Compressed Texture Transmission Format

Mark Callow

Siggraph, Vancouver, Aug 15&16, 2018

Follow Along

Google Docs version of this talk: http://bit.ly/cttf20180815.

KHRONOS GROUP

Required Specifications

- Format(s) for the image bits
- Container
  - Textures usually consist of
  -

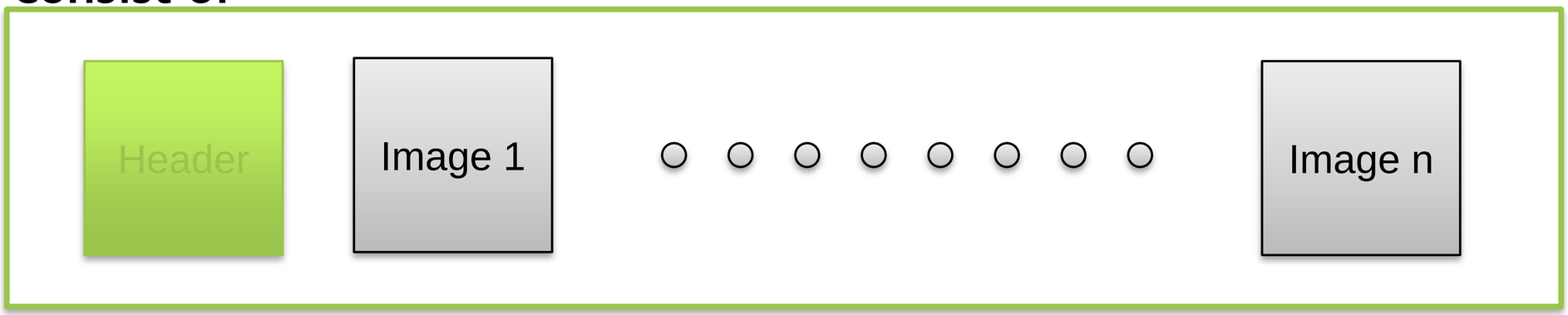


Image Bits - Issues

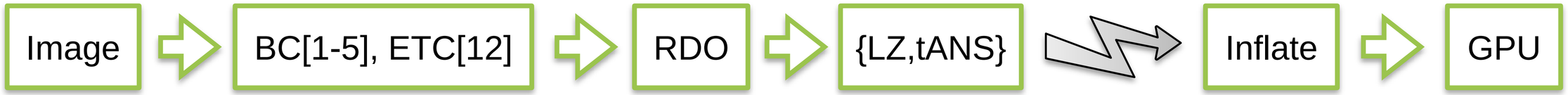
- Can use image formats defined by the GPU APIs but
  - uncompressed formats too large for transmission
  - block-compressed formats suboptimal for transmission
- JPEG-compressed files can’t be randomly accessed
- Compression to GPU formats unavailable on most clients and slow
- A googol[1] of GPU/Platform-Specific formats. Nightmare!

