HDR Image and Video Compression

dr. Francesco Banterle

francesco.banterle@isti.cnr.it

HDR Images and Frames

- The main problem with HDR images is that they require floating point encoding for representing all intensities values that HVS can see

- Smart formats exist:

  - RGBE

  - LogLuv

  - Half-precision

HDR Formats: comparisons

Encoding	Color Space	Bpp	Dynamic Range (log_{10})	Relative Error (%)
IEEE RGB	full RGB	96	79	0.000003
RGBE	positive RGB	32	76	1.0
LogLuv24	logY + (u,v)	24	4.8	1.1
LogLuv32	logY + (u,v)	32	38	0.3
Half RGB	RGB	48	10.7	0.1

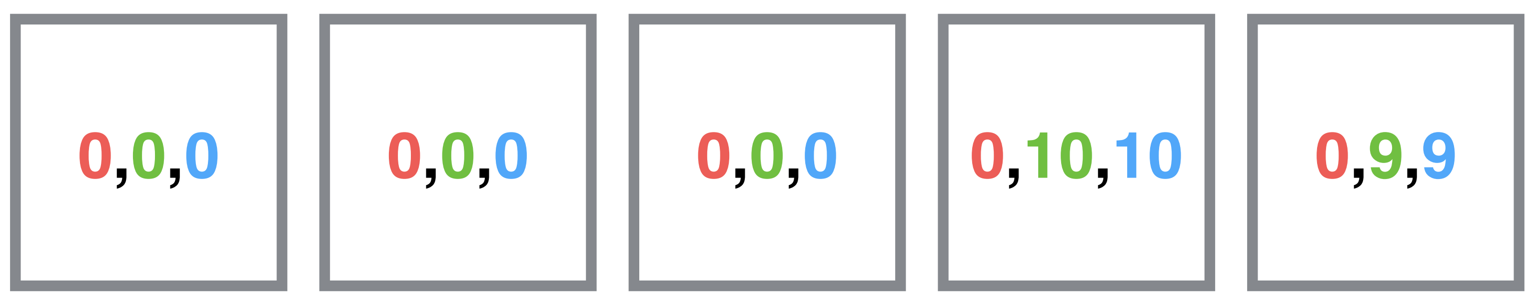
HDR Images and Frames

- Even encoding with these there are some issue:
  - A full HD image, 1920x1080, encoded with RGBE (32-bit per pixel or bpp)
    - 7.9Mb for a single frame!

a quick recall…

LDR Images Compression

- A solution for compression is RLE:



- Encoded as:

  Value: 0 Count: 10; Value: 10 Count: 2;

  Value: 0 Count: 1; Value: 9 Count: 2

LDR Images Compression

- RLE or other string compression methods are loseless —> no loss of information
- The HVS does not notice small variations
- The signal is locally similar in patches without edges

LDR Image Compression: Binary Truncation Coding

- **Idea**: to compress images taking into account of pixel values locality and assuming two distributions per block
- The method is lossy —> information is lost!
- Bpp is constant
  - Grayscale images: 2bpp
  - Color images: 4-8bpp

LDR Image Compression: Binary Truncation Coding

22	33	44	47
24	31	38	40
25	24	30	32
20	21	24	22

Original Block

0	1	1	1
0	1	1	1
0	0	1	1
0	0	0	0

$M = 30$

$M_0 = 23 \quad M_1 = 38$

23	38	1	38
23	38	38	38
23	23	38	38
23	23	23	23

Decoded Block

2 bytes (M_0 and M_1), 2 byte the block —> 4 byte

This means 2bpp instead of 8bpp (for a gray scale image)

JPEG

- **Idea**: to take advantage that the HVS perceive differently high and low frequencies
- Steps:
  - Color conversion: YCrCb
  - DCT
  - DCT coefficient quantization
  - Encoding

JPEG: YCrCb

- **Idea**: to separate color information, or chrominance, and luminance in values
- Chrominance can be subsampled
    - Why?
        - HVS perceives less color variations
- Which color space? YCrCb, an ITU-R BT.601 standard

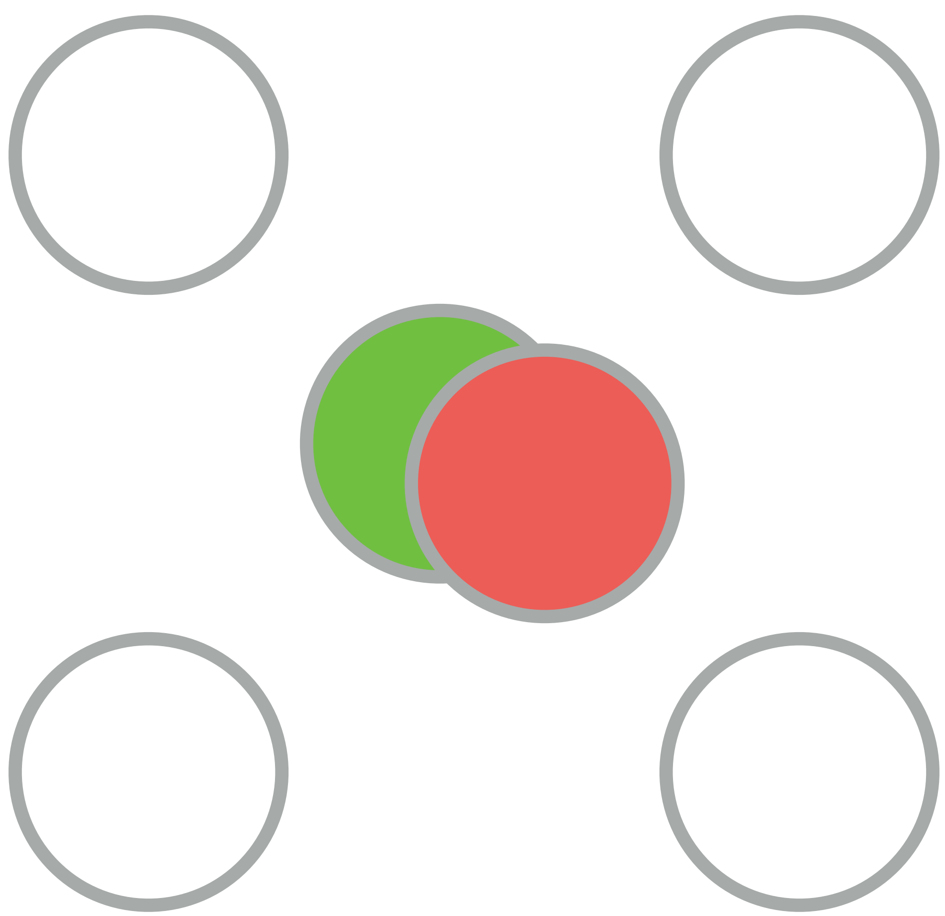
JPEG: YCrCb

$$\mathbf{M}_{RGB \rightarrow YCrCb} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & 0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix}$$

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \mathbf{M}_{RGB \rightarrow YCrCb} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

JPEG: Chroma Subsampling

- Chroma subsampling (4:2:0)



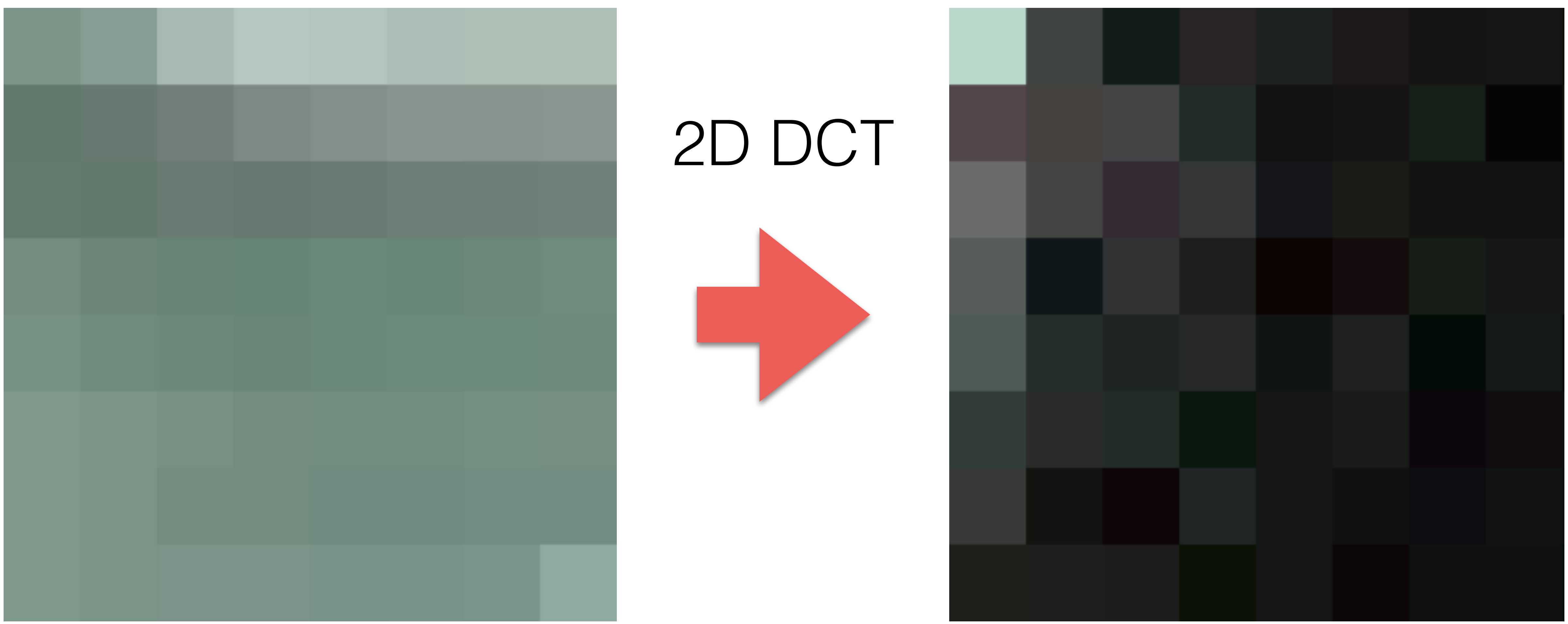
JPEG: Discrete Cosine Transform

- Discrete Cosine Transform (DCT) separates a block (8x8 in JPEG) into low and high frequency bands.
- DCT is invertible and separable
- DCT is related to FFT, but only real coefficients

$$F(u,v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Lambda(i)\Lambda(j) \cos\left(\frac{\pi u}{2N}(2i+1)\right) \cos\left(\frac{\pi v}{2N}(2j+1)\right) f(i,j)$$

$$\Lambda(x) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } x = 0 \\ 1 & \text{otherwise} \end{cases}$$

