ASTC: The Future of Texture Compression

Tom Olson

Director, Graphics Research - ARM, Ltd

Chair, OpenGL® ES™ Working Group - Khronos

Outline

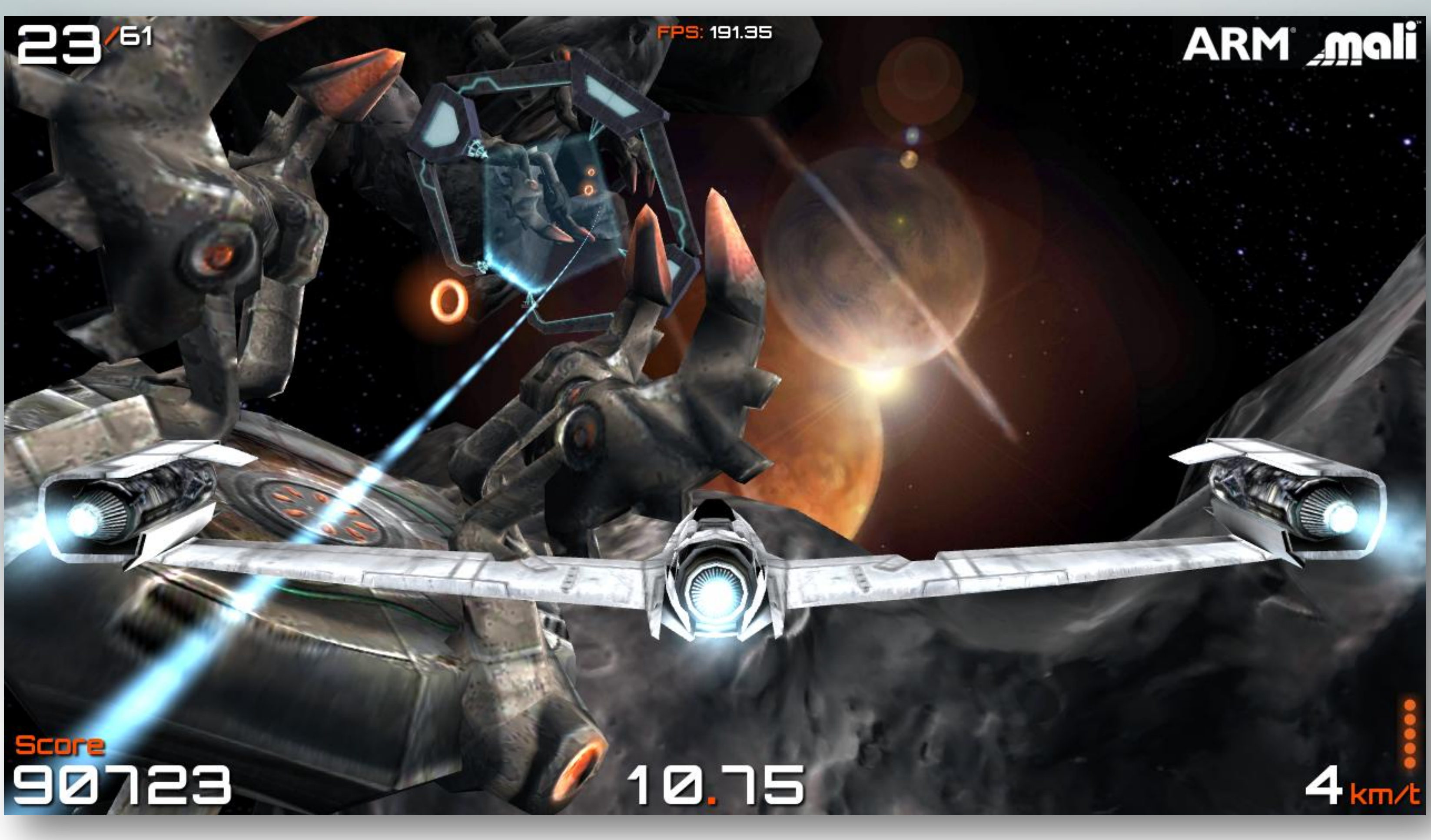
Why texture compression matters

Texture compression today

Introducing ASTC

- Features
- Quality
- Access

Graphics: It's all about the textures



But there’s a problem…

ARM “Timbuktu” demo

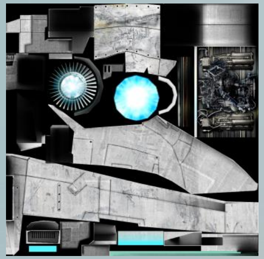
Textures are big

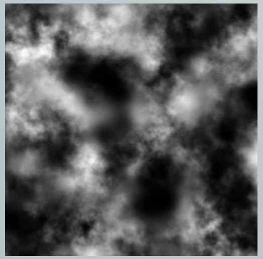
- Memory footprint and bandwidth
- Performance and power

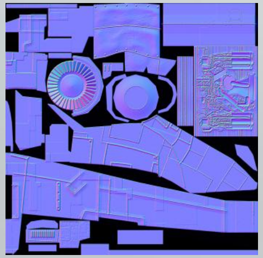
We need texture compression!

What kind of compression system do we need?

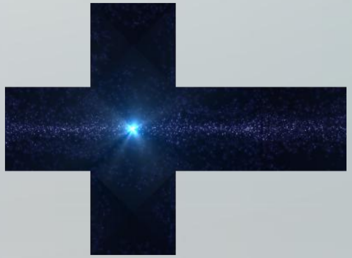
Textures are used for many different things:

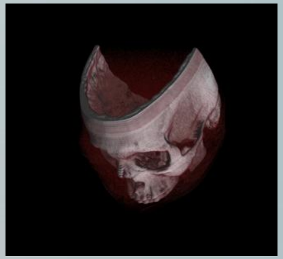

Reflectance


Gloss, Height, etc


Normals

Illuminance


Lighting environment

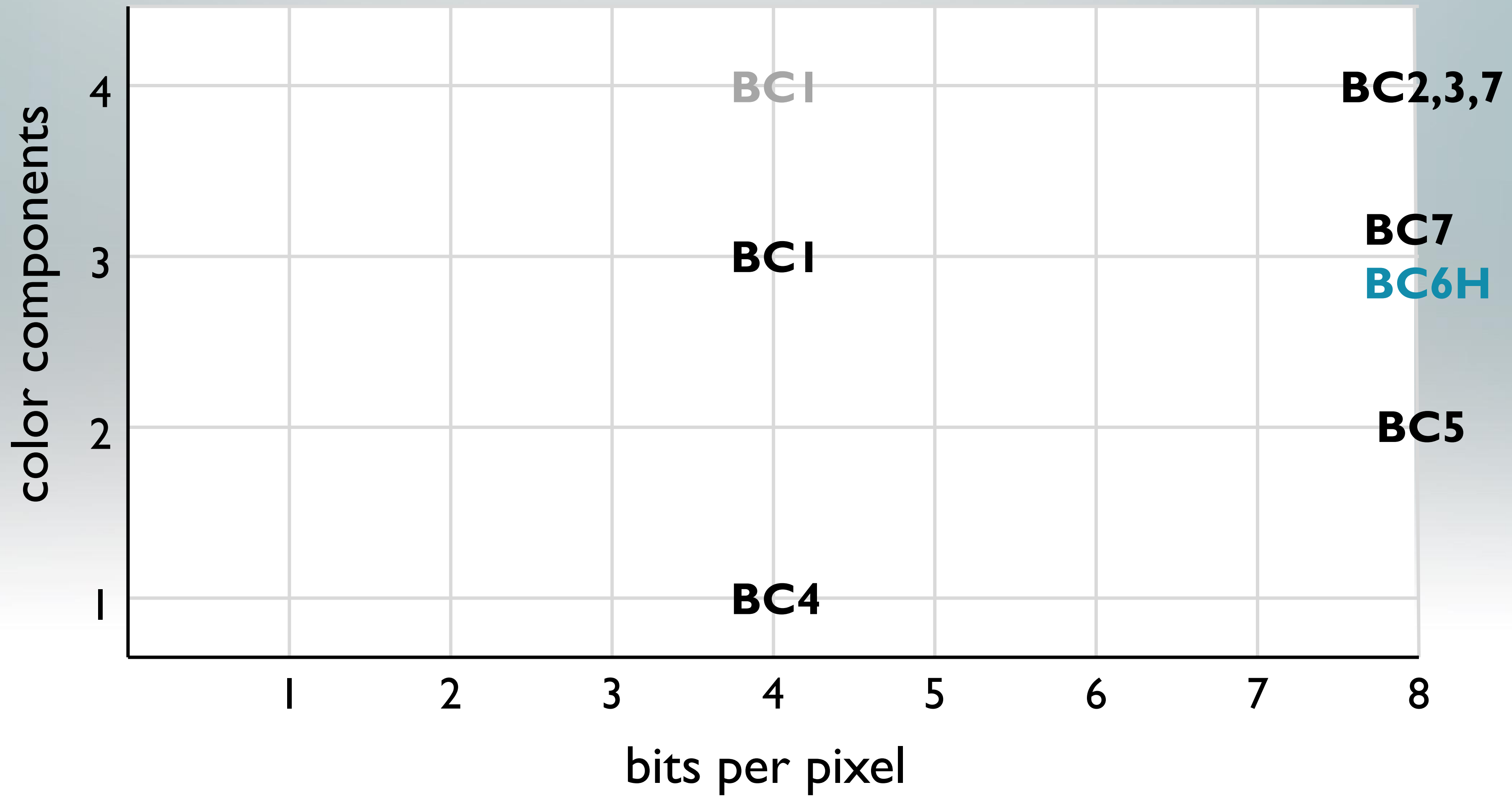

3D Properties

Each use has its own requirements

- Number of color components
- Dynamic range (LDR vs HDR)
- Dimensionality (2D vs 3D)
- Quality (≈ bit rate)