1. 10^{100}

Image Bits - Solving Transmission Size

Supercompression

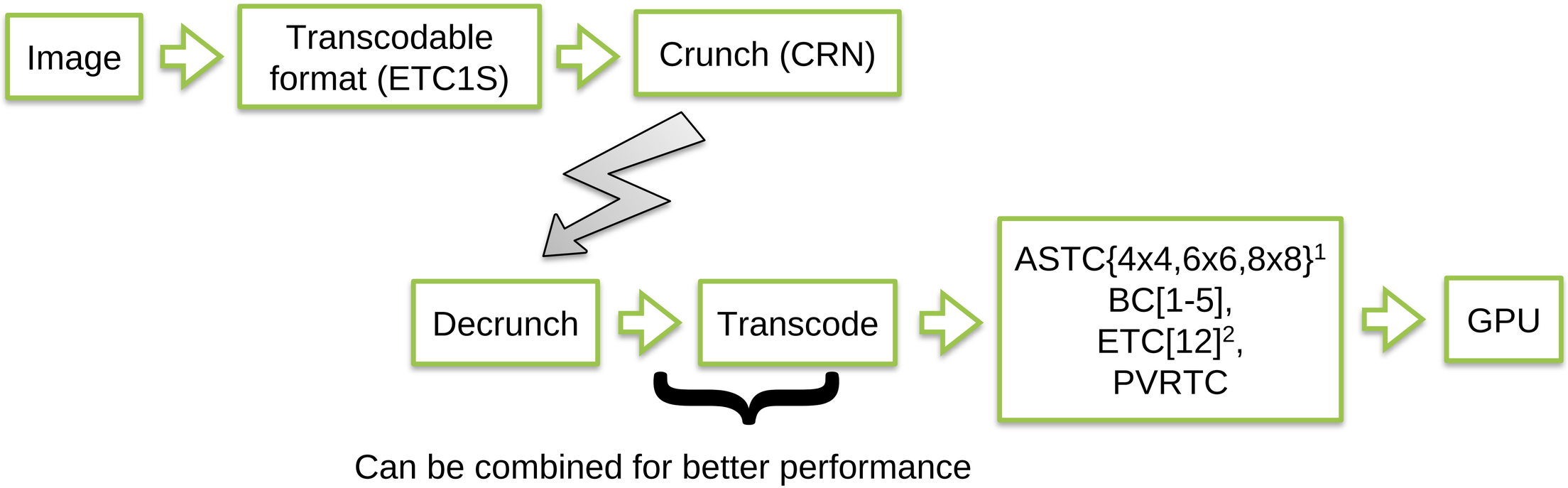
Rate Distortion Optimization (Crunch RDO Mode)



Crunch (Crunch CRN mode)