JPEG:
Discrete Cosine Transform



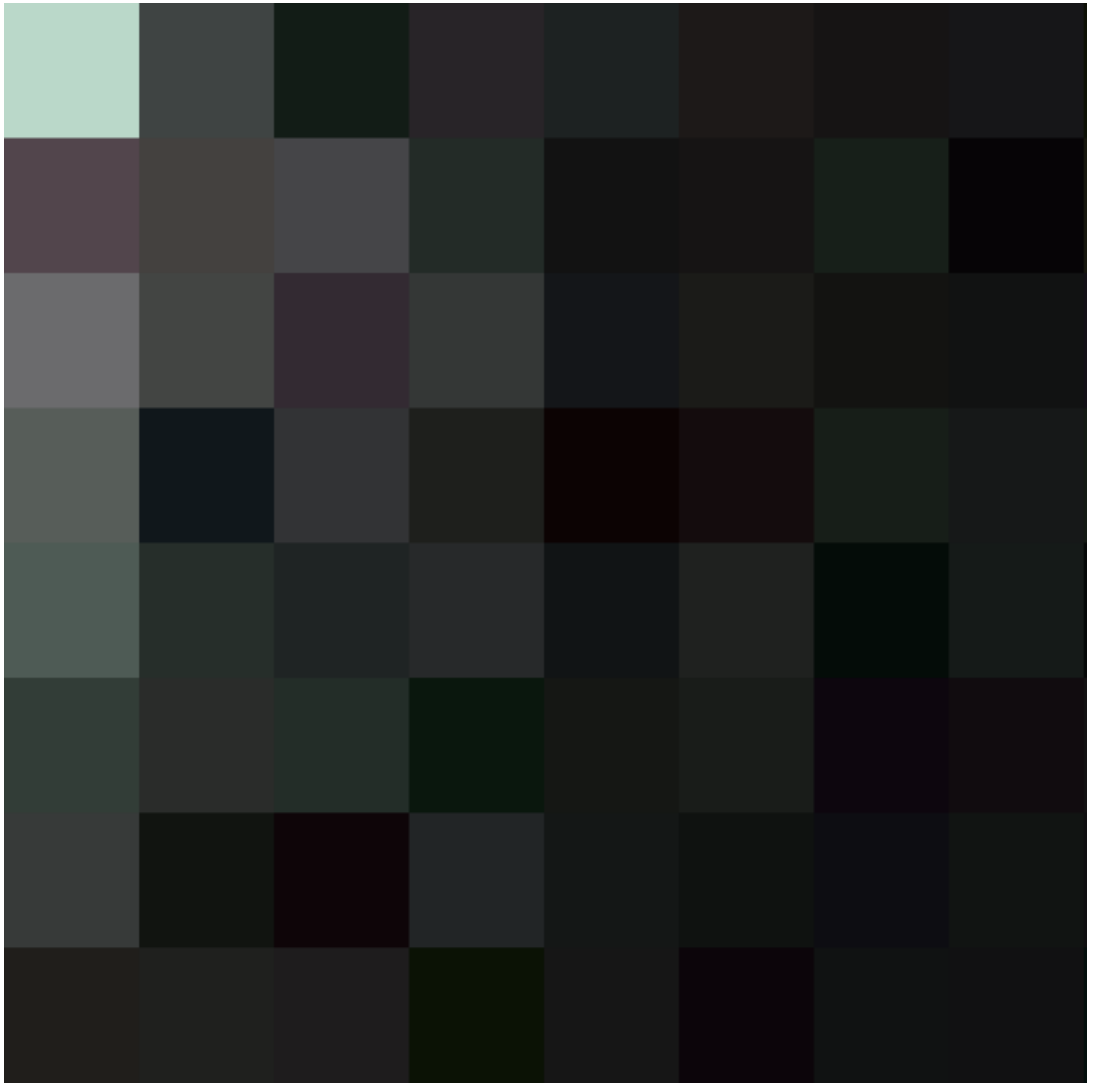
JPEG: Quantization

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

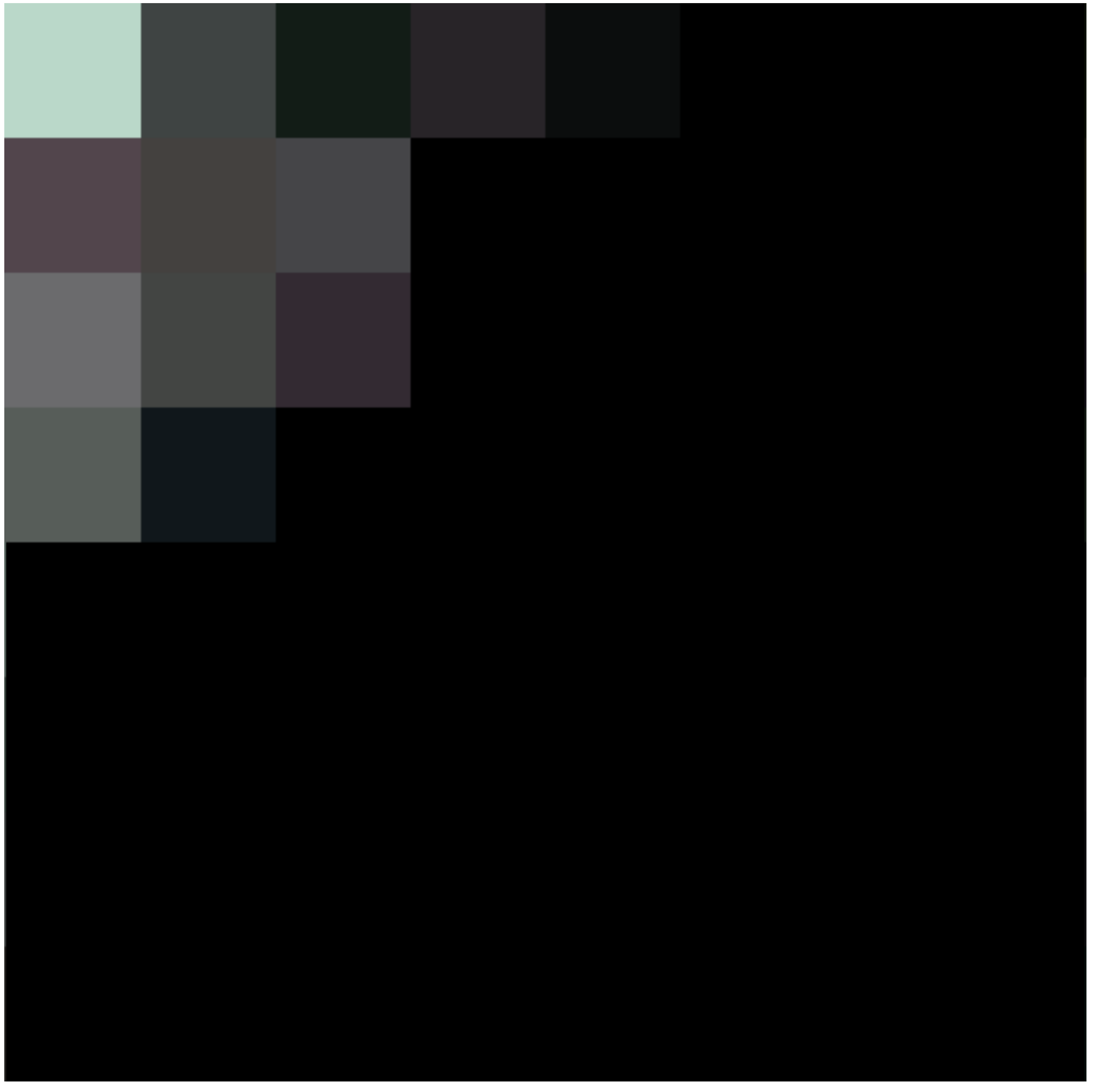
Quantization matrix

Values are in [-128, 128], then encoded in [0,255]