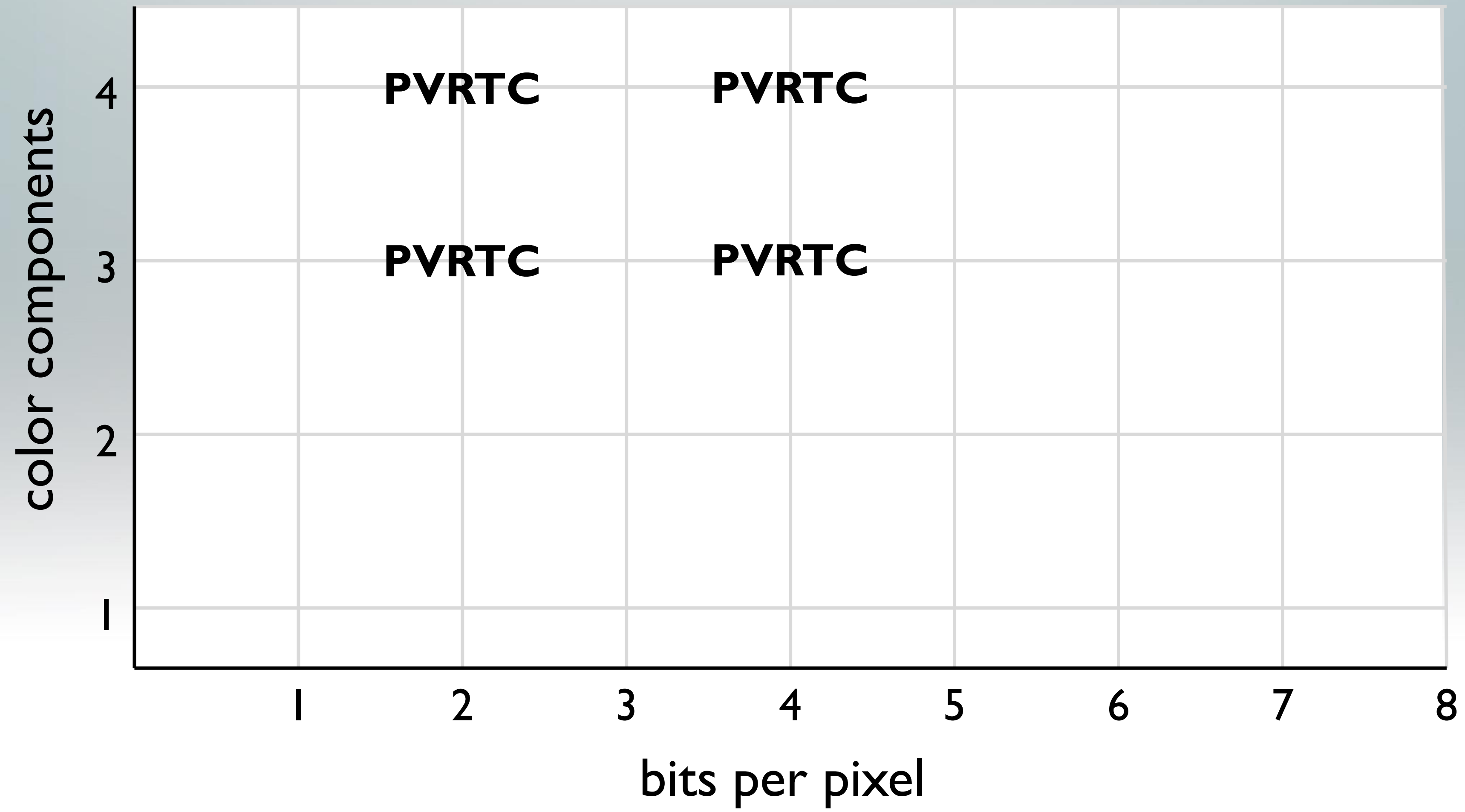
Compression Today

Microsoft® DirectX®



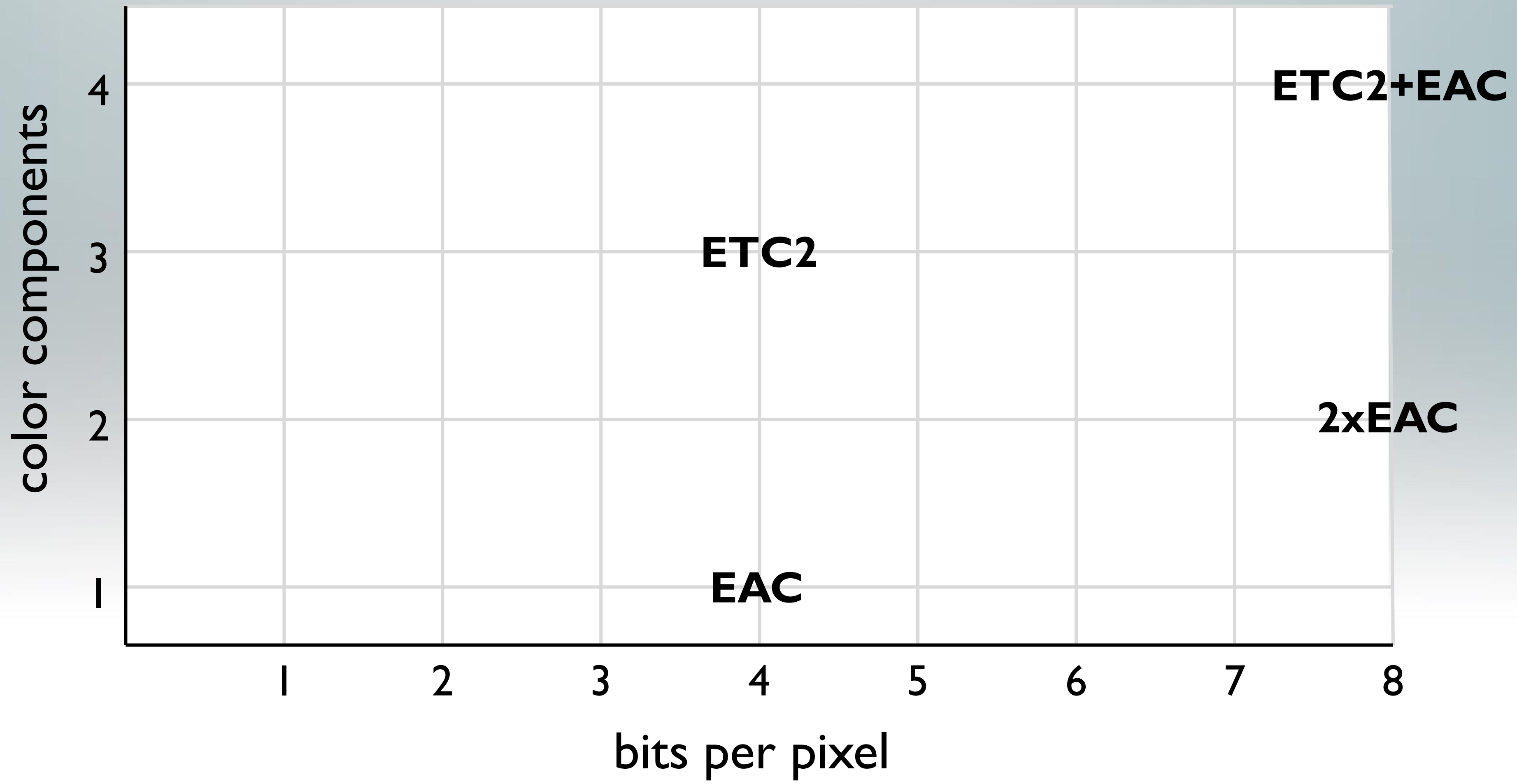
Compression Today

Imagination Technologies® PVRTC™



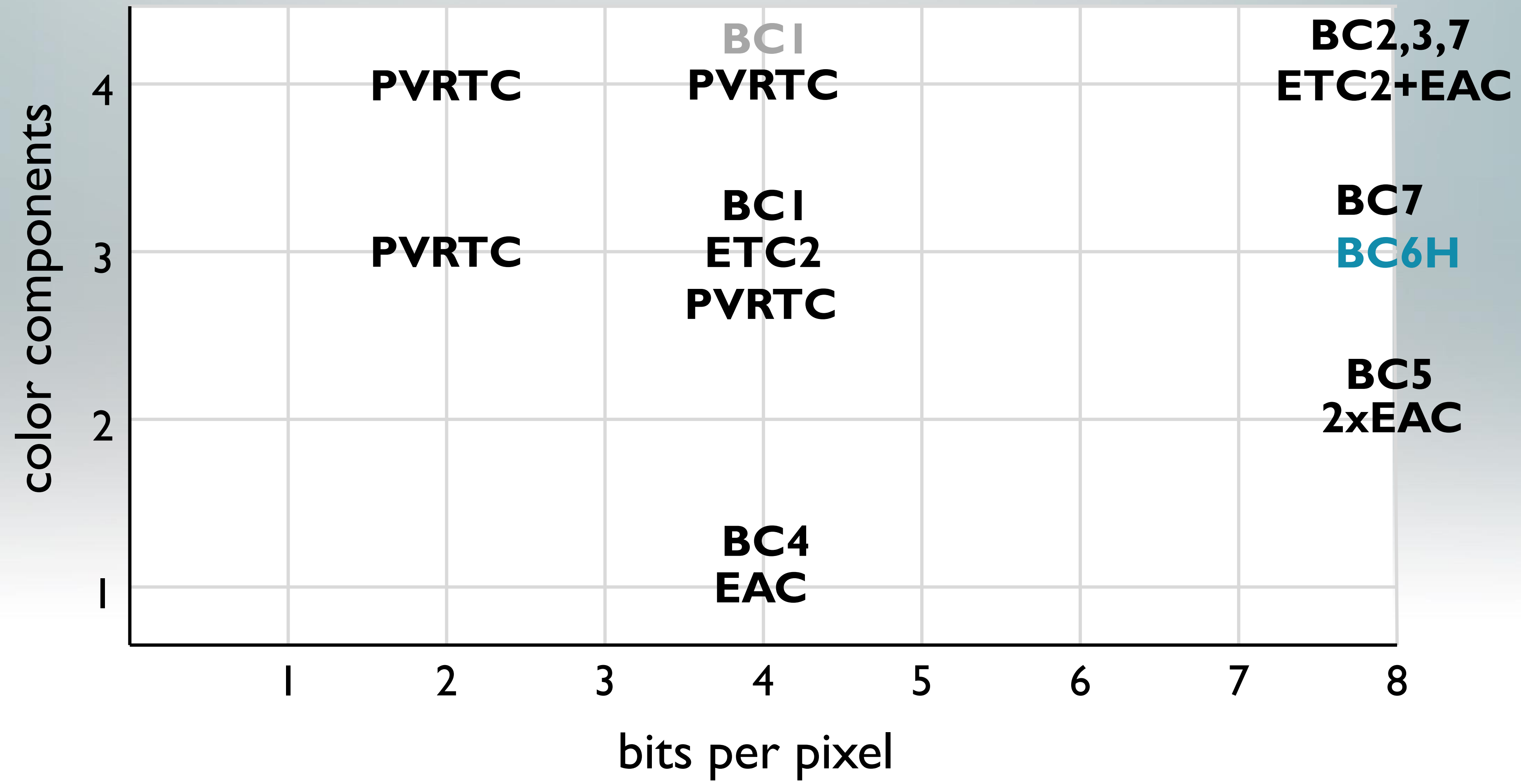
Compression Today

OpenGL® ES™ 3.0



Compression Today

Putting them all together



Compression Today: Observations

What a mess!

- Horribly fragmented
- Many formats are proprietary
- Must recondition / requalify assets for every format

Where's my use case?

- Only one low-bit-rate format (PVRTC RGB/RGBA, 2bpp)
- Only one HDR format (RGB, 8bpp)
- Poor support for 1 and 2 channel images
- Very coarse quality / size tradeoff

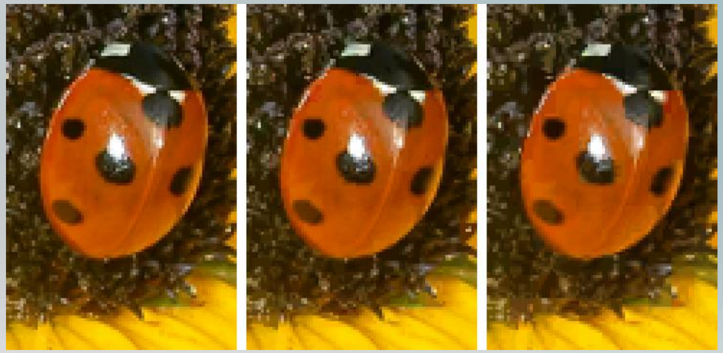
Introducing ASTC

Adaptive Scalable Texture Compression

- Created by ARM in response to a Khronos competition
- …with a valuable technical contribution from AMD

Functionality

- Scalable bit rate: 8bpp down to <1bpp in fine steps
- Orthogonal choice of base format (L, LA, RGB, RGBA)
- Both LDR and HDR pixel formats
- Both 2D and 3D textures
- Very high quality