Image Bits - Solving the Googol of formats

Universal Transcodable Format

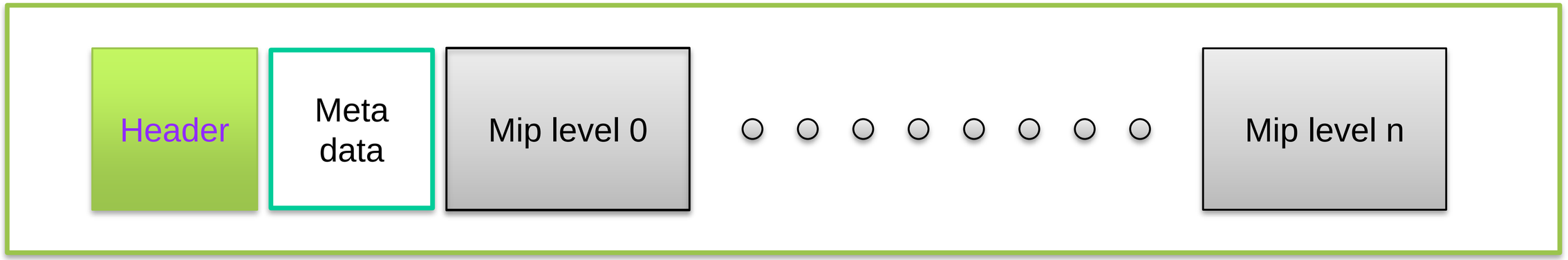


1. Expected. Code not yet in repo.
2. No transcode necessary

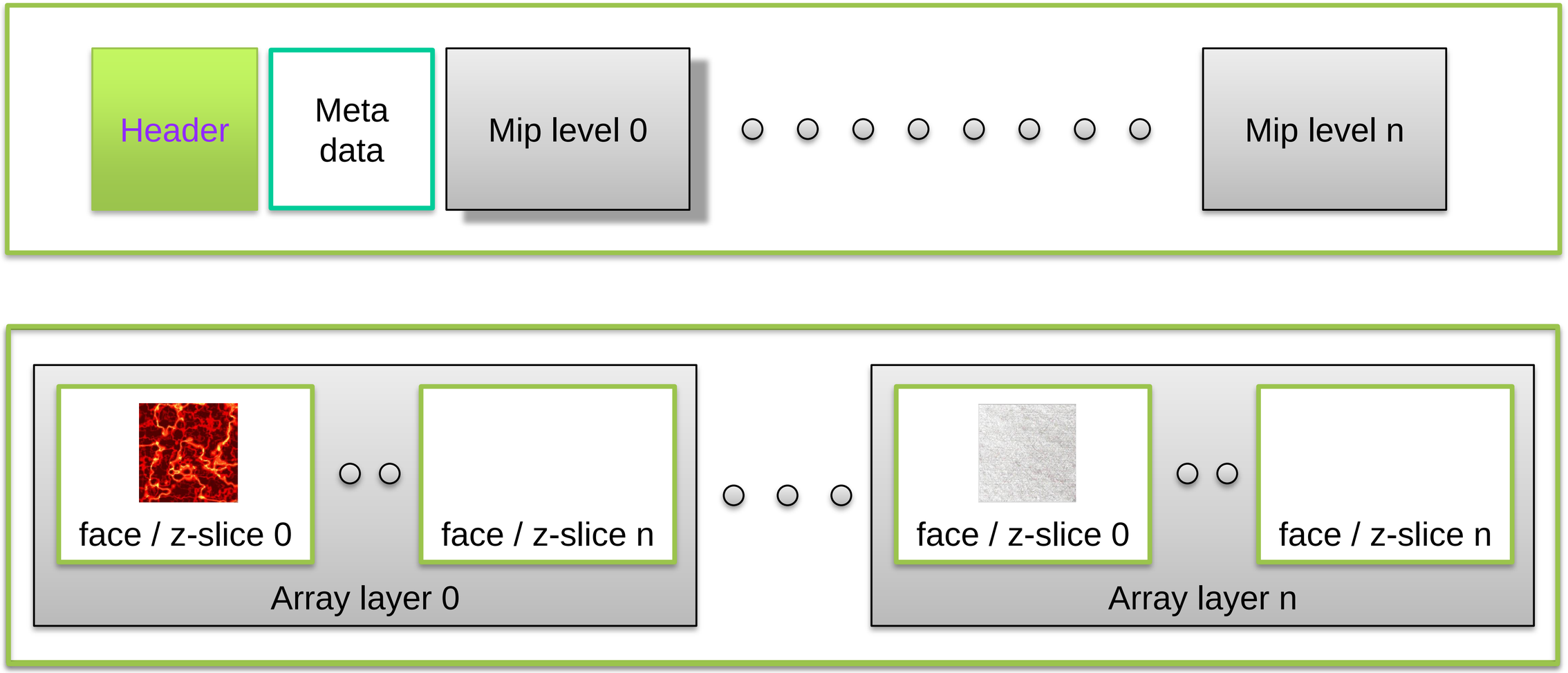
Quality

- Will show comparison tests later.
- ETC1S-CRN works well for natural images.
- Not so well for line drawings, icons etc.
  - LZ compression often works well in these cases
    - but will be uncompressed in GPU
- Will add support for variable block size and hi-precision formats: ASTC, BC6H/BC7, ETC2 in an incremental revision.