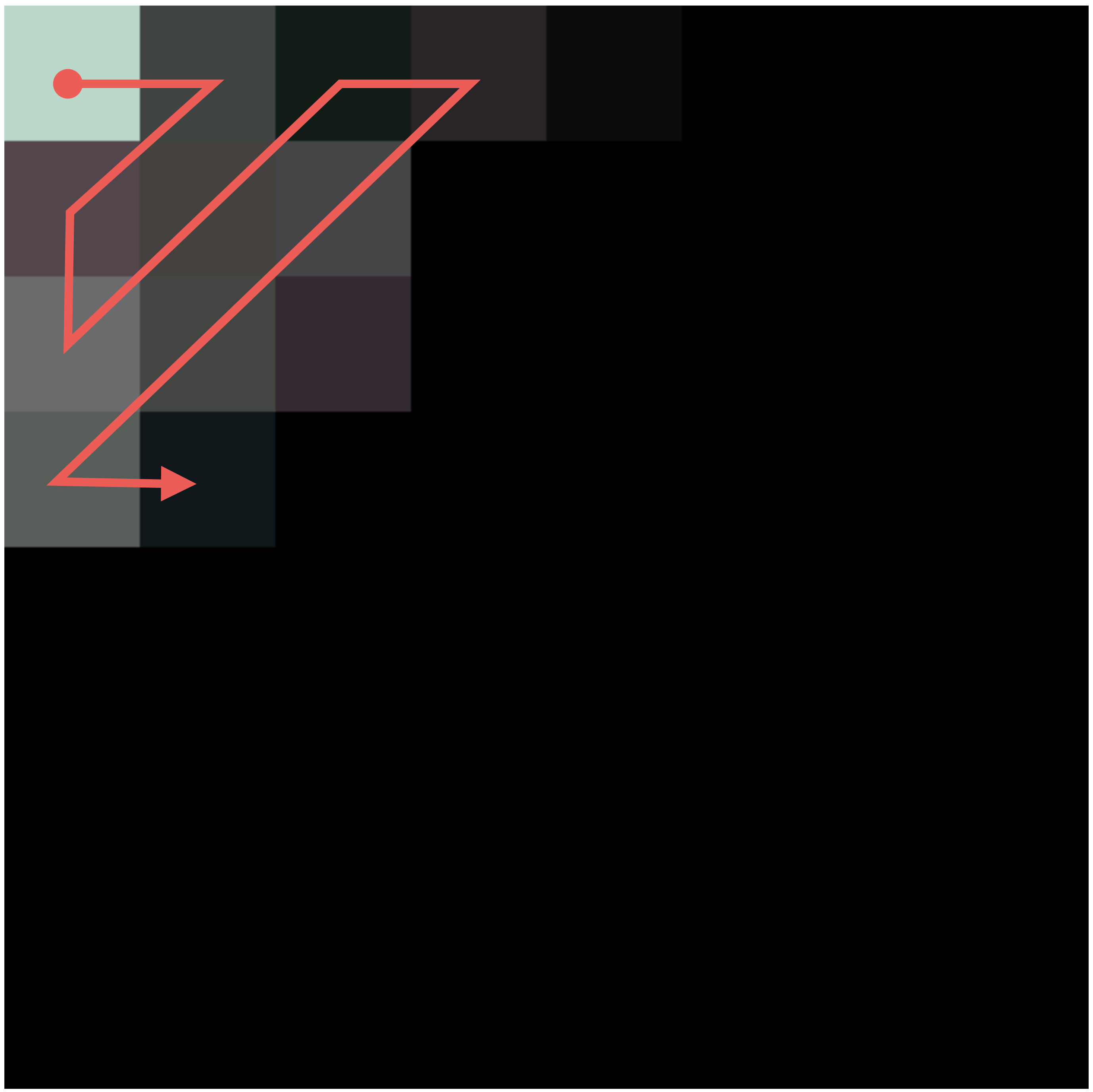
JPEG: Quantization



JPEG: Quantization



JPEG: Encoding



Similar frequencies are put together

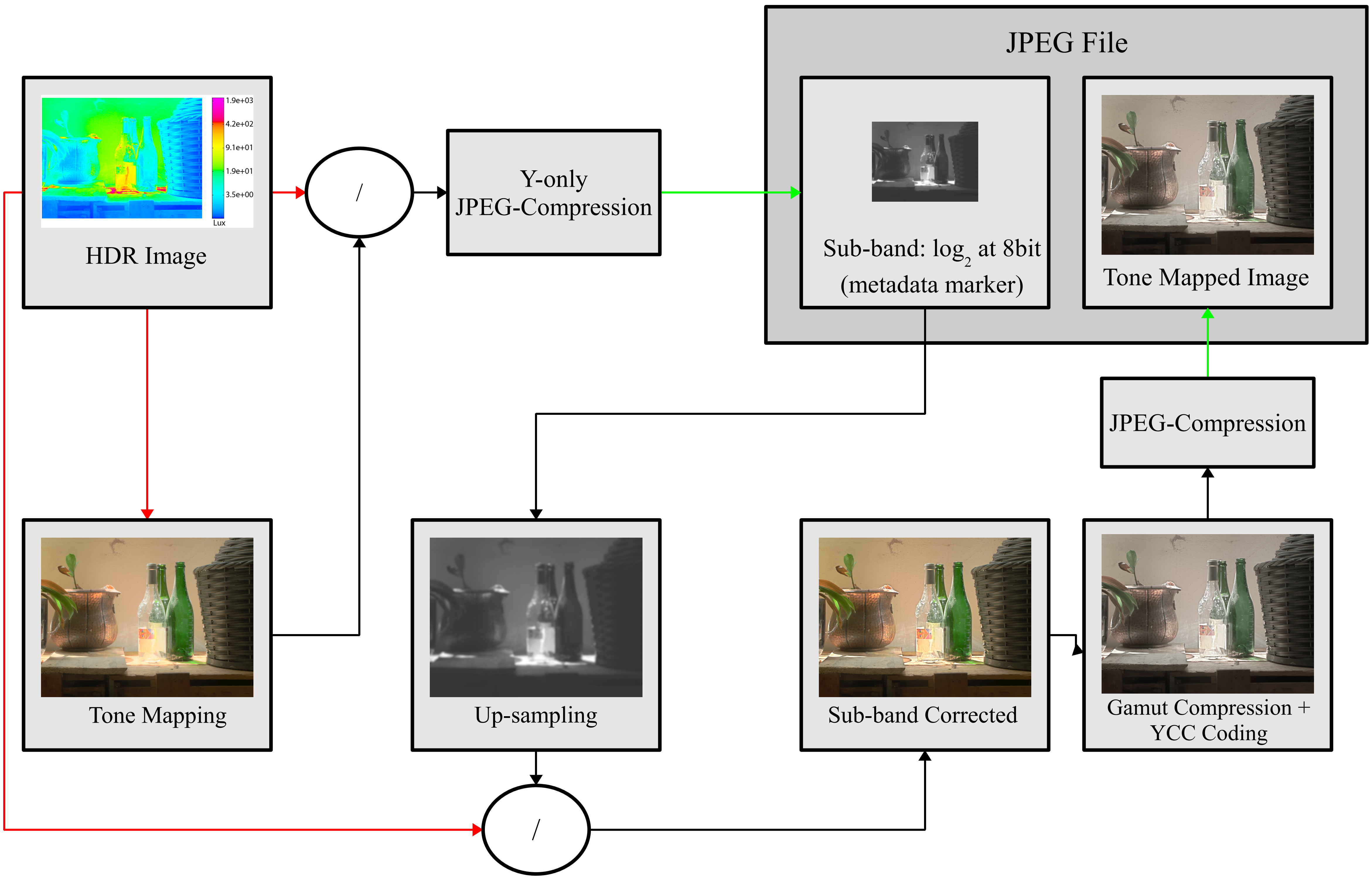
Values are encoded using:
- Huffman
- Arithmetic Encoding

and now back to
HDR images…

JPEG-HDR

- **Idea**: to tone map an HDR image and store tone mapped version using HDR [Ward and Simmons 2004]
- How to reconstruct the HDR image?
  - to store the inverse of the TMO spatially
  - Spatial inverse TMO is stored at low resolution in 64Kb

JPEG-HDR



JPEG File
HDR Image
/
Y-only
JPEG-Compression
Sub-band: log2 at 8bit
(metadata marker)
Tone Mapped Image
JPEG-Compression
Tone Mapping
Up-sampling
Sub-band Corrected
Gamut Compression +
YCC Coding
/

HDR JPEG-2000

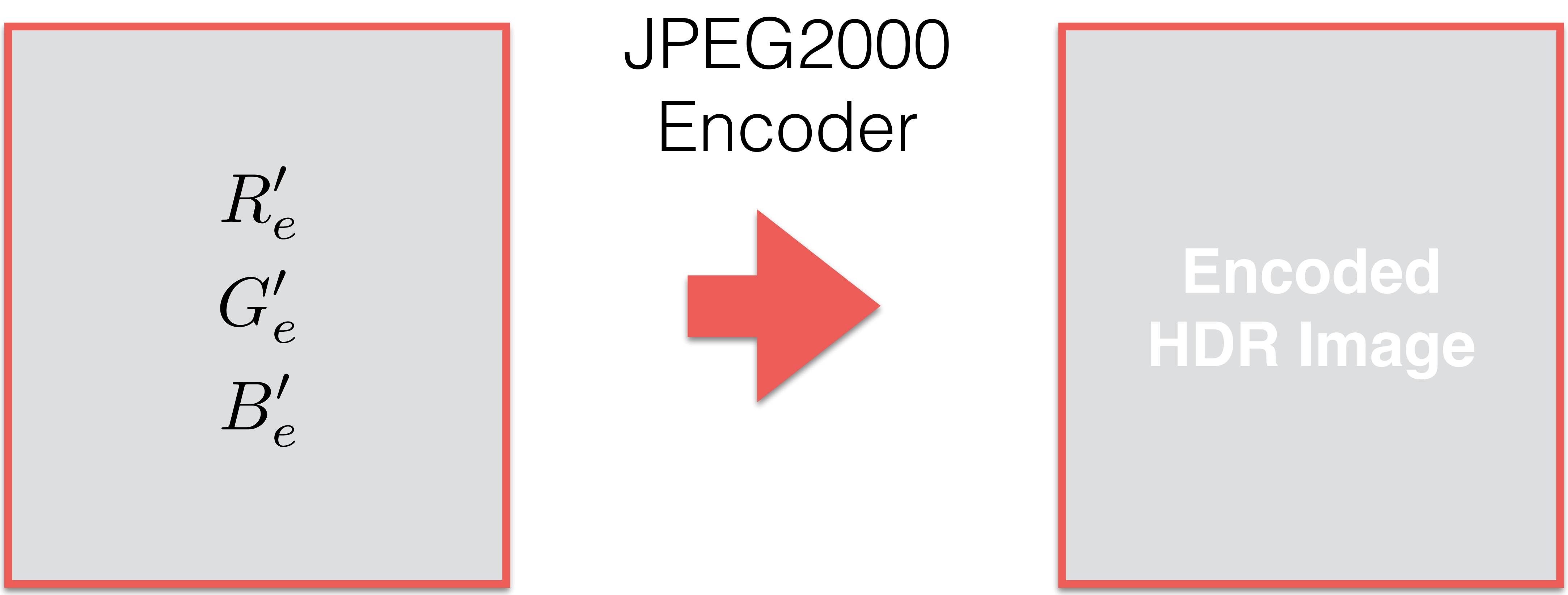
- **Idea**: JPEG-2000 standard allows 16-bit integer encoding per color channel!
- What to do:
  - For each color channel:
    - Apply a logarithm base two
    - Compute maximum value
    - Compute minimum value

HDR JPEG-2000

$$C_e(\mathbf{x}) = \frac{\log_2(C(\mathbf{x}) + \epsilon) - \log_2(C_{\max} + \epsilon)}{\log_2(C_{\max} + \epsilon) - \log_2(C_{\min} + \epsilon)} \qquad \epsilon > 0$$

$$C'_e(\mathbf{x}) = \left\lceil (2^{16} - 1) C_e(\mathbf{x}) \right\rceil$$

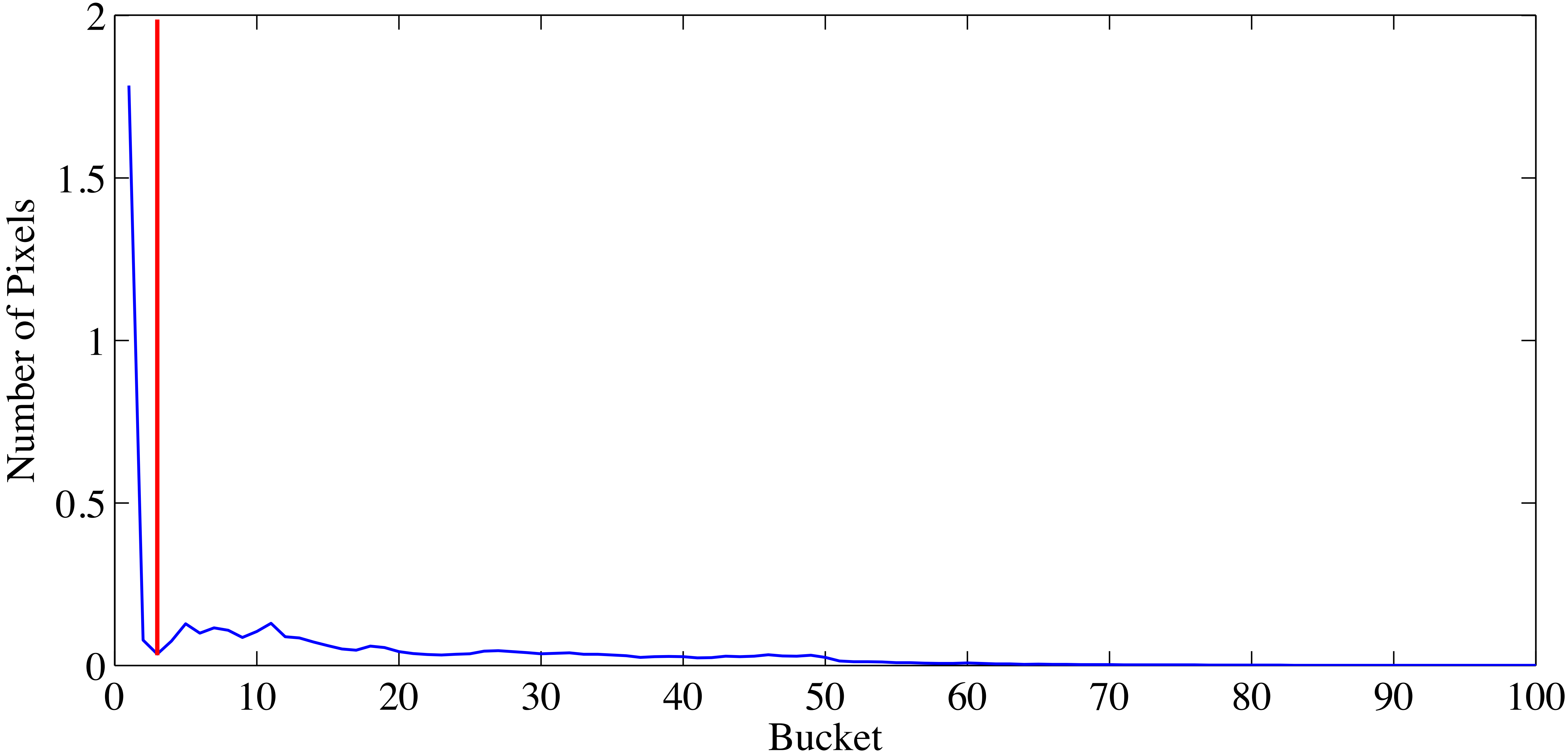
HDR JPEG-2000



HDR Split

- **Idea**: to separate brigh and dark areas in an image via histogram and to encode them separately [Wang et al. 2007]
- How?
  - Minimization function for finding a separation axis in the histogram
  - Encoding with S3TC a BTC method
  - The method can fail when separation axis do not exist