ASTC Compression

8bpp 3.56bpp 2bpp

ASTC Bit Rates

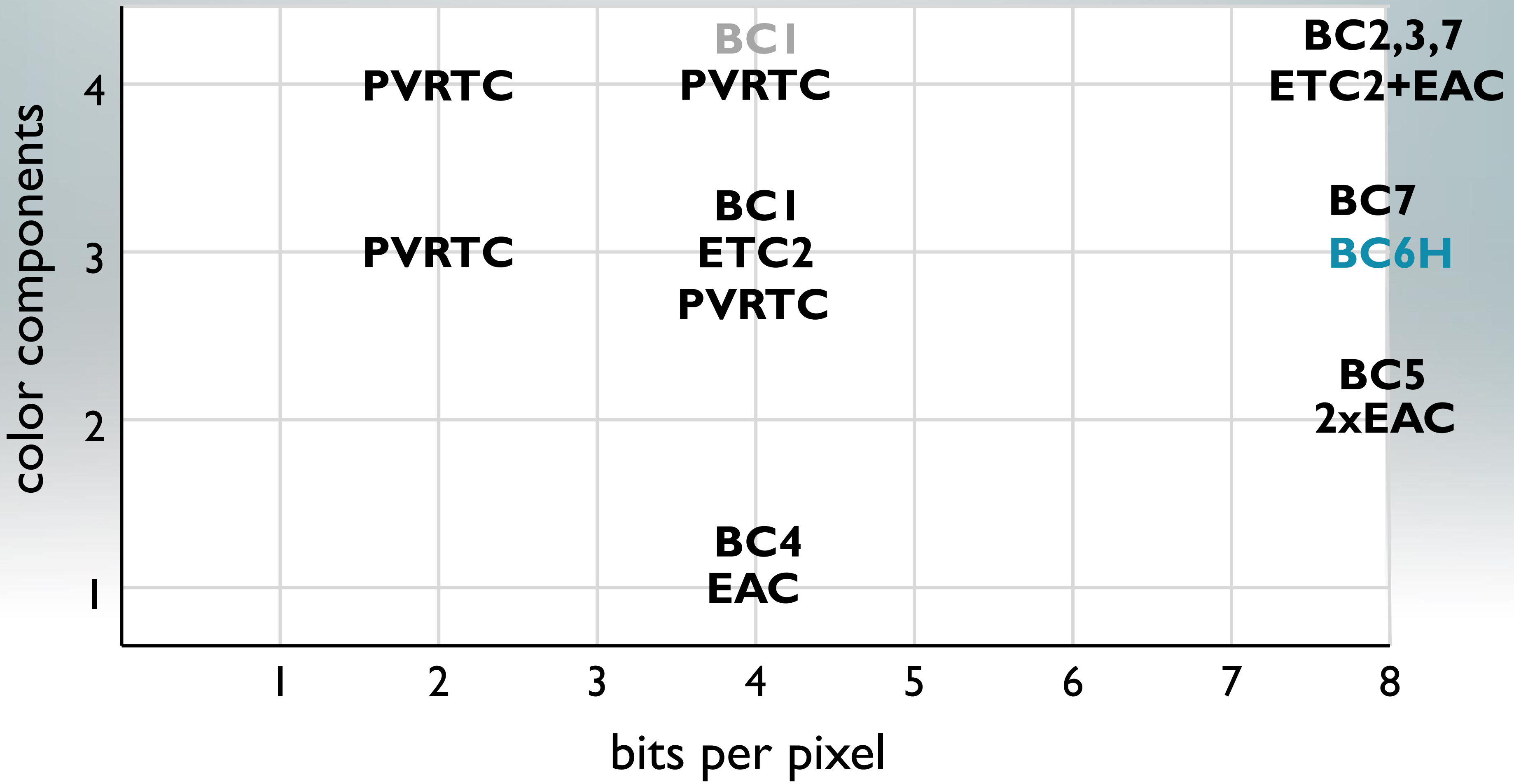
Block-based paradigm generalized to 3D

- Fixed block size of 128 bits
- Bit rate determined by block size

2D Bit Rates				3D Bit Rates			
4x4	8.00 bpp	10x5	2.56 bpp	3x3x3	4.74 bpp	5x5x4	1.28 bpp
5x4	6.40 bpp	10x6	2.13 bpp	4x3x3	3.56 bpp	5x5x5	1.02 bpp
5x5	5.12 bpp	8x8	2.00 bpp	4x4x3	2.67 bpp	6x5x5	0.85 bpp
6x5	4.27 bpp	10x8	1.60 bpp	4x4x4	2.00 bpp	6x6x5	0.71 bpp
6x6	3.56 bpp	10x10	1.28 bpp	5x4x4	1.60 bpp	6x6x6	0.59 bpp
8x5	3.20 bpp	12x10	1.07 bpp				
8x6	2.67 bpp	12x12	0.89 bpp				