KTX File Structure

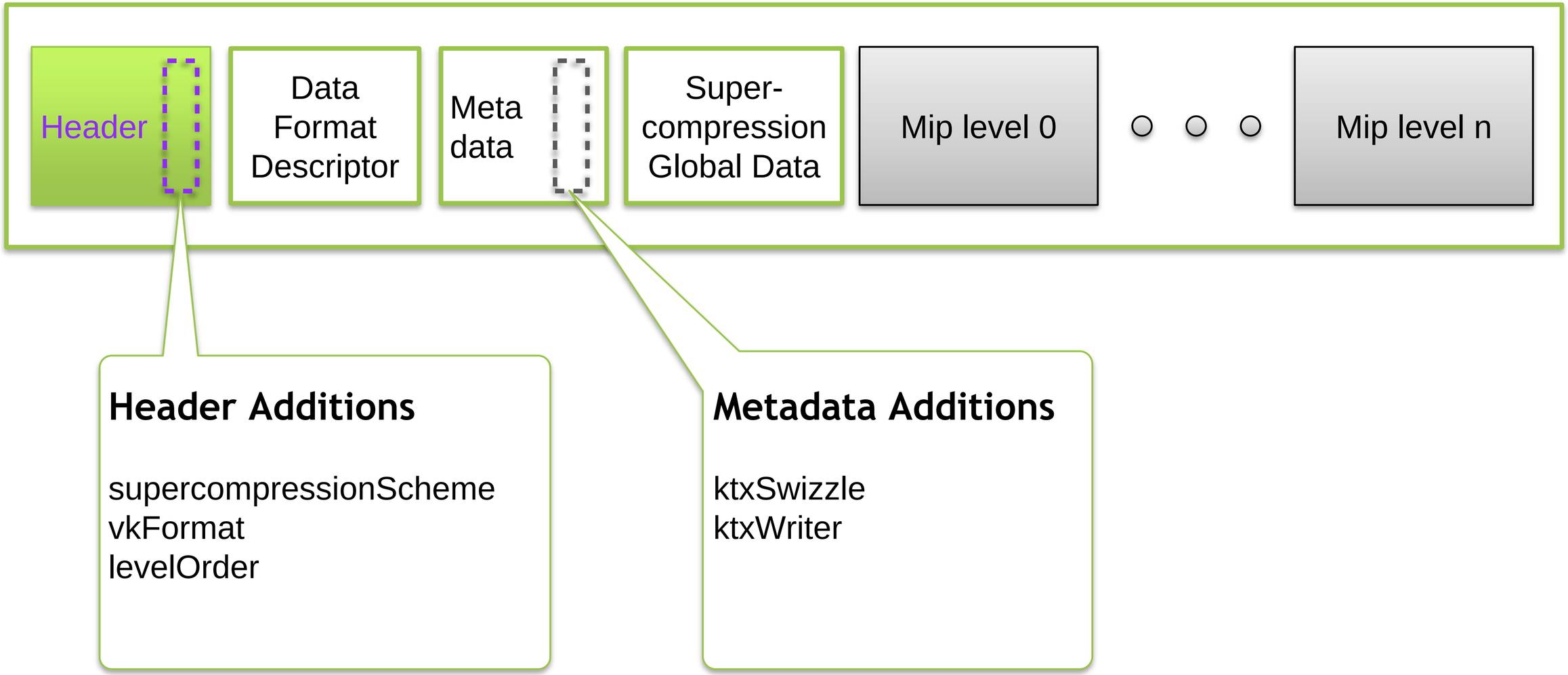


KTX File Structure



Mip Level Structure

KTX2 File Structure



KTX2 Header Additions: Supercompression

- supercompressionScheme
  - Deflate (a.k.a zlib)
  - Zstandard
  - Crunch CRN mode
  - LZ4 - under discussion
  - tANS - under discussion

  - Except for CRN, use these only with Crunch RDO-mode-processed or uncompressed images.
  - Use CRN for supercompression of block-compressed textures.
  - Transcodable format is supercompressed using CRN.

KTX 2 Header Additions: Others

- vkFormat field
  - makes loading of Vulkan textures easier
- levelOrder field
  - lets mip levels be ordered from smallest first, enabling streaming

Data Format Descriptor[1]

- Exact description of texel format and color space
  - non-OpenGL and non-Vulkan applications can use the image data without understanding OpenGL or Vulkan enums
  - can save & load multi-sample images to/from KTX files
  - applications that care can do color correction

1. See https://www.khronos.org/dataformat.

DFD Color Space Information

- Enums are provided to identify
  - common primaries
    - BT709/sRGB, BT2020, Display P3, Adobe RGB & more.
  - common transfer functions
    - Linear[1], sRGB[1], ITU (BT 601, 709 & 2020), Adobe RGB & more
- A full ICC profile could be incorporated via a KDFS extension

1. D3D/OpenGL/Vulkan/WebGL support only Linear & sRGB in hardware

Metadata Additions

- KTXswizzle
  - Indicates desired component mapping for a texture
- KTXwriter
  - File writer should identify itself with this

Open KTX2 Issues

- Issues list at end of specification:

  http://github.khronos.org/KTX-Specification/
- Please look. I want your opinion.
- File suggestions and new issues at

  https://github.com/KhronosGroup/KTX-Specification

Status

- KHR_texture_transmission v1.0 Provisional spec in Q4
  - Universal supercompressed transcodable format
    - Target device transcodes to its own native GPU format in real-time
  - 'RDO mode' - allows target bitrate to be set (compression ratio vs quality)
  - Only lower precision formats supported for now [BC1-5, ETC1, PVRTC, ASTC (LDR subset)]

- Incremental spec revisions
  - Universal format for precision encoding and high dynamic range
    - Will support variable block size and hi-precision formats: ASTC, BC6H/BC7, HDR
  - New minor revisions will not introduce breaking changes

- Call for Industry Collaboration!
  - Actively seeking industry feedback and contributions
  - Development is open to the community - github links to specs, libs & samples to follow
  - Consider participating directly in the Khronos 3D Formats WG!