HDR Split



HDR Split

Dark areas

Bright areas

Spatially Varying RGBE

- **Idea**: RGBE works very well, why not extending to take advantage of spatial coherency? [Boschetti et al. 2010]

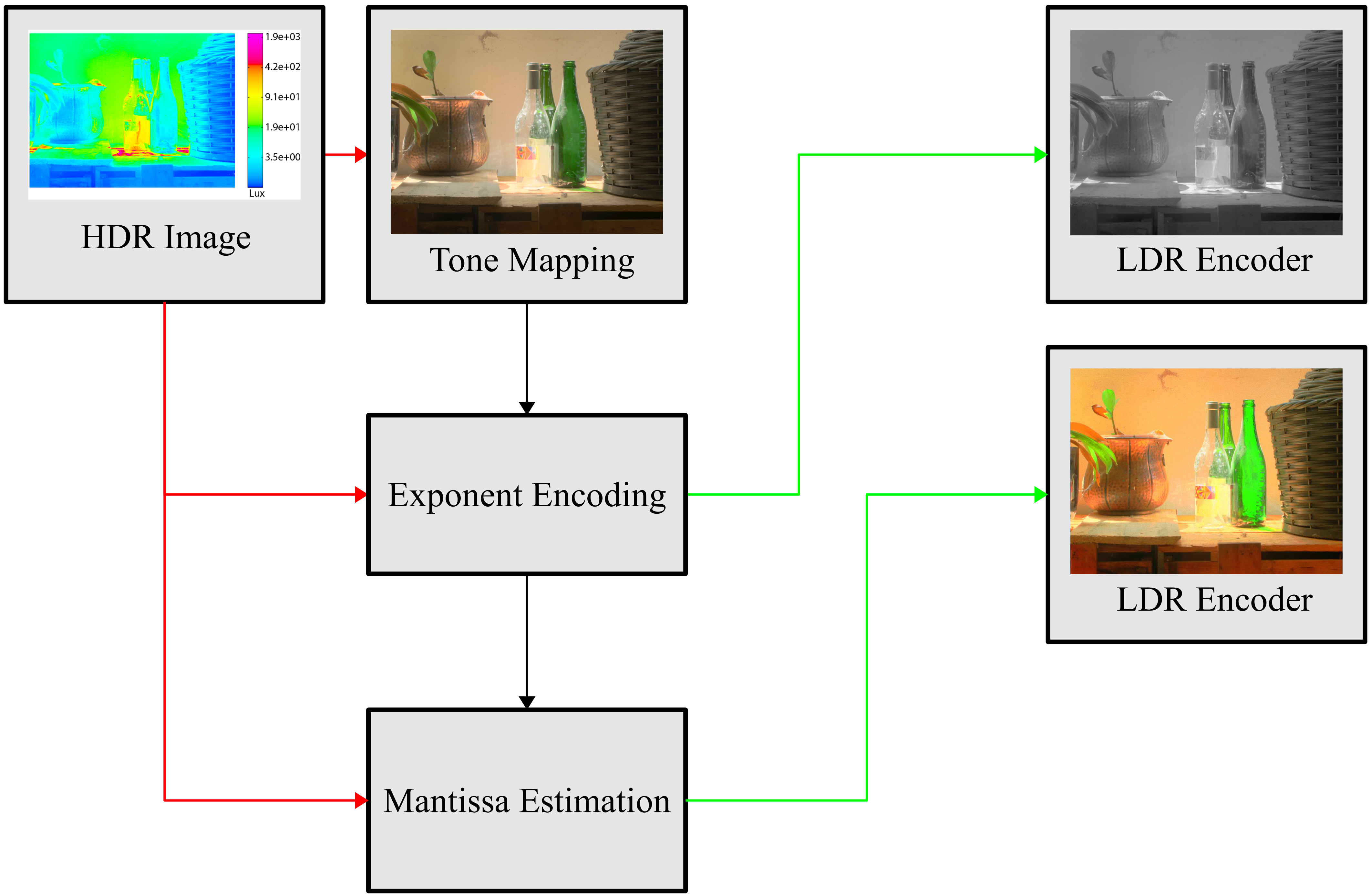
$$E_m = \left\lceil \log_2 \max(R, G, B) + 128 \right\rceil$$

$$R_m = \left\lfloor \frac{256R}{2^{E_m - 128}} \right\rfloor$$

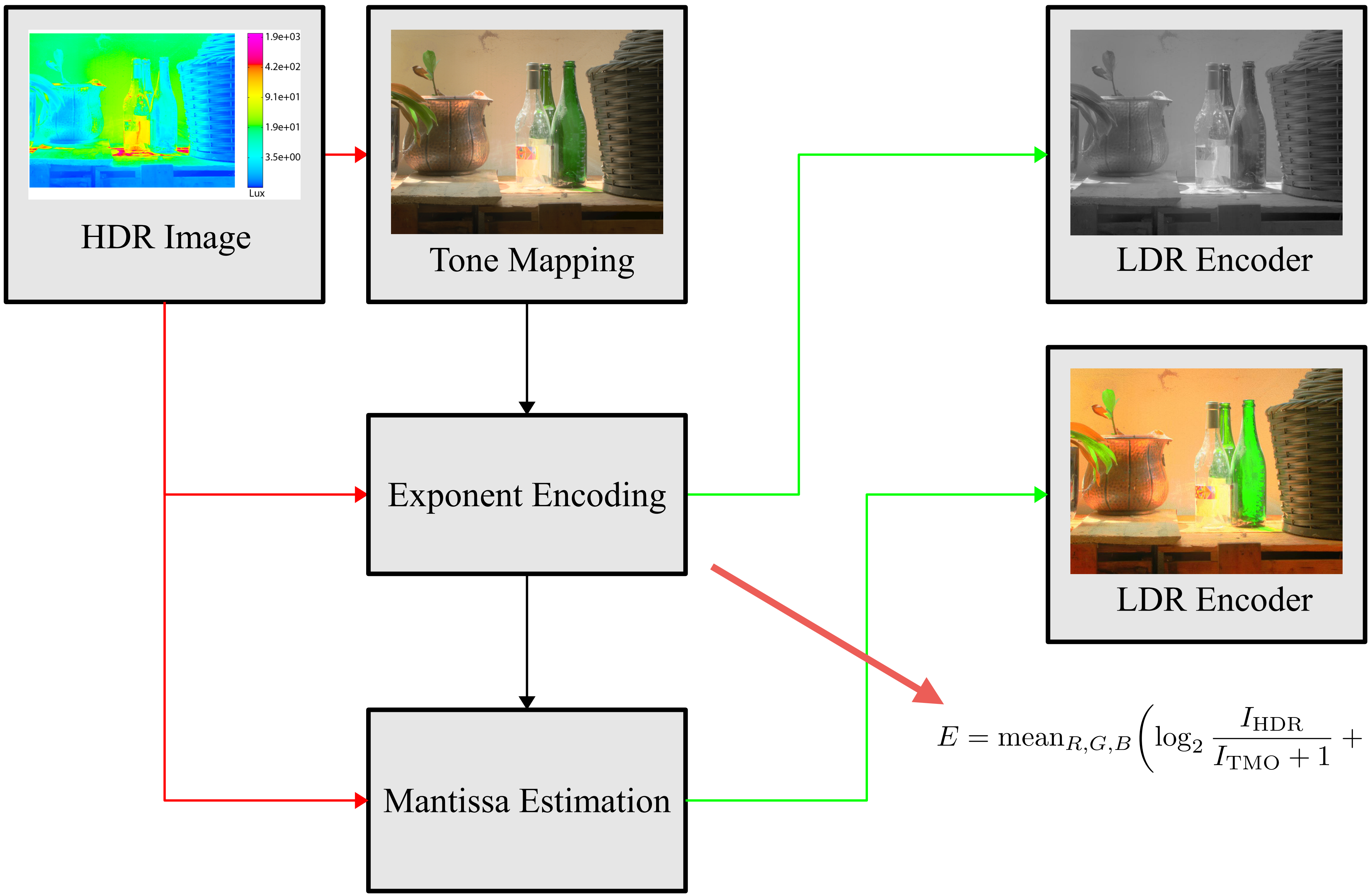
$$G_m = \left\lfloor \frac{256G}{2^{E_m - 128}} \right\rfloor$$