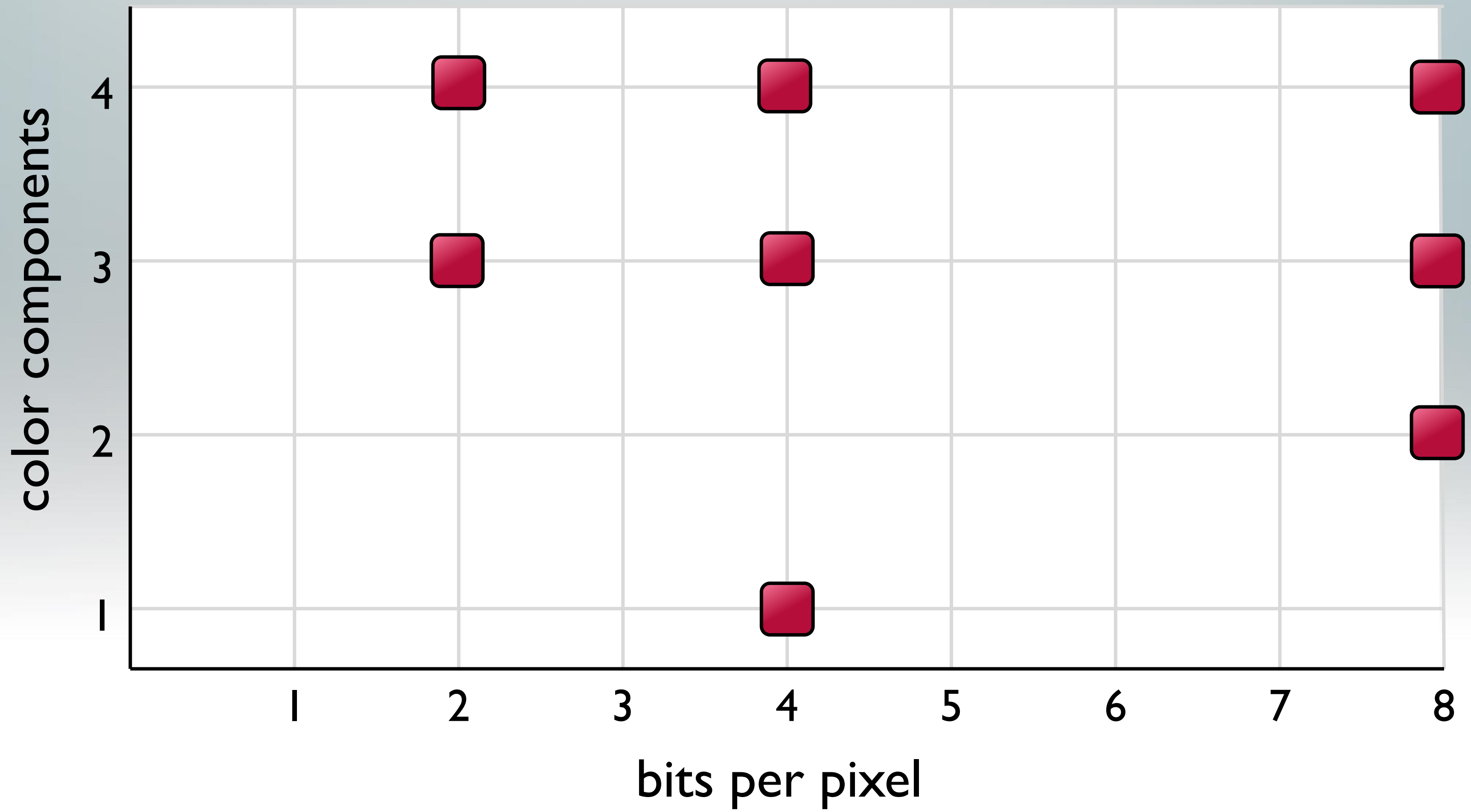
Compression Today

Putting it all together

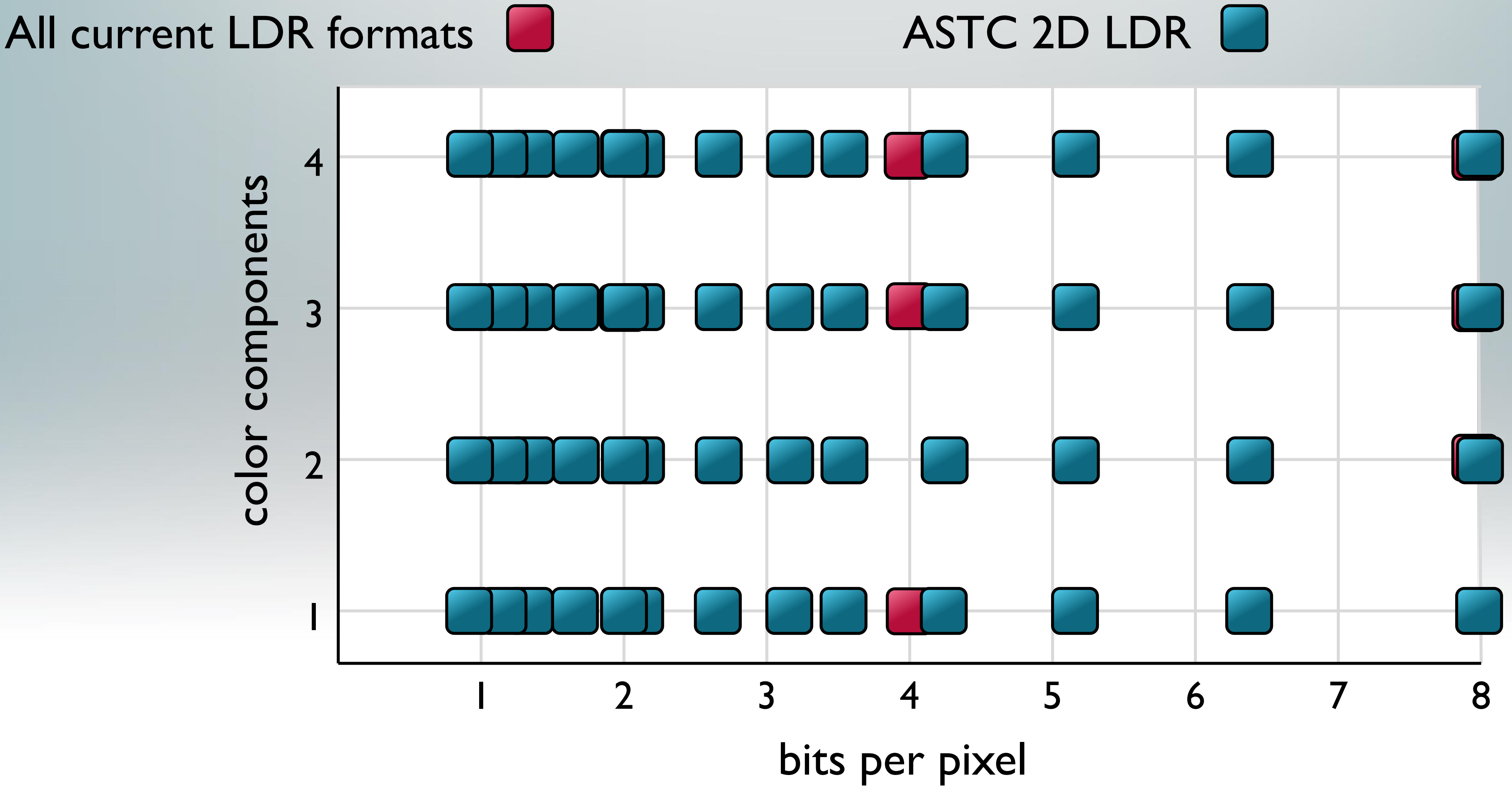


Compression Today

All current 2D LDR formats 🟥

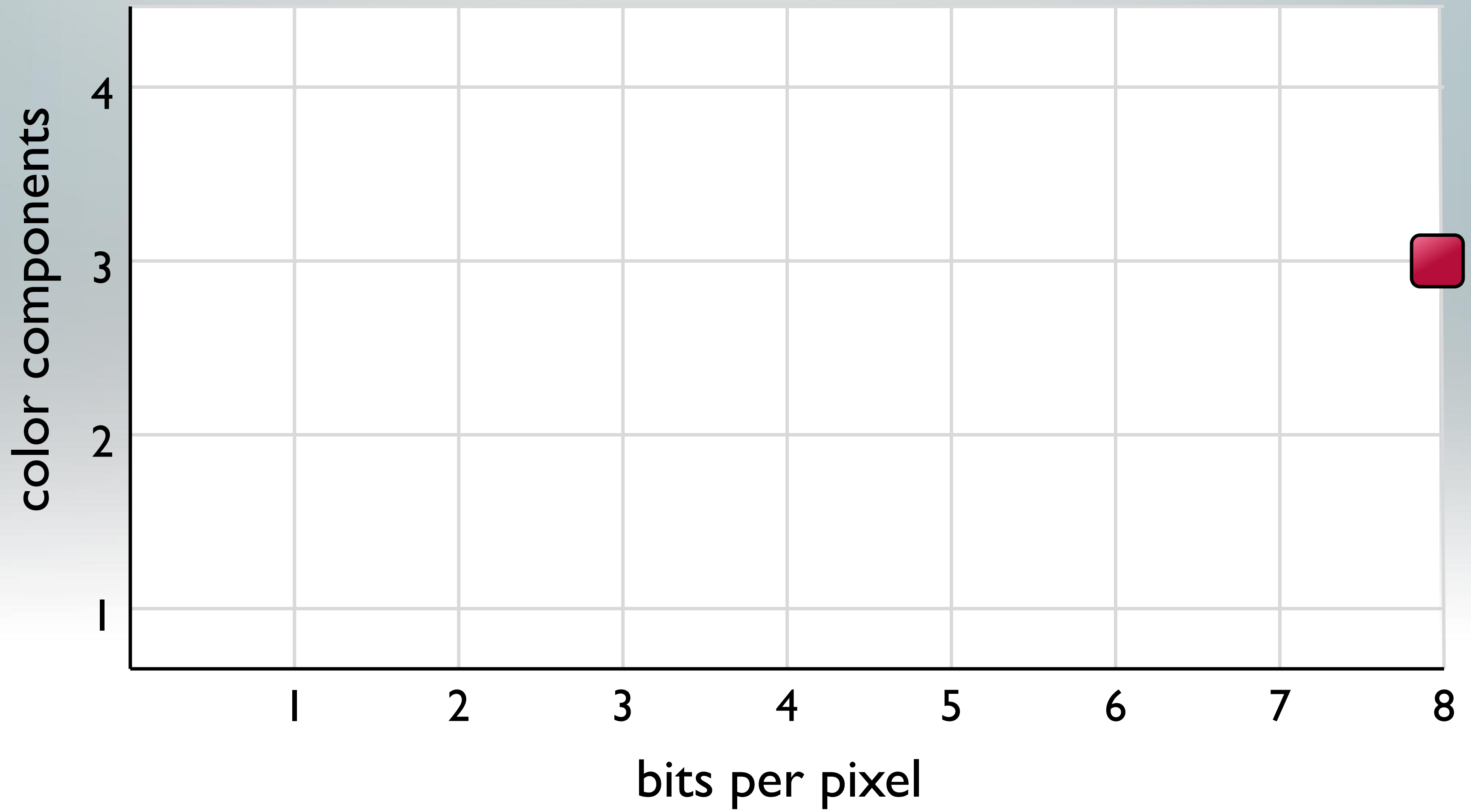


Compression Today

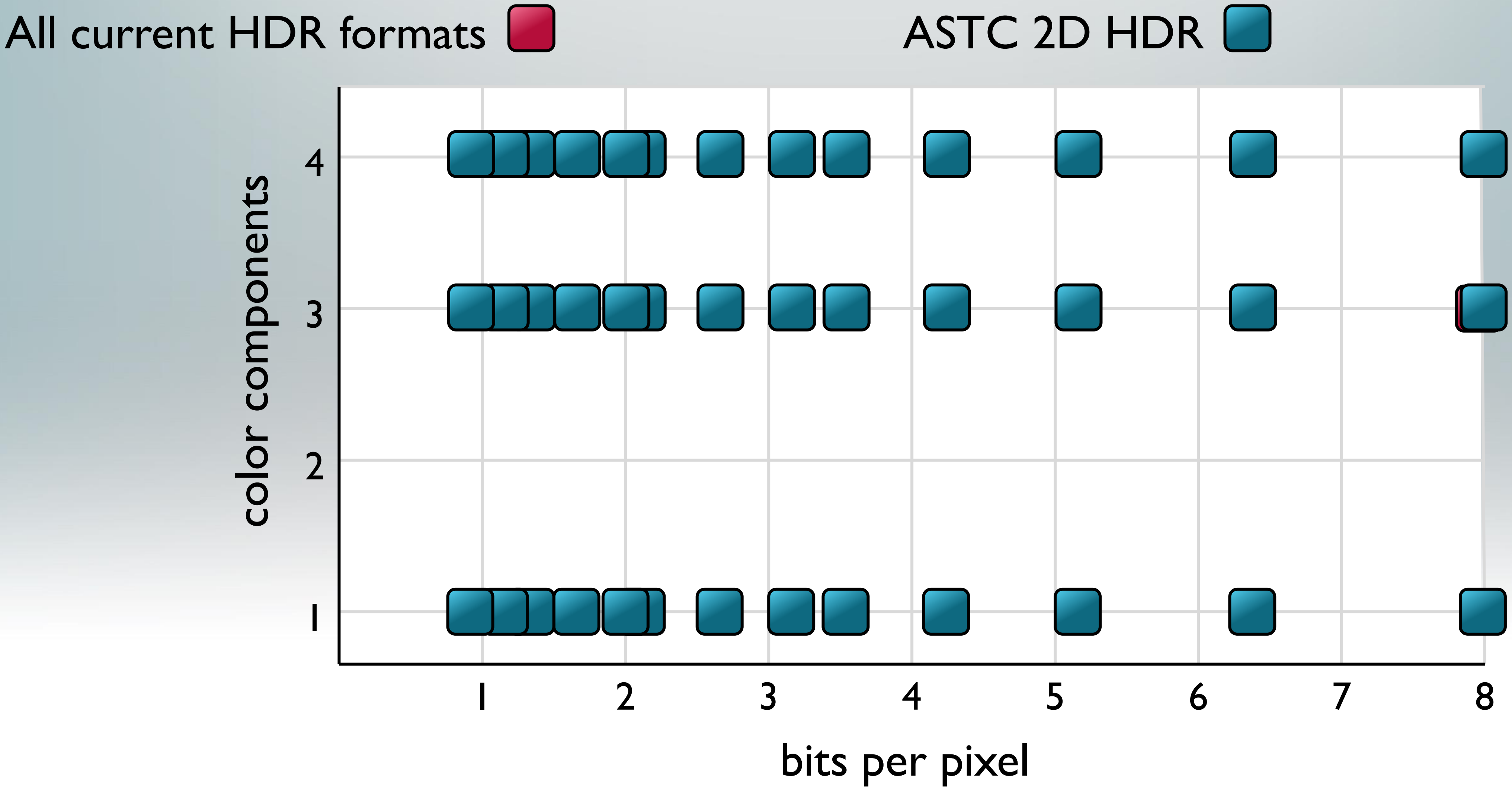


Compression Today (HDR)