Reference DXT1 vs. CTTF Comparison Tests

- Quality comparisons of a reference DXT1 image vs equivalent CTTF encodings

- Reference DXT1 image created with Crunch encoder
- CTTF bitrates set to best match reference DXT1 quality
- Test 1: Reference DXT1 vs CTTF ETC1S-CRN transcoder
- Test 2: Reference DXT1 vs CTTF RDO+LZMA
- Test 3: CTTF mode comparison: ETC1S-CRN vs RDO+LZMA

KHRONOS GROUP

DXT1 vs. CTTF ETC1S-CRN Transcoder

DXT1

ETC1S-CRN -> DXT1

Format	DXT1	CTTF ETC1S-CRN
Transmitted bits/pixel	4.0	1.8*
Stream Compression	6:1	13:1
PSNR (dB)	37.78	36.32
MSE[1]	10.83	15.19
SSIM[2]	0.97	0.96

*Best bitrate setting to approx ref DXT1 PSNR

[1] Mean Squared Error

[2] Structured Similarity Index

DXT1 vs. CTTF RDO+LZMA (DXT1)

DXT1

Unpacked RDO DXT1

Format	DXT1	CTTF RDO+LZMA
Transmitted bits/pixel	4.0	2.1*
Stream Compression	6:1	11:1
PSNR (dB)	37.78	36.45
MSE	10.83	14.71
SSIM	0.97	0.95

*Best bitrate setting to approx ref DXT1 PSNR

CTTF Mode Comparison- ETC1S-CRN vs.RDO+LZMA (DXT1)

ETC1S-CRN -> DXT1

Unpacked RDO DXT1

Format	CTTF ETC1S-CRN	CTTF RDO+LZMA
Transmitted bits/pixel	1.8	2.1
Stream Compression	13:1	11:1
PSNR (dB)	36.32	36.45
MSE	15.19	14.71
SSIM	0.96	0.95

CTTF RDO mode + LZMA does not give as high compression perf as ETC1S-CRN. However, given that it is just an optimizing DXT encoder, the resulting texture can be used by existing workflows without modification

ETC1S-CRN has better compression with no extra encoder stage needed, but requires work to integrate universal format support into existing workflows

Acknowledgements

David Wilkinson, AMD	Texture Transmission TSG Chairman. Comparison Image Producer.
Rich Geldreich, Binomial	Core Technology Creator. Masochist.* 😉 * How else to describe someone with the willingness & stamina to wade through *enormously long* compressed texture format specs looking for commonalities? 😵
Alexander Suvorov, Unity	Crunch support for ETC{,2} & performance improvements.
Andrew Garrard, Samsung	Khronos Data Format Specification Guru.
Alexey Knyazev, Independent	glTF Integration Expert.
Patrick Cozzi, Cesium	3D Formats WG Chairman.

Watch these places for progress

Crunch & Transcoders: https://github.com/KhronosGroup/glTF-Texture-Transmission-Tools.

Texture Compression GUI: https://github.com/KhronosGroup/glTF-Compressonator.

Samples & Test Data: https://github.com/KhronosGroup/glTF-Texture-Transmission-Samples.

Readable KTX2 specification: https://github.khronos.org/KTX-Specification/.

KTX specification source: https://github.com/KhronosGroup/KTX-Specification.

KTX software (currently only supports KTX1): https://github.com/KhronosGroup/KTX-Software.

Original Crunch GitHub Repo: https://github.com/BinomialLLC/crunch. Use the fork above.

Blogs

Alexander Suvorov on his improvements to Crunch to support ETC:
https://blogs.unity3d.com/2017/12/15/crunch-compression-of-etc-textures/

Rich Geldreich on the “universal format” and the transcoders
http://richg42.blogspot.com/2018/06/etc1s-texture-format-encoding.html
http://richg42.blogspot.com/2018/05/some-basis-baseline-universal-format.html

Supplemental Material

DXT1 vs. CTTF ETC1S-CRN Transcoding

- Quality and Risk Analysis

  - Note how the ETC1S-CRN transcodable format compresses better than an RDO+LZMA DXT
  - Highly compact - potentially up to 4x smaller than equivalent GPU format encoding
  - Supercompression is gained through a set of entropy reduction stages + clusterization +VQ
  - ETC1S-CRN also features an implicit RDO stage - so quality vs bitrate can be controlled
  - Supporting ETC1S-CRN requires changes to existing asset pipelines and app code - some risk
  - Unlike CTTF RDO mode, where the optimized DXT textures can be used as-is - no risk