$$B_m = \left\lfloor \frac{256B}{2^{E_m - 128}} \right\rfloor$$

Spatially Varying RGBE

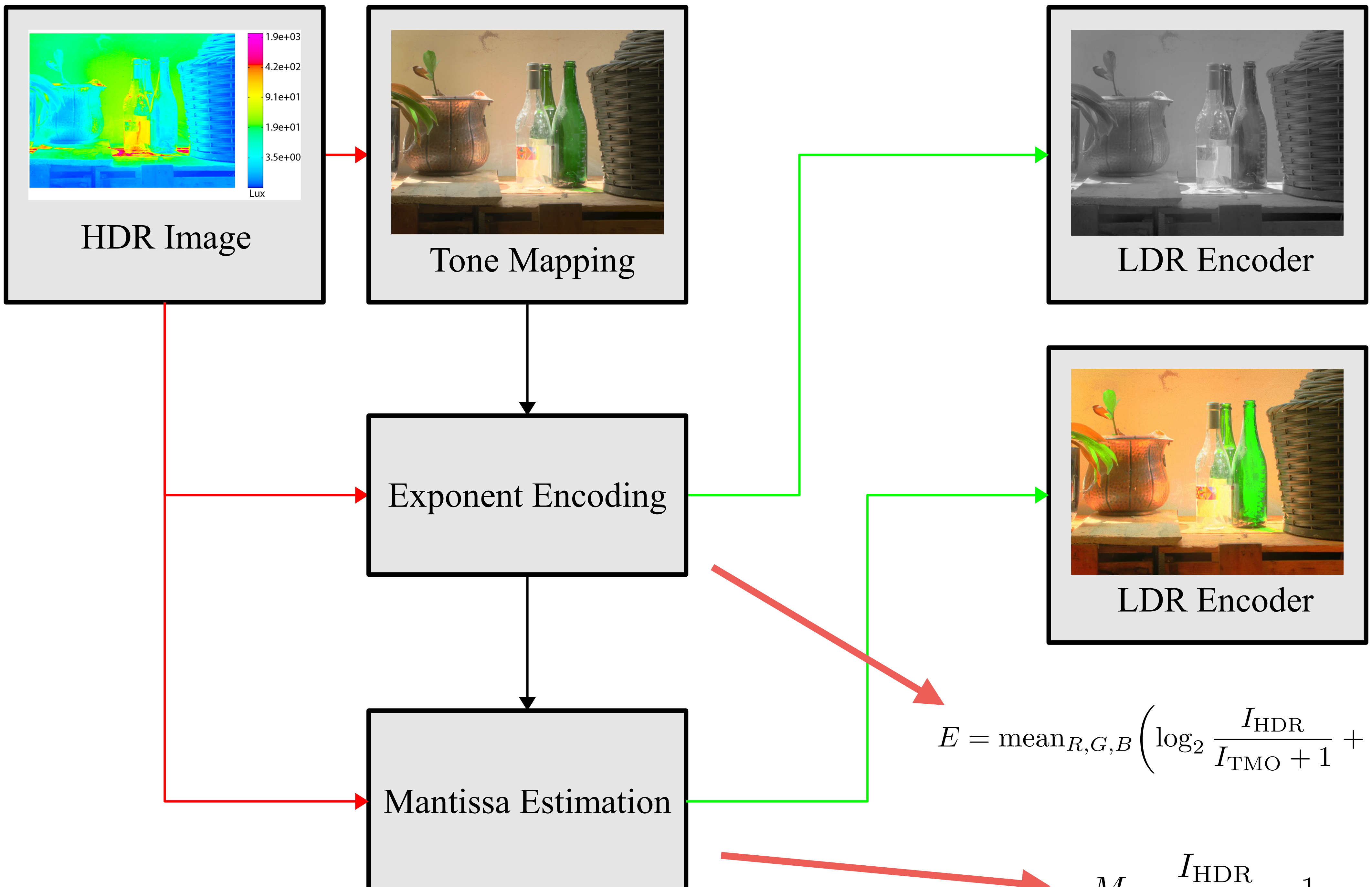


Spatially Varying RGBE



$$E = \text{mean}_{R,G,B}\left(\log_2 \frac{I_{\text{HDR}}}{I_{\text{TMO}}+1} + \epsilon\right) \quad \epsilon > 0$$

Spatially Varying RGBE



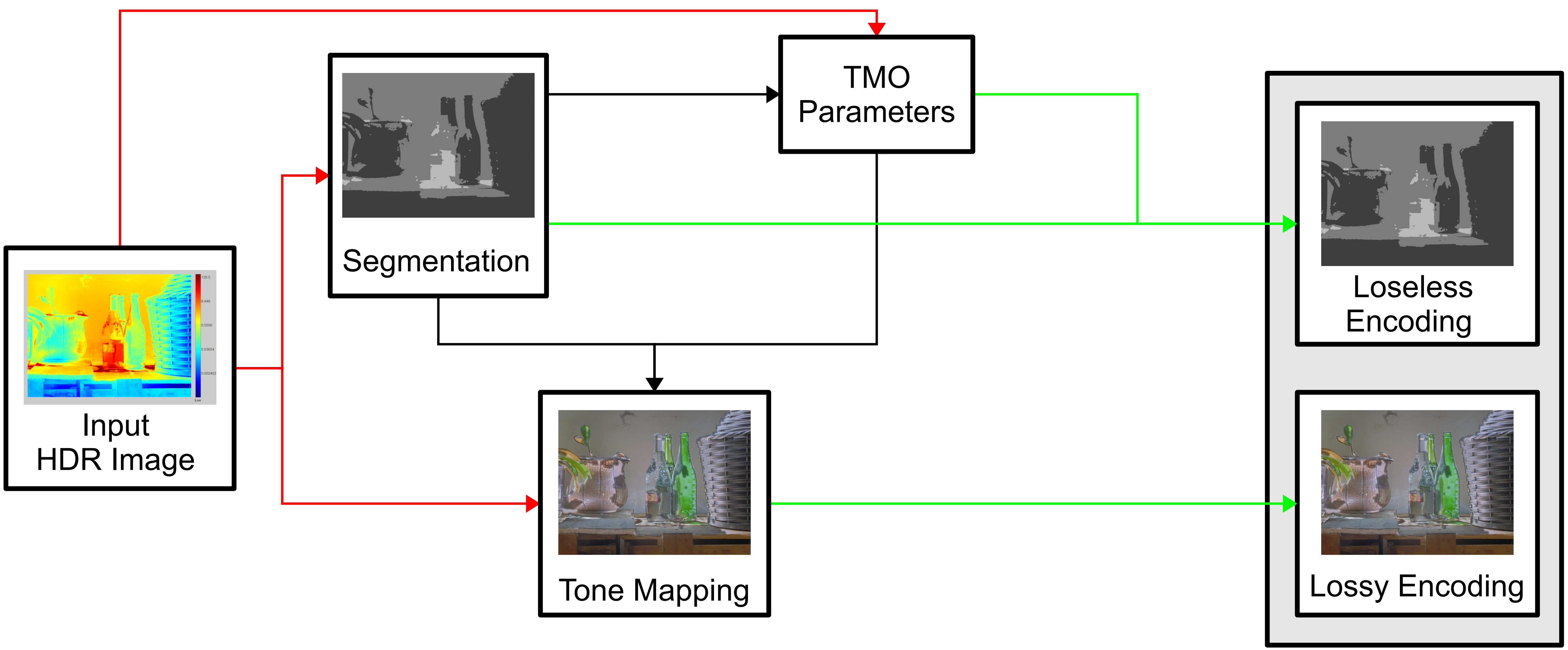
$$E = \text{mean}_{R,G,B}\left(\log_2 \frac{I_{\text{HDR}}}{I_{\text{TMO}}+1} + \epsilon\right) \quad \epsilon > 0$$

$$M = \frac{I_{\text{HDR}}}{E^E} - 1$$

BoostHDR

- **Idea**: to segment the image and to apply to each segment a linear compression factor [Banterle et al. 2012]
  - High efficiency
  - Semi backward compatible: the image looks a bit strange; i.e. seams and no global contrast
- Different encoders: JPEG, JPEG2000

BoostHDR



TMO Parameters
Segmentation
Input HDR Image
Tone Mapping
Loseless Encoding
Lossy Encoding

BoostHDR: semi backward compatible

Evaluation

- Perceptual metrics:
  - HDR-VDP
  - DRIIQM
- Objective metrics:
  - mPSNR
  - logRMSE

Evaluation: mPSNR

- **Issue**: classic PSNR definition do not work well because the peak can be an outlier

$$\mathrm{MSE}(I,\hat{I}) = \frac{1}{n}\sum_{j=1}^{n}\Big(I(\mathbf{x}_j) - \hat{I}(\mathbf{x}_j)\Big)^2$$

$$\mathrm{PSNR}(I,\hat{I}) = 10\log_{10}\left(\frac{I_{\max}^2}{\mathrm{MSE}(I,\hat{I})}\right)$$

- **Idea**: mean of PSNR values of all exposure images (LDR images) that can be extracted from an HDR image [Munkberg et al. 2006]

Evaluation: mPSNR

$$T(v,c) = \left[255(2^c v)^{\frac{1}{\gamma}} \right]_0^{255}$$

$$\mathrm{MSE}(I,\hat{I}) = \frac{1}{n \times p} \sum_{c=1}^{p} \sum_{i=1}^{n} \left(\Delta R_{i,c}^2 + \Delta G_{i,c}^2 + \Delta B_{i,c}^2 \right)$$

$$\mathrm{mPSNR}(I,\hat{I}) = 10 \log_{10} \left(\frac{3 \times 255^2}{MSE(I,\hat{I})} \right)$$

$$\Delta R_{i,c} = T(R(\mathbf{x}_i), c) - T(\hat{R}^*(\mathbf{x}_i), c)$$

$$\Delta G_{i,c} = T(G(\mathbf{x}_i), c) - T(\hat{G}^*(\mathbf{x}_i), c)$$

$$\Delta B_{i,c} = T(B(\mathbf{x}_i), c) - T(\hat{B}^*(\mathbf{x}_i), c)$$

Evaluation: logRMSE

- **Issues**: high values may have outliers and exacerbate per pixel differences
- **Idea**: apply logarithmic function to reduce high values influence

$$\mathrm{RMSE}(I, \hat{I}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\log_2 \frac{R(\mathbf{x}_i)}{\hat{R}(\mathbf{x}_i)}\right)^2 + \left(\log_2 \frac{G(\mathbf{x}_i)}{\hat{G}(\mathbf{x}_i)}\right)^2 + \left(\log_2 \frac{B(\mathbf{x}_i)}{\hat{B}(\mathbf{x}_i)}\right)^2}$$