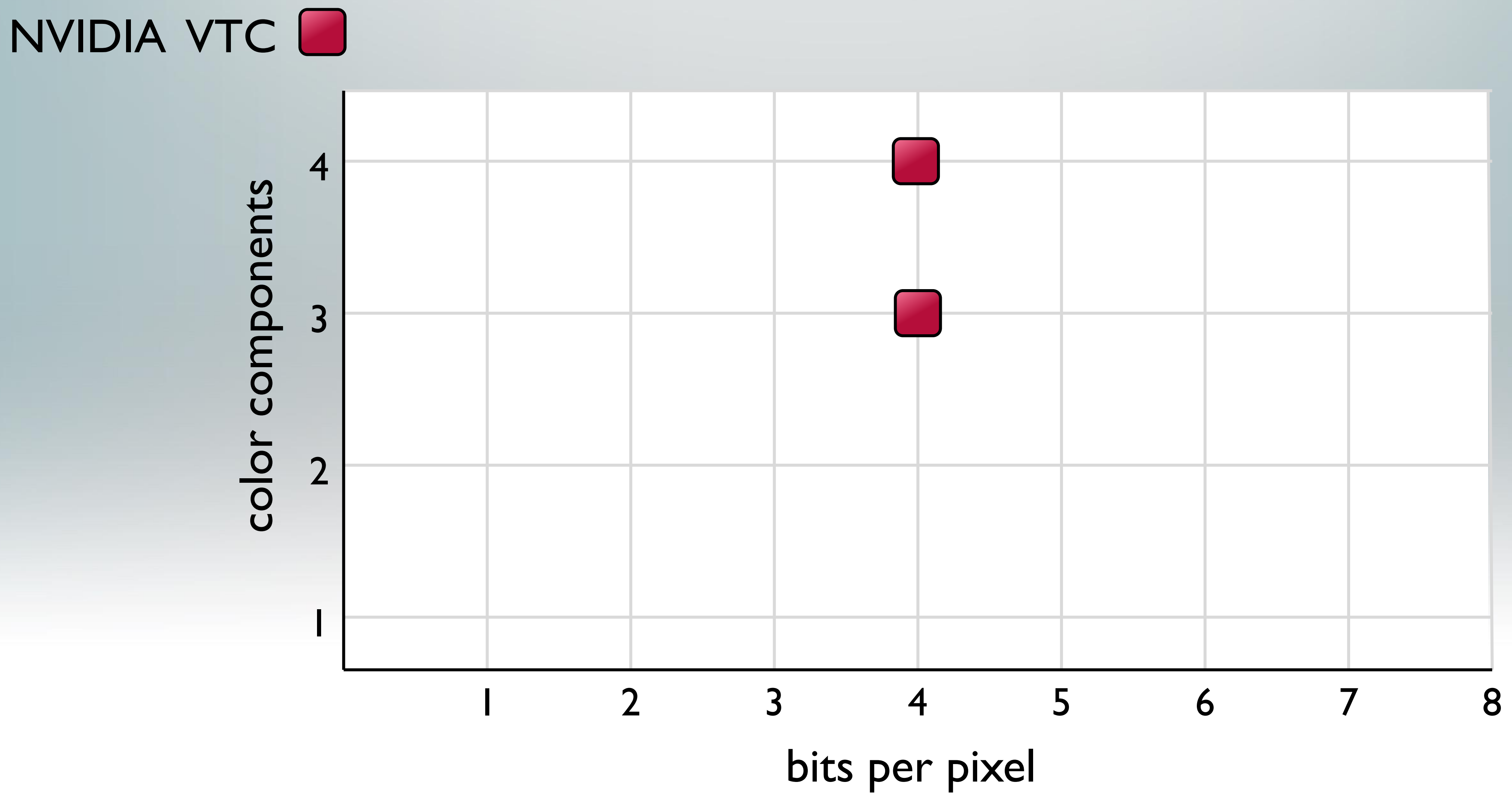
All current HDR formats 🟥



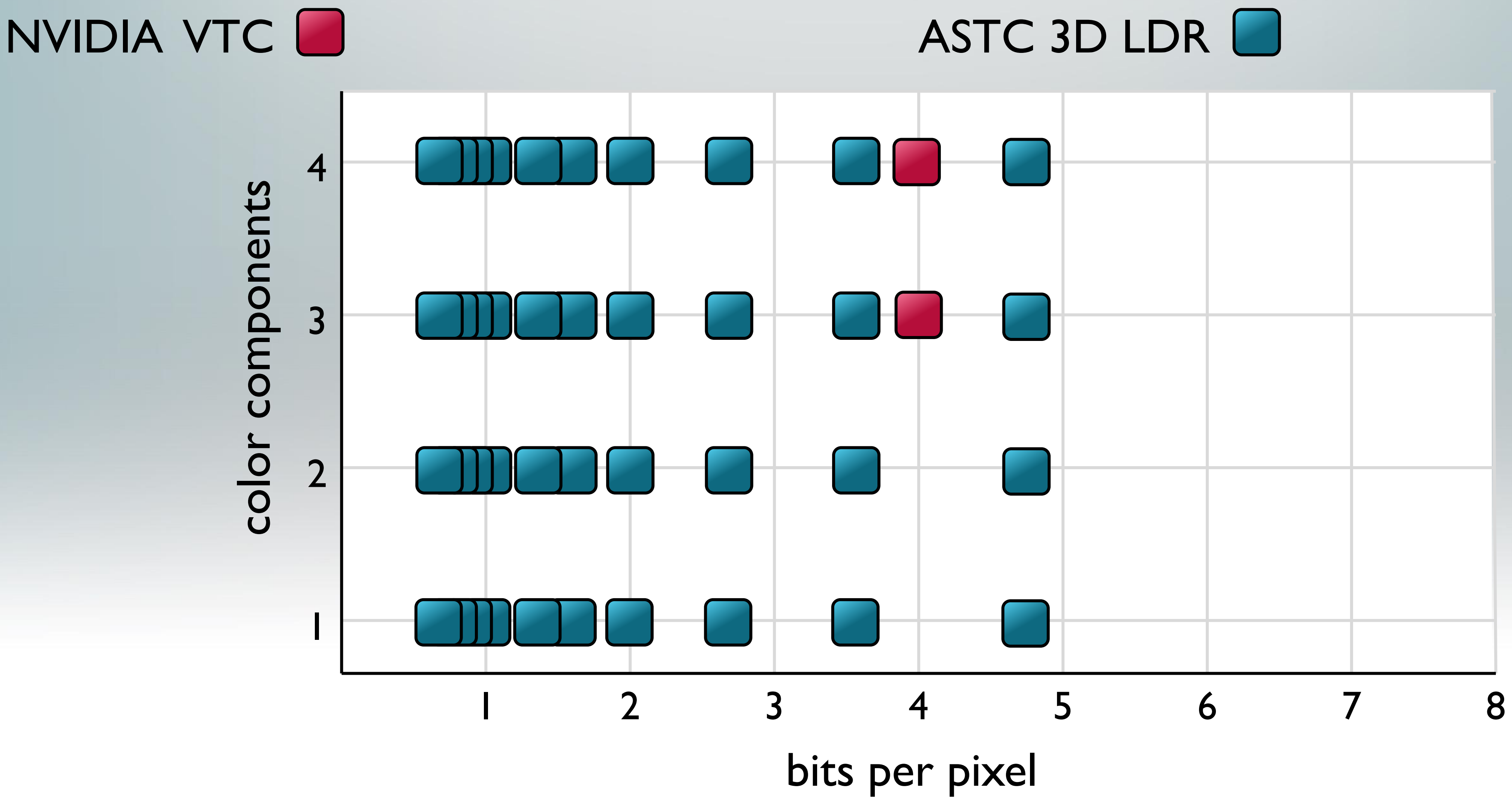
Compression Today (HDR)



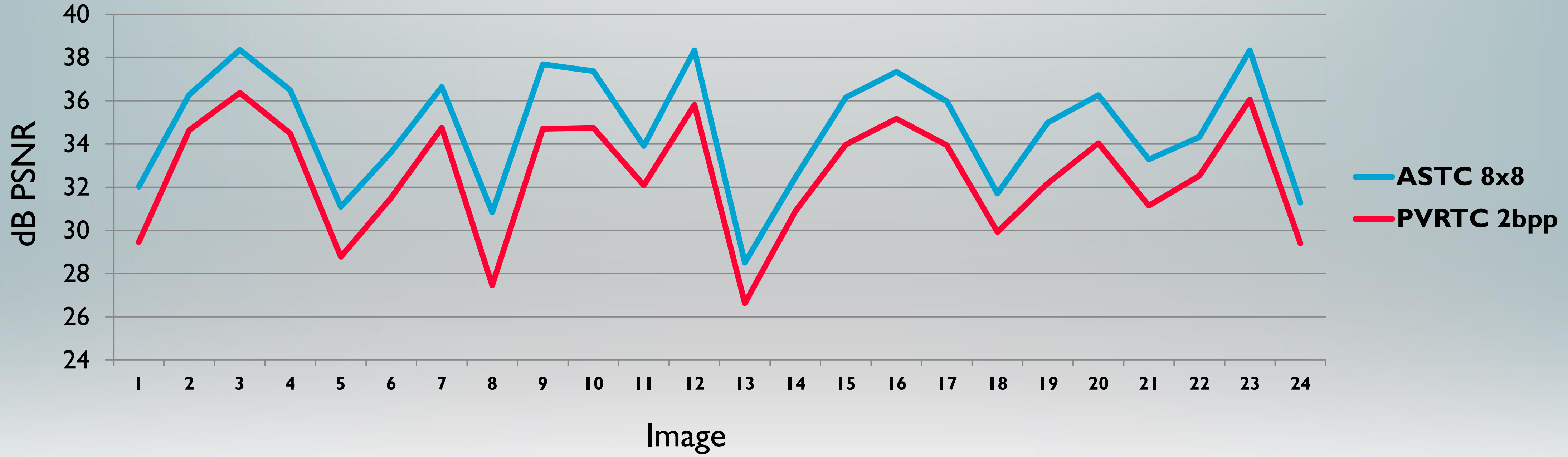
Compression Today (3D)



Compression Today (3D LDR)



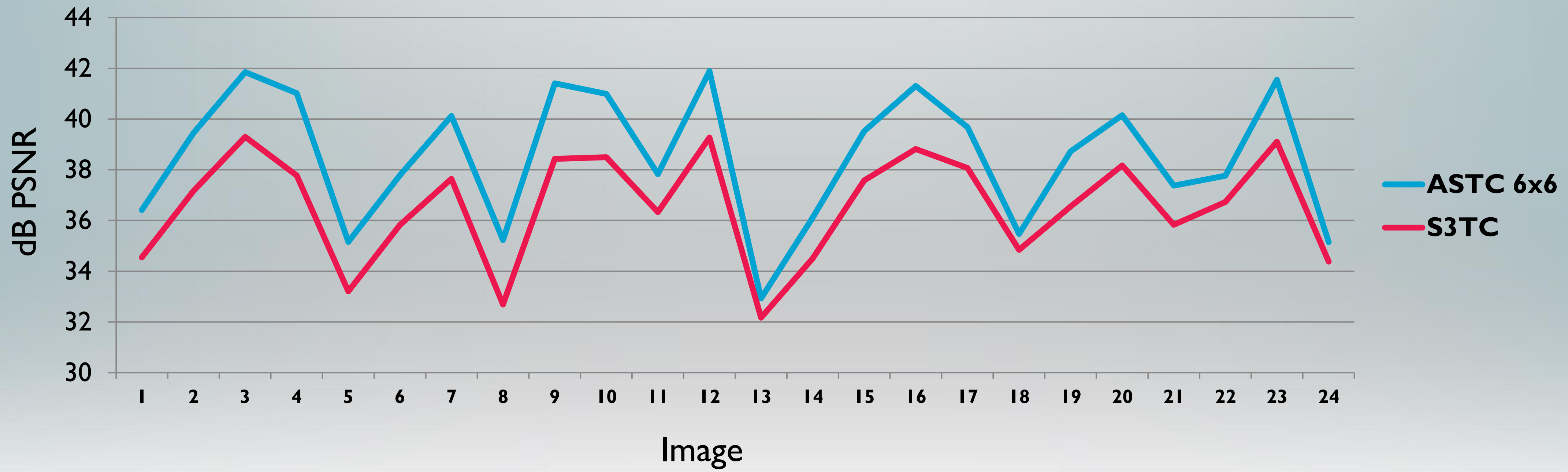
Quality Comparison – RGB LDR 2bpp



24 natural images

- ASTC vs PVRTC at 2bpp

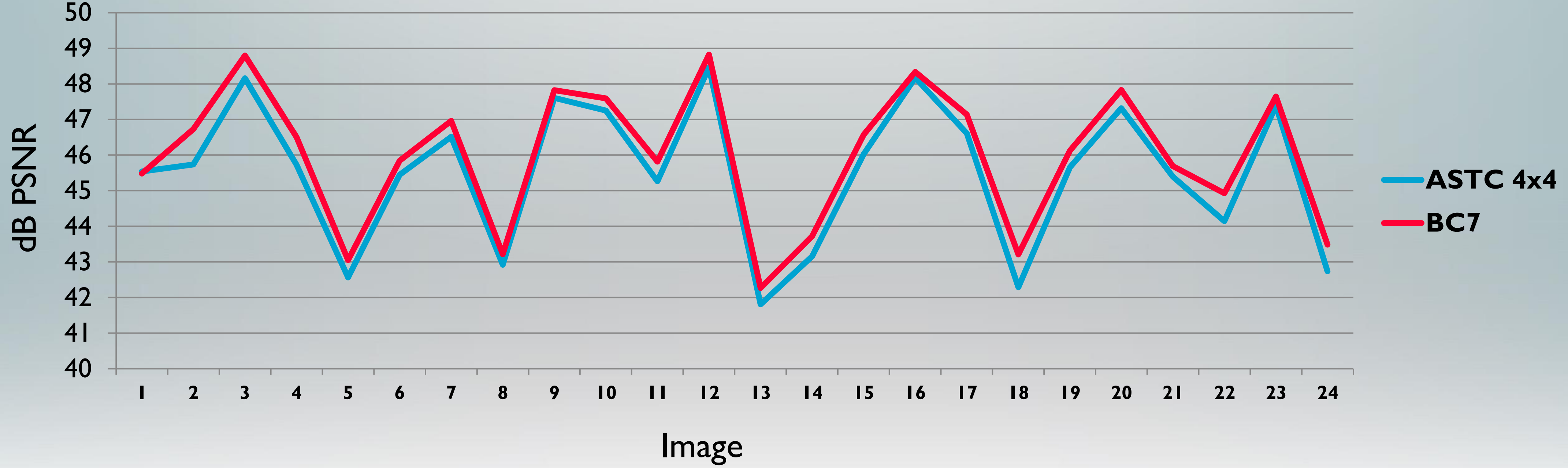
Quality Comparison – RGB LDR medium bit rate



24 natural images

- ASTC 3.56 bpp vs S3TC (DXT1) at 4bpp

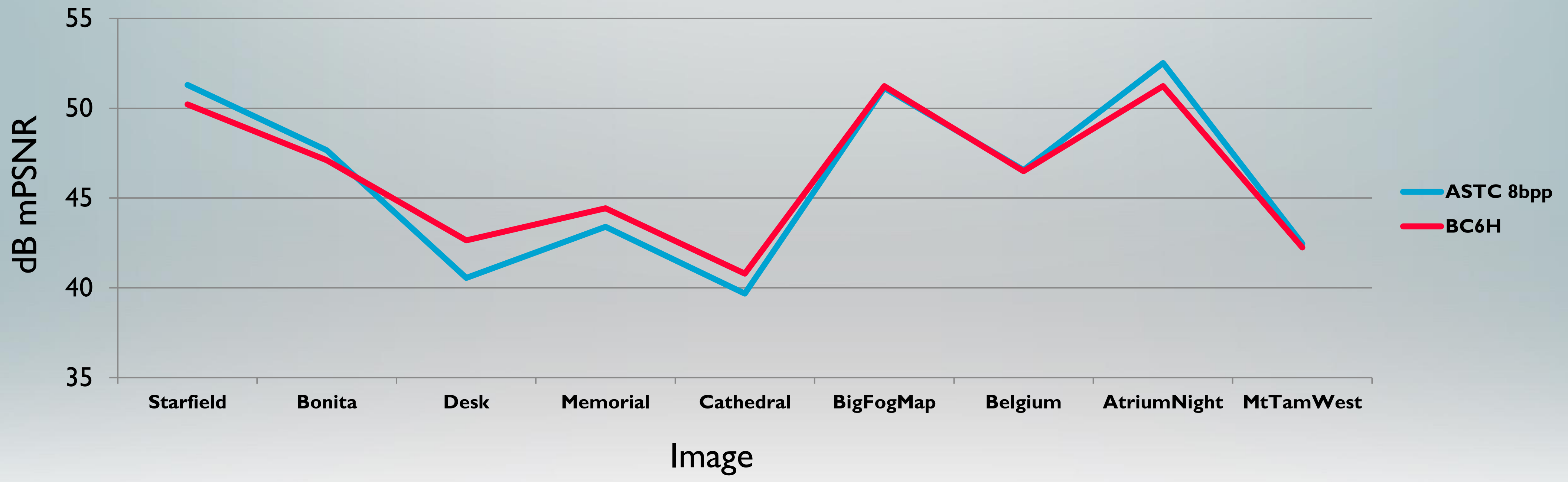
Quality Comparison – RGB LDR medium bit rate



