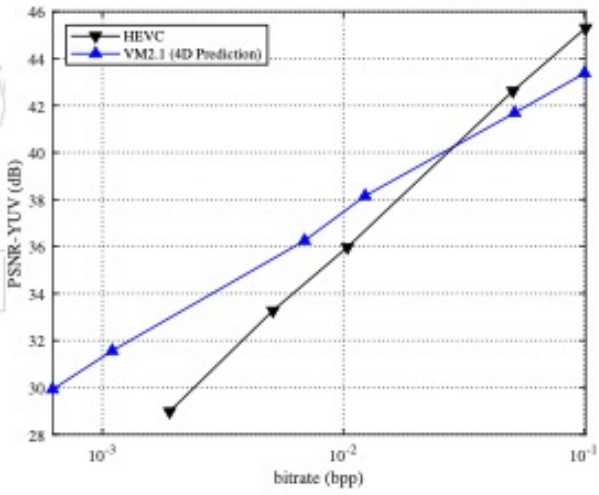


Sideboard

Performance assessment



Tarot



Set2 2K sub

4D-TRANSFORM IME WASP 4D-PREDICTION MULE UFF IT ISO SAMSUNG MULE 4D-TRANSFORM IME UFRJ

Conclusions

Conclusions

- 4D-Prediction mode
  - It presents a better RD performance than the 4D-Transform mode for low inter-view redundancy LFs (non densely sampled LFs).
  - It presents a worse RD performance than the 4D-Transform mode for high inter-view redundancy LFs (densely sampled LFs)
  - It relies on depth data, warping and inpainting techniques.
  - Uses existing codecs, such as JPEG 2000.

Conclusions

- 4D-Transform mode
  - It presents a better RD performance than the 4D-Prediction mode for high inter-view redundancy LFs (densely sampled LFs).
  - Each 4D-block is independently encoded, providing random access.
  - It is agnostic to depth information.
  - It does not employ any legacy codec.
  - It is a 4D-native light field coding solution.

References

[1] A. Gershun. The light field. Journal of Mathematics and Physics, 18(1- 4):51–151, 1939.

[2] E. Adelson, and J. Bergen. The Plenoptic Function and the Elements of Early Vision, Computational Models of Visual Processing, page 3--20. MIT Press, (1991).

[3] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96, pages 43–54, New York, NY, USA, 1996. ACM

[4] M. Levoy and P. Hanrahan. Light field rendering. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH'96, pages 31–42, New York, NY, USA, 1996. ACM.

[5] P. Astola, L. A. da Silva Cruz, E. A. B. da Silva, T. Ebrahimi, P. G.Freitas, A. Gilles, K.-J. Oh, C. Pagliari, F. Pereira, C. Perra, S. Perry,A. M. G. Pinheiro, P. Schelkens, I. Seidel, and I. Tabus, "JPEG Pleno:Standardizing a coding framework and tools for plenoptic imagingmodalities,"ITU Journal: ICT Discoveries, vol. III, 2020

[6] ISO/IEC JTC1/SC29/WG1, "JPEG Pleno Part 1 - ISO/IEC FDIS 21794-1," Jan. 2020, WG1N86056, 86th JPEG Meeting, Sydney, Australia

[7] ISO/IEC JTC1/SC29/WG1, "JPEG Pleno Part 2 - ISO/IEC FDIS 21794-2," Apr. 2020, WG1N87033, 87th JPEG Meeting, Erlangen (online), Germany

References

[8] P. Schelkens, P. Astola, E. A. B. da Silva, C. Pagliari, C. Perra, I. Tabus, et al., "JPEG Pleno light field coding technologies", Proc. SPIE, vol. 11137, pp. 391-401, Sep. 2019.

[9] P. Astola and I. Tabus, "WaSP: Hierarchical warping merging and sparse prediction for light field image compression", Proc. 7th Eur. Workshop Vis. Inf. Process. (EUVIP), pp. 1-6, Nov. 2018

[10] P. Astola and I. Tabus, "Coding of Light Fields Using Disparity-Based Sparse Prediction," in IEEE Access, vol. 7, pp. 176820-176837, 2019.

[11] M. B. de Carvalho, M. P. Pereira, G. Alves, E. A. B. da Silva, C. L. Pagliari, F. Pereira, et al., "A 4D DCT-based lenslet light field codec", Proc. 25th IEEE Int. Conf. Image Process. (ICIP), pp. 435-439, Oct. 2018

[12] 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. B. Da Silva., "The JPEG Pleno Light Field Coding Standard 4D-Transform Mode: How to Design an Efficient 4D-Native Codec," in IEEE Access, vol. 8, pp. 170807-170829, 2020.

[13] Pereira, F., Pagliari, C., da Silva, E. A. B., Tabus, I., Amirpour, H., Bernardo, M., and Pinheiro, A.,"ISO/IEC JTC 1/SC29/WG1N84049: Information technology - JPEG Pleno Light Field Coding CommonTest Conditions V3.3, 84th JPEG Meeting, Brussels, Belgium," Jul. 2019.

[14] JPEG Pleno Light Field Verification Model 2.1. https://gitlab.com/wg1/jpeg-pleno-vm

4D-TRANSFORM
IME
4D-PRED
MuLE
UFF
IT
ISO
SAMSUNG
TRANSFORM
IME
UFRJ