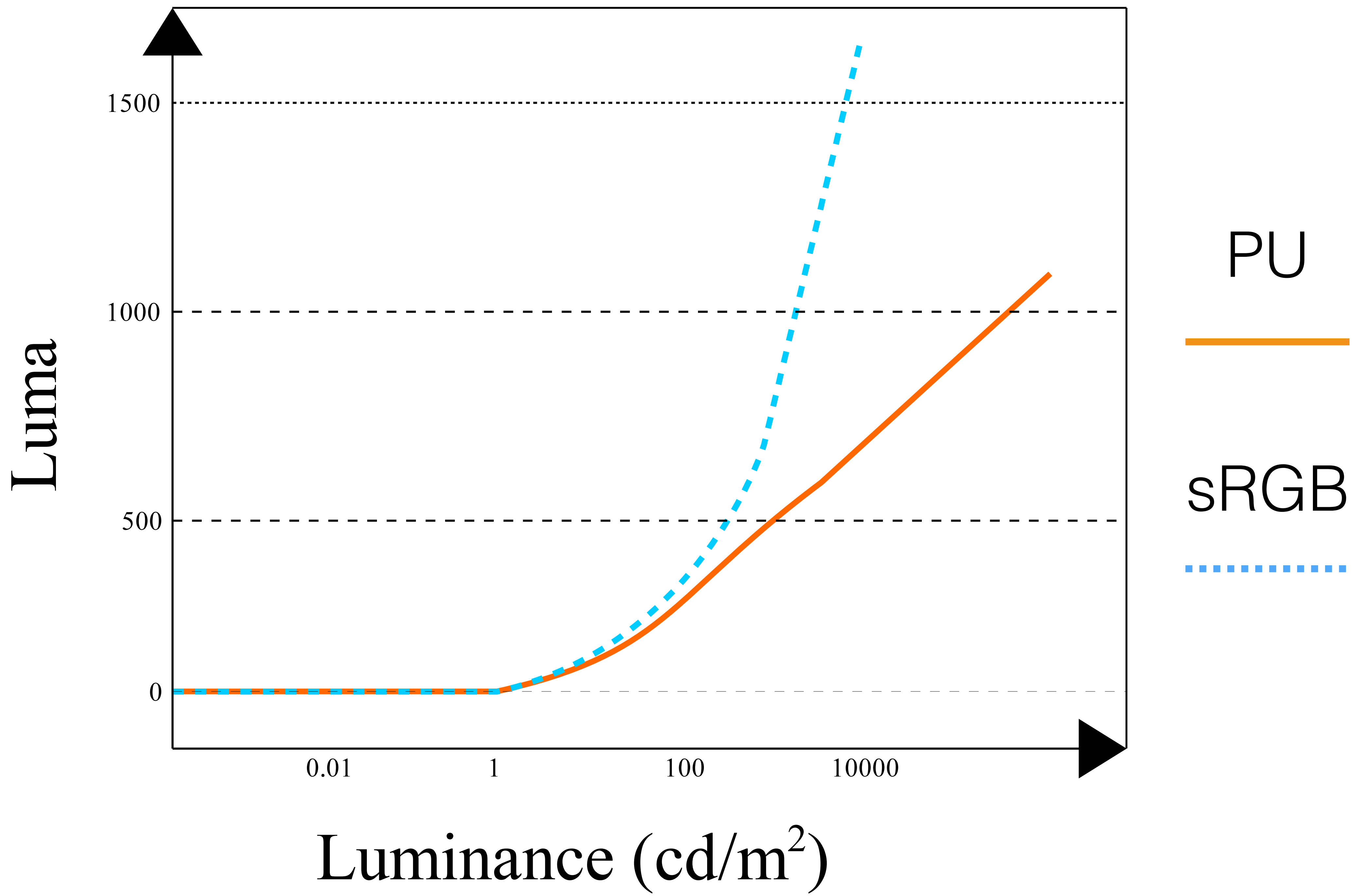
Evaluation: PU Encoding

- **Idea**: to reuse existing objective metrics. [Aydin et al. 2008]
    - CRT monitors (gamma): range [0.1, 80] cd/m^2
    - LCD monitors (gamma): peak 500 cd/m^2
    - HDR monitors (mostly linear): peak 4,000 cd/m^2

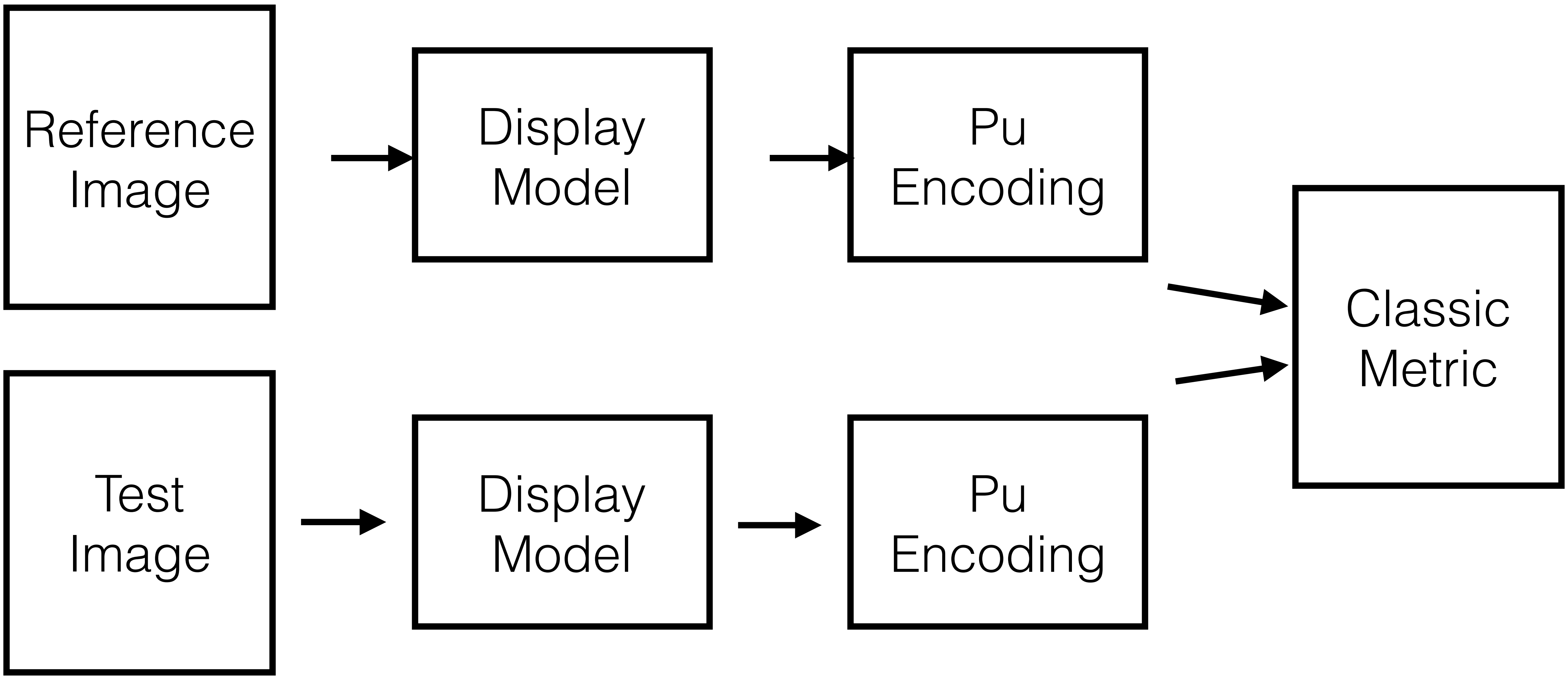
Evaluation: PU Encoding

- PU encoding is a non-linear curve which simulates the response of the HVS to luminance values
- Similar behavior of sRGB in [0.1, 80] cd/m^2