24 natural images

- ASTC vs BC7 at 8bpp

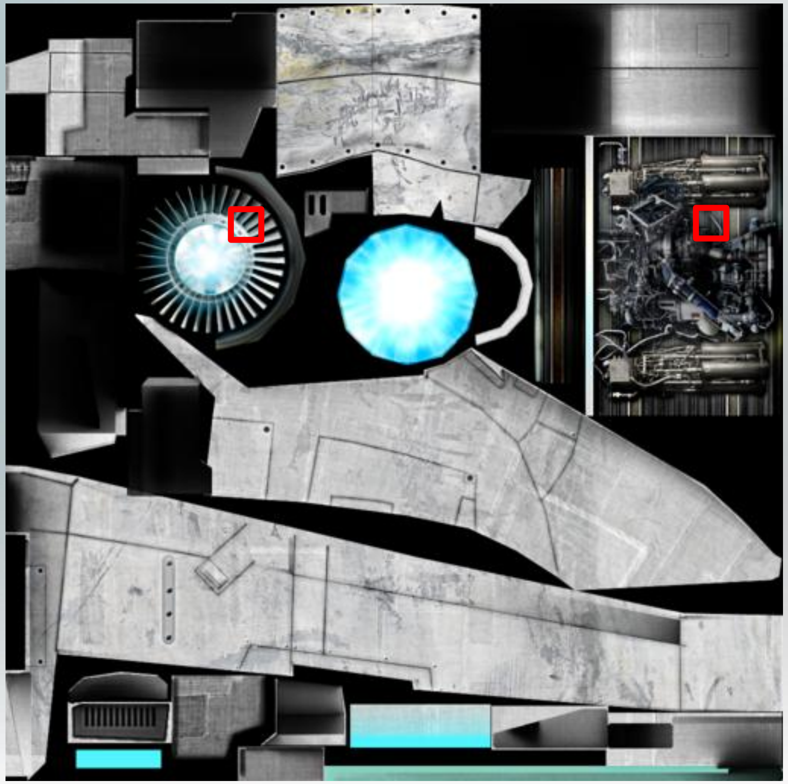
Quality Comparison – RGB HDR high bit rate



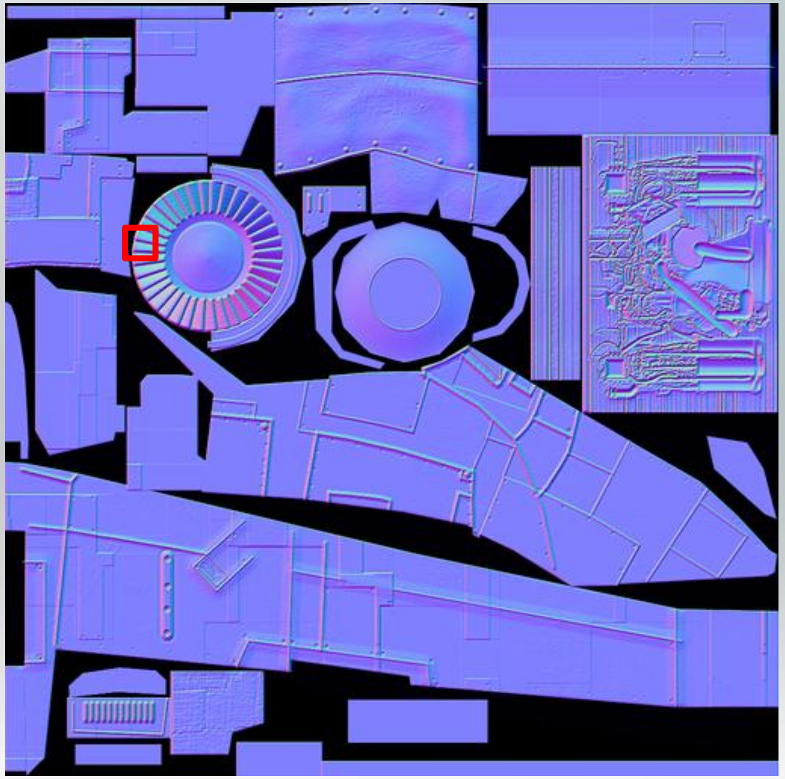
Selected OpenEXR example images

- ASTC vs BC6H at 8bpp

Quality Comparison – Images



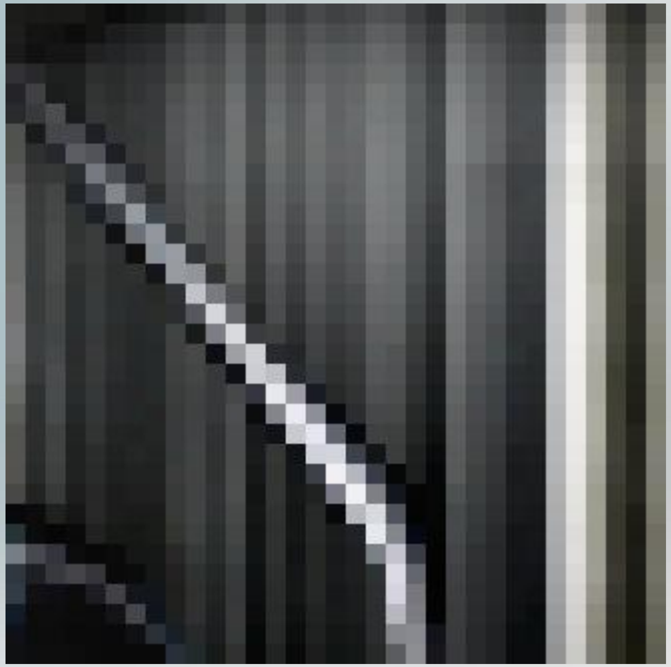
DIFFUSE MAP



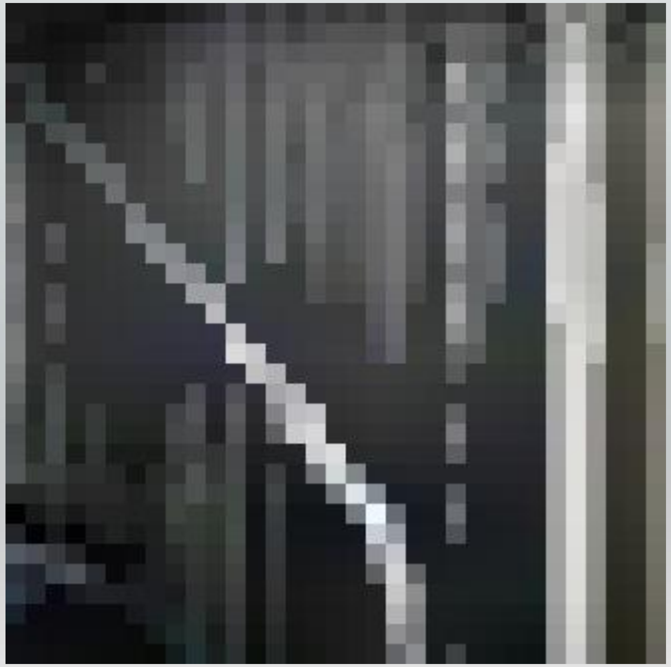
NORMAL MAP

Assets from ARM Trueforce demo

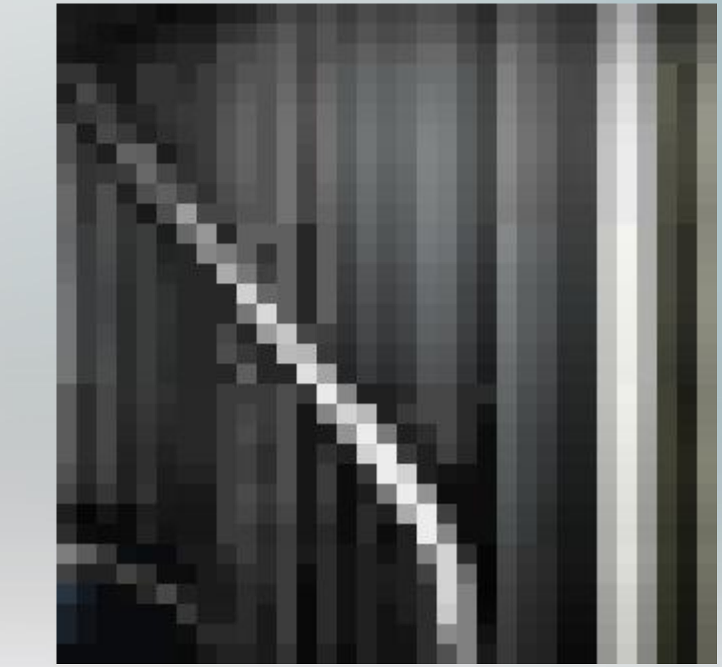
Quality Comparison – diffuse map, low bit rate



ORIGINAL



PVRTC 2 bpp: PSNR 31.7 dB

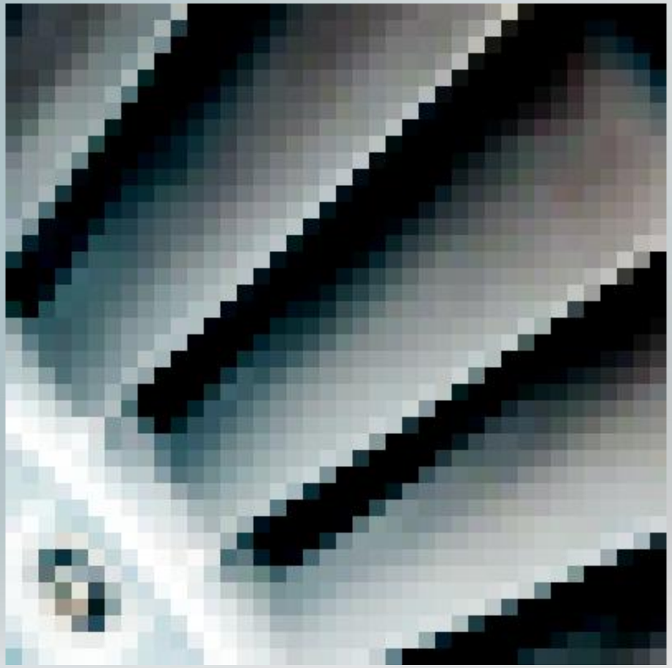


ASTC 8x8: PSNR 36.0 dB

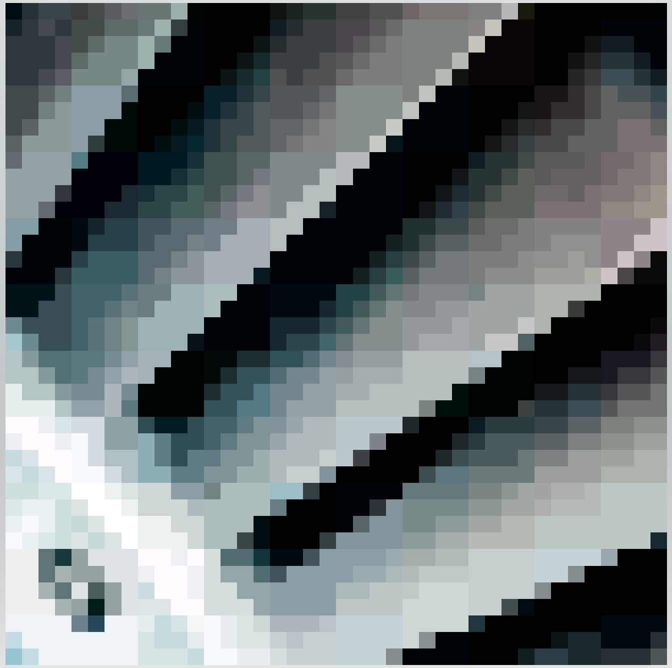
ASTC at 2.0 bpp vs PVRTC at 2 bpp

- 4.3 dB PSNR advantage

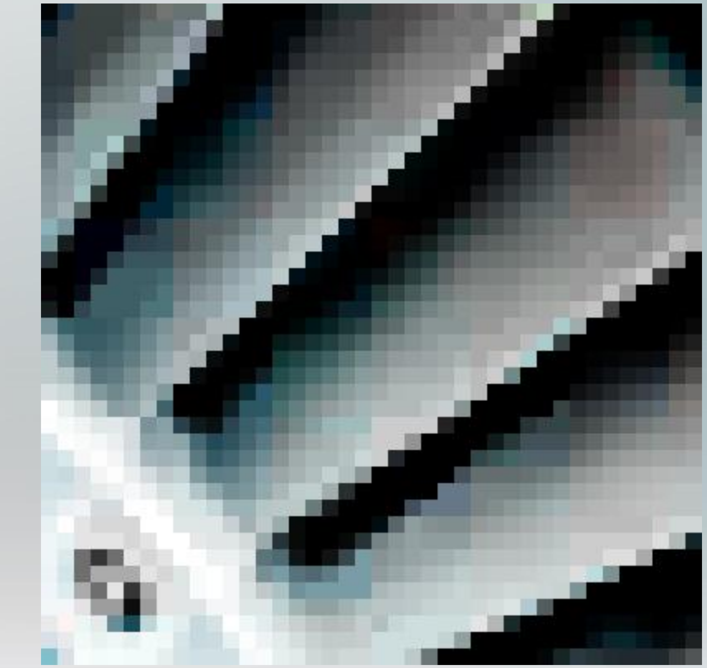
Quality Comparison – diffuse map, medium bit rate