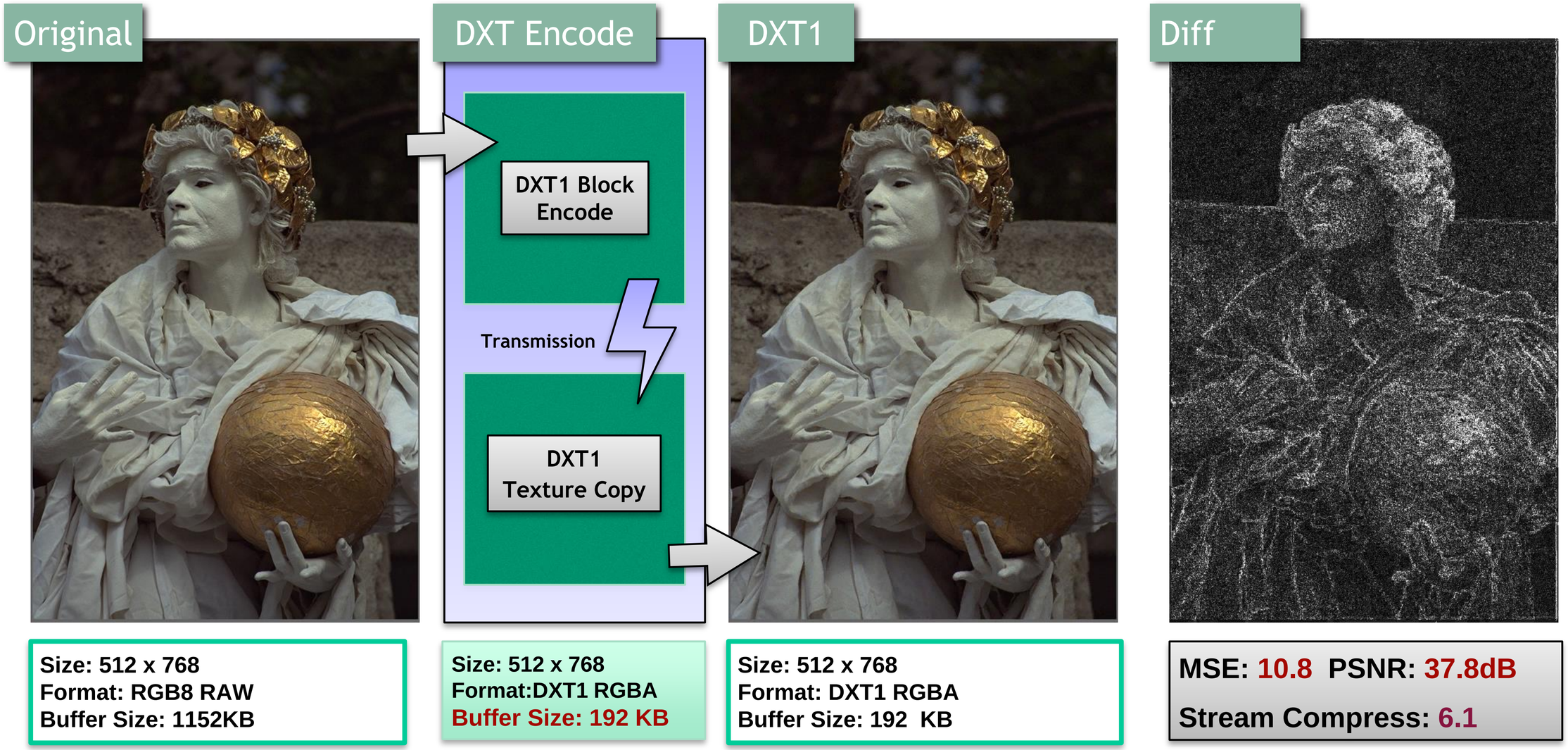
DXT1 vs. CTTF RDO+LZMA mode

- Quality and Risk Analysis

  - Note performance slides showed RDO+LZMA mode performing WORSE than ETC1S-CRN!
  - ETC1S-CRN applies supercompression on encode, so no need for a lossless encoding stage
  - RDO mode adjusts bitrate of encoded DXT to desired level, for use with a lossless encoder
  - Works with existing workflows - no changes to asset pipelines or app code, zero risk
  - Unlike the ETC1S-CRN universal format, which would require integration into tools and apps