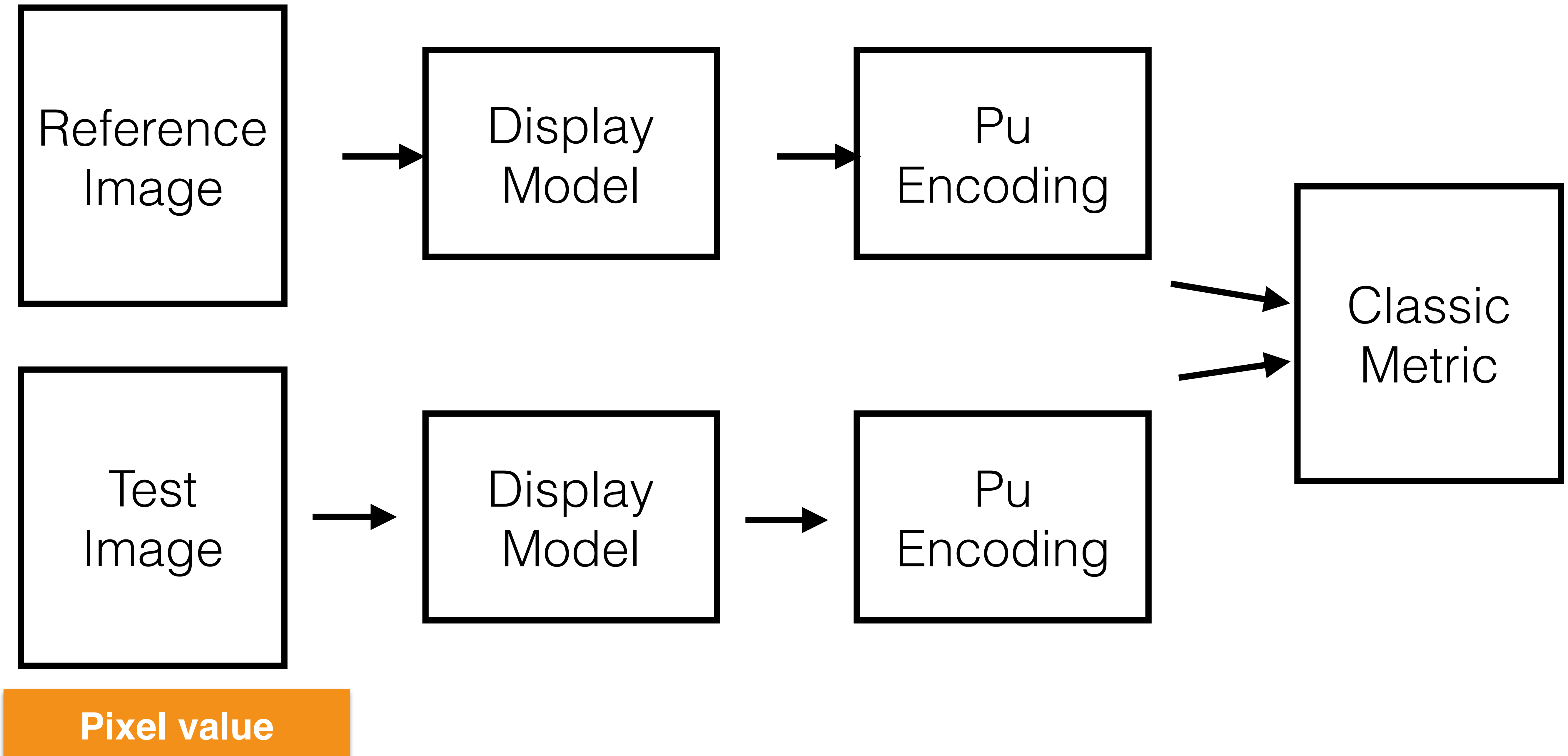
Evaluation: PU Encoding



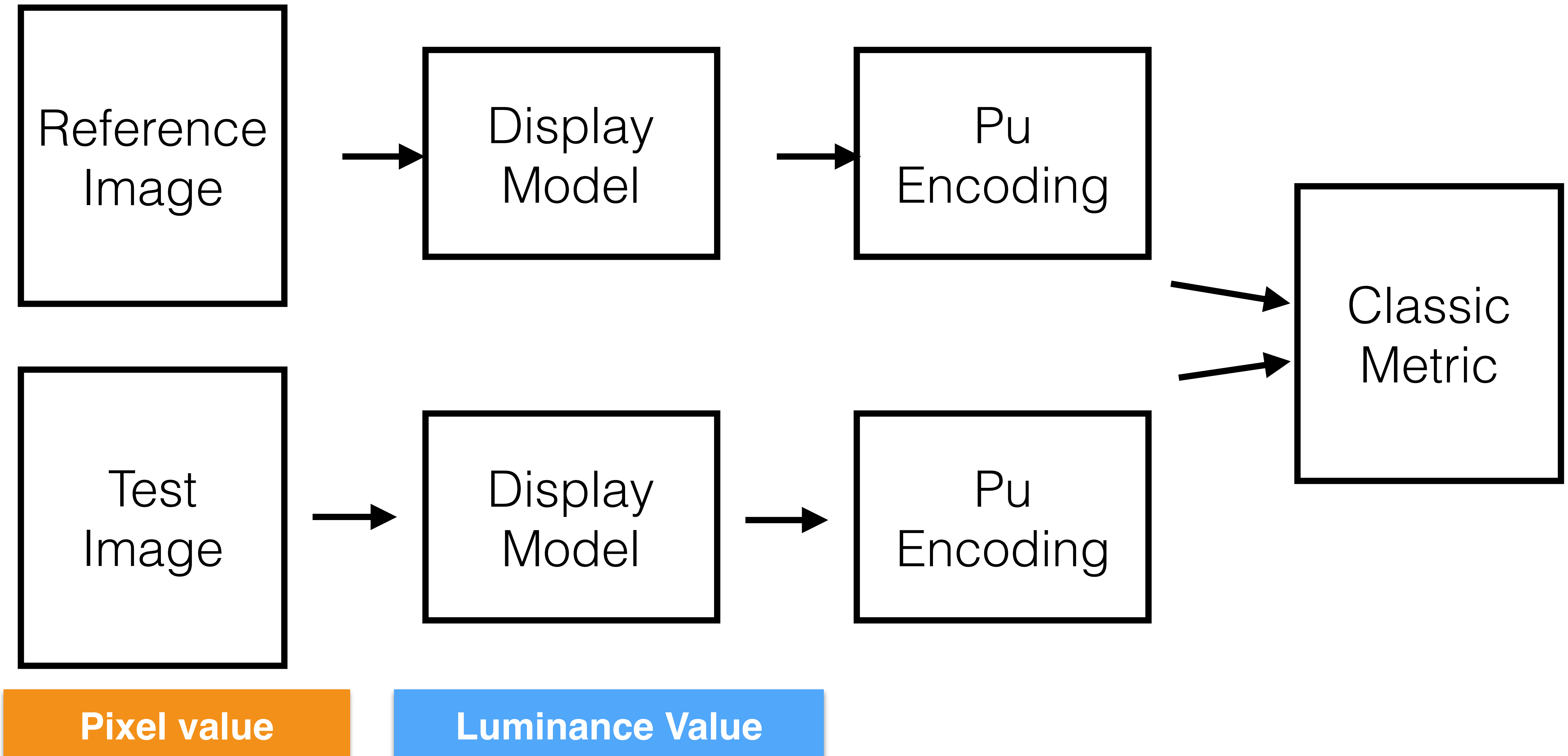
Evaluation: PU Encoding



Evaluation: PU Encoding



Evaluation: PU Encoding



the present…

Standardization: JPEG-XR

- A JPEG standard
- It is not backward compatible
- Proposed by Microsoft (it is the old PhotoHD format)
- Add support for:
    - 48bit integer RGB
    - 16-bit/32-bit floating point per color channel

Standardization: JPEG-XR

- It supports RGBE encoding
- Loseless UYV color encoding
- Hierarchical transform (2 layers): 4x4 and 16x16
- Official website:
  - http://www.jpeg.org/jpegxr/index.html

Standardization: JPEG-XT

- It is an ISO standard extension of JPEG (ISO/IEC 10918-1)
- Backward compatible with JPEG
- Three compression profiles: A, B, and C
- Capability to encode HDR images
- Official website:
    - http://www.jpeg.org/jpegxt/index.html

let’s talk about videos…

LDR Video Compression

- Existing video standard: MPEG-1 (H.261), MPEG-2 (H.262), MPEG-4 Part 2 (H.623), H.264 (AVC), H. 265 (HEVC)

- How do they work?

LDR Compression: I-Frames

- They are reference frames which are basically encoded using JPEG
- Also called anchor frame

LDR Compression: P-Frames

- They are predicted frame:
  - exploitation of temporal redundancy
  - It stores differences between the frame to be encoded and the I-frame
  - How? By using motion vector:
    - motion compensation!

LDR Compression: P-Frames

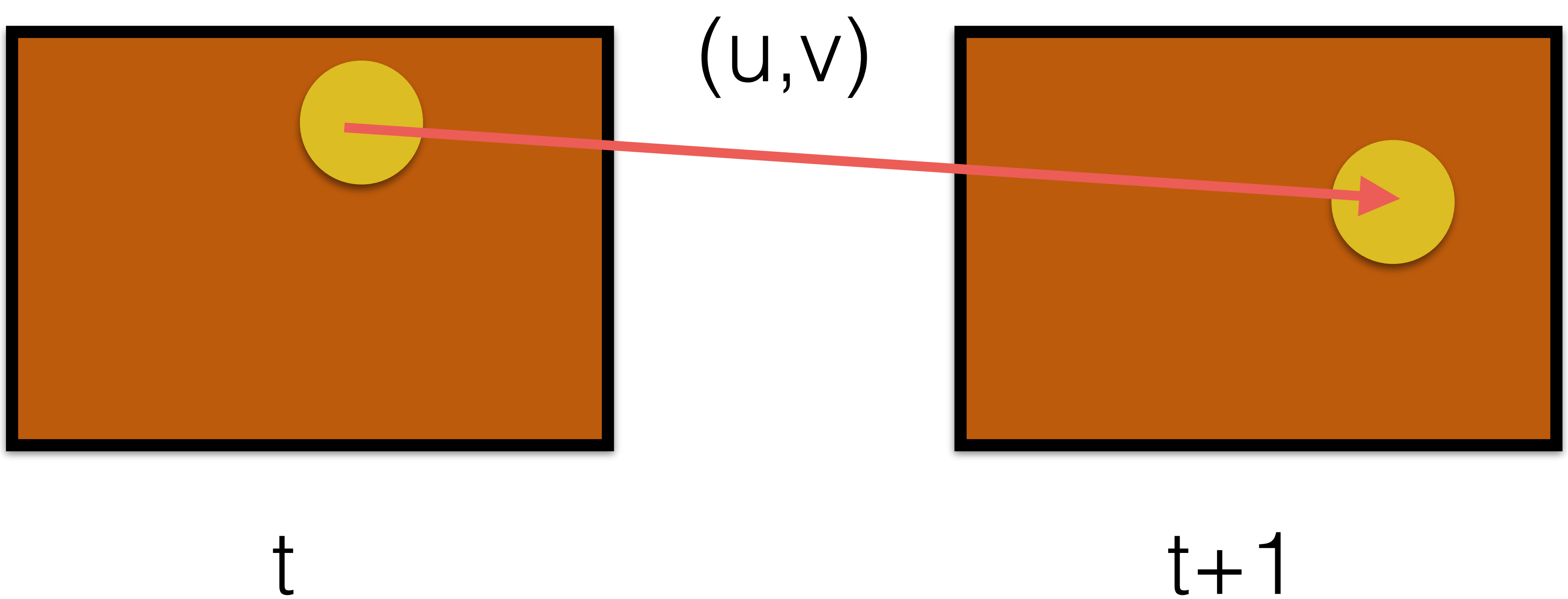
t

t+1

LDR Compression: P-Frames

Difference frame time t and t+1

LDR Compression: Motion Estimation

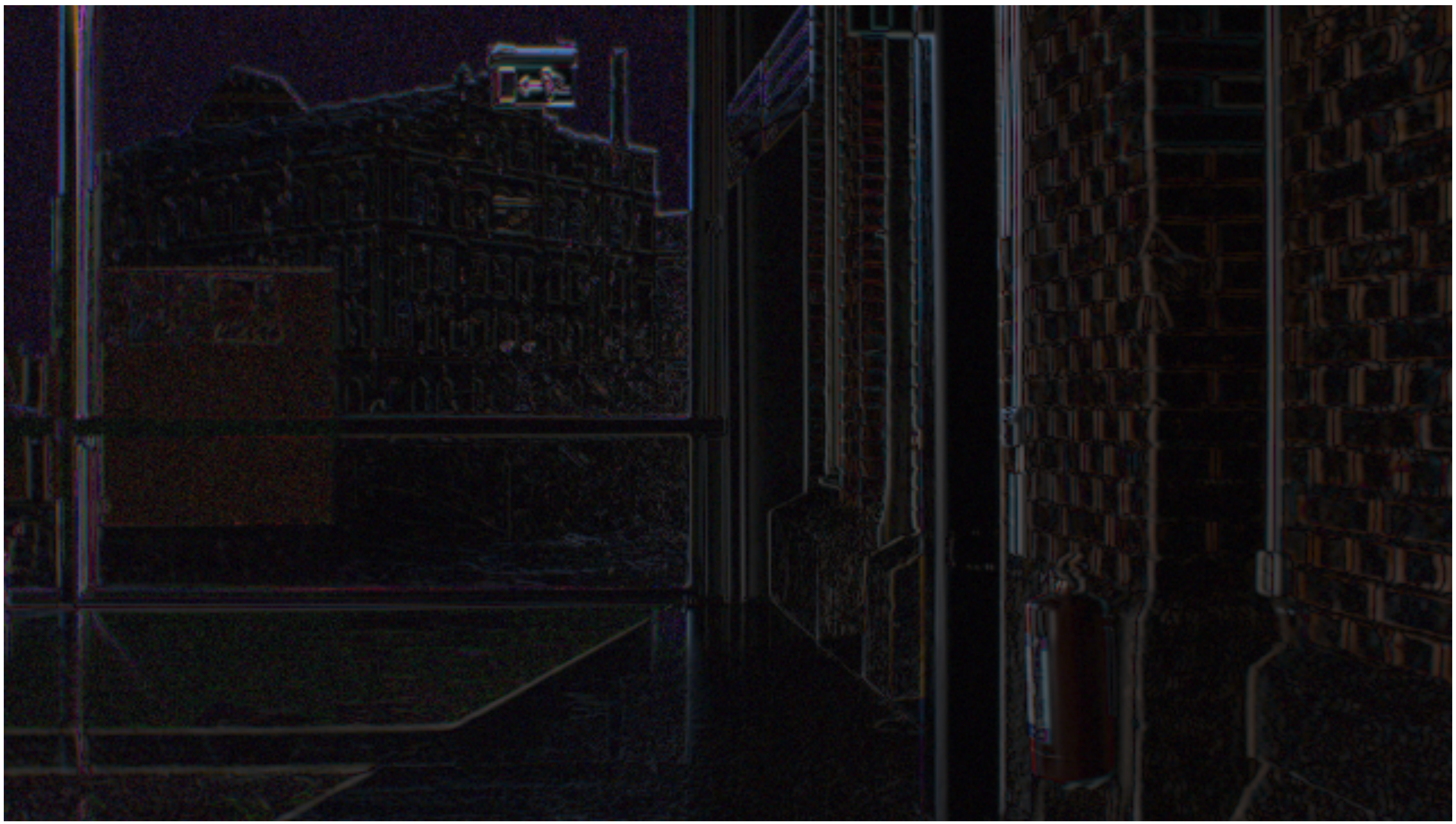


stored per macroblock (16x16)