ORIGINAL



S3TC 4 bpp: PSNR 30.7 dB

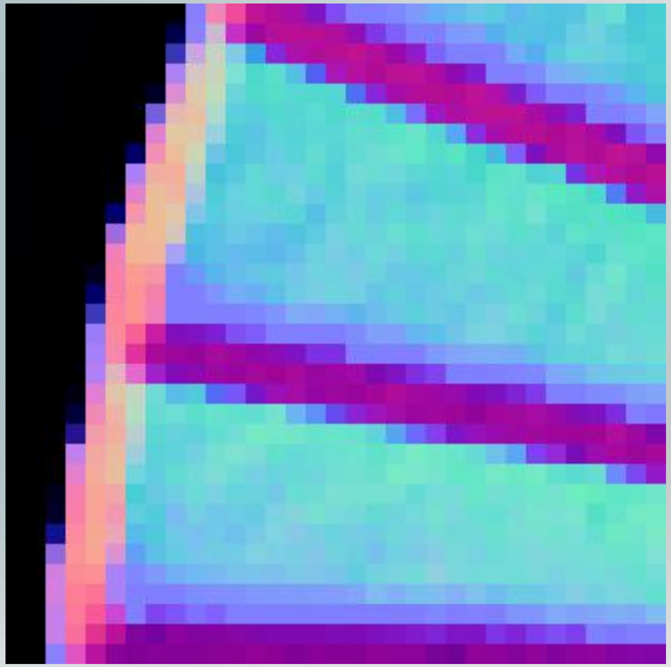


ASTC 6x6: PSNR 33.5 dB

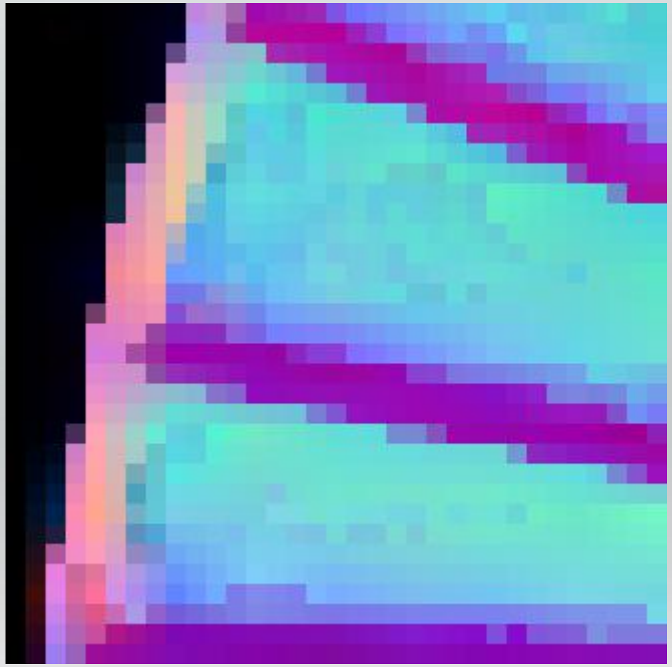
ASTC at 3.56 bpp vs S3TC at 4 bpp

- 2.8 dB PSNR advantage

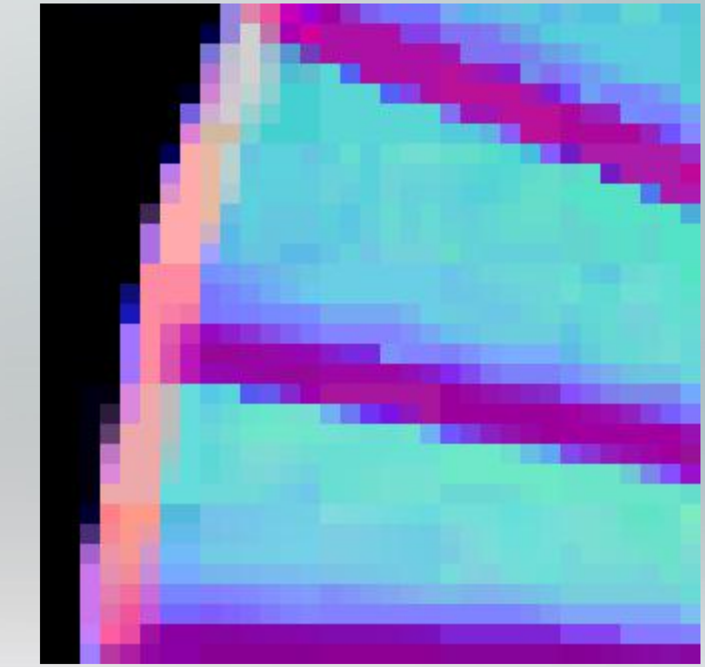
Quality Comparison – normal map



ORIGINAL



PVRTC 4 bpp 3-component

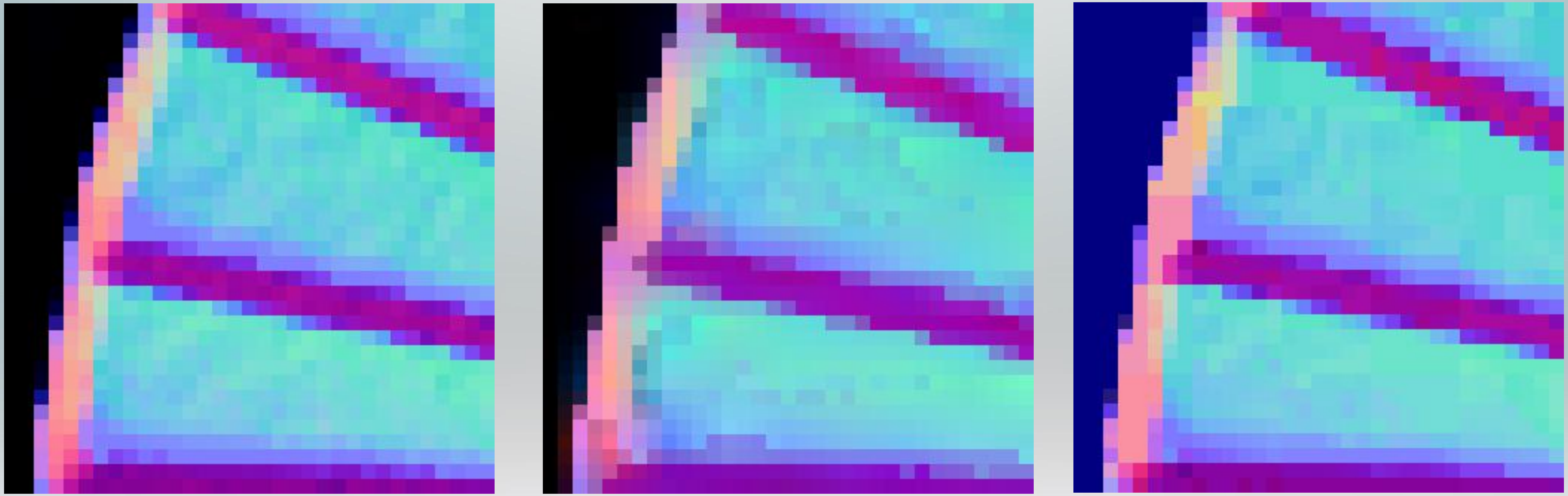


ASTC 6x6 3-component

ASTC at 3.56 bpp vs S3TC at 4 bpp

- 3-component normals

Quality Comparison – normal map



ORIGINAL

PVRTC 4 bpp 3-component

ASTC 6x6 2-component

ASTC at 3.56 bpp vs PVRTC at 4 bpp

- 3-component vs 2-component normals

Support – Documentation and Tools

How and why it works

- Nystad et al, *Adaptive Scalable Texture Compression*, Proc. HPG 2012

Evaluation codec (source)

- http://www.malideveloper.com/ and navigate to "tools"

Now supported across the Mali development tool chain

- *Mali Texture Compression Tool*
- *Mali OpenGL ES 3.0 Emulator*

Support - Standards

ASTC 2D-LDR subset defined to promote fast adoption

- No 3D, no HDR
- Pure subset – fully compatible with a full ASTC decoder

ASTC LDR extension approved by The Khronos Group

- KHR_texture_compression_astc_ldr
- Defined for both OpenGL® and OpenGL ES™

Support - Hardware

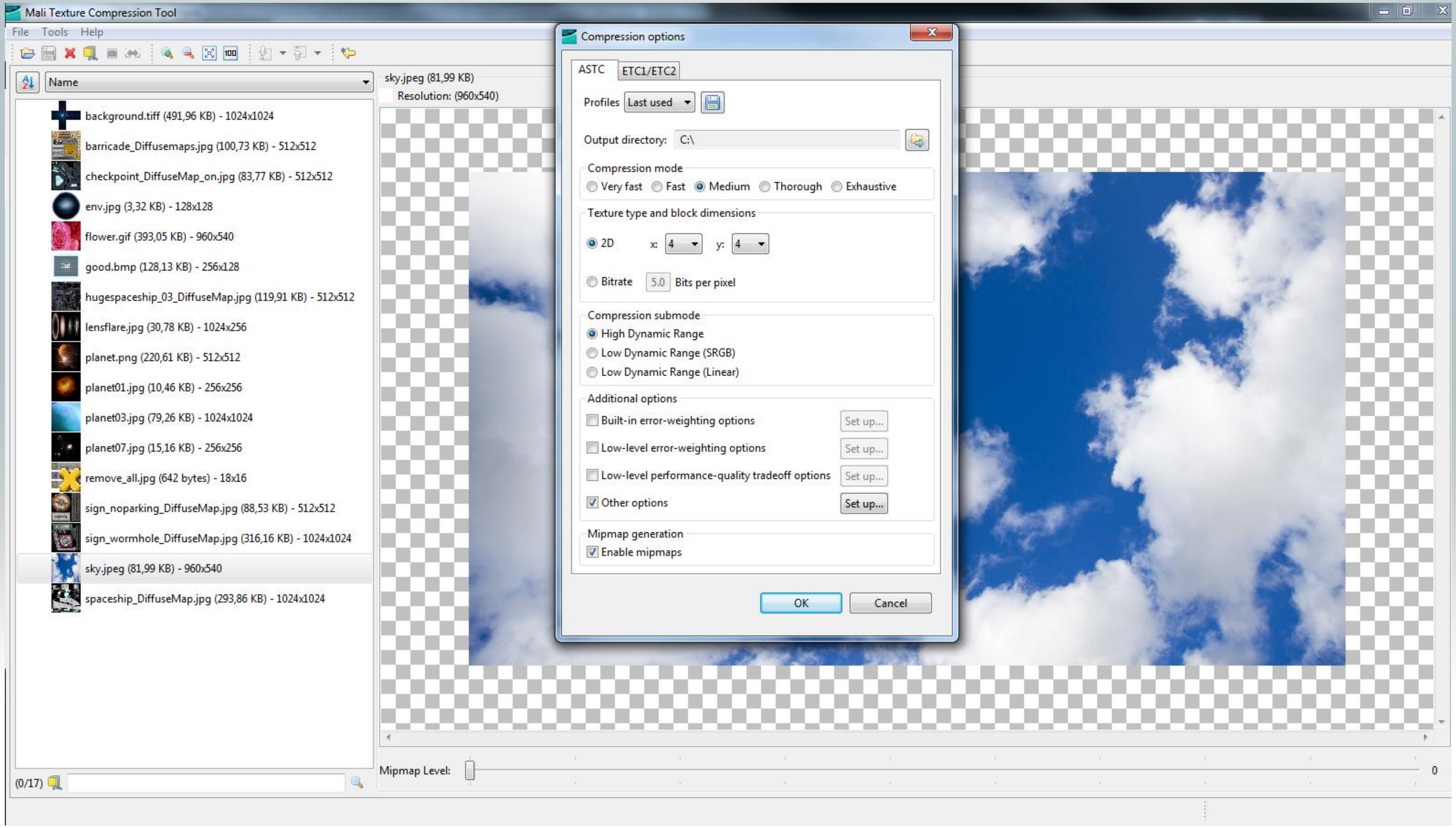
ASTC will be supported in all upcoming ARM GPUs

Currently available in

- ARM Mali-T624 and Mali-T628
- ARM Mali-T678

Partner silicon is on the way!