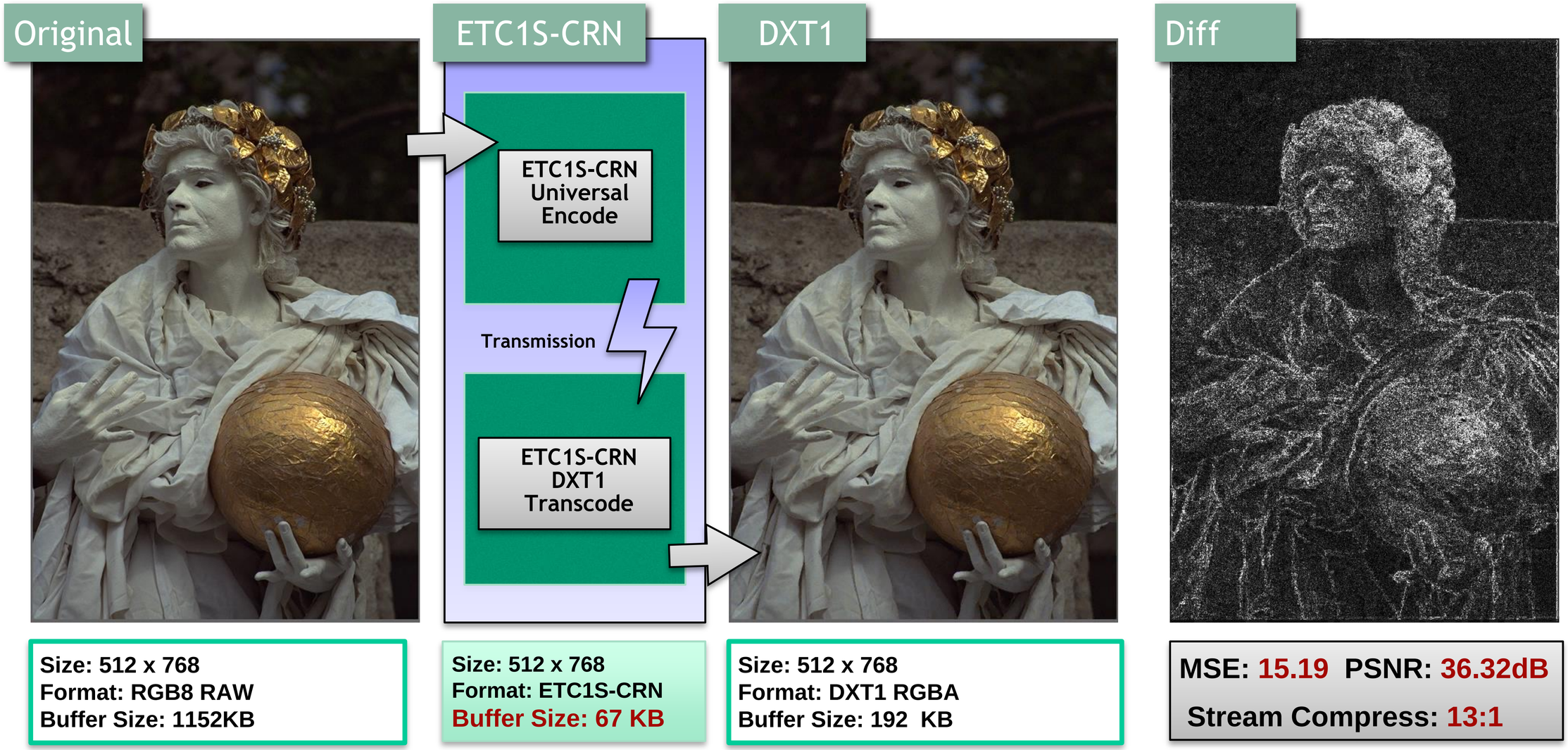
Bitstream Compression Performance

Transmission with standard DXT1 Encoding



Bitstream Compression Performance

CTTF ETC1S-CRN bitstream with DXT1 transcode on receiver



Bitstream Compression Performance

CTTF RDO+LZMA lossless DXT1 bitstream with unpack on receiver