LDR Compression: Motion Estimation

Difference frame time t and t+1 with motion estimation

LDR Compression: Motion Estimation



Difference frame time t and t+1 with motion estimation

LDR Compression: Conclusions

- There are more other mechanisms such as:
  - B-frames
  - Adaptive macroblocks
  - etc…

and now back to HDR videos…

HDRV

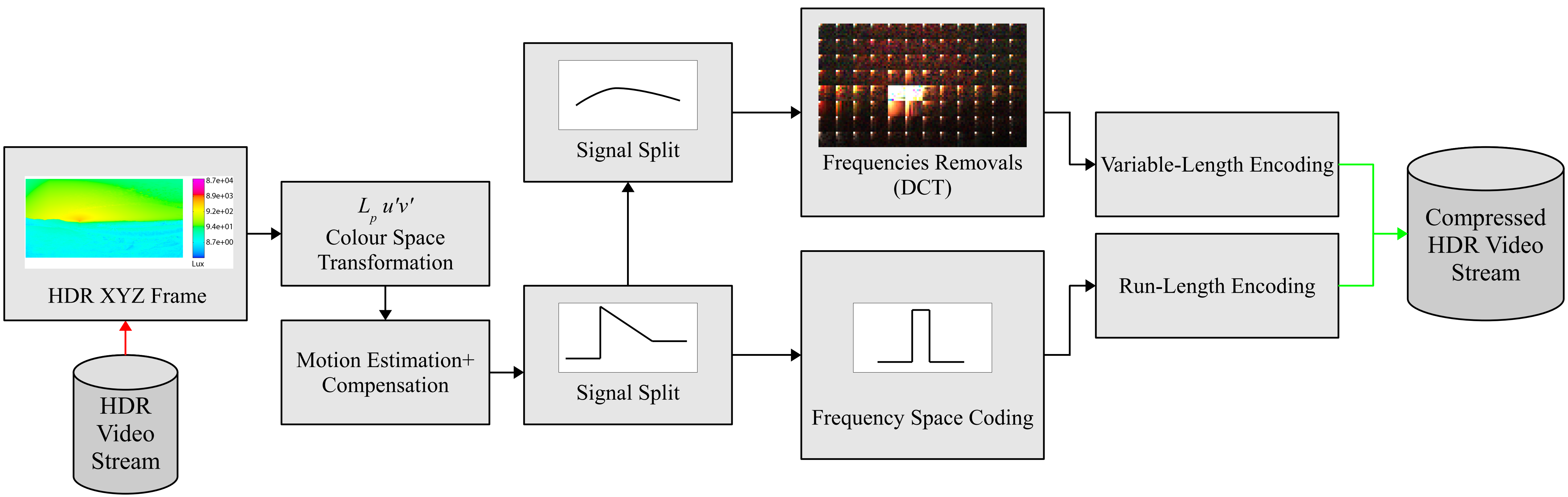
- **Idea**: to extend MPEG for handling HDR [Mantiuk et al. 2004]
- Steps:
  - Applying a compression function perceptually based (11bit for encoding luma):

$$\frac{d\Psi(l)}{dl} = 2\frac{tvi(\Psi(l))}{f}$$

$$\Psi(0) = 10^{-4}\mathrm{cd/m}^2 \qquad \Psi(l_{\max}) = 10^{8}\mathrm{cd/m}^2 \qquad l_{\max} = 2^{\mathrm{nbits}} - 1$$

  - Modifying the DCT to handle hard case: patches with strong edges

HDRV Compression



MPEG-HDR

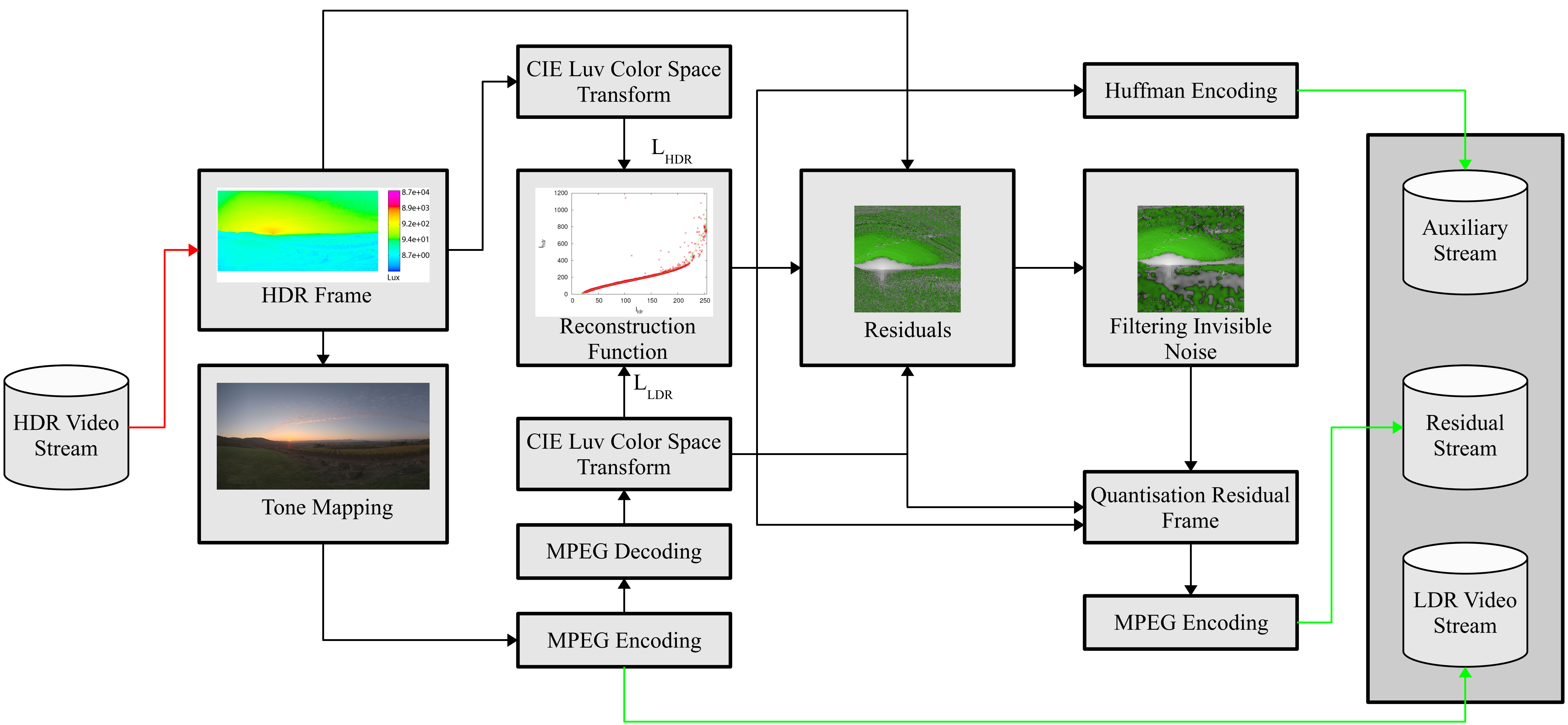
- **Idea**: HDRV works great, but it is not drawback compatible, why not using a similar idea from JPEG-HDR? [Mantiuk et al. 2006]

- Steps:

    - Tone map the image; any TMO —> this impacts on size of the stream

    - Compute residuals filtering what the HVS cannot perceive!

MPEG-HDR


HDR Video Stream
HDR Frame
8.7e+04
8.9e+03
9.2e+02
9.4e+01
8.7e+00
Lux
Tone Mapping
CIE Luv Color Space Transform
L HDR
Reconstruction Function
L LDR
CIE Luv Color Space Transform
MPEG Decoding
MPEG Encoding
Residuals
Huffman Encoding
Filtering Invisible Noise
Quantisation Residual Frame
MPEG Encoding
Auxiliary Stream
Residual Stream
LDR Video Stream

Temporal Gradient Compression

- **Idea**: to exploit a temporal TMO and residual filtered using the bilateral filter [Lee and Kim 2008]

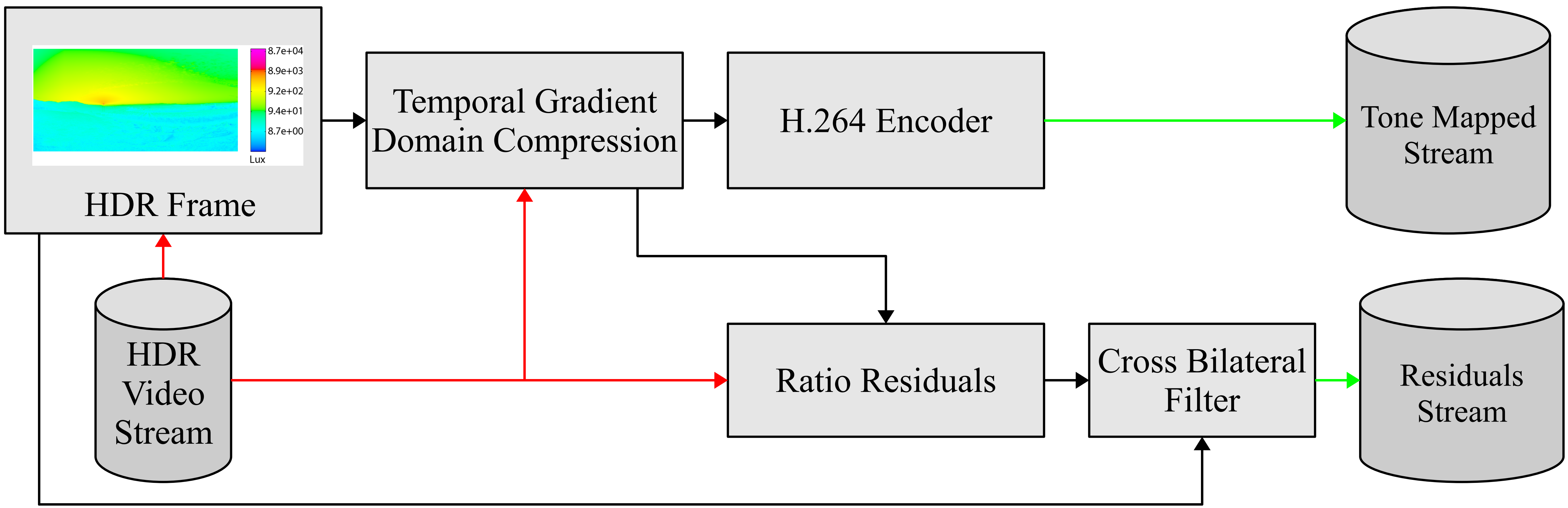
- Residuals:

$$R(\mathbf{x}) = \log_2\left(\frac{L_w(\mathbf{x})}{L_d(\mathbf{x})}\right)$$

- Introduction of a rate for controlling size of the residuals stream

$$QP_{\mathrm{ratio}} = 0.77QP_d + 13.42$$

Temporal Gradient Compression



Current research

- Backward compatible methods
  - Optimized TMOs for encoding
  - Best exposure method

the present…

Standardization

- MPEG has started standardization for HDR video content
- Proposals were submitted
  - long process; a couple of years

Questions?