Questions



Mali Texture Compression Tool

- background.tiff (491,96 KB) - 1024x1024
- barricade_Diffusemaps.jpg (100,73 KB) - 512x512
- checkpoint_DiffuseMap_on.jpg (83,77 KB) - 512x512
- env.jpg (3,32 KB) - 128x128
- flower.gif (393,05 KB) - 960x540
- good.bmp (128,13 KB) - 256x128
- hugespaceship_03_DiffuseMap.jpg (119,91 KB) - 512x512
- lensflare.jpg (30,78 KB) - 1024x256
- planet.png (220,61 KB) - 512x512
- planet01.jpg (10,46 KB) - 256x256
- planet03.jpg (79,26 KB) - 1024x1024
- planet07.jpg (15,16 KB) - 256x256
- remove_all.jpg (642 bytes) - 18x16
- sign_noparking_DiffuseMap.jpg (88,53 KB) - 512x512
- sign_wormhole_DiffuseMap.jpg (316,16 KB) - 1024x1024
- sky.jpeg (81,99 KB) - 960x540
- spaceship_DiffuseMap.jpg (293,86 KB) - 1024x1024

(0/17)

sky.jpeg (81,99 KB)

Resolution: (960x540)

Mipmap Level: 0

Compression options

Profiles: Last used

Output directory: C:\

Compression mode
○ Very fast ○ Fast ● Medium ○ Thorough ○ Exhaustive

Texture type and block dimensions
● 2D x: 4 y: 4
○ Bitrate 5.0 Bits per pixel

Compression submode
● High Dynamic Range
○ Low Dynamic Range (SRGB)
○ Low Dynamic Range (Linear)

Additional options
□ Built-in error-weighting options Set up...
□ Low-level error-weighting options Set up...
□ Low-level performance-quality tradeoff options Set up...
☑ Other options Set up...

Mipmap generation
☑ Enable mipmaps

How texture compression works

Image is divided into blocks

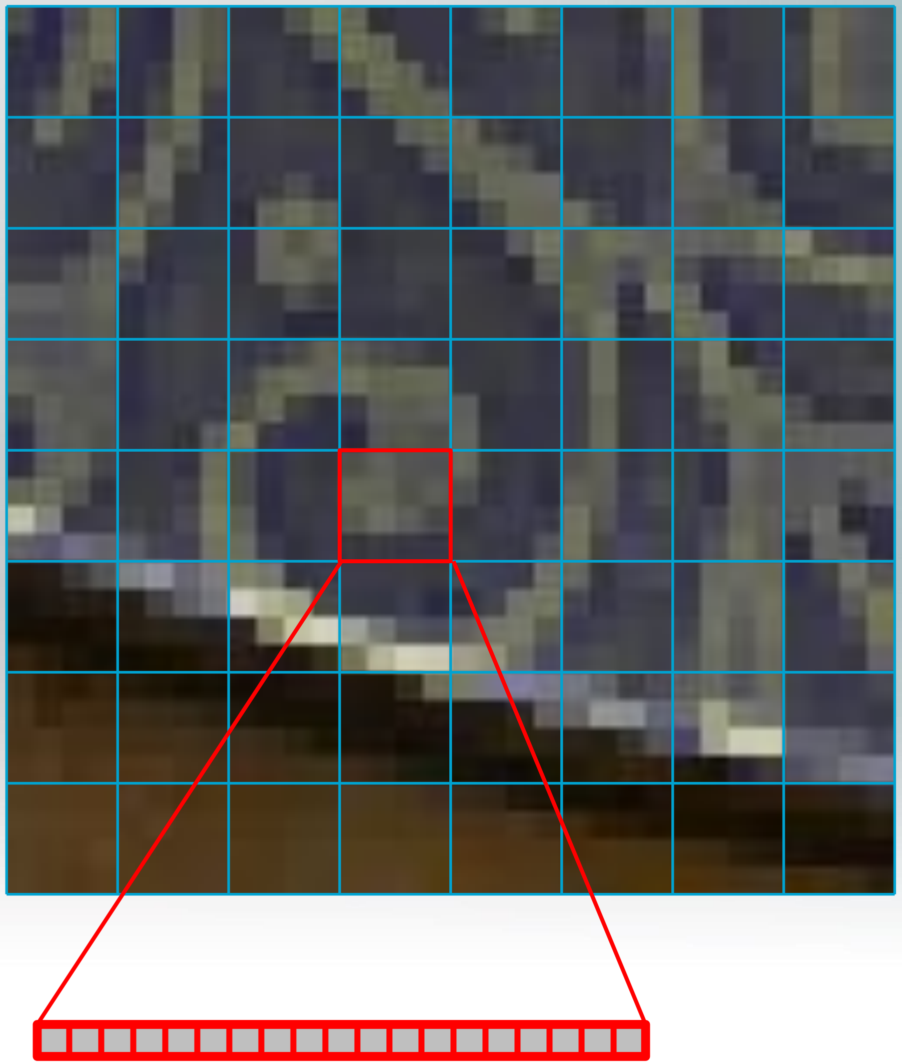
- E.g. 4x4 pixels

Encode blocks as bit strings

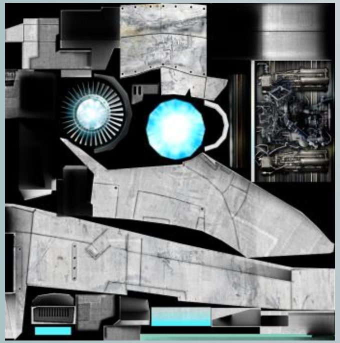
- Fixed length, e.g. 64 bits / block
- Fixed rate, lossy encoding
- Bpp = bits per block / pix per block
- Constant-time random access

Everybody does this

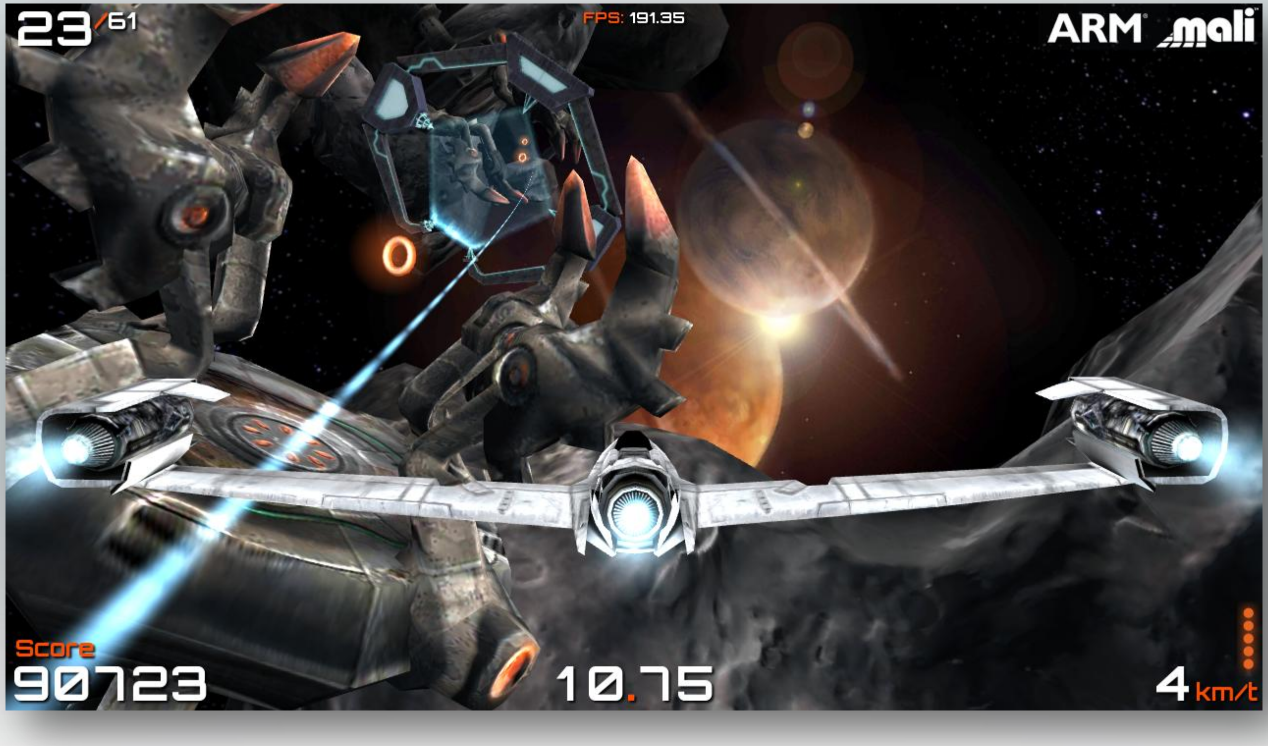
- DXTn, RGTC, BC7, PVRTC (sort of)…



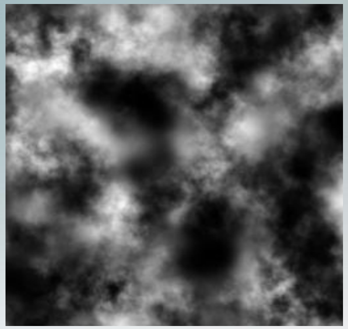
A universal tool



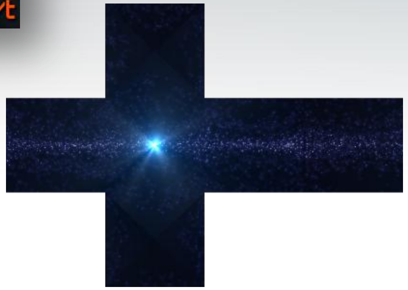
Reflectance



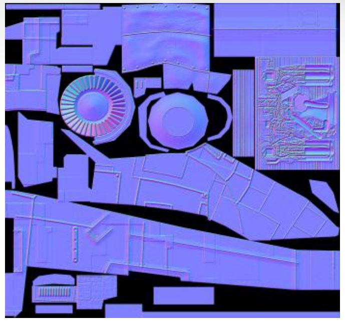
Illuminance



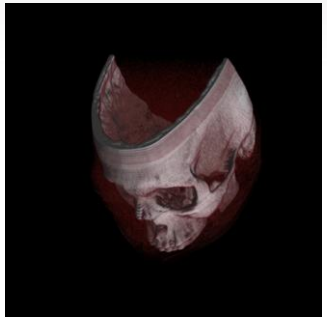
Gloss, Height, etc



Lighting environment



Normals



Everything else

Graphics: It's all about the textures

The Sword & Shield