Compressed Texture Transmission Format

Mark Callow

glTF Meetup, June 14th, 2018
This Talk Describes a Work in Progress
Required Specifications

- Container
  - Textures often consist of multiple images
  - A container makes for easier use.

- Format(s) for the image bits
Image Bits - Issues

- Can use image formats defined by the GPU APIs but
  - Uncompressed formats too large for transmission
  - GPU block-compressed formats too large for transmission
  - Compression to GPU formats slow or unavailable on most clients
  - Nightmare of many GPU/Platform-Specific formats
Image Bits - Solutions Under Discussion

Rate Distortion Optimization (Crunch RDO Mode)

Image → BC[1-5], ETC[12] → RDO → {zlib,zstd} → Inflate → GPU

Supercompression (Crunch CRN mode)

Image → BC[1-5], ETC[12] → Crunch → Decrunch → GPU
Image Bits - Solutions Under Discussion

Universal Transcodable Format

Image ➔ Transcodable format ➔ Crunch ➔ Decrunch ➔ Transcode ➔ ASTC, BC[1-7], ETC[12], PVRRTC ➔ GPU

Can be combined for better performance
KTX File Structure

Mip Level Structure

- **Header**
- **MetaData**
- **Mip level 0**
- **Mip level n**

Array layer 0
- face / z-slice 0
- face / z-slice n

Array layer n
- face / z-slice 0
- face / z-slice n
KTX2 File Structure

- **Header**
  - Data Format Descriptor
  - Metadata
  - Global Compression Scheme
    - CRN
    - LZ (zlib)
    - zstd
    - tANS
- **Mip level 0**
- **Mip level n**
KTX 2 Header Additions

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

- KTXswizzle
  - Indicates desired component mapping for a texture
Data Format Descriptor*

- provides exact description of texel format and color space
  - non-OpenGL and non-Vulkan applications no longer need to understand OpenGL or Vulkan enums to use the image data.
  - provides applications that care about correct color with the necessary information.
  - KTX files can now contain multisample images

* See https://www.khronos.org/dataformat.
Global Compression

- CRN, LZ (zlib) compression.
  - Inclusion of zstd and tANS under discussion.
  - Use only with RDO mode or uncompressed images
- Transcodable format can be encoded by either CRN or RDO so KTX2 also supports it.
Open Issues

- Specification has several unresolved open issues listed inside.
- Please look. I want your opinion.
  - Read the specification at http://github.khronos.org/KTX-Specification/
  - File issues at https://github.com/KhronosGroup/KTX-Specification
WiP Reminder
Watch these places for progress

Crunch GitHub Repo: https://github.com/BinomialLLC/crunch

Improvements to Crunch to support ETC were done by Alexander Suvorov of Unity. His blog describing the work is:

Crunch and the transcoders were developed by Rich Geldreich now at Binomial LLC. Some relevant blogs are:
http://richg42.blogspot.com/2018/06/etc1s-texture-format-encoding.html
http://richg42.blogspot.com/2018/05/some-basis-baseline-universal-format.html

KTX2 specification source: https://github.com/KhronosGroup/KTX-Specification

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

I am about to land a huge change bringing Vulkan support and much easier use when not using OpenGL or Vulkan.